EPA Hands-On Lab #2

Version 1.0

Date 30th Nov 2018

1 Objective

Using an OpenStack Queens based Multi-Node cluster, help get an understanding of Single root I/O virtualization (SR-IOV) as a simple option for network performance enhancement, though hard to automate for non-static setups. Plan is to use netperf as a tool of choice running on an CentoS Instance using SR-IOV based network ports and validate both East-West and North-South communication.

2 Prerequisites

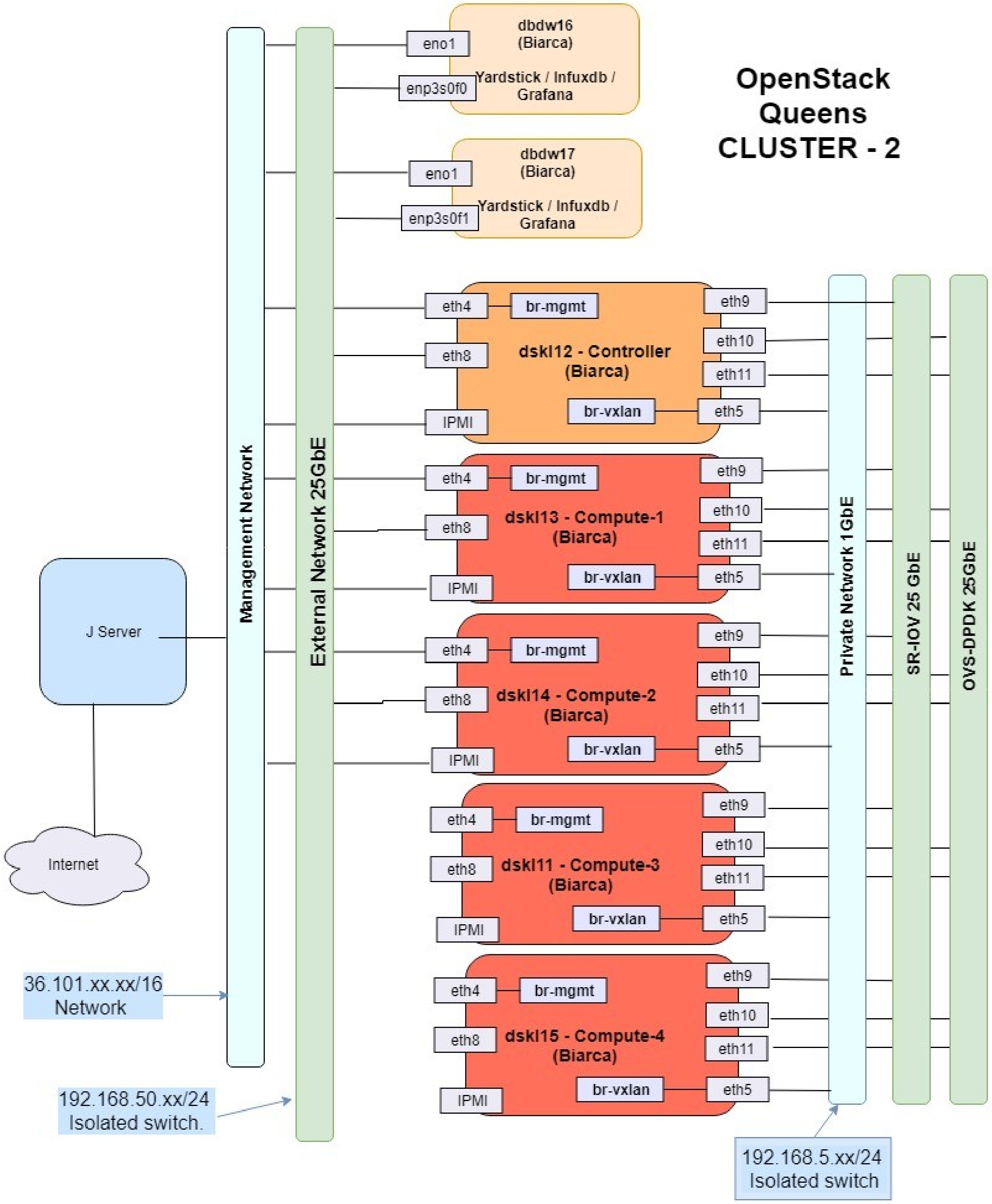
- Networking fundamentals
- Basic knowledge of SR-IOV and familiarity with OpenStack environment
- Hands-On EPA Lab – Introduction
- Overview of current Infrastructure
- Host to be used for connecting to SR-IOV capable OpenStack Instances
- Infra-access details available (**Did you receive a note/slip? - If NO! - please STOP here and ASK!!!**)
    - Credentials to login into jumpserer
    - Host Access details
    - OpenStack Dashboard Credentials
    - SR-IOV Capable Instances details for East-West and North-South
        - External/Floating IP
        - Guest Net IP
        - SR-IOV Network IP - East - West & North-South
        - SSH Key pair Host Info - same as your access host
        - User Name: ‘centos’
    - SR-IOV Capable Compute Nodes Availability Zone
- Commands in BLUE
- Dark Orange 2 text needs to be **updated before you attempt the** command
- Dark Magenta - “Cambria” font - **sample/console output - DO NOT USE for COPY/PASTE**
- Highlighted information has yellow background

3 SR-IOV Instance(s) Basic Details

- Per Instance info
    - 2 vCPUs & 2 GB RAM
    - 10 GB Disk
    - Centos 7.0
    - Availability zones used “compute-3” and “compute-4” - Note Availability zones may be specific to user
    - Key pair as needed for SSH connection from host
- 3 Instances per user/group (sr-gr1-i1 and sr-gr1-i2 for East-West & sr-gr1-i1 and sr-gr1-i3 for North-South)

- Additional details on SR-IOV are captured SR-IOV Information

4 Network Diagram

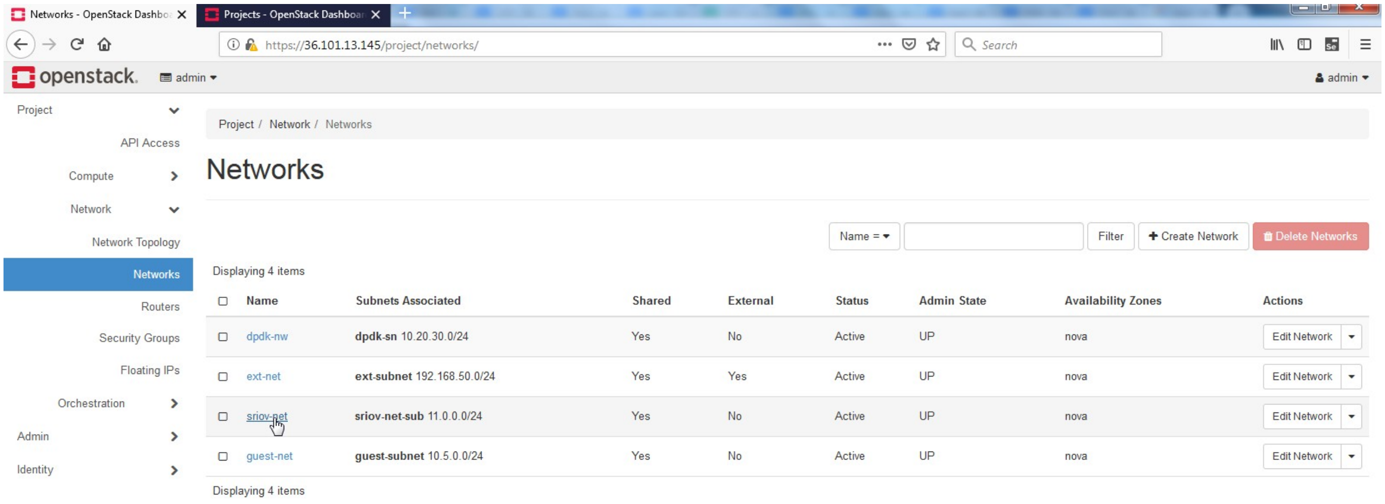

OpenStack Queens CLUSTER - 2
dbdw16 (Biarca)
Yardstick / Infuxdb / Grafana
eno1
enp3s0f0
dbdw17 (Biarca)
Yardstick / Infuxdb / Grafana
eno1
enp3s0f1
dskl12 - Controller (Biarca)
dskl13 - Compute-1 (Biarca)
dskl14 - Compute-2 (Biarca)
dskl11 - Compute-3 (Biarca)
dskl15 - Compute-4 (Biarca)
eth4
eth8
IPMI
br-mgmt
br-vxlan
eth9
eth10
eth11
eth5
Management Network
External Network 25GbE
Private Network 1GbE
SR-IOV 25 GbE
OVS-DPDK 25GbE
J Server
Internet
36.101.xx.xx/16 Network
192.168.50.xx/24 Isolated switch.
192.168.5.xx/24 Isolated switch

