OpenStack Deployment in the Enterprise & Service Provider

Shannon McFarland – CCIE #5245
Distinguished Engineer
@eyepv6
BRKDCN-2367
Questions?
Use Cisco Spark to chat with the speaker after the session

How
1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install Spark or go directly to the space
4. Enter messages/questions in the space

Cisco Spark spaces will be available until July 3, 2017.
Call to Action

• Breakout/Labs sessions:
  • OpenStack Deployment for Enterprise and Service Provider (BRKDCN-2367)
  • OpenStack Integration with Cisco ACI (LTRACI-2225)
  • Mastering OpenStack and ACI (BRKACI-3456)
  • Cisco Virtualized Packet Core on OpenStack (LABSPM-2013)
  • Programmable VXLAN Fabric using VTS with OpenStack/VMware Integration (LTRDCN-2001)

• DevNet zone related sessions:
  • Getting Started with OpenStack (DEVNET-1101)
  • OpenStack Enabling DevOps (DEVNET-1104)
  • Workshop - Getting Started with OpenStack (DEVNET-1211)
  • OpenStack with OpenDaylight (DEVNET-2041)
  • Best REST in OpenStack (DEVNET-2004)
  • Cisco VIM for OpenStack based NFVI Solution (DEVNET-2570)
Agenda

• Cloud Trends
• What is OpenStack?
• OpenStack Participation
• What are Enterprises & SPs doing with OpenStack?
• OpenStack Deployment
• Conclusion
Enterprise Trends – Cloud

Virtualization
(Server, Storage, App, etc.)

Public/Hybrid Cloud

Public Cloud Retraction

Private Cloud

- Cost driven - mistake

Missed expectations:
- Cost
- HA
- Performance
- Ops

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

Some skip the public cloud step
Legacy IT Change Control <> Diametrically Opposed to Cloud

• Cool and exciting technologies are borderline useless if IT process & change control don’t adapt

• Elastic and self-service are all the enemy of legacy IT models

• DevOps won’t provide a clean slate to traditional IT processes – It’s as much ‘mindset’ as it is process and tools
Hard Decisions - Where Does OpenStack Fit?

- **Line of Biz**
  - VDI
  - Dev/QA
  - Pet VMs

- **Cloud Native/Modern Line of Business**
  - Dev/QA
  - Pet VMs

- **Cloud Native/Modern Line of Business**
  - Dev/QA

- **VMware/Microsoft Virtualization**

- **OpenStack**

- **Container/PaaS**
Hard Decisions - Where Does OpenStack Fit?

VMware/Microsoft Virtualization → Cloud Native/Modern Line of Business

Line of Biz: VDI, Dev/QA, Pet VMs

Dev/QA: Container/PaaS
Hard Decisions - Where Does OpenStack Fit?

• Bottom line: You will likely have multiple environments - each built for the correct use case

• OpenStack can be the ‘dump all’ for all of them but it is going to hurt…at least for now
Continuous Integration/Continuous Deployment Operational Process for an Upgradeable OpenStack

- The biggest issues with OpenStack is less about OpenStack itself but the operational processes that surround it

- DevOps – Learn it, Live it, Love it

- CI/CD – The make or break process that you have to understand (for Cloud Native apps and the infrastructure that supports it)

- 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

Revision Control System

Code Review Tool
(Gerrit/Git pull request)

Code Repo
(GitHub)

Artifact Creation
Artifact Rep Mgr
(rpmbuild/Jenkins/etc)

Integration Server

Test Jobs
(Tempest/Rally/etc)

Deployment Jobs

Continuous Integration

Continuous Deployment

• 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 in to a single vendor
- Built by a growing community of contributors
- Opportunities for vendors to develop their own solutions and services

```
nova boot --flavor 1 --image 397e713c-b95b-4186-ad46-6126863ea0a9 --key-name key_pair1 my_server

nova list

swift upload my_container ~/this_object
```
## OpenStack Projects

<table>
<thead>
<tr>
<th>Compute (Nova)</th>
<th>Dashboard (Horizon)</th>
<th>Database (Trove)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network (Neutron)</td>
<td>Image (Glance)</td>
<td>Orchestration (Heat)</td>
</tr>
<tr>
<td>Object Storage (Swift)</td>
<td>Identity (Keystone)</td>
<td>Data Processing (Sahara)</td>
</tr>
<tr>
<td>Block Storage (Cinder)</td>
<td>Telemetry (Ceilometer)</td>
<td>Deployment (Kolla)</td>
</tr>
<tr>
<td>Bare Metal (Ironic)</td>
<td>DNS (Designate)</td>
<td>Application Catalog (Murano)</td>
</tr>
<tr>
<td>Containers (Magnum)</td>
<td>Key Management (Barbican)</td>
<td>Policy (Congress)</td>
</tr>
<tr>
<td>File System (Manila)</td>
<td>Messaging (Zaqar)</td>
<td>....</td>
</tr>
</tbody>
</table>
OpenStack Progress

- **Ocata** – April 2017
- **Mitaka** – April 2016
- **Kilo** – May 2015
- **Icehouse** – April 2014
- **Grizzly** – April 2013
- **Essex** – April 2012
- **Cactus** – April 2011
- **Austin** – Oct 2010
- **Bexar** – Feb 2011
- **Diablo** – Sept 2011
- **Newton** – Oct 2016
- **Juno** – Oct 2014
- **Liberty** – Oct 2015
- **Havana** – Oct 2013
- **Folsom** – Sept 2012
- **Pike** – Oct 2017

Started with Compute and Storage service
OpenStack Participation
Why Does OpenStack Matter?

• Choice
  • There is no one-size fits all option for cloud computing
  • 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, organizational
  • Better time-to-market and faster feature velocity

• Commercialization
  • 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 vendors – Cisco, IBM, etc.…

• Providers – Rackspace, AT&T, Comcast, etc.…

• Distributions & Support – Red Hat, Canonical, SUSE

• Some are focused on only parts of OpenStack such as driving object storage features (SwiftStack) or high-performance block storage (NetApp/SolidFire)

• Large amounts of consolidation happening
  • Cisco acquires Metacloud and PistonCloud
  • IBM acquires Blue Box
  • …
Cisco OpenStack Strategy
For Data Center and Cloud Solutions

Open

Relevant, large contributions to open source code

Innovation

Integrate with leading distributions

Improve OpenStack with supplemental functionality

Drive OpenStack innovation into Cisco products

Success

Build world-class global Cisco Cloud Services

Programs for every OpenStack deployment model
What are Enterprises/SPs doing with OpenStack?
Top Workloads - OpenStack

Common Enterprise Use Cases

• New/Modern As-a-Service applications (Cloud native stuff)
• A means to provide multi-tenancy to their container workloads (Docker Swarm, Kubernetes, Mesos)
• An alternative to their existing non-API driven virtualization solution
• 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/QA 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
OpenStack and Network Function Virtualization

- Performance - High-performance packet processing
- Scalability – Cells, AZ’s
- High Availability – Network resource HA
- Multi-site - application-level redundancy across different datacenters
- Service Function Chaining - overriding the basic destination-based forwarding that is typical of IP networks
Telco’s are Turning to OpenStack for NFV

- Resource Allocation & Optimization
- 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
- High Availability
  - Mitigation of failures
  - Fault monitoring and health check

https://wiki.openstack.org/wiki/TelcoWorkingGroup

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public
Dashboard vs. API + Automation

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: 2014-09-06 Submit

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

Cisco live!
Cloud App Deployment – Automate it

Boot the Instance

- Template image already app configured
- Cloud-init for Puppet/Chef/etc..
- Baseline image + Ansible
- Heat Orchestration

Rinse & Repeat

App is Deployed

Config Management

```
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-ef1c0631ee5e sales-web-01
```

- hosts: dbservers
  remote_user: root
  roles:
  - db

- hosts: webservers
  remote_user: root
  roles:
  - base-apache
  - web

- hosts: lbservers
  remote_user: root
  roles:
  - haproxy
Cloud App Deployment - Heat

- Growing interest in Heat-backed deployments
- Today, Heat orchestrates resources inside a tenant space
- https://wiki.openstack.org/wiki/Heat
- https://github.com/shmcfarl/my-heat-templates
OpenStack Deployment
Common Baseline Components - Example

OpenStack Platform

Network
- Neutron
- ML2
- OVS

Infrastructure
- Haproxy/Keepalived
- Linux Bridge

Compute
- Nova
- KVM
- ESXi/vSphere

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

Orchestration
- etc..

etc..
Common Premium Components - Example

OpenStack Platform

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

Compute
- Nova
- KVM
- ESXi/vSphere

Storage
- Swift
- Cinder

Orchestration
- Glance

Etc..
OpenStack Installation
To Automate or Not and How Much to Automate

• Why bother? Have someone else do it for you!!

