keyup Documentation

Release 1.3.1

Blake Huber

Nov 06, 2022

README

CHAPTER 1

Purpose

keyup automates IAM user access key rotation from the cli by allowing ad hoc or scheduled renewal of your access key credentials via the Amazon API's.

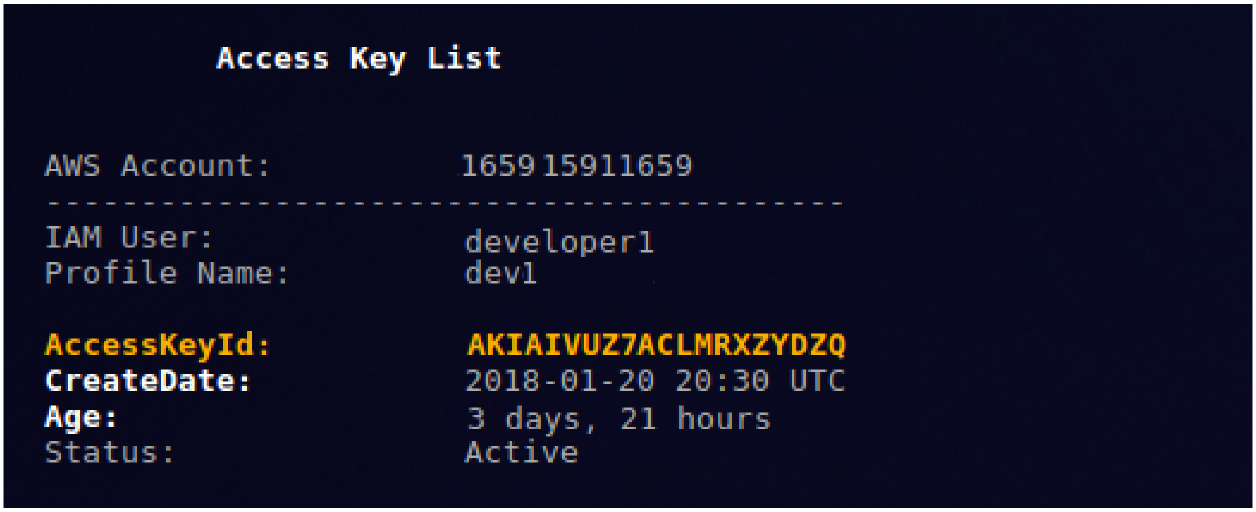
keyup:

- is a safe and reliable way to rotate (renew) access keys to Amazon Web Services as frequently as you wish, with minimal effort and risk.
- requires only the profile name of your IAM user in your local awscli configuration

Features:

- access key rotation via the Amazon APIs
- key rotation includes:
  - creation of new access keys
  - automated installation in the local awscli configuration
  - deprecated key deletion
- automated, unattended key rotation
- rotate keys as frequently as you wish (daily, for example)

(back to Table Of Contents)



```
          Access Key List


AWS Account:            165915911659
------------------------------------------
IAM User:               developer1
Profile Name:           dev1

AccessKeyId:            AKIAIVUZ7ACLMRXZYDZQ
CreateDate:             2018-01-20 20:30 UTC
Age:                    3 days, 21 hours
Status:                 Active
```

CHAPTER 2

Getting Started

Before starting, please take a moment to read the following:

- Frequently Asked Questions (FAQ)
- Use Cases

Other Resources:

- Source Code: keyup bitbucket repository
- Amazon Secure Token Service (STS) Documentation
- Amazon Web Services' Command Line Interface (awscli) Documentation
- **keyup** Open Source License Agreement

Current Release:

- Ensure you are running the most current release. See the most *Current Release*.

(back to Table Of Contents)

CHAPTER 3

Documentation

Online:

- Complete html documentation available at http://keyup.readthedocs.io.

Download: Available via download in the formats below

- pdf format
- Amazon Kindle (epub) format

(back to Table Of Contents)

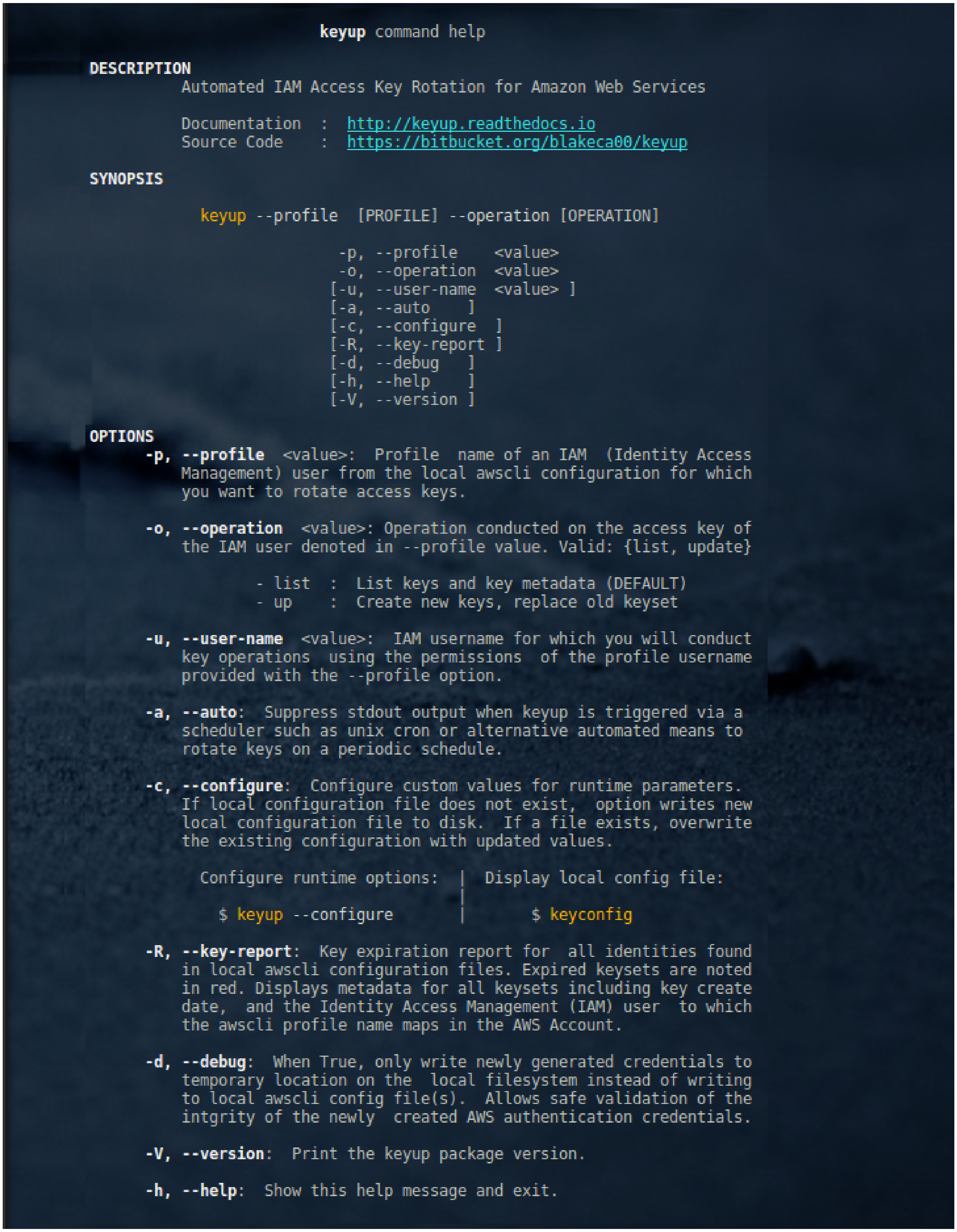
CHAPTER 4

Parameters & Options

Display help menu to see available options and functionality.

```
$ keyup --help
```

(back to Table Of Contents)



```
                          keyup command help

DESCRIPTION
          Automated IAM Access Key Rotation for Amazon Web Services

          Documentation  :  http://keyup.readthedocs.io
          Source Code    :  https://bitbucket.org/blakeca00/keyup

SYNOPSIS

            keyup --profile  [PROFILE] --operation [OPERATION]

                           -p, --profile    <value>
                           -o, --operation  <value>
                          [-u, --user-name  <value> ]
                          [-a, --auto     ]
                          [-c, --configure  ]
                          [-R, --key-report ]
                          [-d, --debug    ]
                          [-h, --help     ]
                          [-V, --version ]

OPTIONS
      -p, --profile  <value>:  Profile  name of an IAM  (Identity Access
          Management) user from the local awscli configuration for which
          you want to rotate access keys.

      -o, --operation  <value>: Operation conducted on the access key of
          the IAM user denoted in --profile value. Valid: {list, update}

                  - list  :  List keys and key metadata (DEFAULT)
                  - up    :  Create new keys, replace old keyset

      -u, --user-name  <value>:  IAM username for which you will conduct
          key operations  using the permissions  of the profile username
          provided with the --profile option.

      -a, --auto:  Suppress stdout output when keyup is triggered via a
          scheduler such as unix cron or alternative automated means to
          rotate keys on a periodic schedule.

      -c, --configure:  Configure custom values for runtime parameters.
          If local configuration file does not exist,  option writes new
          local configuration file to disk.  If a file exists, overwrite
          the existing configuration with updated values.

            Configure runtime options:  |  Display local config file:
                                        |
              $ keyup --configure       |       $ keyconfig

      -R, --key-report:  Key expiration report for  all identities found
          in local awscli configuration files. Expired keysets are noted
          in red. Displays metadata for all keysets including key create
          date,  and the Identity Access Management (IAM) user  to which
          the awscli profile name maps in the AWS Account.

      -d, --debug:  When True, only write newly generated credentials to
          temporary location on the  local filesystem instead of writing
          to local awscli config file(s).  Allows safe validation of the
          intgrity of the newly  created AWS authentication credentials.

      -V, --version:  Print the keyup package version.

      -h, --help:  Show this help message and exit.
```

CHAPTER 5

Author & Copyright

All works contained herein copyrighted via below author unless work is explicitly noted by an alternate author.

License

Back to Table Of Contents

CHAPTER 6

Disclaimer

Code is provided "as is". No liability is assumed by either the code's originating author nor this repo's owner for their use at AWS or any other facility. Furthermore, running function code at AWS may incur monetary charges; in some cases, charges may be substantial. Charges are the sole responsibility of the account holder executing code obtained from this library.

Additional terms may be found in the complete license agreement.

Back to *Purpose* Top

Table Of Contents

CHAPTER 7

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
```

(continues on next page)

(continued from previous page)

```
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)
```

(continues on next page)

(continued from previous page)

```
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:

```
{project}  Copyright (C) {year}  {fullname}
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.

Table Of Contents

CHAPTER 8

Frequently Asked Questions

8.1 General Questions

8.1.1 Q: How do I rotate keys for the default awscli user?

A: Keys operations are conducted for the IAM user configured for the default profile user in the local awscli using the following:

```
$ keyup --profile default --operation up          # rotate (change out) keys
```

List keys for the default awscli user:

```
$ keyup --profile default --operation list        # list key information
```

Back to *Frequently Asked Questions* Top

8.1.2 Q: Which set of access keys will `keyup` replace if my IAM user has 2 sets of keys?

Answer

If you have more than one keyset associated with your IAM user, keyup compares the age of the two keysets in your account. An IAM user identity may be associated with a at a time.

Keyup examines the keys associated with your IAM user and replaces the oldest, *active* keyset.

- Oldest keys are deprecated
- Must be active (enabled)
- If only 1 active set of keys, will replace active and leave the inactive keys untouched.

Back to *Frequently Asked Questions* Index

8.1.3 Q: How often can I rotate my access keys? Is there a limit or does Amazon charge me?

A: There is no (practical) limit to how often you change your access keys.

Amazon does not charge for issuing new access keys. IAM Access Keys are a feature of every AWS Account.

Back to *Frequently Asked Questions* Top

8.1.4 Q: How do I renew access keys for another IAM user account instead of my own account?

A: Assuming your IAM user possess elevalated IAM priviledges to create and delete access keys for other users, you can rotate keys for another user via the following:

```
$ keyup --profile myIAMuser --user-name JoeUser --operation up

    # myIAMuser:  profile name alias for an IAM user with admin IAM permissions
    # JoeUser:    The IAM user account for which you wish to rotate access keys
```

Back to *Frequently Asked Questions* Top

8.2 Logging & Log Configuration

8.2.1 Q: How do I configure logging for `keyup`?

A: Logging is turned off by default for any new installation.

To turn on logging, start the configure menu via the following command:

```
$ keyup --configure
```

Answer 'True' when asked to enable logging.

Next select from one of two options:

- **SYSLOG**: logging to the system log file (/var/log/syslog)
- **FILE**: logging to a file you specify

8.2.2 Q: Why doesn't `keyup` produce any log output?

A: Logging is turned off by default for any new installation of **keyup**.

To turn on logging, start the configure menu via the following command:

```
$ keyup --configure
```

Answer 'True' when asked to enable logging when answering the guided configuration questions.

To verify logging is enabled in your specific keyup configuration, check that `ENABLE_LOGGING` parameter is set to True by viewing the contents of your configuration file with the following command:

```
$ keyconfig
```

It should display syntax similar to below:

```
{
    "KEY_BACKUP": {
        "BACKUP_ENABLE": false,
        "BACKUP_LOCATION": "/home/user/Backup/keysets"
    },
    "KEY_METADATA": {
        "KEYAGE_MAX_DAYS": 30,
        "KEYAGE_MAX_LIMIT": 365,
        "KEYAGE_MIN_DAYS": 1,
        "KEYAGE_WARNING": 2,
        "KEY_DEPRECATION": "AGE",
        "KEY_ENABLE_DELAY": 9
    },
    "LOGGING": {
        "ENABLE_LOGGING": true,
        "LOG_FILENAME": "keyup.log",
        "LOG_MODE": "FILE",
        "LOG_PATH": "/home/user/logs/keyup.log",
        "SYSLOG_FILE": false
    },
```

(continues on next page)

(continued from previous page)

```
    "PROJECT": {
        "CONFIG_DATE": "2019-03-19 16:53 UTC",
        "CONFIG_DIR": ".config",
        "CONFIG_FILENAME": "config.json",
        "CONFIG_PATH": "/home/user/.config/keyup/config.json",
        "CONFIG_SUBDIR": "keyup",
        "CONFIG_VERSION": "1.1.2",
        "HOME": "/home/user",
        "PACKAGE": "keyup"
    },
    "TEMP_CREDENTIALS": {
        "PROFILE_NAMES": []
    }
}
```

Back to *Frequently Asked Questions* Top

8.3 Automated Key Rotation

8.3.1 Q: How do I automate rotation of access keys for my IAM user?

A: Use of scheduler is a common simple way to automate AWS access key rotation.

> On Linux, crontab can be setup as follows:

