Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform

Software Deployment Guide

Version 10.0.1

Dell EMC Converged Platforms and Solutions

Contents

List of Figures

List of Tables

Trademarks

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.

Chapter 1

Overview

Topics:

This guide provides information necessary to deploy the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform.

- **Servers**:
  - Dell EMC PowerEdge R630 and Dell EMC PowerEdge R730xd servers with the Dell EMC PowerEdge H730 disk controller
  - Dell EMC PowerEdge FX with Dell EMC PowerEdge FC630 and Dell EMC PowerEdge FD332
- **Networking**:
  - Dell Networking S4048T-ON and S4048-ON switches

Summary

This guide provides a detailed set of instructions on how to deploy the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform. The main topics described are:

- *Software Automation Deployment* on page 14 for **automated** deployment of:
  - Dell Dell EMC PowerEdge R-Series
  - Dell EMC PowerEdge FX
- *Software Manual Deployment* on page 22 for **manual** deployment of:
  - Dell Dell EMC PowerEdge R-Series
- *Overcloud Validation* on page 58

Several appendices provide file references, example files, and other helpful procedures.

Note: This document contains procedures for both automated and manual deployment. Each of these procedures has dependencies, including automation and other files that are required. Please see the *Dependencies* sections of *Software Automation Deployment* on page 14 and *Software Manual Deployment* on page 22, respectively, to understand the dependencies before you begin the procedures.

Prerequisites

The following prerequisites must be satisfied before proceeding with a Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform deployment:

Note: All nodes in the same roles must be of the same server models, with identical HDD, RAM, and NIC configurations. So, all Controller nodes must be identical to each other; all Compute nodes must be identical to each other; and so on. See the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Reference Architecture for configuration options for each node role.

- Hardware racked and wired per the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Reference Architecture
- Hardware configured as per the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Hardware Deployment Guide
- Hardware is powered off after the hardware is configured per the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Hardware Deployment Guide
- Internet access, including but not limited to, Red Hat's subscription manager service and repositories
- Valid Red Hat subscriptions
- Workstation used to to extract the *automated.tar.gz* file and begin building the collateral for the SAH node

Red Hat Subscriptions

Once all prerequisites have been met, you must determine the appropriate Red Hat subscription entitlements for each cluster node.

Determining Pool IDs

You must determine the pool ID to use for the Solution Admin Host (SAH) and each node in the cluster before proceeding with the installation. To determine the pool IDs, you must have an existing server that is registered to the Red Hat Hosted Services. This server must also be registered using the same credentials as the ones being used in this environment.

1. Once the server is correctly registered, execute the following command to see the available subscription pools.

   ```
   # subscription-manager list --all --available
   ```

   The command will output a list of available pools. Each section of information lists what the subscription provides, its pool ID, how many are available, the type of system it is for, as well as other information.
2. Determine the correct pool ID needed for this environment and take note of it.

   Note: Pay close attention to the **System Type**. The System Type can be *Virtual* or *Physical*. If necessary you can use a physical license for a virtual node. However, you cannot use a virtual license for a physical node.

```
# subscription-manager list --all --available

[OUTPUT ABBREVIATED]

Subscription Name: Red Hat Cloud Infrastructure, Standard (8-sockets)
Provides:          Red Hat Beta
                   Red Hat OpenStack Beta
                   JBoss Enterprise Application Platform
                   Red Hat Software Collections (for RHEL Server)
                   Red Hat Enterprise Virtualization
                   Oracle Java (for RHEL Server)
                   Red Hat OpenStack
                   Red Hat Enterprise MRG Messaging
                   Red Hat Enterprise Linux Server
                   Red Hat Enterprise Linux High Availability (for RHEL
 Server)
                   Red Hat Software Collections Beta (for RHEL Server)
                   Red Hat Enterprise Linux Load Balancer (for RHEL Server)
                   Red Hat CloudForms
SKU:               MCT2861
Pool ID:           aaaa111bbb222ccc333ddd444eee5556
Available:         7
Suggested:         1
Service Level:     Standard
Service Type:      L1-L3
Multi-Entitlement: No
Ends:              03/23/2017
System Type:       Physical

[OUTPUT ABBREVIATED]
```

The above output shows a subscription that contains the Red Hat OpenStack entitlement. The required entitlement types for each node are shown in *Table 1: Red Hat Subscription Entitlements* on page 12.

Table 1: Red Hat Subscription Entitlements

Node Role	Entitlement	System Type
Solution Admin Host	Red Hat Enterprise Linux Server	physical
Director Node	Red Hat OpenStack	virtual
Red Hat Storage Console VM	Red Hat Storage Console	physical (no virtual available at this time)
Controller Node	Red Hat OpenStack	physical

Node Role	Entitlement	System Type
Compute Node	Red Hat OpenStack	physical
Storage Node	Red Hat Ceph Storage	physical

Chapter 2

Software Automation Deployment

Topics:

This chapter describes the procedures used to deploy the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform using an automation framework developed by Dell EMC and validated by Red Hat.

Before You Begin

This guide assumes that you have racked the servers and networking hardware, and completed power and network cabling, as per the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Reference Architecture Guide.

The high-level steps required to install the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform using the automated installation procedures include:

1. Ensuring that your environment meets the *Prerequisites* on page 11
2. Ensuring that the *Dependencies* on page 15 are met
3. *Determining Pool IDs* on page 11
4. *Downloading and Extracting Automation Files* on page 15
5. *Preparing the Solution Admin Host Deployment* on page 16
6. *Deploying the SAH Node* on page 19
7. *Deploying the Undercloud and the OpenStack Cluster* on page 19

Dependencies

For customers performing a self-installation, these files are available upon request from Dell EMC. Please contact your account representative, or email *openstack@dell.com* for instructions.

Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform automated deployment dependencies include:

- *Downloading and Extracting Automation Files* on page 15
- The automated install also requires that you have a RHEL 7.3 ISO. It can be downloaded from the Red Hat Customer Portal here: *https://access.redhat.com/downloads/content/69/ver=/rhel---7/7.3/x86_64/product-software*

Automation Configuration Files

This chapter details obtaining the required configuration files.

Downloading and Extracting Automation Files

The following procedure installs the required configuration files and scripts used to build the collateral (*osp_ks.img*) to begin deploying the solution. This system must be a RHEL 7.3 system and is only used to build up the initial kickstart file. It will not be used again as it is a one-time use, and will not be allocated permanently in the customer's OpenStack deployment.

Note: The first archive *JetPack-automation-DERH-RB4NFV-10.0.1.tgz* contains the code related to automation of the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform 10.0.1. And second archive *NFV-Features-DERH-RB4NFV-10.0.1.tgz* contains code related to the post deployment features of the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform 10.0.1. If you need additional services or implementation help, contact your Dell EMC sales representative or email to *openstack@dell.com*

1. Log into your RHEL 7.3 system as user *root*.
2. Get the *JetPack-automation-DERH-RB4NFV-10.0.1.tgz* and NFV-Features-DERH-RB4NFV-10.0.1.tgz files to the */root* directory.
3. Change the working directory to */root*.

4. Extract the tar file contents:

```
# tar -xvf JetPack-automation-DERH-RB4NFV-10.0.1.tgz
```

```
# tar -xvf NFV-Features-DERH-RB4NFV-10.0.1.tgz
```

5. Download or copy the ISO of the Red Hat Enterprise Linux Server 7.3 installation DVD to */root/rhel73.iso*.

Preparing and Deploying the Solution Admin Host

This topic describes preparing for, and performing, the Solution Admin Host (SAH) deployment.

- If you are using Dell EMC PowerEdge R-Series servers the Open Source Hardware Configuration Toolkit (OS-HCTK) must be run **only on the SAH**. See the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform PowerEdge R-Series Hardware Deployment Guide.
- If you are using Dell EMC PowerEdge FX servers the OS-HCTK must be run **on all nodes**. See the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform PowerEdge FX Hardware Deployment Guide.

Note: PowerEdge FX BIOS and RAID configuration is not enabled in version 10.0.1 of the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform.

Preparing the Solution Admin Host Deployment

Caution: This operation will destroy all data on the Solution Admin Host, with no option for recovery.

1. Log in as your RHEL 7.3 system as the *root* user.
2. Change the working directory to */root/JetPack/src/deploy/osp_deployer/settings*.

```
cd ~/JetPack/src/deploy/osp_deployer/settings
```

3. Copy the settings files to stamp-specific files in different directories. Each architecture uses a specific set of sample settings files to which you can refer:

   Dell EMC PowerEdge R-Series

   - `/root/JetPack/src/deploy/osp_deployer/settings/sample.ini`
   - `/root/JetPack/src/deploy/osp_deployer/settings/sample.properties`

   Dell EMC PowerEdge FX

   - `/root/JetPack/src/deploy/osp_deployer/settings/sample-fx.ini`
   - `/root/JetPack/src/deploy/osp_deployer/settings/sample-fx.properties`

   a. If deploying a Dell EMC PowerEdge R-Series series hardware configuration:

      a. Copy the *sample.ini* and *sample.properties* files to stamp-specific files (e.g.,*acme.ini* and *acme.properties*), and place them in the ~/ directory:

```
# cp ~/JetPack/src/deploy/osp_deployer/settings/sample.properties ~/
acme.properties
# cp ~/JetPack/src/deploy/osp_deployer/settings/sample.ini ~/acme.ini
```

   b. If deploying a Dell EMC PowerEdge FX hardware configuration:

a. Copy the *sample-fx.ini* and *sample-fx.properties* files to stamp-specific files (e.g.,*acme.ini* and *acme.properties*), and place them in the ~/ directory:

```
# cp ~/JetPack/src/deploy/osp_deployer/settings/sample-fx.properties
 ~/acme.properties
# cp ~/JetPack/src/deploy/osp_deployer/settings/sample-fx.ini ~/
acme.ini
```

4. Edit your hardware stamp's *.ini* and *.properties* files to match your hardware stamp documentation (i.e., a Solution Workbook). Use a text editor of your choice; our example uses *vi*:

```
# vi ~/acme.ini
```

5. Change the values in your stamp-specific *.ini* file to match your specific environment. You must supply a value for each *CHANGEME* token in the file. In addition, the IP addresses and the Subscription Manager Pool IDs must be changed to match your deployment. Each section will have a brief description of the attributes.

Note: Instance HA and Ephemeral storage selections must both be performed at installation time, as Dell EMC does not support changing these settings post-installation.

Note: Numa and Hugepages features can be installed during automated deployment. Refer to Appendix D and E of this guide for additional information.

Note: SR-IOV feature can be installed after the solution has been deployed. Refer to Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform SR-IOV User Guide for additional information.

Note: OVS-DPDK can be installed during automated deployment. Refer to Appendix F of this Guide for additional information.

6. Edit the stamp-specific *.properties* file:

```
# vi ~/acme.properties
```

7. Change the values in your *.properties* file to match your specific environment. You must supply a value for IP addresses, host names, passwords, interfaces, and storage OSDs/journals.

Note: Additional nodes can be added to your stamp-specific *.properties* file if your environment contains more than that supported by the base architecture, as described in the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Architecture Guide.

The examples in this file are based on the Dell EMC Ready Bundle for Red Hat OpenStack NFV PlatformReference Architecture Guide, and the installation scripts rely on the VLAN IDs as specified in this file. For example, the Private API VLAN ID is 140. So, all addresses on the Private API network must have 140 as the third octet (e.g., 192.168.**140**.114). *Table 2: VLAN IDs* on page 17 below lists the VLAN IDs.

Table 2: VLAN IDs

VLAN ID	Name
110	Management/Out of Band (OOB) Network (iDRAC)
120	Provisioning Network
130	Tenant Network
140	Private API Network
170	Storage Network
180	Storage Clustering Network

VLAN ID	Name
190	Public API Network
191	External Tenant Network (Used for floating IP addresses)
201-250	Internal Tenant Network

Note: The *anaconda_ip* is used for the initial installation of the SAH node, and requires an address that can access the Internet to obtain Red Hat software. The *anaconda_iface* must be a dedicated interface that is only used for this purpose, and is not used in any other part of the configuration.

a. Configure the Overcloud nodes' iDRACs to use either DHCP or statically-assigned IP addresses. A mix of these two choices is supported.

  a. Determine the service tag of the Overcloud nodes whose iDRAC is configured to use DHCP.
  b. Determine the IP addresses of the Overcloud nodes whose iDRAC is configured to use static IP addresses.
  c. When creating the automation *.properties* file:

  - Add the following line to each node using DHCP, substituting the service tag for the node:

```
"service_tag": "<serviceTagHere>",
```

  - Add the following line to each node using static IP addressing:

```
"idrac_ip": "<idracIpHere>",
```

  Only `service_tag` or `idrac_ip` should be specified for each Overcloud node, not both.

  The iDRACs using DHCP will be assigned an IP address from the management allocation pool specified in the *.ini* file. The parameters that specify the pool range are:

  - `management_allocation_pool_start`
  - `management_allocation_pool_end`

  During deployment, the iDRACs using DHCP will be automatically assigned an IP address and discovered. The IP addresses assigned to the nodes can be seen:

  - In `/var/lib/dhcpd/dhcpd.leases` on the SAH Node
  - In `~/instackenv.json` on the Director Node
  - By executing the following commands on the Director Node:

```
$ ironic node-list
$ ironic node-show <node_guid>
```

8. Update your `python` path:

```
# export PYTHONPATH=/usr/bin/python:/lib/python2.7:/lib/python2.7/\
site-packages:~/JetPack/src/deploy
```

9. You can install the SAH node using either of the following methods:

- Using a physical USB key (key must have 8GBs minimum of capacity):

  1. Plug your USB key into your RHEL 7.3 system.
  2. Run the setup script to prepare your USB key, passing in the USB device ID (*/dev/sdb* in the example below). This process can take up to 10 minutes to complete.

  Note: Use full paths.

```
# cd ~/JetPack/src/deploy/setup
```

```
# python setup_usb_idrac.py -s /root/acme.ini -usb_key /dev/sdb
```

- Using an iDRAC virtual media image file. This requires your RHEL 7.3 system to have access to the iDRAC consoles to attach the image.

  1. Run the setup script to generate an image file that can later be attached to the SAH node.

     Note: Use full paths.

```
# cd ~/JetPack/src/deploy/setup
python setup_usb_idrac.py -s /root/acme.ini -idrac_vmedia_img
```

  2. The output will be an image file generated in *~/osp_ks.img*.

Deploying the SAH Node

You can deploy the SAH node by one of two methods:

- Using a physical USB key generated above, plugged into the SAH node, or
- Using an iDRAC virtual media image generated above, made available using the **Map Removable Media** option on the iDRAC.

Proceed to *Presenting the Image to the RHEL OS Installation Process* on page 19.

Presenting the Image to the RHEL OS Installation Process

1. Attach the Red Hat Enterprise Linux Server 7.3 ISO as a virtual CD/DVD using the **Virtual Media** -> **Map CD/DVD** option.
2. Attach the `~/osp_ks.img` created above by using either of the following methods:
   - As a removable disk using the **Virtual Media** -> **Map Removable Disk** option, or
   - Plug in the USB key created above into the SAH.
3. Set the SAH node to boot from the virtual CD/DVD using the **Next Boot** -> **Virtual CD/DVD/ISO** option.
4. Boot the SAH node.
   a. At the installation menu, select the **Install** option. *Do not press the [Enter] key*.
   b. Press the **Tab** key.
   c. Move the cursor to the end of the line that begins with `vmlinuz`.
   d. Append the following to the end of the line:

```
ks=hd:sdb:/osp-sah.ks
```

   Note: The device `sdb` can change, depending upon the quantity of disks being presented to the installation environment. These instructions assume that a single disk is presented. If otherwise, adjust accordingly.

5. Press the **[Enter]** key to start the installation.

Note: It may take a few minutes before progress is seen on the screen. Press the *[ESC]* key at the memory check to speed up the process.

Deploying the Undercloud and the OpenStack Cluster

Now that the SAH node is installed you can deploy and validate the rest of the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform nodes.

Deploying and Validating the Cluster

Caution: This operation will destroy all data on the identified servers, with no option for recovery.

To deploy and validate the rest of the cluster:

1. Log in through the iDRAC console as *root*, or `ssh` into the SAH node.
2. Mount the USB media:

```
# mount /dev/sdb /mnt
```

3. Copy all the files locally:

```
# cp -rfv /mnt/* /root
```

4. Start a `tmux` session to avoid losing progress if the connection drops:

```
# tmux
```

5. There are some post-deployment validation options in the `[Bastion Settings]` group of the stamp-specific initialization file you should consider prior to deployment:
   - `run_sanity_test` - If set to *true* the `sanity_test.sh` script will be executed that will verify the basic functionality of your overcloud deployment.
   - `run_tempest` - If set to *true* the Tempest integration test suite will be executed against your overcloud deployment.

     Note: Tempest requires that the sanity test must be run first so `run_sanity`, above, must also be set to *true*. For more information about Tempest please refer to the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Overcloud Validation Guide.
   - `tempest_smoke_only` - If `run_tempest`, above, is set to *true* this option, which is set to *true* by default, will cause Tempest to run only a small subset of the test suite, where the tests are tagged as *"smoke"*. If set to *false* the entire Tempest suite will be run, which can take an hour or more to complete.
6. Run the deployment by executing the `deployer.py` command:

```
# cd /root/JetPack/src/deploy/osp_deployer
# python deployer.py -s <path_to_settings_ini_file> [-undercloud_only]
[-overcloud_only] [-skip_rhscon_vm]
```

   Optional arguments include:
   - `-undercloud_only` = Reinstall **only** the Undercloud
   - `-overcloud_only` = Reinstall **only** the Overcloud
   - `-skip_rhscon_vm` = Do not reinstall the Red Hat Storage Console VM
7. For installation details, execute a `tail` command on the */auto_results/deployer.log.xxx* file on the SAH node. For example:

```
# tail -f /auto_results/deployer.log.2017.04.17-09.09
```

8. If issues are discovered during the installation process:
   a. Identify the issue in the *deployer.log*
   b. Address the issue.
   c. Rerun the `python deployer.py` command above.
9. If the installation is successful, the *deployment_summary.log* file will display some useful information for accessing the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform.

```
# cd /auto_results
# cat deployment_summary.log
```

   The output will appear similar to this:

```
====================================
```

```
### nodes ip information ###
### Controllers ###
mercury-controller-0  :
     - provisioning ip  : 192.168.120.134
     - nova private ip  : 192.168.140.22
     - nova public ip   : 192.168.190.32
     - storage ip       : 192.168.170.21
mercury-controller-1  :
     - provisioning ip  : 192.168.120.128
     - nova private ip  : 192.168.140.26
     - nova public ip   : 192.168.190.34
     - storage ip       : 192.168.170.26
mercury-controller-2  :
     - provisioning ip  : 192.168.120.126
     - nova private ip  : 192.168.140.23
     - nova public ip   : 192.168.190.33
     - storage ip       : 192.168.170.22
### Compute  ###
mercury-compute-0     :
     - provisioning ip  : 192.168.120.133
     - nova private ip  : 192.168.140.25
     - storage ip       : 192.168.170.25
mercury-compute-1     :
     - provisioning ip  : 192.168.120.132
     - nova private ip  : 192.168.140.24
     - storage ip       : 192.168.170.24
### Storage  ###
mercury-cephstorage-0 :
     - provisioning ip    : 192.168.120.131
     - storage cluster ip : 192.168.180.20
     - storage ip         : 192.168.170.23
mercury-cephstorage-1 :
     - provisioning ip    : 192.168.120.127
     - storage cluster ip : 192.168.180.21
     - storage ip         : 192.168.170.27
====================================
OverCloud Horizon        : http://192.168.190.31:5000/v2.0

OverCloud admin password : GbXkxG99KtxHtbmTVzMK9QnUv
```

Network Checks

The deployment process performs a series of network connectivity checks, from the following hosts:

- **SAH Node** — checks are run immediately when you run `deployer.py`
- **Director Node VM** — checks are run immediately when the node is deployed
- **Red Hat Storage Console VM** — checks are run immediately when the node is deployed

Note: If any of the checks fail the deployment will abort.

The network connectivity checks include:

- Ping the `public_api_gateway`
- Ping a Google DNS server (`8.8.8.8`)
- Ping the Google website (`www.google.com`)
- Execute `subscription-manager status`, which uses:
  - `subscription.rhsm.redhat.com` on port *443*
  - `cdn.redhat.com` on port *443*

Chapter 3

Software Manual Deployment

Topics:

This chapter describes the procedures used to deploy the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform using a manual framework developed by Dell EMC.

Note: Manual deployment is not supported for any of the NFV features. Manual deployment is supported only for base open stack installation.

Before You Begin

Deployment of the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform on Dell EMC hardware begins with installation of the Solution Admin Host (SAH). Before proceeding to the SAH installation there are some prerequisites that must first be satisfied, such as obtaining Red Hat subscriptions and gathering of proxy information.

Dell EMC recommends that you first review this chapter to familiarize yourself with terms and concepts before proceeding. Another document that would be helpful to review and print out is the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Workbook. The workbook contains essential information that you will need as you proceed through this document.

To deploy the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform, complete the following tasks:

1. *Determining Pool IDs* on page 11
2. *Solution Admin Host Deployment* on page 23
3. *Red Hat OpenStack Platform Director Node Deployment* on page 29
4. *Red Hat Storage Console VM Deployment* on page 33
5. *Provisioning the Nodes* on page 36
6. *Red Hat Storage Console Configuration* on page 57

Note: Performing all of these tasks is very complex, so please take your time and follow the steps closely.

Dependencies

For customers performing a self-installation, these files are available on request from Dell EMC. Please contact your account representative, or email *openstack@dell.com* for instructions.

Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform manual deployment dependencies include:

- *Solution Files* on page 63

Solution Admin Host Deployment

The SAH hosts at least two (2) virtual machines (VMs):

- **Red Hat OpenStack Platform Director Node** - Used for hosting Red Hat OpenStack Platform Director deployment software, configuring OpenStack, and deployment of OpenStack software to the Controller, Compute, and Storage nodes
- **Red Hat Storage Console VM** - Used for hosting the Red Hat Storage Console web-based management platform

Solution Admin Host Deployment Overview and Prerequisites

Installation of the Dell EMC Solution Admin Host begins with the installation of Red Hat Enterprise Linux Server 7.3. The *osp-sah.ks* kickstart file is provided to assist automation of this process. The installation process can be accomplished using different processes (CD-ROM, CD image, or via a PXE installation). Instructions for how to include the kickstart file are provided later in this document.

This kickstart file performs the following steps when properly configured:

- Partitions the system

- Sets SELinux to *permissive* mode
- Disables `firewalld`, and uses `iptables`
- Disables NetworkManager
- Configures networking, including:
  - Bonding
  - Bridges
  - Static IP addresses
  - Gateway
  - Name resolution
  - NTP service
- Registers the system using the Red Hat Subscription Manager

Additionally, there are some requirements that must be satisfied prior to installation of the OS:

- A Red Hat subscription
- Access to the Subscription Manager hosts

Note: If your network configuration/firewall require them, you must provide the proxy values in order to access Red Hat's Subscription Manager servers.

