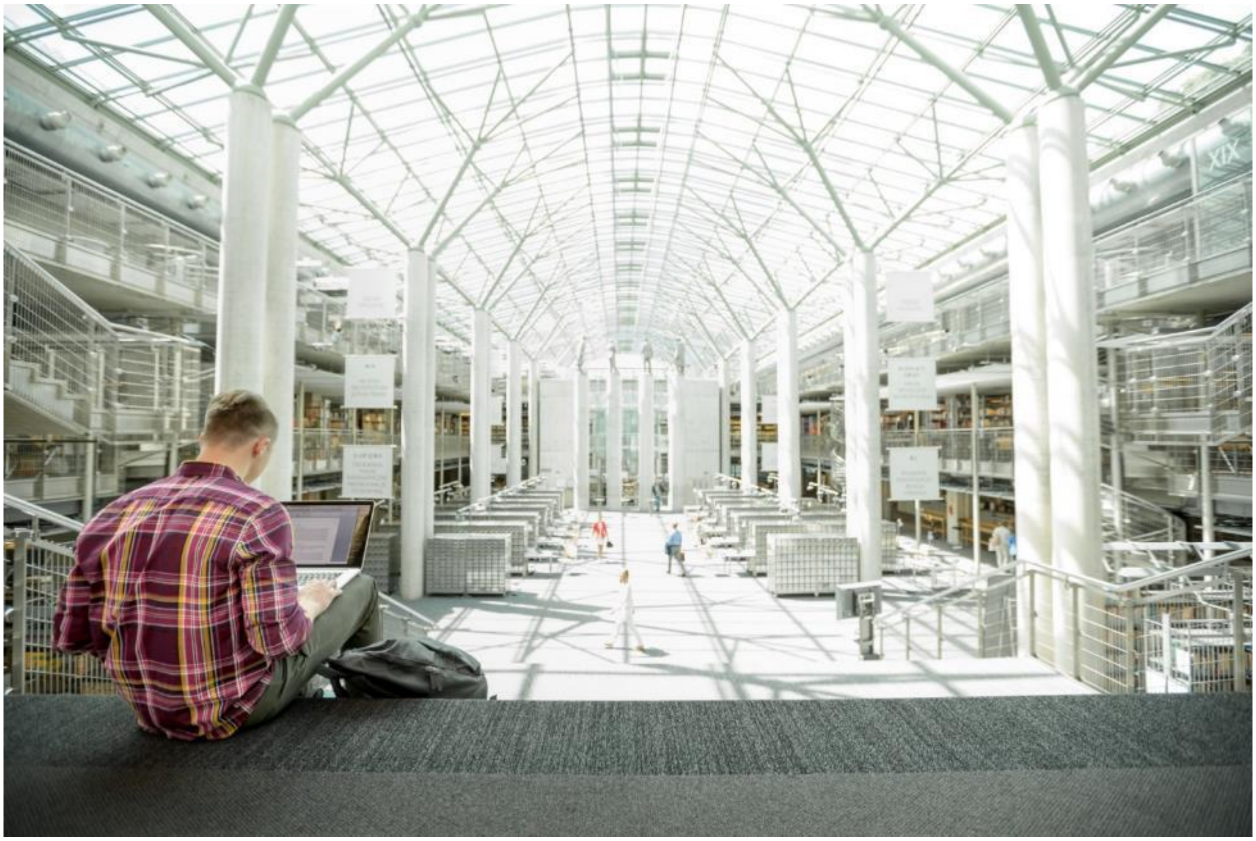


FlexPod Datacenter for AI/ML with Cisco UCS 480 ML for Deep Learning

Deployment Guide for FlexPod Datacenter for AI/ML with Cisco UCS 480 ML for Deep Learning

Published: January 16, 2020

About the Cisco Validated Design Program

The Cisco Validated Design (CVD) program consists of systems and solutions designed, tested, and documented to facilitate faster, more reliable, and more predictable customer deployments. For more information, go to:

http://www.cisco.com/go/designzone.

ALL DESIGNS, SPECIFICATIONS, STATEMENTS, INFORMATION, AND RECOMMENDATIONS (COLLECTIVELY, " DESIGNS") IN THIS MANUAL ARE PRESENTED " AS IS," WITH ALL FAULTS. CISCO AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTY 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 THE DESIGNS, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE DESIGNS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS ARE SOLELY RESPONSIBLE FOR THEIR APPLICATION OF THE DESIGNS. THE DESIGNS DO NOT CONSTITUTE THE TECHNICAL OR OTHER PROFESSIONAL ADVICE OF CISCO, ITS SUPPLIERS OR PARTNERS. USERS SHOULD CONSULT THEIR OWN TECHNICAL ADVISORS BEFORE IMPLEMENTING THE DESIGNS. RESULTS MAY VARY DEPENDING ON FACTORS NOT TESTED BY CISCO.

Table of Contents

Executive Summary

Cisco Validated Designs deliver systems and solutions that are designed, tested, and documented to facilitate and improve customer deployments. These designs incorporate a wide range of technologies and products into a portfolio of solutions that have been developed to address the business needs of the customers and to guide them from design to deployment.

Customers looking to deploy applications using a shared data center infrastructure face several challenges. A recurring infrastructure challenge is to achieve the required levels of IT agility and efficiency that can effectively meet the company's business objectives. Addressing these challenges requires having an optimal solution with the following key characteristics:

- Availability: Help ensure applications and services availability at all times with no single point of failure
- Flexibility: Ability to support new services without requiring underlying infrastructure modifications
- Efficiency: Facilitate efficient operation of the infrastructure through re-usable policies
- Manageability: Ease of deployment and ongoing management to minimize operating costs
- Scalability: Ability to expand and grow with significant investment protection
- Compatibility: Minimize risk by ensuring compatibility of integrated components

Cisco and NetApp have partnered to deliver a series of FlexPod solutions that enable strategic data center platforms with the above characteristics. FlexPod solution delivers an integrated architecture that incorporates compute, storage, and network design best practices thereby minimizing IT risks by validating the integrated architecture to ensure compatibility between various components. The solution also addresses IT pain points by providing documented design guidance, deployment guidance and support that can be used in various stages (planning, designing and implementation) of a deployment.

This document is intended to provide deployment details and guidance around the integration of the Cisco UCS C480 ML M5 platform into the FlexPod datacenter solution to deliver a unified approach for providing AI and ML capabilities within the converged infrastructure. This document also explains NVIDIA GPUs configuration on Cisco UCS C220 and C240 platforms. For a detailed design discussion about the platforms and technologies used in this solution, refer to the FlexPod Datacenter for AI/ML with Cisco UCS 480 ML for Deep Learning Design Guide.

Solution Overview

Introduction

FlexPod Datacenter solution is a pre-designed, integrated and validated architecture for data center that combines Cisco UCS servers, Cisco Nexus family of switches, Cisco MDS fabric switches and NetApp Storage Arrays into a single, flexible architecture. FlexPod solutions are designed for high availability, with no single points of failure, while maintaining cost-effectiveness and flexibility in the design to support a wide variety of workloads. FlexPod design can support different hypervisor options, bare metal servers and can also be sized and optimized based on customer workload requirements.

The FlexPod Datacenter for AI/ML with Cisco UCS C480 ML M5 solution aims to deliver a seamless integration of the Cisco UCS C480 ML M5 platform into the current FlexPod portfolio to enable you to easily utilize the platform's extensive GPU capabilities for their workloads without requiring extra time and resources for a successful deployment. This document also explains the deployment details of NVIDIA GPU equipped Cisco UCS C240 M5 and Cisco UCS C220 M5 servers within the FlexPod datacenter for AI/ML architecture and is a detailed walk through of the solution build out for supporting AI/ML workload.

Audience

The intended audience of this document includes but is not limited to data scientists, IT architects, sales engineers, field consultants, professional services, IT managers, partner engineering, and customers who want to take advantage of an infrastructure built to deliver IT efficiency and enable IT innovation.

What's New in this Release?

The following design elements distinguish this version of FlexPod from previous models:

- Optimized integration of Cisco UCS C480 ML M5 platform into the FlexPod design
- Integration of NetApp A800 NVMe based all flash storage system to support AI/ML dataset.
- Showcase AI/ML workload acceleration using NVIDIA V100 32G GPUs on Cisco UCS C480 ML M5 and Cisco UCS C240 M5 platforms.
- Showcase AI/ML workload acceleration using NVIDIA T4 GPUs on Cisco UCS C220 M5 platform.
- Showcase NVIDIA Virtual Compute Server (vComputeServer) software and Virtual GPU (vGPU) capabilities on various UCS platforms.
- Support for Intel 2nd Gen Intel Xeon Scalable Processors (Cascade Lake) processors*.
- NetApp FlexGroup volumes and NetApp ONTAP 9.6 release.

*** The UCS software version 4.0(4e) (explained in this validation) and RHEL 7.6 support Cascade Lake CPUs on Cisco UCS C220 M5 and C240 M5 servers. Support for Cisco UCS C480ML M5 will be available in the upcoming Cisco UCS release.**

Solution Design

Architecture

FlexPod Datacenter for AI/ML with Cisco UCS 480 ML M5 comprises of following core components:

- High-Speed Cisco NxOS based Nexus 9336C-FX2 switching design supporting up to 100GbE connectivity
- Cisco UCS Manager (UCSM) on Cisco 4th generation 6454 Fabric Interconnects to support 10GbE, 25GbE and 100 GbE connectivity from various components
- Cisco UCS C480 ML M5 server with 8 NVIDIA V100-32GB GPUs for AI/ML applications
- NetApp AFF A800 NVMe storage for both traditional and AI/ML workloads with 100GbE connectivity
- (Optional) Cisco UCS C220 M5 and Cisco UCS C240 M5 server(s) with NVIDIA V100 or NVIDIA T4 GPUs can also be utilized for AI/ML workload processing depending on customer requirements.

In this validation, Cisco UCS C240 M5 server was equipped with two NVIDIA V100-32GB PCIE GPUs and a Cisco UCS C220 M5 was equipped with two NVIDIA T4 GPUs.

The solution focuses on the integration of the Cisco UCS C480 ML platform into the FlexPod datacenter solution to deliver support for GPU intensive artificial intelligence and machine learning capabilities in the converged infrastructure. This FlexPod datacenter solution for machine learning closely aligns with latest NX-OS based FlexPod CVD located here: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_datacenter_vmware_netappaffa_u2.html .

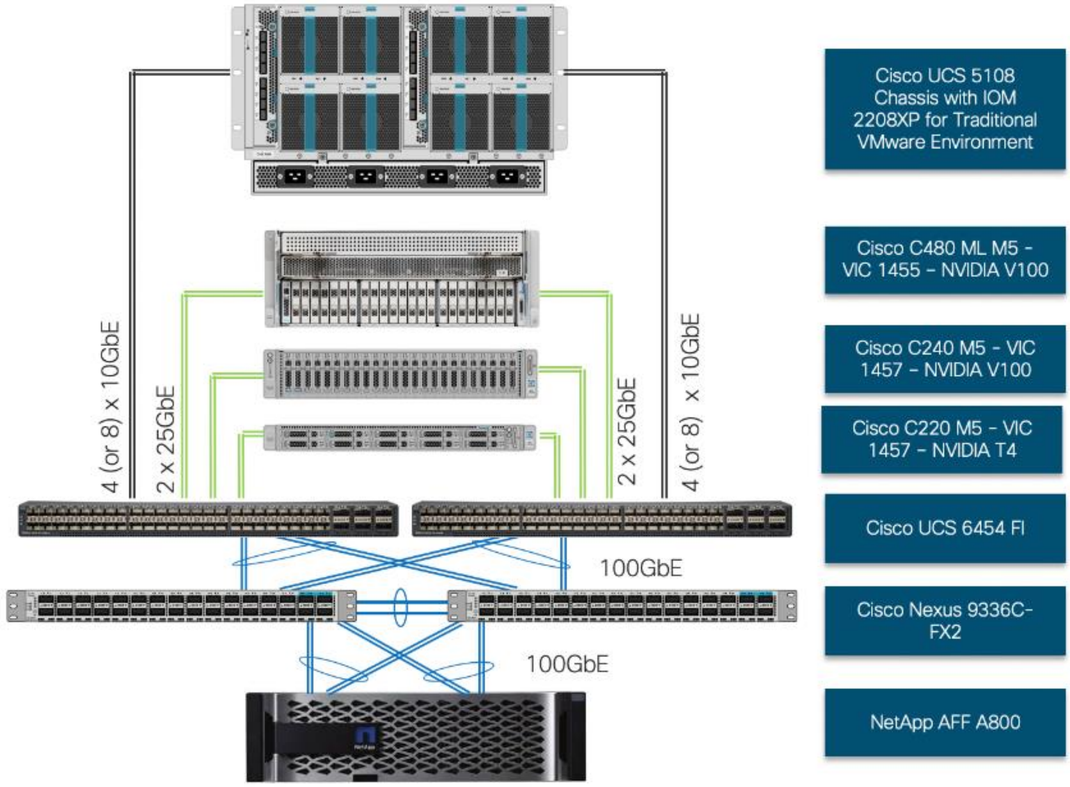
The following design requirements were considered for the GPU equipped Cisco UCS C-series server integration into the FlexPod DC:

1. Ability of the Cisco UCS Manager to manage these AI/ML compute platforms like any other Cisco UCS B-Series or C-Series compute node in the design.
2. Support for stateless compute design where the operating system disk is hosted on shared storage (NetApp) and accessed using iSCSI. This operating system disk will coexist on the NetApp A800 controller being used for traditional FlexPod Datacenter environment.
3. Cisco UCS Service Profile based deployment for both Red Hat Enterprise Linux and VMware ESXi deployments.
4. High availability and redundancy for platform connectivity such that the system can handle one or more link, FI or a storage controller failure.
5. Switching architecture to enable efficient NFS access from AI/ML platforms, both bare-metal and virtual machines, to the training and inference datasets hosted on NetApp A800 controller.
6. Automatic load balancing and parallelized data access and scaling using NetApp FlexGroup volumes.
7. Ability to deploy and live migrate a vGPU equipped VM across matching-GPU platforms.

Physical Topology

The physical topology for the integration of Cisco UCS C480 ML M5 platform(s) into a typical FlexPod datacenter design is shown in Figure 1.

Figure 1 FlexPod for Deep Learning – Physical Topology


Cisco UCS 5108 Chassis with IOM 2208XP for Traditional VMware Environment
Cisco C480 ML M5 - VIC 1455 – NVIDIA V100
Cisco C240 M5 - VIC 1457 – NVIDIA V100
Cisco C220 M5 - VIC 1457 – NVIDIA T4
Cisco UCS 6454 FI
Cisco Nexus 9336C-FX2
NetApp AFF A800
4 (or 8) x 10GbE
2 x 25GbE
2 x 25GbE
4 (or 8) x 10GbE
100GbE
100GbE

To validate the Cisco UCS C480 ML M5 integration into FlexPod datacenter design, an environment with the following components was setup to support both virtual machines and bare metal AI/ML servers:

- Cisco UCS 6454 Fabric Interconnects (FI) to support Cisco UCS 5108 chassis and Cisco UCS C-Series M5 servers.
- Cisco UCS 5108 chassis connects to FIs using 2208XP IOMs. The 2208XP IOMs supports up to 8 10GbE connections to each FI. Customer can utilize 4 or all 8 10GbE connections to each FI based on their bandwidth requirements.
- Cisco UCS C480 ML M5 is connects to each FI using Cisco VIC 1455. Cisco VIC 1455 has 4 25GbE ports. The server is connected to each FI using 2 x 25GbE connections, configured as port-channels.
- Cisco UCS C220 M5 and C240 M5 are connected to each FI using Cisco VIC 1457. Cisco VIC 1457 has 4 25GbE ports. The servers are connected to each FI using 2 x 25GbE connections, configured as port-channels.
- Cisco Nexus 9336C running in NX-OS mode provides the switching fabric.
- Cisco UCS 6454 FI's 100GbE uplink ports are connected to Nexus 9336C as port-channels.
- NetApp AFF A800 controllers are connected to Nexus 9336C switch using 100GbE port-channels.

Base Infrastructure

The reference architecture described in this document leverages the components explained in the Cisco UCS 4th Generation, and NetApp AFF A-Series Deployment Guide. The FlexPod Datacenter for AI/ML extends the virtual infrastructure architecture by adding the Cisco UCS C480 ML M5 platform to the infrastructure and deploying bare metal Red Hat Enterprise Linux (RHEL) to support AI/ML workloads. The design also enables you to deploy NVIDIA vComputeServer and vGPU functionality within their VMware environment using various GPU equipped C-Series platforms. This deployment guide explains the relevant switching, server and storage configuration.

This deployment guide explains hardware integration aspects of both virtual infrastructure and AI/ML platform as well as configuration of Cisco UCS GPU enabled platforms. However, the base hardware configuration as well as base virtual machine infrastructure configuration and setup is not explained in this document. Customers are encouraged to refer to https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_datacenter_vmware_netappaffa_u2.html the for step-by-step configuration procedures.

Hardware and Software Revisions

Table 1 lists the software versions for hardware and software components used in this solution. Each version used has been certified within interoperability matrixes supported by Cisco, NetApp, and VMware. For more information about supported versions, consult the following sources:

- Cisco UCS Hardware and Software Interoperability Tool
- NetApp Interoperability Matrix
- VMware Compatibility Guide

Table 1 Hardware and Software Revisions

Component		Software
Network	Nexus 9336C-FX2	7.0(3)I7(6)
Compute	Cisco UCS Fabric Interconnect 6454	4.0(4e)*
	Cisco UCS C-series Servers	4.0(4e)*
	Red Hat Enterprise Linux (RHEL)	7.6
	RHEL ENIC driver	3.2.210.18-738.12
	NVIDIA Driver for RHEL	418.40.04
	NVIDIA driver for ESXi	430.46
	NVIDIA CUDA Toolkit	10.1 Update 2
	VMware vSphere	6.7U3
	VMware ESXi ENIC Ethernet Driver	1.0.29.0
Storage	NetApp A800	9.6
	NetApp NFS Plugin for VMware VAAI	1.1.2-3

	NetApp Virtual Storage Console	9.6

*** In this deployment guide, the Cisco UCS release 4.0(4e) was only verified for C-Series hosts participating in AIML workloads.**

Required VLANs

Table 2 list various VLANs configured for setting up the FlexPod environment including their specific usage.

Table 2 VLAN Usage

VLAN ID	Name	Usage
2	Native-VLAN	Use VLAN 2 as Native VLAN instead of default VLAN (1)
20	IB-MGMT-VLAN	Management VLAN to access and manage the servers
220 (optional)	Data-Traffic	VLAN to carry data traffic for both VM and bare-metal Servers
3111 (Fabric A only)	iSCSI-A	iSCSI-A path for booting both B Series and C Series servers
3121 (Fabric B only)	iSCSI-B	iSCSI-B path for booting both B Series and C Series servers
3151	ESXi-NFS-VLAN	NFS VLAN for mounting ESXi datastores in ESXi environment
3152	AI-ML-NFS	NFS VLAN to access AI/ML NFS volume hosting ImageNet data

Some of the key highlights of VLAN usage are as follows:

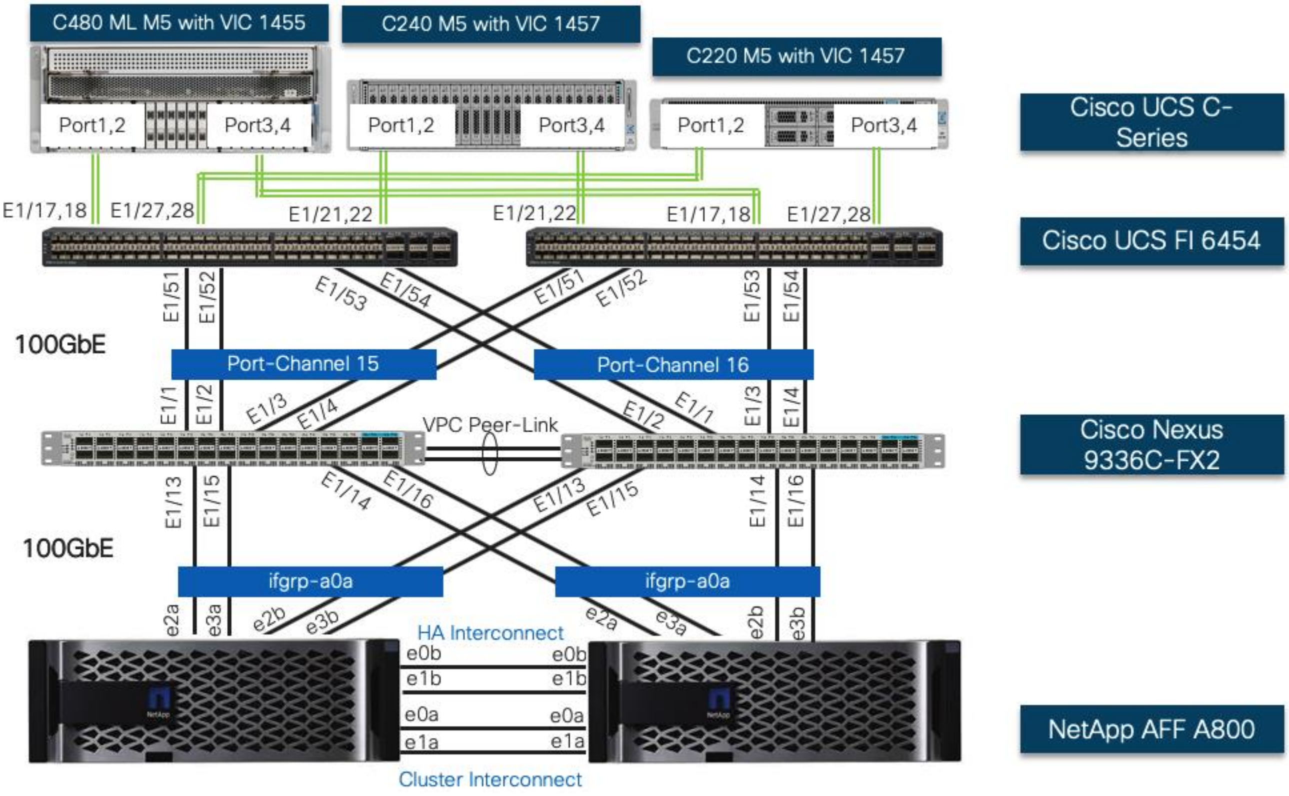
- Both virtual machines and bare-metal servers are managed using same management in-band VLAN: IB-MGMT-VLAN (20).
- A single VLAN: Data-Traffic (220) is used for virtual machine and bare-metal data communication. You can adjust this data VLAN according to their specific setup. If this communication is not needed or can utilize the management VLAN: IB-MGMT-VLAN (20), this VLAN and associated vNIC configuration (defined later in the document) can be skipped.
- Utilizing dedicated NFS VLANs for AI/ML hosts provides path selection flexibility and the ability to configure specific QoS policies. You are encouraged to use separate, dedicated VLANs for ESXi and AI/ML host NFS traffic.
- A pair of iSCSI VLANs is utilized to access boot LUNs for Cisco UCS C480 AI/ML servers. You can also utilize the existing iSCSI VLANs depending on their storage system setup.

Physical Infrastructure

FlexPod Cabling

The information in this section is provided as a reference for cabling the physical equipment in a FlexPod environment. Customers can adjust the ports according to their individual setup. This document assumes that out-of-band management ports are plugged into an existing management infrastructure at the deployment sites. The interfaces shown in Figure 2will be used in various configuration steps. Additional 1Gb management connections will be needed for an out-of-band network switch that sits apart from the FlexPod infrastructure. Each Cisco UCS fabric interconnect, Cisco Nexus switch and NetApp AFF controller is connected to the out-of-band network switch. Layer 3 network connectivity is required between the Out-of-Band (OOB) and In-Band (IB) Management Subnets.

Figure 2 Physical Cabling for Various Cisco UCS C-Series Servers



Network Switch Configuration

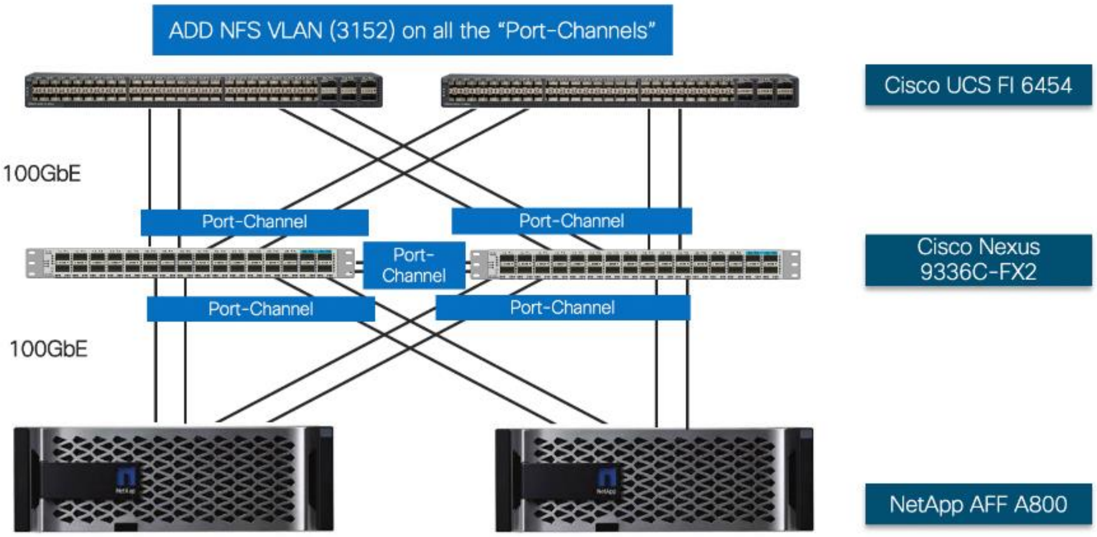
This section provides the configuration required on the Cisco Nexus 9000 switches for FlexPod AI/ML setup. The configuration and the port information can be adjusted depending on customer setup and bandwidth requirements. The following procedures assume the use of Cisco Nexus 9000 7.0(3)I7(6), the Cisco suggested Nexus switch release at the time of this validation. This configuration allows deployment of Cisco AI/ML platforms in bare-metal server configuration.

With Cisco Nexus 9000 release 7.0(3)I7(6), auto-negotiation (40G/100G) is not supported on certain ports of the Cisco Nexus 9336C-FX2 switch. To avoid any misconfiguration and confusion, the port speed and duplex are manually set for all the 100GbE connections.

vGPU-only Deployment in Existing VMware Environment

If you requires vGPU functionality in existing VMware infrastructure and does not need to deploy Bare-Metal RHEL servers, adding the NFS VLAN (3152) for accessing AI/ML data (Imagenet) to the VPCs connected to NetApp and UCS and the VPC peer-link is all that is needed as shown in Figure 3. Enabling the NFS VLAN on appropriate VPC links at the switches allows you to access NFS LIF using a VM port-group on the ESXi hosts.

Figure 3 NFS VLAN on Nexus Switch for vGPU Support Only



The configuration sections below help configure the Nexus switches for deploying bare-metal servers and include the addition of the NFS VLAN (3152) on the appropriate interfaces.

Enable Features

Cisco Nexus A and Cisco Nexus B

To enable the required features on the Cisco Nexus switches, follow these steps:

1. Log in as admin.
2. Run the following commands:

```
config t
feature udld
feature interface-vlan
feature lacp
feature vpc
feature lldp
```

Global Configurations

Cisco Nexus A and Cisco Nexus B

To set global configurations, complete the following step on both switches:

1. Run the following commands to set (or verify) various global configuration parameters:

```
config t
spanning-tree port type network default
spanning-tree port type edge bpduguard default
spanning-tree port type edge bpdufilter default
!
port-channel load-balance src-dst l4port
!
ntp server <NTP Server IP> use-vrf management
!
vrf context management
ip route 0.0.0.0/0 <ib-mgmt-vlan Gateway IP>
!
copy run start
```

Make sure as part of the basic Nexus configuration, the management interface Mgmt0 is setup with an ib-mgmt-vlan IP address.

Create VLANs

Cisco Nexus A and Cisco Nexus B

To create the necessary virtual local area networks (VLANs), complete the following step on both switches:

1. From the global configuration mode, run the following commands to create the VLANs. The VLAN IDs can be adjusted based on customer setup.

```
vlan 2
name Native-VLAN
vlan 20
name IB-MGMT-VLAN
vlan 220
name Data-Traffic
vlan 3111
name iSCSI-A
vlan 3121
name iSCSI-B
vlan 3152
name AI-ML-NFS
```

Configure Virtual Port-Channel Parameters

Cisco Nexus A

```
vpc domain 10
  peer-switch
  role priority 10
  peer-keepalive destination <Nexus-B-Mgmt-IP> source <Nexus-A-Mgmt-IP>
  delay restore 150
  peer-gateway
  no layer3 peer-router syslog
  auto-recovery
  ip arp synchronize
!
interface port-channel10
  description vPC peer-link
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  spanning-tree port type network
  speed 100000
  duplex full
  no negotiate auto
  vpc peer-link
!
interface Ethernet1/35
  description Nexus-B:1/35
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  speed 100000
  duplex full
  no negotiate auto
  channel-group 10 mode active
  no shutdown
!
interface Ethernet1/36
  description Nexus-B:1/36
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  speed 100000
  duplex full
  no negotiate auto
  channel-group 10 mode active
  no shutdown
!
```

Cisco Nexus B

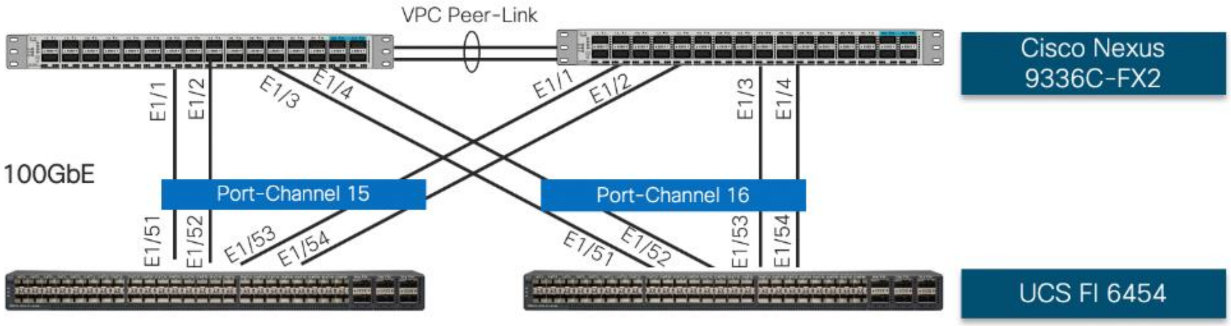
```
vpc domain 10
  peer-switch
  role priority 20
  peer-keepalive destination <Nexus-A-Mgmt0-IP> source <Nexus-B-Mgmt0-IP>
  delay restore 150
  peer-gateway
  no layer3 peer-router syslog
  auto-recovery
  ip arp synchronize
!
interface port-channel10
  description vPC peer-link
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  spanning-tree port type network
  speed 100000
  duplex full
  no negotiate auto
  vpc peer-link
!
interface Ethernet1/35
  description Nexus-A:1/35
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  speed 100000
  duplex full
  no negotiate auto
  channel-group 10 mode active
  no shutdown
!
interface Ethernet1/36
  description Nexus-A:1/36
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  speed 100000
  duplex full
  no negotiate auto
  channel-group 10 mode active
  no shutdown
!
```

Configure Virtual Port-Channels

Cisco UCS 6454 Fabric Interconnect to Nexus 9336C-FX2 Connectivity

Cisco UCS 6454 Fabric Interconnect (FI) is connected to the Nexus switch using 100GbE uplink ports as shown in Figure 4. Each FI connects to each Nexus 9336C using 2 100GbE ports for a combined bandwidth of 400GbE from each FI to the switching fabric. The Nexus 9336C switches are configured for two separate vPCs, one for each FI.

Figure 4 Cisco UCS 6454 FI to Nexus 9336C Connectivity



Nexus A Configuration

```
! FI-A
!
interface port-channel11
  description UCS FI-A
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  spanning-tree port type edge trunk
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  vpc 11
!
interface Ethernet1/1
  description UCS FI-A E1/51
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  udld enable
  channel-group 11 mode active
  no shutdown
!
interface Ethernet1/2
  description UCS FI-A E1/52
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  udld enable
  channel-group 11 mode active
  no shutdown


!
! FI-B
!
interface port-channel12
  description UCS FI-B
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  spanning-tree port type edge trunk
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  vpc 12
!
interface Ethernet1/3
  description UCS FI-B E1/1
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  udld enable
  channel-group 12 mode active
  no shutdown

interface Ethernet1/4
  description UCS FI-B E1/2
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  udld enable
  channel-group 12 mode active
  no shutdown
!
```

Nexus B Configuration

```
! FI-A
!
interface port-channel11
  description UCS FI-A
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  spanning-tree port type edge trunk
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  vpc 11
!
interface Ethernet1/1
  description UCS FI-A E1/53
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  udld enable
  channel-group 11 mode active
  no shutdown
!
interface Ethernet1/2
  description UCS FI-A E1/54
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  udld enable
  channel-group 11 mode active
  no shutdown


!
! FI-B
!
interface port-channel12
  description UCS FI-B
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  spanning-tree port type edge trunk
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  vpc 12
!
interface Ethernet1/3
  description UCS FI-B E1/53
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  udld enable
  channel-group 12 mode active
  no shutdown
```

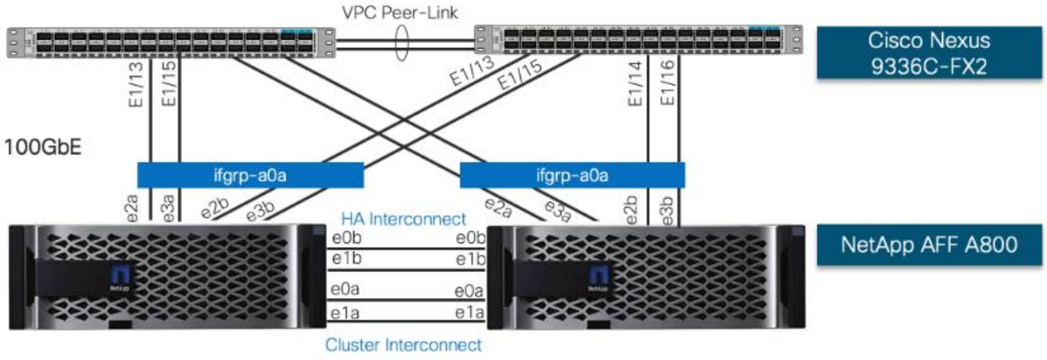
```
interface Ethernet1/4
  description UCS FI-B E1/54
  switchport mode trunk
  switchport trunk native vlan 2
  switchport trunk allowed vlan 20,220,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  udld enable
  channel-group 12 mode active
  no shutdown
!
```

NetApp A800 to Nexus 9336C-FX2 Connectivity

NetApp A800 controllers are connected to Cisco Nexus 9336C-FX2 switches using 100GbE connections. Figure 5depicts the physical connectivity design of the NetApp AFF A800 system running ONTAP 9.5.

In Figure 5, **the two storage controllers in the high availability pair are drawn separately for clarity. Physically, the two controllers exist within a single chassis.**

Figure 5 NetApp A800 Storage Design



Nexus-A Configuration

```
!
NetApp Controller-1
!
interface port-channel113
  description A800-1
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  spanning-tree port type edge trunk
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  vpc 113
!
interface Ethernet1/13
  description A800-1:e2a
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  channel-group 113 mode active
  no shutdown
```

```
!
interface Ethernet1/15
  description A800-1:e3a
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  channel-group 113 mode active
  no shutdown
!
NetApp Controller-2
!
interface port-channel114
  description A800-2
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  spanning-tree port type edge trunk
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  vpc 114
!
interface Ethernet1/14
  description A800-2:e2a
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  channel-group 114 mode active
  no shutdown
!
interface Ethernet1/16
  description A800-2:e3a
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  channel-group 114 mode active
  no shutdown
!
```

Nexus-B Configuration

```
!
NetApp Controller-1
!
interface port-channel113
  description A800-1
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  spanning-tree port type edge trunk
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  vpc 113
!
interface Ethernet1/13
  description A800-1:e2b
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  channel-group 113 mode active
  no shutdown
!
interface Ethernet1/15
  description A800-1:e3b
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  channel-group 113 mode active
  no shutdown
!
NetApp Controller-2
!
interface port-channel114
  description A800-2
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  spanning-tree port type edge trunk
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  vpc 114
!
interface Ethernet1/14
  description A800-2:e2b
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  channel-group 114 mode active
  no shutdown
!
interface Ethernet1/16
  description A800-2:e3b
  switchport mode trunk
  switchport trunk allowed vlan 20,3111,3121,3152
  mtu 9216
  speed 100000
  duplex full
  no negotiate auto
  channel-group 114 mode active
  no shutdown
!
```

Storage Configuration

This solution uses the FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2 solution as a base infrastructure. This deployment guide only explains the storage configuration steps that need to be implemented in addition to the configuration explained in the base CVD.

The referenced solution is built using the All Flash FAS AFF A320 however, in this deployment the AFF A800 is used.

The initialization phase and initial configuration steps remain the same, therefore follow these steps to set up the NetApp storage for the first time:

1. Configure ONTAP nodes.
2. Set up node (using NetApp A800-specific recommended interfaces).
3. Log in to the cluster.
4. Verify storage failover.
5. Set auto-revert on cluster management.
6. Zero all spare disks.
7. Set up Service Processor network interface.
8. Create aggregates.

For customers interested only in vGPU functionality for the existing VMware environment, configuring iSCSI LIFs and boot LUNs for the bare-metal OS installation is not required.

Remove Ports from Default Broadcast Domain

Except for e0M, all other ports should be removed from the default Broadcast domain.

Disable flow control on 100GbE ports.

Two X1146A 100GbE cards in Slots 2 and 3 were used for the AI/ML networking.

NetApp recommends disabling flow control on all the 10/40/100GbE and UTA2 ports that are connected to external devices. To disable flow control, follow these steps:

1. Run the following commands to configure node 01:

```
network port modify -node <st-node01> -port e2a,e2b,e3a,e3b -flowcontrol-admin none
Warning: Changing the network port settings will cause a several second interruption in carrier.
Do you want to continue? {y|n}: y
```

2. Run the following commands to configure node 02:

```
network port modify -node <st-node02> -port e2a,e2b,e3a,e3b -flowcontrol-admin none
Warning: Changing the network port settings will cause a several second interruption in carrier.
```

```
Do you want to continue? {y|n}: y

network port show -fields flowcontrol-admin
```

Disable Auto-Negotiate on 100GbE Ports

To disable the auto-negotiate on the 100GbE ports, follow these steps:

1. Run the following command to configure the ports on node 01:

```
network port modify -node <st-node01> -port e2a,e2b,e3a,e3b -autonegotiate-admin false -speed-admin
100000 -duplex-admin full -flowcontrol-admin none
```

2. Run the following command to configure the ports on node 02:

```
network port modify -node <st-node02> -port e2a,e2b,e3a,e3b -autonegotiate-admin false -speed-admin
100000 -duplex-admin full -flowcontrol-admin none

network port show -node * -port e2a,e2b,e3a,e3b -fields speed-admin,duplex-admin,flowcontrol-admin
  (network port show)
node        port duplex-admin speed-admin flowcontrol-admin
----------- ---- ------------ ----------- -----------------
aa14-a800-1 e2a  full         100000      none
aa14-a800-1 e2b  full         100000      none
aa14-a800-1 e3a  full         100000      none
aa14-a800-1 e3b  full         100000      none
aa14-a800-2 e2a  full         100000      none
aa14-a800-2 e2b  full         100000      none
aa14-a800-2 e3a  full         100000      none
aa14-a800-2 e3b  full         100000      none
8 entries were displayed.
```

Enable Cisco Discovery Protocol

Follow the steps found in FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2.

Enable Link-layer Discovery Protocol on all Ethernet Ports

Follow the steps found in FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2.

Create Management Broadcast Domain

If the management interfaces are required to be on a separate VLAN, create a broadcast domain for those interfaces.

```
network port broadcast-domain create -broadcast-domain AI-ML-IB-MGMT -mtu 1500
network port broadcast-domain show
```

Create NFS Broadcast Domain

Two NFS broadcast domains were created, one for the NFS VLAN interfaces to be used with ESXi hosts and the other one for the VLAN interfaces to be used for the AI/ML workload.

```
network port broadcast-domain create -broadcast-domain AI-ML-NFS -mtu 9000
network port broadcast-domain create -broadcast-domain AI-ML-ESXi-NFS -mtu 9000
network port broadcast-domain show
```

Create Interface Groups

To create the LACP interface groups for the 100GbE data interfaces, run the following commands:

```
network port ifgrp create -node <st-node01> -ifgrp a0a -distr-func port -mode multimode_lacp
network port ifgrp add-port -node <st-node01> -ifgrp a0a -port e2a
network port ifgrp add-port -node <st-node01> -ifgrp a0a -port e2b
network port ifgrp add-port -node <st-node01> -ifgrp a0a -port e3a
network port ifgrp add-port -node <st-node01> -ifgrp a0a -port e3b

network port ifgrp create -node <st-node02> -ifgrp a0a -distr-func port -mode multimode_lacp
network port ifgrp add-port -node <st-node02> -ifgrp a0a -port e2a
network port ifgrp add-port -node <st-node02> -ifgrp a0a -port e2b
network port ifgrp add-port -node <st-node02> -ifgrp a0a -port e3a
network port ifgrp add-port -node <st-node02> -ifgrp a0a -port e3b

network port ifgrp show
```

Change MTU on Interface Groups

Change the MTU size on the Interface Group ports before creating the VLAN ports.

```
network port modify –node <st-node01> -port a0a –mtu 9000
network port modify –node <st-node02> -port a0a –mtu 9000
```

Create VLANs

To create VLANs, follow these steps:

1. Create the NFS VLAN ports and add them to the NFS broadcast domains. The NFS VLAN is used to mount the AI/ML dataset ONTAP FlexGroup volume to the Cisco C-Series servers running RHEL.

```
network port vlan create –node <st-node01> -vlan-name a0a-<ai-ml-nfs-vlan-id>
network port vlan create –node <st-node02> -vlan-name a0a-<ai-ml-nfs-vlan-id>

network port broadcast-domain add-ports -broadcast-domain AI-ML-NFS -ports <st-node01>: a0a-<ai-ml-nfs-
vlan-id>,<st-node02>: a0a-<ai-ml-nfs-vlan-id>

network port vlan show
```

2. Create the NFS VLAN ports for ESXi and add them to the corresponding NFS broadcast domains. This NFS VLAN is used to mount a datastore to the vSphere environment.

If the ESXi environment has already been setup using the steps in FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2, this step may not be required.

```
network port vlan create –node <st-node01> -vlan-name a0a-<ai-ml-esxi-nfs-vlan-id>
network port vlan create –node <st-node02> -vlan-name a0a-<ai-ml-esxi-nfs-vlan-id>

network port broadcast-domain add-ports -broadcast-domain AI-ML-ESXi-NFS -ports <st-node01>: a0a-<ai-ml-
esxi-nfs-vlan-id>,<st-node02>: a0a-<ai-ml-esxi-nfs-vlan-id>

network port vlan show
```

3. Create the iSCSI-A and iSCSI-B VLAN ports and add them to the respective iSCSI broadcast domains.

```
network port vlan create –node <st-node01> -vlan-name a0a-<ai-ml-iscsi-a-vlan-id>
network port vlan create –node <st-node02> -vlan-name a0a-<ai-ml-iscsi-a-vlan-id>

network port broadcast-domain add-ports -broadcast-domain AI-ML-iSCSI-A -ports <st-node01>:a0a-<ai-ml-
iscsi-a-vlan-id>,<st-node02>:a0a-<ai-ml-iscsi-a-vlan-id>

network port vlan create –node <st-node01> -vlan-name a0a-<ai-ml-iscsi-b-vlan-id>
network port vlan create –node <st-node02> -vlan-name a0a-<ai-ml-iscsi-b-vlan-id>

network port broadcast-domain add-ports -broadcast-domain AI-ML-iSCSI-B -ports <st-node01>:a0a-<ai-ml-
iscsi-b-vlan-id>,<st-node02>:a0a-<ai-ml-iscsi-b-vlan-id>
```

Configure Network Time Protocol

Follow the steps found in FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2.

Configure SNMP

Follow the steps found in FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2.

Configure SNMPv1 Access

Follow the steps found in FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2.

Create SVM

To create an SVM for the AI/ML workload, follow these steps:

1. Run the `vserver create` command.

```
vserver create -vserver AI-ML-SVM -subtype default -rootvolume AI_ML_SVM_root_vol -rootvolume-security-
style unix -aggregate aggr1_node01
```

2. Remove the unused data protocols from the SVM:

```
vserver remove-protocols -vserver AI-ML-SVM -protocols fcp,cifs
```

3. Add the two data aggregates to the AI-ML-SVM aggregate list for the NetApp VSC.

```
vserver modify -vserver AI-ML-SVM -aggr-list aggr1_node01, aggr1_node02 -allowed-protocols nfs,iscsi
```

4. Enable and run the NFS protocol in the AI-ML-SVM.

```
vserver nfs create -vserver AI-ML-SVM -udp disabled
```

If the NFS license was not installed during the cluster configuration, make sure to install the license before starting the NFS service.

5. Set the SVM vstorage parameter for the NetApp NFS VAAI plug-in to on.

```
vserver nfs modify –vserver AI-ML-SVM –vstorage enabled
vserver nfs show -fields vstorage
```

Create Load-Sharing Mirrors of SVM Root Volume

To create a load-sharing mirror of an SVM root volume, follow these steps:

1. Create a volume to be the load-sharing mirror of the AI-ML-SVM root volume on each node.

```
volume create -vserver AI-ML-SVM -volume ai_ml_svm_root_m01 -aggregate aggr1_node01 -size 1GB -type DP

volume create -vserver AI-ML-SVM -volume ai_ml_svm_root_m02 -aggregate aggr1_node02 -size 1GB -type DP
```

2. Create a job schedule to update the root volume mirror relationships every 15 minutes.

```
job schedule interval create -name 15min -minutes 15
```

3. Create the mirroring relationships.

```
snapmirror create -source-path AI-ML-SVM:ai_ml_svm_root -destination-path AI-ML-SVM:ai_ml_svm_root_m01 -
type LS -schedule 15min

snapmirror create -source-path AI-ML-SVM:ai_ml_svm_root -destination-path AI-ML-SVM:ai_ml_svm_root_m02 -
type LS -schedule 15min
```

4. Initialize the mirroring relationship.

```
snapmirror initialize-ls-set -source-path AI-ML-SVM:ai_ml_svm_root
snapmirror show -type ls
```

Create iSCSI Service

Enable and run the iSCSI protocol in the AI/ML SVM.

If the iSCSI license was not installed during the cluster configuration, make sure to install the license before starting the iSCSI service.

```
iscsi create -vserver AI-ML-SVM
```

Configure HTTPS Access

Follow the steps found FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2.

Configure NFSv3

To configure NFSv3 on the AI-ML-SVM, follow these steps:

1. Create a new rule for the infrastructure NFS subnet in the default export policy.

```
vserver export-policy rule create -vserver AI-ML-SVM -policyname default -ruleindex 1 -protocol nfs -
clientmatch <ai-ml-nfs-subnet-cidr> -rorule sys -rwrule sys -superuser sys -allow-suid false
```

2. Assign the FlexPod export policy to the AI/ML SVM root volume.

```
volume modify -vserver AI-ML-SVM -volume rootvol -policy default
```

Create ONTAP FlexGroup Volume

A single NetApp ONTAP FlexGroup volume was created to host the Imagenet dataset used in this validation. To create the ONTAP FlexGroup volume, run the following command:

```
flexgroup deploy -size 10TB -type RW -space-guarantee none -foreground true -vserver AI-ML-SVM -volume
imagenet_dataset
```

Create FlexVol Volumes for Bare-Metal Hosts

```
volume create -vserver AI-ML-SVM -volume Cseries_boot_AI_ML -aggregate aggr1_node01 -size 1TB -state
online -policy default -space-guarantee none -percent-snapshot-space 0

snapmirror update-ls-set -source-path AI-ML-SVM:rootvol
```

Create FlexVol Volumes for VMware ESXi Hosts

This step can be skipped if ESXi environment has already been setup using the steps in FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2

```
volume create -vserver AI-ML-SVM -volume esxi_boot_AI_ML -aggregate aggr1_node01 -size 100GB -state
online -policy default -space-guarantee none -percent-snapshot-space 0

volume create -vserver AI-ML-SVM -volume infra_datastore_AI_ML_1 -aggregate aggr1_node01 -size 500GB -
state online -policy default -junction-path /infra_datastore_AI_ML_1 -space-guarantee none -percent-
snapshot-space 0

volume create -vserver AI-ML-SVM -volume infra_datastore_AI_ML_2 -aggregate aggr1_node02 -size 500GB -
state online -policy default -junction-path /infra_datastore_AI_ML_2 -space-guarantee none -percent-
snapshot-space 0

volume create -vserver AI-ML-SVM -volume infra_swap -aggregate aggr1_node01 -size 100GB -state online -
policy default -junction-path /infra_swap -space-guarantee none -percent-snapshot-space 0 -efficiency-
policy none -snapshot-policy none

snapmirror update-ls-set -source-path AI-ML-SVM:rootvol
```

Create Bare-Metal Server Boot LUNs

The boot LUNs listed below were created and mapped to the C-series servers for installing RHEL. You can adjust the number and size of the boot LUNs based on their individual requirements.

```
lun create -vserver AI-ML-SVM -path /vol/Cseries_boot_AI_ML/C220-AI-ML-01 -size 300GB -ostype linux -
space-reserve disabled

lun create -vserver AI-ML-SVM -path /vol/Cseries_boot_AI_ML/C240-AI-ML-01 -size 300GB -ostype linux -
space-reserve disabled

lun create -vserver AI-ML-SVM -path /vol/Cseries_boot_AI_ML/C480-AI-ML-01 -size 300GB -ostype linux -
space-reserve disabled

(Optional – for any additional servers)

lun create -vserver AI-ML-SVM -path /vol/Cseries_boot_AI_ML/C240-AI-ML-02 -size 300GB -ostype linux -
space-reserve disabled

lun create -vserver AI-ML-SVM -path /vol/Cseries_boot_AI_ML/C480-AI-ML-02 -size 300GB -ostype linux -
space-reserve disabled
```

Create ESXi Boot LUNs

This step can be skipped if ESXi environment has already been setup using FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2.