• 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
Cisco Metacloud

Advanced Operational Support

- 24x7 Cloud Operations and Supports
- Infrastructure Capacity Planning
- Monitoring and Error Detection
- SLA Guarantees
- Platform and Security Updates
- Cloud Design and Deployment

In Your Data Center, on Your Hardware, Delivered as a Service
Distro/Vendor Supported Installers

- Red Hat OpenStack (RHOS/RDO) – PackStack and Red Hat Director:


- Kolla – Docker hosting OpenStack Services
  - [https://wiki.openstack.org/wiki/Kolla](https://wiki.openstack.org/wiki/Kolla)
  - [https://github.com/openstack/kolla](https://github.com/openstack/kolla)
  - [http://docs.openstack.org/developer/kolla/](http://docs.openstack.org/developer/kolla/)
  - [http://docs.openstack.org/developer/kolla/quickstart.html](http://docs.openstack.org/developer/kolla/quickstart.html) <<<-Check this out

- Kolla Kubernetes – Kubernetes hosting OpenStack Services:
  - [http://docs.openstack.org/developer/kolla-kubernetes/](http://docs.openstack.org/developer/kolla-kubernetes/)
  - [https://github.com/openstack/kolla-kubernetes](https://github.com/openstack/kolla-kubernetes)
CentOS/RHEL – RDO - Packstack

• Meant for single/few host deployments in NON-production deployments: https://www.rdoproject.org/install/quickstart/

• 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`
Kolla Workflow

OpenStack Source or Binaries

Kolla Source

Kolla-build

Docker Registry

glance_data

nova-api

nova-libvirt

heat-api

horizon

neutron-agents

Install Target

glance_data

nova-api

nova-libvirt

heat-api

horizon

neutron-agents

...
$ docker ps -a

CONTAINER ID        IMAGE                                                                        COMMAND
28d4d4f108f0c      operator.local:4000/lokolla/centos-binary-horizon:latest  "kolla_start"
778086b0b86c      operator.local:4000/lokolla/centos-binary-heat-engine:latest  "kolla_start"
c7c9679f6344      operator.local:4000/lokolla/centos-binary-heat-api-cfn:latest  "kolla_start"
f4c4814427d9      operator.local:4000/lokolla/centos-binary-heat-api:latest  "kolla_start"
b32d6f62372c      operator.local:4000/lokolla/centos-binary-neutron-agents:latest  "start.sh"
17cbf2fc6774      operator.local:4000/lokolla/centos-binary-neutron-openvswitch-agent:latest  "kolla_start"
9893da70df2f      operator.local:4000/lokolla/centos-binary-neutron-server:latest  "kolla_start"
815485091f3c      operator.local:4000/lokolla/centos-binary-neutron-vswitchd:latest  "kolla_start"
c7b68bf3feb3      operator.local:4000/lokolla/centos-binary-neutron-db-server:latest  "kolla_start"
f4c4814427d9      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
778086b0b86c      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
9893da70df2f      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
815485091f3c      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
622ae869d3c       operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
a51feef1857c      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
90e02c62b01f      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
2f10990f1e1c      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
l4977c9377a7      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
45ccdad7ef4b      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
8f31326ba6f5      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
d6e73b211a17      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
4830569d2f21      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
8b9899ac3df3      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
acac5f3aa58d      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
38bbce5c2056      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
b6abeb782ca8      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
d83e6400f58b      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
68ff8a563412      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
d48848e57060      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
59951f2abdce      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
bbee75a7541      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"
950372a37b08      operator.local:4000/lokolla/centos-binary-data:latest  "/bin/sh -c '/bin/sle"

registry:0.9.1
OpenStack Deployment Overview - Rack/Node Scale
All-in-One (AIO) – Getting Started

AIO Controller/Compute/Storage

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

AIO Controller

Compute/Storage

Compute/Storage

Compute
Storage

Compute
Storage

Compute
Storage

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public
All-in-One (AIO) Compressed HA

Data Center Infrastructure

Spine/Agg Layer

TOR(s)

Spine/Agg Layer

TOR(s)

TOR(s)

Infrastructure Services

SLB

Build/PXE

Automation

DNS

DHCP

NTP

Logging

AIO Controllers:
- Galera/MySQL
- RabbitMQ
- API Endpoints:
  - Keystone
  - Glance
  - Nova
  - Neutron
  - Cinder
  - Heat
  - Swift

AIO Controllers:
Service Cloud + Tenant Cloud

Data Center Infrastructure

Service Cloud

Tenant Cloud

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public
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
OpenStack Deployment Overview - Network
What Really Changes in my Data Center?

• 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, orchestration and management tiers
Network Decisions

- OpenStack Networking
  - [http://docs.openstack.org/admin-guide/networking.html](http://docs.openstack.org/admin-guide/networking.html)
  - Many vendor plugins
    - ML2/OVS, ML2/Linux Bridge
    - Cisco Nexus Mechanism driver for VLAN and VXLAN
  - Cisco APIC with OpenStack:

- Scale/Performance
  - VLAN number limitations for large tenant + networking environments
  - VXLAN Offload
  - SR-IOV
OpenStack Network Architecture

<table>
<thead>
<tr>
<th>Network</th>
<th>Purpose</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management Network</td>
<td>Used for internal communication between OpenStack Components</td>
<td>Reachable only within the data center</td>
</tr>
<tr>
<td>External Network</td>
<td>Used to provide VMs with Internet access</td>
<td>Reachable by anyone from the Internet</td>
</tr>
<tr>
<td>API Network</td>
<td>Exposes all OpenStack APIs, including the OpenStack Networking API, to tenants</td>
<td>Reachable to Tenants</td>
</tr>
<tr>
<td>Data Network</td>
<td>Used for VM data communication within the cloud deployment.</td>
<td>Reachable within the Tenant address space</td>
</tr>
</tbody>
</table>

- **Compute Node (s)**: Running Compute and Network agents
- **Controller Node (s)**: Running Database, Message Queue Server, API Services, Scheduler..
- **Network Node(s)**: Running Network Service Agents
- **Management Network**
- **Data Network**
- **API Network**
- **External Network**
- **Internet**
- **Router**
Neutron Overview

Logical Model

Tenant A Router

Subnet Red
Subnet Blue

VM 1
VM 2
VM 1

Network Node Node(s)

Namespace

vswitch

Compute Node

VM1

vswitch

Compute Node

VM2
VM1

Controller Node(s)

External Network

Data Network

API Network

Management Network

Internet

BRKDCN-2367 © 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public
OpenStack Neutron Architecture

- Core API
  - Network
  - Port
  - Subnet

- REST API
  - Neutron Server
    - Neutron Core plugins
      - ML2
      - Other vendor plugins
    - Neutron Service plugins
      - Load Balancer
      - Firewall
      - VPN
      - L3 Services
      - QoS
      - BGP
      - IPTables
      - Network Node
      - L2 Agent
      - vSwitch
      - HA Proxy
      - IPTables
      - StrongSwan
      - Namespace
      - LB/OVS/SR-IOV
      - Ryu

- Resource and Attribute Extension API
  - ProviderNetwork
  - PortBinding
  - Router
  - Quotas
  - SecurityGroups
  - AgentScheduler
  - LBaaS
  - FWaaS
  - VPNaaS

- Type Drivers
  - VLAN
  - GRE
  - VXLAN

- Mechanism Drivers
  - Cisco Nexus
  - OVS
  - OpenDayLight
  - APIC
  - More vendor drivers

- Message Queue
  - DHCP Agent
  - dnsmasq
  - L3 Agent
  - IPTables on Network Node
  - L2 Agent
  - vSwitch

- Core + Extension REST API’s
- Message Queue for communicating with Neutron Agents
- Core and Service Plugins
- Different vendor core plugins
- Different network technology support
- ML2 plugin with Type and Mechanism Drivers
- Service plugins with backend drivers
Neutron Networking for Layer 2 Tenant Isolation

Network Type
- Tenant Networks
- Admin Provider Networks
- Provisioned Externally

Network Segmentation Scheme for tenant isolation
- VLAN
- VXLAN
- GRE

Device implementing Network Segmentation Scheme
- vSwitch, ToR
- vSwitch
- ToR, Fabric
- vSwitch

Neutron driver
- Direct Device Configuration
- Device Configuration through Controller
Layer 2 Network Tenant Topologies

Host and Network based VLAN

Host based overlays

Network based overlays

Compute Node

VM3

VM1

vswitch

Data Network

Fabric Leaf, Top of Rack

Compute Node

VM4

VM2

vswitch

Data Network

Fabric Leaf, Top of Rack

Compute Node

VM3

VM1

vswitch

Data Network

Fabric Leaf, Top of Rack

Compute Node

VM4

VM2

vswitch

Data Network

Fabric Leaf, Top of Rack

Compute Node

VM3

VM1

vswitch

Data Network

Fabric Leaf, Top of Rack

Compute Node

VM4

VM2

vswitch

Data Network

Fabric Leaf, Top of Rack

VLAN Overlay

BRKDCN-2367 © 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public
Layer 2 Network Tenant Topologies – Design Considerations

• Number of Tenant Network Segments
• VLAN based tenant networks
  - Host (vSwitch v/s SR-IOV)
  - Host and Network
• VXLAN based tenant networks
  - Host (L2 population)
  - VXLAN offload - Network
  - Multicast v/s Controller (EVPN)
• Provider Networks
L2-Fabric Design
A Deeper View

• VPC between VIC and ToR
• vNICs with Bonds per network
• Trunks up each VPC
• Tenant VXLAN runs over common spanned VLAN (options for this)
• Vertical use cases may use VLANs, VLAN transparency, provider networks, etc..
Neutron Layer 2 Default Implementation

Network REST API requests

- Neutron Server
- Neutron Core plugin (ML2)
- Open vSwitch/Linux Bridge Mechanism Drivers

RPC message to agent on nodes

- Implements Neutron Core Resources
- Open vSwitch and Linux Bridge Mechanism Drivers
- Agents on Network and Compute Nodes
- Host based VLAN or Overlay (VXLAN, GRE) Type Drivers
Packet path animation for packet traveling from VM1 → VM3.
Neutron Routing Implementation

Routing REST API requests

- Neutron Server
- Neutron Service plugin (L3)
- Agent Scheduler

Picks a L3 agent on a Network Node

Neutron router HA capabilities using VRRP

- L3 agent on Network Nodes
- Default Gateway, Namespace and IPTables

Namespaces maps to a Neutron logical router. IPTables handle address translations

- Compute Nodes
- L3 traffic goes through Network node

VM VM
Layer 3 Tenant Network Topologies

Linux Host

- Network Node(s)
- Namespace
- vSwitch
- Data Network
- Fabric, Top of Rack

Service VMs

- Compute Node
- VM1
- VM2
- Service VMs
- vSwitch
- Data Network
- Fabric, Top of Rack

Fabric or Service Node

- Network Node(s)
- Compute Node
- VM1
- VM2
- Service VMs
- vSwitch
- Data Network
- Fabric, Service Node
Layer 3 Network Tenant Topologies – Design Considerations

