FROM THE SOLUTIONS CENTER

Red Hat OpenStack with Brocade ML2 and SVI driver - KILO

This guide describes how to deploy Red Hat OpenStack using Pack Stack and enable the Brocade ML2 and SVI driver for Neutron networking

GLOSSARY

Term	Meaning
ML2	Modular Layer 2
RHEL	Red Hat Enterprise Linux
SVI	Switched virtual interface

INTRODUCTION

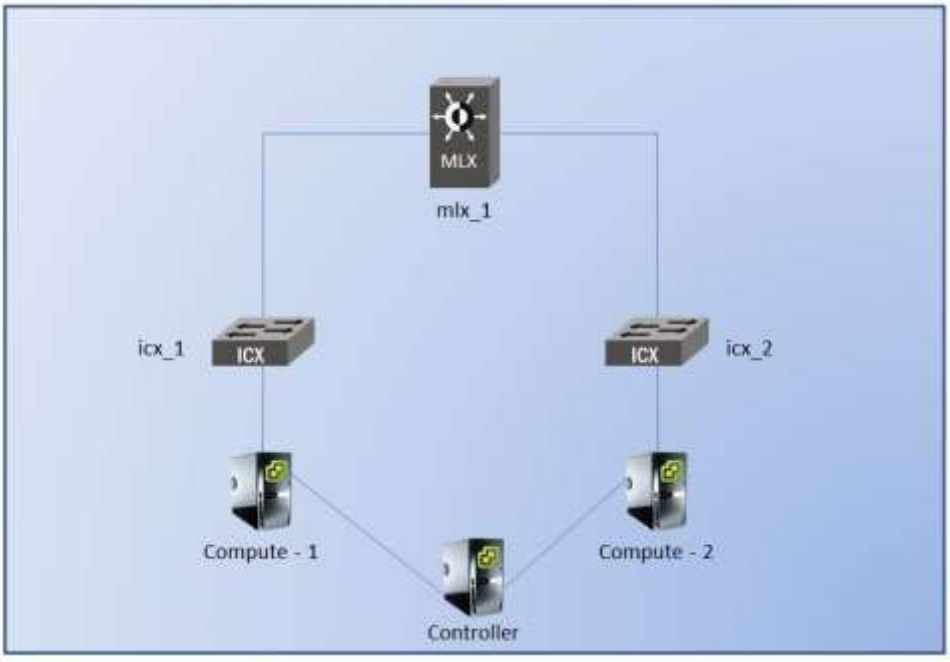
This document describes a Red Hat OpenStack 7.0 installation via PackStack with OpenvSwitch plugin. Once complete, Neutron can be reconfigured to use the Brocade ML2 and SVI driver for managing both virtual and physical networking infrastructure through OpenStack API.

Brocade ML2 and SVI plugin helps to configure L2 and L3 Networking on the underlying Ethernet fabrics from OpenStack Neutron Service.

This document provides an overview of Brocade's based ML2 plugin and SVI L3 routing solutions.

This guide has been tested using RHEL 7.0.

PHYSICAL TOPOLOGY



NOTE: The L2 device can be ICX, MLX, VDX or any vendor specific L2 devices, here we used ICX as L2 devices and all the ML2 configurations will be corresponding to that.

SERVER INTERFACE CONFIGURATION:

For Tenant networks, edit the configuration of the physical interface connected to the MLX. It should be configured with no IP address and in promiscuous mode. All nodes should have a similar configuration.

```
NAME=eth1
BOOTPROTO=static
ONBOOT=yes
TYPE=Ethernet
```

One method to configure the interface for promiscuous mode during boot, is to create /sbin/ifup-local with the following content

```
#!/bin/bash
if [[ "$1" == "eth1" ]]
then
  /sbin/ifconfig $1 promisc
  RC=$?
fi
```

Set executable bit. This script will run during boot right after network interfaces are brought online.

```
# chmod +x /sbin/ifup-local
# /etc/init.d/network restart
```

INSTALL THE SOFTWARE REPOS

Red Hat Enterprise Linux OpenStack Platform requires that each system in the OpenStack environment is running Red Hat Enterprise Linux Server and that all systems be signed up to receive updates from the Customer Portal Subscription Management using Subscription Manager.

Below steps in this procedure must be executed while logged in to the account of the `root` user on the system being registered.

```
#subscription-manager register
# subscription-manager list --available
# subscription-manager attach --pool=<pool ID from previous output>
# yum repolist
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-openstack-7.0-rpms
# subscription-manager repos --enable=rhel-7-server-rh-common-rpms
# subscription-manager repos --enable=rhel-7-server-optional-rpms
# subscription-manager repos --enable=rhel-7-server-openstack-7.0-installer-rpms
# subscription-manager repos --enable=rhel-server-rhscl-7-rpms
# yum -y update
# reboot
```

INSTALLATION USING PACKSTACK

Below procedure will walk through the process of deploying OpenStack on multi node (one controller and two compute nodes).

