TripleO: OpenStack on OpenStack

Clint Byrum - HP Cloud, Master Technologist
Twitter: @SpamapS
Email: clint@fewbar.com
### Community Size & Growth

<table>
<thead>
<tr>
<th>Ecosystem Size</th>
<th>Individual Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;200 Companies</td>
<td>&gt;10,000</td>
</tr>
</tbody>
</table>

**Members:** 22 Total (4 New)  
**Sponsors:** 22 Total (6 New)  
**Supporters:** 104 Total (4 New)

**Cumulative Contributors:** 733  
**Average Monthly Contributors:** 175

**Patches Merged:** >4000/month
Velocity

→ Cloud Enables Increased Velocity
→ Cloud Drives Agility
→ Develop and Test and Deploy in Cloud
<table>
<thead>
<tr>
<th>Install / Reconfigure / Upgrade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bugs</td>
</tr>
<tr>
<td>Cruft / Entropy</td>
</tr>
<tr>
<td>Hardware failure</td>
</tr>
</tbody>
</table>
TripleO: OpenStack on OpenStack

→ Continuous Integration and Delivery
→ Drive installation and maintenance costs down
→ Encapsulate the installation and upgrade process
→ Common API and infrastructure for above and below cloud
<table>
<thead>
<tr>
<th>Install / Reconfigure / Upgrade</th>
<th>Bugs</th>
<th>Cruft / Entropy</th>
<th>Hardware failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI/CD</td>
<td></td>
<td>Golden images</td>
<td>HA setup</td>
</tr>
</tbody>
</table>
Developer uploads a change

Jenkins determines which images are affected, builds them, and tests that change in isolation

Bare Metal Nodes

build image

save result

deploy on bare metal & test the image

Inject latest image into a Heat template, then test both deployment and upgrade.

and finally use Heat to orchestrate an upgrade of any image(s) affected by the change

Bare Metal Nodes

start bootstrap cloud

load all necessary images, including the new image(s)

use Heat to orchestrate a multi-node baremetal deployment

generate load and test it with Tempest

Have confidence and publish the image(s)

Bare Metal Nodes

use Heat to deploy to production
Provisioning  |  Software  |  Configuration  |  State  |  Orchestration
--- | --- | --- | --- | ---
Nova  | diskimage-builder  | os-apply-config  | os-refresh-config  | Heat
MaaS  |  |  |  |  
Crowbar  |  |  |  |  
Razor  |  |  |  |  
Manual Install  |  |  |  |  
Stand-alone Chef, Puppet, etc  |  |  |  |  
~~~ Chef ~~~  |  |  |  |  
~~~ Puppet ~~~  |  |  |  |  
~~~ Puppet ~~~  |  |  |  |  
~~~ Chef ~~~  |  |  |  |  
Crowbar  |  |  |  |  

Components

→ Nova bare metal / Ironic™
→ Heat
→ diskimage-builder
→ os-apply-config
→ os-refresh-config
→ os-collect-config
→ tripleo-image-elements
→ tripleo-heat-templates
Deployment

→ Heat stack defines the cluster
→ Heat drives the Nova API to deliver images to machines
→ Virtual machines in developer test
→ Bare metal Nova for CI/CD and production deployment
Nova bare metal / Ironic™

- nova-compute
- PXE
- IPMI
- Your machine image
Heat

→ Focus on orchestration
→ Supports any CM system within a machine
  → Use Puppet or Chef or Salt if you like
→ Delivers configuration metadata to machines
→ Accepts exported metadata from machines
→ OpenStack templates - tripleo-heat-templates
Golden Images

→ Encapsulate a known good set of software
→ Excludes configuration and persistent state
  → these are placed on a separate state partition
  → / is then mounted read-only
→ Equivalent of packages at a cluster level
→ Each image can be tested and then deployed as-is
  → Because the configuration is not part of the image
→ Small focused toolchain to build images
  → diskimage-builder
  → tripleo-image-elements
os-*-config

→ os-collect-config grabs new metadata from heat

os-refresh-config:
→ Quiesce fragile services
→ (If needed) Upgrade software from glance
→ (If needed) os-apply-config applies config files
→ (If needed) Reboot
→ Ensure required services are running and/or restarted
→ Perform any migrations (such as seeding initial data)
→ Notify heat that the deploy is complete on the machine
Performance

→ Installation code executes at Image Build time
→ FAST deployment

6 minutes from power off to working machine
Under and Over cloud

Nova cannot reliably run two different hypervisors in one cloud today

So we run two+ clouds:
→ the undercloud, a bare metal cloud that runs on, and owns, all the hardware
→ the overcloud, a regular VM based cloud running as a tenant on the bare metal cloud
→ additional VM clouds can run as parallel tenants on the undercloud (e.g. for testing).
Undercloud

→ Fully HA Bare metal OpenStack
→ Self hosted: nodes in the control plane are tenants within it
→ Aiming for as few as 2 machines for the control plane
→ All additional nodes are available for the overcloud tenant
Overcloud

→ Fully HA KVM based OpenStack hosted by the undercloud
→ Orchestrated by Heat running in the undercloud
→ Can (optionally) use the same disk images for most services
Installation

→ Special case of normal deployment
→ Run a collapsed cluster:
  - a single image with Heat + Nova Bare metal in a VM
→ Bridge that to the new data center network
→ Enroll the machines
→ Tell Heat that we want an HA configuration
→ Wait while it scale the undercloud out
→ Switch off the VM image
→ Tell Heat to recover from the loss of the VM node (by scaling out again)
→ Deploy the overcloud as a tenant
Upgrade, the simple version

→ HA/scale out services know how to react to dead nodes
→ Rolling upgrade:
  - heat deletes one node
  - heat creates new node with new version
  - lather, rinse, repeat
Upgrade, the complex version

→ Some nodes have precious data
→ Cloudify the install:
   - Node and root FS are ephemeral
   - Root FS specialization through cloud metadata
   - Precious data goes in attached volumes
→ Make the Root Image metadata passed in via heat
→ Upgrade changes the image id of the instance
→ takeovernode:
   - Download image
   - Mount in temp dir
   - Overwrite old Root FS with new Root FS
→ Services need to be restarted or node needs reboot:
   - Remember os-refresh-config?
Future engineering

→ cinder: Local volumes
→ neutron: Switch configuration
→ ironic: Support booting from newer kernels
Thank you