



Cisco Virtualized Infrastructure Manager Installation Guide, 3.2.2

First Published: 2019-07-04

Last Modified: 2020-01-29

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2019 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Overview to Cisco Network Function Virtualization Infrastructure	1
Cisco Network Function Virtualization Infrastructure Overview	2
Cisco Virtualized Infrastructure Manager Overview	9
Cisco VIM Features	10
Cisco VIM Networking Overview	17
UCS C-Series Network Topologies	25
Cisco VIM Management Node Networking	33
IPv6 Support on Management Network	36
UCS C-Series and B-Series -Topologies	36
Cisco NFVI High Availability	38
Cisco NFVI Storage Node Overview	40
Overview to Cisco Virtual Topology System	41
Overview to Cisco NFVIMON	43
Overview to Cisco NFVIMON High Availability	45
Overview to CVIM-MON	45
Comparative Analysis	46
Metrics Collection	48
Alerting Rules	51
CVIM-MON Web User Interface	56
SNMP for Monitoring	59
Telemetry Service through OpenStack	60
Overview to Cisco VIM Unified Management	61
Overview to NFVbench	62
Overview to ACI Plugin Integration	64
Auto-ToR Configuration via ACI API	65
NCS-5500 as a ToR Option	66

Disk Management in VIM	66
OSD Maintenance	67
Power Management of Computes for C-Series	67
Physical Cores and Memory Reserved for Cisco VIM Infrastructure	67
Cisco VIM Software Hub	68
Cisco VIM VXLAN EVPN Design	69
VPP Port Mirroring Support	72
Container Workload Support	73

CHAPTER 2**Overview to Cisco NFVI Installation** 75

Cisco NFVI Installation Overview	75
----------------------------------	----

CHAPTER 3**Preparing for Installation on Servers Without Internet Access** 77

Preparing to Install Cisco NFVI on Management Nodes Without Internet Access	77
---	----

CHAPTER 4**Preparing for Cisco NFVI Installation** 81

Installing Cisco NFVI Hardware	81
Configuring ToR Switches for C-Series Pods	85
Configuring ToR Switches for UCS B-Series Pods	89
Preparing Cisco IMC and Cisco UCS Manager	92
Installing Management Node on UCS C-series (M4/M5)	92
Installing Management Node on Quanta Servers	95
Installing Cisco VIM Software Hub	96
Installing Cisco VIM Software Hub Node	98
Setting up Cisco VIM Software Hub for Cisco VIM Artifact Distribution	98
Installing Cisco VIM Software Hub in Connected Mode	99
Installing Cisco VIM Software Hub in Air-Gapped Mode	99
Installing Pod from Cisco VIM Software Hub Server	102
Day 2 Operations on Cisco VIM Software Hub	103
Setting Up the UCS C-Series Pod	103
Utility Details	106
Setting Up the UCS B-Series Pod	108
Configuring the Out-of-Band Management Switch	110
Support of 3rd Party Compute (HP DL 360 Gen9)	110

CHAPTER 5**Installing Cisco VTS 111**

Overview to Cisco VTS Installation in Cisco NFVI 111
Cisco VTS Usernames and Passwords in Cisco NFVI 113
Modes of TOR Configuration with VTS 114
System Requirements for VTC VM 116
System Requirements for VTSR VM 117
Supported Virtual Machine Managers 117
Supported Platforms 117
Installing Cisco VTS in Cisco NFVI Environment 119
Installing VTC VM - Automatic Configuration Using ISO File 119
Installing VTC VM - Manual Configuration Using Virt-Manager 120
Installing VTC VM - Manual Configuration using VNC 122
Installing the VTSR VMs 123
Creating VTSR VM 123
Bringing up the KVM-based VTSR VM 123
Creating an ISO for IOS VTSR 124
Verifying Cisco VTS Installation in Cisco NFVI 126
Verifying VTSR VM Installation 126
Verifying VTC VM Installation 127
Troubleshooting VTF Registration 127
Configuring Cisco VTS and VTSR After Installation 128
Installing VTS in an HA Configuration 129
Completing VTSR HA Configuration 133
Uninstalling VTC HA 133
Sample Cisco VTS Configurations for Cisco NFVI 133

CHAPTER 6**Installing Cisco VIM 139**

Cisco VIM Installation Overview 139
Installing Cisco VIM 140
Cisco VIM Client Details 142
Re-installing Pod with same Image version 145
Cisco VIM Configuration Overview 146
Configuring ToR Automatically 146

Setting Up Cisco VIM Data Configuration	146
Setting up ToR Configurations for B-series and C-series	147
Support for Custom Configuration	149
Setting Up Cisco VIM OpenStack Configuration	166
Control and Data Plane Testing in Cisco VIM	169
Optional Services in Cisco VIM	169
OpenStack Object Storage Integration with Cisco VIM	176
SolidFire Integration with Cisco VIM	180
Cisco VIM Configurations for VPP/VLAN Installation	181
Cisco VIM Configuration for Cisco VTS Installation	181
Enabling ACI in Cisco VIM	183
Additional Settings for Auto-ToR via ACI API on Day 0	188
Setting of Memory Oversubscription Usage	190
Setting of CPU Oversubscription Usage	190
Disabling Management Node Accessibility to Cloud API Network	191
Enabling NFVbench on Cisco VIM	191
Enabling NFVIMON on Cisco VIM	195
Enabling CVIM-MON on Cisco VIM	198
Enabling or Disabling Autobackup of Management Node	200
Enabling Custom Policy for VNF Manager	201
Forwarding ELK logs to External Syslog Server	201
Support of NFS for ELK Snapshot	201
Support for TTY Logging	202
Configuring Additional VIM Administrators	202
Support of LDAP for Management Node	203
Horizon Hosting Through NAT or DNS Aliases	203
DHCP Reservations for VM's MAC Addresses	203
Customizing SSH Login Banner	204
Cinder Volume Encryption	204
Configuring Support for Read-only OpenStack Role	204
VPP Port Mirroring Support	205
Setting up VXLAN/EVPN in Cisco VIM	207
Setting up Trusted Virtual Functions	209
Setting up Reception/Transmission Buffer Size	209

Updating Cisco NFVI Software **210**

CHAPTER 7

Installing Cisco VIM Unified Management **211**

Installing Cisco VIM Unified Management with Internet Access **211**

Installing Cisco VIM Unified Management with Cisco VIM Software Hub **216**

Installing Cisco VIM Unified Management with LDAP **217**

Installing Cisco VIM Unified Management Without SMTP **217**

Installing Cisco VIM Unified Management without Internet Access **219**

Installing Cisco VIM Unified Management with Optional Services **222**

Cisco VIM Insight Post Bootstrap Validation Checks **223**

VIM UM Admin Login for Standalone Setup **226**

VIM UM Pod Admin Login for Standalone Setup **227**

CHAPTER 8

Installing Cisco VIM through Cisco VIM Unified Management **229**

Unified Management Dashboard **229**

Pods **230**

Pod Users **231**

Revoking User **231**

Deleting Users **231**

Pod Administrator **232**

Adding Pod Admin **232**

Revoking Pod Admin **232**

Unified Management (UM) Administrator **233**

Adding UM Admin **233**

Revoking UM Admin **233**

Registering New Pod to Insight **234**

Configuring OpenStack Installation **235**

Post Installation Features for Active Blueprint **337**

Monitoring the Pod **337**

Cross Launching Horizon **338**

NFVI Monitoring **338**

Run VMTP **339**

Run CloudPulse **339**

Run Nfvbench **339**

Fixed Rate Test	340
POD Management	341
System Update	341
Reconfiguring CIMC Password through Insight	341
Reconfiguring OpenStack Password	342
Reconfiguring OpenStack Services, TLS certs and ELK configurations	343
Reconfiguring Optional Services	343
Pod User Administration	345
Managing Users	345
Managing Roles	345
Managing Root CA Certificate	346

CHAPTER 9

Verifying the Cisco NFVI Installation 347

Displaying Cisco NFVI Node IP Addresses	347
Verifying Cisco VIM Client CLI Availability	348
Displaying Cisco NFVI Logs	349
Accessing OpenStack API Endpoints	349
Assessing Cisco NFVI Health with CloudPulse	350
Displaying HA Proxy Dashboard and ELK Stack Logs	352
Checking Cisco NFVI Pod and Cloud Infrastructure	352

APPENDIX A

Appendix 353

Cisco VIM Wiring Diagrams	353
---------------------------	-----



CHAPTER 1

Overview to Cisco Network Function Virtualization Infrastructure

This section contains the following topics:

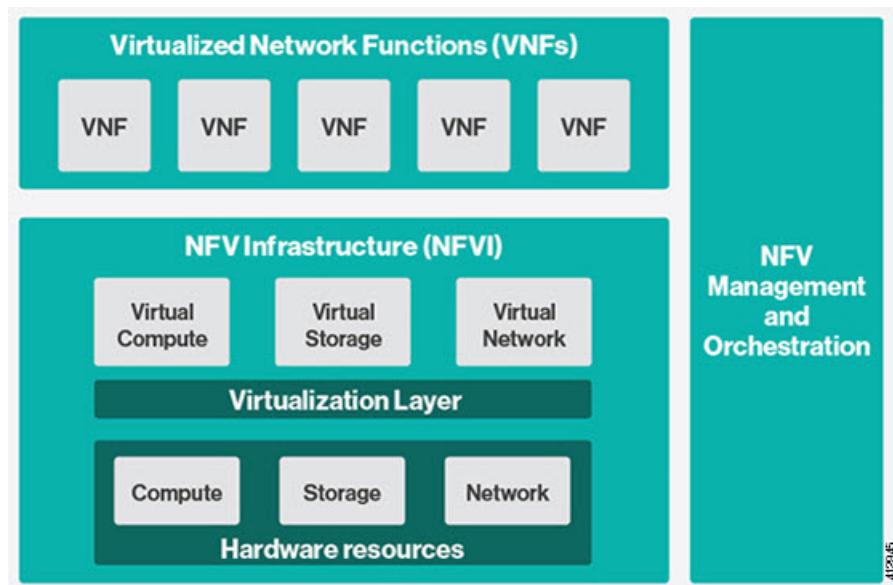
- [Cisco Network Function Virtualization Infrastructure Overview, on page 2](#)
- [Cisco Virtualized Infrastructure Manager Overview, on page 9](#)
- [Cisco VIM Networking Overview, on page 17](#)
- [UCS C-Series Network Topologies, on page 25](#)
- [Cisco VIM Management Node Networking, on page 33](#)
- [IPv6 Support on Management Network, on page 36](#)
- [UCS C-Series and B-Series -Topologies, on page 36](#)
- [Cisco NFVI High Availability, on page 38](#)
- [Cisco NFVI Storage Node Overview, on page 40](#)
- [Overview to Cisco Virtual Topology System, on page 41](#)
- [Overview to Cisco NFVIMON, on page 43](#)
- [Overview to Cisco NFVIMON High Availability, on page 45](#)
- [Overview to CVIM-MON, on page 45](#)
- [Telemetry Service through OpenStack, on page 60](#)
- [Overview to Cisco VIM Unified Management, on page 61](#)
- [Overview to NFVbench, on page 62](#)
- [Overview to ACI Plugin Integration, on page 64](#)
- [Auto-ToR Configuration via ACI API, on page 65](#)
- [NCS-5500 as a ToR Option, on page 66](#)
- [Disk Management in VIM, on page 66](#)
- [OSD Maintenance, on page 67](#)
- [Power Management of Computes for C-Series, on page 67](#)
- [Physical Cores and Memory Reserved for Cisco VIM Infrastructure, on page 67](#)
- [Cisco VIM Software Hub, on page 68](#)
- [Cisco VIM VXLAN EVPN Design, on page 69](#)
- [VPP Port Mirroring Support, on page 72](#)
- [Container Workload Support, on page 73](#)

Cisco Network Function Virtualization Infrastructure Overview

Cisco Network Function Virtualization Infrastructure (NFVI) provides the virtual layer and hardware environment in which virtual network functions (VNFs) can operate. VNFs provide well-defined network functions such as routing, intrusion detection, domain name service (DNS), caching, network address translation (NAT), and other network functions. While these network functions require a tight integration between network software and hardware, the use of VNF enables to decouple the software from the underlying hardware.

The following figure shows the high level architecture of Cisco NFVI.

Figure 1: General NFV Infrastructure



Cisco NFVI includes a virtual infrastructure layer (Cisco VIM) that embeds the Red Hat OpenStack Platform (OSP 13). Cisco VIM includes the Queens release of OpenStack, which is an open source cloud operating system that controls large pools of compute, storage, and networking resources. Cisco VIM manages the OpenStack compute, network, and storage services, and all NFVI management and control functions. Key Cisco NFVI roles include:

- Control (including Networking)
- Compute
- Storage
- Management (including logging, and monitoring)

Hardware that is used to create the Cisco NFVI pods include a specific combination of the following based on pre-defined BOMs. For more details, contact Cisco VIM Product Management.

- Cisco UCS® C240 M4: Performs management and storage functions and services. Includes dedicated Ceph (UCS 240-M4) distributed object store and file system. (Only Red Hat Ceph is supported).
- Cisco UCS C220/240 M4: Performs control and compute services.
- HP DL360 Gen9: It is a third-party compute where the control plane is Cisco UCS servers.

- Cisco UCS 220/240 M5 (SFF): In a Micropod environment, expandable to maximum of 16 computes.
- Cisco UCS B200 M4 blades: It can be used instead of the UCS C220 for compute and control services. The B200 blades and C240 Ceph server are connected with redundant Cisco Fabric Interconnects managed by UCS Manager.
- Combination of M5 series servers are supported in M5-based Micropod and VIC/NIC (pure 40G) based Hyper-Converged and Micropod offering.
- Quanta servers as an alternate to Cisco UCS servers: Use of specific Quanta servers for the installation of the cloud both at the core and edge. An automated install of Central Ceph cluster to the edge pods is supported for Glance image services.

The UCS C240 and C220 servers are of type M4 or M5 Small Form Factor (SFF) models where the nodes can boot off a pair of HDDs or SSD as specified in BOM. Each UCS C240, UCS C220, and UCS B200 have two 10 GE Cisco UCS Virtual Interface Cards.

The B-Series pod consists of Cisco UCS B200 M4 blades for the Cisco NFVI compute and controller nodes with dedicated Ceph on a UCS C240 M4. The blades and Ceph server are connected via redundant fabric interconnects (FIs) managed by Cisco UCS Manager. The Cisco VIM installer performs bare metal installation and deploys OpenStack services using Docker™ containers to allow for OpenStack services and pod management software updates.

The following table shows the functions, hardware, and services managed by Cisco NFVI nodes.

Table 1: Cisco NFVI Node Functions

Function	Number	Hardware	Services
Management	1	<ul style="list-style-type: none"> • UCS C240 M4 SFF with 8, 16, or 24 1.2 TB HDDs (24 is recommended) • UCS C240 M5 SFF with 8, 16, or 24 1.2 TB HDDs (24 is recommended) • UCS C220 M5 SFF with 8x1.2 TB HDDs • Quanta Server (D52BE-2U) with 2x1.2TB HDD • Quanta Server (D52BQ-2U 3UPI) with 2x3.8TB HDD 	<ul style="list-style-type: none"> • Cisco VIM Installer • Cobbler server • Docker Registry • ELK server • CVIM MON components: Prometheus and TSDB

Function	Number	Hardware	Services
Control	3	<ul style="list-style-type: none"> • UCS C220/C240 M4/M5 with 2x 1.2 TB HDDs or 2x960G SSDs (in a Micropod or Full Pod environment) • UCS B200 with two 1.2 TB HDDs • Quanta Server (D52BE-2U) with 2x960 G SSD • Quanta Server (D52BQ-2U 3UPI) with 2x960 G SSD for edge pod 	<ul style="list-style-type: none"> • Maria Database/Galera • RabbitMQ • HA Proxy/Keepalive • Identity Service • Image Service • Compute management • Network service • Storage service • Horizon dashboard • Fluentd
Compute	2+	<ul style="list-style-type: none"> • UCS C220/C240 M4/M5 with two 1.2 TB HDDs, or 2x9.6 GB SSDs (in a Micropod or Full Pod environment) • UCS B200 with two 1.2 TB HDDs • HP DL360 Gen9 • Quanta Server (D52BE-2U/ D52BQ-2U 3UPI) with 2x960 G SSD 	<ul style="list-style-type: none"> • Virtual Networking Service • Compute service • Fluentd

Function	Number	Hardware	Services
Storage	3 or more	<p>SSD and HDD drives must be in a 1:4 ratio per storage node minimum.</p> <p>Storage node configuration options:</p> <p>Fullon environment::</p> <ul style="list-style-type: none"> • UCS C240 M4/M5 with two internal SSDs, 1-4 external SSD, 4-20x- 1.2 TB HDDs • SSD-based Ceph: UCS C240 M4/M5 with 2 internal SSDs, minimum of 4 external SSDs, expandable to 24 SSDs • Quanta Server (D52BE-2U) HDD Based: 4 SSD 960GB for Journal + 16 SAS HDD (16x2.4 TB) for OSD + 2 (2x2.4 TB SAS 10krpm HDD) for OS • Quanta Server (D52BE-2U) SSD Based: 20 SSD (3.8 TB) OSD + 2 OSBoot (2x3.8TB SSD) <p>Micropod/UMHC/NGENAHC environment:</p> <ul style="list-style-type: none"> • UCS C240 M4/M5 with two 1.2TB HDD for OS boot, one/2 SSDs and 5/10x1.2 TB HDDs • UCS C240 M4/M5 with 2x960GB SSD for OS boot and 4 or 8 x960 GB SSDs 	<ul style="list-style-type: none"> • Storage service
Top of Rack (ToR)	2	<p>Recommended Cisco Nexus 9000 series switch software versions:</p> <ul style="list-style-type: none"> • 7.0(3)I4(6) • 7.0(3)I6(1) <p>Cisco NCS 5500 as ToRs or Cisco Nexus 9000 switches running ACI 3.0 (when ACI is used)</p>	<p>ToR services</p> <ul style="list-style-type: none"> • Cisco NCS 5500 provides ToR service with VIM running on C-series with Intel NIC and VPP as the mechanism driver for deployment.

**Note**

- Internal SSD is the boot device for the storage node
- You can use any ToR that supports virtual port channel. Cisco recommends you to use Cisco Nexus 9000 SKUs as ToR, which is released as part of Cisco VIM. When Cisco NCS 5500 acts as a ToR, auto-ToR config is mandatory.
- You must use the automated ToR configuration feature for Cisco NCS 5500.

Software applications that manage Cisco NFVI hosts and services include:

- Red Hat Enterprise Linux 7.6 with OpenStack Platform 13.0—Provides the core operating system with OpenStack capability. RHEL 7.6 and OPS 13.0 are installed on all target Cisco NFVI nodes.
- Cisco Virtual Infrastructure Manager (VIM)—An OpenStack orchestration system that helps to deploy and manage an OpenStack cloud offering from bare metal installation to OpenStack services, taking into account hardware and software redundancy, security and monitoring. Cisco VIM includes the OpenStack Queens release with more features and usability enhancements that are tested for functionality, scale, and performance.
- Cisco Unified Management—Deploys, provisions, and manages Cisco VIM on Cisco UCS servers.
- Cisco UCS Manager—Used to perform certain management functions when UCS B200 blades are installed. Supported UCS Manager firmware versions are 2.2(5a) and above.
- Cisco Integrated Management Controller (IMC)—Cisco IMC 2.0(13i) or later is supported, when installing Cisco VIM 2.4.

For the Cisco IMC lineup, the recommended version is as follows:

UCS-M4 servers	Recommended: Cisco IMC 2.0(13n) or later. It is also recommended to switch to 3.0(3a) or later for pure intel NIC based pods.
----------------	---

For the Cisco IMC 3.x lineup, the recommended version is as follows:

UCS-M4 servers	Cisco IMC versions are 3.0(3a) or later, except for 3.0(4a), 4.0(2c), or 4.0(2d). Recommended: Cisco IMC 3.0(4d). Expanded support of CIMC 4.0(1a), 4.0(1b), and 4.0(1c).
UCS-M5 servers	Recommended to use CIMC 3.1(2b). Do not use 3.1(3c) to 3.1(3h), 3.0(4a), 4.0(2c), or 4.0(2d). Expanded support of CIMC 4.0(1a), 4.0(1c), 4.0(2f), and 4.0(4d). The Bundle version of a minimum of CIMC 4.0(4d) is needed for Cascade Lake support.

Enables embedded server management for Cisco UCS C-Series Rack Servers. Supports Cisco IMC firmware versions of 2.0(13i) or greater for the fresh install of Cisco VIM. Because of recent security

fixes, we recommend you to upgrade Cisco IMC to 2.0(13n) or higher. Similarly, Cisco IMC version of 3.0 lineup is supported. For this, you must install Cisco IMC 3.0 (3a) or above.

The Quanta servers need to run with a minimum version of BMC and BIOS as listed below:

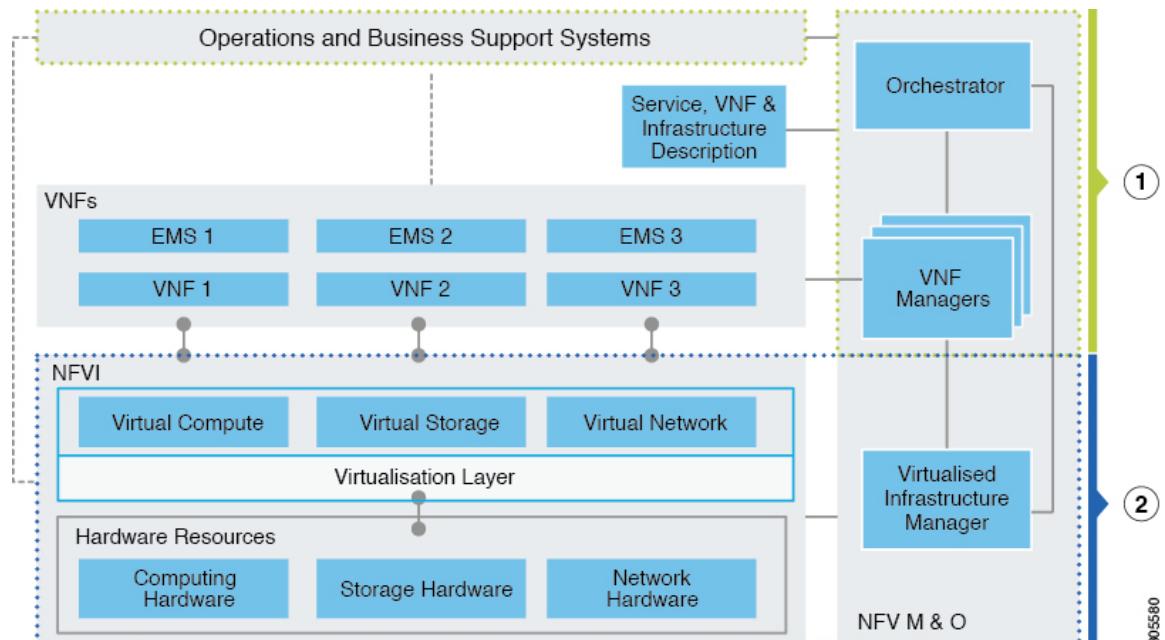
SKU Type	BMC Version	BIOS Version
D52BQ-2U 3UPI (CDC SKU)	4.54	3A11.BT10
D52BE-2U (GC SKU)	4.60	3A11.BT15

- Cisco Virtual Topology System (VTS)—It is an open, overlay management and provisioning system for data center networks. VTS automates DC overlay fabric provisioning for physical and virtual workloads. This is an optional service that is available through Cisco VIM.
- Cisco Virtual Topology Forwarder (VTF)—Included with VTS. VTF leverages Vector Packet Processing (VPP) to provide high performance Layer 2 and Layer 3 VXLAN packet forwarding.

Two Cisco VNF orchestration and management applications that are used with Cisco NFVI include:

- Cisco Network Services Orchestrator, enabled by Tail-f—Provides end-to-end orchestration spanning multiple network domains to address NFV management and orchestration (MANO) and software-defined networking (SDN). For information about Cisco NSO, see [Network Services Orchestrator Solutions](#).
- Cisco Elastic Services Controller—Provides a single point of control to manage all aspects of the NFV lifecycle for VNFs. ESC allows you to automatically instantiate, monitor, and elastically scale VNFs end-to-end. For information about Cisco ESC, see the [Cisco Elastic Services Controller Data Sheet](#).

Figure 2: NFVI Architecture With Cisco NFVI, Cisco NSO, and Cisco ESC



At a high level, the NFVI architecture includes a VNF Manager and NFV Infrastructure.

1	<ul style="list-style-type: none"> • Cisco Network Services Orchestrator • Cisco Elastic Services Controller
2	<p>Cisco NFVI:</p> <ul style="list-style-type: none"> • Cisco VIM + • Cisco UCS/Quanta/3rd Party Compute and Cisco Nexus Hardware + • Logging and Monitoring Software + • Cisco Virtual Topology Services (optional) + • Accelerated Switching with VPP (optional) • Cisco Unified Management (optional) • Pod Monitoring (optional)

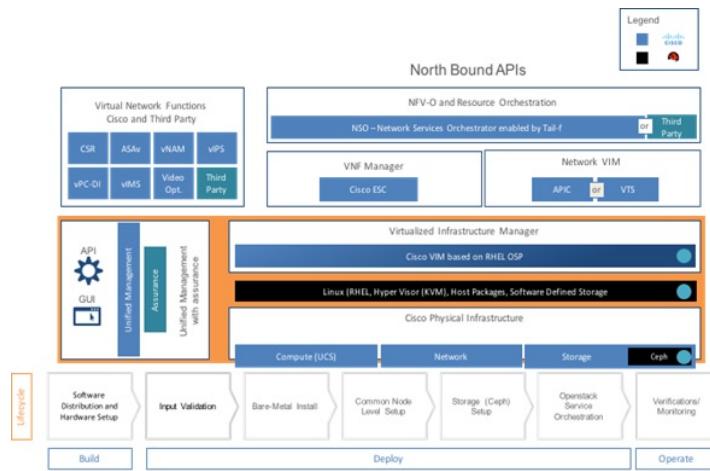
For cloud networking, Cisco NFVI supports Open vSwitch over VLAN as the cloud network solution for both UCS B-series and UCS C-Series pods. Both B-Series and C-Series deployments support provider networks over VLAN.

In addition, with a C-series pod, you can choose:

- To run with augmented performance mechanism by replacing OVS/LB with VPP/VLAN (for Intel NIC) or ACI/VLAN (virtual packet processor).
- To have cloud that is integrated with VTC which is an SDN controller option.

The Cisco NFVI uses OpenStack services running inside containers with HAProxy load balancing and providing high availability to API and management network messaging. Transport Layer Security (TLS) protects the API network from external users to the HAProxy. Cisco VIM installation also includes service assurance, OpenStack CloudPulse, built-in control, and data plane validation. Day two pod management allows you to add and remove both compute and Ceph nodes, and replace the controller nodes. The Cisco VIM installation embeds all necessary RHEL licenses as long as you use the Cisco VIM supported BOM and the corresponding release artifacts.

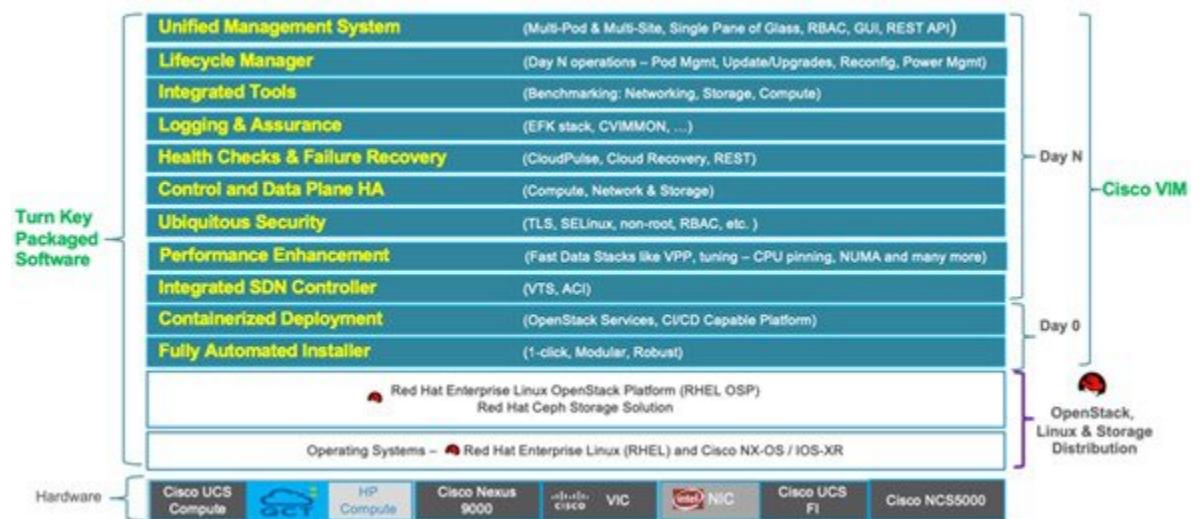
The following illustration shows a detailed view of the Cisco NFVI architecture and the Cisco NFVI installation flow.

Figure 3: Detailed Cisco NFVI Architecture View

Cisco Virtualized Infrastructure Manager Overview

Cisco Virtualized Infrastructure Manager (VIM) is a fully automated cloud lifecycle management system. Cisco VIM helps to bring up a fully functional cloud in hours, with integrated end-to-end control and data plane verification in place. Cisco VIM offers fully automated day 1 to day n cloud lifecycle management. These include capabilities such as pod scaling (expansion), software update, upgrade, or reconfigure parameters, consolidated logging with rotation and export, software update and upgrade. These have been implemented in line with the operational and security best practices of service providers and enterprises.

The following figure provides the high-level overview of all day-0 and day-n items of Cisco VIM.

Figure 4: Cisco VIM Capability Overview

Cisco VIM Features

Cisco VIM is the only standalone fully automated cloud lifecycle manager offering from Cisco for a private cloud. The current version of VIM integrates with Cisco UCS C-series (with or without HP as third-party Compute) or B-series, or Quanta (D52BQ-2U 3UPI or D52BE-2U) servers, and Cisco VIC or Intel NIC. This document and its accompanying administrator guide help the cloud administrators to set up and manage the private cloud.

Following are the features of the Cisco VIM:

Feature Name	Comments
OpenStack Version	RHEL 7.6 with OSP 13 (Queens).
Hardware Support Matrix	<ul style="list-style-type: none"> 1. UCS C220/B200 M4 controller or compute with Intel V3 (Haswell). 2. UCS C240/220 M4 controller or compute + Intel V4 (Broadwell). 3. HP DL360 Gen 9 with control plane on Cisco UCS M4 servers.. 4. UCS C220/240 M5 in Micropod environment, with an option to add up to 16 220/240-M5 computes. 5. UCS C240/220 M5 controller or compute with Intel X710 support with SRIOV and Cisco Nexus 9000 /Cisco NCS-5500 series switch as ToR. 6. Quanta servers as an alternate to Cisco UCS servers for Full on and edge deployment of the cloud. 7. Quanta servers for Central Ceph cluster for Edge pod to offer glance image services.
NIC support	<ul style="list-style-type: none"> 1. Cisco VIC: VIC 1227, 1240, 1340, 1380, 1387 (for M5) in 40G VIC/NIC offering. 2. Intel NIC: X710, 520, XL710, xxv710 (25G).

POD Type	
----------	--

1. Fullon: Dedicated control, compute, and storage (C-series) node running on Cisco VIC (M4) or Intel X710 (for M4 or M5) (full on) with Cisco Nexus 9000 or Cisco NCS 5500 series switch (only for Intel NIC and VPP as mechanism driver) as ToR. For fullon pods based on Quanta (D52BE-2U) servers, the NIC is xxv710 (25G) with Cisco Nexus 9000 as ToR.
 2. Dedicated control, compute, and storage (C-series) node running on Cisco VIC and Intel NIC (full on) with Cisco Nexus 9000 as ToR. Only SRIOV is supported on Intel NIC. Support of Intel X520 (with 2 NIC cards/compute) on M4 pods or XL710 (2 or 4 NIC cards/compute) on M4/M5 pods for SRIOV cards in the VIC/NIC combination. Few computes can run with/without SRIOV in a given pod. For M4 pods, VIC/NIC computes running XL710 and X520 can reside in the same pod.
 3. Dedicated control, compute, and storage (B-Series) node running on Cisco NIC.
 4. Micropod: Integrated (AIO) control, compute, and storage (C-series) node running on Cisco VIC, Intel X710X or VIC and NIC combo. Micropod can be optionally expanded to accommodate more computes (up to 16) running with the same NIC type. This can be done as a day-0 or day-1 activity. The computes can boot off HDD or SSD. Intel NIC-based Micropod supports SRIOV, with the M5-based Micropod supporting only XL710 as an option for SRIOV.
 5. Hyper-converged on M4(UMHC): Dedicated control and compute nodes, with all storage acting as compute nodes (M4 C-series) and running on a combination of 1-Cisco VIC (1227) and 2x10GE 520 or 2x40GE 710XL Intel NIC with an option to migrate from one to another. The pod can be extended to M5-based computes with 40G Cisco VIC along with 2x40GE 710XLNIC (optionally).
- Note** In a full-on (VIC based), or Hyper-Coverged pod, computes can either have a combination of 1-Cisco VIC (1227) and (2x10GE 520/2x40GE 710XL Intel NIC) or 1-CiscoVIC (1227). The compute running pure Cisco VIC does not run SR-IOV. In 2.4, Cisco supports HP DL360 Gen9 as a third party compute.
- Currently, we do not support a mix of computes from different vendors for the same pod.
6. NGENA Hyper-Converged (NGENAHC): Dedicated control and compute nodes, with all storage acting as compute (C-series) nodes. All nodes have a combination of 1-Cisco VIC (1227) for control plane, and 1x10GE 710X Intel NIC for Data plane over VPP.

	<p>7. Hyper-Converged on M5: Dedicated control and compute nodes, with all storage acting as compute (C-series) nodes, running on a combination of 1-Cisco VIC (40G) and 2x40GE 710XL Intel NIC.</p> <p>8. Edge: Available with restricted power and limited rack space. Quanta (D52BQ-2U 3UPI) servers with three converged control and compute nodes, expandable to 16 additional compute nodes. The edge cloud communicates with Quanta server based Central Ceph cluster for glance service. Persistent storage is not available.</p> <p>9. Ceph: Designed to provide glance image services to edge cloud. Quanta (D52BE-2U) servers with three converged ccephontrol and cephosd nodes, expandable to additional cephosd nodes for additional storage.</p>
ToR and FI support	<ol style="list-style-type: none"> 1. For VTS-based installation, use the following Nexus version: 7.0(3)I2(2a) and 7.0(3)I6(2). 2. For the mechanism driver other than VTS, use the following Nexus software version: 7.0(3)I4(6) 7.0(3)I6(1). If you are using auto-ToR configuration and CONFIGURE_TORS set to True, the nxos version - 7.0(3)I6(1) automation fails irrespective of the mechanism driver due to the defect CSCve16902. 3. UCS-FI-6296. 4. Support of Cisco NCS 5500 (with recommended Cisco IOS XR version 6.1.33.02I or 6.5.1) with splitter cable option. Also, extending day-0 configuration to support user defined route-target and ethernet segment id (ESI) 5. Cisco Nexus 9000 series switches running ACI 3.2 (4d) with plugin version of 4.0.1 (for the mechanism driver ACI).
IPv6 Support for Management Network	<ol style="list-style-type: none"> 1. Static IPv6 management assignment for servers 2. Support of IPv6 for NTP, DNS, LDAP, external syslog server, and AD. 3. Support of IPv6 for the Cloud API endpoint. 4. Support of CIMC over IPv6 5. RestAPI over IPv6 6. Support of UM over IPv6
Mechanism Drivers	OVS/VLAN, ACI/VLAN, VPP/VLAN (Fast Networking, Fast Data FD.io VPP/VLAN, based on the FD.io VPP fast virtual switch over intel NIC).

SDN controller integration	VTS 2.6.2 with optional feature of Managed VTS; ACI (ships in the night, or ToR automation via APIC API or with Unified ACI Plugin) with Cisco VIC or Intel NIC on the UCS C-series M4 platform.
Install methodology	<ul style="list-style-type: none"> • Fully automated online or offline installation. • Support of Cisco VIM Software Hub to mitigate the problem associated with logistics of USB distribution for air-gapped installation. • Support of USB 3.0 64GB for M5 and Quanta based Management node. Support of UCS 2.0 64GB for M4 based management node.
Scale	<ol style="list-style-type: none"> 1. LA: Total of 120 nodes (compute and OSD) with Ceph OSD max at 20. <p>Note It is recommended to deploy 30 nodes at a time. Also, after day-0, you can add only one ceph node at a time.</p> 2. Micropod: Maximum of 16 standalone compute nodes. <p>Note Ceph OSDs can be either HDD or SSD based across the pod. Computes can boot off 2x1.2TB HDD or 2x960GB SSD). In the same pod, some computes can have SSD, while others can have HDD.</p>
Automated pod life cycle management	<ol style="list-style-type: none"> 1. Add or remove compute and Ceph nodes and replace the controller node. 2. Static IP management for storage network 3. Reduction of tenant/provider VLAN via reconfiguration to a minimum of two. 4. Reconfiguration of passwords and selected optional services. 5. Automated software update

Platform security	<ul style="list-style-type: none"> Secure OS, RBAC, network isolation, TLS, source IP filtering, Keystone v3, Bandit, CSDL-compliant, hardened OS, SELinux. Change the CIMC password post install for maintenance and security. Non-root log in for Administrators. Read-only role available for OpenStack users. Enabling custom policy for VNF Manager. Optionally, you can disable the reachability of the management node to the cloud API network. Hosting of Horizon behind NAT or with a DNS alias Cinder volume encryption via LUKS Support of configurable login banner for SSH sessions Access of NFVIMON via non-root user
EPA	NUMA, CPU pinning, huge pages, SRIOV with Intel NIC. Ability to allocate user defined CPU (upto 6) cores to VPP. Ability to allocate user defined CPU (upto 12) cores to Ceph for Micropod and hyper-converged nodes.
HA and Reliability	<ol style="list-style-type: none"> Redundancy at hardware and software level. Automated backup and restore of the management node.
Unified Management Support	Single pane of glass in a single or multi instance (HA) mode. Supports multi-tenancy and manages multiple pods from one instance.
Central Logging	ELK integrated with external syslog (over v4 or v6) for a log offload, with optional support of NFS with ELK snapshot.
External Syslog Servers	Support of multiple external syslog servers over IPv4 or IPv6. The minimum and the maximum number of external syslog servers that is supported is 1 and 3, respectively
VM Migration	Cold migration and resizing. Live Migration

Storage	<ul style="list-style-type: none"> Object store with SwiftStack, Block storage with Ceph, or NetApp. Option to use Ceph for Glance and SolidFire for Cinder. Option to have multi-backend (HDD and SSD based) Ceph in the same cluster to support various I/O requirements and latency.
Monitoring	<ul style="list-style-type: none"> CVIM-MON for monitoring, as a Cisco solution over v4 and/or v6. Ceilometer for resource tracking and alarming capabilities across core OpenStack components is applicable only for fullon pod. Third-party integration with Zenoss (called NFVIMON)
Optional OpenStack features	<ul style="list-style-type: none"> Enable trusted Virtual Function on a per server basis DHCP reservation for virtual MAC addresses
Support of External Auth System	<ol style="list-style-type: none"> LDAP with anonymous bind option. Active Directory (AD)
Software Update	Update of Cloud Software for bug fixes on the same release.
Software Upgrade	Upgrade of non-VTS cloud from release 3.0.0 to release 3.2.0.
CIMC Upgrade Capability	Central management tool to upgrade the CIMC bundle image of one or more servers.
VPP port mirroring	Ability to trace or capture packets for debugging and other administrative purposes.
VXLAN extension into the cloud	<p>Extending native external VXLAN network into VNFs in the cloud.</p> <p>Support of Layer 3 adjacency for BGP.</p> <p>Support of single VXLAN or multi-VXLAN (with head-end replication as an option) network terminating on the same compute node.</p> <p>Note Only two-VXLAN network is supported for now.</p>
Power Management of Computes	Option to power off or on computes selectively to conserve energy.
Technical support for CIMC	Collection of technical support for CIMC.
Enable TTY logging as an option	Enables TTY logging and forwards the log to external syslog server and ELK stack running on management node. Optionally, log is sent to remote syslog if that option is available

Power management of computes	Option to selectively turn OFF or ON the power of computes to conserve energy
Unified Management Authentication	Supports authentication through local and LDAP.
Support of workload types	Extending Cisco VIM to support bare-metal (ironic based) and container (Cisco Container Platform (CCP)) based workloads.
Cloud adaptation for low latency workload	<ul style="list-style-type: none"> • Enable real-time kernel to support on edge pod • Automated BIOS configuration • Custom flavor is supported
Automated enablement of Intel X710/XL710 NIC's PXE configuration on Cisco UCS-C series	Utility to update Intel X710/XL710 NIC's PXE configuration on Cisco UCS-C series.
Disk maintenance for Pod Nodes	Ability to replace faulty disks on the Pod nodes without the need for add, remove or replace node operation.
Integrated Test Tools	<ol style="list-style-type: none"> 1. Open Source Data-plane Performance Benchmarking: VMTP (an open source data plane VM to the VM performance benchmarking tool) and NFVbench (NFVI data plane and a service chain performance benchmarking tool). Extending VMTP to support v6 over provider network. 2. NFVbench support for VXLAN. 3. Services Health Checks Integration: Cloudpulse and Cloudsanity.

**Note**

Configure the LACP on the data plane ports of the Cisco Nexus 9000 ToR, when Cisco VIM is running on Intel NIC for data plane with VPP as the mechanism driver. When Cisco NCS 5500 is the ToR (with mechanism driver VPP), the LACP configuration on the data plane is done through the Auto-ToR configuration feature of Cisco VIM.

Cisco VIM Networking Overview

Cisco VIM supports installation on two different type of pods. The blade B-series and rack C-series based offering supports NICs that are from Cisco (called as Cisco VIC). You can choose the C-series pod to run in a pure Intel NIC environment, and thereby obtain SRIOV support on the C-series pod. This section calls out the differences in networking between the Intel NIC and Cisco VIC installations.

To achieve network level security and isolation of tenant traffic, Cisco VIM segments the various OpenStack networks. The Cisco NFVI network includes six different segments in the physical infrastructure (underlay). These segments are presented as VLANs on the Top-of-Rack (ToR) Nexus switches (except for the provider network) and as vNIC VLANs on Cisco UCS servers. You must allocate subnets and IP addresses to each

segment. Cisco NFVI network segments include: API, external, management and provisioning, storage, tenant and provider.

API Segment

The API segment needs one VLAN and two IPv4 addresses (four if you are installing Cisco VTS) in an externally accessible subnet different from the subnets assigned to other Cisco NFVI segments. These IP addresses are used for:

- OpenStack API end points. These are configured within the control node HAProxy load balancer.
- Management node external connectivity.
- Cisco Virtual Topology Services (VTS) if available in your Cisco NFVI package.
- Virtual Topology Controller (VTC). It is optional for VTS.

External Segment

The external segment needs one VLAN to configure the OpenStack external network. You can provide the VLAN during installation in the Cisco NFVI setup_data.yaml file, but you must configure the actual subnet using the OpenStack API after the installation. Use the external network to assign OpenStack floating IP addresses to VMs running on Cisco NFVI.

Management and Provisioning Segment

The management and provisioning segment needs one VLAN and one subnet with an address pool large enough to accommodate all the current and future servers planned for the pod for initial provisioning (PXE boot Linux) and, thereafter, for all OpenStack internal communication. This VLAN and subnet can be local to Cisco NFVI for C-Series deployments. For B-Series pods, the UCS Manager IP and management network must be routable. You must statically configure Management IP addresses of Nexus switches and Cisco UCS server Cisco IMC IP addresses, and not through DHCP. They must be through the API segment. The management/provisioning subnet can be either internal to Cisco NFVI (that is, in a lab it can be a non-routable subnet limited to Cisco NFVI only for C-Series pods), or it can be an externally accessible and routable subnet. All Cisco NFVI nodes (including the Cisco VTC node) need an IP address from this subnet.

Storage Segment

Cisco VIM has a dedicated storage network used for Ceph monitoring between controllers, data replication between storage nodes, and data transfer between compute and storage nodes. The storage segment needs one VLAN and /29 or larger subnet internal to Cisco NFVI to carry all Ceph replication traffic. All the participating nodes in the pod will have IP addresses on this subnet.

Tenant Segment

The tenant segment needs one VLAN and a subnet large enough to manage pod tenant capacity internal to Cisco NFVI to carry all tenant virtual network traffic. Only Cisco NFVI control and compute nodes have IP addresses on this subnet. The VLAN/subnet can be local to Cisco NFVI.

Provider Segment

Provider networks are optional for Cisco NFVI operations but are often used for real VNF traffic. You can allocate one or more VLANs for provider networks after installation is completed from OpenStack.

Cisco NFVI renames interfaces based on the network type it serves. The segment Virtual IP (VIP) name is the first letter of the segment name. Combined segments use the first character from each segment for the VIP, with the exception of provisioning whose interface VIP name is "mx" instead of "mp" to avoid ambiguity with the provider network. The following table shows Cisco NFVI network segments, usage, and network and VIP names.

Table 2: Cisco NFVI Networks

Network	Usage	Network Name	VIP Name
Management/Provisioning	<ul style="list-style-type: none"> OpenStack control plane traffic. Application package downloads. Server management; management node connects to servers on this network. Host default route. PXE booting servers during bare metal installations. 	Management and provisioning	mx
API	<ul style="list-style-type: none"> Clients connect to API network to interface with OpenStack APIs. OpenStack Horizon dashboard. Default gateway for HAProxy container. Integration with endpoints served by SwiftStack cluster for native object storage, cinder backup service or Identity service with LDAP or AD. 	api	a
Tenant	VM to VM traffic. For example, VXLAN traffic.	tenant	t
External	Access to VMs using floating IP addresses.	external	e
Storage	Transit network for storage back-end. Storage traffic between VMs and Ceph nodes.	storage	s
Provider Network	Direct access to existing network infrastructure.	provider	p
ACIINFRA	Internal ACI Network for Policy management (only allowed when deployed with ACI)	aciinfra	o
Installer API	<ul style="list-style-type: none"> Administrator uses installer API network to ssh to the management node. Administrator connects to installer API to interface with secured services. For example, Kibana on the management node. 	VIM installer API	br_api

For each C-series pod node, two vNICs are created using different ports and bonded for redundancy for each network. Each network is defined in setup_data.yaml using the naming conventions listed in the preceding table. The VIP Name column provides the bonded interface name (for example, mx or a) while each vNIC name has a 0 or 1 appended to the bonded interface name (for example, mx0, mx1, a0, a1).

The Cisco NFVI installer creates the required vNICs, host interfaces, bonds, and bridges with mappings created between all elements. The number and type of created vNICs, interfaces, bonds, and bridges depend on the Cisco NFVI role assigned to the UCS server. For example, the controller node has more interfaces than

the compute or storage nodes. The following table shows the networks that are associated with each Cisco NFVI server role.

Table 3: Cisco NFVI Network-to-Server Role Mapping

	Management Node	Controller Node	Compute Node	Storage Node
Management/Provisioning	+	+	+	+
ACIINFRA*		+	+	
API		+		
Tenant		+	+	
Storage		+	+	+
Provider			+	
External		+		



Note

*ACIINFRA is only applicable when using ACI as a mechanism driver.

** Provider network is extended to controller nodes, when VMs are on provider network with virtio.

The network arrangement on third-party HP compute is slightly different from that of Cisco compute running with Intel NIC, because the HP computes have 2 less NIC ports than that are available in the Cisco Intel NIC BOM.

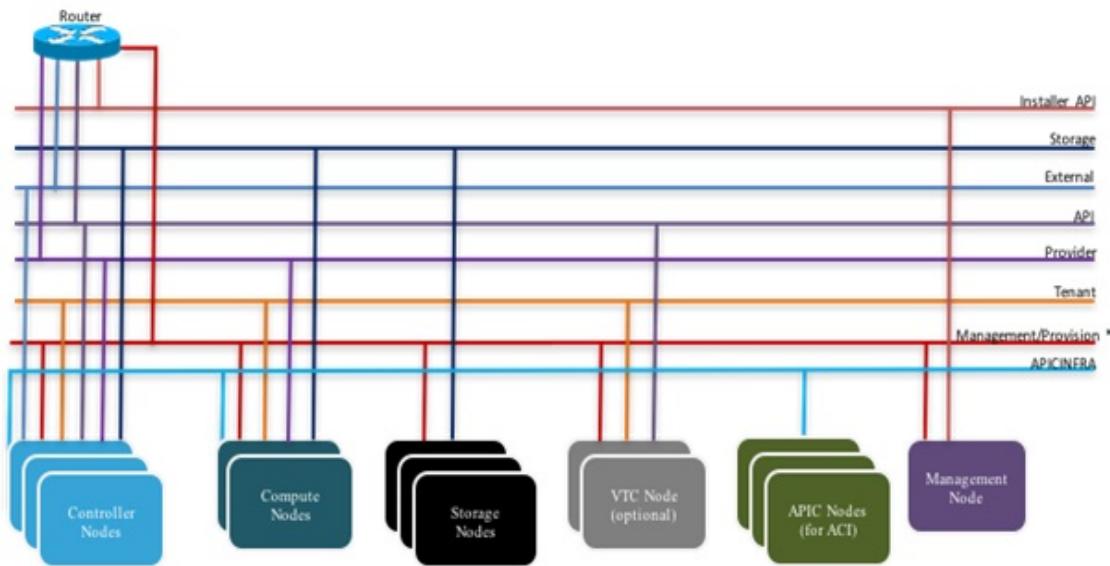
Following table lists the differences in the network arrangement between the Cisco compute and third-party HP compute.

Network Interface	Cisco UCS Ce220/Ce240M4/M5 Compute	HPE ProLiant DL360 Gen9 and Quanta Compute
mx	Management control plane network	N/A

Network Interface	Cisco UCS Ce220/Ce240M4/M5 Compute	HPE ProLiant DL360 Gen9 and Quanta Compute
samxpet		<p>Control and data plane network for everything other than SRIOV:</p> <ol style="list-style-type: none"> 1. Management network on "br_mgmt" bridge interface with "samxpet" main interface as one of the member interface (native VLAN configuration required on the top-of-rack switches) 2. Storage network on the sub-interface "samxpet.<storage VLAN>" 3. Tenant and provider networks on veth interface "pet/pet-out" as one of the member interface with "br_mgmt" bridge interface
p	Provider data plane network	
sriov[0-3]	Provider data plane SRIOV networks	Provider data plane SRIOV networks
s	Storage control and data plane network	N/A
t	Tenant data plane network	N/A

In the initial Cisco NFVI deployment, two bridges are created on the controller nodes, and interfaces and bonds are attached to these bridges. The br_api bridge connects the API (a) interface to the HAProxy. The HAProxy and Keepalive container has VIPs running for each OpenStack API endpoint. The br_mgmt bridge connects the Management and Provisioning (mx) interface to the HAProxy container as well.

The following diagram shows the connectivity between Cisco NFVI nodes and networks.

Figure 5: Cisco NFVI Network Connectivity

* For C series, Cisco VIM Non-routable is recommended.
For B series, UCSM IP should be reachable from the management network.

Supported Layer 2 networking protocols include:

- VLAN over Open vswitch(SRIOV with Intel 710NIC).
- VLAN over VPP/VLAN for C-series Only.
- Single Root Input/Output Virtualization (SRIOV) for UCS B-Series pods. SRIOV allows a single physical PCI Express to be shared on a different virtual environment. The SRIOV offers different virtual functions to different virtual components, for example, network adapters, on a physical server.

The footprint of the cloud offering supported by Cisco VIM has continued to evolve over multiple releases to support customer needs that can vary across multiple dimensions such as cloud capacity, power, physical space, and affordability. The following table shows the available Cisco NFVI hardware and data path deployment combinations.

Table 4: Cisco NFVI Hardware and Data Path Deployment Combination

POD Type	NIC Type	Hardware Vendor	Mechanism Driver	TOR Type
fullon	Cisco VIC	UCS C series M4 UCS C series M5	OVS/VLAN	N9K
fullon	Cisco VIC	UCS B Series	OVS/VLAN with SRIOV	N9K
fullon	Cisco VIC	UCS C series M4	VTF with VTC (VXLAN)	N9K

POD Type	NIC Type	Hardware Vendor	Mechanism Driver	TOR Type
fullon	Cisco VIC	UCS C series M4	ACI	N9K
fullon	Intel NIC	UCS C series M4	ACI with SRIOV	N9K
fullon	Intel NIC	UCS C series M4 UCS C series M5	OVS/VLAN with SRIOV	N9K
fullon	Intel NIC	Quanta D52BQ-2U 3UPI	OVS/VLAN with SRIOV	N9K
fullon	Intel NIC	UCS C series M4 UCS C series M5	VPP/VLAN with SRIOV	N9K NCS-5500
fullon	VIC for Control & Intel NIC for Data Plane	UCS C series M4 with HP as third-party Compute	OVS/VLAN with SRIOV	N9K
fullon	Cisco VIC with Intel NIC	UCS C series M4/M5 computes UCS C series M5	OVS/VLAN (VIC) with SRIOV (Intel NIC)	N9K
micro	Cisco VIC	UCS C series M4 UCS C series M5	OVS/VLAN	N9K
micro	Intel NIC	UCS C series M4 UCS C series M5	OVS/VLAN	N9K
micro	Intel NIC	UCS C series M4 UCS C series M5	VPP/VLAN	N9K NCS-5500
UMHC	Cisco VIC with Intel NIC	UCS C series M4 UCS C series M5	OVS/VLAN (VIC) with SRIOV (Intel NIC)	N9K
NGENAHC	VIC for Control & Intel NIC for Data Plane	UCS C series M4	VPP/VLAN	N9K
edge	Intel NIC	Quanta D52BE-2U	OVS/VLAN with SRIOV	N9K
ceph	Intel NIC	Quanta D52BQ-2U 3UPI	N/A	N9K

In the above table:

- fullon indicates the dedicated control, compute and ceph nodes.
- micro indicates converged control, compute and ceph nodes with expandable computes.

- Hyperconverged (HC) indicates dedicated control and compute nodes, but all ceph nodes are compute nodes.
- edge indicates converged control and compute nodes with expandable computes. It communicates with Central ceph cluster for Glance Image service. Persistent storage is not supported.
- ceph indicates converged cephcontrol & cephosd nodes, with an option to add cephosd nodes for glance image services.



Note The SRIOV support is applicable only for Intel NIC-based pods.



Note VTF with VTC is only supported on C-series Cisco VIC.

Pod with Intel NICs— In case of the pod having Intel NICs (X710), the networking is slightly different. You need to have atleast two NICs (4x10G) on a single server to support NIC level redundancy. Each NIC is connected to each ToR (connections explained later in the chapter). Since vNICs are not supported in the Intel card, bond the physical interfaces at the host and then create sub-interfaces based on the segment VLAN. Lets call the two NIC cards as NIC_1 and NIC_2 and call their four ports as A, B, C, D. Unlike Cisco VIC based pod, the traffic here is classified as follows:

1. Control plane
2. Data plane (external, tenant and non-SRIOV provider network).
3. SRIOV (optional for provider network). If SRIOV is used, the data plane network only carries external and tenant network traffic.

Control Plane

The control plane is responsible for carrying all the control and management traffic of the cloud. The traffic that flows through control plane are:

1. Management/Provision
2. Storage
3. API

The control plane interface is created by bonding the NIC_1 A port with NIC_2 A port. The bonded interface name is called as samx, indicating that it is carrying Storage, API, Management/Provision traffic (naming convention is similar to Cisco VIC pod). The slave interfaces (physical interfaces) of the bonded interface are renamed as samx0 and samx1. samx0 belongs to NIC_1 and samx1 belongs to NIC_2. Sub interfaces are then carved out of this samx interface based on the Storage, API VLANs. The management/provision traffic will be untagged/native VLAN in order to support pxe booting.

Data Plane

The data plane is responsible for carrying all the VM data traffic. The traffic that flows through the data plane are

- Tenant

- Provider
- External

The data plane is created by bonding the NIC_1 B port with NIC_2 B port. The bonded interface name here would be pet, indicating that it is carrying Provider, External and Tenant traffic. The slave interfaces of this bonded interface would be visible as pet0 and pet1. pet0 belongs to the NIC_1 and pet1 belongs to NIC_2.

In case of OVS/VLAN, the "pet" interface is used as it is (trunked to carry all the data VLANs) to the Openstack cloud, as all the tagging and untagging happens at the Openstack level. In case of Linux Bridge/VXLAN, there will be sub-interface for tenant VLAN to act as the VXLAN tunnel endpoint.

SRIOV

In case of Intel NIC pod, the third (and optionally the fourth) port from each NIC can be used for SRIOV traffic. This is optional and is set or unset through a setup_data.yaml parameter. Unlike the control and data plane interfaces, these interfaces are not bonded and hence there is no redundancy. Each SRIOV port can have maximum of 32 Virtual Functions and the number of virtual function to be created are configurable through the setup_data.yaml. The interface names of the SRIOV will show up as sriov0 and sriov1 on each host, indicating that sriov0 belongs to NIC_1 C port and sriov1 belongs to NIC_2 C port.

In the case of Intel NIC pod, the following table summarizes the above discussion

Network	Usage	Type of traffic	Interface name
Control Plane	To carry control/management traffic	Storage, API, Management/Provision	samx
Data Plane	To carry data traffic	Provider, External, Tenant	pet
SRIOV	To carry SRIOV traffic	SRIOV	sriov0, sriov1

The following table shows the interfaces that are present on each type of server (role based).

	Management Node	Controller Node	Compute Node	Storage Node
Installer API	+			
Control plane	+	+	+	+
Data plane		+	+	
SRIOV			+	



Note On an Intel pod, all kind of OpenStack networks are created using the **physnet1** as the physnet name.

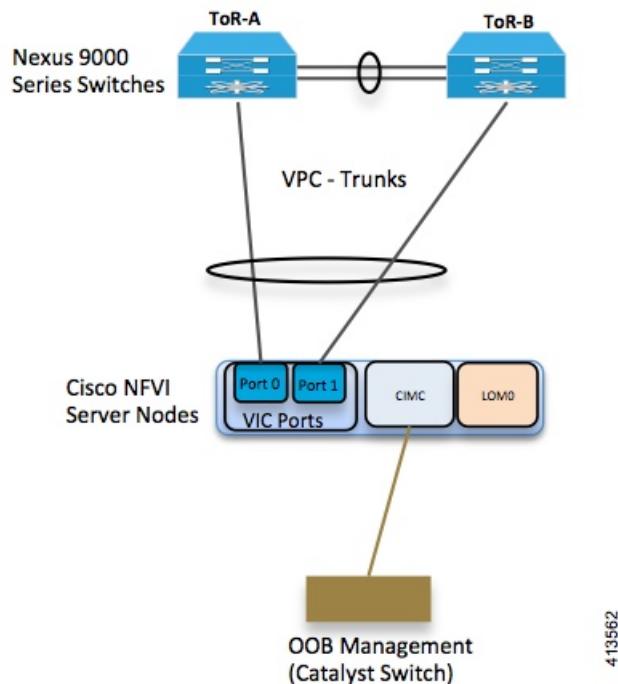
UCS C-Series Network Topologies

Cisco NFVI UCS servers connect to the ToR switches using Cisco UCS dual-port Virtual Interface Cards (VICs). The VIC is an Enhanced Small Form-Factor Pluggable (SFP+) 10 Gigabit Ethernet and Fiber Channel

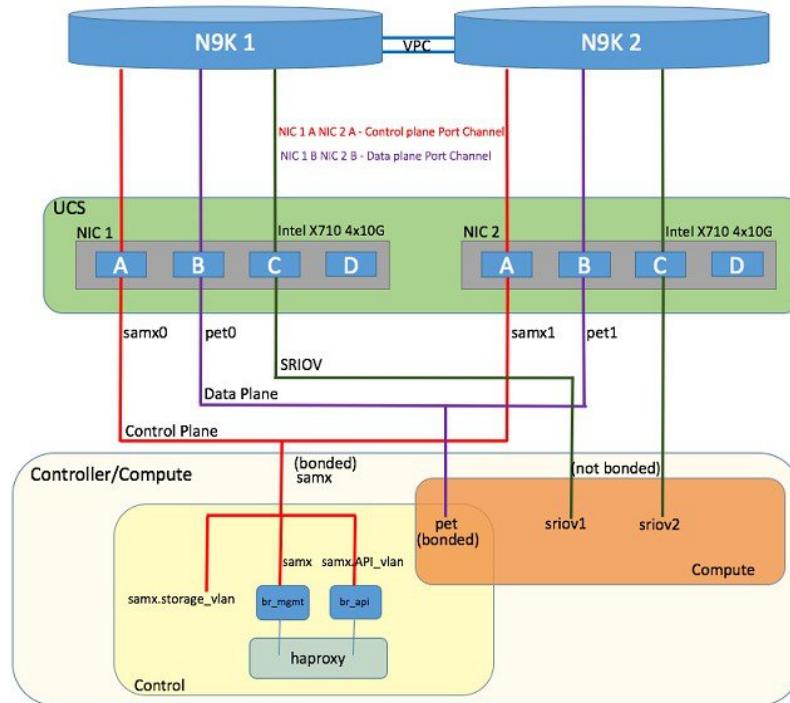
over Ethernet (FCoE)-capable PCI Express (PCIe) card designed for Cisco UCS C-Series Rack Servers. Each port connects to a different ToR using a Virtual Port Channel (VPC). Each VIC is configured with multiple vNICs that correspond to specific Cisco VIM networks. The UCS Cisco IMC port is connected to an out-of-band (OOB) Cisco management switch.

The following figure shows the UCS C-Series pod Cisco NFVI host to ToR topology.

Figure 6: UCS C-Series Host to ToR Topology



In the case of Intel NIC, a single two port Cisco VIC in the preceding diagram, is replaced with two 4-port 710 Intel NIC. An extra Intel NIC is added to provide card level redundancy.

Figure 7: UCS C-Series Intel NIC Details

Of the four ports that are available in each NIC card, port A is used for management traffic (provision, API, storage, etc), whereas the port B is used for data plane (tenant and provider network) traffic. Port C (and optionally Port D) is dedicated for SRIOV (configured optionally based on setup_data.yaml). Sub-interfaces are carved out of the data and control plane interfaces to provide separate traffic based on specific roles. While the ports A and B from each NIC help in forming bonded interface, the ports C and D over which SRIOV traffic for provider network flows is not bonded. Extreme care should be taken during pod setup, so that ports A, B and C for the Intel NIC is connected to the ToRs. Port D can be optionally used as a second pair of SRIOV ports by appropriate intent defined in the setup_data.yaml file. From Cisco VIM release 2.4.2 onwards, this port option is available for both M4 and M5 based systems or pods.

The following table provides the default link aggregation member pairing support for the pods based on server type:

Table 5: Default Link Aggregation Members Pairing

Server/POD Type	Target Functions	Default NIC Layout
M4 Intel NIC based	Control Plane	NIC-1 A + NIC-2 A
	Data Plane	NIC-1 B + NIC-2 B
	SRIOV 0/1	NIC-1 C + NIC-2 C
	SRIOV 2/3	NIC-1 D + NIC-2 D

Server/POD Type	Target Functions	Default NIC Layout
M5 Intel NIC based	Control Plane	NIC-1 A + NIC-1 B
	Data Plane	NIC-1 C + NIC-1 D
	SRIOV 0/1	NIC-2 A + NIC-2 B
	SRIOV 2/3	NIC-2 C + NIC-2 D



Note In M5, a NIC_LEVEL_REDUNDANCY option is introduced to support the M4 default option for link aggregation settings.

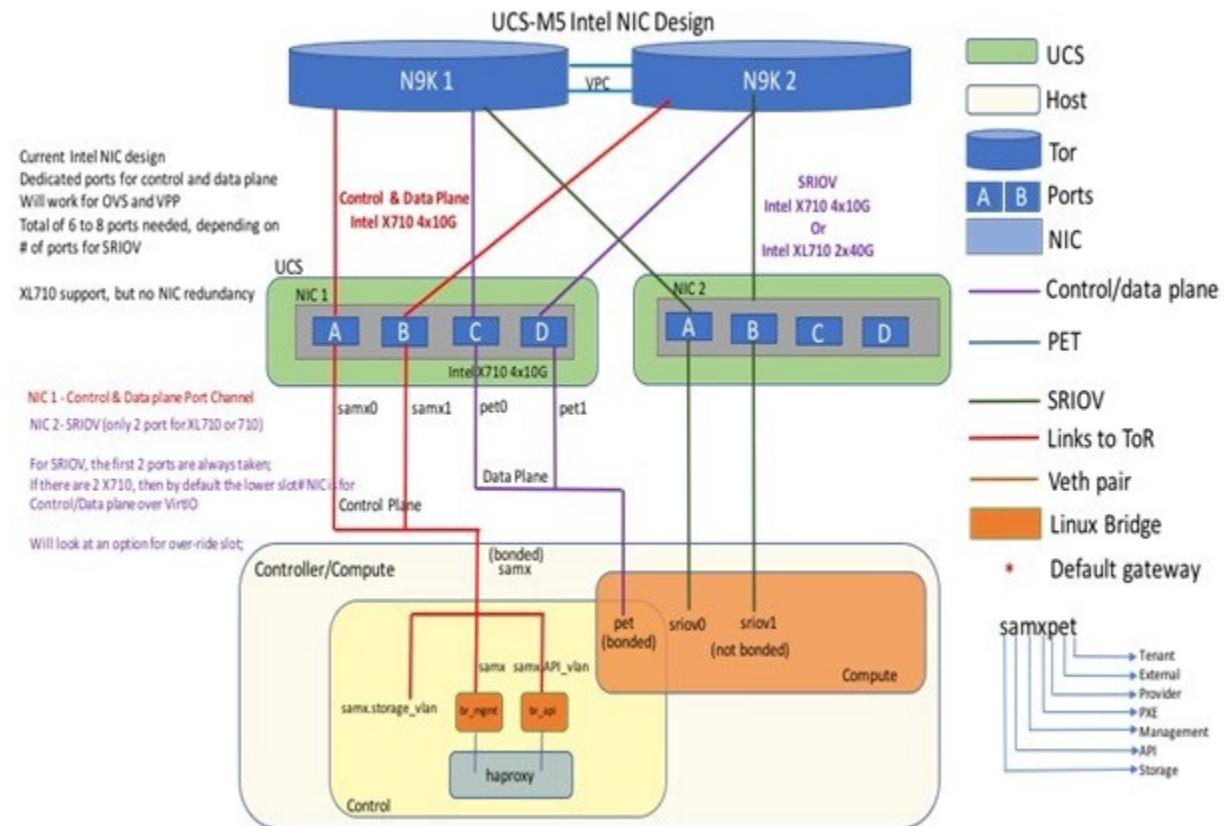
From Cisco VIM 2.4.2 onwards, support of M5 full on pods with two port XL-710 across control, compute and dedicated Ceph Nodes, and with NIC_LEVEL_REDUNDANCY is available. This deployment can be achieved with Cisco Nexus 9000 series or Cisco NCS 5500 as ToR. SRIOV is not supported in computes with XL-710. However, the pod can also support computes with four port X-710, where SRIOV is over port C and D.

In Cisco VIM, computes (M4 based testbed) running a Cisco 1227 VIC, and 2 2-port Intel 520 NIC are supported. In this combination, SRIOV is running on the Intel NIC, whereas the control and data plane are carried by virtual interfaces over Cisco VIC.

Cisco VIM 2.4 introduces the support of C220/C240 M5 servers in a micropod configuration with an option to augment the pod with additional computes (upto a max of 16). The M5 micropod environment is based on X710 for control and data plane and an additional XL710 or 2xX710 for SRIOV. The SRIOV card is optional. Once the SRIOV card is chosen, all the computes must have same number of SRIOV ports across the pod.

The following diagram depicts the server network card diagram for the M5 setup.

Figure 8: Networking Details of UCS-M5 Micropod Deployment



Cisco VIM 2.4 introduces the first third-party compute. The first SKU chosen is HPE ProLiant DL360 Gen9. In Cisco VIM 2.4, the supported deployment is a full-on pod, with OVS as the mechanism driver, where the management, control, and storage nodes are based on existing Cisco UCS c220/240M4 BOM, and the compute nodes are on HPE ProLiant DL360 Gen9 hardware:

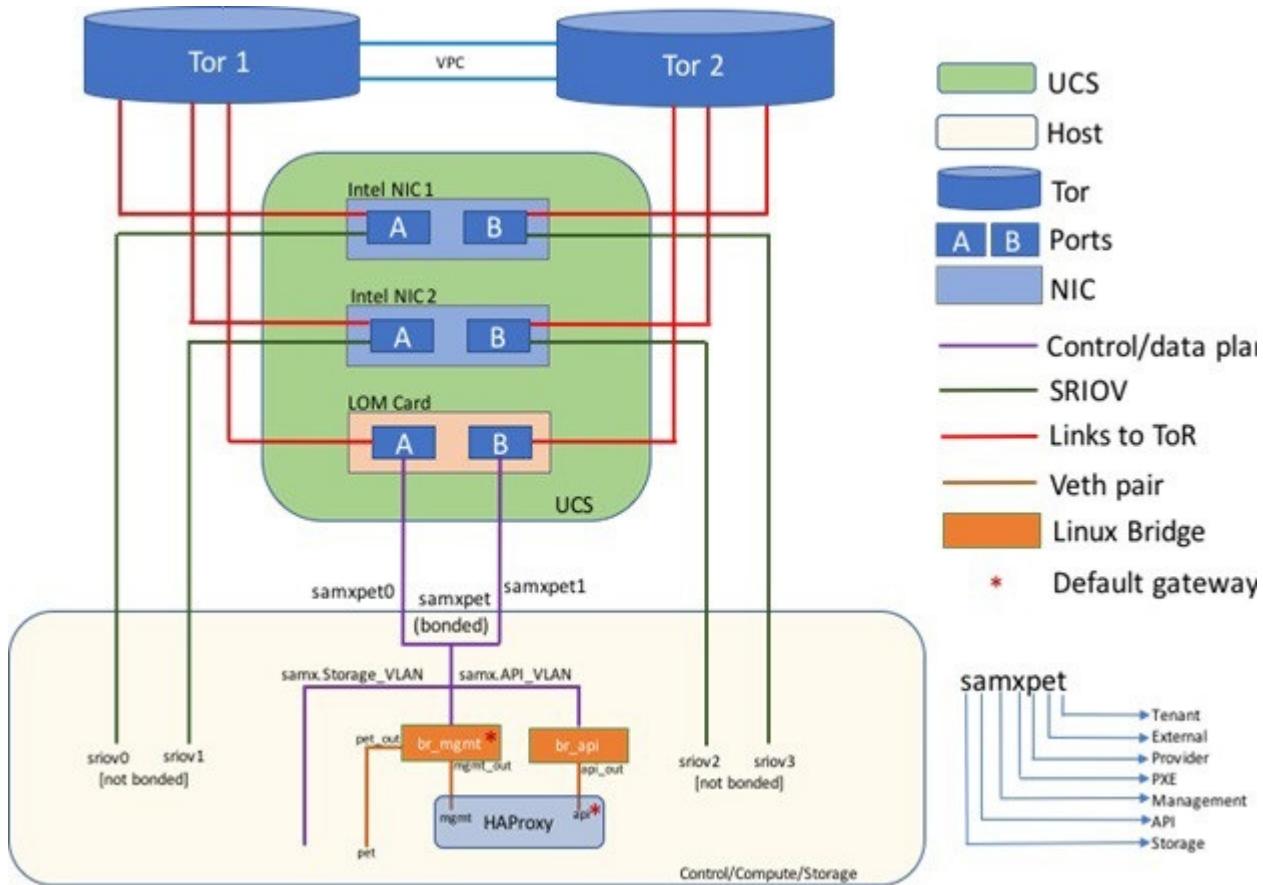
```

ProLiant DL360 Gen9 with HP Ethernet 1Gb 4-port 331i Adapter - NIC (755258-B21) 2 x E5-2695
v4 @ 2.10GHz CPU
8 x 32GB DDR4 memory (Total 256GB)
1 x Smart Array P440ar hardware RAID card with battery
2 x 1.2 TB - SAS 12GB/S 10k RPM HDD
1 x FlexLOM HP Ethernet 10Gb 2-port 560FLR-SFP+ Adapter
2 x PCIe HP Ethernet 10Gb 2-port 560SFP+ Adapter
System ROM: P89 v2.40 (02/17/2017)
iLO Firmware Version: 2.54 Jun 15 2017

```

In the case of HP Computers, the FlexLOM HP Ethernet 10Gb interface is used for management and tenant network, and the two additional HP Ethernet 10Gb 2-port 560SFP+ Adapters are used for SRIOV for the provider network. Listed below is network schematic of the HP Compute node.

Figure 9: Networking details of HP DL360GEN9



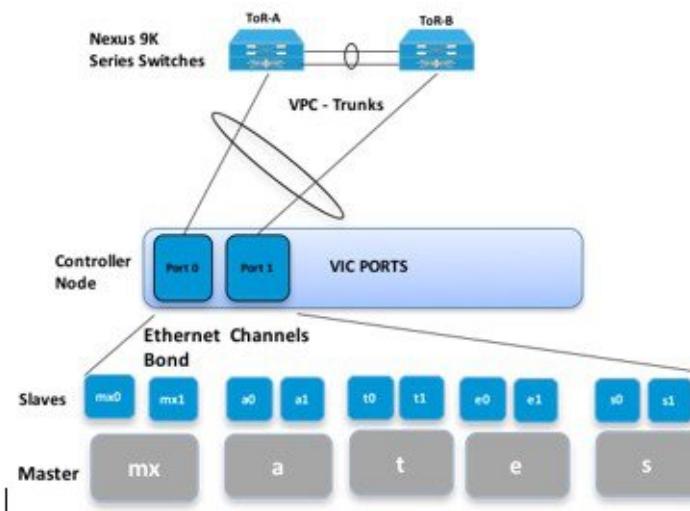
The Cisco NFVI controller node has four bonds: mx, a, t, and e. Each has a slave interface that is named with the network name association and a mapped number. For example, the management and provisioning network, mx, maps to mx0 and mx1, the API network, a, to a0 and a1, and so on. The bonds map directly to the vNICs that are automatically created on the controller node when it is deployed.

Cisco VIM 3.0 manages a third-party infrastructure based on Quanta servers, thereby bringing in true software abstraction. In the implementation, the supported deployment is a full-on or edge pod, with OVS as the mechanism driver. With the power limitation and rack restrictions on the edge pod, it cannot support hard-drives for the Ceph service. As the Edge pod does not need persistent storage, it is designed to communicate with a central ceph cluster for providing glance image services only.

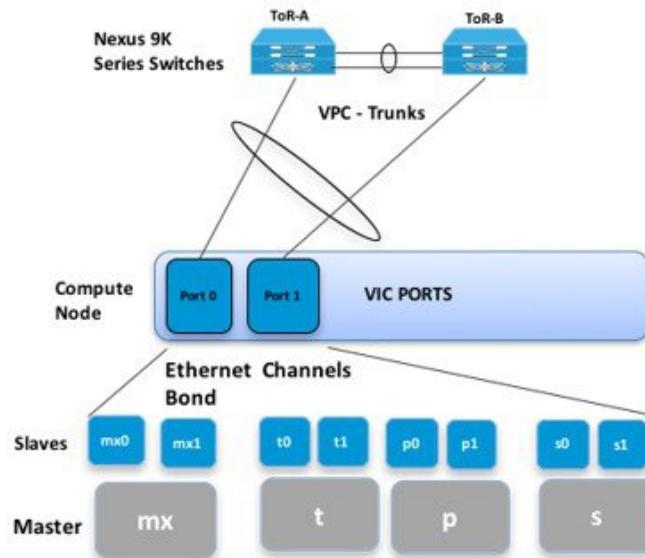
The installation and management of the Central Ceph cluster is fully automated and it is assumed that the management network of the edge cloud is routable to that of the central Ceph cluster.

In the case of Quanta servers, the networking is similar to that of the HP computes except for the two port 25G (xxv710) Intel NICs. The 2x25GE OCP card is used for control and data plane network over virtio, and the two additional 25GE 2-port xxv710 based Intel NIC Adapters are used for SRIOV via the provider network.

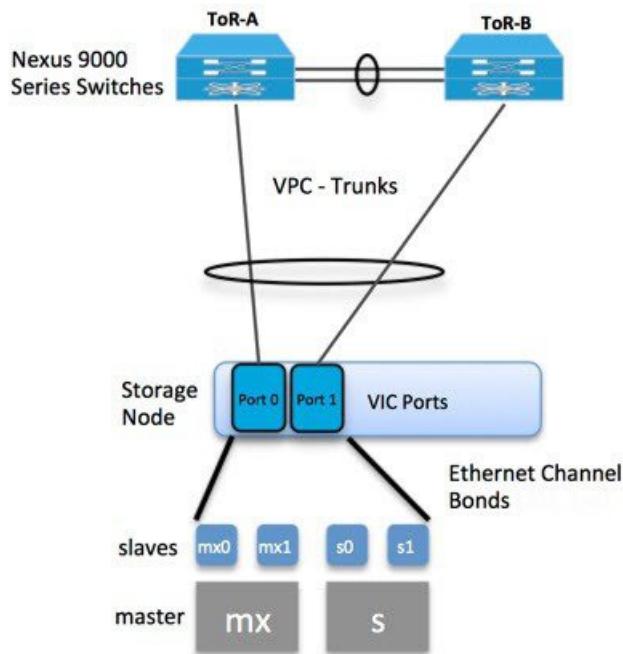
The following figure shows the controller node network-to-bond-to-vNIC interface mapping.

Figure 10: Controller Node Network to Bond Mapping

The Cisco NFVI compute node has three bonds: mx, t, and p. Each has a slave interface that is named with the network name association and a mapped number. For example, the provider network, p, maps to p0 and p1. The bonds map directly to the vNICs that are automatically created on the compute node when it is deployed. The following figure shows the compute node network-to-bond-to-vNIC interfaces mapping.

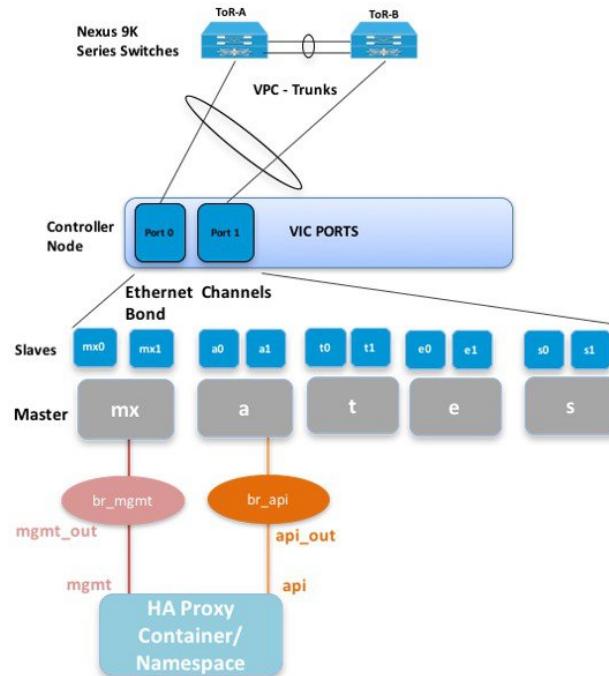
Figure 11: Compute Node Network to Bond Mapping

The Cisco NFVI storage node has two bonds: mx and s. Each has a slave interface that is named with the network name association and a mapped number. For example, the storage network, s, maps to s0 and s1. Storage nodes communicate with other storage nodes over the mx network. The storage network is only used for Ceph backend traffic. The bonds map directly to the vNICs that are automatically created on the storage node when it is deployed. The following figure shows the network-to-bond-to-vNIC interfaces mapping for a Cisco NFVI storage node.

Figure 12: Storage Node Networking to Bond Mapping

Cisco NFVI installation creates two bridges on the controller nodes and interfaces and bonds are attached to the bridges. The br_api bridge connects the API (a) interface to the HAProxy container. The HAProxy and Keepalive container has VIPs running for each OpenStack API endpoint. The br_mgmt bridge connects the Management and Provisioning (mx) interface to the HAProxy container as well.

The following figure shows the connectivity between the mx interface and the br_mgmt bridge. It also shows the connectivity between the br_mgmt and the HAProxy container/namespace using mgmt_out and mgmt interfaces. The figure shows the connectivity between the api interface and the br_api bridge as well as the link between the br_mgmt bridge and the HAProxy container using api_out and mgmt_out interfaces.

Figure 13: Bridge and Network Namespace Layout

A sample routing table is shown below. br_api is the default route and br_mgmt is local to the pod.

```
[root@c43-bot-mgmt ~]# ip route
default via 172.26.233.193 dev br_api proto static metric 425
172.26.233.0/25 dev br_mgmt proto kernel scope link src 172.26.233.104 metric 425
172.26.233.192/26 dev br_api proto kernel scope link src 172.26.233.230 metric 425

[root@c43-bot-mgmt ~]# ip addr show br_api
6: br_api: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 58:ac:78:5c:91:e0 brd ff:ff:ff:ff:ff:ff
    inet 172.26.233.230/26 brd 172.26.233.255 scope global br_api
        valid_lft forever preferred_lft forever
    inet6 fe80::2c1a:f6ff:feb4:656a/64 scope link
        valid_lft forever preferred_lft forever

[root@c43-bot-mgmt ~]# ip addr show br_mgmt
7: br_mgmt: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 58:ac:78:5c:e4:95 brd ff:ff:ff:ff:ff:ff
    inet 172.26.233.104/25 brd 172.26.233.127 scope global br_mgmt
        valid_lft forever preferred_lft forever
    inet6 fe80::403:14ff:fef4:10c5/64 scope link
        valid_lft forever preferred_lft forever
```

Cisco VIM Management Node Networking

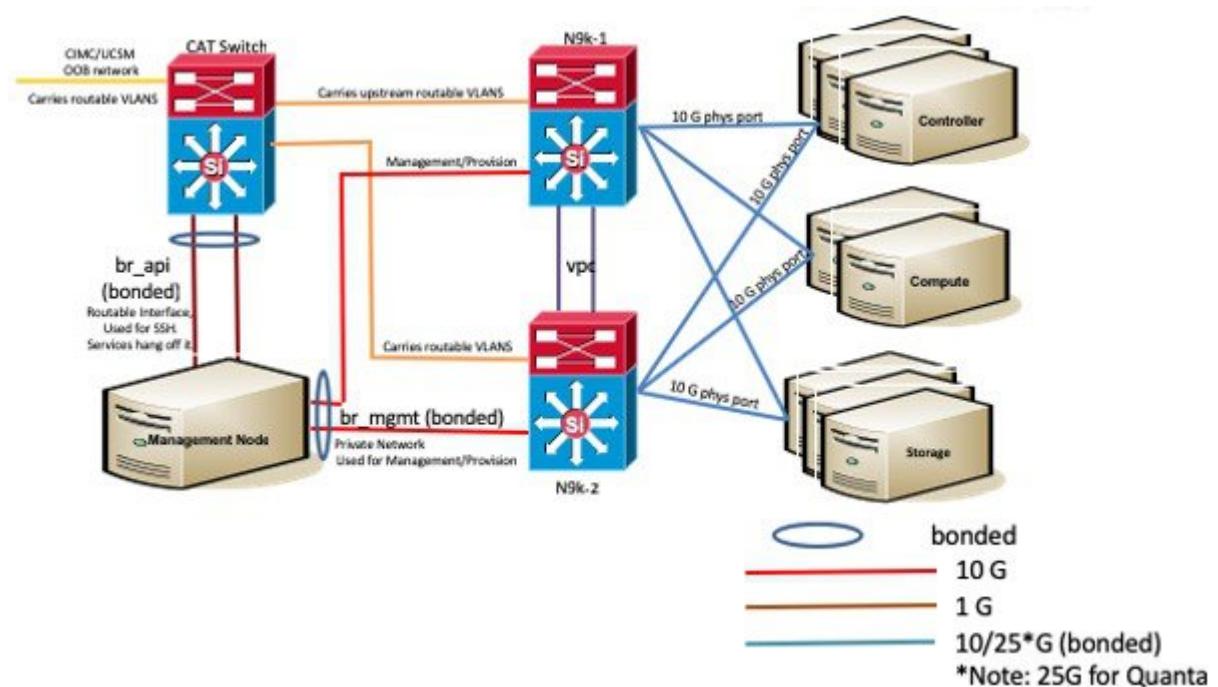
In Cisco VIM, the management node has an interface for API and another interface for provisioning. This is primarily done for security reasons so that internal pod management or control plane messages (RabbitMQ, Maria DB, and so on) do not leak out, and hence reduce the attack vector to the pod. As the name indicates, the API interface is to access the VIM installer API and is also used to SSH to the management node. All

external services (installer API, Insight, ELK, and so on) are password that is protected and hang off the API interface. Default route of the management node points to the API interface.

The second interface, also called the provisioning interface is used to PXE boot the various nodes that constitute the OpenStack pod. Typically, provisioning interface is a non-routable interface that is reserved for OpenStack management traffic.

In B-series pod, the networks between provisioning and the UCSM IP need to be routable. Proper ACL has to be applied in the upstream router so that other networks do not interfere with the provisioning network. Depending on the overall deployment, the management node acts as a jump-server to the OpenStack nodes.

Figure 14: Cisco VIM Management Node Networking



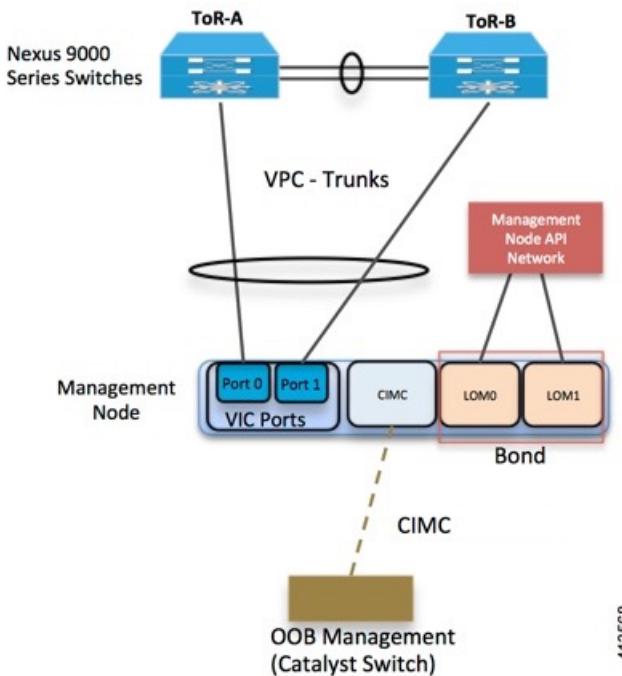
Cisco NFVI UCS C-series management node physically connects to the network. Unlike other nodes, the management node does not use multiple vNICs corresponding to specific Cisco NFVI networks. Instead, it connects to the management and API networks using two different physical connections. The management node connects to the management network using a Cisco two-port VIC or first two ports of intel X710-DA4, with each port connecting to a different ToR switch in a VPC configuration. The Cisco VIC card utilizes the default vNICs, but requires the vNICs to be in trunk mode and the default VLAN set to the management network VLAN. The management node connects to the API network using both one Gbps LAN On Motherboard (LOM) ports connected in a port channel configuration. These ports can either connect to the Nexus 9000 Series switch in a VPC configuration, or to an operator-managed switch(es), depending on how the operator wants to segment their network. The Cisco IMC port can optionally be connected to an out-of-band management Catalyst switch.

Management node services, which are required to start the other topology nodes, listen on the management network and the traffic flowing over the vNICs or NICs on that network. These services and the other management network services are unsecured. Secure management node services listen on the management node API network, and their traffic flows over the LOM ports. This service division allows tenants to utilize tighter network access control to the management network than the management node API network. The following figure shows the Cisco NFVI management node (UCS C-Series) API network connections.



Note Connecting Cisco IMC port to a Cisco OOB management switch is optional.

Figure 15: Management Node API Network Connections



413568

For the day-0 server automation in Cisco VIM, ensure that the reachability to:

CIMC/ILO/BMC of the individual servers from the management node is available through the br_api network.

Cloud API, external network (for ssh to floating IPs) and provider network from the management node is available, as the VMTP and NFVbench are typically run from the management node.



Note From the Cisco VIM release 2.4.3 onwards, you can enable or disable the default behavior of the management node reachability from cloud API, external network, and provider network as part of their day-0 configuration.

If you disable the reachability to cloud api, external, and provider network for security reasons, then:

- VMTP and NFVbench are not accessible from the management node.
- Cloud api, external network and provider network must be properly routed as the Cisco VIM cannot automatically validate the same.

IPv6 Support on Management Network

You can switch from IPv4 to IPv6 as the number of available routable IPv4 networks is limited. In Cisco VIM, the management network uses the default IPv4 route to reach external service like NTP, DNS, AD/LDAP, SwiftStack, and so on, if it is not locally hosted.

Due to the limited availability of IPv4 address space, if you cannot provide a routable IPv4 network or local or dual-home of the external services that require routing, for example, AD or LDAP, deployment hindrance can occur.

IPv4 is obligatory in Cisco VIM, as the provision network collocates with the management network (mx/samx interface) for baremetal PXE install and Ansible orchestration.

As CEPH and OpenStack control plane communication are on the same management network, you cannot completely remove IPv4 from the management network. However, you can run IPv4+IPv6 dual stack in which IPv4 network can exist in a non-routable private network and IPv6 network can exist in a routable semi private network. This ensures to satisfy the requirements of the CiscoVIM and accessibility to the external services.

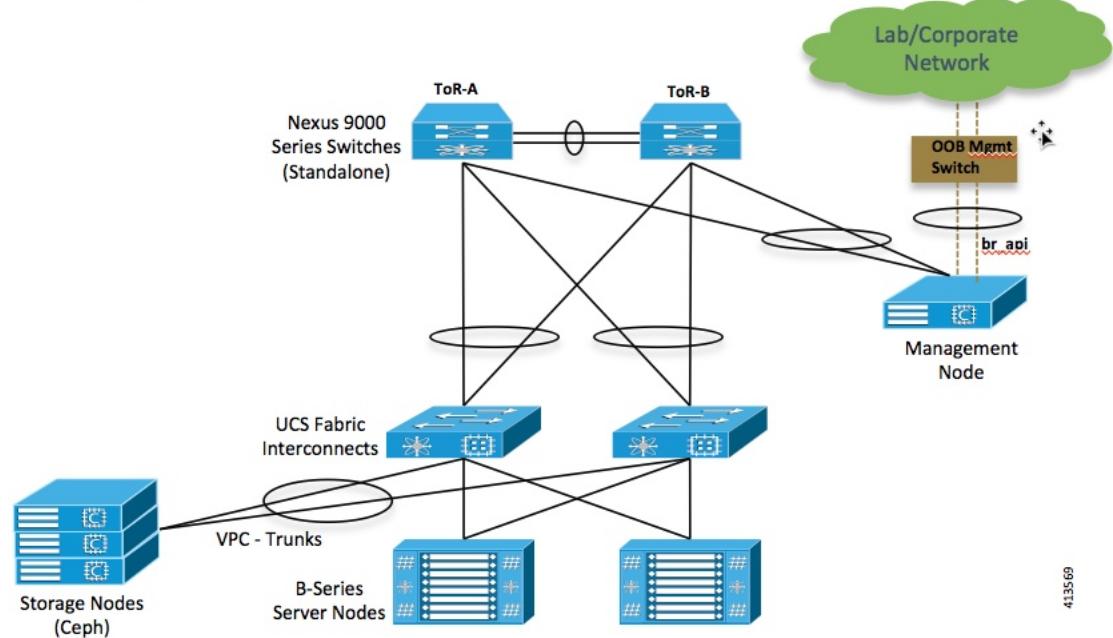
In Cisco VIM, the management network supports IPv6 addresses for servers, while the management node is statically allocated from a given pool. The external services that support both IPv4 and IPv6 addresses are DNS, NTP, and AD or LDAP. You can run IPv4+IPv6 (optionally) as the cloud API endpoint. CIMC/BMC can have IPv6 addresses.

UCS C-Series and B-Series -Topologies

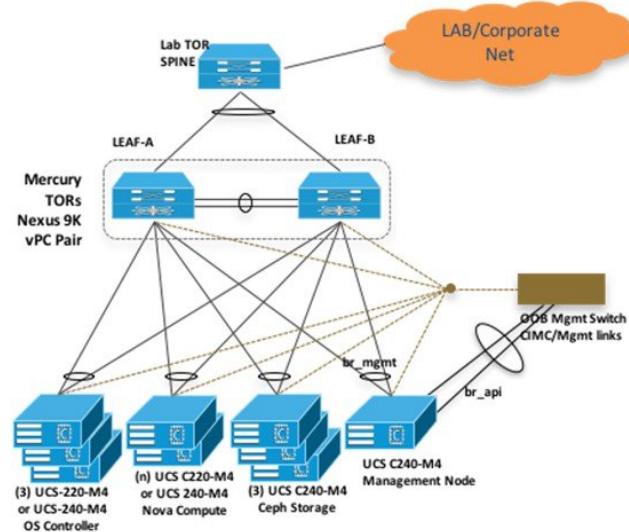
You can deploy Cisco NFVI using a combination of Cisco C-Series and B-Series servers. The C-Series management node is connected to the Cisco Nexus 9000 Series ToRs through the Cisco VIC in a VPC configuration. The UCS Fabric Interconnects (FIs) are connected to the ToRs and the UCS B-Series blade chassis is connected to the FIs. The C-Series storage nodes are connected to the ToRs as well. For C-series implementation, see *Cisco NFVI Networking Overview*. For the combination of the C-Series and B-Series implementation, two exceptions are listed below:

- For UCS B-Series, the Cisco UCS Manager IP address must be available to the Cisco NFVI management network. For UCS C-Series, this requirement is optional.
- The UCS Manager cluster and VIP connections are not attached to one of the Cisco NFVI network segments.