Begin by deploying OpenStack as documented in the RHEL OpenStack Platform –
Deploying OpenStack: Proof-of-Concept Environment (PackStack) guides at

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/7/html/Deploying_OpenStack_Proof_of_Concept_Environments/index.html

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

Followed PackStack interactive method to deploy controller and compute nodes. Additional compute nodes can be specified as mentioned below ('novacompute-hosts' field).

```
# packstack

Installer will be installed using the following configuration:
============================================================
ssh-public-key:                /root/.ssh/id_rsa.pub
```

```
default-password:
mariadb-install:                y
os-glance-install:              y
os-cinder-install:              y
os-nova-install:                y
os-neutron-install:             y
os-horizon-install:             y
os-swift-install:               y
os-ceilometer-install:          y
os-heat-install:                n
os-client-install:              y
ntp-servers:
nagios-install:                 y
exclude-servers:
os-debug-mode:                  n
os-controller-host:             10.24.41.110
os-compute-hosts:               10.24.41.110,10.24.42.66,10.24.42.68
os-network-hosts:               10.24.41.110
os-vmware:                      n
unsupported:                    n
use-epel:                       n
additional-repo:
rh-username:
rhn-satellite-server:
amqp-backend:                   rabbitmq
amqp-host:                      10.24.41.110
amqp-enable-ssl:                n
amqp-enable-auth:               n
mariadb-host:                   10.24.41.110
mariadb-pw:                     ********
keystone-db-passwd:             ********
keystone-admin-passwd:          ********
keystone-demo-passwd:           ********
glance-db-passwd:               ********
glance-ks-passwd:               ********
glance-backend:                 file
cinder-db-passwd:               ********
cinder-ks-passwd:               ********
cinder-backend:                 lvm
cinder-volumes-create:          y
cinder-volumes-size:            20G
nova-db-passwd:                 ********
nova-ks-passwd:                 ********
novasched-cpu-allocation-ratio:16.0
novasched-ram-allocation-ratio:1.5
novacompute-migrate-protocol:  tcp
os-neutron-ks-password:         ********
os-neutron-db-password:         ********
os-neutron-l3-ext-bridge:       br-ex
os-neutron-metadata-pw:         ********
os-neutron-lbaas-install:       n
os-neutron-metering-agent-install:n
neutron-fwaas:                  n
os-neutron-ml2-type-drivers:   vlan
os-neutron-ml2-tenant-network-types:vlan
os-neutron-ml2-mechanism-drivers:openvswitch
os-neutron-ml2-flat-networks:  *
os-neutron-ml2-vlan-ranges:    physnet1
os-neutron-ml2-tunnel-id-ranges:
os-neutron-ml2-vxlan-group:
os-neutron-ml2-vni-ranges:     10:100
```

```
os-neutron-l2-agent:           openvswitch
os-neutron-ovs-bridge-mappings:physnet1:br-eth1
os-neutron-ovs-bridge-interfaces:br-eth1:eth1
os-neutron-ovs-tunnel-if:
os-horizon-ssl:                n
os-swift-ks-passwd:            ********
os-swift-storages:
os-swift-storage-zones:        1
os-swift-storage-replicas:     1
os-swift-storage-fstype:       ext4
os-swift-storage-size:         2G
provision-demo:                y
provision-tempest:             n
provision-tempest-user:
provision-tempest-user-passwd: ********
provision-demo-floatrange:     172.24.4.224/28
provision-cirros-url:          http://download.cirros-cloud.net/0.3.3/cirros-0.3.3-x86_64-
disk.img
provision-tempest-repo-uri:    https://github.com/openstack/tempest.git
provision-tempest-repo-revision:master
provision-all-in-one-ovs-bridge:n
ceilometer-ks-passwd:          ********
mongodb-host:                  10.24.41.110
redis-host:                    10.24.41.110
redis-port:                    6379
nagios-passwd:                 ********
Proceed with the configuration listed above? (yes|no): yes
```

BROCADE NEUTRON PLUGINS

MI2 driver description

NOTE: The L2 device can be ICX, MLX, VDX or any vendor specific L2 devices, here we used ICX as L2 devices and all the ML2 configurations will be corresponding to that.

Brocade plugin for OpenStack Neutron Service implements the Neutron API to manage L2 network and SVI on MLX devices.

