# Ultra M Solutions Guide, Release 6.0

**First Published:** 2018-01-25

**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

# CONTENTS

# About This Guide

This preface describes the *Ultra M Solution Guide*, how it is organized, and its document conventions.

Ultra M is a pre-packaged and validated virtualized mobile packet core solution designed to simplify the deployment of virtual network functions (VNFs).



# Conventions Used

The following tables describe the conventions used throughout this documentation.

| Notice Type | Description |
|---|---|
| Information Note | Provides information about important features or instructions. |
| Caution | Alerts you of potential damage to a program, device, or system. |
| Warning | Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards. |

| Typeface Conventions | Description |
|---|---|
| Text represented as a `screen display` | This typeface represents displays that appear on your terminal screen, for example:<br>`Login:` |
| Text represented as **commands** | This typeface represents commands that you enter, for example:<br>**show ip access-list**<br>This document always gives the full form of a command in lowercase letters. Commands are not case sensitive. |

| Typeface Conventions | Description |
|---|---|
| Text represented as a **command** *variable* | This typeface represents a variable that is part of a command, for example:<br>**show card** *slot_number*<br>*slot_number* is a variable representing the desired chassis slot number. |
| Text represented as menu or sub-menu names | This typeface represents menus and sub-menus that you access within a software application, for example:<br>Click the **File** menu, then click **New** |

# Supported Documents and Resources

## Related Documentation

The most up-to-date information for the UWS is available in the product *Release Notes* provided with each product release.

The following common documents are available:

- *Ultra Gateway Platform System Administration Guide*
- *Ultra-M Deployment Guide*
- *Ultra Services Platform Deployment Automation Guide*
- *Ultra Services Platform NETCONF API Guide*
- *VPC-DI System Administration Guide*
- *StarOS Product-specific and Feature-specific Administration Guides*

## Obtaining Documentation

### Nephelo Documentation

The most current Nephelo documentation is available on the following website: http://nephelo.cisco.com/page_vPC.html

### StarOS Documentation

The most current Cisco documentation is available on the following website: http://www.cisco.com/cisco/web/psa/default.html

Use the following path selections to access the StarOS documentation:

Products > Wireless > Mobile Internet > Platforms > Cisco ASR 5000 Series > Configure > Configuration Guides

# Contacting Customer Support

Use the information in this section to contact customer support.

Refer to the support area of http://www.cisco.com for up-to-date product documentation or to submit a service request. A valid username and password are required to access this site. Please contact your Cisco sales or service representative for additional information.

CHAPTER 1

# Ultra M Overview

Ultra M is a pre-packaged and validated virtualized mobile packet core solution designed to simplify the deployment of virtual network functions (VNFs).

The solution combines the Cisco Ultra Service Platform (USP) architecture, Cisco Validated OpenStack infrastructure, and Cisco networking and computing hardware platforms into a fully integrated and scalable stack. As such, Ultra M provides the tools to instantiate and provide basic lifecycle management for VNF components on a complete OpenStack virtual infrastructure manager.



## VNF Support

In this release, Ultra M supports the Ultra Gateway Platform (UGP) VNF.

The UGP currently provides virtualized instances of the various 3G and 4G mobile packet core (MPC) gateways that enable mobile operators to offer enhanced mobile data services to their subscribers. The UGP addresses the scaling and redundancy limitations of VPC-SI (Single Instance) by extending the StarOS boundaries beyond a single VM. UGP allows multiple VMs to act as a single StarOS instance with shared interfaces, shared service addresses, load balancing, redundancy, and a single point of management.

## Ultra M Model(s)

The Ultra M Extra Small (XS) model is currently available. It is based on OpenStack 10 and implements a Hyper-Converged architecture that combines the Ceph Storage and Compute node. The converged node is referred to as an OSD compute node.

This model includes 6 Active Service Functions (SFs) per VNF and is supported in deployments from 1 to 4 VNFs.

# Functional Components

As described in Hardware Specifications, on page 7, the Ultra M solution consists of multiple hardware components including multiple servers that function as controller, compute, and storage nodes. The various functional components that comprise the Ultra M are deployed on this hardware:

- **OpenStack Controller**: Serves as the Virtual Infrastructure Manager (VIM).

  **Important** In this release, all VNFs in a multi-VNF Ultra M are deployed as a single “site” leveraging a single VIM.

- **Ultra Automation Services (UAS)**: A suite of tools provided to simplify the deployment process:
- **Cisco Elastic Services Controller (ESC)**: Serves as the Virtual Network Function Manager (VNFM).

  **Important** ESC is the only VNFM supported in this release.

- **VNF Components**: USP-based VNFs are comprised of multiple components providing different functions:
  - **Ultra Element Manager (UEM)**: Serves as the Element Management System (EMS, also known as the VNF-EM); it manages all of the major components of the USP-based VNF architecture.
  - **Control Function (CF)**: A central sub-system of the UGP VNF, the CF works with the UEM to perform lifecycle events and monitoring for the UGP VNF.

◦ **Service Function (SF)**: Provides service context (user I/O ports), handles protocol signaling, session processing tasks, and flow control (demux).

*Figure 1: Ultra M Components*

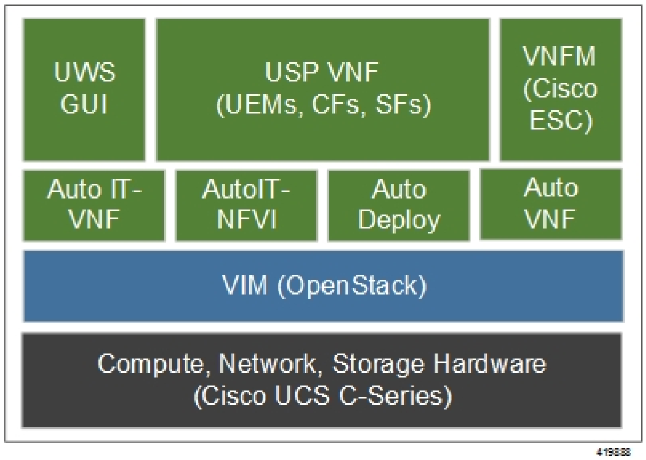
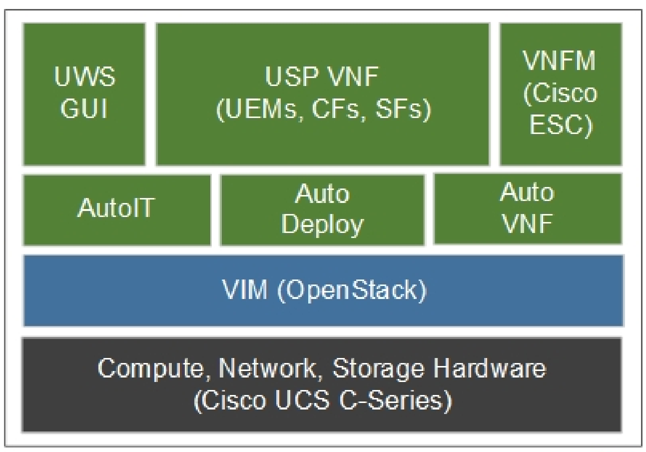



*Figure 2: Ultra M Components*




# Virtual Machine Allocations

Each of the Ultra M functional components are deployed on one or more virtual machines (VMs) based on their redundancy requirements as identified in Table 1: Function VM Requirements per Ultra M Model, on page 4. Some of these component VMs are deployed on a single compute node as described in VM Deployment per Node Type, on page 11. All deployment models use three OpenStack controllers to provide VIM layer redundancy and upgradability.

*Table 1: Function VM Requirements per Ultra M Model*

| | Hyper-Converged | |
|---|---|---|
| **Function(s)** | **XS Single VNF** | **XS Multi VNF** |
| OSP-D* | 1 | 1 |
| AutoIT** | 2 | 2 |
| AutoDeploy** | 2 | 2 |
| AutoVNF | 2 | 2 |
| ESC (VNFM) | 2 | 2 per VNF |
| UEM | 3 | 3 per VNF |
| CF | 2 | 2 per VNF |
| SF | 8 | 8 per VNF |
| * OSP-D is deployed as a VM for Hyper-Converged Ultra M models.<br>** AutoIT and AutoDeploy each require 2 VMs when deployed in HA mode (recommeded). | | |

# VM Resource Requirements

The CF, SF, UEM, and ESC VMs require the resource allocations identified in Table 2: VM Resource Allocation, on page 4. The host resources are included in these numbers.

*Table 2: VM Resource Allocation*

| Virtual Machine | vCPU | RAM (GB) | Root Disk (GB) |
|---|---|---|---|
| OSP-D* | 16 | 32 | 200 |
| AutoIT** | 2 | 8 | 80 |
| AutoDeploy** | 2 | 8 | 80 |
| AutoVNF | 2 | 4 | 40 |
| ESC | 2 | 4 | 40 |
| UEM | 2 | 4 | 40 |
| CF | 8 | 16 | 6 |

| Virtual Machine | vCPU | RAM (GB) | Root Disk (GB) |
|---|---|---|---|
| SF | 24 | 96 | 4 |

**Note** 4 vCPUs, 2 GB RAM, and 54 GB root disks are reserved for host reservation.

* OSP-D is deployed as a VM for Hyper-Converged Ultra M models. Though the recommended root disk size is 200GB, additional space can be allocated if available.

** AutoIT is used to deploy the VIM Orchestrator (Undercloud) and VIM (Overcloud) for Hyper-Converged Ultra M models. AutoIT, AutoDeploy, and OSP-D are installed as VMs on the same physical server in this scenario.

CHAPTER 2

# Hardware Specifications

Ultra M deployments use the following hardware:

**Note** The specific component software and firmware versions identified in the sections that follow have been validated in this Ultra M solution release.



# Cisco Catalyst Switches

Cisco Catalyst Switches provide as physical layer 1 switching for Ultra M components to the management and provisioning networks. One of two switch models is used based on the Ultra M model being deployed:


- Catalyst C2960XR-48TD-I Switch, on page 7
- Catalyst 3850-48T-S Switch, on page 8


## Catalyst C2960XR-48TD-I Switch

The Catalyst C2960XR-48TD-I has 48 10/100/1000 ports.

***Table 3: Catalyst 2960-XR Switch Information***

| Ultra M Model(s) | Quantity | Software Version | Firmware Version |
|---|---|---|---|
| Ultra M XS Single VNF | 2 | IOS 15.2.(2) E5 | Boot Loader: 15.2(3r)E1 |
| Ultra M XS Multi-VNF | 1 per rack | IOS 15.2.(2) E5 | Boot Loader: 15.2(3r)E1 |

## Catalyst 3850-48T-S Switch

The Catalyst 3850 48T-S has 48 10/100/1000 ports.

***Table 4: Catalyst 3850-48T-S Switch Information***

| Ultra M Models | Quantity | Software Version | Firmware Version |
|---|---|---|---|
| Ultra M XS Single VNF | 2 | IOS: 03.06.06E | Boot Loader: 3.58 |
| Ultra M XS Multi-VNF | 1 per Rack | IOS: 03.06.06E | Boot Loader: 3.58 |

# Cisco Nexus Switches

Cisco Nexus Switches serve as top-of-rack (TOR) leaf and end-of-rack (EOR) spine switches provide out-of-band (OOB) network connectivity between Ultra M components. Two switch models are used for the various Ultra M models:

- Nexus 93180-YC-EX, on page 8
- Nexus 9236C , on page 8

## Nexus 93180-YC-EX

Nexus 93180 switches serve as network leafs within the Ultra M solution. Each switch has 48 10/25-Gbps Small Form Pluggable Plus (SFP+) ports and 6 40/100-Gbps Quad SFP+ (QSFP+) uplink ports.

***Table 5: Nexus 93180-YC-EX***

| Ultra M Model(s) | Quantity | Software Version | Firmware Version |
|---|---|---|---|
| Ultra M XS Single VNF | 2 | NX-OS: 7.0(3)I5(2) | BIOS: 7.59 |
| Ultra M XS Multi-VNF | 2 per Rack | NX-OS: 7.0(3)I5(2) | BIOS: 7.59 |

## Nexus 9236C

Nexus 9236 switches serve as network spines within the Ultra M solution. Each switch provides 36 10/25/40/50/100 Gbps ports.

*Table 6: Nexus 9236C*

| Ultra M Model(s) | Quantity | Software Version | Firmware Version |
|---|---|---|---|
| Ultra M XS Single VNF | 2 | NX-OS: 7.0(3)I5(2) | BIOS: 7.59 |
| Ultra M XS Multi-VNF | 2 | NX-OS: 7.0(3)I5(2) | BIOS: 7.59 |

# UCS C-Series Servers

Cisco UCS C240 M4S SFF servers host the functions and virtual machines (VMs) required by Ultra M.

## Server Functions and Quantities

Server functions and quantity differ depending on the Ultra M model you are deploying:

- **Ultra M Manager Node**: Required only for Ultra M models based on the Hyper-Converged architecture, this server hosts the following:
    - AutoIT HA VMs
    - AutoDeploy HA VMs
    - OSP-D VM
- **OpenStack Controller Nodes**: These servers host the high availability (HA) cluster that serves as the VIM within the Ultra M solution. In addition, they facilitate the Ceph storage monitor function required by the Ceph Storage Nodes and/or OSD Compute Nodes.
- **OSD Compute Nodes**: Required only for Hyper-converged Ultra M models, these servers provide Ceph storage functionality in addition to hosting VMs for the following:
    - AutoVNF HA VMs
    - Elastic Services Controller (ESC) Virtual Network Function Manager (VNFM) active and standby VMs
    - Ultra Element Manager (UEM) VM HA cluster
    - Ultra Service Platform (USP) Control Function (CF) active and standby VMs
- 

*Table 7: Ultra M Server Quantities by Model and Function*

| Ultra M Model(s) | Server Quantity (max) | Ultra M Manager Node | Controller Nodes | OSD Compute Nodes | Compute Nodes (max) | Additional Specifications |
|---|---|---|---|---|---|---|
| Ultra M XS Single VNF | 15 | 1 | 3 | 3 | 8 | Based on node type as described in Table 8: Hyper-Converged Ultra M Single and Multi-VNF UCS C240 Server Specifications by Node Type, on page 13. |
| Ultra M XS Multi-VNF | 45 | 1 | 3 | 3 | 38* | Based on node type as described in Table 8: Hyper-Converged Ultra M Single and Multi-VNF UCS C240 Server Specifications by Node Type, on page 13. |
| * Supports a maximum of 4 VNFs – 8 for the first VNF, 10 for each subsequent VNF. | | | | | | |

# VM Deployment per Node Type

*Figure 3: VM Distribution on Server Nodes for Hyper-converged Ultra M Single VNF Models*

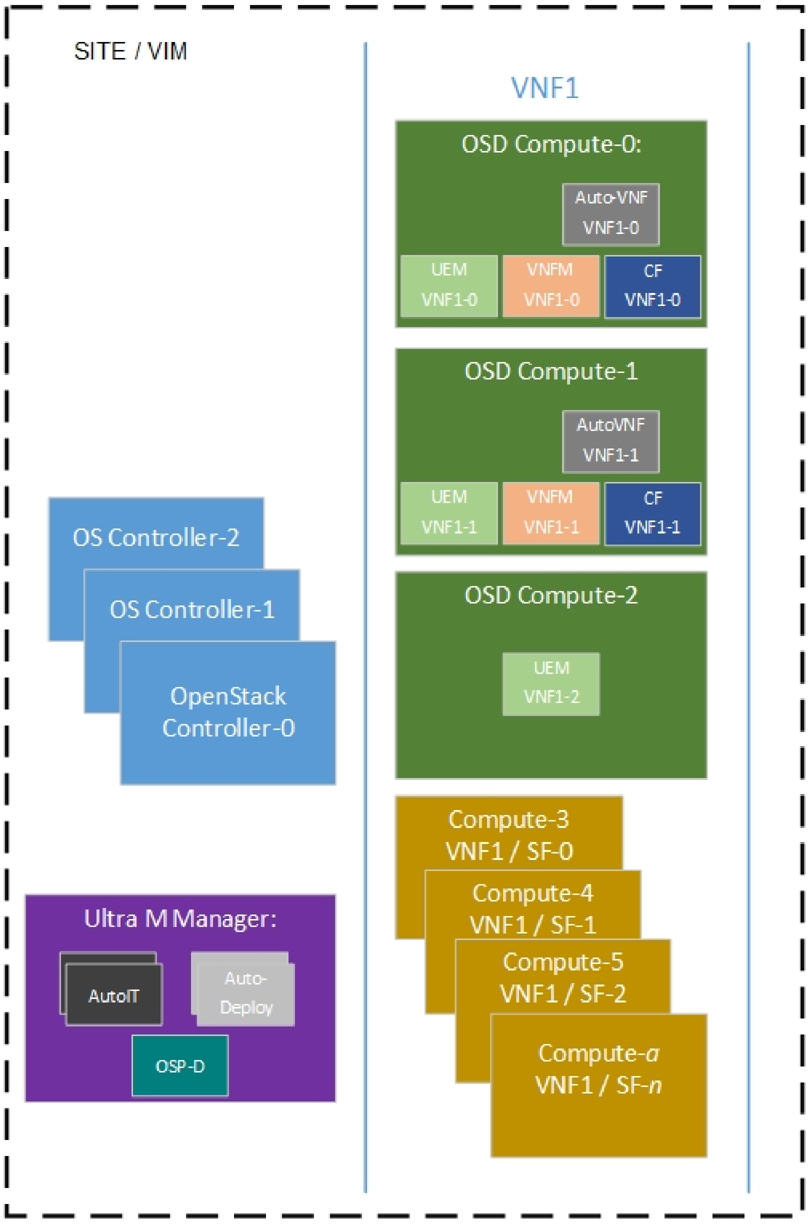
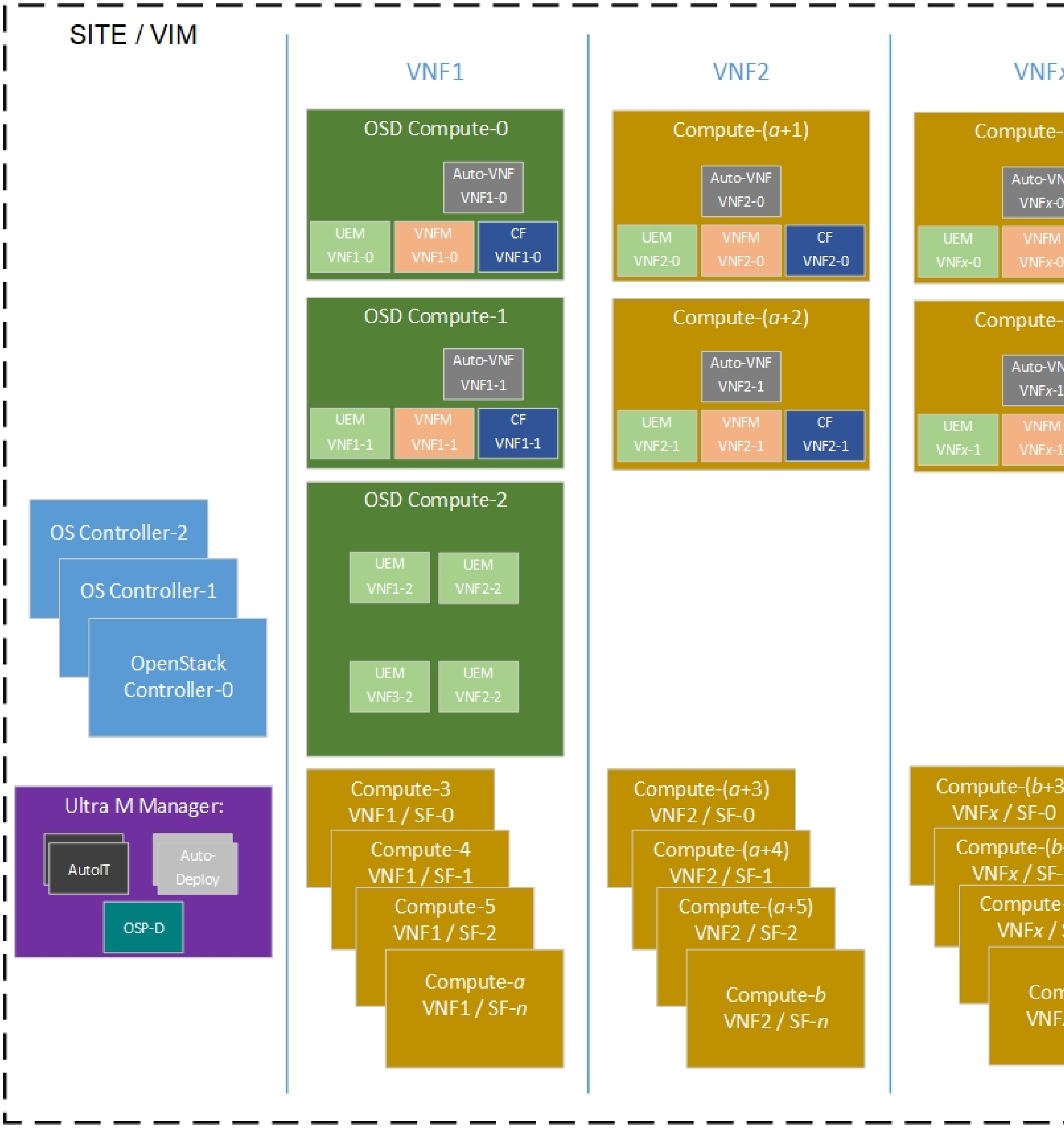



*Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models*


SITE / VIM
VNF1
VNF2
OSD Compute-0
Auto-VNF VNF1-0
UEM VNF1-0
VNFM VNF1-0
CF VNF1-0
OSD Compute-1
Auto-VNF VNF1-1
UEM VNF1-1
VNFM VNF1-1
CF VNF1-1
OSD Compute-2
UEM VNF1-2
UEM VNF2-2
UEM VNF3-2
UEM VNF2-2
Compute-(a+1)
Auto-VNF VNF2-0
UEM VNF2-0
VNFM VNF2-0
CF VNF2-0
Compute-(a+2)
Auto-VNF VNF2-1
UEM VNF2-1
VNFM VNF2-1
CF VNF2-1
UEM VNFx-0
UEM VNFx-1
OS Controller-2
OS Controller-1
OpenStack Controller-0
Ultra M Manager:
AutoIT
Auto-Deploy
OSP-D
Compute-3 VNF1 / SF-0
Compute-4 VNF1 / SF-1
Compute-5 VNF1 / SF-2
Compute-a VNF1 / SF-n
Compute-(a+3) VNF2 / SF-0
Compute-(a+4) VNF2 / SF-1
Compute-(a+5) VNF2 / SF-2
Compute-b VNF2 / SF-n

# Server Configurations

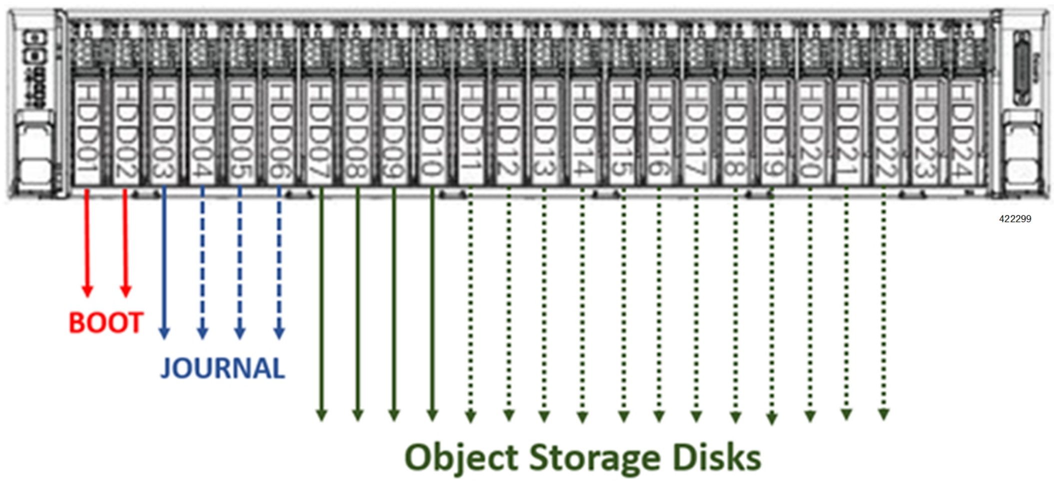
***Table 8: Hyper-Converged Ultra M Single and Multi-VNF UCS C240 Server Specifications by Node Type***

| Node Type | CPU | RAM | Storage | Software Version | Firmware Version |
|---|---|---|---|---|---|
| Ultra M Manager Node* | 2x 2.60 GHz | 4x 32GB DDR4-2400-MHz RDIMM/PC4 | 2x 1.2 TB 12G SAS HDD | MLOM: 4.1(3a) | CIMC: 3.0(3e)<br>System BIOS: C240M4.3.0.3c.0.0831170228 |
| Controller | 2x 2.60 GHz | 4x 32GB DDR4-2400-MHz RDIMM/PC4 | 2x 1.2 TB 12G SAS HDD | MLOM: 4.1(3a) | CIMC: 3.0(3e)<br>System BIOS: C240M4.3.0.3c.0.0831170228 |
| Compute | 2x 2.60 GHz | 8x 32GB DDR4-2400-MHz RDIMM/PC4 | 2x 1.2 TB 12G SAS HDD | MLOM: 4.1(3a) | CIMC: 3.0(3e)<br>System BIOS: C240M4.3.0.3c.0.0831170228 |
| OSD Compute | 2x 2.60 GHz | 8x 32GB DDR4-2400-MHz RDIMM/PC4 | 4x 1.2 TB 12G SAS HDD<br>2x 300G 12G SAS HDD HDD<br>1x 480G 6G SAS SATA SSD | MLOM: 4.1(3a) | CIMC: 3.0(3e)<br>System BIOS: C240M4.3.0.3c.0.0831170228 |
| * OSP-D is deployed as a VM on the Ultra M Manager Node for Hyper-Converged Ultra M model(s). | | | | | |

# Storage

Figure 5: UCS C240 Front-Plane, on page 14 displays the storage disk layout for the UCS C240 series servers used in the Ultra M solution.

***Figure 5: UCS C240 Front-Plane***




**NOTES**:

- The Boot disks contain the operating system (OS) image with which to boot the server.
- The Journal disks contain the Ceph journal file(s) used to repair any inconsistencies that may occur in the Object Storage Disks.
- The Object Storage Disks store object data for USP-based VNFs.
- Ensure that the HDD and SSD used for the Boot Disk, Journal Disk, and object storage devices (OSDs) are available as per the Ultra M BoM and installed in the appropriate slots as identified in Table 9: UCS C240 M4S SFF Storage Specifications by Node Type, on page 14.

***Table 9: UCS C240 M4S SFF Storage Specifications by Node Type***

| | |
|---|---|
| Ultra M Manager Node and Staging Server: | 2 x 1.2 TB HDD – For Boot OS configured as Virtual Drive in RAID1 – placed on Slots 1 & 2 |
| Controllers, Computes: | 2 x 1.2 TB HDD – For Boot OS configured as Virtual Drive in RAID1 – placed on Slots 1 & 2 |

| | |
|---|---|
| OSD Computes: | 2 x 300 GB HDD – For Boot OS configured as Virtual Drive in RAID1 – placed on Slots 1 & 2<br>1 x 480 GB SSD – For Journal Disk as Virtual Drive in RAID0 – Slot 3<br>(Reserve for SSD Slot 3,4,5,6 future scaling needs)<br>4 x 1.2 TB HDD – For OSD's configured as Virtual Drive in RAID0 each – Slot 7,8,9,10<br>(Reserve for OSD 7,8,9,10….,24) |

- Ensure that the RAIDs are sized such that:

  **Boot Disks < Journal Disk(s) < OSDs**
- Ensure that FlexFlash is disabled on each UCS-C240 M4 (default Factory).
- Ensure that all nodes are in *Unconfigured Good* state under **Cisco SAS RAID Controllers** (factory default).

