IP Liberator Documentation

Release 0.2.2

Wagner Macedo

Sep 18, 2019

Contents:

1 IP Liberator **1**
- 1.1 Features . . . 1
- 1.2 Installation . . . 1
- 1.3 Quickstart . . . 1
- 1.4 Credits . . . 2

2 Installation **3**
- 2.1 Stable release . . . 3
- 2.2 From sources . . . 3

3 Usage **5**
- 3.1 Quickstart . . . 5

4 Contributing **7**
- 4.1 Types of Contributions . . . 7
- 4.2 Get Started! . . . 8
- 4.3 Pull Request Guidelines . . . 9
- 4.4 Tips . . . 9
- 4.5 Deploying . . . 9

5 Credits **11**
- 5.1 Development Lead . . . 11
- 5.2 Contributors . . . 11

6 History **13**
- 6.1 0.2.2 (Unreleased) . . . 13
- 6.2 0.2.1 (2019-04-19) . . . 13
- 6.3 0.2.0 (2019-04-19) . . . 14
- 6.4 0.1.1 (2018-10-16) . . . 14
- 6.5 0.1.0 (2018-09-27) . . . 14

7 Indices and tables **15**

CHAPTER 1

IP Liberator

A command line utility to update AWS Security Groups rules.

- Free software: GNU General Public License v3
- Documentation: https://ip-liberator.readthedocs.io.

1.1 Features

- Update a list of security groups of your AWS account at once.
- Grant access to informed ports for your current IP address or an informed IP.
- Read profile files in JSON with all the information needed to contact.
- Fit for use as script (e.g. to update your dynamic IP regularly).

1.2 Installation

```
$ pip install ip-liberator
```

1.3 Quickstart

Consider a file `/path/my-profile.json` with the following content:

```
{
  "credentials": {
    "access_key": "<AWS_ACCESS_KEY>",
    "secret_key": "<AWS_SECRET_KEY>",
    "region_name": "<AWS REGION>"
  },
  "config": {
    "operator": "John",
    "services": [
      {
        "name": "FTP+SFTP",
        "port": "21-22"
      },
      {
        "name": "HTTPS",
        "port": "443"
      }
    ],
    "security_groups": [
      "sg-<GROUP_ID_1>",
      "sg-<GROUP_ID_2>"
    ]
  }
}
```

Using the profile defined above will create or update two entries in the informed security groups:

- **John FTP+SFTP** granting access for the current IP the ports 21 and 22.
- **John HTTPS** granting access for the current IP the port 443.

To accomplish it, simply run:

```
$ ip-liberator --profile /path/my-profile.json
Authorizing rules ['John FTP+SSH', 'John HTTPS'] to IP 192.30.253.112/32
- sg-<GROUP_ID_1>
- sg-<GROUP_ID_2>
```

1.4 Credits

1.4.1 Authors

- Wagner Macedo <wagnerluis1982@gmail.com> (maintainer)

CHAPTER 2

Installation

2.1 Stable release

To install IP Liberator, run this command in your terminal:

```
$ pip install ip-liberator
```

This is the preferred method to install IP Liberator, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for IP Liberator can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/wagnerluis1982/ip-liberator
```

Or download the tarball:

```
$ curl  -OL https://github.com/wagnerluis1982/ip-liberator/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 3

Usage

TODO: replace quickstart with a well detailed usage.

3.1 Quickstart

Consider a file `/path/my-profile.json` with the following content:

```
{
  "credentials": {
    "access_key": "<AWS_ACCESS_KEY>",
    "secret_key": "<AWS_SECRET_KEY>",
    "region_name": "<AWS REGION>"
  },
  "config": {
    "operator": "John",
    "services": [
      {
        "name": "FTP+SFTP",
        "port": "21-22"
      },
      {
        "name": "HTTPS",
        "port": "443"
      }
    ],
    "security_groups": [
      "sg-<GROUP_ID_1>",
      "sg-<GROUP_ID_2>"
    ]
  }
}
```

Using the profile defined above will create or update two entries in the informed security groups:

- **John FTP+SFTP** granting access for the current IP the ports 21 and 22.
- **John HTTPS** granting access for the current IP the port 443.

To accomplish it, simply run:

```
$ ip-liberator --profile /path/my-profile.json
Authorizing rules ['John FTP+SSH', 'John HTTPS'] to IP 192.30.253.112/32
- sg-<GROUP_ID_1>
- sg-<GROUP_ID_2>
```

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/wagnerluis1982/ip-liberator/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

4.1.4 Write Documentation

IP Liberator could always use more documentation, whether as part of the official IP Liberator docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/wagnerluis1982/ip-liberator/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *ip-liberator* for local development.

1. Fork the *ip-liberator* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/ip-liberator.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv ip-liberator
$ cd ip-liberator/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 ip_liberator tests
$ python setup.py test or py.test
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for 3.6 onwards, and for PyPy. Check https://travis-ci.org/wagnerluis1982/ip-liberator/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_ip_liberator
```

4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CHAPTER 5

Credits

5.1 Development Lead

- Wagner Macedo <wagnerluis1982@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog, and this project adheres to Semantic Versioning.

6.1 0.2.2 (Unreleased)

This release marks a breaking change. Now the script "tags" recorded entries in the security groups, e.g. `[ip-liberator] SSH John` instead of only `SSH John`. That helps to identify what IP Liberator added and what was added by hand.

By default, the tag is **ip-liberator**, but can be change through the new option `--tag`. If the user wants the previous behavior, i.e. without a tag, he or she must pass the option `--no-tag`.

6.1.1 Added

- Add option `--operator` to change the profile operator.
- Add short option `-p` for `--profile`
- Add option `--version` to show current script version.

6.1.2 Changed

- Add option `--tag` to identify entries added by the script.
- Migrate build system to Poetry

6.2 0.2.1 (2019-04-19)

- Fix documentation

6.3 0.2.0 (2019-04-19)

This release marks as the first to be published to PyPI.

No new functionality was added. The version was changed was to place a history mark.

- Added documentation.
- Added full coverage tests.
- Code refactoring.

6.4 0.1.1 (2018-10-16)

- Better console output.
- Added option `--revoke-only`.
- Don't reauthorize if the IP address is already in the security group.
- Authorizing and revoking in batch to be more efficient.
- Bugfixes

6.5 0.1.0 (2018-09-27)

- Added option `--my-ip` to inform an IP address explicitly.
- Show in console the security groups being processed.
- Allow use as script by reading JSON as external config.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search