You can adjust the number and size of the ESXi boot LUNs based on their individual requirements.

```
lun create -vserver AI-ML-SVM -path /vol/esxi_boot/VM-Host-Infra-AI-ML-01 -size 15GB -ostype vmware -
space-reserve disabled
lun create -vserver AI-ML-SVM -path /vol/esxi_boot/VM-Host-Infra-AI-ML-02 -size 15GB -ostype vmware -
space-reserve disabled
lun create -vserver AI-ML-SVM -path /vol/esxi_boot/VM-Host-Infra-AI-ML-03 -size 15GB -ostype vmware -
space-reserve disabled
lun create -vserver AI-ML-SVM -path /vol/esxi_boot/VM-Host-Infra-AI-ML-04 -size 15GB -ostype vmware -
space-reserve disabled
```

Modify Volume Efficiency

This step can be skipped if ESXi environment has already been setup using FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2.

On NetApp All Flash FAS systems, deduplication is enabled by default. To disable the efficiency policy on the infra_swap volume, complete the following step:

```
vol efficiency off -vserver AI-ML-SVM -volume infra_swap
```

Schedule Deduplication

On NetApp AFF systems, deduplication is enabled by default. To schedule deduplication, assign a once-a-day deduplication schedule to the various volumes:

```
vol efficiency modify -vserver AI-ML-SVM -volume imagenet_dataset -schedule sun-sat@0
vol efficiency modify -vserver AI-ML-SVM -volume C480_boot_AI_ML -schedule sun-sat@0
```

Create NFS LIFs

To create NFS LIFs for ESXi and AI/ML traffic, run the following commands:

```
network interface create -vserver AI-ML-SVM -lif AI-ML-NFS-1 -role data -data-protocol nfs -home-node
<st-node01> -home-port a0a-<ai-ml-nfs-vlan-id> -netmask <nfs_netmask> -status-admin up -address <ai_ml
_nfs_lif_01>

network interface create -vserver AI-ML-SVM -lif AI-ML-NFS-2 -role data -data-protocol nfs -home-node
<st-node02> -home-port a0a-<ai-ml-nfs-vlan-id> -netmask <nfs_netmask> -status-admin up -address <ai_ml
_nfs_lif_02>
```

This step assumes that the NFS LIFs for ESXi hosts in the VMware environment have already been configured using steps in FlexPod Datacenter with NetApp ONTAP 9.6, Cisco UCS 4th Generation, and VMware vSphere 6.7 U2.

Create iSCSI LIFs

To create iSCSI LIFs booting AI/ML hosts, run the following commands:

```
network interface create -vserver AI-ML-SVM -lif AI-ML-iSCSI-LIF-01-A -role data -data-protocol iscsi -
address <ai_ml_iscsi_lif_01a> -netmask <iscsi_netmask> -home-node <st-node01> -home-port a0a-<ai-ml-
iscsi-a-vlan-id> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false

network interface create -vserver AI-ML-SVM -lif AI-ML-iSCSI-LIF-02-A -role data -data-protocol iscsi -
address <ai_ml_iscsi_lif_02a> -netmask <iscsi_netmask> -home-node <st-node02> -home-port a0a-<ai-ml-
iscsi-a-vlan-id> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false

network interface create -vserver AI-ML-SVM -lif AI-ML-iSCSI-LIF-01-B -role data -data-protocol iscsi -
address <ai_ml_iscsi_lif_01b> -netmask <iscsi_netmask> -home-node <st-node01> -home-port a0a-<ai-ml-
iscsi-b-vlan-id> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false

network interface create -vserver AI-ML-SVM -lif AI-ML-iSCSI-LIF-02-B -role data -data-protocol iscsi -
address <ai_ml_iscsi_lif_02b> -netmask <iscsi_netmask> -home-node <st-node02> -home-port a0a-<ai-ml-
iscsi-b-vlan-id> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false
```

Add AI-ML SVM Administrator

To add the infrastructure SVM administrator and SVM administration LIF in the out-of-band management network, follow these steps:

1. Run the following commands:

```
network interface create –vserver AI-ML-SVM –lif ai_ml_svm_mgmt –role data –data-protocol none –home-node
<st-node02> -home-port  a0a-<ib-mgmt-vlan-id> –address <ai_ml_svm_mgmt_ip> –netmask <ai_ml_svm_mgmt_mask>
-status-admin up –failover-policy broadcast-domain-wide –firewall-policy mgmt –auto-revert true
```

2. Create a default route to allow the SVM management interface to reach the outside world.

```
network route create –vserver AI-ML-SVM -destination 0.0.0.0/0 –gateway <ai_ml_svm_mgmt_gateway>

network route show
```

3. Set a password for the SVM vsadmin user and unlock the user.

```
security login password –username vsadmin –vserver AI-ML-SVM
Enter a new password:  <password>
Enter it again:  <password>

security login unlock –username vsadmin –vserver AI-ML-SVM
```

Cisco UCS Configuration for VMware with vGPU

If you are deploying GPUs in the pre-existing VMware environment, this section explains the configuration additions required to support the AI/ML workloads.

Cisco UCS Base Configuration

For the base configuration for the Cisco UCS 6454 Fabric Interconnect, follow the Cisco UCS Configuration section here: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_datacenter_vmware_netappaffa_u2.html. To enable VMs in the existing VMware Infrastructure to access the NFS VLAN where the AI/ML dataset is hosted, the following configuration is required on the Cisco UCS:

- Define the NFS VLAN (3152) on the UCS
- Add the NFS VLAN to the (updating) vNIC templates

Create NFS VLAN

To create a new VLAN in the Cisco UCS, follow these steps:

1. In Cisco UCS Manager, click the LAN icon.
2. Select LAN > LAN Cloud.
3. Right-click VLANs.
4. Select Create VLANs.
5. Enter “`AI-ML-NFS`” as the name of the VLAN to be used to access NFS datastore hosting imagenet data.
6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter the native VLAN ID <3152>.
8. Keep the Sharing Type as None.
9. Click OK and then click OK again.

ADD VLAN to vNIC Template

To add the newly created VLAN in existing vNIC templates configured for ESXi hosts, follow these steps:

1. In the UCS Manager, click the LAN icon.
2. Select LAN > Policies > root > vNIC Templates (select the sub-organization if applicable).
3. Select the Fabric-A vNIC template used for ESXi host (for example, vSwitch0-A).
4. In the main window “General”, click Modify VLANs.
5. Check the box to add the NFS VLAN (3152) and click OK.
6. Repeat this procedure to add the same VLAN to the Fabric-B vNIC template (for example, vSwitch0-B).

When the NFS VLAN is added to appropriate vSwitch on the ESXi host, a port-group can be created in the VMware environment to provide VMs access to the NFS share.

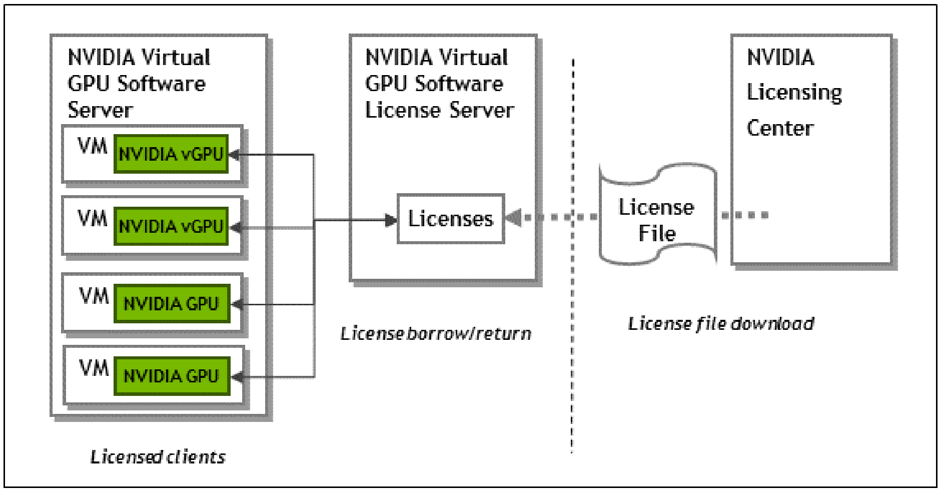
VMware Setup and Configuration for vGPU

This deployment assumes you have deployed Cisco UCS C220 M5, C240 M5 or C480 ML M5 servers using the configuration for vSphere setup as explained in: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_datacenter_vmware_netappaffa_u2.html.

Obtaining and installing NVIDIA vGPU Software

NVIDIA vGPU software is a licensed product. Licensed vGPU functionalities are activated during guest OS boot by the acquisition of a software license served over the network from an NVIDIA vGPU software license server. The license is returned to the license server when the guest OS shuts down.

Figure 6 NVIDIA vGPU Software Architecture



To utilize GPUs in a VM environment, following configuration steps must be completed:

- Create an NVIDIA Enterprise Account and add appropriate product licenses
- Deploy a Windows based VM as NVIDIA vGPU License Server and install license file
- Download and install NVIDIA software on the hypervisor
- Setup VMs to utilize GPUs

For detailed installation instructions, visit NVIDA vGPU installation guide: https://docs.nvidia.com/grid/latest/grid-software-quick-start-guide/index.html

NVIDIA Licensing

To obtain the NVIDIA vGPU software from NVIDIA Software Licensing Center, follow these steps:

1. Create NVIDIA Enterprise Account using following steps: https://docs.nvidia.com/grid/latest/grid-software-quick-start-guide/index.html#creating-nvidia-enterprise-account
2. To redeem the product activation keys (PAK), use the following steps: https://docs.nvidia.com/grid/latest/grid-software-quick-start-guide/index.html#redeeming-pak-and-downloading-grid-software

Download NVIDIA vGPU Software

To download the NVIDIA vGPU software, follow these steps:

1. After the product activation keys have seen successfully redeemed, login to the Enterprise NVIDIA Account (if needed): https://nvidia.flexnetoperations.com/control/nvda/content?partnerContentId=NvidiaHomeContent
2. Click Product Information and then NVIDIA Virtual GPU Software version 9.1 (https://nvidia.flexnetoperations.com/control/nvda/download?element=11233147)
3. Click NVIDIA vGPU for vSphere 6.7 and download the zip file (NVIDIA-GRID-vSphere-6.7-430.46-431.79.zip).
4. Scroll down and click 2019.05 64-bit License Manager for Windows to download the License Manager software for the Windows (NVIDIA-ls-windows-64-bit-2019.05.0.26416627.zip).

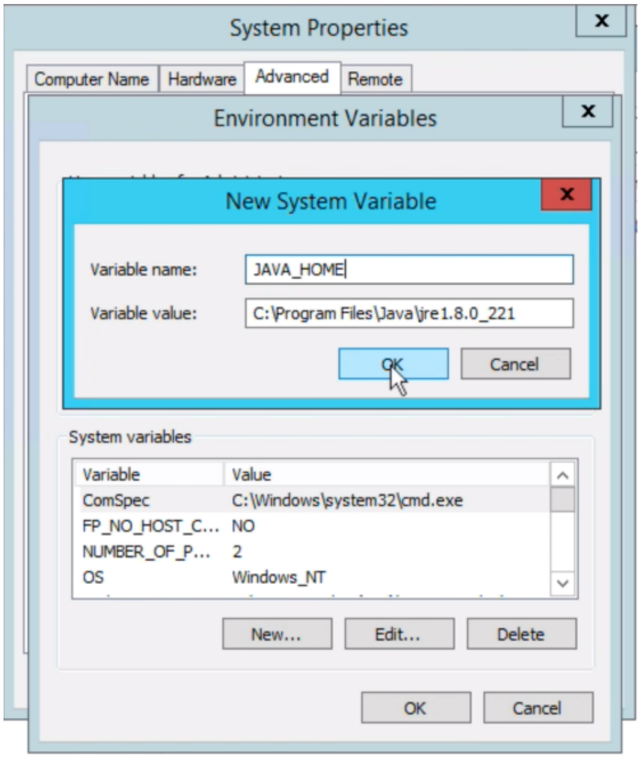
Setup NVIDIA vGPU Software License Server

The NVIDIA vGPU software License Server is used to serve a pool of floating licenses to NVIDIA vGPU software licensed products. The license server is designed to be installed at a location that is accessible from a customer's network and be configured with licenses obtained from the NVIDIA Software Licensing Center.

Refer to the NVIDIA Virtual GPU Software License Server Documentation: https://docs.nvidia.com/grid/ls/latest/grid-license-server-user-guide/index.html for setting up the vGPU software license server.

To setup a standalone license server, follow these steps::

1. Deploy a windows server 2012 VM with following hardware parameters:
   a. 2 vCPUs
   b. 4GB RAM
   c. 100GB HDD
   d. 64-bit Operating System
   e. Static IP address
   f. Internet access
   g. Latest version of Java Runtime Environment
2. Copy the previously downloaded License Manager installation file (NVIDIA-ls-windows-64-bit-2019.05.0.26416627.zip) to the above VM, unzip and double click Setup-x64.exe to install the License Server.
3. If a warning about JAVA_HOME environmental variable not defined is received, add the variable manually using following steps:
   a. Open Control Panel and change view by on top right to Large Icons
   b. Click and open System
   c. Click and open Advanced system settings
   d. Click Environmental Variables
   e. Click New under System variables
   f. Add the variable name and path where Java Runtime Environment is deployed:



Variable name: JAVA_HOME

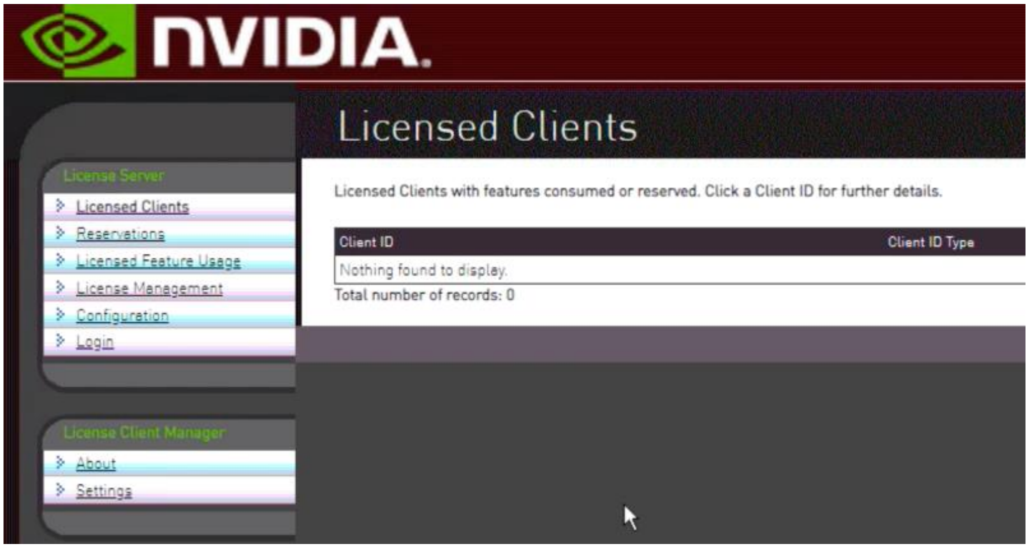
Variable value: C:\Program Files\Java\jre1.8.0_221

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT

g. Click OK multiple times to accept the changes and close the configuration dialog boxes.

h. Run the installer again and follow the prompts.

4. When the installation is complete, open a web browser and enter the following URL to access the License Server: http://localhost:8080/licserver



Licensed Clients

Licensed Clients with features consumed or reserved. Click a Client ID for further details.

Client ID	Client ID Type
Nothing found to display.	

Total number of records: 0

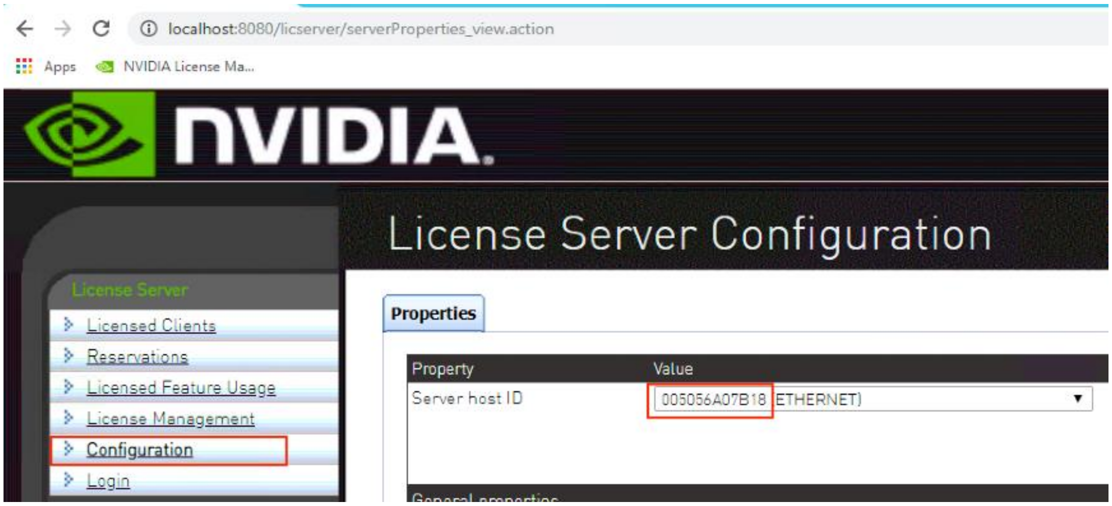
The license server uses Ports 8080 and 7070 to manager the server and for client registration. These ports should be enabled across the firewalls (if any).

In actual customer deployments, redundant license servers must be installed for high availability. Refer to the NVIDIA documentation for the high availability requirements: https://docs.nvidia.com/grid/ls/latest/grid-license-server-user-guide/index.html#license-server-high-availability-requirements

Register License Server to NVIDIA Software Licensing Center

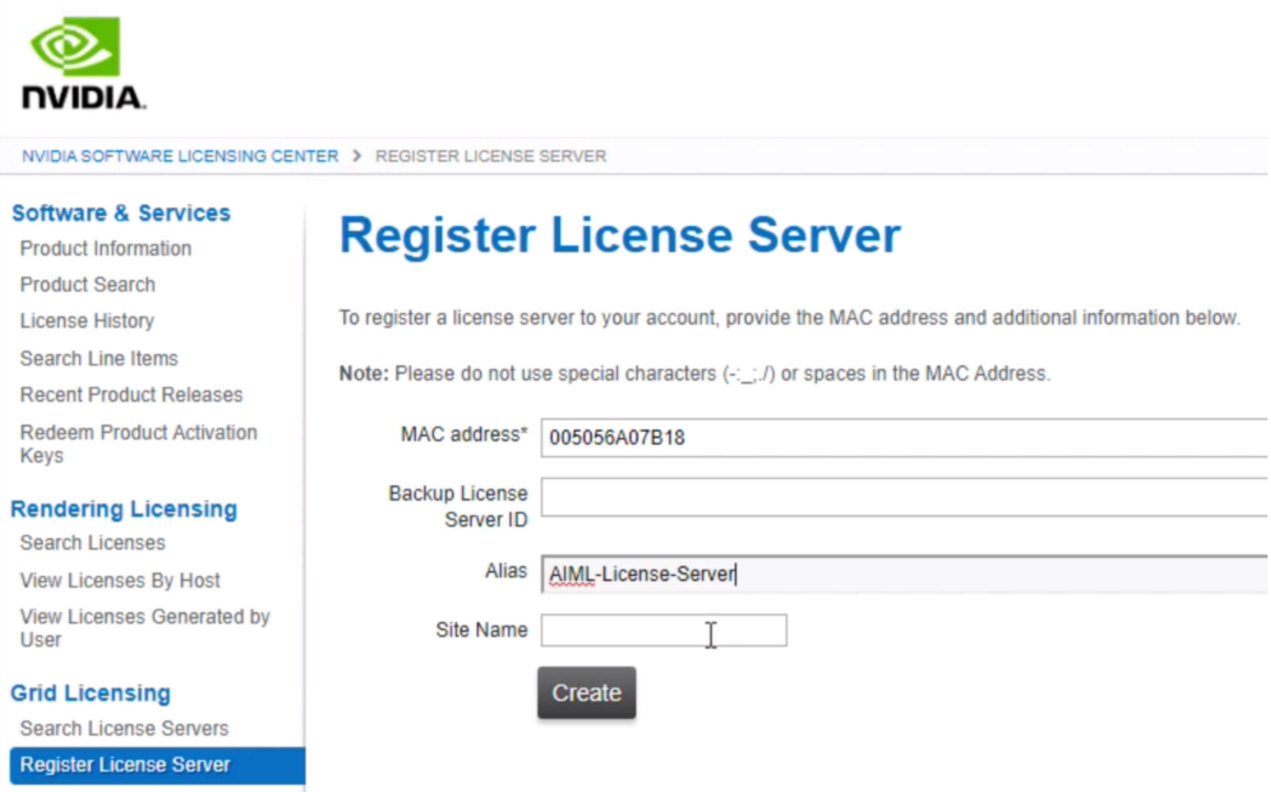
To enable the License server to obtain and distribute licenses to the clients, the license server must be registered to NVIDIA Software Licensing Center.

1. Log into the NVIDIA Enterprise account and browse to NVIDIA Software License Center.
2. Click Register License Server link.
3. The license server registration form requires the MAC address of the license server being registered. This information can be retrieved by opening the license server management interface (http://localhost:8080/licserver) and clicking on the Configuration.



Property	Value
Server host ID	005056A07B18 ETHERNET)

4. Enter the MAC address and an alias and click Create.



Register License Server

To register a license server to your account, provide the MAC address and additional information below.

Note: Please do not use special characters (-:_;./) or spaces in the MAC Address.

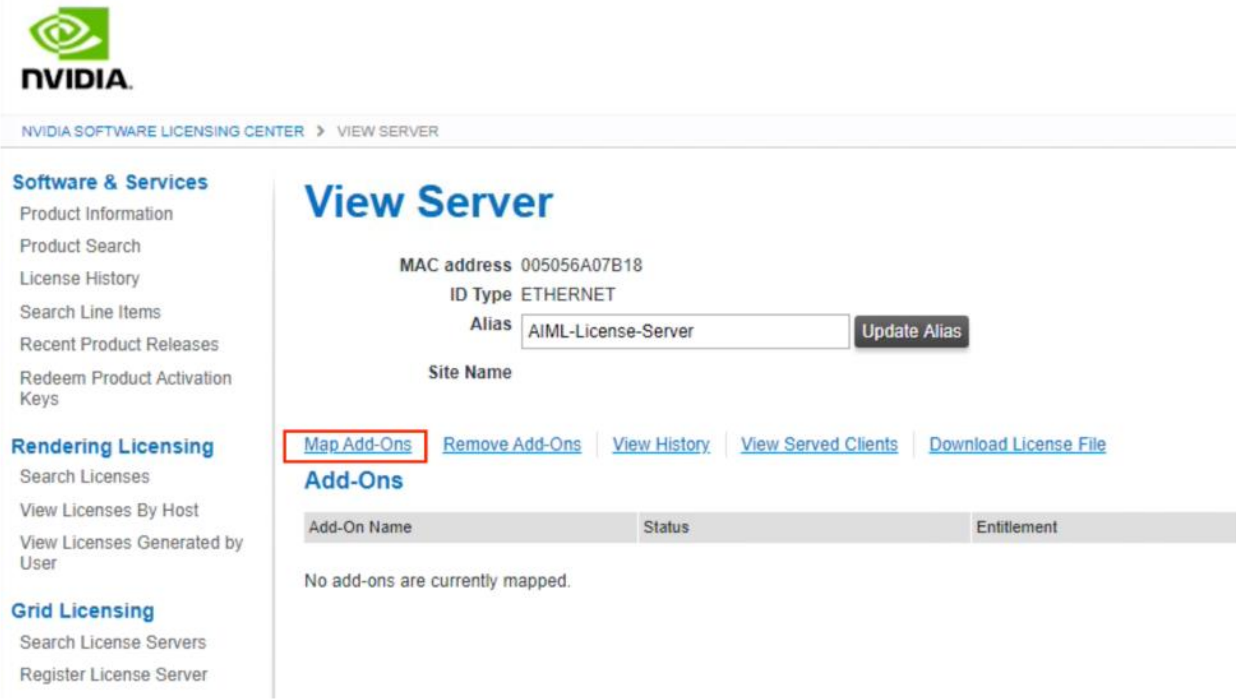
MAC address* 005056A07B18

Backup License Server ID

Alias AIML-License-Server

Site Name

5. On the next page, Click Map Add-Ons to map the appropriate license feature(s).



View Server

MAC address 005056A07B18

ID Type ETHERNET

Alias AIML-License-Server

Site Name

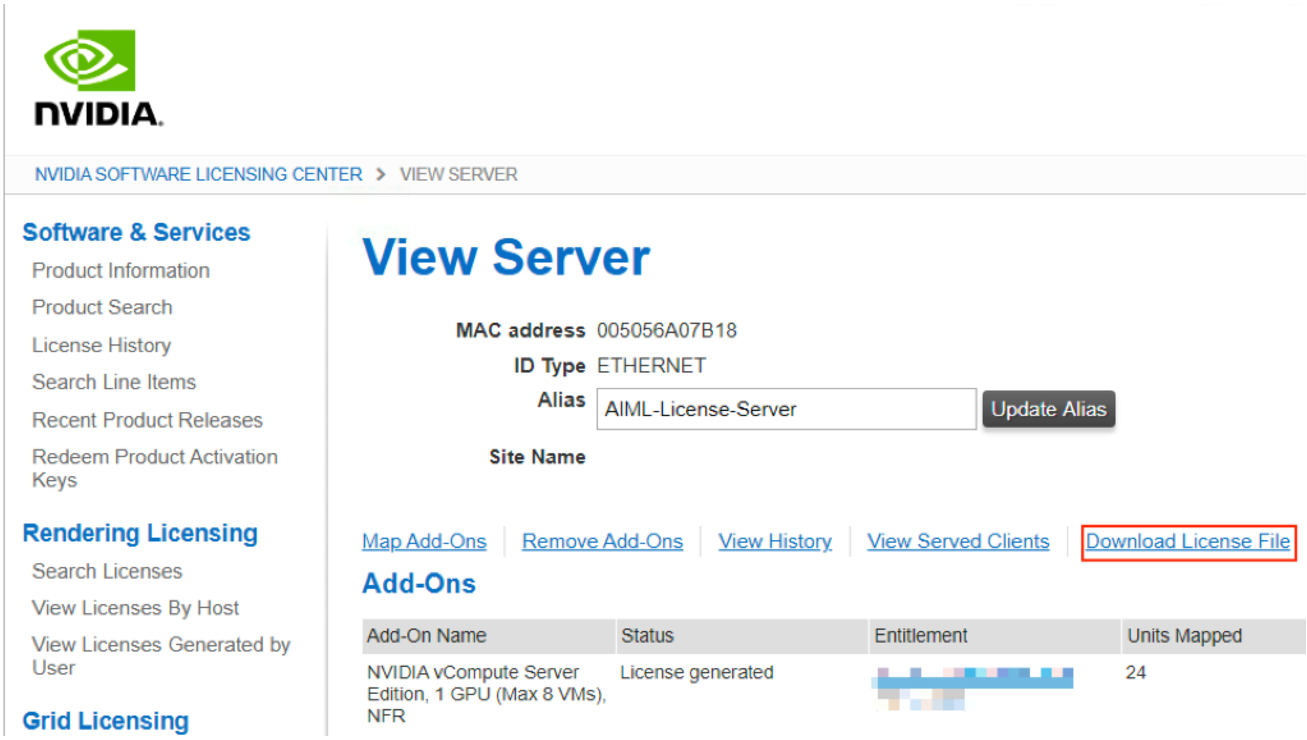
Add-Ons

Add-On Name	Status	Entitlement

No add-ons are currently mapped.

6. On the following page, select the appropriate licensed feature (NVIDIA vCompute Server Edition) and quantity and click Map Add-Ons.

7. Click Download License File and copy this file over to the license server VM if the previous steps were performed in a different machine.



View Server

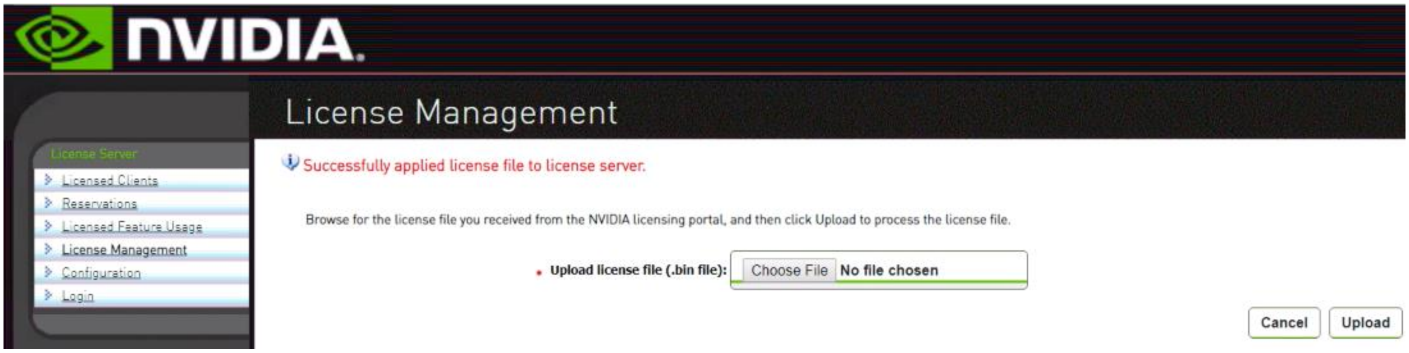
MAC address	005056A07B18
ID Type	ETHERNET
Alias	AIML-License-Server
Site Name	

Add-Ons

Add-On Name	Status	Entitlement	Units Mapped
NVIDIA vCompute Server Edition, 1 GPU (Max 8 VMs), NFR	License generated	[redacted]	24

8. On the license server management console, click License Management and Choose File to select the file downloaded in the last step.

9. Click Upload to upload the file to the license server.



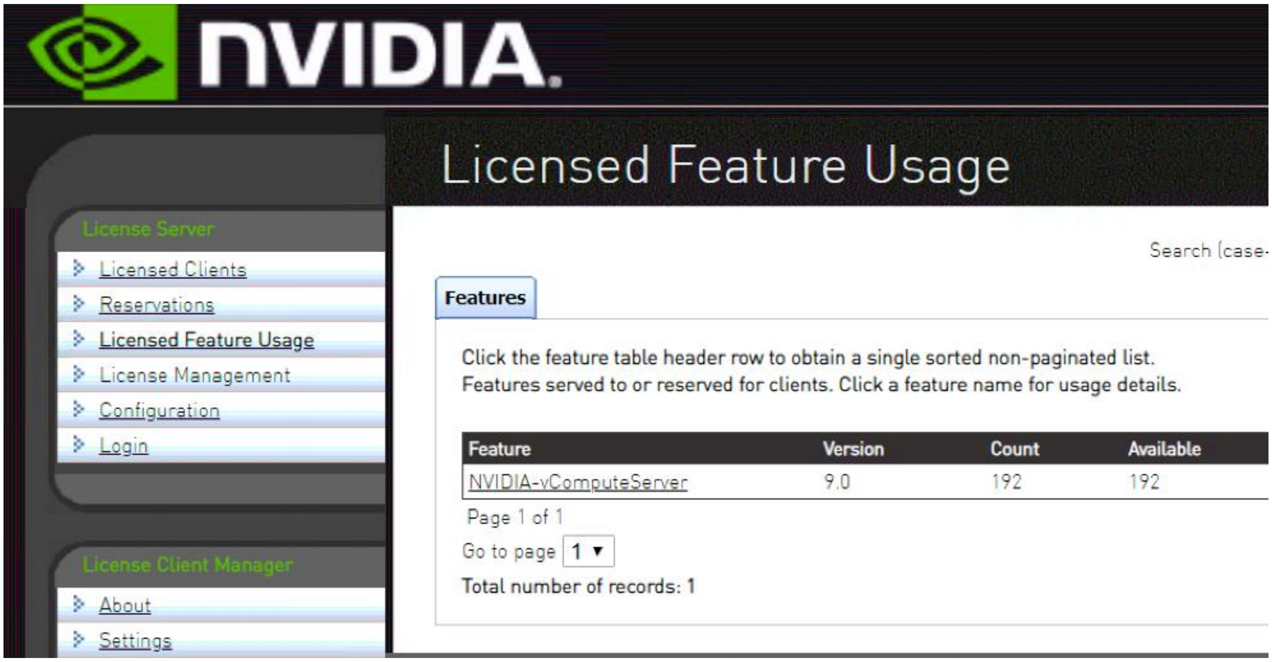
License Management

Successfully applied license file to license server.

Browse for the license file you received from the NVIDIA licensing portal, and then click Upload to process the license file.

• Upload license file (.bin file): No file chosen

10. Click the License Feature Usage to verify the license was installed properly.



Licensed Feature Usage

Click the feature table header row to obtain a single sorted non-paginated list.
Features served to or reserved for clients. Click a feature name for usage details.

Feature	Version	Count	Available
NVIDIA-vComputeServer	9.0	192	192

Page 1 of 1

Go to page 1 ▾

Total number of records: 1

The License Server is now configured to serve licenses to the VMs.

Install NVIDIA vGPU Manager in ESXi

Before guests enabled for NVIDIA vGPU can be configured, the NVIDIA Virtual GPU Manager must be installed on the ESXi hosts by following these steps:

1. Unzip the downloaded file NVIDIA-GRID-vSphere-6.7-430.46-431.79.zip to extract the software VIB file: NVIDIA-VMware_ESXi_6.7_Host_Driver-430.46-1OEM.670.0.0.8169922.x86_64.vib.
2. Copy the file to one of the shared datastores on the ESXi servers; in this example, the file was copied to the datastore infra_datastore_1.
3. Right-click the ESXi host and select Maintenance Mode -> Enter Maintenance Mode.
4. SSH to the ESXi server and install the vib file:

```
[root@AIML-ESXi:~] esxcli software vib install -v /vmfs/volumes/infra_datastore_1/NVIDIA-
VMware_ESXi_6.7_Host_Driver-430.46-1OEM.670.0.0.8169922.x86_64.vib
Installation Result
   Message: Operation finished successfully.
   Reboot Required: false
   VIBs Installed: NVIDIA_bootbank_NVIDIA-VMware_ESXi_6.7_Host_Driver_430.46-1OEM.670.0.0.8169922
   VIBs Removed:
   VIBs Skipped:
```

5. Reboot the host from vSphere client or from the CLI.
6. Log back into the host once the reboot completes and issue the following command to verify the driver installation on the ESXi host:

```
 [root@AIML-ESXi:~] nvidia-smi
Fri Oct 11 05:33:09 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 430.46       Driver Version: 430.46       CUDA Version: N/A      |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:1B:00.0 Off |                    0 |
| N/A   43C    P0    49W / 300W |     61MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  On   | 00000000:1C:00.0 Off |                    0 |
| N/A   42C    P0    46W / 300W |     61MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-SXM2...  On   | 00000000:42:00.0 Off |                    0 |
| N/A   42C    P0    45W / 300W |     61MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-SXM2...  On   | 00000000:43:00.0 Off |                    0 |
| N/A   43C    P0    43W / 300W |     61MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   4  Tesla V100-SXM2...  On   | 00000000:89:00.0 Off |                    0 |
| N/A   42C    P0    46W / 300W |     61MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   5  Tesla V100-SXM2...  On   | 00000000:8A:00.0 Off |                    0 |
| N/A   42C    P0    46W / 300W |     61MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   6  Tesla V100-SXM2...  On   | 00000000:B2:00.0 Off |                    0 |
| N/A   41C    P0    45W / 300W |     61MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   7  Tesla V100-SXM2...  On   | 00000000:B3:00.0 Off |                    0 |
| N/A   41C    P0    46W / 300W |     61MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0   2102601      G   Xorg                                          5MiB |
|    1   2102618      G   Xorg                                          5MiB |
|    2   2102639      G   Xorg                                          5MiB |
|    3   2102658      G   Xorg                                          5MiB |
|    4   2102679      G   Xorg                                          5MiB |
|    5   2102696      G   Xorg                                          5MiB |
|    6   2102716      G   Xorg                                          5MiB |
|    7   2102736      G   Xorg                                          5MiB |
+-----------------------------------------------------------------------------+
```

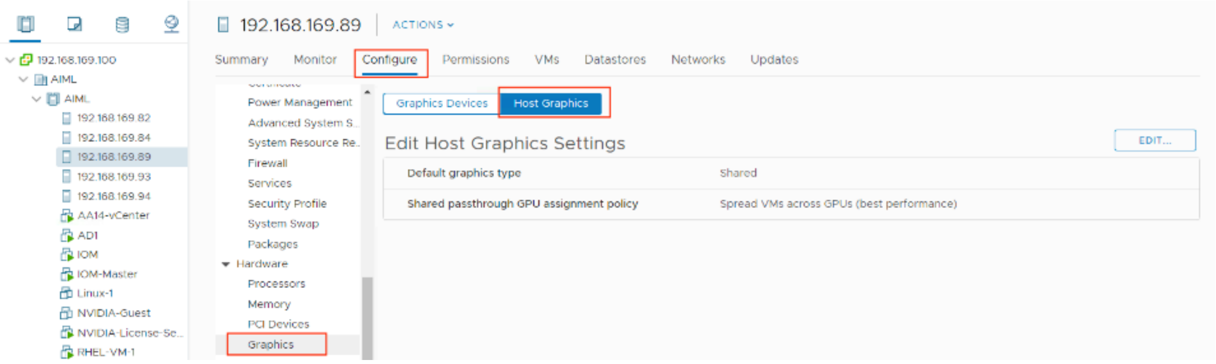
The output of the command “nvidia-smi” will vary depending on the ESXi host and the type and number of GPUs.

7. Right-click the ESXi host and select Maintenance Mode -> Exit Maintenance Mode.
8. Repeat steps 1-7 to install the vGPU manager on all the appropriate ESXi hosts.

Set the Host Graphics to SharedPassthru

A GPU card can be configured in shared virtual graphics mode or the vGPU (SharedPassthru) mode. For the AI/ML workloads, the NVIDIA card should be configured in the SharedPassthru mode. A server reboot is required when this setting is modified. To set the host graphics, follow these steps:

1. Click the ESXi host in the vSphere client and select Configure.
2. Scroll down and select Graphics and select Host Graphics from the main windows.

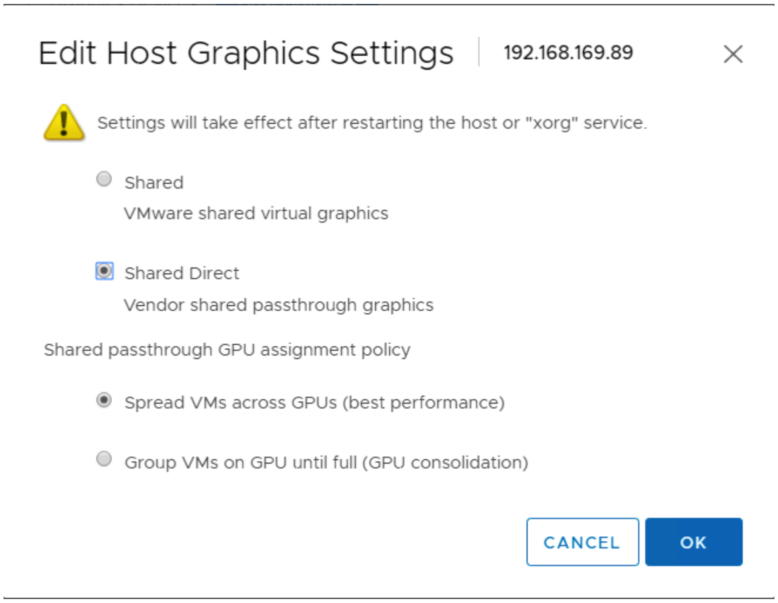


Edit Host Graphics Settings

Default graphics type	Shared
Shared passthrough GPU assignment policy	Spread VMs across GPUs (best performance)

3. Click Edit.

4. Select Shared Direct and click OK.


Edit Host Graphics Settings | 192.168.169.89

Settings will take effect after restarting the host or "xorg" service.

○ Shared
VMware shared virtual graphics

● Shared Direct
Vendor shared passthrough graphics

Shared passthrough GPU assignment policy

● Spread VMs across GPUs (best performance)

○ Group VMs on GPU until full (GPU consolidation)

5. Reboot the ESXi host after enabling Maintenance Mode. Remember to exit Maintenance Mode when the host comes back up.

6. Repeat steps 1-5 for all the appropriate ESXi hosts.

(Optional) Enabling vMotion with vGPU

To enable VMware vMotion with vGPU, an advanced vCenter Server setting must be enabled.

For details about which VMware vSphere versions, NVIDIA GPUs, and guest OS releases support VM with vGPU migration, see: https://docs.nvidia.com/grid/latest/grid-vgpu-release-notes-vmware-vsphere/index.html

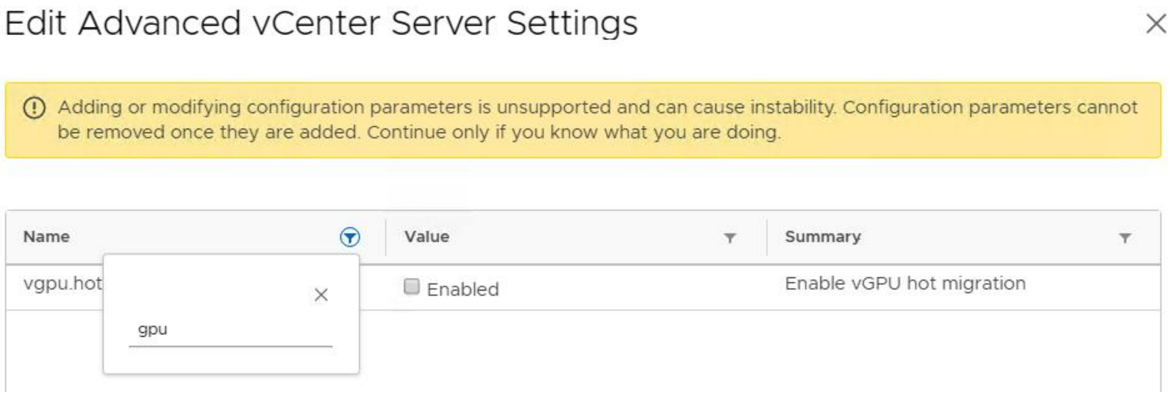
To enable vMotion with vGPU, follow these steps:

1. Log into vCenter Server using the vSphere Web Client.

2. In the Hosts and Clusters view, select the vCenter Server instance.

Ensure that the vCenter Server instance is selected, not the vCenter Server VM.

3. Click the Configure tab.
4. In the Settings section, select Advanced Settings and click Edit.
5. In the Edit Advanced vCenter Server Settings window that opens, type vGPU in the search field.
6. When the vgpu.hotmigrate.enabled setting appears, set the Enabled option and click OK.



Edit Advanced vCenter Server Settings

Adding or modifying configuration parameters is unsupported and can cause instability. Configuration parameters cannot be removed once they are added. Continue only if you know what you are doing.

Name	Value	Summary
vgpu.hot	□ Enabled	Enable vGPU hot migration

Add a Port-Group to access AI/ML NFS Share

You can choose to access the NFS share hosting Imagenet data (AI/ML data set) in one of the following two ways:

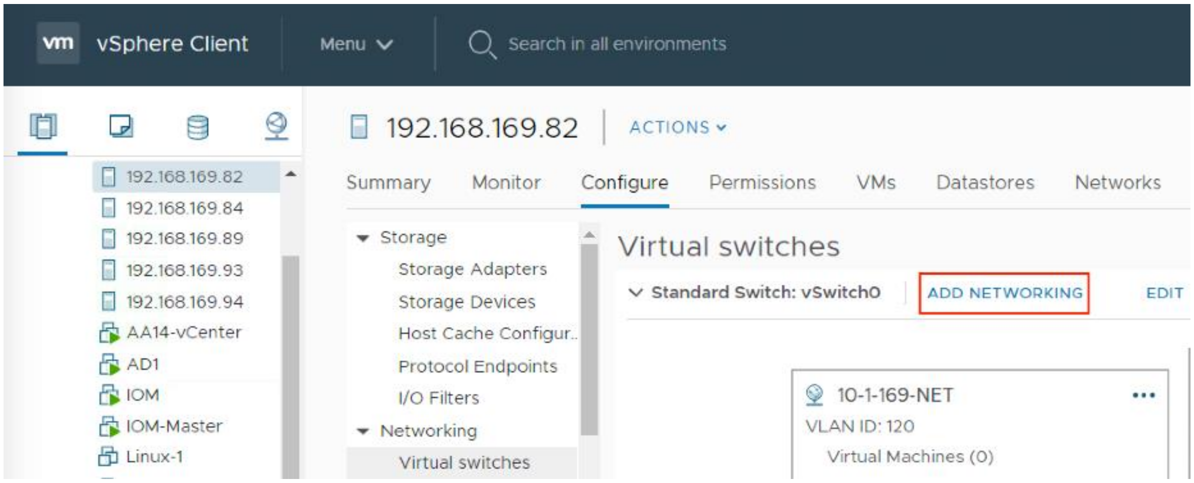
1. Using a separate NIC assigned to the port-group setup to access AI/ML NFS VLAN (for example, 3152).
2. Over the VM's management interface if the network is setup for routing between VM's IP address and the NFS LIF IP address.

In this deployment, a separate NIC was used to access the NFS share to keep the management traffic separate form NFS traffic and to be able to access the NFS share over directly connected network without having to route.

To define a new port-group follow the steps below for all the ESXi hosts:

1. Log into the vSphere client and click the host under Hosts and Clusters in the left side bar.
2. In the man window, select Configure > Networking > Virtual Switches.
3. Select ADD NETWORKING next to the vSwitch0.

In this example, NFS VLAN was added to the vNIC template associated with vSwitch0. If a customer decides to use a different vSwitch and has added the VLAN to corresponding vNIC template, select the appropriate vSwitch here.



4. In the Add Networking window, select Virtual Machine Port Group for a Standard Switch and click NEXT.
5. Select an existing vSwitch and make sure vSwitch0 is selected and click NEXT.
6. Provide a Network Label (e.g. 192-168-52-NFS) and VLAN (e.g. 3152). Click NEXT.
7. Verify the information and click FINISH.

The port-group is now configured to be assigned to the VMs.

Red Hat Enterprise Linux VM Setup

NVIDIA V100 and T4 GPUs support various vGPU profiles. These profiles, along with their intended use, are outlined in the NVIDIA documentation:

- NVIDIA T4 vGPU Types:

  https://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html#vgpu-types-tesla-t4
  https://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html - vgpu-types-tesla-t4

- NVIDIA V100 SXM2 32GB vGPU Types:

  https://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html#vgpu-types-tesla-v100-sxm2-32gb

- NVIDIA V100 PCIE 32GB vGPU Types:

  https://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html#vgpu-types-tesla-v100-pcie-32gb

GPU profiles for VComputeServer workloads end with “C” in the profile name. For example, NVIDIA T4 GPU supports following vGPU profiles: T4-16C, T4-8C and T4-4C where 16, 8, and 4 represent frame buffer memory in GB. Because C-Series vComputeServer vGPUs have large BAR (Base Address Registers) memory settings, using these vGPUs has some restrictions in VMware ESXi:

- The guest OS must be a 64-bit OS.
- 64-bit MMIO and EFI boot must be enabled for the VM.
- The guest OS must be able to be installed in EFI boot mode.