• External connectivity for tenant networks
• Floating IPs
• L3 Traffic Pattern E-W and N-S Routing
• Central v/s Distributed Routing
• Host v/s Network based routing
L3-Fabric Design

• VPC between VIC and ToR
• vNICs with Bonds per network
• Trunks up each VPC
• Tenant VXLAN runs over common spanned VLAN (can use VLANs as complete replacement for VXLAN)
• L3 routed at Access/ToR
Packet path animation for packet traveling from VM1 → VM4
Packet path animation for packet traveling from VM1 → Internet
Cisco Integration into Neutron
Neutron Cisco Nexus Driver

Features

- Works with multiple Nexus platforms
- VLAN configuration
- VXLAN configuration
  - Nexus_VXLAN Type Driver
  - Multicast
  - VLAN to VNI association

Benefits

- No need trunk all tenant VLANs on compute node interfaces on ToR
- Dynamic provisioning/deprovisioning on ToR
- Network based overlays
Sample Nexus Mechanism Driver configuration for VXLAN

```
[ml2_type_vlan]
network_vlan_ranges = physnet1:10:500

[ml2_type_nexus_vxlan]
vni_ranges=50000:55000
mcast_ranges=225.1.1.1:225.1.1.2

[ml2_mech_cisco_nexus:192.168.1.1]
ComputeHostA=1/10
username=admin
password=secretPassword
ssh_port=22
physnet=physnet1
```
Neutron Cisco UCSM Driver (KVM)

Features:
- Nova and Neutron enhancements to support SR-IOV
- Supports VLAN configuration of SR-IOV ports (using port profiles) and vNIC ports (using Service Profiles)
- Enables configuration of VLAN profiles and automatic association with network ports

Benefits
- SR-IOV and non SR-IOV based UCS Fabric Interconnect configurations
Neutron DHCP Implementation

- Namespace and dnsmasq for every network
- Dnsmasq Reloads with every port add/delete
DHCP request/response animation for packet traveling from VM1 \(\rightarrow\) DHCP port.
Neutron DHCP Implementation with Cisco Prime Network Registrar (CPNR)

- DHCP configuration includes CPNR API end point configuration
- Mapping –
  - Network to Virtual Private Network (VPN)
  - Subnet to Scope
- Request and Responses handled using UDP ports
- **Benefits**
  - Relay is stateless and can be run in Active-Active
  - Highly Available CPNR Server for all tenants
Average time needed to receive a DHCP lease

Packet drop percentage
Issues in Neutron Reference L3 and ASR1K Solutions

- NAT for External Connectivity:
  - Issue - Scale limitation in Linux iptables software NAT
  - Solution - ASR1K can scale up to 4 million dynamic NAT entries and 16K static NAT entries

- Tenant Routing:
  - Issue - Scale limitations in Linux namespaces based software tenant networking
  - Solution - ASR1K uses Virtual Routing and Forwarding (VRF) instances for tenant routers. ASR1K can scale up to 4k VRFs (8k in upcoming release)

- Tenant Networks:
  - Issue - Scale limitations in Linux software based interfaces
  - Solution - ASR1K plugin maps tenant networks to sub-interfaces on ASR1K. ASR1K supports up to 64k sub-interfaces

- Data Throughput:
  - Issue - Performance limitations with software packet forwarding and NAT on generic compute hardware
  - Solution - ASR1K can perform packet forwarding and NAT at rates up to 230 Gbps
Neutron Cisco ASR1000 for Neutron L3 Service

- Mapping of Neutron reference L3 implementation -
  - Linux namespaces - ASR1K VRF
  - Internal Router ports – ASR1K VLAN or Port Channel sub interfaces
  - External Gateway ports – ASR1K VLAN or Port Channel sub interfaces
  - Linux IPTables – ASR1K NAT

- Benefits
  - Routing using physical infrastructure
  - Support for HSRP and Port Channel
OpenStack Neutron + Nexus + ASR: Physical Topology Example

Layer-3 Network Core

Nexus Layer-2 Fabric
Tenant VLANs and External Traffic

Nova Compute Nodes

OpenStack Controller
Neutron Server with Cisco Config Agent

ASR 1000 Routers

Management Network (NETCONF provisioning)
ML2 Nexus and ASR1K - East-West L3 (Routed) Traffic

ASR1K L3 Plugin

ML2 Nexus Driver

Neutron Host(s)

Nova Host

VM1

VM2

vSW

Nexus TOR

External Network

Controller Node(s)

Data Network (L3 routed)

Management Network

API Network

Router

Internet

Note: Packet animation included – VM1 → VM4
ASR1K L3 Plugin

Neutron Host(s)

ML2 Nexus Driver

ASR1K Virtual Router

VRF with default GW and NAT (to global routing).

Nexus TOR

VM1

vswitch

VM2

vswitch

VM3

vswitch

VM4

vswitch

VM5

vswitch

VM6

Data Network (L3 routed)

Controller Node(s)

External Network

API Network

Management Network

Internet

Note: Packet animation included – VM1 → Internet
ACI Integration with OpenStack

• Multiple OpenStack Driver Options:
  • APIC native Group-Based Policy
  • Neutron ML2

• APIC VMM Domain support for OpenStack
  • Endpoint statistics, health, faults in APIC

• Hypervisor local enforcement security policies
  • Security Groups (ML2 driver) via IP Tables
  • Group-Based Policies via OpenFlow in Open vSwitch

• Distributed NAT support on each compute node
  • Floating IP
  • sNAT (via hypervisor host IP)

• Distributed Neutron services per compute node
  • L3 / Anycast gateway, metadata, DHCP

• Multiple VRF support

• Support for VLAN / VXLAN to ACI fabric
Neutron Cisco Application Policy Infrastructure Controller (APIC) Driver

- Neutron API: Network, Router, Subnet, Security Group
- L2 / L3 enforced in fabric, security groups enforced on hypervisor

OpenStack Deployment Overview - High Availability
High Availability Decisions

• Touches every service and every component of the design – Network, Storage, Compute, Logging, etc..

• Network L3 HA – Built-in L3 HA with Keepalived/HAProx, external L3 HA (Agg Layer, ASR1K Plugin)

• Pick your release – Major changes in HA across all parts of OpenStack have progressed on each release

• Many components are:
  • Databases: Options include MySQL-WSREP and Galera
  • Message Queue: RabbitMQ Clustering and RabbitMQ Mirrored Queues
  • API/Web services: HAProx, Keepalived, traditional SLB
  • Swift proxy nodes: HAProx, 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
Tenant L3 Network High-Availability (HA)

- Helps resolve issue of single tenant router going down and isolating the tenant instances
- Can be configured manually via neutron client by an Admin:
  - ‘neutron router-create --ha True|False’
- Or set as a system default within the neutron.conf/l3_agent.ini files
- Uses Keepalived for VRRP between L3 agents
- Existing non-HA enabled routers can be updated to HA:
  - ‘neutron router-update <router-name> --ha=True’
- Requires a minimum of two network nodes (or controllers) that have L3 agents running
L3 HA – Tenant View

• Tenant sees one router with a single gateway IP address

• non-Admin users cannot control if the router is HA or non-HA

• From the tenant’s perspective the router behaves the same in HA or non-HA mode
L3 HA – Routing View

- Tenant network has 10.10.30.x/24 assigned
- VRRP is using 169.254.xx over a dedicated HA-only network that traverse the same tenant network type
- Router (L3 agent) on the left is the VRRP master and is the tenant GW (10.10.30.1)
L3 HA – Traffic Flow

Compute Node

Network Node 1

Network Node 2

Management/Underlay Network

Public Network

VRRPv2
169.254.192.x >> 224.0.0.18

BRKDCN-2367 © 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public
Testing a failure

Check who is master:

[root@net1 ~]# cat /var/lib/neutron/ha_confs/719b853f-539e-420b-a76b-0440146f05de/state
master
[root@net2 ~]# cat /var/lib/neutron/ha_confs/719b853f-539e-420b-a76b-0440146f05de/state
backup

Simulate a failure by shutting down the HA interface (remember this was in the ‘track’ list):

[root@net1 ~]# ip netns exec qrouter-719b853f-539e-420b-a76b-0440146f05de ifconfig ha-0d655b16-c6 down

Check that VRRP switched to the other node as master:

[root@net1 ~]# cat /var/lib/neutron/ha_confs/719b853f-539e-420b-a76b-0440146f05de/state
fault
[root@net2 ~]# cat /var/lib/neutron/ha_confs/719b853f-539e-420b-a76b-0440146f05de/state
master

Ping – the ultimate test of HA 😊:

ubuntu@server1:~$ ping 8.8.8.8
64 bytes from 8.8.8.8: icmp_seq=20 ttl=127 time=65.4 ms
64 bytes from 8.8.8.8: icmp_seq=21 ttl=127 time=107 ms
64 bytes from 8.8.8.8: icmp_seq=22 ttl=127 time=64.5 ms
64 bytes from 8.8.8.8: icmp_seq=23 ttl=127 time=67.6 ms

Increased delay but no loss
High-Availability/Load-Balancing of API Services

- Load-balancers are used to front-end OpenStack API services
- HAProxy and Keepalived are commonly used together to create an HA/Load-balanced design for these services
- HAProxy/Keepalived can be used as native Linux services or as containers
Advanced Neutron Considerations
IPv4 No NAT – Address Scope

- [http://www.debug-all.com/?p=197](http://www.debug-all.com/?p=197)

- A complex and convoluted way of disabling IPv4 NAT in Neutron

- Requirements and limitations
  - Address Scope + Subnet Pool support = very crude IPv4 No NAT deployment
  - Requires Address Scopes and Subnet Pool use
  - You cannot manually assign a subnet allocation pool range OR a custom gateway – This is a major issue when you need to define a custom upstream gateway address (I worked around this by editing the subnet pool after creation)
  - You have to sub-allocate prefixes/CIDR ranges out of the subnet pool which is typically something the cloud/network operator would do upstream at the DC Agg Layer, routed access or border leaf
The Hard Stuff – IPv6 + Cloud

