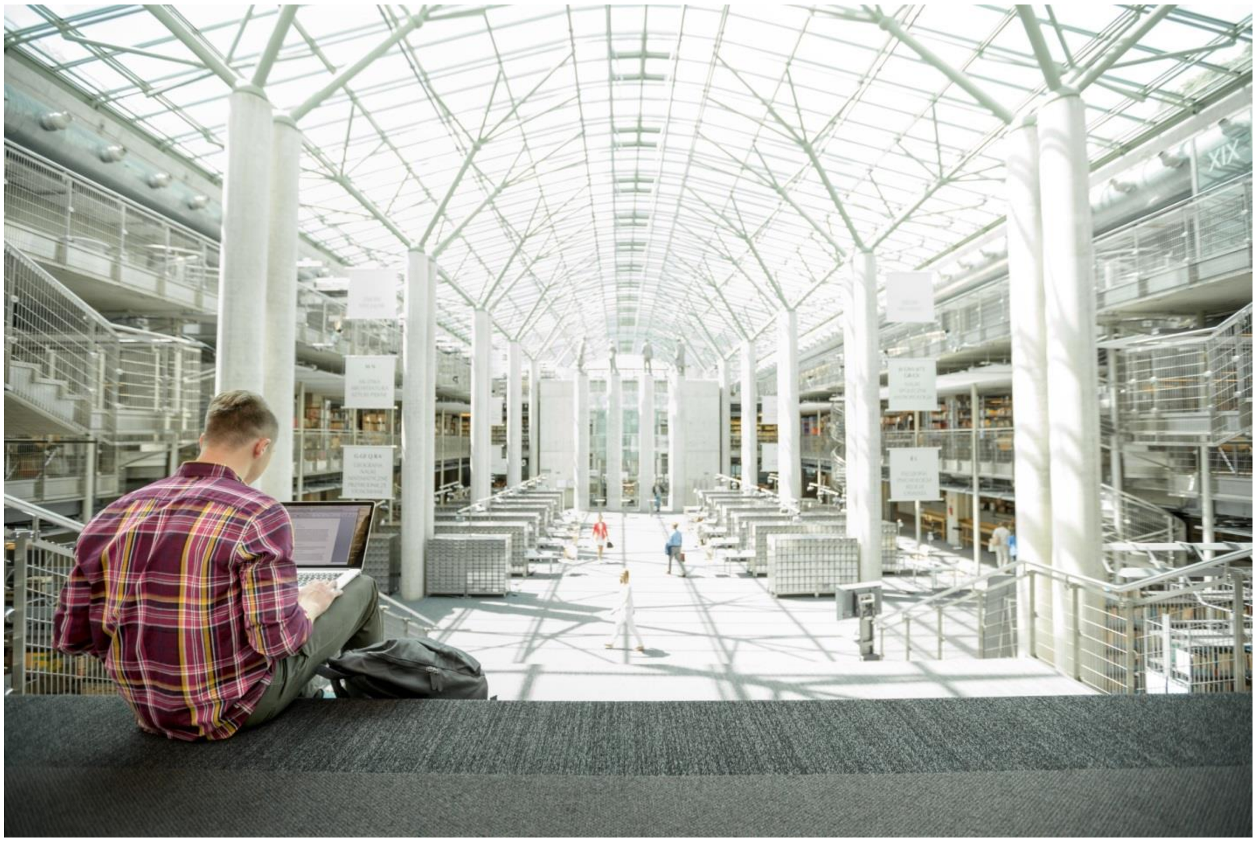


Cisco Data Intelligence Platform with Cloudera Enterprise Data Hub 6.2 and Cloudera Data Science Workbench 1.5

Deployment Guide for the Cisco Data Intelligence Platform with Cloudera Enterprise Data Hub 6.2

Last Updated: January 8, 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.

Table of Contents

Executive Summary

Data scientists are constantly searching for newer techniques and methodologies that can unlock the value of big data and distill this data further to identify additional insights which could transform productivity and provide business differentiation.

One such area is Artificial Intelligence/Machine Learning (AI/ML), which has seen tremendous development with bringing in new frameworks and new forms of compute (CPU, GPU and FPGA) to work on data to provide key insights. While data lakes have historically been data intensive workloads, these advancements in technologies have led to a new growing demand of compute intensive workloads to operate on the same data.

While data scientists want to be able to use the latest and greatest advancements in AI/ML software and hardware technologies on their datasets, the IT team is also constantly looking at enabling these data scientists to be able to provide such a platform to a data lake. This has led to architecturally siloed implementations. When data, which is ingested, worked, and processed in a data lake, needs to be further operated by AI/ML frameworks, it often leaves the platform and must be on-boarded to a different platform to be processed. This would be fine if this demand is seen only on a small percentage of workloads. However, AI/ML workloads working closely on the data in a data lake are seeing an increase in adoption. For instance, data lakes in customer environment are seeing deluge of data from new use cases such as IoT, autonomous driving, smart cities, genomics and financials, who are all seeing more and more demand of AI/ML processing of this data.

IT is demanding newer solutions to enable data scientists to operate on both a data lake and an AI/ML platform (or a compute farm) without worrying about the underlying infrastructure. IT also needs this to seamlessly grow to cloud scale while reducing the TCO of this infrastructure and without affecting utilization. Thus, driving a need to plan a data lake along with an AI/ML platform in a systemic fashion.

Seeing this increasing demand by IT, and also envisioning this as a natural extension of a data lake, we announced the Cisco Data Intelligence Platform. Cisco Data Intelligence Platform is discussed in detail here.

This CVD implements Cisco Data Intelligence Platform on Cisco UCS using Cloudera Enterprise Data Hub 6.2.0.

Solution Overview

Introduction

Both Big Data and machine learning technology have progressed to the point where they are being implemented in production systems running 24x7. There exists a very clear need for a proven, dependable, high-performance platform for the ingestion, processing, storage and analysis of the data, as well as the seamless dissemination of the output, results and insights of the analysis.

This solution implements the Cisco UCS Integrated Infrastructure for Big Data and Analytics based on Cisco Data Intelligence Platform (CDIP) architecture, a world-class platform specifically designed for demanding workloads that is both easy to scale and easy to manage, even as the requirements grow to thousands of servers and petabytes of storage; and the Cloudera Data Science Workbench, an integrated set of tools designed to enable flexible, fast access to the entire data store.

Many companies, recognizing the immense potential of big data and machine learning technology, are gearing up to leverage these new capabilities, building out departments and increasing hiring. However, these efforts face a new set of challenges:

- Making the data available to the diverse set of people who need it
- Enabling access to high-performance computing resources, GPUs, that also scale with the data growth
- Allowing people to work with the data using the environments in which they are familiar
- Publishing their results so the organization can make use of it
- Enabling the automated production of those results
- Managing the data for compliance and governance
- Scaling the system as the data grows
- Managing and administering the system in an efficient, cost-effective way

This solution is based on the Cisco UCS Integrated Infrastructure for Big Data and Analytics and includes computing, storage, connectivity, and unified management capabilities to help companies manage the immense amount of data being collected. It is built on Cisco Unified Computing System (Cisco UCS) infrastructure, using Cisco UCS 6332 Series Fabric Interconnects, and Cisco UCS C-Series Rack Servers. This architecture is specifically designed for performance and linear scalability for big data and machine learning workload.

Audience

The intended audience of this document includes sales engineers, field consultants, professional services, IT managers, partner engineering and customers who want to deploy the Cloudera Distribution with Apache Hadoop (CDH 6.2.0) and Cloudera Data Science Workbench (CDSW 1.5.0) on the Cisco UCS Integrated Infrastructure for Big Data and Analytics (Cisco UCS M5 Rack mount servers).

Purpose of this Document

This document describes the architecture and deployment procedures for Cloudera 6.2.0 with Cloudera Data Science Workbench 1.5.0 on a 30-node Cisco UCS C240 M5 and Cisco UCS S3260 cluster based on Cisco UCS Integrated Infrastructure for Big Data and Analytics.

This document describes the architecture and step by step guidelines of deployment procedures for Cisco Data Intelligence Platform using Cloudera Enterprise Data Hub (CDH) 6.2.0 on Cisco UCS C240 M5.

This document explains the process of deploying the three independently scalable components of the architecture including data lake, AI/ML and Object stores:

- Data Lake with Cloudera Enterprise Data Hub 6.2
- Kubernetes / AI farm with Cloudera Data Science Workbench
- Hadoop Tiered storage with S3260

What's New in this Release?

This solution implements the Cisco UCS Integrated Infrastructure for Big Data and Analytics based on Cisco Data Intelligence Platform (CDIP) architecture , a world-class platform specifically designed for demanding workloads that is both easy to scale and easy to manage, even as the requirements grow to thousands of servers and petabytes of storage; and Hortonworks Data Platform, an integrated set of tools designed to enable flexible, fast access to the entire data store, while enabling AI workloads on servers with GPUs.

This CVD implements the following:

- Data Lake with Cloudera Enterprise Data Hub 6.2.0 on Cisco UCS Integrated Infrastructure for Big Data and Analytics
- Kubernetes / AI farm with Cloudera Data Science Workbench (CDSW) 1.5
  - Enable CUDA for the GPUs
  - Enable GPU as a resource to the Docker Containers through CDSW
  - Enable GPU isolation and scheduling (with Docker Containers) through YARN 2.0
  - Downloading a TensorFlow image from NVIDIA Cloud (NGC)
  - Adding trusted registries for Docker for YARN 2.0
  - Execute a sample TensorFlow job accessing data from Hadoop and running on a Docker container with GPU as a resource scheduled by YARN 2.0
- Installation and setup of the above through Cloudera Manager (CM).
- Tiered storage on Cisco UCS S3260 M5 storage server chassis.

What's Next?

This CVD showcases Cisco UCS Manager (UCSM). This solution can also be deployed using Cisco Intersight. Additional Cisco UCS features will be added to the Appendix in the following months. Some of these include the following:

- Cisco Intersight
- Cloudera Data Science Workbench
- Tiered Storage with HDFS on S3260
- Cisco Boot optimized M.2 Raid Controller for hardware RAID
- 4th Generation FI
- Hadoop data offload to S3 compliant storage

Solution Summary

This CVD details the process of installing Cloudera 6.2.0 with Cloudera Data Science Workbench (CDSW) 1.5.0 and the configuration details of the cluster. The current version of Cisco UCS Integrated Infrastructure for Big Data and Analytics offers the following configurations depending on the compute and storage requirements.

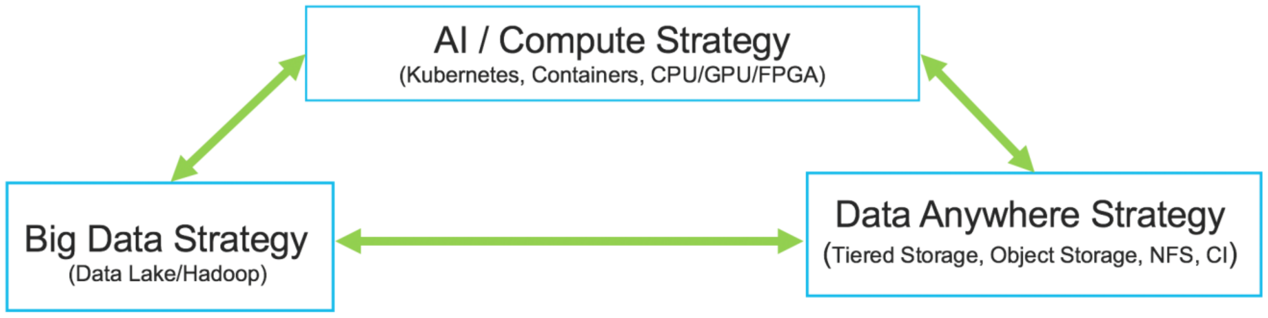
Cisco Data Intelligence Platform

Cisco Data Intelligence Platform (CDIP) is a cloud scale architecture which brings together big data, AI/compute farm, and storage tiers to work together as a single entity while also being able to scale independently to address the IT issues in the modern data center. This architecture allows for:

- Extremely fast data ingest, and data engineering done at the data lake
- AI compute farm allowing for different types of AI frameworks and compute types (GPU, CPU, FPGA) to work on this data for further analytics
- A storage tier, allowing to gradually retire data which has been worked on to a storage dense system with a lower $/TB providing a better TCO
- Seamlessly scale the architecture to thousands of nodes with a single pane of glass management using Cisco Application Centric Infrastructure (ACI)

Cisco Data Intelligence Platform caters to the evolving architecture bringing together a fully scalable infrastructure with centralized management and fully supported software stack (in partnership with industry leaders in the space) to each of these three independently scalable components of the architecture including data lake, AI/ML and Object stores.

Figure 1 Cisco Data Intelligent Platform

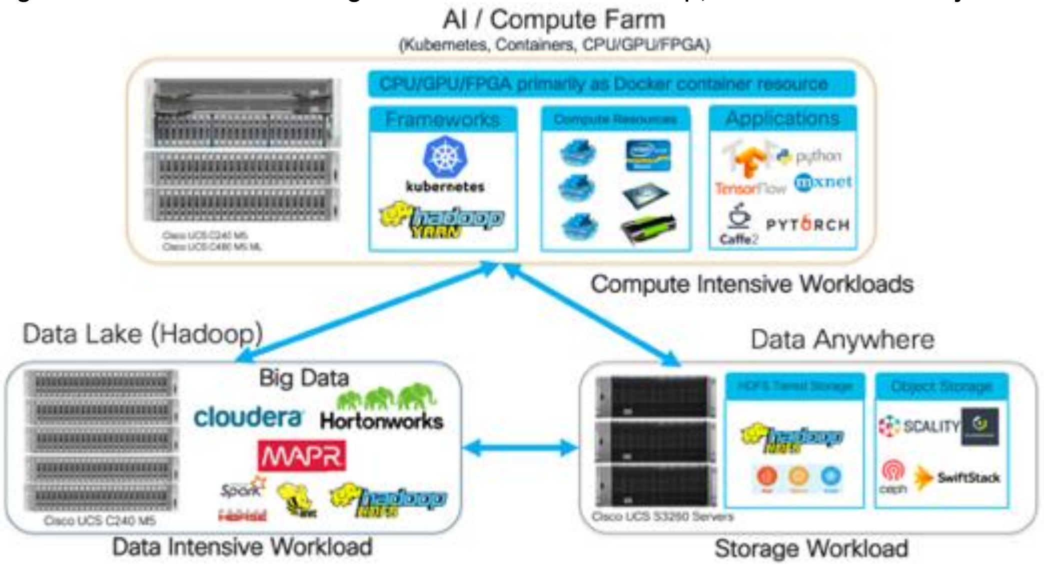

AI / Compute Strategy
(Kubernetes, Containers, CPU/GPU/FPGA)
Big Data Strategy
(Data Lake/Hadoop)
Data Anywhere Strategy
(Tiered Storage, Object Storage, NFS, CI)

Cisco has developed numerous industry leading Cisco Validated Designs (reference architectures) in the area of Big Data (CVDs with Cloudera, Hortonworks and MapR), compute farm with Kubernetes (CVD with RedHat OpenShift) and Object store (Scality, SwiftStack, Cloudian, and others).

This Cisco Data Intelligence Platform can be deployed in these variants:

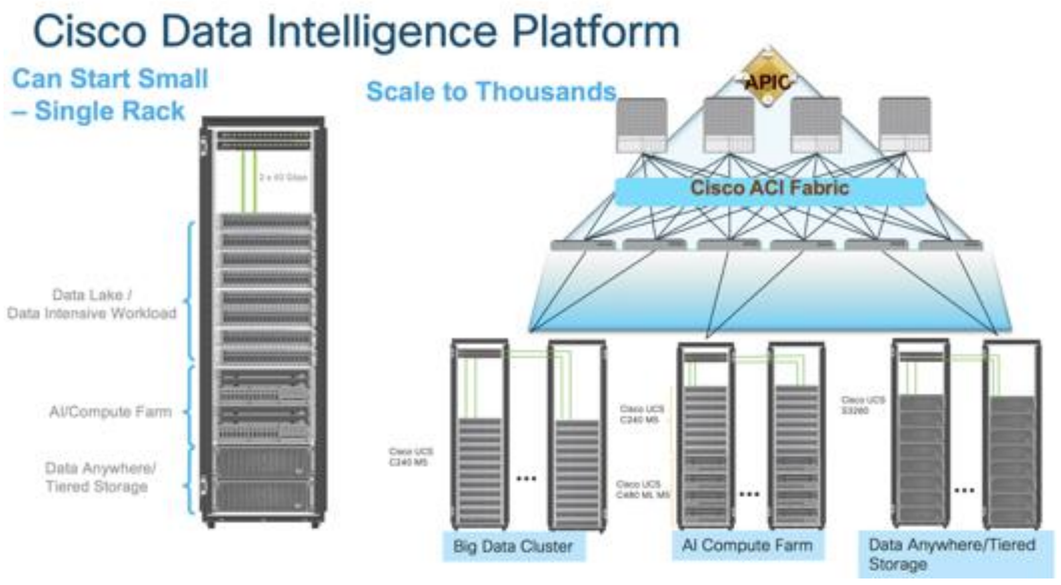
- CDIP with Cloudera with Data Science Workbench (powered by Kubernetes) and Tiered Storage with Hadoop
- CDIP with Hortonworks with Apache Hadoop 3.1 and Data Science Workbench (powered by Kubernetes) and Tiered Storage with Hadoop

Figure 2 Cisco Data Intelligence Platform with Hadoop, Kubernetes and Object Store



This architecture can start from a single rack and scale to thousands of nodes with a single pane of glass management with Cisco Application Centric Infrastructure (ACI).

Figure 3 Cisco Data Intelligent Platform at Scale


Cisco Data Intelligence Platform
Can Start Small – Single Rack
Scale to Thousands
APIC
Cisco ACI Fabric
Data Lake / Data Intensive Workload
AI/Compute Farm
Data Anywhere/ Tiered Storage
Cisco UCS C240 M5
Cisco UCS C240 M5
Cisco UCS C480 ML M5
Cisco UCS S3260
Big Data Cluster
AI Compute Farm
Data Anywhere/Tiered Storage

Reference Architecture

Table 1 , Table 2 , and Table 3 summarize the reference architecture configuration details for the data lake, AI/ML components of the data lake, and tiered storage.

Data Lake Reference Architecture

Table 1 lists the data lake reference architecture configuration details for Cisco UCS Integrated Infrastructure for Big Data and Analytics.

Table 1 Cisco UCS Integrated Infrastructure for Big Data and Analytics Configuration Options

	Performance (UCS-SP-C240M5-A2)	Capacity (UCS-SPC240M5L-S1)	High Capacity (UCS-SP-S3260-BV)
Servers	16 x Cisco UCS C240 M5 Rack Servers with small-form-factor (SFF) drives	16 x Cisco UCS C240 M5 Rack Servers with large-form-factor (LFF) drives	8 x Cisco UCS S3260 Storage Servers
CPU	2 x 2nd Gen Intel® Xeon® Scalable 6230 processors (2 x 20 cores, at 2.1 GHz)	2 x 2nd Gen Intel Xeon Scalable 6230 processors (2 x 20 cores, at 2.1 GHz)	2 x 2nd Gen Intel Xeon Processor Scalable Family 5220 (2 x 18 cores, 2.2 GHz)
Memory	12 x 32GB DDR4 (384 GB)	12 x 32GB DDR4 (384 GB)	12 x 32GB DDR4 (384 GB)
Boot	M.2 with 2 x 240-GB SSDs	M.2 with 2 x 240-GB SSDs	2 x 240-GB SATA SSDs
Storage	26 x 2.4TB 10K rpm SFF SAS HDDs or 12 x 1.6-TB Enterprise Value SATA SSDs	12 x 8-TB 7.2K rpm LFF SAS HDDs	28 x 6 TB 7.2K rpm LFF SAS HDDs
Virtual interface card (VIC)	40 Gigabit Ethernet (Cisco UCS VIC 1387) or 25 Gigabit Ethernet (Cisco UCS VIC 1455)	40 Gigabit Ethernet (Cisco UCS VIC 1387) or 25 Gigabit Ethernet (Cisco UCS VIC 1455)	40 Gigabit Ethernet (Cisco UCS VIC 1387)
Storage controller	Cisco 12-Gbps SAS modular RAID controller with 4-GB flash-based write cache (FBWC) or Cisco 12-Gbps modular SAS host bus adapter (HBA)	Cisco 12-Gbps SAS modular RAID controller with 2-GB FBWC or Cisco 12-Gbps modular SAS host bus adapter (HBA)	Cisco 12-Gbps SAS Modular RAID Controller with 4-GB flash-based write cache (FBWC)
Network connectivity	Cisco UCS 6332 Fabric Interconnect or Cisco UCS 6454 Fabric Interconnect	Cisco UCS 6332 Fabric Interconnect or Cisco UCS 6454 Fabric Interconnect	Cisco UCS 6332 Fabric Interconnect
GPU (optional)	2 x NVIDIA TESLA V100 with 32 GB of memory each	2 x NVIDIA TESLA V100 with 32 GB of memory each	

AI Computing Farm Reference Architecture

Table 2 lists the AI computing farm reference architecture configuration details for high-density CPU cores and GPU nodes.

Table 2 High-density CPU Cores and GPU Nodes

	Select stack	Elite stack	Premier stack

	Select stack	Elite stack	Premier stack
Servers	8 x Cisco UCS C240 M5 Rack Servers 4 x Cisco UCS C480 M5 Rack Servers	8 x Cisco UCS C240 M5 Rack Servers 4 x Cisco UCS C480 ML M5 Rack Servers	8 x Cisco UCS C4200 Rack Servers Each with: 4 x Cisco UCS C125 M5 Rack Servers
CPU	2 x 2nd Gen Intel Xeon Scalable 6230 processors (2 x 20 cores, at 2.1 GHz)	2 x Intel Xeon Scalable 6132 processors (2 x 16 cores, at 2.6 GHz)	2 x AMD EPYC 7401 processors (2 x 24 cores, at 2.0 or 2.8 GHz)
Memory	12 x 32GB DDR4 (384 GB)	12 x 32GB DDR4 (384 GB)	16 x 32GB DDR4 (512 GB)
Boot	M.2 with 2 x 960GB SSDs	M.2 with 2 x 960GB SSDs	M.2 with 2 x 240GB SATA SSDs
Storage	24 x 1.8-TB 10K rpm SFF SAS HDDs or 12 x 1.6-TB Enterprise Value SATA SSDs	24 x 1.8-TB 10K rpm SFF SAS HDDs or 12 x 1.6-TB Enterprise Value SATA SSDs	6 x 3.8-TB Enterprise Value SATA SSDs
VIC	40 Gigabit Ethernet (Cisco UCS VIC 1387) or 25 Gigabit Ethernet (Cisco UCS VIC 1455)	40 Gigabit Ethernet (Cisco UCS VIC 1387) or 25 Gigabit Ethernet (Cisco UCS VIC 1455)	25 Gigabit Ethernet (Cisco UCS VIC 1455)
Storage controller	Cisco 12-Gbps SAS modular RAID controller with 4-GB FBWC or Cisco 12-Gbps modular SAS HBA	Cisco 12-Gbps SAS modular RAID controller with 4-GB FBWC or Cisco 12-Gbps modular SAS HBA	Cisco 12-Gbps SAS 9460-8i RAID controller with 2-GB FBWC
Network connectivity	Cisco UCS 6332 Fabric Interconnect or Cisco UCS 6454 Fabric Interconnect	Cisco UCS 6332 Fabric Interconnect or Cisco UCS 6454 Fabric Interconnect	Cisco UCS 6454 Fabric Interconnect
GPU	For C240 M5: 2 x NVIDIA TESLA V100 with 32-GB memory each or up to 6 x NVIDIA T4 For C480 M5: 4 x NVIDIA TESLA v100 with 32-GB memory each or 4 x NVIDIA T4	For C240 M5: 2 x NVIDIA TESLA V100 with 32-GB memory each or up to 6 x NVIDIA T4 For C480 M5 ML: 8 x NVIDIA TESLA V100 with 32-GB memory each and with NVLink	

High density GPU servers have higher storage for OS M.2 drives for docker volumes on the OS drives.

Tiered Storage

Table 3 lists the tiered storage reference architecture configuration details for Cisco UCS Integrated Infrastructure for Big Data and Analytics.

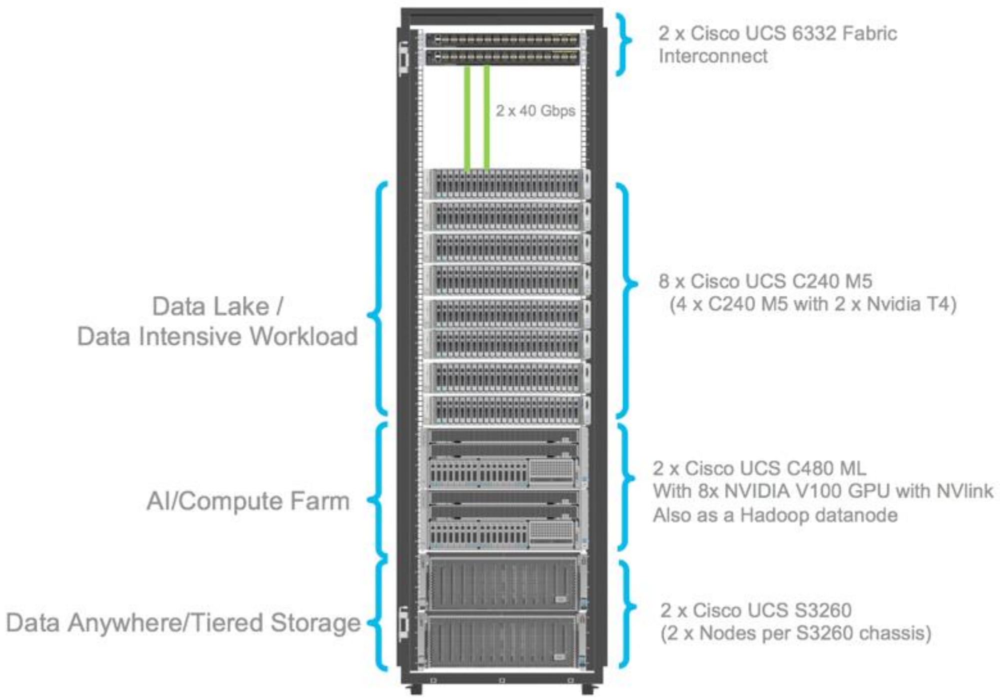
Table 3 Cisco UCS Integrated Infrastructure for Big Data and Analytics Configuration Options

	High capacity
Servers	8 x Cisco UCS S3260 Storage Servers
CPU	2 x Intel Xeon Scalable 5220 processors (2 x 18 cores, at 2.2 GHz)
Memory	12 x 32GB 2666 MHz (384 GB)
Boot	2 x 240GB SATA Boot SSDs
Storage	28 x 6TB 7.2K rpm LFF SAS HDDs
VIC	40 Gigabit Ethernet (Cisco UCS VIC 1387)
Storage controller	Cisco UCS S3260 dual RAID controller
Network connectivity	Cisco UCS 6332 Fabric Interconnect

This configuration can also be deployed with 4th Generation Cisco UCS 6454 Fabric Interconnect with 25G VIC. However, this could lead to a performance slow down compared to a 40G VIC and Fabric Interconnect 6332.

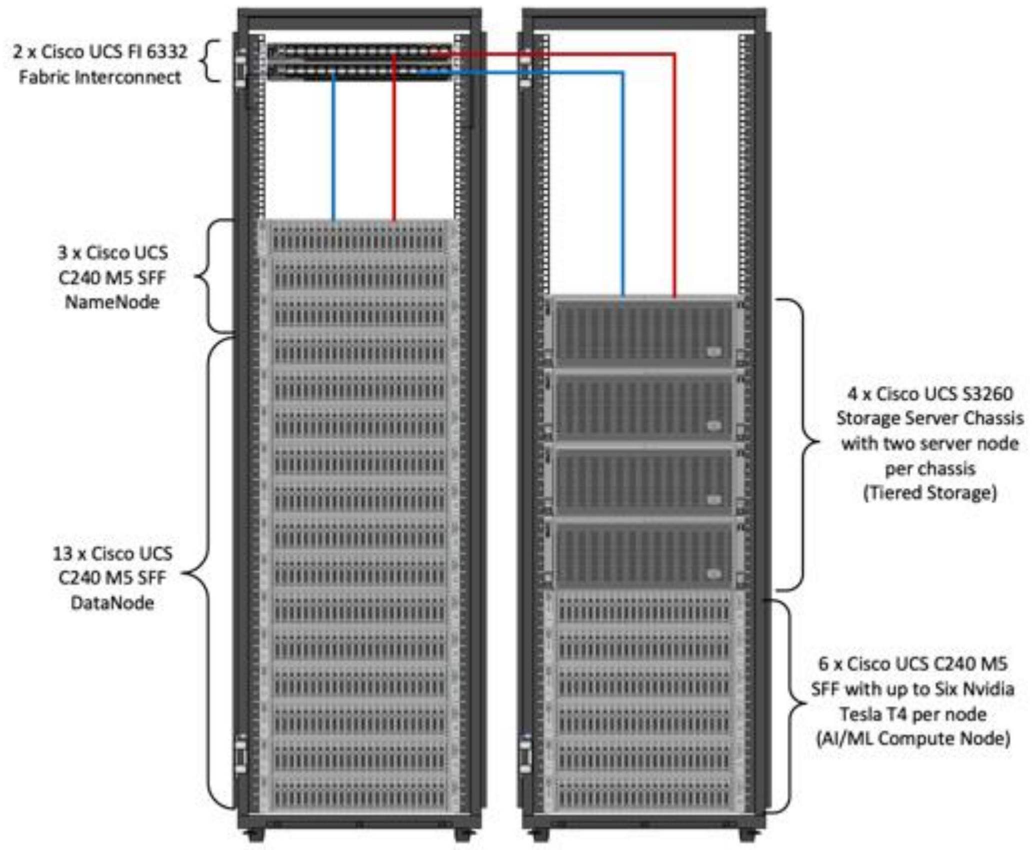
Figure 4 illustrates a 12-node starter cluster with all the 3 components in a single Rack. The top 8 node has Cisco UCS C240 M5 servers as a data lake. Each link in the figure represents a 40 Gigabit Ethernet link from each of the 12 servers directly connected to a Fabric Interconnect. The second 2 x Cisco UCS C480 ML M5 Servers and the last 4 servers illustrate a data tiering on 2xS3260 servers. Every server is connected to both Fabric Interconnects.

Figure 4 Single Rack Starter Cluster Topology


2 x Cisco UCS 6332 Fabric Interconnect
2 x 40 Gbps
Data Lake / Data Intensive Workload
8 x Cisco UCS C240 M5 (4 x C240 M5 with 2 x Nvidia T4)
AI/Compute Farm
2 x Cisco UCS C480 ML With 8x NVIDIA V100 GPU with NVlink Also as a Hadoop datanode
Data Anywhere/Tiered Storage
2 x Cisco UCS S3260 (2 x Nodes per S3260 chassis)

As illustrated in Figure 5, a 30-node starter cluster. Rack #1 has sixteen Cisco UCS C240 M5 servers. Each link in the figure represents a 40 Gigabit Ethernet link from each of the sixteen servers directly connected to a Fabric Interconnect. Rack #2 has six Cisco UCS C240 M5 and four Cisco UCS S3260 servers. Every server is connected to both Fabric Interconnects.

Figure 5 Cisco Data Intelligence Platform – 30 Node Configuration with CDH 6.2 and CDSW 1.5



An alternate configuration for cases where more GPU capacity is needed. Four of the Cisco UCS C240 M5 servers from the previous configuration in Figure 5 are replaced with Cisco UCS C480 M5 ML M5 server which support up to eight V100 MXM GPUs.

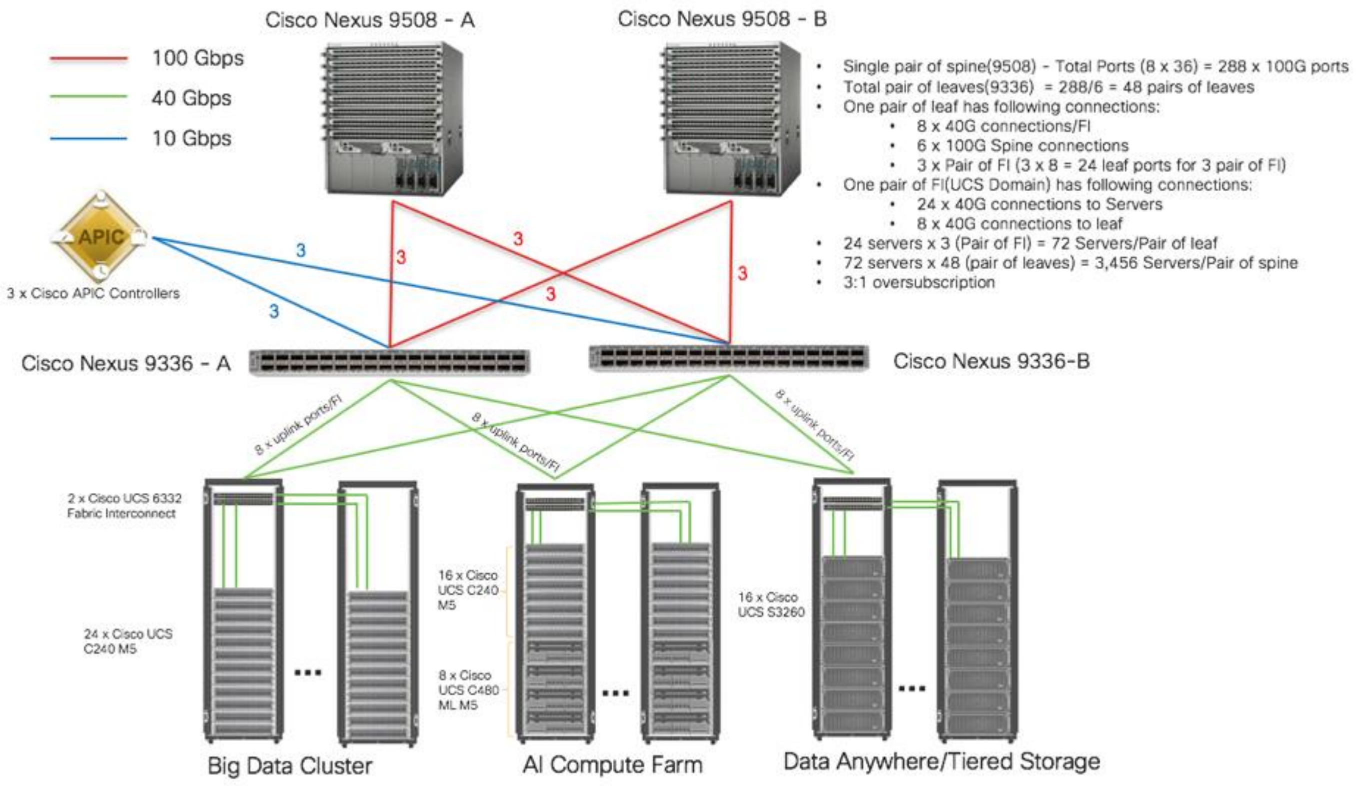
Each Cisco UCS C480 ML M5 has 8 x NVIDIA SXM2 V100 32GB modules with NVLink interconnect. Each Cisco UCS C240 M5 supports up to two PCIe GPU adapters with NVIDIA Tesla V100. For more information about Cisco UCS C240 M5 Sever installation and GPU card configuration rules, go to https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c/hw/C240M5/install/C240M5/C240M5_appendix_0101.html

Power requirements per rack must be calculated since the exact values will change based on the power needs of the GPUs.

Scaling the Solution

Figure 6 illustrates how to scale the solution. Each pair of Cisco UCS 6332 Fabric Interconnects has 24 Cisco UCS C240 M5 servers connected to it. This allows for eight uplinks from each Fabric Interconnect to the Cisco Nexus 9332 switch. Six pairs of 6332 FI's can connect to a single switch with four uplink ports each. With 24 servers per FI, a total of 144 servers can be supported. Additionally, the can scale to thousands of nodes with the Nexus 9500 series family of switches.

Figure 6 Scaling the Solution



- Single pair of spine(9508) - Total Ports (8 x 36) = 288 x 100G ports
- Total pair of leaves(9336) = 288/6 = 48 pairs of leaves
- One pair of leaf has following connections:
  - 8 x 40G connections/FI
  - 6 x 100G Spine connections
  - 3 x Pair of FI (3 x 8 = 24 leaf ports for 3 pair of FI)
- One pair of FI(UCS Domain) has following connections:
  - 24 x 40G connections to Servers
  - 8 x 40G connections to leaf
- 24 servers x 3 (Pair of FI) = 72 Servers/Pair of leaf
- 72 servers x 48 (pair of leaves) = 3,456 Servers/Pair of spine
- 3:1 oversubscription

In the reference architectures discussed here, each of the components is scaled separately, and for the purposes of this example, scaling is uniform. Two scale scenarios are as follows:

- Scaled architecture with 3:1 oversubscription with Cisco fabric interconnects and Cisco ACI
- Scaled architecture with 2:1 oversubscription with Cisco ACI

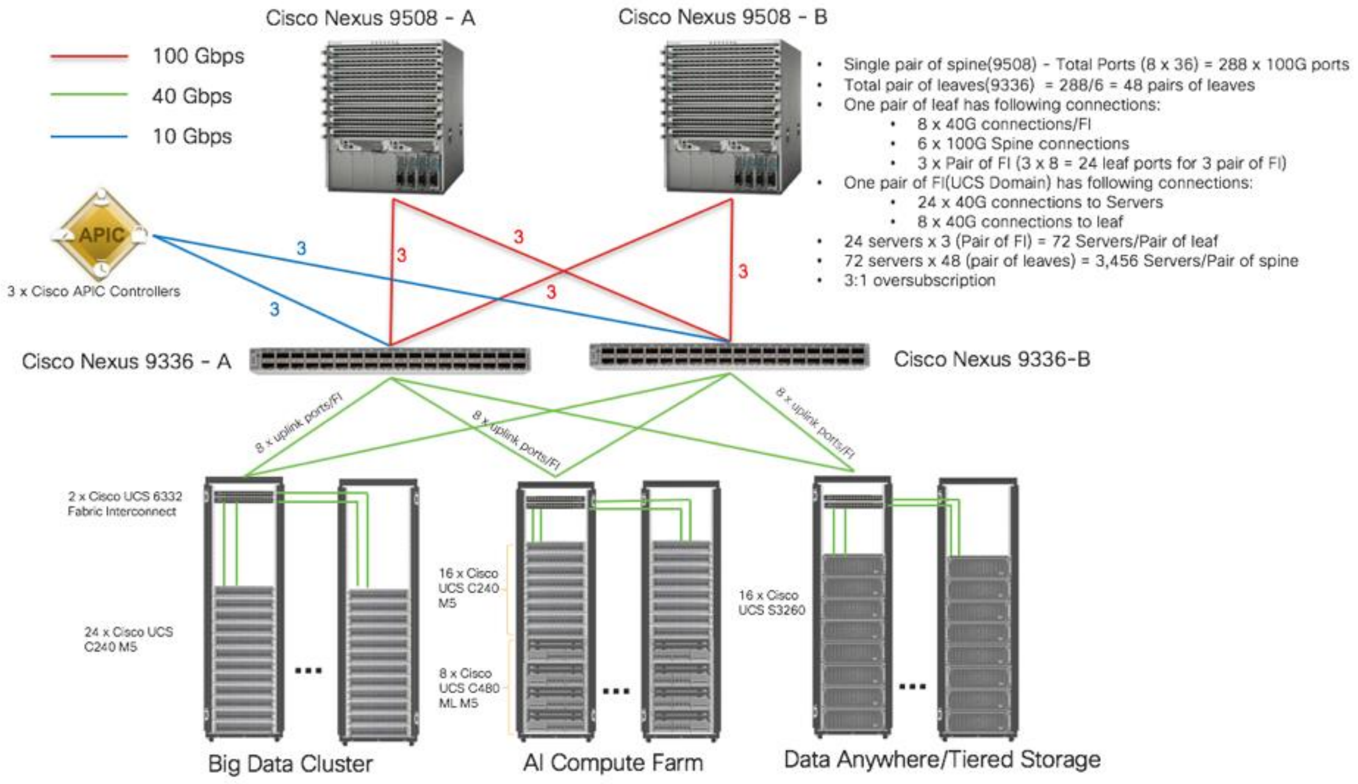
In the following scenarios, the goal is to populate up to a maximum of 200 leaf nodes in a Cisco ACI domain. Not all cases reach that number because they use the Cisco Nexus® 9508 Switch for this sizing and not the Cisco Nexus 9516 Switch.

Scaled Architecture with 3:1 Oversubscription with Cisco Fabric Interconnects and Cisco ACI

The architecture discussed here and shown in Figure 7 supports 3:1 network oversubscription from every node to every other node across a multidomain cluster (nodes in a single domain within a pair of Cisco fabric interconnects are locally switched and not oversubscribed).

From the viewpoint of the data lake, 24 Cisco UCS C240 M5 Rack Servers are connected to a pair of Cisco UCS 6332 Fabric Interconnects (with 24 x 40-Gbps throughput). From each fabric interconnect, 8 x 40-Gbps links connect to a pair of Cisco Nexus 9336 Switches. Three pairs of fabric interconnects can connect to a single pair of Cisco Nexus 9336 Switches (8 x 40-Gbps links per Fabric Interconnect to Nexus switch). Each of these Cisco Nexus 9336 Switches connects to a pair of Cisco Nexus 9508 Cisco ACI switches with 6 x 100-Gbps uplinks (connecting to a Cisco N9K-X9736C-FX line card). the Cisco Nexus 9508 Switch with the Cisco N9K-X9736C-FX line card can support up to 36 x 100-Gbps ports, each and 8 such line cards.

Figure 7 Scaled Architecture with 3:1 Oversubscription with Cisco Fabric Interconnects and Cisco ACI


Cisco Nexus 9508 - A
Cisco Nexus 9508 - B
100 Gbps
40 Gbps
10 Gbps
APIC
3 x Cisco APIC Controllers
3
3
3
3
3
3
Single pair of spine(9508) - Total Ports (8 x 36) = 288 x 100G ports
Total pair of leaves(9336) = 288/6 = 48 pairs of leaves
One pair of leaf has following connections:
8 x 40G connections/FI
6 x 100G Spine connections
3 x Pair of FI (3 x 8 = 24 leaf ports for 3 pair of FI)
One pair of FI(UCS Domain) has following connections:
24 x 40G connections to Servers
8 x 40G connections to leaf
24 servers x 3 (Pair of FI) = 72 Servers/Pair of leaf
72 servers x 48 (pair of leaves) = 3,456 Servers/Pair of spine
3:1 oversubscription
Cisco Nexus 9336 - A
Cisco Nexus 9336-B
8 x uplink ports/FI
8 x uplink ports/FI
8 x uplink ports/FI
2 x Cisco UCS 6332 Fabric Interconnect
24 x Cisco UCS C240 M5
16 x Cisco UCS C240 M5
8 x Cisco UCS C480 ML M5
16 x Cisco UCS S3260
Big Data Cluster
AI Compute Farm
Data Anywhere/Tiered Storage

Scaled Architecture with 2:1 Oversubscription with Cisco ACI

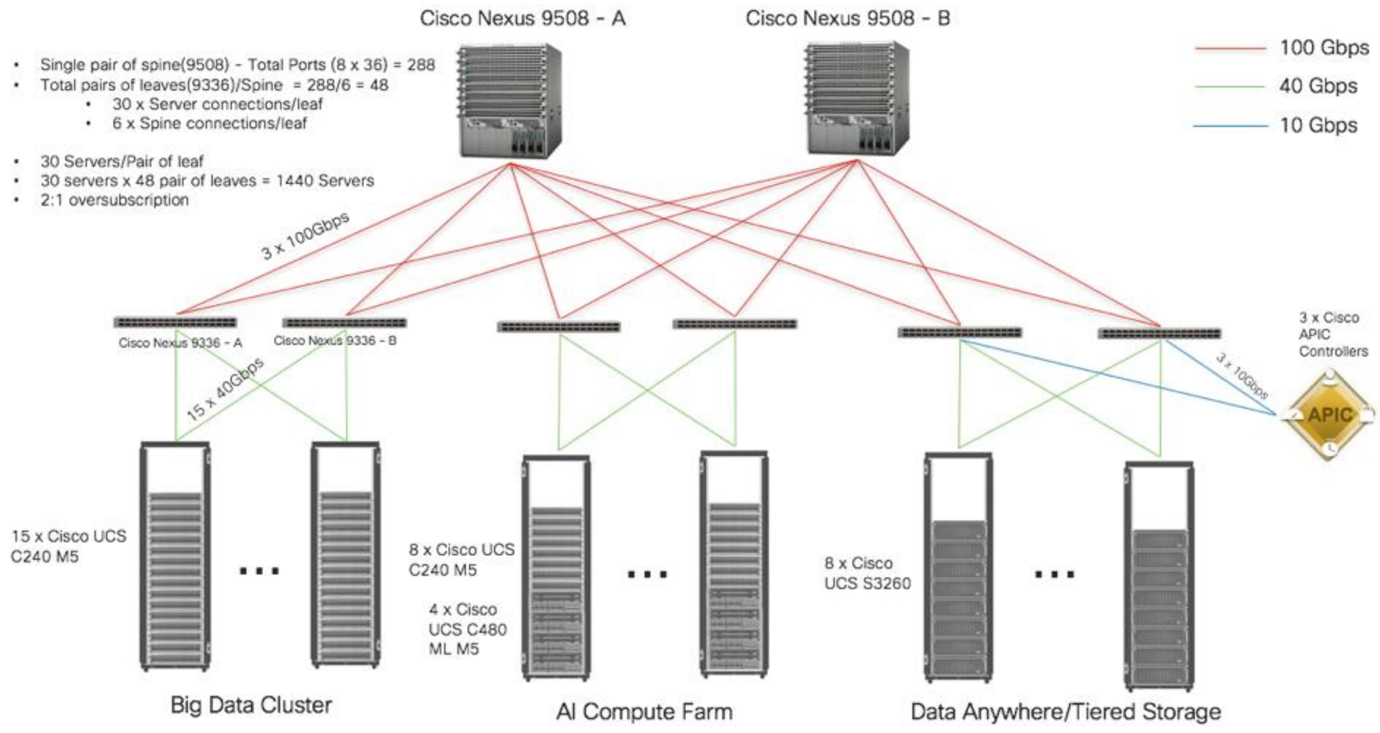
In the scenario discussed here and shown in Figure 8, the Cisco Nexus 9508 Switch with the Cisco N9K-X9736C-FX line card can support up to 36 x 100-Gbps ports, each and 8 such line cards.

Here, for the 2:1 oversubscription, 30 Cisco UCS C240 M5 Rack Servers are connected to a pair of Cisco Nexus 9336 Switches, and each Cisco Nexus 9336 connects to a pair of Cisco Nexus 9508 Switches with three uplinks each. A pair of Cisco Nexus 9336 Switches can support 30 servers and connect to a spine with 6 x 100-Gbps links on each spine. This single pod (pair of Cisco Nexus 9336 Switches connecting to 30 Cisco UCS C240 M5 servers and 6 uplinks to each spine) can be repeated 48 times (288/6) for a given Cisco Nexus 9508 Switch and can support up to1440 servers.

To reduce the oversubscription ratio (to get 1:1 network subscription from any node to any node), you can use just 15 servers under a pair of Cisco Nexus 9336 Switches and then move to Cisco Nexus 9516 Switches (the number of leaf nodes would double).

To scale beyond this number, multiple spines can be aggregated.

Figure 8 Scaled Architecture with 2:1 Oversubscription with Cisco ACI


Cisco Nexus 9508 - A
Cisco Nexus 9508 - B
• Single pair of spine(9508) - Total Ports (8 x 36) = 288
• Total pairs of leaves(9336)/Spine = 288/6 = 48
• 30 x Server connections/leaf
• 6 x Spine connections/leaf
• 30 Servers/Pair of leaf
• 30 servers x 48 pair of leaves = 1440 Servers
• 2:1 oversubscription
100 Gbps
40 Gbps
10 Gbps
3 x 100Gbps
Cisco Nexus 9336 - A
Cisco Nexus 9336 - B
15 x 40Gbps
3 x Cisco APIC Controllers
3 x 10Gbps
APIC
15 x Cisco UCS C240 M5
8 x Cisco UCS C240 M5
4 x Cisco UCS C480 ML M5
8 x Cisco UCS S3260
Big Data Cluster
AI Compute Farm
Data Anywhere/Tiered Storage

Technology Overview

Cisco UCS Integrated Infrastructure for Big Data and Analytics

The Cisco UCS Integrated Infrastructure for Big Data and Analytics solution for Cloudera is based on Cisco UCS Integrated Infrastructure for Big Data and Analytics, a highly scalable architecture designed to meet a variety of scale-out application demands with seamless data integration and management integration capabilities built using the components described in this section.

Cisco UCS Manager

Cisco UCS Manager (UCSM) resides within the Cisco UCS Fabric Interconnect. It makes the system self-aware and self-integrating, managing all the system components as a single logical entity. Cisco UCS Manager can be accessed through an intuitive graphical user interface (GUI), a command-line interface (CLI), or an XML application-programming interface (API). Cisco UCS Manager uses service profiles to define the personality, configuration, and connectivity of all resources within Cisco UCS, radically simplifying provisioning of resources so that the process takes minutes instead of days. This simplification allows IT departments to shift their focus from constant maintenance to strategic business initiatives.

Key Features

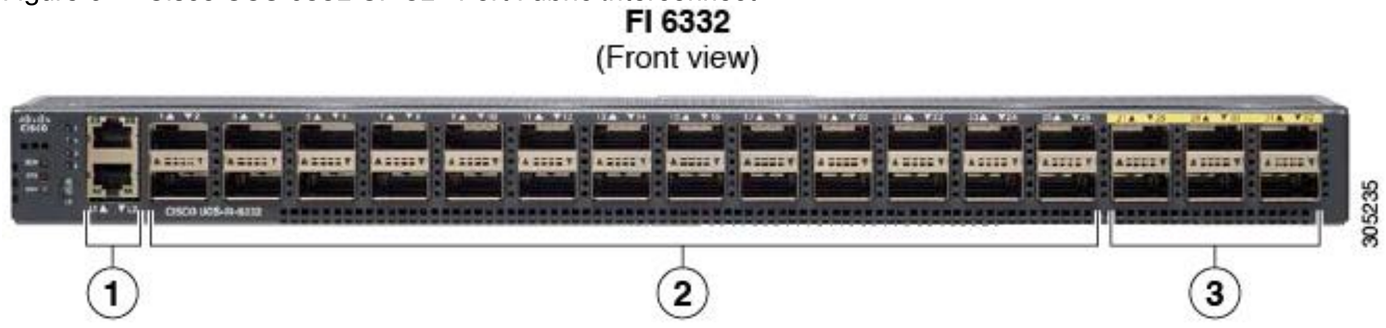
- Supports Cisco UCS B-Series Blade and Cisco UCS C-Series Rack Servers, the Cisco UCS C3260 storage server, Cisco UCS Mini, and the Cisco HyperFlex hyperconverged infrastructure.
- Programmatically controls server, network, and storage resources, with a unified, policy-driven management, so they can be efficiently managed at scale through software.
- Works with HTML 5, Java, or CLI graphical user interfaces.
- Can automatically detect, inventory, manage, and provision system components that are added or changed.
- Facilitates integration with third-party systems management tools.
- Builds on existing skills and supports collaboration across disciplines through role-based administration

Cisco UCS 6300 Series Fabric Interconnects

Cisco UCS 6300 Series Fabric Interconnects provide high-bandwidth, low-latency connectivity for servers, with integrated, unified management provided for all connected devices by Cisco UCS Manager. Deployed in redundant pairs, Cisco fabric interconnects offer the full active-active redundancy, performance, and exceptional scalability needed to support the large number of nodes that are typical in clusters serving big data applications. Cisco UCS Manager enables rapid and consistent server configuration using service profiles, automating ongoing system maintenance activities such as firmware updates across the entire cluster as a single operation. Cisco UCS Manager also offers advanced monitoring with options to raise alarms and send notifications about the health of the entire cluster.

The Cisco UCS 6300 series Fabric interconnects are a core part of Cisco UCS, providing low-latency, lossless 10 and 40 Gigabit Ethernet, Fiber Channel over Ethernet (FCoE), and Fiber Channel functions with management capabilities for the entire system. All servers attached to Fabric interconnects become part of a single, highly available management domain.

Figure 9 Cisco UCS 6332 UP 32 -Port Fabric Interconnect



Cisco UCS C-Series Rack-Mount Servers

Cisco UCS C-Series Rack-Mount Servers keep pace with Intel Xeon processor innovation by offering the latest processors with increased processor frequency and improved security and availability features. With the increased performance provided by the Intel Xeon Scalable Family Processors, Cisco UCS C-Series servers offer an improved price-to-performance ratio. They also extend Cisco UCS innovations to an industry-standard rack-mount form factor, including a standards-based unified network fabric, Cisco VN-Link virtualization support, and Cisco Extended Memory Technology.

It is designed to operate both in standalone environments and as part of Cisco UCS managed configuration, these servers enable organizations to deploy systems incrementally—using as many or as few servers as needed—on a schedule that best meets the organization's timing and budget. Cisco UCS C-Series servers offer investment protection through the capability to deploy them either as standalone servers or as part of Cisco UCS. One compelling reason that many organizations prefer rack-mount servers is the wide range of I/O options available in the form of PCIe adapters. Cisco UCS C-Series servers support a broad range of I/O options, including interfaces supported by Cisco and adapters from third parties.

Cisco UCS C240 M5 Rack-Mount Server

The Cisco UCS C240 M5 Rack-Mount Server (Figure 10) is a 2-socket, 2-Rack-Unit (2RU) rack server offering industry-leading performance and expandability. It supports a wide range of storage and I/O-intensive infrastructure workloads, from big data and analytics to collaboration. Cisco UCS C-Series Rack Servers can be deployed as standalone servers or as part of a Cisco Unified Computing System (Cisco UCS) managed environment to take advantage of Cisco's standards-based unified computing innovations that help reduce customers' Total Cost of Ownership (TCO) and increase their business agility.

In response to ever-increasing computing and data-intensive real-time workloads, the enterprise-class Cisco UCS C240 M5 server extends the capabilities of the Cisco UCS portfolio in a 2RU form factor. It incorporates the 2nd generation Intel® Xeon® Scalable and Intel® Xeon® Scalable processors, supporting up to 20 percent more cores per socket, twice the memory capacity, and five times more Non-Volatile Memory Express (NVMe) PCI Express (PCIe) Solid-State Disks (SSDs) compared to the previous generation of servers. These improvements deliver significant performance and efficiency gains that will improve your application performance. The Cisco UCS C240 M5 delivers outstanding levels of storage expandability with exceptional performance, along with the following:

- Latest Intel Xeon Scalable CPUs with up to 28 cores per socket
- Up to 24 DDR4 DIMMs for improved performance

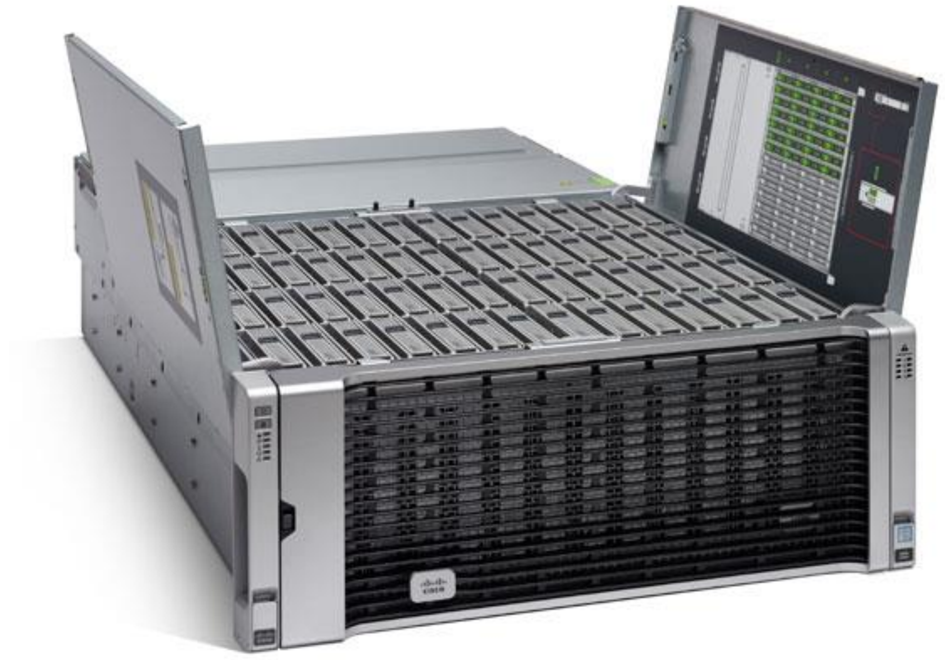
- Up to 26 hot-swappable Small-Form-Factor (SFF) 2.5-inch drives, including 2 rear hot-swappable SFF drives (up to 10 support NVMe PCIe SSDs on the NVMe-optimized chassis version), or 12 Large-Form-Factor (LFF) 3.5-inch drives plus 2 rear hot-swappable SFF drives
- Support for 12-Gbps SAS modular RAID controller in a dedicated slot, leaving the remaining PCIe Generation 3.0 slots available for other expansion cards
- Modular LAN-On-Motherboard (mLOM) slot that can be used to install a Cisco UCS Virtual Interface Card (VIC) without consuming a PCIe slot, supporting dual 10- or 40-Gbps network connectivity
- Dual embedded Intel x550 10GBASE-T LAN-On-Motherboard (LOM) ports
- Modular M.2 or Secure Digital (SD) cards that can be used for boot

Figure 10 Cisco UCS C240 M5 Rack-Mount Server

Cisco UCS S3260 Storage Servers

The Cisco UCS S3260 Storage Server is a modular storage server with dual M5 server nodes and is optimized to deliver efficient, industry-leading storage for data-intensive workloads. The Cisco UCS S3260 server with dual-node capability that is based on the 2nd Gen Intel® Xeon® Scalable and Intel® Xeon® Scalable processors, the server features up to 840 TB of local storage in a compact 4-Rack-Unit (4RU) form factor. The drives can be configured with enterprise-class Redundant Array of Independent Disks (RAID) redundancy or with a pass-through Host Bus Adapter (HBA) controller. Network connectivity is provided with dual-port 40-Gbps nodes in each server, with expanded unified I/O capabilities for data migration between Network-Attached Storage (NAS) and SAN environments. This storage-optimized server comfortably fits in a standard 32-inch-depth rack, such as the Cisco® R 42610 Rack.

