Dell EMC Red Hat OpenStack NFV Solution v6.0 - SR-IOV Support Guide

Contents

Trademarks

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Red Hat®, Red Hat Enterprise Linux®, and Ceph are trademarks or registered trademarks of Red Hat, Inc., registered in the U.S. and other countries. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries. Oracle® and Java® are registered trademarks of Oracle Corporation and/or its affiliates.

DISCLAIMER: The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/ service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries, and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation or the OpenStack community.

The Midokura® name and logo, as well as the MidoNet® name and logo, are registered trademarks of Midokura SARL.

Notes, Cautions, and Warnings

A **Note** indicates important information that helps you make better use of your system.

A **Caution** indicates potential damage to hardware or loss of data if instructions are not followed.

A **Warning** indicates a potential for property damage, personal injury, or death.

This document is for informational purposes only and may contain typographical errors and technical inaccuracies. The content is provided as is, without express or implied warranties of any kind.

Glossary

BMC/IDRAC Enterprise

Baseboard management controller. An on-board microcontroller that monitors the system for critical events by communicating with various sensors on the system board and sends alerts and log events when certain parameters exceed their preset thresholds.

Bundle

A customer-orderable solution that consists of:

- All server, network, and storage hardware needed to install and operate the solution as outlined
- All necessary solution software licenses needed to install and operate the solution as outlined

Cloud Computing

See *http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf*

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cluster

A set of servers dedicated to OpenStack that can be attached to multiple distribution switches.

Compute Node

The hardware configuration that best supports the hypervisor server or Nova compute roles.

CNode

CNode is referred to as compute node.

DevOps

Development Operations (DevOps) is an operational model for managing data centers using improved automated deployments, shortened lead times between fixes, and faster mean time to recovery. See *https://en.wikipedia.org/wiki/DevOps*.

HA

High Availability or HA is the level of operational performance, usually uptime, for a higher than normal period. .

Hypervisor

Software that runs virtual machines (VMs).

IaaS

Infrastructure as a Service.

Infrastructure Node

Systems that handle the control plane and deployment functions.

IOMMU

Input/Output Memory Management Unit.

ISV

Independent Software Vendor.

IXGBE

IXGBE is Intel 10Gbps driver for linux operating system.

LAG

Link Aggregation Group.

LOM

LAN on motherboard.

NIC

Network Interface Card.

Node

One of the servers in the cluster.

OSP

OpenStack Platform or OSP is a RedHat product that includes all the modules of OpenStack.

Overcloud

The functional cloud that is available to run guest VMs and workloads.

PCIe

Peripheral Component Interconnect Express (PCIe or PCI-E) is a serial expansion bus standard for connecting a server to NIC adaptors and/or more peripheral devices.

PCI SIG

The PCI-SIG or Peripheral Component Interconnect Special Interest Group is an electronics industry consortium responsible for specifying the Peripheral Component Interconnect, PCI-X, and PCI Express computer buses.

PF

A Physical Function or a PF is a physical NIC card that is used to create virtual functions (VFs).

Pod

An installation comprised of three racks, consists of servers, storage, and networking.

SAH

The Solution Admin Host (SAH) is a physical server that supports VMs for the Undercloud machines needed for the cluster to be deployed and operated.

SDS

Software-defined storage (SDS) is an approach to computer data storage in which software is used to manage policy-based provisioning and management of data storage, independent of the underlying hardware.

SFP+ and QSFP

Small Form-factor Pluggable or Quad Small Form-factor Pluggable is a tranceiver used for data communication with NIC cards.

SR-IOV

Single Root Input/Output Virtualization or SR-IOV is a network interface that allows the isolation of the PCI Express resources for manageability and performance reasons. A single physical PCI Express can be shared on a virtual environment using the SR-IOV specification.

Storage Node

The hardware configuration that best supports SDS functions such as Red Hat Ceph Storage.

ToR

Top-of-rack switch/router.

Undercloud

The Undercloud is the system used to control, deploy, and monitor the Overcloud. The Undercloud is *not* HA configured.

VF

A Virtual Function or a VF is a virtual PCIe device that is used to share a physical NIC in a virtualized environment.

VLT

A Virtual Link Trunk (VLT) is the combined port channel between an attached device (ToR switch) and the VLT peer switches.

VMM

A Virtual Machine Manager or a VMM is software used to create, operate, modify and manage the lifecycle of VMs running on a Linux operating system.

VNF

A Virtualized Network Function, or VNF, is responsible for handling specific network functions that run in one or more virtual machines on top of the hardware networking infrastructure, which can include routers, switches, servers, cloud computing systems and more.

VLTi

A Virtual Link Trunk Interconnect (VLTi) is an interconnect used to synchronize states between the VLT peer switches. Both endpoints must be on 10G or 40G interfaces; 1G interfaces are not supported.

SR-IOV Enablement in Dell EMC Red Hat OpenStack NFV Solution v6.0

SR-IOV is an abbreviation for "Single Root I/O Virtualization". The SR-IOV technology has been available since ca. 2008, and has been enabled in several generations of Dell EMC's server line-up, as well as in Red Hat's Enterprise Linux platform. So far, however, the use of this technology has been quite cumbersome, with the end-user expected to make significant programmatic decisions about various parameters around the use of SR-IOV.

In the Dell EMC Red Hat OpenStack NFV Solution v6.0, SR-IOV has been made simple to consume as an external network connectivity mechanism to typical VNFs (virtual network functions). The design is based on the following guiding principles:

1. **Wire-speed peformance**: SR-IOV provides the ability to take a physical port (Physical Function or PF), and logically slice it into Virtual Functions or VFs. Each VF can be programmed to provide a certain amount of bandwidth to the VNF connected to it. It is the primary objective of the SR-IOV facility provided in this solution to enable *end to end* wire-speed performance to the extent possible.
2. **Fail-over support**: Sometimes referred to as HA or high-availability, the SR-IOV implementation in this solution support physical port, wire, NIC, and first-level switching element failover. If any of these elements fails, connectivity to the Virtual Functions (a pair of VFs is always exposed to a VNF) is not compromised. Application-level HA models are overlayed on top of the SR-IOV connectivity, and can be independently implemented based on the needs of the VNFs and the service function chains within which they might be individual links.
3. **Co-existence with OpenStack platform native networks**: This SR-IOV implementation does not make an attempt to integrate with OpenStack networks. Indeed, the built-in tenant and external networks supported by Red Hat OpenStack Platform, and by the Dell EMC Red Hat cloud solution are untouched. Individual VNF instances are created, managed, and lifecycle-managed by the OpenStack Platform. The SR-IOV networks are additional networks that co-exist with the OSP networks, and provide a direct, external path into and out of the VNFs. This makes them an excellent choice for functions typically placed at traffic ingress/egress points of an NFV solution, e.g. a firewall, a session-border controller, or, an application-delivery controller.
4. **Separate network resources**: The SR-IOV implentation requires separate NICs, SPF+ connectors, cables, and switch ports to carry SR-IOV enabled traffic. The SR-IOV networks are not routed through the Master Controller of the OpenStack Platform, for doing so would require the use of a shared resource (the virtual network router) in the Master Controller, thus breaking the promise of end-to-end wire-speed performance to and from a VNF.

The SR-IOV implementation outlined in this support guide is strictly optional. It is expected to be carried out by a Red Hat Services professional, post-deployment of an instance of the Dell EMC Red Hat OpenStack NFV Solution v6.0. Please proceed with caution when attempting the enablement of this feature, for it may render a Compute node unusable immediately following an erroneous execution of the enablement procedure. If that happens, the only recourse is re-deployment of the Compute node via procedures outlined in other guides accompanying the solution.