- Inside of a 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

- It has been a bumpy road to getting a solid IPv6 implementation in OpenStack

- Most of the core IPv6 requirements are met except for IPv6 PD HA and IPv6-only Metadata (config-drive seems to be good enough)

- Tenant IPv6 Address Assignment via:
  - SLAAC, Stateful DHCPv6, Stateless DHCPv6
  - ipv6_ra_mode attribute - Control of router advertisements for a subnet
  - ipv6_address_mode attribute - Control of how addressing is handled by OpenStack

- 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)
### Cloud Stack – IP Version Options

#### 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>VM Operating System</td>
</tr>
<tr>
<td>IPv4/IPv6 Database(s)</td>
<td>Virtual Networking (L2/L3)</td>
</tr>
<tr>
<td>IPv4/IPv6 Automation</td>
<td>Virtual Network Services (SLB/FW)</td>
</tr>
<tr>
<td>IPv4/IPv6 Interface (GUI, CLI)</td>
<td>Tenant Interface (GUI, CLI)</td>
</tr>
</tbody>
</table>

#### Conditional Dual-Stack

<table>
<thead>
<tr>
<th>Service Tier</th>
<th>Tenant 1 Access Tier</th>
<th>Tenant 2 Access Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 API endpoints</td>
<td>IPv4/IPv6 VM Operating System</td>
<td>IPv6 VM Operating System</td>
</tr>
<tr>
<td>IPv4 Database(s)</td>
<td>IPv4/IPv6 Virtual Networking (L2/L3)</td>
<td>IPv6 Virtual Networking (L2/L3)</td>
</tr>
<tr>
<td>IPv4 Automation</td>
<td>IPv4/IPv6 Interface (GUI, CLI)</td>
<td>IPv6 Interface (GUI, CLI)</td>
</tr>
<tr>
<td>IPv4/IPv6 Virtual Network Services (SLB/FW)</td>
<td>IPv4/IPv6 Tenant Interface (GUI, CLI)</td>
<td>IPv6 Tenant Interface (GUI, CLI)</td>
</tr>
</tbody>
</table>
Tenant IPv6 Address Options

Option 1
Cloud Provider-assigned Addressing

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

Option 2
Tenant Brings Addressing

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

Option 3
Prefix Translation

Don’t do this
Network Function Virtualization

Admin provisioned Service

Tenant provisioned Service
Neutron and NFV

• Issue
  • Anti-spoofing rules to ensure traffic originates and terminates as expected
  • Doesn’t work for NFV VNF use cases

• Solution
  • Added Port Security Extension
    • Adds new “Port Security enabled” attribute to Network and Port Resources
    • Only tenant owner can set this attribute on the resources
    • Security Group and Allowed Address Pair are not allowed to be set

• Issue
  • VXLAN for tenant isolation and VLAN for app traffic isolation within the tenant
  • No means to identify VLAN transparent networks

• Solution
  • Added Network Resource Extension
    • Adds new “VLAN Transparent” attribute to Network Resource
    • Only tenant owner can set this attribute on the resources
    • No firewalling on VLAN tagged packets
Conclusion

• OpenStack is for real and maturing at a rapid pace
• Align yourself with market leaders who have strong partnerships
• There are multiple options for installation
• Keep your design simple and reduce/remove technical debt wherever possible – it can kill your upgrades and scale out plans
• Get involved in the community – open source enjoys the major advantage of feature velocity
Call to Action

• Breakout/Labs sessions:
  • OpenStack Deployment for Enterprise and Service Provider (BRKDCN-2367)
  • OpenStack Integration with Cisco ACI (LTRACI-2225)
  • Mastering OpenStack and ACI (BRKACI-3456)
  • Cisco Virtualized Packet Core on OpenStack (LABSPM-2013)
  • Programmable VXLAN Fabric using VTS with OpenStack/VMware Integration (LTRDCN-2001)

• DevNet zone related sessions:
  • Getting Started with OpenStack (DEVNET-1101)
  • OpenStack Enabling DevOps (DEVNET-1104)
  • Workshop - Getting Started with OpenStack (DEVNET-1211)
  • OpenStack with OpenDaylight (DEVNET-2041)
  • Best REST in OpenStack (DEVNET-2004)
  • Cisco VIM for OpenStack based NFVI Solution (DEVNET-2570)
Complete Your Online Session Evaluation

• Give us your feedback to be entered into a Daily Survey Drawing. A daily winner will receive a $750 gift card.

• Complete your session surveys through the Cisco Live mobile app or on www.CiscoLive.com/us.

Don’t forget: Cisco Live sessions will be available for viewing on demand after the event at www.CiscoLive.com/Online.
Continue Your Education

- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Lunch & Learn
- Meet the Engineer 1:1 meetings
- Related sessions
## Cloud Cisco Education Offerings

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
<th>Cisco Certification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding Cloud Fundamentals (CLDFND)</td>
<td>Learn how to perform foundational tasks related to Cloud computing, and the essentials of Cloud infrastructure, administration and operations</td>
<td>CCNA Cloud</td>
</tr>
<tr>
<td>Introducing Cloud Administration (CLDADM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementing and Troubleshooting the Cisco Cloud Infrastructure (CLDINF); Designing the Cisco Cloud (CLDDES); Automating the Cisco Enterprise Cloud (CLDAUT); Building the Cisco Cloud with Application Centric Infrastructure (CLDACI)</td>
<td>Obtain professional level skills to design, automate, secure, provision and manage private and hybrid Clouds</td>
<td>CCNP Cloud</td>
</tr>
<tr>
<td>Product Training Portfolio: UCS Director: UCSDF, UCSDACI Prime Service Catalog: PSCF, PSCI, PSCD MetaPod: MPODF20</td>
<td>Gain in-depth hands-on skills using Cisco solutions to configure, deploy, manage and troubleshoot Cloud deployments</td>
<td></td>
</tr>
</tbody>
</table>

For more details, please visit: [http://learningnetwork.cisco.com](http://learningnetwork.cisco.com)

Questions? Visit the Learning@Cisco Booth
Thank you
Reference Slides
IPv6
# Neutron Addressing Schemes

<table>
<thead>
<tr>
<th>ipv6_ra_mode</th>
<th>ipv6_address_mode</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLAAC</td>
<td>N/S</td>
<td>Address using Neutron router</td>
</tr>
<tr>
<td>N/S</td>
<td>SLAAC</td>
<td>Address using external router</td>
</tr>
<tr>
<td>SLAAC</td>
<td>SLAAC</td>
<td>Address using Neutron router</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ipv6_ra_mode</th>
<th>ipv6_address_mode</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCPv6-stateless</td>
<td>N/S</td>
<td>Address using Neutron router and optional information using external service</td>
</tr>
<tr>
<td>N/S</td>
<td>DHCPv6-stateless</td>
<td>Address using external router and optional information using Neutron DHCP implementation</td>
</tr>
<tr>
<td>DHCPv6-stateless</td>
<td>DHCPv6-stateless</td>
<td>Address and optional information using Neutron router and DHCP implementation respectively</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ipv6_ra_mode</th>
<th>ipv6_address_mode</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCPv6-stateful</td>
<td>N/S</td>
<td>Address and optional information using external service</td>
</tr>
<tr>
<td>N/S</td>
<td>DHCPv6-stateful</td>
<td>Address and optional information using Neutron DHCP implementation</td>
</tr>
<tr>
<td>DHCPv6-stateful</td>
<td>DHCPv6-stateful</td>
<td>Address and optional information using Neutron DHCP implementation</td>
</tr>
</tbody>
</table>

## Reference

http://docs.openstack.org/mitaka/networking-guide/config-ipv6.html
Tenant IPv6 - Neutron L3 Example

- VLAN10: 2001:db8:cafe:a::1/64
- VLAN12: 2001:db8:cafe:d::1/64

VLANs are trunked to Agg Layer


ToR

Management Network: 172.16.21.0/24


all-in-one

Instances/VMs

Private Networks

Per-Tenant Neutron Router
Create the Public Network/Subnet

```
neutron net-create public --router:external

neutron subnet-create --name public-subnet --allocation-pool start=172.16.12.5, end=172.16.12.254 public 172.16.12.0/24

```
### SLAAC Mode

**neutron net-create private**

**neutron subnet-create --ip-version=6 --name=private_v6_subnet --ipv6-address-mode=slaac --ipv6-ra-mode=slaac** private 2001:db8:cafe::/64

+------------------------------------------------------------------------------------------------------------------+
| Field             | Value                                                                                                                                 |
+------------------------------------------------------------------------------------------------------------------+
| allocation_pools  | 
| cidr              | 2001:db8:cafe::/64                                                                                                                |
| dns_nameservers   | True                                                                                                                                 |
| enable_dhcp       | True                                                                                                                                 |
| gateway_ip        | 2001:db8:cafe::1                                                                                                                  |
| host_routes       |                                                                                                                                  |
| id                | 42cc3dbc-938b-4ad6-b12e-59aeef7618477                                                                                              |
| ip_version        | 6                                                                                                                                    |
| ipv6_address_mode | slaac                                                                                                                               |
| ipv6_ra_mode      | slaac                                                                                                                               |
| name              | private_v6_subnet                                                                                                                 |
| network_id        | 7166ce15-c581-4195-9479-ad283193d06                                                                                               |
| subnetpool_id     |                                                                                                                                      |
| tenant_id         | f057804eb39b4618b40e06196e16265b                                                                                                   |
+------------------------------------------------------------------------------------------------------------------+

**IPv4: 172.16.12.0/24**

**IPv6: 2001:db8:cafe::/64**

---

**IPv4: 10.0.0.0/24**

**IPv6: 2001:db8:cafe::/64**

---

**IPv4: 10.0.0.9**

**IPv6: 2001:db8:cafe:0:e816:3eff:fe79:5acc**

---

**DC**

**DNS**

**Instance**

**Router**

---

**2001:db8:cafe:a::e**

---

**Reference**

BRKDCN-2367 © 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public 107
Router Example

neutron router-create private-router
neutron router-gateway-set private-router public
neutron router-interface-add private-router private-v4-subnet
neutron router-interface-add private-router private-v6-subnet
SLAAC Mode Info