5 SR-IOV Capable Instance creation from OpenStack Dashboard (Pre-configured to skip)

1. Login to OpenStack Dashboard using the link and access details provided to you

5.1 SR-IOV Port Creation

1. Go to “sriov network” → ports



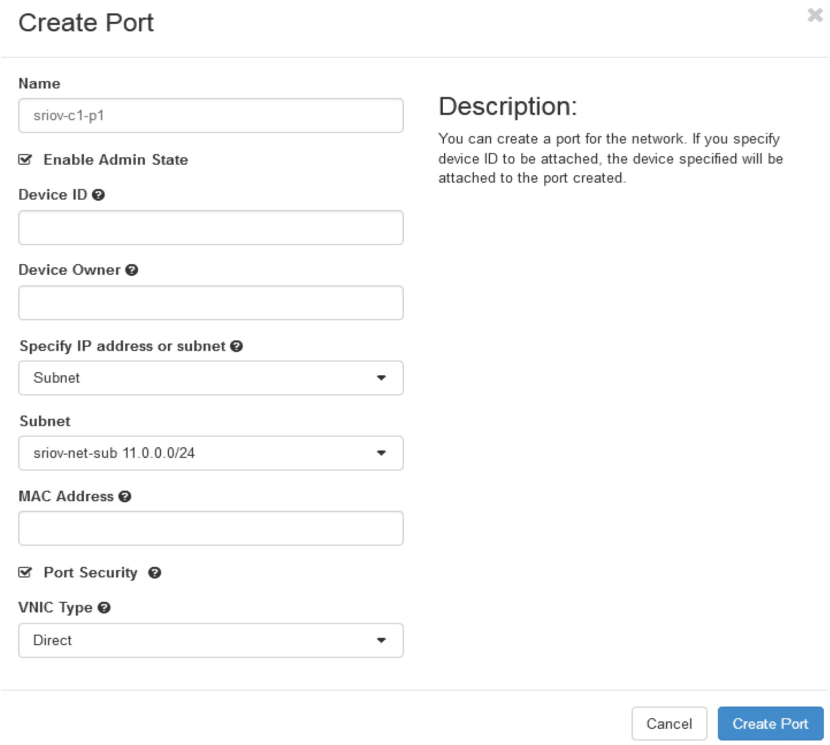
Networks

Displaying 4 items

Name	Subnets Associated	Shared	External	Status	Admin State	Availability Zones	Actions
dpdk-nw	dpdk-sn 10.20.30.0/24	Yes	No	Active	UP	nova	Edit Network
ext-net	ext-subnet 192.168.50.0/24	Yes	Yes	Active	UP	nova	Edit Network
sriov-net	sriov-net-sub 11.0.0.0/24	Yes	No	Active	UP	nova	Edit Network
guest-net	guest-subnet 10.5.0.0/24	Yes	No	Active	UP	nova	Edit Network

Displaying 4 items

2. Select/Click on “Create port”, Enter the name, select the subnet and choose the available subnet. Select vnic type “direct”.



Create Port

Name: sriov-c1-p1

☑ Enable Admin State

Device ID:

Device Owner:

Specify IP address or subnet: Subnet

Subnet: sriov-net-sub 11.0.0.0/24

MAC Address:

☑ Port Security

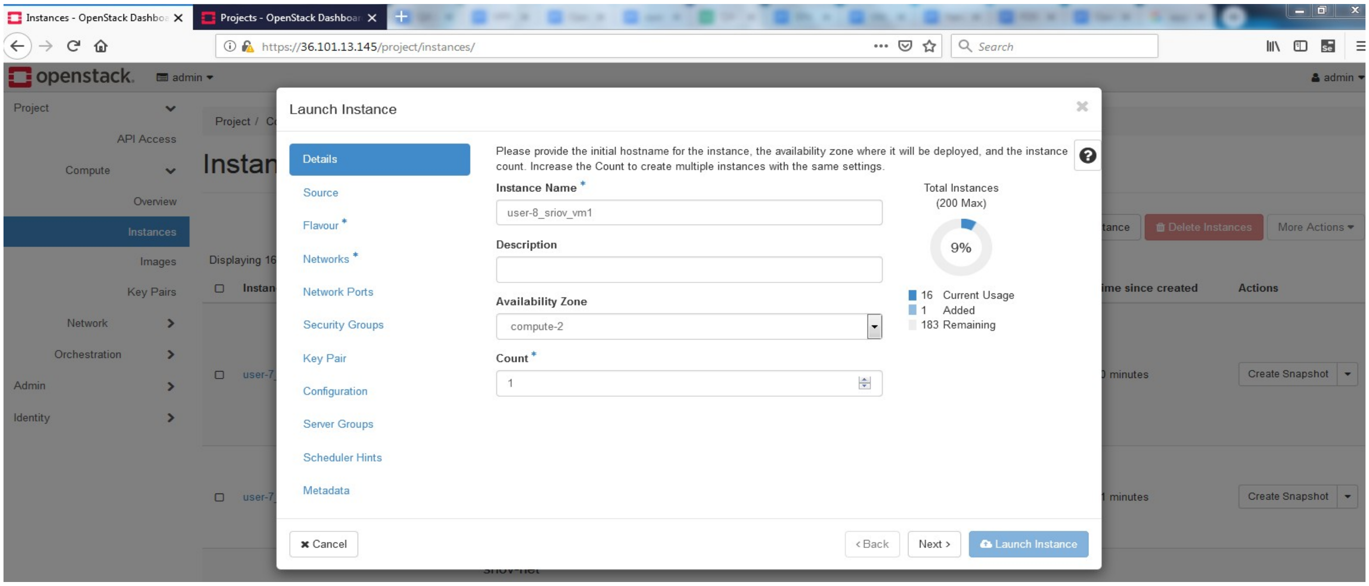
VNIC Type: Direct

Description:

You can create a port for the network. If you specify device ID to be attached, the device specified will be attached to the port created.

5.2 Instance Creation with SR-IOV Interface/Port

1. Go Project → compute → instances. Click on new instance. Enter the name and select availability zone.



Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Instance Name * user-8_sriov_vm1

Description

Availability Zone compute-2

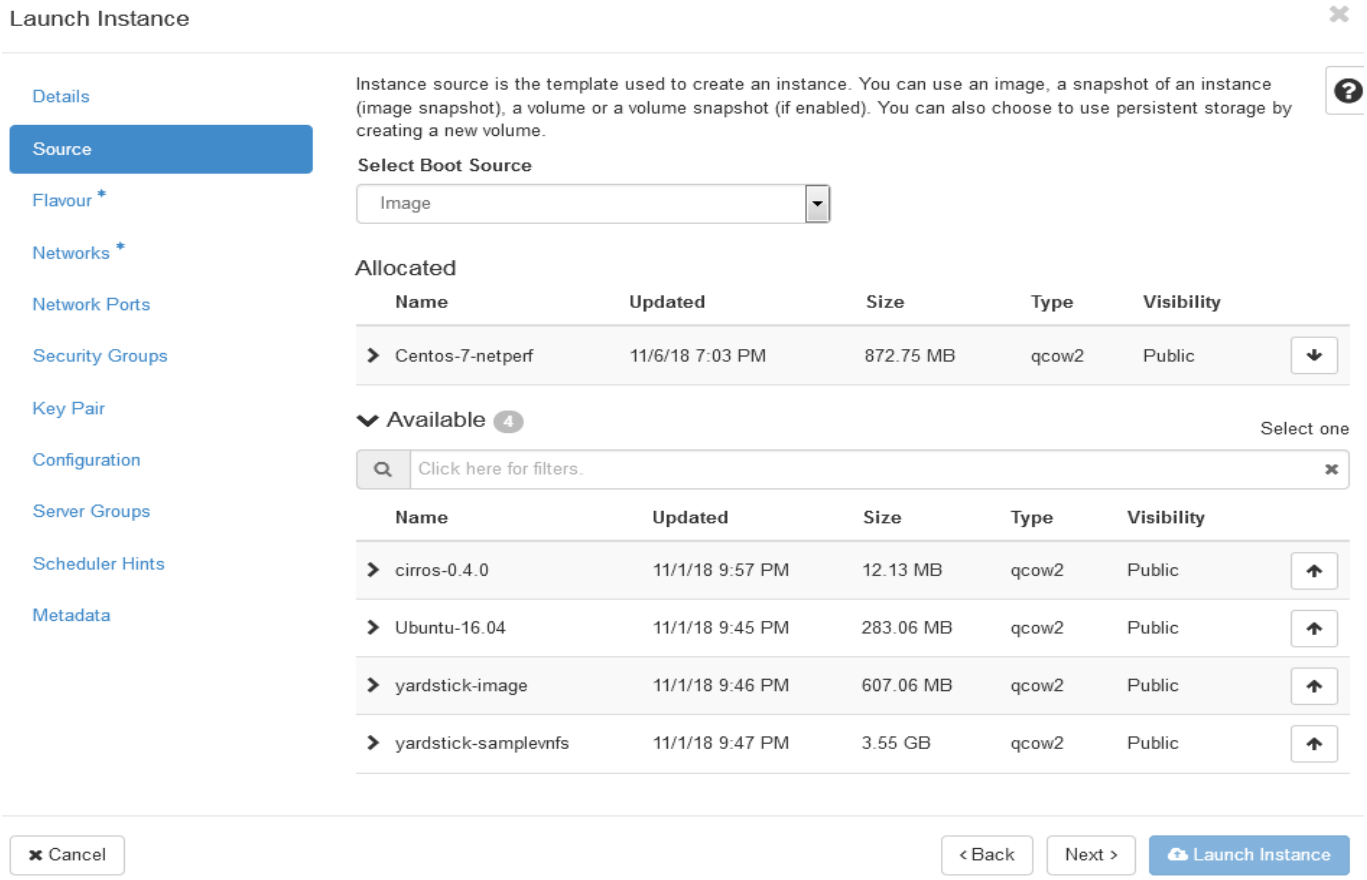
Count * 1

Total Instances (200 Max)

9%

16 Current Usage
1 Added
183 Remaining

2. Select Centos-7-netperf image for your instance.



Launch Instance

Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source: Image

Allocated

Name	Updated	Size	Type	Visibility
Centos-7-netperf	11/6/18 7:03 PM	872.75 MB	qcow2	Public

