Installation Runbook for

OpenStack® Deployment with SR-IOV Configuration

QLogic® 3400, 8400, and 45000 Series Adapters

Application Type	Single Root-I/O Virtualization
Application Version	
Mirantis® OpenStack Version	7.0/8.0
OpenStack Version	Kilo and Liberty

Contents

Document History

Version	Revision Date	Description
1.0	**05-24-16**	**Initial Version**

1 Introduction

The OpenStack® project is an open source cloud computing platform that supports all types of cloud environments and works as infrastructure as a service (IAAS).

The technology consists of a series of interrelated projects that control pools of processing, storage, and networking resources throughout a data center, which users manage through a web-based dashboard, command line tools, or representational state transfer (REST) APIs.

OpenStack (starting with Juno) adds inbox support to request virtual machine (VM) access to the virtual network through single root input/output virtualization (SR-IOV) NIC. With the introduction of SR-IOV-based NICs, each SR-IOV port is associated with virtual functions (VFs). SR-IOV ports are provided by hardware-based virtual Ethernet bridging (HW VEB).

1.1 Audience

This guide is intended for OpenStack (Kilo and Liberty) users who want to configure SR-IOV with the QLogic 3400/8400/45000 Series Adapters. You will need the following knowledge and equipment:

- One or more QLogic 3400/8400/45000 Series Adapters. For more information about these adapters, see the following links:

  3400/45000 Series Intelligent Ethernet Adapters
  http://www.qlogic.com/Products/adapters/Pages/IntelligentEthernetAdapters.aspx

  8400 Series Converged Network Adapters
  http://www.qlogic.com/Products/adapters/Pages/ConvergedNetworkAdapters.aspx

- An understanding of OpenStack deployment and experience with OpenStack 3 node configuration using flat mode networking with virtual LAN (VLAN) with the ML2/OVS plug-in for public and private (that is, net1) networks.

  More information about OpenStack 3 node configuration can be found at the following locations:

  OpenStack (Kilo):
  http://docs.openstack.org/kilo/install-guide/install/yum/content/

  OpenStack (Liberty):
  http://docs.openstack.org/liberty/install-guide-rdo/

- The compute node must have one or more QLogic 3400/8400/45000 Series Adapters present in system with SR-IOV support enabled.

- For the VM instance, QLogic uses a Red Hat 7 KVM Guest image to test SR-IOV configurations that support the 3400/8400/45000 Series Adapters' inbox driver or out-of-box driver, as described here:
  http://docs.openstack.org/image-guide/content/ch_obtaining_images.html

- OpenStack 3 node configuration without SR-IOV support must work with flat networking/VLAN. You must be able to assign floating IP access and log in to the VM.

2 Application Overview

Mirantis OpenStack is the orchestration solution that is commonly used in network data centers and has several components: compute, storage, dashboard, and networking. For networking connectivity under Neutron, 3400/8400/45000 Series Adapters support a comprehensive list of virtualization and multi-tenant services, including SR-IOV and virtual extensible LAN (VXLAN) offloads for the most sophisticated enterprise data centers, as well as private, public, and hybrid cloud deployments. With the Neutron networking component, QLogic offers several optimizations including VXLAN offloads and Open vSwitch (OVS) integration. The QLogic poll mode driver (PMD) allows the use of optimized networking for OpenStack using virtual network appliances running on the OpenStack Nova compute platform.

3 Joint Reference Architecture

This chapter describes the OpenStack 3 node architecture.

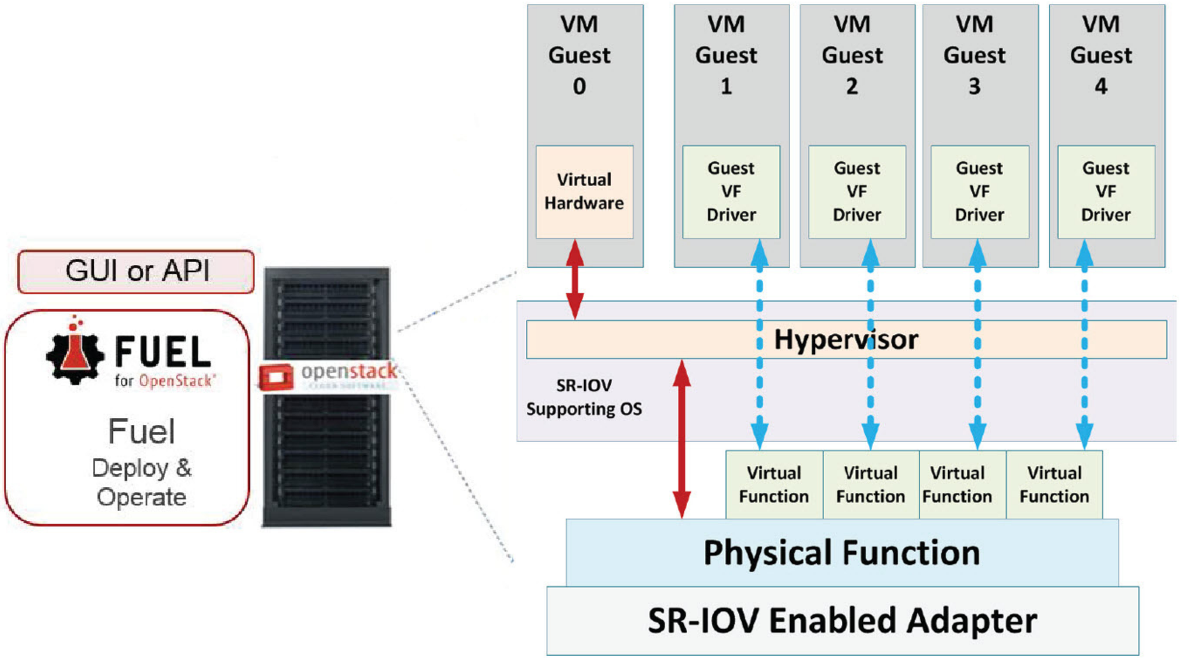


Figure 3-1. Mirantis OpenStack with SR-IOV

PCI-SIG, the special interest group that owns and manages PCI specifications as open industry standards, introduced a suite of specifications for SR-IOV to allow multiple operating systems to share a physical interconnect. The SR-IOV standard allows multiple virtual machines (VMs) to share an I/O device, while allowing for close to bare-metal runtime performance.

The SR-IOV specification details how a single PCI Express (PCIe) device can be shared between various guest operating systems—the VMs. Devices with SR-IOV support multiple virtual functions (VFs) on top of the physical function. VFs are enabled in hardware as a lightweight PCIe function that can be directly assigned to a VM without hypervisor mediation. These VFs operate in the context of a VM, and must be associated with a physical function (PF), a full-featured PCIe function that operates in the context of the hypervisor or parent partition.

SR-IOV provides direct VM connectivity and isolation across VMs. It allows the data to bypass the software virtual switch (vSwitch) and provides near-native performance. The benefits to deploying hardware-based SRIOV- enabled NICs include reduction of CPU and memory usage compared to vSwitches. Moving the network virtualization into hardware (SR-IOV adapters) relieves the performance problems associated with vSwitches. By directing VM I/O directly to VFs and bypassing the hypervisor, the vSwitches are no longer part of the data path. In addition, it significantly increases the number of virtual networking functions for a single physical server.

3.1 OpenStack Nodes

This guide is intended for use with OpenStack 3 node architecture, which consists of the following nodes:

- Controller
- Compute

These nodes are described in the following sections.

3.1.1 Controller Node

In the example architecture used in this guide, the Controller node runs the following services and functions:

- Identity service (Keystone)
- Image service (Glance)
- Management portions of the Compute service (Nova Management) and Networking service (Neutron Server/modular layer 2 (ML2) plug-in)
- Networking plug-in
- Dashboard (Horizon)
- Networking plug-in (ML2)
- Layer 3 agent
- DHCP agent

The architecture also includes supporting services such as a database (mysql), message broker (Rabbitmq), network time protocol (NTP), and external (Internet) connectivity for tenant VMs for the Compute node.

3.1.2 Compute Node

The Compute node runs the hypervisor portion of Compute, which operates tenant virtual machines. In this architecture, the Compute node uses a kernel-based virtual machine (KVM) as the hypervisor (KVM is the default hypervisor). The Compute node runs the Networking node plug-in (ML2), layer 2 agent open virtual search (OVS), and NIC switch agent (SR-IOV switch).

