FlexRAN and Mobile Edge Compute (MEC) Platform

Setup Guide

December 2018

Revision 1.4

Intel Confidential

Document Number: 575891-1.4

Contents

Figures

Tables

Revision History

Document Number	Revision Number	Description	Revision Date
575891	1.4	Updated for FlexRAN Software release v18.12. • Chapter 2.0 Wind River* Titanium Cloud 5 Installation • Chapter 3.0 Enabling Ironic Compute Node	December 2018
575891	1.3	Updated for FlexRAN Software release v18.09. • Section 2.8 Support Wind River* Titanium Server 5 • Section 2.9 Install Backup Controller • Section 2.10 Configure Backup Controller • Section 2.11 Increase volume sizes	October 2018
575891	1.2	• Section 2.9.2. Added Kernel Parameters • Section 2.13.1. Updated Guest Image Command Line • Section 3. BIOS setting updated with summary tables included. • Section 3.1.2. TDP settings updated • Section 3.3. Added Setting environment without VMs or Containers. • Section 5. 1588 setup [Early access] - NEW	June 2018
575891	1.1	• Table 2. Updated "Storage" Category row. • Minor edits throughout for clarity. • Sections 2.3. Added paragraph to "Controller + Storage" sub-section. • Section 2.4. Updated Step 7 in the numbered list. • Section 2.5. Updated first three bullets and last bullet. • Section 2.6. Added warning at beginning of section. • Section 2.8. Updated first paragraph and #3 in numbered list. • Section 2.9.2. Updated third paragraph. • Section 2.14. Updated Step 7 in numbered list. • Section 2.15. Updated Steps 5 and 7 in numbered list. • Section 4.1. Updated Steps 1 and 2 in numbered list.	April 2018
575891	1.0	Initial release.	January 2018

§

1.0 Introduction

This document covers the typical steps involved in setting up the hardware and infrastructure software such as basic Input/Output System (BIOS), Operating system (OS), and Virtual Machines (VMs) for running FlexRAN and Mobile Edge Compute (MEC). The document provides a baseline platform configuration for engineering, validation, and Best Known Configuration (BKC). This document does not cover the details of FlexRAN or MEC workloads. Refer to Table 2 *FlexRAN Reference Solution L1 User Guide* (570228) and *Network Edge Virtualization (NEV) Software Development Kit (SDK) User Guide* (569396) for more details about FlexRAN and MEC, respectively.

1.1 Acronyms

Table 1. Acronyms

Acronym	Description
ATCA	Advanced Telecommunications Computer Architecture
ART	Advanced Radio Tester
BBU	Base Band Unit
BDW	Broadwell
BIOS	Basic Input/Output System
BKC	Best Known Configuration
BSP	Board Support Package
CA	Carrier Aggregation
CC	Component Carrier
CLI	Command Line Interface
CMP	Compute Node
CPU	Central Processor Unit
DHCP	Dynamic Host Configuration Protocol
EPC	Evolved Packet Core
FPGA	Field Programmable Gate Array
FTP	File Transfer Protocol
GW	Gate Way
HO	Hardware Offload
IPMI	Intelligent Platform Management Interface
L1	Layer 1 or Physical Layer
L2	Layer 2 or Media Access Control Layer
MEC	Mobile Edge Compute

Acronym	Description
NEV	Network Edge Virtualization
NIC	Network Interface Card
OAM	Operations, Administration, and Maintenance
OS	Operating System
OTA	Over The Air
OVP 6	Open Virtualization Platform 6
PCIe*	Peripheral Component Interconnect Express*
PF	Physical Function
Intel® QAT	Intel® Quick Assist Technology
RAN	Radio Access Network
RRH	Remote Radio Head (e.g. Ace Axis ART)
SDK	Software Development Kit
SRIOV	Single-Root I/O Virtualization
TiC5	Titanium Cloud 5
USB	Universal Serial Bus
VM	Virtual Machine
VF	Virtual Function
vPTP	Virtual Precision Time Protocol
VT	Virtualization Technology
WFP	Wolf Pass
WR	Wind River*

1.2 Reference Documents and Resources

Table 2. Reference Documents and Resources

Title	Document Number
FlexRAN Reference Solution Software Release Notes	575822
FlexRAN 4G Reference Solution L1 User Guide	570228
Intel® Network Edge Virtualization Software Development Kit (Intel® NEV SDK) Release Notes	572124
Intel® Network Edge Virtualization Software Development Kit (Intel® NEV SDK) User Guide	569396

§

2.0 Wind River* Titanium Cloud Installation Guide

This section describes how to install and configure Wind River Titanium Cloud 5.

2.1 Hardware

This guide was validated with the following hardware.

Table 3. Hardware Used for Validation

Category	Wireless BKC Compute Node – Wolf Pass	Wireless BKC Control Node - Wildcat Pass
Board	Wolf Pass S2600WFQ server board (with symmetric Intel® Quick Assist Technology (Intel® QAT)	Wildcat Pass - S2600WT2R
Processor	2 x Skylake Server H-0 27.5MB 20c 2.4 GHz 150W 768 GB XCC Intel® Xeon® GOLD 6148	2 x Intel® Xeon® processor E5-2680 v4 @ 2.40 GHz
	2 x associated Heatsink	2 x associated Heatsink
Memory	12x Micron* 16 GB DDR4 2667 MHz DIMMS	12x Kingston* 16 GB DDR4 2133 MHz DIMMS
Chassis	2U Rackmount Server Enclosure	2U Rackmount Server Enclosure
Storage	One 2.5" 2TB SATA HDD (Seagate*) (A miniSAS to SATA cable also required)	2 x 2.5" 2TB SATA HDD (Seagate*)
NIC (Used for data plane traffic)	1x Intel Fortville NIC X710DA4 SFP+ (PCIe* Add-in-card direct to CPU-0)	Wildcat Pass - Optional for FlexRAN 1x Intel® Fortville NIC X710DA4 SFP+ (PCIe Add-in-card direct to processor-0)
Intel® QAT	Intel® Quick Assist Adapter Device 37c8 (Symmetric design)	N/A for Wildcat Pass
other add-in card	Intel® Ethernet Network Connection OCP X557-T2 (Used for Openstack management, OAM and Data)	N/A for Wildcat Pass (On board 1G copper NIC used for Openstack Management, OAM, and Data)
	3x PCIe Riser cards	N/A
Firmware	BIOS/BMC/FRUSDR	BIOS/BMC/FRUSDR
	FVL25G_FW	N/A
Switch	8 to 16 port Copper Switch	

Table 4. VM Hardware Configuration

Category	Wireless BKC Compute Node – Wolf Pass	Wireless BKC Control Node - Wildcat Pass
FlexRAN VM	WindRiver customized Centos VM 19x vCPU 20GB RAM 8x 1G huge page memory 2x passthrough vNIC	WindRiver customized Centos VM 15x vCPU 20GB RAM 8x 1G huge page memory 2x passthrough vNIC

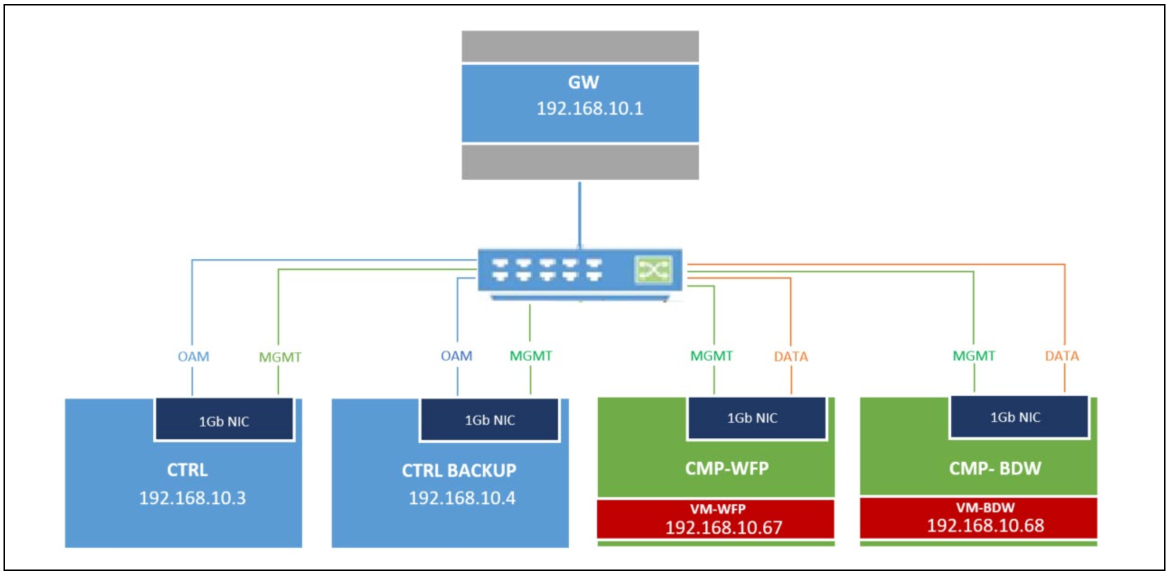
2.2 Network Topology

`CMP-WFP` and `CMP-BDW` are Skylake-SP Wolf Pass and Broadwell-EP Wildcat Pass compute node names, respectively. `Compute-0` is used throughout the document as an example. Figure 1 outlines the network topology for two compute nodes that are managed by one control node. This setup is for customers who would like to have both the Skylake-SP and Broadwell-EP compute node FlexRAN setup.

The Operations, Administration, and Maintenance (OAM) and Management traffic mentioned in Figure 1 is part of the Openstack communication traffic. Data Traffic mentioned in Figure 1 is part of the user management traffic such as ssh and File transfer. Data traffic should not be confused with the Data plane/Fast path traffic.

The recommended best practice is to have separate switches for the MGMT DATA and OAM networks. It is also ideal to have two control nodes instead of one, both nodes connected in parallel, where one automatically becomes active while the other is in standby mode.

Figure 1. Network Topology – Two Compute Nodes Managed by a Control Node with Backup



2.3 Titanium Cloud Hardware Reference Platform

Controller + Storage: Provides cloud and infrastructure management services and Virtual Machine (VM) storage services if storage nodes are not used (two hard drives for controller are required). Provides dedicated storage services for the cluster.

Note: For the two controller hard drives, a lower capacity drive (such as SSD) can be used in the sda slot and higher capacity disk in the sdb slot, as the disk in sdb is used for volume storage.

Compute: Provides processing resources and optionally ephemeral storage for VMs.

Note: 1G switch is necessary.

2.4 Summary of Hardware Configuration Procedures

The following section outlines specific configuration procedures for the following hardware:

- Installing Controller Node.
- Configure Controller Node.
- Install and configure backup controller.
- Install the Compute Node.
- Configure local storage on the compute node, for use by VMs.
- Create Provider Network.
- Configure the Compute Node with data interfaces.
- Configure the Compute Node with storage nodes.
- Unlock Compute Node.
- Create networks.
- Prepare flavor for VM.
- Create volume with VM image.
- Create VM instance.
- Install FlexRAN on created VM.

2.5 Before Installing Software and License

Make sure the following software components are available before proceeding:

- Wind River* TS5 Cloud installer:

  Contact Wind River* Customer Support team for acquiring this installer package: https://www.windriver.com/support/

- Sample Guest Image to run FlexRAN and MEC (Data plane and Routing):

  Contact Wind River* Customer Support team for acquiring this Sample Guest Image.

- FlexRAN Software and Intel® NEV SDK Release Software:

  Contact Intel Customer Support team for acquiring the release software.

- Intel® Parallel Studio license:

Note: (Important) Replace the license file in the `/opt/intel/licenses` with your license file.

— Contact Intel Customer Support team for acquiring the Intel® Parallel studio license.

- Wind River* TS Cloud License

  — Contact Wind River* Customer Support team for acquiring the Windriver TS cloud license.

2.6 Installing Controller Node

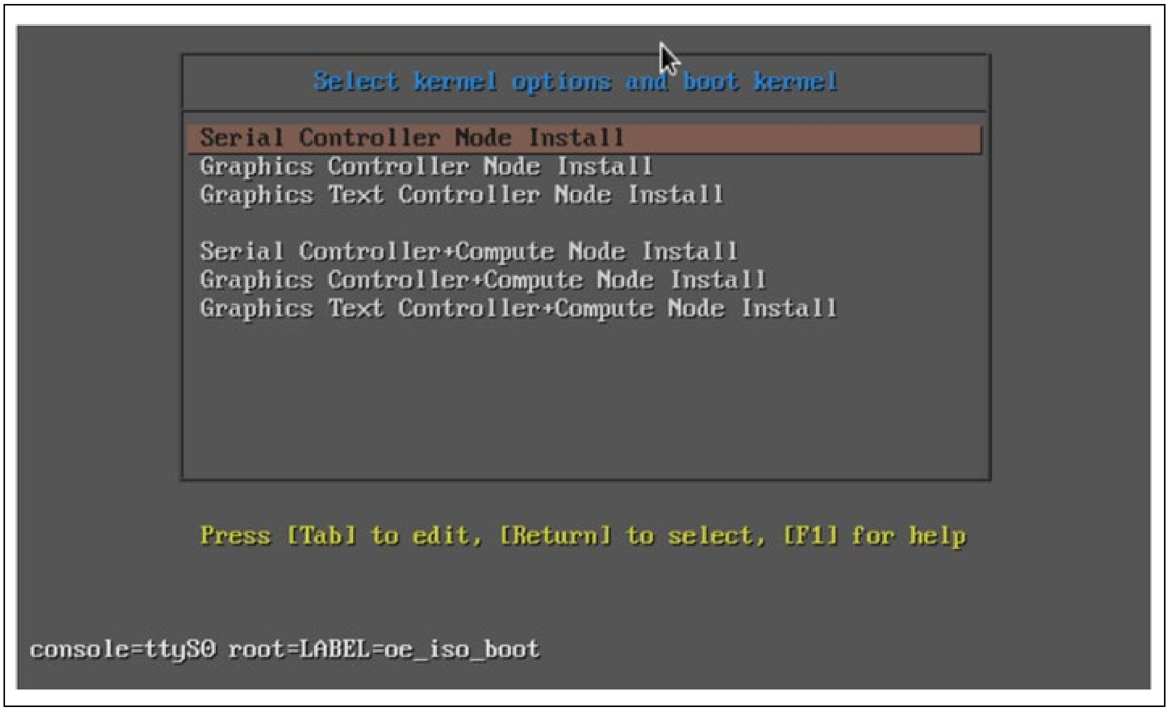
Warning: Controller installation will destroy any previous image without warning. Only proceed if new image is required.

1. Prepare a bootable USB flash drive from the boot image file.
2. Boot from USB flash drive.
3. To install a standalone controller, select Serial Controller Mode Install as shown in Figure 2.

Note: Graphics Controller Node Install is preferred.

4. From menu select Standard Controller Configuration -> Graphical Console -> Standard Controller Configuration.

Figure 2. Select Kernel Options and Boot Kernel



Note: Figure 2 is for the Titanium Server 3. TC5 screen is slightly different.

5. Monitor the initialization until it is complete. Installation process is automated.

   The installer initializes the target hard drive with the Titanium Cloud image. When initialization is complete, a reboot is initiated on the host.

6. After initialization is complete, immediately remove the USB flash drive from the host to ensure that the host reboots from the hard drive.

   After a few minutes, the host reboots from the hard drive.

7. Log into the host as `wrsroot`, with password `wrsroot`.

Note: The first time you log in as `wrsroot`, you are required to change your password.

8. The host is now ready for configuration as controller-0.

The Titanium Cloud clusters are now up to date.

2.7 Licensing and Authentication Requirements

2.7.1 Installing a License

To install a license, initially copy the license file to a designated license directory on the controller host. The default designated directory is `/home/wrsroot`. A different directory can be specified during installation.

2.7.2 Updating a License

After you have installed a license, update it by copying the new license file to the designated license directory ("example `/usr/licenses`") on the active controller host and then run the license-install utility as shown in the following example:

```
$ sudo /usr/sbin/license-install <license_file>
```

Caution: It is recommended that you update licenses obtained in Section 2.5 before they expire. Manual recovery of VMs may be required if the license is upgraded after expiration.

2.8 Updating TiC5 with patches

It is advised to install available Wind River* patches for Titanium Cloud 5 (TiC5) before the configuration.

To install the patches:

1. Copy the patch files to `/home/wrsroot`.

```
$ sudo sw-patch upload-dir /home/wrsroot/
$ sudo sw-patch apply --all
$ sudo sw-patch install-local
```

2. Reboot the system when prompted after patch installation.

2.9 Configure Controller Node

Before running the configuration script, ensure that the management interface on `controller-0` is connected and operational. Check which physical interface is going to be used for compute nodes management. `enp4s0f` is used for the following example.

1. Connect a server containing to the OAM network.
2. Copy the Titanium Cloud 5 license file to /home/wrsroot/license.lic on the controller.
3. Create a initialization file in the `/home/wrsroot` directory.

```
$ vim  intel-TiC_config.ini
```

With following content:

Note: Interface ports and password should be adjusted accordingly.

```
[LOGICAL_INTERFACE_2]
LAG_INTERFACE = N
INTERFACE_MTU = 1500
INTERFACE_LINK_CAPACITY = 1000
INTERFACE_PORTS = enp4s0f

[LOGICAL_INTERFACE_1]
LAG_INTERFACE = N
INTERFACE_MTU = 1500
INTERFACE_PORTS = eno1

[VERSION]
RELEASE = 18.03

[MGMT_NETWORK]
CIDR = 192.168.204.0/24
MULTICAST_CIDR = 239.1.1.0/28
DYNAMIC_ALLOCATION = Y
LOGICAL_INTERFACE = LOGICAL_INTERFACE_2

[OAM_NETWORK]
CIDR = 192.168.10.0/24
GATEWAY = 192.168.10.1
IP_FLOATING_ADDRESS = 192.168.10.2
IP_UNIT_0_ADDRESS = 192.168.10.3
IP_UNIT_1_ADDRESS = 192.168.10.4
LOGICAL_INTERFACE = LOGICAL_INTERFACE_1

[AUTHENTICATION]
ADMIN_PASSWORD = YourControllerPassword!
```

4. Start the controller configuration script with initialization file.

Warning: Do not run `config_controller` over ssh.

```
$ sudo config_controller --config-file /home/wrsroot/intel-TiC_config.ini
```

2.9.1 Set up Cinder and LVM Storage

1. From controller set up keystone admin:

```
$ echo 'source /etc/nova/openrc' >> ~/.bashrc
$ source /etc/nova/openrc
```

2. Set up Cinder Volumes:

```
$ system host-lvg-add controller-0 cinder-volumes
$ system host-lvg-modify controller-0 cinder-volumes -l thin
$ dev_uuid=`system host-disk-list controller-0 |grep -i sdb | \
     awk '{print $2}'`
$ disk_size=10240
$ idisk_uuid=`system host-disk-partition-add controller-0 ${dev_uuid} \
     ${disk_size} - t lvm_phys_vol | grep -i idisk_uuid | awk '{print $4}'`
$ system host-disk-partition-list controller-0 --disk $idisk_uuid
```

3. When above command shows status “Ready”:

```
$ partition_id=`system host-disk-partition-list controller-0 \
     --disk $idisk_uuid |grep -i sdb| awk '{print $2}'`
$ system host-pv-add controller-0 cinder-volumes $partition_id
```

4. Set up LVM Storage.
5. Verify there are no errors:

```
$ system storage-backend-add lvm -s cinder
```

   You should see the following output (if you do not see it, fix errors):

```
WARNING : THIS OPERATION IS NOT REVERSIBLE AND CANNOT BE CANCELLED. By
confirming this operation, the LVM backend will be created. Please refer to
the system admin guide for minimum spec for LVM storage. Set the 'confirmed'
field to execute this operation for the lvm backend.
```

6. Add LVM backend:

```
$ system storage-backend-add lvm -s cinder –confirmed
```

7. To check status of configuration run:

```
$ system storage-backend-list
```

8. When each status is “Configured” you can unlock the controller.

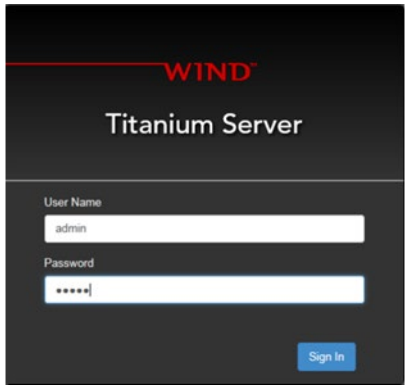
```
$ system host-unlock controller-0
```

2.9.2 Connect to Titanium Cloud Web Administration

On the Gate Way (GW) machine, launch a Web browser (ensuring to disable proxy settings if used), navigate to the Titanium Cloud Web Administration Interface (use controller IP address as previously defined 192.168.10.3).

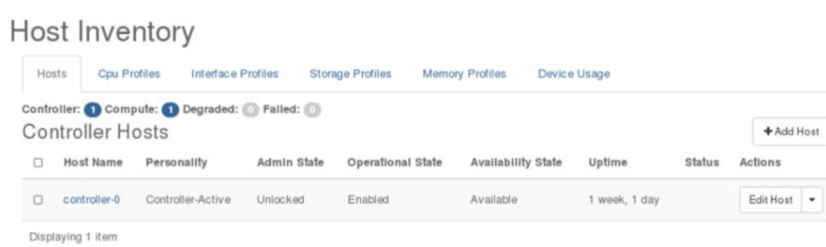
1. Login `admin/admin` (or password previously set)

Figure 3. Titanium Cloud Web Administration Interface



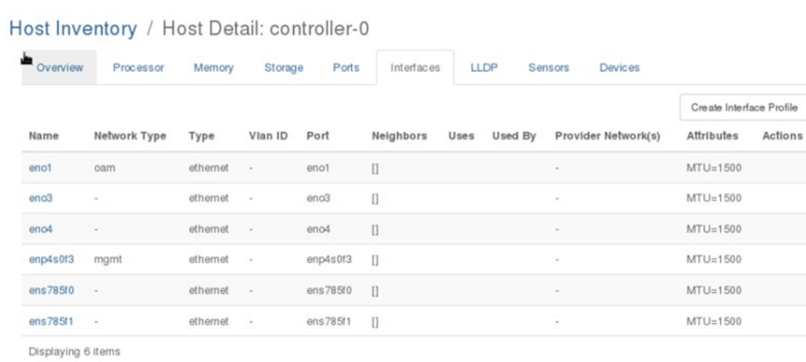
2. In the left-side pane, select **Admin > Platform > Host Inventory**, and then select the **Hosts** tab. `Controller-0` is reported in the host inventory list.