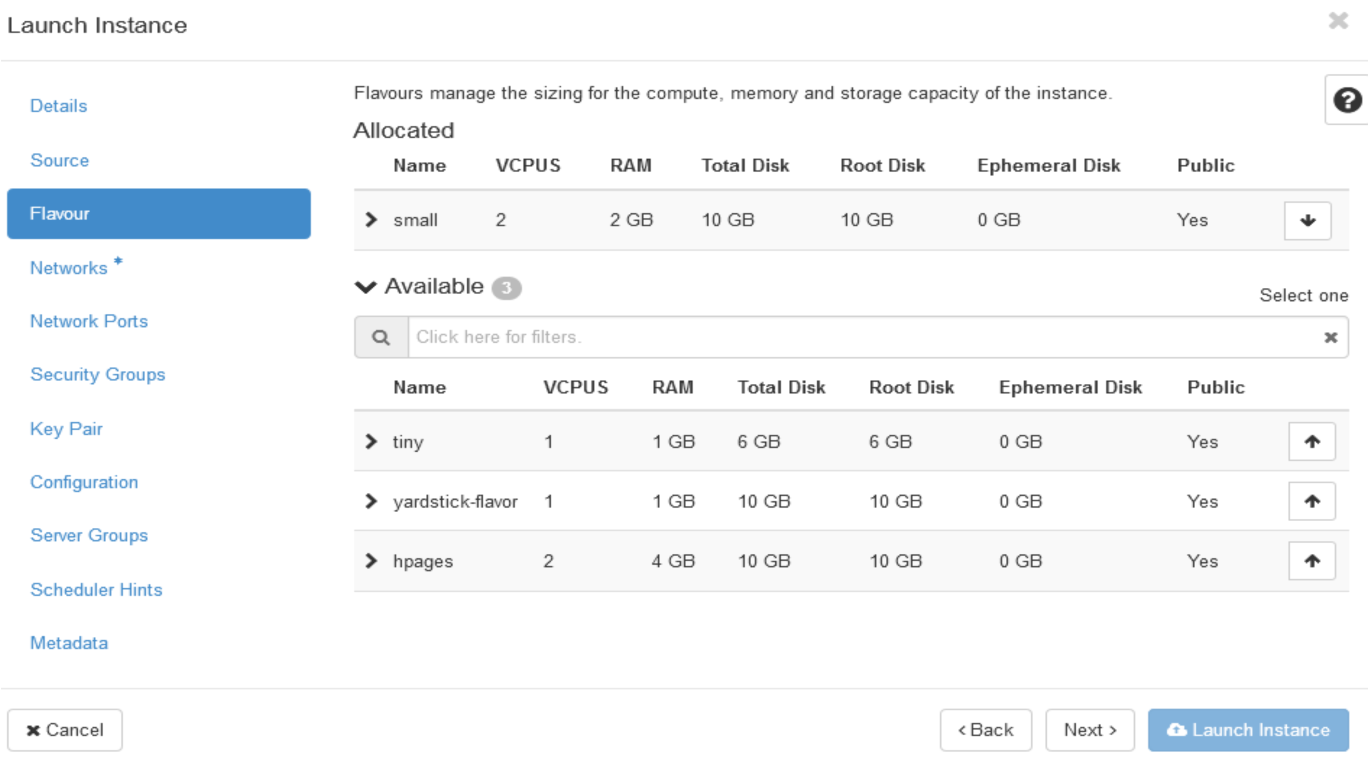
Available 4

Select one

Click here for filters.

Name	Updated	Size	Type	Visibility
cirros-0.4.0	11/1/18 9:57 PM	12.13 MB	qcow2	Public
Ubuntu-16.04	11/1/18 9:45 PM	283.06 MB	qcow2	Public
yardstick-image	11/1/18 9:46 PM	607.06 MB	qcow2	Public
yardstick-samplevnfs	11/1/18 9:47 PM	3.55 GB	qcow2	Public

3. Select the small flavor



Flavours manage the sizing for the compute, memory and storage capacity of the instance.

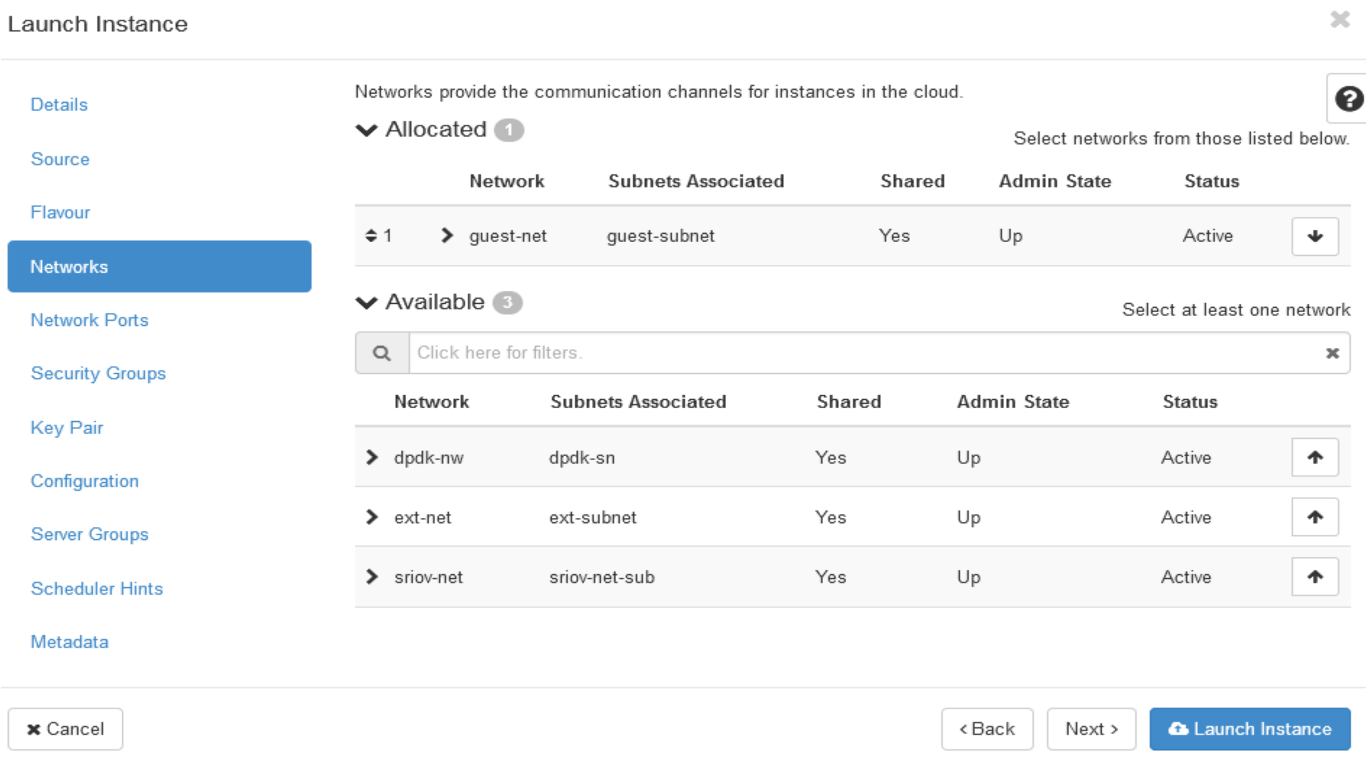
Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
small	2	2 GB	10 GB	10 GB	0 GB	Yes

Available 3

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
tiny	1	1 GB	6 GB	6 GB	0 GB	Yes
yardstick-flavor	1	1 GB	10 GB	10 GB	0 GB	Yes
hpages	2	4 GB	10 GB	10 GB	0 GB	Yes

4. Select the guest-net ONLY, **do not select the sriov-net.**



Networks provide the communication channels for instances in the cloud.

Allocated 1

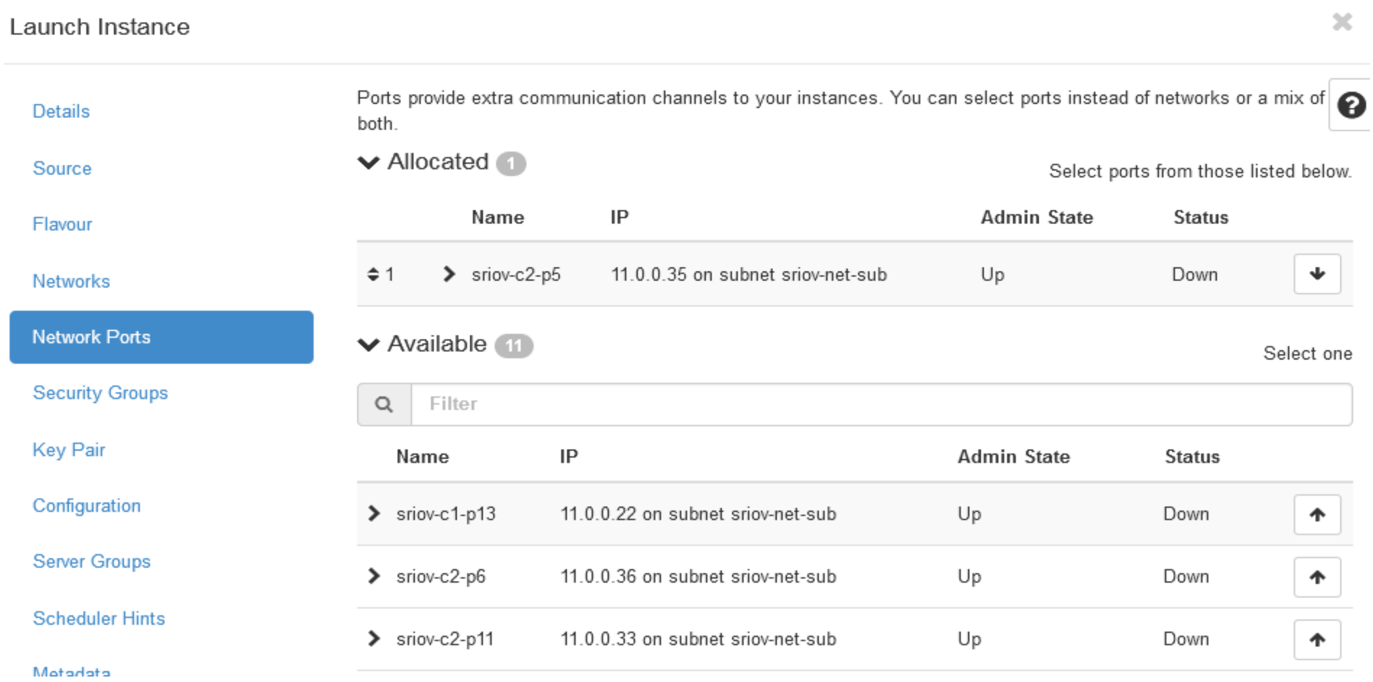
	Network	Subnets Associated	Shared	Admin State	Status
1	guest-net	guest-subnet	Yes	Up	Active

