Neutron networking with Red Hat Enterprise Linux OpenStack Platform

Nir Yechiel,
Networking Technology Product Manager, OpenStack
Red Hat
Agenda

- Neutron refresher
- Deep dive into ML2/Open vSwitch
  - Focus on L2, DHCP, and L3
- Our partner ecosystem and other commercial plugins
- Overview of recent major enhancements
  - IPv6, L3 HA, Distributed Virtual Routing (DVR)
- Q&A
RHEL OpenStack Platform 6

IaaS+
- Monitoring: Ceilometer
- Data Processing: Sahara
- Orchestration: Heat
- DBaaS: Trove
- Deployment: TripleO

IaaS
- Compute: Nova
- Storage: Cinder, Glance, Swift
- Networking: Neutron
- Bare Metal Provisioning: Ironic

Shared Services
- Identity
- Keystone
- Dashboard
- Horizon

Red Hat Enterprise Linux

Tech preview
What is Neutron?

- Fully supported and integrated OpenStack project
- Exposes an API for defining rich network configuration
- Offers multi-tenancy with self-service
What Neutron is not?

- Neutron does not implement the networks
  - Using the concept of plugins
The Plugin Matters...

- Feature set
- Scale
- Performance
- High Availability
- Manageability
- Network topology
- Traffic flow
- Operational tools
Neutron Key Features

● L2 connectivity
● IP Address Management
● Security Groups
● L3 routing
● External gateway, NAT and floating IPs
● Load balancing, VPN and firewall
Dashboard View
Red Hat Neutron Focus

- ML2 with Open vSwitch Mechanism Driver (today)
  - Overlay networks with VXLAN
- ML2 with OpenDaylight Mechanism Driver (roadmap)
- Broad range of commercial partners
Neutron with
ML2 and Open vSwitch
(Tenant networks, VXLAN)
Refresher: Open vSwitch (OVS)

- Multi-layer software switch
- Included with RHEL OpenStack Platform
- Highlights:
  - Multi-threaded user space switching daemon for increased scalability
  - Support for wildcard flows in Kernel datapath
  - Kernel based hardware offload for GRE and VXLAN
  - OpenFlow and OVSDB management protocols
Refresher: Network Namespaces (ip netns)

- Multiple discrete copies of the networking stack in Linux
- Analogous to VRFs on network devices
- Make it possible to separate network domains
  - Interfaces, IP addresses, routing tables, iptable rules, sockets, etc.
ML2/OVS Plugin

- Software only solution, hardware agnostic
- Support for VLAN, GRE, and VXLAN dataplane
- Tenant routers and DHCP servers implemented as network namespaces
  - Recommended deployment is using the concept of Network Nodes
Main Components

- OVS L2 agent
- DHCP agent
- L3 agent
- Metadata agent and proxy
- Load balancing, VPN and firewall served by distinct plugins/agents
Common Deployment - Placement

Controller Nodes
- Neutron server
- ML2 core plugin
- Service plugins
- OVS ML2 driver

Network Nodes
- L3 agent
- Metadata agent
- DHCP agent
- Service agents
- OVS agent
- Open vSwitch

Compute Nodes
- OVS agent
- Open vSwitch
Common Deployment - Networks

Controller Nodes

Compute Nodes
VM VM VM

Network Nodes
Routing, NAT, DHCP

Management
Tenant Data
External

#redhat #rhsummit
L2 Connectivity
Network Separation

- **802.1Q VLANs**
  - Require end-to-end provisioning
  - Number of IDs: 4K (theoretically)
  - VM MAC addresses typically visible in the network core
  - Well known by network admins as well as the network equipment
Network Separation

- **802.1Q VLANs**
  - Require end-to-end provisioning
  - Number of IDs: 4K (theoretically)
  - VM MAC addresses typically visible in the network core
  - Well known by network admins as well as the network equipment

- **Overlay tunnels (GRE, VXLAN)**
  - Decouple virtual networking from physical fabric
  - Network provides only IP transport
  - Various design and performance considerations
    - MAC to VTEP mapping, MTU, hardware offload, load sharing
Leaf/Spine with VLANs

STP
MLAG
TRILL
802.1ad (QinQ)

L2 (trunk) links

resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node
resource node

ToR/leaf switch

spine switch

spine switch

spine switch

ToR/leaf switch

ToR/leaf switch

ToR/leaf switch
Leaf/Spine with Overlays

OSPF, BGP
ECMP
VXLAN, GRE

L3 (routed) links

ToR/leaf switch

spine switch

VTEP

resource node

resource node

resource node

resource node

resource node

resource node

resource node

resource node

resource node

resource node
L2 Connectivity

- Between VMs on the same Compute
  - Traffic is bridged locally using normal MAC learning
  - Each tenant gets a local VLAN ID
  - No need to leave ‘br-int’
Tenant flows are separated by internal, locally significant, VLAN IDs. VMs that are connected to the same tenant network get the same VLAN tag.
L2 Connectivity

- Between VMs on different Computes
  - OVS acts as the VTEP
  - Flow rules are installed on ‘br-tun’ to encapsulate the traffic with the correct VXLAN VNI
Tenant flows are separated by internal, locally significant, VLAN IDs. VMs that are connected to the same tenant network get the same VLAN tag.

