Preparing for Cisco NFVI Installation

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

Installing the Cisco NFVI Hardware

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

Table 1: Cisco NFVI Hardware and Network Connectivity Protocol

UCS Pod Type	Compute and Controller Node	Storage Node	Network Connectivity Protocol
C-Series	UCS C220/240 M4	UCS C240 M4 (SFF) with two internal SSDs	VXLAN/Linux Bridge or OVS/VLAN or VPP/VLAN, or ACI/VLAN
C-Series	Controller: UCS C220/240 Compute: HP DL360 Gen9	UCS C240 M4 (SFF) with two internal SSDs	OVS/VLAN

UCS Pod Type	Compute and Controller Node	Storage Node	Network Connectivity Protocol
C-Series with Cisco VTS	UCS C220/240 M4	UCS C240 M4 (SFF) with two internal SSDs	For tenants: VTF with VXLAN. For providers: VLAN
C-Series Micropod	UCS 240 M4/M5 with 12 HDD and 2 external SSDs. Pod can be expanded to 16 computes. Each compute will have 2x1.2 TB HDD or UCS 220 M4/M5 with 7 HDD and 1 external SSDs. Pod can be expanded to 16 computes. Each compute will have 2x1.2 TB HDD. **Note** Refer to the BOM for SSD based install for M5; M5 BOM is based on Intel X710 for control and data plane and XL710 for SRIOV. For exact BOM details, reach out to Cisco VIM product marketing.	Not applicable as it is integrated with Compute and Controller.	• UCS M4 Support: • OVS/VLAN or VPP/VLAN or ACI/VLAN. • UCS M5 Support: OVS/VLAN or VPP/VLAN.
C-Series Hyperconverged	UCS 240 M4.	UCS C240 M4 (SFF) with 12 HDD and two external SSDs, acts as compute node	OVS/VLAN

UCS Pod Type	Compute and Controller Node	Storage Node	Network Connectivity Protocol
B-Series	UCS B200 M4.	UCS C240 M4 (SFF) with two internal SSDs.	VXLAN/Linux Bridge or OVS/VLAN.
B-Series with UCS Manager Plugin	UCS B200 M4s	UCS C240 M4 (SFF) with two internal SSDs.	OVS/VLAN

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

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

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

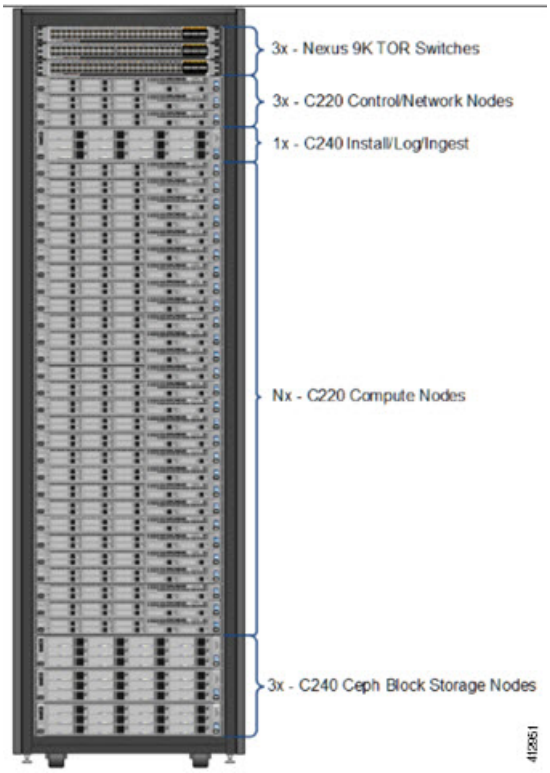
In addition, the Cisco Nexus 9372 or 93180YC, or 9396PX is also available to serve the Cisco NFVI ToR function.