Figure 11 Cisco UCS S3260 Storage Server



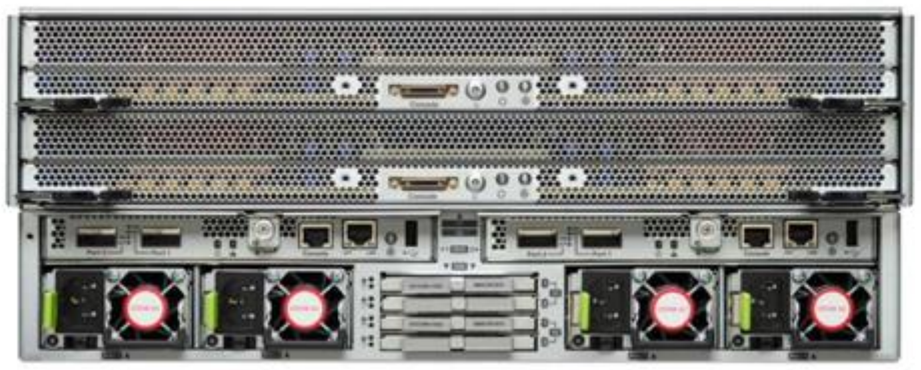
The Cisco UCS S3260 Storage Server chassis has 56 top-load LFF HDDs option as shown above with a maximum capacity of 4 TB per HDD and can be mixed with up to 28 SSDs.

The modular Cisco UCS S3260 Storage Server chassis offers flexibility with more computing, storage, and PCIe expansion on the second slot in the chassis. This second slot can be used for:

- An additional server node
- Four additional LFF HDDs with up to 10 TB capacity per HDD
- New PCIe expansion tray with up to two x8 half-height, half-width PCIe slots that can use any industry-standard PCIe card including Fibre Channel and Ethernet cards.

The Cisco UCS S3260 Storage Server Chassis includes a Cisco UCS Virtual Interface Card (VIC) 1300 platform chip onboard the system I/O controller, offering high-performance bandwidth with dual-port 40 Gigabit Ethernet and FCoE interfaces per system I/O controller.

Figure 12 Cisco UCS S3260 Storage Server: Rear View



Cisco UCS C480 M5 Rack-Mount Server

The Cisco UCS C480 M5 Rack-Mount Server is a storage and I/O-optimized enterprise-class rack-mount server that delivers industry-leading performance for in-memory databases, big data analytics, virtualization, Virtual Desktop Infrastructure (VDI), and bare-metal applications. The Cisco UCS C480 M5 (Figure 13) delivers outstanding levels of expandability and performance for standalone or Cisco Unified Computing System managed environments in a 4RU form-factor. Because of its modular design, you pay for only what you need. It offers these capabilities:

- Latest 2nd Gen Intel® Xeon® Scalable and Intel® Xeon® processors and support for two-or four-processor configurations
- 2666-MHz DDR4 memory and 48 DIMM slots for up to 6 Terabytes (TB) of total memory
- 12 PCI Express (PCIe) 3.0 slots
  - Six x 8 full-height, full length slots
  - Six x16 full-height, full length slots
- Flexible storage options with support up to 32 Small-Form-Factor (SFF) 2.5-inch, SAS, SATA, and PCIe NVMe disk drives
- Cisco 12-Gbps SAS Modular RAID Controller in a dedicated slot
- Internal Secure Digital (SD) and M.2 boot options
- Dual embedded 10 Gigabit Ethernet LAN-On-Motherboard (LOM) ports

Figure 13 Cisco UCS C480 M5 Rack-Mount Server – Front View

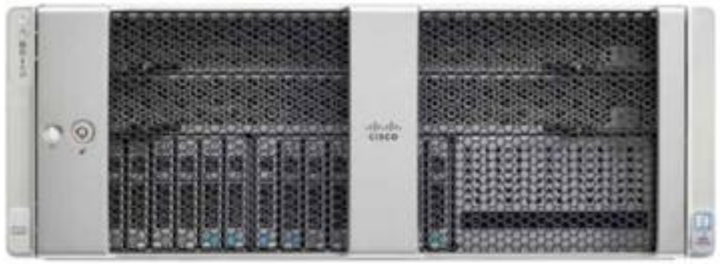
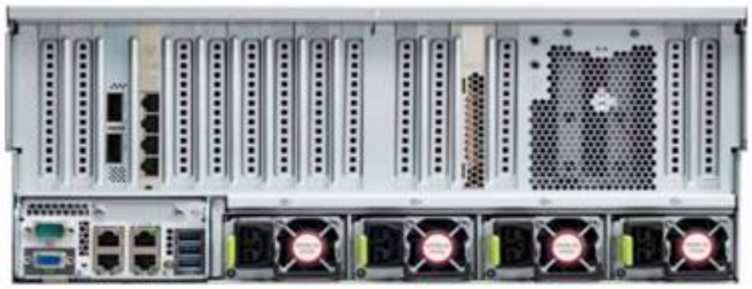


Figure 14 Cisco UCS C480 M5 Rack-Mount Server – Rear View



For more information about Cisco UCS C480 M5 Rack Server, go to: https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/datasheet-c78-739291.html

Cisco UCS C480 ML M5 Rack Server

The Cisco UCS C480 ML M5 Rack Server is a purpose-built server for Deep Learning. It is storage and I/O optimized to deliver an industry-leading performance for training Models. The Cisco UCS C480 ML M5 delivers outstanding levels of storage expandability and performance options for standalone or Cisco Unified Computing System managed environments in a 4RU form factor. Because of its modular design, you pay for only what you need. It offers these capabilities:

- 8 NVIDIA SXM2 V100 32G modules with NVLink interconnect
- Dual socket latest 2nd Gen Intel® Xeon® Scalable and Intel® Xeon® processors
- 2666-MHz DDR4 memory and 24 DIMM slots for up to 3 terabytes (TB) of total memory
- 4 PCI Express (PCIe) 3.0 slots for 100G Cisco UCS VIC 1495
- Flexible storage options with support for up to 24 Small-Form-Factor (SFF) 2.5-inch, SAS/SATA Solid-State Disks (SSDs) and Hard-Disk Drives (HDDs)
- Up to 6 PCIe NVMe disk drives
- Cisco 12-Gbps SAS Modular RAID Controller in a dedicated slot
- M.2 boot options
- Dual embedded 10 Gigabit Ethernet LAN-On-Motherboard (LOM) ports

Figure 15 Cisco UCS C480 ML M5 Purpose Built Deep Learning Server – Front View

Figure 16 Cisco UCS C480 ML M5 Purpose Built Deep Learning Server – Rear View

For more information about Cisco UCS C480 ML M5 Server, review the specification sheet: https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/c480m5-specsheet-ml-m5-server.pdf

Table 4 lists the features and benefits of Cisco UCS C480 ML M5 Server.

Table 4 Features and Benefits of Cisco UCS C480 ML M5 Server

Feature	Benefits

Feature	Benefits
8 x NVIDIA SXM2 V100 32GB modules with NVLink interconnect	Fast Deep Learning model training
Modular storage support with up to 24 front accessible hot-swappable Hard Disk Drives (HDDs) and Solid-State Disks (SSDs)	Modularity to right-size storage options to match training requirements Flexibility to expand as storage needs increase
High-capacity memory support of up to 3 TB using 128-GB DIMMs	Large memory footprint to deliver performance and capacity for large model training
Up to 6 PCIe NVMe drives	Up to 6 Gen3 x4 lanes NVMe drives for extreme I/O performance for faster model training
Support for up to 4 PCIe Generation 3.0 slots	Support for up to four 10/25 or 40/100G Cisco VICs
Hot-swappable, redundant power supplies	Increased high availability
Integrated dual 10-Gbps Ethernet	Increased network I/O performance and additional network options

Cisco UCS Virtual Interface Cards (VICs)

Cisco UCS Virtual Interface Cards (VIC) are unique to Cisco. Cisco UCS Virtual Interface Cards incorporate next-generation converged network adapter (CNA) technology from Cisco and offer dual 10- and 40-Gbps ports designed for use with Cisco UCS servers. Optimized for virtualized networking, these cards deliver high performance and bandwidth utilization, and support up to 256 virtual devices.

The Cisco UCS Virtual Interface Card 1387 offers dual-port Enhanced Quad Small Form-Factor Pluggable (QSFP+) 40 Gigabit Ethernet and Fiber Channel over Ethernet (FCoE) in a modular-LAN-on-motherboard (mLOM) form factor. The mLOM slot can be used to install a Cisco VIC without consuming a PCIe slot providing greater I/O expandability.

Figure 17 Cisco UCS VIC 1387

NVIDIA GPU

Graphics Processing Units or GPUs are specialized processors designed to render images, animation and video for computer displays. They perform this task by running many operations simultaneously. While the number and kinds of operations they can do are limited, they make up for it by being able run many thousands in parallel. As the graphics capabilities of GPUs increased, it soon became apparent that the massive parallelism of GPUs could be put to other uses beside rendering graphics.

NVIDIA GPU used in this document, NVIDIA Tesla V100, is advanced data center GPU built to accelerate AI, HPC, and graphics. It is powered by NVIDIA Volta architecture, comes in 16 and 32 GB configurations.

NVIDIA GPUs bring two key advantages to the table. First, they make possible solutions that were simply not computationally possible before. Second, by providing the same processing power as scores of traditional CPUs they reduce the requirements for rack space, power, networking and cooling in the data center.

NVIDIA CUDA

GPUs are very good at running the same operation on different data simultaneously. This is often referred to as single instruction, multiple data, or SIMD. This is exactly what's needed to render graphics, but many other computing problems can benefit from this approach. As a result, NVIDIA created CUDA. CUDA is a parallel computing platform and programming model that makes it possible to use a GPU for many general-purpose computing tasks via commonly used programming languages like C and C++.

In addition to the general-purpose computing capabilities that CUDA enables there is also a special CUDA library for deep learning called the CUDA Deep Neural Network library, or cuDNN. cuDNN makes it easier to implement deep machine learning architectures that take full advantage of the GPU's capabilities.

Cloudera Enterprise Data Hub and Hortonworks Data Platform

The CVDs implement with Cloudera Enterprise Data Hub.

Cloudera (CDH 6.2.0)

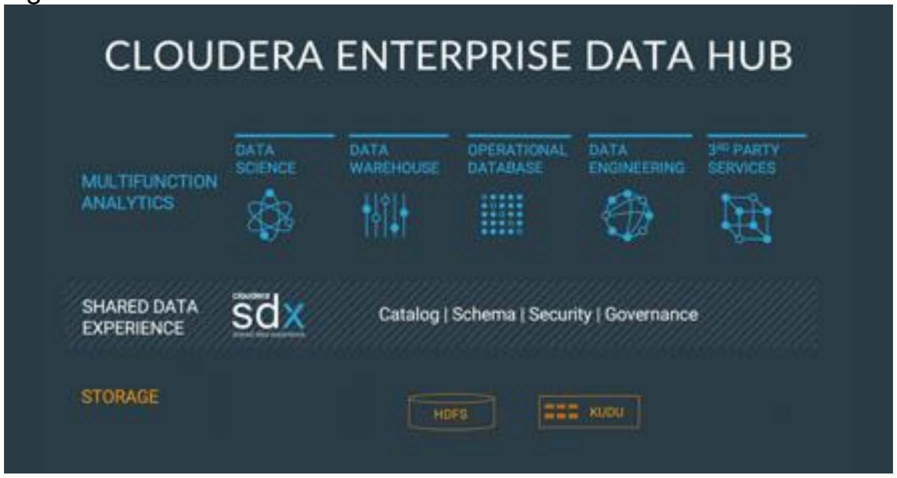
Built on the transformative Apache Hadoop open source software project, Cloudera Enterprise is a hardened distribution of Apache Hadoop and related projects designed for the demanding requirements of enterprise customers. Cloudera is the leading contributor to the Hadoop ecosystem, and has created a rich suite of complementary open source projects that are included in Cloudera Enterprise.

All the integration and the entire solution is thoroughly tested and fully documented. By taking the guesswork out of building out a Hadoop deployment, CDH gives a streamlined path to success in solving real business problems.

Cloudera Enterprise with Apache Hadoop is:

- Unified – one integrated system, bringing diverse users and application workloads to one pool of data on common infrastructure; no data movement required
- Secure – perimeter security, authentication, granular authorization, and data-protection
- Governed – enterprise-grade data auditing, data lineage, and data-discovery
- Managed – native high-availability, fault-tolerance and self-healing storage, automated backup and disaster recovery, and advanced system and data management
- Open – Apache-licensed open source to ensure both data and applications remain copy righted, and an open platform to connect with all the existing investments in technology and skills.

Figure 18 Cloudera Data Hub



Cloudera provides the following products and tools:

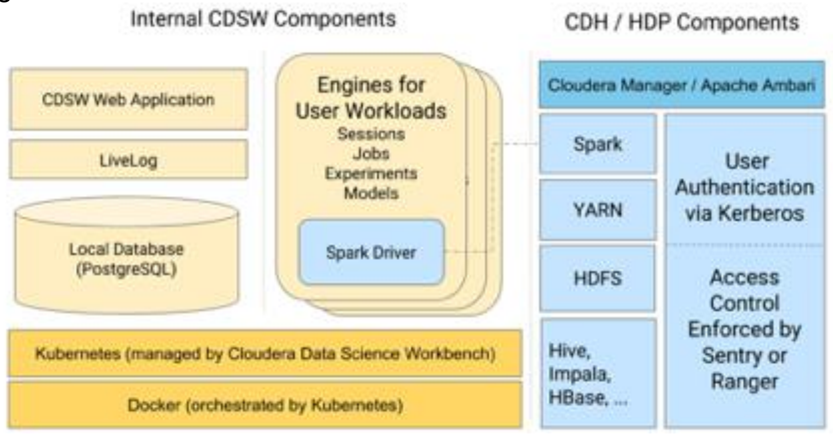
- CDH—The Cloudera distribution of Apache Hadoop and other related open-source projects, including Spark. CDH also provides security and integration with numerous hardware and software solutions.
- Apache Spark—An integrated part of CDH and supported with Cloudera Enterprise, Spark is an open standard for flexible in-memory data processing for batch, real time and advanced analytics. Via the one platform Cloudera is committed to adopting Spark as the default data execution engine for analytic workloads.
- Cloudera Manager—A sophisticated application used to deploy, manage, monitor, and diagnose issues with CDH deployments. Cloudera Manager provides the Admin Console, a web-based user interface that makes administration of any enterprise data simple and straightforward. It also includes the Cloudera Manager API, which can be used to obtain cluster health information and metrics, as well as configure Cloudera Manager.
- Cloudera Navigator—An end-to-end data management tool for the CDH platform. Cloudera Navigator enables administrators, data managers, and analysts to explore the large amounts of data in Hadoop. The robust auditing, data management, lineage management, and life cycle management in Cloudera Navigator allow enterprises to adhere to stringent compliance and regulatory requirements.

Cloudera Data Science Workbench

Cloudera Data Science Workbench (CDSW) is a web application that allows data scientists to use a variety of open source languages and libraries to directly and securely access the data in the Hadoop cluster. Direct access to the big data cluster means no more working with small subsets of the data on desktop systems; no sampling is required as the entire data set is available for use directly by the user. Further, users are not restricted to a single environment. Many popular open source libraries and languages are supported, including R, Python and Scala, as well as all the ML/DL frameworks such as TensorFlow, Theano, PyTorch, and so on. In addition, CDSW enables access to available GPU resources for deep learning workloads which means users become productive faster with no need for retraining and no time lost learning a new programming language.

CDSW is addressing the key challenge that every team or user may require a different language, library or framework in order to be productive while the organization requires reproducibility and collaboration. By making the entire set of data in the cluster available to the user, CSDW eliminates the problem that what works on small samples or extracts of the data on a user’s desktop computer may not scale across a large cluster. Cloudera Data Science Workbench gives data scientists the flexibility and simplicity they need to be productive and innovative at scale.

Figure 19 Cloudera Data Science Architecture


Internal CDSW Components
CDH / HDP Components
CDSW Web Application
LiveLog
Local Database (PostgreSQL)
Engines for User Workloads
Sessions
Jobs
Experiments
Models
Spark Driver
Kubernetes (managed by Cloudera Data Science Workbench)
Docker (orchestrated by Kubernetes)
Cloudera Manager / Apache Ambari
Spark
YARN
HDFS
Hive, Impala, HBase, ...
User Authentication via Kerberos
Access Control Enforced by Sentry or Ranger

In addition, CDSW enables seamless access to high-performance processors in the form of GPUs. CSDW makes use of lightweight container architecture to rapidly and securely provide the environment and resources to the users.

Cloudera Data Science Workbench is directly aimed at helping data scientists build and test new analyses and analytics projects as quickly as possible in secure manner even in large scale environments. This flexibility improves the efficiency of the exploration process, a key requirement to meet in order to move rapidly from idea to answer. Most analytics problems, especially those with transformative power, are not standard analyses and require advanced models and iterative methods. Experimentation and innovation are the heart and soul of data science, but security is needed for compliance and governance.

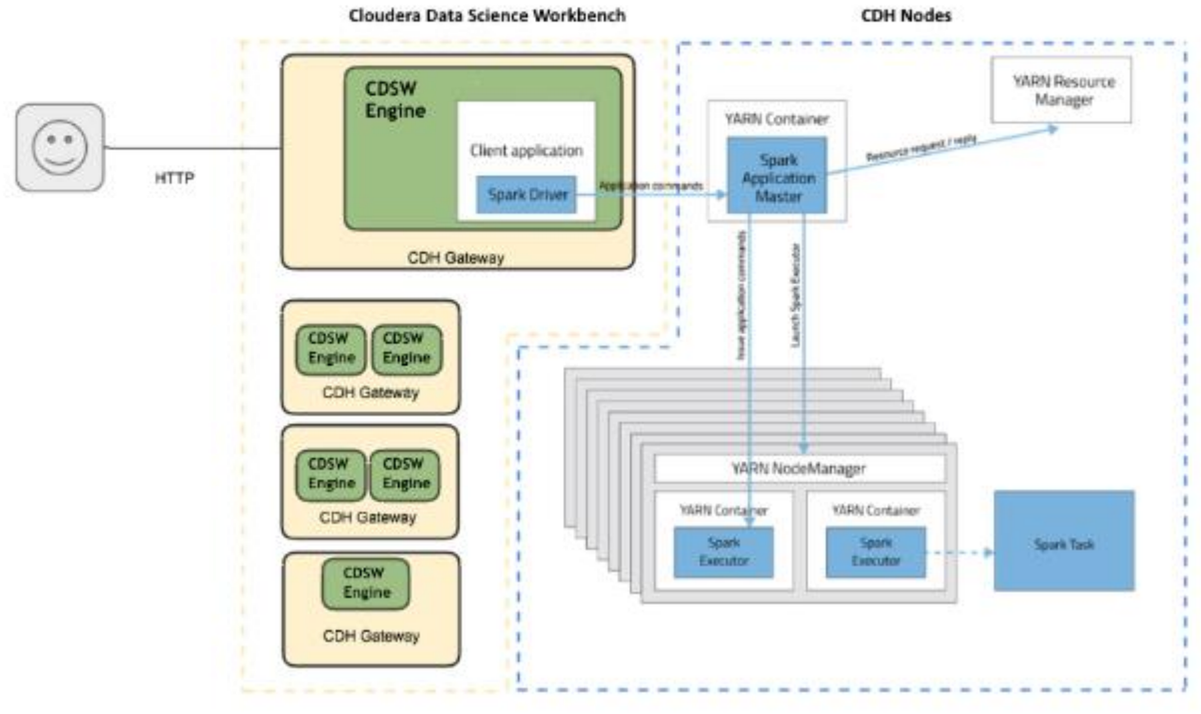
Data has become one of the most strategic assets in the organization. Leveraging the data to drive the business forward is the primary motivation for building an enterprise data hub to support advanced analytics. Typically, when forced to make a choice between the security of the data and the flexibility to access it, security wins locking away the data from the people who most need it. CDSW address this issue by providing full authentication and access controls against data in the cluster, including complete Kerberos integration. It offers data science teams per-project isolation and reproducibility with no effort.

Cloudera Data Science Workbench allows you to automate analytics workloads with a built-in job and pipeline scheduling system that supports real-time monitoring, job history, and email alerts. Jobs are created and can be configured to run on a recurring schedule, as well as providing alerts for successful and failed runs. Multiple jobs can be scheduled together to create an automated pipeline; for example, the first job performs data acquisition, the next data cleansing, then analytics, and so on.

Collaboration and sharing of results are implemented via project sharing (either globally or to specific users, and project forking. To share results, CSDW enables publishing output for viewing via a browser, and even makes the console log itself available for viewing both during and after the run. Cloudera Data Science Workbench is a web application. It has no desktop footprint making it very easy to administer and maintain.

Cloudera Data Science Workbench's interactive mode allows you to launch a Spark application and work iteratively in R, Python, or Scala, rather than the standard workflow of launching an application and waiting for it to complete to view the results. Because of its interactive nature, Cloudera Data Science Workbench works with Spark on YARN's client mode, where the driver persists through the lifetime of the job and runs executors with full access to the CDH cluster resources. This architecture is illustrated in Figure 20.

Figure 20 Cloudera Data Science Workbench with Spark on Yarn

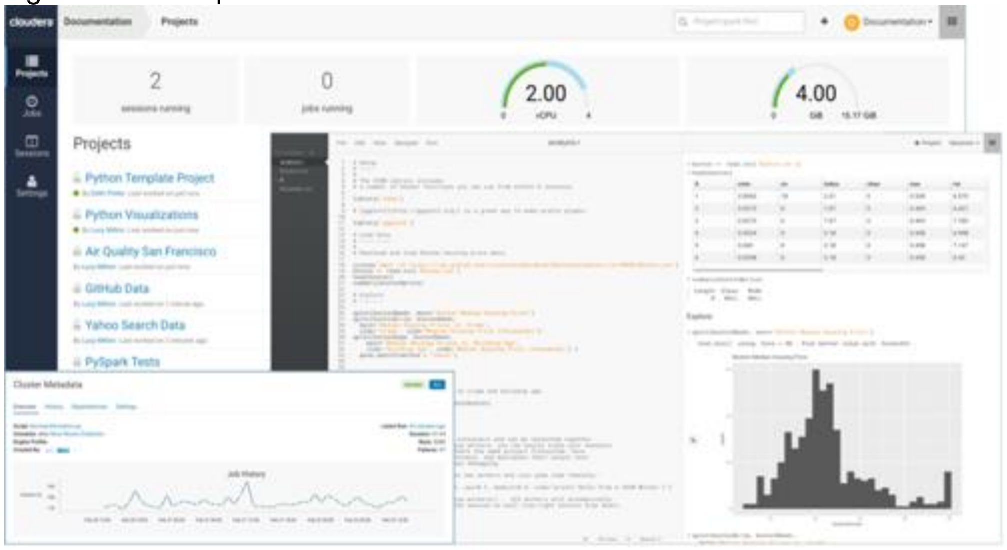


Data Science Workbench provides the following features:

- CPU and GPU as a resource: Data Science Workbench provides basic support for the use of existing general-purpose CPUs for each stage of the workflow and, optionally, accelerates the math-intensive steps with the selective application of special-purpose GPUs all through a Docker container, with Kubernetes scheduling these resources in the back end.
- Self-service portal: The Data Science Workbench web user interface console provides a self-service portal for data scientists to create an environment for their workloads (Figure 21). Currently, R, Python, and Scala are supported.
- Jupyter Notebook: Most data scientists use Jupyter Notebooks for AI/ML analysis and development. Data Science Workbench provides a Jupyter Notebook environment when data scientists create a portal, and these notebooks can be shared or worked in a collaborative manner.

We deployed Cloudera Data Science Workbench version 1.5 in this CVD.

Figure 21 Example of Cloudera Data Science Workbench WebUI



Hortonworks Data Platform

The Hortonworks Data Platform (HDP 3.1.0) delivers essential capabilities in a completely open, integrated and tested platform that is ready for enterprise usage. With Hadoop YARN at its core, HDP provides flexible enterprise data processing across a range of data processing engines, paired with comprehensive enterprise capabilities for governance, security and operations.

All the integration of the entire solution is thoroughly tested and fully documented. By taking the guesswork out of building out a Hadoop deployment, HDP gives a streamlined path to success in solving real business problems.

Hortonworks Data Platform (HDP) 3.0 delivers significant new features, including the ability to launch apps in a matter of minutes and address new use cases for high-performance deep learning and machine learning apps. In addition, this new version of HDP enables enterprises to gain value from their data faster, smarter, in a hybrid environment.

Apache Ambari

Apache Ambari is a completely open source management platform. It performs provisioning, managing, securing, and monitoring Apache Hadoop clusters. Apache Ambari is a part of Hortonworks Data Platform and it allows enterprises to plan and deploy HDP cluster. It also provides ongoing cluster maintenance and management.

Ambari provides an intuitive Web UI as well as an extensive REST API framework which is very useful for automating cluster operations.

Below are the core benefits that Hadoop operators get with Ambari:

- Simplified Installation, Configuration and Management. Easily and efficiently create, manage and monitor clusters at scale. Takes the guesswork out of configuration with Smart Configs and Cluster Recommendations. Enables repeatable, automated cluster creation with Ambari Blueprints.
- Centralized Security Setup. Reduce the complexity to administer and configure cluster security across the entire platform. Helps automate the setup and configuration of advanced cluster security capabilities such as Kerberos and Apache Ranger.

- Full Visibility into Cluster Health. Ensure your cluster is healthy and available with a holistic approach to monitoring. Configures predefined alerts — based on operational best practices — for cluster monitoring. Captures and visualizes critical operational metrics — using Grafana — for analysis and troubleshooting. Integrated with Hortonworks SmartSense for proactive issue prevention and resolution.
- Highly Extensible and Customizable. Fit Hadoop seamlessly into your enterprise environment. Highly extensible with Ambari Stacks for bringing custom services under management, and with Ambari Views for customizing the Ambari Web UI.

HDP for Data Access

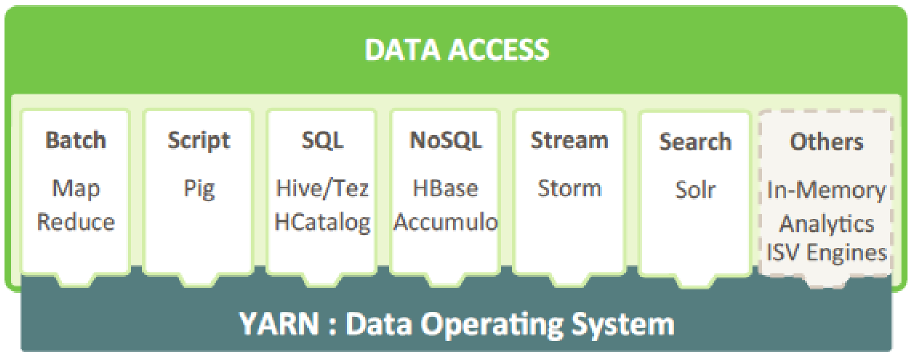
With YARN at its foundation, HDP provides a range of processing engines that allow users to interact with data in multiple and parallel ways, without the need to stand up individual clusters for each data set/application. Some applications require batch while others require interactive SQL or low-latency access with NoSQL. Other applications require search, streaming or in-memory analytics. Apache Solr, Storm and Spark fulfill those needs respectively.

To function as a true data platform, the YARN-based architecture of HDP enables the widest possible range of access methods to coexist within the same cluster avoiding unnecessary and costly data silos.

As shown in Figure 22, HDP Enterprise natively provides for the following data access types:

- Batch – Apache MapReduce has served as the default Hadoop processing engine for years. It is tested and relied upon by many existing applications.
- Interactive SQL Query - Apache Hive is the de facto standard for SQL interactions at petabyte scale within Hadoop. Hive delivers interactive and batch SQL querying across the broadest set of SQL semantics.
- Search - HDP integrates Apache Solr to provide high-speed indexing and sub-second search times across all your HDFS data.
- Scripting - Apache Pig is a scripting language for Hadoop that can run on MapReduce or Apache Tez, allowing you to aggregate, join and sort data.
- Low-latency access via NoSQL - Apache HBase provides extremely fast access to data as a columnar format, NoSQL database. Apache Accumulo also provides high-performance storage and retrieval, but with fine-grained access control to the data.
- Streaming - Apache Storm processes streams of data in real time and can analyze and act on data as it flows into HDFS.

Figure 22 YARN



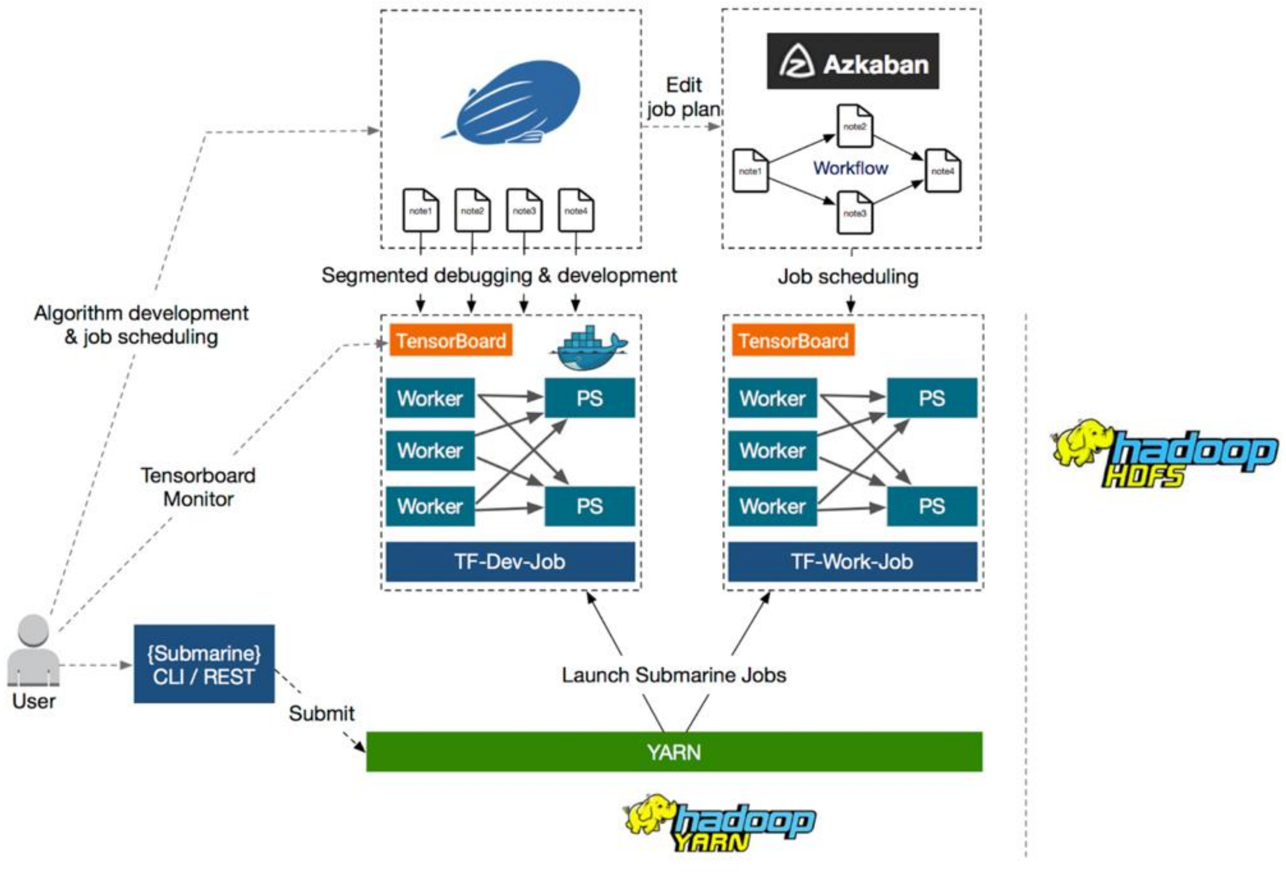
Submarine

Deep learning is useful for enterprises tasks in the field of speech recognition, image classification, AI chatbots, machine translation, just to name a few. In order to train deep learning/machine learning models, frameworks such as TensorFlow / MXNet / PyTorch / Caffe / XGBoost can be leveraged. Sometimes these frameworks are used together to solve different problems.

To make distributed deep learning/machine learning applications easily launched, managed and monitored, Hadoop community initiated the Submarine project along with other improvements such as first-class GPU support, Docker container support, container-DNS support, scheduling improvements, and so on.

These improvements make distributed deep learning/machine learning applications run on Apache Hadoop YARN as simple as running it locally, which can let machine-learning engineers focus on algorithms instead of worrying about underlying infrastructure. By upgrading to latest Hadoop, users can now run deep learning workloads with other ETL/streaming jobs running on the same cluster. This can achieve easy access to data on the same cluster and achieve better resource utilization.

Figure 23 Submarine Workflow



Docker Containerization

Cloudera Data Science Workbench makes use of container technology. Containers are conceptually like virtual machines, but instead of virtualizing the hardware, a container virtualizes the operating system. With a VM there is an entire operating system sitting on top of the hypervisor. Containers dispense with this time-consuming and resource hungry requirement by sharing the host system's kernel. As a result, a container is far smaller, and its lightweight nature means they can be instantiated quickly. In fact, they can be instantiated so quickly that new application architectures are possible.

Docker is an open-source project based on Linux containers. It uses Linux kernel features like namespaces and control groups to create containers. These features are not new, but Docker has taken these concepts and improved them in the following ways:

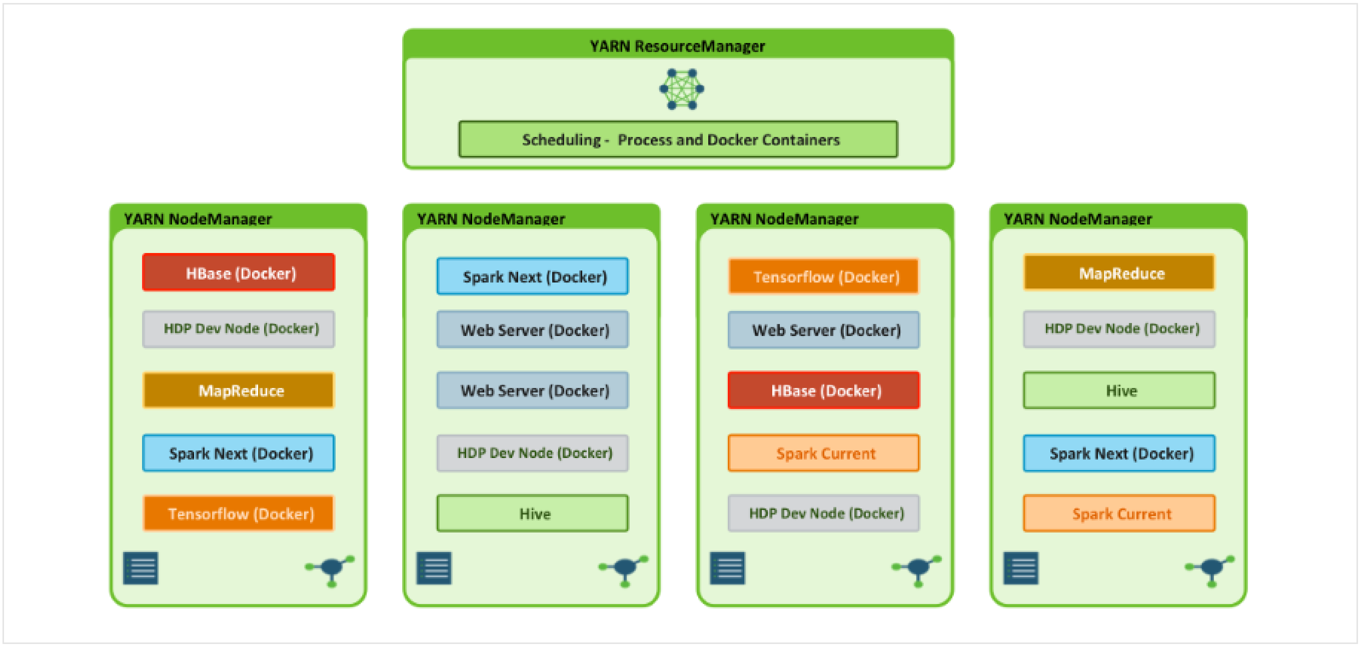
- Ease of use: Docker makes easier for anyone — developers, systems admins, architects and others — to take advantage of containers in order to quickly build and test portable applications. It allows anyone to package an application on their development system, which can then run *unmodified* on any cloud or bare metal server. The basic idea is to create a “build once, run anywhere” system.
- Speed: Docker containers are very fast with a small footprint. Ultimately, containers are just sandboxed environments running on the kernel, so they take up few resources. You can create and run a Docker container in seconds. Compare this to a VM which takes much longer because it must boot up a full virtual operating system every time.
- Modularity: Docker makes it easy to take an application and breaks its functionality into separate individual containers. These containers can then be spun up and run as needed. This is particularly useful for cases where an application needs to hold and lock a resource, like a GPU, and then release it once it’s done using it. Modularity also enables each component, i.e., container to be updated independently.
- Scalability: modularity enables scalability. With different parts of the system running in different containers it becomes possible, and with Docker, it becomes easy to connect these containers together to create an application, which can then be scaled out as needed.

YARN Support for Docker

Containerization provides YARN support for Docker containers, which makes it easier to bundle libraries and dependencies along with their application, allowing third-party applications to run on Apache Hadoop (for example, containerized applications), enabling:

- Faster time to deployment by enabling third-party apps.
- The ability to run multiple versions of an application, enabling users to rapidly create features by developing and testing new versions of services without disrupting old ones.
- Improved resource utilization and increased task throughput for containers, yielding faster time to market for services.
- Orchestration of stateless distributed applications.
- Packaging libraries for Spark application, eliminating the need for operations to deploy those libraries cluster wide.

Figure 24 Containerized Application on Apache Hadoop YARN 3.1



As shown in Figure 24, YARN Services Framework in addition with Docker containerization, it is now possible to run both existing Hadoop frameworks, such as Hive, Spark, and so on, and new containerized workloads on the same underlying infrastructure. Apache Hadoop 3.1 further improved these capabilities to enable advanced use cases such as TensorFlow and HBase.

Kubernetes

Applications built using container technology provide a great deal of flexibility in terms of their architecture, deployment and scaling. Since containers provide VM-like separation of concerns but with far less overhead they allow system developers to package different services of the same application into separate containers. These containers can then be deployed in a very flexible manner including across clusters of physical and virtual machines. This builds the ability to scale directly into the application architecture. This ability in turn requires a tool to aid in deploying, managing and scaling container-based applications.

Kubernetes is an open source project specifically designed for deploying and managing multi-container applications at scale. Kubernetes automates and simplifies the following tasks:

Deploying multi-container applications. With the application split into separate containers for different services, Kubernetes manages the deployment of the containers both at initial startup and in real-time as needed by the application.

Scaling containers. Applications need to spin up and down containers to suit demand, to balance incoming load, and make better use of physical resources. Kubernetes provides the mechanisms for doing these things in a completely automated way.

Updating applications. One advantage of container-based application development is the ability to independently change, improve and fix individual containers. Kubernetes has mechanisms for allowing graceful updates to new versions of container images, including rollbacks if something does not go as planned.

Kubernetes manages application status and any replication and load balancing needs. It also handles hardware resource allocation including GPUs. Kubernetes also has facilities for maximizing the use of hardware resources including memory, storage I/O, and network bandwidth. Applications can have soft and hard limits set on their resource usage. For example, many small applications that use minimal resources can be run together on the same hardware while resource hungry applications can be placed on different hardware and scale out as needed.

NVIDIA Docker

Docker containers are platform-agnostic, but also hardware-agnostic. This presents a problem when using specialized hardware such as NVIDIA GPUs which require kernel modules and user-level libraries to operate. As a result, Docker does not natively support NVIDIA GPUs within containers.

One of the early workarounds to this problem was to fully install the NVIDIA drivers inside the container and map in the character devices corresponding to the NVIDIA GPUs (for example, /dev/nvidia0) on launch. This solution is brittle because the version of the host driver must exactly match the version of the driver installed in the container. This requirement drastically reduced the portability of these early containers, undermining one of Docker's more important features.

To enable portability in Docker images that leverage NVIDIA GPUs, NVIDIA developed nvidia-docker, an open-source project hosted on GitHub that provides the two critical components needed for portable GPU-based containers:

- driver-agnostic CUDA images; and a Docker command line wrapper that mounts the user mode components of the driver and the GPUs (character devices) into the container at launch.
- nvidia-docker is essentially a wrapper around the docker command that transparently provisions a container with the necessary components to execute code on the GPU.

As of the publishing of this CVD, Hortonworks only supports nvidia-docker version 1.

GPU Pooling and Isolation

GPU pooling and isolation allows GPU to be a first-class resource type in Hadoop, making it easier for customers to run machine learning and deep learning workloads.

- Compute-intensive analytics require not only a large compute pool, but also a fast and expensive processing pool with GPUs in tandem
- Customers can share cluster-wide GPU resources without having to dedicate a GPU node to a single tenant or workload
- GPU isolation dedicates a GPU to an application so that no other application has access to that GPU

When it comes to resource scheduling, it is important to recognize GPU as a resource. YARN extends the resource model to more flexible mode which makes it easier to add new countable resource-types. When GPU is added as resource type, YARN can schedule applications on GPU machines. Furthermore, by specifying the number of requested GPU to containers, YARN can find machines with available GPUs to satisfy container requests.

When GPU scheduling is enabled, YARN can schedule non-GPU applications such as LLAP, Tez, and so on, to servers without GPU. Moreover, YARN can allocate GPU applications such as TensorFlow, Caffe, MXNet, and so on, to servers with GPU.

Red Hat Ansible Automation

Red Hat Ansible Automation is a powerful IT automation tool. It is capable of provisioning numerous types of resources and deploying applications. It can configure and manage devices and operating system components. Due to its simplicity, extensibility, and portability, this solution extensively utilizes Ansible for performing repetitive deployment steps across the nodes.

For more information about Ansible, go to: https://www.redhat.com/en/technologies/management/ansible

Solution Design

Requirements

This CVD describes architecture and deployment procedures for Cloudera Enterprise Data Hub (CDH 6.2.0) and Cloudera Data Science Workbench (CDSW 1.5.0) on a 30-node cluster based on Cisco UCS Integrated Infrastructure for Big Data and Analytics. The solution goes into detail configuring CDH 6.2.0 on the infrastructure, as well as the complete installation and configuration of CDSW 1.5.0 and all its dependencies.

The cluster configuration consists of the following:

- 2 Cisco UCS 6332UP Fabric Interconnects
- 22 Cisco UCS C240 M5 Rack-Mount servers
- 4 Cisco UCS S3260 Storage Server
- 8 NVIDIA Tesla T4 GPU
- 2 Cisco R42610 standard racks
- 4 Vertical Power distribution units (PDUs) (Country Specific)

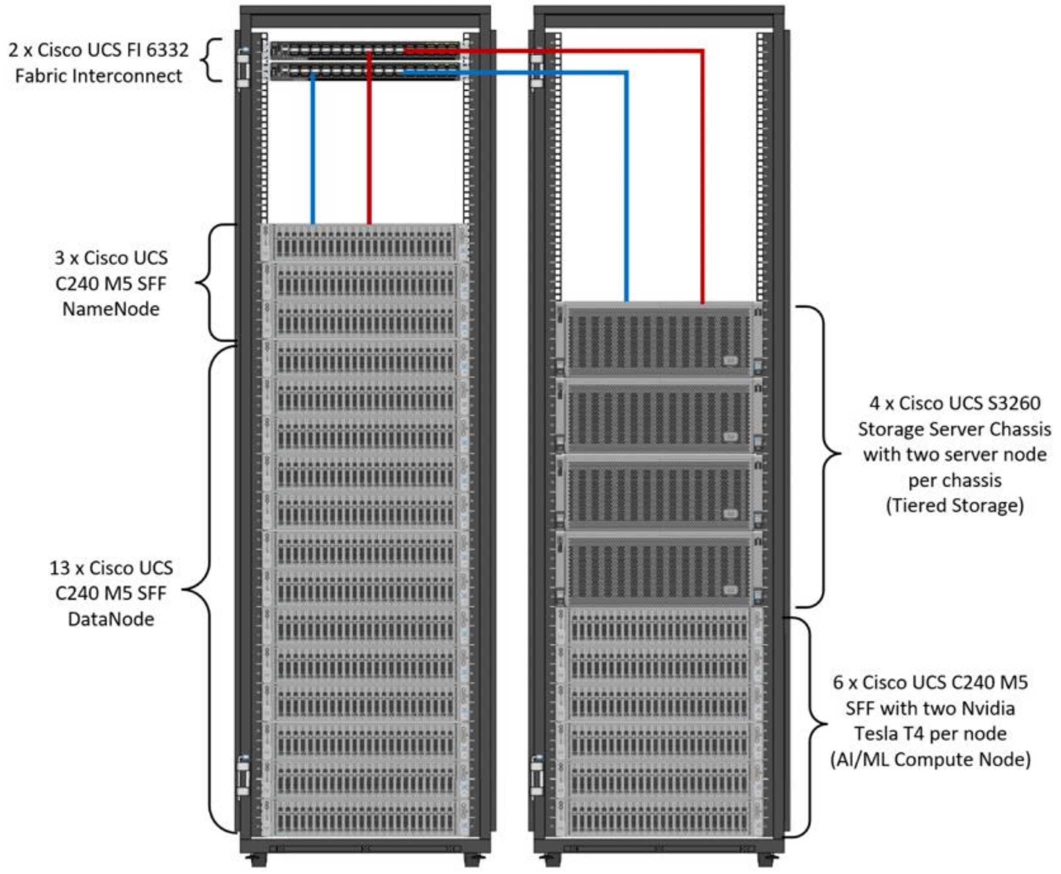
Physical Topology

Each rack consists of two vertical PDUs. The first rack consists of two Cisco UCS 6332UP Fabric Interconnects, 16 Cisco UCS C240 M5 Rack Servers connected to each of the vertical PDUs for redundancy; thereby, ensuring availability during power source failure. The second rack consists of 6 Cisco UCS C240 M5 Servers and 4 Cisco UCS S3260 Modular Storage Server connected to each of the vertical PDUs for redundancy; thereby, ensuring availability during power source failure, like the first rack.

Please contact your Cisco representative for country specific information.

As illustrated in Figure 25, a 30-node starter cluster. Rack #1 has sixteen Cisco UCS C240 M5 servers. Each link in the figure represents a 40 Gigabit Ethernet link from each of the sixteen servers directly connected to a Fabric Interconnect. Rack #2 has six Cisco UCS C240 M5 and four Cisco UCS S3260 servers. Every server is connected to both Fabric Interconnects.

Figure 25 Cisco Data Intelligence Platform – 30 Node Configuration with Cloudera CDH 6.2 and CDSW 1.5



Port Configuration on Fabric Interconnect

Table 5 lists the port configuration on Cisco UCS FI 6332 Fabric Interconnect.

Table 5 Port Configuration on Fabric Interconnect

Port Type	Port Number
Server	1-26
Network	29-32

Server Configuration and Cabling for Cisco UCS C240 M5

The Cisco UCS C240 M5 rack server is equipped with 2 x Intel Xeon Scalable Family Processor 6230 (2 x 20 cores, 2.1 GHz), 384 GB of memory, Cisco UCS Virtual Interface Card 1337, Cisco 12-Gbps SAS Modular Raid Controller with 4-GB FBWC, 26 x 1.8 TB 10K rpm SFF SAS HDDs or 12 x 1.6 TB Enterprise Value SATA SSDs, M.2 with 2 x 240-GB SSDs for Boot.

Figure 26 illustrates the port connectivity between the Cisco UCS FI 6332 and Cisco UCS C240 M5 Rack Server. Twenty-two Cisco UCS C240 M5 servers are installed in this configuration.

Figure 26 Fabric Topology for Cisco UCS C240 M5 Rack Server

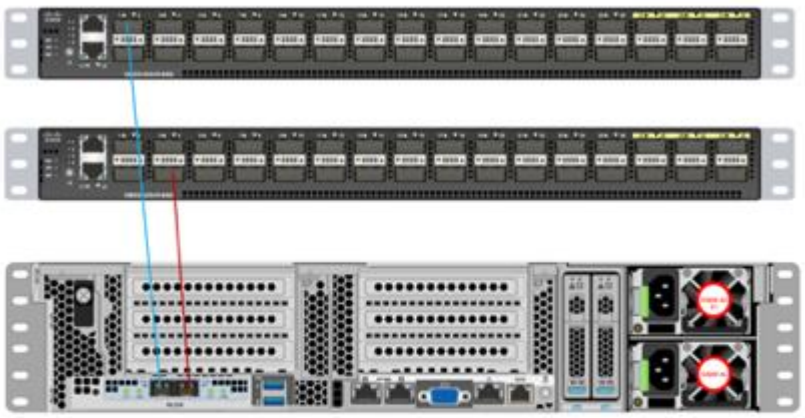


Figure 27 illustrates the port connectivity between the Cisco UCS FI 6332 and Cisco UCS S3260 Storage Server. Four Cisco UCS S3260 M5 servers are installed in this configuration.

Figure 27 Fabric Topology for Cisco UCS S3260 Storage Server

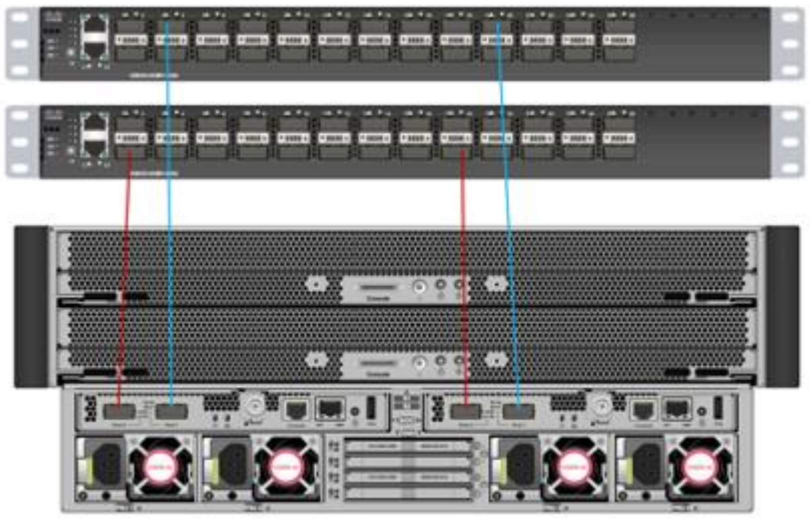


Figure 27 shows the connectivity between Cisco UCS S3260 Storage Server chassis and Cisco UCS 6300 Fabric Interconnects. Each chassis has two Cisco UCS C3260 M5 server nodes. Each link in Figure 27 represents a 40 Gigabit Ethernet link from the Cisco UCS S3260 Storage Server chassis connecting to a Fabric Interconnect. Every chassis is connected to both Fabric Interconnects represented with dual links. Since each chassis will have two server nodes, the top server node works with the left SIOC and the bottom server node works with right SIOC. Similarly, for the boot drives, the top two SSD slots are assigned for server node 1 and the bottom two SSD slots are assigned for server node 2.

For information on physical connectivity and single-wire management, see:

https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c-series_integration/ucsm4-0/b_C-Series-Integration_UCSM4-0/b_C-Series-Integration_UCSM4-0_chapter_01.html

Software Distributions and Versions

The software distributions required versions are listed in Table 6

Table 6 Software distribution and Version

Layer	Component	Version or Release
Compute	Cisco UCS C240 M5	C240M5.4.0.4h
	Cisco UCS S3260	S3X60M5.4.0.4g
Network	Cisco UCS 6332	UCS 4.0(4c) A
	Cisco UCS VIC1387 Firmware	4.3(3b)
	S3260 SIOC with Cisco UCS VIC1380 Included Firmware	4.3(3b)
Storage	SAS Expander	65.09.16.00
	Cisco 12G Modular Raid controller	50.8.0-2649
	Storage Controller SAS	29.00.1-0356
	LSI MegaRAID SAS Driver	07.708.03.00
Software	Red Hat Enterprise Linux Server	7.6
	Cisco UCS Manager	4.0(4c)
	Cloudera CDH	6.2.0
	Cloudera Data Science Workbench	1.5.0
NVidia GPU	CUDA Version	10.1
	Driver Version	418.67

Latest drivers can be downloaded from the link below: https://software.cisco.com/download/home/283862063/type/283853158/release/4.0(4)

Support for Intel second generation scalable family processor added in UCSM version 4.0.4a.

Deployment Hardware and Software

Cisco Unified Computing System Configuration

This section details the Cisco Unified Computing System (Cisco UCS) configuration that was done as part of the infrastructure build out. The racking, power, and installation of the Cisco UCS Rack Server is described in the physical topology section earlier in this document. Please refer to the Cisco UCS Manager Getting Started Guide. For more information about each step, see the Cisco UCS Manager - Configuration Guides.

Configure Cisco UCS Fabric Interconnect

This document assumes you are using Cisco UCS Manager Software version 4.0(4c). To upgrade the Cisco UCS Manager software and the Cisco UCS 6332 Fabric Interconnect software to a higher version of the firmware, see the Cisco UCS Manager Install and Upgrade Guides.

Alternatively, if you intend to clear the existing Cisco UCS Manager configuration, follow these steps:

1. Connect a console cable to the console port on what will become the primary fabric interconnect.
2. If the fabric interconnects were previously deployed and you want to erase it to redeploy, follow these steps:
   a. Login with the existing username and password.

```
#connect local-mgmt

#erase config

#yes (to confirm)
```

3. After the fabric interconnect restarts, the out-of-box first time installation prompt appears, type "console" and press Enter.
4. Follow the Initial Configuration steps as outlined in Cisco UCS Manager Getting Started Guide. When configured, log into UCSM IP Address via the web interface to perform the base Cisco UCS configuration.

Configure Fabric Interconnects for a Cluster Setup

To configure the Cisco UCS Fabric Interconnects, follow these steps:

1. Verify the following physical connections on the fabric interconnect:
   - The management Ethernet port (mgmt0) is connected to an external hub, switch, or router.
   - The L1 ports on both fabric interconnects are directly connected to each other.
   - The L2 ports on both fabric interconnects are directly connected to each other

Configure Fabric Interconnect A

To configure Fabric Interconnect A, follow these steps:

1. Connect to the console port on the first Cisco UCS 6332 Fabric Interconnect.

```
At the prompt to enter the configuration method, enter console to continue.
```

```
If asked to either perform a new setup or restore from backup, enter setup to
continue.
Enter y to continue to set up a new Fabric Interconnect.
Enter y to enforce strong passwords.
```

2. Enter the password for the admin user.

3. Enter the same password again to confirm the password for the admin user.

```
When asked if this fabric interconnect is part of a cluster, answer y to continue.
Enter A for the switch fabric.
```

4. Enter the cluster name for the system name.

5. Enter the Mgmt0 IPv4 address.

6. Enter the Mgmt0 IPv4 netmask.

7. Enter the IPv4 address of the default gateway.

8. Enter the cluster IPv4 address.

```
To configure DNS, answer y.
```

9. Enter the DNS IPv4 address.

```
Answer y to set up the default domain name.
```

10. Enter the default domain name.

```
Review the settings that were printed to the console, and if they are correct,
answer yes to save the configuration.
```

11. Wait for the login prompt to make sure the configuration has been saved.

Configure Fabric Interconnect B

To configure Fabric Interconnect B, follow these steps:

1. Connect to the console port on the second Cisco UCS 6332 Fabric Interconnect.

```
When prompted to enter the configuration method, enter console to continue.
The installer detects the presence of the partner Fabric Interconnect and adds this
fabric interconnect to the cluster. Enter y to continue the installation.
```

2. Enter the admin password that was configured for the first Fabric Interconnect.

3. Enter the Mgmt0 IPv4 address.

4. Answer yes to save the configuration.

5. Wait for the login prompt to confirm that the configuration has been saved.

For more information about configuring Cisco UCS 6332 Series Fabric Interconnect, go to: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/ucs-manager/GUI-User-Guides/Getting-Started/4-0/b_UCSM_Getting_Started_Guide_4_0.html

Log into Cisco UCS Manager

To log into Cisco UCS Manager, follow these steps:

1. Open a Web browser and navigate to the Cisco UCS 6332 Fabric Interconnect cluster address.
2. Click the Launch link to download the Cisco UCS Manager software.
3. If prompted to accept security certificates, accept as necessary.
4. When prompted, enter admin for the username and enter the administrative password.
5. Click Login to log in to the Cisco UCS Manager.

Upgrade Cisco UCS Manager Software to Version 4.0(4c)

This document assumes you're using Cisco UCS 4.0(4c). Refer to the Cisco UCS 4.0 Release (upgrade Cisco UCS Manager software and Cisco UCS 6332 Fabric Interconnect software to version 4.0(2a)). Also, make sure the Cisco UCS C-Series version 4.0(4c) software bundles are installed on the Fabric Interconnects.

Upgrading Cisco UCS firmware is beyond the scope of this document. However for complete Cisco UCS Install and Upgrade Guides, go to: https://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-manager/products-installation-guides-list.html

Configure Cisco UCS Manager

The following are the high-level steps involved for a Cisco UCS Manager configuration:

1. Configure Fabric Interconnects for a Cluster Setup.
2. Set Fabric Interconnects to Fibre Channel End Host Mode.
3. Synchronize Cisco UCS to NTP.
4. Configure Fabric Interconnects for Rack or Chassis and Blade Server Discovery.
5. Configure Global Policies.
6. Configure Server Ports.
7. Configure LAN on Cisco UCS Manager.
8. Configure Ethernet LAN Uplink Ports.
9. Set QoS system class and Jumbo Frames in both the Cisco Fabric Interconnect.
10. Create Uplink Port Channels to Cisco Nexus Switches.
11. Configure FC SAN Uplink Ports

12. Configure VLAN

13. Configure IP, UUID, Server, MAC Pool and policy:

    a. IP Pool Creation
    b. UUID Suffix Pool Creation
    c. Server Pool Creation
    d. Configure Server BIOS Policy.
    e. Create Adapter Policy.
    f. Configure Default Maintenance Policy.
    g. Configure vNIC Template
    h. Create Server Boot Policy

Details for each step are discussed in the following sections.

