Using Automation Framework – Tempest

Contents

[Introduction 1](#_Toc466984509)

[Overview of our tempest framework 1](#_Toc466984510)

[Setting up tempest framework in your env 1](#_Toc466984511)

[Pre-requisites 1](#_Toc466984512)

[Next Steps 1](#_Toc466984513)

## Introduction

Tempest is an open source automation framework specially designed to test OpenStack and OpenStack based applications. It is written in python. Capabilities if tempest includes launching instances, creating volumes, attaching/detaching volumes to/from instances, creating images, users, projects etc. Framework has all the OpenStack clients which includes keystone, nova, glance, neutron, cinder, swift etc.

## Overview of our tempest framework

We have created our own copy of tempest from opensource tempest repo and plugged-in our API/CLI client (workloadmgr client) to it. Now this tempest can perform tvault operations too using workloadmgr client. You have two ways to perform tvault operations using tempest, 1 . REST API 2. CLI. To make task easir we have written a wrapper classes for both interfaces (API & CLI) where we have written wrapper functions which internally calls APIs/CLIs of tvault. So you just need to import those functions and use it in your test scripts.

## Setting up tempest framework in your env

### Pre-requisites

You need to deploy tvault and do all needed configurations to make it ready to use.

### Next Steps

1. *Create a centos7 or Ubuntu14 VM with enough resources (4 GB RAM, 100 GB disk, 2VCPUs)*
2. *Install following packages*

***yum -y install python-devel libffi-devel gcc python-subunit openssl-devel  
 pip install virtualenv mysql-connector***

1. *Fork ‘tempest’ repo to your space on github*
2. Clone your tempest framework from your forked repos  
   ***git clone*** [***https://github.com/<git\_user\_name>/tempest.git***](https://github.com/%3cgit_user_name%3e/tempest.git)
3. *Get python-workloadmgrclient from landing page of tvault as follows  
   curl –O* [*http://192.168.1.105:8081/packages/python-workloadmgrclient-2.2.11.tar.gz*](http://192.168.1.105:8081/packages/python-workloadmgrclient-2.2.11.tar.gz) *tar –xvzf python-workloadmgrclient-2.2.11.tar.gz  
   mkdir /opt/stack  
   mv python-workloadmgrclient-2.2.11 /opt/stack/python-workloadmgrclient  
   cd python-workloadmgrclient  
   python setup.py install*
4. *Change the dir to tempest  
   Tune following config files  
   1. etc/tempest.conf : All the openstack details  
   2. etc/accounts.yaml : Static tenant credentials to use for creating vms and volumes  
   3. tempest/tvaultconf.py : Used by tvault test cases for better management*
5. *Change dir to tempest, from tempest dir run following command  
   ./run\_tempest.sh tempest.api.compute.images.test\_image\_metadata  
   It will ask for creating virtual environment, say ‘yes’.  
   This is basic openstack test and will create a virtual env in current directory under .venv folder, which will be used by tempest.*
6. *Now, if you see any error in above step, you need to resolve it. Generally errors would be of package/module not available. You need to install those packages using pip or native system package manager to get over the issues.*
7. *Once you pass this test, you are all set, you tempest is running well now.*
8. *To verify basic tvault test cases are running well, you need to execute following test cases  
    ./run\_tempest.sh tempest.api.workloadmgr.integration.test\_create\_workload  
   This is simple test cases which creates a workload contains a instance and a volume*
9. *If you see any error, you need to revisit setp-6, tunning configuration files. Make sure all the details you provided are correct.*
10. *You are all set.*