CHAPTER 3

# Software Specifications

*Table 10: Required Software*

| Software | Value/Description |
|---|---|
| Operating System | Red Hat Enterprise Linux 7.3 |
| Hypervisor | Qemu (KVM) |
| VIM | **Hyper-converged Ultra M Single and Multi-VNF Models:**<br>Red Hat OpenStack Platform 10 (OSP 10 - Newton) |
| VNF | 21.6 |
| VNFM | ESC 3.1.0.145 |
| UEM | UEM 6.0 |
| USP | USP 6.0 |

CHAPTER 4

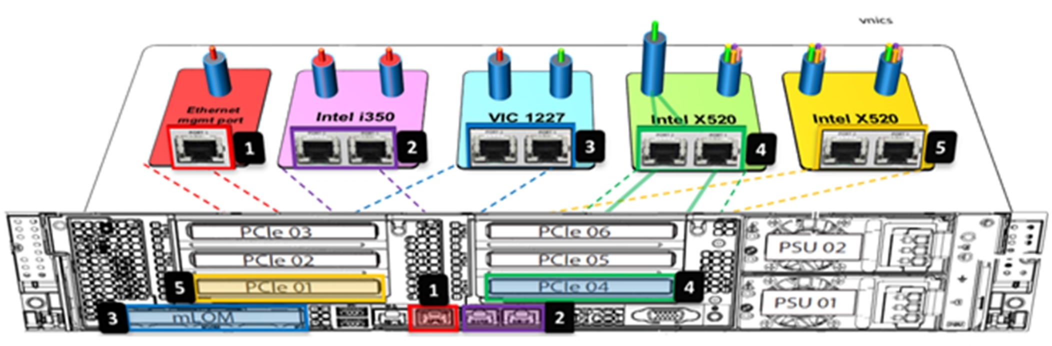
# Networking Overview

This section provides information on Ultra M networking requirements and considerations.



# UCS-C240 Network Interfaces

***Figure 6: UCS-C240 Back-Plane***



<table>
<tr><th>Number</th><th>Designation</th><th>Description</th><th>Applicable Node Types</th></tr>
<tr><td>1</td><td>CIMC/IPMI/M</td><td>The server's Management network interface used for accessing the UCS Cisco Integrated Management Controller (CIMC) application, performing Intelligent Platform Management Interface (IPMI) operations.</td><td>All</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Intel Onboard</td><td>Port 1: VIM Orchestration (Undercloud) Provisioning network interface.</td><td>All</td></tr>
<tr><td>Port 2: External network interface for Internet access. It must also be routable to External floating IP addresses on other nodes.</td><td>Ultra M Manager Node<br>Staging Server</td></tr>
</table>

| Number | Designation | Description | Applicable Node Types |
|---|---|---|---|
| 3 | Modular LAN on Motherboard (mLOM) | VIM networking interfaces used for: | |
| | | • External floating IP network. | Controller |
| | | • Internal API network | Controller |
| | | • Storage network | Controller<br>Compute<br>OSD Compute<br>Ceph |
| | | • Storage Management network | Controller<br>Compute<br>OSD Compute<br>Ceph |
| | | • Tenant network (virtio only – VIM provisioning, VNF Management, and VNF Orchestration) | Controller<br>Compute<br>OSD Compute |
| 4 | PCIe 4 | Port 1: With NIC bonding enabled, this port provides the active Service network interfaces for VNF ingress and egress connections. | Compute |
| | | Port 2: With NIC bonding enabled, this port provides the standby *Di-internal* network interface for inter-VNF component communication. | Compute<br>OSD Compute |

| Number | Designation | Description | Applicable Node Types |
|---|---|---|---|
| 5 | PCIe 1 | Port 1: With NIC bonding enabled, this port provides the active *Di-internal* network interface for inter-VNF component communication. | Compute<br>OSD Compute |
| | | Port 2: With NIC bonding enabled, this port provides the standby Service network interfaces for VNF ingress and egress connections. | Compute |

# VIM Network Topology

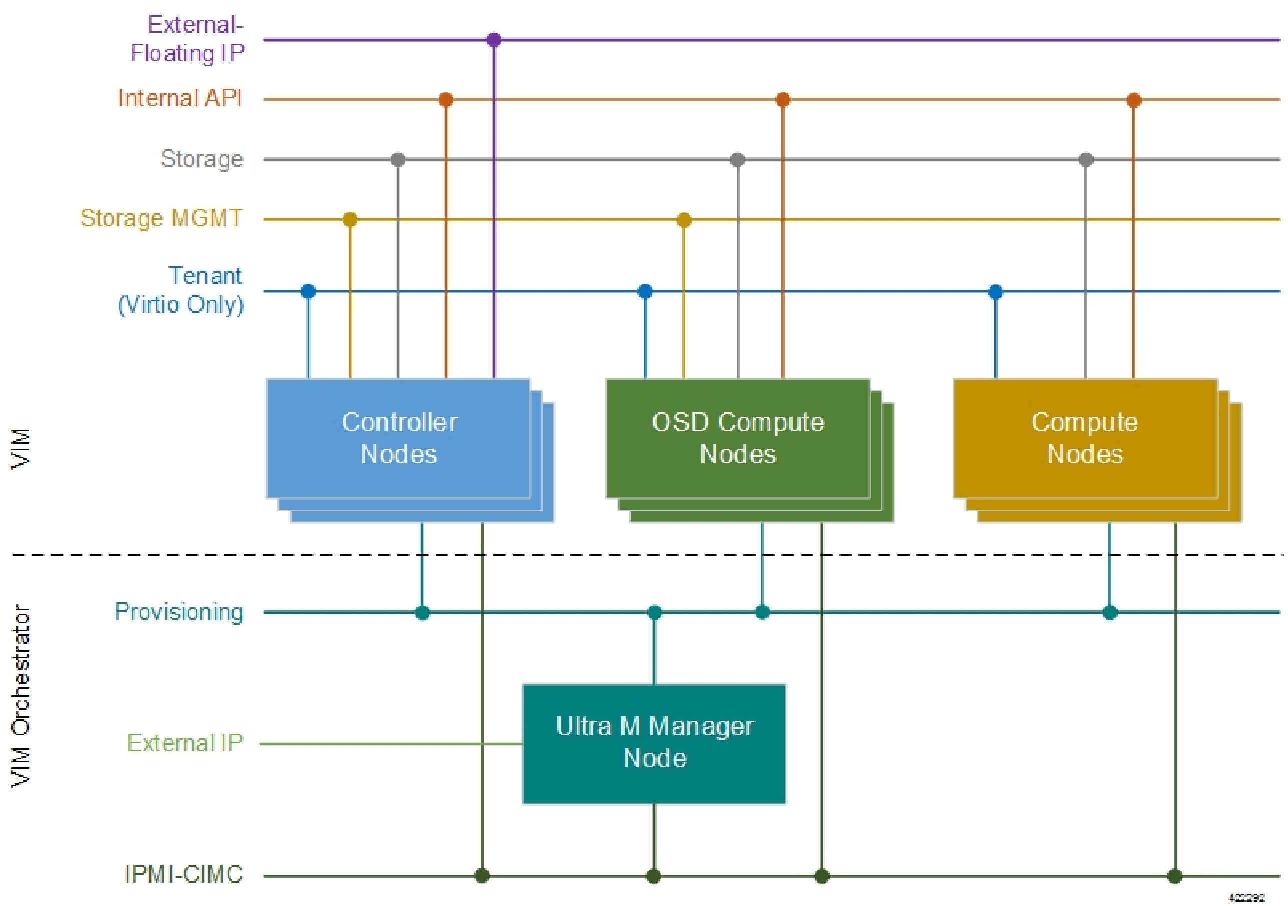
Ultra M's VIM is based on the OpenStack project TripleO ("OpenStack-On-OpenStack") which is the core of the OpenStack Platform Director (OSP-D). TripleO allows OpenStack components to install a fully operational OpenStack environment.

Two cloud concepts are introduced through TripleO:

- **VIM Orchestrator (Undercloud)**: The VIM Orchestrator is used to bring up and manage the VIM. Though OSP-D and Undercloud are sometimes referred to synonymously, the OSP-D bootstraps the Undercloud deployment and provides the underlying components (e.g. Ironic, Nova, Glance, Neutron, etc.) leveraged by the Undercloud to deploy the VIM. Within the Ultra M Solution, OSP-D and the Undercloud are hosted on the same server.

- **VIM (Overcloud)**: The VIM consists of the compute, controller, and storage nodes on which the VNFs are deployed.

***Figure 7: Hyper-converged Ultra M Single and Multi-VNF Model OpenStack VIM Network Topology***




Some considerations for VIM Orchestrator and VIM deployment are as follows:

- External network access (e.g. Internet access) can be configured in one of the following ways:
  - Across all node types: A single subnet is configured on the Controller HA, VIP address, floating IP addresses and OSP-D/Staging server's external interface provided that this network is data-center routable as well as it is able to reach the internet.
  - Limited to OSP-D: The *External IP* network is used by Controllers for HA and Horizon dashboard as well as later on for Tenant Floating IP address requirements. This network must be data-center routable. In addition, the *External IP* network is used only by OSP-D/Staging Server node's external interface that has a single IP address. The *External IP* network must be lab/data-center routable must also have internet access to Red Hat cloud. It is used by OSP-D/Staging Server for subscription purposes and also acts as an external gateway for all controllers, computes and Ceph-storage nodes.
- IPMI must be enabled on all nodes.
- Two networks are needed to deploy the VIM Orchestrator:

    - IPMI/CIMC Network
    - Provisioning Network

- The OSP-D/Staging Server must have reachability to both IPMI/CIMC and Provisioning Networks. (VIM Orchestrator networks need to be routable between each other or have to be in one subnet.)
- DHCP-based IP address assignment for Introspection PXE from Provisioning Network (Range A)
- DHCP based IP address assignment for VIM PXE from Provisioning Network (Range B) must be separate from Introspection.
- The Ultra M Manager Node/Staging Server acts as a gateway for Controller, Ceph and Computes. Therefore, the external interface of this node/server needs to be able to access the Internet. In addition, this interface needs to be routable with the Data-center network. This allows the External interface IP-address of the Ultra M Manager Node/Staging Server to reach Data-center routable Floating IP addresses as well as the VIP addresses of Controllers in HA Mode.
- Prior to assigning floating and virtual IP addresses, make sure that they are not already allocated through OpenStack. If the addresses are already allocated, then they must be freed up for use or you must assign a new IP address that is available in the VIM.
- Multiple VLANs are required in order to deploy OpenStack VIM:
    - 1 for the Management and Provisioning networks interconnecting all the nodes regardless of type
    - 1 for the Staging Server/OSP-D Node external network
    - 1 for Compute, Controller, and Ceph Storage or OSD Compute Nodes
    - 1 for Management network interconnecting the Leafs and Spines
- Login to individual Compute nodes will be from OSP-D/Staging Server using heat user login credentials.

  The OSP-D/Staging Server acts as a "jump server" where the br-ctlplane interface address is used to login to the Controller, Ceph or OSD Computes, and Computes post VIM deployment using heat-admin credentials.

Layer 1 networking guidelines for the VIM network are provided in Layer 1 Leaf and Spine Topology, on page 27. In addition, a template is provided in Network Definitions (Layer 2 and 3), on page 83 to assist you with your Layer 2 and Layer 3 network planning.

# Openstack Tenant Networking

The interfaces used by the VNF are based on the PCIe architecture. Single root input/output virtualization (SR-IOV) is used on these interfaces to allow multiple VMs on a single server node to use the same network interface as shown in Figure 8: Physical NIC to Bridge Mappings, on page 25. SR-IOV Networking is network type *Flat* under OpenStack configuration. NIC Bonding is used to ensure port level redundancy for PCIe Cards involved in SR-IOV Tenant Networks as shown in Figure 9: NIC Bonding, on page 25.

***Figure 8: Physical NIC to Bridge Mappings***

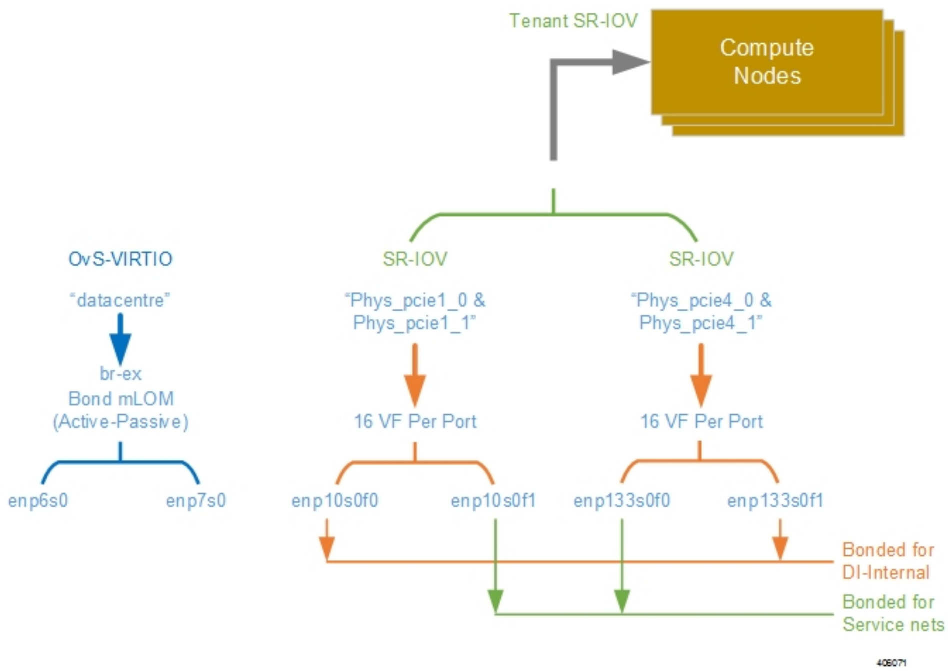
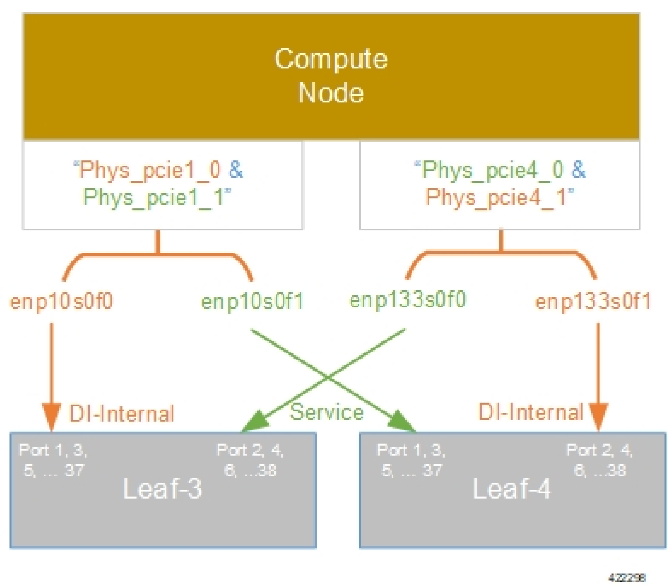


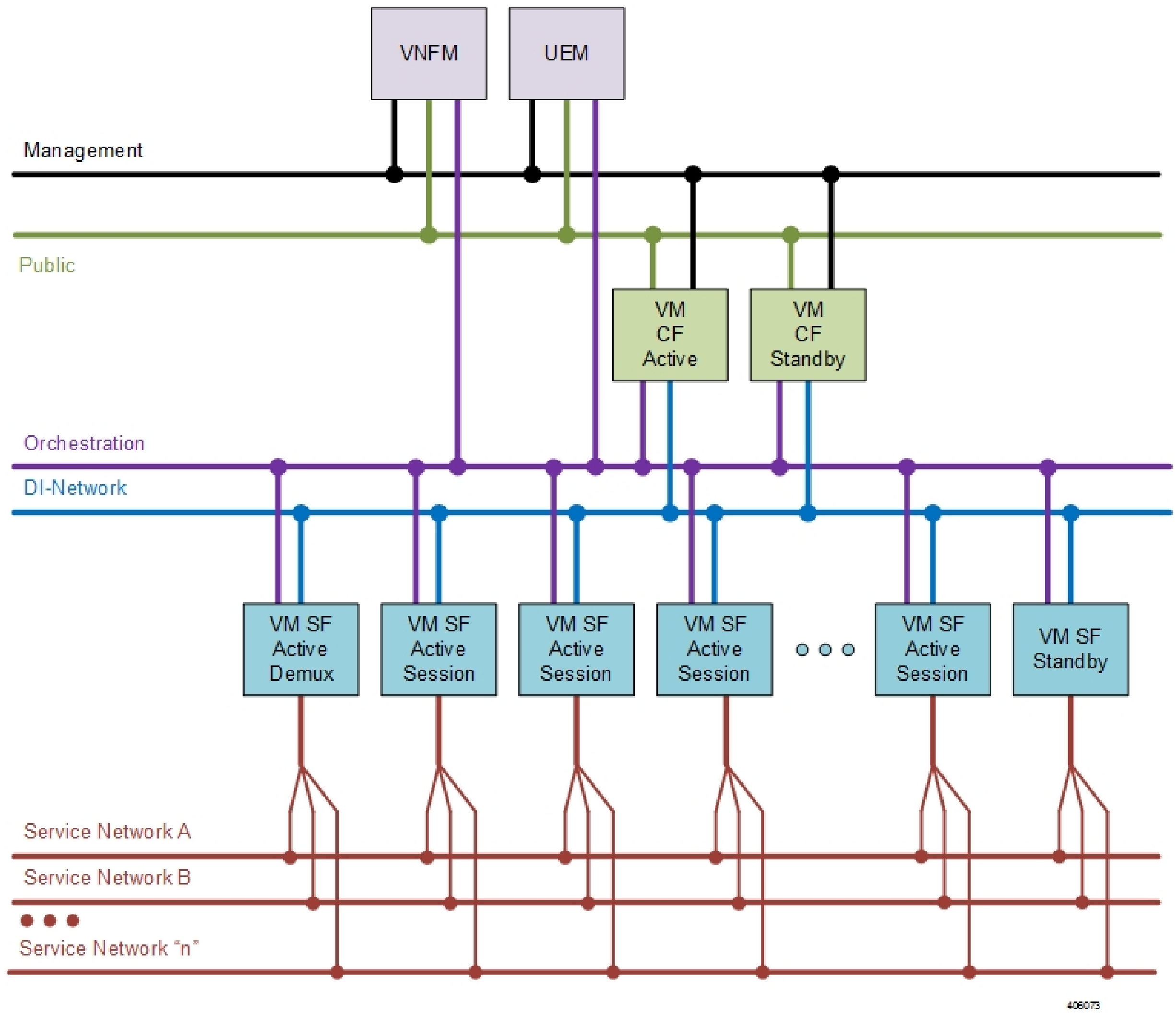

***Figure 9: NIC Bonding***



# VNF Tenant Networks

While specific VNF network requirements are described in the documentation corresponding to the VNF, Figure 10: Typical USP-based VNF Networks, on page 26 displays the types of networks typically required by USP-based VNFs.

***Figure 10: Typical USP-based VNF Networks***




The USP-based VNF networking requirements and the specific roles are described here:

- **Public**: *External public network*. The router has an external gateway to the public network. All other networks (except DI-Internal and ServiceA-*n*) have an internal gateway pointing to the router. And the router performs secure network address translation (SNAT).
- **DI-Internal**: This is the DI-internal network which serves as a 'backplane' for CF-SF and CF-CF communications. Since this network is internal to the UGP, it does not have a gateway interface to the

router in the OpenStack network topology. A unique DI internal network must be created for each instance of the UGP. The interfaces attached to these networks use performance optimizations.

- **Management**: This is the local management network between the CFs and other management elements like the UEM and VNFM. This network is also used by OSP-D to deploy the VNFM and AutoVNF. To allow external access, an OpenStack floating IP address from the Public network must be associated with the UGP VIP (CF) address.

  You can ensure that the same floating IP address can assigned to the CF, UEM, and VNFM after a VM restart by configuring parameters in the AutoDeploy configuration file or the UWS service delivery configuration file.

  **Note** Prior to assigning floating and virtual IP addresses, make sure that they are not already allocated through OpenStack. If the addresses are already allocated, then they must be freed up for use or you must assign a new IP address that is available in the VIM.

- **Orchestration**: This is the network used for VNF deployment and monitoring. It is used by the VNFM to onboard the USP-based VNF.

- **ServiceA-*n***: These are the service interfaces to the SF. Up to 12 service interfaces can be provisioned for the SF with this release. The interfaces attached to these networks use performance optimizations.

  Layer 1 networking guidelines for the VNF network are provided in Layer 1 Leaf and Spine Topology, on page 27. In addition, a template is provided in Network Definitions (Layer 2 and 3), on page 83 to assist you with your Layer 2 and Layer 3 network planning.

## Supporting Trunking on VNF Service ports

Service ports within USP-based VNFs are configured as trunk ports and traffic is tagged using the VLAN command. In This configuration is supported by trunking to the uplink switch via the *sriovnicswitch mechanism* driver.

This driver supports Flat network types in OpenStack, enabling the guest OS to tag the packets.

Flat networks are untagged networks in OpenStack. Typically, these networks are previously existing infrastructure, where OpenStack guests can be directly applied.

# Layer 1 Leaf and Spine Topology

Ultra M implements a Leaf and Spine network topology. Topology details differ between Ultra M models based on the scale and number of nodes.

**Note** When connecting component network ports, ensure that the destination ports are rated at the same speed as the source port (e.g. connect a 10G port to a 10G port). Additionally, the source and destination ports must support the same physical medium (e.g. Ethernet) for interconnectivity.

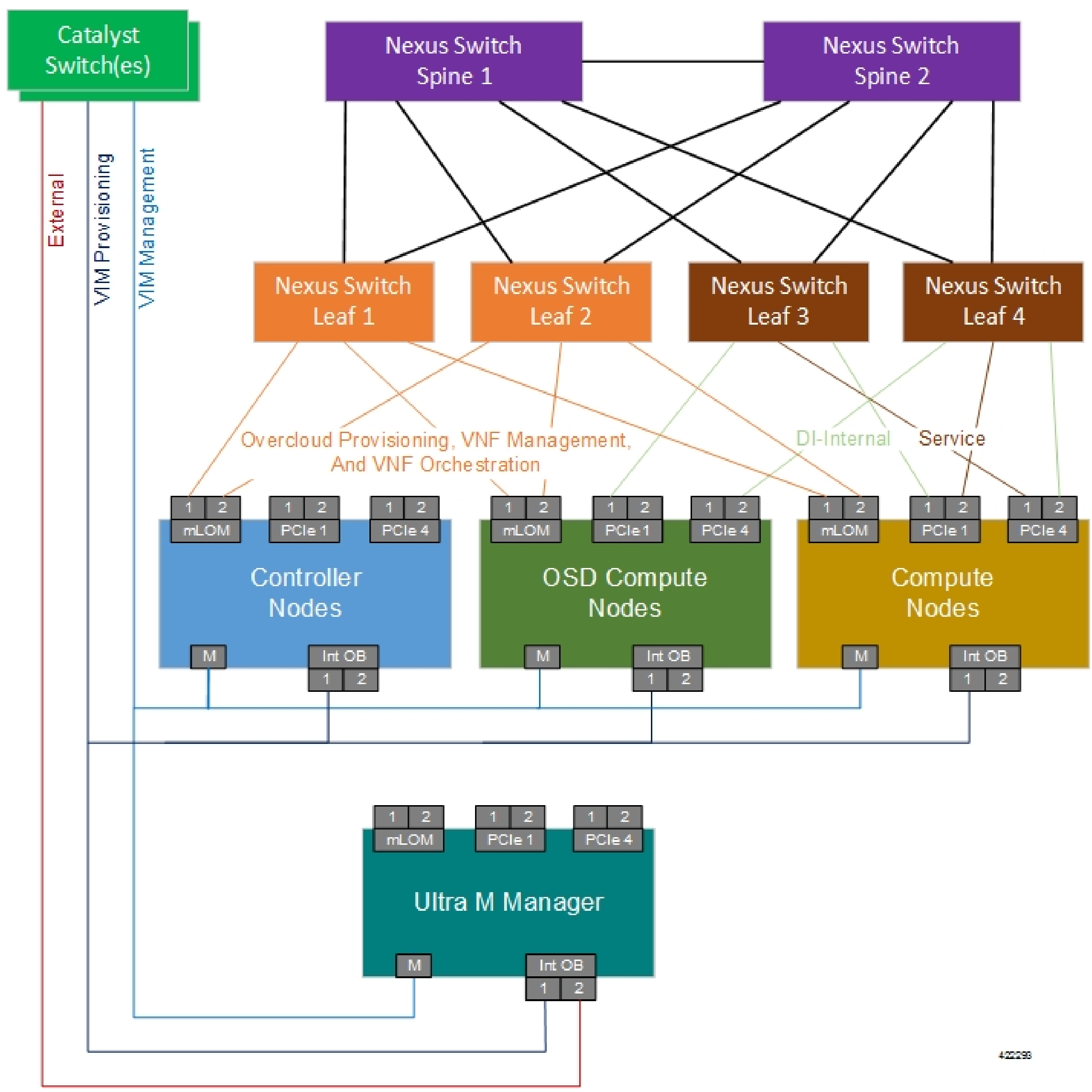
# Hyper-converged Ultra M Single and Multi-VNF Model Network Topology

Figure 11: Hyper-converged Ultra M Single and Multi-VNF Leaf and Spine Topology, on page 28 illustrates the logical leaf and spine topology for the various networks required for the Hyper-converged Ultra M models.

In this figure, two VNFs are supported. (Leafs 1 and 2 pertain to VNF1, Leafs 3 and 4 pertain to VNF 2). If additional VNFs are supported, additional Leafs are required (e.g. Leafs 5 and 6 are needed for VNF 3, Leafs 7 and 8 for VNF4). Each set of additional Leafs would have the same meshed network interconnects with the Spines and with the Controller, OSD Compute, and Compute Nodes.

For single VNF models, Leaf 1 and Leaf 2 facilitate all of the network interconnects from the server nodes and from the Spines.

***Figure 11: Hyper-converged Ultra M Single and Multi-VNF Leaf and Spine Topology***



As identified in Cisco Nexus Switches, on page 8, the number of leaf and spine switches differ between the Ultra M models. Similarly, the specific leaf and spine ports used also depend on the Ultra M solution model being deployed. That said, general guidelines for interconnecting the leaf and spine switches in an Ultra M XS multi-VNF deployment are provided in Table 11: Catalyst Management Switch 1 (Rack 1) Port Interconnects, on page 29 through Table 20: Spine 2 Port Interconnect Guidelines, on page 41. Using the information in these tables, you can make appropriate adjustments to your network topology based on your deployment scenario (e.g. number of VNFs and number of Compute Nodes).

