

Deploying Vyatta vRouter L3 OpenStack plugin

This guide describes how to install Vyatta vRouter L3 OpenStack plugin on Ubuntu 12.0.4 LTS server with OpenStack Juno release

BROCADE

OPENSTACK SERVER REQUIREMENTS

Server with Ubuntu 12.04 LTS and KVM

- OpenStack Juno installation
- One network interface for external network
- OpenvSwitch installation

VYATTA VRROUTER L3 PLUGIN INSTALLATION

Vyatta vRouter L3 plugin package file:

git clone <https://github.com/BRCDDcomm/vyatta.git>

or

Download from <https://github.com/BRCDDcomm/vyatta>

- Copy plugin files to OpenStack installation using the command:

```
cp -r ./vyatta/plugin/* /opt/stack/neutron/neutron/plugins/
```
- Copy plugin config file to OpenStack installation using the command:

```
cp -r ./vyatta/conf/* /etc/neutron/plugins/
```
- Apply the neutron patch using the command: (to disable ip spoofing iptables rules)

```
cp ./vyatta/neutron_patch/iptables_firewall.py  
/opt/stack/neutron/neutron/agent/linux/
```

OPEN vSWITCH BRIDGE SETUP

Create the Open vSwitch bridges for management and external networks:

```
sudo ovs-vsctl add-br br-int  
sudo ovs-vsctl add-br br-mng  
sudo ovs-vsctl add-br br-ext
```

Add network interfaces as ports in the OVS bridges. Ip-address of the interfaces should be deleted before adding to the bridges. Make sure to use console access for external network interface configuration as terminal connectivity will be lost while removing ip address.

```
sudo ip addr del <ip address> dev <ethX>  
sudo ovs-vsctl add-port br-ext <ethX>
```

Edit the file /etc/network/interfaces for assigning ip-address to the bridges as shown in below:

```
# External network interface  
auto eth0  
iface eth0 inet manual
```

```

up ip address add 0.0 dev $IFACE
up ip link set $IFACE up
down ip link set $IFACE down
auto br-ext
    iface br-ext inet static
    address 10.0.0.150
    netmask 255.255.255.0
    network 10.0.0.0
    broadcast 10.0.0.255
    gateway 10.0.0.1
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 10.0.0.1

# Internal Management network bridge
auto br-mng
    iface br-mng inet static
    address 192.168.0.1
    netmask 255.255.0.0
    network 192.168.0.0
    broadcast 192.168.255.255

```

Restart the network service to apply the configurations:

```
sudo service networking restart
```

Verify all the created bridges using the command:

```
sudo ovs-vsctl show
```

ML2 CONFIGURATION

Edit ML2 configuration file `/etc/neutron/plugins/ml2/ml2_conf.ini` to add external, management networks and the corresponding OVS bridges.

```

tenant_network_types = local,flat,vlan,gre,vxlan
type_drivers = local,flat,vlan,gre,vxlan
mechanism_drivers = openvswitch,linuxbridge
flat_networks = mng,ext
[ovs]
bridge_mappings = mng:br-mng,ext:br-ext

```

Edit Nova configuration file `/etc/nova/nova.conf` to set the `vif_type`:

```
vif_type = ovs
```

OPENSTACK FIREWALL CONFIGURATION

Edit ip-spoofing network filter configuration:

```

virsh nwfilter-edit no-ip-spoofing
Change 'drop to 'accept' as highlighted below:
<filter name='no-ip-spoofing' chain='ipv4' priority='-700'>
  <uuid>fce8ae33-e69e-83bf-262e-30786c1f8072</uuid>

```

```
<rule action='accept' direction='out' priority='500'>
  <ip match='no' srcipaddr='$IP' />
</rule>
</filter>
```

Edit the file /etc/nova/nova.conf and update firewall driver configuration as shown below:

```
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

IMPORT VYATTA vROUTER IMAGE

Openstack services should be running in order to import Vyatta vRouter image into glance repository. If you are using devstack, run rejoin-stack.sh in devstack directory to start Openstack. Set the admin tenant environment using the command: source openrc admin admin

Run the below command to import Vyatta vRouter image (distributed in the plugin package).