Internal VLANs are converted to tunnels with unique GRE Key or VXLAN VNI per network.

Source interface is determined from "local_ip" configuration through routing lookup.
GRE/VXLAN - Tunnel Layout

- Tunnel creation -
  - L2 agent goes up and notifies Neutron server via RPC
  - Neutron notifies other nodes that a new node has joined
  - Tunnel is formed between the new node and every pre-existing node

- VXLAN IP Multicast control plane was not implemented in OVS

- Broadcast, unknown unicast and multicast are forwarded out all tunnels via multiple unicast packets
  - Optimization to this available using the l2-population driver
L2 Population Mechanism Driver

- Neutron service has full knowledge of the topology
  - MAC and IP of each Neutron port
  - The node (VTEP) that the port was scheduled on

- Forwarding tables are programmed beforehand

- Processing of ARPs can be further optimized
  - Reply from the local vSwitch instead of traversing the network
With L2 Population

(1) A broadcast from VM B is sent and replicated to nodes hosting VMs on that network only.

(2) The broadcast will reach the VM on this compute successfully.

The agents decide when to create new tunnels or destroy ones based on forwarding entries.
Local ARP Response

- ARP messages are treated as normal broadcasts by default
  - Even with l2-pop enabled - still need to traverse the network

- Enter ARP Responder
  - A new table is inserted into br-tun, to be used as an ARP table
  - The table is filled whenever new L2 pop address changes come in
  - Local switch construct an ARP Reply contains the MAC address of the remote VM
L2 Population with ARP Responder

Open vSwitch

- **Table: Unicast**
- **Table: Broadcast/Multicast**
- **Table: ARP**

**br-int**

- Local traffic is bridged on br-int

**br-tun**

- Unicast packets are replicated to the correct tunnel
- Broadcasts are replicated to the correct tunnels

ARP frames (Ethertype 0x806) from local VMs are matched and directed into the ARP table. Response is generated locally.
Security Groups
Security Groups

- Per VM stateless ACLs
- Increased intra-subnet and inter-subnet security
- Default group drops all ingress traffic and allows all egress
- Current solution implemented with iptables
- User flow:
  - Assign VMs to groups
  - Specify filtering rules between groups
  - Can match based on IP addresses, ICMP codes, TCP/UDP ports, etc.
# Security Groups

## Manage Security Group Rules: standard

### Security Group Rules

<table>
<thead>
<tr>
<th>Direction</th>
<th>Ether Type</th>
<th>IP Protocol</th>
<th>Port Range</th>
<th>Remote</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingress</td>
<td>-</td>
<td>ICMP</td>
<td>-1 (All ICMP)</td>
<td>0.0.0.0/0 (CIDR)</td>
<td>Delete Rule</td>
</tr>
<tr>
<td>Ingress</td>
<td>-</td>
<td>TCP</td>
<td>22 (SSH)</td>
<td>0.0.0.0/0 (CIDR)</td>
<td>Delete Rule</td>
</tr>
<tr>
<td>Ingress</td>
<td>-</td>
<td>TCP</td>
<td>80 (HTTP)</td>
<td>0.0.0.0/0 (CIDR)</td>
<td>Delete Rule</td>
</tr>
<tr>
<td>Ingress</td>
<td>-</td>
<td>TCP</td>
<td>443 (HTTPS)</td>
<td>0.0.0.0/0 (CIDR)</td>
<td>Delete Rule</td>
</tr>
</tbody>
</table>

Displaying 4 items
Bridge device is necessary - offers a route to the kernel for filtering

OVS can’t directly attach a TAP device where iptables rules are applied
DHCP Service (IPv4)
DHCP

- IPv4 subnets are enabled with DHCP by default
- Neutron is the single source of truth
  - IP addresses are allocated by Neutron and reserved in the Neutron DB
- Standard DHCP is used to populate the information to VMs
  - UDP ports 67/68
  - DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, DHCPACK
- Default solution implemented with Dnsmasq
DHCP - Network Node

- Virtual interface
- Open vSwitch

Each service is separated by internal VLAN ID per tenant

DHCP namespace is created per tenant subnet. This namespace is managed by the dhcp-agent

Internal VLANs are converted to tunnels with unique GRE Key or VXLAN VNI per network

Source interface is determined from “local_ip” configuration through routing lookup

DHCP namespace is created per tenant subnet. This namespace is managed by the dhcp-agent

Internal VLANs are converted to tunnels with unique GRE Key or VXLAN VNI per network

Source interface is determined from “local_ip” configuration through routing lookup
L3 Routing and NAT (IPv4)
Routing/NAT Features

- East/West routing
- VMs with public IP addresses (floating IPs)
  - Static stateless (1:1) NAT
- Default access to outside system
  - Dynamic stateful NAPT (aka SNAT)
- Implemented with Linux IP stack and iptables
  - Network namespaces with ‘net.ipv4.ip_forward=1’
Each router interface is separated by internal VLAN ID per tenant.