• OpenStack will not inject the IPv6 DNS entry from the subnet dns_nameservers entry

• Options
  • Manually setting the IPv6 DNS server entry in the resolv.conf file allows for correct IPv6-based name resolution
  • Bake DNS settings into your image
  • Cloud-init to inject the DNS configuration

• You do get A and AAAA records back over IPv4 transport

• Basically, it works as it should
SLAAC Mode – Sniffer Capture

15:08:01.520353 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 16) fe80::f816:3eff:fe79:5acc > ff02::2: [icmp6 sum ok] ICMP6, router solicitation, length 16
   source link-address option (1), length 8 (1): fa:16:3e:79:5a:cc
   0x0000: fa16 3e79 5acc
15:08:01.520667 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 56) fe80::f816:3eff:fe79:5acc > ff02::1: [icmp6 sum ok] ICMP6, router advertisement, length 56
   hop limit 64, Flags [none], pref medium, router lifetime 30s, reachable time 0s, retrans time 0s
   prefix info option (3), length 32 (4): 2001:db8:cafe::/64, Flags [onlink, auto], valid time 86400s, pref. time 14400s
   0x0000: 40c0 0001 5180 0000 3840 0000 0000 2001
   0x0010: 0db8 cafe 0000 0000 0000 0000 0000 0000
   source link-address option (1), length 8 (1): fa:16:3e:17:b4
   0x0000: fa16 3ec3 17b4
15:08:01.526004 IP6 (hlim 1, next-header Options (0) payload length: 36) fe80::f816:3eff:fe79:5acc > ff02::16: HBH (rtalert: 0x00000) (padn) [icmp6 sum ok] ICMP6, multicast listener report v2, 1 group record(s) [gaddr ff02::1:ff79:5acc is_ex { }]
15:08:02.484047 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 24) :: > ff02::1:ff79:5acc: [icmp6 sum ok] ICMP6, neighbor solicitation, length 24, who has 2001:db8:cafe:0:f816:3eff:fe79:5acc
Stateless DHCPv6 Mode

```
neutron net-create private-dhcpv6-stateless

neutron subnet-create --ip-version=6 --name=private_dhcpv6_stateless_subnet
--ipv6-address-mode=dhcpv6-stateless --ipv6-ra-mode=dhcpv6-stateless private-dhcpv6-stateless

+---------------+----------------------------------|
| Field          | Value                            |
+---------------+----------------------------------|
| cidr           | 2001:db8:cafe:2::/64             |
| dns_nameservers | 2001:db8:cafe:a::e               |
| enable_dhcp    | True                             |
| gateway_ip     | 2001:db8:cafe:2::1               |
| host_routes    |                                  |
| id             | e63e72d5-493a-4a49-8f7d-8846c2bc7a8f |
| ip_version     | 6                                |
| ipv6_address_mode | dhcpv6-stateless               |
| ipv6_ra_mode   | dhcpv6-stateless                 |
| name           | private_dhcpv6_stateless_subnet  |
| network_id     | 27618d5e-318c-46a4-b6a2-a155beed9643 |
| subnetpool_id  |                                  |
| tenant_id      | f057804eb39b4618b40e06196e16265b |
+---------------+----------------------------------|
```


Router

DC

DNS

Instance

IPv4: 10.0.2.0/24 IPv6: 2001:db8:cafe:2::/64


2001:db8:cafe:a::e

Reference

BRKDCN-2367 © 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public 111
DHCPv6 Stateless Mode Info

- Enable client for DHCPv6 Stateless:

  Ubuntu
  
  /etc/network/interfaces
  
  auto eth0
  iface eth0 inet dhcp
  iface eth0 inet6 auto
dhcp 1

  CentOS/RHEL/Fedora
  
  /etc/sysconfig/network-scripts/ifcfg-xxxx
  
  IPV6INIT="yes"
  DHCPV6C="yes"
  DHCPV6C_OPTIONS="-S"

- Then you get addressing and options:

  ubuntu@dhcpv6-stateless-4:$ more /etc/resolv.conf
  
  # Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
  # DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
  nameserver 172.16.10.14
  nameserver 2001:db8:cafe:a::e
  search openstacklocal
DHCPv6 Stateless Mode – Sniffer Capture

15:43:23.911172 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 56) fe80::f816:3eff:fecl:bc52 > ff02::1: [icmp6 sum ok] ICMP6, router advertisement, length 56
   hop limit 64, Flags [other stateful], pref medium, router lifetime 30s, reachable time 0s, retrans time 0s
   prefix info option (3), length 32 (4): 2001:db8:cafe:2::/64, Flags [onlink, auto], valid time 86400s, pref. time 14400s
     0x0000:  40c0 0001 5180 0000 3840 0000 0000 2001
     0x0010:  0db8 cafe 0002 0000 0000 0000 0000 0000
   source link-address option (1), length 8 (1): fa16:3e:c1:bc:52

15:43:25.353331 IP6 (hlim 1, next-header UDP (17) payload length: 44) fe80::f816:3eff:fefe:d157.546 > ff02::1:2.547: [udp sum ok] dhcp6 inf-req
   (xid=d2dbc8 (client-ID hwaddr type 1 fa163efed157) (option-request DNS-server DNS-search-list Client-FQDN SNTP-servers) (elapsed-time 94))

15:43:25.353578 IP6 (class 0xc0, hlim 64, next-header UDP (17) payload length: 88) fe80::f816:3eff:fe2d:a6de.546 > fe80::f816:3eff:fefe:d157.546: [udp sum ok] dhcp6 reply
   (xid=d2dbc8 (client-ID hwaddr type 1 fa163efed157) (server-ID hwaddr type 1 fa163e2da6de) (DNS-search-list openstacklocal.) (DNS-server 2001:db8:cafe:a::e) (lifetime 86400))
Stateful DHCPv6 Mode

neutron net-create private-dhcpv6

eutron subnet-create --ip-version=6 --name=private_dhcpv6_subnet --ipv6-address-mode=dhcpv6-stateful --ipv6-ra-mode=dhcpv6-stateful private-dhcpv6 2001:db8::1/64 --dns-nameserver 2001:db8:cafe:a:e

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cidr</td>
<td>2001:db8:cafe::1/64</td>
</tr>
<tr>
<td>dns_nameservers</td>
<td>2001:db8:cafe:a::e</td>
</tr>
<tr>
<td>enable_dhcp</td>
<td>True</td>
</tr>
<tr>
<td>gateway_ip</td>
<td>2001:db8:cafe:e:1::1</td>
</tr>
<tr>
<td>host_routes</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>545ea206-9d14-4dca-8bae-7940719bdab5</td>
</tr>
<tr>
<td>ip_version</td>
<td>6</td>
</tr>
<tr>
<td>ipv6_address_mode</td>
<td>dhcpv6-stateful</td>
</tr>
<tr>
<td>ipv6_ra_mode</td>
<td>dhcpv6-stateful</td>
</tr>
<tr>
<td>name</td>
<td>private_dhcpv6_subnet</td>
</tr>
<tr>
<td>network_id</td>
<td>55ed8333-2876-400a-92c1-ef49bc10aa2b</td>
</tr>
<tr>
<td>subnetpool_id</td>
<td></td>
</tr>
<tr>
<td>tenant_id</td>
<td>f057804eb39b4618b40e06196e16265b</td>
</tr>
</tbody>
</table>


DHCPv6 Stateful Mode Info

- Enable client for DHCPv6:
  
  **Ubuntu**
  
  ```
  /etc/network/interfaces
  auto eth0
  iface eth0 inet dhcp
  iface eth0 inet6 dhcp
  ```

  **CentOS/RHEL/Fedora**
  
  ```
  /etc/sysconfig/network-scripts/ifcfg-xxxx
  IPV6INIT="yes"
  DHCPV6C="yes"
  ```

- Then you get addressing and options:
  
  ```
  ubuntu@dhcpv6-1:~$ more /etc/resolv.conf
  # Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
  #     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
  nameserver 172.16.10.14
  nameserver 2001:db8:cafe:a::e
  search openstacklocal
  ```
DHCPv6 Stateful Mode – Sniffer Capture
Address Assignment: Provider Networks
Provider Networks

DNS Server:
2001:db8:cafe:a::e

Agg Layer

VLAN21:
2001:db8:cafe:15::1/64

VLAN22:
2001:db8:cafe:16::1/64

VLANs are trunked to
Agg Layer

ToR

ToR

VPC

Cisco VIC

eth4/5 = bond0  # Management/API
eth6/7 = bond1  # Provider Networks
bond0 = 172.16.21.17
bond0 = 2001:db8:cafe:15:f2f7:55ff:feaa:c2e3
bond1 = VLAN trunk (example is VLAN22)
bond1 = VLAN22:172.16.22.0/24
bond1 = VLAN22.2001:DB8:CAFE:16::1/64

All-in-one
c7-os-1
Provider Network - Examples


neutron subnet-create external-net 172.16.22.0/24 --name external-subnet --gateway 172.16.22.1 --allocation-pool start=172.16.22.5,end=172.16.22.254

# SLAAC

# Stateless DHCPv6

# Stateful DHCPv6
# SLAAC
interface Vlan22
description Provider Network trunked for C7-os-1
ip address 172.16.22.2 255.255.255.0
ipv6 address 2001:DB8:CAFE:16::1/64
standby version 2
standby 2 ipv6 autoconfig
standby 2 timers msec 250 msec 750
standby 2 priority 110
standby 2 preempt
standby 2 authentication OPEN

