Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform

SR-IOV User Guide

Version 10.0.1

Dell EMC Service Provider Solutions

Contents

List of Figures

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.

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.

Chapter 1

Introduction

Topics:

This section explains the concept of SR-IOV and gives a thorough explanation of its integration into Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform.

Note: SR-IOV has only been validated on Dell EMC servers with Intel Ethernet Converged Network Adapter x520 NICs.

Note: Update/Upgrade is not supported for SR-IOV.

SR-IOV

Single root I/O virtualization (SR-IOV) is an extension to the PCI Express (PCIe) specification. SR-IOV enables a single PCIe device to appear as multiple, separate devices. Traditionally in a virtualized environment, a packet has to go through an extra layer of the hypervisor, that results in multiple CPU interrupts per packet. These extra interrupts cause a bottleneck in high traffic environments. SR-IOV enabled devices, have the ability to dedicate isolated access to its resources among various PCIe hardware functions. These functions are later assigned to the virtual machines which allow direct memory access (DMA) to the network data. This enables efficient sharing of PCIe devices, optimization of performance and reduction in hardware costs.

Architecture

Physical Function (PF):

Physical Function is a full-featured PCIe function of a network adapter that is SR-IOV capable. Physical Functions are discovered, managed, and configured as normal PCIe devices. PCIe devices have a set of registers known as configuration space. Physical Function behaves like an L2 switch and performs traffic forwarding between physical port and Virtual Functions.

Virtual Function (VF):

Virtual Functions (VFs) are simple PCIe functions that lack the configuration resources and only have the ability to move data in and out. Each Virtual Function is associated with a PCIe Physical Function. Each VF represents a virtualized instance of the network adaptor and has a separate PCI Configuration space. These VFs are assigned to virtual machines later.

I/O MMU:

Input–Output Memory Management Unit (IOMMU) connects a Direct-Memory-Access–capable (DMA-capable) I/O bus to the main memory. The IOMMU maps device addresses or I/O addresses to physical addresses. IOMMU helps in accessing physical devices directly from virtual machines.

Hypervisor:

A hypervisor is a software that allows running multiple virtual machines to share a single underlying hardware platform. In case of SR-IOV, either the hypervisor or the guest OS must be aware that they are not using full PCIe devices. Hypervisor maps VF's configuration space to the configuration space presented to the guest using IOMMU.

Virtual Machine:

A VF can only be assigned to one virtual machine at a time however, a virtual machine may have multiple virtual functions associated with it. VF appears as a single network interface card inside of a virtual machine. The virtual machine must be aware of using VF in case the hypervisor is not aware of using an SR-IOV enabled adapter.

Basic SR-IOV architecture is depicted in Figure 1.

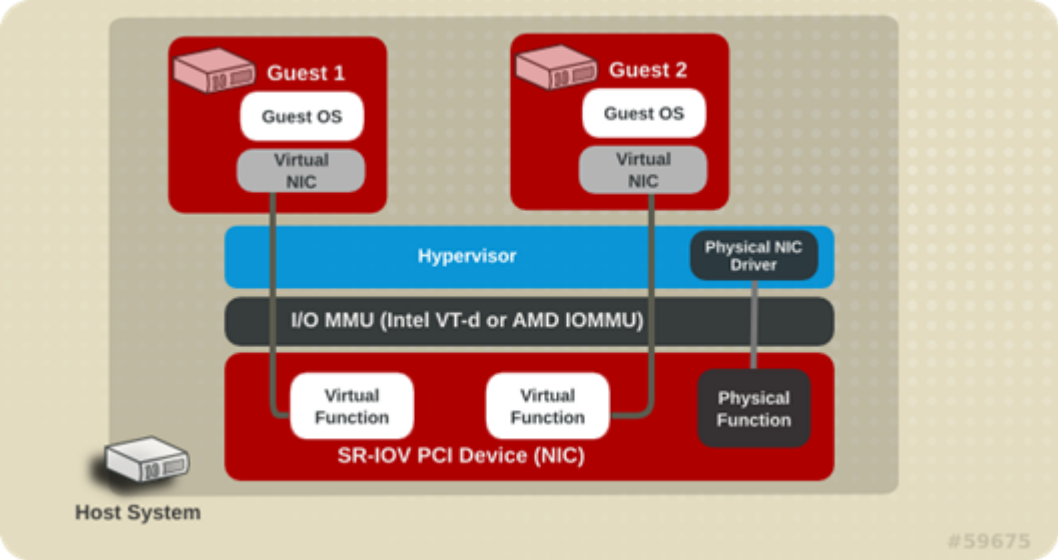

Figure 1: Basic SR-IOV Architecture

Features

Wire-speed performance:

SR-IOV takes a physical port (Physical Function or PF) and logically slices it into Virtual Functions (VFs). Each VF can be programmed to provide a certain amount of bandwidth to the Virtual Network Function (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.