Available 3

Network	Subnets Associated	Shared	Admin State	Status
dpdk-nw	dpdk-sn	Yes	Up	Active
ext-net	ext-subnet	Yes	Up	Active
sriov-net	sriov-net-sub	Yes	Up	Active

5. Select one of the SR-IOV ports available, which matches with the instance name to

identify easily.



Ports provide extra communication channels to your instances. You can select ports instead of networks or a mix of both.

Allocated 1

Select ports from those listed below.

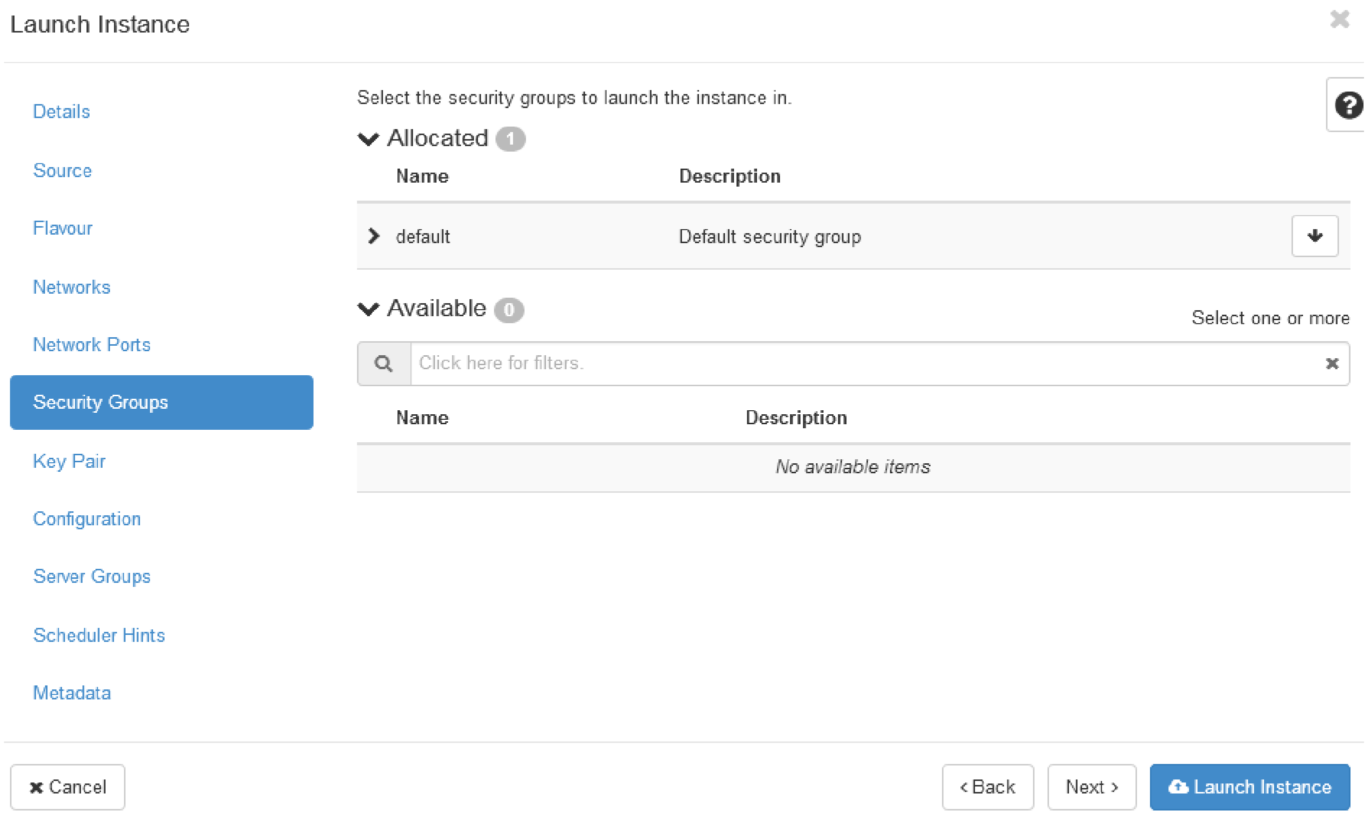
	Name	IP	Admin State	Status
1	sriov-c2-p5	11.0.0.35 on subnet sriov-net-sub	Up	Down

Available 11

Select one

Name	IP	Admin State	Status
sriov-c1-p13	11.0.0.22 on subnet sriov-net-sub	Up	Down
sriov-c2-p6	11.0.0.36 on subnet sriov-net-sub	Up	Down
sriov-c2-p11	11.0.0.33 on subnet sriov-net-sub	Up	Down

6. Select the security group. If there is only one security group available, it will be selected by default.



Select the security groups to launch the instance in.

Allocated 1

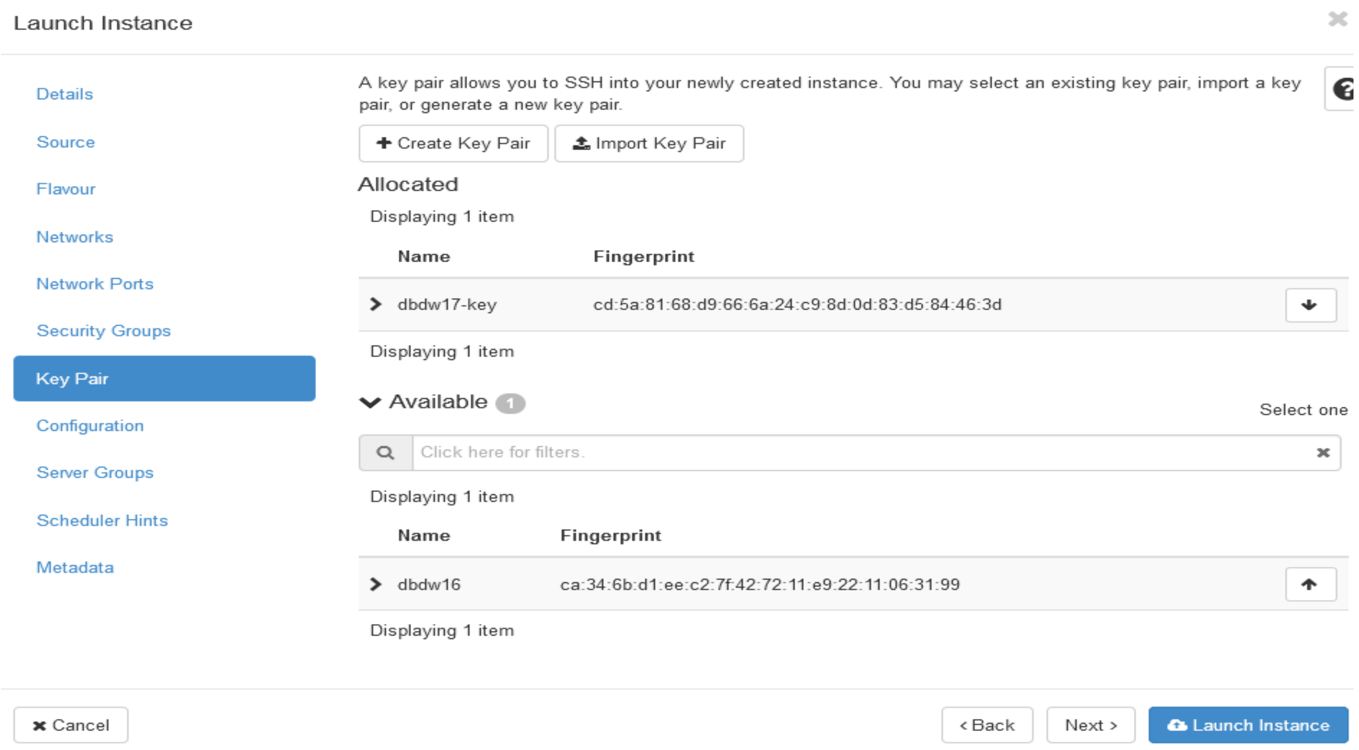
Name	Description
default	Default security group

Available 0

Select one or more

Name	Description
No available items	

7. Select the key pair, this very important to ssh login to the instance using floating ip.



A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair.