Brocade ICX Plugin will work with KILO release of OpenStack and extend the following Neutron API's

1. CREATE_NETWORK –
   On receiving a create_network call, Brocade Plugin would create VLAN in ICX device

   Note:

   - vlan number is allocated by the Neutron service based on the range provided in the configuration. For example, if the VLAN range provided in configuration is 100 – 200 then the first network, Red Network is carried as VLAN 100 and then Green Network is carried as VLAN 101 on the ICX box
   - Brocade ML2 driver doesn't support multiple physical network so external network creation will fail
   - MI2 driver supports vlan provider network

2. DELETE_NETWORK –
   On receiving a delete_network, Brocade Plugin would delete the VLAN in ICX device corresponding to that Network.

L3 driver description

3. Create_router, Update_router
   Create_router and update_router methods will create routers on MLX device.

4. add_interface_to_router

   Openstack subnet_id is provided in the request parameters. Using this API, Brocade Plugin will create ve interface and assign gateway ip of the subnet to the MLX device

5. Remove_interface_from_router

   Using this API, Brocade Plugin will remove ve interface along with assigned gateway ip of the subnet from MLX device

6. Delete_router

   Delete router deletes the router created.

Deploy MLX SVI driver

Install the Brocade ML2 plugin from the repository on controller nodes.

```
# git clone https://github.com/stackforge/networking-brocade.git -b stable/<branch-name>
    cd networking-brocade
    sudo python setup.py install


for Example: git clone https://github.com/stackforge/networking-brocade.git -b stable/kilo
```

Edit /etc/neutron/plugins/ml2/ml2_conf.ini

```
[ml2]
tenant_network_types = vlan
type_drivers = vlan
mechanism_drivers = openvswitch,brocade_fi_ni

[ml2_type_vlan]
network_vlan_ranges = physnet1:100:200

[securitygroup]
enable_security_group = True

[database]
connection = mysql://neutron:password@10.24.41.110/neutron

[ovs]
local_ip = 10.24.41.110
bridge_mappings = physnet1:br-eth1
integration_bridge = br-int
```

```
[ml2_brocade_fi_ni]

switch_names=icx_1, icx_2


[icx_1]
address  = 10.24.20.22
username = admin
password = password
physical_networks = physnet1
ports = 1/1/1, 1/1/2
transport = SSH
ostype   = FI

[icx_2]
address  = 10.24.20.32
username = admin
password = password
physical_networks = physnet1
ports = 1/1/1, 1/1/2
transport = SSH
ostype   = FI



[l3_brocade_mlx]
switch_names=mlx_1

[mlx_1]
address  = 10.25.225.30
username = admin
password = password
physical_networks = physnet1
ports = 1/12, 1/13
transport = SSH
ostype   = NI
```

Edit /etc/neutron/neutron.conf

```
core_plugin = ml2
```

Make sure the correct network interface is configured to bridges. If interface name is renamed add the renamed interface to the bridge.

```
#ovs-vsctl add-br br-eth1
#ovs-vsctl add-port br-eth1 eth1
#ovs-vsctl show
941f36d3-1dff-44ec-a930-517bf4083c8d
    Bridge br-int
        Port "int-br-eth1"
            Interface "int-br-eth1"
        Port br-int
            Interface br-int
                type: internal
    Bridge "br-eth1"
        Port "br-eth1"
            Interface "br-eth1"
```

```
                type: internal
        Port "phy-br-eth1"
            Interface "phy-br-eth1"
        Port "eth1"
            Interface "eth1"
    Bridge br-ex
        Port br-ex
            Interface br-ex
                type: internal
    ovs_version: "2.0.0"
```

SVI - L3 NETWORKING DRIVER.

This section describes how SVI feature can be leveraged to provide internetworking between networks configured using OpenStack.

Edit /etc/neutron/neutron.conf

NOTE: the below configuration is for KILO release, the service plugin value will be differ for releases.

```
service_plugins = neutron.services.l3_router.brocade.mlx.l3_router_plugin.BrocadeRouterPlugin
```

TESTING THINGS OUT – SVI L3 DRIVER

Use the CLI or Horizon to create router, add interface to the routers.

Configuring the SVI (Ve interface) IP address to be same as the Network's Gateway IP address would enable us to provide internetworking between the L3 networks.

For Eg: VLAN range is configured as 100 – 200,

Consider the Green Network subnet is 5.5.5.0/24 and Red Network subnet is 6.6.6.0/24

Red Network is carried as VLAN 100 and Green Network is carried as VLAN 101 on the ICX box (This is achieved using brocade ml2 driver Integration with OpenStack).

MLX Plugin would now configure the router interface IP address as the SVI (Ve) IP address on the switch.

Ve 100 is configured with 6.6.6.1 and Ve 101 is configured with 5.5.5.1

This would allow VM1 (6.6.6.2) on compute-1 to communicate with VM7(5.5.5.7) on compute-2