```
$ crontab -e
```

Once your users' crontab is open (any scheduler will work), enter the following to for daily access key rotation @ 0700 hrs every morning:

```
# keyup auto rotate
0 7 * * *  keyup --profile johndoe --operation up --auto
```

The `--auto` switch supresses all output to stdout, so `keyup` produces no output to the cli. Logging output is retained and the only way to troubleshoot any issues while operating in auto mode; thus, it is highly recommended that you ensure that logging is enabled and working.

If you haven't enabled logging yet, please see:

> *Q: How do I configure logging for keyup?*

Verify your crontab entry was saved by issuing:

```
$ crontab -l
```

You should see the following output:

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
```

(continues on next page)

(continued from previous page)

```
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.

# keyup auto rotate
0 7 * * 5  keyup --profile myUser --operation up --auto
```

Other Crontab Scheduler Examples:

Weekly Rotation every Sat 10:00 AM:

```
# weekly
0 10 * * 6  KEYUP=$(which keyup);  $KEYUP --profile johndoe --operation up --auto
```

Weekly access key rotation (every Friday) for another IAM user, but using your account permissions to perform the rotation:

```
# keyup auto rotate
0 7 * * 5 keyup --profile myUser  --operation up  --auto
```

Back to *Frequently Asked Questions* Top

8.3.2 Q: Is it ok to rotate access keys while I have STS temporary credentials present in my local awscli configuration?

A: `keyup` has special logic to avoid disrupting STS Temporary Credentials if they are present in your local awscli configuration. The following summarizes the logic `keyup` employs when dealing with temporary credentials:

- `keyup` detects when STS credentials are present in the local credentials files
- If temporary credentials are detected, `keyup` attempts to determine if they are active by trying to autheni-cate to Amazon STS using 2 sets of temporary credentials.
- **If authentication fails**: `keyup` assumes the temporary credentials are expired and ignores them. key rotation proceeds automatically as normal
- **If authentication succeeds**: `keyup` assumes the temporary credentials are active and prohibits access key rotation.

Back to *Frequently Asked Questions* Top

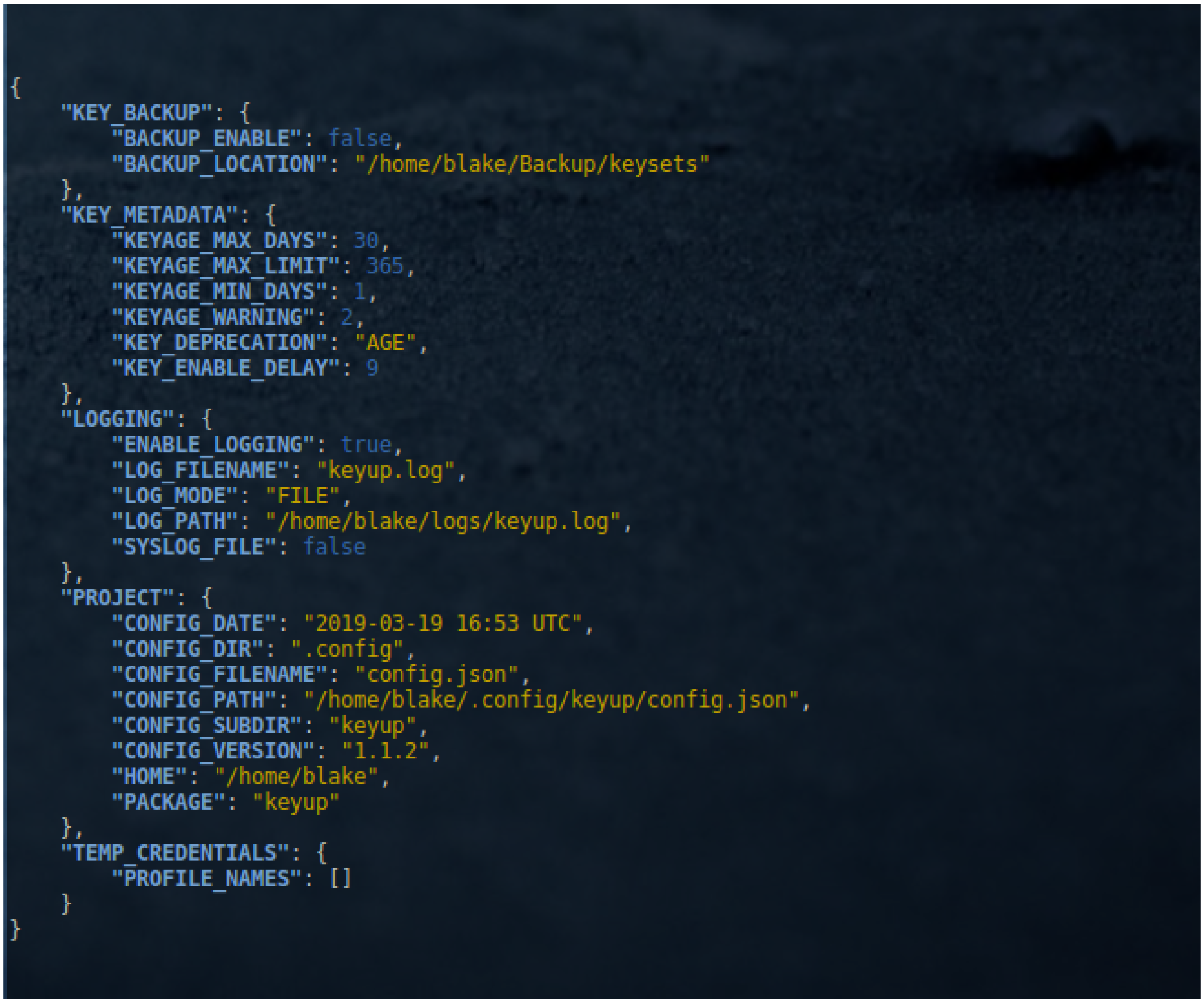
8.4 Miscellaneous Questions

8.4.1 Q: How do I know what options are set in my local config file?

A: Call the `keyconfig` script.

You can display keyup's local configuration file via by calling the console script:

```
$ keyconfig
```



```
{
    "KEY_BACKUP": {
        "BACKUP_ENABLE": false,
        "BACKUP_LOCATION": "/home/blake/Backup/keysets"
    },
    "KEY_METADATA": {
        "KEYAGE_MAX_DAYS": 30,
        "KEYAGE_MAX_LIMIT": 365,
        "KEYAGE_MIN_DAYS": 1,
        "KEYAGE_WARNING": 2,
        "KEY_DEPRECATION": "AGE",
        "KEY_ENABLE_DELAY": 9
    },
    "LOGGING": {
        "ENABLE_LOGGING": true,
        "LOG_FILENAME": "keyup.log",
        "LOG_MODE": "FILE",
        "LOG_PATH": "/home/blake/logs/keyup.log",
        "SYSLOG_FILE": false
    },
    "PROJECT": {
        "CONFIG_DATE": "2019-03-19 16:53 UTC",
        "CONFIG_DIR": ".config",
        "CONFIG_FILENAME": "config.json",
        "CONFIG_PATH": "/home/blake/.config/keyup/config.json",
        "CONFIG_SUBDIR": "keyup",
        "CONFIG_VERSION": "1.1.2",
        "HOME": "/home/blake",
        "PACKAGE": "keyup"
    },
    "TEMP_CREDENTIALS": {
        "PROFILE_NAMES": []
    }
}
```

Back to *Frequently Asked Questions* Top

Table Of Contents

CHAPTER 9

Dependencies

Python3 installed, one of the following versions:

-
-

  Python 3 supported version *is a minimum requirement*.

Linux Operating System, one of the following:

- Redhat Enterprise Linux v7.X
- Ubuntu 14.04, Ubuntu 16.04, and corresponding Ubuntu variants
- Amazon Linux (2016.03+)

Follow to verify the python version installed on linux.

Windows Operating System:

- Windows 7
- Windows 8 (not tested)
- Windows 10

Follow to verify the python version installed on windows.

Installation via Source

- GNU make v4+ binary installed
- bash v4+ v4+
- Python3 v3.6+

- Python Package Manager (pip)

IAM Permissions

- Minimum permissions for each resource (next section)

Back to *Dependencies* top

Table Of Contents

CHAPTER 10

IAM Permissions

Required permissions to use **keyup**.

10.1 Permissions Contents

10.2 Required IAM Policy Permissions

There are minimal permissions required to return valid results for an IAM user. If an iam user referenced in the local awscli configuration file does not have adequate permissions to return access key information, **keyup** will simply skip the user and omit the iam user from the key report.

The following IAM policy must be assigned to each IAM user either via group policy assignment or directly attached to the IAM user identity in the AWS Account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:ListAccountAliases"
      ],
      "Resource": [
```

(continues on next page)

(continued from previous page)

```
        "arn:aws:iam::000000000000:user/*"
      ],
      "Effect": "Allow",
      "Sid": "iamAccountInfo"
    },
    {
      "Action": [
        "iam:CreateAccessKey",
        "iam:DeleteAccessKey",
        "iam:ListAccessKeys",
        "iam:GetAccessKeyLastUsed"
      ],
      "Resource": [
        "arn:aws:iam::742134111111:user/${aws:username}"
      ],
      "Effect": "Allow",
      "Sid": "iamUserChangeOwnAccessKeys"
    }
  ]
}
```

Back to *IAM Permissions* Top

10.3 Recommended IAM Policy Permissions

The policy below is highly recommended as a complement to the required IAM permissions policy above. The recommended policy will require a 2-factor MFA code when attempting to provision resources in an AWS Account.

```
{
  "Statement": [
    {
      "Action": [
        "iam:ChangePassword",
        "iam:CreateLoginProfile",
        "iam:DeleteLoginProfile",
        "iam:GetAccountPasswordPolicy",
        "iam:GetAccountSummary",
        "iam:GetLoginProfile",
        "iam:UpdateLoginProfile"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:iam::000000000000:user/${aws:username}"
      ],
      "Sid": "AllowIndividualUserToSeeTheirAccountInformation"
    },
    {
      "Action": [
        "iam:ListVirtualMFADevices",
        "iam:ListMFADevices"
      ],
      "Effect": "Allow",
      "Resource": [
```

(continues on next page)

(continued from previous page)

```
        "arn:aws:iam::000000000000:user/${aws:username}",
        "arn:aws:iam::000000000000:mfa/*"
      ],
      "Sid": "AllowIndividualUserToListTheirMFA"
    },
    {
      "Action": [
        "iam:CreateVirtualMFADevice",
        "iam:DeactivateMFADevice",
        "iam:DeleteVirtualMFADevice",
        "iam:EnableMFADevice",
        "iam:ResyncMFADevice"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:iam::000000000000:user/${aws:username}",
        "arn:aws:iam::000000000000:mfa/*"
      ],
      "Sid": "AllowIndividualUserToManageThierMFA"
    },
    {
      "Condition": {
        "Null": {
          "aws:MultiFactorAuthAge": "true"
        }
      },
      "Effect": "Deny",
      "NotAction": "iam:*",
      "Resource": "*",
      "Sid": "DoNotAllowAnythingOtherThanAboveUnlessMFAd"
    }
  ],
  "Version": "2012-10-17"
}
```

Back to *IAM Permissions* Top

10.4 IAM Policy Details

The iam policy provides explicit permissions to an individual user to read and update *only* the user's own iam access keys.

Note: The IAM policy permissions allow keyup to rotate access keys *without requiring a Multi-factor Authorization (MFA, 2-factor) code*. This is a recommended policy provided all other permissions to provision or modify resources in the AWS Account mandate a 2-factor MFA code.

an attacker can obtain keys with a compromised iam user account, but can nothing with them.

`iam:ListAccountAliases`:

- Required to query the AWS Account for an assigned alias (human-readable account name).
- Not all AWS Accounts have an alias assigned.

- If no alias is returned, the AWS Account Id is displayed instead.

`iam:CreateAccessKey`:

- Required for keyup to generate new access keys when rotating keys.

`iam:DeleteAccessKey`:

- Required by keyup to delete deprecated access keys after a new set is generated.

`iam:ListAccessKeys`:

- Required by keyup to list access keys for a given IAM user.

`iam:GetAccessKeyLastUsed`:

- Required by keyup to retrieve access key meta data (data about a user's key set).

Back to *IAM Permissions* top

Table Of Contents

CHAPTER 11

Installation

Choose your operating system for correct installation instructions:

Linux Distributions

- *Redhat Enterprise Linux v7.X / Centos 7.X*
- *Ubuntu v16.04+, Ubuntu Variants*
- *Amazon Linux v2016+*

Windows Distributions

- *Microsoft Windows 7*
- *Microsoft Windows 10*

Installation via Source

- *Build Source Code*

Post-Installation

- *Verify Your Installation*

Note

- Generally, any modern Linux distribution may work.
- Python 3 supported version *is a hard requirement*.

11.1 Redhat Enterprise Linux v7.X / Centos 7.X

- Install Python3 Package Manager

```
$ sudo yum install python3-pip
```

- Install awscli

  Detailed instructions can be found in the README located at: https://github.com/aws/aws-cli/

  The easiest method, provided your platform supports it, is via pip.

```
$ pip3 install awscli --user
```

- If you have the aws-cli installed and want to upgrade to the latest version you can run:

```
$ pip3 install --upgrade awscli --user
```

- Installation via pip3 (python3 packages via pip package manager)

```
$ pip3 install keyup --user
```

- *Verify Your Installation*

Back to *Installation* Table of Contents

11.2 Ubuntu v16.04+, Ubuntu Variants

- Install Python3 Package Manager

```
$ sudo apt-get install python3-pip
```

- Install awscli

  Detailed instructions can be found in the README located at: https://github.com/aws/aws-cli/

  The easiest method, provided your platform supports it, is via pip.

