TOMORROW starts here.
OpenStack for Service Providers and Enterprise

BRKSPG-2644

Shannon McFarland – CCIE #5245
Principal Engineer

#clmel
Agenda

- Enterprise Cloud Trends (listen up Cloud providers ;-) )
- What is OpenStack?
- OpenStack Participation
- What are Enterprise/SPs doing with OpenStack?
- OpenStack Deployment
- Cisco Product Integration
- Conclusion
Enterprise Cloud Trends
Enterprise Trends – Cloud

Virtualisation (Server, Storage, App, etc)

Cost driven
Horrible mistake

Public/ Hybrid Cloud

Missed expectations:
- Cost
- HA
- Performance
- Ops

Public Cloud Retraction

Cloud done their way:
- Self-service
- Reset cost expectations
- Elastic
- Understand Cloud HA
- Multi-tenancy
- IT meet DevOps

Private Cloud
Enterprise Trends – Cont.

Old-School Server Virtualisation

- Learned lots of stuff:
  - Hypervisors
  - Multi-DC
  - Virtual networking
  - etc..

Cloud with Hypervisors

- OpenStack
  - KVM
  - Zen
  - ESXi
  - Hyper-V
- Eucalyptus
- Cloudstack

Cloud with Containers

- OpenStack
  - Docker
- It’s about the App, stupid
  - Docker + stuff
  - CoreOS
  - Kubernetes
  - Mezos
  - etc..

?
Insanity = Doing the Same Thing Over and Over Expecting a Different Result

• Cool and exciting technologies are borderline useless if IT process & change control don’t adapt
• Elastic, self-service, FastIT, are all the enemy of legacy IT models

Changing technology hoping for different results when IT policies don’t change to meet new requirements
CI/CD as a Prereq to OpenStack

- The biggest issue with OpenStack is actually not OpenStack itself but the operational processes that surround it
- DevOps – Learn it, Live it, Love it: http://www.jedi.be/blog/2012/05/12/codifying-devops-area-practices/
- CI/CD – The make or break process that your customer has to understand
- Build the processes BEFORE building the OpenStack environment
- Remember, OpenStack was built for modern-day distributed web applications that are driven by developers
High-Level CI/CD Overview

Continuous Integration

- Revision Control System
  - git

- Code Review Tool
  - Gerrit/Git pull request

- Code Repo
  - GitHub

- Integration Server
  - Jenkins/Hudson, Zuul, CloudBees, Go, Maven, etc..

- Test Jobs
  - Tempest, Rally, puppet-rspec, tox, etc..

- Artifacts
  - rpmbuild, Jenkins, Artifactory, Apache Archiva, etc..

- RCS: Subversion, Mercurial, CVS, Bazaar, Perforce, ClearCase, etc..

- Code Review: Gerrit, Git pull request, Phabricator, Barkeep, Gitlab, etc..

- Code Repo: GitHub, BitBucket, BitKeeper, Gitorious, etc..

- Integration Server: Jenkins/Hudson, Zuul, CloudBees, Go, Maven, etc..

- Test Jobs: Tempest, Rally, puppet-rspec, tox, etc..

- Artifacts: rpmbuild, Jenkins, Artifactory, Apache Archiva, etc..

*See notes for logo credits
What is OpenStack?
“OpenStack is a collection of open source technologies delivering a massively scalable cloud operating system”

- openstack.org
OpenStack Cloud Computing Software

- Freely available, open source software allowing anyone to build their own private or public clouds.
- Open source and open APIs allows the customer to avoid being locked into a single vendor.
- Built by a growing community of contributors.
- Opportunities for vendors to develop their own solutions and services.