The osp-sah.ks Kickstart File

The *JetPack-manual-10.0.1.tgz* archive file (see *File References* on page 62) contains the *osp-sah.ks* kickstart file, and must be customized for the environment into which it is being installed.

Note: All edits that usually require changes are in the section marked **CHANGEME** and **END of CHANGEME**. Do not make other edits outside of these lines.

There are many changes that you will need to make, so a brief description of the SAH networks might help clarify the need for the variables in *Figure 1: Solution Admin Host Internal Network Fabric* on page 24.

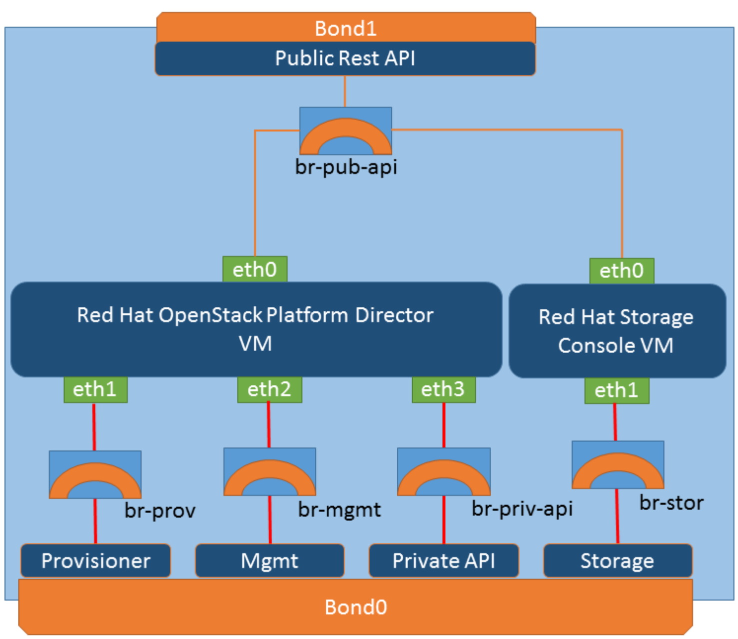

Figure 1: Solution Admin Host Internal Network Fabric

There are 4 network interface files, created during the OS installation, that are required by the SAH:

- `ifcfg-em1`
- `ifcfg-em2`
- `ifcfg-p1p1`
- `ifcfg-p1p2`

> **Note:** The interfaces names (`ifcfg-em1`, `ifcfg-em2`, `ifcfg-p1p1` and `ifcfg-p1p2`) might be different on different system configurations, so these might need to change for your hardware configuration.

These are combined to build the bond interfaces (`bond0` and `bond1`). Interfaces `em1` and `p1p1` (`private_ifaces`) are combined to form `bond0` (`private_bond_name`) which is used by both the provisioning and storage networks. Interfaces `em2` and `p1p2` (`public_ifaces`) are combined to form `bond1` (`public_bond_name`). The other bond variables listed in the table are the `public_boot_opts`, `public_bond_opts`, `private_boot_opts` and `private_bond_opts`. These four variables can be left as is unless your configuration requires their modification.

Two VLANs are created to segregate the provisioning and storage traffic across the private network. For our example, we set the `provision_bond_name` to `bond0.120` and `storage bond_name` to `bond0.170` and the `provision_boot_opts` and `storage boot_opts` to `"onboot none vlan"`. You must configure the values for these variables to match your configuration.

The SAH requires three bridges to allow network traffic to pass through the SAH to and from the VM instances. The `public_bridge_boot_opts, provision_bridge_boot_opts` and `storage_bridge_boot_opts` variables are the last changes you must make to the kickstart file. For these variables you must set the boot options, IP address and netmask. The `osp-sah.ks` kickstart file provides an example of the format required.

To customize the kickstart file:

1. Extract the contents of the archive file using the `tar -zxf` command or a Windows® archive utility (*7zip*, *WinRaR*, etc).
2. Extract the contents of the *dell-mgmt-node.tgz* file, where you will find the *osp-sah.ks* that you will change as described below.
3. Set the following variables:

Table 3: Kickstart File Variables

Variable	Description
HostName	The FQDN of the server, e.g., sah.acme.com.
SystemPassword	The root user password for the system.
SubscriptionManagerUser	The user credential when registering with Subscription Manager.
SubscriptionManagerPassword	The user password when registering with Subscription Manager.
SubscriptionManagerPool	The pool ID used when attaching the system to an entitlement.
SubscriptionManagerProxy	Optional proxy server to use when attaching the system to an entitlement.
SubscriptionManagerProxyPort	Optional port for the proxy server.
SubscriptionManagerProxyUser	Optional user name for the proxy server.
SubscriptionManagerProxyPassword	Optional password for the proxy server.
Gateway	The default gateway for the system.

Variable	Description
NameServers	A comma-separated list of nameserver IP addresses.
NTPServers	A comma-separated list of time servers. This can be IP addresses or FQDNs.
TimeZone	The time zone in which the system resides.
anaconda_interface	The public interface that allows connection to Red Hat Subscription services.
extern_bond_name	The name of the bond that provides access to the external network.
extern_boot_opts	The boot options for the bond on the external network. Typically, there is no need to change this variable.
extern_bond_opts	The bonding options for the bond on the external network. Typically, there is no need to change this variable.
extern_ifaces	A space delimited list of interface names to bond together for the bond on the external network.
internal_bond_name	The name of the bond that provides access for all internal networks.
internal_boot_opts	The boot options for the bond on the internal network. Typically, there is no need to change this variable.
internal_bond_opts	The bonding options for the bond on the internal network. Typically, there is no need to change this variable.
internal_ifaces	A space delimited list of interface names to bond together for the bond on the internal network.
mgmt_bond_name	The VLAN interface name for the management network.
mgmt_boot_opts	The boot options for the management VLAN interface. Typically, there is no need to change this variable.
prov_bond_name	The VLAN interface name for the provisioning network.
prov_boot_opts	The boot options for the provisioning VLAN interface. Typically, there is no need to change this variable.
stor_bond_name	The VLAN interface name for the storage network.
stor_boot_opts	The boot options for the storage VLAN interface. Typically, there is no need to change this variable.
pub_api_bond_name	The VLAN interface name for the public API interface.

Variable	Description
pub_api_boot_opts	The boot options for the public API VLAN interface. Typically, there is no need to change this variable.
priv_api_bond_name	The VLAN interface name for the private API interface.
priv_api_boot_opts	The boot options for the private API VLAN interface. Typically, there is no need to change this variable.
br_mgmt_boot_opts	The bonding options, IP address and netmask for the management bridge.
br_prov_boot_opts	The bonding options, IP address and netmask for the provisioning bridge.
br_stor_boot_opts	The bonding options, IP address and netmask for the storage bridge.
br_pub_api_boot_opts	The bonding options, IP address and netmask for the public API bridge.
br_priv_api_boot_opts	The bonding options, IP address and netmask for the private API bridge.

Making the Kickstart File Available for Installation

This procedure places the kickstart file in the top level of a USB image and makes it available for installation. This is useful if you are using the iDRAC virtual media to install.

Note: The USB key must be formatted as *vfat* or *ext2*.

The following example performs these steps:

1. Creates a *ks* image
2. Mounts the image through a loopback device
3. Copies the *osp-sah.ks* file to the image
4. Unmounts the image from the system

The resulting image can be used as removable media for PXE boot or iDRAC. As this is an example, please refer to the man pages or the reference manual for further information on the commands.

There are several options for presenting the *osp-sah.ks* to the OS installation. Below are two ways of preparing the image prior to presenting it to the OS installation for inclusion into the installation process:

- *Preparing an Image File for use with iDRAC (Option 1)* on page 27
- *Preparing a USB Key for Physical Boot (Option 2)* on page 28

Preparing an Image File for use with iDRAC (Option 1)

1. From an existing RHEL 7.3 system, create a USB image:

```
mkfs.vfat -C ks_usb.img 1024
```

2. Mount the image:

```
mount -o loop ks_usb.img /mnt
```

3. Place the `osp-sah.ks` file into the image:

```
cp osp-sah.ks /mnt
```

4. Unmount the image:

```
sync; umount /mnt
```

5. Make the image file, ks_usb.img, available using the *Map the image as Removable Media* option on the iDRAC.

   Note: If only one physical hard disk is presented to the server, the device name presented to the installer should be `sdb`.

Preparing a USB Key for Physical Boot (Option 2)

1. From an existing RHEL 7.3 system, format a USB key:

```
mkfs.ext3 /dev/sdb
```

2. Mount the USB key.

```
mount /dev/sdb /mnt
```

3. Place the `osp-sah.ks` file onto the USB key:

```
cp osp-sah.ks /mnt
```

4. Unmount the image:

   Note: If only one physical hard disk is presented to the server, the device name presented to the installer should be `sdb`.

```
sync; umount /mnt
```

Presenting the Image to the RHEL OS Installation Process

1. Boot the Solution Admin Host using the Red Hat Enterprise Linux Server 7.3 installation media.
   a. At the installation menu, select the **Install** option. *Do not press the [Enter] key*.
   b. Press the **Tab** key.
   c. Move the cursor to the end of the line that begins with `vmlinuz`.
   d. Append the following to the end of the line:

      Note: The device `sdb` can change, depending upon the quantity of disks being presented to the installation environment. These instructions assume that a single disk is presented. If otherwise, adjust accordingly.

```
ks=hd:sdb:/osp-sah.ks
```

2. Press the **[Enter]** key to start the installation.

Note: It may take a few minutes before progress is seen on the screen. Press the ESC key at the memory check to speed up the process.

Next Steps

After the SAH is installed:

1. Copy the ISO of the **Red Hat Enterprise Linux Server 7.3 installation DVD** to the `/store/data/iso` directory. **Only RHEL 7.3 is supported**. This ISO is used to install the Red Hat OpenStack Platform Director Node and Red Hat Storage Console VM.
2. Set up the Director Node by following the procedures in *Red Hat OpenStack Platform Director Node Deployment* on page 29.

Red Hat OpenStack Platform Director Node Deployment

The deployment of the Red Hat OpenStack Platform Director Node (Director Node) is performed using the `deploy-director-vm.sh` script. This script creates a kickstart file and then executes the `virt-install` command to install the system.

The Director Node Kickstart file

The generated kickstart script performs the following steps:

- Partitions the system
- Sets SELinux to enforcing mode
- Configures `iptables` to ensure the following services can pass traffic:
  - HTTP
  - HTTPS
  - DNS
  - TFTP
  - TCP port 8140
- Configures networking, including:
  - Static IP addresses
  - Gateway
  - Name resolution
  - NTP time service
- Registers the system using the Red Hat Subscription Manager
- Installs the Red Hat OpenStack Platform Director installer

Setup

To set up the Director Node deployment:

1. Log into the SAH node as the *root* user.
2. Ensure that a copy of the **Red Hat Enterprise Linux Server 7.3 Installation DVD** (RHEL 7.3 ISO) is in the `/store/data/iso` directory.
3. Download the *dell-mgmt-node* archive (zip or tgz) file and extract the contents of the archive file into the */root* directory as per the example below:

```
# cd /root
# tar zxvf /PATH/TO/FILE/dell-mgmt-node.tgz
```

Configuration

To configure the Director Node deployment:

1. Edit the configuration file, named `director.cfg`, in the `/root/mgmt` directory.
2. Set the following variables in the `director.cfg` file:

Table 4: Director Node Configuration Parameters

Parameter	Description
rootpassword	The root user password for the system.
timezone	The timezone the system is in.

Parameter	**Description**
smuser	The user credential when registering with Subscription Manager.
smpassword	The user password when registering with Subscription Manager. The password must be enclosed in single quotes if it contains certain special characters.
smpool	The pool ID used when attaching the system to an entitlement.
smproxy	Optional proxy server to use when deploying the Director Node. Format is `<proxy address>`:`<proxy port>` (e.g., `192.168.75.230`:3128).
smproxyuser	Optional user credential for the proxy server.
smproxypassword	Optional user password for the proxy server.
hostname	The FQDN of the Director Node.
gateway	The default gateway for the system.
nameserver	A comma-separated list of nameserver IP addresses.
ntpserver	A comma-separated list of time servers. This can consist of IP addresses or FQDNs.
user	The ID of an admin user to create to use for installing Red Hat OpenStack Platform Director. Default admin user is *stack*.
password	The password for the admin user.
eth0	This line specifies the IP address and network mask for the public API network. The line begins with `eth0`, followed by at least one space, the IP address of the VM on the public API network, another set of spaces, and then the network mask.
eth1	This line specifies the IP address and network mask for the provisioning network. The line begins with `eth1`, followed by at least one space, the IP address of the VM on the provisioning network, another set of spaces, and then the network mask.
eth2	This line specifies the IP address and network mask for the management network. The line begins with `eth2`, followed by at least one space, the IP address of the VM on the management network, another set of spaces, and then the network mask.

Parameter	Description
eth3	This line specifies the IP address and network mask for the private API network. The line begins with `eth3`, followed by at least one space, the IP address of the VM on the private API network, another set of spaces, and then the network mask.

Installing the Red Hat OpenStack Platform Director Node

To install the Director Node:

1. Invoke the `deploy-director-vm.sh` script.
   a. Pass `director.cfg` as the first parameter.
   b. Pass the full path to the Red Hat Enterprise Linux Server 7.3 Installation media as the second parameter.

```
# ./deploy-director-vm.sh director.cfg /store/data/iso/rhel73.iso
Starting install...
Retrieving file .treeinfo...
| 3.2 kB 00:00:00
Retrieving file vmlinuz...
| 7.9 MB 00:00:00
Retrieving file initrd.img...
| 64 MB 00:00:00
Creating storage file director.img
| 16 GB 00:00:00
Creating domain...
| 0 B 00:00:00
Domain installation still in progress. You can reconnect to
the console to complete the installation process.
```

   The installation will begin, but no console will be displayed.
2. To display the console:
   a. Ensure that you are logged into a GUI environment.
   b. Open a terminal.
   c. Enter the following command:

   Note: If you are connected to the Director Node using a Windows® system, you must install and configure an **Xwin Server** before executing `virt-viewer director` to see the output.

```
virt-viewer director
```

3. A console for the Director Node will open.
4. After the Director Node completes the installation, it will power itself off.
5. The power state of the Director Node can be viewed using the `virsh list --all` command:

```
Id Name State
----------------------------------------------------
2 director shut off
```

6. The Director Node can be started using the following command:

```
# virsh start director
```

Installing Red Hat OpenStack Platform Director

To install Red Hat OpenStack Platform Director:

1. Log into the Director Node using the user name and password specified in *director.cfg*.
2. Download the *dell-pilot.tgz* archive file. See *Solution Files* on page 63 for a list of solution files for the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform.
3. Extract the contents of the `tar` archive file into the home directory of the user:

```
$ cd
$ tar xzvf /PATH/TO/FILE/dell-pilot.tgz
```

4. Subscribe to, and enable, RHOSP repositories:

```
$ sudo ~/pilot/enable-repos.py [--proxy <proxy>]
```

Where:

- `proxy` = `'http://<proxy_user>:<proxy_password>@<proxy_address>:<proxy_port>'`

Example:

```
sudo ~/pilot/enable-repos.py --proxy 'http:// \
 proxyuser:password123@192.168.75.230:3128'
```

5. Edit the `~/pilot/undercloud.conf` configuration file.
6. Set the following variables in `undercloud.conf`:

Table 5: Undercloud Configuration Parameters

Parameter	Description
undercloud_hostname	Defines the fully-qualified host name for the Undercloud.
local_ip	The IP address and prefix of the Director Node on the provisioning network in Classless Inter-Domain Routing (CIDR) format (xx.xx.xx.xx/yy). This must be the IP address used for *eth1* in *director.cfg*. The prefix used here must correspond to the netmask for *eth1* as well (usually *24*).
network_gateway	The network gateway for Neutron-managed Overcloud instances.
network_cidr	The network and prefix in CIDR format for the Neutron managed network for Overcloud instances.
masquerade_network	The network address and prefix of the Director Node on the provisioning network in CIDR format (xx.xx.xx.xx/yy). This must be the network used for *eth1* in *director.cfg*. The prefix used here must correspond to the netmask for *eth1* as well (usually *24*).
dhcp_start	The starting IP address on the provisioning network to use for OpenStack cloud nodes. **Note:** Ensure the IP address of the Director Node is not included.

Parameter	Description
dhcp_end	The ending IP address on the provisioning network to use for OpenStack cloud nodes.
inspection_iprange	An IP address range on the provisioning network to use during node inspection. **Note:** This should not overlap with the *dhcp_start/dhcp_end* range.

7. Set the passwords, if desired, in the `[auth]` section; otherwise passwords will be randomly generated.

   Note: The *undercloud_heat_encryption_key* parameter **must** be either 16, 24, or 32 characters in length in order for Red Hat OpenStack Platform Director to successfully create a stack.

8. Determine the IP address of a DNS server for the Overcloud nodes to use.
9. Execute the following command:

```
$ ~/pilot/install-director.sh --dns <dns_ip> --sm_user \
 <subscription_manager_user> --sm_pwd <subscription_manager_pass> \
    [--sm_pool <subscription_manager_poolid>] [--proxy <proxy> \
    --nodes_pwd <overcloud_nodes_password>]
```

   Where:

- `dns_ip` = IP address of the DNS server
- `subscription_manager_user` = Red Hat Subscription Manager user, as specified in *Determining Pool IDs* on page 11
- `subscription_manager_pass` = Red Hat Subscription Manager user password, as specified in *Determining Pool IDs* on page 11
- `subscription_manager_poolid` = ID of the **Red Hat Ceph Storage** pool, as specified in *Determining Pool IDs* on page 11
- `proxy` = `'http://<proxy_user>:<proxy_password>@<proxy_address>:<proxy_port>'`
- `overcloud_nodes_password` = Root password for the Overcloud nodes

The Red Hat OpenStack Platform Director installation can take approximately 30 minutes to complete.

Note: The installation log is available at *~/pilot/install-director.log*.

Using Red Hat OpenStack Platform Director

The Red Hat OpenStack Platform Director installer creates an *rc* file for using the CLI commands, and a file containing all passwords.

1. Before running any CLI commands, first `source` the *rc* file:

```
source ~/stackrc
```

2. All of the passwords are in the `undercloud-passwords.conf` file.

Next Steps

After the Red Hat OpenStack Platform Director Node is installed:

1. Install the Red Hat Storage Console VM by following the procedures in *Red Hat Storage Console VM Deployment* on page 33.

Red Hat Storage Console VM Deployment

The Red Hat Storage Console VM is deployed using the `deploy-rhscon-vm.py` script. This script creates a kickstart file and then executes the `virt-install` command to deploy the VM.

The Red Hat Storage Console VM Kickstart File

The generated kickstart script performs the following steps:

- Partitions the system
- Sets SELinux to enforcing mode
- Configures `iptables` to run on the system and disables `firewalld`
- Configures networking, including:
  - Static IP addresses
  - Gateway
  - Name resolution
  - NTP time service
- Registers the system using the Red Hat Subscription Manager

Setup

To set up the Red Hat Storage Console VM deployment:

1. Log into the SAH node as the *root* user.
2. Ensure that a copy of the **Red Hat Enterprise Linux Server 7.3 Installation DVD** (RHEL 7.3 ISO) is in the `/store/data/iso` directory.

   Several steps in this document use files to configure the environment. See *File References* on page 62 for a list of required files.

   Note: Installation of these files should have been completed earlier in *Red Hat OpenStack Platform Director Node Deployment* on page 29.

Configuration

To configure the Red Hat Storage Console VM deployment:

1. Edit the `rhscon.cfg` configuration file, in the `/root/mgmt` directory.
2. Set the following variables in `rhscon.cfg`:

Table 6: Red Hat Storage Console VM Configuration Variables

Parameter	Description
rootpassword	The root user password for the Red Hat Storage Console VM.
timezone	The timezone in which the Red Hat Storage Console VM is located.
smuser	The user credential when registering with Subscription Manager.
smpassword	The user password when registering with Subscription Manager. The password must be enclosed in single quotes if it contains certain special characters.
smpool	The pool ID used when attaching the Red Hat Storage Console VM to an entitlement.
smproxy	Optional proxy server to use when deploying the Director Node. Format is `<proxy address>`:`<proxy port>` (e.g., `192.168.75.230`:3128).

Parameter	Description
smproxyuser	Optional user credential for the proxy server.
smproxypassword	Optional user password for the proxy server.
hostname	The FQDN of the Red Hat Storage Console VM.
gateway	The default gateway for the Red Hat Storage Console VM.
nameserver	A comma-separated list of nameserver IP addresses.
ntpserver	A comma-separated list of time servers. This can consist of IP addresses or FQDNs.
eth0	This line specifies the IP address and network mask for the public API network. The line begins with `eth0`, followed by at least one space, the IP address, another set of spaces, and then the network mask.
eth1	This line specifies the IP address and network mask for the storage network. The line begins with `eth1`, followed by at least one space, the IP address, another set of spaces, and then the network mask.

Deploying the Red Hat Storage Console VM

To deploy the Red Hat Storage Console VM:

1. Invoke the `deploy-rhscon-vm.py` script.
   a. Pass **`rhscon.cfg`** as the first parameter.
   b. Pass the **full path to the Red Hat Enterprise Linux Server 7.3 Installation media** as the second option.

```
# ./deploy-rhscon-vm.py rhscon.cfg /store/data/iso/rhel73.iso

Retrieving file .treeinfo...                             | 2.1 kB 00:00:00
Retrieving file vmlinuz...                               | 5.1 MB 00:00:00
Retrieving file initrd.img...                            |  43 MB 00:00:00
Creating domain...                                       |    0 B 00:00:00
Domain installation still in progress. You can reconnect to
the console to complete the installation process.
```

   The installation begins, but no Linux console is displayed.

2. To display the Linux console:
   a. Ensure you are logged into a GUI environment.
   b. Open a terminal.
   c. Enter the following command:

   Note: If you are connected to the Red Hat OpenStack Platform Director Node using a Windows® system, you need to install **Xwin Server** before executing `virt-viewer rhscon`.

```
# virt-viewer rhscon
```

3. A console for the Red Hat Storage Console VM will open.

4. After the Red Hat Storage Console VM completes the installation, it will power itself off.
5. The power state of the Red Hat Storage Console VM can be viewed using the following command:

```
# virsh list --all
```

You will see output similar to the following:

```
Id    Name                            State
 ----------------------------------------------------
 2     rhscon                         shut off
```

6. You can start the Red Hat Storage Console VM by entering the following command:

```
# virsh start rhscon
```

Next Steps

After the Red Hat Storage Console VM is installed:

1. Provision the Controller, Compute, and Storage nodes by following the procedures in *Provisioning the Nodes* on page 36.
2. Configure the Red Hat Storage Console by following the procedures in *Red Hat Storage Console Configuration* on page 57.