```
$ pip3 install awscli --user
```

- If you have the aws-cli installed and want to upgrade to the latest version you can run:

```
$ pip3 install --upgrade awscli --user
```

- Installation via pip3 (python3 packages via pip package manager)

```
$ pip3 install keyup --user
```

- *Verify Your Installation*

Back to *Installation* Table of Contents

11.3 Amazon Linux v2016+

- Install Python3 Package Manager:

```
$ sudo yum install python36-pip

OR

$ sudo yum install python35-pip / python34-pip    # latest your distribution supports
```

- Install keyup:

```
$ pip3 install keyup --user
```

- *Verify Your Installation*

Back to *Installation* Table of Contents

11.4 Microsoft Windows 7

- Install Python3 by downloading the latest Python3 version for Windows
- Install **keyup**:

```
$ pip install keyup
```

- *Verify Your Installation*

Back to *Installation* Table of Contents

11.5 Microsoft Windows 10

- Install Python3 by downloading the latest Python3 version for Windows
- Install **keyup**:

```
$ pip install keyup
```

- *Verify Your Installation*

Back to *Installation* Table of Contents

11.6 Build Source Code

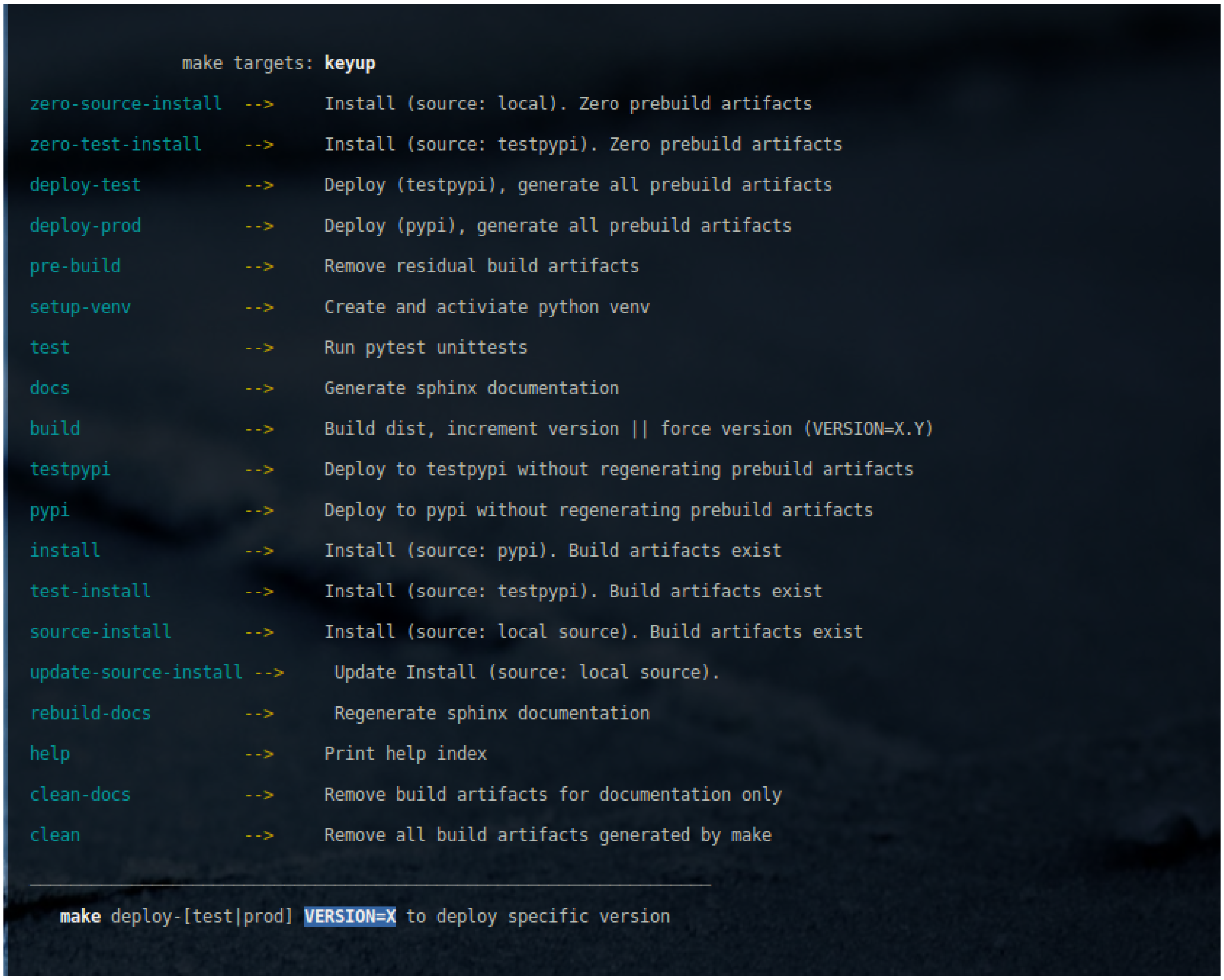
To install locally from source code, do the following:

Check out the `keyup` code repository:

```
$ git clone https://blakeca00@bitbucket.org/blakeca00/keyup.git

$ cd keyup/
```

Display the list of make targets:

```
$ make help
```



```
               make targets: keyup
zero-source-install  -->     Install (source: local). Zero prebuild artifacts
zero-test-install    -->     Install (source: testpypi). Zero prebuild artifacts
deploy-test          -->     Deploy (testpypi), generate all prebuild artifacts
deploy-prod          -->     Deploy (pypi), generate all prebuild artifacts
pre-build            -->     Remove residual build artifacts
setup-venv           -->     Create and activate python venv
test                 -->     Run pytest unittests
docs                 -->     Generate sphinx documentation
build                -->     Build dist, increment version || force version (VERSION=X.Y)
testpypi             -->     Deploy to testpypi without regenerating prebuild artifacts
pypi                 -->     Deploy to pypi without regenerating prebuild artifacts
install              -->     Install (source: pypi). Build artifacts exist
test-install         -->     Install (source: testpypi). Build artifacts exist
source-install       -->     Install (source: local source). Build artifacts exist
update-source-install -->     Update Install (source: local source).
rebuild-docs         -->      Regenerate sphinx documentation
help                 -->     Print help index
clean-docs           -->     Remove build artifacts for documentation only
clean                -->     Remove all build artifacts generated by make

   make deploy-[test|prod] VERSION=X to deploy specific version
```

To install locally in virtual environment, make the install from source target:

```
$ make source-install
```

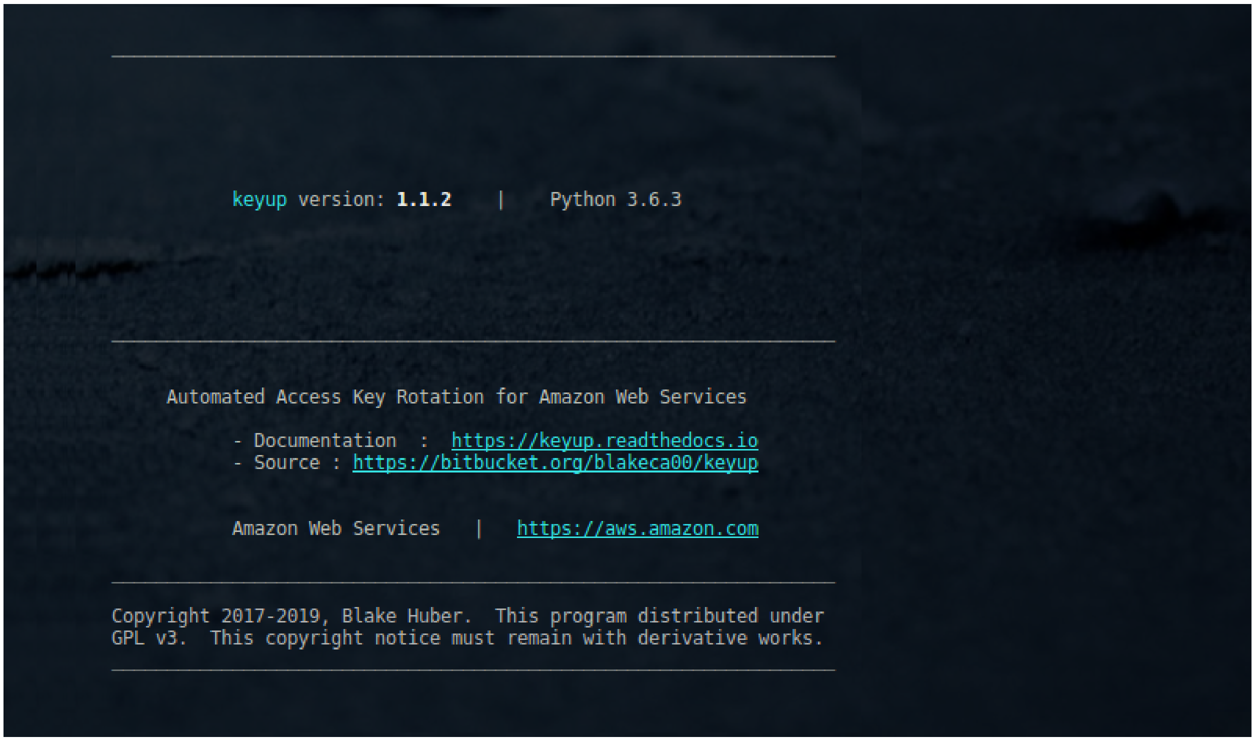
- *Verify Your Installation*

CHAPTER 12

Post-Installation

12.1 Verify Your Installation

```
$ keyup --version
```



```
    ___________________________________________________________________



              keyup version: 1.1.2    |    Python 3.6.3



    ___________________________________________________________________

         Automated Access Key Rotation for Amazon Web Services

               - Documentation  :  https://keyup.readthedocs.io
               - Source : https://bitbucket.org/blakeca00/keyup

               Amazon Web Services   |   https://aws.amazon.com
    ___________________________________________________________________

    Copyright 2017-2019, Blake Huber.  This program distributed under
    GPL v3.  This copyright notice must remain with derivative works.
    ___________________________________________________________________
```

Back to *Installation* Top

Table Of Contents

CHAPTER 13

Use Cases

3 Main Use Cases for keyup:

- *List Access Keys*
- *Rotate Keys (User-Initiated)*
- *Rotate Keys (Automated)*

13.1 List Access Keys

Display access keys for a specific IAM user in an AWS Account.

```
$ keyup --profile [PROFILE] --operation list
```

`[PROFILE]:`

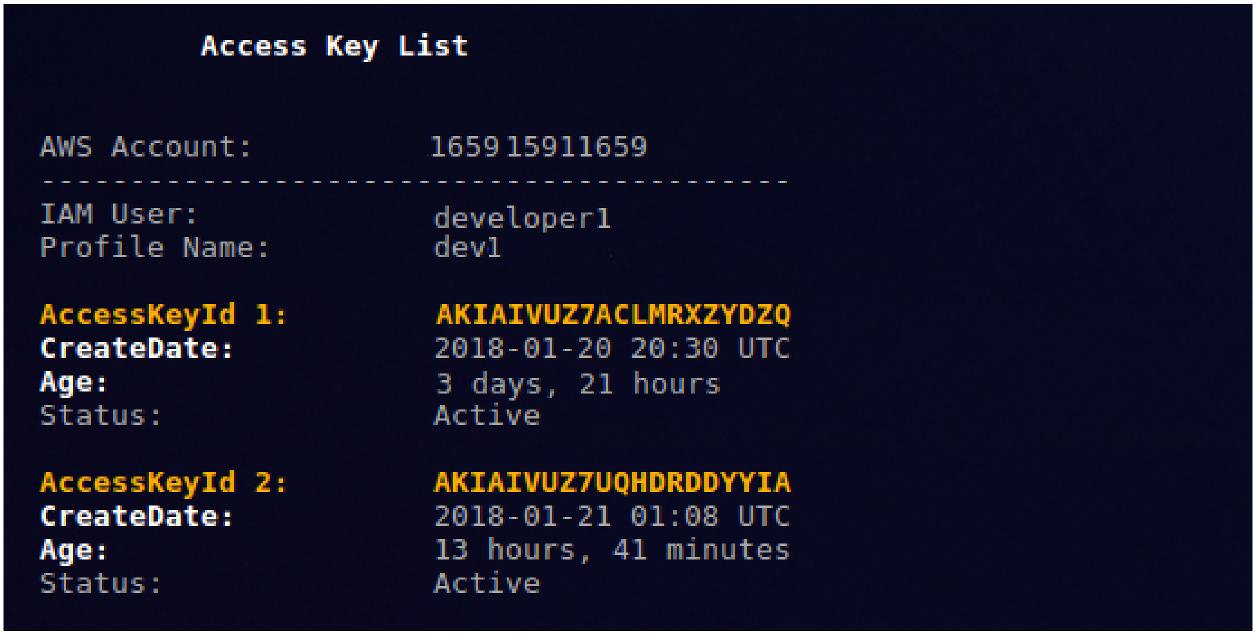
- awscli profile name profile from the local awscli config
- IAM user for whom you wish to conduct key operations

`list:`

- list operation, lists key information for the profile username given in PROFILE.
- Prints report showing access key id, key age (time since creation), and status (active or inactive)

13.2 Rotate Keys (User-Initiated)

Attended mode, manual operation run from cli. Operator can enter MFA code if required



```
          Access Key List


 AWS Account:           1659 15911659
 ------------------------------------------
 IAM User:              developer1
 Profile Name:          dev1

 AccessKeyId 1:         AKIAIVUZ7ACLMRXZYDZQ
 CreateDate:            2018-01-20 20:30 UTC
 Age:                   3 days, 21 hours
 Status:                Active

 AccessKeyId 2:         AKIAIVUZ7UQHDRDDYYIA
 CreateDate:            2018-01-21 01:08 UTC
 Age:                   13 hours, 41 minutes
 Status:                Active
```