The implementation consists of a Python script which automates the enablement of SR-IOV virtual functions in the solution. The rest of this document outlines the high-level architecture extension to the base Dell EMC Red Hat OpenStack NFV Solution Reference Architecture. Further, the document describes the two passes in which the enablement script is invoked: first, to create, program, and validate the SR-IOV Virtual Functions in a Compute node, and second: to assign, expose, bond, and validate the use of a pair of the Virtual Functions to a running VM instance under the control of the OpenStack Plaform. Several requirements that must be met in order for this implementation to be successful are outlined. Finally, details of typical successful execution of each script pass, along with a list of possible errors that a solutions professional may encounter, are presented.

Reference Architecture Overview using SR-IOV

The Cluster network architecture including Compute nodes enabled to use SR-IOV is shown in Figure *Figure 1: Logical Network Architecture using SR-IOV Enabled Compute Nodes in Dell EMC Red Hat OpenStack NFV Solution v6.0* on page 11.

The figure below is the updated reference architecture of Dell EMC Red Hat OpenStack NFV Solution v6.0 in combination of SR-IOV solution architecture. The outlined box in the diagram below reflects the enhancements for SR-IOV enablement carried out on top of Dell EMC Red Hat OpenStack NFV Solution v6.0 standard architecture.

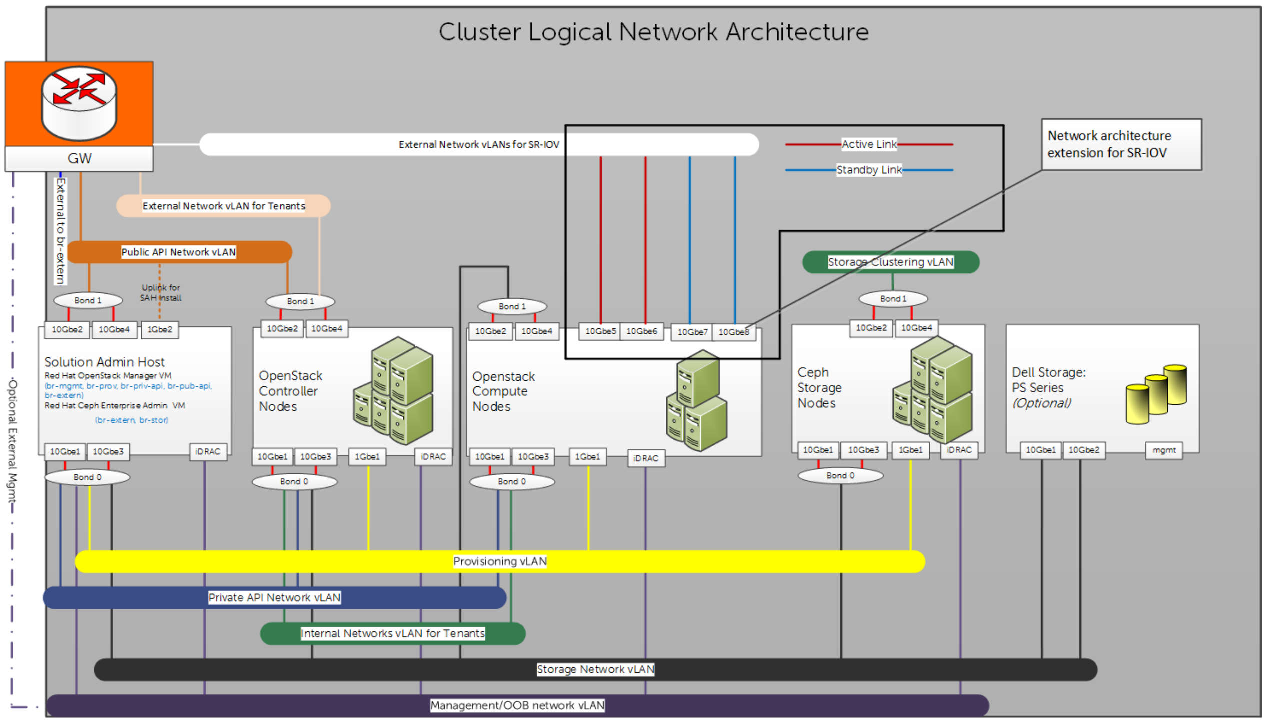


Figure 1: Logical Network Architecture using SR-IOV Enabled Compute Nodes in Dell EMC Red Hat OpenStack NFV Solution v6.0

Dell EMC SR-IOV solution is powered by Intel®X520 dual port 10G NIC adaptors. The NIC adaptors are SR-IOV capable and therefore provides wire-speed performance in an OpenStack environment. SR-IOV scripts automate the following key functionalities in Dell EMC RedHat NFV solution 6.0:

Python based SR-IOV scripts will provide zero-touch provisioning to enable SR-IOV and create virtual functions on respective compute nodes. SR-IOV scripts will assign the bandwidth of each virtual function. In curent architecture, each VF is assigned 2.5Gbps of bandwidth. A total of 4 VFs can be used to occupy the 10Gbps bandwidth of Intel®X520 NIC port.

SR-IOV scripts will automatically assign the VFs to the instance. The script takes input of the instance in the settings.ini file and generate XML code and modify the XML file of the instance.

SR-IOV scripts also creates the bonds inside the instances, this provides the virtual NIC level HA inside the instance. The bond configured inside an instance consists of 2 VFs from 2 different PFs. If one of the VF/ PF is down, the traffic will be re-routed through standby VFs and hence providing no service distruption to VNF.

In addition to standard NFV 6.0 compute nodes, SR-IOV enablement requires additional hardware in each compute node. The additional hardware includes dual port Intel X520, SFP, QSFP connectors, and fiber breakout cables.

While enabling SR-IOV on Dell EMC RedHat NFV 6.0, new networks will be configured on physical switches, in addition to default NFV 6.0 networks. To assign the networks, VLANs will configured on the physical switches and the compute node ports will be added to the VLAN, in access mode. This will provide network connectivity between physical functions and the external network gateway. SR-IOV requires an external network which should be routeable to datacenter and internet. And all the IP traffic from the VNF instance to the exteral network is routed through a unique VLAN ID on the lead switch.

SR-IOV Reference Topology

A reference SR-IOV topology is shown in Figure *Figure 2: An SR-IOV-enabled Compute Node in Dell EMC Red Hat OpenStack NFV Solution v6.0* on page 12.

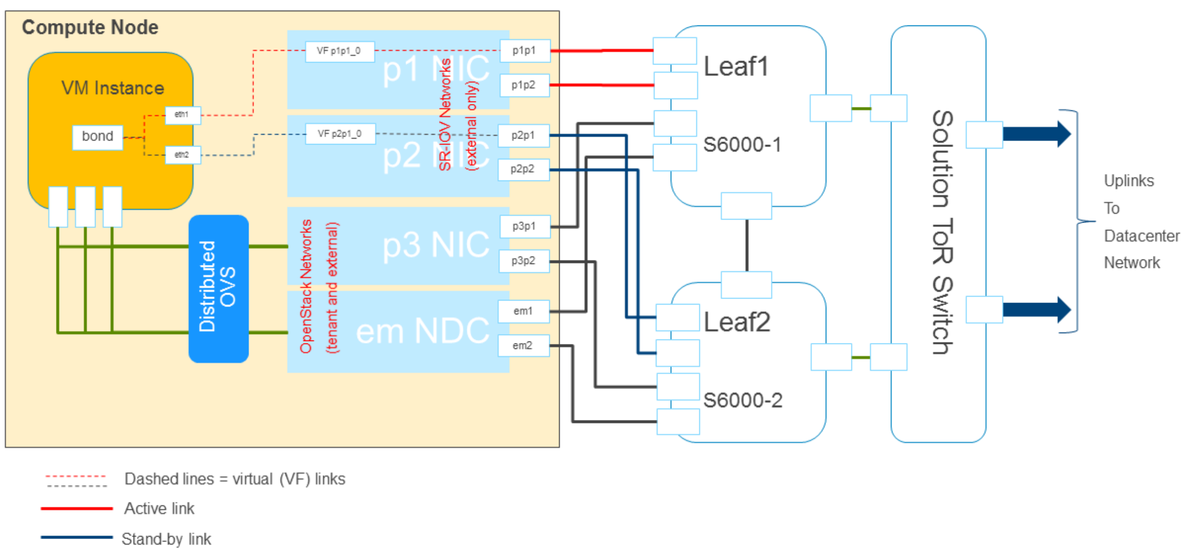


Figure 2: An SR-IOV-enabled Compute Node in Dell EMC Red Hat OpenStack NFV Solution v6.0