Fail-over support:

Sometimes referred to as high-availability (HA), the SR-IOV implementation in this solution supports 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.

Following failover Scenarios validated:

On same compute nodes	On different compute nodes
NIC is down	NIC is down
Switch is down	Switch is down
Cable is plugged out	Cable is plugged out

Co-existence with OpenStack platform native networks:

The SR-IOV implementation does not attempt to merge with OpenStack networks. The built-in tenant and external networks supported by the OpenStack 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, such as a firewall, a session-border controller, or an application-delivery controller.

Separate network resources:

The SR-IOV implementation 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. This requires the use of a shared resource (the virtual network router) in the Master Controller, and disrupts the end-to-end wire-speed performance to and from a VNF.

Chapter 2

Pre-Deployment planning

Topics:

To proceed with the deployment, setup the environment as per the requirements specified in this section.

System requirements

The basic hardware requirements for Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform can be found in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform - Hardware Deployment Guide. Given below are the additional hardware requirements that must be supported to enable SR-IOV.

SR-IOV system requirements:

This table lists the number of hardware devices like Adaptors and Connectors required by SR-IOV.

Device Name	Total
Intel® X520 Adapter	6
SFP Connector	12
QSFP Connector	4

SR-IOV NIC details:

This table lists the number of NICs required to enable SR-IOV.

NIC	NICs/compute node	Ports/NIC	Bandwidth/ Port	VFs/Port	Bandwidth/VF
Intel® X520 Connector	2	2	10G	4	2.5G

Networking requirements

SR-IOV enabled instances, communicate over the external network. External network VLAN must be configured for the SR-IOV enabled NICs in the switches according to *Switch configuration.*

Additional VLAN requirements:

External Network VLAN	Dedicated/Shared

Network and switch configurations

- Insert both Intel® X520 in the P2 and P3 slot of computer nodes as shown below:

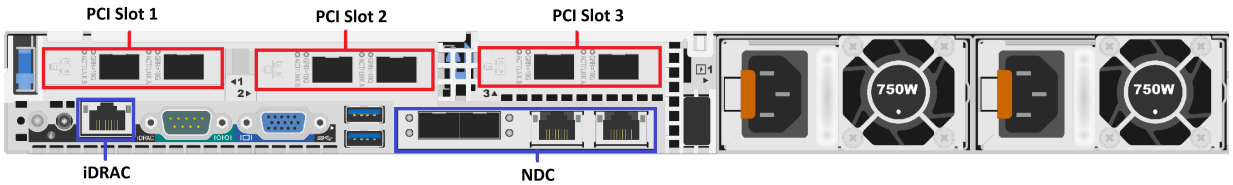

Figure 2: Intel X520 in slot P2 and P3

- Wire the newly added X520s in compute nodes to Dell EMC network S6010 switch leaf switch as per the reference diagram below:

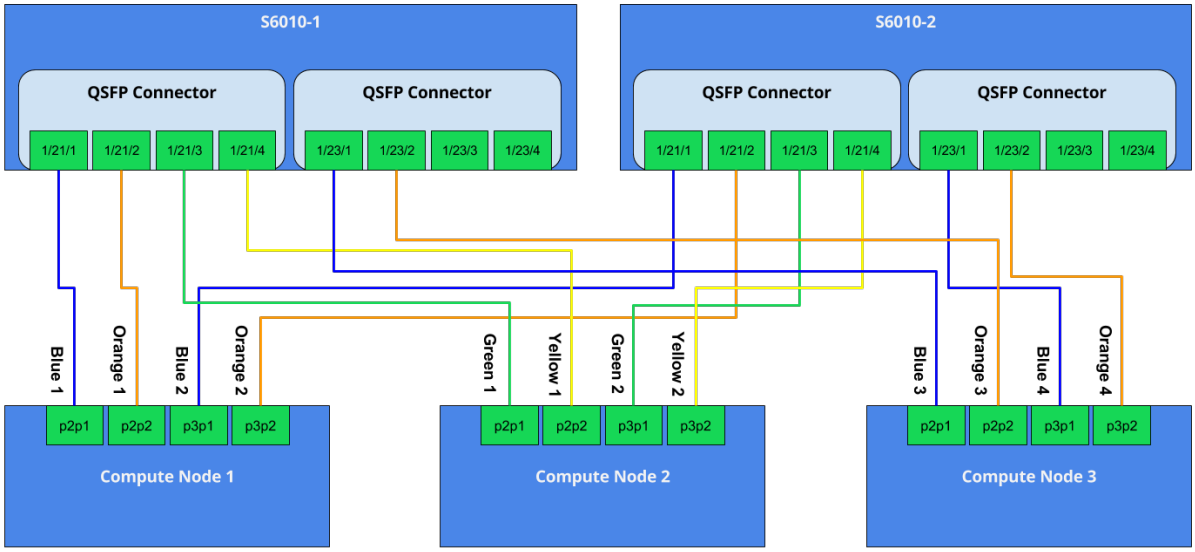

Figure 3: Reference Diagram for network connectivity on S6010 switch

- There are two Dell EMC Network S6010 switches running with the latest version of FTOS. Also included in this solution is the Dell EMC Network S4048 switch, which serves as a Top of Rack, or ToR switch used for external connectivity. The OpenStack Director node runs the SR-IOV script. Each compute node has three PCIe NIC slots p1, p2 and p3. For SR-IOV enablement use NIC cards in slots p2 and p3.
- It is recommended to connect Dell EMC Network S4048 switch uplink to an external network with dedicated 2x10Gbps link between them.
- Connect the SR-IOV NICs in compute0, compute1 and compute2 to the Dell S6010-1 and S6010-2 switches according to the above reference diagram
- Dell S6010-1 sample configurations are provided below:

```
interface TenGigabitEthernet 1/21/1
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/21/2
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/21/3
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/21/4
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/23/1
no ip address
switchport
no shutdown
!
```

```
interface TenGigabitEthernet 1/23/2
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/23/3
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/23/4
no ip address
switchport
no shutdown
!
interface Vlan 24
no ip address
tagged Port-channel 2
untagged TenGigaBitEthernet 1/21/1-1/21/4,1/23/1-1/23/4
no shutdown
```

Note: Note: The configuration changes above are to be made in addition to Dell EMC Red Hat Cloud Infrastructure Platform for NFV switch configurations.

- Dell S6010-2 sample configurations are provided below:

```
interface TenGigabitEthernet 1/21/1
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/21/2
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/21/3
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/21/4
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/23/1
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/23/2
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/23/3
no ip address
switchport
no shutdown
!
interface TenGigabitEthernet 1/23/4
no ip address
switchport
```

```
no shutdown
!
interface Vlan 24
tagged Port-channel 2
untagged TenGigabitEthernet 1/21/1-1/21/4,1/23/1-1/23/4
no shutdown
!
```

Note: The configuration changes shown above are to be made in addition to Dell EMC Red Hat Cloud Infrastructure Platform for NFV switch configurations.

- Verify that the ports which connect to the ToR are also included in this VLAN. In the example configuration, Port-Channel 2 is connected to the upstream ToR ports and are included in VLAN 24 with tagging enabled.
- Verify that L2 functionality with the switch port command.

Software Requirements

Software requirements for SR-IOV enablement:

Intel networking driver	Ixgbe v4.0.1 or higher
Intel IOMMU	True
Root privileges for osp_admin user	True
Undercloud and Overcloud RC files access	True

Chapter 3

Deployment

Topics:

This chapter provides an overview of the SR-IOV feature in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Version 10.0.1.

SR-IOV Enablement Prerequisites

- Ensure that a valid subscription and outbound network access to Red Hat's Content Delivery Network is available.
- Verify that the Dell EMC Red Hat NFV 10.0 solution is deployed successfully and that all of the OpenStack services are up and running.
- Compute nodes are equipped with additional hardware as mentioned in *Pre-Deployment planning*.
- Verify that the wiring of additional NIC adaptors to physical Dell S6000 or S6010 switches are according to the hardware deployment and wiring.

What's New

- In this release, the user is now allowed to use any PCI slot for SR-IOV NICs.
- The user was restricted to use at least two PF pairs in the previous release. In the SR-IOV and OvS-DPDK integrated environment, only two PFs will be available for SR-IOV. In this release, SR-IOV enablement script allows the user to enable SR-IOV on any number of PF groups.

  Note: For OVS-DPDK configurations, please refer to the OVS-DPDK appendix in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Software Deployment Guide
- All the PFPair parameters in the Cnode settings file are replaced with "bond_groups". It is a comma-separated list of interface sets.