Synchronize Cisco UCSM to NTP

To synchronize the Cisco UCS environment to the NTP server, follow these steps:

1. In Cisco UCS Manager, in the navigation pane, click the Admin tab.

2. Select All > Time zone Management.

3. In the Properties pane, select the appropriate time zone in the Time zone menu.

4. Click Save Changes and then click OK.

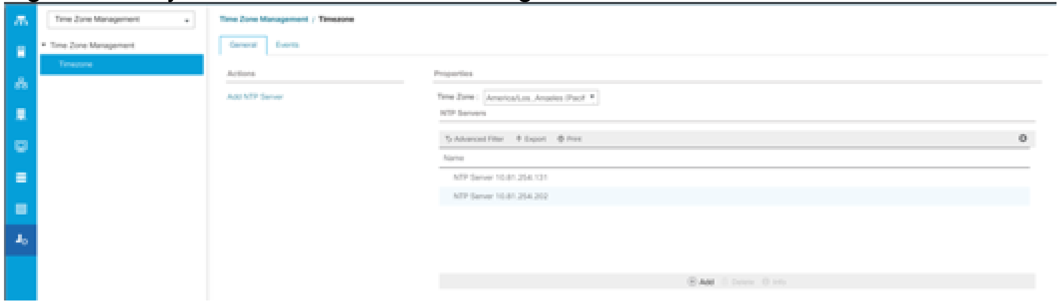
5. Click Add NTP Server.

6. Enter the NTP server IP address and click OK.

7. Click OK to finish.

8. Click Save Changes.

Figure 28 Synchronize Cisco UCS Manager to NTP



Time Zone : America/Los_Angeles (Pacif

NTP Servers

Name
NTP Server 10.81.254.131
NTP Server 10.81.254.202

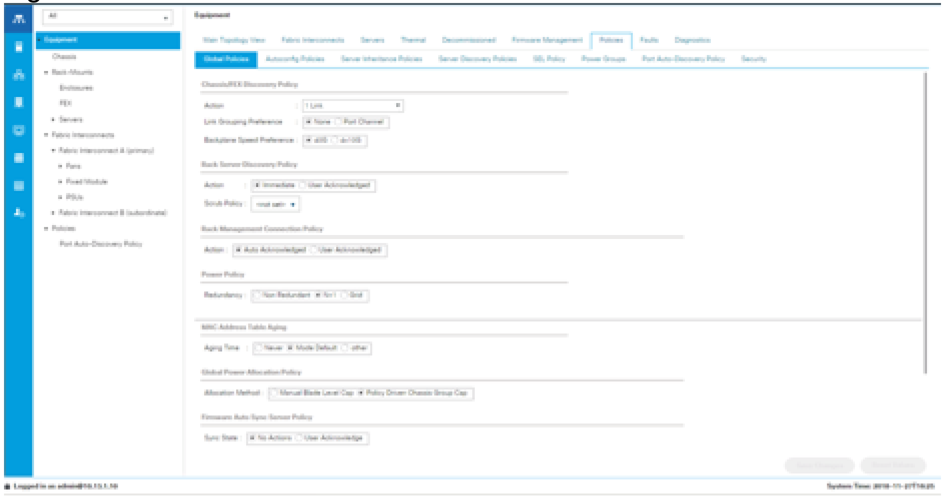
Configure Global Policies

The rack server and chassis discovery policy determine how the system reacts when you add a new rack server or chassis. We recommend using the platform max value as shown. Using platform max helps ensure that Cisco UCS Manager uses the maximum number of IOM uplinks available.

To configure the global policies, follow this step:

1. In Cisco UCS Manager; Go to Equipment > Policies (right pane) > Global Policies as shown in Figure 29.

Figure 29 Global Policies in UCSM



Chassis/FEX Discovery Policy
Action : 1 Link
Link Grouping Preference : [✓ None] Port Channel
Backplane Speed Preference : [✓ 40G] 4x10G

Rack Server Discovery Policy
Action : [✓ Immediate] User Acknowledged
Scrub Policy : <not set>

Rack Management Connection Policy
Action : [✓ Auto Acknowledged] User Acknowledged

Power Policy
Redundancy : Non Redundant [✓ N+1] Grid

MAC Address Table Aging
Aging Time : Never [✓ Mode Default] other

Global Power Allocation Policy
Allocation Method : Manual Blade Level Cap [✓ Policy Driven Chassis Group Cap]

Firmware Auto Sync Server Policy
Sync State : [✓ No Actions] User Acknowledge

Configure Server Ports

Configure Server Ports to initiate Chassis and Blade discovery. To configure server ports, follow these steps:

1. Go to Equipment > Fabric Interconnects > Fabric Interconnect A > Fixed Module > Ethernet Ports.

2. Select the ports (for this solution ports are 1-28) which are connected to the Cisco UCS VIC 1387 on Cisco UCS C240 M5 rack server.

3. Right-click and select Configure as Server Port.

Figure 30 Configure Server Port on Cisco UCS Manager Fabric Interconnect for Server/Chassis Discovery

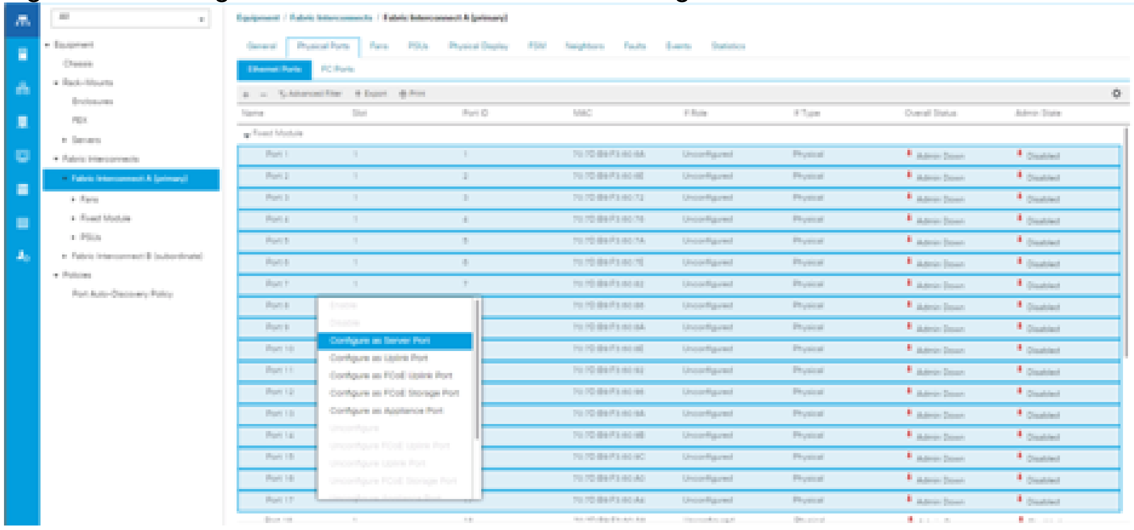
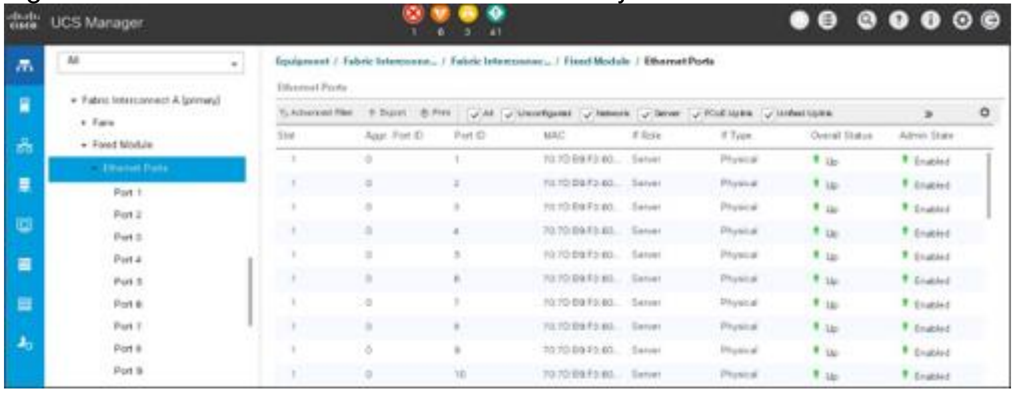

Configure as Server Port
Configure as Uplink Port
Configure as FCoE Uplink Port
Configure as FCoE Storage Port
Configure as Appliance Port

Figure 31 Ports Status after the Server Discovery



Slot	Aggr. Port ID	Port ID	MAC	If Role	If Type	Overall Status	Admin State
1	0	1	70:70:D9:F3:60...	Server	Physical	Up	Enabled
1	0	2	70:70:D9:F3:60...	Server	Physical	Up	Enabled
1	0	3	70:70:D9:F3:60...	Server	Physical	Up	Enabled
1	0	4	70:70:D9:F3:60...	Server	Physical	Up	Enabled
1	0	5	70:70:D9:F3:60...	Server	Physical	Up	Enabled
1	0	6	70:70:D9:F3:60...	Server	Physical	Up	Enabled
1	0	7	70:70:D9:F3:60...	Server	Physical	Up	Enabled
1	0	8	70:70:D9:F3:60...	Server	Physical	Up	Enabled
1	0	9	70:70:D9:F3:60...	Server	Physical	Up	Enabled
1	0	10	70:70:D9:F3:60...	Server	Physical	Up	Enabled

Configure Uplink Ports

Configure Network Ports to connect to the datacenter network switch.

In our solution study we connected to Nexus 9000 series switch.

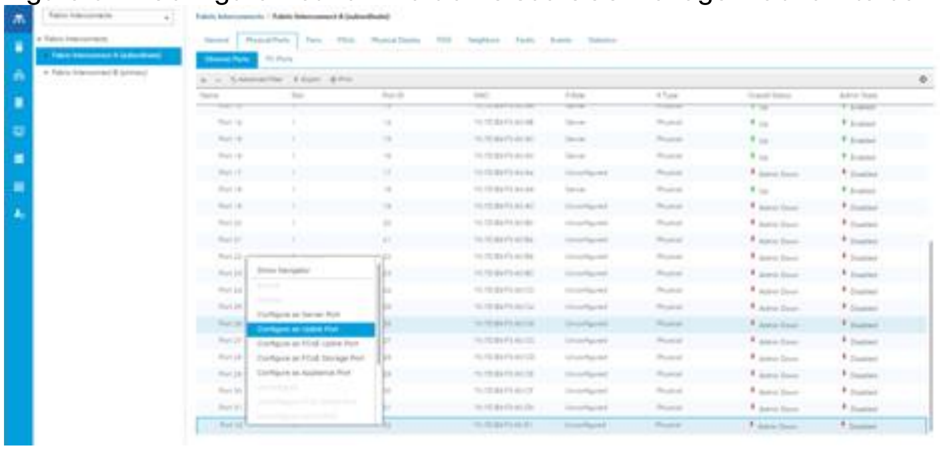
To configure Network ports, follow these steps:

1. Go to Equipment > Fabric Interconnects > Fabric Interconnect A > Fixed Module > Ethernet Ports.

2. Select the ports (for this solution ports are 29-32) which are connected to the Cisco Nexus 9000 series switch for northbound network connectivity.

3. Right-click and select Configure as Network Port.

Figure 32 Configure Network Port on Cisco UCS Manager Fabric Interconnect



After the Server port and network port configuration on Cisco UCS FI 6332, Ports 1-30 are utilized for server management and data traffic and 31-32 will be a Network Port.

Create New Organization

To configure the necessary Organization for the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.

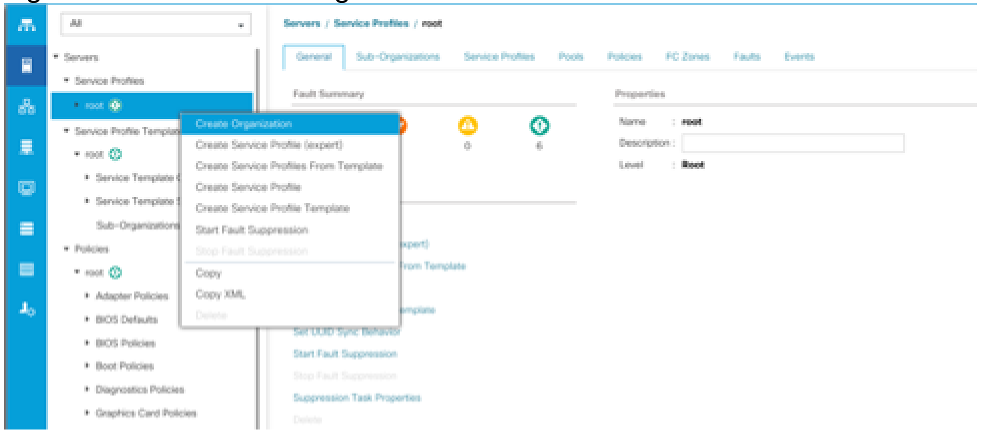
2. Select root > Sub-Organization.

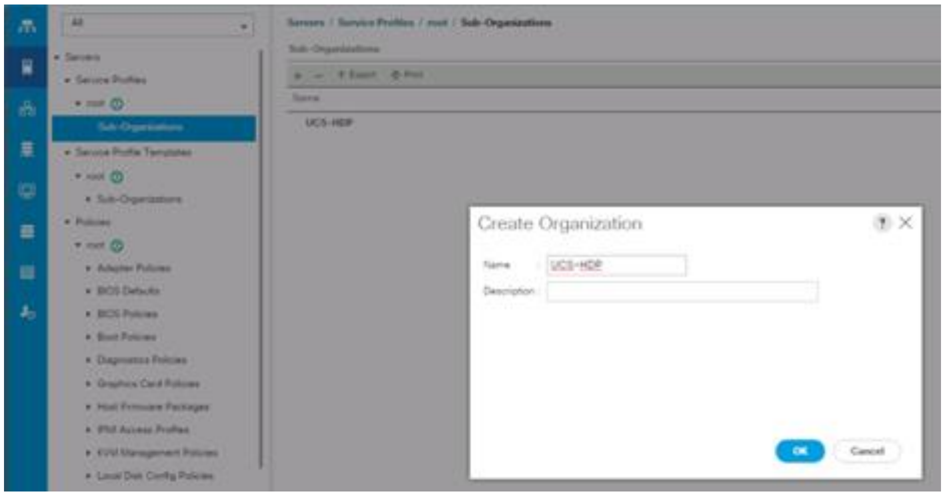
3. Right-click Sub-Organization.

4. Enter the name of the Organization.

5. Click Ok.

Figure 33 Create New Organization





Cisco UCS Manager pools and policies required for this solution were created under new "UCS-HDP" Organization created.

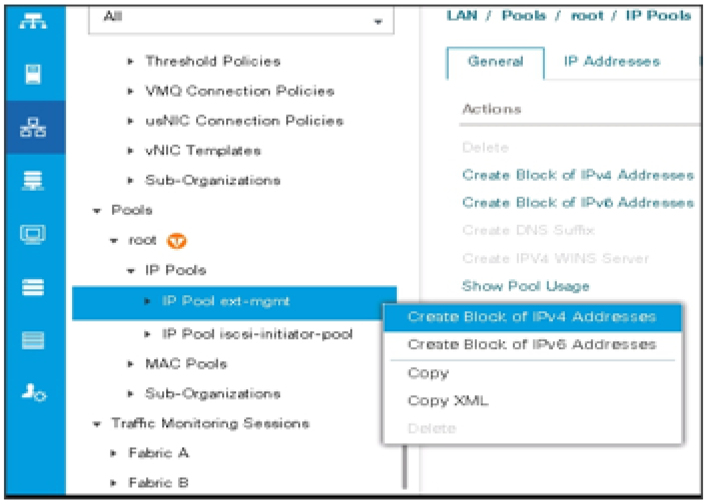
Configure IP, UUID, Server and MAC Pools

IP Pool Creation

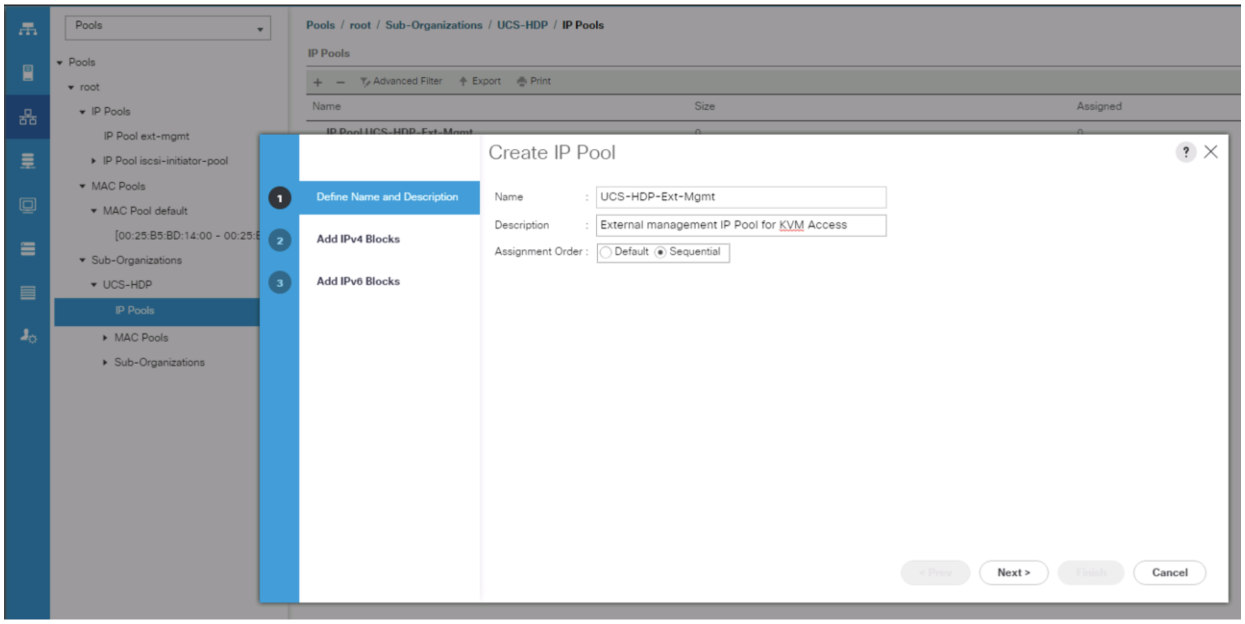
An IP address pool on the out of band management network must be created to facilitate KVM access to each compute node in the Cisco UCS domain. To create a block of IP addresses for server KVM access in the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, in the navigation pane, click the LAN tab.

2. Select Pools > root > Sub-Organizations > UCS-HDP > IP Pools > click Create IP Pool.



3. Enter name for the IP Pool, select option Sequential to assign IP in sequential order then click Next.


Pools
Pools / root / Sub-Organizations / UCS-HDP / IP Pools
Create IP Pool
Define Name and Description
Add IPv4 Blocks
Add IPv6 Blocks
< Prev
Next >
Finish
Cancel

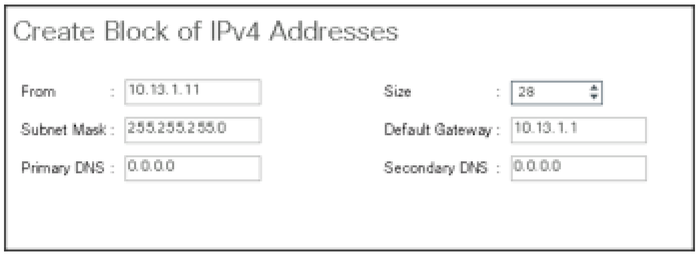
Name : UCS-HDP-Ext-Mgmt

Description : External management IP Pool for KVM Access

Assignment Order : Default [✓ Sequential]

4. Click Add IPv4 Block.

5. Enter the starting IP address of the block and the number of IP addresses required, and the subnet and gate-way information as shown below.



Create Block of IPv4 Addresses

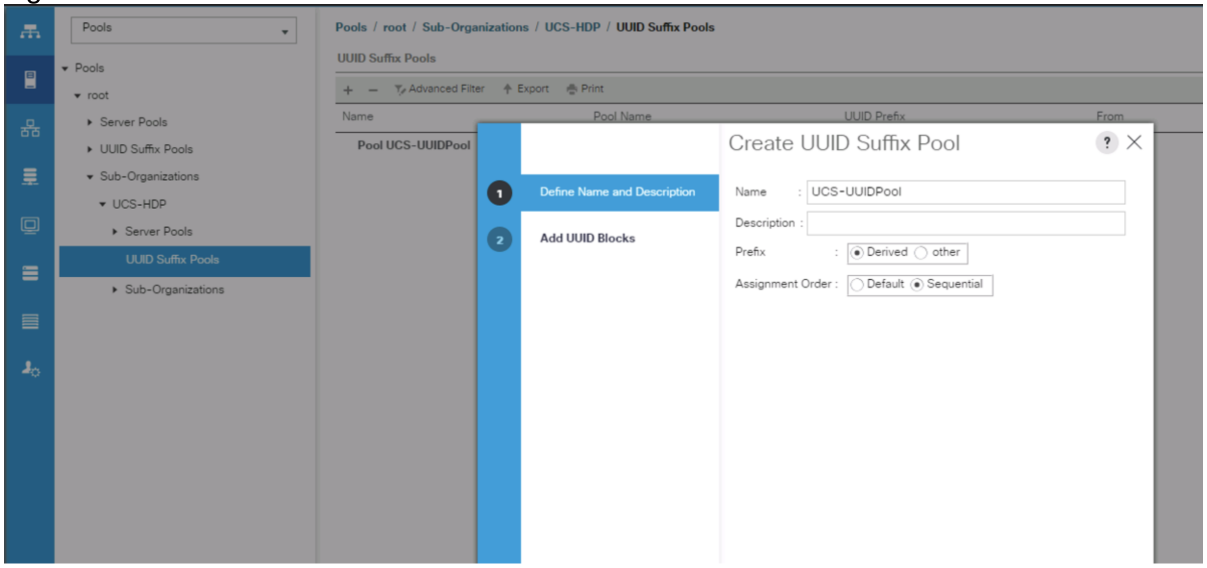
From :	10.13.1.11	Size :	28
Subnet Mask :	255.255.255.0	Default Gateway :	10.13.1.1
Primary DNS :	0.0.0.0	Secondary DNS :	0.0.0.0

UUID Suffix Pool Creation

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 tab in the navigation pane.
2. Select Pools > root > Sub-Organization > UCS-HDP.
3. Right-click UUID Suffix Pools and then select Create UUID Suffix Pool.
4. Enter the name of the UUID name.
5. Optional: Enter a description for the UUID pool.
6. Keep the prefix at the derived option and select Sequential in as Assignment Order then click Next.

Figure 34 UUID Suffix Pool Creation



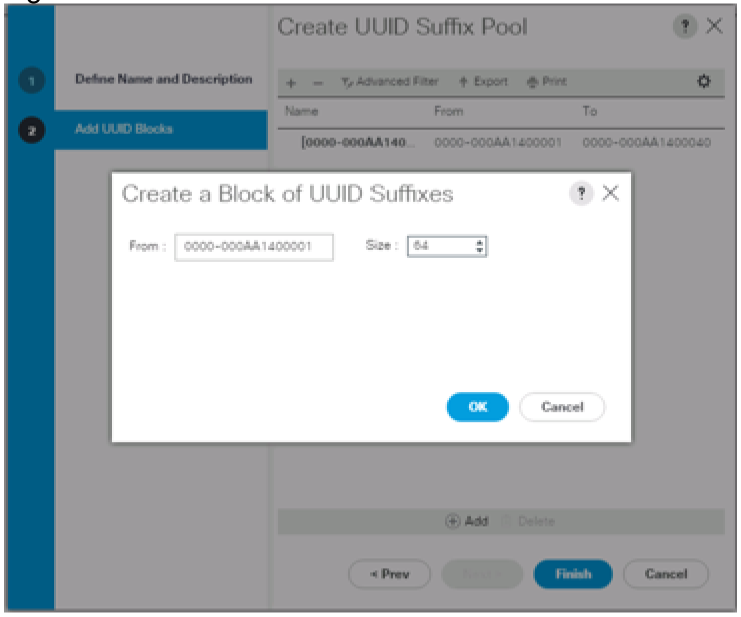
Name : UCS-UUIDPool

Description :

Prefix : [✓ Derived] other

Assignment Order : Default [✓ Sequential]

Figure 35 Create a Block of UUID Suffixes



Create a Block of UUID Suffixes

Name	From	To
[0000-000AA140...	0000-000AA1400001	0000-000AA1400040

From : 0000-000AA1400001
Size : 64

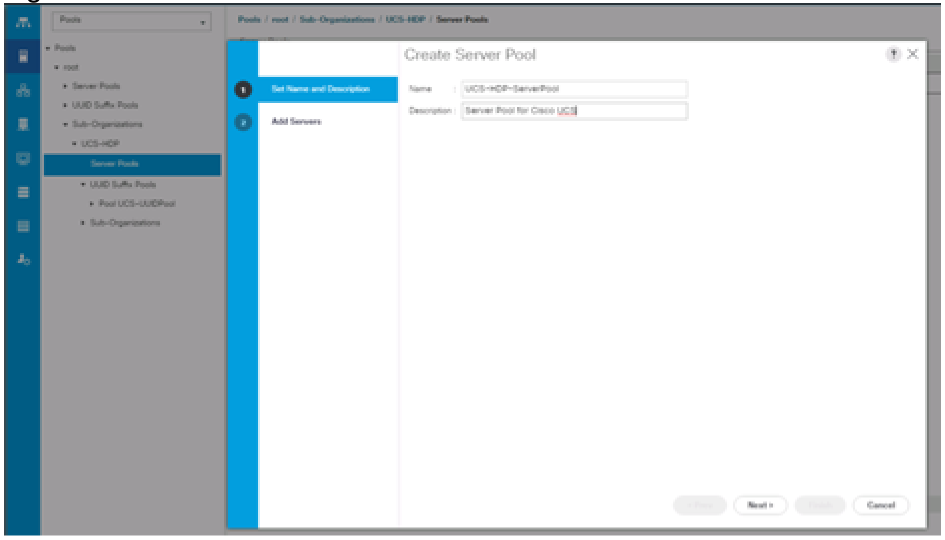
Server Pool Creation

To configure the necessary server pool for the Cisco UCS environment, follow these steps:

Consider creating unique server pools to achieve the granularity that is required in your environment.

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Pools > root > Sub-Organization > UCS-HDP> right-click Server Pools > Select Create Server Pool.
3. Enter name of the server pool.
4. Optional: Enter a description for the server pool then click Next.

Figure 36 Create Server Pool

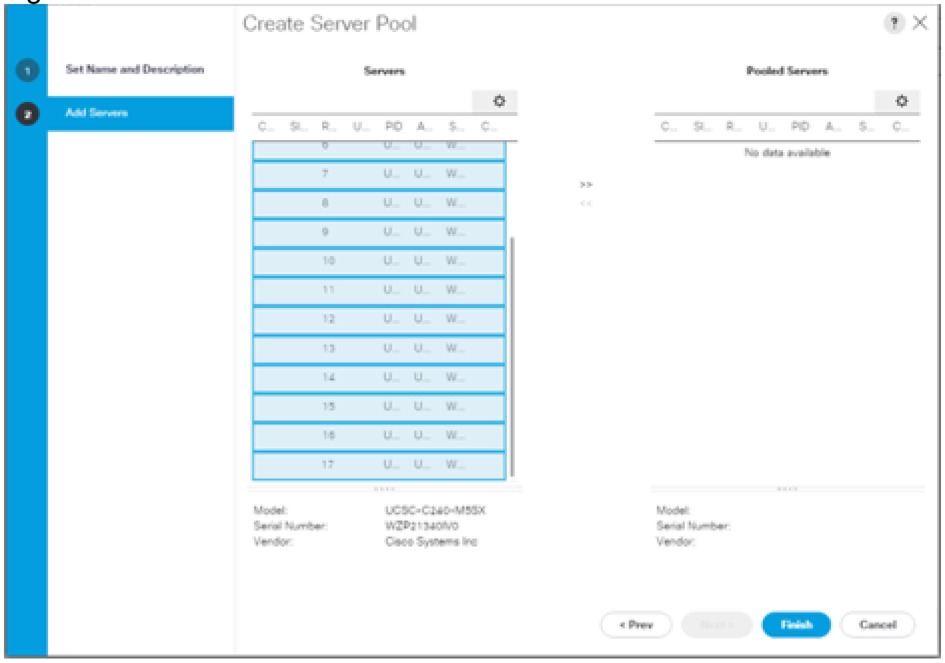


Name : UCS-HDP-ServerPool
Description : Server Pool for Cisco UCS

5. Select servers to be used for the deployment and click > to add them to the server pool. In our case we added thirty servers in this server pool.

6. Click Finish and then click OK.

Figure 37 Add Server in the Server Pool



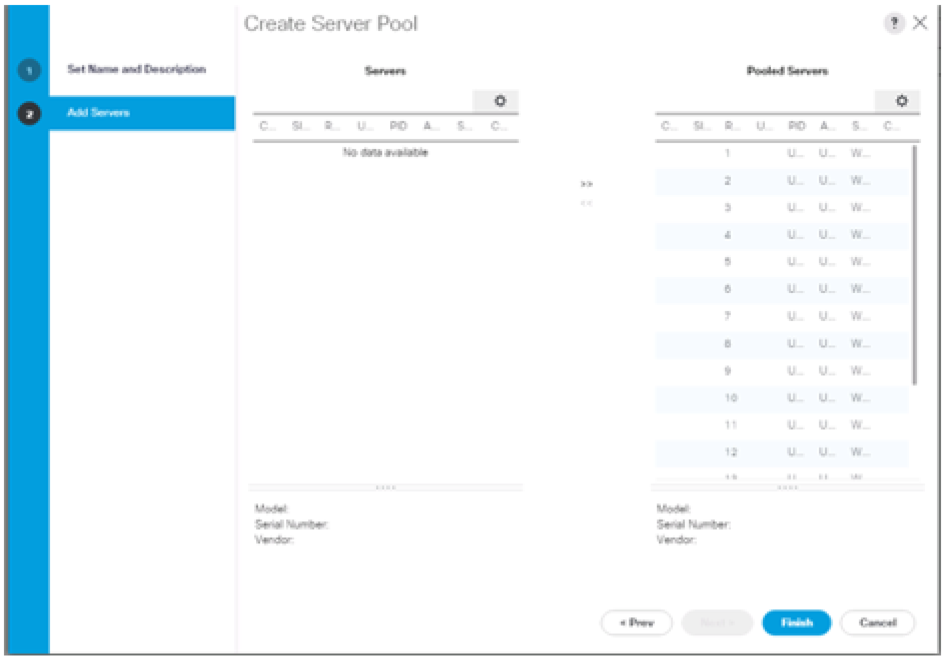
C...	Sl...	R...	U...	PID	A...	S...	C...
		6		U...	U...	W...	
		7		U...	U...	W...	
		8		U...	U...	W...	
		9		U...	U...	W...	
		10		U...	U...	W...	
		11		U...	U...	W...	
		12		U...	U...	W...	
		13		U...	U...	W...	
		14		U...	U...	W...	
		15		U...	U...	W...	
		16		U...	U...	W...	
		17		U...	U...	W...	

Model: UCSC-C240-M5SX
Serial Number: WZP21340N0
Vendor: Cisco Systems Inc

No data available

Model:
Serial Number:
Vendor:

7. Once the added Servers are in the Pooled servers, click Finish.



No data available

Model:
Serial Number:
Vendor:

C...	Sl...	R...	U...	PID	A...	S...	C...
		1		U...	U...	W...	
		2		U...	U...	W...	
		3		U...	U...	W...	
		4		U...	U...	W...	
		5		U...	U...	W...	
		6		U...	U...	W...	
		7		U...	U...	W...	
		8		U...	U...	W...	
		9		U...	U...	W...	
		10		U...	U...	W...	
		11		U...	U...	W...	
		12		U...	U...	W...	

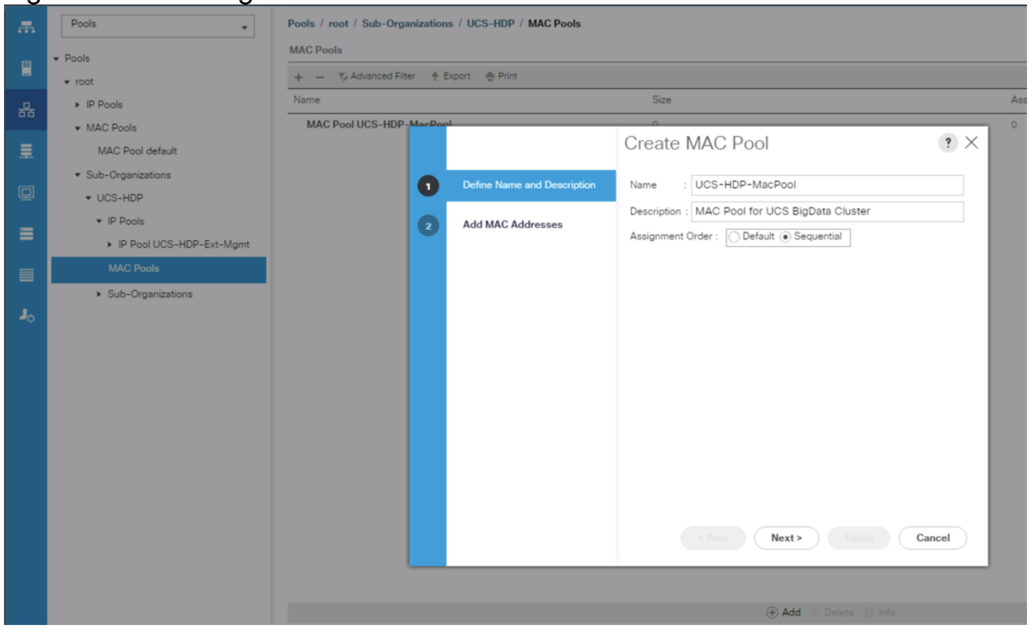
Model:
Serial Number:
Vendor:

MAC Pool Creation

To configure the necessary MAC address pools for the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Pools > root > Sub-Organization > UCS-HDP > right-click MAC Pools.
3. Select Create MAC Pool to create the MAC address pool.
4. Enter name for MAC pool. Select Assignment Order as “Sequential”.
5. Enter the seed MAC address and provide the number of MAC addresses to be provisioned.
6. Click OK and then click Finish.
7. In the confirmation message, click OK.

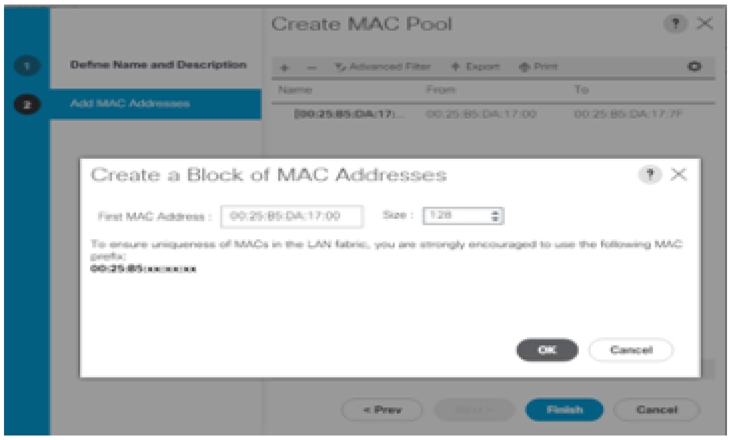
Figure 38 Creating a Block of MAC Addresses



Name	Size
MAC Pool UCS-HDP-MacPool	0

Create MAC Pool

Name : UCS-HDP-MacPool
Description : MAC Pool for UCS BigData Cluster
Assignment Order : Default [✓ Sequential]



Name	From	To
[00:25:B5:DA:17:...	00:25:B5:DA:17:00	00:25:B5:DA:17:7F

Create a Block of MAC Addresses

First MAC Address : 00:25:B5:DA:17:00 Size : 128

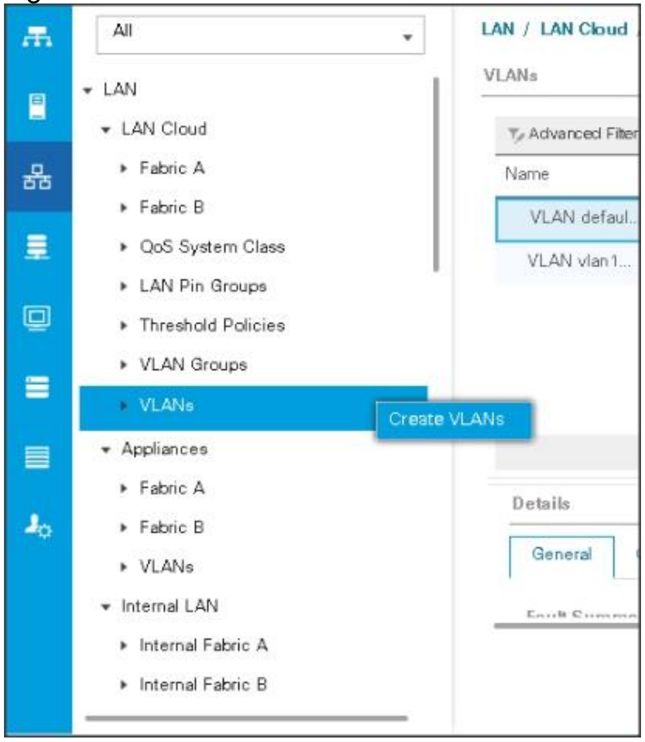
To ensure uniqueness of MACs in the LAN fabric, you are strongly encouraged to use the following MAC prefix:
00:25:B5:xx:xx:xx

Configure VLAN

To configure the necessary virtual local area networks (VLANs) for the Cisco UCS environment, follow these steps:

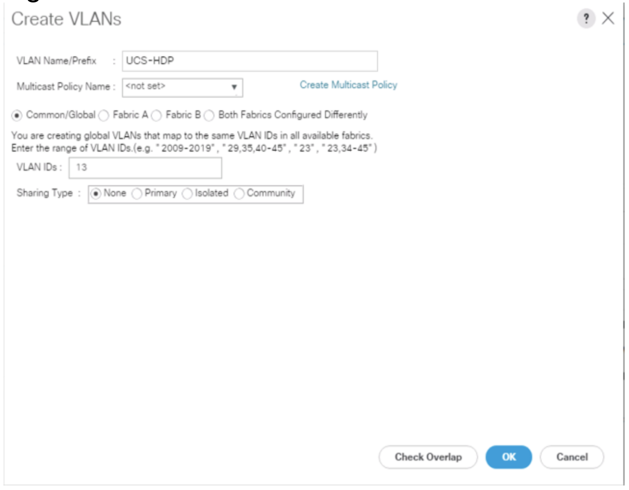
1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select LAN > LAN Cloud.
3. Right-click VLANs
4. Select Create VLANs
5. Enter the name of the VLAN to be used.
6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter <VLAN Number> as the ID of the VLAN ID.
8. Keep the Sharing Type as None.

Figure 39 Create VLAN



The NIC will carry the data traffic from VLAN13. A single vNIC is used in this configuration and the Fabric Failover feature in Fabric Interconnects will take care of any physical port down issues. It will be a seamless transition from an application perspective.

Figure 40 Create VLANs



Create VLANs

VLAN Name/Prefix : UCS-HDP

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 : 13

Sharing Type : [✓ None] Primary Isolated Community

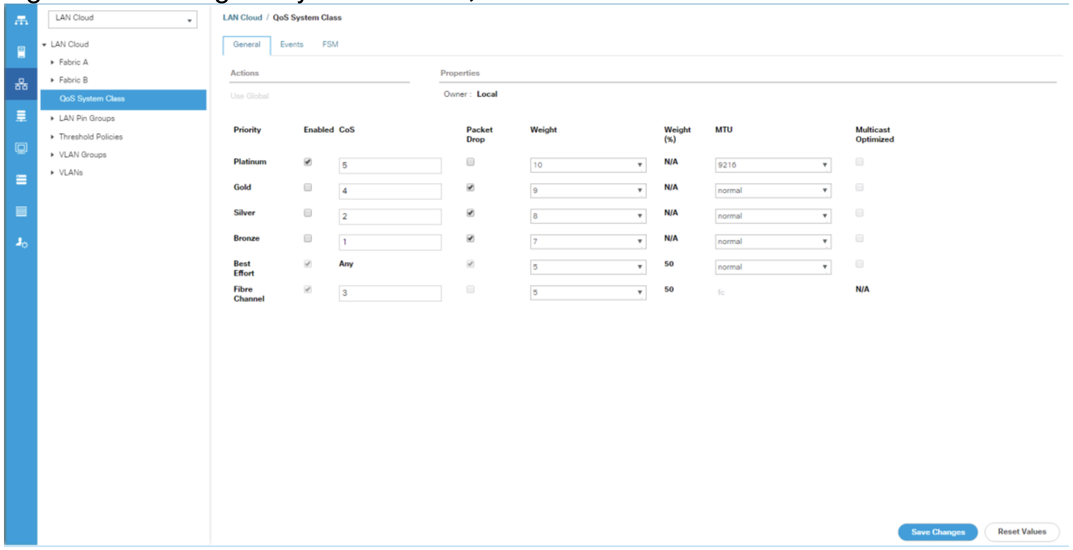
Set System Class QoS and Jumbo Frame in Both Cisco Fabric Interconnects

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

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

Changing QoS system class MTU requires reboot of Cisco UCS Fabric Interconnect for changes to be effective.

Figure 41 Configure System Class QoS on Cisco UCS Fabric Interconnects



Owner : Local

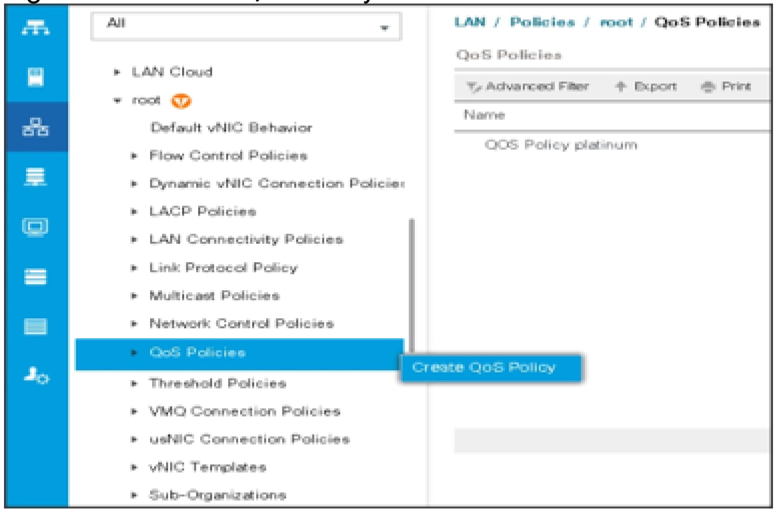
Priority	Enabled	CoS	Packet Drop	Weight	Weight (%)	MTU	Multicast Optimized
Platinum	✓	5	☐	10	N/A	9216	☐
Gold	☐	4	✓	9	N/A	normal	☐
Silver	☐	2	✓	8	N/A	normal	☐
Bronze	☐	1	✓	7	N/A	normal	☐
Best Effort	✓	Any	✓	5	50	normal	☐
Fibre Channel	✓	3	☐	5	50	fc	N/A

Create QoS Policies

To create the QoS policy to assign priority based on the class using the Cisco UCS Manager GUI, follow these steps:

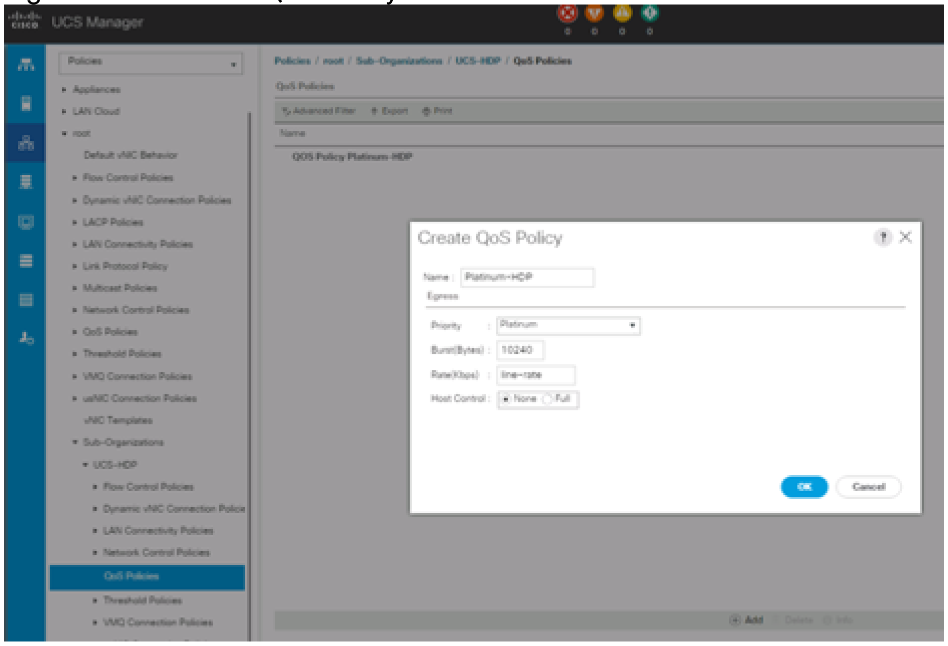
1. Select LAN tab in the left pane in the Cisco UCS Manager GUI.
2. Select LAN > Policies > root > UCS-HDP > QoS Policies.
3. Right-click QoS Policies.
4. Select Create QoS Policy.

Figure 42 Create QoS Policy



We created a Platinum class QoS policy for this solution.

Figure 43 Platinum QoS Policy



Create QoS Policy

Name : Platinum-HDP

Egress

Priority : Platinum

Burst(Bytes) : 10240

Rate(Kbps) : line-rate

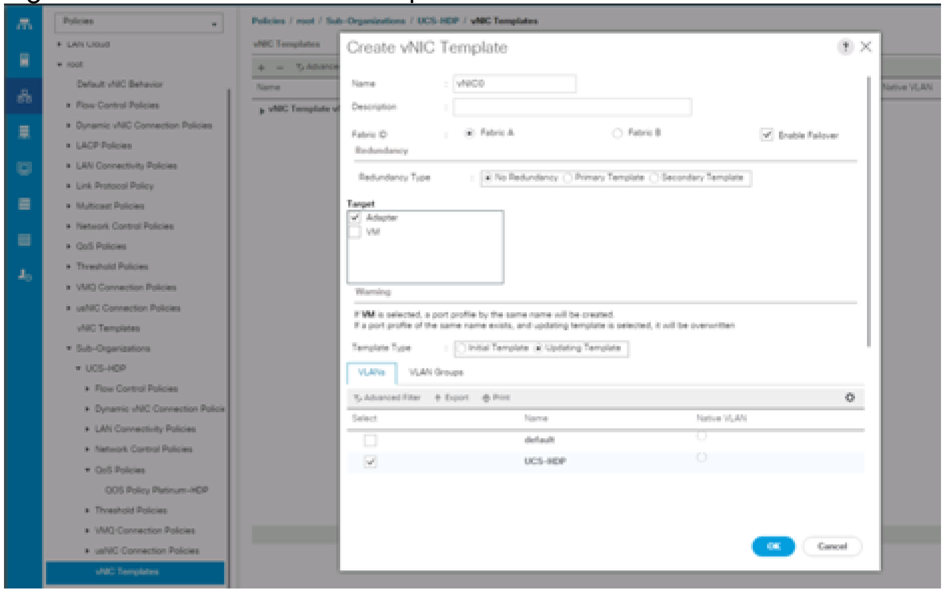
Host Control : [✓ None] Full

Create vNIC Templates

To create multiple virtual network interface card (vNIC) templates for the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Policies > root > Sub-Organization > UCS-HDP > vNIC Template.
3. Right-click vNIC Templates.
4. Select Create vNIC Template.
5. Enter name for vNIC template.
6. Keep Fabric A selected. Select the Enable Failover checkbox.
7. Select Updating Template as the Template Type.
8. Under VLANs, select the checkboxes for desired VLANs to add as part of the vNIC Template.
9. Set Native-VLAN as the native VLAN.
10. For MTU, enter 9000.
11. In the MAC Pool list, select MAC Pool configured.
12. Select QOS policy created earlier.
13. Select default Network Control Policy.
14. Click OK to create the vNIC template.

Figure 44 Create the vNIC Template


Policies / root / Sub-Organizations / UCS-HDP / vNIC Templates

Create vNIC Template

Name : vNIC0

Description :

Fabric ID : [✓ Fabric A] Fabric B [✓ Enable Failover]

Redundancy

Redundancy Type : [✓ No Redundancy] Primary Template Secondary Template

Target

[✓ Adapter]

VM

Warning

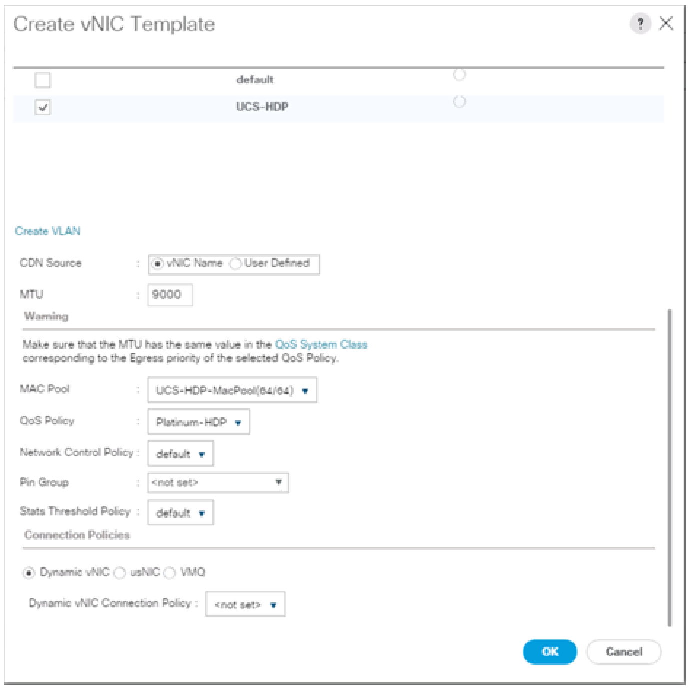
If **VM** is selected, a port profile by the same name will be created.
If a port profile of the same name exists, and updating template is selected, it will be overwritten

Template Type : Initial Template [✓ Updating Template]

VLANs | VLAN Groups

Select	Name	Native VLAN
	default	
✓	UCS-HDP	

OK Cancel


Create vNIC Template

	default	
✓	UCS-HDP	

Create VLAN

CDN Source : [✓ vNIC Name] User Defined

MTU : 9000

Warning

Make sure that the MTU has the same value in the QoS System Class corresponding to the Egress priority of the selected QoS Policy.

MAC Pool : UCS-HDP-MacPool(64/64)

QoS Policy : Platinum-HDP

Network Control Policy : default

Pin Group : <not set>

Stats Threshold Policy : default

Connection Policies

[✓ Dynamic vNIC] usNIC VMQ

Dynamic vNIC Connection Policy : <not set>

OK Cancel

Create 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 create a firmware management policy for a given server configuration in the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root > Sub-Organization > UCS-HDP > Host Firmware Packages.

3. Right-click Host Firmware Packages.

4. Select Create Host Firmware Package.

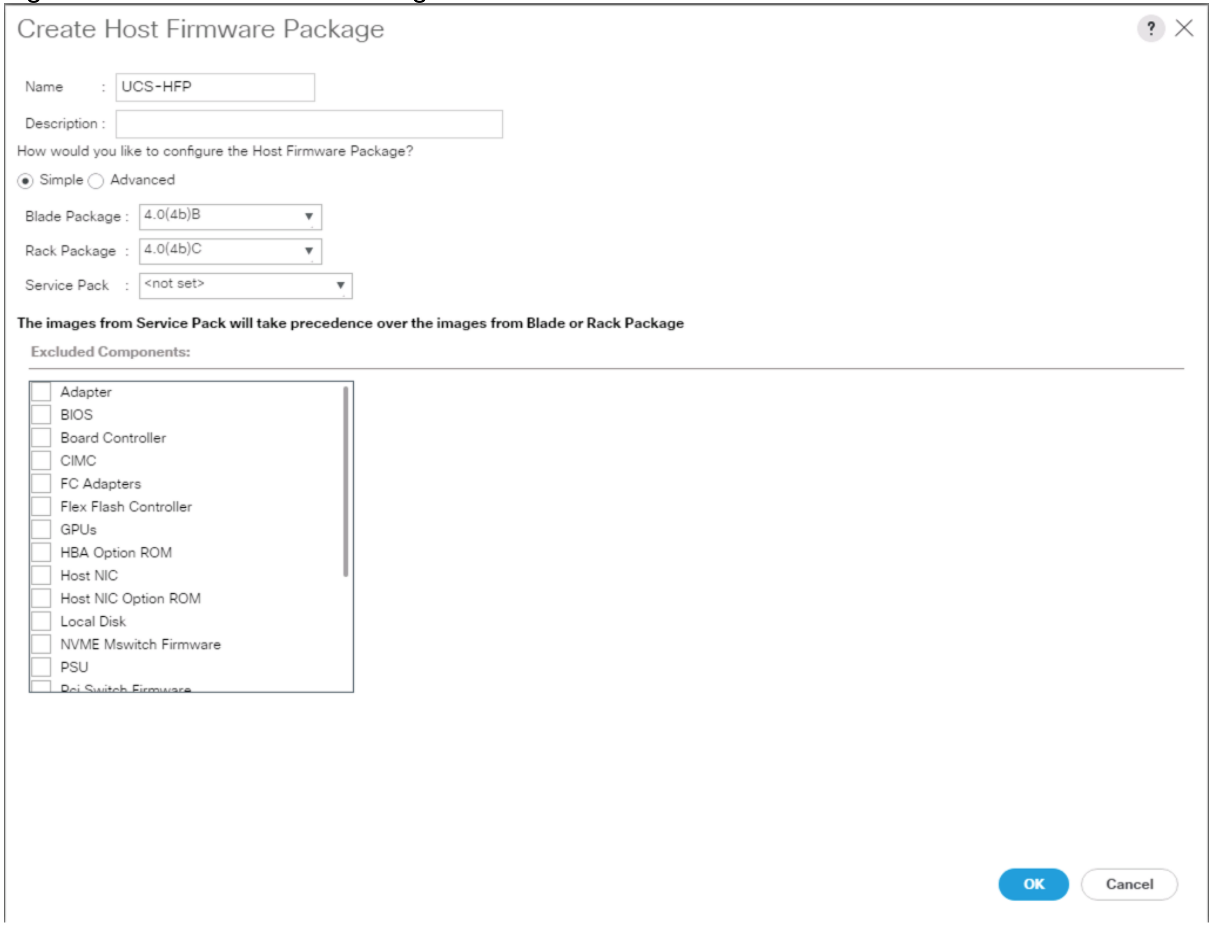
5. Enter name of the host firmware package.

6. Leave Simple selected.

7. Select the version.

8. Click OK to create the host firmware package.

Figure 45 Host Firmware Package



Create Host Firmware Package

Name : UCS-HFP

Description :

How would you like to configure the Host Firmware Package?

[✓ Simple] Advanced

Blade Package : 4.0(4b)B

Rack Package : 4.0(4b)C

Service Pack : <not set>

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

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 tab in the navigation pane.

2. Select Policies > root > Sub-Organization > UCS-HDP > Power Control Policies.

3. Right-click Power Control Policies.

4. Select Create Power Control Policy.

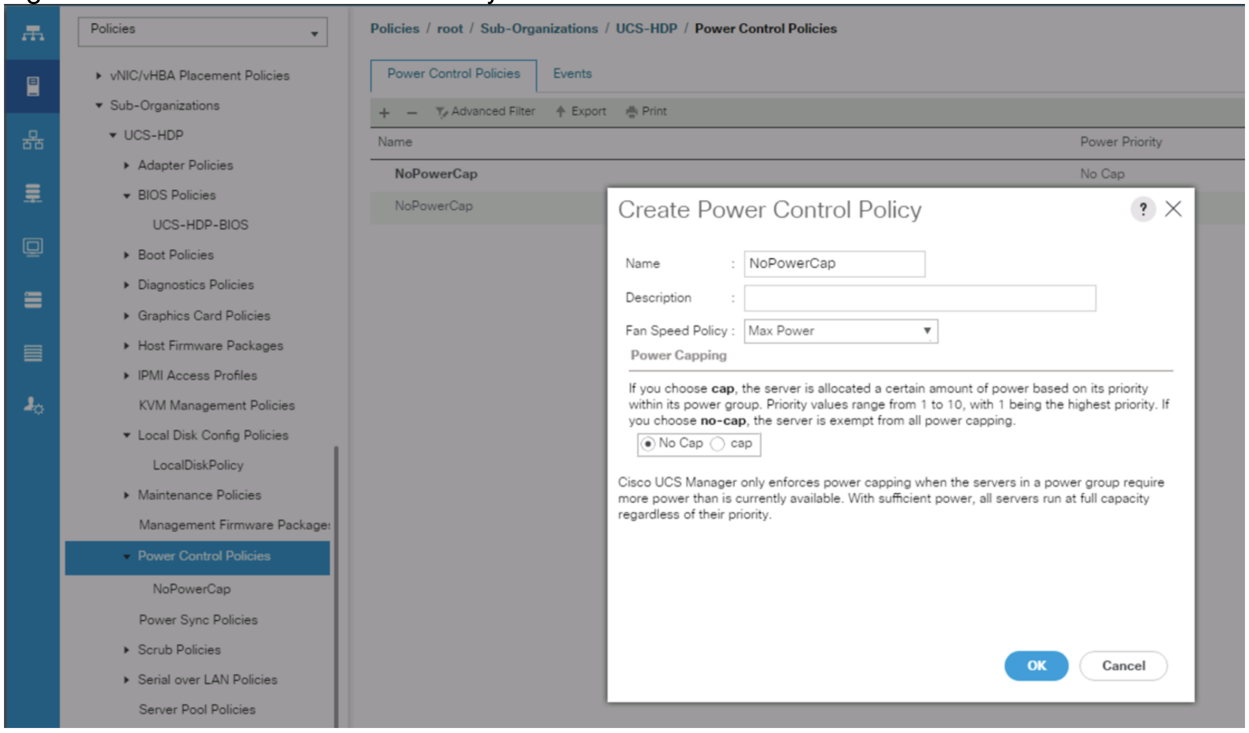
5. Select Fan Speed Policy as "Max Power".