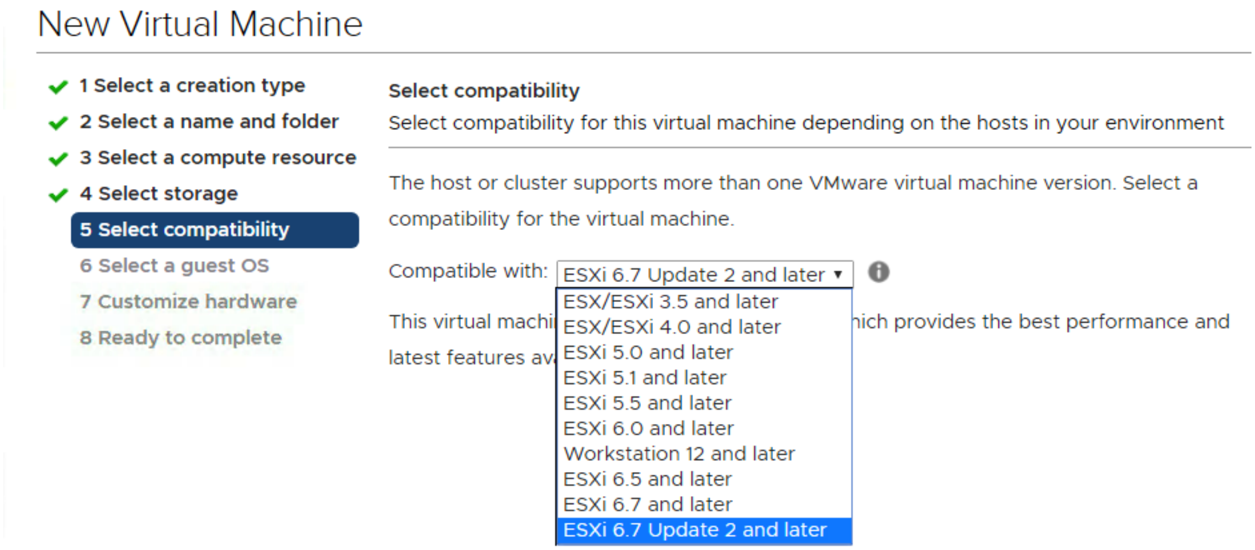
- The VM's MMIO space must be increased to 64 GB (refer to VMware KB article: https://kb.vmware.com/s/article/2142307). When using multiple vGPUs with single VM, this value might need to be increased to match the total memory for all the vGPUs.
- To use multiple vGPUs in a VM, set the VM compatibility to vSphere 6.7 U2.

Refer to the NVIDIA vGPU software documentation: https://docs.nvidia.com/grid/latest/grid-vgpu-release-notes-vmware-vsphere/index.html#validated-platforms for various device settings and requirements

VM Hardware Setup

To setup a RHEL VM for running AI/ML workloads, follow these steps:

1. In the vSphere client, right-click in the ESXi host and select New Virtual Machine.
2. Select Create a new virtual machine and click NEXT.
3. Provide Virtual Machine Name and optionally select an appropriate folder. Click NEXT.
4. Make sure correct Host is selected and Compatibility checks succeeded. Click NEXT.
5. Select a datastore (e.g. infra_datastore_1) and click NEXT.
6. From the drop-down menu, select ESXi 6.7 update 2 and later and click NEXT.



New Virtual Machine

Select compatibility

Select compatibility for this virtual machine depending on the hosts in your environment

The host or cluster supports more than one VMware virtual machine version. Select a compatibility for the virtual machine.

Compatible with: ESXi 6.7 Update 2 and later

This virtual machi hich provides the best performance and latest features av

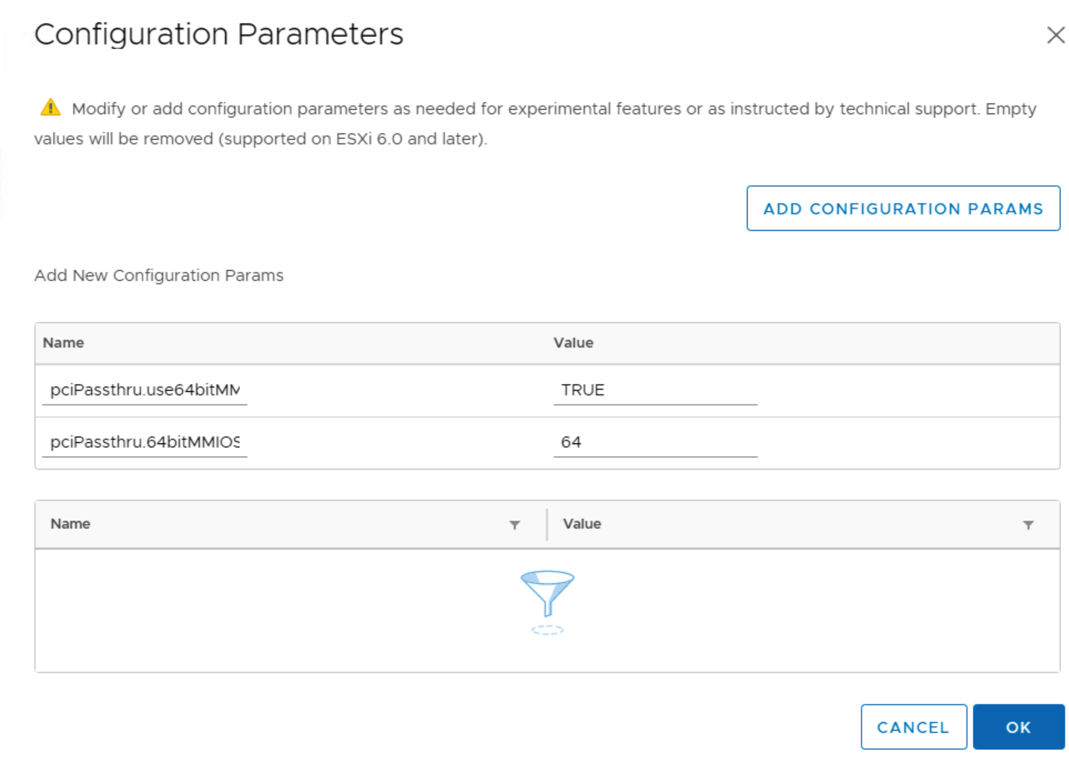
7. From the drop-down list, select Linux as the Guest OS Family and Red Hat Enterprise Linux 7 (64-bit) as the Guest OS Version. Click NEXT.
8. Change the number of CPUs and Memory to match workload requirements (2 vCPUs and 16GB memory was selected in this example).
9. Select appropriate network under NEW Network.
10. (Optional) Click ADD NEW DEVICE and add a second Network Adapter.
11. For the network, select the NFS Port-Group (e.g. 192-168-52-NFS) where AIML data (imagenet) resides.

This deployment assumes each ESXi host is pre-configured with a VM port-group providing layer-2 access to NetApp datastore where Imagenet dataset is hosted.

If this VM is going to be converted into a base OS template, do not add vGPUs at this time. The vGPUs will be added later.

12. Click VM Options.
13. Expand Boot Options and under Firmware, select EFI (ignore the warning since this is a fresh install).
14. Expand Advanced and click EDIT CONFIGURATION…
15. Click ADD CONFIGURATION PARAMS twice and add pciPassthru.64bitMMIOSizeGB with value of 64* and pciPassthru.use64bitMMIO with value of TRUE. Click OK.

This value should be adjusted based on number of GPUs assigned to the VM. For example, if a VM is assigned 4 x 32GB V100 GPUs, this value should be 128.


Configuration Parameters

⚠ Modify or add configuration parameters as needed for experimental features or as instructed by technical support. Empty values will be removed (supported on ESXi 6.0 and later).

Add New Configuration Params

Name	Value
pciPassthru.use64bitMM	TRUE
pciPassthru.64bitMMIOS	64

Name	Value

16. Click NEXT and after verifying various selections, click FINISH.
17. Right-click the newly created VM and select Open Remote Console to bring up the console.
18. Click the Power On button.

Download RHEL 7.6 DVD ISO

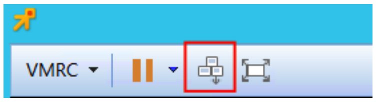
If the RHEL DVD image has not been downloaded, follow these steps to download the ISO:

1. Click the following link RHEL 7.6 Binary DVD.
2. A user_id and password are required on the website (redhat.com) to download this software.
3. Download the .iso (rhel-server-7.6-x86_64-dvd.iso) file.
4. Follow the prompts to launch the KVM console.

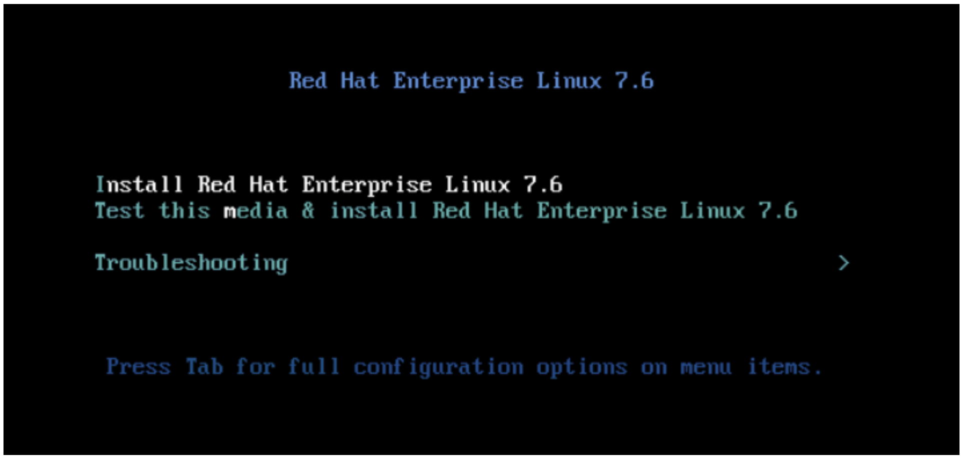
Operating System Installation

To prepare the server for the OS installation, make sure the VM is powered on and follow these steps:

1. In the VMware Remote Console window, click VMRC -> Removable Devices -> CD/DVD Drive 1 -> Connect to Disk Image File (iso)
2. Browse and select the RHEL ISO file and click Open.
3. Press the Send Ctl+Alt+Del to Virtual machine button.



4. On reboot, the VM detects the presence of the RHEL installation media. From the Installation menu, use arrow keys to select Install Red Hat Enterprise Linux 7.6. This should stop automatic boot countdown.

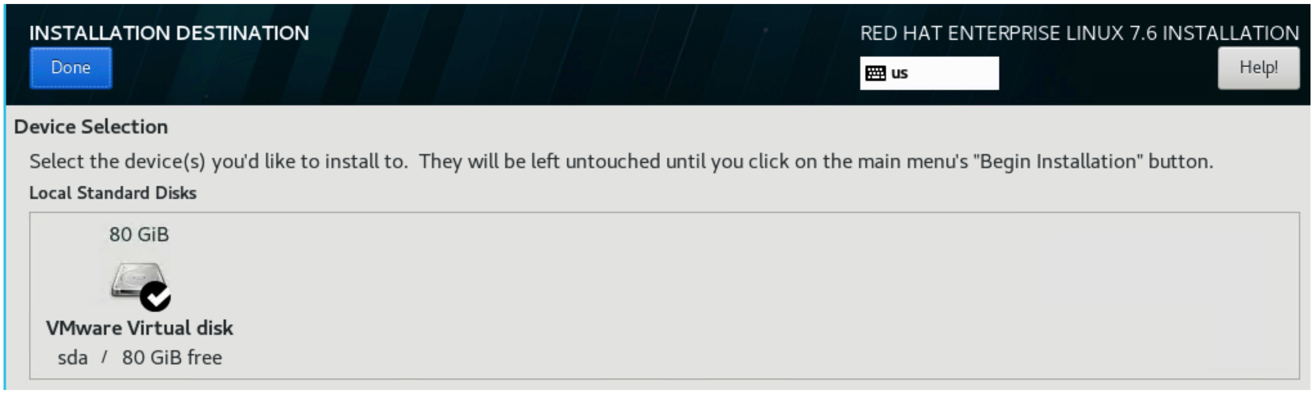


```
Red Hat Enterprise Linux 7.6

Install Red Hat Enterprise Linux 7.6
Test this media & install Red Hat Enterprise Linux 7.6

Troubleshooting                                          >

Press Tab for full configuration options on menu items.
```

5. Press Enter to continue the boot process.
6. After the installer finishes loading, select the language and press Continue.
7. On the Installation Summary screen, leave the software selection to Minimal Install.

It might take a minute for the system to check the installation source. During this time, Installation Source will be grayed out. Wait for the system to load the menu items completely.

8. Click the Installation Destination to select the VMware Virtual disk as installation disk.



Device Selection

Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.

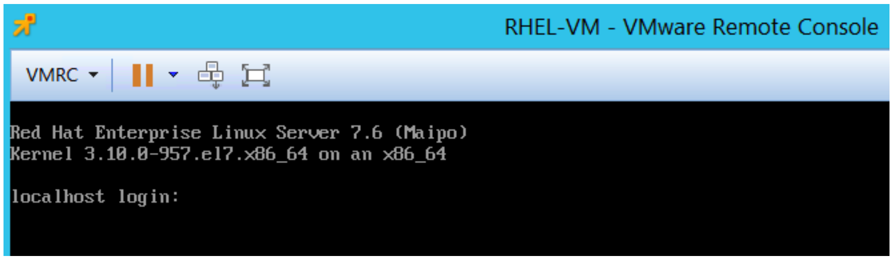
Local Standard Disks

80 GiB

VMware Virtual disk

sda / 80 GiB free

9. Leave Automatically configure partitioning checked and Click Done.
10. Click Begin Installation to start RHEL installation.
11. Enter and confirm the root password and click Done.
12. (Optional) Create another user for accessing the system.
13. After the installation is complete, click VMRC -> Removable Devices -> CD/DVD Drive 1 -> Disconnect <iso-file-name>
14. Click Reboot to reboot the system. The system should now boot up with RHEL.



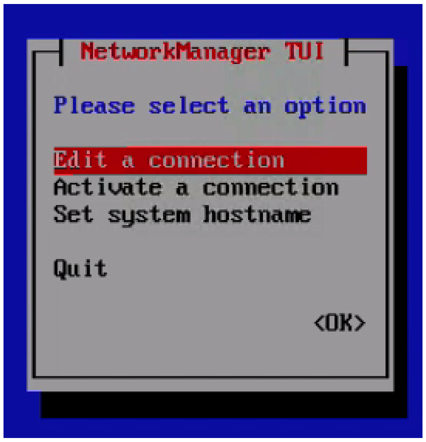
```
Red Hat Enterprise Linux Server 7.6 (Maipo)
Kernel 3.10.0-957.el7.x86_64 on an x86_64

localhost login:
```

If the VM does not reboot properly and seems to hang, click the VMRC button on top left and select Power -> Restart Guest.

Network and Hostname Setup

Adding a management network for each VM is necessary for remotely logging in and managing the VM. During this configuration step, all the network interfaces and the hostname will be setup using the VMware Remote Console. To setup the network and hostname, follow these steps:

1. Log into the RHEL using the VMware Remote Console and make sure the VM has finished rebooting and login prompt is visible.
2. Log in as `root`, enter the password set during the initial setup.
3. After logging in, type `nmtui` and press <Return>.
4. Using arrow keys, select Edit a connection and press <Return>.



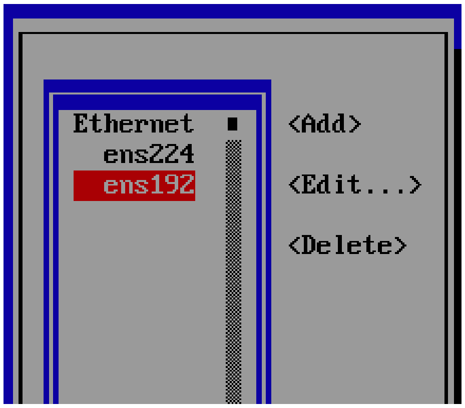
NetworkManager TUI

Please select an option

Edit a connection
Activate a connection
Set system hostname

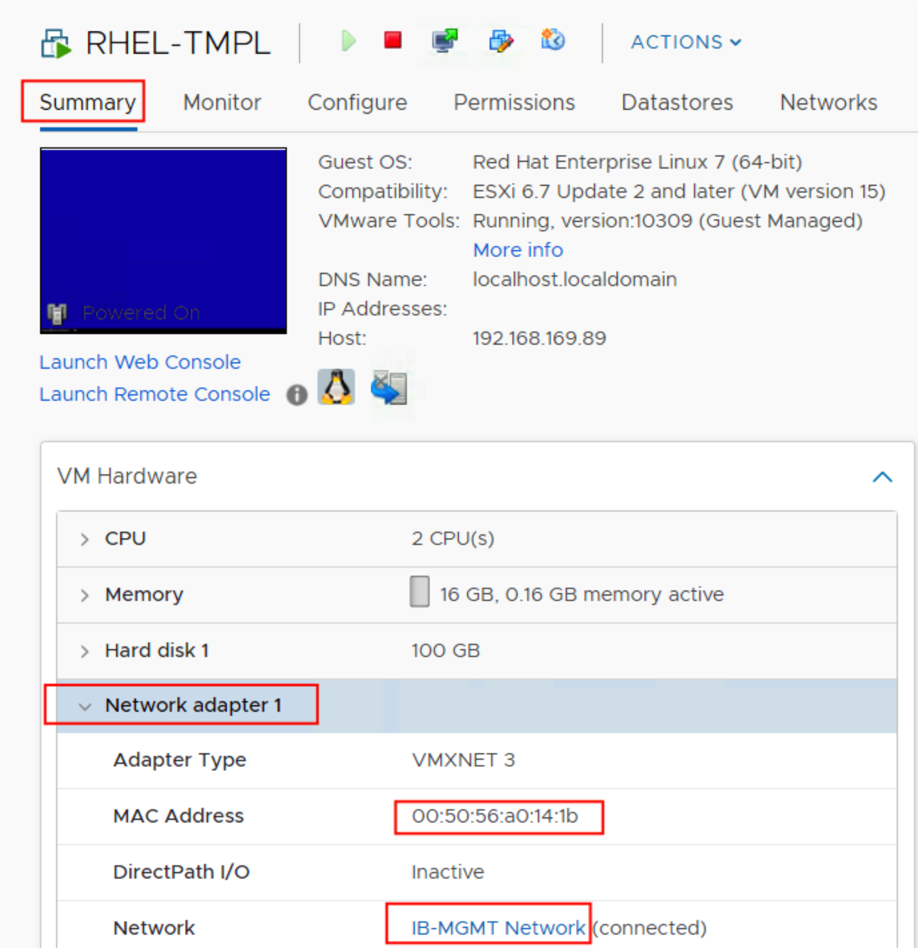
Quit

5. In the connection list, Select the connection with the lowest ID (ens192 in this example) and press <Return>.



```
Ethernet
  ens224
  ens192
```

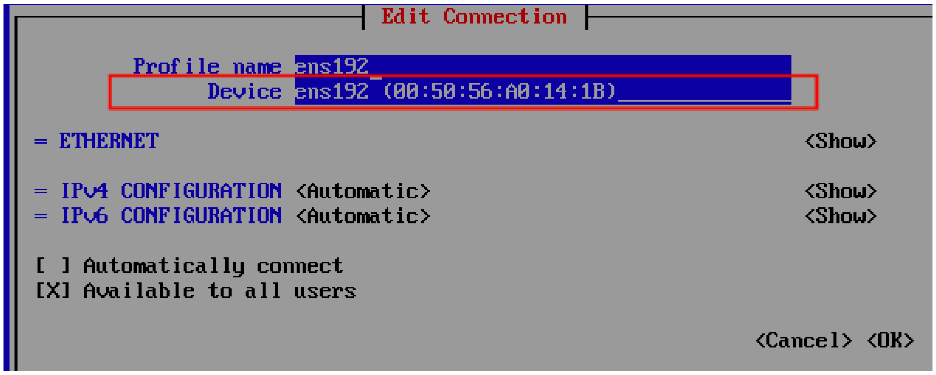
6. When setting up the VM, the first interface should have been assigned to the management port-group. This can be verified by going to vSphere vCenter and clicking on the VM. Under Summary -> VM Hardware, expand Network Adapter 1 and verify the MAC address and Network information.



Guest OS: Red Hat Enterprise Linux 7 (64-bit)
Compatibility: ESXi 6.7 Update 2 and later (VM version 15)
VMware Tools: Running, version:10309 (Guest Managed)
DNS Name: localhost.localdomain
IP Addresses:
Host: 192.168.169.89

CPU	2 CPU(s)
Memory	16 GB, 0.16 GB memory active
Hard disk 1	100 GB
Network adapter 1	
Adapter Type	VMXNET 3
MAC Address	00:50:56:a0:14:1b
DirectPath I/O	Inactive
Network	IB-MGMT Network (connected)

7. This MAC address should match the MAC address information in the VMware Remote Console.



```
                    Edit Connection

        Profile name ens192
              Device ens192 (00:50:56:A0:14:1B)

= ETHERNET                                              <Show>

= IPv4 CONFIGURATION <Automatic>                        <Show>
= IPv6 CONFIGURATION <Automatic>                        <Show>

[ ] Automatically connect
[X] Available to all users

                                                   <Cancel> <OK>
```

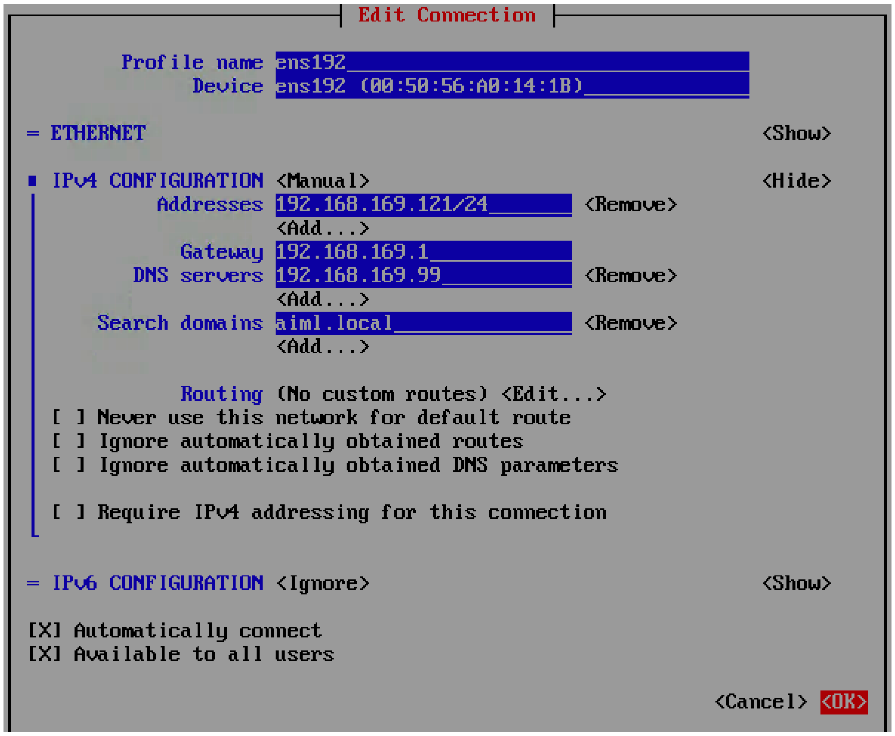
8. After the interface is correctly identified, in the Remote Console, using arrow keys scroll down to IPv4 CONFIGURATION <Automatic> and press <Return>. Select Manual.

9. Scroll to <Show> next to IPv4 CONFIGURATION and press <Return>.

10. Scroll to <Add…> next to Addresses and enter the management IP address with a subnet mask in the following format: x.x.x.x/nn (e.g. 192.168.169.121/24)

Remember to enter a subnet mask when entering the IP address. The system will accept an IP address without a subnet mask and then assign a subnet mask of /32 causing connectivity issues.

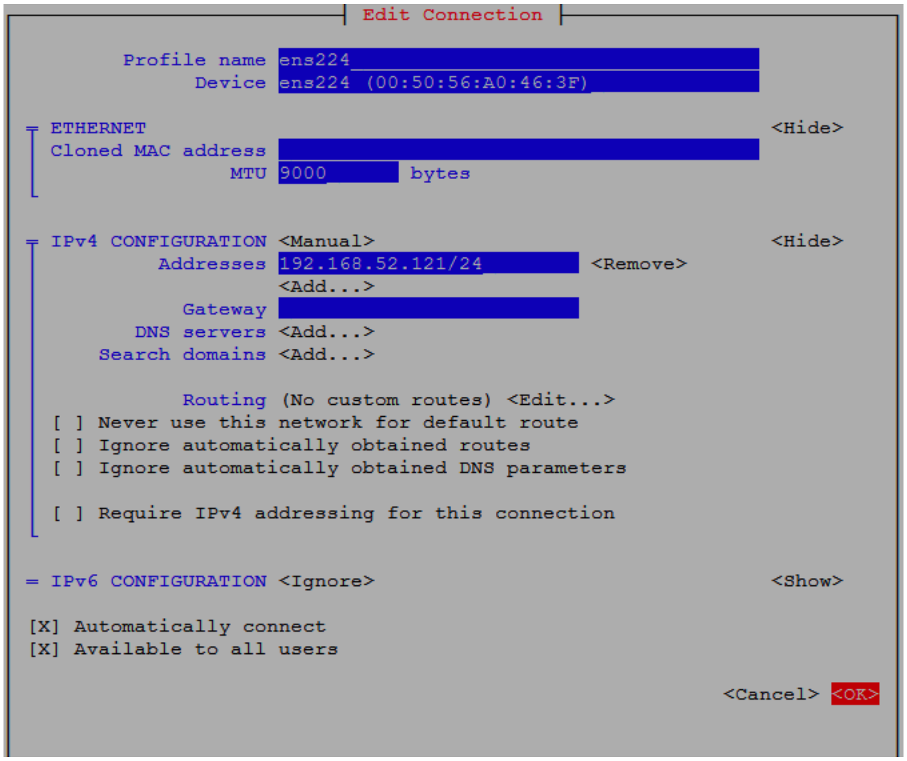
11. Scroll down to Gateway and enter the gateway IP address.
12. Scroll down to <Add..> next to DNS server and add one or more DNS servers.
13. Scroll down to <Add…> next to Search Domains and add a domain (if applicable).
14. Scroll down to <Automatic> next to IPv6 CONFIGURATION and press <Return>.
15. Select Ignore and press <Return>.
16. Scroll down and Check Automatically connect
17. Scroll down to <OK> and press <Return>



```
                         Edit Connection

        Profile name ens192_______________________________________
              Device ens192 (00:50:56:A0:14:1B)_____________________

 = ETHERNET                                                   <Show>

 ▪ IPv4 CONFIGURATION <Manual>                                <Hide>
 |          Addresses 192.168.169.121/24_______ <Remove>
 |                    <Add...>
 |            Gateway 192.168.169.1____________
 |        DNS servers 192.168.169.99___________ <Remove>
 |                    <Add...>
 |     Search domains aiml.local_______________ <Remove>
 |                    <Add...>
 |
 |            Routing (No custom routes) <Edit...>
 | [ ] Never use this network for default route
 | [ ] Ignore automatically obtained routes
 | [ ] Ignore automatically obtained DNS parameters
 |
 | [ ] Require IPv4 addressing for this connection
 └

 = IPv6 CONFIGURATION <Ignore>                                <Show>

 [X] Automatically connect
 [X] Available to all users

                                                       <Cancel> <OK>
```

18. Repeat this procedure to setup NFS* interface.

* For the NFS interface, expand the Ethernet settings by selecting Show and set the MTU to 9000. Do not set a Gateway.



```
                          Edit Connection

         Profile name ens224
               Device ens224  (00:50:56:A0:46:3F)

 ═ ETHERNET                                                   <Hide>
 │ Cloned MAC address
 │               MTU 9000          bytes
 └

 ═ IPv4 CONFIGURATION <Manual>                                <Hide>
 │          Addresses 192.168.52.121/24       <Remove>
 │                    <Add...>
 │            Gateway
 │        DNS servers <Add...>
 │     Search domains <Add...>
 │
 │            Routing (No custom routes) <Edit...>
 │ [ ] Never use this network for default route
 │ [ ] Ignore automatically obtained routes
 │ [ ] Ignore automatically obtained DNS parameters
 │
 │ [ ] Require IPv4 addressing for this connection
 └

 ═ IPv6 CONFIGURATION <Ignore>                                <Show>

 [X] Automatically connect
 [X] Available to all users

                                                        <Cancel> <OK>
```

19. Scroll down to <Back> and press <Return>.

20. From the main Network Manager TUI screen, scroll down to Set system hostname and press <Return>.

21. Enter the fully qualified domain name for the server and press <Return>.

22. Press <Return> and scroll down to Quit and press <Return> again.

23. At this point, the network services can be restarted for these changes to take effect. In the lab setup, the VM was rebooted (type `reboot` and press <Return>) to ensure all the changes were properly saved and applied across the future server reboots.

RHEL VM – Base Configuration

In this step, the following items will be configured on the RHEL host:

- Setup Subscription Manager
- Enable repositories
- Install Net-Tools
- Install FTP
- Enable EPEL Repository
- Install NFS utilities and mount NFS share
- Update ENIC drivers
- Setup NTP

- Disable Firewall
- Install Kernel Headers
- Install gcc
- Install wget
- Install DKMS

Log into RHEL Host using SSH

To log in to the host(s), use an SSH client and connect to the previously configured management IP address of the host. Use the username: `root` and the <password> set up during RHEL installation.

Setup Subscription Manager

To setup the subscription manager, follow these steps:

1. To download and install packages, setup the subscription manager using valid redhat.com credentials:

```
[root@ rhel-tmpl~]# subscription-manager register --username= <Name> --password=<Password> --auto-attach
Registering to: subscription.rhsm.redhat.com:443/subscription
The system has been registered with ID: <***>
The registered system name is: rhel-tmpl.aiml.local
```

2. To verify the subscription status:

```
[root@ rhel-tmpl~]# subscription-manager attach --auto
Installed Product Current Status:
Product Name: Red Hat Enterprise Linux Server
Status:       Subscribed
```

Enable Repositories

To setup repositories for downloading various software packages, run the following command:

```
[root@ rhel-tmpl~]# subscription-manager repos --enable="rhel-7-server-rpms" --enable="rhel-7-server-
extras-rpms"
Repository 'rhel-7-server-rpms' is enabled for this system.
Repository 'rhel-7-server-extras-rpms' is enabled for this system.
```

Install Net-Tools and Verify MTU

To enable helpful network commands (including ifconfig), install net-tools:

```
[root@rhel-tmpl ~] yum install net-tools
Loaded plugins: product-id, search-disabled-repos, subscription-manager

<SNIP>

Installed:
  net-tools.x86_64 0:2.0-0.25.20131004git.el7

Complete!
```

Using the `ifconfig` command, verify the MTU is correctly set to 9000 on the NFS interface. If the MTU is not set correctly, modify the MTU and set it to 9000 (using nmtui).

Install FTP

Install the FTP client to enable copying files to the host using ftp:

```
[root@rhel-tmpl ~]# yum install ftp
Loaded plugins: product-id, search-disabled-repos, subscription-manager
epel/x86_64/metalink                                                            |  17 kB  00:00:00

<SNIP>

Installed:
  ftp.x86_64 0:0.17-67.el7

Complete!
```

Enable EPEL Repository

EPEL (Extra Packages for Enterprise Linux) is open source and free community-based repository project from Fedora team which provides 100 percent high quality add-on software packages for Linux distribution including RHEL. Some of the packages installed later in the setup require EPEL repository to be enabled. To enable the repository, run the following:

```
[root@rhel-tmpl ~]# yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
Loaded plugins: product-id, search-disabled-repos, subscription-manager
epel-release-latest-7.noarch.rpm                                                      |  15 kB  00:00:00
Examining /var/tmp/yum-root-Gfcqhh/epel-release-latest-7.noarch.rpm: epel-release-7-12.noarch
Marking /var/tmp/yum-root-Gfcqhh/epel-release-latest-7.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-12 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=========================================================================================================
 Package                   Arch                Version            Repository                        Size
=========================================================================================================
Installing:
 epel-release              noarch              7-12               /epel-release-latest-7.noarch      24 k

Transaction Summary
=========================================================================================================
Install  1 Package

Total size: 24 k
Installed size: 24 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : epel-release-7-12.noarch                                                               1/1
  Verifying  : epel-release-7-12.noarch                                                               1/1

Installed:
  epel-release.noarch 0:7-12
```

Install NFS Utilities and Mount NFS Share

To mount NFS share on the host, NFS utilities need to be installed and the `/etc/fstab` file needs to be modified. To do so, follow these steps:

1. To install the `nfs-utils`:

```
[root@rhel-tmpl ~]# yum install nfs-utils
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies

<SNIP>

Installed:
  nfs-utils.x86_64 1:1.3.0-0.65.el7

Dependency Installed:
  gssproxy.x86_64 0:0.7.0-26.el7      keyutils.x86_64 0:1.5.8-3.el7          libbasicobjects.x86_64
0:0.1.1-32.el7
  libcollection.x86_64 0:0.7.0-32.el7 libevent.x86_64 0:2.0.21-4.el7         libini_config.x86_64
0:1.3.1-32.el7
  libnfsidmap.x86_64 0:0.25-19.el7    libpath_utils.x86_64 0:0.2.1-32.el7    libref_array.x86_64 0:0.1.5-
32.el7
  libtirpc.x86_64 0:0.2.4-0.16.el7    libverto-libevent.x86_64 0:0.2.5-4.el7 quota.x86_64 1:4.01-19.el7
  quota-nls.noarch 1:4.01-19.el7      rpcbind.x86_64 0:0.2.0-48.el7          tcp_wrappers.x86_64 0:7.6-
77.el7

Complete!
```

2. Using text editor (such as vi), add the following line at the end of the `/etc/fstab` file:

```
<IP Address of NFS LIF>:/imagenet_dataset /mnt/imagenet nfs
auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo =1800 0 0
```

where the `/imagenet_dataset` is the NFS mount point (as defined in NetApp).

To obtain the correct storage system mount point information, issue the following command: `volume show -vserver AI-ML-SVM -volume <imagenet_dataset>` **and note down the** `Junction Path`**.**

3. Verify that the updated `/etc/fstab` file looks like:

```
#
# /etc/fstab
# Created by anaconda on Wed Mar 27 18:33:36 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/rhel01-root /                       xfs     defaults,_netdev,_netdev 0 0
UUID=36f667cf-xxxxxxxxx /boot                   xfs     defaults,_netdev,_netdev,x-initrd.mount 0 0
/dev/mapper/rhel01-home /home                   xfs     defaults,_netdev,_netdev,x-initrd.mount 0 0
/dev/mapper/rhel01-swap swap                    swap    defaults,_netdev,x-initrd.mount 0 0
192.168.52.251:/imagenet_dataset /mnt/imagenet nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp 0 0
```

4. Issue the following commands to mount NFS at the following location: `/mnt/imagenet`

```
[root@rhel-tmpl ~]# mkdir /mnt/imagenet
[root@rhel-tmpl ~]# mount /mnt/imagenet
```

5. To verify that the mount was successful:

```
[root@rhel-tmpl ~]# mount | grep imagenet
```

```
192.168.52.251:/imagenet_dataset on /mnt/imagenet type nfs
(rw,noatime,vers=3,rsize=65536,wsize=65536,namlen=255,hard,nolock,proto=tcp,timeo=600,retrans=2,sec=sys,m
ountaddr=192.168.52.251,mountvers=3,mountport=635,mountproto=tcp,local_lock=all,addr=192.168.52.251)
```

Setup NTP

To setup NTP, follow these steps:

1. To synchronize the host time to an NTP server, install NTP package:

```
[root@rhel-tmpl ~]# yum install  ntp
<SNIP>
Installed:
  ntp.x86_64 0:4.2.6p5-29.el7

Dependency Installed:
  autogen-libopts.x86_64 0:5.18-5.el7                        ntpdate.x86_64 0:4.2.6p5-29.el7
```

2. If the default NTP servers defined in `/etc/ntp.conf` file are not reachable or to add additional local NTP servers, modify the `/etc/ntp.conf` file (using a text editor such as vi) and add the server(s) as shown below:

“#” in front of a server name or IP address signifies that the server information is commented out and will not be used

```
[root@rhel-tmpl~]# more /etc/ntp.conf | grep server
server 192.168.169.1 iburst
# server 0.rhel.pool.ntp.org iburst
# server 1.rhel.pool.ntp.org iburst
# server 2.rhel.pool.ntp.org iburst
# server 3.rhel.pool.ntp.org iburst
```

3. To verify the time is setup correctly, use the **`date`** command:

Disable Firewall

To make sure the installation goes smoothly, Linux firewall and the Linux kernel security module (SEliniux) is disabled. To do so, follow these steps:

Customer Linux server management team should review and enable these security modules with appropriate settings once the installation is complete.

1. To disable Firewall:

```
[root@rhel-tmpl ~]# systemctl stop firewalld
[root@rhel-tmpl ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multiuser.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
```

2. To disable SELinux:

```
[root@rhel-tmpl ~]# sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
[root@rhel-tmpl ~]# setenforce 0
```

3. Reboot the host:

```
[root@rhel-tmpl l~]# reboot
```

Disable IPv6 (Optional)

If IPv6 addresses are not being used in the customer environment, IPv6 can be disabled on the RHEL host:

```
[root@rhel-tmpl ~]# echo 'net.ipv6.conf.all.disable_ipv6 = 1' >> /etc/sysctl.conf
[root@rhel-tmpl ~]# echo 'net.ipv6.conf.default.disable_ipv6 = 1' >> /etc/sysctl.conf
[root@rhel-tmpl ~]# echo 'net.ipv6.conf.lo.disable_ipv6 = 1' >> /etc/sysctl.conf
[root@rhel-tmpl ~]# reboot
```

Install Kernel Headers

To install the Kernel Headers, run the following commands:

```
[root@rhel-tmpl~]# uname -r
3.10.0-957.el7.x86_64
[root@rhel-tmpl~]# yum install kernel-devel-$(uname -r) kernel-headers-$(uname -r)
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies

<SNIP>

Installed:
  kernel-devel.x86_64 0:3.10.0-957.el7                    kernel-headers.x86_64 0:3.10.0-957.el7

Dependency Installed:
  perl.x86_64 4:5.16.3-294.el7_6                             perl-Carp.noarch 0:1.26-244.el7
  perl-Encode.x86_64 0:2.51-7.el7                            perl-Exporter.noarch 0:5.68-3.el7
  perl-File-Path.noarch 0:2.09-2.el7                         perl-File-Temp.noarch 0:0.23.01-3.el7
  perl-Filter.x86_64 0:1.49-3.el7                            perl-Getopt-Long.noarch 0:2.40-3.el7
  perl-HTTP-Tiny.noarch 0:0.033-3.el7                        perl-PathTools.x86_64 0:3.40-5.el7
  perl-Pod-Escapes.noarch 1:1.04-294.el7_6                   perl-Pod-Perldoc.noarch 0:3.20-4.el7
  perl-Pod-Simple.noarch 1:3.28-4.el7                        perl-Pod-Usage.noarch 0:1.63-3.el7
  perl-Scalar-List-Utils.x86_64 0:1.27-248.el7               perl-Socket.x86_64 0:2.010-4.el7
  perl-Storable.x86_64 0:2.45-3.el7                          perl-Text-ParseWords.noarch 0:3.29-4.el7
  perl-Time-HiRes.x86_64 4:1.9725-3.el7                      perl-Time-Local.noarch 0:1.2300-2.el7
  perl-constant.noarch 0:1.27-2.el7                          perl-libs.x86_64 4:5.16.3-294.el7_6
  perl-macros.x86_64 4:5.16.3-294.el7_6                      perl-parent.noarch 1:0.225-244.el7
  perl-podlators.noarch 0:2.5.1-3.el7                        perl-threads.x86_64 0:1.87-4.el7
  perl-threads-shared.x86_64 0:1.43-6.el7

Complete!
```

Install gcc

To install the C compiler, run the following commands:

```
 [root@rhel-tmpl ~]# yum install gcc-4.8.5

<SNIP>

Installed:
  gcc.x86_64 0:4.8.5-39.el7

Dependency Installed:
  cpp.x86_64 0:4.8.5-39.el7        glibc-devel.x86_64 0:2.17-292.el7      glibc-headers.x86_64 0:2.17-
292.el7
  libmpc.x86_64 0:1.0.1-3.el7      mpfr.x86_64 0:3.1.1-4.el7

Dependency Updated:
  glibc.x86_64 0:2.17-292.el7         glibc-common.x86_64 0:2.17-292.el7       libgcc.x86_64 0:4.8.5-
39.el7
  libgomp.x86_64 0:4.8.5-39.el7

Complete!
```

```
[root@rhel-tmpl ~]# yum install gcc-c++
Loaded plugins: product-id, search-disabled-repos, subscription-manager

<SNIP>
```

```

Installed:
  gcc-c++.x86_64 0:4.8.5-39.el7

Dependency Installed:
  libstdc++-devel.x86_64 0:4.8.5-39.el7

Dependency Updated:
  libstdc++.x86_64 0:4.8.5-39.el7

Complete!
```

Install wget

To install wget for downloading files from Internet, run the following command:

```
[root@rhel-tmpl ~]# yum install wget
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies

<SNIP>

Installed:
  wget.x86_64 0:1.14-18.el7_6.1
```

Install DKMS

To enable Dynamic Kernel Module Support, run the following command:

```
[root@rhel-tmpl ~]# yum install dkms
Loaded plugins: product-id, search-disabled-repos, subscription-manager
epel/x86_64/metalink                                                            |  17 kB  00:00:00

<SNIP>

Installed:
  dkms.noarch 0:2.7.1-1.el7

Dependency Installed:
  elfutils-libelf-devel.x86_64 0:0.176-2.el7                    zlib-devel.x86_64 0:1.2.7-18.el7

Dependency Updated:
  elfutils-libelf.x86_64 0:0.176-2.el7                      elfutils-libs.x86_64 0:0.176-2.el7

Complete!
```

A VM template can be created at this time for cloning any future VMs. NVIDIA driver installation is GPU specific and if you have a mixed GPU environment, NVIDIA driver installation will have a dependency on GPU model.

NVIDIA and CUDA Drivers Installation

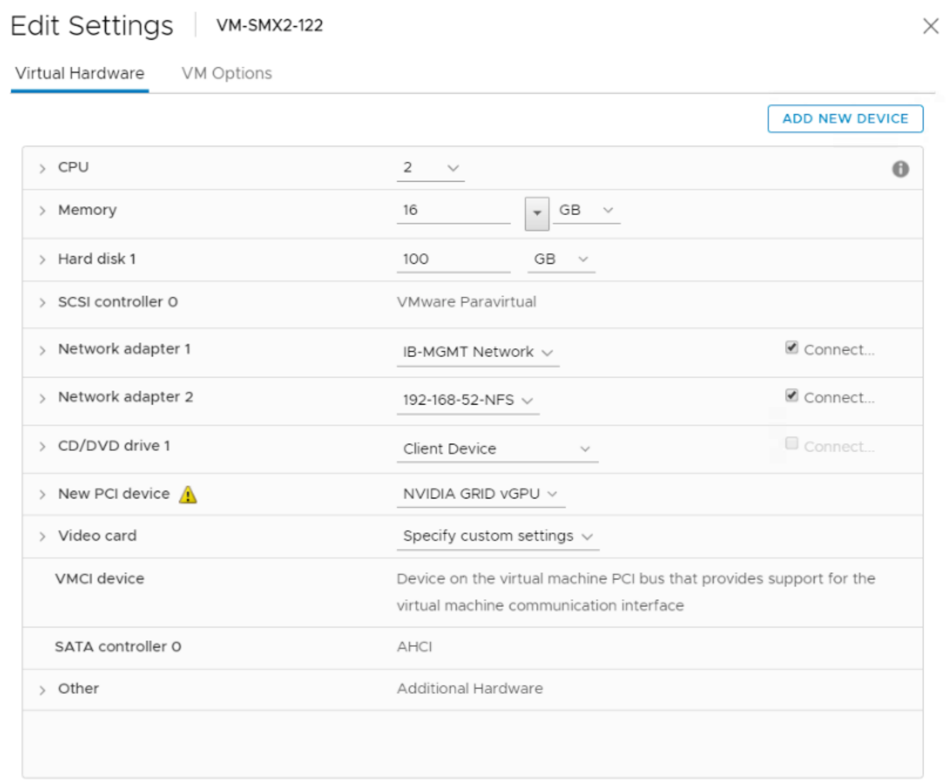
In this step, the following components will be installed:

- Add vGPU to the VM
- Install NVIDIA Driver
- Install CUDA Toolkit

Add vGPU to the VM

To add one or more vGPUs to the VM, follow these steps:

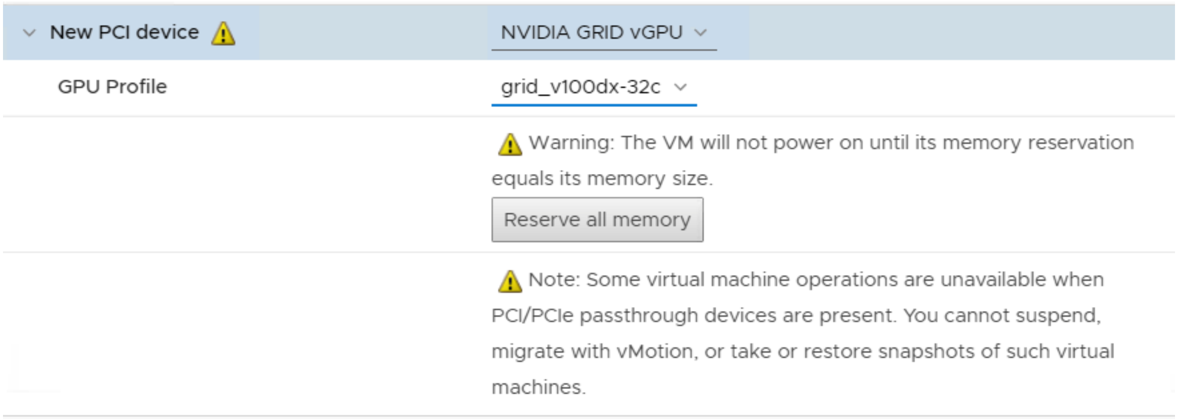
1. In the vSphere client, make sure the VM is shutdown. If not, shutdown the VM using VM console.
2. Right-click the VM and select Edit Settings…
3. Click ADD NEW DEVICE and select Shared PCI Device. Make sure NVIDIA GRID vGPU is shown for New PCI Device.



Edit Settings | VM-SMX2-122

CPU	2	
Memory	16 GB	
Hard disk 1	100 GB	
SCSI controller 0	VMware Paravirtual	
Network adapter 1	IB-MGMT Network	☑ Connect...
Network adapter 2	192-168-52-NFS	☑ Connect...
CD/DVD drive 1	Client Device	□ Connect...
New PCI device ⚠	NVIDIA GRID vGPU	
Video card	Specify custom settings	
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface	
SATA controller 0	AHCI	
Other	Additional Hardware	

4. Click the arrow next to New PCI Device and select a GPU profile. For various GPU profile options, refer to the NVIDIA documentation.



New PCI device ⚠	NVIDIA GRID vGPU
GPU Profile	grid_v100dx-32c

⚠ Warning: The VM will not power on until its memory reservation equals its memory size.

⚠ Note: Some virtual machine operations are unavailable when PCI/PCIe passthrough devices are present. You cannot suspend, migrate with vMotion, or take or restore snapshots of such virtual machines.

5. Click Reserve all memory.

Since all the VM memory is reserved, vSphere vCenter generates memory usage alarms. These alarms can be ignored or disabled as described in the VMware documentation: https://kb.vmware.com/s/article/2149787

6. (Optional) Repeat the process to add more PCI devices (vGPU).
7. Click OK
8. Power On the VM.

If the VM compatibility is not set to vSphere 6.7 Update 2, only one GPU can be added to the VM.