3.2 Physical & Logical Network Topology

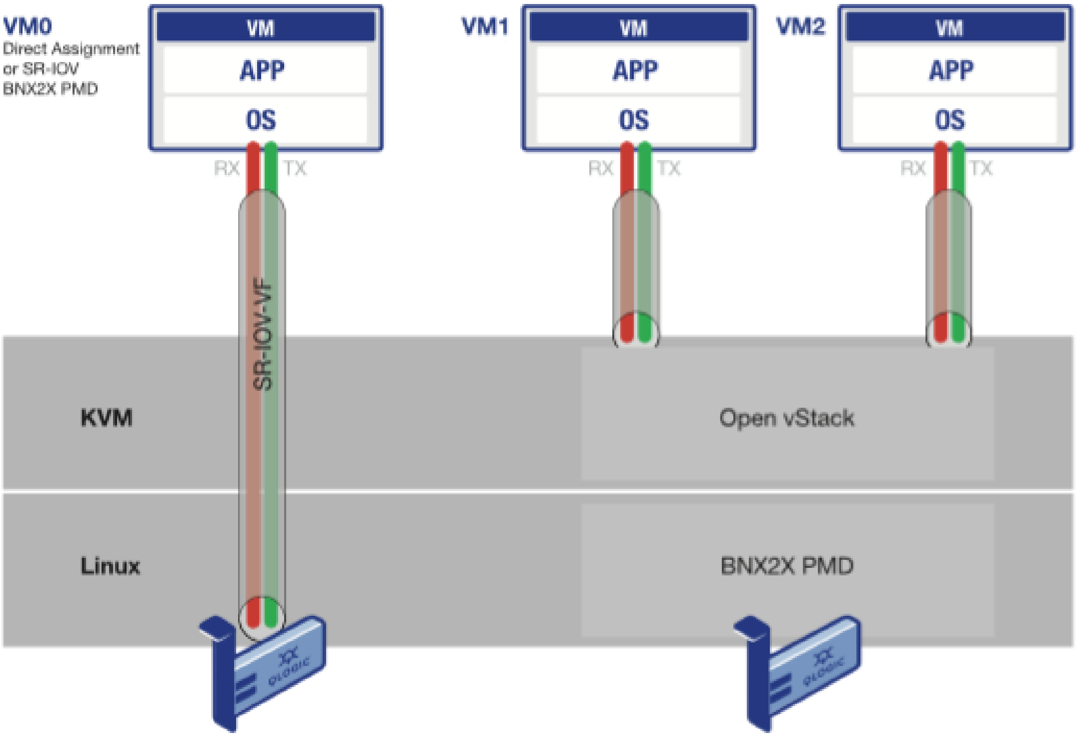


Figure 3-2. Physical Topology

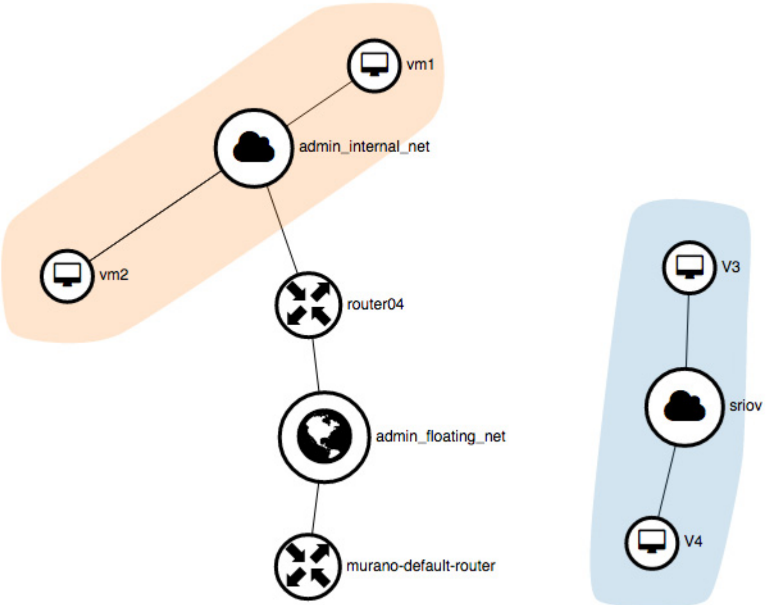


Figure 3-3. Logical Topology

4 Installation & Configuration

4.1 Environment Preparation

For information about preparing the Fuel environment, refer to the respective guides:

- For Fuel 7.0: https://docs.mirantis.com/openstack/fuel/fuel-7.0/user-guide.html#boot-the-fuel-master-node
- For Fuel 8.0 http://docs.openstack.org/developer/fuel-docs/userdocs/fuel-install-guide.html

MOS 7.0/8.0 ISO are available at https://www.mirantis.com/products/mirantis-openstack-software/

4.1.1 Nodes

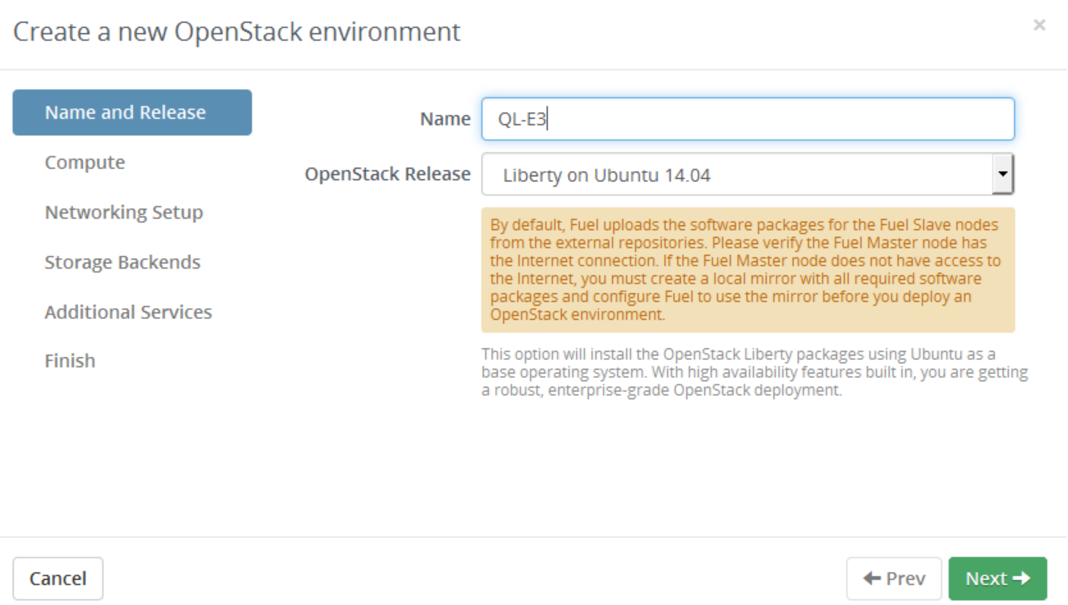
Number of nodes: 1 Controller/Compute/Storage

Hypervisor: KVM

Networking segmentation type: Neutron VLAN

To configure MOS:

1. Select **Name and Release**. Specify the OpenStack name (QL-E3)and release (Liberty on Ubuntu 14.04), and then click **Next**.



Create a new OpenStack environment

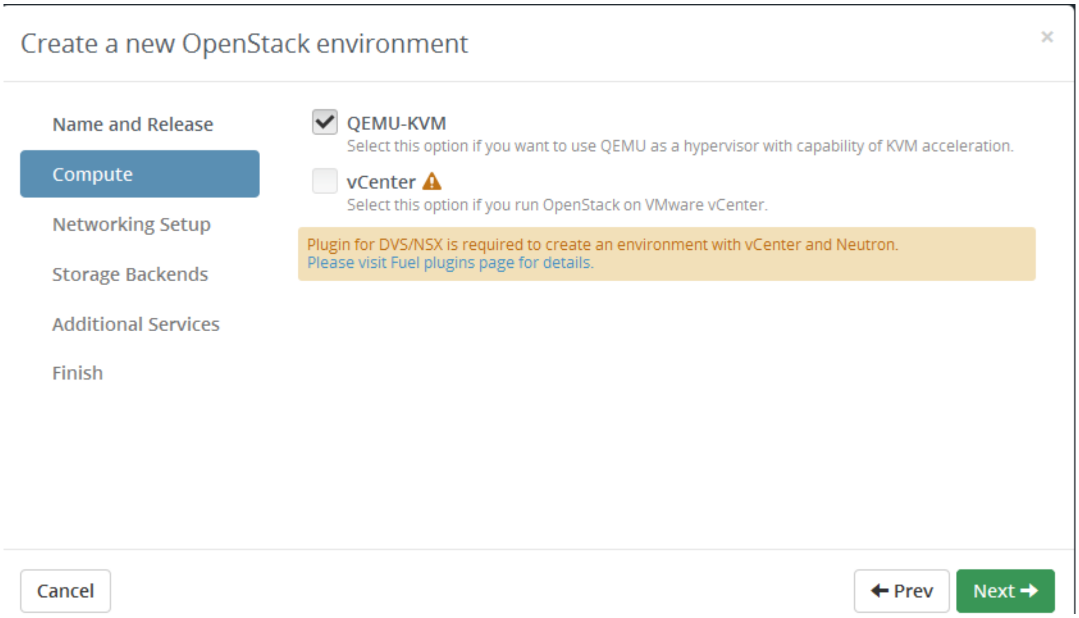
Name: QL-E3

OpenStack Release: Liberty on Ubuntu 14.04

By default, Fuel uploads the software packages for the Fuel Slave nodes from the external repositories. Please verify the Fuel Master node has the Internet connection. If the Fuel Master node does not have access to the Internet, you must create a local mirror with all required software packages and configure Fuel to use the mirror before you deploy an OpenStack environment.

This option will install the OpenStack Liberty packages using Ubuntu as a base operating system. With high availability features built in, you are getting a robust, enterprise-grade OpenStack deployment.

2. Select **Compute**. Select QEMU-KVM to use QEMU as the hypervisor with capability of KVM acceleration, and then click **Next**.



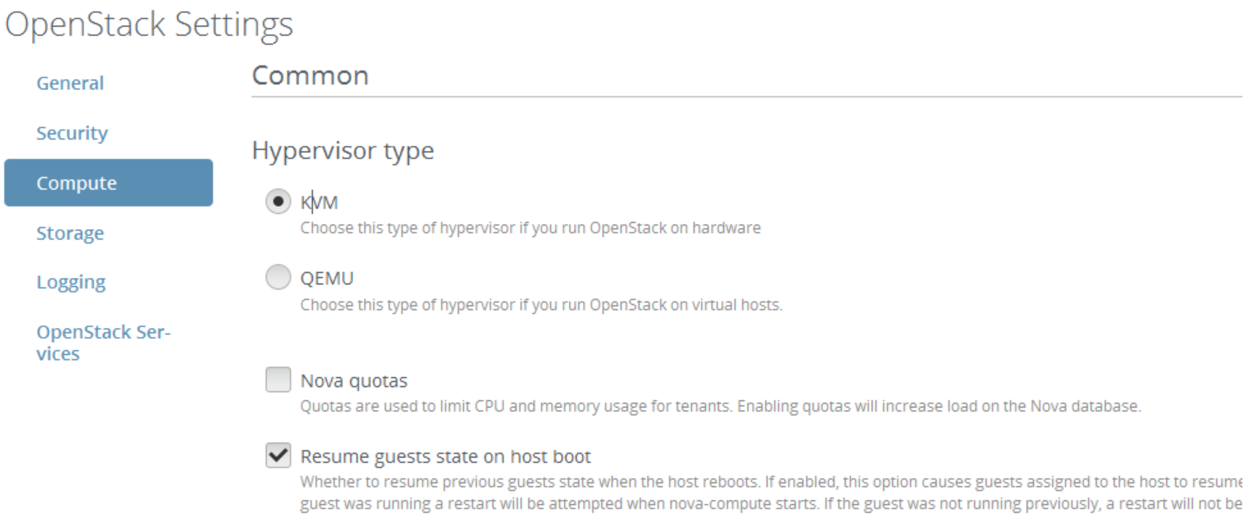
Create a new OpenStack environment

☑ QEMU-KVM
Select this option if you want to use QEMU as a hypervisor with capability of KVM acceleration.

□ vCenter ⚠
Select this option if you run OpenStack on VMware vCenter.

Plugin for DVS/NSX is required to create an environment with vCenter and Neutron.
Please visit Fuel plugins page for details.

For MOS 8.0, under Hypervisor, select **KVM** and check **Resume guests state on host boot**. Click **Next**.