Allocated

Displaying 1 item

Name	Fingerprint
dbdw17-key	cd:5a:81:68:d9:66:6a:24:c9:8d:0d:83:d5:84:46:3d

Displaying 1 item

Available 1

Displaying 1 item

Name	Fingerprint
dbdw16	ca:34:6b:d1:ee:c2:7f:42:72:11:e9:22:11:06:31:99

Displaying 1 item

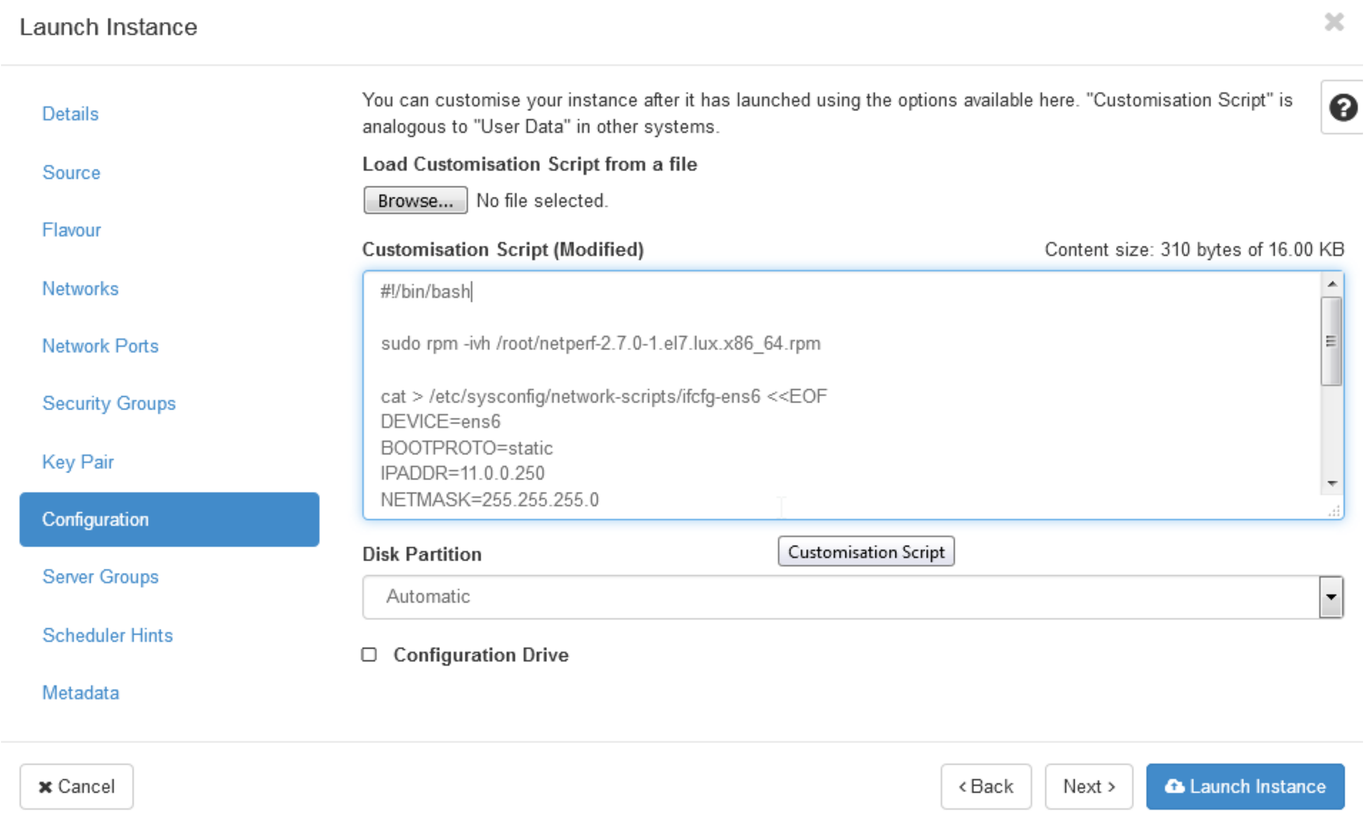
8. Copy and paste or browse to the cloud-init script to run post instance creation operations during firstboot.
Here we are installing netperf tool and setting up the network configuration file using cloud-init script below.

```
#!/bin/bash

sudo rpm -ivh /root/netperf-2.7.0-1.el7.lux.x86_64.rpm

cat > /etc/sysconfig/network-scripts/ifcfg-ens6 <<EOF
DEVICE=ens6
BOOTPROTO=static
IPADDR=11.0.0.250
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Ethernet
NM_CONTROLLED=no
EOF
```

```
ethtool -K eth0 tso off
echo "ethtool -K eth0 tso off" >> /etc/rc.local
```



You can customise your instance after it has launched using the options available here. "Customisation Script" is analogous to "User Data" in other systems.

Load Customisation Script from a file

No file selected.

Customisation Script (Modified) Content size: 310 bytes of 16.00 KB

```
#!/bin/bash

sudo rpm -ivh /root/netperf-2.7.0-1.el7.lux.x86_64.rpm

cat > /etc/sysconfig/network-scripts/ifcfg-ens6 <<EOF
DEVICE=ens6
BOOTPROTO=static
IPADDR=11.0.0.250
NETMASK=255.255.255.0
```

Disk Partition

Automatic

□ **Configuration Drive**

6 SR-IOV Capable Instance Login

1. Find the floating ip of the instance from the OpenStack UI along with the host info from which the key-pair was used to create the instance.
2. Have 3 terminal sessions and use each of the sessions to connect to the jump server and then to the host in the previous step
3. From one session, "ssh" using the floating ip to the instance from the host

```
ubuntu@<host-with-keypair>:$ sudo su
ubuntu@<host-with-keypair>:# ssh centos@<Instance external IP>
```

```
ubuntu@<dbdw16>:# ssh centos@192.168.50.94
```

Note: If you see WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! during login to the instance, run below command with the instance ip address. You will also prompted to run the following command at the bottom of the warning message, you can just copy paste the command and hit enter.

```
ubuntu@<hostname>:# ssh-keygen -f "/root/.ssh/known_hosts" -R <Instance external IP>
```

4. Need to repeat the same steps for 2 other instances for 2 other terminal sessions

7 Assigning IP Address to the SR-IOV Ports in the CentOS Instance

1. Create **ifcfg-<device-name>** file and add below lines to it. Make sure the IPADDR is of your instance sriov-net ip. Most of the time device-name will be **ens6**.

```
# vi /etc/sysconfig/network-scripts/ifcfg-<device-name>
DEVICE=<device-name>
BOOTPROTO=static
IPADDR=<IP Address provided>
NETMASK=255.255.255.0
ONBOOT=yes
```

```
TYPE=Ethernet
NM_CONTROLLED=no
```

```
# vi /etc/sysconfig/network-scripts/ifcfg-ens6
DEVICE=ens6
BOOTPROTO=static
IPADDR=11.11.1.21
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Ethernet
NM_CONTROLLED=no
```

2. Restart the network service or instance.

```
#service network restart
```

```
# service network restart
```

3. Repeat the same steps for the 2 other instances in their own terminal sessions

8 Latency Test

8.1 East-West

Of the 2 Instances on the same compute node, pick one of your choice to ping the other instance.

1. From the terminal session of the selected instance, ping floating IP of other instance

```
[centos@<instance1-name>~]$ping  -c 10 <floating ip of other instance>
```

```
[centos@<instance1-name>~]$ ping -c 10 192.168.50.65
PING 192.168.50.65 (192.168.50.65) 56(84) bytes of data.
64 bytes from 192.168.50.65: icmp_seq=1 ttl=63 time=1.33 ms
```

2. From the terminal session of the selected instance, ping guest-net IP of other instance.

```
[centos@<instance1-name>~]$ping -c 10 <guest-net ip of other instance>
```

```
[centos@<instance1-name>~]$ ping -c 10 10.5.0.21
```

```
PING 10.5.0.21 (10.5.0.21) 56(84) bytes of data.
64 bytes from 10.5.0.21: icmp_seq=1 ttl=64 time=0.639 ms
```

3. From the terminal session of the selected instance, ping sr-iov IP of other instance

```
[centos@<instance1-name>~]$ping -c 10 <SR-IOV ip of other instance>
```

```
[[centos@<instance1-name>~]$ ping -c 10 11.0.1.13
PING 11.0.1.13 (11.0.1.13) 56(84) bytes of data.
64 bytes from 11.0.1.13: icmp_seq=1 ttl=64 time=0.162 ms
```

8.2 North-South

Pick one instance from each compute node where the SR-IOV instances are created. Ping from one instance to another instance on different host.