Following figure shows a high-level view of Cisco UCS C-Series and B-Series servers that are used in a Cisco NFVI deployment.

Figure 16: UCS B-Series Topology

For C-Series pods, each host has a 2x10-GE Cisco network card 1227 from which the installer creates two vNICs for each network to ensure that the network topology has built-in redundancy. The provider network, if needed, is also created from the same network card. Each link of a given network type terminates to a unique Cisco Nexus 9000 switch, which acts as the ToR. The Cisco Nexus 9000s are configured in VPC mode to ensure that the network redundancy. The networking redundancy is extended to the management node, which has a redundant vNIC for the installer API and management or provisioning networks. The following figure shows the C-Series topology.

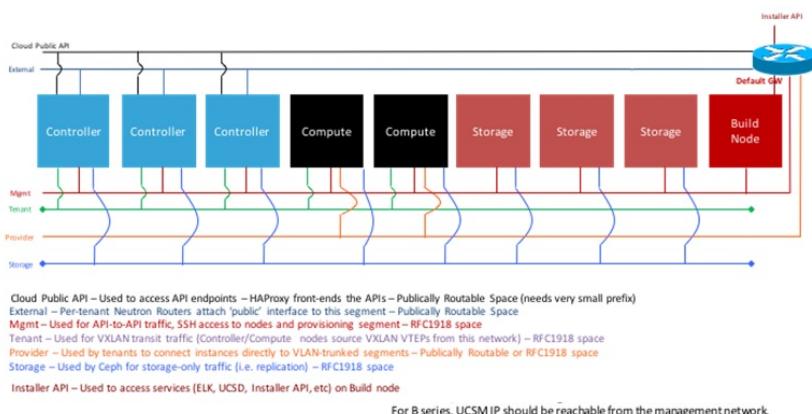
Figure 17: Cisco NFVI C-Series Topology



Note While the figure depicts UCS 220 M4s as the controller and compute, it also supports UCS 240 M4s as control and compute nodes.

Cisco NFVI uses multiple networks and VLANs to isolate network segments. For UCS C-series management and storage nodes, VLANs are trunked between the ToR switches and the Cisco VICs on the C-Series nodes. For UCS B-series controllers and compute nodes, VLANs are trunked between the ToR switches, the UCS Fabric interconnects, and the B-Series blades. The figure shows the network segments and how each node is attached to them. The network segments are VLANs that are trunked between the respective upstream switch/FI and the C-Series or B-Series node.

Figure 18: Network and VLAN Layout for Combined C-Series and B-Series Installation



Cisco NFVI High Availability

Cisco NFVI high availability (HA) is provided by HAProxy, a single-threaded, event-driven, non-blocking engine combining a fast I/O layer with a priority-based scheduler. HAProxy architecture is layered with bypass mechanisms at each level to ensure that the data does not reach higher levels than needed. Most processing is performed in the kernel.

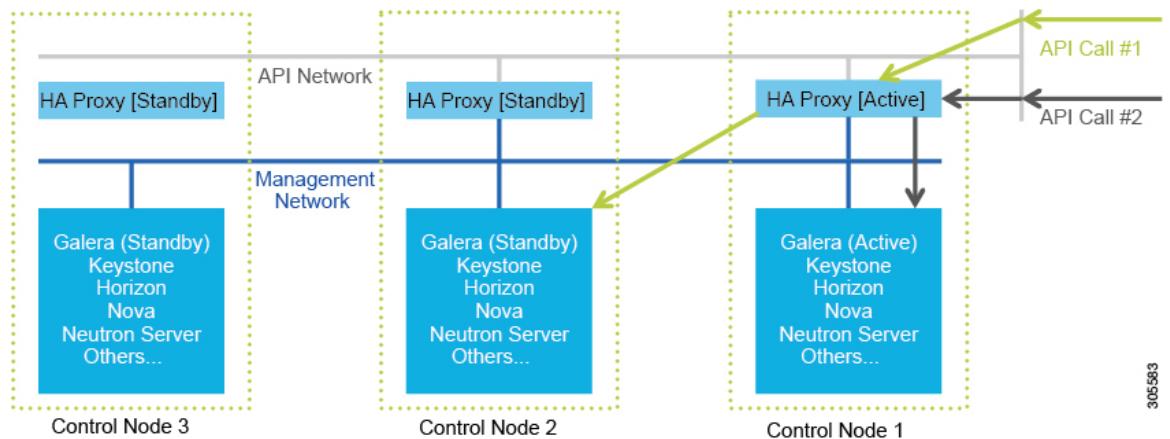
The following figure shows a detailed view of Cisco NFVI controllers connecting to the API and Management and Provisioning network. It also shows how the bridges are configured and the roles of the HAProxy container and network namespace. The dedicated HAProxy container network namespace was created to avoid split default gateway problems. The namespace allows API segment ingress and egress traffic to have a different default gateway than the one configured on each controller host for non-API traffic. In the illustration, two of the three Cisco NFVI controllers have HAProxy containers and a dedicated Linux network namespace. (Cisco NFVI supports three HAProxy containers).

In the figure, Control Node 1 is attached to the API network segment through the br_api bridge. The br_api bridge connects to the Linux network namespace where the HAProxy container has an interface that is mapped through the api <> api_out interface mapping. The HAProxy container has a default gateway configured that points to the upstream API Layer 3 First Hop Redundancy Protocol (FHRP) VIP. This gateway is used for the HAProxy container incoming and outgoing API traffic.

Outside traffic coming in through the API interface is routed into the API network. The traffic traverses the br_api bridge, goes into the Linux network namespace and then the API VIP (based on the IP address or port)

that is listening on the HAProxy container. The HAProxy container establishes a connection with the backend API endpoint (for example, the OpenStack Horizon dashboard) and the return traffic passes through the container and back out the API network following the default gateway for the container on the API network. All other non-API traffic such as the management access over SSH to the Cisco VIM controller comes into the management or provisioning network and access the node directly. Return traffic uses the host-level default gateway that is configured on the Linux (RHEL) operating system.

Figure 19: HAProxy Control Node Flow



If an HA event occurs in a Cisco NFVI pod, Cisco VIM automatically shuts down machines by failing over services. Examples include:

- For API servers, HAProxy automatically ensures that the other redundant control services handle requests, avoiding the shutdown/terminated/non-responding one.
- For quorum services, such as Galera, the remaining members of the quorum continue to provide service and HAProxy ensures that new requests go to the remaining processes.
- For an active/standby process such as HAProxy, the system moves the endpoint IP to a standby copy and continues to operate.

All these behaviors are automatic and do not require manual intervention. When the server is restarted, the services automatically come into service and are added to the load balancing pool, joining their quorums or are added as backup services, depending on the service type.

While manual intervention is not needed, some specific failure scenarios (for example, Mariadb, rabbit) can cause problems that require manual intervention. For example, if a complete network failure occurs, the Galera and RabbitMQ clusters can go into three-way partition. While the Cisco NFVI cluster is resilient to single-point failures, two switches failing simultaneously—something highly unlikely in long-running systems—can sometimes happen due to administrative error, in which case, manual intervention is needed. To repair the pod, the management node must be up and running and all the nodes accessible through password-less SSH from the management node. From the installer<tagid> dir, execute:

```
# ciscovim cluster-recovery
```

Control nodes recovers after the network partitions are resolved. After executing this command, control nodes services come back to working state. To make sure that the Nova services are good across the compute nodes, execute the following command after sourcing /root/openstack-configs/openrc:

```
# nova service-list
```

To check for the overall cloud status, execute the following:

```
# ciscovim cloud-sanity create test all
```

To view the results of cloud-sanity, use the following command:

```
#ciscovim cloud-sanity show result all -id <uid of the test >
```

Cisco NFVI Storage Node Overview

Block Storage

Cisco NFVI storage nodes utilize Ceph, an open source software for creating redundant, scalable data storage using clusters of standardized servers to store petabytes of accessible data. OpenStack Object Storage is a long-term storage system for large amounts of static data that can be retrieved, leveraged, and updated. It uses a distributed architecture with no central point of control, providing greater scalability, redundancy, and permanence. Objects are written to multiple hardware devices, with the OpenStack software responsible for ensuring data replication and integrity across the cluster. Storage clusters scale horizontally by adding new nodes. If a node fails, OpenStack replicates its content across other active storage nodes. Because Ceph uses software logic to ensure data replication and distribution across different devices, inexpensive commodity hard drives and servers can be used in lieu of more expensive equipment.

Cisco NFVI storage nodes include object storage devices (OSDs), hard disk drives (HDDs), and solid state drives (SSDs). OSDs organize data into containers called objects that a user or application determines are related. The objects reside in a flat address space where they all exist at the same level and cannot be placed inside one another. Each OSD has a unique object identifier (OID) that allows the Cisco NFVI control node to retrieve it without knowing the physical location of the data it contains.

HDDs store and retrieve digital information using one or more rigid rapidly rotating disks coated with magnetic material. The disks are paired with magnetic heads arranged on a moving actuator arm, which read and write data to the disk surfaces. Data is accessed in a random-access manner; individual data blocks can be stored or retrieved in any order and not only sequentially. HDDs are a type of non-volatile memory, retaining stored data even when powered off.

SSDs are solid-state storage devices that use integrated circuit assemblies as memory to store data persistently. SSDs primarily use electronic interfaces compatible with traditional block input/output (I/O) hard disk drives, which permit simple replacements in common applications.

Cisco NFVI storage nodes are managed by the control node applications including Ceph monitoring dashboard, Glance, and Cinder. The Ceph monitoring dashboard provides a view into the overall storage node health. Glance virtualizes pools of block storage devices and provides a self-storage API to request and consume those resources. Cinder is an OpenStack block storage service designed to present storage resources to the OpenStack compute node.

In Cisco VIM, depending on the needs of the user, the number of OSDs a pod can have is between 3 and 20. From release Cisco VIM 3.0.0 onwards, you can choose to have multi-backend Ceph in the same pod, to support different I/O requirements. Currently, this is a day-0 decision. You must decide whether to start with single or multi back-end ceph, with a minimum of three nodes for each backend type. Only 2 backends (one of type HDD and another of type SSD) for each pod is supported. For details on how to use HDD or SSD based ceph, see *Cisco Virtualized Infrastructure Administrator Guide*.

Cisco VIM supports NetApp devices running ONTAP 9.X or higher. NetApp devices are added as an alternate to Ceph for block storage. Cisco VIM has been integrated and tested with FAS2650 SKU of NetApp, however it does not preclude Cisco VIM from working with SKUs of NetApp that are compatible FAS2650. Now, you have to choose the blockstorage and the hardware from Day 0.

Object Storage

Cisco VIM provides an integration with SwiftStack, an object storage solution. In this case, the SwiftStack is installed and managed outside the Cisco VIM ahead of time, and the VIM orchestrator adds the relevant Keystone configuration to access the SwiftStack endpoint. In addition to Keystone integration, the Cinder service is also configured to support backup of the volumes to SwiftStack object store. In the current integration, the SwiftStack endpoint has to be in a network routable to/from the CiscoVIM API network (as the VIM API is the same as the Keystone public endpoint network). In the current release, because of limitations in SwiftStack, Cisco VIM is integrated only with KeystoneV2.

In Cisco VIM, you can choose to use Solidfire as an option for block storage along with Ceph. In this scenario, the backend for Glance is Ceph, and the customers have a choice for the Cinder backend to be Ceph or Solidfire. The Cinder block storage service manages the creation, attachment, and detachment of these volumes between a storage system, such as, SolidFire, and different host servers. Also, in Cisco VIM, the data in Solidfire will be backed by Ceph. The Solidfire cluster is pre-deployed and has 2 networks: management and storage. It is recommended that:

- The storage network for Cisco VIM is same as that for Solidfire.
- The management network for Solidfire is reachable from Cisco VIM control nodes.

Overview to Cisco Virtual Topology System

The Cisco Virtual Topology System (VTS) is a standards-based, open, overlay management and provisioning system for data center networks. It automates the data center overlay fabric provisioning for both physical and virtual workloads.

Cisco VTS provides a network virtualization architecture and software-defined networking (SDN) framework that meets multitenant data center cloud service requirements. It enables a policy-based approach for overlay provisioning.

Cisco VTS automates network overlay provisioning and management tasks, integrates with OpenStack and simplifies the management of heterogeneous network environments. Cisco VTS provides an embedded Cisco VTS GUI and a set of northbound Representational State Transfer (REST) APIs that is consumed by orchestration and cloud management systems.

Cisco VTS architecture has two main components: the Policy Plane and the Control Plane. These perform core functions such as SDN control, resource allocation, and core management function.

- Policy Plane—Enables Cisco VTS to implement a declarative policy model that captures user intent and converts it into specific device-level constructs. Cisco VTS includes a set of modular policy constructs that can be organized into user-defined services for use cases across service provider and cloud environments. The policy constructs are exposed through REST APIs that is consumed by orchestrators and applications to express user intent, or instantiated through the Cisco VTS GUI. Policy models are exposed as system policies or service policies.
- Control Plane—Serves as the SDN control subsystem that programs the various data planes including the VTFs residing on the x86 servers, hardware leafs, DCI gateways. The control plane hosts the Cisco IOS XRv Software instance that provides route peering capabilities between the DCI gateways or to a BGP route reflector. (Cisco IOS XRv is the virtualized version of Cisco IOS XR Software.) The control plane enables an MP-BGP EVPN-based control plane for VXLAN overlays originating from leafs or software VXLAN tunnel endpoints (VTEPs)

The Cisco NFVI implementation of Cisco VTS includes the VTS Virtual Topology Forwarder (VTF). VTF provides a Layer 2/Layer 3 (L2/L3) software switch that can act as a software VXLAN terminal endpoint

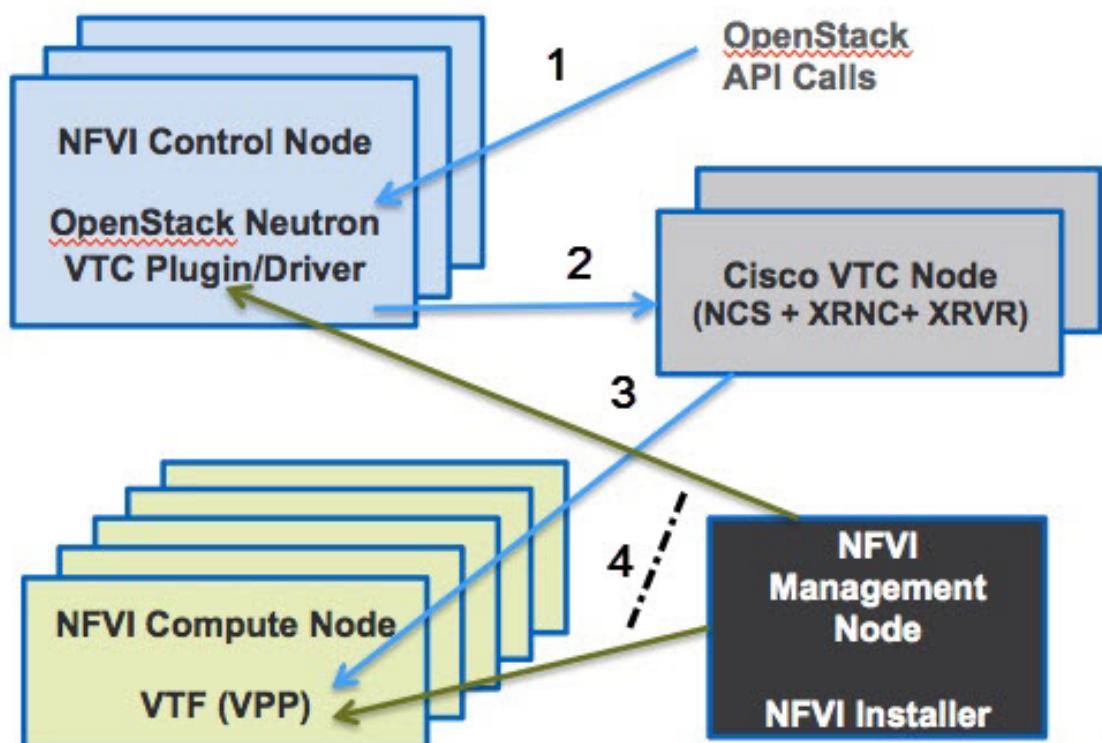
(VTEP). VTF is a lightweight, multitenant software data plane designed for high performance packet processing on x86 servers. VTF uses Vector Packet Processing (VPP). VPP is a full-featured networking stack with a software forwarding engine. VTF leverages VPP and the Intel Data Path Development Kit (DPDK) for high performance L2, L3, and VXLAN packet forwarding.

VTF allows Cisco VTS to terminate VXLAN tunnels on host servers by using the VTF as a software VXLAN Tunnel Endpoint (VTEP). Cisco VTS also supports hybrid overlays by stitching together physical and virtual endpoints into a single VXLAN segment.

The figure below shows the Cisco VTS architecture and high-level flow when installed in Cisco NFVI. Cisco VTS is installed on separate UCS servers, the Virtual Topology Controller plugin is installed on the control node, and the VTF is installed on the compute node.

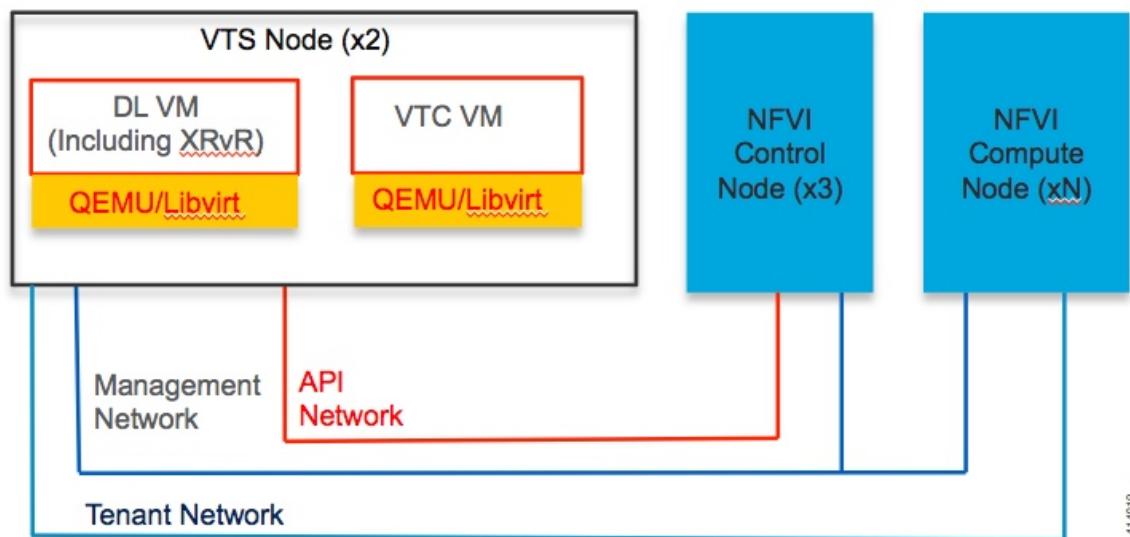
1. The OpenStack user invokes the OpenStack Neutron API.
2. Neutron uses the VTS plugin and driver to make calls to the VTC REST API.
3. VTS control components interact with the VTF agent to carry out the corresponding dataplane setup.
4. During Cisco NFVI installation, the Cisco NFVI Installer installs the OpenStack Neutron VTC plugin and driver on the Cisco NFVI controller node, and installs the VTF component (including VPP) on the Cisco NFVI compute node.

Figure 20: Cisco VTS in Cisco NFVI



414018

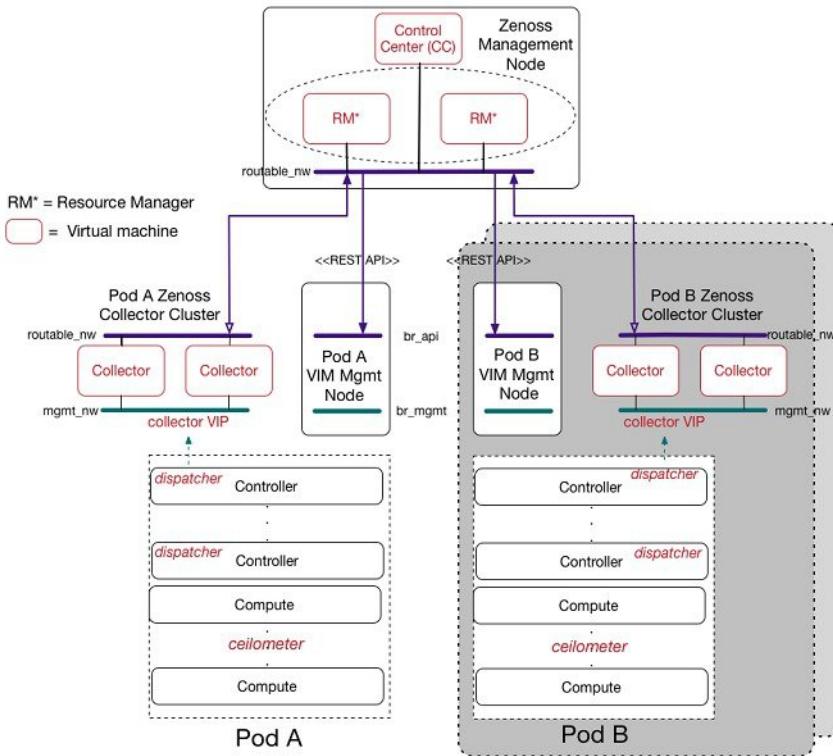
The following illustration shows that the Cisco NFVI networking after the Cisco VTS is installed. The SDN controller nodes are an addition to the existing Cisco NFVI pod.

Figure 21: Cisco VTS Networking Inside Cisco NFVI

414019

Overview to Cisco NFVIMON

Cisco VIM solution uses Cisco NFVI Monitor (NFVIMON) to monitor the health and performance of the NFVI. This includes monitoring both the physical and logical components of one or multiple NFVI pods. NFVIMON feature is enabled by the Zenoss which provides for extensive monitoring and collection of performance data for various components of the cloud infrastructure including Cisco UCS blade and rack servers, service profiles, Nexus top of rack switches, fabric interconnects, and also the OpenStack instances. The monitoring system is designed such that it can monitor single or multiple pods from a single management system. NFVIMON is integrated into Cisco VIM as an optional component. NFVIMON is enabled by extending the `setup_data.yaml` file with relevant information. To enable the NFVIMON, refer to *Enabling NFVIMON on Cisco VIM*. Also, NFVIMON can be enabled on an existing pod, through the reconfigure option. To reconfigure through Insight UI, refer to *Reconfiguring Optional Services*. Then, the pod is added as a new VIM resource to be monitored in the Monitoring UI.

Figure 22: NFVIMON Architecture

The NFVIMON architecture supports monitoring of one or more Cisco VIM pods. There is no limit on the number of pods, but note that the setup supports up to ~ 25000 managed resources across pods, where a managed resource is a physical device, network device or virtual machine tracked from a monitoring perspective.

NFVIMON consists of four components: ceilometer services (for data collection), collector, resource manager (RM), and control-center (CC) with Cisco Zenpacks. As NVIFMON is a third-party software, its integration with the VIM is loosely coupled and the VIM automation only deals with installing the ceilometer service software required to monitor the pod.

The installation of other NFVIMON components (collector, RM, and CC with Cisco NFVI Zenpacks) are Cisco Advance Services led activity and those steps are outside the scope of the current install guide. Make sure that you have engaged with Cisco Advance Services on the planning, image information (of collector with Cisco NFVI Zenpacks and RM), and installation of the NFVIMON accessories along with its network requirements.

Start with one Cisco VIM pod (Pod A in the picture) and two external nodes (one to host 2 Collector VMs and one for remote management to host 1 control-center with Cisco Zenpacks and 2 RM VMs) of multiple pods.

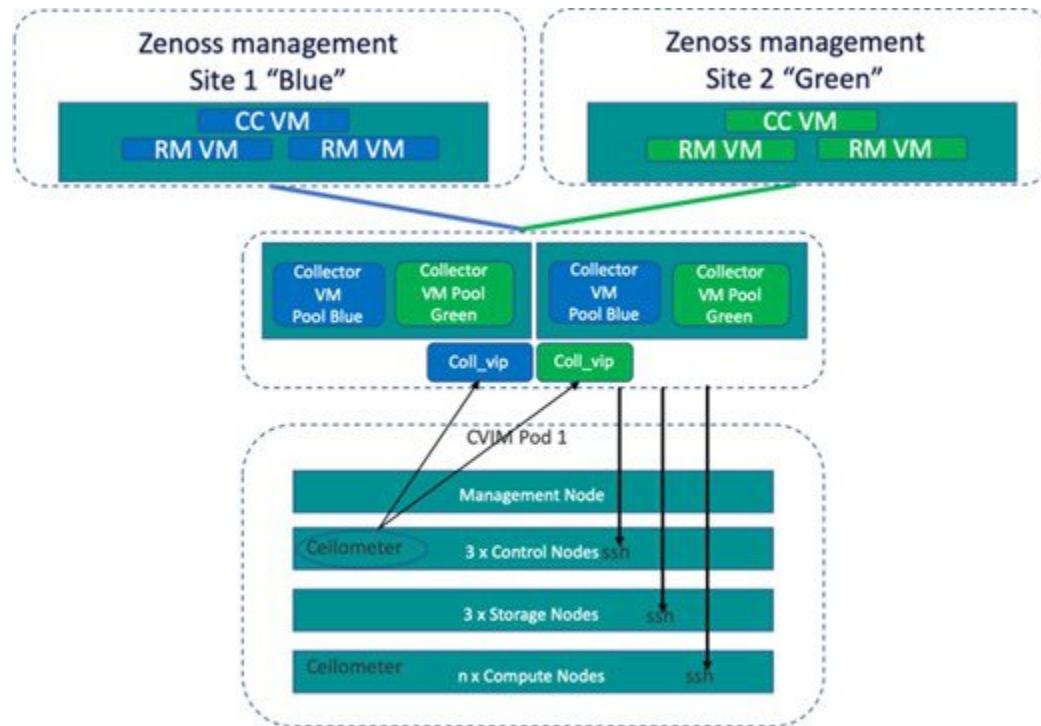
Monitor the Cisco VIM pods at the time of installation if NFVIMON is enabled, or by adding NFVIMON after installation. Install the collectors manually in the external collector node, so that the pod is added for monitoring in the control center.



Note From release Cisco VIM 3.2.0, you can use non-root admin keys for monitoring purposes.

Overview to Cisco NFVIMON High Availability

NFVIMON supports the functionality of high availability (HA). HA is achieved through dual polling of the redundant collectors over an active-active deployment. VM is deployed between the two physical collector servers with two sets of collectors. Two separate Zenoss CC-RMs are connected to one set of collectors each, to aid in simultaneous monitoring of the pod. Ceilometer is deployed in Cisco VIM pod such that it sends data to two separate collector VIPs simultaneously. To enable the NFVIMON, refer to [Enabling NFVIMON on Cisco VIM](#). The NFVIMON HA architecture is depicted in the below figure.



Note Pods running with NFVIMON in standalone mode, cannot be moved to HA mode through reconfiguration.

You can enable NFVIMON HA on day-0 or day-1, when the pod is not running with NFVIMON in the first place.

Overview to CVIM-MON

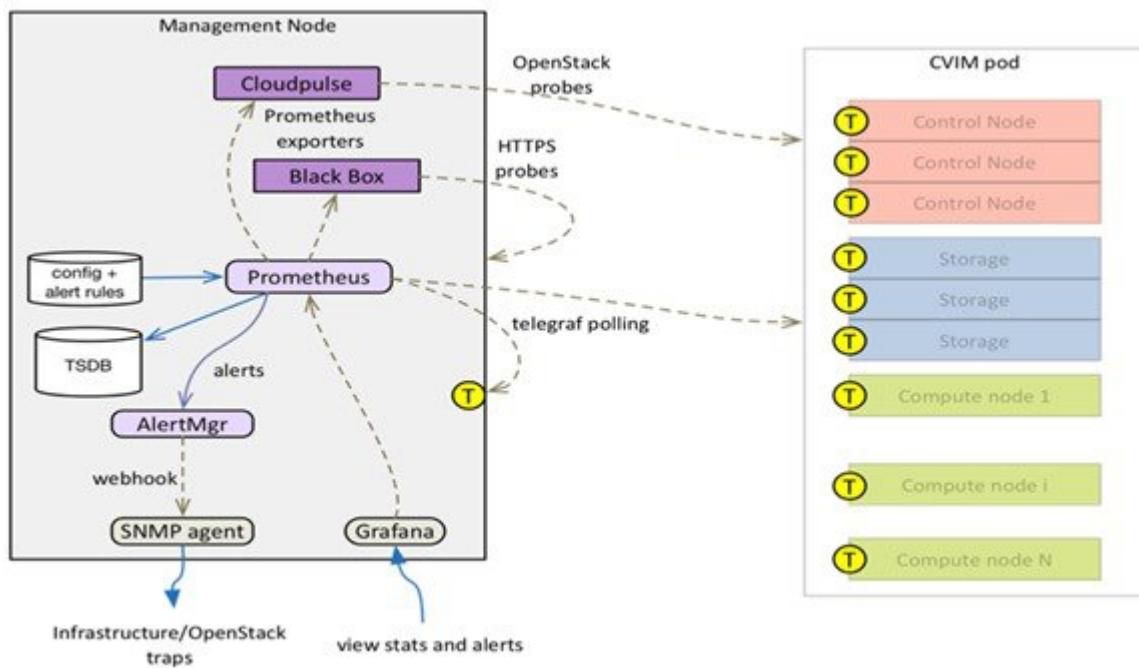
You can deploy Cisco VIM with a lightweight pod-level monitoring solution known as CVIM-MON which is based on the open source PTG stack (Prometheus, Telegraf, Grafana). This solution is available as an add-on from both commercial and feature point of view, and provides the following services:

- Infrastructure-level metric collection based on metric collection agents installed on all nodes in the pod and on specialized collectors running on the management node.
- Metric aggregation into a time series database (TSDB) installed on the management node.

Comparative Analysis

- Rule-based alerting engine integrated in the management node.
- TSDB visualization web server installed on the management node with pre-defined dashboards customized for Cisco VIM.

Figure 23: CVIMMON Architecture



All CVIM-MON components are containerized, except for the Telegraf agents which run on bare metal on all nodes in the pod (including the management node). The two sub-components of CVIM-MON are:

CVIM_MON: Provides the base functionality of monitoring and KPIs.

SNMP: It is enabled only if CVIM_MON is enabled.

Comparative Analysis

The comparison of the two monitoring solutions of Cisco VIM is listed below:

Table 6: Comparison of CVIM-MON and NFVIMON

Features	CVIM-MON	NFVIMON/Zenoss
Open source	Yes	Yes
Collector	Telegraf and Prometheus exporters	Direct ssh to each node
Metrics manager	Prometheus	Zenoss
TSDB	Prometheus	Zenoss
Typical metric frequency	Few seconds or more	Few minutes

Features	CVIM-MON	NFVIMON/Zenoss
Web UI	Grafana	Zenoss
Smart metrics	Yes	No
Alerts	Yes	Yes
SNMP traps	Yes	No
Installation	Integrated with Cisco VIM	External/separate
Hardware requirements	Runs on management node	Requires additional servers

TSDB size and Retention Policy

The size of the TSDB depends on the frequency of the polling (configurable) and the number of compute nodes. By default, the metrics collected in each management node are kept for 15 days.

Smart Metrics

The Cisco VIM deployment blueprint assigns different roles to different hardware or software resources for operational and optimization purposes. CVIM-MON leverages the metric labelling feature in Telegraf and Prometheus, to associate important contextual information with the metrics associated to the resources. This labelling enables monitoring the pod in a precise manner than with traditional unlabelled metrics.

Node Type Label

The nodes in a Cisco CVIM pod can play different roles based on the deployment model. All metrics originating from a node are labelled with the node type (label name = "node_type") and the node name (label name="host").

The following node types are defined:

Table 7: Node Type and its metric source

Node Type	Source of Metric
mgmt	Management node
controller	Controller node
compute	Compute node
storage	Storage node
aio	all-in-one node(micro-pod deployment)
hc	hyper-converged node (hyper-converged deployment)

CPU Role Label

CPUs in a Cisco VIM pod are statically categorized to perform specific functions. This partitioning is critical to guarantee proper level of service for each subsystem independent of the load in the other subsystem. For example, it is imperative to isolate the CPUs reserved for the VPP virtual switch, from any other activity on

the same compute node, to guarantee the virtual switch forwarding performance. The CPU metrics are labeled with a role (label name = "role") to indicate the function of each CPU. This allows to aggregate CPU metrics based on category, which is a lot more useful than aggregating all CPUs.

This categorization cannot be done with unlabeled metrics (by reading CPU time series from a TSDB), due to the following reasons:

- Identification of CPU role based on the core number.
- Existence of multiple types of nodes.
- Each node type has a different CPU partitioning map. The CPU partitioning map may depend on the Cisco VIM release default mapping or customer specific deployment configuration (for example, on a hyper converged node, the number of cores reserved for CEPH can vary from deployment to deployment).

CVIM-MON uses the following roles to label CPU metrics:

Table 8: Role label and static CPU assignment

Role	Static CPU Assignment
host	System and OpenStack tasks
ceph	CEPH OSD tasks (note that ceph-mon is in the host category)
vpp	VPP virtual switch
vm	VM vCPUs
mgmt	Management tasks on the management node

Metrics Collection

Telegraf Metrics

CVIM-MON collects hundreds of different metrics from each node through the Telegraf plugin. The metrics range from low-level kernel to infrastructure services. The interval between metrics collections is configurable between 10 seconds to 5 minutes.

The following table describes the Telegraf plugins installed as part of the CVIM-MON deployment:

Table 9: List of plug-in and their metric name

Plug-in	Metric Name	Notes
ceph	ceph_osdmap_* ceph_pgmap_* ceph_pool_* ceph_usage_total_*	Collects performance metrics from the MON and OSD nodes in a Ceph storage cluster
cpu	cpu_usage_*	Detailed stats for every CPU (with role label)

Plug-in	Metric Name	Notes
conntrack	conntrack_ip_conntrack_*	Collects stats from Netfilter's conntrack-tools
cvim_net_stats	cvim_net_stats_if_*	Detailed metrics for physical and virtual network interfaces in Cisco VIM environment
disk	disk_*	Detailed stats for every disk
diskio	diskio_*	Disk activity
docker	docker_container_* docker_n_containers	Detailed metrics on running docker containers
exec	directory_plugin_bytes	Monitor EFK and Prometheus own storage usage
haproxy	haproxy_*	
http_response	http_response_*	Monitor HTTP services availability
hugepages	hugepages_*	Monitors huge pages usage per NUMA node
internal	internal_*	Collects metrics about the telegraf agent itself
ipmi_sensor	ipmi_sensor_*	Bare metal metrics, including power usage, fan speeds, temperatures, and voltage
kernel	kernel_boot_time kernel_context_switches kernel_interrupts kernel_processes_forkewd kernel_entropy_avail	
libvirt	libvirt_*	Nova and libvirt data and metrics from VMs running on compute or aio nodes
linkstate	linkstate_actor linkstate_sriov linkstate_partner	Monitoring LACP, SRIOV links status
mem	mem_*	Host level memory stats

Plug-in	Metric Name	Notes
net	net_bytes_* net_packets_* net_conntrack_* net_drop_* net_err_* net_icmp_* net_ip_* net_tcp_* net_udp_*	Metrics about network interface and protocol usage (only for interfaces used by CVIM)
ntpq	ntpq_*	NTP query metrics
openstack	cp_hypervisor_up_* cp_openstack_* cp_ceph_health	OpenStack related metrics, comes as a replacement to cloudpulse
processes	processes_*	
rabbitmq	rabbitmq_overview_* rabbitmq_node_* rabbitmq_queue_* rabbitmq_exchange_*	RabbitMQ metrics, currently disabled by default
swap	swap_*	
system	system_*	Checks system load, uptime, and number of users logged in



Note All metrics are part of the high frequency collection group. The collection interval is in seconds or minutes:

Table 10: Frequency group and metrics collection interval

Frequency_group	Default Interval	Min	Max
High	15s	10s	medium_frequency
Medium	deprecated	high_frequency	low_frequency
Low	deprecated	medium_frequency	5m

OpenStack and infrastructure service metrics

Each Cisco VIM pod can monitor the essential OpenStack services. CVIM-MON gathers OpenStack services data through a custom telegraf plugin. The following metrics are available in Prometheus:

Metric	Metric Name	Notes
ceph check	ceph_health	Checks if ceph is healthy
hypervisor checks	cp_hypervisor_up	Check the state of each hypervisor.
openstack service	cp_openstack_service_upcp	Checks the state of an openstack service. Monitors nova, glance, cinder, keystone, and neutron
rabbitmq status	rabbitmq_	Describes the state of each rabbitmq server. RabbitMQ monitoring is disabled by default.

Etcdb monitoring

When the ML2/VPP Neutron plug-in is deployed, Telegraf is configured to poll directly the etcd cluster to retrieve etcd metrics every 15 seconds.

Alerting Rules

CVIM-MON provides a list of predefined alerting rules that trigger the alerts based on the value of time series metrics polled by Prometheus. To avoid flapping caused by transient conditions, the rules are set to have a grace period and an alert is defined to be in one of the two states:

- Pending — Rule is triggered but the grace period has not expired.
- Fired — Rule is triggered for a period longer than the grace period.

The alerts can be monitored using the web user interface or API and can optionally be converted into SNMP traps. You can configure CVIM-MON to send alerts as SNMP traps to any registered SNMP managers. The maximum number of SNMP managers supported is three, and a combination of SNMPv2 or v3 managers in different servers is supported.

Table 11:

Alert Name	Fault Code	Severity	Description
instance_down	serviceFailure	critical	The node is not reachable or is down, when Prometheus server tries to scrape a target to retrieve its metrics. An instance down means that metrics from that target cannot be retrieved.

Alerting Rules

Alert Name	Fault Code	Severity	Description
disk_used_percent	resourceThreshold	major	The storage device is used at over 90% capacity.
disk_filling_up_in_4h	resourceUsage	critical	The storage device is likely to run out of space in less than 4h
docker_container_down	serviceFailure	critical	The docker container running a Cisco VIM infrastructure service is down. This should never happen and indicates that an infrastructure container is failed or could not start.
link_down_lacp	hardwareFailure	warning	The LACP bonded link is in an error state, if one of the two bonded links is no longer operating properly. For example, the error could be caused by the defective cable connection with the NIC, ToR, or a ToR port misconfiguration. The connectivity may still allow traffic to pass but at half the usual throughput. The defective link must be repaired quickly, to reinstate full bandwidth.
link_down_sriov	hardwareFailure	warning	The SRIOV link is in down state. Usually indicates an issue with the physical cable wiring or a misconfiguration of the corresponding port on the ToR.

Alert Name	Fault Code	Severity	Description
mem_available_percent	resourceThreshold	informational	There is less than 10% of available system memory. Regular 4K pages memory is used by both the system and openstack infrastructure services, and does not include huge pages. This alert can indicate either insufficient amount of RAM or an abnormal memory usage by the system or infrastructure.
memory_running_out_in_4h	resourceUsage	critical	This node is likely to run out of system memory in less than 4h. Based on the historical memory usage, this alert predicts that all the system memory will be used up in less than 4h. This condition should never happen and requires immediate troubleshooting by TAC before the system memory runs out.
swap_used_percent	resourceThreshold	warning	The node is using over 80% of the available swap space. Nodes should normally use only very little swap space. More than that the nodes will not use any swapping at all.
conntrack_percent	resourceThreshold	warning	The node is using more than 80% of the available conntrack objects. This is mostly useful for OVS deployments. This indicates an abnormal use of host kernel conntrack resources.

Alerting Rules

Alert Name	Fault Code	Severity	Description
reboot	hardwareFailure	warning	The node is rebooted in less than 10 minutes. Node reboots should be infrequent and be triggered only by the administrator when the node can safely be rebooted. Spontaneous and spurious node reboots should never happen.
system_n_users	resourceThreshold	warning	The node has more than 10 logged-in users.
ceph_error	serviceFailure	critical	The CEPH cluster is in error state and needs to be repaired immediately.
ceph_warning	serviceFailure	warning	The CEPH cluster is in warning state. Requires attention for the repair to be done.
ceph_osdmap_num_in_osds	resourceThreshold	critical	The CEPH cluster has at least 1 OSD in the OUT state.
ceph_osdmap_num_up_osds	resourceThreshold	critical	The CEPH cluster has at least 1 OSD in the DOWN state.
ceph_pgmap_state_count	resourceUsage	critical	The CEPH cluster has at least 1 placement group that is not in active+clean state
ceph_pgmap_bytes_avail_fling_up_in_4h	resourceUsage	critical	CEPH may run out of space within 4 hours.
ceph_pgmap_bytes_used_percent	resourceThreshold	warning	CEPH used capacity is over 70%.
ceph_pgmap_bytes_used_percent	resourceThreshold	critical	CEPH used capacity is over 80%.
haproxy_plugin_data_absent	other	informational	Not receiving any metrics from HAProxy for 10 minutes or more (should never happen).

Alert Name	Fault Code	Severity	Description
haproxy_active_servers_down	serviceFailure	critical	Indicates that one or more HAProxy active server is not in the UP state.
haproxy_active_servers_backend	serviceFailure	critical	The number of haproxy active server backends is not three.
haproxy_active_servers_galera	serviceFailure	critical	The number of haproxy active galera servers is not one.
haproxy_backup_servers_galera	serviceFailure	critical	The number of haproxy backup galera servers is not two.
http_service_unavailable	serviceFailure	warning	The infrastructure HTTP service at given URL is not responding or is not reachable. This should never happen and may indicate an issue with the availability of the infrastructure service.
rabbitmq_node_running	serviceFailure	critical	At least one of the three rabbitMQ nodes is not running.
rabbitmq_node_mem_used_percent	resourceThreshold	critical	Memory used by rabbitMQ is at 90% of its maximum configured limit.
rabbitmq_queue_consumers	resourceThreshold	critical	One or more rabbitMQ queues have no consumer.
rabbitmq_queue_messages	resourceUsage	critical	The number of queued/unread ready and unacked messages is over 300.
ntp_offset	resourceThreshold	warning	The mean offset (phase) in the times reported between the local host and remote peer or server is over 2500 milliseconds.
cp_openstack_service_down	serviceFailure	critical	The indicated openstack service is not reachable and likely to be down.

Alert Name	Fault Code	Severity	Description
cp_hypervisor_down	serviceFailure	critical	The Nova hypervisor is down.
certificate_expiring_5d	other	critical	The certificate is expiring in less than 5 days and must be replaced.
certificate_expiring_10d	other	warning	The certificate is expiring in less than 10 days.
certificate_expiring_45d	other	informational	The certificate is expiring in less than 45 days .

CVIM-MON Web User Interface

The CVIM-MON graphical user interface allows the pod administrator to monitor the status of the pod using any web browser. This interface is based on Grafana and comes with a set of predefined dashboards.

Access Login

The CVIM-MON web user interface is available by pointing a web browser to the management node IPv4 or IPv6 address (br_api) at port 3000 using https. To access this interface, enter **admin** as username and password.. The password is auto-generated at the time of deployment and can be retrieved from the Cisco VIM password repository (`openstack-configs/secrets.yaml` file) in the `CVIM_MON_PASSWORD` entry.

From release Cisco VIM 3.2.1, an additional read-only user is created. To access the interface, enter **cvim** as the username and `CVIM_MON_READ_ONLY_PASSWORD` (from `openstack-configs/secrets.yaml`) as the password.



Note

- The **Forgot your password?** option in the Grafana login page is disabled.
- New password can be generated for Grafana, by running Cisco VIM reconfiguration with the regenerate secrets option.

Pod <pod-name> Dashboard

The pod dashboard is named as Pod <pod-name> where <pod-name> is configured in `setup_data.yaml` under the option `PODNAME` to provide the following:

- High level view of the pod.
- Total number of nodes grouped by node type.
- Total number of cores grouped by role.
- Total load in the pod or sum of the load for all nodes.
- Average usage of all the CPUs reserved for VMs.
- Hardware information of the pod.

- Dataplane statistics of the pod (Networking metrics like throughputs, errors and packet sizes)

Node Level Metrics Dashboard

This dashboard provides a detailed view of the state of the most important resources for any node in the pod including the management node. A list of drop-down menus allows to select:

- Node to display (only one)
- Disk devices to display (all or any selection)
- Network interfaces to display (all or any selection)
- CPUs to display (all or any selection)

The dashboard provides the utilization charts for the following:

- Alerts
- System
- CPU
- Memory
- Processes
- Disks
- Network interfaces

Pod Level Metrics Dataplane Statistics Dashboard

This dashboard provides a detailed view of the networking metrics and data coming from the libvirt and cvim_net_stats telegraf plugins. The following panels are available as part of the dataplane statistics:

- Top 5 nodes drop rate: Top nodes with physical interfaces TX/RX drops rate out of all TX/RX packets in a 20m timeslot.
- Top 10 VMs drop rate : Top VMs with virtual interfaces TX/RX drops rate out of all TX/RX packets in a 20m timeslot.
- Pod throughput in packet-per-second (pps): Total throughput in pps on all physical interfaces.
- Top 5 nodes throughput in pps: Top nodes throughput in pps on node physical interfaces.
- Top 10 VMs throughput in pps: Top VMs throughput in pps on VM virtual interfaces.
- Pod throughput in bits-per-second (bps): Total throughput in bps on all physical interfaces.
- Top 5 nodes throughput in bps : Top nodes throughput in bps on node physical interfaces.
- Top 10 VMs throughput in bps: Top VMs throughput in bps on VM virtual interfaces.
- Top 5 Nodes error rate: It is the error rate on physical interfaces TX/RX out of all TX/RX packets in a 20m timeslot.
- Average pod packet size: It is calculated from total per interface bytes divided by total packets on all pod physical interfaces.

Node Dataplane Statistics Dashboard

This dashboard provides per node and per VM view of networking metrics and data coming from the libvirt and cvim_net_stats telegraf plugins. The following panels are available as part of the nde dataplane statistics dashboard:

- Two gauges with aggregated (all TX+RX) throughputs in PPS and bps across physical interfaces on the specific node.
- One gauge with total virtual interfaces (attached to VMs) running on the specific node.
- Specific VM drop rate: The specific VMs virtual interfaces TX/RX drops rate out of all TX/RX packets on that VM in a 20m timeslot.
- Node throughput in packet-per-second (pps): It is the total throughput in pps on all physical interfaces on that specific node.
- Node throughput in bits-per-second (bps): It is the total throughput in bps on all physical interfaces on that specific node.
- Average Node packet size: It is calculated from total per interface bytes divided by total packets on all node's physical interfaces.
- VM throughput in packet-per-second (pps) : It is the total throughput in pps on all physical interfaces on that specific VM and per VM interface.
- VM throughput in bits-per-second (bps) : It is the total throughput in bps on all physical interfaces on that specific VM and per VM interface.
- Average VM packet size: It is calculated from total per interface bytes divided by total packets on all VM's virtual interfaces.
- VM error rate: It is the error rate on virtual interfaces TX/RX out of all TX/RX packets in a 20m timeslot.

Specialized Dashboards

Table 12: List of specialized dashboards

Dashboard Name	Description
OpenStack services	Chart shows the state of all OpenStack services, infrastructure containers and hypervisors.
Alerts	Alerts that are triggered passed the grace period or pending (triggered but still within their grace period).
HAProxy	Chart to monitor the HAProxy service.
CEPH	CEPH storage chart, for example, overall OSD CPU load.
NTP	Chart to monitor NTP on the pod.
RabbitMQ	Chart related to rabbitMQ
Etcdb	Chart related to etcd. Only available for ML2/VPP deployments.

Dashboard Name	Description
Memcached	Chart to monitor Memcached on the pod.
Advanced Metrics	Chart that monitor the management node activity such as: <ul style="list-style-type: none">• Prometheus and Elasticsearch disk usage• Prometheus scraping stats
IPMI	Chart that monitor all the nodes and presents bare metal information: <ul style="list-style-type: none">• Temperature• Voltage• Fan Speed• Power

SNMP for Monitoring

Along with CVIM-MON, you can enable SNMP in Cisco VIM to send SNMP Traps to the remote SNMP managers. The SNMP traps are identified from the following, only when the SERVER-MON is enabled in the setup_data.yaml file.

- Alerts collected on Prometheus
- Faults reported by the CIMC of the Cisco C-series-servers via SERVER-MON option

The SNMP Trap sends a notification, when the fault occurs or gets resolved. The notification types are listed below:

- cvimFaultActiveNotif: Notification sent when the fault gets triggered.
- cvimFaultClearNotif: Notification sent when the fault gets resolved.

The SNMP trap contains the following information:

- cvimPodID: PODNAME configured in setup_data.yaml file
- cvimNodeID: Node that generated the fault, or N/A
- cvimFaultSource: Component name that generated the fault
- cvimFaultSeverity: Severity of the fault following the guidelines:
 - emergency (1): System level fault impacting multiple services.
 - critical (2): Critical fault specific to a service.
 - major (3): Component level fault within a service.
 - alert (4): Warning condition for service. It may eventually impact the service.

- informational (5): Informative message and does not impact any service.
- cvimFaultCode: Code. Guidelines followed for code:
 - other(1) : Type of event not specified in the other labels.
 - resourceUsage(2): Resource usage exhausted event.
 - resourceThreshold(3): Resource threshold reached event.
 - serviceFailure(4): Software failure service event.
 - hardwareFailure(5): Hardware failure event.
 - networkConnectivity(6) :Networking issues.

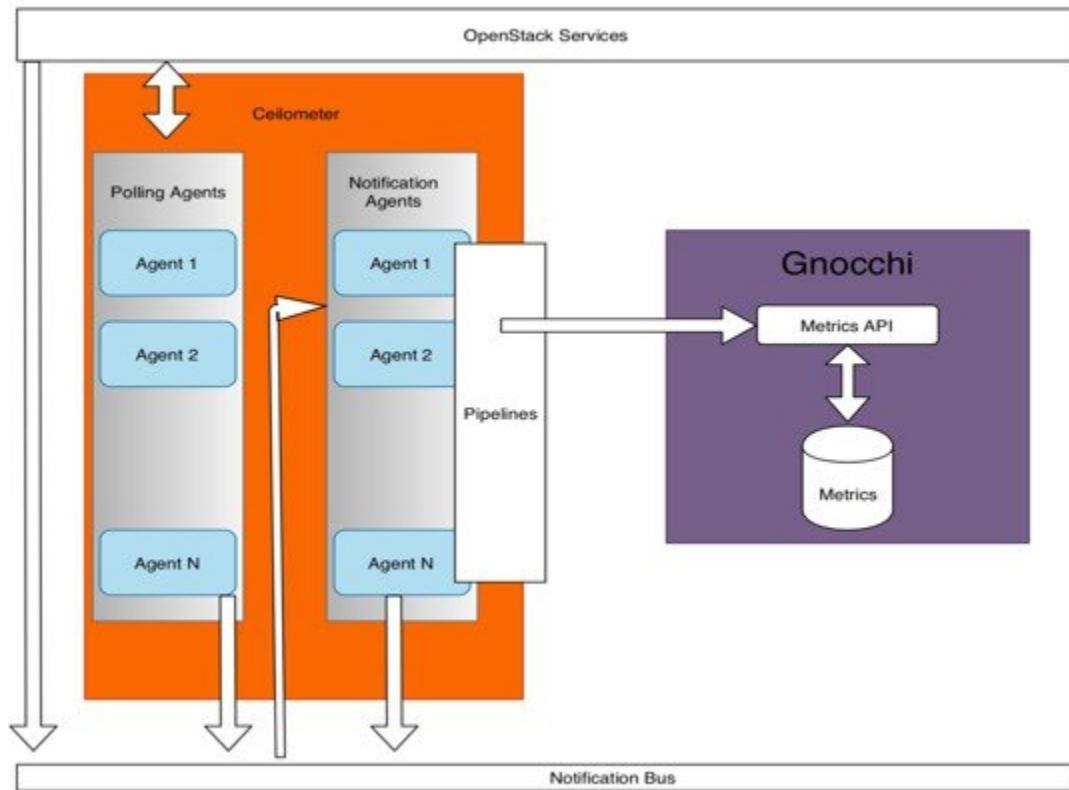
For more details, refer CISCO-VIM-MIB.my.4.0 definition of the MIB at <ftp://ftp.cisco.com/pub/mibs/v2/>.

CVIM-MON is integrated into Cisco VIM as an optional component, and is offered as an add-on with additional license. CVIM-MON is enabled by extending the `setup_data.yaml` file with relevant information. To enable CVIMON, refer to [Enabling CVIM-MON on Cisco VIM, on page 198](#).

You can enable CVIM-MON on an existing pod through the reconfigure option, if the pod is fresh installed with Cisco VIM 2.4.3 or later versions. To reconfigure through Unified Management, refer to [Reconfiguring Optional Services](#). Then, add the pod as a new VIM resource to be monitored so that it is available through the Unified Management portal.

Telemetry Service through OpenStack

Cisco VIM provides telemetry services to collect meters within an OpenStack deployment. Cisco VIM Telemetry service is built on Ceilometer and Gnocchi in OpenStack Queens release. You can retrieve metrics using OpenStack CLI and REST APIs. Pods must have Ceph for persistent storage of the metrics which are collected every five minutes and retained for 48 hours. As Ceph is required for ceilometer, you can install ceilometer as part of fresh installation of the cloud, that is, ceilometer cannot be brought in as a reconfigure option. Also, the ceilometer is supported only on fullon pod. The diagram below illustrates the high-level architecture of the telemetry services.

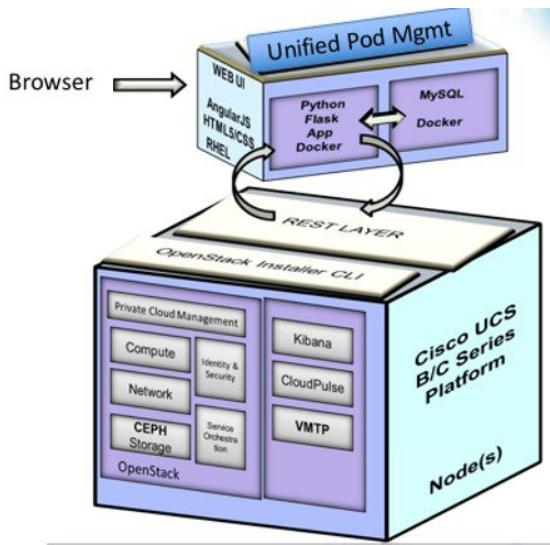
Figure 24: Architecture of Telemetry Services in Cisco VIM

To view the summary of the metrics, see [Telemetry for OpenStack](#) of *Cisco Virtualized Infrastructure Admin Guide, Release 3.0.0*.

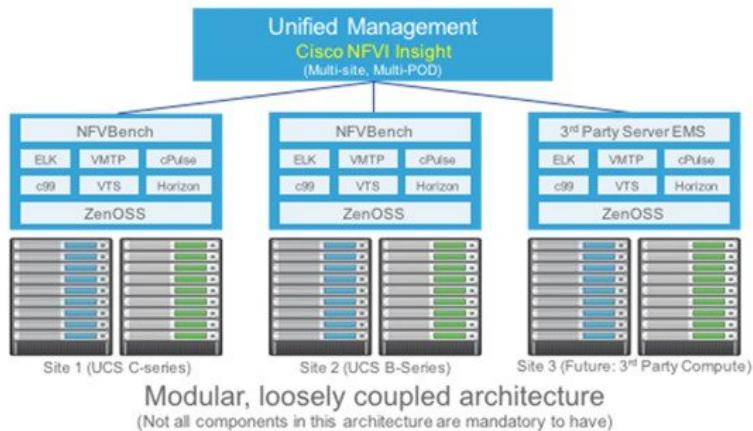
Overview to Cisco VIM Unified Management

Cisco VIM UM, a light-weight UI, is introduced in Cisco VIM to ease the deployment and management of the NFVI platform. This feature is available as an add-on from both commercial and feature point of view. Also, Cisco VIM Insight offers a single pane of glass service to provide deployment visualization and to manage multiple Cisco VIM pods thereby reducing user-errors.

Cisco VIM UM supports multi-tenancy with local RBAC support and is easily integrated with the CiscoVIM REST layer. The container based UI platform is loosely coupled, and can help manage multiple CiscoVIM pods right from day-0, or later in the lifecycle of the cloud.

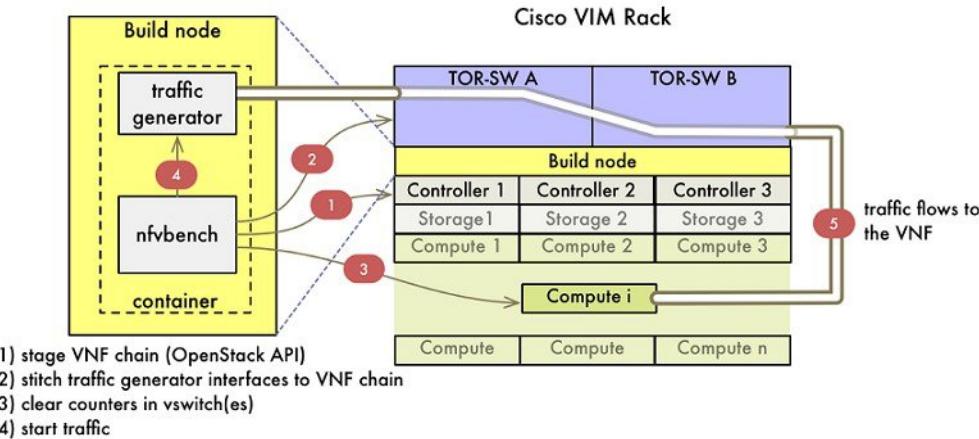
Figure 25: Cisco VIM UM Interaction with a Pod

The architecture of the CiscoVIM UM is light-weight, hierarchical and scalable. While it introduces an ease of management from the global UI, each local site is autonomous with localized toolsets. The Global Unified Management UI, provides ease of management with multi-site multi-pod capability for distributed NFV deployment at scale. Also, CiscoVIM UM is designed to operate in HA as an option. The platform is a modular, loosely coupled architecture, that will provide the capability to manage multiple pods, with RBAC support as shown in the figure .

Figure 26: Cisco VIM UM Architecture

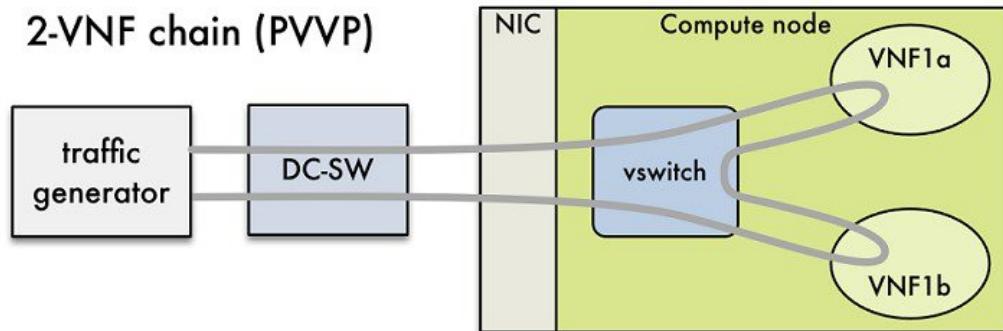
Overview to NFVbench

NFVbench is a containerized network benchmarking tool that is introduced in Cisco VIM, to bring consistent methodology to measure the network performance of the cloud. NFVbench is released in a container that is preinstalled on the management node if the NFVBENCH option is selected in the Cisco VIM configuration file.

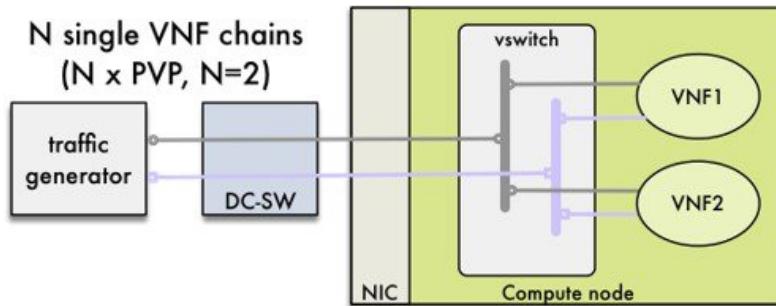
Figure 27: Order of Steps Performed in NFVbench Test

The main goal of NFVbench is to measure the cloud performance that is based on real cloud deployment traffic patterns. During the test, the packet path traverses through every network element that participates in the production environment; that is traffic flows through a switch (ToR) to v-switch on compute node, continues to VM representing any basic VNF in NFV deployment and comes back in similar way on different ports. Network performance or throughput is computed based on sent and received traffic.

NFVbench can bring up one or more chains of test VMs, where each chain consists of one or two VMs. The example below illustrates a case with a two-VM chain.

Figure 28: Packet Path with Two VNFs

Reports from NFVbench show data measurements from every hop in the path, which makes it easier to detect configuration errors or potential bottlenecks. NFVbench sends UDP packets that are generated by open-source traffic generator (TRex) already included in the container. Advanced testing using NFVbench allows to conduct multi-chain and multi-flow testing. Multi-chain testing enables running multiple parallel independent packet paths at the same time, while the multi-flow testing performs IP ranging in packet headers within every chain. The below figure illustrates a NFVbench result test execution with two parallel chains with one VM each.

Figure 29: Multi-chain Example with Two Chains

NDR/PDR and Fixed Rate Tests

NDR/PDR Test: NFVbench offers a more advanced test (called the NDR/PDR test), provides information about network throughput using any of the standard defined packet sizes - 64B, IMIX, 1518B. NDR (No Drop Rate) value represents throughput at which no packets are dropped (satisfied by less than 0.001% of packets being dropped). Similarly, PDR (Partial Drop Rate) represents throughput at which only small number of packets is dropped (less than 0.1% of packets sent).

Fixed Rate Test: NFVbench offers a simple test to run traffic at fixed rate, which verifies that every network component of packet path works properly. It is useful for identifying bottlenecks in the test environment. Traffic generator generates packets at fixed rate for the given time by the user. From the statistics that is collected, drop rates and latencies are computed and displayed.

Both the NDR/PDR Test and Fixed Rate Test types of test provide a way of verifying network performance of NFV solution.

Supported Encapsulation and Protocols

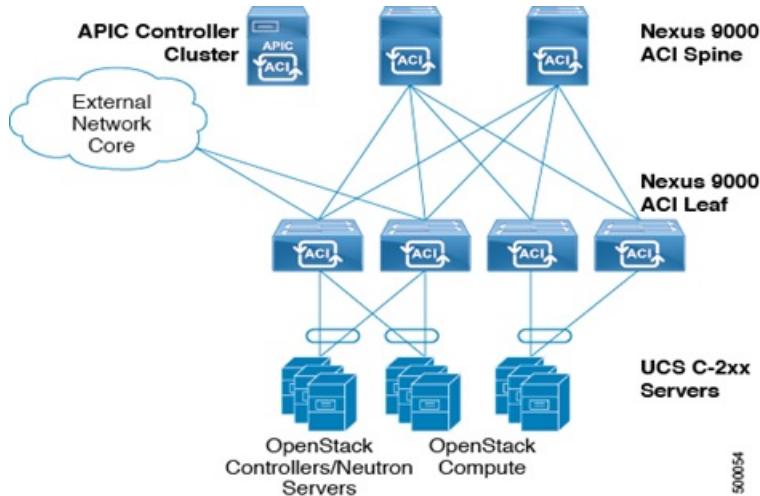
NFVbench supports all networking options that can be deployed with Cisco VIM:

- OVS
- VPP with VLAN or VxLAN
- SR-IOV

Overview to ACI Plugin Integration

The following section gives you an overview of a typical architecture for an ACI fabric with an OpenStack deployment. An ACI with OpenStack deployment consists of a Nexus 9000 Spine/Leaf topology, an APIC cluster, a minimum of 3-node cluster of Controllers (which also acts as the Neutron network node), and two or more compute nodes to host Virtual Machine (VM) instances.

ACI External Routed Network connection is a Layer 3 connection outside the fabric. It is used to provide connectivity outside the OpenStack cloud, as depicted in the following figure.

Figure 30: ACI with OpenStack Physical Topology

Note Basic ACI architecture can be obtained at documentation available in CCO.

The Opflex ML2 plugin integrated with Cisco VIM in Unified mode manages the tenant VLANs dynamically, as and when the VMs operate in the cloud. By utilizing OpFlex, the policy model native to ACI can be extended all the way down into the virtual switches running on OpenStack Nova compute hosts. OpFlex extension to the compute host allows ACI to use Open vSwitch (OVS) to support common OpenStack features such as Source NAT (SNAT) and Floating IP in a distributed manner.

Cisco VIM automates the day-0 ToR level configuration with ACI, except for L3 out. The exception for L3 out is made as you can configure their upstream infrastructure in different ways.

In addition to Cisco VIC (1227), the installer supports pure Intel NIC (x710) deployments with optional SRIOV support. The network model of how an SRIOV network is extended off the fabric is out of the scope of automation. No additional change is required in `setup_data.yaml` to use SRIOV with the ACI plugin, other than defining the two SRIOV ports on a per server basis. The same SRIOV configuration is applicable to enable it with other ToR switches



Note Cisco VIM is validated against APIC 3.2 (4d) with the APIC plugin version of 4.0.1, hence it is imperative to use APIC 3.2(4d) version only.

Auto-ToR Configuration via ACI API

While the use of ACI plugin brings in the flexibility of dynamic allocation of tenant and provider VLANs on demand, it also ties the OVS version to the ACI plugin. This leads to an extreme tight coupling of Cisco VIM and ACI. Also, with an APIC plugin there are might be gaps to cover certain use-cases, for example, where there is a need to have flexibility of different access type (tagged vs non-tagged) for the same VLAN but for different servers.

To address such use-case or avoid tight coupling of OVS with ACI plugin, an optional solution is available to automate the target VLANs on the right switch port based on server role on day-0 along with corresponding fabric access and tenant policy configurations via the ACI API.

With this option, the setup_data for each Cisco VIM instance is the single source for the server to switch port mappings. This solution can handle switch provisioning with the correct VLANs during addition/removal of server and provider/tenant VLAN range expansion via reconfiguration option. This solution is based on the fact that the PV (port*VLAN) count in a given ACI Fabric domain is under the scale limits 10000 PV/ToR and 450000 PV/Fabric.

NCS-5500 as a ToR Option

Cisco VIM supports NCS-5500 as an alternate to a Nexus ToR. NCS-5500 is an IOS XR-based router, which is similar to Nexus switches. You can use the 48 10/25G ports or the 6 40/100G uplink ports model to implement NCS-5500 (port-numbers depend on NCS version). Also, other SKUs of NCS-5500 are supported as long as the NCS-5500 software supports the EVLAG feature. NCS-5500 uses the technology of bridge domain to connect to the server. Enable the Auto ToR configuration feature to support NCS-5500 as ToR. NCS-5500 supports a micropod with more computes running on Intel 710 NICs with the mechanism driver of VPP over LACP. The support is extended to include 40G/100G based NCS-5500 SKUs with splitter cables (of 4x10) connecting to the servers, which helps in increasing the server port density by four folds.

Disk Management in VIM

Cisco VIM uses the disk-maintenance tool that gives you the ability to check the status of all hard disk drives present in the running and operational mode in the following nodes:

- management node
- specific or all controller servers
- specific or all compute servers

Status of the disks such as online, offline, rebuilding helps you to identify which particular disks in which slot has potentially gone bad and require to be physically replaced in the server. It can be run on servers that have either a RAID controller or an SAS passthrough controller.

Once the disk is physically replaced, Disk management tool can be used to add the new disk back into the system as part of the RAID system (recommended one server at a time).



Note

Disk Maintenance tool is useful only when one or at most two (in RAID6) go bad. Failure of more than one disk at a time puts the entire server in an irrecoverable state. Replace the server using remove and add operations through ciscovim. Disk management is not supported on a third party compute due to the licensing issue with the HPE SmartArray Utility tool.

OSD Maintenance

OSD maintenance tool gives you the ability to check the status of all OSDs and their corresponding physical hard disk drives present in the running and operational storage nodes. The status of the OSDs is reported along with the HDD mapping.

OSD Maintenance tool helps you to identify the status of the OSD (Up or Down) and its corresponding hard disk drive slot in the server that requires to be physically replaced. OSD Maintenance tool can run on servers that have either a RAID or an SAS passthrough controller.

Once the HDD to be physically replaced is identified, the same OSD tool can be used to rebalance the ceph tree, remove the OSD from the cluster, and unmount the disk drive, in preparation for the disk removal. After the disk has been physically replaced, the tool can be used to add the new disk back into the system as part of the Ceph cluster and recreate the OSD (only one HDD/OSD at a time). It ensures to replace a bad HDD, it is not required to remove the ceph cluster from operation and then add it back through remove-storage and add-storage options in ciscovim.

**Note**

OSD tool does not support the replacement of the internal OS drives and the external journal drives, for which you still have to use add or remove of OSD nodes.

Power Management of Computes for C-Series

Cisco VIM pods has many compute servers, but the actual usage of the compute servers are limited at times. To optimize the overall power consumption of the data center, we have to power down the server through an API/CLI.

To prevent the cloud destabilization, you cannot power off all the compute nodes. For example, one cannot power off all the compute nodes, at least one pod has to be Active.

Pod management operation(s) applies to the entire pod during updating and reconfigure, the server.

Updating and reconfiguration are not possible under the following circumstances:

- If one or more compute nodes are powered off.
- Computes on which VMs are running cannot be powered-off.
- Computes with All-in-one (AIO) nodes in a micro-pod) cannot be powered-off through this API.

When there is a power-off, internally cloud-sanity is run and if the cloud sanity fails, then the power-off action is aborted.

Physical Cores and Memory Reserved for Cisco VIM Infrastructure

Cisco VIM has been tuned to deliver performance from an infrastructure and VNF point of view. The following are the details of the physical cores (regardless of hyper-thread enabled or not) that the infrastructure needs.

Number of cores that are reserved for the system (host system + OpenStack services) is 2 in all cases and is included in the count that is shown in the following table.

Table 13: Number of Physical Cores and RAM Reserved for Cisco VIM Infrastructure

Pod Type/Node Types	Control	Storage	Compute	AIO	HC
Full On	all	all	CPU: 2+V cores RAM: 25+Vr GB	n/a	n/a
Hyper-Converged (hc)		n/a		n/a	CPU: 2+C+V cores RAM: 41+Vr GB
Micro-Pod (aio)	n/a	n/a		CPU: 2+C+V cores RAM: 41+Vr GB	N/A

Table 14: Number of Physical Cores and RAM Reserved for Cisco VIM Infrastructure

Variables	Usage	Valid range	Default
C	Cores reserved for CEPH (aio and hc)	2..12	2
V	Cores reserved for VPP vswitch	2..4	2
Vr	RAM reserved for VPP		2GB

For OVS deployments, use V=0 and Vr=0

Some VPP deployments with high throughput requirements may require more than 2 VPP cores.

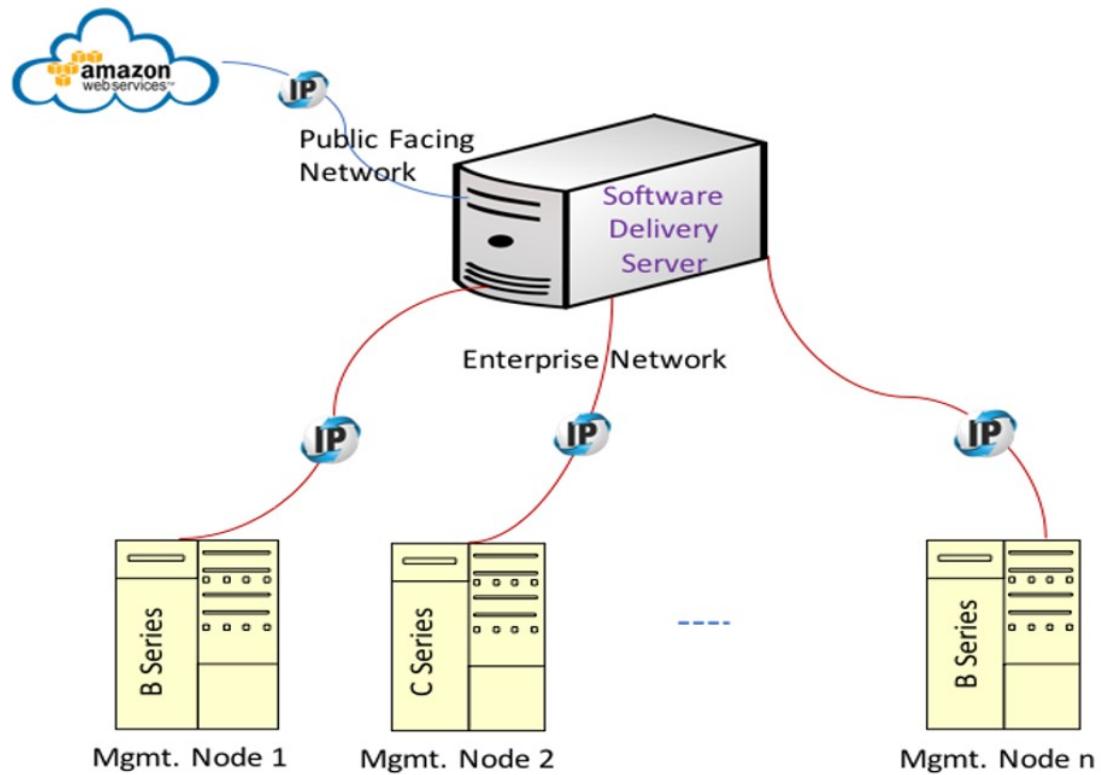
Cisco VIM Software Hub

Cisco VIM is supported in an air-gapped (disconnected mode) environment. You can use a USB or Cisco VIM Software Hub for an air-gapped install. When the number of pods is more, shipping USBs for an air-gapped install and update is not scalable. In such scenarios, we recommend that you use Cisco VIM Software Hub.

Cisco VIM Software Hub contains the Cisco VIM release artifacts such as buildnode ISO, Cisco VIM code, docker registry, and docker images. Using the management node, you can access the release artifacts from the Cisco VIM Software Hub.

You can install the artifacts available on the Cisco VIM Software Hub server through a connected or a disconnected install procedure. For a connected install, one end of the Cisco VIM Software Hub server is connected to the internet, and the other end is connected to the datacenter.

The following figure shows the architecture of a connected install.

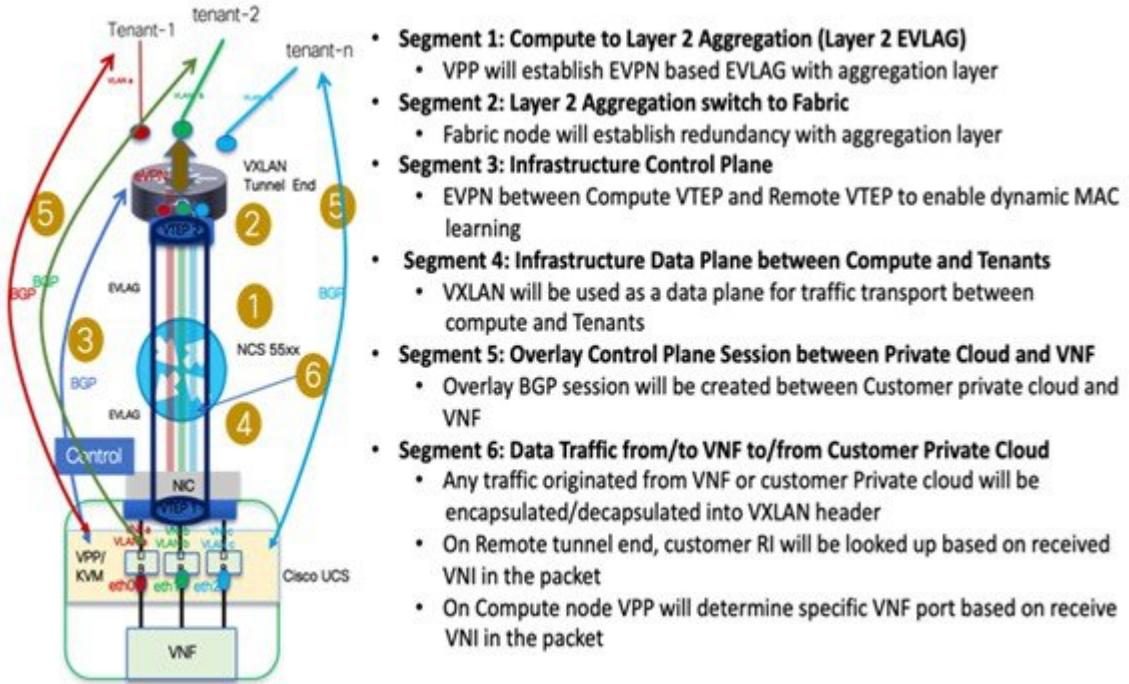
Figure 31: Architecture of a Connected Install

For a disconnected install, both interfaces are private and the artifacts are installed on the Cisco VIM Software Hub using the USB procedure. You must ensure that the ssh interface (br_api) of the management node for each Cisco VIM pod can connect to the enterprise facing interface of the Cisco VIM Software Hub server through Layer 2 or Layer 3 networking. From release Cisco VIM 3.0.0, the Cisco VIM Software Hub is supported over dual-stack network.

Cisco VIM VXLAN EVPN Design

From release Cisco VIM 2.4.3 onwards, seamless connectivity from VNFs of the private cloud to the customer premise private cloud is enabled. The architecture of the Cisco VIM Tenant L2 Connectivity is depicted below:

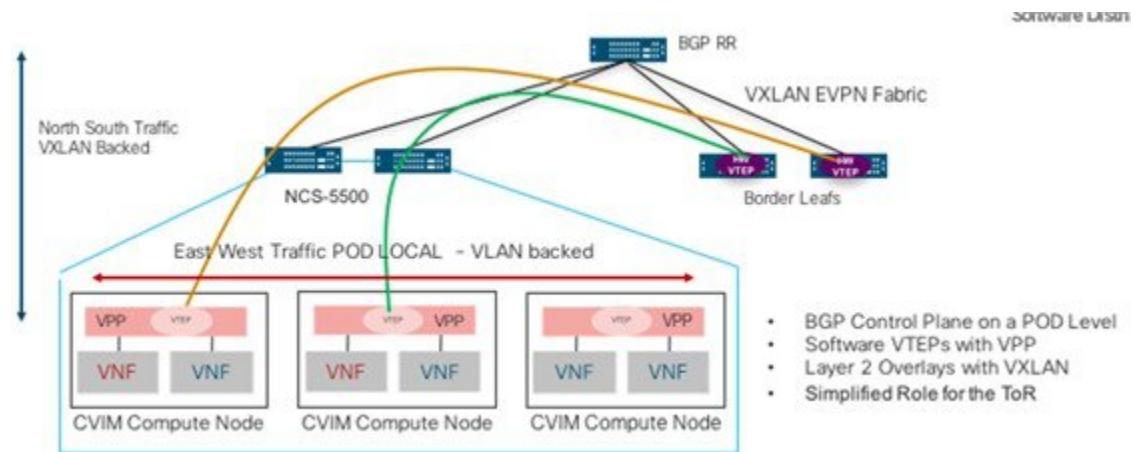
Figure 32: High Level NFVI Tenant L2 Connectivity Architecture



To set up Cisco VIM tenant L2 connectivity architecture, the following assumptions are made:

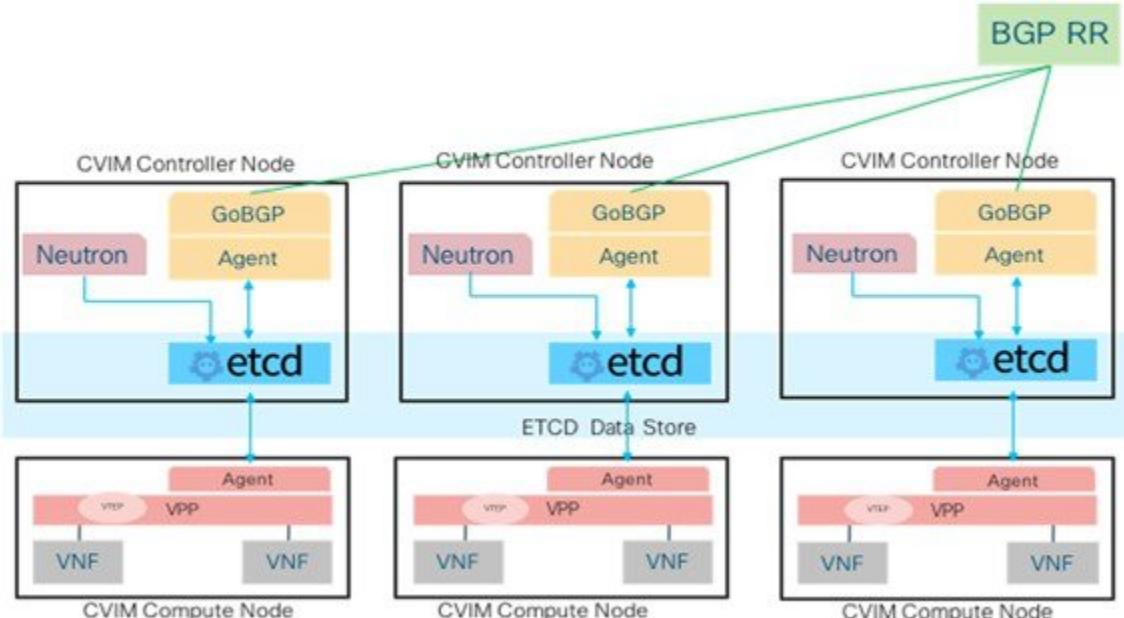
- OpenStack can manage VLAN allocation.
- You must manage VXLAN network and subnet for overlays, and enable OpenStack to use the EVI/VNID by creating appropriate networks/subnets in OpenStack. Cisco VIM supports VNI ranging from 1 to 65535.
- BGP configuration (peer, ASes) will be provided at the time of Cisco VIM cloud deployment through setup_data.yaml.

VXLAN tunnel is used for traffic between the VNF and customer Private cloud, while the VLAN is used for the traffic within the pod or across VNFs. EVPN is used to share L2 reachability information to the remote end, and Cisco NCS 5500 in EVLAG mode acts as a conduit for the traffic. For the VXLAN/EVPN solution to work, Cisco VIM and VXLAN tunnel peers with an external BGP route reflector to exchange IP address to Mac Binding information as shown in the below figure.

Figure 33: Cisco VIM VXLAN EVPN Setup

From a control plane point of view, three instances of GoBGP (in Active-Active-Active mode) run on the controller nodes to establish L3 peering with the external BGP RR for importing or exporting VxLAN routes into or from Cisco VIM respectively. The imported information is then pushed into etcd, to maintain a single source of the information within Cisco VIM.

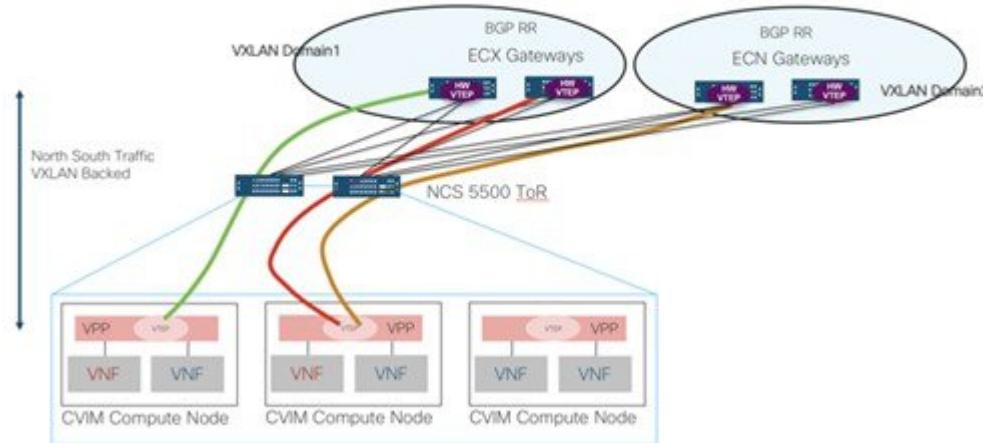
VPP agents create and program VTEP on VPP, and also create a VXLAN tunnel interface for the VM based on the VNI information from Neutron. VPP updates VNF IP/MAC mapping in etcd, which gets exported out through EVPN to the BGP RR.

Figure 34: Cisco VIM VXLAN EVPN Control Plan Design

Multi-VXLAN EVPN Design

From release Cisco VIM 2.4.6 onwards, multiple-AS VXLAN EVPN overlay networks are supported. The following image depicts the schematic view of the multiple-AS VXLAN EVPN overlay network.

North South VXLAN traffic



One set of VXLAN overlays manage the Cloud exchange traffic, while the other set of VXLAN overlays manage the Cloud management traffic. The multi-VXLAN (multi refers to 2) is used to conserve the number of bridge domains (BD) consumed on the Cisco NCS 5500 ToR.

From the control plane point of view, it is similar to that of a single VXLAN architecture.

The multi-VXLAN EVPN based design optionally supports a static implementation of VXLAN technology through head-end replication (HER). HER helps leverage the VXLAN technology, regardless of the hardware/software limitation in the VXLAN feature set at the remote end of the VTEP tunnel.

With the static information defined in the setup_data, VPP performs the HER to all defined remote VTEPs and updates L2FIB (MAC-IP) table based on flood and learn. If EVPN co-exists with HER, Cisco VIM treats it as if two different sets of BGP speakers exist and provides information from each speaker in the same etcd FIB table.

Only drawback of this implementation is that VPP may perform unnecessary flooding. Cisco VIM uses EVPN as the primary mechanism and HER as the fallback methodology. You can add or remove HER to or from an existing EVPN pod through Cisco VIM reconfigure option.

VPP Port Mirroring Support

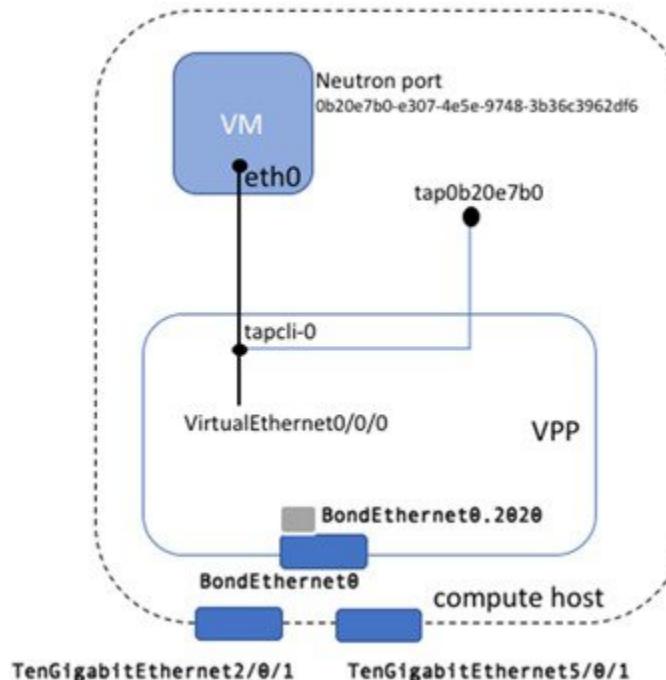
From release CVIM 2.4.3 onwards, all the network traffic between the VM and VPP is over a vhost interface which is in memory and does not use a traditional kernel side interface, when VPP is used as the vSwitch in OpenStack. The network interface is no longer on the host and available within VM, to trace packets or capture them for debugging or other administrative purposes.

Underlying Architecture of the Port Mirroring Tool

Port mirroring works by setting up the following:

1. A span port on vpp to mirror the VirtualEthernet interface corresponding to the VMs vhost interface. This is a tap interface in VPP
2. A tap device (tap0b20e7b0) on the compute host side is set as a kernel interface. A veth pair is created between the tap device on the VPP side (tapcli-0) and kernel side tap device (tap0b20e7b0) as shown in the below figure.

Figure 35: Port mirror components



Limitations of the Port Mirroring Tool

- The port mirror feature uses tap as the interface type for the mirrored traffic. VPP may drop packets designated for this interface, under high load conditions or high traffic scenarios.
- You can only run the Port mirror CLI tools from the VPP container. This require access to the compute node where the VM is running.
- You can only mirror the neutron ports managed by vpp-agent. This means that these have to be vhost interfaces belonging to Openstack VMs. Non VirtualEthernet interfaces are not supported.

Container Workload Support

Cisco VIM supports VM, baremetal, or container-based workloads. To support the container-based workloads, Cisco VIM hosts Cisco Container Platform as an application. The orchestrator creates a common OpenStack tenant and deploys the Cisco Container Platform control plane on it. The orchestrator can also create a tenant cluster if needed.

The Kubernetes clusters deployed are multi-master clusters with three master nodes and N worker nodes. The Cisco Container Platform control plane consists of three masters and three workers. The master and worker nodes run as VMs on OpenStack.

For more information on enabling Cisco Container Platform over Cisco VIM, see [Container Support in Cisco VIM, on page 172](#).



CHAPTER 2

Overview to Cisco NFVI Installation

This chapter describes the Cisco NFVI installation procedures.

- [Cisco NFVI Installation Overview, on page 75](#)

Cisco NFVI Installation Overview

Cisco NFVI installation is divided into two processes:

- Preparation—Preparing the Cisco NFVI pod hardware and configuring all supporting applications including Cisco Integrated Management Controller (IMC) and Cisco UCS Manager.
- Installation—Installing the Cisco NFVI component applications such as Cisco Virtual Infrastructure Manager (VIM), Cisco Insight (Unified Management), and Cisco Virtual Topology System (VTS) with Virtual Topology Forwarder (VTF) based on your Cisco NFVI package.

Cisco NFVI installation depends on the component applications that you install. For example, if you are installing Cisco VTS, install VTC before installing Cisco VIM or Cisco Unified Management (UM). When installing Cisco VIM UM, install the Cisco VIM management node and Insight in a sequence to complete the Cisco VIM installation through Cisco VIM UM. However, if you have Cisco VIM without other Cisco NFVI applications in your package, you can install the Cisco VIM alone in your system.

Consider the following factors before installing the Cisco NFVI components:

- **Internet Access**—Internet access is required to download the Cisco NFVI installation files from cvim-registry.com. If you do not have an Internet access to your management node, you need an alternate server with an Internet access to download the installation files to a USB stick. You can copy the installation files from USB stick to the management node.
- **Cisco NFVI Configurations**—Cisco NFVI configurations are included in the setup_data.yaml file. If you are installing Cisco VIM and not Cisco VIM Insight, you can enter the configurations directly into the setup_data.yaml file with a yaml editor. You can refer to the examples in `setup_data` file (for C and B-series) at the `openstack-configs` directory in the `target install` folder in the management node. For more information on Cisco NFVI data and OpenStack parameters, see [Setting Up Cisco VIM Data Configuration, on page 146](#). If you are installing Cisco VIM Insight, run Cisco NFVI using Insight UI wizard. For more information, see [Installing Cisco VIM Unified Management , on page 211](#).

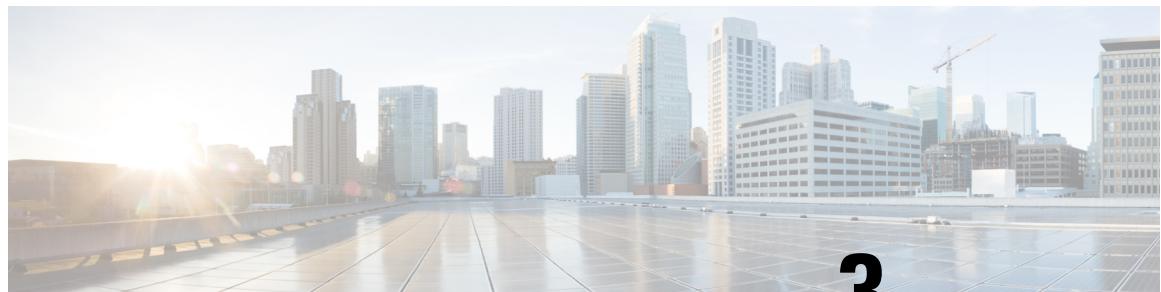
Following are the license options for installing Cisco NFVI:

- Cisco NFVI Basic—Includes Cisco Virtual Infrastructure Manager (VIM), which is an OpenStack Queens release software solution used to enhance the functionality, scale, and performance of the node.
- Cisco NFVI Standard—Includes Cisco VIM and Cisco VIM Insight. Cisco VIM Insight deploys, provisions, and manages Cisco NFVI on Cisco UCS servers.
- Cisco NFVI with third-party monitoring - Includes Cisco VIM with or without Cisco VIM Insight based on the license option chosen, with monitoring of the pod through Zenoss.
- Optional Cisco NFVI Applications—Cisco Virtual Topology System (VTS) is an optional application that can be installed with both Cisco VIM and Cisco VIM Insight. Cisco VTS is a standard-based, open software-overlay management and provisioning system. It automates the data center network fabric provisioning, for virtual and physical infrastructure.

