OpenStack QA

Walkthrough of Tools, Code and Processes

Most Important Things to Remember

- Participate
  - The more you participate, the more people you meet and the more knowledge you gain
  - The more knowledge you gain, the more you can share that knowledge with others
- Ask questions
  - Don't be afraid to ask questions
  - There's no such thing as a stupid or silly question
- Be public
  - Prefer public questions and discussion over private email threads
  - Cast a wide net, catch more fish

Things all contributors should have

- Launchpad account
  - Upload your SSH keys to Launchpad
  - Gerrit imports your SSH keys from Launchpad
  - Subscribe to main openstack mailing list
- IRC
  - freenode.net #openstack and #openstack-dev
  - Typically 340+ on #openstack, 140+ on #openstack-dev
  - Best place to find help
  - #openstack-meeting for weekly meetings
    - Wednesdays at 16:00 UTC
- devstack installed locally

Running with devstack

- Always run your test code against a *real* system
- devstack makes things easy for you
- Running stack.sh "resets" a development OpenStack environment for you
- Point tests like Tempest at your local devstack environment
- Great for one-off tests or monkey-testing

Running devstack (`stack.sh`)

```
$> ./stack.sh
<snip lots of output!>
horizon is now available at http://127.0.0.1/
keystone is serving at http://127.0.0.1:5000/v2.0/
examples on using novaclient command line is in exercise.sh
the default users are: admin and demo
the password: 4f9a953e98ee57d922d9
This is your host ip: 127.0.0.1
stack.sh completed in 79 seconds.
```

Tempest

- Tempest is a project that contains functional integration tests intended to be run against actual OpenStack deployments
- Contains a functional testing framework that uses the **unittest2** and **nose** Python libraries as a base
- Test cases execute a series of API calls against OpenStack service endpoints and validate the response from the endpoint
- Uses a simple config file that describes test environment

Easy way to generate a test config

```
jpipes@librebox:~/repos/tempest$ ./tempest/tools/conf_from_devstack -D ../devstack/ -o etc/tempest.
conf
Output file already exists. Overwrite? [y/N]Y
jpipes@librebox:~/repos/tempest$ cat etc/tempest.conf
[nova]
host=127.0.0.1
port=5000
apiVer=v2.0
path=tokens
user=admin
api_key=4f9a953e98ee57d922d9
tenant_name=admin
ssh_timeout=300
build_interval=10
build_timeout=600

[environment]
image_ref=3712ca26-f926-4725-9132-08ec1f6e452e
image_ref_alt=4
flavor_ref=1
flavor_ref_alt=2
create_image_enabled=true
resize_available=true
```

Bolded values were gathered from the devstack configuration

Running Tempest

Run Tempest with `nosetests`

```
jpipes@librebox:~/repos/tempest$ nosetests -v tempest
List of all extensions ... ok
<snip>

----------------------------------------------------------------------
Ran 61 tests in 2283.166s

FAILED (SKIP=2, errors=5)

jpipes@librebox:~/repos/tempest$ nosetests -v tempest.tests.test_flavors
The expected flavor details should be returned ... ok
flavor details are not returned for non existant flavors ... ok
List of all flavors should contain the expected flavor ... ok
Detailed list of all flavors should contain the expected flavor ... ok

----------------------------------------------------------------------
Ran 4 tests in 0.722s

OK
```

Running a single test case...

```
jpipes@librebox:~/repos/tempest$ nosetests -v  tempest.tests.test_flavors:
FlavorsTest.test_list_flavors
List of all flavors should contain the expected flavor ... ok

----------------------------------------------------------------------
Ran 1 test in 0.502s

OK
```

Top tip: Separate test module from test class with colon, not dot!

What errors look like...

```
======================================================================
ERROR: The server should be rebuilt using the provided image and data
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/home/jpipes/repos/tempest/tempest/tests/test_server_actions.py", line 79, in test_rebuild_server
        self.client.wait_for_server_status(rebuilt_server['id'], 'ACTIVE')
  File "/home/jpipes/repos/tempest/tempest/services/nova/json/servers_client.py", line 150, in wait_for_server_status
        raise exceptions.TimeoutException(message)
TimeoutException: Request timed out
Details: Server 8bd71a0c-64d7-4ffc-aeb0-227bf4bbb80c failed to reach ACTIVE status within the required time (600.0 s).
```