Provisioning the Nodes

This topic describes provisioning the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform cluster nodes. The following procedures are discussed in the order in which they are to be performed:

Configuring Networking

To configure network environment parameters:

1. On the Red Hat OpenStack Platform Director node, navigate to the *~/pilot/templates/* directory.
2. Open the *network-environment.yaml* file in an editor.
3. Search for the term, *CHANGEME*, to go to the lines in which changes can be made.
4. Make changes, as required, to the parameters listed in *Table 7: network-environment.yaml File Parameters* on page 37.

Note: Each parameter has a default value, which may or may not require a change depending upon your environment.

Table 7: network-environment.yaml File Parameters

Parameter Name	Default Value	Description
ManagementNetCidr	192.168.110.0/24	CIDR block for the Management network
InternalApiNetCidr	192.168.140.0/24	CIDR block for the Private API network.
TenantNetCidr	192.168.130.0/24	CIDR block for the Tenant network. For future support of Generic Routing Encapsulation (GRE) or VXLAN networks.
StorageNetCidr	192.168.170.0/24	CIDR block for the Storage network.
StorageMgmtNetCidr	192.168.180.0/24	CIDR block for the Storage Clustering network.
ExternalNetCidr	192.168.190.0/24	CIDR block for the External network.
ManagementAllocationPools	`[{'start': '192.168.110.10', 'end': '192.168.110.149'}]`	IP address range on the Management network for use by the iDRAC DHCP server.
InternalApiAllocationPools	`[{'start': '192.168.140.20', 'end': '192.168.140.200'}]`	IP address range for the Private API network.[1]
TenantAllocationPools	`[{'start': '192.168.130.20', 'end': '192.168.130.200'}]`	IP address range for the Tenant network. Not used unless you wish to configure Generic Routing Encapsulation (GRE) or VXLAN networks.
StorageAllocationPools	`[{'start': '192.168.170.20', 'end': '192.168.170.200'}]`	IP address range for the Storage network.
StorageMgmtAllocationPools	`[{'start': '192.168.180.20', 'end': '192.168.180.200'}]`	IP address range for the Storage Clustering. network
ExternalAllocationPools	`[{'start': '192.168.190.20', 'end': '192.168.190.120'}]`	IP address range for the External network.[2]
ExternalInterfaceDefaultRoute	192.168.190.1	Router gateway on the External network.

[1] To avoid IP address conflicts that could result in nodes not being provisioned, ensure that the Director Node's Internal API network interface resides outside the *InternalApiAllocationPools* parameter range of 192.168.140.40 - 192.168.140.200.

[2] Similarly, ensure that the Director Node's External Network interface resides outside the *ExternalAllocationPools* parameter range of 192.168.190.20 - 192.168.190.120.

Parameter Name	Default Value	Description
ManagementNetworkGateway	192.168.110.1	The IP address of the gateway on the Management network.
ProvisioningNetworkGateway	192.168.120.1	The IP address of the gateway on the Provisioning network, which allows access to the Management network.
ControlPlaneDefaultRoute	192.168.120.13	Router gateway on the provisioning network (or Undercloud IP address).
ControlPlaneSubnetCidr	24	CIDR of the control plane network.
EC2MetadataIp	192.168.120.13	IP address of the Undercloud.
DnsServers	`["8.8.8.8", "8.8.4.4"]`	DNS servers for the Overcloud nodes to use (maximum 2).[3]
InternalApiNetworkVlanID	140	VLAN ID of the Private API network.
StorageNetworkVlanID	170	VLAN ID of the Storage network.
StorageMgmtNetworkVlanID	180	VLAN ID of the Storage Clustering network.
TenantNetworkVlanID	130	VLAN ID of the Tenant network. For future support of Generic Routing Encapsulation (GRE) or VXLAN networks.
ExternalNetworkVlanID	190	VLAN ID of the External network.
BondInterfaceOptions	`"mode=802.3ad miimon=100"`	Bonding mode for all nodes.[4]
NeutronExternalNetworkBridge	`""`	Empty string for External VLAN, or *br-ex* if on the native VLAN.

Configure iDRAC DHCP Server

To configure the DHCP server that the iDRACs use:

1. Execute the following command, passing in the IP address of the SAH on the provisioning network:

```
$ cd ~/pilot
./config_idrac_dhcp.py <SAH_provisioning_IP>
```

Note: The DHCP server is only used for iDRACs configured to use DHCP

2. Enter the *root* password for the SAH node when prompted.

Node Discovery

To discover the nodes in the cluster:

[3] This list should include the DNS server passed to the `install-director.sh` command.
[4] If you wish to use different bonding modes for different roles, delete the *BondInterfaceOptions*, and specify the bonding options per role type in the *nic-configs* files.

1. Log into the Director Node using the user name and password specified in *director.cfg*.
2. Determine the node iDRAC IP address(es) to scan on the management network.

   Note: If one or more iDRACs are configured to use DHCP, then the `ManagementAllocationPool` range specified in *network-environment.yaml* must be discovered.

   Note: If one or more iDRACs are configured to use static IP addresses, then each of the iDRACs' static IP addresses must be discovered.

   Note: The IP address(es) should only include nodes that you wish to provision. It should not include other nodes, such as the SAH.
3. Navigate to the *discover_nodes* subdirectory:

   ```
   $ cd ~/pilot/discover_nodes
   ```
4. Discover the nodes using a range of IP addresses, a space-separated list of IP addresses, or a CIDR block, the iDRAC user and the iDRAC password. For example:

   ```
   $ ./discover_nodes.py x.x.x.x-y.y.y.y -u root -p calvin \
    > ~/instackenv.json
   ```

   ```
   $ ./discover_nodes.py x.x.x.x x.x.x.y x.x.x.z -u root -p calvin \
    > ~/instackenv.json
   ```

   ```
   $ ./discover_nodes.py x.x.x.0/24 -u root -p calvin \
    > ~/instackenv.json
   ```

   a. Optional arguments include:
      - `-h`, `--help` — Display help for `discover_nodes.py`
      - `-u USERNAME`, `--u USERNAME` — Specify the iDRAC access user name. Default = *root*.
      - `-p PASSWORD`, `--p PASSWORD` — Specify the iDRAC access user password Default = *calvin*.
      - `-n {1,10,25,40,50,100}`, `--nics {1,10,25,40,50,100}` — Specify the link speed of the provisioning network interfaces in gigabits per second (Gbps). Default = *1*.
5. When discovery is complete, examine *~/instackenv.json* to verify that it contains an entry for every cluster node to be provisioned, and no other nodes:

   ```
   $ less ~/instackenv.json
   ```

   a. If `discover_nodes.py` cannot identify a provisioning network interface for a node(s), the node's entry in *~/instackenv.json* will indicate that; the value for its `mac` attribute will be `"FIXME and rerun discover_nodes"`.
      a. Ensure that the interface(s) are up and provisioned correctly.
      b. Then rerun `discover_nodes.py`.
6. Perform initial iDRACs configuration. This includes setting the appropriate NIC port to PXE boot for provisioning, as well as performing other basic iDRAC configuration.

   a. Change to the *pilot* directory:

      ```
      $ cd ~/pilot
      ```
   b. If you do not want to change the root password on any of the Overcloud nodes, and all NICs in the Overcloud nodes are in RA-compliant slots, execute the following command:

      ```
      $ ./config_idracs.py
      ```

c. If you want to change the root password on the iDRACs, or the slot that the NICs on one or more of the Overcloud nodes are in, execute the following command **for each node**:

```
$ ./config_idrac.py [-c <root_password>] [-p <fqdd_of_pxe_nic_port]] \
 <idrac_ip_or_node_service_tag>
```

Note: If the Overcloud nodes' configurations need to be changed, then the nodes will be rebooted. This can take some time.

7. Load the discovered nodes into Red Hat OpenStack Platform Director:

```
$ ~/pilot/import_nodes.py
```

8. Power off the nodes and set them to PXE boot:

```
$ ~/pilot/prep_overcloud_nodes.py
```

Launch Node Introspection

To launch introspection of the nodes:

1. Execute the following command:

```
$ ~/pilot/introspect_nodes.py
```

Out-of-band introspection is performed by default. Alternately, you can use in-band introspection by passing the `-i` argument to the `introspect_nodes.py` command.

Note: In-band node introspection can take approximately 10 minutes to complete.

Assigning Node Roles

Before deployment, nodes are assigned the roles that they will perform. This in turn causes the appropriate BIOS and RAID configuration to occur on the nodes prior to provisioning.

Table 8: RAID Mappings on page 40 lists and describes RAID setting profiles that will be created by the `assign_role` command:

Table 8: RAID Mappings

Menu Choice	BIOS/RAID Setting
OpenStack Infrastructure (Controller)	OpenStack Controller, RAID10
OpenStack Compute	OpenStack Compute, RAID10
OpenStack Storage	OpenStack Storage, RAID1 and JBOD

Note: Dell EMC PowerEdge R730xd Storage nodes are available in chassis options that support different drive configurations:

- **3.5" Drive Configuration** — Three (3) SSD journal drives, twelve (12) 3.5" OSD drives, and two (2) flex bay drives
- **2.5" Drive Configuration** — Twenty four (24) external 2.5" drives and two (2) flex bay drives

To assign the roles to the nodes in Red Hat OpenStack Platform Director:

1. Make a list of the IP addresses of the iDRAC interfaces on all nodes.
2. Decide what role each node will perform. Node roles include:

- controller
- compute

- storage

Note: You must assign the *controller* role to three (3) or more nodes.

Note: You must assign the *storage* role to three (3) or more nodes.

3. Log into the Director Node.
4. Navigate to the *~/pilot* directory:

```
cd ~/pilot
```

5. For each node, run the following command:

```
./assign_role.py <ip_mac_service_tag> <role_index>
```

a. Required arguments include:

- `ip_mac_service_tag` — IP address of the iDRAC, MAC address of the interface on the provisioning network, or service tag of the node
- `role_index` — Role that the node will play, with an optional index that indicates placement order in the rack; choices are `controller[-<index>]`, `compute[-<index>]`, and `storage[-<index>]`

Note: If the `role_index` argument is used, the `--node_placement` argument MUST be included with the `deploy-overcloud.py` command.

For example, to place a given node role on a specific server, specify the node role with an index number as follows. This example places the *controller-0* role on the server *192.168.110.125*:

```
~/pilot/assign_role.py 192.168.110.125 controller-0
```

b. Optional arguments include:

- `-h`, `--help` — Display help for `assign_role.py`
- `-m`, `--model-properties` — File that defines Dell EMC system model properties, including the FQDD of the network interface from which to PXE boot
- `-f`, `--flavor-settings` — File that contains flavor settings
- `-n`, `--node-definition` — Node definition template file that defines the node being assigned
- `-s`, `--skip-raid-config` — Do not perform RAID configuration

Note: The indices must start at 0 for each role, and must increment by 1 for each node in that role. For example, *controller-0*, *controller-1*, *controller-2*, *compute-0*, *compute-1*, etc. Node placement is performed **for all nodes or none**; if you specify an index for one node, you must specify indices for all nodes.

Note: When assigning the role of *storage* to a node, the output will display the size of the OS drive (the size of the RAID configured for the Flex Bay drives). This is an indicator to the admin that the assignment was correct.

The `assign_role.py` command can take some time to complete, since it may require a reboot.

Configuring Node Interfaces

At a high level, configuring nodes network interfaces in the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform consists of editing a small set of YAML files so that their networking parameters conform to your environment.

The following sections present the networking information you can change:

- *Configuring Networking* on page 36
- *Controller Node Bond and VLAN Parameters* on page 42
- *Compute Node Bond and VLAN Parameters* on page 42

- *Storage Node Bond and VLAN Parameters* on page 43

Controller Node Bond and VLAN Parameters

To configure Controller node bond and VLAN parameters:

1. On the Director Node, navigate to the *~/pilot/templates/nic-configs* directory.
2. Open the *controller.yaml* file in an editor.
3. Search for the term, *CHANGEME*, to go to the lines in which changes can be made.
4. Make changes, as required, to the parameters listed in *Table 9: controller.yaml File Parameters* on page 42.

Note: Each parameter has a default value, which may or may not require a change depending upon your environment.

Table 9: controller.yaml File Parameters

Parameter Name	Default Value	Description
BondInterfaceOptions	" "	Interface bonding mode. For example, `lacp=active` and/or `bond_mode=balance-slb`.
Provisioning Interface Name	em3	Name of the provisioning network interface.
bond0	em1	First interface to include in bond0.
	p3p1	Second interface to include in bond0.
bond1	em2	First interface to include in bond1.
	p3p2	Second interface to include in bond1.

The Controller NIC template assumes that the iDRACs are on the Management network. If the iDRACs are on the Provisioning network, then no routing is required; you must perform the following procedure to remove the associated route.

1. Remove the route that references the *ManagementNetCidr* parameter from the `routes:` section, indicated by the ***bold italicized*** example below:

```
routes:
 -
 ip_netmask: 169.254.169.254/32
 next_hop: {get_param: EC2MetadataIp}
 # The following route is used to route from the provisioning
 # network to the management network.  If the iDRACs are on the
 # provisioning network, then remove this route.
 -
   ip_netmask: {get_param: ManagementNetCidr}
   next_hop: {get_param: ProvisioningNetworkGateway}
```

Note: The following change should be applied **only** if deploying behind a proxy server.

Compute Node Bond and VLAN Parameters

To configure Compute node bond and VLAN parameters:

1. On the Red Hat OpenStack Platform Director node, navigate to the *~/pilot/templates/nic-configs* directory.

2. Open the *compute.yaml* file in an editor.
3. Search for the term, *CHANGEME*, to go to the lines in which changes can be made.
4. Make changes, as required, to the parameters listed in *Table 10: compute.yaml File Parameters* on page 43.

Note: Each parameter has a default value, which may or may not require a change depending upon your environment.

Table 10: compute.yaml File Parameters

Parameter Name	Default Value	Description
BondInterfaceOptions	" "	Interface bonding mode. For example, `lacp=active` and/or `bond_mode=balance-slb`.
Provisioning Interface Name	em3	Name of the provisioning network interface.
bond0	em1	First interface to include in bond0.
	p3p1	Second interface to include in bond0.
bond1	em2	First interface to include in bond1.
	p3p2	Second interface to include in bond1.

Storage Node Bond and VLAN Parameters

To configure Storage node bond and VLAN parameters:

1. On the Red Hat OpenStack Platform Director node, navigate to the *~/pilot/templates/nic-configs* directory.
2. Open the *ceph-storage.yaml* file in an editor.
3. Search for the term, *CHANGEME*, to go to the lines in which changes can be made.
4. Make changes, as required, to the parameters listed in *Table 11: ceph-storage.yaml File Parameters* on page 43.

Note: Each parameter has a default value, which may or may not require a change depending upon your environment.

Table 11: ceph-storage.yaml File Parameters

Parameter Name	Default Value	Description
BondInterfaceOptions	" "	Interface bonding mode. For example, `lacp=active` and/or `bond_mode=balance-slb`.
Provisioning Interface Name	em3	Name of the provisioning network interface.
bond0	em1	First interface to include in bond0.
	p2p1	Second interface to include in bond0.
bond1	em2	First interface to include in bond1.

Parameter Name	Default Value	Description
	p2p2	Second interface to include in bond1.

Static IP Addresses (Optional)

By default, IP addresses are assigned to the networks using DHCP. This will result in a somewhat random assignment of IP addresses to nodes. You can optionally override this default behavior by configuring static IP addresses for the Overcloud nodes on the networks.

The selected IP addresses must lie outside the allocation pools defined in *network-environment.yaml*, and must not conflict with IP addresses assigned to networking equipment or servers on the networks, such as:

- SAH
- Director Node
- Red Hat Storage Console VM
- etc.

To configure static IP addresses for the Overcloud nodes:

1. On the Red Hat OpenStack Platform Director Node, navigate to the *~/pilot/templates/* directory.
2. Open the *static-ip-environment.yaml* file in an editor.
3. Search for the term, *CHANGEME*, to go the lines where you can change the nodes' static IP addresses.
4. Make changes, as required, to the parameters listed in *Table 12: static-ip-environment.yaml File Parameters* on page 44.
5. Save the file.

Note: IP addresses are assigned to the nodes in the order listed in *Table 12: static-ip-environment.yaml File Parameters* on page 44. For example, controller0 is assigned the address 192.168.140.21 on the Private API network; controller1 is assigned 192.168.140.22, etc.

Table 12: static-ip-environment.yaml File Parameters

Parameter Name	Default Values	Description
	ControllerIPs	
tenant	192.168.130.21 192.168.130.22 192.168.130.23	Controller Node IP addresses on the Tenant network
internal_api	192.168.140.21 192.168.140.22 192.168.140.23	Controller node IP addresses on the Private API network
storage	192.168.170.21 192.168.170.22 192.168.170.23	Controller node IP addresses on the Storage network
external	192.168.190.21 192.168.190.22 192.168.190.23	Controller node IP addresses on the External network
	NovaComputeIPs	

Parameter Name	Default Values	Description
tenant	192.168.130.31 192.168.130.32 192.168.130.33	Compute node IP addresses on the Tenant network
internal_api	192.168.140.31 192.168.140.32 192.168.140.33	Compute node IP addresses on the Private API network
storage	192.168.170.31 192.168.170.32 192.168.170.33	Compute node IP addresses on the Storage network
CephStorageIPs		
storage	192.168.170.76 192.168.170.77 192.168.170.78	Red Hat Ceph Storage node IP addresses on the Storage network
storage_mgmt	192.168.180.76 192.168.180.77 192.168.180.78	Red Hat Ceph Storage node IP addresses on the Storage Clustering network

You can now proceed to *Static Virtual IP Addresses (Optional)* on page 45.

Static Virtual IP Addresses (Optional)

Virtual IP Addresses (VIPs) are used to provide IP addresses for OpenStack services on Controller nodes. Assigning a static VIP to a service enables clients to contact an OpenStack service on a Controller node by using that VIP, without knowledge of which Controller node they are communicating. This provides high availability, so that if a Controller node goes down clients can still contact the VIP. Their requests are sent to an active Controller node.

To configure static VIPs for the Controller nodes:

1. On the Red Hat OpenStack Platform Director Node, navigate to the *~/pilot/templates/* directory.
2. Open the *static-vip-environment.yaml* file in an editor.
3. Search for the term, *CHANGEME*, to go the lines where you can change the static virtual IP addresses.
4. Make changes, as required, to the parameters listed in *Table 13: static-vip-environment.yaml File Parameters* on page 45.
5. Save the file.

Table 13: static-vip-environment.yaml File Parameters

Parameter Name	Default Value	Description
redis	192.168.140.251	VIP for the `redis` service on the Private API network. This address must reside outside the `InternalApiAllocationPools` range specified in *network-environment.yaml*.

Parameter Name	Default Value	Description
ControlPlaneIP	192.168.120.250	VIP on the Provisioning network. This address must reside outside the `dhcp_start/dhcp_end range` specified in *undercloud.conf*.
InternalApiNetworkVip	192.168.140.250	VIP on the Private API network. This address must reside outside the `InternalApiAllocationPools` range specified in *network-environment.yaml*.
ExternalNetworkVip	192.168.190.250	VIP on the Public API network. This address must reside outside the `ExternalAllocationPools` range specified in *network-environment.yaml*.
StorageNetworkVip	192.168.170.250	VIP on the Storage network. This address must reside outside the `StorageAllocationPools` range specified in *network-environment.yaml*.
StorageMgmtNetworkVip	192.168.120.249	VIP on the Provisioning network. The Storage Clustering network is not connected to the Controller nodes, so the VIP for this network must be mapped to the Provisioning network. This address must reside outside the `dhcp_start/dhcp_end` range specified in *undercloud.conf*.

You can now proceed to *Configuring Dell Storage* on page 46.

Configuring Dell Storage

Dell Storage PS Series and SC Series storage servers can be integrated into the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform as backends for OpenStack Block Storage (Cinder). This topic describes the following prerequisites and required parameters:

- *PS Series Configuration (Optional)* on page 46
- *SC Series Configuration (Optional)* on page 48

PS Series Configuration (Optional)

These options apply if the Dell Storage PS Series is included in the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform as one of the Cinder backends. Requirements for Dell Storage PS Series Configuration include:

- Configure the Dell Storage PS Series according to the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Reference Architecture.

- The Dell Storage PS Series should have access to the Storage Network VLAN. The Controller nodes and the Compute nodes use the Storage Network VLAN to interact with the Dell Storage PS Series through the iSCSI OpenStack driver.

Note: For more information about the OpenStack driver configuration or post-deployment configuration, refer to the *OpenStack Volume Driver Documentation*.

To configure the Dell Storage PS Series as storage backend(s) along with Red Hat Ceph Storage, see:

- *Single PS Series Backend* on page 47
- *Multiple PS Series Backends* on page 47
- *PS Series Configuration Parameters* on page 47

Single PS Series Backend

To configure a single Dell Storage PS Series server as a storage backend along with Red Hat Ceph Storage:

1. On the Director Node, open the *~/pilot/templates/dell-cinder-backends.yaml* file in a text editor.
2. Change any applicable values listed in *Table 14: PS Series Parameters* on page 48.

Multiple PS Series Backends

To configure multiple Dell Storage PS Series servers as storage backends along with Red Hat Ceph Storage:

1. On the Director Node, open the *~/pilot/templates/dell-cinder-backends.yaml* file in a text editor.
2. Make a copy of the configuration section for each Dell Storage PS Series backend.
3. In each new section, change any applicable parameter values for the backends as listed in *Table 14: PS Series Parameters* on page 48.
   a. Assign a unique section name for each backend (e.g., *eqlx1*, *eqlx2*, etc.).
   b. Assign a unique name for the *volume_backend_name*.
   c. List the section names in the *cinder_user_enabled_backends* array with the other enabled Dell Storage PS Series backends.

The configuration example below is for two Dell Storage PS Series backend arrays:

```
...
#EQLX
#Backend1
eqlx1/volume_backend_name:
  value: eqlx1
eqlx1/volume_driver:
  value: cinder.volume.drivers.eqlx.DellEQLSanISCSIDriver
eqlx1/san_ip:
  ...
#Backend2
eqlx2/volume_backend_name:
  value: eqlx2
eqlx2/volume_driver:
  value: cinder.volume.drivers.eqlx.DellEQLSanISCSIDriver
eqlx2/san_ip:
 ...
#EQLX-END
 ...
cinder_user_enabled_backends: ['eqlx1','eqlx2',...]
```

PS Series Configuration Parameters

PS Series configuration parameters include:

Table 14: PS Series Parameters

Parameter Name	Description
eqlx_san_ip	String containing SAN IP address
eqlx_san_login	String containing SAN login ID
eqlx_san_password	String containing SAN IP password
eqlx_san_thin_provision	Boolean: if set to *true*, thin provisioning enabled
eqlx_group_name	String containing Storage Array Group Name
eqlx_pool	String containing Storage Pool Name
eqlx_use_chap	Boolean: if set to *true*, CHAP authentication enabled
eqlx_chap_login	String containing CHAP Account Name
eqlx_chap_password	String containing CHAP Account Password
cinder_user_enabled_backends	Array of strings containing enabled backends. List valid backends only

Note: For more information about using Dell Storage PS Series in a Red Hat OpenStack Platform Overcloud, see *https://access.redhat.com/documentation/en/red-hat-openstack-platform/version-8/dell-equallogic-back-end-guide/*.

