Ultra M Solutions Guide with CVIM, Release 6.2.bx

First Published: 2018-10-16

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

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

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

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

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

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

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

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

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

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

CONTENTS

PREFACE

CHAPTER 1

CHAPTER 2

CHAPTER 3

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: `Login:`
Text represented as **commands**	This typeface represents commands that you enter, for example: **show ip access-list** 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: **show card** *slot_number* *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: 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 Virtualized Infrastructure Manager (CVIM), 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 the CVIM.

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 C2.1 micropod model is currently available. It is based on OpenStack 10 and implements a All-in-one architecture that combines the Controller, Ceph Storage and Compute nodes. The converged node is referred to as a Micropod node.

This model can have one SF per compute node and/or two SFs per compute node. It can support one or two VNFs.

Functional Components

As described in Hardware Specifications, on page 5, the Ultra M solution consists of multiple hardware components including multiple servers that function as compute and micropod nodes.

A typical Ultra-M POD comprises the following functional components:

- Compute nodes hosting Service Function (SF) VMs
- Micropod nodes hosting controller, storage, AutoVNF, ESC, UEM and CF
- CVIM Management node
- Leaf routers
- OAM routers

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, on page 2. Some of these component VMs are deployed on a single compute node as described in VM Deployment per Node Type, on page 6. The deployment model uses three OpenStack controllers to provide VIM layer redundancy and upgradability.

Table 1: Function VM Requirements

Function(s)	VMs
AutoVNF	2
ESC (VNFM)	2
UEM	3 per VNF
CF	2 per VNF
SF	4 to 16 per VNF
Important In the micropod model one or two VNFs are supported. For a 2 VNF deployment, CF and SF VMs for VNF1 are placed on NUMA0 and uses NIC1 while CF and SF VMs for VNF2 are placed on NUMA1 and uses NIC2. For a single VNF system, both NUMAs are used.	

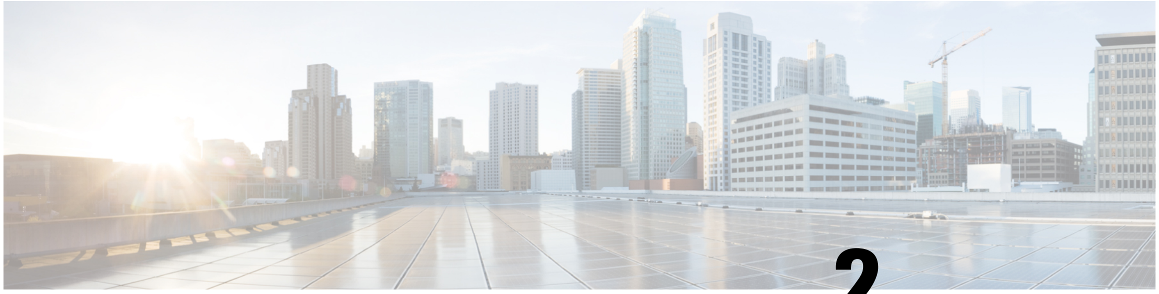
VM Resource Requirements

The CF, SF, UEM, and ESC VMs require the resource allocations identified in Table 2: VM Resource Allocation, on page 3.

Table 2: VM Resource Allocation

Virtual Machine	vCPU	RAM (GB)	Root Disk (GB)
AutoVNF	4	8	40
ESC	4	8	40
UEM	2	4	40
CF	8	16	6
SF	For 2 VNFs : 22 For single VNF: 44	For 2 VNFs: 164 For single VNF: 328	6

Note For micropod nodes, the host reservations are 2 vCPUs and 41GB RAM. For compute nodes, the host reservations are 2 vCPUs and 25GB RAM.



CHAPTER 2

Hardware Specifications

Ultra M deployment uses 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 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 93108-TC-FX

Nexus 93108 switches serve as network leafs within the Ultra M solution. Each switch has 48 1/10GBASE-T ports and 6 40/100-Gbps Quad SFP+ (QSFP+) uplink ports.

Table 3: Nexus 93108-TC-FX

Ultra M Model(s)	Quantity	Software Version	Firmware Version
Ultra M - Micropod	2	NX-OS: 7.0(3)I7(5)	BIOS: 5.28

Nexus 9364C

Nexus 9364 switches serve as network spines within the Ultra M solution. Each switch provides 64 40/100G Quad SFP+ (QSFP+) ports.

Table 4: Nexus 9364C

Ultra M Model(s)	Quantity	Software Version	Firmware Version
Ultra M - Micropod	2	NX-OS: 7.0(3)I7(5)	BIOS: 5.28

UCS C-Series Servers

Cisco UCS C220 M5SX Small Form Factor (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:

- CVIM Manager Node
- Micropod Nodes
- Compute Nodes

Table 5: Ultra M Server Quantities by Function, on page 6 provides information on server quantity requirements per function for each Ultra M model.

Table 5: Ultra M Server Quantities by Function

Server Quantity (max)	CVIM Manager Node	Micropod Nodes	Compute Nodes (max)	Additional Specifications
20	1	3	16	Based on node type as described in Table 6: Ultra M Single and Multi-VNF UCS C220 Server Specifications by Node Type, on page 9.

VM Deployment per Node Type

Figure 1: VM Distribution on Server Nodes for Ultra M Single VNF Model, on page 7 and Figure 2: VM Distribution on Server Nodes for Ultra M Two VNF Models, on page 8 depict the VM Distribution on Server Nodes for Ultra M single VNF and two VNF models.

Figure 1: VM Distribution on Server Nodes for Ultra M Single VNF Model

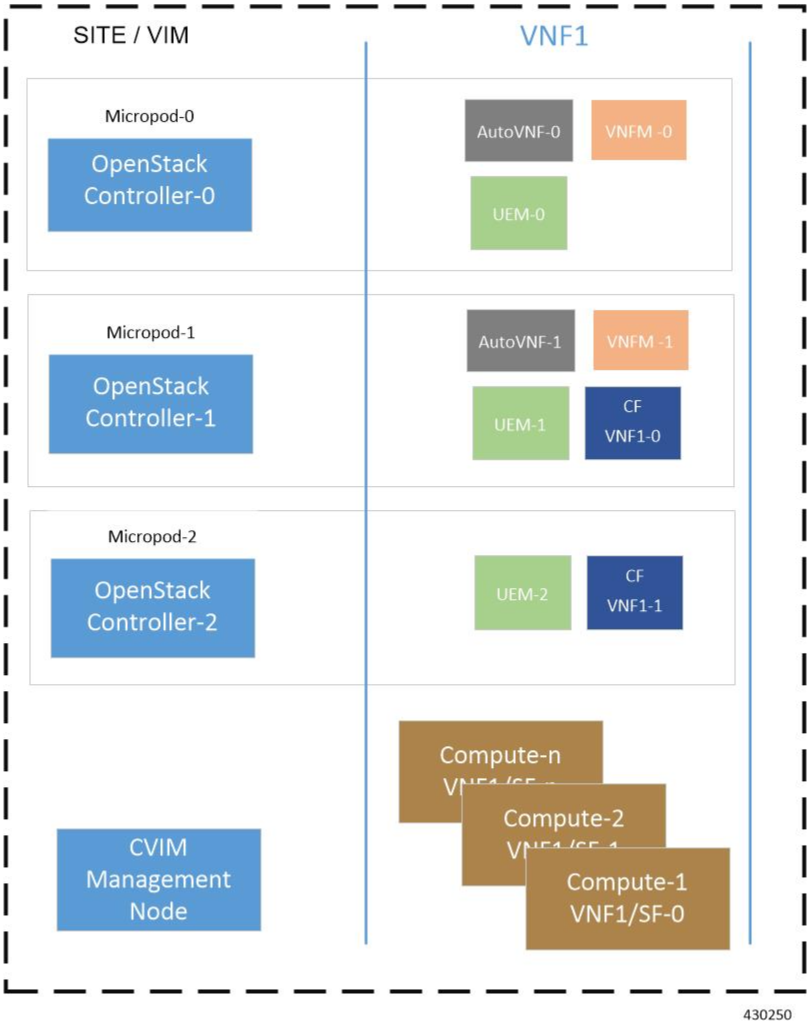
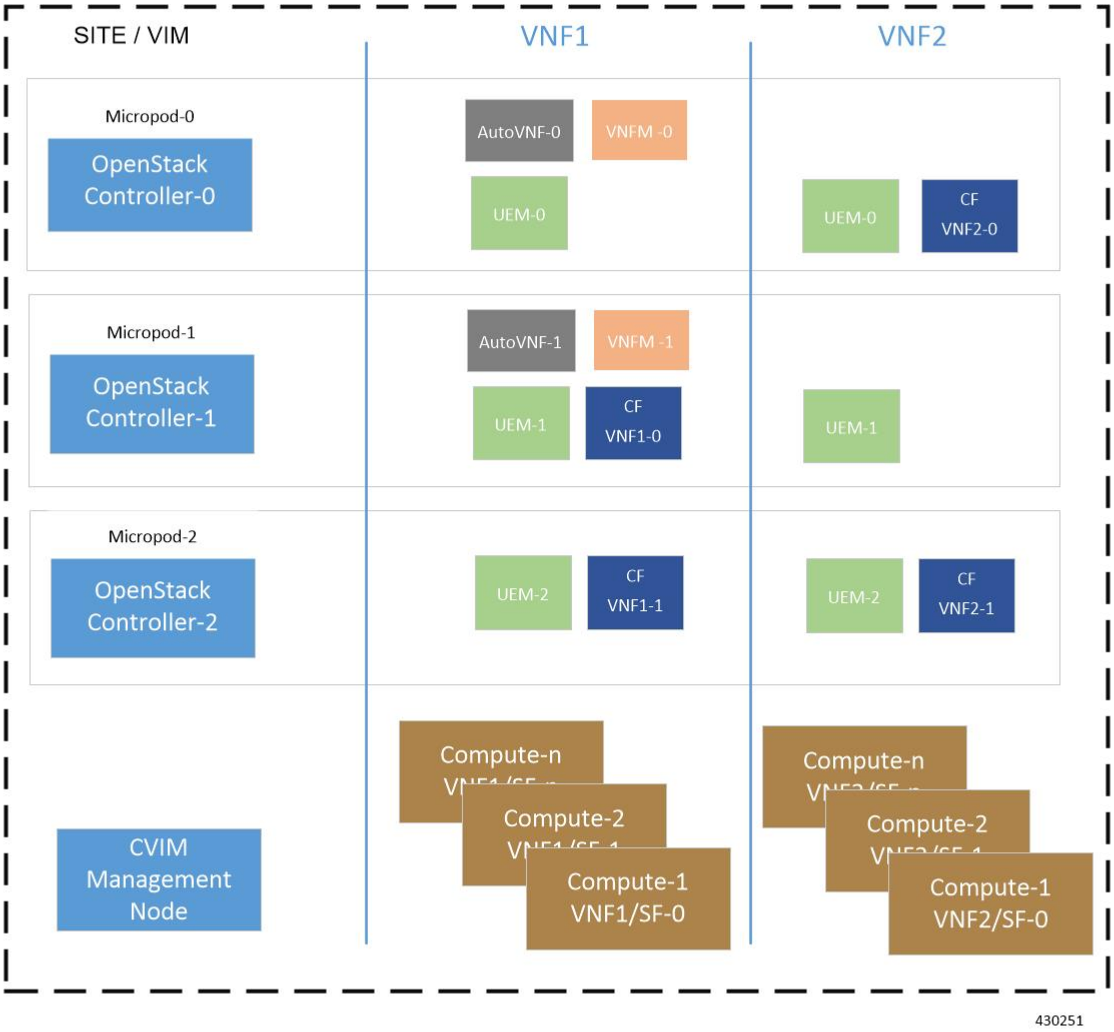


Figure 2: VM Distribution on Server Nodes for Ultra M Two VNF Models



Important In the case of 2 VNF deployments, the AutoVNF and VNFM instances are shared between the two VNFs.

