pyArchOps/pyarchops Documentation

Release 0.0.1

Azul

Dec 27, 2018

Contents:

CHAPTER 1

pyArchOps

pyarchops

- Free software: MIT license
- Documentation: https://pyarchops.readthedocs.io.

1.1 Features

- pyarchops

1.2 Instalation

```
$ pip install pyarchops
```

1.3 Usage

```
import os
import pyarchops

api = Api(
    '127.0.0.1:22',
    connection='smart',
    remote_user='ubuntu',
    private_key_file=os.getenv('HOME') + '/.ssh/id_rsa',
    become=True,
    become_user='root',
    sudo=True,
```

(continues on next page)

(continued from previous page)

```
    ssh_extra_args='-o StrictHostKeyChecking=no'
)
result = pyarchops.os_updates.apply(api)
print(result)
```

1.4 Development

Install requirements:

```
$ sudo pacman -S tmux python-virtualenv python-pip libjpeg-turbo gcc make vim git tk␣
↪tcl
```

Git clone this repository

```
$ git clone https://github.com/pyarchops/pyarchops.git pyarchops.pyarchops
$ cd pyarchops.pyarchops
```

2. See the *Makefile*, to get started simply execute:

```
$ make up
```

1.5 Credits

- TODO

CHAPTER 2

Installation

2.1 Stable release

The pyArchOps software should be installed through the main repository, .. _pyArchOps : https://github.com/pyarchops/pyarchops.git

```
$ pip install pyarchops
```

2.2 Latest pyarchops release

To install pyArchOps/pyarchops, run this command in your terminal:

```
$ pip install pyarchops
```

This is the preferred method to install pyArchOps/pyarchops, 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.

2.3 From sources

The sources for pyArchOps/pyarchops can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/pyarchops/pyarchops
```

Or download the tarball:

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

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

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

CHAPTER 3

Usage

To use pyArchOps/pyarchops in a project:

```
import pyarchops
```

CHAPTER 4

pyarchops

4.1 pyarchops package

4.1.1 Module contents

__init__.py for pyarchops

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:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/pyArchOps/pyarchops/pyarchops/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.

5.1.2 Fix Bugs

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

5.1.3 Implement Features

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

5.1.4 Write Documentation

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

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/pyArchOps/pyarchops/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 :)

5.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/pyarchops/pyarchops.git
```

3. set a development environment:

   $ sudo pacman -Sy tmux libffi pkgconf base-devel make $ cd pyarchops/pyarchops/ $ make up

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 pylint and the tests.
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.

5.3 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 3.7. Check https://gitlab.com/pyarchops/pyarchops/pipelines and make sure that the tests pass for all supported Python versions.

5.4 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

CHAPTER 6

Credits

6.1 Development Lead

- Azul <pyarchops@azulinho.com>

6.2 Contributors

None yet. Why not be the first?

CHAPTER 7

History

7.1 0.0.1 (2018-12-11)

- First release on PyPI.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Index

P