SC Series Configuration (Optional)

These options apply if the Dell Storage SC Series is included in the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform as one of the Cinder backends. Requirements for Dell Storage SC Series Configuration include:

- Dell Storage Center(s) with Dell Storage Enterprise Manager Platform.
- Documents and software can be downloaded from *https://portal.compellent.com/*.
- The Dell Storage SC Series is configured with the Dell Storage Enterprise Manager Node, according to the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Reference Architecture.
- The Dell Storage Enterprise Manager Node should have access to the Storage Network VLAN and External Network. The Controller nodes will use the Storage Network VLAN to access the Dell Storage Enterprise Manager Node for management of volumes and snapshots.
- Dell Storage Center should have access to the Storage Network VLAN. The Compute nodes must have access to the Dell Storage SC Series through the Dell Storage Center iSCSI ports in order for the iSCSI driver on that node to interact with the volumes associated with Virtual Machines hosted by that node.
- Dell Storage Center Front-End Connectivity Mode should be configured to use virtual port mode.

Note: For more information about the OpenStack driver configuration or post-deployment configuration, refer to the *OpenStack Volume Driver Documentation*.

To configure the Dell Storage SC Series as storage backend(s) along with Red Hat Ceph Storage, see:

- *Single SC Series Backend* on page 48
- *Multiple SC Series Backends* on page 49
- *PS Series Configuration Parameters* on page 47

Single SC Series Backend

To configure a single Dell Storage SC Series server as a storage backend along with Red Hat Ceph Storage:

1. On the Director Node, open the *~/pilot/templates/dell-cinder-backends.yaml* file in a text editor
2. Change any applicable values listed in *Table 15: SC Series Parameters* on page 49.

Multiple SC Series Backends

To configure multiple Dell Storage SC Series servers as storage backends along with Red Hat Ceph Storage:

1. On the Director Node, open the *~/pilot/templates/dell-cinder-backends.yaml* file in a text editor.
2. Make a copy of the configuration section for each Dell Storage SC Series backend.
3. Change any applicable parameter values for the backends as listed in *Table 15: SC Series Parameters* on page 49.
   a. Assign a unique section name for each backend (e.g., *dellsc1*, *dellsc2*, etc.).
   b. Assign a unique name for the *volume_backend_name*.
   c. List the section names in the *cinder_user_enabled_backends* array with the other enabled Dell Storage SC Series backends.

The configuration example below is for two Dell Storage SC Series backend arrays:

```
        ...
        #DELLSC
        #Backend1
        dellsc1/volume_backend_name:
        value: dellsc1
        dellsc1/volume_driver:
        value:
cinder.volume.drivers.dell.dell_storagecenter_iscsi.DellStorageCenterISCSIDriver
        dellsc1/san_ip:
        ...
        #Backend2
        dellsc1/volume_backend_name:
        value: dellsc2
        dellsc1/volume_driver:
        value:
cinder.volume.drivers.dell.dell_storagecenter_iscsi.DellStorageCenterISCSIDriver
        dellsc1/san_ip:
        ...
        #DELLSC-END
        ...
        cinder_user_enabled_backends: [...,'dellsc1','dellsc1']
```

SC Series Configuration Parameters

SC Series configuration parameters include:

Table 15: SC Series Parameters

Parameter Name	Description
dellsc_san_ip	String containing Enterprise Manager IP address
dellsc_san_login	String containing Enterprise Manager login ID
dellsc_san_password	String containing Enterprise Manager password
dellsc_iscsi_ip_address	String containing virtual port iSCSI IP address
dellsc_iscsi_port	String containing iSCSI API port
dellsc_sc_ssn	String containing the Storage Center serial numbers to use
dellsc_sc_api_port	String containing Enterprise Manager API port
dellsc_server_folder	String containing server folder in which to place new server definition

Parameter Name	Description
dellsc_volume_folder	String containing volume folder in which to place new volume
cinder_user_enabled_backends	Array of strings containing enabled backends. List valid backends only

Note: For more information about using Dell Storage SC Series in a Red Hat OpenStack Platform Overcloud, see *https://access.redhat.com/documentation/en/red-hat-openstack-platform/8/dell-storage-center-back-end-guide/dell-storage-center-back-end-guide*.

Configuring Red Hat Ceph Storage

Red Hat Ceph Storage interacts with physical or logical storage units via Object Storage Devices (OSDs). This topic describes how multiple OSDs are configured across the Red Hat Ceph Storage nodes.

Some points to consider:

- The RHOSP Controllers are configured to be highly available (HA).
- The RHOSP Storage back end is configured to use Red Hat Ceph Storage via the RBD client library.
- Red Hat Ceph Storage pools will be created for RHOSP images and volumes.
- All node access will be controlled via the `cephx` authentication protocol.

Red Hat Ceph Storage Configuration Parameters

To configure Red Hat Ceph Storage parameters:

1. On the Red Hat OpenStack Platform Director node, navigate to the *~/pilot/templates/* directory.
2. Open the `dell-environment.yaml` file in an editor.
3. Search for the term, *CHANGEME*, to go to the lines in which changes can be made.
4. Make changes, as required, to the parameters listed in *Table 16: dell-environment.yaml File Parameters* on page 50.

   Note: Each parameter has a default value, which may or may not require a change depending upon your environment.

   Caution: YAML files are highly sensitive to whitespace. If you edit them, be sure to follow the files' whitespace formats precisely.

Table 16: dell-environment.yaml File Parameters

Parameter	Default Value	Description
ceph::profile::params::osds	See *OSD Configuration Parameters* on page 50.	List of drives to be used as OSDs and journals

OSD Configuration Parameters

Table 17: OSD Configuration Parameters on page 51 displays the default Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform's OSDs layout. If your configuration differs from that displayed below, modify the *osds* parameter in the `dell-environment.yaml` file according to the guidance contained within the file's comments.

Every OSD has an associated data device (where the data is stored) and a journal where write operations are staged prior to the data being committed to the data device. The data and journal may be collocated on the same physical drive, or the journal may be located on a separate drive (such as an SSD) to optimize write performance.

For the 3.5" drive configuration, the data is stored on HDDs and the journals are located on high-speed SSDs. Because the journals are relatively small (10 GB), multiple journals can be located on the same physical drive. That way, multiple OSDs share access to the same journal device. For the 2.5" drive configuration, each OSD's data and journal are collocated, and no dedicated journal devices are used.

Table 17: OSD Configuration Parameters

OSD Host	Journal Device	Data Devices
3.5" R730xd	/dev/sda	/dev/sdd, /dev/sde, /dev/sdf, /dev/sdg
	/dev/sdb	/dev/sdh, /dev/sdi, /dev/sdj, /dev/sdk
	/dev/sdc	/dev/sdl, /dev/sdm, /dev/sdn, /dev/sdo
2.5 " R730xd	N/A	/dev/sda, /dev/sdb, /dev/sdc, /dev/sdd, /dev/sde /dev/sdf, /dev/sdg, /dev/sdh, /dev/sdi, /dev/shj, /dev/sdk, /dev/sdl, /dev/sdm, /dev/sdn, /dev/sdo, /dev/sdp, /dev/sdq, /dev/sdr, /dev/sds, /dev/sdt, /dev/sdu, /dev/sdv, /dev/sdw, /dev/sdx

Caution: Any existing data on the drives will be destroyed upon Overcloud deployment.

Configuring Local Ephemeral Storage

Every VM created in OpenStack has a virtual disk that the operating system is installed upon. Ephemeral Storage refers to this OS disk and where it is stored. An ephemeral disk is created when a VM is created, and destroyed when a VM is destroyed.

By default the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform stores ephemeral disks in Red Hat Ceph Storage. You can store the ephemeral disk on the Compute Node that hosts the VM; this concept is known as Local Ephemeral Storage.

Local Ephemeral Storage Configuration Parameters

To configure the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform to use local ephemeral storage instead of Red Hat Ceph Storage:

1. Open the *~/pilot/templates/dell-environment.yaml* file in an editor.
2. Change the *NovaEnableRbdBackend* parameter from *true* to *false*.
3. Save the file.

Note: Local ephemeral storage currently supports the `images_type` values *rbd* (shared storage like Red Hat Ceph Storage) or *default* (local Compute node storage). If you choose *default*, then the Instance HA feature is not supported.

You can now proceed to *Configuring the Overcloud Domain Name* on page 51.

Configuring the Overcloud Domain Name

Note: This procedure is optional. If not configured, then a default domain name of *localdomain* will be used.

To configure the domain name for Overcloud nodes:

1. Edit the *~/pilot/templates/dell-environment.yaml* file.
2. Search for the *CloudDomain* parameter.
3. Change the value of the *CloudDomain* parameter to the domain name you want to use for the Overcloud.
4. Save the file.

You can now proceed to *Deploying the Overcloud* on page 52.

Deploying the Overcloud

Once you have completed editing the environment and nodes networking YAML configuration files, you can begin deploying the Overcloud.

To deploy the Overcloud:

1. Remove all removable media, such as USB drives.
2. Log onto the Red Hat OpenStack Platform Director node using the user name and password specified in *director.cfg*.
3. Deploy the Overcloud by executing the following command (optional arguments are enclosed in square brackets):

```
$ cd ~/pilot
$ ./deploy-overcloud.py [-h][--controllers <NUM_CONTROLLERS>] \
--computes <NUM_COMPUTES> --storage <NUM_STORAGE> \
--vlans <VLAN_RANGE> [--ntp <NTP_SERVER_FQDN>] [--timeout <MINUTES>] \
[--overcloud_name <OVERCLOUD_NAME>] [--enable_eqlx] [--enable_dellsc] \
[--disable_rbd] [--static_ips] [--static_vips] [--node_placement] \
[--debug]
```

Where:

- <VLAN_RANGE> of the format: <startingVlanNumber>:<endingVlanNumber>. For example, --vlans 201:219.
- The number of Controller nodes defaults to *3*. If you have configured a different number of Controller nodes using assign_role, then be sure to pass the --controllers option and specify the number configured. See *Assigning Node Roles* on page 40.
- If you specify *2* for the number of Compute nodes, then you must have assigned the *compute* role using the assign_role.py command to two or more nodes.
- If you specify *3* for the number of Storage nodes, then you must have assigned the *storage* role using the assign_role.py command to three or more nodes.
- You can specify an NTP time server by adding --ntp <NTP_SERVER_FQDN> to the deploy-overcloud.py command. If you do not specify a time server, then it will default to *0.centos.pool.ntp.org*.
- You can specify a deployment timeout by adding --timeout <MINUTES> to the deploy-overcloud.py command. If you do not specify a timeout, then it will default to *120 minutes*.
- You can specify the name of the Overcloud by adding --overcloud_name <OVERCLOUD_NAME> to the deploy-overcloud.py command. Node names will be prepended with the Overcloud name and a dash.
  - For example, if your Overcloud is named *rack42*, then the first Controller node will be named *rack42-controller-0*. If you do not specify an Overcloud name, then the name will default to *overcloud*. In this case, the first Controller node will be named *overcloud-controller-0*.

  Note: This enables you to distinguish Overcloud hostnames as belonging to one of multiple Overcloud deployments in the node subscription section of the Red Hat customer portal.
- You can specify whether you are using Dell Storage PS Series or SC Series as storage backends for OpenStack Block Storage (Cinder) by adding --enable_eqlx or --enable_dellsc to the deploy-overcloud.py command.
- You can disable Cinder from using Ceph Storage RBD as a backend by adding --disable_rbd to the deploy-overcloud.py command. This switch can be used in conjunction with --enable_eqlx and --enable_dellsc; if neither is used, then LVM storage will be used.
- You can specify the assignment of Overcloud nodes' static IP addresses, as described in *Static IP Addresses (Optional)* on page 44, by adding --static_ips to the deploy-overcloud.py command.

- You can specify the assignment of Overcloud nodes' static VIP addresses, as described in *Static Virtual IP Addresses (Optional)* on page 45 , by adding `--static_vips` to the `deploy-overcloud.py` command.
- You can specify node placement, as described in *Assigning Node Roles* on page 40, by adding `--node_placement` to the `deploy-overcloud.py` command.
- You can obtain debug-level logging by adding `--debug` to the `deploy-overcloud.py` command.

  Note: The `deploy-overcloud.py` command will take some time to complete, since it is installing all of the nodes in the OpenStack cluster. It can take approximately 60-120 minutes to complete depending on the options used above.

4. Once the Overcloud has been successfully deployed, use the `agent_fencing.sh` command to enable fencing, which allows the system to detect and correct problematic nodes in the Overcloud. It can also be used to disable fencing. The syntax is:

```
$ agent_fencing.sh <idrac_user> <idrac_password> < [ enable | disable ] >
```

   a. Example to enable fencing:

```
$ ~/pilot/agent_fencing.sh root calvin enable
```

   b. Example to disable fencing:

```
$ ~/pilot/agent_fencing.sh root calvin disable
```

   c. Example output from pcs status after fencing has been enabled.:

```
# [heat-admin@controller-0 ~]$ sudo pcs status
controller-1-ipmi  (stonith:fence_ipmilan):  Started controller-2
controller-0-ipmi  (stonith:fence_ipmilan):  Started controller-1
controller-2-ipmi  (stonith:fence_ipmilan):  Started controller-0
```

You can now proceed to *Scripted HA Installation* on page 53.

Scripted HA Installation

The `install-instanceHA.py` script aids the installation and configuration of Instance HA.

Note: Fencing must be enabled.

To install Instance HA via the `install-instanceHA.py` script:

1. Login into the Director Node using the user name and password specified in director.cfg.
2. Execute the following commands:

```
$ cd ~pilot
$ ./update_ssh_config.py
$ ./install-instanceHA.py
```

You can now proceed to *Registering Overcloud Nodes with CDN* on page 53.

Registering Overcloud Nodes with CDN

If your environment requires the ability to update the Overcloud nodes' software packages, you must first register them with the Red Hat Content Delivery Network (CDN). Overcloud nodes include:

- Controller Nodes
- Compute Nodes
- Red Hat Ceph Storage Nodes

The RHEL repositories listed in *Table 18: Overcloud Nodes CDN Registration Parameters* on page 54 are those necessary for updating the RPMs that are installed on the nodes, if required in your environment. Although permissible, you do not need to modify the repository values.

Registering Overcloud Nodes

To register the Overcloud nodes:

1. Edit the *~/pilot/subscription.json* file, changing any applicable values listed in *Table 18: Overcloud Nodes CDN Registration Parameters* on page 54.
   a. If you would rather not put CDN and/or proxy credentials into the JSON file, you can alternatively pass them in on the command line. See *Passing CDN and Proxy Credentials via the Command Line* on page 56.
   b. If you are using a proxy, be sure to remove the leading underscore from `"_proxy_credentials"` in the *~/pilot/subscription.json* file. See *Example subscription.json File* on page 55.

Table 18: Overcloud Nodes CDN Registration Parameters

Parameter	Value
CDN Credentials	
cdn_username	CDN username
cdn_password	CDN password
Proxy Credentials	
proxy_url	Proxy hostname:port
proxy_username	Proxy username
proxy_password	Proxy password
Roles	
control	
pool_ids	Pool ID (requires a Red Hat OpenStack Platform subscription and a Red Hat Ceph Storage subscription)
repos	RHEL repositories: • rhel-7-server-rpms • rhel-7-server-extras-rpms • rhel-7-server-rh-common-rpms • rhel-ha-for-rhel-7-server-rpms • rhel-7-server-openstack-10-rpms • rhel-7-server-openstack-10-devtools-rpms • rhel-7-server-rhceph-2-mon-rpms • rhel-7-server-rhceph-2-tools-rpms
compute	
pool_ids	Pool ID (requires a Red Hat OpenStack Platform subscription)

Parameter	**Value**
repos	RHEL repositories: • rhel-7-server-rpms • rhel-7-server-extras-rpms • rhel-7-server-rh-common-rpms • rhel-ha-for-rhel-7-server-rpms • rhel-7-server-openstack-10-rpms • rhel-7-server-openstack-10-devtools-rpms • rhel-7-server-rhceph-2-tools-rpms
ceph-storage	
pool_ids	Pool ID (requires a Red Hat OpenStack Platform subscription and a Red Hat Ceph Storage subscription)
repos	RHEL repositories: • rhel-7-server-rpms • rhel-7-server-extras-rpms • rhel-7-server-rh-common-rpms • rhel-ha-for-rhel-7-server-rpms • rhel-7-server-openstack-10-rpms • rhel-7-server-openstack-10-devtools-rpms • rhel-7-server-rhceph-2-osd-rpms

2. Navigate to the *~/pilot* directory:

```
# cd ~/pilot
```

3. Register the Overcloud nodes by executing the following command:

```
# ./register_overcloud.py
```

Example subscription.json File

```
{
    "cdn_credentials": {
        "cdn_username": "CHANGEME_username",
        "cdn_password": "CHANGEME_password"
    },
    "_comment": [ "If using a proxy, remove the leading underscore from",
                  "_proxy_credentials below and fill in the following proxy",
                  "information." ],
    "_proxy_credentials": {
        "proxy_url": "CHANGEME_hostname:CHANGEME_port",
        "proxy_username": "CHANGEME_username",
        "proxy_password": "CHANGEME_password"
    },
    "roles": {
        "control": {
            "pool_ids": [ "CHANGEME_openstack_pool_id",
                          "CHANGEME_ceph_pool_id"  ],
            "repos": [ "rhel-7-server-rpms",
                       "rhel-7-server-extras-rpms",
                       "rhel-7-server-rh-common-rpms",
                       "rhel-ha-for-rhel-7-server-rpms",
                       "rhel-7-server-openstack-10-rpms",
                       "rhel-7-server-openstack-10-devtools-rpms",
```

```
                        "rhel-7-server-rhceph-2-mon-rpms",
                        "rhel-7-server-rhceph-2-tools-rpms" ]
        },
        "compute": {
            "pool_ids": [ "CHANGEME_openstack_pool_id" ],
            "repos": [ "rhel-7-server-rpms",
                        "rhel-7-server-extras-rpms",
                        "rhel-7-server-rh-common-rpms",
                        "rhel-ha-for-rhel-7-server-rpms",
                        "rhel-7-server-openstack-10-rpms",
                        "rhel-7-server-openstack-10-devtools-rpms",
                        "rhel-7-server-rhceph-2-tools-rpms" ]
        },
        "ceph-storage": {
            "pool_ids": [ "CHANGEME_openstack_pool_id",
                           "CHANGEME_ceph_pool_id" ],
            "repos": [ "rhel-7-server-rpms",
                        "rhel-7-server-extras-rpms",
                        "rhel-7-server-rh-common-rpms",
                        "rhel-ha-for-rhel-7-server-rpms",
                        "rhel-7-server-openstack-10-rpms",
                        "rhel-7-server-openstack-10-devtools-rpms",
                        "rhel-7-server-rhceph-2-osd-rpms" ]
        }
    }
}
```

Passing CDN and Proxy Credentials via the Command Line

To pass CDN and/or proxy credentials via the command line:

1. Navigate to the ~/pilot directory:

```
# cd ~/pilot
```

2. Register the Overcloud nodes by executing the following command:

```
# ./register_overcloud.py
```

   a. Optional arguments to `register_overcloud.py` include:
      - `--cdn_username <username>`
      - `--cdn_password <password>`
      - `--proxy_url <host>:<port>`
      - `--proxy_username <username>`
      - `--proxy_password <password>`

Adding Chassis Post-Deployment

If you add additional chassis at some point after you deploy the cluster:

1. Rerun the `deploy_overcloud.py` command as specified in *Deploying the Overcloud* on page 52.

This ensures that the new chassis is recognized, and its nodes deployed correctly.

Troubleshooting Node Provisioning

If you encounter difficulties during the node provisioning process, you can find troubleshooting tips and information at:

- *https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/10/html/director_installation_and_usage/chap-troubleshooting_director_issues*

Red Hat Storage Console Configuration

This topic provides instructions for configuring the Red Hat Storage Console to integrate with Red Hat Ceph Storage in the Overcloud.

Configuring the Red Hat Storage Console

The Red Hat Storage Console is the management and monitoring service for Red Hat Ceph Storage. This section describes how to configure the Red Hat Storage Console running on the Red Hat Storage Console VM to integrate with the Red Hat Ceph Storage in the Overcloud.

Configuring the Red Hat Storage Console VM and Overcloud Ceph Storage Nodes

To configure the Red Hat Storage Console VM and Overcloud Ceph Storage nodes:

1. Log onto the Director Node as the *admin_user*, or the user as configured in *Table 4: Director Node Configuration Parameters* on page 29 and change into the pilot directory:

```
# cd ~/pilot
```

2. Execute the `config_rhscon.py` script:

```
# ./config_rhscon.py <rhscon_node_ip> <root_password>
```

   The `config_rhscon.py` script performs three functions:

   - Performs the Red Hat Storage Console server configuration as described in *Red Hat Storage Console Server Configuration*.
   - Installs the Red Hat Storage Console agent on the Overcloud nodes as described in *Installing and Configuring the Red Hat Storage Console Agent*.
   - Configures the Red Hat Storage Console server on Controller 0 as described in *Calamari Server Installation*.

3. To access the Red Hat Storage Console:

   a. Navigate to the following URL in a Web browser:

```
http://<storage_console_hostname>/skyring
```

   b. Enter the default username/password of *admin/admin*, and then click on the **Log in** button.

   See *Red Hat Storage Console Web Interface Login* for more information.

   Note: The web browser will be redirected to another port (10443), which is the actual port used by the Red Hat Storage Console.

   Note: Best practice is to change the default administrator password upon first login. See *Changing User Password* for instructions.

4. Follow the procedure described in *Importing Cluster* to import the Overcloud Red Hat Ceph Storage cluster.

   Caution: The monitor host running the Red Hat Storage Console server is on Controller 0. When selecting the monitor host, ensure that you select **Controller 0**.

Chapter 4

Overcloud Validation

Topics:

This chapter provides instructions for configuring and running the Tempest test suite to validate the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform.

Tempest Testing Configuration

To configure Tempest on the Director Node:

1. Create OpenStack networks for use by Tempest by:
   a. SSH into the Director Node as the user defined during the Red Hat OpenStack Platform Director installation process.
   b. Execute the *Creating the Networks* section of Appendix C in the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Deployment Guide.
   c. Add an additional route from the Director Node to the public network, created above in Appendix C, so Tempest can communicate with instances created by tests.