## OpenStack is “Project” Based

### Core Projects Shown

<table>
<thead>
<tr>
<th><strong>Compute</strong></th>
<th><strong>Storage</strong></th>
<th><strong>Dashboard</strong></th>
<th><strong>Identity</strong></th>
<th><strong>Networking</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>“Nova”</td>
<td>“Glance, Swift, Cinder”</td>
<td>“Horizon”</td>
<td>“Keystone”</td>
<td>“Neutron”</td>
</tr>
<tr>
<td>- Houses VMs</td>
<td>- Instance/VM image storage</td>
<td>- Web app for controlling OpenStack resources</td>
<td>- Centralised policies</td>
<td>- Networking as a service</td>
</tr>
<tr>
<td>- API driven</td>
<td>- Cloud object storage</td>
<td>- Self-service portal</td>
<td>- Tenant mgmt.</td>
<td>- Multiple models</td>
</tr>
<tr>
<td>- Support for multi-hypervisors</td>
<td>- Persistent block level storage</td>
<td></td>
<td>- RBAC</td>
<td>- IP address mgmt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Plugins to external HW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Telemetry</strong></th>
<th><strong>Orchestration</strong></th>
<th><strong>Database</strong></th>
<th><strong>Data Processing</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>“Ceilometer”</td>
<td>“Heat”</td>
<td>“Trove”</td>
<td>“Sahara”</td>
</tr>
<tr>
<td>- Central collection point</td>
<td>- Template-based orchestration engine</td>
<td>- DBaaS</td>
<td>- Fast provisioning of Hadoop clusters</td>
</tr>
<tr>
<td>- Metering and monitoring</td>
<td>- More rapid deployment of applications</td>
<td>- Single-tenant DB within instance</td>
<td></td>
</tr>
</tbody>
</table>

*New!*. *Cisco Public*
What’s New in Juno

• PTL Juno Summary: [https://www.youtube.com/playlist?list=PLKqaoAnDyfgqpx5f3PCuOgsDm-UJu2aU](https://www.youtube.com/playlist?list=PLKqaoAnDyfgqpx5f3PCuOgsDm-UJu2aU)

• Nova
  – NFV
  – Improvements in live upgrades (introduced in Icehouse)

• Heat
  – Rollback
  – non-Admin resource creation

• Neutron
  – Distributed Virtual Router
  – L3 HA
  – New LBaaS API
Why Does OpenStack Matter?

• Choice
  – There is no one-size fits all option for cloud computing – *Amazon is cool but not the be all/end all solution*
  – There is no single vendor who can fill all needs of a cloud stack – You will likely engage with multiple partners

• Community
  – Open Source
  – Community driven – Individual, organisational
  – Better time-to-market and faster feature velocity

• Commercialisation
  – Start with the ‘baseline’ OpenStack components
  – Vendor opportunities for value-add integration on top of OpenStack baseline
    • Design, deployment, automation, operation, high-availability, applications, etc…
Who is Involved in OpenStack?

- You name it – Compute, Storage, Networking vendors, Universities, Gov’t, massive pile of OpenStack-specific startups
- Traditional HW vendors – Cisco, HP, Dell, etc…
- Providers – Rackspace, AT&T, Comcast, etc…
- Startups – PistonCloud, SwiftStack and many, many more…
- Distributions & Support – Red Hat, Canonical, SUSE
- Some are focused on only small parts of OpenStack such as driving object storage features (SwiftStack), or automated deployment and support (PistonCloud) or networking and compute pull-thru as well as project leadership (Cisco – Nexus, UCS, services, Neutron)
Cisco’s Focus on OpenStack - Today

Cisco Designs on specific releases in ‘beachhead’ accounts
- Start simple, build from there – Focus on automation and HA
- Evangelisation of what Cisco is doing - Thought Leadership – Help customers know What, When, Where & How

Community
- Neutron – Network Service
- Horizon – Dashboard
- Keystone – Identity
- Swift – Object Storage
- Ceph/Cinder – Block Storage
- Automation – PuppetLabs
- HA Design

Engineering
- Cisco Product Integration
- Nexus Plugins – Neutron
- UCS
- CSR/ASR
- Co-developed solutions (Red Hat, Canonical, SUSE)

Customers
Cisco + Other Distributions/Vendors


- Red Hat:


Distro/Vendor Supported Installers


• Canonical/Ubuntu – MAAS and JuJu: http://www.ubuntu.com/cloud

• SUSE: https://www.suse.com/products/suse-cloud/features/

• Mirantis Fuel: http://software.mirantis.com/main/

• Piston Cloud: http://www.pistoncloud.com/

• Others …
Red Hat - Packstack

- Meant for single/few host deployments in NON-production deployments:
  - https://openstack.redhat.com/Main_Page

- Install Packstack:
  ```
yum install -y openstack-packstack
  ```

- Generate SSH keys (or let Packstack do it):
  ```
  ssh-keygen
  ```

- Generate an answer file (or just run ‘packstack’ and follow the prompts):
  ```
  packstack --gen-answer-file=~/ answers.cfg
  ```