*Table 11: Catalyst Management Switch 1 (Rack 1) Port Interconnects*

| From Switch Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 1, 2, 11 | OSD Compute Nodes | Management | CIMC | 3 non-sequential ports - 1 per OSD Compute Node |
| 3-10 | Compute Nodes | Management | CIMC | 6 sequential ports - 1 per Compute Node |
| 12 | Ultra M Manager Node | Management | CIMC | Management Switch 1 only |
| 13 | Controller 0 | Management | CIMC | |
| 21, 22, 31 | OSD Compute Nodes | Provisioning | Mgmt | 3 non-sequential ports - 1 per OSD Compute Node |
| 23-30 | Compute Nodes | Provisioning | Mgmt | 6 sequential ports - 1 per Compute Node |
| 32-33 | Ultra M Manager Node | Provisioning | Mgmt | 2 sequential ports |
| 34 | Controller 0 | Management | CIMC | |
| 47 | Leaf 1 | Management | 48 | Switch port 47 connects with Leaf 1 port 48 |
| 48 | Leaf 2 | Management | 48 | Switch port 48 connects with Leaf 2 port 48 |

*Table 12: Catalyst Management Switch 2 (Rack 2) Port Interconnects*

| From Switch Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 1-10 | Compute Nodes | Management | CIMC | 10 sequential ports - 1 per Compute Node |

| From Switch Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 14 | Controller 1 | Management | CIMC | |
| 15 | Controller 2 | Management | CIMC | |
| 21-30 | Compute Nodes | Provisioning | Mgmt | 10 sequential ports - 1 per Compute Node |
| 35 | Controller 1 | Provisioning | Mgmt | |
| 36 | Controller 2 | Provisioning | Mgmt | |
| 47 | Leaf 3 | Management | 48 | Switch port 47 connects with Leaf 3 port 48 |
| 48 | Leaf 4 | Management | 48 | Switch port 48 connects with Leaf 4 port 48 |

***Table 13: Catalyst Management Switch 3 (Rack 3) Port Interconnects***

| From Switch Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 1-10 | Compute Nodes | Management | CIMC | 10 sequential ports - 1 per Compute Node |
| 21-30 | Compute Nodes | Provisioning | Mgmt | 10 sequential ports - 1 per Compute Node |
| 47 | Leaf 5 | Management | 48 | Switch port 47 connects with Leaf 5 port 48 |
| 48 | Leaf 6 | Management | 48 | Switch port 48 connects with Leaf 6 port 48 |

***Table 14: Catalyst Management Switch 4 (Rack 4) Port Interconnects***

| From Switch Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 1-10 | Compute Nodes | Management | CIMC | 10 sequential ports - 1 per Compute Node |

| From Switch Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 21-30 | Compute Nodes | Provisioning | Mgmt | 10 sequential ports - 1 per Compute Node |
| 47 | Leaf 7 | Management | 48 | Switch port 47 connects with Leaf 7 port 48 |
| 48 | Leaf 8 | Management | 48 | Switch port 48 connects with Leaf 8 port 48 |

*Table 15: Leaf 1 and 2 (Rack 1) Port Interconnects**

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| **Leaf 1** | | | | |
| 1, 2, 11 | OSD Compute Nodes | Management & Orchestration (active) | MLOM P1 | 3 non-sequential ports - 1 per OSD Compute Node |
| 12 | Controller 0 Node | Management & Orchestration (active) | MLOM P1 | |
| 17, 18, 27 | OSD Compute Nodes | Di-internal (active) | PCIe01 P1 | 3 non-sequential ports - 1 per OSD Compute Node |
| 3 - 10 (inclusive) | Compute Nodes | Management & Orchestration (active) | MLOM P1 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node |
| 19-26 (inclusive) | Compute Nodes | Di-internal (active) | PCIe01 P1 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node |
| 33-42 (inclusive) | Compute Nodes / OSD Compute Nodes | Service (active) | PCIe04 P1 | Sequential ports based on the number of Compute Nodes and/or OSD Compute Nodes - 1 per OSD Compute Node and/or Compute Node<br>**Note** Though the OSD Compute Nodes do not use the Service Networks, they are provided to ensure compatibility within the OpenStack Overcloud (VIM) deployment. |

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 48 | Catalyst Management Switches | Management | 47 | Leaf 1 connects to Switch 1 |
| 49-50 | Spine 1 | Downlink | 1-2 | Leaf 1 port 49 connects to Spine 1 port 1<br>Leaf 1 port 50 connects to Spine 1 port 2 |
| 51-52 | Spine 2 | Downlink | 3-4 | Leaf 1 port 51 connects to Spine 2 port 3<br>Leaf 1 port 52 connects to Spine 2 port 4 |
| **Leaf 2** | | | | |
| 1, 2, 11 | OSD Compute Nodes | Management & Orchestration (redundant) | MLOM P2 | 3 non-sequential ports - 1 per OSD Compute Node |
| 12 | Controller 0 Node | Management & Orchestration (redundant) | MLOM P2 | |
| 17, 18, 27 | OSD Compute Nodes | Di-internal (redundant) | PCIe04 P2 | 3 non-sequential ports - 1 per OSD Compute Node |
| 3 - 10 (inclusive) | Compute Nodes | Management & Orchestration (redundant) | MLOM P2 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node |
| 19-26 (inclusive) | Compute Nodes | Di-internal (redundant) | PCIe04 P2 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node |
| 33-42 (inclusive) | Compute Nodes / OSD Compute Nodes | Service (redundant) | PCIe01 P2 | Sequential ports based on the number of Compute Nodes and/or OSD Compute Nodes - 1 per OSD Compute Node and/or Compute Node<br>**Note** Though the OSD Compute Nodes do not use the Service Networks, they are provided to ensure compatibility within the OpenStack Overcloud (VIM) deployment. |
| 48 | Catalyst Management Switches | Management | 48 | Leaf 2 connects to Switch 1 |

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | **Device** | **Network** | **Port(s)** | |
| 49-50 | Spine 1 | Downlink | 1-2 | Leaf 2 port 49 connects to Spine 1 port 1<br>Leaf 2 port 50 connects to Spine 1 port 2 |
| 51-52 | Spine 2 | Downlink | 3-4, 7-8,<br>11-12, 15-16 | Leaf 2 port 51 connects to Spine 2 port 3<br>Leaf 2 port 52 connects to Spine 2 port 4 |

*Table 16: Leaf 3 and 4 (Rack 2) Port Interconnects*

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | **Device** | **Network** | **Port(s)** | |
| **Leaf 3** | | | | |
| 1 - 10 (inclusive) | Compute Nodes | Management & Orchestration (active) | MLOM P1 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 1 and 2 are used for the first two Compute Nodes on VNFs other than VNF1 (Rack 1). These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |
| 13-14 (inclusive) | Controller Nodes | Management & Orchestration (active) | MLOM P1 | Leaf 3 port 13 connects to Controller 1 MLOM P1 port<br>Leaf 3 port 14 connects to Controller 1 MLOM P1 port |
| 17-26 (inclusive) | Compute Nodes | Di-internal (active) | PCIe01 P1 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 17 and 18 are used for the first two Compute Nodes on VNFs other than VNF1. These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 33-42 (inclusive) | Compute Nodes | Service (active) | PCIe04 P1 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node |
| 48 | Catalyst Management Switches | Management | 47 | Leaf 3 connects to Switch 2 |
| 49-50 | Spine 1 | Downlink | 5-6 | Leaf 3 port 49 connects to Spine 1 port 5<br>Leaf 3 port 50 connects to Spine 1 port 6 |
| 51-52 | Spine 2 | Downlink | 7-8 | Leaf 3 port 51 connects to Spine 2 port 7<br>Leaf 3 port 52 connects to Spine 2 port 8 |
| **Leaf 4** | | | | |
| 1 - 10 (inclusive) | Compute Nodes | Management & Orchestration (redundant) | MLOM P2 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 1 and 2 are used for the first two Compute Nodes on VNFs other than VNF1. These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |
| 13-14 (inclusive) | Controller Nodes | Management & Orchestration (redundant) | MLOM P2 | Leaf 4 port 13 connects to Controller 1 MLOM P2 port<br>Leaf 4 port 14 connects to Controller 1 MLOM P2 port |
| 17-26 (inclusive) | Compute Nodes | Di-internal (redundant) | PCIe04 P2 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 17 and 18 are used for the first two Compute Nodes on VNFs other than VNF1. These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 33-42 (inclusive) | Compute Nodes | Service (redundant) | PCIe01 P2 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node |
| 48 | Catalyst Management Switches | Management | 48 | Leaf 4 connects to Switch 2 |
| 49-50 | Spine 1 | Downlink | 5-6 | Leaf 4 port 49 connects to Spine 1 port 5<br>Leaf 4 port 50 connects to Spine 1 port 6 |
| 51-52 | Spine 2 | Downlink | 7-8 | Leaf 4 port 51 connects to Spine 2 port 7<br>Leaf 4 port 52 connects to Spine 2 port 8 |

*Table 17: Leaf 5 and 6 (Rack 3) Port Interconnects*

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| **Leaf 5** | | | | |
| 1 - 10 (inclusive) | Compute Nodes | Management & Orchestration (active) | MLOM P1 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 1 and 2 are used for the first two Compute Nodes on VNFs other than VNF1. These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | **Device** | **Network** | **Port(s)** | |
| 17-26 (inclusive) | Compute Nodes | Di-internal (active) | PCIe01 P1 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 17 and 18 are used for the first two Compute Nodes on VNFs other than VNF1. These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |
| 33-42 (inclusive) | Compute Nodes | Service (active) | PCIe04 P1 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node |
| 48 | Catalyst Management Switches | Management | 47 | Leaf 5 connects to Switch 3 |
| 49-50 | Spine 1 | Downlink | 9-10 | Leaf 5 port 49 connects to Spine 1 port 9<br>Leaf 5 port 50 connects to Spine 1 port 10 |
| 51-52 | Spine 2 | Downlink | 3-4, 7-8,<br>11-12, 15-16 | Leaf 5 port 51 connects to Spine 2 port 11<br>Leaf 5 port 52 connects to Spine 2 port 12 |
| **Leaf 6** | | | | |
| 1 - 10 (inclusive) | Compute Nodes | Management & Orchestration (redundant) | MLOM P2 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 1 and 2 are used for the first two Compute Nodes on VNFs other than VNF1. These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 17-26 (inclusive) | Compute Nodes | Di-internal (redundant) | PCIe04 P2 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 17 and 18 are used for the first two Compute Nodes on VNFs other than VNF1. These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |
| 33-42 (inclusive) | Compute Nodes | Service (redundant) | PCIe01 P2 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node |
| 48 | Catalyst Management Switches | Management | 48 | Leaf 6 connects to Switch 3 |
| 49-50 | Spine 1 | Downlink | 9-10 | Leaf 6 port 49 connects to Spine 1 port 9<br>Leaf 6 port 50 connects to Spine 1 port 10 |
| 51-52 | Spine 2 | Downlink | 11-12 | Leaf 6 port 51 connects to Spine 2 port 11<br>Leaf 6 port 52 connects to Spine 2 port 12 |

*Table 18: Leaf 7 and 8 (Rack 4) Port Interconnects*

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| **Leaf 7** | | | | |

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 1 - 10 (inclusive) | Compute Nodes | Management & Orchestration (active) | MLOM P1 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 1 and 2 are used for the first two Compute Nodes on VNFs other than VNF1. These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |
| 17-26 (inclusive) | Compute Nodes | Di-internal (active) | PCIe01 P1 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 17 and 18 are used for the first two Compute Nodes on VNFs other than VNF1. These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |
| 33-42 (inclusive) | Compute Nodes | Service (active) | PCIe04 P1 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node |
| 48 | Catalyst Management Switches | Management | 47 | Leaf 7 connects to Switch 4 |
| 49-50 | Spine 1 | Downlink | 13-14 | Leaf 7 port 49 connects to Spine 1 port 13<br>Leaf 7 port 50 connects to Spine 1 port 14 |
| 51-52 | Spine 2 | Downlink | 15-16 | Leaf 7 port 51 connects to Spine 2 port 15<br>Leaf 7 port 52 connects to Spine 2 port 16 |
| **Leaf 8** | | | | |

| From Leaf Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 1 - 10 (inclusive) | Compute Nodes | Management & Orchestration (redundant) | MLOM P2 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 1 and 2 are used for the first two Compute Nodes on VNFs other than VNF1. These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |
| 17-26 (inclusive) | Compute Nodes | Di-internal (redundant) | PCIe04 P2 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node<br>**Important** Leaf Ports 17 and 18 are used for the first two Compute Nodes on VNFs other than VNF1. These are used to host management-related VMs as shown in Figure 4: VM Distribution on Server Nodes for Hyper-converged Ultra M Multi-VNF Models, on page 11. |
| 33-42 (inclusive) | Compute Nodes | Service (redundant) | PCIe01 P2 | Sequential ports based on the number of Compute Nodes - 1 per Compute Node |
| 48 | Catalyst Management Switches | Management | 48 | Leaf 8 connects to Switch 4 |
| 49-50 | Spine 1 | Downlink | 13-14 | Leaf 8 port 49 connects to Spine 1 port 13<br>Leaf 8 port 50 connects to Spine 1 port 14 |
| 51-52 | Spine 2 | Downlink | 15-16 | Leaf 8 port 51 connects to Spine 2 port 15<br>Leaf 8 port 52 connects to Spine 2 port 16 |

*Table 19: Spine 1 Port Interconnect Guidelines*

| From Spine Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 1-2,<br>5-6,<br>9-10,<br>13-14 | Leaf 1, 3, 5, 7 | Downlink | 49-50 | Spine 1 ports 1 and 2 connect to Leaf 1 ports 49 and 50<br>Spine 1 ports 5 and 6 connect to Leaf 3 ports 49 and 50<br>Spine 1 ports 9 and 10 connect to Leaf 5 ports 49 and 50<br>Spine 1 ports 13 and 14 connect to Leaf 7 ports 49 and 50 |
| 3-4,<br>7-8,<br>11-12,<br>15-16 | Leaf 2, 4, 6, 8 | Downlink | 49-50 | Spine 1 ports 3 and 4 connect to Leaf 2 ports 49 and 50<br>Spine 1 ports 7 and 8 connect to Leaf 4 ports 49 and 50<br>Spine 1 ports 11 and 12 connect to Leaf 6 ports 49 and 50<br>Spine 1 ports 15 and 16 connect to Leaf 8 ports 49 and 50 |
| 29-30,<br>31, 32,<br>33-34 | Spine 2 | Interlink | 29-30,<br>31, 32,<br>33-34 | Spine 1 ports 29-30 connect to Spine 2 ports 29-30<br>Spine 1 port 31 connects to Spine 2 port 31<br>Spine 1 port 32 connects to Spine 2 port 32<br>Spine 1 ports 33-34 connect to Spine 2 ports 33-34 |
| 21-22,<br>23-24,<br>25-26 | Router | Uplink | - | |

*Table 20: Spine 2 Port Interconnect Guidelines*

| From Spine Port(s) | To | | | Notes |
|---|---|---|---|---|
| | Device | Network | Port(s) | |
| 1-2,<br>5-6,<br>9-10,<br>13-14 | Leaf 1, 3, 5, 7 | Downlink | 51-52 | Spine 1 ports 1 and 2 connect to Leaf 1 ports 51 and 52<br>Spine 1 ports 5 and 6 connect to Leaf 3 ports 51 and 52<br>Spine 1 ports 9 and 10 connect to Leaf 5 ports 51 and 52<br>Spine 1 ports 13 and 14 connect to Leaf 7 ports 51 and 52 |
| 3-4,<br>7-8,<br>11-12,<br>15-16 | Leaf 2, 4, 6, 8 | Downlink | 51-52 | Spine 1 ports 3 and 4 connect to Leaf 2 ports 51 and 52<br>Spine 1 ports 7 and 8 connect to Leaf 4 ports 51 and 52<br>Spine 1 ports 11 and 12 connect to Leaf 6 ports 51 and 52<br>Spine 1 ports 15 and 16 connect to Leaf 8 ports 51 and 52 |
| 29-30,<br>31, 32,<br>33-34 | Spine 1 | Interconnect | 29-30,<br>31, 32,<br>33-34 | Spine 2 ports 29-30 connect to Spine 1 ports 29-30<br>Spine 2 port 31 connects to Spine 1 port 31<br>Spine 2 port 32 connects to Spine 1 port 32<br>Spine 2 ports 33-34 connect to Spine 1 ports 33-34 |
| 21-22,<br>23-24,<br>25-26 | Router | Uplink | - | |

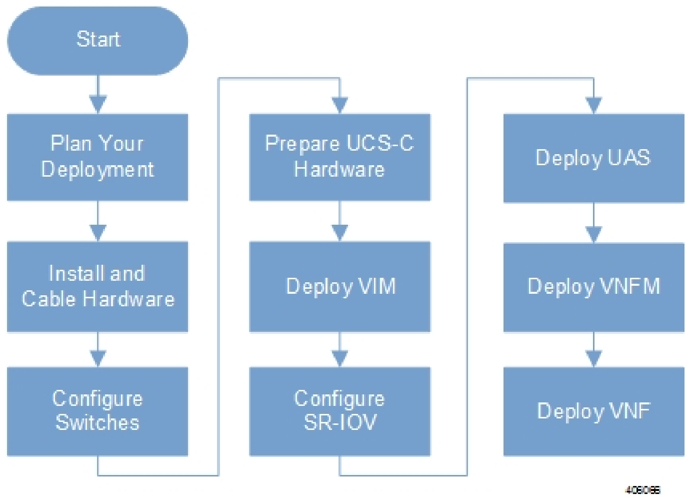
CHAPTER 5

# Deploying the Ultra M Solution

Ultra M is a multi-product solution. Detailed instructions for installing each of these products is beyond the scope of this document. Instead, the sections that follow identify the specific, non-default parameters that must be configured through the installation and deployment of those products in order to deploy the entire solution.

# Deployment Workflow

*Figure 12: Ultra M Deployment Workflow*




# Plan Your Deployment

Before deploying the Ultra M solution, it is very important to develop and plan your deployment.

## Network Planning

Networking Overview, on page 19 provides a general overview and identifies basic requirements for networking the Ultra M solution.

With this background, use the tables in Network Definitions (Layer 2 and 3), on page 83 to help plan the details of your network configuration.

# Install and Cable the Hardware

This section describes the procedure to install all the components included in the Ultra M Solution.

## Related Documentation

To ensure hardware components of the Ultra M solution are installed properly, refer to the installation guides for the respective hardware components.

- **Catalyst 2960-XR Switch** — http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960xr/hardware/installation/guide/b_c2960xr_hig.html

- **Catalyst 3850 48T-S Switch** — http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3850/hardware/installation/guide/b_c3850_hig.html
- **Nexus 93180-YC 48 Port** — http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/hw/n93180ycex_hig/guide/b_n93180ycex_nxos_mode_hardware_install_guide.html
- **Nexus 9236C 36 Port** — http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/hw/n9236c_hig/guide/b_c9236c_nxos_mode_hardware_install_guide.html
- **UCS C240 M4SX Server** — http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c/hw/C240M4/install/C240M4.html

# Rack Layout

## Hyper-converged Ultra M XS Single VNF Deployment

Table 21: Hyper-converged Ultra M XS Single VNF Deployment Rack Layout, on page 45 provides details for the recommended rack layout for the Hyper-converged Ultra M XS Single VNF deployment model.

***Table 21: Hyper-converged Ultra M XS Single VNF Deployment Rack Layout***

| | **Rack #1** | **Rack #2** |
|---|---|---|
| RU-1 | Empty | Empty |
| RU-2 | Spine EOR Switch A: Nexus 9236C | Spine EOR Switch B: Nexus 9236C |
| RU-3 | Empty | Empty |
| RU-4 | VNF Mgmt Switch: Catalyst C3850-48T-S OR C2960XR-48TD | Empty |
| RU-5 | VNF Leaf TOR Switch A: Nexus 93180YC-EX | Empty |
| RU-6 | VNF Leaf TOR Switch B: Nexus 93180YC-EX | Empty |
| RU-7/8 | Ultra UEM 1A: UCS C240 M4 SFF | Empty |
| RU-9/10 | Ultra UEM 1B: UCS C240 M4 SFF | Empty |
| RU-11/12 | Empty | Empty |
| RU-13/14 | Demux SF: UCS C240 M4 SFF | Empty |
| RU-15/16 | Standby SF: UCS C240 M4 SFF | Empty |
| RU-17/18 | Active SF 1: UCS C240 M4 SFF | Empty |

| | Rack #1 | Rack #2 |
|---|---|---|
| RU-19/20 | Active SF 2: UCS C240 M4 SFF | Empty |
| RU-21/22 | Active SF 3: UCS C240 M4 SFF | Empty |
| RU-23/24 | Active SF 4: UCS C240 M4 SFF | Empty |
| RU-25/26 | Active SF 5: UCS C240 M4 SFF | Empty |
| RU-27/28 | Active SF 6: UCS C240 M4 SFF | Empty |
| RU-29/30 | Empty | Empty |
| RU-31/32 | Empty | Empty |
| RU-33/34 | Empty | Empty |
| RU-35/36 | Ultra UEM 1C | OpenStack Control C: UCS C240 M4 SFF |
| RU-37/38 | Ultra M Manager: UCS C240 M4 SFF | Empty |
| RU-39/40 | OpenStack Control A: UCS C240 M4 SFF | OpenStack Control B: UCS C240 M4 SFF |
| RU-41/42 | Empty | Empty |
| Cables | Controller Rack Cables | Controller Rack Cables |
| Cables | Spine Uplink/Interconnect Cables | Spine Uplink/Interconnect Cables |
| Cables | Leaf TOR To Spine Uplink Cables | Empty |
| Cables | VNF Rack Cables | Empty |

## Hyper-converged Ultra M XS Multi-VNF Deployment

Table 22: Hyper-converged Ultra M XS Multi-VNF Deployment Rack Layout, on page 46 provides details for the recommended rack layout for the Hyper-converged Ultra M XS Multi-VNF deployment model.

*Table 22: Hyper-converged Ultra M XS Multi-VNF Deployment Rack Layout*

| | Rack #1 | Rack #2 | Rack #3 | Rack #4 |
|---|---|---|---|---|
| RU-1 | Empty | Empty | Empty | Empty |

| | Rack #1 | Rack #2 | Rack #3 | Rack #4 |
|---|---|---|---|---|
| RU-2 | Spine EOR Switch A: Nexus 9236C | Spine EOR Switch B: Nexus 9236C | Empty | Empty |
| RU-3 | Empty | Empty | Empty | Empty |
| RU-4 | VNF Mgmt Switch: Catalyst C3850-48T-S OR C2960XR-48TD | VNF Mgmt Switch: Catalyst C3850-48T-S OR C2960XR-48TD | VNF Mgmt Switch: Catalyst C3850-48T-S OR C2960XR-48TD | VNF Mgmt Switch: Catalyst C3850-48T-S OR C2960XR-48TD |
| RU-5 | VNF Leaf TOR Switch A: Nexus 93180YC-EX | VNF Leaf TOR Switch A: Nexus 93180YC-EX | VNF Leaf TOR Switch A: Nexus 93180YC-EX | VNF Leaf TOR Switch A: Nexus 93180YC-EX |
| RU-6 | VNF Leaf TOR Switch B: Nexus 93180YC-EX | VNF Leaf TOR Switch B: Nexus 93180YC-EX | VNF Leaf TOR Switch B: Nexus 93180YC-EX | VNF Leaf TOR Switch B: Nexus 93180YC-EX |
| RU-7/8 | Ultra UEM 1A: UCS C240 M4 SFF | Ultra UEM 2A: UCS C240 M4 SFF | Ultra UEM 3A: UCS C240 M4 SFF | Ultra UEM 4A: UCS C240 M4 SFF |
| RU-9/10 | Ultra UEM 1B: UCS C240 M4 SFF | Ultra UEM 2B: UCS C240 M4 SFF | Ultra UEM 3B: UCS C240 M4 SFF | Ultra UEM 4B: UCS C240 M4 SFF |
| RU-11/12 | Empty | Empty | Empty | Empty |
| RU-13/14 | Demux SF: UCS C240 M4 SFF | Demux SF: UCS C240 M4 SFF | Demux SF: UCS C240 M4 SFF | Demux SF: UCS C240 M4 SFF |
| RU-15/16 | Standby SF: UCS C240 M4 SFF | Standby SF: UCS C240 M4 SFF | Standby SF: UCS C240 M4 SFF | Standby SF: UCS C240 M4 SFF |
| RU-17/18 | Active SF 1: UCS C240 M4 SFF | Active SF 1: UCS C240 M4 SFF | Active SF 1: UCS C240 M4 SFF | Active SF 1: UCS C240 M4 SFF |
| RU-19/20 | Active SF 2: UCS C240 M4 SFF | Active SF 2: UCS C240 M4 SFF | Active SF 2: UCS C240 M4 SFF | Active SF 2: UCS C240 M4 SFF |
| RU-21/22 | Active SF 3: UCS C240 M4 SFF | Active SF 3: UCS C240 M4 SFF | Active SF 3: UCS C240 M4 SFF | Active SF 3: UCS C240 M4 SFF |
| RU-23/24 | Active SF 4: UCS C240 M4 SFF | Active SF 4: UCS C240 M4 SFF | Active SF 4: UCS C240 M4 SFF | Active SF 4: UCS C240 M4 SFF |
| RU-25/26 | Active SF 5: UCS C240 M4 SFF | Active SF 5: UCS C240 M4 SFF | Active SF 5: UCS C240 M4 SFF | Active SF 5: UCS C240 M4 SFF |

| | Rack #1 | Rack #2 | Rack #3 | Rack #4 |
|---|---|---|---|---|
| RU-27/28 | Active SF 6: UCS C240 M4 SFF | Active SF 6: UCS C240 M4 SFF | Active SF 6: UCS C240 M4 SFF | Active SF 6: UCS C240 M4 SFF |
| RU-29/30 | Empty | Empty | Empty | Empty |
| RU-31/32 | Empty | Empty | Empty | Empty |
| RU-33/34 | Empty | Empty | Empty | Empty |
| RU-35/36 | Ultra UEM 1C,2C,3C,4C | OpenStack Control C: UCS C240 M4 SFF | Empty | Empty |
| RU-37/38 | Ultra M Manager: UCS C240 M4 SFF | Empty | Empty | Empty |
| RU-39/40 | OpenStack Control A: UCS C240 M4 SFF | OpenStack Control B: UCS C240 M4 SFF | Empty | Empty |
| RU-41/42 | Empty | Empty | Empty | Empty |
| Cables | Controller Rack Cables | Controller Rack Cables | Controller Rack Cables | Empty |
| Cables | Spine Uplink/Interconnect Cables | Spine Uplink/Interconnect Cables | Empty | Empty |
| Cables | Leaf TOR To Spine Uplink Cables | Leaf TOR To Spine Uplink Cables | Leaf TOR To Spine Uplink Cables | Leaf TOR To Spine Uplink Cables |
| Cables | VNF Rack Cables | VNF Rack Cables | VNF Rack Cables | VNF Rack Cables |

## Cable the Hardware

After the hardware has been installed, install all power and network cabling for the hardware using the information and instructions in the documentation for the specific hardware product. Refer to Related Documentation, on page 44 for links to the hardware product documentation. Ensure that you install your network cables according to your network plan.

# Configure the Switches

All of the switches must be configured according to your planned network specifications.

**Note** Refer to Network Planning, on page 44 for information and consideration for planning your network.

Refer to the user documentation for each of the switches for configuration information and instructions:

- **Catalyst C2960XR-48TD-I:** http://www.cisco.com/c/en/us/support/switches/catalyst-2960xr-48td-i-switch/model.html
- **Catalyst 3850 48T-S:** http://www.cisco.com/c/en/us/support/switches/catalyst-3850-48t-s-switch/model.html
- **Nexus 93180-YC-EX:** http://www.cisco.com/c/en/us/support/switches/nexus-93180yc-fx-switch/model.html
- **Nexus 9236C:** http://www.cisco.com/c/en/us/support/switches/nexus-9236c-switch/model.html

# Prepare the UCS C-Series Hardware

UCS-C hardware preparation is performed through the Cisco Integrated Management Controller (CIMC). The tables in the following sections list the non-default parameters that must be configured per server type:


- Prepare the Staging Server/Ultra M Manager Node, on page 50
- Prepare the Controller Nodes, on page 50
- Prepare the Compute Nodes, on page 52
- Prepare the OSD Compute Nodes, on page 53


Refer to the UCS C-series product documentation for more information:

- **UCS C-Series Hardware** — https://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-c240-m4-rack-server/model.html
- **CIMC Software** — https://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-c-series-integrated-management-controller/tsd-products-support-series-home.html

**Note** Part of the UCS server preparation is the configuration of virtual drives. If there are virtual drives present which need to be deleted, select the **Virtual Drive Info** tab, select the virtual drive you wish to delete, then click **Delete Virtual Drive**. Refer to the CIMC documentation for more information.

**Note** The information in this section assumes that the server hardware was properly installed per the information and instructions in Install and Cable the Hardware, on page 44.