6. Enter NoPowerCap as the power control policy name.

7. Change the power capping setting to No Cap.

8. Click OK to create the power control policy.

Figure 46 Create Power Control Policy



Policies / root / Sub-Organizations / UCS-HDP / Power Control Policies

Create Power Control Policy

Name : NoPowerCap

Description :

Fan Speed Policy : Max Power

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 environment, follow these steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.

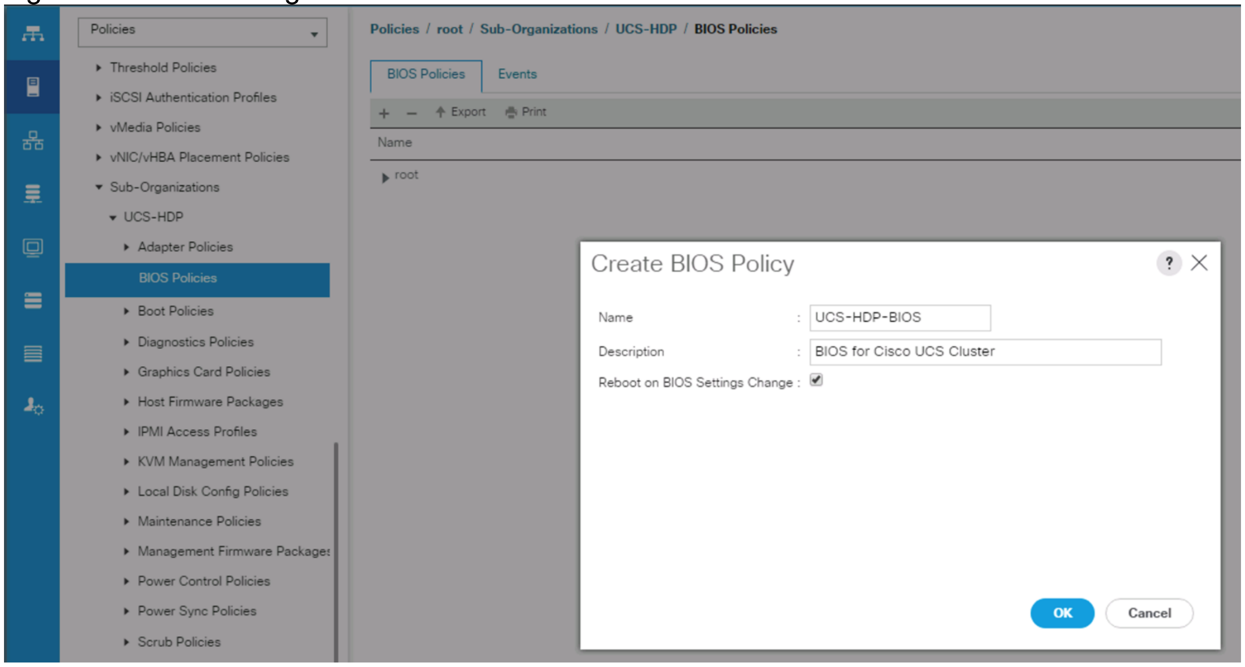
2. Select Policies > root > Sub-Organization > UCS-HDP > BIOS Policies.

3. Right-click BIOS Policies.

4. Select Create BIOS Policy.

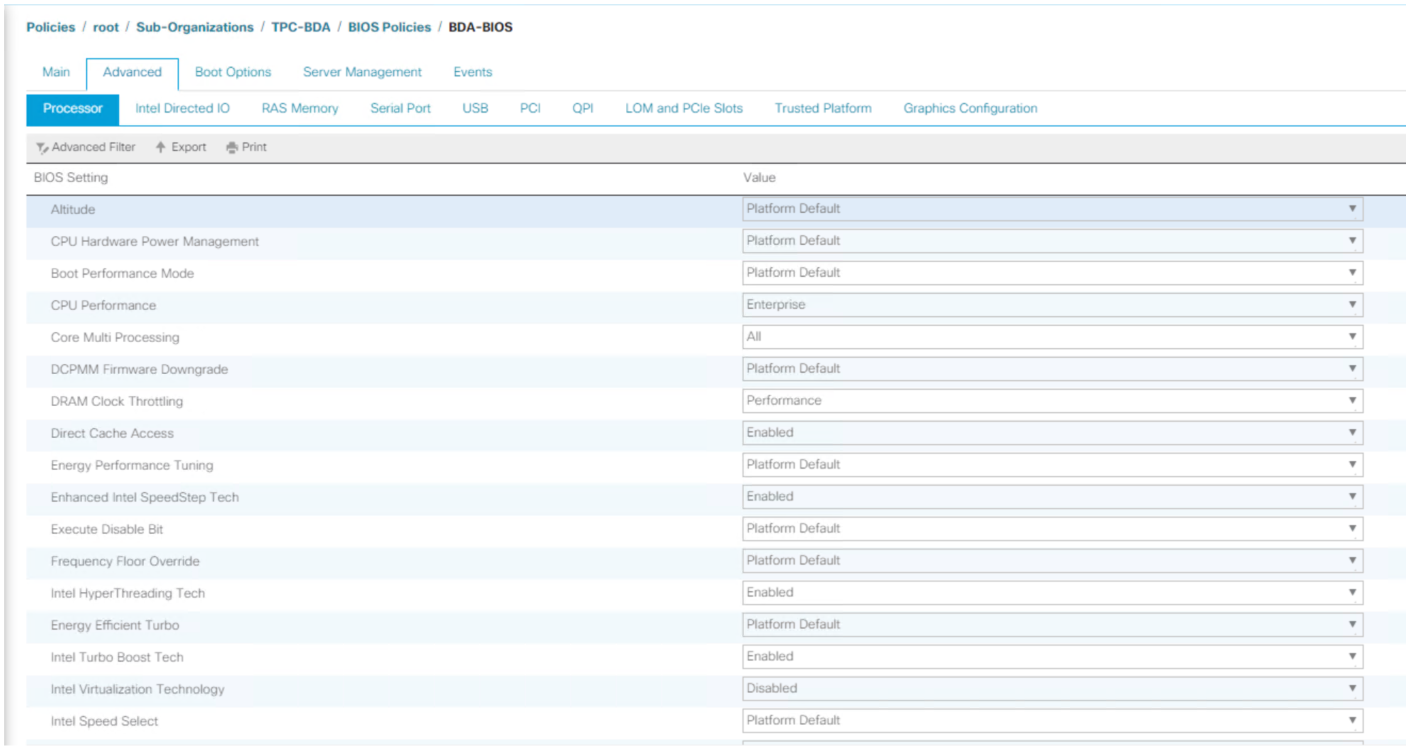
5. Enter the BIOS policy name.

Figure 47 BIOS Configuration

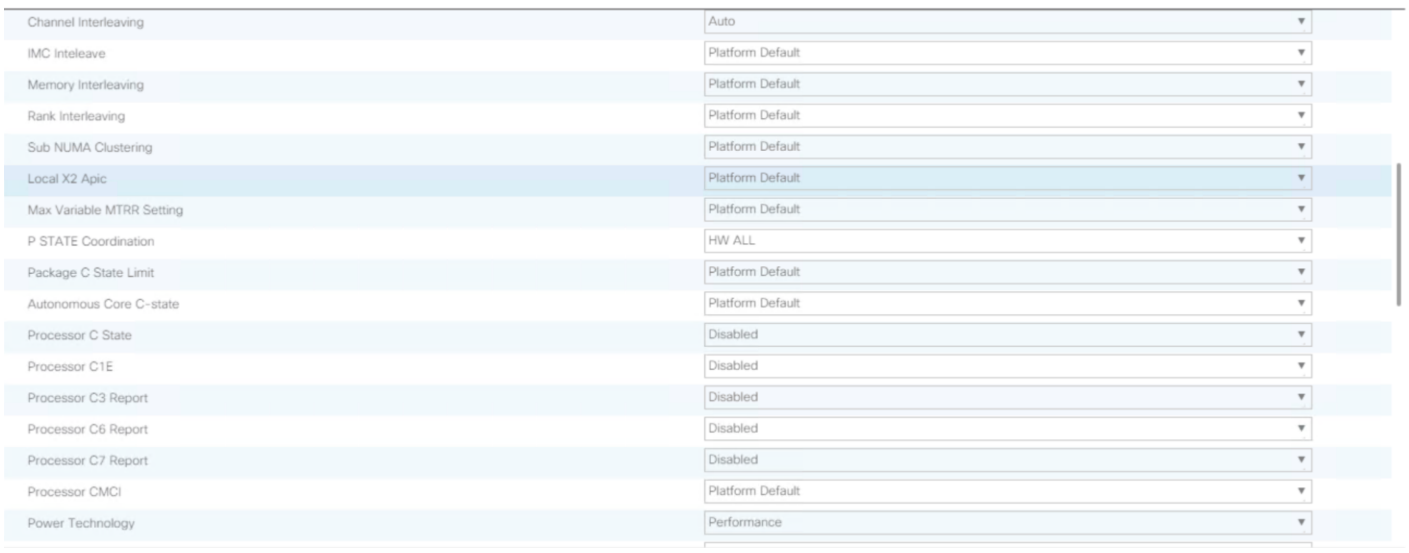


Create BIOS Policy

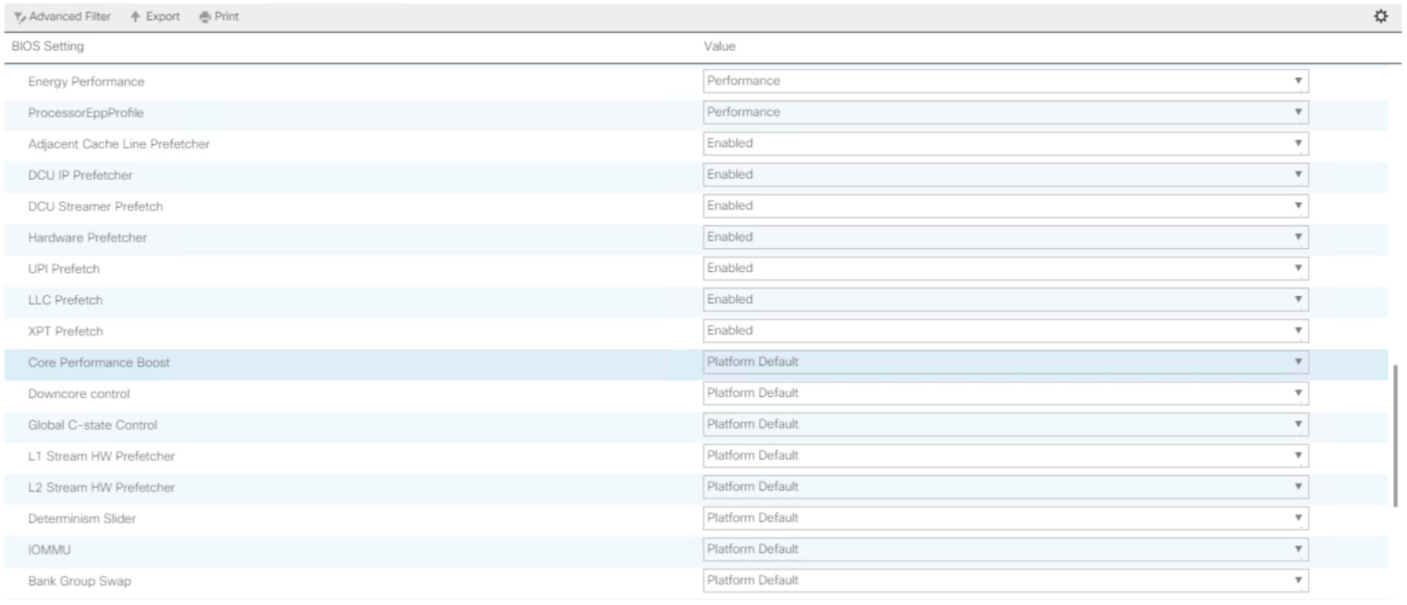
Name :	UCS-HDP-BIOS
Description :	BIOS for Cisco UCS Cluster
Reboot on BIOS Settings Change :	✓



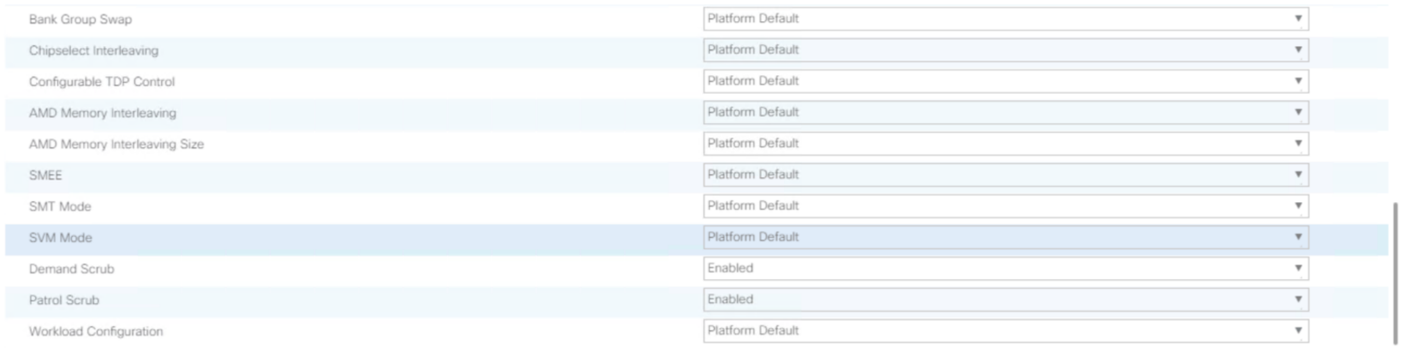
BIOS Setting	Value
Altitude	Platform Default
CPU Hardware Power Management	Platform Default
Boot Performance Mode	Platform Default
CPU Performance	Enterprise
Core Multi Processing	All
DCPMM Firmware Downgrade	Platform Default
DRAM Clock Throttling	Performance
Direct Cache Access	Enabled
Energy Performance Tuning	Platform Default
Enhanced Intel SpeedStep Tech	Enabled
Execute Disable Bit	Platform Default
Frequency Floor Override	Platform Default
Intel HyperThreading Tech	Enabled
Energy Efficient Turbo	Platform Default
Intel Turbo Boost Tech	Enabled
Intel Virtualization Technology	Disabled
Intel Speed Select	Platform Default



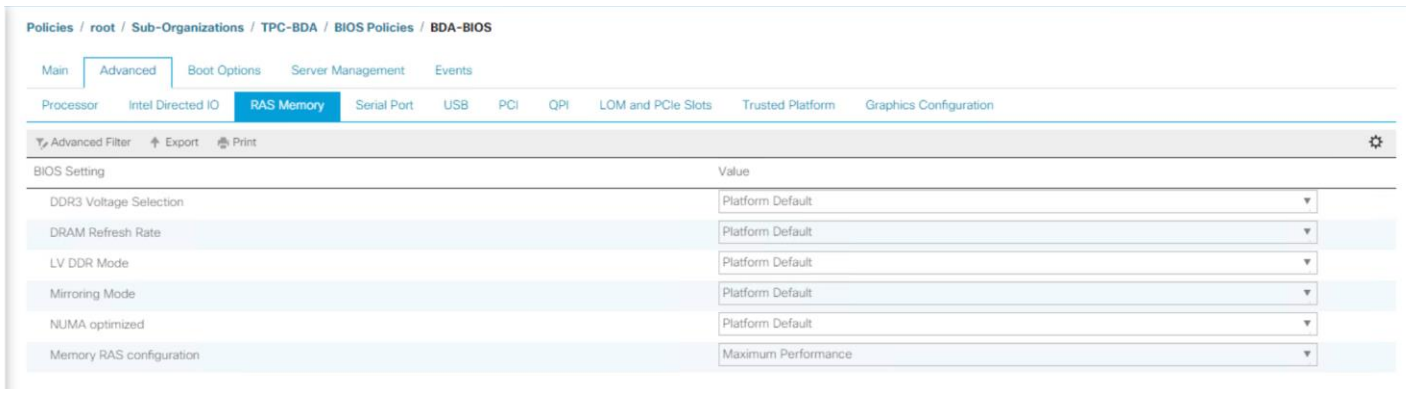
Channel Interleaving	Auto
IMC Inteleave	Platform Default
Memory Interleaving	Platform Default
Rank Interleaving	Platform Default
Sub NUMA Clustering	Platform Default
Local X2 Apic	Platform Default
Max Variable MTRR Setting	Platform Default
P STATE Coordination	HW ALL
Package C State Limit	Platform Default
Autonomous Core C-state	Platform Default
Processor C State	Disabled
Processor C1E	Disabled
Processor C3 Report	Disabled
Processor C6 Report	Disabled
Processor C7 Report	Disabled
Processor CMCI	Platform Default
Power Technology	Performance



BIOS Setting	Value
Energy Performance	Performance
ProcessorEppProfile	Performance
Adjacent Cache Line Prefetcher	Enabled
DCU IP Prefetcher	Enabled
DCU Streamer Prefetch	Enabled
Hardware Prefetcher	Enabled
UPI Prefetch	Enabled
LLC Prefetch	Enabled
XPT Prefetch	Enabled
Core Performance Boost	Platform Default
Downcore control	Platform Default
Global C-state Control	Platform Default
L1 Stream HW Prefetcher	Platform Default
L2 Stream HW Prefetcher	Platform Default
Determinism Slider	Platform Default
IOMMU	Platform Default
Bank Group Swap	Platform Default



Bank Group Swap	Platform Default
Chipselect Interleaving	Platform Default
Configurable TDP Control	Platform Default
AMD Memory Interleaving	Platform Default
AMD Memory Interleaving Size	Platform Default
SMEE	Platform Default
SMT Mode	Platform Default
SVM Mode	Platform Default
Demand Scrub	Enabled
Patrol Scrub	Enabled
Workload Configuration	Platform Default



BIOS Setting	Value
DDR3 Voltage Selection	Platform Default
DRAM Refresh Rate	Platform Default
LV DDR Mode	Platform Default
Mirroring Mode	Platform Default
NUMA optimized	Platform Default
Memory RAS configuration	Maximum Performance

Cisco UCS M5 Server Performance Tuning guide: https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-b-series-blade-servers/whitepaper_c11-740098.pdf.

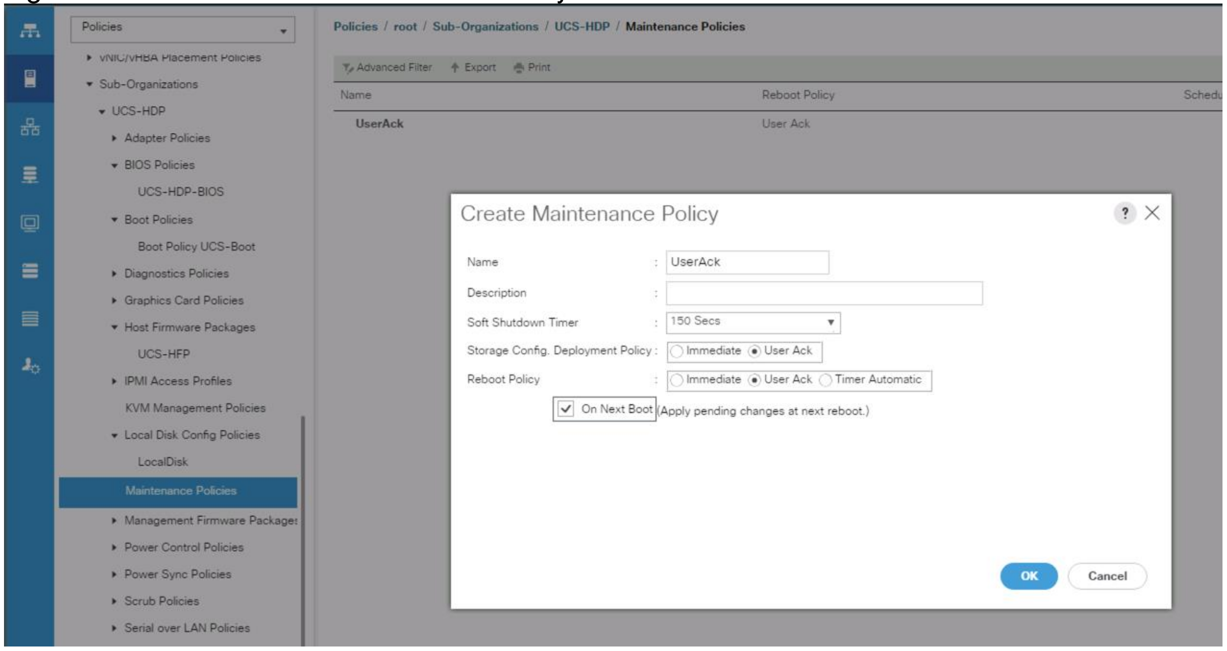
BIOS settings can have a significant performance impact, depending on the workload and the applications. The BIOS settings listed in this section is for configurations optimized for best performance which can be adjusted based on the application, performance, and energy efficiency requirements.

Configure Maintenance Policy

To update the default Maintenance Policy, follow these steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root > Sub-Organization > UCS-HDP > Maintenance Policies.
3. Right-click Maintenance Policies to create a new policy.
4. Enter name for Maintenance Policy
5. Change the Reboot Policy to User Ack.
6. Click Save Changes.
7. Click OK to accept the change.

Figure 48 Create Server Maintenance Policy



Create Maintenance Policy

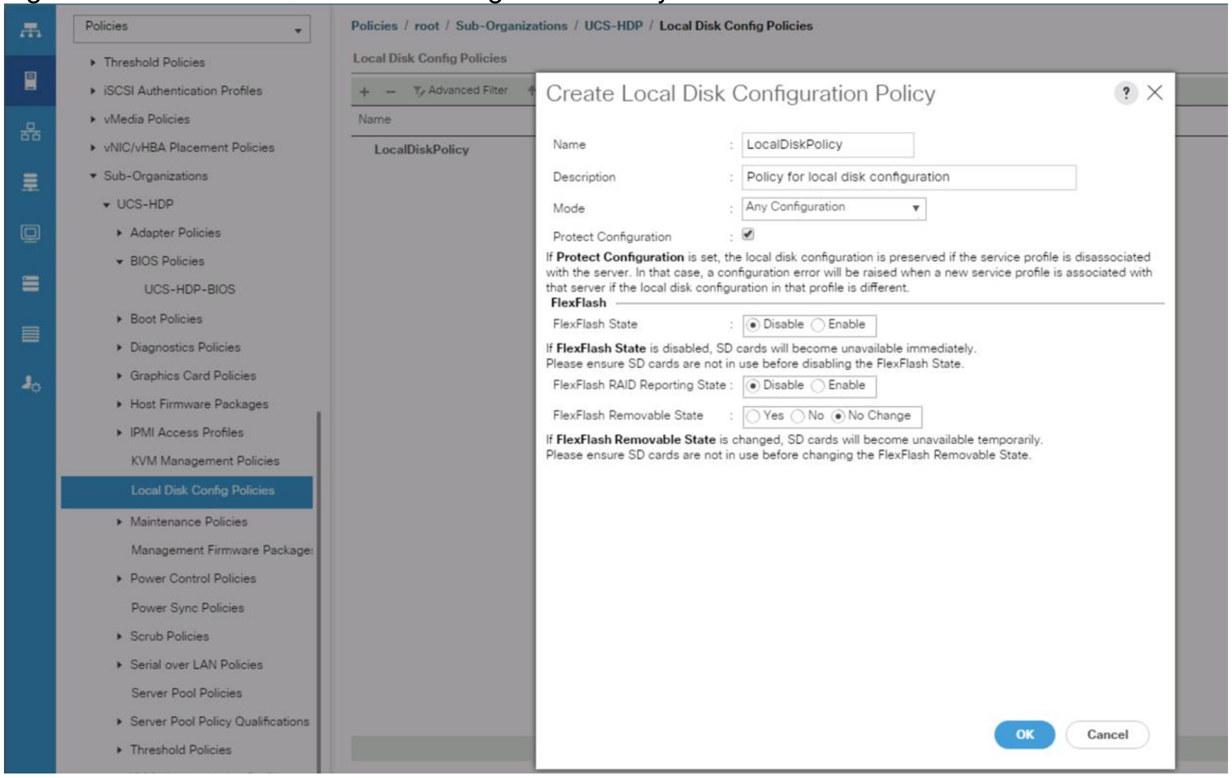
Name : UserAck
Description :
Soft Shutdown Timer : 150 Secs
Storage Config. Deployment Policy : Immediate [✓ User Ack]
Reboot Policy : Immediate [✓ User Ack] Timer Automatic
[✓ On Next Boot] (Apply pending changes at next reboot.)

Create the Local Disk Configuration Policy

To create local disk configuration in the Cisco UCS Manager GUI, follow these steps:

1. Select the Servers tab on the left pane in the Cisco UCS Manager GUI.
2. Select Policies > root > Sub-Organization > UCS-HDP > Local Disk Config Policies.
3. Right-click Local Disk Config Policies and Select Create Local Disk Config Policies.
4. Enter UCS-Boot as the local disk configuration policy name.
5. Change the Mode to Any Configuration. Check the Protect Configuration box.
6. Keep the FlexFlash State field as default (Disable).
7. Keep the FlexFlash RAID Reporting State field as default (Disable).
8. Click OK to complete the creation of the Local Disk Configuration Policy.
9. Click OK.

Figure 49 Create the Local Disk Configuration Policy



Create Local Disk Configuration Policy

Name :	LocalDiskPolicy
Description :	Policy for local disk configuration
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

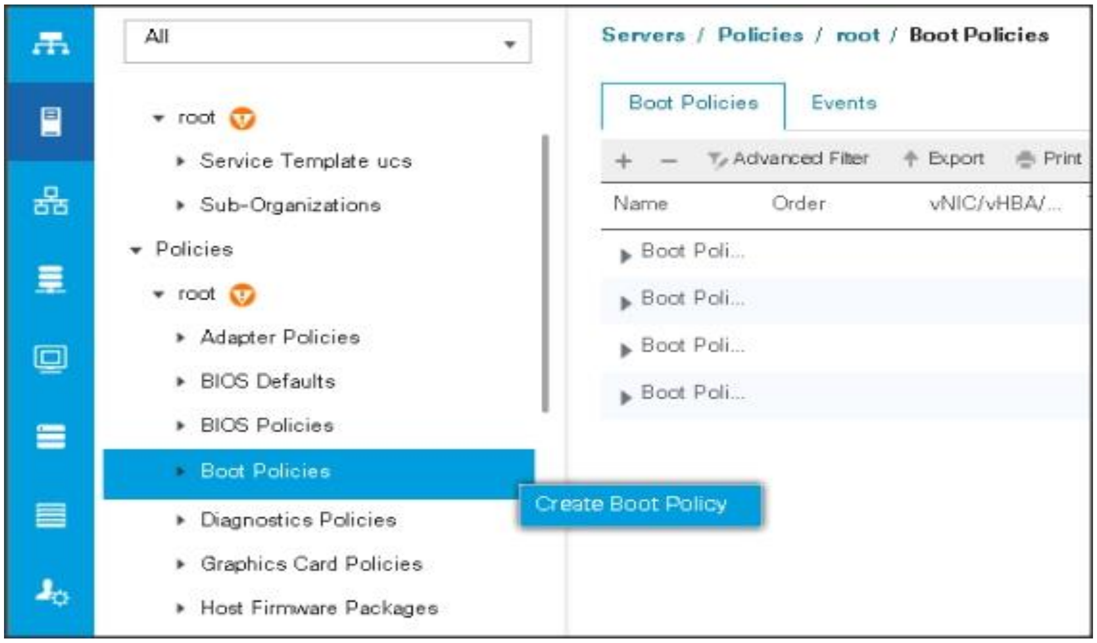
FlexFlash Removable State : Yes No [✓ No Change]

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.

Create Boot Policy

To create boot policies within the Cisco UCS Manager GUI, follow these steps:

1. Select the Servers tab in the left pane in the Cisco UCS Manager GUI.
2. Select Policies > root.
3. Right-click the Boot Policies.
4. Select Create Boot Policy.



5. Enter ucs for the boot policy name.

6. (Optional) enter a description for the boot policy.

7. Keep the Reboot on Boot Order Change check box unchecked.

8. Keep Enforce vNIC/vHBA/iSCSI Name check box checked.

9. Keep Boot Mode Default (Legacy).

10. Expand Local Devices > Add CD/DVD and select Add Local CD/DVD.

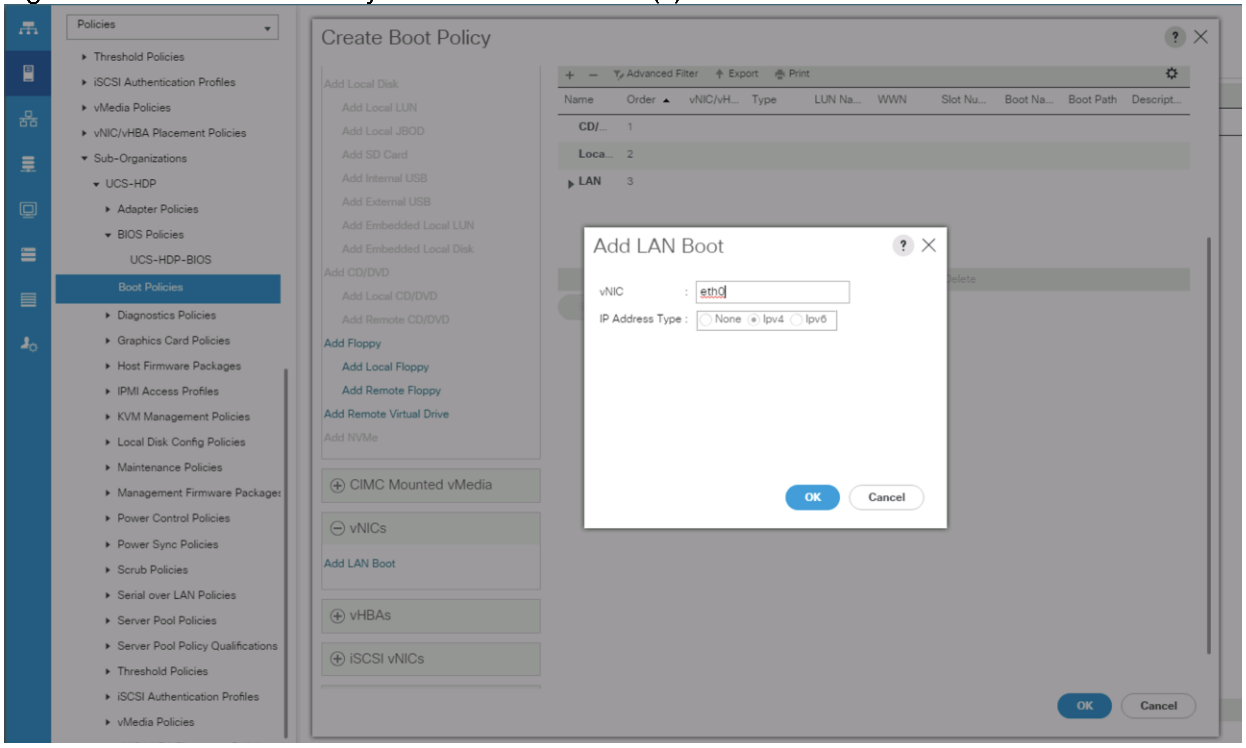
11. Expand Local Devices and select Add Local Disk.

12. Expand vNICs and select Add LAN Boot and enter eth0.

13. Click OK to add the Boot Policy.

14. Click OK.

Figure 50 Create Boot Policy for Cisco UCS Server(s)

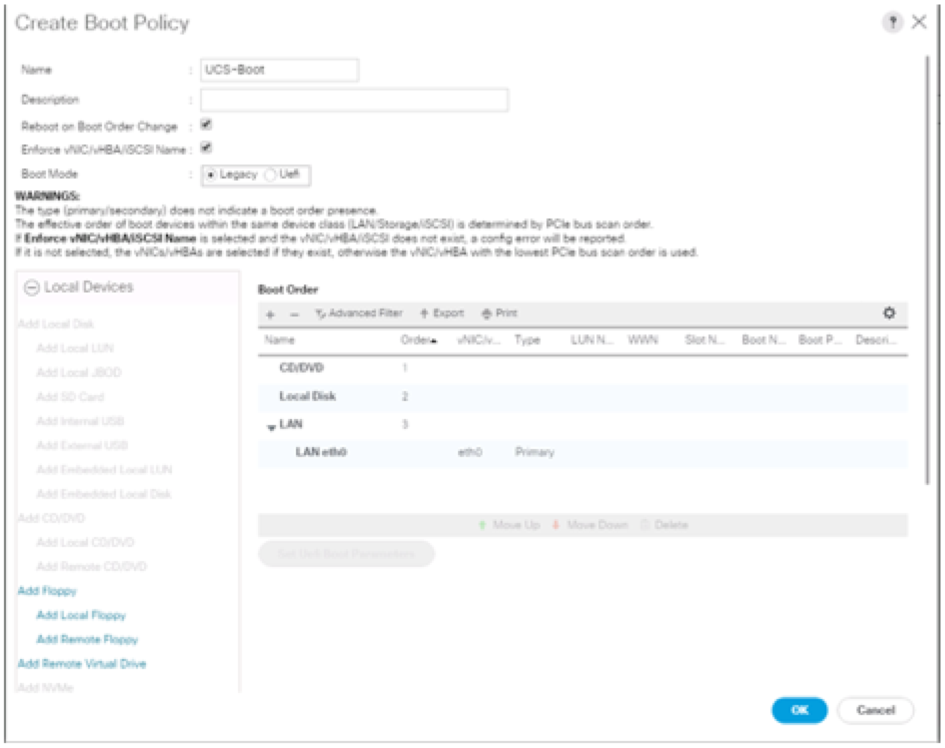


Name	Order	vNIC/vH...	Type	LUN Na...	WWN	Slot Nu...	Boot Na...	Boot Path	Descript...
CD/...	1								
Loca...	2								
LAN	3								

Add LAN Boot

vNIC : eth0

IP Address Type : None [✓ Ipv4] Ipv6



Create Boot Policy

Name : UCS-Boot

Description :

Reboot on Boot Order Change : ✓

Enforce vNIC/vHBA/iSCSI Name : ✓

Boot Mode : [✓ Legacy] Uefi

WARNINGS:
The type (primary/secondary) does not indicate a boot order presence.
The effective order of boot devices within the same device class (LAN/Storage/iSCSI) is determined by PCIe bus scan order.
If **Enforce vNIC/vHBA/iSCSI Name** is selected and the vNIC/vHBA/iSCSI does not exist, a config error will be reported.
If it is not selected, the vNICs/vHBAs are selected if they exist, otherwise the vNIC/vHBA with the lowest PCIe bus scan order is used.

Boot Order

Name	Order	vNIC/v...	Type	LUN N...	WWN	Slot N...	Boot N...	Boot P...	Descri...
CD/DVD	1								
Local Disk	2								
LAN	3								
LAN eth0		eth0	Primary						

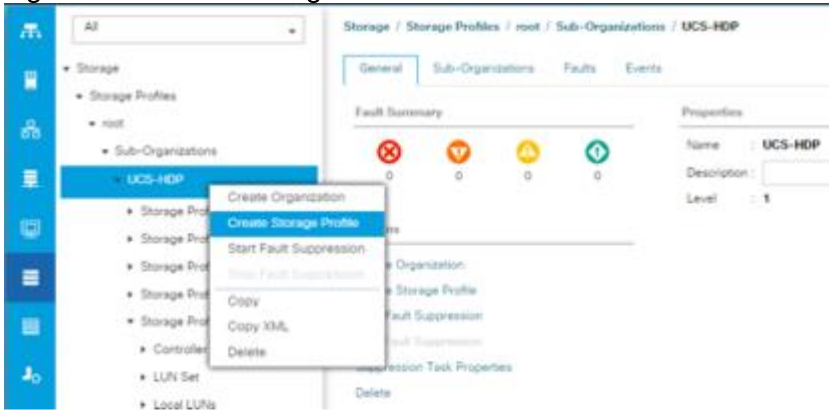
Create Storage Profile for Individual RAID0

To create the storage profile for the individual RAIDP, follow these steps:

1. On the UCSM navigation page on the left-hand side select Storage tab.

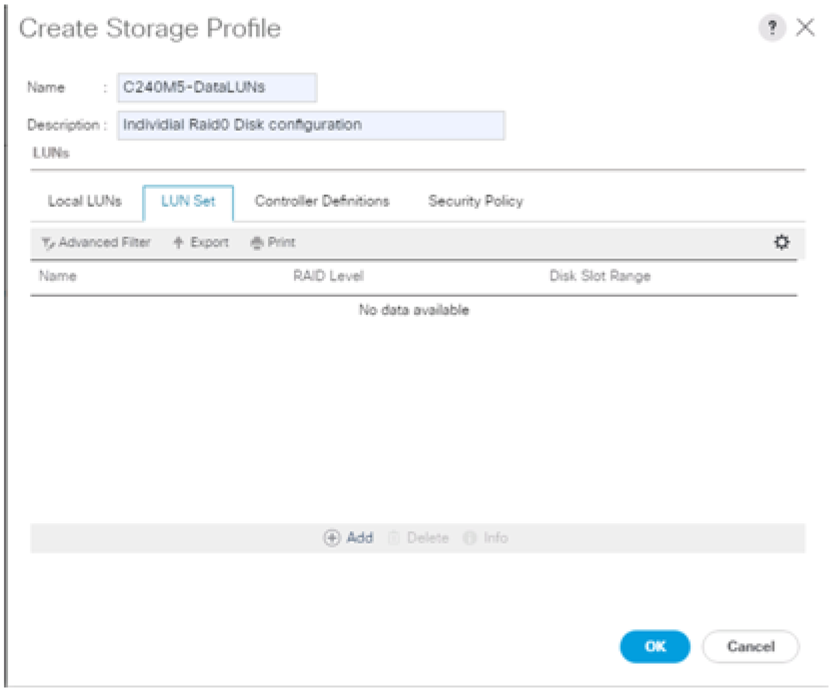
2. From the Storage Profiles drop-down list, right-click and select Create Storage Profile.

Figure 51 Create Storage Profile



3. Enter a name for the Storage Profile and click the LUN Set tab.

4. Click Add.



Create Storage Profile

Name : C240M5-DataLUNs

Description : Individial Raid0 Disk configuration

LUNs

Local LUNs | LUN Set | Controller Definitions | Security Policy

Advanced Filter | Export | Print

Name	RAID Level	Disk Slot Range
No data available		

Add | Delete | Info

OK | Cancel

The LUN Set policy configures all disks managed through Cisco UCS S3260 Dual Raid Controller on S3260 and Cisco 12G Modular Raid controller to individual disk RAID0.

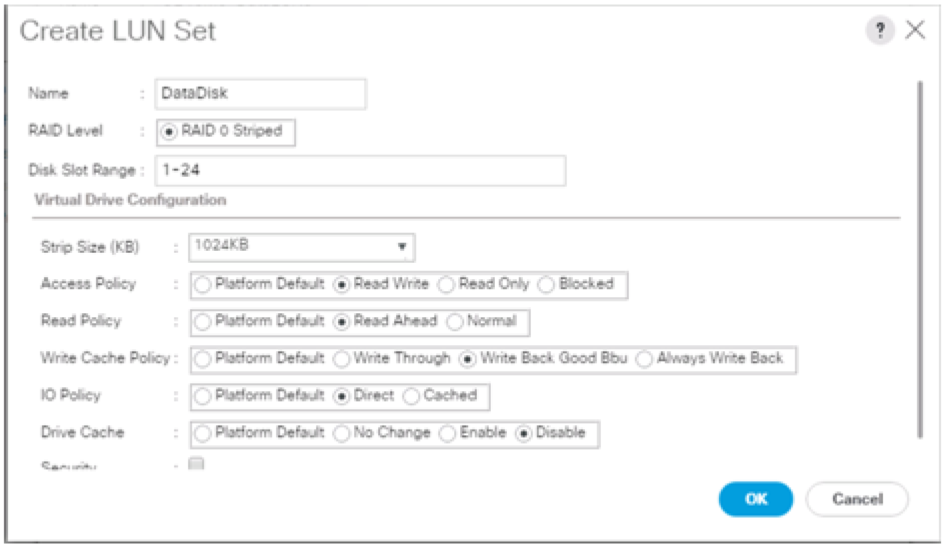
5. Select the properties for the LUN set:

   a. Enter a name for LUN set.

   b. Disk Slot Range – 1 – 24/26/56 (Depends on number of drives installed in a server).

   c. Enter Virtual Drive configuration:

i. Strip Size(kb) – 1024KB
ii. Access Policy – Read Write
iii. Read Policy – Read Ahead
iv. Write Cache Policy – Write Back Good Bbu
v. IO Policy – Direct
vi. Drive Cache – Disable


Create LUN Set

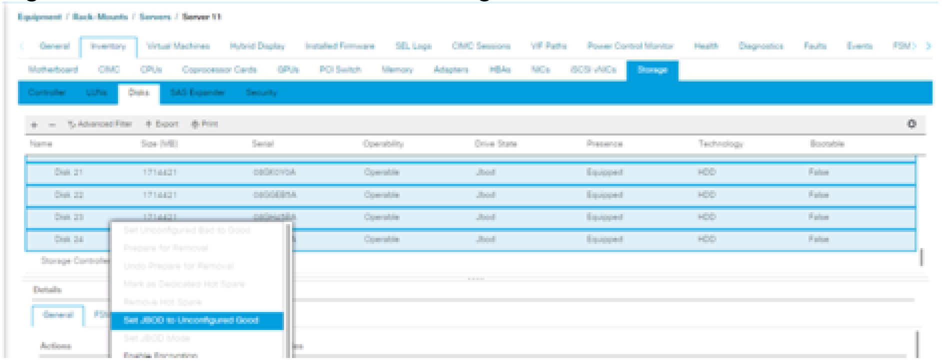
Name	DataDisk
RAID Level	[✓ RAID 0 Striped]
Disk Slot Range	1-24

Virtual Drive Configuration

Strip Size (KB)	1024KB
Access Policy	Platform Default [✓ Read Write] Read Only Blocked
Read Policy	Platform Default [✓ Read Ahead] Normal
Write Cache Policy	Platform Default Write Through [✓ Write Back Good Bbu] Always Write Back
IO Policy	Platform Default [✓ Direct] Cached
Drive Cache	Platform Default No Change Enable [✓ Disable]
Security	

For a LUN set based configuration, set the JBOD disks to unconfigured by selecting all JBOD disk in Server > Inventory > Disks, right-click and select "Set JBOD to Unconfigured Good".

Figure 52 Set JBOD Disks to Unconfigured Good


Equipment / Rack-Mounts / Servers / Server 11

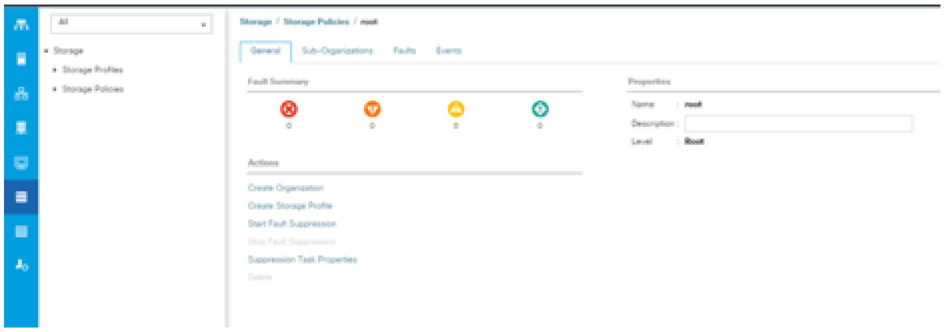
Name	Size (MB)	Serial	Operability	Drive State	Presence	Technology	Bootable
Disk 21	1714421	08GK0Y0A	Operable	Jbod	Equipped	HDD	False
Disk 22	1714421	08GGEB5A	Operable	Jbod	Equipped	HDD	False
Disk 23	1714421	[illegible]	Operable	Jbod	Equipped	HDD	False
Disk 24	[illegible]	[illegible]	Operable	Jbod	Equipped	HDD	False

Create Storage Policy and Storage Profile

To create a Storage Profile with multiple RAID LUNs, create Storage Policies and attach them to a Storage Profile.

To create a Storage Policy and attach them to a Storage Profile, follow these steps:

1. Go to the Storage tab on the left side panel selection, select "Storage Policies".



Storage / Storage Policies / root

General | Sub-Organizations | Faults | Events

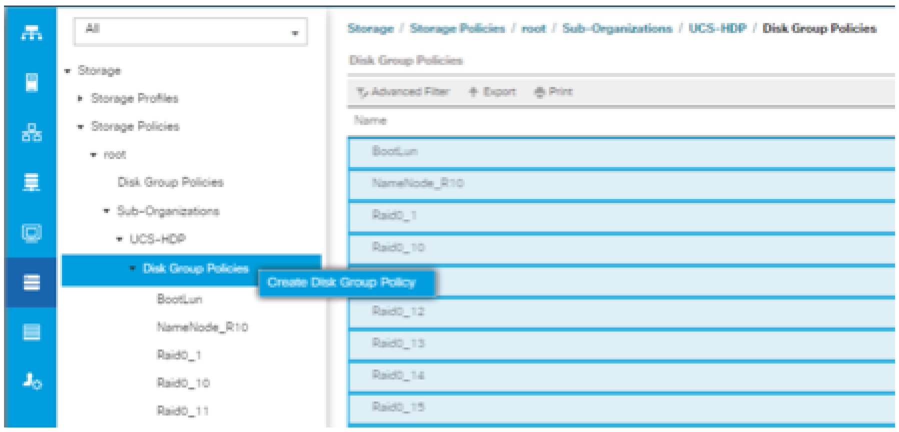
Fault Summary: 0 0 0 0

Actions: Create Organization, Create Storage Profile, Start Fault Suppression, Stop Fault Suppression, Suppression Task Properties, Delete

Properties

Name : root
Description :
Level : Root

2. From the Storage Policies drop-down list, select and right-click “Disk Group Policies”. Select “Create Disk Group Policy”.



Storage / Storage Policies / root / Sub-Organizations / UCS-HDP / Disk Group Policies

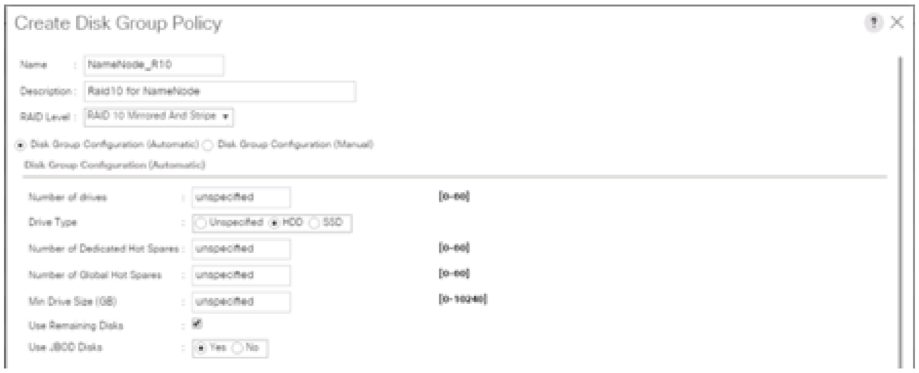
Disk Group Policies

Name
BootLun
NameNode_R10
Raid0_1
Raid0_10
Raid0_12
Raid0_13
Raid0_14
Raid0_15

3. Enter name for Disk Group Policy, Select RAID level.

4. Select “Disk Group Configuration” (Automatic/Manual).

5. Disk Group Configuration.



Create Disk Group Policy

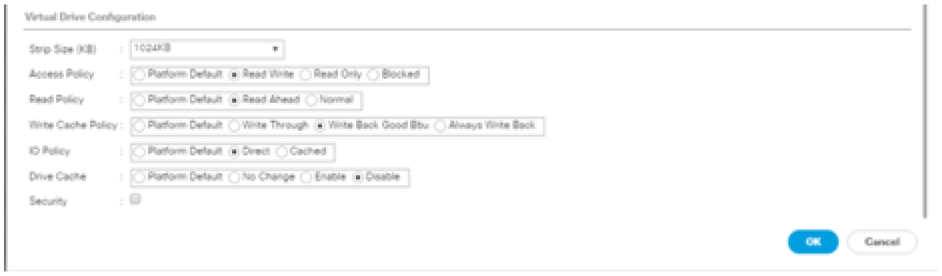
Name : NameNode_R10
Description : Raid10 for NameNode
RAID Level : RAID 10 Mirrored And Stripe

[✓ Disk Group Configuration (Automatic)] Disk Group Configuration (Manual)

Disk Group Configuration (Automatic)

Number of drives	unspecified	[0-60]
Drive Type	Unspecified [✓ HDD] SSD	
Number of Dedicated Hot Spares	unspecified	[0-60]
Number of Global Hot Spares	unspecified	[0-60]
Min Drive Size (GB)	unspecified	[0-10240]
Use Remaining Disks	✓	
Use JBOD Disks	[✓ Yes] No	

6. Virtual Drive Configuration.



Virtual Drive Configuration

Strip Size (KB) : 1024KB

Access Policy : Platform Default [✓ Read Write] Read Only Blocked

Read Policy : Platform Default [✓ Read Ahead] Normal

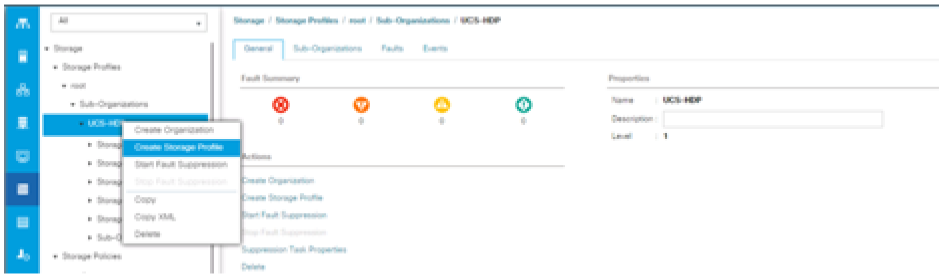
Write Cache Policy : Platform Default Write Through [✓ Write Back Good Bbu] Always Write Back

IO Policy : Platform Default [✓ Direct] Cached

Drive Cache : Platform Default No Change Enable [✓ Disable]

Security : ☐

7. Select Storage Profiles, right-click and select Create Storage Profile.



Storage / Storage Profiles / root / Sub-Organizations / UCS-HDP

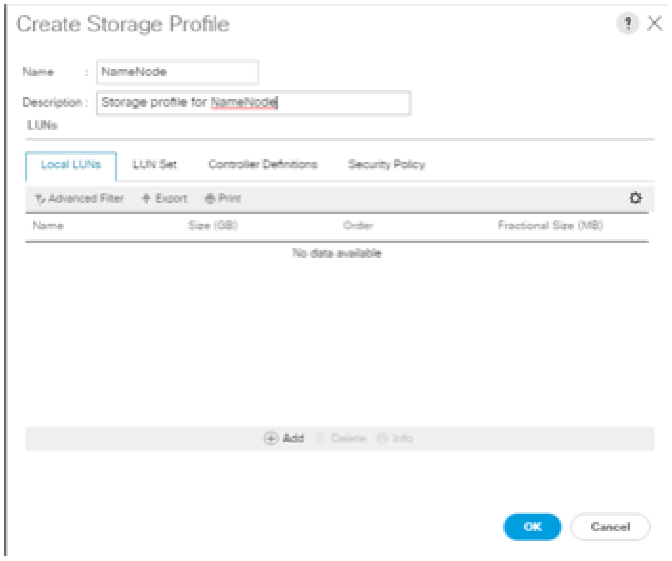
Properties

Name : UCS-HDP

Description :

Level : 1

8. Enter a name for the Storage profile and click Add.



Create Storage Profile

Name : NameNode

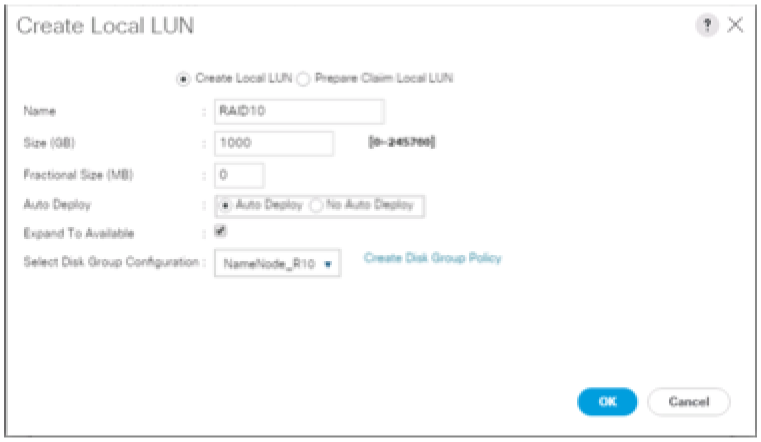
Description : Storage profile for NameNode

LUNs

Name	Size (GB)	Order	Fractional Size (MB)
No data available			

9. Enter a Local LUN name and select Auto Deploy.

10. Check the box for Expand to Available and from the drop-down list select the storage policy you want to attach with the Storage Profile. Click OK.



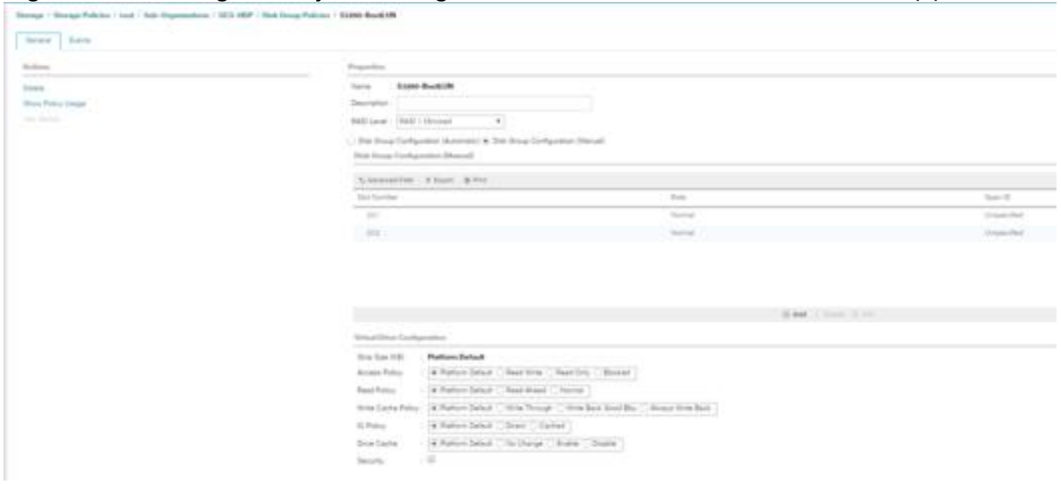
Create Local LUN

[✓ Create Local LUN] Prepare Claim Local LUN

Name	RAID10
Size (GB)	1000 [0-245760]
Fractional Size (MB)	0
Auto Deploy	[✓ Auto Deploy] No Auto Deploy
Expand To Available	✓
Select Disk Group Configuration	NameNode_R10

For Cisco UCS S3260, we created a Storage Profile with a Storage Policy to create a Boot LUN and attached it to a Storage Profile as shown above. The LUN set policy for an individual server node (server node 1 and server node 2) to create an individual RAID0 is shown in Figure 53.