Caution: BEFORE you use this facility - it is important to note that the scripts described in this document WILL REBOOT the compute node in the solution several times. You are strongly advised to save all work, stop all services, and ask users to log out before proceeding any further.

Warning: Do not reboot any nodes when the optional feature scripts are being used.

SR-IOV Enablement

SR-IOV enablement script is subdivided into two parts, Cnode Pass and Instance Pass scripts. Each script makes use of a different Settings INI file. The Settings file requires the user to input the parameters required for executing the individual scripts.

Cnode Pass

Cnode pass is the first among the two scripts that should be run. Cnode pass enables SR-IOV on a single compute node in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Version 10.0.1. Before starting to execute the Cnode pass script, make sure all the pre-requisites are met as mentioned in the *Prerequisites* section. The script verifies whether all of the prerequisites have been met or not and makes sure the PF pairs are not part of any existing bonds.

It has two modes:

1. **Ephemeral:** The ephemeral mode of Cnode pass is the non-persistent setup of SR-IOV on the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Version 10.0.1. It enables the SR-IOV environment on a Compute node and will be reset after this Compute node is rebooted. All the VFs and the rates set are no longer present after the reboot.
2. **Persistent:** The persistent mode of Cnode pass creates the SR-IOV environment that would be persistent across multiple reboots of a compute node. This is done by creating a *ifup-dell-nfv-lcl* in */sbin* file and adding all the VF creation and VF rate set commands in it. This file is called from */etc/sysconfig/*

network-scripts/ifup-post at boot up time. If this file is missing, the Persistent Cnode pass mode cannot be enabled. In order to not corrupt the *ifup-post* file, the script creates the backup of the original file before proceeding further.

Script Parameters:

The following script parameters are included in the Cnode Settings INI file:

Parameter	Supported Values	Description	Allowed to change
[UNIVERSAL]	-	This keyword indicates the start of the UNIVERSAL parameters section. This keyword should be present at the start of the settings file.	No
settingsVersion	10.0.1	This parameter reflects the version of the settings file that is being used. Example: settingsVersion=10.0.1	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	10.0.1	This parameter is the version of the script that is being executed. Only supported value is 10.0.1. Example: scriptVersion=10.0.1	No
RHOSPVersion	10	This parameter is the RedHat OpenStack version on which scripts are being executed. Currently, version 10 is the only supported version. Example: RHOSPVersion=10	No
JSVersion 10.0.1	10.0.1	This specifies the Jet Stream version on which script is being executed. Only supported Jetstream version is 10.0.1. Example: JSVersion=10.0.1	No

Parameter	Supported Values	Description	Allowed to change
scriptPass	“cnode”	This parameter is the argument to the script to run on compute node or VM instance. The 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”	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 persistent: scriptPassMode=persistent	Yes
num_vfs_per_pf	4	This parameter reflects the number of virtual functions created per physical function. This release supports 4 VFs per PF. Example: num_vfs_per_pf=4	No
bond_groups	{interface1,interface2}, {interface3,interface4}	This parameter is used to define the logical groups of physical functions that the script creates in order to provide NIC level redundancy. It is a comma-separated list of interface sets. For Example: {p1p1,p2p1},{p1p2,p2p2}	Yes
bw_available_per_pf	10	This parameter describes the bandwidth (in Gbps) available for each physical function. The scripts have been tested only on Intel® X520 dual port adaptors. Therefore, the only supported value is 10. Example : bw_available_per_pf=10	No

Parameter	Supported Values	Description	Allowed to change
bw_strategy	equal	This parameter reflects the distribution among virtual functions. Example : bw_strategy=equal	No
[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=R141nfv-compute-1	Yes

Steps to execute Cnode pass

Use the following steps to run the Cnode pass script:

1. Cnode pass script requires the following arguments:

```
usage: enable_sriov.py [-h] --settings_file SETTINGS_FILE --script_pass
                       SCRIPT_PASS
                       [--director_install_user DIRECTOR_INSTALL_USER]
                       [--logfile LOGFILE] --ucrc UCRC --ocrc OCRC

Enable SR-IOV in Dell NFV Platform

Required Parameters:
  --settings_file SETTINGS_FILE
                        ini settings file, e.g ./sriov.ini
  --script_pass SCRIPT_PASS
                        script pass can be: {cnode|instance}
  --ucrc UCRC           Undercloud RC file
  --ocrc OCRC           Overcloud RC file

Optional Parameters:
  -h, --help            show this help message and exit
  --director_install_user DIRECTOR_INSTALL_USER
                        User for undercloud/overcloud installation
  --logfile LOGFILE     name of the logfile
```