1. From the terminal session of the selected instance, ping floating IP of other instance

```
[centos@<instance1-name>~]$ping  -c 10 <floating ip of other instance>
```

```
[root@sr-gr1-i1 centos]# ping 192.168.50.138 -c 10
PING 192.168.50.138 (192.168.50.138) 56(84) bytes of data.
64 bytes from 192.168.50.138: icmp_seq=1 ttl=64 time=0.674 ms
--- 192.168.50.138 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9000ms
rtt min/avg/max/mdev = 0.627/0.691/0.731/0.040 ms
```

2. From the terminal session of the selected instance, ping guest-net IP of other instance.

```
[centos@<instance1-name>~]$ping -c 10 <guest-net ip of other instance>
```

```
[root@sr-gr1-i1 centos]# ping -c 10 10.5.0.66
PING 10.5.0.66 (10.5.0.66) 56(84) bytes of data.
64 bytes from 10.5.0.66: icmp_seq=1 ttl=64 time=2.29 ms
--- 10.5.0.66 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9011ms
rtt min/avg/max/mdev = 0.813/1.132/2.294/0.407 ms
```

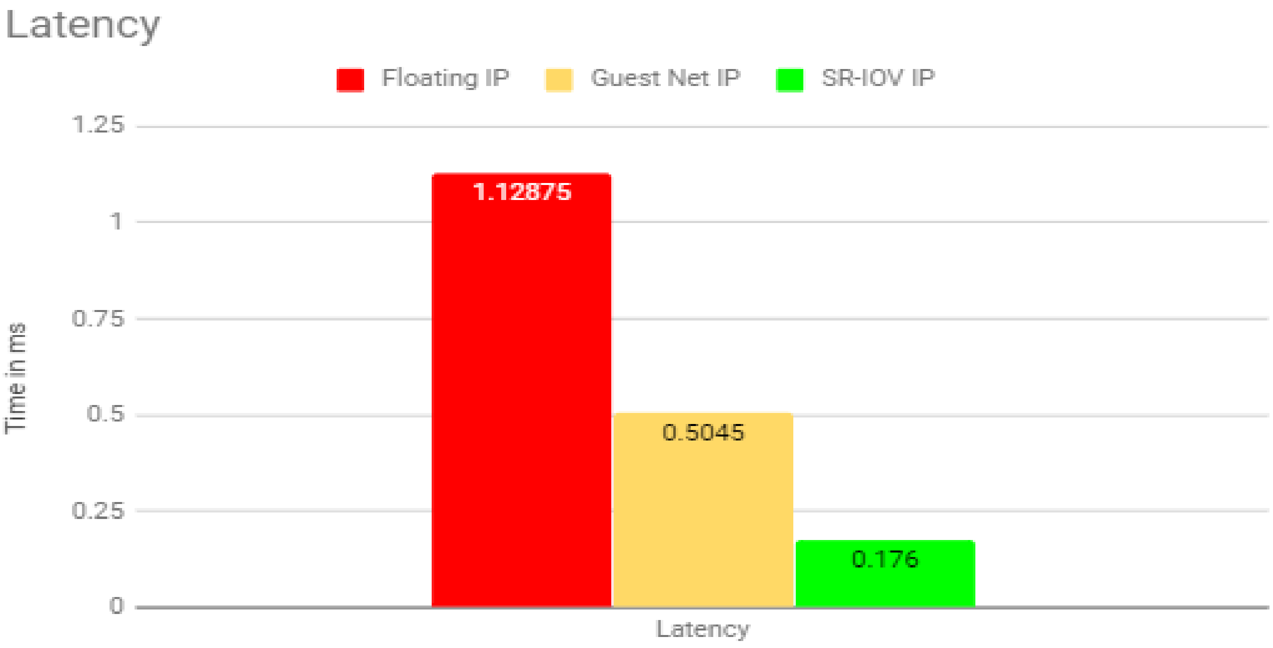
3. From the terminal session of the selected instance, ping sr-iov IP of other instance

```
[centos@<instance1-name>~]$ping -c 10 <SR-IOV ip of other instance>
```

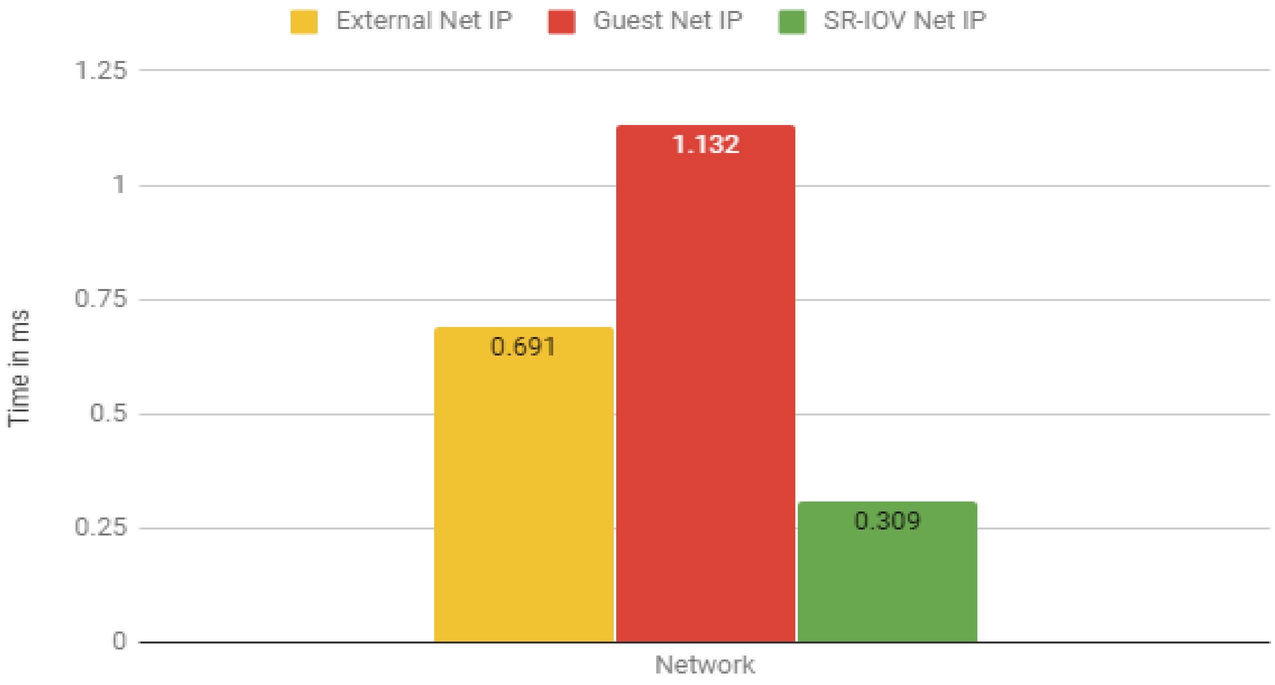
```
[root@sr-gr1-i1 centos]# ping -c 10 11.0.1.45
PING 11.0.1.45 (11.0.1.45) 56(84) bytes of data.
64 bytes from 11.0.1.45: icmp_seq=1 ttl=64 time=0.439 ms
--- 11.0.1.45 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9001ms
rtt min/avg/max/mdev = 0.231/0.309/0.543/0.099 ms
```

8.3 Latency Comparison Graphs

8.3.1 East - West



8.3.2 North - South



9 Running netperf

9.1 East-West

Of the 2 Instances on the same compute node, pick one of your choice as Net-Perf server and the other as net-perf client

1. From the terminal session of the selected instance, start “netperf server”

```
[centos@<instance1-name>~]$netserver -p 12346
```

```
[centos@<instance1-name>~]$ netserver -p 12346
Starting netserver with host 'IN(6)ADDR_ANY' port '12345' and family
AF_UNSPEC
[centos@<instance1-name>~]$
```

2. Start watching the IO on the sriov nic using the below command.

```
[centos@<instance1-name>~]$watch ifconfig <interface name of the SR-IOV port>
```

```
[centos@<instance1-name>~]$watch ifconfig ens6
```

3. Start netperf client on the **other** Instance which you chose as client and run test on sriov interface.

```
[centos@<instance2-name>~]$netperf -H <netperf server sriov ip> -p 12346 -l 120
```

```
[centos@<instance2-name>~]$ netperf -H 11.0.1.26 -p 12346 -l 120
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 11.0.1.26
() port 0 AF_INET
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

 87380  16384  16384    120.00   16365.81   #This represents 16.36Gbps
```

4. Once the test completes (120 seconds), note down the throughput displayed. 16365.81 Mbits/sec is in this case.

5. Re-start netperf client using the floating IP

```
#netperf -H <netperf server floating ip> -p 12346 -l 120
```

```
[centos@user3-sriov-vm2 ~]$netperf -H 192.168.50.55 -p 12346 -l 120
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to
192.168.50.55 () port 0 AF_INET
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

 87380  16384  16384    120.04    889.69    #This represents 0.88Gbps
```

6. Once the test completes (120 seconds), note down the throughput displayed. 889.69 Mbits/sec in this case.

7. Re-start netperf client using the guest-net IP

```
#netperf -H <netperf server guest net ip> -p 12346 -l 120
```

```
[centos@user3-sriov-vm2 ~]$ netperf -H 10.5.0.11 -p 12346 -l 120
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 10.5.0.11
() port 0 AF_INET
Recv   Send    Send
Socket Socket  Message  Elapsed
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

 87380  16384  16384    120.04    16345.01
```

8. Once the test completes (120 seconds), note down the throughput displayed. 889.69 Mbits/sec in this case.

9. Compare the throughput between steps 4, 6 & 8. Step 4 throughput is in general higher than the throughput in steps 6 & 8. See the highlighted part in steps 4, 6 & 8 to observe the difference.

10. If unable to ping between the instances, then check arp command. It should look like the below. If the SR-IOV IP does not have a MAC or Flag set, reboot the instance.

```
[root@user-7-sriov-vm1 centos]# arp
Address                      HWtype      HWaddress          Flags Mask        Iface
host-10-5-0-2.openstack  ether      fa:16:3e:49:93:09C                 eth0
11.0.0.3                     ether      fa:16:3e:c6:bc:aa C                ens6
host-10-5-0-9.openstack  ether      fa:16:3e:8f:ef:18        C                 eth0
host-10-5-0-1.openstack  ether      fa:16:3e:a0:62:fb        C                 eth0
```

9.2 North-South

Of the 2 Instances across the compute nodes, continue to use the one in previous section as Netperf server.
In the CentOS instance in the other compute, we can run net-perf client to check and validate network performance.