OpenStack Settings

Common

Hypervisor type

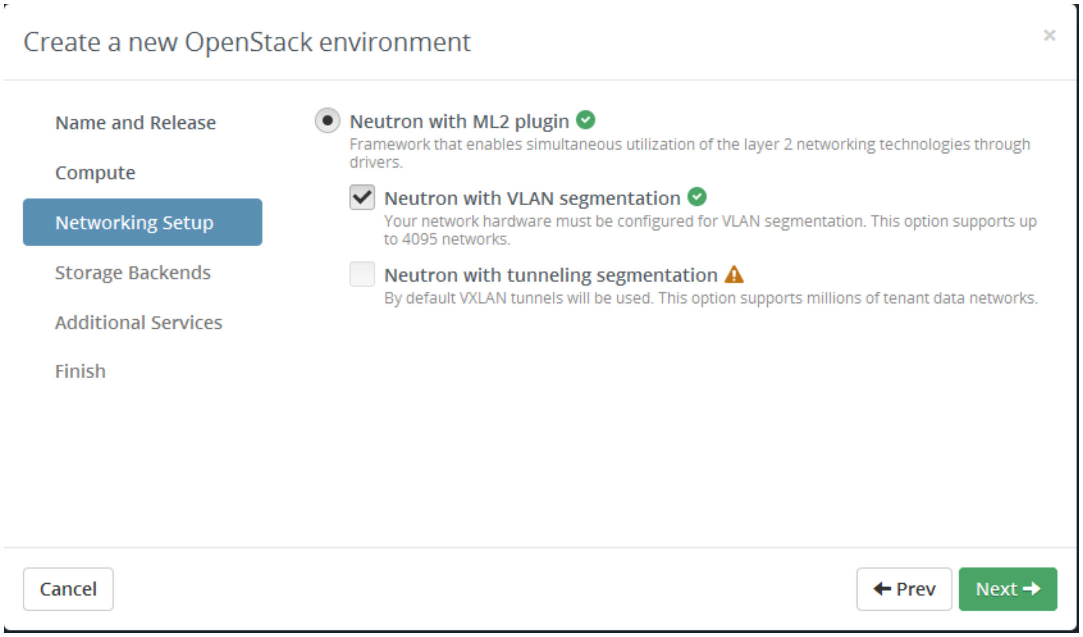
● KVM
Choose this type of hypervisor if you run OpenStack on hardware

○ QEMU
Choose this type of hypervisor if you run OpenStack on virtual hosts.

□ Nova quotas
Quotas are used to limit CPU and memory usage for tenants. Enabling quotas will increase load on the Nova database.

☑ Resume guests state on host boot
Whether to resume previous guests state when the host reboots. If enabled, this option causes guests assigned to the host to resume guest was running a restart will be attempted when nova-compute starts. If the guest was not running previously, a restart will not be

3. Select **Networking Setup**. Select **ML2 Plugin with Neutron**, and **Neutron with VLAN segmentation**, and then click **Next**.

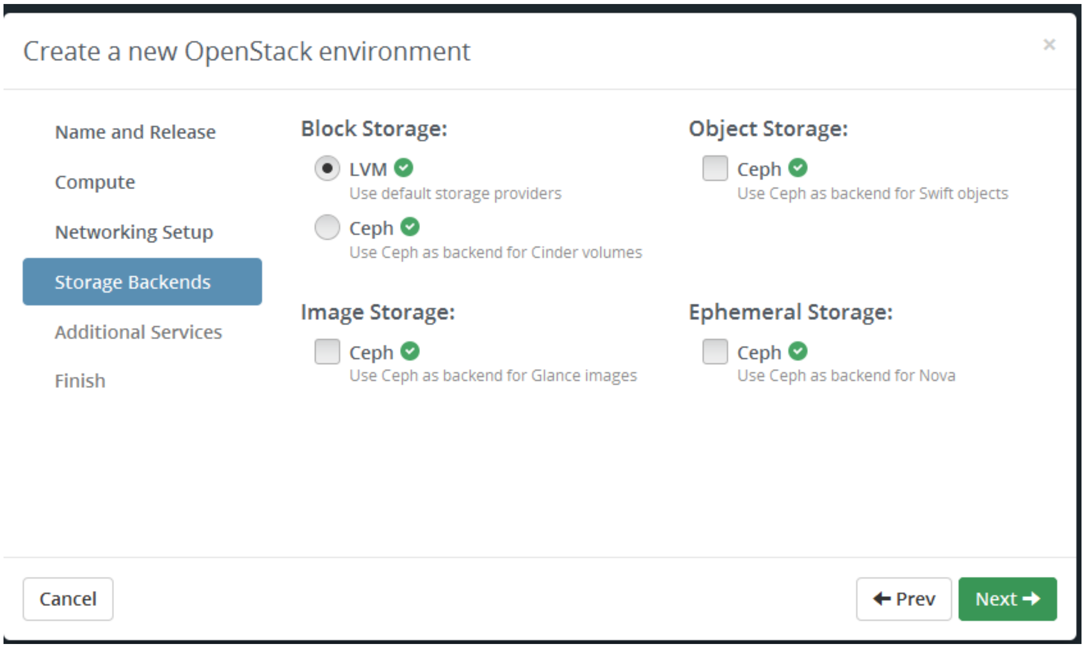


● Neutron with ML2 plugin ✓
Framework that enables simultaneous utilization of the layer 2 networking technologies through drivers.

☑ Neutron with VLAN segmentation ✓
Your network hardware must be configured for VLAN segmentation. This option supports up to 4095 networks.

□ Neutron with tunneling segmentation ⚠
By default VXLAN tunnels will be used. This option supports millions of tenant data networks.

4. Select **Storage Backends**. Under Block Storage, select **LVM**, and then click **Next**.



Block Storage:

● LVM ✓
Use default storage providers

○ Ceph ✓
Use Ceph as backend for Cinder volumes

Object Storage:

□ Ceph ✓
Use Ceph as backend for Swift objects

Image Storage:

□ Ceph ✓
Use Ceph as backend for Glance images

Ephemeral Storage:

□ Ceph ✓
Use Ceph as backend for Nova

4.1.2 MOS installation

For information about installing MOS, refer to the respective Quick Start Guides:

- For Fuel 7.0: https://docs.mirantis.com/openstack/fuel/fuel-7.0/#guides
- For Fuel 8.0: https://docs.mirantis.com/openstack/fuel/fuel-8.0/

4.1.3 Health Check Results

For information about performing health checks, refer to the respective user's guides:

- For Fuel 7.0, under *Post-Deployment Check*: https://docs.mirantis.com/openstack/fuel/fuel-7.0/user-guide.html#post-deployment-check
- For Fuel 8.0, under *Verify your OpenStack environment*: https://docs.mirantis.com/openstack/fuel/fuel-8.0/pdf/Fuel-8.0-UserGuide.pdf

The deployment should pass for basic sanity test, functional test, and HA test if the controllers are configured in HA mode.

4.2 SR-IOV Configuration Using the 3400/8400/45000 Series Adapters

This section describes how to configure SR-IOV using the QLogic 3400/8400/45000 Series Adapters, as described in the following sections:

- Enabling SR-IOV on the 3400/8400/45000 Series Adapters (Compute Node)
- Configuring SR-IOV in the Controller Node
- Configuring SR-IOV on the Compute Node

4.2.1 Enabling SR-IOV on the 3400/8400/45000 Series Adapters (Compute Node)

To enable SR-IOV on the 3400/8400/45000 Series Adapters in the Compute node:

1. Ensure that one or more 3400/8400/45000 Series Adapters with SR-IOV support are present on the Compute node.
2. Install the adapter driver on the Compute node.
   - For 3400/8400 Series Adapters install the bnx2x NIC driver.
   - For 45000 Series Adapters, install the qed/qede drivers.

   To download the most recent drivers, go to driverdownloads.qlogic.com.
3. Load the NIC drivers by issuing the following commands for your adapter model.

   3400/8400 Series Adapters:

```
modprobe bnx2x
```

   45000 Series Adapters:

```
modprobe qed
modprobe qede
```

4. Ensure that the port is up by issuing the following command:

```
ifup ethx / ifconfig ethx up
```

5. To create eight VFs, issue the following command:

```
echo 8 > /sys/bus/pci/devices/0000\:0x\:00.x/sriov_numvfs
```

6. Verify that the virtual interfaces are visible to host operating system by issuing the following command:

```
lspci |grep Qlogic
```

7. Enable SR-IOV on the port where the tenant network has been defined.

4.2.2 Configuring SR-IOV in the Controller Node

Since the Neutron server is running on the Controller node, the file configuration for the Neutron server must be changed to enable SR-IOV, as described in the following procedure.

To configure SR-IOV in the Controller node:

1. Modify the ML2 Neutron plug-in by editing the `/etc/neutron/plugins/ml2/ml2_conf.ini` file on Controller node as follows:

```
[ml2]
tenant_network_types = flat,vlan
type_drivers = vlan
mechanism_drivers = openvswitch,sriovnicswitch

[ml2_type_vlan]
network_vlan_ranges = physnet2:1000:2000,physnet3:2001:2100
```

2. Add the supported PCI vendor VF devices defined by <*device_id*>:<*vendor_id*> according to the PCI ID Repository and make agent_require=False (MOS 7.0) or agent_require=True (MOS 8.0) by editing the `/etc/neutron/plugins/ml2/ml2_conf_sriov.ini` file as follows:

```
[ml2_sriov]
supported_pci_vendor_devs = <device_id>:<vendor_id>
agent_require = False (MOS 7.0)
```

or

```
agent_require = True (MOS 8)
```

3. Add `scheduler_default_filters` and `scheduler_available_filters` to the `/etc/nova/nova.conf` file as follows:

```
scheduler_default_filters=RetryFilter,AvailabilityZoneFilter,RamFilter,CoreFilter,DiskFilter,ComputeFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter,
ServerGroupAntiAffinityFilter,ServerGroupAffinityFilter,PciPassthroughFilter

scheduler_available_filters=nova.scheduler.filters.all_filters
```

4. Change the Neutron server configuration.

   The Neutron server must be run with the following configuration file:

   `/etc/neutron/plugins/ml2/ml2_conf_sriov.ini`

   a. Navigate to the configuration files' location by issuing the following command:

```
# cd /etc/init
```

   b. Edit the neutron-server.conf file and add the `ml2_conf_sriov.ini` configuration file as follows:

```
exec start-stop-daemon --start --chuid neutron:neutron --exec /usr/bin/neutron-server --\

--config-file /etc/neutron/neutron.conf \

--log-file /var/log/neutron/server.log $CONF_ARG --config-file
/etc/neutron/plugins/ml2/ml2_conf_sriov.ini
```

5. Restart the Neutron server by issuing the following commands:

```
# service neutron-server restart
```

4.2.3 Configuring SR-IOV on the Compute Node

To configure SR-IOV on the Compute node, associate the available VF with the physical network as follows:

1. Enable intel_iommu on the Compute node by adding the following line to the `/etc/default/grub` file:

   `GRUB_CMDLINE_LINUX_DEFAULT="nomdmonddf nomdmonisw intel_iommu=on"`

   Reboot the system and then examine the `cat/proc/cmdline` file to confirm that intel_iommu is enabled (intel_iommu=on).