- Run the answer file:
  ```
  packstack --answer-file=~/ answers.cfg
  ```
Installer Demo
What are Enterprise/SPs Doing with OpenStack?
Common Enterprise Use Cases

• OpenStack, at least today, is targeted at hosting modern day distributed applications written for the cloud – This isn’t your grandpa’s server virtualisation platform built for individual VM HA/Mobility

• Sandbox environments
  – A place to research, learn and test CI/CD processes
  – PoC web applications along with ‘practicing’ the new DevOps methodology
  – A place to learn the whole cloud deployment framework, document, train, move to production

• Development environments
  – Using the lessons learned in the sandbox phase:
    • Build Dev, QA and production environments
    • Apply CI/CD processes
    • Slow-role Web application deployment either on ‘standard’ OpenStack or in conjunction with a PaaS deployment

• Data Processing environments – Big Data clusters, etc..

• Training systems – Cheap and fast to build and tear down for each class

• Revenue generating applications – Vertical applications
Telco’s are Turning to OpenStack for NFV

- Resource Allocation & Optimisation
- Resource Isolation
- Networking
  - WAN orchestration
  - VNF provisioning
- Real Time Response
  - Interrupt servicing
  - OVS latency
- Carrier Grade Security
  - Multi-tenancy with end-to-end isolation
- Software Management and Upgrade Support
  - Hitless & automated upgrades
- Backup and Restore
  - Automatic backup
- Audit and Trouble Shooting
  - Audit log, monitor
- Assurance:
  - High Availability
    - Mitigation of failures
    - Fault monitoring and health check

https://wiki.openstack.org/wiki/Teams/NFV
Shock-and-Awe: Dashboard is Not Where Tenants Do Their Work

Overview

Limit Summary

- Instances: Used 3 of 10
- VCPUs: Used 10 of 20
- RAM: Used 20.0 GB of 50.0 GB
- Floating IPs: Used 0 of 50
- Security Groups: Used 1 of 10

Select a period of time to query its usage:

From: 2014-09-01 To: 2014-09-01

The date should be in YYYY-mm-dd format.

Active Instances: 3
Active RAM: 20GB
This Period's VCPU-Hours: 49.85
This Period's GB-Hours: 1994.02

Usage Summary
Cloud Apps Deployment – Automate it

- Cloud-init for Puppet/Chef/etc..
- Image already has agent/script

```bash
nova boot --user-data ./cloud-config-puppet.txt --image precise-x86_64 --flavor m1.tiny --key-name ctrl-key --nic net-id=42823c88-bb86-4e9a-9f7b-ef1c0631e5e sales-web-01
```

```puppet
node 'sales-web-01' {
  include lamp
}
```

```
root@build-server:~# tree /etc/puppet/modules/lamp/
/etc/puppet/modules/lamp/
|-- files
   |-- apache2.conf
   |-- index.php
   `-- php5.conf
   `-- manifests
       `-- init.pp
```
Cloud Apps Deployment - Heat

- Growing interest in Heat-based deployments
- Today, Heat orchestrates resources inside a tenant space
- [http://blog.scottlowe.org/2014/05/01/an-introduction-to-openstack-heat/](http://blog.scottlowe.org/2014/05/01/an-introduction-to-openstack-heat/)
Heat Demo

https://github.com/shmcfarl/my-heat-templates
Baseline vs. Premium OpenStack Deployments
Common Baseline Components - Example

OpenStack Platform

- Network
  - Neutron
  - ML2
  - OVS

- Compute
  - Nova
  - KVM

- Infrastructure
  - Haproxy/Keepalived

- Storage
  - Swift
  - Ceph Object GW
  - Ceph Block RBD

- Orchestration
  - Cinder
  - Glance

- etc.
Common Premium Components - Example

OpenStack Platform

Network
- Neutron
- Infrastructure
  - ML2
  - OVS
  - Cisco Nexus
  - Linux Bridge

Compute
- Nova
  - KVM
  - Zen

Storage
- Swift
  - SwiftStack
  - SOLIDFIRE

Orchestration
- Cinder
- Glance

etc...
What Really Changes in My Data Centre?