Figure 53 Storage Policy to Configure Boot LUN for S3260 Server Node(s)

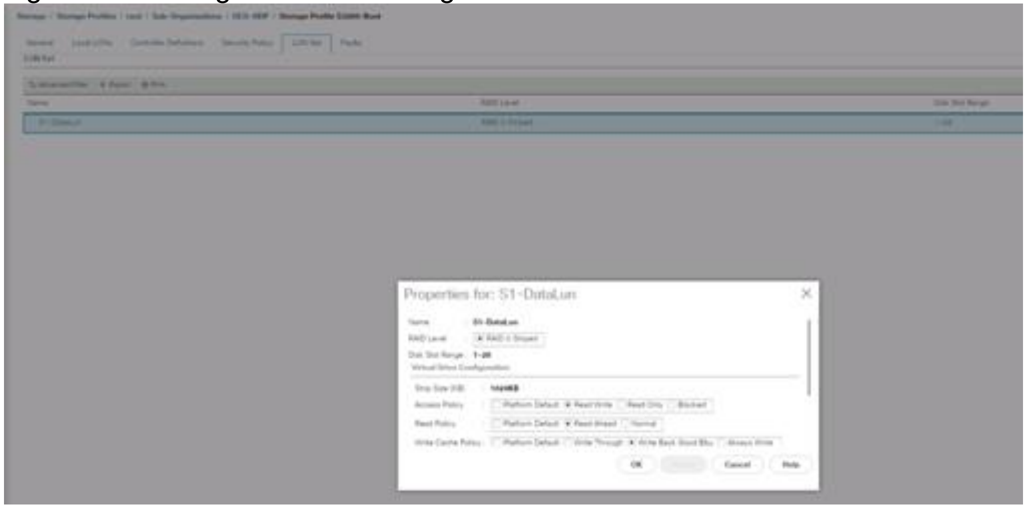


Name : S3260-BootLUN
RAID Level : RAID 1 Mirrored
Disk Group Configuration (Automatic) [✓ Disk Group Configuration (Manual)]

Slot Number	Role	Span ID
201	Normal	Unspecified
202	Normal	Unspecified

Strip Size (KB) : Platform Default
Access Policy : [✓ Platform Default] Read Write Read Only Blocked
Read Policy : [✓ Platform Default] Read Ahead Normal
Write Cache Policy : [✓ Platform Default] Write Through Write Back Good Bbu Always Write Back
IO Policy : [✓ Platform Default] Direct Cached
Drive Cache : [✓ Platform Default] No Change Enable Disable

Figure 54 Storage Profile to Configure Individual RAID 0 on Server Node 1: Disk Slot 1-28

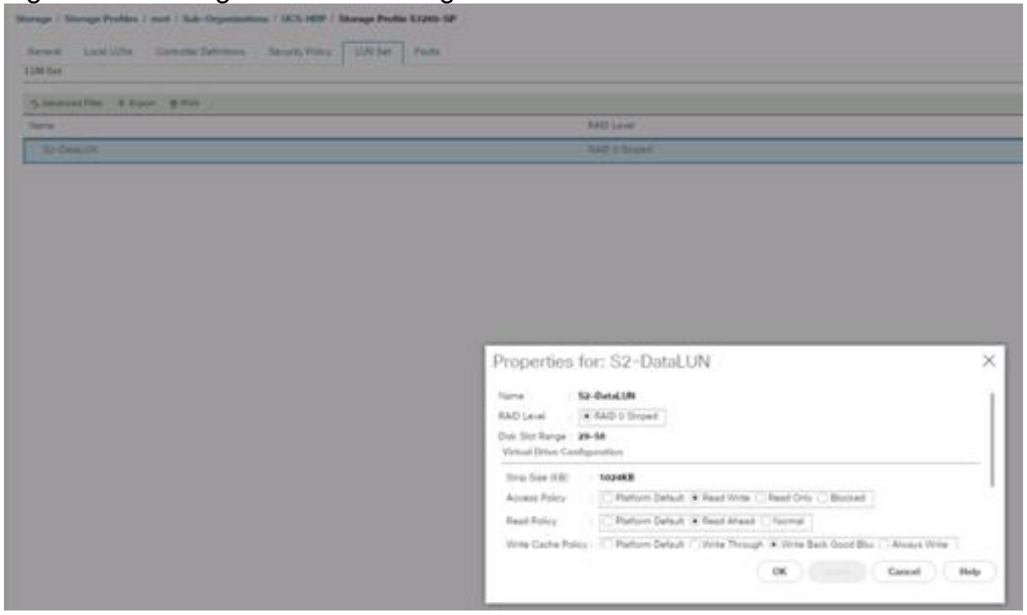


Name	RAID Level	Disk Slot Range
S1-DataLun	RAID 0 Striped	1-28

Properties for: S1-DataLun

Name : S1-DataLun
RAID Level : RAID 0 Striped
Disk Slot Range : 1-28
Strip Size (KB) : 1024KB
Access Policy : Platform Default [✓ Read Write] Read Only Blocked
Read Policy : Platform Default [✓ Read Ahead] Normal
Write Cache Policy : Platform Default Write Through [✓ Write Back Good Bbu] Always Write

Figure 55 Storage Profile to Configure Individual RAID 0 on Server Node 1: Disk Slot 29-56



Name	RAID Level
S2-DataLUN	RAID 0 Striped

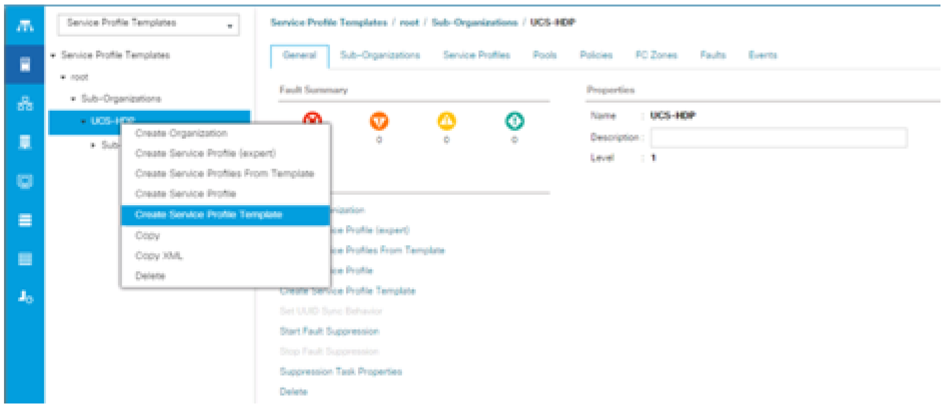
Properties for: S2-DataLUN

Name : S2-DataLUN
RAID Level : [✓ RAID 0 Striped]
Disk Slot Range : 29-56
Virtual Drive Configuration
Strip Size (KB) : 1024KB
Access Policy : Platform Default [✓ Read Write] Read Only Blocked
Read Policy : Platform Default [✓ Read Ahead] Normal
Write Cache Policy : Platform Default Write Through [✓ Write Back Good Bbu] Always Write

Create Service Profile Template

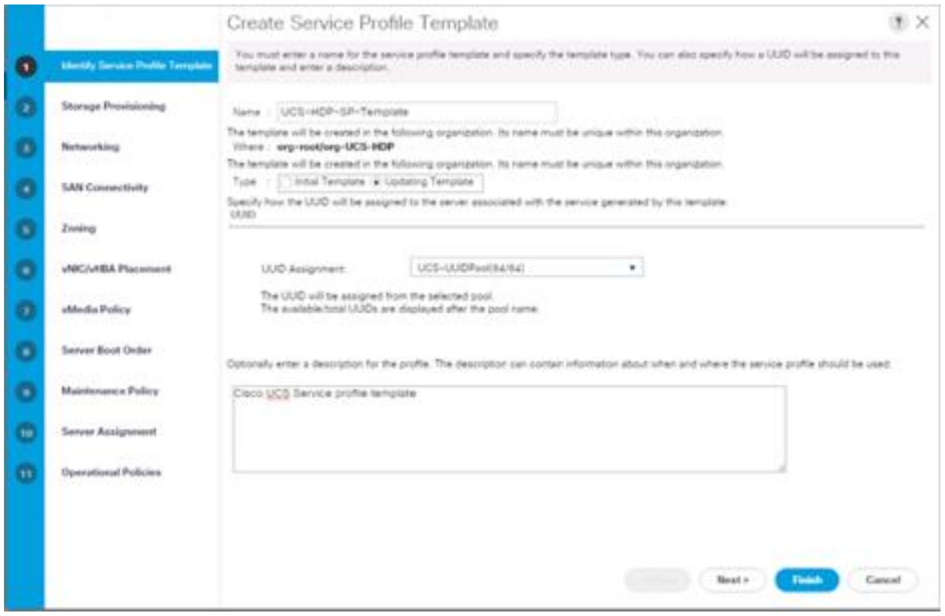
To create a service profile template, follow these steps:

1. In the Cisco UCS Manager, go to Servers > Service Profile Templates > root Sub Organization > UCS-HDP > and right-click “Create Service Profile Template” as shown below.


Service Profile Templates
Service Profile Templates
root
Sub-Organizations
UCS-HDP
Create Organization
Create Service Profile (expert)
Create Service Profiles From Template
Create Service Profile
Create Service Profile Template
Copy
Copy XML
Delete
Service Profile Templates / root / Sub-Organizations / UCS-HDP
General
Sub-Organizations
Service Profiles
Pools
Policies
FC Zones
Faults
Events
Fault Summary
0
0
0
0
Properties
Create Service Profile Template
Set UUID Sync Behavior
Start Fault Suppression
Stop Fault Suppression
Suppression Task Properties
Delete

Name : UCS-HDP
Description :
Level : 1

2. Enter the Service Profile Template name, Updating Template as type of template and select the UUID Pool that was created earlier. Click Next.



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 : UCS-HDP-SP-Template

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

Where : **org-root/org-UCS-HDP**

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

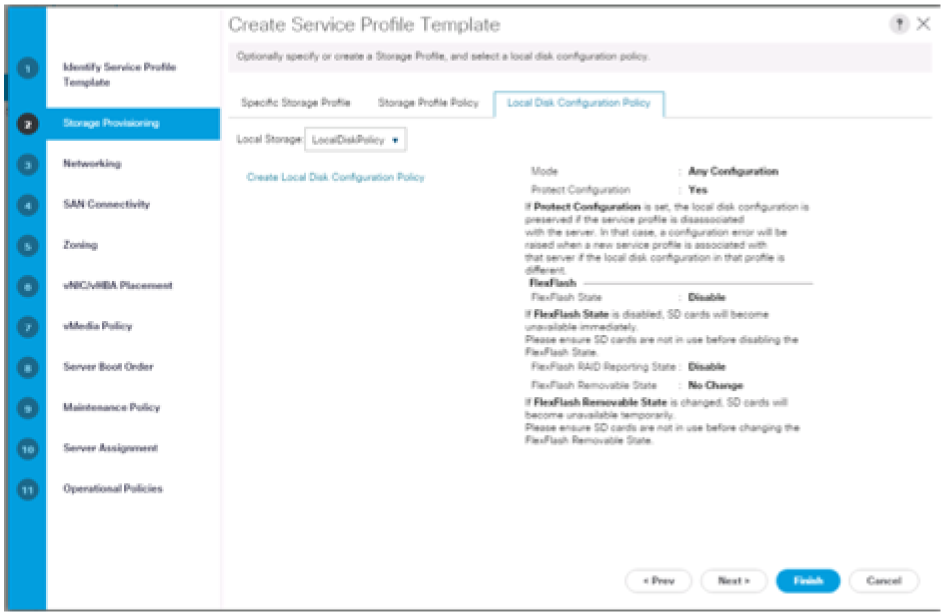
UUID Assignment: UCS-UUIDPool(64/64)

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

Optionally enter a description for the profile. The description can contain information about when and where the service profile should be used.

Cisco UCS Service profile template

3. Select Local Disk Configuration Policy tab and select Local Storage policy from the drop-down list.



Create Service Profile Template

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

Local Storage: LocalDiskPolicy

Mode : **Any Configuration**

Protect Configuration : **Yes**

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**

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

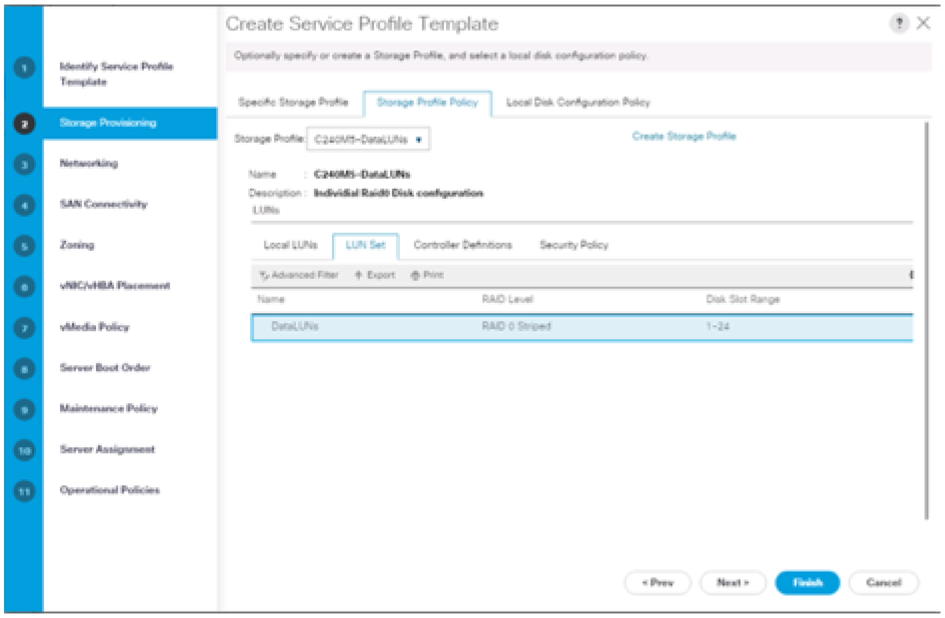
FlexFlash RAID Reporting State : **Disable**

FlexFlash Removable State : **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.

4. On Storage Profile Policy; select Storage Profile to attach with the server.

Based on the server model or the role of the server, we created and attached a Storage Profile for NameNode(s), DataNode(s) and Cisco UCS S3260 Storage server in different Service Profile Template for each.



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

Storage Profile: C240M5-DataLUNs

Name : C240M5-DataLUNs

Description : Individual Raid0 Disk configuration

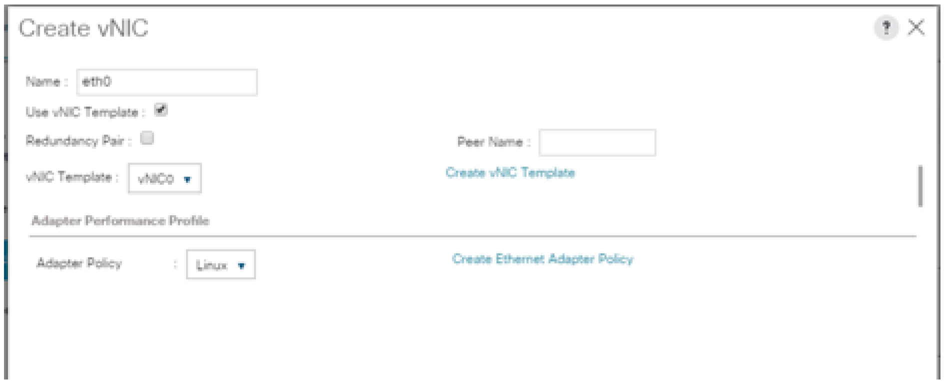
LUNs

Name	RAID Level	Disk Slot Range
DataLUNs	RAID 0 Striped	1-24

5. In the networking window, select “Expert” and click “Add” to create vNICs. Add one or more vNICs that the server should use to connect to the LAN.

6. In the create vNIC menu as vNIC name.

7. Select vNIC Template as vNIC0 and Adapter Policy as Linux.



Name : eth0

Use vNIC Template : ☑

Redundancy Pair : ☐

Peer Name :

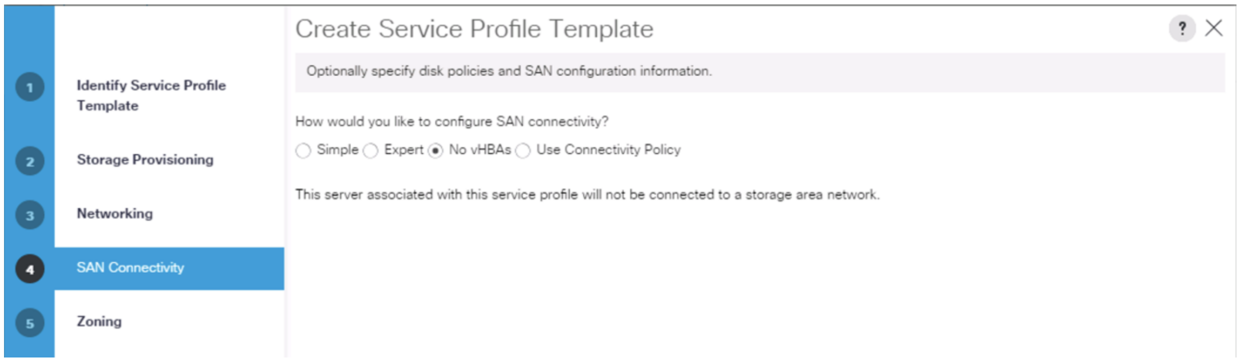
vNIC Template : vNIC0

Adapter Performance Profile

Adapter Policy : Linux

Optionally, Network Bonding can be setup on the vNICs for each host for redundancy as well as for increased throughput.

8. In the SAN Connectivity menu, select no vHBAs.



Create Service Profile Template

1 Identify Service Profile Template
2 Storage Provisioning
3 Networking
4 SAN Connectivity
5 Zoning

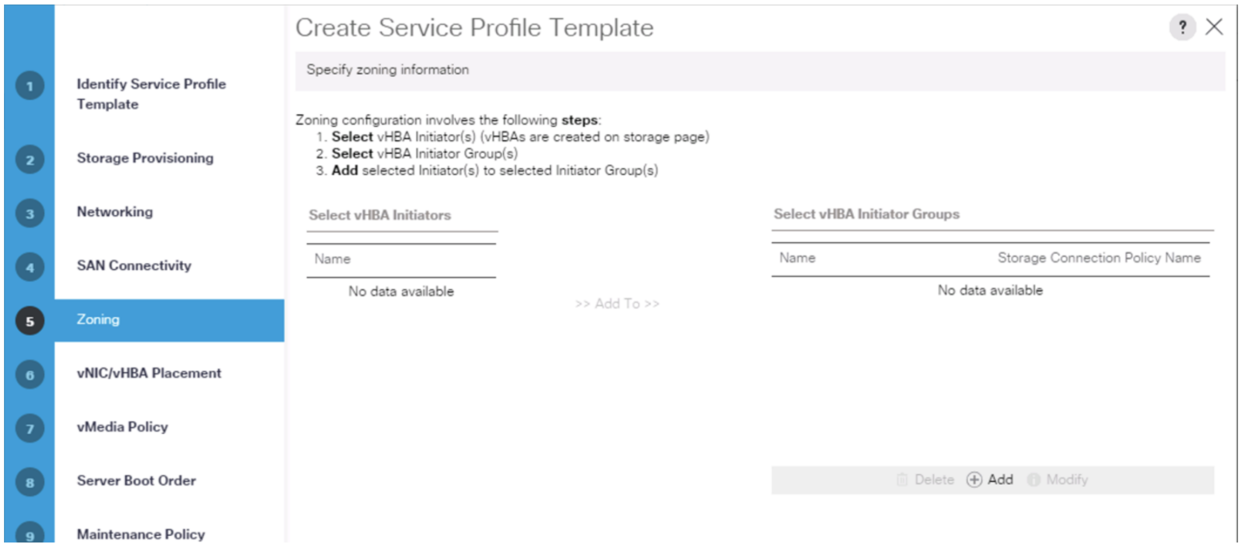
Optionally specify disk policies and SAN configuration information.

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.

9. Click Next on the Zoning tab.



Create Service Profile Template

1 Identify Service Profile Template
2 Storage Provisioning
3 Networking
4 SAN Connectivity
5 Zoning
6 vNIC/vHBA Placement
7 vMedia Policy
8 Server Boot Order
9 Maintenance Policy

Specify zoning information

Zoning configuration involves the following **steps**:
1. **Select** vHBA Initiator(s) (vHBAs are created on storage page)
2. **Select** vHBA Initiator Group(s)
3. **Add** selected Initiator(s) to selected Initiator Group(s)

Select vHBA Initiators

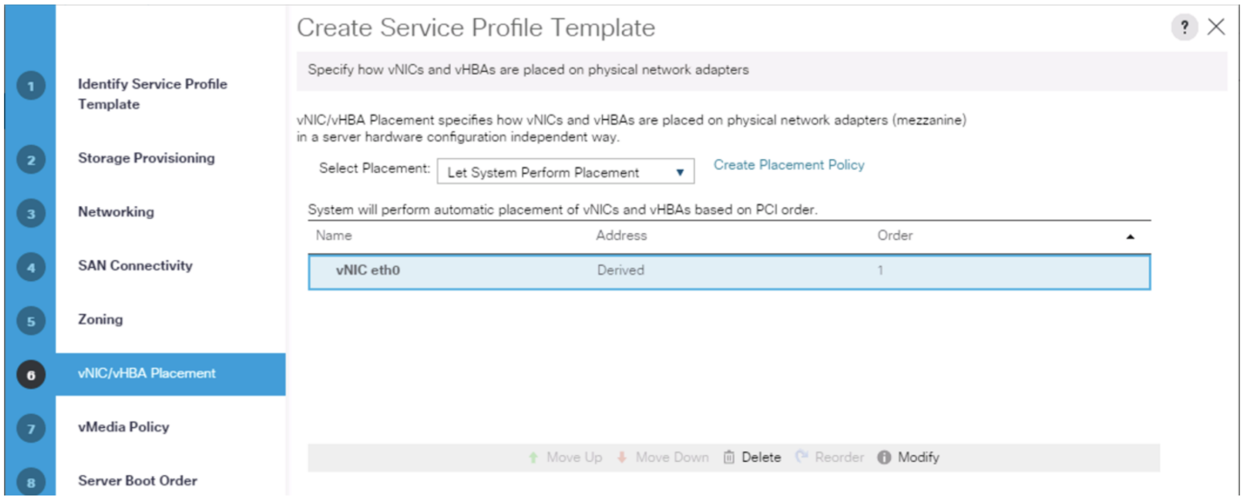
Name
No data available

>> Add To >>

Select vHBA Initiator Groups

Name	Storage Connection Policy Name
No data available	

10. Select Let System Perform Placement for vNIC/vHBA Placement. Click Next.



Create Service Profile Template

1 Identify Service Profile Template
2 Storage Provisioning
3 Networking
4 SAN Connectivity
5 Zoning
6 vNIC/vHBA Placement
7 vMedia Policy
8 Server Boot Order

Specify how vNICs and vHBAs are placed on physical network adapters

vNIC/vHBA Placement specifies how vNICs and vHBAs are placed on physical network adapters (mezzanine) in a server hardware configuration independent way.

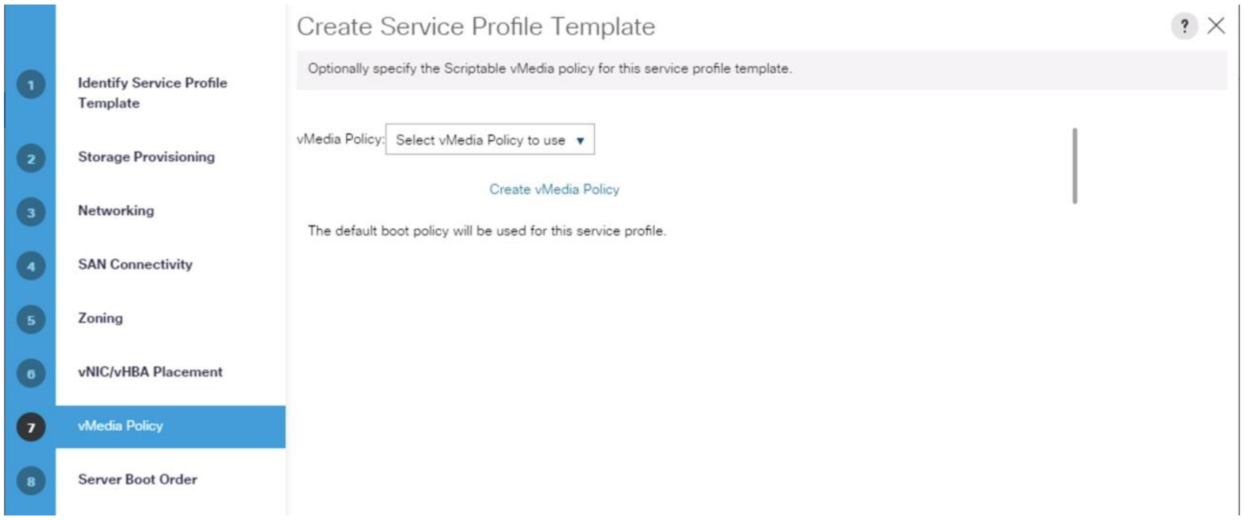
Select Placement: Let System Perform Placement

Create Placement Policy

System will perform automatic placement of vNICs and vHBAs based on PCI order.

Name	Address	Order
vNIC eth0	Derived	1

11. Click Next on the vMedia Policy tab.

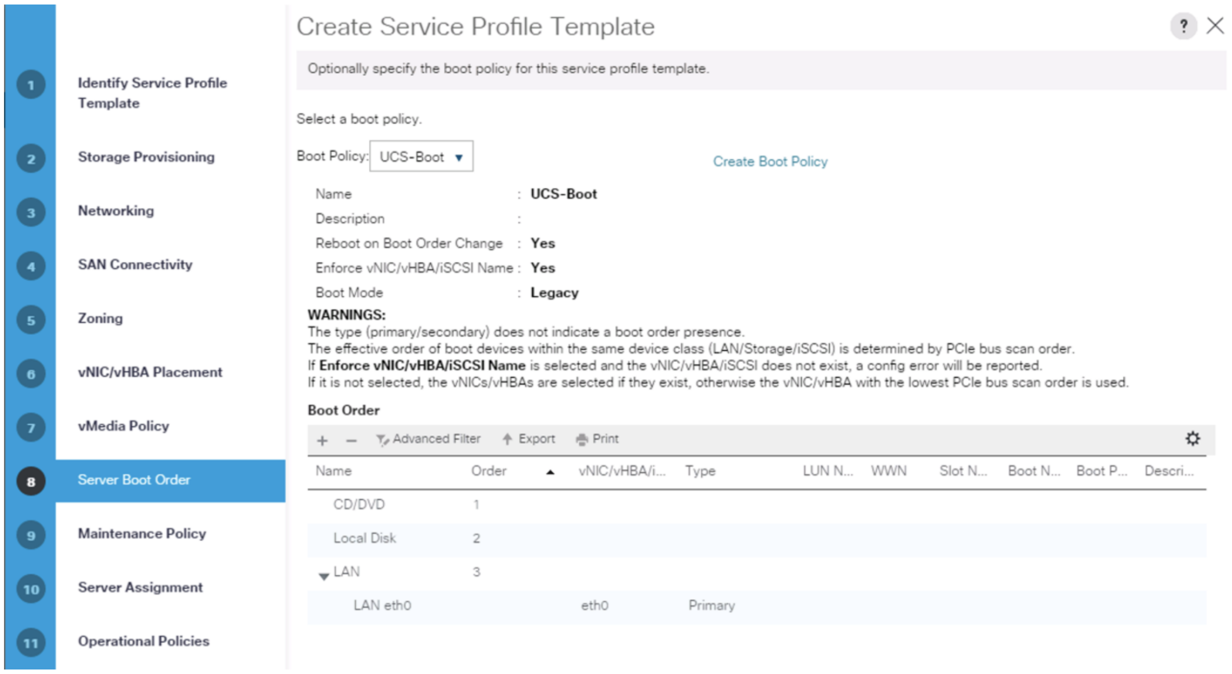


Optionally specify the Scriptable vMedia policy for this service profile template.

vMedia Policy: Select vMedia Policy to use

The default boot policy will be used for this service profile.

12. Select Boot Policy in the Server Boot Order tab.



Optionally specify the boot policy for this service profile template.

Select a boot policy.

Boot Policy: UCS-Boot

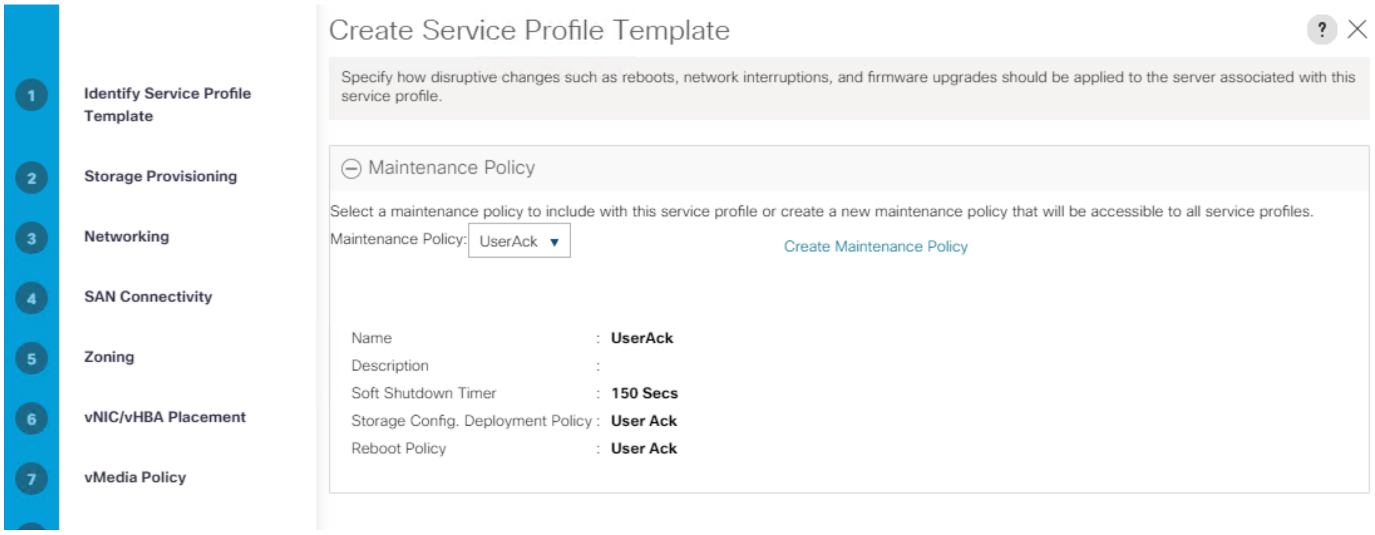
Name	UCS-Boot
Description	
Reboot on Boot Order Change	Yes
Enforce vNIC/vHBA/iSCSI Name	Yes
Boot Mode	Legacy

WARNINGS:
The type (primary/secondary) does not indicate a boot order presence.
The effective order of boot devices within the same device class (LAN/Storage/iSCSI) is determined by PCIe bus scan order.
If **Enforce vNIC/vHBA/iSCSI Name** is selected and the vNIC/vHBA/iSCSI does not exist, a config error will be reported.
If it is not selected, the vNICs/vHBAs are selected if they exist, otherwise the vNIC/vHBA with the lowest PCIe bus scan order is used.

Boot Order

Name	Order	vNIC/vHBA/i...	Type	LUN N...	WWN	Slot N...	Boot N...	Boot P...	Descri...
CD/DVD	1								
Local Disk	2								
LAN	3								
LAN eth0		eth0	Primary						

13. Select UserAck maintenance policy, which requires user acknowledgement prior rebooting server when making changes to policy or pool configuration tied to a service profile.



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

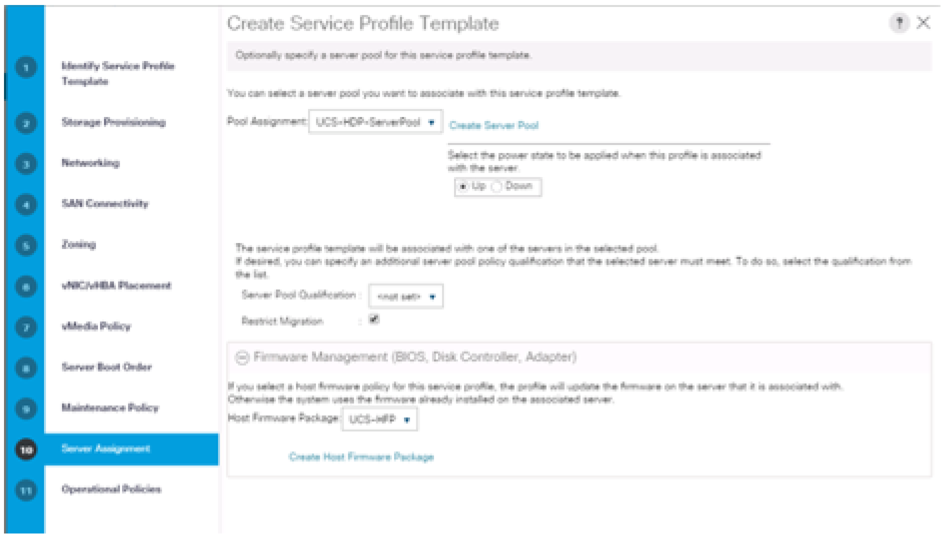
Select a maintenance policy to include with this service profile or create a new maintenance policy that will be accessible to all service profiles.

Maintenance Policy: UserAck

Name	UserAck
Description	
Soft Shutdown Timer	150 Secs
Storage Config. Deployment Policy	User Ack
Reboot Policy	User Ack

14. Select the Server Pool policy to automatically assign a service profile to a server that meets the requirements for server qualification based on the pool configuration. Select Power state when the Service Profile is associated to server

15. On the same page you can configure "Host firmware Package Policy" which helps to keep the firmware in sync when associated to server.



Optionally specify a server pool for this service profile template.

You can select a server pool you want to associate with this service profile template.

Pool Assignment: UCS-HDP-ServerPool

Select the power state to be applied when this profile is associated with the server.

[✓ Up] Down

The service profile template will be associated with one of the servers in the selected pool.
If desired, you can specify an additional server pool policy qualification that the selected server must meet. To do so, select the qualification from the list.

Server Pool Qualification : <not set>

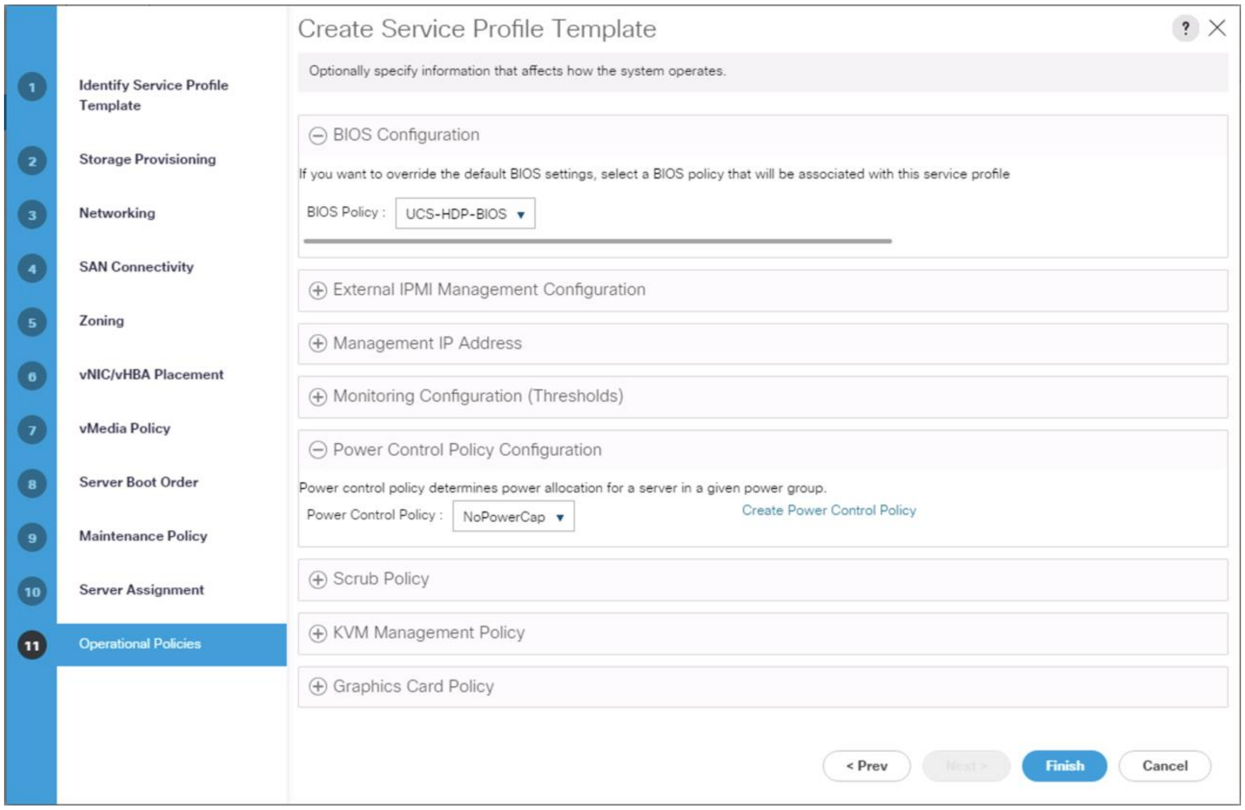
Restrict Migration : ✓

Firmware Management (BIOS, Disk Controller, Adapter)

If you select a host firmware policy for this service profile, the profile will update the firmware on the server that it is associated with.
Otherwise the system uses the firmware already installed on the associated server.

Host Firmware Package: UCS-HFP

On the Operational Policy page, we configured the BIOS policy for a Cisco UCS C240 M5 Rack server with the Power Control Policy set to "NoPowerCap" for maximum performance.



Create Service Profile Template

Optionally specify information that affects how the system operates.

BIOS Configuration

If you want to override the default BIOS settings, select a BIOS policy that will be associated with this service profile

BIOS Policy : UCS-HDP-BIOS

Power Control Policy Configuration

Power control policy determines power allocation for a server in a given power group.

Power Control Policy : NoPowerCap

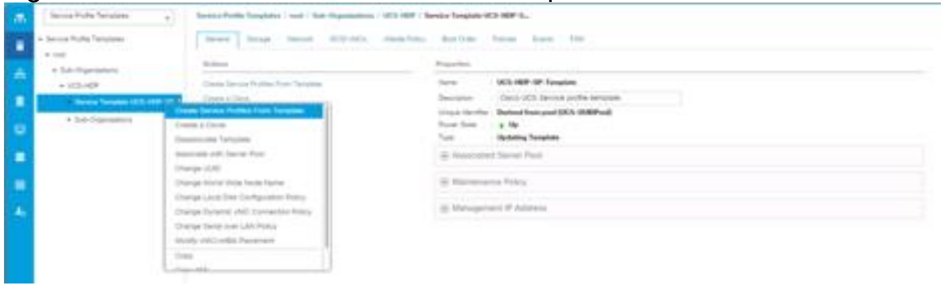
16. Click Finish to create the Service Profile template.

Create Service Profiles from Template

To create a Service Profile from a template, follow these steps:

1. Right-click the Service Profile Template and select Create Service profile from Template.

Figure 56 Create Service Profile from Template


Service Profile Templates
root
Sub-Organizations
UCS-HDP
Create Service Profiles From Template
Create a Clone
Disassociate Template
Associate with Server Pool
Change UUID
Change World Wide Node Name
Change Local Disk Configuration Policy
Change Dynamic vNIC Connection Policy
Change Serial over LAN Policy
Modify vNIC/vHBA Placement
Copy
Associated Server Pool
Maintenance Policy
Management IP Address

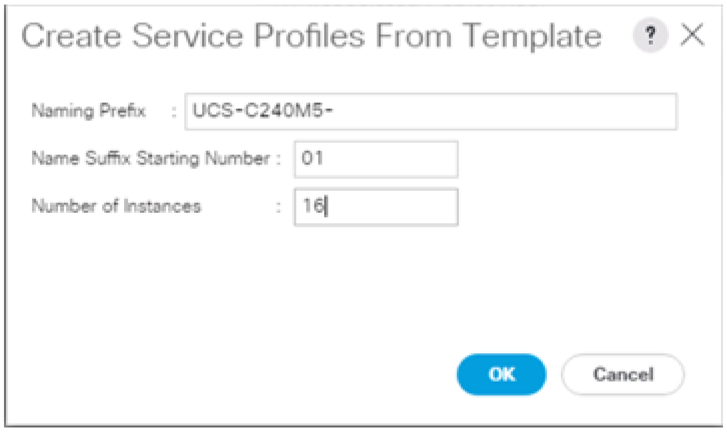
Name : UCS-HDP-SP-Template

Description : Cisco UCS Service profile template

Unique Identifier : Derived from pool (UCS-UUIDPool)

Power State : Up

Type : Updating Template



Naming Prefix : UCS-C240M5-
Name Suffix Starting Number : 01
Number of Instances : 16

The Service profile will automatically assign to servers discovered and meets the requirement of Server Pool.

2. Repeat the steps above to create service profile template(s) and service profile(s) for Cisco UCS S3260, Cisco C240 M5 according to different deployment scenario.

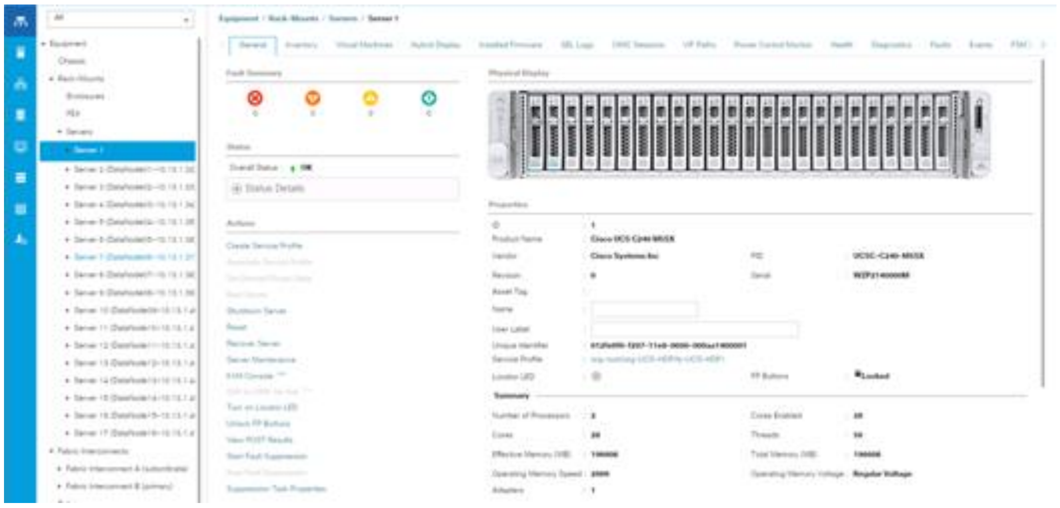
Install Red Hat Enterprise Linux 7.6

This section provides detailed procedures for installing Red Hat Enterprise Linux Server using Software RAID (OS based Mirroring) on Cisco UCS C240 M5 servers. There are multiple ways to install the RHEL operating system. The installation procedure described in this deployment guide uses KVM console and virtual media from Cisco UCS Manager.

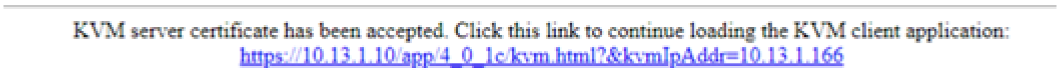
In this study RHEL version 7.6 DVD/ISO was utilized for OS the installation on Cisco UCS C240 M5 Rack Servers.

To install the Red Hat Enterprise Linux 7.6 operating system, follow these steps:

1. Log into the Cisco UCS Manager.
2. Select the Equipment tab.
3. In the navigation pane expand Rack-Mounts and then Servers.
4. Right-click the server and select KVM console.
5. In the right pane, click the KVM Console >>.

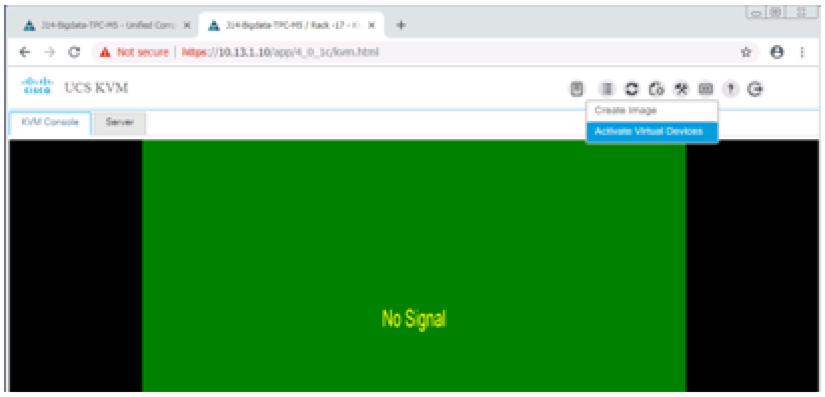


6. Click the link to launch the KVM console.

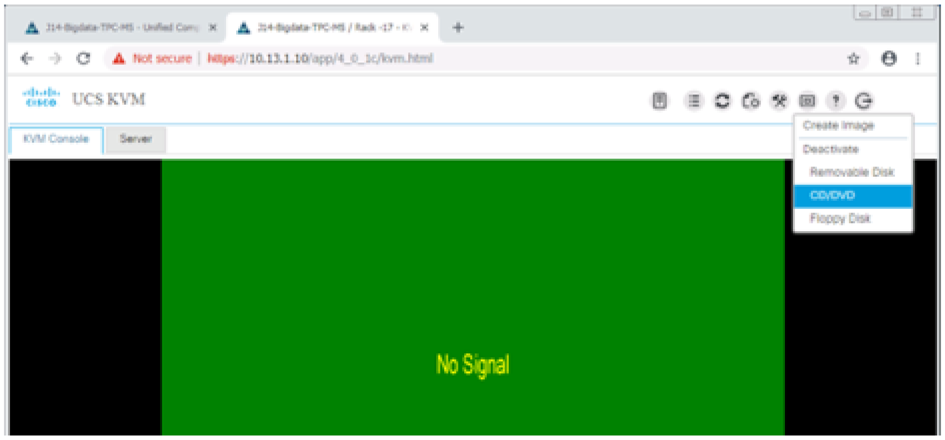

KVM server certificate has been accepted. Click this link to continue loading the KVM client application: https://10.13.1.10/app/4_0_1c/kvm.html?&kvmIpAddr=10.13.1.166

7. Point the cursor over the top right corner and select the Virtual Media tab.

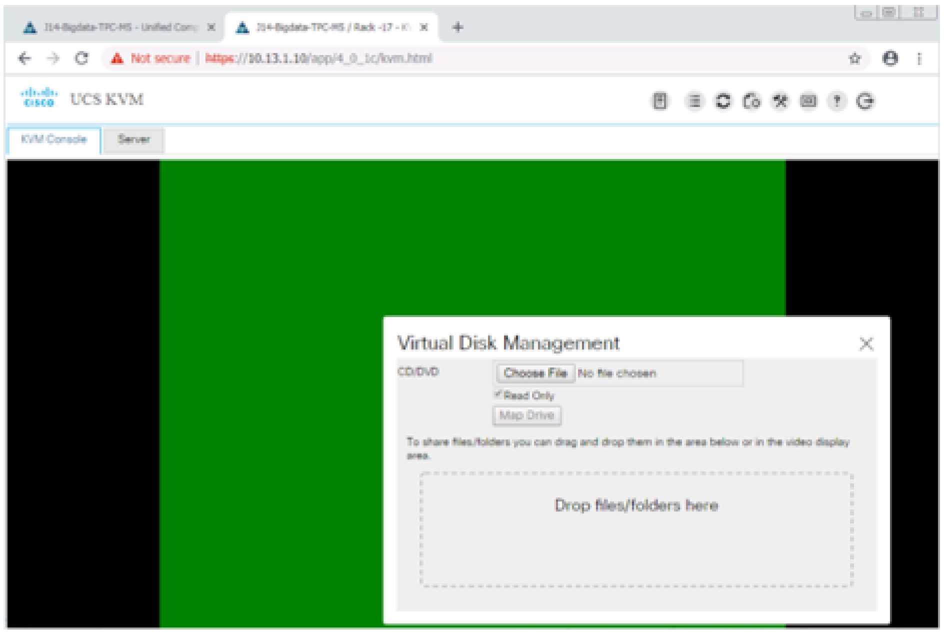
8. Click the Activate Virtual Devices found in Virtual Media tab.


Not secure | https://10.13.1.10/app/4_0_1c/kvm.html

9. Click the Virtual Media tab to select CD/DVD.

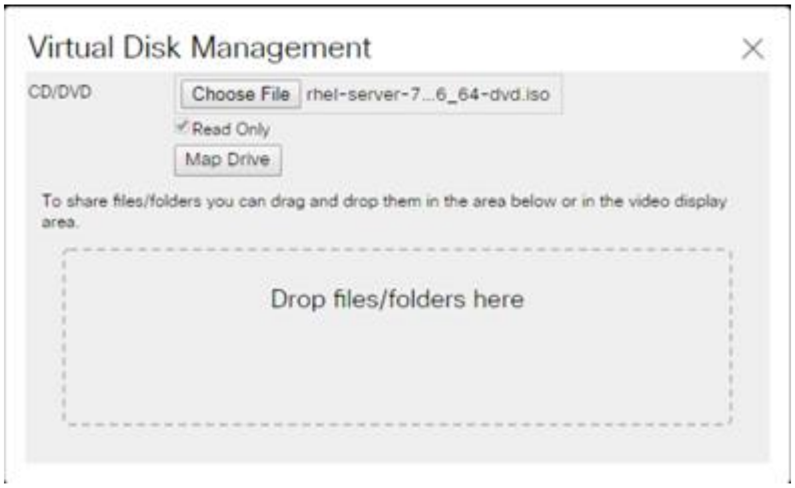


10. Select Map Drive in the Virtual Disk Management windows.



11. Browse to the Red Hat Enterprise Linux 7.6 installer ISO image file.

The Red Hat Enterprise Linux 7.6 Server DVD is assumed to be on the client machine.



Virtual Disk Management

CD/DVD: Choose File rhel-server-7...6_64-dvd.iso

☑ Read Only

Map Drive

To share files/folders you can drag and drop them in the area below or in the video display area.

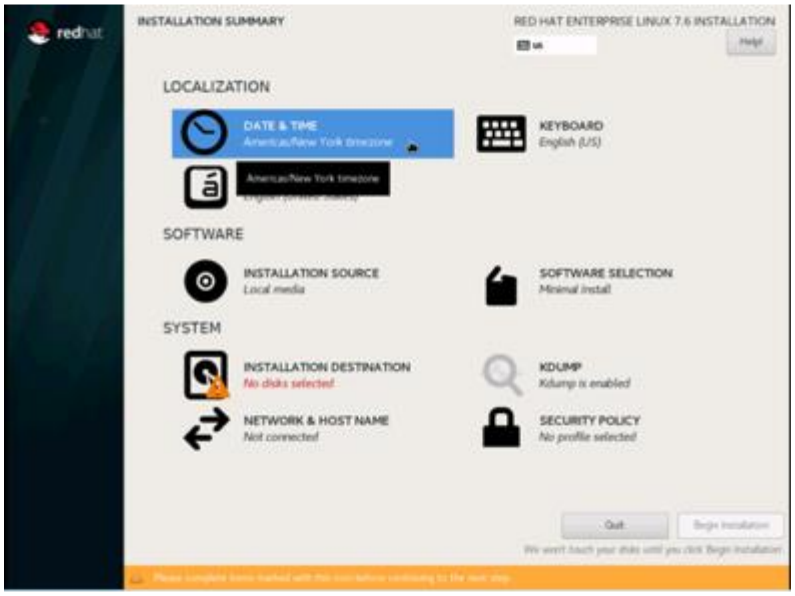
Drop files/folders here

12. Click Open to add the image to the list of virtual media.

13. Select the Installation option from Red Hat Enterprise Linux 7.6.

14. Select the language for the installation and click Continue.

15. Select date and time, which pops up another window as shown below.



INSTALLATION SUMMARY

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

LOCALIZATION

DATE & TIME: Americas/New York timezone

KEYBOARD: English (US)

SOFTWARE

INSTALLATION SOURCE: Local media

SOFTWARE SELECTION: Minimal install

SYSTEM

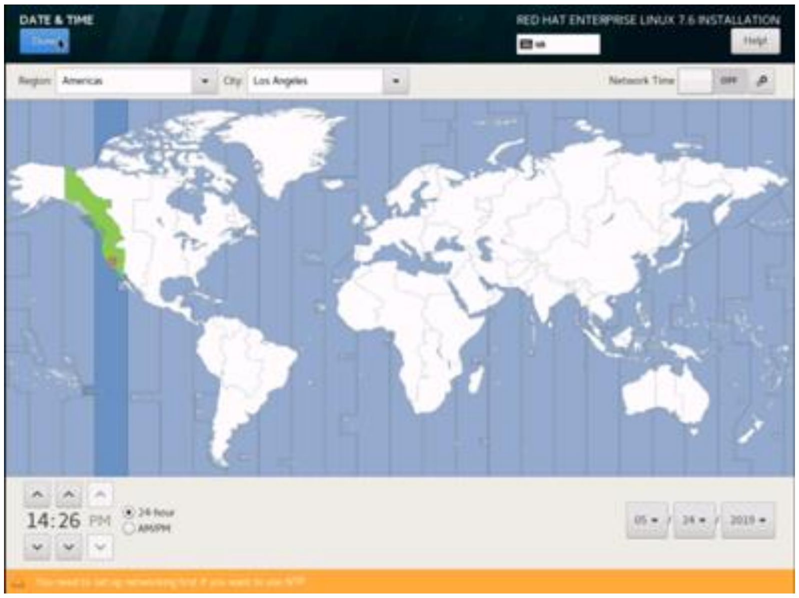
INSTALLATION DESTINATION: No disks selected

KDUMP: Kdump is enabled

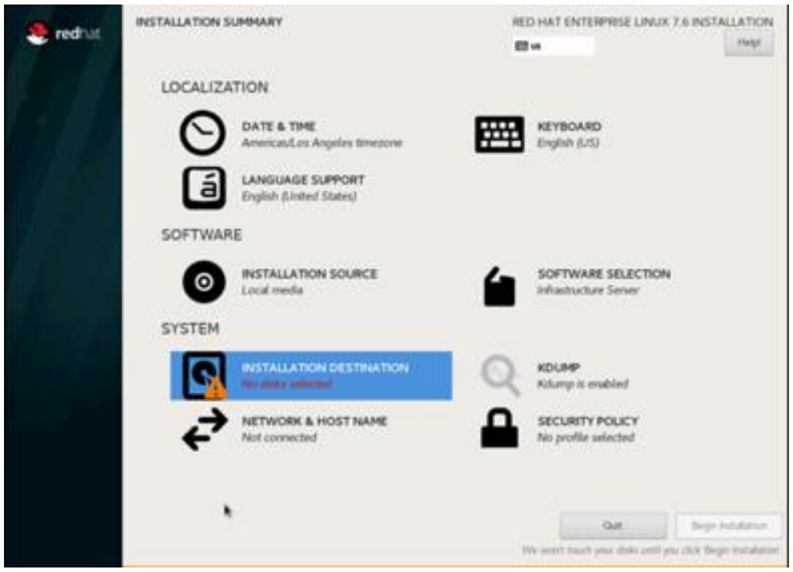
NETWORK & HOST NAME: Not connected

SECURITY POLICY: No profile selected

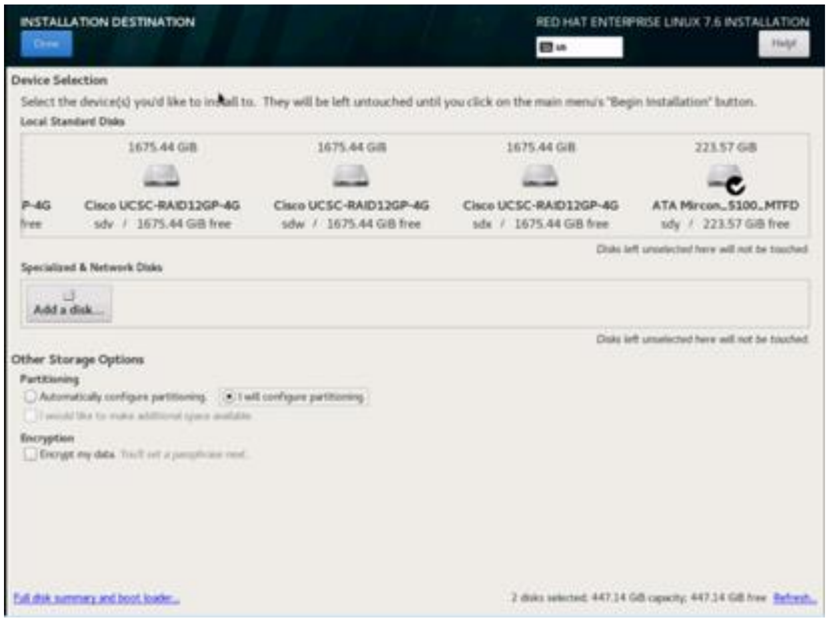
16. Select the location on the map, set the time, and click Done.



17. Click Installation Destination.



18. This opens a new window with the boot disks. Make the selection and choose “I will configure partitioning”. Click Done. We selected two M.2 SATA SSDs.



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

1675.44 GiB	1675.44 GiB	1675.44 GiB	223.57 GiB
Cisco UCSC-RAID12GP-4G	Cisco UCSC-RAID12GP-4G	Cisco UCSC-RAID12GP-4G	ATA Mircon_5100_MTFD
sdv / 1675.44 GiB free	sdw / 1675.44 GiB free	sdx / 1675.44 GiB free	sdy / 223.57 GiB free

Disks left unselected here will not be touched

Specialized & Network Disks

Add a disk...

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

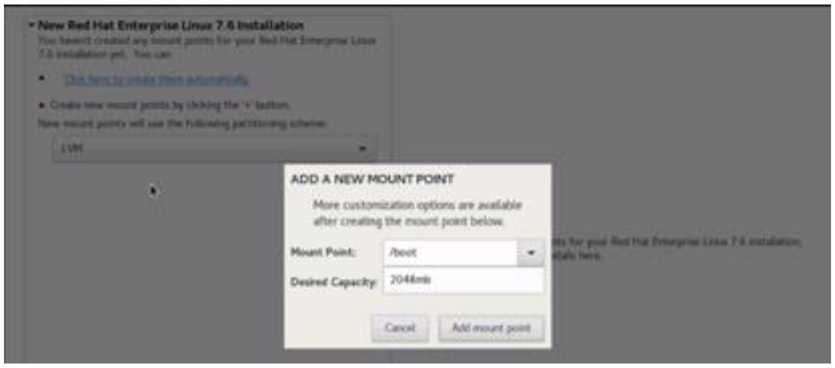
Encrypt my data. You'll set a passphrase next.

Full disk summary and boot loader...

2 disks selected; 447.14 GB capacity; 447.14 GB free Refresh...

19. This opens a window to create the partitions. Click the + sign to add a new partition as shown below with a boot partition size 2048 MB.

20. Click Add Mount Point to add the partition.



New Red Hat Enterprise Linux 7.6 Installation

You haven't created any mount points for your Red Hat Enterprise Linux 7.6 installation yet. You can:

- Click here to create them automatically.
- Create new mount points by clicking the '+' button.

New mount points will use the following partitioning scheme:

LVM

ADD A NEW MOUNT POINT

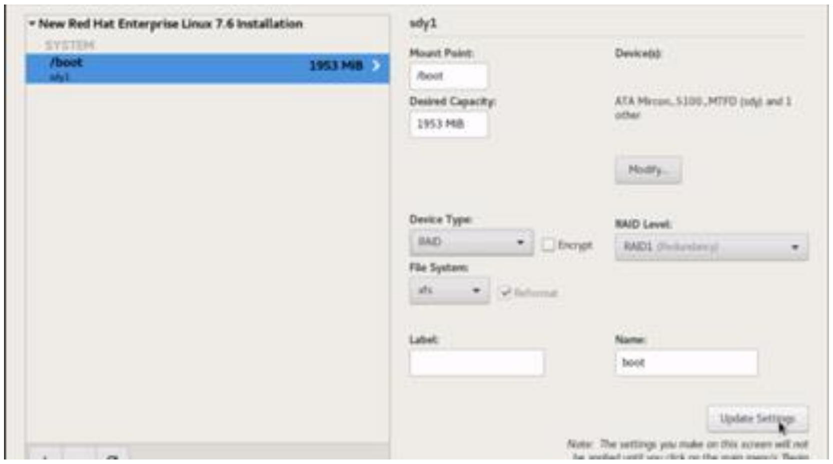
More customization options are available after creating the mount point below.