   ```
   $ sudo ip route add [public_network_ip]/24 dev [interface_name]
   ```

   For example:

   ```
   $ sudo ip route add 192.168.191.0/24 dev eth4
   ```

2. Check that the *heat_stack_owner* and *swiftoperator* roles exists in your Overcloud:

   ```
   $ source ~/overcloudrc
   $ openstack role list
   +----------------------------------+------------------+
   | ID                               | Name             |
   +----------------------------------+------------------+
   | 6226a517204846d1a26d15aae1af208f | swiftoperator    |
   | 7c7eb03955e545dd86bbfeb73692738b | heat_stack_owner |
   +----------------------------------+------------------+
   ```

   If the roles do not exist, create them:

   ```
   $ openstack role create heat_stack_owner
   $ openstack role create swiftoperator
   ```

3. Set up a *tempest* directory in your stack user's home directory and install a local version of the Tempest suite:

   ```
   $ mkdir ~/tempest
   $ cd ~/tempest
   $ /usr/share/openstack-tempest-*/tools/configure-tempest-directory
   ```

   This creates a local version of the Tempest tool set. After the Overcloud creation process completed, the director created a file named ~/tempest-deployer-input.conf. This file provides a set of Tempest configuration options relevant to your Overcloud.

4. Again SSH into the Director Node. Change directory to the *tempest* install directory:

   ```
   $ cd ~/tempest
   ```

5. Configure the Tempest instance for the environment.
   a. The following Tempest configuration script (*~/tempest/tools/config_tempest.py*) can be used to generate the required *tempest.conf* file.

Note: Be sure to use the Virtual IP address (VIP) of the Keystone endpoint. The syntax below overrides the command syntax from the Red Hat documentation.

```
$ tools/config_tempest.py --create --deployer-input ~/tempest-
deployerinput.conf service_available.swift False \
service_available.sahara False service_available.aodh
True object-storage-feature-enabled.discoverability False \
network-feature-enabled.ipv6_subnet_attributes False identity.uri
 [keystone_endpoint] identity.admin_username [os_admin_username] \
identity.admin_password [os_admin_password] identity.admin_tenant_name
 [admin_tenant_name]
```

For example:

```
$ tools/config_tempest.py --create --deployer-input ~/tempest-
deployerinput.conf service_available.swift False \
service_available.sahara False service_available.aodh
True object-storage-feature-enabled.discoverability False \
network-feature-enabled.ipv6_subnet_attributes False identity.uri
 http://10.148.44.200:5000/v2.0 identity.admin_username admin \
identity.admin_password my_password identity.admin_tenant_name admin
```

6. It is also recomemnded that you create a skip file that tells Tempest what tests you do NOT want to run. By default the `Mistral`, and `Heat Integration` plugins are installed, but are not configured. By adding both of these sets of tests to the skip file you avoid running many tests that will fail without first configuring the services and configuring Tempest for these service tests. To create a skip_file:

```
vi ~/tempest/etc/skip_file
```

and add the following lines to it and save it:

```
-heat_integrationtests\.*
-mistral_tempest_tests\.*
```

Then when exectuting a Tempest run you will add the argument *--skip-file ~/tempest/etc/skip_file* and the tests in the skip file will NOT be executed.

7. Initialize the Tempest cleanup utility by executing the following command.

Note: Prior to running Tempest you must initialize the cleanup utility, which will capture the existing state of your OpenStack deployment prior to running any tests. This will allow the cleanup utility to restore your deployment to where it was prior to running Tempest, should any failed tests leave data behind.

```
$ cd ~/tempest
$ tempest cleanup --init-saved-state
```

This will create a file named *~/tempest/saved_state.json*. You can edit the file and remove the *admin*, *demo*, and *alt_demo* tenants and users, as the cleanup utility has logic that handles these particular users and tenants.

Running Tempest Tests and Post-run Cleanup

To run Tempest tests:

1. To run the full suite of Tempest tests use following command:

```
$ tools/run-tests.sh --concurrency=4 --skip-file ~/tempest/etc/skip_file
```

> **Note:** The full Tempest test suite might take hours. Alternatively, run part of the tests using the '.*smoke' option.
>
> ```
> $ tools/run-tests.sh --concurrency=4 --skip-file ~/tempest/etc/
> skip_file '.*smoke'
> ```

Each test runs against the Overcloud, and the subsequent output displays each test and its result. You can see more information about each test in the tempest.log file generated in the same directory. For example, the output might show the following failed test:

```
{2} tempest.api.compute.servers.test_servers.
 ServersTestJSON.test_create_specify_keypair [18.305114s] ... FAILED
```

2. Once the Tempest run is complete you can use the cleanup utility to delete any objects left over by Tempest.
    a. Execute the cleanup utility in *dry-run* mode, which creates a file named *dry_run.json*. This file contains all of the objects that will be deleted when you run the cleanup utility in *standard mode*.

       ```
       $ cd ~/tempest
       $ tempest cleanup --dry-run
       ```

    b. Review dry_run.json and be sure that you intend to delete all the objects listed prior to running the tool in *standard mode*.
    c. Execute the cleanup utility in *standard mode* to permanently delete the objects contained in *dry_run.json*:

    ```
    $ cd ~/tempest
    $ tempest cleanup
    ```

    At this point your deployment should have deleted any objects left behind by Tempest.

    > **Note:** Full documentation for the cleanup utility can be found at *http://docs.openstack.org/developer/tempest/cleanup.html*.

Please refer to */usr/share/openstack-tempest-liberty/README.rpm* on your Director Node for additional information about the Tempest installation package.

In addition to this document complete documentation for Tempest can be found at *http://docs.openstack.org/developer/tempest/*.

Tempest source code and additional technical information can be found on Github at *https://github.com/redhat-openstack/tempest*.

Appendix A

File References

Topics:

- *Solution Files*

This appendix lists documents and script archives that are required to install and deploy the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform. Please contact your Dell EMC representative for copies if required.

Solution Files

Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform files include:

- *NFV-Features-DERH-RB4NFV-10.0.1.tgz* — Contains all post deployment solution scripts
- `JetPack-automation-DERH-RB4NFV-10.0.1.tgz` — Contains all automation deployment solution scripts
- *Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Dell EMC PowerEdge R-Series Architecture Guide*
- *Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Dell EMC PowerEdge R-Series Hardware Deployment Guide*
- *Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Software Deployment Guide*
- *Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform SR-IOV User Guide*
- *Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Ease of Use Guide*
- *Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Release Notes*

Appendix B

Updating RPMs on Version Locked Nodes

Topics:

- *Updating the RPMs*

At a high level, updating RPMs on a version locked node (Red Hat OpenStack Platform Director Node or Red Hat Storage Console VM):

1. Identifies the RPMs that need to be updated.
2. Removes them from the version lock list on that node.
3. Updates the RPMs.
4. Adds the updated RPMs back into the version lock list.

This topic provides detailed information to perform those steps.

Caution: Updates on NFV feature enabled OpenStack stamp is not supported.

Updating the RPMs

To update the RPMs:

Note: All of the following commands should be run as the **root** user.

1. Produce a list of RPMs that are version locked on a node:
    a. Login to a node.
    b. Execute the following command to produce a list of RPMs that are version locked:

    ```
    # yum versionlock list
    ```
2. Identify the RPMs to be updated from the output of the above command.
3. Remove the selected RPMs from the version lock list:
    a. Execute the following command, substituting *VLockListEntry* with an RPM name from the output of the `versionlock list` command above:

    Note: The *VLockListEntry* must **exactly** match an RPM name in the output of the `yum versionlock list` command.

    ```
    # yum versionlock delete VLockListEntry
    ```
    b. Repeat for each RPM.
4. Update each of the selected RPMs:
    a. Execute the following command for an RPM, substituting *RPMNameWithoutVersion* with the name of the RPM **without** the version number:

    ```
    # yum update RPMNameWithoutVersion
    ```
    b. Repeat for each subsequent RPM
5. Add each of the selected RPMs back into the version lock list:
    a. Execute the following command, again substituting *RPMNameWithoutVersion* with the name of the RPM **without** the version number:

    ```
    # yum versionlock add RPMNameWithoutVersion
    ```

Appendix C

OpenStack Operations Functional Test (Optional)

Topics:

- *Creating Neutron Networks in the Overcloud*
- *Manual RHOSP Test*
- *Scripted RHOSP Sanity Test*

This is an optional section. It includes instructions for creating the networks and testing a majority of your RHOSP environment using Glance configured with Red Hat Ceph Storage, PS Series, SC Series or any backend. These command line instructions are working examples that are found on the OpenStack website (*http://docs.openstack.org/juno/install-guide/install/apt/content/neutron_initial-tenant-network.html*).

Creating Neutron Networks in the Overcloud

The following example commands create the required tenant and public networks, and their network interfaces. You must complete them prior to creating instances and volumes, and testing of the functional operations of OpenStack.

Note: The following commands and those in the following section should be executed on the Director Node.

1. Log into the Director Node using the user name and password specified when creating the node and source the *overcloudrc* file, or the name of the stack defined when deploying the overcloud :

```
# cd ~/
# source overcloudrc
```

2. Create the tenant network by executing the following commands:

   Note: Replace *tenant_network_name* with your desired values. (e.g., `neutron net-create tenant_net --shared`).

```
# neutron net-create <tenant_network_name> --shared
```

3. Create the tenant subnet on the tenant network:

   Note: Replace *tenant_network_name*, *vlan_network*, *vlan_name* and *vlan_gateway* with your desired values (e.g., `neutron subnet-create tenant_net 192.168.201.0/24 --name tenant_201`).

```
# neutron subnet-create <tenant_network_name> <vlan_network> \
--name <vlan_name>
```

4. Create the router:

   Note: Replace *tenant_router* with your desired values (e.g., `neutron router-create tenant_201_router`).

```
# neutron router-create <tenant_router>
```

5. Before you add the tenant network interface, you will need the subnets ID. Execute the following command to display them:

```
# neutron net-list
```

   The displayed output will be similar to the following (example truncated for brevity):

```
+--------------------------------------+-------------+-------------- ...
| id                                   | name        | subnets       ...
+--------------------------------------+-------------+-------------- ...
| 52411536-ec43-402f-9736-4cabdc8c875d | tenant_net        | 7329d413 ...
| 0af01763-539e-41c7-ac32-abbaa62ee575 | HA network tenant | bdae0b72 ...
+--------------------------------------+-------------+-------------  ...
```

6. Add the tenant network interface between the router and the tenant network:

   Note: Replace *tenant_router* and *subnets_id* with your desired values (e.g., `neutron router-interface-add tenant_201_router 7329d413-ac23-56cf-8867-133b5ff8fc12`).

```
# neutron router-interface-add <tenant_router> <subnets_id>
```

7. Create the external network by executing the following commands:

Note: Replace *external_network_name* and *external_vlan_id* with your desired value. (e.g., `neutron net-create public --router:external --provider:network_type vlan --provider:physical_network physext --provider:segmentation_id 191`).

```
# neutron net-create <external_network_name> --router:external \
--provider:network_type vlan --provider:physical_network physext \
--provider:segmentation_id <external_vlan_id>
```

8. Create the external subnet with floating IP addresses on the external network:

Note: Replace *external_subnet_name*, *start_ip*, *end_ip*, *external_network_name*, *external_vlan_network* and *external_gateway* with your desired values (e.g., `neutron subnet-create --name external_sub --allocation-pool start=192.168.191.2,end=192.168.191.30 --disable-dhcp --gateway 192.168.191.254 public 192.168.191.0/24`).

```
# neutron subnet-create --name <external_subnet_name> \
--allocation-pool start=<start_ip>,end=<end_ip> \
--disable-dhcp --gateway <gateway_ip> <external_network_name>
 <external_vlan_network>
```

9. Set the external network gateway for the router:

Note: Replace *tenant_router_name* with the router name *external_nework_name* with the external network name (e.g., `neutron router-gateway-set tenant_201_router public`).

```
# neutron router-gateway-set <tenant_router_name> <external_network_name>
```

Manual RHOSP Test

This example uses the Cirros image to test high-level functional operations of OpenStack.

1. Log into the Director Node using the user name and password specified when creating the node.
2. Download the Cirros image:

```
# wget http://download.cirros-cloud.net/0.3.3/cirros-0.3.3-x86_64-disk.img
```

3. Source your Overcloud credentials:

```
# cd ~/
# source <overcloud_name>rc
```

4. Create and upload the Glance image:

```
# openstack image create --disk-format <format> \
--container-format <format> --public --file <file_path>
```

For example:

```
# openstack image create --disk-format qcow2 \
--container-format bare --public --file cirros-0.3.3-x86_64-disk.img
```

5. List available images to verify that your image was uploaded successfully:

```
# openstack image list
```

6. To view more detailed information about an image, use the identifier of the image from the output of the `openstack image list` command above:

```
# openstack image show <id>
```

7. Launch an instance using the boot image that you uploaded:

   a. Get the ID of the flavor you will use:

   ```
   # nova flavor-list
   ```

   b. Get the image ID:

   ```
   # nova image-list
   ```

   c. Get the tenant network ID:

   ```
   # nova network-list
   ```

   d. Generate a key pair. The command below generates a new key pair; if you try using an existing key pair in the command, it fails.

   Note: `MY_KEY.pem` is an output file created by the `nova keypair-add` command, and will be used later.

   ```
   # nova keypair-add <key_name> > MY_KEY.pem
   ```

   e. Create an instance using the `nova boot` command.

   Note: Change the *IDs* to your IDs from Steps 7a-c, and the *nameofinstance* and the *key_name* from Step 7c:

   ```
   # nova boot --flavor <flavor_id> --key_name <key_name> \
   --image <imageid>  --nic <net-id=<tenantNetID> <nameofinstance>
   ```

   For example:

   ```
   # nova boot --flavor 2 --key_name key_name \
   --image 0bde34f6-fba6-4174-a3ea-ff2a7918de2e \
   --nic net-id=52411536-ec43-402f-9736-4cabdc8c875d  cirros-test
   ```

   f. List the instance you created:

   ```
   # nova list
   ```

8. If you have multiple backends, create a Cinder volume type for each backend. Get the `<volume_backend_name>` from the */etc/cinder/cinder.conf* file on the Controller node.

```
 # cinder type-create <type_name>
 # cinder type-key <type_name> set
 volume_backend_name=<volume_backend_name>
```

For example:

```
# cinder type-create rbd_backend
# cinder type-key rbd_backend set volume_backend_name=tripleo_ceph

# cinder type-create eqlx_backend
# cinder type-key eqlx_backend set volume_backend_name=eqlx

# cinder type-create dellsc_backend
# cinder type-key dellsc_backend set volume_backend_name=dellsc
```

Multiple PS Series backend example:

```
# cinder type-create eqlx1_backend
# cinder type-key eqlx1_backend set volume_backend_name=eqlx1

# cinder type-create eqlx2_backend
# cinder type-key eqlx2_backend set volume_backend_name=eqlx2
```

9. Create a new volume to test the Cinder volumes:

> **Note:** If you have multiple backends defined, you must append the optional arguments `--volume-type` <*type-name*> from Step 7 to the command below.

```
# cinder create --display-name <name> <sizeinGB>
```

For example:

```
# cinder create --display-name vol_test1 1
```

a. List the Cinder volumes:

```
# cinder list
```

b. Attach the volume to the instance, specifying the server ID and the volume ID.

> **Note:** Replace the *server_id* with the ID returned from the `nova list` command, and replace the *volume_id* with the ID returned from the `cinder list` command, from the previous steps.

```
# nova volume-attach <server_id> <volume_id> <device>
```

For example:

```
# nova volume-attach 84c6e57d-a6b1-44b6-81eb-fcb36afd31b5 \
573e024d-5235-49ce-8332-be1576d323f8 /dev/vdb
```

10. Access the instance.

a. Find the active Controller by executing the following commands from the Director Node:

```
# cd ~/
# source stackrc
# nova list (make note of the controllers ips)
# ssh heat-admin@<controller ip>
# sudo -i
# pcs cluster status
```

The displayed output will be similar to the following:

```
Cluster name: tripleo_cluster
Last updated: Wed Apr 6 20:48:10 2016
Last change: Mon Apr 4 18:49:20 2016 by root via cibadmin on overcloud-
controller-1
Stack: corosync
Current DC: overcloud-controller-1 (version 1.1.13-10.el7_2.2-44eb2dd) -
 partition with quorum
3 nodes and 112 resources configured
```

b. Initiate an SSH session to the active Controller, as **heat-admin**.

c. Find the instances by executing the following command:

```
# sudo -i
# ip netns
```

The displayed output will be similar to the following:

```
qrouter-21eba0b0-b849-4083-ac40-44b794744e9f
qdhcp-f4a2c88f-1bc9-4785-b070-cc82d7c334f4
```

d. Access an instance namespace by executing the following command:

```
# ip netns exec <namespace> bash
```

For example:

```
# ip netns exec qdhcp-f4a2c88f-1bc9-4785-b070-cc82d7c334f4 bash
```

e. Verify that the namespace is the desired tenant network, by executing the following command:

```
# ip a
```

The displayed output will be similar to the following:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
19: tap05a22fb4-4f: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
 noqueue state UNKNOWN
link/ether fa:16:3e:99:b9:88 brd ff:ff:ff:ff:ff:ff
inet 192.168.201.2/24 brd 192.168.201.255 scope global tap05a22fb4-4f ->
 Tenant network
valid_lft forever preferred_lft forever
inet6 fe80::f816:3eff:fe99:b988/64 scope link
valid_lft forever preferred_lft forever
```

f. Ping the IP address of the instance.

g. SSH into the instance, as **cirros**, using the keypair generated above:

```
# ssh -i MY_KEY.pem cirros@<ip>
```

11. Format the drive and access it.

a. List storage devices:

```
# fdisk -l
```

b. Format the drive:

```
# mkfs.ext3 /dev/vdb
```

c. Mount the device, access it, and then unmount it:

```
# mkdir ~/mydrive
# mount /dev/vdb ~/mydrive
# cd ~/mydrive
# touch helloworld.txt
# ls
# umount ~/mydrive
```

Scripted RHOSP Sanity Test

As an alternative to manually testing your deployment script, we provide `sanity_test.sh`, which tests all of the basic functionality outlined in *Creating Neutron Networks in the Overcloud* on page 67 and *Manual RHOSP Test* on page 68.

To run the sanity test script:

1. Log into the Director Node using the user name and password specified when creating the node.
2. Review the `pilot/deployment-validation/sanity.ini` file, and then modify the parameters as appropriate for your environment.

   Note: The sanity test generates the public/private SSH key pair using the name specified in `sanity.ini` in the `sanity_key_name` parameter. The public key is named *~/<sanity_key_name>.pub*, and the private key is named *~/<sanity_key_name>*.
3. From your home directory, execute the `sanity_test.sh` script:

```
# cd ~/
# ./pilot/deployment-validation/sanity_test.sh
```

4. If you wish to clean the environment once the `sanity_test.sh` script has run successfully:

```
# cd ~/
# ./pilot/deployment-validation/sanity_test.sh clean
```

Appendix D

Hugepages

Topics:

This appendix details the guidelines for configuration of hugepages at the time of deployment and post-deployment of Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform version 10.0.1.

Note: Update/Upgrade is not supported for Hugepages.

Hugepages Overview

The Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform version 10.0.1 provides the ability to enable hugepage support on all Nova compute nodes in the solution.

Linux kernel measures memory as pages, where the basic unit of the physical memory is partitioned. The default size of unit page is 4096 bytes in the x86 architecture. Hugepages allow the Linux kernel to utilize the multiple page size capabilities of modern hardware architecture.

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

Note: This facility allows you to create a new Nova flavor with default values of 4 vCPUs and 4GB RAM.

Caution: The flavor created by this script uses 4GB of RAM. Ensure the number of hugepages configured comprises of 8GB of RAM to successfully launch an instance using this flavor.

Dell NFV Compute Role

Introduction

A new role named `dell nfv compute` is introduced which consists services like Hugepages and NUMA.

Hugepages and Numa can be configured by using JetPack. Hugepage number is fixed to `49152` for Hugepage size `2MB` and `96` for `1GB`.

A flag `--update_stack` is introduced to update the overcloud with change in only configuration data (e.g. hugepage size from `2MB` to `1GB` or vice versa) without having to re-deploy overcloud.

A new section is added in `settings_sample.ini` file called the `[Dell NFV Settings]`. This section includes the Hugepages and NUMA related configuration specification like `hpg_enable, hpg_size, numa_enable, numa_hostos_cpus` etc.

A new section is added in `sample.properties` file called `"is_dellnfv_compute"` which is set to `"true"`.

Note: The section for `"is_compute"` is set to `"false"` when `"is_dellnfv_compute"` is set to `"true"`.

Post deployment, refer section *Checking OpenStack Nova Hypervisor health*, to check the status of nodes with enabled `dell nfv compute` role.

Enabling Dell NFV Compute Role

For enabling Numa and Hugepage NFV features, we need to enable `Dell NFV Compute` role.

- In `sample.properties` file, `"is_dellnfv_compute"` must be set to `"true"`, and `"is_compute"` to `"false".` Please refer section *Example of sample.properties* for details.
- To configure `settings_sample.ini`, refer section *Example of settings_sample.ini.*

  1. To configure Hugepages, refer to section *Enabling Hugepages*

Deployment time - enabling and deploying Hugepages

The Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform version 10.0.1 provides the ability to enable `hugepages` support on all Nova compute nodes during deployment of the solution. This section provides the instructions to configure deployment.

Enabling Hugepages

List of HugePages Input Parameters - *(Required)*

- `hpg_enable`: (`True/False`)
- `hpg_flavor_name_list`: (`m1.tiny, m1.small, m1.medium, m1.large,m1.xlarge`, and `custom flavor`)
- `hpg_size`: Hugepages size to be that can be configured on any compute-node –choices=[`"2MB", "1GB"`] ; default = `“2MB”`

Follow the procedure provided below to enable Hugepages on Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform version 10.0.1:

1. Open an SSH session to the SAH node.
2. Log in as the root user.
3. Change the working directory to the directory containing settings file `/root/deploy/osp-deployer/settings/`

```
#cd /root/deploy/osp-deployer/settings/
```

4. Edit the `settings_sample.ini` e.g. (`acme.ini`) file. Change the settings under section `[Dell NFV Settings]` based on the desired *`hpg_size`* and *`hpg_flavor_name_list`*. Please refer section *Example of settings_sample.ini* for details. Flavor names can be existing or newly created flavors.

Only two values are supported for `hpg_size` (size of hugepages) i.e. `1GB` and `2MB`. Number of hugepages in case of `2MB` would be `49152` and for `1GB` would be `96`.

- For a 1GB Hugepage:

```
hpg_enable=true
hpg_size=1GB
hpg_flavor_name_list=<flavor_name_list>
#[any user defined flavor name list]
```

- For a 2MB Hugepage:

```
hpg_enable=true
hpp_size=2MB
hpg_flavor_name_list=<flavor_name_list>
#[any user defined flavor name list]
```

Deploy Openstack with Hugepages

After enabling hugepages in `sample_settings.ini` file, perform the following steps to deploy the hugepages in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform version 10.0.1.

1. Open an SSH session to the SAH node.
2. Ensure all hardware in the OpenStack Cluster is powered off.
3. Run the following command to enable hugepages:

```
#cd /root/deploy/osp-deployer/
#python deployer.py -s <path-of-settings-file>
```

4. Upon successful execution, success log will be generated. Please refer section *Logging* for details.
5. In case of failure during enabling of `Hugepages`; error will be logged in 2 files: `/auto_results/deployer.log<time_stamp>` of SAH node and `/pilot/overcloud_deploy_out.log` of director node.

Post deployment - enabling and deploying Hugepages

Post deployment, to change the `hugepages` configuration, perform the following steps:

1. Edit `sample_settings.ini` file and modify `hpg_size` from `2MB` to `1GB` or vice-versa.
2. Run the following command to change hugepages size:

```
#cd /root/deploy/osp-deployer/
#python deployer.py --update_stack -s <path-of-settings-file>
```

Once `hugepages` is successfully configured on ALL `dellnfv compute` nodes, the message will be displayed in the log file `deployer.log<time_stamp>` at `/auto_results/` in SAH node.