# Prepare the Staging Server/Ultra M Manager Node

*Table 23: Staging Server/Ultra M Manager Node Parameters*

| Parameters and Settings | Description |
|---|---|
| **CIMC Utility Setup** | |
| Enable IPV4<br>Dedicated<br>No redundancy<br>IP address<br>Subnet mask<br>Gateway address<br>DNS address | Configures parameters for the dedicated management port. |
| **Admin > User Management** | |
| Username<br>Password | Configures administrative user credentials for accessing the CIMC utility. |
| **Admin > Communication Services** | |
| IPMI over LAN Properties = Enabled | Enables the use of Intelligent Platform Management Interface capabilities over the management port. |
| **Server > BIOS > Configure BIOS > Advanced** | |
| Intel(R) Hyper-Threading Technology = Disabled | Disable hyper-threading on server CPUs to optimize Ultra M system performance. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Physical Drive Info** | |
| Status = *Unconfigured Good* | Ensures that the hardware is ready for use. |

# Prepare the Controller Nodes

*Table 24: Controller Node Parameters*

| Parameters and Settings | Description |
|---|---|
| **CIMC Utility Setup** | |

| Parameters and Settings | Description |
|---|---|
| Enable IPV4<br>Dedicated<br>No redundancy<br>IP address<br>Subnet mask<br>Gateway address<br>DNS address | Configures parameters for the dedicated management port. |
| **Admin > User Management** | |
| Username<br>Password | Configures administrative user credentials for accessing the CIMC utility. |
| **Admin > Communication Services** | |
| IPMI over LAN Properties = Enabled | Enables the use of Intelligent Platform Management Interface capabilities over the management port. |
| **Admin > Communication Services** | |
| IPMI over LAN Properties = Enabled | Enables the use of Intelligent Platform Management Interface capabilities over the management port. |
| **Server > BIOS > Configure BIOS > Advanced** | |
| Intel(R) Hyper-Threading Technology = Disabled | Intel(R) Hyper-Threading Technology = Disabled |
| Storage > Cisco 12G SAS Modular RAID Controller > Physical Drive Info | |
| Status = *Unconfigured Good* | Ensures that the hardware is ready for use. |
| Storage > Cisco 12G SAS Modular RAID Controller > Controller Info | |

| Parameters and Settings | Description |
|---|---|
| Virtual Drive Name = OS<br>Read Policy = No Read Ahead<br>RAID Level = RAID 1<br>Cache Policy: Direct IO<br>Strip Size: 64KB<br>Disk Cache Policy: Unchanged<br>Access Policy: Read Write<br>Size: 1143455 MB<br>Write Policy: Write Through | Creates the virtual drives required for use by the operating system (OS). |
| **Storage > Cisco 12G SAS Modular RAID Controller > Virtual Drive Info** | |
| Initialize Type = Fast Initialize | Initializes this virtual drive. A fast initialization quickly writes zeroes to the first and last 10-MB regions of the new virtual drive and completes the initialization in the background. |

# Prepare the Compute Nodes

*Table 25: Compute Node Parameters*

| Parameters and Settings | Description |
|---|---|
| **CIMC Utility Setup** | |
| Enable IPV4<br>Dedicated<br>No redundancy<br>IP address<br>Subnet mask<br>Gateway address<br>DNS address | Configures parameters for the dedicated management port. |
| **Admin > User Management** | |
| Username<br>Password | Configures administrative user credentials for accessing the CIMC utility. |
| **Admin > Communication Services** | |

| Parameters and Settings | Description |
|---|---|
| IPMI over LAN Properties = Enabled | Enables the use of Intelligent Platform Management Interface capabilities over the management port. |
| **Server > BIOS > Configure BIOS > Advanced** | |
| Intel(R) Hyper-Threading Technology = Disabled | Intel(R) Hyper-Threading Technology = Disabled |
| **Storage > Cisco 12G SAS Modular RAID Controller > Physical Drive Info** | |
| Status = *Unconfigured Good* | Ensures that the hardware is ready for use. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Controller Info** | |
| Virtual Drive Name = BOOTOS<br>Read Policy = No Read Ahead<br>RAID Level = RAID 1<br>Cache Policy: Direct IO<br>Strip Size: 64KB<br>Disk Cache Policy: Unchanged<br>Access Policy: Read Write<br>Size: 1143455 MB<br>Write Policy: Write Through | Creates the virtual drives required for use by the operating system (OS). |
| **Storage > Cisco 12G SAS Modular RAID Controller > Virtual Drive Info, BOOTOS** | |
| Initialize Type = Fast Initialize | Initializes this virtual drive. A fast initialization quickly writes zeroes to the first and last 10-MB regions of the new virtual drive and completes the initialization in the background. |
| Set as Boot Drive | Sets the BOOTOS virtual drive as the system boot drive. |

# Prepare the OSD Compute Nodes

**Note** OSD Compute Nodes are only used in Hyper-converged Ultra M models as described in UCS C-Series Servers, on page 9.

*Table 26: OSD Compute Node Parameters*

| Parameters and Settings | Description |
|---|---|
| **CIMC Utility Setup** | |
| Enable IPV4<br>Dedicated<br>No redundancy<br>IP address<br>Subnet mask<br>Gateway address<br>DNS address | Configures parameters for the dedicated management port. |
| **Admin > User Management** | |
| Username<br>Password | Configures administrative user credentials for accessing the CIMC utility. |
| **Admin > Communication Services** | |
| IPMI over LAN Properties = Enabled | Enables the use of Intelligent Platform Management Interface capabilities over the management port. |
| **Server > BIOS > Configure BIOS > Advanced** | |
| Intel(R) Hyper-Threading Technology = Disabled | Intel(R) Hyper-Threading Technology = Disabled |
| **Storage > Cisco 12G SAS Modular RAID Controller > Physical Drive Info** | |
| Status = *Unconfigured Good* | Ensures that the hardware is ready for use. |
| SLOT-HBA Physical Drive Numbers =<br>1<br>2<br>3<br>7<br>8<br>9<br>10 | Ensure the UCS slot host-bus adapter for the drives are configured accordingly. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Physical Drive Number = 1** | |

| Parameters and Settings | Description |
|---|---|
| Virtual Drive Name = BOOTOS<br>Read Policy = No Read Ahead<br>RAID Level = RAID 1<br>Cache Policy: Direct IO<br>Strip Size: 64KB<br>Disk Cache Policy: Unchanged<br>Access Policy: Read Write<br>Size: 285148 MB<br>Write Policy: Write Through | Creates a virtual drive leveraging the storage space available to physical drive number 1.<br>**Note** Ensure that the size of this virtual drive is less than the size of the designated journal and storage drives. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Virtual Drive Info, BOOTOS, Physical Drive Number = 1** | |
| Initialize Type = Fast Initialize | Initializes this virtual drive. A fast initialization quickly writes zeroes to the first and last 10-MB regions of the new virtual drive and completes the initialization in the background. |
| Set as Boot Drive | Sets the BOOTOS virtual drive as the system boot drive. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Physical Drive Number = 2** | |
| Virtual Drive Name = BOOTOS<br>Read Policy = No Read Ahead<br>RAID Level = RAID 1<br>Cache Policy: Direct IO<br>Strip Size: 64KB<br>Disk Cache Policy: Unchanged<br>Access Policy: Read Write<br>Size: 285148 MB<br>Write Policy: Write Through | Creates a virtual drive leveraging the storage space available to physical drive number 2.<br>**Note** Ensure that the size of this virtual drive is less than the size of the designated journal and storage drives. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Virtual Drive Info, BOOTOS, Physical Drive Number = 2** | |
| Initialize Type = Fast Initialize | Initializes this virtual drive. A fast initialization quickly writes zeroes to the first and last 10-MB regions of the new virtual drive and completes the initialization in the background. |
| Set as Boot Drive | Sets the BOOTOS virtual drive as the system boot drive. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Physical Drive Number = 3** | |

| Parameters and Settings | Description |
|---|---|
| Virtual Drive Name = JOURNAL<br>Read Policy = No Read Ahead<br>RAID Level = RAID 0<br>Cache Policy: Direct IO<br>Strip Size: 64KB<br>Disk Cache Policy: Unchanged<br>Access Policy: Read Write<br>Size: 456809 MB<br>Write Policy: Write Through | Creates a virtual drive leveraging the storage space available to physical drive number 3. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Virtual Drive Info, JOURNAL, Physical Drive Number = 3** | |
| Initialize Type = Fast Initialize | Initializes this virtual drive. A fast initialization quickly writes zeroes to the first and last 10-MB regions of the new virtual drive and completes the initialization in the background. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Physical Drive Number = 7** | |
| Virtual Drive Name = OSD1<br>Read Policy = No Read Ahead<br>RAID Level = RAID 0<br>Cache Policy: Direct IO<br>Strip Size: 64KB<br>Disk Cache Policy: Unchanged<br>Access Policy: Read Write<br>Size: 1143455 MB<br>Write Policy: Write Through | Creates a virtual drive leveraging the storage space available to physical drive number 7. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Virtual Drive Info, OSD1, Physical Drive Number = 7** | |
| Initialize Type = Fast Initialize | Initializes this virtual drive. A fast initialization quickly writes zeroes to the first and last 10-MB regions of the new virtual drive and completes the initialization in the background. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Physical Drive Number = 8** | |

| Parameters and Settings | Description |
| --- | --- |
| Virtual Drive Name = OSD2<br>Read Policy = No Read Ahead<br>RAID Level = RAID 0<br>Cache Policy: Direct IO<br>Strip Size: 64KB<br>Disk Cache Policy: Unchanged<br>Access Policy: Read Write<br>Size: 1143455 MB<br>Write Policy: Write Through | Creates a virtual drive leveraging the storage space available to physical drive number 8. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Virtual Drive Info, OSD2, Physical Drive Number = 8** | |
| Initialize Type = Fast Initialize | Initializes this virtual drive. A fast initialization quickly writes zeroes to the first and last 10-MB regions of the new virtual drive and completes the initialization in the background. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Physical Drive Number = 9** | |
| Virtual Drive Name = OSD3<br>Read Policy = No Read Ahead<br>RAID Level = RAID 0<br>Cache Policy: Direct IO<br>Strip Size: 64KB<br>Disk Cache Policy: Unchanged<br>Access Policy: Read Write<br>Size: 1143455 MB<br>Write Policy: Write Through | Creates a virtual drive leveraging the storage space available to physical drive number 9. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Virtual Drive Info, OSD3, Physical Drive Number = 9** | |
| Initialize Type = Fast Initialize | Initializes this virtual drive. A fast initialization quickly writes zeroes to the first and last 10-MB regions of the new virtual drive and completes the initialization in the background. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Physical Drive Number = 10** | |

| Parameters and Settings | Description |
|---|---|
| Virtual Drive Name = OSD4<br>Read Policy = No Read Ahead<br>RAID Level = RAID 0<br>Cache Policy: Direct IO<br>Strip Size: 64KB<br>Disk Cache Policy: Unchanged<br>Access Policy: Read Write<br>Size: 1143455 MB<br>Write Policy: Write Through | Creates a virtual drive leveraging the storage space available to physical drive number 10. |
| **Storage > Cisco 12G SAS Modular RAID Controller > Virtual Drive Info, OSD4, Physical Drive Number = 10** | |
| Initialize Type = Fast Initialize | Initializes this virtual drive. A fast initialization quickly writes zeroes to the first and last 10-MB regions of the new virtual drive and completes the initialization in the background. |

# Deploy the Virtual Infrastructure Manager

Within the Ultra M solution, OpenStack Platform Director (OSP-D) functions as the virtual infrastructure manager (VIM).

The method by which the VIM is deployed depends on the architecture of your Ultra M model. Refer to the following section for information related to your deployment scenario:

- Deploy the VIM for Hyper-Converged Ultra M Models, on page 58

## Deploy the VIM for Hyper-Converged Ultra M Models

Deploying the VIM for Hyper-Converged Ultra M Models is performed using an automated workflow enabled through software modules within Ultra Automation Services (UAS). These services leverage user-provided configuration information to automatically deploy the VIM Orchestrator (Undercloud) and the VIM (Overcloud).

For information on using this automated process, in the *USP Deployment Automation Guide*, refer to the *Virtual Infrastructure Manager Installation Automation* section.

# Deploy the USP-Based VNF

After the OpenStack Undercloud (VIM Orchestrator) and Overcloud (VIM) have been successfully deployed on the Ultra M hardware, you must deploy the USP-based VNF.

This process is performed through the Ultra Automation Services (UAS). UAS is an automation framework consisting of a set of software modules used to automate the USP-based VNF deployment and related components such as the VNFM.

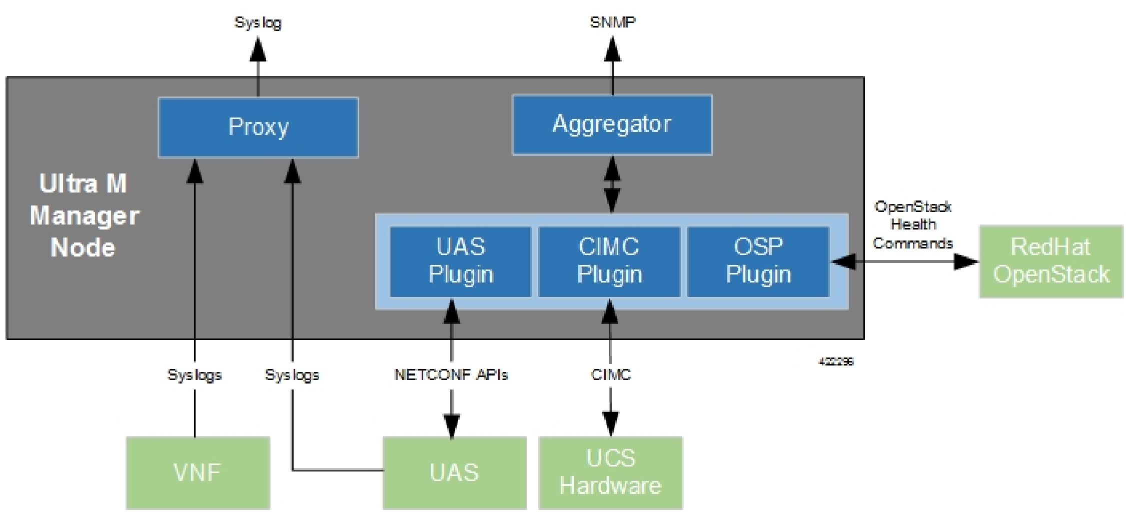
For detailed information on the automation workflow, refer to the *Ultra Service Platform Deployment Automation Guide*.

CHAPTER 6

# Event and Syslog Management Within the Ultra M Solution

Hyper-Converged Ultra M solution models support a centralized monitor and management function. This function provides a central aggregation point for events (faults and alarms) and a proxy point for syslogs generated by the different components within the solution as identified in Table 27: Component Event Sources, on page 66. This monitor and management function runs on the Ultra M Manager Node.

***Figure 13: Ultra M Manager Node Event and Syslog Functions***




The software to enable this functionality is distributed as a both a stand-alone RPM and as part of the Ultra Services Platform (USP) release ISO as described in Install the Ultra M Manager RPM, on page 76. Once installed, additional configuration is required based on the desired functionality as described in the following sections:

# Syslog Proxy

The Ultra M Manager Node can be configured as a proxy server for syslogs received from UCS servers and/or OpenStack. As a proxy, the Ultra M Manager Node acts a single logging collection point for syslog messages from these components and relays them to a remote collection server.

**NOTES:**

- This functionality is currently supported only with Ultra M deployments based on OSP 10 and that leverage the Hyper-Converged architecture.

  .

- You must configure a remote collection server to receive and filter log files sent by the Ultra M Manager Node.
- Though you can configure syslogging at any severity level your deployment scenario requires, it is recommended that you only configure syslog levels with severity levels 0 (emergency) through 4 (warning).

Once the Ultra M Manager RPM is installed, a script provided with this release allows you to quickly enable syslog on the nodes and set the Ultra M Manager Node as the proxy. Leveraging inputs from a YAML-based configuration file, the script:

- Inspects the nodes within the Undercloud and Overcloud
- Logs on to each node
- Enables syslogging at the specified level or both the UCS hardware and for OpenStack
- Sets the Ultra M Manager Node's address as the syslog proxy

**Important** The use of this script assumes that all of the nodes use the same login credentials.

To enable this functionality:

1 Install the Ultra M Manager bundle RPM using the instructions in Install the Ultra M Manager RPM, on page 76.

**Note** This step is not needed if the Ultra M Manager bundle was previously installed.

2 Become the root user.

**sudo -i**

3 Verify that there are no previously existing configuration files for logging information messages in /etc/rsyslog.d.

**a** Navigate to */etc/rsyslog.d*.

**cd /etc/rsyslog.d**
**ls -al**
Example output:

```
total 24
drwxr-xr-x.   2 root root  4096 Sep  3 23:17 .
drwxr-xr-x. 152 root root 12288 Sep  3 23:05 ..
-rw-r--r--.   1 root root    49 Apr 21 00:03 listen.conf
-rw-r--r--.   1 root root   280 Jan 12  2017 openstack-swift.conf
```

**b** Check the *listen.conf* file.

**cat listen.conf**
Example output:

```
$SystemLogSocketName /run/systemd/journal/syslog
```

**c** Check the configuration of the *openstack-swift.conf*.

**cat openstack-swift.conf**
Example configuration:

```
# LOCAL0 is the upstream default and LOCAL2 is what Swift gets in
# RHOS and RDO if installed with Packstack (also, in docs).
# The breakout action prevents logging into /var/log/messages, bz#997983.
local0.*;local2.* /var/log/swift/swift.log
& stop
```

**4** Enable syslogging to the external server by configuring the */etc/rsyslog.conf* file.

**vi /etc/rsyslog.conf**

**a** Enable TCP/UDP reception.

```
# provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514

# provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514
```

**b** Disable logging for private authentication messages.

```
# Don't log private authentication messages!
#*.info;mail.none;authpriv.none;cron.none                /var/log/messages
```

**c** Configure the desired log severity levels.

```
# log 0-4 severity logs to external server 172.21.201.53
*.4,3,2,1,0     @<external_syslog_server_ipv4_address>:514
```

This enables the collection and reporting of logs with severity levels 0 (emergency) through 4 (warning).

**Caution** Though it is possible to configure the system to locally store syslogs on the Ultra M Manager Node, it is highly recommended that you avoid doing so to avoid the risk of data loss and to preserve disk space.

**5** Restart the syslog server.

**service rsyslog restart**

**6** Navigate to */etc*.

**cd /etc**

**7** Create and/or edit the *ultram_cfg.yaml* file based your VIM Orchestrator and VIM configuration. A sample of this configuration file is provided in Example ultram_cfg.yaml File, on page 89.

**Important** The *ultram_cfg.yaml* file pertains to both the syslog proxy and event aggregation functionality. Some parts of this file's configuration overlap and may have been configured in relation to the other function.

**vi ultram_cfg.yaml**

a *Optional*. Configure your Undercloud settings if they are not already configured.

```
under-cloud:
  OS_AUTH_URL: <auth_url>
  OS_USERNAME: admin
  OS_TENANT_NAME: <tenant_name>
  OS_PASSWORD: <admin_user_password>
  ssh-key: /opt/cisco/heat_admin_ssh_key
```

b *Optional.* Configure your Overcloud settings if they are not already configured.

```
over-cloud:
  enabled: true
  environment:
    OS_AUTH_URL: <auth_url>
    OS_TENANT_NAME: <tenant_name>
    OS_USERNAME: <user_name>
    OS_PASSWORD: <user_password>
    OS_ENDPOINT_TYPE: publicURL
    OS_IDENTITY_API_VERSION: 2
    OS_REGION_NAME: regionOne
```

c Specify the IP address of the Ultra M Manager Node to be the proxy server.

```
<-- SNIP -->
rsyslog:
  level: 4,3,2,1,0
  proxy-rsyslog: <ultram_manager_address>
```

**Important**

- You can modify the syslog levels to report according to your requirements using the **level** parameter as shown above.
- *<ultram_manager_address>* is the internal IP address of the Ultra M Manager Node reachable by OpenStack and the UCS servers.
- If you are copying the above information from an older configuration, make sure the **proxy-rsyslog** IP address does not contain a port number.

d *Optional*. Configure the CIMC login information for each of the nodes on which syslogging is to be enabled.

```
ucs-cluster:
  enabled: true
  user: <username>
  password: <password>
```

**Important** The use of this script assumes that all of the nodes use the same login credentials.

8 Navigate to */opt/cisco/usp/ultram-manager*.

**cd /opt/cisco/usp/ultram-manager**

9 Encrypt the clear text passwords in the *ultram_cfg.yaml* file.

**utils.py --secure-cfg /etc/ultram_cfg.yaml**

**Important** Executing this scripts encrypts the passwords in the configuration file and appends "encrypted: true" to the end of the file (e.g. *ultram_cfg.yamlencrypted: true*). Refer to Encrypting Passwords in the ultram_cfg.yaml File, on page 80 for more information.

10 *Optional*. Disable rsyslog if it was previously configured on the UCS servers.

**./ultram_syslogs.py --cfg /etc/ultram_cfg.yaml –u -d**

11 Execute the *ultram_syslogs.py* script to load the configuration on the various nodes.

**./ultram_syslogs.py --cfg /etc/ultram_cfg.yaml -o –u**

**Important** Additional command line options for the *ultram_syslogs.py* script can be seen by entering **ultram_syslogs.py –help** at the command prompt. An example of the output of this command is below:

```
usage: ultram_syslogs.py [-h] -c CFG [-d] [-u] [-o]

optional arguments:
  -h, --help            show this help message and exit
  -c CFG, --cfg CFG     Configuration file
  -d, --disable-syslog  Disable Syslog
  -u, --ucs             Apply syslog configuration on UCS servers
  -o, --openstack       Apply syslog configuration on OpenStack
```

Example output:

```
2017-09-13 15:24:23,305 - Configuring Syslog server 192.200.0.1:514 on UCS cluster
2017-09-13 15:24:23,305 - Get information about all the nodes from under-cloud
2017-09-13 15:24:37,178 - Enabling syslog configuration on 192.100.3.5
2017-09-13 15:24:54,686 - Connected.
2017-09-13 15:25:00,546 - syslog configuration success.
2017-09-13 15:25:00,547 - Enabling syslog configuration on 192.100.3.6
2017-09-13 15:25:19,003 - Connected.
2017-09-13 15:25:24,808 - syslog configuration success.
<---SNIP--->

<---SNIP--->
2017-09-13 15:46:08,715 - Enabling syslog configuration on vnf1-osd-compute-1
[192.200.0.104]
2017-09-13 15:46:08,817 - Connected
2017-09-13 15:46:09,046 - - /etc/rsyslog.conf
2017-09-13 15:46:09,047 -  Enabling syslog ...
2017-09-13 15:46:09,130 -  Restarting rsyslog
2017-09-13 15:46:09,237 -  Restarted
2017-09-13 15:46:09,321 - - /etc/nova/nova.conf
2017-09-13 15:46:09,321 -  Enabling syslog ...
2017-09-13 15:46:09,487 -  Restarting Services 'openstack-nova-compute.service'
```

12 Ensure that client log messages are being received by the server and are uniquely identifiable.

**NOTES:**

- If necessary, configure a unique tag and hostname as part of the syslog configuration/template for each client.
- Syslogs are very specific in terms of the file permissions and ownership. If need be, manually configure permissions for the log file on the client using the following command:

  **chmod +r** *<URL>/<log_filename>*

# Event Aggregation

The Ultra M Manager Node can be configured to aggregate events received from different Ultra M components as identified in Table 27: Component Event Sources, on page 66.

**Important** This functionality is currently supported only with Ultra M deployments based on OSP 10 and that leverage the Hyper-Converged architecture.

*Table 27: Component Event Sources*

| Solution Component | Event Source Type | Details |
| --- | --- | --- |
| UCS server hardware | CIMC | Reports on events collected from UCS C-series hardware via CIMC-based subscription.<br>These events are monitored in real-time. |
| VIM (Overcloud) | OpenStack service health | Reports on OpenStack service fault events pertaining to:<br>• Failures (stopped, restarted)<br>• High availability<br>• Ceph / storage<br>• Neutron / compute host and network agent<br>• Nova scheduler (VIM instances)<br>Refer to Table 28: Monitored OpenStack Services, on page 68 for a complete list of services.<br>**Important** In order to ensure optimal performance, it is strongly recommended that you do not change the default polling-interval. |
| UAS (AutoVNF, UEM, and ESC) | UAS cluster/USP management component events | Reports on UAS service fault events pertaining to:<br>• Service failure (stopped, restarted)<br>• High availability<br>• AutoVNF<br>• UEM<br>• ESC (VNFM)<br>**Important** In order to ensure optimal performance, it is strongly recommended that you do not change the default polling-interval. |

| Solution Component | Event Source Type | Details |
|---|---|---|
| VNF VM Status | ESC (VNFM) event notifications | Reports on VNF VM deployment state events generated by ESC (the VNFM). The following events are supported:<br>• VM_DEPLOYED<br>• VM_ALIVE<br>• VM_UNDEPLOYED<br>• VM_REBOOTED<br>• VM_RECOVERY_REBOOT<br>• VM_RECOVERY_UNDEPLOYED<br>• VM_RECOVERY_DEPLOYED<br>• VM_RECOVERY_COMPLETE<br>• VM_STOPPED<br>**Important** This feature is not fully qualified in this release. It is available only for testing purposes. AutoVNF monitors for event notifications from ESC in real time. Though AutoVNF updates the VNFR for the VNF and VNFC the event pertains to upon receipt of an event, it does not generate a corresponding SNMP trap. |

*Table 28: Monitored OpenStack Services*

| Node Type | OpenStack Module | OpenStack Services |
|---|---|---|
| Controller | aodh | • openstack-aodh-evaluator.service,<br>• openstack-aodh-listener.service,<br>• openstack-aodh-notifier.service |
| | ceilometer | • openstack-ceilometer-central.service,<br>• openstack-ceilometer-collector.service,<br>• openstack-ceilometer-notification.service |
| | cinder | • openstack-cinder-api.service,<br>• openstack-cinder-scheduler.service |
| | glance | • openstack-glance-api.service,<br>• openstack-glance-registry.service |
| | gnocchi | • openstack-gnocchi-metricd.service,<br>• openstack-gnocchi-statsd.service |
| | heat-engine | openstack-heat-engine.service |
| | heat-api | • openstack-heat-api-cfn.service,<br>• openstack-heat-api-cloudwatch.service,<br>• openstack-heat-api.service |
| | heat | • openstack-heat-api-cfn.service,<br>• openstack-heat-api-cloudwatch.service,<br>• openstack-heat-api.service |
| | nova | |

| Node Type | OpenStack Module | OpenStack Services |
|---|---|---|
| | | • openstack-nova-api.service,<br>• openstack-nova-conductor.service,<br>• penstack-nova-consoleauth.service,<br>• openstack-nova-novncproxy.service,<br>• openstack-nova-scheduler.service |
| | swift-object | • openstack-swift-object-auditor.service,<br>• openstack-swift-object-replicator.service,<br>• openstack-swift-object-updater.service,<br>• openstack-swift-object.service |
| | swift-account | • openstack-swift-account-auditor.service,<br>• openstack-swift-account-reaper.service,<br>• openstack-swift-account-replicator.service,<br>• openstack-swift-account.service |
| | swift-container | • openstack-swift-container-auditor.service,<br>• openstack-swift-container-replicator.service,<br>• openstack-swift-container-updater.service,<br>• openstack-swift-container.service |
| | swift-proxy | openstack-swift-proxy.service |
| | swift | All above swift services |
| | ntpd | ntpd.service |
| | mongod | mongod.service |
| | memcached | memcached |
| | neutron-dhcp-agent | neutron-dhcp-agent.service |
| | neutron-l3-agent | neutron-l3-agent.service |
| | neutron-metadata-agent | neutron-metadata-agent.service |

| Node Type | OpenStack Module | OpenStack Services |
|---|---|---|
| | neutron-openvswitch-agent | neutron-openvswitch-agent.service |
| | neutron-server | neutron-server.service |
| | httpd | httpd.service |
| OSD Compute | ceph-mon.target | ceph-mon.target |
| | ceph-radosgw.target | ceph-radosgw.target |
| | ceph.target | ceph.target |
| | openvswitch.service | openvswitch.service |
| | neutron-sriov-nic-agent | neutron-sriov-nic-agent.service |
| | neutron-openvswitch-agent | neutron-openvswitch-agent.service |
| | ntpd | ntpd.service |
| | nova-compute | openstack-nova-compute.service |
| | libvirtd | libvirtd.service |
| Compute | ceph-mon.target | ceph-mon.target |
| | ceph-radosgw.target | ceph-radosgw.target |
| | ceph.target | ceph.target |
| | openvswitch.service | openvswitch.service |
| | neutron-sriov-nic-agent | neutron-sriov-nic-agent.service |
| | neutron-openvswitch-agent | neutron-openvswitch-agent.service |
| | ntpd | ntpd.service |
| | nova-compute | openstack-nova-compute.service |
| | libvirtd | libvirtd.service |

Events received from the solution components, regardless of the source type, are mapped against the Ultra M SNMP MIB (CISCO-ULTRAM-MIB.my, refer to Ultra M MIB, on page 93). The event data is parsed and categorized against the following conventions:

- **Fault code:** Identifies the area in which the fault occurred for the given component. Refer to the "CFaultCode" convention within the Ultra M MIB for more information.
- **Severity:** The severity level associated with the fault. Refer to the "CFaultSeverity" convention within the Ultra M MIB for more information. Since the Ultra M Manager Node aggregates events from different components within the solution, the severities supported within the Ultra M Manager Node MIB map to those for the specific components. Refer to Ultra M Component Event Severity and Fault Code Mappings, on page 99 for details.
- **Domain:** The component in which the fault occurred (e.g. UCS hardware, VIM, UEM, etc.). Refer to the "CFaultDomain" convention within the Ultra M MIB for more information.

