S3analystDocumentation

Release 0.1.1

Loïc Messal

Sep 16, 2017

Contents

Contents:

CHAPTER 1

Coveo DevOps Challenge

The Challenge

Your challenge, should you choose to accept it, is to develop an AWS S3 storage analysis tool. To test your tool, you will have to create a free Amazon account (if you don't already have one).

Specifications

The tool is a shell command line utility (could be either Windows, Mac or Linux) that returns informations over all S3 buckets in an Amazon account.

The tool must returns the following informations:

- Bucket name
- Creation date (of the bucket)
- Number of files
- Total size of files
- Last modified date (most recent file of a bucket)

The following options should be supported:

- Ability to get the size results in bytes, KB, MB, ...
- Organize the information by storage type (Standard, IA, RR)
- Filter the results in a list of buckets (bonus point for regex support)
- Ability to group information by regions

Some additional features that could be useful (optional)

It would be nice to support prefix in the bucket filter (e.g.: s3://mybucket/Folder/SubFolder/log*). It may also be useful to organize the results according to the encryption type, get additional buckets informations (life cycle, cross-region replication, etc.) or take into account the previous file versions in the count + size calculation.

Some statistics to check the percentage of space used by a bucket, or any other good ideas you could have, are more than welcome.

Rules

- Your are free to use the programming language and the SDK of your choice.
- We will test your work over our environment (which contains millions of files). The overall performance of your tool will be evaluated.
- Your code must be made available as a git fork of our challenge or any other public version control software.

Advices

- **Try to design and implement your solution as you would do for real production code**. Show us how you create clean, maintainable code that does awesome stuff. Build something that we'd be happy to contribute to. This is not a programming contest where dirty hacks win the game.
- Feel free to add more features! Really, we're curious about what you can think of. We'd expect the same if you worked with us.
- Documentation and maintainability is a plus.
- Don't you forget those unit tests.

CHAPTER 2

S3 analyst

An AWS S3 storage analysis tool.

- Free software: GNU General Public License v3
- Documentation: http://devops-coding-challenge.readthedocs.io.

Features

- TODO

Credits

This package was created with Cookiecutter *and the* `audreyr/cookiecutter-pypackage` project template.

Cookiecutter

audreyr/cookiecutter-pypackage

CHAPTER 3

Installation

Stable release

To install S3_analyst, run this command in your terminal:

```
$ pip install s3_analyst
```

This is the preferred method to install S3_analyst, 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.

From sources

The sources for S3_analyst can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/tofull/s3_analyst
```

Or download the tarball:

```
$ curl  -OL https://github.com/tofull/s3_analyst/tarball/master
```

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

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

CHAPTER 4

Usage

To use S3_analyst in a project:

```
import s3_analyst
```

CHAPTER 5

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:

Types of Contributions

Report Bugs

Report bugs at https://github.com/tofull/devops-coding-challenge/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.

Fix Bugs

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

Implement Features

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

Write Documentation

S3_analyst could always use more documentation, whether as part of the official S3_analyst docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/tofull/devops-coding-challenge/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 :)

Get Started!

Ready to contribute? Here's how to set up `s3_analyst` for local development.

1. Fork the `s3_analyst` repo on GitHub.
2. Clone your fork locally::

```
$ git clone git@github.com:tofull/devops-coding-challenge.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:

   For python2: `$ mkvirtualenv devops-coding-challenge-python2` or for python3: `$ mkvirtualenv --python=/usr/bin/python3 devops-coding-challenge-python3`

   Then install all the requirements:

```
$ cd devops-coding-challenge/
$ pip install -r requirements_dev.txt
$ python setup.py develop
```

   You're now ready to make magic with programming.

4. Install the hooks to keep the documentation updated:

```
$ ln -s ../../hooks/pre-commit .git/hooks/pre-commit
$ ln -s ../../hooks/post-commit .git/hooks/post-commit
```

5. Create a branch for local development::

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

   Now you can make your changes locally.

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

```
$ flake8 s3_analyst tests
$ python setup.py test or py.test
$ tox
```

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

7. 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
```

8. Submit a pull request through the GitHub website.

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 Python 2.7 and 3.5, and for PyPy. Check https://travis-ci.org/tofull/s3_analyst/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ py.test tests.test_s3_analyst
```

CHAPTER 6

Credits

Development Lead

- Loïc Messal loic.messal@orange.fr

Contributors

None yet. Why not be the first?

CHAPTER 7

Bibliography

Here are some links to understand choices.

Project tools

- Structure your python project
  - cookiecutter,
  - virtualenv,
  - git flow,
  - unittest with pytest,
  - doc coverage test,
  - tox,
  - documentation with sphinx,
  - pypi packaging,
  - travisCI,
  - readthedocs CI
- Understand generated files from cookiecutter and build an opensource CLI
  - cookiecutter,
  - MANIFEST.in,
  - setup.cfg,
  - setup.py,
  - docopts
- Best practice for using git flow

  - git flow
- Master the use of virtual environments (FR)
- Using python 3 inside a virtual environment
- Cookiecutter - pypackage

Documentation tool - sphinx

- Using markdown in sphinx (documentation tool) thanks to recommonmark

Hooks

Because I would automate the update of docs/FromMakefile [because of `include ../file.md` is missing in markdown, even with recommonmark]

- git hook add files to commit
- git hook example to understand how it works
- useful git hook example

Unittest tools

Tox

- Why I keep tox.ini away from setup.cfg

Pytest

- Understanding fixture

Versionning tools

Bumpversion

- Understand the tool

CHAPTER 8

History

0.1.0 (2017-09-10)

- First release on PyPI.

CHAPTER 9

Indices and tables

- genindex
- modindex
- search