```
HugePages has been successfully configured with size <2MB/1GB>.
OverCloud deployment status: UPDATE_COMPLETE
```

If a NOVA flavor already exists by the name supplied in the script, the flavor is updated as per the script. If however such a flavor does not exist, it is created and the hugepages feature is enabled on it.

Logging

Below messages will be displayed in `deployer.log` on SAH node on *day 0* and `/pilot/overcloud-deploy_out.log` for *day N*.

1. After deployment is successfully completed on *day 0* the following messages will be displayed on SAH node `console` and `/auto_results/deployer<time_stamp>.log`:

```
HUGEPAGES has been successfully configured with hpg_size:
 <1GB/2MB>.OverCloud deployment status: CREATE_COMPLETE
```

2. After deployment is successfully completed on *day N* the following messages will be displayed on SAH node `console` and `/auto_results/deployer<time_stamp>.log`:

```
HUGEPAGES has been successfully configured with hpg_size:
 <1GB/2MB>.OverCloud deployment status: UPDATE_COMPLETE
```

3. In case the deployment fails due to either validation failure or other errors, following message is displayed:

```
Overcloud deployment skipped, please refer log for details log: /
auto_results/ or log: /pilot/overcloud_deploy_out.log of director node
```

Below is the table of log messages and actions to be taken upon encountering such errors. Apart from these errors if any other error is received, please email to openstack@dell.com.

Error log messages

Sr. No.	Error description	Custom Error Message	Action	Log location	Further Action
1	Failure in getting undercloud details from rc file	`Failed to get undercloud details from the undercloud rc file`	Log in `/pilot/overcloud_deploy_out.log`	director node	Check whether undercloud is deployed. Check `undercloud_rc` file is present in location `/home/osp_admin/`. Redeploy undercloud
2	Failure in creating aggregate	`Failed to create aggregate <aggregate name>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Redeploy or contact support team
3	Failure in setting aggregate metadata	`Failed to set aggregate metadata for aggregate with ID <id>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Redeploy or contact support team
4	Failure in adding hosts to aggregate	`Failed to add hosts <hostname> to aggregate with ID <id>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Redeploy or contact support team
5	Failure in retrieving dell nfv nodes	`Failed to get the Dell Nfv Compute nodes.`	Log in `/pilot/overcloud_deploy_out.log`	director node	Overcloud deployment must have failed or ssh connectivity to overcloud nodes not possible. Verify,troubleshoot and redeploy
6	Failure in configuring dell nfv aggregates	`Failed to configure dell nfv aggregate <aggregate name>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Ssh connection to overcloud might have failed. Redeploy or contact support team
7	If flavor is already present in openstack flavor list	`Flavor already present in flavor list`	Log in `/pilot/overcloud_deploy_out.log`	director node	Change flavor name from `/root/settings.ini` and redeploy or ignore the warning

Sr. No.	Error description	Custom Error Message	Action	Log location	Further Action
8	Failure in flavor creation	`Failed to create flavor with name <flavor name> with error <error message>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Redeploy or contact support team
9	If metadata is already added in the flavor	`Flavor metadata <metadata property> already present, skipping setting metadata.`	Log in `/pilot/overcloud_deploy_out.log`	director node	Change flavor name from `/root/settings.ini` and redeploy or ignore the warning
10	Failure in setting metadata	`Failed to set metadata <metadata name> with error <error message>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Redeploy or contact support team
11	Failure in retrieving controller nodes	`Failed to get the list of controller nodes`	Log in `/pilot/overcloud_deploy_out.log`	director node	Ssh connection to overcloud might have failed. Redeploy or contact support team
12	Failure in retrieving dell nfv compute nodes	`Failed to get the list of dell_nfv_compute nodes`	Log in `/pilot/overcloud_deploy_out.log`	director node	Ssh connection to overcloud might have failed. Redeploy or contact support team
13	Failure in retrieving dell nfv compute node uuids	`Failed to get the list of dell_nfv_compute nodes uuid`	Log in `/pilot/overcloud_deploy_out.log`	director node	Ssh connection to overcloud might have failed. Redeploy or contact support team
14	Failure in rebooting dell nfv compute nodes	`Failed to reboot dell nfv nodes with error <error message>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Ssh connection to overcloud might have failed. Redeploy or contact support team

Sr. No.	Error description	Custom Error Message	Action	Log location	Further Action
15	Ssh connection error with overcloud nodes	`Failed to establish SSH connection to remote machine with error <error message>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Ssh connection to overcloud failed. Redeploy or contact support team
16	Execute the update filter command failed	`Command execution failed with error <error message>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Ssh connection to overcloud might have failed. Redeploy or contact support team
17	Failure to restart scheduler service	`Failed to restart the scheduler service`	Log in `/pilot/overcloud_deploy_out.log`	director node	Ssh connection to overcloud might have failed. Redeploy or contact support team
18	Failure in updating filter on remote machine	`Failed to update filter on remote machine`	Log in `/pilot/overcloud_deploy_out.log`	director node	Ssh connection to overcloud might have failed. Redeploy or contact support team
19	Failure in pulling puppet log	`Failed to pull file <filename> with error <error message>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Ssh connection to overcloud might have failed. Redeploy or contact support team
20	While deleting puppet log from dellnfv nodes and no file found	`File does not exist`	Log in `/pilot/overcloud_deploy_out.log`	director node	Overcloud deployment must have failed or ssh connectivity to overcloud nodes not possible. Verify,troubleshoot and redeploy
21	Invalid flavor name	`Not a valid flavor name <flavor name>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Check in `/root/settings.ini` file for the flavor name and give valid input

Sr. No.	Error description	Custom Error Message	Action	Log location	Further Action
22	Invalid hugepage size	`Invalid huge page size <size>. Valid values are 2MB, 1GB`	Log in `/pilot/overcloud_deploy_out.log` on day N and / auto_results/ on day 0	director node on day N and SAH node on day 0	Check in `/root/settings.ini` file for the `hpg_size` and and set value as either `1GB` or `2MB`
23	Failure in modifying dellnfv_environment file	`Failed to modify the dellnfv_environment.yaml at location <path>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Check whether `dellnfv_environment.yaml` is present at location `/pilot/templates/overcloud` on director node
24	Failure in Parsing dellnfv config log file	`Failed to parse the file <filename> with error <error message>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Overcloud deployment must have failed or ssh connectivity to overcloud nodes not possible. Verify,troubleshoot and redeploy
25	Failure in deleting puppet log	`Failed to delete the file from local machine at location <path> with error <error message>`	Log in `/pilot/overcloud_deploy_out.log`	director node	Log file could not be retrieved from overcloud in the first place due to ssh error or overcloud deployment failure. Please redeploy
26	hpg_flavor_name_list is not given	`hpg_flavor_name_list is not provided.`	Log in `/pilot/overcloud_deploy_out.log` on day N and / `auto_results/` on day 0	director node on day N and SAH node on day 0	Check in `/root/settings.ini` file and provide value of `hpg_flavor_name_list` with valid data.

Checking OpenStack Nova Hypervisor health

Following the deployment, user is strongly encouraged to check the health of the hypervisors across the pool of compute nodes. Run the following command on the controller node to list all compute node hypervisors.

```
$ source <overcloud-rc>
```

```
$ nova hypervisor-list
```

In the report displayed, note the state of the hypervisors. An "up" state indicates that the hypervisors are in good health. Also note that the status should read "enabled" for all compute nodes.

```
+----+------------------------+-------+---------+
| ID | Hypervisor hostname    | State | Status  |
+----+------------------------+-------+---------+
| 1  | dell-nfv-compute-node0 | up    | enabled |
| 2  | dell-nfv-compute-node1 | up    | enabled |
| 3  | dell-nfv-compute-node2 | up    | enabled |
+----+------------------------+-------+---------+
```

Example of settings_sample.ini snippet for Hugepage and Numa

```
#########################################################################
#                                                                       #
#Add below sections in the settings.ini file for your own stamp.        #
#                                                                       #
#########################################################################

[Cluster Settings]
# Bonding options configuration by node type
dellnfv_compute_bond_opts=802.3ad miimon=100

# Interfaces per node type
dellnfv_compute_bond0_interfaces=em1 p1p1
dellnfv_compute_bond1_interfaces=em2 p2p2
dellnfv_compute_provisioning_interface=em3

[Dell NFV Settings]
#Provide NFV features here.

#Enter value of enable_hpg as True/False for HugePages
hpg_enable=False
#User should give this parameter in same format.
#Supported values for hpg_size(Size of hugepages) is 2MB and 1 GB.
#For 2MB (# of hugepages), hpg_num is 49152
#For 1GB (# of hugepages), hpg_num is 96
hpg_size=2MB
#Only use of  Uppercase and Lowercase alphabets, digits, '.','-','_' are
 allowed.
#Comma separated values of flavor_names in list
hpg_flavor_name_list=m1.hpg,m1.small,m1.large

#Enter value of enable_numa as True/False for NUMA
numa_enable=True
#User should give this parameter in same format.
#numa_hostos_cpus should be enumerated list of cpus to be used for host eg.
 0,1,2,3,4,5,6,7
numa_hostos_cpus=0-7
```

```
#Only use of  Uppercase and Lowercase alphabets, digits, '.','-','_' are
 allowed.
#Comma separated values of flavor_names in list
numa_flavor_name_list=m1.numa,m1.large,m1.xlarge
```

Stamp-specific Properties File changes for Numa and Hugepage

```
    {
        "is_compute": "false",
        "idrac_ip": "192.168.110.31",

        "private_api_ip": "192.168.140.31",
        "storage_ip": "192.168.170.31",
        "tenant_tunnel_ip": "192.168.130.31"

    },
    {
        "is_compute": "false",
        "service_tag": "DEFUVW",

        "private_api_ip": "192.168.140.32",
        "storage_ip": "192.168.170.32",
        "tenant_tunnel_ip": "192.168.130.32"

    },
    {
        "is_compute": "false",
        "idrac_ip": "192.168.110.33",

        "private_api_ip": "192.168.140.33",
        "storage_ip": "192.168.170.33",
        "tenant_tunnel_ip": "192.168.130.33"

    },
    {
        "is_dellnfv_compute": "true",
        "idrac_ip": "192.168.110.41",

        "private_api_ip": "192.168.140.41",
        "storage_ip": "192.168.170.41",
        "tenant_tunnel_ip": "192.168.130.41"

    },
    {
        "is_dellnfv_compute": "true",
        "service_tag": "JKLOPQ",

        "private_api_ip": "192.168.140.42",
        "storage_ip": "192.168.170.42",
        "tenant_tunnel_ip": "192.168.130.42"

    },
    {
        "is_dellnfv_compute": "true",
        "idrac_ip": "192.168.110.43",

        "private_api_ip": "192.168.140.43",
        "storage_ip": "192.168.170.43",
        "tenant_tunnel_ip": "192.168.130.43"
```

```
    }
```

Appendix E

NUMA

Topics:

- *NUMA overview*
- *Enabling Dell NFV Compute Role*
- *Deployment time - enabling and deploying NUMA*
- *Making changes to NUMA settings in a flavor*
- *Logging*
- *Checking OpenStack Nova Hypervisor health*

This appendix details the guidelines for configuration of NUMA during and after the deployment of Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform version 10.0.1.

Note: Update/Upgrade is not supported for NUMA.

NUMA overview

The Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform version 10.0.1 provides the ability to enable NUMA optimization and CPU pinning support on all Nova compute nodes in the solution.

Non-uniform memory access or NUMA allows multiple CPUs to share local memory, which improves performance due to improved memory access times.

This document provides the instructions to configure this feature at the time of deployment or post-deployment.

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

Note: This facility allows you to create a new Nova flavor with default values of 4 vCPUs and 4GB RAM.

Enabling Dell NFV Compute Role

For enabling Numa and Hugepage NFV features, we need to enable `Dell NFV Compute` role.

- In `sample.properties` file, `"is_dellnfv_compute"` must be set to `"true"`, and `"is_compute"` to `"false"`. Please refer section *Example of sample.properties* for details.
- To configure `settings_sample.ini`, refer section *Example of settings_sample.ini.*

  1. To configure NUMA, refer to section *Enabling NUMA*

Deployment time - enabling and deploying NUMA

The Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform version 10.0.1 provides the ability to enable Linux NUMA and CPU pinning support on all Nova compute nodes during deployment of the solution. This section provides the instructions to configure Deployment.

Enabling NUMA

List of NUMA Input Parameters - *(Required)*

- `numa_enable`: (`True/False`)
- `numa_flavor_name_list`: (`m1.tiny, m1.small, m1.medium, m1.large,m1.xlarge, and custom flavor`)
- `numa_hostos_cpus`: `0-7` or `0,1,2,3,4,5,6,7` (default : `0,1,2,3,4,5,6,7`)

Follow the procedure provided below to enable NUMA optimization and CPU pinning on Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform version 10.0.1:

1. Open an SSH session to the SAH node.
2. Log in as the root user.
3. Change the working directory to `/root/deploy/osp-deployer/settings/`

```
#cd /root/deploy/osp-deployer/settings/
```

4. Edit the `settings_sample.ini` (e.g. `acme.ini`) file. Change the settings under section `[Dell NFV Settings]` based on the desired *`numa_hostos_cpus`* and *`numa_flavor_name_list`*. Please

refer section *Example of settings_sample.ini* for details. Flavor names can be existing or newly created flavors.

```
numa_enable=true
numa_flavor_name_list=<flavor_name_list>
numa_hostos_cpus=0,1,2,3,4,5,6,7
```

Deploying NUMA

After enabling NUMA in `sample_settings.ini` file, perform the following steps to deploy NUMA in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform.

1. Open an SSH session to the SAH node.
2. Ensure all hardware in the OpenStack Cluster is powered off.
3. Post deployment, run the following command to enable NUMA:

```
# cd /root/deploy/osp-deployer/
# python deployer.py -s <path-of-settings-file>
```

4. Upon successful execution, success log will be generated. Please refer section *Logging* for details.
5. In case of failure during enabling of `NUMA`; error will be logged in 2 files: `/auto_results/deployer.log<time_stamp>` of SAH node and `/pilot/overcloud_deploy_out.log` of director node.

Making changes to NUMA settings in a flavor

Note: The NUMA optimization and CPU pinning facility is supported in the numa-enabled flavors created during deployment. This is part of the deployment automation now. No further actions need to be taken to set CPU Pinning in flavors.

Logging

Below messages will be displayed in `deployer.log` on SAH node on *day 0* and `/pilot/overcloud-deploy_out.log` for *day N*.

1. After deployment is successfully completed on *day 0* the following messages will be displayed on SAH node `console` and `/auto_results/deployer<time_stamp>.log`:

```
NUMA has been successfully configured with hostos_cpus: <0-7 or
 0,1,2,3,4,5,6,7>.OverCloud deployment status: CREATE_COMPLETE
```

2. After deployment is successfully completed on *day N* the following messages will be displayed on SAH node `console` and `/auto_results/deployer<time_stamp>.log`:

```
NUMA has been successfully configured with hostos_cpus: <0-7 or
 0,1,2,3,4,5,6,7>.OverCloud deployment status: UPDATE_COMPLETE
```

3. In case the deployment fails due to either validation failure or other errors following message is displayed:

```
Overcloud deployment skipped, please refer log for details log: /
auto_results/ or log: /pilot/overcloud_deploy_out.log of director node
```

Below is the table of log messages and actions to be taken upon encountering such errors. Apart from these errors if any other error is received, please email to openstack@dell.com.

Error log messages

Sr. No.	Error description	Custom Error Message	Action	Log location	Further Action
1	numa_flavor_name _list is not given	`numa_flavor_name_list is not provided`	Log in `/pilot/overcloud_deploy_out.log` on day N and `/auto_results/` on day 0	director node on day N and SAH node on day 0	Check in `/root/settings.ini` file and provide value of `numa_flavor_name_list` with valid data.

Checking OpenStack Nova Hypervisor health

Following the deployment, user is strongly encouraged to check the health of the hypervisors across the pool of compute nodes. Run the following command on the controller node to list all compute node hypervisors.

```
$ source <overcloud-rc>
```

```
$ nova hypervisor-list
```

In the report displayed, note the state of the hypervisors. An "up" state indicates that the hypervisors are in good health. Also note that the status should read "enabled" for all compute nodes.

```
+----+------------------------+-------+---------+
| ID | Hypervisor hostname    | State | Status  |
+----+------------------------+-------+---------+
| 1  | dell-nfv-compute-node0 | up    | enabled |
| 2  | dell-nfv-compute-node1 | up    | enabled |
| 3  | dell-nfv-compute-node2 | up    | enabled |
+----+------------------------+-------+---------+
```

Appendix F

OVS-DPDK

Topics:

This appendix details the guidelines for configuration of OVS-DPDK at the time of deployment (DTE) of Dell EMC Red Hat OpenStack NFV Solution.

Note: OVS-DPDK has only been validated on Dell EMC servers with Intel Ethernet Converged Network Adpater x520 NICs.

Note: Update/Upgrade is not supported for OVS-DPDK.

OvS

Open vSwitch (OvS) is a multilayer software/virtual switch used to interconnect virtual machines in the same host and between different hosts.

Open vSwitch makes use of kernel for packet forwarding through a data path known as “fastpath” which consists of a simple flow table with action rules for the received packets. Exception packets or packets with no corresponding forwarding rule in the flow table are sent to the user space (slowpath) first. Switching between two memory spaces creates a lot of overheard, thus making the user space “slowpath”. User space takes a decision and updates the flow table in the kernel space accordingly so they can make use of fastpath in future.

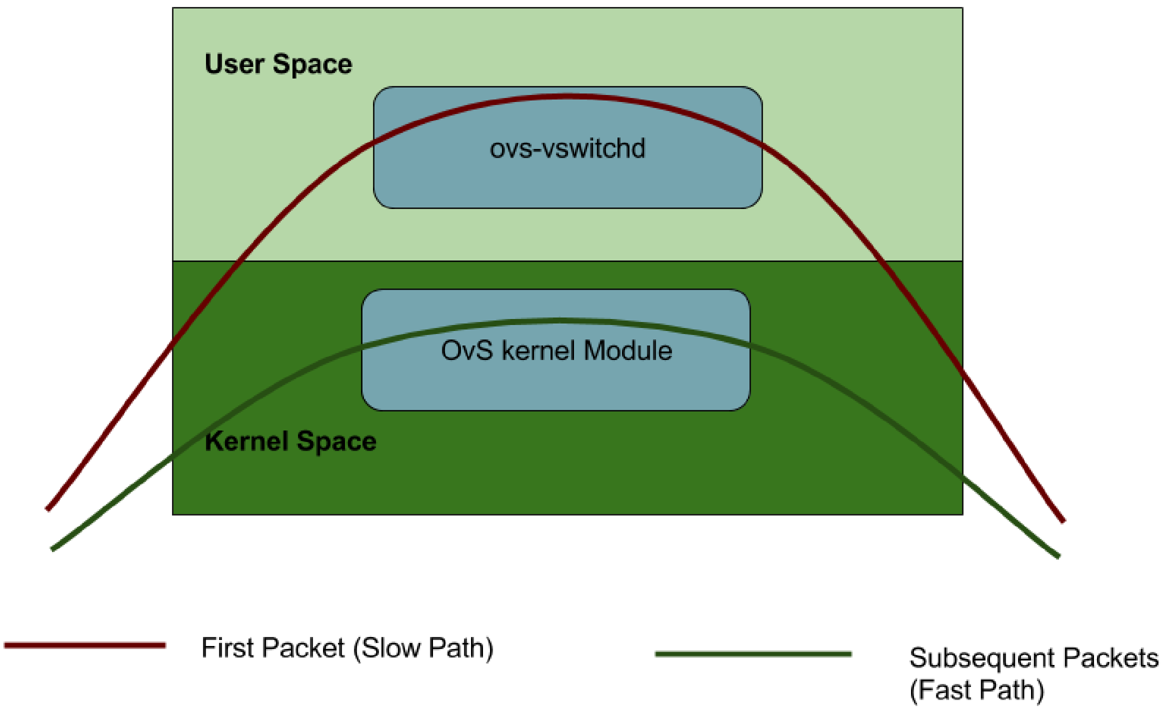

Figure 2: OvS Architecture

As can be seen in the Figure 1, OvS kernel module acts as a cache for the user space. And just like a cache, its performance decreases as the number of rules increase in the user space.

DPDK

The Data Plane Development Kit (DPDK) is a set of data plane libraries and network interface controller drivers for fast packet processing from Intel. DPDK runs inside user space, and gives applications direct access to raw traffic from NIC, completely bypassing kernel and kernel IP Stack. It further utilizes poll mode drivers (PMDs) and Hugepages to increase network performance.

OVS-DPDK

With the help of DPDK, all the OvS processes are moved to user space. Since all the paths are in single space, it removes the bottleneck created by continuous switching between two spaces.

OVS-DPDK in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform

OVS-DDPK is enabled at the deployment time as part of the automation of RHOSP deployment. There is no post deployment enablement support in this release.

If enabled, OVD-DPDK must be so on all compute nodes.

Note: OVS-DPDK deployment is done only on standard 'Compute' role. Dell NFV Compute role is not enabled.

OVS-DPDK requires an extra network bond; the already existing bonds (bond0 and bond1) carry storage network, OpenStack tunnelled tenant network and internal API network. The OVS-DPDK bond will carry OpenStack VLAN tenant networks.

There are two modes in which OVS-DPDK can be enabled, in the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform, each with its own pros and cons. The two modes are:

1. All four ports for OVS-DPDK bond, SR-IOV not enabled.
2. Two ports for OVS-DPDK bond, the remaining to be used for SR-IOV.

Note: Hugepages and NUMA Awareness scripts do not work with OVS-DPDK as both needs to be enabled for OVS-DPDK and they are handled by OVS-DPDK enablement.

Before You Begin

In this guide, it is assumed that the user has complete knowledge about the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Version 10.0.1. This includes:

1. Knowledge about different nodes in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform version 10.0.1, like SAH, Director, Controller, Compute and Ceph-storage as explained in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Version 10.0.1 Hardware Deployment Guide.
2. Hardware configurations including switch configurations as explained in Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Version 10.0.1 Hardware Deployment Guide.
3. Automation scripts along with Settings and properties files required for deploying the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform version 10.0.1 are explained in this Guide.

Prerequisites

Before starting the deployment of Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Version 10.0.1 with OVS-DPDK, following prerequisites must be met:

1. **Extra NICs:**

   Two extra Intel x520 NICs in the compute nodes are attached. For DELL PowerEdge R630, OVS-DPDK NICs are plugged in PCI Slot 2 & 3 and for DELL PowerEdge R730 OVS-DPDK NICs are plugged in

PCI Slot 4 & 5. NDC (Network Daughter Card) and PCI Slot 1 NICs are used for deployment (bond0 and bond1).

2. **Cabling and switch configuration:**

To setup the switch configurations, connect the OVS-DPDK NICs to leaf switches. The connections need to be such that high availability (HA) is ensured. This means that one NIC is connected to leaf switch 1, and other to leaf switch 2. The bond is created over two NICs to ensure that network is up even when one NIC or a leaf switch goes down.

Switch configurations depend on the OVS-DPDK mode. With simple OvS, tenant networks are carried over bond0. In this case, assume that the VLAN range for tenant networks is 201-250.

Note: For successfully enabling OVS-DPDK, the following section provides guidance on the configuration changes done to the Dell EMC Networking Switch S6010. For guidance on other Dell EMC Networking Switch models or for implementation assistance please contact Dell EMC sales representatives.

Mode I, Tenant Networks Only:

For this scenario, switch configuration will have three major changes:

a. Creation of new port channels for the new ports that will be used for OVS-DPDK bond. In this mode, LACP will be enabled on 4 ports.

b. Removal of VLANs 201-250 from bond0. This means removing the tagged port channel from these VLANs that correspond with the compute nodes' bond0.

c. Addition of new tagged port channels to these VLANs.

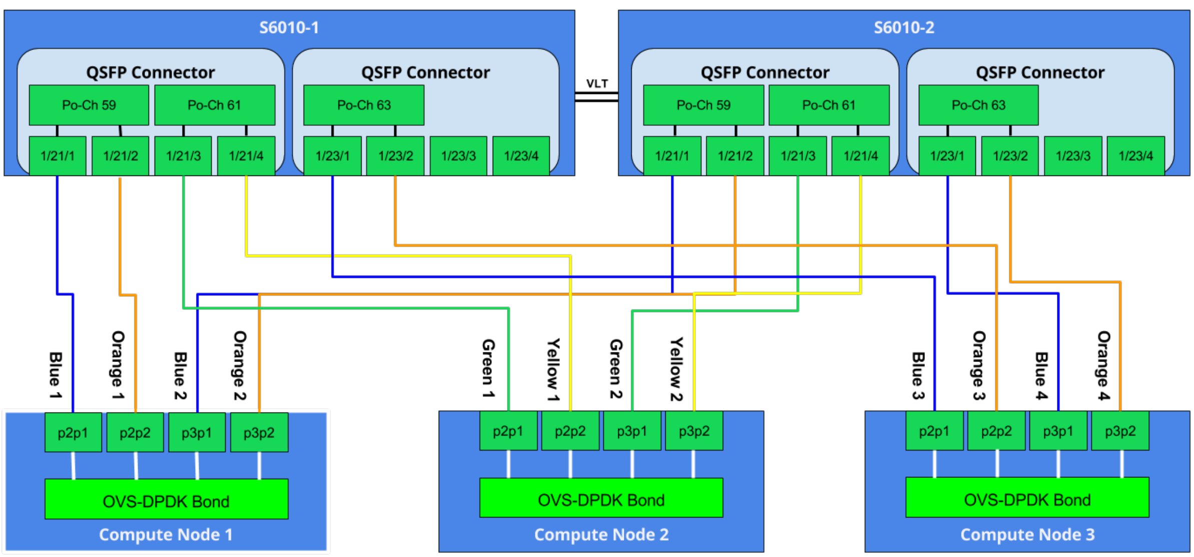

Figure 3: OVS-DPDK Mode 1 reference wiring diagram

Assuming that one of the compute node's PCI Slot 2 is connected to Leaf-1 switch's interfaces 1/21/1 and 1/21/2 and PCI Slot 3 to Leaf-2 switch's interfaces 1/21/1 and 1/21/2, The configuration for these interfaces on both switches will be:

```
!
interface TenGigabitEthernet 1/21/1
  no ip address
!
port-channel-protocol LACP
  port-channel 59 mode active
```

```
  no shutdown
  !
  interface TenGigabitEthernet 1/21/2
    no ip address
  !
  port-channel-protocol LACP
    port-channel 59 mode active
  no shutdown
```

Note: Two interfaces are configured per port channel on each switch.

Also the port channel will have the following configuration on both switches:

```
  !
  interface Port-channel 59
   no ip address
   switchport
   vlt-peer-lag port-channel 59
   no shutdown
```

Mode II, Tenant and External Networks:

For this scenario, switch configuration will have three major changes:

a. Creation of new port channels for the new ports that will be used for OVS-DPDK bond.

b. Removal of VLANs 201-250 from bond0. This means removing the tagged port channel from these VLANs that correspond with the compute nodes' bond0.

c. Addition of new tagged port channels to these VLANs.

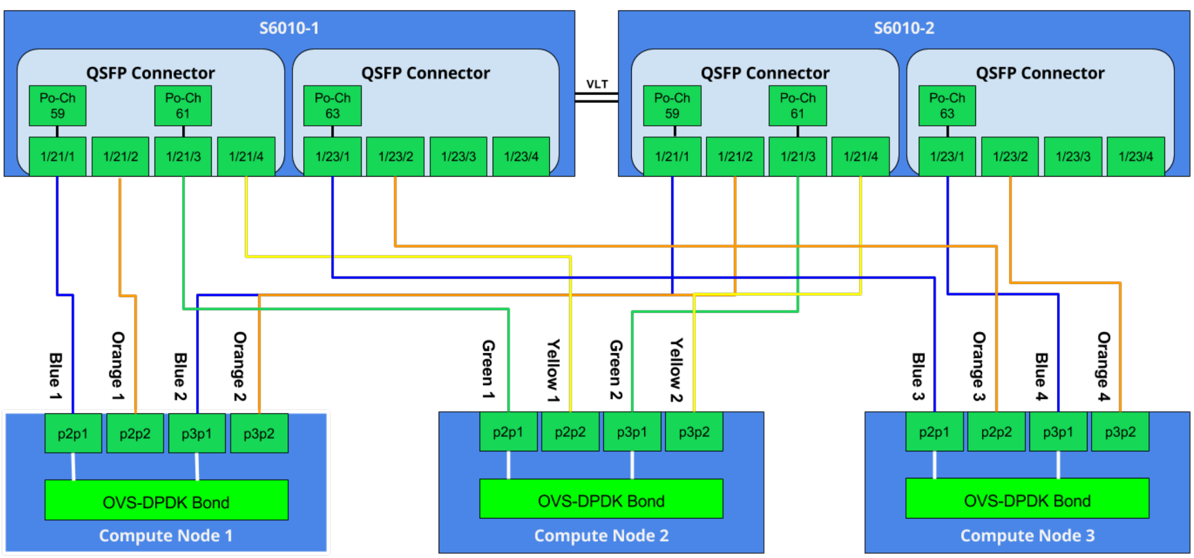

Figure 4: OVS-DPDK Mode 2 reference wiring diagram

Assuming that one of the compute node’s PCI Slot 2 is connected to Leaf-1 switch’s interfaces 1/21/1 and 1/21/2 and PCI Slot 3 to Leaf-2 switch’s interfaces 1/21/1 and 1/21/2, The configuration for these interfaces on both switches will be:

```
!
interface TenGigabitEthernet 1/21/1
 no ip address
!
 port-channel-protocol LACP
```

```
   port-channel 59 mode active
  no shutdown
 !
```

Note: Only one interface is configured per port channel on each switch.

Also the port channel will have the following configuration on both switches:

```
 !
 interface Port-channel 59
  no ip address
  switchport
  vlt-peer-lag port-channel 59
  no shutdown
```

3. **Settings file parameters**

To enable OVS-DPDK in the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform Version 10.0.1, following changes needs to be made in the settings file.

a. Open an SSH terminal as "root" user, to the SAH Node.
b. Open the settings file, and find "ovs_dpdk_enable". Assuming the name of settings file is "acme.ini" and it is present in the "/root" directory:

```
vi /root/acme.ini
```

All the parameters that are related to enabling OVS-DPDK are described in the table below.

Parameter	Value	Description
ovs_dpdk_enable	"None" | "Tenant-networks-only" | "Tenant-and-External-networks"	"None" means OVS-DPDK will not be enabled i.e. simple OvS is used. The other two options enable OVS-DPDK and set the mode accordingly. "Tenant-networks-only" will select Mode I (4 ports) while "Tenant-and-External-networks" will select Mode II (2 ports).
ovs_dpdk_role_list	"Compute"	A comma separated list of all the roles on which OVS-DPDK is to be enabled. **Currently only supported value is "Compute"**.
ovs_dpdk_policy	"Balanced"	Sets the cores distribution policy to be used. **Currently only supported value is "Balanced"**. To learn more about policies read the *policies* section.

- **Example mode I:**

  Settings for enabling OVS-DPDK in mode I are following:

```
    ovs_dpdk_enable="Tenant-networks-only"
    ovs_dpdk_role_list="Compute"
    ovs_dpdk_policy="Balanced"
```

- **Example mode II:**

Settings for enabling OVS-DPDK in mode II are following:

```
ovs_dpdk_enable="Tenant-and-External-networks"
ovs_dpdk_role_list="Compute"
ovs_dpdk_policy="Balanced"
```

Start Deployment

Change the directory to /root/JetPack/src/deploy/osp_deployer.

```
# cd /root/JetPack/src/deploy/osp_deployer
```

Start the “deployer.py” script execution and pass the settings file using “-s” parameter.

```
# python deployer.py –s /root/acme.ini
```

Monitor the output of the script. For detailed logs open a new SSH terminal and tail the logs file in “/auto_results/” directory. The log files are timestamped.

Success

Output for successful deployment looks like this.

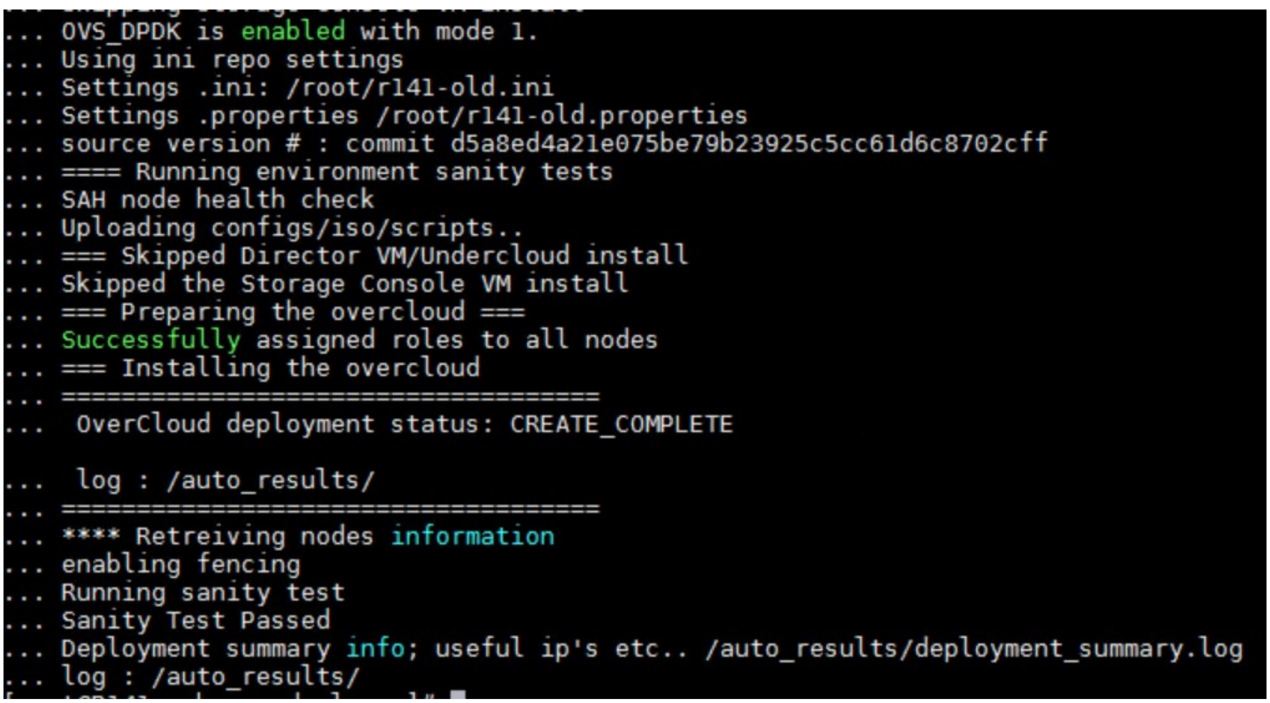

Figure 5: Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform with OVS-DPDK successful deployment

At the start of deployment a log message shows whether OVS-DPDK is enabled or not. For a successful deployment, overcloud deployment status should be CREATE_COMPLETE. And if sanity test was set to true in the settings file, then it should pass.

Failure

There are three types of deployment failures that can occur:

1. At the start of deployment, failures related to settings and properties file input validation may occur.
2. At the time of overcloud deployment preparations, in this part OVS-DPDK related parameters are configured.
3. During the deployment, failure can be due to multiple reasons: a subset of these failures is related to “FAILED_OVERCLOUD”. The OVS-DPDK related failure are part of this subset. The most likely reasons for these failures are following:
    - Switch configurations.
    - Hardware configuration like placing the extra NICs in the right PCI Slots.

Screenshots and solutions of common failure are given below.

Start of deployment

Following errors can occur at the start of deployment i.e. during input validation.

Unsupported value of ovs_dpdk_enable:

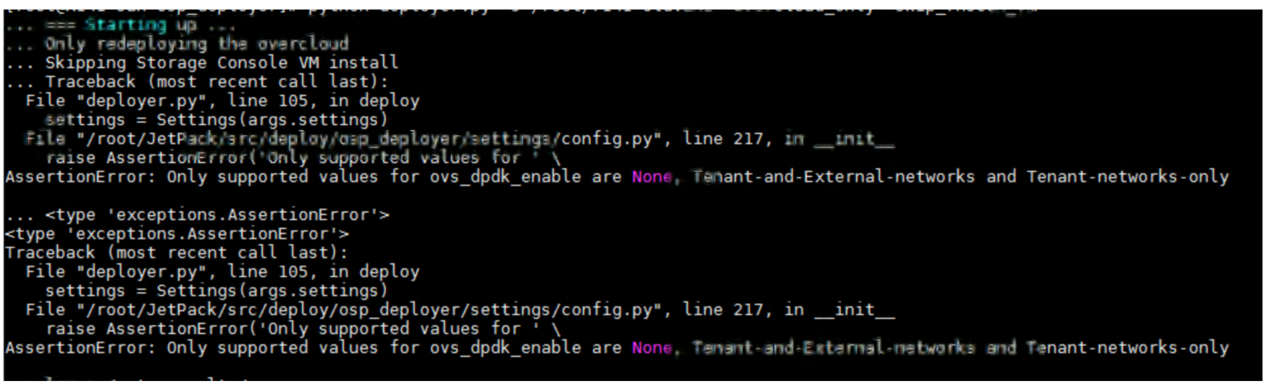

Figure 6: Unsupported value for ovs_dpdk_enable in the settings file

Change the value in the settings to a supported one and deploy again.

Unsupported value of ovs_dpdk_role_list:

```
... === Starting up ...
... Only redeploying the overcloud
... Skipping Storage Console VM install
... Traceback (most recent call last):
  File "deployer.py", line 105, in deploy
    settings = Settings(args.settings)
  File "/root/JetPack/src/deploy/osp_deployer/settings/config.py", line 231, in __init__
    raise AssertionError('Only supported value for ' \
AssertionError: Only supported value for ovs_dpdk_role_list is Compute

... <type 'exceptions.AssertionError'>
<type 'exceptions.AssertionError'>
Traceback (most recent call last):
  File "deployer.py", line 105, in deploy
    settings = Settings(args.settings)
  File "/root/JetPack/src/deploy/osp_deployer/settings/config.py", line 231, in __init__
    raise AssertionError('Only supported value for ' \
AssertionError: Only supported value for ovs_dpdk_role_list is Compute
```

Figure 7: Unsupported value for ovs_dpdk_role_list in the settings file

Change the value in the settings to a supported one and deploy again.

Unsupported value of ovs_dpdk_policy:

```
... === Starting up ...
... Only redeploying the overcloud
... Skipping Storage Console VM install
... Traceback (most recent call last):
  File "deployer.py", line 105, in deploy
    settings = Settings(args.settings)
  File "/root/JetPack/src/deploy/osp_deployer/settings/config.py", line 226, in __init__
    raise AssertionError('Only supported value for ' \
AssertionError: Only supported value for ovs_dpdk_policy is Balanced

... <type 'exceptions.AssertionError'>
<type 'exceptions.AssertionError'>
Traceback (most recent call last):
  File "deployer.py", line 105, in deploy
    settings = Settings(args.settings)
  File "/root/JetPack/src/deploy/osp_deployer/settings/config.py", line 226, in __init__
    raise AssertionError('Only supported value for ' \
AssertionError: Only supported value for ovs_dpdk_policy is Balanced
```

Figure 8: Unsupported value for ovs_dpdk_policy in the settings file

Change the value in the settings to a supported one and deploy again.

Unsupported type of hardware:

```
... === Starting up ...
... Only redeploying the overcloud
... Skipping Storage Console VM install
... Traceback (most recent call last):
  File "deployer.py", line 105, in deploy
    settings = Settings(args.settings)
  File "/root/JetPack/src/deploy/osp_deployer/settings/config.py", line 247, in __init__
    raise AssertionError('OVS-DPDK is not supported with FX ' \
AssertionError: OVS-DPDK is not supported with FX series hardware.

... <type 'exceptions.AssertionError'>
<type 'exceptions.AssertionError'>
Traceback (most recent call last):
  File "deployer.py", line 105, in deploy
    settings = Settings(args.settings)
  File "/root/JetPack/src/deploy/osp_deployer/settings/config.py", line 247, in __init__
    raise AssertionError('OVS-DPDK is not supported with FX ' \
AssertionError: OVS-DPDK is not supported with FX series hardware.

... log : /auto_results/
```

Figure 9: Unsupported type of hardware

OVS-DPDK is only supported on "poweredge" hardware.

Using unsupported NICs for compute bonds 0 and 1

```
... === Starting up ...
... Only redeploying the overcloud
... Skipping Storage Console VM install
... Traceback (most recent call last):
  File "deployer.py", line 105, in deploy
    settings = Settings(args.settings)
  File "/root/JetPack/src/deploy/osp_deployer/settings/config.py", line 242, in __init__
    raise AssertionError('With OVS-DPDK following compute node ' \
AssertionError: With OVS-DPDK following compute node NICs needs to be used for deployment: em1 em2 p1p1 and p1p2.

... <type 'exceptions.AssertionError'>
<type 'exceptions.AssertionError'>
Traceback (most recent call last):
  File "deployer.py", line 105, in deploy
    settings = Settings(args.settings)
  File "/root/JetPack/src/deploy/osp_deployer/settings/config.py", line 242, in __init__
    raise AssertionError('With OVS-DPDK following compute node ' \
AssertionError: With OVS-DPDK following compute node NICs needs to be used for deployment: em1 em2 p1p1 and p1p2.

... log : /auto_results/
```

Figure 10: Using unsupported NICs for compute nodes bonds 0 and 1

NDC and PCI slot 1 needs to be used for compute nodes' bonds 0 and 1.

OVS-DPDK Environment Setup

Following errors can occur during the OVS-DPDK environment setup.

Compute nodes having different models

```
... ==== Running environment sanity tests
... SAH node health check
... Uploading configs/iso/scripts..
... === Skipped Director VM/Undercloud install
... Skipped the Storage Console VM install
... === Preparing the overcloud ===
... Successfully assigned roles to all nodes
... Traceback (most recent call last):
  File "deployer.py", line 203, in deploy
    director_vm.setup_templates()
  File "/root/JetPack/src/deploy/osp_deployer/director.py", line 437, in setup_templates
    self.setup_networking()
  File "/root/JetPack/src/deploy/osp_deployer/director.py", line 994, in setup_networking
    "error " + setup_output)
AssertionError: OVS-DPDK environment setup failed with error ('ERROR: Multiple server models found for compute nodes. All compu
te nodes must have same server models.\r\n', '', 1)

... <type 'exceptions.AssertionError'>
<type 'exceptions.AssertionError'>
Traceback (most recent call last):
  File "deployer.py", line 203, in deploy
    director_vm.setup_templates()
  File "/root/JetPack/src/deploy/osp_deployer/director.py", line 437, in setup_templates
    self.setup_networking()
  File "/root/JetPack/src/deploy/osp_deployer/director.py", line 994, in setup_networking
    "error " + setup_output)
AssertionError: OVS-DPDK environment setup failed with error ('ERROR: Multiple server models found for compute nodes. All compu
te nodes must have same server models.\r\n', '', 1)
```

Figure 11: Having multiple types of compute nodes

All computes nodes must have the same server model.

Miscellaneous Errors

OVS-DPDK environment setup can fail if Ironic client is not available, or if the required server model information is not present. Check if the ironic service is working properly on the director nodes. Also check the details of ironic nodes to make sure they have the required server model information.

During overcloud deployment

In this case deployment fails with overcloud deployment status as "CREATE_FAILED". These errors can be due to a number of reasons. Following commands can be used to debug the issue. All of these commands are to be run on the director node.

Use the following command to get the stack name.

```
$ openstack stack list
```

List all the resources of the stack and using grep, filter out the resources which are created successfully and list only failed or in progress resources.

```
$ openstack stack resource list <stack-name> | grep -v COMPLETE
```

This will provide a general idea of which resource creation failed.

The following command can be used to list all the software deployments, which are either in "IN PROGRESS" or "FAILED" state.

```
$ openstack software deployment list | grep -v COMPLETE
```

To further filter out which server has failed or in progress, software deployment server option can be given. First list all the servers and obtain the UUID of the servers.

```
$ openstack server list
```

Use the UUID to filter the output of software deployment list command.

```
$ openstack software deployment list server <server-uuid> | grep -v
 COMPLETE
```

Failed software deployment reports also contain the reason for failure. Use the software deployment uuid to display all the information.

