Ansible customer Documentation

Release 0.1.0-alpha+001

Alexandre Chaussier

Jul 04, 2020

Contents

Contents:

CHAPTER 1

Ansible customer

Python module to manage an Ansible project, linked to Infopen Ansible customer cookiecutter template.

It expose some cli to manage an Ansible project, with settings for each environment.

We use Invoke tasks, linked to their cli to run commands inside contexts.

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

1.1 Testing

You must have **Docker >= 1.13.0** installed to run the tests. We use it to check Ansible commands

To run tests locally, just run needed environments using tox:

```
$ TOXENV=py27-ansible23 tox
```

You can enable Paramiko debug if you have an error on Docker fixture create (ex: `Exception: Timeout reached while waiting on service!`)

```
$ PARAMIKO_DEBUG=1 TOXENV=py27-ansible23 tox
```

1.2 Features

1.2.1 Ansible cli wrapper

Entry point: `aci-ansible`

This is the wrapper over the `ansible` command.

Ping

```
aci-ansible ping my_hosts [--limit=foo]
```

Simple task to run ping module over an host list.

Setup

```
aci-ansible setup my_hosts [--limit=foo]
```

Simple task to run setup module over an host list.

1.2.2 Ansible-playbook cli wrapper

Entry point: `aci-ansible-playbook`

This is the wrapper over the `ansible-playbook` command.

List-tags

```
aci-ansible-playbook list_tags my_playbook
```

Simple task to list tags managed by a playbook.

List-tasks

```
aci-ansible-playbook list_tasks my_playbook
```

Simple task to list tasks managed by a playbook.

Run

```
aci-ansible-playbook run my_playbook [--limit=foo]
```

Simple task to run a playbook.

1.2.3 Ansible-galaxy cli wrapper

Entry point: `aci-ansible-galaxy`

This is the wrapper over the `ansible-galaxy` command.

Install

```
aci-ansible-galaxy install requirement_file [-f]
```

Simple task to install roles managed by the requirement file.

List-roles

```
aci-ansible-galaxy list_roles [--role-name=my_role]
```

Simple task to list role(s) installed.

Remove

```
aci-ansible-galaxy remove my_role1[,my_role2,...]
```

Simple task to remove role(s).

1.2.4 Molecule cli wrapper

Entry point: `aci-molecule`

This is the wrapper over the `molecule` command.

Create

```
aci-molecule create scenario_name [--driver=docker]
```

Simple task to start test instances.

Converge

```
aci-molecule converge scenario_name
```

Simple task to configure test instances.

Dependency

```
aci-molecule dependency scenario_name
```

Simple task to download dependencies required by scenario.

Destroy

```
aci-molecule destroy scenario_name [--driver=docker]
```

Simple task to destroy test instances.

List

```
aci-molecule list scenario_name [--output=simple]
```

Simple task to list test instances status.

Login

```
aci-molecule login scenario_name host
```

Simple task to login into test instance.

Test

```
aci-molecule test scenario_name [--driver=docker]
```

Simple task to run tests against instances and destroy them.

Verify

```
aci-molecule verify scenario_name
```

Simple task to run automated tests against instances.

1.3 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation

2.1 Stable release

To install Ansible customer, run this command in your terminal:

```
$ pip install ansible_customer
```

This is the preferred method to install Ansible customer, 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.

2.2 From sources

The sources for Ansible customer can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/infOpen/ansible_customer
```

Or download the tarball:

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

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

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

CHAPTER 3

Usage

To use Ansible customer in a project:

```
import ansible_customer
```

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/infOpen/ansible_customer/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" and "help wanted" is open to whoever wants to implement it.

4.1.3 Implement Features

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

4.1.4 Write Documentation

Ansible customer could always use more documentation, whether as part of the official Ansible customer 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/infOpen/ansible_customer/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 *ansible_customer* for local development.

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

```
$ git clone git@github.com:your_name_here/ansible_customer.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 ansible_customer
$ cd ansible_customer/
$ 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 ansible_customer tests
$ python setup.py test or 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, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/infOpen/ansible_customer/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_ansible_customer
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search