• OpenStack components live South of the Top-of-Rack switch
• Your existing DC, Internet Edge and BN architecture stays the same
• It’s about the compute, storage and orchestration/management tiers
• Your apps go largely unchanged
To Automate or Not and How Much to Automate

• Single Shot – Manually setup everything:
  – Deep appreciation for what installers do
  – Best way to learn how the components of OpenStack communicate

• Semi-Automatic – Use automation for ‘some’ of the setup and maintain/modify manually:
  – See slide on installers

• Automatic – Install > Operate > Upgrade
  – CI/CD a huge part of this flow
Network Decisions

- OpenStack Networking
  - [http://docs.openstack.org/admin-guide-cloud/content/section_networking-scenarios.html](http://docs.openstack.org/admin-guide-cloud/content/section_networking-scenarios.html)
  - Many vendor plugins (ML2/OVS, Ryu, etc.)
  - Flat, Routers with NAT, VLAN Trunking, GRE, VXLAN

- Scale
  - VLAN number limitations for large tenant + networking environments
  - GRE/VXLAN – Throughput impact, especially on older releases
  - Service scale – i.e. VPNaaS mesh

- IPv6 – Minimally useable in Juno (without patches), MUCH better in Kilo

- Network Tuning – Linux kernel, networking and vSwitch-specific (OVS) tuning is critical:
Network Topologies – Overlays (GRE, VXLAN, etc) and VLANs
Tenant IPv6 Address Options

Option 1
Cloud Provider-assigned Addressing

Tenant 1
2001:420::/32

Web Server
App Server

::A
::2

::BAD:BEEF::/64
::DEAD:BEEF::/64

Option 2
Cloud Provider-assigned Addressing + ULA

Tenant 1
2001:420::/32

Web Server
App Server

::A
::2

::BAD:BEEF::/64
::DEAD:BEEF::/64
::1000::/64
::2000::/64

Tenant 2
2001:420::/32

Web Server
App Server

::A
::2

::BAD:BEEF::/64
::DEAD:BEEF::/64
::1000::/64
::2000::/64

Option 3
Tenant Brings Addressing

Tenant 1
Tenant 2
Tenant 1 = 2001:DB8:1::/48
Tenant 2 = 2001:DB8:2::/48

ULA Block/48
ULA Block/48

Option 4
Prefix Translation

FD9C:58ED:7D73:1::/64
FDDE:50EE:79DA:1::/64

Xlate/Proxy

Don’t do this

Tenant 1
Tenant 2
Tenant 1 = 2001:DB8:1::/48
Tenant 2 = 2001:DB8:2::/48

ULA Block/48
ULA Block/48
The Hard Stuff – IPv6 + Cloud

• If you look at most of the private cloud infrastructure components – most of the ‘magic’ occurs South of the ToR

• North of the ToR is mostly plain-Jane networking (L2/L3, SLB, Security)

• Inside of a private cloud stack you have a lot of moving parts and they all ride on IP:
  – API endpoints
  – Provisioning, Orchestration and Management services
  – Boatload of protocols and databases and high-availability components
  – Virtual networking services <> Physical networking

• Two common approaches for IPv6 support:
  – Dual-Stack everything (Service Tier + Tenant Access Tier [Tenant management interface along with VM network access])
  – Conditional Dual stack (Tenant Access Tier only – API endpoints & DBs are still IPv4)
# Dual-Stack Everything

<table>
<thead>
<tr>
<th>Service Tier</th>
<th>Tenant Access Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4/IPv6 API endpoints</td>
<td>IPv4/IPv6 VM Operating System</td>
</tr>
<tr>
<td>IPv4/IPv6 Database(s)</td>
<td>IPv4/IPv6 Virtual Networking (L2/L3)</td>
</tr>
<tr>
<td>IPv4/IPv6 Interface (GUI, CLI)</td>
<td>IPv4/IPv6 Tenant Interface (GUI, CLI)</td>
</tr>
</tbody>
</table>