You must perform extra manual installation procedures while installing Cisco VIM. If your package includes Cisco VIM and UM, you must do Cisco VIM manual setup and configuration procedures through the Unified management system (VIM UM). You can manage cloud in Cisco VIM through Cisco VIM UM. Once you start managing the cloud, Cisco recommends you to continue using Cisco VIM UM for future use as well.

The following table helps you to understand the installation sequence.

#	Chapter Title	Audience	Notes
1	Overview to Cisco Network Function Virtualization Infrastructure, on page 1	Pod Administrator	Understanding the Cisco NFVI architecture and networking ensures a successful installation.
2	Overview to Cisco NFVI Installation, on page 75	Pod Administrator	Describes the Cisco NFVI installation procedures.
3	Preparing for Installation on Servers Without Internet Access, on page 77	Pod Administrator	Provides information on the hardware and application preparation procedures, before installing and configuring Cisco NFVI.
4	Preparing for Cisco NFVI Installation, on page 81	Users	Refer to this section, if your management node does not have Internet access.
5	Installing Cisco VTS, on page 111	Users	Refer to this section, if your package includes Cisco Virtual Topology System.. You must install Cisco VTS before you install other Cisco NFVI applications.
6	Installing Cisco VIM, on page 139	Pod Administrator	Describes how to configure and install Cisco VIM. Users with Cisco VIM UM can proceed with the Cisco VIM Insight installation, while users with only Cisco VIM have to complete the full procedure.
7	Installing Cisco VIM Unified Management , on page 211	Users	Refer to this section, if your package includes Cisco VIM UM.
8	Installing Cisco VIM through Cisco VIM Unified Management, on page 229	Users	Describes Cisco VIM UM installation and configuration procedures.
9	Verifying the Cisco NFVI Installation, on page 347	Pod Administrator	Provides methods to verify the Cisco NFVI installation.



CHAPTER 3

Preparing for Installation on Servers Without Internet Access

This section describes the procedures to install Cisco NFVI in a management node without Internet access.

In this scenario, you must:

1. Download the Cisco NFVI installation files to a 64 GB (minimum) USB 2.0 drive on a staging server with Internet access. If the management node is based on M5, you can optionally use USB 3.0 64GB to increase the installation speed significantly.
2. Copy the files to the management node.
 - [Preparing to Install Cisco NFVI on Management Nodes Without Internet Access, on page 77](#)

Preparing to Install Cisco NFVI on Management Nodes Without Internet Access

Following procedure describes how to download the Cisco NFVI installation files onto a USB drive of the staging server with Internet access. You can use the USB to load the Cisco NFVI installation files onto the management node without Internet access.



Note

Cisco recommends you to use Virtual Network Computing (VNC), other terminal multiplexer, or similar screen sessions to complete these steps.

Before you begin

You must have a CentOS 7 staging server (VM, laptop, or UCS server) with a 64 GB USB 2.0 drive only. You can use USB 3.0 64GB if the management node is of type M5. The staging server must have wired Internet connection to download the Cisco VIM installation files onto the USB drive. Once downloaded, you can copy the installation files onto the management node from USB drive.



Note Downloading of the installation files (over 25 GB in size) to the USB drive might take several hours depending on the speed of your Internet connection. Ensure that you disable the CentOS to the sleep mode, for faster installation.

Step 1

On the staging server, use yum to install the following packages:

- PyYAML (yum install PyYAML)
- python-requests (yum install python-requests)

Step 2

Log into Cisco VIM software download site and download the `getartifacts.py` script from external registry:

```
# download the new getartifacts.py file (see example below)
curl -o getartifacts.py
https://username:password@cvim-registry.com/mercury-releases/cvim24-rhel7-osp10/releases/2.4.4/getartifacts.py

curl -o getartifacts.py-checksum.txt
https://username:password@cvim-registry.com/mercury-releases/cvim24-rhel7-osp10/releases/2.4.4/getartifacts.py-checksum.txt

# calculate the checksum and verify that with one in getartifacts.py-checksum.txt
sha512sum getartifacts.py

# Change the permission of getartifacts.py
chmod +x getartifacts.py
```

Step 3

Run `getartifacts.py`. The script formats the USB 2.0 drive (or USB 3.0 drive for M5/Quanta based management node) and downloads the installation files. You must provide the registry username and password, tag ID, and USB partition on the staging server.

```
# ./getartifacts.py -h
usage: getartifacts.py [-h] -t TAG -u USERNAME -p PASSWORD -d DRIVE
                      [--proxy PROXY] [--retry]
                      [--artifacts [ARTIFACTS [ARTIFACTS ...]]]

Script to pull container images.

optional arguments:
  -h, --help            show this help message and exit
  -t TAG, --tag TAG    installer version to pull
  -u USERNAME, --username USERNAME
                        Registry username
  -p PASSWORD, --password PASSWORD
                        Registry password
  -d DRIVE, --drive DRIVE
                        Provide usb drive path
  --proxy PROXY         https_proxy if needed
  --retry              Try to complete a previous fetch
  --artifacts [ARTIFACTS [ARTIFACTS ...]]

Only supported parameter is all and defaults to all if not passed anything
```

This script pulls images from remote registry and copies the contents to usb drive

To identify the USB drive, execute the `lsblk` command before and after inserting the USB drive. The command displays a list of available block devices. The output data will help you to find the USB drive location. Provide the entire drive

path in the `-d` option instead of any partition as shown below. Here, the `tag_id` refers to the Cisco VIM release version 2.4.x.

For example:

```
sudo ./getartifacts.py -t <tag_id> -u <username> -p <password> -d </dev/sdc> [--artifacts ...] [--proxy proxy.example.com] -
```

Note Ensure that you do not remove the USB drive during synchronization.

Note On executing `getartifacts.py`, the following message: *stderr: mount: wrong fs type, bad option, bad superblock on /dev/sdyl, missing codepage or helper program, or other error.* is displayed to notify bad superblock and mount failure. In this case, reformat the drive and use the `fsck` command to recover the drive: `fsck.ext4 -pv /dev/sdc`.

Note As the size of the artifacts is greater than 25G, Cisco recommends you to execute this step over a wired internet connection. It will take few hours to download and populate data on USB drive, depending on the internet connectivity.

The `getartifacts.py` script downloads the following:

- Packages
 - `buildnode-K9.iso`
 - `mercury-installer.tar.gz`
 - `registry-2.3.1.tar.gz`
- `insight-K9.tar.gz`
- `mariadb-app-K9.tar.gz`
- Respective checksums

Step 4 Use the following command to verify the downloaded artifacts and container images:

```
# create a directory
sudo mkdir -p /mnt/Cisco

# /dev/sdc is the USB drive, same as supplied in getartifacts.py python script

# You need to mount the partition with the steps given below:
sudo mount /dev/sdc1 /mnt/Cisco
cd /mnt/Cisco

# execute the test-usb help to look at the options
./test-usb -h

usage: ./test-usb [-h] -- Show this program to check integrity of artifacts in this USB drive
                  [-a] -- Check integrity of all (core and insight) artifacts in this USB drive
                  [-l] -- Location of artifacts

# execute the verification script
./test-usb

# failures will be explicitly displayed on screen, sample success output below
# sample output of ./test-usb execution with 2.4 release
./test-usb
INFO: Checking the integrity of this USB drives
INFO: Checking artifact buildnode-K9.iso
INFO: Checking artifact registry-2.3.1.tar.gz
```

Preparing to Install Cisco NFVI on Management Nodes Without Internet Access

```
INFO: Checking required layers:

# ./test-usb -a
INFO: Checking the integrity of this USB drive
INFO: Checking artifact buildnode-K9.iso
INFO: Checking artifact registry-2.3.1.tar.gz
INFO: Checking artifact mariadb-app-K9.tar.gz
INFO: Checking artifact haproxy-K9.tar.gz
INFO: Checking artifact insight-K9.tar.gz
INFO: Checking required layers:
INFO: 548 layer files passed checksum.
```

If the download fails, an error message is displayed.

For example:

```
# ./test-usb
INFO: Checking the integrity of this USB stick
INFO: Checking artifact buildnode-K9.iso
ERROR: Checksum for artifact buildnode-K9.iso does not match ('SHA512 (buildnode-K9.iso) =
96ec62a0932a0d69daf60acc6b8af2dc4e5ecal32cd3781fc17a494592feb52a7f171eda25e59c0d326fb09194eeda66036cbdc3870dafe74f59cf1f2dce225'
!= 'SHA512 (buildnode-K9.iso) =
a6a9e79fa08254e720a80868555679baeee2dd8f26a0360ad47540eda831617bea0514a117b12ee5f36415b7540afa112a1c904cd69e40d704a8f25d78867acf')
INFO: Checking artifact registry-2.3.1.tar.gz
ERROR: Artifact registry-2.3.1.tar.gz is not present
INFO: Checking required layers:
ERROR: Layer file sha256:002aa1f0fbdaeae7ea25da1d906e732fe9a9b7458d45f8ef7216d1b4314e05207 has a bad
checksum
ERROR: Layer file sha256:5be3293a81773938cdb18f7174bf595fe7323fdc018c715914ad41434d995799 has a bad
checksum
ERROR: Layer file sha256:8009d9e798d9acea2d5a3005be39bcbfe77b9a928e8d6c84374768ed19c97059 has a bad
checksum
ERROR: Layer file sha256:ea55b2fc29b95d835d16d7eeac42fa82f17e985161ca94a0f61846defff1a9c8 has a bad
checksum
INFO: 544 layer files passed checksum.
```

Step 5 To resolve download artifact failures, unmount the USB and run the getartifacts command again with the --retry option.

```
sudo ./getartifacts.py -t <tag_id> -u <username> -p <password> -d </dev/sdc> --retry
```

Step 6 Mount the USB and then run the test-usb command to validate if all the files are downloaded:

```
# /dev/sdc is the USB drive, same as supplied in get artifacts.py python script
sudo mount /dev/sda1 /mnt/Cisco
cd /mnt/Cisco

# execute the verification script
./test-usb

# In case of failures the output of the above command will explicitly display the same on the screen
```

Step 7 When the USB integrity test is done, unmount the USB drive by running the following command:

```
sudo umount /mnt/Cisco
```



CHAPTER 4

Preparing for Cisco NFVI Installation

Before you can install and configure Cisco NFVI, you must complete the following hardware and application preparation procedures provided in the following topics.

- [Installing Cisco NFVI Hardware, on page 81](#)
- [Configuring ToR Switches for C-Series Pods, on page 85](#)
- [Configuring ToR Switches for UCS B-Series Pods, on page 89](#)
- [Preparing Cisco IMC and Cisco UCS Manager, on page 92](#)
- [Installing Management Node on UCS C-series \(M4/M5\), on page 92](#)
- [Installing Management Node on Quanta Servers, on page 95](#)
- [Installing Cisco VIM Software Hub, on page 96](#)
- [Setting Up the UCS C-Series Pod, on page 103](#)
- [Setting Up the UCS B-Series Pod, on page 108](#)
- [Configuring the Out-of-Band Management Switch, on page 110](#)
- [Support of 3rd Party Compute \(HP DL 360 Gen9\), on page 110](#)

Installing Cisco NFVI Hardware

Switch on the Cisco UCS C-Series or B-Series hardware, before you install the Cisco VIM. Depending upon the pod type, you need to set up the CIMC connection or UCSM IP ahead of time. The following table lists the UCS hardware options and network connectivity protocol used with virtual extensible LAN (VXLAN) over a Linux bridge, VLAN over OVS or VLAN over VPP. If Cisco Virtual Topology Services (VTS), an optional Cisco NFVI application, is installed, Virtual Topology Forwarder (VTF) is used with VXLAN for tenants, and VLANs for providers on C-Series pods.

Table 15: Cisco NFVI Hardware and Network Connectivity Protocol

UCS Pod Type	Compute and Controller Node	Storage Node	Network Connectivity Protocol
Rack Type	UCS C220/240 M4/M5	UCS C240 M4 (SFF) with two internal SSDs	OVS/VLAN or VPP/VLAN (only on intel NIC) or ACI/VLAN

UCS Pod Type	Compute and Controller Node	Storage Node	Network Connectivity Protocol
Rack Type	Controller: UCS C220/240 Compute: HP DL360 Gen9 Quanta servers for Fullon or Edge pod	UCS C240 M4 (SFF) with two internal SSDs	OVS/VLAN
C-Series with Cisco VTS	UCS C220/240 M4	UCS C240 M4 (SFF) with two internal SSDs	For tenants: VTF with VXLAN. For providers: VLAN

UCS Pod Type	Compute and Controller Node	Storage Node	Network Connectivity Protocol
C-Series Micropod	<p>UCS 240 M4/M5 with 12 HDD and 2 external SSDs. Pod can be expanded to 16 computes. Each compute will have 2x1.2 TB HDD or</p> <p>UCS 220 M4/M5 with 6 HDD and 1 external SSDs. Pod can be expanded to 16 computes. Each compute will have 2x1.2 TB HDD.</p> <p>Note Refer to the BOM for SSD based install for M5; M5 BOM is based on Intel X710 for control and data plane and XL710 for SRIOV. For exact BOM details, reach out to Cisco VIM product marketing.</p>	Not applicable as it is integrated with Compute and Controller.	OVS/VLAN or VPP/VLAN (on intel NIC).
C-Series Hyperconverged	UCS 240 M4/M5.	UCS C240 M4/M5 (SFF) with 10 HDD and two external SSDs, acts as compute node	OVS/VLAN
B-Series	UCS B200 M4.	UCS C240 M4 (SFF) with two internal SSDs.	OVS/VLAN.



Note The storage nodes boot off two internal SSDs. It also has four external SSDs for journaling, which gives a 1:5 SSD-to-disk ratio (assuming a chassis filled with 20 spinning disks). Each C-Series pod has either a dual-port 10 GE Cisco vNIC 1227 card or dual-port/quad-port Intel X 710 card. UCS B-Series blade servers only support Cisco 1340 and 1380 NICs. For more information on Cisco vNICs, see [LAN and SAN Connectivity for a Cisco UCS Blade](#). Cisco VIM has a Micropod (based on UCS-M4/M5 hardware) which works on Cisco VIC 1227 or Intel NIC 710, with OVS/VLAN or VPP/VLAN (for Intel NIC only) as the virtual network protocol. The Micropod supports with a small, functional, but redundant cloud with capability of adding standalone computes (maximum of 16) to an existing pod.

Cisco VIM supports M4/M5-based Micropod on a VIC/NIC system with OVS, to extend the SRIOV support on a 2x2-port Intel 520 or 2x40G XL710 NIC card. The same pod can be extended to include M5 computes having 40G Cisco VIC with an option to have 2x40G XL710 intel NIC as SRIOV.



Note M5 can only use 2x40G XL710 for SRIOV.

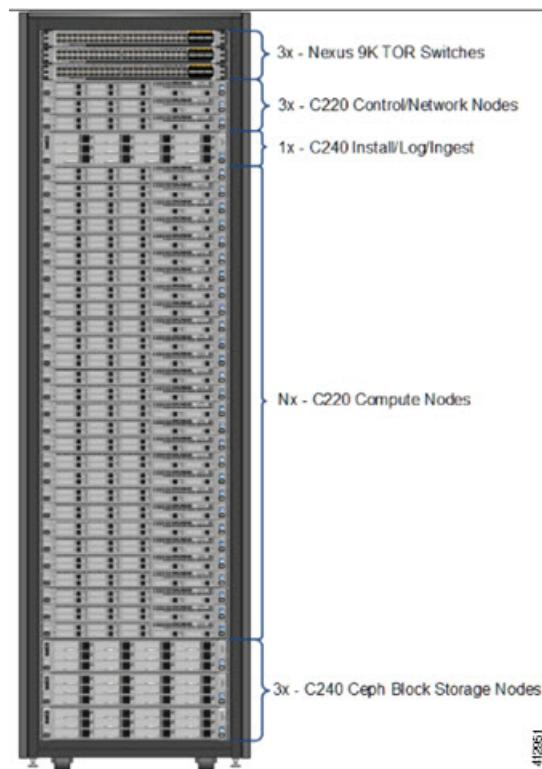
The M5-based Micropod is based on Intel NIC 710 and supports SRIOV over XL710, with OVS/VLAN or VPP/VLAN as the virtual network protocol. From release Cisco VIM 2.4.2 onwards, 40G M5-based Micropod is supported on a VIC (40G)/NIC (2-XL710 for SRIOV) system.

In addition, the Cisco Nexus 9372 or 93180YC, or 9396PX is also available to serve the Cisco NFVI ToR function.

After verifying that you have required Cisco UCS servers, blades and Nexus 93xx, install the hardware following procedures at the following links:

- [Cisco UCS C220 M4 Server Installation and Service Guide](#)
- [Cisco UCS C240 M4 Server Installation and Service Guide](#)
- [Cisco UCS B200 Blade Server and Installation Note](#)
- [Cisco Nexus 93180YC,9396PX, 9372PS and 9372PX-E NX-OS Mode Switches Hardware Installation Guide](#)

The figure below shows C-Series Cisco NFVI pod. Although the figure shows a full complement of UCS C220 compute nodes, the number of compute nodes vary depending on the implementation requirements. The UCS C220 control and compute nodes can be replaced with UCS 240 series. However, in that case the number of computes fitting in one chassis system is reduced by half.

Figure 36: Cisco NFVI C-Series Pod

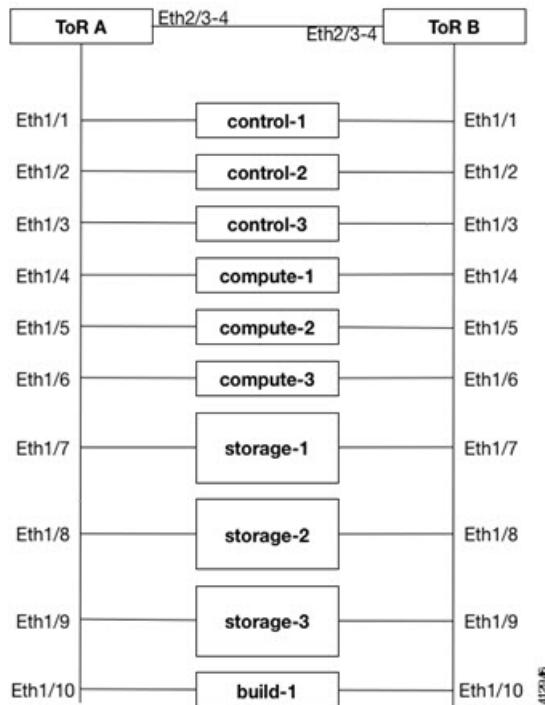
Note The combination of UCS-220 and UCS-240 within the compute and control nodes is not supported.

For more information on wiring schematic of various pod configuration, see [Appendix, on page 353](#).

Configuring ToR Switches for C-Series Pods

During installation, the Cisco VIM installer creates vNICs on each of the two physical interfaces and creates a bond for the UCS C-Series pod. Before this, manually configure the ToR switches to create a vPC with the two interfaces connected to each server. Use identical Cisco Nexus 9372, or 93180YC, or 9396PX switches for the ToRs. Cisco recommends you to use the N9K ToR software versions for setup: 7.0(3)I4(6) 7.0(3)I6(1). For information on the wiring details for each pod type on a C-series-based install, see [Appendix](#) section

Complete the following steps to create a vPC on a pair of Cisco Nexus ToR switches. The steps use the following topology as an example. Modify the configuration as it applies to your environment. Cisco VIM optionally supports auto-configuration of ToR for N9K series only. If auto-configuration of ToR is opted, you can skip the following steps:

Figure 37: ToR Configuration Sample**Step 1**

Change the vPC domain ID for your configuration. The vPC domain ID can be a unique number. The IP address on the other switch mgmt0 port is used for the keepalive IP. Change it to the IP used for your network.

For the preceding example, the following is the configuration:

```

ToR-A (mgmt0 is 172.18.116.185)
feature vpc
vpc domain 116
peer-keepalive destination 172.18.116.186
ToR-B (mgmt0 is 172.18.116.186)
feature vpc
vpc domain 116
peer-keepalive destination 172.18.116.185

```

Because both switches are cabled identically, the remaining configuration is identical on both switches. In this example, topology Eth2/3 and Eth2/4 are connected to each other and combined into a port channel that functions as the vPC peer link.

```

feature lacp
interface Ethernet2/3-4
channel-group 116 mode active
interface port-channel116
switchport mode trunk
vpc peer-link

```

Step 2

For each VLAN type, (mgmt_vlan, tenant_vlan_range, storage, api, external, provider), execute the following on each ToR:

```
vlan <vlan_type>
  no shut
```

- Step 3** Configure all the interfaces that are connected to the servers as the members of the port channels. In the example, only ten interfaces are shown. But you must configure all interfaces that are connected to the server.

Note If interfaces have configuration from previous deployments, you can remove them by entering `default interface Eth1/1-10, then no interface Po1-10.`

a. For deployment with any mechanism driver on Cisco VIC

There is no configuration differences among different roles (controllers/computes/storages). The same configuration applies to all interfaces.

```
interface Ethernet 1/1
channel-group 1 mode active
interface Ethernet 1/2
channel-group 2 mode active
interface Ethernet 1/3
channel-group 3 mode active
interface Ethernet 1/4
channel-group 4 mode active
interface Ethernet 1/5
channel-group 5 mode active
interface Ethernet 1/6
channel-group 6 mode active
interface Ethernet 1/7
channel-group 7 mode active
interface Ethernet 1/8
channel-group 8 mode active
interface Ethernet 1/9
channel-group 9 mode active
interface Ethernet 1/10
channel-group 10 mode active
```

b. For deployment with OVS/VPP with VLAN on Intel NIC

The interface configuration is same as Cisco VIC as shown in the above section. However, number of switch interfaces that are configured is more in the case of Intel NIC as it has dedicated control and data physical ports. For SRIOV switchport, no port channel is configured and the participating VLAN can be in trunk mode. In case of pod based on Quanta servers, or HPE as computes, configure the control and data plane VLANs in trunk mode on the switch ports connected to the OCP and LOM cards, respectively.

- Step 4** Configure the port channel interface as vPC and trunk all VLANs. For Intel NIC, you must configure native vlan and set it to mgmt vlan on the control ports so that PXE boot does not fail. Skip to listen or learn in spanning tree transitions, and ensure that you do not suspend the ports if LACP packets are not received. Also, configure it with large MTU of 9216 to avoid Ceph installation failure. The last configuration allows you to start the servers before the bonding is set up.

```
interface port-channel1-9
shutdown
spanning-tree port type edge trunk
spanning-tree bpduguard enable
switchport mode trunk
switchport trunk native vlan mgmt_vlan for the control ports when Intel NIC is used
switchport trunk allowed vlan <mgmt_vlan, tenant_vlan_range, storage, api, external, provider>
no lACP suspend-individual
mtu 9216
```

Configuring ToR Switches for C-Series Pods

```
vpc <1-9>
no shutdown
```

Step 5 Identify the port channel interface that connects to the management node on the ToR:

```
interface port-channel10
shutdown
spanning-tree port type edge trunk
switchport mode trunk
switchport trunk allowed vlan <mgmt_vlan>
no lacp suspend-individual
vpc 10
no shutdown
```

Step 6 Check the port channel summary status. The ports connected to the neighbor switch have to be in (P) state. Before the server installation, the server facing interfaces must be in (I) state. After installation, they have to be in (P) state, which means they are up and in port channel mode.

```
gen-leaf-1# show port-channel summary
Flags: D - Down P - Up in port-channel (members)
I - Individual H - Hot-standby (LACP only)
s - Suspended r - Module-removed
S - Switched R - Routed
U - Up (port-channel)
M - Not in use. Min-links not met
-----
Group Port- Type Protocol Member Ports
Channel
-----
1 Po1 (SD) Eth LACP Eth1/1(I)
2 Po2 (SD) Eth LACP Eth1/2(I)
3 Po3 (SD) Eth LACP Eth1/3(I)
4 Po4 (SD) Eth LACP Eth1/4(I)
5 Po5 (SD) Eth LACP Eth1/5(I)
6 Po6 (SD) Eth LACP Eth1/6(I)
7 Po7 (SD) Eth LACP Eth1/7(I)
8 Po8 (SD) Eth LACP Eth1/8(I)
9 Po9 (SD) Eth LACP Eth1/9(I)
10 Po10 (SD) Eth LACP Eth1/10(I)
116 Po116 (SD) Eth LACP Eth1/116(I)
```

Step 7 Enable automatic Cisco NX-OS errdisable state recovery:

```
errdisable recovery cause link-flap
errdisable recovery interval 30
```

Cisco NX-OS places links that flap repeatedly into errdisable state to prevent spanning tree convergence problems caused by non-functioning of hardware. During Cisco VIM installation, the server occasionally triggers the link flap threshold, so enabling automatic recovery from this error is recommended.

```
errdisable recovery cause link-flap
errdisable recovery interval 30
```

Step 8 If you are installing Cisco Virtual Topology Systems, an optional Cisco NFVI application, enable jumbo packets and configure 9216 MTU on the port channel or Ethernet interfaces. For example:

Port channel:

```
interface port-channel10
  switchport mode trunk
  switchport trunk allowed vlan 80,323,680,860,2680,3122-3250
```

```
mtu 9216
vpc 10
```

Ethernet:

```
interface Ethernet1/25
switchport mode trunk
switchport trunk allowed vlan 80,323,680,860,2680,3122-3250
mtu 9216
```

Configuring ToR Switches for UCS B-Series Pods

Complete the following steps to create a vPC on a pair of Cisco Nexus ToR switches for a UCS B-Series pod. The steps are similar to configuring ToR switches for C-Series pods, with some differences. Here, the two ToR switches are Storm-tor-1 (mgmt0 is 172.18.116.185) and Storm-tor-2 (mgmt0 is 172.18.116.186). Modify the configuration as applicable to your environment.

- Step 1** Change the vPC domain ID for your configuration. The vPC domain ID can be any unique number. The IP address on the other switch mgmt0 port is used for the keepalive IP. Change it to the IP used for your network.

Storm-tor-1 (mgmt0 is 172.18.116.185).

```
feature vpc
vpc domain 116
    peer-keepalive destination 172.18.116.186
for each vlan_type (mgmt_vlan, tenant_vlan_range, storage, api, external, provider); # execute the
following for each vlan
    vlan <vlan_type>
        no shut
vrf context management
    ip route 0.0.0.0/0 172.18.116.1

interface mgmt0
    vrf member management
    ip address 172.18.116.185/24
```

Storm-tor-2 (mgmt0 is 172.18.116.186).

```
feature vpc
vpc domain 116
    peer-keepalive destination 172.18.116.185
for each vlan_type (mgmt_vlan, tenant_vlan_range, storage, api, external, provider); # execute the
following for each vlan
    vlan <vlan_type>
        no shut
vrf context management
    ip route 0.0.0.0/0 172.18.116.1

interface mgmt0
    vrf member management
    ip address 172.18.116.186/24
```

- Step 2** As both switches are cabled identically, the rest of the settings are identical on both the switches. Configure all the interfaces that are connected to the fabric interconnects for VPC.

Configuring ToR Switches for UCS B-Series Pods

```

feature lacp
interface port-channel1
    description "to fabric interconnect 1"
    switchport mode trunk
    vpc 1
interface port-channel2
    description "to fabric interconnect 2"
    switchport mode trunk
    vpc 2
interface Ethernet1/43
    description "to fabric interconnect 1"
    switchport mode trunk
    channel-group 1 mode active
interface Ethernet1/44
    description "to fabric interconnect 2"
    switchport mode trunk
    channel-group 2 mode active

```

Step 3 Create the port-channel interface on the ToR that connects to the management node:

```

interface port-channel3
    description "to management node"
    spanning-tree port type edge trunk
    switchport mode trunk
    switchport trunk allowed vlan <mgmt_vlan>
    no lacp suspend-individual
    vpc 3
interface Ethernet1/2
    description "to management node"
    switchport mode trunk
    channel-group 3 mode active

```

Step 4 To enable multicast traffic for Cisco VIM, change the Nexus 9000 configuration including enabling the PIM routing and OSPF:

```

feature ospf
feature pim
feature interface-vlan
feature hsrp

ip pim rp-address 192.1.1.1 group-list 224.0.0.0/4
ip pim ssm range 232.0.0.0/8
ip pim anycast-rp 192.1.1.1 192.168.100.1
ip pim anycast-rp 192.1.1.1 192.168.100.2

interface Ethernet1/18
    description "Mcast Sender Example"
    switchport trunk allowed vlan <provider/tenant vlan id>

interface loopback7
    ip address 192.1.1.1/32
    ip router ospf 777 area 0.0.0.0
    ip pim sparse-mode

router ospf 777
    router-id 1.1.1.1
    area 0.0.0.0 default-cost 10

interface Vlan<provider/tenant vlan id>
    no shutdown
    ip address <IP address/mask>
    no ip ospf passive-interface
    ip router ospf 777 area 0.0.0.0
    ip pim sparse-mode

```

```
hsrp 101
priority 11
ip <provider/tenant gateway address>
```

Storm-tor-1

```
interface loopback0
  ip address 192.168.100.1/32
  ip router ospf 777 area 0.0.0.0
  ip pim sparse-mode
```

Storm-tor-2

```
interface loopback0
  ip address 192.168.100.2/32
  ip router ospf 777 area 0.0.0.0
  ip pim sparse-mode
```

Step 5

If Cisco VIM implementation has extensive multicast traffic, prioritize the multicast traffic by setting up the following service classes on the ToR switches and enabling the media QOS profile as described in the *UCS Manager Common Access Information for B-Series Pods* in [Setting Up Cisco VIM Data Configuration, on page 146](#). The Nexus 9000 configuration is as follows:

```
class-map type qos match-all class-silver
  match cos 2
class-map type qos match-all class-bronze
  match cos 1

policy-map type qos system-level-qos
  class class-silver
    set qos-group 3
  class class-bronze
    set qos-group 2

class-map type queuing class-silver
  match qos-group 3
class-map type queuing class-bronze
  match qos-group 2

policy-map type queuing Uplink-out_policy
  class type queuing class-silver
    bandwidth percent 60
    priority
  class type queuing class-bronze
    bandwidth percent 30
  class type queuing class-default
    bandwidth percent 10
class-map type network-qos class-silver
  match qos-group 3
class-map type network-qos class-bronze
  match qos-group 2

policy-map type network-qos system-level-net-qos
  class type network-qos class-silver
    set cos 2
    mtu 9126
    multicast-optimize
  class type network-qos class-bronze
    set cos 1
    mtu 9126
  class type network-qos class-default
    mtu 9126

system qos
service-policy type queuing input fcoe-default-in-policy
```

```
service-policy type queuing output Uplink-out_policy
service-policy type qos input system-level-qos
service-policy type network-qos system-level-net-qos
```

Step 6 Enable jumbo frames for each ToR port-channel that connects to the Fabric Interconnects:

```
interface port-channel<number>
  mtu 9216
```

Note Ensure that you enable jumbo frames in the `setup_data.yaml` file. See the *UCS Manager Common Access Information for B-Series Pods* section in [Setting Up Cisco VIM Data Configuration, on page 146](#).

Preparing Cisco IMC and Cisco UCS Manager

Cisco NFVI requires specific Cisco Integrated Management Controller (IMC) and Cisco UCS Manager firmware versions and parameters. The Cisco VIM bare metal installation uses the Cisco IMC credentials to access the Cisco IMC interface which is used to delete and create vNICS and to create bonds.

Complete the following steps to verify if Cisco IMC and UCS Manager are ready for Cisco NFVI installation:

- Step 1** Verify that each Cisco UCS server uses Cisco IMC firmware version of either 2.0 series (2.0(13i) or greater preferably 2.0(13n)) or 3.0 series (use 3.0.3(f) or later). You can download the latest Cisco IMC ISO image from the Cisco Software Download site. For upgrade procedures, see the [Cisco UCS C-Series Rack-Mount Server BIOS Upgrade Guide](#).
 - Step 2** For UCS B-Series pods, verify that the Cisco UCS Manager version is one of the following: 2.2(5a), 2.2(5b), 2.2(6c), 2.2(6e), 3.1(c).
 - Step 3** For UCS C-Series pods, verify the following Cisco IMC information is added: IP address, username, and password.
 - Step 4** For UCS B-Series pods, verify the following UCS Manager information is added: username, password, IP address, and resource prefix. The resource prefix maximum length is 6. The provisioning network and the UCS Manager IP address must be connected.
 - Step 5** Verify that no legacy DHCP/Cobbler/PXE servers are connected to your UCS servers. If so, disconnect or disable the interface connected to legacy DHCP, Cobbler, or PXE server. Also, delete the system from the legacy cobbler server.
 - Step 6** Verify Cisco IMC has NTP enabled and is set to the same NTP server and time zone as the operating system.
-

Installing Management Node on UCS C-series (M4/M5)

This procedure installs RHEL 7.6 with the following modifications:

- Hard disk drives are setup in RAID 6 configuration with one spare HDD for eight HDDs deployment, two spare HDDs for 9 to 16 HDDs deployment, or four spare HDDs for 17 to 24 HDDs deployment.
- Networking: Two bridge interfaces are created; one for the installer API (`br_api` off the LOM interfaces) and the other for provisioning (`br_mgmt` off the Cisco VIC on the MLOM or off a X710 based Intel NIC depending on the BOM). Each bridge interface has underlying interfaces bonded together with 802.3ad. Provision interfaces are 10/40 GE interfaces (either off Cisco VICs or X710 Intel NIC (first 2 ports of Intel NIC)). API interfaces are 1/10 GE LOMs based on the BOM. For using NFVbench, you require another NIC card constituting off 2xIntel 520, or 2xIntel 710XL, or 4xIntel710 X. For management node

BOM (Intel NIC based), ensure that you place the NIC for NFVbench at a slot higher than that of the br_mgmt based Intel NIC.

- The installer code is placed in /root/.
- SELinux is enabled on the management node for security.

Before you begin

Verify that the Cisco NFVI management node where you plan to install the Red Hat for Enterprise Linux (RHEL) operating system is a Cisco UCS C240 M4/M5 Small Form Factor (SFF) with 8, 16, or 24 hard disk drives (HDDs). In addition, the management node must be connected to your enterprise NTP and DNS servers. If your management node server does not meet these requirements, do not continue until you install a qualified UCS C240 server. Also, verify that the pod has MRAID card.

Step 1 Log into the **CIMC GUI** of Cisco NFVI management node.

Step 2 Follow steps in [Configuring the Server Boot Order](#) to set the boot order to boot from Local HDD.

Step 3 Follow steps in Cisco UCS [Configure BIOS Parameters](#) to set the following advanced BIOS settings:

For Management node based on UCS M4 boxes set the following for BIOS Parameters:

- PCI ROM CLP—Disabled
- PCH SATA Mode—AHCI
- All Onboard LOM Ports—Enabled
- LOM Port 1 OptionROM—Disabled
- LOM Port 2 OptionROM—Disabled
- All PCIe Slots OptionROM—Enabled
- PCIe Slot:1 OptionROM—Enabled
- PCIe Slot:2 OptionROM—Enabled
- PCIe Slot: MLOM OptionROM—Disabled
- PCIe Slot:HBA OptionROM—Enabled
- PCIe Slot:FrontPcie1 OptionROM—Enabled
- PCIe Slot:MLOM Link Speed—GEN3
- PCIe Slot:Riser1 Link Speed—GEN3
- PCIe Slot:Riser2 Link Speed—GEN3
- MLOM OptionROM—Enabled

For Management node based on UCS M5 boxes set the following for BIOS Parameters:

- All Onboard LOM Ports—Enabled
- LOM Port 1 OptionROM—Disabled
- LOM Port 2 OptionROM—Disabled

- PCIe Slot:1 OptionROM—Enabled
- PCIe Slot:2 OptionROM—Enabled
- MLOM OptionROM—Enabled
- MRAID OptionROM—Enabled

Other parameters must be set to default.

Step 4

Click **Save Changes**.

Step 5

Add the management node vNICs to the provisioning VLAN to provide the management node with access to the provisioning network:

- a) In the CIMC navigation area, click the **Server** tab and select **Inventory**.
- b) In the main window, click the **Cisco VIC Adapters** tab.
- c) Under Adapter Card, click the **vNICs** tab.
- d) Click the first vNIC and choose **Properties**.
- e) In the vNIC Properties dialog box, enter the provisioning VLAN in the Default VLAN field and click **Save Changes**.
- f) Repeat Steps **a** through **e** for the second vNIC.

Note Delete any additional vNICs configured on the UCS server beyond the two default ones.

Step 6

Download the Cisco VIM Buildnode ISO image to your computer from the given location.

Step 7

In CIMC, launch the KVM console.

Step 8

Mount the Cisco VIM Buildnode ISO image as a virtual DVD.

Step 9

Reboot the UCS server, then press **F6** to enter the boot menu.

Step 10

Select the KVM-mapped DVD to boot the Cisco VIM Buildnode ISO image provided with the install artifacts.

Step 11

In boot menu, select **Install Cisco VIM Management Node**. This is default selection and it gets automatically selected after the timeout.

Step 12

At the prompts, answer the following questions to install the Management node as unified management node only or not:

- Hostname—Enter the management node hostname (The hostname length must be 32 or less characters).
- Select **Yes** to Install as Unified Management only when required. Migration from one to another is not supported.
- API IPv4 address—Enter the management node API IPv4 address in CIDR (Classless Inter-Domain Routing) format. For example, 172.29.86.62/26
- API Gateway IPv4 address—Enter the API network default gateway IPv4 address.
- MGMT IPv4 address—Enter the management node MGMT IPv4 address in CIDR format. For example, 10.30.118.69/26

Note The MGMT IPv4 entry is not required, if the management node is installed as “unified management node only”

- Prompt to enable static IPv6 address configuration—Enter **Yes** to continue input similar IPv6 address configuration for API and MGMT network, or **No** to skip if IPv6 is not needed.
- API IPv6 address—Enter the management node API IPv6 address in CIDR (Classless Inter-Domain Routing) format. For example, 2001:c5c0:1234:5678:1001::5/8.
- Gateway IPv6 address—Enter the API network default gateway IPv6 address.

- MGMT IPv6 address—Enter the management node MGMT IPv6 address in CIDR format. For example, 2001:c5c0:1234:5678:1002::5/80
- DNS server—Enter the DNS server IPv4 address or IPv6 address if static IPv6 address is enabled.
- Option for Teaming Driver for Link Aggregation (answer **yes** when Nexus Switch is the ToR, and answer **no** when Cisco NCS 5500 is ToR): <yes|no> "

After you enter the management node IP addresses, the Installation options menu appears. In the installation menu, there are several options, fill in the options that are listed below (option8 and 2) and leave everything else as it is. If you are unable to start the installation, enter **r** to refresh the Installation menu.

Step 13 In the Installation menu, select option **8** to enter the root password.

Step 14 At the Installation Menu, select option **2** to enter the time zone.

Step 15 At the Timezone settings, select the option **1** as option **2** is not supported.

Step 16 Enter the number corresponding to your time zone.

Step 17 Enter the number for your region.

Step 18 Choose the city and then confirm the time zone settings.

Note NTP server IP must not be entered at the time of setting time zone.

Step 19 After confirming your time zone settings, enter **b** to start the installation.

Step 20 After the installation is complete, press **Return** to reboot the server.

Step 21 After the reboot, check the management node clock using the Linux **date** command to ensure that the TLS certificates are valid, for example:

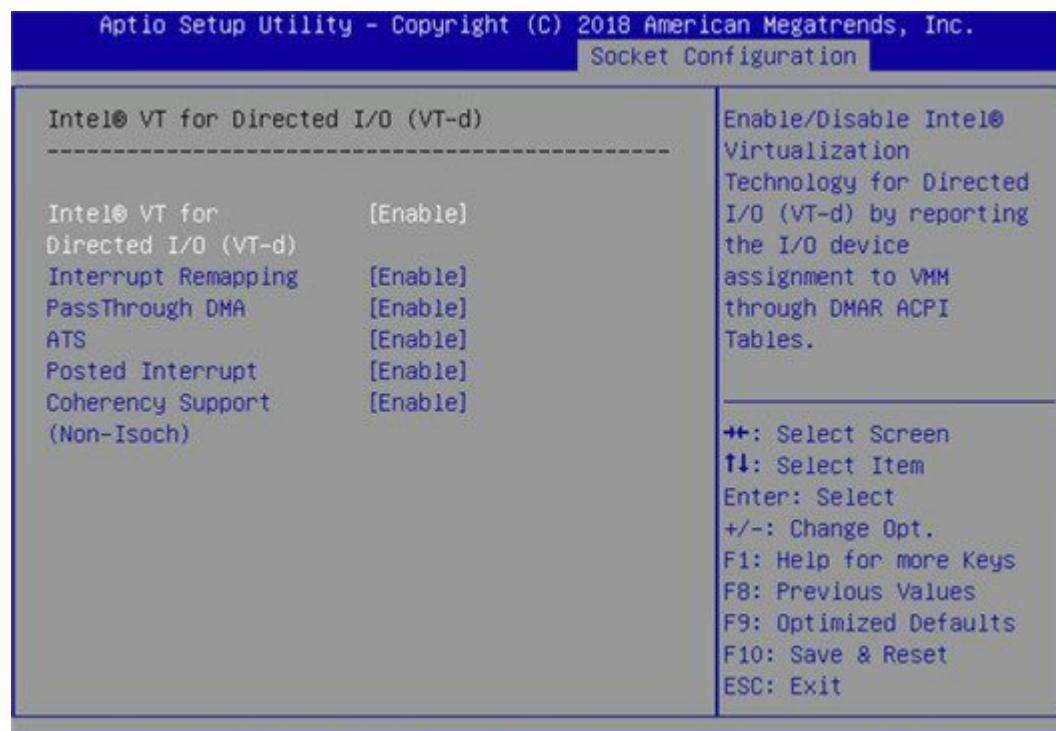
```
#date  
Mon Aug 22 05:36:39 PDT 2016  
  
To set date:  
#date -s '2016-08-21 22:40:00'  
Sun Aug 21 22:40:00 PDT 2016  
  
To check for date:  
#date  
Sun Aug 21 22:40:02 PDT 2016
```

Installing Management Node on Quanta Servers

Most of the settings in the server remains default. To run NFVbench, you must enable the Intel VT for Directed I/O (VT-d) on the Management node.

To enable the Intel VT for Directed I/O, navigate to the following path:

BIOS Setup -> Socket Configuration -> IIO Configuration -> Intel VT for Directed I/O (VT-d) -> Intel VT for Directed I/O (VT-d) -> Enable



To enable NFVbench on a Quanta management node:

- Reboot the MGMT node, hit F2 or DEL to enter BIOS:
- Choose Socket Configuration -> IIO Configuration -> Intel(R) VT for Directed I/O (VT-d)
- Set Intel(R) VT for Directed I/O (VT-d) to **Enable**
- Press the key F10 to save and exit.

Installing Cisco VIM Software Hub

Cisco VIM Software Hub alleviates the need for Cisco VIM management nodes to have internet connectivity and helps to remove the logistics of shipping USBs to multiple pods across the enterprise for software installation or update of the cloud.



Note

The project name for Cisco VIM Software Hub was SDS (Software Delivery Server), therefore you might encounter references to SDS in the configuration files, directory paths and automation outputs.

Before you begin

Prerequisites for Cisco VIM Software Hub Nodes

- Ensure that the Cisco VIM management nodes have connectivity to Cisco VIM Software Hub.

- Ensure that the Cisco VIM Software Hub node where you want to install the `buildnode.iso` is a Cisco UCS C240 M4 Small Form Factor (SFF) with 16 or 24 hard disk drives (HDDs).
- Ensure that the Cisco VIM Software Hub node is connected to the enterprise NTP and DNS servers.
- Ensure that the Cisco VIM Software Hub node has a hardware MRAID and a cache card.

Prerequisites for Cisco VIM Software Hub Server

• TLS certificate (For production environment)

On the Cisco VIM Software Hub server, configure a secure registry so that the pods can obtain the container images over TLS. You need to provide a certificate signed by a trusted third-party CA authority and the **CommonName** in the certificate must match the Cisco VIM Software Hub Registry FQDN name. The `sds_setup_data.yaml` has 3 fields:

- `SSL_CERT_FILE`: Path of x509 certificate obtained from a trusted CA authority
- `SSL_CERT_KEY_FILE`: Path of private key obtained from a trusted CA authority
- `SSL_CERT_CHAIN_FILE`: Path of a single ssl cert chain file. The trusted CA authority might provide you the x509 cert for your domain, intermediate x509 cert and root CA cert. You need to create a single ssl cert chain file using the commands below:

```
# cat <x509 domain cert> >> ssl_chain_file.cer
# cat <intermediate ca cert> >> ssl_chain_file.cer
# cat <root ca cert> >> ssl_chain_file.cer
```

• Self-signed certificate (For internal use)

Cisco recommends to use a trusted CA signed certificate when a Cisco VIM Software Hub node is used in production environment. For internal testing and POC, Cisco supports Cisco VIM Software Hub node with self signed certificate. Follow the below steps to generate the self-signed certificate:

```
# openssl genrsa -des3 -out https_reverse_proxy.key 2048
# openssl req -new -key https_reverse_proxy.key -out https_reverse_proxy.csr
# cp https_reverse_proxy.key https_reverse_proxy.key.org
# openssl rsa -in https_reverse_proxy.key.org -out https_reverse_proxy.key
# openssl x509 -req -days 365 -in https_reverse_proxy.csr -signkey
https_reverse_proxy.key -out https_reverse_proxy.cer
```

Generate the certificate with the same FQDN as specified in the `sds_setup_data.yaml`. Populate the `SSL_CERT_FILE`, `SSL_CERT_KEY_FILE` and `SSL_CERT_CHAIN_FILE` in `sds_setup_data.yaml`. In case of self-signed certificate, use the same x509 certificate for both cert file and cert chain file. You need to manually trust the self-signed certificate. The operator needs to execute the commands below on both Cisco VIM Software Hub server and CVIM pod management node:

```
# cp <x509 cert> /etc/pki/ca-trust/source/anchors/ca.crt
# update-ca-trust extract

For docker registry to work with self signed certificates, execute the commands
below on SDS server.
# mkdir /etc/docker/certs.d/<fqdn>
# cp <x509 cert> /etc/docker/certs.d/<fqdn>/ca.crt
```

• DNS server

Ensure that the pods and the Cisco VIM Software Hub server are reachable to the DNS server and the DNS server must be able to resolve the Cisco VIM Software Hub Registry FQDN. If the enterprise does not have a unified DNS, then you need to populate the `/etc/hosts` file with FQDN after provisioning a node using the ISO archive file.

Installing Cisco VIM Software Hub Node

The steps to install an Cisco VIM Software Hub node are similar to the steps in [Installing Management Node on UCS C-series \(M4/M5\), on page 92](#). The only difference being, in Step 11 of the task, you need to choose the option to configure the server as an Cisco VIM Software Hub server. In the subsequent prompts, you can enter information such as the hostname, ipv4 or ipv6 addresses for br_public and br_private interfaces, and gateway addresses, similar to the [Installing Management Node on UCS C-series \(M4/M5\), on page 92](#) task.

The node is installed with RHEL 7.4 with the following modifications:

- Hard disk drives are set up in RAID 6 configuration with two spare HDDs for a 16 HDDs deployment or four spare HDDs for a 24 HDDs deployment.
- Two bridge interfaces are created, namely, `br_public` and `br_private`. In case of a connected Cisco VIM Software Hub server, the `br_public` interface is connected to the internet. The `br_private` interface is local to your datacenter. The management node for every Cisco VIM pod must be reachable to the `br_private` interface of Cisco VIM Software Hub server through the `br_api` interface. Each bridge interface has underlying interfaces bonded together with 802.3ad. For the Cisco VIM Software Hub, the private interfaces are over 10 GE Cisco VICs, while the public interfaces are 1 GE LOMs.
- Security_Enhanced Linux (SELinux) is enabled on the management node for security.
- The Cisco VIM Software Hub code consists of packages with installer code. After provisioning the server with ISO, the installer code is placed in the following path:

`/root/cvim_sds-<tag>`

Setting up Cisco VIM Software Hub for Cisco VIM Artifact Distribution

You must configure a `sds_setup_data.yaml` file for each installer workspace.

Step 1

Copy the EXAMPLE file from the `openstack-configs` directory and save it as `sds_setup_data.yaml`.

Step 2

If you want to install a release tag on a Cisco VIM Software Hub server, update the fields in the `sds_setup_data.yaml` file as necessary.

```
## Configuration File:
# This file is used as an inventory file to setup CVIM SDS (software delivery server).
#####
# User Defined Configuration File.
# Information in this file is specific to the SDS setup.
#####
SSL_CERT_FILE: <abs_location_for_cert_path of x509 certificate>
SSL_CERT_KEY_FILE: <abs_location_for_cert_priv_key of x509 certificate>
SSL_CERT_CHAIN_FILE: <abs_location_for_cert_chain_file of x509 certificate>
#####
# Registry credentials to access the CVIM registry (Cisco Supplied)
#####
CVIM_REGISTRY_USERNAME: <username>
CVIM_REGISTRY_PASSWORD: <password>
NETWORKING:
```

```

## Max. NTP servers = 4, min of 1
ntp_servers: <ntp.server1.fqdn.com, ntp.server2.fqdn.com >
or
ntp_servers: [ipv6_address, 'ipv4_address'] # "," separated IPv4 or IPv6 address info
http_proxy_server: <proxy.domain.com:8080> # optional, needed if the pod is behind a proxy
https_proxy_server: <proxy.domain.com:8080> # optional, needed if the pod is behind a proxy
SDS_REGISTRY_NAME: <satellite.fqdn.com> #SDS registry name needs to resolve to valid IP
SDS_REGISTRY_USERNAME: <username>
SDS_REGISTRY_PASSWORD: <password>
# (Optional) SDS users who can only pull images from SDS docker registry
SDS_READ_ONLY_USERS:
  - username: <user1>
    password: <password1>
  - username: <user2>
    password: <password2>

```

Step 3 Save the `sds_setup_data.yaml` file in the following path:

```
openstack-configs directory under /root/cvim_sds-<tag>
```

Installing Cisco VIM Software Hub in Connected Mode

In the Connected mode, the Cisco VIM Software Hub server has a publicly routable IP address, and the server can connect to the cvim-registry. When the Cisco VIM Software Hub server is initially configured with the ISO, Cisco VIM Cisco VIM Software Hub workspace of that release is preinstalled in the `/root/` directory.

Step 1 Download the `mercury-installer.tar.gz` file of the release that you want.

Step 2 Unzip the zip file manually and rename the unzipped file as `cvim_sds-<release>`.

Step 3 Perform the following steps:

- Place a valid TLS certificate in the `/root/cvim_sds-<tag>/openstack-configs` directory.
- Update the fields of the Cisco VIM Software Hub setup data file and save it in the following directory:
`/root/cvim_sds-<tag> openstack-configs`

Step 4 To install the release on the Cisco VIM Software Hub server, navigate to the `/root/cvim_sds-<target-tag>` directory on the Cisco VIM Software Hub server and run the following command:

```
# cd to /root/cvim_sds-<target-tag>
# ./sds_runner/runner.py
```

The command validates the Cisco VIM Software Hub node hardware, the contents of the `sds_setup_data.yaml` file, and the validity of the TLS certificate, and then obtains the artifacts from the external Cisco VIM release registry and populates the Cisco VIM Software Hub server.

Installing Cisco VIM Software Hub in Air-Gapped Mode

Cisco VIM Software Hub is installed in the air-gapped mode when the Cisco VIM Software Hub server in the datacenter does not have internet connectivity. You can use the USB drive to load the installation files on the Cisco VIM Software Hub node. The installation files are over 25 GB in size. Downloading them to the USB drive may take several hours depending on the speed of your internet connection.

Before you begin

- Ensure that you have set up a CentOS 7 staging server (VM, laptop, or UCS server) with a 64 GB USB 2.0 drive.
- Ensure that you have internet, preferably a wired connection, to download the Cisco VIM installation files, which you want to load onto the USB drive.
- Ensure that you have disabled the CentOS sleep mode.

Step 1

On the staging server, use yum to install PyYAML and the python-requests package.

Step 2

Access the Cisco VIM software download web site using a web browser.

Step 3

Log in with the credentials provided by your account representative and download the `getartifacts.py` script from the external registry.

```
# download the new getartifacts.py file
curl -o getartifacts.py
https://username:password@cvim-registry.com/mercury-releases/cvim24-rhel7-osp13/releases/<3.0.x>/getartifacts.py

curl -o getartifacts.py-checksum.txt
https://username:password@cvim-registry.com/mercury-releases/cvim24-rhel7-osp13/releases/<3.0.x>/getartifacts.py-checksum.txt

# calculate the checksum by executing "sha512sum getartifacts.py", and verify that the output is
# same as that listed in getartifacts.py-checksum.txt
# Change the permission of getartificats.py via "chmod +x getartifacts.py"
```

Step 4

Run the `getartifacts.py` script.

The script formats the USB 2.0 drive (or USB 3.0 drive for M5-based management node) and downloads the installation files. You must provide the registry username and password, tag ID, and USB partition on the staging server.

```
getartifacts.py [-h] -t TAG -u USERNAME -p PASSWORD -d DRIVE
[--proxy PROXY] [--retry]
[--artifacts [ARTIFACTS [ARTIFACTS ...]]]
Script to pull container images en masse.
optional arguments:
-h, --help show this help message and exit
-t TAG, --tag TAG installer version to pull
-u USERNAME, --username USERNAME
Registry username
-p PASSWORD, --password PASSWORD
Registry password
-d DRIVE, --drive DRIVE
Provide usb drive path
--proxy PROXY https_proxy if needed
--retry Try to complete a previous fetch
--artifacts [ARTIFACTS [ARTIFACTS ...]]
Only supported parameter is all and defaults to all if nothing is passed
```

The `getartifacts.py` script gets the images from the remote registry and copies the contents to the USB drive.

Step 5

To identify the USB drive, execute the `lsblk` command before and after inserting the USB drive.

The command displays a list of available block devices. You can use the output data to find the location of the USB drive. You must provide the entire drive path in the `-d` option instead of any partition.

For example: `sudo ./getartifacts.py -t <tag_id> -u <username> -p <password> -d </dev/sdc> --artifacts all [--proxy proxy.example.com]`

For Cisco VIM Software Hub disconnected installation, you must use the `--artifacts all` options. These options enable you to save all the artifacts in the USB device, which is useful to create a replica of the Cisco VIM external releases.

Step 6

Verify the integrity of the downloaded artifacts and container images.

```
# create a directory sudo mkdir -p /mnt/Cisco
# /dev/sdc is the USB drive, same as supplied in getartifacts.py python script sudo mount /dev/sdc1
# /mnt/Cisco
cd /mnt/Cisco
# execute the test-usb help to look at the options
./test-usb -h
usage: ./test-usb
[-h] -- Show this program to check integrity of artifacts in this USB drive
[-a] -- Check integrity of all (core and insight) artifacts in this USB drive
[-l] -- Location of artifacts
# execute the verification script
./test-usb
# failures will be explicitly displayed on screen, sample success output below
# sample output of ./test-usb execution with 3.0.0 release
#./test-usb
INFO: Checking the integrity of this USB drive
INFO: Checking artifact buildnode-K9.iso
INFO: Checking artifact registry-3.0.0.tar.gz
INFO: Checking the integrity of this USB drive
INFO: Checking artifact buildnode-K9.iso
INFO: Checking artifact registry-3.0.0.tar.gz
INFO: Checking artifact mariadb-app-K9.tar.gz
INFO: Checking artifact haproxy-K9.tar.gz
INFO: Checking artifact insight-K9.tar.gz
Node
INFO: Checking required layers:
INFO: 548 layer files passed checksum.
If a failure occurs, an error message is displayed. For example:
# ./test-usb
INFO: Checking the integrity of this USB drive
INFO: Checking artifact buildnode-K9.iso
ERROR: Checksum for artifact buildnode-K9.iso does not match ('SHA512 (buildnode-K9.iso) =
96ec62a0932a0d69daf60acc6b8af2dc4e5eca132cd3781fc17a494592feb52a7f171eda25e59c0d326fb09194eeda66036dbc3870dafe74f59cf1f2dce225'
!= 'SHA512 (buildnode-K9.iso) =
a6a9e79fa08254e720a80868555679baeea2dd8f26a0360ad47540eda831617bea0514a117b12ee5f36415b7540afa112a1c904cd69e40d704a8f25d78867acf')
INFO: Checking artifact registry-3.0.0.tar.gz
ERROR: Artifact registry-3.0.0.tar.gz is not present INFO: Checking required layers:
ERROR: Layer file sha256:002aa1f0fbdae7ea25da1d906e732fe9a9b7458d45f8ef7216d1b4314e05207 has a bad
checksum
ERROR: Layer file sha256:5be3293a81773938cdb18f7174bf595fe7323fdc018c715914ad41434d995799 has a bad
checksum
ERROR: Layer file sha256:8009d9e798d9acea2d5a3005be39bcbfe77b9a928e8d6c84374768ed19c97059 has a bad
checksum
ERROR: Layer file sha256:ea55b2fc29b95d835d16d7eeac42fa82f17e985161ca94a0f61846defff1a9c8 has a bad
checksum
INFO: 544 layer files passed checksum.
```

Step 7

To resolve failure in downloading artifacts, unmount the USB and run the getartifacts command again with the **--retry** option.

```
sudo ./getartifacts.py -t <tag_id> -u <username> -p <password> -d </dev/sdc> --retry
```

Step 8

Mount the USB and then run the **test-usb** command to validate if all the files are downloaded.

```
# /dev/sdc is the USB drive, same as supplied in get artifacts.py python script
sudo mount /dev/sdc1 /mnt/Cisco
cd /mnt/Cisco
```

Execute the verification script.

Installing Pod from Cisco VIM Software Hub Server

```
# ./test-usb
# In case of failures the output of the command displays a message indicating the same on the screen.
```

- Step 9** When the USB integrity test completes, unmount the USB.

```
sudo umount /mnt/Cisco
```

- Step 10** After the artifacts of a target release are saved on the USB, you must unplug the USB from the staging server, connect it to the Cisco VIM Software Hub server, and then perform the following steps on the Cisco VIM Software Hub server:

- Provision your Cisco VIM Software Hub server with the buildnode ISO of that release and then connect the USB to the Cisco VIM Software Hub server.
- To copy the contents of the USB to the Cisco VIM Software Hub server, navigate to the `/root/cvim_sds-<tag>` directory, and then execute the import artifacts command.
- Place a valid TLS certificate in `/root/cvim_sds-<tag>/openstack-configs` directory.
- Configure the Cisco VIM Software Hub setup data file with all the fields and placed the file in the `/root/cvim_sds-<tag>/openstack-configs` directory.
- Install the release on the Cisco VIM Software Hub server.

Navigate to the `cvim_sds` directory on the Cisco VIM Software Hub server and execute the following command:

```
# cd /root/cvim_sds-<tag>
# ./sds_runner/runner.py
Usage: runner.py [options]
Runner
Options:
-h, --help show this help message and exit
-l, --list_steps List steps
-s SKIP_STEPS, --skip_steps=SKIP_STEPS
Comma separated list of steps to skip. eg -s 2,3
-p PERFORM_STEPS, --perform=PERFORM_STEPS
-y, --yes Yes option to skip steps without prompt
```

Installing Pod from Cisco VIM Software Hub Server

When you want to install a Cisco VIM pod using the artifacts obtained from the Cisco VIM Software Hub server, you need to provide an additional parameter in `setup_data.yaml`. Ensure that the release artifacts are pre-installed on the Cisco VIM Software Hub server and that the `setup_data.yaml` file is populated with the pod details. Provide the registry FQDN name for install through Cisco VIM Software Hub. For example, `your.domain.com`.

```
REGISTRY_NAME: '<registry_name>' # Mandatory Parameter.
```

Cisco VIM pod `setup_data.yaml` require the `REGISTRY_USERNAME` and `REGISTRY_PASSWORD` to connect to the docker registry and fetch docker images. To fetch the docker images from Cisco VIM Software Hub node, provide the user credentials available in the `SDS_READ_ONLY_USERS` section of `sds_setup_data.yaml`. The details of an admin user with read/write access to docker registry are provided in `SDS_REGISTRY_USERNAME` and `SDS_REGISTRY_PASSWORD` field. So, it is recommended to have a read-only user on Cisco VIM pod.



Note The Cisco VIM management node must have connectivity to the organization DNS server to resolve the Cisco VIM Software Hub server domain.

Day 2 Operations on Cisco VIM Software Hub

The following Day-2 operations are supported on the Cisco VIM Software Hub server:

- Reconfigure Cisco VIM Software Hub TLS certificate and Cisco VIM Software Hub registry credentials
- Cisco VIM Software Hub server Backup and Restore
- Registry Cleanup Script
- Manual update of few packages in the **Maintenance** window

For more information on these topics, refer to the *Cisco Virtual Infrastructure Manager Administrator Guide*.

Setting Up the UCS C-Series Pod

After you install the RHEL OS on the management node, perform the following steps to set up the Cisco UCS C-Series servers:

Step 1 Log into CIMC GUI of Cisco NFVI management node.

Step 2 Follow steps in [Configuring the Server Boot Order](#) to set the boot order to boot from Local HDD

Step 3 Follow steps in [Configure BIOS Parameters](#) to set the LOM, HBA, and PCIe slots to the following settings:

For servers based on UCS M4 boxes, set the following for BIOS Parameters:

- CDN Support for VIC—Disabled
- PCI ROM CLP—Disabled
- PCH SATA Mode—AHCI
- All Onboard LOM Ports—Enabled
- LOM Port 1 OptionROM—Disabled
- LOM Port 2 OptionROM—Disabled
- All PCIe Slots OptionROM—Enabled
- PCIe Slot:1 OptionROM—Enabled
- PCIe Slot:2 OptionROM—Enabled
- PCIe Slot: MLOM OptionROM—Enabled
- PCIe Slot:HBA OptionROM—Enabled
- PCIe Slot:N1 OptionROM—Enabled

Setting Up the UCS C-Series Pod

- PCIe Slot:N2 OptionROM—Enabled
- PCIe Slot:HBA Link Speed—GEN3

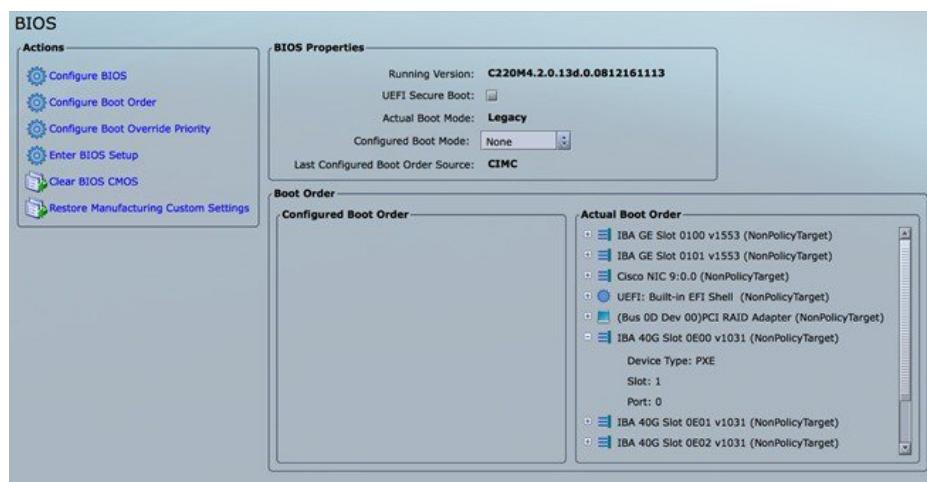
For servers based on UCS M5 boxes, set the following for BIOS Parameters:

- All Onboard LOM Ports—Enabled
- LOM Port 1 OptionROM—Disabled
- LOM Port 2 OptionROM—Disabled
- PCIe Slot:1 OptionROM—Enabled
- PCIe Slot:2 OptionROM—Enabled
- MLOM OptionROM—Enabled
- MRAID OptionROM—Enabled

Other parameters must be set to their default values.

To setup C-series pod with Intel 710 NIC:

- Each C-series server must have two 4-port Intel 710 NIC cards.
- Ports A, B, and C for each Intel 710 NIC card are connected to the respective ToR.
- PCI slot in which the Intel NIC cards are inserted are enabled in the BIOS setting (BIOS > Configure BIOS >Advanced > LOM and PCI Slot Configuration -> All PCIe Slots OptionROM-Enabled and enable respective slots).
- Slots are identified by checking the slot-id information under the **Network-Adapter** tab listed under the Inventory link on the **CIMC** pane.
- All the Intel NIC ports must be indicated in the BIOS summary page under the **Actual Boot Order** pane, as IBA 40G Slot xyza with Device Type is set to PXE.



For UCS M5 look for “IBA 40G Slot ...” under the BIOS Properties

The screenshot shows the 'BIOS Properties' section of a management interface. It includes fields for 'Running Version' (C240M5.3.1.2a.0.0918172340), 'UEFI Secure Boot' (unchecked), 'Actual Boot Mode' (Legacy), 'Configured Boot Mode' (Legacy), 'Last Configured Boot Order Source' (CIMC), and 'Configured One time boot device' (empty). A 'Save Changes' button is at the bottom left. Below this, a 'Configured Boot Devices' list shows 'Basic' and 'Advanced' sections with options like 'localhdd', 'pxeboot', and 'vmmedia'. To the right, an 'Actual Boot Devices' list shows hardware components: 'IBA XE Slot 0100 v2372 (pxeboot)', 'IBA XE Slot 0101 v2372 (pxeboot)', 'IBA 40G Slot 5E00 v1047 (pxeboot)', '(Bus 16 Dev 00)PCI RAID Adapter (localhdd)', and 'UEFI: Built-in EFI Shell (NonPolicyTarget)'.

If the boot order for the Intel NICs is not listed as above, enable the PXE boot setting for each UCS-C series server by using either Intel's BootUtil tool on a pre-installed Linux system or boot a special ISO image. This is time consuming especially on a large POD with many nodes. Hence, an automated tool has been developed to help with this painstaking process.

While the pxe-boot tool simplifies the job of flashing the intel NIC cards, the restrictions of COSI compliance prevents us from shipping third-party utility. Administrators must download the PREBOOT.exe file from Intel website:

<https://downloadcenter.intel.com/download/27539/>

[Ethernet-Intel-Ethernet-Connections-Boot-Utility-Preboot-Images-and-EFI-Drivers](#)

Version: 22.10

Date: 12/7/2017

OS Independent

Language: English

Size: 16.54 MB

MD5: ace485e8a3ef9039212f52b636ce48e3

PREBOOT.EXE

Ensure that there is unrestricted network access from Cisco VIM Management node to UCS-C series server's CIMC over following ports:

- TCP/2400 - serial-over-lan (SOL)
- TCP/22 - XMLAPI

Ensure that there is unrestricted network access from UCS-C series server's CIMC to Cisco VIM Management node's API interface over following port:

TCP/80 - HTTP

This utility updates only the Intel PXE configuration and not the card's firmware or Option ROM.

Utility Details

Two scripts available in the Cisco VIM Installer's tools directory are:

- create-bootutil-img.sh
- intel-bootutil-update.py

Usage

```
[root@cologne-mgmt tools]# ./create-bootutil-img.sh
```

Usage: ./create-bootutil-img.sh <PREBOOT.exe file> <output image name>

You can download PREBOOT.exe file from :

<https://downloadcenter.intel.com/download/27862/Ethernet-Intel-Ethernet-Connections-Boot-Utility-Preboot-Images-and-EFI-Drivers>

Version: 22.10

Date: 12/7/2017

OS Independent

Language: English

Size: 16.54 MB

MD5: ace485e8a3ef9039212f52b636ce48e3

PREBOOT.EXE

To toggle Intel PXE configuration on UCS C-series, use the script below:

```
[root@cologne-mgmt tools]# ./intel-bootutil-update.py -h
usage: intel-bootutil-update.py [-h] [--hosts HOSTS]
[--exclude-hosts EXCLUDE_HOSTS] [-v] [-y]
--setupfile SETUPFILE --bootutil-image
BOOTUTIL_IMAGE --port {0,1,2,3} --state
{enable,disable}
```

Optional arguments:

- h --help show this help message and exit
- hosts HOSTS comma separated list of servers
- setup_data.yaml file target for PXE configuration
- exclude-hosts EXCLUDE_HOSTS comma separated list of servers
- setup_data.yaml file to exclude for PXE configuration
- v, --verbose enable verbose output
- y, --yes skip prompt

Required arguments:

- setupfile SETUPFILE setup_data.yaml file location
- bootutil-image BOOTUTIL_IMAGE BootUtil image location
- port {0,1,2,3} port #, multiple entries allowed

--state {enable,disable} enable or disable PXE configuration

Example to enable all port A:

```
./intel-bootutil-update.py --setupfile /root/openstack-configs/setup_data.yaml
--bootutil-image /root/bootutil.img --port 0 --state enable
:
```

Example to enable all port A and B:

```
./intel-bootutil-update.py --setupfile /root/openstack-configs/setup_data.yaml
--bootutil-image /root/bootutil.img --port 0 --port 1 --state enable
```

Example to disable all port C:

```
./intel-bootutil-update.py --setupfile /root/openstack-configs/setup_data.yaml
--bootutil-image /root/bootutil.img --port 2 --state disable
```

Flow:

Multiple scripts are required as Intel's PREBOOT.exe utility is not packaged with Cisco VIM for COSI compliance:

1. Download PREBOOT.exe version 23.1 from Intel's website.
2. Go to Cisco VIM Installer's tools directory.
3. Run 'create-bootutil.img' script to create a CIMC-KVM mountable USB image.
4. Run 'intel-bootutil-update.py' script, to configure Intel NIC for enabling or disabling PXE.

Utility in action examples:

```
[root@cologne-mgmt installer]# cd tools
[root@cologne-mgmt tools]#
[root@cologne-mgmt tools]# ./create-bootutil-img.sh

Usage: ./create-bootutil-img.sh <PREBOOT.exe file> <output image name>

You can download PREBOOT.exe file from Intel:https://downloadcenter.intel.com/download/27862/Ethernet-Intel-Ethernet-Connections-Boot-Utility-Preboot-Images-and-EFI-Drivers
```

Version: 23.1

Date: 2/21/2018

OS Independent

Language: English

Size: 16.54 MB

MD5: dadd5c85777164d8476670774b4459fc

PREBOOT.EXE

Setting Up the UCS B-Series Pod

```
[root@cologne-mgmt tools]#
[root@cologne-mgmt tools]# ./create-bootutil-img.sh /root/PREBOOT.exe /root/bootutil.img
...
Unmounting temporary mount point /tmp/tmp_bootutil.img
Cleaning up temporary workspaces
Successfully created image file with BOOTUTIL64E.EFI
-rw-r--r--. 1 root root 5.0M Jul 20 17:52 /root/bootutil.img

[root@cologne-mgmt tools]#
[root@cologne-mgmt tools]# ./intel-bootutil-update.py --setupfile
/root/openstack-configs/setup_data.yaml --bootutil-image /root/bootutil.img --port 0 --state
enable

All servers will be rebooted as part of PXE configuration, would you like to continue? <y|n>
y
2018-07-18 18:34:36,697 INFO Enabling temporary HTTP server hosting BootUtil.img on
172.29.86.10
2018-07-18 18:34:36,790 INFO Successfully enabled temporary HTTP server hosting BootUtil.img
on 172.29.86.10
...
2018-07-18 18:40:28,711 INFO Disabling temporary HTTP server hosting BootUtil.img on
172.29.86.10
2018-07-18 18:40:28,810 INFO Successfully disabled temporary HTTP server hosting BootUtil.img
on 172.29.86.10
Server(s) successfully updated PXE configuration:
cologne-control-1,cologne-control-3,cologne-control-2,cologne-compute-1,cologne-compute-2,cologne-storage-1,cologne-storage-3,cologne-storage-2
[root@cologne-mgmt tools]#
```

Setting Up the UCS B-Series Pod

After you install the RHEL OS on the management node, complete the following steps to configure a Cisco NFVI B-Series pod:

Step 1 Log in to Cisco UCS Manager, connect to the console of both fabrics and execute the following commands:

```
# connect local-mgmt
# erase config
All UCS configurations are erased and system starts to reboot. Are you sure? (yes/no): yes
Removing all the configuration. Please wait....
```

Step 2 Go through the management connection and clustering wizards to configure Fabric A and Fabric B:

Fabric Interconnect A

```
# connect local-mgmt
# erase config
Enter the configuration method. (console/gui) console
Enter the setup mode; setup newly or restore from backup. (setup/restore) ? setup
You have chosen to setup a new Fabric interconnect. Continue? (y/n): y
Enforce strong password? (y/n) [y]: n
Enter the password for "admin":
Confirm the password for "admin":
Is this Fabric interconnect part of a cluster(select 'no' for standalone)? (yes/no) [n]: yes
Enter the switch fabric (A/B) []: A
Enter the system name: skull-fabric
Physical Switch Mgmt0 IPv4 address : 10.30.119.58
Physical Switch Mgmt0 IPv4 netmask : 255.255.255.0
IPv4 address of the default gateway : 10.30.119.1
Cluster IPv4 address : 10.30.119.60
```

```

Configure the DNS Server IPv4 address? (yes/no) [n]: y
DNS IPv4 address : 172.29.74.154
Configure the default domain name? (yes/no) [n]: y
Default domain name : ctocllab.cisco.com

Join centralized management environment (UCS Central)? (yes/no) [n]: n

Following configurations are applied:
Switch Fabric=A
System Name=skull-fabric
Enforced Strong Password=no
Physical Switch Mgmt0 IP Address=10.30.119.58
Physical Switch Mgmt0 IP Netmask=255.255.255.0
Default Gateway=10.30.119.1
DNS Server=172.29.74.154
Domain Name=ctocllab.cisco.com
Cluster Enabled=yes
Cluster IP Address=10.30.119.60
NOTE: Cluster IP is configured only after both Fabric Interconnects are initialized

Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes
Applying configuration. Please wait..

```

Fabric Interconnect B

Enter the configuration method. (console/gui) ? **console**

```

Installer has detected the presence of a peer Fabric interconnect. This Fabric interconnect is added
to the cluster. Continue (y/n) ? y

Enter the admin password of the peer Fabric interconnect:
Connecting to peer Fabric interconnect... done
Retrieving config from peer Fabric interconnect... done
Peer Fabric interconnect Mgmt0 IP Address: 10.30.119.58
Peer Fabric interconnect Mgmt0 IP Netmask: 255.255.255.0
Cluster IP address : 10.30.119.60
Physical Switch Mgmt0 IPv4 address : 10.30.119.59
Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes
Applying configuration. Please wait.

```

Step 3

Configure the NTP:

- In UCS Manager navigation area, click the **Admin** tab.
- In the Filter drop-down list, choose **Time Zone Management**.
- In the main window under Actions, click **Add NTP Server**.
- In the Add NTP Server dialog box, enter the NTP hostname or IP address, then click **OK**.

Step 4

Following instructions in [Cisco UCS Manager GUI Configuration Guide, Release 2.4](#), "Configuring Server Ports with the Internal Fabric Manager" section, configure the Fabric Interconnect A and Fabric Interconnect B uplinks to the Cisco NFVI top of rack (ToR) switches as **Uplink Ports**, **Server Ports**, and **Port Channels**.

Step 5

Configure the downlinks to the B-Series server chassis as **Server Ports**.

Step 6

Acknowledge all chassis.

Configuring the Out-of-Band Management Switch

For Cisco VIM installer API and SSH bonded interface, use 1-GB Intel NICs that connect the Cisco NFVI management node and Cisco Catalyst switch. Following is a sample configuration for creating a port channel on a Catalyst switch. Modify the configuration for your environment:

```
interface GigabitEthernet0/39
  channel-group 2 mode active
  speed 1000

interface GigabitEthernet0/40
  channel-group 2 mode active
  speed 1000

interface Port-channel2
  switchport access vlan 165
  switchport mode access
```

Support of 3rd Party Compute (HP DL 360 Gen9)

Before you begin

Cisco VIM manages all aspects of the cloud through full automation, with no manual intervention beyond initial infrastructure setup. To extend this approach to third-party computes, specifically HP DL360 Gen9, distribute the HP SmartArray Utility Tools as part of the platform offering.

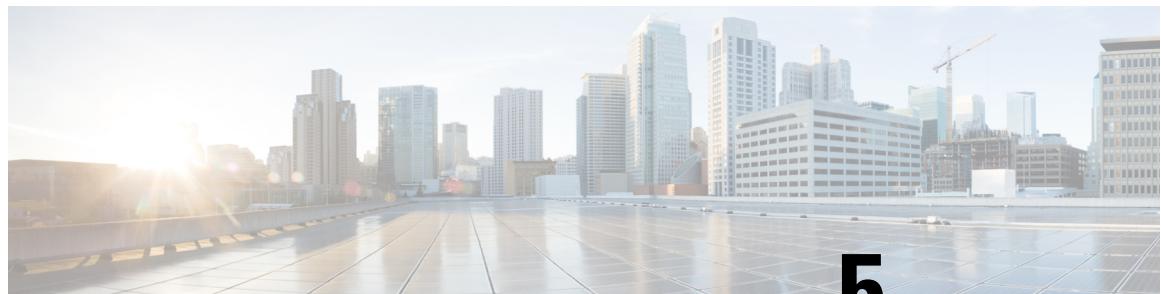
To support third-party computes in Cisco VIM perform the following steps:

- Step 1** Download the **ssaclı** tool directly from HPE's website and place the RPM file in "/root/installer-<tagid>/openstack-configs/" directory.

Note Currently Cisco VIM supports ssaclı-3.10-3.0.x86_64.rpm.

- Step 2** Location and checksum of the target RPM is:

https://downloads.linux.hpe.com/SDR/repo/spp-gen9/RHEL/7/x86_64/2017.07.1/ssaclı-3.10-3.0.x86_64.rpm SHA1
checksum: 51ef08cd972c8e65b6f904fd683bed8e40fce377



CHAPTER 5

Installing Cisco VTS

If your Cisco NFVI package includes Cisco Virtual Topology System (VTS), refer this section on how to install Cisco VTS for use with Cisco NFVI. The Cisco VTS installation procedures are customized for Cisco NFVI from the standard Cisco VTS 2.6.2 installation procedures located on the [Cisco VTS product site](#). You must install Cisco VTS before you install Cisco VIM.

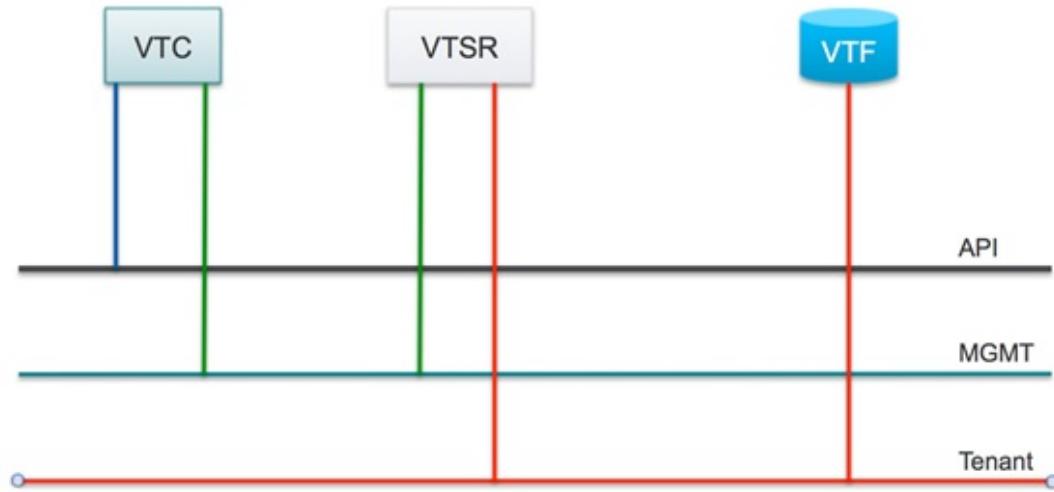
- [Overview to Cisco VTS Installation in Cisco NFVI, on page 111](#)
- [System Requirements for VTC VM, on page 116](#)
- [System Requirements for VTSR VM, on page 117](#)
- [Supported Virtual Machine Managers, on page 117](#)
- [Supported Platforms, on page 117](#)
- [Installing Cisco VTS in Cisco NFVI Environment, on page 119](#)
- [Installing the VTSR VMs, on page 123](#)
- [Verifying Cisco VTS Installation in Cisco NFVI, on page 126](#)
- [Configuring Cisco VTS and VTSR After Installation, on page 128](#)
- [Installing VTS in an HA Configuration, on page 129](#)
- [Sample Cisco VTS Configurations for Cisco NFVI, on page 133](#)

Overview to Cisco VTS Installation in Cisco NFVI

The Cisco Virtual Topology System is an overlay management and provisioning system for data center networks. It automates data center overlay fabric provisioning for both physical and virtual workloads. It provides a policy-based approach for overlay provisioning, and can be used for multitenant data centers for cloud services, including Cisco NFVI.

To install Cisco VTS with Cisco NFVI, you must manually install the Cisco VTS Virtual Topology Controller (VTC) and its VTSR VMs before you start the Cisco VIM installation. The VTC and VTSR VMs must be run on an independent pair of servers, and not on a Cisco NFVI control, compute, storage, or management node. You can set up the networking on those servers as described in the installation procedures. When you run the Cisco VIM installer, you have to provide the VTC VIP and appropriate VTS credentials.

The following figure shows how Cisco VTS Virtual Topology Controller (VTC) and VTSR virtual machines (VMs) connect to the Cisco NFVI networks.

Figure 38: Cisco VTS Connectivity to Cisco NFVI

The following table maps Cisco VTS network names to Cisco VIM network names.

Table 16: Cisco VTS to Cisco VIM Network Name Mapping

Cisco VTS VM	Cisco VTS Network Name	Cisco VIM Network Name
VTC	Management Network	API (a)
VTC	Underlay Network	Management or Provision (mx)
VTSR	Management Network	Management or Provision (mx)
VTSR	Underlay Network	Tenant (t)

The following table describes the required IP address allocations for VTS components.

Table 17: Cisco VTS IP Address Allocations

Cisco VIM Network	Required Cisco VTS IP Addresses	Description
API (a)	3 total (1 VIP + 1 IP per VTC VM)	Set up in the VTC config.iso and cluster.conf
Management or Provisioning (mx)	<ul style="list-style-type: none"> • 5 total—Three for VTC (one VTC VIP called as VTS_NCS_IP in setup_data and one IP per VTC VM) • Two for VTSR: one IP per VTSR VM. 	Set up in VTSR config.iso. Note: VTS component IP addresses cannot overlap with the pool ranges configured in the Cisco VIM setup_data.yaml.

Cisco VIM Network	Required Cisco VTS IP Addresses	Description
Tenant (t)	2 total—(one IP address VTSR VM.)	<p>Set up in VTSR config.iso</p> <p>Note: The VTS component IPs cannot overlap with pool ranges that are configured in the Cisco VIM setup_data.yaml.</p>

The following is the VTS IP distribution and setup mechanism.

VIM API network

- VTC1—api (a) network IP1 (associated through the VTC1 config ISO)
- VTC2—api (a) network IP2 (associated through the VTC2 config ISO)
- VTC VIP—api (a) network IP3 (associated through the HA step cluster.conf)

VIM Management/Provisioning network

- VTC1—management/provisioning (mx) network IP1 (associated through the VTC1 config ISO)
- VTC2—management/provisioning (mx) network IP2 (associated through the VTC2 config ISO)
- VTC VIP—management/provisioning (mx) network IP3 (associated through the HA step cluster.conf)
- VTSR 1—management/provisioning (mx) network IP4 (associated through the VTSR-1 config ISO)
- VTSR 2—management/provisioning (mx) network IP5 (associated through the VTSR-2 config ISO)

VIM Tenant network:

- VTSR 1—tenant (t) network IP1 (associated through the VTSR-1 config ISO)
- VTSR 2—tenant (t) network IP2 (associated through the VTSR-2 config ISO)

Cisco VTS Usernames and Passwords in Cisco NFVI

The following table lists the Cisco VTS usernames and passwords that are deployed after you install Cisco VTS in Cisco NFVI.

Table 18: Cisco VTS Usernames and Passwords in Cisco NFVI

Configuration Location	Value Requirements	Description/Comments
CVIM: openstack-configs/setup_data.yaml VTS_PARAMETERS: VTS_USERNAME VTS_PASSWORD VTS_SITE_UUID The following parameters are optional, only required if VTS_DAY0 is enabled. VTC_SSH_PASSWORD VTC_SSH_USERNAME VTS_SITE_UUID Optional: MANAGED	VTS_USERNAME must be admin. VTS_PASSWORD must match VTC UI login password for the admin user. Password must have a minimum of 8 characters and at least one uppercase letter, one digit, and one special character. VTS_SITE_UUID is unique UUID of VTS SITE controlled by Cisco VIM. The VTS_SITE_UUID must be in a generic UUID format (Unique Pod UUID to indicate which pod the VTS is controlling) The VTC_SSH_PASSWORD and VTC_SSH_USERNAME are ssh credentials to login to VTC VMs. MANAGED is either True or False. By default, it is false. If it is True, VTS deployment mode is managed.	Used by VTF to register with the VTC / VTSR.
VTC ISO config.txt : vts-adminPassword AdministrativeUser AdministrativePassword	Must match the Cisco VIM setup_data.yaml VTC_SSH_PASSWORD parameter. AdministrativeUser must match with setup_data.yaml VTC_SSH_USERNAME parameter AdministrativePassword matches with VTC_SSH_PASSWORD parameter.	Configures VTC admin user's initial password. SSH username/password for VTC VM.
VTSR ISO: USERNAME PASSWORD		VTSR VM SSH username/password The VTSR adds this in VTS Inventory > Authorization Group > vtsgroup Device User Name associated with VTC admin user

Modes of TOR Configuration with VTS

Cisco VTS supports two modes of TOR configuration:

- Unmanaged TOR: It is the default mode of operation for VTS with Cisco VIM. VTS network inventory is added as “Unmanaged” device instead of actual TOR switches. BGP EVPN ingress replication mode mechanism is used for admin domain, but the port configuration does not push configuration to the TOR switches.
- Managed TOR: VTS network inventory is added with actual TOR switches. Control and compute nodes information are added with their corresponding interfaces connected with TOR in the VTS host inventory. BGP EVPN multicast replication mode is used for admin domain, while the port configuration enables multicast Internet Group Management Protocol (IGMP) snooping and PIM configuration for Tenant VLAN on actual TOR switches.



Note As the storage nodes do not have VTF, the switch ports hanging off the storage nodes are configured statically.

To maintain consistency, add the `tor_info` to the storage nodes in the `setup_data` of the pod. .

Listed below is the snippet of the Multicast configuration push to Cisco Nexus 9000, when port is configured with Tenant VLAN ID 111.

```
interface Vlan111
no shutdown
no ip redirects
ip address 22.22.22.200/24
no ipv6 redirects
ip router ospf 100 area 0.0.0.0
ip pim sparse-mode
ip igmp version 3
ip igmp static-oif 239.0.0.1
hsrp 22
ip 22.22.22.1
vlan configuration 111
ip igmp snooping static-group 239.0.0.1 interface port-channel12
ip igmp snooping static-group 239.0.0.1 interface port-channel13
ip igmp snooping static-group 239.0.0.1 interface port-channel14
```



Note Due to limitation of VTS, Tenant VLAN ID needs to be selected as lowest number in the TOR interface. If not, Multicast configuration will be pushed incorrectly.

The following table lists the configurations required to enable the functionality of TORs “managed” through VTS.

Table 19: Cisco VTS Parameters for TORs managed through VTS

Configuration Location	Value Requirements	Description
CVIMmercury: openstack-configs/setup_data.yaml VTS_PARAMETERS: MANAGED:	MANAGED: Set to True or False. By default, it is False.	MANAGED: Must be configured as True, when VTS deployment mode is managed. It is a day-0 configuration, and cannot be enabled as a reconfigure option.

Configuration Location	Value Requirements	Description
TORSWITCHINFO: CONFIGURE_TORS	CONFIGURE_TORS: False	CONFIGURE_TORS value has been set to False to indicate that CVIM is not configuring the TORs; this is a way for VTC to know what switches to access and manage.
SWITCHDETAILS:	Hostname, ssh_ip, username, and password of the switches for VTC to manage {switch_a_hostname: ethx/y, switch_b_hostname: ethx/y}	Need minimum switch details to access it.
SERVERS: <SERVER_NAME>: tor_info:		For each server, list the tor_info associated to the server, so that VTC can manage the switch ports. Note that the storage nodes do not have VTF and hence switch ports hanging off the storage nodes are configured statically. To maintain consistency, add the tor_info to the storage nodes in the setup_data of the pod.

From an architecture point of view, the following are configured automatically in VTC Node when Managed TOR mode is selected in setup_data.yaml:

- VTS System Settings and Route reflector are configured in VTC.
- Openstack Virtual Machine Manager is configured.
- Global VNI POOL is configured.
- Multicast pools are created to allocate multicast IP address for Tenant VLAN ID.
- Authentication Group is created for device.
- TOR switches are configured under Network Inventory.
- Admin domain is created with BGP EVPN multicast replication mode for L2 and L3 Gateway.
- TOR switches and VTSR are added to L2 and L3 Gateway in admin domain.
- Controller and Compute Node are added under host inventory with corresponding TOR interfaces.
- All VTFs are registered with VTSRs and appear under Virtual Forwarding Groups.

System Requirements for VTC VM

The following table provides information about the minimum system requirements for the VTC virtual machine:

Requirement	Details
Disk space	48 GB

Requirement	Details
CPU	8
Memory	32 GB
Computing host	Certified with Cisco UCS B-series, Cisco UCS C-series Rack Servers

System Requirements for VTSR VM

The following table gives details about the minimum system requirements for the VTSR virtual machine:


Note

The VTSR VM serves two purposes. It is required to enable VTS High Availability. It also acts as the control plane for the VTF. You need to install VTSR only if you consider enabling High Availability or if you plan to have a VTF in your set up.

Requirement	Details
Disk Space	Primary disk must be 77 GB.
CPUs	14
Memory	48 GB RAM
Computing Host	Certified with Cisco UCS B-series, Cisco UCS C-series Rack Servers

Supported Virtual Machine Managers

You can install Cisco VTS on the following supported versions of Virtual Machine manager (VMM):

Table 20: Openstack Versions

	OpenStack Liberty	OpenStack Newton/Queens
On RHEL	12.0.0; 12.0.1; 12.0.2; 12.0.3; 12.0.4; 12.0.5; 12.0.6	14.0.3 On CentOS
On CentOS	12.0.0; 12.0.1; 12.0.2	N/A

Supported Platforms

The following tables provide information about the Cisco VTS supported platforms and their role.

**Note**

VTS supports VXLAN overlays using the BGP EVPN control plane.

Role	Platform Supported
Top-of-rack (ToR) leaf switch	<ul style="list-style-type: none"> • Cisco Nexus 9300TX and 9300PX platform switches • Cisco Nexus 9332PQ and 93128TX switches • Cisco Nexus 9200 platform switches • Cisco Nexus 9500 platform switches
Data center spine	<ul style="list-style-type: none"> • Cisco Nexus 9300TX and 9300PX platform switches • Cisco Nexus 9500 platform switches • Cisco Nexus 9200 platform switches
Border leaf	<ul style="list-style-type: none"> • Cisco Nexus 9300TX and 9300PX platform switches • Cisco Nexus 9500 platform switches • Cisco Nexus 9200 platform switches
Data center interconnect (DCI)	<ul style="list-style-type: none"> • Cisco ASR 9000 Series Aggregation Services routers • Cisco Nexus 9300 platform switches
Virtual machine manager (VMM)	OpenStack Queens on RHEL versions
Hypervisor	<ul style="list-style-type: none"> • Red Hat Enterprise Linux 7.3 with KVM • Red Hat Enterprise Linux 7.6 • CentOS
Virtual forwarders	Cisco Virtual Topology Forwarder (VTF)

The following table lists the software images supported for the different devices.

Table 21: Software Images Supported

Cisco Nexus 93xx	NX OS Release 7.0.3.I7.2 or 9.2(1)
Cisco Nexus 95xx	NX OS Release 7.0.3.I7.2 or 9.2(1)
Cisco ASR 9000	Cisco IOS XR Software Release 6.5.1.

The following table lists the VPC modes supported for different devices.

Note If Cisco Nexus 9000 series ToR is not configured with vPC related configuration, including peer-link, (also known as a multichassis etherChannel trunk (MCT)), you must not configure vpc on the ToR. This may bring loopback interface used for NVE to admin down state.