Machine-readable output with xUnit

```
jpipes@librebox:~/repos/tempest$ nosetests -v --with-xunit tempest.tests.test_server_actions
The server's password should be set to the provided password ... ok
The server should be power cycled ... ok
The server should be signaled to reboot gracefully ... ok
The server should be rebuilt using the provided image and data ... ERROR
The server's RAM and disk space should be modified to that of ... ERROR
The server's RAM and disk space should return to its original ... ERROR
<snip>

======================================================================
ERROR: The server's RAM and disk space should return to its original
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/home/jpipes/repos/tempest/tempest/tests/test_server_actions.py", line 112, in test_resize_server_revert
      self.client.wait_for_server_status(self.server_id, 'VERIFY_RESIZE')
  File "/home/jpipes/repos/tempest/tempest/services/nova/json/servers_client.py", line 150, in
wait_for_server_status
      raise exceptions.TimeoutException(message)
TimeoutException: u'Server d7f1d3e3-1d3a-4c22-b3ef-1fe6660fe0b5 failed to reach ACTIVE status within the
      required time (600.0 s).'

----------------------------------------------------------------------
XML: nosetests.xml
----------------------------------------------------------------------
Ran 6 tests in 1918.131s

FAILED (errors=3)
jpipes@librebox:~/repos/tempest$ cat nosetests.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<testsuite name="nosetests" tests="6" errors="3" failures="0" skip="0">
 <testcase classname="tempest.tests.test_server_actions.ServerActionsTest" name="test_change_server_password" time="22" />
 <testcase classname="tempest.tests.test_server_actions.ServerActionsTest" name="test_reboot_server_hard" time="22" />
 <testcase classname="tempest.tests.test_server_actions.ServerActionsTest" name="test_reboot_server_soft" time="22" />
 <testcase classname="tempest.tests.test_server_actions.ServerActionsTest" name="test_rebuild_server" time="617">
  <error type="tempest.exceptions.TimeoutException" message="u'Server ac4a2f89-f905-42e5-ba14-801a3146cf9c failed to reach ACTIVE status
within the                    required time (600.0 s).'">
   <![CDATA[Traceback (most recent call last):
  File "/usr/lib/pymodules/python2.7/unittest2/case.py", line 340, in run
        testMethod()
  File "/home/jpipes/repos/tempest/tempest/tests/test_server_actions.py", line 79, in test_rebuild_server
        self.client.wait_for_server_status(rebuilt_server['id'], 'ACTIVE')
  File "/home/jpipes/repos/tempest/tempest/services/nova/json/servers_client.py", line 150, in wait_for_server_status
        raise exceptions.TimeoutException(message)
TimeoutException: u'Server ac4a2f89-f905-42e5-ba14-801a3146cf9c failed to reach ACTIVE status within the                       required
time (600.0 s).'
]]>
  </error>
 </testcase>
...
</testsuite>
```

Tempest Code Walkthrough

Tempest directory structure

```
$src_dir/
   etc/   <-- contains sample Tempest config
   tempest/
      common/ <-- common code like rest_client
      services/
         nova/ <-- client for compute
      tests/ <-- test cases
      tools/ <-- helpers scripts
```