# Stateless DHCPv6
interface Vlan22
description Provider Network trunked for C7-os-1
ip address 172.16.22.2 255.255.255.0
ipv6 address 2001:DB8:CAFE:16::1/64
ipv6 nd other-config-flag
standby version 2
standby 2 ipv6 autoconfig
standby 2 timers msec 250 msec 750
standby 2 priority 110
standby 2 preempt
standby 2 authentication OPEN

# Stateful DHCPv6
interface Vlan22
description Provider Network trunked for C7-os-1
ip address 172.16.22.2 255.255.255.0
ipv6 address 2001:DB8:CAFE:16::1/64
ipv6 nd managed-config-flag
standby version 2
standby 2 ipv6 autoconfig
standby 2 timers msec 250 msec 750
standby 2 priority 110
standby 2 preempt
standby 2 authentication OPEN
IPv6 Only
IPv6-Only Instances

- IPv6-only works out-of-the-box for IP connectivity but you are hosed on metadata
- It is a real drag to have to deploy an IPv4 subnet along with IPv6 just to get basic functionality working such as FQDN, SSH keys and other metadata
- The metadata service ONLY supports IPv4
- A wish list bug to work on this issue expired: https://bugs.launchpad.net/neutron/+bug/1460177 😞
- Workarounds:
  - Build all/most of what you want in the image itself
  - Use config-drive
Basic IPv6-only Config-Drive Example

```
[root@c7-os-1 latest]# cat user_data
#cloud-config
fqdn: v6onlyinstance.example.com
users:
  - name: cloud-user
    ssh-authorized-keys:
      - ssh-rsa
AAAAB3NzaC1yc2EAAAADQABAAABAAQCC4W4RPlOiBjY14iJwW9kd3Chys5bUBjy2VKJKFa5az8JHcVVo3LO5BHdhc6WryT+b1mX9LKGyVSc0rfzSEAfQ91dXJChul15BNk9pLibs3oe8s/1r/vjtxQcpKB1GIJG3PyiuvspVLVe1KrhbdIDLvuZCq824LVPuAozLqFatshu/Y21U5WyiTi15VJjwbbzc67BFHz2ov2bBGcFfWFyUQMkiiyIrAv5hVcqAD7XaQ7P5JaOaHAcNcFmY8RbtE
MSIyw8feleyrY42ikMNAn/eU5Zmc1819R4CwI9wYqYdpVyiNULRWH9pK30dqhhtgElzCax+WqmM6GXP root@c7-os-1.example.com

[root@c7-os-1 ~]# nova boot --flavor m1.small --image rh7-stateless --key-name new-aio-key --security-groups default --nic net-name=external-net rhv6-only-drive --config-drive true --user-data user_data.yaml

[root@c7-os-1 ~]# nova list
+--------------------------------------+-----------------+--------+---------------+-------------------+-----------------
| ID                                   | Name            | Status | Task State   | Power State       | Networks        |
+--------------------------------------+-----------------+--------+---------------+-------------------+-----------------|
| 2244a346-a34b-4ab6-905f-71dc207a92e6 | rhv6-only-drive | ACTIVE | Running       | external-net=2001:db8:cafe:16:f816:3eff:feec:3c59 |
+--------------------------------------+-----------------+--------+---------------+-------------------+-----------------|

[root@c7-os-1 ~]# ssh cloud-user@2001:db8:cafe:16:f816:3eff:feec:3c59

[root@cloud-user@v6onlyinstance ~]# ip a
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
 link/ether fa:16:3e:ec:3c:59 brd ff:ff:ff:ff:ff:ff
 inet6 2001:db8:cafe:16:f816:3eff:feec:3c59/64 scope global dynamic
   valid_lft 2591952sec preferred_lft 604752sec
   inet6 fe80::f816:3eff:feec:3c59/64 scope link
     valid_lft forever preferred_lft forever

[root@cloud-user@v6onlyinstance ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search openstacklocal.example.com
nameserver 2001:db8:cafe:a::e
```

```
IPv6 Prefix Delegation
IPv6 PD in OpenStack

• Super important feature in OpenStack:
  • No IPv6 NAT support in OpenStack (Yee Haw!)
  • Can assign prefixes to tenants via:
    • Manual assignment (“Sally, here is your prefix”) - HORRIBLE method due to human errors in tracking and assignment (fat finger syndrome)
    • IPv6 Subnet Pool Support - Pool of addresses that tenants can pull from - http://docs.openstack.org/newton/networking-guide/config-subnet-pools.html
      • No overlap between two subnets
      • Stable address management across projects
      • Leverages default IPv6 subnet pool support

• Does not support HA of the PD agent (future work)
IPv6 Prefix Delegation
RFC 3633 & 3769

- IPv6 PD was originally designed to allow a downstream router (Requesting router) to request a prefix from an upstream router (Delegating router) and use the assigned IPv6 prefix for the subscriber-side networks.

- Basically, it is for routers asking other routers/relays/servers for a whole prefix that can be used to ‘seed’ downstream networks with an IPv6 prefix of their own.

Sure, take 2001:db8:bad:face::/64

Sure, use prefix: 2001:db8:bad:face::/64

Can I have a prefix?

2001:db8:bad:face::f816:3eff:fe4f:a2cd/64

2001:db8:bad:face::/64

2001:db8:bad:face::/127

2001:db8:bad:face::/64

Can I have a prefix?
IPv6 PD - Example Topology

- OpenStack All-in-One host is connected to an L2-switch
- Dibbler server connected to the L2-switch
- Pre-defined IPv6 prefix on “public network”
- We need an IPv6 prefix assigned to the tenant interface of the Neutron router
IPv6 PD - Configuration

/etc/dibbler/server.conf

```bash
pd-class {
    pd-pool 2001:db8:face::/48
    pd-length 64
}
```

```
neutron net-create public --provider:network_type flat --provider:physical_network public --router:external

neutron subnet-create public --ip-version 6 --name public-v6-subnet 2001:db8:bad:cafe::/64

neutron router-create pd-rtr

neutron router-gateway-set pd-rtr public

neutron net-create ipv6-pd

neutron subnet-create ipv6-pd --name ipv6-pd-1 --ip_version 6 --ipv6_ra_mode slaac --ipv6_address_mode slaac --use-default-subnetpool

# Look for “subnetpool_id | prefix_delegation” in the output

neutron router-interface-add pd-rtr ipv6-pd-1
```

/etc/neutron/neutron.conf

```
ipv6_pd_enabled = True
```

Magic happens here
Logs and tcpdumps and stuff

Neutron L3 Log

2016-10-17 15:03:46.822 DEBUG neutron.agent.linux.dibbler [-] Enable IPv6 PD for router 561ed48c-182c-4073-b157-77130280d5b9 subnet 3bc82226-816f-4d71-983e-7429d3d5475a ri_ifname qr-98120bddd1 from (pid=56056) enable /opt/stack/neutron/neutron/agent/linux/dibbler.py:123


2016-10-17 15:03:46.847 DEBUG neutron.agent.linux.dibbler [-] dibbler client enabled for router 561ed48c-182c-4073-b157-77130280d5b9 subnet 3bc82226-816f-4d71-983e-7429d3d5475a ri_ifname qr-98120bddd1 from (pid=56056) enable /opt/stack/neutron/neutron/agent/linux/dibbler.py:129

15:03:46.852214 IP6 (flowlabel 0x7bf3b, hlim 1, next-header UDP (17) payload length: 60) fe80::f816:3eff:ff:cc00::dchpv6-client > ff02::1:2. dchpv6-server: [udp sum ok] dchp6 solicit (xid=800a54 (client-ID vid 000022b83bc82226) (IA_PD IAIID:1 T1:4294967295 T2:4294967295) (elapsed-time 0))

15:03:46.853654 IP6 (flowlabel 0x80c10, hlim 64, next-header UDP (17) payload length: 134) fe80::20c:29ff:fe87:2f6b. dhcpv6-server > fe80::f816:3eff:ff:cc00::dchpv6-client: [udp sum ok] dchp6 advertise (xid=800a54 (IA_PD IAIID:1 T1:2000 T2:3000 (IA_PD-prefix 2001:db8::2ff2::/64 pltime:3600 vtime:7200) (status-code success)) (server-ID hwaddr/time type 1 time 529454623 00c29872f6b) (client-ID vid 000022b83bc82226) (preference 0))

15:03:47.955793 IP6 (flowlabel 0x7bf3b, hlim 1, next-header UDP (17) payload length: 107) fe80::f816:3eff:ff:cc00::dchpv6-client > ff02::1:2. dchpv6-server: [udp sum ok] dchp6 request (xid=561e28 (client-ID vid 000022b83bc8226) (IA_PD IAIID:1 T1:4294967295 T2:4294967295 (IA_PD-prefix 2001:db8::2ff2::/64 pltime:3600 vtime:7200) (server-ID hwaddr/time type 1 time 529454623 00c29872f6b) (elapsed-time 0))

15:03:47.956239 IP6 (flowlabel 0x80c10, hlim 64, next-header UDP (17) payload length: 134) fe80::f816:3eff:ff:cc00::dchpv6-server > fe80::f816:3eff:ff:cc00::dchpv6-client: [udp sum ok] dchp6 reply (xid=561e28 (IA_PD IAIID:1 T1:2000 T2:3000 (IA_PD-prefix 2001:db8::2ff2::/64 pltime:3600 vtime:7200) (status-code success)) (server-ID hwaddr/time type 1 time 529454623 00c29872f6b) (client-ID vid 000022b83bc82226) (preference 0))
Dibbler Server Log

03:46 Server Notice  Received SOLICIT on br-ex/8, trans-id=0x800a54, 3 opts: 1 25 8 (non-relayed)
. . .

03:46 Server Debug  PD: Adding PD (iaid=1) to addrDB.
03:46 Server Debug  PD: Adding 2001:db8:face:2ff2:: prefix to PD (iaid=1) to addrDB.
. . .

03:46 Server Notice  Sending ADVERTISE on br-ex/8, transID=0x800a54, opts: 25 2 1 7, 0 relay(s).
. . .

03:47 Server Notice  Received REQUEST on br-ex/8, trans-id=0x561e28, 4 opts: 1 25 2 8 (non-relayed)
. . .

03:47 Server Debug  Checking prefix 2001:db8:face:2ff2:: against reservations ...
03:47 Server Debug  PD: Requested prefix (2001:db8:face:2ff2::) is free, great!
03:47 Server Debug  PD: Adding PD (iaid=1) to addrDB.
03:47 Server Debug  PD: Adding 2001:db8:face:2ff2:: prefix to PD (iaid=1) to addrDB.
03:47 Server Debug  PD: Prefix usage for class 0 increased to 2.
03:47 Server Info  PD: assigned prefix(es):2001:db8:face:2ff2::/64

Output truncated for clarity
IPv6 with Heat
Orchestrating IPv6 Deployment with Heat

- Heat with IPv6 uses similar parameters and resource conventions as with IPv4

<table>
<thead>
<tr>
<th>IPv4</th>
<th>IPv6</th>
</tr>
</thead>
</table>
| **private_net_cidr:**  
  type: string  
  description: Tenant IPv4 network address (CIDR notation)  
  default: 10.10.30.0/24 | **private_net_v6:**  
  type: string  
  description: Tenant IPv6 subnet address  
  default: 2001:db8:cafe:1e::/64 |

- [https://github.com/shmcfarl/my-heat-templates](https://github.com/shmcfarl/my-heat-templates)
IPv6-only Example

Parameters

- **private_net_v6**: type: string
description: Private IPv6 subnet address
default: 2001:db8:cafe:1e::/64

- **private_net_v6_gateway**: type: string
description: Private network gateway address
default: 2001:db8:cafe:1e::1

- **private_net_v6_pool_start**: type: string
description: Start of private network IP address allocation pool
default: 2001:db8:cafe:1e::2

- **private_net_v6_pool_end**: type: string
description: End of private network IP address allocation pool

Resources

- **router_interface_v6**: type: OS::Neutron::RouterInterface
properties:
  router: { get_resource: router }
  subnet: { get_resource: private_v6_subnet }

- **server_security_group**: type: OS::Neutron::SecurityGroup
  properties:
    description: Heat-deployed security group.
    name: heat-security-group
    rules:
      - remote_ip_prefix: "::/0",
        ethertype: IPv6,
        protocol: tcp,
        port_range_min: 22,
        port_range_max: 22
      - remote_ip_prefix: "::/0",
        ethertype: IPv6,
        protocol: icmp
      - remote_ip_prefix: "::/0",
        ethertype: IPv6,
        protocol: tcp,
        port_range_min: 80,
        port_range_max: 80

https://github.com/shmcfarl/my-heat-templates/blob/master/v6-only-SLAAC.yaml
IPv6 with L3 High-Availability
L3 HA – Tenant View

• Tenant sees one router with a single gateway IP address
• non-Admin users cannot control if the router is HA or non-HA
• From the tenant’s perspective the router behaves the same in HA or non-HA mode
L3 HA – Routing View

- Tenant network has 2001:db8:cafe:beef::/64 prefix assigned
- VRRP is using a dedicated HA-only network that traverses the same tenant network type
- Router (L3 agent) on the left is the VRRP master and is the tenant IPv6 GW (fe80::f816:3eff:fee7:f435)
```
# cat /var/lib/neutron/ha_confs/0772d696-ec0f-46f3-b7d0-a984612fcdca/keepalived.conf