```
$ keyup --profile [PROFILE] --operation up
```

`[PROFILE]`:

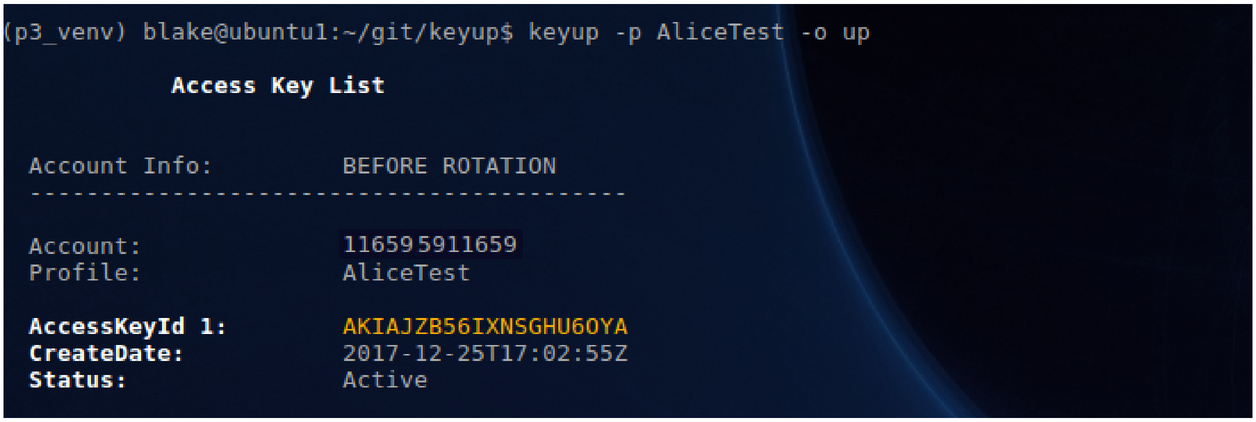
- awscli profile name profile from the local awscli config
- IAM user for whom you wish to conduct key operations

`up`:

- update (rotate, or renew) key operation.
- Creates new access keyset (access key, secret key), installs keys, delete old keys from AWS
- Short for `keyup`, which can also be used

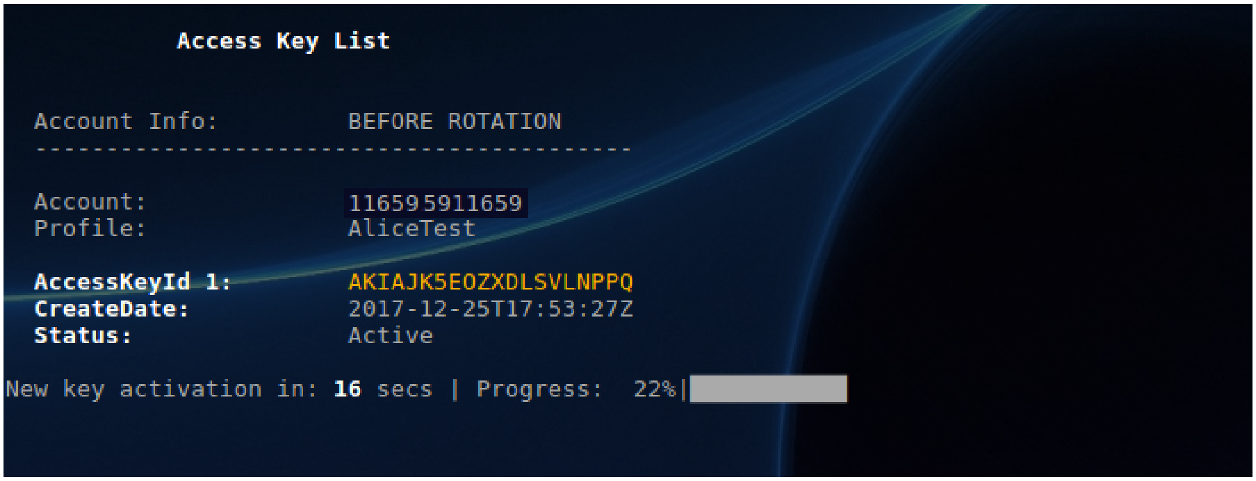
Rotate Keys output (click below to enlarge imges)

(1) Initiate access key rotation ('up'). Displays access keys for user prior to key renewal:



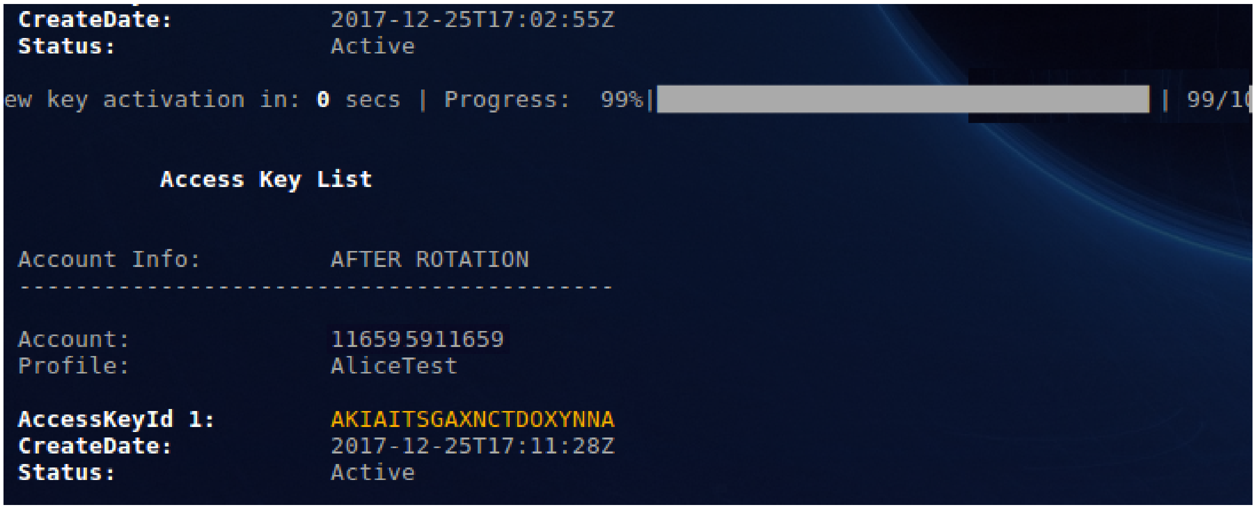
```
(p3_venv) blake@ubuntu1:~/git/keyup$ keyup -p AliceTest -o up

          Access Key List


 Account Info:          BEFORE ROTATION
 ------------------------------------------

 Account:               11659 5911659
 Profile:               AliceTest

 AccessKeyId 1:         AKIAJZB56IXNSGHU6OYA
 CreateDate:            2017-12-25T17:02:55Z
 Status:                Active
```

(2) Progress during key renewal:



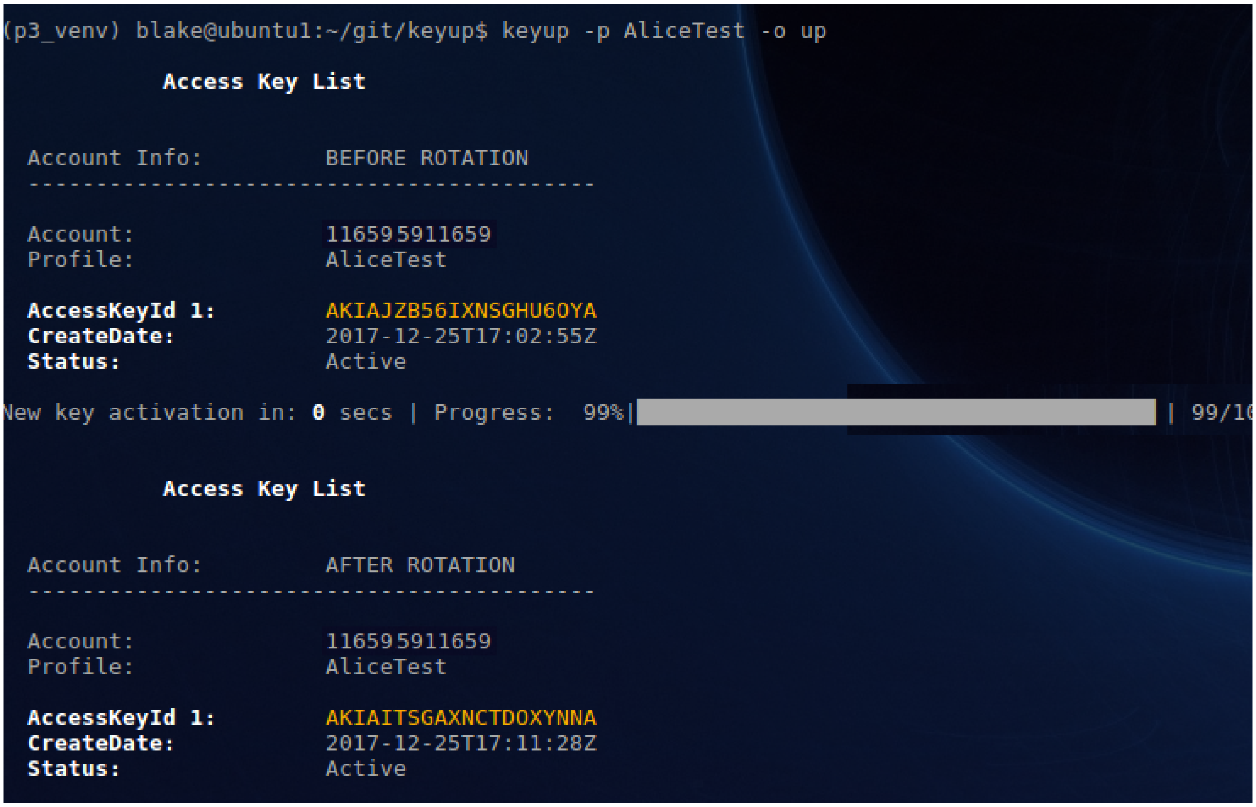
```
          Access Key List


  Account Info:          BEFORE ROTATION
  ------------------------------------------

  Account:               116595911659
  Profile:               AliceTest

  AccessKeyId 1:         AKIAJK5EOZXDLSVLNPPQ
  CreateDate:            2017-12-25T17:53:27Z
  Status:                Active

New key activation in: 16 secs | Progress:  22%|███████
```

(3) Display access keys after key rotation showing new keyset generated:



```
 CreateDate:            2017-12-25T17:02:55Z
 Status:                Active

ew key activation in: 0 secs | Progress:  99%|██████████████████████| 99/10


          Access Key List


 Account Info:          AFTER ROTATION
 ------------------------------------------

 Account:               116595911659
 Profile:               AliceTest

 AccessKeyId 1:         AKIAITSGAXNCTDOXYNNA
 CreateDate:            2017-12-25T17:11:28Z
 Status:                Active
```

(4) End-to-end cli view of the user-initiated key rotation process:



```
(p3_venv) blake@ubuntu1:~/git/keyup$ keyup -p AliceTest -o up
            Access Key List

  Account Info:          BEFORE ROTATION
  ------------------------------------------

  Account:               116595911659
  Profile:               AliceTest

  AccessKeyId 1:         AKIAJZB56IXNSGHU6OYA
  CreateDate:            2017-12-25T17:02:55Z
  Status:                Active

New key activation in: 0 secs | Progress:  99%|██████████████████████████████████████| 99/10

            Access Key List

  Account Info:          AFTER ROTATION
  ------------------------------------------

  Account:               116595911659
  Profile:               AliceTest

  AccessKeyId 1:         AKIAITSGAXNCTDOXYNNA
  CreateDate:            2017-12-25T17:11:28Z
  Status:                Active
```

13.3 Rotate Keys (Automated)

Automated rotation of access keys; typically executed via cron or similar scheduling mechanism

```
$ keyup --profile [PROFILE] --operation up --auto
```

[PROFILE]:

- profile from the local awscli config that is the IAM user for whom you wish to keyup

up:

- up: create new access keys, install them, delete old keys from AWS

--auto:

- For use when calling keyup from an automated scheduler or other programmatic execution mechanism.
- Flag suppresses all stdout output
- Log output to the filesystem is retained.

Back to *Use Cases* Top

Table Of Contents

CHAPTER 14

Key Report

Overview of keyup’s Key Report and required Identity Access Management (IAM) permissions.

14.1 Key Report Contents

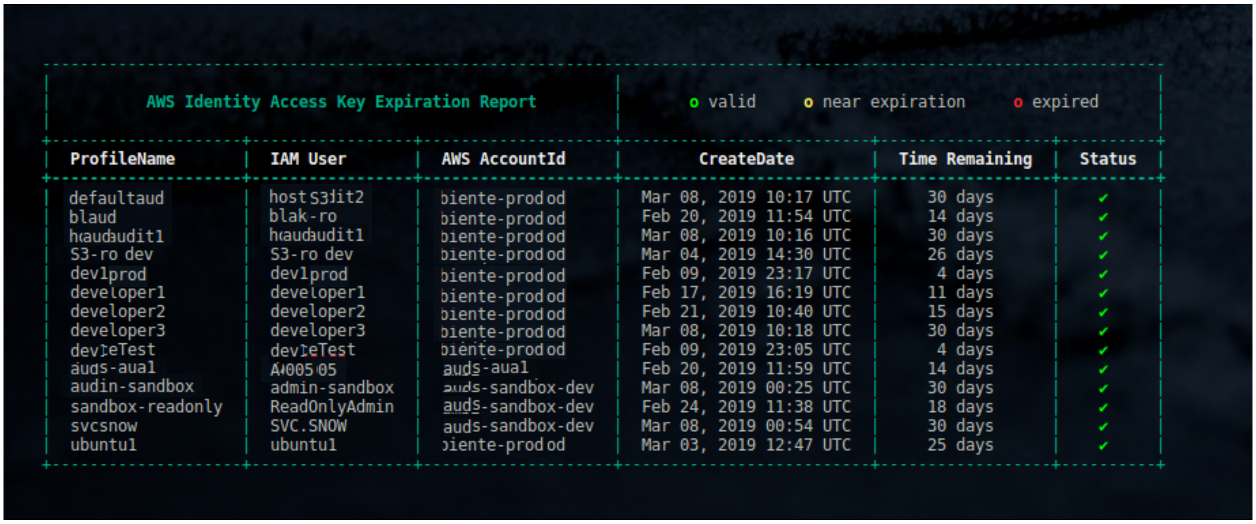
14.2 Report Overview

`keyup` provides a parameter to produce a report for all iam users referenced in the local configuration file. The report shows the expired status of all access keys for these users. This is valuable because it allows the user to understand which iam users require renewed access keys without running **keyup** for each iam user account individually.

Execute the following command to run the key report for all iam users in the local :