2. For MOS 8.0, the Neutron agent is running on the Compute node and `agent_require=True` in the Controller node. Therefore, add the following to the `sriov_agent.ini` file:

```
[sriov_nic]

physical_device_mappings = physnet3:p5p2


[securitygroup]

firewall_driver = neutron.agent.firewall.NoopFirewallDriver
```

3. In MOS 8.0 there is an issue with the Libvirt/SRIOV configuration while creating a VM. Add the following entries to the `/etc/apparmor.d/abstractions/libvirt-qemu` file in the Compute node.

```
# Those rules are required for SR-IOV to function properly
/sys/devices/system/** r,
/sys/bus/pci/devices/ r,
/sys/bus/pci/devices/** r,
/sys/devices/pci*/** rw,
/{,var/}run/openvswitch/vhu* rw,
```

4. Add `pci_passthrough_whitelist` to the `/etc/nova/nova.conf` file as follows:

```
pci_passthrough_whitelist={"devname": "p5p2", "physical_network":"physnet3"}
```

5. Restart the OpenStack Nova service by issuing the following command:

```
# service nova-compute restart
```

6. For MOS 8.0 only, install the SR-IOV Neutron agent on the Compute node by issuing the following command:

```
# apt-get install neutron-plugin-sriov-agent
```

7. Restart the Neutron Open vSwitch agent and SR-IOV NIC agent services by issuing the following commands:

   For MOS 7.0 and 8.0:

```
# service neutron-plugin-openvswitch-agent restart
```

   For MOS 8.0 only:

```
# service neutron-plugin-sriov-agent restart
```

4.3 Configuration Deployment with SR-IOV

This section describes how to deploy OpenStack with SR-IOV, as follows:

- Creating an SR-IOV Network
- Creating the Subnet “subnet2” for the SR-IOV Network
- Creating the SR-IOV Port
- Creating a VM with the SR-IOV Network
- Quality of Service Policy Updates for MOS 8.0

4.3.1 Creating an SR-IOV Network

To create an SR-IOV network from the Controller node:

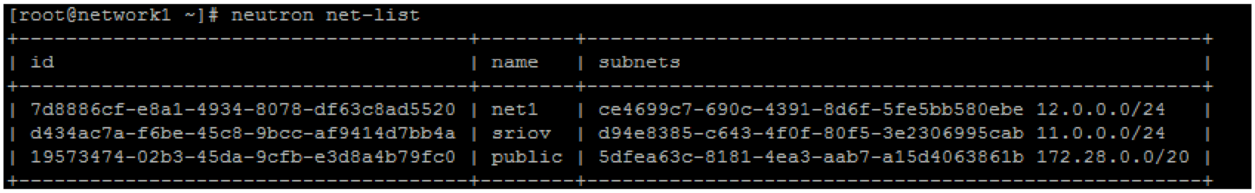
1. Create an SR-IOV network with a VLAN network type by issuing the following command:

```
# neutron net-create --provider:physical_network=physnet3 --provider:network_type=vlan sriov
```

2. Issue the following command to view the configuration for OpenStack 3 nodes:

```
# [root@network1 ~]# neutron net-list
```

Following is a sample output.



```
[root@network1 ~]# neutron net-list
+--------------------------------------+--------+----------------------------------------------------+
| id                                   | name   | subnets                                            |
+--------------------------------------+--------+----------------------------------------------------+
| 7d8886cf-e8a1-4934-8078-df63c8ad5520 | net1   | ce4699c7-690c-4391-8d6f-5fe5bb580ebe 12.0.0.0/24   |
| d434ac7a-f6be-45c8-9bcc-af9414d7bb4a | sriov  | d94e8385-c643-4f0f-80f5-3e2306995cab 11.0.0.0/24   |
| 19573474-02b3-45da-9cfb-e3d8a4b79fc0 | public | 5dfea63c-8181-4ea3-aab7-a15d4063861b 172.28.0.0/20 |
+--------------------------------------+--------+----------------------------------------------------+
```

The values in the `name` column are described in the following paragraphs.

net1 A private network that provides internal network access for instances of the Compute node.

Public and private networks are configured in the prerequisite section.

sriov An SR-IOV network for VM access to a virtual network using SR-IOV NIC.

public An external network that provides Internet access for instances of the Compute node using a network address translation (NAT) /floating IP address and a qualified security group.

4.3.2 Creating the Subnet “subnet2” for the SR-IOV Network

To create the subnet2 subnet for the SR-IOV network:

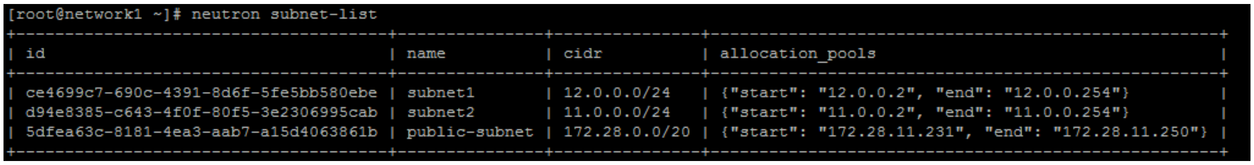
1. Create a subnet attached to the SR-IOV network by issuing the following command:

```
# neutron subnet-create sriov 11.0.0.0/24 --name subnet2 --dns-nameserver 8.8.4.4 --gateway 11.0.0.1
```

2. Issue the following command to view the subnet network:

```
# neutron subnet-list
```

Following is a sample output.



```
[root@network1 ~]# neutron subnet-list
+--------------------------------------+---------------+---------------+----------------------------------------------------+
| id                                   | name          | cidr          | allocation_pools                                   |
+--------------------------------------+---------------+---------------+----------------------------------------------------+
| ce4699c7-690c-4391-8d6f-5fe5bb580ebe | subnet1       | 12.0.0.0/24   | {"start": "12.0.0.2", "end": "12.0.0.254"}         |
| d94e8385-c643-4f0f-80f5-3e2306995cab | subnet2       | 11.0.0.0/24   | {"start": "11.0.0.2", "end": "11.0.0.254"}         |
| 5dfea63c-8181-4ea3-aab7-a15d4063861b | public-subnet | 172.28.0.0/20 | {"start": "172.28.11.231", "end": "172.28.11.250"} |
+--------------------------------------+---------------+---------------+----------------------------------------------------+
```

The values in the `name` column are described in the following paragraphs.

subnet1	A private subnet that uses DHCP Private subnets are configured in the prerequisite section.
subnet2	An SRI-OV subnet
public-subnet	A subnet for external connectivity

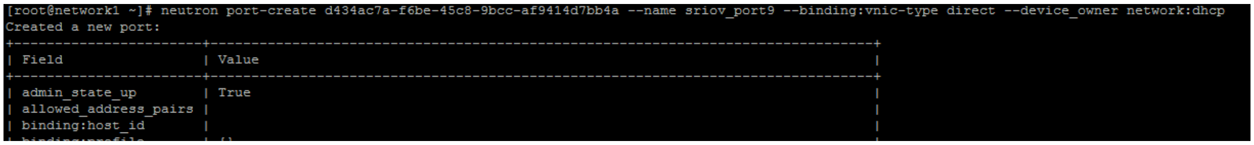
4.3.3 Creating the SR-IOV Port

To create the SR-IOV port:

1. Issue the following command to create an SR-IOV port:

```
#neutron port-create <sriov net id>  --name  <port_name>  --binding:vnic-type direct --device_owner network:dhcp
```

Following is a sample output.

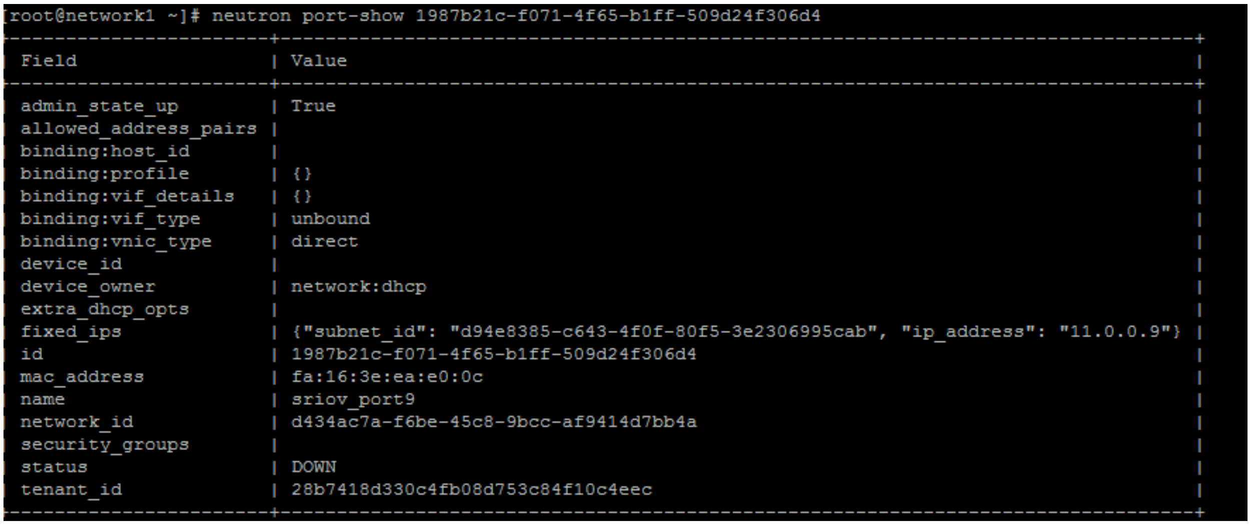


```
[root@network1 ~]# neutron port-create d434ac7a-f6be-45c8-9bcc-af9414d7bb4a --name sriov_port9 --binding:vnic-type direct --device_owner network:dhcp
Created a new port:
+-----------------------+-----------------------------------------------------------------------------+
| Field                 | Value                                                                       |
+-----------------------+-----------------------------------------------------------------------------+
| admin_state_up        | True                                                                        |
| allowed_address_pairs |                                                                             |
| binding:host_id       |                                                                             |
| binding:profile       | {}                                                                          |
```