Figure 4. Host Inventory



3. Select Admin > Platform > Host Inventory > controller-0 > Interfaces.

Figure 5. Host Detail: `controller-0`



2.10 Configure Backup Controller

This section describes how to set up the backup controller (`controller-1`).

2.10.1 Install Backup Controller Node

Before initializing a backup controller node, ensure that the following conditions are satisfied:

- Controller-0 must be installed and configured.
- The node must be connected to the internal management network or PXE boot network using an Ethernet interface configured for PXE boot.
- The node must be configured in the BIOS to boot from the internal management network or PXE boot network.

Follow these steps to initialize the backup controller node:

1. With controller-0 running, start the host. The host boots from the network and then displays a message that it is waiting to be configured.

Figure 6. Waiting for Node Message

2. From controller-0 list the hosts connected to TC5 cluster:

```
$ system host-list
```

3. Determine the “id” of the node from the list and install it as controller host.

Note: `id` equals 2 in this case.

```
$ system host-update 2 personality=controller
```

4. The backup controller installs.
5. Wait for installation to finish and for the system to reboot.

   On success the node is reported as `Locked`, `Disabled` and `Online` in the Hosts list.

2.10.2 Configure Network and Cinder Storage

Note: The backup controller is configured from the master controller (controller-0).

To configure OAM network run the following command (where `ens4f1` = the name of interface connected to OAM):

```
$ system host-if-modify -nt oam controller-1 ens4f1
```

To configure cinder storage:

1. Set up Cinder Volumes.

```
$ system host-lvg-add controller-1 cinder-volumes
$ system host-lvg-modify controller-1 cinder-volumes -l thin
$ dev_uuid=`system host-disk-list controller-1 |grep -i sdb | \
    awk '{print $2}'`
$ disk_size=10240
$ idisk_uuid=`system host-disk-partition-add controller-1 ${dev_uuid} \
    ${disk_size} - t lvm_phys_vol | grep -i idisk_uuid | awk '{print $4}'`
$ system host-disk-partition-list controller-1 --disk $idisk_uuid
```

2. When above command shows status Ready, enter the commands:

```
$ partition_id=`system host-disk-partition-list controller-1 \
    --disk $idisk_uuid |grep -i sdb| awk '{print $2}'`
$ system host-pv-add controller-1 cinder-volumes $partition_id
$ system host-unlock controller-1
```

2.11 Increase Volume Sizes

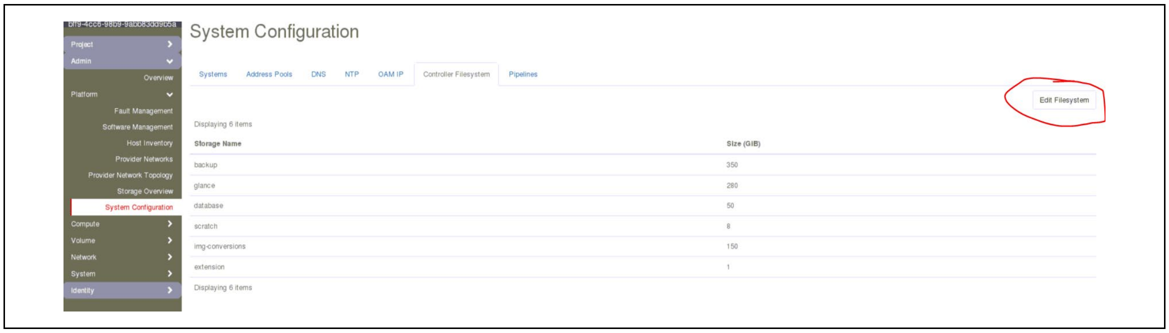
Default space dedicated for the following listed storage is small and therefore needs to be increased according to the size of the storage used. The following example sets and increase the storage size for:

- Database
- Glance
- Backup
- Image Conversion

From the dashboard:

1. Navigate into Admin -> Platform -> System Configuration -> Controller Filesystem.
2. Then click Edit Filesystem.

Figure 7. Edit Filesystem



3. Edit the size of the Storage and click **Save**.

   The filesystem will now update and the controllers will go into degraded mode.

4. Wait until both systems are in sync (Online, Unlocked and Available)

2.12 Compute Node

This section describes how to install and configure a compute node.

2.12.1 Install the Compute Node

Before initializing a compute node, ensure that the following preconditions are satisfied:

- `Controller-0` must be installed and configured
- It is also recommended that `Controller-1` is installed and configured.

Note: It is recommended best practice to use FlexRAN on Numa Node 0 (`CPU 1`). The corresponding NIC for connecting to the FerryBridge FPGA via passthrough FerryBridge should be connected to the correct PCIe slot available and indicated on the motherboard below at the riser card location.

- The node must be connected to the internal management network or PXE boot network using an Ethernet interface configured for PXE boot.
- The node must be configured in the BIOS to boot from the internal management network or PXE boot network.

2.12.1.1 Initialize the Compute Node

1. Power on the host with the `controller-0` node running. The host boots from the network and displays a wait message for the personality of the node to be configured.

Figure 8. Configure Node Wait Screen

2. Using the web administration interface, assign the node as a compute host.

   a. Select the **Hosts list**.

      The new node is shown in Figure 9 with empty cells for the Host Name and Personality.

Figure 9. UnProvisioned Hosts

UnProvisioned Hosts

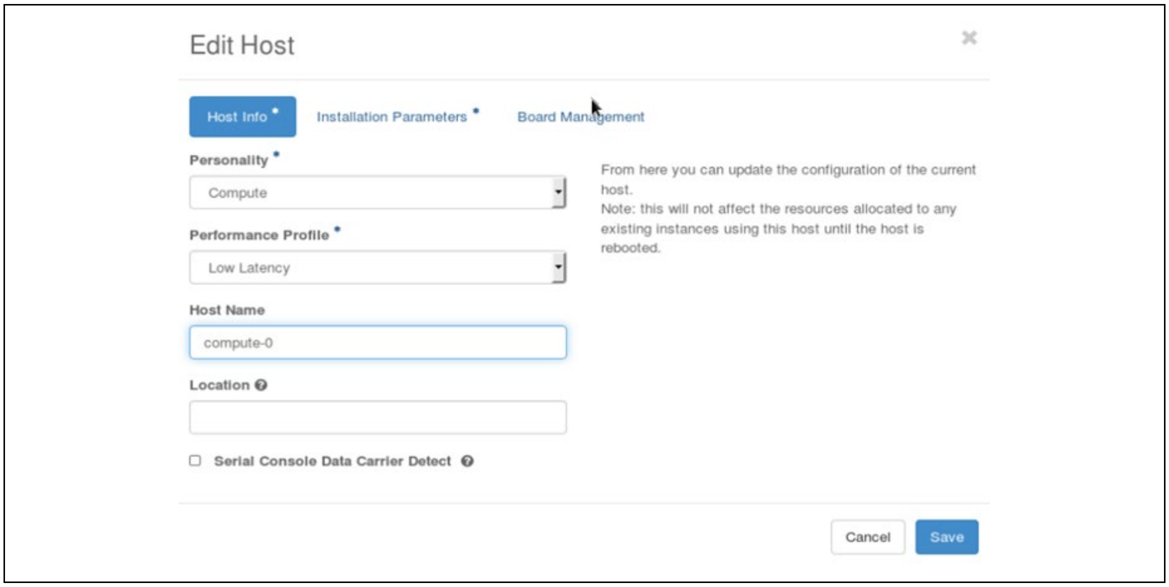
Host Name	Personality	Admin State	Operational State	Availability State	Uptime	Status	Actions
-	-	Locked	Disabled	Offline	0 minutes		Edit Host
-	-	Locked	Disabled	Offline	0 minutes		Edit Host

Displaying 2 items

   b. Click **Edit Host** for the new host.

      The Edit Host window appears as shown in Figure 10. Using the Performance Profile dropdown, select **Performance Profile > Low Latency**.

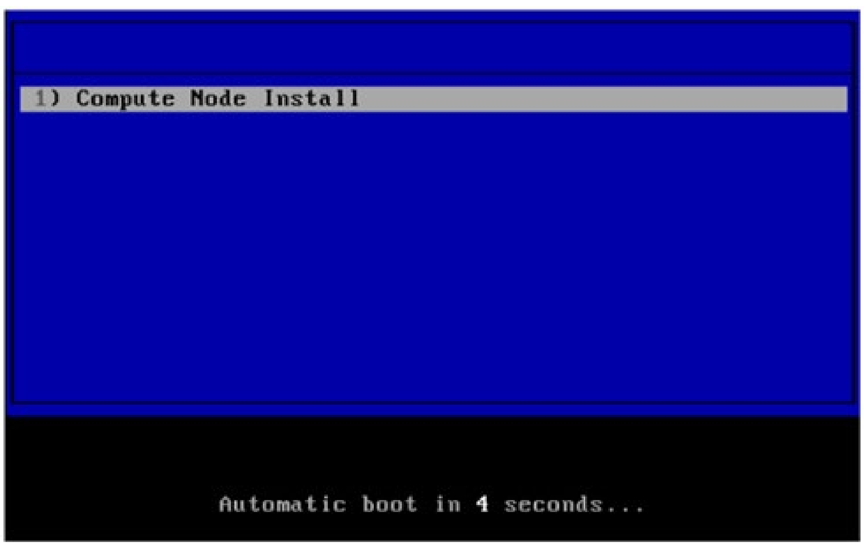
Figure 10. Edit Host



c. Click **Save** to initialize and configure the new node.

The node is restarted automatically and a display-device menu appears on the node console.

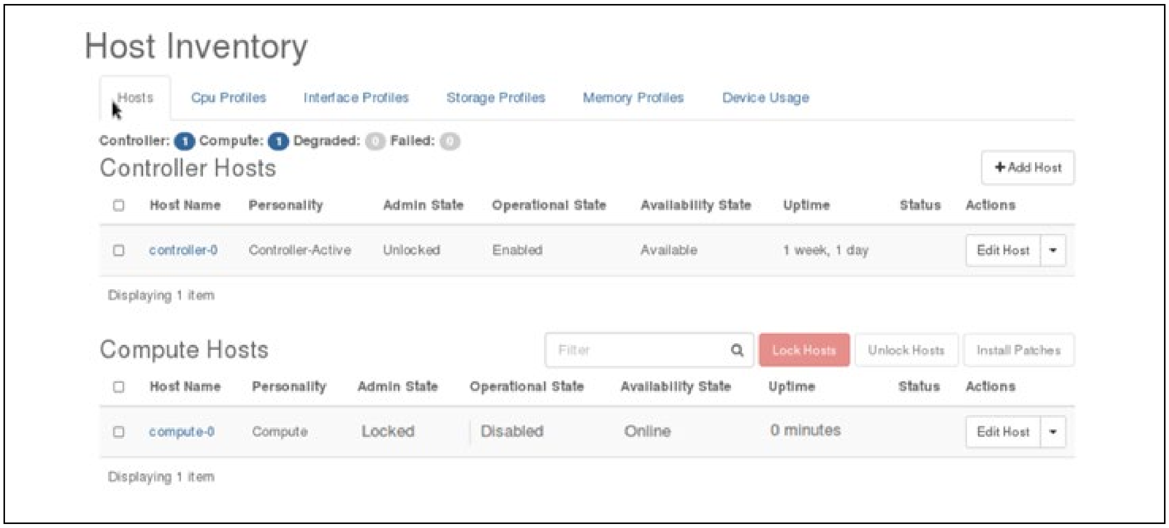
Figure 11. Display-Device Menu



d. Wait while the node is configured and rebooted.

A reboot can take up to 20 minutes. When the reboot is complete, the node is reported as Locked, Disabled, or Online in the Hosts list as shown in Figure 12.

Figure 12. Node Complete Screen



2.12.2 Kernel Parameters

Login to the installed compute node and verify kernel command line parameters.

```
cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-693.21.1.rt56.639.el7.tis.42.x86_64
root=UUID=a8362873-1942-4551-adcc-0a46cfff6d94 ro security_profile=standard
module_blacklist=integrity,ima audit=0 tboot=false crashkernel=auto
biosdevname=0 iommu=pt usbcore.autosuspend=-1 selinux=0 enforcing=0
nmi_watchdog=0 softlockup_panic=0 intel_iommu=on cgroup_disable=memory
skew_tick=1 hugepagesz=1G hugepages=2 hugepagesz=2M hugepages=0
default_hugepagesz=2M isolcpus=1-39 rcu_nocbs=1-39 kthread_cpus=0 irqaffinity=0
nohz_full=1-39 nopti nospectre_v2
```

2.12.3 Provider Network

Complete the interface provisioning on the host.

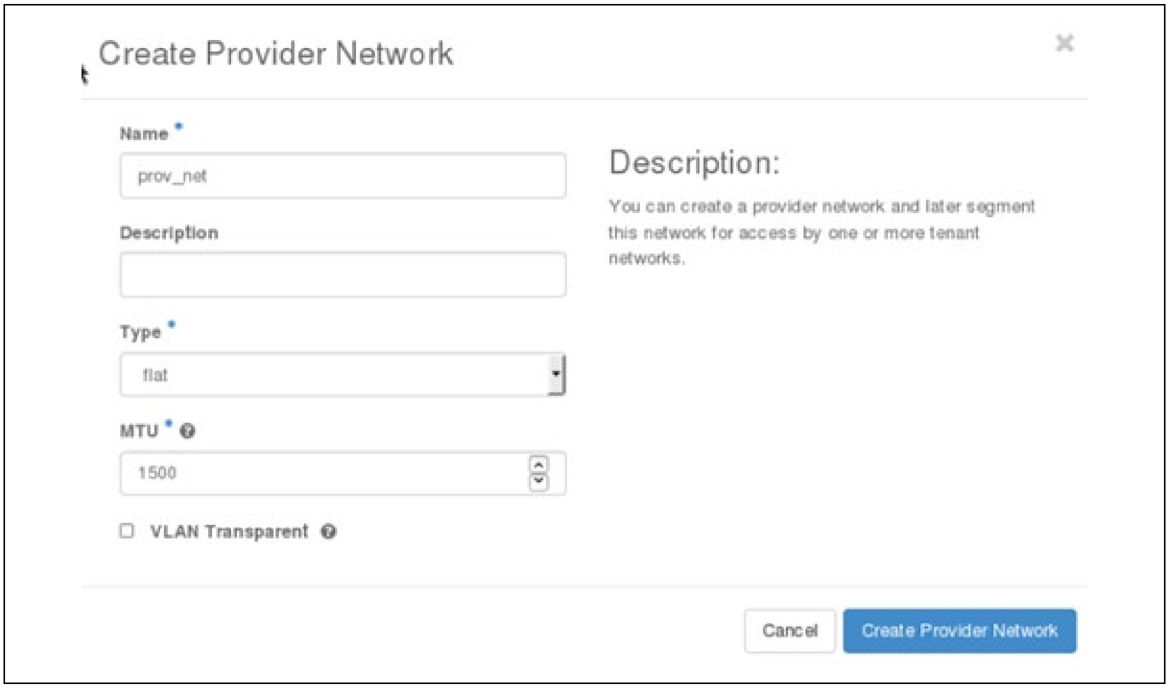
1. Configure provider networks.

   This is required to provision data interfaces on the compute node.

2. In the left-side pane of the horizontal dashboard, select Admin > **Platform**, and then select the Provider Networks tab > **Create Provider Network**.

3. In the Create Provider Network window, complete the fields as required.

Figure 13. Create Provider Network Interface

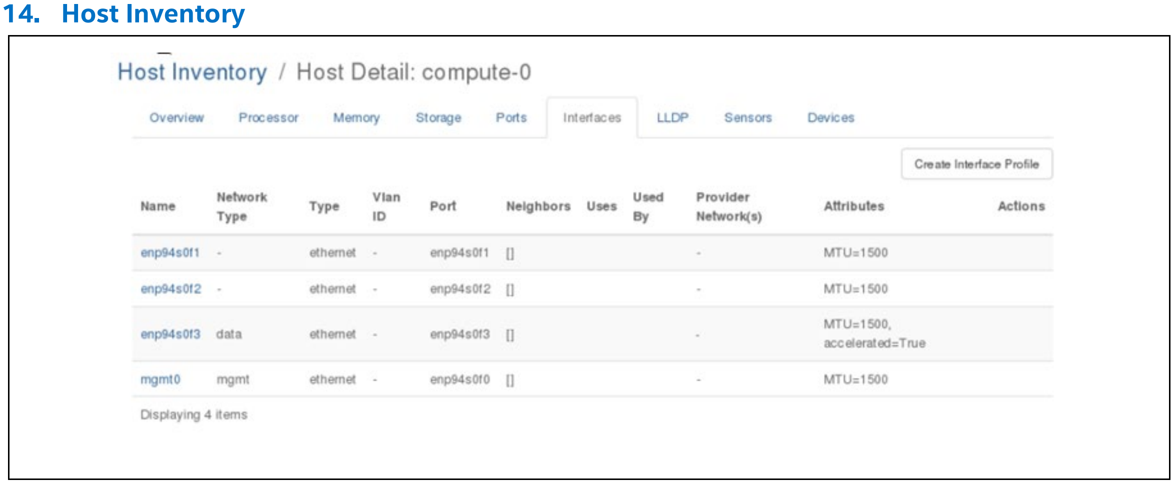


2.12.4 Data Interface

Configure the data and optional infrastructure interfaces.

1. Manually attach interfaces for the data networks.
2. Attach an Ethernet interface to a network by editing the interface.
3. Select **Admin > Platform > Host Inventory**, and then in the Host Name column, click the `compute-0` host.
4. Select the **Interfaces** tab.

Figure 14. Host Inventory

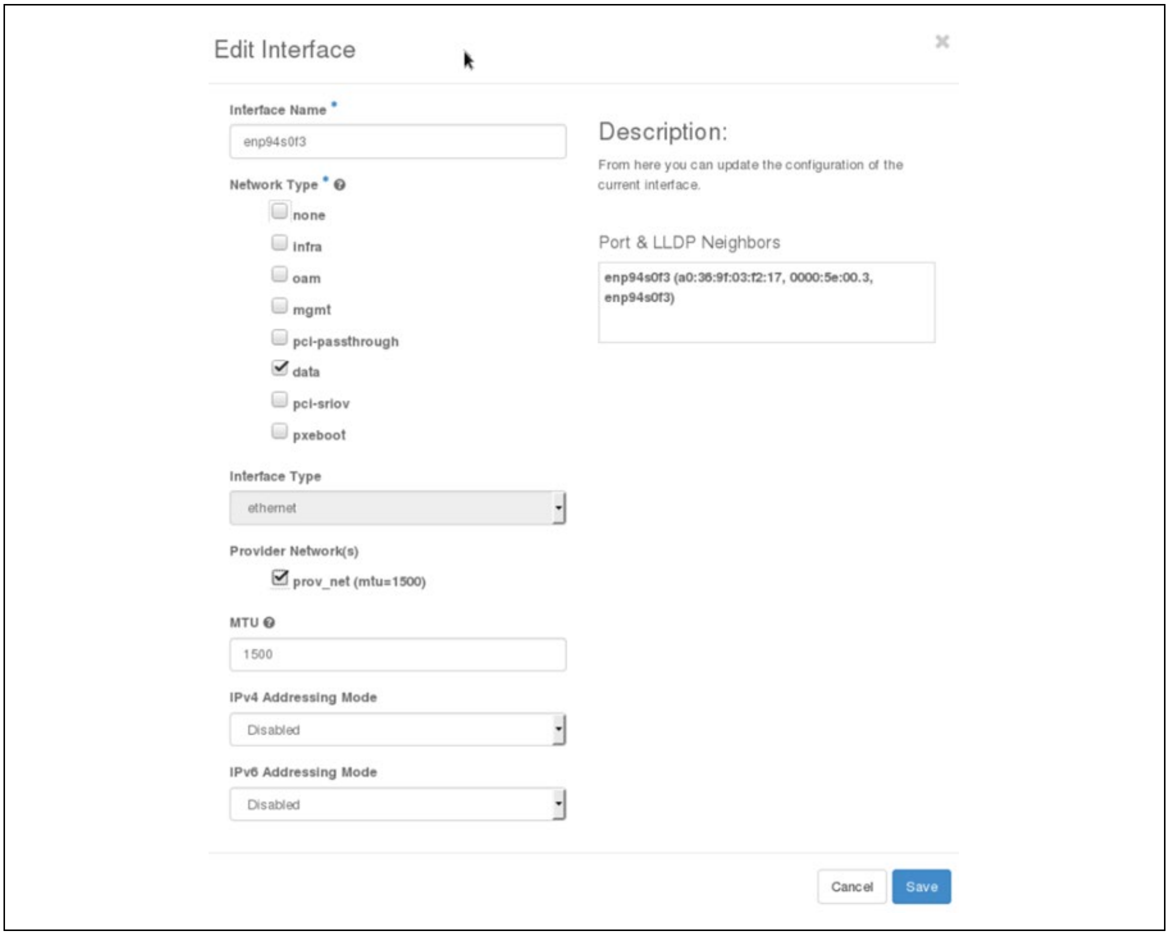


5. Edit the Interface to attach to a network.

   In this case, edit data interface (`enp94s0f3`) and attach created `prov-net`.

Note: Interfaces can only be edited in the Locked compute state.

Figure 15. Edit Interfaces

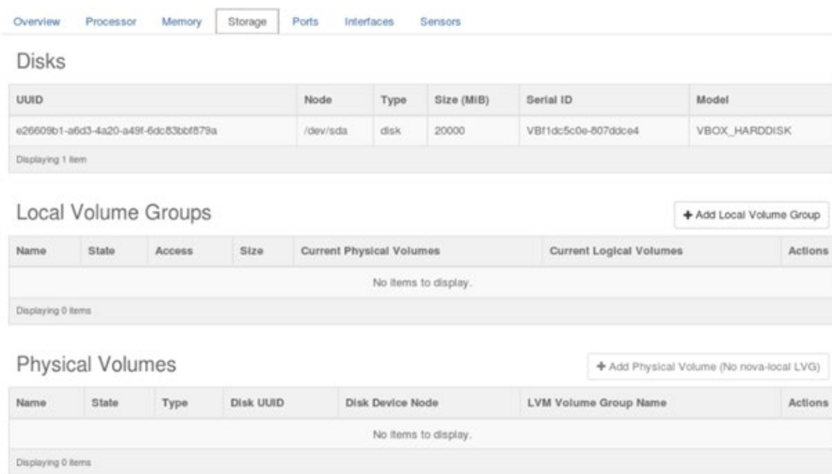


2.13 Local Volume Group

Add a local volume group to provide local storage.