UAS and OpenStack events are monitored at the configured polling interval as described in Table 29: SNMP Fault Entry Table Element Descriptions, on page 73. At the polling interval, the Ultra M Manager Node:

1. Collects data from UAS and OpenStack.
2. Generates/updates .log and .report files and an SNMP-based fault table with this information. It also includes related data about the fault such as the specific source, creation time, and description.
3. Processes any events that occurred:
    a. If an error or fault event is identified, then a .error file is created and an SNMP trap is sent.
    b. If the event received is a clear condition, then an informational SNMP trap is sent to "clear" an active fault.
    c. If no event occurred, then no further action is taken beyond Step 2.

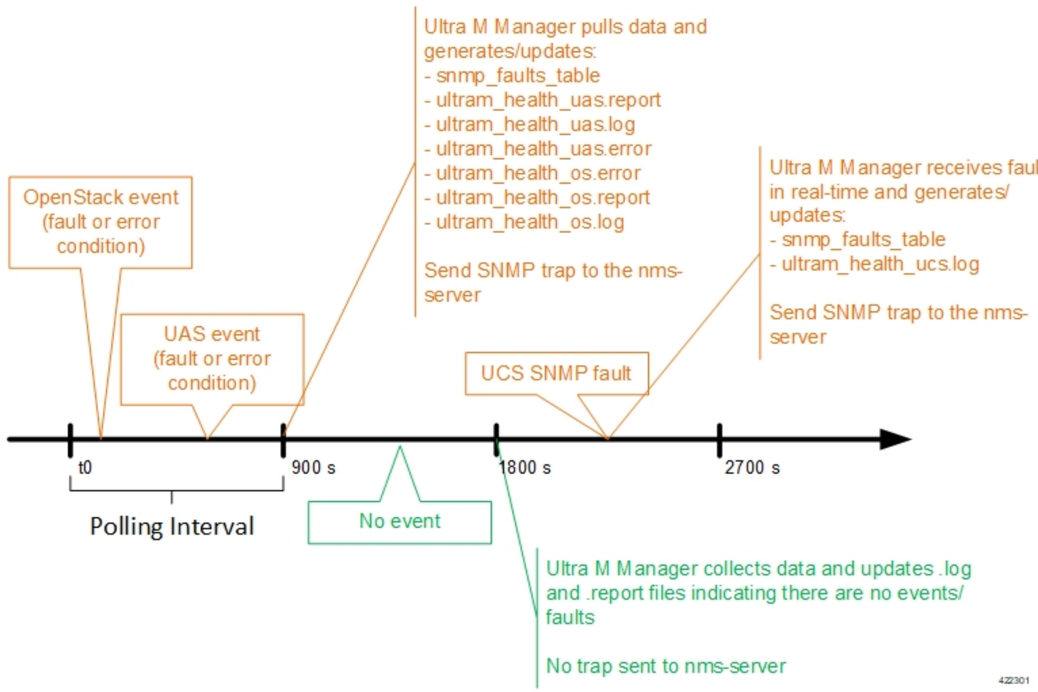
UCS and ESC VM events are monitored and acted upon in real-time. When events occur, the Ultra M Manager generates a .log file and the SNMP fault table. In the case of VM events reported by ESC, upon receipt of an event, AutoVNF updates the VNFR for the VNF and VNFC the event pertains to. In parallel, it passes the event information to the Ultra M Manager functionality within AutoIT. The Ultra M Manager then generates corresponding SNMP traps for each event.

Active faults are reported "only" once and not on every polling interval. As a result, there is only one trap as long as this fault is active. Once the fault is "cleared", an informational trap is sent.

**Important** UCS events are considered to be the "same" if a previously received fault has the same distinguished name (DN), severity, and lastTransition time. UCS events are considered as "new" only if any of these elements change.

These processes are illustrated in Figure 14: Ultra M Manager Node Event Aggregation Operation, on page 72. Refer to About Ultra M Manager Log Files, on page 113 for more information.

*Figure 14: Ultra M Manager Node Event Aggregation Operation*

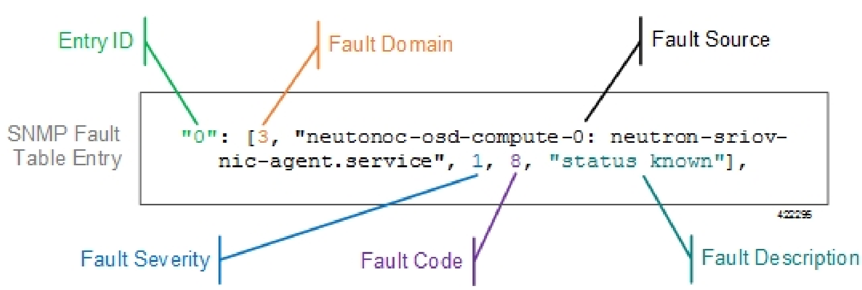



An example of the snmp_faults_table file is shown below and the entry syntax is described in Figure 15: SNMP Fault Table Entry Description, on page 72:

```
"0": [3 "neutonoc-osd-compute-0: neutron-sriov-nic-agent.service" 1 8 "status known"] "1":
 [3 "neutonoc-osd-compute-0: ntpd" 1 8 "Service is not active state: inactive"] "2": [3
"neutonoc-osd-compute-1: neutron-sriov-nic-agent.service" 1 8 "status known"] "3": [3
"neutonoc-osd-compute-1: ntpd" 1 8 "Service is not active state: inactive"] "4": [3
"neutonoc-osd-compute-2: neutron-sriov-nic-agent.service" 1 8 "status known"] "5": [3
"neutonoc-osd-compute-2: ntpd" 1 8 "Service is not active state: inactive"]
```

Refer to About Ultra M Manager Log Files, on page 113 for more information.

*Figure 15: SNMP Fault Table Entry Description*




Each element in the SNMP Fault Table Entry corresponds to an object defined in the Ultra M SNMP MIB as described in Table 29: SNMP Fault Entry Table Element Descriptions, on page 73. (Refer also to Ultra M MIB, on page 93.)

*Table 29: SNMP Fault Entry Table Element Descriptions*

| SNMP Fault Table Entry Element | MIB Object | Additional Details |
|---|---|---|
| Entry ID | cultramFaultIndex | A unique identifier for the entry |
| Fault Domain | cultramFaultDomain | The component area in which the fault occurred. The following domains are supported in this release:<br>• **hardware(1) :** Harware including UCS servers<br>• **vim(3) :** OpenStack VIM manager<br>• **uas(4) :** Ultra Automation Services Modules |
| Fault Source | cultramFaultSource | Information identifying the specific component within the Fault Domain that generated the event.<br>The format of the information is different based on the Fault Domain. Refer to Table 30: cultramFaultSource Format Values, on page 75 for details. |
| Fault Severity | cultramFaultSeverity | The severity associated with the fault as one of the following:<br>• **emergency(1) :** System level FAULT impacting multiple VNFs/Services<br>• **critical(2) :** Critical Fault specific to VNF/Service<br>• **major(3) :** component level failure within VNF/service.<br>• **alert(4) :** warning condition for a service/VNF, may eventually impact service.<br>• **informational(5) :** informational only, does not impact service<br>Refer to Ultra M Component Event Severity and Fault Code Mappings, on page 99 for details on how these severities map to events generated by the various Ultra M components. |

| SNMP Fault Table Entry Element | MIB Object | Additional Details |
|---|---|---|
| Fault Code | cultramFaultCode | A unique ID representing the type of fault as. The following codes are supported:<br>• **other(1) :** Other events<br>• **networkConnectivity(2) :** Network Connectivity Failure Events<br>• **resourceUsage(3) :** Resource Usage Exhausted Event<br>• **resourceThreshold(4) :** Resource Threshold crossing alarms<br>• **hardwareFailure(5) :** Hardware Failure Events<br>• **securityViolation(6) :** Security Alerts<br>• **configuration(7) :** Config Error Events<br>• **serviceFailure(8) :** Process/Service failures<br>Refer to Ultra M Component Event Severity and Fault Code Mappings, on page 99 for details on how these fault codes map to events generated by the various Ultra M components. |
| Fault Description | cultramFaultDescription | A message containing details about the fault. |

*Table 30: cultramFaultSource Format Values*

| FaultDomain | Format Value of cultramFaultSource |
|---|---|
| Hardware (UCS Servers) | **Node:** *<UCS-SERVER-IP-ADDRESS>*, **affectedDN:** *<FAULT-OBJECT-DISTINGUSIHED-NAME>*<br>Where:<br>*<UCS-SERVER-IP-ADDRESS> :* The management IP address of the UCS server that generated the fault.<br>*<FAULT-OBJECT-DISTINGUSIHED-NAME> :* The distinguished name of the affected UCS object. |
| UAS | **Node:** *<UAS-MANAGEMENT-IP>*<br>Where:<br>*<UAS-MANAGEMENT-IP> :* The management IP address for the UAS instance. |
| VIM (OpenStack) | *<OS-HOSTNAME>*: *<SERVICE-NAME>*<br>Where:<br>*<OS-HOSTNAME>* : The OpenStack node hostname that generated the fault.<br>*<SERVICE-NAME>* : Then name of the OpenStack service that generated the fault. |

Fault and alarm collection and aggregation functionality within the Hyper-Converged Ultra M solution is configured and enabled through the *ultram_cfg.yaml* file. (An example of this file is located in Example ultram_cfg.yaml File, on page 89.) Parameters in this file dictate feature operation and enable SNMP on the UCS servers and event collection from the other Ultra M solution components.

To enable this functionality on the Ultra M solution:

1. Install the Ultra M Manager bundle RPM using the instructions in Install the Ultra M Manager RPM, on page 76.

**Important** This step is not needed if the Ultra M Manager bundle was previously installed.

2. Become the root user.

   **sudo -i**
3. Navigate to /etc.

   **cd /etc**
4. Create and/or edit the *ultram_cfg.yaml* file based on your deployment scenario.

**Important** The *ultram_cfg.yaml* file pertains to both the syslog proxy and event aggregation functionality. Some parts of this file's configuration overlap and may have been configured in relation to the other function.

5 Navigate to */opt/cisco/usp/ultram-manager*.

**cd /opt/cisco/usp/ultram-manager**

6 Encrypt the clear text passwords in the *ultram_cfg.yaml* file.

**utils.py --secure-cfg /etc/ultram_cfg.yaml**

**Important** Executing this scripts encrypts the passwords in the configuration file and appends "encrypted: true" to the end of the file (e.g. *ultram_cfg.yamlencrypted: true*). Refer to Encrypting Passwords in the ultram_cfg.yaml File, on page 80 for more information.

7 Start the Ultra M Manager Service, on page 79.

**Important** Subsequent configuration changes require you restart the health monitor service. Refer to Restarting the Ultra M Manager Service, on page 77 for details.

8 Verify the configuration by checking the *ultram_health.log* file.

**cat /var/log/cisco/ultram_health.log**

# Install the Ultra M Manager RPM

The Ultra M Manager functionality described in this chapter is enabled through software distributed both as part of the USP ISO and as a separate RPM bundle.

Ensure that you have access to either of these RPM bundles prior to proceeding with the instructions below.

To access the Ultra M Manager RPM packaged within the USP ISO, onboard the ISO and navigate to the *ultram_manager* directory. Refer to the *USP Deployment Automation Guide* for instructions on onbarding the USP ISO.

1 *Optional*. Remove any previously installed versions of the Ultra M Manager per the instructions in Uninstalling the Ultra M Manager, on page 79.

2 Log on to the Ultra M Manager Node.

3 Become the root user.

**sudo -i**

4 Copy the "ultram-manager" RPM file to the Ultra M Manager Node.

5 Navigate to the directory in which you copied the file.

6 Install the ultram-manager bundle RPM that was distributed with the ISO.

**yum install -y ultram-manager-**<*version*>**.x86_64.rpm**

A message similar to the following is displayed upon completion:

```
Installed:
  ultram-health.x86_64 0:5.1.6-2
```

```
Complete!
```

7 Verify that log rotation is enabled in support of the syslog proxy functionality by checking the *logrotate* file.

**cd /etc/cron.daily**
**ls -al**

Example output:

```
total 28
drwxr-xr-x.   2 root root  4096 Sep 10 18:15 .
drwxr-xr-x. 128 root root 12288 Sep 11 18:12 ..
-rwx------.   1 root root   219 Jan 24  2017 logrotate
-rwxr-xr-x.   1 root root   618 Mar 17  2014 man-db.cron
-rwx------.   1 root root   256 Jun 21 16:57 rhsmd
```

**cat /etc/cron.daily/logrotate**

Example output:

```
#!/bin/sh

/usr/sbin/logrotate -s /var/lib/logrotate/logrotate.status /etc/logrotate.conf
EXITVALUE=$?
if [ $EXITVALUE != 0 ]; then
    /usr/bin/logger -t logrotate "ALERT exited abnormally with [$EXITVALUE]"
fi
exit 0
```

8 Create and configure the *ultram_health* file.

**cd /etc/logrotate.d**
**vi ultram_health**

```
/var/log/cisco/ultram-manager/* {
   size 50M
   rotate 30
   missingok
   notifempty
   compress
   }
```

9 Proceed to either Syslog Proxy, on page 62 or Event Aggregation , on page 66 to configure the desired functionality.

# Restarting the Ultra M Manager Service

In the event of configuration change or a server reboot, the Ultra M Manager service must be restarted.

To restart the Ultra M Manager service:

1 Check the Ultra M Manager Service Status, on page 78.

2 Stop the Ultra M Manager Service, on page 78.

3 Start the Ultra M Manager Service, on page 79.

4 Check the Ultra M Manager Service Status, on page 78.

# Check the Ultra M Manager Service Status

It may be necessary to check the status of the Ultra M Manager service.

**Note** These instructions assume that you are already logged into the Ultra M Manager Node as the *root* user.

To check the Ultra M Manager status:

1 Check the service status.

**service ultram_health.service status**
Example Output – Inactive Service:

```
Redirecting to /bin/systemctl status ultram_health.service
 ultram_health.service - Cisco UltraM Health monitoring Service
   Loaded: loaded (/etc/systemd/system/ultram_health.service; enabled; vendor preset:
disabled)
   Active: inactive (dead)
```

Example Output – Active Service:

```
Redirecting to /bin/systemctl status ultram_health.service
 ultram_health.service - Cisco UltraM Health monitoring Service
   Loaded: loaded (/etc/systemd/system/ultram_health.service; enabled; vendor preset:
disabled)
   Active: active (running) since Sun 2017-09-10 22:20:20 EDT; 5s ago
 Main PID: 16982 (start_ultram_he)
   CGroup: /system.slice/ultram_health.service
           ├─16982 /bin/sh /usr/local/sbin/start_ultram_health
           ├─16983 python /opt/cisco/usp/ultram-manager/ultram_health.py
/etc/ultram_cfg.yaml
           ├─16991 python /opt/cisco/usp/ultram-manager/ultram_health.py
/etc/ultram_cfg.yaml
           └─17052 /usr/bin/python /bin/ironic node-show
19844e8d-2def-4be4-b2cf-937f34ebd117

Sep 10 22:20:20 ospd-tb1.mitg-bxb300.cisco.com systemd[1]: Started Cisco UltraM Health
monitoring Service.
Sep 10 22:20:20 ospd-tb1.mitg-bxb300.cisco.com systemd[1]: Starting Cisco UltraM Health
 monitoring Service...
Sep 10 22:20:20 ospd-tb1.mitg-bxb300.cisco.com start_ultram_health[16982]: 2017-09-10
22:20:20,411 - UCS Health Check started
```

2 Check the status of the mongo process.

**ps -ef | grep mongo**
Example output:

```
mongodb   3769    1  0 Aug23 ?       00:43:30 /usr/bin/mongod --quiet -f /etc/mongod.conf
 run
```

# Stop the Ultra M Manager Service

It may be necessary to stop the Ultra M Manager service under certain circumstances.

**Note** These instructions assume that you are already logged into the Ultra M Manager Node as the root user.

To stop the Ultra M Manager service, enter the following command from the */opt/cisco/usp/ultram-manager* directory:

**./service ultram_health.service stop**

## Start the Ultra M Manager Service

It is necessary to start/restart the Ultra M Manager service in order to execute configuration changes and or after a reboot of the Ultra M Manager Node.

**Note** These instructions assume that you are already logged into the Ultra M Manager Node as the root user.

To start the Ultra M Manager service, enter the following command from the */opt/cisco/usp/ultram-manager* directory:

**./service ultram_health.service start**

# Uninstalling the Ultra M Manager

If you have previously installed the Ultra M Manager, you must uninstall it before installing newer releases.

To uninstall the Ultra M Manager:

1. Log on the Ultra M Manager Node.
2. Become the root user.

   **sudo -i**
3. Make a backup copy of the existing configuring file (e.g. /etc/ultram_cfg.yaml).
4. Check the installed version.

   **yum list installed | grep ultra**
   Example output:

```
ultram-manager.x86_64         5.1.3-1            installed
```

5. Uninstall the previous version.

   **yum erase ultram-manager**
   Example output:

```
Loaded plugins: enabled_repos_upload, package_upload, product-id, search-disabled-repos,
 subscription-manager, versionlock
Resolving Dependencies
--> Running transaction check
---> Package ultram-manager.x86_64 0:5.1.5-1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

=========================================================================================

 Package                    Arch                Version                 Repository
        Size
=========================================================================================
Removing:
 ultram-health              x86_64              5.1.5-1                 installed
       148 k

Transaction Summary
=========================================================================================
Remove  1 Package

Installed size: 148 k
Is this ok [y/N]:
```

   Enter **y** at the prompt to continue.

A message similar to the following is displayed upon completion:

```
Removed:
  ultram-health.x86_64 0:5.1.3-1

Complete!
Uploading Enabled Reposistories Report
Loaded plugins: product-id, versionlock
```

6 Proceed to Install the Ultra M Manager RPM, on page 76

# Encrypting Passwords in the *ultram_cfg.yaml* File

The ultram_cfg.yaml file requires the specification of passwords for the managed components. These passwords are entered in clear text within the file. To mitigate security risks, the passwords should be encrypted before using the file to deploy Ultra M Manager-based features/functions.

To encrypt the passwords, the Ultra M Manager provides a script called *utils.py* in the */opt/cisco/usp/ultram-manager/* directory. The script can be run against your *ultram_cfg.yaml* file by navigating to that directory and executing the following command as the root user:

**utils.py --secure-cfg /etc/ultram_cfg.yaml**

**Important** Data is encrypted using AES via a 256 bit key that is stored in the MongoDB. As such, an OSPD user on OSPD is able to access this key and possibly decrypt the passwords. (This includes the *stack* user as it has sudo access.)

Executing this scripts encrypts the passwords in the configuration file and appends "encrypted: true" to the end of the file (e.g. *ultram_cfg.yamlencrypted: true*) to indicate that the passwords have been encrypted.

**Note** Do not rename the file once the filename has been changed.

If need be, you can make edits to parameters other than the passwords within the *ultram_cfg.yaml* file after encrypting the passwords.

For new installations, run the script to encrypt the passwords before applying the configuration and starting the Ultra M Manager service as described in Syslog Proxy, on page 62 and Event Aggregation , on page 66.

To encrypt passwords for exsiting installations:

1 Stop the Ultra M Manager Service, on page 78.

2 *Optional.* Installing an updated version of the Ultra M Manager RPM.

   a Save a copy of your *ultram_cfg.yaml* file to alternate location outside of the Ultra M Manager installation.

   b Uninstall the Ultra M Manager using the instructions in Uninstalling the Ultra M Manager, on page 79.

   c Install the new Ultra M Manager version using the instructions in Install the Ultra M Manager RPM, on page 76.

   d Copy your backed-up *ultram_cfg.yaml* file to the */etc directory*.

3 Navigate to /opt/cisco/usp/ultram-manager/.

**cd /opt/cisco/usp/ultram-manager/**

4 Encrypt the clear text passwords in the *ultram_cfg.yaml* file.

**utils.py --secure-cfg /etc/ultram_cfg.yaml**

**Note** Executing this scripts encrypts the passwords in the configuration file and appends “encrypted: true” to the end of the file (e.g. ultram_cfg.yamlencrypted: true).

5 Start the Ultra M Manager Service, on page 79.

APPENDIX A

# Network Definitions (Layer 2 and 3)

Table 31: Layer 2 and 3 Network Definition, on page 83 is intended to be used as a template for recording your Ultra M network Layer 2 and Layer 3 deployments.

Some of the Layer 2 and 3 networking parameters identified in Table 31: Layer 2 and 3 Network Definition, on page 83 are configured directly on the UCS hardware via CIMC. Other parameters are configured as part of the VIM Orchestrator or VIM configuration. This configuration is done through various configuration files depending on the parameter:

- undercloud.conf
- network.yaml
- layout.yaml

***Table 31: Layer 2 and 3 Network Definition***

| VLAN ID / Range | Network | Gateway | IP Range Start | IP Range End | Description | Where Configured | Routable? |
|---|---|---|---|---|---|---|---|
| **External-Internet Meant for OSP-D Only** | | | | | | | |
| 100 | 192.168.1.0 /24 | 192.168.1.1 | | | Internet access required:<br>- 1 IP Address for OSP-D<br>- 1 IP for default gateway | On Ultra M Manger Node hardware | Yes |
| **External – Floating IP Addresses (Virtio)*** | | | | | | | |

| VLAN ID / Range | Network | Gateway | IP Range Start | IP Range End | Description | Where Configured | Routable? |
|---|---|---|---|---|---|---|---|
| 101 | 192.168.10.0 /24 | 192.168.10.1 | | | Routable addresses required:<br>- 3 IP addresses for Controllers<br>- 1 VIP for master Controller Node<br>- 4:10 Floating IP Addresses per VNF assigned to management VMs (CF, VNFM, UEM, and UAS software modules)<br>- 1 IP for default gateway | *network. yaml* and/or *layout.yaml*** | Yes |
| **Provisioning** | | | | | | | |
| 105 | **192.0.0.0/ 8** | | **192.200.0. 100** | **192.200.0. 254** | Required to provision all configuration via PXE boot from OSP-D for Ceph, Controller and Compute. Intel-On-Board Port 1 (1G). | *undercloud. conf* | No |
| **IPMI-CIMC** | | | | | | | |
| 105 | **192.0.0.0/ 8** | | **192.100.0. 100** | **192.100.0. 254** | | On UCS servers through CIMC | No |
| **Tenant (Virtio)** | | | | | | | |

| VLAN ID / Range | Network | Gateway | IP Range Start | IP Range End | Description | Where Configured | Routable? |
|---|---|---|---|---|---|---|---|
| 17 | 11.17.0.0/ 24 | | | | All Virtio based tenant networks. (MLOM) | *network. yaml* and/or *layout.yaml*** | No |
| **Storage (Virtio)** | | | | | | | |
| 18 | 11.18.0.0/ 24 | | | | Required for Controllers, Computes and Ceph for read/write from and to Ceph. (MLOM) | *network. yaml* and/or *layout.yaml*** | No |
| **Storage-MGMT (Virtio)** | | | | | | | |
| 19 | 11.19.0.0/ 24 | | | | Required for Controllers and Ceph only as Storage Cluster internal network. (MLOM) | *network. yaml* and/or *layout.yaml*** | No |
| **Internal-API (Virtio)** | | | | | | | |
| 20 | 11.20.0.0/ 24 | | | | Required for Controllers and Computes for openstack manageability. (MLOM) | *network. yaml* and/or *layout.yaml*** | No |
| **Mgmt (Virtio)** | | | | | | | |
| 21 | 172.16.181.0/ 24 | | 172.16.181. 100 | 172.16.181. 254 | Tenant based virtio network on openstack. | *network. yaml* and/or *layout.yaml*** | No |
| **Other-Virtio** | | | | | | | |
| 1001: 1500 | | | | | Tenant based virtio networks on openstack. | *network. yaml* and/or *layout.yaml*** | No |

| VLAN ID / Range | Network | Gateway | IP Range Start | IP Range End | Description | Where Configured | Routable? |
|---|---|---|---|---|---|---|---|
| **SR-IOV (Phys-PCIe1)** | | | | | | | |
| **2101:**<br>**2500** | | | | | Tenant SRIOV network on openstack. (Intel NIC on PCIe1) | *network. yaml* and/or *layout.yaml*** | Yes |
| **SR-IOV (Phys-PCIe4)** | | | | | | | |
| **2501:**<br>**2900** | | | | | Tenant SRIOV network on openstack. (Intel NIC on PCIe4) | *network. yaml* and/or *layout.yaml*** | Yes |

**NOTE:** **Bold underlined** text is provided as example configuration information. Your deployment requirements will vary. The IP addresses in **bold** text are the recommended address used for internal routing between VNF components. All other IP addresses and VLAN IDs may be changed/assigned.

* You can ensure that the same floating IP address can assigned to the AutoVNF, CF, UEM, and VNFM after a VM restart by configuring parameters in the AutoDeploy configuration file or the UWS service delivery configuration file. Refer to Table 32: Floating IP address Reuse Parameters, on page 86 for details.

** For Hyper-converged Ultra M models based on OpenStack 10, these parameters must configured in the both the *networks.yaml* and the *layout.yaml* files unless the VIM installation automation feature is used. Refer to the *Ultra Services Platform Deployment Automation Guide* for details.

**Caution** IP address ranges used for the Tenant (Virtio), Storage (Virtio), and Internal-API (Virtio) in *network.yaml* cannot conflict with the IP addresses specified in *layout.yaml* for the corresponding networks. Address conflicts will prevent the VNF from functioning properly.