Mount Point: /boot

Desired Capacity: 2048mb

Cancel | Add mount point

21. Change the device type to RAID and make sure the RAID level is RAID1 (redundancy) and click Update Settings to save the changes.



New Red Hat Enterprise Linux 7.6 Installation

SYSTEM

/boot 1953 MiB

sdy1

sdy1

Mount Point: /boot

Desired Capacity: 1953 MiB

Device(s): ATA Mircon_5100_MTFD (sdy) and 1 other

Modify...

Device Type: RAID

Encrypt

RAID Level: RAID1 (Redundancy)

File System: xfs

Reformat

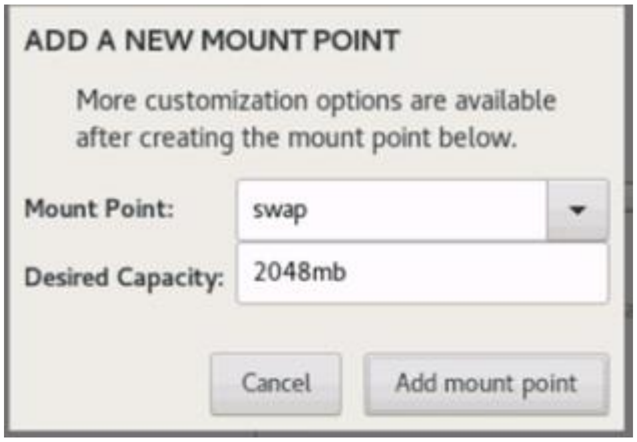
Label:

Name: boot

Update Settings

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin

22. Click the + sign to create the swap partition of size 2048 MB. Click Add Mount Point.



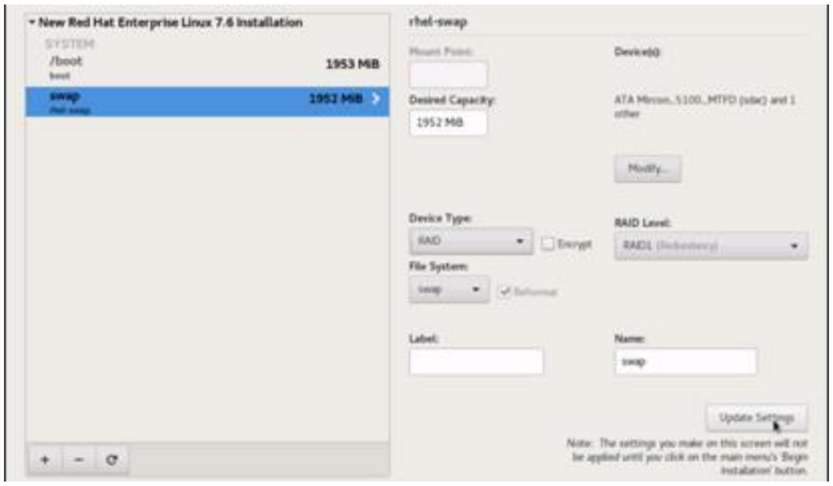
ADD A NEW MOUNT POINT

More customization options are available after creating the mount point below.

Mount Point: swap

Desired Capacity: 2048mb

23. Change the Device type to RAID and RAID level to RAID1 (Redundancy) and click Update Settings.



New Red Hat Enterprise Linux 7.6 Installation

SYSTEM

/boot 1953 MiB

swap 1952 MiB

rhel-swap

Mount Point:

Desired Capacity: 1952 MiB

Device(s): ATA Micron_5100_MTFD (sdac) and 1 other

Device Type: RAID

RAID Level: RAID1 (Redundancy)

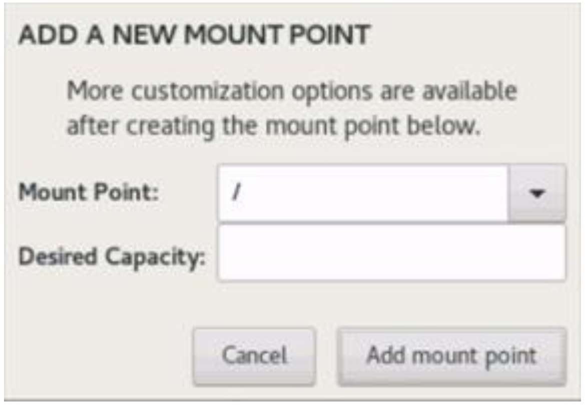
File System: swap

Label:

Name: swap

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

24. Click + to add the / partition. The size can be left empty so it will use the remaining capacity. Click Add Mountpoint.



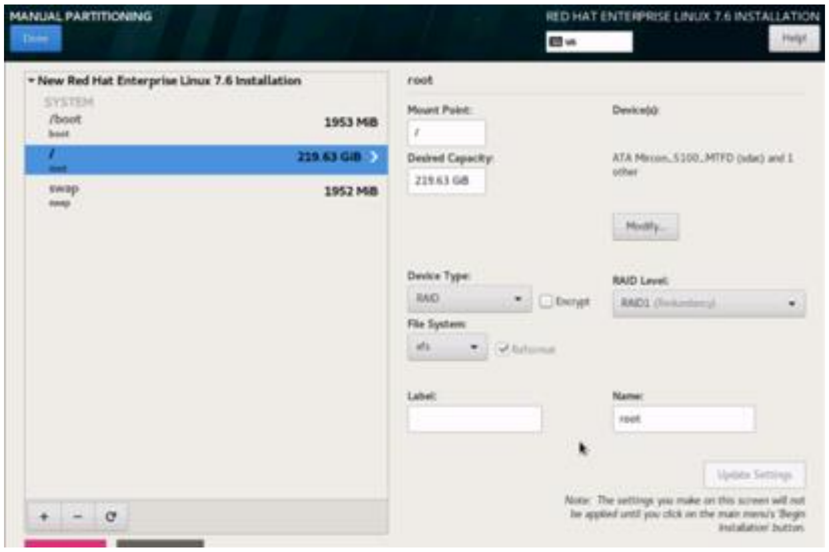
ADD A NEW MOUNT POINT

More customization options are available after creating the mount point below.

Mount Point: /

Desired Capacity:

25. Change the Device type to RAID and RAID level to RAID1 (Redundancy). Click Update Settings.


MANUAL PARTITIONING

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

New Red Hat Enterprise Linux 7.6 Installation

SYSTEM

/boot 1953 MiB
boot

/ 219.63 GiB
root

swap 1952 MiB
swap

root

Mount Point: /

Desired Capacity: 219.63 GiB

Device(s): ATA Micron_5100_MTFD (sdac) and 1 other

Device Type: RAID

File System: xfs

RAID Level: RAID1 (Redundancy)

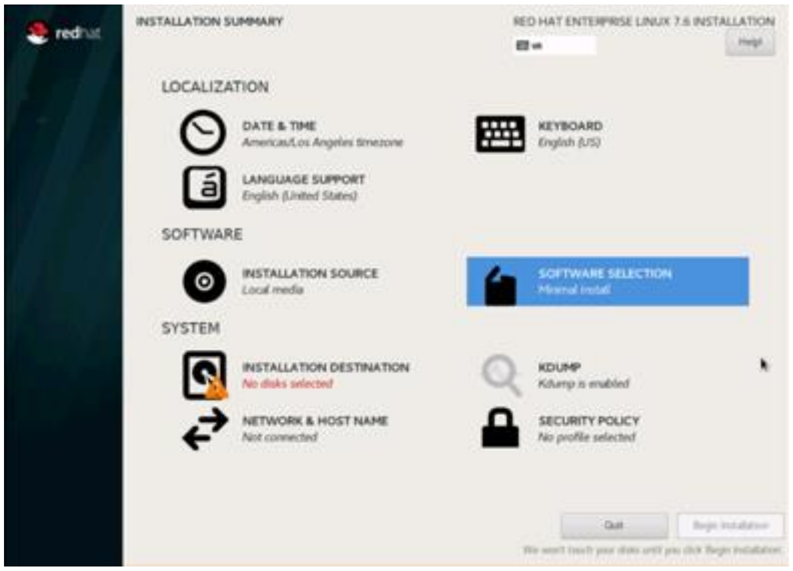
Label:

Name: root

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

26. Click Done to go back to the main screen and continue the Installation.

27. Click Software Selection.


INSTALLATION SUMMARY

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

LOCALIZATION

DATE & TIME
Americas/Los Angeles timezone

KEYBOARD
English (US)

LANGUAGE SUPPORT
English (United States)

SOFTWARE

INSTALLATION SOURCE
Local media

SOFTWARE SELECTION
Minimal Install

SYSTEM

INSTALLATION DESTINATION
No disks selected

KDUMP
Kdump is enabled

NETWORK & HOST NAME
Not connected

SECURITY POLICY
No profile selected

We won't touch your disks until you click 'Begin Installation'.

28. Select Infrastructure Server and select the Add-Ons as noted below then click Done:

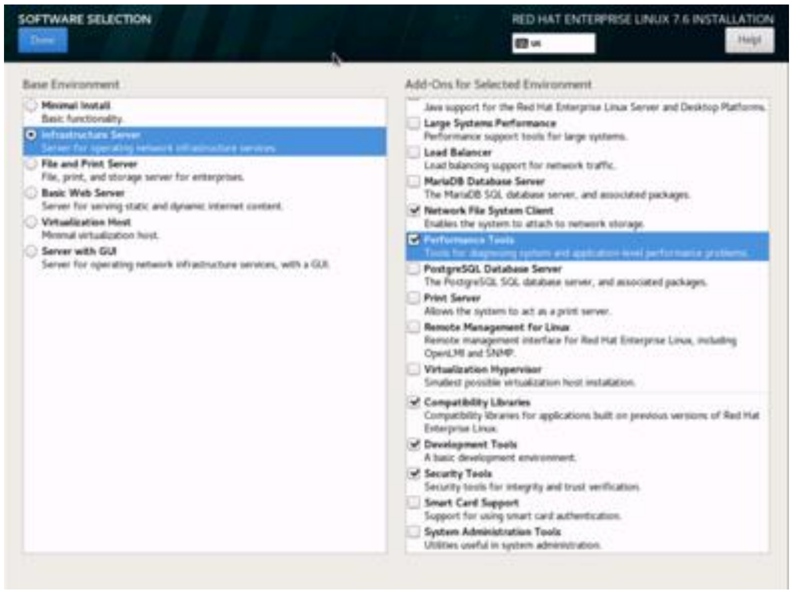
    a. Network File System Client

    b. Performance Tools

    c. Compatibility Libraries

    d. Development Tools

    e. Security Tools



SOFTWARE SELECTION

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

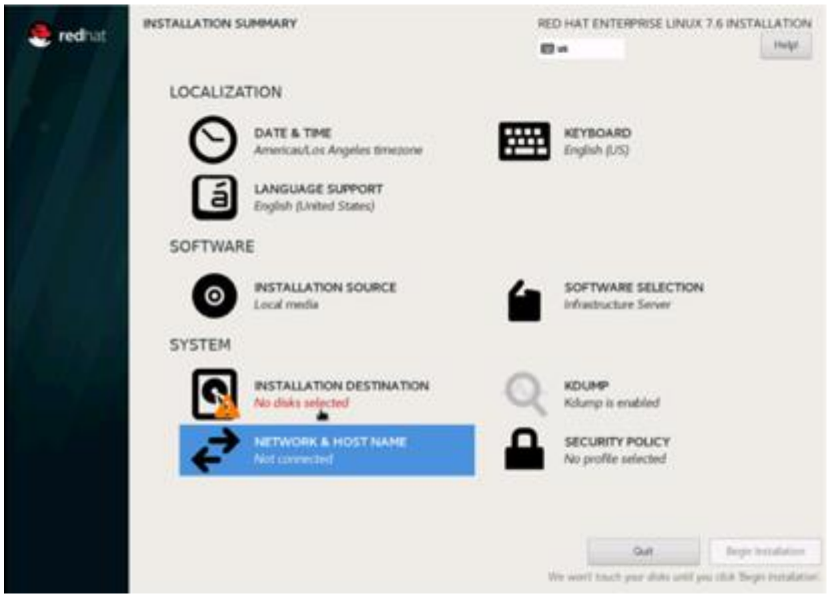
Base Environment

- ( ) **Minimal Install** — Basic functionality.
- (●) **Infrastructure Server** — Server for operating network infrastructure services.
- ( ) **File and Print Server** — File, print, and storage server for enterprises.
- ( ) **Basic Web Server** — Server for serving static and dynamic internet content.
- ( ) **Virtualization Host** — Minimal virtualization host.
- ( ) **Server with GUI** — Server for operating network infrastructure services, with a GUI.

Add-Ons for Selected Environment

- Java support for the Red Hat Enterprise Linux Server and Desktop Platforms.
- [ ] **Large Systems Performance** — Performance support tools for large systems.
- [ ] **Load Balancer** — Load balancing support for network traffic.
- [ ] **MariaDB Database Server** — The MariaDB SQL database server, and associated packages.
- [✓] **Network File System Client** — Enables the system to attach to network storage.
- [✓] **Performance Tools** — Tools for diagnosing system and application-level performance problems.
- [ ] **PostgreSQL Database Server** — The PostgreSQL SQL database server, and associated packages.
- [ ] **Print Server** — Allows the system to act as a print server.
- [ ] **Remote Management for Linux** — Remote management interface for Red Hat Enterprise Linux, including OpenLMI and SNMP.
- [ ] **Virtualization Hypervisor** — Smallest possible virtualization host installation.
- [✓] **Compatibility Libraries** — Compatibility libraries for applications built on previous versions of Red Hat Enterprise Linux.
- [✓] **Development Tools** — A basic development environment.
- [✓] **Security Tools** — Security tools for integrity and trust verification.
- [ ] **Smart Card Support** — Support for using smart card authentication.
- [ ] **System Administration Tools** — Utilities useful in system administration.

29. Click Network and Hostname and configure Hostname and Networking for the Host.



INSTALLATION SUMMARY

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

LOCALIZATION

- DATE & TIME — America/Los Angeles timezone
- KEYBOARD — English (US)
- LANGUAGE SUPPORT — English (United States)

SOFTWARE

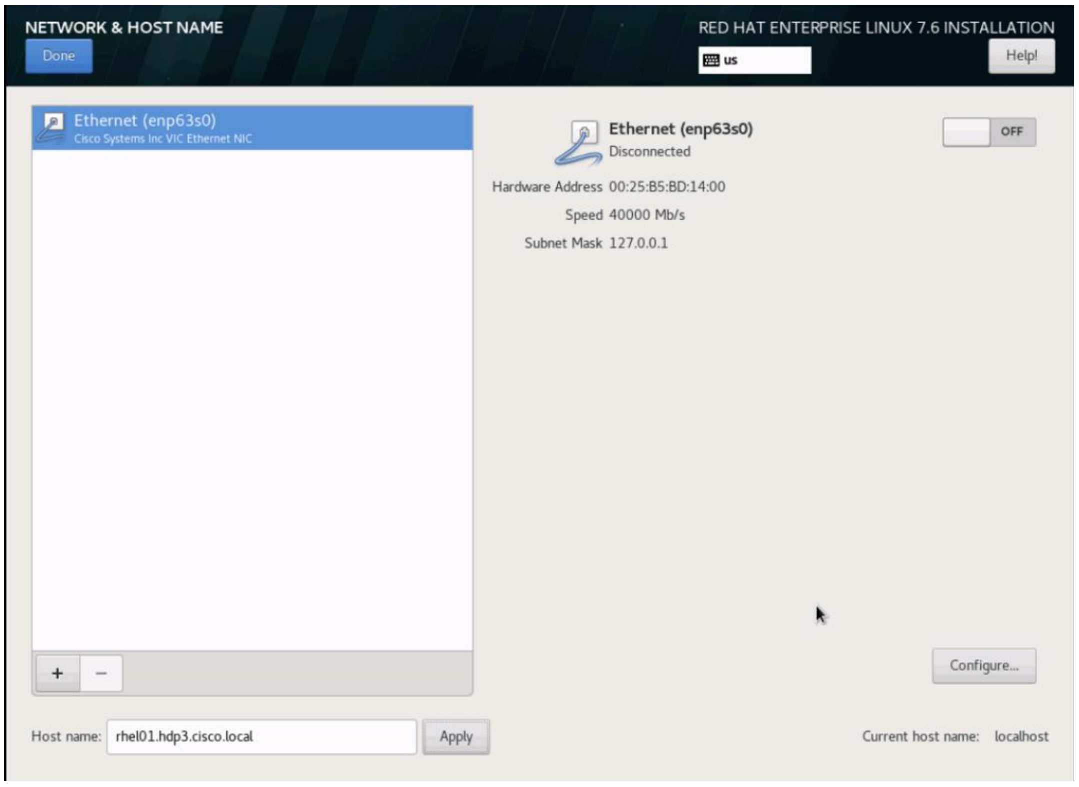
- INSTALLATION SOURCE — Local media
- SOFTWARE SELECTION — Infrastructure Server

SYSTEM

- INSTALLATION DESTINATION — No disks selected
- KDUMP — Kdump is enabled
- NETWORK & HOST NAME — Not connected
- SECURITY POLICY — No profile selected

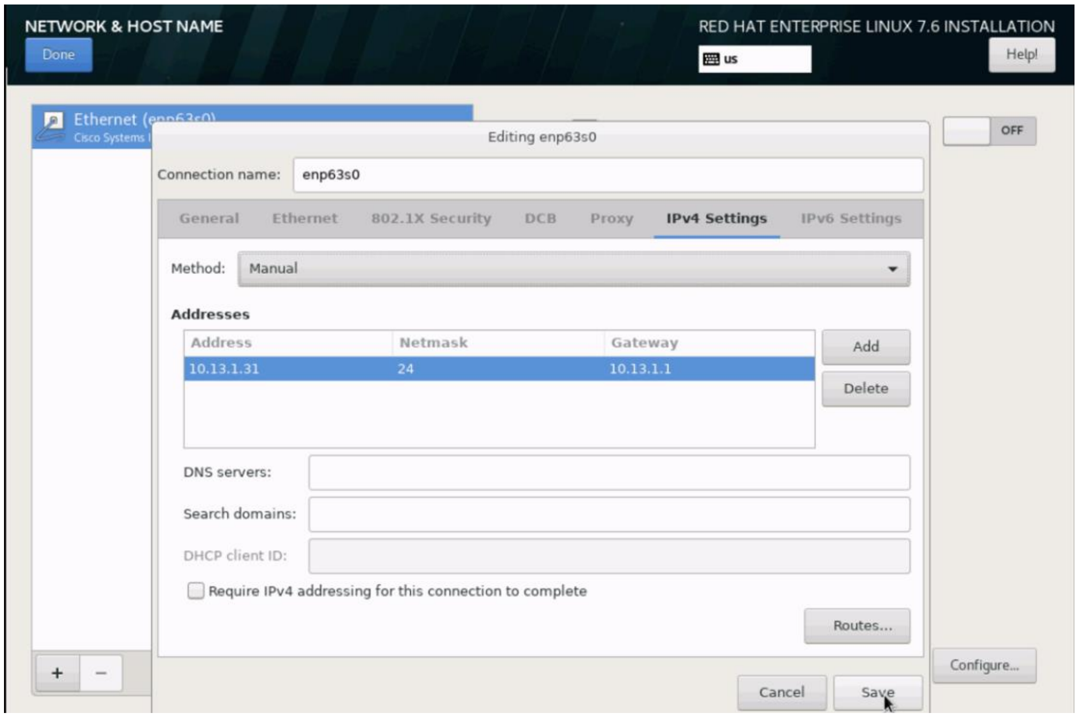
We won't touch your disks until you click 'Begin Installation'.

30. Type in the hostname as shown below.


NETWORK & HOST NAME
Done
RED HAT ENTERPRISE LINUX 7.6 INSTALLATION
us
Help!
Ethernet (enp63s0)
Cisco Systems Inc VIC Ethernet NIC
Ethernet (enp63s0)
Disconnected
OFF
Hardware Address 00:25:B5:BD:14:00
Speed 40000 Mb/s
Subnet Mask 127.0.0.1
Configure...
Host name: rhel01.hdp3.cisco.local
Apply
Current host name: localhost

31. Click Configure to open the Network Connectivity window. Click IPv4 Settings.

32. Change the Method to Manual and click Add to enter the IP Address, Netmask and Gateway details.



Connection name: enp63s0

Method: Manual

Addresses

Address	Netmask	Gateway
10.13.1.31	24	10.13.1.1

DNS servers:

Search domains:

DHCP client ID:

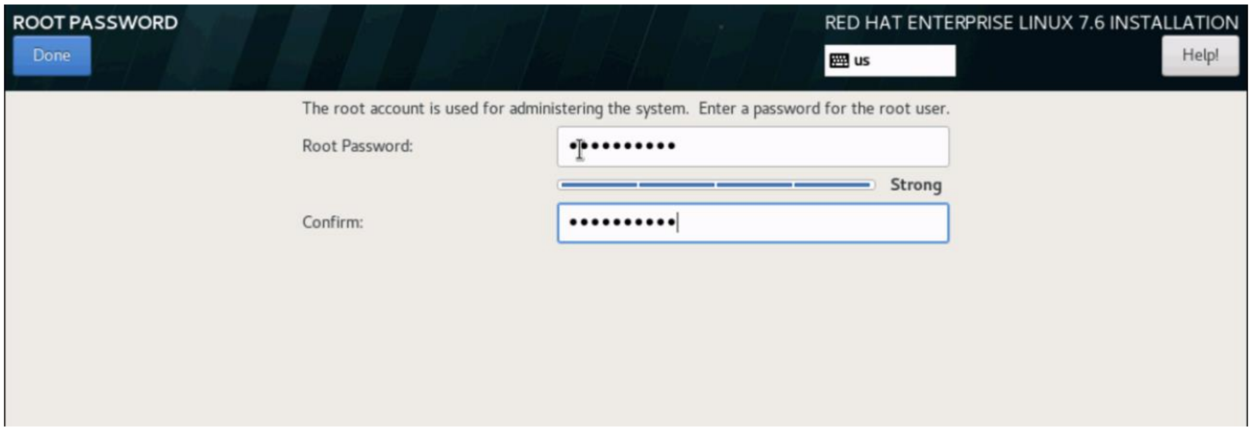
☐ Require IPv4 addressing for this connection to complete

33. Click Save, update the hostname, and turn Ethernet ON. Click Done to return to the main menu.

34. Click Begin Installation in the main menu.

35. Select Root Password in the User Settings.

36. Enter the Root Password and click Done.



The root account is used for administering the system. Enter a password for the root user.

Root Password: ••••••••••

Strong

Confirm: •••••••••••

37. Once the installation is complete reboot the system.

38. Repeat steps 1 to 37 to install Red Hat Enterprise Linux 7.6 on Servers 2 through 30.

The OS installation and configuration of the nodes that is mentioned above can be automated through PXE boot or third-party tools.

The hostnames and their corresponding IP addresses are shown in Table 7 .

Table 7 Hostname and IP address

Hostname	Eth0
rhel01	10.13.1.31
rhel02	10.13.1.32
rhel03	10.13.1.33
rhel04	10.13.1.34
rhel05	10.13.1.35
…..	…..
Rhel29	10.13.1.59
Rhel30	10.13.1.60

Multi-homing configuration is not recommended in this design, so please assign only one network interface on each host.

For simplicity, outbound NATing is configured for internet access when desired, such as accessing public repos and/or accessing Red Hat Content Delivery Network. However, configuring outbound NAT is beyond the scope of this document.

Post OS Install Configuration

Choose one of the nodes of the cluster or a separate node as the Admin Node for management, such as CDH installation, Ansible, creating a local Red Hat repo, and others. In this document, we used rhel01 for this purpose.

Configure /etc/hosts

Setup `/etc/hosts` on the Admin node; this is a pre-configuration to setup DNS as shown in the next section.

For the purpose of simplicity, /etc/hosts file is configured with hostnames in all the nodes. However, in large scale production grade deployment, DNS server setup is highly recommended. Furthermore, /etc/hosts file is not copied into containers running on the platform.

Below are the sample A records for DNS configuration within Linux environment.

```
ORIGIN hdp3.cisco.local
rhel01   A 10.13.1.31
rhel02   A 10.13.1.32
rhel03   A 10.13.1.33
…
…
rhel29  A 10.13.1.59
rhel30  A 10.13.1.60
```

To create the host file on the admin node, follow these steps:

1. Login to the Admin Node (rhel01).

   ```
   #ssh 10.13.1.31
   ```

2. Populate the host file with IP addresses and corresponding hostnames on the Admin node (rhel01) and other nodes as follows:

3. On Admin Node (rhel01):

```
[root@rhel01 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.13.1.31 rhel01 rhel01.hdp3.cisco.local
10.13.1.32 rhel02 rhel02.hdp3.cisco.local
10.13.1.33 rhel03 rhel03.hdp3.cisco.local
10.13.1.34 rhel04 rhel04.hdp3.cisco.local
10.13.1.35 rhel05 rhel05.hdp3.cisco.local
10.13.1.36 rhel06 rhel06.hdp3.cisco.local
10.13.1.37 rhel07 rhel07.hdp3.cisco.local
10.13.1.38 rhel08 rhel08.hdp3.cisco.local
10.13.1.39 rhel09 rhel09.hdp3.cisco.local
10.13.1.40 rhel10 rhel10.hdp3.cisco.local
10.13.1.41 rhel11 rhel11.hdp3.cisco.local
```

```
10.13.1.42 rhel12 rhel12.hdp3.cisco.local
10.13.1.43 rhel13 rhel13.hdp3.cisco.local
10.13.1.44 rhel14 rhel14.hdp3.cisco.local
10.13.1.45 rhel15 rhel15.hdp3.cisco.local
10.13.1.46 rhel16 rhel16.hdp3.cisco.local
10.13.1.47 rhel17 rhel17.hdp3.cisco.local
10.13.1.48 rhel18 rhel18.hdp3.cisco.local
10.13.1.49 rhel19 rhel19.hdp3.cisco.local
10.13.1.50 rhel20 rhel20.hdp3.cisco.local
10.13.1.51 rhel21 rhel21.hdp3.cisco.local
10.13.1.52 rhel22 rhel22.hdp3.cisco.local
10.13.1.53 rhel23 rhel23.hdp3.cisco.local
10.13.1.54 rhel24 rhel24.hdp3.cisco.local
10.13.1.55 rhel25 rhel25.hdp3.cisco.local
10.13.1.56 rhel26 rhel26.hdp3.cisco.local
10.13.1.57 rhel27 rhel27.hdp3.cisco.local
10.13.1.58 rhel28 rhel28.hdp3.cisco.local
10.13.1.59 rhel29 rhel29.hdp3.cisco.local
10.13.1.60 rhel30 rhel30.hdp3.cisco.local
```

Set Up Passwordless Login

To manage all the nodes in a cluster from the admin node password-less login needs to be setup. It assists in automating common tasks with Ansible, and shell-scripts without having to use passwords.

To enable password-less login across all the nodes when Red Hat Linux is installed across all the nodes in the cluster, follow these steps:

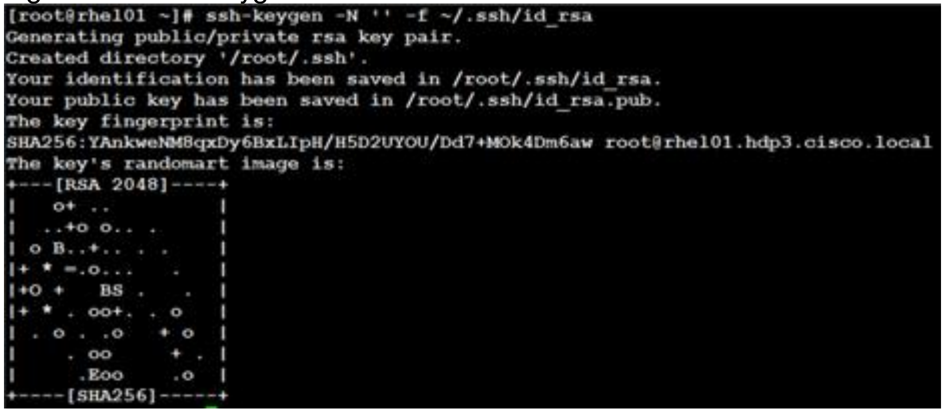
1. Log into the Admin Node (rhel01).

```
#ssh 10.13.1.31
```

2. Run the ssh-keygen command to create both public and private keys on the admin node.

```
# ssh-keygen -N '' -f ~/.ssh/id_rsa
```

Figure 57 Ssh-keygen



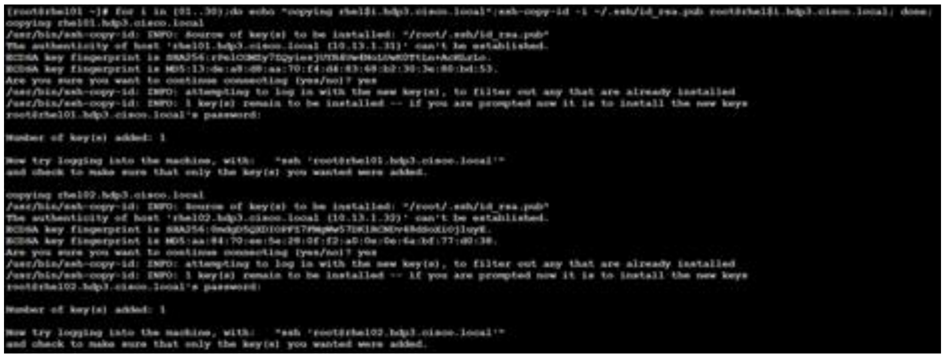
```
[root@rhel01 ~]# ssh-keygen -N '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
Created directory '/root/.ssh'.
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:YAnkweNM8qxDy6BxLIpH/H5D2UYOU/Dd7+MOk4Dm6aw root@rhel01.hdp3.cisco.local
The key's randomart image is:
+---[RSA 2048]----+
|    o+ ..        |
|   ..+o o.. .    |
| o B..+.. . .    |
|+ * =.o...   .   |
|+O +   BS .   .  |
|+ * . oo+. . o   |
| . o . .o   + o  |
|     . oo     + .|
|      .Eoo    .o |
+----[SHA256]-----+
```

3. Run the following command from the admin node to copy the public key id_rsa.pub to all the nodes of the cluster. `ssh-copy-id` appends the keys to the remote-hosts `.ssh/authorized_keys`.

```
# for i in {01..30}; do    echo "copying rhel$i.hdp3.cisco.local"; ssh-copy-id -i
~/.ssh/id_rsa.pub root@rhel$i.hdp3.cisco.local; done;
```

4. Enter yes for Are you sure you want to continue connecting (yes/no)?

5. Enter the password of the remote host.



```
[root@rhel01 ~]# for i in {01..30};do echo "copying rhel$i.hdp3.cisco.local";ssh-copy-id -i ~/.ssh/id_rsa.pub root@rhel$i.hdp3.cisco.local; done;
copying rhel01.hdp3.cisco.local
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'rhel01.hdp3.cisco.local (10.13.1.31)' can't be established.
ECDSA key fingerprint is SHA256:[illegible].
ECDSA key fingerprint is MD5:13:de:a8:d8:aa:70:f4:d4:83:48:b2:30:3e:80:bd:53.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@rhel01.hdp3.cisco.local's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@rhel01.hdp3.cisco.local'"
and check to make sure that only the key(s) you wanted were added.

copying rhel02.hdp3.cisco.local
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'rhel02.hdp3.cisco.local (10.13.1.32)' can't be established.
ECDSA key fingerprint is SHA256:[illegible].
ECDSA key fingerprint is MD5:aa:84:70:ee:5e:28:0f:f2:a0:0e:0e:6a:bf:77:d0:38.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@rhel02.hdp3.cisco.local's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@rhel02.hdp3.cisco.local'"
and check to make sure that only the key(s) you wanted were added.
```

Create a Red Hat Enterprise Linux (RHEL) 7.6 Local Repository

To create a repository using RHEL DVD or ISO on the admin node (in this deployment rhel01 is used for this purpose), create a directory with all the required RPMs, run the “`createrepo`” command and then publish the resulting repository.

To create a RHEL 7.6 local repository, follow these steps:

1. Log on to rhel01. Create a directory that would contain the repository.

```
# mkdir -p /var/www/html/rhelrepo
```

2. Copy the contents of the Red Hat DVD to `/var/www/html/rhelrepo`

3. Alternatively, if you have access to a Red Hat ISO Image, Copy the ISO file to rhel01.

4. Log back into rhel01 and create the mount directory.

```
# scp rhel-server-7.6-x86_64-dvd.iso rhel01:/root/
# mkdir -p /mnt/rheliso
# mount -t iso9660 -o loop /root/rhel-server-7.6-x86_64-dvd.iso /mnt/rheliso/
```

5. Copy the contents of the ISO to the `/var/www/html/rhelrepo` directory.

```
# cp -r /mnt/rheliso/* /var/www/html/rhelrepo
```

6. On rhel01 create a .repo file to enable the use of the yum command.

```
# vi /var/www/html/rhelrepo/rheliso.repo
[rhel7.6]
name=Red Hat Enterprise Linux 7.6
baseurl=http://10.13.1.31/rhelrepo
gpgcheck=0
enabled=1
```

7. Copy rheliso.repo file from /var/www/html/rhelrepo to /etc/yum.repos.d on rhel01.

```
# cp /var/www/html/rhelrepo/rheliso.repo /etc/yum.repos.d/
```

Based on this repository file, yum requires httpd to be running on rhel01 for other nodes to access the repository.

8. To make use of repository files on rhel01 without httpd, edit the baseurl of repo file `/etc/yum.repos.d/rheliso.repo` to point repository location in the file system.

This step is needed to install software on Admin Node (rhel01) using the repo (such as httpd, create-repo, and so on.)

```
# vi /etc/yum.repos.d/rheliso.repo
[rhel7.6]
name=Red Hat Enterprise Linux 7.6
baseurl=file:///var/www/html/rhelrepo
gpgcheck=0
enabled=1
```

Create the Red Hat Repository Database

To create the Red Hat repository database, follow these steps:

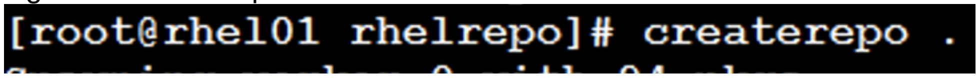
1. Install the "createrepo" package on admin node (rhel01). Use it to regenerate the repository database(s) for the local copy of the RHEL DVD contents.

```
# yum -y install createrepo
```

2. Run "createrepo" on the RHEL repository to create the repo database on admin node

```
# cd /var/www/html/rhelrepo
# createrepo .
```

Figure 58 createrepo



```
[root@rhel01 rhelrepo]# createrepo .
```

Set Up Ansible

To set up Ansible, follow these steps:

1. Download Ansible rpm from the following link: https://releases.ansible.com/ansible/rpm/release/epel-7-x86_64/ansible-2.7.11-1.el7.ans.noarch.rpm

```
# wget https://releases.ansible.com/ansible/rpm/release/epel-7-x86_64/ansible-2.7.11-1.el7.ans.noarch.rpm
```

2. Run the following command to install ansible:

```
# yum localinstall -y ansible-2.7.11-1.el7.ans.noarch.rpm
```

3. Verify Ansible installation by running the following commands:

```
# ansible --version
ansible 2.7.11
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules',
u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.5 (default, Sep 12 2018, 05:31:16) [GCC 4.8.5 20150623 (Red
Hat 4.8.5-36)]

# ansible localhost -m ping
 [WARNING]: provided hosts list is empty, only localhost is available. Note that the
implicit localhost does not match 'all'

localhost | SUCCESS => {
    "changed": false,
    "failed": false,
    "ping": "pong"
}
```

4. Prepare the host inventory file for Ansible as shown below. Various host groups have been created based on any specific installation requirements of certain hosts.

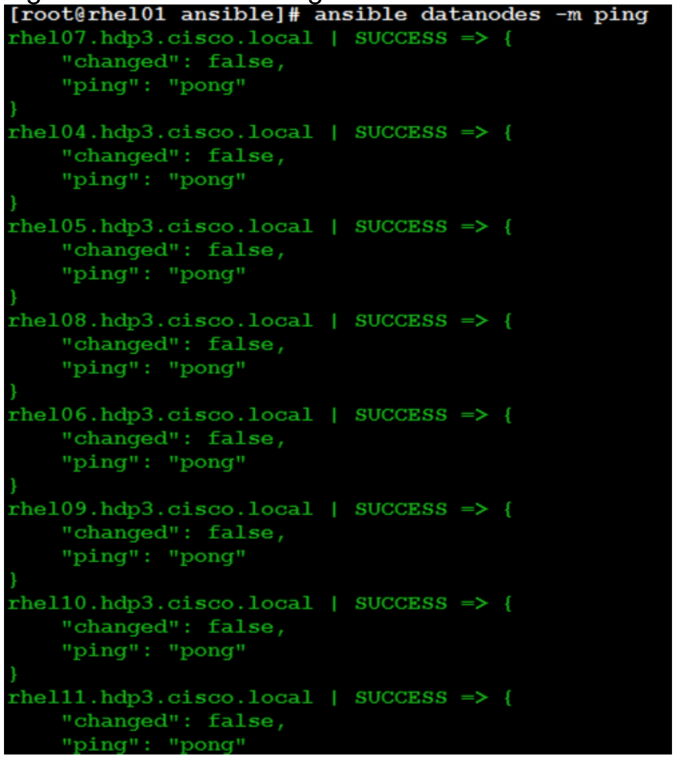
```
[root@rhel01 ~]# cat /etc/ansible/hosts
[admin]
rhel01.hdp3.cisco.local

[namenodes]
rhel01.hdp3.cisco.local
rhel02.hdp3.cisco.local
rhel03.hdp3.cisco.local


[datanodes]
rhel04.hdp3.cisco.local
rhel05.hdp3.cisco.local
rhel06.hdp3.cisco.local
rhel07.hdp3.cisco.local
rhel08.hdp3.cisco.local
rhel09.hdp3.cisco.local
rhel10.hdp3.cisco.local
rhel11.hdp3.cisco.local
rhel12.hdp3.cisco.local
rhel13.hdp3.cisco.local
rhel14.hdp3.cisco.local
rhel15.hdp3.cisco.local
rhel16.hdp3.cisco.local
rhel17.hdp3.cisco.local
rhel18.hdp3.cisco.local
rhel19.hdp3.cisco.local
rhel20.hdp3.cisco.local
rhel21.hdp3.cisco.local
rhel22.hdp3.cisco.local
rhel23.hdp3.cisco.local
```

```
rhel24.hdp3.cisco.local
rhel25.hdp3.cisco.local
rhel26.hdp3.cisco.local
rhel27.hdp3.cisco.local
rhel28.hdp3.cisco.local
rhel29.hdp3.cisco.local
rhel30.hdp3.cisco.local


[gpunodes]
rhel17.hdp3.cisco.local
rhel18.hdp3.cisco.local
rhel19.hdp3.cisco.local
rhel20.hdp3.cisco.local
rhel21.hdp3.cisco.local
rhel22.hdp3.cisco.local


[nodes]
rhel01.hdp3.cisco.local
rhel02.hdp3.cisco.local
rhel03.hdp3.cisco.local
rhel04.hdp3.cisco.local
rhel05.hdp3.cisco.local
rhel06.hdp3.cisco.local
rhel07.hdp3.cisco.local
rhel08.hdp3.cisco.local
rhel09.hdp3.cisco.local
rhel10.hdp3.cisco.local
rhel11.hdp3.cisco.local
rhel12.hdp3.cisco.local
rhel13.hdp3.cisco.local
rhel14.hdp3.cisco.local
rhel15.hdp3.cisco.local
rhel16.hdp3.cisco.local
rhel17.hdp3.cisco.local
rhel18.hdp3.cisco.local
rhel19.hdp3.cisco.local
rhel20.hdp3.cisco.local
rhel21.hdp3.cisco.local
rhel22.hdp3.cisco.local
rhel23.hdp3.cisco.local
rhel24.hdp3.cisco.local
rhel25.hdp3.cisco.local
rhel26.hdp3.cisco.local
rhel27.hdp3.cisco.local
rhel28.hdp3.cisco.local
rhel29.hdp3.cisco.local
rhel30.hdp3.cisco.local
```

5. Verify host group by running the following commands. Figure 59 shows the outcome of the ping command.

```
# ansible datanodes -m ping
```

Figure 59 Ansible – Ping Hosts



```
[root@rhel01 ansible]# ansible datanodes -m ping
rhel07.hdp3.cisco.local | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
rhel04.hdp3.cisco.local | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
rhel05.hdp3.cisco.local | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
rhel08.hdp3.cisco.local | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
rhel06.hdp3.cisco.local | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
rhel09.hdp3.cisco.local | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
rhel10.hdp3.cisco.local | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
rhel11.hdp3.cisco.local | SUCCESS => {
    "changed": false,
    "ping": "pong"
```

Install httpd

Setting up the RHEL repository on the admin node requires httpd. To set up RHEL repository on the admin node, follow these steps:

1. Install httpd on the admin node to host repositories:

The Red Hat repository is hosted using HTTP on the admin node; this machine is accessible by all the hosts in the cluster.

```
# yum -y install httpd
```

2. Add ServerName and make the necessary changes to the server configuration file:

```
   # vi /etc/httpd/conf/httpd.conf
    ServerName 10.13.1.31:80
```

3. Start httpd:

```
# service httpd start
# chkconfig httpd on
```

Disable the Linux Firewall

The default Linux firewall settings are far too restrictive for any Hadoop deployment. Since the Cisco UCS Big Data deployment will be in its own isolated network there is no need for that additional firewall.

```
# ansible all -m command -a "firewall-cmd --zone=public --add-port=80/tcp --
permanent"
# ansible all -m command -a "firewall-cmd --reload"
# ansible all -m command -a "systemctl disable firewalld"
```

Set Up All Nodes to use the RHEL Repository

To set up all noes to use the RHEL repository, follow these steps:

Based on this repository file, yum requires httpd to be running on rhel1 for other nodes to access the repository.

1. Copy the rheliso.repo to all the nodes of the cluster:

```
# ansible nodes -m copy -a "src=/var/www/html/rhelrepo/rheliso.repo
dest=/etc/yum.repos.d/."
```

2. Copy the /etc/hosts file to all nodes:

```
# ansible nodes -m copy -a "src=/etc/hosts dest=/etc/hosts"
```

3. Purge the yum caches:

```
# ansible nodes -a "yum clean all"
# ansible nodes -a "yum repolist"
```

While the suggested configuration is to disable SELinux as shown below, if for any reason SELinux needs to be enabled on the cluster, run the following command to make sure that the httpd can read the Yum repofiles.

```
  #chcon -R -t httpd_sys_content_t /var/www/html/
```

Upgrade the Cisco Network Driver for VIC1387

The latest Cisco Network driver is required for performance and updates. The latest drivers can be downloaded from the link below:

https://software.cisco.com/download/home/283862063/type/283853158/release/4.0(4)

```
In the ISO image, the required driver kmod-enic-3.2.210.18-738.12.rhel7u6.x86_64.rpm
can be located at  \Network\Cisco\VIC\RHEL\RHEL7.6\.
To upgrade the Cisco Network Driver for VIC1387, follow these steps:
From a node connected to the Internet, download, extract and transfer kmod-enic-.rpm
to rhel01 (admin node).
```

1. Copy the rpm on all nodes of the cluster using the following Ansible commands. For this example, the rpm is assumed to be in present working directory of rhel01:

```
[root@rhel01 ~]# ansible all -m copy -a "src=/root/ kmod-enic-3.2.210.18-
738.12.rhel7u6.x86_64.rpm dest=/root/."
```

2. Use the yum module to install the enic driver rpm file on all the nodes through Ansible:

```
[root@rhel01 ~]# ansible all -m yum -a "name=/root/kmod-enic-3.1.137.6-
700.16.rhel7u5.x86_64.rpm state=present"
Make sure that the above installed version of kmod-enic driver is being used on all
nodes by running the command "modinfo enic" on all nodes:
[root@rhel01 ~]# ansible all -m shell -a "modinfo enic | head -5"
```

3. It is recommended to download the kmod-megaraid driver for higher performance. The RPM can be found in the same package at: \Storage\LSI\Cisco_Storage_12G_SAS_RAID_controller\RHEL\RHEL7.6\kmod-megaraid_sas-07.708.03.00_el7.6-2.x86_64.rpm

4. Copy the rpm on all nodes of the cluster using the following Ansible commands. For this example, the rpm is assumed to be in present working directory of rhel01:

```
[root@rhel01 ~]# ansible all -m copy -a "src=/root/ kmod-megaraid_sas-
07.708.03.00_el7.6-2.x86_64.rpm dest=/root/."
```

5. Use the yum module to install the enic driver rpm file on all the nodes through Ansible:

```
[root@rhel01 ~]# ansible all -m yum -a "name=/root/ kmod-megaraid_sas-
07.708.03.00_el7.6-2.x86_64.rpm state=present"
Make sure that the above installed version of kmod-megaraid_sas driver is being used
on all nodes by running the command "modinfo enic" on all nodes:
[root@rhel01 ~]# ansible all -m shell -a "modinfo megaraid_sas | head -5"
```

Set Up JAVA

To setup JAVA, follow these steps:

CDH 6 requires JAVA 8.

1. Download jdk-8u211-linux-x64.rpm and src the rpm to admin node (rhel01) from the link: https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

2. Copy JDK rpm to all nodes:

```
# ansible nodes -m copy -a "src=/root/jdk-8u211-linux-x64.rpm dest=/root/."
```

3. Extract and Install JDK on all nodes:

```
# ansible all -m command -a "rpm -ivh jdk-8u211-linux-x64.rpm"
```

4. Create the following files java-set-alternatives.sh and java-home.sh on admin node (rhel01):

```
# vi java-set-alternatives.sh
#!/bin/bash
for item in java javac javaws jar jps javah javap jcontrol jconsole jdb; do
 rm -f /var/lib/alternatives/$item
 alternatives --install /usr/bin/$item $item /usr/java/jdk1.8.0_211-amd64/bin/$item
9
 alternatives --set $item /usr/java/jdk1.8.0_211-amd64/bin/$item
```

```
done

# vi java-home.sh
export JAVA_HOME=/usr/java/jdk1.8.0_211-amd64
```

5. Make the two java scripts created above executable:

```
chmod 755 ./java-set-alternatives.sh ./java-home.sh
```

6. Copying java-set-alternatives.sh to all nodes.

```
ansible nodes -m copy -a “src=/root/java-set-alternatives.sh dest=/root/.”
ansible nodes -m file -a "dest=/root/java-set-alternatives.sh mode=755"
ansible nodes -m copy -a “src=/root/java-home.sh dest=/root/.”
ansible nodes -m file -a "dest=/root/java-home.sh mode=755"
```

7. Setup Java Alternatives

```
[root@rhel01 ~]# ansible all -m shell -a "/root/java-set-alternatives.sh"
```

8. Make sure correct java is setup on all nodes (should point to newly installed java path).

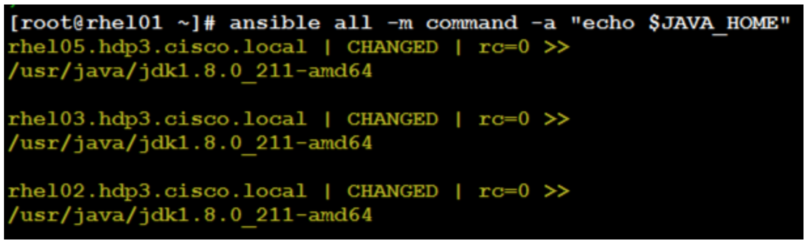
```
# ansible all -m shell -a "alternatives --display java | head -2"
```

9. Setup JAVA_HOME on all nodes.

```
# ansible all -m copy -a "src=/root/java-home.sh dest=/etc/profile.d"
```

10. Display JAVA_HOME on all nodes.

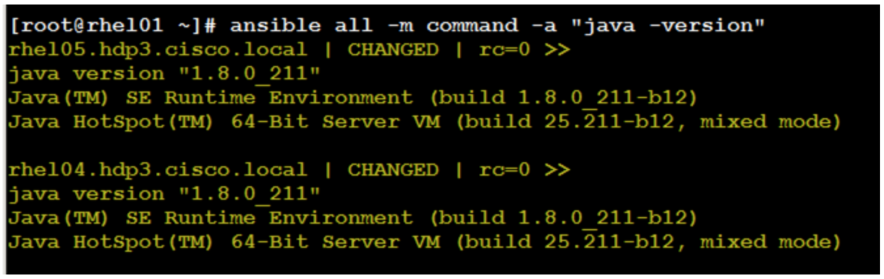
```
# ansible all -m command -a "echo $JAVA_HOME"
```



```
[root@rhel01 ~]# ansible all -m command -a "echo $JAVA_HOME"
rhel05.hdp3.cisco.local | CHANGED | rc=0 >>
/usr/java/jdk1.8.0_211-amd64

rhel03.hdp3.cisco.local | CHANGED | rc=0 >>
/usr/java/jdk1.8.0_211-amd64

rhel02.hdp3.cisco.local | CHANGED | rc=0 >>
/usr/java/jdk1.8.0_211-amd64
```

11. Display current java –version.

```
# ansible all -m command -a "java -version"
```



```
[root@rhel01 ~]# ansible all -m command -a "java -version"
rhel05.hdp3.cisco.local | CHANGED | rc=0 >>
java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)

rhel04.hdp3.cisco.local | CHANGED | rc=0 >>
java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)
```

Enable Syslog

Syslog must be enabled on each node to preserve logs regarding killed processes or failed jobs. Modern versions such as syslog-ng and rsyslog are possible, making it more difficult to be sure that a syslog daemon is present.

Use one of the following commands to confirm that the service is properly configured:

```
# ansible all -m command -a "rsyslogd -v"
# ansible all -m command -a "service rsyslog status"
```

Set the ulimit

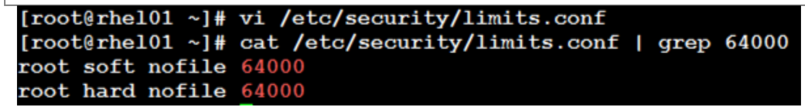
On each node, `ulimit -n` specifies the number of inodes that can be opened simultaneously. With the default value of 1024, the system appears to be out of disk space and shows no inodes available. This value should be set to 64000 on every node.

Higher values are unlikely to result in an appreciable performance gain.

To set ulimit, follow these steps:

1. For setting the ulimit on Redhat, edit "/etc/security/limits.conf" on admin node rhel01 and add the following lines:

```
root soft nofile 64000
root hard nofile 64000
```



```
[root@rhel01 ~]# vi /etc/security/limits.conf
[root@rhel01 ~]# cat /etc/security/limits.conf | grep 64000
root soft nofile 64000
root hard nofile 64000
```

2. Copy the /etc/security/limits.conf file from admin node (rhel01) to all the nodes using the following command:

```
# ansible nodes -m copy -a "src=/etc/security/limits.conf
dest=/etc/security/limits.conf"
```

3. Make sure that the /etc/pam.d/su file contains the following settings:

```
#%PAM-1.0
auth            sufficient      pam_rootok.so
# Uncomment the following line to implicitly trust users in the "wheel" group.
#auth           sufficient      pam_wheel.so trust use_uid
# Uncomment the following line to require a user to be in the "wheel" group.
#auth           required        pam_wheel.so use_uid
auth            include         system-auth
auth            include         postlogin
```

```
account         sufficient      pam_succeed_if.so uid = 0 use_uid quiet
account         include         system-auth
password        include         system-auth
session         include         system-auth
session         include         postlogin
session         optional        pam_xauth.so
```

The ulimit values are applied on a new shell, running the command on a node on an earlier instance of a shell will show old values.

Disable SELinux

SELinux must be disabled during the install procedure and cluster setup. SELinux can be enabled after installation and while the cluster is running.

SELinux can be disabled by editing `/etc/selinux/config` and changing the `SELINUX` line to `SELINUX=disabled`.

To disable SELinux, follow these steps:

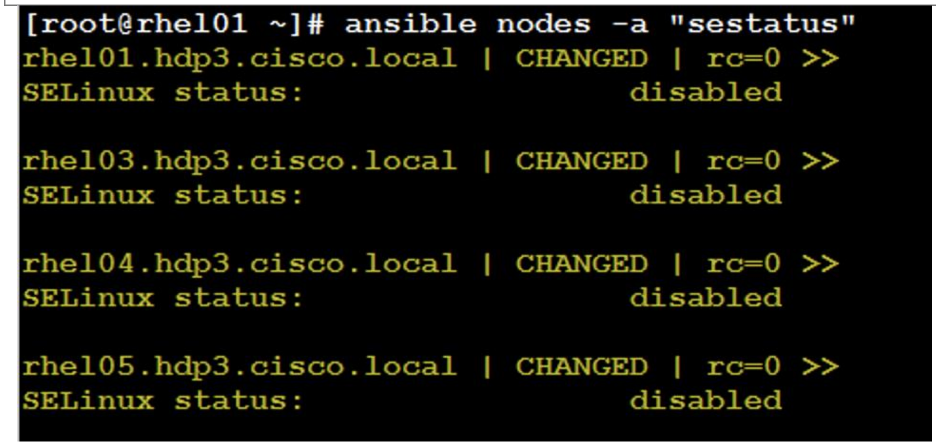
1. The following command will disable `SELINUX` on all nodes:

```
# ansible nodes -m shell -a "sed -i 's/SELINUX=enforcing/SELINUX=disabled/g'
/etc/selinux/config"
# ansible nodes -m shell -a "setenforce 0"
```

The above command may fail if SELinux is already disabled. This requires reboot to take effect.

2. Reboot the machine, if needed for SELinux to be disabled in case it does not take effect. It can be checked using the following command:

```
# ansible nodes -a "sestatus"
```



```
[root@rhel01 ~]# ansible nodes -a "sestatus"
rhel01.hdp3.cisco.local | CHANGED | rc=0 >>
SELinux status:                 disabled

rhel03.hdp3.cisco.local | CHANGED | rc=0 >>
SELinux status:                 disabled

rhel04.hdp3.cisco.local | CHANGED | rc=0 >>
SELinux status:                 disabled

rhel05.hdp3.cisco.local | CHANGED | rc=0 >>
SELinux status:                 disabled
```

Set TCP Retries

Adjusting the tcp_retries parameter for the system network enables faster detection of failed nodes. Given the advanced networking features of UCS, this is a safe and recommended change (failures observed at the operating system layer are most likely serious rather than transitory).

To set TCP retries, follow these steps:

On each node, set the number of TCP retries to 5 can help detect unreachable nodes with less latency.

1. Edit the file /etc/sysctl.conf and on admin node rhel01 and add the following lines:

```
net.ipv4.tcp_retries2=5
Copy the /etc/sysctl.conf file from admin node (rhel01) to all the nodes using the
following command:
# ansible nodes -m copy -a “src=/etc/sysctl.conf dest=/etc/sysctl.conf”
```

2. Load the settings from default sysctl file /etc/sysctl.conf by running the following command:

```
# ansible nodes -m command -a “sysctl -p”
```

Disable IPv6 Defaults

To disable IPv6 defaults, follow these steps:

1. Run the following command:

```
# ansible all -m shell -a "echo 'net.ipv6.conf.all.disable_ipv6 = 1' >>
/etc/sysctl.conf"
# ansible all -m shell -a "echo 'net.ipv6.conf.default.disable_ipv6 = 1' >>
/etc/sysctl.conf"
# ansible all -m shell -a "echo 'net.ipv6.conf.lo.disable_ipv6 = 1' >>
/etc/sysctl.conf"
```

2. Load the settings from default sysctl file /etc/sysctl.conf:

```
# ansible all -m shell -a "sysctl –p"
```

Disable Swapping

To disable swapping, follow these steps:

1. Run the following on all nodes. Variable `vm.swappiness` defines how often swap should be used, 60 is default:

```
# ansible all -m shell -a “echo 'vm.swappiness=1' >> /etc/sysctl.conf"
```

2. Load the settings from default sysctl file /etc/sysctl.conf and verify the content of sysctl.conf:

```
# ansible all -m shell -a "sysctl –p"
# ansible all -m shell -a “cat /etc/sysctl.conf”
```

Disable Memory Overcommit

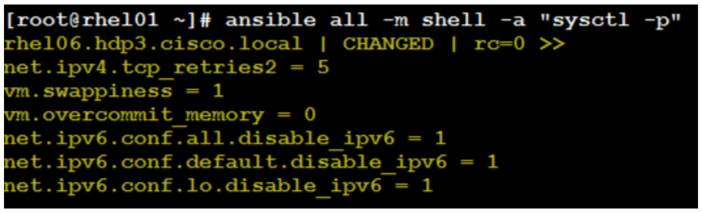
To disable Memory Overcommit, follow these steps:

1. Run the following on all nodes. Variable `vm.overcommit_memory=0`

```
# ansible all -m shell -a “echo 'vm.overcommit_memory=0' >> /etc/sysctl.conf"
```

2. Load the settings from default sysctl file /etc/sysctl.conf and verify the content of sysctl.conf:

```
# ansible all -m shell -a "sysctl -p"
# ansible all -m shell -a “cat /etc/sysctl.conf”
```



```
[root@rhel01 ~]# ansible all -m shell -a "sysctl -p"
rhel06.hdp3.cisco.local | CHANGED | rc=0 >>
net.ipv4.tcp_retries2 = 5
vm.swappiness = 1
vm.overcommit_memory = 0
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Disable Transparent Huge Pages

Disabling Transparent Huge Pages (THP) reduces elevated CPU usage caused by THP.

To disable Transparent Huge Pages, follow these steps:

1. You must run the following commands for every reboot; copy this command to /etc/rc.local so they are executed automatically for every reboot:

```
# ansible all -m shell -a "echo never > /sys/kernel/mm/transparent_hugepage/enabled”
# ansible all -m shell -a "echo never > /sys/kernel/mm/transparent_hugepage/defrag"
```

2. On the Admin node, run the following commands:

```
#rm -f /root/thp_disable
#echo "echo never > /sys/kernel/mm/transparent_hugepage/enabled" >>
/root/thp_disable
#echo "echo never > /sys/kernel/mm/transparent_hugepage/defrag " >>
/root/thp_disable
```

3. Copy file to each node:

```
# ansible nodes -m copy -a “src=/root/thp_disable dest=/root/thp_disable”
```

4. Append the content of file thp_disable to /etc/rc.local:

```
# ansible nodes -m shell -a “cat /root/thp_disable >> /etc/rc.local”
```

NTP Configuration

The Network Time Protocol (NTP) is used to synchronize the time of all the nodes within the cluster. The Network Time Protocol daemon (ntpd) sets and maintains the system time of day in synchronism with the timeserver located in the admin node (rhel01). Configuring NTP is critical for any Hadoop Cluster. If server clocks in the cluster drift out of sync, serious problems will occur with HBase and other services.