2. SSH into director node by executing the following commands:

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

3. Edit the path to settings file as per the sample provided in *Appendix-A*.

4. Run script using the 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> --director_install_user <username>
```

The warning that user will get before executing the script is shown in the figure below:

```
INFO: 2017-10-10 12:29:46,910: Settings file loaded successfully.
INFO: 2017-10-10 12:29:47,126: Sanity checks completed successfully.
INFO: 2017-10-10 12:29:47,126:
INFO: 2017-10-10 12:29:47,127:
INFO: 2017-10-10 12:29:47,127: #     #
INFO: 2017-10-10 12:29:47,127: #  #  #    ##    #####   #    #     #    #    #   ####
INFO: 2017-10-10 12:29:47,127: #  #  #   #  #   #    #  ##   #     #    ##   #  #    #
INFO: 2017-10-10 12:29:47,127: #  #  #  #    #  #    #  # #  #     #    # #  #  #
INFO: 2017-10-10 12:29:47,127: #  #  #  ######  #####   #  # #     #    #  # #  #  ###
INFO: 2017-10-10 12:29:47,127: #  #  #  #    #  #   #   #   ##     #    #   ##  #    #
INFO: 2017-10-10 12:29:47,128:  ## ##   #    #  #    #  #    #     #    #    #   ####
INFO: 2017-10-10 12:29:47,128:
INFO: 2017-10-10 12:29:47,128:
INFO: 2017-10-10 12:29:47,128: The enable_sriov script MAY reboot OpenStack Compute nodes multiple times.
INFO: 2017-10-10 12:29:47,128: Please save your work, stop pertinent services,
INFO: 2017-10-10 12:29:47,128:     and ask users to completely log out before proceeding.
INFO: 2017-10-10 12:29:47,128: Also, please do not reboot nodes manually until the
INFO: 2017-10-10 12:29:47,129:    enable_sriov script has completed execution.
INFO: 2017-10-10 12:29:47,129:
INFO: 2017-10-10 12:29:47,129:
Do you want to continue (y/n):
```

Figure 4: SR-IOV Cnode warning

This is an example of 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 5: Successful completion of Cnode pass

Instance pass

Additional prerequisites:

1. Verify that the SR-IOV script Cnode pass has been successfully completed.
2. Create an instance in the OpenStack Horizon. The following attributes should be noted:
    - Name of the instance as it appears in the OpenStack dashboard.
    - Name of the keypair file associated with that instance. The SSH keypair file through which instance is launched should be copied to /home/osp_admin in director node.
3. Login credentials of the instance.

> **Note:** For NIC Alignment feature, NUMA awareness and CPU pinning must be enabled in the OpenStack environment. For additional information, refer to the NUMA Appendix E in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Software Deployment Guide.