Server Configurations

Table 6: Ultra M Single and Multi-VNF UCS C220 Server Specifications by Node Type

Node Type	CPU	RAM	Storage	NIC	VIC	CIMC/BIOS
CVIM Manager Node	2x 2.7 GHz 8168/205W 24C/33MB Cache/DDR4 2666MHz	12x 32GB DDR4-2666-MHz RDIMM/PC4-21300/dual	8x 1.2 TB 12G SAS 10K RPM SFF HDD	2x Intel XL710 dual-port 40G QSFP+ NIC XL710 Version: 2.4.10	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM: 4.2(3b)	CIMC: 3.1(3h) or later System BIOS: C220M5S 3.1.3d.0
Micropod Nodes	2x 2.7 GHz 8168/205W 24C/33MB Cache/DDR4 2666MHz	12x 32GB DDR4-2666-MHz RDIMM/PC4-21300/dual	2x 1.2 TB 12G SAS 10K RPM SFF HDD 4x 800GB 2.5in Enterprise Performance 12G SAS SSD(3x endurance) 1x 800GB 2.5in U.2 HGST SN200 NVMe High Perf. High Endurance	2x Intel XL710 dual-port 40G QSFP+ NIC XL710 Version: 2.4.10	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM: 4.2(3b)	CIMC: 3.1(3h) or later System BIOS: C220M5S 3.1.3d.0
Compute Node	2x 2.7 GHz 8168/205W 24C/33MB Cache/DDR4 2666MHz	12x 32GB DDR4-2666-MHz RDIMM/PC4-21300/dual	2x 1.2 TB 12G SAS 10K RPM SFF HDD	2x Intel XL710 dual-port 40G QSFP+ NIC XL710 Version: 2.4.10	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM: 4.2(3b)	CIMC: 3.1(3h) or later System BIOS: C220M5S 3.1.3d.0

CHAPTER 3

Software Specifications

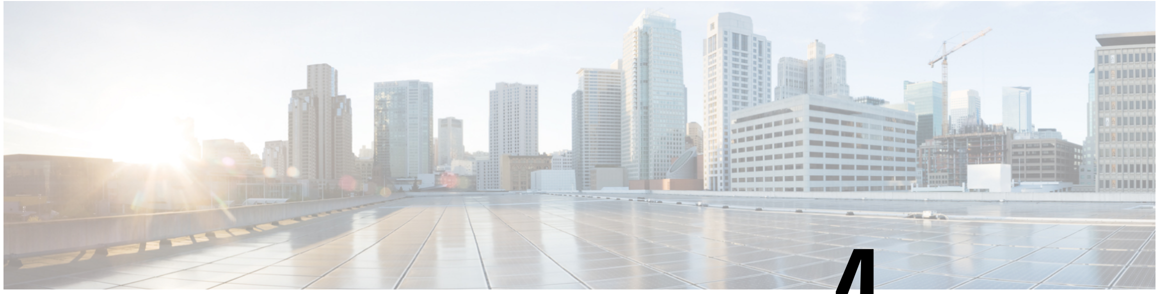
This chapter provides information on the required software for Ultra M deployments.

Required Software

Ultra M deployments use the following software:

Table 7: Required Software

Software	Value/Description
Operating System	Red Hat Enterprise Linux 7.4
Hypervisor	Qemu (KVM)
VIM	**Ultra M Micropod 2-VNF Model:** Cisco VIM 2.4.5 RedHat OpenStack Platform 10 (OSP 10 - Newton)
VNF	21.6.bx/21.6.b13
VNFM	ESC 4.0.0.104
UEM	UEM 6.2
USP	6.2.bx/6.2.b3



CHAPTER 4

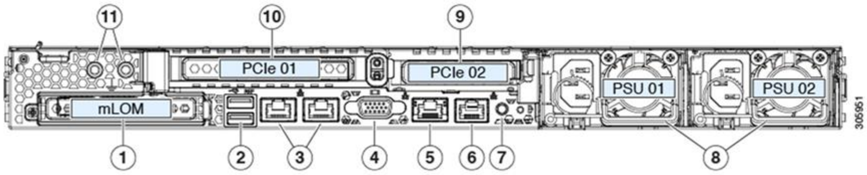
Networking Overview

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

UCS C220 M5SX Network Interfaces

Figure 3: UCS-C220 Back-Plane, on page 13 illustrates the backplane image of UCS C220 M5SX server.

Figure 3: UCS-C220 Back-Plane



Number	Designation	Description	Applicable Node Types
5	CIMC/IPMI/M	The server's *Management* network interface used for accessing the UCS Cisco Integrated Management Controller (CIMC) application, performing Intelligent Platform Management Interface (IPMI) operations. Only CVIM management node CIMC connection uses this port.	CVIM Management

Number	Designation	Description	Applicable Node Types
3	Shared-LOM	The server's Shared LOM ports are used for CIMC connectivity for Micropod nodes and Compute nodes. These Shared LOM ports are used for br_api in case of CVIM Management node.	All
		Port 2: *External* network interface for Internet access. It must also be routable to External floating IP addresses on other nodes.	Ultra M Manager Node Staging Server
1	Modular LAN on Motherboard (mLOM)	VIM networking interfaces used for:	
		• External floating IP network.	Micropod
		• Internal API network	Micropod
		• Storage network	Micropod Compute
		• Storage Management network	Micropod Compute
		• Tenant network (virtio only – VIM provisioning, VNF Management, and VNF Orchestration)	Micropod Compute
10, 9	PCIe 01, PCIe 02	Port 1 , Port 2: These ports are used for Service Net interfaces for VNF ingress and egress connections and DI-Internal network for inter-VNF component communication.	Compute

VIM Network Topology

Ultra M's VIM is based on CVIM. For information on Cisco VIM Network Topology, see the *Cisco Virtualized Infrastructure Manager Installation Guide*.

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 4: Physical NIC to Bridge Mappings, on page 15. 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 5: NIC Bonding for Single VNF, on page 15 and Figure 6: NIC Bonding for 2 VNFs, on page 16.

Figure 4: Physical NIC to Bridge Mappings

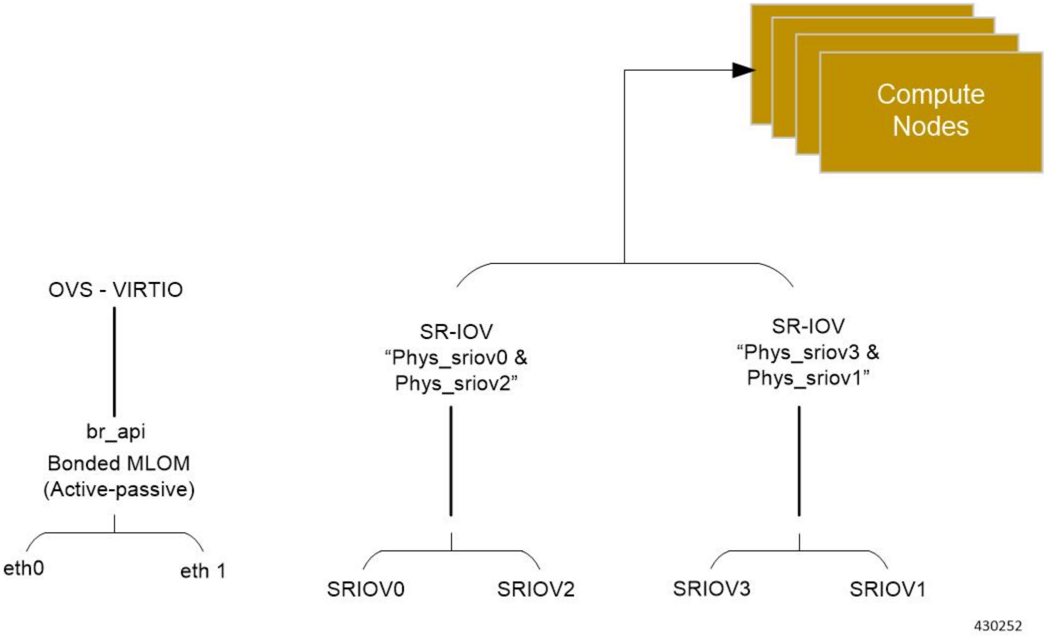


Figure 5: NIC Bonding for Single VNF

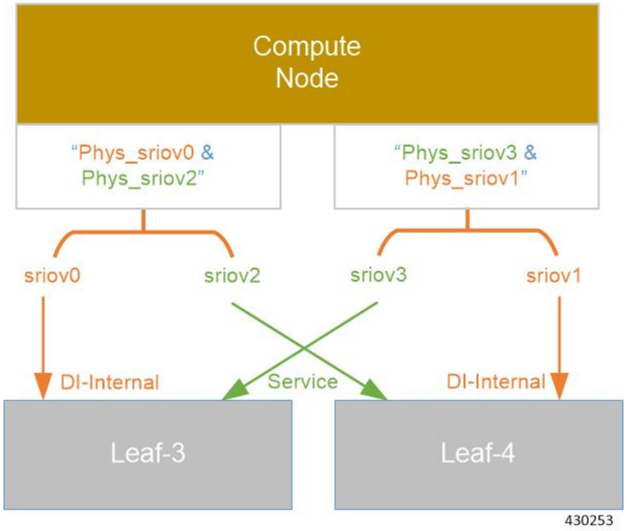
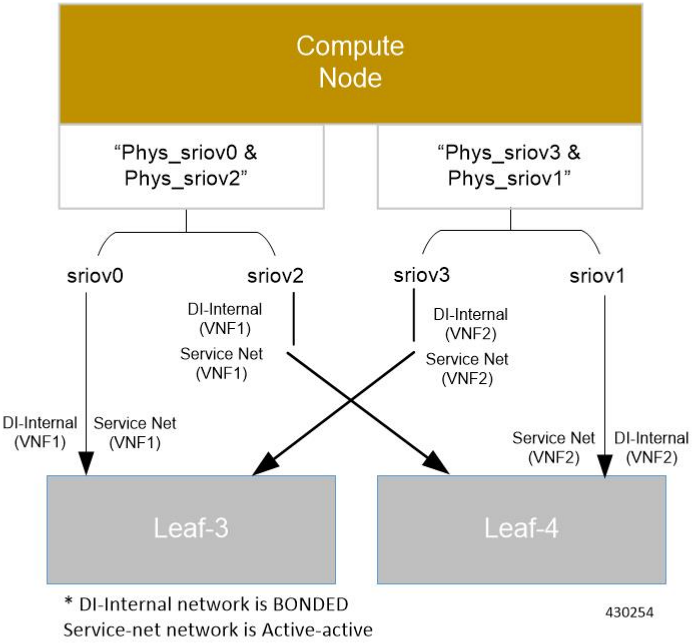


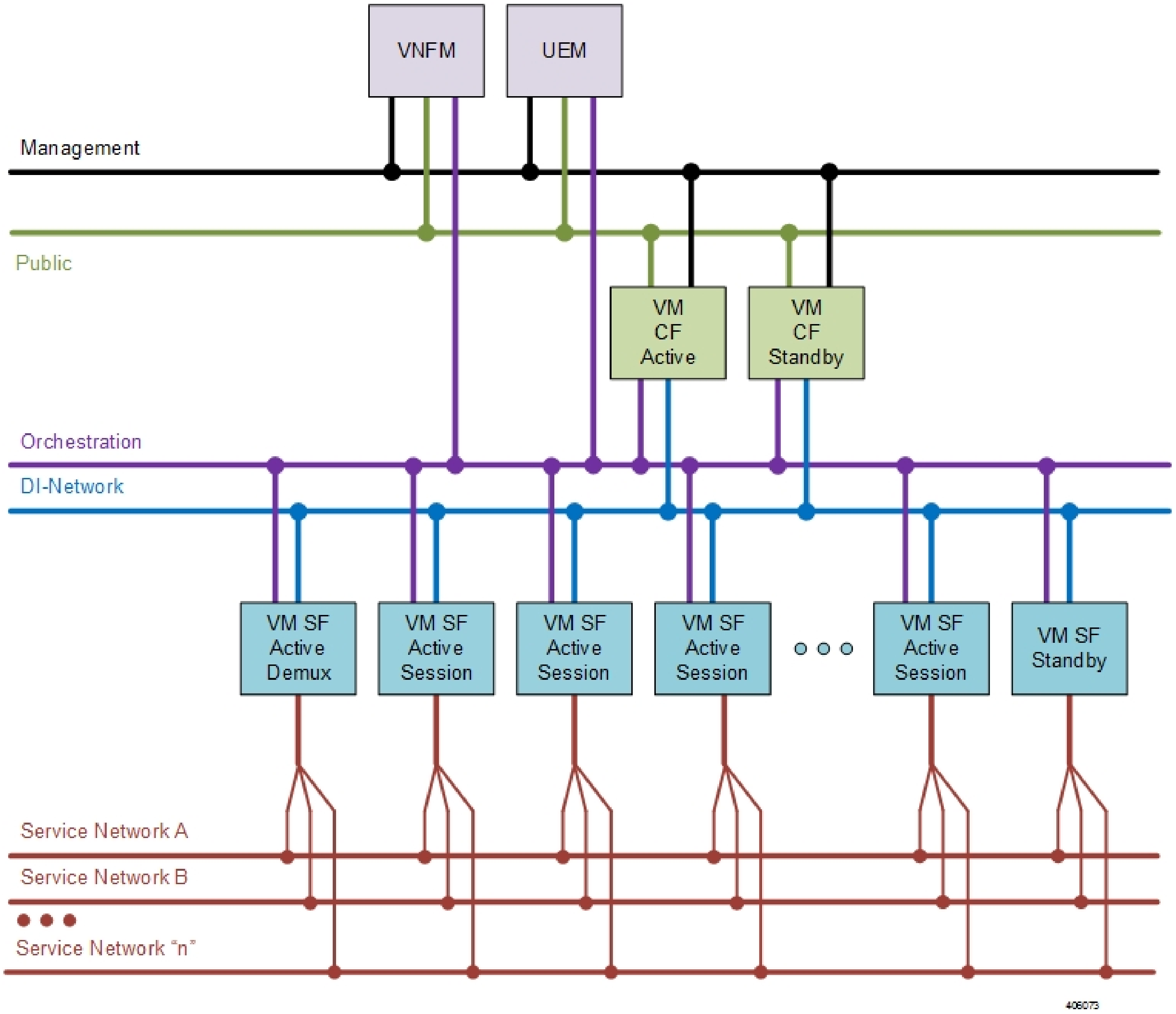
Figure 6: NIC Bonding for 2 VNFs



VNF Tenant Networks

While specific VNF network requirements are described in the documentation corresponding to the VNF, Figure 7: Typical USP-based VNF Networks, on page 17 displays the types of networks typically required by USP-based VNFs.