```
$ keyup --key-report
```

Back to *Key Report Contents*



```
+-------------------------------------------------------------------------------------------------------------------------------+
|                                                          |                                                                    |
|      AWS Identity Access Key Expiration Report           |        o valid       o near expiration       o expired             |
|                                                          |                                                                    |
+-----------------------+------------------+---------------+-------------------------+-------------------+------------------+
| ProfileName           | IAM User         | AWS AccountId |        CreateDate       | Time Remaining    | Status           |
+-----------------------+------------------+---------------+-------------------------+-------------------+------------------+
| defaultaud            | host S3jit2      | biente-prodod | Mar 08, 2019 10:17 UTC  |     30 days       |    ✓             |
| blaud                 | blak-ro          | biente-prodod | Feb 20, 2019 11:54 UTC  |     14 days       |    ✓             |
| hcaudaudit1           | hcaudaudit1      | biente-prodod | Mar 08, 2019 10:16 UTC  |     30 days       |    ✓             |
| S3-ro dev             | S3-ro dev        | biente-prodod | Mar 04, 2019 14:30 UTC  |     26 days       |    ✓             |
| dev1prod              | dev1prod         | biente-prodod | Feb 09, 2019 23:17 UTC  |      4 days       |    ✓             |
| developer1            | developer1       | biente-prodod | Feb 17, 2019 16:19 UTC  |     11 days       |    ✓             |
| developer2            | developer2       | biente-prodod | Feb 21, 2019 10:40 UTC  |     15 days       |    ✓             |
| developer3            | developer3       | biente-prodod | Mar 08, 2019 10:18 UTC  |     30 days       |    ✓             |
| devIeTest             | devIeTest        | biente-prodod | Feb 09, 2019 23:05 UTC  |      4 days       |    ✓             |
| auds-aual             | A4005 05         | auds-aual     | Feb 20, 2019 11:59 UTC  |     14 days       |    ✓             |
| audin-sandbox         | admin-sandbox    | auds-sandbox-dev | Mar 08, 2019 00:25 UTC |   30 days       |    ✓             |
| sandbox-readonly      | ReadOnlyAdmin    | auds-sandbox-dev | Feb 24, 2019 11:38 UTC |   18 days       |    ✓             |
| svcsnow               | SVC.SNOW         | auds-sandbox-dev | Mar 08, 2019 00:54 UTC |   30 days       |    ✓             |
| ubuntu1               | ubuntu1          | biente-prodod | Mar 03, 2019 12:47 UTC  |     25 days       |    ✓             |
+-----------------------+------------------+---------------+-------------------------+-------------------+------------------+
```

14.3 Required User Permissions

There are minimal Identity Access Management (IAM) permissions required to return valid results for a each IAM user. If an iam user referenced in the local awscli configuration file does not have adequate permissions to return results for the key report, **keyup** will simply skip the user and omit the iam user from the key report.

The following IAM policy must be assigned to each IAM user either via group policy assignment or directly attached to the IAM user identity in the AWS Account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:ListAccountAliases"
      ],
      "Resource": [
        "arn:aws:iam::000000000000:user/*"
      ],
      "Effect": "Allow",
      "Sid": "iamAccountInfo"
    },
    {
      "Action": [
        "iam:CreateAccessKey",
        "iam:DeleteAccessKey",
        "iam:ListAccessKeys",
        "iam:GetAccessKeyLastUsed"
      ],
      "Resource": [
        "arn:aws:iam::742134111111:user/${aws:username}"
      ],
      "Effect": "Allow",
      "Sid": "iamUserChangeOwnAccessKeys"
    }
  ]
}
```

Back to *Key Report Contents*

14.4 Recommended User Permissions

The policy below is highly recommended as a complement to the required IAM permissions policy above. The recommended policy will require a 2-factor MFA code when attempting to provision resources in an AWS Account.

```
{
  "Statement": [
    {
      "Action": [
        "iam:ChangePassword",
        "iam:CreateLoginProfile",
        "iam:DeleteLoginProfile",
        "iam:GetAccountPasswordPolicy",
        "iam:GetAccountSummary",
        "iam:GetLoginProfile",
        "iam:UpdateLoginProfile"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:iam::000000000000:user/${aws:username}"
      ],
      "Sid": "AllowIndividualUserToSeeTheirAccountInformation"
    },
    {
      "Action": [
        "iam:ListVirtualMFADevices",
        "iam:ListMFADevices"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:iam::000000000000:user/${aws:username}",
        "arn:aws:iam::000000000000:mfa/*"
      ],
      "Sid": "AllowIndividualUserToListTheirMFA"
    },
    {
      "Action": [
        "iam:CreateVirtualMFADevice",
        "iam:DeactivateMFADevice",
        "iam:DeleteVirtualMFADevice",
        "iam:EnableMFADevice",
        "iam:ResyncMFADevice"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:iam::000000000000:user/${aws:username}",
        "arn:aws:iam::000000000000:mfa/*"
      ],
      "Sid": "AllowIndividualUserToManageThierMFA"
    },
    {
      "Condition": {
        "Null": {
```

(continues on next page)

(continued from previous page)

```
          "aws:MultiFactorAuthAge": "true"
        }
      },
      "Effect": "Deny",
      "NotAction": "iam:*",
      "Resource": "*",
      "Sid": "DoNotAllowAnythingOtherThanAboveUnlessMFAd"
    }
  ],
  "Version": "2012-10-17"
}
```

Back to *Key Report Contents*

14.5 Policies Explained

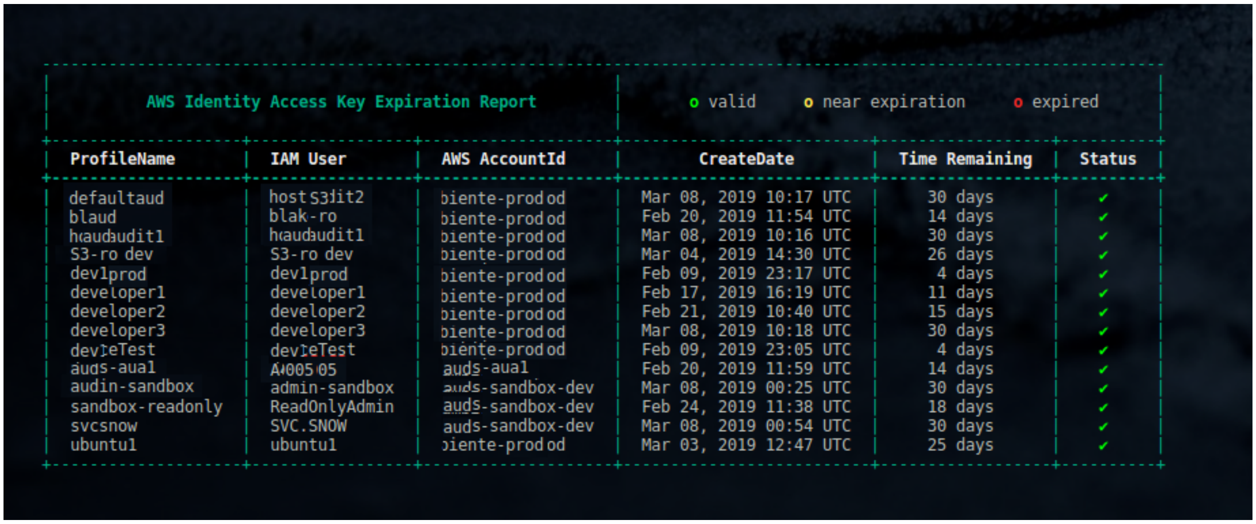
See the *IAM Policy Details* section to gain an understanding of each each IAM policy provision and the permissions granted to a user.

Back to *Key Report Contents*

14.6 Screenshots

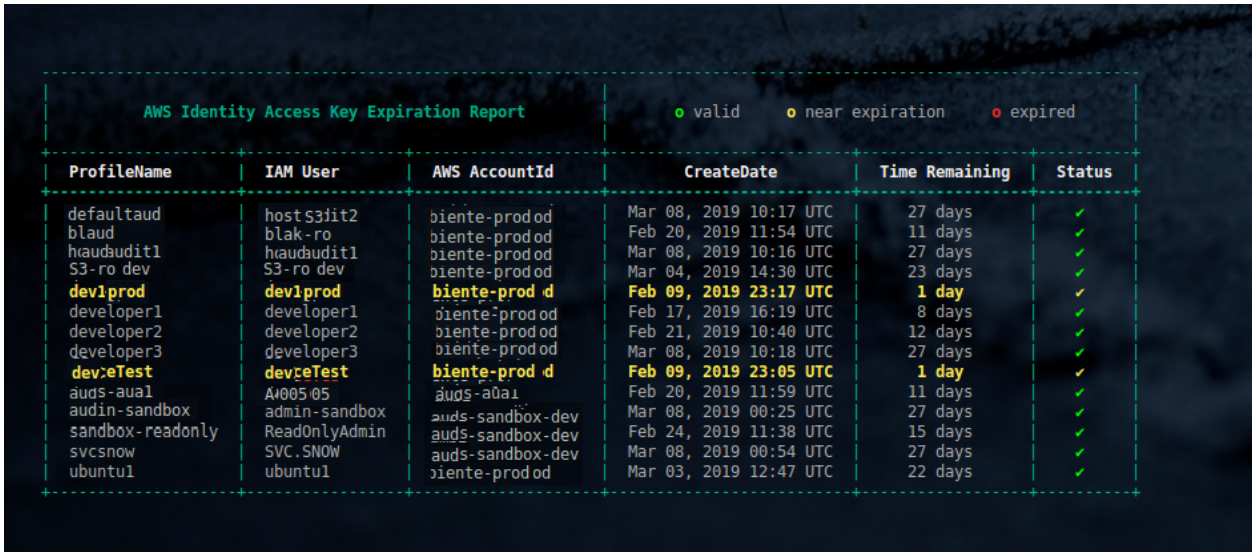
Sample key report screenshots (click to enlarge).

- Key report showing all users with valid access keys (none expired):



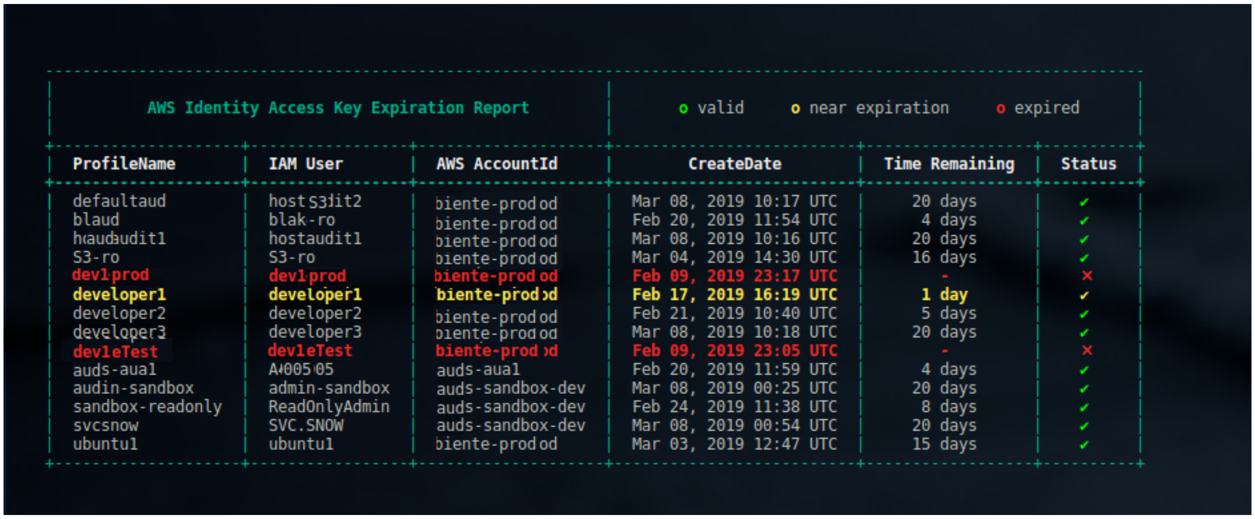
```
-------------------------------------------------------------------------------------------------------------------
|                                                                  |                                                  |
|           AWS Identity Access Key Expiration Report              |     o valid     o near expiration     o expired  |
|                                                                  |                                                  |
+-----------------+-------------+----------------+--------------------------+------------------+---------+
| ProfileName     | IAM User    | AWS AccountId  |        CreateDate        | Time Remaining   | Status  |
+-----------------+-------------+----------------+--------------------------+------------------+---------+
| defaultaud      | host S3it2  | biente-prod od |  Mar 08, 2019 10:17 UTC  |    30 days       |    ✓    |
| blaud           | blak-ro     | biente-prod od |  Feb 20, 2019 11:54 UTC  |    14 days       |    ✓    |
| hcaudaudit1     | hcaudaudit1 | biente-prod od |  Mar 08, 2019 10:16 UTC  |    30 days       |    ✓    |
| S3-ro dev       | S3-ro dev   | biente-prod od |  Mar 04, 2019 14:30 UTC  |    26 days       |    ✓    |
| dev1prod        | dev1prod    | biente-prod od |  Feb 09, 2019 23:17 UTC  |     4 days       |    ✓    |
| developer1      | developer1  | biente-prod od |  Feb 17, 2019 16:19 UTC  |    11 days       |    ✓    |
| developer2      | developer2  | biente-prod od |  Feb 21, 2019 10:40 UTC  |    15 days       |    ✓    |
| developer3      | developer3  | biente-prod od |  Mar 08, 2019 10:18 UTC  |    30 days       |    ✓    |
| devieTest       | devIeTest   | biente-prod od |  Feb 09, 2019 23:05 UTC  |     4 days       |    ✓    |
| auds-aual       | A4005 05    | auds-aual      |  Feb 20, 2019 11:59 UTC  |    14 days       |    ✓    |
| audin-sandbox   | admin-sandbox | auds-sandbox-dev |  Mar 08, 2019 00:25 UTC  |    30 days     |    ✓    |
| sandbox-readonly | ReadOnlyAdmin | auds-sandbox-dev |  Feb 24, 2019 11:38 UTC |    18 days     |    ✓    |
| svcsnow         | SVC.SNOW    | auds-sandbox-dev |  Mar 08, 2019 00:54 UTC  |    30 days     |    ✓    |
| ubuntu1         | ubuntu1     | biente-prod od |  Mar 03, 2019 12:47 UTC  |    25 days       |    ✓    |
+-----------------+-------------+----------------+--------------------------+------------------+---------+
```