Table 22: VPC Modes Supported

Cisco Nexus 93xx	Server VPC
Cisco Nexus 95xx	Server VPC

Installing Cisco VTS in Cisco NFVI Environment

Installing Cisco VTS within Cisco NFVI involves installing the Virtual Topology Controller (VTC) VM. You can install the VTC VM using either the automatic or manual configuration option.

- To install the VTC VM using an ISO file (auto configuration), see [Installing VTC VM - Automatic Configuration Using ISO File, on page 119](#).
- To install the VTC VM using the virt-manager application (manual configuration), see [Installing VTC VM - Manual Configuration Using Virt-Manager, on page 120](#).
- To install the VTC VM using VNC (manual configuration), see [Installing VTC VM - Manual Configuration using VNC, on page 122](#)

Installing VTC VM - Automatic Configuration Using ISO File

To install a VTC VM and enable configuration using an ISO file, create a text file with the VM settings, wrap the text file in an ISO file, and then attach the ISO file to the VM CD drive.

-
- Step 1** Connect to the controller node via SSH, and copy the `vtc.qcow2` file to `/var/lib/libvirt/images/` folder.
- Step 2** Copy the `vtc.sample.xml` file to your controller. The [Installing Cisco VTS in Cisco NFVI Environment, on page 119](#) topic provides the file contents.
- Step 3** Create a `config.txt` file containing the following parameters:

```
Hostname=vtc
ManagementIPv4Method=Static
ManagementIPv4Address= <VM's a-net IP address in a.b.c.d form>
ManagementIPv4Netmask= <a-net IP mask in a.b.c.d form>
ManagementIPv4Gateway= <a-net gateway IP address in a.b.c.d form>
UnderlayIPv4Method=Static
UnderlayIPv4Address= <VM's mx-net IP address in a.b.c.d form>
UnderlayIPv4Netmask=<mx-net IP mask in a.b.c.d form>
DNSv4=<DNS server--ie. setup_data.yaml::NETWORKING['domain_name_servers'][0]>
Domain=<domain name--ie. setup_data.yaml::NETWORKING['domain_name']>
NTP=<NTP server--ie. setup_data.yaml::NETWORKING['ntp_servers'][0]>
vts-adminPassword=<password for user 'admin'--setup_data.yaml::VTS_PARAMETERS['VTC_SSH_PASSWORD']>
AdministrativeUser=<VM ssh login user--can be setup_data.yaml::VTS_PARAMETERS['VTC_SSH_USERNAME']>
AdministrativePassword=<VM ssh login user--can be setup_data.yaml::VTS_PARAMETERS['VTC_SSH_PASSWORD']>
ManagementIPv6Method: Unused by NFVI
```

Installing VTC VM - Manual Configuration Using Virt-Manager

UnderlayIPv6Method: Unused by NFVI

Note *config.txt* file must have a blank line at the end.

Note Before entering the VTS_PASSWORD, review [Cisco VTS Usernames and Passwords in Cisco NFVI](#), on page 113.

Parameter descriptions:

- Hostname—The VM hostname.
- ManagementPv4Method—Whether to use DHCP or static addressing for the Cisco NFVI API network (a-net) interface (eth0).
- ManagementIPv4Address—The api (a) network IPv4 address of the VM (required only for static addressing).
- ManagementIPv4Netmask—The a network IPv4 net mask of the VM (required only for static addressing).
- ManagementIPv4Gateway—The a network API IPv4 gateway of the VM (required only for static addressing).
- UnderlayIPv4Method—Whether to use DHCP or static addressing for the Cisco NFVI management/provisioning (mx) network interface (eth1).
- UnderlayIPv4Address—The mx network IPv4 address of the VM (required only for static addressing).
- UnderlayIPv4Netmask—The mx network IPv4 net mask of the VM (required only for static addressing).
- DNSv4—DNS IPv4 address (required only for static addressing).
- Domain—DNS search domain (required only for static addressing).
- NTPv4—NTP IPv4 address or FQDN (required only for static addressing).
- vts-admin Password—Password for the vts-admin user. This should match the value in setup_data.yaml::VTS_PARAMETERS['VTS_PASSWORD'] or subsequently changed through the VTC UI to match the value in setup_data.yaml::VTS_PARAMETERS['VTS_PASSWORD']
- Administrative User—New administrative user for login using SSH.
- Administrative Password—Sudo password for the administrative user.

Step 4 Use mkisofs to create an ISO file, for example:

```
mkisofs -o config.iso config.txt
```

Step 5 Create the VTC VM using following command:

```
virsh create vtc.sample.xml
```

Installing VTC VM - Manual Configuration Using Virt-Manager

To install VTC VM, configure it manually using the virt-manager application:

Step 1 Connect to the controller node through SSH, and copy the vtc.qcow2 file to /var/lib/libvirt/images/ folder.

Step 2 Copy the Cisco NFVI vtc.sample.xml file to your controller. Modify it as per your setup. See [Sample Cisco VTS Configurations for Cisco NFVI, on page 133](#) for examples.

Step 3 Create the VTC VM using following command:

```
virsh create vtc.sample.xml
```

Step 4 Run the command:

```
virsh list --all
```

It should display:

Id	Name	State
2	VTC	running

Step 5 Start virt-manager. Run:

```
virt-manager
```

Step 6 After the virt-manager window opens, click the VTC VM to open up the VTC VM console.

The console displays an installation wizard that takes you through the initial VTC VM configuration.

Step 7 Enter the following:

Note For items that take multiple values such as DNS and NTP, each value must be separated by a space.

- VTS Hostname
- DHCP / Static IP configuration for static IP
- Management IP address for VTC—This is the Cisco NFVI api (a) network IP address.
- Management IP Netmask (api network)
- Management Gateway address (api network)
- DNS Address—One of the DNS servers in setup_data.yaml::NETWORKING['domain_name_servers']
- DNS Search domain—setup_data.yaml::NETWORKING['domain_name']
- Underlay IP address—This is the IP address for Cisco NFVI management/provisioning (mx) network.
- Underlay IP Netmask (mx network)
- NTP address—One of the setup_data.yaml::NETWORKING['ntp_servers'] addresses
- Password change for user vts-admin—Enter the default user vts-admin password. The vts-admin user is used for password recovery and to revisit a configuration screen for editing the information. If you log in to the VTC VM using vts-admin username and password again, you get the same dialog to go through the VTC VM setup again. The password must match the value in setup_data.yaml::VTS_PARAMETERS['VTS_PASSWORD'] or subsequently changed through the VTC UI to match the value in setup_data.yaml::VTS_PARAMETERS['VTS_PASSWORD']

Before entering the VTS_PASSWORD, reviewing [Cisco VTS Usernames and Passwords in Cisco NFVI, on page 113](#) is recommended.

- Administrator User—Enter administrative username and password. This username and password are used to login to the VM via SSH.
- Password for administrator user

VTC VM reboots at this time. Wait for two minutes for the VTC VM to be up. You can ping the IP address given for VTC VM in the setup process to verify whether the VTC VM is up.

- Step 8** SSH into VTC VM using the IP address, administrative username/password given in the setup process (not vts-admin user).
-

Installing VTC VM - Manual Configuration using VNC

If the server where you install VTC is in a remote location with network latency or low bandwidth, you can use VNC to access the VTC VM and manually configure it using the CTC VM graphic console. To do this:

- Step 1** Connect to the controller node via SSH, and copy the vtc.qcow2 file to /var/lib/libvirt/images/ folder.
- Step 2** Copy the vtc.sample.xml file to your controller. Modify it as per your setup. The sample VTC XML file output is provided in [Sample Cisco VTS Configurations for Cisco NFVI, on page 133](#).
- Step 3** Replace the following sections of the vtc.sample.xml file:

```
<graphics type='spice' port='5900' autoport='yes' listen='127.0.0.1'>
    <listen type='address' address='127.0.0.1' />
</graphics>
```

with the following:

```
<graphics type='vnc' port='5900' autoport='yes' listen='0.0.0.0'>
    <listen type='address' address='0.0.0.0' />
</graphics>
```

Note Setting the listen address to 0.0.0.0 allows external clients to connect to the VNC port (5900). You have to make sure that iptables configuration (if any) allows inbound TCP port 5900 connections.

- Step 4** Create the VTC VM using following command:

```
virsh create vtc.sample.xml
```

You should now be able to use a VNC client to connect to the VTC VM graphic console and continue the setup.

- Step 5** Enter the following:

Note For items that take multiple values, such as DNS and NTP, use a space to separate each value.

- VTS Hostname
- DHCP/Static IP configuration for static IP
- Management IP address for VTC—This is the Cisco NFVI api (a) network IP address.
- Management IP Netmask (api network)
- Management Gateway address (api network)
- DNS Address—One of the DNS servers in setup_data.yaml::NETWORKING['domain_name_servers']
- DNS Search domain--- setup_data.yaml::NETWORKING['domain_name']
- Underlay IP address—This is the IP address for Cisco NFVI management/provisioning (mx) network.
- Underlay IP Netmask (mx network)

- NTP address—One of the `setup_data.yaml::NETWORKING['ntp_servers']` addresses
- Password change for user `vts-admin`—Enter the default user `vts-admin` password. The `vts-admin` user is used for password recovery and to revisit a configuration screen if you make a mistake or need to change the information. If you log into the VTC VM using `vts-admin` username and password again, you get the same dialog to go through the VTC VM setup again. This should match the value in `setup_data.yaml::VTS_PARAMETERS['VTS_PASSWORD']` or subsequently changed through the VTC UI to match the value in `setup_data.yaml::VTS_PARAMETERS['VTS_PASSWORD']`
- Administrator User—Enter administrative username and password. This username and password are used to login to the VM via SSH.
- Password for administrator user.

When VTC VM reboots at this time, wait for two minutes for the VTC VM to come up. You can ping the IP address given for VTC VM in the setup process to verify whether the VTC VM is up.

- Step 6** SSH into VTC VM using the IP address, administrative username/password given in the setup process (not `vts-admin` user).
-

Installing the VTSR VMs

Before you can install Cisco VTS for Cisco NFVI, you must install the VTSR VM and register it to VTS. VTSR VM is the control plane VM. Installing and registering the VTSR VM requires you to complete the following procedures:

- [Creating VTSR VM , on page 123](#)
- [Creating an ISO for IOS VTSR, on page 124](#)

Creating VTSR VM

The VTSR VM is essential to the Virtual VTEP topology. The VTSR VM contains a nested VM so VTSR must enable nesting.

Before you begin

You must complete VTS VM installation and change the VTC UI initial password to the password that you enter for Cisco VIM when you install Cisco VIM. This password is set in `setup_data.yaml` or Cisco VIM Insight. Login to VTC UI and create a site with Unique UUID and EVPN VxLAN Type. Then, update the site UUID in `setup_data.yaml` as `VTS_SITE_UUID`.

Bringing up the KVM-based VTSR VM

- Step 1** Create the VTSR VM XML referring the Cisco NFVI sample (`VTSR.XML`).
- Step 2** Generate an ISO file for the VTSR. See [Creating an ISO for IOS VTSR, on page 124](#) .
- Step 3** Create the VM using the XML.

```
virsh create VTSR.xml
```

Creating an ISO for IOS VTSR

To create an ISO file for VTSR:

Step 1 Create the system.cfg file based on the sample below.

- Note**
- Verify that the configuration file has no space or extra characters.
 - Before you enter the VTS_USERNAME and VTS_PASSWORD, review [Cisco VTS Usernames and Passwords in Cisco NFVI](#), on page 113.

```
# This is a sample VTSR configuration file
# Copyright (c) 2015 cisco Systems

# Protect the generated ISO, as it contains authentication data
# in plain text.

# The following are the common configurations for VTSR
# VTS Registration Information:
# VTS_ADDRESS should be the VTS IP. The value must be either an IP or a mask.
# VTS_ADDRESS is mandatory. If only the V4 version is specified,
# the V4 management interface for the VTSR (NODE1_MGMT_NETWORK_IP_ADDRESS)
# will be used. If the V6 version is specified, the V6 management interface
# for the VTSR (NODE1_MGMT_NETWORK_IPV6_ADDRESS) must be specified and will be used.
VTS_ADDRESS="10.85.88.152"
#VTS_IPV6_ADDRESS="a1::10"
# VTS_REGISTRATION_USERNAME used to login to VTS.
VTS_REGISTRATION_USERNAME="admin"
# VTS_REGISTRATION_PASSWORD is in plaintext.
VTS_REGISTRATION_PASSWORD="Cisco123!"
# VTSR VM Admin user/password
USERNAME="cisco"
PASSWORD="cisco123"

# Mandatory Management-VRF name for VTSR.
VTS_MANAGEMENT_VRF="vtsr-mgmt-vrf"

# VTSR VM Network Configuration for Node 1:
# NETWORK_IP_ADDRESS, NETWORK_IP_NETMASK, and NETWORK_IP_GATEWAY
# are required to complete the setup. Netmask can be in the form of
# "24" or "255.255.255.0"
# The first network interface configured with the VTC VM is used for
# underlay connectivity, while the second interface is used for the management network.
# For both MGMT and UNDERLAY networks, a <net-name>_NETWORK_IP_GATEWAY
# variable is mandatory and used for monitoring purposes.
#
# V6 is only supported on the mgmt network and dual stack is
# not supported. If both are specified, V6 will take priority (and
# requires VTS_IPV6_ADDRESS to be set).
# The *V6* parameters for the mgmt network are optional. Note that if V6 is used for mgmt
# it must be V6 on both nodes. Netmask must be the prefix length for V6.
NODE1_MGMT_NETWORK_IP_ADDRESS="19.1.0.20"
NODE1_MGMT_NETWORK_IP_NETMASK="255.255.255.0"
NODE1_MGMT_NETWORK_IP_GATEWAY="19.1.0.1"
```

```

#NODE1_MGMT_NETWORK_IPV6_ADDRESS="a1::20"
#NODE1_MGMT_NETWORK_IPV6_NETMASK="64"
#NODE1_MGMT_NETWORK_IPV6_GATEWAY="a1::1"
NODE1_UNDERLAY_NETWORK_IP_ADDRESS="19.0.128.20"
NODE1_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
NODE1_UNDERLAY_NETWORK_IP_GATEWAY="19.0.128.1"
# AUX network is optional
#NODE1_AUX_NETWORK_IP_ADDRESS="169.254.20.100"
#NODE1_AUX_NETWORK_IP_NETMASK="255.255.255.0"
#NODE1_AUX_NETWORK_IP_GATEWAY="169.254.20.1"

# XR Hostname
NODE1_XR_HOSTNAME="vtsr01"
# Loopback IP and netmask
NODE1_LOOPBACK_IP_ADDRESS="128.0.0.10"
NODE1_LOOPBACK_IP_NETMASK="255.255.255.255"

# Operational username and password - optional
# These need to be configured to start monit on VTSR

#VTSR_OPER_USERNAME="monit-ro-oper"
# Password needs an encrypted value
# Example : "openssl passwd -1 -salt <salt-string> <password>" 
#VTSR_OPER_PASSWORD="$1$cisco$b88M8bkCN2ZpXgEEc2sG9/"

# VTSR monit interval - optional - default is 30 seconds
#VTSR_MONIT_INTERVAL="30"

# VTSR VM Network Configuration for Node 2:
# If there is no HA, the following Node 2 configurations will remain commented and
# will not be used and Node 1 configurations alone will be applied.

# For HA , the following Node 2 configurations has to be uncommented
# VTSR VM Network Configuration for Node 2
# NETWORK_IP_ADDRESS, NETWORK_IP_NETMASK, and NETWORK_IP_GATEWAY
# are required to complete the setup. Netmask can be in the form of
# "24" or "255.255.255.0"
#
# The first network interface configured with the VTC VM is used for
# underlay connectivity, while the second interface is used for the management network.

# For both MGMT and UNDERLAY networks, a <net-name>_NETWORK_IP_GATEWAY
# variable is mandatory and used for monitoring purposes.
#
# V6 is only supported on the mgmt network and dual stack is
# not supported.If both are specified, V6 will take priority (and
# requires VTS_IPV6_ADDRESS to be set).
# The *V6* parameters for the mgmt network are optional. Note that if V6 is used for mgmt
# it must be V6 on both nodes. Netmask must be the prefix length for V6.

#NODE2_MGMT_NETWORK_IP_ADDRESS="19.1.0.21"
#NODE2_MGMT_NETWORK_IP_NETMASK="255.255.255.0"
#NODE2_MGMT_NETWORK_IP_GATEWAY="19.1.0.1"
##NODE2_MGMT_NETWORK_IPV6_ADDRESS="a1::21"
##NODE2_MGMT_NETWORK_IPV6_NETMASK="64"
##NODE2_MGMT_NETWORK_IPV6_GATEWAY="a1::1"
#NODE2_UNDERLAY_NETWORK_IP_ADDRESS="19.0.128.21"
#NODE2_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
#NODE2_UNDERLAY_NETWORK_IP_GATEWAY="19.0.128.1"
# AUX network is optional
# Although Aux network is optional it should be either present in both nodes
# or not present in both nodes.
# It cannot be present on Node1 and not present on Node2 and vice versa
#NODE2_AUX_NETWORK_IP_ADDRESS="179.254.20.200"
#NODE2_AUX_NETWORK_IP_NETMASK="255.255.255.0"
#NODE2_AUX_NETWORK_IP_GATEWAY="179.254.20.1"

```

Verifying Cisco VTS Installation in Cisco NFVI

```
# XR Hostname
#NODE2_XR_HOSTNAME="vtsr02"
# Loopback IP and netmask
#NODE2_LOOPBACK_IP_ADDRESS="130.0.0.1"
#NODE2_LOOPBACK_IP_NETMASK="255.255.255.255"

# VTS site uuid
VTS_SITE_UUID="abcdefab-abcd-abcd-abcd-abcdefabcdef"
```

Step 2 Copy your VTSR system.cfg files to the same path where the script resides. For example:

```
admin:/opt/cisco/package/vts/bin$ ls -l
total 1432
-rwxr-xr-x 1 vts-admin vts-admin 4767 Sep 29 16:40 build_vts_config_iso.sh
-rw-r--r-- 1 root      root      1242 Sep 29 23:54 system.cfg
```

Step 3 Create the ISO file as shown below (you need to log in as root):

```
root:/opt/cisco/package/vts/bin# ./build_vts_config_iso.sh vtsr system.cfg.
Validating input.
Generating ISO File. Done!
```

Step 4 Spawn the VTSR VM with the ISO connected to it.

Step 5 Power on the VM.

In case you spawn a new VTSR VM later, it comes up with VTSR Day Zero configuration and get re-registered with the VTC. Use the **sync-to** option available in the Config Sync feature to synchronize the configuration with the latest VTC configuration. See the *Synchronizing Configuration* section for more information.

Verifying Cisco VTS Installation in Cisco NFVI

The following procedures provide information about how to verify the Cisco VTS installation in Cisco NFVI.

Verifying VTSR VM Installation

To verify VTSR VM installation:

Before you begin

Ensure that the tenant network (t) gateway and management network (mx) gateway are reachable from the VTSR server.

Step 1 Log into the VTSR VM using the VTC VM console. If you had installed the VTC VM in an RedHat KVM based-OpenStack environment, use virt-manager or VNC console to log into the VM. See [Installing VTC VM - Manual Configuration using VNC, on page 122](#)

Step 2 Ping the Cisco NFVI tenant (t) network gateway IP address.

In case ping fails, verify Cisco NFVI tenant network.

Step 3 Ping the VTC Cisco NFVI management/provisioning (mx) network IP address.

In case ping fails, verify the mx network.

Note You should be able to ping the gateway IP address for both Cisco NFVI mx and t networks, as VTSR registers to the VTC using the VTC mx network IP address.

Verifying VTC VM Installation

To verify VTC VM installation:

Step 1 Log into the VTC VM just created using the VTC VM console.

- If you installed the VTC VM in an RedHat KVM based-OpenStack environment, - telnet 0 <console-port> (The console port is the Telnet port in the VTC.xml file.)

Step 2 Ping the Cisco NFVI api network gateway.

If ping fails, verify the VM networking to the Cisco NFVI api network.

Step 3 For the VTC VM CLI, ping the Cisco NFVI management/provisioning (mx) network gateway.

If ping fails, verify VM networking to the mx network.

Note Underlay network gateway is the switched virtual interface (SVI) created for IOSXRv and VTF on the leaf where the controller is connected.

Step 4 After a few minutes, verify whether the VTS UI is reachable by typing in the VTS api network IP in the browser.

Troubleshooting VTF Registration

If VTF registration issues arise, you can use the following commands to find the VTF registration logs on each Cisco NFVI compute node:

```
[root@devstack-71 neutron]# docker exec -it neutron_vtf_4269 bash
[root@devstack-71 /]# cd /var/log/vpfa
[root@devstack-71 vpfa]# ls
vpfa_err.log    vpfa_med.log   vpfa_server.log          vpfa_server_frequent.log  vpfa_stdout.log
vpfa_freq.log   vpfa_reg.log   vpfa_server_errors.log  vpfa_server_slow.log
[root@devstack-71 vpfa]# tail vpfa_reg.log
2016-06-23 02:47:22,860:INFO:VTF-REG: Sent PATCH {"vtf": {"username": "admin",
"vpp-client-name": "devstack-71", "ip": "34.34.34.5", "binding-host-name": "devstack-71",
"gateway-ip": "34.34.34.1", "local-mac": "00:3a:7d:6a:13:c9"} } to
https://172.18.96.15:8888/api/running/cisco-vts/vtfs/vtf
2016-06-23 02:47:23,050:INFO:VTF-REG-ERR: Failure:400!!!
```

A successful log example is shown below:

```
[root@devstack-71 vpfa]# tail vpfa_reg.log
2016-06-23 15:27:57,338:INFO:AUTH: Successful Login - User: admin
URI:/yang-api/datastore/interfaces Host:IPv4Address(TCP, '34.34.34.5', 21345) Method:GET
2016-06-23 15:28:07,340:INFO:AUTH: Successful Login - User: admin
URI:/yang-api/datastore/interfaces Host:IPv4Address(TCP, '34.34.34.5', 21345) Method:GET
```

If a VTF registration fails, check the following:

- IP network connectivity between the compute nodes and the VTC and VTSR VMs (Cisco NFVI tenant and management/provisioning networks)
- VTS_PARAMETERS—The VTS_USERNAME must be admin.
- The VTC and VTSR must be up and the VTS configurations must be applied. The VTSR must be registered with VTC.
- Check that the VTS UI shows "vtsgroup3" in Inventory->Authorization Groups.
- Check that the VTC Admin Username is admin and Device Username is what was set for XRVR_USERNAME in the VTSR config ISO.

Configuring Cisco VTS and VTSR After Installation

The following steps cover the Cisco VTS configurations you need to provision after installation.

Step 1 If you had changed the Cisco VTS username/password when you configured the VTS HA configuration, continue with Step 3. If not, log into the Cisco VTS GUI using the default username/password admin/admin.

Step 2 Change the Cisco VTS password using the UI Change Password tab.

Note Before you enter the Cisco VTS password, review [Cisco VTS Usernames and Passwords in Cisco NFVI](#), on page 113.

Step 3 Log into the VTC VM using the following command:

```
cd /opt/vts/bin
sudo ./vts-cli.sh -applyTemplate vtsr-underlay-loopback-template

./vts-cli.sh -applyTemplate vtsr-underlay-loopback-template command is applyTemplate and template
name is vtsr-underlay-loopback-template
Enter device name: <hostname of vtsr>
Enter loopback-interface: <loopback interface name>
Enter ipaddress: <loopback interface ip>
Enter netmask: <loopback interface netmask>
```

Similarly configure IGP config in VTSR

Step 4 Log into the VTC VM using the following command:

```
cd /opt/vts/bin
sudo ./vts-cli.sh -applyTemplate vtsr-underlay-ospf-template

./vts-cli.sh -applyTemplate vtsr-underlay-ospf-template command is applyTemplate and template name
is vtsr-underlay-ospf-template
Enter device name: <hostname of vtsr>
Enter process-name: <ospf process id >
Enter router-id: <ospf router id>
Enter area-address: <ospf area address>
Enter physical-interface: <VTSR interface connected to NFVI t-network>
Enter loopback-interface: <vtsr loopback interface>
Enter default-cost: <ospf default >
```

Installing VTS in an HA Configuration

Complete the following steps to install Cisco VTS in a Layer 2 HA configuration.

- Step 1** Create two VTC VMs. (In the following steps, these are referred to as VTC1 and VTC2.) When you create the VMs, reserve three IP addresses for each Cisco VIM network to which the VTC VM are connected as described in [Overview to Cisco VTS Installation in Cisco NFVI, on page 111](#).
- Step 2** If you changed the initial VTC password in a previous installation step, proceed to Step 4. If not, log into the VTC GUI using the default username/password admin/admin.
- Step 3** Change the VTC password using the UI Change Password tab. See [Cisco VTS Usernames and Passwords in Cisco NFVI, on page 113](#) for information about Cisco VTS usernames and passwords.
- Step 4** Edit the `cluster.conf` file on VTC1 and VTC2 located in `/opt/vts/etc/`. Both VTCs must have identical information in the `cluster.conf` file. Parameters includes:
- `vip_public`—VIP address used for the Cisco VIM API (a) network.
 - `vip_private`—VIP address used for VTS on the Cisco VIM management/provisioning (mx) network. Cisco VIM uses VTFs, so this field must be entered. The `vip_private` field is the VIP for the VTS master private interface.
 - `master_name`—Enter the name of the primary VTC in the HA configuration.
 - `master_ip`—The master VTC IP address used for the Cisco NFVI API network.
 - `slave_name`—Enter the name of the secondary VTC in the HA configuration.
 - `slave_ip`—The secondary VTC IP address used for the Cisco NFVI API network.
 - `external_ip`—The external IP address. This comes from the Cisco VIM `setup_data.yaml` file after you complete the Cisco VIM installation and Cisco VIM configuration for Cisco VTS installation. For details on Cisco VIM configuration, see [Cisco VIM Configuration for Cisco VTS Installation , on page 181](#) procedure.

```
###Virtual Ip of VTC Master on the public interface. Must fill in at least 1
vip_public=
vip_public_ipv6=

###VTC1 Information. Must fill in at least 1 ip address
master_name=
master_ip=
master_ipv6=

###VTC2 Information. Must fill in at least 1 ip address
slave_name=
slave_ip=
slave_ipv6=

###In the event that a network failure occurs evenly between the two routers, the cluster needs an
outside ip to determine where the failure lies
###This can be any external ip such as your vmm ip or a dns but it is recommended to be a stable ip
within your environment
###Must fill in at least 1 ip address
external_ip=
external_ipv6=


#####
### Non-mandatory fields ###
```

Installing VTS in an HA Configuration

```
#####
###If you intend to use a virtual topology forwarder (VTF) in your environment, please fill in the
vip for the underlay as well as the underlay gateway. Otherwise leave blank.
###Virtual Ip of VTC Master on the private interface. You can fill in ipv4 configuration, ipv6, or
both if you use both
vip_private=
private_gateway=

vip_private_ipv6=
private_gateway_ipv6=

###If you have your vtc's in different subnets, xrvr needs to be configured to route traffic and the
below section needs to be filled in
###If you have your vtc's on the same subnet, the below section has be skipped

###Name of your vrf. Example: VTS_VIP
vrf_name=

###Ip of your first Xrvr. Example: 11.1.1.5
xrvr1_mgmt_ip=

###List of neighbors for xrvr1, separated by comma. Example: 11.1.1.1,11.1.1.2
xrvr1_bgp_neighbors=
xrvr1_bgp_neighbors_ipv6=

###Ip of your second Xrvr. Example: 12.1.1.5
xrvr2_mgmt_ip=

###List of neighbors for xrvr2, separated by comma. Example: 12.1.1.1,12.1.1.2
xrvr2_bgp_neighbors=
xrvr2_bgp_neighbors_ipv6=

###Username for Xrvr
xrvr_user=

###Xrvr ASN information
remote ASN=
local ASN=

###Xrvr BGP information
bgp_keepalive=
bgp_hold=

###Update source for Xrvr1 (i.e. loopback)
xrvr1_update_source=

###Update source for Xrvr2 (i.e. loopback)
xrvr2_update_source=

###Router BGP Id for Xrvr1
xrvr1_router_id=

###Router BGP Id for Xrvr2
xrvr2_router_id=

###XRVR1 name
xrvr1_name=

###XRVR2 name
xrvr2_name=

###If you plan on having your VTC's on different subnets and intend to use a virtual topology forwarder
(VTF) in your environment,
```

```

### please fill out the following fields. Otherwise, leave blank

###List of neighbors for xrvr1, separated by comma. Example: 2.2.2.2,2.2.2.3
xrvr1_underlay_neighbors=
xrvr1_underlay_neighbors_ipv6=

###List of neighbors for xrvr2, separated by comma. Example: 3.3.3.2,3.3.3.3
xrvr2_underlay_neighbors=
xrvr2_underlay_neighbors_ipv6=

###Directly connected Tor information for Xrvr1
xrvr1_directly_connected_device_ip=
xrvr1_directly_connected_device_ipv6=
xrvr1_directly_connected_device_user=
xrvr1_directly_connected_device_neighbors=
xrvr1_directly_connected_device_neighbors_ipv6=
xrvr1_directly_connected_ospf=
xrvr1_directly_connected_router_id=
xrvr1_directly_connected_update_source=

###Directly connected Tor information for Xrvr2
xrvr2_directly_connected_device_ip=
xrvr2_directly_connected_device_user=
xrvr2_directly_connected_device_neighbors=
xrvr2_directly_connected_device_neighbors_ipv6=
xrvr2_directly_connected_ospf=
xrvr2_directly_connected_router_id=
xrvr2_directly_connected_update_source=

###VPC Peer information if any. Otherwise leave blank
xrvr1_vpc_peer_ip=
xrvr1_vpc_peer_user=
xrvr1_vpc_peer_ospf=
xrvr1_vpc_peer_router_id=
xrvr1_vpc_peer_update_source=

xrvr2_vpc_peer_ip=
xrvr2_vpc_peer_user=
xrvr2_vpc_peer_ospf=
xrvr2_vpc_peer_router_id=
xrvr2_vpc_peer_update_source=

###VTC Underlay Addresses
vtc1_underlay=
vtc2_underlay=
vtc1_underlay_ipv6=
vtc2_underlay_ipv6=

##Gateway of secondary L3 underlay
vtc2_private_gateway=
vtc2_private_gateway_ipv6=

```

Step 5

Execute the cluster installer script, `cluster_install.sh`, located in `/opt/vts/bin/` on VTC1 and VTC2. Do not run the script until have completed Steps 1-5.

```

admin@vtc1:/opt/vts/bin$ sudo ./cluster_install.sh
[sudo] password for admin:
Change made to ncs.conf file.
Need to restart ncs
Created symlink from /etc/systemd/system/multi-user.target.wants/pacemaker.service to
/lib/systemd/system/pacemaker.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/corosync.service to
/lib/systemd/system/corosync.service.

```

Installing VTS in an HA Configuration

```
Please run cluster_install.sh on vtc2.waits until finished Both nodes are online.
Configuring master Configuring Pacemaker resources
Master node configuration finished
HA cluster is installed
```

Note In order for HA to run, the cluster_install.sh script updates /etc/hosts with the VTC information. If run on the node you specified as master, it completes the basic cluster setup, then wait for the slave to complete. Once the slave is finished, the master completes the remainder of the setup.

When the cluster_install script is finished on the master, you can see both the public and private VIP using 'ip addr'. If you use VTFs, now that the VIP is up, both VTSRs completes their auto-registration.

Step 6 Verify the HA Status:

```
admin@vtc1:/opt/cisco/package/vtc/bin$ sudo crm status
Last updated: Wed May 4 00:00:28 2016
Last change: Wed May 4 00:00:10 2016 via crm_attribute on vtc2
Stack: corosync
Current DC: vtc2 (739533872) - partition with quorum
Version: 1.1.10-42f2063
2 Nodes configured
4 Resources configured

Online: [ vtc1 vtc2 ]

ClusterIP      (ocf::heartbeat:IPAddr2):           Started vtc1
Master/Slave Set: ms_vtc_ha [vtc_ha]
  Masters: [ vtc1 ]
  Slaves: [ vtc2 ]
ClusterIP2     (ocf::heartbeat:IPAddr2):           Started vtc1

admin@vtc1:/opt/cisco/package/vtc/bin$ sudo ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
  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
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
  link/ether 52:54:00:00:bd:0f brd ff:ff:ff:ff:ff:ff
    inet 11.1.1.4/24 brd 11.1.1.255 scope global eth0
      valid_lft forever preferred_lft forever
    inet 11.1.1.2/32 brd 11.1.1.2 scope global eth0
      valid_lft forever preferred_lft forever
    inet6 2001:420:10e:2010:5054:ff:fe00:bd0f/64 scope global dynamic
      valid_lft 2591955sec preferred_lft 604755sec
    inet6 fe80::5054:ff:fe00:bd0f/64 scope link
      valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
  link/ether 52:54:00:4c:11:13 brd ff:ff:ff:ff:ff:ff
    inet 15.15.15.4/24 brd 11.1.1.255 scope global eth1
      valid_lft forever preferred_lft forever
    inet 15.15.15.20/32 brd 11.1.1.20 scope global eth1
```

Completing VTSR HA Configuration

Complete the following steps to set up the VTSR HA configuration:

Before you begin

You must complete a VTS VM installation and change the VTC UI initial password to the password that you enter for Cisco VIM when you install Cisco VIM. This password is set in setup_data.yaml or the Cisco VIM Insight.

Login to VTC UI and create a site with Unique UUID and EVPN VxLAN Type. Update this UUID as VTS_SITE_UUID in setup_data.yaml.

Ensure the tenant network (t) gateway and management network (mx) gateway are reachable from the VTSR server.

Power on the 2 VTSR VM's as per the VTSR install step. The VTSR VM comes up in active/active HA mode.

Uninstalling VTC HA

To move VTC back to its original pre-HA state, run the following script on both the active and standby nodes.

```
sudo /opt/vts/bin/cluster_uninstall.sh
```

Sample Cisco VTS Configurations for Cisco NFVI

Sample VTC VM libvirt Domain Configuration

```
<domain type='kvm' id='1332'>
  <name>VTC-release2.1</name>
  <uuid>5789b2bb-df35-4154-a1d3-e38cefc856a3</uuid>
  <memory unit='KiB'>32389120</memory>
  <currentMemory unit='KiB'>32388608</currentMemory>
  <vcpu placement='static'>8</vcpu>
  <resource>
    <partition>/machine</partition>
  </resource>
  <os>
    <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
    <boot dev='hd' />
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <cpu mode='custom' match='exact'>
    <model fallback='allow'>Westmere</model>
    <feature policy='require' name='vmx' />
  </cpu>
  <clock offset='utc' />
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
```

Sample Cisco VTS Configurations for Cisco NFVI

```

<devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
        <driver name='qemu' type='qcow2' cache='none' />
        <source file='/home/cisco/VTS2.1/vtc.qcow2' />
        <target dev='vda' bus='virtio' />
        <alias name='virtio-disk0' />
        <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0' />
    </disk>
    <controller type='usb' index='0'>
        <alias name='usb0' />
        <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2' />
    </controller>
    <controller type='pci' index='0' model='pci-root'>
        <alias name='pci.0' />
    </controller>
    <controller type='virtio-serial' index='0'>
        <alias name='virtio-serial0' />
        <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0' />
    </controller>
    <interface type='bridge'>
        <mac address='52:54:00:5b:12:3a' />
        <source bridge='br-ex' />
        <virtualport type='openvswitch'>
            <parameters interfaceid='263c1aa6-8f7d-46f0-b0a3-bdbdad40fe41' />
        </virtualport>
        <target dev='vnet0' />
        <model type='virtio' />
        <alias name='net0' />
        <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0' />
    </interface>
    <interface type='bridge'>
        <mac address='52:54:00:8d:75:75' />
        <source bridge='br-control' />
        <virtualport type='openvswitch'>
            <parameters interfaceid='d0b0020d-7898-419e-93c8-15dd7a08eebd' />
        </virtualport>
        <target dev='vnet1' />
        <model type='virtio' />
        <alias name='net1' />
        <address type='pci' domain='0x0000' bus='0x00' slot='0x0b' function='0x0' />
    </interface>
    <serial type='tcp'>
        <source mode='bind' host='127.0.0.1' service='4888' />
        <protocol type='telnet' />
        <target port='0' />
        <alias name='serial0' />
    </serial>
    <console type='tcp'>
        <source mode='bind' host='127.0.0.1' service='4888' />
        <protocol type='telnet' />
        <target type='serial' port='0' />
        <alias name='serial0' />
    </console>
    <channel type='spicevmc'>
        <target type='virtio' name='com.redhat.spice.0' />
        <alias name='channel0' />
        <address type='virtio-serial' controller='0' bus='0' port='1' />
    </channel>
    <input type='mouse' bus='ps2' />
    <graphics type='spice' port='5900' autoport='yes' listen='127.0.0.1'>
        <listen type='address' address='127.0.0.1' />
    </graphics>
    <sound model='ich6' />

```

```

        <alias name='sound0' />
        <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0' />
    </sound>
    <video>
        <model type='qxl' ram='65536' vram='65536' heads='1' />
        <alias name='video0' />
        <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0' />
    </video>
    <memballoon model='virtio'>
        <alias name='balloon0' />
        <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0' />
    </memballoon>
</devices>
<seclabel type='dynamic' model='selinux' relabel='yes'>
    <label>system_u:system_r:svirt_t:s0:c26,c784</label>
    <imagelabel>system_u:object_r:svirt_image_t:s0:c26,c784</imagelabel>
</seclabel>
</domain>

```

Sample VTSR VM libvirt Domain Configuration

```

<domain type='kvm' id='20'>
    <name>SAMPLE-VTSR-1</name>
    <memory unit='GiB'>48</memory>
    <cpu mode='host-passthrough' />
    <vcpu placement='static'>14</vcpu>
    <resource>
        <partition>/machine</partition>
    </resource>

    <os>
        <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
        <boot dev='hd' />
        <boot dev='cdrom' />
    </os>
    <features>
        <acpi/>
        <apic/>
        <pae/>
    </features>
    <clock offset='localtime' />
    <on_poweroff>destroy</on_poweroff>
    <on_reboot>restart</on_reboot>
    <on_crash>restart</on_crash>
    <devices>
        <emulator>/usr/libexec/qemu-kvm</emulator>

        <disk type='file' device='cdrom'>
            <driver name='qemu' />
            <source file='/home/admin/VTS20/images/vtsr_node1_cfg.iso' />
            <target dev='hda' bus='ide' />
            <readonly/>
        </disk>

        <disk type='file' device='disk'>
            <driver name='qemu' type='qcow2' />
            <source file='/home/admin/VTS20/images/vtsr.qcow2' />
            <target dev='vda' bus='virtio' />
            <alias name='virtio-disk0' />
            <address type='pci' domain='0x0000' bus='0x00' slot='0x09' function='0x0' />
        </disk>

        <controller type='usb' index='0'>
            <alias name='usb0' />
            <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2' />

```

Sample Cisco VTS Configurations for Cisco NFVI

```

</controller>
<controller type='ide' index='0'>
    <alias name='ide0' />
    <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1' />
</controller>
<controller type='pci' index='0' model='pci-root'>
    <alias name='pci.0' />
</controller>

<interface type='bridge'>
    <source bridge='br-ex' />
    <virtualport type='openvswitch'>
        <parameters interfaceid='4ffa64df-0d57-4d63-b85c-78b17fcac60a' />
    </virtualport>
    <target dev='vtsr-dummy-mgmt' />
    <model type='virtio' />
    <alias name='vnet1' />
    <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0' />
</interface>

<interface type='bridge'>
    <source bridge='br-inst' />
    <virtualport type='openvswitch'>
        <parameters interfaceid='4ffa64df-0d67-4d63-b85c-68b17fcac60a' />
    </virtualport>
    <target dev='vtsr-dummy-2' />
    <model type='virtio' />
    <alias name='vnet1' />
    <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0' />
</interface>

<interface type='bridge'>
    <source bridge='br-inst' />
    <virtualport type='openvswitch'>
        <parameters interfaceid='4ffa64df-0f47-4d63-b85c-68b17fcac70a' />
    </virtualport>
    <target dev='vtsr-dummy-3' />
    <model type='virtio' />
    <alias name='vnet1' />
    <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0' />
</interface>

<interface type='bridge'>
    <source bridge='br-inst' />
    <virtualport type='openvswitch'>
        <parameters interfaceid='4ffa64df-0d47-4d63-b85c-58b17fcac60a' />
    </virtualport>
    <vlan>
        <tag id='800' />
    </vlan>
    <target dev='vtsr-gig-0' />
    <model type='virtio' />
    <alias name='vnet1' />
    <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0' />
</interface>

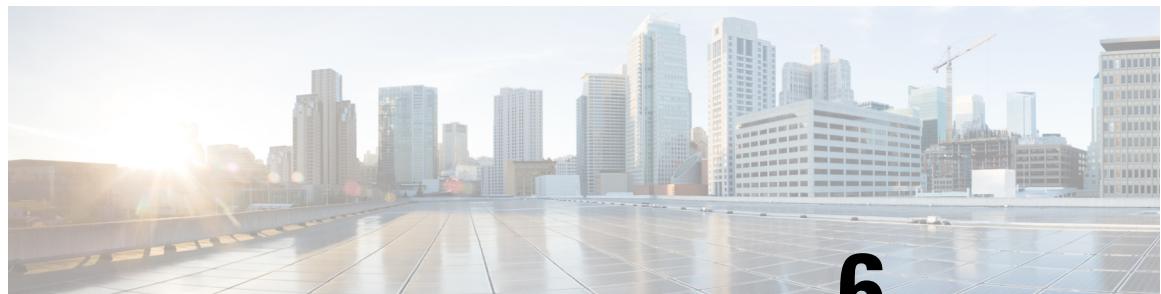
<interface type='bridge'>
    <source bridge='br-ex' />
    <virtualport type='openvswitch'>
        <parameters interfaceid='3ffa64df-0d47-4d63-b85c-58b17fcac60a' />
    </virtualport>
    <target dev='vtsr-gig-1' />

```

```
<model type='virtio'/>
<alias name='vnet1' />
<address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0' />
</interface>

<interface type='bridge'>
    <source bridge='br-inst' />
    <virtualport type='openvswitch'>
        <parameters interfaceid='a2f3e85a-4de3-4ca9-b3df-3277136c4054' />
    </virtualport>
    <vlan>
        <tag id='800' />
    </vlan>
    <target dev='vtsr-gig-2' />
    <model type='virtio' />
    <alias name='vnet3' />
    <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0' />
</interface>

<serial type='pty'>
    <source path='/dev/pts/0' />
    <target port='0' />
    <alias name='serial0' />
</serial>
<console type='pty' tty='/dev/pts/0'>
    <source path='/dev/pts/0' />
    <target type='serial' port='0' />
    <alias name='serial0' />
</console>
<input type='tablet' bus='usb'>
    <alias name='input0' />
</input>
<input type='mouse' bus='ps2' />
<graphics type='vnc' port='5900' autoport='yes' listen='0.0.0.0' keymap='en-us'>
    <listen type='address' address='0.0.0.0' />
</graphics>
<video>
    <model type='cirrus' vram='9216' heads='1' />
    <alias name='video0' />
    <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0' />
</video>
<memballoon model='virtio'>
    <alias name='balloon0' />
    <address type='pci' domain='0x0000' bus='0x00' slot='0x0a' function='0x0' />
</memballoon>
</devices>
</domain>
```

CHAPTER 6

Installing Cisco VIM

The following topics tell you how to configure and install Cisco VIM:

- [Cisco VIM Installation Overview, on page 139](#)
- [Installing Cisco VIM, on page 140](#)
- [Cisco VIM Client Details, on page 142](#)
- [Re-installing Pod with same Image version, on page 145](#)
- [Cisco VIM Configuration Overview, on page 146](#)

Cisco VIM Installation Overview

Before you can install Cisco Virtual Infrastructure Manager, complete the procedures in *Preparing for Cisco NFVI Installation*. If your management node does not have Internet access, complete the *Preparing to Install Cisco NFVI on Management Nodes Without Internet Access* procedure. The Cisco VIM installation procedure provides two methods for downloading and installing the Cisco VIM installation files, from USB stick prepared for installation, or from the Internet.

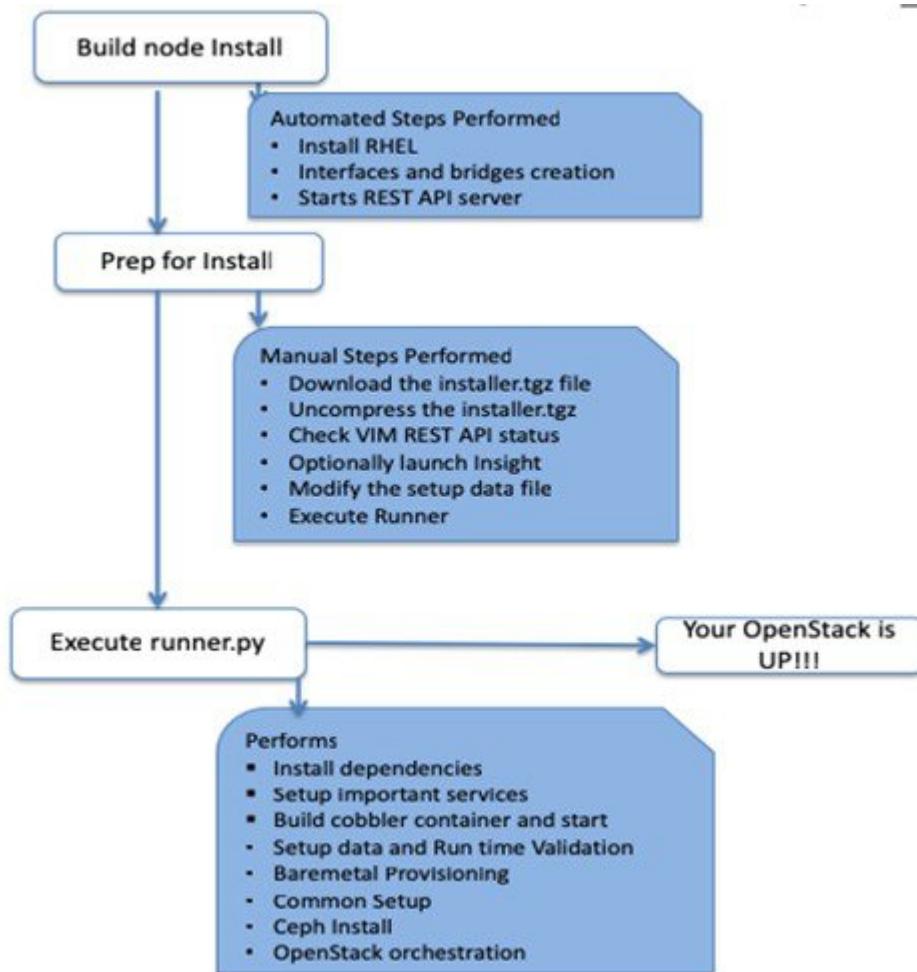
Completing these procedures ensures the Cisco NFVI network infrastructure is set up before the Cisco VIM installation. The bootstrap script is then kicked off, which downloads installer repository, installs Docker and dependencies and starts installer web service,

The Cisco VIM installer can then be launched. It validates the testbed configuration file (`setup_data.yaml`), creates new vNICs on the controller, compute, and dedicated storage nodes based on the configuration provided in the `setup_data.yaml` file. This is followed by the PXEboot Execution Environment (PXE) boot of RHEL onto the target nodes (control, compute and storage) through the Cobbler server set up on the management node. After the installation, the Cisco VIM installer performs common steps across all the Cisco NFVI nodes.

Next, Ceph related packages required for managing the cluster and creating OSD and monitor nodes are installed on the control and storage nodes. By default, the minimum three Ceph monitor nodes are installed at the host level on the control nodes. These serve as management nodes and have the administration keyring. Ceph configurations, such as `ceph.conf` and Ceph client keyrings files, are stored under `/etc/ceph` on each controller. Each Ceph storage node associates an Object Storage Daemon (OSD) to a physical hard drive with a write journal on a separate SSD to support small block random I/O.

The following illustration provides an overview to the Cisco VIM installation.

Figure 39: Cisco VIM Installation Flow



If you have Cisco Unified Management, complete only part of the Cisco VIM installation procedure and proceed to the [Installing Cisco VIM Insight](#) on page procedure followed by [Installing Cisco VIM through Cisco VIM Unified Management](#) to complete the configuration and setup of Cisco VIM using the Cisco VIM Insight. If you do not have Cisco VIM UM, configure Cisco VIM by editing the data_setup.yaml as described in the Cisco VIM installation.

Installing Cisco VIM

This procedure allows you to install the Cisco VIM on a Cisco NFVI management node:

Before you begin

- You need to get Cisco NFVI installation file download site credentials from your Cisco account representative.

- For management nodes with no Internet access, you need a USB stick containing the Cisco NFVI installation files. To prepare the USB stick, see [Preparing to Install Cisco NFVI on Management Nodes Without Internet Access, on page 77](#)
- The private networks 192.168.1.0/24 and 192.168.2.0/24 are internally reserved for testing the cloud from a control and data plane point of view. Cisco recommends that you do not use these reserved networks while preparing network layouts.
- You need to provide a valid certificate signed by a trusted certificate authority, for the Cisco VIM deployment. It needs to be a server certificate with a common name matching the IP address and DNS name specified in the setup data file under "external_lb_vip_address" and "external_lb_vip_fqdn". To ensure security, use only the valid certificate signed by a trusted certificate authority in a production environment. For details on generating self-signed certificate, see [Setting Up Cisco VIM OpenStack Configuration, on page 166](#)

Step 1

If your management node does not have Internet access, use the prepared USB stick and complete the following steps:

- Insert the USB stick into the management node drive.
- Run the import_artifacts.sh script to copy all artifacts onto the management node, for example:

```
cd ~/installer-<tag_id>/tools
./import_artifacts.sh
```

All the installation artifacts are copied to /var/cisco/artifacts/ on the management node

Step 2

If you are installing Cisco VIM Insight, navigate to [Installing Cisco VIM Unified Management](#) and complete the Cisco VIM Insight installation.

If you are not installing Cisco VIM Insight, complete the following steps.

Step 3

Change to the installer directory by running the following command:

```
cd ~/installer-<tag_id>
```

Step 4

Create a dir (for example, ~/Save/) to contain a copy of the setup_data.yaml file, the file that configures the Cisco NFVI for your particular implementation.

Step 5

Change to the openstack-configs directory and copy the example Cisco VIM setup_data.yaml file into the directory you just created:

```
cd openstack-configs/
cp setup_data.yaml.<C_or_B>_Series_EXAMPLE setup_data.yaml
~/Save/setup_data.yaml
```

Note Only the CPU and MEM allocation ratio needs to be changed for the target pod. Update the following to your target value:

NOVA_RAM_ALLOCATION_RATIO: 1.5 # range of 1.0 to 4.0

NOVA_CPU_ALLOCATION_RATIO: 16.0 # range of 1.0 to 16.0

Step 6

With a yaml editor, modify the copied example setup_data.yaml file as the data setup file for your implementation. This includes both Cisco NFVI data and OpenStack parameters.

Step 7

If you intend to run the cloud over TLS, see [Setting Up Cisco VIM OpenStack Configuration, on page 166](#) for TLS certificate generation.

Step 8

Run the installation:

```
ciscovim --setupfile ~/Save/setup_data.yaml run
```

After the installation is complete, you can view the installation logs at /var/log/mercury.

Cisco VIM Client Details

Cisco VIM combines the CLI and API so that you can use the CLI or API installer transparently.



Note For a complete list of Cisco VIM REST API commands, see the *Cisco NFVI Administrator Guide*.

Before you use the Cisco VIM CLI, check that the API server is up and pointing to the right installer directory. You can execute the following command to validate the state of the API server and the installer directory it is referencing:

```
# cd installer-<tagid>/tools
./restapi.py -a status
Status of the REST API Server: active (running) since Thu 2016-08-18 09:15:39 UTC; 9h ago
REST API launch directory: /root/installer-<tagid>/
```

Verify the server status is active and the restapi launch directory is the same the directory from where the installation is launched. If the installer directory, or the REST API state is not correct, go to the target installer directory and execute the following:

```
# cd new-installer-<tagid>/tools
./restapi.py -a setup
```

```
Check if the REST API server is running from the correct target directory
#./restapi.py -a status
Status of the REST API Server: active (running) since Thu 2016-08-18 09:15:39 UTC; 9h ago
REST API launch directory: /root/new-installer-<tagid>/
```

The REST API tool also provides the options to restart, tear down and reset password for the REST API server as listed:

```
# ./restapi.py --h
usage: restapi.py [-h] --action ACTION [--yes] [--verbose]

REST API setup helper

optional arguments:
  -h, --help            show this help message and exit
  --action ACTION, -a ACTION
                        setup - Install and Start the REST API server.
                        teardown - Stop and Uninstall the REST API server.
                        restart - Restart the REST API server.
                        regenerate-password - Regenerate the password for REST API server.
                        reconfigure-tls - Reconfigure SSL certificates and key.
                        upgrade - Upgrade to new workspace.
                        reset-password - Reset the REST API password with user given
password.
  --status              Check the status of the REST API server.
  --yes, -y              Skip the dialog. Yes to the action.
  --verbose, -v          Perform the action in verbose mode.
```

If the REST API server is not running, executing **ciscovim** shows the following error message:

```
# ciscovim --setupfile ~/Save/<setup_data.yaml> run
```

If the installer directory, or the REST API state is not correct or it is pointing to an incorrect REST API launch directory, go to the installer-<tagid>/tools dir and execute:

```
# ./restapi.py --action setup
```

To confirm that the Rest API server state and launch directory is correct, execute:

```
# ./restapi.py --action status
```

If you ran the REST API recovery step on an existing pod, run the following command to ensure that the REST API server continues to manage the existing pod:

```
# ciscovim --setup_file <setup_data_file_path> --perform 7 -y
```

For an overview to the commands you can execute from the CLI, enter the following command:

```
ciscovim --help
usage: ciscovim [--setupfile <setupdata_file>] <subcommand> ...

Command-line interface to the Cisco Virtualized manager

Positional arguments:
<subcommand>
  run                  Perform/terminate an install operation
  install-status       Status of installation of the Openstack cloud
  list-steps          List steps
  add-computes        Add compute-nodes to the Openstack cloud
  add-storage         Add a storage-node to the Openstack cloud
  list-nodes          List the nodes in the Openstack cloud
  remove-computes    Remove compute-nodes from the Openstack cloud
  remove-storage      Remove a storage-node from the Openstack cloud
  replace-controller Replace a controller in the Openstack cloud
  list-openstack-configs List of Openstack configs that can be changed
                        using reconfigure
  list-password-keys List of password keys that can be changed
                      using reconfigure
  reconfigure         Reconfigure the Openstack cloud
  cluster-recovery   Recover the Openstack cluster after a network
                     partition or power outage
  mgmtnode-health    Show health of the Management node
  commit              Commit an update
  rollback             Rollback an update
  update              Update the Openstack cloud
  update-status       Status of the update operation
  upgrade             Upgrade the Openstack cloud
  check-fernet-keys  Check whether the fernet keys are successfully
                     synchronized across keystone nodes
  NFVbench            Launch NFVbench Flows
  nfvimon             NFVI Monitoring / Zenoss management operations
  resync-fernet-keys Resynchronize the fernet keys across all the
                     keystone nodes
  rotate-fernet-keys Trigger rotation of the fernet keys on
                     keystone
  client-version     Show Virtualized Infrastructure Manager
                     Version
  version             Show Virtualized Infrastructure Manager
                     Version
  help                Display help about this program or one of its
```

```

subcommands.

Optional arguments:
  --setupfile <setupdata_file>

See "ciscovim help COMMAND" for help on a specific command.

To look at the help for a sub-command (e.g. run) execute the following:
# ciscovim help run
usage: ciscovim run [--join] [--perform <perform>] [--skip <skip>] [-y] Perform a install
operation
Optional arguments:
--join Join the installation process
--perform <perform> Perform the following steps.
--skip <skip> Skip the following steps.
-y, --yes Yes option to skip steps without prompt [root@MercRegTB1 installer]#
You can also run the installer in multiple smaller steps. To understand the steps involved
during installation
execute the following command:
# ciscovim list-steps
Virtualized Infrastructure Manager:
=====
+-----+-----+
| Operations | Operation ID |
+-----+-----+
| INPUT_VALIDATION | 1 |
| MGMTNODE_ORCHESTRATION | 2 |
| VALIDATION | 3 |
| BAREMETAL | 4 |
| COMMONSETUP | 5 |
| CEPH | 6 |
| ORCHESTRATION | 7 |
| VMTP | 8 |
+-----+-----+

```

To execute the installer in steps, include specific steps from above. For example:

```
$ ciscovim run --perform 1,3 -y
```

Similarly, you can execute the installation using the skip option, where you explicitly indicate which options to skip. For example

```
$ ciscovim run --skip 1,3 -y
```



Note When using the step-by-step installation, keep a track of what steps are already completed, or unpredictable results might occur.

While the install time varies from pod to pod, typical installation times through the Internet for a UCS C-series with three controller, nine compute, and three storage are listed in the following table.

Table 23:

Operation ID	Operation	Estimated Time
1	Input validation	6 minutes
2	Management node orchestration	40 minutes

Operation ID	Operation	Estimated Time
3	Run time Validation	30 seconds
4	Bare metal	60 minutes
5	Host setup	10 minutes
6	Ceph	5 minutes
7	Orchestration	25 minutes
8	VMTP (external and provider networks)	14 minutes

Re-installing Pod with same Image version

In unforeseen circumstances there might be a need to reinstall the pod with the same image version. To alleviate the need of a reimaging of the management node, followed by re-install, you can take the following steps to re-install the pod on assuming that the management node is compatible to the same tag. Ensure that you use the same servers for re-installation. If a different set of servers are used for the re-installation, the servers from the previous install which are no longer participating in the new install must be powered off to avoid the duplicate IP floating in the network.

Listed below are the steps to reinstall the pod without reimaging the management node.

- Step 1** Copy the setup_data.yaml from /root/openstack-configs/ directory to ~/Save/

```
cd ~/installer-<3.2.0>
./unbootstrap.sh -k
```

- Step 2** Verify that no docker containers are running

```
docker ps -a
```

- Step 3** Verify that no docker images are present

```
docker images
```

- Step 4** Setup RestAPI

```
cd ~/installer-3.2.0/tools
./restapi -a setup
```

- Step 5** Regenerate TLS certificate, if needed or TLS is enabled.

```
cd ~/installer-3.2.0
tools/tls_cert_gen.sh -f ~/Save/setup_data.yaml
```

- Step 6** Re-run Cisco VIM installation

```
ciscovim run --setupfile ~/Save/setup_data.yaml
```

Cisco VIM Configuration Overview

The following topics provide a list of Cisco NFVI configurations you must enter in setup_data.yaml with a yaml editor. These configurations has to be performed prior to running the Cisco VIM installation. If you are installing Cisco Insight, you have to complete the Cisco VIM data and OpenStack configurations using VIM Insight as described in [Installing Cisco VIM through Cisco VIM Unified Management](#).

Configuring ToR Automatically

Cisco VIM provides a complete automation of the cloud deployment. It automates day-0 configuration of N9xxx series Top of Rack (ToR) switches. This feature is optional and applicable only to the Pods that are running with or without ACI. For ToR switch details related to ACI, see [Enabling ACI in Cisco VIM](#).

It automates Power-On Auto Provisioning (post-POAP) configuration on ToR with one or more pair of identical Cisco N9300 series switches. The day-0 ToR automation configures the interfaces that are connected to the management (br_mgmt), control, compute, and storage nodes of the pod. In addition, it configures the VPC peer link interfaces for ToR pairs. The automation handles both B and C-series pods. The automation includes configuration of the edge ports in the leaf switches off which the hosts hang-out and the VPC peer link between the switches.

Auto-configuration feature does not include the configuration of the spine switches and the connectivity between the leaf and the spine; that is the upstream link of the spine switches that carry the external VLAN connectivity.

As the feature is a post-POAP automation provisioning, ensure that the management interface, vrf, and admin user are pre-provisioned on each ToR switch. Also, you must enable ssh in each ToR.

The recommended N9K switch software versions are 7.0(3)I4(6) and 7.0(3)I6(1). Bootstrapping the ToR image is still a manual process. Ensure that the installer API interface (br_api) is up and running on the management node with SSH. You can access each ToR through its management interface from the Cisco VIM management node using SSH.

Setting Up Cisco VIM Data Configuration

You can install and configure the Cisco VIM deployment using the Cisco VIM configuration file (setup_data.yaml). Ensure that you take extreme care while creating the configuration file, as any change in the configuration after deployment, with the exception (example: NFVIMON, of adding and removing nodes and so on) causes a stack redeployment.



Note Any change done to the pod networking layout plan configured in setup_data.yaml requires the pod to be reinstalled.

If your configuration is correct, the installation goes smoothly. Cisco recommends using a YAML editor on Linux (PyCharm, Komodo or vi/vim with YAML plugin) to edit this file. Items shown in brown must be

changed to your specific testbed. Do not copy the examples shown below into your YAML file, as your browser might render the characters differently.

If you are using the Cisco VIM installer, you cannot update the OpenStack config files (for example, ml2_conf.ini, and other files) directly. All OpenStack configurations must be in the setup_data.yaml file. This ensures that the installer has a view of the OpenStack deployment, so that it can reliably perform software updates and upgrades. This ensures a consistent and repeatable installation. Key setup file parts are shown in the following sections.

Setting up ToR Configurations for B-series and C-series

The ToR configuration is driven through the mercury setup_data.yaml configuration. The information for automated TOR configuration is provided in two parts in the setup_data.yaml file. The common information is in the TORSWITCHINFO section, whereas the information on individual switch ports connected to specific nodes are under SERVERS section for C-seires, and UCSM-COMMON section for B-series. If the TORSWITCHINFO section is not provided or CONFIGURE_TORS attribute under TORSWITCHINFO then all the ToR provisioning related steps are skipped. The ToR section contains attributes related to ToR connection, configuration for the management interface for the management node, and vPC peer details in case of ToR pairs.


Note

The port-channel number for the vPC peer link interfaces, is derived from the Vpc domain. The ToRs are paired with each other based on their corresponding vpc_peer_link addresses.

```
TORSWITCHINFO:
    CONFIGURE_TORS: True
    SWITCHDETAILS:
        -
        hostname: K09-n9k-a # mandatory for NFVbench
        username: admin # mandatory for NFVbench
        password: <redacted> # mandatory for NFVbench
        ssh_ip: <a.b.c.d> # mandatory for NFVbench
        ssn_num: <xyz>
        vpc_peer_keepalive: <f.g.h.i>
        vpc_domain: <int>
        vpc_peer_port_info: <'eth1/45,eth1/46,eth1/47'>
        vpc_peer_vlan_info: <'NNNN,NNNN-NNNN'>
        br_mgmt_port_info: 'eth1/19'
        br_mgmt_po_info: <'NN'>
        -
        hostname: K09-n9k-b # mandatory for NFVbench
        username: admin # mandatory for NFVbench
        password: <redacted> # mandatory for NFVbench
        ssh_ip: <f.g.h.i> # mandatory for NFVbench
        ssn_num: < xyz>
        vpc_peer_keepalive: < a.b.c.d>
        vpc_domain: <int>
        vpc_peer_port_info: <'eth1/45,eth1/46,eth1/47'>
        vpc_peer_vlan_info: <'NNNN,NNNN-NNNN'>
        br_mgmt_port_info: 'eth1/19'
        br_mgmt_po_info: <'NN'>
```

The attributes for vpc peer vlan info, vpc domain and br_mgmt_po_info have to match across the ToRs, and should only be defined in only two of the TORs, where the management node is hanging off. The attribute for vpc_peer_vlan_info is optional. If it is not specified, it derives a list of VLAN ids from the host/FI facing

Setting up ToR Configurations for B-series and C-series

interfaces and br_mgmt interface. Also, the attribute for ssn_num which represents the chassis serial number is optional.

The chassis serial number can be obtained by executing the following command on each of the ToRs:

```
show license host-id
```

In the case of B-series, Cisco VIM configures the UCSCOMMON section to declare the interface configuration under **tor_info_fi** and **tor_info_fi_redundant** for the FI.



Note ToR names need to match with names provided in the TORSWITCHINFO section.

```
UCSCOMMON:
  ucsm_ip: <p.q.r.s>,
  ucsm_password: <redacted>,
  ucsm_resource_prefix: c43b,
  ucsm_username: admin,
  tor_info_fi: {po: 18, K09-n9k-a: eth1/17, K09-n9k-b: eth1/17}
  tor_info_fi_redundant: {po: 19, K09-n9k-a: eth1/19, K09-n9k-b: eth1/19}
```

In this example of B-Series, tor_info is not declared in the SERVERS section as all connectivity is through the FI (controller, compute, and storage) declared in the UCSCOMMON section. VLANs for the FI facing interfaces are derived from the NETWORK segment ROLES for controller, compute, and storage nodes.

The SERVERS section declares the interface configurations for each of the controller, compute, and storage nodes under **tor_info**.

```
SERVERS:
  controller-1:
    rack_info: {rack_id: rack43X}
    cimc_info: {cimc_ip: <ip_addr>}
    tor_info: {po: 5, B9-TOR-9K-1: eth1/5, B9-TOR-9K-2: eth1/5}
  controller-2:
    rack_info: {rack_id: rack43Y}
    cimc_info: {cimc_ip: <ip_addr>}
    tor_info: {po: 7, B9-TOR-9K-1: eth1/7, B9-TOR-9K-2: eth1/7}
  controller-3:
    rack_info: {rack_id: rack43Z}
    cimc_info: {cimc_ip: <ip_addr>}
    tor_info: {po: 9, B9-TOR-9K-1: eth1/9, B9-TOR-9K-2: eth1/9}
  compute-1:
    rack_info: {rack_id: rack43}
    cimc_info: {cimc_ip: <ip_addr>}
    tor_info: {po: 11, B9-TOR-9K-1: eth1/11, B9-TOR-9K-2: eth1/11}
  compute-2:
    rack_info: {rack_id: rack43}
    cimc_info: {cimc_ip: <ip_addr>}
    tor_info: {po: 13, B9-TOR-9K-1: eth1/13, B9-TOR-9K-2: eth1/13}
  storage-1:
    rack_info: {rack_id: rack43}
    cimc_info: {cimc_ip: <ip_addr>}
    tor_info: {po: 14, B9-TOR-9K-1: eth1/14, B9-TOR-9K-2: eth1/14}
  storage-2:
    rack_info: {rack_id: rack43}
    cimc_info: {cimc_ip: <ip_addr>}
    tor_info: {po: 15, B9-TOR-9K-1: eth1/15, B9-TOR-9K-2: eth1/15}
  storage-3:
    rack_info: {rack_id: rack43}
    cimc_info: {cimc_ip: <ip_addr>}
    tor_info: {po: 16, B9-TOR-9K-1: eth1/16, B9-TOR-9K-2: eth1/16}
```

VLANs for host facing interfaces are derived from NETWORK section based on the server ROLES definition of each of the servers and their corresponding network profile roles assigned for each of the segments.

Server Level Setup_data info for C-series with Intel NIC

When the C-series pod is configured to run in a complete Intel NIC environment, the ToR have an additional configuration that is dp_tor_info section. Control plane and data plane traffic are broken out into two separate interfaces with VLAN limiting applied on each of the interfaces facing the controller and compute nodes.

```
c43b-control-1:
    rack_info: {rack_id: rack43}
    cimc_info: {cimc_ip: <ip_addr>}
        tor_info: {po: 9, K09-n9k-a: 'eth1/9, eth1/12'}
        dp_tor_info: {po: 12, K09-n9k-a: 'eth1/12, eth1/12'}
c43b-compute-1:
    rack_info: {rack_id: rack43}
    cimc_info: {cimc_ip: <ip_addr>}
        tor_info: {po: 10, K09-n9k-a: 'eth1/10, eth1/13'}
        dp_tor_info: {po: 13, K09-n9k-a: 'eth1/13, eth1/13'}
```

Server Level Setup_data info for C-series with Intel NIC with SRIOV

When the C-series pod is configured to support SRIOV with Intel NIC, a third interface is configured to allow SRIOV traffic for the compute nodes. Switchports configured for SRIOV are not placed in a port-channel. VLAN limiting is applied to this interface for all the data plane related VLAN IDs.

```
c43b-compute-1:
    rack_info: {rack_id: rack43}
    cimc_info: {cimc_ip: <ip_addr>}
        tor_info: {po: 10, K09-n9k-a: 'eth1/10, eth1/13'}
        dp_tor_info: {po: 13, K09-n9k-a: 'eth1/13, eth1/13'}
        sriov_tor_info: {K09-n9k-a: eth1/33, K09-n9k-b: eth1/33}
```

Support for Custom Configuration

Custom Configuration is an optional procedure. The setup_data.yaml file has a section called CUSTOM_CONFIG to support custom configuration. Under the CUSTOM_CONFIG section, raw CLI commands can be provided at the global, port channel, and switchport level. CUSTOM_CONFIG is applied at the time of bootstrap and add-interfaces workflow steps.

For example: setup_data.yaml

```
TORSWITCHINFO:
  CONFIGURE_TORS: true
  CUSTOM_CONFIG:
    GLOBAL:
      [<'cli line 1'>,
       <'cli line 2'>,]
    PORTCHANNEL:
      [<'cli line 1'>]
    SWITCHPORT:
      [<'cli line 1'>,
       <'cli line 2'>,]
```



Setting Up ToR Configurations for NCS-5500

Note

In Cisco VIM, the following caveats apply to a Cisco VIM deployment with NCS:

- **BGP:** For a fresh install of Cisco VIM, assure no BGP configuration is present on the NCS, otherwise the peering between the two NCS does not come up properly. Un-configure any existing BGP configuration. If additional BGP complimentary configuration is needed, add it after a successful Cisco VIM install.
- **Segment-Routing:** The global block of Segment Routing IDs have to be pre-defined by the admin. Make sure that the prefix defined within the setup_data.yaml is within the Segment Routing global block range.
- **NCS Interface Naming:** There are a set of different Interface naming variations. We support the following: [Te0/0/0/0, TenGigE0/0/0/0, Gi0/0/0/0, Hu0/0/1/0, HundredGigE 0/0/1/0, FortyGigE0/0/0/0].
- Any manual adjustments to the ISIS, L2VPN sections (on top of the configuration provided by the CVIM automation) causes subsequent Cisco VIM installs to fail.

For a Cisco VIM with NCS-5500 Auto-ToR is a must-have. You can use the Auto-ToR configuration feature to setup NCS-5500. The mercury Cisco VIM setup_data.yaml configuration file is used as an input file for the configuration.

The setup_data.yaml file contains the following three sections:

- **TORSWITCHINFO:** This section provides the general information.
 - **SERVERS section for C-series:** This section provides the information on the switch ports that are connected to the specific nodes. When the micro pod is configured to run in a complete Intel NIC environment with NCS-5500 as the ToR, the SERVER level configurations include tor_info (for control plane) and dp_tor_info (data plane) section. Control plane and data plane traffic are broken out into two separate interfaces with bridge domains applied on each of the control and data interfaces facing each for the controller and compute nodes.
 - **MULTI_SEGMENT_ROUTING_INFO:** This section provides the information related to routing.
- NCS-5500 supports a micro-pod with additional computes running on Intel 710 NICs with no SR-IOV with mechanism driver of VPP.



Note

The current release supports the use of two NCS-5500 within a single pod.

The following snippet shows an example of the mercury setup_data.yaml configuration file for NCS-5500

```
TORSWITCHINFO:
  CONFIGURE_TORS: true    # Mandatory
  TOR_TYPE: NCS-5500     # Mandatory

SWITCHDETAILS:
  -
    hostname: <NCS-5500-1>  # hostname of NCS-5500-1
    username: admin
    password: <ssh_password of NCS-5500-1>
    ssh_ip: <ssh_ip_address of NCS-5500-1>
    vpc_peer_keepalive: <ssh IP address of the peer NCS-5500-2>
```

```

br_mgmt_port_info: <interface of which br_mgmt of management node is hanging of
NCS-5500-1>
    br_mgmt_po_info: <int; bundle Ethernet interface to pxe the management node>
    vpc_peer_port_info: <local interface to which peer NCS-5500 is connected, "," separated,
max of 2 entries' >
        vpc_peer_port_address: <local address with mask for vpc_peer_port_info, "," separated,
max of 2 entries' can have a mask of /31>
        isis_loopback_addr: <local isis loopback interface address without mask> # assumes
/32
    isis_net_entity_title: <isis network_entity_title>
    isis_prefix_sid: <int between 16000-1048575> # has to be unique in the ISIS domain
and depends on the
global segment routing block define by the admin
    -
        hostname: <NCS-5500-2> # hostname of NCS-5500-2
        username: admin
        password: <ssh_password of NCS-5500-2>
        ssh_ip: <ssh_ip_address of NCS-5500-2>
        vpc_peer_keepalive: <ssh IP address of the peer NCS-5500-1>
        br_mgmt_port_info: <interface of which br_mgmt of management node is hanging of
NCS-5500-2>
    br_mgmt_po_info: <int; bundle Ethernet interface to pxe the management node>
    vpc_peer_port_info: <local interface to which peer NCS-5500 is connected>,"," seperated,
max of two entries
        vpc_peer_port_address: <local address with mask for vpc_peer_port_info>,"," seperated,
max of two entries
        isis_loopback_addr: <local isis loopback interface address without mask> # assumes
/32
    isis_net_entity_title: <isis network_entity_title>
    isis_prefix_sid: <int between 16000-1048575> has to be unique in the ISIS domain and
depends on the global segment routing block defined by the admin.
        Not allowed when ESI_PREFIX is defined
    splitter_opt_4_10: 'FortyGigE<C/D/X/Y>,HundredGigE<E/F/A/B>' # Optional for NCS-5500,
only when splitter is needed on per switch basis (that is, the peer switch may or maynot
have the entry)

SERVER SECTION FOR C SERIES:
    a27-fretta-micro-1:
        cimc_info: {cimc_ip: 172.28.121.172}
        dp_tor_info: {NCS-5500-1: TenGigE0/0/0/1, NCS-5500-2: TenGigE0/0/0/1, po: 1}
        hardware_info: {VIC_slot: MLOM}
        rack_info: {rack_id: RackA}
        tor_info: {NCS-5500-1: TenGigE0/0/0/0, NCS-5500-2: TenGigE0/0/0/0, po: 2}
    # Optional
    sriov_tor_info: {NCS-5500-1: TenGigE0/0/0/6, NCS-5500-2: TenGigE0/0/0/6} or
    sriov_tor_info: {NCS-5500-1: 'TenGigE0/0/0/6, TenGigE0/0/0/7', NCS-5500-2: 'TenGigE0/0/0/6,
    TenGigE0/0/0/7' }

    a27-fretta-micro-2:
        cimc_info: {cimc_ip: 172.28.121.174}
        dp_tor_info: {NCS-5500-1: TenGigE0/0/0/3, NCS-5500-2: TenGigE0/0/0/3, po: 3}
        hardware_info: {VIC_slot: MLOM}
        rack_info: {rack_id: RackB}
        tor_info: {NCS-5500-1: TenGigE0/0/0/2, NCS-5500-2: TenGigE0/0/0/2, po: 4}

    a27-fretta-micro-3:
        cimc_info: {cimc_ip: 172.28.121.175}
        dp_tor_info: {NCS-5500-1: TenGigE0/0/0/5, NCS-5500-2: TenGigE0/0/0/5, po: 5}
        hardware_info: {VIC_slot: MLOM}
        rack_info: {rack_id: RackC}
    # optional
    sriov_tor_info: {NCS-5500-1: 'TenGigE0/0/0/8, TenGigE0/0/0/9', NCS-5500-2: 'TenGigE0/0/0/8,
    TenGigE0/0/0/9' }

```

Customization of Cisco NCS 5500 Configurations for Ethernet Segment ID and Route-Target

#Note: if sriov is defined, it need not be present on all servers; However, when present on a given server, the number of SRIOV port need to be 4 and consistent across the servers; Also, please set the INTEL_SRIOV_PHYS_PORTS to 4, when using SRIOV with NCS-5500 as ToR. Please set the value of INTEL_SRIOV_VFS as per the settings of your VNF (see details later for the default values, etc)

```
tor_info: {NCS-5500-1: TenGigE0/0/0/4, NCS-5500-2: TenGigE0/0/0/4, po: 6}
```

```
MULTI_SEGMENT_ROUTING_INFO:
    bgp_as_num: <1 to 65535>
    isis_area_tag: <string>
    loopback_name: <loopback<0-2147483647>>
    api_bundle_id: <1 to 65535>
    api_bridge_domain: <string> #Optional, only needed when br_api of mgmt node is also going via NCS-5500; #this item and api_bundle_id are mutually exclusive
    ext_bridge_domain: <string> # user pre-provisions physical, bundle interface, subinterface and external BD" for external uplink and provides external BD info in the setup_data
```

Customization of Cisco NCS 5500 Configurations for Ethernet Segment ID and Route-Target

Cisco VIM automatically generates the Ethernet Segment Identifier (ESI) for EVPN segments (as defined under each Bundle-Ether interface) and route-targets during Cisco NCS 5500 ToR configuration.

You can set the ESI for EVPN segments only during day-0 configuration. To customize the configuration, define the following in the setup_data as part of the day-0 configuration:

```
ESI_PREFIX: 91.<Pod_number>.<pod_region_number>.00.00.00.00
```

Sample ESI

```
evpn
interface Bundle-Ether<BE#>
  ethernet-segment
    ethernet-segment identifier type 0
91.<Pod_number>.<pod_region_number>.00.00.00.00.00.00.<BE#_in_hex>
```

Example:

```
evpn
interface Bundle-Ether10
  ethernet-segment
    ethernet-segment identifier type 0 91.05.02.00.00.00.00.0a
```

If ESI defined in RFC 7432 is appended with the Bundle ID in hex, it will add up to a total of 9 octects, that is, the ESI_PREFIX must have a max length of 7 octects.

Similar to ESI_PREFIX, Cisco VIM supports custom-defined route-targets for management, storage, and tenant network segment when Cisco NCS 5500 is set as ToR switch. This configuration is optional on per network segment basis, but Cisco VIM generates route-target automatically if not defined. To avail this configuration, the pod administrator must define a rt_suffix and rt_prefix in each network segment as listed below:

```
NETWORKING:
networks:
  - gateway: 5.0.0.1
    pool: [5.0.0.11 to 5.0.0.50]
    segments: [management, provision]
    subnet: 5.0.0.0/24
    vlan_id: 200
```

```

rt_prefix: <Local to POD>
rt_suffix: < Region>:< pod_region_number >

- gateway: 172.25.34.161
  segments: [storage]
  subnet: 172.25.34.160/28
  vlan_id: 2438
rt_prefix: <Local to POD>
rt_suffix: < Region>:< pod_region_number >

```

Resultant Route-Target

<Local to POD>:<Region>< POD number in the region><vlan_id>

Example:

3000:10100214

Each route-target is unique with its respective vlan-id. Route targets associated to tenant vlans are generated by appending each vlan id from TENANT_VLAN_RANGES to the rt_suffix and rt_prefix as defined in the network segments.

Resulting route-targets (“rt_prefix”, plus “:”, plus “rt_suffix”, plus the VLAN ID) must not exceed the 6 octets as per RFC 4360 for the Extended Communities. The maximum value is 8 octets with first 2 being reserved for type information.

NCS Day-0 Configuration (Prior to starting Cisco VIM install)

The following snippets have to be defined on the NCS before starting Cisco VIM installation:

```

SSH:
ssh server v2
ssh server vrf default
ssh server netconf port 831
ssh server netconf vrf default
ssh timeout 60
ssh server rate-limit 600

USERNAME:
username admin
group root-lr
group cisco-support
secret 0 <password>

```



Note For SSH to work generate a key using *crypto key generate rsa*.

Pre-requisites for Segment Routing Global Block and ISIS Prefix

The segment routing configuration has to be predefined by the admin.

The following snippet provides an example:

```

segment-routing
global-block 16000 20000

```

The prefix within the ISIS setup_data.yaml configuration has to be within the global-block IDs. Example:

TORSWITCHINFO:

Pre-requisites for API and External Network Segments with NCS-5500 as TOR

```

CONFIGURE_TORS: true
SWITCHDETAILS:
  - {br_mgmt_po_info: 1, br_mgmt_port_info: TenGigE0/0/0/10, hostname: a25-ncs5500-1-ru30,
    isis_loopback_addr: 10.10.10.10, isis_net_entity_title: 49.0001.1720.1625.5011.00,
    isis_prefix_sid: 16001, password: CTO1234!, ssh_ip: 172.28.123.176, username: admin,
    vpc_peer_keepalive: 172.28.123.177, vpc_peer_port_address:
    '100.100.100.2/29,100.100.101.2/29',
    vpc_peer_port_info: 'HundredGigE0/0/1/4,HundredGigE0/0/1/5'}
  - {br_mgmt_po_info: 1, br_mgmt_port_info: TenGigE0/0/0/10, hostname: a25-ncs5500-2-ru29,
    isis_loopback_addr: 20.20.20.20, isis_net_entity_title: 49.0001.1720.1625.4022.00,
    isis_prefix_sid: 16002, password: CTO1234!, ssh_ip: 172.28.123.177, username: admin,
    vpc_peer_keepalive: 172.28.123.176, vpc_peer_port_address:
    '100.100.100.3/29,100.100.101.3/29',
    vpc_peer_port_info: 'HundredGigE0/0/1/2,HundredGigE0/0/1/3'}
TOR_TYPE: NCS-5500

```

Pre-requisites for API and External Network Segments with NCS-5500 as TOR

Pre-Provision the NCS-5500 with the Bridge domains for API and External network segments. The configured bridge domain names for api and external need to be the same as those defined in setup_data.yaml (api_bridge_domain and ext_bridge_domain) under the MULTI_SEGMENT_ROUTING_INFO section defined above.

A check on each of the NCS-5500 should show the following:

```

RP/0/RP0/CPU0:NCS-5500-2#sh run 12vpn bridge group cvim
  12vpn
  bridge group cvim
    bridge-domain api
12vpn
  bridge group cvim
    bridge-domain external

```

During the deployment of NCS-5500 as TOR, we also support the workloads off the provider network along with the tenant network.

Listed below are some of the assumptions under which this combination works.

- Provider network segment has to be in scope from day-0. Few of the PROVIDER_VLAN_RANGES has to be defined.
- You can always expand the PROVIDER_VLAN_RANGES with additional VLAN range (minimum starting VLAN range is 2)
- The maximum number of PROVIDER_VLAN_RANGES and TENANT_VLAN_RANGES should add up to 200.
- Bridge domain for provider starts with prefix: provider VLANId. They are created manually on the NCS-5500, before the VIM deployment begins; and upstream interfaces are stitched in.

Support and pre-requisites for Provider Network with NCS-Concept

In a deployment of NCS-5500 as TOR, along with the tenant network, we also support provider networks. The following points are key to use provider_networks with a NCS TOR:

- Provider network segment has to be defined on day-0; also, a handful of PROVIDER_VLAN_RANGES has to be defined in the setup_data.yaml.



Note You cannot add it after a Cisco VIM deployment!

- The PROVIDER_VLAN_RANGES can be extended after a Cisco VIM install by running reconfigure with a updated setup_data.yaml (min starting VLAN range is 2, for example PROVIDER_VLAN_RANGES: 3200:3202 (existing range),3204:3206 (newly added range))
- The maximum number of PROVIDER_VLAN_RANGES and TENANT_VLAN_RANGES should not exceed 200.
- Bridge domain for provider starts with prefix: provider<VLANId> and are created manually on the NCS-5500 before VIM deployment begins with necessary upstream interfaces configured accordingly.

Pre-requisites for Provider Network with NCS-5500 as TOR

Provider network support requires the following pre-requisites:

Step 1 Define the network and provider vlan ranges sections in setup_data.yaml.

NETWORKING:

```
- segments: [provider]
  vlan_id: None
PROVIDER_VLAN_RANGES: 127,3406:3409
```

Step 2 Pre-provisioning the NCS with bridge-domains for corresponding VLANs and plumbing the uplink configuration into these bridge-domains.

```
RP/0/RP0/CPU0:NCS-5500-2#sh run 12vpn bridge group cvim
12vpn
  bridge group cvim
    bridge-domain provider127

12vpn
  bridge group cvim
    bridge-domain provider3406

12vpn
  bridge group cvim
    bridge-domain provider3407
```

Note The Cisco VIM Automation will then configure all the host facing subinterfaces for these provider vlans, EVIs and plumb them into each of the pre-provisioned provider bridge-domains.

Note When pre-provisioning bridge-domain, ensure that the BD names follow the naming convention of "provider<vlan-id>".

Intel NIC Support

Cisco VIM supports C-series pod running with either all Intel 710X NICs or Cisco VICs for control and data plane. In the Intel NIC setup, M4 and M5 (Micropod) based pods need to have 2-4 port and 1 or 2 4 port X710 respectively, for control and data plane connectivity. The orchestrator identifies the NIC support based on the following INTEL_NIC_SUPPORT values:

- False-This is the default value. The orchestrator assumes that all the servers have Cisco VIC
- True-The orchestrator assumes that all the servers have Intel NIC.

To define the value, run the following command

```
# INTEL_NIC_SUPPORT: <True or False>
```

The X710 based NIC redundancy is enabled by default for M4-based Intel NIC system, but not for M5-based Intel NIC system. See *Figure 7: UCS C-Series Intel NIC Details* in [UCS C-Series Network Topologies, on page 25](#). To bring in NIC redundancy across the X710s for M5-based Intel NIC systems, define the following global parameter in the setup_data.

```
# NIC_LEVEL_REDUNDANCY: <True or False> # optional and only applies when INTEL_NIC_SUPPORT is set to True
```

A C-series pod, running Intel NIC, also supports SRIOV as an option when defined in a setup_data. To enable SRIOV as an option, define a value in the range 1-32 (32 is maximum number of INTEL_SRIOV_VFS: <integer>).

By default, in the C-series pod running with 4 port Intel 710 card, 1 port (port #c) from each of the Intel NICs are used for SRIOV. However, some VNFs needs additional SRIOV ports to function. To meet the requirement, an additional variable has been introduced in the setup_data.yaml file by which you can include a second port (port d) of the Intel NIC for SRIOV.

To adjust the number of SRIOV ports, set the following option in the setup_data.yaml file:

```
#INTEL_SRIOV_PHYS_PORTS: <2 or 4>
```

The parameter, INTEL_SRIOV_PHYS_PORTS is optional, and if nothing is defined a value of 2 is used. The only values the parameter takes is 2 or 4. For NCS-5500, the only value supported for INTEL_SRIOV_PHYS_PORTS is 4, and has to be defined for SRIOV support on NCS-5500. As the M5 Micropod environment is based on X710 for control and data plane and an additional XL710 or 2 port X710 for SRIOV only INTEL_SRIOV_PHYS_PORTS of 2 is supported.

SRIOV support on a Cisco VIC POD

Cisco VIM supports M4 based C-series pod running with one 2-port Cisco VIC for control plane and two 2-port Intel 520s or two 2-port XL710 for SRIOV (called VIC/NIC deployment). We also support M5 based C-series pod running with one 2-port Cisco VIC for control plane and two 2-port XL710 for SRIOV.

The orchestrator identifies the VIC/NIC support based on the following CISCO_VIC_INTEL_SRIOV values:

- False-This is the default value. The orchestrator assumes that all the servers have Cisco VIC.
- True-The orchestrator assumes that all the servers have Intel NIC.

To define the value, run the following command:

```
# CISCO_VIC_INTEL_SRIOV: <True or False>
```

A C-series M4 pod, running Cisco VIC/Intel NIC (2x520 or 2xXL710), also supports SRIOV on the Intel NIC. To enable SRIOV define a value in the range 1-63 (63 is maximum) (for X520) or 1-32 (32 is maximum for XL710) number of INTEL_SRIOV_VFS: <integer>

By default in the C-series M4 pod running with Cisco VIC and Intel 520/XL710, the control plane runs on the Cisco VIC ports, and all the 4 ports from the 2 Intel 520 NICs or 2 intel XL710 are used for SRIOV.

In C-Series M5 pods running with Cisco VIC and Intel XL710, the control plane runs on the Cisco VIC ports and all the 4 or 8 ports from the 2 intel XL710 are used for SRIOV.

In M5-based VIC/NIC pods, define INTEL_SRIOV_PHYS_PORTS: <4 or 8>, with default value as 4, to indicate the number of ports participating in SRIOV.

In the pods running with CISCO_VIC_INTEL_SRIOV option, some computes can run only with Cisco VIC without SRIOV option if they do not have Intel NIC cards.

Define the following parameter in the setup_data.yaml to setup the card type, in SRIOV (only for M4 based pod).

```
#SRIOV_CARD_TYPE: <X520 or XL710>
```

Compute supports different types of the card. If SRIOV_CARD_TYPE is not provided, Cisco VIM chooses the first 2 slots from all SRIOV compute nodes. If SRIOV_CARD_TYPE is provided, Cisco VIM chooses the first 2 slots matching the target card type from each of the SRIOV compute nodes, so that a match between intent and reality exist.

For Quanta-based pods, the SRIOV slot order starts from the higher slot number, that is, for NUMA, NIC at higher slot has value 0, 2. You can override this, by defining the following as ascending, in which case NIC at higher slot has value of 1, 3.

```
# SRIOV_SLOT_ORDER: <ascending or descending> # Optional, applicable to Quanta-based pods
```



Note

From release Cisco VIM 2.4.4 onwards, some computes have XL710 while others have X520 for SRIOV in an M4 settings. This is achieved by defining the SRIOV_CARD_TYPE at a per compute level (see the SERVERS section of the setup_data in example file). From Cisco VIM 2.4.9 onwards, 40G based M5 computes are supported.

Support of Third-Party Compute in Hybrid Mode (HP DL360 Gen9)

Cisco VIM 2.4 introduces the first third-party compute. The first SKU chosen is HPE ProLiant DL360 Gen9. With this support, the Cisco VIM software is flexible enough to accommodate for other SKUs. In Cisco VIM 2.4, the supported deployment is a full-on pod, with OVS as the mechanism driver, where the management, control, and storage nodes are based on existing Cisco UCS c220/240M4 BOM, and the compute nodes are on HPE ProLiant DL360 Gen9 hardware. From Cisco VIM 2.4.5 onwards, Cisco VIM supports the same HP SKU with both "HP" and "HPE" brand.

To minimize the changes done to the existing orchestration workflow and Insight UI, you can reuse the existing Cisco VIC+NIC combo deployment scenario. This minimizes the changes needed for the hardware topology and the "setup_data.yaml" configuration file. For NIC settings that need to be passed to enable HPE ProLiant DL360 Gen9 third-party compute, see "Intel NIC Support for SRIOV only".

In case of Quanta servers, the support of third-party has been extended to all nodes (servers in control, compute, storage and management role).

The following table shows the port type mapping between Cisco UCS C-series, HPE ProLiant DL360, and Quanta computes:

Port Type	Cisco UCS c220/c240Compute	HPE ProLiant DL360 Gen9 Compute	Quanta Server
Control and Data Plane	M4: MLOM - VIC 1227 M5: MLOM - VIC 1387	FlexLOM - HP Ethernet 10Gb 2-port 560FLR-SFP+ Adapter	OCP 25G 2 port xxv710 based card

Port Type	Cisco UCS c220/c240 Compute	HPE ProLiant DL360 Gen9 Compute	Quanta Server
SRIOV	M4: PCIe - Intel X520-DA2 10 Gbps or Intel XL710 DA2 40 Gbps 2 port NIC M5: PCIe - Intel XL710 DA2 40 Gbps 2 port NIC	PCIe - HP Ethernet 10Gb 2-port 560SFP+ Adapter	PCIe - Intel xxv710 DA2 25 Gbps 2 port NIC
SRIOV	PCIe - Intel X520-DA2 10 Gbps or Intel XL710 DA2 40 Gbps 2 port NIC	PCIe - HP Ethernet 10Gb 2-port 560SFP+ Adapter	PCIe - Intel xxv710 DA2 25 Gbps 2 port NIC

As this deployment do not support Auto-ToR configuration, the TOR switch needs to have Trunk configuration with native VLAN, jumbo MTU, and no LACP suspend-individual on the control and data plane switch ports.

Sample Nexus 9000 port-channel configuration is as follows:

```
interface port-channel130
  description compute-server-hp-1 control and data plane
  switchport mode trunk
  switchport trunk native vlan 201
  spanning-tree port type edge trunk
  spanning-tree bpdufilter enable
  mtu 9216
  no lACP suspend-individual
  vpc 30
!
interface Ethernet1/30
  description compute-server-hp-1 flexlom port 1
  switchport mode trunk
  switchport trunk native vlan 201
  mtu 9216
  channel-group 30 mode active
```

Once the physical connection to the top-of-rack switches and the switch ports' configuration have been completed, enable/add the following additional variables in the VIM's "setup_data.yaml" configuration file:

```
CISCO_VIC_INTEL_SRIOV: True
INTEL_SRIOV_VFS: 63
```

Remote Registry Credentials

```
REGISTRY_USERNAME: '<username>'
REGISTRY_PASSWORD: '<password>'
REGISTRY_EMAIL: '<email@address.com>'
REGISTRY_NAME: <hostname of Cisco VIM software hub> # optional only if Cisco VIM software Hub is used
```

Common CIMC Access Information for C-series POD

```
CIMC-COMMON:
cimc_username: "admin"
cimc_password: "<password>"
```

UCSM Common Access Information for B-series POD

```

UCSMCOMMON:
ucsm_username: "admin"
ucsm_password: <"password">
ucsm_ip: <"a.b.c.d">
ucsm_resource_prefix: <"skull"> # max of 6 chars
ENABLE_UCSM_PLUGIN: <True> #optional; if True, Cisco-UCSM is used, if not defined, default is False
MRAID_CARD: <True or False>

```



Note In Cisco VIM 3.x, UCSM plugin support is not enabled.