Figure 7: 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 18. In addition, a template is provided in Network Definitions (Layer 2 and 3), on page 41 Appendix to assist you with your Layer 2 and Layer 3 network planning.

Layer 1 Leaf and Spine Topology

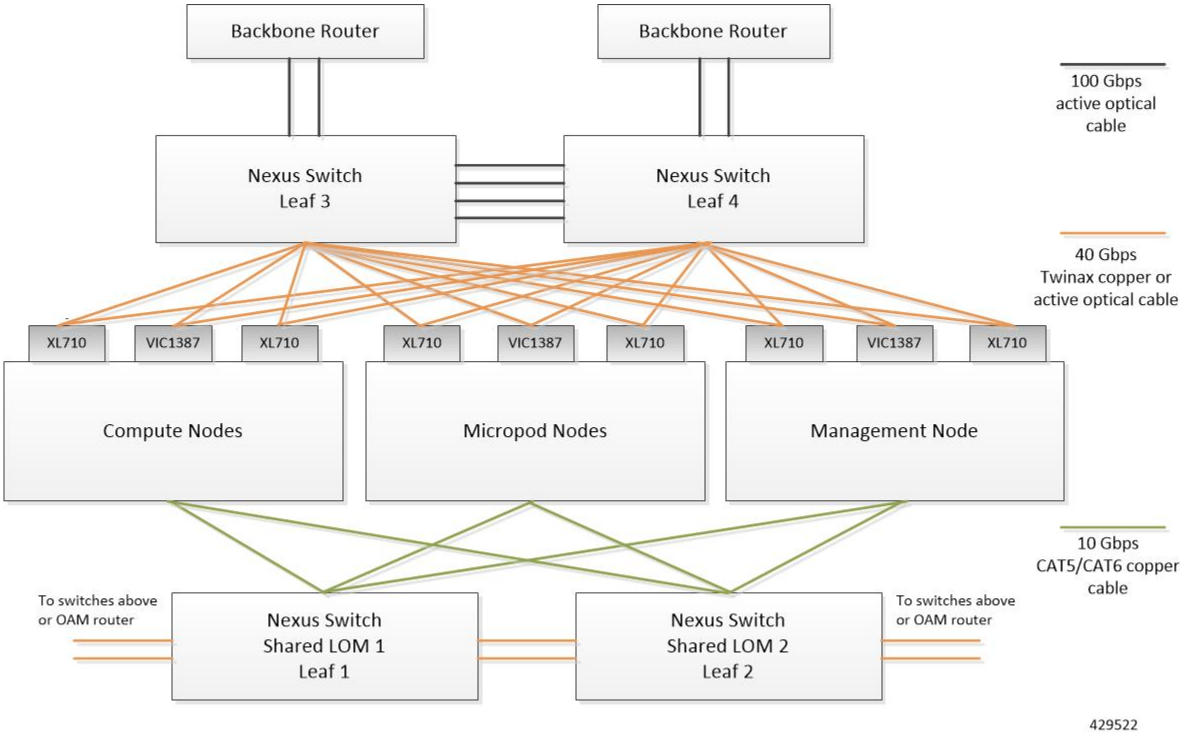
Ultra-M topology details differ between Ultra M models based on the scale and number of nodes.

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 40G port to a 40G port). Additionally, the source and destination ports must support the same physical medium (e.g. Ethernet) for interconnectivity.

Figure 8: Ultra M Single and Multi-VNF Leaf Topology, on page 19 illustrates the logical leaf topology for the various networks required for the micropod model.

In this figure, Leaf 1 & Leaf 2 are the OAM routers and Leaf 3 & Leaf 4 are the Leaf Routers peering to the backbone router. If additional VNFs are supported, additional Leafs are required.

Figure 8: Ultra M Single and Multi-VNF Leaf Topology


Backbone Router
Backbone Router
100 Gbps active optical cable
Nexus Switch Leaf 3
Nexus Switch Leaf 4
40 Gbps Twinax copper or active optical cable
XL710
VIC1387
XL710
XL710
VIC1387
XL710
XL710
VIC1387
XL710
Compute Nodes
Micropod Nodes
Management Node
10 Gbps CAT5/CAT6 copper cable
To switches above or OAM router
Nexus Switch Shared LOM 1 Leaf 1
Nexus Switch Shared LOM 2 Leaf 2
To switches above or OAM router
429522

As identified in Cisco Nexus Switches, on page 5, 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 multi-VNF deployment are provided in Table 8: Leaf 1 and 2 — Port Interconnects, on page 19 and Table 9: Leaf 3 and 4 — Port Interconnects, on page 20. 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 8: Leaf 1 and 2 — Port Interconnects

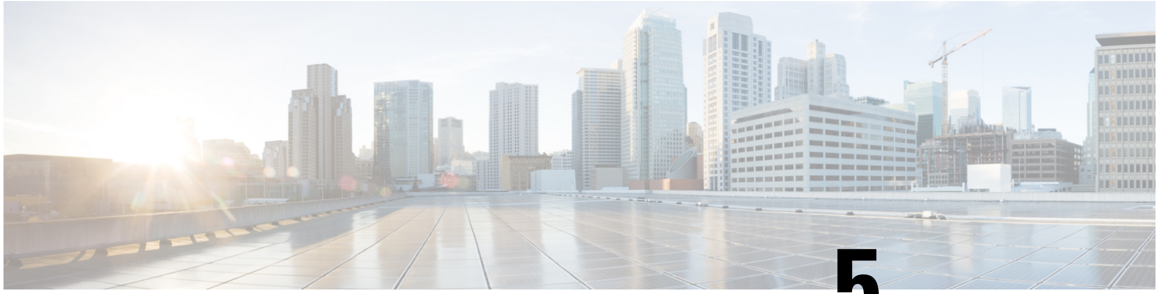
From Leaf Port(s)	**To**			**Notes**
	Device	**Network**	**Port(s)**	
Leaf 1				
Mgmt	OOB	Management		
1	CVIM Mgmt	br_api	sLOM P1	
2, 3, 4	Micropod Nodes	CIMC	sLOM P1	
5-20 (inclusive)	Compute Nodes	CIMC	sLOM P1	Sequential ports based on the number of Compute Nodes - 1 per Compute Node.
49, 50	Leaf2	PortChannel to Leaf1	49, 50	

From Leaf Port(s)	To			Notes
	Device	Network	Port(s)	
Leaf 2				
Mgmt	OOB	Management		
1	CVIM Mgmt	br_api	sLOM P2	
2, 3, 4	Micropod Nodes	CIMC	sLOM P2	
5-20 (inclusive)	Compute Nodes	CIMC	sLOM P2	Sequential ports based on the number of Compute Nodes - 1 per Compute Node.
49, 50	Leaf1	PortChannel to Leaf2	49, 50	

Table 9: Leaf 3 and 4 — Port Interconnects

From Leaf Port(s)	To			Notes
	Device	Network	Port(s)	
Leaf 3				
Mgmt	OOB	Management		
1	CVIM Mgmt	br_mgmt	mLOM P1	
2, 3, 4	Micropod Nodes	Storage, API, Management, Provisioning, External, Tenant	mLOM P1	
5-20 (inclusive)	Compute Nodes	Storage, API, Management, Provisioning, External, Tenant	mLOM P1	Sequential ports based on the number of Compute Nodes - 1 per Compute Node.
21, 22, 23	Micropod Nodes	Provider-Network [DI-Net,Service-Net]	PCIe01 P1	
24-39 (inclusive)	Compute Nodes	Provider-Network [DI-Net,Service-Net]	PCIe01 P1	Sequential ports based on the number of Compute Nodes - 1 per Compute Node.
40,41,42	Micropod Nodes	Provider-Network [DI-Net,Service-Net]	PCIe02 P1	
43-58 (inclusive)	Compute Nodes	Provider-Network [DI-Net,Service-Net]	PCIe02 P1	Sequential ports based on the number of Compute Nodes - 1 per Compute Node.

From Leaf Port(s)	To			Notes
	Device	Network	Port(s)	
59, 60, 61, 62	Leaf4	PortChannel to Leaf4	59, 60, 61, 62	
63, 64	Backbone router	Ingress and Egress to VNF		
Leaf 4				
Mgmt	OOB	Management		
1	CVIM Mgmt	br_mgmt	mLOM P2	
2, 3, 4	Micropod Nodes	Storage, API, Management, Provisioning, External, Tenant	mLOM P2	
5-20 (inclusive)	Compute Nodes	Storage, API, Management, Provisioning, External, Tenant	mLOM P2	Sequential ports based on the number of Compute Nodes - 1 per Compute Node.
21, 22, 23	Micropod Nodes	Provider-Network [DI-Net,Service-Net]	PCIe01 P2	
24-39 (inclusive)	Compute Nodes	Provider-Network [DI-Net,Service-Net]	PCIe01 P2	Sequential ports based on the number of Compute Nodes - 1 per Compute Node.
40, 41, 42	Micropod Nodes	Provider-Network [DI-Net,Service-Net]	PCIe02 P2	
43-58 (inclusive)	Compute Nodes	Provider-Network [DI-Net,Service-Net]	PCIe02 P2	Sequential ports based on the number of Compute Nodes - 1 per Compute Node.
59, 60, 61, 62	Leaf3	PortChannel to Leaf3	59, 60, 61, 62	
63, 64	Backbone Router	Ingress and Egress to VNF		



CHAPTER 5

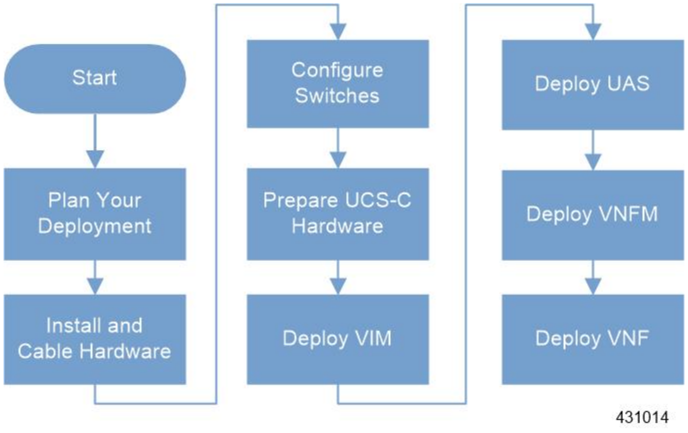
Deploying the Ultra M Solution

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

The following figure illustrates the deployment workflow of VNF on CVIM in Ultra M C2.1 micropod model.

Figure 9: Ultra M C2.1 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 13 provides a general overview and identifies basic requirements for networking the Ultra M solution.

See the Network Definitions (Layer 2 and 3), on page 41 Appendix 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.

- Nexus 93108-TC-FX — https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/hw/n93108tcfx_hig/guide/b_c93108tc_fx_nxos_mode_hardware_install_guide/b_c93108tc_fx_nxos_mode_hardware_install_guide_chapter_01.html
- Nexus 9364C — https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/hw/n9364c_hig/guide/b_c9364c_nxos_mode_hardware_install_guide/b_c9364c_nxos_mode_hardware_install_guide_chapter_01.html
- UCS C220 M5SX Server — https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c/hw/C220M5/install/C220M5.html

Rack Layout

Table 10: Ultra M C2.1 Micropod Deployment Rack Layout, on page 24 provides details for the recommended rack layout for the Ultra M C2.1 micropod deployment model.

Table 10: Ultra M C2.1 Micropod Deployment Rack Layout

Rack Layout for C2.1 - Rack W8		
RU Numbering	**Rack**	
39	SW4	Nexus 9364C
38		

Rack Layout for C2.1 - Rack W8		
RU Numbering	**Rack**	
37	SW3	Nexus 9364C
36		
35	SW2	Nexus 93108TC-FX
34	SW1	Nexus 93108TC-FX
21 to 33	Empty	Empty
20	Compute16	UCSC-C220-M5SX
19	Compute15	UCSC-C220-M5SX
18	Compute14	UCSC-C220-M5SX
17	Compute13	UCSC-C220-M5SX
16	Compute12	UCSC-C220-M5SX
15	Compute11	UCSC-C220-M5SX
14	Compute10	UCSC-C220-M5SX
13	Compute9	UCSC-C220-M5SX
12	Compute8	UCSC-C220-M5SX
11	Compute7	UCSC-C220-M5SX
10	Compute6	UCSC-C220-M5SX
9	Compute5	UCSC-C220-M5SX
8	Compute4	UCSC-C220-M5SX
7	Compute3	UCSC-C220-M5SX
6	Compute2	UCSC-C220-M5SX
5	Compute1	UCSC-C220-M5SX
4	Micropod3	UCSC-C220-M5SX
3	Micropod2	UCSC-C220-M5SX
2	Micropod1	UCSC-C220-M5SX
1	CVIM Manager	UCSC-C220-M5SX

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 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.

Important Refer to Network Planning, on page 24 for information and consideration for planning your network.

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

- Nexus 93108-TC-FX: https://www.cisco.com/c/en/us/support/switches/nexus-93108tc-fx-switch/model.html
- Nexus 9364C: https://www.cisco.com/c/en/us/support/switches/nexus-9364c-switch/model.html

Prepare the UCS C-Series Hardware

UCS-C hardware preparation is performed through the Cisco Integrated Management Controller (CIMC).

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-c220-m5-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

Important 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.

