Python DNS Failover Documentation

Release 0.1.0

Marc Cerrato

April 07, 2016

Contents

Contents:

CHAPTER 1

Python DNS Failover

Python script to automatically update DNS records for a bunch of servers participating in a Round-Robin DNS setup.

- Free software: BSD license
- Documentation: http://python-dns-failover.rtfd.org.

This project is mainly based on @dotcloud's work on his project autodnsfailover.

The main difference is that **python-dns-failover** script is aimed to **run in an external machine** while autodnsfailover is to run in the participating servers themselves.

1.1 DNS Backends

A DNS backend is responsible for updating the DNS records. It must provide the following methods:

- **get_a_records(fqdn)**: Returns the list of ip adresses records associated with the given FQDN.
- **add_a_record(fqdn, ip)**: Adds a resource record of type A to the DNS list. Optionally return the new record created.
- **delete_a_record(fqdn, ip)**: Deletes all DNS A-type resource records targeting the given ip. Optionally return the number of deleted records.

The constructor will typically be implementation-dependent, and allow to set the credentials and/or the zone to act upon.

1.1.1 Available DNS Backends

- **CloudFlareDNS**
  - *email*: E-mail address of your CloudFlare account.
  - *key*: API key of your CloudFlare account.
  - *zone*: target DNS full qualified domain name.
  - *ttl*: TTL of record in seconds. 1 = Automatic, otherwise, value must in between 120 and 4,294,967,295 seconds. Defaults to 1.
  - *url*: CloudFlare client gateway interface url. Defaults to 'https://www.cloudflare.com/api_json.html'.

1.2 TODO

- Documentation
- Testing

CHAPTER 2

Installation

At the command line:

```
$ easy_install python-dns-failover
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv python-dns-failover
$ pip install python-dns-failover
```

CHAPTER 3

Usage

To use Python DNS Failover in a project:

```
import dns_failover
```

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/marccerrato/python-dns-failover/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" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

4.1.4 Write Documentation

Python DNS Failover could always use more documentation, whether as part of the official Python DNS Failover 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/marccerrato/python-dns-failover/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 *python-dns-failover* for local development.

1. Fork the *python-dns-failover* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/python-dns-failover.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 python-dns-failover
$ cd python-dns-failover/
$ 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 dns_failover 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.

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 Python 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/marccerrato/python-dns-failover/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

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

CHAPTER 5

Credits

5.1 Development Lead

- Marc Cerrato <marccerrato@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

6.1 0.1.0 (2013-12-29)

- Core functionality.
- CloudFlare DNS backend.
- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search