1. From the terminal session of the selected instance, start netperf client and run test on sriov interface.

```
#netperf -H <netperf server sriov ip in other compute> -p 12346 -l 120
```

```
[root@sr-gr1-i1 centos]# netperf -H 11.0.1.45 -p 12346 -l 120
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 11.0.1.45
() port 0 AF_INET
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

 87380  16384  16384    120.00   16984.98     #This represents 16.98Gbps
```

2. Once the test completes (120 seconds), note down the throughput displayed.

3. Re-start netperf client using the guest net IP

```
#netperf -H <netperf server guest net ip in other compute> -p 12346 -l 120
```

```
[root@sr-gr1-i1 centos]# netperf -H 10.5.0.66 -p 12346 -l 120
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 10.5.0.66
() port 0 AF_INET
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

 87380  16384  16384    120.03     769.61    #This represents 0.7Gbps
```

4. Once the test completes (120 seconds), note down the throughput displayed.

5. Re-start netperf client using the floating IP

```
#netperf -H <netperf server floating ip> -p 12346 -l 120
```

```
[root@sr-gr1-i1 centos]# netperf -H 192.168.50.138 -p 12346 -l 120
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to
192.168.50.138 () port 0 AF_INET
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec
```

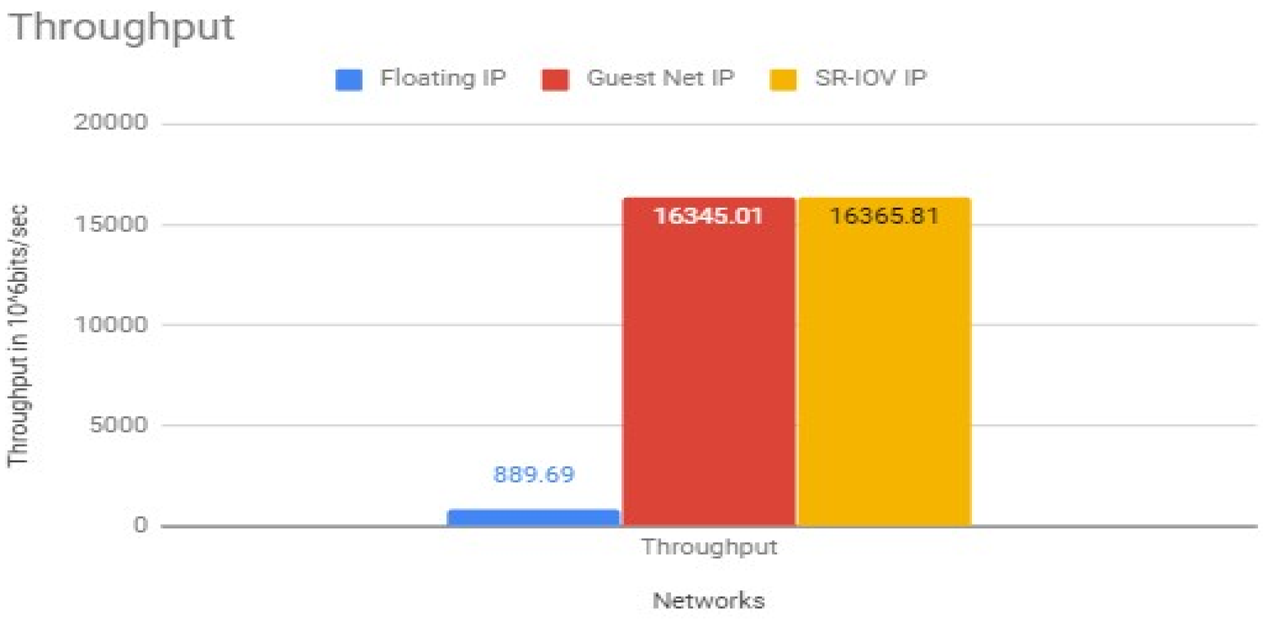
```
 87380  16384  16384     120.00   6910.54     #This represents 0.7Gbps
```

6. Once the test completes (120 seconds), note down the throughput displayed. 889.69 Mbits/sec in this case.

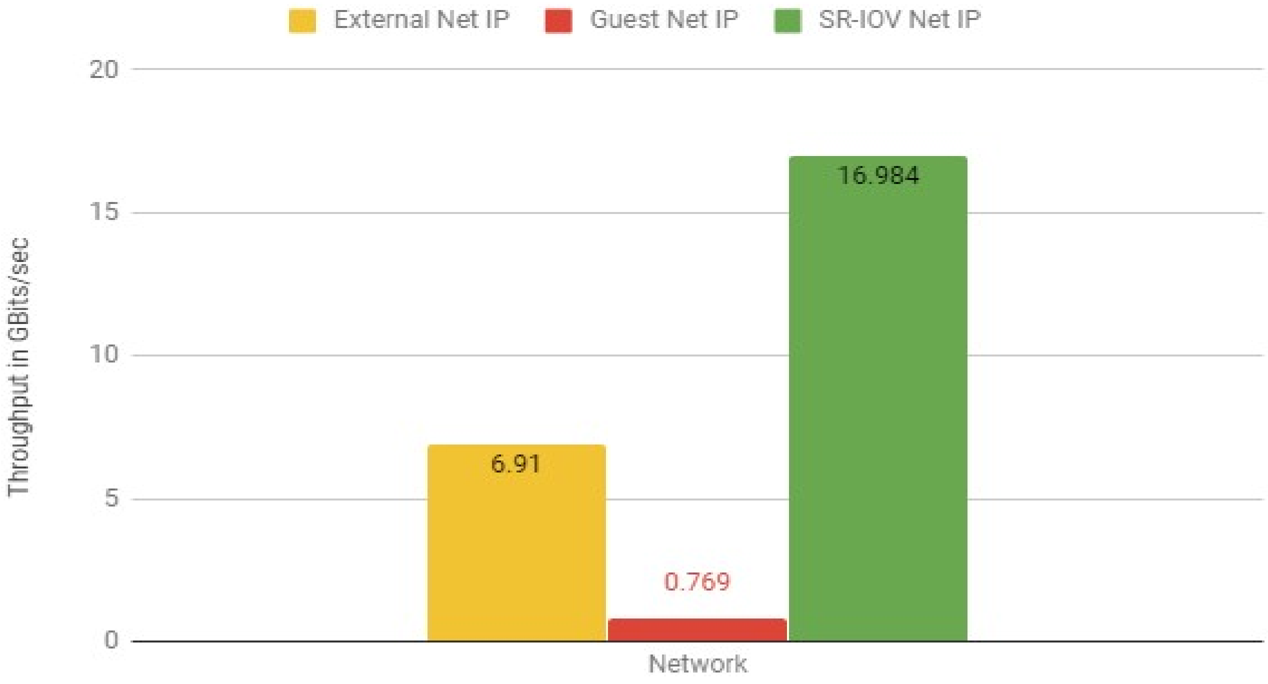
7. Compare the throughput between steps 2 & 4. Step 2 throughput is in general higher than the throughput in step 4. See the highlighted part in steps 2 & 4 to observe the difference.

9.3 Netperf Comparison Graphs

9.3.1 East - West



9.3.2 North - South



10 Appendix

10.1 SR-IOV Information

<TBD details on feature, advantages and limitations from Yardstick talk about TC0083 and vFW Sample VNF>

- On each Compute Host, make sure that all the Physical (PF) & Virtual Functions (VFs) are UP

10.2 Configuring SR-IOV on Openstack Environment

10.2.1 Compute Node Preparation

Configuration changes before starting the deployment of openstack cluster using openstack-ansible.

- By following the below openstack-ansible deployment guide deploy openstack cluster.

  https://docs.openstack.org/project-deploy-guide/openstack-ansible/queens/

- To enable SR-IOV support on openstack deployment, we need to make few changes to couple of files before starting the deployment.
    - We need to enable Virtual Functions(VFs) on a Physical Function(PFs) before starting the installation.
    - Make sure to bring PF and all the associated VFs interfaces up.
    - Add add sriov agent under group_binds for a SR-IOV specific network in the following file ***/etc/openstack_deploy/openstack_user_config.yml***

      ```
      - network:
              container_bridge: "br-flat"
              container_type: "veth"
              container_interface: "eth12"
              type: "flat"
              net_name: "provider"
              group_binds:
                - neutron_linuxbridge_agent
                - neutron_sriov_nic_agent
      ```

    - Add SR-IOV overrides in the following file ***/etc/openstack_deploy/user_variables.yml***

      ```
      neutron_plugin_types:
        - ml2.lxb
        - ml2.sriov
      neutron_ml2_drivers_type: "flat,vlan,vxlan"


      neutron_ml2_conf_ini_overrides:
        securitygroup:
          firewall_driver: "iptables_hybrid"
      neutron_l2_population: true
      neutron_provider_networks:
        network_flat_networks: "*"
        network_types: "vxlan,flat,vlan"
        network_mappings: "provider:eth8,vxlan:br-vxlan"
        network_vxlan_ranges: "10000:50000"
        network_sriov_mappings: "physnet_sriov:eth9"
      ```

```
nova_nova_conf_overrides:
    pci_passthrough_whitelist: '{"vendor_id": "8086","product_id": "154c",
"address":"af:.*", "physical_network":"physnet_sriov"}'
    filter_scheduler:
        enabled_filters: "RetryFilter, AvailabilityZoneFilter, RamFilter,
AggregateRamFilter, ComputeFilter, AggregateCoreFilter, DiskFilter,
AggregateDiskFilter, AggregateNumInstancesFilter, AggregateIoOpsFilter,
ComputeCapabilitiesFilter, ImagePropertiesFilter, ServerGroupAntiAffinityFilter,
ServerGroupAffinityFilter, NUMATopologyFilter, PciPassthroughFilter,
AggregateInstanceExtraSpecsFilter"
```