Tempest test case class

- Contains a set of related tests
  - For instance, a test case may test operations that list servers in various ways
- Resources shared by methods of the test case should be created in the setUpClass() method and destroyed in the tearDownClass() method

```python
12 class ListImagesTest(unittest.TestCase):
13
14     @classmethod
15     def setUpClass(cls):
16         cls.os = openstack.Manager()
17         cls.client = cls.os.images_client
18         cls.servers_client = cls.os.servers_client
19         cls.config = cls.os.config
20         cls.image_ref = cls.config.env.image_ref
21         cls.flavor_ref = cls.config.env.flavor_ref
22
23         name = rand_name('server')
24         resp, cls.server1 = cls.servers_client.create_server(name,
25                                                              cls.image_ref,
26                                                              cls.flavor_ref)
27         cls.servers_client.wait_for_server_status(cls.server1['id'], 'ACTIVE')
28
29         name = rand_name('server')
30         resp, cls.server2 = cls.servers_client.create_server(name,
31                                                              cls.image_ref,
32                                                              cls.flavor_ref)
33         cls.servers_client.wait_for_server_status(cls.server2['id'], 'ACTIVE')
34
35         #Create images to be used in the filter tests
36         image1_name = rand_name('image')
37         resp, body = cls.client.create_image(cls.server1['id'], image1_name)
38         cls.image1_id = _parse_image_id(resp['location'])
39         cls.client.wait_for_image_resp_code(cls.image1_id, 200)
40         cls.client.wait_for_image_status(cls.image1_id, 'ACTIVE')
41         resp, cls.image1 = cls.client.get_image(cls.image1_id)
42
43         image2_name = rand_name('image')
44         resp, body = cls.client.create_image(cls.server1['id'], image2_name)
45         cls.image2_id = _parse_image_id(resp['location'])
46         cls.client.wait_for_image_resp_code(cls.image2_id, 200)
47         cls.client.wait_for_image_status(cls.image2_id, 'ACTIVE')
48         resp, cls.image2 = cls.client.get_image(cls.image2_id)
49
50         image3_name = rand_name('image')
51         resp, body = cls.client.create_image(cls.server2['id'], image3_name)
52         cls.image3_id = _parse_image_id(resp['location'])
53         cls.client.wait_for_image_resp_code(cls.image3_id, 200)
54         cls.client.wait_for_image_status(cls.image3_id, 'ACTIVE')
55         resp, cls.image3 = cls.client.get_image(cls.image3_id)
56
57     @attr(type='smoke')
58     def test_get_image(self):
59         """Returns the correct details for a single image"""
60         resp, image = self.client.get_image(self.image_ref)
61         self.assertEqual(self.image_ref, image['id'])
```

- setUpClass() method is used to create resources (in this case a pair of server instances and a few snapshot images) that are referenced by test methods in the test case
- What's wrong with this picture? hint: https://bugs.launchpad.net/tempest/+bug/899701
- Pattern you will see is that instead of making HTTP calls directly, you use the various client objects attached to the `tempest.openstack.Manager`

Tempest test case methods

- Test methods should validate a related set of actions
- Test methods may be decorated with the `nose.plugins.attrib.attr` decorator to indicate that a method contains a particular type of test
- If a test method creates any resources, it should always clean up after itself

Tempest test case methods (cont'd)

- Test methods should not modify any *shared* resources -- doing so may create ordering dependencies
- When using various assert methods, include a corresponding failure message that is descriptive and provides the tester with details they may need to diagnose an issue

Example: writing a good test case

Add a test case that does the following:

1. Create a new server from a base image
2. Change the name of the server
3. Validate the new name of the server appears when showing details about the server

```
def test_scenario(self):
    name = rand_name('server')
    resp, body = self.client.create_server(name, self.image_ref,
                                           self.flavor_ref)

    resp, body = self.client.update_server(server['id'], name='newname')
    self.assertEqual(200, resp.status)

    resp, server = self.client.get_server(server['id'])
    self.assertEqual('newname', server['name'])
```

```
@attr(type='smoke')
def test_update_server_name(self):
    """The server name should be changed to the the provided value"""
    name = rand_name('server')
    resp, body = self.client.create_server(name, self.image_ref,
                                           self.flavor_ref)

    resp, body = self.client.update_server(server['id'], name='newname')
    self.assertEqual(200, resp.status)

    resp, server = self.client.get_server(server['id'])
    self.assertEqual('newname', server['name'])
```

```
@attr(type='smoke')
def test_update_server_name(self):
    """The server name should be changed to the the provided value"""
    name = rand_name('server')
    resp, server = self.client.create_server(name, self.image_ref,
                                             self.flavor_ref)
    self.assertEqual(201, resp.status)

    # Update the server with a new name
    resp, body = self.client.update_server(server['id'], name='newname')
    self.assertEqual(200, resp.status)

    # Verify the name of the server has changed
    resp, server = self.client.get_server(server['id'])
    self.assertEqual('newname', server['name'])
```

```
@attr(type='smoke')
def test_update_server_name(self):
    """The server name should be changed to the the provided value"""
    name = rand_name('server')
    resp, server = self.client.create_server(name, self.image_ref,
                                             self.flavor_ref)
    self.assertEqual(201, resp.status)
    server_id = server['id']

    # Update the server with a new name
    resp, body = self.client.update_server(server_id, name='newname')
    resp_code = resp.status
    fail_msg = ("Failed to update server %(server_id)s. "
                "Got HTTP response code %(resp_code)d with "
                "body %(body)s") % locals()
    self.assertEqual(200, resp.status, fail_msg)

    # Verify the name of the server has changed
    resp, server = self.client.get_server(server_id)
    fail_msg = ("Failed to find updated server name. Expected 'newname' "
                "Got %s") % server['name']
    self.assertEqual('newname', server['name'], fail_msg)
```

```
@attr(type='smoke')
def test_update_server_name(self):
    """The server name should be changed to the the provided value"""
    name = rand_name('server')
    resp, server = self.client.create_server(name, self.image_ref,
                                             self.flavor_ref)
    self.assertEqual(201, resp.status)
    server_id = server['id']

    # Update the server with a new name
    resp, body = self.client.update_server(server_id, name='newname')
    resp_code = resp.status
    fail_msg = ("Failed to update server %(server_id)s. "
                "Got HTTP response code %(resp_code)d with "
                "body %(body)s") % locals()
    self.assertEqual(200, resp.status, fail_msg)

    # Verify the name of the server has changed
    resp, server = self.client.get_server(server_id)
    fail_msg = ("Failed to find updated server name. Expected 'newname' "
                "Got %s") % server['name']
    self.assertEqual('newname', server['name'], fail_msg)

    # Clean up after ourselves...
    self.client.delete_server(server['id'])
```