Important The information in this section assumes that the server hardware was properly installed per the information and instructions in Cable the Hardware, on page 26.

For servers based on UCS M5SX boxes set the following for BIOS parameters:

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

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

For other parameters, leave it at their default settings.

Additional steps should be performed to setup C-series pod with Intel NIC. In the Intel NIC testbed, each C-series server has 2, 4-port Intel 710 NIC cards. Ports A, B, and C for each Intel NIC card has to be connected to the respective TOR. Also, ensure that the PCI slot in which the Intel NIC cards are inserted are enabled in the BIOS setting (BIOS > Configure BIOS >Advanced > LOM and PCI Slot Configuration -> All PCIe Slots OptionROM-Enabled and enable respective slots). To identify the slots, check the slot-id information under the Network-Adapter tab listed under the Inventory link on the CIMC pane. All the Intel NIC ports should be displayed in the BIOS summary page under the Actual Boot Order pane, as IBA 40G Slot xyza with Device Type is set to PXE.

Table 11: Cisco UCS BIOS Options, on page 27 lists the non-default parameters that must be configured per server type.

Table 11: Cisco UCS BIOS Options

Parameters and Settings	Description
Processor Configuration	
Enhanced Intel Speedstep	Whether the processor uses Enhanced Intel SpeedStep Technology, which allows the system to dynamically adjust processor voltage and core frequency. This technology can result in decreased average power consumption and decreased average heat production. This can be one of the following: • disabled—The processor never dynamically adjusts its voltage or frequency. • enabled—The processor utilizes Enhanced Intel SpeedStep Technology and enables all supported processor sleep states to further conserve power. • Platform Default—The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Default value**: disabled We recommend that you contact your operating system vendor to make sure the operating system supports this feature.

Parameters and Settings	Description
Turbo Boost	Whether the processor uses Intel Turbo Boost Technology, which allows the processor to automatically increase its frequency if it is running below power, temperature, or voltage specifications. This can be one of the following: • disabled—The processor does not increase its frequency automatically. • enabled—The processor utilizes Turbo Boost Technology if required. • Platform Default—The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Default value**: disabled
Hyper Threading	Whether the processor uses Intel Hyper-Threading Technology, which allows multithreaded software applications to execute threads in parallel within each processor. This can be one of the following: • disabled—The processor does not permit hyperthreading. • enabled—The processor allows for the parallel execution of multiple threads. • Platform Default—The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Default value**: enabled We recommend that you contact your operating system vendor to make sure the operating system supports this feature.

Parameters and Settings	Description
Core Multi Processing	Sets the state of logical processor cores in a package. If you disable this setting, Hyper Threading is also disabled. This can be one of the following: all—Enables multi processing on all logical processor cores. 1 through 10—Specifies the number of logical processor cores that can run on the server. To disable multi processing and have only one logical processor core running on the server, select 1. Platform Default—The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Default value**: all We recommend that you contact your operating system vendor to make sure the operating system supports this feature.
Power/Performance	
CPU Performance	Sets the CPU performance profile for the server. This can be one of the following: • enterprise—All prefetchers and data reuse are disabled. • high-throughput—All prefetchers are enabled, and data reuse is disabled. • hpc—All prefetchers and data reuse are enabled. This setting is also known as high performance computing. • Platform Default—The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Default value**: high-throughput
Workload Configuration	Set the value of this parameter as IO sensitive.

Parameters and Settings	Description
Fan Policy	Set the Fan Policy for the server to **High Power** as mentioned in the https://www.cisco.com/en/US/docs/unified_computing/ucs/c/sw/gui/config/guide/1.5/b_Cisco_UCS_C-series_GUI_Configuration_Guide.151_chapter_011.html#concept_8CB787DF70304E98BE25D120466418B9. This setting can be used for server configurations that require fan speeds ranging from 60% to 85%. This policy is ideal for servers that contain PCIe cards that overheat easily and have high temperatures. The minimum fan speed set with this policy varies for each server, but it is approximately in the range of 50 to 85%.
Memory	
NUMA	Whether the BIOS supports NUMA. This can be one of the following: • disabled—The BIOS does not support NUMA. • enabled—The BIOS includes the ACPI tables that are required for NUMA-aware operating systems. If you enable this option, the system must disable Inter-Socket Memory interleaving on some platforms. • Platform Default—The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Default value**: enabled

Deploy the Virtual Infrastructure Manager

Within the Ultra M solution, Cisco Virtualized Infrastructure Manager (CVIM) functions as the virtual infrastructure manager (VIM).

The method by which the VIM is deployed depends on the architecture of your Ultra M model. For the micropod model, see the https://www.cisco.com/c/en/us/td/docs/net_mgmt/network_function_virtualization_Infrastructure/2_4_3/install_guide/Cisco_VIM_Install_Guide_2_4_3/Cisco_VIM_Install_Guide_2_4_3_chapter_00.html.

Deploying VNFs Using AutoVNF in Generic Mode

This section describes the following topics:

- Introduction, on page 31

Introduction

USP-based VNFs can be deployed using a AutoVNF instance in generic mode. In this scenario, AutoVNF VM (in HA mode) is deployed on the VIM and is used to deploy VNFM and VNF(s).

Important AutoVNF deploys Cisco Elastic Services Controller (ESC) as the VNFM and is only supported VNFM in this release.

A single AutoVNF can deploy one or more VNFs in one or more tenants within the same VIM.

VNF Deployment Automation Overview

Figure 10: AutoVNF Deployment Automation Workflow for a Single VNF, on page 32 and Figure 11: AutoVNF Deployment Automation Workflow for a Multi-VNF, on page 33 provide an overview of the VNF deployment automation process for when using AutoVNF in generic mode. Details are provided in Table 12: VNF Deployment Automation Workflow Descriptions, on page 33.

NOTES:

- The workflow described in this section is supported only with VNF deployments performed through AutoVNF and that are based on OSP 10.
- This information assumes that you have deployed the NFVI and VIM.
- This information assumes that all artifacts required during configuration must be pre-created in OpenStack.

Figure 10: AutoVNF Deployment Automation Workflow for a Single VNF

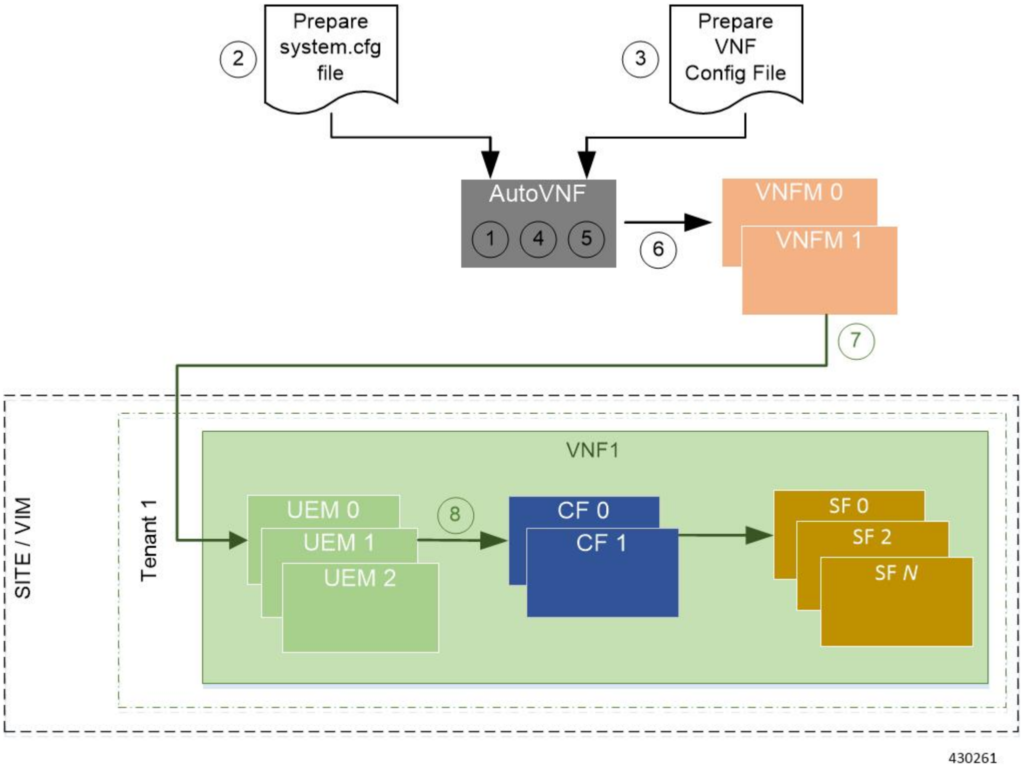


Figure 11: AutoVNF Deployment Automation Workflow for a Multi-VNF

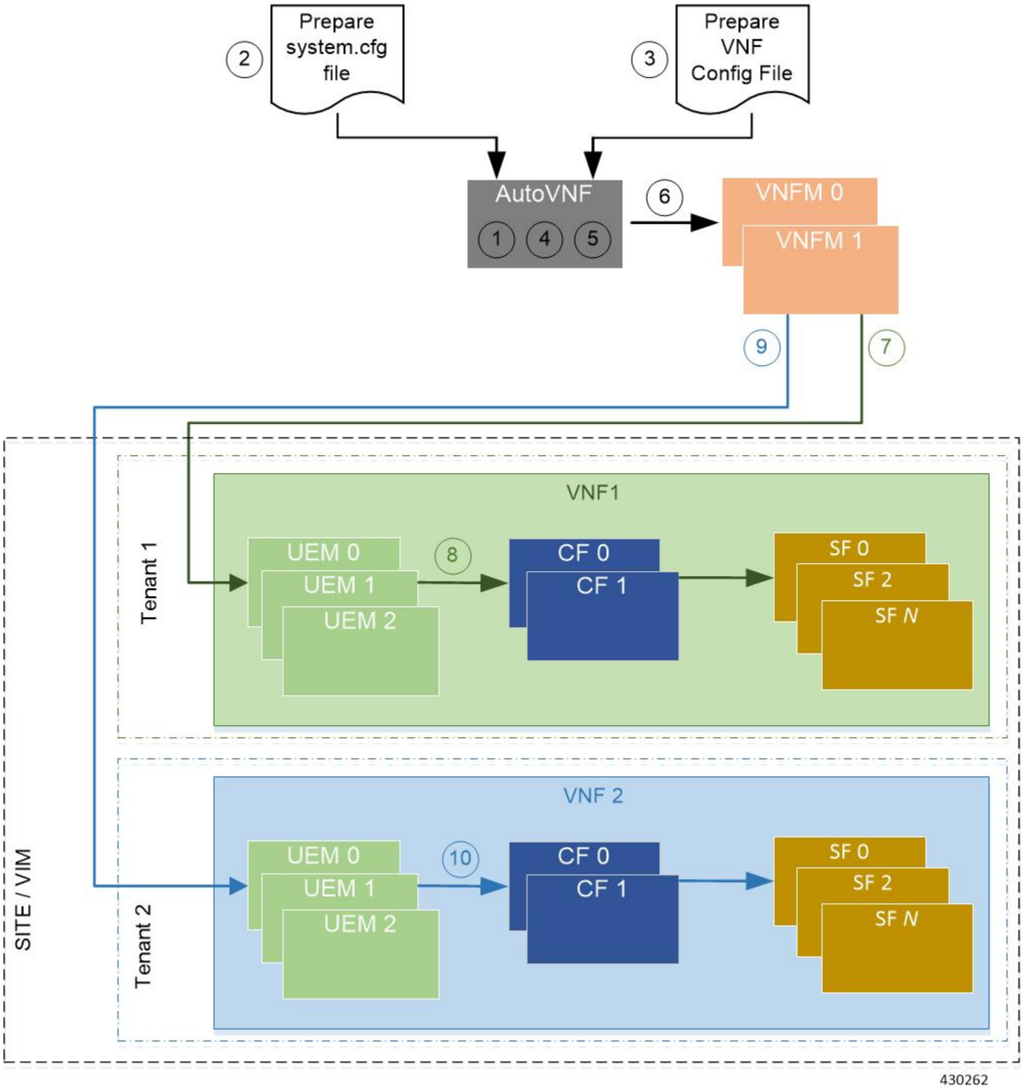


Table 12: VNF Deployment Automation Workflow Descriptions

Callout	Description
1	Deploy AutoVNF using the *boot_uas.py* script provided as part of the release ISO.
2	Prepare the *system.cfg* file to the AutoVNF VM. This file provides the VNF's Day-0 configuration.
3	Prepare the AutoVNF configuration file that is used by AutoVNF to initiate the VNFM and VNF deployment process. This file includes the configuration information required to deploy VNFM and all the VNF components (VNFCs) such as secure tokens, network catalogs, VDU catalogs, and VDUs.