Internal VLANs are converted to tunnels with unique GRE Key or VXLAN VNI per network.

Routing namespace is created per router. This namespace is managed by the l3-agent.

IP is assigned from the external pool.

Interface on external network. This network should have externally reachable IP pool.

Virtual interface

Open vSwitch

veth pair

Tenant default gateway

Uplink used for NAT

VLAN ID

br-int

patch-tun

patch-int

br-tun

int-br-ex

phy-br-ex

br-ex

eth

eth

br-ex

qg-xxxx

qr-xxxx

Namespace-1

br-tun

br-int

eth

eth

int-br-ex

phy-br-ex

veth pair
Routing - Example

**Network Diagram**

- **172.17.17.1**
  - qr-xxxx
  - br-tun
  - eth

- **192.168.101.2**
  - qg-xxxx
  - br-ex
  - eth

**Network Configuration**

- **172.17.17.1**
  - qr-xxxx
  - br-tun
  - eth

- **192.168.101.2**
  - qg-xxxx
  - br-ex
  - eth
Routing - Example

Default SNAT -
-A quantum-l3-agent-snat -s 172.17.17.0/24 -j SNAT --to-source 192.168.101.2

Floating IP (1:1 NAT) -
-A quantum-l3-agent-PREROUTING -d 192.168.101.3/32 -j DNAT --to-destination 172.17.17.2
Neutron with Our Commercial Partners
Commercial Neutron Plugins

- Two main models:
  - **Software centric** - hardware is general-purpose
    - Decouple virtual networking from physical “fabric”
    - e.g Midokura MidoNet, Nuage VSP, PLUMgrid ONS
  - **Hardware centric** - specific network hardware is required
    - Ability to control and interact with the physical network
    - e.g Cisco ACI, Brocade VCS

- ML2 drivers, core plugins, advanced services
Certification at Red Hat

- Collaboration between Red Hat and technology partners
- Assure our customers that:
  - Technology stack has been tested and validated
  - Solution is fully supported by Red Hat and partners
Certification at Red Hat

- Covers two main areas:
  - Validation that the product implements the right OpenStack interfaces
  - Verification that the production version of RHEL OpenStack Platform stack is used, and that the product is not configured in a way that would invalidate support

- Current Certification for Neutron covers core plugins, ML2 drivers, and service plugins for LBaaS
  - Find out more at https://access.redhat.com/certifications
Our Neutron Ecosystem
Certified Neutron Plugins (RHEL OpenStack Platform 5)

- Big Switch Networks - Big Cloud Fabric
- Brocade - VCS
- CPLANE NETWORKS - Dynamic Virtual Networks
- Cisco - Nexus, N1KV, Application Policy Infrastructure Controller (APIC)
- Mellanox - Embedded Switch
- Pluribus Networks - Netvisor
- Midokura - Midokura Enterprise MidoNet
- NEC - Programmable Flow
- Nuage - Virtualized Services Platform (VSP)
- PLUMgrid - Open Networking Suite (ONS)
- One Convergence - Network Virtualization and Service Delivery
- Radware - Alteon LBaaS for OpenStack Neutron
- Avi Networks - Cloud Application Delivery Platform (CADP)
Certified Neutron Plugins (RHEL OpenStack Platform 6)

- **Big Switch Networks** - *Big Cloud Fabric*
- **Brocade** - *VCS*
- **Cisco** - *Nexus, N1KV, Application Policy Infrastructure Controller (APIC)*
- **Midokura** - *Midokura Enterprise MidoNet*
- **NEC** - *Programmable Flow*
- **Nuage** - *Virtualized Services Platform (VSP)*
- **PLUMgrid** - *Open Networking Suite (ONS)*
- **Radware** - *Alteon LBaaS for OpenStack Neutron*
- **Avi Networks** - *Cloud Application Delivery Platform (In Progress)*
- **F5** - *BIG-IP OpenStack Neutron LBaaS (In Progress)*
- **Mellanox** - *Embedded Switch (In Progress)*
Recent Enhancements
IPv6
Do You Need IPv6?