- Key report showing keys within 48 hours of expiration:



```
+--------------------------------------------------------------------------------------------------------------------+
|                                                        |                                                           |
|       AWS Identity Access Key Expiration Report        |      o valid     o near expiration     o expired           |
|                                                        |                                                           |
+--------------------+-----------------+-----------------+-------------------------+------------------+--------------+
| ProfileName        | IAM User        | AWS AccountId   |        CreateDate       | Time Remaining   |  Status      |
+--------------------+-----------------+-----------------+-------------------------+------------------+--------------+
| defaultaud         | host S3lit2     | biente-prod od  | Mar 08, 2019 10:17 UTC  |     27 days      |    ✓         |
| blaud              | blak-ro         | biente-prod od  | Feb 20, 2019 11:54 UTC  |     11 days      |    ✓         |
| haudaudit1         | haudaudit1      | biente-prod od  | Mar 08, 2019 10:16 UTC  |     27 days      |    ✓         |
| S3-ro dev          | S3-ro dev       | biente-prod od  | Mar 04, 2019 14:30 UTC  |     23 days      |    ✓         |
| dev1prod           | dev1prod        | biente-prod d   | Feb 09, 2019 23:17 UTC  |      1 day       |    ✓         |
| developer1         | developer1      | biente-prod od  | Feb 17, 2019 16:19 UTC  |      8 days      |    ✓         |
| developer2         | developer2      | biente-prod od  | Feb 21, 2019 10:40 UTC  |     12 days      |    ✓         |
| developer3         | developer3      | biente-prod od  | Mar 08, 2019 10:18 UTC  |     27 days      |    ✓         |
| dev:eTest          | dev:eTest       | biente-prod d   | Feb 09, 2019 23:05 UTC  |      1 day       |    ✓         |
| auds-aua1          | A4005 05        | auds-aua1       | Feb 20, 2019 11:59 UTC  |     11 days      |    ✓         |
| audin-sandbox      | admin-sandbox   | auds-sandbox-dev| Mar 08, 2019 00:25 UTC  |     27 days      |    ✓         |
| sandbox-readonly   | ReadOnlyAdmin   | auds-sandbox-dev| Feb 24, 2019 11:38 UTC  |     15 days      |    ✓         |
| svcsnow            | SVC.SNOW        | auds-sandbox-dev| Mar 08, 2019 00:54 UTC  |     27 days      |    ✓         |
| ubuntu1            | ubuntu1         | biente-prod od  | Mar 03, 2019 12:47 UTC  |     22 days      |    ✓         |
+--------------------+-----------------+-----------------+-------------------------+------------------+--------------+
```

- Key report showing expired keys:



```
+--------------------------------------------------------------------------------------------------------------------+
|                                                        |                                                           |
|       AWS Identity Access Key Expiration Report        |      o valid     o near expiration     o expired           |
|                                                        |                                                           |
+--------------------+-----------------+-----------------+-------------------------+------------------+--------------+
| ProfileName        | IAM User        | AWS AccountId   |        CreateDate       | Time Remaining   |  Status      |
+--------------------+-----------------+-----------------+-------------------------+------------------+--------------+
| defaultaud         | host S3lit2     | biente-prod od  | Mar 08, 2019 10:17 UTC  |     20 days      |    ✓         |
| blaud              | blak-ro         | biente-prod od  | Feb 20, 2019 11:54 UTC  |      4 days      |    ✓         |
| haudaudit1         | hostaudit1      | biente-prod od  | Mar 08, 2019 10:16 UTC  |     20 days      |    ✓         |
| S3-ro              | S3-ro           | biente-prod od  | Mar 04, 2019 14:30 UTC  |     16 days      |    ✓         |
| dev1prod           | dev1prod        | biente-prod od  | Feb 09, 2019 23:17 UTC  |       -          |    ✗         |
| developer1         | developer1      | biente-prod od  | Feb 17, 2019 16:19 UTC  |      1 day       |    ✓         |
| developer2         | developer2      | biente-prod od  | Feb 21, 2019 10:40 UTC  |      5 days      |    ✓         |
| developer3         | developer3      | biente-prod od  | Mar 08, 2019 10:18 UTC  |     20 days      |    ✓         |
| dev1eTest          | dev1eTest       | biente-prod od  | Feb 09, 2019 23:05 UTC  |       -          |    ✗         |
| auds-aua1          | A4005 05        | auds-aua1       | Feb 20, 2019 11:59 UTC  |      4 days      |    ✓         |
| audin-sandbox      | admin-sandbox   | auds-sandbox-dev| Mar 08, 2019 00:25 UTC  |     20 days      |    ✓         |
| sandbox-readonly   | ReadOnlyAdmin   | auds-sandbox-dev| Feb 24, 2019 11:38 UTC  |      8 days      |    ✓         |
| svcsnow            | SVC.SNOW        | auds-sandbox-dev| Mar 08, 2019 00:54 UTC  |     20 days      |    ✓         |
| ubuntu1            | ubuntu1         | biente-prod od  | Mar 03, 2019 12:47 UTC  |     15 days      |    ✓         |
+--------------------+-----------------+-----------------+-------------------------+------------------+--------------+
```

Back to *Key Report Contents* Top

Table Of Contents

CHAPTER 15

Code Summary Index

Module keyup.cli

authenticated	Tests generic authentication status to AWS Account
boto3_session	Summary:
help_menu	Displays help menu contents
get_current_key	Extracts the STS AccessKeyId currently utilised in user's
remove_temporary_credentials	Filers temporary credentials from the list of profile-names
clean_config	Test local awscli config for Active temporary credentails
parse_awscli	Parse, update local awscli config credentials
set_logging	Enable or disable logging per config object parameter
precheck	Verify project runtime dependencies
create_keyset	Creates new access key, secret key pair for iam user
set_keyset	Sets new access keys in memory to execute requests to Amazon APIs
write_keyset	Write out new awscli credentials to local config
write_keyset_backup	Writes newly created keyset to disk provided configuration file flag set:
configure_keyset	Parses local awscli config and reconfigures it with
delete_keyset	Deletes oldest access key credentials associated with a user
options	Parse cli parameter options
package_version	Prints package version and requisite PACKAGE info
shared_credentials_location	Discover alterate location for awscli shared credentials file
source_globals	Source all global variable definitions here
init	Caller function; initializes all functionality
main	End-to-end renew of access keys for a specific profile in local awscli config

Module `keyup.cauth`

authentication	class def for generation and retention of a single set of credentials for generating key-report via a single iam user priviledges
convert	Convert days to hours
discover_account_affiliations	Associates each profile name in local awscli configuration
display_table	Print Table Object offset from left by tabspaces
expired_keys	Convert datetime objects into human readable
format_remaining	Formats days remaining value
print_header	Paints title header grid of a vpt Table
time_remaining	Calculate the days until expiration
`_postprocessing`	
setup_table	Renders Table containing data elements via cli stdout
source_globals	global environment variable definitions
truncate_fields	Truncates table field data to align with max column width
prepare_reportdata	Prints out key expiration info for all profilenames associated with the primary profilename given to access the account information

Back to *Code Summary Index* Top

Table Of Contents

CHAPTER 16

keyup Package

16.1 Module List

- *keyup.cauth module*
- *keyup.cli module*
- *keyup.colors module*
- *keyup.configuration module*
- *keyup.logd module*
- *keyup.help_menu module*
- *keyup.keyconfig module*
- *keyup.map module*
- *keyup.menu module*
- *keyup.statics module*
- *keyup.thread_progress module*
- *keyup.iam_operations module*
- *keyup.script_utils module*
- *keyup.list_ops module*
- *keyup.oscodes_unix module*
- *keyup.oscodes_win module*

16.2 keyup.cauth module

Key Report Generation Module:

- Centralized authentication module for producing Key Report
- Uses threading for concurrent processing

Module Functions:

- **convert:** Converts time units to hours
- **discover_account_affiliations:** maps awscli profile users to corresponding iam user ids
- **expired_keys:** determines if an access keyset is aged beyond max age value in keyup's configuration file
- **display_table:** renders vpt table to cli stdout

class keyup.cauth.**authentication**(*profile*)

Bases: `object`

class def for generation and retention of a single set of credentials for generating key-report via a single iam user priviledges

generate_token(*user*)

keyup.cauth.**convert**(*dt*)

Convert days to hours

keyup.cauth.**discover_account_affiliations**()

Associates each profile name in local awscli configuration to an iam username and an AWS Account Number

Returns: affiliation info, TYPE: dict

keyup.cauth.**display_skipped**(*iam_users*)

Display iam users exceptions skipped in the key report

keyup.cauth.**display_table**(*table*, *exceptions*, *tabspaces=4*)

Print Table Object offset from left by tabspaces

keyup.cauth.**expired_keys**(*dt*)

Convert datetime objects into human readable

keyup.cauth.**format_remaining**(*days: int*)

Formats days remaining value

Returns: days (int) with appropriate color, spacing format applied

keyup.cauth.**prepare_reportdata**(*debug=False*)

Prints out key expiration info for all profilenames associated with the primary profilename given to access the account information

keyup.cauth.**print_header**(*title*, *indent=4*, *spacing=4*)

Paints title header grid of a vpt Table

keyup.cauth.**setup_table**(*user_data*, *exception_list*)

Renders Table containing data elements via cli stdout

keyup.cauth.**source_globals**()

global environment variable definitions

`keyup.cauth.`**`spacing`**`(`*days*`)`

`keyup.cauth.`**`time_remaining`**`(`*dt*`)`
Calculate the days until expiration

`keyup.cauth.`**`truncate_fields`**`(`*element*`)`

Truncates table field data to align with max column width

Returns: truncated element, TYPE: dict or str

Module List

16.3 keyup.cli module

Summary: keyup (python3) | Scripted rotation of access keys for an IAM User.

- Display of key report showing key metadata
- Creation of new access keys
- Keyset installation in awscli local config
- Deletion of deprecated keyset

Author: Blake Huber Copyright Blake Huber, All Rights Reserved.

License: GNU General Public License v3.0 (GPL-3) Additional terms may be found in the complete license agreement: https://bitbucket.org/blakeca00/keyup/src/master/LICENSE.txt

OS Support:

- RedHat Linux, Amazon Linux, Ubuntu & variants
- Windows 7+

Dependencies:

- Requires python3, tested under py3.5 and py3.6

`class` `keyup.cli.`**`SetLogging`**`(`*mode*, *disable=False*`)`
Bases: `object`

Initializes project level logging

Args:

- **mode (str)**: log_mode, either ‘stream’ or ‘FILE’
- **disable (bool)**: when True, disables logging output

Returns: TYPE: bool, Success | Failure

`set``(`*mode*, *disable*`)`
create logger object, enable or disable logging

`keyup.cli.`**`clean_config`**`(`*quiet*`)`

Test local awscli config for Active temporary credentails

Args:

credentials_file (str) location of local awscli credentials file

config (configParser obj) GLOBAL object representing parsed awscli credentials file

Returns: TYPE: bool, Success | Failure

Note: Conditions when `clean_config` returns False (Failure):

If parsed_config contains **active credentials**, key rotation prohibited and keyup will exit.

If parsed_config contains **inactive credentials**, `clean_config` returns `bool True` and key rotation proceeds

`keyup.cli.`**`configure_keyset`**`(`*keyset*, *profile*, *surrogate=”*`)`

Parses local awscli config and reconfigures it with newly created access keys

Args:

keyset (json) access keys, newly created

profile (str) iam user alias in the local awscli config

surrogate (str) iam username on which access key operations are conducted by another iam user denoted in profile

Returns:

parsed (configParser) object configured with new access key signatures

configfile_path (str) os dependent path to awscli credentials file

access_key (str) sts access key string

secret_key (str) sts secret key string

`keyup.cli.`**`create_keyset`**`(`*iam_user*, *profile*, *surrogate=”*`)`

Creates new access key, secret key pair for iam user

Args:

- **access_key (str)**: AccessKeyId of the keyset to delete
- **profile (str)**: iam user alias in the local awscli config
- **surrogate (str): iam username on which access key operations** are conducted by another iam user denoted in profile

Returns: Success | Failure, TYPE: bool, aws access keys (dict)

`keyup.cli.`**`delete_keyset`**`(`*access_key*, *profile*, *surrogate=”*`)`

Deletes oldest access key credentials associated with a user

Args:

- **access_key (str)**: AccessKeyId of the keyset to delete
- **profile (str)**: iam user alias in the local awscli config
- **surrogate (str): iam username on which access key operations** are conducted by another iam user denoted in profile

Returns: TYPE: bool, Success | Failure

keyup.cli.**get_current_key**(*profile_name*, *surrogate=''*)

Extracts the STS AccessKeyId currently utilised in user's profile in the local awscli configuration

Args: profile_name: a username in local awscli profile

Returns: key_id (str): Amazon STS AccessKeyId

Raises: Exception if profile_name not found in config

keyup.cli.**help_menu**()

Displays help menu contents

keyup.cli.**init**()

Caller function; initializes all functionality

keyup.cli.**main**(*operation*, *profile*, *auto*, *debug*, *user_name=''*)

End-to-end renew of access keys for a specific profile in local awscli config

keyup.cli.**options**(*parser*, *help_menu=False*)

Parse cli parameter options

Returns: TYPE: argparse object, parser argument set

keyup.cli.**package_version**()

Prints package version and requisite PACKAGE info

keyup.cli.**parse_awscli**()

Parse, update local awscli config credentials

Args:

user (str) USERNAME, only required when run on windows os

Returns: TYPE: configparser object, parsed config file

keyup.cli.**precheck**()

Verify project runtime dependencies

keyup.cli.**remove_temporary_credentials**(*config_object*, *prefix=''*, *profiles=False*)

Filers temporary credentials from the list of profilenames present in the local awscli configuration

Args:

config_object (configParser object) object containing profilenames from the local awscli credentials file

prefix (str) any prefix prepended to profilenames to indicate temporary role credentials in the local awscli credentials file

profiles (bool) flag indicating return of a profilename list instead of

Returns: configParser object | profilename list (list)

keyup.cli.**set_keyset**(*access_key*, *secret_key*, *clear=False*)

Sets new access keys in memory to execute requests to Amazon APIs during rewrite of local awscli credentials filename

Args:

clear (bool) reset keys set as env variables, if present

`keyup.cli.`**`set_logging`**`(`*cfg_obj*`)`

Enable or disable logging per config object parameter

`keyup.cli.`**`shared_credentials_location`**`()`

Discover alterate location for awscli shared credentials file

Returns: TYPE: str, Full path of shared credentials file, if exists

`keyup.cli.`**`source_globals`**`()`

Source all global variable definitions here

`keyup.cli.`**`write_keyset`**`(`*configparser_obj*, *filename*, *debug=False*`)`

Write out new awscli credentials to local config

Args:

- **configparser_obj (configparser)**: parsed awscli config containing new keyset
- **filename (str)**: path to file to which keyset written
- **debug (bool)**: debug flag

Returns: TYPE: bool, Success | Failure

`keyup.cli.`**`write_keyset_backup`**`(`*keys*, *user*, *quiet*`)`

Writes newly created keyset to disk provided configuration file flag set:

Args:

keys New keyset object

quiet When set, supresses all output to stdout

Returns: TYPE: bool, Success | Failure

Module List

16.4 keyup.colors module

Summary: ANSI color and formatting code class See: http://www.lihaoyi.com/post/BuildyourownCommandLinewithANSIescapecodes.html#256-colors

Args: None

Returns: ansi codes

Raises: None. AttributeError if no code match returns the reset ansi codes

class `keyup.colors.`**`Colors`**

Bases: `object`

Class attributes provide different format variations

AQUA = '\x1b[38;5;14m'