Configure Cobbler

```

## Cobbler specific information.
## kickstart: static values as listed below
## cobbler_username: cobbler #username to access cobbler server; static value of Cobbler; not user configurable
## admin_username: root # static value of root; not user configurable
## admin_ssh_keys: This is a generated key which is put on the hosts.
## This is needed for the next install step, using Ansible.
COBBLER:
    pxe_timeout: 45                                # Optional parameter (in minutes); min of 30
    and max of 120, defaults to 45 mins
    cobbler_username: cobbler # cobbler UI user; currently statically mapped to cobbler; not user configurable
    admin_username: root      # cobbler admin user; currently statically mapped to root; not user configurable
    #admin_password_hash has be the output from:
    # python -c "import crypt; print crypt.crypt('<plaintext password>')"
    admin_password_hash: <Please generate the admin pwd hash using the step above; verify the output starts with $6>
    admin_ssh_keys:                                     # Optional parameter
        - ssh-rsa AAAAB3NzaC1yc2EAAAQEAoMrVHLwpDJX8j2DiE55WtJ5NWdiryP5+FjvPEZcjLdtdWaWA7W
        dP6EBAeskmyyU9B8ZJr1uClIN/sT6yD3gw6IkQ73Y6b11kZxu/ZlcUUSNY4RVjsAz52/oLKs6n3wqKnn
        7rQuLGEZDvXnyLbqMoxHdc4PDFWiGXdlg5DIVGigO9KUncPK cisco@cisco-server
    kickstart: # not user configurable, optional
    control: ucs-b-and-c-series.ks
    compute: ucs-b-and-c-series.ks
    block_storage: ucs-b-and-c-series.ks

```

Configure Network

```

NETWORKING:
    domain_name: domain.example.com
#max of 4 NTP servers
    ntp_servers:
        - <1.ntp.example.com>
        - <2.ntp.example2.com >
    or
    ntp_servers: ['2001:c5c0:1234:5678:1002::1', 15.0.0.254]      <== support for IPv6 address
#max of 3 DNS servers
    domain_name_servers:
        - <a.b.c.d>
    or
    domain_name_servers: ['2001:c5c0:1234:5678:1002::5', 15.0.0.1]      <== support for IPv6 address
    http_proxy_server: <a.b.c.d:port> # optional, needed if install is through internet, and the pod is behind a proxy
    https_proxy_server: <a.b.c.d:port> # optional, needed if install is through internet, and the pod is behind a proxy
    admin_source_networks: # optional, host based firewall to white list admin's source IP

```

```
(v4 or v6)
  - 10.0.0.0/8
  - 172.16.0.0/12
  - <"2001:xxxx::/64">
```



Note External access to the management node is made through the IP address configured on the br_api interface. To provide additional security for this connection, the optional **admin_source_networks** parameter is provided. When specified, access to administrator services is only allowed from the IP addresses specified on this list. Use this setting with care, since a misconfiguration can lock out an administrator from accessing the management node through the network. Recovery can be made by logging in through the console and reconfiguring this setting.

Define Network Segments

```
networks:
- # CIMC network section is applicable only for B-series
  vlan_id: <int> # between 1 and 4096
  subnet: <cidr with mask> # true routable network, e.g. 10.30.115.192/28
  gateway: <ip address>
  pool:
    - ip_address_1 to ip_address_2 in the current network segment
  segments:
    - cimc
    -
  vlan_id: <int>
    subnet: <cidr with mask> # true routable network
    gateway: <ipv4_address>

  ipv6_gateway: <ipv6_address> <== required if IPv6 based OpenStack public API is enabled
  ipv6_subnet: <v6 cidr with mask>
  segments:
    - api
    -
  vlan_id: <int> subnet: <cidr/mask>
  gateway: <ipaddress>
  pool:
    # specify the pool range in form of <start_ip> to <end_ip>, IPs without the "to" # is treated
    # as an individual IP and is used for configuring
    - ip_address_1 to ip_address_2 in the current network segment

  # optional, required if managemen_ipv6 is defined at server level
  ipv6_gateway: <ipv6_address>
  ipv6_subnet: <v6 cidr with mask>
  ipv6_pool: ['ipv6_address_1 to ipv6_address_2']

  segments: #management and provisioning are always the same
  - management
  - provision

  # OVS-VLAN requires VLAN-id as "None"
  # LinuxBridge-VXLAN requires valid VLAN-id
  -
  vlan_id: <vlan_id or None> subnet: <v4_cidr w/ mask>
  gateway: <v4 ip address>
  pool:
    - ip_address_1 to ip_address_2 in the current network segment
  segments:
    - tenant
    -
  vlan_id: <vlan_id>
  subnet: <v4_cidr w/ mask>
```

```

gateway: <ipv4_addr>
pool:

- ip_address_1 to ip_address_2 in the current network segment
segments:
- storage

# optional network "external"
-
vlan_id: <int> segments:
- external

# optional network "provider"; None for C-series, vlan range for B-series
-
vlan_id: "<None or 3200-3210>" segments:
- provider

```

**Note**

For PODTYPE: ceph, the storage segment needs to be replaced with segment named “cluster”. Also, for central ceph pod, the only other segment allowed is management/provision.

Define Server Roles

In the Roles section, add the hostname of the servers and their corresponding roles. In case of Micropod, specify the same server names under control, compute, and ceph. Ensure that the number of servers under each role must be three for Micropod. You can optionally expand the Micropod to include additional computes. In the case of HC (Hyperconverged deployment), all storage nodes acts as compute nodes, but not vice-versa.

In the case of Edge pod (to support low latency workloads without persistent storage), specify the same server names under control (total of 3), and compute role (there is no server with storage role). You can optionally expand the edge pod, to include additional computes. The edge pod can connect to a central Ceph cluster via its management network, so that the Ceph cluster offers glance image service.

The central Ceph cluster to which the edge pod is communicating to for the glance image service is called the “ceph” pod-type. For the pod-type “ceph”, specify the same server names under cephcontrol (total of 3), and cephosd role. You can optionally expand the ceph pod, to include additional cephosd nodes.

```

ROLES:    -> for PODTYPE: fullon
control:
- Your-Controller-Server-1-HostName
- Your-Controller-Server-2-HostName
- Your-Controller-Server-3-HostName
compute:
- Your-Compute-Server-1-HostName
- Your-Compute-Server-2-HostName
- ....
- Your-Compute-Server-n-HostName
block_storage:
- Your-Ceph-Server-1-HostName
- Your-Ceph-Server-2-HostName
- Your-Ceph-Server-3-HostName
ROLES:    -> for PODTYPE: micro
control:
- Your-Server-1-HostName
- Your-Server-2-HostName
- Your-Server-3-HostName
compute:
- Your-Server-1-HostName
- Your-Server-2-HostName
- Your-Server-3-HostName

```

```

- Your-Server-4-HostName (optional expansion of computes)
- Your-Server-5-HostName (optional expansion of computes)

block_storage:
- Your-Server-1-HostName
- Your-Server-2-HostName
- Your-Server-3-HostName
object_storage:
networker:

ROLES: -> for PODTYPE: UMHC
control:
- Your-Controller-Server-1-HostName
- Your-Controller-Server-2-HostName
- Your-Controller-Server-3-HostName
compute:
- Your-Compute-Server-1-HostName
- Your-Compute-Server-2-HostName
- Your_HC_Server-1_HostName
- Your_HC_Server-2_HostName
- Your_HC_Server-3_HostName
block_storage:
- Your_HC_Server-1_HostName
- Your_HC_Server-2_HostName
- Your_HC_Server-3_HostName
object_storage:
networker:

ROLES: -> for PODTYPE: edge
control:
- Your-Server-1-HostName
- Your-Server-2-HostName
- Your-Server-3-HostName compute:
- Your-Server-1-HostName
- Your-Server-2-HostName
- Your-Server-3-HostName
- Your-Server-4-HostName (optional expansion of computes)
- Your-Server-5-HostName (optional expansion of computes)

ROLES: -> for PODTYPE: ceph
cephcontrol:
- Your-Server-1-HostName
- Your-Server-2-HostName
- Your-Server-3-HostName cephosd:
- Your-Server-1-HostName
- Your-Server-2-HostName
- Your-Server-3-HostName
- Your-Server-4-HostName (optional expansion of Ceph OSD Nodes)
- Your-Server-5-HostName (optional expansion of Ceph OSD Nodes)
object_storage: networker:

# Server common
# Provide the username (default: root)
SERVER_COMMON:
    server_username: root

# Allow static override value for platform vendor instead of dynamic
# discovery at runtime, optional value.
#
# Allowed values
# CSCO - Cisco Systems Inc
# HPE - Hewlett Packard Enterprise
# QCT - Quanta Cloud Technology Inc

```

```

#
# vendor: <CSCO or QCT>    <= Global level override, all servers
# control:
#   hardware_info:
#     vendor: <CSCO or QCT> <= Role level override, all controls
# compute:
#   hardware_info:
#     vendor: <CSCO, HPE, or QCT> <= Role level override, all computes
# block_storage:
#   hardware_info:
#     vendor: <CSCO or QCT> <= Role level override, all storages

```

**Note**

The maximum length of non-FQDN hostname is 32 characters. The length of Your-Controller-Server-1-HostName hostname is 32 characters in both the ROLES and SERVERS section. The maximum length including the FQDN is 64 characters, where the hostname can only have characters that are in any combination of “A-Za-z0-9-.”, and the TLD is not all numeric. Cisco VIM does not allow “_” in the hostnames.

Cisco VIM introduces a new topology type called Micropod to address solutions that have requirements of high availability, but with limited compute and storage needs. In this deployment model, the control, compute, and storage services reside on each of the three nodes that constitute the pod. Cisco VIM also supports the expansion of the Micropod to accommodate additional compute nodes. Each cloud application can decide the type of pod needed based on their resource (mem, storage consumption) requirements. The Micropod option supports only OVS/VLAN (with Cisco-VIC or Intel 710 NIC) or VPP/VLAN (only on Intel NIC) on a specific BOM.

To enable the Micropod option, update the setup_data as follows:

PODTYPE: micro

Cisco VIM supports the hyper-convergence (UMHC) option of UMHC and NGENAHC. The UMHC option supports only OVS/VLAN with a combination of Cisco-VIC and Intel 520 NIC on a specific BOM, while the NGENAHC option supports only VPP/VLAN with control plane over Cisco-VIC and data plane over 2-port Intel X-710.

To enable the hyper convergence with (UMHC) option, update the setup_data as follows:

PODTYPE: UMHC

To enable the hyper convergence with NGENAHC option, update the setup_data as follows:

PODTYPE: NENAHC

On Quanta server, you can also enable edge cloud functionality for low-latency workloads, for example, vRAN that does not need persistent storage. To enable such deployment, update the setup_data as follows:

PODTYPE: edge

If the edge pod is communicating with a central Ceph cluster that is managed by Cisco VIM, update the setup_data for the respective central-ceph cluster as follows:

PODTYPE: ceph

Define Servers - Rack (C-Series, Quanta) Pod Example

**Note**

The maximum host name length is 32 characters.

```

SERVERS:
Your_Controller_Server-1_HostName:
cimc_info: {'cimc_ip': <IPv4 or IPv6>}
rack_info: {'rack_id': 'RackA'}
#hardware_info: {'VIC_slot': '7'} # optional; only needed if vNICs need to be created on a
specific slot, e.g. slot 7
#management_ip: <static_ip from management pool> #optional, if defined for one server, has
to be defined for all nodes
#cimc username, password at a server level is only needed if it is different from the one
defined in the CIMC-COMMON section
# management_ipv6: <Fixed ipv6 from the management_ipv6 pool>      # <== optional, allow
manual static IPv6 addressing, also if defined management_ip has to be defined
#storage_ip: <Fixed IP from the storage pool> # optional, but if defined for one server,
then it must be defined for all, also if defined management_ip has to be defined

Your_Controller_Server-2_HostName:
cimc_info: {'cimc_ip': '<v4 or v6>', 'cimc_username': 'admin', 'cimc_password': 'abc123'}
rack_info: {'rack_id': 'RackB'}

Your_Controller_Server-3_HostName:
cimc_info: {'cimc_ip': 'v4 or v6'}
rack_info: {'rack_id': 'RackC'}
hardware_info: {'VIC_slot': '7'} # optional only if the user wants a specific VNIC to be
chosen

Your_Storage_or_Compute-1_HostName:
cimc_info: {'cimc_ip': '<v4 or v6>'}
rack_info: {'rack_id': 'RackA'}
hardware_info: {'VIC_slot': '3'} # optional only if the user wants a specific VNIC to be
chosen
VM_HUGHPAGE_PERCENTAGE: <0 - 100> # optional only for compute nodes and when NFV_HOSTS:
ALL and
MECHANISM_DRIVER: openvswitch or ACI
VM_HUGHPAGE_SIZE: <2M or 1G> # optional, only for compute nodes and when NFV_HOSTS is ALL
and MECHANISM_DRIVER is openvswitch or ACI
trusted_vf: <True or False> # optional, only for compute nodes which have in SRIOV
rx_tx_queue_size: <512 or 1024> # optional, only for compute nodes

hardware_info: {'VIC_slot': '<7>', SRIOV_CARD_TYPE: <XL710 or X520>} # VIC_Slot is optional,
defined for location of Cisco VIC

Your_Storage_HostName:
cimc_info: {'cimc_ip': 'v4 or v6'} rack_info: {'rack_id': 'RackA'}
hardware_info: {osd_disk_type: <HDD or SSD>} # optional only the pod is multi-backend ceph,
and a minimum of three storage servers should be available for each backend type.

```

**Note**

SRIOV_CARD_TYPE option is valid only when CISCO_VIC_INTEL_SRIOV is True; and can be defined at per compute level for M4 pod. If it is not defined at a per compute level, the global value is taken for that compute. If not defined at the compute nor at the global level, the default of X520 is set. The compute can be standalone or hyper-converged node.

**Note**

Cisco VIM installation requires that controller node Rack IDs be unique. The intent is to indicate the physical rack location so that physical redundancy is provided within the controllers. If controller nodes are installed all in the same rack, you must assign a unique rack ID to prepare for future Cisco NFVI releases that include rack redundancy. However, compute and storage nodes do not have rack ID restrictions.



Note For Central Ceph cluster, swap the “storage_ip” with “cluster_ip”.



Define Servers - B-Series Pod Example



Note For UCS B-Series servers, the maximum host name length is 16 characters.

```
SERVERS:  
Your_Controller_Server-1_HostName:  
rack_info: {'rack_id': 'rack2'}  
ucsm_info: {'server_type': 'blade',  
'chassis_id': 1,  
'blade_id' : 1}  
Your_Controller_Server-2_HostName:  
rack_info: {'rack_id': 'rack3'}  
ucsm_info: {'server_type': 'blade',  
'chassis_id': 2,  
'blade_id' : 1}  
Your_Controller_Server-3_HostName:  
rack_info: {'rack_id': 'rack4'}  
ucsm_info: {'server_type': 'blade',  
'chassis_id': 2,  
'blade_id' : 4}  
#management_ip: <static_ip from management pool> #optional, if defined for one server, it  
must be defined for all nodes  
#storage_ip: <Fixed ip from the storage pool> # optional, but if defined for one server,  
then it must be defined for all,  
also if defined management_ip has to be defined  
Your_Compute-1_HostName:  
rack_info: {'rack_id': 'rack2'}  
ucsm_info: {'server_type': 'blade',  
'chassis_id': 2,  
'blade_id' : 2}  
.. add more computes as needed  
  
Your_Storage-1_HostName:  
rack_info: {'rack_id': 'rack2'}  
ucsm_info: {'server_type': 'rack',  
'rack-unit_id': 1}  
Your_Storage-2_HostName:  
rack_info: {'rack_id': 'rack3'}  
ucsm_info: {'server_type': 'rack',  
'rack-unit_id': 2}  
Your_Storage-3_HostName:  
rack_info: {'rack_id': 'rack4'}  
ucsm_info: {'server_type': 'rack',  
'rack-unit_id': 3}  
  
# max # of chassis id: 24  
# max # of blade id: 8  
#max # of rack-unit_id: 96
```



Note Cisco VIM requires the controller Rack IDs to be unique to indicate the physical rack location and provide physical redundancy for controllers. If your controllers are all in the same rack, you must still assign a unique rack ID to the controllers to provide for future rack redundancy. Compute and storage nodes have no Rack ID restrictions.

Setting Up Cisco VIM OpenStack Configuration

The following sections provide examples of Cisco VIM OpenStack configuration in the setup_data.yaml file.

OpenStack Admin Credentials

```
ADMIN_USER: <admin>
ADMIN_TENANT_NAME: <admin tenant>
```

OpenStack HAProxy and Virtual Router Redundancy Protocol Configuration

```
external_lb_vip_address: An externally routable ipv4 address in API network
external_lb_vip_ipv6_address: An externally routable ipv6 address in API network
VIRTUAL_ROUTER_ID: vrrp_router_id #eg: 49 (range of 1-255)
internal_lb_vip_address: <Internal IP address on mgmt network>
internal_lb_vip_ipv6_address: <Internal IPv6 address on mgmt network> # optional, only for
dual stack environment
```

OpenStack DNS Name Configuration

For web and REST interfaces, names are commonly used instead of IP addresses. You can set the optional external_lb_vip_fqdn parameter to assign a name that resolves to the external_lb_vip_address. You must configure the services to ensure the name and address match. Resolution can be made through DNS and the Linux /etc/hosts files, or through other options supported on your hosts. The Cisco VIM installer adds an entry to /etc/hosts on the management and other Cisco NFVI nodes to ensure that this resolution can be made from within the pod. You must ensure the resolution can be made from any desired host outside the pod.

```
external_lb_vip_fqdn: host or DNS name matching external_lb_vip_address
```

OpenStack TLS and HTTPS Configuration

Enabling TLS is important to ensure the Cisco VIM network is secure. TLS encrypts and authenticates communication to the cloud endpoints. When TLS is enabled, two additional pieces of information must be provided to the installer: haproxy.pem and haproxy-ca-crt. These must be placed in the ~/installer-xxxx/openstack-configs directory.

haproxy.pem is the server side certificate file in PEM format. It must include the server certificate, any intermediate certificates, and the private key for the server. The common name of the certificate must match the external_lb_vip_address and/or the external_lb_vip_fqdn as configured in the setup_data.yaml file. haproxy-ca.crt is the certificate of the trusted certificate authority that signed the server side.

For production clouds, these certificates are provided by a trusted third-party CA according to your company IT policy. For test or evaluation clouds, self-signed certificates can be used quickly enable TLS. For convenience, the installer includes a script that creates and install self-signed certificates



Note Do not use the certificates generated by this tool for production. They are for test purposes only.

To use this tool, make the following changes to the setup data file, then run the tool:

```
external_lb_vip_address: <IP address on external network>
external_lb_vip_tls: True
external_lb_vip_fqdn: host or DNS name matching external_lb_vip_address (if FQDN is needed)
```

To run the tool, from the /working_dir/ directory, execute #./tools/tls_cert_gen.sh -f openstack-configs/setup_data.yaml.

OpenStack Glance Configuration with Dedicated Ceph/Netapp

For OpenStack Glance (OpenStack image service), the dedicated Ceph object storage configuration is shown below. Do not change it. The Ceph and Glance keys are generated during the Ceph installation step, so you do not need to specify the keys in setup_data.yaml file.

```
STORE_BACKEND: ceph/netapp #supported as 'ceph' for ceph backend store;and netapp for netapp backend
```

CPU Allocation for Ceph in Hyper-converged or Micropod systems

As the storage node is shared with other node types (e.g. compute for Hyper-converged and control and compute for micropod), there are deployments where the number of CPU cores allocated to the Ceph role needs to be higher than the default value of 2. From release Cisco VIM 2.4.2 onwards, the option CEPH OSD_RESERVED_PCORES is available on fresh install only in the case of Micropod and hyperconverged pods.

This option is set using the following commands in setup_data, where the value can range between 2 and 12.

```
# Number of cores associated to CEPH-OSD in a micro, UMHC or NGNENAHC deployment,
# default value if not defined is 2
#CEPH OSD_RESERVED_PCORES: <2 - 12>
```

CEPH Placement Group Info (Optional)

If you need to change the default percentages for placement group calculation use this section to indicate the amount of data you expect in cinder/glance/nova. For NOVA_BOOT_FROM local, provide the values for cinder and glance. Additionally, for NOVA_BOOT_FROM ceph provide nova_percentage_data for ephemeral data. All Percentages need to add up to 100. If no information is provided, the code defaults to 60% cinder and 40% glance for NOVA_BOOT_FROM local. Similarly, if no information is provided the code defaults to 40% cinder, 30% glance and 30% nova ephemeral for NOVA_BOOT_FROM ceph. You cannot change these values after deployment via update or reconfiguration.

```
# For NOVA_BOOT_FROM local
# CEPH_PG_INFO: {cinder_percentage_data: x, glance_percentage_data: y}
# where x and y are integers and must add up to 100

# For NOVA_BOOT_FROM Ceph
# CEPH_PG_INFO: {cinder_percentage_data: x, glance_percentage_data: y,
# nova_percentage_data: z}
# where x, y and z are integers and must add up to 100
```

OpenStack Glance Configuration

```
STORE_BACKEND: <ceph or netapp based on backend storage>
```

OpenStack Cinder Configuration with Dedicated Ceph/Netapp

For OpenStack Cinder (OpenStack storage service), the dedicated Ceph object storage configuration is shown below. Do not change it. The Ceph and Cinder keys are generated during the Ceph installation step, so you do not need to specify the keys in setup_data.yaml file. Use the **vgs** command to check your volume groups available on your controller nodes. The controller nodes run the Cinder volume containers and hold the volume groups for use by Cinder. If you have available disks and want to create a new volume group for Cinder use the **vgcreate** command.

```
VOLUME_DRIVER: ceph/netapp
```

OpenStack Settings on PODTYPE: Ceph for Glance Image service

Following are the examples for central_ceph setup_data details:

```
STORE_BACKEND: 'ceph'
VOLUME_DRIVER: 'ceph'
```

OpenStack Settings on PODTYPE: Edge for Glance Image service

For the edge pod installation to be successful, the central Ceph cluster with which it will communicate for glance image service must be up and running. For the edge pod to communicate with the central Ceph cluster, the following configurations are needed:

```
MON_HOSTS: <3 IPv4 or IPv6 addresses, of the cephcontrol servers in the central ceph cluster>
MON_MEMBERS: <3 IPv4 or IPv6 addresses, of the cephcontrol servers in the central ceph
cluster>
CLUSTER_ID: <ceph_cluster_id>
# to fetch the CLUSTER_ID of the central ceph cluster, ssh to the management node of the
"ceph" pod, and execute the following:
# cat /root/openstack-configs/ceph/fetch/ceph_cluster_uuid.conf to get the CLUSTER_ID
GLANCE_RBD_POOL: images
GLANCE_CLIENT_KEY: <key_info>
# to fetch the GLANCE_CLIENT_KEY, ssh to the management node of the "ceph" pod, and execute
the following:
# cd /root/openstack-configs/ceph/fetch/
# ls to get the UUID
# cd /root/openstack-configs/ceph/fetch/<UUID>/
# cat etc/ceph/ceph.client.glance.keyring
```

OpenStack Nova Configuration

To reduce the boot time, the NOVA_BOOT_FROM parameter is set to local for Cisco VIM. While this reduces the boot time, it does not provide Ceph back end redundancy. For typical NFVI workloads, you must not enable this option (it will default to local). To overwrite it, you can set NOVA_BOOT_FROM to **ceph**. This is applicable only when the backend is ceph. For Netapp, no entry for this parameter is allowed.

```
# Nova boot from CEPH/local
NOVA_BOOT_FROM: <ceph or local> #optional, if not defined will default to local
```

OpenStack Neutron Configuration

OpenStack Neutron configuration is shown below.

```
# ML2 Conf - reference implementation of OVS/VLAN

MECHANISM_DRIVERS: openvswitch
TENANT_NETWORK_TYPES: "VLAN"
# VLAN ranges can be a single continuous range or comma separated discontinuous range
TENANT_VLAN_RANGES: 3001:3100,3350:3400
# Jumbo MTU functionality.
# ENABLE_JUMBO_FRAMES: True

# for Provider networks, just specifying the provider in the segments under
# the NETWORKING section is enough. Use phys_prov as physical_network name when creating a
provider network
```

Ensure that you include the PROVIDER_VLAN_RANGES information in the setup_data as given in the following syntax:

PROVIDER_VLAN_RANGES: <a,b:c,d:e>, where the VLAN ranges can be a continuous range or comma separated discontinuous range.



Note When creating an external or provider network, use physical_network=phys_ext (need to be specified) or physical_network=phys_prov (need to be specified), respectively.

The JUMBO_MTU functionality is available only for OVS over VLAN in a UCS B-Series pod. In a VLAN setup, by default the MTU size is set to 1500 (1450 for VXLAN) and 8972 bytes. When JUMBO_MTU is enabled (with 28 bytes left for the header), the VLAN MTU is 9000 and VXLAN is 8950.

Control and Data Plane Testing in Cisco VIM

Cisco VIM offers an integrated test to validate the control and data plane sanity of the cloud. Virtual Machine Through Put (VMTP), an optional test is available to check the Layer 2 and Layer 3 data plane traffic between Cisco NFVI compute nodes. VMTP performs ping connectivity, round trip time measurement (latency), and TCP/UDP throughput measurement for the following Cisco NFVI east to west VM-to-VM flows:

- Same network (private fixed IP, flow number 1).
- Different network using fixed IP (same as intra-tenant L3 fixed IP, flow number 2).
- Different network using floating IP and NAT (same as floating IP inter-tenant L3, flow number 3.)

To enable VMTP for basic Cisco VIM installation, update the setup_data with the following commands:

```

VMTP_VALIDATION:
EXT_NET:          # Only for V4 with External network with floating IP, min of 5 cont. IP
    NET_NAME: <name of external network>
    NET_SUBNET: <external cidr>
    NET_IP_START: <floating ip start>
    NET_IP_END: <floating ip end>
    NET_GATEWAY: <external net gateway>
    DNS_SERVER: <dns server for external net>

PROV_NET:          # Either for V4 or V6 for Provider network
    NET_NAME: <provider network name>
    NET_SUBNET: <provider net cidr>
    NET_IP_START: <starting IP for provider net>
    NET_IP_END: <end IP for provider net>
    NET_GATEWAY: <provider net gateway>
    DNS_SERVER: <dns server for provider net>
    SEGMENTATION_ID: <segmentation id for provider net>      # Needs to match a vlan defined
under PROVIDER_VLAN_RANGES
        IPV6_MODE: <"slaac" or "dhcpv6-stateless" or "dhcpv6-stateful"> # only for IPv6;
        VNIC_TYPE: <"direct" or normal>      # use value of direct for SRIOV, default is over
virtio (value of normal)
        PHYSNET_NAME: <physnet_name>    # needed for SRIOV, entry has to be of the name:
phys_sriov0, or phys_sriov1, ... phys_sriovn, where n is total num of SRIOV port-1

```

Optional Services in Cisco VIM

Cisco VIM supports the installation of optional services, namely, ceilometer, ironic, and load balance as a service (lbass). OpenStack Heat is an orchestration service that allows you to spin up multiple instances, logical networks, and other cloud services in an automated fashion. To enable Heat, add the following in the setup_data.yaml.

```
# Optional Services:
OPTIONAL_SERVICE_LIST:
- heat
```

To disable Heat, remove the optional services section from the setup_data.yaml file. The optional services support provides an infrastructure to support additional services in the future.



Note Auto-scaling is not supported in Cisco VIM.

Ceilometer Support in Cisco VIM

The reference implementation of ceilometer is available from Cisco VIM 3.0 onwards. The ‘ceilometer’ service can be brought in as a day-0 option for fullon pod. To enable this service, update the setup_data with the following:

```
# Optional Services:
OPTIONAL_SERVICE_LIST:
- ceilometer
```



Note Ceilometer is enabled when the pod type is fullon.

LBASS Support

The reference implementation of LBASS is available from Cisco VIM 3.2.2 onwards. The **lbass** service can be brought in as a day-0 option. To enable this service, update the setup_data with the following:

```
# Optional Services:
OPTIONAL_SERVICE_LIST:
- lbass
```

Ironic Support in Cisco VIM

The reference implementation of ironic is available in Cisco VIM. The ironic service can be brought in day-0 or as a reconfigure option. Once enabled, it cannot be disabled. Ironic support is only available with Cisco UCS C baremetal servers and when Cisco VIM is deployed with OVS as the mechanism driver. The ironic interface to be used on the baremetal servers for openstack can be either an MLOM interface, an Intel NIC, or the onboard 1G LOM port. Ironic supports only the configuration of a single interface on the baremetal server.

You must have one separate network segment that is used for ironic_management and ironic inspector. The inspector is a service used to automate the creation of the openstack baremetal port with switch interface, for example, eth 1/39 and MAC address information of both the switch MAC and server interface MAC apart from automatically adding the deploy image information to the ironic node.

You must ensure that the ironic management, ironic inspector, Cisco VIM management, and ironic CIMC networks are routed to each other.

The Cisco VIM management must be able to reach:

- Ironic management network and vice-versa

- CIMC network of the ironic nodes so that the Cisco VIM controller servers can directly reach the CIMC IP of the ironic servers.

To enable network reachability:

- All three networks such as Cisco VIM management, Ironic management and CIMC must be private networks with SVI interfaces on the ToR.
- Routed network must be deployed for all three network segments. In this case, the need for SVI interfaces on the ToR is eliminated.



Note

It is mandatory to include the ironic management/ironic inspector VLANs on the ToR interfaces connected to all the mercury controller servers. This must be manually configured at present.

While deploying ironic, follow the below steps before installing VIM:

- Create a separate ironic_inventory.yaml with CIMC/ IPMI details of the servers to be used as ironic baremetals. For example,
`/root/installer-XXX/openstack-configs/ironic_inventory.yaml`.
- Save this file with your ironic server details in
`/root/installer-XXX/openstack-configs/ironic_inventory.yaml`
- Specify the ironic management/ironic inspector VLAN in all control interfaces of the mercury controller servers. This is essential to perform ironic introspection so as to transfer the images from the controller to the baremetal server.
- If ironic is deployed in a Nexus mode of ToR, ensure that no existing configuration exists on the interface of the ToR connected to the baremetal. The interface is in ACCESS mode. Only the ironic inspector VLAN needs to be set as the access VLAN.
- If ironic is deployed in an ACI mode testbed, you must ensure that ironic management network VLAN and all the tenant VLANs from setup_data are configured on the interface of the ToR connected to the baremetal the ironic inspector VLAN. The interface is in TRUNK mode. You need to set the ironic inspector network as the native VLAN.
- Verify whether the following are done in the baremetal server CIMC before proceeding
 - Check if IPMI connections are allowed over LAN
 - In BIOS configured Boot order, only pxeboot is present and available as the first option.
 - PXE is enabled in VNIC adapters, if VNICs are used as the interface for ironic. If deploying on an Intel NIC or the onboard LOM interface, this step is not needed..
 - Set the VLAN mode on the VNIC being used as TRUNK, if VNICs are used as the interface for ironic. This step is not required for deployment on an Intel NIC or the onboard LOM interface.
 - Turn ON the baremetal node, to have access to all parameters of CIMC. VIM installer will verify the node at Step 1.
 - Disable LLDP on Cisco VIC Adaptor of all the servers used for ironic by doing the following and then reboot the server:

```
sh admin@X.X.X.X (CIMC IP)
C240-FCH1832V1HW# scope chassis
C240-FCH1832V1HW /chassis # show adapter
C240-FCH1832V1HW /chassis # scope adapter <PCI slot>
C240-FCH1832V1HW /chassis/adapter # set lldp disabled
C240-FCH1832V1HW# commit
C240-FCH1832V1HW /chassis/adapter # show detail <To Verify LLDP is disabled>
```

To enable this service, update the setup_data with the following:

```
# Optional Services:
OPTIONAL_SERVICE_LIST:
  - ironic

IRONIC:
  IRONIC_SWITCHDETAILS: # list of switches off which the ironic servers are hanging. This
  is mainly used to provide ironic switch details to neutron
  - {hostname: <switch_name>, password: <password>, ssh_ip: <ssh_ip>, username:
  <switch_admin_username>, switch_type: <"Nexus", "ACI", or "BypassNeutron">}

NETWORKING:
.....
  - gateway: <gateway_information> # Mandatory if ironic is present
  pool: [<ip_start1 to ip_end1>]
  segments: [ironic]
  subnet: <subnet with/mask>
  vlan_id: <unique vlan id across the pod>
  inspector_pool: [ip_add_1 to ip_add_2, ip_add_3 to ip_add_4, ip_add_5 to ip_add_6] (# of entry pool : 3, same network as ironic but doesn't overlap with the pool of IPs defined in the ironic segment)
  # alternate format for pool (# of entry pool : 3)
  - ip_add_1 to ip_add_2
  - ip_add_3 to ip_add_4
  - ip_add_5 to ip_add_6
```

Container Support in Cisco VIM

Cisco VIM leverages the offering of Cisco Container Platform and hosts it as an application to host container workloads. To host Cisco Container Platform which is a kubernetes based container platform, the orchestrator creates a common OpenStack tenant and deploys the Cisco Container Platform control plane within it. Optionally, it can also create a tenant cluster if needed. The Kubernetes clusters deployed are multi master with three masters and N workers. The Cisco Container Platform control plane itself consists of three masters and three workers. The master and worker nodes run as virtual machines.

Following assumptions are set to enable Cisco Container Platform on Cisco VIM:

- Cisco VIM is up and running on the pod.
- Cisco Container Platform can be brought in during day-0 or day-2 as part of reconfiguration.
- Cisco Container Platform workload will run as an OpenStack Tenant: Cisco Container Platform control plane VMs and all tenant cluster VMs are expected to run.
- User in tenant: A user must pre-exist for the OpenStack tenant where Cisco Container Platform can run.
- The virtual IP (VIP) for the Cisco Container Platform Installer can either be a provider or floating IP in the tenant.

- SSH key in tenant: Key will have ssh access to Cisco Container Platform control plane VMs.
- LBAAS feature must be enabled in Cisco VIM.
- Cisco Container Platform can run on a provider or floating IP based environment:
 - If Cisco Container Platform is running of a tenant network paradigm, networking is defined as managed L3 connectivity and public external network is leveraged to host floating IPs that need a minimum of 10 IPs.
 - If Cisco Container Platform is running of a provider network paradigm, networking is defined as lacking managed L3 connectivity and the public provider network is leveraged for direct attachment to hosts and load balancers that need a minimum of 20 IPs.

Following are the steps to enable Cisco Container Platform over Cisco VIM:

1. Enable LBAAS in Cisco VIM.
2. Use the installer in the VM of Cisco VIM to create a new OpenStack tenant and to deploy Cisco Container Platform control plane so that the Cisco Container Platform control plane brings up the Cisco Container Platform endpoint which is exposed as an external Kubernetes service using neutron LBAAS. The control plane API can then be reached to create multiple tenant clusters.
3. Use Calico network plugin as an overlay. The VMs communicate with each other using VM node IP address through neutron networking. There are no requirements out of the fabric to run BGP, as the Calico packet is encapsulated within a standard neutron packet. Cisco Container Platform clusters does not have monitoring and Cisco Container Platform Add-ons.
4. For persistent storage, use Cinder as storage provider. This is dynamic and created automatically, when a kubernetes-based workload request is received for a persistent volume.

Prerequisites for Cisco Container Platform installation:

Ensure that lbaas is enabled as part of "optional_service_list". If not, run the following command:

```
ciscovm reconfigure --setupfile <path to new setupfile containing lbass>;
for example:
OPTIONAL_SERVICE_LIST: [heat, lbaas]
```

Cisco Container Platform Installation:

To install Cisco Container Platform on Cisco VIM, follow the below steps:

1. Generate a SSH key of ecdsa type using the below command:

```
# ssh-keygen -f /root/ecdsa-key -t ecdsa -b 521 (Press Enter till keys are generated)
```

2. Download the tenant and installer images from the following link:

```
https://software.cisco.com/download/redirect?config=3d26225bd6d385d843a8bcbfa1f43146
```

```
Tenant Image Name: ccp-tenant-image-1.13.5-ubuntu18-4.0.1.qcow2.tar.gz
CCP Installer Image Name: kcp-vm-4.0.1.qcow2.tar.gz
```

3. Establish the networking type (tenant or provider) on the basis of which the CCP_DEPLOYMENT section is defined in setup_data:

```
CCP_DEPLOYMENT: # Parameters for CCP Deployment Optional services LBAAS mandatory
    CCP_CONTROL: # Installer creates a new tenant in Openstack based on below information
    and set all quotas in that tenant
        UI_PASSWORD: <UI_PASSWORD> # password for CCP UI (required)
        ccp_subnet_cidr: <ip_address/mask> # subnet to create to deploy CCP control plane
```

```
(required for tenant network should be removed for provider network)
    installer_subnet_cidr: <ip_address/mask> # subnet to create for bootstrap installer
(required for tenant network should be removed for provider network)
    installer_subnet_gw: <ip_address> # gateway to use for bootstrap installer (required
for tenant network should be removed for provider network)
    password: <password> # password for the Openstack tenant (required)
    private_key: <absolute path for ed25519 based key> # private key to be used to SSH
to VM must be ed25519 (required)
    project_name: <tenant_name> # Tenant name to create for CCP control Plane installer
will create this Openstack tenant (required)
    public_key: <absolute path for ed25519 based public key> # Public key for CCP VMs,
e.g. /root/ecdsa-key.pub
    username: <string> # username for the CCP control plane tenant (required)
    CCP_INSTALLER_IMAGE: <qcow2 absolute image path> # Pointer to the CCP Installer image
(required)
    CCP_TENANT: # Test only option not supported in production to create demo tenant cluster
using CCP API (Optional NA in production)
    password: <password> # password for tenant (required)
    project_name: <project_name> # tenant name to create in Openstack to host tenant
cluster (required)
    username: <username> # username for openstack tenant (required)
    workers: 1 # no of kubernetes workers in tenant cluster (required)
    subnet_cidr: <ip_address/mask> # tenant subnet CIDR
    CCP_TENANT_IMAGE: <qcow2 based abs path of tenant cluster image> # Pointer to CCP
tenant cluster image (required)
    DNS_SERVER: [list of IPv4 based DNS servers] # DNS server to be reachable from cloud
(required)
    KUBE_VERSION: <x.y.z> # Version of Kubernetes to install (required) normally can be
deciphered from tenant image name; e.g. 2.3.4
    NETWORK_TYPE: <tenant or provider> # Network Type valid values provider or tenant
network (required)
    POD_CIDR: <ip_address/mask> # POD CIDR to use for calico network optional if not to
be changed (optional)
    PUBLIC_NETWORK_UUID: <UUID of Openstack external network or provider network; Fake
UUID incase of Day-0 Install > (optional initially but mandatory when ccp is being run)

    CCP_FLAVOR: <flavor> optional initially, but mandatory when NFV_HOSTS is enabled during
ccp install
```

4. To enable Cisco Container Platform on Cisco VIM on day-0, update the setup_data with CCP_DEPLOYMENT section and execute the standard ciscovim install command. After the installation, update the PUBLIC_NETWORK_UUID (from the output of "neutron net-list" after sourcing openrc from /root/openstack-configs) and CCP_FLAVOR.

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml /root/MyDir/
[root@mgmt1 ~]# # update the setup_data to update the PUBLIC_NETWORK_UUID and CCP_FLAVOR
information
[root@mgmt1 ~]# cd /root/MyDir/
[root@mgmt1 ~]# vi setup_data.yaml
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ccp install --setupfile /root/MyDir/setup_data.yaml
```

5. You can initiate the installation of Cisco Container Platform using the following command:

```
# ciscovim ccp install --setupfile <path_to_setup_file>
```

6. After the installation, execute the following command to see Cisco Container Platform control plane login URL information:

```
# ciscovim ccp show
```

**Note**

For information on Cisco Container Platform installation, see

<https://www.cisco.com/c/en/us/support/cloud-systems-management/container-platform/products-installation-guides-list.html>

**Note**

For details on verification and management of Cisco Container Platform cluster, see *Cisco Virtualized Infrastructure Manager Administrator Guide*

LDAP support in Cisco VIM

To continue enhancing the security portfolio and multi-tenancy with the use of domains, Keystone v3 support is now default in Cisco VIM 3.0.0.

With the introduction of Keystone v3, the OpenStack service authentication can now be delegated to an external LDAP server. In Cisco VIM, this feature has been introduced optionally if the authorization is done by Keystone v3.

An important pre-requisite for enabling LDAP integration is that the LDAP endpoint has to be reachable from all the Controller nodes that run OpenStack Keystone Identity Service.

To benefit LDAP support with Keystone v3 feature, the setup_data needs to be augmented with the following information during the installation of the pod.

LDAP:

```

domain: <Domain specific name>
user_objectclass: <objectClass for Users> # e.g organizationalPerson
group_objectclass: <objectClass for Groups> # e.g. groupOfNames
user_tree_dn: '<DN tree for Users>' # e.g. 'ou=Users,dc=cisco,dc=com'
group_tree_dn: '<DN tree for Groups>' # e.g. 'ou=Groups,dc=cisco,dc=com'
suffix: '<suffix for DN>' # e.g. 'dc=cisco,dc=com'
url: '<ldap:// host:port>' # e.g. 'ldap://172.26.233.104:389'
user: '<DN of bind user>' # e.g. 'dc=admin,dc=cisco,dc=com'
password: <password> # e.g. password of bind user
user_filter: '(memberOf=CN=os-users,OU=OS-Groups,DC=mercury,DC=local)' # Optional
user_id_attribute: sAMAccountName
user_name_attribute: sAMAccountName
user_mail_attribute: mail # Optional
group_name_attribute: sAMAccountName
group_filter: '(&(objectClass=group) (|(cn=server-ops)(cn=admins)))' # Optional
group_member_attribute: memberUid # Optional
group_id_attribute: gidNumber # Optional
group_members_are_ids: True # Optional
chase_referrals: <True or False> # Optional

```

Conditions for LDAP user and password parameters:

- 1 – Can be optional (for group option).
- 2 – It must be mutually inclusive.
- 3 – If defined, it cannot be empty.



Note The values for the parameters may differ based on the Directory Service provider. For Example: OpenLDAP or Microsoft Active Directory.

Integrating identity with LDAP over TLS: The automation supports keystone integration with LDAP over TLS. In order to enable TLS, the CA root certificate must be presented as part of the /root/openstack-configs/haproxy-ca.crt file. The url parameter within the LDAP stanza must be set to *ldaps*.

url parameter supports the following formats

```
url: '<ldaps | ldap>://<FQDN | IP-Address>:[port]'
```

The protocol can be ldap for non-ssl OR ldaps if TLS is to be enabled

The ldap host can be a fully-qualified domain name (FQDN) or an IP Address depending on how the SSL certificates are generated.

The port number is optional and if it is not provided it is assumed that the ldap services are running on the default ports For Example:389 for non-ssl and 636 for ssl. However, if these ports are not the default ports, then the non-standard port numbers must be provided.

Support for Anonymous LDAP Bind

The automation provides support for anonymous simple bind where the LDAP configuration for a “user” representing the **bindDN** and **password** is optional and may not be provided.



Note Ensure that the LDAP server allows the clients to bind and search anonymously.

OpenStack Object Storage Integration with Cisco VIM

Cisco VIM supports automated integration with a customer-managed object storage solution. The integration points reside primarily in the OpenStack Identity (Keystone) component of Cisco VIM. In the current release, this integration is restricted to Keystone v2 only. It currently integrates with SwiftStack as the choice of object storage solution. The deployment assumes a customer-managed SwiftStack solution. Installation of the SwiftStack Controller/PACO cluster is out of scope of this document and customer has to reach out to the SwiftStack team for license and installation details. While OpenStack can support multiple endpoints for a given object-store service, the current setup in the context of automation supports a single Keystone object-store service per SwiftStack PACO cluster endpoint.

The current automation uses the admin role for authentication and authorization of SwiftStack users between the Keystone SwiftStack tenant and SwiftStack account.

Pre-requisites

Since it is a customer-managed deployment model, the minimum pre-requisite is to have a SwiftStack controller, Cluster deployed with appropriate PAC (Proxy/Account/Container) and Object configured ahead of time. The swift endpoint of the PAC outward facing ip address, the corresponding admin user, password and service tenant information is known at the time of configuring Keystone integration. The networking has to be configured in such a way that the PAC outward facing ip address and the POD API network can talk to each other. Also the Keystone Auth and Keystone Auth Token middleware are pre-configure in SwiftStack (see the steps in subsequent section).

In order for Horizon and Cinder Backup Service to talk to the SwiftStack endpoints, it is necessary for the OpenStack controllers to have network reachability to the SwiftStack API endpoints.

Keystone Configuration Requirements in SwiftStack

Configuring Keystone Authorization: From the SwiftStack controller, select the **Cluster > Manage > Middleware > Keystone Auth** option.


Note

reseller_prefix enables the Keystone Auth middleware invocation at the time of authentication.

Figure 40: Configuring Keystone

The screenshot shows the Keystone Auth configuration page. At the top, there is a breadcrumb navigation: Home / Clusters / Manage mercury-dev / Manage Middleware / Keystone Auth. The main title is "Keystone Auth" with a subtitle "Configuring Keystone Authorization". Below this, a note states: "This middleware is required for Keystone Authentication/Authorization (along with the "Keystone Auth Token Support" middleware). The "reseller_prefix" must match the value used in your Keystone endpoint's publicurl and privateurl and must not be AUTH_, because that is used by SwiftStack's Authentication Middleware." A note below that says: "For example, if your Keystone endpoint's publicurl was [http://192.168.22.100:80/v1/KEY_\\${tenant_id}s](http://192.168.22.100:80/v1/KEY_${tenant_id}s), then you would set reseller_prefix to KEY_ here." The "Settings" section contains the following fields:

<input checked="" type="checkbox"/> Enabled	
operator_roles	admin
reseller_prefix	KEY_
reseller_admin_role	admin

At the bottom are "Submit" and "Cancel" buttons.

Configuring Keystone Auth Token Support: From the SwiftStack controller, select the **Cluster > Manage > Middleware > Keystone Auth Token Support** option.


Note

auth_uri is deprecated

Figure 41: Keystone Auth

The screenshot shows the Keystone Auth Token Support configuration page. At the top, there is a breadcrumb navigation: Home / Clusters / Manage mercury-dev / Manage Middleware / Keystone Auth Token Support. The main title is "Keystone Auth Token Support" with a subtitle "Configuring Keystone Auth Token Support". Below this, a note states: "This middleware is required for Keystone Authentication/Authorization (along with the "Keystone Auth" middleware)." The "Settings" section contains the following fields:

<input checked="" type="checkbox"/> Enabled	
identity_uri	http://172.26.233.235:5000/
auth_uri	http://172.26.233.235:5000/
admin_user	swift
admin_password	cisco123
admin_tenant_name	swiftstack

Usage in Cisco VIM

To support SwiftStack endpoint configuration, update the setup_data.yaml with the following:

```
#####
# Optional Swift configuration section
#####
# SWIFTSTACK: # Identifies the objectstore provider by name
#   cluster_api_endpoint: <IP address of PAC (proxy-account-container) endpoint>
#   reseller_prefix: <Reseller_prefix configured in Swiftstack Keystone middleware E.g KEY_>
#   admin_user: <admin user for swift to authenticate in keystone>
#   admin_password: <swiftstack_admin_password>
#   admin_tenant: <The service tenant corresponding to the Account-Container used by
Swiftstack>
#   protocol: <http or https> # protocol that swiftstack is running on top
```

The automation supports two modes of Integration with SwiftStack- Integration during fresh install of the pod and a reconfigure option to add a SwiftStack endpoint to an existing Pod running CiscoVIM 2.0.

In the Fresh Install mode, adding the setup_data.yaml is automatically provision the following in Keystone.

- Keystone service for Object Store.
- Keystone endpoints for the Object Store service.
- A SwiftStack admin user with admin role in a SwiftStack tenant.

Integration Testing: In order to test if the Keystone integration has been successful, request a token for the configured swift user, tenant

Output must contain a properly generated endpoint for the object-store service that points to the SwiftStack PAC cluster endpoint with the expected "reseller_prefix" For example: KEY_

```
curl -d '{"auth":{"passwordCredentials":{"username": "<username>", "password": "<password>"}, "tenantName": "<swift-tenant>"} }' -H "Content-type: application/json" < OS_AUTH_URL >/tokens
```

Output has to list endpoints generated by Keystone for the object-store cluster endpoint of SwiftStack for the user tenant (SwiftStack account).

Sample output snippet (all IP and Keys are just examples, they vary from Pod to Pod):

```
{
  "access": {
    "metadata": {
      "is_admin": 0,
      "roles": [
        "33f4479e42eb43529ec14d3d744159e7"
      ]
    },
    "serviceCatalog": [
      {
        "endpoints": [
          {
            "adminURL": "http://10.30.116.252/v1",
            "id": "3ca0f1fee75d4e2091c5a8e15138f78a",
            "internalURL": "http://10.30.116.252/v1/KEY_8cc56cbe99ae40b7bleaeabb7984c77d",
            "publicURL": "http://10.30.116.252/v1/KEY_8cc56cbe99ae40b7bleaeabb7984c77d",
            "region": "RegionOne"
          }
        ],
        "endpoints_links": []
      }
    ]
  }
}
```

```

        "name": "object-store",
        "type": "object-store"
    },
    .....

```

Verify that the Keystone user has access to the SwiftStack cluster. Using the token generated preceding for the swiftstack user and tenant, make a request to the SwiftStack cluster

```
curl -v -H "x-auth-token: <auth-token>"  
http://10.30.116.252/v1/KEY_8cc56cbe99ae40b7b1eaeabb7984c77d
```

This lists all the containers (if present) for the SwiftStack tenant (account)

Integrating SwiftStack over TLS: The automation supports SwiftStack integration over TLS. To enable TLS, the CA root certificate must be presented as part of the /root/openstack-configs/haproxy-ca.crt file. The **protocol** parameter within the SWIFTSTACK stanza must be set to **https**. As a pre-requisite, the SwiftStack cluster has to be configured to enable HTTPS connections for the SwiftStack APIs with termination at the proxy servers.

Cinder Volume Backup on SwiftStack

Cisco VIM, enables cinder service to be configured to backup its block storage volumes to the SwiftStack object store. Cinder Volume Backup on SwiftStack feature is automatically configured if the SWIFTSTACK stanza is present in the setup_data.yaml. The mechanism to authenticate against SwiftStack during volume backups leverages the same keystone SwiftStack endpoint configured for use to manage objects. The default SwiftStack container to manage cinder volumes within the Account (Keystone Tenant as specified by "admin_tenant") is currently defaulted to **volumebackups**.

Once configured, cinder backup service is automatically be enabled as follows.

```
cinder service-list
+-----+-----+-----+-----+-----+
| Binary      | Host       | Zone | Status | State | Updated_at
| Disabled Reason |           |
+-----+-----+-----+-----+-----+
| cinder-backup | c43b-control-1 | nova | enabled | up   | 2017-03-27T18:42:29.000000
| -           |           |
| cinder-backup | c43b-control-2 | nova | enabled | up   | 2017-03-27T18:42:35.000000
| -           |           |
| cinder-backup | c43b-control-3 | nova | enabled | up   | 2017-03-27T18:42:33.000000
| -           |           |
| cinder-scheduler | c43b-control-1 | nova | enabled | up   | 2017-03-27T18:42:32.000000
| -           |           |
| cinder-scheduler | c43b-control-2 | nova | enabled | up   | 2017-03-27T18:42:32.000000
| -           |           |
| cinder-scheduler | c43b-control-3 | nova | enabled | up   | 2017-03-27T18:42:31.000000
| -           |           |
| cinder-volume | c43b-control-1 | nova | enabled | up   | 2017-03-27T18:42:35.000000
| -           |           |
| cinder-volume | c43b-control-2 | nova | enabled | up   | 2017-03-27T18:42:30.000000
| -           |           |
| cinder-volume | c43b-control-3 | nova | enabled | up   | 2017-03-27T18:42:32.000000
| -           |           |
+-----+-----+-----+-----+-----+
```

Backing up of an existing cinder volume is as follows

```
openstack volume list
+-----+-----+-----+-----+
| ID          | Display Name | Status | Attached to |
+-----+-----+-----+-----+
| f046ed43-7f5e-49df-bc5d-66de6822d48d | ss-vol-1     | available | 1 |
+-----+-----+-----+-----+
```

SolidFire Integration with Cisco VIM

```

openstack volume backup create f046ed43-7f5e-49df-bc5d-66de6822d48d
+-----+
| Field | Value
+-----+
| id    | 42a20bd1-4019-4571-a2c0-06b0cd6a56fc |
| name  | None
+-----+

openstack container show volumebackups
+-----+
| Field | Value
+-----+
| account | KEY_9d00fa19a8864db1a5e609772a008e94 |
| bytes_used | 3443944
| container | volumebackups
| object_count | 23
+-----+

swift list volumebackups

volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00001
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00002
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00003
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00004
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00005
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00006
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00007
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00008
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00009
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00010
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00011
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00012
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00013
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00014
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00015
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00016
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00017
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00018
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00019
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00020
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc-00021
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc_metadata
volume_f046ed43-7f5e-49df-bc5d-66de6822d48d/20170327185518/az_nova_backup_42a20bd1-4019-4571-a2c0-06b0cd6a56fc_sha256file

```

SolidFire Integration with Cisco VIM

Cisco VIM supports the automated integration with a customer-managed SolidFire cluster for a block-storage option. SolidFire supports Cinder service for backup of block-storage. The pre-deployed SolidFire cluster has two HA networks such as management network and storage network. The management network is on 1G interface with active/Passive configuration for two ports, while the storage network is on 10G interface with active/active Link Aggregation Control Protocol (LACP) configuration.

It is recommended that the :

- Storage network of Cisco VIM is same as that of SolidFire.
- Management network of Solidfire to be reachable from Cisco VIM control nodes.

SolidFire is available only as a day-0 configuration. To enable SolidFire, update the setup_data.yaml file with the following code prior to the installation.

```

SOLIDFIRE:
  cluster_mvip: <management IP of SolidFire cluster> # must be reachable from the controller
  Nodes
  cluster_svip: <storage VIP on SolidFire cluster to be used by CVIM> # must be in Cisco
  VIM storage/management network; recommended to have it in storage network for better
  performance
  admin_username: <admin user on SolidFire cluster to be used by CVIM>
  admin_password: <password for admin user defined above; password criteria is:
    "satisfy at least 3 of the following conditions: " \
      "at least 1 letter between a to z, " \
      "at least 1 letter between A to Z, " \
      "at least 1 number between 0 to 9, " \
      "at least 1 character from !$#0%^-_+=, " \
    "AND password length is between 8 and 20 characters."

```

Cisco VIM Configurations for VPP/VLAN Installation

If you are installing Cisco VIM with VPP/VLAN, the mechanism driver in the setup_yaml file should reflect the same.

Cisco VPP/VLAN Mechanism Driver Configuration

```

MECHANISM_DRIVERS: vpp
TENANT_NETWORK_TYPES: "VLAN"
TENANT_VLAN_RANGES: <START>:<END>           # arbitrary VLAN range***
NFV_HOSTS: ALL
NR_RESERVED_VSWITCH_PCORES: <int> # Optional, defaults to 2; takes values in the range 2
to 4, in order to increase performance by
allocating more cores to VPP

```

Cisco VIM Configuration for Cisco VTS Installation

If you are installing Cisco VIM with Cisco Virtual Topology Systems, you must enter the Cisco VTS parameters in Cisco VIM setup_yaml file.

Cisco VTS Mechanism Driver Configuration

```

MECHANISM_DRIVERS: vts
TENANT_NETWORK_TYPES: "VLAN"
TENANT_VLAN_RANGES: <START>:<END> # arbitrary VLAN range***
ENABLE_JUMBO_FRAMES: True

```



Note

VLAN range overlap on the physical network could occur if a hardware VTEP is configured on a top of rack (ToR) switch. (VTEPs are Virtual Extensible Local Area Network (VXLAN) tunnel end points.)

NFV Parameters

```

NFV_HOSTS: ALL
# Only enabled when NFV_HOSTS is set to ALL
#####
## Only 2 Values allowed is: 2M or 1G (defaults to 2M)
#VM_HUGEPAGE_SIZE: 2M or 1G

```

Along with supporting it globally, Cisco VIM also supports VM_HUGEPAGE_SIZE on a per server basis with OVS/ACI/VTS/VPP as mechanism driver.

```

SERVERS:
compute-server-1:
  VM_HUGEPAGE_SIZE: <2M or 1G> # <== optional

```

```

## Percentage of huge pages assigned to VM
## On NVF_HOSTS enabled hosts, VM memory can be a mix of regular pages and huge pages. This
## setting sets the ratio. By default, all VM memories (100%)
## has huge pages.
## Only input of type integer is allowed, in the range of 0-100 (including 0 and 100)
# values < 100 is only supported for mechanism driver of openvswitch or ACI
#VM_HUGEPAGE_PERCENTAGE: 100

```

Along with supporting it globally, Cisco VIM also supports VM_HUGEPAGE_PERCENTAGE on a per server basis with openvswitch/aci as mechanism driver.

SERVERS:

```

compute-server-1:
    VM_HUGEPAGE_PERCENTAGE: <0 to 100> # <== optional, only for mechanism driver
openvswitch/aci

```



Note

If huge page is used, the memory used in the flavor must be exact multiples of the huge page sizes. For example, memory must be multiple of 2s if 2M huge page is used, and multiple of 1024s if 1G huge page is used.

VMTP Parameters

```

VMTP_VALIDATION parameters: #Required if vmtp is enabled
VMTP_VALIDATION:
    VTS_NET:           #Required if VMTP is enabled for VTS (for VTS only this block is
                       needed)
    ENABLED: <true or false>

```

Networking Parameters

```

NETWORKING:
    ...
networks:
    ...
    - 
        vlan_id: <VLAN to carry VTS tenant traffic>      # required for VTS
        subnet: <subnet IP cidr>
        gateway: <tenant GW IP>
        pool:
            - "<begin tenant IP> to <end tenant IP>"      # ***
        segments:
            - tenant

```



Note

The tenant network pool size has to take into account the IP addresses that are statically assigned through the VTS VTSR VM bootstrap configuration. For more information, see the [Installing Cisco VTS](#)

Cisco VTS Parameters

```

VTS_PARAMETERS:
VTS_USERNAME: 'admin'                      # Required to be 'admin'
VTS_PASSWORD: <VTC UI password>
VTS_NCS_IP:   <VTC mx-net IP>             # VTC mx-net VIP for VTC HA (cannot be in mx-net pool
                                             range)
VTS_SITE_UUID: <VTS site uuid> # VTS SITE UUID mandatory VTS parameter (Unique Pod UUID

```

```

to indicate which pod the VTS is controlling)
VTC_SSH_USERNAME: '<vtc_ssh_username>' # Required parameter when VTS Day0 is enabled or
running NFVbench and/or VMTP
VTC_SSH_PASSWORD: '<vtc_ssh_password>' # Required parameter when VTS Day0 is enabled or
running NFVbench and/or VMTP

VTS_Day0_PARAMETERS:
VTS_2.5 mandates the VTC inventory generation and day0 configuration for VTF's to register.
without VTS_DAY0 the cloud is not operational as VTF does not register to VTC. Hence all
cloud operations fail
This is a boolean variable set as True or False. If set True, VTC day0 can be configured
by the CiscoVIM Installer
By default values is 'False', i.e. if VTS_DAY0 is not set, the orchestrator sets it internally
to 'False'
VTS_DAY0: '<True|False>'

## Optional, BGP ASN:
BGP ASN: int # Optional, min=1, max=65535; if it is not defined, the default to 23
## Optional, MANAGED:
MANAGED : <TRUE OR FALSE> #Optional; if it is true, tor_info in SERVERS becomes mandatory,
CONFIGURE_TORS under TORSWITCHINFO should be false and VTS deployment mode is
managed.

```

**Note**

The mx-net IP pool configuration must take into account the IP addresses that are allocated to the VTC (VTS_NCS_IP). For more information, see the [Installing Cisco VTS](#)

Enabling ACI in Cisco VIM

Cisco VIM integrates with ACI in two fashion: with and without the APIC plugin. When Cisco VIM integrates with the Opflex ML2 plugin (in Unified mode), the tenant and provider VLANs are managed dynamically as the VMs come and go in the cloud. In addition, Cisco VIM supports the administrator driven automated workflow to provision the provider networks. This Opflex ML2 plugin integration is supported on a C-series based Fullon or Micropod running with Cisco VIC 1227 and Intel NIC (X710) with redundancy on M4-based pods.

Cisco VIM orchestrator configures the day-0 aspects of the ACI fabric, along with the Opflex ML2 plugin integration. The only exception is the manual configuration of L3 out.

If ACI plugin is not used, Cisco VIM invokes the APIC APIs to pre-provision the right set of VLANs (along with the day-0 aspects) on the corresponding server ports ahead of time.

In both the cases, pod management operations are supported.

Before you begin

As Cisco VIM does the day-0 configuration of the ACI, following are the assumptions that Cisco VIM makes for the integration to happen.

- Before the Cisco VIM installation, the APIC 3.2 controllers running in a cluster of three must be installed and active.
- All spine and leaf switches are booted in ACI mode and discovered under fabric inventory. The number of leaf switches cannot be changed after the initial install.

The IP address should be assigned to each device from the TEP_ADDRESS_POOL.

Serial Number	Pod ID	Node ID	Node Name	Rack Name	Model	Role	IP	Decommissioned	Supported Model	SSL Certificate
SAL1843ZGZK	1	201	spine1		N9K-C9336PQ	spine	10.0.112.94/32	False	True	yes
FDO0210171PSA	1	102	leaf2		N9K-C93180VC-EX	leaf	10.0.112.64/32	False	True	yes
FDO0210812V9	1	101	leaf1		N9K-C93180VC-EX	leaf	10.0.112.95/32	False	True	yes

- Network should be designed such that the management node and controllers are reachable to APIC controllers.
- ACIINFRA a new networking segment is introduced for ACI policy management; ACIINFRA segment should not overlap with the VLANID across the infrastructure
- Tunnel end point address pool (TEP_ADDRESS_POOL) is set to ACI default at 10.0.0.0/16. Ensure that this address space is not assigned in the cloud.
- Multicast address pool is set to ACI default at 225.0.0.0/15. Ensure that this address space is not assigned anywhere in the cloud.
- ACIINFRA VLANID, TEP_ADDRESS_POOL, and multicast address pool are immutable for the lifecycle of the infrastructure.
- Pre-provision of L3 out API network is done before the installation of Cisco VIM and listed below:
 - Create installer tenant and VRF and provide the name of it in setup_data.
 - Create L3out routed outside object and provide its name in the setup_data.
 - Ensure that this api-l3out must be associated to the tenant VRF.



Note The L3-out object for OpenStack API network needs to be consistently named, that is, the Name of the L3 Outside object must be the same as the name provided for its corresponding External Network Instance Profile. Example: If you provide api_l3out_network: api-l3out in setup_data, then your dn for the api network should resolve to something like the following:
cvim-installer-tenant|uni/tn-cvim-installer-tenant/out-api-l3out/instP-api-l3out.



Note By default optimized DHCP and optimised metadata services are deployed with ACI integration.



Note The plugin automation configures DHCP and Metadata agents in optimized mode. No option is provided in setup_data to change that setting in the current implementation.



Note Using Auto-TOR provisioning via APIC API, the port PV (port*VLAN) count in a given ACI Fabric domain is under the scale limits 10000 PV/ToR and 450000 PV/Fabric.

Run the following setup_data in the VIM to add a new APICINFO:

```

APICINFO:
apic_hosts: [<ipv4_1|ipv6_1|host1>, ..., <ipv4_5|ipv6_5|host5>] # max of 5, min of 1, not 2;
reconfigurable
apic_username: # common across the 3; apic_password: # common across the 3; apic_system_id: # string
max length of 8
apic_resource_prefix: string e.g. cvim-1 # max length of 6 apic_tep_address_pool: 10.0.0.0/16 # static
today multicast_address_pool: 225.0.0.0/15 # static, today apic_pod_id: <int> #All(int, Range(min=1,
max=65535)), apic_installer_tenant: # String, max length 32
apic_installer_vrf: # string (max length32) VRF which is associated with the pre-provisioned
l3out info and in context of apic_installer_tenant
apic_installer_vlan_pool: # Optional, String, max length 32
apic_installer_physdom: # Optional, String, max length 32
apic_installer_app_profile: # Optional, String, max length 32, if provided must belong in context
of apic_installer_tenant
apic_installer_aep: # Optional, String, max length 32
api_l3out_network: # String, max length 32
configure_fabric: <true or false> # optional, default is True; False as an option when Auto-ToR via
ACI API is used
# mgmt_l3out_network: # String, max length 32 (optional)
NOTE: mgmt_l3out_network and mgmt_l3out_vrf MUST coexist together if defined # mgmt_l3out_vrf: #
String, max length 32 (optional)
NOTE: mgmt_l3out_network and mgmt_l3out_vrf MUST coexist together if defined

# prov_l3out_network: # String, max length 32 (optional) NOTE: prov_l3out_network and
prov_l3out_vrf MUST coexist together if defined
# prov_l3out_vrf: # String, max length 32 (optional) NOTE: prov_l3out_network and
prov_l3out_vrf MUST coexist together if defined
#
# l2out info
# api_l2out_network: # String, max length 32
# mgmt_l2out_network: # String, max length 32
# prov_l2out_network: # String, max length 32
# ext_l2out_network: # String, max length 32
# Note: l2out and l3out networks are mutually exclusive

```

As the APIC manages the Leaf switches, it is mandatory to define the Leaf switches in the following format:

TORSWITCHINFO: (mandatory)

```

SWITCHDETAILS:
-
hostname: <leaf-hostname-1>
vpc_peer_keepalive: <leaf-hostname-2>
vpc_domain: 1 # Must be unique across pairs
br_mgmt_port_info: 'eth1/27' # br_mgmt_* attributes must exist on at least one pair
br_mgmt_vlan_info: '3401'
node_id: <int> # unique across switches

-
hostname: <leaf-hostname-2>
vpc_peer_keepalive: <leaf-hostname-1>
vpc_domain: 1
br_mgmt_port_info: 'eth1/27' # br_mgmt_* attributes must exist on at least one pair
br_mgmt_vlan_info: '3401'
node_id: <int> # unique across switches

-
hostname: <leaf-hostname-3>
vpc_peer_keepalive: <leaf-hostname-4>
vpc_domain: 2 # Must be unique across pairs
node_id: <int> # unique across switches

```

Enabling ACI in Cisco VIM

```

-
hostname: <leaf-hostname-4>
vpc_peer_keepalive: <leaf-hostname-3>
vpc_domain: 2
node_id: <int> # unique across switches
-
```

Cisco VIM orchestrator does the day-0 configuration of the ACI. The SERVERS section of the setup_data needs to be augmented to include the server and the switch port associations as shown in the following steps:

```

c32-control-1.cisco.com:
cimc_info: {cimc_ip: '<v4 or v6>'}
management_ip: <static_ip from management pool> #optional, if defined for one server, has to be
defined for all nodes
# management_ipv6: <Fixed ipv6 from the management_ipv6 pool>      # <== optional, allow manual
static IPv6 addressing, also if defined management_ip has to be defined
#storage_ip: <Fixed ip from the storage pool> # optional but if defined for 1 server, then has to be
defined for all, also if defined management_ip has to be defined
rack_info: {rack_id: RackC}
tor_info: {<leaf-hostname-1>: eth1/15, <leaf-hostname-2>: eth1/15} c32-control-2.cisco.com:
cimc_info: {cimc_ip: '<v4 or v6>'}
management_ip: <static_ip from management pool> #optional, if defined for one server, has to be
defined for all nodes
# management_ipv6: <Fixed ipv6 from the management_ipv6 pool>      # <== optional, allow manual
static IPv6 addressing, also if defined management_ip has to be defined
#storage_ip: <Fixed ip from the storage pool> # optional but if defined for 1 server, then has to be
defined for all, also if defined management_ip has to be defined
rack_info: {rack_id: RackC}
tor_info: {<leaf-hostname-1>: eth1/16, <leaf-hostname-2>: eth1/16} c32-control-3.cisco.com:
cimc_info: {cimc_ip: '<v4 or v6>'}
management_ip: <static_ip from management pool> #optional, if defined for one server, has to be
defined for all nodes
# management_ipv6: <Fixed ipv6 from the management_ipv6 pool>      # <== optional, allow manual
static IPv6 addressing, also if defined management_ip has to be defined
#storage_ip: <Fixed ip from the storage pool> # optional but if defined for 1 server, then has to be
defined for all, also if defined management_ip has to be defined
rack_info: {rack_id: RackC}
tor_info: {<leaf-hostname-1>: eth1/17, <leaf-hostname-2>: eth1/17} c32-compute-1.cisco.com:
cimc_info: {cimc_ip: '<v4 or v6>'}
management_ip: <static_ip from management pool> #optional, if defined for one server, has to be
defined for all nodes
# management_ipv6: <Fixed ipv6 from the management_ipv6 pool>      # <== optional, allow manual
static IPv6 addressing, also if defined management_ip has to be defined
#storage_ip: <Fixed ip from the storage pool> # optional but if defined for 1 server, then has to be
defined for all, also if defined management_ip has to be defined
# VM_HUGHPAGE_PERCENTAGE: <0 - 100> # optional only for compute nodes and when NFV_HOSTS: ALL and
MECHANISM_DRIVER: openvswitch or ACI
# VM_HUGHPAGE_SIZE: <2M or 1G> # optional only for compute nodes and when NFV_HOSTS: ALL and
MECHANISM_DRIVER: openvswitch or ACI
# trusted_vf: <True or False> # optional only for compute nodes which have in SRIOV
# rx_tx_queue_size: <256 or 512 or 1024> # optional only for compute nodes, default if not defined
is 1024
# seccomp_sandbox: <0 or 1> # optional, default if not defined is 1

rack_info: {rack_id: RackC}
tor_info: {<leaf-hostname-3>: eth1/18, <leaf-hostname-4>: eth1/18}
sriov_access_vlan: <single_vlan_id> # optional applies in the case of Auto TOR via APIC API; to not
trunk all provider VLANs in this server, and allow one untagged network

In the case of Intel x710 based full on BOM, the corresponding configuration looks as follows:
INTEL_NIC_SUPPORT: True
INTEL_SRIOV_VFS: 32 #Only for SRIOV (Adapt this value that works for your VNF, max is 32)
...
c32-control-1.cisco.com:
```

```

cimc_info: {cimc_ip: '<v4 or v6>'}
management_ip: <static_ip from management pool> #optional, if defined for one server, has to be
defined for all nodes
  rack_info: {rack_id: RackC}
    tor_info: {<leaf-hostname-1>: eth1/15,
<leaf-hostname-2>: eth1/15}
      dp_tor_info: {<leaf-hostname-1>: eth1/19, <leaf-hostname-2>: eth1/19}
    ....
c32-compute-1.cisco.com:
  cimc_info: {cimc_ip: '<v4 or v6>'}
  management_ip: <static_ip from management pool> #optional, if defined for one server, has to be
defined for all nodes
  rack_info: {rack_id: RackC}
    tor_info: {<leaf-hostname-3>: eth1/15,
<leaf-hostname-4>: eth1/15}
      dp_tor_info: {<leaf-hostname-3>: eth1/16, <leaf-hostname-4>: eth1/16}
      sriov_tor_info: {<leaf-hostname-3>: eth1/17, <leaf-hostname-4>: eth1/17} 7 Assuming SRIOV is turned
on
  ....
c32-storage-1.cisco.com:
  cimc_info: {cimc_ip: '<v4 or v6>'}
  management_ip: <static_ip from management pool> #optional, if defined for one server, has to be
defined for all nodes
  rack_info: {rack_id: RackC}
    tor_info: {<leaf-hostname-3>: eth1/25,
<leaf-hostname-4>: eth1/25}

Addtionaly the mechanism_driver needs to be "aci" and ACINFRA section needs to be defined in the
networks section.

MECHANISM_DRIVERS: aci TENANT_NETWORK_TYPES: "VLAN" TENANT_VLAN_RANGES: <START>:<END>
# arbitrary VLAN range*** NFV
PROVIDER_VLAN_RANGES: <START>:<END> # arbitrary VLAN range

```

Note For the plugin to orchestrate the VLAN pools in the ACI fabric, the SRIOV VLAN segments must be provided in PROVIDER_VLAN_RANGES.

Networking Parameters

```

NETWORKING:
  networks:
    - segments: [aciinfra]
      vlan_id: user_defined_unique_vlan_id. This vlan should not overlap with any of the vlans defined
      in setup data; new item
    other segments same as OVS/VLAN.

```

Note Refer to the ACI documentation for usage of L3 out external network that is consumed by VMTP below. Also, ensure that the L3 out routed configuration is provisioned in the ACI "common" tenant.

We support execution of VMTP for external network with ACI in place. For VMTP, the NET_NAME key for EXT_NET needs to match the name of the L3out for external network

```

VMTP_VALIDATION:
EXT_NET:
NET_NAME: <name of L3out for the external network>

```

Support for Provider Networks in ACI: OpFlex plugin integration 3.2 does not currently support a fully automated workflow to provide provider networks in neutron. Cisco VIM has provided a utility that will support provisioning neutron provider networks.

- After the installer has completed deployment, ensure that Fabric Access policies for the external link from the border leaf switches have been created manually. This is the link that will carry the L2 traffic between the external ToRs

Additional Settings for Auto-ToR via ACI API on Day 0

and the border leaf switches. These may be configured as desired (direct PC, PC or VPC). This is typically a one-time admin setup.

- Create a neutron network and subnet in the OpenStack tenant as usual. This is the provider subnet that will be carried through L2 into the fabric. Do not provide segmentation_id. Enable DHCP.
- Run the following command to provision the provider network in ACI:

```
cd installer-<tagid>/tools
./apic_create_provider_net.py -netid <neutron-provider-net-id> --staticpath
<path-to-external-interface-on-borderleaf>
--segmentationid <vlan-id> --tenantid <openstack-tenant-id>
```

Additional Settings for Auto-ToR via ACI API on Day 0

When using the option of ToR automation via ACI API on day-0 without the APIC plugin, FABRIC_INTERFACE_POLICIES (under SERVER_COMMON section) and vim_apic_network section are required. The FABRIC_INTERFACE_POLICIES include global fabric policies which are defined in APIC and to be applied to the relevant switch ports. Listed below is the definition of the same:

```
SERVER_COMMON:
...
FABRIC_INTERFACE_POLICIES:
    global:          # Global policies can be overridden per server role
    tor_info:        # <list of global fabric policies for control plane>
    - <str>          # string, E.g. hintfpol-25G-On-RS-FEC. Must be pre-provisioned.
    - <str>          # string E.g cdpIfP-CdpDisable
    dp_tor_info:    # <list of policies for data plane for Intel NIC>
    - <str>          # string, E.g. lacplagp-LacpActive. Must be pre-provisioned.
    - <str>          # string E.g lldpIfP-LldpEnable
    sriov_tor_info: # <list of policies for sriov interfaces>
    - <str>          # string, E.g. hintfpol-25G-On-RS-FEC. Must be pre-provisioned.
    control:        # Optional in case needs to over-ridden. Must be one of the SERVER
                    roles.
    tor_info:        # <list of policies for control plane>
    - <str>          # string, E.g. hintfpol-25G-On-RS-FEC. Must be pre-provisioned.
    compute:         # Optional in case needs to over-ridden. Must be one of the SERVER
                    roles.
    dp_tor_info:    # <list of policies for data plane>
    - <str>          # string, E.g. hintfpol-25G-On-RS-FEC. Must be pre-provisioned.

    # Pre-provision EPG/BD policies to be configured for management and tenant/provider
    # EPGs (FHS policy)
    EPG_POLICIES: # Goes by Network Segments and is entirely Optional
        management:      # Optional, list of policy for management segment
        - <path_to_epg_policy> # Must be pre-provisioned.
tn-cvim-installer-tenant/trustctrlpol-Trust-DHCP-Sv.
    provider:        # Optional, list of policy for provider segment
    - <path_to_epg_policy1> # Must be pre-provisioned.
    - <path_to_epg_policy2> # Must be pre-provisioned.
    tenant:          # Optional, list of policy for tenant segment
    - <path_to_epg_policy1> # Must be pre-provisioned.
    - <path_to_epg_policy2> # Must be pre-provisioned.
```

In the vim_apic_networks section, the provider and tenant VLAN definitions are listed as below:

```
vim_apic_networks:
    EPG_NAME: 'VL-%s-EPG' # required; pattern substituted with vlan_id
```

```

BD_NAME: 'VL-%s-BD'      # required; pattern substituted with vlan_id
PROVIDER:
# Support static vlans with the following attributes defined (these vlans will only be
referred to bind and unbind Static Ports)
- app_profile: <str>    # string, E.g. Must be pre-provisioned in ACI POD. 'Core-AP'
  EPG_NAME: <str>        # <optional string. Will prefix the pattern in the global EPG_NAME
definition>
  mode: trunk|access    # string, default trunk
  tenant: <str>          # string, E.g. Must be pre-provisioned in ACI POD. 'Core'
  vlan_ids: '<3550>'    # Can be a only a single id
  config_type: pre-provisioned
  vlan_pools:
    - <str-1>            # string E.g. 'Server-VlanPool'

# The "vlan_ids" can be a VLAN Range only for L2 networks provided they belong
#to the same VLAN pool and EPGs map to the same Phydom, App Profile, VRF
- vlan_ids: '<3550>'    # <Can be a single id or range and/or comma separated list>
  EPG_NAME: <str>        # <optional string. Will prefix the pattern in the global EPG_NAME
definition>
  BD_NAME: <str>          # <optional string. Will prefix the pattern in the global BD_NAME
definition>
  vlan_pools:             # List of vlan pool names. Must be pre-provisioned in ACI POD
    - <str-1>              # string E.g. 'Server-VlanPool'
    - <str-2>              # string E.g. 'Ext-VlanPool'
  phys_dom: <str>         # string. Must be pre-provisioned in ACI POD. E.g. Server-PhysDom

  description: <str>      # optional; string. Must be pre-provisioned in ACI POD. E.g.
'provider net 3550'
  tenant: <str>            # string, E.g. Must be pre-provisioned in ACI POD. 'Core'
  app_profile: <str>       # string, E.g. Must be pre-provisioned in ACI POD. 'Core-AP'
  vrf: <str>                # string, E.g. Must be pre-provisioned in ACI POD. 'Core'
  subnets:
    - scope: <str>          # string. Can be '<private'|'public'|'private,shared'
      gateway_cidr:        # IPv4 or IPv6 network gateway with cidr E.g.
'240b:c010:101:2839::ffff/64'
  ctrl: <no-default-gateway> or "nd" or "nd, no-default-gateway" or "no-default-gateway,nd"
or "unspecified" # when gateway cidr is of type IPv6
  or
  ctrl: <no-default-gateway> or "querier" or "querier,no-default-gateway" or
"no-default-gateway,querier" or "unspecified" # when gateway cidr is of type IPv4
  13-out:                  # optional, List of L3out External Routed Network Instances. Must
be pre-provisioned
    - <External Routed Network/Instance Profile>    # E.g. Core-Ba-Ma-L3out/Core-Ba-Ma-ExtEPG

    - <External Routed Network/Instance Profile>    # E.g. cel-epc-CP-L3out/cel-epc-CP-ExtEPG

  mode: trunk|access    # string, default trunk
  l2_unknown_unicast: <flood or proxy>
  limit_ip_learning: <true or false>
  preferred_group_member: <include or exclude>    # Optional, default is exclude
  arp_flood: <true or false>
  unicast_routing: <true or false>
  nd_policy: <true or false>          # When true, ensure that path/ndifpol is defined under
SERVER_COMMON -> EPG_POLICIES -> provider section

TENANT:
# Does not contain 13out
# Can be a VLAN Range only for L2 networks provided they belong to the
# same VLAN pool and EPGs map to the same Phydom, App Profile, VRF
- vlan_ids: '<2251:2260,2280,2290>'
  EPG_NAME: <str>        # <optional string. Will prefix the pattern in the global EPG_NAME
definition>
  BD_NAME: <str>          # <optional string. Will prefix the pattern in the global BD_NAME

```

Setting of Memory Oversubscription Usage

```

definition>
  vlan_pools:
    - 'Server-VlanPool'
  phys_dom: Server-PhysDom
  tenant: 'Core'
  app_profile: <str>  # string, E.g. Must be pre-provisioned in ACI POD. 'Core-AP'
  vrf: Nokia-LI
  mode: trunk
  subnets:           # May contain a subnet in which case only valid value is a single
  vlan id
    - scope: <str>      # string. Can be <'private'|'public'|'private,shared'>
      gateway_cidr:   # IPv4 or IPv6 network gateway with cidr  E.g.
      '240b:c010:101:2839::ffff/64'
      l2_unknown_unicast: <flood or proxy>
      limit_ip_learning: <true or false>
      preferred_group_member: <include or exclude>    # Optional, default is exclude
      arp_flood: <true or false>
      unicast_routing: <true or false>
      nd_policy: <true or false>      # When true, ensure that path/ndifpol is defined under
      SERVER_COMMON -> EPG_POLICIES -> provider section

```



- Note** Ensure that you update right VLAN information when using this option in context of APIC during reconfiguration of VLANs.

Setting of Memory Oversubscription Usage

Cloud allows you for over-subscription of resources (CPU, Memory, storage). The memory oversubscription value is set to 1.5. Cisco VIM gives the flexibility to change the default values at the beginning of the installation. You can adjust the memory oversubscription value between 1.0 to 4.0.

Run the following command to set the NOVA_RAM_ALLOCATION_RATIO, on fresh install:

```
# cd installer-<tagid>/openstack-configs/
# update NOVA_RAM_ALLOCATION_RATIO value in openstack_config.yaml
```

What to do next

Once the NOVA_RAM_ALLOCATION_RATIO is set, continue with the rest of the steps as planned for installation.

Setting of CPU Oversubscription Usage

Cloud allows you for over-subscription of CPU, storage and memory. The CPU oversubscription value is set to 16.0. Cisco VIM gives the flexibility to change the default values before the installation begins. You can adjust the CPU oversubscription value in the range of 1.0 to 16.0.

Run the following command to set the NOVA_CPU_ALLOCATION_RATIO on fresh install:

```
# cd installer-<tagid>/openstack-configs/
# update NOVA_CPU_ALLOCATION_RATIO value in openstack_config.yaml
```

What to do next

Once the NOVA_CPU_ALLOCATION_RATIO is done, continue with the rest of the steps as planned for installation.

Disabling Management Node Accessibility to Cloud API Network

Cisco VIM provides cloud connectivity verification from the data and control plane point of view using tools like cloud-sanity, VMTP, and NFVbench, which are typically run from the Management node. For these tools to work, reachability to the Cloud API, external, and provider network is a must.

From release Cisco VIM 2.4.3 onwards, you can set the MGMTNODE_EXTAPI_REACH variable to True in the setup_data file to override the need to ensure reachability of management node from Cloud API, external, and provider network.

For example:

```
MGMTNODE_EXTAPI_REACH: True
```

By default, the MGMTNODE_EXTAPI_REACH variable is set to True. If you do not want to use the MGMTNODE_EXTAPI_REACH variable, you can set it to False as part of the day-0 settings.



Note

- The MGMTNODE_EXTAPI_REACH variable must be set during the initial install, and cannot be changed later.
- You must ensure that the Cloud API, external, and provider network are properly routable, as Cisco VIM cannot automatically validate the same.

When MGMTNODE_EXTAPI_REACH is set to True, features such as VMTP and NFVbench are no longer accessible from the management node.

Enabling NFVbench on Cisco VIM

This section describes how to setup and use NFVbench with Cisco VIM.

Once the pre-requisites for the management node hardware (Intel NIC) are met, add the NFVbench configuration in the setup_data.yaml. By default, NFVbench configuration is not enabled in Cisco VIM as it needs additional hardware.

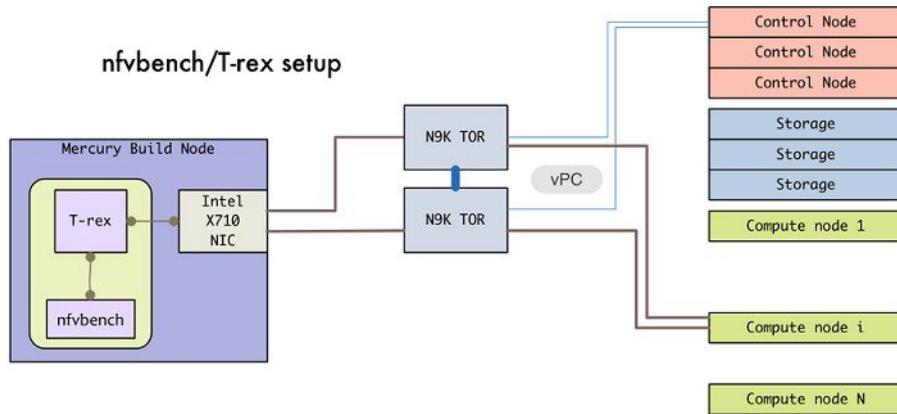
Before you begin

- If you are using Quanta servers, see [Installing Management Node on Quanta Servers](#), for the day-0 BIOS setting of the management node.
- NFVbench offering in Cisco VIM, requires an extra Intel NIC (Intel X710 NIC (4 x 10G) or Intel XL710 (2x40G)) or Intel xxv710 (25G) to be installed on the management node.

- To interact with Intel NIC, TReX traffic generator uses DPDK interface, and makes use of hardware instead of just software to generate packets. This approach is more scalable and enables NFVbench to perform tests without software limitations.

If your NIC has more than two ports, use the first two ports only. Connect the first port to the first ToR switch (order is given by setup_data.yaml) and the second port to the second TOR switch. In case of only one ToR switch connect the first two ports to it as shown in the NFVbench Topology figure.

Figure 42: NFVbench topology setup



Step 1

To enable the NFVbench, set the following command:

```
NFVBENCH:
  enabled: True      # True or False
  tor_info: {switch_a_hostname: ethx/y, switch_b_hostname: ethx/y} # mandatory
#  tor_info: {switch_c_hostname: 'etha/b,ethx/y'} # use if there is only one TOR switch
  vtep_vlans: vlan_id1,vlan_id2 # mandatory only when mechanism driver is VTS, or tenant type is
VXLAN
#  nic_ports: int1,int2          # Optional input, indicates which 2 of the 4 ports in the 10G
intel NIC ports on the
traffic. If nothing is
specified, the tool assumes it is Port 1,2 i.e. the first 2
ports will be used

# nic_slot: <int> # Optional, defaults to 1st set of unbonded pair of NIC ports in an Intel 710 or
520
card the code finds; Via this option, one can choose to run NFVbench via XL710, 520 or X710 card

# For VTS/VXLAN
#  vteps: "vtep_ip1,vtep_ip2"      # Mandatory and needed only for VTS/VXLAN. Specify reconfigurable
and separated IP pairs in tenant network
                                         and not in the tenant pool.
#
# For VXLAN over vxlan-tenant network
#  vteps: "vtep_ip1,vtep_ip2"      # Mandatory, specify reconfigurable and separated IP pairs in
vxlan-tenant network and not in the
                                         vxlan-tenant pool, reconfigurable
#  vnids: "vni_id1, vni_id2"       # Mandatory, specify reconfigurable and separated vni_id pairs
#
# Note: if nic_ports are defined, then nic_slot has to be defined and vice-versa
```

```

# refer to the VTS_PARAMETERS and TORSWITCHINFO if NFVbench is enabled
# Required when mechanism driver is VTS
VTS_PARAMETERS:
...
    VTS_NCS_IP: '<vtc_ssh_username>'           # Required parameter when VTS enabled
    VTC_SSH_USERNAME: '<vtc_ssh_username>'       # Mandatory for NFVbench
    VTC_SSH_PASSWORD: '<vtc_ssh_password>'        # Mandatory for NFVbench
    VTS_SITE_UUID : <vts_site_uuid>                # Mandatory if VTS is enabled (Unique Pod UUID to indicate
which pod the VTS                                         is controlling)
    # Minimal settings always required with NFVbench
TORSWITCHINFO:
CONFIGURE_TORS: True
...
SWITCHDETAILS:
- hostname: <switch_a_hostname>
  username: admin
  password: <password>
  ssh_ip: <ssh access to the switch a

- hostname: <switch_b_hostname>
  username: admin
  password: <password>
  ssh_ip: <ssh access to the switch b

```

The tor_info provides the information to configure the TOR switches. Two ports specified by interfaces will be configured in trunk mode in the same port-channel **po**. NFVbench needs the login details to access ToR details and retrieve TX/RX counters. Manual configuration is required if the 'CONFIGURE_TORS' is set to 'True'.

With VTS as mechanism driver additional settings are needed. NFVbench needs access to VTS NCS to perform cleanup after it detaches traffic generator port from VTS. Also a pair of VTEP VLANs is required for VLAN to VxLAN mapping. Value can be any random VLAN ID. Note that vtep_vlans field is required if VxLAN is used as encapsulation without VTS.

Step 2

To do manual configuration on the ToRs, we recommend you to perform the following configurations:

```

interface Ethernetx/y
  switchport mode trunk
  switchport trunk allowed vlan <3000-3049>
  spanning-tree bpdufilter enable

```

Customization of Edge

From release Cisco VIM 3.0.0 onwards, you need to specify a flavor metadata key "hw:vcpu0_pin_to_shared" to use the optional flavor in OpenStack, that can be set only at day-0.

When a VM is spawned with the flavor that contains the above metadata sets to **yes**, NOVA allocates additional vCPU on top of the vCPU count specified in the flavor and pin vCPU0 to the pCPU that is reserved in the pool. The pinning of vCPU to pCPU is load balanced, if hyper-threading is enabled in the host level.

To enable this configuration, set **hw:cpu_policy** to **dedicated**. And it is often used together with **hw:emulator_threads_policy** being set to **share**, so that the VM emulator threads are also pinned to the same dedicated pool to enable better real time processing for latency and performance sensitive VNFs.

To enable this feature, set the following command in setup_data on day-0.

```
ENABLE_VM_EMULATOR_PIN: <True or False> # optional, default is false
```

The number of cores reserved is determined by VM_EMULATOR_PCORES_PER_SOCKET, which is also pre-defined at the day-0 configuration.

```
VM_EMULATOR_PCORES_PER_SOCKET: < 1 to 4> # Optional, takes effect only when  
ENABLE_VM_EMULATOR_PIN is true, and if undefined default to value of 1.
```

You can set the NOVA_OPT_LOW_LATENCY flag to enable further optimization on nova libvirt, to achieve lower latency for VM applications. To be specific, it will set **cpu_mode** to **host-passthrough** and **cpu_model_extra_flags** to **tsc-deadline** in nova.conf.

```
NOVA_OPT_FOR_LOW_LATENCY: True or False # Optional, default to False
```

From release Cisco VIM 3.2.1 onwards, an option to enable Intel's Resource Director Technology (RDT) by Cache Allocation Technology (CAT) is available. To enable CAT, you must enable NFV_HOSTS option. You can enable the CAT option only as a day-0 option with the following option in the setup_data:

```
INTEL_RDT:  
  ENABLE_CAT: false # Enable Intel CAT, optional and default to False  
  #Reserved cachelines per socket for sockets, allowed value of 1 to 32.  
  #Only valid when ENABLE_CAT is sets to True.  
  RESERVED_L3_CACHELINES_PER_SOCKET: 3
```

The cachelines reserved for hosts are not immediately applied. When first VM with the cacheline requirements lands on the any NUMA node of one compute node, Cisco VIM performs the cacheline partitioning on the host. If VM with no cacheline requirements are spawned (as defined via flavor) on one compute node, all VMs are allowed to use all cachelines available in the CPU. When the last VM with cacheline requirements is deleted from any NUMA node of one compute node, Cisco VIM resets the cacheline masks so that all new and existing VMs are allowed to use all available cachelines again.

To support extreme low latency (less than 50 micro-seconds) requirements for vRAN workload, Cisco VIM integrates with Intel N3000 FPGA card for both hardware offload and I/Os. The option of N3000 Intel card is only allowed with Quanta servers, and the following item in the setup_data enables the cards on the servers. These configurations have effect only on computes where the N3000 cards are installed.

```
# Intel FPGA N3000 NIC (for QCT now)  
# By default, FPGA VF is not enabled.  
# To enable, define a value in the range from 1 to 8.  
# INTEL_FPGA_VFS: <integer value from 1 to 8>  
  
# By default, FPGA VF is not enabled.  
# VFS support for Intel FPGA N3000 NIC (for QCT now) for SRIOV  
# INTEL_VC_SRIOV_VFS: <integer value from 1 to 32>
```

You can enable the virtual function (VFS) values optionally at a per server level, however the global configuration is needed, as listed below.