# Conditional Dual-Stack

<table>
<thead>
<tr>
<th>Service Tier</th>
<th>Tenant Access Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4/IPv6 API endpoints</td>
<td>IPv4/IPv6 VM Operating System</td>
</tr>
<tr>
<td>IPv4/IPv6 Database(s)</td>
<td>IPv4/IPv6 Virtual Networking (L2/L3)</td>
</tr>
<tr>
<td>IPv4/IPv6 Interface (GUI, CLI)</td>
<td>IPv4/IPv6 Tenant Interface (GUI, CLI)</td>
</tr>
<tr>
<td>IPv4/IPv6 Tenant Interface (GUI, CLI)</td>
<td>IPv6 Tenant Interface (GUI, CLI)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service Tier</th>
<th>Tenant Access Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4/IPv6 API endpoints</td>
<td>IPv4/IPv6 VM Operating System</td>
</tr>
<tr>
<td>IPv4/IPv6 Database(s)</td>
<td>IPv4/IPv6 Virtual Networking (L2/L3)</td>
</tr>
<tr>
<td>IPv4/IPv6 Interface (GUI, CLI)</td>
<td>IPv4/IPv6 Tenant Interface (GUI, CLI)</td>
</tr>
<tr>
<td>IPv4/IPv6 Tenant Interface (GUI, CLI)</td>
<td>IPv6 Tenant Interface (GUI, CLI)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service Tier</th>
<th>Tenant Access Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4/IPv6 API endpoints</td>
<td>IPv4/IPv6 VM Operating System</td>
</tr>
<tr>
<td>IPv4/IPv6 Database(s)</td>
<td>IPv4/IPv6 Virtual Networking (L2/L3)</td>
</tr>
<tr>
<td>IPv4/IPv6 Interface (GUI, CLI)</td>
<td>IPv4/IPv6 Tenant Interface (GUI, CLI)</td>
</tr>
<tr>
<td>IPv4/IPv6 Tenant Interface (GUI, CLI)</td>
<td>IPv6 Tenant Interface (GUI, CLI)</td>
</tr>
</tbody>
</table>
High Availability Decisions

• Know what you don’t know

• Pick your release – HA matures on every release: Folsom (sucked for HA) -> Grizzly (getting better) -> Havana (Making progress) -> Icehouse (you get the idea) - You may have to use other open source tools to get a complete system highly available

• Many components are:
  – Databases: Options include MySQL-WSREP and Galera
  – Message Queue: RabbitMQ Clustering and RabbitMQ Mirrored Queues
  – API/Web services: HAProxy, Keepalived, traditional SLB
  – Swift proxy nodes: HAProxy, Keepalived, traditional SLB
  – Swift nodes: Architecturally designed to be available (i.e. multiple copies of objects)
  – Compute node: Nothing directly HA, but can use Migration for planned maintenance windows

• Puppet HA: Search “puppet master redundancy” or “masterless puppet” – you will land plenty of reading choices ;)

All-in-One (AIO) – Getting Started

AIO Controller/Compute/Storage

- MySQL, MariaDB, etc
- RabbitMQ, Qpid, etc.
- API Endpoints:
  - Keystone
  - Glance
  - Nova
  - Neutron
  - Cinder
  - Heat
  - Swift

Compute/Storage

AIO Controller

- Compute/Storage

AIO Controller

- Compute
- Storage
- Compute
- Storage
- Compute
- Storage
All-in-One (AIO) Compressed HA

AIO Controllers:
- Galera/MySQL
- RabbitMQ

API Endpoints:
- Keystone
- Glance
- Nova
- Neutron
- Cinder
- Heat
- Swift

Data Centre Infrastructure

Infrastructure Services

- SLB
- Build/PXE
- Automation
- DNS
- DHCP
- NTP
- Logging

Spine/Agg Layer

TOR(s)

OOB

AIO Controller

Compute

Network Node(s)

Block Storage

Object Storage

Block Storage

Object Storage

Object Storage
Service Cloud + Tenant Cloud

Data Centre Infrastructure

- Spine/Agg Layer
  - TOR(s)
  - Spine/Agg Layer
  - TOR(s)
  - TOR(s)
  - TOR(s)

Object Storage
- Galera
- RabbitMQ
- API Endpoints
- Compute
- Block Storage
- Object Storage

Service Cloud
- Compute
- Block Storage
- Object Storage
- Network Node(s)

Tenant Cloud
- Compute
- Block Storage
- Object Storage
- Network Node(s)
What’s a Service Cloud?