BKGND_BLACK = '\x1b[40m'

BKGND_BLUE = '\x1b[44m'

BKGND_BRIGHT_BLACK = '\x1b[40;1m'

BKGND_BRIGHT_BLUE = '\x1b[44;1m'

BKGND_BRIGHT_CYAN = '\x1b[46;1m'

BKGND_BRIGHT_GREEN = '\x1b[42;1m'

BKGND_BRIGHT_MAGENTA = '\x1b[45;1m'

BKGND_BRIGHT_RED = '\x1b[41;1m'

BKGND_BRIGHT_WHITE = '\x1b[47;1m'

BKGND_BRIGHT_YELLOW = '\x1b[43;1m'

BKGND_CYAN = '\x1b[46m'

BKGND_GREEN = '\x1b[42m'

BKGND_MAGENTA = '\x1b[45m'

BKGND_RED = '\x1b[41m'

BKGND_WHITE = '\x1b[47m'

BKGND_WHITE_BOLD = '\x1b[47;1m'

BKGND_YELLOW = '\x1b[43m'

BLUE = '\x1b[94m'

BOLD = '\x1b[1m'

BRIGHTBLUE = '\x1b[38;5;51m'

BRIGHTCYAN = '\x1b[38;5;36m'

BRIGHTGREEN = '\x1b[38;5;95;38;5;46m'

BRIGHTPURPLE = '\x1b[38;5;68m'

BRIGHTRED = '\x1b[31;1m'

BRIGHTWHITE = '\x1b[38;5;15m'

BRIGHTYELLOW = '\x1b[38;5;11m'

BRIGHTYELLOW2 = '\x1b[38;5;95;38;5;226m'

BRIGHTYELLOWGREEN = '\x1b[38;5;95;38;5;155m'

CYAN = '\x1b[96m'

DARKBLUE = '\x1b[38;5;95;38;5;24m'

DARKCYAN = '\x1b[36m'

DARKGRAY1 = '\x1b[90m'

DARKGRAY2 = '\x1b[38;5;95;38;5;8m'

DARKGREEN = '\x1b[38;5;2m'

END = '\x1b[0m'

`GREEN = '\x1b[92m'`

`ITALIC = '\x1b[3m'`

`LT1GRAY = '\x1b[38;5;95;38;5;245m'`

`LT2GRAY = '\x1b[38;5;249m'`

`ORANGE = '\x1b[38;5;95;38;5;214m'`

`PURPLE = '\x1b[95m'`

`RED = '\x1b[91m'`

`RESET = '\x1b[0;0m'`

`REVERSE = '\x1b[;7m'`

`TITLE = '\x1b[4m\x1b[1m'`

`UNBOLD = '\x1b[22m'`

`UNDERLINE = '\x1b[4m'`

`URL = '\x1b[4m\x1b[96m'`

`WHITE = '\x1b[37m'`

`WHITEGRAY = '\x1b[38;5;95;38;5;250m'`

`YELLOW = '\x1b[93m'`

Module List

16.5 keyup.configuration module

Summary. local_config Module, creates local config file (json) to override default values set in statics module

Module Attributes:

- **current_config (TYPE str):** JSON object resulting from parsing an existing local config file. If no config file exists, object is the starting seed schema stored in statics module
- **config_file (TYPE str):** Path to local config file, usually found in ~/.config/PACKAGE/config.json
- **logger (TYPE logging obj):** system logger, output set by log_mode project-level attribute
- **user_home (TYPE str):** os-specific path to home directory determined in statics module

class keyup.configuration.**ReadConfig**(*local_file=”*)
Bases: object

read(*cfg=”*)
reads values from local config file

class keyup.configuration.**UpdateConfig**(*local_file*, *update=False*, *debug=False*)
Bases: object

Class def for parsing, update, and writing of local fs configuration file

assemble(*arg_dict*)
Summary.

Assembles new parameters in json format for write to new conf file

Returns:

local_config (json) json schema of configuration parameters to be written to local filesystem as new keyup configuration file

config_directory(*cfg*)
Checks config_path to ensure directories exist; if not create

preload_parameters(*current_config*)
Summary.

preloads existing configuration parameters or loads defaults if no preexisting local config file

print_header(*header*)
prints header strings to stdout

update(*cfg*, *debug=False*)
Summary.

updates values in local config file

Args:

cfg (configParser object) parsed local awscli credentials file

debug (boot) debug flag

Returns: TYPE: bool, update Success | Failure

write_config(*parameter_dict*, *cfg*)
create new config file

keyup.configuration.**converge_answer**(*question*, *choices*, *answer=”*)

Summary: prompt user for input until answer in appropriate responses received

Args:

answer (str) user response to question

choices (list) list of valid responses. Responses are strings

Returns:

valid answer (str) valid response from choices

keyup.configuration.**display_table**(*header*, *title=False*, *alignment=’c’*, *border=True*, *offset=45*, *tabspaces=4*, *color=’\x1b[94m’*)
Print Table Object offset from left by tabspaces

keyup.configuration.**exit_processing**(*code=None*, *clear=False*)
Reset terminal screen colors on exit

keyup.configuration.**expand_home_path**(*path*)
Substitute ~ for actual home path

keyup.configuration.**init**(*debug=False*, *cfg=None*)

Summary: Initiates read, write, or update of local_config file

Args:

debug (bool) debug flag

cfg (str) path to fs object containing project k,v config parameters

Returns: TYPE: bool, Success | Failure `keyup.configuration.`**`remove_trailing_slash`**`(`*path*`)`

Removes a trailing slash from provided fs path

`keyup.configuration.`**`set_writeable_location`**`(`*default_location*, *header*, *message*, *confirmation_msg*`)`

Summary:

- Takes user input for filesystem location.
- Tests to ensure location is writable
- Removes trailing slash (if applicable)

Args:

default_location (str) filesystem location if no user input

header (str) Header question/title to print prior to user input

message (str) Message to solicit user input

confirmation_msg (str) Display text after user input accepted

Returns:

fs_location (str) writeable filesystem path

`keyup.configuration.`**`validate_fs_location`**`(`*path*`)`

Summary: Validate existence of a path or create it

Args:

path (str)

Returns: Success | Failure, TYPE: bool

Module List

16.6 keyup.help_menu module

Help Menu Help menu object containing body of help content. For printing with formatting

Module List

16.7 keyup.keyconfig module

Summary: Display module of configuration file contents

`keyup.keyconfig.`**`display_content`**`(`*data_object*, *halt=False*`)`

Summary: Display contents of object correctly whether display on a terminal (tty) or redirected to a file

`keyup.keyconfig.`**`is_tty`**`()`

Summary: Determines if output is displayed to the screen or redirected

Returns: True if tty terminal | False is redirected, TYPE: bool `keyup.keyconfig.`**`option_configure`**`(`*debug=False*, *path=None*`)`

Summary: Initiate configuration menu to customize keyup runtime options. Console script `` `keyconfig` `` invokes this option_configure directly in debug mode to display the contents of the local config file (if exists)

Args:

path (str) full path to default local configuration file location

debug (bool) debug flag, when True prints out contents of local config file

Returns: TYPE (bool): Configuration Success | Failure

Module List

16.8 keyup.logd module

Summary: Project-level logging module

`keyup.logd.`**`getLogger`**`(`**args*, ***kwargs*`)`

Summary: custom format logger

Args: mode (str): The Logger module supprts the following log modes:

- log to console / stdout. Log_mode = 'stream'
- log to file
- log to system logger (syslog)

Returns: logging object | TYPE: logging singleton

`keyup.logd.`**`logprep`**`(`*mode*`)`

Summary: prerequisites for logging to file mode

Args:

mode (str) valid value is 'FILE'; parameter used for logging type validation only

Return: Success | Failure, TYPE: bool

`keyup.logd.`**`mode_assignment`**`(`*mode*`)`
Translates arg to enforce proper assignment

Module List

16.9 keyup.map module

Summary:

- Mapping Module
- maps profile names from local awscli to iam usernames in AWS Account

`keyup.map.`**`map_iam_username`**`(`*username*, *profilename*`)`

Summary: Triangulates if provided username is a profile name from local awscli configuration or an IAM username from AWS

Returns: IAM username (str)

`keyup.map.`**`map_identity`**`(`*profile*`)`

Summary: retrieves iam user info for profiles in awscli config

Args:

user (str) string, local profile user from which the current boto3 session object created

Returns:

iam_user (str) AWS iam user corresponding to the provided profile user in local config

Module List

16.10 keyup.menu module

Curses-menu

`keyup.menu.`**`main`**`()`

Builds main menu, branches to submenus

`keyup.menu.`**`selection_menu_example`**`()`

`keyup.menu.`**`submenu_backup`**`(`*menu_obj*`)`

`keyup.menu.`**`submenu_logging`**`(`*menu_obj*`)`

Module List

16.11 keyup.statics module

Summary.

keyup Project-level Defaults and Settings

- **Local Default Settings**: Local defaults for your specific installation are derived from settings found in:

```
~/.config/keyup/config.json
```

Module Attributes:

- **user_home (TYPE str):** $HOME environment variable, present for most Unix and Unix-like POSIX systems
- **config_dir (TYPE str):** directory name default for stsaval config files (.stsaval)
- **config_path (TYPE str):** default for stsaval config files, includes config_dir (~/.stsaval)
- **key_deprecation (TYPE str):** Deprecation logic that keyup uses when 2 keys exist for a user.

  2 values possible:

- 'AGE': keyup deprecates based on age, replacing the oldest key
- 'AWSCLI': keyup replaces keys currently in the local awscli config

`keyup.statics.`**`import_file_object`**(*filename*)

Summary: Imports block filesystem object

Args:

filename (str) block filesystem object

Returns: dictionary obj (valid json file), file data object

`keyup.statics.`**`os_parityPath`**(*path*)

Converts unix paths to correct windows equivalents. Unix native paths remain unchanged (no effect)

`keyup.statics.`**`read_local_config`**(*cfg*)

Parses local config file for override values

Args:

local_file (str) filename of local config file

Returns: dict object of values contained in local config file

Module List

16.12 keyup.thread_progress module

class `keyup.thread_progress.`**`ProgressBarThread`**(*label='Working'*, *delay=0.1*, *cycles=500*)

Bases: `threading.Thread`

`run`()

Method representing the thread's activity.

You may override this method in a subclass. The standard run() method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the args and kwargs arguments, respectively.

`start`()

Start the thread's activity.

It must be called at most once per thread object. It arranges for the object's run() method to be invoked in a separate thread of control.

This method will raise a RuntimeError if called more than once on the same thread object.

`stop`()

class `keyup.thread_progress.`**`SpinnerThread`**(*label='Working'*, *delay=0.2*)

Bases: `threading.Thread`

`run`()

Method representing the thread's activity.

You may override this method in a subclass. The standard run() method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the args and kwargs arguments, respectively.

start()

Start the thread's activity.

It must be called at most once per thread object. It arranges for the object's run() method to be invoked in a separate thread of control.

This method will raise a RuntimeError if called more than once on the same thread object.

stop()

Module List

16.13 keyup.iam_operations module

Summary.

Prints iam usernames from local awscli configuration. A usernames may be omitted from the output simply by listing them with a space between them after the call:

```
$ python3 iam_users.py default
```

Returns:

Will return all iam usernames in the local configuration except the default user (username "default")

keyup.iam_operations.**awscli_profiles**(*conf*)

Summary.

Returns IAM usernames from local awscli configuration

keyup.iam_operations.**create_userlist**(*content*, *exclusions*)

Summary: Return usernames from configParser object if not in the exclusion list

Args: content (configParser object): local awscli credentials file parsed content exclusions (list): profilenames to be excluded from return

Returns: list of profile names from localhost awscli configuration

keyup.iam_operations.**iam_users**(*profile*)

keyup.iam_operations.**local_profilenames**(*exceptions=[]*)

keyup.iam_operations.**print_profiles**(*config*, *args*)

Execution when no parameters provided

keyup.iam_operations.**shared_credentials_location**()

Summary: Discover alterate location for awscli shared credentials file

Returns: TYPE: str, Full path of shared credentials file, if exists

keyup.iam_operations.**temporary_profilenames**(*conf*, *exclusions=[]*)

Summary: Return usernames from configParser object which represent temporary (iam role) credentials

Args:

conf (str) path to awscli credentials file

exclusions (list) profilenames to be excluded from return

Returns: temporary profile names (role names) from localhost awscli configuration TYPE: str

Module List

16.14 keyup.list_ops module

Summary. List iam keyset operations (read-only)

`keyup.list_ops.`**`list_keys`**(*account*, *profile*, *iam_user*, *surrogate=''*, *stage=None*, *quiet=False*)

Summary. Displays iam user available access keys

Args:

account (str) AWS account number

profile (str) name of the iam user for which we are interrogating keys

iam_user (str) name of the iam user which corresponds to profile name from local awscli configuration

surrogate (str) name of profile user used to execute key operations in place of the profile user

stage (str) stage of key rotation; ie, either BEFORE | AFTER rotation

quiet (bool) No output to stdout (True) | Show output (False)

Returns: TYPE: list, AccessKeyIds listed for the IAM user

`keyup.list_ops.`**`query_keyinfo`**(*account*, *profile*, *surrogate=''*, *quiet=False*)

Summary. boto3 client instantiation and error handling

Args:

account (str) AWS account number

profile (str) name of the iam user for which we are interrogating keys

surrogate (str) name of profile user used to execute key operations in place of the profile user

quiet (bool) No output to stdout (True) | Show output (False)

Returns: boto3 response, TYPE: dict

Module List

16.15 keyup.script_utils module

Command-line Interface (CLI) Utilities Module

Module Functions:

- **bool_assignment:** set bool depending up user answer from stdin
- **config_init:** Initializes config file if none exists
- **debug_mode:** Provide additional log output for debugging
- **read_local_config:** parse local config file

`keyup.script_utils.`**`bool_assignment`**(*arg*, *patterns=None*)

Summary: Enforces correct bool argment assignment

Arg:

arg (*) arg which must be interpreted as either bool True or False

Returns: bool assignment | TYPE: bool

`keyup.script_utils.`**`config_init`**(*config_file*, *json_config_obj*, *config_dirname=None*)

Summary: Creates local config from JSON seed template

Args:

config_file (str) filesystem object containing json dict of config values

json_config_obj (json) data to be written to config_file

config_dirname (str) dir name containing config_file

Returns: TYPE: bool, Success | Failure

`keyup.script_utils.`**`debug_mode`**(*header*, *data_object*, *debug=False*, *halt=False*)
debug output

`keyup.script_utils.`**`import_file_object`**(*filename*)

Summary: Imports block filesystem object

Args:

filename (str) block filesystem object

Returns: dictionary obj (valid json file), file data object

`keyup.script_utils.`**`read_local_config`**(*cfg*)
Parses local config file for override values

Args:

local_file (str) filename of local config file

Returns: dict object of values contained in local config file

Module List

16.16 keyup.oscodes_unix module

Standard OS Module Exit Codes

- See https://docs.python.org/3.6/library/os.html#process-management

Module Attributes:

- exit_codes (dict): exit error codes for Unix, Linux

Module List

16.17 keyup.oscodes_win module

Standard OS Module Exit Codes

- See https://docs.python.org/3.6/library/os.html#process-management

Module Attributes:

- exit_codes (dict): exist error codes for Microsoft Windows

Module List

Table Of Contents

CHAPTER 17

Current Release

17.1 v1.3.0 | Release Notes

Release date: November 1, 2022

Enhancement & Bug Fix release

17.1.1 Features, v1.3.0

- **Prompt Toolkit v3 Compatibility Enhancement**:
  - Upgraded code for dependency to utilise a queue in threaded operations. This is required for compatibility with (current version as of this writing).
  - **Note**: Since this upgrade breaks backward compatibility with prompt-toolkit v2 used in keyup version 1.2.X, only keyup version 1.3+ may be used with prompt-toolkit v3.X installed on your local machine.