Submitting Code

Code Submission Guidelines

- Be consistent
  - Consistency in your code -- style, comments, documentation, etc -- shows you care
- Respond in a timely manner to reviews
- Ensure commit messages are proper
  - A Launchpad bug number or blueprint is referenced when appropriate
  - First line is short description of patch
  - More detailed description of patch follows
  - Do not put successive "fixup messages" in commit message

Basic contribution process

1. Assign yourself to an unassigned bug
2. Create local topic branch
3. Make code changes
4. Run Tempest against standing environments
5. If all tests pass, commit code changes
6. Write descriptive commit message
7. Propose for review
8. Address any review comments
9. Amend commit and re-propose

Common Scenarios

Use git-review to propose patch

Scenario:

You have assigned yourself to a bug, created or modified code that addresses the bug, run tests and now want to propose your changes for review.

Solution:

Commit local changes and then call `git review`

Basic code submission

1) `git commit -a`
2) Write a descriptive commit message
3) Save and close your editor
4) `git review`

Use `--amend` for small fixups

Scenario:

You have pushed a patch and gotten one or more reviews that call for some minor fixups. You make the fixes on your local branch and need to push the changes for review.

Solution:

Commit local changes, but *amend* the original commit.

Amending a commit after fixes

1) `git commit -a --amend`
2) Optionally edit the commit message
3) Save and close your editor
4) `git review`

Oops! You forgot to use `--amend`!

Scenario:

You made fixups based on review and then did a `git commit -a` and then `git review`, but you forgot to use `--amend`. This generated a new patchset to Gerrit instead of updating the original commit.

Solution:

First, take a deep breath and remember that every contributor has done this before. After that, Abandon the incorrect new changeset in Gerrit and then use `git reset` to undo your mistake.

Undoing your --amend mistake

1) Go to your newly-created Gerrit changeset and click the "Abandon" button to mark the changeset as obselete

2) `git reset HEAD^`

3) `git commit -a --amend`

4) Optionally edit the commit message to indicate any *major* changes you may have made during fixups

5) `git review`

```
jpipes@librebox:~/repos/tempest$  git commit -a  # Ooops! Forgot to --amend!
[bug912596 08439dc] Bad Jay!
 1 files changed, 1 insertions(+), 0 deletions(-)
jpipes@librebox:~/repos/tempest$  git reset HEAD^
Unstaged changes after reset:
M    tempest/openstack.py
jpipes@librebox:~/repos/tempest$  git commit -a --amend
[bug912596 9e320c0] Fixes LP Bug #912596 - image_ref_alt not found
 Author: Jay Pipes <jpipes@librebox.gateway.2wire.net>
 3 files changed, 89 insertions(+), 1 deletions(-)
 create mode 100644 tempest/tests/utils.py
```

Use git stash to save changes

Scenario:

You have a bunch of uncommitted code changes locally. You want to pull code that just made it into trunk.