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

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

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

Figure 1: Cisco NFVI C-Series Pod



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

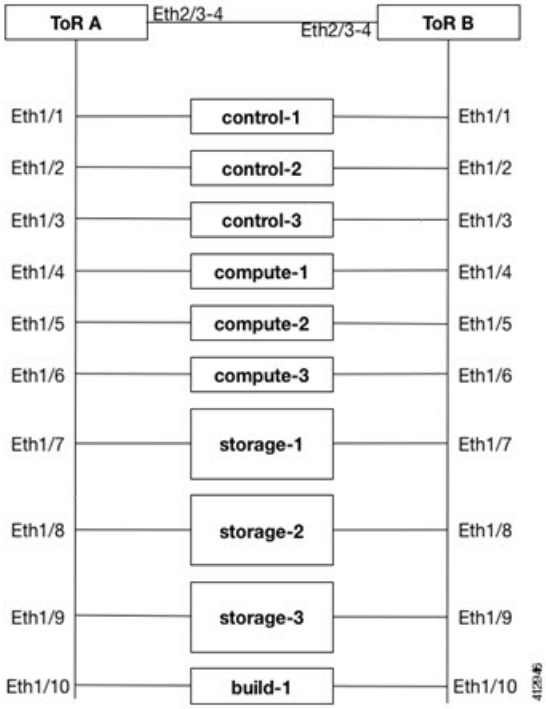
For more information on wiring schematic of various pod configuration, see Appendix.

Configuring ToR Switches for C-Series Pods

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

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

Figure 2: ToR Configuration Sample



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

For the preceding example, the following is the configuration:

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

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

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

Step 2 For each VLAN type, (mgmt_vlan, tenant_vlan_range, storage, api, external, provider), execute the following on each ToR:

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

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

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

1. **For deployment with any mechanism driver on Cisco VIC**

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

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

2. **For deployment with OVS/VPP with VLAN or LinuxBridge on Intel NIC**

   The interface configuration is same as Cisco VIC as shown in the above section. However, number of switch interfaces that are configured is more in the case of Intel NIC as it has dedicated control and data physical ports. For SRIOV switchport, no port channel is configured and the participating VLAN can be in trunk mode.

3. **For deployment with VTS on Intel NIC**

   In this scenario, VTS is used as the mechanism driver. The interface configuration varies based on the server roles. Assume Ethernet1/1 to Ethernet1/3 are controller interfaces, Ethernet1/4 to Ethernet1/6 are storage interfaces, and Ethernet1/7 to Ethernet1/10 are compute interfaces. The sample configuration is as follows:

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

```
interface Ethernet 1/9
channel-group 9
interface Ethernet 1/10
channel-group 10
```

Note When using VTS with Intel NIC, ensure that LACP is turned off for those port channels that are connected to the compute nodes. In the sample configuration, the preceding codes correspond to Ethernet 1/7 to 1/10.

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

```
interface port-channel1-9
shutdown
spanning-tree port type edge trunk
switchport mode trunk
switchport trunk native vlan mgmt_vlan for the control ports when Intel NIC is used
switchport trunk allowed vlan <mgmt_vlan, tenant_vlan_range, storage, api, external, provider>
no lacp suspend-individual
mtu 9216
vpc <1-9>
no shutdown
```

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

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

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

```
gen-leaf-1# show port-channel summary
Flags: D - Down P - Up in port-channel (members)
I - Individual H - Hot-standby (LACP only)
s - Suspended r - Module-removed
S - Switched R - Routed
U - Up (port-channel)
M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port- Type Protocol Member Ports
Channel
--------------------------------------------------------------------------------
1 Po1(SD) Eth LACP Eth1/1(I)
2 Po2(SD) Eth LACP Eth1/2(I)
3 Po3(SD) Eth LACP Eth1/3(I)
4 Po4(SD) Eth LACP Eth1/4(I)
5 Po5(SD) Eth LACP Eth1/5(I)
6 Po6(SD) Eth LACP Eth1/6(I)
7 Po7(SD) Eth LACP Eth1/7(I)
8 Po8(SD) Eth LACP Eth1/8(I)
9 Po9(SD) Eth LACP Eth1/9(I)
10 Po10(SD) Eth LACP Eth1/10(I)
116 Po116(SU) Eth LACP Eth2/3(P) Eth2/4(P)
```

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

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

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

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

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

Port channel:

```
interface port-channel10
   switchport mode trunk
  switchport trunk allowed vlan 80,323,680,860,2680,3122-3250
  mtu 9216
  vpc 10
```

Ethernet:

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

Configuring ToR Switches for UCS B-Series Pods

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

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

Storm-tor-1 (mgmt0 is 172.18.116.185).