1. **Compute Node Network Adaptors:** Each compute node in Dell EMC RedHat NFV 6.0 is equipped with 4xIntel X520 dual port network adaptors. 2xIntel X520 are connected to distributed OVS running in compute nodes. However, additional 2xIntel X520 are used as physical functions to enable SR-IOV. These two additional adaptors consists of 4 x 10G ports, lableled as p1p1, p1p2, p2p1 and p2p2. These 4x10G ports are referred to as physical functions or PFs.
2. **Physical Functions:** Physical Function is a unit 10G port of Intel X520 adaptors that is used to create VFs on top of it. Physical Function behaves like a L2 switch and performs traffic forwarding between wire and VFs.
3. **Virtual Functions:** Virtual Functions (VFs) are the virtual PCIe slots created in ixgbe driver against each NIC port on a compute node. Each VF has a unique PCIe bus ID, which is used to distinguish the PF correspoding to that VF. The PCIe ID of VF is also used fpr assignment of VF to an instance.
4. **Uplink from solution ToR to Datacenter** It is recommended to use atleast 2x10G uplink from solution ToR switch to external datacenter. In a fully populated SR-IOV environment, if all the VFs are fully consumed at the allocated bandwidth i.e. 2500Mbps, considering 4 VFs/PF, it will generate 20Gbps of traffic. Therefore, to avoid any bandwidth botttleneck it is highly recommended to use atleast 2x10G uplink from solution ToR to external datacenter network.

5. **Active/Standby Links** Dell EMC SR-IOV architecture provides full Highly Availability on a PF level. If one of the PF goes down, the standby PF will take over and start forwarding the traffic, without any disruption to VNF service.
6. **Physical Switches Topology** Dell EMC SR-IOV architecture consists of 2xLeaf switches and 1xToR switch. Two PFs are connected to separate leaf switches. For Example: In figure 2, p1p1 is connected to Leaf1 switch, and p2p1 is connected to Leaf2 switch. In a scenario if one of the Leaf1 switch goes down, the traffic will be routed through standby PF (p2p1) via Leaf2 switch. Therefore, providing end-to-end high availability in the solution.

Dell EMC NFV SR-IOV scripts Architecture

Dell EMC SR-IOV scripts architecture provides a python based enable_sriov.py script which has two script passes that will automate the setup of SR-IOV and assignment of VFs to instances. Following is the architectural description of each passes:

cnode pass

PCI-SIG developed Single Root I/O Virtulization (SR-IOV)to provide sharing of I/O subsystems. A single physical function is seen as multiple virtual functions and hence the name SR-IOV. In the PCI list commands these devices are seen as virtual functions. SR-IOV allows direct access of resources bypassing the Vitual Machine Monitor(VMM). The ability of SR-IOV lies in the fact that it can create multiple virtual nic's, called virtual function, from a single physical function (A physical function can be a ethernet port). SR-IOV is enabled on Intel®X520 adapters, which part of Dell EMC NFV 6.0 solution.

CNode pass is the first among the two scripts that should be should be run. cnode pass enables SR-IOV on compute nodes in Dell EMC RedHat NFV solution 6.0. Before starting to execute the CNode pass scipt make sure all the pre-requisites are met as mentioned in the pre-requisite section. The script has the checks to verify if all the pre-requisites are met.

Each compute node has a settings file where the user need to input the paramaters required for executing the script. The PF's on which you wish to enable SR-IOV should be passed in through the settings file. The PF's are given as pairs in the settings file. Each pair will have PF from a different NIC i.e, PFpair1 will have a pf from NIC in slot 1 which is p1p1 and another pf from a NIC in slot 2 which p2p2. Each PF in a PFpair goes into a separate switch. It is hightly recommended that you include PF's from different NIC's in a pair, this provides NIC level redundancy in our solution. The number of vf's per pf supported in this release in 4 and the bandwidth per PF should either be 10Gbps or 40Gbps. The bandwidth among the vf's is equally distributed. Intel Ethernet devices supporting SR-IOV supports the ability to set rate on the vf's.

Cnode pass has two modes:

1. **Ephemeral :** Ephemeral mode of cnode pass is the non-persistent setup of SR-IOV on NFV 6.0. It enables the SR-IOV environment on compute node and will be reset after reboot of the compute node. All the vf's and the rates set are no longer present after the reboot.
2. **Persistent :** Persistent mode of cnode pass will create the SR-IOV environment that would be persistent across multiple reboots of the compute nodes. This is done by creating a file ifup-dell-nfv-lcl in /sbin and adding all the vf creation commands, vf rate set commands in it and this file is called from /etc/sysconfig/network-scripts/ifup-post at boot up time. If this file is missing, persistence cannot be enabled. In order not corrput the ifup-post file the script creates the bakup of original file and then proceeds further.

Recommendation

It is recommended for users to use “Ephemeral” mode if the setup is being deployed for the first time. After initial validation, if there is a need to make it permanent, then switch on the “Persistent” and re-run the Cnode Pass

instance pass

Instances are created in Red Hat OpenStack Platform via Horizon. Launching an instance is a pre-requisite before starting the instance pass. The compute nodes on which the instances are hosted is completely opaque to the user.Also as a best pratice always make sure you run the cnode pass atleast once. However checks are in place to let the user know if the instance pass is executed before the cnode pass. The input parameters for the instance is taken from the settings file. A separate setting file should be created every time a new instance is created, this keeps the instances independent of each other. Instance pass looks for the instance name from the settings file and the instance name should match the name assigned during instantiation from Openstack Horizon. Another key parameter in the settings file of the instance pass is the key file. Make sure you have the right key file mapped to your instance. The name of key file should match the key file name that is given in the settings file. Generally each user in an instance has its own key, so the key file that is given in the settings file of instance pass should be the key file of the user that will login. Login to an instance using the user id and the corresponding key file atleast once after launching the instance and before running the instance pass.

The instance pass is the final step in the enablement of SR-IOV based solution on Dell EMC NFV 6.0. Instance pass assigns VF's to the instance, provided the instance is active. This is done by choosing a pair of available VF's. The VF pair contains equivalent VF numbers from a PF pair i.e, the VF's p1p1_0 and p2p1_0 will form the first VF pair taken from the PF pait p1p1 and p2p1.The precedence in which the VF pair is choosen is 0,1,2,3. The order in which the VF's are assigned is fixed and unaltered. The VF's are first assigned from the p1p1,p2p1 pair and then p1p2, p2p1 pair. The VF's from the second pair is chosen after all the VF pairs of the first PF pair is consumed. The life cycle of the VF's that are attached to instance is valid for the life time of the vm. The VF's are given back to the pool if the instance is rebooted. This pair of VF's is available for re-assignment to the current insance after it is brought back again or it can be assigned to a new instance.