2. To show the port information, issue the following command:

# neutron port-show *<ID>*

Following is a sample output. Make sure the output shows the correct `vif_type` and `vnic_type`.



```
[root@network1 ~]# neutron port-show 1987b21c-f071-4f65-b1ff-509d24f306d4
-----------------------+----------------------------------------------------------------------------------+
 Field                 | Value                                                                            |
-----------------------+----------------------------------------------------------------------------------+
 admin_state_up        | True                                                                             |
 allowed_address_pairs |                                                                                  |
 binding:host_id       |                                                                                  |
 binding:profile       | {}                                                                               |
 binding:vif_details   | {}                                                                               |
 binding:vif_type      | unbound                                                                          |
 binding:vnic_type     | direct                                                                           |
 device_id             |                                                                                  |
 device_owner          | network:dhcp                                                                     |
 extra_dhcp_opts       |                                                                                  |
 fixed_ips             | {"subnet_id": "d94e8385-c643-4f0f-80f5-3e2306995cab", "ip_address": "11.0.0.9"} |
 id                    | 1987b21c-f071-4f65-b1ff-509d24f306d4                                             |
 mac_address           | fa:16:3e:ea:e0:0c                                                                |
 name                  | sriov_port9                                                                      |
 network_id            | d434ac7a-f6be-45c8-9bcc-af9414d7bb4a                                             |
 security_groups       |                                                                                  |
 status                | DOWN                                                                             |
 tenant_id             | 28b7418d330c4fb08d753c84f10c4eec                                                 |
-----------------------+----------------------------------------------------------------------------------+
```

4.3.4 Creating a VM with the SR-IOV Network

This section describes how to create a VM with multiple networks: a private network for a floating IP address and an SR-IOV network with direct VM access using a VF.

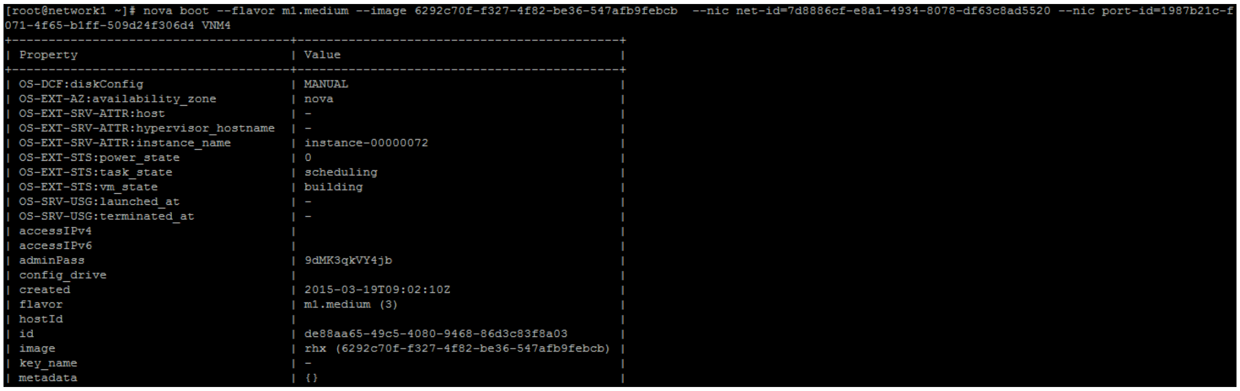
To create a VM with the SR-IOV network:

1. Issue the following command to create a VM:

nova boot --flavor m1.medium --image *<image-id>* --nic net-id=*<net id of private network net1>* --nic port-id=*<port id of sriov network from port-create command in* Configuring SR-IOV on the Compute Node> *<VM name>*

In this command, a `–nic net-id` is given for the private network (`net1`) and a `–nic port-id` is given for the SR-IOV network with a Red Hat 7 KVM guest image.

Following is a sample output.

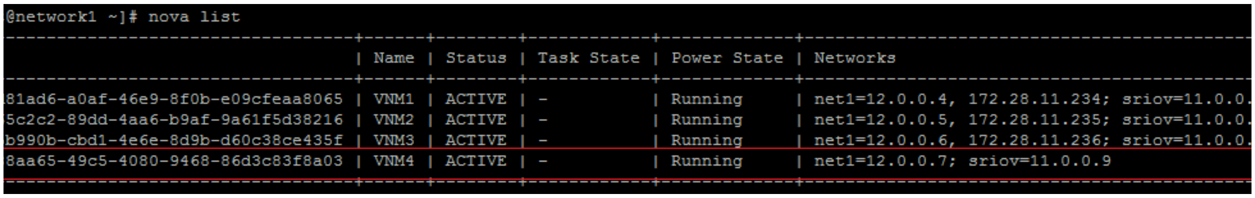


```
[root@network1 ~]# nova boot --flavor m1.medium --image 6292c70f-f327-4f82-be36-547afb9febcb  --nic net-id=7d8886cf-e8a1-4934-8078-df63c8ad5520 --nic port-id=1987b21c-f
071-4f65-b1ff-509d24f306d4 VNM4
+-------------------------------------+--------------------------------------------+
| Property                            | Value                                      |
+-------------------------------------+--------------------------------------------+
| OS-DCF:diskConfig                   | MANUAL                                     |
| OS-EXT-AZ:availability_zone         | nova                                       |
| OS-EXT-SRV-ATTR:host                | -                                          |
| OS-EXT-SRV-ATTR:hypervisor_hostname | -                                          |
| OS-EXT-SRV-ATTR:instance_name       | instance-00000072                          |
| OS-EXT-STS:power_state              | 0                                          |
| OS-EXT-STS:task_state               | scheduling                                 |
| OS-EXT-STS:vm_state                 | building                                   |
| OS-SRV-USG:launched_at              | -                                          |
| OS-SRV-USG:terminated_at            | -                                          |
| accessIPv4                          |                                            |
| accessIPv6                          |                                            |
| adminPass                           | 9dMK3qkVY4jb                               |
| config_drive                        |                                            |
| created                             | 2015-03-19T09:02:10Z                       |
| flavor                              | m1.medium (3)                              |
| hostId                              |                                            |
| id                                  | de88aa65-49c5-4080-9468-86d3c83f8a03       |
| image                               | rhx (6292c70f-f327-4f82-be36-547afb9febcb) |
| key_name                            | -                                          |
| metadata                            | {}                                         |
```

2. Issue the following command to ensure that the VM is up and running:

```
#nova list
```

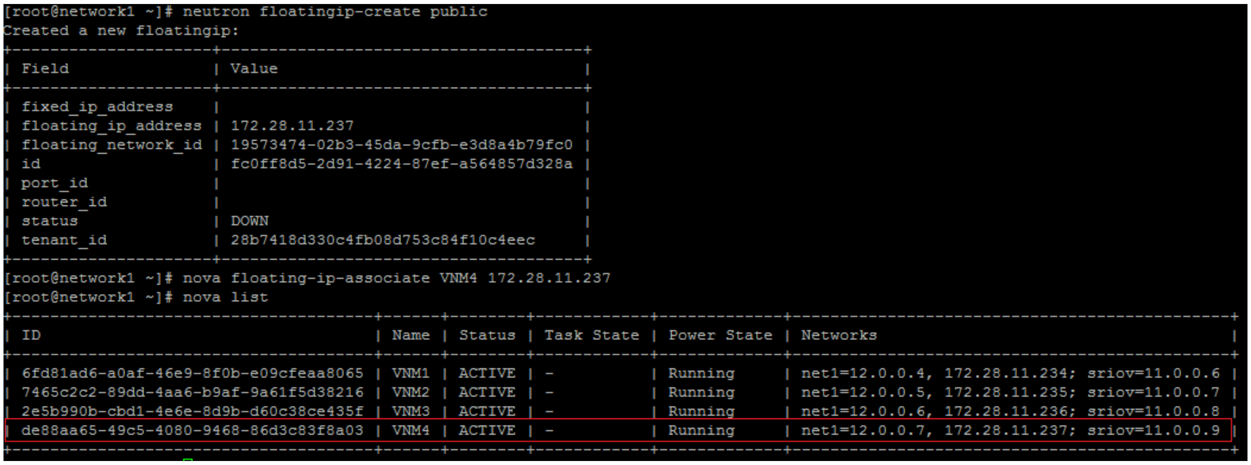
Following is a sample output.



```
@network1 ~]# nova list
----------------------------------+------+--------+------------+-------------+------------------------------------------------
                                  | Name | Status | Task State | Power State | Networks
----------------------------------+------+--------+------------+-------------+------------------------------------------------
81ad6-a0af-46e9-8f0b-e09cfeaa8065 | VNM1 | ACTIVE | -          | Running     | net1=12.0.0.4, 172.28.11.234; sriov=11.0.0.
5c2c2-89dd-4aa6-b9af-9a61f5d38216 | VNM2 | ACTIVE | -          | Running     | net1=12.0.0.5, 172.28.11.235; sriov=11.0.0.
b990b-cbd1-4e6e-8d9b-d60c38ce435f | VNM3 | ACTIVE | -          | Running     | net1=12.0.0.6, 172.28.11.236; sriov=11.0.0.
8aa65-49c5-4080-9468-86d3c83f8a03 | VNM4 | ACTIVE | -          | Running     | net1=12.0.0.7; sriov=11.0.0.9
----------------------------------+------+--------+------------+-------------+------------------------------------------------
```

3. Assign a floating IP address to the private network to access using ssh.

   Following is a sample output.



```
[root@network1 ~]# neutron floatingip-create public
Created a new floatingip:
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    |                                      |
| floating_ip_address | 172.28.11.237                        |
| floating_network_id | 19573474-02b3-45da-9cfb-e3d8a4b79fc0 |
| id                  | fc0ff8d5-2d91-4224-87ef-a564857d328a |
| port_id             |                                      |
| router_id           |                                      |
| status              | DOWN                                 |
| tenant_id           | 28b7418d330c4fb08d753c84f10c4eec     |
+---------------------+--------------------------------------+
[root@network1 ~]# nova floating-ip-associate VNM4 172.28.11.237
[root@network1 ~]# nova list
+--------------------------------------+------+--------+------------+-------------+-----------------------------------------------+
| ID                                   | Name | Status | Task State | Power State | Networks                                      |
+--------------------------------------+------+--------+------------+-------------+-----------------------------------------------+
| 6fd81ad6-a0af-46e9-8f0b-e09cfeaa8065 | VNM1 | ACTIVE | -          | Running     | net1=12.0.0.4, 172.28.11.234; sriov=11.0.0.6 |
| 7465c2c2-89dd-4aa6-b9af-9a61f5d38216 | VNM2 | ACTIVE | -          | Running     | net1=12.0.0.5, 172.28.11.235; sriov=11.0.0.7 |
| 2e5b990b-cbd1-4e6e-8d9b-d60c38ce435f | VNM3 | ACTIVE | -          | Running     | net1=12.0.0.6, 172.28.11.236; sriov=11.0.0.8 |
| de88aa65-49c5-4080-9468-86d3c83f8a03 | VNM4 | ACTIVE | -          | Running     | net1=12.0.0.7, 172.28.11.237; sriov=11.0.0.9 |
+--------------------------------------+------+--------+------------+-------------+-----------------------------------------------+
```