Install NVIDIA Driver

To install NVIDIA Driver on the RHEL VM, follow these steps:

1. From the previously downloaded zip file NVIDIA-GRID-vSphere-6.7-430.46-431.79.zip, extract the LINUX driver file NVIDIA-Linux-x86_64-430.46-grid.run.
2. Copy the file to the VM using FTP or sFTP.
3. Install the driver by running the following command:

```
[root@rhel-tmpl ~]# sh NVIDIA-Linux-x86_64-430.46-grid.run
```

4. For “Would you like to register the kernel module sources with DKMS? This will allow DKMS to automatically build a new module, if you install a different kernel later.”, select Yes.
5. Select OK for the X library path warning
6. (Optional) For “Install NVIDIA's 32-bit compatibility libraries?”, select Yes if 32-bit libraries are needed.
7. Select OK when the installation is complete.
8. Verify the correct vGPU profile is reported using the following command:

```
[root@rhel-tmpl ~]# nvidia-smi --query-gpu=gpu_name --format=csv,noheader --id=0 | sed -e 's/ /-/g'
GRID-V100DX-32C
```

9. Blacklist the Nouveau Driver by opening the `/etc/modprobe.d/blacklist-nouveau.conf` in a text editor (for example vi) and adding following commands:

```
blacklist nouveau
options nouveau modeset=0
```

10. Verify the contents of the file. If the file does not exist, create the file and add the configuration lines.

```
 [root@rhel-tmpl ~]# more /etc/modprobe.d/blacklist-nouveau.conf
# RPM Fusion blacklist for nouveau driver - you need to run as root:
# dracut -f /boot/initramfs-$(uname -r).img $(uname -r)
# if nouveau is loaded despite this file.
blacklist nouveau
options nouveau modeset=0
```

11. Regenerate the kernel initramfs and reboot the system:

```
[root@c480ml~]# dracut --force
[root@c480ml~]# reboot
```

If the nouveau kernel module is not disabled, the NVIDIA kernel module will not load properly

12. After the reboot, verify the NVIDIA vGPUs are reported correctly:

The output of nvidia-smi will be different depending on the difference in number and profile of vGPUs

```
 [root@rhel-tmpl ~]# nvidia-smi
Thu Oct 17 17:28:46 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 430.46       Driver Version: 430.46       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GRID V100DX-32C     On   | 00000000:02:02.0 Off |                    0 |
| N/A   N/A    P0    N/A /  N/A |   2064MiB / 32638MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  GRID V100DX-32C     On   | 00000000:02:03.0 Off |                    0 |
| N/A   N/A    P0    N/A /  N/A |   2064MiB / 32638MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Install CUDA Toolkit

To install the CUDA toolkit, follow these steps:

1. Download CUDA driver version 10.1 Update 2 from NVIDIA website using wget:

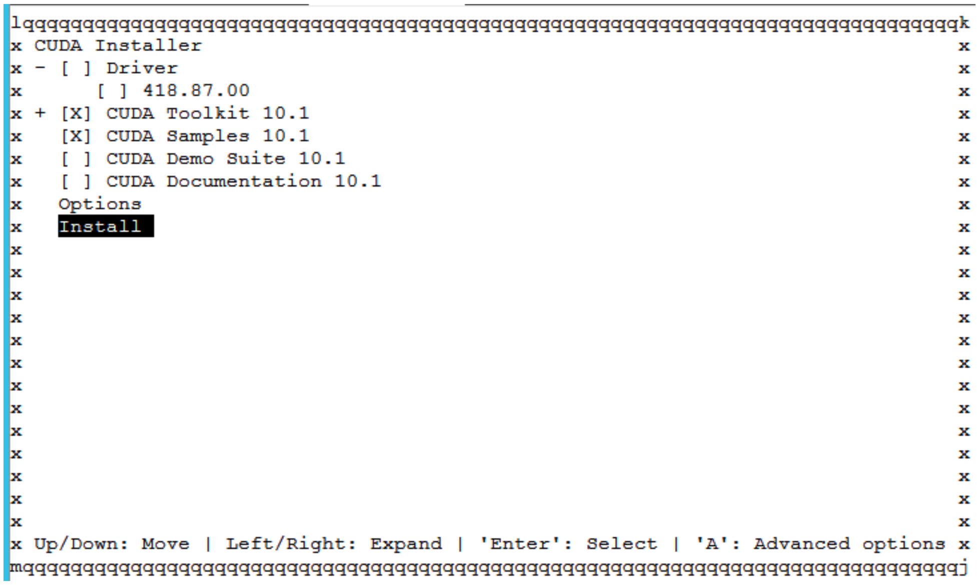
```
[root@rhel-tmpl ~]# wget
http://developer.download.nvidia.com/compute/cuda/10.1/Prod/local_installers/cuda_10.1.243_418.87.00_linu
x.run
--2019-09-18 16:23:05--  Resolving developer.download.nvidia.com (developer.download.nvidia.com)...
192.229.211.70, 2606:2800:21f:3aa:dcf:37b:1ed6:1fb
Connecting to developer.download.nvidia.com (developer.download.nvidia.com)|192.229.211.70|:80...
connected.
HTTP request sent, awaiting response... 200 OK

<SNIP>

2019-09-18 16:23:46 (69.1 MB/s) - ‘cuda-repo-rhel7-10-1-local-10.1.243-418.87.00-1.0-1.x86_64.rpm’ saved
[2660351598/2660351598]
```

Preserve the previously installed driver version 430.46 when installing CUDA toolkit.

2. Install the CUDA 10.1 Tool Kit without updating the NVIDIA driver:

```
[root@rhel-tmpl~]# sh cuda_10.1.243_418.87.00_linux.run
```

3. From the text menu, using arrow keys, select Continue and press Enter.
4. Type accept to accept the end user license agreement and press Enter.
5. Using arrow keys and space bar, deselect Driver.
6. Optionally, deselect CUDA Demo Suite 10.1 and CUDA Documentation.



```
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x CUDA Installer                                                             x
x - [ ] Driver                                                               x
x      [ ] 418.87.00                                                         x
x + [X] CUDA Toolkit 10.1                                                    x
x   [X] CUDA Samples 10.1                                                    x
x   [ ] CUDA Demo Suite 10.1                                                 x
x   [ ] CUDA Documentation 10.1                                              x
x   Options                                                                  x
x   Install                                                                  x
x                                                                            x
x                                                                            x
x                                                                            x
x                                                                            x
x                                                                            x
x                                                                            x
x                                                                            x
x                                                                            x
x                                                                            x
x                                                                            x
x                                                                            x
x                                                                            x
x                                                                            x
x Up/Down: Move | Left/Right: Expand | 'Enter': Select | 'A': Advanced options x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

7. Select Install and press Enter.

```
===========
= Summary =
===========

Driver:   Not Selected
Toolkit:  Installed in /usr/local/cuda-10.1/
Samples:  Installed in /root/

Please make sure that
 -   PATH includes /usr/local/cuda-10.1/bin
 -   LD_LIBRARY_PATH includes /usr/local/cuda-10.1/lib64, or, add /usr/local/cuda-10.1/lib64 to
/etc/ld.so.conf and run ldconfig as root

To uninstall the CUDA Toolkit, run cuda-uninstaller in /usr/local/cuda-10.1/bin

Please see CUDA_Installation_Guide_Linux.pdf in /usr/local/cuda-10.1/doc/pdf for detailed information on
setting up CUDA.
***WARNING: Incomplete installation! This installation did not install the CUDA Driver. A driver of
version at least 418.00 is required for CUDA 10.1 functionality to work.
To install the driver using this installer, run the following command, replacing <CudaInstaller> with the
name of this run file:
    sudo <CudaInstaller>.run --silent --driver

Logfile is /var/log/cuda-installer.log
```

8. Reboot the server (reboot).

9. Modify path variables by typing the following lines at the shell prompt and adding them to `.bashrc`:

```
export PATH=/usr/local/cuda-10.1/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-10.1/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

10. Verify the PATH variables:

```
[root@rhel-tmpl~]# echo $PATH
/usr/local/cuda-10.1/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
[root@rhel-tmpl~]# echo $LD_LIBRARY_PATH
/usr/local/cuda-10.1/lib64
```

11. Verify the variables are defined in .bashrc:

```
[root@rhel-tmpl ~]# more .bashrc | grep PATH
export PATH=/usr/local/cuda-10.1/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-10.1/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

12. Add the following line to `/etc/ld.so.conf` file:

```
/usr/local/cuda-10.1/lib64
```

13. Verify the `/etc/ld.so.conf` file configuration:

```
[root@rhel-tmpl ~]# more /etc/ld.so.conf
include ld.so.conf.d/*.conf
/usr/local/cuda-10.1/lib64
```

14. Execute the following command:

```
[root@rhel-tmpl ~]# ldconfig
```

15. Verify that CUDA version is 10.1:

```
[root@rhel-tmpl ~]# cat /usr/local/cuda/version.txt
CUDA Version 10.1.243
```

Verify the NVIDIA and CUDA Installation

Use the various commands shown below to verify the system is properly setup with CUDA and NVIDIA drivers and the GPUs are correctly identified. These commands will show slightly different output for Cisco UCS C480 ML and Cisco UCS C240 servers.

Verify CUDA Driver

To verify the CUDA driver, run a device query as shown below:

```
[root@rhel-tmpl ~]# cd /usr/local/cuda-10.1/samples/1_Utilities/deviceQuery
[root@rhel-tmpl deviceQuery]# make

<SNIP>

```

```
[root@rhel-tmpl deviceQuery]# ./deviceQuery
./deviceQuery Starting...

 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 2 CUDA Capable device(s)

Device 0: "GRID V100DX-32C"
  CUDA Driver Version / Runtime Version          10.1 / 10.1
  CUDA Capability Major/Minor version number:    7.0
  Total amount of global memory:                 32638 MBytes (34223423488 bytes)
  (80) Multiprocessors, ( 64) CUDA Cores/MP:     5120 CUDA Cores
  GPU Max Clock rate:                            1530 MHz (1.53 GHz)
  Memory Clock rate:                             877 Mhz
  Memory Bus Width:                              4096-bit
  L2 Cache Size:                                 6291456 bytes

<SNIP>

> Peer access from GRID V100DX-32C (GPU0) -> GRID V100DX-32C (GPU1) : Yes
> Peer access from GRID V100DX-32C (GPU1) -> GRID V100DX-32C (GPU0) : Yes

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 10.1, CUDA Runtime Version = 10.1, NumDevs = 2
Result = PASS
```

Verify NVIDIA Driver

To verify the NVIDIA driver, follow these steps:

```
[root@rhel-tmpl~]# modinfo nvidia
filename:       /lib/modules/3.10.0-957.el7.x86_64/extra/nvidia.ko.xz
alias:          char-major-195-*
version:        430.46
supported:      external
license:        NVIDIA
retpoline:      Y
rhelversion:    7.6
srcversion:     60D33C4E3271024E4954DDE
alias:          pci:v000010DEd00000E00sv*sd*bc04sc80i00*
alias:          pci:v000010DEd*sv*sd*bc03sc02i00*
alias:          pci:v000010DEd*sv*sd*bc03sc00i00*
depends:        ipmi_msghandler
vermagic:       3.10.0-957.el7.x86_64 SMP mod_unload modversions
parm:           NvSwitchRegDwords:NvSwitch regkey (charp)
parm:           NVreg_Mobile:int
parm:           NVreg_ResmanDebugLevel:int
parm:           NVreg_RmLogonRC:int
parm:           NVreg_ModifyDeviceFiles:int
parm:           NVreg_DeviceFileUID:int
parm:           NVreg_DeviceFileGID:int
parm:           NVreg_DeviceFileMode:int
parm:           NVreg_InitializeSystemMemoryAllocations:int
parm:           NVreg_UsePageAttributeTable:int
parm:           NVreg_MapRegistersEarly:int
parm:           NVreg_RegisterForACPIEvents:int
parm:           NVreg_EnablePCIeGen3:int
parm:           NVreg_EnableMSI:int
parm:           NVreg_TCEBypassMode:int
parm:           NVreg_EnableStreamMemOPs:int
parm:           NVreg_EnableBacklightHandler:int
parm:           NVreg_RestrictProfilingToAdminUsers:int
parm:           NVreg_PreserveVideoMemoryAllocations:int
parm:           NVreg_DynamicPowerManagement:int
parm:           NVreg_EnableUserNUMAManagement:int
parm:           NVreg_MemoryPoolSize:int
parm:           NVreg_KMallocHeapMaxSize:int
parm:           NVreg_VMallocHeapMaxSize:int
parm:           NVreg_IgnoreMMIOCheck:int
parm:           NVreg_NvLinkDisable:int
parm:           NVreg_RegistryDwords:charp
parm:           NVreg_RegistryDwordsPerDevice:charp
parm:           NVreg_RmMsg:charp
parm:           NVreg_GpuBlacklist:charp
parm:           NVreg_TemporaryFilePath:charp
parm:           NVreg_AssignGpus:charp
```

```
[root@rhel-tmpl ~]# lspci | grep -i nvidia
02:02.0 3D controller: NVIDIA Corporation GV100GL [Tesla V100 SXM2 32GB] (rev a1)
02:03.0 3D controller: NVIDIA Corporation GV100GL [Tesla V100 SXM2 32GB] (rev a1)
```

Log into the ESXi server to see which physical GPU is assigned to the VM:

```
[root@ESXi-host:~] nvidia-smi
Thu Oct 17 23:13:20 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 430.46       Driver Version: 430.46       CUDA Version: N/A      |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:1B:00.0 Off |                    0 |
| N/A   40C    P0    48W / 300W |  32625MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  On   | 00000000:1C:00.0 Off |                    0 |
| N/A   40C    P0    45W / 300W |  32625MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-SXM2...  On   | 00000000:42:00.0 Off |                    0 |
| N/A   40C    P0    44W / 300W |     40MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-SXM2...  On   | 00000000:43:00.0 Off |                    0 |
| N/A   41C    P0    42W / 300W |     40MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   4  Tesla V100-SXM2...  On   | 00000000:89:00.0 Off |                    0 |
| N/A   40C    P0    45W / 300W |     40MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   5  Tesla V100-SXM2...  On   | 00000000:8A:00.0 Off |                    0 |
| N/A   40C    P0    45W / 300W |     40MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   6  Tesla V100-SXM2...  On   | 00000000:B2:00.0 Off |                    0 |
| N/A   38C    P0    44W / 300W |     40MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   7  Tesla V100-SXM2...  On   | 00000000:B3:00.0 Off |                    0 |
| N/A   38C    P0    45W / 300W |     40MiB / 32767MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0   2206637    C+G   rhel-tmpl                                  32574MiB |
|    1   2206637    C+G   rhel-tmpl                                  32574MiB |
+-----------------------------------------------------------------------------+
```

Setup NVIDIA vGPU Licensing on the VM

In order to obtain the license for vGPU usages, set up the licensing configuration using the NVIDIA documentation: https://docs.nvidia.com/grid/latest/grid-licensing-user-guide/index.html#licensing-grid-vgpu-linux-config-file . The following configuration steps allow the VM to obtain the NVIDIA vGPU license from the previously configured Software License Server:

1. On the Linux VM, log into the shell as root
2. Copy the /etc/nvidia/gridd.conf.template file to /etc/nvidia/gridd.conf file.

```
[root@rhel-tmpl ~]# cp /etc/nvidia/gridd.conf.template /etc/nvidia/gridd.conf
```

3. Edit the /etc/nvidia/gridd.conf file using text editor such as vi.
4. Enter the IP address of the previously configured License Server.

```
ServerAddress=192.168.169.10
```

5. If the ports were changed from the default values, enter the ServerPort value to the file.
6. If a backup server was setup for high availability, add the BackupServerAddress.
7. Set the FeatureType to 1 to license the vGPU.

```
FeatureType=1
```

There is no need to specify the type of the license. NVIDIA vGPU software automatically selects the correct type of license based on the vGPU type.

8. Save the configuration file (overwrite the file if necessary).

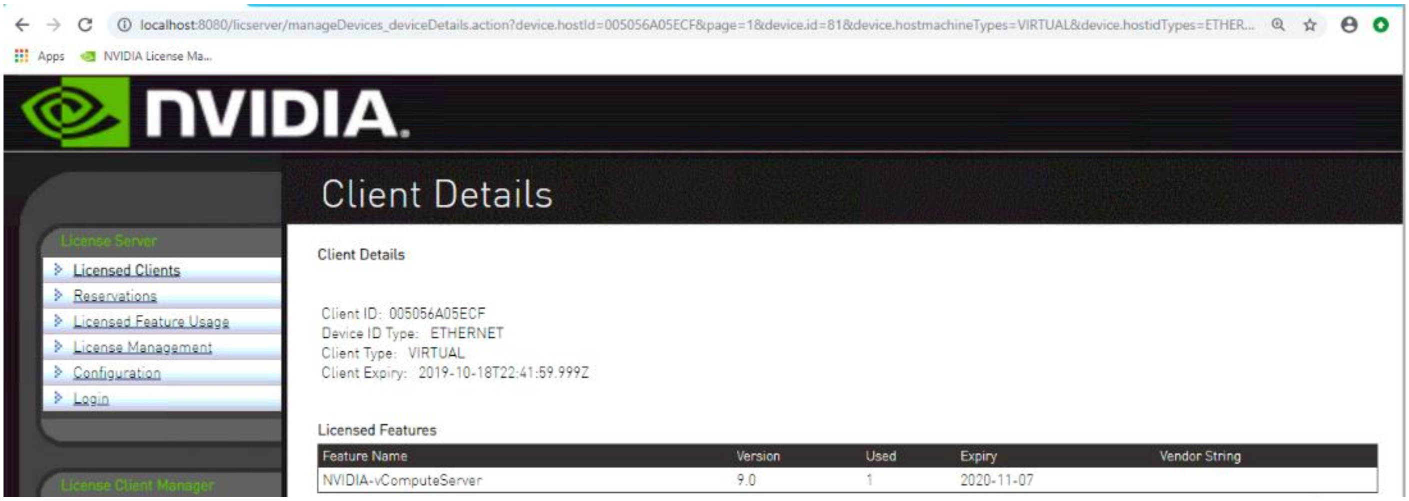
9. Restart the nvidia-gridd service:

```
[root@rhel-tmpl ~]# service nvidia-gridd restart
[root@rhel-tmpl ~]#
```

10. Verify service obtained correct address:

```
[root@rhel-tmpl ~]# grep gridd /var/log/messages
<SNIP>
Oct 17 18:41:17 VM-SMX2-122 nvidia-gridd: serverUrl is NULL
Oct 17 18:41:17 VM-SMX2-122 nvidia-gridd: Calling load_byte_array(tra)
Oct 17 18:41:18 VM-SMX2-122 nvidia-gridd: serverUrl is NULL
Oct 17 18:41:18 VM-SMX2-122 nvidia-gridd: Shutdown (6070)
Oct 17 18:41:18 VM-SMX2-122 nvidia-gridd: Started (7275)
Oct 17 18:41:19 VM-SMX2-122 nvidia-gridd: Ignore service provider licensing
Oct 17 18:41:20 VM-SMX2-122 nvidia-gridd: Service provider detection complete.
Oct 17 18:41:20 VM-SMX2-122 nvidia-gridd: Calling load_byte_array(tra)
Oct 17 18:41:21 VM-SMX2-122 nvidia-gridd: Acquiring license for GRID vGPU Edition.
Oct 17 18:41:21 VM-SMX2-122 nvidia-gridd: Calling load_byte_array(tra)
Oct 17 18:41:22 VM-SMX2-122 nvidia-gridd: License acquired successfully. (Info:
http://192.168.169.10:7070/request; NVIDIA-vComputeServer,9.0)
```

11. On the license server management interface, click Licensed Clients and click the MAC address of the VM to view the license:



localhost:8080/licserver/manageDevices_deviceDetails.action?device.hostId=005056A05ECF&page=1&device.id=81&device.hostmachineTypes=VIRTUAL&device.hostidTypes=ETHER...

Client Details

Client Details

Client ID: 005056A05ECF
Device ID Type: ETHERNET
Client Type: VIRTUAL
Client Expiry: 2019-10-18T22:41:59.999Z

Licensed Features

Feature Name	Version	Used	Expiry	Vendor String
NVIDIA-vComputeServer	9.0	1	2020-11-07	

Cisco UCS Configuration for Bare Metal Workload

Cisco UCS Base Configuration

For the base configuration for the Cisco UCS 6454 Fabric Interconnect, follow the Cisco UCS Configuration section here: FlexPod Datacenter with VMware vSphere 6.7 U1, Cisco UCS 4th Generation and NetApp AFF A-Series. This FlexPod deployment guide explains the necessary configuration steps required for deploying Cisco UCS C-Series servers for bare-metal Red Hat Enterprise Linux (RHEL) installation. These configuration sections assumes following items have been pre-configured using the CVD referenced above:

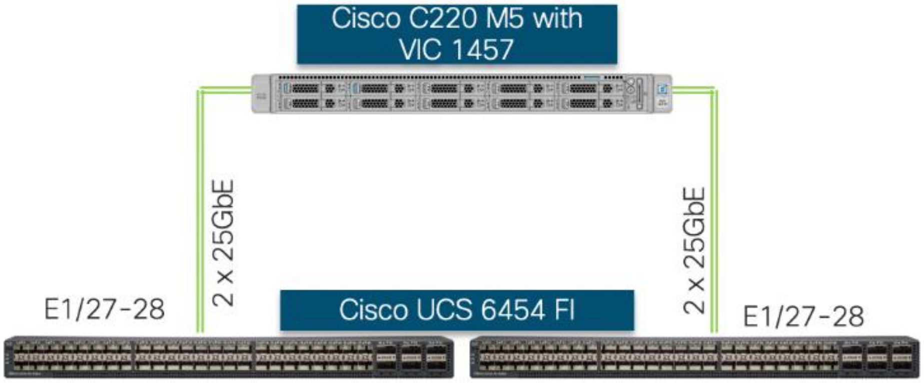
- Cisco UCS initial setup
- Cisco UCS software upgrade
- Configuring anonymous reporting and call home setup
- Configuring Block of IP addresses for KVM access
- Configuring NTP, DNS and additional users
- Configuring Info policy and Chassis Discovery Policy
- Configuring Server and Uplink ports and acknowledging various chassis and servers
- Configuring uplink port-channels to Cisco switches
- Configuring UDLD on the uplink port-channels

Some of the configuration parameters explained below (Policies, MAC or IP pools, and so on) might already be present when adding Cisco UCS C480 ML M5 servers to existing FlexPod environments. Based on their setup, you can reuse their existing pools and policies or define new policies as outlined in this document

Cisco UCS C220 M5 Connectivity

To manage the Cisco UCS C220 M5 platform with dual NVIDIA T4 GPUs using Cisco UCS Manager, the Cisco UCS C220 M5 is connected to the Cisco UCS 6454 FIs as shown in Figure 7. The ports connected to a fabric interconnect form a port-channel providing an effective 50GbE bandwidth to each fabric interconnect.

Figure 7   Cisco UCS C220 M5 to Cisco UCS 6454 FI Connectivity



Enable Server Ports

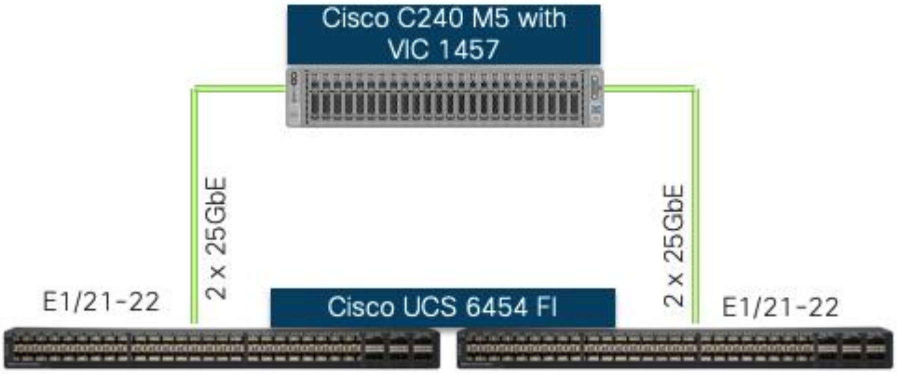
To enable and verify server ports, follow these steps:

1. In Cisco UCS Manager, click Equipment.
2. Expand Equipment > Fabric Interconnects > Fabric Interconnect A > Fixed Module.
3. Expand and select Ethernet Ports.
4. Select the ports that are connected to Cisco C240 M5 server (1/27 and 1/28), right-click them, and select Configure as Server Port.
5. Click Yes to confirm server ports and click OK.
6. Repeat this procedure to verify and set the C220 M5 ports connected to Fabric Interconnect B (1/27 and 1/28) as server ports

Cisco UCS C240 M5 Connectivity

To manage the Cisco UCS C240 M5 platform with dual GPUs using Cisco UCS Manager, the Cisco C240 M5 is connected to the Cisco UCS 6454 FIs as shown in Figure 8. The ports connected to a fabric interconnect form a port-channel providing an effective 50GbE bandwidth to each fabric interconnect.

Figure 8   Cisco UCS C240 M5 to Cisco UCS 6454 FI Connectivity



Enable Server Ports

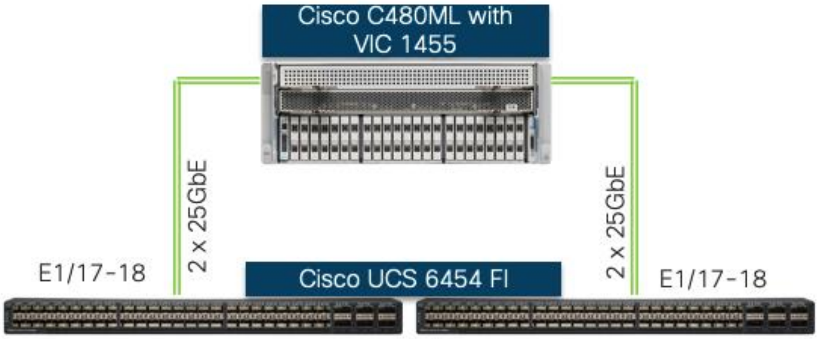
To enable and verify server ports, follow these steps:

1. In Cisco UCS Manager, click Equipment.
2. Expand Equipment > Fabric Interconnects > Fabric Interconnect A > Fixed Module.
3. Expand and select Ethernet Ports.
4. Select the ports that are connected to Cisco C240 M5 server (1/21 and 1/22), right-click them, and select Configure as Server Port.
5. Click Yes to confirm server ports and click OK.
6. Repeat this procedure to verify and set the Cisco C240 M5 ports connected to Fabric Interconnect B (1/19 and 1/20) as server ports

Cisco UCS C480 ML M5 Connectivity

To manage the Cisco UCS C480 ML platform using Cisco UCS Manager, the Cisco UCS C480 ML is connected to the Cisco UCS 6454 FIs as shown in Figure 9. The ports connected to a fabric interconnect form a port-channel providing an effective 50GbE bandwidth to each fabric interconnect.

Figure 9 Cisco UCS C480 ML M5 to Cisco UCS 6454 FI Connectivity



Enable Server Ports

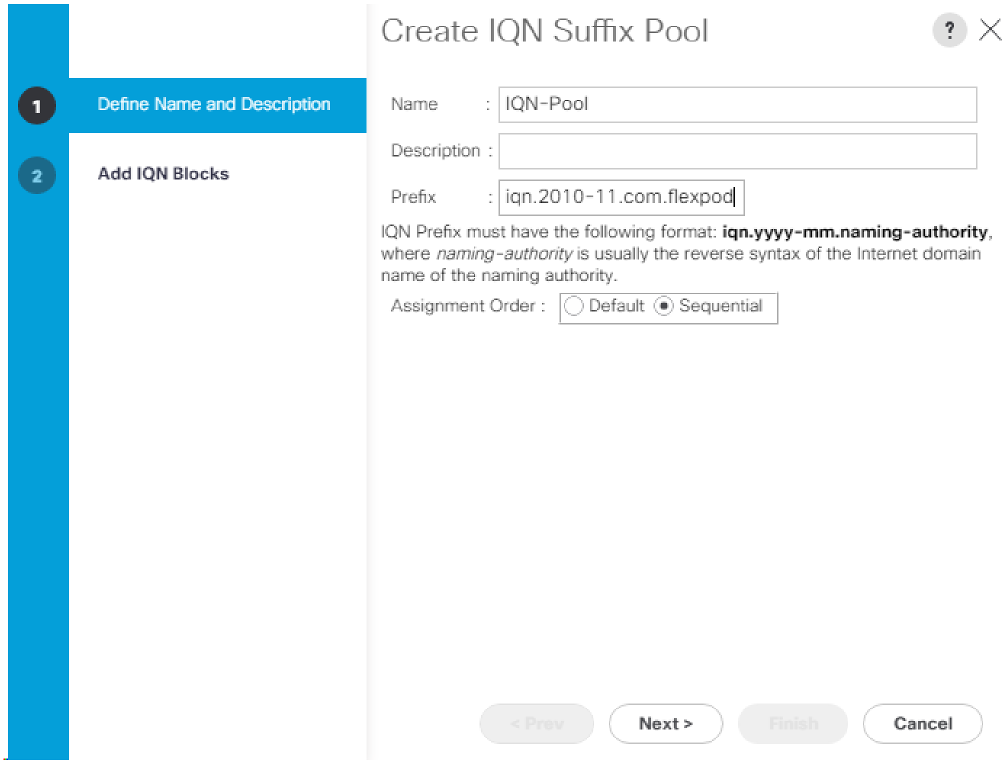
To enable and verify server ports, follow these steps:

1. In Cisco UCS Manager, click Equipment.
2. Expand Equipment > Fabric Interconnects > Fabric Interconnect A > Fixed Module.
3. Expand and select Ethernet Ports.
4. Select the ports that are connected to Cisco C480 ML server (1/17 and 1/18), right-click them, and select Configure as Server Port.
5. Click Yes to confirm server ports and click OK.
6. Repeat this procedure to verify and set the C480 ML ports connected to Fabric Interconnect B (1/17 and 1/18) as server ports
7. Repeat these steps for all the Cisco UCS C480 servers in the environment.

Create an IQN Pool for iSCSI Boot

To configure the necessary IQN pool to enabled iSCSI boot from SAN, follow these steps: on Cisco UCS Manager.

1. Select the SAN icon.
2. Select Pools > root.
3. Right-click IQN Pools under the root organization.
4. Select Create IQN Suffix Pool to create the IQN pool.
5. Enter `IQN-Pool` for the name of the IQN pool.
6. Optional: Enter a description for the IQN pool.
7. Enter `iqn.2010-11.com.flexpod` for the Prefix.
8. Select Sequential for Assignment Order.



Name : IQN-Pool

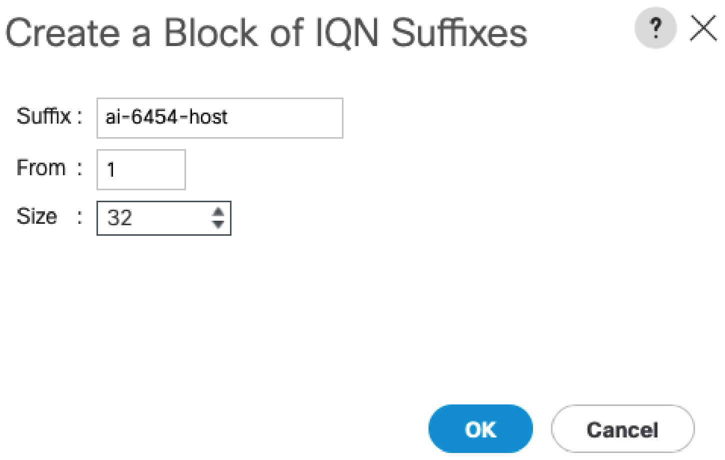
Description :

Prefix : iqn.2010-11.com.flexpod

IQN Prefix must have the following format: **iqn.yyyy-mm.naming-authority**, where *naming-authority* is usually the reverse syntax of the Internet domain name of the naming authority.

Assignment Order : ○ Default ● Sequential

9. Click Next.
10. Click Add under Create IQN Suffix Pool.
11. Enter a name to identify the individual UCS host for the Suffix.
12. Enter 1 for the From field.
13. Specify a size of the IQN block sufficient to support the available server resources.



Create a Block of IQN Suffixes

Suffix : ai-6454-host

From : 1

Size : 32

14. Click OK.
15. Click Finish and OK to complete creating the IQN pool.

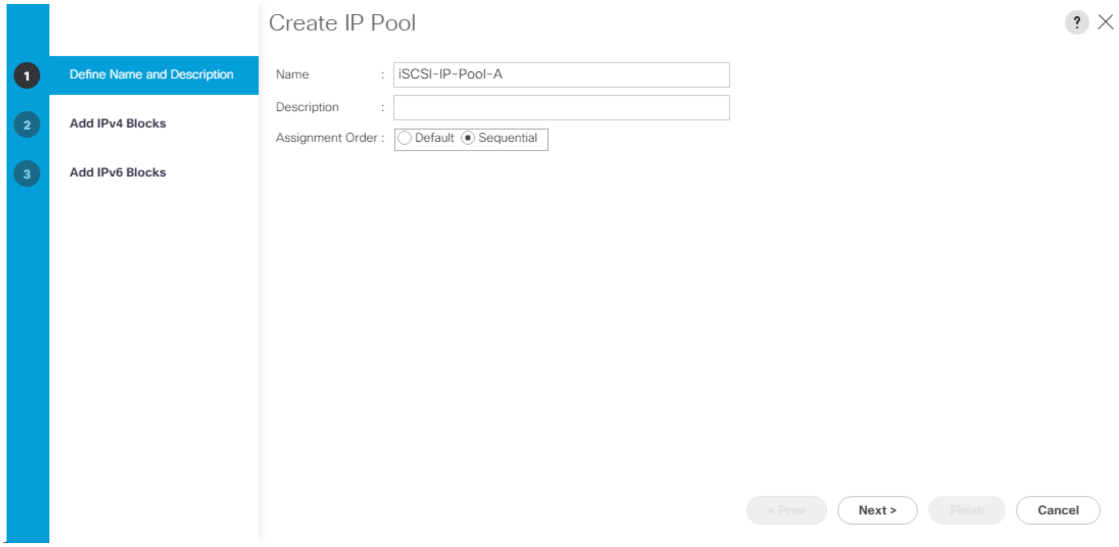
Create iSCSI Boot IP Address Pools

To configure the necessary iSCSI IP Address pools for the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the LAN icon.
2. Select and expand Pools > root.

In this procedure, two IP pools are created, one for each switching fabric.

3. Right-click IP Pools.
4. Select Create IP Pool to create the IP pool.
5. Enter `iSCSI-IP-Pool-A` as the name of the first IP pool.
6. Optional: Enter a description for the IP pool.
7. Select Sequential for Assignment Order.



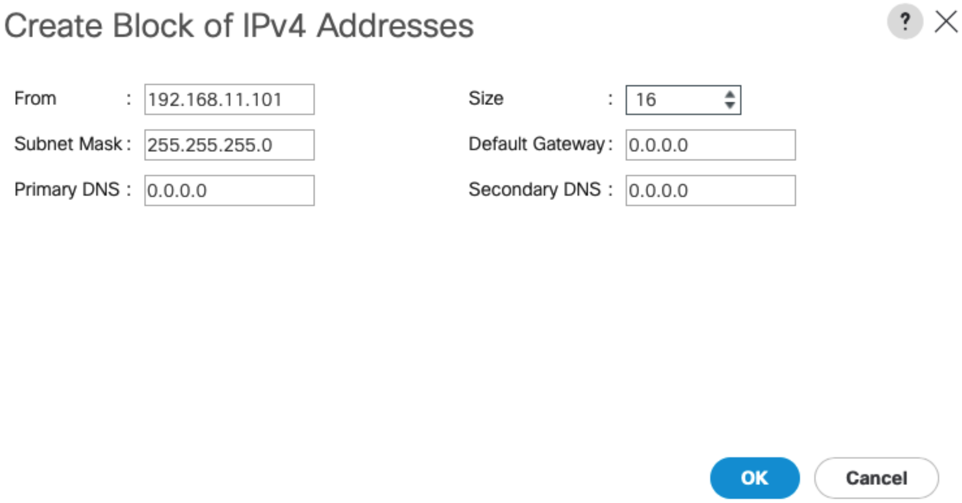
Create IP Pool

Name : iSCSI-IP-Pool-A

Description :

Assignment Order : ○ Default ● Sequential

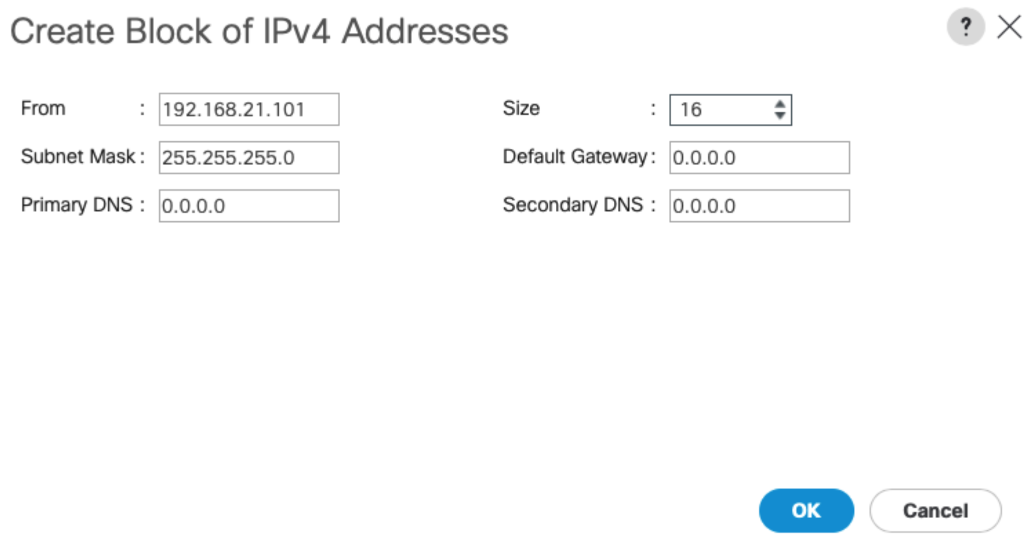
8. Click Next.
9. Click Add to add a Block of IPs to the pool.
10. Specify a starting IP address and subnet mask in the subnet <192.168.11.101/24> for iSCSI boot on Fabric A. It is not necessary to specify the Default Gateway or DNS server addresses.
11. Specify a size for the IP pool that is sufficient to support the available blade or server resources.



Create Block of IPv4 Addresses

From :	192.168.11.101	Size :	16
Subnet Mask :	255.255.255.0	Default Gateway :	0.0.0.0
Primary DNS :	0.0.0.0	Secondary DNS :	0.0.0.0

12. Click OK.
13. Click Next.
14. Click Finish.
15. In the confirmation message, click OK.
16. Right-click IP Pools.
17. Select Create IP Pool to create the IP pool.
18. Enter `iSCSI-IP-Pool-B` as the name of the second IP pool.
19. Optional: Enter a description for the IP pool.
20. Select Sequential for Assignment Order
21. Click Next.
22. Click Add to add a Block of IPs to the pool.
23. Specify a starting IP address and subnet mask in the subnet <192.168.21.101/24> for iSCSI boot on Fabric B. It is not necessary to specify the Default Gateway or DNS server addresses.
24. Specify a size for the IP pool that is sufficient to support the available blade or server resources.



Create Block of IPv4 Addresses

From :	192.168.21.101	Size :	16
Subnet Mask :	255.255.255.0	Default Gateway :	0.0.0.0
Primary DNS :	0.0.0.0	Secondary DNS :	0.0.0.0

25. Click OK.
26. Click Next.
27. Click Finish.
28. In the confirmation message, click OK.

Create MAC Address Pools

To configure the necessary MAC address pools, follow these steps:

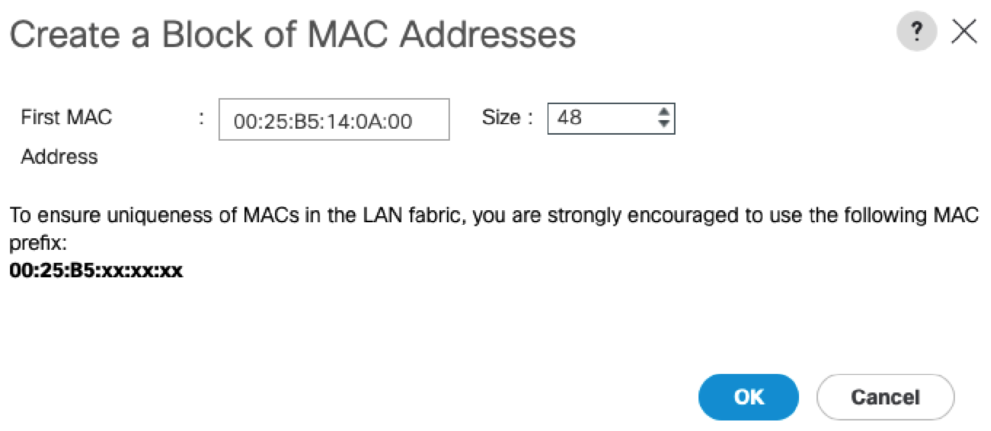
1. In Cisco UCS Manager, click the LAN icon.
2. Select Pools > root.

In this procedure, two MAC address pools are created, one for each switching fabric.

3. Right-click MAC Pools and select Create MAC Pool to create the MAC address pool.
4. Enter `MAC-Pool-A` as the name of the MAC pool.
5. Optional: Enter a description for the MAC pool.
6. Select Sequential as the option for Assignment Order.
7. Click Next.
8. Click Add.
9. Specify a starting MAC address.

For the FlexPod solution, it is recommended to place 0A in the next-to-last octet of the starting MAC address to identify all of the MAC addresses as Fabric A addresses. In this example, the rack number (14) information was also included in the MAC address: 00:25:B5:14:0A:00 as our first MAC address.

10. Specify a size for the MAC address pool that is sufficient to support the available blade or server resources assuming that multiple vNICs can be configured on each server.



Create a Block of MAC Addresses

First MAC Address : 00:25:B5:14:0A:00 Size : 48

To ensure uniqueness of MACs in the LAN fabric, you are strongly encouraged to use the following MAC prefix:
00:25:B5:xx:xx:xx

11. Click OK.
12. Click Finish.
13. In the confirmation message, click OK.
14. Right-click MAC Pools and select Create MAC Pool to create the MAC address pool.
15. Enter `MAC-Pool-B` as the name of the MAC pool.
16. Optional: Enter a description for the MAC pool.
17. Select Sequential as the option for Assignment Order.
18. Click Next.
19. Click Add.
20. Specify a starting MAC address.

For the FlexPod solution, it is recommended to place 0B in the next-to-last octet of the starting MAC address to identify all of the MAC addresses as Fabric B addresses. In this example, the rack number (14) information was also included in the MAC address: 00:25:B5:14:0B:00 as our first MAC address.

21. Specify a size for the MAC address pool that is sufficient to support the available blade or server resources.
22. Click OK.
23. Click Finish.
24. In the confirmation message, click OK.

Create UUID Suffix Pool

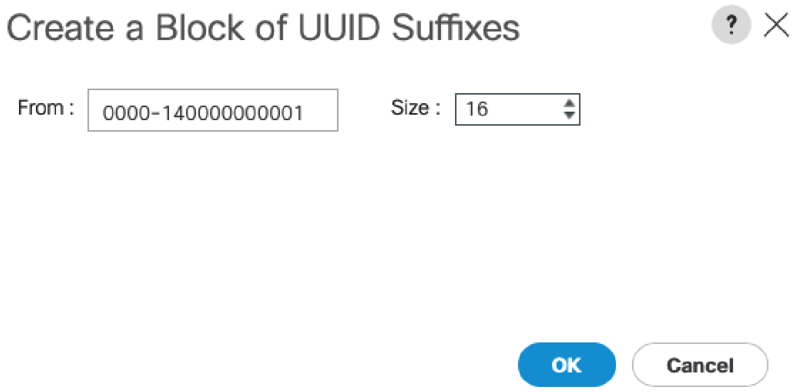
To configure the necessary universally unique identifier (UUID) suffix pool for the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the Servers icon.
2. Select Pools > root.
3. Right-click UUID Suffix Pools.
4. Select Create UUID Suffix Pool.

5. Enter `UUID-Pool` as the name of the UUID suffix pool.
6. Optional: Enter a description for the UUID suffix pool.
7. Keep the prefix at the Derived option.
8. Select Sequential for the Assignment Order.
9. Click Next.
10. Click Add to add a block of UUIDs.
11. Keep the From field at the default setting.

Optional: An identifier such as Rack Number or LAB ID can be embedded in the UUID.

12. Specify a size for the UUID block that is sufficient to support the available blade or server resources.



Create a Block of UUID Suffixes

From : 0000-140000000001

Size : 16

13. Click OK.
14. Click Finish.
15. Click OK.

Create Server Pool

To configure the necessary server pool for the Cisco UCS C480 ML and/or Cisco UCS C240M5 servers (with GPUs), follow these steps:

Consider creating unique server pools to achieve the deployment granularity.

1. In Cisco UCS Manager, click the Servers icon.
2. Expand Pools > root.
3. Right-click Server Pools and Select Create Server Pool.
4. Enter `C480ML` as the name of the server pool.
5. Optional: Enter a description for the server pool.
6. Click Next.
7. Select the Cisco UCS C480 ML servers to be used in the environment and click >> to add them to the server pool.

8. Click Finish.
9. Click OK.
10. Right-click Server Pools and Select Create Server Pool.
11. Enter C240 as the name of the server pool.
12. Optional: Enter a description for the server pool.
13. Click Next.
14. Select the Cisco UCS C240 M5 servers to be used in the environment and click >> to add them to the server pool.
15. Click Finish.
16. Click OK.
17. Right-click Server Pools and Select Create Server Pool.
18. Enter C220 as the name of the server pool.
19. Optional: Enter a description for the server pool.
20. Click Next.
21. Select the Cisco UCS C220 M5 servers to be used in the environment and click >> to add them to the server pool.
22. Click Finish.
23. Click OK.

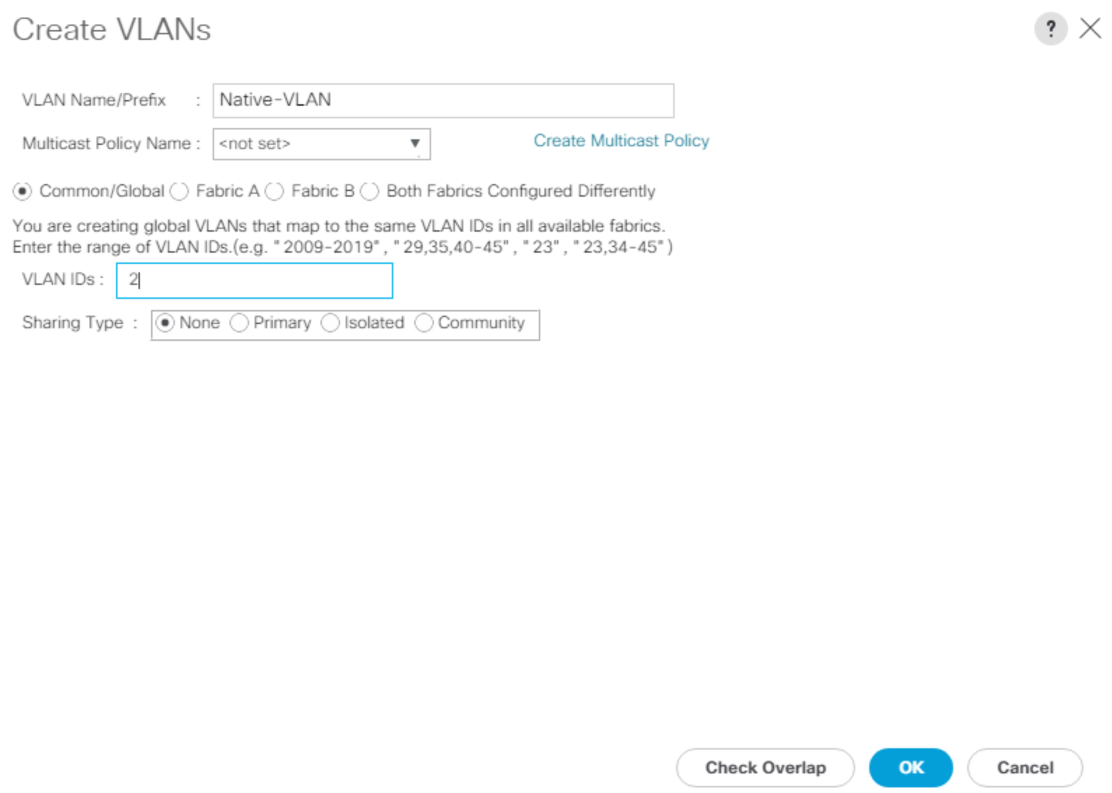
Create VLANs

To configure the necessary VLANs explained in Table 2, follow these steps:

1. In Cisco UCS Manager, click the LAN icon.

In this procedure, 6 unique VLANs (Table 2) are created.

2. Select LAN > LAN Cloud.
3. Right-click VLANs.
4. Select Create VLANs.
5. Enter “`Native-VLAN`” as the name of the VLAN to be used as the native VLAN.
6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter the native VLAN ID <2>.
8. Keep the Sharing Type as None.
9. Click OK and then click OK again.