Each VF has an xml config file which contains the PCI details of the VF's. Using these PCI details we can determine the order of the VF's. The assignment of the VF's to the instance is done using the virsh tool. The virsh tool uses these xml config files of the VF's for attaching it to the instance. This attachment is done while the instance is active and running, this is called live attach. Once the attachment of the VF's to the instance is done, inside the instance these interfaces are identified by differnt name in different vm images. The only way to determine the interfaces inside the instance is by comparing the mac address of the interfaces with the mac address of the corresponding VF's. Also an important point to note is that, the mac address of the interfaces inside the VF's is randomly generated and may not always be the same for an instance. Once the instance is rebooted the VF interfaces inside the instance is no longer present. Upon reassignment of the old VF pair there is no guarentee that it will assign the interfaces with the same mac address that it was previously assigned.

After successfully provisioning the VFs to the instance, the script will login into the instance using the credentials provided in the instance pass settings file. The script identifies the new interfaces and bonds the newly attached interfaces. Bonding is the only supported mode for this release in Dell EMC Red Hat OpenStack NFV Solution v6.0. The bond device is then assigned with an ip address, subnet mask and a gateway ip as provided in the instance pass settings file. This release supports only active-standby mode of bonding. In active-standby by mode one interface is choosen to be active and other interface is in standby.

Followings are the parameters that must be provided with the bonding command at the bonding time enablement:

1. miimon or apr_interval
2. arp_ip_target
3. fail_over_mac

These parameter ensures that we have HA at the instance level as well. It is critical either miimon or arp_interval and arp_ip_target is specified, otherwise serious network degradation occurs during link failure. We made the choice of using arp as our method of link monitoring for HA specific reasons. Additional information on bonding modes, link monitoring and fail_over_mac is provided in *https://www.kernel.org/doc/Documentation/networking/bonding.txt* The bonding mode can only be set to activestandy from the instance pass settings file for this release. A default route to the gateway tor is also inside the instance after the ip assignment. Finally, the instance pings the TOR to verify that external network connectivity is provided to the configured instance. As a best practice always ensure that the leaf switch is configured with required configuration before the instance pass, so that the ping is successful. Please refer to network and switch configuration section to know about required switch configuration detials.

Additional SR-IOV notes

1. SR-IOV scripts require a pair of two Physical Functions. PF in each pair should be part of different X520. For Example: In pair 1, first PF is P1P1 then the second PF should belong to slot P2, in this case, P2P1. Such pairing mechanism is used to provide NIC level redundancy, in case if one of the NIC goes down.
2. While assigning the VFs to an instance, script will assign first VF from the first PF in the pair, and second VF from the second PF in the pair.

Preparation for SR-IOV enablement

Following are the preliminary list of pre-conditions for SR-IOV to be installed:

Hardware Requirements

1. For NFV 6.0 full deployment, see the for Dell EMC RedHat NFV 6.0 deployment guide.

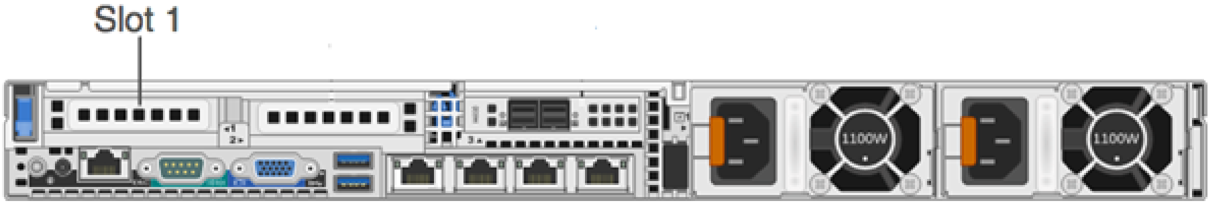
   Note: The compute nodes, as specified in the solution reference architecture, will be either Dell R630 or R730 servers.

   Note: SR-IOV enablement described in this guide was validated on Dell S6000 switches. Replace the configurations herein with suitable configuation as it pertains to your switch hardware.
2. In addition to the standard NFV 6.0 Hardware following additional Hardware is required:
   a. 6 x Intel® X520 (2 per Compute Node)
      a. 2 x Intel ® X520 will be installed in each compute node at following slots respectively:
         - PCI slot 1
         - PCI slot 2
   b. 4 x QSFP connectors
      a. 2 x QSFP connector will be connected to S6000-1
      b. 2 x QSFP connector will be connected to S6000-2
   c. 12 x SFP connectors
      a. The SFP connectors will be connected to each Intel ® X520 NIC ports.
   d. 4 x QSFP to SFP breakout fiber cables.

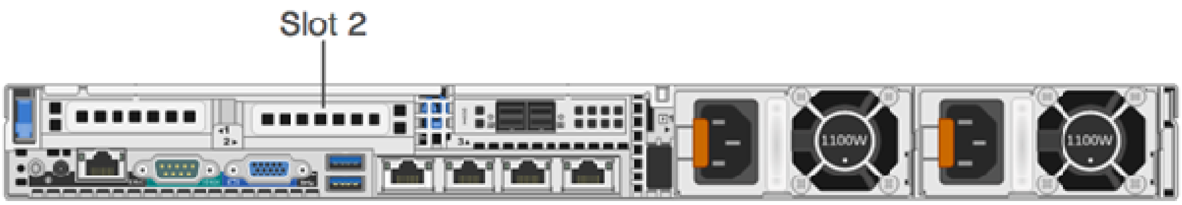
Hardware Deployment and Wiring

Please follow the steps below to deploy hardware and wiring reference:

1. Check BIOS setting of each compute node, and make sure that “SR-IOV Global Enable” is “Enables”.
2. In BIOS, under “Processor Settings”, “Virtualization Technology” flag should be enabled as well.
3. Insert Intel ® X520 in P1 slot of compute node 1, as shown below



4. Insert Intel ® X520 in P2 slot of compute node 1, as shown below



5. Repeat the steps (3) and (4) for compute 2 and 3 as well.
6. Now wire the newly added X520s in compute nodes to S6000 leaf switch, following is the reference wiring diagram:

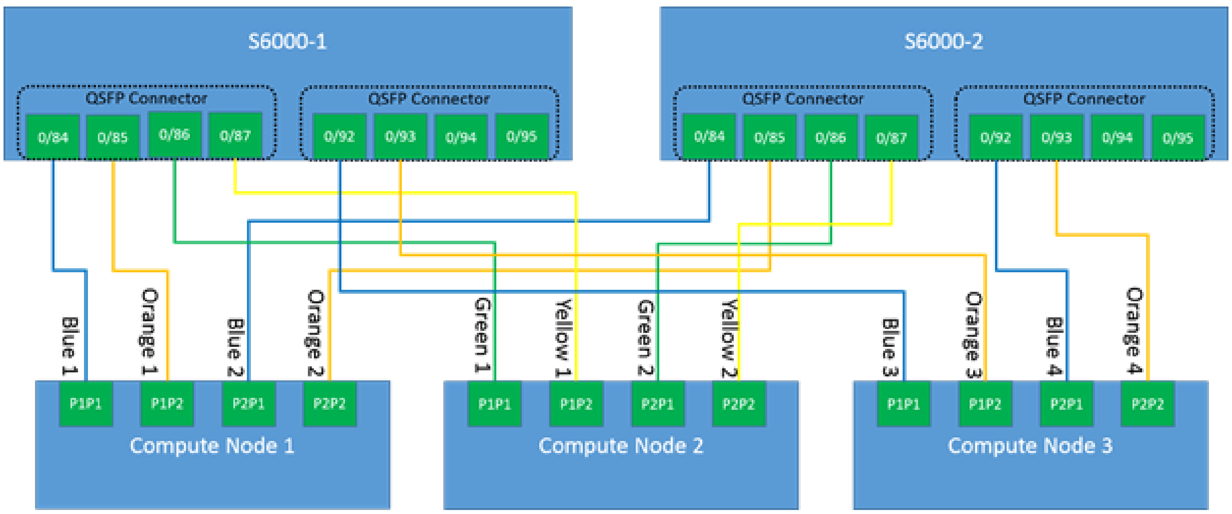


Figure 3: Wiring Topology between compute nodes and S6000 Leaf switches

Additional Wiring Notes

a. 1x breakout cable will be connected to p1 of Compute Node 1
b. 1x breakout cable will be connected to p2 of Compute Node 1
c. 1x breakout cable will be connected to p1 of Compute Node 2 and 3
d. 1x breakout cable will be connected to p2 of Compute Node 2 and 3

7. Double check the QSFP port coloring for each vendor and document that the solution is only validated against the particular one.
8. Connect S4820T switch uplink to external datacenter. It is highly recommended to allocate a dedicated 2x10Gbps link between S4820T and external datacenter.

Network and Switch Configurations

The current SR-IOV RA consists of three compute nodes - compute0, compute1, compute2, three controller nodes - controller0, controller1, controller2. Of these three controller nodes one serves as the master controller node and the other two are the redundant controller nodes, which act as backup to the current master controller node. The instances can be hosted on any of the compute nodes and all the comminucation to the instances is through the master controller. If the master controller fails then one of the stand by controllers becomes the master. There are two Dell S6000 switches running the latest version of FTOS. The current version of FTOS the is recommended to run of the Dell S6000 switches is 9.10(0.1P2). Also included in this solution is an S4820 switch, which serves as a Top of the Rack switch used for external connectivity. This VM is called the director node which runs the SR-IOV script. The Director node should be able to reach and have remote login access to all the compute and controller nodes.

Each compute node has three PCIe NIC slots namely p1,p2,p3. We use NIC cards in slots p1,p2 for SR-IOV enablement. Below example explains the additional switch configuration required for SR-IOV on top of the existing cofiguralion for NFV 6.0. In this example we take only one compute node and in that compute node we take only one of the PF pair into consideration. In this configuration example the switch ports, vlan id's, used are for demonstration purposes only, in practice the user is is free to use the ports of his/her choice on the Dell S6000 switch.

Consider the following scenario as a reference for the switch configurations:

1. Consider the PF pair p1p1 and p2p1 from compute0.
2. Connect p1p1 from comput0 to te 0/86 which is the port on first switch, Dell S6000-1 and p2p1 from compute0 to te 0/87 whihc the port on second switch, Dell S6000-2

```
!
interface TenGigabitEthernet 0/86
 no ip address
 switchport
 no shutdown
```

```
!
interface TenGigabitEthernet 0/87
 no ip address
 switchport
 no shutdown
```

3. Include the port te 0/86 on Dell S6000-1 to which p1p1 on compute0 is connected in vlan 24. These ports which connect to the servers act as access ports and hence they must be untagged. Below is the sample output from the switch:

```
!
interface Vlan 24
 untagged TenGigabitEthernet 0/86
 no shutdown
!
```

4. Include the port te 0/87 on Dell S6000-2 to which p2p1 on compute0 is connected in vlan 24. These ports which connect to the servers act as access ports and hence they must be untagged. Below is the sample output from the switch:

```
!
interface Vlan 24
 untagged TenGigabitEthernet 0/87
 no shutdown
!
```

5. Dell S6000-1 sample configurations are provided below:

```
interface TenGigabitEthernet 0/84
 no ip address
 switchport
 no shutdown
!
interface TenGigabitEthernet 0/85
 no ip address
 switchport
 no shutdown
!
```

```
interface TenGigabitEthernet 0/86
 no ip address
 switchport
 no shutdown
!
interface TenGigabitEthernet 0/87
 no ip address
 switchport
 no shutdown
!
interface TenGigabitEthernet 0/92
 no ip address
 switchport
 no shutdown
!
interface TenGigabitEthernet 0/93
 no ip address
 switchport
 no shutdown
!
interface TenGigabitEthernet 0/94
 no ip address
 switchport
 no shutdown
!
interface TenGigabitEthernet 0/95
 no ip address
 switchport
 no shutdown
!
interface Vlan 24
 no ip address
 tagged Port-channel 2
 untagged TenGigaBitEthernet 0/86
 no shutdown
!
```

Note: The configurations changes shown above are to be made in addition to Dell EMC RedHat NFV solution switch configurations.

6. Dell S6000-2 sample configurations are provided below:

```
interface TenGigabitEthernet 0/84
 no ip address
 switchport
 no shutdown
!
interface TenGigabitEthernet 0/85
 no ip address
 switchport
 no shutdown
!
interface TenGigabitEthernet 0/86
 no ip address
 switchport
 no shutdown
!
interface TenGigabitEthernet 0/87
 no ip address
 switchport
 no shutdown
!
```

```
interface TenGigabitEthernet 0/92
 no ip address
 switchport
  no shutdown
!
interface TenGigabitEthernet 0/93
 no ip address
 switchport
 no shutdown
!
interface TenGigabitEthernet 0/94
 no ip address
 switchport
 no shutdown
!
interface TenGigabitEthernet 0/95
 no ip address
 switchport
 no shutdown
!
interface Vlan 24
 tagged Port-channel 2
 untagged TenGigabitEthernet 0/87
 no shutdown
!
```

Note: The configurations changes shown above are to be made in addition to Dell RedHat NFV solution switch configurations.

7. Make sure the ports which connect to the TOR are also include in this vlan
8. In this example configuration as you can find that Port-Channel 2 is connected to the upstream TOR ports and hence included in vlan 24 with tagging enabled.
9. Make sure to enable L2 functionality with the switchport command.
10. Ideally all the VLAN IDs in Dell S-6000 switch connecting to PF's that are SR-IOV enabled on compute node, should be present on the Dell S4820 TOR switch and be enabled as management vlan. This is done so that the bond device inside the instance can reach the external network.

Software Requirements

1. Make sure ixgbe driver v4.0.1 (or higher) is loaded in compute host. If you are using RHEL 7.2, it will be loaded out of the box. If you are using some other version, make sure ixgbe v4.0.1 (or higher) is loaded.
2. The script will insert “intel_iommu=on” in the grub config file.
3. Make sure you have root privileges, and are able to successfully do "su osp_admin" on the Director node ssh command line.
4. Make sure you have access to overcloud and undercloud RC files (usually residing in /home/osp_admin or the folder from which Director-based OSP solution deployment was ran.

How to get the software

Note: The SR-IOV enablement is carried out using the enable_sriov script written in Python. The script will be made available on the Director Node. TODO: Instructions to do so will be inserted here just prior to RTS.

SR-IOV script cnode pass

Pre-Requisite

Followings are the items that needs to be completed before we start executing Pass-1 of the scripts:

1. Dell EMC Redhat NFV solution 6.0 is deployed successfully and all the OpenStack services are up and running.
2. Compute Nodes are equipped with additional hardware as mentioned in Hardware reference section.
3. Wiring of additional NIC adaptors to physical Dell S6000 switches are according to the wiring reference mentioned above.
4. Log in as “osp_admin” on director node to execute the scripts.
5. Scripts must be cloned from the upstream repo of Dell EMC NFV.

