A Lot of Effort Documentation

Release 0.4.1

Audrey Roy

Sep 27, 2017

Contents

Contents:

CHAPTER 1

A Lot of Effort

Instantly deploy static HTML sites to S3 at the command line.

I created this out of frustration, after spending a lot of effort trying to find a PyPI package that did this without problems.

Documentation

The full documentation is at http://alotofeffort.rtfd.org.

Quickstart

Install it:

```
pip install alotofeffort
```

Configure Boto the standard way in *~/.boto*:

```
[Credentials]
aws_access_key_id = ...
aws_secret_access_key = ...
```

Then use it to deploy a static HTML website to an S3 bucket:

```
$ alotofeffort www/ mybucket
```

Features

- Uses standard Boto configuration.

- Prints the S3 endpoint URL after deploying.
- Auto-configures the bucket to be a website, with all files public.
- Only files that have changed get uploaded. Files are checked for changes by comparing the local and remote MD5 hashes of the files.
- Never auto-deletes. In fact, it doesn't delete files at all! (In the future, it will check if any files need to be deleted from S3, and prompt you before deleting anything.)

CHAPTER 2

Installation

Install the “alotofeffort” package

At the command line:

```
$ easy_install alotofeffort
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv alotofeffort
$ pip install alotofeffort
```

Configure boto

Save the following in *~/.boto*:

```
[Credentials]
aws_access_key_id = ...
aws_secret_access_key = ...
```

Replace ... with your AWS access credentials, of course.

CHAPTER 3

Usage

Deploy a static website with this command:

```
$ alotofeffort www/ mybucket
```

- *www/*: A directory containing the static HTML/JS/CSS to be deployed.
- *mybucket*: The name of your S3 bucket.

CHAPTER 4

Contributing

Contributions are welcome!

Submitting Feedback

The best way to send feedback is to file an issue at https://github.com/audreyr/alotofeffort/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.

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 :)

Getting Started

Here's how to set up *alotofeffort* for local development.

1. Fork the *alotofeffort* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/alotofeffort.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 alotofeffort
$ cd alotofeffort/
$ 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 alotofeffort tests
    $ python setup.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.

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 2.6+ and 3.3+. Check https://travis-ci.org/audreyr/alotofeffort/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ python -m unittest tests.test_alotofeffort
```

CHAPTER 5

Credits

Development Lead

- Audrey Roy <audreyr@gmail.com>

Contributors

None yet. Why not be the first?

CHAPTER 6

History

0.4.0 (2015-09-15)

- Upgraded boto to 2.38.0.
- Added tox envs for Python 3.3, 3.4, 3.5.
- PEP 8 cleanup.
- README cleanup.
- Improvements to setup.py.

0.3 (2013-07-27)

- Only files that have changed get uploaded. Files are checked for changes by comparing the local and remote MD5 hashes of the files.

0.2 (2013-07-17)

- It works on Python 2.6 and 2.7.

0.1 (2013-07-14)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search