Create VLANs

VLAN Name/Prefix : Native-VLAN

Multicast Policy Name : <not set>

● Common/Global ○ Fabric A ○ Fabric B ○ Both Fabrics Configured Differently

You are creating global VLANs that map to the same VLAN IDs in all available fabrics.
Enter the range of VLAN IDs.(e.g. " 2009-2019" , " 29,35,40-45" , " 23" , " 23,34-45")

VLAN IDs : 2

Sharing Type : ● None ○ Primary ○ Isolated ○ Community

10. Expand the list of VLANs in the navigation pane, right-click the newly created "`Native-VLAN`" and select Set as Native VLAN.
11. Click Yes and then click OK.
12. Right-click VLANs.
13. Select Create VLANs.
14. Enter "`IB-MGMT-VLAN`" as the name of the VLAN to be used for management traffic.
15. Keep the Common/Global option selected for the scope of the VLAN.
16. Enter the In-Band management VLAN ID <20>.
17. Keep the Sharing Type as None.
18. Click OK, and then click OK again.
19. Right-click VLANs.
20. Select Create VLANs.
21. Enter "`AI-ML-NFS`" as the name of the VLAN.
22. Keep the Common/Global option selected for the scope of the VLAN.
23. Enter the NFS VLAN ID <3152>.
24. Keep the Sharing Type as None.
25. Click OK, and then click OK again.
26. (Optional) Right-click VLANs to create the Data Traffic VLAN.

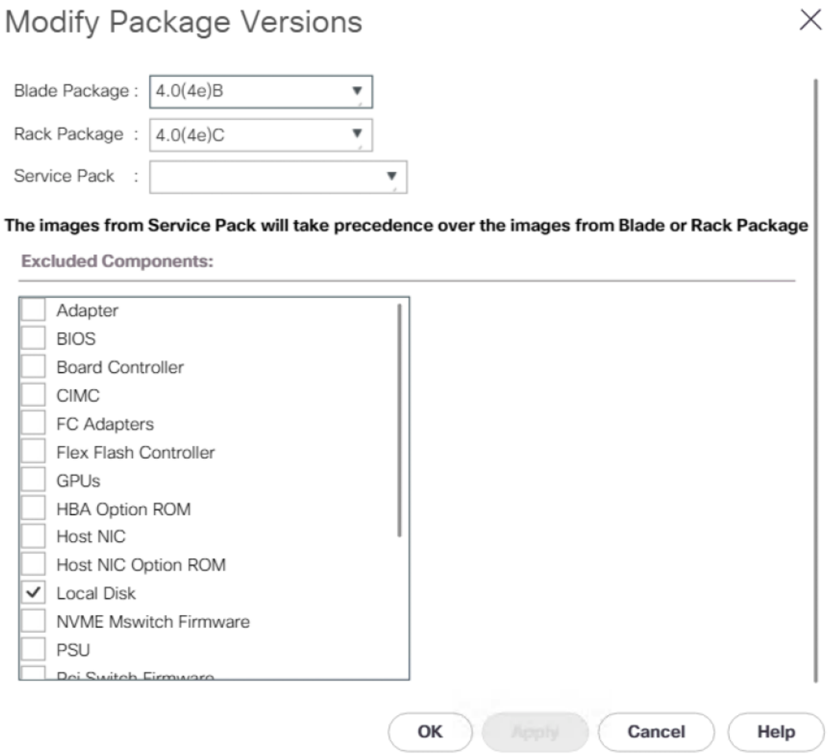
27. Select Create VLANs.
28. Enter “`Data-Traffic`” as the name of the VLAN to be used for VMware vMotion.
29. Keep the Common/Global option selected for the scope of the VLAN.
30. Enter the traffic VLAN ID <220>.
31. Keep the Sharing Type as None.
32. Click OK, and then click OK again.
33. Right-click VLANs.
34. Select Create VLANs.
35. Enter “`iSCSI-A`” as the name of the VLAN to be used for UCS Fabric A iSCSI boot.
36. Select Fabric A as the scope of the VLAN.
37. Enter the UCS Fabric A iSCSI boot VLAN ID <3111>.
38. Keep the Sharing Type as None.
39. Click OK, and then click OK again.
40. Right-click VLANs.
41. Select Create VLANs.
42. Enter “`iSCSI-B`” as the name of the VLAN to be used for UCS Fabric B iSCSI boot.
43. Select Fabric B as the scope of the VLAN.
44. Enter the UCS Fabric B iSCSI boot VLAN ID <3121>.
45. Keep the Sharing Type as None.
46. Click OK, and then click OK again.

Modify Default Host Firmware Package

Firmware management policies allow the administrator to select the corresponding packages for a given server configuration. These policies often include packages for adapter, BIOS, board controller, FC adapters, host bus adapter (HBA) option ROM, and storage controller properties.

To specify the Cisco UCS 4.0(4e) release for the Default firmware management policy for a given server configuration in the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the Servers icon.
2. Select Policies > root.
3. Expand Host Firmware Packages.
4. Select default.
5. In the Actions pane, select Modify Package Versions.
6. Select the version 4.0(4e)B (Optional) for the Blade Package, and 4.0(4e)C for the Rack Package.
7. Leave Excluded Components with only Local Disk selected.



Modify Package Versions

Blade Package : 4.0(4e)B

Rack Package : 4.0(4e)C

Service Pack :

The images from Service Pack will take precedence over the images from Blade or Rack Package

Excluded Components:

- ☐ Adapter
- ☐ BIOS
- ☐ Board Controller
- ☐ CIMC
- ☐ FC Adapters
- ☐ Flex Flash Controller
- ☐ GPUs
- ☐ HBA Option ROM
- ☐ Host NIC
- ☐ Host NIC Option ROM
- ☑ Local Disk
- ☐ NVME Mswitch Firmware
- ☐ PSU
- ☐ Pci Switch Firmware

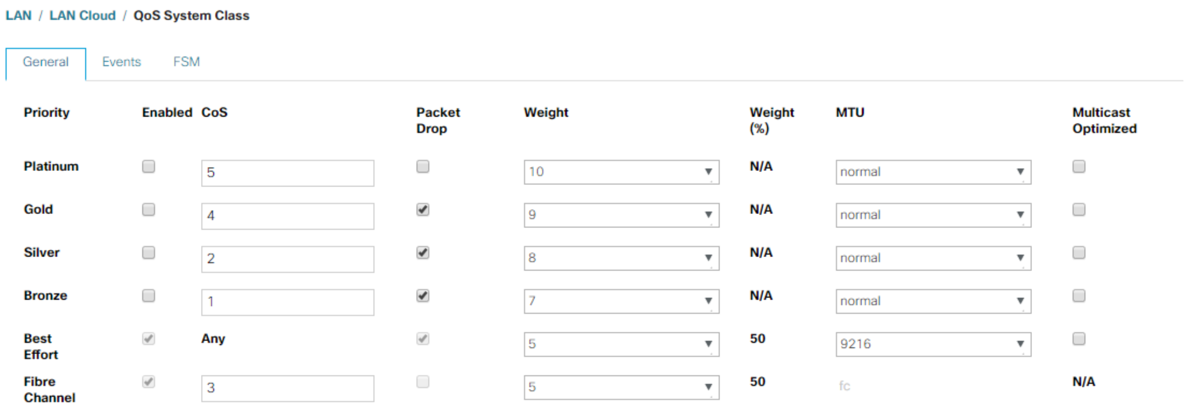
8. Click OK then click OK again to modify the host firmware package.

Set Jumbo Frames in Cisco UCS Fabric

To configure jumbo frames and enable the base quality of service in the Cisco UCS fabric, follow these steps:

This procedure does not apply to Cisco UCS 6454 Fabric Interconnect since the default normal MTU for Best Effort Class is 9216 and cannot be changed.

1. In Cisco UCS Manager, click the LAN icon.
2. Select LAN > LAN Cloud > QoS System Class.
3. In the right pane, click the General tab.
4. On the Best Effort row, enter 9216 in the box under the MTU column.
5. Click Save Changes in the bottom of the window.
6. Click OK.



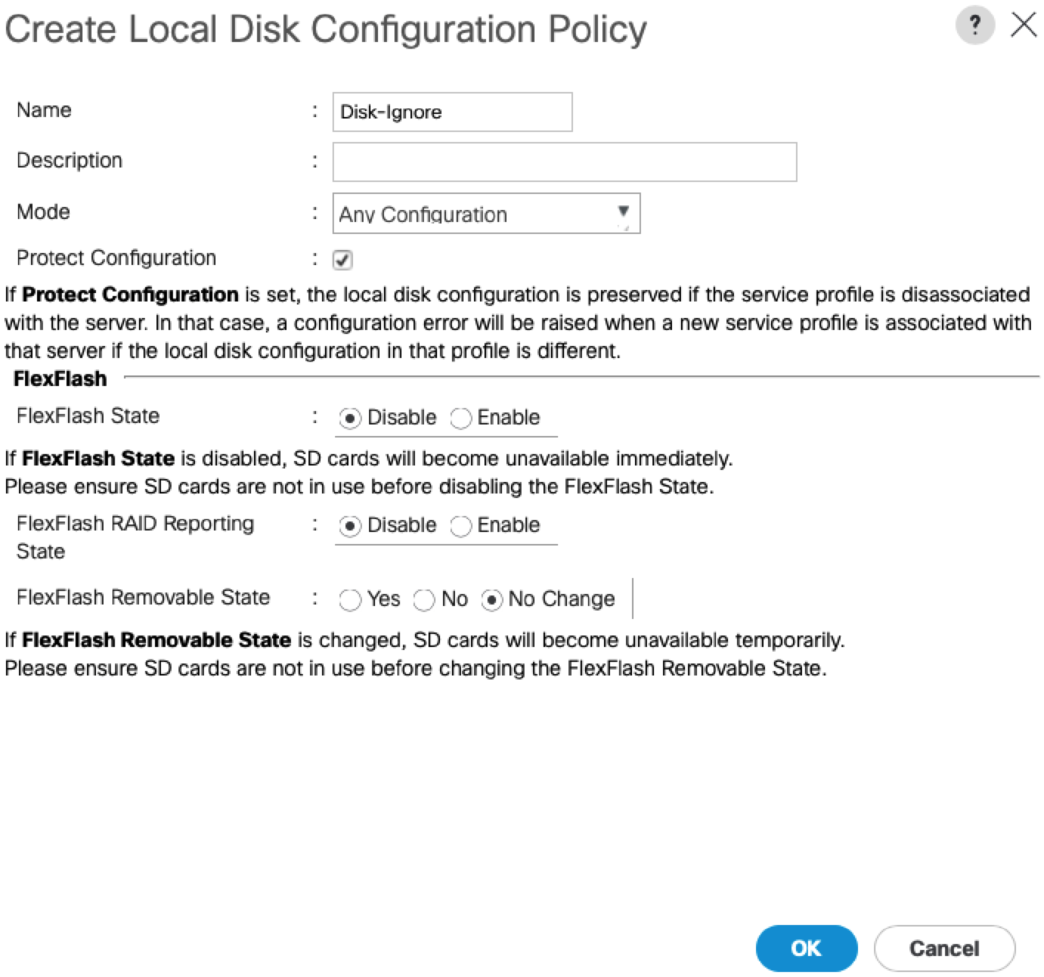
LAN / LAN Cloud / QoS System Class

Priority	Enabled	CoS	Packet Drop	Weight	Weight (%)	MTU	Multicast Optimized
Platinum	□	5	□	10	N/A	normal	□
Gold	□	4	☑	9	N/A	normal	□
Silver	□	2	☑	8	N/A	normal	□
Bronze	□	1	☑	7	N/A	normal	□
Best Effort	☑	Any	☑	5	50	9216	□
Fibre Channel	☑	3	□	5	50	fc	N/A

Create Local Disk Configuration Policy

To create a local disk configuration policy to ignore any local disks (when present), follow these steps:

1. In Cisco UCS Manager, click the Servers icon.
2. Select Policies > root.
3. Right-click Local Disk Config Policies.
4. Select Create Local Disk Configuration Policy.
5. Enter `Disk-Ignore` as the local disk configuration policy name.
6. Change the mode to Any Configuration.
7. Click OK to create the local disk configuration policy.



Create Local Disk Configuration Policy

Name	:	Disk-Ignore
Description	:	
Mode	:	Any Configuration
Protect Configuration	:	☑

If **Protect Configuration** is set, the local disk configuration is preserved if the service profile is disassociated with the server. In that case, a configuration error will be raised when a new service profile is associated with that server if the local disk configuration in that profile is different.

FlexFlash

FlexFlash State : ● Disable ○ Enable

If **FlexFlash State** is disabled, SD cards will become unavailable immediately.
Please ensure SD cards are not in use before disabling the FlexFlash State.

FlexFlash RAID Reporting State : ● Disable ○ Enable

FlexFlash Removable State : ○ Yes ○ No ● No Change

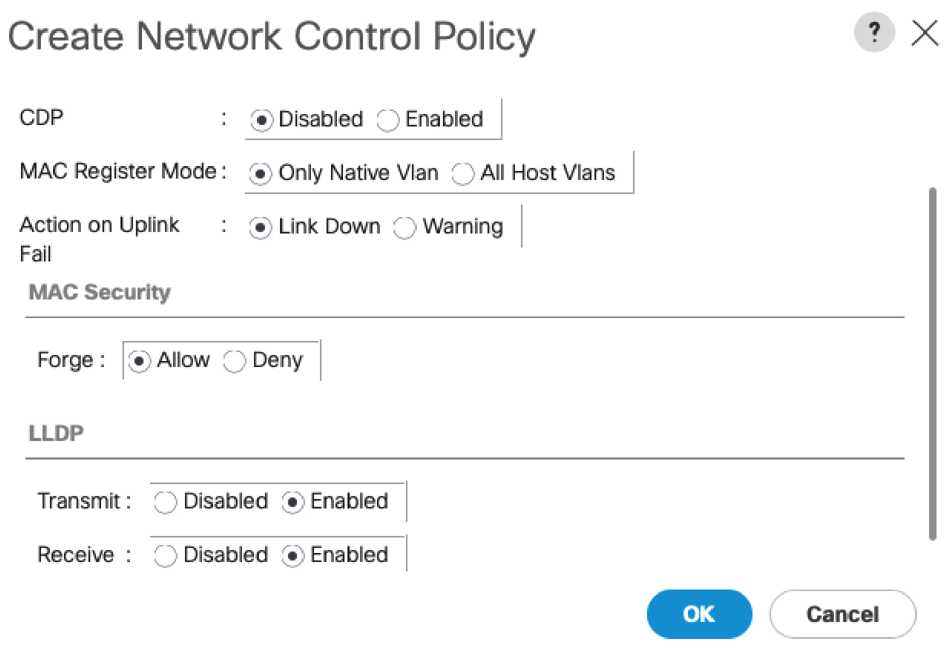
If **FlexFlash Removable State** is changed, SD cards will become unavailable temporarily.
Please ensure SD cards are not in use before changing the FlexFlash Removable State.

8. Click OK.

Create Network Control Policy to Enable Link Layer Discovery Protocol (LLDP)

To create a network control policy to enable LLDP on virtual network ports, follow these steps:

1. In Cisco UCS Manager, click the LAN icon.
2. Select Policies > root.
3. Right-click Network Control Policies.
4. Select Create Network Control Policy.
5. Enter `Enable-LLDP` as the policy name.
6. For LLDP, scroll down and select Enabled for both Transmit and Receive.
7. Click OK to create the network control policy.



Create Network Control Policy

CDP : ● Disabled ○ Enabled

MAC Register Mode : ● Only Native Vlan ○ All Host Vlans

Action on Uplink Fail : ● Link Down ○ Warning

MAC Security

Forge : ● Allow ○ Deny

LLDP

Transmit : ○ Disabled ● Enabled

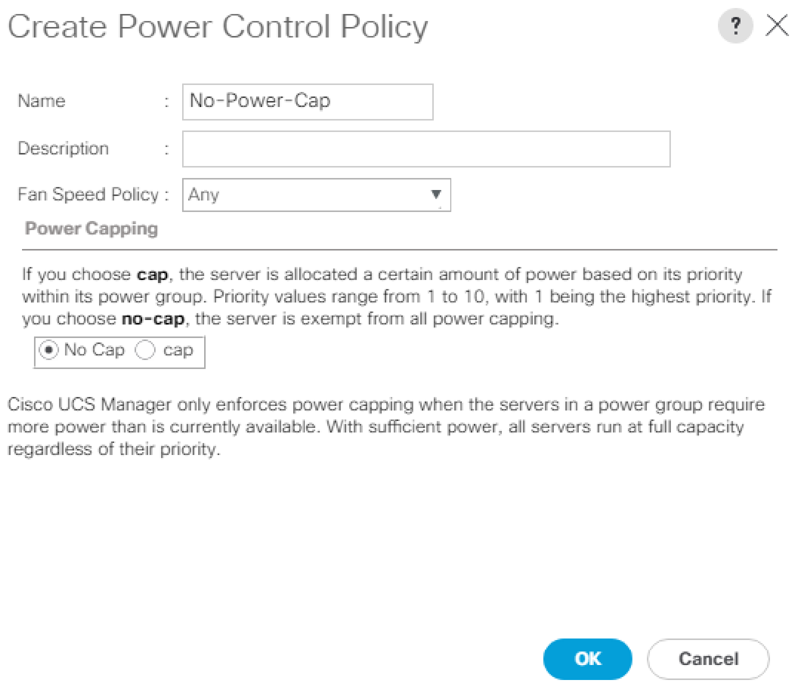
Receive : ○ Disabled ● Enabled

8. Click OK.

Create Power Control Policy

To create a power control policy for the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the Servers icon.
2. Select Policies > root.
3. Right-click Power Control Policies.
4. Select Create Power Control Policy.
5. Enter `No-Power-Cap` as the power control policy name.
6. Change the power capping setting to No Cap.
7. Click OK to create the power control policy.
8. Click OK.



Create Power Control Policy

Name : No-Power-Cap

Description :

Fan Speed Policy : Any

Power Capping

If you choose **cap**, the server is allocated a certain amount of power based on its priority within its power group. Priority values range from 1 to 10, with 1 being the highest priority. If you choose **no-cap**, the server is exempt from all power capping.

● No Cap ○ cap

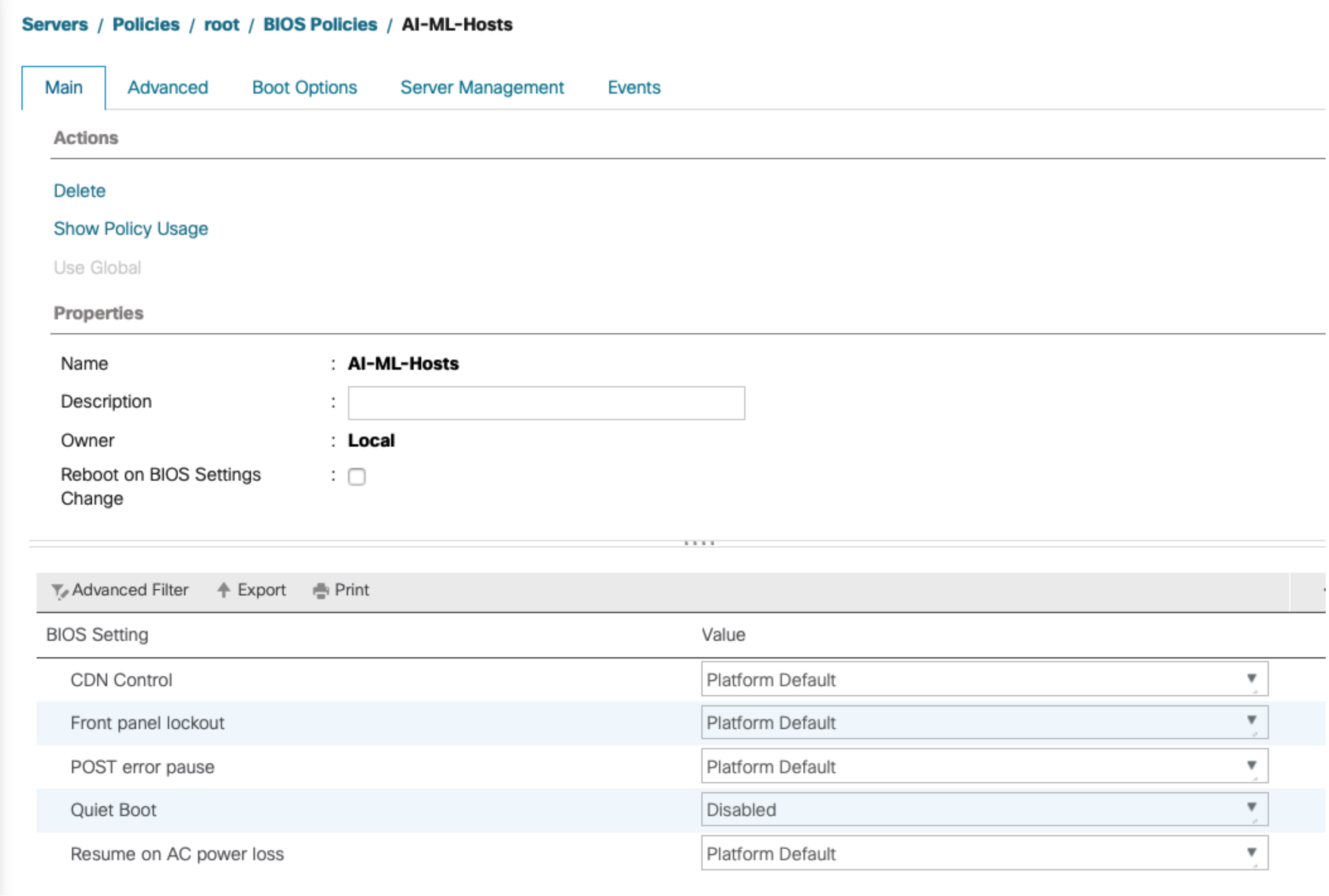
Cisco UCS Manager only enforces power capping when the servers in a power group require more power than is currently available. With sufficient power, all servers run at full capacity regardless of their priority.

Create Server BIOS Policy

To create a server BIOS policy for the Cisco UCS C480 ML M5 and Cisco UCS C240 M5, follow these steps:

BIOS settings can have a significant performance impact, depending on the workload and the applications. The BIOS settings listed in this section are for configurations optimized for enhanced performance. These setting can be adjusted based on the application, performance, and energy efficiency requirements.

1. In Cisco UCS Manager, click the Servers icon.
2. Select Policies > root.
3. Right-click BIOS Policies.
4. Select Create BIOS Policy.
5. Enter `AI-ML-Hosts` as the BIOS policy name.
6. Click OK then OK again.
7. Expand BIOS Policies and select AI-ML-Hosts.
8. Set the following within the Main tab:
   a. Quiet Boot > Disabled



Name : **AI-ML-Hosts**

Description :

Owner : **Local**

Reboot on BIOS Settings Change : □

BIOS Setting	Value
CDN Control	Platform Default
Front panel lockout	Platform Default
POST error pause	Platform Default
Quiet Boot	Disabled
Resume on AC power loss	Platform Default

9. Click Save Changes and OK.

Further changes will only be made in "Processor" and "RAS Memory" sub-tabs under "Advanced."

10. Click the Advanced tab and then select the Processor tab.
11. Set the following within the Processor tab:
    a. CPU Performance > Enterprise
    b. Core Multi Processing > All
    c. DRAM Clock Throttling > Performance
    d. Direct Cache Access > Enabled
    e. Enhanced Intel SpeedStep Tech >Disabled
    f. Intel HyperThreading Tech > Enabled
    g. Intel Turbo Boost Tech > Enabled
    h. Intel Virtualization Technology > Disabled
    i. Channel Interleaving > Auto
    j. P STATE Coordination > HW ALL

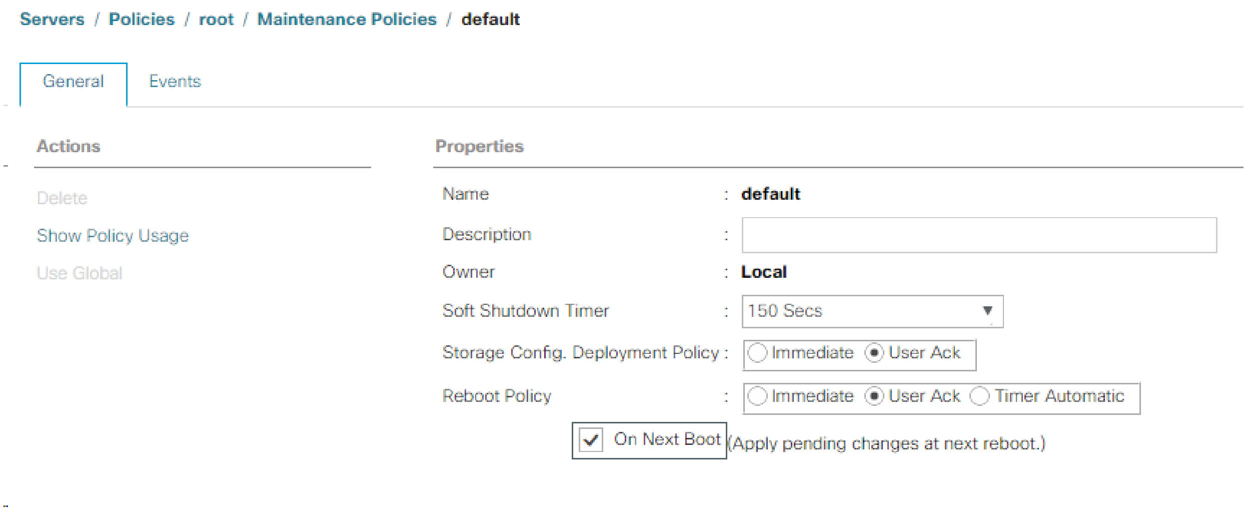
k. Processor C State > Disabled
l. Processor C1E > Disabled
m. Processor C3 Report > Disabled
n. Processor C6 Report > Disabled
o. Processor C7 Report > Disabled
p. Power Technology > Performance
q. Energy Performance > Performance
r. Adjacent Cache Line Prefetcher > Enabled
s. DCU IP Prefetcher > Enabled
t. DCU Streamer Prefetch > Enabled
u. Hardware Prefetcher > Enabled
v. UPI Prefetch > Enabled
w. LLC Prefetch > Enabled
x. XPT Prefetch > Enabled
y. Demand Scrub > Enabled
z. Patrol Scrub > Enabled

12. Click Save Changes and OK.
13. Click the RAS Memory sub-tab and select:
    a. DRAM Refresh Rate > 1x
    b. Memory RAS configuration > Maximum Performance
14. Click Save Changes and OK.

Update the Default Maintenance Policy

To update the default Maintenance Policy, follow these steps:

1. In Cisco UCS Manager, click the Servers icon.
2. Select Policies > root.
3. Select Maintenance Policies > default.
4. Change the Reboot Policy to User Ack.
5. Select “On Next Boot” to delegate maintenance windows to server administrators.



Servers / Policies / root / Maintenance Policies / default

Properties

Name : **default**

Description :

Owner : **Local**

Soft Shutdown Timer : 150 Secs ▼

Storage Config. Deployment Policy : ○ Immediate ● User Ack

Reboot Policy : ○ Immediate ● User Ack ○ Timer Automatic

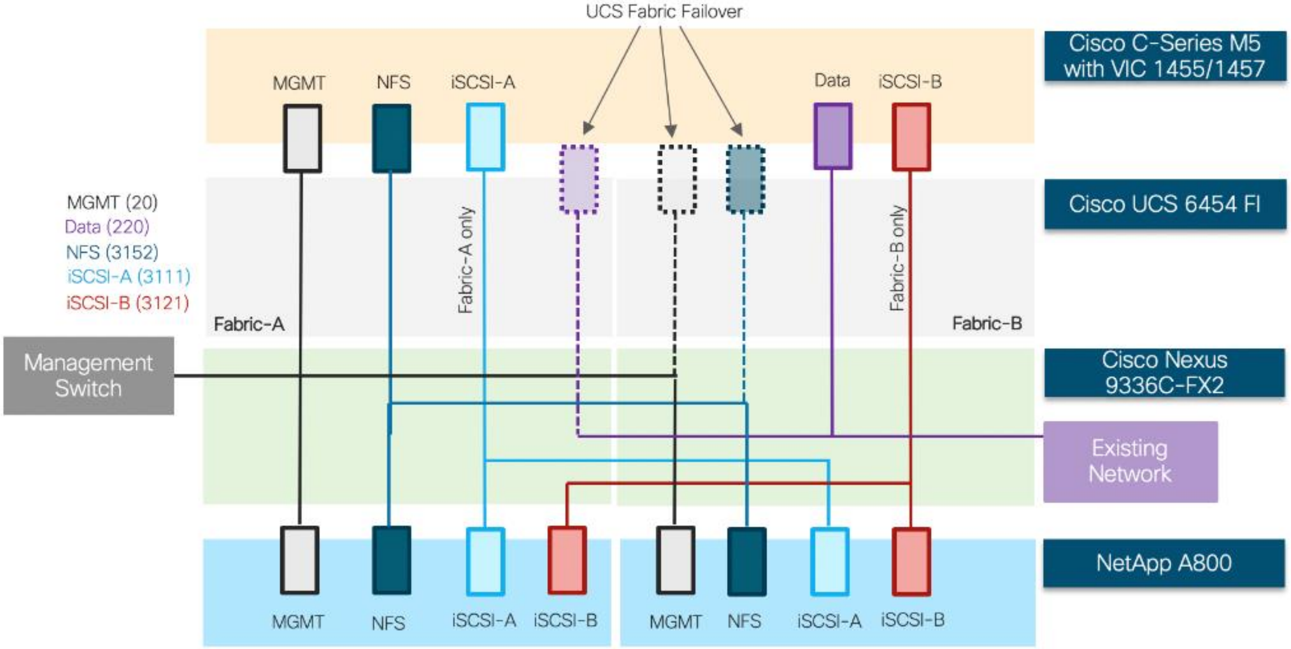
☑ On Next Boot (Apply pending changes at next reboot.)

6. Click OK to save changes.
7. Click OK to accept the change.

Create vNIC Templates

Five vNICs are deployed for each Cisco UCS C-Series server as shown in Figure 10.

Figure 10 Cisco UCS C-Series vNIC configuration



These five vNICs are configured as follows:

- 1 management vNIC interface where management VLAN (20) is configured as native VLAN. The management interface is configured on Fabric A with fabric failover is enabled. This vNIC uses standard MTU value of 1500.

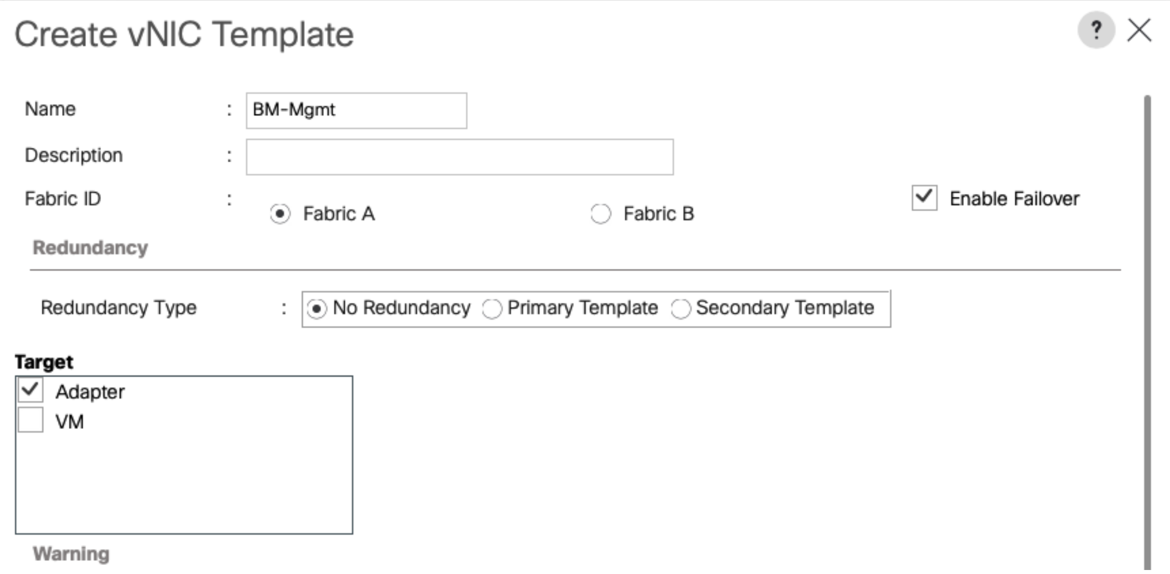
- 1 iSCSI-A vNIC that utilizes iSCSI-A VLAN (3111) as the native VLAN to provide access to iSCSI-A path. The MTU value for this interface is set as a Jumbo MTU (9000).
- 1 iSCSI-B vNIC utilizes iSCSI-B VLAN (3121) as a native VLAN to provide access to iSCSI-B path. The MTU value for this interface is set as a Jumbo MTU (9000).
- 1 NFS vNIC interface where NFS VLAN (3152) is configured as native VLAN. The NFS interface is configured on Fabric A with fabric failover is enabled. The MTU value for this interface is set as a Jumbo MTU (9000).
- (Optional) 1 Data vNIC interface where data traffic VLAN (220) is configured as native VLAN. The Data interface is configured on Fabric B with fabric failover enabled. The MTU value for this interface is set as a Jumbo MTU (9000).

The following section provides the steps to create multiple virtual network interface card (vNIC) templates for the Cisco UCS environment.

Create Management vNIC Template

To create vNIC template for host management access, follow these steps:

1. In Cisco UCS Manager, click the LAN icon.
2. Expand Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template.
5. Enter `BM-Mgmt` as the vNIC template name.
6. Keep Fabric A selected.
7. Select the Enable Failover checkbox.
8. Select No Redundancy for Redundancy Type.
9. Under Target, make sure that only the Adapter checkbox is selected.
10. Select Updating Template as the Template Type.



Create vNIC Template

Name : BM-Mgmt

Description :

Fabric ID : ● Fabric A ○ Fabric B ☑ Enable Failover

Redundancy

Redundancy Type : ● No Redundancy ○ Primary Template ○ Secondary Template

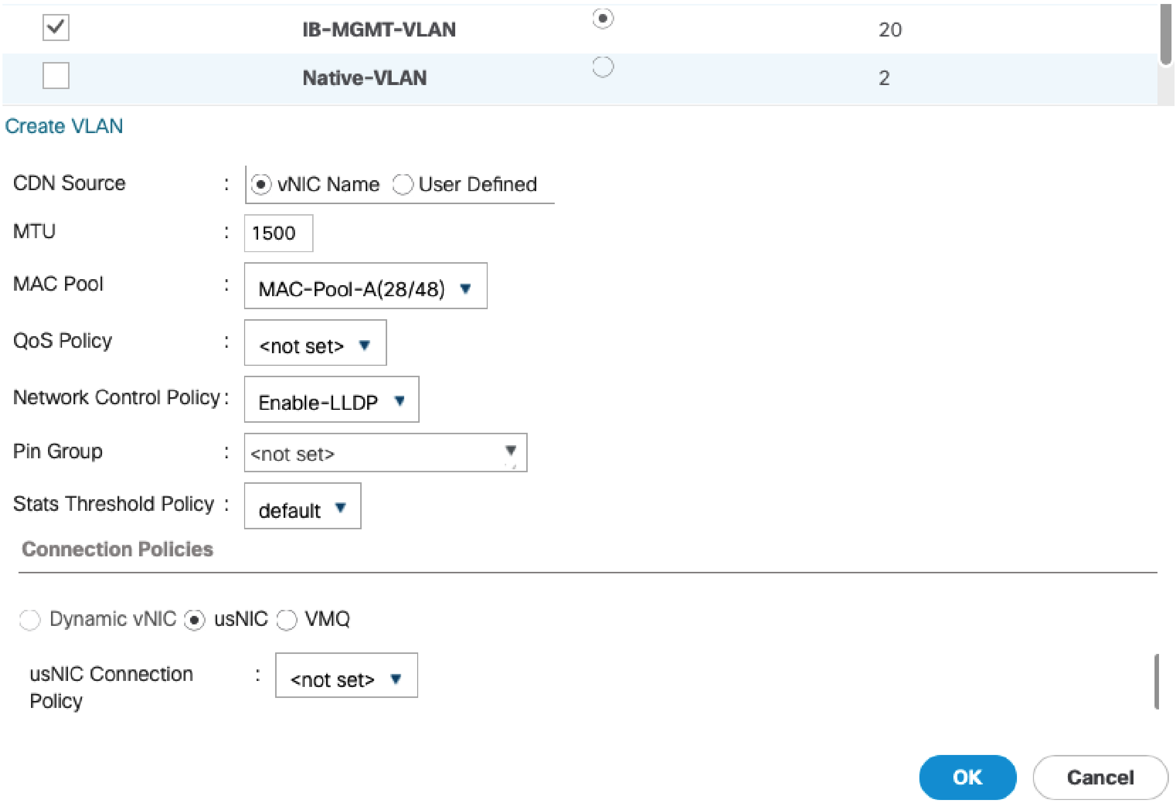
Target

☑ Adapter

□ VM

Warning

11. Under VLANs, select the checkboxes for the IB-MGMT-VLAN.
12. Set IB-MGMT-VLAN as the native VLAN.
13. Leave MTU at 1500.
14. In the MAC Pool list, select MAC-Pool-A.
15. In the Network Control Policy list, select Enable-LLDP.



☑	IB-MGMT-VLAN	●	20
□	Native-VLAN	○	2

Create VLAN

CDN Source : ● vNIC Name ○ User Defined

MTU : 1500

MAC Pool : MAC-Pool-A(28/48)

QoS Policy : <not set>

Network Control Policy : Enable-LLDP

Pin Group : <not set>

Stats Threshold Policy : default

Connection Policies

○ Dynamic vNIC ● usNIC ○ VMQ

usNIC Connection Policy : <not set>

16. Click OK to create the vNIC template.
17. Click OK.

Create iSCSI Boot vNIC Templates

To create iSCSI Boot vNIC templates, follow these steps:

1. In Cisco UCS Manager, click the LAN icon.
2. Expand Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template.
5. Enter `BM-iSCSI-A` as the vNIC template name.
6. Keep Fabric A selected.
7. Do not select the Enable Failover checkbox.
8. Select No Redundancy for Redundancy Type.

9. Under Target, make sure that only the Adapter checkbox is selected.
10. Select Updating Template as the Template Type.
11. Under VLANs, select the checkbox for iSCSI-A.
12. Set iSCSI-A as the native VLAN.
13. Select vNIC Name for the CDN Source.
14. For MTU, enter 9000.
15. In the MAC Pool list, select MAC-Pool-A.
16. In the Network Control Policy list, select Enable-LLDP.
17. Click OK to create the vNIC template.
18. Click OK.
19. Right-click vNIC Templates again.
20. Select Create vNIC Template.
21. Enter `BM-iSCSI-B` as the vNIC template name.
22. Select Fabric B.
23. Do not elect the Enable Failover checkbox.
24. Select No Redundancy for Redundancy Type.
25. Under Target, make sure that only the Adapter checkbox is selected.
26. Select Updating Template as the Template Type.
27. Under VLANs, select the checkbox for iSCSI-B.
28. Set iSCSI-B as the native VLAN.
29. Select vNIC Name for the CDN Source.
30. For MTU, enter 9000.
31. In the MAC Pool list, select MAC-Pool-B.
32. In the Network Control Policy list, select Enable-LLDP.
33. Click OK to create the vNIC template.
34. Click OK.

Create NFS vNIC Template

To create vNIC template for accessing NFS storage over Fabric A, follow these steps:

1. In Cisco UCS Manager, click the LAN icon.
2. Expand Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template.
5. Enter `BM-NFS-A` as the vNIC template name.
6. Keep Fabric A selected.

7. Select the Enable Failover checkbox.
8. Set the Redundancy Type to No Redundancy.
9. Under Target, make sure that only the Adapter checkbox is selected.
10. Select Updating Template as the Template Type.
11. Under VLANs, select the `AI-ML-NFS` VLAN.
12. Set the `AI-ML-NFS` VLAN as the native VLAN.
13. Select vNIC Name for the CDN Source.
14. For MTU, enter 9000.
15. In the MAC Pool list, select MAC-Pool-A.
16. In the Network Control Policy list, select Enable-LLDP.
17. Click OK to create the vNIC template.
18. Click OK.

(Optional) If a customer environment requires access to NFS storage over Fabric B, use the same procedure but select Fabric B in step 6 and MAC-Pool-B in step 15.

(Optional) Create Traffic vNIC Template

To create a dedicated vNIC for AI-ML host to communicate with other hosts and virtual machines, follow these steps:

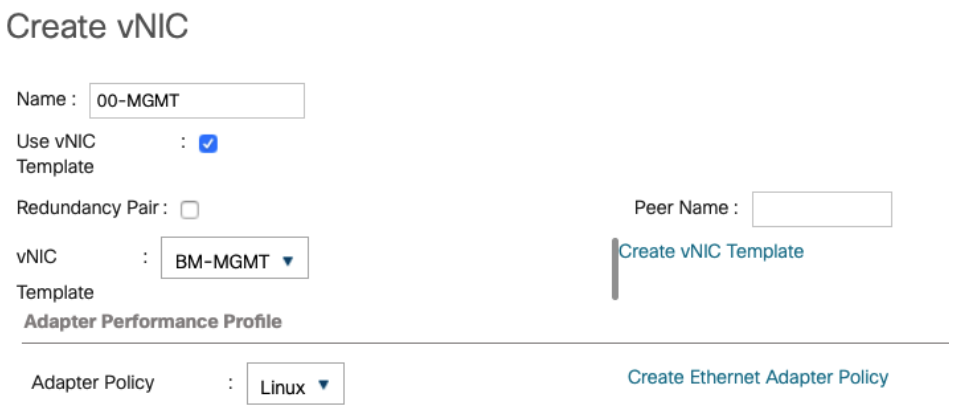
1. In Cisco UCS Manager, click the LAN icon.
2. Expand Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template.
5. Enter `BM-Traffic-B` as the vNIC template name.
6. Select Fabric B.
7. Select the Enable Failover checkbox.
8. Set the Redundancy Type to No Redundancy
9. Under Target, make sure that only the Adapter checkbox is selected.
10. Select Updating Template as the Template Type.
11. Under VLANs, select the `DATA-Traffic` VLAN.
12. Set the `Data-Traffic` VLAN as the native VLAN.
13. Select vNIC Name for the CDN Source.
14. For MTU, enter 9000.
15. In the MAC Pool list, select MAC-Pool-B.
16. In the Network Control Policy list, select Enable-LLDP.
17. Click OK to create the vNIC template.

18. Click OK.

Create LAN Connectivity Policy for iSCSI Boot

To configure the necessary Infrastructure LAN Connectivity Policy, follow these steps:

1. In Cisco UCS Manager, click the LAN icon.
2. Expand Policies > root.
3. Right-click LAN Connectivity Policies.
4. Select Create LAN Connectivity Policy.
5. Enter `BM-NFS-FabA` as the name of the policy (to signify this policy utilizes NFS vNIC on Fabric-A).
6. Click the + Add button to add a vNIC.
7. In the Create vNIC dialog box, enter `00-MGMT` as the name of the vNIC.
8. Select the Use vNIC Template checkbox.
9. In the vNIC Template list, select BM-MGMT.
10. In the Adapter Policy list, select Linux.
11. Click OK to add this vNIC to the policy.



Create vNIC

Name : 00-MGMT

Use vNIC Template : ☑

Redundancy Pair : □

Peer Name :

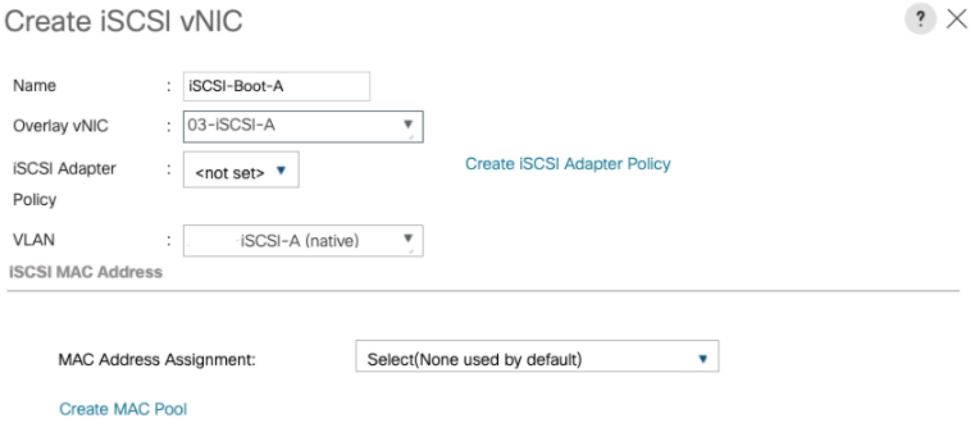
vNIC Template : BM-MGMT ▾

Adapter Performance Profile

Adapter Policy : Linux ▾

12. Click the + Add button to add another vNIC to the policy.
13. In the Create vNIC box, enter `01-NFS` as the name of the vNIC.
14. Select the Use vNIC Template checkbox.
15. In the vNIC Template list, select `BM-NFS-A`.
16. In the Adapter Policy list, select Linux.
17. Click OK to add the vNIC to the policy.
18. Click the + Add button to add another vNIC to the policy.
19. (Optional) In the Create vNIC box, enter `02-Traffic` as the name of the vNIC.
20. Select the Use vNIC Template checkbox.
21. In the vNIC Template list, select `BM-Traffic-B`.

22. In the Adapter Policy list, select Linux.
23. Click OK to add the vNIC to the policy.
24. Click + Add button to add another vNIC to the policy.
25. In the Create vNIC box, enter `03-iSCSI-A` as the name of the vNIC.
26. Select the Use vNIC Template checkbox.
27. In the vNIC Template list, select `BM-iSCSI-A`.
28. In the Adapter Policy list, select Linux.
29. Click OK to add the vNIC to the policy.
30. Click the + Add button to add another vNIC to the policy.
31. In the Create vNIC box, enter `03-iSCSI-B` as the name of the vNIC.
32. Select the Use vNIC Template checkbox.
33. In the vNIC Template list, select `BM-iSCSI-B`.
34. In the Adapter Policy list, select Linux.
35. Click OK to add the vNIC to the policy.
36. Expand the Add iSCSI vNICs section.
37. Click the + Add button (Under Add iSCSI vNICs) to add an iSCSI boot vNIC to the policy.
38. In the Create iSCSI vNIC box, enter `iSCSI-Boot-A` as the Name of the vNIC.
39. Select `03-iSCSI-A` for the Overlay vNIC.
40. Select the default iSCSI Adapter Policy.
41. iSCSI-A (native) should be selected as the VLAN.
42. Do not select anything for MAC Address Assignment.



Create iSCSI vNIC

Name : iSCSI-Boot-A

Overlay vNIC : 03-iSCSI-A

iSCSI Adapter Policy : <not set>

VLAN : iSCSI-A (native)

iSCSI MAC Address

MAC Address Assignment: Select(None used by default)

43. Click OK to add the vNIC to the policy.
44. Click the + Add button (Under Add iSCSI vNICs) to add another iSCSI boot vNIC to the policy.
45. In the Create iSCSI vNIC box, enter `iSCSI-Boot-B` as the name of the vNIC.
46. Select 04-iSCSI-B for the Overlay vNIC.

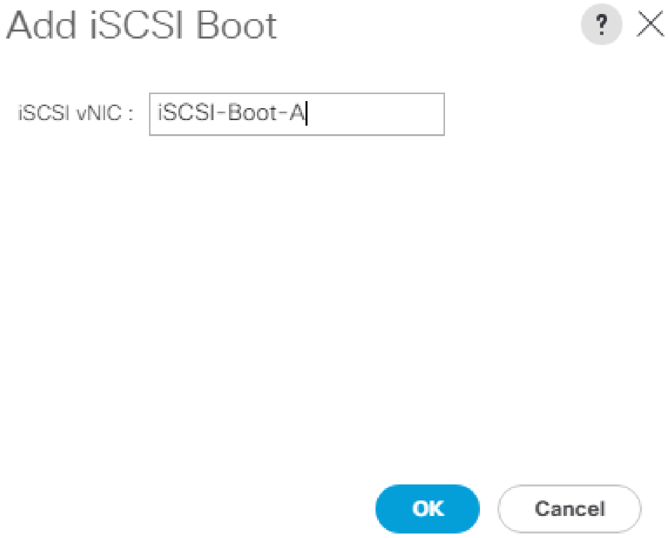
47. Select the default iSCSI Adapter Policy.
48. iSCSI-B (native) should be selected as the VLAN.
49. Do not select anything for MAC Address Assignment.
50. Click OK to add the vNIC to the policy.
51. Click OK, then OK again to create the LAN Connectivity Policy.

