Tempest Stable Interfaces for OpenStack Integration Testing

Andrea Frittoli andrea.frittoli@hpe.com andreaf on Freenode

Apr 25, 2016

https://github.com/andreafrittoli/tempest stable interfaces

Neutron or Newton

/'njuxt(r)pn/

What is OpenStack QA?

► Official Mission Statement:

Develop, maintain, and initiate tools and plans to ensure the upstream stability and quality of OpenStack, and its release readiness at any point during the release cycle.

Current QA Projects

- ► eslint-config-openstack
- ► bashate
- ► hacking
- ► tempest
- ► tempest-lib (deprecated)
- ► grenade
- ► devstack
- ► devstack-plugin-ceph
- ► devstack-vagrant
- ► stackviz
- ▶ openstack-health dashboard
- ▶ os-testr
- devstack-plugin-cookiecutter
- ► tempest-plugin-cookiecutter
- ► os-performance-tools

What is Tempest



Tempest in the Big Tent

- ► tempest lib or tempest.lib
- ► tempest plugins

Current Projects QA directly supports in-tree

- Keystone
- ► Nova
- ► Glance
- ► Cinder
- ► Neutron
- ► Swift
- ► Tests executed in integrated-gate jobs

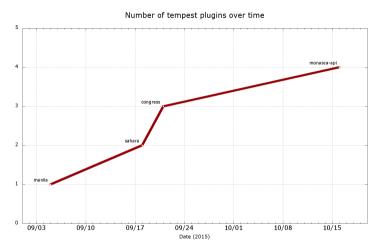
Where do other Tempest based tests live in OpenStack?

Integration Tests outside of Tempest tree

- ► Tempest Plugins (29)
- ► CLI tests for clients (16)
- ► Other functional/integration tests

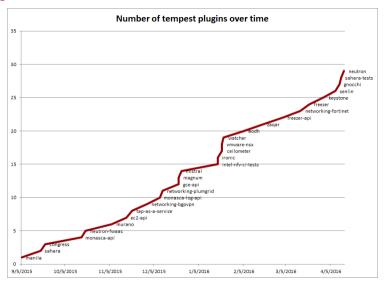
- ► Not executed against tempest
- ► Should only use tempest stable interfaces

Tempest Plugins at the end of Liberty

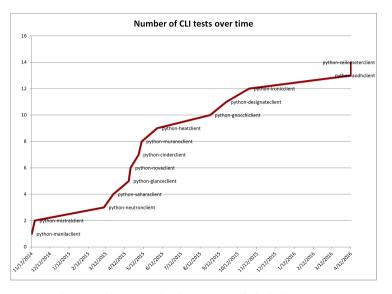


Source: git blame -L '/^tempest/,+1' setup.cfg | awk '{print \$1}' | xargs git log -1 -format=%cd -date=short

Tempest Plugins over time



CLI Tests over time



Source: git blame -L '/^(from|import) tempest[_ .]+lib.cli/,+1' *.py

Functional/Integration Tests

- ► WIP Tempest Plugins
 - kingbird
 - vitrage
- ► Tempest Tests (potential plugins)
 - ▶ blazar
 - ► designate
 - ► networking-l2gw
 - ► networking-vsphere
 - ► neutron-lbaas
 - neutron-vpnaas

Functional/Integration Tests

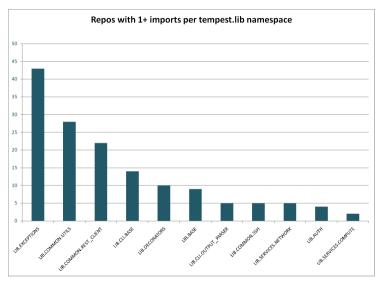
- ► Rest Client and other interfaces
 - cerberus
 - ▶ cue
 - ► solum
- ► Test base class or utils
 - astara
 - ▶ barbican
 - python-keystoneclient
 - python-openstackclient
 - ► tacker

What are Tempest (Stable) Interfaces, how are they used?