Refer below link to configure overides.

https://docs.openstack.org/project-deploy-guide/openstack-ansible/newton/app-advanced-config-override.html

- Once the above change were made, we can start the deployment of openstack.
- Once the deployment is done check all the required agent files are created or not under ***/etc/neutron/plugins/ml2/*** Verify all the ini files and make sure the configuration was set properly. Use the below links for reference.

  https://docs.openstack.org/mitaka/networking-guide/config-sriov.html

  https://www.cavium.com/solutions/Documents/UsersGuide_OpenStack_SR-IOV.pdf

- Edit the ***/etc/nova/nova.conf*** file and add below lines on each compute node based on the "***lspci -nn | grep net"*** output. Make sure this parameter was set according to the pci interface info on respective hosts.

```
pci_passthrough_whitelist = {"vendor_id": "8086","product_id": "154c",
"address":"af:0a.*", "physical_network":"physnet_sriov"}
```

10.2.2 Openstack Dashboard Specific Changes

- Create a network for SR-IOV with Provider Network Type as Flat/VLAN and for Physical Network, mention the name provided for physical_device_mappings in sriov_nic_agent.ini file
- Under SR-IOV NW, create an SR-IOV port of type **Direct**
- Create seperate availability zones for each SR-IOV configured host, so that instances can be created based on need/suitability for a particular use-case/application
- OpenStack does not support hot-plugin of SR-IOV ports - meaning we cannot add an SR-IOV port to an existing Instance
- While creating instances, select a guest NW. DO NOT select SR-IOV NW as a

guest NW. Instead in the **Network Ports** page of the instance creation wizard, select an SR-IOV port

- While creating instances, use centos#7 image only. SR-IOV interfaces are not detected on ubuntu images
- Live migration of instances using SR-IOV ports is not supported

10.3 Reference

Add references of SR-IOV from Intel and OpenStack <TBD>

https://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/xl710-sr-iov-config-guide-gbe-linux-brief.pdf

https://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/sr-iov-nfv-tech-brief.pdf

https://docs.openstack.org/neutron/rocky/admin/config-sriov.html#known-limitations

https://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/sr-iov-nfv-tech-brief.pdf

10.4 Netperf Info & Help

Netperf is a benchmark that can be used to measure various aspects of networking performance. Its primary focus is on bulk data transfer and request/response performance using either TCP or UDP and the Berkeley Sockets interface. There are optional tests available to measure the performance of DLPI, Unix Domain Sockets, the Fore ATM API and the HP HiPPI LLA interface.

TCP Stream Performance:

```
$ /opt/netperf/netperf -H remotehost
```

UDP Stream Performance:

```
$ /opt/netperf/netperf -H remotehost -t UDP_STREAM -- -m 1024
```

DLPI Connection Oriented Stream Performance:

```
$ /opt/netperf/netperf -H remotehost -t DLCO_STREAM -- -m 1024
```

DLPI Connectionless Stream:

```
$ /opt/netperf/netperf -H remotehost -t DLCL_STREAM -- -m 1024
```

10.4.1 References

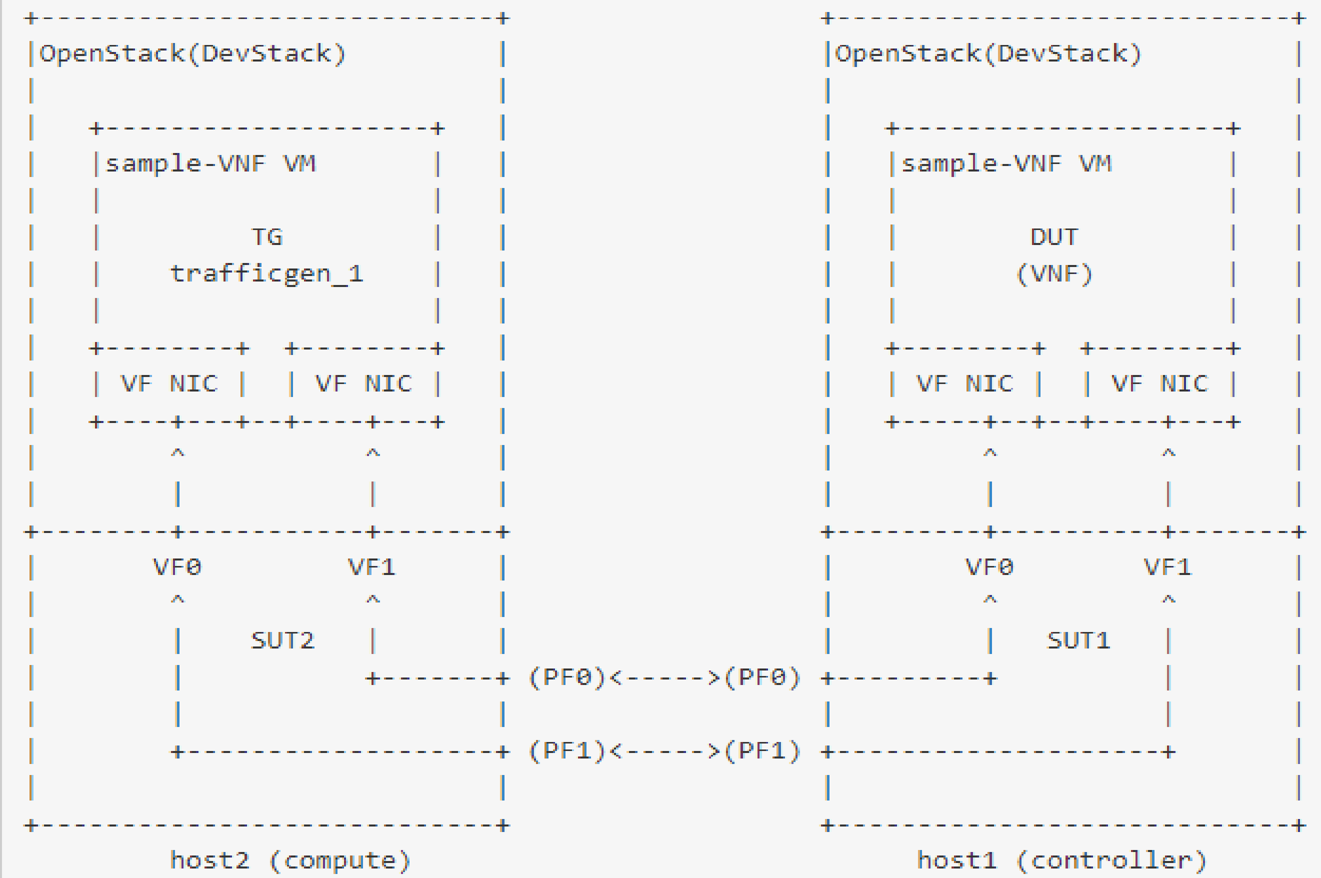
10.5 Yardstick Tests

https://opnfv-yardstick.readthedocs.io/en/stable-gambia/testing/user/userguide/13-nsb-installation.html#network-service-benchmarking-openstack-with-sr-iov-support

https://opnfv-yardstick.readthedocs.io/en/stable-gambia/testing/user/userguide/13-nsb-installation.html#multi-node-openstack-tg-and-vnf-setup-two-nodes

https://github.com/opnfv/yardstick/blob/c9b1716fd37284d09b4c327b65fd5d7c8769fc7f/docs/testing/user/userguide/opnfv_yardstick_tc083.rst

10.5.1 Multi node OpenStack TG and VNF setup(Two nodes)



```
+----------------------------+                   +----------------------------+
|OpenStack(DevStack)         |                   |OpenStack(DevStack)         |
|                            |                   |                            |
|   +--------------------+   |                   |   +--------------------+   |
|   |sample-VNF VM       |   |                   |   |sample-VNF VM       |   |
|   |                    |   |                   |   |                    |   |
|   |         TG         |   |                   |   |        DUT         |   |
|   |    trafficgen_1    |   |                   |   |       (VNF)        |   |
|   |                    |   |                   |   |                    |   |
|   +--------+  +--------+   |                   |   +--------+  +--------+   |
|   | VF NIC |  | VF NIC |   |                   |   | VF NIC |  | VF NIC |   |
|   +----+---+--+----+---+   |                   |   +-----+--+--+----+---+   |
|        ^           ^       |                   |         ^          ^       |
|        |           |       |                   |         |          |       |
+--------+-----------+-------+                   +---------+----------+-------+
|       VF0         VF1      |                   |        VF0        VF1      |
|        ^           ^       |                   |         ^          ^       |
|        |   SUT2    |       |                   |         |   SUT1   |       |
|        |           +-------+ (PF0)<----->(PF0) +---------+          |       |
|        |                   |                   |                    |       |
|        +-------------------+ (PF1)<----->(PF1) +--------------------+       |
|                            |                   |                            |
+----------------------------+                   +----------------------------+
         host2 (compute)                               host1 (controller)
```