Create iSCSI Boot Policies

This procedure applies to a Cisco UCS environment in which two iSCSI logical interfaces (LIFs) are on storage cluster node 1 (iscsi_lif01a and iscsi_lif01b) and two iSCSI LIFs are on storage cluster node 2 (iscsi_lif02a and iscsi_lif02b). This boot policy sets the primary target to be iscsi_lif01a.

To create a boot policy for the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the Servers icon.
2. Expand Policies > root.
3. Right-click Boot Policies.
4. Select Create Boot Policy.
5. Enter `iSCSI-Boot` as the Name of the boot policy.
6. Optional: Enter a description for the boot policy.
7. Keep the Reboot on Boot Order Change option cleared.
8. Expand the Local Devices drop-down list and select **`Add Remote CD/DVD`**.
9. Expand the iSCSI vNICs drop-down list and select Add iSCSI Boot.
10. Enter `iSCSI-Boot-A` in the iSCSI vNIC field.



Add iSCSI Boot

iSCSI vNIC : iSCSI-Boot-A

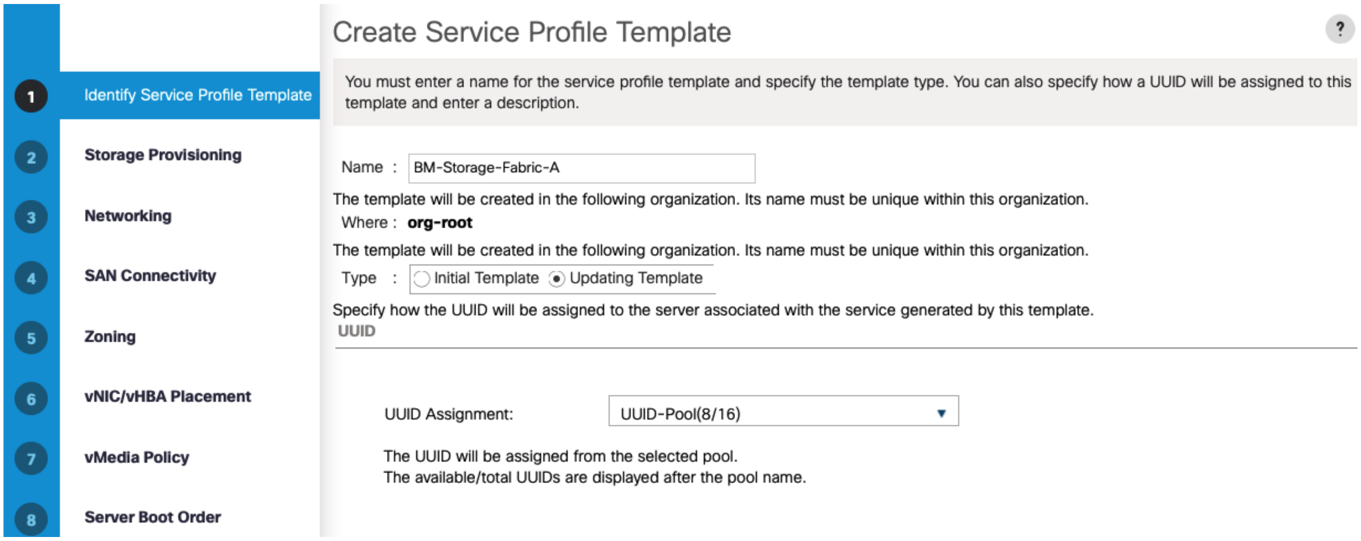
11. Click OK.
12. From the iSCSI vNICs drop-down list, select Add iSCSI Boot.

13. Enter `iSCSI-Boot-B` in the iSCSI vNIC field.
14. Click OK.
15. Click OK, then click OK again to create the boot policy.

Create Service Profile Template

To create the service profile template that utilizes Fabric A as the primary boot path, follow these steps:

1. In Cisco UCS Manager, click the Servers icon.
2. Expand Service Profile Templates > root.
3. Select and right-click root.
4. Select Create Service Profile Template to open the Create Service Profile Template wizard.
5. Enter `BM-Storage-FabricA` (to signify NFS storage access uses Fabric-A as primary path) as the name of the service profile template.
6. Select the Updating Template option.
7. Under UUID, select `UUID-Pool` as the UUID pool.



Create Service Profile Template

You must enter a name for the service profile template and specify the template type. You can also specify how a UUID will be assigned to this template and enter a description.

Name : BM-Storage-Fabric-A

The template will be created in the following organization. Its name must be unique within this organization.

Where : **org-root**

The template will be created in the following organization. Its name must be unique within this organization.

Type : ○ Initial Template ● Updating Template

Specify how the UUID will be assigned to the server associated with the service generated by this template.

UUID

UUID Assignment: UUID-Pool(8/16) ▾

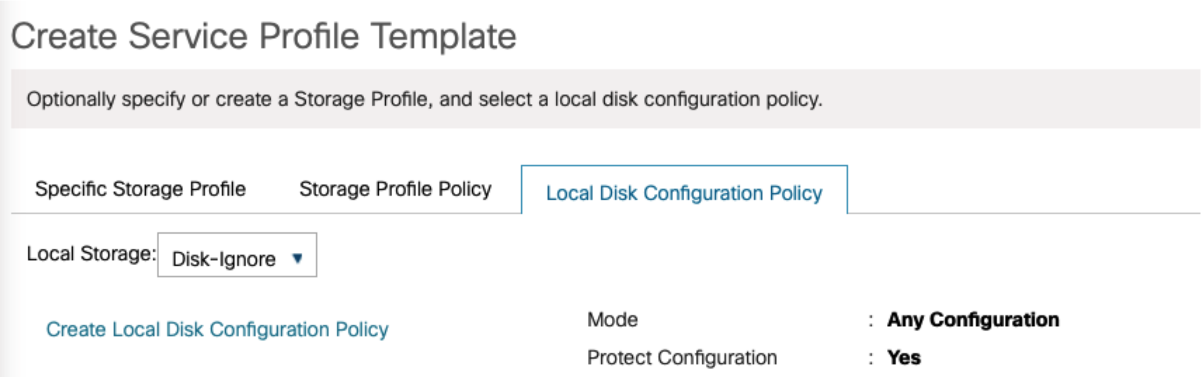
The UUID will be assigned from the selected pool.
The available/total UUIDs are displayed after the pool name.

8. Click Next.

Configure Storage Provisioning

To configure the storage provisioning, follow these steps:

1. Click the Local Disk Configuration Policy and select the `Disk-Ignore` Local Storage Policy.



Create Service Profile Template

Optionally specify or create a Storage Profile, and select a local disk configuration policy.

Specific Storage Profile
Storage Profile Policy
Local Disk Configuration Policy

Local Storage: Disk-Ignore ▾

Create Local Disk Configuration Policy

Mode : **Any Configuration**

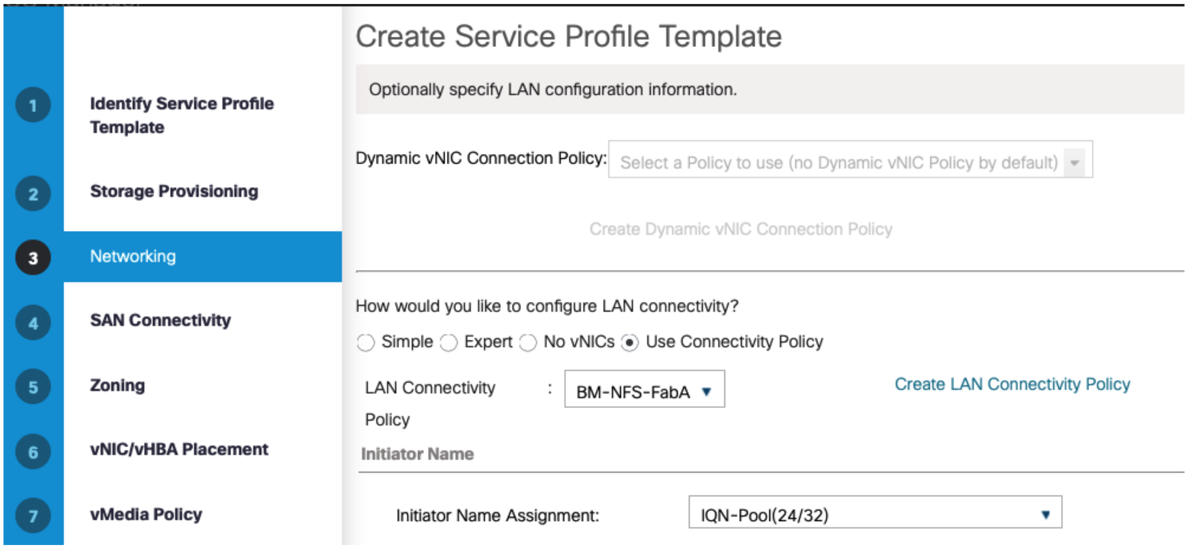
Protect Configuration : **Yes**

2. Click Next.

Configure Networking Options

To configure the networking options, follow these steps:

1. Keep the default setting for Dynamic vNIC Connection Policy.
2. Select the Use Connectivity Policy option to configure the LAN connectivity.
3. Select `BM-NFS-FabA` from the LAN Connectivity drop-down list.
4. Select `IQN-Pool` from the Initiator Name Assignment drop-down list.



Create Service Profile Template

Optionally specify LAN configuration information.

Dynamic vNIC Connection Policy: Select a Policy to use (no Dynamic vNIC Policy by default)

How would you like to configure LAN connectivity?

○ Simple ○ Expert ○ No vNICs ● Use Connectivity Policy

LAN Connectivity Policy : BM-NFS-FabA ▾

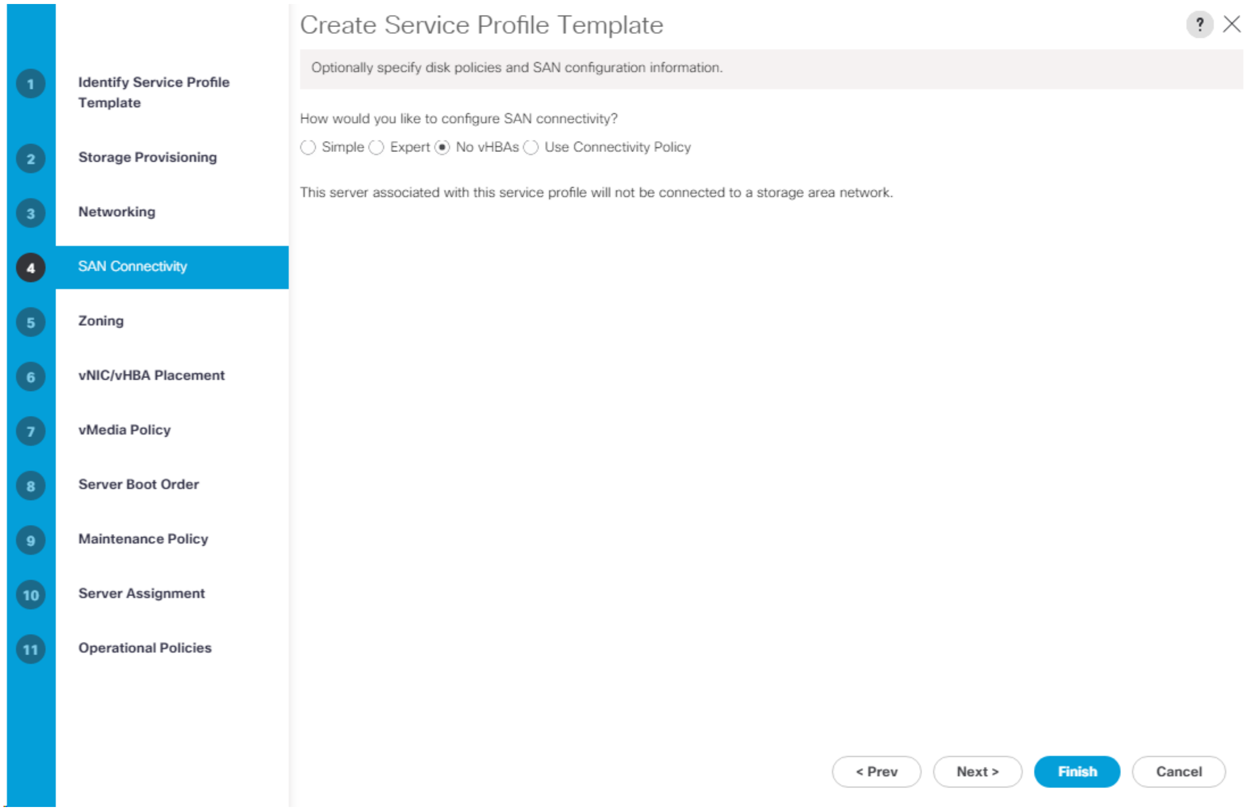
Initiator Name Assignment: IQN-Pool(24/32) ▾

5. Click Next.

Configure SAN Connectivity Options

To configure the SAN connectivity options, follow these steps:

1. Select the No vHBAs option for the “How would you like to configure SAN connectivity?” field.



Create Service Profile Template

Optionally specify disk policies and SAN configuration information.

How would you like to configure SAN connectivity?

○ Simple ○ Expert ● No vHBAs ○ Use Connectivity Policy

This server associated with this service profile will not be connected to a storage area network.

2. Click Next.

Configure Zoning Options

1. Ignore the Zoning Options and click Next.

Configure vNIC/HBA Placement

1. In the Select Placement list, leave the placement policy as “Let System Perform Placement.”
2. Click Next.

Configure vMedia Policy

1. Do not select a vMedia Policy.
2. Click Next.

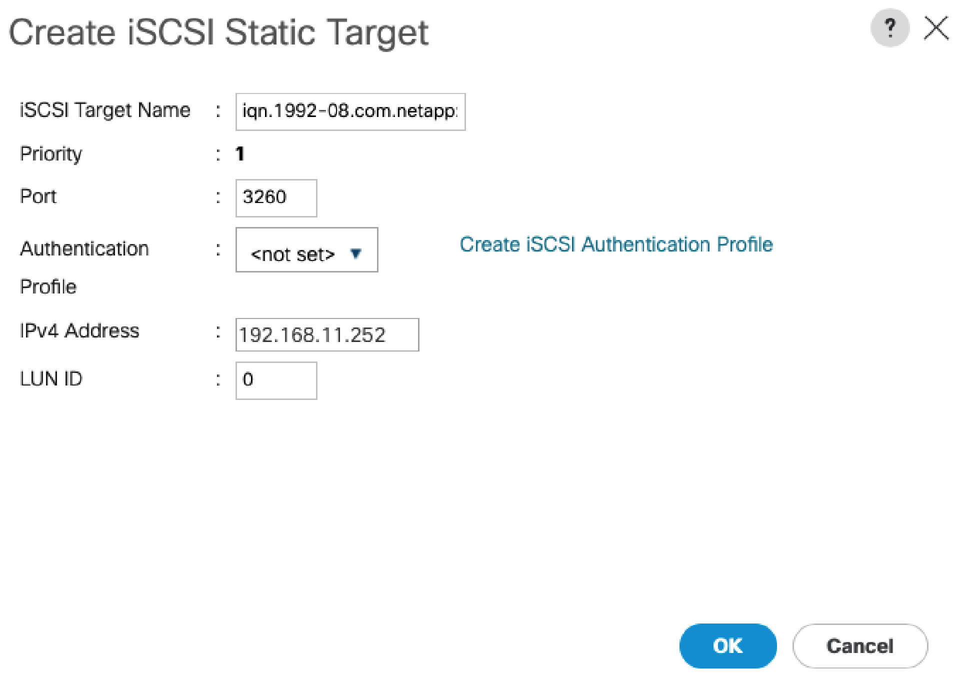
Configure Server Boot Order

1. Select `iSCSI-Boot` for Boot Policy.
2. Under Boot Order, expand Boot Order and iSCSI and select the iSCSI-Boot-A row.
3. Select the Set iSCSI Boot Parameters button.
4. Select `iSCSI-IP-Pool-A` for the Initiator IP Address Policy.
5. Scroll to the bottom of the window and click + Add.
6. Enter the IQN (Target Name) from the iSCSI Target Name.

To get this IQN, ssh into the storage cluster interface and type "`iscsi show`".

7. For IPv4 address, enter the IP address of `iscsi_a_lif02a` from the `AI-ML-SVM`.

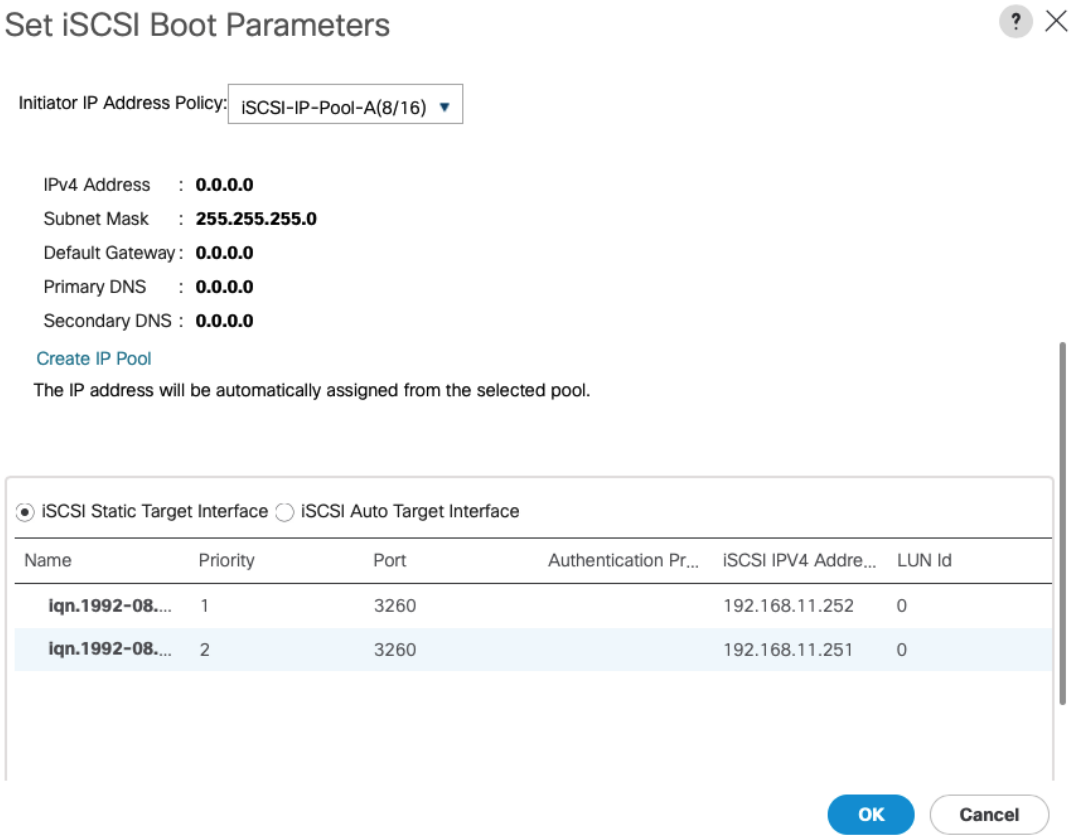
To get this IP, ssh into the storage cluster interface and type "`network interface show -vserver AI-ML-SVM`".



Create iSCSI Static Target

iSCSI Target Name :	iqn.1992-08.com.netapp:
Priority :	1
Port :	3260
Authentication Profile :	<not set>
IPv4 Address :	192.168.11.252
LUN ID :	0

8. Click OK to complete configuring the iSCSI target.
9. Click + Add to add a second target.
10. Enter the previously captured IQN (Target Name) again.
11. For IPv4 address, enter the IP address of `iscsi_a_lif01a` from the `AI-ML-SVM` SVM.
12. Click OK to complete configuring the iSCSI target.



Initiator IP Address Policy: iSCSI-IP-Pool-A(8/16)

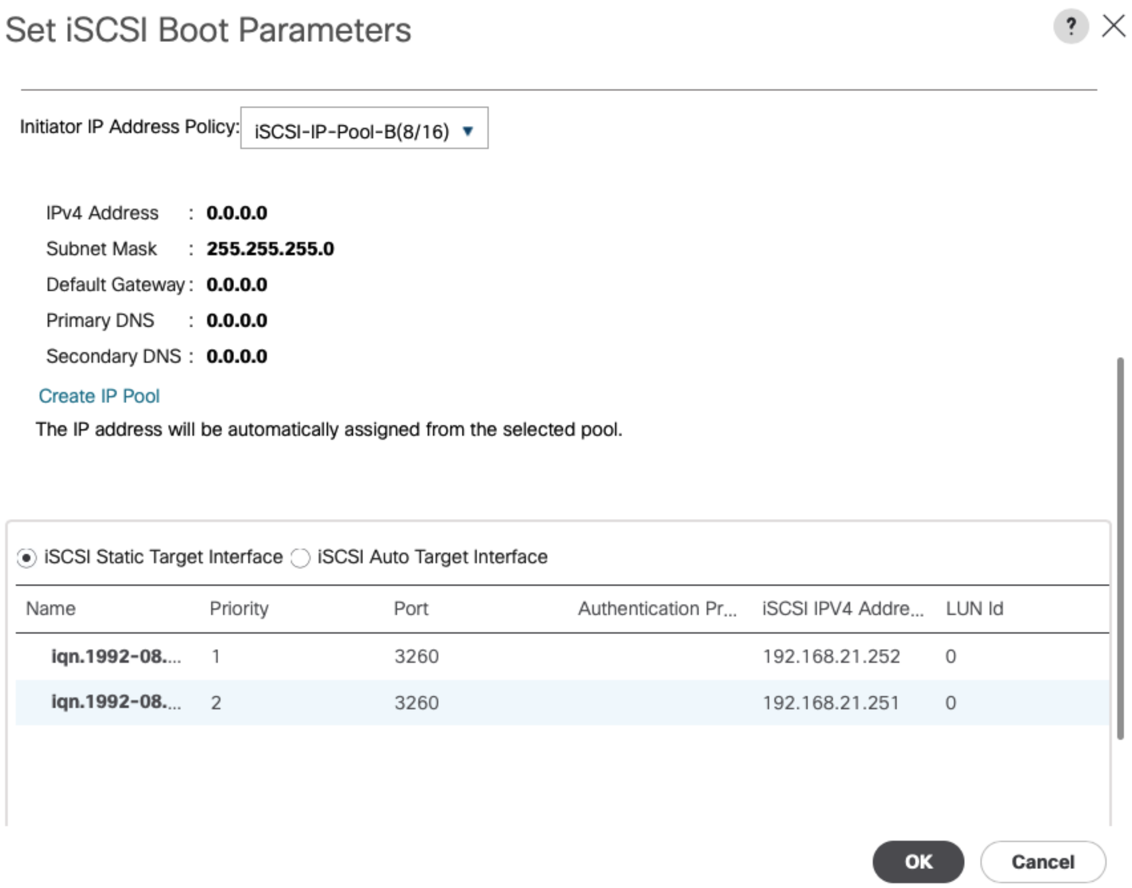
IPv4 Address : 0.0.0.0
Subnet Mask : 255.255.255.0
Default Gateway : 0.0.0.0
Primary DNS : 0.0.0.0
Secondary DNS : 0.0.0.0

The IP address will be automatically assigned from the selected pool.

● iSCSI Static Target Interface ○ iSCSI Auto Target Interface

Name	Priority	Port	Authentication Pr...	iSCSI IPV4 Addre...	LUN Id
iqn.1992-08....	1	3260		192.168.11.252	0
iqn.1992-08....	2	3260		192.168.11.251	0

13. Click OK to complete setting the iSCSI Boot Parameters for Fabric A Boot.
14. Under Boot Order, select the iSCSI-Boot-B row.
15. Select the Set iSCSI Boot Parameters button.
16. Select `iSCSI-IP-Pool-B` for the Initiator IP Address Policy.
17. Scroll to the bottom of the window and click + Add.
18. Enter the previously captured IQN (Target Name)
19. For IPv4 address, enter the IP address of `iscsi_lif02b` from the `AI-ML-SVM` SVM.
20. Click OK to complete configuring the iSCSI target.
21. Click + Add to add a second target.
22. Enter the previously captured IQN (Target Name.
23. For IPv4 address, enter the IP address of `iscsi_lif01b` from the `AI-ML-SVM` SVM.
24. Click OK to complete configuring the iSCSI target.



Set iSCSI Boot Parameters

Initiator IP Address Policy: iSCSI-IP-Pool-B(8/16)

IPv4 Address : 0.0.0.0
Subnet Mask : 255.255.255.0
Default Gateway : 0.0.0.0
Primary DNS : 0.0.0.0
Secondary DNS : 0.0.0.0

Create IP Pool

The IP address will be automatically assigned from the selected pool.

● iSCSI Static Target Interface ○ iSCSI Auto Target Interface

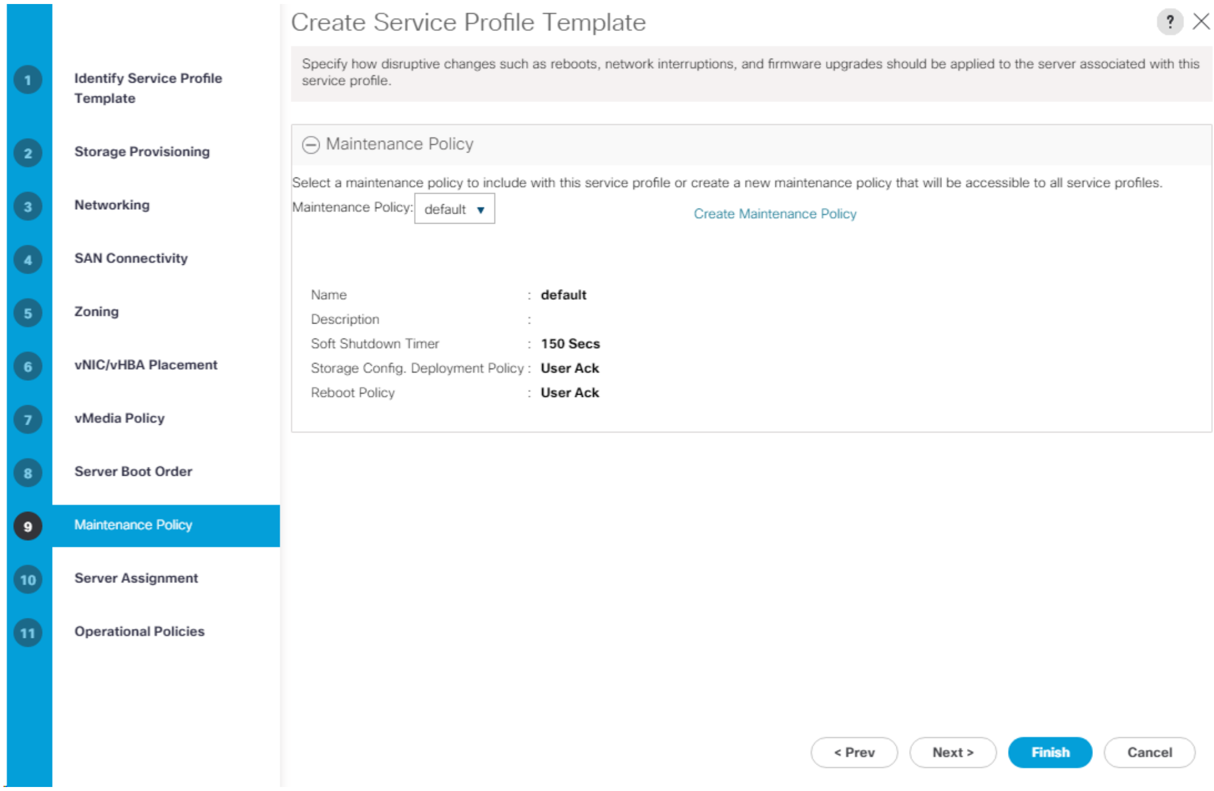
Name	Priority	Port	Authentication Pr...	iSCSI IPV4 Addre...	LUN Id
iqn.1992-08....	1	3260		192.168.21.252	0
iqn.1992-08....	2	3260		192.168.21.251	0

25. Click OK to complete setting the iSCSI Boot Parameters for Fabric B Boot.
26. Click Next.

Configure Maintenance Policy

To configure the Maintenance Policy, follow these steps:

1. Change the Maintenance Policy to default.



Create Service Profile Template

Specify how disruptive changes such as reboots, network interruptions, and firmware upgrades should be applied to the server associated with this service profile.

Maintenance Policy

Select a maintenance policy to include with this service profile or create a new maintenance policy that will be accessible to all service profiles.

Maintenance Policy: default

Create Maintenance Policy

Name : **default**
Description :
Soft Shutdown Timer : **150 Secs**
Storage Config. Deployment Policy : **User Ack**
Reboot Policy : **User Ack**

2. Click Next.

Configure Server Assignment

To configure server assignment, follow these steps:

1. In the Pool Assignment list, select the appropriate pool for the platform being deployed.
2. Expand Firmware Management at the bottom of the page and select the default policy.
3. Click Next.

Configure Operational Policies

To configure the operational policies, follow these steps:

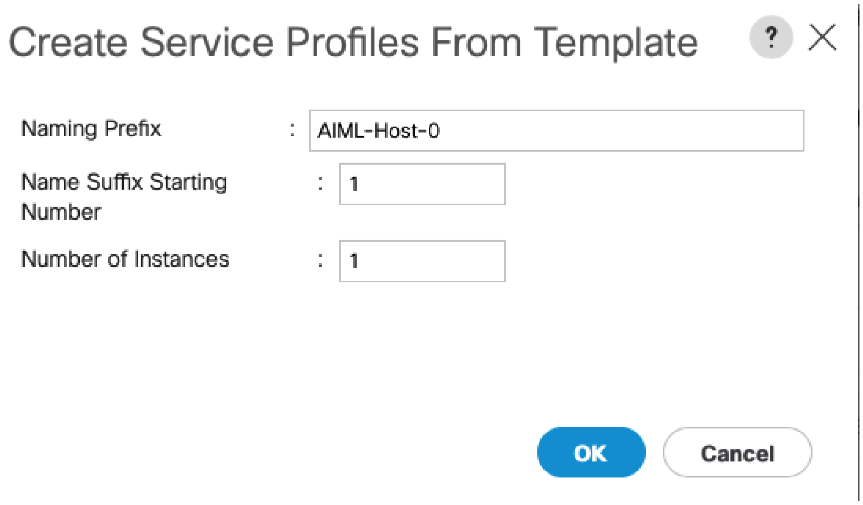
1. In the BIOS Policy list, select `AI-ML-Hosts`.
2. Expand Power Control Policy Configuration and select `No-Power-Cap` in the Power Control Policy list.
3. Click Finish to create the service profile template.
4. Click OK in the confirmation message.

Create Service Profiles

To create service profiles from the service profile template, follow these steps:

1. In the UCS Manager, click the Servers icon.
2. Select Service Profile Templates > root > Service Template BM-Storage-FabricA.

3. Right-click Service Template BM-Storage-FabricA and select Create Service Profiles from Template.
4. Enter `AIML-Host-0` as the service profile prefix.
5. Enter `1` as Name Suffix Starting Number.
6. Enter `<1>` as the Number of Instances.
7. Click OK to create the service profiles.



Create Service Profiles From Template

Naming Prefix	: AIML-Host-0
Name Suffix Starting Number	: 1
Number of Instances	: 1

8. Click OK in the confirmation message.

Storage Configuration – Boot LUNs

ONTAP Boot Storage Setup

Create igroups

Create igroups by entering the following commands from the storage cluster management node SSH connection:

```
igroup create -vserver AI-ML-SVM -igroup C480-AI-ML-01 -protocol iscsi -ostype linux -initiator
<C480ML_01_iscsi_iqn>

igroup create -vserver AI-ML-SVM -igroup C240-AI-ML-01 -protocol iscsi -ostype linux -initiator
<C240M5_01_iscsi_iqn>

igroup create -vserver AI-ML-SVM -igroup C220-AI-ML-01 -protocol iscsi -ostype linux -initiator
<C220M5_01_iscsi_iqn>

(Optional – for any additional servers)

igroup create -vserver AI-ML-SVM -igroup C480-AI-ML-02 -protocol iscsi -ostype linux -initiator
<C480ML_02_iscsi_iqn>

igroup create -vserver AI-ML-SVM -igroup C240-AI-ML-02 -protocol iscsi -ostype linux -initiator
<C240M5_02_iscsi_iqn>
```

To obtain the server IQN's, from Cisco UCS Manager, go to **Server** tab, click a Service Profile name and select **iSCSI vNICs** in the main window. Note the **IQN Name** (for example, iqn.2010-11.com.flexpod:ai-6454-host:5).

To view the three igroups just created, use the command lun igroup show:

```
igroup show -vserver AI-ML-SVM -protocol iscsi
```

Map Boot LUNs to igroups

From the storage cluster management SSH connection, enter the following commands:

```
lun mapping create -vserver AI-ML-SVM -path /vol/C480_boot_AI_ML/C480-AI-ML-01 -igroup C480-AI-ML-01 –
lun-id 0

lun mapping create -vserver AI-ML-SVM -path /vol/C480_boot_AI_ML/C240-AI-ML-01 -igroup C240-AI-ML-01 –
lun-id 0

lun mapping create -vserver AI-ML-SVM -path /vol/C480_boot_AI_ML/C220-AI-ML-01 -igroup C220-AI-ML-01 –
lun-id 0

(Optional – for any additional servers)

lun mapping create -vserver AI-ML-SVM -path /vol/C480_boot_AI_ML/C480-AI-ML-02 -igroup C480-AI-ML-02 –
lun-id 0

lun mapping create -vserver AI-ML-SVM -path /vol/C480_boot_AI_ML/C240-AI-ML-02 -igroup C240-AI-ML-02 –
lun-id 0
```

Bare Metal Server Setup and Configuration

Red Hat Enterprise Linux (RHEL) Bare Metal Installation

This section provides the instructions for installing and configuring RHEL 7.6 on Cisco UCS C220, C240 and C480 ML M5 servers. After the setup is completed, bare metal server(s) will be deployed with capability to download and run AI/ML container images with NVIDIA GPU Cloud (NGC). This guide explains downloading and running a CNN benchmark setup in a TensorFlow container. The procedure applies to all the Cisco UCS C-Series platforms since all the platforms are using Cisco 25Gbps VICs and NVIDIA GPUs.

Several methods exist for installing operating system on the servers. The procedure below focuses on using the built-in keyboard, video and mouse (KVM) console and mapped CD/DVD in Cisco UCS Manager to map remote installation media to individual servers.

Download RHEL 7.6 DVD ISO

If the RHEL DVD image has not been downloaded, follow these steps to download the ISO:

1. Click the following link RHEL 7.6 Binary DVD.
2. A user_id and password are required on the website (redhat.com) to download this software.
3. Download the .iso (rhel-server-7.6-x86_64-dvd.iso) file.

Log into Cisco UCS Manager

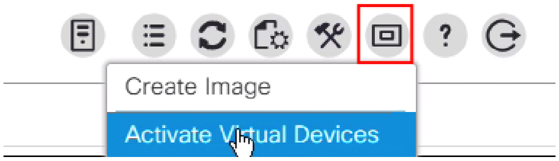
The Cisco UCS IP KVM enables the UCS administrator to begin the installation of the operating system (OS) through remote media. It is necessary to log in to the Cisco UCS environment to run the IP KVM. To do so, follow these steps:

1. Log into the Cisco UCS Manager using a web browser.
2. From the main menu, click Servers .
3. Select the Service Profile for the appropriate Cisco UCS C-series server.
4. On the right, under the General tab, click the **>>** to the right of KVM Console.
5. Follow the prompts to launch the KVM console.

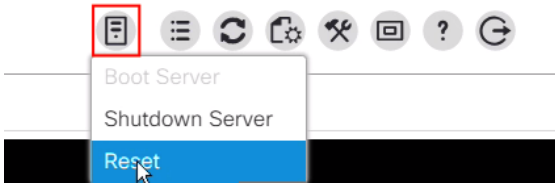
Operating System Installation

To prepare the server for the OS installation, follow these steps on each host:

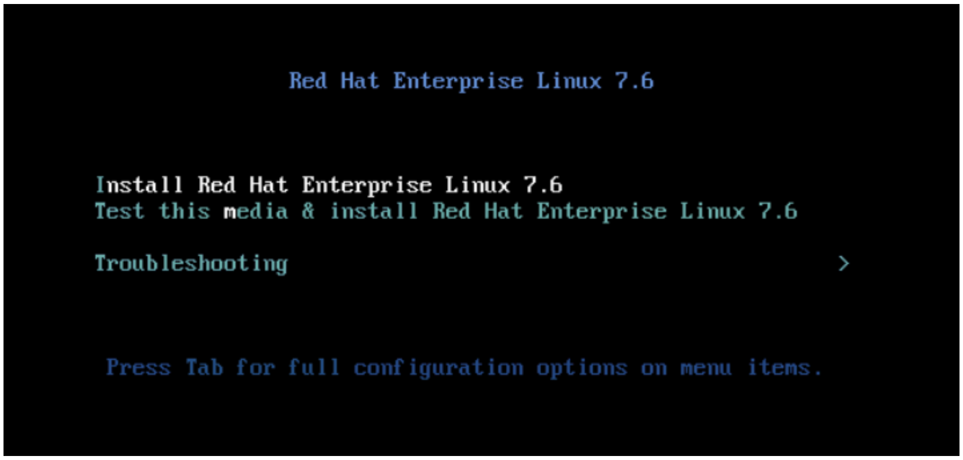
1. In the KVM window, click Virtual Media.
2. Click Activate Virtual Devices.



3. If prompted to accept an Unencrypted KVM session, accept as necessary.
4. Click Virtual Media and select Map CD/DVD.
5. Browse to the RHEL 7.6 ISO image file and click Open.
6. Click Map Device.
7. Boot the server by selecting Boot Server and clicking OK, then click OK two more times. If the system is already booted up to Shell> prompt, click Server Actions and click Reset. From the Server Reset options, select Power Cycle and click OK and then OK.

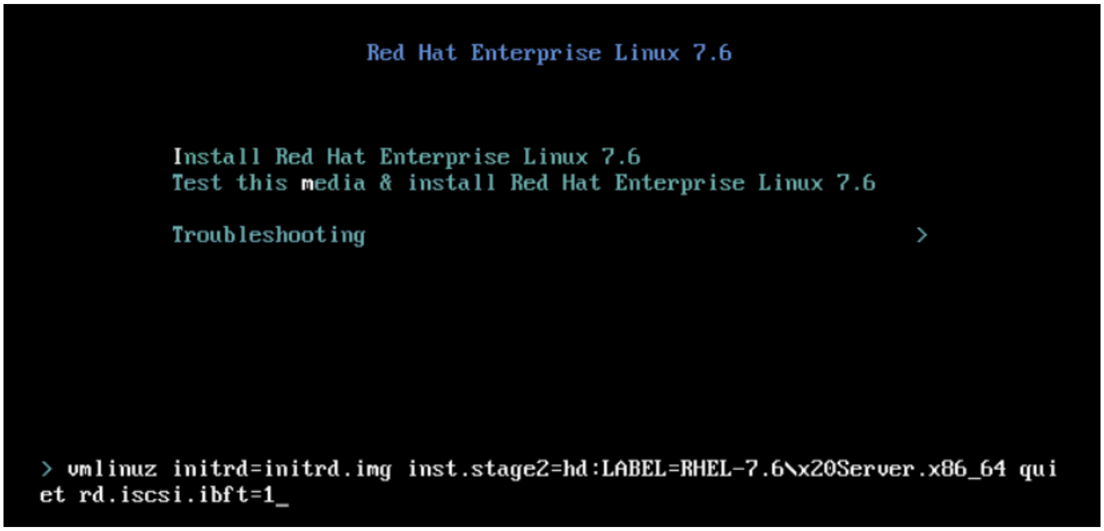


8. On reboot, the server detects the presence of the RHEL installation media. From the Installation menu, use arrow keys to select Install Red Hat Enterprise Linux 7.6. This should stop automatic boot countdown.



```
              Red Hat Enterprise Linux 7.6


   Install Red Hat Enterprise Linux 7.6
   Test this media & install Red Hat Enterprise Linux 7.6

   Troubleshooting                                      >



    Press Tab for full configuration options on menu items.
```

9. Press <TAB> for configuration options and add `rd.iscsi.ibft=1` to enable installation of RHEL on iSCSI boot LUN.


Red Hat Enterprise Linux 7.6
Install Red Hat Enterprise Linux 7.6
Test this media & install Red Hat Enterprise Linux 7.6
Troubleshooting >

```
> vmlinuz initrd=initrd.img inst.stage2=hd:LABEL=RHEL-7.6\x20Server.x86_64 qui
et rd.iscsi.ibft=1_
```

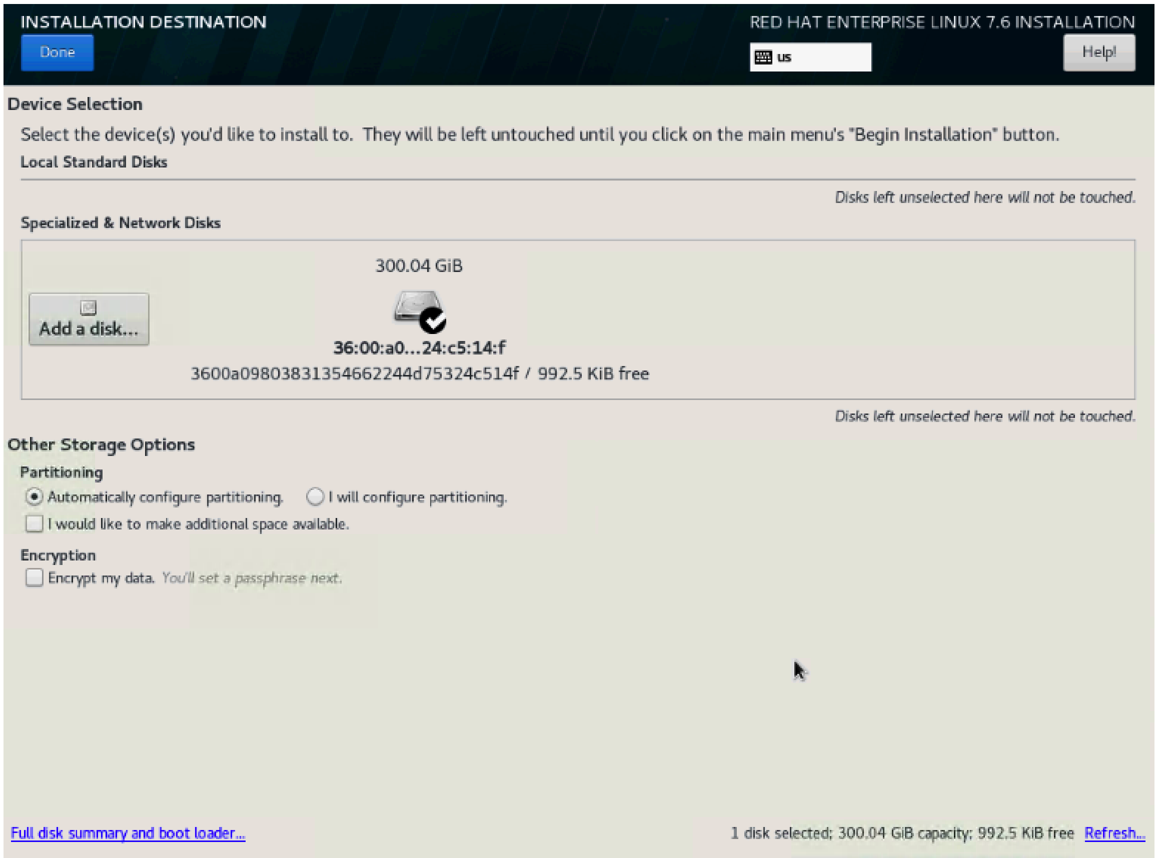
10. Press Enter to continue the boot process.
11. After the installer finishes loading, select the language and press Continue.
12. On the Installation Summary screen, leave the software selection to Minimal Install.

It might take a couple of minutes for the system to check the installation source. During this time, Installation Source will be grayed out. Wait for the system to load the menu items completely.

13. Click the Installation Destination to select the iSCSI boot LUN as installation disk.
14. If the system has local HDDs, these disks will be visible under Local Standard Disks.

If the iSCSI boot LUN does not appear under the Specialized & Network Disks, click Add a disk, select the LUN, and click Done.

15. The iSCSI LUN should now have a check mark to identify it as the installation disk.



INSTALLATION DESTINATION

Done

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

us

Help!

Device Selection

Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.

Local Standard Disks

Disks left unselected here will not be touched.

Specialized & Network Disks

Add a disk...

300.04 GiB

36:00:a0...24:c5:14:f

3600a0980383135466224 4d75324c514f / 992.5 KiB free

Disks left unselected here will not be touched.

Other Storage Options

Partitioning

● Automatically configure partitioning. ○ I will configure partitioning.

□ I would like to make additional space available.

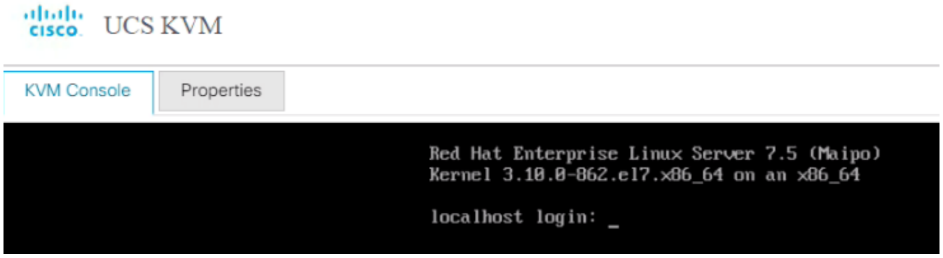
Encryption

□ Encrypt my data. *You'll set a passphrase next.*

Full disk summary and boot loader...

1 disk selected; 300.04 GiB capacity; 992.5 KiB free Refresh...

16. Leave Automatically configure partitioning checked and click Done.
17. Click Begin Installation to start RHEL installation.
18. Enter and confirm the root password and click Done.
19. (Optional) Create another user for accessing the system.
20. After the installation is complete, click in the Virtual Media button.
21. Click the CD/DVD – Mapped and click Unmap Drive in the pop-up window.
22. Click Virtual Media button again and click Deactivate.
23. Click Reboot to reboot the system. The system should now boot up with RHEL.



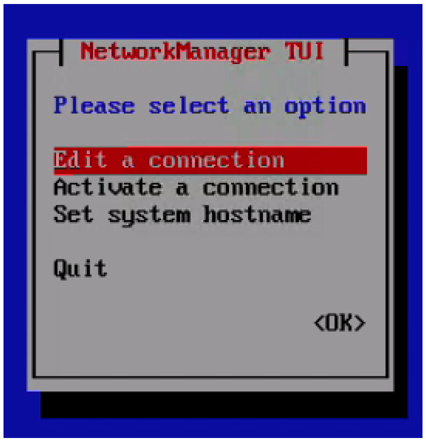
```
Red Hat Enterprise Linux Server 7.5 (Maipo)
Kernel 3.10.0-862.el7.x86_64 on an x86_64

localhost login: _
```

In some cases, the server does not reboot properly and seems to hang. Click the Server Actions button on top right and select Reset. Click OK and then select Power Cycle and click OK a couple of times to force a reboot.