4.3.5 Quality of Service Policy Updates for MOS 8.0

To update quality of service (QoS) on the Controller node and the Compute node for MOS 8.0:

1. Add QoS to the Controller node by editing the `/etc/neutron/neutron.conf` file and adding the following line:

```
service_plugins =neutron.services.l3_router.l3_router_plugin.L3RouterPlugin,neutron.services.
metering.metering_plugin.MeteringPlugin,qo
```

2. n the [qos] section of the `/etc/neutron/neutron.conf` file, confirm that `notification_drivers = message_queue`.

```
[qos]
# Drivers list to use to send the update notification
# notification_drivers = message_queue
notification_drivers = message_queue
```

3. In [ml2] section of the `/etc/neutron/plugins/ml2/ml2_conf.ini`, confirm that `qos` is included for the `extension_drivers`.

```
extension_drivers =port_security,qos
```

4. Type the following command to complete the Controller node settings:

```
service neutron-server restart
```

5. In the `/etc/neutron/plugins/ml2/sriov_agent.ini` file, confirm the Compute node settings by ensuring that `extensions = qos`:

```
[agent]
# (ListOpt) Extensions list to use
# Example: extensions = qos
# extensions = qos
```

6. Type the following command to complete the Compute node settings:

```
service neutron-plugin-sriov-agent restart
```

7. Edit the `/etc/neutron/policy.json` file to include the following lines to allow regular users to change bandwidth on the Controller node:

```
"get_policy": "rule:regular_user",
"create_policy": "rule:regular_user",
"update_policy": "rule:regular_user",
"delete_policy": "rule:regular_user",
"get_policy_bandwidth_limit_rule": "rule:regular_user",
"create_policy_bandwidth_limit_rule": "rule:regular_user",
"delete_policy_bandwidth_limit_rule": "rule:regular_user",
"update_policy_bandwidth_limit_rule": "rule:regular_user",
"get_rule_type": "rule:regular_user",
```

8. You now want to confirm that QoS is working: type the following command to create a QoS policy:

```
neutron qos-policy-create bw-limiter
```

9. Type the following command to create a bandwidth setting rule:

```
neutron qos-bandwidth-limit-rule-create bw-limiter --max-kbps 7000000 --max-burst-kbps 700000
```

10. Type the following commands to associate the QoS bandwidth policy with existing Neutron ports:

```
neutron port-update 5056e2f1-025d-46fc-9131-959ae708fe40 --qos-policy bw-limiter
```

```
neutron port-update 98148013-d210-4922-a4ae-fa92c57c2cd3 --qos-policy bw-limiter
```

11. Run traffic between two VMs over the SRIOV-VFS port. You should observe that the bandwidth is less than the specified limit (7,000,000kbps).

    To modify the bandwidth rule, use the following command format:

```
neutron qos-bandwidth-limit-rule-update BANDWIDTH_LIMIT_RULE QOS_POLICY
[--max-kbps MAX_KBPS] [--max-burst-kbps MAX_BURST_KBPS]
```

    For example:

```
neutron qos-bandwidth-limit-rule-update 9914f100-9e0e-48dc-b964-b2afa47332d8 bw-limiter
--max-kbps 3000000 --max-burst-kbps 3000000
```

4.4 Troubleshooting

Should you encounter problems with your configuration, consider the following:

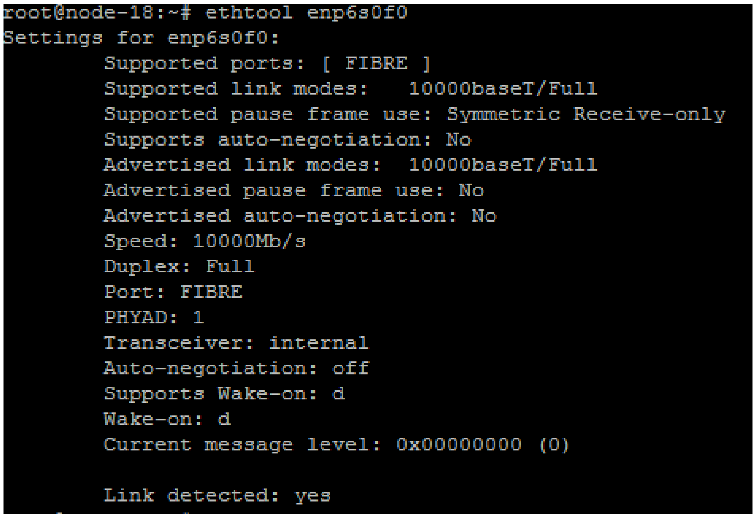
- Compute Node—QLogic NIC Adapter Connection
- Compute Node—SR-IOV VFs
- Controller Node—Neutron Server
- Server BIOS—Enable SR-IOV
- Controller Node—Neutron Server
- Controller Node—Nova Services
- Controller Node—Neutron Agents
- SR-IOV VF Connectivity

For adapter related issues, see the troubleshooting chapters in the following user's guides available at driverdownloads.qlogic.com:

- *QLogic FastLinQ 8400/3400 Series Converged Network Adapters and Intelligent Ethernet Adapters User's Guide*
- *QLogic 25Gb/40Gb/100Gb FastLinQ 45000 Series Intelligent Ethernet Adapters User's Guide*

4.4.1 Compute Node—QLogic NIC Adapter Connection

To confirm that the Compute node QLogic adapter is connected to the switch network, use the `ethtool ethx` command, as shown:

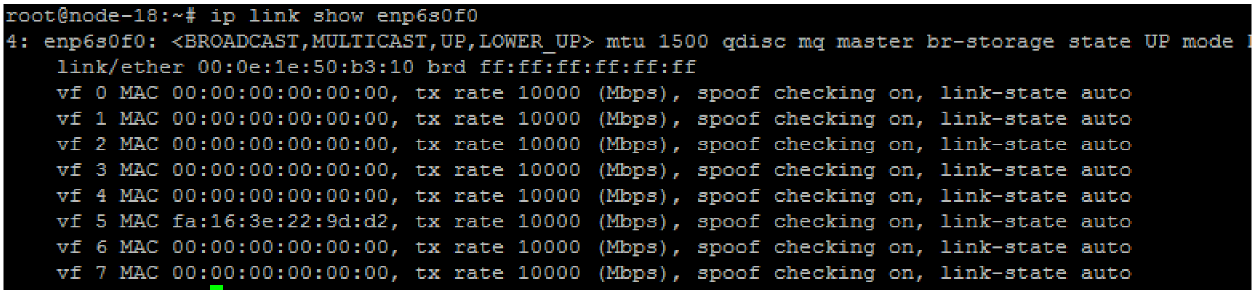


```
root@node-18:~# ethtool enp6s0f0
Settings for enp6s0f0:
        Supported ports: [ FIBRE ]
        Supported link modes:   10000baseT/Full
        Supported pause frame use: Symmetric Receive-only
        Supports auto-negotiation: No
        Advertised link modes:  10000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: No
        Speed: 10000Mb/s
        Duplex: Full
        Port: FIBRE
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: off
        Supports Wake-on: d
        Wake-on: d
        Current message level: 0x00000000 (0)

        Link detected: yes
```

4.4.2 Compute Node—SR-IOV VFs

To confirm that the SR-IOV VFs have been created on the Compute node correctly, use the `ip link show` command, as shown:



```
root@node-18:~# ip link show enp6s0f0
4: enp6s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master br-storage state UP mode
    link/ether 00:0e:1e:50:b3:10 brd ff:ff:ff:ff:ff:ff
    vf 0 MAC 00:00:00:00:00:00, tx rate 10000 (Mbps), spoof checking on, link-state auto
    vf 1 MAC 00:00:00:00:00:00, tx rate 10000 (Mbps), spoof checking on, link-state auto
    vf 2 MAC 00:00:00:00:00:00, tx rate 10000 (Mbps), spoof checking on, link-state auto
    vf 3 MAC 00:00:00:00:00:00, tx rate 10000 (Mbps), spoof checking on, link-state auto
    vf 4 MAC 00:00:00:00:00:00, tx rate 10000 (Mbps), spoof checking on, link-state auto
    vf 5 MAC fa:16:3e:22:9d:d2, tx rate 10000 (Mbps), spoof checking on, link-state auto
    vf 6 MAC 00:00:00:00:00:00, tx rate 10000 (Mbps), spoof checking on, link-state auto
    vf 7 MAC 00:00:00:00:00:00, tx rate 10000 (Mbps), spoof checking on, link-state auto
```

4.4.3 Compute Node—Enable intel_iommu

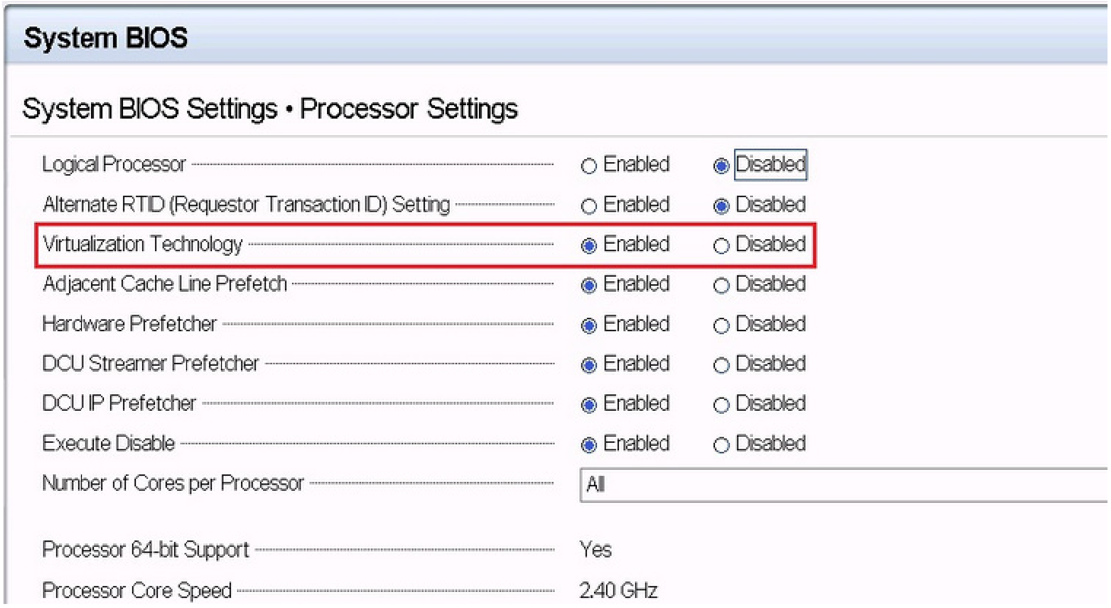
On the Compute node only, confirm that intel_iommu is enabled (`intel_iommu=on`) by examining the `cat/proc/cmdline` file, as shown:

```
# cat /proc/cmdline
```

```
BOOT_IMAGE=/vmlinuz-3.13.0-85-generic root=/dev/mapper/os-root ro console=tty0
net.iframes-0 biosdevname-0 rootdelay-00 nomodeset
root=UUID=645be0e7-9113-48d7-aae0-6b2369442 nomdmonddf nomdmonisw intel_iommu=on
```