vrrp_instance VR_1 {
    state BACKUP
    interface ha-7cf36911-75
    virtual_router_id 1
    priority 50
    garp_master_delay 60
    nopreempt
    advert_int 2
    track_interface {
        ha-7cf36911-75
    }
    virtual_ipaddress {
        169.254.0.1/24 dev ha-7cf36911-75
    }
    virtual_ipaddress_excluded {
        10.0.0.1/24 dev qr-50deb6c5-c7
        192.168.81.102/24 dev qg-e93ae851-38
        2001:db8:cafe:17::66/64 dev qg-e93ae851-38
        2001:db8:cafe:beef::1/64 dev qr-1b8ada84-61
        fe80::f816:3eff:fe50:ab1/64 dev qr-50deb6c5-c7 scope link
        fe80::f816:3eff:feca:b249/64 dev qg-e93ae851-38 scope link
        fe80::f816:3eff:fee7:f435/64 dev qr-1b8ada84-61 scope link
    }
    virtual_routes {
        ::/0 via 2001:db8:cafe:17::1 dev qg-e93ae851-38
        0.0.0.0/0 via 192.168.81.2 dev qg-e93ae851-38
    }
}
```
# ip netns exec qrouter-0772d696-ec0f-46f3-b7d0-a984612fcdca ip a

9: ha-7cf36911-75: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN
   link/ether fa:16:3e:7e:bc:ae brd ff:ff:ff:ff:ff:ff
   inet 169.254.192.2/18 brd 169.254.255.255 scope global ha-7cf36911-75
      valid_lft forever preferred_lft forever
   inet 169.254.0.1/24 scope global ha-7cf36911-75
      valid_lft forever preferred_lft forever
   inet6 fe80::f816:3eff:fe7e:bcae/64 scope link
      valid_lft forever preferred_lft forever
11: qr-50deb6c5-c7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN
   link/ether fa:16:3e:50:0a:b1 brd ff:ff:ff:ff:ff:ff
   inet 10.0.0.1/24 scope global qr-50deb6c5-c7
      valid_lft forever preferred_lft forever
   inet6 fe80::f816:3eff:fe50:ab1/64 scope link
      valid_lft forever preferred_lft forever
12: qr-1b8ada84-61: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN
   link/ether fa:16:3e:e7:f4:35 brd ff:ff:ff:ff:ff:ff
   inet6 2001:db8:cafe:beef::1/64 scope global nodad
      valid_lft forever preferred_lft forever
   inet6 fe80::f816:3eff:fee7:f435/64 scope link
      valid_lft forever preferred_lft forever
13: qg-e93ae851-38: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN
   link/ether fa:16:3e:ca:b2:49 brd ff:ff:ff:ff:ff:ff
   inet 192.168.81.102/24 scope global qg-e93ae851-38
      valid_lft forever preferred_lft forever
   inet6 2001:db8:cafe:17::66/64 scope global nodad
      valid_lft forever preferred_lft forever
   inet6 fe80::f816:3eff:feca:b249/64 scope link
      valid_lft forever preferred_lft forever

L3 HA Interface

Tenant Network

External/Public Network
VRRPv2 Advertisement

# ip netns exec qrouter-0772d696-ec0f-46f3-b7d0-a984612fcdca tcpdump -n -i ha-7cf36911-75
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ha-7cf36911-75, link-type EN10MB (Ethernet), capture size 65535 bytes
15:17:19.021100 IP 169.254.192.2 > 224.0.0.18: VRRPv2, Advertisement, vrid 1, prio 50, authtype none, intvl 2s, length 20
15:17:21.021783 IP 169.254.192.2 > 224.0.0.18: VRRPv2, Advertisement, vrid 1, prio 50, authtype none, intvl 2s, length 20
15:17:23.023316 IP 169.254.192.2 > 224.0.0.18: VRRPv2, Advertisement, vrid 1, prio 50, authtype none, intvl 2s, length 20
15:17:25.025260 IP 169.254.192.2 > 224.0.0.18: VRRPv2, Advertisement, vrid 1, prio 50, authtype none, intvl 2s, length 20
Testing a failure

Check who is master:

[root@c7-m-aio ~]# cat /var/lib/neutron/ha_confs/0772d696-ec0f-46f3-b7d0-a984612fcdca/state
master
[root@c7-m-net-cmp ~]# cat /var/lib/neutron/ha_confs/0772d696-ec0f-46f3-b7d0-a984612fcdca/state
backup

Simulate a failure by shutting down the HA interface (remember this was in the ‘track’ list):

[root@c7-m-aio ~]# ip netns exec qroutertollue-7cf36911-75 down

Check that VRRP switched to the other node as master:

[root@c7-m-aio ~]# cat /var/lib/neutron/ha_confs/0772d696-ec0f-46f3-b7d0-a984612fcdca/state
fault
[root@c7-m-net-cmp ~]# cat /var/lib/neutron/ha_confs/0772d696-ec0f-46f3-b7d0-a984612fcdca/state
master
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

compute-server01

control-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

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

“br-int” view qr-xx & tapxx

A tap interface for each network used for DHCP service:

- qr-024a0619-71
- qr-10f02a4b-ab
- qr-b37e1034-06
- qr-ef7c1e0c-79
- tap2340872e-68
- tap271689cd-23
- tap3fe91abf-c8
- tap60a25081-14
- tap6d3911a5-44

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

br-int

Control-Server

compute-server01

Public: 192.168.238.x/24

VM

VM

VM

bridge-to-router 1 for each tenant
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
```

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
qbr180f8458-7b   8000.1a425eeda354       no
qbr3e60deda-cc   8000.8a70b498c8ce        no
qbr92774056-da   8000.3e21bdf7dd5b        no
```
Example – OVS Bridge/Neutron Router

```bash
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

DHCP

- control-server

IP Tables/Floating IP

- Public: 192.168.238.x/24

Management Network: 10.121.13.x

VM Boots

- br-int

- patch-tun

- patch-int

- br-tun

- eth0

- 10.121.13.50

- 10.121.13.51

GRE tunnel

- BRKDCN-2367

- © 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public
OpenStack Storage Projects
OpenStack Storage - Summary
(More Details to follow)

- Glance: Image repository for OpenStack
- Cinder: Controls data volumes for file systems and snapshots
- Swift: Object data storage service for “web scale” objects, and archive/backup
- Manila – Shared File System service
- Savanna>Sahara – HDFS File System (Hadoop)
- Local Storage on Compute Nodes: VM’s boot disk. Lives as long as an instance does. NOT persistent. Disappears when an instance is terminated
Glance

OpenStack’s Image Repository

• “The Glance project provides a service where users can upload and discover data assets that are meant to be used with other services. This currently includes images and metadata definitions.”