```
SERVERS:  
compute-server-1:  
  INTEL_FPGA_VFS: <integer value from 1 to 8>  
  INTEL_SRIOV_VFS: <integer value from 1 to 32>  
  INTEL_VC_SRIOV_VFS: <integer value from 1 to 32>
```



Note You can enable single or multiple options listed above on a per server basis.

NFV Host Configuration

NFV Host configuration describes how to configure NFV hosts and Cisco VIM monitoring.

Cisco VIM supports CPU pinning and huge page on the compute nodes. To enable non-uniform memory access (NUMA), you can use ALL (case insensitive) to configure all compute nodes. For VTS and VPP/VLAN, only the value of ALL is allowed. For OVS/VLAN, alternatively, you can list the compute nodes where NUMA must be enabled.

```
# For VPP and VTS, only NFV_HOSTS: ALL is allowed
NFV_HOSTS: ALL
or
NFV_HOSTS: ['compute-server-1']
```

By default, hyper-threading is enabled across compute nodes in Cisco VIM. Based on certain VNF characteristics, Cisco VIM offers user the capability to disable hyper-threading across the pod on day-0. You can also disable it on a single compute node on day-n, updating the setup_data and doing remove or add of compute nodes (see Utilizing NUMA features in Cisco NFV Infrastructure section in the Cisco VIM Admin Guide for details on day-n operation). To disable hyper-threading, update the setup_data with the following name or value pair before starting the installation.

```
DISABLE_HYPERTHREADING: True or False; this is optional and default value is false.
```

Install Mode

You can deploy Cisco VIM on the setup in one of the following install modes:

1. Connected: In this mode, the setup must be connected to Internet to fetch artifacts and docker images.
2. Disconnected: In this mode, Cisco VIM is not connected to Internet. The artifacts and docker images are loaded from USB device

Based on the deployment type, select the install mode as connected or disconnected.

```
# Install Mode: connected/disconnected
INSTALL_MODE: connected
```

Enabling NFVIMON on Cisco VIM

The Cisco VIM solution uses Cisco NFVI Monitor (NFVIMON) to monitor the health and performance of the NFVI. This includes monitoring both the physical and logical components of single or multiple NFVI pods. The NFVIMON feature enables extensive monitoring and collection of performance data for various components of the cloud infrastructure including Cisco UCS blade and rack servers, service profiles, Nexus top of rack switches, fabric connections, and OpenStack instances.

The monitoring system is designed such that it can monitor single or multiple pods from a single management system. NFVIMON is enabled by extending the setup_data.yaml file with relevant information. You can enable NFVIMON on an existing pod through the reconfigure option. Then, add the pod as the VIM resource to be monitored in a Control Center.

NFVIMON consists of four components: ceilometer service (for data collection), collector, resource manager (RM), and control-center with Cisco Zenpacks (CZ). Integration of NFVIMON into VIM is loosely coupled and the VIM automation only deals with installing the ceilometer service software needed to monitor the pod. The installing of the other NFVIMON components (collector, resource manager (RM) and control-center with Cisco Zenpacks (CZ), are outside the scope of the install guide.

Before you Begin

Installation of NFVIMON

Ensure that you have engaged with the account team for services engagement on the planning and installation of the NFVIMON accessories along with its network requirements. The image information of collector, Resource Manager (RM) and control-center with Cisco Zenpacks (CZ) is available only through Cisco Advance Services. At a high level, have a node designated to host a pair of collector VM for each pod, and a common node to host CC and RM VMs, which can aggregate and display monitoring information from multiple pods.

The collector VMs must have two interfaces:

- Interface with br_mgmt of the VIM.
- Interface that is routable and reachable to the VIM Installer REST API and RM VMs.

As the collector VM is in an independent node, four IPs from the management network of the pod must be pre-planned and reserved. The installation steps of the collector, resource manager (RM) and control-center with Cisco Zenpacks (CZ) are part of Cisco advance services activities.

Installation of NFVIMON

The ceilometer service is the only component in NFVIMON that is managed by Cisco VIM orchestrator. While the ceilometer service collects the metrics to pass OpenStack information of the pod to the collectors, the Cisco Zenpack available in the controller node gathers the node level information.

To enable NFVIMON as part of the VIM installation, update the setup_data with the following information:

```
#Define the PODNAME
PODNAME: <PODNAME with no space>; ensure that this is unique across all the pods
NFVIMON:
    MASTER:                      # Master Section
        admin_ip: <IP address of Control Centre VM>
    COLLECTOR:                   # Collector Section
        management_vip: <VIP for ceilometer/dispatcher to use> #Should be unique across the VIM
Pod; Should be part of br_mgmt network
        Collector_VM_Info:
            -
                hostname: <hostname of Collector VM 1>
                password: <password_for_collector_vm1>  # max length of 32
                ccuser_password: <password from master for 'ccuser' (to be used for self monitoring)>
# max length of 32
                admin_ip: <ssh_ip_collector_vm1> # Should be reachable from br_api network
                management_ip: <mgmt_ip_collector_vm1> # Should be part of br_mgmt network
            -
                hostname: <hostname of Collector VM 2>
                password: <password_for_collector_vm2>  # max length of 32
                ccuser_password: <password from master for 'ccuser' (to be used for self monitoring)>
# max length of 32
                admin_ip: <ssh_ip_collector_vm2> # Should be reachable from br_api network
                management_ip: <mgmt_ip_collector_vm2> # Should be part of br_mgmt network
    COLLECTOR_TORECONNECTIONS: # Optional. Indicates the port where the collector is hanging
off. Recommended when Cisco NCS 5500 is used as ToR
        - tor_info: {po: <int>, switch_a_hostname: ethx/y, switch_b_hostname: ethx/y}

# Section of MASTER_2 and COLLECTOR_2 are optional and only needed to support NFVIMON in
HA
MASTER_2: # Master Section
    admin_ip: <IP address of Control Centre VM>
COLLECTOR_2: # Collector Section
    management_vip: <VIP for ceilometer/dispatcher to use> #Should be unique across the VIM
Pod; Should be part of br_mgmt network
    Collector_VM_Info:
        -
            hostname: <hostname of Collector VM 1>
```

```

password: <password_for_collector_vml> # max length of 32
ccuser_password: <password from master for 'ccuser' (to be used for self monitoring)>
# max length of 32
admin_ip: <ssh_ip_collector_vml> # Should be reachable from br_api network
management_ip: <mgmt_ip_collector_vml> # Should be part of br_mgmt network
-
hostname: <hostname of Collector VM 2>
password: <password_for_collector_vm2> # max length of 32
ccuser_password: <password from master for 'ccuser' (to be used for self monitoring)>
# max length of 32
admin_ip: <ssh_ip_collector_vm2> # Should be reachable from br_api network
management_ip: <mgmt_ip_collector_vm2> # Should be part of br_mgmt network
COLLECTOR_TORCONNECTIONS: # Optional. Indicates the port where the collector is hanging off. Recommended when Cisco NCS 5500 is used as ToR
- tor_info: {po: <int>, switch_a_hostname: ethx/y, switch_b_hostname: ethx/y}

DISPATCHER:
rabbitmq_username: admin # Pod specific user for dispatcher module

NFVIMON_ADMIN: admin_name # Optional, once enabled, you need to have only one admin that is reconfigurable to add/update non-root user id

```



Note If NFVIMON HA is enabled, ensure that all the admin IPs are on the same subnet for NFVIMON VMs and deployed servers.

To monitor ToR, ensure that the following **TORSWITCHINFO** sections are defined in the setup_data.yaml file.

```

TORSWITCHINFO:
SWITCHDETAILS:
-
    hostname: <switch_a_hostname>: # Mandatory for NFVIMON if switch monitoring is needed
    username: <TOR switch username> # Mandatory for NFVIMON if switch monitoring is needed
    password: <TOR switch password> # Mandatory for NFVBENCH; Mandatory for NFVIMON if switch monitoring is needed
    ssh_ip: <TOR switch ssh ip> # Mandatory for NFVIMON if switch monitoring is needed
    ....
-
    hostname: <switch_b_hostname>: # Mandatory for NFVIMON if switch monitoring is needed
    username: <TOR switch username> # Mandatory for NFVIMON if switch monitoring is needed
    password: <TOR switch password> # Mandatory for NFVIMON if switch monitoring is needed
    ssh_ip: <TOR switch ssh ip> # Mandatory for NFVIMON if switch monitoring is needed
    ....

```



Note TORSWITCH monitoring is disabled when running Cisco VIM with ACI plugin enabled.

Enabling CVIM-MON on Cisco VIM

The Cisco VIM solution offers the use of Cisco VIM Monitor (CVIM-MON) to monitor the health and performance of NFVI. This includes monitoring both the physical and logical (openstack services) components at each NFVI pod level.

The CVIM-MON feature enables extensive monitoring and collection of performance data for various components of the cloud infrastructure, and also the OpenStack instances. The monitoring system is designed at a single pod level.

CVIM-MON is enabled by extending the setup_data.yaml file with relevant information.

You can enable CVIM-MON on an existing pod that is installed with Cisco VIM 2.4.3 or later, through the reconfigure option.

The components of CVIM-MON are as follows:

- **CVIM_MON:** It provides the base functionality of monitoring and KPIs.
- **SNMP:** It is enabled for SNMP traps. This component is available only if CVIM_MON is enabled.
- **SERVER-MON:** If SNMP is enabled, you can enable SERVER_MON to use SNMP from the Cisco IMC of Cisco UCS C-series server only. This component is available only if the SNMP option is enabled.

Install the CVIM-MON using the standard Cisco VIM installer after enabling it in the setup_data configuration file. It is assumed that the pod is newly installed with Cisco VIM 2.4.3 or later. To install CVIM-MON, CVIM_MON and PODNAME keys must be added to the setup_data.yaml file.

The CVIM_MON key has:

- **enabled:** A boolean value indicating whether CVIM_MON is enabled.
- **polling_intervals:** It is a dictionary having three different levels of data collection frequencies. Defining polling_intervals is optional and a default value is used if the polling_interval is not defined.
- **ui_access:** A boolean indicating whether CVIM-MON UI access is enabled or not.

PODNAME is mandatory for CVIM-MON.

CVIM-MON, SNMP and SERVER-MON can be installed by the standard Cisco VIM installer, if they are enabled in the setup_data configuration file.

Following are the aspects of the SNMP key:

- If SNMP is enabled, CVIM-MON must be enabled.
- List of SNMP managers to send the SNMP traps. This list contains SNMPv2 or SNMPv3 managers. For SNMPv2, community and port field can be set. For SNMPv3, the engine_id and list of users must be specified, where the Engine_id is the EngineContextID which is used to send trap of the SNMP Manager.



Note

SNMP-Traps are sent without setting any authentication or security engine_id for the user.

Property Group and Name	Values	Default Value	Description
PODNAME:	<string>	(required)	Must be provided for identifying each pod if CVIM_MON is enabled.
CVIM_MON: enabled	true false	false	A boolean indicating whether CVIM-MON is enabled or not. Set to True to enable CVIM_MON.
CVIM_MON: ui_access	true false	true	A boolean indicating whether CVIM-MON UI access is enabled or not.
CVIM_MON: polling_intervals:	-	-	Metric collection frequency low_frequency <= low frequency < med frequency < high frequency <= 1 hour
low_frequency	-	deprecated	Must be higher than med_frequency integer following with time sign (m/h)
medium_frequency	-	deprecated	Must be higher than high_frequency integer following with time sign (s/m/h)
high_frequency	10s to 30s	15s	Integer following with time sign (s/m/h)
SNMP:enabled	true false	false	A Boolean indicating whether SNMP Trap is enabled or not. If true, CVIM_MON:enabled must also be set to true.
SNMP:managers:	-	-	A list of up to 3 SNMP managers to send traps
address	<ipv4 or ipv6>	(required)	IPv4 or IPv6 address of the SNMP manager
port	1-65535	162	Optional, port to send traps
version	v2c v3	v2c	SNMP manager version

Enabling or Disabling Autobackup of Management Node

Property Group and Name	Values	Default Value	Description
community	<string>	public	Used for SNMPv2c
SNMP:managers:users:			Required for SNMPv3, up to 3 users.
engine_id	<hexadecimal string>	(required v3)	ContextEngineId (unique across all managers) Minimum length is 5 and max length is 32 Cannot be all 00s or FFs; and cannot start with 0x
name	<string>	(required v3)	User name
auth_key	<string>	(required v3)	Authorization password, must be eight characters at least
authentication	SHA MD5	SHA	Authentication protocol
privacy_key	<str>	(auth_key)	Encryption key
encryption	'AES128' 'AES192' 'AES256'	'AES128'	Encryption protocol
SERVER_MON: enabled	true false	false	Enable SNMP traps for CIMC faults (UCS C-series only)
host_info:	'ALL' or list of servers	'ALL'	Specifies the UCS-C servers to be monitored.
rsyslog_severity	emergency alert critical error warning notice informational debug	(Optional)	Specifies the minimum severity from the UCS C-server logs that are to be sent to remote syslog servers



Note

If SERVER_MON.rsyslog_severity is configured, you must configure SYSLOG_EXPORT_SETTINGS as well to indicate the remote syslog servers to send the logs.

Enabling or Disabling Autobackup of Management Node

Cisco VIM supports the backup and recovery of the management node. By default, the feature is enabled. Auto-snapshots of the management node happens during pod management operation. You can disable the autobackup of the management node.

To enable or disable the management node, update the setup_data.yaml file as follows:

```
# AutoBackup Configuration
# Default is True
#autobackup: <True or False>
```

Enabling Custom Policy for VNF Manager

Some of the VNF managers operates, using specific OpenStack features that require the admin role. Cisco VIM introduces a feature to enable non-admin role for VNF managers (such as Cisco ESC). VNF manager is used to operate and manage tenant VMs in the OpenStack cloud, with minimally enhanced privileges.

To enable this option, the administrator needs to add the following line in the setup_data.yaml:

```
ENABLE_ESC_PRIV: True # optional; default is false
```

Forwarding ELK logs to External Syslog Server

Cisco VIM supports backup and recovery of the management node, to keep the process predictable and avoid loss of logs. The software supports the capability of forwarding the ELK logs to multiple external syslog server. It supports minimum of one and maximum of four external syslog servers.

Before launching the installation, update the setup_data.yaml file with the following information:

```
#####
## SYSLOG EXPORT SETTINGS
#####
SYSLOG_EXPORT_SETTINGS:
  -
    remote_host: <Syslog_ipv4_or_v6_addr> # required IP address of the remote syslog
    server protocol : udp # defaults to udp
    facility : <string> # required; possible values local[0-7]or user
    severity : <string; suggested value: debug>
    port : <int>; # defaults, port number to 514
    clients : 'ELK' # defaults and restricted to ELK;

    remote_host: <Syslog_ipv4_or_v6_addr> # IP address of the remote syslog #2 (optional)
    server protocol : udp # defaults to udp
    facility : <string> # required; possible values local[0-7]or user
    severity : <string; suggested value: debug>
    port : <int>; # defaults, port number to 514
    clients : 'ELK' # defaults and restricted to ELK;

  # Please note other than the remote host info, most of the other info is not needed; Also
  # the client list is restricted to ELK only
```

With this configuration, the ELK logs are exported to an external syslog server. You can add this configuration to a pod that is already up and running. For more details, refer to Forwarding ELK logs to External Syslog Server section in the admin guide.

Support of NFS for ELK Snapshot

Cisco VIM optionally supports NFS for ELK snapshots. In this configuration, the remote location specified in the configuration has to allow user elasticsearch (2020) and group mercury (500) to read/write into the path specified in remote_path of the remote_host server.

Before launching the installation, update the setup_data.yaml file with the following information:

```
#####
## ES_REMOTE_BACKUP
#####
#ES_REMOTE_BACKUP:           # Set if Elasticsearch backups will use a remote host
```

Support for TTY Logging

```
# service: 'NFS'          # Only value supported is NFS
# remote_host: <ip_addr>  # IP of the NFS server
# remote_path: </root/es_remote> # Path to location of the backups in the remote server
```

With this configuration, the ELK snapshots are hosted at the remote NFS location, thereby ensuring that the management node does not run out of disk space. You can add this configuration to a pod that is already up and running. For more details, refer to Support of NFS for ELK Snapshot section in the admin guide.

Support for TTY Logging

Cisco VIM supports enabling of TTY logging on the management node and all of the cluster hosts through the option in the setup_data.yaml file. By default, the TTY logging feature is not enabled. The feature is made available only at the time of installation. If SYSLOG_EXPORT_SETTINGS is configured, the TTY audit messages are available in local syslog, Kibana dashboard, and remote syslog.

For the TTY logging to take effect in the management node, reboot the management node based on the customer downtime window.

At the end of the installation, the following message is displayed: Management node needs to be rebooted for TTY Logging to take effect.

Before launching the installation, update the setup_data.yaml file with the following information:

```
# TTY Logging with pam.d and auditd. Events available in Kibana and remote syslog, if syslog
# export is enabled
ENABLE_TTY_LOGGING: <True or False>      # default value is False
```

Configuring Additional VIM Administrators

Cisco VIM supports management of VIM administrators. VIM administrator can login to the management node or Unified Management node through SSH or the console using the configured password. Administrators have their own accounts. After the VIM administrator account creation, the administrator can manage their own password using the Linux “passwd” command. You can change the vim_admins parameter to add and remove VIM administrators during reconfiguration, while the passwords for existing accounts remain unchanged.

Before launching the installation, update the setup_data.yaml file with the following information:

```
vim_admins:
- vim_admin_username: <username>
  vim_admin_password_hash: <sha512-password-hash>#
- vim_admin_username: <username>
  vim_admin_password_hash: <sha512-password-hash>
- vim_admin_username: <username>
  vim_admin_password_hash: <sha512-password-hash>
```

The value of password hash must be in the standard sha512 format.

With the preceding configuration, administrators have access to a shell with system privileges on the management node. To go hand in hand with the management of VIM administrator, Cisco VIM offers the option of disabling “root login”. Listed below are the available options:

```
# Permit Root Login (optional, default True)
#   True: admin can ssh to management node with root userid and password
#   False: admin can not use root userid for ssh; must use vim_admin_username
```

```
#           At least one vim_admin must be configured if this is False
permit_root_login: True
```

Support of LDAP for Management Node

Cisco VIM supports enabling of LDAP for admin access to the management node. It can be added as a day-0 or day-1 activity. Multiple LDAP entries are allowed as only the domain_name and ldap_uri in each entry are mandatory. Ensure that the ldap_uris is secured over ldaps, and the TLS is enabled for the external api (external_lb_vip_tls: True).

To obtain sudo access to the management node and execute ciscovim commands, you must manually add the user with root privileges to the wheel group in the corresponding LDAP domain, for example, usermode -aG wheel user1.

To enable this feature, update the setup_data with the following during installation.

```
vim_ldap_admins:
  - domain_name: corp_ldap1
    ldap_uri: "ldaps://10.30.116.253:636,ldaps://10.30.116.254:636"
    ldap_search_base: "dc=cisco,dc=com"      # Optional
    ldap_schema: rfc2307                 # Optional
    ldap_user_object_class: posixAccount   # Optional
    ldap_user_uid_number: uidNumber        # Optional
    ldap_user_gid_number: gidNumber        # Optional
    ldap_group_member: memberUid          # Optional
  - ...
```

Horizon Hosting Through NAT or DNS Aliases

From release Cisco VIM 3.0.0, you can deploy the Horizon portal through NAT or DNS alias. As a security measure, Horizon accepts a list of host addresses (IP or DNS) that are accessible. By default, this list includes the external_lb_vip_addr, the external_lb_vip_fqdn, and the ipv6 address (if applicable) only.

An optional parameter HORIZON_ALLOWED_HOSTS added in the setup_data accepts the list of IP addresses and/or DNS names that you want to add as allowed hosts. Mostly, this IP address list match with the NAT address used for the deployment.

Before launching the installation, update the setup_data.yaml file with the following information:

```
HORIZON_ALLOWED_HOSTS:
  - <NAT-IP>
  - <NAT-IP>
```

With the preceding configuration, administrator can access the Horizon dashboard through aliases or NAT IPs.

DHCP Reservations for VM's MAC Addresses

From release Cisco VIM 3.2.0, you can have DHCP reservations for virtual machine MAC addresses, to get the same IP address always regardless of the host hypervisor or operating system they are running. To avail this optional feature, few restrictions exist.

If the MAC address ends with 00:00, then

- First entry of the first octet must be a Hex

- Second entry of the first octet must be 2, 6, a or e

For example, the MAC address entry can be [a-f][2,6,a,e]:yz:uv:ws:00:00.

To enable this feature, add the following entry in the setup_data file:

```
BASE_MACADDRESS: <[a-f][2,6,a,e]:[a-f0-9][a-f0-9]:[a-f0-9][a-f0-9]:[a-f0-9]:00:00>
```



- Note** To avoid mac-address collision, ensure that a minimum of last three octets is 00. For example:
BASE_MACADDRESS: <[a-f][2,6,a,e]:[a-f0-9][a-f0-9]:[a-f0-9][a-f0-9]:00:00>

Customizing SSH Login Banner

From release Cisco VIM 3.0.0, you can provide a customized banner that will be displayed when an administrator attempts to login to the management node or Unified Management node. An optional parameter ssh_banner in the setup_data accepts a string or message to be displayed before the login prompt. This message indicates a warning consistent with a company's IT policies.

Before launching the installation, update the setup_data.yaml file with the following information:

```
ssh_banner: |
  WARNING: Unauthorized access to this system is forbidden and will be
  prosecuted by law. By accessing this system, you agree that your actions
  may be monitored if unauthorized usage is suspected.
```

Cinder Volume Encryption

From release Cisco VIM 3.0.0, you can encrypt Cinder volumes using Linux Unified Key Setup (LUKS). This encryption is enabled by default, and does not require any installation. For more information on creating encrypted Cinder volumes, see *Cisco VIM Admin Guide*.

Configuring Support for Read-only OpenStack Role

By default, Cisco VIM deployment of OpenStack supports two user roles: admin and user. Admin have privilege to view and change all OpenStack resources including system and project resources. Users have privileges to view and change only project resources.

Optionally, Cisco VIM provides OpenStack user role which is read-only or readonly. Read-only users can view the project resources, but cannot make any changes. Use the optional parameter ENABLE_READONLY_ROLE to enable this feature.

The admin can only assign the readonly role using the Horizon dashboard or OpenStack CLI, to the target user for accessing each project. A user can be given the readonly role to multiple projects.



- Note** Ensure that the admin role is not given for the user having only readonly access, as the conflict of access will not constrain the user to read-only operations.

Enabling this feature provides the following enhancements to the Cisco VIM Pod.

- "readonly" role is added to the OpenStack deployment.

- OpenStack service policies are adjusted to grant read permissions such as "list" and "show", but not "create", "update", or "delete".
- **All Projects** tab is added to the Horizon interface. This allows the readonly user to see all instances for which the user have access. Under the **Project** tab, you can see the resources for a single project. You can change the projects using the Project pulldown in the header.

Before launching the installation, update the setup_data.yaml file with the following information:

```
ENABLE_READONLY_ROLE: True
```

With the preceding configuration, the readonly role is created in OpenStack. After deployment, the administrators have the privilege to create new users assigned with this role.



Note If the ENABLE_READONLY_ROLE is False (by default), the readonly role will not have special permissions or restrictions, but have create, update, and delete permissions to project resources similar to that of project member. You need to assign the users with readonly role, when ENABLE_READONLY_ROLE is set to True.

VPP Port Mirroring Support

The VPP Port Mirror feature enables you to selectively create a mirror port to a VM. This mirror port detects all the packets sent and received by the VM without having access to the VM. The packets captured in this manner can be saved as pcap files, which can be used for further analysis by tools like Wireshark and so on.

The following CLIs are available in Cisco VIM:

- **vpp-portmirror-create:** Tool to create mirrored ports corresponding to Openstack ports
- **vpp-portmirror-delete:** Tool to delete mirrored ports
- **vpp-portmirror-list:** Tool to get a list of current mirrored port

In addition, the VPP port mirror tools perform the following tasks:

- Checks if the port specified is a valid neutron port with valid UUID pattern
- Checks if there is a corresponding Vhost interface in the VPP instance for the neutron port specified
- Checks if the port has already mirrored

VPP Port Mirroring Usage

Step 1 Identify the VM that you want to monitor and the compute host on which it runs.

From the Management node, execute the following:

```
#cd /root/openstack-configs
# source openrc
# openstack server show vm-7
+-----+
| Field | Value |
+-----+
| OS-DCF:diskConfig | AUTO |
| OS-EXT-AZ:availability_zone | nova |
| OS-EXT-SRV-ATTR:host | k07-compute-1 |
```

VPP Port Mirroring Usage

```
| OS-EXT-SRV-ATTR:hypervisor_hostname | k07-compute-1 |
| OS-EXT-SRV-ATTR:instance_name | instance-0000004d |
| OS-EXT-STS:power_state | Running |
| OS-EXT-STS:task_state | None |
| OS-EXT-STS:vm_state | active |
| OS-SRV-USG:launched_at | 2018-05-10T02:40:58.000000 |
| OS-SRV-USG:terminated_at | None |
| accessIPv4 | |
| accessIPv6 | |
| addresses | net1=10.0.1.4 |
| config_drive | |
| created | 2018-05-10T02:40:37Z |
| flavor | m1.medium (ac4bbdd7f-ff05-4f0d-90a5-d7376e5e4c75) |
| hostId | 8e7f752ab34153d99b17429857f86e30ecc24c830844e9348936bafcc |
| id | 46e576c1-539b-419d-a7d3-9bdde3f58e35 |
| image | cirros (e5e7e9d8-9585-46e3-90d5-4ead5c2a94c2) |
| key_name | None |
| name | vm-7 |
| os-extended-volumes:volumes_attached | [] |
| progress | 0 |
| project_id | 434cf25d4b214398a7445b4fafaf8956a |
| properties | |
| security_groups | [{u'name': u'my_sec_group'}] |
| status | ACTIVE |
| updated | 2018-05-10T02:40:58Z |
| user_id | 57e3f11eaf2b4541b2371c83c70c2686 |
+-----+

```

Step 2

Identify the neutron port that corresponds to the interface that you want to mirror.

```
# openstack port list | grep 10.0.1.4
| ed8caee2-f56c-4156-8611-55dde24f742a | | fa:16:3e:6a:d3:e8 | ip_address='10.0.1.4',
subnet_id='6d780f2c-0eef-4c6c-a26c-c03f47f37a45' |
```

Step 3

ssh to the target compute node on which the VM is running and join the VPP docker container.

```
# vpp
neutron_vpp_13881 [root@k07-compute-1 /]#
```

The syntax of the Port mirror create tool is as follows:

```
neutron_vpp_13881 [root@k07-compute-1 /]# vpp-portmirror-create
Option -p (--port) requires an argument
-p --port [arg] Port in openstack port uuid format. Required.
-d --debug Enables debug mode
-h --help This page
-n --no-color Disable color output
VPP port mirror utility.
```

Step 4

Create a port mirror using the Neutron port ID identified in Step 2.

The CLI tool displays the mirrored interface name.

```
neutron_vpp_13881 [root@k07-compute-1 /]# vpp-portmirror-create -p ed8caee2-f56c-4156-8611-55dde24f742a
===== [ Port Mirroring ] =====
2018-05-14 22:48:26 UTC [ info] Interface inside vpp is VirtualEthernet0/0/1 for Openstack port:
ed8caee2-f56c-4156-8611-
55dde24f742a
2018-05-14 22:48:26 UTC [ info] Port:ed8caee2-f56c-4156-8611-55dde24f742a is now mirrored at taped8caee2
2018-05-14 22:48:26 UTC [ notice] Note! Please ensure to delete the mirrored port when you are done
with debugging
```

Note Use the –debug flag to troubleshoot the Linux/VPP commands that are used to set up the port mirror.

Step 5

Use the tap device as a standard Linux interface and use tools such as tcpdump to perform packet capture.

```
neutron_vpp_13881 [root@k07-compute-1 /]# tcpdump -leni taped8caee2
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on taped8caee2, link-type EN10MB (Ethernet), capture size 262144 bytes
16:10:31.489392 fa:16:3e:6a:d3:e8 > fa:16:3e:0e:58:7b, ethertype IPv4 (0x0800), length 98: 10.0.1.4
> 10.0.1.10: ICMP echo
request, id 32513, seq 25752, length 64
16:10:31.489480 fa:16:3e:0e:58:7b > fa:16:3e:6a:d3:e8, ethertype IPv4 (0x0800), length 98: 10.0.1.10
> 10.0.1.4: ICMP echo
reply, id 32513, seq 25752, length 64
16:10:32.489560 fa:16:3e:6a:d3:e8 > fa:16:3e:0e:58:7b, ethertype IPv4 (0x0800), length 98: 10.0.1.4
> 10.0.1.10: ICMP echo
request, id 32513, seq 25753, length 64
16:10:32.489644 fa:16:3e:0e:58:7b > fa:16:3e:6a:d3:e8, ethertype IPv4 (0x0800), length 98: 10.0.1.10
> 10.0.1.4: ICMP echo
reply, id 32513, seq 25753, length 64
16:10:33.489685 fa:16:3e:6a:d3:e8 > fa:16:3e:0e:58:7b, ethertype IPv4 (0x0800), length 98: 10.0.1.4
> 10.0.1.10: ICMP echo
request, id 32513, seq 25754, length 64
16:10:33.489800 fa:16:3e:0e:58:7b > fa:16:3e:6a:d3:e8, ethertype IPv4 (0x0800), length 98: 10.0.1.10
> 10.0.1.4: ICMP echo
reply, id 32513, seq 25754, length 64
^C
```

Step 6 Obtain a list of all the mirrored ports.

```
neutron_vpp_13881 [root@k07-compute-1 /]# vpp-portmirror-list
VPP interface VPP-side span port Kernel-side span port Neutron port
-----
VirtualEthernet0/0/0 tapcli-0 tap88b637e4 net-vpp.port:88b637e4-43cc-4ea2-8a86-2c9b940408ec
VirtualEthernet0/0/1 tapcli-1 taped8caee2 net-vpp.port:ed8caee2-f56c-4156-8611-55dde24f742a
```

Step 7 Remove the mirrored port.

```
neutron_vpp_13881 [root@k07-compute-1 /]# vpp-portmirror-delete -p ed8caee2-f56c-4156-8611-55dde24f742a
======[ Port Mirroring Operation]=====
2018-05-14 23:18:49 UTC [ info] Interface inside vpp is VirtualEthernet0/0/1 for Openstack
port:ed8caee2-f56c-4156-8611-55dde24f742a
Deleted.
2018-05-14 23:18:49 UTC [ info] Port:ed8caee2-f56c-4156-8611-55dde24f742a is now un-mirrored
```

Setting up VXLAN/EVPN in Cisco VIM

Choose single VXLAN or multi-VXLAN (multi refers to 2) network terminating on the same box on day-0. Two vxlan segments such as vxlan-tenant and vxlan-ecn are defined.

For single VXLAN network, define only the vxlan-tenant. For two-VXLAN network, define vxlan-ecn segment along with vxlan-tenant network.

To enable VXLAN/EVPN in Cisco VIM, define the following in the setup-data file during the Day-0 deployment. Optionally, you can overload the configuration with that of head-end-replication for static VXLAN configuration.

Step 1 In the Networking section, define the segment vxlan-tenant.

```
NETWORKING:
...
networks:
....
```

Setting up VXLAN/EVPN in Cisco VIM

```

- # only needed when NETWORK_OPTIONS is vxlan, and TOR is Cisco NCS5500
vlan_id: <2003>
subnet: <191.168.11.0/25>
gateway: <191.168.11.1>
## 'pool' can be defined with single ip or a range of ip
pool:
- <191.168.11.2,191.168.11.5>
- <191.168.11.7 to 191.168.11.12>
- <191.168.11.20>
segments:
- vxlan-tenant
- # only needed when NETWORK_OPTIONS is vxlan, and TOR is Cisco NCS5500, and second VXLAN segment is
  required
vlan_id: <2005>
subnet: <191.165.11.0/25>
gateway: <191.165.11.1>
## 'pool' can be defined with single ip or a range of ip pool:
- <191.165.11.2,191.165.11.5>
- <191.165.11.7 to 191.165.11.12>
- <191.165.11.20>
segments:
- vxlan-ecn
-

```

Step 2 Define the vxlan section under NETWORK_OPTIONS, only allowed for Cisco NCS 5500 as ToR.

```

# Optional, only allowed for NCS-5500 as tor
NETWORK_OPTIONS:
vxlan:
vxlan-tenant:
provider_network_name: <name of provider network>
bgp_as_num: <int value between 1 and 232-1>
bgp_peers: ['ip1', 'ip2'] ---> list of min length 1, Peer Route Reflector IPs
bgp_router_id: 'ip3' ---> The router ID to use for local GoBGP cluster, part of vxlan-tenant network
but not in the pool
head_end_replication: # Optional, can be brought in as reconfigure
- vtep_ips: vni_id1:vni_id2, vni_id3, ... (upto as many Remote POD vteps, as required)

vxlan-ecn:
provider_network_name: <name of provider network>
bgp_as_num: <int value between 1 and 232-1>
bgp_peers: ['ip1', 'ip2'] ---> list of min length 1, Peer Route Reflector IPs
bgp_router_id: 'ip3' ---> The router ID to use for local GoBGP cluster, part of vxlan-ecn network
but not in the pool
head_end_replication: # Optional and reconfigurable
- vtep_ips: vni_id1:vni_id2, vni_id3, ... (upto as many Remote POD vteps, as required)

```

Note Following are the assumptions for the HER feature:

- VNIs can repeat across two or more remote POD VTEPs for HA.
- VNIs cannot repeat for the same remote POD VTEP.
- Within the same network segment, no remote POD VTEPs IP address can repeat.

Step 3 In the SERVERS section, define vxlan_bgp_speaker_ip for each controller node.

Note The vxlan_bgp_speaker_ip belongs to the vxlan network, however, it is not part of the IP pool defined in the vxlan segment.

```

SERVERS:
control-server-1:
...

```

```
# bgp_speaker_addresses: {vxlan-tenant: <ip address> # <== optional, only when NETWORK_OPTIONS is
vxlan network, for
    controller node only; IP belongs to the vxlan-tenant network but not part of the pool as
defined in the network section
    vxlan-ecn: <ip address>} # <== optional, only needed for multi-vxlan scenario and only when
NETWORK_OPTIONS is vxlan network,
    for controller nodes only; IP belongs to the vxlan-ecn network but not part of the pool as
defined in the network section
```

Note Setting up the BGP route-reflector and accessing it over the VXLAN network from the three controllers is outside the scope of Cisco VIM automation.

For head-end-replication option, define Local POD vtep_ips on all servers that act as compute nodes.

```
# vtep_ips: {vxlan-tenant: <ip address>, vxlan-ecn: <ip address>} #IPs must belong to the associated
IP pool of vxlan-tenant and vxlan-ecn
networks
```

From release Cisco VIM 2.4.9, the BGP session between the controllers and route-reflector is set to be Layer 3 adjacent. By default, it is L2 adjacent. To support Layer 3 adjacency, define bgp_mgmt_address for each controller.

```
# bgp_mgmt_addresses: {vxlan-tenant: <ip address >, vxlan-ecn: <ip address>} # <== optional, only
when
    NETWORK_OPTIONS is vxlan network, for controller node only, needed when BGP peer is
over L3;
    the ip addresses are unique and are from management network, but are not part of the
pool
```

Setting up Trusted Virtual Functions

The kernel feature allows the Virtual Functions to become trusted by the Physical Function and perform some privileged operations such as enabling VF promiscuous mode and changing VF MAC address within the guest. The inability to modify MAC addresses in the guest prevents the users from being able to easily setup up two VFs in a fail-over bond in a guest.

To avail this feature, enable the following under each of the target compute nodes that are enabled with SRIOV.

```
SERVERS:
compute-server-1:
trusted_vf: <True or False> # <== optional, only applicable if its SRIOV node
```

You can avail this feature on day-0 or enable in a compute on day-2 by removing it and adding it back into the cloud after updating the setup_data with the configuration.

Setting up Reception/Transmission Buffer Size

By default, the transmission and reception buffer for the interfaces on each server is set to 1024. This feature allows you to set the rx_tx_queue_size to 256, 512, or 1024 on a per server basis based on the requirement for some VNFs. Also, along with setting the queue size, you can disable the seccomp syscall sandbox in QEMU to avail this feature.

To avail this feature, enable the following under each of the target compute nodes.

```
SERVERS:
compute-server-1:
rx_tx_queue_size: <256 or 512 or 1024> # optional only for compute nodes, default if not
```

```
defined is 1024
seccomp_sandbox: <0 or 1> # optional, Set to 1 by default, if not defined.
```

Updating Cisco NFVI Software

The Cisco VIM installer provides a mechanism to update all OpenStack services and some infrastructure services such as RabbitMQ, MariaDB, HAProxy, and VMTP. Updating host-level packages and management node ELK and Cobbler containers are not supported. Updating Cisco NFVI software has minimal service impact because the update runs serially, component-by-component, one node at a time. If errors occur during an update, an automatic rollback will bring the cloud back to its previous state. After an update is completed, check for any functional cloud impacts. If everything is fine, you can commit the update which clears the old containers from the system. Cisco recommends that you commit the update before you perform any other pod management functions. Skipping the commit option might lead to double faults. If you see any functional impact on the cloud, perform a manual rollback to start the old containers again.



Note Cisco NFVI software updates are not supported for registry related containers and authorized_keys. Also, after the management node repo containers are updated, they cannot be rolled back to the older versions because this requires node packages to be deleted, which might destabilize the cloud.



Note Update of Cisco NFVI software is within the same major version, that is from 3.2.0 to 3.2.1, and not from 2.4 to 3.0.

To prevent double faults, a cloud sanity check is done both before and after the update.

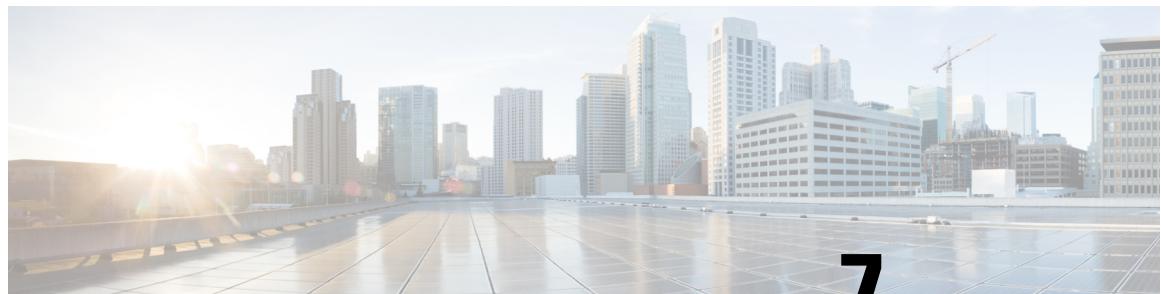
To complete the software update, perform the Installing Cisco VIM [m_Install_VIM.ditamap#id_33373](#). If your management node does not have Internet, complete the [m_Preparing_USB_Stick.ditamap#id_38540](#) procedure first, then follow the Cisco VIM installation instructions. Differences between a software update and regular Cisco VIM installation:

- You do not need to modify setup_data.yaml like you did during the first installation. In most cases, no modifications are needed.
- You do not need to repeat the Cisco VIM Insight installation.
- Minor differences between NFVI software installation and updates are listed in the installation procedure.



Note After you complete a software update, you must commit it before you perform any pod management operations. During software updates, the following operations are locked: add/remove compute/storage node, replace controllers, and rotate fernet key. Before you commit, you can roll back the update to return the node to its previous software version.

For information on updating the Cisco NFVI software, see *Managing Cisco NFVI* of the corresponding *Cisco VIM Administrator Guide*



CHAPTER 7

Installing Cisco VIM Unified Management



Note Cisco VIM Insight is also known as Cisco VIM Unified Management. They are interchangeable across the guide.

Cisco VIM offers the Unified Management solution in standalone mode on a dedicated node to manage multiple Cisco VIM pods. As the UI interacts with the REST API, it is not necessary that the pod must be managed by Insight from Day 0. You can register a pod, with an Insight instance after it is up and running.

The UI supports two types of Admin: UI Admin and Pod Admin. The UI Admin is for the administrators who can add more folks as UI Admin or Pod admin. The Pod Admin has privileges only at the pod level, whereas an UI Admin has privileges both at UI and pod level.

Complete the following procedure to install Cisco VIM Insight on the Cisco NFVI management node.

- [Installing Cisco VIM Unified Management with Internet Access, on page 211](#)
- [Installing Cisco VIM Unified Management with Cisco VIM Software Hub, on page 216](#)
- [Installing Cisco VIM Unified Management with LDAP, on page 217](#)
- [Installing Cisco VIM Unified Management Without SMTP, on page 217](#)
- [Installing Cisco VIM Unified Management without Internet Access , on page 219](#)
- [Installing Cisco VIM Unified Management with Optional Services, on page 222](#)
- [Cisco VIM Insight Post Bootstrap Validation Checks, on page 223](#)
- [VIM UM Admin Login for Standalone Setup, on page 226](#)
- [VIM UM Pod Admin Login for Standalone Setup, on page 227](#)

Installing Cisco VIM Unified Management with Internet Access

Complete the following steps to install Cisco VIM Insight on the Cisco NFVI management node. As security is paramount to pod management, the web-service hosting the single pane of glass is protected through TLS. Following are the steps to get the TLS certificate setup going.

You can select one of the following approaches for the TLS certificate configurations:

1. Provide your own certificate: You can bring in your certificate on the management node and provide the absolute path of .pem and CA certificate files in the insight_setup_data.yaml file. The path must be provided as a value for the key 'PEM_PATH' in the insight_setup_data.yaml file.

2. Generate a new certificate on the node. You can create a new certificate on the node by running the following command:

```
# cd /root/installer-<tag_id>/insight/
#./tls_insight_cert_gen.py -f <path_to_insight_setup_data.yaml>/insight_setup_data.yaml.
```

This script searches for the 'PEM_PATH' inside the insight_setup_data.yaml. As the path is not provided, it creates a new certificate inside install-dir/openstack-configs.



- Note** The self-signed certificate generation utility script is provided for lab/testing deployment only. Ensure that you do not use self-signed certificate generated by this utility for the production deployment.

Before you begin

Complete all Cisco NFVI preparation tasks that are described in [Preparing for Cisco NFVI Installation](#), and the management node that are described [Cisco VIM Management Node Networking](#). The procedure to bootstrap the node hosting the Insight is same as installing the buildnode.iso. Make sure that you plan for a standalone unified management node for production. Click the Yes option if the node is to be used in the production.

Step 1

Enter **ip a** to verify the br_mgmt and br_api interfaces are up and are bound to bond0 and bond1 respectively. For example:

```
$ ip a
br_api: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP link/ether 00:42:68:6f:79:f2
  brd ff:ff:ff:ff:ff:ff
inet nnn.nnn.nnn.nnn/25 brd nnn.nnn.nnn.nnn scope global br_api valid_lft forever preferred_lft
  forever
inet6 fe80::3c67:7aff:fe9:6035/64 scope link valid_lft forever preferred_lft forever
bond1: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue master br_api state UP link/ether
  00:42:68:6f:79:f2 brd ff:ff:ff:ff:ff:ff
br_mgmt: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP link/ether 00:78:88:46:ee:6e
  brd ff:ff:ff:ff:ff:ff
inet nnn.nnn.nnn.nnn/24 brd nnn.nnn.nnn.nnn scope global br_mgmt valid_lft forever preferred_lft
  forever
inet6 fe80::278:88ff:fe46:ee6e/64 scope link valid_lft forever preferred_lft forever
bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue master br_mgmt state UP
  link/ether 00:78:88:46:ee:6e brd ff:ff:ff:ff:ff:ff
```

- Note** The br_mgmt and br_api interfaces are created when you install the RHEL on the management node in [Installing the Management Node](#).

Step 2

Run the following commands to copy the installer directory and the standalone insight_setup_data.yaml.

- a) Copy the installer dir to a directory in /root/. Start the name of the new directory with Insight-tag_id.

```
# cd /root/
# cp -pr installer-<tag_id> <Insight-tag_id>
```

- b) Copy the Standalone insight_setup_data.yaml. Standalone_EXAMPLE file from the Insight-dir/openstack-configs to any other location on the management node or the BOM.

```
# cp /root/Insight-<tag_id>/openstack-configs/insight_setup_data.yaml.
Standalone_EXAMPLE /root/insight_setup_data.yaml
```

Step 3 Modify the insight setup data according to your requirements.

```
#Configuration File:

#####
# User Defined Configuration File.
# Information in this file is specific to the user setup.
#####

# This file is used as an inventory file to setup Insight Container.

#####
# Registry credentials

#####
REGISTRY_USERNAME: '<username>'  
REGISTRY_PASSWORD: '<password>'

# Install Mode: connected/disconnected, Optional parameter; default is connected  
INSTALL_MODE: connected

# https_proxy: <Name of the proxy server without https://> ; Optional Parameter for INSTALL_MODE  
# Needed for connected install only and not required for disconnected mode.

#####
# Super Admin Username Password
#####

# This user is the default Super Admin of the system and can grant Aacess to all other users getting  
registered to PODs.  
# This is a mandatory field and is required to be filled every time.  
UI_ADMIN_USERNAME: '<username>'  
UI_ADMIN_EMAIL_ID: '<email_id@domain.com>'

# Please define the mail server off which the Insight email alias works;  
# For example, outbound.cisco.com  
# Optional: Valid SMTP Server is required for sending mails to the customers. By default, it is set  
as True.  
INSIGHT_SMTP_SERVER: <smtp.domain.com>  
#INSIGHT_SMTP_PORT: <port no.>  
#optional, defaults to 25, if undefined

# for Insight UI, customer needs to create a mailer, so that automated mails come from that alias;  
# For example, vim-insight-admin@cisco.com  
# Mandatory: You need to create a valid email alias that would be responsible for sending email  
notification for users and UI Admin.  
INSIGHT_EMAIL_ALIAS: <Email-Alias@domain.com>  
# Optional: Insight Email Alias Password is required if log in on a SMTP server requires authentication.  
INSIGHT_EMAIL_ALIAS_PASSWORD: <password>      #Optional

#####
# LDAP Configuration
#####
LDAP_MODE: <True or False>      # Required, True when ldap server is available.  
#  
# Following LDAP settings are required only when LDAP_MODE is True.  
LDAP_SERVER: <IP Address of the LDAP Server>  
LDAP_PORT: <port no.>  
LDAP_ADMIN: '<user-DN for admin>' # e.g Complete DN of admin user for bind and search. <cn=admin,  
dc=example, dc=com>  
LDAP_ADMIN_PASSWORD: '<password>' # e.g. password of bind user  
LDAP_BASE_DN: '<DN tree for Groups>' # e.g. 'ou=Groups,dc=cisco,dc=com'  
LDAP_SECURE: '<True or False>' # For protocol to be followed. True is for ldaps and False is for ldap  
# LDAP certificate path for self-signed certificates only;
```

```

# Required when LDAP_SECURE is True for self-signed certificate.
# In case of trusted Root-CA-Certificate, this key is not required.
LDAP_CERT_PATH: '<abs_location_for_cert_path>'
LDAP_USER_ID_ATTRIBUTE: 'LDAP attribute which can be used as user-id' # e.g. <'uid' or 'cn' or 'mail'>
LDAP_GROUP_SEARCH_FILTER: 'LDAP search filter to search groups on LDAP' # e.g.
<LDAP_GROUP_SEARCH_FILTER: "(objectClass=posixGroup)">
LDAP_GROUP_USER_SEARCH_FILTER: 'LDAP search filter to search group-members on LDAP' # e.g.
<LDAP_GROUP_USER_SEARCH_FILTER: "(objectClass posixAccount)">

#TLS certificate path;
#Absolute TLS certificate path, can also be generated using the script tls_insight_cert_gen.py located
at
# installer-<tagid>/insight/; if generated by: tls_insight_cert_gen.py, then entry of the info is
optional;
# the script copies the certs to installer-<tagid>/openstack-configs/ dir
PEM_PATH: <abs_location_for_cert_path>
SSL_CERT_CHAIN_FILE: <abs_location_for_cert_chain_file of x509 certificate> #Mandatory if PEM_PATH
is defined in the setupdata.

#If using tls_insight_cert_gen.py to create the cert, please define the following:
CERT_IP_ADDR: <br_api of the insight node> # Mandatory
CERT_HOSTNAME: <Domain name for Cert> # Optional
And then execute:
# cd installer-<tagid>/insight
# ./tls_insight_cert_gen.py --file <absolute path of insight_setup_data.yaml>

The script generates the certs at installer-<tagid>/openstack-configs/ dir

If bringing in a 3rd part Cert, skip the above step and define the following
CERT_IP_ADDR: <br_api of the insight node> # Mandatory
CERT_HOSTNAME: <Domain name for Cert> # Optional
PEM_PATH in insight_setup_data.yaml, and go to step 4 instead of executing # ./tls_insight_cert_gen.py

As part of insight bootstrap the script copy the certs to installer-<tagid>/openstack-configs/ dir

```

Step 4 Save the edited insight_setup_data.yaml file.

Step 5 Start the insight installation process.

```

$ cd /root/Insight-<tag_id>/insight/
$./bootstrap_insight.py --help
usage: bootstrap_insight.py [-h] --action ACTION
                            [--regenerate_secrets] [--setpassword]
                            [--file INSIGHTSETUPDATA] [--keep] [--verbose]
                            [--backupdir BACKUPDIR] [-y]

Insight install setup helper.

optional arguments:
  -h, --help            show this help message and exit
  --action ACTION, -a ACTION
                        install - Install Insight UI
                        install-status - Display Insight Install Status
                        reconfigure - reconfigure - Reconfigure Insight DB password,
  TLS Certificate, INSIGHT_SMTP_SERVER,
                        INSIGHT_EMAIL_ALIAS_PASSWORD,
                        INSIGHT_EMAIL_ALIAS, INSIGHT_SMTP_PORT
                        LDAP_MODE, LDAP_SERVER, LDAP_PORT, LDAP_ADMIN
                        LDAP_ADMIN_PASSWORD, LDAP_BASE_DN, LDAP_SECURE
                        LDAP_CERT_PATH, LDAP_USER_ID_ATTRIBUTE,
                        SSL_CERT_CHAIN_FILE, LDAP_GROUP_SEARCH_FILTER,
                        LDAP_GROUP_USER_SEARCH_FILTER

```

```

update - Update Insight UI
update-status - Display Insight Update Status
rollback - Rollback Insight UI update
commit - Commit Insight UI update
backup - Backup Insight UI
uninstall - Uninstall Insight UI

--regenerate_secrets, -r
    System generated INSIGHT_DB_PASSWORD
--setpassword, -s
    User supplied INSIGHT_DB_PASSWORD,
--file INSIGHTSETUPDATA, -f INSIGHTSETUPDATA
    Location of insight_setup_data.yaml
--keep, -k
    Preserve Insight artifacts during uninstall
--verbose, -v
    Verbose on/off
--backupdir BACKUPDIR, -b BACKUPDIR
    Path to backup Insight
-y, --yes
    Option to skip reconfigure or uninstall steps without prompt

```

\$./bootstrap_insight.py -a install -f </root/insight_setup_data.yaml>

VIM Insight install logs are at: /var/log/insight/bootstrap_insight/bootstrap_insight_<date>_<time>.log

Management Node validation!

Rule	Status	Error
Check Kernel Version	PASS	None
Check Ansible Version	PASS	None
Check Docker Version	PASS	None
Check Management Node Tag	PASS	None
Check Bond Intf. Settings	PASS	None
Root Password Check	PASS	None
Check Boot Partition Settings	PASS	None
Check LV Swap Settings	PASS	None
Check Docker Pool Settings	PASS	None
Check Home Dir Partition	PASS	None
Check Root Dir Partition	PASS	None
Check /var Partition	PASS	None
Check LVM partition	PASS	None
Check RHEL Pkgs Install State	PASS	None

Insight standalone Input validation!

Rule	Status	Error
Insight standalone Schema Validation	PASS	None
Valid Key Check in Insight Setup Data	PASS	None
Duplicate Key Check In Insight Setup Data	PASS	None
CVIM/Insight Workspace Conflict Check	PASS	None
Check Registry Connectivity	PASS	None
Check LDAP Connectivity	PASS	None
Test Email Server for Insight	PASS	None

Downloading VIM Insight Artifacts, takes time!!!

Cisco VIM Insight Installed successfully!

Description	Status	Details
VIM Insight UI URL	PASS	https://<br_api:9000>

```
| VIM UI Admin Email ID | PASS      | Check for info @: <abs path of insight_setup_data.yaml> |
|                      |          |
| VIM UI Admin Password | PASS      | Check for info @ /opt/cisco/insight/secrets.yaml           |
| VIM Insight Workspace | PASS      | /root/Insight-<tag_id>/insight/                           |
+-----+-----+-----+-----+
```

Cisco VIM Insight backup Info!

Description	Status	Details
Insight backup Status	PASS	Backup done @ /var/cisco/insight_backup/insight_backup_<release_tag>_<date_time>

Cisco VIM Insight Autobackup Service Info!

Description	Status	Details
VIM Insight Autobackup	PASS	[ACTIVE]: Running 'insight-autobackup.service'

Done with VIM Insight install!
VIM Insight install logs are at: "/var/log/insight/bootstrap_insight/"

Logs of Insight Bootstrap are generated at : /var/log/insight/bootstrap_insight/ on the management node. Log file name for Insight Bootstrap are in the following format : bootstrap_insight_<date>_<time>.log. Only ten bootstrap Insight log files are displayed at a time. Once the bootstrap process is completed a summary table preceding provides the information of the UI URL and the corresponding login credentials. After first login, for security reasons, we recommend you to change the Password.

Insight autobackup takes place after an install and is located at default backup location /var/cisco/insight_backup; details of which is provided in the backup summary table.

To add a new UI Admin in a setup that just got created, login to VIM insight and add a new UI admin user from the Manage UI Admin Users menu. Without doing a fresh install (that is un-bootstrap, followed by bootstrap) of the insight application, the UI admin that was bootstrapped cannot be changed.

Refer Cisco VIM Insight Post Bootstrap Validation Checks section, to verify the bootstrap status of Cisco VIM Insight.

Installing Cisco VIM Unified Management with Cisco VIM Software Hub

To reduce the logistics of the artifact distribution during an air-gapped installation, use Cisco VIM Software Hub. To download the artifacts to the Cisco VIM Software Hub server, follow the instructions available at [Installing Cisco VIM Software Hub in Air-Gapped Mode, on page 99](#). Then, you can use the connected way of installing Unified Management (UM) on the UM node.

To install UM on the UM node through Cisco VIM Software Hub, you need RESGTRY_NAME as an additional field in the setup data for the UM node.

```
REGISTRY_NAME: '<registry_name>' #Mandatory Parameter when SDS is enabled.
```

For example, registry FQDN name [your.domain.com]. When Cisco VIM Software Hub is not enabled, this parameter must not be used.

Once REGISTRY_NAME is defined in the setup data, the UM software fetches the artifacts from the Cisco VIM Software Hub server as long as the INSTALL_MODE is defined to be connected or not defined in the insight_setup_data.yaml file. By default, it is assumed to be connected.

Installing Cisco VIM Unified Management with LDAP

Insight supports both LDAP and LDAPS (Secure over SSL) for an AD (Active Directory) environment. You can choose only one at a time.

LDAPS supports connection using both self-signed and CA-signed certificate. You can choose any type of certificate for LDAPS.

- Selecting self-signed certificate option will require a certificate for verification over LDAPS and to make a secure connection to LDAP over SSL.
- No certificate is required when selecting CA-signed certificate option.

The following are the required keys in setup data for LDAP support:

- LDAP_MODE: < True or False >
- LDAP_SERVER: < IP address of LDAP server >
- LDAP_PORT: < Port no. >
- LDAP_BASE_DN: <DN tree for Groups>
- LDAP_SECURE: < True or False >
- LDAP_USER_ID_ATTRIBUTE: <'uid' or 'cn' or 'mail'>

Following optional key is required in the setup_data file, when LDAP_SECURE is True and a self-signed certificate is used:

LDAP_CERT_PATH: < Path of cert file >

Following optional keys are required in the setup_data file, when LDAP server is configured to support simple binding:

- LDAP_ADMIN: < User-Name of Admin user >
- LDAP_ADMIN_PASSWORD: < Password of user Admin >
- LDAP_GROUP_SEARCH_FILTER: < Filter to search LDAP-Group on Server >
- LDAP_GROUP_USER_SEARCH_FILTER: < Filter to search user in LDAP-Group >

Installing Cisco VIM Unified Management Without SMTP

By default, a SMTP infrastructure is required for Cisco VIM Unified Management service.

For releases starting from Cisco VIM 2.4.2, the Unified Management service is supported in the absence of SMTP.



Note The migration of the Unified Management service to SMTP enabled mode from the mode which does not require SMTP, is not supported.

To install Unified Management without SMTP, follow the below steps:

Step 1 Modify the insight_setup_data.yaml file and add following key:

```
SMTP_MODE: False
```

Step 2 Remove the following keys from the insight_setup_data.yaml:

```
INSIGHT_SMTP_SERVER
INSIGHT_EMAIL_ALIAS
INSIGHT_SMTP_PORT and
INSIGHT_EMAIL_ALIAS_PASSWORD
```

Step 3 Save the yaml file and begin the installation from the insight dir:

```
#./bootstrap_insight.py -a install -f <path to insight_setup_data.yaml>
```

With SMTP disabled, bootstrap insight sets both the Super Admin and Pod Admin as the default user.

The user can login and register the Pod, but cannot perform the following:

- Add new user at POD Level.
- Add new Pod Admin.
- Add new Super Admin.

To add new user or update password for the existing user for Insight without SMTP, use the below script.

```
# ./user_populate.py --help
usage: user_populate.py [-h] [--username USERNAME] [--emailid EMAILID]
                       [--usertype USERTYPE] [--updatepass UPDATEPASS]
```

Optional arguments:

```
-h, --help
                  show the help message and exit
--username USERNAME, -u USERNAME
                  name of the user.
--emailid EMAILID, -e EMAILID
                  Email ID of the user.
--usertype USERTYPE, -t USERTYPE
                  User Type:
                  super_admin - User is Super User for Insight
                  pod_admin - User allowed to register new PODS
                  pod_user - User can only get associated with PODS
--updatepass UPDATEPASS, -p UPDATEPASS
                  Email ID of user whose password needs to be updated.
```

To add a user, enter the below command:

```
#./user_populate.py -u abc -e abc@abc.com -t pod_user
```

- Note**
- -t can take one of the following values such as **super_admin**, **pod_admin**, and **pod_user** as an argument.
 - If the user already exists, an error stating "User already exists" is displayed. If the user is new, the script prompts to enter a new password and confirmation password.

To use forgot password functionality, use the below command:

```
#./user_populate.py -p abc@abc.com
```

If the user is added or password has been changed using "-p" option, then on first login through Unified Management, the user is redirected to the **Change Password** page.

Installing Cisco VIM Unified Management without Internet Access

Complete the following steps to install Cisco VIM Insight on the Cisco NFVI management node.

Management Node setup (without Internet):

For many service providers, the infrastructure on which management node setup is run is air-gapped. This presents an additional dimension for the orchestrator to handle. To support install that is air-gapped, refer to the section [Preparing for Installation on Servers Without InternetAccess](#) and follow the steps to prepare 64G USB 2.0. You can use USB 3.0 if the management node is based on M5.

Before you begin

You must complete all Cisco NFVI preparation tasks described in [Preparing for Cisco NFVI Installation](#) and the management node as described in [Cisco VIM Management Node Networking](#)

Step 1 Enter ip a to verify the br_mgmt and br_api interfaces are up and are bound to bond1 and bond0. For example:

```
$ ip a
br_api: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP link/ether 00:42:68:6f:79:f2
    brd ff:ff:ff:ff:ff:ff
inet nnn.nnn.nnn.nnn/25 brd nnn.nnn.nnn.nnn scope global br_api valid_lft forever preferred_lft
forever
inet6 fe80::3c67:7aff:fe9:6035/64 scope link valid_lft forever preferred_lft forever
bond1: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue master br_api state UP link/ether
    00:42:68:6f:79:f2 brd ff:ff:ff:ff:ff:ff
br_mgmt: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP link/ether 00:78:88:46:ee:6e
    brd ff:ff:ff:ff:ff:ff
inet nnn.nnn.nnn.nnn/24 brd nnn.nnn.nnn.nnn scope global br_mgmt valid_lft forever preferred_lft
forever
inet6 fe80::278:88ff:fe46:ee6e/64 scope link valid_lft forever preferred_lft forever
bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue master br_mgmt state UP
link/ether 00:78:88:46:ee:6e brd ff:ff:ff:ff:ff:ff
```

- Note**
- The br_mgmt and br_api interfaces are created when you install RHEL on the management node in [Installing the Management Node](#), on page 56

Step 2

Run the following commands to copy the installer directory and the standalone insight_setup_data.yaml.

- Copy the installer dir to another directory in /root/. The name of the new directory should start with Insight-

```
# cd /root/
# cp -r installer-<tag_id> Insight-<tag_id>
```

- Copy the Standalone insight_setup_data.yaml.Standalone_EXAMPLE file from the Insight-dir/openstack-configs to any other location on the management node or the BOM.

```
# cp /root/Insight-<tag_id>/openstack-configs/insight_setup_data.yaml.Standalone_EXAMPLE
/root/insight_setup_data.yaml
```

Step 3

Modify the insight setup data according to the requirements. Refer to the insight_setup_data.yaml and cert generation as listed in step 5 of the preceding section.

Step 4

Save the edited insight_setup_data.yaml file.

Step 5

Run Import Artifacts:

```
$ cd /root/insight-<tag_id>/tools
./import_artifacts.sh
```

This verifies that /var/cisco/artifacts on the management node has the following Insight artifacts, along with the other components 'insight-K9.tar', 'mariadb-app-K9.tar'.'

Step 6

Start the insight installation process.

```
$ cd /root/Insight-<tag_id>/insight/
$./bootstrap_insight.py --help
usage: bootstrap_insight.py [-h] --action ACTION

[--regenerate_secrets] [--setpassword]
[--file INSIGHTSETUPDATA] [--keep] [--verbose]
[--backupdir BACKUPDIR] [-y]

Insight install setup helper.

optional arguments:
-h, --help            show this help message and exit
--action ACTION, -a ACTION
                      install - Install Insight UI
                      install-status - Display Insight Install Status
                      reconfigure - reconfigure - Reconfigure Insight DB password,
                      TLS Certificate, INSIGHT_SSMTP_SERVER,
                      INSIGHT_EMAIL_ALIAS_PASSWORD,
                      INSIGHT_EMAIL_ALIAS, INSIGHT_SSMTP_PORT
                      LDAP_MODE, LDAP_SERVER, LDAP_PORT, LDAP_ADMIN
                      LDAP_ADMIN_PASSWORD, LDAP_BASE_DN, LDAP_SECURE
                      LDAP_CERT_PATH, LDAP_USER_ID_ATTRIBUTE, SSL_CERT_CHAIN_FILE, LDAP_GROUP_SEARCH_FILTER,
                      LDAP_GROUP_USER_SEARCH_FILTER
                      update - Update Insight UI
                      update-status - Display Insight Update Status
                      rollback - Rollback Insight UI update
                      commit - Commit Insight UI update
                      backup - Backup Insight UI
                      uninstall - Uninstall Insight UI
--regenerate_secrets, -r
                      System generated INSIGHT_DB_PASSWORD
--setpassword, -s      User supplied INSIGHT_DB_PASSWORD,
--file INSIGHTSETUPDATA, -f INSIGHTSETUPDATA
                      Location of insight_setup_data.yaml
--keep, -k             Preserve Insight artifacts during uninstall
--verbose, -v          Verbose on/off
--backupdir BACKUPDIR, -b BACKUPDIR
```

```
-y, --yes           Path to backup Insight
                   Option to skip reconfigure or uninstall steps without prompt
```

```
$ ./bootstrap_insight.py -a install -f </root/insight_setup_data.yaml> Insight Schema Validation
would be initiated:
```

```
VIM Insight install logs are at: /var/log/insight/<bootstrap_insight_<date>_<time>.log
```

```
Management Node Validations!
```

Rule	Status	Error
Check Kernel Version PASS None		
Check Ansible Version PASS None		
Check Docker Version PASS None		
Check Management Node Tag PASS None		
Check Bond Intf. Settings PASS None		
Root Password Check PASS None		
Check Boot Partition Settings PASS None		
Check LV Swap Settings PASS None		
Check Docker Pool Settings PASS None		
Check Home Dir Partition PASS None		
Check Root Dir Partition PASS None		
Check /var Partition PASS None		
Check LVM partition PASS None		
Check RHEL Pkgs Install State PASS None		

```
Insight standalone Input Validations!
```

Rule	Status	Error
Insight standalone Schema Validation PASS None		
Valid Key Check in Insight Setup Data PASS None		
Duplicate Key Check In Insight Setup Data PASS None		
CVIM/Insight Workspace Conflict Check PASS None		
Check Registry Connectivity PASS None		
Check LDAP Connectivity PASS None		
Test Email Server for Insight PASS None		

```
Setting up Insight, Kindly wait!!!
```

```
Cisco VIM Insight Installed successfully!
```

Description	Status	Details
VIM Insight UI URL PASS https://<br_api:9000>		
VIM UI Admin Email ID PASS Check for info @: <abs path of insight_setup_data.yaml>		
VIM UI Admin Password PASS Check for info @ /opt/cisco/insight/secrets.yaml		
VIM Insight Workspace PASS /root/Insight_<tag_id>/insight/		

```
Cisco VIM Insight backup Info!
```

Description	Status	Details
Insight backup Status PASS Backup done @		

```
|           | /var/cisco/insight_backup/insight_backup_<release_tag>_<date_time>|
+-----+-----+
```

Done with VIM Insight install!
VIM Insight install logs are at: /var/log/insight/bootstrap_insight/

Logs of Insight Bootstrap is generated at : /var/log/insight/bootstrap_insight/ on the management node. Log file name for Insight Bootstrap is in the following format : bootstrap_insight_<date>_<time>.log. Only ten bootstrap Insight log files are displayed at a time. Once the bootstrap process is completed a summary table preceding provides the information of the UI URL and the corresponding login credentials. After first login, for security reasons, we recommend you to change the Password.

Insight autobackup takes place after an install and is located at default backup location /var/cisco/insight_backup; details of which is provided in the backup summary table.

To add a new UI Admin in a setup that just got created, login to VIM insight and add a new UI admin user from the Manage UI Admin Users menu. Without doing a fresh install (that is un-bootstrap, followed by bootstrap) of the insight application, the UI admin that was bootstrapped with cannot be changed.

Refer Cisco VIM Insight Post Bootstrap Validation Checks , on page 128 to verify the bootstrap status of Cisco VIM Insight.

Installing Cisco VIM Unified Management with Optional Services

For releases from Cisco VIM 3.2.0, Cisco VIM Unified Management service provides the following as optional features:

- Automatically add each UM-admin as the default pod-user with Full-Pod-Access to a pod during pod-registration.
- Display all the pod-users as suggested users, while registering a new pod-user.



Note By default, these features are set to False. To use these features, change the value of corresponding keys to True in Insight setup data file.

To install Unified Management with these features, follow the below steps:

Step 1

Modify the insight_setup_data.yaml file and add following key:

- To automatically add each UM admin to pod with Full-Pod-Access during pod registration, set the following key with True as value:

UM_ADMIN_AS POD ADMIN: True

- To display the suggested users during pod-user registration, set the following key with True as value:

DISPLAY_ALL POD USERS: True

Step 2

Save the yaml file and begin the installation from the insight directory:

```
#./bootstrap_insight.py -a install -f <path to insight_setup_data.yaml>
```

Cisco VIM Insight Post Bootstrap Validation Checks

- After the VIM Insight bootstrap, you can view the status of Insight installation through install-status action using bootstrap.

```
$ Cisco VIM Insight Install Status!
+-----+-----+
| Description | Status | Details |
+-----+-----+
| VIM Insight Setup | PASS | Success |
| VIM Insight Version | PASS | <release_tag> |
| VIM Insight UI URL | PASS | https://<br_api:9000> |
| VIM Insight Container | PASS | insight_<tag_id> |
| VIM Mariadb Container | PASS | mariadb_<tag_id> |
| VIM Insight Autobackup| PASS | [ACTIVE]: Running 'insight-autobackup.service' |
| VIM Insight Workspace | PASS | /root/installer-<tag_id>/insight |
+-----+-----+
```

- You can also verify if the Insight and MySQL containers are up or not by running the following command:

```
$ docker ps -a
CONTAINER ID        IMAGE
COMMAND             CREATED          STATUS          NAMES
cbe582706e50        cvim-registry.com/mercury-rhel7-osp10/insight:7434
"/start.sh"         10 hours ago    Up 10 hours    insight_7321
68e3c3a19339        cvim-registry.com/mercury-rhel7-osp10/mariadb-app:7434
"/usr/bin/my_init /ma" 10 hours ago Up      10 hours    mariadb <tag-id>
```

- Check the status of Insight by running the following command :

```
$ systemctl status docker-insight
docker-insight.service - Insight Docker Service
Loaded: loaded (/usr/lib/systemd/system/docker-insight.service; enabled; vendor preset: disabled)
Active: active (running) since Fri 2017-04-07 13:09:25 PDT; 36s ago Main PID: 30768
(docker-current)
Memory: 15.2M
CGroup: /system.slice/docker-insight.service
└─30768 /usr/bin/docker-current start -a insight_<tag-id>

Apr 07 13:09:26 i11-tb2-ins-3 docker[30768]: Tables_in_rbac
Apr 07 13:09:26 i11-tb2-ins-3 docker[30768]: buildnode_master
Apr 07 13:09:26 i11-tb2-ins-3 docker[30768]: permission_master
Apr 07 13:09:26 i11-tb2-ins-3 docker[30768]: role_master
Apr 07 13:09:26 i11-tb2-ins-3 docker[30768]: role_permission
Apr 07 13:09:26 i11-tb2-ins-3 docker[30768]: user_master
Apr 07 13:09:26 i11-tb2-ins-3 docker[30768]: user_role
Apr 07 13:09:26 i11-tb2-ins-3 docker[30768]: user_session
Apr 07 13:09:26 i11-tb2-ins-3 docker[30768]: Starting the apache httpd
Apr 07 13:09:26 i11-tb2-ins-3 docker[30768]: AH00558: httpd: Could not reliably determine
the server's fully qualified domain name, using 2.2.2.6.
Set the 'ServerName' directive gl... this message
Hint: Some lines were ellipsized, use -l to show in full.
```

- Check if the Insight is up by running the following command:

```

$curl https://br_api:9000 -k (or --insecure)
Your response of curl should show the DOCTYPE HTML:
<!DOCTYPE html>
<!--[if lt IE 7]>      <html lang="en" ng-app="myApp" class="no-js lt-ie9 lt-ie8 lt-ie7">
<![endif]-->
<!--[if IE 7]>        <html lang="en" ng-app="myApp" class="no-js lt-ie9 lt-ie8">
<![endif]-->
<!--[if IE 8]>        <html lang="en" ng-app="myApp" class="no-js lt-ie9"> <![endif]-->
<!--[if gt IE 8]><!--> <html lang="en" ng-app="mercuryInstaller" class="no-js">
<!--<![endif]-->
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Cisco VIM Installer</title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=0"/>
    <link rel="stylesheet"
href="..../static/lib/html5-boilerplate/dist/css/normalize.css">
    <link rel="stylesheet" href="..../static/lib/html5-boilerplate/dist/css/main.css">

    <link rel="stylesheet" href="..../static/lib/bootstrap/bootstrap.min.css">
    <link rel="stylesheet" href="..../static/lib/font-awesome/font-awesome.min.css">
    <!--<link
href="http://maxcdn.bootstrapcdn.com/font-awesome/4.1.0/css/font-awesome.min.css"
rel="stylesheet">-->
        <link rel="stylesheet" href="..../static/lib/bootstrap/bootstrap-theme.min.css">
        <link rel="stylesheet" href="..../static/lib/uigrid/ui-grid.min.css">
        <link rel="stylesheet" href="..../static/lib/chart/angular-chart.min.css">
        <script
src="..../static/lib/html5-boilerplate/dist/js/vendor/modernizr-2.8.3.min.js"></script>
        <link rel="stylesheet" href="..../static/css/app.css">
        <!--new dashboard css starts-->
        <link rel="stylesheet" href="..../static/css/dashboard.css">
        <!--new dashboard css end-->
    </head>
    <body class="skin-blue sidebar-collapse" ng-controller="DashboardCtrl"
id="ToggleNavbar">
        <div class="wrapper" id="wrapper">

            <div class="content-wrapper" id="contentclass">
                <mi-header></mi-header>
                <mi-left-side-navbar></mi-left-side-navbar>
                <message-box> </message-box>
                <div class="viewheight" ng-view autoscroll="true"></div>
            </div>

            <mi-footer></mi-footer>
        </div>
        <!--new dashboard js starts-->
        <script src="..../static/lib/bootstrap/jquery.min.js"></script>
        <script src="..../static/lib/jquery/jquery-ui.js"></script>
        <script src="..../static/lib/bootstrap/progressbar.js"></script>
        <!--new dashboard js ends-->
        <script src="..../static/lib/chart/Chart.min.js"></script>
        <script src="..../static/lib/bootstrap/bootstrap.min.js"></script>
        <script src="..../static/lib/angular/angular.js"></script>
        <script src="..../static/lib/chart/angular-chart.min.js"></script>
        <script src="..../static/lib/uigrid/angular-touch.js"></script>
        <script src="..../static/lib/uigrid/angular-animate.js"></script>
        <script src="..../static/lib/uigrid/csv.js"></script>
        <script src="..../static/lib/uigrid/pdfmake.js"></script>

```

```
<script src="../static/lib/uigrid/vfs_fonts.js"></script>
<script src="../static/lib/uigrid/ui-grid.js"></script>
<script src="../static/lib/angular/smrt-table.min.js"></script>
<script src="../static/lib/angular-route/angular-route.js"></script>
<script src="../static/lib/angular-cookies/angular-cookies.js"></script>
<script src="../static/lib/angular/angular-translate.js"></script>
<script
src="../static/lib/angular/angular-translate-loader-static-files.min.js"></script>
<script
src="../static/lib/angular/angular-translate-storage-cookie.min.js"></script>
<script
src="../static/lib/angular/angular-translate-storage-local.min.js"></script>
<script src="../static/lib/yamltojson/yaml.js"></script>
<script src="../static/lib/yaml/js-yaml.min.js"></script>
<script src="../static/lib/d3/d3min.js"></script>
<script src="../static/utility/utility.js"></script>
<script src="../static/widgets/widgets.js"></script>
<script src="../static/app.js"></script>
<script src="../static/layout/layout.js"></script>
<script src="../static/login/login.js"></script>
<script src="../static/globals/globals.js"></script>
<script src="../static/dashboard/dashboard.js"></script>
<script src="../static/cloudpulse/cloudpulse.js"></script>
<script src="../static/blueprintsetup/physicalsetupwizard/ucsmcommon.js"></script>
<script src="../static/blueprintsetup/physicalsetupwizard/cimccommon.js"></script>
<script src="../static/vmtp/runvmtp.js"></script>
<script src="../static/blueprintsetup/physicalsetupwizard/networking.js"></script>

<script
src="../static/blueprintsetup/physicalsetupwizard/serverandroles.js"></script>
<script src="../static/blueprintsetup/openstacksetupwizard/cephsetup.js"></script>

<script
src="../static/blueprintsetup/openstacksetupwizard/cindersetup.js"></script>
<script
src="../static/blueprintsetup/openstacksetupwizard/glancesetup.js"></script>
<script src="../static/blueprintsetup/openstacksetupwizard/haproxy.js"></script>

<script
src="../static/blueprintsetup/openstacksetupwizard/keystonesetup.js"></script>
<script
src="../static/blueprintsetup/openstacksetupwizard/swiftstack.js"></script>
<script
src="../static/blueprintsetup/openstacksetupwizard/neutronsetup.js"></script>
<script src="../static/blueprintsetup/openstacksetupwizard/vmtpsetup.js"></script>

<script
src="../static/blueprintsetup/physicalsetupwizard/physicalsetupwizard.js"></script>
<script src="../static/blueprintsetup/servicesSetupWizard/systemlog.js"></script>
<script src="../static/blueprintsetup/servicesSetupWizard/NFVbench.js"></script>

<script
src="../static/blueprintsetup/servicesSetupWizard/servicesSetupWizard.js"></script>
<script
src="../static/blueprintsetup/openstacksetupwizard/openstacksetupwizard.js"></script>
<script src="../static/blueprintsetup/blueprintsetup.js"></script>
<script src="../static/blueprintmanagement/blueprintmanagement.js"></script>
<script src="../static/topology/topology.js"></script>
<script src="../static/monitoring/monitoring.js"></script>
```

```

<script src="../static/horizon/horizon.js"></script>
<script src="../static/podmanagement/podmanagement.js"></script>
<script
src="../static/blueprintsetup/openstacksetupwizard/tlssupport.js"></script>
<script src="../static/blueprintsetup/openstacksetupwizard/elksetup.js"></script>

<script src="../static/systemupdate/systemupdate.js"></script>
<script
src="../static/blueprintsetup/physicalsetupwizard/registrysetup.js"></script>
<script src="../static/registertestbed/registertestbed.js"></script>
<script src="../static/registerasaas/registerasaas.js"></script>
<script src="../static/useradministration/manageusers.js"></script>
<script src="../static/useradministration/rolemanagement.js"></script>
<script src="../static/saasadminashboard/saasadminashboard.js"></script>
<script src="../static/saasadminashboard/buildnodes.js"></script>
<script src="../static/saasadminashboard/buildnodeusers.js"></script>
<script src="../static/saasadminashboard/managesaasuser.js"></script>
<script src="../static/saasadminusermanagement/saasadminusermgmt.js"></script>
<script src="../static/blueprintsetup/physicalsetupwizard/nfvsetup.js"></script>

<script src="../static/blueprintsetup/physicalsetupwizard/torswitch.js"></script>

<script src="../static/blueprintsetup/openstacksetupwizard/vtssetup.js"></script>

<script src="../static/rbacutilities/rbacutility.js"></script>
<script src="../static/forgotpassword/forgotpassword.js"></script>
<script src="../static/changepassword/changepassword.js"></script>
<script src="../static/passwordreconfigure/passwordreconfigure.js"></script>
<script
src="../static/openstackconfigreconfigure/openstackconfigreconfigure.js"></script>
<script
src="../static/reconfigureoptionalservices/reconfigureoptionalservices.js"></script>
</body>

```

- VIM Insight Autobackup: Insight will invoke Insight Autobackup as a daemon process. Autobackup is taken as an incremental backups of database and /opt/cisco/insight/mgmt_certs dir if there is any change.

You can check the status of Insight Autobackup service:

```

systemctl status insight-autobackup
insight-autobackup.service - Insight Autobackup Service
   Loaded: loaded (/usr/lib/systemd/system/insight-autobackup.service; enabled; vendor
preset: disabled)
     Active: active (running) since Mon 2017-09-04 05:53:22 PDT; 19h ago
       Process: 21246 ExecStop=/bin/kill ${MAINPID} (code=exited, status=0/SUCCESS)
      Main PID: 21287 (python)
        Memory: 9.2M
          CGroup: /system.slice/insight-autobackup.service
                  └─21287 /usr/bin/python
/var/cisco/insight_backup/insight_backup_2.1.10_2017-08-31_03:02:06/root
/rohan/installer-10416/insight/playbooks/../insight_autobackup.py