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

interface mgmt0
```

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

Storm-tor-2 (mgmt0 is 172.18.116.186).

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

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

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

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

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

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

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

```
feature ospf
feature pim
feature interface-vlan
feature hsrp

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

```
ip pim anycast-rp 192.1.1.1 192.168.100.2

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

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

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

interface Vlan<provider/tenant vlan id>
    no shutdown
    ip address <IP address/mask>
    no ip ospf passive-interface
    ip router ospf 777 area 0.0.0.0
    ip pim sparse-mode
    hsrp 101
    priority 11
    ip <provider/tenant gateway address>
```

Storm-tor-1

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

Storm-tor-2

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

Step 5 If Cisco VIM implementation has extensive multicast traffic, prioritize the multicast traffic by setting up the following service classes on the ToR switches and enabling the media QOS profile as described in the *UCS Manager Common Access Information for B-Series Pods* in Setting Up Cisco VIM Data Configuration . The Nexus 9000 configuration is as follows:

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

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

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

policy-map type queuing Uplink-out_policy
  class type queuing class-silver
    bandwidth percent 60
    priority
  class type queuing class-bronze
```

```
    bandwidth percent 30
  class type queuing class-default
    bandwidth percent 10
class-map type network-qos class-silver
  match qos-group 3
class-map type network-qos class-bronze
  match qos-group 2

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

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

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

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

Note Ensure that you enable jumbo frames in the `setup_data.yaml` file. See the *UCS Manager Common Access Information for B-Series Pods* section in Setting Up Cisco VIM Data Configuration.

Preparing Cisco IMC and Cisco UCS Manager

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

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

Step 1 Verify that each Cisco UCS server uses Cisco IMC firmware version of either 2.0 series (2.0(13i) or greater preferably 2.0(13n)) or 3.0 series (use 3.0.3(f) or later). You can download the latest Cisco IMC ISO image from the Cisco Software Download site. For upgrade procedures, see the Cisco UCS C-Series Rack-Mount Server BIOS Upgrade Guide.

Step 2 For UCS B-Series pods, verify that the Cisco UCS Manager version is one of the following: 2.2(5a), 2.2(5b), 2.2(6c), 2.2(6e), 3.1(c).

Step 3 For UCS C-Series pods, verify the following Cisco IMC information is added: IP address, username, and password.

Step 4 For UCS B-Series pods, verify the following UCS Manager information is added: username, password, IP address, and resource prefix. The resource prefix maximum length is 6. The provisioning network and the UCS Manager IP address must be connected.

Step 5 Verify that no legacy DHCP/Cobbler/PXE servers are connected to your UCS servers. If so, disconnect or disable the interface connected to legacy DHCP, Cobbler, or PXE server. Also, delete the system from the legacy cobbler server.

Step 6 Verify Cisco IMC has NTP enabled and is set to the same NTP server and time zone as the operating system.

Installing the Management Node

This procedure installs RHEL 7.4 with the following modifications:

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

Before you begin

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

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

Step 2 Follow steps in Configuring the Server Boot Order to set the boot order to boot from Local HDD.

Step 3 Follow steps in Cisco UCS Configure BIOS Parameters to set the following advanced BIOS settings:

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

- PCI ROM CLP—Disabled
- PCH SATA Mode—AHCI
- All Onboard LOM Ports—Enabled
- LOM Port 1 OptionROM—Disabled
- LOM Port 2 OptionROM—Disabled
- All PCIe Slots OptionROM—Enabled
- PCIe Slot:1 OptionROM—Enabled
- PCIe Slot:2 OptionROM—Enabled

- PCIe Slot: MLOM OptionROM—Disabled
- PCIe Slot:HBA OptionROM—Enabled
- PCIe Slot:FrontPcie1 OptionROM—Enabled
- PCIe Slot:MLOM Link Speed—GEN3
- PCIe Slot:Riser1 Link Speed—GEN3
- PCIe Slot:Riser2 Link Speed—GEN3
- MLOM OptionROM—Enabled

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

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

Other parameters must be set to default.

Step 4 Click **Save Changes**.

Step 5 Add the management node vNICs to the provisioning VLAN to provide the management node with access to the provisioning network:

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

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

Step 6 Download the Cisco VIM Buildnode ISO image to your computer from the given location.

Step 7 In CIMC, launch the KVM console.

Step 8 Mount the Cisco VIM Buildnode ISO image as a virtual DVD.

Step 9 Reboot the UCS server, then press **F6** to enter the boot menu.

Step 10 Select the KVM-mapped DVD to boot the Cisco VIM Buildnode ISO image provided with the install artifacts.

Step 11 In boot menu, select **Install Cisco VIM Management Node**. This is default selection and it gets automatically selected after the timeout.