4.4.4 Server BIOS—Enable SR-IOV

To confirm that SR-IOV is enabled on the server hardware BIOS configuration (Advance option), open the system BIOS and inspect the Processor Settings and the Integrated Devices. Under Processor Settings, ensure that **Virtualization Technology** is enabled, as shown:



System BIOS

System BIOS Settings • Processor Settings

Logical Processor ○ Enabled ● Disabled

Alternate RTID (Requestor Transaction ID) Setting ○ Enabled ● Disabled

Virtualization Technology ● Enabled ○ Disabled

Adjacent Cache Line Prefetch ● Enabled ○ Disabled

Hardware Prefetcher ● Enabled ○ Disabled

DCU Streamer Prefetcher ● Enabled ○ Disabled

DCU IP Prefetcher ● Enabled ○ Disabled

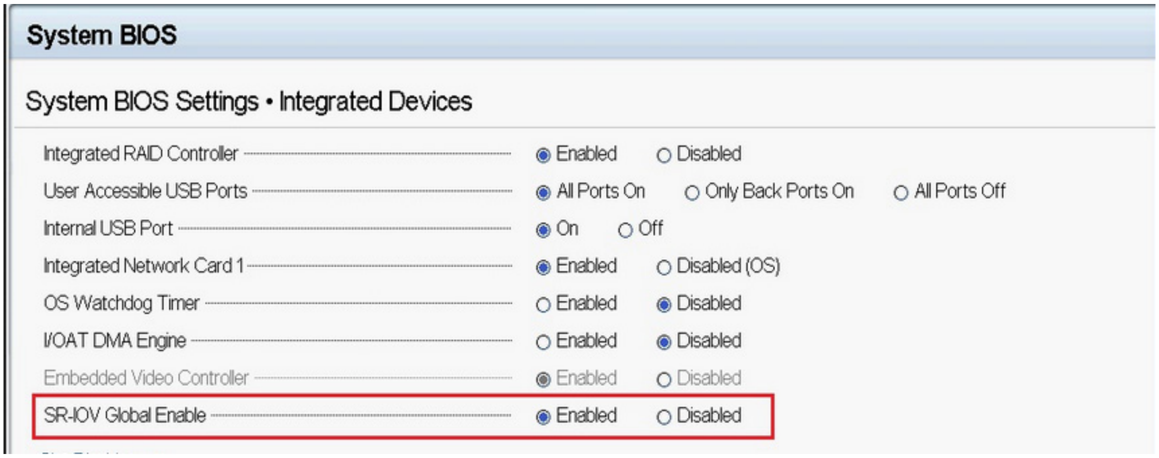
Execute Disable ● Enabled ○ Disabled

Number of Cores per Processor All

Processor 64-bit Support Yes

Processor Core Speed 2.40 GHz

Under Integrated Devices, ensure that **SR-IOV Global Enable** is enabled, as shown:



System BIOS

System BIOS Settings • Integrated Devices

Integrated RAID Controller ● Enabled ○ Disabled
User Accessible USB Ports ● All Ports On ○ Only Back Ports On ○ All Ports Off
Internal USB Port ● On ○ Off
Integrated Network Card 1 ● Enabled ○ Disabled (OS)
OS Watchdog Timer ○ Enabled ● Disabled
I/OAT DMA Engine ○ Enabled ● Disabled
Embedded Video Controller ● Enabled ○ Disabled
SR-IOV Global Enable ● Enabled ○ Disabled

4.4.5 Controller Node—Neutron Server

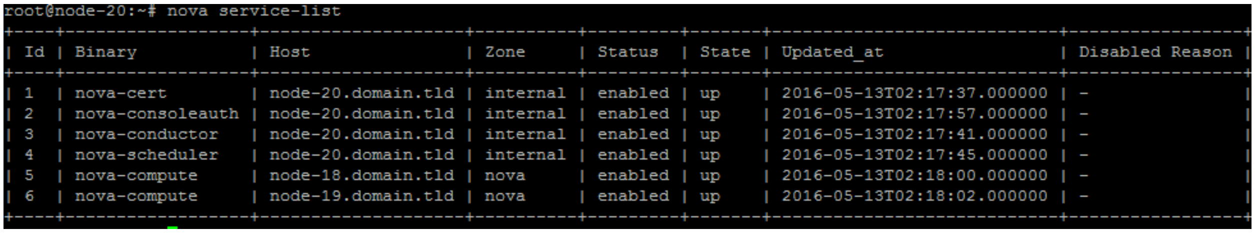
To confirm that the Neutron server is running properly on the Controller node, use the `service neutron-server status` command, as shown:

```
# service neutron-server status

neutron-server start/running, process 7681
```

4.4.6 Controller Node—Nova Services

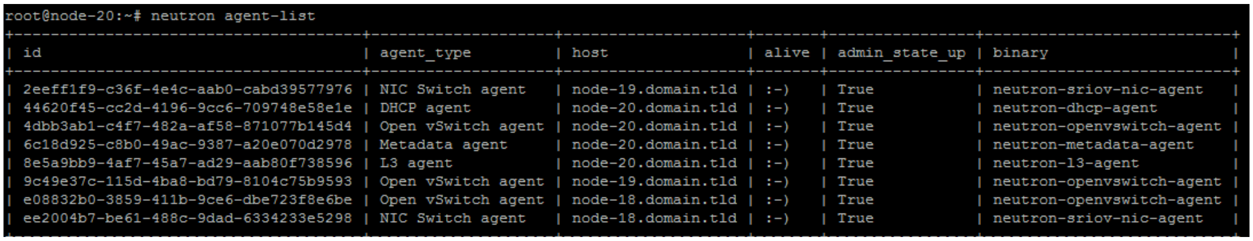
To confirm that all Nova services (consoleauth/conductor/scheduler/compute/cert) are running properly, use the `nova service-list` command, as shown:



```
root@node-20:~# nova service-list
+----+------------------+--------------------+----------+---------+-------+----------------------------+-----------------+
| Id | Binary           | Host               | Zone     | Status  | State | Updated_at                 | Disabled Reason |
+----+------------------+--------------------+----------+---------+-------+----------------------------+-----------------+
| 1  | nova-cert        | node-20.domain.tld | internal | enabled | up    | 2016-05-13T02:17:37.000000 | -               |
| 2  | nova-consoleauth | node-20.domain.tld | internal | enabled | up    | 2016-05-13T02:17:57.000000 | -               |
| 3  | nova-conductor   | node-20.domain.tld | internal | enabled | up    | 2016-05-13T02:17:41.000000 | -               |
| 4  | nova-scheduler   | node-20.domain.tld | internal | enabled | up    | 2016-05-13T02:17:45.000000 | -               |
| 5  | nova-compute     | node-18.domain.tld | nova     | enabled | up    | 2016-05-13T02:18:00.000000 | -               |
| 6  | nova-compute     | node-19.domain.tld | nova     | enabled | up    | 2016-05-13T02:18:02.000000 | -               |
+----+------------------+--------------------+----------+---------+-------+----------------------------+-----------------+
```

4.4.7 Controller Node—Neutron Agents

To confirm that the Neutron agents for MOS 8.0 (DHCP/metadata/openvswitch/sriov) are running properly on the Controller node, use the `neutron agent-list` command, as shown:



```
root@node-20:~# neutron agent-list
+--------------------------------------+--------------------+--------------------+-------+----------------+---------------------------+
| id                                   | agent_type         | host               | alive | admin_state_up | binary                    |
+--------------------------------------+--------------------+--------------------+-------+----------------+---------------------------+
| 2eeff1f9-c36f-4e4c-aab0-cabd39577976 | NIC Switch agent   | node-19.domain.tld | :-)   | True           | neutron-sriov-nic-agent   |
| 44620f45-cc2d-4196-9cc6-709748e58e1e | DHCP agent         | node-20.domain.tld | :-)   | True           | neutron-dhcp-agent        |
| 4dbb3ab1-c4f7-482a-af58-871077b145d4 | Open vSwitch agent | node-20.domain.tld | :-)   | True           | neutron-openvswitch-agent |
| 6c18d925-c8b0-49ac-9387-a20e070d2978 | Metadata agent     | node-20.domain.tld | :-)   | True           | neutron-metadata-agent    |
| 8e5a9bb9-4af7-45a7-ad29-aab80f738596 | L3 agent           | node-20.domain.tld | :-)   | True           | neutron-l3-agent          |
| 9c49e37c-115d-4ba8-bd79-8104c75b9593 | Open vSwitch agent | node-19.domain.tld | :-)   | True           | neutron-openvswitch-agent |
| e08832b0-3859-411b-9ce6-dbe723f8e6be | Open vSwitch agent | node-18.domain.tld | :-)   | True           | neutron-openvswitch-agent |
| ee2004b7-be61-488c-9dad-6334233e5298 | NIC Switch agent   | node-18.domain.tld | :-)   | True           | neutron-sriov-nic-agent   |
+--------------------------------------+--------------------+--------------------+-------+----------------+---------------------------+
```

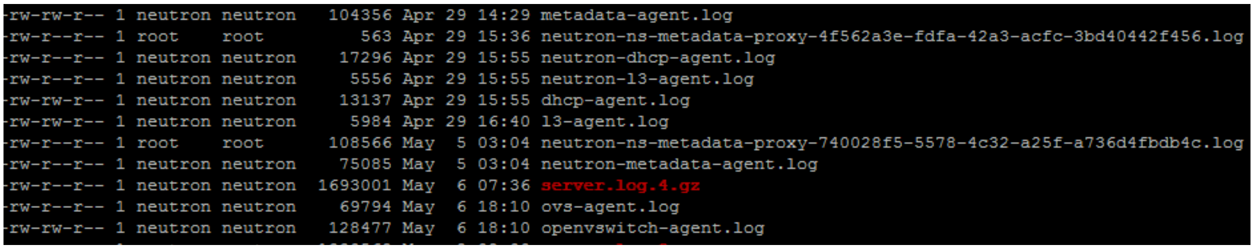
4.4.8 SR-IOV VF Connectivity

After creating a VM, if you cannot ping between the SR-IOV VFs, examine the related Neutron and Nova logs. On the Controller node, examine the Neutron and Nova server logs on the Compute node and the Controller node for diagnostic information.