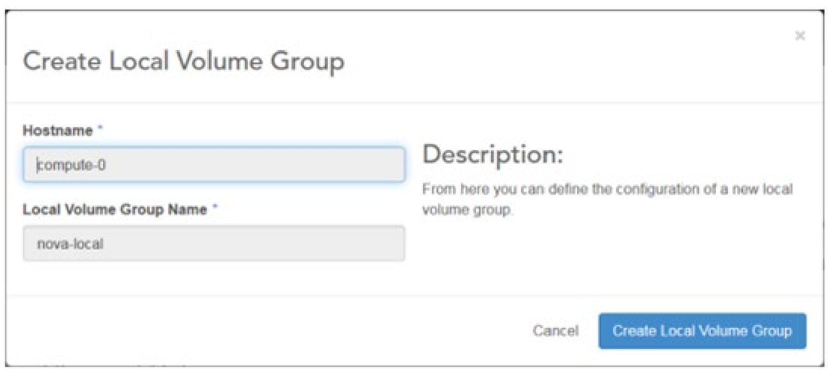
1. In the Hosts lists, click the tabs **compute-0 host->storage tab.**

Figure 16. Local Volume Group Local Storage



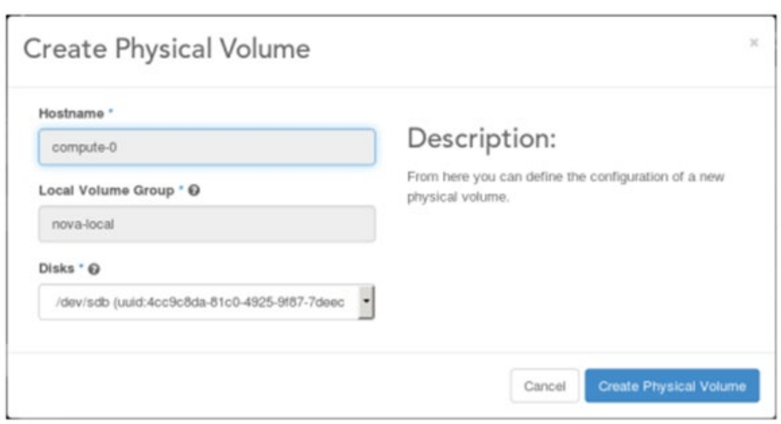
2. Add local volume group.

Figure 17. Create Local Volume Group



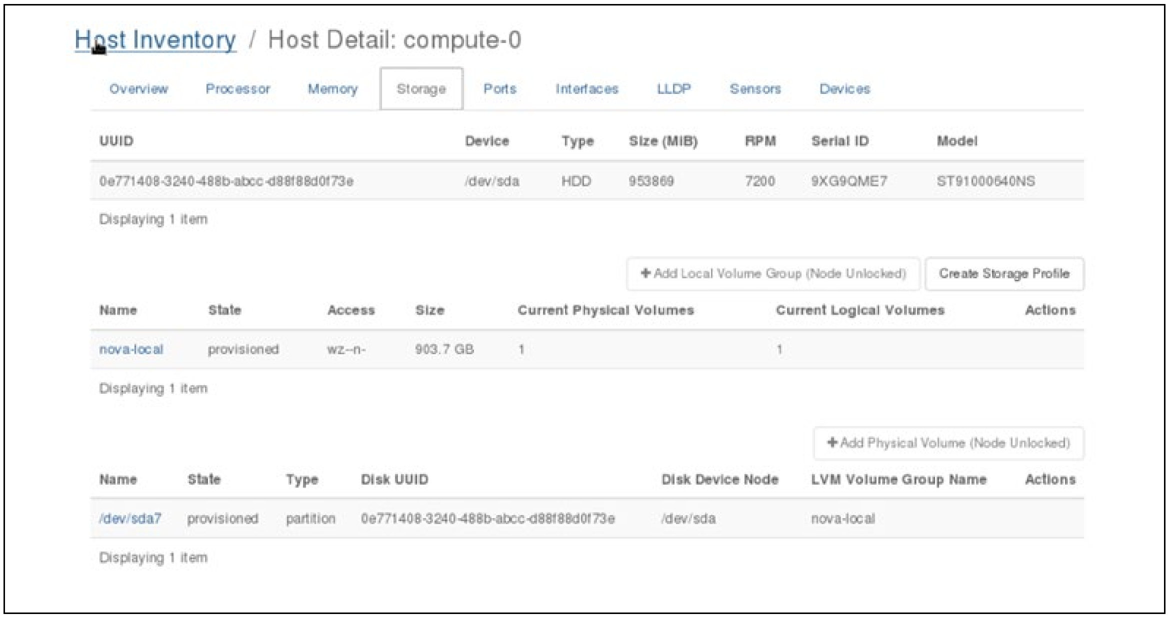
3. Add physical volume.

Figure 18. Create Physical Volume



4. Configure storage after unlocking node.

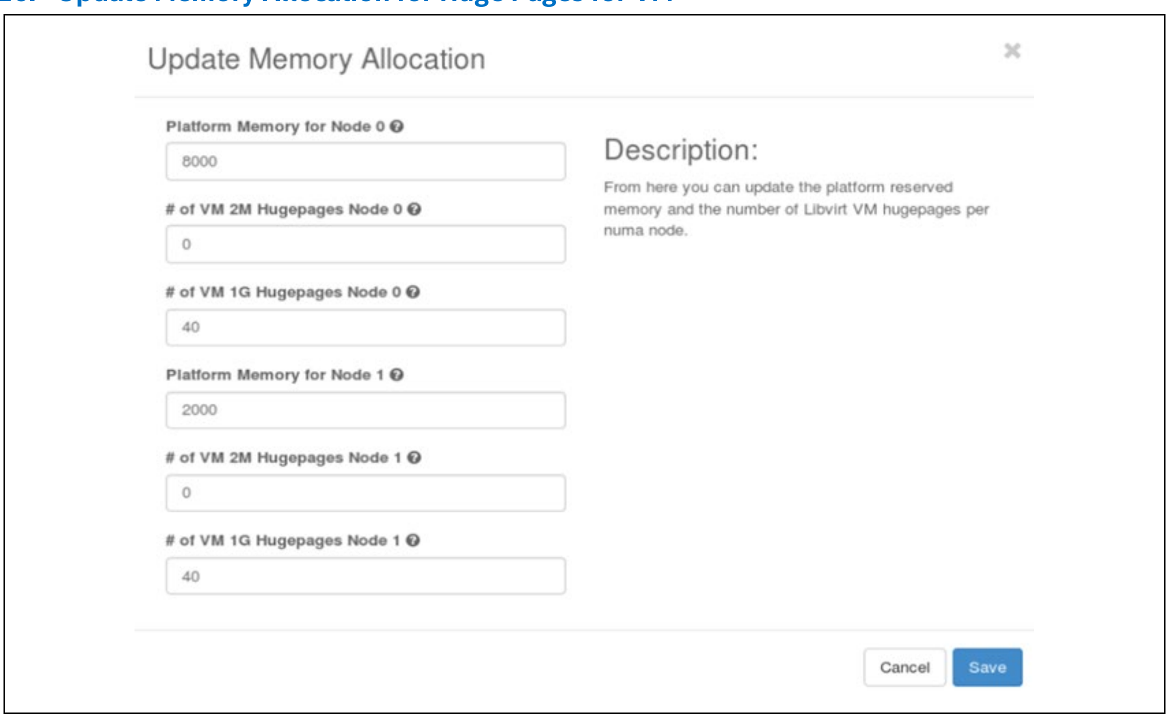
Figure 19. Configured Storage-Node Unlocked



2.13.1 Update Huge Pages for VM

In the Hosts lists, click the **compute-0 host** and **Memory > Update Memory**.

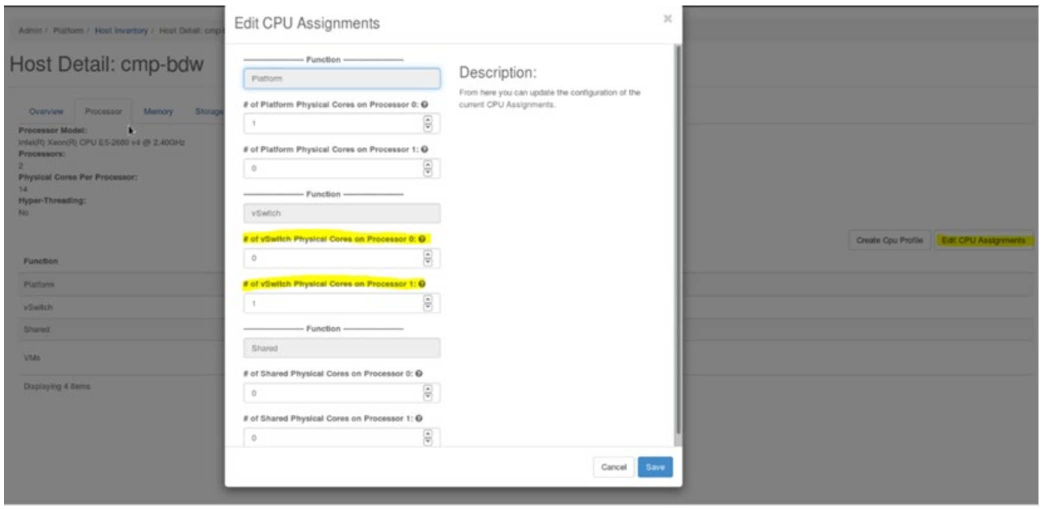
Figure 20. Update Memory Allocation for Huge Pages for VM



2.13.2 Change Vswitch Core Assignment (Optional)

In the host links, click the **compute-0 host -> processor** tab to access the **Edit CPU Assignments** screen.

Figure 21. Edit CPU Assignments Screen



2.13.3 Unlock the Host

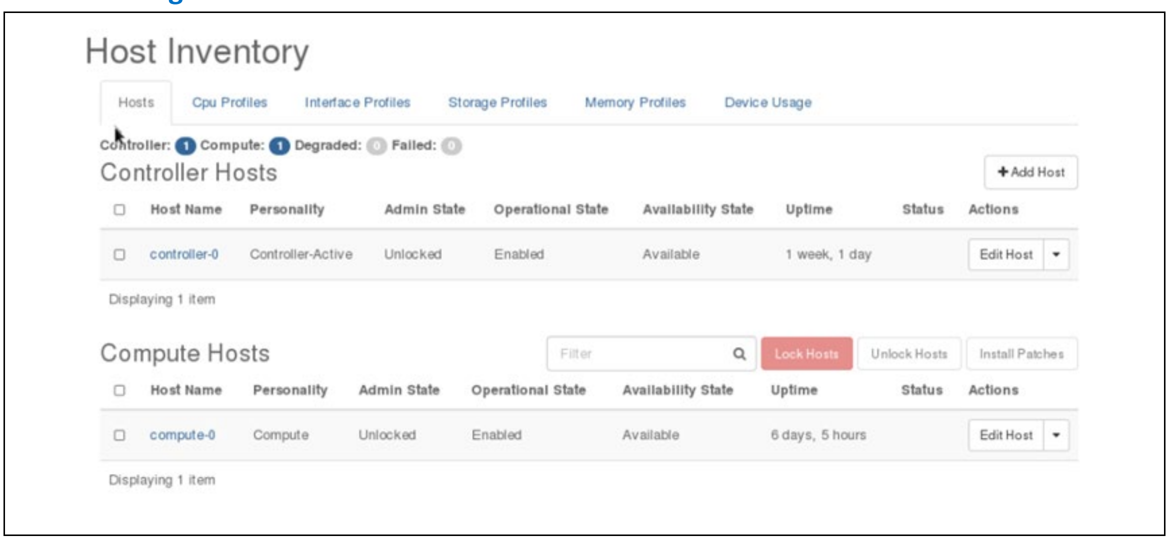
Unlock the host to make it available for use.

In the Hosts list, on the row associated with the node, click More > **Unlock Host.**

The host is rebooted, and its Availability State is reported as In-Test.

After a few minutes, it is reported as Unlocked, Enabled, and Available as shown in Figure 22.

Figure 22. Unlocking the Host

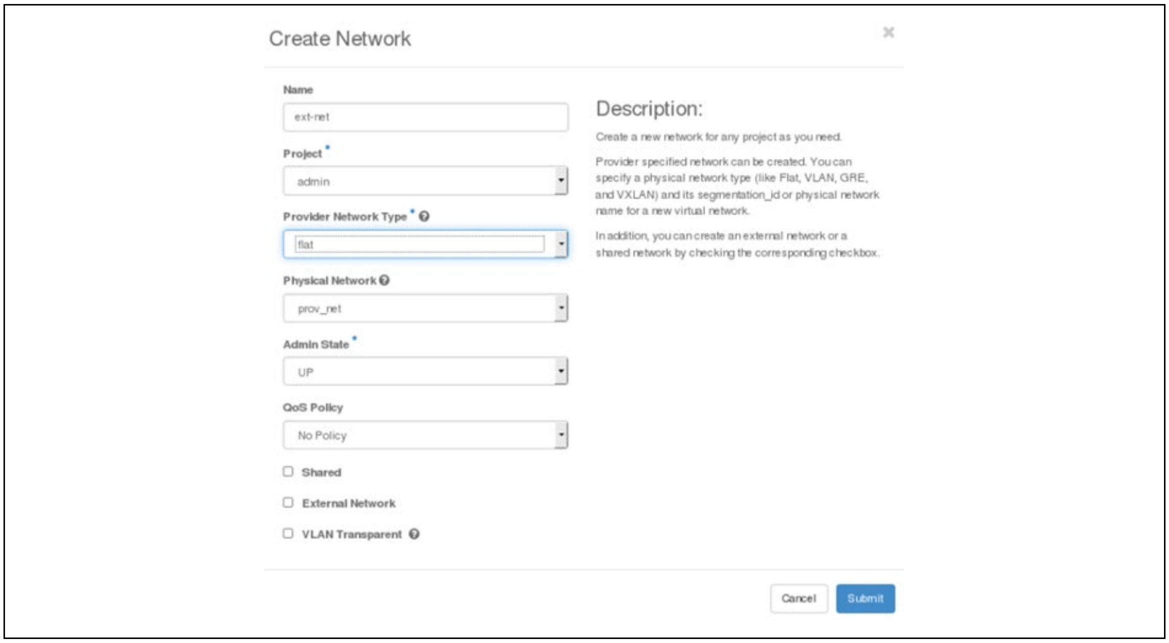


2.13.4 Creating Networks for VMs

For accessing VM, a proper network is required.

1. Select Admin > System > Networks > **Create Network**.

Figure 23. Create VM Network



2. Create a subnet and assign the network address and gateway IP, uncheck **Enabled DHCP** and check **System Managed Subnet**: **192.168.10.0/24**

Figure 24. Create Subnet

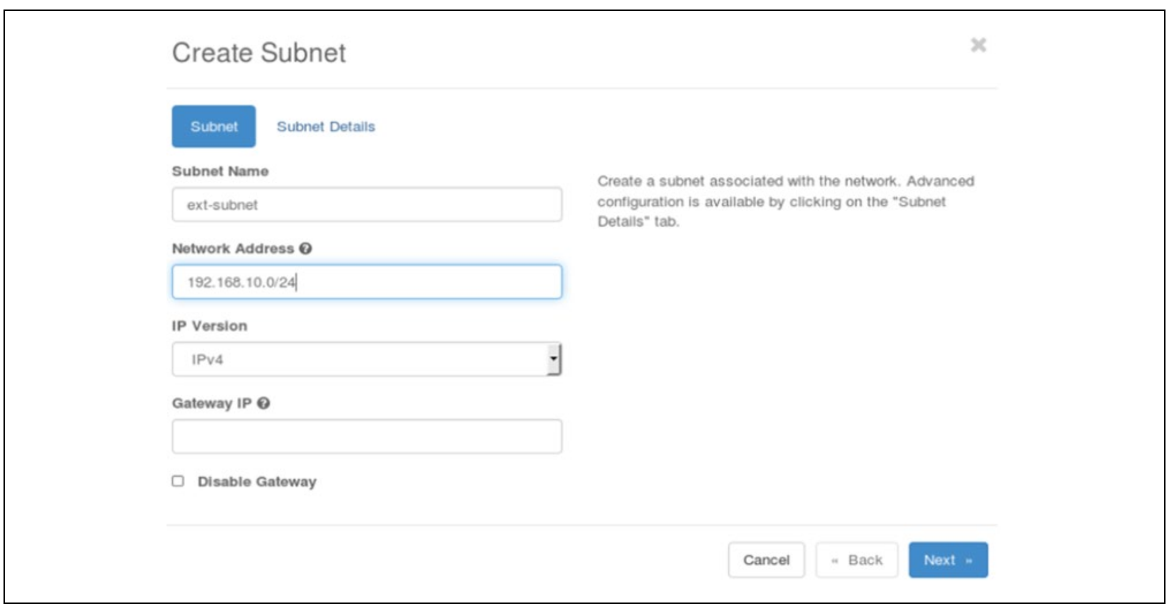
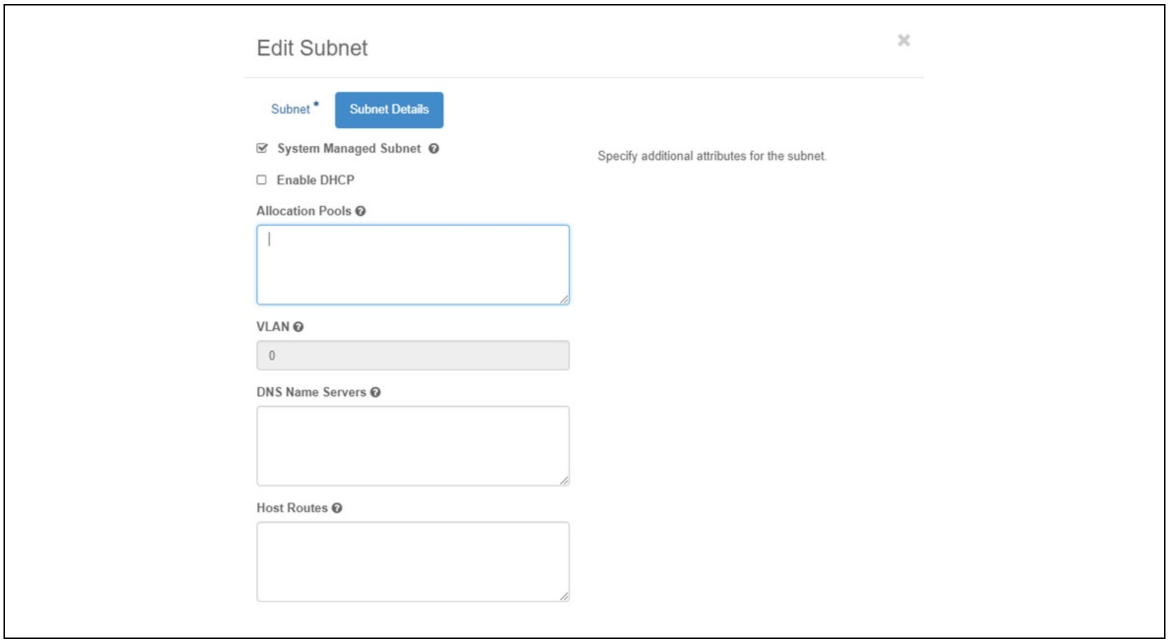


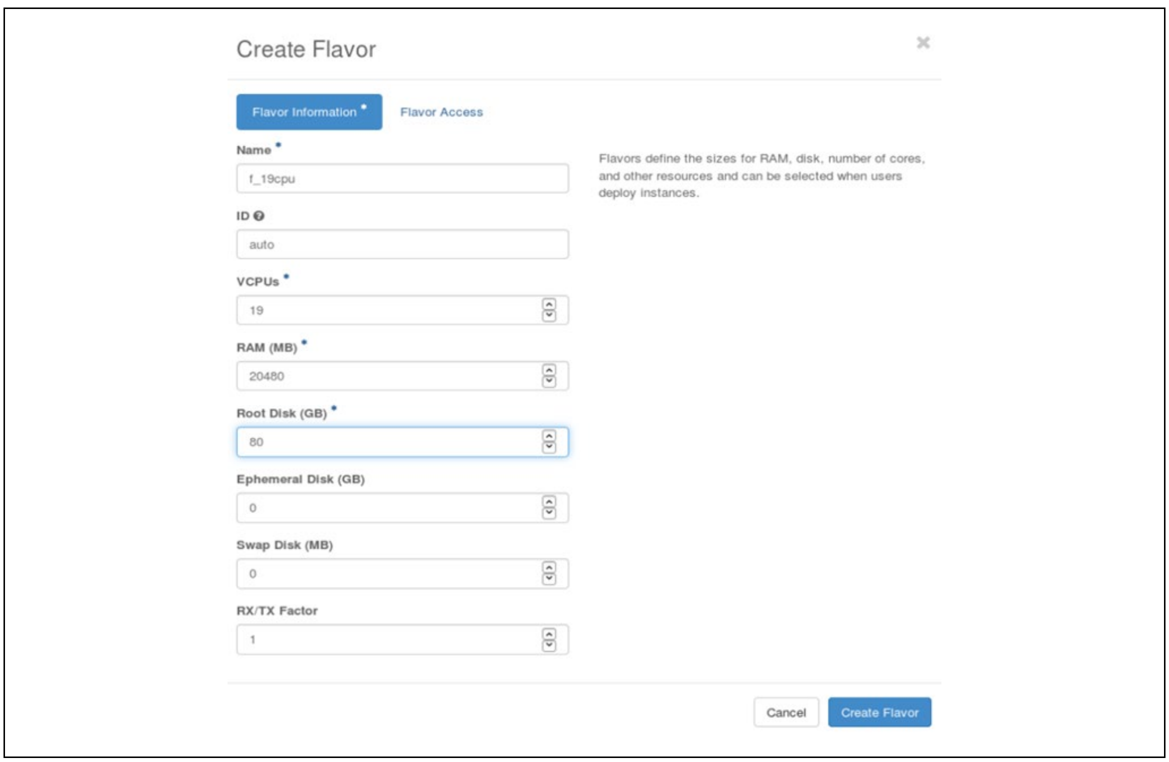
Figure 25. Edit Subnet



2.13.5 Preparing Processor Flavor

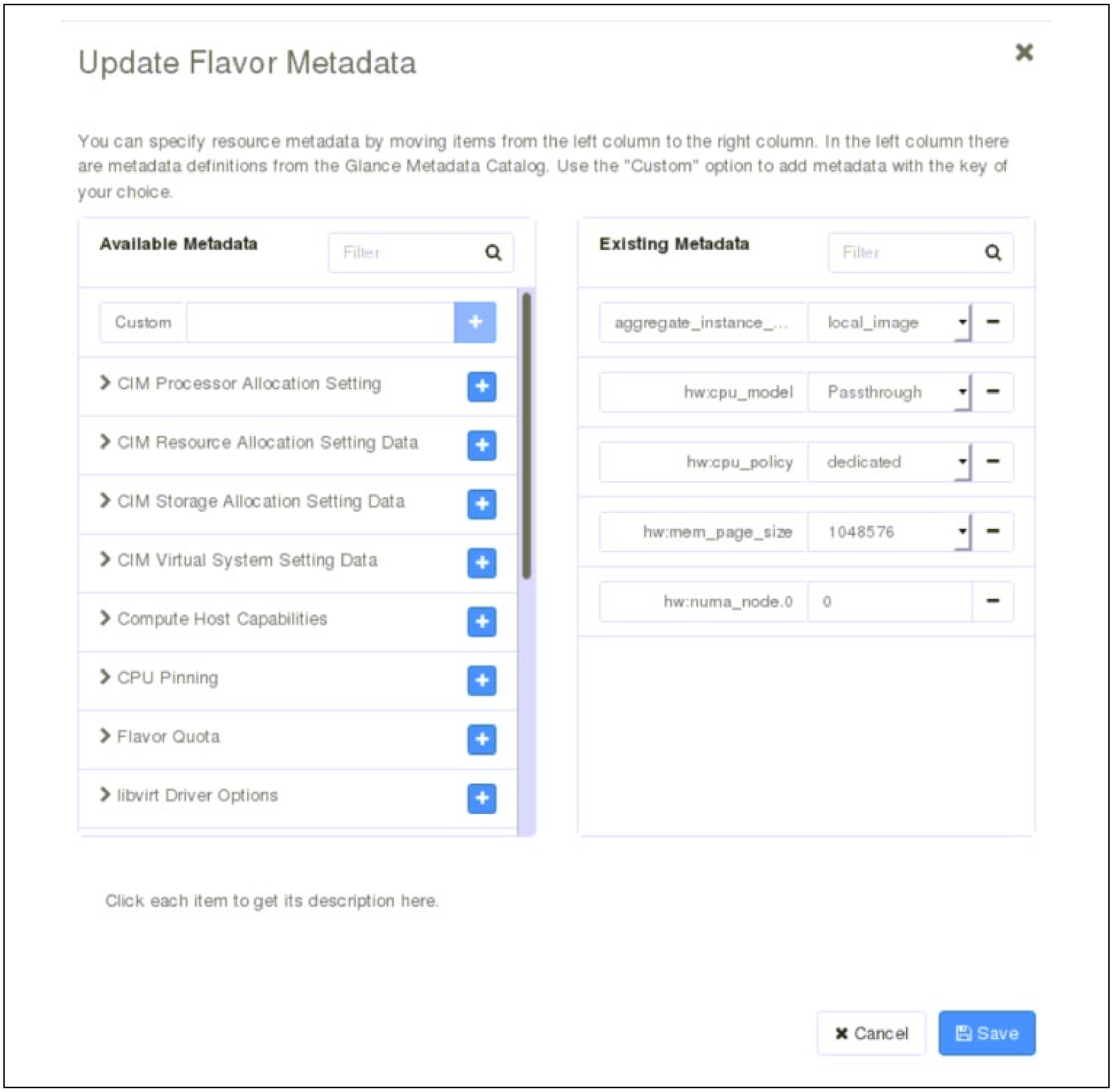
3. Navigate to **Admin > System > Flavors > Create Flavor**.