```
glance image-create --name "Vyatta vRouter" --is-public true --disk-format qcow2 --file vyatta_vrouter.qcow2 --container-format bare
```

Note the image id from the command output. This value needs to be added to the Vyatta plugin configuration file.

CREATE MANAGEMENT AND EXTERNAL NETWORKS

Create the management and external networks in OpenStack using the below commands:

```
neutron net-create management --provider:network_type flat
--provider:physical_network mng
```

Note the management network id from the command output. This value needs to be added to the Vyatta plugin configuration file.

Create the management subnet as shown below:

```
neutron subnet-create management --no-gateway 192.168.0.0/16 --allocation-pool start=192.168.0.128,end=192.168.254.254
```

Create the external network as shown below. Make sure to use the right values of external subnet and allocation pool from your environment.

```
neutron net-create external --provider:network_type flat
--provider:physical_network ext --router:external=True
```

```
neutron subnet-create external 10.0.0.0/24 --allocation-pool start=10.0.0.160,end=10.0.0.180 --disable-dhcp
```

vyatta plugin configuration

Edit Neutron configuration file `/etc/neutron/neutron.conf` and add the following line:

```
service_plugins =  
neutron.plugins.brocade.vyatta.vrouter_neutron_plugin.VyattaVRouterPlugin
```

Vyatta plugin configuration file should be provided to the neutron server during the startup. For devstack installation, update the file `stack-screenrc` in `devstack` directory with the highlighted configuration

```
stuff "cd /opt/stack/neutron && python /usr/local/bin/neutron-server --config-  
file /etc/neutron/neutron.conf --config-file  
/etc/neutron/plugins/ml2/ml2_conf.ini --config-file  
/etc/neutron/plugins/brocade/vyatta/vrouter.ini^M"
```

Edit the file `/etc/neutron/plugins/brocade/vyatta/vrouter.ini` to update `image-id` and `management network-id` values captured from the previous sections. Also update the values for the properties `keystone_url`, `tenant_id`, `tenant_admin_name` and `tenant_admin_password`.

Testing the functionality

Restart Openstack for the configurations to take effect. If you are using devstack, run the following commands:

```
./unstack.sh  
sudo ovs-vsctl add-br br-int  
./rejoin-stack.sh  
sudo service apache2 restart
```

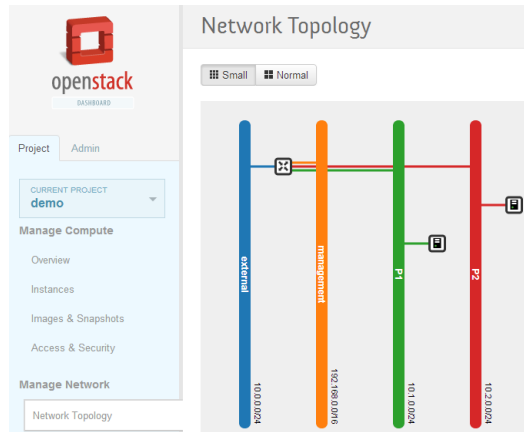
Create security group rules to allow ICMP, TCP, UDP and other protocol traffic between the VMs. Run the following three commands to add the UDP, TCP and ICMP rules to the 'default' security group.

```
nova secgroup-add-rule default udp 1 65535 0.0.0.0/0  
nova secgroup-add-rule default tcp 1 65535 0.0.0.0/0  
nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0
```

When a tenant router is created, Vyatta vRouter plugin will automatically create a vRouter VM instance in admin tenant and map it to the tenant router.

Add router interfaces to attach private networks to the router. Set external gateway to the tenant router using the configured external network. SNAT rules for the private network addresses are automatically created the plugin.

Assign Floating IPs (from the external network) to the VMs. SNAT and DNAT rules are automatically created by the plugin.



Here is the link to demo video: https://docs.google.com/file/d/0B_-dXko4DIFNNmZlXzj2N21KaG8/edit