*Table 32: Floating IP address Reuse Parameters*

| Component | Construct | AutoDeploy Configuration File Parameters | UWS Service Deployment Configuration File |
|---|---|---|---|
| AutoVNF | autovnfd | **networks management floating-ip true**<br>**networks management ha-vip**<*vip_address*><br>**networks management floating-ip-address** <*floating_address*> | <management><br><---SNIP---><br>**<floating-ip>true </floating-ip>**<br>**<ha-vip>** *vip_address***</ha-vip>**<br>**<floating-ip-address>** *floating_address* **</floating-ip-address>**<br></management> |

| Component | Construct | AutoDeploy Configuration File Parameters | UWS Service Deployment Configuration File |
|---|---|---|---|
| VNFM | vnfmd | **floating-ip true**<br>**ha-vip** *<vip_address>*<br>**floating-ip-address** *<floating_address>* | <management><br><---SNIP---><br>**<floating-ip>true </floating-ip>**<br>**<ha-vip>** *vip_address***</ha-vip>**<br>**<floating-ip-address>***floating_address* **</floating-ip-address>**<br></management> |
| UEM | vnfd | **vnf-em ha-vip** *<vip_address>*<br>**vnf-em floating-ip true**<br>**vnf-em floating-ip-address** *<floating_address>* | <vnf-em><br><---SNIP---><br>**<ha-vip>** *vip_address***</ha-vip>**<br><---SNIP---><br>**<floating-ip>true </floating-ip>**<br>**<floating-ip-address>** *floating_address* **</floating-ip-address>**<br><---SNIP---><br></vnf-em> |
| CF | vnfd | interfaces mgmt<br><---SNIP---><br>**enable-ha-vip** *<vip_address>*<br>**floating-ip true**<br>**floating-ip-address** *<floating_address>*<br><---SNIP---> | <interfaces><br><---SNIP---><br>**<enable-ha-vip>** *vip_address***</enable-ha-vip>**<br>**<floating-ip>true </floating-ip>**<br>**<floating-ip-address>** *floating_address* **</floating-ip-address>**<br><---SNIP---><br></interfaces> |

**Note** This functionality is disabled by default. Set the **floating-ip** and/or **<floating-ip>** parameters to *true* to enable this functionality.

**Note** Prior to assigning floating and virtual IP addresses, make sure that they are not already allocated through OpenStack. If the addresses are already allocated, then they must be freed up for use or you must assign a new IP address that is available in the VIM.

APPENDIX **B**

# Example ultram_cfg.yaml File

The *ultram_cfg.yaml* file is used to configure and enable syslog proxy and event aggregation functionality within the Ultra M Manager function. Refer to Event and Syslog Management Within the Ultra M Solution, on page 61 for details.

> **Caution** This is only a sample configuration file provided solely for your reference. You must create and modify your own configuration file according to the specific needs of your deployment.

```
#---------------------------------------------------------------------
# Configuration data for Ultra-M Health Check
#---------------------------------------------------------------------

# Health check polling frequency 15min
# In order to ensure optimal performance, it is strongly recommended that
# you do not change the default polling-interval of 15 minutes (900 seconds).
polling-interval: 900

# under-cloud info, this is used to authenticate
# OSPD and mostly used to build inventory list (compute, controllers, OSDs)
under-cloud:
  environment:
    OS_AUTH_URL: http://192.200.0.1:5000/v2.0
    OS_USERNAME: admin
    OS_TENANT_NAME: admin
    OS_PASSWORD: *******
  prefix: neutonoc


# over-cloud info, to authenticate OpenStack Keystone endpoint
over-cloud:
  enabled: true
  environment:
    OS_AUTH_URL: http://172.21.201.217:5000/v2.0
    OS_TENANT_NAME: user1
    OS_USERNAME: user1
    OS_PASSWORD: *******
    OS_ENDPOINT_TYPE: publicURL
    OS_IDENTITY_API_VERSION: 2
    OS_REGION_NAME: regionOne
  modules:
    - ceph
    - cinder
    - nova
    - pcs
    - rabbitmqctl
    - neutron
    - ntpdc
    - systemctl
  controller-services:
```

```
    - aodh
    - ceilometer
    - cinder
    - glance
    - gnocchi
    - heat
    - nova
    - swift
    - ntpd
    - mongod
    - memcached
    - neutron-dhcp-agent
    - neutron-l3-agent
    - neutron-metadata-agent
    - neutron-openvswitch-agent
    - neutron-server
    - httpd
  compute-services:
    - ceph-mon.target
    - ceph-radosgw.target
    - ceph.target
    - openvswitch.service
    - neutron-sriov-nic-agent
    - neutron-openvswitch-agent
    - ntpd
    - nova-compute
    - libvirtd
  osd-compute-services:
    - ceph-mon.target
    - ceph-radosgw.target
    - ceph.target
    - openvswitch.service
    - neutron-sriov-nic-agent
    - neutron-openvswitch-agent
    - ntpd
    - nova-compute
    - libvirtd

  # SSH Key to be used to login without username/password
  auth-key: /home/stack/.ssh/id_rsa

  # Number of OpenStack controller nodes
  controller_count:  3

  # Number of osd-compute nodes
  osd_compute_count:  3

  # Number of OSD disks per osd-compute node
  osd_disk_count_per_osd_compute: 4

  # Mark "ceph df" down if raw usage exceeds this setting
  ceph_df_use_threshold: 80.0

  # Max NTP skew limit in miliseconds
  ntp_skew_limit: 100

snmp:
  severity: 5
  nms-server:
    10.105.248.149:
      community: public
    10.105.248.149:
      user:
        name: test
        auth-protocol: md5
        auth-key: admin12345
        priv-protocol: cbc-des
        priv-key: admin12345
  agent:
    community: public

ucs-cluster:
  enabled: true
```

```
  user: admin
  password: *******
  data-dir: '/opt/cisco/usp/ultram_health.data/ucs'
  log-file: '/var/log/cisco/ultram_ucs.log'


uas-cluster:
  enabled: false
  log-file: '/var/log/cisco/ultram_uas.log'
  data-dir: '/opt/cisco/usp/ultram_health.data/uas'
  autovnf:
    172.21.201.53:
      autovnf:
        login:
          user: ubuntu
          password: *******
        netconf:
          user: admin
          password: admin
      em:
        login:
          user: ubuntu
          password: *******
        netconf:
          user: admin
          password: *******
      esc:
        login:
          user: admin
          password: *******
    172.21.201.54:
      autovnf:
        login:
          user: ubuntu
          password: *******
        netconf:
          user: admin
          password: *******
      em:
        login:
          user: ubuntu
          password: *******
        netconf:
          user: admin
          password: *******
      esc:
        login:
          user: admin
          password: *******


#rsyslog configuration, here proxy-rsyslog is IP address of Ultra M Manager Node (NOT
remote rsyslog):
rsyslog:
  level: 4,3,2,1,0
  proxy-rsyslog: 192.200.0.251
```

APPENDIX C

# Ultra M MIB

**Note** Not all aspects of this MIB are supported in this release. Refer to Event and Syslog Management Within the Ultra M Solution, on page 61 for information on the capabilities supported in this release.

```
-- *****************************************************************
-- CISCO-ULTRAM-MIB.my
-- Copyright (c) 2017 by Cisco Systems Inc.
-- All rights reserved.
--
-- *****************************************************************
CISCO-ULTRAM-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY
    OBJECT-TYPE
    NOTIFICATION-TYPE
    Unsigned32
        FROM SNMPv2-SMI
    MODULE-COMPLIANCE
    NOTIFICATION-GROUP
    OBJECT-GROUP
        FROM SNMPv2-CONF
    TEXTUAL-CONVENTION
    DateAndTime
        FROM SNMPv2-TC
    ciscoMgmt
        FROM CISCO-SMI;
ciscoUltramMIB MODULE-IDENTITY
    LAST-UPDATED    "201707060000Z"
    ORGANIZATION    "Cisco Systems Inc."
    CONTACT-INFO
            "Cisco Systems
            Customer Service
            Postal: 170 W Tasman Drive
            San Jose CA  95134
            USA
            Tel: +1 800 553-NETS"
    DESCRIPTION
        "The MIB module to management of Cisco Ultra Services Platform
        (USP) also called Ultra-M Network Function Virtualization (NFV)
        platform. The Ultra-M platform is Cisco validated turnkey
        solution based on ETSI(European Telecommunications Standards
        Institute) NFV architetcure.
        It comprises of following architectural domains:
         1. Management and Orchestration (MANO) these componets
        enables infrastructure virtualization and life cycle management
        of Cisco Ultra Virtual Network Functions (VNFs).
         2. NFV Infrastructure (NFVI) set of physical resources to
        provide NFV infrastructre for example servers switch chassis
```

```
        and so on.
         3. Virtualized Infrastructure Manager (VIM)
         4. One or more Ultra VNFs.
        Ultra-M platform provides a single point of management
        (including SNMP APIs Web Console and CLI/Telnet Console) for
        the resources across these domains within NFV PoD (Point of
        Delivery).
        This is also called Ultra-M manager throughout the context of
        this MIB."
    REVISION        "201707050000Z"
    DESCRIPTION
        "- cultramFaultDomain changed to read-only in compliance.
        - Added a new fault code serviceFailure under
        'CultramFaultCode'.
        - Added a new notification cultramFaultClearNotif.
        - Added new notification group ciscoUltramMIBNotifyGroupExt.
        - Added new compliance group ciscoUltramMIBModuleComplianceRev01
        which deprecates ciscoUltramMIBModuleCompliance."
    REVISION        "201706260000Z"
    DESCRIPTION
        "Latest version of this MIB module."
    ::= { ciscoMgmt 849 }
CFaultCode ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        "A code identifying a class of fault."
    SYNTAX          INTEGER  {
                        other(1) -- Other events
                        networkConnectivity(2) -- Network Connectivity
                                               -- Failure Events.
                        resourceUsage(3) -- Resource Usage Exhausted
                                         -- Event.
                        resourceThreshold(4) -- Resource Threshold
                                             -- crossing alarms
                        hardwareFailure(5) -- Hardware Failure Events
                        securityViolation(6) -- Security Alerts
                        configuration(7) -- Config Error Events
                        serviceFailure(8) -- Process/Service failures

                    }
CFaultSeverity ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        "A code used to identify the severity of a fault."
    SYNTAX          INTEGER  {
                        emergency(1) -- System level FAULT impacting
                                     -- multiple VNFs/Services
                        critical(2) -- Critical Fault specific to
                                    -- VNF/Service
                        major(3) -- component level failure within
                                 -- VNF/service.
                        alert(4) -- warning condition for a service/VNF
                                 -- may eventually impact service.
                        informational(5) -- informational only does not
                                         -- impact service
                    }
CFaultDomain ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        "A code used to categorize Ultra-M fault domain."
    SYNTAX          INTEGER  {
                        hardware(1) -- Harware including Servers L2/L3
                                    -- Elements
                        vimOrchestrator(2) -- VIM under-cloud
                        vim(3) -- VIM manager such as OpenStack
                        uas(4) -- Ultra Automation Services Modules
                        vnfm(5) -- VNF manager
                        vnfEM(6) -- Ultra VNF Element Manager
                        vnf(7) -- Ultra VNF
                    }
-- Textual Conventions definition will be defined before this line
ciscoUltramMIBNotifs  OBJECT IDENTIFIER
    ::= { ciscoUltramMIB 0 }
```

```
ciscoUltramMIBObjects  OBJECT IDENTIFIER
    ::= { ciscoUltramMIB 1 }
ciscoUltramMIBConform  OBJECT IDENTIFIER
    ::= { ciscoUltramMIB 2 }
-- Conformance Information Definition
ciscoUltramMIBCompliances  OBJECT IDENTIFIER
    ::= { ciscoUltramMIBConform 1 }
ciscoUltramMIBGroups  OBJECT IDENTIFIER
    ::= { ciscoUltramMIBConform 2 }
ciscoUltramMIBModuleCompliance MODULE-COMPLIANCE
    STATUS          deprecated
    DESCRIPTION
        "The compliance statement for entities that support
        the Cisco Ultra-M Fault Managed Objects"
    MODULE          -- this module
    MANDATORY-GROUPS {
                        ciscoUltramMIBMainObjectGroup
                        ciscoUltramMIBNotifyGroup
                    }
    ::= { ciscoUltramMIBCompliances 1 }
ciscoUltramMIBModuleComplianceRev01 MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION
        "The compliance statement for entities that support
        the Cisco Ultra-M Fault Managed Objects."
    MODULE          -- this module
    MANDATORY-GROUPS {
                        ciscoUltramMIBMainObjectGroup
                        ciscoUltramMIBNotifyGroup
                        ciscoUltramMIBNotifyGroupExt
                    }
    OBJECT          cultramFaultDomain
    MIN-ACCESS      read-only
    DESCRIPTION
        "cultramFaultDomain is read-only."
    ::= { ciscoUltramMIBCompliances 2 }
ciscoUltramMIBMainObjectGroup OBJECT-GROUP
    OBJECTS         {
                        cultramNFVIdenity
                        cultramFaultDomain
                        cultramFaultSource
                        cultramFaultCreationTime
                        cultramFaultSeverity
                        cultramFaultCode
                        cultramFaultDescription
                    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing Ultra-M fault information."
    ::= { ciscoUltramMIBGroups 1 }
ciscoUltramMIBNotifyGroup NOTIFICATION-GROUP
   NOTIFICATIONS    { cultramFaultActiveNotif }
    STATUS          current
    DESCRIPTION
        "The set of Ultra-M notifications defined by this MIB"
    ::= { ciscoUltramMIBGroups 2 }
ciscoUltramMIBNotifyGroupExt NOTIFICATION-GROUP
   NOTIFICATIONS    { cultramFaultClearNotif }
    STATUS          current
    DESCRIPTION
        "The set of Ultra-M notifications defined by this MIB"
    ::= { ciscoUltramMIBGroups 3 }
cultramFaultTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF CUltramFaultEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A table of Ultra-M faults. This table contains active
        faults."
    ::= { ciscoUltramMIBObjects 1 }
cultramFaultEntry OBJECT-TYPE
    SYNTAX          CUltramFaultEntry
    MAX-ACCESS      not-accessible
```

```
    STATUS          current
    DESCRIPTION
        "An entry in the Ultra-M fault table."
    INDEX           { cultramFaultIndex }
    ::= { cultramFaultTable 1 }
CUltramFaultEntry ::= SEQUENCE {
        cultramFaultIndex        Unsigned32
        cultramNFVIdenity        OCTET STRING
        cultramFaultDomain       CFaultDomain
        cultramFaultSource       OCTET STRING
        cultramFaultCreationTime DateAndTime
        cultramFaultSeverity     CFaultSeverity
        cultramFaultCode         CFaultCode
        cultramFaultDescription  OCTET STRING
}
cultramFaultIndex OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object uniquely identifies a specific instance of a
        Ultra-M fault.
        For example if two separate computes have a service level
        Failure then each compute will have a fault instance with a
        unique index."
    ::= { cultramFaultEntry 1 }
cultramNFVIdenity OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE  (1..512))
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This object uniquely identifies the Ultra-M PoD on which this
        fault is occurring.
        For example this identity can include host-name as well
        management IP where manager node is running
        'Ultra-M-San-Francisco/172.10.185.100'."
    ::= { cultramFaultEntry 2 }
cultramFaultDomain OBJECT-TYPE
    SYNTAX          CFaultDomain
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "A unique Fault Domain that has fault."
    ::= { cultramFaultEntry 3 }
cultramFaultSource OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE  (1..512))
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object uniquely identifies the resource with the fault
        domain where this fault is occurring. For example this can
        include host-name as well management IP of the resource
        'UCS-C240-Server-1/192.100.0.1'."
    ::= { cultramFaultEntry 4 }
cultramFaultCreationTime OBJECT-TYPE
    SYNTAX          DateAndTime
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The date and time when the fault was occured."
    ::= { cultramFaultEntry 5 }
cultramFaultSeverity OBJECT-TYPE
    SYNTAX          CFaultSeverity
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "A code identifying the perceived severity of the fault."
    ::= { cultramFaultEntry 6 }
cultramFaultCode OBJECT-TYPE
    SYNTAX          CFaultCode
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
```

```
        "A code uniquely identifying the fault class."
    ::= { cultramFaultEntry 7 }
cultramFaultDescription OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE  (1..2048))
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "A human-readable message providing details about the fault."
    ::= { cultramFaultEntry 8 }
cultramFaultActiveNotif NOTIFICATION-TYPE
    OBJECTS         {
                        cultramNFVIdenity
                        cultramFaultDomain
                        cultramFaultSource
                        cultramFaultCreationTime
                        cultramFaultSeverity
                        cultramFaultCode
                        cultramFaultDescription
                    }
    STATUS          current
    DESCRIPTION
        "This notification is generated by a Ultra-M manager whenever a
        fault is active."
   ::= { ciscoUltramMIBNotifs 1 }
cultramFaultClearNotif NOTIFICATION-TYPE
    OBJECTS         {
                        cultramNFVIdenity
                        cultramFaultDomain
                        cultramFaultSource
                        cultramFaultCreationTime
                        cultramFaultSeverity
                        cultramFaultCode
                        cultramFaultDescription
                    }
    STATUS          current
    DESCRIPTION
        "This notification is generated by a Ultra-M manager whenever a
        fault is cleared."
   ::= { ciscoUltramMIBNotifs 2 }
END
```

APPENDIX **D**

# Ultra M Component Event Severity and Fault Code Mappings

Events are assigned to one of the following severities (refer to CFaultSeverity in Ultra M MIB, on page 93):

- emergency(1), -- System level FAULT impacting multiple VNFs/Services
- critical(2), -- Critical Fault specific to VNF/Service
- major(3), -- component level failure within VNF/service.
- alert(4), -- warning condition for a service/VNF, may eventually impact service.
- informational(5) -- informational only, does not impact service

Events are also mapped to one of the following fault codes (refer to cFaultCode in the Ultra M MIB):

- other(1), -- Other events
- networkConnectivity(2), -- Network Connectivity -- Failure Events.
- resourceUsage(3), -- Resource Usage Exhausted -- Event.
- resourceThreshold(4), -- Resource Threshold -- crossing alarms
- hardwareFailure(5), -- Hardware Failure Events
- securityViolation(6), -- Security Alerts
- configuration(7), -- Config Error Events serviceFailure(8) -- Process/Service failures

The Ultra M Manager Node serves as an aggregator for events received from the different Ultra M components. These severities and fault codes are mapped to those defined for the specific components. The information in this section provides severity mapping information for the following:

# OpenStack Events

## Component: Ceph

*Table 33: Component: Ceph*

| Failure Type | Ultra M Severity | Fault Code |
|---|---|---|
| CEPH Status is not healthy | Emergency | serviceFailure |
| One or more CEPH monitors are down | Emergency | serviceFailure |
| Disk usage exceeds threshold | Critical | resourceThreshold |
| One or more OSD nodes are down | Critical | serviceFailure |
| One or more OSD disks are failed | Critical | resourceThreshold |
| One of the CEPH monitor is not healthy. | Major | serviceFailure |
| One or more CEPH monitor restarted. | Major | serviceFailure |
| OSD disk weights not even across the board. | | resourceThreshold |

## Component: Cinder

*Table 34: Component: Cinder*

| Failure Type | Ultra M Severity | Fault Code |
|---|---|---|
| Cinder Service is down | Emergency | serviceFailure |

# Component: Neutron

*Table 35: Component: Neutron*

| Failure Type | Ultra M Severity | Fault Code |
|---|---|---|
| One of Neutron Agent Down | Critical | serviceFailure |

# Component: Nova

*Table 36: Component: Nova*

| Failure Type | Ultra M Severity | Fault Code |
|---|---|---|
| Compute service down | Critical | serviceFailure |

# Component: NTP

*Table 37: Component: NTP*

| Failure Type | Ultra M Severity | Fault Code |
|---|---|---|
| NTP skew limit exceeds configured threshold. | Critical | serviceFailure |

# Component: PCS

*Table 38: Component: PCS*

| Failure Type | Ultra M Severity | Fault Code |
|---|---|---|
| One or more controller nodes are down | Critical | serviceFailure |
| Ha-proxy is down on one of the node | Major | serviceFailure |
| Galera service is down on one of the node. | Critical | serviceFailure |

| Failure Type | Ultra M Severity | Fault Code |
|---|---|---|
| Rabbitmq is down. | Critical | serviceFailure |
| Radis Master is down. | Emergency | serviceFailure |
| One or more Radis Slaves are down. | Critical | serviceFailure |
| corosync/pacemaker/pcsd - not all daemons active | Critical | serviceFailure |
| Cluster status changed. | Major | serviceFailure |
| Current DC not found. | Emergency | serviceFailure |
| Not all PCDs are online. | Critical | serviceFailure |

# Component: Rabbitmqctl

***Table 39: Component: Rabbitmqctl***

| Failure Type | Ultra M Severity | Fault Code |
|---|---|---|
| Cluster Status is not healthy | Emergency | serviceFailure |

# Component: Services

***Table 40: Component: Services***

| Failure Type | Ultra M Severity | Fault Code |
|---|---|---|
| Service is disabled. | Critical | serviceFailure |
| Service is down. | Emergency | serviceFailure |
| Service Restarted. | Major | serviceFailure |

The following OpenStack services are monitored:

- Controller Nodes:
    - httpd.service
    - memcached

- mongod.service
- neutron-dhcp-agent.service
- neutron-l3-agent.service
- neutron-metadata-agent.service
- neutron-openvswitch-agent.service
- neutron-server.service
- ntpd.service
- openstack-aodh-evaluator.service
- openstack-aodh-listener.service
- openstack-aodh-notifier.service
- openstack-ceilometer-central.service
- openstack-ceilometer-collector.service
- openstack-ceilometer-notification.service
- openstack-cinder-api.service
- openstack-cinder-scheduler.service
- openstack-glance-api.service
- openstack-glance-registry.service
- openstack-gnocchi-metricd.service
- openstack-gnocchi-statsd.service
- openstack-heat-api-cfn.service
- openstack-heat-api-cloudwatch.service
- openstack-heat-api.service
- openstack-heat-engine.service
- openstack-nova-api.service
- openstack-nova-conductor.service
- openstack-nova-consoleauth.service
- openstack-nova-novncproxy.service
- openstack-nova-scheduler.service
- openstack-swift-account-auditor.service
- openstack-swift-account-reaper.service
- openstack-swift-account-replicator.service
- openstack-swift-account.service
- openstack-swift-container-auditor.service

    - openstack-swift-container-replicator.service
    - openstack-swift-container-updater.service
    - openstack-swift-container.service
    - openstack-swift-object-auditor.service
    - openstack-swift-object-replicator.service
    - openstack-swift-object-updater.service
    - openstack-swift-object.service
    - openstack-swift-proxy.service

- Compute Nodes:
    - ceph-mon.target
    - ceph-radosgw.target
    - ceph.target
    - libvirtd.service
    - neutron-sriov-nic-agent.service
    - neutron-openvswitch-agent.service
    - ntpd.service
    - openstack-nova-compute.service
    - openvswitch.service

- OSD Compute Nodes:
    - ceph-mon.target
    - ceph-radosgw.target
    - ceph.target
    - libvirtd.service
    - neutron-sriov-nic-agent.service
    - neutron-openvswitch-agent.service
    - ntpd.service
    - openstack-nova-compute.service
    - openvswitch.service

# UCS Server Events

UCS Server events are described here:https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/ts/faults/reference/ErrMess/FaultsIntroduction.html

The following table maps the UCS severities to those within the Ultra M MIB.

*Table 41: UCS Server Severities*

| UCS Server Severity | Ultra M Severity | Fault Code |
|---|---|---|
| Critical | Critical | hardwareFailure |
| Info | Informational | hardwareFailure |
| Major | Major | hardwareFailure |
| Warning | Alert | hardwareFailure |
| Alert | Alert | hardwareFailure |
| Cleared | Informational | Not applicable |

# UAS Events

*Table 42: UAS Events*

| Failure Type | Ultra M Severity | Fault Code |
|---|---|---|
| UAS Service Failure | Critical | serviceFailure* |
| UAS Service Recovered | Informational | serviceFailure* |
| * *serviceFailure* is used except where the Ultra M Health Monitor is unable to connect to any of the modules. In this case, the fault code is set to *networkConnectivity*. | | |

# ESC VM Events

By default, the Ultra M Manager continuously monitors and process VNF VM event notifications from ESC as reported through NETCONF.

*Table 43: ESC VM Event Severities*

| ESC Event | Ultra M Severity | Fault Code | VNFR State |
|---|---|---|---|
| VM_DEPLOYED | Major | Service Failure | deployed |
| VM_ALIVE | Info | Other | alive |

| ESC Event | Ultra M Severity | Fault Code | VNFR State |
|---|---|---|---|
| VM_UNDEPLOYED | Critical | Service Failure | offline |
| VM_REBOOTED | Major | Service Failure | rebooting |
| VM_RECOVERY_REBOOT | Major | Service Failure | If the event completion status is successful, then it is rebooting.<br>If the event completion status is failure, then it is an error. |
| VM_RECOVERY_UNDEPLOYED | Critical | Service Failure | If the event completion status is successful, then it is offline.<br>If the event completion status is failure, then it is an error. |
| VM_RECOVERY_DEPLOYED | Major | Service Failure | If the event completion status is successful, then it is rebooting.<br>If the event completion status is failure, then it is an error. |
| VM_RECOVERY_COMPLETE | Info | Other | If the event completion status is successful, then it is alive.<br>If the event completion status is failure, then it is an error. |
| VM_STOPPED | Alert | Service Failure | stop_requested |

APPENDIX E

# Ultra M Troubleshooting



# Ultra M Component Reference Documentation

The following sections provide links to troubleshooting information for the various components that comprise the Ultra M solution.

## UCS C-Series Server

- Obtaining Showtech Support to TAC
- Display of system Event log events
- Display of CIMC Log
- Run Debug Firmware Utility
- Run Diagnostics CLI
- Common Troubleshooting Scenarios
- Troubleshooting Disk and Raid issues
- DIMM Memory Issues
- Troubleshooting Server and Memory Issues
- Troubleshooting Communication Issues

## Nexus 9000 Series Switch

- Troubleshooting Installations, Upgrades, and Reboots

- Troubleshooting Licensing Issues
- Troubleshooting Ports
- Troubleshooting vPCs
- Troubleshooting VLANs
- Troubleshooting STP
- Troubleshooting Routing
- Troubleshooting Memory
- Troubleshooting Packet Flow Issues
- Troubleshooting PowerOn Auto Provisioning
- Troubleshooting the Python API
- Troubleshooting NX-API
- Troubleshooting Service Failures
- Before Contacting Technical Support
- Troubleshooting Tools and Methodology

# Catalyst 2960 Switch

- Diagnosing Problems
- Switch POST Results
- Switch LEDs
- Switch Connections
- Bad or Damaged Cable
- Ethernet and Fiber-Optic Cables
- Link Status
- 10/100/1000 Port Connections
- 10/100/1000 PoE+ Port Connections
- SFP and SFP+ Module
- Interface Settings
- Ping End Device
- Spanning Tree Loops
- Switch Performance
- Speed, Duplex, and Autonegotiation
- Autonegotiation and Network Interface Cards
- Cabling Distance

- Clearing the Switch IP Address and Configuration
- Finding the Serial Number
- Replacing a Failed Stack Member

## Red Hat

- Troubleshooting Director issue
- Backup and Restore Director Undercloud

## OpenStack

- Red Hat Openstack Troubleshooting commands and sceanrios

## UAS

Refer to the *USP Deployment Automation Guide*.

## UGP

Refer to the *Ultra Gateway Platform System Administration Guide*.

# Collecting Support Information

## From UCS:

- Collect support information:

  **chassis show tech support**
  **show tech support** `(if applicable)`

- Check which UCS MIBS are being polled (if applicable). Refer to https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/sw/mib/c-series/b_UCS_Standalone_C-Series_MIBRef/b_UCS_Standalone_C-Series_MIBRef_chapter_0100.html

## From Host/Server/Compute/Controler/Linux:

- Identify if Passthrought/SR-IOV is enabled.
- Run sosreport:

> **Note** This functionality is enabled by default on Red Hat, but not on Ubuntu. It is recommended that you enable *sysstat* and *sosreport* on Ubuntu (run **apt-get install sysstat** and **apt-get install sosreport**). It is also recommended that you install sysstat on Red Hat (run **yum install sysstat**).

- Get and run the **os_ssd_pac** script from Cisco:
  - Compute (all):

    **./os_ssd_pac.sh -a**
    **./os_ssd_pac.sh -k -s**

    > **Note** For initial collection, it is always recommended to include the **-s** option (sosreport). Run **./os_ssd_pac.sh -h** for more information.

  - Controller (all):

    **./os_ssd_pac.sh -f**
    **./os_ssd_pac.sh -c -s**

    > **Note** For initial collection it is always recommended to include the -s option (sosreport). Run **./os_ssd_pac.sh -h** for more information.