   For more detail, refer to “How to get Software” section.

Script Parameters:

Followings are the script parameters that will be input in the sriov_settings_pass-cnode_R141-compute-1.ini file.

Parameter	**Supported values**	**Description/Example**	**Allowed to change**
[UNIVERSAL]	-	This keyword indicates the start of the UNIVERSAL parameters section. This keyword should be present in the start of the settings file.	No
settingsVersion	1	This parameter reflects the version of the settings file that is being used. Only supported value is 1. Example : `settingsVersion=1`	No
scriptName	enable_sriov	This parameter represents the script name that is being run. Only supported value is enable_sriov. Example: `scriptName=enable_sriov`	No
scriptVersion	0.01	This parameter is the version of the script that is being executed. Only supported value is ‘0.01’ Example : `scriptVersion=0.01`	No

Parameter	Supported values	Description/Example	Allowed to change
RHOSPVersion	9	This parameter is the RedHat OpenStack version on which scripts are being executed. Currently, only supported version is 9 Example : `RHOSPVersion=9`	No
JSVersion	6.0	This specifies the Jet Stream version on which script are being executed. Only supported JetStream version is 6.0. Example : `JSVersion=6.0`	No
scriptPass	"cnode" "instance_pass"	This parameter is the argument to the script to run on compute node or VM instance. First pass to the script should be "cnode" pass, this will create virtual functions and enable compute nodes for SR-IOV. Example : `scriptPass=cnode`	No

scriptPassMode	"ephemeral"
	"persistent"
		This parameter describes weather the SR-IOV and virtual functions will be ephemeral or persistent across multiple reboots of compute nodes. The default value is "ephemeral".
		Example for ephemeral:
		`scriptPassMode=ephemeral`
		Example for ephemeral:
		`scriptPassMode=persistent`
			Yes

num_vfs_per_pf	4	This parameter reflects the number of virtual functions created per physical functions. This release supports 4 VFs per PF. Example : `num_vfs_per_pf=4`	No

PFPair1pf1	interface name	This parameter is to define the logical pairs of physical functions that script creates in order to provide NIC level redundancy. Two PF pairs are created by script. Each pair consists of 2 PF for redundancy.
		Example PF Pair 1:
		`PFPair1pf1=p1p1`
		`PFPair1pf2=p2p1`
		Similarly, example for PF Pair 2:
		`PFPair2pf1=p1p2`
		`PFPair2pf2=p2p2`
			Not in v6.0

Parameter	Supported values	Description/Example	Allowed to change
bw_available_per_pf	10	This parameter describes the bandwidth (in Gbps) available for each physical function. The scripts has been only tested for Intel®X520 dual port adaptors. Therefore, the only supported value is 10. Example : `bw_available_per_pf=10`	Not in v6.0
bw_strategy	equal	This parameter reflects the distribution among virtual functions. Example : `bw_strategy=equal`	Not in v6.0
[CNODE]	-	This keyword indicates the start of the CNODE parameters section. This keyword should be present at the end of UNIVERSAL parameters section.	No
cnodeName	Compute Node Name	This parameter contains the name of the compute node on which the SR-IOV enablement is being triggered. Example : `cnodeName=R141-compute-1`	Yes

Example settings file: Below is a sample of settings file that has been used in one of the Dell EMC RedHat NFV 6.0 environment to setup SR-IOV:

```
[UNIVERSAL]
settingsVersion=1

# These fields must match the values in the script
# that uses this settings file.
# Value(s) below are the only ones supported in NFV Wave 6.0
scriptName=enable_sriov
scriptVersion=0.01
RHOSPVersion=9
JSVersion=6.0

# Valid values are: cnode | instance
# Must be specified by the user
scriptPass=cnode
# Valid values are: ephemeral (default mode) | persistent
# In NFV Wave 6.0, the default mode is ephemeral
scriptPassMode=ephemeral
# scriptPassMode=persistent

# min = 2, max is 64
# these limits are NIC dependent
# Value(s) below are the only ones supported in NFV Wave 6.0
num_vfs_per_pf=4

# PF specs are of the form: (PCI slot) (NIC port)
# e.g. p2p1 = PCI slot 2, NIC port 1
# At least two pairs of PFs are expected: Pair1, and Pair2
```

```
# Value(s) below are the only ones supported in NFV Wave 6.0
PFPair1pf1=p1p1
PFPair1pf2=p2p1
PFPair2pf1=p1p2
PFPair2pf2=p2p2

# All PFs are expected to offer the same b/w.
# Values can be 10 | 40 | 100
# Value(s) below are the only ones supported in NFV Wave 6.0
bw_available_per_pf=10
# Strategy to spread the total available b/w across the pool of VFs
# Valid values are: equal | assigned
# Value(s) below are the only ones supported in NFV Wave 6.0
bw_strategy=equal

[CNODE]
cnodeName=R141-compute-1

# ---------------
```

Steps to execute cnode pass

Followings are the steps that will be followed in the order to execute cnode pass scripts:

1. cnode pass the script requires following arguments described below:
   a. **--settings_file:** This argument requires an input of settings file name. The file contains all the parameters described in the parameter section in above section.

      This file comes with script software package cloned from github repository. The file contains default values for standard Dell EMC RedHat NFV solution 6.0.
   b. **--ucrc:** Under Cloud RC files, it contains the authentication URL and credentials for Under Cloud Director node.
   c. **--ocrc:** Over Cloud RC files, it contains the authentication URL and credentials for Over Cloud OpenStack environment.
   d. **--script_pass:** This arguments describes the type of pass mode to execute the scripts. Following 2 modes are applicable for input:
      a. cnode
      b. instance

      In this section “cnode” mode will be used to execute the scripts
   e. **--logfile:** This is an optional argument to specify the log file on which logs will be stored. In case the file does not exist, script will create the file and if the file exist, script will overwrite the existing log file. If this argument is not passed, log will be stored in the default location.
2. SSH into director node by executing the following commands:

```
$ ssh osp_admin@ <director_node_IP>
$ cd /home/osp_admin/
```

3. Run script through following command:

```
python enable_sriov.py --settings_file <settings file name> -–ucrc
 <undersloud RC file name> --ocrc <overcloud RC file name> --script_pass
 cnode --logfile <file name>
```

Example:

```
python enable_sriov.py --settings_file sriov_settings_pass-cnode.ini --
ucrc stackrc --ocrc r141rc --script_pass cnode --logfile cnode-pass-1.log
```

Validate cnode Pass

Following is snippet that shows the successful completion of the SR-IOV cnode pass:

```
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,058:
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,058: pfPairs[0]:
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,058:     pf1 p1p1, domain 0000, bus 83, slot 00, function 0, VFCount 4
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,058:             vf[0] slot 10, function 0, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,058:             vf[1] slot 10, function 2, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,058:             vf[2] slot 10, function 4, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[3] slot 10, function 6, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:     pf2 p2p1, domain 0000, bus 81, slot 00 function 0, VFCount 4
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[0] slot 10, function 0, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[1] slot 10, function 2, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[2] slot 10, function 4, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[3] slot 10, function 6, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059: pfPairs[1]:
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:     pf1 p1p2, domain 0000, bus 83, slot 00, function 1, VFCount 4
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[0] slot 10, function 1, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[1] slot 10, function 3, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[2] slot 10, function 5, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[3] slot 10, function 7, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:     pf2 p2p2, domain 0000, bus 81, slot 00 function 1, VFCount 4
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[0] slot 10, function 1, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[1] slot 10, function 3, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[2] slot 10, function 5, bw 2500, link UP
[DellNFV enable_sriov v0.01] INFO: 2016-09-21 22:52:44,059:             vf[3] slot 10, function 7, bw 2500, link UP
```

Figure 4: successful completion of cnode pass

In case of successful completion of the cnode pass, script will return the output similar to Figure 4.