```
$ openstack software deployment show <uuid>
```

Post Deployment Steps

Flavors created on the OVS-DPDK enabled compute nodes need to have the following metadata tags :

1. hw:cpu_policy=dedicated
2. hw:cpu_thread_policy=require
3. hw:mem_page_size=large
4. hw:numa_nodes=1
5. hw:numa_mempolicy=preferred

To create a custom flavor for OVS-DPDK instance, follow the steps below.

1. Source the overcloud resource configuration file.

   ```
   $ source <overcloudrc>
   ```

2. Create the custom flavor.

   ```
   $ openstack flavor create <flavor-name> --disk 40 --ram 4096 --vcpu 4
   ```

   Note: Disk size, RAM and number of vCPUs can vary.

3. Add the metadata tags to the newly created flavor.

   ```
   $ openstack flavor set <flavor-name> \
       --property hw:cpu_policy=dedicated \
       --property hw:cpu_thread_policy=require \
       --property hw:mem_page_size=large \
       --property hw:numa_nodes=1 \
       --property hw:numa_mempolicy=preferred
   ```

   Note: To add metadata tags to an existing flavor, only step 3 is required.

Why two NICs?

Two extra NICs are required to achieve a NIC level high availability. When creating a network bond, ports from two different NICs are selected, the two ports then are attached to two different leaf switches. This removes any single point of failure.

Cores distribution

Cores are distributed into three mutually exclusive groups. This distribution is based on performance and utilization policies and architecture of compute node.

Groups

There are limited number of physical cores available per compute node. To optimize the performance, all cores are divided into three different groups. Each group will be used for a dedicated purpose. This eliminates random scheduling by the kernel, and results in a more optimized use.

Currently the cores are distributed into three groups, which are as follow:

1. **PMD core list:** These vCPUs are used to run DPDK PMD threads. For best performance each thread is run on a dedicated vCPU. This list can be further divided into two parts:
   a. PMD cores for physical interfaces, connecting OvS to physical interfaces. Cores needs to be from the same CPU sockets as the respective NICs are attached to.
   b. PMD for virtual interfaces (vNIC), connecting OvS to Virtual Machines. It needs to be from the socket on which the Instance is running.
2. **Host core list:** It runs all the host processes and other OvS processes like handlers and revalidators.
3. **VM core list:**Dedicated to Nova and used for computation of instances.

Policies

A policy based approach is used for core distribution. By changing the policy, the overall core distribution can be changed. Currently the only supported policy is “Balanced”. By using NICs on both CPU sockets, and utilizing both CPU sockets for Instances, a balance is achieved.

Following are other policies that can be enabled in future releases.

1. **Optimized for Performance:** To achieve optimal performance through lowest latency, Instance are scheduled only on the CPU socket which has both interfaces attached to it.
2. **Optimized for Utilization:** For optimal CPU utilization, performance is traded off for utilizing all the available resources.

Architecture

The Cores distribution may vary, depending upon the number of total cores and affinity of NICs.

With two extra NICs for OVS-DPDK, there are total of three Intel x520 10G cards on each compute node. Currently the OVS-DPDK enablement is supported on two types of compute nodes that are listed below:

1. DELL PowerEdge R630

   Considering the current architecture of DELL PowerEdge R630 servers, two NICs (PCI Slot 1 & PCI Slot 2) have affinity with socket 1, while one (PCI Slot 3) has affinity with socket 0.

   Selecting two slots for OVS-DPDK from the available PCI slots creates the following possibilities:

   a. PCI Slot 1 and PCI slot 2
   b. PCI Slot 2 and PCI slot 3
   c. PCI Slot 3 and PCI slot 1

   Note: In this release only Option b is supported.

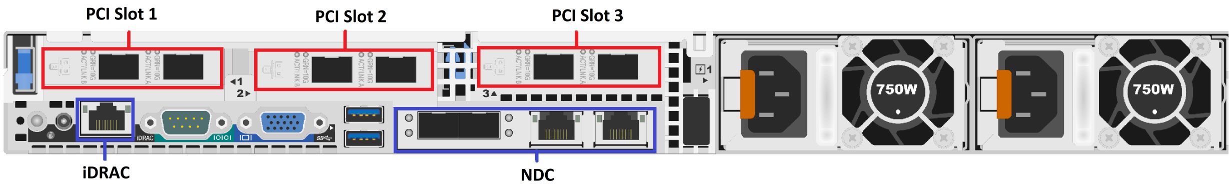

Figure 12: DELL PowerEdge R630 backplane with 3 NICs in PCI Slots

2. DELL PowerEdge R730

   DELL PowerEdge R730 have six PCI Slots as compared to three in DELL PowerEdge R630. In this release it is required that OVS-DPDK NICs should be in PCI Slot 4 & 5. PCI Slot 4 has affinity with CPU Socket 1 while PCI Slot 5 has affinity with CPU Socket 0.

   This NIC attachment is similar to that of DELL PowerEdge R630.

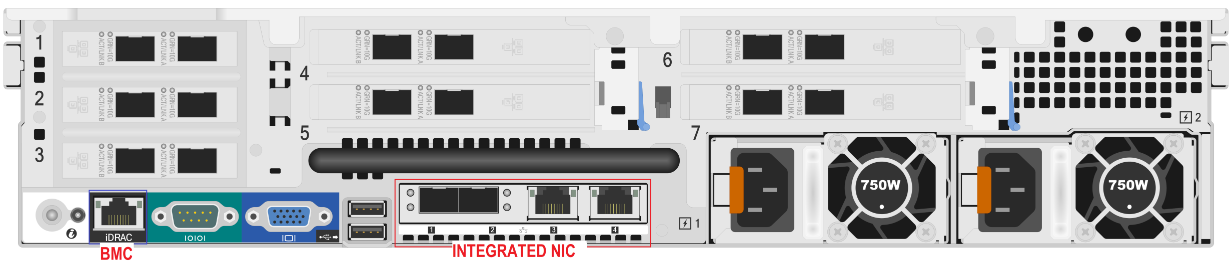

Figure 13: DELL PowerEdge R730 backplane with 6 NICs in PCI Slots

Example Cores distribution

Note: All the calculations are done with 48 vCPUs. Increasing or decreasing the number of cores per socket will affect the total number of instances that can be run on the compute node.

One interface is attached to socket 0, and the other to socket 1, so the cores distribution for both modes will be as follows:

Mode I, Tenant Networks Only:

Initially, each socket has 24 vCPUs available.

1. **Host Core List:**

   Two sibling pairs i.e. 4 vCPUs from each socket are assigned to the host core list. The list includes core 0,24,2,26,1,25,3,27.

2. **PMD Core List:**

   a. **Physical:**

      Having assigned cores to the Host list, it leaves 20 vCPUs on both sockets to be selected from. A sibling pair is assigned from each socket. The list includes vCPUs 6,30,5,29.

   b. **Virtual:**

      This distribution leaves 18 vCPUs in each socket.

      PMD core list for virtual interfaces and the VM core list are interdependent on each other. For each instance, dedicated vCPUs need to be assigned for computation and for handling the vNICs.

In this example, 4 vCPUs are assigned to the VM core list for computation while 1 vCPU is assigned to PMD core list for vNIC.

Out of these 18 vCPUs 4 vCPUs are assigned to this list, and these vCPUs will be used by vNICs that are attached to Instances spawned on the other 14 vCPUs.

3. **VM Core List:**

   Having assigned the vCPUs to the PMD core list, it leaves 14 vCPUs in each CPU socket. These remaining vCPUs are assigned to the VM core list to be used for instances.

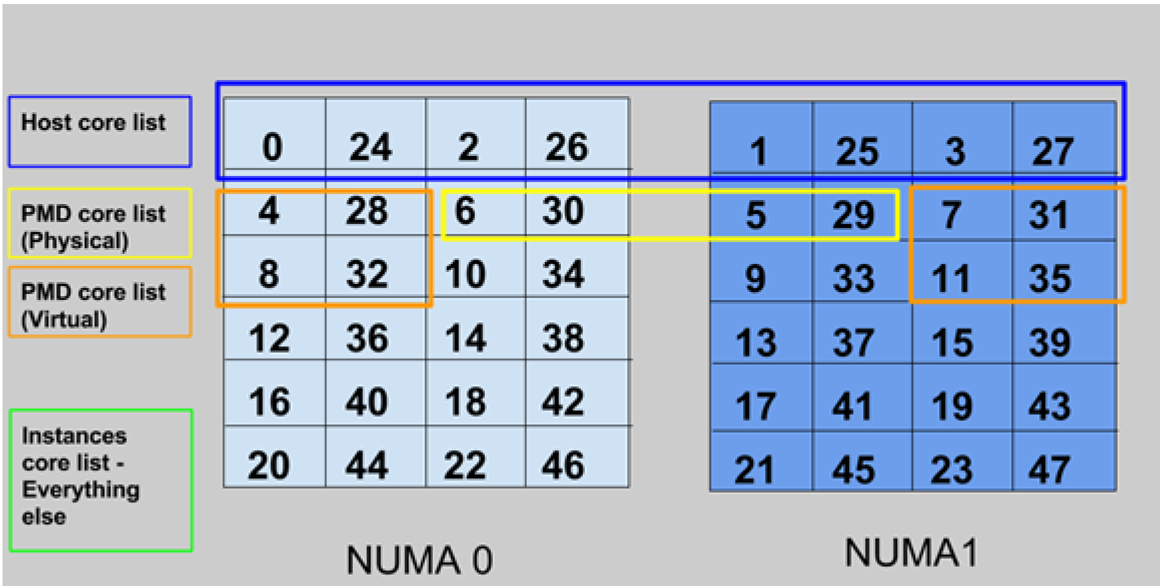

Figure 14: 48 cores system, with 4 ports for OVS-DPDK, 2 port attached to each socket

Note: With Mode I, up to 4 vNICs on each socket will be polled using dedicated vCPUs. Creating more vNICs will force vCPUs to poll more than one vNIC, which can lead to performance degradation.

Mode II, Tenant and External Networks:

Initially, each socket has 24 vCPUs available.

1. **Host Core List:**

   Two sibling pairs of 4 vCPUs from each socket are assigned to the host core list. The list includes core 0,24,2,26,1,25,3,27.

2. **PMD Core List:**

   a. **Physical:**

      Having assigned cores to the Host list, it leaves 20 vCPUs on both sockets to be selected from. One vCPU is assigned from each socket. The list includes Cores 5 and 30.

      Note that sibling pair is not assigned here. The siblings of core 5 and 30 are 29 and 6 respectively. These two cores cannot be assigned to VM core list.

   b. **Virtual:**

      This distribution leaves 19 vCPUs in each socket.

      PMD core list for virtual interfaces and the VM core list are interdependent on each other. For each instance, dedicated vCPUs need to be assigned for computation and for handling the vNICs.

In this example, 4 vCPUs are assigned to the VM core list for computation while 1 vCPU is assigned to PMD core list for vNIC.

Sibling pair cannot be shared between PMD core Lists and VM core List. This forces that the left over sibling vCPUs 29 & 6 are assigned to PMD core list (virtual).

This leaves 18 vCPUs or 9 sibling pairs in both sockets. Out of these, 4 more vCPUs are assigned to virtual PMD list, and these vCPUs will be used by vNICs that are attached to the instances spawned on the other 14 vCPUs.

3. **VM Core List:**

Having assigned the vCPUs to the PMD core list, it leaves 14 vCPUs in each CPU socket. These remaining vCPUs are assigned to the VM core list to be used for instances.

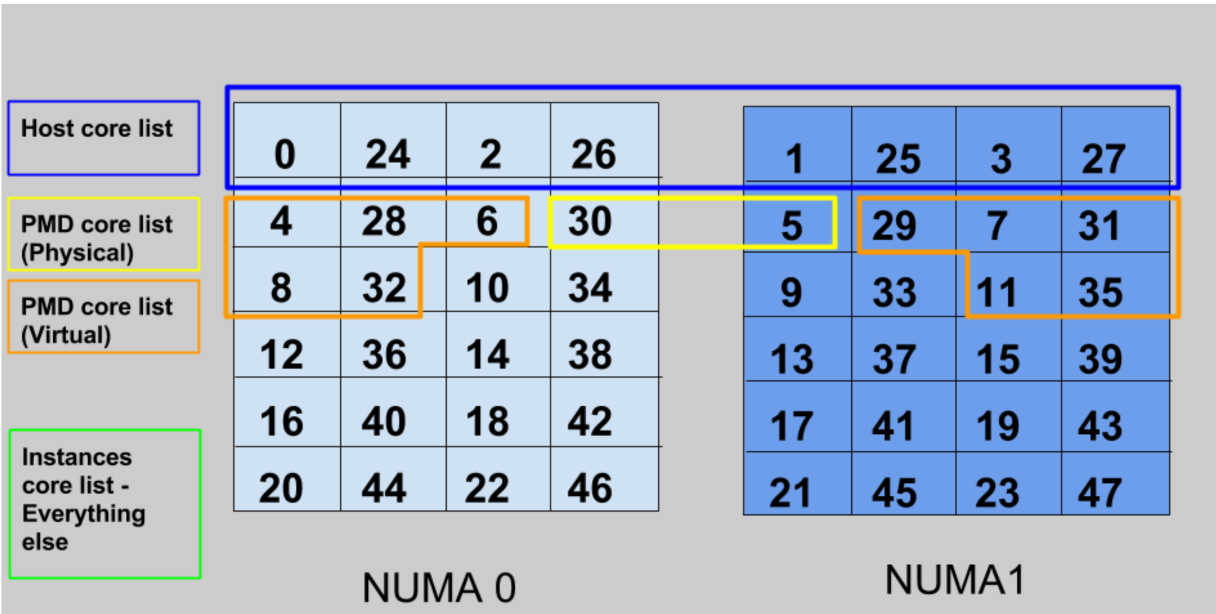

Figure 15: 48 cores system, with 2 ports for OVS-DPDK, 1 port attached to each socket

Note: With Mode II, up to 5 vNICs on each socket will be polled using dedicated vCPUs. Creating more vNICs will force vCPUs to poll more than one vNIC, which can lead to performance degradation.

Appendix G

Post Deployment features

Topics:

This appendix provides the information about the post deployment features for cloud-admin.

Ease of Use

Ease of Use is used for the post-deployment customization of Dell EMC Red Hat NFV Solution 10.0.1 for various use cases of NFV. Multiple projects, networks and Security Groups can be created and deleted with great ease by running the Ease of Use playbooks. Cloud administrator has the flexibility to provide the VNF related information in an Excel spreadsheet, which will then be used by the Ansible playbooks to deploy the virtual network functions. Following resources can be created/deleted automatically using this feature:

- Projects
- Users
- Networks
- Subnets
- Routers
- Security Groups

The feature contains creation and deletion playbooks which take input from the excel spreadsheet. As the name suggests, the creation playbook is used to create the OpenStack resources. Multiple resources can be created using a single command. Similarly, the deletion playbook is to be used for deleting the created projects. The user can delete one project at a time by providing its name in the deletion command. Deleting a project also deletes all associated virtual resources.

Using this excel spreadsheet, Ease of Use feature can create up to 500 projects. Each project can have up to one Router; 3 networks along with one subnet for each; one User and up to 9 security groups.

Note: See the document Dell EMC RedHat Openstack NFV Feature Ease of Use Guide for further information on Ease of Use feature.

SR-IOV

SR-IOV feature is enabled in Dell EMC Red Hat NFV Solution 10.0.1 to enable external network access to the instances through the lowest latency path.

Single root I/O virtualization (SR-IOV) is an extension to the PCI Express (PCIe) specification. SRIOV 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 hypervisor, resulting in multiple CPU interrupts per packet. These extra interrupts can result in 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.

SR-IOV enablement script is sub divided 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 is the first of the two scripts that should be run. Cnode pass enables SR-IOV on a single compute node in Dell EMC Red Hat NFV Solution version 10.0.1. It has two modes: ephemeral and persistent. The ephemeral mode of Cnode pass is the non-persistent setup of SR-IOV.It creates the supported number of Virtual Functions for the desired Physical Functions on a compute node. All the created VFs will be removed after the compute node is rebooted. The persistent mode of Cnode pass creates the SR-IOV environment that would be persistent across multiple reboots of a compute node.

Instance pass is the final step in enabling the SR-IOV feature in Dell EMC Red Hat NFV Solution version 10.0.1. This script attaches two VFs to the desired instance and creates a supported bond over the attached VFs. Connectivity to the external network is enabled using this bond for the desired instance overriding the OpenStack tenant network default route.

SR-IOV feature can further be enhanced by enabling NIC Alignment feature where the vCPUs in the instance and physical NIC are aligned with the same CPU socket.

Note: See the document Dell EMC RedHat Openstack NFV Feature SR-IOV guide for further information on SR-IOV feature.

Appendix H

IPv6 addressing for tenant VM (optional)

Topics:

This appendix provide instructions for enabling IPv6 addressing on Tenant VM.

Enabling Tenant VM IPv6 Addressing (optional)

The Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform provides the ability to enable tenant VM IPv6 addressing support on all Nova compute nodes in the solution. This Appendix provides the instructions to enable the feature. There is no special script needed for this enablement.

Note: The user must have admin scoped OpenStack permissions to execute the following steps.

Alternative 1: CLI-based approach

There are two steps to take to enable tenant VM IPv6 addressing capability.

STEP 1: On a Controller node in the solution, create a network using the command:

```
# neutron net-create <network-name>
```

The Neutron facility will print a table of field-value parameters for the new network <network-name>, after it is created.

STEP 2: On the Controller node in the solution, create a new subnet using the command:

```
$ neutron subnet-create --name <network-name> \
  --ip-version 6 IPv6_NETWORK <network_CIDR> \
  --ipv6_ra_mode slaac \
  --ipv6_address_mode slaac
```

The Neutron facility will print a table of field-value parameters for the new subnet <network-name>, after it is created.

Now, create the desired tenant VM instance, and assign its NIC to use the IPv6 network that was just created.

Alternative 2: UI-based approach

Take the following steps to enable tenant VMs to communicate using IPv6.

1. Log in to the Horizon UI of OpenStack.
2. Select **Project** on the top toolbar.
3. Select **Network**
4. Then, select **Networks**.
5. Click the **Create Networks** button.

These steps will lead you to the following screen shown in *Figure 16: OpenStack Horizon UI form used to enable IPv6 communication between tenant VMs* on page 109. Enter appropriate values in the form fields, as they pertain to your solution deployment.

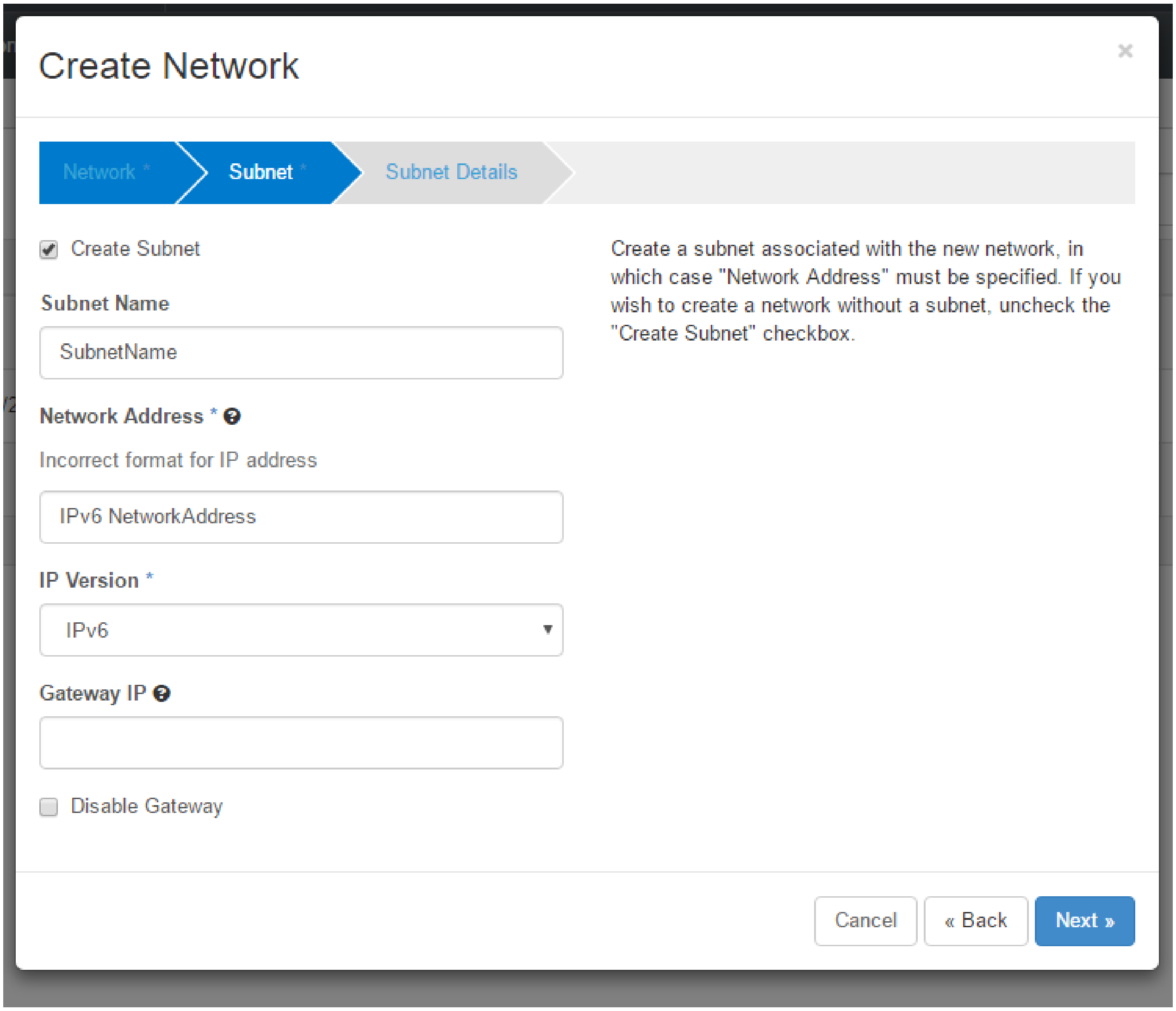

Figure 16: OpenStack Horizon UI form used to enable IPv6 communication between tenant VMs

Appendix I

Security-enhanced Linux

Topics:

This appendix provides basic introduction for SELinux.

Notes on Enablement of Security-enhanced (SELinux) protection capability

Starting with version 10.0.1, the Dell EMC Ready Bundle for Red Hat OpenStack NFV Platform provides enablement of Security-Enhanced Linux (SELinux) protection on all Controller and Compute nodes in the solution. This feature is enabled by default, and does not require the end-user to take any actions.

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

The solution components that have been enabled for SELinux protection are listed below. SELinux is enabled by default and runs in enforcing mode. No separate actions are necessary.

```
Glance
HAProxy
KeepAlive
Keystone
MongoDB
MySQL
Neutron
Nova
OVS
RabbitMQ
Redis
Swift
```

The following two solution components are not enabled for SELinux protection:

```
Ceph
Cinder
```

Appendix J

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*.