# OpenStack Configuration for SR-IOV Support

You need to follow RedHat OpenStack documentation for detailed instructions for SR-IOV configuration. The details given below may be referred to in addition to this.

**On Controller Node**

1. In /etc/neutron/plugins/ml2/ml2_conf.ini file:

```
[ml2]

tenant_network_types = vxlan, vlan

# Add support for vlan, vxlan and flat type drivers

type_drivers = vlan, vxlan, flat

# Add support for sriov and vts mechanism drivers, in the following sequence

mechanism_drivers = sriovnicswitch, cisco_vts
```

2. In /etc/neutron/plugins/ml2/ml2_conf.ini file:

```
 [ml2_type_vlan]

# Add all the physnet names that will be used on your compute hosts along with the VLAN
 ranges

# reserved for the provider networks for those physnets

network_vlan_ranges = physnet1:2000:2100, physnet2:2500:2600

# Note: network vlan range should be in between the range specified in VTS.


ml2_type_flat]

# List of physical_network names with which flat networks can be created.

# Use default '*' to allow flat networks with arbitrary physical_network names.

flat_networks = *
```

3. If your deployment has a /etc/neutron/plugins/ml2/ml2_conf_sriov.ini file, include the following section in your ml2_conf_sriov.ini file.

```
[ml2_sriov]

# The default supported_pci_vendor_devs value for the installation may have the value
```

OpenStack Configuration for SR-IOV Support

```
only for the PFs

# in your compute. Use lspci —nn | grep Ethernet to find the ids for the Virtual functions
 and add that

# as well in this.

supported_pci_vendor_devs = 8086:10fb,8086:10ed

agent_required = True
```

Update ExecStart section in the file /usr/lib/systemd/system/neutron-server.service to include ml2_conf_sriov.ini config file:

```
ExecStart=/usr/bin/neutron-server --config-file /usr/share/neutron/neutron-dist.conf
--config-dir /usr/share/neutron/server --config-file /etc/neutron/neutron.conf
--config-file /etc/neutron/plugin.ini —config-file
/etc/neutron/plugins/ml2/ml2_conf_sriov.ini --config-dir /etc/neutron/conf.d/common
--config-dir /etc/neutron/conf.d/neutron-server --log-file /var/log/neutron/server.lo
```

4. If the ml2_conf_sriov.inif file is not present, then add the ml2_sriov section to /etc/neutron/plugins/ml2/ml2_conf.ini file.
5. Restart neutron service using **systemctl restart neutron-server.service**.

On each compute hosts' /etc/nova/nova.conf file, define pci_passthroughs_whitelist.

```
pci_passthrough_whitelist = [ {"devname": "eth4", "physical_network":"physnet1"}, {"devname":
 "eth5", "physical_network":"physnet2"}]
```

For multiple SR-IOV nics , there should be a mapping entry per physnet/NIC card

### Enable the OpenStack Networking SR-IOV agent

If not present already as part of the OSPD/Packstack installation (check the neutron agent-list on your director/controller node), then you will need to install "openstack-neutron-sriov-nic-agent" on your compute hosts and start that agent/service. After you have installed, follow RedHat documentation. It is important to have the "physical_device_mappings section" in the /etc/neutron/plugins/ml2/sriov_agent.ini file. You can leave the exclude_devices section blank.

Then proceed to create the SR-IOV port instances.

**Note**

- SR-IOV support is present only for network types VLAN and Flat.
- While creating a provider network, when choosing VLAN network type, choose a segmentation ID (VLAN) from within the physnet range.
- While creating a provider network of type Flat, segmentation id can be any value or null. VTS will provide native VLAN for Flat networks.
- While creating a port, as described in RedHat OpenStack documentation, choosing binding:vnic_type = direct is a must for SR-IOV.
- Cirros image is not supported for creating instances using SR-IOV ports.


- Sample for SR-IOV Trunk (No-Bonding), on page 3
- Sample for SR-IOV Trunk (Bonding), on page 3

# Sample for SR-IOV Trunk (No-Bonding)

This section provides an example for enabling SR-IOV trunk (no-bonding).

**Step 1** Create a Flat network.

```
openstack network create --provider-network-type flat --provider-physical-network SRIOV-B vma-flat-net
flatneta=$(openstack network list -f value -c ID -c Name | grep vma-flat-net | awk '{print $1}')
```

**Step 2** Create a VLAN network.

```
openstack network create --provider-network-type vlan --provider-physical-network SRIOV-A
--provider-segment xxx vma-vlanxxx-net
vlanneta=$(openstack network list -f value -c ID -c Name | grep vma-vlanxxx-net | awk '{print $1}')
```

**Step 3** Create subnet on the Flat and VLAN network.

```
openstack subnet create --network $flatneta --subnet-range 1.1.1.0/24 --gateway 1.1.1.1 --no-dhcp
vma-flat-subnet
openstack subnet create --network $vlanneta --subnet-range 2.1.1.0/24 --gateway 2.1.1.1 --no-dhcp
vma-vlanxxx-subnet
```

**Step 4** Create port on Flat and VLAN network.

```
openstack port create --vnic-type direct --network $ flatneta flatneta-port
openstack port create --vnic-type direct --network $vlanneta vma-child-port-vlanxxx-a
```

**Step 5** Create trunk with parent (Flat) and sub-port (VLAN).

```
openstack network trunk create --parent-port flatneta-port \
 --subport port=vma-child-port-vlanxxx-a, segmentation-type=vlan, segmentation-id=xxx vma-trunk-a
```

**Step 6** Attach the parent port to Nova instance.

```
parentaid=$( openstack port list -f value -c ID -c Name | grep flatneta-port | awk '{print $1}’)
nova boot --flavor m1.medium --image cents7-1800-custom --nic port-id=$parentaid  vma
```

# Sample for SR-IOV Trunk (Bonding)

This section provides an example for enabling SR-IOV trunk (bonding).

