

DNS as code with Octodns

Versioning and keep track of your dns records changes and automate all the thing via travis-ci

Who I am

Matteo Valentini

Developer @ Nethesis (mostly Infrastrutture Developer)

- <https://github.com/Amygos>
- https://twitter.com/_Amygos

Problem to solve

Assigned task:

- Move current DNS management from a web console based to something more modern, on premise or in the “Cloud”.

Constraints:

- One day to production
- The solution must be versionable and/or programmable.

Why octoDNS

- **Files based configurations:** configurations and zones definition are stored in files and can be easily committed in to a CVS
- **Multi provider:** you don't have to choose only one provider
- **Providers agnostic:** you can use the same zone definition with different providers



octoDNS

Overview

“In the vein of [infrastructure as code](#) OctoDNS provides a set of tools & patterns that make it easy to manage your DNS records across multiple providers. The resulting config can live in a repository and be [deployed](#) just like the rest of your code, maintaining a clear history and using your existing review & workflow.” from: <https://github.com/github/octodns>

- Created and used by GitHub for manage their DNS infrastructure
- Released as OSS project at “Wed Mar 15 15:38:10 2017 -0700”
- YAML configurations files format

Simple step up: config.yaml

```
---
providers:
  config:
    class: octodns.provider.yaml.YamlProvider
    directory: ./zones
    enforce_order: false
  do:
    class: octodns.provider.digitalocean.DigitalOceanProvider
    token: env/DO_TOKEN

zones:
  acme.org.:
    sources:
      - config
    targets:
      - do
```

Simple step up: zones/acme.org.yaml

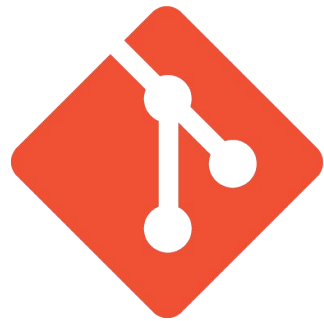
```
---
"":
  - type: MX
    ttl: 600
    values:
      - priority: 10
        value: mail.acme.org.
  - type: A
    ttl: 600
    value: 1.2.3.4
```

```
www:
  ttl: 600
  type: A
  value: 1.2.3.4
```


Usage

- **install:** `$ pip install --user octodns`
- **use:**
 - **verify:** `$ octodns-validate --config-file=config.yaml`
 - **test:** `$ octodns-sync --config-file=config.yaml`
 - **apply:** `$ octodns-sync --config-file=config.yaml --doit`
- **Directory layout:**

```
.
├── config.yaml
├── zones
│   └── acme.org.yaml
```



git

Why

- History of DNS record changes
- Relative easy rollback in case of errors
- Add a “Who” and “Why” to DNS changes
- Facilitate the review process of proposed DNS changes



Travis CI

Benefit of automation

- **Continuous integration:** run a automatic verification test for every proposed change.
- **Continuous deployment:** automatic apply of change when merge the pull request to master.
- **Avoid credential leak:** make possible for an user to do privileged action without acknowledgment of any secret key or token.

How to travis-ci work: /.travis.yml

```
language: python
```

```
cache: pip
```

```
install: pip install octodns
```

```
script:
```

```
- octodns-validate --config-file=config.yaml
```

```
- octodns-sync --config-file=config.yaml
```

```
deploy:
```

```
  provider: script
```

```
  script: octodns-sync --config-file=config.yaml --doit
```

```
on:
```

```
  branch: master
```

How to travis-ci work: environment variables

Environment Variables

Notice that the values are not escaped when your builds are executed. Special characters (for bash) should be escaped accordingly.

DO_TOKEN





Please make sure your secret key is never related to the repository, branch name, or any other guessable string. For more tips on generating keys [read our documentation](#).

Name

Value



Display value in
build log

Add

Pull Request workflow

1. User make a pull request with the requested changes
2. Travis-ci test the pull request
3. User ask a review of a pull request
4. Administrator start the review
5. Administrator approve the changes
6. User merge the pull request to master
7. Travis-ci apply the pull request changes

Multi providers setup

Add a backup provider

```
---
providers:
  config:
    class: octodns.provider.yaml.YamlProvider
    directory: ./zones
    enforce_order: false
  do:
    class: octodns.provider.digitalocean.DigitalOceanProvider
    token: env/DO_TOKEN
  cf:
    class: octodns.provider.cloudflare.CloudflareProvider
    email: env/CF_EMAIL
    token: env/CF_TOKEN
    cdn: false

zones:
  acme.org.:
    sources:
      - config
    targets:
      - do
      - cf
```

Move the zone between two different providers

```
---
providers:
  do:
    class: octodns.provider.digitalocean.DigitalOceanProvider
    token: env/DO_TOKEN
  cf:
    class: octodns.provider.cloudflare.CloudflareProvider
    email: env/CF_EMAIL
    token: env/CF_TOKEN
    cdn: false

zones:
  acme.org.:
    sources:
      - do
    targets:
      - cf
```

Quirks & Recommendations

Quirks

- Travis-ci secure variables are not available to untrusted builds triggered by pull requests from another repository.

script:

```
- >
  if [ "$TRAVIS_SECURE_ENV_VARS" = "false" ]; then
    export DO_TOKEN="";
    export CF_EMAIL="";
    export CF_TOKEN="";
  fi;
  octodns-validate --config-file=config.yaml
- if [ "$TRAVIS_SECURE_ENV_VARS" = "true" ]; then octodns-sync --config-file=config.yaml; fi
```

Recommendations

- **Protect the master branch**, permit writes on master only via pull requests
- Make mandatory for a pull request to be **up to date before merge**
- Block merge until almost **one review approve the change**

Mission Accomplishment!

- Move from a web based “point and click” paradigm to a “infrastructure as code” paradigm
- Dns management are now versionable
- start in the morning and to production in the late afternoon! (ok, actually it was fully in production the day after, we want to make sure that all the records are migrated correctly ;))

Thanks for listening!

Questions?