Preventing craziness
A deep dive into OpenStack testing automation

Thierry Carrez (@tcarrez)
Release manager, OpenStack
OpenStack is large & growing

- 95+ code repositories
- 1.9+ MLOC
OpenStack is complex

- 9 integrated projects
- Each project is a framework
OpenStack is painful

- Difficult to deploy
- Difficult to test
- Difficult to keep up with changes
OpenStack is open innovation

- Anyone can propose a change
- Lots of different companies
- No traditional management structure
- Technical meritocracy (no DFL)
- Global community
And yet...

- We release every 6 months
- Master branches are always usable
- We keep on growing
- We retained (most of) our sanity
1. Code reviews

- Everyone's code should be reviewed
- At worse it's a learning opportunity
- Elite committers vs. Peer reviewers
- Everyone can review
2. Automated testing

- Not tested $\rightarrow$ Not working
- Not automatically tested $\rightarrow$ Not tested
2. Automated testing

- Unit tests
- Integration tests
- Upgrade tests
- Syntax tests (PEP8)
- Hacking style tests (flake8/hacking)
3. Trunk gating

- Do not merge unless it passes tests
- « Automation for code review »
- Always-good trunk
All together now...

- Your git branch
- github
- git.o.o
- review.o.o
- master git branch
- master branch
- PEP8
- Hacking
- Unit tests
- Integration tests
git

- git is used everywhere
- git is fast & powerful
- git-review lets us submit to Gerrit
- Github only used as a repo mirror
Gerrit

- Code review
- Inline comments, vote tracking
Gerrit
Gerrit
Gerrit

- Code review
- Inline comments, vote tracking
- Issues:
  - Automating configuration (Jeepyb)
  - no private reviews
Zuul

- Triggered by gerrit event firehose
- Handles test pipelines (see later)
- Launches tests
- Uses Gearman to distribute tests
- Reports results back to Gerrit
Jenkins

- Runs tests on slaves
- Nodepool ensures slaves availability
- Devstack-gate scripts prepare slaves
- Issues:
  - Jjb
  - Avoid plugins
  - Not really useful at this point
All together now...

Repository

Gerrit

Zuul

Nodepool

Jenkins

Git clone

Git review

Gerrit stream-event

Zuul Gerrit reporter

Gearman jenkins plugin

Job results

Nodepool
Everyone loves numbers

- 900 patchsets proposed per day
- 4000 Gerrit comments/votes per day
- 7 Jenkins masters, 250-500 slaves
- 15K test jobs per day
- 200 patchsets merged per day
Scaling Jenkins & Gerrit

- Multiple master setup
- Log storage: logstash, Gearman
- Log search: ElasticSearch / Kibana
- Mirroring master git repo to cgit pool
Gating at scale

- Serial gating is easy but limited
Serial gating
Serial gating
Serial gating
Serial gating
Gating at scale

- Serial gating is easy but limited
- Parallel gating? Yay clouds
- But parallel gating may introduce fails
Parallel gating
Parallel gating
Gating at scale

- Serial gating is easy but limited
- Parallel gating? Yay clouds
- But parallel gating may introduce fails
- Zuul: Speculative execution of tests
Speculative gating

1

1 2

1 2 3

1 2 3 4
Speculative gating
Speculative gating
Speculative gating
Speculative gating
Speculative gating

Diagram showing speculative gating with numbered nodes and connections.
Speculative vs. serial gating
Visualizing Zuul

Check

Newly uploaded patches enter this pipeline to receive an initial +/-1 Verified vote from Jenkins.

Gate

Changes that have been approved by core developers are enqueued in order in this pipeline, and if they pass tests in Jenkins, will be merged.

Post

This pipeline runs jobs that operate after each change is merged.

Tag

This pipeline runs jobs in response to any tag event.

Pre-release

This pipeline runs jobs on projects in response to pre-release tag events.

Release

When a tag is tagged as a release, this pipeline runs jobs that publish archives and documentation.

Silent

This pipeline is used for silently testing new jobs.