```
$ python enable_numa.py --flavor_name [flavor_name] --cpu_group numanode1
```

Script parameters

The following script parameters are found in the VM Settings file:

Parameter	Supported values	Description	Allowed to change
[UNIVERSAL]	-	This keyword indicates the start of the UNIVERSAL parameters section. This keyword should be present at the start of the settings file.	No
settingsVersion	10.0.1	This parameter reflects the version of the settings file that is being used. Example: settingsVersion=10.0.1	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	10.0.1	This parameter is the version of the script that is being executed. Only supported value is 10.0.1. Example: scriptVersion=10.0.1	No
RHOSPVersion	10	This parameter is the RedHat OpenStack version on which scripts are being executed. Currently, version 10 is the only supported version. Example: RHOSPVersion=10	No
JSVersion	10.0.1	This specifies the Jet Stream version on which script is being executed. Only supported Jetstream version is 10.0.1. Example: JSVersion=10.0.1	No
scriptPass	“instance”	This parameter is the argument to the script to run on compute node or VM instance. The first pass to the script should be “cnode” pass, this will create virtual functions and enable compute nodes for SR-IOV. Example: scriptPass=instance	No

Parameter	Supported values	Description	Allowed to change
scriptPassMode	“ephemeral”	The attached SR-IOV virtual functions and created bond/bonds will be ephemeral or persistent across multiple soft/hard reboots of the instance. The default value is “ephemeral”. Example for ephemeral: scriptPassMode=ephemeral Example for persistent: scriptPassMode=persistent	No
num_vfs_per_pf	4	This parameter reflects the number of virtual functions created per physical function. This release supports 4 VFs per PF. Example: num_vfs_per_pf=4	No
bond_groups	{interface1,interface2}, {interface3,interface4}	This parameter is used to define the logical groups of physical functions that the script creates in order to provide NIC level redundancy. It is a comma-separated list of interface sets. For Example: {p1p1,p2p1},{p1p2,p2p2}	Yes
bw_available_per_pf	10	This parameter describes the bandwidth (in Gbps) available for each physical function. The scripts have been only tested for Intel® X520 dual port adaptors. 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

Parameter	Supported values	Description	Allowed to change
instanceName	<Instance Name>	This parameter contains the name of the compute node on which the SR-IOV enablement is being triggered. Example : instanceName=sriov_test	Yes
[<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" parameter in [INSTANCE] section.	Yes
Behavior	bonding	This parameter describes the behavior of virtual functions after being assigned to a VM. Currently supported behavior is “bonding”. This behavior will create a bond with the number of VFs inside the virtual machines according to the interfaces in the "bond_groups" parameter. Example : behavior=bonding	No
bonding_on_platform	rhel7.2 | rhel7.3 | centos7.2	This parameter is the name of the guest operating system running inside the virtual machine. In this release, 4 platforms are supported. Example: behavior=bonding bonding_on_platform=centos7.2	Yes
numBonds	1	This parameter is the number of bonds to be created inside the VM. The only value supported is 1. Example: numBonds=1	No

Parameter	Supported values	Description	Allowed to change
bonding_failover_mode	“a-s”| “a-a”	This parameter specifies the bonding failover mode in case the physical or virtual function goes down. Both failover modes "a-s" (Active-Standby) and "a-a" (Active-Active) are supported in this release. Example : This example will configure the 2 virtual functions in Active-Standby mode bonding_failover_mode=a-s bonding_failover_mode=a-a	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=100.67.141.182	Yes
bondIPMaskLen	< valid subnet mask >	This parameter reflects the subnet mask of the IP address to be assigned to the 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=100.67.141.254	Yes
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

Parameter	Supported values	Description	Allowed to change
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
nic_alignment	< true or false >	Want to use Nic_alignment feature in SR-IOV make: Example : nic_alignment=true	Yes

Steps to execute the Instance pass

1. Add SSH into the Director node using the following commands:

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

2. For Instance pass, the script requires the arguments described below:

```
usage: enable_sriov.py [-h] --settings_file SETTINGS_FILE --script_pass
                       SCRIPT_PASS
                       [--director_install_user DIRECTOR_INSTALL_USER]
                       [--logfile LOGFILE] --ucrc UCRC --ocrc OCRC

Enable SR-IOV in Dell NFV Platform

Required Parameters:
  --settings_file SETTINGS_FILE
                        ini settings file, e.g ./sriov.ini
  --script_pass SCRIPT_PASS
                        script pass can be: {cnode|instance}
  --ucrc UCRC           Undercloud RC file
  --ocrc OCRC           Overcloud RC file
```

Optional Parameters:

```
-h, --help            show this help message and exit
  --director_install_user DIRECTOR_INSTALL_USER
                        User for undercloud/overcloud installation
  --logfile LOGFILE     name of the logfile
```

3. Edit the settings file as per the sample provided in *Appendix-A*.
4. Run script using the 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> --director_install_user <username>
```

Before executing the script, the following warning prompt displays:

```
INFO: 2017-10-10 12:29:46,910: Settings file loaded successfully.
INFO: 2017-10-10 12:29:47,126: Sanity checks completed successfully.
INFO: 2017-10-10 12:29:47,126:
INFO: 2017-10-10 12:29:47,127:
INFO: 2017-10-10 12:29:47,127: #     #
INFO: 2017-10-10 12:29:47,127: #  #  #    ##    #####   #    #     #    #    #   ####
INFO: 2017-10-10 12:29:47,127: #  #  #   #  #   #    #  ##   #     #    ##   #  #    #
INFO: 2017-10-10 12:29:47,127: #  #  #  #    #  #    #  # #  #     #    # #  #  #
INFO: 2017-10-10 12:29:47,127: #  #  #  ######  #####   #  # #     #    #  # #  #  ###
INFO: 2017-10-10 12:29:47,127: #  #  #  #    #  #   #   #   ##     #    #   ##  #    #
INFO: 2017-10-10 12:29:47,128:  ## ##   #    #  #    # #    #     #    #    #   ####
INFO: 2017-10-10 12:29:47,128:
INFO: 2017-10-10 12:29:47,128:
INFO: 2017-10-10 12:29:47,128: The enable_sriov script MAY reboot OpenStack Compute nodes multiple times.
INFO: 2017-10-10 12:29:47,128: Please save your work, stop pertinent services,
INFO: 2017-10-10 12:29:47,128:     and ask users to completely log out before proceeding.
INFO: 2017-10-10 12:29:47,128: Also, please do not reboot nodes manually until the
INFO: 2017-10-10 12:29:47,129:    enable_sriov script has completed execution.
INFO: 2017-10-10 12:29:47,129:
INFO: 2017-10-10 12:29:47,129:
Do you want to continue (y/n):
```