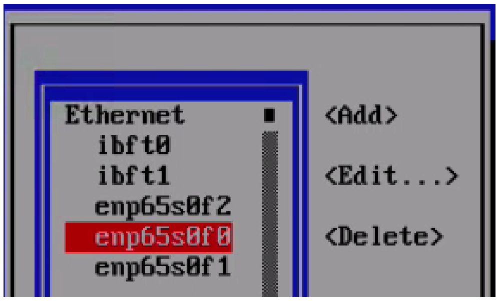
Network and Hostname Setup

Adding a management network for each RHEL host is necessary for remotely logging in and managing the host. To setup all the network interfaces and the hostname using the UCS KVM console, follow these steps:

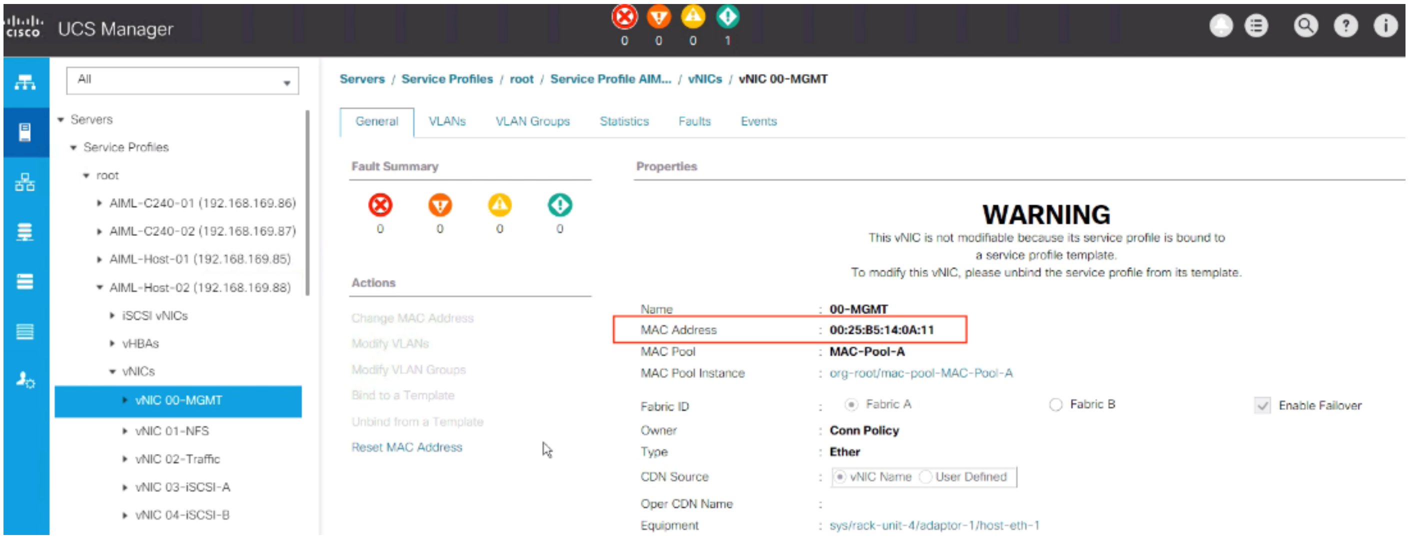
1. Log into the RHEL using the Cisco UCS KVM console and make sure the server has finished rebooting and login prompt is visible.
2. Log in as `root`, enter the password during the initial setup.
3. After logging on, type `nmtui` and press <Return>.
4. Using arrow keys, select Edit a connection and press <Return>.



5. In the connection list, ibft0 and ibft1 connections are iSCSI interfaces and should not be changed. Select the connection with the lowest ID “0” (enp65s0f0 in this example) and press <Return>.



6. As defined in the Cisco UCS Lan connectivity Policy, the first interface should be the management interface. This can be verified by going to Cisco UCS Manager and then Server > Service Profile > <Service Profile Name>. Expand the <Service Profile Name> and vNICs. Click the vNIC 00-MGMT and note the MAC address in the main window.



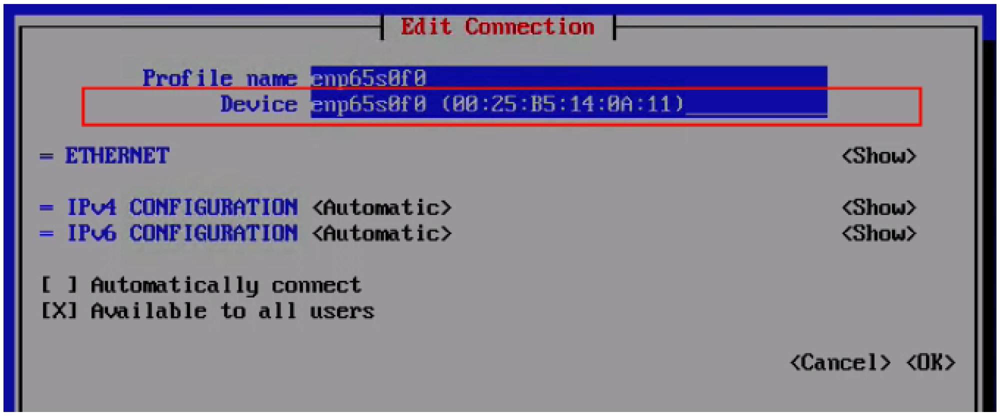
WARNING

This vNIC is not modifiable because its service profile is bound to a service profile template.

To modify this vNIC, please unbind the service profile from its template.

Name : 00-MGMT
MAC Address : 00:25:B5:14:0A:11
MAC Pool : MAC-Pool-A
MAC Pool Instance : org-root/mac-pool-MAC-Pool-A
Fabric ID : ● Fabric A ○ Fabric B ☑ Enable Failover
Owner : Conn Policy
Type : Ether
CDN Source : ● vNIC Name ○ User Defined
Oper CDN Name :
Equipment : sys/rack-unit-4/adaptor-1/host-eth-1

7. This MAC address should match the MAC address information in the KVM console.



```
                 Edit Connection

        Profile name enp65s0f0
              Device enp65s0f0 (00:25:B5:14:0A:11)

= ETHERNET                                        <Show>

= IPv4 CONFIGURATION <Automatic>                  <Show>
= IPv6 CONFIGURATION <Automatic>                  <Show>

[ ] Automatically connect
[X] Available to all users

                                          <Cancel> <OK>
```

8. After the connection is verified, in the KVM console. using arrow keys scroll down to IPv4 CONFIGURATION <Automatic> and press <Return>. Select Manual.

9. Scroll to <Show> next to IPv4 CONFIGURATION and press <Return>.

10. Scroll to <Add…> next to Addresses and enter the management IP address with a subnet mask in the following format: x.x.x.x/nn (for example, 192.168.169.85/24)

Remember to enter a subnet mask when entering the IP address. The system will accept an IP address without a subnet mask and then assign a subnet mask of /32 causing unnecessary issues.

11. Scroll down to Gateway and enter the gateway IP address.

12. Scroll down to <Add..> next to DNS server and add one or more DNS servers.

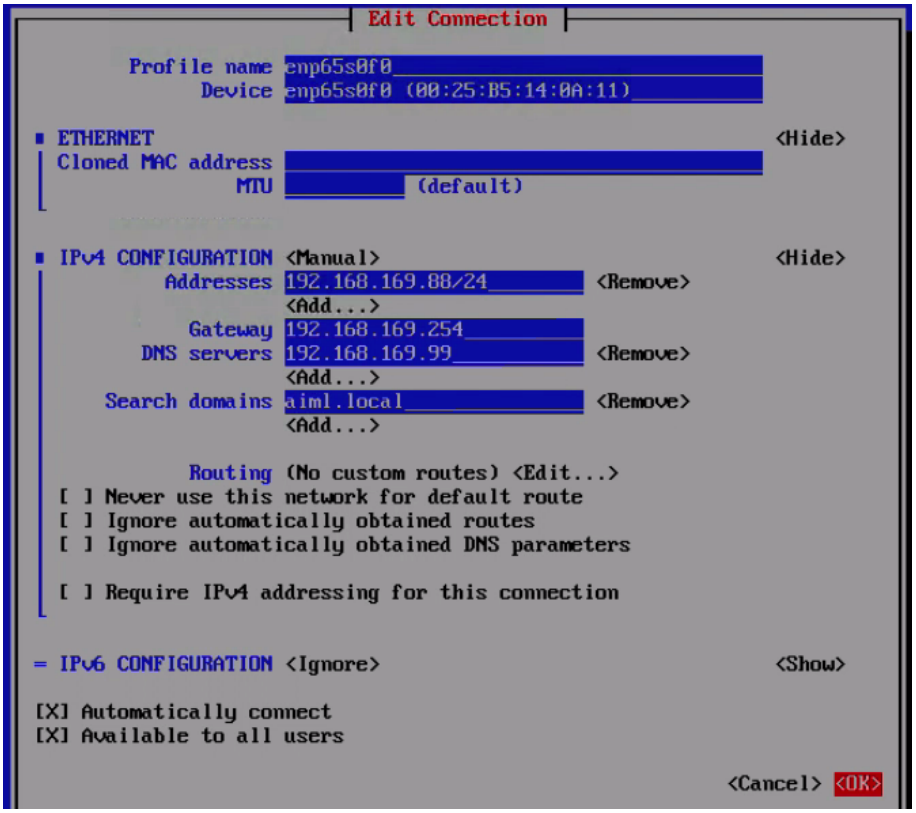
13. Scroll down to <Add…> next to Search Domains and add a domain (if applicable).

14. Scroll down to <Automatic> next to IPv6 CONFIGURATION and press <Return>.

15. Select Ignore and press <Return>.

16. Scroll down and Check Automatically connect.

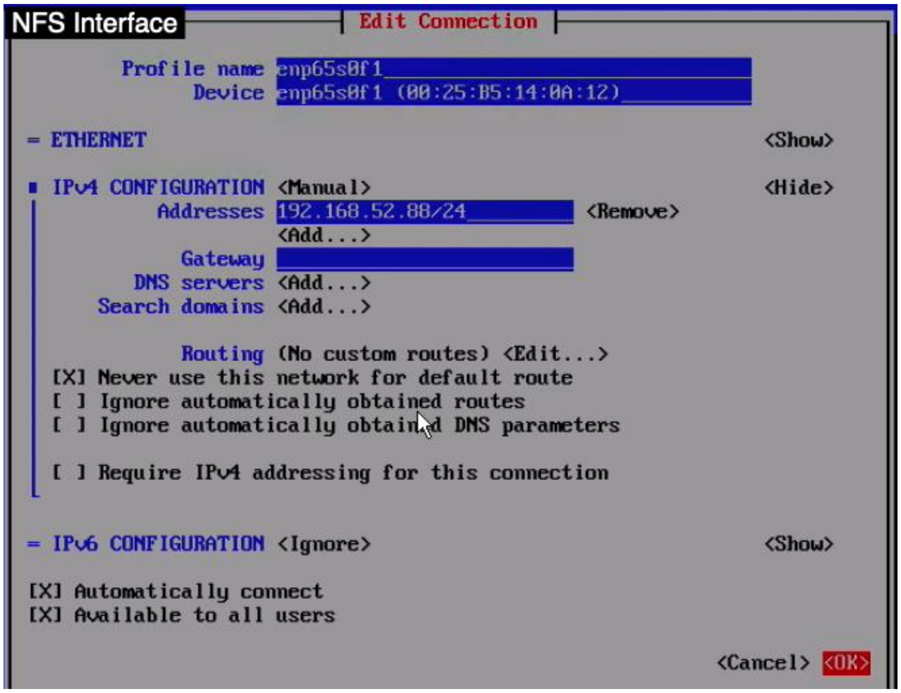
17. Scroll down to <OK> and press <Return>.



```
                         ┤ Edit Connection ├

            Profile name enp65s0f0_________________________________
                  Device enp65s0f0 (00:25:B5:14:0A:11)_____________

■ ETHERNET                                                         <Hide>
│ Cloned MAC address ______________________________________
│                MTU __________ (default)
└

■ IPv4 CONFIGURATION <Manual>                                      <Hide>
│          Addresses 192.168.169.88/24________ <Remove>
│                    <Add...>
│            Gateway 192.168.169.254__________
│        DNS servers 192.168.169.99___________ <Remove>
│                    <Add...>
│     Search domains aiml.local_______________ <Remove>
│                    <Add...>
│
│            Routing (No custom routes) <Edit...>
│ [ ] Never use this network for default route
│ [ ] Ignore automatically obtained routes
│ [ ] Ignore automatically obtained DNS parameters
│
│ [ ] Require IPv4 addressing for this connection
└

= IPv6 CONFIGURATION <Ignore>                                      <Show>

[X] Automatically connect
[X] Available to all users

                                                           <Cancel> <OK>
```

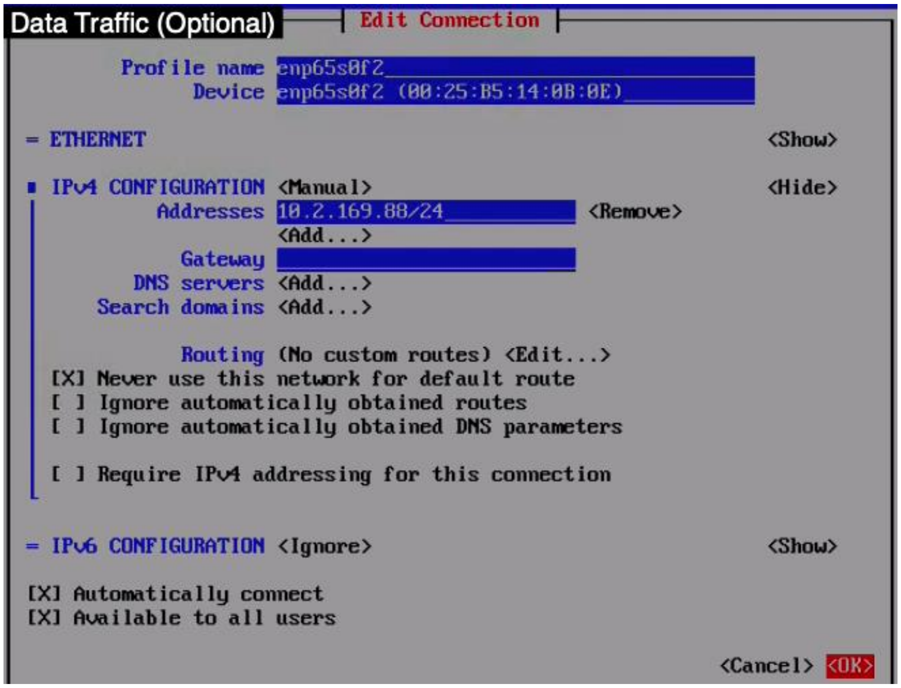
18. Repeat this procedure to setup NFS and Data Traffic (optional) interface.

For the NFS and Data Traffic interface(s), do not set a Gateway.



NFS Interface

```
                         ┤ Edit Connection ├

            Profile name enp65s0f1_________________________________
                  Device enp65s0f1 (00:25:B5:14:0A:12)_____________

= ETHERNET                                                         <Show>

■ IPv4 CONFIGURATION <Manual>                                      <Hide>
│          Addresses 192.168.52.88/24_________ <Remove>
│                    <Add...>
│            Gateway _________________________
│        DNS servers <Add...>
│     Search domains <Add...>
│
│            Routing (No custom routes) <Edit...>
│ [X] Never use this network for default route
│ [ ] Ignore automatically obtained routes
│ [ ] Ignore automatically obtained DNS parameters
│
│ [ ] Require IPv4 addressing for this connection
└

= IPv6 CONFIGURATION <Ignore>                                      <Show>

[X] Automatically connect
[X] Available to all users

                                                           <Cancel> <OK>
```



Data Traffic (Optional)

```
                                Edit Connection
         Profile name enp65s0f2
               Device enp65s0f2 (00:25:B5:14:0B:0E)

= ETHERNET                                                   <Show>

■ IPv4 CONFIGURATION <Manual>                                <Hide>
|           Addresses 10.2.169.88/24          <Remove>
|                     <Add...>
|             Gateway
|         DNS servers <Add...>
|      Search domains <Add...>
|
|             Routing (No custom routes) <Edit...>
| [X] Never use this network for default route
| [ ] Ignore automatically obtained routes
| [ ] Ignore automatically obtained DNS parameters
|
| [ ] Require IPv4 addressing for this connection
└

= IPv6 CONFIGURATION <Ignore>                                <Show>

[X] Automatically connect
[X] Available to all users

                                                     <Cancel> <OK>
```

19. Scroll down to <Back> and press <Return>.
20. From the main Network Manager TUI screen, scroll down to Set system hostname and press <Return>.
21. Enter the fully qualified domain name for the server and press <Return>.
22. Press <Return> and scroll down to Quit and press <Return> again.
23. At this point, the network services can be restarted for these changes to take effect. In the lab setup, the host was rebooted (type `reboot` and press <Return>) to ensure all the changes were properly saved and applied across the future server reboots.

RHEL Host Configuration

In this step, the following items are configured on the RHEL host:

- Setup Subscription Manager
- Enable repositories
- Install Net-Tools
- Install FTP
- Enable EPEL Repository
- Install NFS utilities and mount NFS share
- Update ENIC drivers
- Setup NTP
- Disable Firewall

Log into RHEL Host using SSH

To log in to the host(s), use an SSH client and connect to the previously configured management IP address of the host. Use the username: `root` and the <password> set up during RHEL installation.

Setup Subscription Manager

To setup the subscription manager, follow these steps:

1. To download and install packages, setup the subscription manager using valid redhat.com credentials:

```
[root@c480ml~]# subscription-manager register --username= <Name> --password=<Password> --auto-attach
Registering to: subscription.rhsm.redhat.com:443/subscription
The system has been registered with ID: <***>
The registered system name is: c480ml.aiml.local
```

2. To verify the subscription status:

```
[root@c480ml~]# subscription-manager attach --auto
Installed Product Current Status:
Product Name: Red Hat Enterprise Linux Server
Status:       Subscribed
```

Enable Repositories

To setup repositories for downloading various software packages, run the following command:

```
[root@c480ml~]# subscription-manager repos --enable="rhel-7-server-rpms" --enable="rhel-7-server-extras-
rpms"
Repository 'rhel-7-server-rpms' is enabled for this system.
Repository 'rhel-7-server-extras-rpms' is enabled for this system.
```

Install Net-Tools and Verify MTU

To enable helpful network commands (including ifconfig), install net-tools:

```
[root@c480ml~] yum install net-tools
Loaded plugins: product-id, search-disabled-repos, subscription-manager

<SNIP>

Installed:
  net-tools.x86_64 0:2.0-0.24.20131004git.el7

Complete!
```

Using the `ifconfig` command, verify the MTU is correctly set to 9000 on the NFS and (optional) Data-Traffic interfaces. If the MTU is not set correctly, modify the MTU and set it to 9000.

Install FTP

Install the FTP client to enable copying files to the host using ftp:

```
[root@c480ml~]# yum install ftp
Loaded plugins: product-id, search-disabled-repos, subscription-manager
epel/x86_64/metalink                                                           |  17 kB  00:00:00

<SNIP>

Installed:
  ftp.x86_64 0:0.17-67.el7

Complete!
```

Enable EPEL Repository

EPEL (Extra Packages for Enterprise Linux) is open source and free community-based repository project from Fedora team which provides 100 percent high quality add-on software packages for Linux distribution including RHEL. Some of the packages installed later in the setup require EPEL repository to be enabled. To enable the repository, run the following:

```
[root@c480ml~]# yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
Loaded plugins: product-id, search-disabled-repos, subscription-manager
epel-release-latest-7.noarch.rpm                                              |  15 kB  00:00:00
Examining /var/tmp/yum-root-HoB_fs/epel-release-latest-7.noarch.rpm: epel-release-7-11.noarch
Marking /var/tmp/yum-root-HoB_fs/epel-release-latest-7.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

======================================================================================================
 Package                Arch             Version        Repository                               Size
======================================================================================================
Installing:
 epel-release           noarch           7-11           /epel-release-latest-7.noarch            24 k

Transaction Summary
======================================================================================================
Install  1 Package

Total size: 24 k
Installed size: 24 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Warning: RPMDB altered outside of yum.
  Installing : epel-release-7-11.noarch                                                           1/1
  Verifying  : epel-release-7-11.noarch                                                           1/1

Installed:
  epel-release.noarch 0:7-11
```

Install NFS Utilities and Mount NFS Share

To mount NFS share on the host, NFS utilities need to be installed and the `/etc/fstab` file needs to be modified. To do so, follow these steps:

1. To install the `nfs-utils`:

```
 [root@c480ml~]# yum install nfs-utils
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
---> Package nfs-utils.x86_64 1:1.3.0-0.61.el7 will be installed
--> Processing Dependency: gssproxy >= 0.7.0-3 for package: 1:nfs-utils-1.3.0-0.61.el7.x86_64

<SNIP>

Installed:
  nfs-utils.x86_64 1:1.3.0-0.61.el7
```

```
Dependency Installed:
  gssproxy.x86_64 0:0.7.0-21.el7      keyutils.x86_64 0:1.5.8-3.el7        libbasicobjects.x86_64 
0:0.1.1-32.el7   libcollection.x86_64 0:0.7.0-32.el7
  libevent.x86_64 0:2.0.21-4.el7      libini_config.x86_64 0:1.3.1-32.el7  libnfsidmap.x86_64 0:0.25-
19.el7        libpath_utils.x86_64 0:0.2.1-32.el7
  libref_array.x86_64 0:0.1.5-32.el7  libtirpc.x86_64 0:0.2.4-0.15.el7     libverto-libevent.x86_64 
0:0.2.5-4.el7  quota.x86_64 1:4.01-17.el7
  quota-nls.noarch 1:4.01-17.el7      rpcbind.x86_64 0:0.2.0-47.el7        tcp_wrappers.x86_64 0:7.6-
77.el7

Complete!
```

2. Using text editor (such as vi), add the following line at the end of the `/etc/fstab` file:

```
<IP Address of NFS LIF>:/imagenet_dataset /mnt/imagenet nfs 
auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo =1800 0 0
```

where the `/imagenet_dataset` is the NFS mount point (as defined in NetApp).

To obtain the correct storage system mount point information, issue the following command: `volume show -vserver AI-ML-SVM -volume <imagenet_dataset>` and note down the `Junction Path`.

3. Verify that the updated `/etc/fstab` file looks similar to:

```
#
# /etc/fstab
# Created by anaconda on Wed Mar 27 18:33:36 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/rhel01-root /                       xfs     defaults,_netdev,_netdev 0 0
UUID=36f667cf-xxxxxxxxx /boot                   xfs     defaults,_netdev,_netdev,x-initrd.mount 0 0
/dev/mapper/rhel01-home /home                   xfs     defaults,_netdev,_netdev,x-initrd.mount 0 0
/dev/mapper/rhel01-swap swap                    swap    defaults,_netdev,x-initrd.mount 0 0
192.168.52.251:/imagenet_dataset /mnt/imagenet nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp 0 0
```

4. Issue the following commands to mount NFS at the following location: `/mnt/imagenet`

```
[root@c480ml~]# mkdir /mnt/imagenet
[root@c480ml~]# mount /mnt/imagenet
```

5. To verify that the mount was successful:

```
[root@c480ml~]# mount | grep imagenet
192.168.52.251:/imagenet_dataset on /mnt/imagenet type nfs 
(rw,noatime,vers=3,rsize=65536,wsize=65536,namlen=255,hard,nolock,proto=tcp,timeo=600,retrans=2,sec=sys,m
ountaddr=192.168.52.251,mountvers=3,mountport=635,mountproto=tcp,local_lock=all,addr=192.168.52.251)
```

Update ENIC Drivers

To update the ENIC drivers, follow these steps:

1. To check the current version of the enic driver, issue the following command:

```
[root@c480ml~]# modinfo enic
filename:       /lib/modules/3.10.0-862.el7.x86_64/kernel/drivers/net/ethernet/cisco/enic/enic.ko.xz
version:        2.3.0.42

<SNIP>
```

2. To update the driver, download the ISO image of UCS-Rack Linux drivers from Cisco UCS C-Series UCS-Managed Server software here: https://software.cisco.com/download/home/286318809/type/283853158/release/4.0(4) .
3. Provide the cisco.com login credentials and download the following file: ucs-cxxx-drivers-linux.4.0.4.iso.
4. Mount the ISO file on your PC and browse to the following folder: Network > Cisco > VIC > RHEL > RHEL7.6 and copy the file kmod-enic-3.2.210.18-738.12.rhel7u6.x86_64.rpm to the RHEL server using ftp or sftp. In the lab, this file was copied to the `/root` directory of the server.
5. Issue the following command to update and verify the drivers.

Reboot the host after the update completes successfully.

```
[root@c480ml~]# rpm -ivh /root/kmod-enic-3.2.210.18-738.12.rhel7u6.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1: kmod-enic-3.2.210.18-738.12.rhel7################################# [100%]

[root@c480ml~]# modinfo enic
filename:       /lib/modules/3.10.0-957.el7.x86_64/extra/enic/enic.ko
version:        3.2.210.18-738.12
license:        GPL v2
<SNIP>

[root@cc480ml~]#reboot
```

Setup NTP

To setup NTP, follow these steps:

1. To synchronize the host time to an NTP server, install NTP package:

```
[root@c480ml~]# yum install  ntp
<SNIP>
```

2. If the default NTP servers defined in `/etc/ntp.conf` file are not reachable or to add additional local NTP servers, modify the `/etc/ntp.conf` file (using a text editor such as vi) and add the server(s) as shown below:

“#” in front of a server name or IP address signifies that the server information is commented out and will not be used

```
[root@c480ml~]# more /etc/ntp.conf | grep server
server 192.168.169.1 iburst
# server 0.rhel.pool.ntp.org iburst
# server 1.rhel.pool.ntp.org iburst
# server 2.rhel.pool.ntp.org iburst
# server 3.rhel.pool.ntp.org iburst
```

3. To verify the time is setup correctly, use the `date` command:

```
[root@c480ml~]# date
Wed May  8 12:17:48 EDT 2019
```

Disable Firewall

To make sure the installation goes smoothly, Linux firewall and the Linux kernel security module (SEliniux) is disabled. To do so, follow these steps:

Customer Linux server management team should review and enable these security modules with appropriate settings once the installation is complete.

1. To disable Firewall:

```
[root@c480ml~]# systemctl stop firewalld
[root@c480ml~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multiuser.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
```

2. To disable SELinux:

```
[root@c480ml~]# sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
[root@c480ml~]# setenforce 0
```

3. Reboot the host:

```
[root@c480ml~]# reboot
```

Disable IPv6 (Optional)

If IPv6 addresses are not being used in the customer environment, IPv6 can be disabled on the RHEL host:

```
[root@c480ml~]# echo 'net.ipv6.conf.all.disable_ipv6 = 1' >> /etc/sysctl.conf
[root@c480ml~]# echo 'net.ipv6.conf.default.disable_ipv6 = 1' >> /etc/sysctl.conf
[root@c480ml~]# echo 'net.ipv6.conf.lo.disable_ipv6 = 1' >> /etc/sysctl.conf
[root@c480ml~]# reboot
```

NVIDIA and CUDA Drivers Installation

In this step, the following components will be installed:

- Install Kernel Headers
- Install gcc
- Install wget
- Install DKMS
- Install NVIDIA Driver
- Install CUDA Driver
- Install CUDA Toolkit

Install Kernel Headers

To install the Kernel Headers, run the following commands:

```
[root@c480ml~]# uname -r
3.10.0-957.el7.x86_64
[root@c480ml~]# yum install kernel-devel-$(uname -r) kernel-headers-$(uname -r)
```

```
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies

<SNIP>

Installed:
  kernel-devel.x86_64 0:3.10.0-957.el7                    kernel-headers.x86_64 0:3.10.0-957.el7

Dependency Installed:
  perl.x86_64 4:5.16.3-294.el7_6                              perl-Carp.noarch 0:1.26-244.el7
  perl-Encode.x86_64 0:2.51-7.el7                             perl-Exporter.noarch 0:5.68-3.el7
  perl-File-Path.noarch 0:2.09-2.el7                          perl-File-Temp.noarch 0:0.23.01-3.el7
  perl-Filter.x86_64 0:1.49-3.el7                             perl-Getopt-Long.noarch 0:2.40-3.el7
  perl-HTTP-Tiny.noarch 0:0.033-3.el7                         perl-PathTools.x86_64 0:3.40-5.el7
  perl-Pod-Escapes.noarch 1:1.04-294.el7_6                    perl-Pod-Perldoc.noarch 0:3.20-4.el7
  perl-Pod-Simple.noarch 1:3.28-4.el7                         perl-Pod-Usage.noarch 0:1.63-3.el7
  perl-Scalar-List-Utils.x86_64 0:1.27-248.el7                perl-Socket.x86_64 0:2.010-4.el7
  perl-Storable.x86_64 0:2.45-3.el7                           perl-Text-ParseWords.noarch 0:3.29-4.el7
  perl-Time-HiRes.x86_64 4:1.9725-3.el7                       perl-Time-Local.noarch 0:1.2300-2.el7
  perl-constant.noarch 0:1.27-2.el7                           perl-libs.x86_64 4:5.16.3-294.el7_6
  perl-macros.x86_64 4:5.16.3-294.el7_6                       perl-parent.noarch 1:0.225-244.el7
  perl-podlators.noarch 0:2.5.1-3.el7                         perl-threads.x86_64 0:1.87-4.el7
  perl-threads-shared.x86_64 0:1.43-6.el7

Complete!
```

Install gcc

To install the C compiler, run the following command:

```
 [root@c480ml~]# yum install gcc-4.8.5

<SNIP>

Installed:
  gcc.x86_64 0:4.8.5-39.el7

Dependency Installed:
  cpp.x86_64 0:4.8.5-39.el7        glibc-devel.x86_64 0:2.17-292.el7      glibc-headers.x86_64 0:2.17-
292.el7
  libmpc.x86_64 0:1.0.1-3.el7      mpfr.x86_64 0:3.1.1-4.el7

Dependency Updated:
  glibc.x86_64 0:2.17-292.el7         glibc-common.x86_64 0:2.17-292.el7       libgcc.x86_64 0:4.8.5-
39.el7
  libgomp.x86_64 0:4.8.5-39.el7

Complete!

[root@rhel-tmpl ~]# yum install gcc-c++
Loaded plugins: product-id, search-disabled-repos, subscription-manager

<SNIP>

Installed:
  gcc-c++.x86_64 0:4.8.5-39.el7

Dependency Installed:
  libstdc++-devel.x86_64 0:4.8.5-39.el7

Dependency Updated:
  libstdc++.x86_64 0:4.8.5-39.el7

Complete!
```

Install wget

To install wget for downloading files from Internet, run the following command:

```
 [root@c480ml~]# yum install wget
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies

<SNIP>

Installed:
  wget.x86_64 0:1.14-18.el7_6.1
```

Install DKMS

To enable Dynamic Kernel Module Support, run the following command:

```
# [root@c480ml~]# yum install dkms
Loaded plugins: product-id, search-disabled-repos, subscription-manager
epel/x86_64/metalink                                                            |  17 kB  00:00:00

<SNIP>

Installed:
  dkms.noarch 0:2.7.1-1.el7

Dependency Installed:
  elfutils-libelf-devel.x86_64 0:0.176-2.el7                    zlib-devel.x86_64 0:1.2.7-18.el7

Dependency Updated:
  elfutils-libelf.x86_64 0:0.176-2.el7                      elfutils-libs.x86_64 0:0.176-2.el7

Complete!
```

Install NVIDIA Driver

To install NVIDIA Driver on the RHEL host, follow these steps:

1. Download the driver from NVIDIA using wget:

```
[root@c480ml~]# wget http://us.download.nvidia.com/tesla/418.40.04/nvidia-diag-driver-local-repo-rhel7-
418.40.04-1.0-1.x86_64.rpm
--2019-09-18 15:05:26--  http://us.download.nvidia.com/tesla/418.40.04/nvidia-diag-driver-local-repo-
rhel7-418.40.04-1.0-1.x86_64.rpm
Resolving us.download.nvidia.com (us.download.nvidia.com)... 192.229.211.70,
2606:2800:21f:3aa:dcf:37b:1ed6:1fb
<SNIP>
```

2. Verify the file was successfully downloaded:

```
[root@c480ml~]# ls -l
-rw-r--r--  1 root root 161530513 Mar 15  2019 nvidia-diag-driver-local-repo-rhel7-418.40.04-1.0-
1.x86_64.rpm
```

3. Change file mode and install the downloaded rpm:

```
[root@c480ml~]# rpm -ivh nvidia-diag-driver-local-repo-rhel7-418.40.04-1.0-1.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:nvidia-diag-driver-local-repo-rhe################################# [100%]
```

4. Clean the yum cache files:

```
[root@c480ml~]# yum clean all
```

```
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Cleaning repos: epel nvidia-diag-driver-local-418.40.04 rhel-7-server-extras-rpms rhel-7-server-rpms
Cleaning up everything
Maybe you want: rm -rf /var/cache/yum, to also free up space taken by orphaned data from disabled or
removed repos
```

Install CUDA Driver

To install the CUDA driver, follow these steps:

1. Enable RHEL Server Options repository. This repository is needed to install vulkan-filesystem, a requirement for CUDA drivers:

```
[root@c480ml ~]# subscription-manager repos --enable rhel-7-server-optional-rpms
Repository 'rhel-7-server-optional-rpms' is enabled for this system.
```

2. Install the cuda-drivers using the following command:

```
[root@c480ml~]# yum install cuda-drivers
Loaded plugins: product-id, search-disabled-repos, subscription-manager
rhel-7-server-extras-rpms                                                   | 3.4 kB  00:00:00
rhel-7-server-optional-rpms                                                 | 3.2 kB  00:00:00
<SNIP>

Transaction Summary
=========================================================================================================
==============
Install  1 Package (+59 Dependent packages)

Total download size: 149 M
Installed size: 405 M
Is this ok [y/d/N]: y
<SNIP>

Installed:
  cuda-drivers.x86_64 0:418.40.04-1

Dependency Installed:
  adwaita-cursor-theme.noarch 0:3.28.0-1.el7                  adwaita-icon-theme.noarch 0:3.28.0-1.el7
  at-spi2-atk.x86_64 0:2.26.2-1.el7                           at-spi2-core.x86_64 0:2.28.0-1.el7
  cairo-gobject.x86_64 0:1.15.12-4.el7                        colord-libs.x86_64 0:1.3.4-1.el7
  dconf.x86_64 0:0.28.0-4.el7                                 dkms-nvidia.x86_64 3:418.40.04-1.el7
  glib-networking.x86_64 0:2.56.1-1.el7                       gnutls.x86_64 0:3.3.29-9.el7_6
  gsettings-desktop-schemas.x86_64 0:3.28.0-2.el7             gtk3.x86_64 0:3.22.30-3.el7
  json-glib.x86_64 0:1.4.2-2.el7                              lcms2.x86_64 0:2.6-3.el7
  libX11-devel.x86_64 0:1.6.7-2.el7                           libXau-devel.x86_64 0:1.0.8-2.1.el7
  libXdmcp.x86_64 0:1.1.2-6.el7                               libXfont2.x86_64 0:2.0.3-1.el7
  libXtst.x86_64 0:1.2.3-1.el7                                libepoxy.x86_64 0:1.5.2-1.el7
  libfontenc.x86_64 0:1.1.3-3.el7                             libglvnd-gles.x86_64 1:1.0.1-
0.8.git5baa1e5.el7
  libglvnd-opengl.x86_64 1:1.0.1-0.8.git5baa1e5.el7           libgusb.x86_64 0:0.2.9-1.el7
  libmodman.x86_64 0:2.0.1-8.el7                              libproxy.x86_64 0:0.4.11-11.el7
  libsoup.x86_64 0:2.62.2-2.el7                               libusbx.x86_64 0:1.0.21-1.el7
  libva-vdpau-driver.x86_64 0:0.7.4-19.el7                    libwayland-cursor.x86_64 0:1.15.0-1.el7
  libwayland-egl.x86_64 0:1.15.0-1.el7                        libxcb-devel.x86_64 0:1.13-1.el7
  libxkbcommon.x86_64 0:0.7.1-3.el7                           libxkbfile.x86_64 0:1.0.9-3.el7
  mesa-filesystem.x86_64 0:18.3.4-5.el7                       nettle.x86_64 0:2.7.1-8.el7
  nvidia-driver.x86_64 3:418.40.04-4.el7                      nvidia-driver-NVML.x86_64 3:418.40.04-4.el7
  nvidia-driver-NvFBCOpenGL.x86_64 3:418.40.04-4.el7          nvidia-driver-cuda.x86_64 3:418.40.04-4.el7
  nvidia-driver-cuda-libs.x86_64 3:418.40.04-4.el7            nvidia-driver-devel.x86_64 3:418.40.04-
4.el7
  nvidia-driver-libs.x86_64 3:418.40.04-4.el7                 nvidia-libXNVCtrl.x86_64 3:418.40.04-1.el7
  nvidia-libXNVCtrl-devel.x86_64 3:418.40.04-1.el7            nvidia-modprobe.x86_64 3:418.40.04-1.el7
  nvidia-persistenced.x86_64 3:418.40.04-1.el7                nvidia-settings.x86_64 3:418.40.04-1.el7
  nvidia-xconfig.x86_64 3:418.40.04-1.el7                     ocl-icd.x86_64 0:2.2.12-1.el7
  opencl-filesystem.noarch 0:1.0-5.el7                        rest.x86_64 0:0.8.1-2.el7
  trousers.x86_64 0:0.3.14-2.el7                              vulkan-filesystem.noarch 0:1.1.97.0-1.el7
  xkeyboard-config.noarch 0:2.24-1.el7                        xorg-x11-proto-devel.noarch 0:2018.4-1.el7
  xorg-x11-server-Xorg.x86_64 0:1.20.4-7.el7                  xorg-x11-server-common.x86_64 0:1.20.4-
7.el7
  xorg-x11-xkb-utils.x86_64 0:7.7-14.el7

Complete!
```

3. Blacklist the Nouveau Driver by opening the `/etc/modprobe.d/blacklist-nouveau.conf` in a text editor (for example vi) and adding following commands:

```
blacklist nouveau
options nouveau modeset=0
```

4. Verify the contents of the file. If the file does not exist, create the file and add the configuration lines.

```
 [root@c480ml modprobe.d]# more /etc/modprobe.d/blacklist-nouveau.conf
# RPM Fusion blacklist for nouveau driver - you need to run as root:
# dracut -f /boot/initramfs-$(uname -r).img $(uname -r)
# if nouveau is loaded despite this file.
blacklist nouveau
options nouveau modeset=0
```

5. Regenerate the kernel initramfs and reboot the system:

```
[root@c480ml~]# dracut --force
[root@c480ml~]# reboot
```

If the nouveau kernel module is not disabled, the NVIDIA kernel module will not load properly

6. Verify the correct GPU type is reported using the following command:

```
[root@c480ml~]# nvidia-smi --query-gpu=gpu_name --format=csv,noheader --id=0 | sed -e 's/ /-/g'
Tesla-V100-SXM2-32GB
```

Install CUDA Toolkit

To install the CUDA toolkit, follow these steps:

1. Download CUDA driver version 10.1 Update 2 from NVIDIA website using wget:

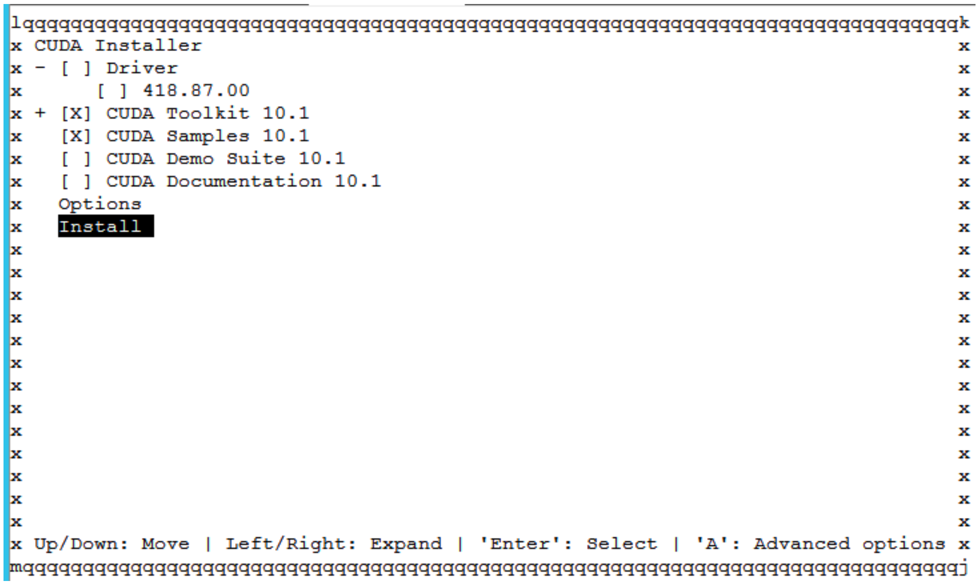
```
[root@c480ml~]# wget
http://developer.download.nvidia.com/compute/cuda/10.1/Prod/local_installers/cuda_10.1.243_418.87.00_linu
x.run
--2019-09-18 16:23:05--  Resolving developer.download.nvidia.com (developer.download.nvidia.com)...
192.229.211.70, 2606:2800:21f:3aa:dcf:37b:1ed6:1fb
Connecting to developer.download.nvidia.com (developer.download.nvidia.com)|192.229.211.70|:80...
connected.
HTTP request sent, awaiting response... 200 OK

<SNIP>

2019-09-18 16:23:46 (69.1 MB/s) - ‘cuda-repo-rhel7-10-1-local-10.1.243-418.87.00-1.0-1.x86_64.rpm’ saved
[2660351598/2660351598]
```

At the time of writing this document, driver version 418.87 is not supported by Cisco HCL therefore the previously installed driver version 418.40 is preserved when installing CUDA toolkit.

2. Install the CUDA 10.1 Tool Kit without updating the NVIDIA driver:

```
[root@c480ml~]# sh cuda_10.1.243_418.87.00_linux.run
```

3. From the text menu, using arrow keys, select Continue and press Enter.
4. Type accept to accept the end user license agreement and press Enter.
5. Using arrow keys and space bar, deselect Driver.
6. Optionally, deselect CUDA Demo Suite 10.1 and CUDA Documentation.



```
CUDA Installer
- [ ] Driver
     [ ] 418.87.00
+ [X] CUDA Toolkit 10.1
  [X] CUDA Samples 10.1
  [ ] CUDA Demo Suite 10.1
  [ ] CUDA Documentation 10.1
  Options
  Install

Up/Down: Move | Left/Right: Expand | 'Enter': Select | 'A': Advanced options
```

7. Select Install and press Enter.

```
===========
= Summary =
===========

Driver:   Not Selected
Toolkit:  Installed in /usr/local/cuda-10.1/
Samples:  Installed in /root/

Please make sure that
 -   PATH includes /usr/local/cuda-10.1/bin
 -   LD_LIBRARY_PATH includes /usr/local/cuda-10.1/lib64, or, add /usr/local/cuda-10.1/lib64 to
/etc/ld.so.conf and run ldconfig as root

To uninstall the CUDA Toolkit, run cuda-uninstaller in /usr/local/cuda-10.1/bin

Please see CUDA_Installation_Guide_Linux.pdf in /usr/local/cuda-10.1/doc/pdf for detailed information on
setting up CUDA.
***WARNING: Incomplete installation! This installation did not install the CUDA Driver. A driver of
version at least 418.00 is required for CUDA 10.1 functionality to work.
To install the driver using this installer, run the following command, replacing <CudaInstaller> with the
name of this run file:
    sudo <CudaInstaller>.run --silent --driver

Logfile is /var/log/cuda-installer.log
```

8. Reboot the server (reboot).
9. When the server is back up, issue the `nvidia-smi` command to verify all the GPUs are visible.

The output of nvidia-smi is different depending on the difference in number and types of GPUs.

Cisco UCS C220 M5 with T4 GPUs

```
[root@c220-1 ~]# nvidia-smi

+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.87.00    Driver Version: 418.87.00    CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:5E:00.0 Off |                    0 |
| N/A   51C    P0    27W /  70W |      0MiB / 15079MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            Off  | 00000000:D8:00.0 Off |                    0 |
| N/A   51C    P0    29W /  70W |      0MiB / 15079MiB |      4%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Cisco UCS C240 M5

```
[root@c240~]# nvidia-smi
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.40.04    Driver Version: 418.40.04    CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  Off  | 00000000:5E:00.0 Off |                    0 |
| N/A   40C    P0    40W / 250W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-PCIE...  Off  | 00000000:86:00.0 Off |                    0 |
| N/A   39C    P0    38W / 250W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Cisco UCS C480 ML M5

```
[root@c480ml~]# nvidia-smi

+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.40.04    Driver Version: 418.40.04    CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  Off  | 00000000:1B:00.0 Off |                    0 |
| N/A   44C    P0    60W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  Off  | 00000000:1C:00.0 Off |                    0 |
| N/A   44C    P0    59W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-SXM2...  Off  | 00000000:42:00.0 Off |                    0 |
| N/A   45C    P0    59W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-SXM2...  Off  | 00000000:43:00.0 Off |                    0 |
| N/A   45C    P0    58W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   4  Tesla V100-SXM2...  Off  | 00000000:89:00.0 Off |                    0 |
| N/A   43C    P0    56W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   5  Tesla V100-SXM2...  Off  | 00000000:8A:00.0 Off |                    0 |
| N/A   43C    P0    60W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   6  Tesla V100-SXM2...  Off  | 00000000:B2:00.0 Off |                    0 |
| N/A   43C    P0    57W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   7  Tesla V100-SXM2...  Off  | 00000000:B3:00.0 Off |                    0 |
| N/A   43C    P0    64W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

10. Modify path variables by typing the following lines at the shell prompt and also adding them to `.bashrc`:

```
export PATH=/usr/local/cuda-10.1/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-10.1/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

11. Verify the PATH variables:

```
 [root@c480ml~]# echo $PATH
/usr/local/cuda-10.1/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
[root@c480ml~]# echo $LD_LIBRARY_PATH
/usr/local/cuda-10.1/lib64
```

12. Add the following line to `/etc/ld.so.conf` file:

```
/usr/local/cuda-10.1/lib64
```

13. Verify the `/etc/ld.so.conf` file configuration:

```
[root@c480ml-2 ~]# more /etc/ld.so.conf
include ld.so.conf.d/*.conf
/usr/local/cuda-10.1/lib64
```

14. Execute the following command:

```
[root@c480ml~]# ldconfig
```

15. Verify that CUDA version is 10.1:

```
[root@c480ml~]# cat /usr/local/cuda/version.txt
CUDA Version 10.1.243
```

Verify the NVIDIA and CUDA Installation

Use the various commands shown below to verify the system is properly setup with CUDA and NVIDIA drivers and the GPUs are correctly identified. These commands will show slightly different output depending on the server (and GPU).