• It’s the ‘under cloud’

• Used as a hosting platform for tenant cloud services – usually in a large cloud (1000s of instances with 100-1000s of tenants)

• It is an OpenStack deployment that will host (virtually) the OpenStack control functions used by each tenant
Services
LBaaS

• A service to provide basic load-balancing of VMs/Instances within the OpenStack cluster
• The default LB provider is HAProxy
• Can leverage plugins for LBaaS to control external virtual or physical load-balancers (i.e. F5, A10, Citrix)
VPNaaS

• A service to provide IPsec VPN connectivity on a per-router/per-tenant basis
• Manual configuration via CLI or OpenStack Dashboard
• As with any IPsec site-to-site VPN, large deployments with lots of sites/tenants will require a lot of configuration due to mesh-type connectivity
• Cisco provides CSR as a means of deploying VPN
VPN Topology

Branch 1

Branch 2

Net10: 10.10.10.0/24

Net15: 10.10.15.0/24

Net20: 10.10.20.0/24

Net25: 10.10.25.0/24

Router IP: 192.168.3.5

Router IP: 192.168.5.30
Storage
References for Storage Info

- OpenStack Storage: https://www.openstack.org/software/openstack-storage/
- Block Storage: http://docs.openstack.org/havana/config-reference/content/ch_configuring-openstack-block-storage.html
- Object Storage: http://docs.openstack.org/havana/config-reference/content/ch_configuring-object-storage.html
- CEPH Storage: http://ceph.com/docs/master/rados/
  - http://www.inktank.com/resource/type/presentations/
  - http://www.slideshare.net/Inktank_Ceph/scaling-ceph-at-cern
Running Applications
Multiple Paths to Managing Images/Apps

• Docker:
  – http://www.docker.io/

• VMBuilder:
  – https://launchpad.net/vmbuilder

• Disk Image Builder:
  – https://github.com/stackforge/diskimage-builder

• Heat – Template based orchestration engine:
  – https://github.com/openstack/heat

• Salt Cloud
  – https://github.com/saltstack/salt-cloud

• Baseline images + automated application deployment (scripts, Puppet, Chef)
• Template images – Prebuilt with apps installed and deployed from Glance
Product Integration Overview


• Cisco Nexus + OpenStack Deployment: http://docwiki.cisco.com/wiki/OpenStack:_Havana:_2-Role_Nexus


Support

- Community model is like any other open source community support model
  - [http://docs.openstack.org/grizzly/openstack-compute/admin/content/community-support.html](http://docs.openstack.org/grizzly/openstack-compute/admin/content/community-support.html)
  - [http://ask.openstack.org](http://ask.openstack.org)

- Cisco AS - Assessments, plans, design, implement, support & optimise

- Cisco + Partnerships

- Channel Partners – Build a practice now!!
Conclusion

• Next time: Scale, HA, apps, network design impact and some new breakouts (OpenStack storage session)

• OpenStack is for real and maturing at a rapid pace

• Many different players involved and it is evolving rapidly

• Align yourself with market leaders who have strong partnerships

• There is still a lot of focus on getting OpenStack Deployed, but we are progressing rapidly towards true operational issues:
  – Scale
  – Application deployment
  – Upgrades

• Start now!

• Get involved in the community – open source enjoys the major advantage of feature velocity
Complete Your Online Session Evaluation

Give us your feedback and receive a Cisco Live 2015 T-Shirt!

Complete your Overall Event Survey and 5 Session Evaluations.

• Directly from your mobile device on the Cisco Live Mobile App
• By visiting the Cisco Live Mobile Site http://showcase.genie-connect.com/clmelbourne2015
• Visit any Cisco Live Internet Station located throughout the venue

T-Shirts can be collected in the World of Solutions on Friday 20 March 12:00pm - 2:00pm

Learn online with Cisco Live!
Visit us online after the conference for full access to session videos and presentations. www.CiscoLiveAPAC.com
Thank you.
Understanding Neutron + OVS for example.com
Example – Network Layout

Host View

Management Network: 10.121.13.x

Public: 192.168.238.x/24
Example – OVS Bridge/Neutron Router

“br-int” view

root@control-server:~# ovs-vsctl list-ports br-int
int-br-ex
patch-tun
qr-024a0619-71
qr-10f02a4b-ab
qr-b37e1034-06
qr-ef7c1e0c-79
tap2340872e-68
tap271689cd-23
tap3fe91abf-c8
tap60a25081-14
tap6d3911a5-44

Public: 192.168.238.x/24
Example – OVS Bridge/Neutron Router

“br-int” view qr-xx & tapxx

compute-server

root@control-server:~# ovs-vsctl list-ports br-int
int-br-ex
patch-tun
qr-024a0619-71
qr-10f02a4b-ab
qr-b37e1034-06
qr-ef7c1e0c-79
tap2340872e-68
tap271689cd-23
tap3fe91abf-c8
tap60a25081-14
tap6d3911a5-44