Figure 26. Create Flavor



4. Go to created **flavor > f_19cpu > Update Metadata (in Actions Column)** and create the following specifications.

Figure 27. Flavor Detail



2.14 Preparing VM Volume from an Image

The following provides instructions on how to import an image and create a VM volume.

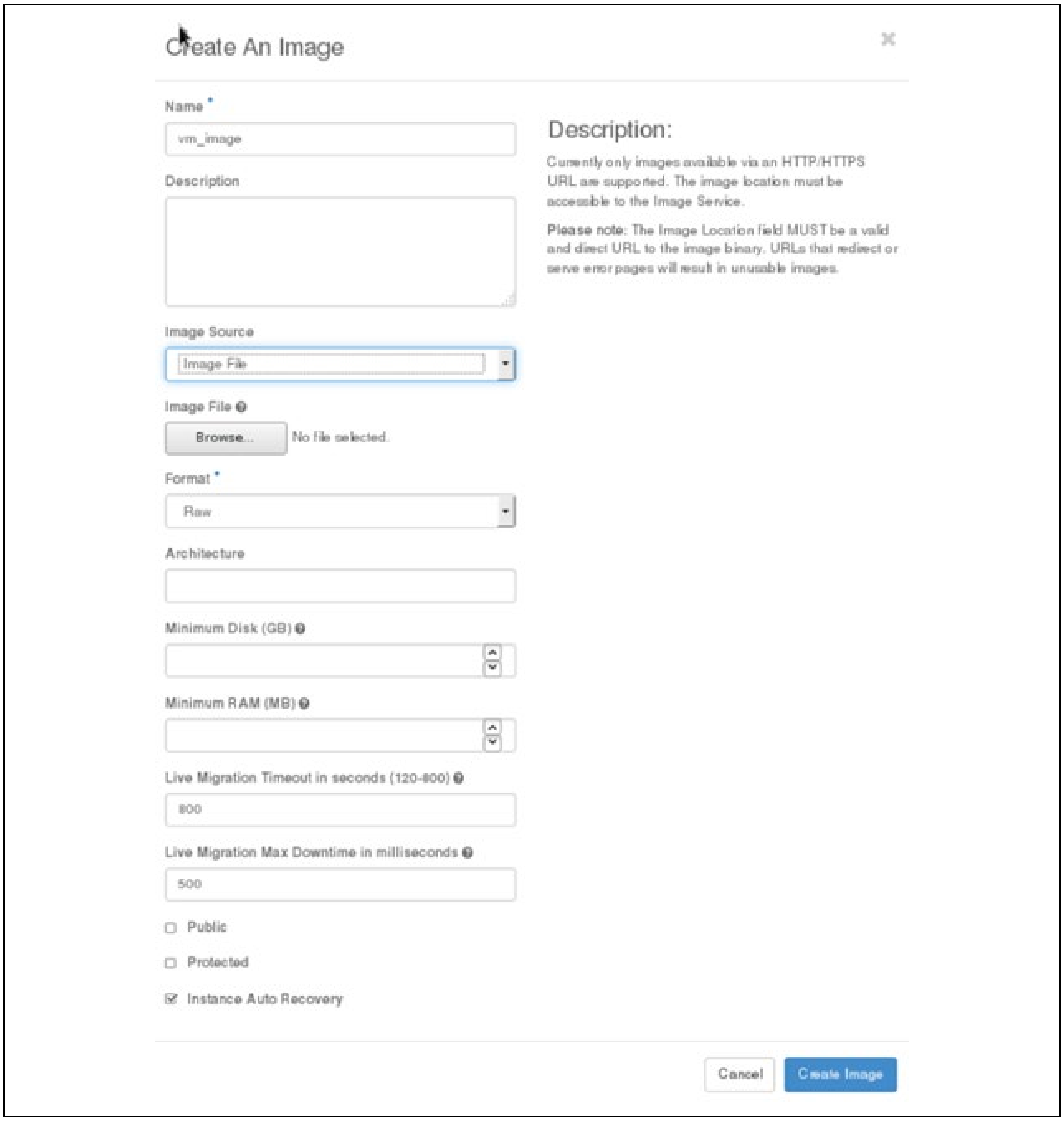
2.14.1 Importing an Image

1. First, import the VM image `tis-centos-guest-rt-basic.img`

   `tis-cent-os-rt-base-icc.qcow2` is a snapshot from `tis-centos-guest-rt-basic.img` with installed Intel® Parallel Studio 2017, update1.

2. Navigate to **Project > Compute > Images > Create Image** to create an image.

Figure 28. Create an Image



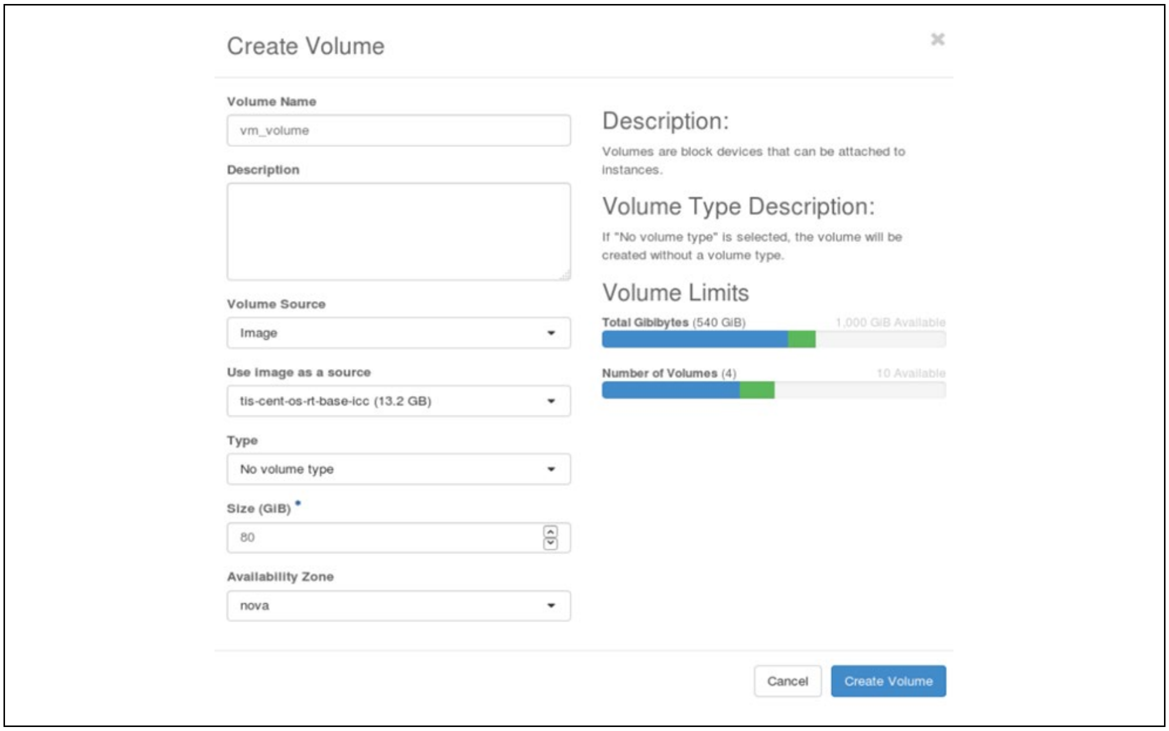
Note: Sometimes an error may occur with `custom qcow2 image` (such as importing a snapshot). If this occurs, try importing using command line on controller:

```
source /etc/nova/openrc
glance image-create --name "tis-cent-os-rt-base-icc" --visibility public --
disk-format=qcow2 --container-format=bare --file tis-cent-os-rt-base-
icc.qcow2 --progress
```

2.14.2 Creating VM Volume

Go to **Project > Compute > Volumes > Create Volume** to create a VM volume.

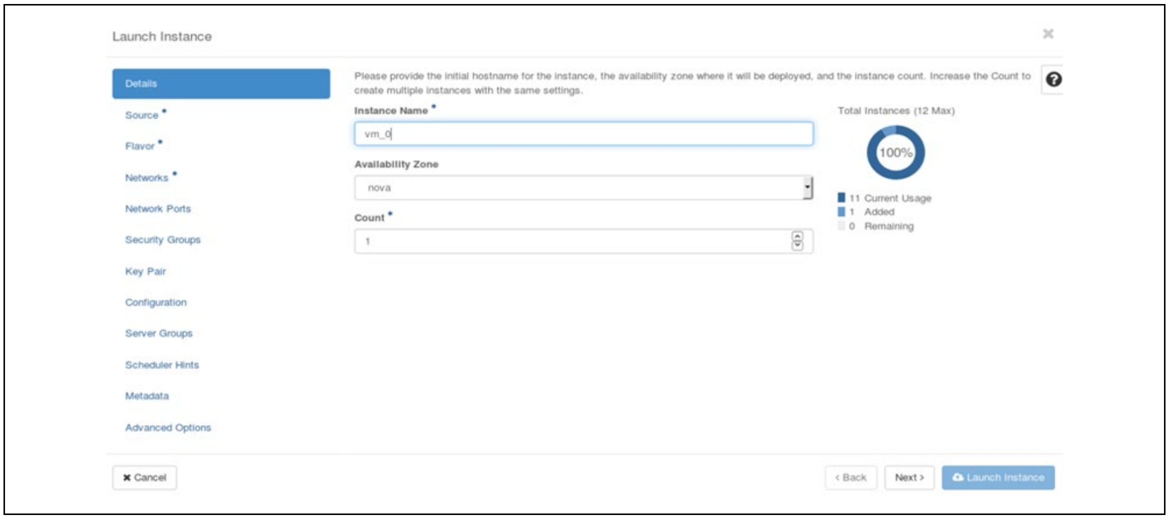
Figure 29. Create Volume



2.15 Creating VM Instance

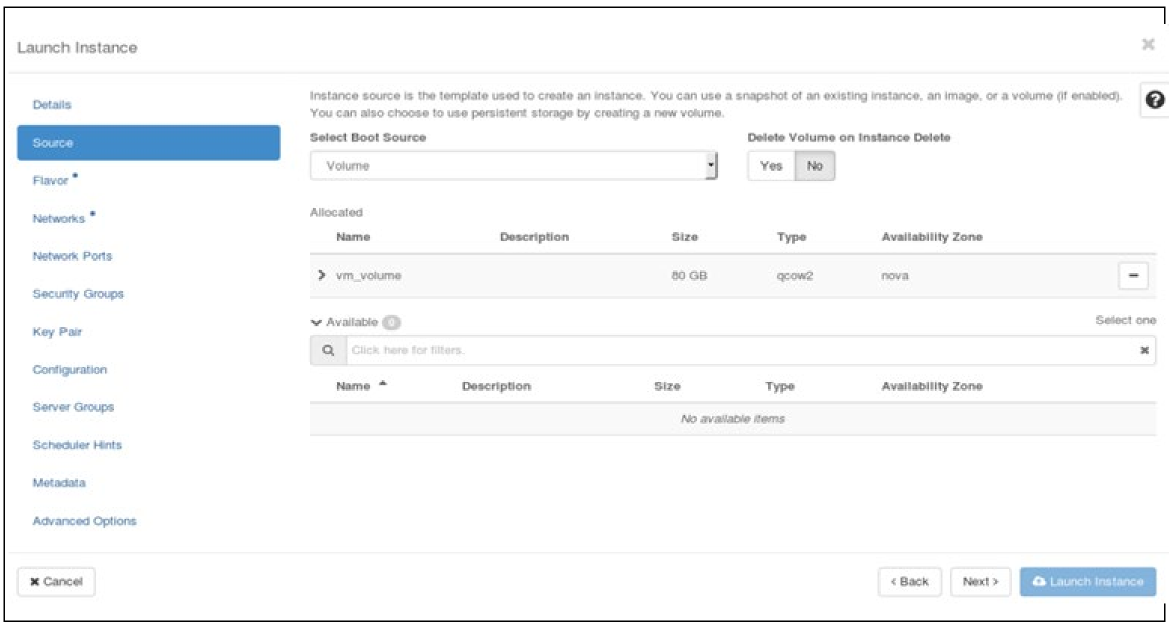
3. Select **Project > Compute > Instances > Launch Instance** and provide **Details > Instance Name.**

Figure 30. Launch Instance



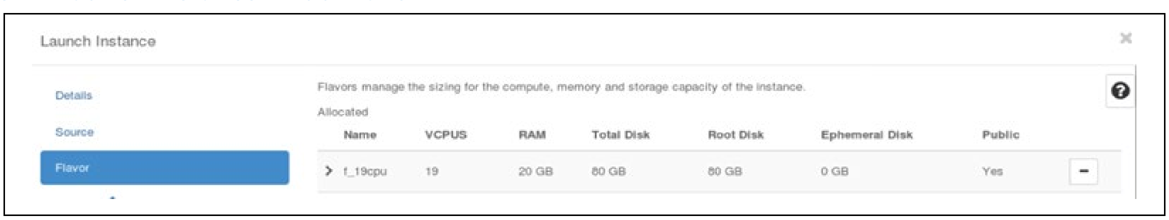
4. Add volume to **Source > Select Boot Source > Volume**.

Figure 31. Launch Instance – Add Volume



5. Add flavor.

Figure 32. Launch Instance – Add Flavor



6. Add network.

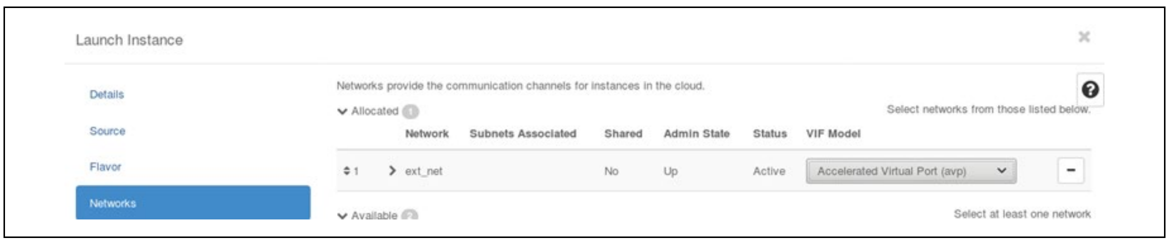
Figure 33. Launch Instance – Add Network

7. Launch instance.

2.15.1 AVP Interface (Optional)

To configure the AVP interface instead of `VirtIO`, select the following option when configuring the VM's Networks.

Figure 34. AVP Interface



2.16 Accessing VM

Each VM can be operated from Web UI in **VM > Console** tab. Also, ssh can be used.

Log in to VM using Web UI Console and configure network.

2.16.1 Guest Image Command Line

1. Modify VM command line to the following parameters (adjust number of processors to the setup):

```
vim /boot/extlinux.conf
rw root=LABEL=wrs_guest clock=pit console=tty0 console=ttyS0 biosdevname=0
net.ifnames=0 no_timer_check clocksource=tsc tsc=perfect
intel_pstate=disable selinux=0 enforcing=0 nmi_watchdog=0 softlockup_panic=0
isolcpus=1-15 nohz_full=0-15 idle=poll default_hugepagesz=1G hugepagesz=1G
hugepages=8 rcu_nocbs=1-15 kthread_cpus=0 irqaffinity=0 rcu_nocb_poll
initrd=initramfs.img BOOT_IMAGE=vmlinuz
```

2. Save and reboot VM.

2.17 NIC Pass-through (Optional)

1. Log in to the controller node and run:

```
source /etc/nova/openrc
```

2. Lock the compute node that you would like to configure the pass-through on system

```
host-lock --force <cmp_node_name>
```

3. Create the provider network:

```
neutron providernet-create prov_pt_0 --type flat
```

Assign the desired pass-through interface to provider network from the previous step:

```
system host-if-modify -m 1500 -n pt0 -p "prov_pt_0" -nt pci-passthrough
<cmp_node_name> <pt_iface_name>
```

4. Create the network and subnetwork (the subnetwork address does not matter. It is used to avoid WR errors about missing subnet configuration):

```
neutron net-create --provider:network_type=flat --
provider:physical_network=prov_pt_0 net_pt_0
neutron subnet-create --name subnet_pt_0 net_pt_0 192.168.1.0/24 --disable-
dhcp
```

5. Unlock the compute node and wait for the reboot to complete:

```
system host-unlock <cmp_node_name>
```

6. Create a VM:

```
export mng_net_UUID=`neutron net-list | grep <ext_net> | awk '{print $2}'`
export pt0_net_UUID=`neutron net-list | grep net_pt_0 | awk '{print $2}'`
export vol_UUID=`cinder list | grep <vol_name> | awk '{print $2}'`

nova boot --flavor=<flavor_name> \
--nic net-id=${mng_net_UUID} \
--nic net-id=${pt0_net_UUID},vif-model=pci-passthrough \
--boot-volume ${vol_UUID} <instance_name>
--availability-zone nova::<cmp_node_name>
```

Note: Omit availability-zone parameter if you are only using one compute node.

Example:

```
source /etc/nova/openrc
system host-lock --force compute-0
neutron providernet-create prov_pt_0 --type flat
system host-if-modify -m 1500 -n pt0 -p "prov_pt_0" -nt pci-passthrough
compute-0 enp134s0f0
neutron net-create --provider:network_type=flat --
provider:physical_network=prov_pt_0 net_pt_0
neutron subnet-create --name subnet_pt_0 net_pt_0 192.168.1.0/24 --disable-
dhcp

system host-unlock compute-0

export mng_net_UUID=`neutron net-list | grep ext-net | awk '{print $2}'`
export pt0_net_UUID=`neutron net-list | grep net_pt_0 | awk '{print $2}'`
export vol_UUID=`cinder list | grep fedora25-vol | awk '{print $2}'`


nova boot --flavor=f_13cpu \
--nic net-id=${mng_net_UUID} \
--nic net-id=${pt0_net_UUID},vif-model=pci-passthrough \
--boot-volume ${vol_UUID} fedora-vm \
--availability-zone nova::compute-0
```

2.18 SRIOV Pass-through (Optional)

1. Log into the controller node and run:

```
source /etc/nova/openrc
```

2. Lock the compute node that you would like to configure the SRIOV on system `host-lock --force <cmp_node_name>`

3. Create the provider network

```
neutron providernet-create prov_sriov_pt_0 --type flat
```

4. Assign the desired Single-Root Input-Output Virtualization (SRIOV) interface to provider network from the previous step.

```
system host-if-modify -m 1500 -n sriov_pt_0 -p "prov_sriov_pt_0" -nt none
<cmp_node_name> <sriov_iface_name>
system host-if-modify -m 1500 -n sriov_pt_0 -p "prov_sriov_pt_0" -nt pci-
sriov -N <max_vf_num> <cmp_node_name> <sriov_iface_name>
```

5. Create the network and subnetwork (the subnetwork address does not matter. It is used to avoid WR errors about missing subnet configuration).

```
neutron net-create --provider:network_type=flat --
provider:physical_network=prov_sriov_pt_0 net_sriov_0
neutron subnet-create --name subnet_sriov_0 net_sriov_0 192.168.1.0/24 --
disable-dhcp
```

6. Unlock the compute node and wait for the reboot to complete:

```
system host-unlock <cmp_node_name>
Create a VM:
export mng_net_UUID=`neutron net-list | grep <ext_net> | awk '{print $2}'`
export pt0_net_UUID=`neutron net-list | grep net_sriov_0 | awk '{print $2}'`

nova boot --flavor=<flavor_name> \
--nic net-id=${mng_net_UUID} \
--nic net-id=${pt0_net_UUID},vif-model=pci-sriov \
--availability-zone nova::<cmp_node_name>
--image <image_name> <vm_name>
```

Omit availability-zone parameter if you are using only one compute node.