Tempest Stable APIs

- ▶ Rest Client and Service clients: identity, network, compute
- ► Authentication providers
- ► API Microversion utils
- ► SSH client
- ► Test data utils
- ► CLI test framework
- ► Decorators, exceptions
- ► Base test class
- Commands: check_uuid, skip tracker

Tempest Stable APIs

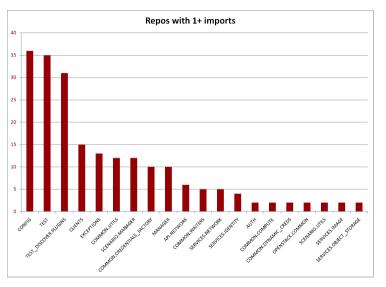


 $Source: \ codes earch.op \ en \ stack.org$

Tempest Internal APIs candidate to become Stable

- ► Service clients: volume, image, object-storage
- ► Credential providers
- ► Client manager
- ► Plugin
- ► Attr decorator

Tempest Internal APIs



 $Source: \ codes earch.op \ en \ stack.org$

How to use these interfaces to write a Tempest plugin?

Tempest Plugins Interface

- ▶ Integrates external tests into a tempest run
- Unifies configuration between plugin(s) and in-tree
- ► Integrates custom service clients (planned)
- ► Based on stevedore extension manager
- Automatically discovered when installed

Tempest Plugins Interface

```
# Import config for 'register_opt_group'
from tempest import config
# Import the plugin base class
from tempest.test_discover import plugins

from manila_tempest_tests import config as config_share

class ManilaTempestPlugin(plugins.TempestPlugin):

    def register_opts(self, conf):
        config_register_opt_group(
            config_register_opt_group(
            config_share.ServiceAvailableGroup)
        config_register_opt_group(config_share.share_group,config_share.ShareGroup)
```

Full_code_at:_http://git.openstack.org/cgit/openstack/manila/tree/manila_tempest_tests/plugin.py

Rest Client and Service Clients

- ► ReST API Calls with different HTTP methods
- Decorate requests using supplied auth provider
- ► Validate HTTP return codes
- ► Handle HTTP non-2xx return codes as custom exceptions
- ► Methods for API calls
- ► Pass any parameter to API calls

Rest Client and Service Clients

```
from tempest.lib.common import rest client
# Use tempest base client manager
from tempest import manager
from tempest services image v1 ison images client import | mages Client
class Telemetry Client (rest client . Rest Client):
    def create sample (self, meter name, sample list):
         uri = "%s/meters/%s" % (self uri prefix meter name)
        body = self serialize (sample list)
        resp body = self post (uri body)
        self.expected success(200, resp. status)
        body = self deserialize (body)
        return rest client ResponseBody (resp. body)
class Manager (manager, Manager):
    def set image client(self):
         sel\overline{f}, image client = lmagesClient(self, auth_provider).
                                            ** self image params)
    def set_telemetry_client(self):
         self. telemetry client = Telemetry Client (self. auth provider,
                                                   ** self. telemetry _ params)
```

Full code at: http://git.openstack.org/cgit/openstack/ceilometer/tree/ceilometer/tests/tempest/service/client.py

Authentication Layer

- ► Credentials object
- ► Select endpoints from the catalogue
- ► Decorate requests with identity v2 and v3 auth data
- ► Inject alternate auth data

Authentication Layer

```
from tempest, lib import auth
def get auth provider class (credentials):
    if isinstance (credentials, auth. Keystone V3 Credentials):
        return auth Keystone V 3 Auth Provider CONF, identity uri v 3
    else:
        return auth. Keystone V 2 Auth Provider, CONF. identity.uri
def get auth provider(credentials, pre auth=False):
    default params = {
         'disable ssl certificate validation':
            CONF. identity disable ssl certificate validation,
         'ca certs': CONF.identity.ca certificates file,
         'trace requests': CONF. debug trace requests
    auth provider class, auth url = get auth provider class(
         credentials)
    auth provider = auth provider class(credentials, auth url,
                                            ** default params)
    if pre auth:
    _auth_provider.set_auth()
return _auth provider
```

Full code at: http://git.openstack.org/cgit/openstack/tempest/tree/tempest/manager.py