4.4.8.1 Server Logs on the Compute Node

For a list of the Neutron server logs on the Compute node, type the following command:

```
# cd /var/log/neutron/
```



```
-rw-rw-r-- 1 neutron neutron   104356 Apr 29 14:29 metadata-agent.log
-rw-r--r-- 1 root    root         563 Apr 29 15:36 neutron-ns-metadata-proxy-4f562a3e-fdfa-42a3-acfc-3bd40442f456.log
-rw-r--r-- 1 neutron neutron    17296 Apr 29 15:55 neutron-dhcp-agent.log
-rw-r--r-- 1 neutron neutron     5556 Apr 29 15:55 neutron-l3-agent.log
-rw-rw-r-- 1 neutron neutron    13137 Apr 29 15:55 dhcp-agent.log
-rw-rw-r-- 1 neutron neutron     5984 Apr 29 16:40 l3-agent.log
-rw-r--r-- 1 root    root      108566 May  5 03:04 neutron-ns-metadata-proxy-740028f5-5578-4c32-a25f-a736d4fbdb4c.log
-rw-r--r-- 1 neutron neutron    75085 May  5 03:04 neutron-metadata-agent.log
-rw-r--r-- 1 neutron neutron  1693001 May  6 07:36 server.log.4.gz
-rw-r--r-- 1 neutron neutron    69794 May  6 18:10 ovs-agent.log
-rw-rw-r-- 1 neutron neutron   128477 May  6 18:10 openvswitch-agent.log
```

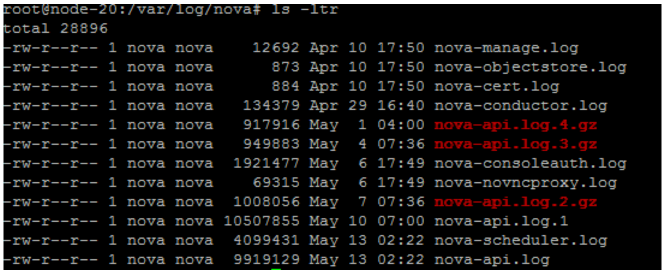
From this list, examine: `neutron-dhcp-agent.log`, `neutron-13-agent.log`, `openswitch-agent.log`, `metadata-agent.log`, and `ovs-agent.log`.

For a list of Nova server logs on the Compute node, type the following command:

```
# cd /var/log/nova
```

```
root@node-20:/var/log/nova# ls -ltr
```

From the following list, examine `nova-manage.log`, `nova-api.log`, `nova-consoleauth.log`, and `nova-novncproxy.log`.



```
root@node-20:/var/log/nova# ls -ltr
total 28896
-rw-r--r-- 1 nova nova    12692 Apr 10 17:50 nova-manage.log
-rw-r--r-- 1 nova nova      873 Apr 10 17:50 nova-objectstore.log
-rw-r--r-- 1 nova nova      884 Apr 10 17:50 nova-cert.log
-rw-r--r-- 1 nova nova   134379 Apr 29 16:40 nova-conductor.log
-rw-r--r-- 1 nova nova   917916 May  1 04:00 nova-api.log.4.gz
-rw-r--r-- 1 nova nova   949883 May  4 07:36 nova-api.log.3.gz
-rw-r--r-- 1 nova nova  1921477 May  6 17:49 nova-consoleauth.log
-rw-r--r-- 1 nova nova    69315 May  6 17:49 nova-novncproxy.log
-rw-r--r-- 1 nova nova  1008056 May  7 07:36 nova-api.log.2.gz
-rw-r--r-- 1 nova nova 10507855 May 10 07:00 nova-api.log.1
-rw-r--r-- 1 nova nova  4099431 May 13 02:22 nova-scheduler.log
-rw-r--r-- 1 nova nova  9919129 May 13 02:22 nova-api.log
```

4.4.8.2 Server Logs on the Controller Node

For a list of the Nova server logs on the Controller node, type the following command:

```
# cd /var/log/neutron/
```

```
root@node-18:/var/log/neutron# ls
```

```
neutron-plugin-sriov-agent.log  ovs-agent.log
```

From this list, examine: `neutron-plugin-sriov-agent.log` and `ovs-agent.log`.

For a list of Nova server logs on the Controller node, type the following command:

```
# cd /var/log/nova
root@node-18:/var/log/nova# ls
nova-compute.log  nova-compute.log.2.gz  nova-compute.log.3.gz  nova-manage.log
```

From the following list, examine `nova-compute.log` and `nova-manage.log`.

5 Limitations

When planning the OpenStack environment, consider the following limitations

- SR-IOV environments were tested with KVM only.
- For 45000 Series Adapters, the adapter drivers are not Inbox. Therefore, you must needs to install the qed/qede driver on the Compute node and VM manually.
- When NPAR is enabled on the adapter, do not configure the PF with SR-IOV and VLAN.
- 3400/8400 Series Adapters are not able to set the quality of service (QoS) bandwidth limit on SR-IOV VFs.
- The SR-IOV IP Network interface does not receive the DHCP IP address while creating the VM. Therefore, you must assign the IP address for the SR-IOV interface manually.

6 Testing

This chapter provides a sample testing procedure. To verify the SR-IOV configuration with OpenStack:

1. Create two VMs with different ports using the procedures in Creating the SR-IOV Port and Creating a VM with the SR-IOV Network.

   This example creates VMs named VNM3 and VNM4.

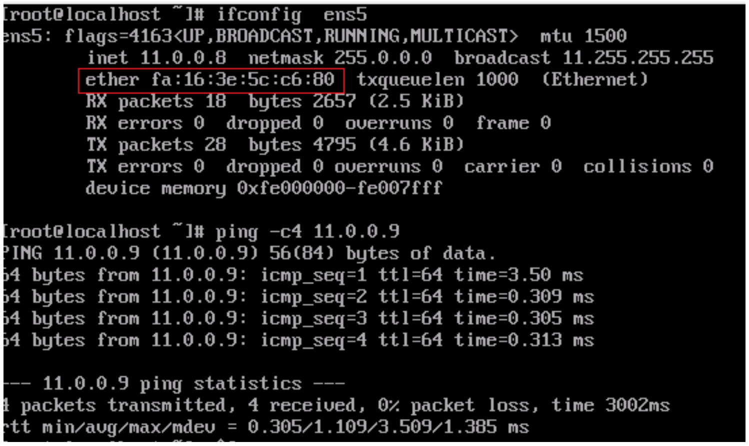
2. Log in to the VMs using the floating IP address, which has been assigned to the private network net1 port.

3. Ideally, the SR-IOV port should get DHCP IP address 11.0.0.*x* in the VM, but in this case, it is not getting the IP address automatically.

   Assign an IP address to the SR-IOV port in the VMs using the `ifconfig` command, as follows:

   - VNM3—SR-IOV port IP address 11.0.0.8
   - VNM4—SR-IOV port IP address 11.0.0.9

   Following is a sample output for VNM3.

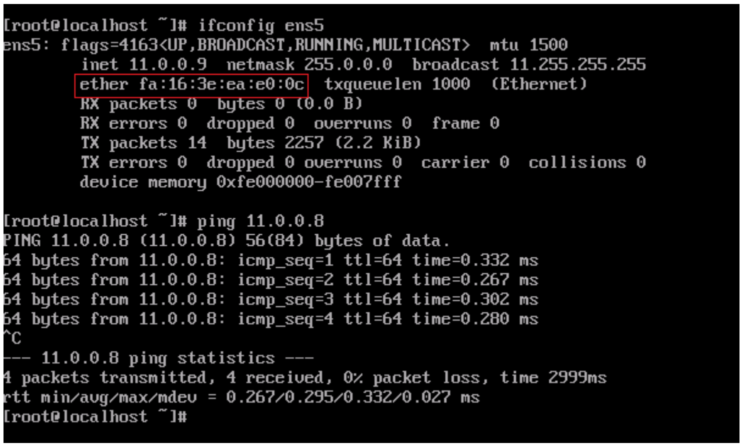


```
[root@localhost ~]# ifconfig  ens5
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 11.0.0.8  netmask 255.0.0.0  broadcast 11.255.255.255
        ether fa:16:3e:5c:c6:80  txqueuelen 1000  (Ethernet)
        RX packets 18  bytes 2657 (2.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28  bytes 4795 (4.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xfe000000-fe007fff

[root@localhost ~]# ping -c4 11.0.0.9
PING 11.0.0.9 (11.0.0.9) 56(84) bytes of data.
64 bytes from 11.0.0.9: icmp_seq=1 ttl=64 time=3.50 ms
64 bytes from 11.0.0.9: icmp_seq=2 ttl=64 time=0.309 ms
64 bytes from 11.0.0.9: icmp_seq=3 ttl=64 time=0.305 ms
64 bytes from 11.0.0.9: icmp_seq=4 ttl=64 time=0.313 ms

--- 11.0.0.9 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.305/1.109/3.509/1.385 ms
```

   Following is a sample output for VNM4.



```
[root@localhost ~]# ifconfig ens5
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 11.0.0.9  netmask 255.0.0.0  broadcast 11.255.255.255
        ether fa:16:3e:ea:e0:0c  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 2257 (2.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xfe000000-fe007fff

[root@localhost ~]# ping 11.0.0.8
PING 11.0.0.8 (11.0.0.8) 56(84) bytes of data.
64 bytes from 11.0.0.8: icmp_seq=1 ttl=64 time=0.332 ms
64 bytes from 11.0.0.8: icmp_seq=2 ttl=64 time=0.267 ms
64 bytes from 11.0.0.8: icmp_seq=3 ttl=64 time=0.302 ms
64 bytes from 11.0.0.8: icmp_seq=4 ttl=64 time=0.280 ms
^C
--- 11.0.0.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.267/0.295/0.332/0.027 ms
[root@localhost ~]#
```