Source: https://twitter.com/SCOTTHOGG/status/603213942429601792
IPv6: The Basics

- No more broadcasts, no ARP
  - Neighbor Solicitation with ICMPv6 Neighbor Discovery

- Link Local addresses
  - Mandatory on each interface, start with FE80
  - Used for communication among IPv6 hosts on a link (no routing)

- Global Unicast addresses
  - Globally routed addresses, start with 2000:: /3

- Router is required for SLAAC, and for advertising default-route
IPv6: Address Assignment

- **Static**

- **Stateless Address Autoconfiguration (RFC 4862)**
  - Nodes listen for Router Advertisements (RA) messages
  - Create a Global Unicast IPv6 address by combining:
    - EUI-64 address
    - Link Prefix

- **DHCPv6 (RFC 3315)**
  - Stateless
  - Stateful
IPv6 with RHEL OpenStack Platform 6

- Two new Subnet attributes introduced:
  - `ipv6-ra-mode` - determine who sends Router Advertisements
  - `ipv6-address-mode` - determine how VM obtains IPv6 address, default gateway, and/or optional information

- VMs can obtain address via SLAAC or DHCPv6
  - Routers send out Router Advertisements (RAs)
  - Neutron can generate an address via EUI-64 specification
  - Implementation uses Dnsmasq and radvd

- Security Groups support IPv6
IPv6 with RHEL OpenStack Platform 6

- **BYOA (bring your own address) model**
  - Tenants are trusted to choose their own IPv6 addressing

- **No NAT or floating IP support for IPv6**
  - Assumption is that tenants are assigned with globally routed addresses
  - Neutron router is configured with a default gateway to external network
IPv6 - Network Node

Routing namespace is created per router. This namespace is managed by the l3-agent.

Internal VLANs are converted to tunnels with unique GRE Key or VXLAN VNI per network.

DHCP namespace is created per tenant subnet. This namespace is managed by the dhcp-agent.

Each service is separated by internal VLAN ID per tenant.

Source interface is determined from “local_ip” configuration through routing lookup.
L3 Agent HA
L3 High Availability

- L3 HA architecture based on keepalived/VRRP protocol
  - Supported since RHEL OpenStack Platform 6
- Designed to provide HA for centralized Network nodes
L3 High Availability

- Virtual Router Redundancy Protocol - RFC 5798
  - Uses IP protocol number 112
  - Communicates via multicast 224.0.0.18
  - Master/Backup election based on priority
  - Virtual MAC in format 00-00-5E-00-01-XX
L3 High Availability

- Routers are scheduled on two or more Network nodes
- Internal HA network is created per tenant
  - Used to transport the VRRP messages
  - Hidden from tenant CLI and Dashboard
  - Uses the tenant default segmentation (e.g. VLAN, VXLAN)
- keepalived process is spawned per virtual router
  - HA group is maintained for each router
  - IPv4 Link Local addresses (default 169.254.192.0/18) are being used
  - Master/Backup are placed randomly
L3 High Availability

Network Node 1
- vRouter Master
- vRouter Backup
- vRouter Backup
- vRouter Backup

Network Node 2
- vRouter Backup
- vRouter Master
- vRouter Backup
- vRouter Backup

Network Node 3
- vRouter Backup
- vRouter Backup
- vRouter Master
- vRouter Backup
L3 High Availability

Router Namespace

Tenant D.G
qr-xxxx

HA
ha-xxxx

Uplink
qg-xxxx

VLAN
br-int

patch-tun

patch-int

br-tun

eth

VLAN
br-int

int-br-ex

phy-br-ex

br-ex

eth

VLAN
br-int

int-br-ex

phy-br-ex

br-ex

eth

Virtual interface

Open vSwitch

veth pair
Distributed Virtual Routing (Technology Preview)
What is DVR?

- Distributed east/west routing and floating IPs
  - L3 agents running on each and every compute node
  - Metadata agent distributed as well
- Default SNAT still centralized
- Implementation is specific to ML2 with OVS driver
- Fundamentally changes the deployment architecture
  - External network is required on Compute nodes for north/south connectivity
Deployment with DVR

Controller Nodes

Compute Nodes
agent_mode = dvr
L3 Agent
Metadata Agent

Network Nodes
agent_mode = dvr_snat
L3 Agent
Metadata Agent

Management
Tenant Data
External
What’s Next

- Role-based Access Control (RBAC) for networks
- Neutron quality of service (QoS)
- Pluggable IPAM
- IPv6 Prefix Delegation
- L3 HA + L2 Population
- L3 HA support for IPv6
- Stateful OVS firewall
- VLAN trunking into a VM
Questions?

Don’t forget to submit feedback using the Red Hat Summit app.

✉️ nyechiel@redhat.com
🐦 @nyechiel
LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.