Experimental

On-demand pipeline for requesting a run against a set of jobs that are not yet going. Leave review comment of "check experimental" to run jobs in this pipeline.
False negatives
## Tracking rechecks

### Jenkins

**Patch Set 1: Fails**

Build failed. For information on how to proceed, see [https://wiki.openstack.org/wiki/GerritJenkinsGit#Test_Failures](https://wiki.openstack.org/wiki/GerritJenkinsGit#Test_Failures)

<table>
<thead>
<tr>
<th>Task</th>
<th>Status</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>gate-swift-docs</td>
<td>SUCCESS</td>
<td>2m 04s</td>
</tr>
<tr>
<td>gate-swift-pep8</td>
<td>SUCCESS</td>
<td>1m 29s</td>
</tr>
<tr>
<td>gate-swift-python26</td>
<td>SUCCESS</td>
<td>3m 54s</td>
</tr>
<tr>
<td>gate-swift-python27</td>
<td>SUCCESS</td>
<td>3m 11s</td>
</tr>
<tr>
<td>gate-swift-dsvm-functional</td>
<td>SUCCESS</td>
<td>12m 43s</td>
</tr>
<tr>
<td>gate-tempest-dsvm-full</td>
<td>SUCCESS</td>
<td>54m 51s</td>
</tr>
<tr>
<td>gate-tempest-dsvm-neutron</td>
<td>SUCCESS</td>
<td>56m 12s</td>
</tr>
<tr>
<td>gate-tempest-dsvm-postgres-full</td>
<td>FAILURE</td>
<td>52m 05s</td>
</tr>
<tr>
<td>gate-tempest-dsvm-large-ops</td>
<td>SUCCESS</td>
<td>13m 50s</td>
</tr>
<tr>
<td>gate-tempest-dsvm-neutron-large-ops</td>
<td>SUCCESS</td>
<td>13m 44s</td>
</tr>
<tr>
<td>gate-grenade-dsvm</td>
<td>SUCCESS</td>
<td>50m 44s</td>
</tr>
</tbody>
</table>

### Samuel Merritt

**Patch Set 1:**

reverify bug 1254890

### Jenkins

Patch Set 1: No score Starting gate ...
Tracking rechecks

Bug 1264755: tempest.scenario.test_minimum_basic.TestMinimumBasicScenario failed
First seen: 2013-12-28 22:44:17 UTC
Last seen: 2014-01-17 11:13:29 UTC
Rechecks: 49
Affecting changes: 52867, 58292, 64622, 64655, 60991, 64893, 61597, 64874, 59733, 65772, 65395, 62216, 65637, 63147, 64361, 65472, 65966, 66166, 66187, 63658, 61786, 65841, 61930, 63723, 65943, 64271, 66541, 65358, 66854, 65515, 64217, 67003, 66796, 66983, 66514, 67295

Bug 1269192: tempest.api.compute.servers.test_server_actions.ServerActionsTestXML.test_create_backup fails gate sporadically
First seen: 2014-01-14 23:52:00 UTC
Last seen: 2014-01-17 11:10:08 UTC
Rechecks: 7
Affecting projects: openstack/swift, openstack/python-cinderclient, openstack/keystone, openstack/heat, openstack/nova, openstack/tempest
Affecting changes: 66489, 64003, 66719, 63041, 59778, 64590, 63723
Failure tracking by signature

**Bug 1260654**

**Bug 1260311**
## Elastic rechecks

### Jenkins
Patch Set 3:

*Build failed. For information on how to proceed, see* [https://wiki.openstack.org/wiki/GerritJenkinsGit#Test_Failures](https://wiki.openstack.org/wiki/GerritJenkinsGit#Test_Failures)