Callout	Description
4	On the AutoVNF VM, load and commit the AutoVNF configuration file prepared in the previous step. Once commited, activate the loaded AutoVNF configuration file to deploy the VNFMs.
5	Once VNFMs are ready, AutoVNF pushes the artifacts to bring up the VNF.
6	AutoVNF passes the VNF configuration to the VNFM VM instance. **Note** In this deployment model, AutoVNF in NFVO mode brings up the VNFMs and they are not pre-created. It ensures that the various VM catalogs pertaining to other VNFCs are on-boarded by the VNFM. It accomplishes this through a number of YANG-based definitions which are then used to configure various aspects of the virtualized environment using REST and NETCONF APIs. That VNFM mounts the VNFC catalogs and works with AutoVNF to deploy the various components that comprise the desired VNF use-case (e.g. UGP).
7, 9	The VNFM leverages the VNFC information to deploy the UEM VMs cluster. Though the USP architecture represents a single VNF to other network elements, it is comprised of multiple VM types each having their own separate catalogs. The UEM component of the USP works with the VNFM to deploy these catalogs based on the intended VNF use case (e.g. UGP, etc.).
8, 10	The UEM processes the Day-0 configuration information it received from the VNFM and deploys the Control Function (CF) and Service Function (SF) VNFC VMs. Once all the VNF components (VNFCs) have been successfully deployed, AutoVNF notifies AutoDeploy. **Important** In a multi-VNF environment, the VNFs are deployed concurrently.

Pre-VNF Installation Verification

Prior to installing the USP, please ensure that the following is true:

- The prerequisite hardware is installed and operational with network connectivity.
- The prerequisite software is installed and configured and functioning properly:
  - You have administrative rights to the operating system.
  - VIM Orchestrator is properly installed and operational.
  - VIM components are properly installed and operational. This configuration includes networks, flavors, and sufficient quota allocations to the tenant.

Note Supported and/or required flavors and quota allocations are based on deployment models. Contact your Cisco representative for more information.

- You have administrative rights to the OpenStack setup.

- The Cisco USP software ISO has been downloaded and is accessible by you.

Deploy the USP-based VNF

The AutoVNF software roles within the Ultra Automation Services (UAS) is used to automate the USP-based VNF deployment. The automated deployment process through AutoVNF is described in VNF Deployment Automation Overview, on page 31.

To deploy the USP-based VNF using AutoDeploy:

1. Onboard the USP ISO, on page 35.
2. Extract the UAS Bundle, on page 36.
3. Deploy the AutoVNF VM, on page 37.
4. Activate the AutoVNF Configuration Files, on page 39.

Onboard the USP ISO

The files required to deploy the USP components are distributed as RPMs (called “bundles”) in a single ISO package. They are maintained using YUM on the Onboarding Server. The following bundles are part of the ISO:

USP Bundle Name	Description
usp-em-bundle	The Element Manager (EM) Bundle RPM containing images and metadata for the Ultra Element Manager (UEM) module.
usp-uas-bundle	The Ultra Automation Services Bundle RPM containing AutoIT, AutoDeploy, AutoVNF, Ultra Web Services (UWS), and other automation packages.
usp-ugp-bundle	The Ultra Gateway Platform (UGP) Bundle RPM containing images for Ultra Packet core (VPC-DI). This bundle contains non-trusted images.
usp-vnfm-bundle	The VNFM Bundle RPM containing an image and a boot-up script for ESC (Elastic Service Controller).
usp-yang-bundle	The Yang Bundle RPM containing YANG data models including the VNFD and VNFR.
usp-auto-it-bundle	The bundle containing the AutoIT packages required to deploy the UAS.
ultram-manager	This package contains the script and relevant files needed to deploy the Ultra Health Service.

In addition to the bundles, the ISO bundle also includes scripts used to deploy the bundles including UAS.

Before proceeding with these instructions, ensure that the prerequisites identified in *USP Installation Prerequisites* chapter of the *Cisco Ultra Services Platform Deployment Automation Guide* have been met.

To onboard the ISO package:

1. Log on to the Onboarding Server.
2. Download the USP ISO bundle and related files pertaining to the release.
3. Create a mount point on the Onboarding Server and mount the ISO package:

   ```
   mkdir /var/usp-iso
   ```

4. Mount the USP ISO.

   ```
   sudo mount -t iso9660 -o loop <ISO_download_directory>/<ISO_package_name>
   /var/usp-iso
   ```

   Example: The following command mounts the ISO bundle called *usp-5_5_0-1255.iso* located in a directory called *5_5_0-1283* to */var/usp-iso*:

   ```
   sudo mount -t iso9660 -o loop 5_5_0-1064/usp-5_5_0-1064.iso /var/usp-iso
   ```

   ```
   mount: /dev/loop1 is write-protected, mounting read-only
   ```

5. Verify the mount configuration.

   ```
   df -h
   ```

   Example output:

   ```
   Filesystem      Size  Used Avail Use% Mounted on
   /dev/sda2       187G  178G  316M 100% /
   devtmpfs         63G     0   63G   0% /dev
   tmpfs            63G  4.0K   63G   1% /dev/shm
   tmpfs            63G  1.4M   63G   1% /run
   tmpfs            63G     0   63G   0% /sys/fs/cgroup
   /dev/sda1       477M  112M  336M  25% /boot
   tmpfs            13G     0   13G   0% /run/user/0
   /dev/loop1      4.2G  4.2G     0 100% /var/usp-iso
   ```

6. Proceed to Extract the UAS Bundle, on page 36.

Extract the UAS Bundle

Once the USP ISO has been mounted, the UAS bundle must be extracted from the ISO in order to prepare the configuration files required for deployment.

These instructions assume you are already logged on to the Onboarding Server.

To extract the UAS bundle:

1. Navigate to the tools directory within the ISO mount.

   ```
   cd /var/usp-iso/tools/
   ```

2. Launch the *usp-uas-installer.sh* script.

   ```
   sudo ./usp-uas-installer.sh
   ```

   The script extracts the files that comprise the UAS bundle to */opt/cisco/usp/uas-installer*.

3. Verify that files have been extracted.

   Example output:

   ```
   ll /opt/cisco/usp/uas-installer
   ```

```
total 12
drwxr-xr-x. 5 root root 4096 May 11 08:04 common
drwxr-xr-x. 2 root root 4096 May 11 08:04 images
drwxr-xr-x. 2 root root 4096 May 11 08:04 scripts
```

```
ll /opt/cisco/usp/uas-installer/images/
```

```
total 707580
-rw-r--r--. 1 root root 723898880 May 10 15:40 usp-uas-1.0.0-601.qcow2
```

```
ll /opt/cisco/usp/uas-installer/scripts/
```

```
total 56
-rwxr-xr-x. 1 root root  5460 May 11 08:04 autoit-user.py
-rwxr-xr-x. 1 root root  4762 May 11 08:04 encrypt_account.sh
-rwxr-xr-x. 1 root root  3945 May 11 08:04 encrypt_credentials.sh
-rwxr-xr-x. 1 root root 13846 May 11 08:04 uas-boot.py
-rwxr-xr-x. 1 root root  5383 May 11 08:04 uas-check.py
-rwxr-xr-x. 1 root root 10385 May 11 08:04 usp-tenant.py
```

4. Proceed to Deploy the AutoVNF VM, on page 37.

Deploy the AutoVNF VM

The VM for AutoVNF is deployed using *boot_uas.py* script provided with the UAS bundle. The script is located in the following directory:

/opt/cisco/usp/bundles/uas-bundle/tools

This script includes a number of deployment parameters for the VM. These parameters are described in the help information pertaining to the script which can be accessed by executing the following command:

```
./boot_uas.py -h
```

For the help information, see the *boot_uas.py Help* Appendix in the *Cisco Ultra Services Platform Deployment Automation Guide*.

Important These instructions assume you are already logged on to the Onboarding Server.

To deploy the AutoVNF VM:

1. Navigate to the directory containing the *boot_uas.py* file.

   ```
   cd /opt/cisco/usp/bundles/uas-bundle/tools
   ```

2. Deploy the AutoVNF VM.

   ```
   ./boot_uas.py  --autovnf --openstack --image <image_name> --flavor
   <flavor_name> --net <network_name>
   ```

   There are additional arguments that can be executed with this script based on your deployment scenario. For details, see the *boot_uas.py Help* Appendix in the *Cisco Ultra Services Platform Deployment Automation Guide*.

Important Both version 2 and 3 of OpenStack Keystone APIs are supported. You can specify the desired version using the **--os_identity_api_version** argument with this script. For example to specify the use of version 3, add the argument **--os_identity_api_version 3**. The default is version 2.

Upon executing the script, you are prompted to enter user crendentials for performing operations within the AutoVNF VM.

3. Provide the requested information.

   - **AutoVNF VM Login Password**: The password for the default user account, which is named *ubuntu*.
   - **AutoVNF API Access password for "admin"**: The password for the ConfD administrator user, which is named admin.
   - **AutoVNF API Access password for "oper"**: The password for the ConfD operator user, which is named oper.
   - **AutoVNF API Access password for "security"**: The password for the ConfD security administrator user, which is named security-admin.

Important Ensure that all passwords meet the requirements specified in *Password Requirements and Login Security* section in the *Cisco Ultra Services Platform Deployment Automation Guide*.

4. Log on to the AutoVNF VM as *ubuntu*. Use the password that was created earlier for this user.
5. Become the root user.

   ```
   sudo -i
   ```

6. Prepare the *system.cfg* file. This will serve as the Day-0 config for the VNF. Refer to Sample system.cfg File, on page 69 for an example configuration file.

Important Though administrative user credentials can be specified in clear text in the system.cfg file, it is not recommended. For security purposes, it is recommended that you configure a secure token for the user account in the VNF configuration file and reference that file as part of the VDU catalog pertaining to the CF using the **login-credential** parameter. In the *system.cfg* file, use the *$CF_LOGIN_USER* and *$CF_LOGIN_PASSWORD* variables as follows to call the values configured for the secure token:

```
configure
  context local
    administrator $CF_LOGIN_USER password $CF_LOGIN_PASSWORD ftp
```

7. Prepare the AutoVNF configuration file.

   This file provides the VNF configuration information used by AutoVNF during the deployment process. A sample configuration file is provided for reference in Sample AutoVNF Configuration File, on page 45.

8. Save the AutoVNF configuration file to your home directory on the AutoVNF VM.
9. Upload the USP ISO to home directory on AutoVNF.
10. Proceed to Activate the AutoVNF Configuration Files, on page 39.

Activate the AutoVNF Configuration Files

Once you have completed preparing your AutoVNF configuration files, you must load the configuration and activate the deployment.

Important User credentials are configured through Secure Tokens specified in the configuration file. Ensure that passwords configured with Secure Token meet the requirements specified in the *Password Requirements and Login Security* section of *Cisco Ultra Services Platform Deployment Automation Guide*.

Once activated, AutoVNF proceeds with the deployment automation workflow as described in VNF Deployment Automation Overview, on page 31.

Important These instructions assume you are already logged on to the AutoVNF VM as the *root* user and that your configuration files have been prepared for your deployment as per the information and instructions in Deploy the AutoVNF VM, on page 37. These instructions also assume that AutoVNF has access to the VNFC image files (either locally or on a remote server) provided with the USP ISO.

To activate the USP deployment using AutoVNF:

1. Login to the ConfD CLI as the admin user.

```
confd_cli -u admin -C
```

2. Enter the ConfD configuration mode.

```
config
```

3. Load the AutoVNF configuration file to load the VNFM and VNF information into the AutoVNF database.

```
load merge <your_autovnf_file_name> .cfg
commit
end
```

Important If you are performing this process as a result of an upgrade or redeployment, you must use the load replace variant of this command:

```
load replace <your_autovnf_file_name> .cfg
commit
end
```

4. Activate the AutoVNF configuration file.

```
activate nsd <nsd_name>
```

Important The output of this command is a transaction-id which can be used to monitor the deployment progress. If need be, the VIM deployment can be deactivated using the **deactivate** variant of this command.

5. Once VNFM is deployed and ready, activate the VNF NSD configuration file.

```
activate nsd <nsd_name> vnfd <vnf>
```

Important The output of this command is a transaction-id which can be used to monitor the deployment progress. If need be, the VIM deployment can be deactivated using the **deactivate** variant of this command.

6. Monitor the progress of the deployment by viewing transaction logs:

   ```
   show log <transaction_id> | display xml
   ```

   transaction_id is the ID displayed as a result of the **activate-deployment** command.

   The logs display status messages for each node in each VNF that the configuration file defines. Example success messages for the different components deployed through AutoVNF are shown below:

   - VNF:

     ```
     Fri May 12 21:44:35 UTC 2017 [Task: 1494624612779/tb1vnfd2] Successfully completed
     all Vnf Deployments
     ```

   - Entire Deployment:

     ```
     Fri May 12 21:57:38 UTC 2017 [Task: 1494624612779] Success
     ```

Important If there are any issues seen when executing the above commands, see the *Monitoring and Troubleshooting the Deployment* section in the *Cisco Ultra Services Platform Deployment Automation Guide*.

Upgrading/Redeploying the Stand-alone AutoVNF VM Instance

Use the following procedure to upgrade or redeploy the AutoVNF software image in scenarios where AutoVNF was brought up as stand-alone instance.

Important These instructions assume you are already logged on to the Onboarding Server.

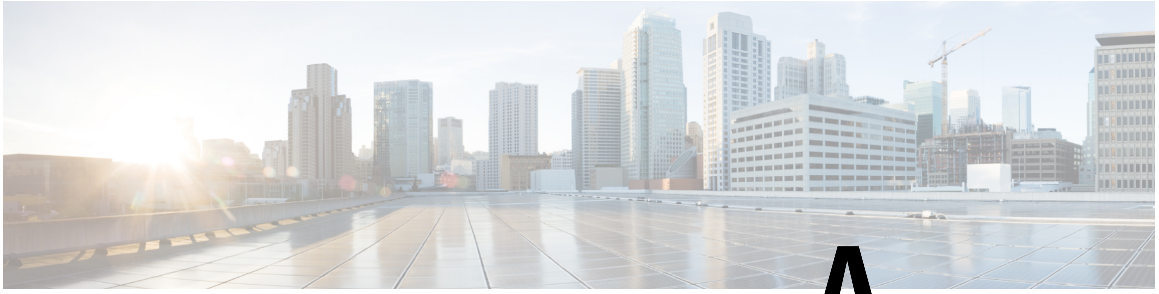
1. Delete the AutoVNF VM instance.