Step 12 At the prompts, answer the following questions to install the Management node as unified management node only or not:

- Hostname—Enter the management node hostname (The hostname length must be 32 or less characters).

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

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

- Prompt to enable static IPv6 address configuration—Enter **Yes** to continue input similar IPv6 address configuration for API and MGMT network, or **No** to skip if IPv6 is not needed.
- API IPv6 address—Enter the management node API IPv6 address in CIDR (Classless Inter-Domain Routing) format. For example, 2001:c5c0:1234:5678:1001::5/8.
- Gateway IPv6 address—Enter the API network default gateway IPv6 address.
- MGMT IPv6 address—Enter the management node MGMT IPv6 address in CIDR format. For example, 2001:c5c0:1234:5678:1002::5/80
- DNS server—Enter the DNS server IPv4 address or IPv6 address if static IPv6 address is enabled.
- Option for Teaming Driver for Link Aggregation (answer **yes** when Nexus Switch is the ToR, and answer **no** when Cisco NCS 5500 is ToR): <yes|no> "

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

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

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

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

Step 16 Enter the number corresponding to your time zone.

Step 17 Enter the number for your region.

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

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

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

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

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

```
#date
 Mon Aug 22 05:36:39 PDT 2016

To set date:
#date -s '2016-08-21 22:40:00'
Sun Aug 21 22:40:00 PDT 2016

To check for date:
```

```
#date
Sun Aug 21 22:40:02 PDT 2016
```

Installing Cisco VIM Software Hub

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

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

Before you begin

Prerequisites for Cisco VIM Software Hub Nodes

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

Prerequisites for Cisco VIM Software Hub Server

- **TLS certificate (For production environment)**

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

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

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

- **Self-signed certificate (For internal use)**

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

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

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

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

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

- **DNS server**

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

Installing Cisco VIM Software Hub Node

The steps to install an Cisco VIM Software Hub node are similar to the steps in Installing the Management Node, on page 12. The only difference being, in Step 11 of the task, you need to choose the option to configure the server as an Cisco VIM Software Hub server. In the subsequent prompts, you can enter information such as the hostname, ipv4 or ipv6 addresses for br_public and br_private interfaces, and gateway addresses, similar to the Installing the Management Node, on page 12 task.

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

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

  `/root/cvim_sds-<tag>`

Setting up Cisco VIM Software Hub for Cisco VIM Artifact Distribution

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

Step 1 Copy the EXAMPLE file from the openstack-configs directory and save it as `sds_setup_data.yaml`.

Step 2 If you want to install a release tag on a Cisco VIM Software Hub server, update the fields in the `sds_setup_data.yaml` file as necessary.

```
## Configuration File:
# This file is used as an inventory file to setup CVIM SDS (software delivery server).
##########################################################
# User Defined Configuration File.
# Information in this file is specific to the SDS setup.
#########################################################
SSL_CERT_FILE: <abs_location_for_cert_path of x509 certificate>
SSL_CERT_KEY_FILE: <abs_location_for_cert_priv_key of x509 certificate>
SSL_CERT_CHAIN_FILE: <abs_location_for_cert_chain_file of x509 certificate>
#######################################################
# Registry credentials to access the CVIM registry (Cisco Supplied)
#######################################################
CVIM_REGISTRY_USERNAME: <username>
CVIM_REGISTRY_PASSWORD: <password>
NETWORKING:
## Max. NTP servers = 4, min of 1
ntp_servers: <ntp.server1.fqdn.com, ntp.server2.fqdn.com >
or
ntp_servers: [ipv6_address, 'ipv4_address'] # ","  separated IPv4 or IPv6 address info
http_proxy_server: <proxy.domain.com:8080> # optional, needed if the pod is behind a proxy
https_proxy_server: <proxy.domain.com:8080> # optional, needed if the pod is behind a proxy
SDS_REGISTRY_NAME: <satellite.fqdn.com> #SDS registry name needs to resolve to valid IP
SDS_REGISTRY_USERNAME: <username>
SDS_REGISTRY_PASSWORD: <password>
# (Optional)SDS users who can only pull images from SDS docker registry
SDS_READ_ONLY_USERS:
  - username: <user1>
    password: <password1>
  - username: <user2>
    password: <password2>
```

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

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

Installing Cisco VIM Software Hub in Connected Mode

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

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

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

Step 3 Perform the following steps:

a) Place a valid TLS certificate in the `/root/cvim_sds-<tag>/openstack-configs` directory.

b) Update the fields of the Cisco VIM Software Hub setup data file and save it in the following directory:
`/root/cvim_sds-<tag> openstack-configs`

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

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

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

Installing Cisco VIM Software Hub in Air-Gapped Mode

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

Before you begin

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

Step 1 On the staging server, use yum to install PyYAML and the python-requests package.