Figure 6: SR-IOV Instance warning

A successfully executed script has the following output:

```
INFO: 2017-05-22 12:21:22,394:
INFO: 2017-05-22 12:21:22,394: ******************** Instance VM3: instance pass FINAL SUMMARY ***************
INFO: 2017-05-22 12:21:22,394:
INFO: 2017-05-22 12:21:22,394: ------------------
INFO: 2017-05-22 12:21:22,394: Instance Pass Summary:
INFO: 2017-05-22 12:21:22,394: ------------------
INFO: 2017-05-22 12:21:22,394: Number of ping requests sent = 10
INFO: 2017-05-22 12:21:22,394:
INFO: 2017-05-22 12:21:22,394: Number of ping responses received = 10
INFO: 2017-05-22 12:21:22,394:
INFO: 2017-05-22 12:21:22,394: Instance Pass was a success!
INFO: 2017-05-22 12:21:22,395:
INFO: 2017-05-22 12:21:22,395: ------------------
[osp_admin@r141-director sriov]$
```

Figure 7: Successful completion of Instance pass

Chapter 4

NIC Alignment

Topics:

This chapter provides an overview of the NIC alignment feature in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Version 10.0.1.

Introduction

The goal of SR-IOV is to provide external network access to instances through the lowest latency path. It can further be optimized by using CPUs from same CPU Socket that is attached to the physical NIC. This ensures that CPUs handling the network interrupts inside the instance can directly access the incoming packets, instead of going through an interconnect to another CPU socket and accessing the required data. NUMA awareness and CPU pinning are a prerequisite for the NIC alignment feature. A short introduction of both terminologies is given below.

NUMA and CPU pinning ensures that the instance uses resources from a single NUMA node and its CPU pin assignment is dedicated. The physical functions are the NIC (Network Interface Cards) attached to the system via PCIe slots. Each PCIe slot has an affinity towards a single NUMA Node which means every PCIe slot is attached to a specific socket/NUMA Node for the purposes of Direct Memory Access (DMA). In SR-IOV, the virtual functions (VFs) are created from Physical Functions (PFs) and can be assigned to an instance. In NIC Alignment feature, it is ensured that VFs attached to an instance are from the NIC that has an affinity to the same socket/NUMA node on which the instance is running. This provides important performance improvements.

What is new

At the end of successful Cnode pass execution:

- The users are notified about the suitable NUMA node/socket for a virtual machine when all the PFs have an affinity to the same NUMA node/socket.
- The user are notified to select active-standby mode when the PFs does not have an affinity to the same NUMA node/socket.

During instance pass execution, when the instance's vCPUs and PFs have CPU affinity to different NUMA nodes/sockets:

- In case of active-active, the user will be warned that NIC Alignment feature cannot be enabled and will be given an option to abort.
- In case of active-standby, NIC Alignment feature will be enabled.

Overview

To use NIC alignment feature.

1. Run the enable_numa.py script with parameter --cpu_group.
2. Create the required instances.
3. Run the enable_sriov.py script with cnode pass mode.
4. Run the enable_sriov.py script with instance pass mode.
5. If instances are required on NUMA Node 0, rerun the enable_numa.py script with --action remove.