   ```
   ./boot_uas.py --openstack --autovnf --delete  <transaction_id>
   ```

2. *Optional.* If required remove the OpenStack artifacts which were created manually to bring up AutoVNF.

3. Follow the procedures in Deploy the USP-based VNF, on page 35 to redeploy AutoVNF with the new software version.

Note Upgrading or redeploying the VNF can be performed as part of this process or it can be performed separately. For details and instructions, see the *Upgrading/Redeploying VNFs Deployed Through a Stand-alone AutoVNF Instance* section in the *Cisco Ultra Services Platform Deployment Automation Guide*.



APPENDIX A

Network Definitions (Layer 2 and 3)

Table 13: Layer 2 and 3 Network Definition, on page 41 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 13: Layer 2 and 3 Network Definition, on page 41 are configured directly on the UCS hardware via CIMC. Other parameters are configured as part of the CVIM configuration. This configuration is done through various configuration files depending on the parameter:

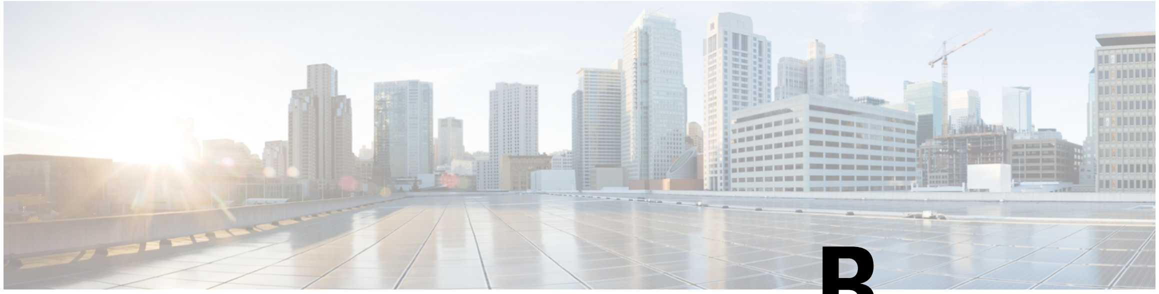
- setup_data.yaml
- AutoVNF Configuration file for the pod

Table 13: 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 CVIM mgmt node Only							
3522	**10.86.67.0** **10.86.67.99/24**	**10.86.67.99**			Internet access required: - 1 IP Address for CVIM Mgmt - 1 IP for default gateway	On CVIM Manger Node hardware	Yes
External – Floating IP Addresses							

VLAN ID / Range	Network	Gateway	IP Range Start	IP Range End	Description	Where Configured	Routable?
1519	10.84.109.64 /27	10.84.109.65			Routable addresses required: 4 Floating IP Addresses per VNF for management VMs (CF, VNFM, UEM, and UAS software modules) - 1 IP for default gateway	*setup_data.yaml*	Yes
Management/Provisioning							
105	**192.168.50.0/ 24**		**192.168.50.100**	**192.168.50.254**	Required to provision all configuration via PXE boot from CVIM Manager node for Micropod and Compute. Management network is used for communication between OpenStack elements using Openstack APIs	*setup_data.yaml*	No
IPMI-CIMC							
106	**192.168.60.0/ 24**		**192.168.60.100**	**192.168.60. 254**		On UCS servers through CIMC	No
Tenant (Virtio)							
	11.117.0.0/ 24				All tenant networks. (MLOM)	*setup_data.yaml*	No
Storage (Virtio)							

VLAN ID / Range	Network	Gateway	IP Range Start	IP Range End	Description	Where Configured	Routable?
18	**11.118.0.0/ 24**				Transit network for storage back-end. Storage traffic between VMs and Ceph nodes (MLOM).	*setup_data.yaml*	No
API (Virtio)							
3522	**10.86.67.0/ 24**				Clients connect to API network to interface with OpenStack APIs. OpenStack Horizon dashboard. Default gateway for HAProxy container.	*setup_data.yaml*	Yes
350: **399**					Tenant based virtio networks on openstack.	*setup_data.yaml*	
SR-IOV ((Phys-PCIe1, Phys PC1e2)							
2001: **2050** **2111, 2112**					Tenant SRIOV network on openstack. (Intel NIC) NOTE: A unique VLAN from this range is used by each VNF for the DI-internal network.		Yes
NOTE: Bold underlined text is provided as example configuration information. Your deployment requirements will vary. * You can ensure that the same floating IP address be assigned to the AutoVNF, CF, UEM, and VNFM after a VM restart by configuring parameters in the AutoVNF configuration file.							



APPENDIX B

Sample AutoVNF Configuration File

The sample AutoVNF configuration file (*autovnf.cfg*) includes all the configuration information required to deploy VNFM and all the VNF components (VNFCs) such as secure tokens, network catalogs, VDU catalogs, and VDUs.

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.

Sample autovnf.cfg for 1xVNF

```
uas-mode generic
nsd TB8-autovnf1_vpc
 version      6.2
 vim-identity default_openstack_vim
 vnf-package    [ usp_6_2 ]
 vld vnf-mgmt
  vl-type          management
  network-instance ext-net
 !
 vld vnf-orch
  vl-type          orchestration
  network-instance vnf-orch
 !
 vld vnf1_svc_1
  vl-type          service
  network-instance nic1_port2_sriov2
 !
 vld vnf1_svc_2
  vl-type          service
  network-instance nic2_port1_sriov3
 !

 vnfd vnf1_esc
  vnf-type          esc
  version           6.2
  high-availability true
  configuration openstack.endpoint publicURL
  configuration secure-login false
  configuration boot-time 1800
  configuration set-vim-instance-name true
  external-connection-point vnf1_esc
   connection-point eth1
   ip-address       10.84.109.88
   floating-ip disabled
  !
  vnfc vnf1_esc
   health-check enabled
   health-check probe-frequency 10
   health-check probe-max-miss 6
   health-check retry-count 3
   health-check recovery-type restart-then-redeploy
   health-check boot-time 300
   vdu vdu-id vdu-esc-vnf1
   vdu flavor ESC_VNF1_FLV
   connection-point eth0
    virtual-link service-vl vnf-orch
   !
   connection-point eth1
    virtual-link service-vl vnf-mgmt
   !
  !
 !


 vnfd vpc1
  vnf-type          ugp
  version           6.2
  high-availability true
  vnfm vnfd vnf1_esc
  configuration internal-network-mtu 1500
  configuration boot-time 1800
  configuration domain-name cisco.com
  configuration set-vim-instance-name true
  configuration dns-server 10.84.96.130
  !
 vld vnf1_di_1
  network-instance nic1_port1_sriov0
 !
 vld vnf1_di_2
  network-instance nic2_port2_sriov1
 !
  external-connection-point cf1
   connection-point eth3
   ip-address       10.84.109.90
   floating-ip disabled
  !
  external-connection-point em1
   connection-point eth1
   ip-address       10.84.109.89
   floating-ip disabled
  !
  vnfc em1
   health-check enabled
   health-check probe-frequency 10
   health-check probe-max-miss 6
   health-check retry-count 6
   health-check recovery-type restart-then-redeploy
   health-check boot-time 300
   vdu vdu-id vdu-em-vnf1
   vdu flavor EM_VNF1_FLV
   number-of-instances 1
   connection-point eth0
    virtual-link service-vl vnf-orch
   !
   connection-point eth1
    virtual-link service-vl vnf-mgmt
   !
  !
  vnfc cf1
   health-check enabled
   health-check probe-frequency 10
   health-check probe-max-miss 6
   health-check retry-count 6
   health-check recovery-type restart-then-redeploy
   health-check boot-time 300
   vdu vdu-id vdu-cf-vnf1
   vdu flavor CF_VNF1_FLV
   number-of-instances 1
   aggregate-connection-points DI_INTERFACE
    aggregate-connection-point eth0
    !
    aggregate-connection-point eth1
    !
   !
   connection-point eth0
    virtual-link internal-vl vnf1_di_1
   !
   connection-point eth1
    virtual-link internal-vl vnf1_di_2
   !
   connection-point eth2
    virtual-link service-vl vnf-orch
   !
   connection-point eth3
    virtual-link service-vl vnf-mgmt
   !
  !
  vnfc sf1
   health-check enabled
   health-check probe-frequency 10
   health-check probe-max-miss 6
   health-check retry-count 6
   health-check recovery-type restart-then-redeploy
   health-check boot-time 300
   vdu vdu-id vdu-sf-vnf1
   vdu flavor SF_VNF1_FLV
   number-of-instances 10
   aggregate-connection-points DI_INTERFACE
    aggregate-connection-point eth0
    !
    aggregate-connection-point eth1
    !
   !
   connection-point eth0
    virtual-link internal-vl vnf1_di_1
   !
   connection-point eth1
    virtual-link internal-vl vnf1_di_2
   !
   connection-point eth2
    virtual-link service-vl vnf-orch
   !
   connection-point eth3
    virtual-link service-vl vnf1_svc_1
   !
   connection-point eth4
    virtual-link service-vl vnf1_svc_2
   !
  !
 !
!
```

```
secure-token cimc
 user     admin
 password Csco@123
!
secure-token ssh-baremetal
 user     admin
 password Csco@123
!
secure-token scm-admin
 user     admin
 password Csco@123
!
secure-token scm-oper
 user     admin
 password Csco@123
!
secure-token scm-security
 user     security-admin
 password Csco@123
!
secure-token stack
 user     core
 password Csco@123
!
secure-token vim-admin-creds
 user     admin
 password Px7OAZIDhjYdhUPG
!
secure-token vim-core-creds
 user     core
 password Csco@123
!
secure-token login
 user     ubuntu
 password Csco@123
!
secure-token em_login
 user     ubuntu
 password Csco@123
!
secure-token staros
 user     admin
 password Csco@123
!
secure-token esc_netconf
 user     admin
 password Csco@123
!
secure-token esc_login
 user     admin
 password Csco@123
!
secure-token cf_login
 user     admin
 password Csco@123
!
scm scm
 admin    scm-admin
 oper     scm-oper
 security scm-security
!
vnf-packaged usp_6_2
 checksum           e18b9b7bb205cb69f0af80ef9259c968
 location           /home/ubuntu/usp-6_2_b3-6122.iso
 validate-signature false
 configuration staros1
  external-url /home/ubuntu/system-vnf1.cfg
 !
 configuration staros2
  external-url /home/ubuntu/system-vnf2.cfg
 !
!
vdu vdu-esc-vnf1
 vdu-type           cisco-esc
 login-credential   esc_login
 netconf-credential esc_netconf
 image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   2
 flavor ram     4096
 flavor root-disk 40
 flavor ephemeral-disk 0
 flavor swap-disk 0
!

vdu vdu-em-vnf1
 vdu-type         element-manager
 login-credential em_login
 scm              scm
  image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   2
 flavor ram     4096
 flavor root-disk 40
 flavor ephemeral-disk 0
 flavor swap-disk 0
!

vdu vdu-cf-vnf1
 vdu-type         control-function
 login-credential cf_login
  image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   8
 flavor ram     16384
 flavor root-disk 6
 flavor ephemeral-disk 0
 flavor swap-disk 0
 upp cores     30
 upp crypto-cores 0
 upp service-mode vpc
 upp disable-mcdma false
 upp disable-numa false
 upp param DI_INTERFACE
  value BOND:TYPE:i40evf-1,TYPE:i40evf-2
 !
 upp param DI_INTERFACE_VLANID
  value 2111
 !
 upp param MULTI_SEG_MBUF_ENABLE
   value 0
 !
 ned netconf
  ned-id         cisco-staros-nc
  port-number    830
  authentication staros
 !
 configuration staros_config.txt
  apply-at day-zero
  package  staros1
  !source-url file:///opt/cisco/usp/uploads/system-vnf1.cfg
 !
 volume boot cf-boot-vnf1
 volume storage cf-cdr-vnf1
 !
!
vdu vdu-sf-vnf1
 vdu-type    session-function
  image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   20
 flavor ram     98304
 flavor root-disk 6
 flavor ephemeral-disk 0
 flavor swap-disk 0
 flavor cpu-policy dedicated
 flavor cpu-thread-policy isolate
 flavor numa-nodes 0
 !
 flavor numa-nodes 1
 !
 upp cores     35
 upp crypto-cores 0
 upp service-mode vpc
 upp disable-mcdma false
 upp disable-numa false
 upp param CARDTYPE
  value 0x42030100
 !
 upp param DI_INTERFACE
  value BOND:TYPE:i40evf-1,TYPE:i40evf-2
 !
 upp param DI_INTERFACE_VLANID
  value 2111
 !
 upp param IFTASK_CORES
   value 35
 !
 upp param IFTASK_MCDMA_CORES
  value 50
 !
 upp param MULTI_SEG_MBUF_ENABLE
  value 0
 !
!
volume cf-boot-vnf1
 type                LUKS
 size                16
 bus                 ide
 bootable            true
 preserve-on-upgrade false
!
volume cf-cdr-vnf1
 type                LUKS
 size                200
 bus                 ide
 bootable            false
 preserve-on-upgrade false
!
volume cf-boot-vnf2
 type                LUKS
 size                16
 bus                 ide
 bootable            true
 preserve-on-upgrade false
!
volume cf-cdr-vnf2
 type                LUKS
 size                200
 bus                 ide
 bootable            false
 preserve-on-upgrade false
!


vnf-rackd TB8-vnf-rack

 host-aggregate TB8-esc-em-vnf1
  host micropod-1
  !
  host micropod-3
  !
 !

 host-aggregate TB8-esc-em-vnf2
  host micropod-2
  !
  host micropod-3
  !
 !


 host-aggregate TB8-vnf1-cf
  host micropod-1
  !
  host micropod-3
  !
 !

 host-aggregate TB8-vnf2-cf
  host micropod-2
  !
  host micropod-3
  !
 !

 host-aggregate TB8-vnf_1_2-sf
  host compute-1
  !
  host compute-2
  !
  host compute-3
  !
  host compute-4
  !
  host compute-5
  !
  host compute-6
  !
  host compute-7
  !
  host compute-8
  !
  host compute-10
  !
  host compute-11
  !
  host compute-12
  !
  host compute-13
  !
  host compute-14
  !
  host compute-15
  !

 !
!

vim vim1
 api-version v2
 auth-url    http://10.86.67.72:5000/v2.0
 user        vim-admin-creds
 tenant      admin
!
vim default_openstack_vim
 api-version v2
 auth-url    http://10.86.67.72:5000/v2.0
 user        vim-core-creds
 tenant      core
!


network-instance ext-net
 ip-prefix 10.84.109.64/27
 type      vlan
 dhcp      true
 gateway   10.84.109.65
!

network-instance vnf-orch
 ip-prefix 182.37.180.0/24
 type      vlan
 dhcp      true
 gateway   182.37.180.1
!

network-instance nic1_port1_sriov0
 ip-prefix 192.168.10.0/24
 type      sriov-flat
 dhcp      true
 vlan-tag  false
 !physnet   phys_sriov0
!
network-instance nic1_port2_sriov2
 ip-prefix 192.168.11.0/24
 type      sriov-flat
 dhcp      true
 vlan-tag  false
 !physnet   phys_sriov2
!
network-instance nic2_port1_sriov3
 ip-prefix 192.168.12.0/24
 type      sriov-flat
 dhcp      true
 vlan-tag  false
 !physnet   phys_sriov3
!
network-instance nic2_port2_sriov1
```

```
 ip-prefix 192.168.13.0/24
 type      sriov-flat
 dhcp      true
 vlan-tag  false
 !physnet   phys_sriov1
!

vim-artifactd vim_art_rack
 vnf-rack [ TB8-vnf-rack ]
!
```