Solution:

Use git stash to save your uncommitted code changes, pull changes from master, rebase your local branch against master and then reapply your stashed code changes.

```
git stash
git checkout master
git pull
git checkout <BRANCH>
git rebase master
git stash pop
```

jpipes@uberbox: ~/repos/tempest

```
jpipes@uberbox:~/repos/tempest$ git checkout master
error: Your local changes to the following files would be overwritten by checkout:
        tempest/openstack.py
Please, commit your changes or stash them before you can switch branches.
Aborting
jpipes@uberbox:~/repos/tempest$ git stash
Saved working directory and index state WIP on working: f008703 Added filter tests to list images tests, addresses lp bug 900088
HEAD is now at f008703 Added filter tests to list images tests, addresses lp bug 900088
jpipes@uberbox:~/repos/tempest$ git checkout master
Switched to branch 'master'
jpipes@uberbox:~/repos/tempest$ git pull
remote: Counting objects: 44, done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 26 (delta 17), reused 19 (delta 10)
Unpacking objects: 100% (26/26), done.
From git://github.com/openstack/tempest
   a37cf00..04b7081  master     -> origin/master
Updating a37cf00..04b7081
Fast-forward
 tempest/common/rest_client.py                   |   32 +++--
 tempest/exceptions.py                           |    8 +
 tempest/services/nova/json/extensions_client.py |    2 +-
 tempest/services/nova/json/flavors_client.py    |    2 +-
 tempest/services/nova/json/images_client.py     |    2 +-
 tempest/services/nova/json/limits_client.py     |    2 +-
 tempest/services/nova/json/servers_client.py    |    3 +-
 tempest/tests/test_images.py                    |   34 ++++-
 tempest/tests/test_list_servers.py              |   16 ++-
 tempest/tests/test_server_metadata.py           |   15 ++
 tempest/tests/utils.py                          |   73 +++++++++
 tempest/tools/conf_from_devstack                |  179 +++++++++++++++++++++++
 12 files changed, 345 insertions(+), 23 deletions(-)
 create mode 100644 tempest/tests/utils.py
 create mode 100755 tempest/tools/conf_from_devstack
jpipes@uberbox:~/repos/tempest$ git checkout working
Switched to branch 'working'
jpipes@uberbox:~/repos/tempest$ git rebase master
First, rewinding head to replay your work on top of it...
Fast-forwarded working to master.
jpipes@uberbox:~/repos/tempest$ git stash pop
Auto-merging tempest/openstack.py
# On branch working
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   tempest/openstack.py
#
no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (78d674786d2c46c2da1907f746808548c4b211bc)
```

Pull a colleague's code from Gerrit

Scenario:

You are reviewing a colleague's code and want to pull the code to your local machine for testing.

Solution:

Use git review -d <PATCH_NUM> to clone the remote branch locally and automatically put you into the branch.

openstack.org https://review.openstack.org/#change,2955

Here is the patch number...

Glance Reviews | Glance Bugs | QA Doc Reviews | Tempest Reviews | Tempest Bugs

All | My | Admin | Documentation

Changes | Drafts | Watched Changes | Starred Changes

Change lcd167c58: Added flavor filter tests * Resolve

Change-Id:	lcd167c58f7782b12e757cde7fa8ce4b4f505cb6c
Owner	Daryl Walleck
Project	openstack/tempest
Branch	master
Topic	bug/899979
Uploaded	Jan 11, 2012 12:37 AM
Updated	Jan 11, 2012 10:31 AM
Status	Review in Progress

```
Added flavor fil
* Resolved lp bu
* Modified flavc
* Added bug flag

Change-Id: Icdl6
```

Permalink

Reviewer	*Verified*	*Code Review*	*Approved*	
Jay Pipes		✓		Looks good to me (core revie

- Need Verified +1 (Verified)
- Need Approved +1 (Approved)

Name or Email

▶ **Dependencies**

```
jpipes@uberbox:~/repos/tempest$ git status
# On branch master
nothing to commit (working directory clean)
jpipes@uberbox:~/repos/tempest$ git review -d 2955
Downloading refs/changes/55/2955/1 from gerrit into review/daryl_walleck/bug/899979
Switched to branch 'review/daryl_walleck/bug/899979'
jpipes@uberbox:~/repos/tempest$
```

Troubleshooting

- If `stack.sh` fails to build and gives errors about domains failing to be removed, manually check the status of VMs that may have been left around after a test run using `virsh list --all`
- Manually `virsh destroy <INSTANCE_NAME>` and `virsh undefine <INSTANCE_NAME>` all instances and try re-running `stack.sh`