Steps to enable NIC alignment:

A detailed description of these steps is given below:

1. As discussed above, SR-IOV NICs have an affinity with NUMA node 1. NIC alignment can only be used if the instances are running on NUMA node 1. By running the enable_numa.py script with parameter

--cpu_group=numanode1, the script will associate NUMA node 1 with Nova and all instances created after that are scheduled on NUMA node 1.

```
$ python enable_numa.py --flavor_name NfvFlavor --cpu_group numanode1
```

2. Create all the required instances using the flavor “NfvFlavor”. This will enable CPU pinning for better performance. The user may add this metadata key to any other flavor and use that instead.

```
$ openstack flavor set <flavor-name> --property hw:cpu_policy=dedicated
$ openstack flavor set <flavor-name> --property hw:numa_nodes=1
```

3. Run the enable_sriov.py script in Cnode pass mode to create the virtual functions on the Compute node.

```
$ python enable_sriov.py --settings_file <compute-settings.ini> --
script_pass cnode --ucrc <undercloud-rc> --ocrc <overcloud-rc>
```

4. If instances are needed on NUMA Node 0, run the enable_numa.py script with --action remove.

Note: This removes the SR-IOV settings as compute nodes are rebooted. It is recommended to do this before applying steps 3 & 4.

```
$ python enable_numa.py --flavor_name NfvFlavor --action remove
```

Note: There are two metadata properties “hw:cpu_policy=dedicated” and “hw:numa_nodes=1”. These properties ensure that the instances spawned have pinned vCPUs are coming from only one NUMA Node. In this release, there is no way to ensure that instances are spawned on NUMA Node 1 that has SR-IOV NICs attached to it.

The output of successfully enabling NIC alignemnt is shown below.

```
INFO: 2017-05-22 12:21:22,394:
INFO: 2017-05-22 12:21:22,394: ****************** Instance VM3: instance pass FINAL SUMMARY ***************
INFO: 2017-05-22 12:21:22,394:
INFO: 2017-05-22 12:21:22,394: ------------------
INFO: 2017-05-22 12:21:22,394: Instance Pass Summary:
INFO: 2017-05-22 12:21:22,394: ------------------
INFO: 2017-05-22 12:21:22,394: Number of ping requests sent = 10
INFO: 2017-05-22 12:21:22,394:
INFO: 2017-05-22 12:21:22,394: Number of ping responses received = 10
INFO: 2017-05-22 12:21:22,394:
INFO: 2017-05-22 12:21:22,394: Instance Pass was a success!
INFO: 2017-05-22 12:21:22,395:
INFO: 2017-05-22 12:21:22,395: ------------------
[osp_admin@r141-director sriov]$
```

Figure 8: Instance pass output

Note: SR-IOV feature has been validation P2 and P3 slots only. P2 and P3 slots do not have an affinity to the same CPU sockets and NUMA nodes, the NIC alignment features will not be functional.

Appendix A

Sample settings files

Topics:

Cnode pass sample settings

```
[UNIVERSAL]
settingsVersion=10.0.1
# 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 10.0.1
scriptName=enable_sriov
scriptVersion=10.0.1
RHOSPVersion=10
JSVersion=10.0.1
# Valid values are: cnode | instance
# Must be specified by the user
scriptPass=cnode
# Valid values are: ephemeral (default mode) | persistent
# In NFV Wave 10.0.1, 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 10.0.1
num_vfs_per_pf=4
# PF specs are of the form: (PCI slot) (NIC port)
# e.g. p2p1 = PCI slot 2, NIC port 1
# Value(s) below are the only ones supported in NFV Wave 10.0.1
bond_groups={p1p1,p2p1},{p1p2,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 10.0.1
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 10.0.1
bw_strategy=equal
[CNODE]
cnodeName=R141nfv-compute-1
# ---------------
```

Instance pass sample settings

```
[UNIVERSAL]
settingsVersion=10.0.1
scriptName=enable_sriov
scriptVersion=10.0.1
RHOSPVersion=10
JSVersion=10.0.1
scriptPass=instance
scriptPassMode=ephemeral
# scriptPassMode=persistent
num_vfs_per_pf=4
bond_groups={p1p1,p2p1},{p1p2,p2p2}
bw_available_per_pf=10
bw_strategy=equal
[INSTANCE]
instanceName=rhel-vm1-compute1
```

```
[rhel-vm1-compute1]
behavior=bonding
bonding_on_platform=rhel7.3
numBonds=1
# bonding_failover_mode=a-s
bonding_failover_mode=a-a
bondIP=100.67.141.182
bondIPMaskLen=24
bondGatewayIP=100.67.141.254
userid=cloud-user
keyfile=key_name.pem
# ---------------
```

Appendix B

References

Topics:

- *To Learn More*

Additional information can be obtained at *http://www.dell.com/en-us/work/learn/openstack-cloud* or by e-mailing *openstack@dell.com*.

If you need additional services or implementation help, please contact your Dell EMC sales representative.

To Learn More

For more information on the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform visit *http://www.dell.com/learn/us/en/04/solutions/red-hat-openstack*.