17.1.2 Maintenance, v1.3.0

- Resolved :
  - Issue resulted from broken forward compatibility with Amazon Web Services' application programming interface (API) that evolved during the last 2 years as API enhancements were made.
  - Defect caused keyup to create a second Identity & Access Management (IAM) access key instead of renewing the existing access key.
  - Although this error was isolated only to IAM users with a single access key, this issue breaks the major use case for keyup (rotation of a single IAM access key). Issue did not affect key rotation operations for IAM users with 2 registered IAM access keys. Since users with 2 access keys would not have noticed any problems with keyup, this led to a wider misconception that was transient.

(Back to Releases)

CHAPTER 18

Release History

Keyup Release Version	Date
	November 1 , 2022
	November 29, 2020
	January 1, 2020
	Sept 25, 2019
	August 22, 2019
	August 13, 2019
Release 1.1.16	August 10, 2019
	August 3, 2019
	July 21, 2019
	July 19, 2019
	June 1, 2019
	May 29, 2019
	April 25, 2019
	April 5, 2019
	April 2, 2019
	March 26, 2019
	March 24, 2019
	March 19, 2019
	March 17, 2019
	February 10, 2019
	January 23, 2019
	January 02, 2019
	May 20, 2018
	April 07, 2018
	January 22, 2018
	January 19, 2018
	January 06, 2018
	January 02, 2018
	December 30, 2017

CHAPTER 19

Release Note Index

19.1 v1.2.8 | Release Notes

Release date: November 29, 2020

Maintenance Release

19.1.1 Maintenance & Bug Fix, v1.2.8

- **Logging via Syslog**: Fixed errors when enabling syslog monitoring on , , and .

(Back to Releases)

19.2 v1.2.6 | Release Notes

Release date: January 1, 2020

Maintenance Release

19.2.1 Maintenance & Bug Fix, v1.2.6

- **Python Dependency Reduction**: Removed dependency on Python module *tqdm* in favor of prompt-toolkit alternative.

(Back to Releases)

19.3 v1.2.5 | Release Notes

Release date: September 25, 2019

Maintenance Release

19.3.1 Maintenance & Bug Fix, v1.2.5

- **Python 3.8 Preparation**: Python3 Standard library functionality set to be deprecated in eliminated and replaced with 3rd party module, .
- **Python Dependency Versioning**: Minimum dependency versions incremented and tested for the following libraries:
  - ()
  - ()

(Back to Releases)

19.4 v1.2.2 | Release Notes

Release date: August 22, 2019

Maintenance Release

19.4.1 Maintenance & Bug Fix, v1.2.2

- **Post-installation Artifact Placement** resolve post-installation artifact issues with `iam_users.py`
- **Surrogate user-name Fix**: Surrogate user_name functionality fixed. Previous to this release, `--user-name` option would not allow another an admin user to rotate access keys of a subordinate user (with lesser privileges) as designed.

(Back to Releases)

19.5 v1.2.0 | Release Notes

Release date: August 13, 2019

Performance, Reliability, & Stability release

19.5.1 Performance & Stability, v1.2.0

- **Extensive installation testing** from pypi to resolve post-installation artifact issues.
- **sudo/ root user installation**: testing resulted in extensive refactor of `setup.py` and other installation artifacts
- is now the default format for all included documentation for keyup deployments, replacing markdown README and other documnetation in this format.
- **common.py**: new commons (as in "common functionality") module introduced in this release. This is part of the deprecation movement for `script_utils.py` module.

19.5.2 Reliability, v1.2.0

- A runtime test to ensure system logger is enabled and functional added in this release. Prevents catastrophic exception failures when keyup is used in a Docker container or LXC environment.
- Applies to the following linux system loggers:
  –
  –

(Back to Releases)

19.6 v1.1.15 | Release Notes

Release date: August 3, 2019

Maintenance release

19.6.1 Maintenance, v1.1.15

- Anchor `libtools` python library () to a version 0.2+.
- Refactor imports to accommodate minor changes upstream (libtools).

(Back to Releases)

19.7 v1.1.14 | Release Notes

Release date: July 21, 2019

Maintenance release

19.7.1 Maintenance, v1.1.14

- Re-implemented `libtools` python library (pypi registry,) . Previous installation errors in this library dependency have been fixed as of v0.1.10.
- Refactor of statics module, `statics.py` to include all dependent functionality.

(Back to Releases)

19.8 v1.1.13 | Release Notes

Release date: July 19, 2019

Maintenance release

19.8.1 Maintenance, v1.1.13

- Implemented fix for failing test 3 in test module `test_keyset_operations`.
- Partial implementation of `libtools` python library (pypi registry,) .
- Refactor of logging module, `logd.py`. This module to be included for generic use in `libtools` i/o tools library in the near future.

(Back to Releases)

19.9 v1.1.12 | Release Notes

Release date: June 1, 2019

Maintenance release

19.9.1 Maintenance, v1.1.12

- Implemented fix for Ubuntu 18.04 / Ubuntu 18.10 indexing of python3 post install artifacts relative to python's home directory.
- `setup.py`: Update to post install logic to ensure bash completion artifacts placed correctly by *data_files* operation.

(Back to Releases)

19.10 v1.1.11 | Release Notes

Release date: May 29, 2019

Maintenance release

19.10.1 Maintenance, v1.1.11

- Key Report output table formatting and alignment improvements
- `setup.py`: Update to home directory determination to ensure multi-platform compatibility

---

(Back to Releases)

19.11 v1.1.10 | Release Notes

Release date: April 25, 2019

Maintenance release

19.11.1 Maintenance, v1.1.10

- Some IAM usernames were skipped in the key report bc of authentication problems; updated key report to name iam usernames excluded from the report to avoid confusion.

19.11.2 Documentation, v1.1.10

- Added IAM Permissions section outlining required permissions for keyup functionality

---

(Back to Releases)

19.12 v1.1.8 | Release Notes

Release date: April 5, 2019

Documentation Release

19.12.1 Maintenance, v1.1.8

- None

19.12.2 Documentation, v1.1.8

- Updated Sphinx configuration with *autosummary* extension, added Code Summary Index section.

(Back to Releases)

19.13 v1.1.7 | Release Notes

Release date: April 2, 2019

Maintenance Release

19.13.1 Maintenance, v1.1.7

- Bug Fix update, correction for updated color codes in .

(Back to Releases)

19.14 v1.1.5 | Release Notes

Release date: March 26, 2019

Documentation Release

19.14.1 Documentation, v1.1.5

- Various restructuredtext updates to support readthedocs.io

(Back to Releases)

19.15 v1.1.3 | Release Notes

Release date: March 24, 2019

Documentation Release

19.15.1 Documentation, v1.1.3

- Multiple updates, Key Report section.
- now enabled with the library for richer syntax highlight effects.

19.15.2 Maintenance, v1.1.3

- Test Modules: Fixed automated test failures resulting from refactoring in Release 1.1.0

(Back to Releases)

19.16 v1.1.0 | Release Notes

Release date: March 17, 2019

Feature Release

19.16.1 Features, v1.1.0

- Key Report added.
- New documentation section detailing Key Report use and content.
- Python3 for asynchronous processing.

19.16.2 Maintenance, v1.1.0

- **Module Size Reductions**:
  - Lengthy modules separated into a series smaller modules to increase maintainability.

(Back to Releases)

19.17 v1.0.6 | Release Notes

Release date: February 10, 2019

Maintenance Release

19.17.1 Features, v1.0.6

- N/A

19.17.2 Maintenance, v1.0.6

- **setup.py**:
  - add valid_os_shell method to PostInstall Class to prohibit execution of post install unless *nix + bash shell environment.
  - pin python dependencies to version number
- **Post-Commit Hook**:
  - fixed git functionality when triggered via post-commit hook
  - See hooks/ directory in official git repository for code example
- **Bug Fix, Various**

(Back to Releases)

19.18 v1.0.5 | Release Notes

Release date: January 23, 2019

Maintenance Release

19.18.1 Features, v1.0.5

- N/A

19.18.2 Maintenance, v1.0.5

- **Dropped Python 3.5 Support**: Python 3.6 feature elements such as f-strings prohibit compatibility with Python 3.5
- **Bug Fix, Various**: Issue 34 resolved.

(Back to Releases)

19.19 v1.0.1 | Release Notes

Release date: January 2, 2019

Feature + Maintenance Release

19.19.1 Features, v1.0.1

- **Bash Autocompletion**: Pressing <tab><tab> when used on Linux under Bash shell will autocomplete command line parameters.
- When using `--profile` or `--user-name` command-line options, bash completion displays iam username from the local cli configuration.

19.19.2 Maintenance, v1.0.1

- **Exception tqdm module**: Fixed exception thrown by current (v4.27.0+) versions of tqdm module under Linux

---

(Back to Releases)

19.20 v0.9.8 | Release Notes

Release date: May 20, 2018

Maintenance Release

19.20.1 Features, v0.9.8

- **Configuration Menu**: Rework of configuration menu to provide error-free functionality on windows platform
- **Automated Tests**: Updates to automated pretest setup

19.20.2 Fixed, v0.9.8

- **Various Bug Fixes**: Fixed various minor issues

(Back to Releases)

19.21 v0.9.6 | Release Notes

Release date: April 7, 2018

19.21.1 Features, v0.9.6

- **Backup Copy of New Keysets:** If set in the configuration file, `keyup` will write all newly generated keysets to both the local awscli configuration as well as a backup filesystem location on disk.

19.21.2 Fixed, v0.9.6

- **Local Configuration File Deletion:** some previous installations of `keyup` versions experience premature deletion of the local configuration file located at:

  ```
  ~/.config/key/config.json
  ```

  Upgrading to version **0.9.6** will remedy this problem; after which you should regenerate the local configuration file by executing:

  ```
  $ keyup --configure
  ```

- **Various Bug Fixes**: Fixed various minor issues

(Back to Releases)

19.22 v0.9.1 | Release Notes

Release date: Jan 22, 2018

19.22.1 Features, v0.9.1

- **Awareness of STS Temporary Credentials.** `keyup` is now temporary-credential aware in that it will detect when the local awscli config contains temporary credentials. If it does, keyup will test up to 2 profiles to determine if the credentials are currently ACTIVE. If ACTIVE, key rotation will not be be permitted. If temp credentials are inactive, key operations proceed normally. When writing to the local config, keyup can keyup can interrupt automated jobs utilising temporary credentials.
- **Console Script: for displaying local config file contents.** Console script `keyconfig` can now be used to display the contents of The local configuration file, if one exists. If the local config file does not exist yet because the user has not yet to run the configuration menu, then a message informing the user how to generate a local configure is displayed.
- **Documentation: FAQ.** Full Frequently Asked Questions section on readthedocs completed and available.
- **Improved Exception Handling** `keyup` now handles IAM usernames that do not exist more gracefully during list-keys or key rotation operations.

19.22.2 Fixed, v0.9.1

- **Various Bug Fixes**: Fixed various minor issues
- **IAM User Reporting Inaccuracies**: Previously reporting IAM Usernames as 'None'

(Back to Releases)

19.23 v0.8.12 | Release Notes

Release date: Jan 19, 2018

19.23.1 Features, v0.8.12

- **Conduct key operations for another user (–user-name Parameter)** Profile user can be used to list or rotate keys for another user given in the –user-name parameter. This enables key operations when a profile is not represented in the local awscli configuration.
- **Key Age Hooks: Hooks put in place for keyup rotate functionality when key age is between** key_min age and key_max age. Beneficial for automated (ie, scheduled) key rotation so that keys only refreshed when older than the min age. Avoids premature key rotation when executing in automated mode via scheduler.

19.23.2 Fixed, v0.8.12

- **Configuration Script**: Fixed issues resolving answers to configuration questionnaire
- **Key Age Inaccuracies**: Fixed key age across multi-day periods. Now accurate.

(Back to Releases)

19.24 v0.8.8 | Release Notes

Release date: Jan 06, 2018

19.24.1 Features

- **List Keys Report**: Enhanced with Key Age, formatted, human-readable CreateDate.
- **Documentation Updates**: more complete documentation with release notes

19.24.2 Fixed

- **Configuration Script**: Additional issues fixed with configuration questionnaire, which now interprets answers multiple dimension, including a case-insensitive manner.
- **Stdout Formatting**: New facility to frame and format messages to stdout.

(Back to Releases)

19.25 v0.8.7 | Release Notes

Release date: Jan 02, 2018

19.25.1 Features, v0.8.7

- **Auto Mode**: Auto mode initiated with the *–auto* switch will silence all output to stdout.

19.25.2 Fixed, v0.8.7

- **Configuration Script**: Configuration questionnaire initiated by *keyup –configure* had many issues, most now fixed.

(Back to Releases)

19.26 v0.8.3 | Release Notes

Release date: Dec 30, 2017

19.26.1 Documentation Release

- **ReadTheDocs.io**: Initial sphinx auto document generation, release to: http://keyup.readthedocs.io.

(Back to Releases)

Table Of Contents

CHAPTER 20

Module Index

- modindex

CHAPTER 21

Site Index

- genindex

CHAPTER 22

Search Project

- search

Python Module Index

Index