SR-IOV script instance pass

Pre requisite

Followings are the items that needs to be completed before we start executing instance pass of the scripts:

1. SR-IOV script cnode pass must be successfully completed.
2. Create an instance in OpenStack Horizon and below attributes of the instance should be noted:
   a. Name of the instance as appeared in OpenStack dashboard.
   b. Name of the keypair file associated with that instance.

      Note: Please refer to Appendix C of the Dell EMC RedHat OSP deployment guide for more details on Keypair association with the instance.

   c. Login credentials of the instance.
3. You must be loged in as “osp_admin” or "stack" user in director node to execute the scripts.
4. Scripts must be obtained from the upstream repo of Dell EMC NFV. For more detail, refer to “How to get Software” section.

Script Parameters:

Followings are the script parameters that will be input in the `sriov_settings_pass-instance_rhel-vm1-compute1.ini` file.

Parameter	Supported values	Description/Example	Allowed to change
[UNIVERSAL]	-	This keyword indicates the start of the UNIVERSAL parameters section. This keyword should be present in the start of the settings file.	No
settingsVersion	1	This parameter reflects the version of the settings file that is being used. Only supported value is 1. Example: `settingsVersion=1`	No
scriptName	enable_sriov	This parameter represents the script name that is being run. Only supported value is enable_sriov. Example: `scriptName=enable_sriov`	No
scriptVersion	0.01	This parameter is the version of the script that is being executed. Only supported value is ‘0.01’ Example: `scriptVersion=0.01`	No

Parameter	Supported values	Description/Example	Allowed to change
RHOSPVersion	9	This parameter is the RedHat OpenStack version on which scripts are being executed. Currently, only supported version is 9 Example: `RHOSPVersion=9`	No
JSVersion	6.0	This specifies the Jet Stream version on which script are being executed. Only supported JetStream version is 6.0. Example: `JSVersion=6.0`	No
scriptPass	"cnode" | "instance_pass"	This parameter is used to choose between cnode and instance pass. Always run cnode pass first. Example: `scriptPass=instance_pass`	No
scriptPassMode	"ephemeral"	This parameter describes weather the SR-IOV and virtual functions will be ephemeral or persistent across multiple reboots of compute nodes. Only supported value is "ephemeral", currently, persistent mode in instance_pass is not supported. Example for ephemeral: `scriptPassMode=ephemeral`	No
num_vfs_per_pf	4	This parameter reflects the number of virtual functions created per physical functions. This release supports 4 VFs per PF Example: `num_vfs_per_pf=4`	Not in v6.0

PFPair1pf1	interface name	This parameter is to define the logical pairs of physical functions that script creates in order to provide NIC level redundancy. Two PF pairs are created by script. Each pair consists of 2 PF for redundancy.
		Example PF Pair 1 :
		`PFPair1pf1=p1p1`
		`PFPair1pf2=p2p1`
		Similarly, example for PF Pair 2:
		`PFPair2pf1=p1p2`
		`PFPair2pf2=p2p2`
			Not in v6.0

Parameter	Supported values	Description/Example	Allowed to change
bw_available_per_pf	10	This parameter describes the bandwidth (in Gbps) available for each physical function. The scripts has been only tested for Intel®X520 dual port adaptors. Therefore, the only supported value is 10. Example: `bw_available_per_pf=10`	No
bw_strategy	equal	This parameter reflects the distribution among virtual functions. Example: `bw_strategy=equal`	No
[INSTANCE]	-	This keyword indicates the start of the INSTANCE parameters section. This keyword should be present at the end of UNIVERSAL parameters section.	No
instanceName	instance Name	This parameter contains the name of the compute node on which the SR-IOV enablement is being triggered. Example: `cnodeName=R141-compute-1`	Yes

Below is the table that reflects the parameters particular to an instance that is being used for SR-IOV:

Parameter	Supported values	Description/Example	Allowed to change
[INSTANCE NAME]	-	This keyword indicates the start of the section which contains the parameters particular to the instance that is being used for SR-IOV . This keyword should be a valid instance name as configured in "instanceName" paramter in [INSTANCE] section.	Yes
behaviour	bonding	This parameter describer the behavior of virtual functions after being assigned to a VM. Current supported behavior is “bonding”. This behavior will create bond of 2 virtual functions inside the virtual machines. Example: `behavior=bonding`	Not in v6.0

Parameter	**Supported values**	**Description/Example**	**Allowed to change**
bonding_on_platform	rhel7.2	This parameter is the name of the guest operating system running inside the virtual machine. The only supported platform in this release is “rhel7.2” Example: `bonding_on_platform=rhel7.2`	Not in v6.0
numBonds	1	This parameter is the number of bonds to be created inside the VM. The only value supported is 1. Example: `numBonds=1`	Not in v6.0
bonding_failover_mode	“a-s ”| “a-a”	This parameter specifies the bonding failover mode, in case, the physical or virtual function goes down. The only supported failover mode is "a-s" (Active-Standby), "a-a" (Active-Active) is not supported in this release. Example: This example will configure the 2 virtual functions into Active-Standby mode. `bonding_failover_mode=a-s`	Yes
bondIP	< Valid IPv4 address>	This parameter is the IP address to be assigned to the bond interface created inside the VM. Instance pass script will automatically assign an IP address to the bond: Example: `bondIP=172.25.141.182`	Yes
bondIPMaskLen	< valid subnet mask>	This parameter reflects the subnet mask of the IP address to be assigned to bond interface. The value should be an integer. Example: `bondIPMaskLen=24`	Yes
bondGatewayIP	< Valid IPv4 address>	This parameter is the gateway IP of the bond network that will be assigned to the bond interface. Example: `bondGatewayIP=172.25.141.254`	Yes

Parameter	Supported values	Description/Example	Allowed to change
userid=	< user id of the instance>	This parameter is the user id of the instance that will be used to SSH into the VM and execute the commands inside the VMs. The following value as mentioned in the example is the default userid of the RedHat cloud image. Example: `userid=cloud-user`	Yes
keyfile	< .pem file name to be used for keypair>	This key file will be used to SSH into the instance. This keyfile name can be obtained from the OpenStack dashboard at the time of keypair creation. Example: `keyfile=key_name.pem`	Yes
password	< password string to SSH into the VM>	This parameter is the password that will be used to SSH into the VM. If keyfile are being used, keep the value of this field as followings: `password=unused_if_keyfile_is_valid`	Yes

Example settings file: Below is a sample of settings file that has been used in one of the Dell EMC RedHat NFV 6.0 environment to setup SR-IOV:

```
[UNIVERSAL]
settingsVersion=1

scriptName=enable_sriov
scriptVersion=0.01
RHOSPVersion=9
JSVersion=6.0

scriptPass=instance
scriptPassMode=ephemeral
# scriptPassMode=persistent

num_vfs_per_pf=4

PFPair1pf1=p1p1
PFPair1pf2=p2p1
PFPair2pf1=p1p2
PFPair2pf2=p2p2

bw_available_per_pf=10
bw_strategy=equal

[INSTANCE]
instanceName=rhel-vm1-compute1

[rhel-vm1-compute1]
behavior=bonding
bonding_on_platform=rhel7.2
numBonds=1
```

```
# bonding_failover_mode=a-s
bonding_failover_mode=a-a
bondIP=172.25.141.182
bondIPMaskLen=24
bondGatewayIP=172.25.141.254
userid=cloud-user
keyfile=key_name.pem
password=unused_if_keyfile_is_valid

# ---------------
```

Steps to execute instance pass

Followings are the steps that will be followed in the order to execute instance pass scripts:

1. For instance pass the script requires following arguments described below:
    a. **--settings_file:** This argument requires an input of settings file name. The file contains all the parameters described in the parameter section in above section.