Example:

```
source /etc/nova/openrc
system host-lock --force compute-0
neutron providernet-create prov_sriov_pt_0 --type flat
system host-if-modify -m 1500 -n sriov_pt_0 -p "prov_sriov_pt_0" -nt none
compute-0 eno3
system host-if-modify -m 1500 -n sriov_pt_0 -p "prov_sriov_pt_0" -nt pci-
sriov -N 2 cmp-enb eno3
neutron net-create --provider:network_type=flat --
provider:physical_network=prov_sriov_pt_0 net_sriov_0
neutron subnet-create --name subnet_sriov_0 net_sriov_0 192.168.1.0/24 --
disable-dhcp

system host-unlock compute-0

export mng_net_UUID=`neutron net-list | grep mng-net | awk '{print $2}'`
export pt0_net_UUID=`neutron net-list | grep net_sriov_0 | awk '{print $2}'`
```

```
nova boot --flavor=f_13cpu \
--nic net-id=${mng_net_UUID} \
--nic net-id=${pt0_net_UUID},vif-model=pci-sriov \
--availability-zone nova::compute-0
--image tis-cent-os-rt-base-icc vm_bdw_vf
```

§

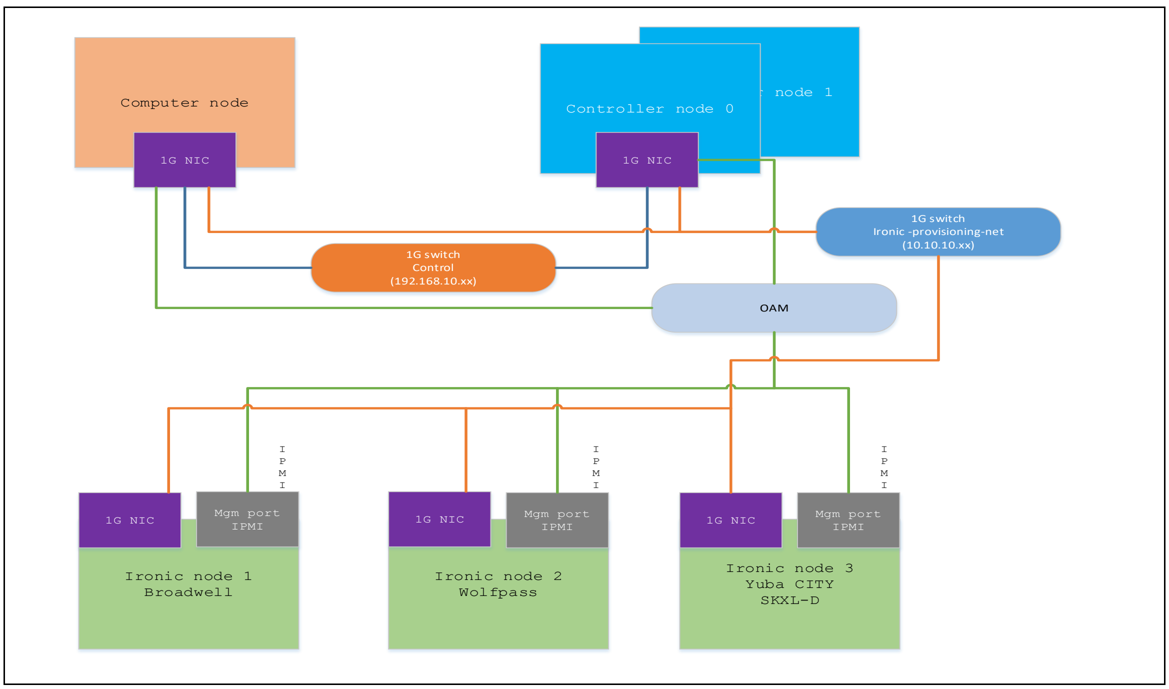
3.0 Enabling Ironic Compute Node

The Ironic is the bare metal platform provisioning service which enables the ability to boot bare metal machines instead of virtual machines. Ironic directly supports the configuration of physical machines within Open Stack, while Nova internally uses Ironic to support the launching of bare metal Nova instances.

3.1 System Topology

Figure 35 shows the topology of a typical Titanium Cloud deployment with ironic service enabled. Enabling ironic service requires a dedicated tenant network (typically on a flat provider network) to be provisioned via Neutron to deploy bare metal nodes, shown as Ironic-Provisioning-Net below. Controllers and bare metal nodes must have an interface on this tenant network; the bare metal nodes' interfaces must be untagged. This network is used for booting the bare metal nodes via network.

Figure 35. Topology with Ironic Service



3.2 Software Components

Table 5. Ironic Compute Node Software

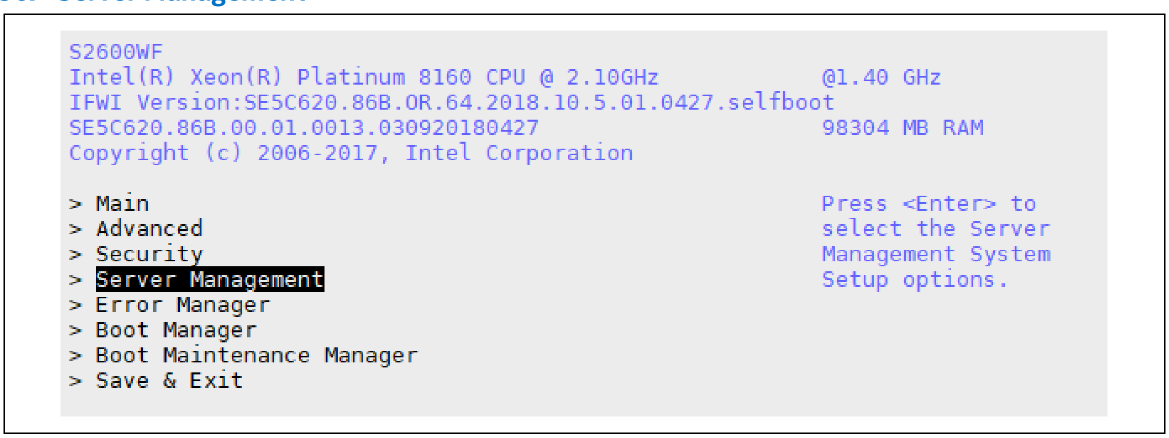
Name	Note Software
Titanium-Cloud-host-installer-18.03-b9.iso	TC_18.03_PATCH_0001.patch
	TC_18.03_PATCH_0002.patch
	TC_18.03_PATCH_0003.patch
	TC_18.03_PATCH_0004.patch
	TC_18.03_PATCH_0005.patch
	TC_18.03_PATCH_0006.patch
	TC_18.03_PATCH_0007.patch
	wrslicense.lic
coreos_production_pxe_image-oem.cpio.gz	https://tarballs.openstack.org/ironic-python-agent/coreos/files/coreos_production_pxe_image-oem.cpio.gz
coreos_production_pxe.vmlinuz	https://tarballs.openstack.org/ironic-python-agent/coreos/files/coreos_production_pxe.vmlinuz
CentOS-7-x86_64-GenericCloud-1708.qcow2	http://cloud.centos.org/centos/7/images/ and then select CentOS-7-x86_64-GenericCloud-1708.qcow2

3.3 BIOS Configuration for IPMI

The Intelligent Platform Management Interface (IPMI) provides management and monitoring capabilities for a host system.

1. Press F2 to enter **System Setup > Server Management** during booting.

Figure 36. Server Management

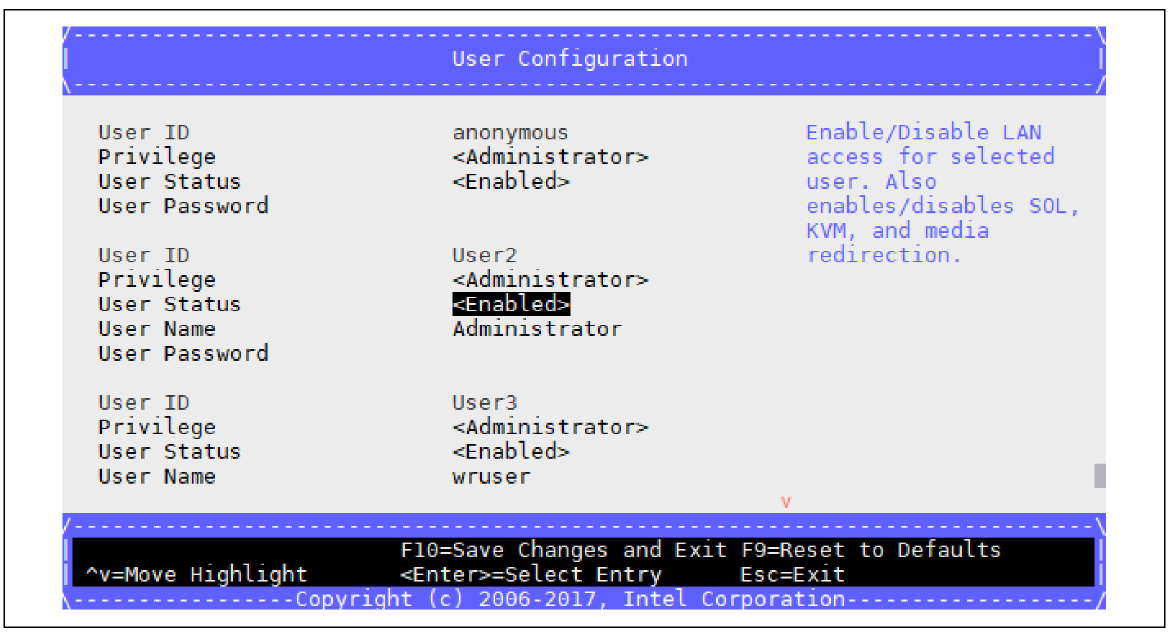


2. Change the IP source to Dynamic, to obtain an IP address via Dynamic Host Configuration Protocol (DHCP).

Figure 37. Obtain DHCP Address

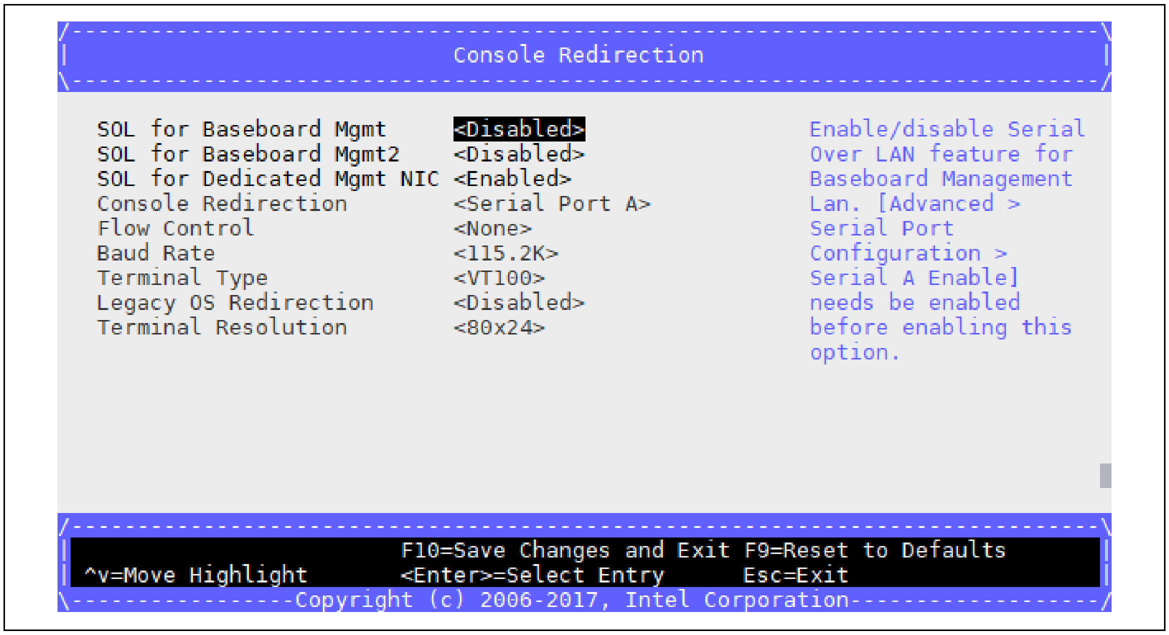
3. Copy the IP address, because you will need it in a later step.
4. Enable User ID 2 using user name **Administrator** and password **FDJHGHLM**.

Figure 38. Enable User ID 2



5. Set **SOL for Dedicated Mgmt NIC** to **Disabled**.

Figure 39. Disable Dedicated Mgmt NIC



6. Activate SOL:

```
#ipmitool -I lanplus -H <IP address from Step 2> -p 623 -U <user name> -P
"<password>" sol activate
```

7. Check if `ipmitool` works by logging into the controller node and running the following command:

```
#sudo ipmitool -I lanplus -H <IP address from Step 2> -p 623 -U Administrator -P
"FDJHGHLM"  sdr list
```

You should see a display similar to:

```
System Airflow   | 33 CFM            | ok
BB Lft Rear Temp | 37 degrees C      | ok
Riser 3 Temp     | 35 degrees C      | ok
BB P1 VR Temp    | 39 degrees C      | ok
Front Panel Temp | 33 degrees C      | ok
SSB Temp         | 48 degrees C      | ok
BB P2 VR Temp    | 41 degrees C      | ok
BB BMC Temp      | 39 degrees C      | ok
BB Rt Rear Temp  | 37 degrees C      | ok
……… .
```

That means the IPMI is successfully configured.

3.4 Preparing Related Images on Controller Node

Note: Refer to Section 3.2 for the location of the images to download in steps 1 and 2 below.

1. Download the Bare Metal user image file `CentOS-7-x86_64-GenericCloud-1708.qcow2`.
2. Download the deployment image files `coreos_production_pxe_image-oem.cpio` and `coreos_production_pxe.vmlinuz`.
3. Upload image to the controller node.

   First copy all images to the home folder in the controller node then upload the image to OpenStack Glance using the following commands:

   ```
   # source /etc/nova/openrc
   # glance image-create --name "ironic_BAREMETAL_USER_IMAGE" --visibility
   public --disk-format=raw --container-format=bare --file=/home/wrsroot/
   CentOS-7-x86_64-GenericCloud-1708.qcow2 --progress
   #glance image-create --name deploy-initrd --visibility public \
   --disk-format ari --container-format ari < coreos_production_pxe_image-
   oem.cpio.gz
   ```

4. Upload the Kernel module for the deployment image:

   ```
   #glance image-create --name deploy-vmlinuz --visibility public \
   --disk-format aki --container-format aki < coreos_production_pxe.vmlinuz
   ```

3.5 Configuring Standard Deployment with Ironic

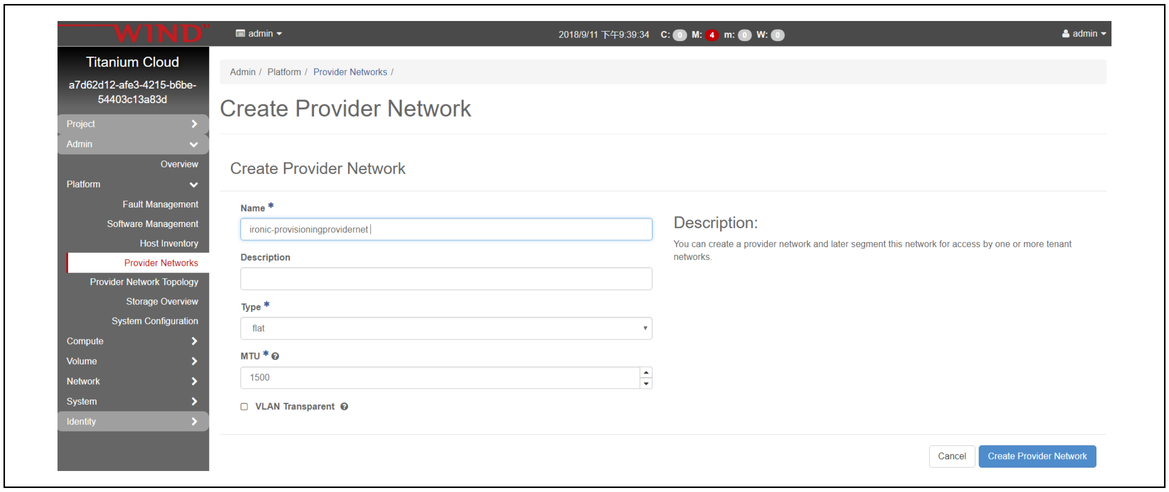
Before enabling the Ironic services, log in as Admin and follow the steps in this section, using either the Horizon graphic user interface or the command line interface (CLI).

3.5.1 Network Configuration

1. Configure a flat provider network for Ironic in Neutron via Horizon or the CLI;

   a. If using Horizon, select: **Admin > Platform > Provider Networks > Create Provider Network.**

Figure 40. Create Provider Network



b. If using the CLI, enter the command:

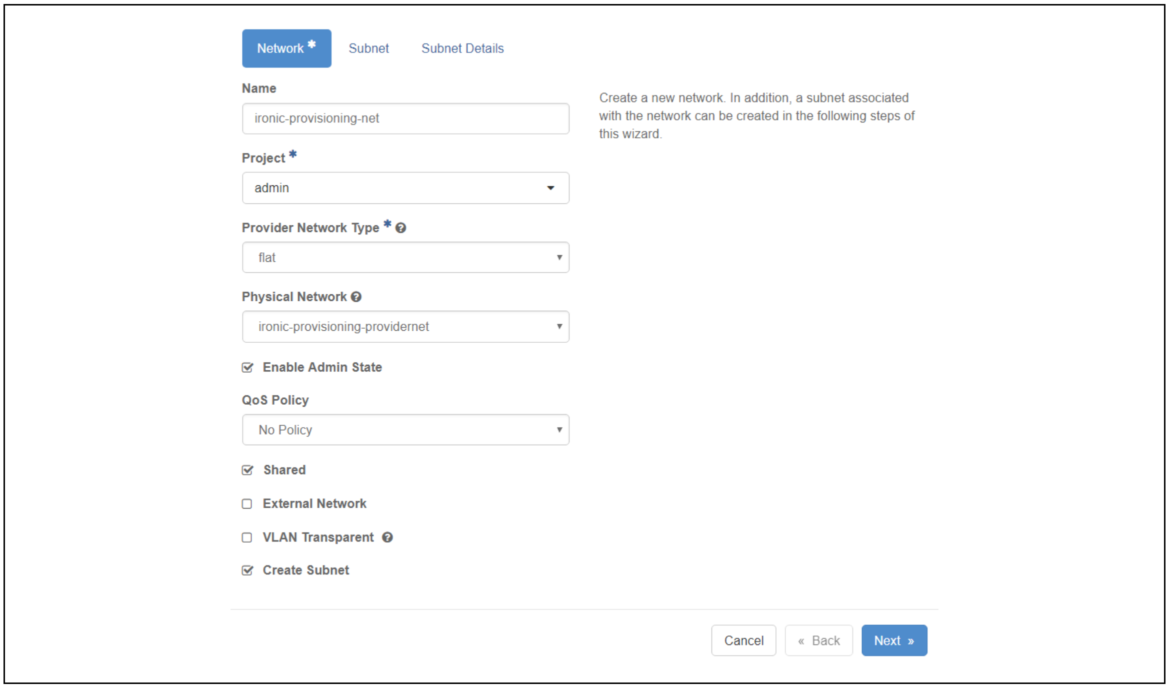
```
#openstack providernet create ironic-provisioningprovidernet –type flat
```

2. Configure an Ironic provisioning network with Neutron on the flat provider network just created (`ironic-provisioning-providernet`).

   a. If using Horizon, select: **Admin > Network > Networks > Create Network**.

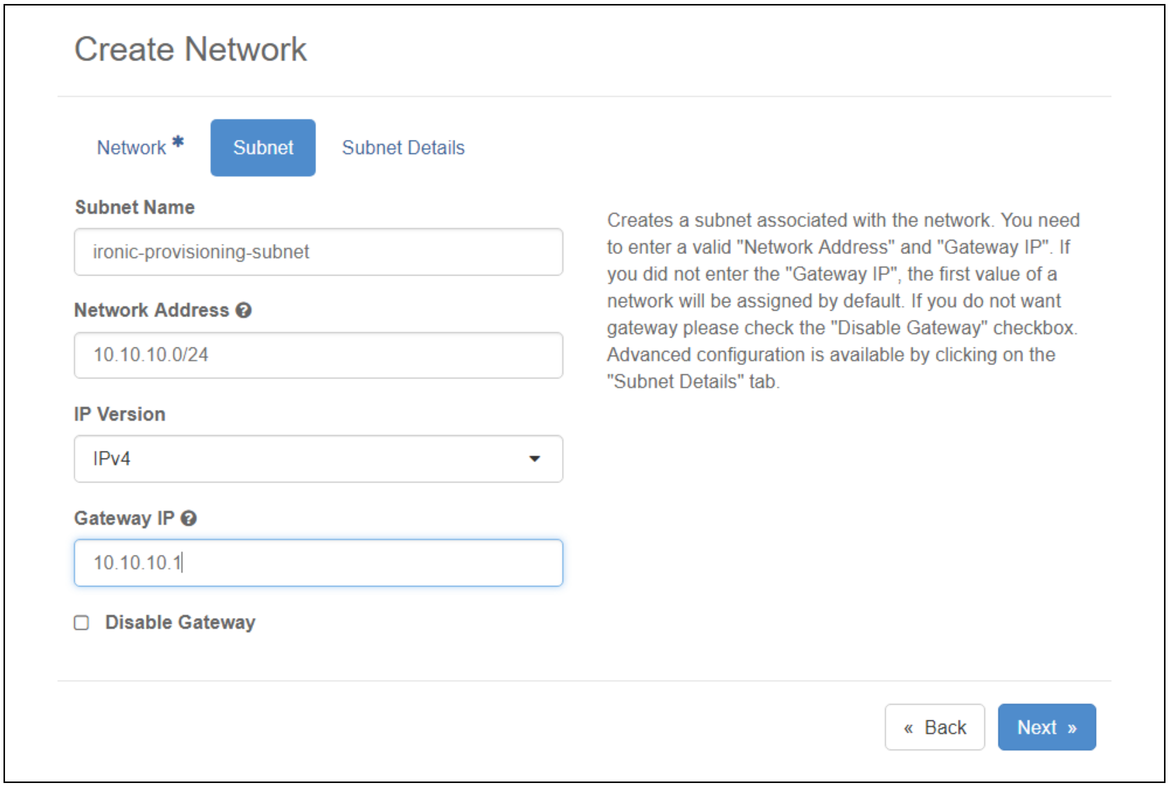
      1. Click **Network**.

