Open Source tools for distributed systems administration

DevOpsDayLA
Friday, January 22, 2016

Elizabeth K. Joseph
Hewlett Packard Enterprise
lyz@princessleia.com
@pleia2
Elizabeth K. Joseph

• OpenStack Infrastructure Team systems administrator, paid by Hewlett Packard Enterprise

• Long time contributor to various open source projects

• Co-author of *The Official Ubuntu Book, 8th ed*
1. Our team
2. The OpenStack CI system
3. Code Review
4. Automated testing
5. Other collaboration tools
6. Communication
How most open source projects do infrastructure

- Team (or company) manages it ...or they just use code hosting
- Requests are submitted via mailing list, bug report or ticketing system
- Request priority is determined by the core team

This may be similar to your organization.

Is there a better way?
OpenStack Infrastructure Team

• Our job is to make sure the OpenStack developers can do *their* job

• All of our system configurations are open source and tracked in git: https://git.openstack.org/cgit/openstack-infra

• Anyone in the world can propose patches for direct inclusion in our infrastructure, instructions at: http://docs.openstack.org/infra/manual/developers.html

• We all work remotely. Worldwide: US, Russia, Australia, Spain.
What we run

- Askbot
- Continuous Integration systems
- Cacti
- Elasticsearch, Logstash and Kibana
- IRC Bots
- Etherpad
- Git
- Paste
- Planet
- Puppetboard
- Mailing Lists
- Translations platform
- Various web services
- Wiki
OpenStack Continuous Integration (CI) System

• Lots of individual projects (800+)
• All projects must work together
• Changes can't break master branch
• Code should be syntactically clean
• Testing must be completed automated
Tools we're using for CI

- Launchpad (someday: openstackid)
- Git
- Gerrit
- Zuul*
- Gearman
- Jenkins (with jenkins-job-builder*, devstack-gate*)
- Nodepool*

* Started by the OpenStack Infrastructure team
Workflow

- Github mirror git.openstack.org mirror
- Git repository
- Gerrit code review
- Zuul
- Gearman Server
- pipy mirrors & other semi-privileged servers
- Jenkins (gearman-plugin)
- Jenkins01,02...
- Ubuntu, Fedora, Centos & devstack slaves
- Local changes submitted via git-review
Why do I care?

The Infrastructure team uses it too!
Automated tests for infrastructure

- flake8 (pep 8 and pyflakes)
- puppet parser validate
- puppet lint
- Puppet application tests
- XML checkers
- Alphabetized files
- IRC channel permissions
Peer review means...

- Multiple eyes on changes prior to merging
- Good infrastructure for developing new solutions
- No special process to go through for commit access
- Trains us to be collaborative by default
- Since anyone can contribute, anyone can devote resources to it
Gerrit in-line comments
Automated deployment

• Change gets approved, tested and merged
• ...Either puppet master gets updated and applies change
• ...Or vcsrepo module in puppet pulls in latest version of project
Can you really manage an infrastructure via git commits?

Cacti (http://cacti.openstack.org/) to keep an eye on server usage
Can you really manage an infrastructure via git commits? (2/3)

PuppetBoard (http://puppetboard.openstack.org/) so you can watch your changes get applied, or not
### PuppetBoard

<table>
<thead>
<tr>
<th>PUPPETBOARD</th>
<th>OVERVIEW</th>
<th>NODERS</th>
<th>FACTS</th>
<th>REPORTS</th>
<th>METRICS</th>
<th>QUERY</th>
</tr>
</thead>
<tbody>
<tr>
<td>changed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>afsdb02.openstack.org</td>
<td>Unknown</td>
<td>Friday, November 6 2015 2:32 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unchanged</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>afstest.openstack.org</td>
<td>Unknown</td>
<td>Friday, November 6 2015 2:32 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unchanged</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>apps.openstack.org</td>
<td>Unknown</td>
<td>Friday, November 6 2015 2:32 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>changed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ask-staging.openstack.org</td>
<td>Unknown</td>
<td>Friday, November 6 2015 2:32 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unchanged</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ask.openstack.org</td>
<td>Unknown</td>
<td>Friday, November 6 2015 2:32 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>changed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cacti.openstack.org</td>
<td>Saturday, July 5 2014 12:20 PM</td>
<td>Friday, November 6 2015 2:29 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © 2013 Daniele Sluijters.
Can you really manage an infrastructure via git commits? (3/3)

Thorough, specific documentation at
http://docs.openstack.org/infra/system-config
Well, not everything

- Automation is not perfect, sometimes you just need to log into a server
- Complicated migrations and upgrades need manual components (but we automate more every time!)
- Initial persistent server deployment still has manual components
- Passwords need to be privately managed (but we use git!)
Maintenance collaboration on Etherpad

1. Prepare a change to the project-config repo to update things like projects.yaml/ACLs, jenkins-job-builder and gerritbot for the new name. Also add changes to update projects.txt in all branches of the requirements repo and devstack-vm-gate-wrap.sh in the devstack-gate repo if necessary.

2. Stop puppet runs on the puppetmaster to prevent early application of configuration changes: [fungi] DONE
   
   ```
   sudo crontab -u root -e
   ```
   
   Comment out the crontab entries. Use `ps` to make sure that a run is not currently in progress. When it finishes, make sure the entry has not been added back to the crontab.

   
   ```
   python /opt/zuul/tools/zuul-changes.py http://zuul.openstack.org gate >gate.sh
   python /opt/zuul/tools/zuul-changes.py http://zuul.openstack.org check >check.sh
   sudo invoke-rc.d zuul stop
   sudo rm -f /var/run/zuul/zuul.pid /var/run/zuul/zuul.lock
   ```

4. Stop Gerrit on review.openstack.org: [nibalizer] DONE
   
   ```
   sudo invoke-rc.d gerrit stop
   ```

5. Update the database on review.openstack.org: [nibalizer] DONE
Human collaboration

- Main IRC channel (#openstack-infra)
- Incident IRC channel (#openstack-infra-incident)
- Sprint IRC channel (#openstack-sprint)
- Weekly IRC-based meetings (#openstack-meeting)
- Channel/Meeting logs on http://eavesdrop.openstack.org/
- Pastebin http://paste.openstack.org/
- In person collaboration at the OpenStack Design Summit every 6 months

No voice or video calls.
And then there are time zones :( 

• Increased coverage is great, but...
  – The first core/root member in a particular region struggles to feel cohesion with the team
  – Increased reluctance to land changes into production
  – Makes for slower on-boarding for key tasks

• Only solved by increasing coverage in that time zone so they're not alone
But mostly it's pretty great!

(and our team always needs help)
Questions

My real email ;) lyz@princessleia.com

HPE email: lyz@hpe.com

Freenode: pleia2

OpenStack CI Resources: http://docs.openstack.org/infra/system-config/