<table>
<thead>
<tr>
<th>Task</th>
<th>Status</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>gate-requirements-pep8</td>
<td>SUCCESS</td>
<td>in 29s</td>
</tr>
<tr>
<td>gate-requirements-python27</td>
<td>SUCCESS</td>
<td>in 1m 55s</td>
</tr>
<tr>
<td>gate-requirements-pypy</td>
<td>SUCCESS</td>
<td>in 2m 05s</td>
</tr>
<tr>
<td>check-requirements-integration-dsvm</td>
<td>SUCCESS</td>
<td>in 1h 01m 06s</td>
</tr>
<tr>
<td>check-tempest-dsvm-full</td>
<td>SUCCESS</td>
<td>in 58m 18s</td>
</tr>
<tr>
<td>check-tempest-dsvm-postgres-full</td>
<td>SUCCESS</td>
<td>in 43m 53s</td>
</tr>
<tr>
<td>check-tempest-dsvm-neutron</td>
<td>SUCCESS</td>
<td>in 39m 14s</td>
</tr>
<tr>
<td>gate-tempest-dsvm-large-ops</td>
<td>SUCCESS</td>
<td>in 13m 41s</td>
</tr>
<tr>
<td>gate-tempest-dsvm-neutron-large-ops</td>
<td>SUCCESS</td>
<td>in 20m 49s</td>
</tr>
<tr>
<td>check-grenade-dsvm</td>
<td>SUCCESS</td>
<td>in 24m 46s</td>
</tr>
<tr>
<td>check-swift-dsvm-functional</td>
<td>SUCCESS</td>
<td>in 12m 16s</td>
</tr>
</tbody>
</table>

### Elastic Recheck
Patch Set 3:

I noticed tempest failed, I think you hit bug(s):

[https://bugs.launchpad.net/bugs/1249065](https://bugs.launchpad.net/bugs/1249065)

We don't automatically recheck or reverify, so please consider doing that manually if someone hasn't already. For a code review which is not yet approved, you can recheck by leaving a code review comment with just the text:

```
recheck bug 1249065
```

For a code review which has been approved but failed to merge, you can reverify by leaving a comment like this:

```
reverify bug 1249065
```

**ChangBo Guo** Patch Set 3: recheck bug 1249065 Dec 25
Automation for the win

How long can you work on making a routine task more efficient before you're spending more time than you save? (Across five years)

<table>
<thead>
<tr>
<th>How much time you shave off</th>
<th>50/day</th>
<th>5/day</th>
<th>Daily</th>
<th>Weekly</th>
<th>Monthly</th>
<th>Yearly</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 second</td>
<td>1 day</td>
<td>2 hours</td>
<td>30 minutes</td>
<td>4 minutes</td>
<td>1 minute</td>
<td>5 seconds</td>
</tr>
<tr>
<td>5 seconds</td>
<td>5 days</td>
<td>12 hours</td>
<td>2 hours</td>
<td>21 minutes</td>
<td>5 minutes</td>
<td>25 seconds</td>
</tr>
<tr>
<td>30 seconds</td>
<td>4 weeks</td>
<td>3 days</td>
<td>12 hours</td>
<td>2 hours</td>
<td>30 minutes</td>
<td>2 minutes</td>
</tr>
<tr>
<td>1 minute</td>
<td>8 weeks</td>
<td>6 days</td>
<td>1 day</td>
<td>4 hours</td>
<td>1 hour</td>
<td>5 minutes</td>
</tr>
<tr>
<td>5 minutes</td>
<td>9 months</td>
<td>4 weeks</td>
<td>6 days</td>
<td>21 hours</td>
<td>5 hours</td>
<td>25 minutes</td>
</tr>
<tr>
<td>30 minutes</td>
<td>6 months</td>
<td>5 weeks</td>
<td>5 days</td>
<td>1 day</td>
<td>2 hours</td>
<td></td>
</tr>
<tr>
<td>1 hour</td>
<td>10 months</td>
<td>2 months</td>
<td>10 days</td>
<td>2 days</td>
<td>5 hours</td>
<td></td>
</tr>
<tr>
<td>6 hours</td>
<td>2 months</td>
<td>2 months</td>
<td>2 weeks</td>
<td>1 day</td>
<td>5 hours</td>
<td></td>
</tr>
<tr>
<td>1 day</td>
<td>8 weeks</td>
<td>5 days</td>
<td>8 weeks</td>
<td>5 days</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

http://xkcd.com/1205
Infrastructure as code

- Puppet-driven infrastructure
- All project infrastructure is open
- Yes, everything is in a code repo
- Apply our key principles to infra too
Join us!

- State-of-the-art dev infrastructure
- Openly developed
- Reused at Wikimedia and elsewhere
Questions?

thierry@openstack.org
http://fnords.wordpress.com
@tcarrez on Twitter
ttx on IRC