Figure 41. Create Network



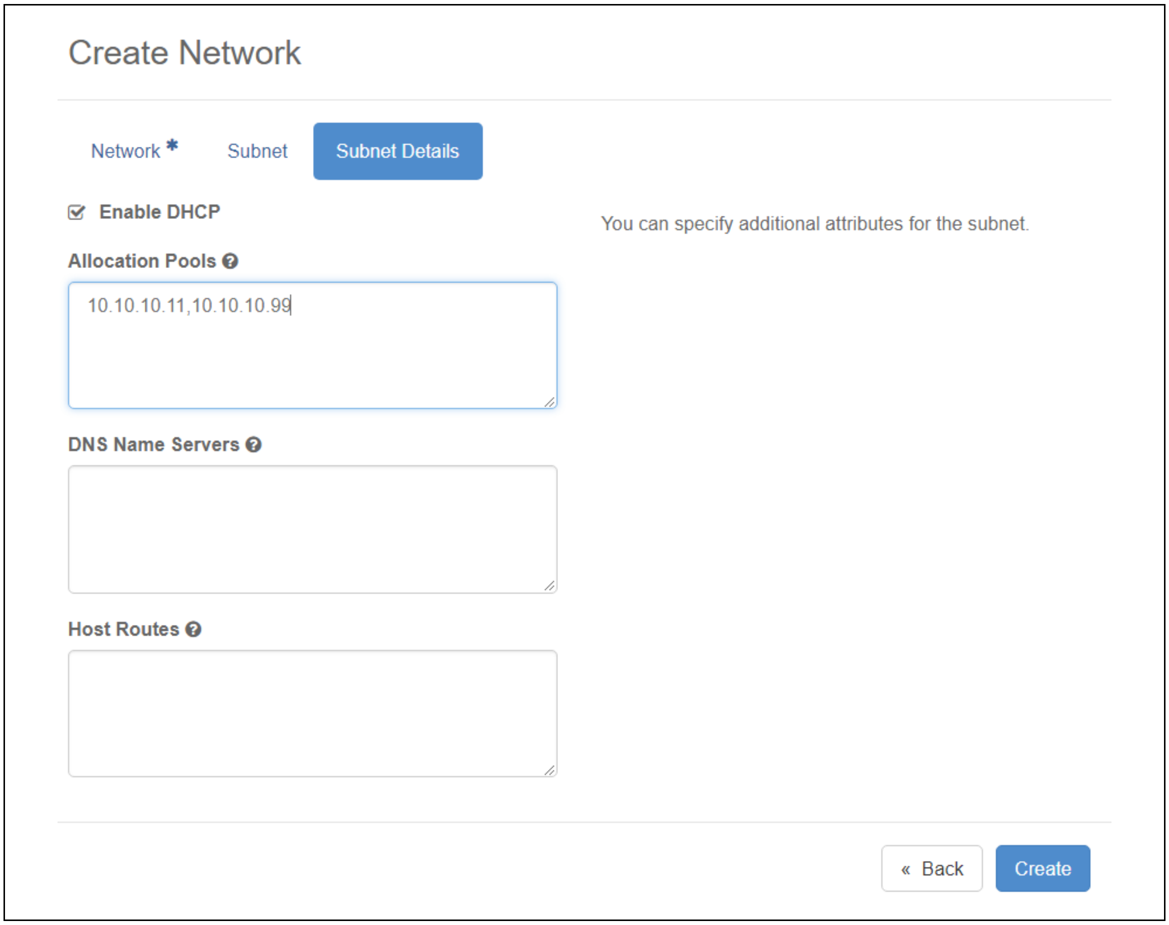
      2. Click **Subnet** and enter a Gateway IP address.

Figure 42. Create Subnet



3. Click **Subnet Details** and enter beginning and ending IP addresses for allocation pools.

Figure 43. Create Allocation Pools



b. If using the CLI, enter the commands:

```
#openstack network create --provider-physicalnetwork=ironic-provisioning-
providernet --provider-network-type=flat --shared ironic-provisioning-net

#openstack subnet create --name ironic-provisioning-subnet
ironic-provisioning-net  --gateway 10.10.10.1 --allocation-pool
start=10.10.10.11,end=10.10.10.99 10.10.10.0/2
```

3. Configure the controller interfaces on both controller-0 and controller-1.

   a. If using Horizon, select **Admin > Platform > Host Inventory >** <controller-hostname> **> Interfaces**, change the interface type from **none** to **control** and specify a static IP address for this interface.

Figure 44. Configuring a Controller Interface

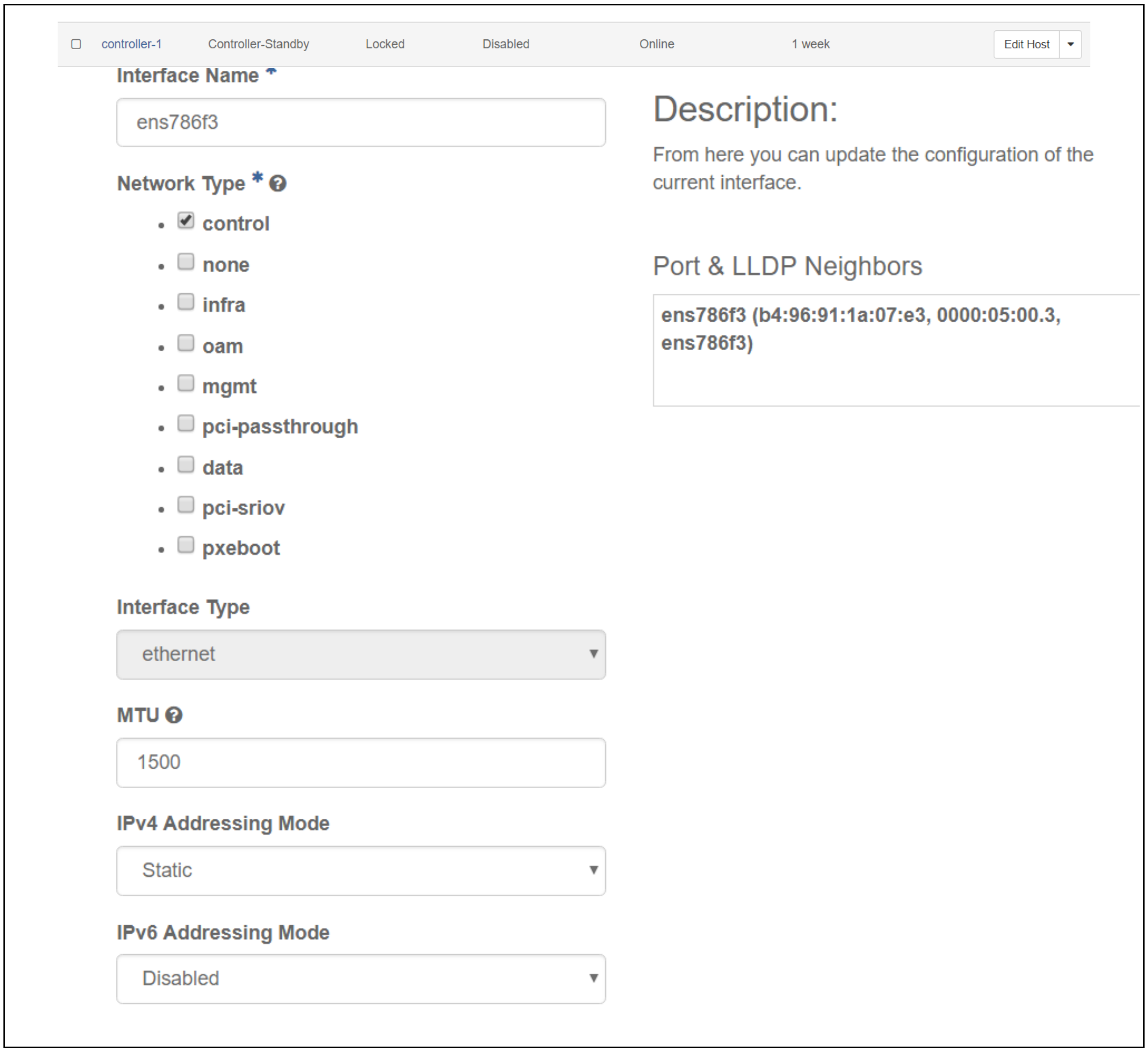
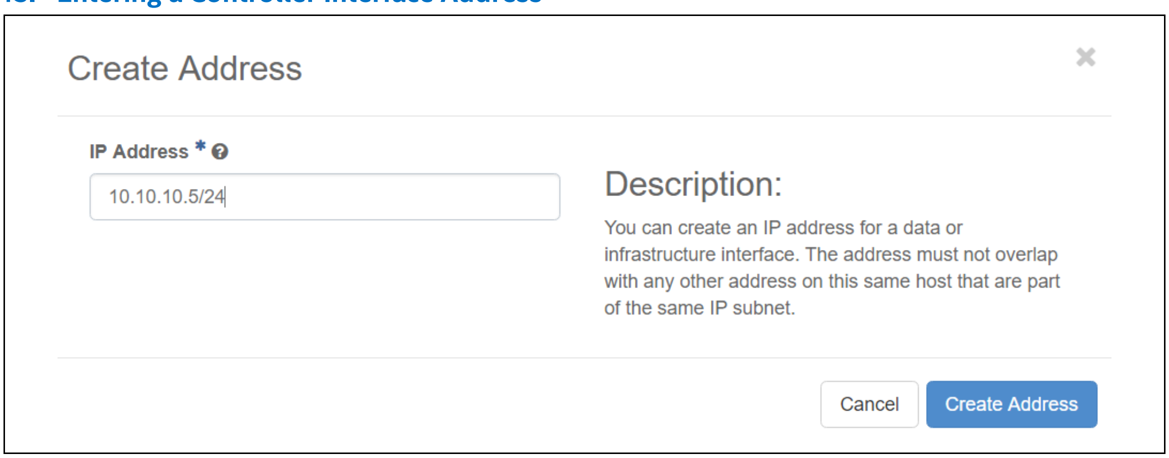


Figure 45. Entering a Controller Interface Address



b. If using the CLI, enter the commands:

```
#system host-if-modify -n <ifname> -m <mtu> -nt control controller-0
<ifname> --ipv4-mode=ip4_mode
#system host-addr-add controller-0 <ifname> ipv4 10.10.10.5 prefix 24
```

c. Using the CLI, it is also possible to specify an IP address for the interface outside of the above allocation-pool for bare metal nodes. (This is not supported with Horizon.)

1. Enter the command:

```
#system host-addr-add controller-0 <ifname> ipv4 10.10.10.5 prefix 24
```

2. Then unlock the controller host using the command:

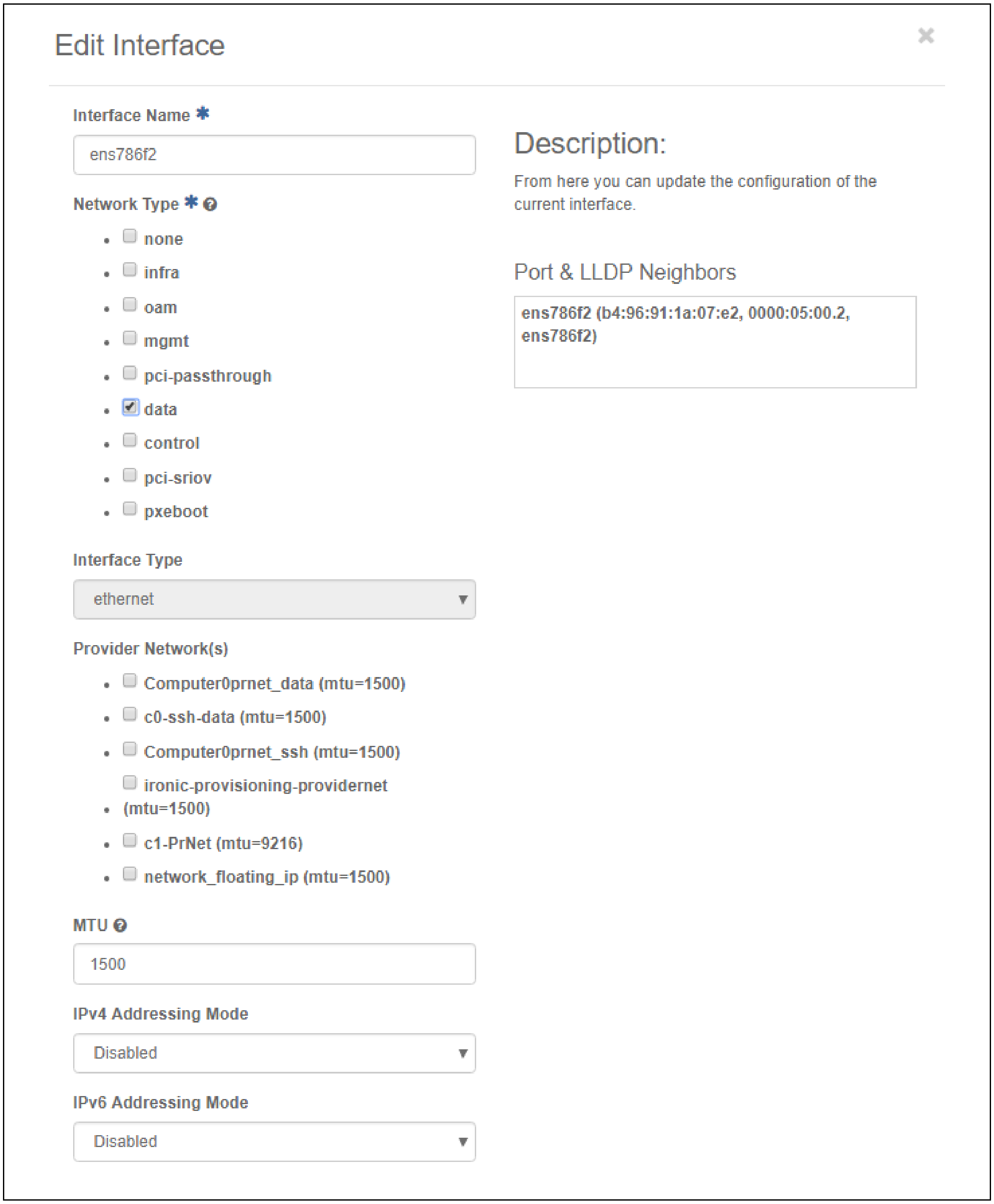
```
#system host-unlock controller-0:
```

4. Add the new provider network, `ironic-provisioning-providernet`.

a. If using Horizon:

1. Select Admin > Platform > Host Inventory > <compute-hostname>
> Interfaces
2. Select `ironic-provisioning-providernet` on the list of provider networks associated with the data interface.

Figure 46. Add New Provider Network



b. If using CLI:

1. Enter the command

```
#System host-if-modify -n <ifname> -p pnet-0,pnet-
1,ironicprovisioning-providernet
```

2. Unlock the compute host after changing the interface type.

3.5.2 Ironic Deployment Images

To enable Ironic deployment on a bare metal server, two types of images are used:

- Deployment images, which are used by Ironic to prepare for user image deployment
- User images, which are installed on the server to be used by the end user.

Note: General use single deployment image is required; different bare metal server hardware may require a unique deployment image for each different server.

3.5.3 Enabling Ironic Service

To enable Ironic services using the CLI:

1. Enter the following commands:

```
#system service-parameter-add ironic
neutron\provisioning_network=<IRONIC_PROVISIONING_NET_UUID>
#system service-parameter-add ironic pxe tftp_server=<TFTP_SERVER_IP>
#system service-parameter-add ironic pxe netmask=24
#system service-parameter-add ironic pxe
\controller_0_if=<CONTROLLER_0_INTERFACE>
#system service-parameter-add ironic pxe
controller_1_if=<CONTROLLER_1_INTERFACE>
```

   Where:

   - The `<CONTROLLER_#_INTERFACE> use ifconfig` command shows the interface name.
   - Ironic configures a Tftp server on the controller interface(s) connected to the ironic provisioning-network for PXE booting images on the bare metal.
   - The <TFTP_SERVER_IP> will be the floating IP Address of the Ironic tftp server.
   - Ensure that the <TFTP_SERVER_IP> is in the ironic-provisioning-subnet, but outside the allocation pool assigned for DHCP purposes within the ironic-provisioning-subnet.

2. Run the following command to enable ironic service after system parameters have been configured:

```
#system service-enable ironic
```

   When Ironic is enabled, both the controller nodes raise the Configuration Out of Date warning. To complete the setup, both the controller nodes need to be locked and unlocked. Once both the nodes have been unlocked, Ironic has worked.

3.5.4 Configuring Ironic Nodes

To configure Ironic nodes using the CLI:

1. Set environment variables for the Ironic CLI clients.

Export the environment variables to use the latest versions of the command line clients, or alternatively use the `--ironic-apiversion` <latest> and `-os-baremetal-api-version` <latest> with the respective clients.

```
#export OS_BAREMETAL_API_VERSION=1.32
#export IRONIC_API_VERSION=1.32
```

2. Enroll an Ironic node by issuing the node-create and node-update Ironic commands.

```
#ironic --ironic-api-version latest node-create -d pxe_ipmitool_socat -i
ipmi_address=<IPMI_IP> -i ipmi_username=<USERNAME> -i
ipmi_password=<PASSWORD>
#ironic --ironic-api-version latest node-update $NODE_ID add name=<example>
#ironic --ironic-api-version latest node-update $NODE_ID add driver_info/
deploy_kernel=$DEPLOY_VMLINUZ_UUID
driver_info/deploy_ramdisk=$DEPLOY_INITRD_UUID
```

Note: To display the values of DEPLOY_VMLINUZ_UUID and DEPLOY_INITRD_UUID, run the command #openstack image list.

```
#ironic --ironic-api-version latest node-update $NODE_ID add
properties/cpu_arch=ARCH properties/cpus=$CPU
properties/capabilities="boot_option:local" properties/ memory_mb=$RAM_MB
properties/local_gb=$DISK_GB
#ironic --ironic-api-version latest node-update $NODE_ID add driver_info/
ipmi_terminal_port=<port>
$ ironic node-set-console-mode $NODE_ID true
```

Where `$ARCH`= `x86_64`, and where the default port number for IPMI is 623.

IPMI uses UDP, not TCP.

After initially creating the Ironic node, the node's state should be in an enrolled state.

1. Create a PXE enabled port for the ironic node specifying the MAC ADDRESS of the node's port that is on the ironic-provisioning-net.

```
# ironic --ironic-api-version latest port-create -n ${NODE_ID} -a
${MAC_ADDRESS} -- pxe-enabled True
```

Where:

   - `$NODE_ID` = Ironic node UUID.
   - `${MAC_ADDRESS}` is the MAC address of the Bare Metal node's port whose associated switch port has its untagged VLAN configured to be the ironic-provisioning-net VLAN.

2. Provision the Ironic node from enroll to manageable, and from manageable to available state.

```
#ironic --ironic-api-version latest node-set-provision-state $NODE_ID manage
```

3. After Ironic node changes to manageable state, then use IPMI tool to activate and retrieve the power state of the node.

4. Then run the command:

```
#ironic --ironic-api-version latest node-set-provision-state $NODE_ID
provide
```

5. To check if the bare metal host has been fully registered, use the commands `openstack hypervisor list` and `openstack hypervisor show` to ensure the `local_gb`, `memory_mb`, and `vcpus` in the list of hypervisor attributes matches the properties of the Ironic node when it was enrolled.

   For example:

```
#openstack hypervisor list
+----+--------------------------------------+-----------------+
| ID | Hypervisor Hostname                  | Hypervisor Type |
+----+--------------------------------------+-----------------+
| 1  | compute-0                            | QEMU            |
| 2  | a515bcd8-7007-494e-9fb2-e360dcca8a61 | ironic          |
+----+--------------------------------------+-----------------+
+---------------------------+-------+
| Host IP                   | State |
+---------------------------+-------+
| abcd::2c8b:1973:4064:75fd | up    |
| 127.0.0.1                 | up    |
+---------------------------+-------+
#openstack hypervisor show a515bcd8-7007-494e-9fb2-e360dcca8a61
+-------------------------+--------------------------------------+
| Field                   | Value                                |
+-------------------------+--------------------------------------+
| aggregates              | []                                   |
| cpu_info                |                                      |
| current_workload        | 0                                    |
| disk_available_least    | 0                                    |
| free_disk_gb            | 100                                  |
| free_ram_mb             | 8192                                 |
| host_ip                 | 127.0.0.1                            |
| hypervisor_hostname     | a515bcd8-7007-494e-9fb2-e360dcca8a61 |
| hypervisor_type         | ironic                               |
| hypervisor_version      | 1                                    |
| id                      | 2                                    |
| l3_cache_by_node        | None                                 |
| l3_cache_granularity    | None                                 |
| l3_cache_used_by_node   | None                                 |
| l3_closids              | None                                 |
| l3_closids_used         | 0                                    |
| local_gb                | 100                                  |
| local_gb_used           | 0                                    |
| memory_mb               | 8192                                 |
| memory_mb_by_node       | None                                 |
| memory_mb_used          | 0                                    |
| memory_mb_used_by_node  | None                                 |
| running_vms             | 1                                    |
| service_host            | controller                           |
| service_id              | 12                                   |
| state                   | up                                   |
| status                  | enabled                              |
| vcpus                   | 10                                   |
| vcpus_by_node           | None                                 |
```

```
| vcpus_used                  | 0                                        |
+----------------------------+------------------------------------------+
```

3.5.5 Specifying Bare Metal in Nova Flavor

Create a bare metal-specific flavor with the property type bare metal associated with the flavor. The Nova scheduler requires a Bare Metal resource to determine the start Nova instance.

Use the following command to create a bare metal-specific flavor:

```
$ openstack flavor create --id auto --ram $RAM_MB --vcpus $CPU --disk $DISK_GB
--property baremetal=true --public <name>
```

The `$RAM_MB`, `$CPU` and `$DISK_GB` are the desired resources for the bare metal instance.