Client Managers

- ► One object to access all service clients
- ► Bound to a set of credentials
- ► Hide the complexity of the auth layer
- ► Not yet in the lib namespace (WIP)
- ► Stable interface to register service clients from plugins
- Lazy loading of clients

Client Managers

```
from tempest import manager
from neutron tests tempest services network ison network client import
      Network Client ISON
class Manager (manager, Manager):
    def __init__(self, credentials=None, service=None):
    super(Manager, self). init (credentials=credentials)
         self.network_client = NetworkClientJSON(
self.auth provider,
              CONF network catalog type,
              CONF network region or CONF identity region,
              endpoint type=CONF.network.endpoint type,
              build interval=CONF.network.build interval,
              build timeout=CONF.network.build timeout,
              ** self. default params)
```

Full code at: http://git.openstack.org/cgit/openstack/neutron/tree/neutron/tests/tempest/api/clients.py

Credential Providers

- ► Supply test cases with credentials
- ► Manage multiple test account for parallel test execution
- ► Manage account specific network resources
- ► Not yet in the lib namespace (WIP)
- ► Dynamic Credential Provider
- Preprovisioned Credential Provider

Testing microversions

- ▶ Define acceptable microversion range for test class
- ▶ Match configured microversion range with tests
- ► Select microversion to be sent via API

Testing microversions

```
from tempest. lib.common import api version utils
import tempest, test
class BaseV2ComputeTest(api version utils.BaseMicroversionTest,
                         tempest test BaseTestCase):
    Oclassmethod
    def skip checks(cls):
        super(BaseV2ComputeTest, cls) skip checks()
        if not CONF service available nova:
            raise cls.skip Exception ("Nova is not available")
        cfg min version = CONF compute min microversion
        cfg max version = CONF compute max microversion
        api version utils check skip with microversion (cls. min microversion,
                                                         cls max microversion
                                                         cfg min version,
                                                         cfg max version)
    Oclassmethod
    def resource setup(cls):
        super (BaseV2ComputeTest, cls) resource setup()
        cls request microversion = (
            api version utils select request microversion (
                 cls min microversion,
                CONF compute min microversion))
```

Full code at: http://git.openstack.org/cgit/openstack/tempest/tree/tempest/api/compute/base.py

Miscellaneous Utils

- ► Generate random test data
- ► SSH client
- ► Skip decorators
- ► Test Attributes (not yet stable)

Which interfaces do you need to implement CLI tests?

CLI Tests Interfaces

- ► An execute command to drive clients via CLI
- ► A *CLIClient* class that wraps *execute* for clients
- ► An output_parser module with helpers to parse clients output
- ► A ClientTestBase class with clients and output parsers

CLI Tests Interfaces

```
from tempest lib.cli import base
class MistralCL|Auth(base, Client Test Base):
    def get clients(self):
        return base. CLIClient (
             username=creds['username'],
             password=creds['password'],
             tenant_name=creds['tenant name'],
             uri=creds['auth url'],
             cli dir=CLI DIR)
    def mistral(self, action, flags='', params='', fail_ok=False):
    """ Executes Mistral command."""
         mistral url op = "--os-mistrmistralal-url %s" % self. mistral url
         if 'WITHOUT AUTH' in os environ:
             return base, execute (
                  'mistral %s' % mistral url op, action, flags, params,
                 fail ok, merge stderr=False, cli dir='')
         elses
             return self clients cmd with auth (
                  'mistral %s' % mistral url op, action, flags, params,
                 fail ok)
```

Full code at: http://git.openstack.org/cgit/openstack/python-mistralclient/tree/mistralclient/tests/functional/cli/base.py

Where to get more information

- ► Tempest Stable Interfaces Docs: http://docs.openstack.org/developer/tempest/library.html
- ► Tempest Plugin Docs: http://docs.openstack.org/developer/tempest/plugin.html
- ► Tempest External Plugins Presentation: https://github.com/mtreinish/external plugins
- ► Tempest Stable Interfaces (this presentation): https://github.com/andreafrittoli/tempest_stable_interfaces
- ► #openstack-qa on Freenode

Questions?