Step 2 Access the Cisco VIM software download web site using a web browser.

Step 3 Log in with the credentials provided by your account representative and download the `getartifacts.py` script from the external registry.

```
# download the new getartifacts.py file
curl -o getartifacts.py
https://username:password@cvim-registry.com/mercury-releases/cvim24-rhel7-osp10/releases/<2.4.x>/getartifacts.py

curl -o getartifacts.py-checksum.txt
https://username:password@cvim-registry.com/mercury-releases/cvim24-rhel7-osp10/releases/<2.4.x>/getartifacts.py-checksum.txt

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

Step 4 Run the getartifacts.py script.
The script formats the USB 2.0 drive (or USB 3.0 drive for M5-based management node) and downloads the installation files. You must provide the registry username and password, tag ID, and USB partition on the staging server.

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

```
Script to pull container images en masse.
optional arguments:
-h, --help show this help message and exit
-t TAG, --tag TAG installer version to pull
-u USERNAME, --username USERNAME
Registry username
-p PASSWORD, --password PASSWORD
Registry password
-d DRIVE, --drive DRIVE
Provide usb drive path
--proxy PROXY https_proxy if needed
--retry Try to complete a previous fetch
--artifacts [ARTIFACTS [ARTIFACTS ...]]
Artifact List values(space separated): core insight
All
```

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

Step 5 To identify the USB drive, execute the **lsblk** command before and after inserting the USB drive.

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

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

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

Step 6 Verify the integrity of the downloaded artifacts and the container images.

```
 # create a directory sudo mkdir -p /mnt/Cisco
# /dev/sdc is the USB drive, same as supplied in getartifacts.py python script sudo mount /dev/sdc1
 /mnt/Cisco
cd /mnt/Cisco
# execute the test-usb help to look at the options
./test-usb -h
usage: ./test-usb
[-h] -- Show this program to check integrity of artifacts in this USB drive
[-c] -- Check integrity of only core artifacts in this USB drive
[-i] -- Check integrity of only insight artifacts in this USB drive
[-a] -- Check integrity of all (core and insight) artifacts in this USB drive
[-l] -- Location of artifacts
# execute the verification script
./test-usb
# failures will be explicitly displayed on screen, sample success output below
# sample output of ./test-usb execution with 2.4.5 release
#./test-usb
INFO: Checking the integrity of this USB drive
INFO: Checking artifact buildnode-K9.iso
INFO: Checking artifact registry-2.4.5.tar.gz INFO: Checking required layers:
INFO: 548 layer files passed checksum.
Following output shows the result when using -a option
# ./test-usb -a
INFO: Checking the integrity of this USB drive
INFO: Checking artifact buildnode-K9.iso
INFO: Checking artifact registry-2.4.5.tar.gz
INFO: Checking artifact mariadb-app-K9.tar.gz
INFO: Checking artifact haproxy-K9.tar.gz
INFO: Checking artifact insight-K9.tar.gz
Node
INFO: Checking required layers:
INFO: 548 layer files passed checksum.
```

```
If a failure occurs, an error message is displayed. For example:
# ./test-usb
INFO: Checking the integrity of this USB drive
INFO: Checking artifact buildnode-K9.iso
ERROR: Checksum for artifact buildnode-K9.iso does not match ('SHA512 (buildnode-K9.iso) =
96ec62a0932a0d69daf60acc6b8af2dc4e5eca132cd3781fc17a494592feb52a7f171eda25e59c0d326fbb09194eeda66036cbdc3870dafe74f59cf1f2dce225'

!= 'SHA512 (buildnode-K9.iso) =
a6a9e79fa08254e720a80868555679baeea2dd8f26a0360ad47540eda831617bea0514a117b12ee5f36415b7540afa112a1c904cd69e40d704a8f25d78867acf')