Sample autovnf.cfg for 2xVNF

```
uas-mode generic
nsd TB8-autovnf1_vpc
 version      6.2
 vim-identity default_openstack_vim
 vnf-package  [ usp_6_2 ]
 vld vnf-mgmt
  vl-type          management
  network-instance ext-net
 !
 vld vnf-orch
  vl-type          orchestration
  network-instance vnf-orch
 !
 vld vnf1_svc_1
  vl-type          service
  network-instance nic1_port1_sriov0
 !
 vld vnf1_svc_2
  vl-type          service
  network-instance nic1_port2_sriov2
 !
 vld vnf2_svc_1
  vl-type          service
  network-instance nic2_port1_sriov3
 !
 vld vnf2_svc_2
  vl-type          service
  network-instance nic2_port2_sriov1
 !

 vnfd vnf1_esc
  vnf-type          esc
  version           6.2
  high-availability true
  configuration openstack.endpoint publicURL
  configuration secure-login false
  configuration boot-time 1800
  configuration set-vim-instance-name true
  external-connection-point vnf1_esc
   connection-point eth1
   ip-address       10.84.109.88
   floating-ip disabled
  !
  vnfc vnf1_esc
   health-check enabled
   health-check probe-frequency 10
   health-check probe-max-miss 6
   health-check retry-count 3
   health-check recovery-type restart-then-redeploy
   health-check boot-time 300
   vdu vdu-id vdu-esc-vnf1
   vdu flavor ESC_VNF1_FLV
   connection-point eth0
    virtual-link service-vl vnf-orch
   !
   connection-point eth1
    virtual-link service-vl vnf-mgmt
   !
  !
 !

 vnfd vnf2_esc
  vnf-type          esc
  version           6.2
  high-availability true
  configuration openstack.endpoint publicURL
  configuration secure-login false
  configuration boot-time 1800
  configuration set-vim-instance-name true
  external-connection-point vnf2_esc
   connection-point eth1
   ip-address       10.84.109.91
   floating-ip disabled
  !
  vnfc vnf2_esc
   health-check enabled
   health-check probe-frequency 10
   health-check probe-max-miss 6
   health-check retry-count 3
   health-check recovery-type restart-then-redeploy
   health-check boot-time 300
   vdu vdu-id vdu-esc-vnf2
   vdu flavor ESC_VNF2_FLV
   connection-point eth0
    virtual-link service-vl vnf-orch
   !
   connection-point eth1
    virtual-link service-vl vnf-mgmt
   !
  !
 !


 vnfd vpc1
  vnf-type          ugp
  version           6.2
  high-availability true
  vnfm vnfd vnf1_esc
  configuration internal-network-mtu 1500
  configuration boot-time 1800
  configuration domain-name cisco.com
  configuration set-vim-instance-name true
  configuration dns-server 10.84.96.130
  !
 vld vnf1_di_1
  network-instance nic1_port1_sriov0
 !
 vld vnf1_di_2
  network-instance nic1_port2_sriov2
 !
  external-connection-point cf1
   connection-point eth3
   ip-address       10.84.109.90
   floating-ip disabled
  !
```

```
external-connection-point em1
 connection-point eth1
 ip-address       10.84.109.89
 floating-ip disabled
!
vnfc em1
 health-check enabled
 health-check probe-frequency 10
 health-check probe-max-miss 6
 health-check retry-count 6
 health-check recovery-type restart-then-redeploy
 health-check boot-time 300
 vdu vdu-id vdu-em-vnf1
 vdu flavor EM_VNF1_FLV
 number-of-instances 1
 connection-point eth0
  virtual-link service-vl vnf-orch
 !
 connection-point eth1
  virtual-link service-vl vnf-mgmt
 !
!
vnfc cf1
 health-check enabled
 health-check probe-frequency 10
 health-check probe-max-miss 6
 health-check retry-count 6
 health-check recovery-type restart-then-redeploy
 health-check boot-time 300
 vdu vdu-id vdu-cf-vnf1
 vdu flavor CF_VNF1_FLV
 number-of-instances 1
 aggregate-connection-points DI_INTERFACE
  aggregate-connection-point eth0
  !
  aggregate-connection-point eth1
  !
 !
 connection-point eth0
  virtual-link internal-vl vnf1_di_1
 !
 connection-point eth1
  virtual-link internal-vl vnf1_di_2
 !
 connection-point eth2
  virtual-link service-vl vnf-orch
 !
 connection-point eth3
  virtual-link service-vl vnf-mgmt
 !
!
vnfc sf1
 health-check enabled
 health-check probe-frequency 10
 health-check probe-max-miss 6
 health-check retry-count 6
 health-check recovery-type restart-then-redeploy
 health-check boot-time 300
 vdu vdu-id vdu-sf-vnf1
 vdu flavor SF_VNF1_FLV
 number-of-instances 14
 aggregate-connection-points DI_INTERFACE
  aggregate-connection-point eth0
  !
```

```
   aggregate-connection-point eth1
   !
  !
  connection-point eth0
   virtual-link internal-vl vnf1_di_1
  !
  connection-point eth1
   virtual-link internal-vl vnf1_di_2
  !
  connection-point eth2
   virtual-link service-vl vnf-orch
  !
  connection-point eth3
   virtual-link service-vl vnf1_svc_1
  !
  connection-point eth4
   virtual-link service-vl vnf1_svc_2
  !
 !
!



vnfd vpc2
 vnf-type          ugp
 version           6.2
 high-availability true
 vnfm vnfd vnf2_esc
 configuration internal-network-mtu 1500
 configuration boot-time 1800
 configuration domain-name cisco.com
 configuration set-vim-instance-name true
 configuration dns-server 10.84.96.130
 !
vld vnf2_di_1
 network-instance nic2_port1_sriov3
!
vld vnf2_di_2
 network-instance nic2_port2_sriov1
!
 external-connection-point cf2
  connection-point eth3
  ip-address       10.84.109.93
  floating-ip disabled
 !
 external-connection-point em2
  connection-point eth1
  ip-address       10.84.109.92
  floating-ip disabled
 !
 vnfc em2
  health-check enabled
  health-check probe-frequency 10
  health-check probe-max-miss 6
  health-check retry-count 6
  health-check recovery-type restart-then-redeploy
  health-check boot-time 300
  vdu vdu-id vdu-em-vnf2
  vdu flavor EM_VNF2_FLV
  number-of-instances 1
  connection-point eth0
   virtual-link service-vl vnf-orch
  !
  connection-point eth1
   virtual-link service-vl vnf-mgmt
  !
 !


 vnfc cf2
  health-check enabled
  health-check probe-frequency 10
  health-check probe-max-miss 6
  health-check retry-count 6
  health-check recovery-type restart-then-redeploy
  health-check boot-time 300
  vdu vdu-id vdu-cf-vnf2
  vdu flavor CF_VNF2_FLV
  number-of-instances 1
  aggregate-connection-points DI_INTERFACE
   aggregate-connection-point eth0
   !
   aggregate-connection-point eth1
   !
  !
  connection-point eth0
   virtual-link internal-vl vnf2_di_1
  !
  connection-point eth1
   virtual-link internal-vl vnf2_di_2
  !
  connection-point eth2
   virtual-link service-vl vnf-orch
  !
  connection-point eth3
   virtual-link service-vl vnf-mgmt
  !
 !
 vnfc sf2
  health-check enabled
  health-check probe-frequency 10
  health-check probe-max-miss 6
  health-check retry-count 6
  health-check recovery-type restart-then-redeploy
  health-check boot-time 300
  vdu vdu-id vdu-sf-vnf2
  vdu flavor SF_VNF2_FLV
  number-of-instances 14
  aggregate-connection-points DI_INTERFACE
   aggregate-connection-point eth0
   !
   aggregate-connection-point eth1
   !
  !
  connection-point eth0
   virtual-link internal-vl vnf2_di_1
  !
  connection-point eth1
   virtual-link internal-vl vnf2_di_2
  !
  connection-point eth2
   virtual-link service-vl vnf-orch
  !
  connection-point eth3
   virtual-link service-vl vnf2_svc_1
  !
  connection-point eth4
   virtual-link service-vl vnf2_svc_2
```

```
   !
  !
 !
!

secure-token cimc
 user     admin
 password Csco@123
!
secure-token ssh-baremetal
 user     admin
 password Csco@123
!
secure-token scm-admin
 user     admin
 password Csco@123
!
secure-token scm-oper
 user     admin
 password Csco@123
!
secure-token scm-security
 user     security-admin
 password Csco@123
!
secure-token stack
 user     core
 password Csco@123
!
secure-token vim-admin-creds
 user     admin
 password Px7OAZIDhjYdhUPG
!
secure-token vim-core-creds
 user     core
 password Csco@123
!
secure-token login
 user     ubuntu
 password Csco@123
!
secure-token em_login
 user     ubuntu
 password Csco@123
!
secure-token staros
 user     admin
 password Csco@123
!
secure-token esc_netconf
 user     admin
 password Csco@123
!
secure-token esc_login
 user     admin
 password Csco@123
!
secure-token cf_login
 user     admin
 password Csco@123
!
scm scm
 admin    scm-admin
 oper     scm-oper
 security scm-security
!
vnf-packaged usp_6_2
 checksum           e18b9b7bb205cb69f0af80ef9259c968
 location           /home/ubuntu/usp-6_2_b3-6122.iso
 validate-signature false
 configuration staros1
  external-url /home/ubuntu/system-vnf1.cfg
 !
 configuration staros2
  external-url /home/ubuntu/system-vnf2.cfg
 !
!
vdu vdu-esc-vnf1
 vdu-type           cisco-esc
 login-credential   esc_login
 netconf-credential esc_netconf
 image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   2
 flavor ram     4096
 flavor root-disk 40
 flavor ephemeral-disk 0
 flavor swap-disk 0
!
vdu vdu-esc-vnf2
 vdu-type           cisco-esc
 login-credential   esc_login
 netconf-credential esc_netconf
 image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   2
 flavor ram     4096
 flavor root-disk 40
 flavor ephemeral-disk 0
 flavor swap-disk 0
!

vdu vdu-em-vnf1
 vdu-type         element-manager
 login-credential em_login
 scm              scm
  image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   2
 flavor ram     4096
 flavor root-disk 40
 flavor ephemeral-disk 0
 flavor swap-disk 0
!
vdu vdu-em-vnf2
 vdu-type         element-manager
 login-credential em_login
 scm              scm
  image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   2
 flavor ram     4096
 flavor root-disk 40
 flavor ephemeral-disk 0
 flavor swap-disk 0
!

vdu vdu-cf-vnf1
```

```
 vdu-type         control-function
 login-credential cf_login
  image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   8
 flavor ram     16384
 flavor root-disk 6
 flavor ephemeral-disk 0
 flavor swap-disk 0
 upp cores     30
 upp crypto-cores 0
 upp service-mode vpc
 upp disable-mcdma false
 upp disable-numa false
 upp param DI_INTERFACE
  value BOND:TYPE:i40evf-1,TYPE:i40evf-2
 !
 upp param DI_INTERFACE_VLANID
  value 2111
 !
 upp param MULTI_SEG_MBUF_ENABLE
   value 0
 !
 ned netconf
  ned-id         cisco-staros-nc
  port-number    830
  authentication staros
 !
 configuration staros_config.txt
  apply-at day-zero
  package  staros1
  !source-url file:///opt/cisco/usp/uploads/system-vnf1.cfg
 !
 volume boot cf-boot-vnf1
 volume storage cf-cdr-vnf1
 !
!
vdu vdu-cf-vnf2
 vdu-type         control-function
 login-credential cf_login
  image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   8
 flavor ram     16384
 flavor root-disk 6
 flavor ephemeral-disk 0
 flavor swap-disk 0
 upp cores     30
 upp crypto-cores 0
 upp service-mode vpc
 upp disable-mcdma false
 upp disable-numa false
 upp param DI_INTERFACE
  value BOND:TYPE:i40evf-1,TYPE:i40evf-2
 !
 upp param DI_INTERFACE_VLANID
  value 2112
 !
 upp param MULTI_SEG_MBUF_ENABLE
   value 0
 !
 ned netconf
  ned-id         cisco-staros-nc
  port-number    830
  authentication staros
 !
 configuration staros_config.txt
  apply-at day-zero
  package  staros2
  !source-url file:///opt/cisco/usp/uploads/system-vnf1.cfg
 !
 volume boot cf-boot-vnf2
 volume storage cf-cdr-vnf2
 !
!
vdu vdu-sf-vnf1
 vdu-type    session-function
  image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   20
 flavor ram     98304
 flavor root-disk 6
 flavor ephemeral-disk 0
 flavor swap-disk 0
 flavor cpu-policy dedicated
 flavor cpu-thread-policy isolate
 flavor numa-nodes 0
 !
 flavor numa-nodes 1
 !
 upp cores     40
 upp crypto-cores 0
 upp service-mode vpc
 upp disable-mcdma false
 upp disable-numa false
 upp param CARDTYPE
  value 0x42030100
 !
 upp param DI_INTERFACE
  value BOND:TYPE:i40evf-1,TYPE:i40evf-2
 !
 upp param DI_INTERFACE_VLANID
  value 2111
 !
 upp param IFTASK_CORES
   value 35
 !
 upp param IFTASK_MCDMA_CORES
  value 50
 !
 upp param MULTI_SEG_MBUF_ENABLE
  value 0
 !
!
vdu vdu-sf-vnf2
 vdu-type    session-function
  image vnf-package
 vnf-package primary usp_6_2
 flavor vcpus   20
 flavor ram     98304
 flavor root-disk 6
 flavor ephemeral-disk 0
 flavor swap-disk 0
 flavor cpu-policy dedicated
 flavor cpu-thread-policy isolate
 flavor numa-nodes 0
 !
 flavor numa-nodes 1
```

```
  !
  upp cores     40
  upp crypto-cores 0
  upp service-mode vpc
  upp disable-mcdma false
  upp disable-numa false
  upp param CARDTYPE
   value 0x42030100
  !
  upp param DI_INTERFACE
   value BOND:TYPE:i40evf-1,TYPE:i40evf-2
  !
  upp param DI_INTERFACE_VLANID
   value 2112
  !
  upp param IFTASK_CORES
    value 35
  !
  upp param IFTASK_MCDMA_CORES
   value 50
  !
  upp param MULTI_SEG_MBUF_ENABLE
   value 0
  !
 !
 volume cf-boot-vnf1
  type                LUKS
  size                16
  bus                 ide
  bootable            true
  preserve-on-upgrade false
 !
 volume cf-cdr-vnf1
  type                LUKS
  size                200
  bus                 ide
  bootable            false
  preserve-on-upgrade false
 !
 volume cf-boot-vnf2
  type                LUKS
  size                16
  bus                 ide
  bootable            true
  preserve-on-upgrade false
 !
 volume cf-cdr-vnf2
  type                LUKS
  size                200
  bus                 ide
  bootable            false
  preserve-on-upgrade false
 !


 vnf-rackd TB8-vnf-rack

  host-aggregate TB8-esc-em-vnf1
   host micropod-1
   !
   host micropod-3
   !
  !
  host-aggregate TB8-esc-em-vnf2
   host micropod-2
   !
   host micropod-3
   !
  !


  host-aggregate TB8-vnf1-cf
   host micropod-1
   !
   host micropod-3
   !
  !

  host-aggregate TB8-vnf2-cf
   host micropod-2
   !
   host micropod-3
   !
  !

  host-aggregate TB8-vnf_1_2-sf
   host compute-1
   !
   host compute-2
   !
   host compute-3
   !
   host compute-4
   !
   host compute-5
   !
   host compute-6
   !
   host compute-7
   !
   host compute-8
   !
   host compute-10
   !
   host compute-11
   !
   host compute-12
   !
   host compute-13
   !
   host compute-14
   !
   host compute-15
   !

  !
 !

 vim vim1
  api-version v2
  auth-url    http://10.86.67.72:5000/v2.0
  user        vim-admin-creds
  tenant      admin
 !
 vim default_openstack_vim
  api-version v2
  auth-url    http://10.86.67.72:5000/v2.0
  user        vim-core-creds
  tenant      core
 !


 network-instance ext-net
  ip-prefix 10.84.109.64/27
  type      vlan
  dhcp      true
  gateway   10.84.109.65
 !

 network-instance vnf-orch
  ip-prefix 182.37.180.0/24
  type      vlan
  dhcp      true
  gateway   182.37.180.1
 !

 network-instance nic1_port1_sriov0
  ip-prefix 192.168.10.0/24
  type      sriov-flat
  dhcp      true
  vlan-tag  false
  !physnet   phys_sriov0
 !
 network-instance nic1_port2_sriov2
  ip-prefix 192.168.11.0/24
  type      sriov-flat
  dhcp      true
  vlan-tag  false
  !physnet   phys_sriov2
 !
 network-instance nic2_port1_sriov3
  ip-prefix 192.168.12.0/24
  type      sriov-flat
  dhcp      true
  vlan-tag  false
  !physnet   phys_sriov3
 !
 network-instance nic2_port2_sriov1
  ip-prefix 192.168.13.0/24
  type      sriov-flat
  dhcp      true
  vlan-tag  false
  !physnet   phys_sriov1
 !

 vim-artifactd vim_art_rack
  vnf-rack [ TB8-vnf-rack ]
 !
```