- For monitoring purposes, from *crontab* use option: **-m** ( for example run every 5 or 10 minutes)

# From Switches

From all switches connected to the Host/Servers. (This also includes other switches which have same vlans terminated on the Host/Servers.)

**show tech-support**
**syslogs**
**snmp traps**

> **Note** It is recommended that mac-move notifications are enabled on all switches in network by running mac address-table notification mac-move.

# From ESC (Active and Standby)

**Note** It is recommended that you take a backup of the software and data before performing any of the following operations. Backups can be taken by executing opt/cisco/esc/esc-scripts/esc_dbtool.py backup. (Refer to https://www.cisco.com/c/en/us/td/docs/net_mgmt/elastic_services_controller/2-3/user/guide/Cisco-Elastic-Services-Controller-User-Guide-2-3/Cisco-Elastic-Services-Controller-User-Guide-2-2_chapter_010010.html#id_18936 for more information.)

**/opt/cisco/esc/esc-scripts/health.sh**
**/usr/bin/collect_esc_log.sh**
**./os_ssd_pac -a**

# From UAS

- Monitor ConfD:

  **confd -status**
  **confd --debug-dump /tmp/confd_debug-dump**
  **confd --printlog /tmp/confd_debug-dump**

  **Note** Once the file **/tmp/confd_debug-dump**> is collected, it can be removed (**rm /tmp/confd_debug-dump**).

- Monitor UAS Components:

  **source /opt/cisco/usp/uas/confd-6.1/confdrc**
  **confd_cli -u admin -C**
  **show uas**
  **show uas ha-vip**
  **show uas state**
  **show confd-state**
  **show running-config**
  **show transactions date-and-time**
  **show logs | display xml**
  **show errors displaylevel 64**
  **show notification stream uas_notify last 1000**
  **show autovnf-oper:vnfm**
  **show autovnf-oper:vnf-em**
  **show autovnf-oper:vdu-catalog**
  **show autovnf-oper:transactions**
  **show autovnf-oper:network-catalog**
  **show autovnf-oper:errors**
  **show usp**
  **show confd-state internal callpoints**
  **show confd-state webui listen**
  **show netconf-state**

- Monitor Zookeeper:

  **/opt/cisco/usp/packages/zookeeper/current/bin/zkCli.sh ls /config/control-function**
  **/opt/cisco/usp/packages/zookeeper/current/bin/zkCli.sh ls /config/element-manager**
  **/opt/cisco/usp/packages/zookeeper/current/bin/zkCli.sh ls /config/session-function**
  **/opt/cisco/usp/packages/zookeeper/current/bin/zkCli.sh ls /**
  **/opt/cisco/usp/packages/zookeeper/current/bin/zkCli.sh ls /stat**
  **/opt/cisco/usp/packages/zookeeper/current/bin/zkCli.sh ls /log**

- Collect Zookeeper data:

  **cd /tmp**
  **tar zcfv zookeeper_data.tgz /var/lib/zookeeper/data/version-2/**
  **ls -las /tmp/zookeeper_data.tgz**

- Get support details

  **./os_ssd_pac -a**

# From UEM (Active and Standby)

- Collect logs

  **/opt/cisco/em-scripts/collect-em-logs.sh**

- Monitor NCS:

  **ncs -status**
  **ncs --debug-dump /tmp/ncs_debug-dump**
  **ncs --printlog /tmp/ncs_debug-dump**

  **Note** Once the file **/tmp/ncs_debug-dump** is collected, it can be removed (**rm /tmp/ncs_debug-dump**).

- Collect support details:

  **./os_ssd_pac -a**

# From UGP (Through StarOS)

- Collect the multiple outputs of the **show support details**.

  **Note** It is recommended to collect at least two samples, 60 minutes apart if possible.

- Collect raw bulkstats before and after events.
- Collect syslogs and snmp traps before and after events.

- Collect PCAP or sniffer traces of all relevant interfaces if possible.

**Note** Familiarize yourself with how running SPAN/RSPAN on Nexus and Catalyst switches. This is important for resolving Passthrough/SR-IOV issues.

- Collect console outputs from all nodes.
- Export CDRs and EDRs.
- Collect the outputs of **monitor subscriber next-call** or **monitor protocol** depending on the activity
- Refer to https://supportforums.cisco.com/sites/default/files/cisco_asr5000_asr5500_troubleshooting_guide.pdf for more information.

# About Ultra M Manager Log Files

All Ultra M Manager log files are created under "/var/log/cisco/ultram-manager".

**cd /var/log/cisco/ultram-manager**
**ls -alrt**
Example output:

```
total 116
drwxr-xr-x. 3 root root  4096 Sep 10 17:41 ..
-rw-r--r--. 1 root root     0 Sep 12 15:15 ultram_health_snmp.log
-rw-r--r--. 1 root root   448 Sep 12 15:16 ultram_health_uas.report
-rw-r--r--. 1 root root   188 Sep 12 15:16 ultram_health_uas.error
-rw-r--r--. 1 root root   580 Sep 12 15:16 ultram_health_uas.log
-rw-r--r--. 1 root root 24093 Sep 12 15:16 ultram_health_ucs.log
-rw-r--r--. 1 root root  8302 Sep 12 15:16 ultram_health_os.error
drwxr-xr-x. 2 root root  4096 Sep 12 15:16 .
-rw-r--r--. 1 root root 51077 Sep 12 15:16 ultram_health_os.report
-rw-r--r--. 1 root root  6677 Sep 12 15:16 ultram_health_os.log
```

**NOTES:**

- The files are named according to the following conventions:
    - ultram_health_os: Contain information related to OpenStack
    - ultram_health_ucs: Contain information related to UCS
    - ultram_health_uas: Contain information related to UAS
- Files with the "*.log" extension contain debug/error outputs from different components. These files get added to over time and contain useful data for debugging in case of issues.
- Files with the ".report" extension contain the current report. These files get created on every tun.
- Files with the ".error" extension contain actual data received from the nodes as part of health monitoring. These are the events that causes the Ultra M health monitor to send traps out. These files are updated every time a component generates an event.

APPENDIX **F**

# Using the UCS Utilities Within the Ultra M Manager

This appendix describes the UCS facilities within the Ultra M Manager.



## Overview

Cisco UCS server BIOS, MLOM, and CIMC software updates may be made available from time to time.

Utilities have been added to the Ultra M Manager software to simplify the process of upgrading the UCS server software (firmware) within the Ultra M solution.

These utilities are available through a script called *ultram_ucs_utils.py* located in the */opt/cisco/usp/ultram-health* directory. Refer to ultram_ucs_utils.py Help, on page 135 for more information on this script.

**NOTES:**

- This functionality is currently supported only with Ultra M deployments based on OSP 10 and that leverage the Hyper-Converged architecture.

- You should only upgrade your UCS server software to versions that have been validated for use within the Ultra M solution.
- All UCS servers within the Ultra M solution stack should be upgraded to the same firmware versions.
- Though it is highly recommended that all server upgrades be performed during a single maintenance window, it is possible to perform the upgrade across multiple maintenance windows based on Node type (e.g. Compute, OSD Compute, and Controller).

There are two upgrade scenarios:

- **Upgrading servers in an existing deployment.** In the scenario, the servers are already in use hosting the Ultra M solution stack. This upgrade procedure is designed to maintain the integrity of the stack.
    - Compute Nodes are upgraded in parallel.
    - OSD Compute Nodes are upgraded sequentially.
    - Controller Nodes are upgraded sequentially.
- **Upgrading bare metal servers.** In this scenario, the bare metal servers have not yet been deployed within the Ultra M solution stack. This upgrade procedure leverages the parallel upgrade capability within Ultra M Manager UCS utilities to upgrade the servers in parallel.

To use UItra M Manager UCS utilities to upgrade software for UCS servers in an existing deployment:

1. Perform Pre-Upgrade Preparation.
2. Shutdown the ESC VMs, on page 120.
3. Upgrade the Compute Node Server Software.
4. Upgrade the OSD Compute Node Server Software, on page 122.
5. Restart the UAS and ESC (VNFM) VMs, on page 125.
6. Upgrade the Controller Node Server Software, on page 125.
7. Upgrade Firmware on the OSP-D Server/Ultra M Manager Node, on page 130.

To use UItra M Manager UCS utilities to upgrade software for bare metal UCS servers:

1. Perform Pre-Upgrade Preparation.
2. Upgrade Firmware on UCS Bare Metal, on page 128.
3. Upgrade Firmware on the OSP-D Server/Ultra M Manager Node, on page 130.

# Perform Pre-Upgrade Preparation

Prior to performing the actual UCS server software upgrade, you must perform the steps in this section to prepare your environment for the upgrade.

**NOTES:**

- These instructions assume that all hardware is fully installed, cabled, and operational.
- These instructions assume that the VIM Orchestrator and VIM have been successfully deployed.

- UCS server software is distributed separately from the USP software ISO.

To prepare your environment prior to upgrading the UCS server software:

1. Log on to the Ultra M Manager Node.
2. Create a directory called */var/www/html/firmwares* to contain the upgrade files.

   **mkdir -p /var/www/html/firmwares**
3. Download the UCS software ISO to the directory you just created.

   UCS software is available for download from https://software.cisco.com/download/type.html?mdfid=286281356&flowid=71443
4. Extract the *bios.cap* file.

   **mkdir /tmp/UCSISO**

   **sudo mount -t iso9660 -o loop ucs-c240m4-huu-**<*version*>**.iso UCSISO/**

   ```
   mount: /dev/loop2 is write-protected, mounting read-only
   ```

   **cd UCSISO/**

   **ls**

   ```
   EFI                    GETFW           isolinux  Release-Notes-DN2.txt  squashfs_img.md5
   tools.squashfs.enc
   firmware.squashfs.enc  huu-release.xml  LiveOS    squashfs_img.enc.md5   TOC_DELNORTE2.xml
     VIC_FIRMWARE
   ```

   **cd GETFW/**

   **ls**

   ```
   getfw  readme.txt
   ```

   **mkdir -p /tmp/HUU**

   **sudo ./getfw -s /tmp/ucs-c240m4-huu-**<*version*>**.iso -d /tmp/HUU**

   ```
   Nothing was selected hence getting only CIMC and BIOS
   FW/s available at '/tmp/HUU/ucs-c240m4-huu-<version>'
   ```

   **cd /tmp/HUU/ucs-c240m4-huu-**<*version*>**/bios/**

   **ls**

   ```
   bios.cap
   ```
5. Copy the *bios.cap* and *huu.iso* to the */var/www/html/firmwares/* directory.

   **sudo cp bios.cap /var/www/html/firmwares/**

   **ls -lrt /var/www/html/firmwares/**

   ```
   total 692228
   -rw-r--r--.  1 root root 692060160 Sep 28 22:43 ucs-c240m4-huu-<version>.iso
   -rwxr-xr-x.  1 root root  16779416 Sep 28 23:55 bios.cap
   ```
6. *Optional.* If it is not already installed, install the Ultra M Manager using the information and instructions in Install the Ultra M Manager RPM, on page 76.
7. Navigate to the */opt/cisco/usp/ultram-manager* directory.

   **cd /opt/cisco/usp/ultram-manager**

   Once this step is completed, if you are upgrading UCS servers in an existing Ultra M solution stack, proceed to 8, on page 118. If you are upgrading bare metal UCS servers, proceed to 9, on page 118.

8 *Optional.* If you are upgrading software for UCS servers in an existing Ultra M solution stack, then create UCS server node list configuration files for each node type as shown in the following table.

| Configuration File Name | File Contents |
|---|---|
| compute.cfg | A list of the CIMC IP addresses for all of the Compute Nodes. |
| osd_compute_0.cfg | The CIMC IP address of the primary OSD Compute Node (osd-compute-0). |
| osd_compute_1.cfg | The CIMC IP address of the second OSD Compute Node (osd-compute-1). |
| osd_compute_2.cfg | The CIMC IP address of the third OSD Compute Node (osd-compute-2). |
| controller_0.cfg | The CIMC IP address of the primary Controller Node (controller-0). |
| controller_1.cfg | The CIMC IP address of the second Controller Node (controller-1). |
| controller_2.cfg | The CIMC IP address of the third Controller Node (controller-2). |

**Note** Each address must be preceded by a dash and a space ("-"). The following is an example of the required format:

```
- 192.100.0.9
- 192.100.0.10
- 192.100.0.11
- 192.100.0.12
```

Separate configuration files are required for each OSD Compute and Controller Node in order to maintain the integrity of the Ultra M solution stack throughout the upgrade process.

9 *Optional.* If you are upgrading software on bare metal UCS servers prior to deploying them as part of the Ultra M solution stack, then create a configuration file called *hosts.cfg* containing a list of the CIMC IP addresses for all of the servers to be used within the Ultra M solution stack except the OSP-D server/Ultra M Manager Node.

**Note** Each address must be preceded by a dash and a space (-). The following is an example of the required format:

```
- 192.100.0.9
- 192.100.0.10
- 192.100.0.11
- 192.100.0.12
```

**10** Create a configuration file called *ospd.cfg* containing the CIMC IP address of the OSP-D Server/Ultra M Manager Node.

**Note** The address must be preceded by a dash and a space ("-"). The following is an example of the required format:

```
- 192.300.0.9
```

**11** Validate your configuration files by performing a sample test of the script to pull existing firmware versions from all Controller, OSD Compute, and Compute Nodes in your Ultra M solution deployment.

**./ultram_ucs_utils.py --cfg** "*<config_file_name>*" **--login** *<cimc_username>* *<cimc_user_password>* **--status 'firmwares'**

The following is an example output for a *hosts.cfg* file with a single Compute Node (192.100.0.7):

```
2017-10-01 10:36:28,189 - Successfully logged out from the server: 192.100.0.7
2017-10-01 10:36:28,190 -
----------------------------------------------------------------------------------------
Server IP      | Component                                       | Version
----------------------------------------------------------------------------------------
192.100.0.7    | bios/fw-boot-loader                            | C240M4.3.0.3c.0.0831170228

               | mgmt/fw-boot-loader                             | 3.0(3e).36
               | mgmt/fw-system                                  | 3.0(3e)
               | adaptor-MLOM/mgmt/fw-boot-loader                | 4.1(2d)
               | adaptor-MLOM/mgmt/fw-system                     | 4.1(3a)
               | board/storage-SAS-SLOT-HBA/fw-boot-loader       |
6.30.03.0_4.17.08.00_0xC6130202
               | board/storage-SAS-SLOT-HBA/fw-system            | 4.620.00-7259
               | sas-expander-1/mgmt/fw-system                   | 65104100
               | Intel(R) I350 1 Gbps Network Controller         | 0x80000E75-1.810.8
               | Intel X520-DA2 10 Gbps 2 port NIC               | 0x800008A4-1.810.8
               | Intel X520-DA2 10 Gbps 2 port NIC               | 0x800008A4-1.810.8
               | UCS VIC 1227 10Gbps 2 port CNA SFP+             | 4.1(3a)
               | Cisco 12G SAS Modular Raid Controller           | 24.12.1-0203
----------------------------------------------------------------------------------------
```

If you receive errors when executing the script, ensure that the CIMC username and password are correct. Additionally, verify that all of the IP addresses have been entered properly in the configuration files.

**Note** It is highly recommended that you save the data reported in the output for later reference and validation after performing the upgrades.

**12** Take backups of the various configuration files, logs, and other relevant information using the information and instructions in the *Backing Up Deployment Information* appendix in the *Ultra Services Platform Deployment Automation Guide*.

**13** Continue the upgrade process based on your deployment status.

- Proceed to Shutdown the ESC VMs, on page 120 if you are upgrading software for servers that were previously deployed as part of the Ultra M solution stack.
- Proceed to Upgrade Firmware on UCS Bare Metal, on page 128 if you are upgrading software for servers that have not yet been deployed as part of the Ultra M solution stack.

# Shutdown the ESC VMs

The Cisco Elastic Services Controller (ESC) serves as the VNFM in Ultra M solution deployments. ESC is deployed on a redundant pair of VMs. These VMs must be shut down prior to performing software upgrades on the UCS servers in the solution deployment.

To shut down the ESC VMs:

1. Login to OSP-D and make sure to "su - stack" and "source stackrc".
2. Run Nova list to get the UUIDs of the ESC VMs.

   **nova list --fields name,host,status | grep** <*vnf_deployment_name*>
   Example output:

```
<--- SNIP --->
| b470cfeb-20c6-4168-99f2-1592502c2057 | vnf1-ESC-ESC-
0                                               | tb5-ultram-osd-compute-2.localdomain |
 ACTIVE |
| 157d7bfb-1152-4138-b85f-79afa96ad97d | vnf1-ESC-ESC-
1                                               | tb5-ultram-osd-compute-1.localdomain |
 ACTIVE |
<--- SNIP --->
```

3. Stop the standby ESC VM.

   **nova stop** <*standby_vm_uuid*>
4. Stop the active ESC VM.

   **nova stop** <*active_vm_uuid*>
5. Verify that the VMs have been shutoff.

   **nova list --fields name,host,status | grep** <*vnf_deployment_name*>
   Look for the entries pertaining to the ESC UUIDs.

   Example output:

```
<--- SNIP --->
| b470cfeb-20c6-4168-99f2-1592502c2057 | vnf1-ESC-ESC-
0                                               | tb5-ultram-osd-compute-2.localdomain |
 SHUTOFF |
| 157d7bfb-1152-4138-b85f-79afa96ad97d | vnf1-ESC-ESC-
1                                               | tb5-ultram-osd-compute-1.localdomain |
 SHUTOFF |
<--- SNIP --->
```

6. Proceed to Upgrade the Compute Node Server Software, on page 120.

# Upgrade the Compute Node Server Software

**NOTES:**

- Ensure that the ESC VMs have been shutdown according to the procedure in Shutdown the ESC VMs, on page 120.
- This procedure assumes that you are already logged in to the Ultra M Manager Node.
- This procedure requires the *compute.cfg* file created as part of the procedure detailed in Perform Pre-Upgrade Preparation, on page 116.

- It is highly recommended that all Compute Nodes be upgraded using this process during a single maintenance window.

To upgrade the UCS server software on the Compute Nodes:

1 Upgrade the BIOS on the UCS server-based Compute Nodes.

**./ultram_ucs_utils.py --cfg "compute.cfg" --login** <*cimc_username*> <*cimc_user_password*> **--upgrade bios --server** <*ospd_server_cimc_ip_address*> **--timeout 30 --file /firmwares/bios.cap**
Example output:

```
2017-09-29 09:15:48,753 - Updating BIOS firmware on all the servers
2017-09-29 09:15:48,753 - Logging on UCS Server: 192.100.0.7
2017-09-29 09:15:48,758 - No session found, creating one on server: 192.100.0.7
2017-09-29 09:15:50,194 - Login successful to server: 192.100.0.7
2017-09-29 09:16:13,269 - 192.100.0.7 => updating | Image Download (5 %), OK
2017-09-29 09:17:26,669 - 192.100.0.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:18:34,524 - 192.100.0.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:19:40,892 - 192.100.0.7 => Activating BIOS
2017-09-29 09:19:55,011 -
---------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
---------------------------------------------------------------------
192.100.0.7     | SUCCESS | NA                  | Status: success, Progress: Done, OK
```

**Note** The Compute Nodes are automatically powered down after this process leaving only the CIMC interface available.

2 Upgrade the UCS server using the Host Upgrade Utility (HUU).

**./ultram_ucs_utils.py --cfg "compute.cfg" --login** <*cimc_username*> <*cimc_user_password*> **--upgrade huu --server** <*ospd_server_cimc_ip_address*> **--file /firmwares/**<*ucs_huu_iso_filename*>
If the HUU script times out before completing the upgrade, the process might still be running on the remote hosts. You can periodically check the upgrade process by entering:

**./ultram_ucs_utils.py --cfg "compute.cfg" --login** <*cimc_username*> <*cimc_user_password*> **--status huu-upgrade**
Example output:

```
---------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
---------------------------------------------------------------------
192.100.0.7    | SUCCESS | 2017-10-20 07:10:11 | Update Complete  CIMC Completed, SasExpDN
 Completed, I350 Completed, X520 Completed, X520 Completed, 3108AB-8i Completed, UCS VIC
 1227 Completed, BIOS Completed,
---------------------------------------------------------------------
```

3 Verify that the BIOS firmware and HUU upgrade was successful by checking the post-upgrade versions.

**./ultram_ucs_utils.py --cfg "compute.cfg" --login** <*cimc_username*> <*cimc_user_password*> **--status firmwares**

4 Set the package-c-state-limit CIMC setting.

**./ultram_ucs_utils.py --mgmt set-bios --bios-param biosVfPackageCStateLimit --bios-values vpPackageC-StateLimit=C0/C1 --cfg compute.cfg --login**<*cimc_username*> <*cimc_user_password*>

5 Verify that the package-c-state-limit CIMC setting has been made.

**./ultram_ucs_utils.py --status bios-settings --cfg compute.cfg --login** <*cimc_username*> <*cimc_user_password*>
Look for **PackageCStateLimit** to be set to *C0/C1*.

6 Modify the Grub configuration on each Compute Node.

a Log into your first compute (compute-0) and update the grub setting with "processor.max_cstate=0 intel_idle.max_cstate=0".

**sudo grubby --info=/boot/vmlinuz-`uname -r`**
**sudo grubby --update-kernel=/boot/vmlinuz-`uname -r` --args="processor.max_cstate=0 intel_idle.max_cstate=0"**

b Verify that the update was successful.

**sudo grubby --info=/boot/vmlinuz-`uname -r`**
Look for the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments in the output.

c Reboot the Compute Nodes.

**sudo reboot**

d Repeat steps 6.a, on page 122 through 6.c, on page 122 for all other Compute Nodes.

7 Recheck all CIMC and kernel settings.

a Log in to the Ultra M Manager Node.

b Verify CIMC settings

**./ultram_ucs_utils.py --status bios-settings --cfg compute.cfg --login***<cimc_username> <cimc_user_password>*

c Verify the processor c-state.

**for ip in `nova list | grep -i compute | awk '{print $12}' | sed 's/ctlplane=//g'`; do ssh heat-admin@$ip 'sudo cat /sys/module/intel_idle/parameters/max_cstate'; done**
**for ip in `nova list | grep -i compute | awk '{print $12}' | sed 's/ctlplane=//g'`; do ssh heat-admin@$ip 'sudo cpupower idle-info'; done**

8 Proceed to Upgrade the OSD Compute Node Server Software.

**Note** Other Node types can be upgraded at a later time. If you'll be upgrading them during a later maintenance window, proceed to Restart the UAS and ESC (VNFM) VMs, on page 125.

# Upgrade the OSD Compute Node Server Software

**NOTES:**

- This procedure requires the *osd_compute_0.cfg*, *osd_compute_1.cfg*, and *osd_compute_2.cfg* files created as part of the procedure detailed in Perform Pre-Upgrade Preparation, on page 116.
- It is highly recommended that all OSD Compute Nodes be upgraded using this process during a single maintenance window.

To upgrade the UCS server software on the OSD Compute Nodes:

1 Move the Ceph storage to maintenance mode.

a Log on to the lead Controller Node (controller-0).

b Move the Ceph storage to maintenance mode.

**sudo ceph status**
**sudo ceph osd set noout**
**sudo ceph osd set norebalance**
**sudo ceph status**

2 *Optional.* If they've not already been shut down, shut down both ESC VMs using the instructions in Shutdown the ESC VMs, on page 120.

3 Log on to the Ultra M Manager Node.

4 Upgrade the BIOS on the initial UCS server-based OSD Compute Node (osd-compute-1).

**./ultram_ucs_utils.py --cfg "osd_compute_0.cfg" --login** *<cimc_username> <cimc_user_password>* **--upgrade bios --server** *<ospd_server_cimc_ip_address>* **--timeout 30 --file /firmwares/bios.cap**
Example output:

```
2017-09-29 09:15:48,753 - Updating BIOS firmware on all the servers
2017-09-29 09:15:48,753 - Logging on UCS Server: 192.100.0.17
2017-09-29 09:15:48,758 - No session found, creating one on server: 192.100.0.17
2017-09-29 09:15:50,194 - Login successful to server: 192.100.0.17
2017-09-29 09:16:13,269 - 192.100.0.17 => updating | Image Download (5 %), OK
2017-09-29 09:17:26,669 - 192.100.0.17 => updating | Write Host Flash (75 %), OK
2017-09-29 09:18:34,524 - 192.100.0.17 => updating | Write Host Flash (75 %), OK
2017-09-29 09:19:40,892 - 192.100.0.17 => Activating BIOS
2017-09-29 09:19:55,011 -
---------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
---------------------------------------------------------------------
192.100.0.17     | SUCCESS | NA                  | Status: success, Progress: Done, OK
```

**Note** The Compute Nodes are automatically powered down after this process leaving only the CIMC interface available.

5 Upgrade the UCS server using the Host Upgrade Utility (HUU).

**./ultram_ucs_utils.py --cfg "osd_compute.cfg" --login** *<cimc_username> <cimc_user_password>* **--upgrade huu --server** *<ospd_server_cimc_ip_address>* **--file /firmwares/***<ucs_huu_iso_filename>*
If the HUU script times out before completing the upgrade, the process might still be running on the remote hosts. You can periodically check the upgrade process by entering:

**./ultram_ucs_utils.py --cfg "osd_compute.cfg" --login** *<cimc_username> <cimc_user_password>* **--status huu-upgrade**
Example output:

```
---------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
---------------------------------------------------------------------
192.100.0.17    | SUCCESS | 2017-10-20 07:10:11 | Update Complete  CIMC Completed, SasExpDN
 Completed, I350 Completed, X520 Completed, X520 Completed, 3108AB-8i Completed, UCS VIC
 1227 Completed, BIOS Completed,
---------------------------------------------------------------------
```

6 Verify that the BIOS firmware and HUU upgrade was successful by checking the post-upgrade versions.

**./ultram_ucs_utils.py --cfg "osd_compute_0.cfg" --login** *<cimc_username> <cimc_user_password>* **--status firmwares**

7 Set the package-c-state-limit CIMC setting.

**./ultram_ucs_utils.py --mgmt set-bios --bios-param biosVfPackageCStateLimit --bios-values vpPackageC-StateLimit=C0/C1 --cfg osd_compute_0.cfg --login** *<cimc_username> <cimc_user_password>*

8 Verify that the package-c-state-limit CIMC setting has been made.

**./ultram_ucs_utils.py --status bios-settings --cfg osd_compute_0.cfg --login** *<cimc_username> <cimc_user_password>*
Look for **PackageCStateLimit** to be set to *C0/C1*.

9 Modify the Grub configuration on the primary OSD Compute Node.

a Log on to the OSD Compute (osd-compute-0) and update the grub setting with "processor.max_cstate=0 intel_idle.max_cstate=0".

**sudo grubby --info=/boot/vmlinuz-`uname -r`**
**sudo grubby --update-kernel=/boot/vmlinuz-`uname -r` --args="processor.max_cstate=0 intel_idle.max_cstate=0"**

b Verify that the update was successful.

**sudo grubby --info=/boot/vmlinuz-`uname -r`**
Look for the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments in the output.

c Reboot the OSD Compute Nodes.

**sudo reboot**

10 Recheck all CIMC and kernel settings.

a Verify the processor c-state.

**cat /sys/module/intel_idle/parameters/max_cstate**
*cpupower idle-info*

b Login to Ultra M Manager Node.

c Verify CIMC settings.

**./ultram_ucs_utils.py --status bios-settings --cfg osd_compute_0.cfg --login** *<cimc_username> <cimc_user_password>*

11 Repeat steps 4, on page 123 through 10, on page 124 on the second OSD Compute Node (osd-compute-1).

**Note** Be sure to use the *osd_compute_1.cfg* file where needed.

12 Repeat steps 4, on page 123 through 10, on page 124 on the third OSD Compute Node (osd-compute-2).

**Note** Be sure to use the *osd_compute_2.cfg* file where needed.

13 Check the ironic node-list and restore any hosts that went into maintenance mode true state.

a Login to OSP-D and make sure to "su - stack" and "source stackrc".

b Perform the check and any required restorations.