Verify CUDA Driver

To verify the CUDA driver, run a device query as shown below:

```
[root@c480ml~]# cd /usr/local/cuda-10.1/samples/1_Utilities/deviceQuery
[root@c480ml deviceQuery]# make

<SNIP>
```

Cisco UCS C220 M5 with T4

```
[root@c220 deviceQuery]# ./deviceQuery
./deviceQuery Starting...

CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 2 CUDA Capable device(s)

Device 0: "Tesla T4"
  CUDA Driver Version / Runtime Version          10.1 / 10.1
  CUDA Capability Major/Minor version number:    7.5
  Total amount of global memory:                 15080 MBytes (15812263936 bytes)
  (40) Multiprocessors, ( 64) CUDA Cores/MP:     2560 CUDA Cores
  GPU Max Clock rate:                            1590 MHz (1.59 GHz)
  Memory Clock rate:                             5001 Mhz
  Memory Bus Width:                              256-bit
  L2 Cache Size:                                 4194304 bytes

<SNIP>

> Peer access from Tesla T4 (GPU0) -> Tesla T4 (GPU1) : Yes
> Peer access from Tesla T4 (GPU1) -> Tesla T4 (GPU0) : Yes

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 10.1, CUDA Runtime Version = 10.1, NumDevs = 2
Result = PASS
```

Cisco UCS C240 M5

```
[root@c240 deviceQuery]# ./deviceQuery
./deviceQuery Starting...

 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 2 CUDA Capable device(s)

Device 0: "Tesla V100-PCIE-32GB"
  CUDA Driver Version / Runtime Version          10.1 / 10.1
  CUDA Capability Major/Minor version number:    7.0
  Total amount of global memory:                 32480 MBytes (34058272768 bytes)
  (80) Multiprocessors, ( 64) CUDA Cores/MP:     5120 CUDA Cores
  GPU Max Clock rate:                            1380 MHz (1.38 GHz)
  Memory Clock rate:                             877 Mhz
  Memory Bus Width:                              4096-bit
  L2 Cache Size:                                 6291456 bytes

<SNIP>

> Peer access from Tesla V100-PCIE-32GB (GPU0) -> Tesla V100-PCIE-32GB (GPU1) : Yes
> Peer access from Tesla V100-PCIE-32GB (GPU1) -> Tesla V100-PCIE-32GB (GPU0) : Yes

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 10.1, CUDA Runtime Version = 10.1, NumDevs = 2
Result = PASS
```

Cisco UCS C480 ML M5

```
[root@c480ml deviceQuery]# ./deviceQuery
./deviceQuery Starting...
CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 8 CUDA Capable device(s)

Device 0: "Tesla V100-SXM2-32GB"
  CUDA Driver Version / Runtime Version          10.1 / 10.1
  CUDA Capability Major/Minor version number:    7.0
  Total amount of global memory:                 32480 MBytes (34058272768 bytes)
  (80) Multiprocessors, ( 64) CUDA Cores/MP:     5120 CUDA Cores
  GPU Max Clock rate:                            1530 MHz (1.53 GHz)
  Memory Clock rate:                             877 Mhz
  Memory Bus Width:                              4096-bit
  L2 Cache Size:                                 6291456 bytes

<SNIP>

> Peer access from Tesla V100-SXM2-32GB (GPU7) -> Tesla V100-SXM2-32GB (GPU5) : Yes
> Peer access from Tesla V100-SXM2-32GB (GPU7) -> Tesla V100-SXM2-32GB (GPU6) : Yes

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 10.1, CUDA Runtime Version = 10.1, NumDevs = 8
Result = PASS
```

Verify NVIDIA Driver

To verify the NVIDIA driver, follow these steps:

1. Install pciutils:

```
 [root@c480ml~]# yum install pciutils
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
---> Package pciutils.x86_64 0:3.5.1-3.el7 will be installed
--> Finished Dependency Resolution

<SNIP>

Installed:
  pciutils.x86_64 0:3.5.1-3.el7

Complete!
```

2. Run the following commands to verify the NVIDIA information:

```
[root@c480ml-2~]# dmesg |grep NVRM
[   14.682164] NVRM: loading NVIDIA UNIX x86_64 Kernel Module  418.40.04  Fri Mar 15 00:59:12 CDT 2019
```

```
[root@c480ml~]# modinfo nvidia
filename:       /lib/modules/3.10.0-862.el7.x86_64/extra/nvidia.ko.xz
alias:          char-major-195-*
version:        418.40.04
supported:      external
license:        NVIDIA
retpoline:      Y
rhelversion:    7.5
srcversion:     86171E965AC9C3AD399B033
alias:          pci:v000010DEd00000E00sv*sd*bc04sc80i00*
alias:          pci:v000010DEd*sv*sd*bc03sc02i00*
alias:          pci:v000010DEd*sv*sd*bc03sc00i00*
depends:        ipmi_msghandler,i2c-core
vermagic:       3.10.0-862.el7.x86_64 SMP mod_unload modversions
parm:           NvSwitchRegDwords:NvSwitch regkey (charp)
parm:           NVreg_Mobile:int
parm:           NVreg_ResmanDebugLevel:int
parm:           NVreg_RmLogonRC:int
```

```
parm:           NVreg_ModifyDeviceFiles:int
parm:           NVreg_DeviceFileUID:int
parm:           NVreg_DeviceFileGID:int
parm:           NVreg_DeviceFileMode:int
parm:           NVreg_UpdateMemoryTypes:int
parm:           NVreg_InitializeSystemMemoryAllocations:int
parm:           NVreg_UsePageAttributeTable:int
parm:           NVreg_MapRegistersEarly:int
parm:           NVreg_RegisterForACPIEvents:int
parm:           NVreg_CheckPCIConfigSpace:int
parm:           NVreg_EnablePCIeGen3:int
parm:           NVreg_EnableMSI:int
parm:           NVreg_TCEBypassMode:int
parm:           NVreg_EnableStreamMemOPs:int
parm:           NVreg_EnableBacklightHandler:int
parm:           NVreg_RestrictProfilingToAdminUsers:int
parm:           NVreg_EnableUserNUMAManagement:int
parm:           NVreg_MemoryPoolSize:int
parm:           NVreg_KMallocHeapMaxSize:int
parm:           NVreg_VMallocHeapMaxSize:int
parm:           NVreg_IgnoreMMIOCheck:int
parm:           NVreg_NvLinkDisable:int
parm:           NVreg_RegistryDwords:charp
parm:           NVreg_RegistryDwordsPerDevice:charp
parm:           NVreg_RmMsg:charp
parm:           NVreg_GpuBlacklist:charp
parm:           NVreg_AssignGpus:charp
```

```
[root@c480ml~]# lspci | grep -i nvidia
1b:00.0 3D controller: NVIDIA Corporation Device 1db5 (rev a1)
1c:00.0 3D controller: NVIDIA Corporation Device 1db5 (rev a1)
42:00.0 3D controller: NVIDIA Corporation Device 1db5 (rev a1)
43:00.0 3D controller: NVIDIA Corporation Device 1db5 (rev a1)
89:00.0 3D controller: NVIDIA Corporation Device 1db5 (rev a1)
8a:00.0 3D controller: NVIDIA Corporation Device 1db5 (rev a1)
b2:00.0 3D controller: NVIDIA Corporation Device 1db5 (rev a1)
b3:00.0 3D controller: NVIDIA Corporation Device 1db5 (rev a1)
```

Setup NVIDIA Docker

The steps in this section explain how to install and run NVIDIA Docker containers on RHEL VMs and the bare-metal servers. Before starting NVIDIA Docker installation, verify the subscription-manager registration has been completed and the correct repositories have been added. To do so, follow these steps:

1. Verify the Subscription Manager status:

```
[root@c480ml~]# subscription-manager attach --auto
Installed Product Current Status:
Product Name: Red Hat Enterprise Linux Server
Status:       Subscribed
```

2. Verify the following Repos have been added:

```
[root@c480ml~]# subscription-manager repos --enable="rhel-7-server-rpms" --enable="rhel-7-server-extras-
rpms"
Repository 'rhel-7-server-rpms' is enabled for this system.
Repository 'rhel-7-server-extras-rpms' is enabled for this system.
```

3. Install yum-utils to enable yum-config-manager:

```
[root@c480ml~]# yum install yum-utils
Loaded plugins: product-id, search-disabled-repos, subscription-manager

<SNIP>

Installed:
  yum-utils.noarch 0:1.1.31-52.el7

Dependency Installed:
  python-chardet.noarch 0:2.2.1-3.el7                      python-kitchen.noarch 0:1.1.1-5.el7

Complete!
```

4. Enable docker-ce repo using yum-config-manager:

```
[root@c480ml~]# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: product-id, subscription-manager
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
```

5. Install container-selinux*, a dependency for docker-ce:

```
[root@c480ml~]# yum install container-selinux*
Loaded plugins: product-id, search-disabled-repos, subscription-manager
docker-ce-stable                                                                         | 3.5 kB
00:00:00
(1/2): docker-ce-stable/x86_64/updateinfo                                                |   55 B
00:00:00
(2/2): docker-ce-stable/x86_64/primary_db                                                |  26 kB
00:00:00

<SNIP>

Installed:
  container-selinux.noarch 2:2.107-3.el7

Dependency Installed:
  audit-libs-python.x86_64 0:2.8.5-4.el7 checkpolicy.x86_64 0:2.5-8.el7             libcgroup.x86_64
0:0.41-21.el7
  libsemanage-python.x86_64 0:2.5-14.el7 policycoreutils-python.x86_64 0:2.5-33.el7 python-IPy.noarch
0:0.75-6.el7
```

```
  setools-libs.x86_64 0:3.3.8-4.el7

Dependency Updated:
  audit.x86_64 0:2.8.5-4.el7                          audit-libs.x86_64 0:2.8.5-4.el7
  libselinux.x86_64 0:2.5-14.1.el7                    libselinux-python.x86_64 0:2.5-14.1.el7
  libselinux-utils.x86_64 0:2.5-14.1.el7              libsemanage.x86_64 0:2.5-14.el7
  libsepol.x86_64 0:2.5-10.el7                        policycoreutils.x86_64 0:2.5-33.el7
  selinux-policy.noarch 0:3.13.1-252.el7.1            selinux-policy-targeted.noarch 0:3.13.1-252.el7.1

Complete!
```

6. Install docker-ce using the following command:

```
[root@c480ml~]# yum install docker-ce
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
Loaded plugins: product-id, search-disabled-repos, subscription-manager

<SNIP>

Installed:
  docker-ce.x86_64 3:19.03.2-3.el7

Dependency Installed:
  containerd.io.x86_64 0:1.2.6-3.3.el7                     docker-ce-cli.x86_64 1:19.03.2-3.el7
Complete!
```

7. Verify that Docker is installed properly and start the service if required:

```
[root@c480ml~]# systemctl status docker
docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: https://docs.docker.com

[root@c480ml~]# systemctl start docker

[root@c480ml~]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:2557e3c07ed1e38f26e389462d03ed943586f744621577a99efb77324b0fe535
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

8. Install Nvidia-Docker2 using following commands:

```
[root@c480ml~]# distribution=$(. /etc/os-release;echo $ID$VERSION_ID)
```

```
[root@c480ml~]# curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.repo | tee
/etc/yum.repos.d/nvidia-docker.repo
[libnvidia-container]
name=libnvidia-container
baseurl=https://nvidia.github.io/libnvidia-container/centos7/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://nvidia.github.io/libnvidia-container/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt

[nvidia-container-runtime]
name=nvidia-container-runtime
baseurl=https://nvidia.github.io/nvidia-container-runtime/centos7/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://nvidia.github.io/nvidia-container-runtime/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt

[nvidia-docker]
name=nvidia-docker
baseurl=https://nvidia.github.io/nvidia-docker/centos7/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://nvidia.github.io/nvidia-docker/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

9. Install nvidia-docker2 using following command:

```
[root@c480ml~]# yum install nvidia-docker2
Loaded plugins: product-id, search-disabled-repos, subscription-manager
libnvidia-container/x86_64/signature                                                  |  488 B
00:00:00
Retrieving key from https://nvidia.github.io/libnvidia-container/gpgkey
Importing GPG key 0xF796ECB0:
 Userid     : "NVIDIA CORPORATION (Open Source Projects) <cudatools@nvidia.com>"
 Fingerprint: c95b 321b 61e8 8c18 09c4 f759 ddca e044 f796 ecb0
 From       : https://nvidia.github.io/libnvidia-container/gpgkey

<SNIP>

Installed:
  nvidia-docker2.noarch 0:2.2.2-1

Dependency Installed:
  libnvidia-container-tools.x86_64 0:1.0.5-1               libnvidia-container1.x86_64 0:1.0.5-1
  nvidia-container-runtime.x86_64 0:3.1.4-1                nvidia-container-toolkit.x86_64 0:1.0.5-2

Complete!
```

10. Configure the default runtime by adding the following to `/etc/docker/daemon.json`:

```
[root@c480ml~]# more /etc/docker/daemon.json
{
    "default-runtime": "nvidia",
    "runtimes": {
        "nvidia": {
            "path": "nvidia-container-runtime",
            "runtimeArgs": []
        }
    }
}
```

11. Enable SE Linux permission for container-runtime:

```
[root@c480ml~]# chcon system_u:object_r:container_runtime_exec_t:s0 /usr/bin/nvidia-docker
[root@c480ml~]# systemctl stop docker
[root@c480ml~]# systemctl start docker
```

12. To check if nvidia-docker is installed properly, execute the `nvidia-docker run` command and make sure the command executes without a run time error:

```
[root@c480ml~]# nvidia-docker run
"docker run" requires at least 1 argument.
See 'docker run --help'.

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container
```

Setup TensorFlow Container

NVIDIA Docker 2 environment was successfully setup in the last step. To download the TensorFlow container from the NVIDIA GPU Cloud (NGC), follow these steps:

1. Download and run the TensorFlow Container using the following command:

```
[root@c480ml~]# nvidia-docker pull nvcr.io/nvidia/tensorflow:19.08-py3
19.08-py3: Pulling from nvidia/tensorflow
7413c47ba209: Pulling fs layer
<SNIP>
Digest: sha256:64e296668d398a106f64bd840772ffb63372148b8c1170b152e7e577013661c9
Status: Downloaded newer image for nvcr.io/nvidia/tensorflow:19.08-py3
nvcr.io/nvidia/tensorflow:19.08-py3

[root@c480ml~]# nvidia-docker run -it --shm-size=1g --ulimit memlock=-1 --ulimit stack=67108864 --rm
nvcr.io/nvidia/tensorflow:19.08-py3

================
== TensorFlow ==
================

NVIDIA Release 19.08 (build 7791926)
TensorFlow Version 1.14.0

Container image Copyright (c) 2019, NVIDIA CORPORATION.  All rights reserved.
Copyright 2017-2019 The TensorFlow Authors.  All rights reserved.

Various files include modifications (c) NVIDIA CORPORATION.  All rights reserved.
NVIDIA modifications are covered by the license terms that apply to the underlying project or file.

NOTE: MOFED driver for multi-node communication was not detected.
      Multi-node communication performance may be reduced.

root@15ae33e28f4a:/workspace#
```

2. Verify the Platform GPUs are visible within the TensorFlow container:

VM with 2 NVIDIA V100DX-32C vGPUs

```
root@88d9d40b8f19:/workspace# nvidia-smi
Thu Oct 17 23:19:55 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 430.46       Driver Version: 430.46       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GRID V100DX-32C     On   | 00000000:02:02.0 Off |                    0 |
| N/A   N/A    P0    N/A /  N/A |   2064MiB / 32638MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  GRID V100DX-32C     On   | 00000000:02:03.0 Off |                    0 |
| N/A   N/A    P0    N/A /  N/A |   2064MiB / 32638MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
root@88d9d40b8f19:/workspace#
```

Cisco UCS C220 with NVIDIA T4 GPUs

```
root@88d9d40b8f19:/workspace# nvidia-smi
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.40.04    Driver Version: 418.40.04    CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:5E:00.0 Off |                    0 |
| N/A   46C    P0    26W /  70W |      0MiB / 15079MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            Off  | 00000000:D8:00.0 Off |                    0 |
| N/A   44C    P0    28W /  70W |      0MiB / 15079MiB |      5%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Cisco UCS C240 with NVIDIA V100 GPUs

```
root@88d9d40b8f19:/workspace# nvidia-smi
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.40.04    Driver Version: 418.40.04    CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  Off  | 00000000:5E:00.0 Off |                    0 |
| N/A   40C    P0    39W / 250W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-PCIE...  Off  | 00000000:86:00.0 Off |                    0 |
| N/A   40C    P0    37W / 250W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Cisco UCS C480ML with NVIDIA V100 GPUs

```
root@88d9d40b8f19:/workspace# nvidia-smi
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.40.04    Driver Version: 418.40.04    CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  Off  | 00000000:1B:00.0 Off |                    0 |
| N/A   44C    P0    60W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  Off  | 00000000:1C:00.0 Off |                    0 |
| N/A   44C    P0    59W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-SXM2...  Off  | 00000000:42:00.0 Off |                    0 |
| N/A   45C    P0    59W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-SXM2...  Off  | 00000000:43:00.0 Off |                    0 |
| N/A   45C    P0    58W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   4  Tesla V100-SXM2...  Off  | 00000000:89:00.0 Off |                    0 |
| N/A   44C    P0    56W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   5  Tesla V100-SXM2...  Off  | 00000000:8A:00.0 Off |                    0 |
| N/A   44C    P0    60W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   6  Tesla V100-SXM2...  Off  | 00000000:B2:00.0 Off |                    0 |
| N/A   43C    P0    57W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   7  Tesla V100-SXM2...  Off  | 00000000:B3:00.0 Off |                    0 |
| N/A   44C    P0    64W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

3. Exit out of the TensorFlow container to terminate the container:

```
root@28ea747714f2:/workspace# exit
```

Setup TensorFlow Convolutional Neural Network (CNN) Benchmark

The tf_cnn_benchmarks contains implementations of several popular convolutional models. To download the benchmark software, follow these steps:

1. Run the TensorFlow container and enable it to access the NFS directory `/mnt/imagenet` mounted from NetApp:

```
[root@c480ml~]# nvidia-docker run -it --shm-size=1g --ulimit memlock=-1 --ulimit stack=67108864 -v
/mnt/imagenet:/mnt/imagenet --rm nvcr.io/nvidia/tensorflow:19.08-py3

================
== TensorFlow ==
================

NVIDIA Release 19.08 (build 7791926)
TensorFlow Version 1.14.0

Container image Copyright (c) 2019, NVIDIA CORPORATION.  All rights reserved.
Copyright 2017-2019 The TensorFlow Authors.  All rights reserved.

Various files include modifications (c) NVIDIA CORPORATION.  All rights reserved.
NVIDIA modifications are covered by the license terms that apply to the underlying project or file.
```

```
NOTE: MOFED driver for multi-node communication was not detected.
      Multi-node communication performance may be reduced.

root@c0b96de271d4:/workspace#
```

2. Download the cnn_tf_v1.13_compatible.zip using the following command:

```
root@c0138c0c1aa2:/workspace# wget
https://github.com/tensorflow/benchmarks/archive/cnn_tf_v1.13_compatible.zip
--2019-09-19 04:45:21--  https://github.com/tensorflow/benchmarks/archive/cnn_tf_v1.13_compatible.zip
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/tensorflow/benchmarks/zip/cnn_tf_v1.13_compatible [following]
--2019-09-19 04:45:22--  https://codeload.github.com/tensorflow/benchmarks/zip/cnn_tf_v1.13_compatible
Resolving codeload.github.com (codeload.github.com)... 192.30.253.120
Connecting to codeload.github.com (codeload.github.com)|192.30.253.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'cnn_tf_v1.13_compatible.zip'

cnn_tf_v1.13_compatible.zip      [ <=>                                ] 321.57K  --.-KB/s    in 0.06s

2019-09-19 04:45:22 (5.39 MB/s) - 'cnn_tf_v1.13_compatible.zip' saved [329287]

root@c0b96de271d4:/workspace#
```

3. Unzip cnn_tf_v1.13_compatible.zip:

```
root@c0b96de271d4:/workspace# unzip cnn_tf_v1.13_compatible.zip
Archive:  cnn_tf_v1.13_compatible.zip
4828965154c424bc61a7ec361edb67bb267869f4
   creating: benchmarks-cnn_tf_v1.13_compatible/
  inflating: benchmarks-cnn_tf_v1.13_compatible/.gitignore

<SNIP>

  inflating: benchmarks-cnn_tf_v1.13_compatible/scripts/tf_cnn_benchmarks/variable_mgr_util.py
  inflating: benchmarks-cnn_tf_v1.13_compatible/scripts/tf_cnn_benchmarks/variable_mgr_util_test.py
root@c0b96de271d4:/workspace#
```

4. Run benchmark test using RESNET50 model on synthetic data, use the following command adjusting the highlighted number of GPUs depending on the platform in use. The command below was run on Cisco UCS C220 M5 with 2 NVIDIA T4 GPUs:

```
root@c0138c0c1aa2:/workspace# cd benchmarks-cnn_tf_v1.13_compatible/scripts/tf_cnn_benchmarks/

root@c0138c0c1aa2:/workspace/benchmarks-cnn_tf_v1.13_compatible/scripts/tf_cnn_benchmarks# python
tf_cnn_benchmarks.py --data_format=NHWC --batch_size=256 --model=resnet50 --optimizer=momentum --
variable_update=replicated --nodistortions --gradient_repacking=8 --num_gpus=2 --num_epochs=50 --
weight_decay=1e-4 --all_reduce_spec=nccl --local_parameter_device=gpu --use_fp16

TensorFlow:  1.14
Model:       resnet50
Dataset:     imagenet (synthetic)
Mode:        training
SingleSess:  False
Batch size:  512 global
             256 per device
Num batches: 125114
Num epochs:  50.00
Devices:     ['/gpu:0', '/gpu:1']
NUMA bind:   False
Data format: NHWC
Optimizer:   momentum
Variables:   replicated
AllReduce:   nccl
==========
Generating model
```

```
<SNIP>

Done warm up
Step       Img/sec total_loss
Step       Img/sec total_loss
1       images/sec: 506.1 +/- 0.0 (jitter = 0.0)        8.752
10      images/sec: 505.3 +/- 0.1 (jitter = 0.4)        8.712
20      images/sec: 505.1 +/- 0.1 (jitter = 0.5)        8.563
30      images/sec: 504.6 +/- 0.2 (jitter = 1.0)        8.502
40      images/sec: 504.0 +/- 0.2 (jitter = 1.6)        8.430
50      images/sec: 503.5 +/- 0.2 (jitter = 2.1)        8.430
60      images/sec: 503.1 +/- 0.2 (jitter = 2.4)        8.366
<SNIP>
```

The GPU power consumption, temperature and load can be verified by opening a second SSH connection to the RHEL host and executing “nvidia-smi” command. The images/sec will vary depending on the number and type of the GPUs in use

Setup CNN Benchmark for ImageNet Data

ImageNet is an ongoing research effort to provide researchers around the world an easily accessible image database. To download ImageNet data, a registered ImageNet account is required. Signup for the account at the following URL: http://www.image-net.org/signup.

The ImageNet data is available in the form of tar and zipped-tar files. This data needs to be converted to a format that TensorFlow and CNN Benchmark can utilize. Three main files required to setup the ImageNet data set for TensorFlow are:

- ILSVRC2012_bbox_train_v2.tar.gz (bounding boxes)
- ILSVRC2012_img_val.tar (validation images)
- ILSVRC2012_img_train.tar (training images)

The TensorFlow container includes appropriate scripts to both download and convert ImageNet data into the required format.

To download the raw images, the user must generate a username and access_key. This username and access_key are required to log into ImageNet and download the images. If the three ImageNet files are already downloaded, create a directory named “/mnt/imagenet/raw-data” and copy these files in the raw-data directory. Run the script (shown in the step below) providing a dummy Username and Access Key. The script will automatically fail download because of incorrect credentials but will continue to process files after finding the necessary files in the raw-data folder.

1. From within the TensorFlow container, find and execute the following script:

```
root@c0138c0c1aa2:/workspace# cd /workspace/nvidia-examples/build_imagenet_data

# Execute the following script
# ./download_and_preprocess_imagenet.sh [data-dir]

root@c0138c0c1aa2:/workspace/nvidia-examples/build_imagenet_data# ./download_and_preprocess_imagenet.sh
/mnt/imagenet/

In order to download the imagenet data, you have to create an account with
image-net.org. This will get you a username and an access key. You can set the
IMAGENET_USERNAME and IMAGENET_ACCESS_KEY environment variables, or you can
```

```
enter the credentials here.
Username: xxxx
Access key: xxxx

<SNIP>
```

The download and conversion process can take a few hours and depends a lot on the Internet download speed. At the time of writing this document, the three files use almost 155GB. At the end of the process, the following files are observed in the /mnt/imagenet directory:

- A directory named `raw-data` containing various files including raw images
- A large number of sequential train and validation files in `/mnt/imagenet`. These files are the processed files ready to be used by the CNN benchmark.

2. To run a CNN benchmark using ImageNet dataset, use the following command (adjust the number of GPUs --num_gpus=<> based on the C-series server in use). The following command was executed on a C220 with two NVIDIA T4 GPUs:

```
root@c0b96de271d4:~# export DATA_DIR=/mnt/imagenet/
root@c0b96de271d4:~# cd /workspace/benchmarks-cnn_tf_v1.13_compatible/scripts/tf_cnn_benchmarks/
root@c0b96de271d4:/workspace/benchmarks-cnn_tf_v1.13_compatible/scripts/tf_cnn_benchmarks# python
tf_cnn_benchmarks.py --data_format=NHWC --batch_size=256 --model=resnet50 --optimizer=momentum --
variable_update=replicated --nodistortions --gradient_repacking=8 --num_gpus=2 --num_epochs=50 --
weight_decay=1e-4 --all_reduce_spec=nccl --local_parameter_device=gpu --use_fp16 --data_dir=${DATA_DIR}

TensorFlow:  1.14
Model:       resnet50
Dataset:     imagenet
Mode:        training
SingleSess:  False
Batch size:  512 global
             256 per device
Num batches: 125114
Num epochs:  50.00
Devices:     ['/gpu:0', '/gpu:1']
NUMA bind:   False
Data format: NHWC
Optimizer:   momentum
Variables:   replicated
AllReduce:   nccl
==========

<SNIP>

2019-09-19 04:57:16.704699: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1326] Created TensorFlow
device (/job:localhost/replica:0/task:0/device:GPU:0 with 14132 MB memory) -> physical GPU (device: 0,
name: Tesla T4, pci bus id: 0000:5e:00.0, compute capability: 7.5)
2019-09-19 04:57:16.706680: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1326] Created TensorFlow
device (/job:localhost/replica:0/task:0/device:GPU:1 with 14132 MB memory) -> physical GPU (device: 1,
name: Tesla T4, pci bus id: 0000:d8:00.0, compute capability: 7.5)
<SNIP>

Done warm up
Step    Img/sec total_loss
1       images/sec: 506.6 +/- 0.0 (jitter = 0.0)        8.670
10      images/sec: 506.0 +/- 0.3 (jitter = 1.0)        8.658
20      images/sec: 505.5 +/- 0.4 (jitter = 1.3)        8.518
30      images/sec: 505.3 +/- 0.3 (jitter = 1.5)        8.521
40      images/sec: 505.0 +/- 0.2 (jitter = 1.2)        8.412
50      images/sec: 504.7 +/- 0.2 (jitter = 1.4)        8.399
60      images/sec: 504.4 +/- 0.2 (jitter = 1.8)        8.341
<SNIP>
```

Performance Metrics

As part of the solution validation, the performance of a few popular Artificial Neural Network (ANN) models was evaluated. The ANN models were run with different supported batch sizes with a minimum of 2 epochs for each run.

The performance tests were carried out on a single Cisco UCS C480 ML server with 8 NVIDIA Tesla SXM2 V100 32GB GPUs. The ImageNet dataset was hosted on the AFF A800 system by using a single FlexGroup volume and was accessed by the Cisco UCS C480 ML server via NFSv3.

Cisco UCS C480 ML M5 Performance Metrics

For various compute related performance metrics, refer to the Cisco UCS C480 ML M5 Performance Characterization white paper:

https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/whitepaper-c11-741689.pdf

A subset of the performance tests outlined in the paper above were also executed on the FlexPod AI setup. These models include the following:

- RESNET 50
- RESNET 152
- VGG 16
- Inception V3

The results from these tests for synthetic as well as ImageNet data were in-line with the performance data explained in the performance white paper. Refer to Figure 8 in the white paper for a plot of various images/second results.

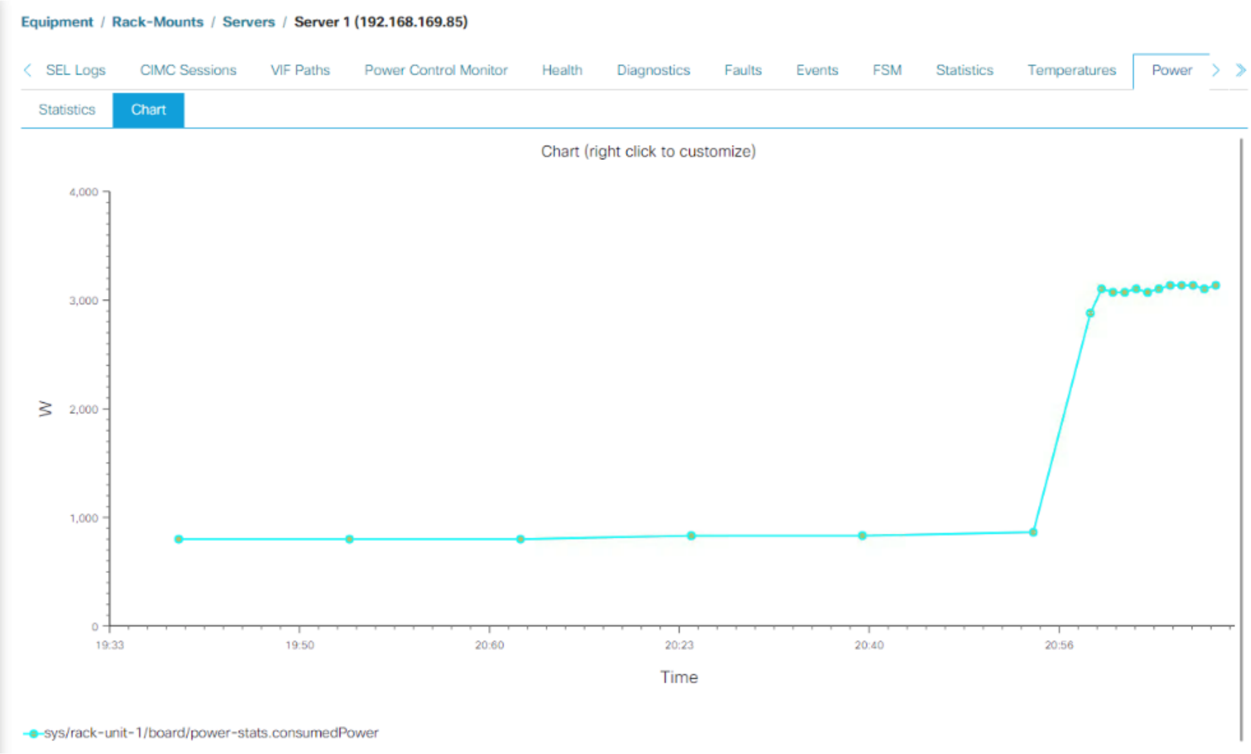
Cisco UCS C480 ML M5 Power Consumption

When a performance benchmark test utilizes all the 8 NVIDIA V100 SXM2 GPUs, the power consumptions of Cisco UCS C480 ML M5 platform increases. The following command shows the GPU utilization, GPU power consumption and temperature:

```
[root@c480ml~]# nvidia-smi

+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.40.04    Driver Version: 418.40.04    CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  Off  | 00000000:1B:00.0 Off |                    0 |
| N/A   62C    P0   265W / 300W |  31281MiB / 32510MiB |     97%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  Off  | 00000000:1C:00.0 Off |                    0 |
| N/A   62C    P0   257W / 300W |  31281MiB / 32510MiB |     96%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-SXM2...  Off  | 00000000:42:00.0 Off |                    0 |
| N/A   61C    P0   268W / 300W |  31281MiB / 32510MiB |     96%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-SXM2...  Off  | 00000000:43:00.0 Off |                    0 |
| N/A   62C    P0   181W / 300W |  31281MiB / 32510MiB |     97%      Default |
+-------------------------------+----------------------+----------------------+
|   4  Tesla V100-SXM2...  Off  | 00000000:89:00.0 Off |                    0 |
| N/A   59C    P0   241W / 300W |  31281MiB / 32510MiB |     97%      Default |
+-------------------------------+----------------------+----------------------+
|   5  Tesla V100-SXM2...  Off  | 00000000:8A:00.0 Off |                    0 |
| N/A   61C    P0   273W / 300W |  31281MiB / 32510MiB |     97%      Default |
+-------------------------------+----------------------+----------------------+
|   6  Tesla V100-SXM2...  Off  | 00000000:B2:00.0 Off |                    0 |
| N/A   62C    P0   266W / 300W |  31281MiB / 32510MiB |     97%      Default |
+-------------------------------+----------------------+----------------------+
|   7  Tesla V100-SXM2...  Off  | 00000000:B3:00.0 Off |                    0 |
| N/A   60C    P0   285W / 300W |  31281MiB / 32510MiB |     96%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0    219995      C   python                                     31262MiB |
|    1    219995      C   python                                     31262MiB |
|    2    219995      C   python                                     31262MiB |
|    3    219995      C   python                                     31262MiB |
|    4    219995      C   python                                     31262MiB |
|    5    219995      C   python                                     31262MiB |
|    6    219995      C   python                                     31262MiB |
|    7    219995      C   python                                     31262MiB |
+-----------------------------------------------------------------------------+
```

To find out the system power utilization, follow these steps:

1. Log into Cisco UCS Manager.
2. Click Server on the right and click the Cisco UCS C480 ML M5’s service profile.
3. Click the Associated Server in the main window to open the physical server properties window.
4. In the main window, click Power.
5. Under Power, click Chart and add Motherboard Power Counters (Consumed Power) to see the power consumption chart:



Cisco UCS 240 M5 Power Consumption

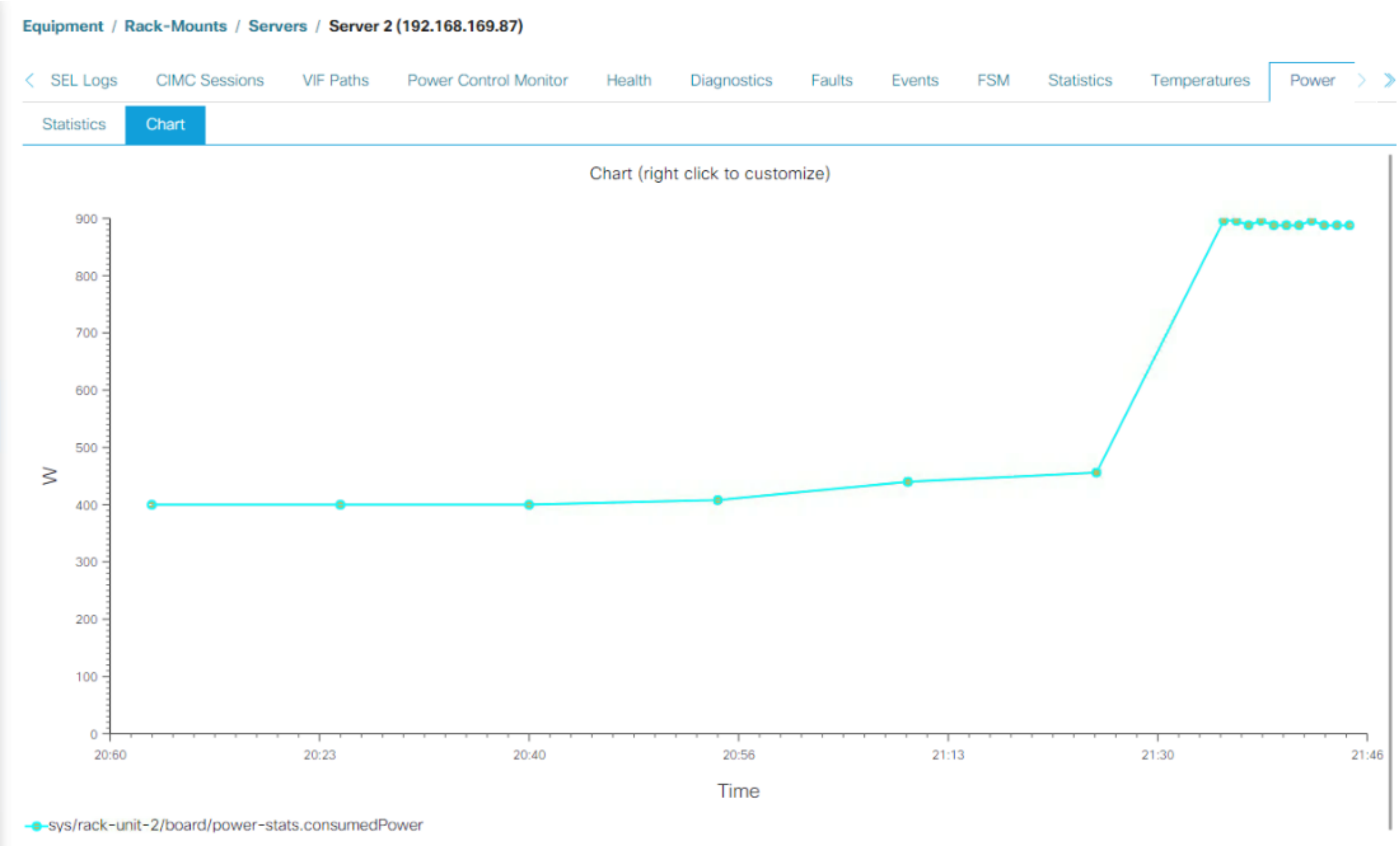
When a performance benchmark test utilizes both the NVIDIA V100 PCIE GPUs, the power consumptions of Cisco UCS C240 M5 platform increases. The following command shows the GPU utilization, GPU power consumption and temperature:

```
[root@c240-2 ~]# nvidia-smi

+-----------------------------------------------------------------------------+
|  NVIDIA-SMI 418.40.04    Driver Version: 418.40.04    CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  Off  | 00000000:5E:00.0 Off |                    0 |
| N/A   56C    P0   204W / 250W |  31207MiB / 32510MiB |     97%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-PCIE...  Off  | 00000000:86:00.0 Off |                    0 |
| N/A   58C    P0   187W / 250W |  31207MiB / 32510MiB |     98%      Default |
+-------------------------------+----------------------+----------------------+


+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0    141772      C   python                                     31194MiB |
|    1    141772      C   python                                     31194MiB |
+-----------------------------------------------------------------------------+
```

To find out the system power utilization, follow these steps:

1. Log into Cisco UCS Manager.
2. Click Server on the right and click the Cisco UCS C240 M5's service profile.
3. Click on the Associated Server in the main window to open the physical server properties window.

4. In the main window, click Power.
5. Under Power, click Chart and add Motherboard Power Counters (Consumed Power) to see the power consumption chart:



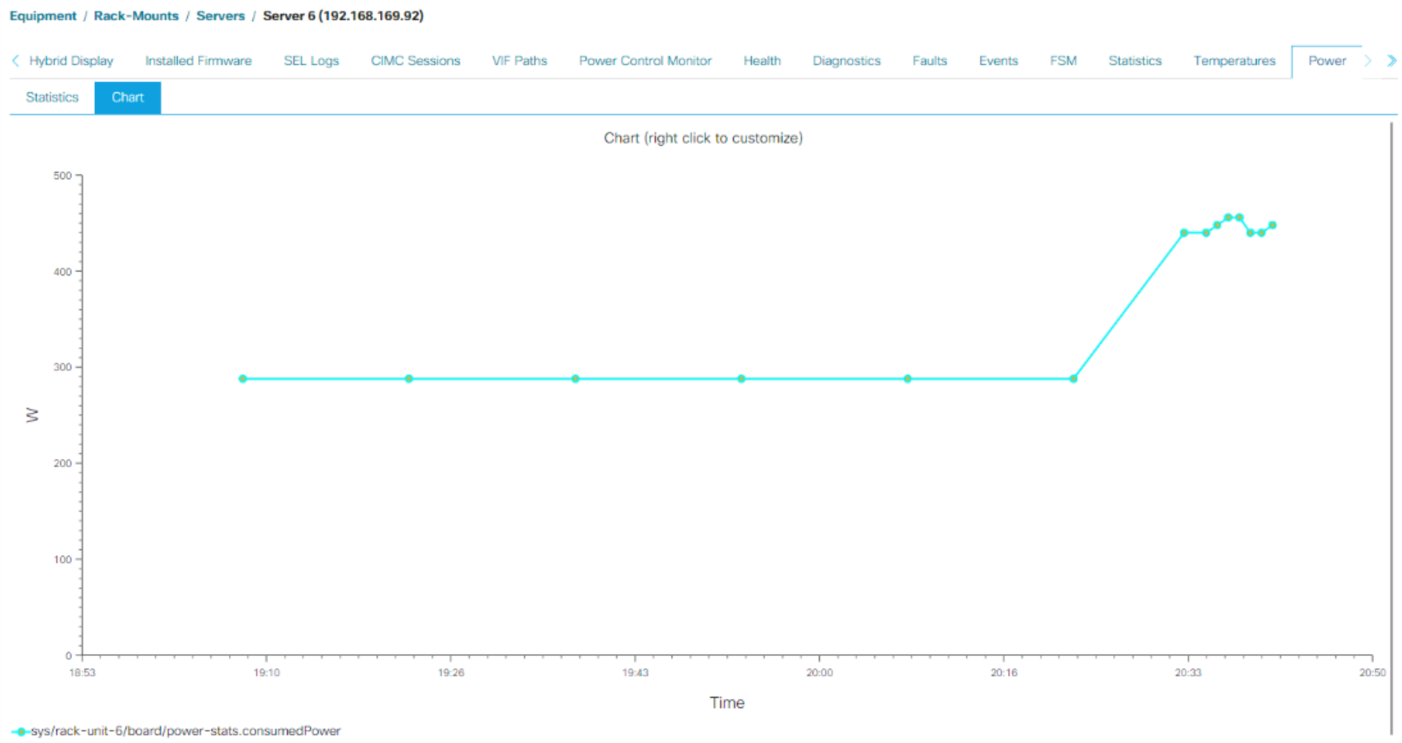
Cisco UCS 220 M5 Power Consumption

When a performance benchmark test utilizes both the NVIDIA T4 GPUs, the power consumptions of Cisco UCS C220 M5 platform increases. The following command shows the GPU utilization, GPU power consumption and temperature:

```
[root@c220-1 ~]# nvidia-smi

+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.40.04    Driver Version: 418.40.04    CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:5E:00.0 Off |                    0 |
| N/A   65C    P0    76W /  70W |  14737MiB / 15079MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            Off  | 00000000:D8:00.0 Off |                    0 |
| N/A   63C    P0    51W /  70W |  14737MiB / 15079MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0     197070      C   python                                    14727MiB |
|    1     197070      C   python                                    14727MiB |
+------------------------------------------------------------------------------+
```

To find out the system power utilization, follow these steps:

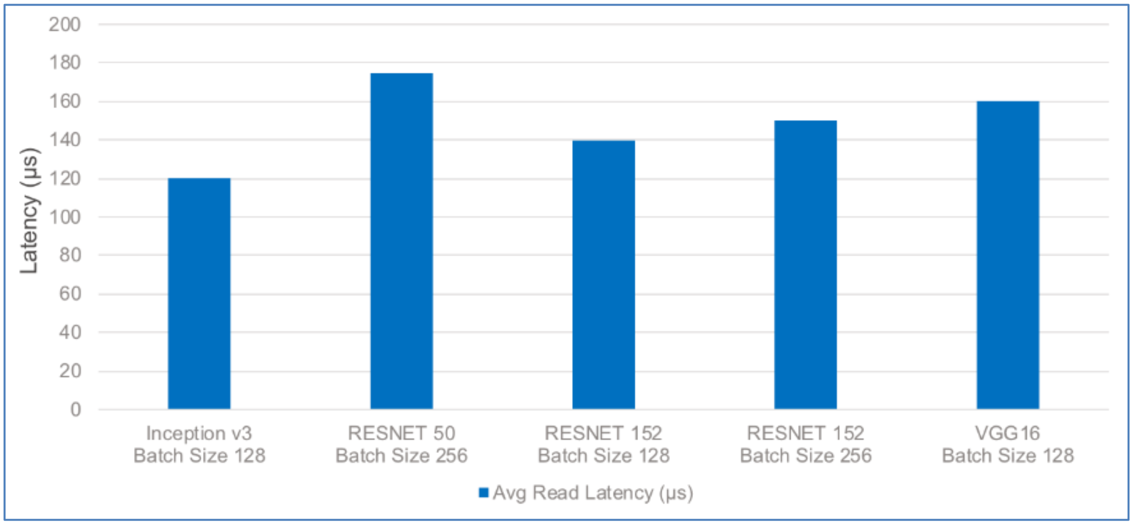
1. Log into Cisco UCS Manager.
2. Click Server on the right and click the Cisco UCS C220 M5 service profile.
3. Click the Associated Server in the main window to open the physical server properties window.
4. In the main window, click Power.
5. Under Power, click Chart and add Motherboard Power Counters (Consumed Power) to see the power consumption chart:



NetApp AFF A800 Performance Metrics

Storage Latency

A NetApp AFF A800 HA pair has been tested and proven to support throughputs up to 25GB/s under 1ms latency for NAS workloads.

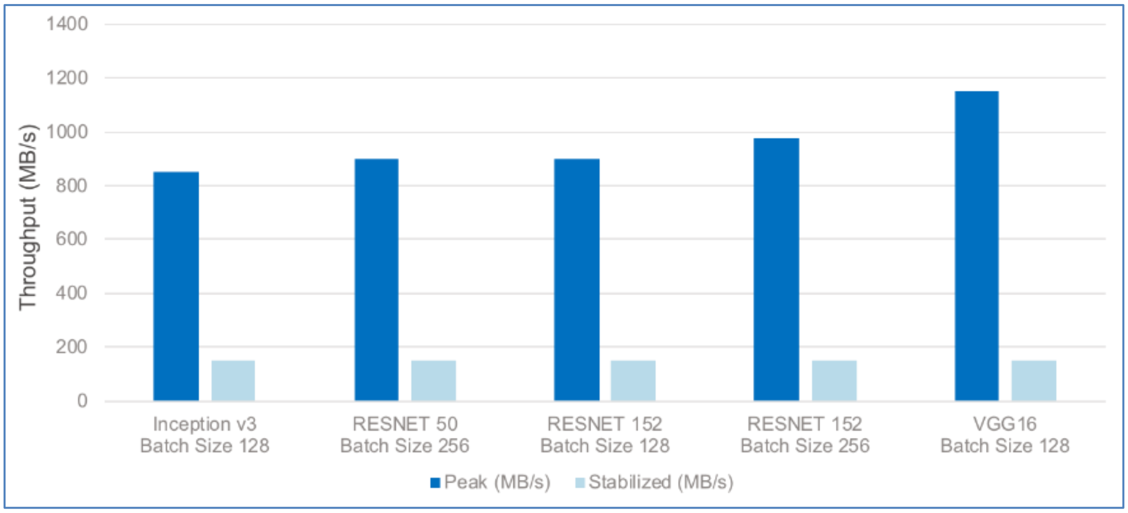


The Average Read Latency across all the ANN models was roughly in the range of 100µs to 200µs, leaving a lot of room to grow workloads with bigger datasets and also to use multiple Cisco UCS C480 ML M5 servers while still keeping the latency under 1ms.

Storage Throughput

An initial rush to load the GPUs with data was observed and the throughput peaked during this time, however this peak did not sustain for long. The throughput stabilized at a much lower value after about a minute as all the data was now loaded in the NVIDIA GPUs.

The ImageNet dataset used during these tests is a relatively small dataset when compared to real-time datasets which can be in petabytes. With such large datasets in petabytes, which exceed the local memory capacity by a huge margin, the storage access throughput will constantly rise as and when new data needs to be pumped to the NVIDIA GPUs, this will typically continue until late in the training epoch.



Summary

Artificial Intelligence (AI) and Machine Learning (ML) initiatives have seen a tremendous growth due to the recent advances in GPU computing technology. The FlexPod Datacenter for AI/ML with Cisco UCS 480 ML solution aims to deliver a seamless integration of the Cisco UCS C480 ML M5 platform into the current FlexPod portfolio to enable you to easily utilize the platform's extensive GPU capabilities for their workloads without requiring extra time and resources for a successful deployment.

The validated solution achieves the following core design goals:

- Optimized integration of Cisco UCS C480 ML M5 platform into the FlexPod design
- Integration of NetApp A800 NVMe based all flash storage system to support AI/ML dataset.
- Showcase AI/ML workload acceleration using NVIDIA V100 32G GPUs on Cisco UCS C480 ML M5 and Cisco UCS C240 M5 platforms.
- Showcase AI/ML workload acceleration using NVIDIA T4 GPUs on Cisco UCS C220 M5 platform.
- Showcase NVIDIA Virtual Compute Servers (vComputeServer) and Virtual GPU (vGPU) capabilities on various UCS platforms.
- Support for Intel 2nd Gen Intel Xeon Scalable Processors (Cascade Lake) processors.
- NetApp FlexGroup volumes and NetApp ONTAP 9.6 release

References

Products and Solutions

Cisco Unified Computing System:

http://www.cisco.com/en/US/products/ps10265/index.html

Cisco UCS 6454 Fabric Interconnects:

https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/datasheet-c78-741116.html

Cisco UCS 5100 Series Blade Server Chassis:

http://www.cisco.com/en/US/products/ps10279/index.html

Cisco UCS B-Series Blade Servers:

http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-b-series-blade-servers/index.html

Cisco UCS C480 ML M5 Rack Server:

https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/datasheet-c78-741211.html

Cisco UCS VIC 1400 Adapters:

https://www.cisco.com/c/en/us/products/collateral/interfaces-modules/unified-computing-system-adapters/datasheet-c78-741130.html

Cisco UCS Manager:

http://www.cisco.com/en/US/products/ps10281/index.html

NVIDIA GPU Cloud

https://www.nvidia.com/en-us/gpu-cloud/

Cisco Nexus 9336C-FX2 Switch:

https://www.cisco.com/c/en/us/support/switches/nexus-9336c-fx2-switch/model.html

VMware vCenter Server:

http://www.vmware.com/products/vcenter-server/overview.html

NetApp Data ONTAP:

http://www.netapp.com/us/products/platform-os/ontap/index.aspx

NetApp AFF A800:

https://www.netapp.com/us/products/storage-systems/all-flash-array/aff-a-series.aspx

Interoperability Matrixes

Cisco UCS Hardware Compatibility Matrix:

https://ucshcltool.cloudapps.cisco.com/public/

VMware Compatibility Guide:

http://www.vmware.com/resources/compatibility

NetApp Interoperability Matric Tool:

http://mysupport.netapp.com/matrix/

About the Authors

Haseeb Niazi, Technical Marketing Engineer, Cisco Systems, Inc.

Haseeb Niazi has over 20 years of experience at Cisco in the Data Center, Enterprise and Service Provider Solutions and Technologies. As a member of various solution teams and Advanced Services, Haseeb has helped many enterprise and service provider customers evaluate and deploy a wide range of Cisco solutions. As a technical marking engineer at Cisco UCS Solutions group, Haseeb focuses on network, compute, virtualization, storage and orchestration aspects of various Compute Stacks. Haseeb holds a master's degree in Computer Engineering from the University of Southern California and is a Cisco Certified Internetwork Expert (CCIE 7848).

Arvind Ramakrishnan, Solutions Architect, NetApp, Inc.

Arvind Ramakrishnan works for the NetApp Infrastructure and Cloud Engineering team. He focusses on development, validation and implementation of Cloud Infrastructure solutions that include NetApp products. Arvind has more than 10 years of experience in the IT industry specializing in Data Management, Security, Cloud and Data Center technologies. Arvind holds a bachelor’s degree in Electronics and Communication.

Acknowledgements

For their support and contribution to the design, validation, and creation of this Cisco Validated Design, the authors would like to thank:

- John George, Technical Marketing Engineer, Cisco Systems, Inc.