• Translation… Think of Glance as the catalog of images I can choose from to launch my VM

• Example: Glance has an Ubuntu 12.04, Ubuntu, 14.04, Centos 7.2, and Cirros 3.3. I’ll launch the image that best fits my application needs
Glance has a “Backend”

- Glance is an abstracted image repository from ”some storage mechanism”
- Cinder might very well be leveraged to host the Glance image store
- …or a more “traditional” storage mechanism such as NFS, a SAN or a NAS

*(Ceph might alternatively be leveraged to host the Glance image store – more on Ceph soon)*
Cinder

Cinder’s Mission Statement

• “To implement services and libraries to provide on demand, self-service access to Block Storage resources. Provide Software Defined Block Storage via abstraction and automation on top of various traditional backend block storage devices”

• Translation… Cinder lets you dynamically attach persistent virtual disks to your virtual machine instances (Think second hard drive)

• Example – I have a database running with a VM. Rather than upgrade the VM, I detach the database, launch a new VM with the desired OS, and re-attach the Cinder Volume
Cinder Volume Attach - LVM

iSCSI is an example, Cinder now supports FC and NFS protocols.
Cinder Volume Attach – External Storage

iSCSI is an example, Cinder now supports FC and NFS protocols
Cinder Volume Attach – Ceph RBD

Ceph volumes are accessed via TCP/IP over the RBD protocol

RADOS = Reliable Autonomic
Distributed Object Store
RBD = RADOS Block Device
Nova-Boot’s Storage Requirements

Nova boot requires storage as well…

- When you launch a VM on a Compute node, in the most basic setup, the VM’s boot volume is running within a directory in the compute node’s local FS. (By default, the boot partition)

- To take advantage of things like “Live Migration”, this directory must be accessible by multiple compute nodes, which typically means an “NFS like” mount or specifically configuring Nova to use another storage mechanism
A Note on Volume based VM’s and/or Snapshots

If you need to, you can build customized images for your project from any ISO image to a volume.
The Converted Volume is now a Standard Image

If you need to, you can build customized images for your project from any ISO.
OpenStack Storage

OpenStack Storage Services & API’s Abstract the underlying storage architecture

- OpenStack API’s
- OpenStack CLI’s
- OpenStack Horizon

OpenStack Storage Related Services

- Nova
- CINDER
- GLANCE
- SWIFT

Local File Systems
On Compute Nodes

- NAS
- SAN
- Object Storage
Typical Data Storage Requirements in OpenStack

- **VM Boot Partitions**
- **Image Repository**
- **Persistent Databases**
- **Document Repo’s**
- **Log Repo’s**
- **Snapshots**

**Image Repository**

**Object Storage for “Web Scale”**
- MP4’s
- Images
- Videos
- Music

**Swift**

**Cinder**

**Manilla**

**Glance**

**Nova Boot**
Introduction to Swift
Why Swift?

Common Use Cases

- Typically for high growth (low change) storage requirements
- Lower the cost for storage of (typically) static objects
  - Images… Movies… Files…
- Backups and archives
- High Availability & Multi-Tenancy
- Seamless failure handling (Usually X3 data replication)
- Compatible with S3 Solutions
- Designed to run on commodity hardware
What Swift is *NOT*

Not to be confused with…

• A drive or filesystem
• A NFS (Network File System)
• Block Storage
• Any SAN / NAS / DAS type solution
• A CDN

Swift is an Object File Storage solution. “Web Scale” storage scale for content that doesn’t often change
Swift Architecture

Swift Chooses Availability and Fault Tolerance

- Swift is Highly Available, distributed and “Eventually Consistent”
- “Eventually Consistent” means that the write is acknowledged before confirming full replication
- 3 “Rings” Hashing algorithm to replicate:
  - Accounts
  - Containers
  - Objects
- The Ring represents a mapping between the names of entities stored on disk and their physical location
Swift Architecture

Typically 2 Types of Nodes running various process

- **Proxy Servers:**
  - **Swift-proxy-server**: As the name suggests, proxies requests to the appropriate storage nodes where the data resides
  - **TempAuth**: WSGI middleware for authentication

- **Storage Servers:**
  - **Swift-account-server**: Account databases
  - **Swift-container-server**: Container databases
  - **Swift-object-server**: The actual storage Objects
Swift Architecture

**Proxy Tier** handling incoming requests

Proxy Layer Scales Horizontally

![Diagram showing the Proxy Tier and Storage](image-url)
Swift Architecture

The Ring – Handling Write Requests

- **3 Rings:**
  - Account Data
  - Container Data
  - Object Data

- Proxy Servers map data to specific storage servers, typically with 3 x redundancy

- Proxy acknowledges after 2\textsuperscript{nd} verified write

- Uploads md5 checksum with each object – Only saves object if checksum matches

- Eventually Consistent
Swift Architecture
Eventually Consistent

• “Eventually Consistent”
• Verify after 2 writes, and the 3rd copy is eventually replicated
Swift Architecture

The Ring – Handling Read Requests

- **3 Rings:**
  - Account Data
  - Container Data
  - Object Data

- Proxy servers lookup data location
- Read from a single disk
Swift Architecture

Replication

- Running in the background continuously, to ensure there are always (by default) 3 copies of everything

- (Typically) leverages a separate physical network segment so as not to compete with VM traffic

- In the event of a storage server failure (resulting in 2 copies of some data), any exposed data is re-copied to other valid servers
Swift Architecture

Replication

- Running in the background continuously, to ensure there are always (by default) 3 copies of everything

- (Typically) leverages a separate physical network segment so as not to compete with VM traffic

- In the event of a storage server failure (resulting in 2 copies of some data), any exposed data is re-copied to other valid servers
Swift Features

• **ACL**
  • Free form implemented by the auth system

• **Health Checking**
  • Simple Health checking page for Load Balancing

• **Rate Limiting**
  • Rate Limiting Requests

• **Static or Temporary Web Containers**
  • index.html generated for objects or containers

• **Form Post**
  • Translates a POST request to a regular Swift PUT

• **Domain Remap**
  • “Pretty” URL with domains based containers
Swift Features

- **Bulk Operations**
  - Run multiple Delete, Upload or even tar uploads

- **Account Quotas**
  - Ability to limit or set only as read accounts

- **Container Quotas**
  - Allows user to restrict a public container

- **Large Object Support (>5GB)**
  - Split large files into multiple objects. Reassembles on download

- **CORS**
  - Upload support directly from a browser via Javascript

- **Versioning**
  - Support for versioning all objects within a container
Swift API

Simple REST API

GET  POST  DELETE  HEAD  Operations

Simple Responses

200’s (Good)  300’s (redirect)  400’s (Bad Client)  500’s (Bad Server)

Bindings exist for different languages: Python, Java, C++ etc…
Multiple CLI tools... Python-client, jcloud, CURL…
The Swift API

- **Swift CLI – Which just leverages the Swift API**
  - POST, List, Delete, Download, Stat, Upload, Capabilities
  - POST – Updates metadata for the container, account, or Object

- **Examples of Metadata (HTTP Headers)**
  - X-account-access-control (ACL Control)
  - X-account-Sysmeta-global-write-ratelimt (Rate Limiting)
  - X-Object-manifest (Large dynamic objects)
  - X-Versions-Location (Object Versioning)
  - X-Container-Sync-<details> (used internally for container synchronization)
  - X-Delete-At / X-Delete-After (controls object expiration)
  - X-Container-Meta-Access-Control (CORS)

- **Other**
  - Crossfomain.xml (Cross Domain Policies)
The Swift API

- **Write an Object**…
  - PUT /v1/my_account_my_container

- **Read an Object**…
  - GET /v1/my_account_my_container

- **Delete an Object**…
  - DELETE /v1/my_account_my_container

Reference:
https://swift.domain.ame/v1/AUTH_acct/container/object
Ceph Overview
Ceph Storage Cluster (RADOS)

• The Ceph Storage Cluster – which is also referred to as RADOS (Reliable Autonomous Distributed Object Store) is the heart of the Ceph Distributed Storage System.

• Ceph OSD (Object Storage Device) Nodes are servers with multiple disks that have intelligent daemons called Ceph OSD Daemons.

• The Ceph OSD Daemons are responsible for joining the Ceph OSDs to the Ceph Storage Cluster, serving requests for data that lives on the OSDs, and communicating the status of the OSDs.

• Another core component of the Ceph Storage Cluster are Monitor Daemons, which monitor the system
Ceph Storage Cluster Services

Objects
Object Gateway
A powerful S3 and Swift compatible gateway that brings the power of the Ceph Object Store to modern Applications

Virtual Disks
Block Device
Distributed virtual block device that delivers high – performance, cost effective storage for virtual machines

Files and Directories
File System
Distributed, scale-out filesystem with POSIX semantics that provides storage for legacy and modern applications

Ceph Storage Cluster
Reliable, easy to manage, next generation distributed object store that provides storage of unstructured data for applications
Ceph Components

LIBRADOS
A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP

RADOSGW
A bucket-based REST gateway, compatible with S3 and Swift

RBD
A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver

CEPH FS
A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE

RADOS
A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes using CRUSH for data distribution.
Core Cluster Components

- **OSDs:**
  - 10s to 10000s in a cluster
  - One per disk (or one per SSD, RAID group…)
  - Serve stored objects to clients
  - Intelligently peer to perform replication and recovery tasks

- **Monitors:**
  - Maintain cluster membership and state
  - Provide consensus for distributed decision-making
  - Small, odd number
  - Do not serve stored objects to client
Ceph OSD Node Components

- Ceph Object Storage Daemon
- Filesystem: XFS
- Enterprise HDD

Write Journals

OSD

XFS

企业发展硬盘

JRN}
Ceph Storage Cluster (RADOS)

Monitor Node

OSD

XFS

Storage Node

Ceph Cluster
Ceph Cluster Topology 3 Nodes