**ironic node-list**
**ironic node-set-maintenance $NODE_***<node_uuid>* **off**

14 Move the Ceph storage out of maintenance mode.

a Log on to the lead Controller Node (controller-0).

b Move the Ceph storage to maintenance mode.

**sudo ceph status**
**sudo ceph osd unset noout**
**sudo ceph osd unset norebalance**
**sudo ceph status**
**sudo pcs status**

15 Proceed to Restart the UAS and ESC (VNFM) VMs, on page 125.

# Restart the UAS and ESC (VNFM) VMs

Upon performing the UCS server software upgrades, VMs that were previously shutdown must be restarted.

To restart the VMs:

1 Login to OSP-D and make sure to "su - stack" and "source stackrc".

2 Run Nova list to get the UUIDs of the ESC VMs.

3 Start the AutoIT VM.

**nova start** <*autoit_vm_uuid*>

4 Start the AutoDeploy VM.

**nova start** <*autodeploy_vm_uuid*>

5 Start the standby ESC VM.

**nova start** <*standby_vm_uuid*>

6 Start the active ESC VM.

**nova start** <*active_vm_uuid*>

7 Verify that the VMs have been restarted and are ACTIVE.

**nova list --fields name,host,status | grep** <*vnf_deployment_name*>
Once ESC is up and running, it triggers the recovery of rest of the VMs (AutoVNF, UEMs, CFs and SFs).

8 Login to each of the VMs and verify that they are operational.

# Upgrade the Controller Node Server Software

**NOTES:**

- This procedure requires the *controller_0.cfg*, *controller_1.cfg*, and *controller_2.cfg* files created as part of the procedure detailed in Perform Pre-Upgrade Preparation, on page 116.
- It is highly recommended that all Controller Nodes be upgraded using this process during a single maintenance window.

To upgrade the UCS server software on the Controller Nodes:

1 Check the Controller Node status and move the Pacemaker Cluster Stack (PCS) to maintenance mode.

a Login to the primary Controller Node (controller-0) from the OSP-D Server.

b. Check the state of the Controller Node Pacemaker Cluster Stack (PCS).

   **sudo pcs status**

> **Note** Resolve any issues prior to proceeding to the next step.

c. Place the PCS cluster on the Controller Node into standby mode.

   **sudo pcs cluster standby** *<controller_name>*

d. Recheck the Controller Node status again and make sure that the Controller Node is in standby mode for the PCS cluster.

   **sudo pcs status**

2. Log on to the Ultra M Manager Node.

3. Upgrade the BIOS on the primary UCS server-based Controller Node (controller-0).

   **./ultram_ucs_utils.py --cfg "controller_0.cfg" --login** *<cimc_username> <cimc_user_password>* **--upgrade bios --server** *<ospd_server_cimc_ip_address>* **--timeout 30 --file /firmwares/bios.cap**
   Example output:

```
2017-09-29 09:15:48,753 - Updating BIOS firmware on all the servers
2017-09-29 09:15:48,753 - Logging on UCS Server: 192.100.2.7
2017-09-29 09:15:48,758 - No session found, creating one on server: 192.100.2.7
2017-09-29 09:15:50,194 - Login successful to server: 192.100.2.7
2017-09-29 09:16:13,269 - 192.100.2.7 => updating | Image Download (5 %), OK
2017-09-29 09:17:26,669 - 192.100.2.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:18:34,524 - 192.100.2.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:19:40,892 - 192.100.2.7 => Activating BIOS
2017-09-29 09:19:55,011 -
---------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
---------------------------------------------------------------------
192.100.2.7     | SUCCESS | NA                  | Status: success, Progress: Done, OK
```

> **Note** The Compute Nodes are automatically powered down after this process leaving only the CIMC interface available.

4. Upgrade the UCS server using the Host Upgrade Utility (HUU).

   **./ultram_ucs_utils.py --cfg "controller_0.cfg" --login** *<cimc_username> <cimc_user_password>* **--upgrade huu --server** *<ospd_server_cimc_ip_address>* **--file /firmwares/***<ucs_huu_iso_filename>*
   If the HUU script times out before completing the upgrade, the process might still be running on the remote hosts. You can periodically check the upgrade process by entering:

   **./ultram_ucs_utils.py --cfg "controller_0.cfg" --login** *<cimc_username> <cimc_user_password>* **--status huu-upgrade**
   Example output:

```
---------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
---------------------------------------------------------------------
192.100.2.7    | SUCCESS | 2017-10-20 07:10:11 | Update Complete  CIMC Completed, SasExpDN
 Completed, I350 Completed, X520 Completed, X520 Completed, 3108AB-8i Completed, UCS VIC
 1227 Completed, BIOS Completed,
---------------------------------------------------------------------
```

5. Verify that the BIOS firmware and HUU upgrade was successful by checking the post-upgrade versions.

   **./ultram_ucs_utils.py --cfg "controller_0.cfg" --login** *<cimc_username> <cimc_user_password>* **--status firmwares**

6 Set the package-c-state-limit CIMC setting.

**./ultram_ucs_utils.py --mgmt set-bios --bios-param biosVfPackageCStateLimit --bios-values vpPackageC-StateLimit=C0/C1 --cfg controller_0.cfg --login** *<cimc_username>* *<cimc_user_password>*

7 Verify that the package-c-state-limit CIMC setting has been made.

**./ultram_ucs_utils.py --status bios-settings --cfg controller_0.cfg --login** *<cimc_username>* *<cimc_user_password>*
Look for **PackageCStateLimit** to be set to *C0/C1*.

8 Modify the Grub configuration on the primary OSD Compute Node.

a Log on to the OSD Compute (osd-compute-0) and update the grub setting with "processor.max_cstate=0 intel_idle.max_cstate=0".

**sudo grubby --info=/boot/vmlinuz-`uname -r`**
**sudo grubby --update-kernel=/boot/vmlinuz-`uname -r` --args="processor.max_cstate=0 intel_idle.max_cstate=0"**

b Verify that the update was successful.

**sudo grubby --info=/boot/vmlinuz-`uname -r`**
Look for the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments in the output.

c Reboot the OSD Compute Nodes.

**sudo reboot**

9 Recheck all CIMC and kernel settings.

a Verify the processor c-state.

**cat /sys/module/intel_idle/parameters/max_cstate**
*cpupower idle-info*

b Login to Ultra M Manager Node.

c Verify CIMC settings.

**./ultram_ucs_utils.py --status bios-settings --cfg controller_0.cfg --login** *<cimc_username>* *<cimc_user_password>*

10 Check the ironic node-list and restore the Controller Node if it went into maintenance mode true state.

a Login to OSP-D and make sure to "su - stack" and "source stackrc".

b Perform the check and any required restorations.

**ironic node-list**
**ironic node-set-maintenance $NODE_***<node_uuid>* **off**

11 Take the Controller Node out of the PCS standby state.

**sudo pcs cluster unstandby** *<controller-0-id>*

12 Wait 5 to 10 minutes and check the state of the PCS cluster to verify that the Controller Node is ONLINE and all services are in good state.

**sudo pcs status**

13 Repeat steps 3, on page 126 through 11, on page 127 on the second Controller Node (controller-1).

**Note** Be sure to use the *controller_1.cfg* file where needed.

14 Repeat steps 3, on page 126 through 11, on page 127 on the third Controller Node (controller-2).

**Note** Be sure to use the *controller_2.cfg* file where needed.

15 Proceed to Upgrade Firmware on the OSP-D Server/Ultra M Manager Node, on page 130.

# Upgrade Firmware on UCS Bare Metal

**NOTES:**

- This procedure assumes that the UCS servers receiving the software (firmware) upgrade have not previously been deployed as part of an Ultra M solution stack.
- The instructions in this section pertain to all servers to be used as part of an Ultra M solution stack except the OSP-D Server/Ultra M Manager Node.
- This procedure requires the *hosts.cfg* file created as part of the procedure detailed in Perform Pre-Upgrade Preparation, on page 116.

To upgrade the software on the UCS servers:

1 Log on to the Ultra M Manager Node.

2 Upgrade the BIOS on the UCS servers.

**./ultram_ucs_utils.py --cfg "hosts.cfg" --login** <*cimc_username*> <*cimc_user_password*> **--upgrade bios --server** <*ospd_server_cimc_ip_address*> **--timeout 30 --file /firmwares/bios.cap**
Example output:

```
2017-09-29 09:15:48,753 - Updating BIOS firmware on all the servers
2017-09-29 09:15:48,753 - Logging on UCS Server: 192.100.0.7
2017-09-29 09:15:48,758 - No session found, creating one on server: 192.100.0.7
2017-09-29 09:15:50,194 - Login successful to server: 192.100.0.7
2017-09-29 09:16:13,269 - 192.100.0.7 => updating | Image Download (5 %), OK
2017-09-29 09:17:26,669 - 192.100.0.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:18:34,524 - 192.100.0.7 => updating | Write Host Flash (75 %), OK
2017-09-29 09:19:40,892 - 192.100.0.7 => Activating BIOS
2017-09-29 09:19:55,011 -
---------------------------------------------------------------------
Server IP       | Overall | Updated-on           | Status
---------------------------------------------------------------------
192.100.0.7     | SUCCESS | NA                   | Status: success, Progress: Done, OK
```

**Note** The Compute Nodes are automatically powered down after this process leaving only the CIMC interface available.

3 Upgrade the UCS server using the Host Upgrade Utility (HUU).

**./ultram_ucs_utils.py --cfg "hosts.cfg" --login** <*cimc_username*> <*cimc_user_password*> **--upgrade huu --server** <*ospd_server_cimc_ip_address*> **--file /firmwares/**<*ucs_huu_iso_filename*>
If the HUU script times out before completing the upgrade, the process might still be running on the remote hosts. You can periodically check the upgrade process by entering:

**./ultram_ucs_utils.py --cfg "hosts.cfg" --login** <*cimc_username*> <*cimc_user_password*> **--status huu-upgrade**

Example output:

```
----------------------------------------------------------------------
Server IP       | Overall | Updated-on          | Status
----------------------------------------------------------------------
192.100.0.7    | SUCCESS | 2017-10-20 07:10:11 | Update Complete  CIMC Completed, SasExpDN
 Completed, I350 Completed, X520 Completed, X520 Completed, 3108AB-8i Completed, UCS VIC
 1227 Completed, BIOS Completed,
----------------------------------------------------------------------
```

4 Verify that the BIOS firmware and HUU upgrade was successful by checking the post-upgrade versions.

**./ultram_ucs_utils.py --cfg "hosts.cfg" --login** *<cimc_username> <cimc_user_password>* **--status firmwares**

5 Set the package-c-state-limit CIMC setting.

**./ultram_ucs_utils.py --mgmt set-bios --bios-param biosVfPackageCStateLimit --bios-values vpPackageC-StateLimit=C0/C1 --cfg hosts.cfg --login** *<cimc_username> <cimc_user_password>*

6 Verify that the package-c-state-limit CIMC setting has been made.

**./ultram_ucs_utils.py --status bios-settings --cfg hosts.cfg --login** *<cimc_username> <cimc_user_password>*
Look for **PackageCStateLimit** to be set to *C0/C1*.

7 Recheck all CIMC and BIOS settings.

a Log in to the Ultra M Manager Node.

b Verify CIMC settings.

**./ultram_ucs_utils.py --status bios-settings --cfg hosts.cfg --login** *<cimc_username> <cimc_user_password>*

8 Modify the "ComputeKernelArgs" statement in the *network.yaml* with the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments.

**vi network.yaml**

```
<---SNIP--->
ComputeKernelArgs: "intel_iommu=on default_hugepagesz=1GB hugepagesz=1G hugepages=12
processor.max_cstate=0 intel_idle.max_cstate=0"
```

9 Modify the Grub configuration on all Controller Nodes after the VIM (Overcloud) has been deployed.

a Log into your first Controller Node (controller-0).

**ssh heat-admin@***<controller_address>*

b Check the grubby settings.

**sudo grubby --info=/boot/vmlinuz-\`uname -r\`**
Example output:

```
index=0
kernel=/boot/vmlinuz-3.10.0-514.21.1.el7.x86_64
args="ro console=tty0 console=ttyS0,115200n8 crashkernel=auto rhgb quiet "
root=UUID=fa9e939e-9e3c-4f1c-a07c-3f506756ad7b
initrd=/boot/initramfs-3.10.0-514.21.1.el7.x86_64.img
title=Red Hat Enterprise Linux Server (3.10.0-514.21.1.el7.x86_64) 7.3 (Maipo)
```

c Update the grub setting with the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments.

**sudo grubby --update-kernel=/boot/vmlinuz-\`uname -r\` --args="processor.max_cstate=0 intel_idle.max_cstate=0"**

d Verify that the update was successful.

**sudo grubby --info=/boot/vmlinuz-\`uname -r\`**
Look for the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments in the output.

Example output:

```
index=0
kernel=/boot/vmlinuz-3.10.0-514.21.1.el7.x86_64
args="ro console=tty0 console=ttyS0,115200n8 crashkernel=auto rhgb quiet
processor.max_cstate=0 intel_idle.max_cstate=0"
root=UUID=fa9e939e-9e3c-4f1c-a07c-3f506756ad7b
initrd=/boot/initramfs-3.10.0-514.21.1.el7.x86_64.img
title=Red Hat Enterprise Linux Server (3.10.0-514.21.1.el7.x86_64) 7.3 (Maipo)
```

e Reboot the Controller Node.

**sudo reboot**

**Important** Do not proceed with the next step until the Controller Node is up and rejoins the cluster.

f Repeat steps 9.a, on page 129 through 9.e, on page 130 for all other Controller Nodes.

10 Proceed to Upgrade Firmware on the OSP-D Server/Ultra M Manager Node, on page 130.

# Upgrade Firmware on the OSP-D Server/Ultra M Manager Node

1 Open your web browser.

2 Enter the CIMC address of the OSP-D Server/Ultra M Manager Node in the URL field.

3 Login to the CIMC using the configured user credentials.

4 Click **Launch KVM Console**.

5 Click **Virtual Media**.

6 Click **Add Image** and select the HUU ISO file pertaining to the version you wish to upgrade to.

7 Select the ISO that you have added in the **Mapped** column of the **Client View**. Wait for the selected ISO to appear as a mapped device.

8 Boot the server and press F6 when prompted to open the **Boot Menu**.

9 Select the desired ISO.

10 Select **Cisco vKVM-Mapped vDVD1.22**, and press **Enter**. The server boots from the selected device.

11 Follow the onscreen instructions to update the desired software and reboot the server. Proceed to the next step once the server has rebooted.

12 Log on to the Ultra M Manager Node.

13 Set the package-c-state-limit CIMC setting.

**./ultram_ucs_utils.py --mgmt set-bios --bios-param biosVfPackageCStateLimit --bios-values vpPackageC-StateLimit=C0/C1 --cfg ospd.cfg --login** *<cimc_username>* *<cimc_user_password>*

14 Verify that the package-c-state-limit CIMC setting has been made.

**./ultram_ucs_utils.py --status bios-settings --cfg controller.cfg --login** *<cimc_username>* *<cimc_user_password>*
Look for **PackageCStateLimit** to be set to C0/C1.

15 Update the grub setting with "processor.max_cstate=0 intel_idle.max_cstate=0".

**sudo grubby --info=/boot/vmlinuz-`uname -r`**
**sudo grubby --update-kernel=/boot/vmlinuz-`uname -r` --args="processor.max_cstate=0 intel_idle.max_cstate=0"**

16 Verify that the update was successful.

**sudo grubby --info=/boot/vmlinuz-`uname -r`**
Look for the "processor.max_cstate=0 intel_idle.max_cstate=0" arguments in the output.

17 Reboot the server.

**sudo reboot**

18 Recheck all CIMC and kernel settings upon reboot.

a Verify CIMC settings

**./ultram_ucs_utils.py --status bios-settings --cfg ospd.cfg --login** <*cimc_username*> <*cimc_user_password*>

b Verify the processor c-state.

**cat /sys/module/intel_idle/parameters/max_cstate**
*cpupower idle-info*

# Controlling UCS BIOS Parameters Using *ultram_ucs_utils.py* Script

The ultram_ucs_utils.py script can be used to modify and verify parameters within the UCS server BIOS. This script is in the */opt/cisco/usp/ultram-manager* directory.

**Important** Refer to the UCS server documentation BIOS documentation for information on parameters and their respective values.

To configure UCS server BIOS parameters:

1 Log on to the Ultra M Manager Node.

2 Modify the desired BIOS parameters.

**./ultram_ucs_utils.py --cfg** "*config_file_name*" **--login** *cimc_username cimc_user_password* **--mgmt 'set-bios' –-bios-param** *bios_paramname* **–-bios-values** *bios_value1 bios_value2*
Example:

```
./ultram_ucs_utils.py --cfg cmp_17 --login admin abcabc --mgmt 'set-bios --bios-param
biosVfUSBPortsConfig --bios-values vpAllUsbDevices=Disabled vpUsbPortRear=Disabled
```

Example output:

```
2017-10-06 19:48:39,241 - Set BIOS Parameters
2017-10-06 19:48:39,241 - Logging on UCS Server: 192.100.0.25
2017-10-06 19:48:39,243 - No session found, creating one on server: 192.100.0.25
2017-10-06 19:48:40,711 - Login successful to server: 192.100.0.25
2017-10-06 19:48:52,709 - Logging out from the server: 192.100.0.25
2017-10-06 19:48:53,893 - Successfully logged out from the server: 192.100.0.25
```

3 Verify that your settings have been incorporated.

**./ultram_ucs_utils.py --cfg** "*config_file_name*" **--login** *cimc_username cimc_user_password* **-- status bios-settings**

Example output:

**./ultram_ucs_utils.py --cfg cmp_17 --login admin abcabc --status bios-settings**

```
2017-10-06 19:49:12,366 - Getting status information from all the servers
2017-10-06 19:49:12,366 - Logging on UCS Server: 192.100.0.25
2017-10-06 19:49:12,370 - No session found, creating one on server: 192.100.0.25
2017-10-06 19:49:13,752 - Login successful to server: 192.100.0.25
2017-10-06 19:49:19,739 - Logging out from the server: 192.100.0.25
2017-10-06 19:49:20,922 - Successfully logged out from the server: 192.100.0.25
2017-10-06 19:49:20,922 -
-------------------------------------------------------------------------
Server IP      | BIOS Settings
------------------------------------------------------------------------
192.100.0.25   | biosVfHWPMEnable
               |  vpHWPMEnable: Disabled
               | biosVfLegacyUSBSupport
               |  vpLegacyUSBSupport: enabled
               | biosVfPciRomClp
               |  vpPciRomClp: Disabled
               | biosVfSelectMemoryRASConfiguration
               |  vpSelectMemoryRASConfiguration: maximum-performance
               | biosVfExtendedAPIC
               |  vpExtendedAPIC: XAPIC
               | biosVfOSBootWatchdogTimerPolicy
               |  vpOSBootWatchdogTimerPolicy: power-off
               | biosVfCoreMultiProcessing
               |  vpCoreMultiProcessing: all
               | biosVfQPIConfig
               |  vpQPILinkFrequency: auto
               | biosVfOutOfBandMgmtPort
               |  vpOutOfBandMgmtPort: Disabled
               | biosVfVgaPriority
               |  vpVgaPriority: Onboard
               | biosVfMemoryMappedIOAbove4GB
               |  vpMemoryMappedIOAbove4GB: enabled
               | biosVfEnhancedIntelSpeedStepTech
               |  vpEnhancedIntelSpeedStepTech: enabled
               | biosVfCmciEnable
               |  vpCmciEnable: Enabled
               | biosVfAutonumousCstateEnable
               |  vpAutonumousCstateEnable: Disabled
               | biosVfOSBootWatchdogTimer
               |  vpOSBootWatchdogTimer: disabled
               | biosVfAdjacentCacheLinePrefetch
               |  vpAdjacentCacheLinePrefetch: enabled
               | biosVfPCISlotOptionROMEnable
               |  vpSlot1State: Disabled
               |  vpSlot2State: Disabled
               |  vpSlot3State: Disabled
               |  vpSlot4State: Disabled
               |  vpSlot5State: Disabled
               |  vpSlot6State: Disabled
               |  vpSlotMLOMState: Enabled
               |  vpSlotHBAState: Enabled
               |  vpSlotHBALinkSpeed: GEN3
               |  vpSlotN1State: Disabled
               |  vpSlotN2State: Disabled
               |  vpSlotFLOMLinkSpeed: GEN3
               |  vpSlotRiser1Slot1LinkSpeed: GEN3
               |  vpSlotRiser1Slot2LinkSpeed: GEN3
               |  vpSlotRiser1Slot3LinkSpeed: GEN3
               |  vpSlotSSDSlot1LinkSpeed: GEN3
               |  vpSlotSSDSlot2LinkSpeed: GEN3
               |  vpSlotRiser2Slot4LinkSpeed: GEN3
               |  vpSlotRiser2Slot5LinkSpeed: GEN3
               |  vpSlotRiser2Slot6LinkSpeed: GEN3
               | biosVfProcessorC3Report
               |  vpProcessorC3Report: disabled
               | biosVfPCIeSSDHotPlugSupport
               |  vpPCIeSSDHotPlugSupport: Disabled
               | biosVfExecuteDisableBit
               |  vpExecuteDisableBit: enabled
               | biosVfCPUEnergyPerformance
```

```
|  vpCPUEnergyPerformance: balanced-performance
| biosVfAltitude
|  vpAltitude: 300-m
| biosVfSrIov
|  vpSrIov: enabled
| biosVfIntelVTForDirectedIO
|  vpIntelVTDATSSupport: enabled
|  vpIntelVTDCoherencySupport: disabled
|  vpIntelVTDInterruptRemapping: enabled
|  vpIntelVTDPassThroughDMASupport: disabled
|  vpIntelVTForDirectedIO: enabled
| biosVfCPUPerformance
|  vpCPUPerformance: enterprise
| biosVfPchUsb30Mode
|  vpPchUsb30Mode: Disabled
| biosVfTPMSupport
|  vpTPMSupport: enabled
| biosVfIntelHyperThreadingTech
|  vpIntelHyperThreadingTech: disabled
| biosVfIntelTurboBoostTech
|  vpIntelTurboBoostTech: enabled
| biosVfUSBEmulation
|  vpUSBEmul6064: enabled
| biosVfMemoryInterleave
|  vpChannelInterLeave: auto
|  vpRankInterLeave: auto
| biosVfConsoleRedirection
|  vpBaudRate: 115200
|  vpConsoleRedirection: disabled
|  vpFlowControl: none
|  vpTerminalType: vt100
|  vpPuttyKeyPad: ESCN
|  vpRedirectionAfterPOST: Always Enable
| biosVfQpiSnoopMode
|  vpQpiSnoopMode: auto
| biosVfPStateCoordType
|  vpPStateCoordType: HW ALL
| biosVfProcessorC6Report
|  vpProcessorC6Report: enabled
| biosVfPCIOptionROMs
|  vpPCIOptionROMs: Enabled
| biosVfDCUPrefetch
|  vpStreamerPrefetch: enabled
|  vpIPPrefetch: enabled
| biosVfFRB2Enable
|  vpFRB2Enable: enabled
| biosVfLOMPortOptionROM
|  vpLOMPortsAllState: Enabled
|  vpLOMPort0State: Enabled
|  vpLOMPort1State: Enabled
| biosVfPatrolScrub
|  vpPatrolScrub: enabled
| biosVfNUMAOptimized
|  vpNUMAOptimized: enabled
| biosVfCPUPowerManagement
|  vpCPUPowerManagement: performance
| biosVfDemandScrub
|  vpDemandScrub: enabled
| biosVfDirectCacheAccess
|  vpDirectCacheAccess: auto
| biosVfPackageCStateLimit
|  vpPackageCStateLimit: C6 Retention
| biosVfProcessorC1E
|  vpProcessorC1E: enabled
| biosVfUSBPortsConfig
|  vpAllUsbDevices: disabled
|  vpUsbPortRear: disabled
|  vpUsbPortFront: enabled
|  vpUsbPortInternal: enabled
|  vpUsbPortKVM: enabled
|  vpUsbPortVMedia: enabled
| biosVfSataModeSelect
|  vpSataModeSelect: AHCI
```

```
               | biosVfOSBootWatchdogTimerTimeout
               |  vpOSBootWatchdogTimerTimeout: 10-minutes
               | biosVfWorkLoadConfig
               |  vpWorkLoadConfig: Balanced
               | biosVfCDNEnable
               |  vpCDNEnable: Disabled
               | biosVfIntelVirtualizationTechnology
               |  vpIntelVirtualizationTechnology: enabled
               | biosVfHardwarePrefetch
               |  vpHardwarePrefetch: enabled
               | biosVfPwrPerfTuning
               |  vpPwrPerfTuning: os
------------------------------------------------------------------------
```

APPENDIX **G**

# ultram_ucs_utils.py Help

Enter the following command to display help for the UCS utilities available through the Ultra M Manager:

**./ultram_ucs_utils.py h**

```
usage: ultram_ucs_utils.py [-h] --cfg CFG --login UC_LOGIN UC_LOGIN
                           (--upgrade  | --mgmt  | --status  | --undercloud UC_RC)
                           [--mode] [--serial-delay SERIAL_DELAY]
                           [--server SERVER] [--file FILE]
                           [--protocol {http,https,tftp,sftp,ftp,scp}]
                           [--access ACCESS ACCESS] [--secure-boot]
                           [--update-type {immediate,delayed}] [--reboot]
                           [--timeout TIMEOUT] [--verify] [--stop-on-error]
                           [--bios-param BIOS_PARAM]
                           [--bios-values BIOS_VALUES [BIOS_VALUES ...]]

optional arguments:
  -h, --help            show this help message and exit
  --cfg CFG             Configuration file to read servers
  --login UC_LOGIN UC_LOGIN
                        Common Login UserName / Password to authenticate UCS servers
  --upgrade             Firmware upgrade, choose one from:
                          'bios': Upgrade BIOS firmware version
                          'cimc': Upgrade CIMC firmware version
                          'huu' : Upgrade All Firmwares via HUU based on ISO
  --mgmt                Server Management Tasks, choose one from:
                          'power-up'      :  Power on the server immediately
                          'power-down'    :  Power down the server (non-graceful)
                          'soft-shut-down':  Shutdown the server gracefully
                          'power-cycle'   :  Power Cycle the server immediately
                          'hard-reset'    :  Hard Reset the server
                          'cimc-reset'    :  Reboot CIMC
                          'cmos-reset'    :  Reset CMOS
                          'set-bios'      :  Set BIOS Parameter
  --status              Firmware Update Status:
                          'bios-upgrade'  :  Last BIOS upgrade status
                          'cimc-upgrade'  :  Last CIMC upgrade status
                          'huu-upgrade'   :  Last ISO upgrade via Host Upgrade Utilties
                         'firmwares'     :  List Current set of running firmware versions

                          'server'        :  List Server status
                          'bios-settings' :  List BIOS Settings
  --undercloud UC_RC    Get the list of servers from undercloud
  --mode                Execute action in serial/parallel
  --serial-delay SERIAL_DELAY
                        Delay (seconds) in executing firmware upgrades on node in case of
 serial mode

Firmware Upgrade Options::
  --server SERVER       Server IP hosting the file via selected protocol
  --file FILE           Firmware file path for UCS server to access from file server
  --protocol {http,https,tftp,sftp,ftp,scp}
```

```
                        Protocol to get the firmware file on UCS server
  --access ACCESS ACCESS
                        UserName / Password to access the file from remote server using
https,sftp,ftp,scp
  --secure-boot         Use CIMC Secure-Boot.
  --update-type {immediate,delayed}
                        Update type whether to send delayed update to server or immediate

  --reboot              Reboot CIMC before performing update
  --timeout TIMEOUT     Update timeout in mins should be more than 30 min and less than
200 min
  --verify              Use this option to verify update after reboot
  --stop-on-error       Stop the firmware update once an error is encountered

BIOS Parameters configuratioon:
  --bios-param BIOS_PARAM
                        BIOS Paramater Name to be set
  --bios-values BIOS_VALUES [BIOS_VALUES ...]
                      BIOS Paramater values in terms of key=value pair separated by space
```