Creating a Tempest plugin for OpenStack

OCT 2015 - #OPENSTACKGDL

Marcela Bonell (@mbonell)

Tempest

Tempest is the OpenStack official test suite. Its purpose is to run tests for OpenStack API validation in an OpenStack cluster, in order to know how healthy our cloud is.

- Used as a gate for validate commits into the OpenStack core projects.

Developer documentation: http://docs.openstack.org/developer/tempest/

Source code repository: https://github.com/openstack/tempest

Tempest Test Plugin Interface

Thanks to a new feature proposed in the community (Kilo release), we can create independent plugins for Tempest's code to run custom test cases over OpenStack:

- Any plugin can run a set of external tests as part of a Tempest run.
- Forget about the complexity of integrating custom test cases into Tempest's code.
- Moving QA in OpenStack to a self-service model.
  - Each new project in OpenStack is responsible for writing and maintaining their deployment scripts and tests, as plugins.
- More independence and flexibility!

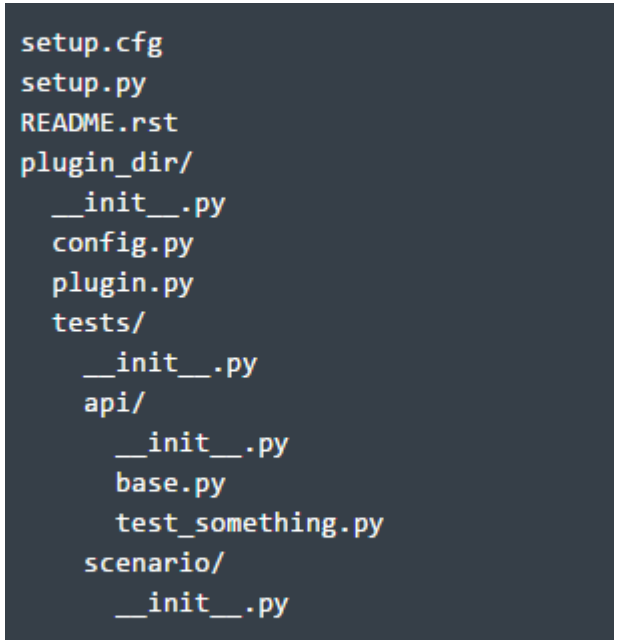
What is a Tempest plugin?

A Tempest plugin is basically a Python package that contains your custom test cases.

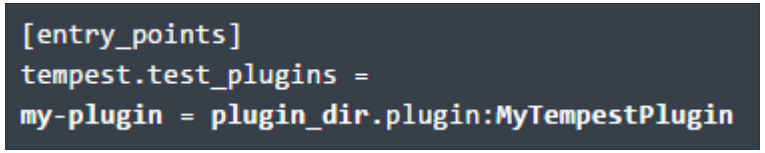
- It needs to be installed on the local or virtual environment (venv) where Tempest is located.
- All of your installed plugins will be discovered by Tempest when it runs.

Python Packaging User Guide: https://python-packaging-user-guide.readthedocs.org

Plugin structure



```
setup.cfg
setup.py
README.rst
plugin_dir/
  __init__.py
  config.py
  plugin.py
  tests/
    __init__.py
    api/
      __init__.py
      base.py
      test_something.py
    scenario/
      __init__.py
```



```
[entry_points]
tempest.test_plugins =
my-plugin = plugin_dir.plugin:MyTempestPlugin
```

Basic plugin example: https://github.com/mbonell/hello-world-tempest-plugin

Plugin installation

Install from the plugin root directory through pip in local environments:

```
$ pip install -e ~/hello-world-tempest-plugin/
```

Installation in virtual environments:

If you run Tempest inside a venv you have to ensure that the Python package containing the plugin is installed in the venv.

```
$ . ~/.rally/tempest/for-deployment-x-x-x-x-x/.venv/bin/activate
$ ~/.rally/tempest/for-deployment-x-x-x-x-x/.venv/bin/pip install -e ~/hello-world-tempest-plugin/
```

Validate that the plugin was installed correctly:

```
$ pip list
$ ~/.rally/tempest/for-deployment-x-x-x-x-x/.venv/bin/pip list
```

How to run the tests?

Testr (Tempest)

Testr is an Openstack test runner. https://wiki.openstack.org/wiki/Testr

- Validate that Tempest discovered your tests in the plugin:

```
$ testr list-tests | grep hello_world_tempest_plugin
```

- Run the test cases by **name** or running the set names used as **decorator**:

```
$ testr run hello_world_tempest_plugin.tests.api.test_hello_world.TestHelloWorld.test_hello_world
```

```
$ testr run --subunit smoke | subunit-2to1 | ./tools/colorizer.py
```

Rally

Rally is a benchmarking tool that automates and unifies multi-node OpenStack deployment, cloud verification, benchmarking & profiling.

Running a plugin through Rally:

```
$ rally verify start --set smoke
```

smoke is the set name used as decorator in the test cases

[IN PROGRESS] My blueprint in Rally about plugin installation: https://blueprints.launchpad.net/rally/+spec/install-tempest-plugins

OpenStack Summit

- Talk in the summit, **External Plugin Interfaces for OpenStack QA Projects**: http://sched.co/49tT

- My article in **opensource.com**: https://opensource.com/business/15/10/creating-a-tempest-plugin-for-openstack