Sep 04 05:53:22 F23-insight-4 systemd[1]: Started Insight Autobackup Service.
Sep 04 05:53:22 F23-insight-4 systemd[1]: Starting Insight Autobackup Service...

```

VIM UM Admin Login for Standalone Setup

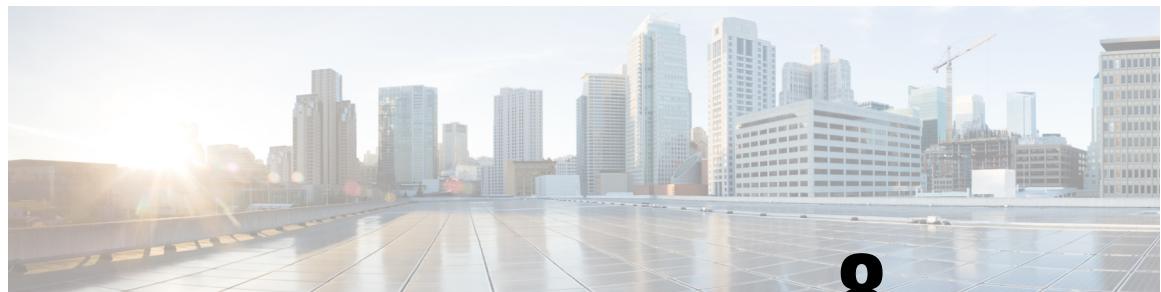
For security reasons, the Insight Admin logs in to the UI with which UM is bootstrapped and Add users. Insight Admin needs to add new users as Pod Admin.

Registration of UM Admin to UM

-
- Step 1** Enter the following address on the browser: `https://<br_api>:9000`.
- Step 2** Enter the **Email ID** and **Password**. The Email ID should be the one specified as 'UI_ADMIN_EMAIL_ID' in `insight_setup_data.yaml` during bootstrap. The Password for UI Admins are generated at: `/opt/cisco/insight/secrets.yaml` and key is 'UI_ADMIN_PASSWORD'. If LDAP mode is True and LDAP user attribute is set to uid, login with LDAP user id credentials.
- Step 3** Click **Login as UI Admin User**. You will be redirected to Insight UI Admin Dashboard.
-

VIM UM Pod Admin Login for Standalone Setup

- Step 1** Log in as Insight UM.
- Step 2** Navigate to **Manage Pod Admin** and click **Add Pod Admin**.
- Step 3** Enter a new Email ID in **Add Pod Admin** pop-up.
- Step 4** Enter the username of the Pod Admin.
- Step 5** Click **Save**. User Registration mail is sent to a newly added Pod Admin with a token.
- Step 6** Click the URL with token and if token is valid then Pod Admin is redirected to Insight-Update Password page.
- Step 7** Enter new password and then confirm the same password.
- Step 8** Click **Submit**.
-



CHAPTER 8

Installing Cisco VIM through Cisco VIM Unified Management

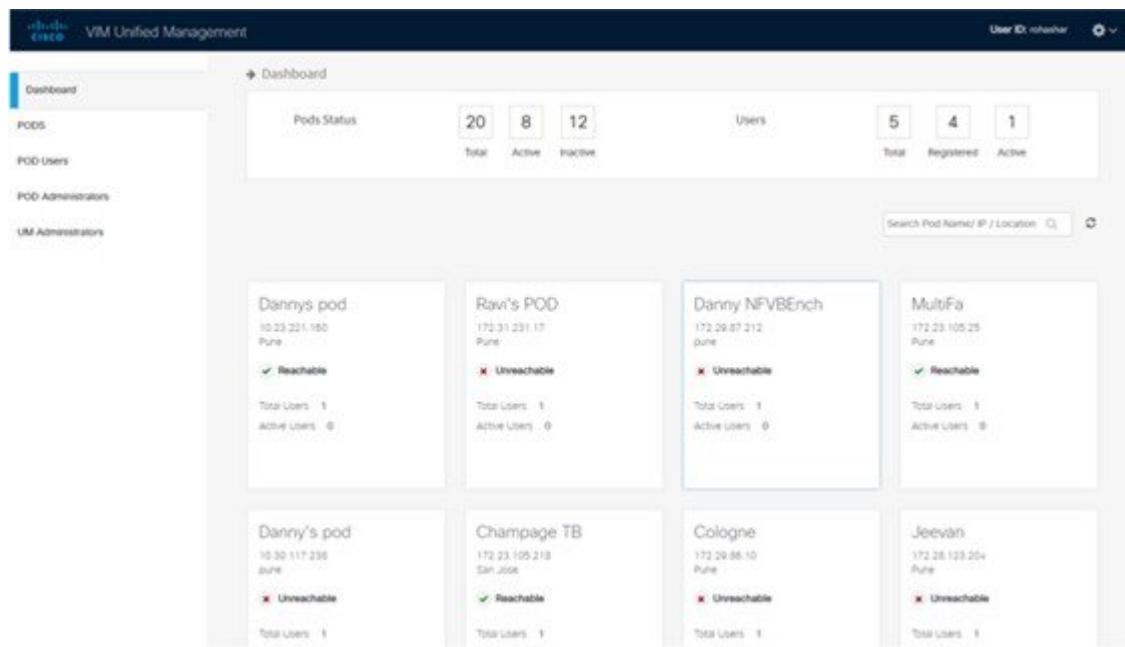
The VIM UM has an UI admin, who has the privilege to manage the UI offering. The Insight UI admin, has the rights to add the right users as Pod administrators. Post bootstrap, the URL for the UI will be: https://br_api:9000.

The following topics helps you to install and configure Cisco Virtual Infrastructure Manager with VIM Insight:

- [Unified Management Dashboard, on page 229](#)
- [Pods, on page 230](#)
- [Pod Administrator, on page 232](#)
- [Unified Management \(UM\) Administrator, on page 233](#)
- [Registering New Pod to Insight , on page 234](#)
- [Configuring OpenStack Installation, on page 235](#)
- [Post Installation Features for Active Blueprint, on page 337](#)

Unified Management Dashboard

When you login as UM admin, you will be redirected to the UM admin Dashboard.



The UM dashboard displays the following information about the pods it is currently managing:

Pod Status

- Active - Number of Pods which has health status OK (Example: Mgmt Node health of the pod is good).
- Inactive - Number of Pods whose health status is not good (Example: Mgmt Node health of the pod is not good).
- Total number of Pods - Number of Pods registered in the system.

Pod Users

- Total – Total number of users registered who are associated with at-least one Pod.
- Registered – Number of users who have completed the registration process and are associated with at-least one Pod.
- Active – Number of Online users who are associated with at-least one Pod.

You can see the list of Pod with its Pod name, description, IP address, location, Pod status along with the Total users and Active users of each pod. You can search for a Pod using Name, IP and location in the search option.

If you click **Get health of current nodes icon (spin)** it does the health check of the Pod.

Pods

Pods allows you to check the health status (indicated through green and red dot) of the pod respectively.

To fetch the latest health status, click **Refresh** which is at the upper right corner.

- Green dot – Pod is reachable and health is good.

- Red dot – Pod is not reachable.

Pod Users

The Pod Users page, gives you the details associated the pod, status (Online or Offline) and their Roles.

UM admin has the right to manage all Pod users in the system. The user with UM admin access can manage the following actions:

- Revoke User's permission from a specific Pod.
- Delete User from the system.

User Name	Email	IP Address	Role Name	Online	Action
Rohan R	rohashar@cisco.com	10.30.116.244	Full-Pod-Access	Online	
Rohan R	rohashar@cisco.com	172.28.123.204	Full-Pod-Access	Offline	
Rohan R	rohashar@cisco.com	10.30.117.238	Full-Pod-Access	Offline	
Rohan R	rohashar@cisco.com	10.23.229.228	Full-Pod-Access	Offline	

Revoking User

UM admin revokes the user's permission from a Pod by clicking **(undo)** icon. If the user is the only user with a Full-Pod-Access role for that particular Pod, then the revoke operation is not permitted. In this case, another user is granted with a Full-Pod-Access role for that Pod and then proceeds with revoking the old user.



Note If the user is revoked from the last associated Pod, then the user is deleted from the system.

Deleting Users

UM admin can delete any user from the system by clicking **X** from an Action column. The delete operation is not permitted if the user has Full-Pod-Access. In such case another user is granted with *Full-Pod-Access* role for that Pod and then proceed with deleting the old user. UM admin must revoke respective permission first and then proceed further.

Pod Administrator

Pod admins are the users who has the permission to register new Pods in the system. UM admin can add any number of Pod admins in the system.

User Name	Email	Action
Rohan R	rohashan@cisco.com	
Aniket C	achotthe@cisco.com	

Adding Pod Admin

- Step 1** Log in as **UI Admin** and navigate to POD Administrator page.
- Step 2** Click **Add Pod Administrator**.
- Step 3** Select User auth for the new user. This option is enabled only if LDAP mode is true.
- Step 4** Enter the Email ID/LDAP user id (if LDAP user attribute is set to uid) of the user.
 - If the email is already registered, the **Username** gets populated automatically.
 - If the email is not registered, an email is sent to the user email ID with the verification token. If User auth is set as LDAP, no verification token email is sent.
- Step 5** Navigate to https://br_api:9000.
- Step 6** Enter the **Email ID** and **Password** of the Pod Admin
- Step 7** Click **Login as Pod User**. It redirects to the landing page where the Pod admin can register a new Pod.

Revoking Pod Admin

UM admin can revoke Pod admin's permission anytime. To revoke Pod admin permission for the user, click **undo** icon.



Note You cannot revoke self permission.

Unified Management (UM) Administrator

UM admins have the access to the UM profile. Only a UM admin can add another UM admin in the system. There should be at least one UM admin in the system.

User Name	Email	Online	Action
Rohan R	rohashar@cisco.com	Online	

Adding UM Admin

To add a UM admin perform the following steps.

Step 1 Log in as **UI Admin** and navigate to UM Administrator page.

Step 2 Click **Add UM Administrator**.

Step 3 Select User auth for the new user. This option is enabled only if LDAP mode is true.

Step 4 Enter the Email ID/ LDAP user id (if LDAP user attribute is set to uid) of the user.

- If email is already registered, the **Username** gets populated automatically.
- If email is not registered, an email is sent to the user email ID with the verification token. If User auth is set as LDAP, no verification token email is sent.

Step 5 Navigate to https://br_api: 9000.

Step 6 Enter the Email ID and Password of the UM Admin.

Step 7 Click **Log in as UM admin** to view the UM dashboard.

Revoking UM Admin

UM admin can revoke another UM admin's permission. To revoke UM Admin permission for any user, click **undo** icon.



Note You cannot revoke a self's permission. You can revoke a user if the user is not associated with any pod. After revoking the user is deleted from the system.

Registering New Pod to Insight

Following are the steps that are required for UI Admin to register a Pod Admin:

Before you begin

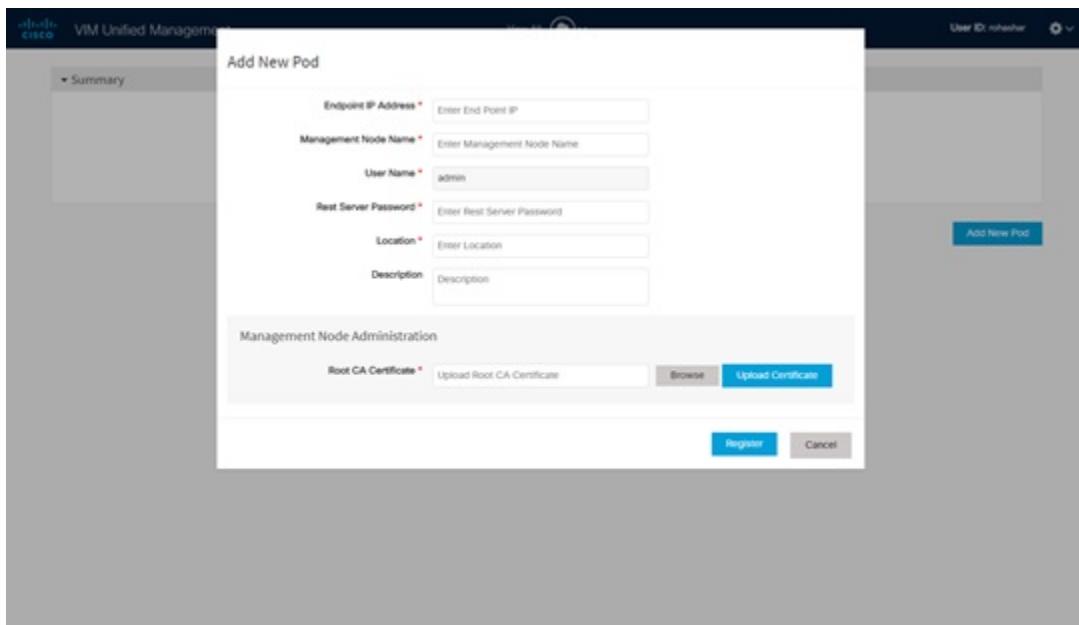
UI Admin has to register a Pod Admin to allow the user to access a pod.

Step 1 Log in as **UM Administrator**.

Step 2 Navigate to Pod Administrator and click **Add Pod Admin**.

Step 3 Enter the Email ID and the Password of the Pod Admin and click **Login as Pod User**. Then, you will be redirected to the landing page.

Step 4 Click **Add New Pod** to register a Pod. The **Add New Pod** popup window appears on the screen.



Step 5 Enter the br_api of the pod management node as the **Endpoint IP Address** and **Rest Server Password** from the file /opt/cisco/ui_config.json.

Step 6 Enter the values for the remaining fields in **Add New Pod**.

Step 7 Click **Browse** to select the Root CA certificate.

For more information on Root CA certificate, see [Managing Root CA Certificate](#)

Step 8 Click **Upload Certificate** to upload the selected Root CA certificate.

Step 9 Click **Register** to start the Pod registration.

The newly created Pod appears on the landing page.

Configuring OpenStack Installation

Before you begin

You need to create a Blueprint (B or C Series) to initiate OpenStack Installation through the VIM.

Step 1 In the navigation pane, choose **Pre-Install > Blueprint Setup**.

Step 2 To create a **B Series Blueprint**:

- On the **Blueprint Initial Setup** pane of the Cisco VIM Insight, complete the following fields:

The screenshot shows the 'Blueprint Initial Setup' tab selected in the Cisco VIM Insight interface. The 'Blueprint Name' field is set to 'Enter Blueprint Name'. The 'Platform Type' dropdown is set to 'B-series'. The 'Tenant Network' dropdown is set to 'LinuxBridge/VXLAN'. The 'Pod Type' dropdown is set to 'Fullon'. Below these fields, there is a section titled 'Optional Features & Services' with several checkboxes. At the bottom, there is a 'Import Existing YAML file' section with 'Browse' and 'Load' buttons.

Name	Description
Blueprint Name field	Enter blueprint configuration name.
Platform Type drop-down list	Choose one of the following platform types: <ul style="list-style-type: none"> • B-Series (By default) choose B series for this section. • C-Series
Tenant Network drop-down list	Choose tenant network type: OVS/VLAN
Pod Type drop-down list	Choose one of the following pod types: <ul style="list-style-type: none"> • Fullon(By Default)

Name	Description
Ceph Mode drop-down list	Choose one of the following Ceph types: <ul style="list-style-type: none"> • Dedicated • Central (By Default) - Not supported in Production
Optional Features and Services Checkbox	Swiftstack, LDAP, Syslog Export Settings, Install Mode, ToR Switch Information, TLS, NFVMON, Pod Name, VMTP, NFV Bench, Auto-backup, Heat, Ceilometer, Keystone v3, Enable Esc Priv, Enable TTY logging, SNMP, ManagementNode_CloudAPI_Reachability. If any one is selected, the corresponding section is visible in various Blueprint sections. SNMP requires CVIM-MON to be enabled. By default, all features are disabled except Auto-backup and Management Node_CloudAPI_Reachability. Select Enable Read-only OpenStack Admins to add a custom role with read-only admin privileges to OpenStack resources.
Import Existing YAML file	Click Browse button to import the existing yaml file. If you have an existing B Series YAML file you can use this feature to upload the file. Unified Management automatically fill in the fields and if any mandatory field is missed then it gets highlighted in the respective section.

- b. Click **Physical Setup** to navigate to the **Registry Setup** configuration page. Fill in the following details for Registry Setup:

Name	Description
Registry User Name text field	Enter the User-Name for Registry (Mandatory).
Registry Password text field	Enter the Password for Registry (Mandatory).

Name	Description
Registry Email text field	Enter the Email ID for Registry (Mandatory).

Once all mandatory fields are filled the **Validation Check Registry Pane** shows a Green Tick.

- c. Click **UCSM Common Tab** and complete the following fields:

The screenshot shows the 'Create Blueprint configuration' wizard in the Cisco VIM Unified Management interface. The 'Physical Setup' tab is active. The 'UCSM Common' step is currently being configured. The form includes fields for User name (admin), Password (password), UCSM IP (UCSM IP), Resource Prefix (Resource Profile), Max VF Count (20), and QoS Policy Type (nfVt). At the bottom, there is a note about the version: ©2018 Cisco and/or its affiliates. All rights reserved. Cisco VIM Unified Management Version: 2.2.2.

Name	Description
User name disabled field	By default the value is Admin.
Password text field	Enter Password for UCSM Common (Mandatory).
UCSM IP text field	Enter IP Address for UCSM Common (Mandatory).
Resource Prefix text field	Enter the resource prefix(Mandatory).
QOS Policy Type drop-down	Choose one of the following types: <ul style="list-style-type: none"> • NFVI (Default) • Media
Max VF Count text field	Select the Max VF Count. <1-54> Maximum VF count 54, default is 20. If VF performance is enabled we recommend you to keep MAX_VF_COUNT to 20 else may fail on some VICs like 1240.

Name	Description
Enable VF Performance optional checkbox	Default is false. Set to true to apply adaptor policy at VF level.
Enable Prov FI PIN optional checkbox	Default is false.
MRAID-CARD optional checkbox	Enables JBOD mode to be set on disks. Applicable only if you have RAID controller configured on Storage C240 Rack servers.
Enable UCSM Plugin optional checkbox	Visible when Tenant Network type is OVS/VLAN.
Enable QoS Policy optional checkbox	Visible only when UCSM Plugin is enabled. If UCSM Plugin is disabled then this option is set to False.
Enable QOS for Port Profile optional checkbox	Visible only when UCSM Plugin is enabled.
SRIOV Multi VLAN Trunk optional grid	Visible when UCSM Plugin is enabled. Enter the values for network and vlans ranges. Grid can handle all CRUD operations such as Add, Delete, Edit and, Multiple Delete.

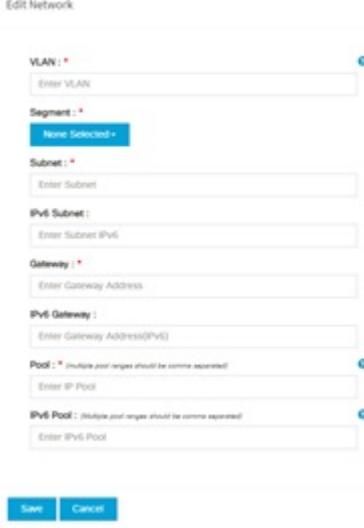
- d. Click **Networking** to advance to the networking section of the Blueprint:

The screenshot shows the Cisco VIM Unified Management web interface. The top navigation bar includes the Cisco logo, 'VIM Unified Management' title, and user information ('Role: Full-Post-Access | User ID: rohith'). The left sidebar has sections for 'Dashboard', 'Pre-Install' (selected), 'Blueprint Setup' (selected), 'Blueprint Management', 'Post-Install', 'View Topology', and 'Pod User Administration'. The main content area is titled 'Create Blueprint configuration' under 'Physical Setup'. A progress bar at the top indicates steps: Registry Setup (red), UCSM Common (red), Networking (blue, currently active), and Servers and Roles (red). The 'Networking' section contains fields for 'Domain Name' (with placeholder 'Enter Domain Name'), 'HTTP Proxy' (placeholder 'Enter HTTP Proxy'), and 'HTTPs Proxy' (placeholder 'Enter HTTPS Proxy'). Below these are three tables for 'IP Tables on Management Pods': 'IP Address' (empty), 'NTP Server' (empty), and 'Domain Name Server' (empty). At the bottom, there is a 'Networks' table with columns: 'View', 'Segment', 'Subnet', 'Subnet IPv6', 'Gateway', 'Gateway IPv6', 'Pool', 'Pool IPv6', and 'Action'. The table lists network entries like 'cmic', 'api', 'management/prov...', 'tenant', and 'storage', each with a delete icon ('X') in the 'Action' column. The footer of the page displays copyright information: '© 2018 Cisco and/or its affiliates. All rights reserved.' and 'Cisco VIM Unified Management Version: 2.2.2'.

Name	Description
Domain Name field	Enter the domain name (Mandatory).
HTTP Proxy Server field	If your configuration uses an HTTP proxy server, enter the IP address of the server.
HTTPS Proxy Server field	If your configuration uses an HTTPS proxy server, enter the IP address of the server.
IP Tables on Management Pods	Specifies the list of IP Address with Mask.
NTP Server	Enter a maximum of four and minimum of one IPv4 and /or IPv6 addresses in the table.
Domain Name Server	Enter a maximum of three and minimum of one IPv4 and/or IPv6 addresses.

Name	Description
Network options	<p>This section is accessible only if ToR type is Cisco NCS 5500.</p> <p>vxlan-tenant:</p> <ul style="list-style-type: none"> Provider network name: It is a unique name. BGP AS num: Takes value between 1 and 65535. BGP Peers: Enter the peer route reflector IPs (IPs to be comma separated) BGP router ID: The router ID is used for local GoBGP cluster. Head-end replication (Optional) : You can add VTEP IP address and comma separated VNI IDs. Multiple entries are allowed. <p>Note VXLAN-TENANT is allowed only when NETWORK_OPTIONS is vxlan network. The IPs defined belong to the vxlan-tenant network, but are not part of the vxlan-tenant network pool.</p> <p>VXLAN-ECN:</p> <ul style="list-style-type: none"> Provider network name: It is the unique name. BGP AS num: It takes the value between 1 and 65535. BGP Peers: Enter the peer route reflector IPs. (IPs to be comma separated) BGP router ID: The router ID is used for local GoBGP cluster. Head-end replication (Optional) : You can add VTEP IP address and comma separated VNI IDs. Multiple entries are allowed. <p>Note</p> <ul style="list-style-type: none"> You cannot have VXLAN-ECN without vxlan-tenant segment defined, however vxlan-tenant can be defined standalone. Ensure that you take care while choosing single or multi-VXLAN (two-VXLAN) option as this is a day-0 configuration. VXLAN_ECN is allowed only when NETWORK_OPTIONS is vxlan network. The IPs defined belong to the vxlan-ecn network, but are not part of the vxlan-ecn network pool.

Name	Description
Network table	

Name	Description						
	<p>Network table is pre-populated with segments. To add Networks you can either clear all the table using Delete All or click Edit icon for each segment and fill in the details.</p> <p>You can add, edit, or delete network information in the table:</p>  <ul style="list-style-type: none"> Click + to enter new entries (networks) to the table. Specify the following fields in the Edit Entry to Networks dialog box. <table border="1"> <thead> <tr> <th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>VLAN field</td><td>Enter the VLAN ID. For Segment - Provider, the VLAN ID value is always <i>none</i>.</td></tr> <tr> <td>Segment drop-down list</td><td>You can select any one segment from the drop-down list. <ul style="list-style-type: none"> API Management/Provision Tenant CIMC Storage External </td></tr> </tbody> </table>	Name	Description	VLAN field	Enter the VLAN ID. For Segment - Provider, the VLAN ID value is always <i>none</i> .	Segment drop-down list	You can select any one segment from the drop-down list. <ul style="list-style-type: none"> API Management/Provision Tenant CIMC Storage External
Name	Description						
VLAN field	Enter the VLAN ID. For Segment - Provider, the VLAN ID value is always <i>none</i> .						
Segment drop-down list	You can select any one segment from the drop-down list. <ul style="list-style-type: none"> API Management/Provision Tenant CIMC Storage External 						

Name	Description	
	Name	Description
		<p>• Provider (optional)</p> <p>Note Some segments do not need some of the values listed in the preceding points.</p>
	Subnet field	Enter the IPv4 address for the subnet.
	IPv6 Subnet field	Enter IPv6 address. This field is available only for Management provision and API.
	Gateway field	Enter the IPv4 address for the Gateway.
	IPv6 Gateway field	Enter IPv6 gateway. This field is available only for Management provision and API network.
	Pool field	Enter the pool information in the following format. For example: 10.30.1.1 or 10.30.1.1 to 10.30.1.12
	IPv6 Pool field	Enter the pool information in the following format. For example: 10.1.15-10.1.1.10,102.1.5-102.1.10 This field is only available for the Mgmt/Provision.
Click Save .		

- e. On the **Servers and Roles** page of the Cisco VIM Suite wizard, you see a pre-populated table filled with Roles: Control, Compute and Block Storage (Only if CEPH Dedicated is selected in Blueprint Initial Setup).

Configuring OpenStack Installation

Create Blueprint configuration

Blueprint Initial Setup Physical Setup OpenStack Setup

Server User Name: root
Disable Hyperthreading:

COBBLER :

- Cobbler Timeout: 45
- Control Kickstart: ucs-b-and-c-series.ks
- Server Host Password: Enter Server Host Password

Server and Roles :

Server Name	Server Type	Rack ID	Chassis ID	Blade ID	Rack unit ID	Role	Management IP	Management IPv6	Action
blade	blade					control			<input checked="" type="checkbox"/> <input type="button" value="X"/>
blade	blade					control			<input checked="" type="checkbox"/> <input type="button" value="X"/>
blade	blade					control			<input checked="" type="checkbox"/> <input type="button" value="X"/>
blade	blade					compute			<input checked="" type="checkbox"/> <input type="button" value="X"/>

©2018 Cisco and/or its affiliates. All rights reserved.
Cisco VIM Unified Management Version: 2.2.2

Name	Description
Server User Name field	Enter the username of the server.
Disable Hyperthreading	Default value is false. You can set it as true or false.

Name	Description	
Cobbler	Enter the Cobbler details in the following fields:	
Name	Description	
Cobbler Timeout field	The default value is 45 min. This is an optional parameter. Timeout is displayed in minutes, and its value ranges from 30 to 120.	
Block Storage Kickstart field	Kickstart file for Storage Node.	
Admin Password Hash field	Enter the Admin Password. Password must be Alphanumeric. Password should contain minimum 8 characters and maximum of 32 characters.	
Cobbler Username field	Enter the cobbler username to access the cobbler server.	
Control Kickstart field	Kickstart file for Control Node.	
Compute Kickstart field	Kickstart file for Compute Node.	
Cobbler Admin Username field	Enter the admin username of the Cobbler.	

Name	Description
Add Entry to Servers and Roles	

Name	Description																
	<p>Click Edit or + to add a new server and role to the table.</p> <p>Server And Roles</p> <p>Server Name * <input type="text"/> Enter Server Name</p> <p>VIC Slot <input type="text"/> Enter VIC Slot</p> <p>CIMC IP * <input type="text"/> Enter CIMC IP Address</p> <p>CIMC User Name <input type="text"/> Enter CIMC Username</p> <p>CIMC Password <input type="text"/> Enter CIMC Password</p> <p>Rack ID * <input type="text"/> Enter Rack ID</p> <p>Role * <input type="text"/></p> <p>Management IP <input type="text"/> Enter Management IP Address</p> <p>Management IPv6 <input type="text"/> Enter Management IPv6 Address</p> <p>Save Cancel</p> <table border="1"> <tr> <td>Server Name</td> <td>Enter a server name.</td> </tr> <tr> <td>Server Type drop-down list</td> <td>Choose Blade or Rack from the drop-down list.</td> </tr> <tr> <td>Rack ID</td> <td>The Rack ID for the server.</td> </tr> <tr> <td>Chassis ID</td> <td>Enter a Chassis ID.</td> </tr> <tr> <td>If Rack is chosen, the Rack Unit ID field is displayed.</td> <td>Enter a Rack Unit ID.</td> </tr> <tr> <td>If Blade is chosen, the Blade ID field is displayed.</td> <td>Enter a Blade ID.</td> </tr> <tr> <td>Select the Role from the drop-down list.</td> <td>If Server type is Blade then select Control and Compute. If server is Rack then select Block Storage.</td> </tr> <tr> <td>VIC Admin FEC mode</td> <td>Applicable only for Cisco VIC that supports to change the admin FEC mode. Can</td> </tr> </table>	Server Name	Enter a server name.	Server Type drop-down list	Choose Blade or Rack from the drop-down list.	Rack ID	The Rack ID for the server.	Chassis ID	Enter a Chassis ID.	If Rack is chosen, the Rack Unit ID field is displayed.	Enter a Rack Unit ID.	If Blade is chosen, the Blade ID field is displayed.	Enter a Blade ID.	Select the Role from the drop-down list.	If Server type is Blade then select Control and Compute . If server is Rack then select Block Storage .	VIC Admin FEC mode	Applicable only for Cisco VIC that supports to change the admin FEC mode. Can
Server Name	Enter a server name.																
Server Type drop-down list	Choose Blade or Rack from the drop-down list.																
Rack ID	The Rack ID for the server.																
Chassis ID	Enter a Chassis ID.																
If Rack is chosen, the Rack Unit ID field is displayed.	Enter a Rack Unit ID.																
If Blade is chosen, the Blade ID field is displayed.	Enter a Blade ID.																
Select the Role from the drop-down list.	If Server type is Blade then select Control and Compute . If server is Rack then select Block Storage .																
VIC Admin FEC mode	Applicable only for Cisco VIC that supports to change the admin FEC mode. Can																

Name	Description	
		be auto/off/cl74/cl91
VIC Port Channel Enable		Optional. By default, it is true. Can be either true or false.
Secure Computing mode		Optional. By default, it is set to 1, if not defined. Can be either 0 or 1.
Management IP		It is an optional field but if provided for one server then it is mandatory to provide details for other Servers as well.
Storage IP		It is an optional field, but if provided for one server then it is mandatory to provide details for other servers.
Management IPv6		Enter the Management IPv6 Address.
Vtep IPs		Two input fields for vxlan-tenant and vxlan-ecn ips are available, for any node having compute role, vxlan-tenant and vxlan-ecn in network option.
BGP management addresses		Two input fields for vxlan-tenant and vxlan-ecn ips, are available for any node having control role and having vxlan-tenant and vxlan-ecn in network option. IPs must be from management subnet, but not from the pool.
trusted_vf		Optional and not reconfigurable. Applicable only for SRIOV node with compute role for C-series pod.
	Click Save .	

- f. Click **ToR Switch** checkbox in **Blueprint Initial Setup** to enable the **TOR SWITCH** configuration page. It is an **Optional** section in Blueprint Setup, but when all the fields are filled it is a part of the Blueprint.

The screenshot shows the 'Blueprint Initial Setup' step in the Cisco VIM Unified Management interface. The 'Configure ToR' section is displayed, featuring a table for 'TorSwitch Information'. The 'ToR Switch' checkbox is checked, indicating it has been selected. The interface includes a navigation bar at the top with tabs for Blueprint Initial Setup, Physical Setup, and OpenStack Setup. The 'Physical Setup' tab is currently active. A progress bar at the bottom of the setup steps shows the 'ToR Switch' step is completed.

Name	Description
Configure ToR optional checkbox.	Enabling this checkbox, changes the configure ToR section from false to true.

Name	Description
ToR Switch Information mandatory table.	

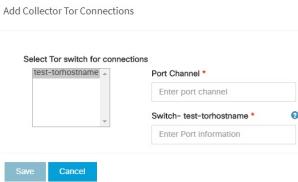
Name	Description																
	<p>Click (+) to add information for ToR Switch.</p> <p>Switch Details</p> <p>Hostname * <input type="text" value="Enter Switch Hostname"/> ?</p> <p>Username * <input type="text" value="Enter Switch Username"/> ?</p> <p>Password * <input type="text" value="Enter Password"/> ?</p> <p>SSH-IP * <input type="text" value="Enter IP Address"/> ?</p> <p>SSN Num <input type="text" value="Enter SSN Num"/> ?</p> <p>VPC Peer Keepalive <input type="text" value="Enter IP Address"/> ?</p> <p>VPC Domain <input type="text" value="Enter VPC Domain"/> ?</p> <p>VPC Peer Port Info <input type="text" value="Enter VPC Port"/> ?</p> <p>VPC Peer VLAN Info <input type="text" value="Enter VPC VLAN Info"/> ?</p> <p>BR Management Port Info <input type="text" value="Enter BR Port Info"/> ?</p> <p>BR Management PO Info <input type="text" value="Enter BR PO Info"/> ?</p> <p>Save Cancel</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Hostname</td> <td>ToR switch hostname.</td> </tr> <tr> <td>Username</td> <td>ToR switch username.</td> </tr> <tr> <td>Password</td> <td>ToR switch password.</td> </tr> <tr> <td>SSH IP</td> <td>ToR switch SSH IP Address.</td> </tr> <tr> <td>SSN Num</td> <td>ToR switch ssn num.</td> </tr> <tr> <td>VPC Peer Keepalive</td> <td>Peer Management IP. You do not define if there is no peer.</td> </tr> <tr> <td>VPC Domain</td> <td>Do not define if peer is absent.</td> </tr> </tbody> </table>	Name	Description	Hostname	ToR switch hostname.	Username	ToR switch username.	Password	ToR switch password.	SSH IP	ToR switch SSH IP Address.	SSN Num	ToR switch ssn num.	VPC Peer Keepalive	Peer Management IP. You do not define if there is no peer.	VPC Domain	Do not define if peer is absent.
Name	Description																
Hostname	ToR switch hostname.																
Username	ToR switch username.																
Password	ToR switch password.																
SSH IP	ToR switch SSH IP Address.																
SSN Num	ToR switch ssn num.																
VPC Peer Keepalive	Peer Management IP. You do not define if there is no peer.																
VPC Domain	Do not define if peer is absent.																

Name	Description	
	VPC Peer Port Info	Interface for vpc peer ports.
	BR Management Port Info	Management interface of the management node.
	BR Management PO Info	Port channel number for management interface of the management node.
ClickSave.		
On clicking save button, Add ToR Info Connected to Fabric field is visible.	Port Channel field.	Enter the Port Channel input.
	Switch Name field.	Enter the name of the Switch.

- g. Click **NFVI Monitoring** checkbox in Blueprint Initial Setup to enable the NFVI Monitoring configuration tab.

The screenshot shows the 'Create Blueprint configuration' screen in the Cisco VIM Unified Management interface. The 'Physical Setup' tab is active. The 'Master' section contains an 'Admin IP' field. The 'Collector' section contains 'Management VIP' and 'Management IP' fields. The 'NFVIMON Admin' section contains an 'Admin Name' field. Below these sections are two tables for 'Collector VM Info'. The first table, for 'Master 2', has one row with 'Host Name', 'Password', 'Cleartext Password', 'Admin IP', 'Management IP', and 'Action' columns. The second table, for 'Collector 2', also has one row with the same columns. Both tables show 'No data available'.

Name	Description
Master - Admin IP	IP Address of Control Center VM
Collector - Management VIP	VIP for ceilometer/dispatcher to use, must be unique across VIM Pod
Master 2	Optional, but becomes mandatory if collector 2 is defined. Must contain valid admin IP.
Collector 2	Collector 2 is secondary set of collector. All the properties must be present as collector. Optional, but becomes mandatory if Master 2 is defined. Contains management VIP and collector VM information.
NFVIMON ADMIN	Optional and reconfigurable to add/update user id. Once enabled, you must have only one admin.

Name	Description
Host Name	Hostname of Collector VM
Password	Password of Collector VM
CCUSER Password	Password of CCUSER
Admin IP	SSH IP of Collector VM
Management IP	Management IP of Collector VM
Collector ToR Connections	<p>1. Click on (+) icon to Add Collector ToR Connections.</p> <p>2. Select the ToR switches from list to add the information.</p> <p>3. It is optional and available for ToR type NCS-5500</p> <p>4. For now, it supports adding only one Collector ToR Connection</p> 
Rabbit MQ User Name	Enter Rabbit MQ username.

- h. Click **CVIMMON** option in Blueprint Initial Setup to enable the CVIMMON configuration tab.

Create Blueprint configuration

Physical Setup

Blueprint Initial Setup **Physical Setup** OpenStack Setup

Registry Setup UCSM Common Networking Servers and Roles CVIMMON

Enable

UI Access: True

Polling Intervals:

Low Frequency	1	m	
Medium Frequency	30	s	
High Frequency	15	s	

Save Form Offline Validation Clear

CVIM-MON is a built-in infrastructure monitoring service based on telegraf/prometheus/grafana.

When enabled, the telegraf service will be deployed on every node on the pod to capture infrastructure level stats (CPU, memory, network, containers, and so on) and a Prometheus server will be installed on the management node to poll for these stats and store them in its time series database. The statistics can then be viewed using the grafana server that is accessible on the management node at port 3000 (password protected).

There are three levels of polling intervals which are used by different telegraf plugins:

- Low frequency interval is used to collect system level metrics like cpu, memory
- Medium frequency interval is used to collect docker metrics
- High frequency interval is used to collect rabbitmq metrics

Defining polling intervals in setup data is optional, if not defined the default values will be used

PODNAME is required when CVIM-MON is enabled

Name	Description
Enable	Default is False
UI-Access	Indicates either True or False. If this option is set in setupdata with a value, the same value is shown as selected in the drop-down list. If this option is not set in the setupdata, the default value of True is selected in the drop-down list.
Polling Intervals	
Low frequency - deprecated	<Integer following with time sign (s/m/h)> # min of 1 minute (1m) if not defined defaults to 1m, also it needs to be higher than medium interval.

Name	Description
Medium frequency - deprecated	<Integer following with time sign (s/m/h)> # min of 30 seconds (30s) if not defined defaults to 30s, also it needs to be higher than high interval.
High frequency	<Integer following with time sign (s/m/h)> # min of 10 seconds (10s) if not defined defaults to 10s.

While CVIM-MON checkbox is checked in Blueprint Initial setup, there is a checkbox provided in the CVIM-MON tab area for enabling the SNMP feature. When user check this enable SNMP checkbox, Add a Manager button appears in the right area.

Clicking on this button shows various fields related to that manager. User can add up to three SNMP managers.

Name	Description
Address	IPv4 or IPv6 address of the remote SNMP manager, unique across all managers
Port	Port (1-65535) to sent the traps; default 162, unique across all managers
Version	SNMP version of the manager; default 'v2c'
Community	For SNMPv2c. Community name; default 'public'
Engine_Id	For SNMPv3. ContextEngineId, min length of 5, max length of 32, unique across all managers; cannot we all 00s or FFs
Users	List of users; maximum: 3
Name	Username has to be unique across users for the same manager
auth_key	Need to be min of 8 chars
authentication	Authentication protocol; default: 'SHA'
privacy_key	Encryption password; by default uses the same as the authentication
encryption	Encryption protocol ; default: 'AES128'

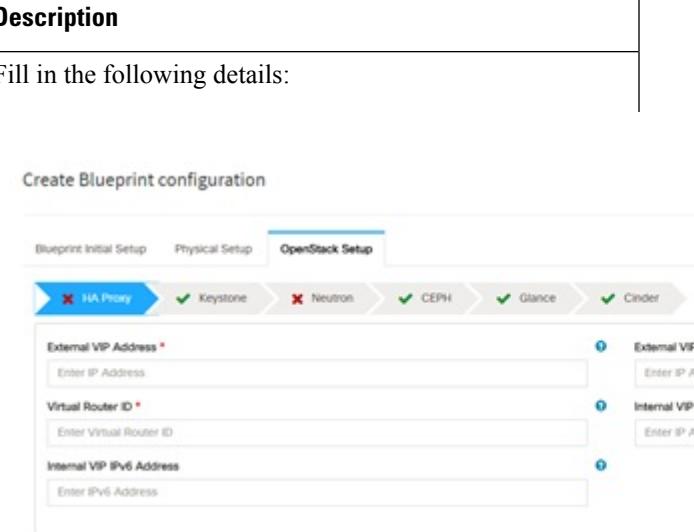
If CVIM-MON is enabled and Platform type is C, then an optional feature to get SNMP traps from Cisco CIMC is available in the CVIM-MON tab area. With this new feature SERVER_MON, there is a checkbox to enable or disable this feature and an input field to provide host information. You can either add comma separated server information or can have ALL to include all the servers.

Table 24:

Name	Description
Enable	True/False

Name	Description
Host information	ALL or list of servers.
Remote syslog severity	Optional. Indicates if cimc is programmed to send rsyslog events with this minimum severity. Possible syslog severity values are: <'emergency' 'alert' 'critical' 'error' 'warning' 'notice' 'informational' 'debug'>. These are optional and values can be changed.

- i. Click **OpenStack Setup** tab to advance to the OpenStack Setup Configuration page. On the **OpenStack Setup** page of the Cisco VIM Insight wizard, complete the following fields:

Name	Description
HA Proxy	<p>Fill in the following details:</p>  <p>External VIP Address field Enter the IP address of the External VIP.</p> <p>External VIP Address IPv6 field Enter the IPv6 address of the External VIP.</p> <p>Virtual Router ID field Enter the Router ID for the HA.</p> <p>Internal VIP Address IPv6 field Enter the IPv6 address of the Internal IP.</p> <p>Internal VIP Address field Enter the IP address of the Internal VIP.</p>

Name	Description										
Keystone	<p>The following are the Pre-populated field values. This option is always set to be true.</p> <p>Create Blueprint configuration</p> <p>Blueprint Initial Setup Physical Setup OpenStack Setup</p> <p>HA Proxy ✗ Keystone ✓ Neutron ✗ CEPH ✓ Glance ✓ Cinder</p> <table border="1"> <tr> <td>Admin Username *</td> <td>admin</td> </tr> <tr> <td>Admin Tenant Name *</td> <td>admin</td> </tr> <tr> <td>Enter Virtual Router ID</td> <td></td> </tr> <tr> <td>Internal VIP IPv6 Address</td> <td>Enter IPv6 Address</td> </tr> <tr> <td>Enter IP Address</td> <td></td> </tr> </table>	Admin Username *	admin	Admin Tenant Name *	admin	Enter Virtual Router ID		Internal VIP IPv6 Address	Enter IPv6 Address	Enter IP Address	
Admin Username *	admin										
Admin Tenant Name *	admin										
Enter Virtual Router ID											
Internal VIP IPv6 Address	Enter IPv6 Address										
Enter IP Address											

Name	Description
LDAP (Only if Keystonev3 is enabled) Note This option is only available with Keystone v3	

Name	Description																				
	<p>This is available only when LDAP is enabled under <i>Optional Features and Services</i> in Blueprint Initial Setup.</p> <table border="1"> <tr> <td>Domain Name field</td> <td>Enter the Domain name.</td> </tr> <tr> <td>Object Class for Users field</td> <td>Enter a string as input.</td> </tr> <tr> <td>Object Class for Groups field</td> <td>Enter a string.</td> </tr> <tr> <td>Domain Name Tree for Users field</td> <td>Enter a string.</td> </tr> <tr> <td>Domain Name Tree for Groups field</td> <td>Enter a string.</td> </tr> <tr> <td>Suffix for Domain Name field</td> <td>Enter a string.</td> </tr> <tr> <td>URL field</td> <td>Enter a URL with ending port number.</td> </tr> <tr> <td>Domain Name of bind user field</td> <td>Enter a string.</td> </tr> <tr> <td>Password field</td> <td>Enter Password as string format.</td> </tr> <tr> <td>User Filter field</td> <td>Enter filter name as string.</td> </tr> </table>	Domain Name field	Enter the Domain name.	Object Class for Users field	Enter a string as input.	Object Class for Groups field	Enter a string.	Domain Name Tree for Users field	Enter a string.	Domain Name Tree for Groups field	Enter a string.	Suffix for Domain Name field	Enter a string.	URL field	Enter a URL with ending port number.	Domain Name of bind user field	Enter a string.	Password field	Enter Password as string format.	User Filter field	Enter filter name as string.
Domain Name field	Enter the Domain name.																				
Object Class for Users field	Enter a string as input.																				
Object Class for Groups field	Enter a string.																				
Domain Name Tree for Users field	Enter a string.																				
Domain Name Tree for Groups field	Enter a string.																				
Suffix for Domain Name field	Enter a string.																				
URL field	Enter a URL with ending port number.																				
Domain Name of bind user field	Enter a string.																				
Password field	Enter Password as string format.																				
User Filter field	Enter filter name as string.																				

Name	Description
	User ID Attribute field Enter a string.
	User Name Attribute field Enter a string.
	User Mail Attribute field Enter a string.
	Group Name Attribute field Enter a string.
	Group_filter field It is optional. Enter a string.
	Group Member Attribute field It is optional. Enter a string.
	Group Id Attribute field It is optional. Enter a string.
	Group Members Are Ids field It is optional. Enter True or False

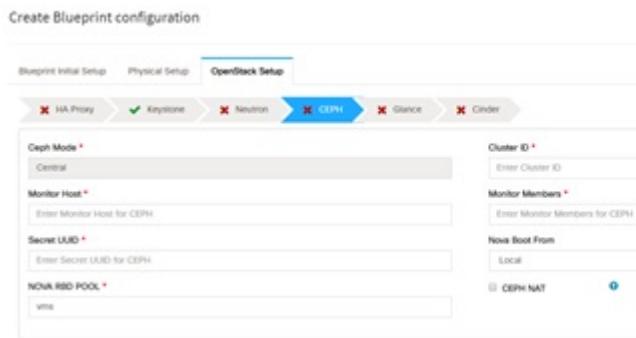
Name	Description
Neutron	

Name	Description
	<p>Neutron fields change on the basis of <i>Tenant Network Type</i> selection from Blueprint Initial Setup. Following are the options available for Neutron for OVS/VLAN:</p> <p>Create Blueprint configuration</p> 
Tenant Network Type field	It is Auto-filled based on the <i>Tenant Network Type</i> selected in the Blueprint Initial Setup page.
Mechanism Drivers field	It is Auto-filled based on the <i>Tenant Network Type</i> selected in Blueprint Initial Setup page.
NFV Hosts field	<p>It is auto-filled with the compute you added in Server and Roles.</p> <p>If you select All in this section NFV_HOSTS: ALL is added to the Blueprint or you can select one particular compute. For example:</p> <p>NFV_HOSTS: compute-server-1, compute-server-2.</p>
ENABLE_CAT	Optional to enable Intel CAT. It is valid only when NFV Host is enabled. By default, it is set to false.
RESERVED_I3CACHE_LINES_SOCKET	Allowed value of reserved cache lines per socket is between 1 and 32. It is valid only when ENABLE_CAT is set to True.

Name	Description
	Tenant VLAN Ranges field List of ranges separated by comma form start:end.
	Provider VLAN Ranges field List of ranges separated by comma form start:end.
	VM Huge Page Size (available for NFV_HOSTS option) field 2M or 1G
	Enable Jumbo Frames field Enable the checkbox.
	Enable VM Emulator Pin <ul style="list-style-type: none"> • Optional, when NFV_HOSTS is enabled. • When a VM is spawned with this parameter enabled, NOVA allocates additional vCPU on top of the vCPU count specified in the flavor, and pin vCPU0 to the pCPU that is reserved in the pool.
	VM Emulator PCORES Per Socket <ul style="list-style-type: none"> • Optional, if <code>ENABLE_VM_EMULATOR_PIN</code> is enabled. • Enter the number of cores per socket. • Defaults to 1. Can be in the range of 1 to 4.
	Base MAC Address

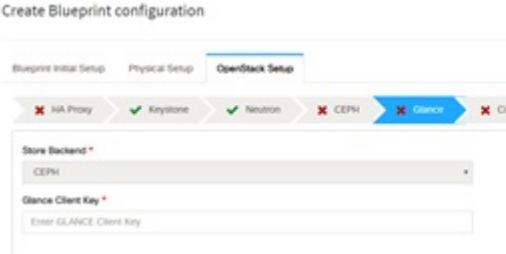
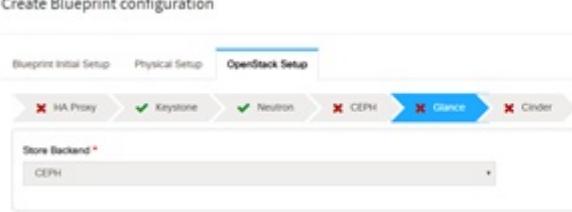
Name	Description
	<p>Option for virtual machine MAC addresses. You can configure DHCP reservations for them so that they always get the same IP address regardless of the host hypervisor or operating system that is running.</p> <p>If the MAC address ends with 00:00,</p> <ul style="list-style-type: none"> • First entry of the first octet must be a Hex • Second entry of the first octet must be 2, 6, a or e <p>For example, [a-f][2,6,a,e]:yz:uv:ws:00:00</p>
Nova Opt for low latency	<p>Optional. You can enable additional real time optimizations in OpenStack NOVA.</p> <p>By default, it is set to False</p>
	<p>For Tenant Network Type, Linux Bridge everything remains the same but Tenant VLAN Ranges is removed.</p>

Name	Description
CEPH	

Name	Description																
	<p>1. 1. When Object Storage Backend is selected as <i>Central</i> in the blueprint initial setup.</p>  <table border="1" data-bbox="913 739 1534 1320"> <tr> <td>Ceph Mode *</td> <td>By default Ceph Mode is Central.</td> </tr> <tr> <td>Cluster ID</td> <td>Enter the Cluster ID.</td> </tr> <tr> <td>Monitor Host</td> <td>Enter the Monitor Host for CEPH</td> </tr> <tr> <td>Monitor Members</td> <td>Enter the Monitor Members for CEPH</td> </tr> <tr> <td>Secret UUID</td> <td>Enter the Secret UUID for CEPH</td> </tr> <tr> <td>NOVA Boot from</td> <td>You can choose CEPH or local from the drop-down list.</td> </tr> <tr> <td>NOVA RBD POOL</td> <td>Enter the NOVA RBD Pool (default's to vms)</td> </tr> <tr> <td>CEPH NAT</td> <td>CEPH NAT is required for Central Ceph and when mgmt network is not routable.</td> </tr> </table> <p>2. When Object Storage Backend is selected as <i>Dedicated</i> in the blueprint initial setup for dedicated Ceph.</p>	Ceph Mode *	By default Ceph Mode is Central.	Cluster ID	Enter the Cluster ID.	Monitor Host	Enter the Monitor Host for CEPH	Monitor Members	Enter the Monitor Members for CEPH	Secret UUID	Enter the Secret UUID for CEPH	NOVA Boot from	You can choose CEPH or local from the drop-down list.	NOVA RBD POOL	Enter the NOVA RBD Pool (default's to vms)	CEPH NAT	CEPH NAT is required for Central Ceph and when mgmt network is not routable.
Ceph Mode *	By default Ceph Mode is Central.																
Cluster ID	Enter the Cluster ID.																
Monitor Host	Enter the Monitor Host for CEPH																
Monitor Members	Enter the Monitor Members for CEPH																
Secret UUID	Enter the Secret UUID for CEPH																
NOVA Boot from	You can choose CEPH or local from the drop-down list.																
NOVA RBD POOL	Enter the NOVA RBD Pool (default's to vms)																
CEPH NAT	CEPH NAT is required for Central Ceph and when mgmt network is not routable.																

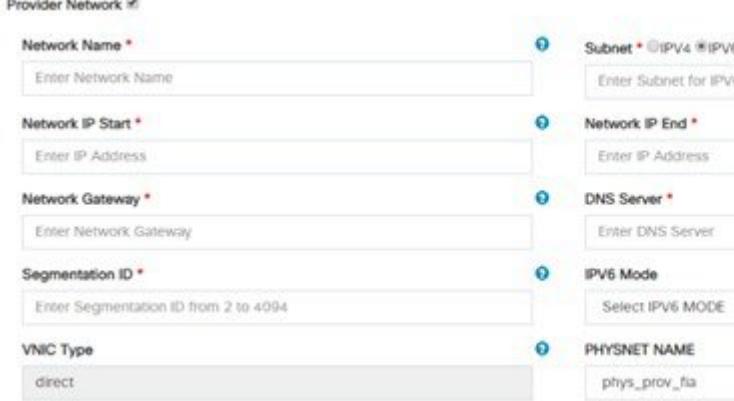
Name	Description
	<p>The screenshot shows the Cisco VIM Unified Management interface for configuring OpenStack. The 'OpenStack Setup' tab is active. The configuration screen includes the following fields:</p> <ul style="list-style-type: none"> Ceph Mode *: Dedicated (selected) Nova Boot From: Ceph Cinder Percentage: 40 Glance Percentage: 30 Nova Percentage: 30 Ceph OSD Cores *: 10 <p>Below the configuration fields, there is descriptive text and requirements:</p> <ul style="list-style-type: none"> • Ceph Mode: By default Dedicated. • NOVA Boot From: Can be <i>Ceph</i> or <i>local</i>. • Cinder Percentage: Available when Nova Boot From is <i>local</i> or <i>Ceph</i>. • Glance Percentage: Available when Nova Boot From is <i>local</i> or <i>Ceph</i>. • Nova Percentage: Available when Nova Boot From is <i>Ceph</i>. <p>If NOVA Boot From is <i>local</i>, the total of Cinder Percentage and Glance Percentage must be 100.</p> <p>If NOVA Boot From is <i>Ceph</i>, the total of Cinder Percentage and Glance Percentage must be 100.</p> <p>CEPH OSD RESERVED PCORES : Default value is 2. Minimum value is 2 and Maximum value is 12 (only for Micropod and hyper-converged pods).</p>

Name	Description
	<p>3. When Object Storage Backend is selected as <i>NetApp</i> in the blueprint initial setup, the</p>  <ul style="list-style-type: none"> Ceph Mode: NetApp is selected by default. Cinder Percentage: Enter Cinder percentage for Ceph. Glance Percentage: Enter glance percentage for Ceph <p>Total of Cinder Percentage and Glance Percentage must be 100.</p>

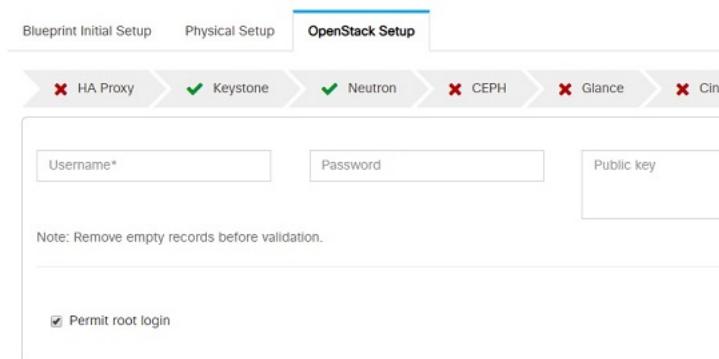
Name	Description						
GLANCE	<p>1. When Object Storage Backend is selected as <i>Central</i> in the blueprint initial setup.</p>  <table border="1" data-bbox="856 665 1491 834"> <tr> <td>Store Backend</td> <td>By default CEPH.</td> </tr> <tr> <td>Glance RBD Pool</td> <td>By default images.</td> </tr> <tr> <td>Glance Client Key</td> <td>Enter GLANCE Client Key</td> </tr> </table> <p>2. When Object Storage Backend is selected as <i>Dedicated</i> in the blueprint initial setup.</p>  <p>By default Populated for CEPH Dedicated with Store Backend value as CEPH.</p>	Store Backend	By default CEPH.	Glance RBD Pool	By default images.	Glance Client Key	Enter GLANCE Client Key
Store Backend	By default CEPH.						
Glance RBD Pool	By default images.						
Glance Client Key	Enter GLANCE Client Key						

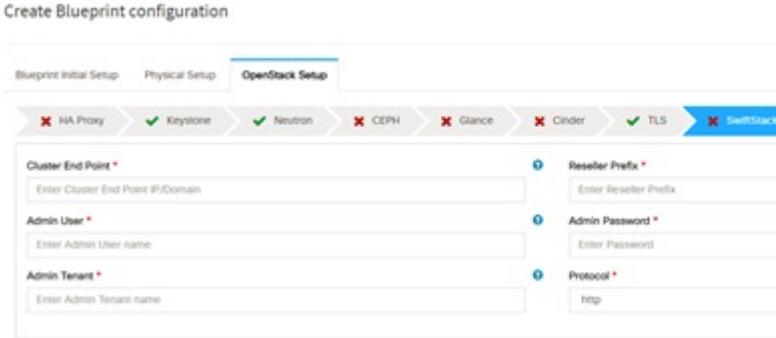
Name	Description						
CINDER	<p>By default Populated for <i>CEPH Dedicated</i> with Volume Driver value as CEPH.</p> <p>Create Blueprint configuration</p> <table border="1"> <tr> <td>Volume Driver</td> <td>By default CEPH.</td> </tr> <tr> <td>Cinder RBD Pool field</td> <td>By default volumes.</td> </tr> <tr> <td>Cinder Client Key</td> <td>Enter Cinder Client Key</td> </tr> </table>	Volume Driver	By default CEPH.	Cinder RBD Pool field	By default volumes.	Cinder Client Key	Enter Cinder Client Key
Volume Driver	By default CEPH.						
Cinder RBD Pool field	By default volumes.						
Cinder Client Key	Enter Cinder Client Key						
<p>Create Blueprint configuration</p>							

Name	Description
VMTP VMTP optional section will only be visible once VMTP is selected from Blueprint Initial Setup.	

Name	Description																				
	<p>Check one of the check boxes to specify a VMTP network:</p> <ul style="list-style-type: none"> • Provider Network • External Network <p>For the Provider Network complete the following:</p>  <table border="1" data-bbox="889 939 1537 1852"> <tr> <td>Network Name field</td><td>Enter the name of the provider network.</td></tr> <tr> <td>IPv4 Or IPv6 field</td><td>Select either IPv4 or IPv6</td></tr> <tr> <td>Subnet field</td><td>Enter the Subnet for Provider Network.</td></tr> <tr> <td>Network IP Start field</td><td>Enter the start of the floating IPv4/IPv6 address.</td></tr> <tr> <td>Network IP End field</td><td>Enter the end of the floating IPv4/IPv6 address.</td></tr> <tr> <td>Network Gateway field</td><td>Enter the IPv4/IPv6 address for the Gateway.</td></tr> <tr> <td>DNS Server field</td><td>Enter the DNS server IPv4/IPV6 address.</td></tr> <tr> <td>Segmentation ID field</td><td>Enter the segmentation ID.</td></tr> <tr> <td>IPv6 Mode field</td><td>Enter the IPv6 address along with the prefix, if IPv6 option is selected.</td></tr> <tr> <td>VNIC Type</td><td>For B-series, Direct is default value. For C –series, it is either ‘Default’ or ‘Normal’</td></tr> </table>	Network Name field	Enter the name of the provider network.	IPv4 Or IPv6 field	Select either IPv4 or IPv6	Subnet field	Enter the Subnet for Provider Network.	Network IP Start field	Enter the start of the floating IPv4/IPv6 address.	Network IP End field	Enter the end of the floating IPv4/IPv6 address.	Network Gateway field	Enter the IPv4/IPv6 address for the Gateway.	DNS Server field	Enter the DNS server IPv4/IPV6 address.	Segmentation ID field	Enter the segmentation ID.	IPv6 Mode field	Enter the IPv6 address along with the prefix, if IPv6 option is selected.	VNIC Type	For B-series, Direct is default value. For C –series, it is either ‘Default’ or ‘Normal’
Network Name field	Enter the name of the provider network.																				
IPv4 Or IPv6 field	Select either IPv4 or IPv6																				
Subnet field	Enter the Subnet for Provider Network.																				
Network IP Start field	Enter the start of the floating IPv4/IPv6 address.																				
Network IP End field	Enter the end of the floating IPv4/IPv6 address.																				
Network Gateway field	Enter the IPv4/IPv6 address for the Gateway.																				
DNS Server field	Enter the DNS server IPv4/IPV6 address.																				
Segmentation ID field	Enter the segmentation ID.																				
IPv6 Mode field	Enter the IPv6 address along with the prefix, if IPv6 option is selected.																				
VNIC Type	For B-series, Direct is default value. For C –series, it is either ‘Default’ or ‘Normal’																				

Name	Description																
	<p>PHYSNET NAME</p> <p>For B-series, the value is phys_prov_fia or phys_prov_fib. For C-series, value like phys_sriov_n is found, where n is number of ports.</p> <p>For External Network fill in the following details:</p>  <table border="1"> <tr> <td>Network Name *</td> <td>Enter Network Name</td> <td>Subnet *</td> <td>Enter Subnet</td> </tr> <tr> <td>Network IP Start *</td> <td>Enter IP Address</td> <td>Network IP End *</td> <td>Enter IP Address</td> </tr> <tr> <td>Network Gateway *</td> <td>Enter Network Gateway</td> <td>DNS Server *</td> <td>Enter DNS Server</td> </tr> <tr> <td>VNIC Type</td> <td>direct</td> <td></td> <td></td> </tr> </table>	Network Name *	Enter Network Name	Subnet *	Enter Subnet	Network IP Start *	Enter IP Address	Network IP End *	Enter IP Address	Network Gateway *	Enter Network Gateway	DNS Server *	Enter DNS Server	VNIC Type	direct		
Network Name *	Enter Network Name	Subnet *	Enter Subnet														
Network IP Start *	Enter IP Address	Network IP End *	Enter IP Address														
Network Gateway *	Enter Network Gateway	DNS Server *	Enter DNS Server														
VNIC Type	direct																
	<p>Network Name field Enter the name for the external network.</p> <p>Subnet field Enter the Subnet for the external Network.</p> <p>Network IP Start field Enter the start of the floating IPv4 address.</p> <p>Network IP End field Enter the endof the floating IPv4 address.</p> <p>Network Gateway field Enter the IPv4 address for the Gateway.</p> <p>DNS Server field Enter the DNS server IPv4 address.</p>																
TLS This optional section will only be visible once TLS is selected from Blueprint Initial Setup Page.	<p>TLS has two options:</p> <ul style="list-style-type: none"> • External LB VIP FQDN - -Text field. • External LB VIP TLS True/False. By default this option is false. 																

Name	Description						
Under the OpenStack setup tab, Vim_admins tab will be visible only when Vim_admins is selected from the Optional Features & Services under the Blueprint Initial setup tab	<p>Following are the field descriptions for VIM Admins:</p> <ul style="list-style-type: none"> • Add Username, Password, Public key or both for the non-root login. • At least one Vim Admin must be configured when Permit root login is false. <p>Create Blueprint configuration</p>  <table border="1" data-bbox="881 1056 1534 1393"> <tr> <td>User Name</td> <td>Enter username for Vim Admin.</td> </tr> <tr> <td>Password</td> <td>Password field. Admin hash password should always start with \$6.</td> </tr> <tr> <td>Public Key</td> <td>Public key for vim admin should always start with 'ssh-rsa AAAA....'</td> </tr> </table>	User Name	Enter username for Vim Admin.	Password	Password field. Admin hash password should always start with \$6.	Public Key	Public key for vim admin should always start with 'ssh-rsa AAAA....'
User Name	Enter username for Vim Admin.						
Password	Password field. Admin hash password should always start with \$6.						
Public Key	Public key for vim admin should always start with 'ssh-rsa AAAA....'						

Name	Description												
SwiftStack optional section will be visible once SwiftStack is selected from Blueprint Initial Setup Page. SwiftStack is only supported with Keystonev2 . If you select Keystonev3, swiftstack will not be available for configuration.	<p>Following are the options that needs to be filled for SwiftStack:</p>  <table border="1"> <tr> <td>Cluster End Point field</td> <td>IP address of PAC (Proxy-Account-Container) endpoint.</td> </tr> <tr> <td>Admin User field</td> <td>Admin user for swift to authenticate in keystone.</td> </tr> <tr> <td>Admin Tenant field</td> <td>The service tenant corresponding to the Account-Container used by the Swiftstack.</td> </tr> <tr> <td>Reseller Prefix field</td> <td>Reseller_prefix as configured for Keysone Auth,AuthToken support in Swiftstack. Example: KEY_</td> </tr> <tr> <td>Admin Password field</td> <td>swiftstack_admin_password</td> </tr> <tr> <td>Protocol</td> <td>http or https</td> </tr> </table>	Cluster End Point field	IP address of PAC (Proxy-Account-Container) endpoint.	Admin User field	Admin user for swift to authenticate in keystone.	Admin Tenant field	The service tenant corresponding to the Account-Container used by the Swiftstack.	Reseller Prefix field	Reseller_prefix as configured for Keysone Auth,AuthToken support in Swiftstack. Example: KEY_	Admin Password field	swiftstack_admin_password	Protocol	http or https
Cluster End Point field	IP address of PAC (Proxy-Account-Container) endpoint.												
Admin User field	Admin user for swift to authenticate in keystone.												
Admin Tenant field	The service tenant corresponding to the Account-Container used by the Swiftstack.												
Reseller Prefix field	Reseller_prefix as configured for Keysone Auth,AuthToken support in Swiftstack. Example: KEY_												
Admin Password field	swiftstack_admin_password												
Protocol	http or https												

- j. For SolidFire, enter the following:

Name	Description
------	-------------

SolidFire is visible for configuration on day0
 SolidFire is not allowed as a day-2 deployment option
 SolidFire is always available with CEPH.

Cluster MVIP	Management IP of SolidFire cluster.
Cluster SVIP field	Storage VIP of SolidFire cluster.
Admin Username	Admin user on SolidFire cluster
Admin Password	Admin password on SolidFire cluster.

- k. For NetApp, enter the following:

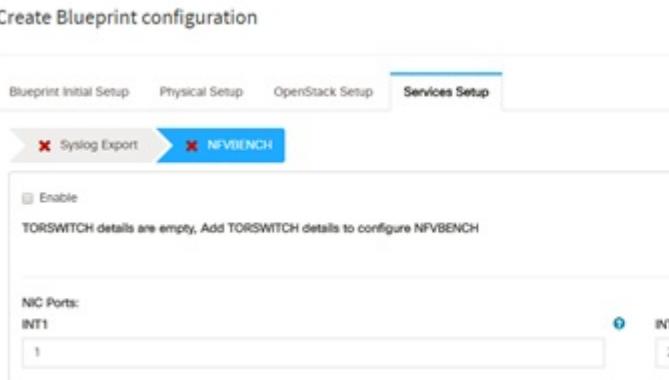
Name	Description
NETAPP	Optional NETAPP configuration. No dedicated Ceph allowed.

Name	Description
	<ul style="list-style-type: none"> Server Hostname: It is the IPv4/IPv6/Hostname/FQDN of NetApp management/API server. Server Port: It is the port of NetApp management/API server. 80 for HTTP 443 for HTTPS. Transport Type: It is HTTP or HTTPS. Server port depends on Transport type. Username : It is the username of Netapp API Server. Password: It is the password of NetApp API Server. Cinder NFS Server: It is the data path IP of NFS Server. Provide the IPv4/IPv6/Hostname/FQDN Cinder NFS Path: It is the path of NFS Server. Nova NFS Server: It is the data path IP of NOVA NFS server. Provide the IPv4/IPv6/Hostname/FQDN. Nova NFS Path: It is the path of NOVA NFS. V Server: SVM for Cinder NFS volume. Provide the IPv4/IPv6/Hostname/FQDN. Glance NFS Server : It is the data path of glance NFS server. Provide the IPv4/IPv6/Hostname/FQDN Glance NFS Path: It is the path of glance NFS server.

- I. If **Syslog Export** or **NFVBENCH** is selected in **Blueprint Initial Setup**, the **Services Setup** pane is enabled for the user to view.

Following are the options under **Services Setup** tab:

Name	Description												
Syslog Export	<p>Following are the options for Syslog Settings:</p> <p>Create Blueprint configuration</p> <table border="1"> <tbody> <tr> <td>Remote Host</td> <td>Enter Syslog IP address.</td> </tr> <tr> <td>Protocol</td> <td>Only UDP is supported.</td> </tr> <tr> <td>Facility</td> <td>Defaults to local5.</td> </tr> <tr> <td>Severity</td> <td>Defaults to debug.</td> </tr> <tr> <td>Clients</td> <td>Defaults to ELK.</td> </tr> <tr> <td>Port</td> <td>Defaults to 514 but can be modified by the User.</td> </tr> </tbody> </table>	Remote Host	Enter Syslog IP address.	Protocol	Only UDP is supported.	Facility	Defaults to local5.	Severity	Defaults to debug.	Clients	Defaults to ELK.	Port	Defaults to 514 but can be modified by the User.
Remote Host	Enter Syslog IP address.												
Protocol	Only UDP is supported.												
Facility	Defaults to local5.												
Severity	Defaults to debug.												
Clients	Defaults to ELK.												
Port	Defaults to 514 but can be modified by the User.												

Name	Description
NFVBENCH	<p>NFVBENCH enable checkbox which by default is <i>False</i>.</p>  <p>Create Blueprint configuration</p> <p>Blueprint Initial Setup Physical Setup OpenStack Setup Services Setup</p> <p><input checked="" type="checkbox"/> Syslog Export <input checked="" type="checkbox"/> NFVBENCH</p> <p><input type="checkbox"/> Enable TORSWITCH details are empty. Add TORSWITCH details to configure NFVBENCH</p> <p>NIC Ports: INT1 <input type="text" value="1"/> INT2 <input type="text" value="2"/></p>
ENABLE_ESC_PRIV	Enable the checkbox to set it as True. By default it is <i>False</i> .

Name	Description						
Horizon Aliases	<p>If the external_lb_vip is behind a NAT router or has a DNS alias, provide a list of those addresses.</p> <p>Horizon Allowed Hosts uses comma separated list of IP addresses and/or DNS names for horizon hosting.</p> <p>Create Blueprint configuration</p> <table border="1"> <thead> <tr> <th colspan="2">Horizon Allowed Hosts *</th> </tr> <tr> <th>NAT IP</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>No Data Available</td> <td></td> </tr> </tbody> </table>	Horizon Allowed Hosts *		NAT IP	Action	No Data Available	
Horizon Allowed Hosts *							
NAT IP	Action						
No Data Available							

Name	Description
Vim LDAP Admins.	

Name	Description
	<p>Optional entry to support LDAP for Admin access to management node. TLS must be enabled for the external api (i.e. external_lb_vip_tls: True).</p> <p>Following are the values to be filled to add vim LDAP admins:</p>  <ul style="list-style-type: none"> domain_name: It is a mandatory field. Indicates the domain name to define vim LDAP admins. ldap_uri: It is a mandatory field. The ldap_uris must be secured over ldaps.

Name	Description
	<ul style="list-style-type: none"> • ldap_search_base: It is mandatory. Enter search base. • ldap_schema: Optional. Enter the schema. • ldap_user_object_class: Optional. Indicates the posixAccount. • ldap_user_uid_number: Optional. Enter the user id number. • ldap_user_gid_number: Optional. Enter the group id number. • ldap_group_member: Optional. Enter the group member ID. • ldap_default_bind_dn: Optional . Enter the default DN • ldap_default_authtok: Optional. Enter the default Auth token • ldap_default_authtok_type :Optional. Enter the default Auth token type • ldap_group_search_base: Optional. Enter the group search base • ldap_user_search_base:Optional. Enter the user search base • access_provider: Optional • simple_allow_groups: Optional • ldap_id_use_start_tls: Optional. Can be true or false • ldap_tls_reqcert: Optional, can be “never”/”allow”/”try”/”demand” • chpass_provider:Optional can be ‘ldap’ or ‘krb5’ or ‘ad’ or ‘none’

Step 3 To create a C Series Blueprint:

- On the **Blueprint Initial Setup** page of the Cisco VIM Insight, complete the following fields:

The screenshot shows the 'Create Blueprint configuration' page in the Cisco VIM Unified Management interface. The left sidebar has 'Blueprint Setup' selected under 'Pre-Install'. The main form has tabs for 'Blueprint Initial Setup', 'Physical Setup', and 'OpenStack Setup'. The 'Blueprint Initial Setup' tab is active. It contains fields for 'Blueprint Name' (with placeholder 'Enter Blueprint Name'), 'Tenant Network' (set to 'Linux Bridge/VXLAN'), 'Object Storage Backend' (set to 'Central'), 'Platform Type' (set to 'C-series'), and 'POD Type' (set to 'Fusion'). Below these are sections for 'Optional Features & Services' with checkboxes for various options like Synclog Export Settings, ES_REMOTE_BACKUP, and NFVI Monitoring. At the bottom is an 'Import Existing YAML file' section with 'Browse' and 'Load' buttons.

Name	Description
Blueprint Name field.	Enter the name for the blueprint configuration.
Platform Type drop-down list	Choose one of the following platform types: <ul style="list-style-type: none"> • B-Series (By default) • C-Series (Select C Series)
Tenant Network drop-down list	Choose one of the following tenant network types: <ul style="list-style-type: none"> • Linux Bridge/VXLAN • OVS/VLAN • VTS/VLAN • VPP/VLAN • ACI/VLAN <p>Note when VTS/VLAN or ACI/VLAN is selected then respective tabs are available on Blueprint setup. When Mechanism driver OVS or ACI is selected, VM_HUGEPAGE_PERCENTAGE field is enabled for all standalone compute nodes, when NFV_HOSTS is enabled.</p>

Name	Description
Pod Type drop-down list	<p>Choose one of the following pod type :</p> <ul style="list-style-type: none"> • Fullon(By Default) • Micro • UMHC • NGENAHC <p>Note</p> <ul style="list-style-type: none"> • UMHC pod type is only supported for OVS/VLAN tenant type. • NGENAHC is supported for VPP/VLAN tenant type with no SRIOV • Pod type micro is supported for OVS/VLAN, ACI/VLAN,VPP/VLAN.
Ceph Mode drop-down list	Choose one of the following Ceph types: <ul style="list-style-type: none"> • Dedicated (By Default) • Central. Central is not supported in Production
Optional and Services Features checkbox	<p>Swiftstack, LDAP, Syslog Export Settings, Install Mode, TorSwitch Information, TLS, NFVMON, Pod Name, VMTP, NFVbench, Autbackup, Heat, Keystone v3, Enable Esc Priv.</p> <p>If any one is selected, the corresponding section is visible in various Blueprint sections.</p> <p>By default all features are disabled except Auto Backup.</p>
Import Existing YAML file	<p>If you have an existing C Series YAML file you can use this feature to upload the file.</p> <p>Insight will automatically fill in the fields and any missed mandatory field will be highlighted in the respective section.</p>

- b. Click **Physical Setup** to advance to the **Registry Setup** configuration page. Fill in the following details for Registry Setup:

The screenshot shows the 'Create Blueprint configuration' page in the Cisco VIM Unified Management interface. The 'Physical Setup' tab is selected. The 'Registry Setup' step is active, displaying fields for 'Registry User Name', 'Registry Password', and 'Registry Email'. A progress bar indicates the current step is 'Registry Setup'.

Name	Description
Registry User Name text field	User-Name for Registry (Mandatory).
Registry Password text field	Password for Registry (Mandatory).
Registry Email text field	Email ID for Registry (Mandatory).

Once all the mandatory fields are filled the **Validation Check Registry Page** will be changed to a Green Tick.

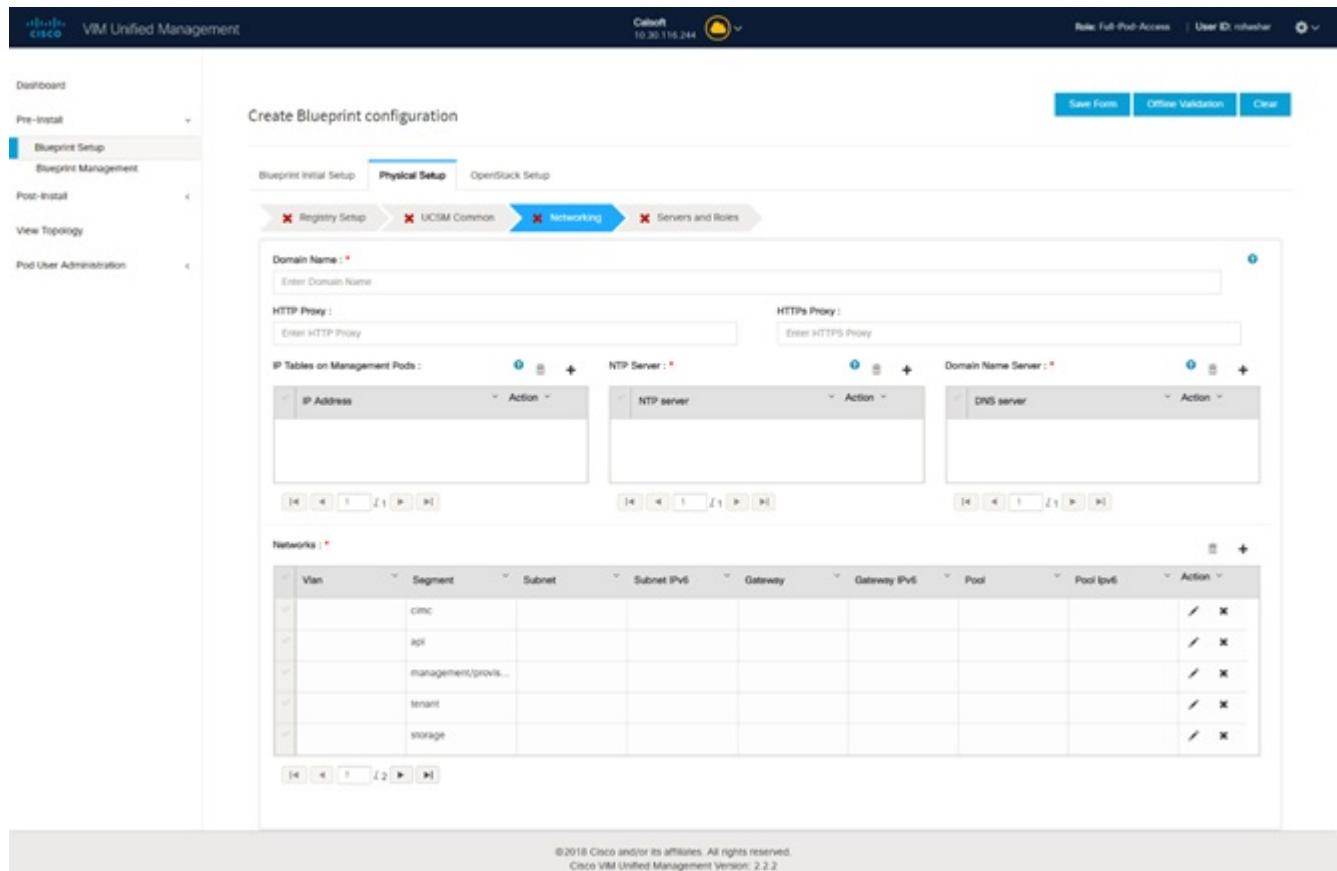
- c. Click **CIMC Common Tab** and complete the following fields:

The screenshot shows the 'Create Blueprint configuration' page in the Cisco VIM Unified Management interface. The 'Physical Setup' tab is selected. The 'CIMC Common' step is active, displaying fields for 'Username' (admin) and 'Password' (password). A progress bar indicates the current step is 'CIMC Common'.

Name	Description
User Name disabled field	By default value is Admin.
Password text field	Enter Password for UCSM Common (Mandatory).

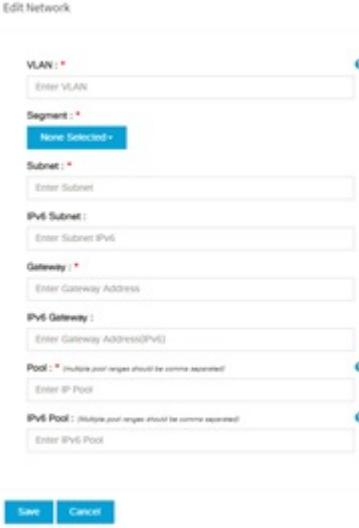
- d. Click **Networking** to advance to the networking section of the Blueprint.

Configuring OpenStack Installation



Name	Description
Domain Name field	Enter the domain name. (Mandatory)
HTTP Proxy Server field	If your configuration uses an HTTP proxy server, enter the IP address of the server.
HTTPS Proxy Server field	If your configuration uses an HTTPS proxy server, enter the IP address of the server.
IP Tables on Management Pods	Specifies the list of IP Address with Mask.
NTP Servers field	Enter a maximum of four and minimum of one IPv4 and/or IPv6 addresses in the table.
Domain Name Servers field	Enter a maximum of three and minimum of one IPv4 and/or IPV6 addresses.

Name	Description
Networks table	

Name	Description						
	<p>Network table is pre-populated with Segments. To add Networks you can either clear all the table with Delete all or click edit icon for each segment and fill in the details.</p> <p>You can add, edit, or delete network information in the table.</p>  <ul style="list-style-type: none"> Click Add (+) to add new entries (networks) to the table. Specify the following fields in the Edit Entry to Networks dialog: <table border="1"> <thead> <tr> <th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>VLAN field</td><td>Enter the VLAN ID. For Segment - Provider, the VLAN ID value is 'none'.</td></tr> <tr> <td>Segment drop-down list</td><td>When you add/edit new segment then following segments types are available in the form of dropdown list and you can select only one. <ul style="list-style-type: none"> API Management/provision Tenant Storage </td></tr> </tbody> </table>	Name	Description	VLAN field	Enter the VLAN ID . For Segment - Provider, the VLAN ID value is 'none'.	Segment drop-down list	When you add/edit new segment then following segments types are available in the form of dropdown list and you can select only one. <ul style="list-style-type: none"> API Management/provision Tenant Storage
Name	Description						
VLAN field	Enter the VLAN ID . For Segment - Provider, the VLAN ID value is 'none'.						
Segment drop-down list	When you add/edit new segment then following segments types are available in the form of dropdown list and you can select only one. <ul style="list-style-type: none"> API Management/provision Tenant Storage 						

Name	Description
	<ul style="list-style-type: none"> • External • Provider • ACIINFRA <p>Note Aciinfra segment is available only when ACI/VLAN tenant type is selected) Depending upon the segment some of the entries below are not needed. Please refer to the example file in openstack-configs dir for details.</p>
Subnet field	Enter the IPv4 address for the subnet.
IPv6 Subnet field	Enter IPv6 address. This field will be available only for Management provision and API
Gateway field	Enter the IPv4 address for the Gateway.
Gateway IPv6 field	Enter the IPv6 address for the gateway. This will support for API and management provision.
Pool field	Enter the pool information in the required format, for example: 10.1.15-10.1.1.10,102.1.5-102.1.10 This field is available only for the Mgmt/Provision, Storage, and Tenant segments.
IPv6 Pool field	Enter the pool information in the required format. For example: 10.1.15-10.1.1.10,102.1.5-102.1.10

Configuring OpenStack Installation

Name	Description
	Click Save .

- e. On the **Servers and Roles** page of the Cisco VIM Suite wizard, a pre-populated table filled with Roles : Control, Compute and Block Storage (Only if CEPH Dedicated is selected in Blueprint Initial Setup is available.

The screenshot shows the 'Create Blueprint configuration' screen in the Cisco VIM Unified Management interface. The 'Physical Setup' tab is active. In the 'COBBLER' section, the 'Control Kickstart' field is set to 'ucb-r-100-c-series.ks'. Below it, the 'Compute Kickstart' field is also set to 'ucb-r-100-c-series.ks'. In the 'Server and Roles' table, there are four entries:

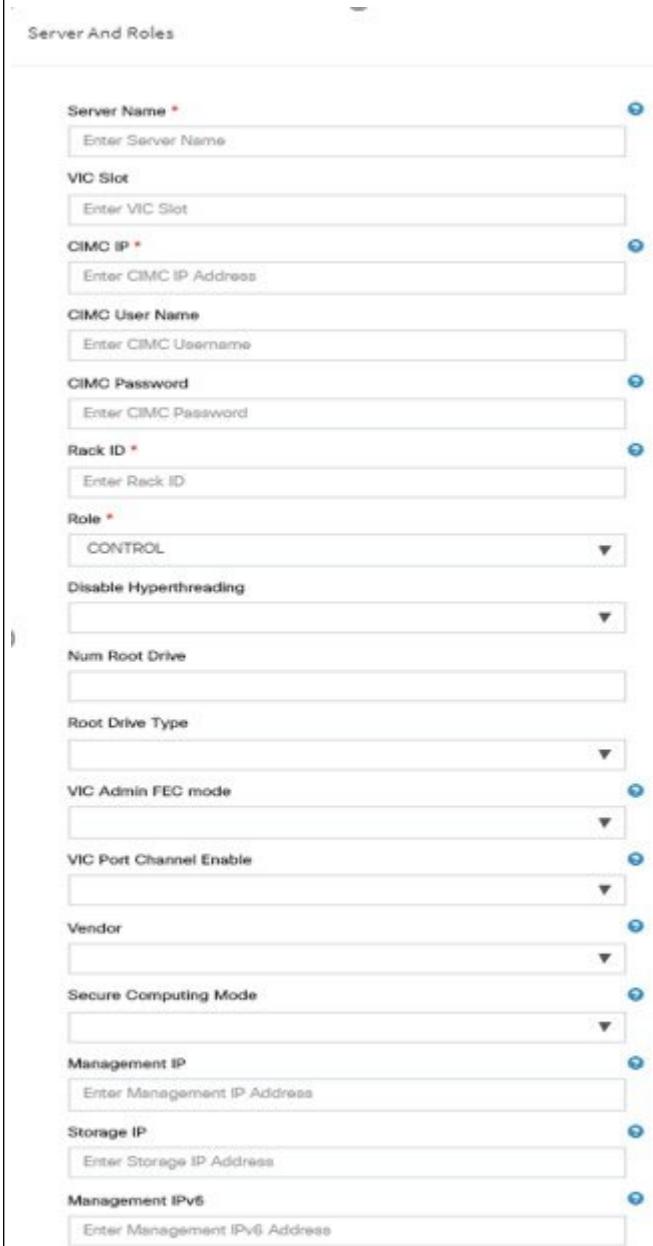
Server Name	CMC IP	CMC User name	CMC Password	Rack ID	Role	Management IP	Management IPv6	Action
control					control			X
compute01					compute			X
compute02					compute			X

Note If you choose mechanism driver as OVS or ACI, VM_HUGE PAGE_PERCENTAGE field column is available for compute nodes, where you can fill values from 0 to 100%, when NFV_HOSTS: ALL is chosen. Also, option of NIC Level Redundancy appears only when Intel Nic Support is set to true. This is applicable only in the case of M5 based pods.

Name	Description
Server User Name field	Enter the username of the server.
Disable Hyperthreading	Default value is false. You can set it as true or false.

Name	Description	
Cobbler	Enter the Cobbler details in the following fields:	
	Name	Description
	Cobbler Timeout field	The default value is 45 min. This is an optional parameter. Timeout is displayed in minutes, and its value ranges from 30 to 120.
	Block Storage Kickstart field	Kickstart file for Storage Node.
	Admin Password Hash field	Enter the Admin Password. Password should be Alphanumeric. Password should contain minimum 8 characters and maximum of 32 characters.
	Cobbler Username field	Enter the cobbler username to access the cobbler server.
	Control Kickstart field	Kickstart file for Control Node.
	Compute Kickstart field	Kickstart file for Compute Node.
	Cobbler Admin Username field	Enter the admin username of the Cobbler.

Name	Description
<p>Add Entry to Servers and Roles</p> <p>Note when Pod type micro is selected then all the three servers will be associated with control, compute and block storage role.</p> <p>For Example:</p> <p>Roles</p> <ul style="list-style-type: none"> • Block Storage <ul style="list-style-type: none"> • -Server 1 • -Server 2 • -Server 3 • Control <ul style="list-style-type: none"> • -Server 1 • -Server 2 • -Server 3 • Compute <ul style="list-style-type: none"> • -Server 1 • -Server 2 • -Server 3 <p>Note When Pod type UMHC is selected then auto ToR configuration is not supported and the ToR info at server and roles level is not allowed to be entered.</p>	

Name	Description		
	<p>Click Edit or + to add a new server and role to the table.</p> <p>If mechanism driver is either OVS or ACI, an additional optional field VM_HUGE PAGE_PERCENTAGE is shown when compute role is chosen; This option is only valid when NFV_HOSTS is set to ALL; If no value is entered then the global value of VM_HUGE PAGE_PERCENTAGE is used.</p>  <table border="1" data-bbox="881 1795 1534 1858"> <tr> <td>Server Name</td> <td>Entry the name of the server.</td> </tr> </table>	Server Name	Entry the name of the server.
Server Name	Entry the name of the server.		

Name	Description
	Rack ID field The rack ID for the server.
	VIC Slot field Enter a VIC Slot.
	CIMC IP field Enter a IP address.
	CIMC Username field Enter a Username.
	CIMC Password field Enter a Password for CIMC.
	Select the Role from the drop down list Choose Control or Compute or Block Storage from the drop-down list. If Podtype is fullon and selected role type is Block storage, an additional field Osd_disk_type is displayed where you can choose either HDD or SSD.
	VIC Admin FEC mode Applicable only for Cisco VIC that supports to change the admin FEC mode. Can be auto/off/cl74/cl91.
	VIC Port Channel Enable Optional. Default is true. Can be either true or false.
	Secure Computing mode Optional. By default, it is set to 1, if not defined. Can be either 0 or 1.
	Management IP It is an optional field but if provided for one Server then it is mandatory to provide it for other Servers as well.
	Storage IP Optional, but if provided for one server then it is mandatory to provide details for other servers.
	Vendor Allow static override value for platform vendor instead of dynamic discovery at runtime. Can be CISCO - Cisco Systems Inc/ QCT - Quanta Cloud Technology Inc/ HPE - Hewlett Packard Enterprise.
	Management IPv6

Name	Description	
		Routable and valid IPv6 address. It is an optional field but if provided for one server then it is mandatory for all other servers as well.
	BGP speaker addressees	Optional, only when NETWORK_OPTIONS is vxlan network, for controller node only, IP belongs to the vxlan-tenant network but not part of the pool.
	INTEL_SRIOV_VFS	Value range is 1 to 32. Can be defined globally and overridden at per compute level via add/remove or fresh installationI, if Intel N3000 card is installed for pod type edge.
	INTEL_FPGA_VFS	Value range is 1 to 8. Can be defined globally and overridden at per compute level via add/remove or fresh installationI, if Intel N3000 card is installed for pod type edge.
	INTEL_VC_SRIOV_VFS	Value range is 1 to 32. Can be defined globally and overridden at per compute level via add/remove or fresh installationI, if Intel N3000 card is installed for pod type edge.
Click Save or Add .	On clicking Save or Add all information related to Servers and Roles gets saved.	
If Configure ToR checkbox is True with at-least one switch detail, these fields will be displayed for each server and this is similar to DP Tor: Port Channel and Switch Name (Mandatory if Configure ToR is true)	<ul style="list-style-type: none"> • Port Channel field • Switch Name field • Switch Port Info field 	<ul style="list-style-type: none"> • Enter the port channel input. • Enter the switch name. • Enter the switch port information.

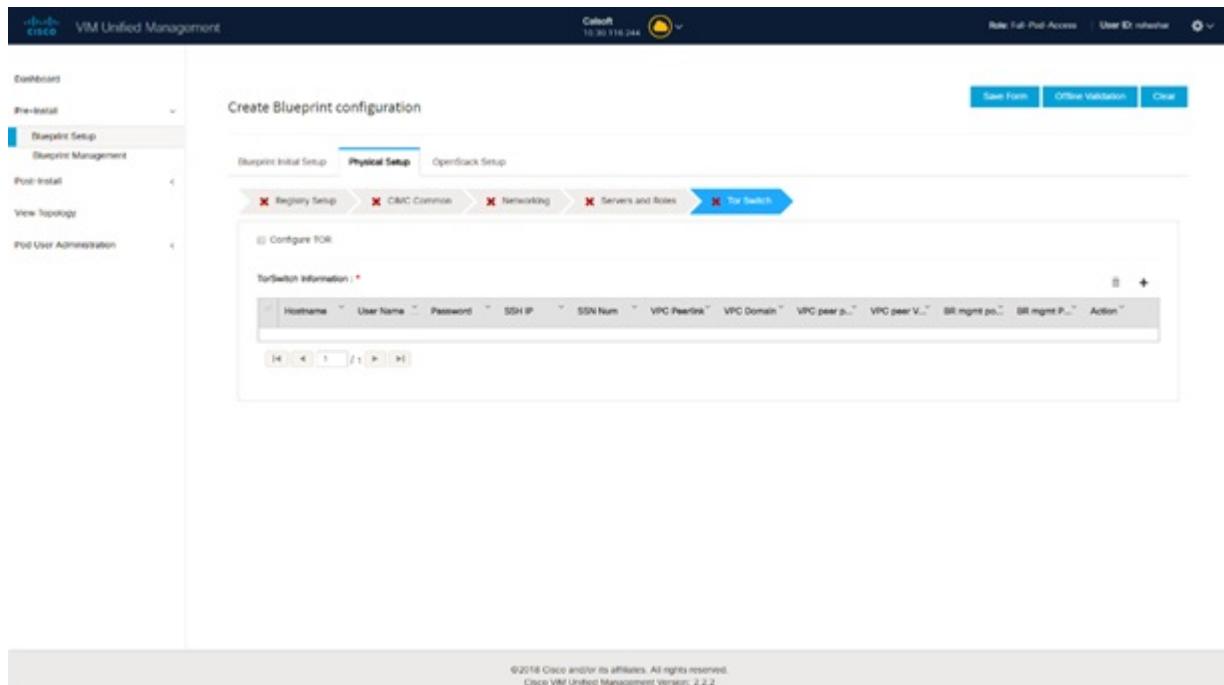
Name	Description	
DP ToR (Only for Control and Compute) : Mandatory if Intel NIC and Configure TOR is True.	<ul style="list-style-type: none"> • Port Channel field • Switch Name field • Switch Port Info field 	<ul style="list-style-type: none"> • Enter the port channel input. • Enter the switch name. • Enter the switch port information.
SRIOV TOR INFO (Only for Compute Nodes). It is mandatory in server and roles if Intel NIC and Configure TOR is True. with TOR TYPE Nexus. For TOR TYPE NCS-5500 these fields are optional Switch Name (Mandatory if Configure ToR is true). This field appears only when Intel NIC support is true, as Auto TOR config is not supported in VIC_NIC combo	<ul style="list-style-type: none"> • Switch Name field • Switch Port Info field 	<ul style="list-style-type: none"> • Enter the switch name. • Enter the switch port information.
Intel SRIOV VFS (valid for Intel NIC testbeds) and can be integer.	For SRIOV support for Intel NIC. By Default, SRIOV support is disabled. To enable, define a value in the range # * 1-32 when INTEL_NIC_SUPPORT is set True (X710 Max VFs = 32) # * 1-63 when CISCO_VIC_INTEL_SRIOV is set True (X520 Max VFs = 63)	
INTEL_SRIOV_PHYS_PORTS (valid for Intel NIC test beds) and can be of value 2 or 4 (default is 2)	In some cases the # of Physical SRIOV port needed is 4; to meet that requirement, define the following: # this is optional, if nothing is defined code will assume it to be 2; the only 2 integer values this parameter # takes is 2 or 4 and is true when INTEL_NIC_SUPPORT is True and INTEL_SRIOV_VFS is valid. For NCS-5500 this value is set to 4 and is non-editable.	
Click Save or Add .	If all mandatory fields are filled click Save or Add to add information on Servers and Roles.	
Disable Hyperthreading	Default value is false. You can set it as true or false.	
Click Save		

Note Maximum two ToR info needs to be configured for each connection type on each node (control, compute and block_storage node).

Note If pod type UMHC is selected then CISCO_VIC_INTEL_SRIOV is enabled to be TRUE. CISCO_VIC_INTEL_SRIOV is also supported on Micro pod with expanded computes

Note For Tenant type ACI/VLAN, port channel for each ToR port will not be available in servers and roles, as APIC will automatically assign port-channel numbers. Also, for ACI in full on mode you can select Intel NIC Support in the “Servers and Roles” section.

- f. Click **ToR Switch** checkbox in **Blueprint Initial Setup** to enable the **TOR SWITCH** configuration page. It is an **Optional** section in Blueprint Setup but once all the fields are filled in then it will become a part of the Blueprint.



©2018 Cisco and/or its affiliates. All rights reserved.
Cisco VIM Unified Management Version: 2.2.2

Name	Description
Configure ToR optional checkbox. Note If UMHC is selected as podtype, configure TOR is not allowed.	Enabling this checkbox, changes the configure ToR section from false to true. Note Configure tor is true then ToR switch info maps in servers

Name	Description
ToR Switch Information	mandatory table if you want to enter ToR information.

Name	Description																
	<p>Click (+) to add information for ToR Switch.</p> <p>Switch Details</p> <p>Hostname * <input type="text" value="Enter Switch Hostname"/></p> <p>Username * <input type="text" value="Enter Switch Username"/></p> <p>Password * <input type="text" value="Enter Password"/></p> <p>SSH-IP * <input type="text" value="Enter IP Address"/></p> <p>SSN Num <input type="text" value="Enter SSN Num"/></p> <p>VPC Peer Keepalive <input type="text" value="Enter IP Address"/></p> <p>VPC Domain <input type="text" value="Enter VPC Domain"/></p> <p>VPC Peer Port Info <input type="text" value="Enter VPC Port"/></p> <p>VPC Peer VLAN Info <input type="text" value="Enter VPC VLAN Info"/></p> <p>BR Management Port Info <input type="text" value="Enter BR Port Info"/></p> <p>BR Management PO Info <input type="text" value="Enter BR PO Info"/></p> <hr/> <p>Save Cancel</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>ToR switch name.</td> </tr> <tr> <td>Username</td> <td>ToR switch username.</td> </tr> <tr> <td>Password</td> <td>ToR switch password.</td> </tr> <tr> <td>SSH IP</td> <td>ToR switch SSH IP.</td> </tr> <tr> <td>SSN Num</td> <td>ToR switch ssn num.</td> </tr> <tr> <td>VPC Peer Keepalive</td> <td>Peer Management IP. You cannot define if there is no peer.</td> </tr> <tr> <td>VPC Domain</td> <td>Cannot define if there is no</td> </tr> </tbody> </table>	Name	Description	Name	ToR switch name.	Username	ToR switch username.	Password	ToR switch password.	SSH IP	ToR switch SSH IP.	SSN Num	ToR switch ssn num.	VPC Peer Keepalive	Peer Management IP. You cannot define if there is no peer.	VPC Domain	Cannot define if there is no
Name	Description																
Name	ToR switch name.																
Username	ToR switch username.																
Password	ToR switch password.																
SSH IP	ToR switch SSH IP.																
SSN Num	ToR switch ssn num.																
VPC Peer Keepalive	Peer Management IP. You cannot define if there is no peer.																
VPC Domain	Cannot define if there is no																

Name	Description
	peer.
VPC Peer Port Info	Interface for vpc peer ports.
VPC Peer VLAN Info	VLAN ids for vpc peer ports (optional).
BR Management Port Info	Management interface of build node.
BR Management PO Info	Port channel number for management interface of build node.
BR Management VLAN info	VLAN ID for management interface of build node (access).
Splitter Optic 4x10	For C Series platform type, Tenant Type is VPP/VLAN and Pod Type is either fullon or Micro, an additional choice will be provided to select the TOR Type. If selected TOR type is NCS-5500, then user can configure splitter cable parameters.
Click Save.	

Note When tenant type ACI/VLAN is selected, the TOR switch information table differs and is mandatory.

Name	Description
Configure ToR optional checkbox. Note If UMHC is selected as podtype, configure TOR is not allowed.	Enabling this checkbox, changes the configure ToR section from false to true. Note Configure tor is true then ToR switch info maps in servers

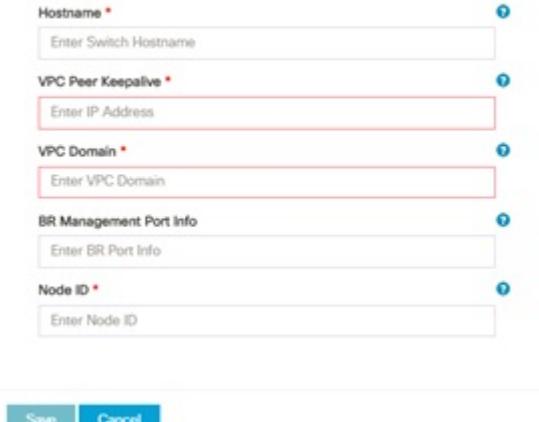
Name	Description
ToR Switch Information	mandatory table if you want to enter ToR information.

Name	Description																
	<p>Click (+) to add information for ToR Switch.</p> <p>Switch Details</p> <p>Hostname * <input type="text"/> <small>Enter Switch Hostname</small></p> <p>Username * <input type="text"/> <small>Enter Switch Username</small></p> <p>Password * <input type="text"/> <small>Enter Password</small></p> <p>SSH-IP * <input type="text"/> <small>Enter IP Address</small></p> <p>SSN Num <input type="text"/> <small>Enter SSN Num</small></p> <p>VPC Peer Keepalive <input type="text"/> <small>Enter IP Address</small></p> <p>VPC Domain <input type="text"/> <small>Enter VPC Domain</small></p> <p>VPC Peer Port Info <input type="text"/> <small>Enter VPC Port</small></p> <p>VPC Peer VLAN Info <input type="text"/> <small>Enter VPC VLAN Info</small></p> <p>BR Management Port Info <input type="text"/> <small>Enter BR Port Info</small></p> <p>BR Management PO Info <input type="text"/> <small>Enter BR PO Info</small></p> <p>Save Cancel</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>ToR switch name.</td> </tr> <tr> <td>Username</td> <td>ToR switch username.</td> </tr> <tr> <td>Password</td> <td>ToR switch password.</td> </tr> <tr> <td>SSH IP</td> <td>ToR switch SSH IP.</td> </tr> <tr> <td>SSN Num</td> <td>ToR switch ssn num.</td> </tr> <tr> <td>VPC Peer Keepalive</td> <td>Peer Management IP. You cannot define if there is no peer.</td> </tr> <tr> <td>VPC Domain</td> <td>Cannot define if there is no</td> </tr> </tbody> </table>	Name	Description	Name	ToR switch name.	Username	ToR switch username.	Password	ToR switch password.	SSH IP	ToR switch SSH IP.	SSN Num	ToR switch ssn num.	VPC Peer Keepalive	Peer Management IP. You cannot define if there is no peer.	VPC Domain	Cannot define if there is no
Name	Description																
Name	ToR switch name.																
Username	ToR switch username.																
Password	ToR switch password.																
SSH IP	ToR switch SSH IP.																
SSN Num	ToR switch ssn num.																
VPC Peer Keepalive	Peer Management IP. You cannot define if there is no peer.																
VPC Domain	Cannot define if there is no																

Name	Description	
		peer.
	VPC Peer Port Info	Interface for vpc peer ports.
	VPC Peer VLAN Info	VLAN ids for vpc peer ports (optional).
	BR Management Port Info	Management interface of build node.
	BR Management PO Info	Port channel number for management interface of build node.
	BR Management VLAN info	VLAN id for management interface of build node (access).

Click **Save**.

Note When the Tenant type ACI/VLAN is selected, the ToR switch information table differs and is mandatory.

Name	Description										
Configure ToR	<p>Is not checked, as by default ACI will configure the ToRs</p> <p>Switch Details</p>  <table border="1" data-bbox="881 908 1485 1330"> <tr> <td>Host Name</td> <td>ToR switch name.</td> </tr> <tr> <td>VPC Peer keep alive</td> <td>Enter Peer must be exist pair.</td> </tr> <tr> <td>VPC Domain</td> <td>Enter an integer.</td> </tr> <tr> <td>BR management port info</td> <td>Enter BR management port info eg. Eth1/19 ,atleast one pair to be exist.</td> </tr> <tr> <td>Enter Node ID</td> <td>Entered integer must be unique.</td> </tr> </table>	Host Name	ToR switch name.	VPC Peer keep alive	Enter Peer must be exist pair.	VPC Domain	Enter an integer.	BR management port info	Enter BR management port info eg. Eth1/19 ,atleast one pair to be exist.	Enter Node ID	Entered integer must be unique.
Host Name	ToR switch name.										
VPC Peer keep alive	Enter Peer must be exist pair.										
VPC Domain	Enter an integer.										
BR management port info	Enter BR management port info eg. Eth1/19 ,atleast one pair to be exist.										
Enter Node ID	Entered integer must be unique.										

Note If TOR_TYPE is selected as NCS-5500, the TOR switch information table differs and is mandatory.

Name	Description
Configure ToR optional checkbox Note If NSC-5500 is selected as TOR_TYPE, configure TOR is set as mandatory.	Enabling this checkbox, changes the configure ToR section from false to true. Note Configure TOR is true then ToR switchinfo maps in servers.

Name	Description
If you want to enter NCS details fill in the NCS-5500 Information table.	

Name	Description												
	<p>Click (+) to add information for NCS-5500 Switch.</p> <p>Switch Details</p> <p>Hostname * <input type="text"/> Enter Switch Hostname</p> <p>Username * <input type="text"/> Enter Switch Username</p> <p>Password * <input type="text"/> Enter Password</p> <p>SSH-IP * <input type="text"/> Enter IP Address</p> <p>VPC Peer Keepalive <input type="text"/> Enter IP Address</p> <p>VPC Peer Port Info <input type="text"/> Enter VPC Port</p> <p>VPC Peer Port Address <input type="text"/> Enter VPC Port Address</p> <p>ISIS Loopback Address <input type="text"/> Enter ISIS Loopback Address</p> <p>ISIS Net Entity Title <input type="text"/> Enter ISIS net entity title</p> <p>ISIS Prefix SID <input type="text"/> Enter ISIS Prefix SID</p> <p>BR Management Port Info <input type="text"/> Enter BR Port Info</p> <p>BR Management PO Info <input type="text"/> Enter BR PO Info</p> <p>Save Cancel</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Enter the NCS-5500 hostname.</td> </tr> <tr> <td>User Name</td> <td>Enter the NCS-5500 username.</td> </tr> <tr> <td>Password</td> <td>Enter the NCS-5500 password.</td> </tr> <tr> <td>SSH IP</td> <td>Enter the NCS-5500 ssh IP Address.</td> </tr> <tr> <td>VPC Peer Link</td> <td>Peer management IP.</td> </tr> </tbody> </table>	Name	Description	Name	Enter the NCS-5500 hostname.	User Name	Enter the NCS-5500 username.	Password	Enter the NCS-5500 password.	SSH IP	Enter the NCS-5500 ssh IP Address.	VPC Peer Link	Peer management IP.
Name	Description												
Name	Enter the NCS-5500 hostname.												
User Name	Enter the NCS-5500 username.												
Password	Enter the NCS-5500 password.												
SSH IP	Enter the NCS-5500 ssh IP Address.												
VPC Peer Link	Peer management IP.												

Name	Description	
	Name	Description
	BR Management PO Info	Port channel number for management interface of build node.
	BR Management VLAN info	VLAN id for management interface of build node (access).
	VPC Peer Port Info	Interface for vpc peer ports.
	VPC Peer Port Address	Address for ISIS exchange.
	ISIS Loopback Interface address	ISIS loopack IP Address.
	ISIS net entity title	Enter a String.
	ISIS prefix SID	Integer between 16000 to 1048575.

When TOR-TYPE selected as NCS-5500 and 2 NCS-5500 are configured it is mandatory to configure MULTI_SEGMENT_ROUTING_INFO

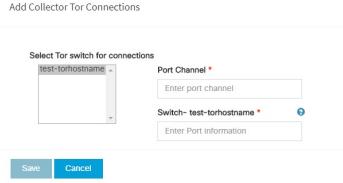
Name	Description
BGP AS Number field	Integer between 1 to 65535.
ISIS Area Tagfield	A valid string.
Loopback Interface name field	Loopback Interface name.
API bundle ID field	Integer between 1 to 65535.
API bridge domain field	String (Optional, only needed when br_api of mgmt node is also going through NCS-5500; this item and api_bundle_id are mutually exclusive).
EXT bridge domain field	A valid string (user pre-provisions physical, bundle interface, sub-interface and external BD for external uplink and provides external BD info setup_data).

- g. Click **NFVI Monitoring** checkbox in Blueprint Initial Setup to enable the NFVI Monitoring configuration tab.

Configuring OpenStack Installation

The screenshot shows the 'Create Blueprint configuration' screen in the Cisco VIM Unified Management interface. The 'Physical Setup' tab is active. The 'Master' section contains fields for 'Admin IP' and 'Admin Port'. The 'Collector' section contains fields for 'Management VIP' and 'Collector VM Info', which includes a table with columns for Host Name, Password, Cluster Password, Admin IP, Management IP, and Action. The 'Dispatcher' section contains fields for 'Rabbit MQ User Name'. The 'NFVMON Admin' section contains fields for 'Admin Name'. Below these sections are two tables for 'Master 2' and 'Collector 2', each with similar fields and a 'Clear All' button.

Name	Description
Master - Admin IP	IP Address of Control Center VM
Collector - Management VIP	VIP for ceilometer/dispatcher to use, must be unique across VIM Pod
Host Name	Hostname of Collector VM
Password	Password of Collector VM
CCUSER Password	Password of CCUSER
Admin IP	SSH IP of Collector VM
Management IP	Management IP of Collector VM

Name	Description				
Master 2	Optional, but becomes mandatory if collector 2 is defined. Must contain a valid Admin IP.				
Collector 2	Optional, but becomes mandatory if Master 2 is defined. Contains Management VIP and Collector VM information. Collector 2 is secondary set to collector, with all the properties of Collector.				
NFVIMON ADMIN	Optional and reconfigurable to add/update user id. Once enabled, you must have only one admin.				
Collector ToR Connections	<p>1. Click on (+) icon to Add Collector ToR Connections.</p> <p>2. Select the ToR switches from list to add the information.</p> <p>3. It is optional and available for ToR type NCS-5500</p> <p>4. For now, it supports adding only one Collector ToR Connection</p>  <table border="1"> <tr> <td>Port Channel</td> <td>Enter port channel.</td> </tr> <tr> <td>Switch - {torSwitch-hostname}</td> <td>Enter port number, For example, eth1/15.</td> </tr> </table> <p>Click Save</p>	Port Channel	Enter port channel.	Switch - {torSwitch-hostname}	Enter port number, For example, eth1/15.
Port Channel	Enter port channel.				
Switch - {torSwitch-hostname}	Enter port number, For example, eth1/15.				
Rabbit MQ User Name	Enter Rabbit MQ username.				

- h. Click **CVIMMON** checkbox in Blueprint Initial Setup to enable the CVIMMON configuration tab.

Configuring OpenStack Installation

The screenshot shows a software interface for configuring a blueprint. At the top, there are buttons for 'Save Form', 'Offline Validation', and 'Clear'. Below this, a navigation bar has tabs: 'Blueprint Initial Setup' (disabled), 'Physical Setup' (selected), and 'OpenStack Setup'. A progress bar at the bottom indicates steps: Registry Setup (disabled), UCSM Common (disabled), Networking (disabled), Servers and Roles (disabled), and CVIMMON (completed with a green checkmark). The main area contains sections for 'Enable' (checkbox checked) and 'Polling Intervals'. Under 'Polling Intervals', there are three rows: 'Low Frequency' (1m), 'Medium Frequency' (30s), and 'High Frequency' (10s). Each row has a small info icon to its right.

CVIM-MON is a built-in infrastructure monitoring service based on telegraf/prometheus/grafana.

When enabled, the telegraf service will be deployed on every node on the pod to capture infrastructure level stats (CPU, memory, network, containers, and so on.) and a Prometheus server will be installed on the management node to poll for these stats and store them in its time series database. The statistics can then be viewed using the grafana server that is accessible on the management node at port 3000 (password protected).

There are three levels of polling intervals which are used by different telegraf plugins:

- Low frequency interval is used to collect system level metrics like cpu, memory.
- Medium frequency interval is used to collect docker metrics.
- High frequency interval is used to collect rabbitmq metrics.

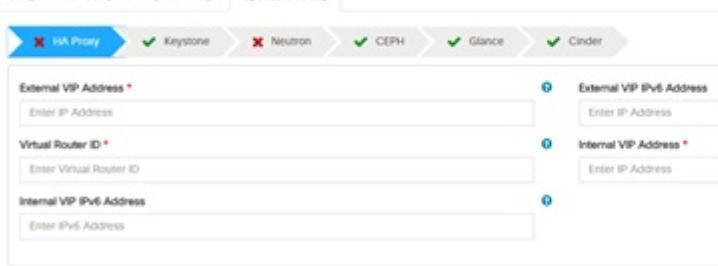
Defining polling intervals in setup data is optional. If not defined, the default values are used.

CVIM-MON is mutually exclusive to NFVIMON.

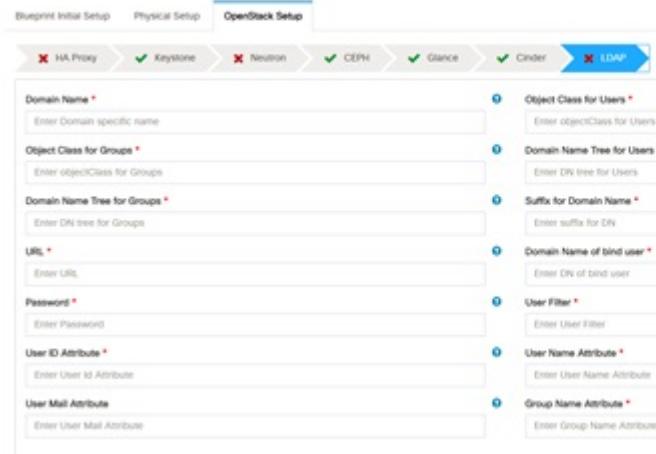
PODNAME is required, when CVIM-MON is enabled.

Name	Description
Enable	Default is False
Polling Intervals	
Low frequency – deprecated	<Integer following with time sign (s/m/h)> # min of 1 minute (1m) if not defined defaults to 1m, also it needs to be higher than medium interval.
Medium frequency – deprecated	<Integer following with time sign (s/m/h)> # min of 30 seconds (30s) if not defined defaults to 30s, also it needs to be higher than high interval.
High frequency	<Integer following with time sign (s/m/h)> # min of 10 seconds (10s) if not defined defaults to 10s.

- i. Click **OpenStack Setup** Tab to advance to the **OpenStack Setup** Configuration page. On the **OpenStack Setup** Configuration page of the Cisco VIM Insight wizard, complete the following fields:

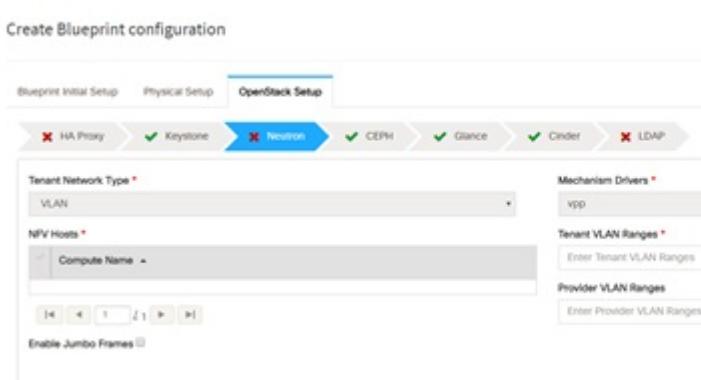
Name	Description										
HA Proxy	<p>Fill in the following details:</p> <p>Create Blueprint configuration</p>  <table border="1"> <tr> <td>External VIP Address field</td> <td>Enter IP address of External VIP.</td> </tr> <tr> <td>External VIP Address IPv6 field</td> <td>Enter IPv6 address of External VIP.</td> </tr> <tr> <td>Virtual Router ID field</td> <td>Enter the Router ID for HA.</td> </tr> <tr> <td>Internal VIP Address IPv6 field</td> <td>Enter IPv6 address of Internal IP.</td> </tr> <tr> <td>Internal VIP Address field</td> <td>Enter IP address of Internal VIP.</td> </tr> </table>	External VIP Address field	Enter IP address of External VIP.	External VIP Address IPv6 field	Enter IPv6 address of External VIP.	Virtual Router ID field	Enter the Router ID for HA.	Internal VIP Address IPv6 field	Enter IPv6 address of Internal IP.	Internal VIP Address field	Enter IP address of Internal VIP.
External VIP Address field	Enter IP address of External VIP.										
External VIP Address IPv6 field	Enter IPv6 address of External VIP.										
Virtual Router ID field	Enter the Router ID for HA.										
Internal VIP Address IPv6 field	Enter IPv6 address of Internal IP.										
Internal VIP Address field	Enter IP address of Internal VIP.										
Keystone	<p>Mandatory fields are pre-populated.</p> <p>Create Blueprint configuration</p>  <table border="1"> <tr> <td>Admin User Name</td> <td>admin.</td> </tr> <tr> <td>Admin Tenant Name</td> <td>admin.</td> </tr> </table>	Admin User Name	admin.	Admin Tenant Name	admin.						
Admin User Name	admin.										
Admin Tenant Name	admin.										

Name	Description
LDAP	

Name	Description																										
	<p>LDAP enable checkbox which by default is false, if LDAP is enabled on keystone.</p> <p>Create Blueprint configuration</p>  <table border="1" data-bbox="848 939 1517 1848"> <tr> <td>Domain Name field</td><td>Enter name for Domain name.</td></tr> <tr> <td>Object Class for Users field</td><td>Enter a string as input.</td></tr> <tr> <td>Object Class for Groups field</td><td>Enter a string.</td></tr> <tr> <td>Domain Name Tree for Users field</td><td>Enter a string.</td></tr> <tr> <td>Domain Name Tree for Groups field</td><td>Enter a string.</td></tr> <tr> <td>Suffix for Domain Name field</td><td>Enter a string.</td></tr> <tr> <td>URL field</td><td>Enter a URL with ending port number.</td></tr> <tr> <td>Domain Name of Bind User field</td><td>Enter a string.</td></tr> <tr> <td>Password field</td><td>Enter Password as string format.</td></tr> <tr> <td>User Filter field</td><td>Enter filter name as string.</td></tr> <tr> <td>User ID Attribute field</td><td>Enter a string.</td></tr> <tr> <td>User Name Attribute field</td><td>Enter a string.</td></tr> <tr> <td>User Mail Attribute field</td><td>Enter a string.</td></tr> </table>	Domain Name field	Enter name for Domain name.	Object Class for Users field	Enter a string as input.	Object Class for Groups field	Enter a string.	Domain Name Tree for Users field	Enter a string.	Domain Name Tree for Groups field	Enter a string.	Suffix for Domain Name field	Enter a string.	URL field	Enter a URL with ending port number.	Domain Name of Bind User field	Enter a string.	Password field	Enter Password as string format.	User Filter field	Enter filter name as string.	User ID Attribute field	Enter a string.	User Name Attribute field	Enter a string.	User Mail Attribute field	Enter a string.
Domain Name field	Enter name for Domain name.																										
Object Class for Users field	Enter a string as input.																										
Object Class for Groups field	Enter a string.																										
Domain Name Tree for Users field	Enter a string.																										
Domain Name Tree for Groups field	Enter a string.																										
Suffix for Domain Name field	Enter a string.																										
URL field	Enter a URL with ending port number.																										
Domain Name of Bind User field	Enter a string.																										
Password field	Enter Password as string format.																										
User Filter field	Enter filter name as string.																										
User ID Attribute field	Enter a string.																										
User Name Attribute field	Enter a string.																										
User Mail Attribute field	Enter a string.																										

Name	Description	
	Group Name Attribute field	Enter a string.

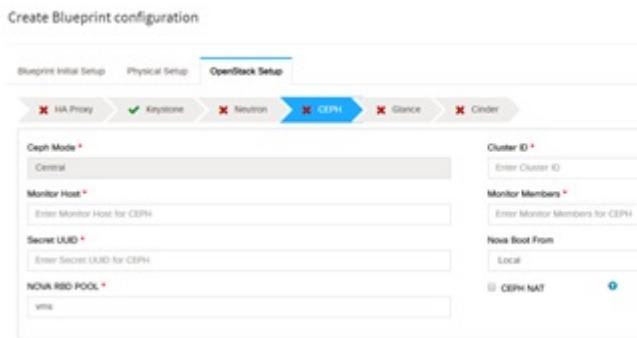
Name	Description
Neutron	

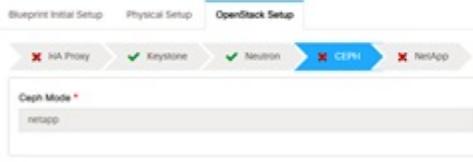
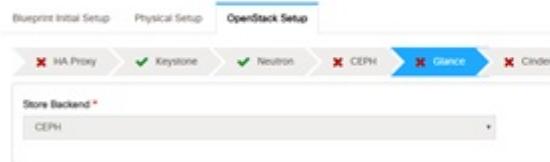
Name	Description
	<p>Neutron fields would change on the basis of Tenant Network Type Selection from Blueprint Initial Setup. Following are the options available for Neutron for OVS/VLAN:</p> 
Tenant Network Type field	Auto Filled based on the Tenant Network Type selected in the Blueprint Initial Setup page.
Mechanism Drivers field	Auto Filled based on the Tenant Network Type selected in Blueprint Initial Setup page.
NFV Hosts field	Auto-filled with the Compute you added in Server and Roles. If you select All in this section NFV_HOSTS: ALL will be added to the Blueprint or you can select one particular compute. For example: NFV_HOSTS: compute-server-1, compute-server-2.
ENABLE_CAT	Optional to enable Intel CAT. It is valid only when NFV Host is enabled. By default, it is set to false.
RESERVEDCACHELINESPERSOCKET	Allowed value of reserved cache lines per socket is between 1 and 32. It is valid only when ENABLE_CAT is set to True .
Tenant VLAN Ranges field	List of ranges separated by

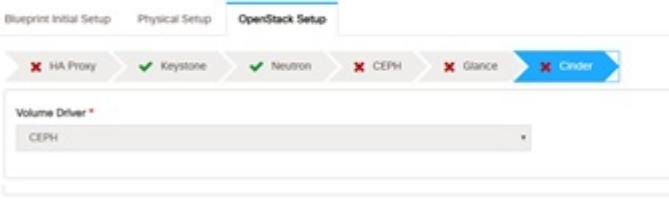
Name	Description
	comma form start:end.
Provider VLAN Ranges field	List of ranges separated by comma form start:end.
VM Huge Page Size (available for NFV_HOSTS option) field	2M or 1G (optional, defaults to 2M)
VM_HUGHPAGE_PERCENTAGE	Optional, defaults to 100%; can range between 0 and 100
VSWITCH_WORKER_PROFILE	Allowed only for VPP Available options are: <ul style="list-style-type: none">• numa_zero: The reserved cores always reside in NUMA node 0.• Even : The reserved cores are evenly distributed across all NUMA
NR_RESERVED_VSWITCH_PCORES	Allowed only for VPP Number of cores associated to VPP, defaults to 2. Takes value of 2 through 6.
Enable Jumbo Frames field	Enable the checkbox
Enable VM Emulator Pin	<ul style="list-style-type: none">• Optional, when NFV_HOSTS is enabled.• When a VM is spawned with this parameter enabled, NOVA allocates additional vCPU on top of the vCPU count specified in the flavor, and pin vCPU0 to the pCPU that is reserved in the pool.
VM Emulator PCORES Per Socket	<ul style="list-style-type: none">• Optional, when ENABLE_VM_EMULATOR_PIN is enabled.• Enter the number of cores per socket.• Defaults to 1. Values can range from 1 to 4.

Name	Description	
	Nova Opt For Low Latency	<ul style="list-style-type: none">• Optional. Used to enable additional real-time optimizations in OpenStack NOVA.• Defaults to False. <p>For Tenant Network Type Linux Bridge everything remains the same but Tenant VLAN Ranges will be removed.</p>

Name	Description
CEPH	

Name	Description																
	<p>1. 1. When Object Storage Backend is selected Central in blueprint initial setup.</p>  <table border="1" data-bbox="840 749 1485 1256"> <tr> <td>CEPH Mode</td> <td>By default Central.</td> </tr> <tr> <td>Cluster ID</td> <td>Enter Cluster ID.</td> </tr> <tr> <td>Monitor Host</td> <td>Enter Monitor Host for CEPH</td> </tr> <tr> <td>Monitor Members</td> <td>Enter Monitor Members for CEPH</td> </tr> <tr> <td>Secret UUID</td> <td>Enter Secret UUID for CEPH</td> </tr> <tr> <td>NOVA Boot from</td> <td>Drop down selection. You can choose CEPH or local.</td> </tr> <tr> <td>NOVA RBD POOL</td> <td>Enter NOVA RBD Pool (default's to vms)</td> </tr> <tr> <td>CEPH NAT</td> <td>Optional, needed for Central Ceph and when mgmt network is not routable</td> </tr> </table> <p>2. When Object Storage Backend is selected Dedicated in blueprint initial setup.</p>  <ul style="list-style-type: none"> • CEPH Mode: By default Dedicated. • NOVA Boot: From drop down selection you can choose CEPH or local. <p>3. When Object Storage Backend is selected NetApp in blueprint initial setup.</p>	CEPH Mode	By default Central.	Cluster ID	Enter Cluster ID.	Monitor Host	Enter Monitor Host for CEPH	Monitor Members	Enter Monitor Members for CEPH	Secret UUID	Enter Secret UUID for CEPH	NOVA Boot from	Drop down selection. You can choose CEPH or local.	NOVA RBD POOL	Enter NOVA RBD Pool (default's to vms)	CEPH NAT	Optional, needed for Central Ceph and when mgmt network is not routable
CEPH Mode	By default Central.																
Cluster ID	Enter Cluster ID.																
Monitor Host	Enter Monitor Host for CEPH																
Monitor Members	Enter Monitor Members for CEPH																
Secret UUID	Enter Secret UUID for CEPH																
NOVA Boot from	Drop down selection. You can choose CEPH or local.																
NOVA RBD POOL	Enter NOVA RBD Pool (default's to vms)																
CEPH NAT	Optional, needed for Central Ceph and when mgmt network is not routable																

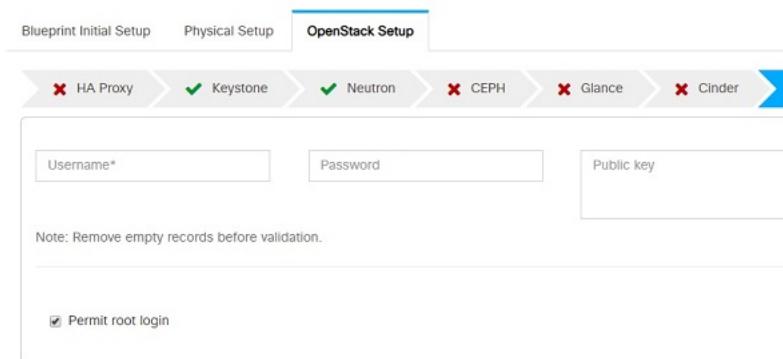
Name	Description
	<p>Create Blueprint configuration</p> 
GLANCE	<p>1. When Object Storage Backend is selected Central in blueprint initial setup.</p> <p>Create Blueprint configuration</p>  <p>When Object Storage Backend is selected Dedicated in blueprint initial setup.</p> <p>Create Blueprint configuration</p>  <p>Note By default Populated for CEPH Dedicated with Store Backend value as CEPH.</p>

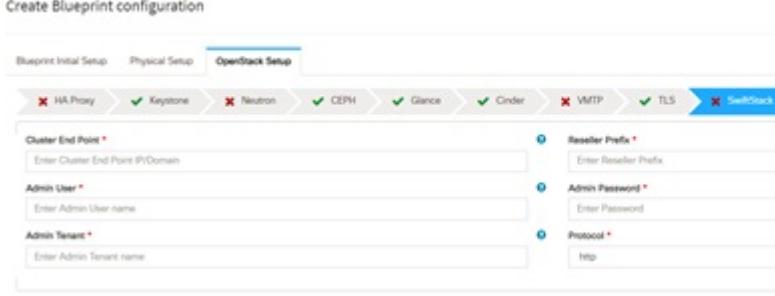
Name	Description
CINDER	<p>By default Populated for CEPH Dedicated with Volume Driver value as CEPH.</p> <p>Create Blueprint configuration</p>  <p>2. When Object Storage Backend is selected Dedicated in blueprint initial setup.</p> <p>Create Blueprint configuration</p>  <p>Note By default Populated for CEPH Dedicated with Volume Driver value as CEPH.</p>

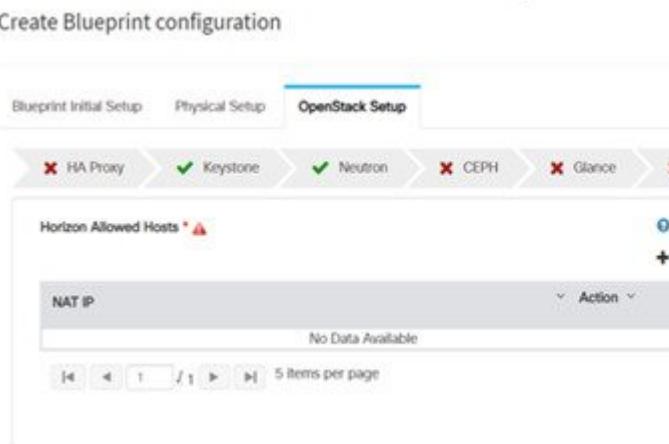
Name	Description
VMTP optional section, this will be visible only if VMTP is selected from Blueprint Initial Setup. For VTS tenant type Provider network is only supported.	

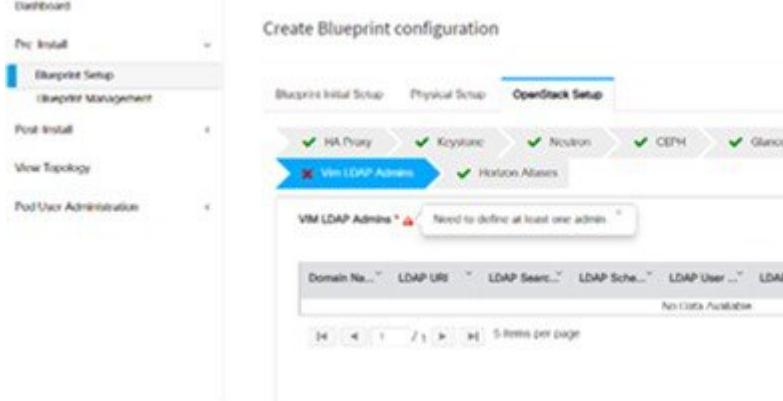
Name	Description														
	<p>Check one of the check boxes to specify a VMTP network:</p> <ul style="list-style-type: none"> Provider Network External Network <p>For the Provider Network complete the following:</p>  <table border="1"> <tr> <td>Network Name field</td><td>Enter the name for the external network.</td></tr> <tr> <td>Subnet field</td><td>Enter the Subnet for Provider Network.</td></tr> <tr> <td>Network IP Start field</td><td>Enter the starting floating IPv4 address.</td></tr> <tr> <td>Network IP End field</td><td>Enter the ending floating IPv4 address.</td></tr> <tr> <td>Network Gateway field</td><td>Enter the IPv4 address for the Gateway.</td></tr> <tr> <td>DNS Server field</td><td>Enter the DNS server IPv4 address.</td></tr> <tr> <td>Segmentation ID field</td><td>Enter the segmentation ID.</td></tr> </table> <p>For External Network fill in the following details:</p> 	Network Name field	Enter the name for the external network.	Subnet field	Enter the Subnet for Provider Network.	Network IP Start field	Enter the starting floating IPv4 address.	Network IP End field	Enter the ending floating IPv4 address.	Network Gateway field	Enter the IPv4 address for the Gateway.	DNS Server field	Enter the DNS server IPv4 address.	Segmentation ID field	Enter the segmentation ID.
Network Name field	Enter the name for the external network.														
Subnet field	Enter the Subnet for Provider Network.														
Network IP Start field	Enter the starting floating IPv4 address.														
Network IP End field	Enter the ending floating IPv4 address.														
Network Gateway field	Enter the IPv4 address for the Gateway.														
DNS Server field	Enter the DNS server IPv4 address.														
Segmentation ID field	Enter the segmentation ID.														

Name	Description
	<p>Network Name field Enter the name for the external network.</p>
	<p>IP Start field Enter the starting floating IPv4 address.</p>
	<p>IP End field Enter the ending floating IPv4 address.</p>
	<p>Gateway field Enter the IPv4 address for the Gateway.</p>
	<p>DNS Server field Enter the DNS server IPv4 address.</p>
	<p>Subnet field Enter the Subnet for External Network.</p>
TLS optional section, this will be visible only if TLS is selected from Blueprint Initial Setup Page.	<p>TLS has two options:</p> <ul style="list-style-type: none"> • External LB VIP FQDN - Text Field. • External LB VIP TLS - True/False. By default this option is false.

Name	Description						
Under the OpenStack setup tab, Vim_admins tab will be visible only when Vim_admins is selected from the Optional Features & Services under the Blueprint Initial setup tab	<p>Following are the field descriptions for VIM Admins:</p> <ul style="list-style-type: none"> • Add Username, Password, Public key or both for the non-root login. • At least one Vim Admin must be configured when Permit root login is false. <p>Create Blueprint configuration</p>  <table border="1"> <tr> <td>User Name</td> <td>Enter username for Vim Admin.</td> </tr> <tr> <td>Password</td> <td>Password field. Admin hash password should always start with \$6.</td> </tr> <tr> <td>Public Key</td> <td>Public key for vim admin should always start with 'ssh-rsa AAAA....'</td> </tr> </table>	User Name	Enter username for Vim Admin.	Password	Password field. Admin hash password should always start with \$6.	Public Key	Public key for vim admin should always start with 'ssh-rsa AAAA....'
User Name	Enter username for Vim Admin.						
Password	Password field. Admin hash password should always start with \$6.						
Public Key	Public key for vim admin should always start with 'ssh-rsa AAAA....'						

Name	Description												
SwiftStack optional section will be visible only if SwiftStack is selected from Blueprint Initial Setup Page. SwiftStack is only supported with Keystonev2 . If you select Keystonev3 , swiftstack will not be available to configure.	<p>Following are the options that needs to be filled for SwiftStack:</p>  <table border="1"> <tbody> <tr> <td>Cluster End Point</td> <td>IP address of PAC (proxy-account-container) endpoint.</td> </tr> <tr> <td>Admin User</td> <td>Admin user for swift to authenticate in keystone.</td> </tr> <tr> <td>Admin Tenant</td> <td>The service tenant corresponding to the Account-Container used by Swiftstack.</td> </tr> <tr> <td>Reseller Prefix</td> <td>Reseller_prefix as configured for Keysone Auth,AuthToken support in Swiftstack E.g KEY_</td> </tr> <tr> <td>Admin Password</td> <td>swiftstack_admin_password</td> </tr> <tr> <td>Protocol</td> <td>http or https</td> </tr> </tbody> </table>	Cluster End Point	IP address of PAC (proxy-account-container) endpoint.	Admin User	Admin user for swift to authenticate in keystone.	Admin Tenant	The service tenant corresponding to the Account-Container used by Swiftstack.	Reseller Prefix	Reseller_prefix as configured for Keysone Auth,AuthToken support in Swiftstack E.g KEY_	Admin Password	swiftstack_admin_password	Protocol	http or https
Cluster End Point	IP address of PAC (proxy-account-container) endpoint.												
Admin User	Admin user for swift to authenticate in keystone.												
Admin Tenant	The service tenant corresponding to the Account-Container used by Swiftstack.												
Reseller Prefix	Reseller_prefix as configured for Keysone Auth,AuthToken support in Swiftstack E.g KEY_												
Admin Password	swiftstack_admin_password												
Protocol	http or https												

Name	Description
Horizon Aliases	<p>If the external_lb_vip is behind a NAT router or has a DNS alias, provide a list of those addresses.</p> <p>Horizon Allowed Hosts uses comma separated list of IP addresses and/or DNS names for horizon hosting.</p> <p>Create Blueprint configuration</p> 

Name	Description
Vim LDAP Admins: Optional entry to support LDAP for admin access to management node. For this feature, TLS has to be enabled for the external api (i.e. external_lb_vip_tls: True).	<p>Following are the values to be filled to add vim LDAP admins:</p>  <ul style="list-style-type: none"> • domain_name: It is a mandatory field. Indicates the domain name to define vim LDAP admins. • ldap_uri : It is a mandatory field. The ldap_uris must be secured over ldaps. • ldap_search_base: It is mandatory. Enter search base. • ldap_schema: Optional. Enter the schema. • ldap_user_object_class: Optional. Indicates the posixAccount. • ldap_user_uid_number: Optional. Enter the user id number. • ldap_user_gid_number: Optional. Enter the group id number. • ldap_group_member: Optional. Enter the group member ID.

Name	Description																								
APICINFO tab is available in Openstack setup, when the Tenant type ACI/VLAN is selected in blueprint initial setup. Note When ACI/VLAN is selected then ToR switch from initial setup is mandatory.	<table border="1"> <thead> <tr> <th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>APIC Hosts field</td><td>Enter host input. Example: <ip1 host1>:[port] . max of 3, min of 1, not 2;</td></tr> <tr> <td>apic_username field</td><td>Enter a string format.</td></tr> <tr> <td>apic_password field</td><td>Enter Password.</td></tr> <tr> <td>apic_system_id field</td><td>Enter input as string. Max length 8.</td></tr> <tr> <td>apic_resource_prefix field</td><td>Enter string max length 6.</td></tr> <tr> <td>apic_tep_address_pool field</td><td>Allowed only 10.0.0.0/16</td></tr> <tr> <td>multiclass_address_pool field</td><td>Allowed only 225.0.0.0/15</td></tr> <tr> <td>apic_pod_id field</td><td>Enter integer(1- 65535)</td></tr> <tr> <td>apic_installer_tenant field</td><td>Enter String, max length 32</td></tr> <tr> <td>apic_installer_vrf field</td><td>Enter String, max length 32</td></tr> <tr> <td>api_l3out_network field</td><td>Enter String, max length 32</td></tr> </tbody> </table>	Name	Description	APIC Hosts field	Enter host input. Example: <ip1 host1>:[port] . max of 3, min of 1, not 2;	apic_username field	Enter a string format.	apic_password field	Enter Password.	apic_system_id field	Enter input as string. Max length 8.	apic_resource_prefix field	Enter string max length 6.	apic_tep_address_pool field	Allowed only 10.0.0.0/16	multiclass_address_pool field	Allowed only 225.0.0.0/15	apic_pod_id field	Enter integer(1- 65535)	apic_installer_tenant field	Enter String, max length 32	apic_installer_vrf field	Enter String, max length 32	api_l3out_network field	Enter String, max length 32
Name	Description																								
APIC Hosts field	Enter host input. Example: <ip1 host1>:[port] . max of 3, min of 1, not 2;																								
apic_username field	Enter a string format.																								
apic_password field	Enter Password.																								
apic_system_id field	Enter input as string. Max length 8.																								
apic_resource_prefix field	Enter string max length 6.																								
apic_tep_address_pool field	Allowed only 10.0.0.0/16																								
multiclass_address_pool field	Allowed only 225.0.0.0/15																								
apic_pod_id field	Enter integer(1- 65535)																								
apic_installer_tenant field	Enter String, max length 32																								
apic_installer_vrf field	Enter String, max length 32																								
api_l3out_network field	Enter String, max length 32																								
VTS tab is available in Openstack setup, when Tenant Type is VTS/VLAN selected. If vts day0 is enabled then SSH username and SSH password is mandatory. If SSH_username is input present then SSH password is mandatory vice-versa	<table border="1"> <thead> <tr> <th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>VTS Day0 (checkbox)</td><td>True or false default is false.</td></tr> <tr> <td>VTS User name</td><td>Enter as string does not contain special characters.</td></tr> <tr> <td>VTS Password</td><td>Enter password</td></tr> <tr> <td>VTS NCS IP</td><td>Enter IP Address format.</td></tr> <tr> <td>VTC SSH Username</td><td>Enter a string</td></tr> <tr> <td>VTC SHH Password</td><td>Enter password</td></tr> </tbody> </table>	Name	Description	VTS Day0 (checkbox)	True or false default is false.	VTS User name	Enter as string does not contain special characters.	VTS Password	Enter password	VTS NCS IP	Enter IP Address format.	VTC SSH Username	Enter a string	VTC SHH Password	Enter password										
Name	Description																								
VTS Day0 (checkbox)	True or false default is false.																								
VTS User name	Enter as string does not contain special characters.																								
VTS Password	Enter password																								
VTS NCS IP	Enter IP Address format.																								
VTC SSH Username	Enter a string																								
VTC SHH Password	Enter password																								

- j. For SolidFire, enter the following:

Name	Description

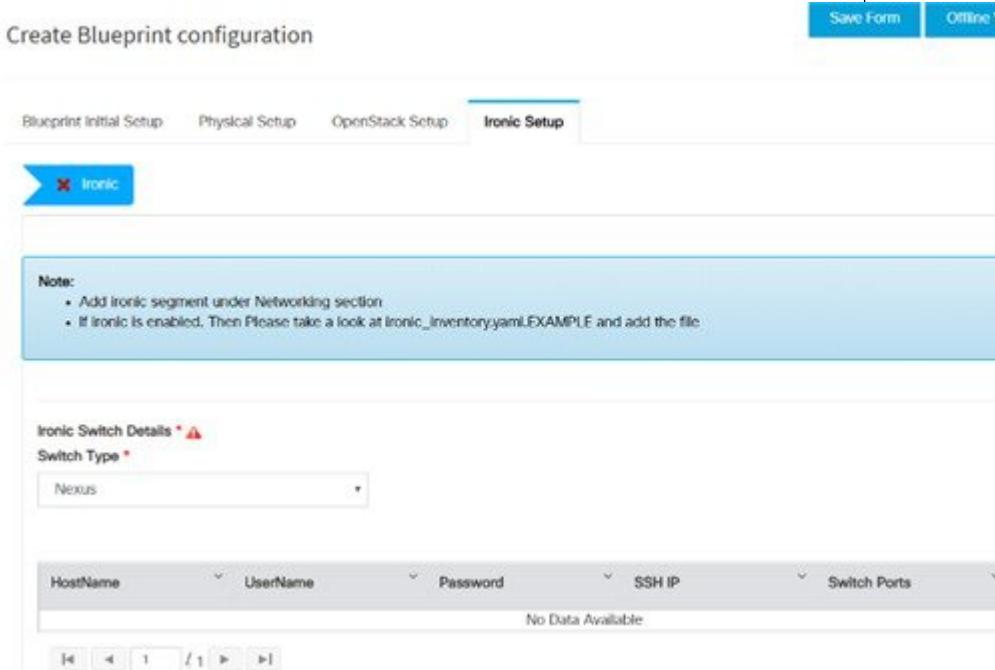
SolidFire is visible for configuration on day0
 SolidFire is not allowed as a day-2 deployment option
 SolidFire is always available with CEPH.

Cluster MVIP field	Management IP of SolidFire cluster.
Cluster SVIP field	Storage VIP of SolidFire cluster.
Admin Username	Admin user on SolidFire cluster
Admin Password	Admin password on SolidFire cluster.

- k. If **Syslog Export** or **NFVBENCH** is selected in **Blueprint Initial Setup** Page, then **Services Setup** page will be enabled for user to view. Following are the options under **Services Setup** Tab:

Name	Description																					
Syslog Export	<p>Following are the options for Syslog Settings:</p> <p>User can add maximum of three entries.</p> <p>To add new SysLog information, click on Add SysLog button, fill all the required information listed below and hit Save button.</p>  <table border="1" data-bbox="633 802 1428 1182"> <thead> <tr> <th>Remote host</th> <th>Protocol</th> <th>Facility</th> <th>Severity</th> <th>Port</th> <th>Clients</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>1.1.1.1</td> <td>udp</td> <td>local5</td> <td>debug</td> <td>514</td> <td>ELK</td> <td></td> </tr> <tr> <td>2.2.2.2</td> <td>udp</td> <td>local5</td> <td>debug</td> <td>514</td> <td>ELK</td> <td></td> </tr> </tbody> </table>	Remote host	Protocol	Facility	Severity	Port	Clients	Action	1.1.1.1	udp	local5	debug	514	ELK		2.2.2.2	udp	local5	debug	514	ELK	
Remote host	Protocol	Facility	Severity	Port	Clients	Action																
1.1.1.1	udp	local5	debug	514	ELK																	
2.2.2.2	udp	local5	debug	514	ELK																	

Name	Description
NFVBENCH	<p>NFVBENCH enable checkbox by default is false.</p> <p>Add ToR information connect to Switch:</p>  <p>The screenshot shows the 'Services Setup' tab selected in the top navigation bar. Under the 'NFVBENCH' section, there is a checked checkbox labeled 'Enable'. Below it, a section titled 'Add tor info connected to switch:' has a required field 'Select TOR Switches' with a dropdown menu showing 'TOR Switches' and an option 'IIS-N9372PK-2' which is selected. To the right, there is a 'Switch' dropdown set to 'IIS-N9372PK-2' with port 'eth1/33,eth1/34' selected. Below these are fields for 'NIC Ports' (INT1 and INT2) and 'NIC Slot' (set to 2).</p> <ul style="list-style-type: none"> • Select a TOR Switch and enter the Switch name. • Enter the port number. For Example: eth1/5 . VTEP VLANS (mandatory and needed only for VTS/VXLAN,): Enter 2 different VLANs for VLAN1 and VLAN2. • NIC Ports: INT1 and INT2 optional input. Enter the 2 port numbers of the 4-port 10G Intel NIC at the management node used for NFVbench. • NIC Slot: Optional input, should be in the range of 1-6, indicates which NIC to use in case there are multiple NICs. If nic_slot is defined, then nic_port has to be defined and vice-versa.
ENABLE_ESC_PRIV	Enable the checkbox to set it as True. By default, it is False .

Name	Description												
Ironic	<p>Following are the options for Ironic :</p> <ul style="list-style-type: none"> Ironic is applicable only for C-series and OVS/VLAN tenant network. Ironic is available in optional service list. If ironic is enabled, the Ironic Segment under Networks Segment and Ironic Switch Details under Ironic are mandatory.  <table border="1" data-bbox="621 1214 1494 1790"> <tbody> <tr> <td>Switch Type</td> <td>It can be Nexus, ACI, or BypassNeutron</td> </tr> <tr> <td>Hostname</td> <td>Enter ironic hostname. Required only if Switch Type is ACI or BypassNeutron.</td> </tr> <tr> <td>Username</td> <td>Enter ironic username. Required only if Switch Type is ACI or BypassNeutron.</td> </tr> <tr> <td>Password</td> <td>Enter the ironic password. Required only if Switch Type is ACI or BypassNeutron.</td> </tr> <tr> <td>SSH IP</td> <td>Enter ironic switch SSH IP. Required only if Switch Type is ACI or BypassNeutron.</td> </tr> <tr> <td>Switch Ports</td> <td>Optional. Indicates the ports that are in use to slap on inspector VLAN through Auto-ToR. Can be specified if Switch Type is ACI or BypassNeutron.</td> </tr> </tbody> </table>	Switch Type	It can be Nexus, ACI, or BypassNeutron	Hostname	Enter ironic hostname. Required only if Switch Type is ACI or BypassNeutron.	Username	Enter ironic username. Required only if Switch Type is ACI or BypassNeutron.	Password	Enter the ironic password. Required only if Switch Type is ACI or BypassNeutron.	SSH IP	Enter ironic switch SSH IP. Required only if Switch Type is ACI or BypassNeutron.	Switch Ports	Optional. Indicates the ports that are in use to slap on inspector VLAN through Auto-ToR. Can be specified if Switch Type is ACI or BypassNeutron.
Switch Type	It can be Nexus, ACI, or BypassNeutron												
Hostname	Enter ironic hostname. Required only if Switch Type is ACI or BypassNeutron.												
Username	Enter ironic username. Required only if Switch Type is ACI or BypassNeutron.												
Password	Enter the ironic password. Required only if Switch Type is ACI or BypassNeutron.												
SSH IP	Enter ironic switch SSH IP. Required only if Switch Type is ACI or BypassNeutron.												
Switch Ports	Optional. Indicates the ports that are in use to slap on inspector VLAN through Auto-ToR. Can be specified if Switch Type is ACI or BypassNeutron.												

Step 4 Click **Offline validation**, to initiate an offline validation of the Blueprint.

Step 5 Blueprint can also be created using an **Upload functionality**:

- In Blueprint Initial Setup.
- Click **Browse** in the blueprint initial setup.
- Select the YAML file you want to upload.
- Click **Select** button.
- Clicking on load button in the Insight UI Application. All the fields present in the YAML file would be uploaded to the respective fields in UI.
- Enter the name of the Blueprint (Make sure you enter unique name while saving Blueprints. There would be no two Blueprints with same name.)
- Click **Offline Validation**.
- If all the mandatory fields in the UI are populated, then Offline Validation of the Blueprint will start else a pop up would be visible which will inform which section of Blueprint Creation has a missing information error.
- On Validation Success of Blueprint **Save Blueprint** button will be enabled with **Cancel** button
- A pop up will be generated asking to initiate the deployment with **Blueprint Name** and the stages you need to run.
On Validation Failure of Blueprint **Cancel** button will be enabled.

Once the **Offline validation** is successful, **Save** option will be enabled which will redirect you to the Blueprint Management Page.

The wizard advances to the Blueprint Management page. On the Blueprint Management page you can select the recently added valid Blueprint and click **Install** button which is disabled by default.

A pop up will be generated asking to initiate the deployment with **Blueprint Name** and the stages you need to run.

By default all stages are selected but you can also do an incremented install.

In case of Incremented Install you should select stages in the order. For Example: If you select **Validation Stage** then the 2nd stage Management Node Orchestration will be enabled. You cannot skip stages and run a deployment.

Once you click **Proceed** the Cloud Deployment would be initiated and the progress can be viewed from "Dashboard".

Note Once the Blueprint is in **Active** State, the **Post-Install** features listed in Navigation Bar will change to **Active** stage.

Post Installation Features for Active Blueprint

This option is only available to a pod, which is successfully deployed. There are multiple sublinks available to manage the day-n operation of the pod. However, often Insight cross-launches the relevant services, through delegating the actual rendering to the individual services.

Monitoring the Pod

Cisco VIM uses ELK (elasticsearch, logstash and Kibana) to monitor the OpenStack services, by cross-launching the Kibana dashboard.

Cross Launching Horizon

To cross launch Kibana, complete the following instructions:

Step 1 Login as **POD User**.

Step 2 Naviagte to **POD**.

Step 3 Navigate to **Post-install**

Step 4 Click **Monitoring**

The **Authentication Required** browser pop up is displayed.

Step 5 Enter the **username** as admin.

Step 6 Enter the ELK_PASSWORD password obtained from /root/installer-<tagid>/openstack-configs/secrets.yaml in the management node.

Kibana is launched in an I-Frame

Note Click **Click here to view Kibana logs in new tab** link to view Kibana Logs in a new tab.

Cross Launching Horizon

Horizon is the canonical implementation of Openstack's Dashboard, which provides a web based user interface to OpenStack services including Nova, Swift and, Keystone.

Step 1 In the Navigation pane, click **Post-Install > Horizon**.

Click **Click here to view Horizon logs in new tab**.

You will be redirected to Horizon landing page in a new tab.

NFVI Monitoring

NFVI monitoring is a Cross launch browser same as Horizon. NFVI monitoring link is available in the post install only if the setupdata has NFVI Monitoring configuration during the cloud deployment which basically pings the monitoring and checks status of **Collector VM1 Info** and **Collector VM2 Info**.

Step 1 Login as **POD User**.

Step 2 Naviagte to **POD**.

Step 3 Navigate to **Post-install**

Step 4 Click **Reconfigure**.

Step 5 Click **NFVI Monitoring**

Step 6 Click the link **Click here to view NFVI monitoring..**

You will be redirected to NFVI monitoring page

Run VMTP

VIM 2.0, provides an integrated data and control plan test tool (called VMTP). VMTP helps you to test the cloud at any given time.

Run VMTP is divided in two sections:

- **Results for Auto Run:** Auto run shows the results of VMTP which was run during the cloud deployment (Blueprint Installation).
- **Results for Manual Run:** To run VMTP on demand click **Run VMTP**.



Note

If VMTP stage was skipped or not-run during Blueprint Installation, this section of POST Install gets disabled for the user.

Run CloudPulse

In VIM, we provide an integrated tool, called Cloud Pulse, that periodically checks the cloud services endpoint. The results of these tests are reflected under the Cloud Pulse link. You can also run these API endpoint tests on demand, and fetch the result of these tests by refreshing the table.

Endpoints Tests:

1. cinder_endpoint
2. glance_endpoint
3. keystone_endpoint
4. nova_endpoint
5. neutron_endpoint
6. all_endpoint_tests

Operator Tests:

1. rabbitmq_check
2. galera_check
3. ceph_check
4. node_check
5. docker_check
6. all_operator_tests

Run NFVbench

You can execute **Run NFV Bench** for BandC series Pod, through Cisco VIM Insight. On a pod running with Cisco VIM, click on the NFVbench link on the NAV-Menu.

Fixed Rate Test

You can run either fixed rate test or NDR/PDR test. As the settings and results for the test types differ, the options to run these tests are presented in two tabs, with its own settings and results.

NDR/PDR Test

Step 1 Log-in to **CISCO VIM Insight**.

Step 2 In the Navigation pane, click **Post-Install >Run NFV Bench**.

Step 3 Click on NDR/PDR test and complete the following fields.

Name	Description
Iteration Duration	Select duration from 10 to 60 sec. Default is 20 sec
Frame Size	Select the correct frame size to run
Run NDR/PDR test	Click on Run NDR/PDR test. Once NDR/PDR test is finished it will display each type of test with its own settings and results.

Fixed Rate Test

Step 1 Log in as **POD User**.

Step 2 Navigate to **POD**.

Step 3 Navigate to **Postinstall**.

Step 4 Click **Run NFV Bench**.

Step 5 Click Fixed rate test and complete the following fields.

Name	Description
Rate	Rate: Select right configuration pps or bps from drop down-list and enter values: For pps: minimum: 2500pps; maximum: 14500000pps (=14.5Mpps); default: 1000000pps (=1Mpps) For bps: minimum: 1400000bps; maximum: 10000000000bps (=10Gbps); default: 10000000000 (=1Gbps)
Iteration Duration	Select duration from 10-60Sec. Default is 20sec.
Frame Size	Select the right frame size(64,IMIX,1518) to run.
Run Fixed Rate Test	Click Run Fixed Rate Test . Once Fixed rate test is finished, it displays each type of test with its own settings and results.

POD Management

One of the key aspects of Cisco VIM is that it provides the ability for the admin to perform pod life-cycle management from a hardware and software perspective. Nodes of a given pod corrupts at times and VIM provides the ability to add, remove or replace nodes, based on the respective roles with some restrictions. Details of pod management will be listed in the admin guide, however as a summary the following operations are allowed on a running pod:

-
- Step 1** **Add or Remove Storage Nodes:** You can add one node at a time, given that we run Ceph as a distributed storage offering.
 - Step 2** **Add or Remove Computes Nodes:** N-computes nodes can be replaced simultaneously; however at any given point, at least one compute node should be active.
 - Step 3** **Replace Control Nodes:** We do not support double fault scenarios, replacement of one controller at a time is supported.
-

System Update

As part of the lifecycle management of the cloud, VIM has the ability to bring in patches (bug fixes related to code, security, etc.), thereby providing the additional value of seamless cloud management from software perspective. Software update of the cloud is achieved by uploading a valid tar file following initiation of a System Update from the Insight as follows:

-
- Step 1** Login as **POD User**.
 - Step 2** Naviagte to **POD**.
 - Step 3** Navigate to **Post-install**
 - Step 4** Click **System Update**.
 - Step 5** Click **Openstack Password**
 - Step 6** Click **Browse** button.
 - Step 7** Select the valid tar file.
 - Step 8** Click **Open > Upload and Update**.

Message stating System Update has been initiated will be displayed. Logs front-ended by hyperlink would be visible in the section below before Update Logs to help see the progress of the update. During the software update, all other pod management activities will be disabled. Post-update, normal cloud management will commence.

Reconfiguring CIMC Password through Insight

Update the cimc_password in the CIMC-COMMON section, and/or the individual cimc_password for each server and then run the update password option.

To update a password, you need to follow the password rules:

- Must contain at least one lower case letter.
- Must contain at least one upper case letter.
- Must contain at least one digit between 0 to 9.

Reconfiguring OpenStack Password

- One of these special characters !\$#@%^-_+=*&
- Your password has to be 8 to 14 characters long.

Before you begin

You must have a C-series pod up and running with Cisco VIM to reconfigure CIMC password.



Note Reconfigure CIMC password section would be disabled if the pod is in failed state as indicated by ciscovim install-status.

Step 1 Login as **POD User**.

Step 2 Naviagte to **POD**.

Step 3 Navigate to **Post-install**

Step 4 Click **Reconfigure**.

Step 5 Click **Openstack Password**

Name	Description
CIMC_COMMON old Password	CIMC_COMMON old password field cannot be edited.
CIMC-COMMON new Password	Enter new CIMC-COMMON password. Password should be alphanumeric according to the password rule.
Click Update Password	Old CIMC-COMMON password will be updated with new CIMC-COMMON password.

Reconfiguring OpenStack Password

Cisco VIM has been designed with security to accommodate users password policy.

There are two options to regenerate the Password:

1. **Regenerate all passwords:** Check the **Regenerate all passwords** checkbox and click **Set Password**. This automatically regenerates all passwords in alphanumeric format.
2. **Regenerate single or more password:** If you want to set a specific password for any service like Horizon's ADMIN_USER_PASSWORD you can add it by doing an inline edit. Double click field under Password and then enter the password which enables **Set Password**.



Note During the reconfiguration of password, all other pod management activities are disabled. Postupdate, normal cloud management commences.

Reconfiguring OpenStack Services, TLS certs and ELK configurations

Cisco VIM supports the reconfiguration of OpenStack log level services, TLS certificates, and ELK configuration. Listed below are the steps to reconfigure the OpenStack and other services:

Step 1 Login as **POD User**.

Step 2 Naviagte to **POD**.

Step 3 Navigate to **Post-install**

Step 4 Click **Reconfigure OpenStack Config**.

Step 5 Click on the specific item to be changed and updated; For TLS certificate it is the path to certificate location.

Step 6 Enter **Set Config** and the process will commence.

During the reconfiguration process, all other pod management activities will be disabled. Post-update, normal cloud management will commence.

Reconfiguring Optional Services

Cisco VIM offers optional services such as heat, NFVbench, NFVIMON, CVIM-MON and so on, that can be enabled as post-pod deployment. Optional services can be un-configured as post-deployment in Cisco VIM feature set. These services can be enabled in one-shot or selectively. Listed below are the steps to enable optional services:

Step 1 Login as **POD User**.

Step 2 Naviagte to **POD**.

Step 3 Navigate to **Post-install**

Step 4 Click **Reconfigure Optional Services**.

Step 5 Choose the right service and update the fields with the right values.

Step 6 Enter **Reconfigure** to commence the process.

During the reconfiguration process, all other pod management activities will be disabled. Post-update, normal cloud management will commence. Once reconfigure is initiated than optional feature would be updated in active blueprint. If reconfigure of Optional Services fail in the time of reconfigure process then it is advised to contact CiscoTAC to resolve the situation through CLI.

Note All reconfigure operation feature contains repeated deployment true or false.

- Repeated re-deployment true - Feature can be re-deployed again.
- Repeated re-deployment false- Deployment of feature allowed only once.

Deployment Status :

Optional Features	Repeated re-deployment Options
APICINFO	True

Optional Features	Repeated re-deployment Options
DHCP Reservation for Virtual MAC Addresses	True
EXTERNAL_LB_VIP_FQDN	False
EXTERNAL_LB_VIP_TLS	False
INSTALL_MODE	True
LDAP	True
NETWORKING	True
NFVBENCH	False
NFVIMON	False
PODNAME	False
PROVIDER_VLAN_RANGES	True
SWIFTSTACK	True
SYSLOG_EXPORT_SETTINGS	False
TENANT_VLAN_RANGES	True
TORSWITCHINFO	False
VIM_ADMINNS	True
VMTP	False
VTS_PARAMETERS	False
AUTOBACKUP	'
	True
Heat	False
Ceilometer	False
HTTP Proxy Server	True
HTTPS Proxy Server	True
Enable TTY LOGGING	False
MGMTNODE_EXTAPI_REACH	False
Cobbler	True
SNMP	True

Optional Features	Repeated re-deployment Options
Base MAC address	True

Pod User Administration

Cisco VIM Insight offers Users (Pod Admin(s) or Pod Users) to manage Users and roles associated with them.

Managing Users

To add new User

Step 1 Click **Login as POD User**.

Step 2 Navigate to **POD User Administration**.

Step 3 Click **Manage Users**.

Step 4 Click **Add Users** to add a new user.

Step 5 Complete the following fields in the **Add Users** page of the Cisco VIM Insight:

Field Name	Field Description
Email ID	Enter the Email ID of the User.
User Name	Enter the User Name if the User is new. If the User is already registered to the Insight the User-Name gets auto-populated.
Role	Select the Role from the drop-down list.

Step 6 Click **Save**.

Managing Roles

To create a new Role:

Step 1 Click **Log in as POD User**.

Step 2 Navigate to **Pod User Administration** and click **Manage Roles**. By default you will see a full-pod-access role in the table.

Step 3 Click **Add Role** to create a new role.

Step 4 Complete the following fields on the **Add Roles** page in Cisco VIM Insight:

Field Name	Field Description
Role	Enter the name of the role.
Description	Enter the description of the role.

Managing Root CA Certificate

Field Name	Field Description
Permission	Check the Permission checkbox to select the permission.

Step 5 Click **Save**. Once, the Blueprint is in an Active state all the permissions are same for C-series and B-series Pods other than Reconfigure CIMC Password which is missing for B-series Pod.

Note Permissions are divided in the granular level where viewing *Dashboard* is the default role that is added while creating a role.

Managing Root CA Certificate

You can update the CA Certificate during the registration of the POD. Once, logged in as POD User and if you have the permission to update the certificate you can view under POD User Administration>> Manage Root CA Certificate.

To update the Certificate:

Step 1 Click **Login as POD User**

Step 2 Navigate to **POD User Administration>>Manage Root CA certificate**.

Step 3 Click **Browse** and select the certificate that you want to upload.

Step 4 Click **Upload**.

- If the certificate is Invalid, and does not matches with the certificate on the management node located at (var/www/mercury/mercury-ca.crt) then Insight will revert the certificate which was working previously.
- If the Certificate is valid, Insight will run a management node health check and then update the certificate with the latest one.

Note The CA Certificate which is uploaded should be same as the one which is in the management node.



CHAPTER 9

Verifying the Cisco NFVI Installation

The following topics provide quick methods for checking and assessing the Cisco NFVI installation.

- [Displaying Cisco NFVI Node IP Addresses, on page 347](#)
- [Verifying Cisco VIM Client CLI Availability, on page 348](#)
- [Displaying Cisco NFVI Logs, on page 349](#)
- [Accessing OpenStack API Endpoints, on page 349](#)
- [Assessing Cisco NFVI Health with CloudPulse, on page 350](#)
- [Displaying HA Proxy Dashboard and ELK Stack Logs, on page 352](#)
- [Checking Cisco NFVI Pod and Cloud Infrastructure, on page 352](#)

Displaying Cisco NFVI Node IP Addresses

To display the IP addresses for all Cisco NFVI nodes, enter the following command:

```
# cd /root/openstack-configs  
[root @nfvi_management_node openstack-configs]# cat  
/root/installer/openstack-configs/mercury_servers_info
```

The following is the sample output:

```
Total nodes: 8  
Controller nodes: 3  
+-----+-----+-----+-----+-----+  
| Server | CIMC | Management | Provision | Tenant | Storage |  
+-----+-----+-----+-----+-----+  
| c44-control-1 | 172.26.233.54 | 10.21.1.25 | 10.21.1.25 | 10.2.2.22 | None |  
| | | | | |  
| c44-control-3 | 172.26.233.56 | 10.21.1.27 | 10.21.1.27 | 10.2.2.24 | None |  
| | | | | |  
| c44-control-2 | 172.26.233.55 | 10.21.1.28 | 10.21.1.28 | 10.2.2.25 | None |  
| | | | | |  
+-----+-----+-----+-----+-----+  
Compute nodes: 2  
+-----+-----+-----+-----+  
| Server | CIMC | Management | Provision | Tenant | Storage |  
+-----+-----+-----+-----+-----+  
| c44-compute-1 | 172.26.233.57 | 10.21.1.26 | 10.21.1.26 | 10.2.2.23 | None |  
| | | | | |  
| c44-compute-2 | 172.26.233.58 | 10.21.1.23 | 10.21.1.23 | 10.2.2.21 | None |  
| | | | | |  
+-----+-----+-----+-----+  
Storage nodes: 3  
+-----+-----+-----+-----+
```

Verifying Cisco VIM Client CLI Availability

Server	CIMC	Management	Provision	Tenant	Storage
c44-storage-3	172.26.233.53	10.21.1.22	10.21.1.22	None	10.3.3.22
c44-storage-2	172.26.233.52	10.21.1.24	10.21.1.24	None	10.3.3.23
c44-storage-1	172.26.233.51	10.21.1.21	10.21.1.21	None	10.3.3.21

[root@c44-top-mgmt openstack-configs]#

Verifying Cisco VIM Client CLI Availability

Cisco VIM Client CLI is used for managing Cisco NFVI pods. After the Cisco NFVI installation is complete, verify that the Cisco VIM user is running and pointing to the right management node in the installer directory. Cisco NFVI provides a tool that you can use to check the REST API server status and directory where it is running.

To start the tool, enter the following:

```
# cd installer-<tagid>/tools
#!/usr/bin/python
Status of the REST API Server: active (running) since Thu 2016-08-18 09:15:39 UTC; 9h ago
REST API launch directory: /root/installer-<tagid>/
```

Confirm that the server status is active and check that the REST API directory matches the directory where the installation is launched.

The REST API command also provides options to start, tear down, and reset the REST API server password. Run the following REST API command to rest the password.

```
# ./restapi.py -h
usage: restapi.py [-h] --action ACTION [--yes] [--verbose]

REST API setup helper

optional arguments:
  -h, --help            show this help message and exit
  --action ACTION, -a ACTION
                        setup - Install and Start the REST API server.
                        teardown - Stop and Uninstall the REST API
                        server.
                        restart - Restart the REST API server.
                        regenerate-password - Regenerate the password for
                        REST API server.
                        reset-password - Reset the REST API password with
                        user given password.
                        status - Check the status of the REST API server
                        Skip the dialog. Yes to the action.
  --yes, -y              Perform the action in verbose mode.
  --verbose, -v
```

If the REST API server is not running, executing **ciscovim** displays the following error message:

```
# cd installer-<tagid>
# ciscovim -setupfile ~/Save/<setup_data.yaml> run
```

If the installer directory or the REST API state is not correct or pointing to an incorrect REST API directory, go to the `installer-<tagid>/tools` dir and execute the following command:

```
# ./restapi.py -action setup
```

Confirm that the REST API server state and directory is correct:

```
# ./restapi.py -action status
```

If the REST API recovery step was run on an existing pod, run the following command to ensure that the REST API server continues to manage it:

```
# cd installer-<tagid>/  
# ciscovim --setupfile <setup_data_file_path> --perform 7 -y
```



Note Detailed information about the Cisco NFVI REST API is provided in the Cisco Network Function Virtualization Infrastructure Administrator Guide.

Displaying Cisco NFVI Logs

Cisco NFVI installation logs are generated in the management node /var/log/mercury//<install_uuid>/ directory. The last 20-log directories are tarred and kept in this directory. The logs are archived (tar.gz file) after each run.

The following table lists the Cisco NFVI installation steps and corresponding log files:

Table 25: Cisco NFVI Installation Logs

Step	Description	Log File
1	INPUT_VALIDATION	mercury_baremetal_install.log
2	MGMTNODE_ORCHESTRATION	mercury_buildorchestration.log
3	VALIDATION	mercury_baremetal_install.log
4	BAREMETAL	mercury_baremetal_install.log
5	COMMONSETUP	mercury_os_install.log
6	CEPH	mercury_ceph_install.log
7	ORCHESTRATION	mercury_os_install.log
8	VMTP	None

Accessing OpenStack API Endpoints

The Cisco NFVI installer stores access credentials in the management node /root/installer-<tag-number>/openstack-configs/openrc. The external_lb_vip_address provided in setup_data.yaml is the IP address where OpenStack APIs are handled.

Following is an example:

```
export OS_AUTH_URL=http://172.26.233.139:5000/v2.0 or
https://172.26.233.139:5000/v2.0 (if TLS is enabled)
export OS_USERNAME=admin
export OS_PASSWORD=xyzabcd
export OS_TENANT_NAME=admin
export OS_REGION_NAME=RegionOne
# For TLS, add
export OS_CACERT=/root/openstack-configs/haproxy-ca.crt
```

The corresponding setup_data.yaml entry:

```
#####
# HA Proxy
#####
external_lb_vip_address: 172.26.233.139
```

Assessing Cisco NFVI Health with CloudPulse

You can use the OpenStack CloudPulse tool to verify Cisco NFVI health. CloudPulse servers are installed in containers on all Cisco NFVI control nodes, and CloudPulse users are installed on the management node. Run the following commands to display Cisco NFVI information. For information about CloudPulse, visit the [OpenStack CloudPulse website](#).

To check the results of periodic CloudPulse runs:

```
# cd /root/openstack-configs
# source openrc
# cloudpulse result
+-----+-----+-----+-----+
| uuid | id | name | testtype | state |
+-----+-----+-----+-----+
| bf7fac70-7e46-4577-b339-b1535b6237e8 | 3788 | glance_endpoint | periodic | success |
| 1f575ad6-0679-4e5d-bc15-952bade09f19 | 3791 | nova_endpoint | periodic | success |
| 765083d0-e000-4146-8235-ca106fa89864 | 3794 | neutron_endpoint | periodic | success |
| c1c8e3ea-29bf-4fa8-91dd-c13a31042114 | 3797 | cinder_endpoint | periodic | success |
| 04b0cb48-16a3-40d3-aa18-582b8d25e105 | 3800 | keystone_endpoint | periodic | success |
| db42185f-12d9-47ff-b2f9-4337744bf7e5 | 3803 | glance_endpoint | periodic | success |
| 90aa9e7c-99ea-4410-8516-1c08beb4144e | 3806 | nova_endpoint | periodic | success |
| d393a959-c727-4b5e-9893-e229efb88893 | 3809 | neutron_endpoint | periodic | success |
| 50c31b57-d4e6-4cf1-a461-8228fa7a9be1 | 3812 | cinder_endpoint | periodic | success |
| d1245146-2683-40da-b0e6-dbf56e5f4379 | 3815 | keystone_endpoint | periodic | success |
| ce8b9165-5f26-4610-963c-3ff12062a10a | 3818 | glance_endpoint | periodic | success |
| 6a727168-8d47-4a1d-8aa0-65b942898214 | 3821 | nova_endpoint | periodic | success |
| 6fbf48ad-d97f-4a41-be39-e04668a328fd | 3824 | neutron_endpoint | periodic | success |
```

To run a CloudPulse test on demand:

```
# cd /root/openstack-configs
# source openrc
# cloudpulse run --name <test_name>
# cloudpulse run --all-tests
# cloudpulse run --all-endpoint-tests
# cloudpulse run --all-operator-tests
```

To run a specific CloudPulse test on demand:

```
[root@vms-line2-build installer-3128.2]# cloudpulse run --name neutron_endpoint
+-----+-----+
| Property | Value |
```

```
+-----+-----+
| name      | neutron_endpoint           |
| created_at | 2016-03-29T02:20:16.840581+00:00 |
| updated_at | None                     |
| state     | scheduled                |
| result    | NotYetRun               |
| testtype   | manual                   |
| id        | 3827                     |
| uuid      | 5cc39fa8-826c-4a91-9514-6c6de050e503 |
+-----+-----+
[root@vms-line2-build installer-3128.2]#
```

To show detailed results from a specific CloudPulse run:

```
[root@vms-line2-build installer-3128.2]# cloudpulse show 5cc39fa8-826c-4a91-9514-6c6de050e503
+-----+-----+
| Property | Value          |
+-----+-----+
| name      | neutron_endpoint           |
| created_at | 2016-03-29T02:20:16+00:00 |
| updated_at | 2016-03-29T02:20:41+00:00 |
| state     | success                  |
| result    | success                  |
| testtype   | manual                   |
| id        | 3827                     |
| uuid      | 5cc39fa8-826c-4a91-9514-6c6de050e503 |
+-----+-----+
```

CloudPulse has two test sets: endpoint_scenario (runs as a cron or manually) and operator test (run manually). Endpoint tests include:

- nova_endpoint
- neutron_endpoint
- keystone_endpoint
- glance_endpoint
- cinder_endpoint

Operator tests include

- ceph_check
- docker_check
- galera_check
- node_check
- rabbitmq_check

The following table lists the operator tests that you can perform with CloudPulse.

Table 26: CloudPulse Operator Tests

Test	Description
Ceph Check	Executes the ceph -f json status command on the Ceph-mon nodes and parses the output. If the result of the output is not HEALTH_OK, the ceph_check reports an error.
Docker Check	Finds out if all Docker containers are in running state on all nodes and reports an error if any containers are in the Exited state. The Docker check runs the command, docker ps -aq --filter 'status=exited'.
Galera Check	Executes the command, mysql 'SHOW STATUS', on the controller nodes and displays the status.
Node Check	Checks if all the nodes in the system are up and online. It also compares the results of the Nova hypervisor list and determines whether all the compute nodes are available.
RabbitMQ Check	Runs the command, rabbitmqctl cluster_status, on the controller nodes and finds out if the RabbitMQ cluster is in quorum. If nodes are offline, the rabbitmq_check reports a failure.

Displaying HA Proxy Dashboard and ELK Stack Logs

You can view the HA Proxy dashboard at: `http://<external_lb_vip_address>:1936` using the following username and password.

- Username—haproxy
- Password—Value for Haproxy_PASSWORD in /root/installer-<tag-number>/openstack-configs/secrets.yaml

You can use the Kibana dashboard to view logs aggregated by Logstash at: `http://<management_node_IP>:5601` using the following username and password.

- Username—admin
- Password—Value for ELK_PASSWORD in /root/installer-<tag-number>/openstack-configs/secrets.yaml

Checking Cisco NFVI Pod and Cloud Infrastructure

To test the Cisco NFVI pod and cloud infrastructure (host connectivity, basic mraiadb, rabbit, ceph cluster check, and RAID disks), you can use the cloud-sanity tool available on the management node.

**Note**

For details on the execution of cloud-sanity with Cisco VIM, see [Assessing Cisco NFVI Status with Cloud-Sanity](#) of *Cisco Virtualized Infrastructure Manager Administrator Guide, Release 3.0.0*



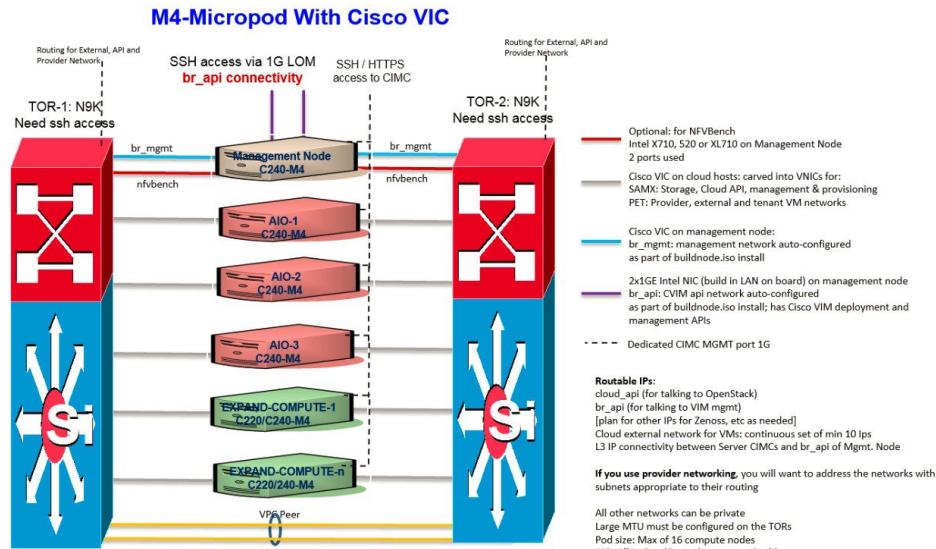
APPENDIX A

Appendix

- Cisco VIM Wiring Diagrams, on page 353

Cisco VIM Wiring Diagrams

Figure 43: M4-Micropod with Cisco VIC



Appendix

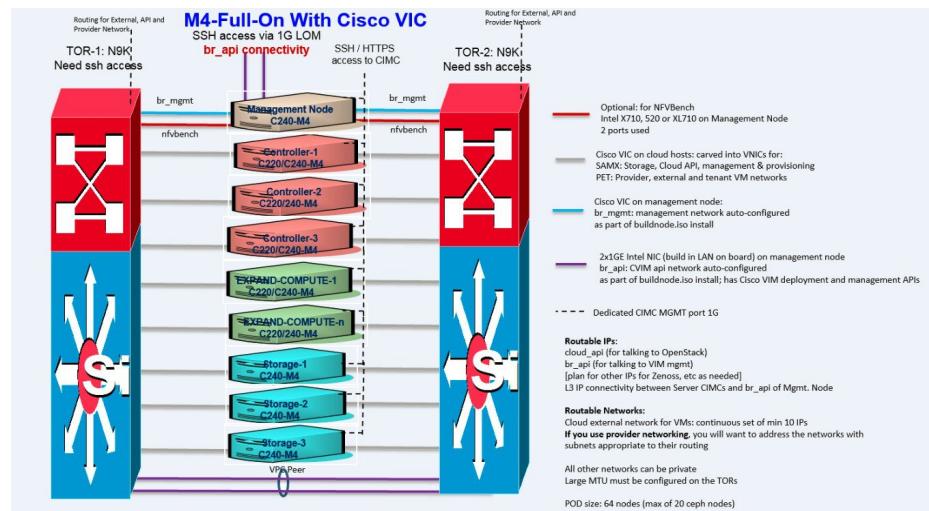
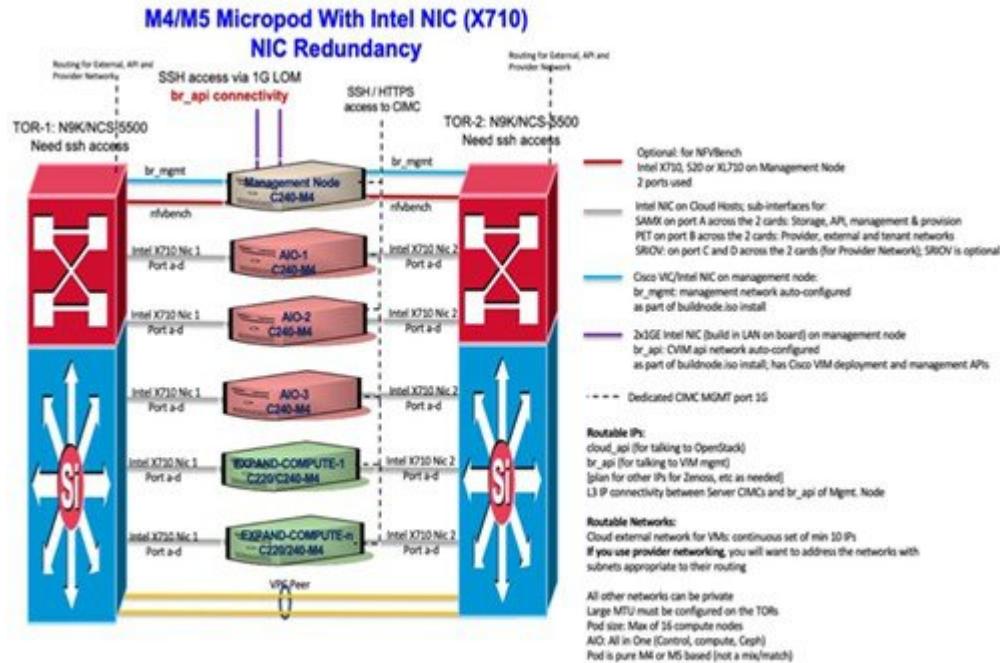
Figure 44: M4-Full-On with Cisco VIC**Figure 45: M4/M5 Micropod with Intel NIC (X710) - NIC Redundancy**

Figure 46: M4 Hyperconverged with Cisco VIC/NIC (1xX710) VPP based; no SRIOV

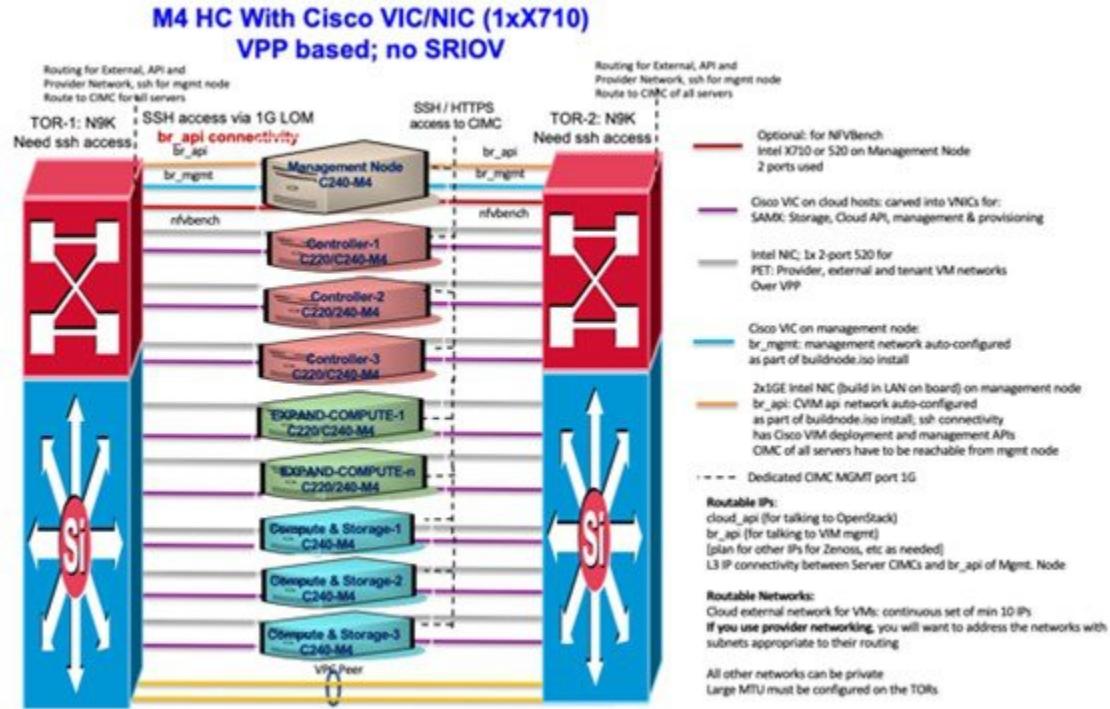


Figure 47: M5-Micropod with Intel NIC (X710) - No NIC Redundancy

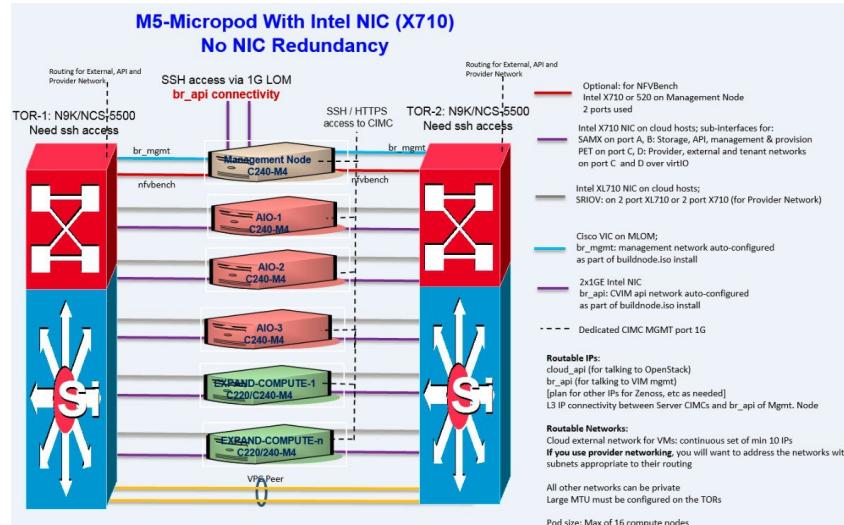


Figure 48: M4/M5 Full-On with Intel NIC (X710) and with NIC Redundancy

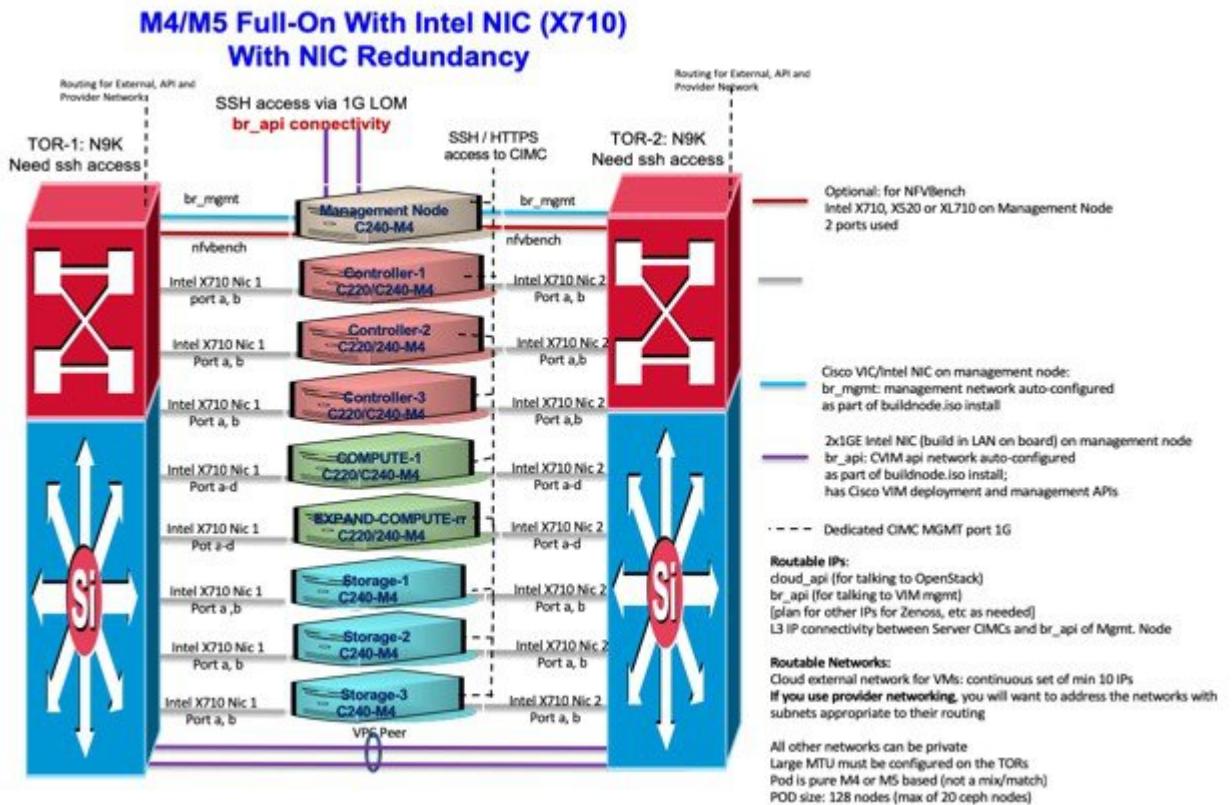


Figure 49: M4/M5 Full-On with Cisco VIC/NIC (2xXL710/2x520)

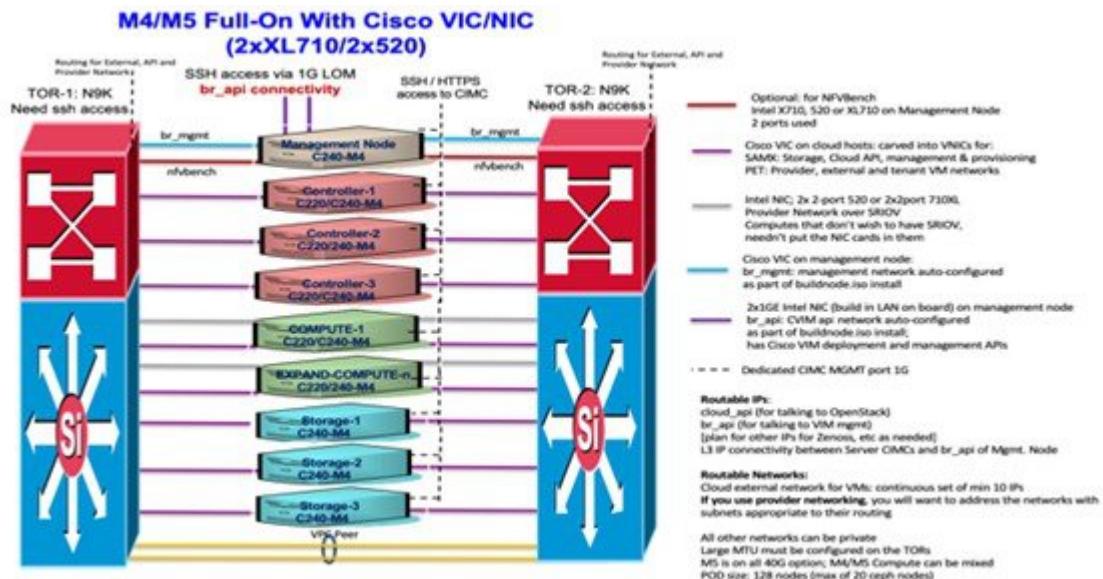


Figure 50: M4/M5 Micropod with Cisco VIC/NIC (2xXL710/2x520)

M4/M5 Micropod With Cisco VIC/NIC (2xXL710/2x520)

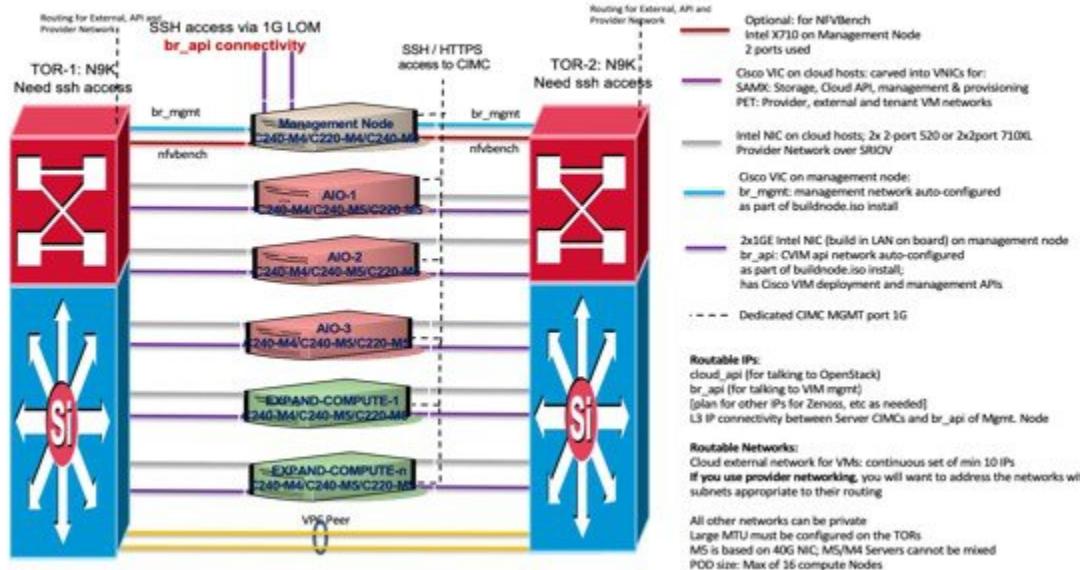
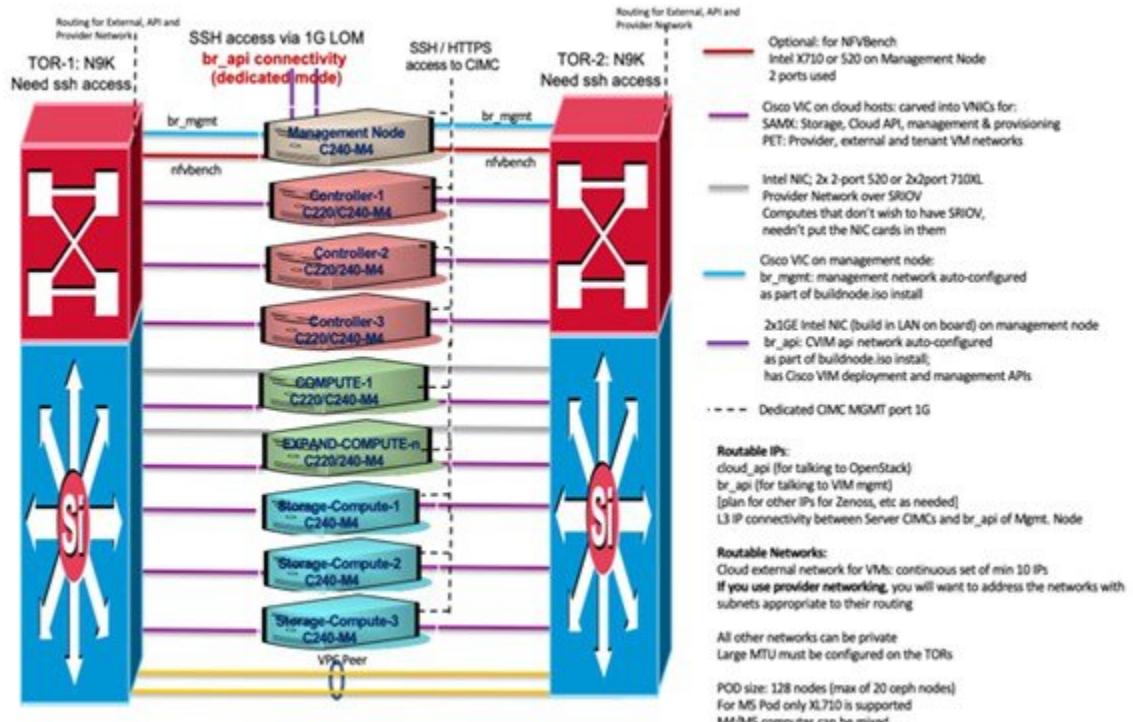


Figure 51: M4/M5-HC with Cisco VIC/NIC (2xXL710/2x520)



Appendix

Figure 52: Quanta (D52BQ-2U 3UPI) Fullon Pod with 25GE Intel NIC (xxv710)

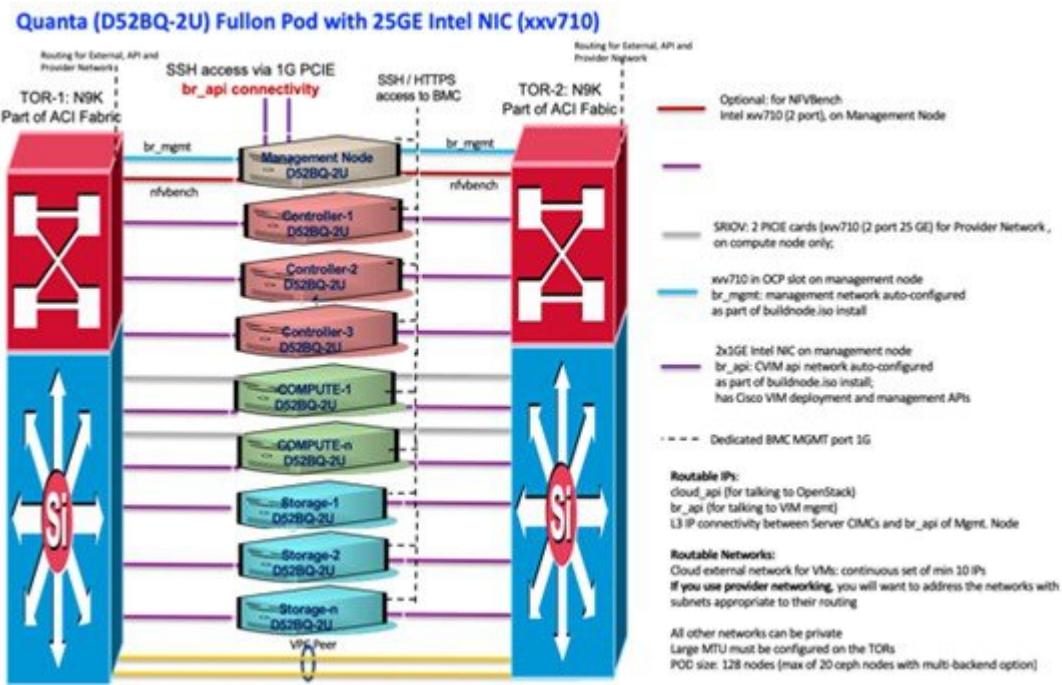


Figure 53: Quanta (D52BE-2U) Edge Pod with 25GE Intel NIC (xxv710)

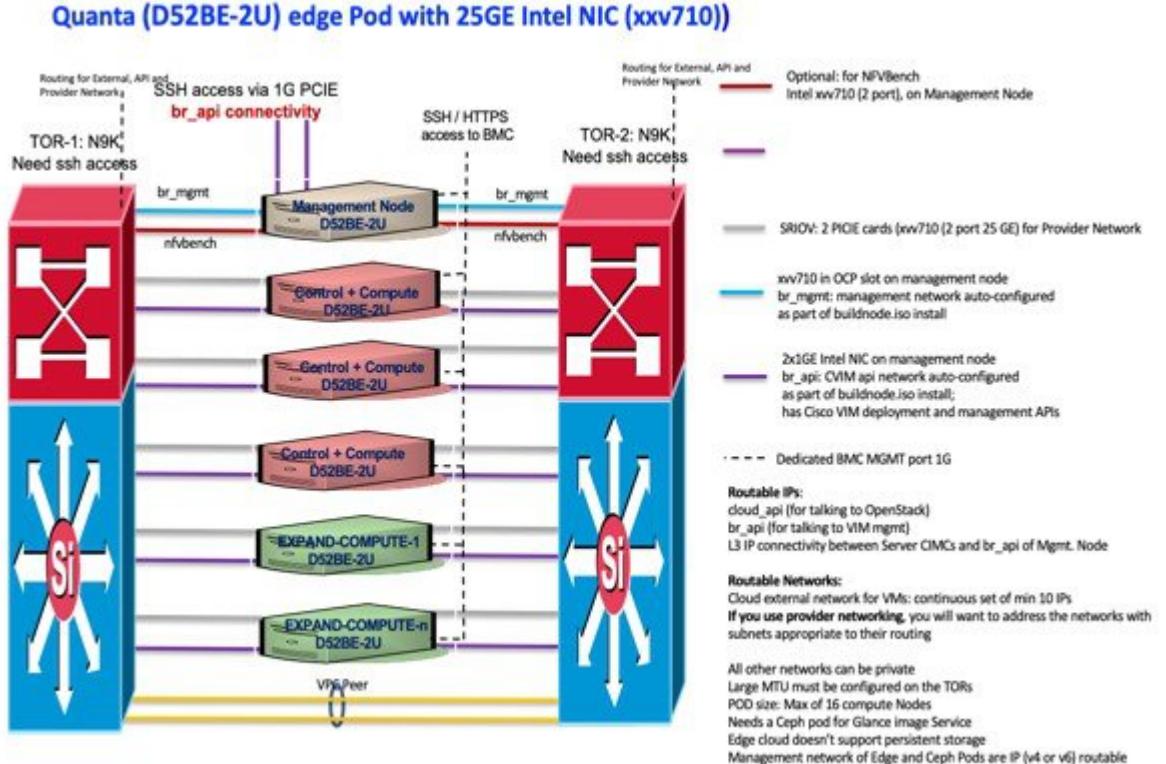


Figure 54: Quanta (D52BQ-2U 3UPI) Ceph Pod with 25GE Intel NIC (xxv710)