INFO: Checking artifact registry-2.3.1.tar.gz
ERROR: Artifact registry-2.3.1.tar.gz is not present INFO: Checking required layers:
ERROR: Layer file sha256:002aa1f0fbdaea7ea25da1d906e732fe9a9b7458d45f8ef7216d1b4314e05207 has a bad
 checksum
ERROR: Layer file sha256:5be3293a81773938cdb18f7174bf595fe7323fdc018c715914ad41434d995799 has a bad
 checksum
ERROR: Layer file sha256:8009d9e798d9acea2d5a3005be39bcbfe77b9a928e8d6c84374768ed19c97059 has a bad
 checksum
ERROR: Layer file sha256:ea55b2fc29b95d835d16d7eeac42fa82f17e985161ca94a0f61846defff1a9c8 has a bad
 checksum
INFO: 544 layer files passed checksum.
```

Step 7 To resolve failure in downloading artifacts, unmount the USB and run the getartifacts command again with the **--retry** option.

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

Step 8 Mount the USB and then run the **test-usb** command to validate if all the files are downloaded.

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

Execute the verification script.

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

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

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

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

a) Provision your Cisco VIM Software Hub server with the buildnode ISO of that release and then connect the USB to the Cisco VIM Software Hub server.

b) To copy the contents of the USB to the Cisco VIM Software Hub server, navigate to the `/root/cvim_sds-<tag>` directory, and then execute the import artifacts command.

```
# cd ~/cvim_sds-<tag>/tools
# ./import_artifacts.sh -s
```

c) Place a valid TLS certificate in `/root/cvim_sds-<tag>/openstack-configs` directory.

d) Configure the Cisco VIM Software Hub setup data file with all the fields and placed the file in the `/root/cvim_sds-<tag>/openstack-configs` directory.

e) Install the release on the Cisco VIM Software Hub server.

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

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

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

Installing Pod from Cisco VIM Software Hub Server

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

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

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

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

Day 2 Operations on Cisco VIM Software Hub

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

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

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

Setting Up the UCS C-Series Pod

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

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

Step 2 Follow steps in Configuring the Server Boot Order to set the boot order to boot from Local HDD

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

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

- CDN Support for VIC—Disabled
- PCI ROM CLP—Disabled
- PCH SATA Mode—AHCI
- All Onboard LOM Ports—Enabled
- LOM Port 1 OptionROM—Disabled
- LOM Port 2 OptionROM—Disabled
- All PCIe Slots OptionROM—Enabled
- PCIe Slot:1 OptionROM—Enabled
- PCIe Slot:2 OptionROM—Enabled
- PCIe Slot: MLOM OptionROM—Enabled
- PCIe Slot:HBA OptionROM—Enabled
- PCIe Slot:N1 OptionROM—Enabled
- PCIe Slot:N2 OptionROM—Enabled
- PCIe Slot:HBA Link Speed—GEN3

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

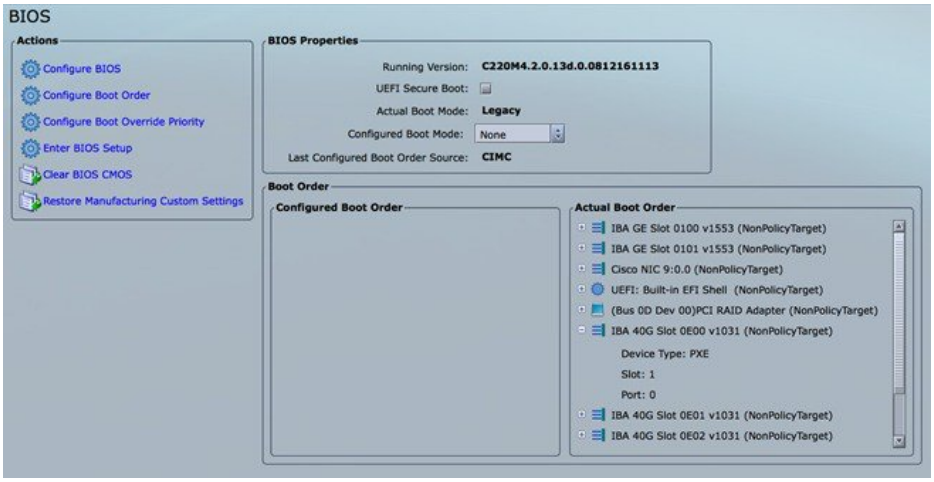
- All Onboard LOM Ports—Enabled
- LOM Port 1 OptionROM—Disabled
- LOM Port 2 OptionROM—Disabled
- PCIe Slot:1 OptionROM—Enabled
- PCIe Slot:2 OptionROM—Enabled
- MLOM OptionROM—Enabled
- MRAID OptionROM—Enabled

Other parameters must be set to their default values.

To setup C-series pod with Intel 710 NIC:

1. Each C-series server must have two 4-port Intel 710 NIC cards.
2. Ports A, B, and C for each Intel 710 NIC card are connected to the respective ToR.
3. 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).

4. Slots are identified by checking the slot-id information under the **Network-Adapter** tab listed under the Inventory link on the **CIMC** pane.
5. All the Intel NIC ports must be indicated in the BIOS summary page under the **Actual Boot Order** pane, as IBA 40G Slot xyza with Device Type is set to PXE.



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

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

While the pxe-boot tool simplifies the job of flashing the intel NIC cards, the restrictions of COSI compliance prevents us from shipping third-party utility. Administrators must download the PREBOOT.exe file from Intel website: https://downloadcenter.intel.com/download/27539/ Ethernet-Intel-Ethernet-Connections-Boot-Utility-Preboot-Images-and-EFI-Drivers

Version: 22.10

Date: 12/7/2017

OS Independent

Language: English

Size: 16.54 MB

MD5: ace485e8a3ef9039212f52b636ce48e3

PREBOOT.EXE

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

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

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

TCP/80 - HTTP

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

Utility Details

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

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

Usage

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

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

You can download PREBOOT.exe file from :

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

Version: 23.1

Date: 2/21/2018

OS Independent

Language: English

Size: 16.54 MB

MD5: dadd5c85777164d8476670774b4459fc

PREBOOT.EXE

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