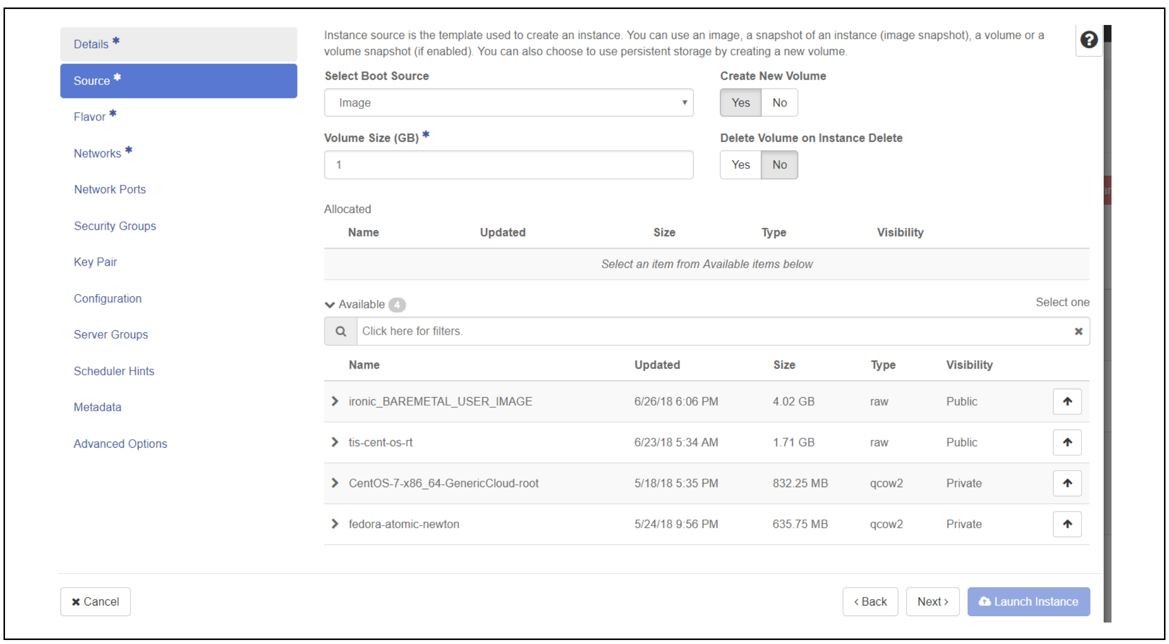
3.5.6 Launching Nova Bare Metal Instances

Use Nova to launch a bare metal instance with Horizon or the CLI.

1. To use Horizon to launch a bare metal instance for ironic node:

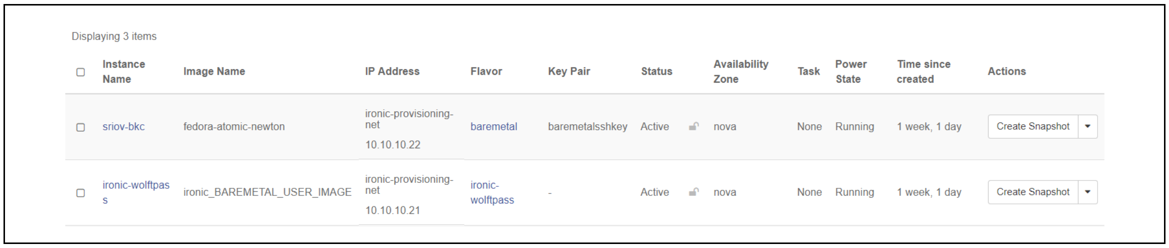
   a. Select the source to launch and then select Launch Instance.

Figure 47. Select Source



Horizon displays a list of launched instances.

Figure 48. Active Instances



2. To launch an instance using CLI:
   a. Ensure the `ironic-provisioning-net` tenant network has been created.
   b. Find the NIC id belonging to the instance.
   c. Enter the command:

```
#nova boot --image $BAREMETAL_USER_IMAGE --flavor $BAREMETAL_FLAVOR --
<nic-id> \ net-id=$IRONIC_PROVISIONING_NETWORK_ID <Instance_name>
```

These commands are an example of how to launch an instance using the CLI:

1. Use CLI to create a flavor.

```
#openstack flavor create --id auto --ram $RAM_MB --vcpus $CPU --disk
$DISK_GB --property baremetal=true --public <name>
```

2. Use CLI to create a instance for ironic node.

```
#nova boot --image $BAREMETAL_USER_IMAGE --flavor $BAREMETAL_FLAVOR --nic \
net-id=$IRONIC_PROVISIONING_NETWORK_ID <Instance_name>
```

3.5.7 Bare Metal Container Setup with Ironic

Details on how to set up containers with Ironic are in “Chapter 3 Installation Guide for Kubernetes*“ in the *FlexRAN_Reference Solution Cloud Native Setup Installation Guide.* (refer to Table 2.)

3.6 Troubleshooting Bare Metal Errors

This section explains how to resolve username and password errors in a bare metal environment.

3.6.1 Bare Metal User Image Username and Password Error

There are two solutions to resolve the issue:

Create a username and password for `CentOS-7-x86_64-GenericCloud-1708.qcow2`

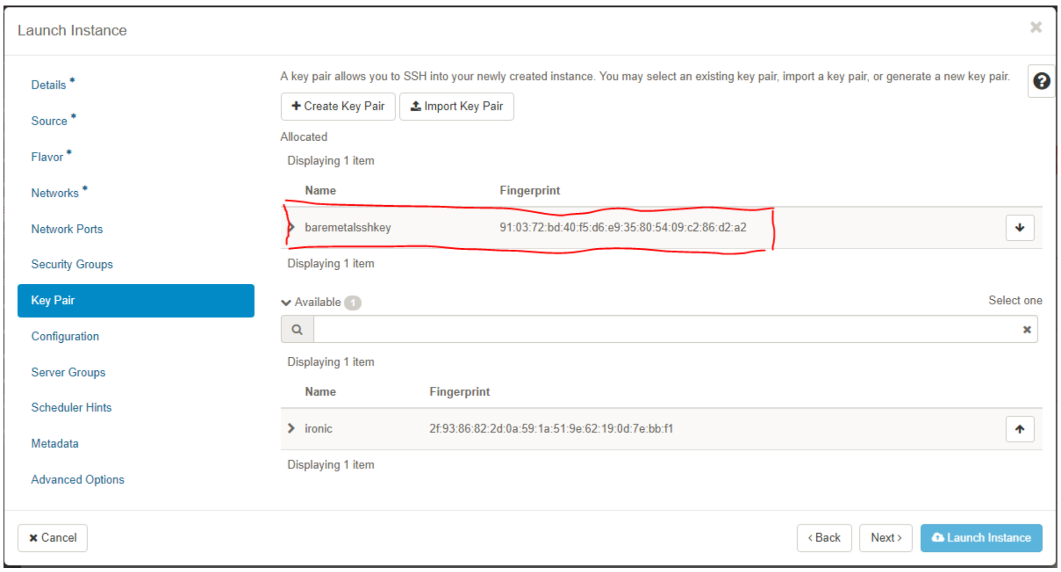
Download `centOS-7-x86_64-GenericCloud-1708.qcow2` to a CentOS* or Fedora* machine and run the commands:

```
#sudo yum install libguestfs-tools
#sudo virt-customize -a fedora-atomic-newton.qcow2 --root-password
password:root
```

Resolve the issue using key pairs on TIS5:

1. To use key pairs with Horizon:
   a. Select Key Pair and select the key pair `baremetalsshkey`

Figure 49. Select Key Pair



2. To use key pairs with CLI:
   a. Enter the commands:

```
$ test -f ~/.ssh/id_rsa.pub || ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
$ nova keypair-add --pub-key ~/.ssh/id_rsa.pub baremetalsshkey
```

§

4.0 BIOS Settings

This section contains BIOS settings for the supported servers:

- Skylake-SP Wolf Pass
- Broadwell-EP Wildcat Pass

4.1 Skylake-SP Wolf Pass BIOS Setting

The following sections provide BIOS configuration settings for Skylake-SP Wolf Pass.

4.1.1 BIOS Configuration Summary

Advanced →Processor Configuration:

- Intel Hyper-Threading Tech: **Disabled**
- Active Processor Cores: **ALL**
- Intel Virtualization Technology: **Enabled**
- LLC Prefetch: **Enabled**
- Intel VT for Directed I/O: **Enabled**
- Coherency Support: **Disabled**

Table 6 and Table 7 provide Power Management settings.

Table 6. Power Management Settings- Default

CPU Power and Performance	Uncore Power Management	CPU P State Control	Hardware P States	CPU C State Control
• CPU Power and Performance Policy <**Performance**> • Workload Configuration <**Balanced**>	• Uncore Frequency Scaling <**Enabled**> • Performance P-limit <**Enabled**>	• Enhanced Intel SpeedStep® Tech <**Disabled**>	• Hardware P-States <**Disabled**> • HardwarePM Interrupt <**Disabled**> • EPP Enable <**Disabled**> • APS rocketing <**Disabled**> • Scalability <**Disabled**> • PPO-Budget **<Disabled>**	• Package C-State <**C0/C1**> • C1E <**Disabled**> • Processor C6 <**Disabled**>

Table 7. Power Management Settings- TDP L2 (as required for FlexRAN)

CPU Power and Performance	Uncore Power Management	CPU P State Control	Hardware P States	CPU C State Control
• CPU Power and Performance Policy <**Performance**> • Workload Configuration <**Balanced**>	• Uncore Frequency Scaling <**Enabled**> • Performance P-limit <**Enabled**>	• Enhanced Intel SpeedStep® Tech <**Enabled**> • Intel Configurable TDP <**Enabled**> • Intel® Turbo Boost Technology <**Enabled**> • Energy Efficeint Turbo • <**Disabled**>	• Hardware P-States <**Disabled**> • HardwarePM Interrupt <**Disabled**> • EPP Enable <**Disabled**> • APS rocketing <**Disabled**> • Scalability <**Disabled**> • PPO-Budget **<Disabled>**	• Package C-State <**C0/C1**> • C1E <**Disabled**> • Processor C6 <**Disabled**>

4.1.1.1 Instructions to Update BIOS Configuration

Figure 50. BIOS Version

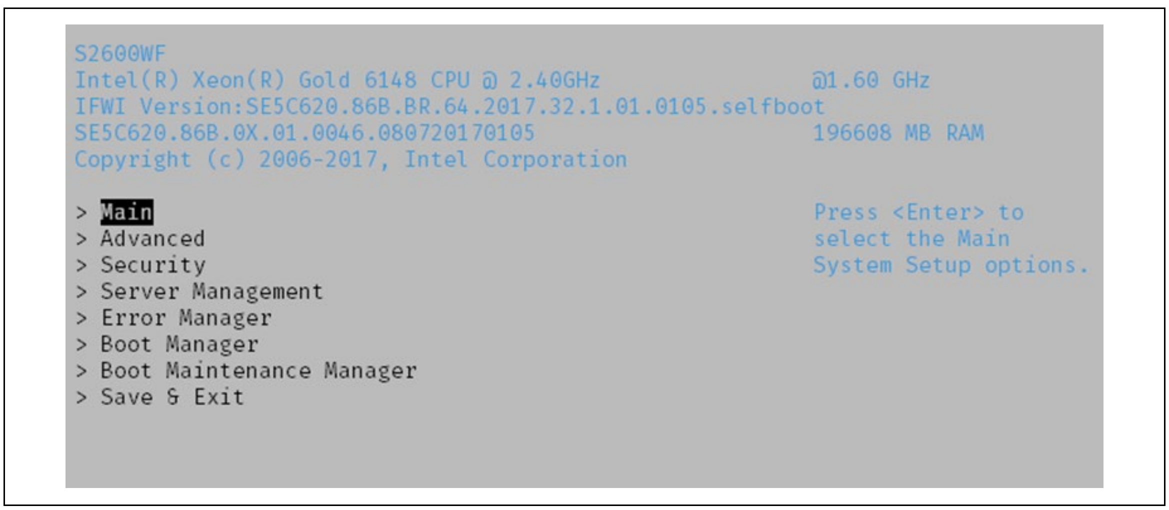


Figure 51. Disable Hyperthreading

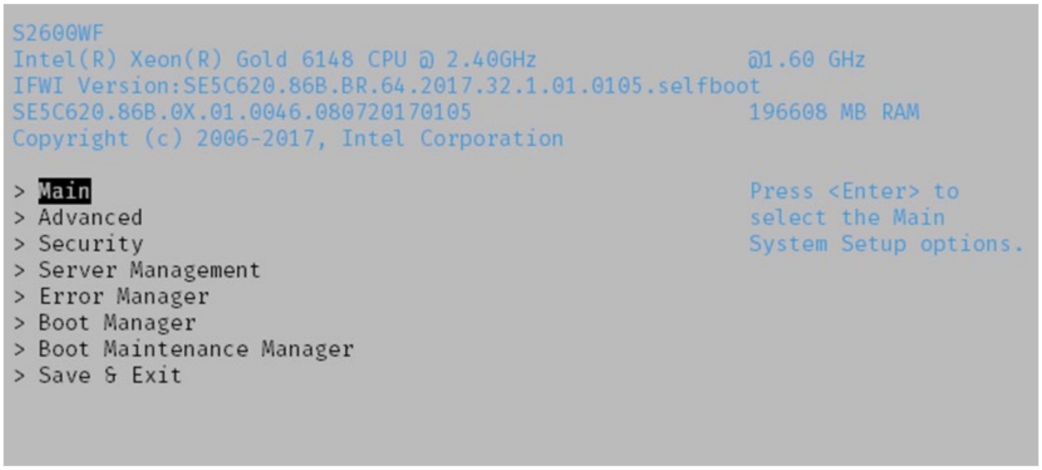
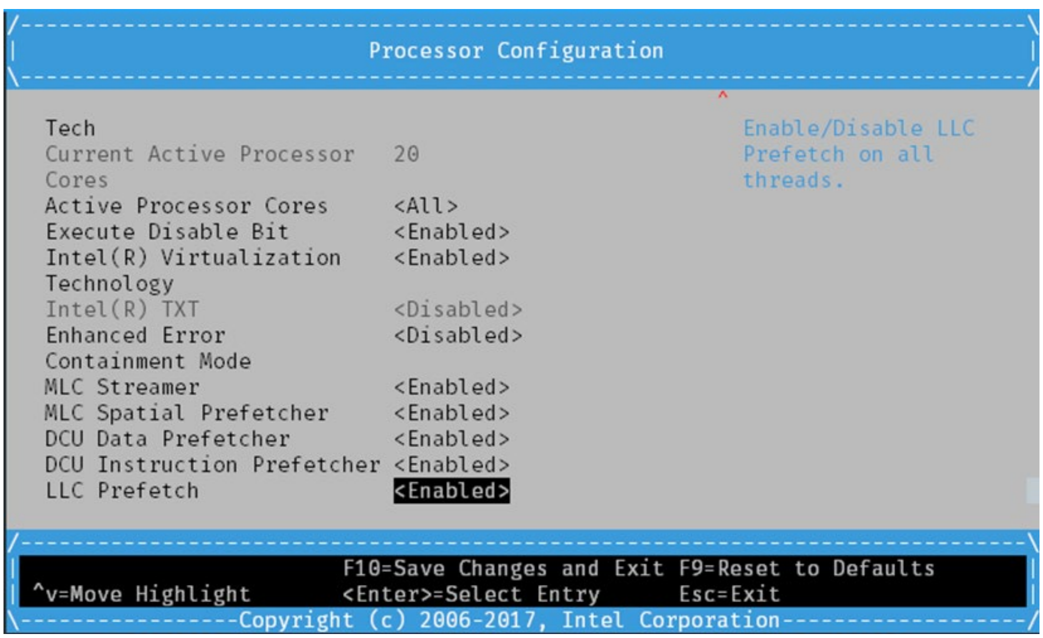
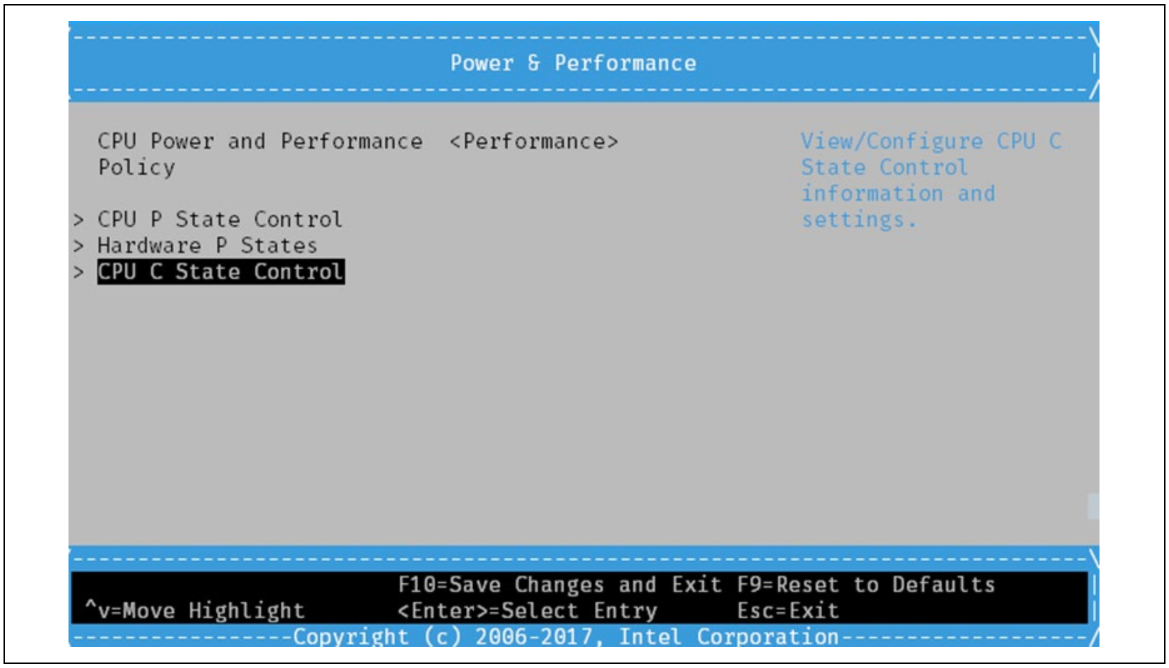


Figure 52. Enable Virtualization Technology (VT) and Prefetchers





Note: It is assumed the "Uncore Power Management" setting is not updated from the default settings by the user.

Figure 53. Set Processor Configuration

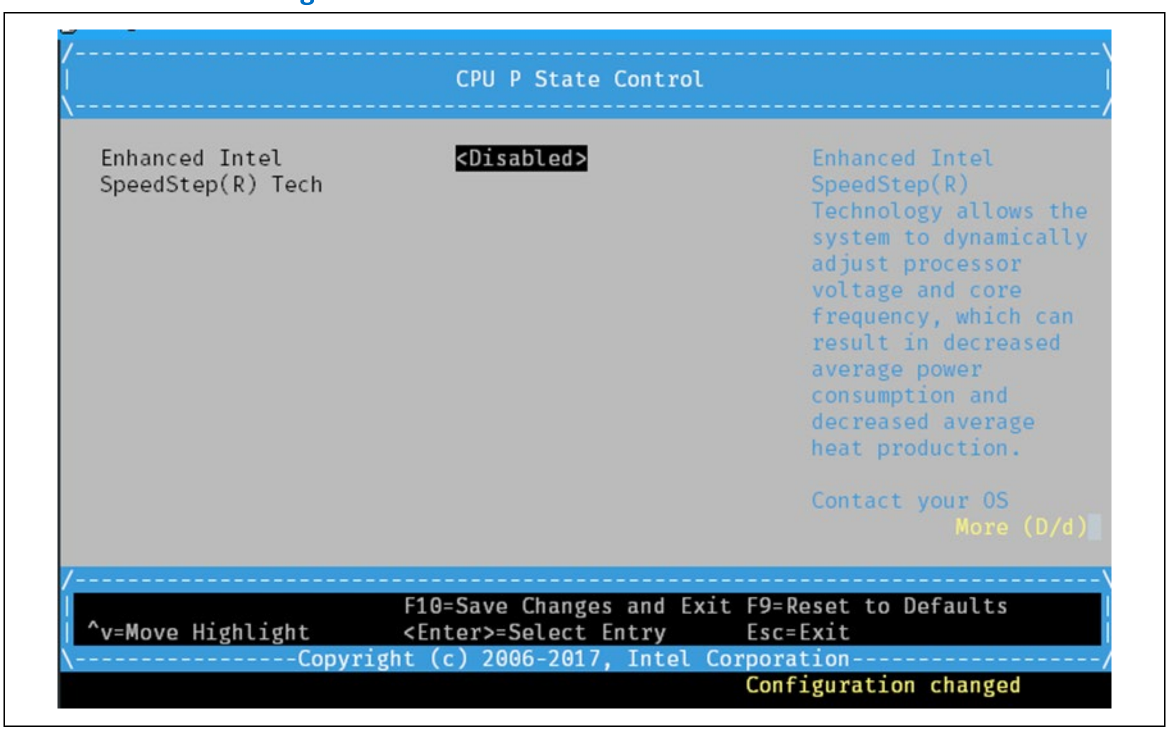


Figure 54. Set Hardware P States

Figure 55. Set CPU C State Control

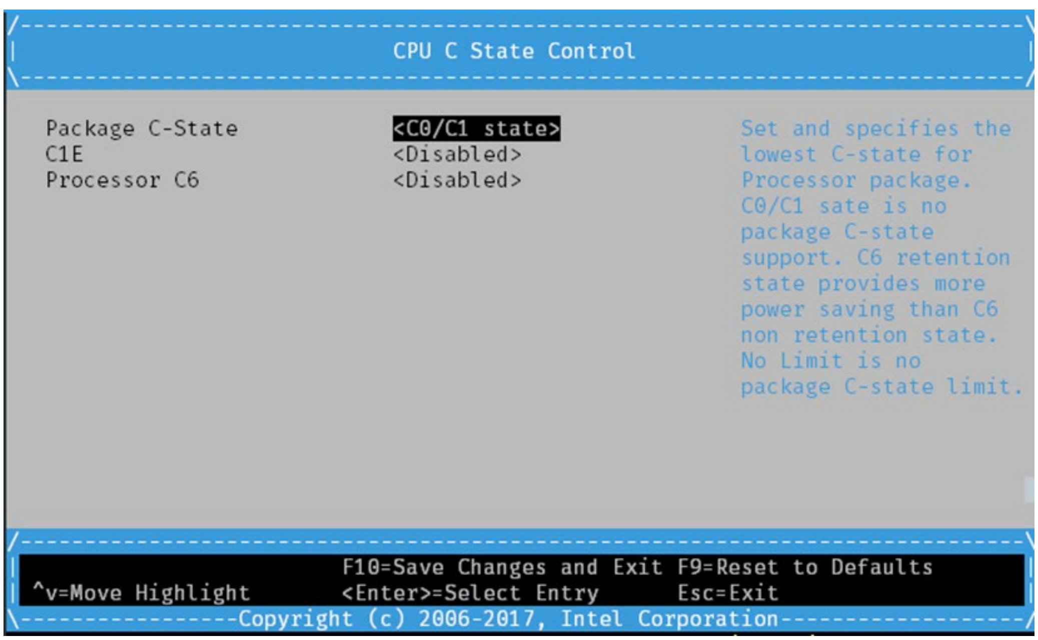
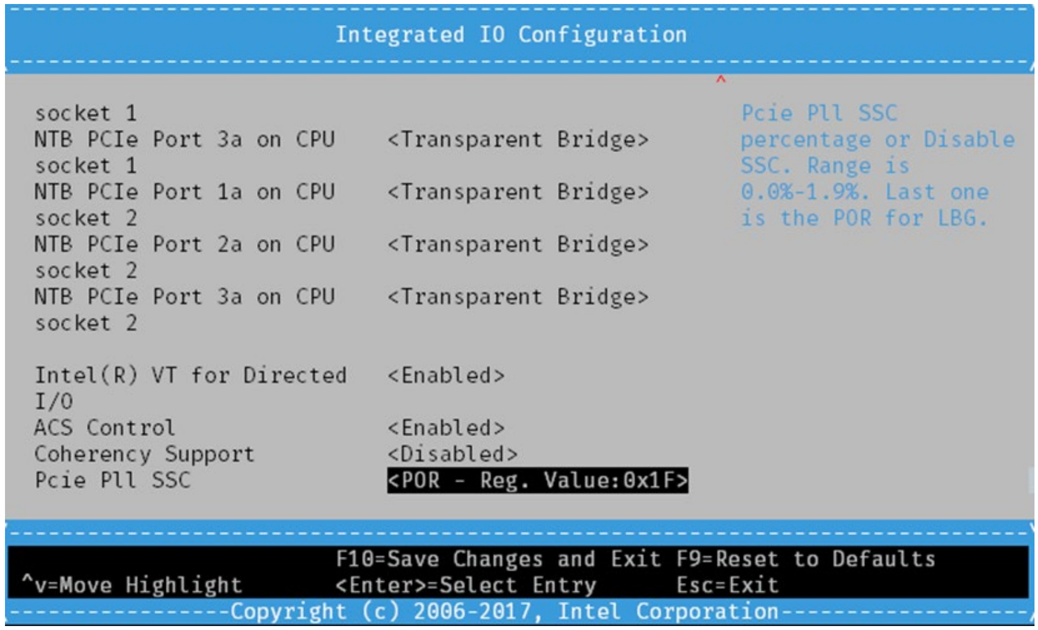


