
terraformlintingcli Documentation

Release 0.1.0

Costas Tyfoxylos

May 25, 2018

Contents

1	terraformlintingcli	3
1.1	Features	3
2	Installation	5
3	Usage	7
4	Contributing	9
4.1	Submit Feedback	9
5	terraformlintingcli	11
5.1	terraformlintingcli package	11
6	Credits	13
6.1	Development Lead	13
6.2	Contributors	13
7	History	15
8	0.1 (2018-05-24)	17
9	Indices and tables	19
	Python Module Index	21

Contents:

A cli to lint and check file positioning for terraform resources

- Documentation: <http://terraformlintingcli.readthedocs.io/en/latest/>

1.1 Features

- Lints terraform resources based on provided rules
- Enforces filename positioning for terraform resources based on provided rules

CHAPTER 2

Installation

At the command line:

```
$ pip install terraformlintingcli
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv terraformlintingcli  
$ pip install terraformlintingcli
```

Or, if you are using pipenv:

```
$ pipenv install terraformlintingcli
```


CHAPTER 3

Usage

To use terraform-lint:

```
terraform-lint --help

[-h] [--log-config LOGGER_CONFIG]
      [--log-level {debug,info,warning,error,critical}]
      -r rules.yaml -p positioning.yaml -s
      tf_plans_dir

Cli to lint naming conventions of terraform plans based on a provided rule set
```

optional arguments:

```
-h, --help            show this help message and exit
--log-config LOGGER_CONFIG, -l LOGGER_CONFIG
                        The location of the logging config json file
--log-level {debug,info,warning,error,critical}, -L {debug,info,warning,error,
critical}
                        Provide the log level. Defaults to INFO.
-n naming.yaml, --naming naming.yaml
-p positioning.yaml, --positioning positioning.yaml
-s tf_plans_dir, --stack tf_plans_dir
```

```
terraform-lint -s dir_with_tf_files/ -n naming.yaml -p positioning.yaml
```

```
Naming convention not followed on file prfxa015-pc01/compute.tf for resource prfxa015-
pc01 for field tags.Name
  Regex not matched : ^prfx[dtaps]a[0-9]{3}-[a-z]{1,3}[0-9]{2}$
  Value              : prfxa015-pc01
```

```
# naming.yaml should follow the following schema
#
# Schema([{'resource': basestring,
#         'regex': is_valid_regex,
```

(continues on next page)

(continued from previous page)

```
#           Optional('fields'): [{'value': basestring,
#                                'regex': is_valid_regex}]])
#
# Example
---
- resource: terraform_resource_name
  regex: .* # regex to lint terraform id
  fields:
    - value: tags.Name
      regex: ^cust[dtaps](?:ewl)-pc[0-9]{2}$ # regex to lint the name of the tag
    - value: tags.Other
      regex: ^cust[dtaps](?:ewl)-other[0-9]{2}$ # regex to lint the name of the tag
```

```
# positioning.yaml should follow the following schema
#
# Schema({And(basestring, lambda x: x.endswith('.tf')): [basestring]})
#
# Example
_data.tf:
  - terraform
  - data
_provider.tf:
  - provider
_variables.tf:
  - variable
compute.tf:
  - azurerm_app_service
  - azurerm_app_service_plan
  - azurerm_virtual_machine
  - aws_instance
```

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

4.1 Submit Feedback

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.

4.1.1 Get Started!

Ready to contribute? Here's how to set up *terraformlintingcli* for local development.

1. Clone your fork locally:

```
$ git clone git@github.com:schubergphilis/terraformlintingcli.git
```

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

```
$ mkvirtualenv terraformlintingcli
$ cd terraformlintingcli/
$ python setup.py develop
```

3. Create a branch for local development:

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

Now you can make your changes locally.

4. Commit your changes and push your branch to the server:

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

5. Submit a merge request

5.1 terraformlintingcli package

5.1.1 Submodules

5.1.2 terraformlintingcli.terraformlintingcli module

Main code for terraformlintingcli

```
class terraformlintingcli.terraformlintingcli.ReadableDirectory (option_strings,
                                                                dest,
                                                                nargs=None,
                                                                const=None,
                                                                default=None,
                                                                type=None,
                                                                choices=None,
                                                                re-
                                                                quired=False,
                                                                help=None,
                                                                metavar=None)
```

Bases: argparse.Action

Argparse action for a directory that is readable

```
class terraformlintingcli.terraformlintingcli.ReadableFile (option_strings,
                                                                dest, nargs=None,
                                                                const=None,
                                                                default=None,
                                                                type=None,
                                                                choices=None,
                                                                required=False,
                                                                help=None,
                                                                metavar=None)
```

Bases: argparse.Action

Argparse action for a file that is readable

`terraformlintingcli.terraformlintingcli.get_arguments()`

Gets us the cli arguments.

Returns the args as parsed from the argparser.

`terraformlintingcli.terraformlintingcli.main()`

Main method.

This method holds what you want to execute when the script is run on command line.

`terraformlintingcli.terraformlintingcli.setup_logging(args)`

Sets up the logging.

Needs the args to get the log level supplied

Parameters `args` – The arguments returned gathered from argparse

5.1.3 terraformlintingcli.terraformlintingcliexceptions module

Custom exception code for terraformlintingcli

5.1.4 Module contents

terraformlintingcli package

Import all parts from terraformlintingcli here

CHAPTER 6

Credits

6.1 Development Lead

- Costas Tyfoxylos <ctyfoxylos@schubergphilis.com>

6.2 Contributors

None yet. Why not be the first?

CHAPTER 7

History

CHAPTER 8

0.1 (2018-05-24)

- First release

CHAPTER 9

Indices and tables

- `genindex`
- `modindex`
- `search`

t

terraformlintingcli, [12](#)

terraformlintingcli.terraformlintingcli,
[11](#)

terraformlintingcli.terraformlintingcliexceptions,
[12](#)

G

`get_arguments()` (in module `terraformlintingcli.terraformlintingcli`), [12](#)

M

`main()` (in module `terraformlintingcli.terraformlintingcli`), [12](#)

R

`ReadableDirectory` (class in `terraformlintingcli.terraformlintingcli`), [11](#)

`ReadableFile` (class in `terraformlintingcli.terraformlintingcli`), [11](#)

S

`setup_logging()` (in module `terraformlintingcli.terraformlintingcli`), [12](#)

T

`terraformlintingcli` (module), [12](#)

`terraformlintingcli.terraformlintingcli` (module), [11](#)

`terraformlintingcli.terraformlintingcliexceptions` (module), [12](#)