        This file comes with script software package obtained from Dell EMC NFV repository. The file contains default values for standard Dell EMC RedHat NFV solution 6.0.
    b. **--ucrc:** Under Cloud RC files, it contains the authentication URL and credentials for Under Cloud Director node.
    c. **--ocrc:** Over Cloud RC files, it contains the authentication URL and credentials for Over Cloud OpenStack environment.
    d. **--script_pass:** This arguments describes the type of pass mode to execute the scripts. Following 2 modes are applicable for input:
        a. cnode
        b. instance

        In this section "instance" mode will be used to execute the scripts
    e. **--logfile:** This is an optional argument to specify the log file on which logs will be stored. In case the file does not exist, script will create the file and if the file exist, script will overwrite the existing log file. If this argument is not passed, log will be stored in the default location.
2. SSH into director node using the following commands

```
$ ssh osp_admin@ < director_node_IP>
$ cd /home/osp_admin/
```

3. Run script through following command:

```
python enable_sriov.py --settings_file <settings file name> -–ucrc
 <undersloud RC file name> --ocrc <overcloud RC file name> --script_pass
 instance --logfile <file name>
```

Example:

```
python enable_sriov.py --settings_file sriov_settings_pass-instance_rhel-
vm1-compute1.ini --ucrc stackrc --ocrc r141rc --script_pass instance --
logfile instance-pass-1.log
```

Validate instance pass

1. Successful execution of script should have the following output:

```
INFO: 2016-10-03 16:15:09,289: ****************** Instance rhel-vm1-compute1: instance pass FINAL SUMMARY **************
INFO: 2016-10-03 16:15:09,289:
INFO: 2016-10-03 16:15:09,290: Run Summary: Common Information:
INFO: 2016-10-03 16:15:09,290:
INFO: 2016-10-03 16:15:09,290: Command run: ['enable_sriov.py', '--ucrc', '/home/osp_admin/stackrc', '--ocrc', '/home/osp_a
INFO: 2016-10-03 16:15:09,290: Args: {'script_pass': 'instance', 'director_install_user': 'osp_admin', 'ocrc': '/home/osp_a
INFO: 2016-10-03 16:15:09,290: Setting read from: ./sriov_settings_pass-instance_rhel-vm1-compute1.ini
INFO: 2016-10-03 16:15:09,290: Settings: {'key_name_fname': 'key_name.pem', 'JSVersion': '6.0', 'RHOSPVersion': '9', 'conf'
INFO: 2016-10-03 16:15:09,290:
INFO: 2016-10-03 16:15:09,290:
INFO: 2016-10-03 16:15:09,290: ------------------
INFO: 2016-10-03 16:15:09,291: Instance Pass Summary:
INFO: 2016-10-03 16:15:09,291: ------------------
INFO: 2016-10-03 16:15:09,291: Number of ping requests sent = 10
INFO: 2016-10-03 16:15:09,291:
INFO: 2016-10-03 16:15:09,291: Number of ping responses received = 10
INFO: 2016-10-03 16:15:09,291:
INFO: 2016-10-03 16:15:09,291: Instance Pass was a success!
INFO: 2016-10-03 16:15:09,291:
INFO: 2016-10-03 16:15:09,291: ------------------
```

Figure 5: Successful completion of the instance pass

2. Make sure the virtual functions bonds are successfully created inside the virtual machines.
3. VM is able to send and receive 2.5GB/VF to physical network
4. Verify SR-IOV HA by pulling the cable of one of PF from the physical switch and make sure VM is still able to send/receive traffic.

Note: In case the OverCloud update fails after reboot, please refer to this known bug *https://bugzilla.redhat.com/show_bug.cgi?id=1383780*

Best Practices for Dell EMC NFV SR-IOV solution

1. Make sure the script contains latest time stamp.
2. It is recommended to verify that SSH is working between director and control/compute. You can go to director node and run the following command from there.

   ```
   ssh heat-admin@<compute-node-ip>
   ```

   This will validate the SSH connectivity between director node and compute node.
3. Save the log files after script has returned successfully or unsuccessfully.
4. It is a pre-requisite to run the cnode pass atleast once before running the instance pass.
5. Double check the key file mapped to your instance. The name of key file should match the key file name that is given in the settings file.
6. It is highly recommended that you login to instance using the user id and the corresponding key file atleast once after launching the instance and before running the instance pass.

   Note: Please refer to the Appendix C of Dell EMC RedHat NFV solution v6.0 OSP deployment guide, for more details.
7. Make sure `ifup-post` is present in `/etc/sysconfig/network-scripts/` directory.

Possible Errors

Following are the list of commonly occurring errors with probable solutions:

1. Cannot ssh into compute/controller node IP

   Solution: Check network connectivity from director to compute node or check if ssh keys are present for the compute/controller nodes. Also check if routes to the compute/controller node are present in the routing table of. The only way to reach the compute/controller is through the director. Final check to see if the network cable is loosend
2. Cnode does not have VFs. Have you run the cnode pass?

   Solution: The order of runs is cnode pass first and then instance pass. Instance pass first checks if the VF's are created and available for use, if it does not find the VF's it will exit out with this error.
3. SR-IOV cannot be enabled in persistent mode on this compute node, as ifup-post file is missing in /etc/sysconfig/network-scripts

   Solution: Check if “ifup-post” file is present in “/etc/sysconfig/network-scripts”. This file is called during the boot time and all parameters in that file are set during boot up time. This init file contains the location of the file where the command for VF's and rate are present.
4. SR-IOV already is enabled in persistent mode on Cnode. Exiting.

   Solution: cnode pass has been already completed successfully. Once the cnode pass is run in persistent mode, the VF's and the rate is sustained across reboots. So you can go ahead with executing instance pass. You can only reset it back manually.
5. NOT Valid: Compute node was not found in the Undercloud.

   Solution: The compute node might be down or undergoing a reboot at the time of executing the script. Also, check if the correct compute node name has been input in the settings file
6. FATAL ERROR: CNode did not come up after changing the grub file

   Solution: GRUB file has been corrupted in the compute node during persistence enablement. A backup grub file, gurb.bak, is created before modifying the grub file. Use this backup file to modify the grub file to its original state.The grub backup file is stored in /etc/sysconfig/
7. Instance < instance name> is SHUTDOWN . Cannot complete enable_sriov.py v0.01 pass

   Solution: The instance created from Open stack is not Active. Make sure the instance is running.
8. No available bond device found in instance

   Solution: Make sure the bonding module is present in the instance you created. Number of bond devices per instance is limited to one in this release.
9. Could not find network id for instance

   Solution: Associate a tenant network to the instance in openstack. This id is required for the script to determine if the instance is connected to the tenant network. The script enters the the tenanat name space using this id.
10. It seems the keyfile setting is incorrect for instance. Are you sure you are using the correct keyfile?

    Solution: Check if the “keyname” parameter in settings file matches with the key file in OpenStack.
11. Instance pass keyfile not found

    Solution: Associate a key file to an instance. This can be done in Access and Security in Openstack Horizon. The name you provide to this key file should also be given in the settings file