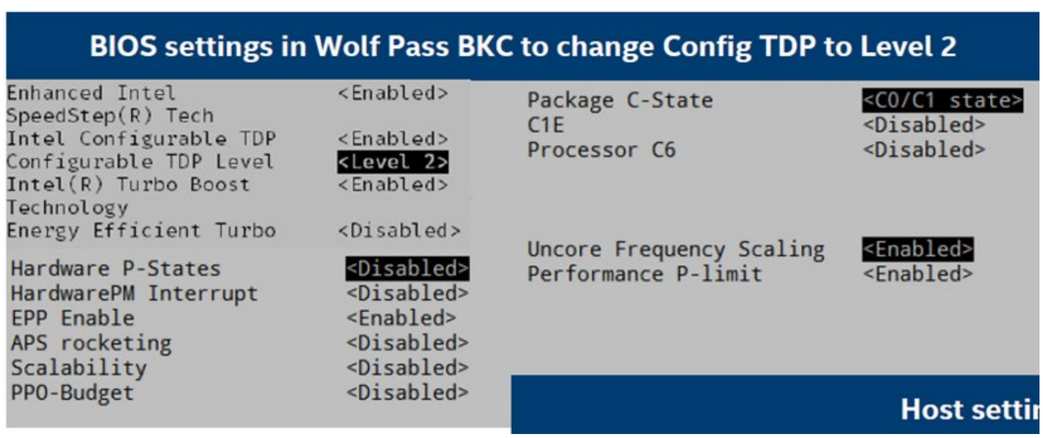
Figure 56. Set Integrated IO Configuration



4.1.2 Setting Config TDP L2 and Setting Uncore and Core Frequency to Maximum on Skylake-SP

Note: This configuration is only needed if you are testing with Config TDP L2. It is not needed for default configuration required to run the FlexRAN PHY.

Figure 57. Set BIOS Configuration



4.1.2.1 Set Frequency to Maximum

Note: Only applicable for Skylake-SP Wolf Pass with Intel® Xeon® Gold 6148 Processors.

1. Download `msr tools` from the link provided below for this operation to another server and copy the tools binary to the Wolf Pass compute note.

Download/Clone:

https://01.org/sites/default/files/downloads/msr-tools/msr-tools-1.3.zip

```
unzip msr-tools-1.3.zip
cd msr-tools-master
make
```

Note: Execute the following commands on the Wolf Pass compute node as `sudo` before running any test.

```
#!/bin/bash
for i in {0..39}
do
/home/wrsroot/msr-tools-master/wrmsr -p $i 0x199 0x19600
done
```

2. Set Uncore max frequency:

```
wrmsr -p 0 0x620 0x1e1e
wrmsr -p 30 0x620 0x1e1e
```

3. Set the frequency governor to performance:

```
cpupower frequency-set -g performance
```

4.2 Broadwell-EP Wildcat Pass BIOS Setting

This BIOS setting applies to Wildcat Pass only.

4.2.1 Reset BIOS

To reset the BIOS configuration, enter the BIOS and follow the instructions below:

1. Navigate to **Setup Menu->Save&Exit->Reset**

   In the bottom right hand corner of screen in yellow there is text `Configuration Changed`.

2. Press **ESC** to go back to the main menu.

4.2.2 Programming BIOS

Use the following script to program the BIOS:

1. Navigate into **Advanced->Power&Performance->CPU Power and Performance Policy** and select **Performance**.
2. Navigate into **Advanced->Power&Performance->CPU P State Control->Enhanced Intel SpeedStep® Tech** and select **Disabled**.
3. Press **ESC**.
4. Navigate into **Advanced->Power&Performance->CPU P State Control->CPU C State Control->CPU C-State** and select **Disabled**.
5. Press **ESC**.

6. Press **ESC**.
7. Navigate into **System Acoustic and Performance Configuration->Set Fan Profile** and select **Performance**.
8. Press **ESC**.
9. Navigate into **Processor Configuration->Intel® HyperThreading Tech** and press **Disabled**.
10. Navigate into **Processor Configuration->Intel® Virtualization** and press **Enabled**.
11. Press **ESC**.
12. Navigate into **Processor Configuration->Integrated IO Configuration->Intel® VT for Directed I/O** and press **Enabled**.
13. Press **ESC**.
14. Press **ESC**.
15. Navigate into **Save & Exit** and press **Save Changes and Exit**.

4.3 Settings for an Environment without VMs or Containers

To set up a host that is not used with VMs or containers, make sure the Virtualization Technology (VT) is disabled in the BIOS.

In the kernel command line parameters, disable and remove iommu parameter using the following:

```
intel_iommu("intel_iommu=off")
```

§

5.0 Preparing a Bootable USB Drive

Prepare a bootable USB drive from the boot image file included on the product DVD. The product DVD includes an image file (`.iso extension`), which is used to create bootable media (for example, a bootable DVD or USB drive).

For convenience, instructions are provided for creating a bootable USB drive. If preferred, prepare a bootable DVD or other bootable media, using any suitable open source or commercially available software.

5.1 Preparing a Bootable USB Drive on a Linux* System

On a Linux* system, use the built-in utilities to prepare a bootable USB drive using the following procedures:

1. Copy the ISO image from the DVD to a Linux workstation. Refer to Chapter 2.0, Wind River* Titanium Server Installation Guide, Titanium Server Software Installation Preparing a Bootable USB Drive.

    This example assumes that the copied image file is `/tmp/Titanium-Server-host-installer-16.10-b5b.iso`

2. Identify the Linux device of the USB drive on which to write the image by monitoring the system log and then attaching the USB drive.

    ```
    $ tail -f /var/log/syslog
    ```

    Attach the USB drive, and then use the displayed log to identify the assigned device.

3. Unmount all partitions on the USB drive.

    a. Identify the mount points for the partitions.

    In the following command, <usb_device> represents the actual device identifier (for example, `/dev/scd`).

    ```
    $ sudo mount | grep <usb_device>
    /dev/scd1 on /media/sdc type vfat (rw,relatime,...)
    ```

    In this example, only one partition, `/dev/sdc1`, is identified. Its mount point is `/media/sdc`.

    b. Unmount the partitions by referencing the mount points.

    In the following command, <mount_point> represents the actual mount point (for example, /media/sdc).

    ```
    $ sudo umount <mount_point>
    ```

    Repeat this command to unmount any additional partitions that may have been identified above.

4. Write the ISO image to the USB drive.

Caution: The following command overwrites any existing content on the USB flash drive.

```
$ sudo dd if=/tmp/TS-host-installer-1.0.iso of=/dev/sdc bs=1M; sync
1825+0 records in 1825+0 records out
1913651200 bytes (1.9 GB) copied, 499.681 s, 3.8 MB/s
```

Use other block sizes with the “bs” option, or omit it entirely and let the Linux kernel determine an optimal size.

The USB drive is ready now and can be safely removed from the workstation.

5.2 Preparing a Bootable USB Drive on a Windows* System

On a Microsoft* Windows* system, prepare a bootable USB flash drive using a variety of downloadable tools.

Note: The selection of tools varies depending on your Windows version.

1. Download and install a tool designed for your version of Windows.

   Use a tool that can burn media from an ISO 9660 image.

2. Follow the instructions provided with the tool.

§

6.0 1588 Set up [Early access]

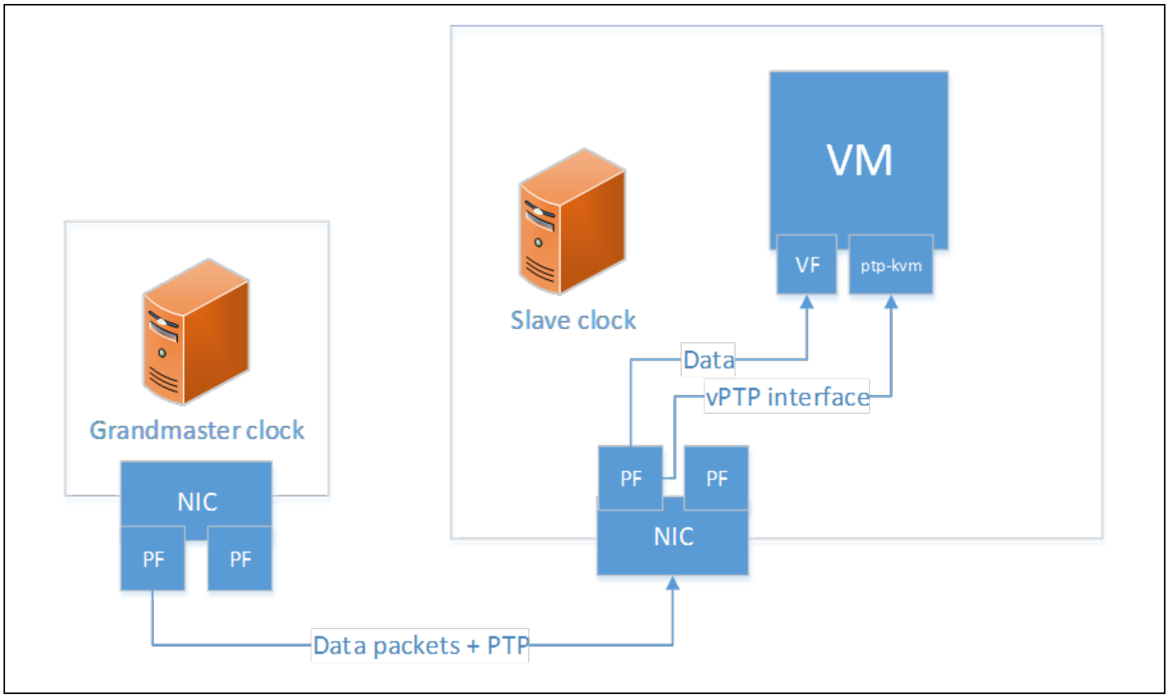
This chapter describes how to set up 1588 synchronization between two nodes.

Note: 1588 Set up was tested only as a standalone and currently it is not supported by FlexRAN.

6.1 Lab Set Up

Figure 58 describes the test bench used for 1588 synchronization.

Figure 58. 1588 Synchronization Test Bench Set Up



6.2 Setting up Precision Time Protocol

The Precision Time Protocol (PTP) provides synchronization services to the environment. Installing the Linux PTP package provides the `ptp4l` and `phc2sys` applications.

Note: PTP must be configured on the Grandmaster clock first, and then the non-master clock is set up and synchronized to it.

To verify the systems NIC uses hardware time stamps, run `ethtool`. Similar output should appear:

```
# ethtool -T eno4
Time stamping parameters for eno4:
Capabilities:
        hardware-transmit     (SOF_TIMESTAMPING_TX_HARDWARE)
        software-transmit     (SOF_TIMESTAMPING_TX_SOFTWARE)
        hardware-receive      (SOF_TIMESTAMPING_RX_HARDWARE)
```

```
        software-receive      (SOF_TIMESTAMPING_RX_SOFTWARE)
        software-system-clock (SOF_TIMESTAMPING_SOFTWARE)
        hardware-raw-clock    (SOF_TIMESTAMPING_RAW_HARDWARE)
PTP Hardware Clock: 3
Hardware Transmit Timestamp Modes:
        off                   (HWTSTAMP_TX_OFF)
        on                    (HWTSTAMP_TX_ON)
Hardware Receive Filter Modes:
        none                  (HWTSTAMP_FILTER_NONE)
        ptpv1-l4-sync         (HWTSTAMP_FILTER_PTP_V1_L4_SYNC)
        ptpv1-l4-delay-req    (HWTSTAMP_FILTER_PTP_V1_L4_DELAY_REQ)
        ptpv2-event           (HWTSTAMP_FILTER_PTP_V2_EVENT)
```

Note: After synchronization of the host to Grandmaster clock, the time in containers is the same as on the host machine.

PTP requires the following Kernel configuration options to be enabled:

- `CONFIG_PPS`
- `CONFIG_NETWORK_PHY_TIMESTAMPING`
- `CONFIG_PTP_1588_CLOCK`
- `CONFIG_PTP_1588_CLOCK_KVM` (for vPTP support)

6.2.1 Grandmaster Clock

On the system with the Grandmaster clock side, look at the `/etc/sysconfig/ptp4l` file (the last character is a lowercase L). It is the daemon configuration file that provides starting options to the PTP. Its content should look like this:

```
OPTIONS="-f /etc/ptp4l.conf -i <if_name>"
```

Where `<if_name>` is the interface name that will be used for time stamping and `/etc/ptp4l.conf` is the `PTP4L` configuration file.

PTP uses a BMC algorithm to choose a Grandmaster clock, and it is not obvious which timer is chosen by default. To specify a given timer as a Grandmaster clock, edit the `/etc/ptp4l.conf` file, setting the `priority1` property to `127`.

Use the following script to start the `ptp4l` service:

```
service ptp4l start
```

Output from the service can be checked at `/var/log/messages`. Output for the master clock should look like:

```
Mar 16 17:08:57 localhost ptp4l: ptp4l[23627.304]: selected /dev/ptp2 as PTP
clock
Mar 16 17:08:57 localhost ptp4l: [23627.304] selected /dev/ptp2 as PTP clock
Mar 16 17:08:57 localhost ptp4l: [23627.306] port 1: INITIALIZING to LISTENING
on INITIALIZE
Mar 16 17:08:57 localhost ptp4l: ptp4l[23627.306]: port 1: INITIALIZING to
LISTENING on INITIALIZE
Mar 16 17:08:57 localhost ptp4l: [23627.307] port 0: INITIALIZING to LISTENING
on INITIALIZE
Mar 16 17:08:57 localhost ptp4l: ptp4l[23627.307]: port 0: INITIALIZING to
LISTENING on INITIALIZE
Mar 16 17:08:57 localhost ptp4l: [23627.308] port 1: link up
```

```
Mar 16 17:08:57 localhost ptp4l: ptp4l[23627.308]: port 1: link up
Mar 16 17:09:03 localhost ptp4l: [23633.664] port 1: LISTENING to MASTER on
ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES
Mar 16 17:09:03 localhost ptp4l: ptp4l[23633.664]: port 1: LISTENING to MASTER
on ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES
Mar 16 17:09:03 localhost ptp4l: ptp4l[23633.664]: selected best master clock
001e67.fffe.d2f206
Mar 16 17:09:03 localhost ptp4l: ptp4l[23633.665]: assuming the grand master
role
Mar 16 17:09:03 localhost ptp4l: [23633.664] selected best master clock
001e67.fffe.d2f206
Mar 16 17:09:03 localhost ptp4l: [23633.665] assuming the grand master role
```

The next step is to synchronize the PHC timer to system time. Use the `phc2sys` daemon.

1. Edit configuration file at `/etc/sysconfig/phc2sys`. Replace `<if_name>` statement with the interface name using the following script.

```
OPTIONS="-c <if_name> -s CLOCK_REALTIME -w"
```

2. Start the `phc2sys` service.

```
service phc2sys start
```

Logs can be viewed at `/var/log/messages` and look like:

```
phc2sys[3656456.969]: Waiting for ptp4l...
phc2sys[3656457.970]: sys offset -6875996252 s0 freq  -22725 delay   1555
phc2sys[3656458.970]: sys offset -6875996391 s1 freq  -22864 delay   1542
phc2sys[3656459.970]: sys offset       -52 s2 freq  -22916 delay   1536
phc2sys[3656460.970]: sys offset       -29 s2 freq  -22909 delay   1548
phc2sys[3656461.971]: sys offset       -25 s2 freq  -22913 delay   1549
```

Grandmaster clock is configured.

6.2.2 Non-master Clock

Non-master clock configuration is the same as for the Grandmaster clock except in the `/etc/ptp4l.conf` the `priority1` property value for `ptp4l` is default value 128.

1. Run the `ptp4l` service.

2. To keep system time synchronized to PHC time, change the `phc2sys` options in `/etc/sysconfig/phc2sys` to:

```
OPTIONS=''phc2sys -s <if_name> -w"
```

3. Replace `<if_name>` with the interface name.

Logs will be available at `/var/log/messages`.

```
phc2sys[28917.406]: Waiting for ptp4l...
phc2sys[28918.406]: phc offset -42928591735 s0 freq +24545 delay 1046
phc2sys[28919.407]: phc offset -42928611122 s1 freq +5162 delay 955
phc2sys[28920.407]: phc offset 308 s2 freq +5470 delay 947
phc2sys[28921.407]: phc offset 408 s2 freq +5662 delay 947
phc2sys[28922.407]: phc offset 394 s2 freq +5771 delay 947
```

After this, both clocks should be synchronized. Docker is using the same clock as the host so its clock will be synchronized as well.

6.3 Setting up vPTP

Virtual Precision Time Protocol (vPTP) can be used to provide precision synchronization between the VM and Host machine. It is done using the `ptp-kvm` module using hypercalls to the host system.

6.3.1 VM Side

A few steps are required to do on the VM to provide synchronization to the host clock.

Configure the VM using the following steps to synchronize the VM with the host clock.

1. Load `ptp-kvm` module using `modprobe ptp-kvm` command. `/dev/ptp0 device` should appear.
2. Add following code to the chrony configuration file (`/etc/chrony.conf`):

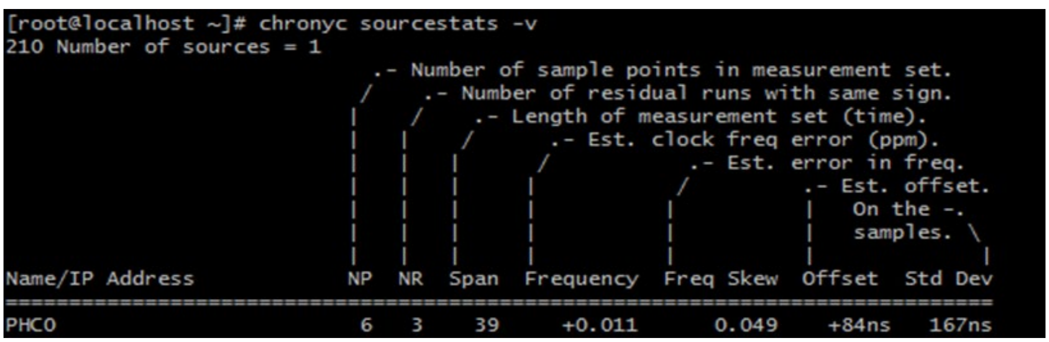
   ```
   refclock PHC /dev/ptp0 poll 3 dpoll -2 offset 0
   ```

Note: The number typed after poll is describing synchronization interval. It is a power of 2 in seconds.

3. Run `chronyd` daemon by command service `chronyd` start.

   Use the `chronyc` tool to collect the results. An example use is shown in the figure below.

Figure 59. Chronyc Source Statistics



§

7.0 Recommended Shutdown and Power-up Sequence for an Installed System

Before moving the system to another location or shut down, the following steps are recommended.

1. Using another computer that is part of the OAM LAN, enter the OAM floating IP address to log into the Controller from the Horzon GUI.
2. Turn off all VM instances, wait for all instances to shut down before proceeding to the next step.
3. Lock the Compute Nodes.

Note: Make sure that Controller is ***not*** locked.

4. From the compute node console – issue a “shutdown now” or equivalent command.
5. Power off the compute node.
6. Set ssh to standby cotroller if any and shutdown from the console using the “shutdown now” command and once completed power off.
7. From the console of the active controller (OAM IP address) issue the "shutdown now" command and then power off the server.
8. Before transporting the computer, make sure all cards are tightly screwed in and the PCIe riser, server lid, all disks etc. are firmly in their pace as expected.

To power up the system again, follow the recommended steps below:

1. Ensure the power and network connections work as expected.

   Refer to Section 2.2 or the last working applicable network connection that was working prior to system disconnect and/or shutdown.

Warning: Do not change PCIe slots, hardware settings, or connections as it may require reinstallation of OS.

2. Power on the active Controller and any standby controller if configured.
3. Power on compute node.
4. From the Horizon GUI, unlock the compute node and wait for the bootup sequence to complete.
5. Find the MGMT IP address of the compute node and ssh to its console.

Note: Repeat the steps to configure the compute node for Setting Config TDP L2, Setting Uncore, Core Frequency maximum peformance as required on Skylake-SP.

6. Start all VM instances in the required sequence (e.g., FlexRAN started as the last VNF).

§