A tap interface for each network used for DHCP service:
68=10.10.10.2
23=10.10.15.2
c8=192.168.238.5
14=10.10.20.2
44=10.10.25.2

control-server

Public: 192.168.238.x/24

VM VM VM
Example – OVS Bridge/Neutron Router

“br-ex” & br-tun view

```
root@control-server:~# ovs-vsctl list-ports br-ex
eth1
phy-br-ex
qg-8a8db076-b3
```

```
root@control-server:~# ovs-vsctl list-ports br-tun
gre-1
gre-3
patch-int
```

Management Network: 10.121.13.x
Public: 192.168.238.x/24
Example – OVS Bridge/Neutron Router

compute-server01 “br-int” view

compute-server01

control-server

root@compute-server01:~# ovs-vsctl list-ports br-int
patch-tun
qvo180f8458-7b
qvo3e60deda-cc
qvo92774056-da

Public: 192.168.238.x/24
Example – OVS Bridge/Neutron Router

*Thanks to Etsuji Nakai for the original detailed overview of OVS/Neutron ports:

```
root@compute-server01:~# brctl show

bridge name   bridge id               STP enabled     interfaces
br-int         0000.5e15d719a548       no              int-br-ex
               qvo180f8458-7b
               qvo3e60deda-cc
               qvo92774056-da
br-tun         0000.febc48d02540       no              qvb180f8458-7b
               vnet0
               qvb3e60deda-cc
               vnet2
               qvb92774056-da
               vnet1
```
Example – OVS Bridge/Neutron Router

```
root@compute-server01:~# ovs-vsctl show
ac44a899-5f10-4ff9-8dad-902fa7c10e5e
...
Bridge br-tun
  Port "gre-2"
    Interface "gre-2"
      type: gre
      options: {in_key=flow, out_key=flow, remote_ip="10.121.13.50"}
  Port patch-int
    Interface patch-int
      type: patch
      options: {peer=patch-tun}
  Port "gre-3"
    Interface "gre-3"
      type: gre
      options: {in_key=flow, out_key=flow, remote_ip="10.121.13.52"}
  Port br-tun
    Interface br-tun
      type: internal
```
Example – Basic VM Traffic Flow

High-Level Walk-Thru

VM Boots
compute-server01

br-int

patch-tun

patch-int

eth0

10.121.13.51

GRE tunnel

Management Network: 10.121.13.x

DHCP

control-server

IP Tables/Floating IP

Public: 192.168.238.x/24

Qrouter

compute-server01

br-int

patch-tun

patch-int

eth10

10.121.13.50

10.10.10.2

NAT

VM Boots

DHCP

comput-server01

br-int

patch-tun

patch-int

eth10

10.121.13.50

10.10.10.2

NAT

VM Boots

DHCP

compute-server01

br-int

patch-tun

patch-int

eth10

10.121.13.50

10.10.10.2

NAT

VM Boots

DHCP

compute-server01

br-int

patch-tun

patch-int

eth10

10.121.13.50

10.10.10.2

NAT

VM Boots

DHCP
Monitoring
Basic Monitoring is Available

Nagios/Graphite/Collectd
- http://<build-server>/nagios3 - Health monitoring of OpenStack nodes

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>compute-server01</td>
<td>UP</td>
<td>2013-03-19 12:03:47</td>
<td>4d 1h 29m 7s</td>
<td>PING OK - Packet loss = 0%, RTA = 0.27 ms</td>
</tr>
<tr>
<td>compute-server02</td>
<td>UP</td>
<td>2013-03-19 12:04:57</td>
<td>4d 1h 28m 57s</td>
<td>PING OK - Packet loss = 0%, RTA = 0.31 ms</td>
</tr>
<tr>
<td>control-server</td>
<td>UP</td>
<td>2013-03-19 12:05:47</td>
<td>0d 1h 9m 8s</td>
<td>PING OK - Packet loss = 0%, RTA = 0.30 ms</td>
</tr>
</tbody>
</table>

- http://<build-server>:8190 – Main Graphite performance console
- http://www.nagios.org/
- http://graphite.wikidot.com/
- http://collectd.org/