```
[root@cologne-mgmt tools]# ./intel-bootutil-update.py -h
```

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

Optional arguments:

-h --help show this help message and exit

--hosts HOSTS comma separated list of servers

setup_data.yaml file target for PXE configuration

--exclude-hosts EXCLUDE_HOSTS comma separated list of servers

setup_data.yaml file to exclude for PXE configuration

-v, --verbose enable verbose output

-y, --yes skip prompt

Required arguments:

--setupfile SETUPFILE setup_data.yaml file location

--bootutil-image BOOTUTIL_IMAGE BootUtil image location

--port {0,1,2,3} port #, multiple entries allowed

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

Example to enable all port A:

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

:

Example to enable all port A and B:

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

Example to disable all port C:

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

Flow:

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

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

Utility in action examples:

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

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

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

Version: 23.1

Date: 2/21/2018

OS Independent

Language: English

Size: 16.54 MB

MD5: dadd5c85777164d8476670774b4459fc

PREBOOT.EXE

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

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

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

Setting Up the UCS B-Series Pod

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

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

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

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

Fabric Interconnect A

```
# connect local-mgmt
# erase config
Enter the configuration method. (console/gui) console
Enter the setup mode; setup newly or restore from backup. (setup/restore) ? setup
You have chosen to setup a new Fabric interconnect. Continue? (y/n): y
Enforce strong password? (y/n) [y]: n
Enter the password for "admin":
Confirm the password for "admin":
Is this Fabric interconnect part of a cluster(select 'no' for standalone)? (yes/no) [n]: yes
Enter the switch fabric (A/B) []: A
Enter the system name: skull-fabric
Physical Switch Mgmt0 IPv4 address : 10.30.119.58
Physical Switch Mgmt0 IPv4 netmask : 255.255.255.0
IPv4 address of the default gateway : 10.30.119.1
Cluster IPv4 address : 10.30.119.60
Configure the DNS Server IPv4 address? (yes/no) [n]: y
DNS IPv4 address : 172.29.74.154
Configure the default domain name? (yes/no) [n]: y
Default domain name : ctocllab.cisco.com

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

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

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

Fabric Interconnect B

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

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

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

Step 3 Configure the NTP:

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

Step 4 Following instructions in Cisco UCS Manager GUI Configuration Guide, Release 2.4, "Configuring Server Ports with the Internal Fabric Manager" section, configure the Fabric Interconnect A and Fabric Interconnect B uplinks to the Cisco NFVI top of rack (ToR) switches as **Uplink Ports**, **Server Ports**, and **Port Channels**.

Step 5 Configure the downlinks to the B-Series server chassis as **Server Ports**.

Step 6 Acknowledge all chassis.

Configuring the Out-of-Band Management Switch

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

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

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

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

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

Before you begin

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

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

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

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

Step 2 Location and checksum of the target RPM is:

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