**Step 1** Create Flat networks for each SR-IOV physnet, but use the same network name for both the networks. Keep the subnets identical.

**Note** It is important to keep the network and subnet names exactly the same.

**a.** Create a Flat network on the first physnet.

```
openstack network create --provider-network-type flat --provider-physical-network SRIOV-A
vma-flat-net
```

**b.** Find and save the ID of the network created.

```
flatneta=$(openstack network list -f value -c ID -c Name | grep vma-flat-net | awk '{print $1}')
```

**c.** Create a Flat network on the second physnet.

```
openstack network create --provider-network-type flat --provider-physical-network SRIOV-B
vma-flat-net
```

**d.** Find and save the ID of the network created, there should be two matching, so exclude the $flatneta network.

```
flatnetb=$(openstack network list -f value -c ID -c Name | grep vma-flat-net | grep -v $flatneta
 | awk '{print $1}')
```

**e.** Create subnet on the Flat networks.

```
openstack subnet create --network $flatneta --subnet-range 1.1.1.0/24 --gateway 1.1.1.1 --no-dhcp
 vma-flat-subnet
openstack subnet create --network $flatnetb --subnet-range 1.1.1.0/24 --gateway 1.1.1.1 --no-dhcp
 vma-flat-subnet
```

**Step 2** Create VLAN Networks for each VLAN tag that should be allowed in this trunk.

Create a pair of networks, one on each physnet for each VLAN. Replace *xxx* with the VLAN ID to be used in the following examples.

**Note** It is important to keep the network name and subnet name exactly the same for the pair created for a specific VLAN.

**a.** Create a VLAN network on the first physnet.

```
openstack network create --provider-network-type vlan --provider-physical-network SRIOV-A
--provider-segment xxx vma-vlanxxx-net
vlanneta=$(openstack network list -f value -c ID -c Name | grep vma-vlanxxx-net | awk '{print
$1}')
```

**b.** Create a VLAN network on the second physnet.

```
openstack network create --provider-network-type vlan --provider-physical-network SRIOV-B
--provider-segment xxx vma-vlanxxx-net
vlannetb=$(openstack network list -f value -c ID -c Name | grep vma-vlanxxx-net | grep -v $vlanneta
 | awk '{print $1}')
```

**c.** Create subnet on VLAN networks.

```
openstack subnet create --network $vlanneta --subnet-range 2.1.1.0/24 --gateway 2.1.1.1 --no-dhcp
 vma-vlanxxx-subnet
openstack subnet create --network $vlannetb --subnet-range 2.1.1.0/24 --gateway 2.1.1.1 --no-dhcp
 vma-vlanxxx-subnet
```

**Step 3** Create ports on the Flat as well as VLAN networks created above.

```
openstack port create --vnic-type direct --network $flatneta vma-parent-port-a
openstack port create --vnic-type direct --network $flatnetb vma-parent-port-b
```

**a.** Create as many child ports as VLANs needed.

```
openstack port create --vnic-type direct --network $vlanneta vma-child-port-vlanxxx-a
openstack port create --vnic-type direct --network $vlannetb vma-child-port-vlanxxx-b
```

**Step 4** Create two trunks, one per physnet. The parent port is the Flat network port, and all the VLAN network ports are the child ports.

```
openstack network trunk create --parent-port vma-parent-port-a \
 --subport port=vma-child-port-vlanxxx-a, segmentation-type=vlan, segmentation-id=xxx vma-trunk-a
openstack network trunk create --parent-port vma-parent-port-b \
 --subport port=vma-child-port-vlanxxx-b, segmentation-type=vlan, segmentation-id=xxx vma-trunk-b
```

**Step 5** Create an instance with NICs mapping to the trunk parent ports.

```
parentaid=$( openstack port list -f value -c ID -c Name | grep vma-parent-port-a | awk '{print $1}')
parentbid=$( openstack port list -f value -c ID -c Name | grep vma-parent-port-b | awk '{print $1}')


 nova boot --flavor m1.medium --image cents7-1801-custom --nic port-id=$parentaid  --nic
port-id=$parentbid  vma
```

**Step 6** Attach networks/subnets to routers.

Create a router and attach the interfaces of the VLAN networks—Only one of the VLAN networks in the pair for each VLAN should be attached. The Flat network subnet (one in the pair) may need to be attached if there are untagged packets from the VM/instance to be routed as well.

**Step 7** Configure instance interfaces with bonded NICs in active/standby.

OpenStack Configuration for SR-IOV Support