APPENDIX C

Sample setup_data.yaml File

The *setup_data.yaml* file is referenced when configuring the Layer 2 and 3 networking parameters.

Note 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.

```
ADMIN_TENANT_NAME: admin
ADMIN_USER: admin
CIMC-COMMON: {cimc_password: Csco@123, cimc_username: admin}
CISCO_VIC_INTEL_SRIOV: true
COBBLER:
  admin_password_hash:
$6$.NVavFhxOnQfyUzc$qSm9Kqqe3qnG0U6t6/tC9R586SubzbNlKlsFpTnIX3ju45l0kdIaml8c5/gnKBO0dS0LbgNkb8XZPYpBEAilh1

  admin_ssh_keys: [ssh-rsa
[illegible]

      cisco@cisco-server]
  admin_username: root
  cobbler_username: cobbler
  kickstart: {block_storage: ucs-b-and-c-series.ks, compute: ucs-b-and-c-series.ks,
    control: ucs-b-and-c-series.ks}
  pxe_timeout: 90
DISABLE_HYPERTHREADING: False
ENABLE_ESC_PRIV: True
ENABLE_JUMBO_FRAMES: True
INSTALL_MODE: connected
INTEL_NIC_SUPPORT: false
INTEL_SRIOV_VFS: 16
SRIOV_CARD_TYPE: XL710
#INTEL_SRIOV_PHYS_PORTS: 2
MECHANISM_DRIVERS: openvswitch
NETWORKING:
  domain_name: cisco.com
  domain_name_servers: [171.70.168.183]
  networks:
  - gateway: 192.168.50.1
    pool: [192.168.50.100 to 192.168.50.2504
    segments: [management, provision]
    subnet: 192.168.50.0/24
    vlan_id: 105
  - gateway: 10.86.67.1
    segments: [api]
    subnet: 10.86.67.0/24
    vlan_id: 3522
  - gateway: 11.117.0.1
    pool: [11.117.0.5 to 11.117.0.254]
    segments: [tenant]
    subnet: 11.117.0.0/24
    vlan_id: 17
  - gateway: 11.118.0.1
    pool: [11.118.0.5 to 11.118.0.254]
    segments: [storage]
    subnet: 11.118.0.0/24
    vlan_id: None
  - segments: [external]
    vlan_id: 400
  - segments: [provider]
    vlan_id: None
  ntp_servers: [1.ntp.esl.cisco.com, 2.ntp.esl.cisco.com]
NFV_HOSTS: ALL
OPTIONAL_SERVICE_LIST: [heat]
PODTYPE: micro
PROVIDER_VLAN_RANGES: 2111,2112,2001:2050
REGISTRY_EMAIL: mercury-installer@cisco.com
REGISTRY_PASSWORD: B4c0n
REGISTRY_USERNAME: installer
ROLES:

  block_storage: [micropod-1, micropod-2, micropod-3]
  control: [micropod-1, micropod-2, micropod-3]
  compute: [micropod-1, micropod-2, micropod-3, compute-1, compute-2, compute-3, compute-4,
 compute-5, compute-6, compute-7, compute-8, compute-10, compute-11, compute-12, compute-13,
 compute-14, compute-15]
SERVERS:
  micropod-1:
    cimc_info: {cimc_ip: 192.168.60.51}
    rack_info: {rack_id: RackW8-1}
    VM_HUGEPAGE_PERCENTAGE: 10
  micropod-2:
    cimc_info: {cimc_ip: 192.168.60.52}
    rack_info: {rack_id: RackW8-2}
    VM_HUGEPAGE_PERCENTAGE: 10
  micropod-3:
    cimc_info: {cimc_ip: 192.168.60.53}
    rack_info: {rack_id: RackW8-3}
    VM_HUGEPAGE_PERCENTAGE: 10
  compute-1:
    cimc_info: {cimc_ip: 192.168.60.54}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-2:
    cimc_info: {cimc_ip: 192.168.60.55}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-3:
    cimc_info: {cimc_ip: 192.168.60.56}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-4:
    cimc_info: {cimc_ip: 192.168.60.57}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-5:
    cimc_info: {cimc_ip: 192.168.60.58}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-6:
    cimc_info: {cimc_ip: 192.168.60.59}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-7:
    cimc_info: {cimc_ip: 192.168.60.60}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-8:
    cimc_info: {cimc_ip: 192.168.60.61}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-10:
    cimc_info: {cimc_ip: 192.168.60.63}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-11:
    cimc_info: {cimc_ip: 192.168.60.64}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-12:
    cimc_info: {cimc_ip: 192.168.60.65}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-13:
    cimc_info: {cimc_ip: 192.168.60.66}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-14:
    cimc_info: {cimc_ip: 192.168.60.67}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
  compute-15:
    cimc_info: {cimc_ip: 192.168.60.68}
    rack_info: {rack_id: RackW8}
    VM_HUGEPAGE_PERCENTAGE: 94
SERVER_COMMON: {server_username: root}
STORE_BACKEND: ceph
TENANT_NETWORK_TYPES: VLAN
TENANT_VLAN_RANGES: 350:399
VIRTUAL_ROUTER_ID: 49
VMTP_VALIDATION:
  EXT_NET: {DNS_SERVER: 171.70.168.183, NET_GATEWAY: 10.84.109.65, NET_IP_END: 10.84.109.94,

    NET_IP_START: 10.84.109.66, NET_NAME: ext-net, NET_SUBNET: 10.84.109.64/27}
VM_HUGEPAGE_SIZE: 1G
VOLUME_DRIVER: ceph
external_lb_vip_address: 10.86.67.72
external_lb_vip_tls: false
internal_lb_vip_address: 192.168.50.2
```

APPENDIX D

Sample system.cfg File

```
config
        system hostname ugp-saegw
        ssh key-gen wait-time 0
        cli hidden
        tech-support test-commands encrypted password ***
        logging filter runtime facility confdmgr level debug critical-info
        logging filter runtime facility vnfma level debug critical-info
        context local
                administrator $CF_LOGIN_USER password $CF_LOGIN_PASSWORD ftp
                        interface LOCAL1
                                ip address $CF_VIP_ADDR 255.255.255.0
                #exit
                ip route 0.0.0.0 0.0.0.0 $NICID_1_GATEWAY LOCAL1
                ssh generate key
                server sshd
                        subsystem sftp
                #exit
                server confd
                        confd-user admin
                #exit
        #exit
        port ethernet 1/1
            bind interface LOCAL1 local
            no shutdown
        #exit
        snmp community public read-only
end
```