```
# ansible all -m yum -a “name=ntp state=present”
```

Installing an internal NTP server keeps your cluster synchronized even when an outside NTP server is inaccessible.

1. Configure /etc/ntp.conf on the admin node only with the following contents:

```
# vi /etc/ntp.conf
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
restrict -6 ::1
server 127.127.1.0
fudge 127.127.1.0 stratum 10
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```

2. Create /root/ntp.conf on the admin node and copy it to all nodes:

```
# vi /root/ntp.conf
server 10.13.1.31
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
restrict -6 ::1
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```

3. Copy ntp.conf file from the admin node to /etc of all the nodes by executing the following commands in the admin node (rhel01):

```
# ansible nodes -m copy -a “src=/root/ntp.conf dest=/etc/ntp.conf”
```

4. Run the following to syncronize the time and restart NTP daemon on all nodes:

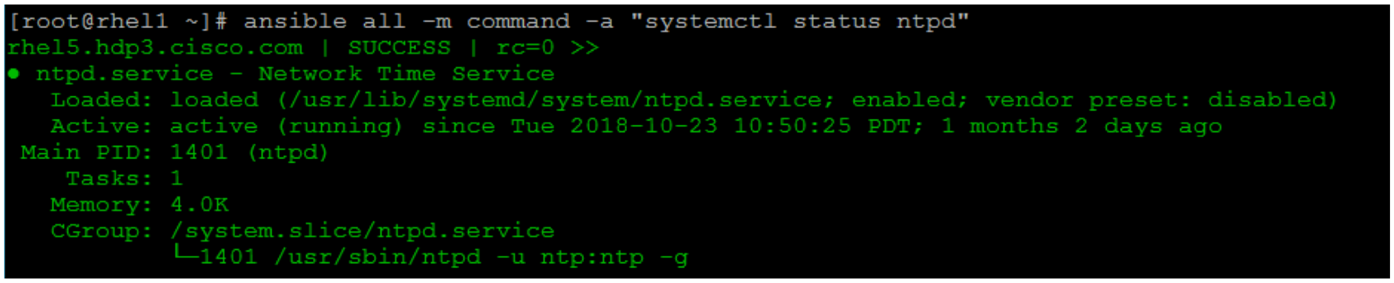
```
# ansible all -m service -a “name=ntpd state=stopped”
# ansible all -m command -a “ntpdate rhel01.hdp3.cisco.local”
# ansible all -m service -a “name=ntpd state=started”
```

5. Make sure to restart of NTP daemon across reboots:

```
# ansible all -a  "systemctl enable ntpd"
```

6. Verify NTP is up and running in all nodes by running the following commands:

```
# ansible all -a “systemctl status ntpd”
```



```
[root@rhel1 ~]# ansible all -m command -a "systemctl status ntpd"
rhel5.hdp3.cisco.com | SUCCESS | rc=0 >>
● ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2018-10-23 10:50:25 PDT; 1 months 2 days ago
 Main PID: 1401 (ntpd)
    Tasks: 1
   Memory: 4.0K
   CGroup: /system.slice/ntpd.service
           └─1401 /usr/sbin/ntpd -u ntp:ntp -g
```

Alternatively, the new Chrony service can be installed, that is quicker to synchronize clocks in mobile and virtual systems.

7. Install the Chrony service:

```
# ansible all -m yum -a "name=chrony state=present"
```

8. Activate the Chrony service at boot:

```
# ansible all -a "systemctl enable chronyd"
```

9. Start the Chrony service:

```
# ansible all -m service -a "name=chronyd state=started"systemctl start chronyd
The Chrony configuration is in the /etc/chrony.conf file, configured similar to
/etc/ntp.conf.
```

Install Megaraid StorCLI

This section explains the steps needed to install StorCLI (Storage Command Line Tool) which is a command line interface designed to be easy to use, consistent, and script. For more details, go to: https://docs.broadcom.com/docs/12352476

To install StorCLI, follow these steps:

1. Download StorCLI: https://www.broadcom.com/support/download-search/?pg=&pf=&pn=&po=&pa=&dk=storcli.

2. Extract the .zip file and copy storcli-1.23.02-1.noarch.rpm from the linux directory.

3. Download StorCLI and its dependencies and transfer to Admin node:

```
#scp storcli-1.23.02-1.noarch.rpm rhel01:/root/
```

4. Copy storcli rpm to all the nodes using the following commands:

```
# ansible all -m copy -a "src=/root/storcli-1.23.02-1.noarch.rpm dest=/root/."
```

5. Run this command to install storcli on all the nodes:

```
# ansible all -m shell -a "rpm -ivh storcli-1.23.02-1.noarch.rpm"
```

6. Run this command to copy storcli64 to root directory:

```
# ansible all -m shell -a "cp /opt/MegaRAID/storcli/storcli64 /root/."
```

7. Run this command to check the state of the disks:

```
# ansible all -m shell -a "./storcli64 /c0 show all"
```

The Cisco UCS Manager configuration explains the steps to deploy the required storage configuration via Storage Policy and Storage Profile attached to Service Profile Template for NameNode(s), Management Node(s), GPU Node(s) and DataNode(s). To configure Storage with StorCLI, go to section Configure Cisco Boot Optimized M.2 RAID Controller

Beginning with 4.0(4a), Cisco UCS Manager supports Cisco boot optimized M.2 RAID controller (UCS-M2-HWRAID), which is based on Marvell® 88SE92xx PCIe to SATA 6Gb/s controller.

The following M.2 drives are managed by the Cisco boot optimized M.2 RAID controller:

- 240GB M.2 6G SATA SSD
- 960GB M.2 6G SATA SSD

The Cisco boot optimized M.2 RAID controller supports only RAID1/JBOD (default - JBOD) mode and only UEFI boot mode.

The following are the limitations of the Cisco boot optimized M.2 RAID controller:

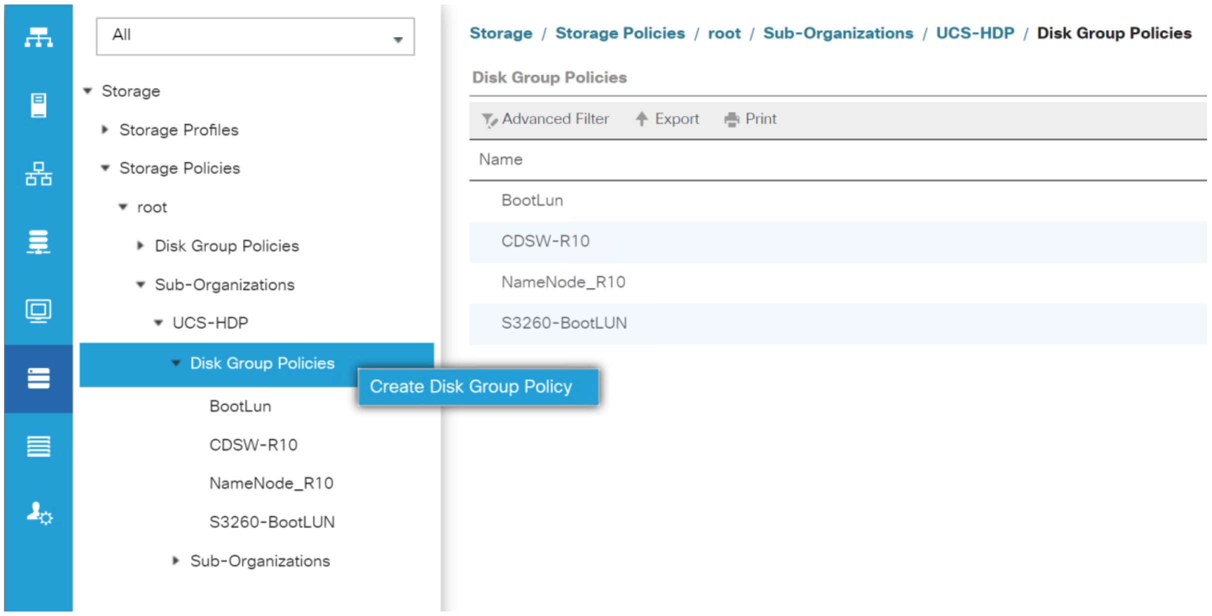
- Existing LUN migration is not supported.
- Local Disk Configuration policy is not supported.
- Entire disk capacity is used while creating single LUN.
- LUN is created using the Local LUN tab (see Configuring Local LUNs) under storage profile and not using the controller definitions.
- You cannot mix different capacity M.2 drives.

To create a `Disk Group Policy` and `Storage Profile Policy` to be attach with `Service Profile` for Cisco Optimized M.2 RAID Controller follow the steps in the following sections.

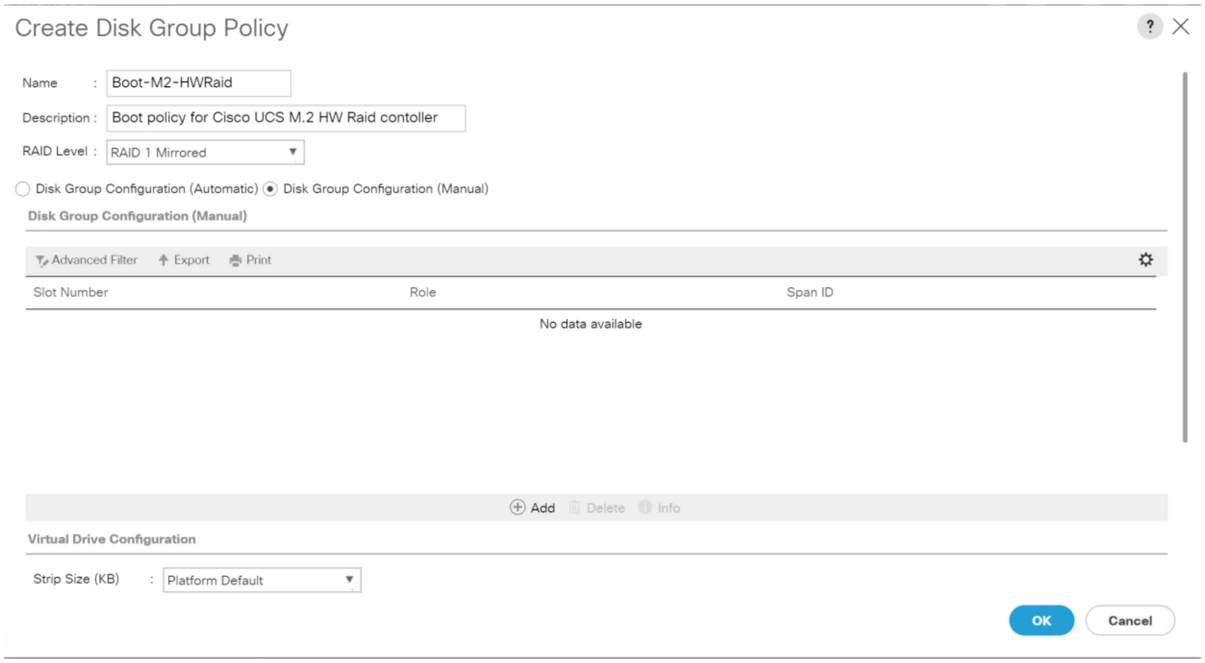
Configure Disk Group Policy

To configure the disk group policy, follow these steps:

1. In the UCSM WebUI, Go to storage tab. In the Storage Policy section, right-click Disk Group Policies. Click Create Disk Group Policy.



2. Enter a name and description for the new Disk Group Policy. Select Manual Disk Group Configuration. Click Add.


Create Disk Group Policy

Name : Boot-M2-HWRaid

Description : Boot policy for Cisco UCS M.2 HW Raid contoller

RAID Level : RAID 1 Mirrored

Disk Group Configuration (Automatic) [✓ Disk Group Configuration (Manual)]

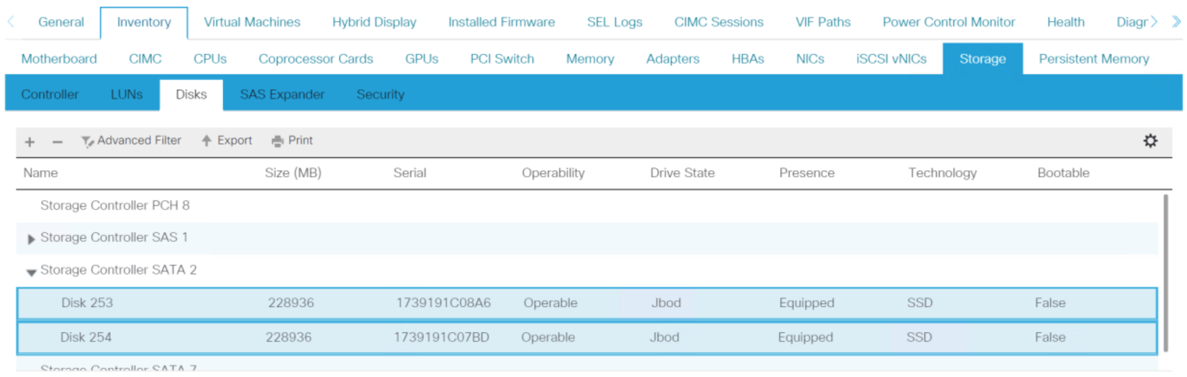
Disk Group Configuration (Manual)

Slot Number	Role	Span ID
No data available		

Virtual Drive Configuration

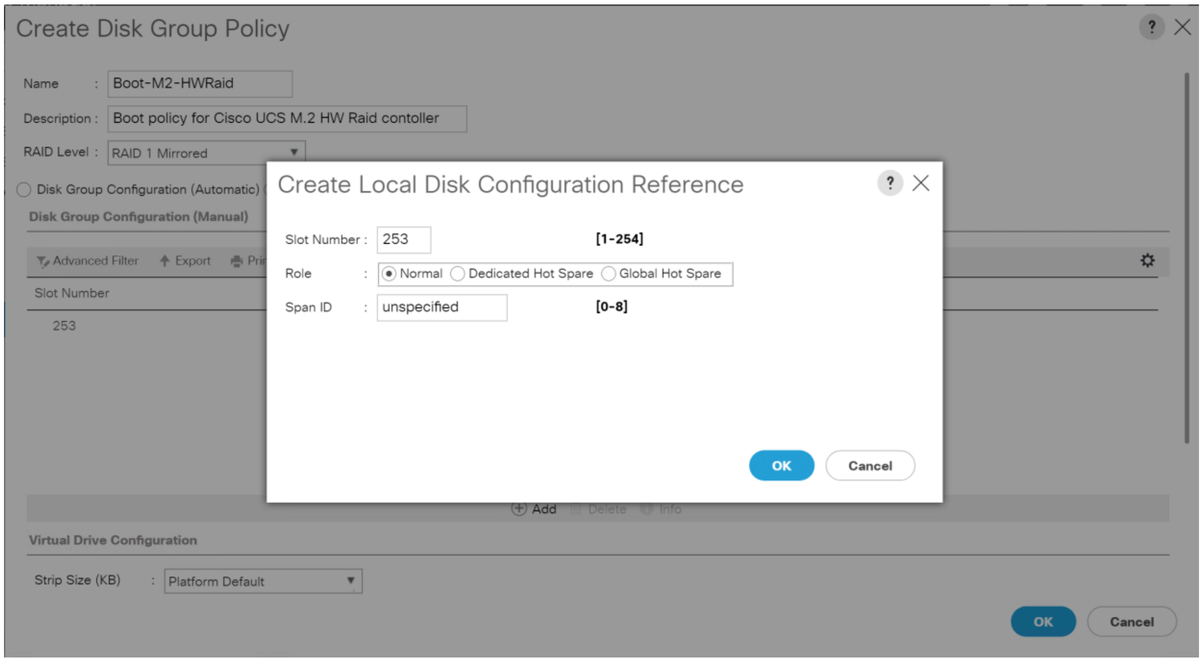
Strip Size (KB) : Platform Default

M.2 disks are allocated Disk slot Number 253 and 254.



Name	Size (MB)	Serial	Operability	Drive State	Presence	Technology	Bootable
Storage Controller PCH 8							
Storage Controller SAS 1							
Storage Controller SATA 2							
Disk 253	228936	1739191C08A6	Operable	Jbod	Equipped	SSD	False
Disk 254	228936	1739191C07BD	Operable	Jbod	Equipped	SSD	False

3. Enter Slot Number 253 for the first disk. Click OK.


Create Disk Group Policy

Name : Boot-M2-HWRaid

Description : Boot policy for Cisco UCS M.2 HW Raid contoller

RAID Level : RAID 1 Mirrored

Disk Group Configuration (Automatic)

Disk Group Configuration (Manual)

Slot Number
253

Create Local Disk Configuration Reference

Slot Number : 253 [1-254]

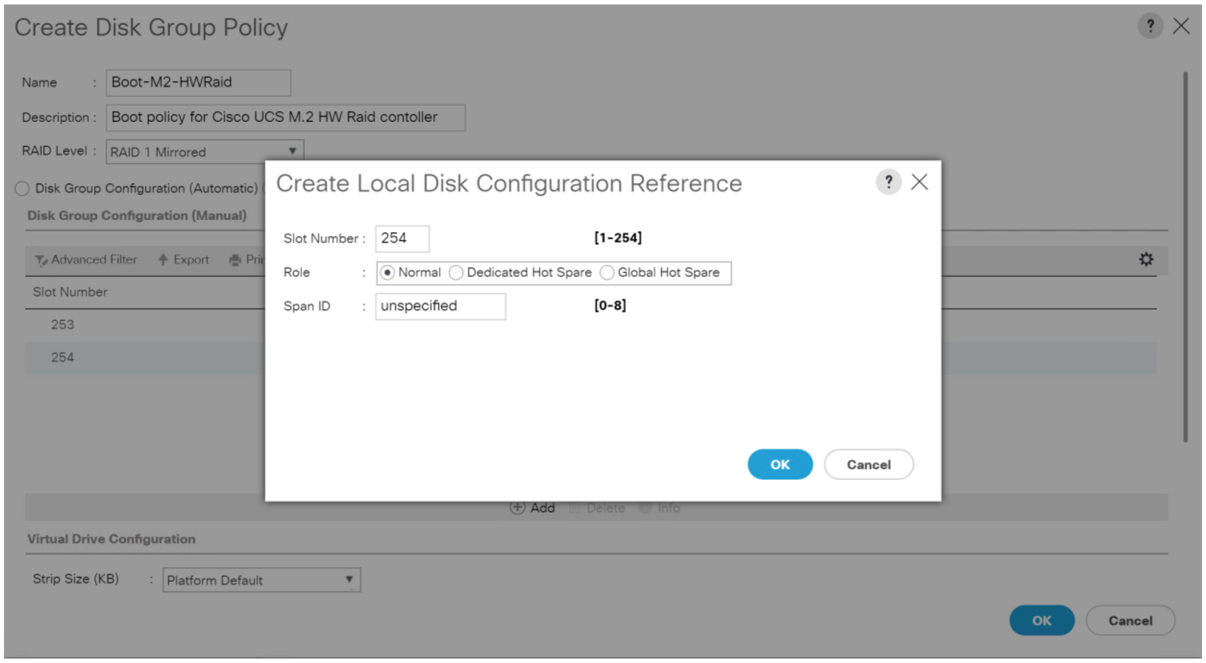
Role : [✓ Normal] Dedicated Hot Spare Global Hot Spare

Span ID : unspecified [0-8]

Virtual Drive Configuration

Strip Size (KB) : Platform Default

4. Click Add to add second disk, enter Slot Number 254.


Create Disk Group Policy

Name : Boot-M2-HWRaid

Description : Boot policy for Cisco UCS M.2 HW Raid contoller

RAID Level : RAID 1 Mirrored

Disk Group Configuration (Automatic)

Disk Group Configuration (Manual)

Slot Number
253
254

Create Local Disk Configuration Reference

Slot Number : 254 [1-254]

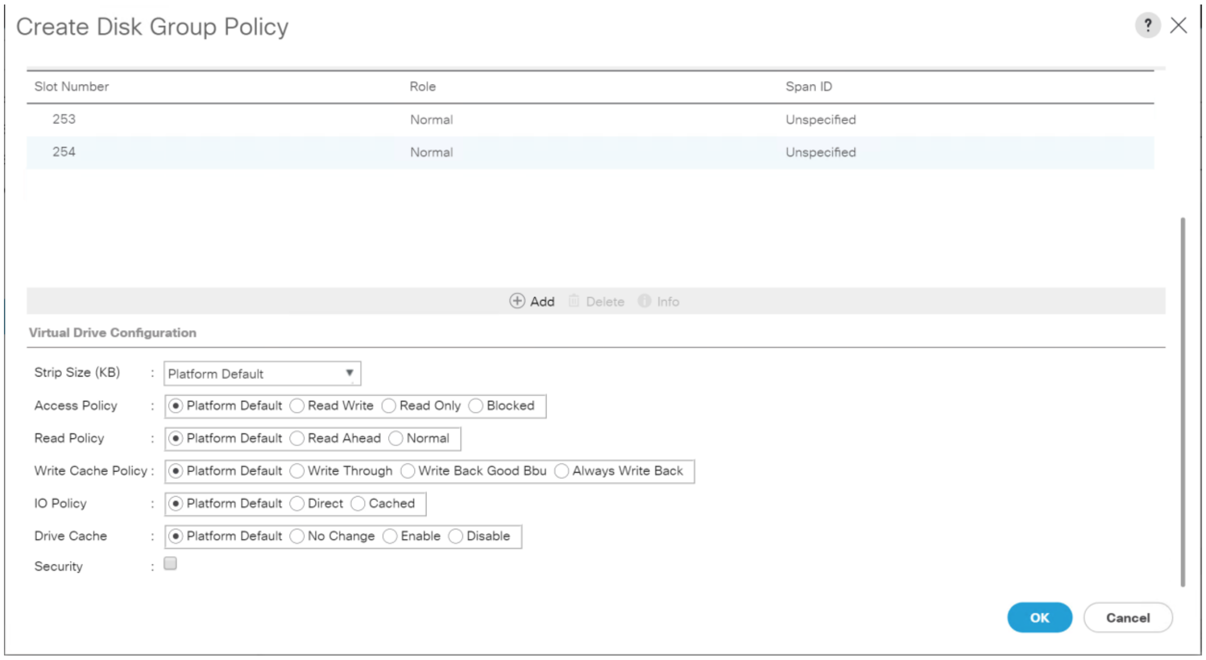
Role : [✓ Normal] Dedicated Hot Spare Global Hot Spare

Span ID : unspecified [0-8]

Virtual Drive Configuration

Strip Size (KB) : Platform Default

5. In Virtual Drive Configuration section leave all option as Platform Default. Click OK.


Create Disk Group Policy

Slot Number	Role	Span ID
253	Normal	Unspecified
254	Normal	Unspecified

Virtual Drive Configuration

Strip Size (KB) : Platform Default

Access Policy : [✓ Platform Default] Read Write Read Only Blocked

Read Policy : [✓ Platform Default] Read Ahead Normal

Write Cache Policy : [✓ Platform Default] Write Through Write Back Good Bbu Always Write Back

IO Policy : [✓ Platform Default] Direct Cached

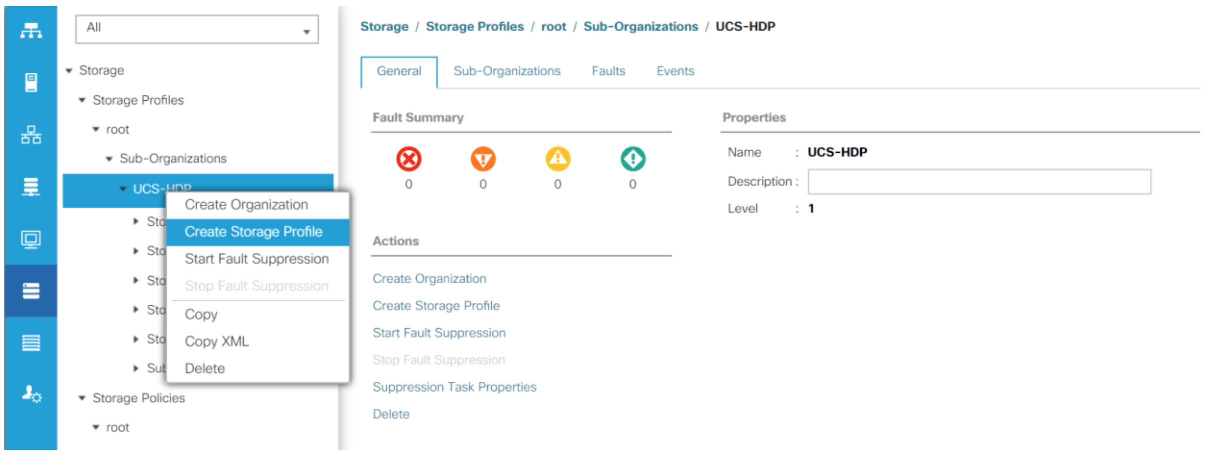
Drive Cache : [✓ Platform Default] No Change Enable Disable

Security : ☐

Configure Storage Profile

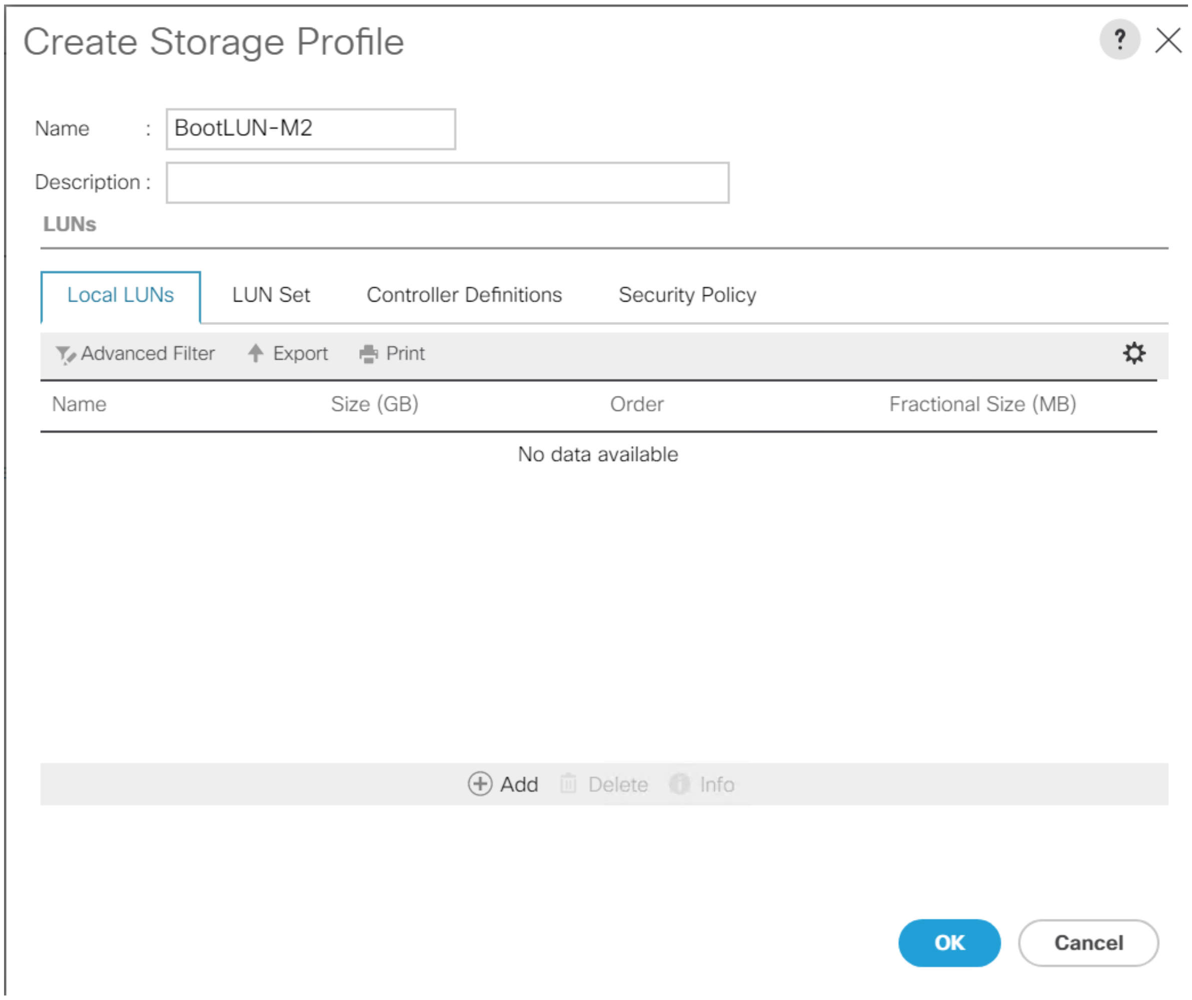
To configure the storage profile, follow these steps:

1. In the Storage Profiles section, select Storage Profiles. Right-click and select Create Storage Profile.



Name : UCS-HDP
Description :
Level : 1

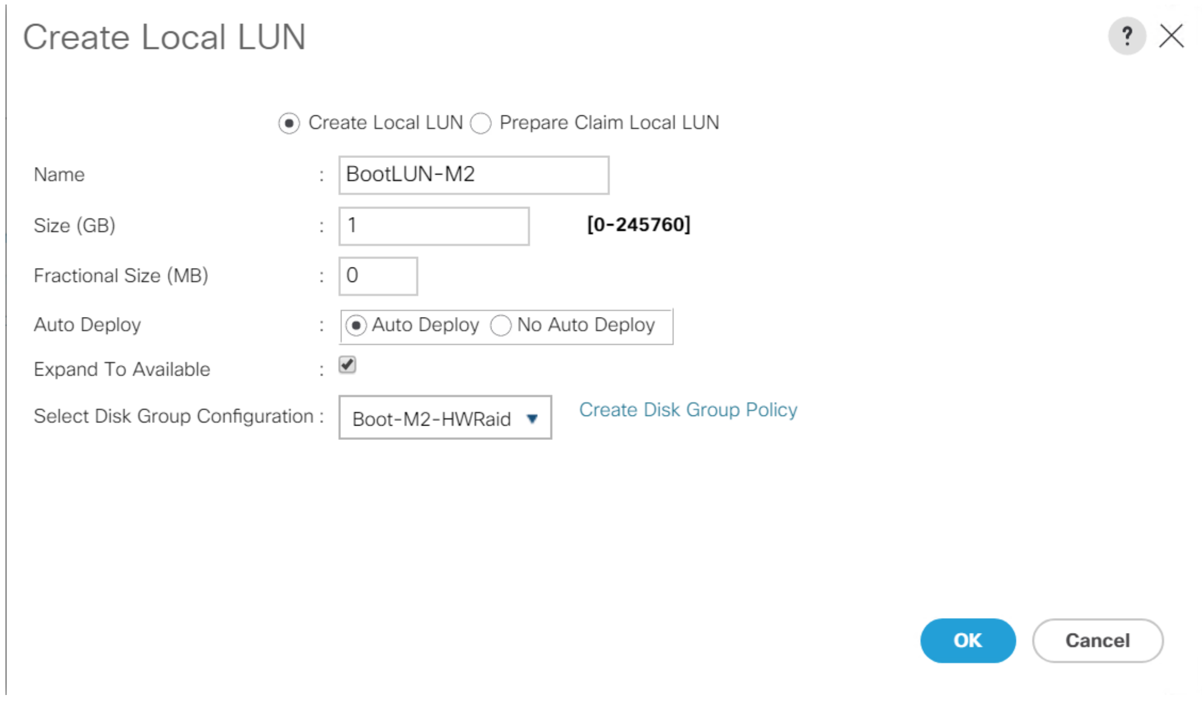
2. Enter a name for the Storage Profile. Click Add.


Create Storage Profile

Name : BootLUN-M2
Description :

Name	Size (GB)	Order	Fractional Size (MB)
No data available			

3. Enter a name for the Local LUN to be created, Click Auto Deploy, check the box for Expand to Available, and from the drop-down list for Disk Group Configuration, select RAID 1 Disk Group Policy created for M.2 SATA Disks. Click OK.


Create Local LUN

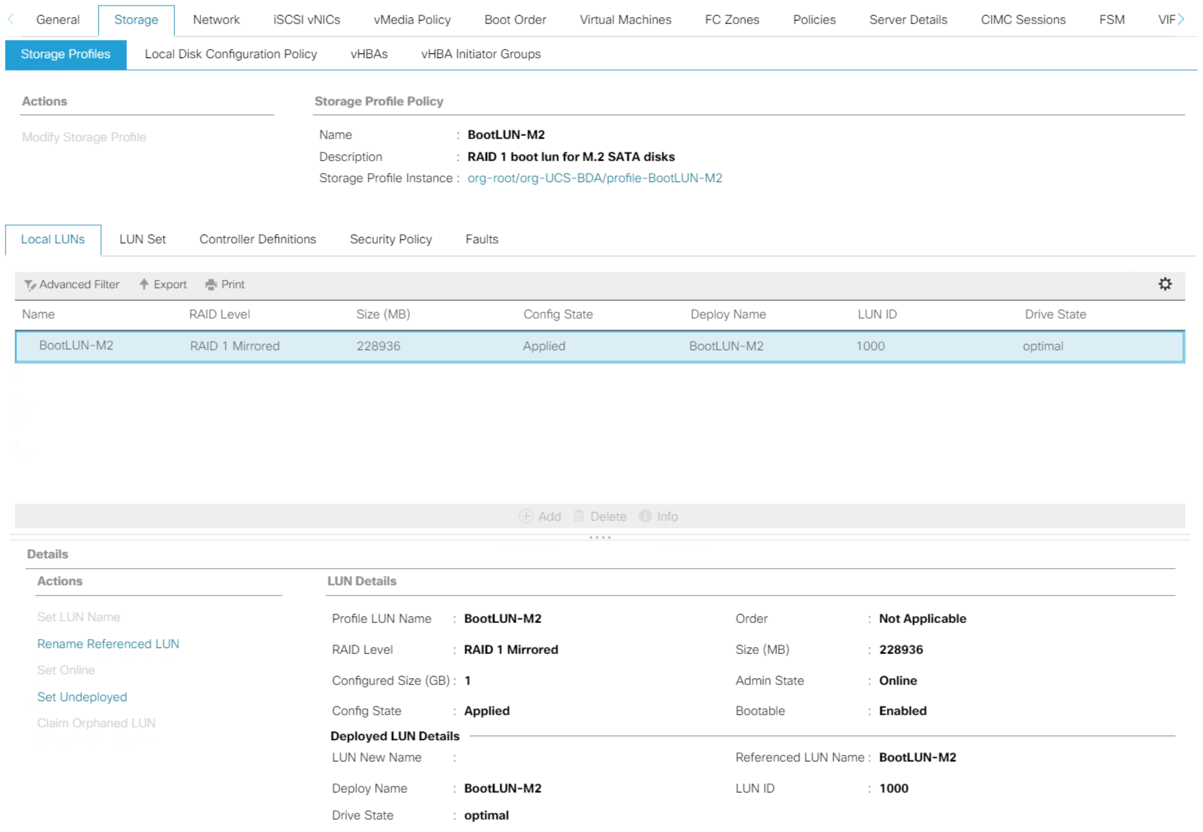
[✓ Create Local LUN] Prepare Claim Local LUN

Name :	BootLUN-M2
Size (GB) :	1 [0-245760]
Fractional Size (MB) :	0
Auto Deploy :	[✓ Auto Deploy] No Auto Deploy
Expand To Available :	✓
Select Disk Group Configuration :	Boot-M2-HWRaid

4. Attach a Storage Profile created to a Service profile or create a new Service Profile.

5. Go to the Storage tab on the Service Profile, select Storage Profile. Select Modify Storage Profile. Select Storage Profile created for M.2 SATA Disks.

Figure 60 Example of the Service Profile Associated to a Cisco UCS C240 M5 server with Cisco UCS-M2-HWRAID and 2 240GB M.2 SATA SSD Installed



Storage Profile Policy

Name	: BootLUN-M2
Description	: RAID 1 boot lun for M.2 SATA disks
Storage Profile Instance	: org-root/org-UCS-BDA/profile-BootLUN-M2

Name	RAID Level	Size (MB)	Config State	Deploy Name	LUN ID	Drive State
BootLUN-M2	RAID 1 Mirrored	228936	Applied	BootLUN-M2	1000	optimal

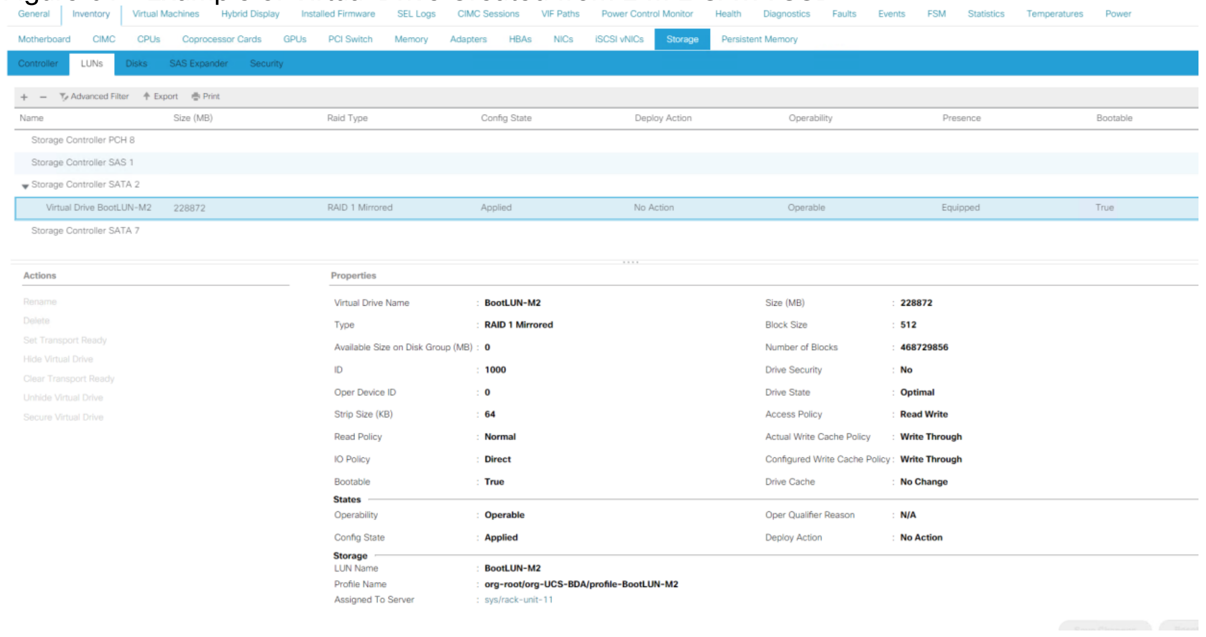
LUN Details

Profile LUN Name	: BootLUN-M2	Order	: Not Applicable
RAID Level	: RAID 1 Mirrored	Size (MB)	: 228936
Configured Size (GB)	: 1	Admin State	: Online
Config State	: Applied	Bootable	: Enabled

Deployed LUN Details

LUN New Name	:	Referenced LUN Name	: BootLUN-M2
Deploy Name	: BootLUN-M2	LUN ID	: 1000
Drive State	: optimal		

Figure 61 Example of Virtual Drive Created from 2 M.2 SATA SSD



Name	Size (MB)	Raid Type	Config State	Deploy Action	Operability	Presence	Bootable
Storage Controller PCH 8							
Storage Controller SAS 1							
Storage Controller SATA 2							
Virtual Drive BootLUN-M2	228872	RAID 1 Mirrored	Applied	No Action	Operable	Equipped	True
Storage Controller SATA 7							

Properties

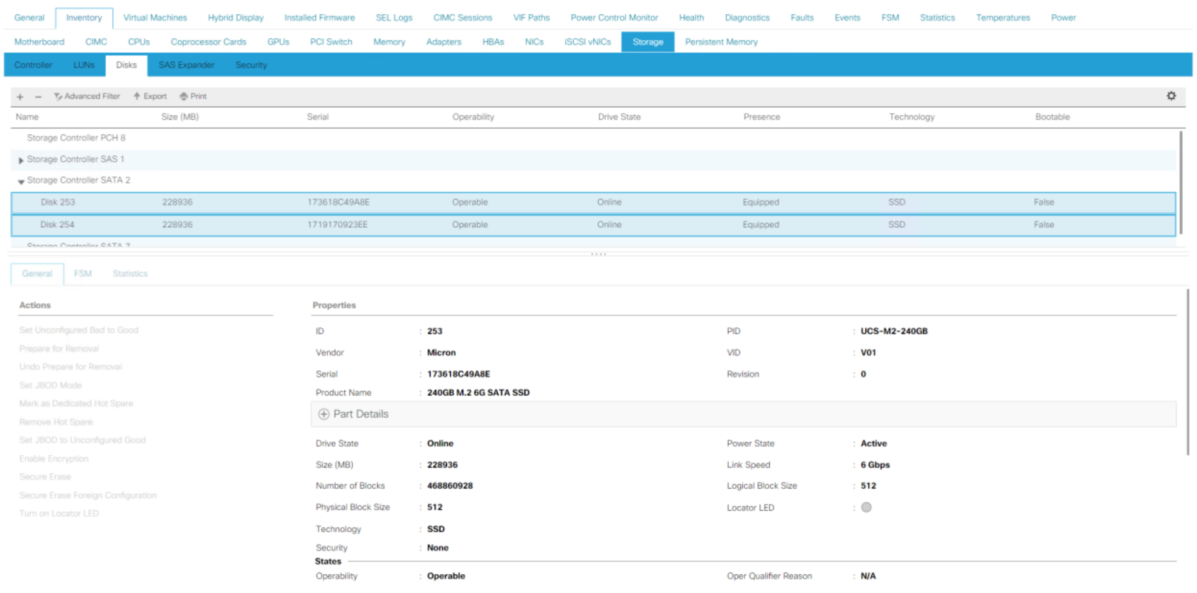
Virtual Drive Name	: BootLUN-M2	Size (MB)	: 228872
Type	: RAID 1 Mirrored	Block Size	: 512
Available Size on Disk Group (MB)	: 0	Number of Blocks	: 468729856
ID	: 1000	Drive Security	: No
Oper Device ID	: 0	Drive State	: Optimal
Strip Size (KB)	: 64	Access Policy	: Read Write
Read Policy	: Normal	Actual Write Cache Policy	: Write Through
IO Policy	: Direct	Configured Write Cache Policy	: Write Through
Bootable	: True	Drive Cache	: No Change

States

Operability	: Operable	Oper Qualifier Reason	: N/A
Config State	: Applied	Deploy Action	: No Action

Storage

LUN Name	: BootLUN-M2
Profile Name	: org-root/org-UCS-BDA/profile-BootLUN-M2
Assigned To Server	: sys/rack-unit-11



Name	Size (MB)	Serial	Operability	Drive State	Presence	Technology	Bootable
Storage Controller PCH 8							
Storage Controller SAS 1							
Storage Controller SATA 2							
Disk 253	228936	173618C49A8E	Operable	Online	Equipped	SSD	False
Disk 254	228936	1719170923EE	Operable	Online	Equipped	SSD	False

Properties

ID	: 253	PID	: UCS-M2-240GB
Vendor	: Micron	VID	: V01
Serial	: 173618C49A8E	Revision	: 0
Product Name	: 240GB M.2 6G SATA SSD		
Drive State	: Online	Power State	: Active
Size (MB)	: 228936	Link Speed	: 6 Gbps
Number of Blocks	: 468860928	Logical Block Size	: 512
Physical Block Size	: 512	Locator LED	: ●
Technology	: SSD		
Security	: None		

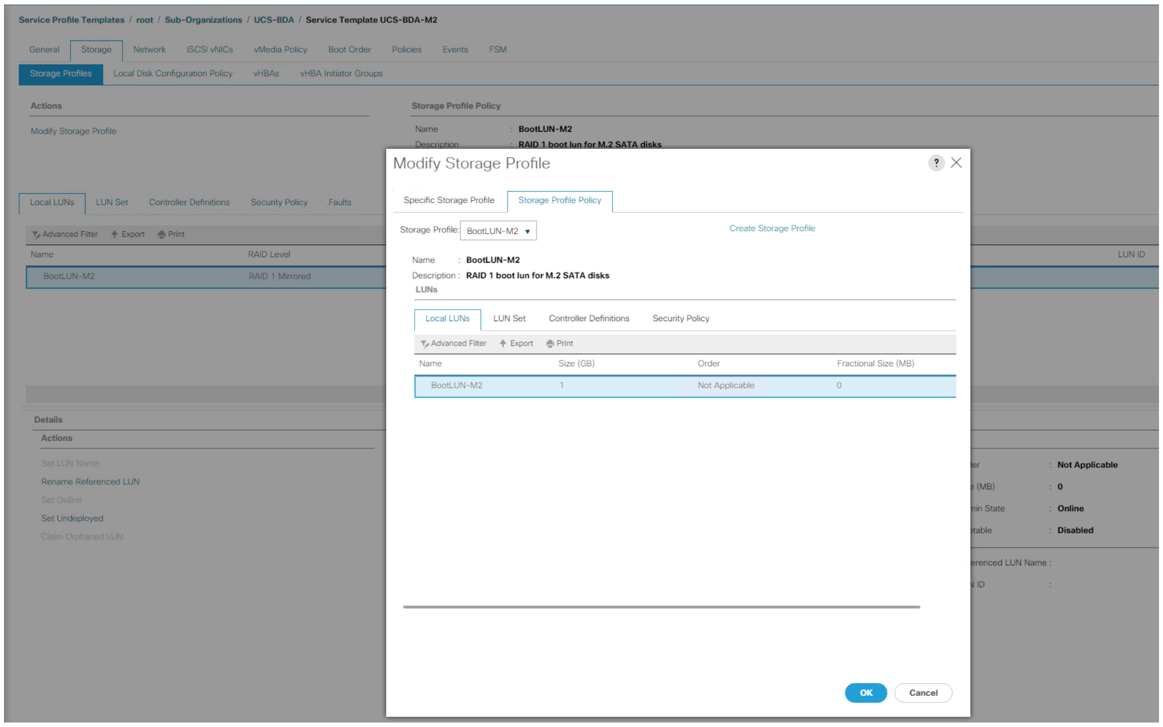
States

Operability	: Operable	Oper Qualifier Reason	: N/A

Apply Storage Profile in Service Profile Template

To create a new Service Profile template or update an existing template for Service Profile to attach a newly created Storage Profile for Cisco Boot Optimized RAID Controller, follow these steps:

1. Go to Service Profile Template.
2. Select Storage tab in Service Profile Template.
3. Select Storage Profile tab. Click Modify Storage Profile.
4. From the Storage Profile drop-down list, select Storage Profile for Cisco Boot Optimized RAID Controller.
5. If updating a Service Profile Template, once saved the changes in the configuration change in the Service Profile Template and are automatically applied to all Service Profile binded with the template.



Storage Profile Policy

Name : BootLUN-M2
Description : RAID 1 boot lun for M.2 SATA disks

Storage Profile: BootLUN-M2

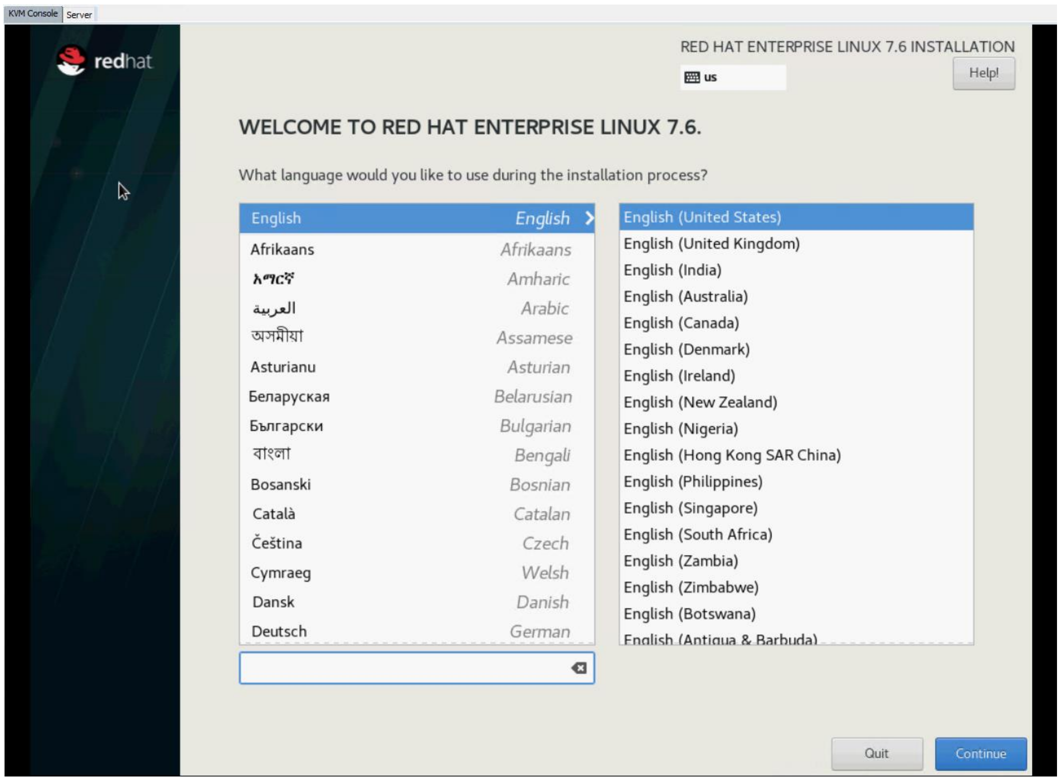
Name : BootLUN-M2
Description : RAID 1 boot lun for M.2 SATA disks
LUNs

Name	Size (GB)	Order	Fractional Size (MB)
BootLUN-M2	1	Not Applicable	0

Install RHEL 7.6 on Cisco Optimized M.2 RAID Controller

To install Red Hat Enterprise Linux 7.6 OS on Cisco UCS server with Virtual Drive created from Cisco Optimized M.2 RAID Controller (UCS-M2-HWRAID) in UEFI Boot Mode, follow these steps:

1. On the Welcome screen, select a language and click Continue.


KVM Console | Server

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

us

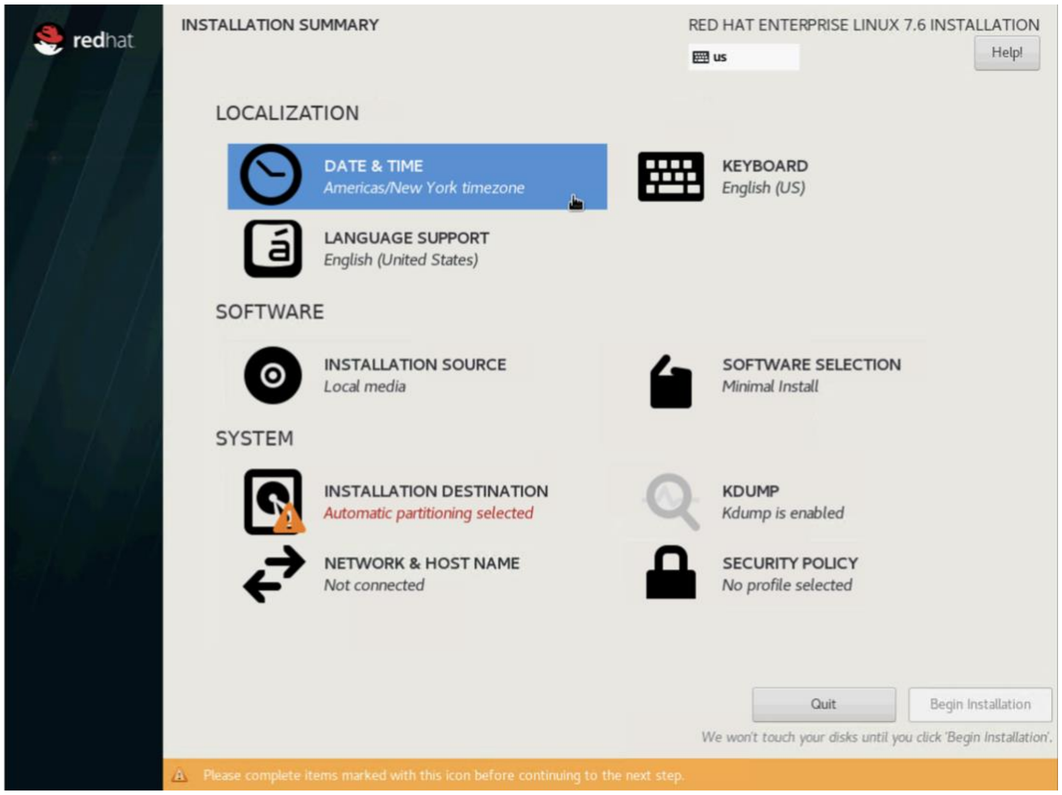
Help!

WELCOME TO RED HAT ENTERPRISE LINUX 7.6.

What language would you like to use during the installation process?

English	English	English (United States)
Afrikaans	Afrikaans	English (United Kingdom)
አማርኛ	Amharic	English (India)
العربية	Arabic	English (Australia)
অসমীয়া	Assamese	English (Canada)
Asturianu	Asturian	English (Denmark)
Беларуская	Belarusian	English (Ireland)
Български	Bulgarian	English (New Zealand)
বাংলা	Bengali	English (Nigeria)
Bosanski	Bosnian	English (Hong Kong SAR China)
Català	Catalan	English (Philippines)
Čeština	Czech	English (Singapore)
Cymraeg	Welsh	English (South Africa)
Dansk	Danish	English (Zambia)
Deutsch	German	English (Zimbabwe)
		English (Botswana)
		English (Antigua & Barbuda)

2. Select DATE & TIME.


INSTALLATION SUMMARY

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

us

Help!

LOCALIZATION

DATE & TIME
Americas/New York timezone

KEYBOARD
English (US)

LANGUAGE SUPPORT
English (United States)

SOFTWARE

INSTALLATION SOURCE
Local media

SOFTWARE SELECTION
Minimal Install

SYSTEM

INSTALLATION DESTINATION
Automatic partitioning selected

KDUMP
Kdump is enabled

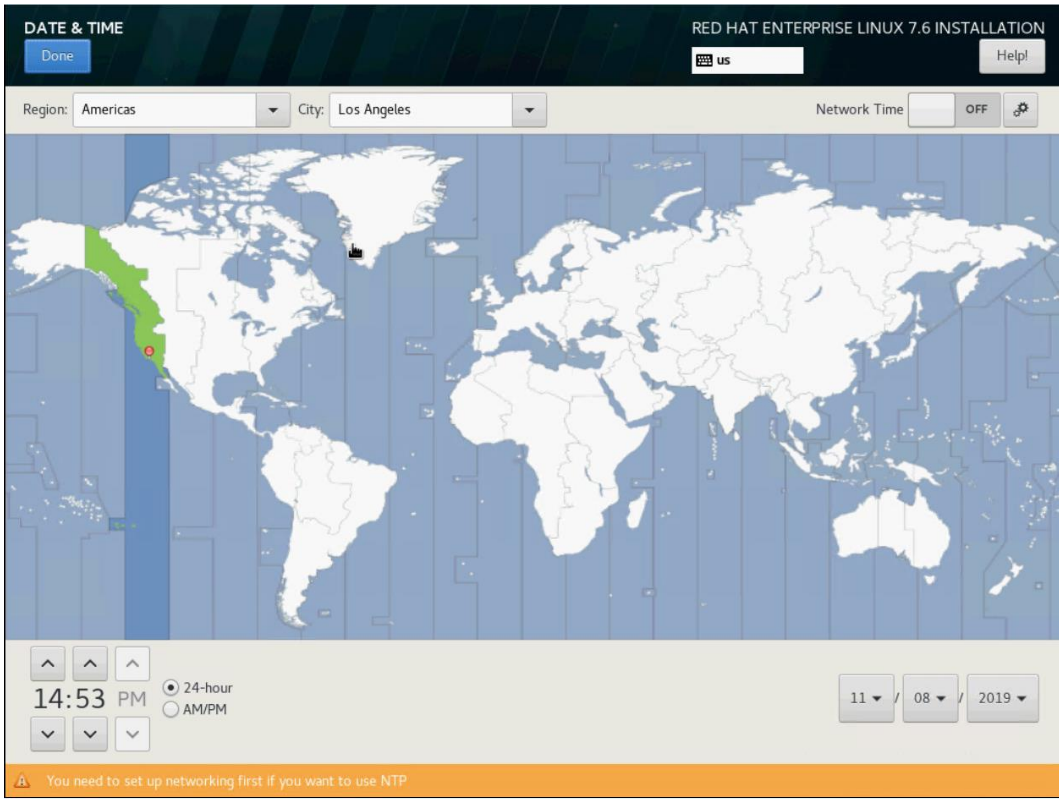
NETWORK & HOST NAME
Not connected

SECURITY POLICY
No profile selected

We won't touch your disks until you click 'Begin Installation'.

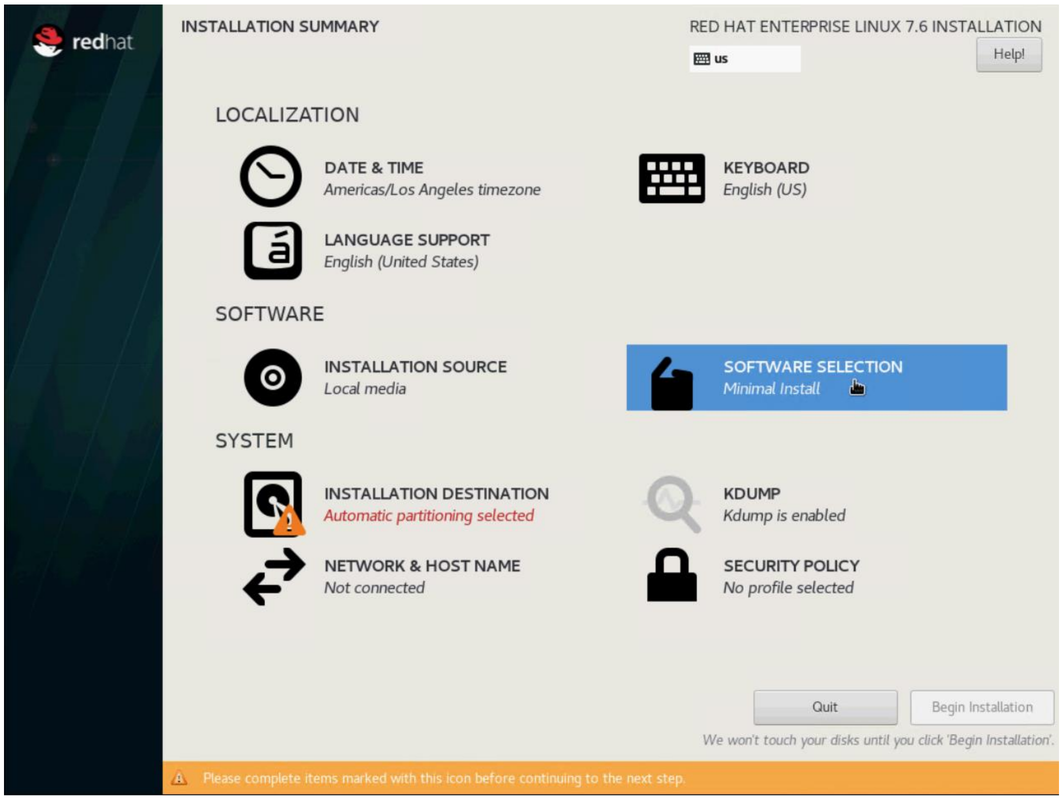
Please complete items marked with this icon before continuing to the next step.

3. Select region and City.



Region: Americas
City: Los Angeles
14:53 PM
11 / 08 / 2019
You need to set up networking first if you want to use NTP

4. Select SOFTWARE SELECTION.



DATE & TIME
Americas/Los Angeles timezone

KEYBOARD
English (US)

LANGUAGE SUPPORT
English (United States)

INSTALLATION SOURCE
Local media

SOFTWARE SELECTION
Minimal Install

INSTALLATION DESTINATION
Automatic partitioning selected

KDUMP
Kdump is enabled

NETWORK & HOST NAME
Not connected

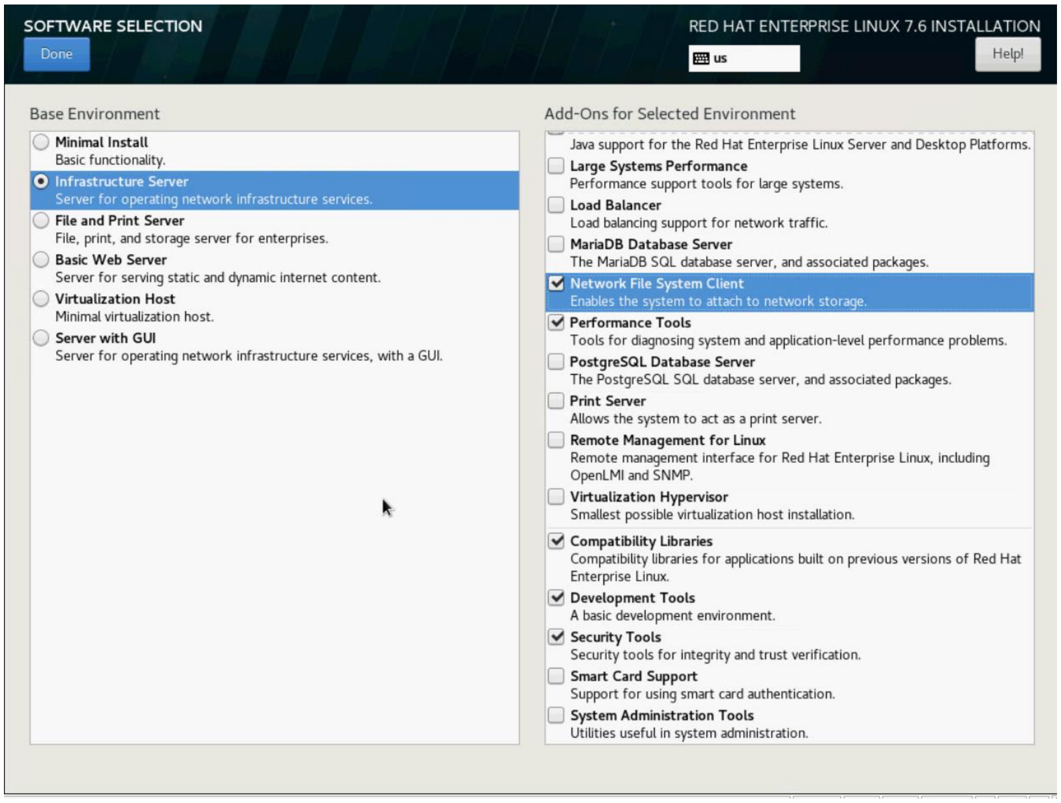
SECURITY POLICY
No profile selected

We won't touch your disks until you click 'Begin Installation'.

Please complete items marked with this icon before continuing to the next step.

5. Select Infrastructure Server in Base Environment. For Add-Ons for the selected environment, select:

- Network File System Client
- Performance Tools
- Compatibility Libraries
- Development Tools
- Security Tools



Base Environment

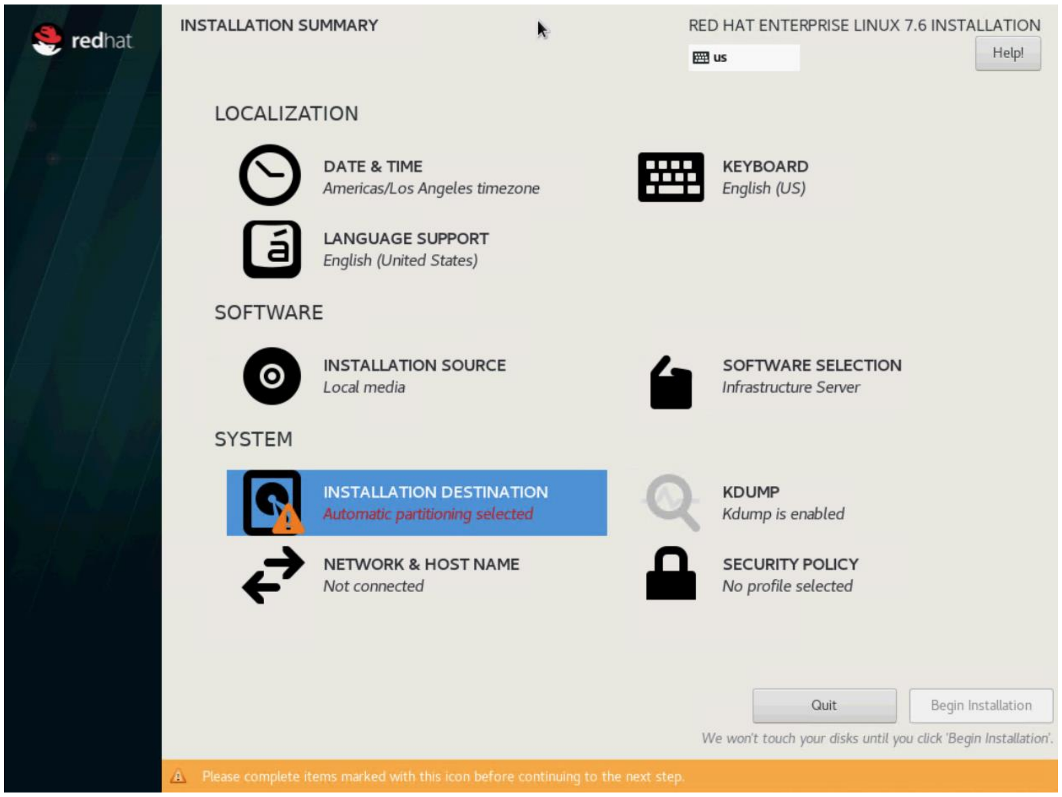
- ( ) **Minimal Install** Basic functionality.
- (●) **Infrastructure Server** Server for operating network infrastructure services.
- ( ) **File and Print Server** File, print, and storage server for enterprises.
- ( ) **Basic Web Server** Server for serving static and dynamic internet content.
- ( ) **Virtualization Host** Minimal virtualization host.
- ( ) **Server with GUI** Server for operating network infrastructure services, with a GUI.

Add-Ons for Selected Environment

Java support for the Red Hat Enterprise Linux Server and Desktop Platforms.

- [ ] **Large Systems Performance** Performance support tools for large systems.
- [ ] **Load Balancer** Load balancing support for network traffic.
- [ ] **MariaDB Database Server** The MariaDB SQL database server, and associated packages.
- [✓] **Network File System Client** Enables the system to attach to network storage.
- [✓] **Performance Tools** Tools for diagnosing system and application-level performance problems.
- [ ] **PostgreSQL Database Server** The PostgreSQL SQL database server, and associated packages.
- [ ] **Print Server** Allows the system to act as a print server.
- [ ] **Remote Management for Linux** Remote management interface for Red Hat Enterprise Linux, including OpenLMI and SNMP.
- [ ] **Virtualization Hypervisor** Smallest possible virtualization host installation.
- [✓] **Compatibility Libraries** Compatibility libraries for applications built on previous versions of Red Hat Enterprise Linux.
- [✓] **Development Tools** A basic development environment.
- [✓] **Security Tools** Security tools for integrity and trust verification.
- [ ] **Smart Card Support** Support for using smart card authentication.
- [ ] **System Administration Tools** Utilities useful in system administration.

6. Select Installation Destination.



DATE & TIME
Americas/Los Angeles timezone

KEYBOARD
English (US)

LANGUAGE SUPPORT
English (United States)

INSTALLATION SOURCE
Local media

SOFTWARE SELECTION
Infrastructure Server

INSTALLATION DESTINATION
Automatic partitioning selected

KDUMP
Kdump is enabled

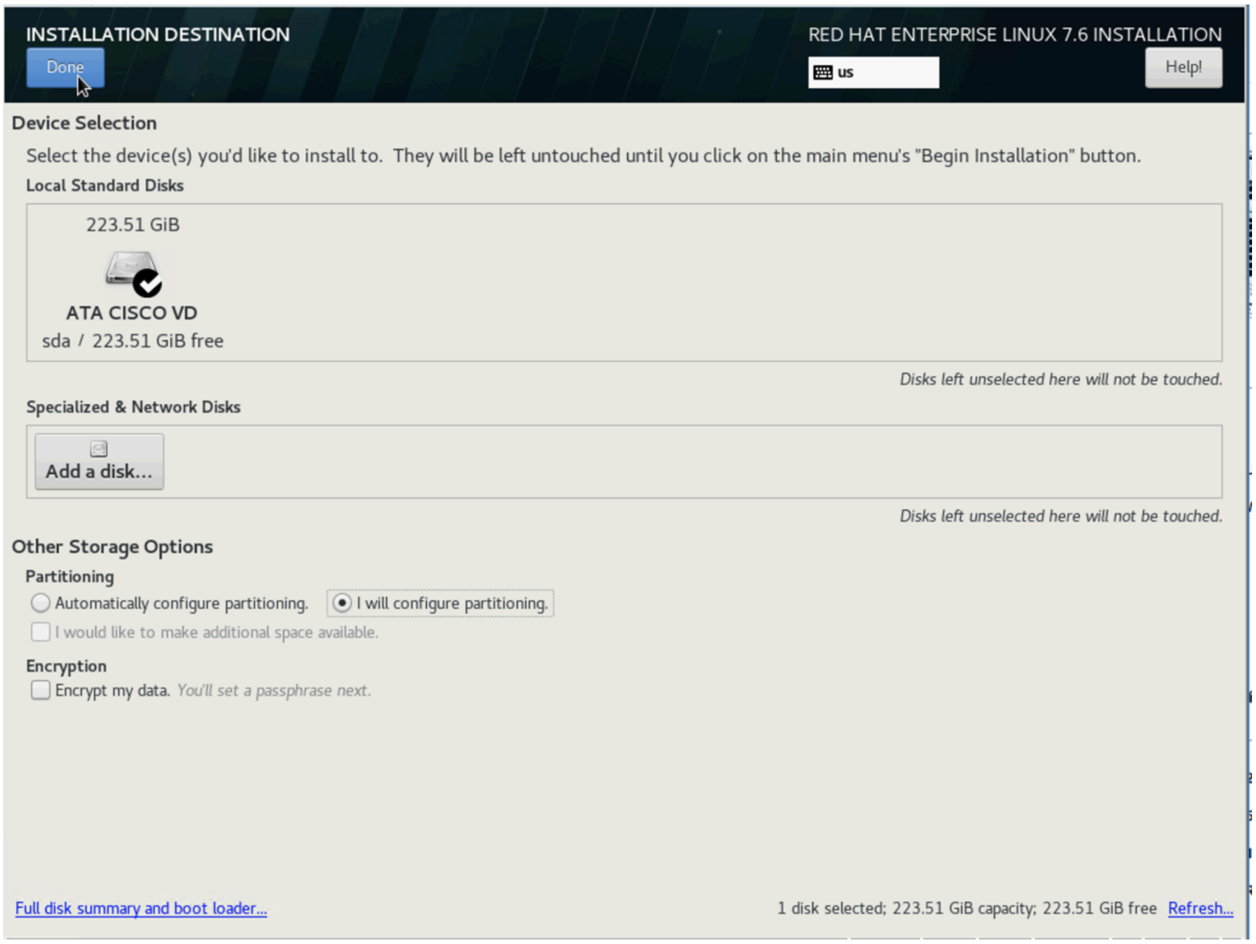
NETWORK & HOST NAME
Not connected

SECURITY POLICY
No profile selected

We won't touch your disks until you click 'Begin Installation'.

Please complete items marked with this icon before continuing to the next step.

7. Select the Virtual Drive created from M.2 SATA SSDs. Select I will Configure Partitioning. Click DONE.



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

223.51 GiB

ATA CISCO VD

sda / 223.51 GiB free

Disks left unselected here will not be touched.

Specialized & Network Disks

Disks left unselected here will not be touched.

Other Storage Options

Partitioning

Automatically configure partitioning. [✓ I will configure partitioning.]

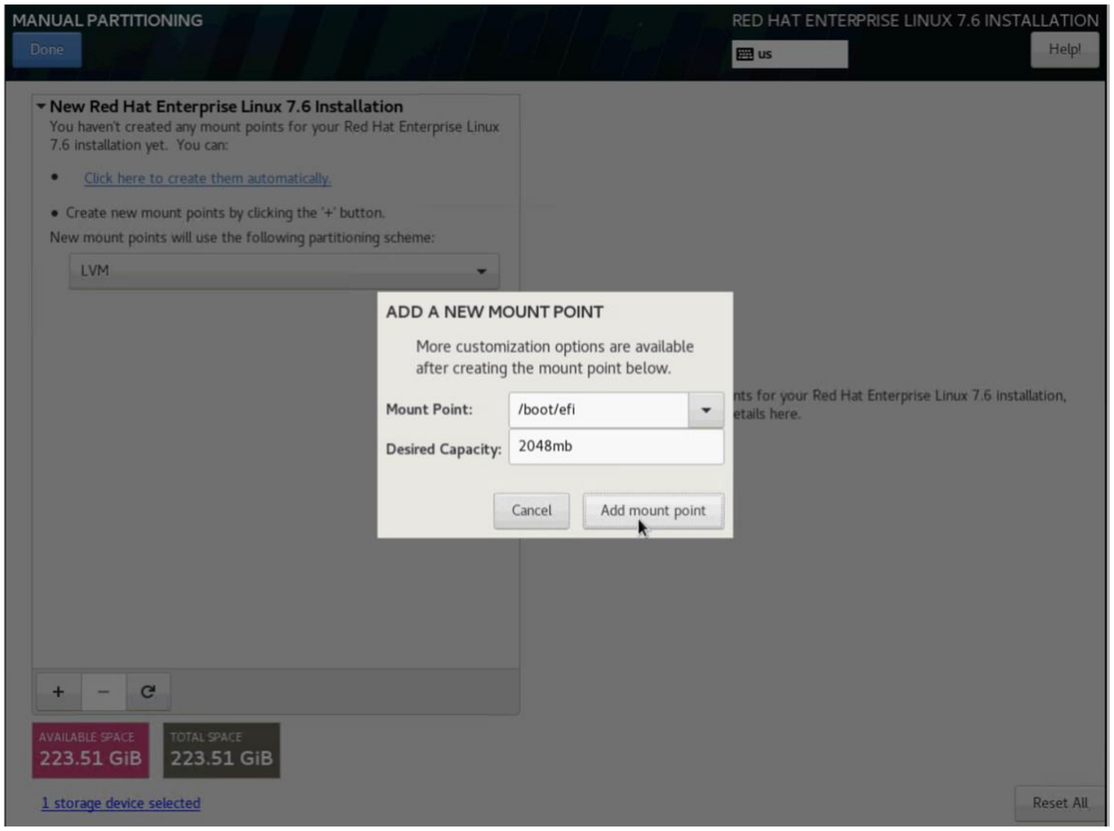
☐ I would like to make additional space available.

Encryption

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

1 disk selected; 223.51 GiB capacity; 223.51 GiB free

8. Click the + button to add manual configuration to install Red Hat Enterprise Linux 7.6. Enter /boot/efi as mount point and 2048mb as Desired Capacity.



New Red Hat Enterprise Linux 7.6 Installation
You haven't created any mount points for your Red Hat Enterprise Linux 7.6 installation yet. You can:

- Click here to create them automatically.
- Create new mount points by clicking the '+' button.

New mount points will use the following partitioning scheme:
LVM

ADD A NEW MOUNT POINT
More customization options are available after creating the mount point below.

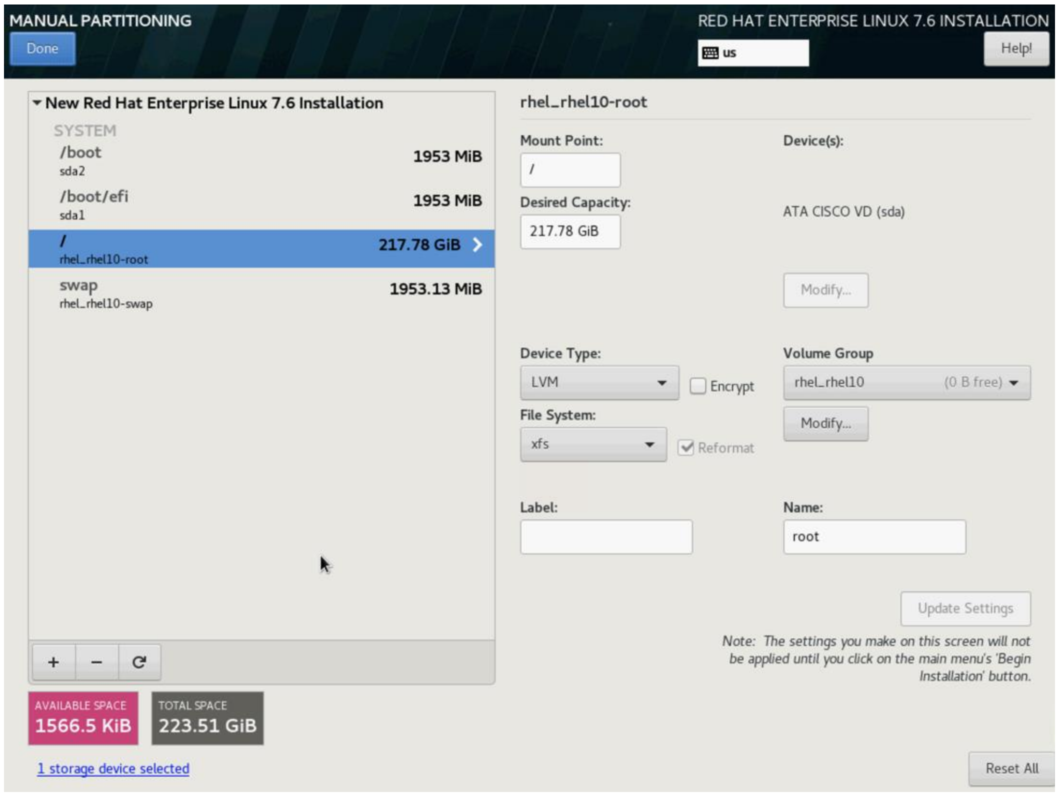
Mount Point: /boot/efi
Desired Capacity: 2048mb

9. Click the + button for the following mountpoint and desired capacity as shown below.

Table 8 Mount Point and Desired Capacity for RHEL Installation

Mountpoint	Desired Capacity
/boot/efi	2048mb
/boot	2048mb
Swap	2048mb
/	

10. Verify the mount points and desired capacity. Click DONE.



New Red Hat Enterprise Linux 7.6 Installation

SYSTEM

/boot sda2	1953 MiB
/boot/efi sda1	1953 MiB
/ rhel_rhel10-root	217.78 GiB
swap rhel_rhel10-swap	1953.13 MiB

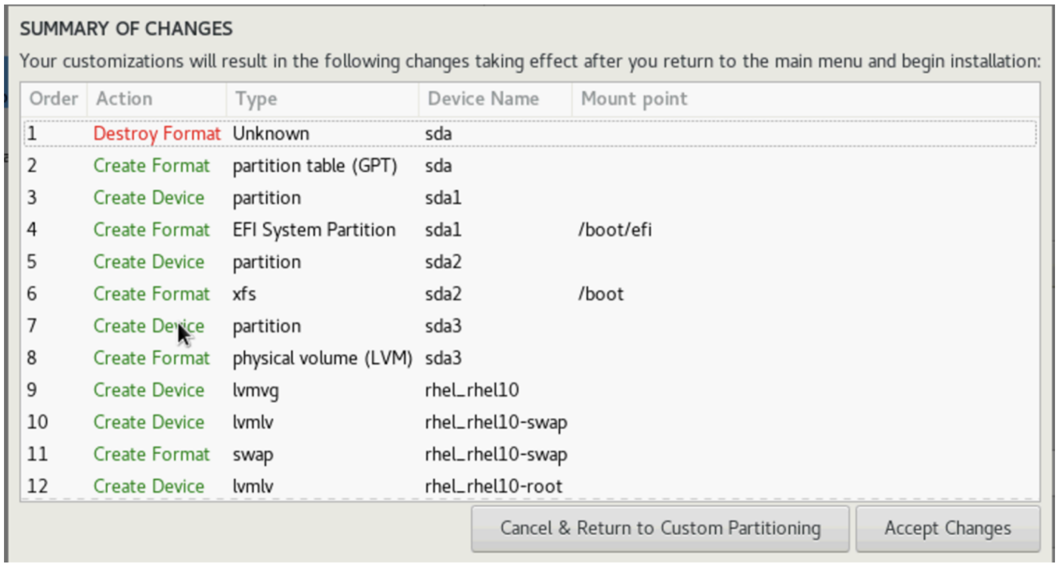
AVAILABLE SPACE 1566.5 KiB
TOTAL SPACE 223.51 GiB

rhel_rhel10-root

Mount Point: /
Desired Capacity: 217.78 GiB
Device(s): ATA CISCO VD (sda)
Device Type: LVM
Encrypt
Volume Group: rhel_rhel10 (0 B free)
File System: xfs
Reformat
Label:
Name: root

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

11. Click Accept Changes.

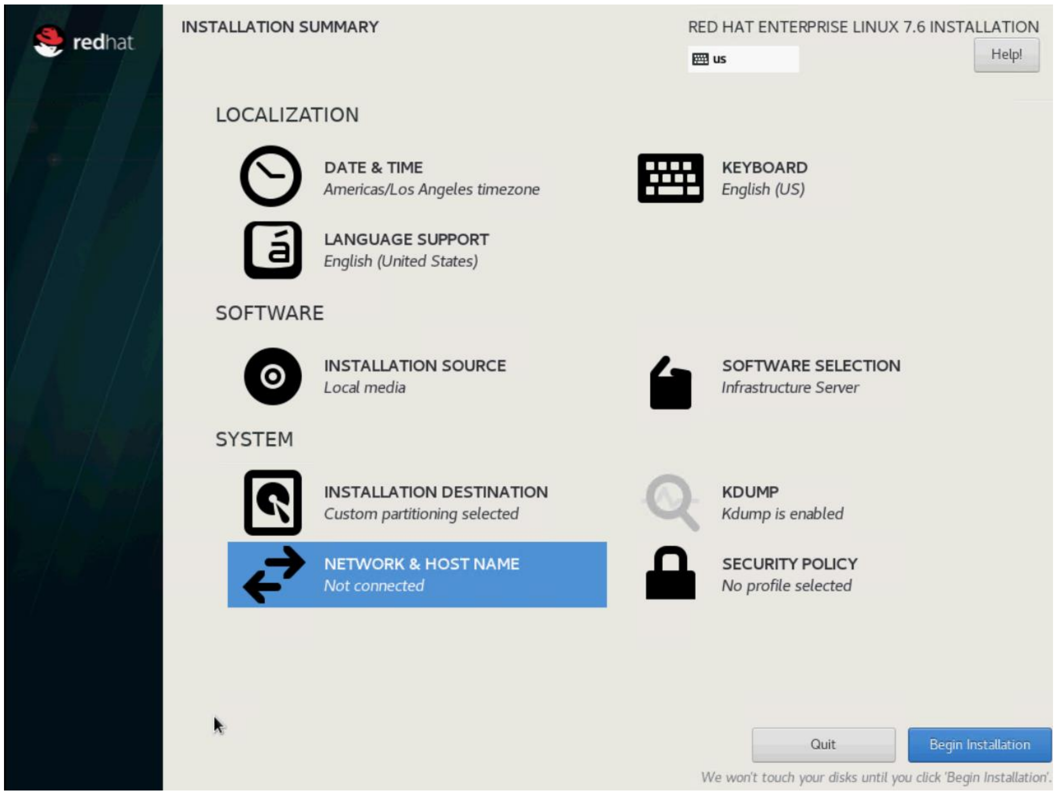


SUMMARY OF CHANGES

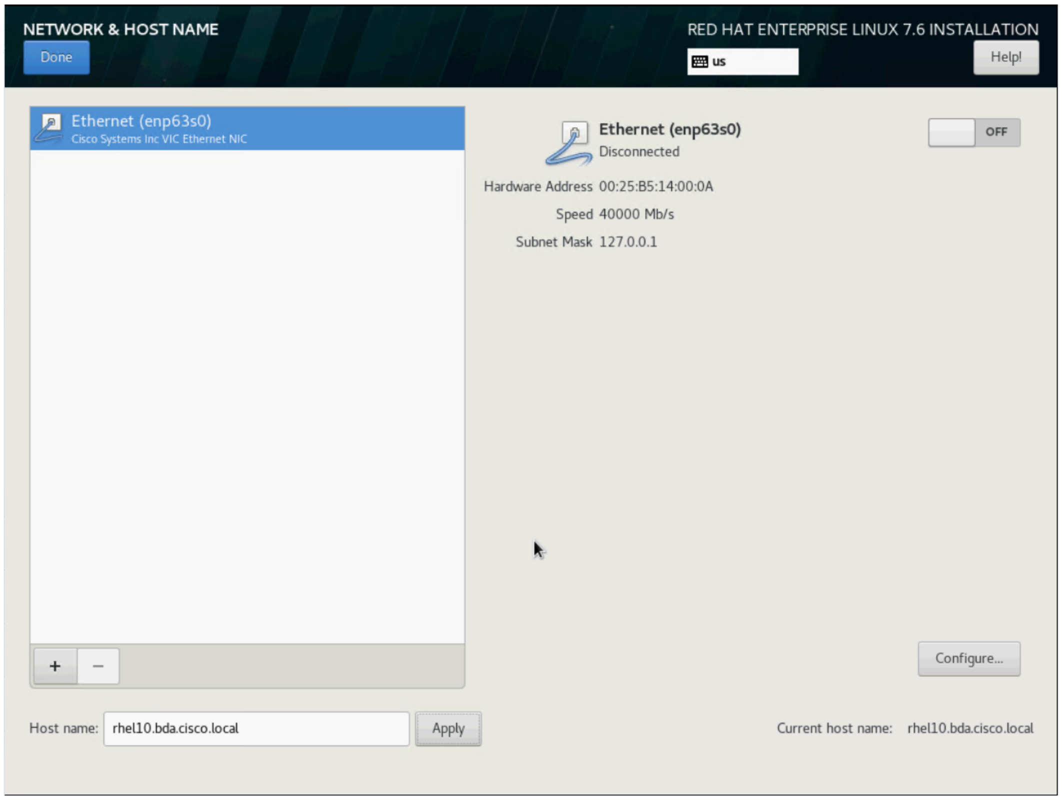
Your customizations will result in the following changes taking effect after you return to the main menu and begin installation:

Order	Action	Type	Device Name	Mount point
1	Destroy Format	Unknown	sda	
2	Create Format	partition table (GPT)	sda	
3	Create Device	partition	sda1	
4	Create Format	EFI System Partition	sda1	/boot/efi
5	Create Device	partition	sda2	
6	Create Format	xfs	sda2	/boot
7	Create Device	partition	sda3	
8	Create Format	physical volume (LVM)	sda3	
9	Create Device	lvmvg	rhel_rhel10	
10	Create Device	lvmlv	rhel_rhel10-swap	
11	Create Format	swap	rhel_rhel10-swap	
12	Create Device	lvmlv	rhel_rhel10-root	

12. Click NETWORK & HOST NAME.



13. Enter Host name, click Apply. Select Configure.


NETWORK & HOST NAME
Done
RED HAT ENTERPRISE LINUX 7.6 INSTALLATION
us
Help!
Ethernet (enp63s0)
Cisco Systems Inc VIC Ethernet NIC
OFF
+
−
Configure...
Apply

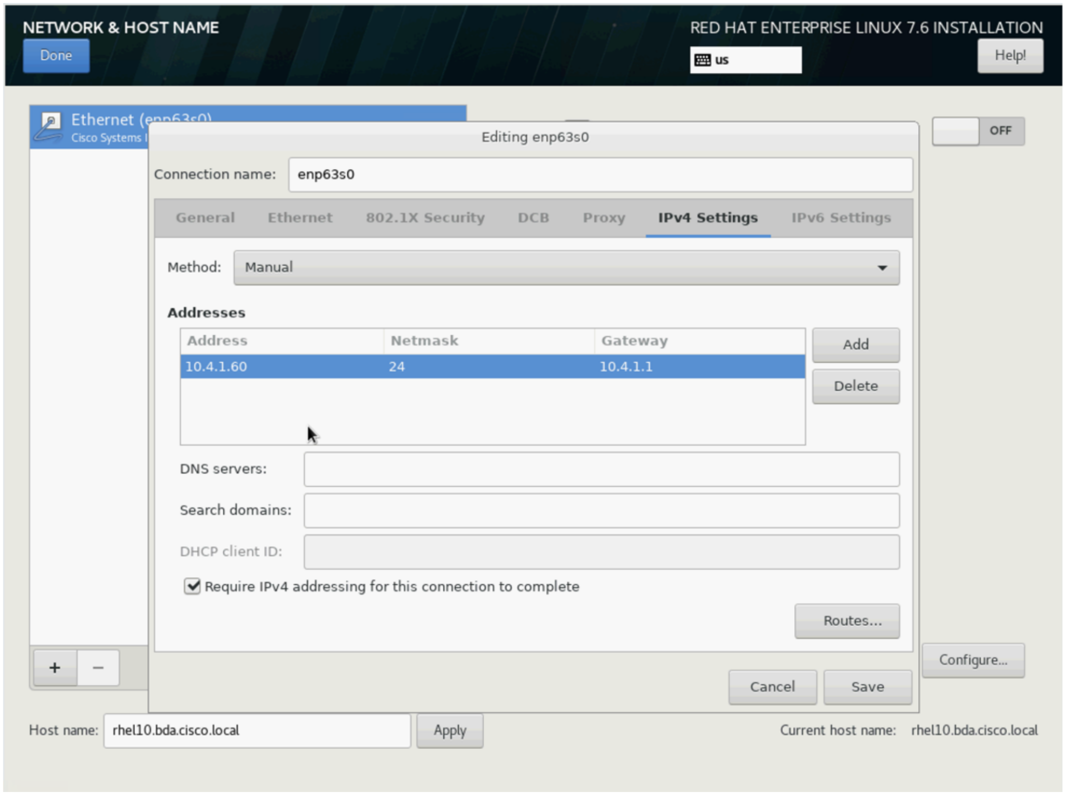
Ethernet (enp63s0)
Disconnected

Hardware Address 00:25:B5:14:00:0A
Speed 40000 Mb/s
Subnet Mask 127.0.0.1

Host name: rhel10.bda.cisco.local

Current host name: rhel10.bda.cisco.local

14. Select IPv4 Settings, Enter IP Address, Netmask and Gateway. Click Save.



Connection name: enp63s0

Method: Manual

Addresses

Address	Netmask	Gateway
10.4.1.60	24	10.4.1.1

DNS servers:

Search domains:

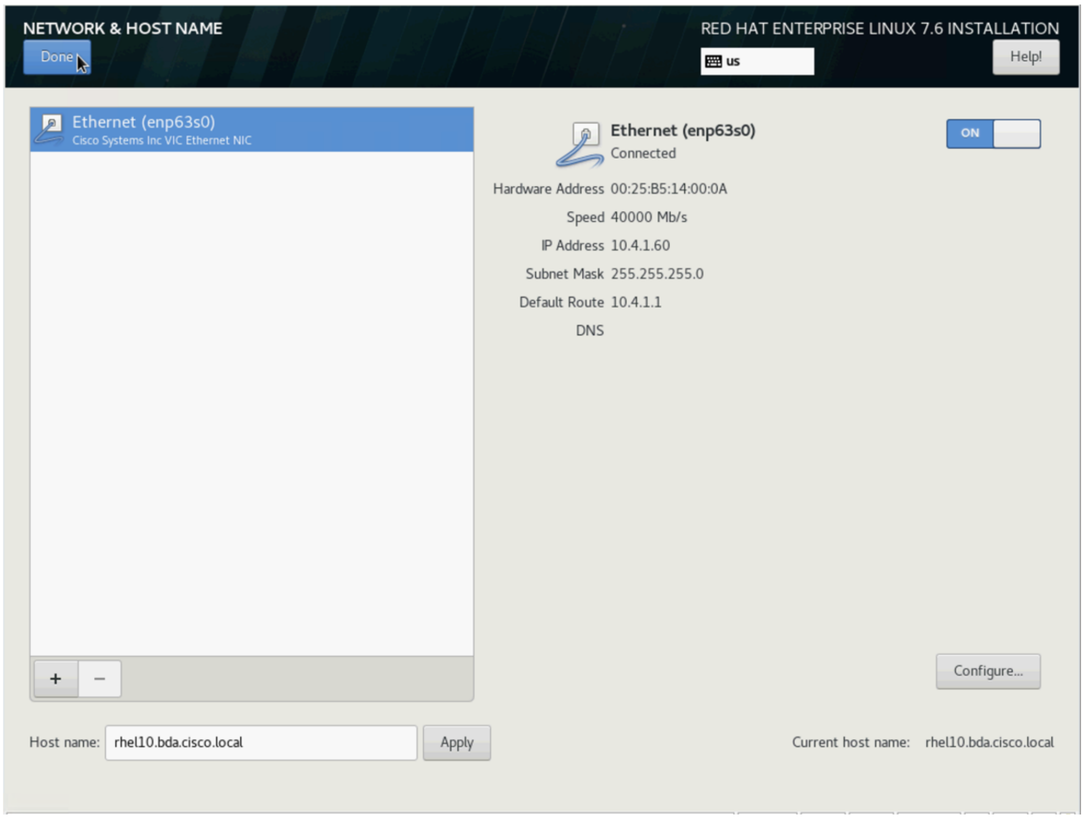
DHCP client ID:

[✓] Require IPv4 addressing for this connection to complete

Host name: rhel10.bda.cisco.local

Current host name: rhel10.bda.cisco.local

15. Click OFF to turn ON the network adapter. Click Done.



Ethernet (enp63s0)
Cisco Systems Inc VIC Ethernet NIC

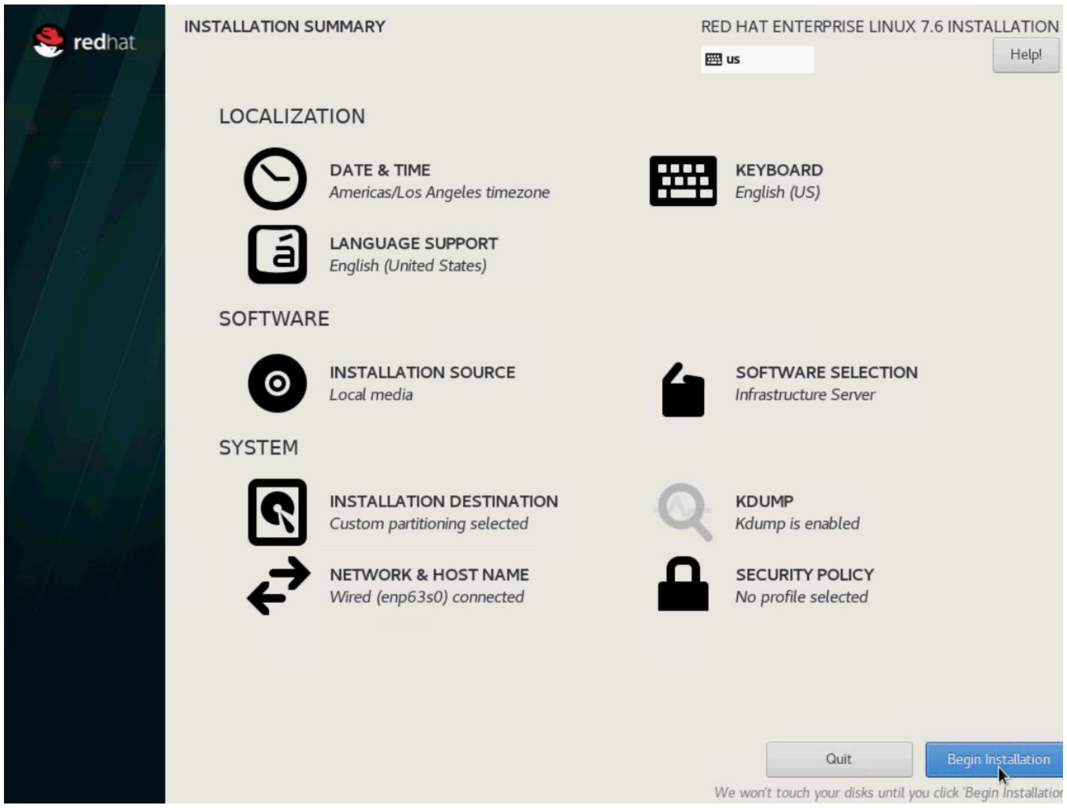
Ethernet (enp63s0)
Connected

Hardware Address	00:25:B5:14:00:0A
Speed	40000 Mb/s
IP Address	10.4.1.60
Subnet Mask	255.255.255.0
Default Route	10.4.1.1
DNS	

Host name: rhel10.bda.cisco.local

Current host name: rhel10.bda.cisco.local

16. Click Begin Installation.



INSTALLATION SUMMARY

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

LOCALIZATION

DATE & TIME
Americas/Los Angeles timezone

KEYBOARD
English (US)

LANGUAGE SUPPORT
English (United States)

SOFTWARE

INSTALLATION SOURCE
Local media

SOFTWARE SELECTION
Infrastructure Server

SYSTEM

INSTALLATION DESTINATION
Custom partitioning selected

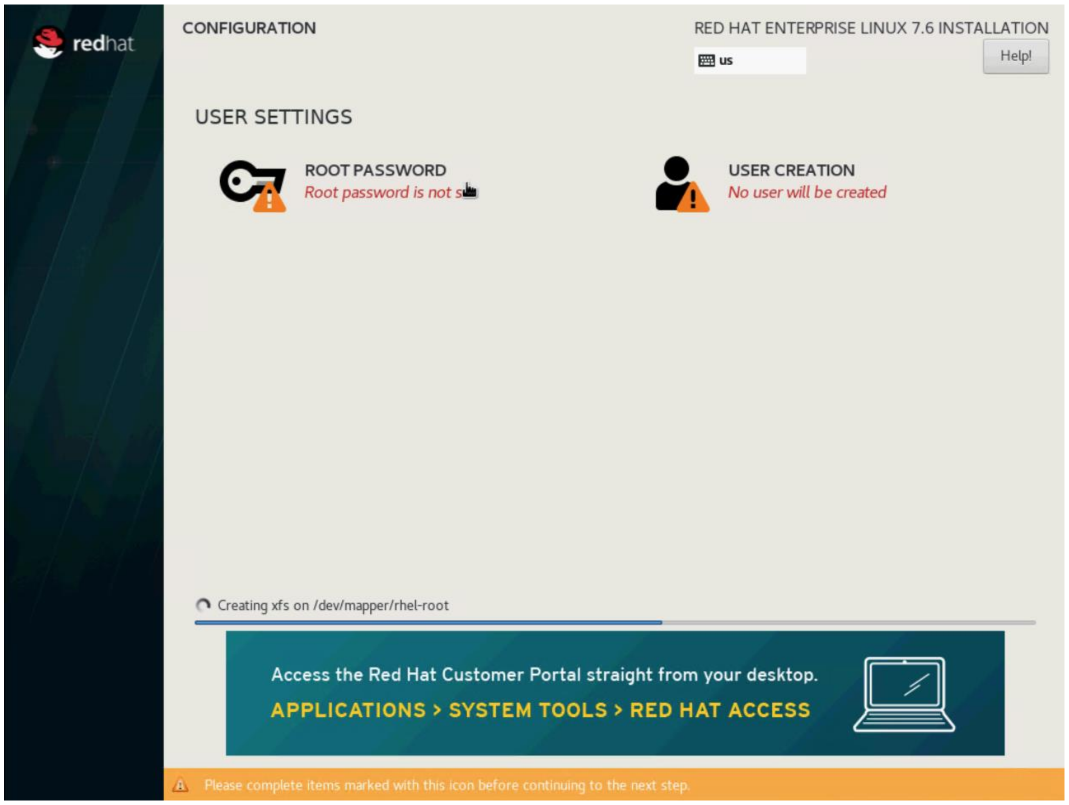
KDUMP
Kdump is enabled

NETWORK & HOST NAME
Wired (enp63s0) connected

SECURITY POLICY
No profile selected

We won't touch your disks until you click 'Begin Installatio

17. Click ROOT PASSWORD.

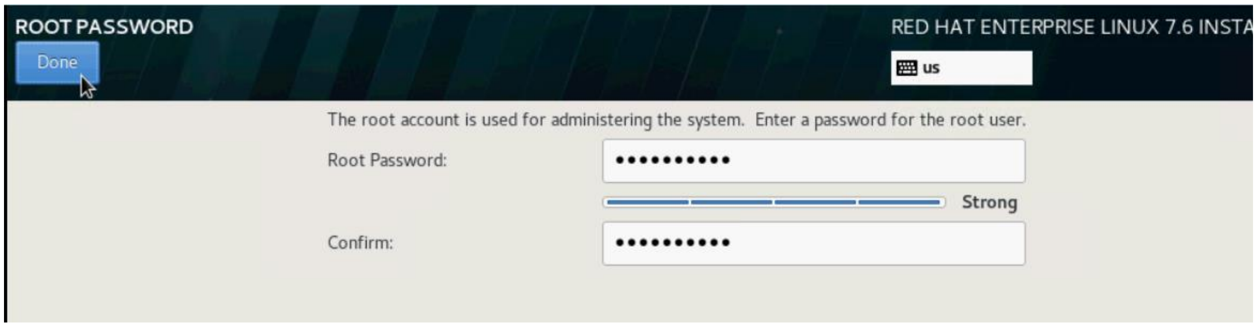


Creating xfs on /dev/mapper/rhel-root

Access the Red Hat Customer Portal straight from your desktop.
APPLICATIONS > SYSTEM TOOLS > RED HAT ACCESS

Please complete items marked with this icon before continuing to the next step.

18. Enter Root Password. Click Done.



The root account is used for administering the system. Enter a password for the root user.

Root Password: ●●●●●●●●●●●

Strong

Confirm: ●●●●●●●●●●●

19. Reboot when installation process completes.

Configure Data Drives on Name Node and Other Management Nodes, in the Appendix.

Configure the Filesystem for NameNodes and DataNodes

The following script formats and mounts the available volumes on each node whether it is NameNode or Data node. OS boot partition will be skipped. All drives are mounted based on their UUID as /data/disk1, /data/disk2, etc. To configure the filesystem for NameNodes and DataNodes, follow these steps:

1. On the Admin node, create a file containing the following script:

```
#vi /root/driveconf.sh
```

2. To create partition tables and file systems on the local disks supplied to each of the nodes, run the following script as the root user on each node:

This script assumes there are no partitions already existing on the data volumes. If there are partitions, delete them before running the script. This process is documented in section Delete Partitions.

```
#vi /root/driveconf.sh
#!/bin/bash
[[ "-x" == "${1}" ]] && set -x && set -v && shift 1
count=1
for X in /sys/class/scsi_host/host?/scan
do
echo '- - -' > ${X}
done
for X in /dev/sd?
do
list+=$(echo $X " ")
done
for X in /dev/sd??
do
list+=$(echo $X " ")
done
for X in $list
do
echo "========"
echo $X
echo "========"
if [[ -b ${X} && `/sbin/parted -s ${X} print quit|/bin/grep -c boot` -
ne 0
]]
then
echo "$X bootable - skipping."
continue
else
Y=${X##*/}1
echo "Formatting and Mounting Drive => ${X}"
166
/sbin/mkfs.xfs –f ${X}
(( $? )) && continue
#Identify UUID
UUID=`blkid ${X} | cut -d " " -f2 | cut -d "=" -f2 | sed 's/"//g'`
/bin/mkdir -p /data/disk${count}
(( $? )) && continue
echo "UUID of ${X} = ${UUID}, mounting ${X} using UUID on
/data/disk${count}"
/bin/mount -t xfs -o inode64,noatime,nobarrier -U ${UUID}
/data/disk${count}
(( $? )) && continue
echo "UUID=${UUID} /data/disk${count} xfs inode64,noatime,nobarrier 0
0" >> /etc/fstab
((count++))
fi
done
```

3. Run the following command to copy driveconf.sh to all the nodes:

```
# chmod 755 /root/driveconf.sh
# ansible datanodes -m copy -a "src=/root/driveconf.sh dest=/root/."
# ansible nodes -m file -a "dest=/root/driveconf.sh mode=755"
```

4. Run the following command from the admin node to run the script across all data nodes:

```
# ansible datanodes -m shell -a "/root/driveconf.sh"
```

5. Run the following from the admin node to list the partitions and mount points:

```
# ansible datanodes -m shell -a "df -h"
# ansible datanodes -m shell -a "mount"
# ansible datanodes -m shell -a "cat /etc/fstab"
```

Delete Partitions

To delete a partition, follow these steps:

1. Run the mount command (`'mount'`) to identify which drive is mounted to which device /dev/sd<?>

2. `umount` the drive for which partition is to be deleted and run `fdisk` to delete as shown below.

Be sure **not to delete the OS partition** since this will wipe out the OS.

```
# mount
# umount /data/disk1 ← (disk1 shown as example)
#(echo d; echo w;) | sudo fdisk /dev/sd<?>
```

Cluster Verification

This section explains the steps to create the script `cluster_verification.sh` that helps to verify the CPU, memory, NIC, and storage adapter settings across the cluster on all nodes. This script also checks additional prerequisites such as NTP status, SELinux status, ulimit settings, JAVA_HOME settings and JDK version, IP address and hostname resolution, Linux version and firewall settings.

To verify a cluster, follow these steps:

The following script uses cluster shell (clush) which needs to be installed and configured.

1. Create the script cluster_verification.sh as shown, on the Admin node (rhel01).

```
#vi cluster_verification.sh
#!/bin/bash
shopt -s expand_aliases,
# Setting Color codes
green='\e[0;32m'
red='\e[0;31m'
NC='\e[0m' # No Color
echo -e "${green} === Cisco UCS Integrated Infrastructure for Big Data and Analytics
\ Cluster Verification === ${NC}"
echo ""
echo ""
echo -e "${green} ==== System Information  ==== ${NC}"
echo ""
echo ""
echo -e "${green}System ${NC}"
clush -a -B " `which dmidecode` |grep -A2 '^System Information'"
echo ""
echo ""
echo -e "${green}BIOS ${NC}"
clush -a -B " `which dmidecode` | grep -A3 '^BIOS I'"
echo ""
echo ""
echo -e "${green}Memory ${NC}"
clush -a -B "cat /proc/meminfo | grep -i ^memt | uniq"
echo ""
echo ""
echo -e "${green}Number of Dimms ${NC}"
clush -a -B "echo -n 'DIMM slots: ';  `which dmidecode` |grep -c \
'^[[:space:]]*Locator:'"
clush -a -B "echo -n 'DIMM count is: ';  `which dmidecode` | grep \ "Size"| grep -c
"MB""
clush -a -B " `which dmidecode` | awk '/Memory Device$/,/^$/ {print}' |\ grep -e
'^Mem' -e Size: -e Speed: -e Part | sort -u | grep -v -e 'NO \ DIMM' -e 'No Module
Installed' -e Unknown"
echo ""
echo ""
# probe for cpu info #
echo -e "${green}CPU ${NC}"
clush -a -B "grep '^model name' /proc/cpuinfo | sort -u"
echo ""
clush -a -B "`which lscpu` | grep -v -e op-mode -e ^Vendor -e family -e\ Model: -e
Stepping: -e BogoMIPS -e Virtual -e ^Byte -e '^NUMA node(s)'"
echo ""
echo ""
# probe for nic info #
echo -e "${green}NIC ${NC}"
clush -a -B "`which ifconfig` | egrep '(^e|^p)' | awk '{print \$1}' | \ xargs -l
`which ethtool` | grep -e ^Settings -e Speed"
echo ""
clush -a -B "`which lspci` | grep -i ether"
echo ""
echo ""
# probe for disk info #
echo -e "${green}Storage ${NC}"
clush -a -B "echo 'Storage Controller: '; `which lspci` | grep -i -e \ raid -e
storage -e lsi"
echo ""
clush -a -B "dmesg | grep -i raid | grep -i scsi"
echo ""
clush -a -B "lsblk -id | awk '{print \$1,\$4}'|sort | nl"
echo ""
echo ""

echo -e "${green} ================ Software  ======================= ${NC}"
echo ""
echo ""
echo -e "${green}Linux Release ${NC}"
```

```
clush -a -B "cat /etc/*release | uniq"
echo ""
echo ""
echo -e "${green}Linux Version ${NC}"
clush -a -B "uname -srvm | fmt"
echo ""
echo ""
echo -e "${green}Date ${NC}"
clush -a -B date
echo ""
echo ""
echo -e "${green}NTP Status ${NC}"
clush -a -B "ntpstat 2>&1 | head -1"
echo ""
echo ""
echo -e "${green}SELINUX ${NC}"
clush -a -B "echo -n 'SElinux status: '; grep ^SELINUX= \ /etc/selinux/config 2>&1"
echo ""
echo ""
clush -a -B "echo -n 'CPUspeed Service: ';  `which service` cpuspeed \ status 2>&1"
clush -a -B "echo -n 'CPUspeed Service: '; `which chkconfig` --list \ cpuspeed 2>&1"
echo ""
echo ""
echo -e "${green}Java Version${NC}"
clush -a -B 'java -version 2>&1; echo JAVA_HOME is ${JAVA_HOME:-Not \ Defined!}'
echo ""
echo ""
echo -e "${green}Hostname LoOKup${NC}"
clush -a -B " ip addr show"
echo ""
echo ""
echo -e "${green}Open File Limit${NC}"
clush -a -B 'echo -n "Open file limit(should be >32K): "; ulimit -n'
```

2. Change permissions to executable:

```
# chmod 755 cluster_verification.sh
```

3. Run the Cluster Verification tool from the admin node. This can be run before starting Hadoop to identify any discrepancies in Post OS Configuration between the servers or during troubleshooting of any cluster / Hadoop issues:

```
#./cluster_verification.sh
```

Install Cloudera

Cloudera's Distribution including Apache Hadoop (CDH) is an enterprise grade, hardened Hadoop distribution. CDH offers Apache Hadoop and several related projects into a single tested and certified product. It offers the latest innovations from the open source community with the testing and quality expected from enterprise quality software.

Prerequisites for CDH Installation

This section details the prerequisites for the CDH installation, such as setting up CDH Repo.

Cloudera Manager Repository

1. From a host connected to the Internet, download the Cloudera's repositories as shown below and transfer it to the admin node:

```
#mkdir -p /tmp/clouderarepo/
```

2. Download Cloudera Manager Repository:

```
#cd /tmp/clouderarepo/
# wget https://archive.cloudera.com/cm6/6.2.0/redhat7/yum/cloudera-manager.repo
#reposync --config=./cloudera-manager.repo --repoid=cloudera-manager
# wget https://archive.cloudera.com/cm6/6.2.0/allkeys.asc
# scp allkeys.asc rhel01:/var/www/html/clouderarepo/cloudera-manager
```

This downloads the Cloudera Manager RPMs needed for the Cloudera repository.

3. Run the following command to move the RPMs:

4. Copy the repository directory to the admin node (rhel1):

```
#scp -r /tmp/clouderarepo/ rhel1:/var/www/html/
```

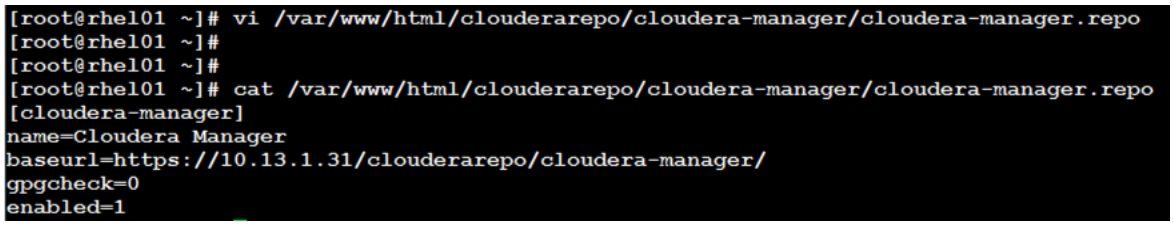
5. On admin node (rhel1) run create repo command:

```
#cd /var/www/html/clouderarepo/
#createrepo --baseurl http://10.13.1.31/clouderarepo/cloudera-manager/
/var/www/html/clouderarepo/cloudera-manager
```

Go to: http://10.13.1.31/clouderarepo/ to verify the files.

6. Create the Cloudera Manager repo file with following contents:

```
#vi /var/www/html/clouderarepo/cloudera-manager/cloudera-manager.repo
[cloudera-manager]
name=Cloudera Manager
baseurl=http://10.13.1.31/clouderarepo/cloudera-manager/
gpgcheck=0
enabled=1
```



```
[root@rhel01 ~]# vi /var/www/html/clouderarepo/cloudera-manager/cloudera-manager.repo
[root@rhel01 ~]#
[root@rhel01 ~]#
[root@rhel01 ~]# cat /var/www/html/clouderarepo/cloudera-manager/cloudera-manager.repo
[cloudera-manager]
name=Cloudera Manager
baseurl=https://10.13.1.31/clouderarepo/cloudera-manager/
gpgcheck=0
enabled=1
```

7. Copy the file cloudera-manager.repo into /etc/yum.repos.d/ on the admin node to enable it to find the packages that are locally hosted:

```
#cp /var/www/html/clouderarepo/cloudera-manager/cloudera-manager.repo
/etc/yum.repos.d/
From the admin node copy the repo files to /etc/yum.repos.d/ of all the nodes of the
cluster:
# ansible all -m copy -a "src=/etc/yum.repos.d/cloudera-manager.repo
dest=/etc/yum.repos.d/."
```

Set Up the Local Parcels for CDH 6.2.0

From a host connected the internet, download the appropriate CDH 6.2.0 parcels that are meant for RHEL7.6 from the URL: http://archive.cloudera.com/cdh6/6.2.0/parcels/ and place them in the directory " /var/www/html/CDH6.2.0parcels" of the Admin node.

The following are the required files for RHEL7.6:

- CDH-6.2.0-1.cdh6.2.0.p0.967373-el7.parcel
- CDH-6.2.0-1.cdh6.2.0.p0.967373-el7.parcel.sha1
- manifest.json

Download Parcels

To download parcels, follow these steps:

1. From a host connected to the Internet, download the Cloudera's parcels as shown below and transfer it to the admin node:

```
#mkdir -p /tmp/clouderarepo/CDH6.2.0parcels
```

2. Download parcels:

```
#cd /tmp/clouderarepo/CDH6.2.0parcels
# wget https://archive.cloudera.com/cdh6/6.2.0/parcels/CDH-6.2.0-
1.cdh6.2.0.p0.967373-el7.parcel
# wget https://archive.cloudera.com/cdh6/6.2.0/parcels/CDH-6.2.0-
1.cdh6.2.0.p0.967373-el7.parcel.sha1
# wget https://archive.cloudera.com/cdh6/6.2.0/parcels/manifest.json
```

3. Now edit the /tmp/clouderarepo/CDH6.2.0parcels/manifest.json file and remove the scripts that are not meant for RHEL7.6. Below is that script which can be copy and pasted.

Please make sure the script starts and end with initial additional braces.

```
{
    "lastUpdated": 1552549852993,
    "parcels": [
        {
            "components": [
                {
                    "name": "hue",
                    "pkg_release": "967373",
                    "pkg_version": "4.2.0+cdh6.2.0",
                    "version": "4.2.0-cdh6.2.0"
                },
                {
                    "name": "hbase-solr",
                    "pkg_release": "967373",
                    "pkg_version": "1.5+cdh6.2.0",
                    "version": "1.5-cdh6.2.0"
                },
                {
                    "name": "hbase",
                    "pkg_release": "967373",
                    "pkg_version": "2.1.0+cdh6.2.0",
                    "version": "2.1.0-cdh6.2.0"
                },
                {
                    "name": "zookeeper",
                    "pkg_release": "967373",
                    "pkg_version": "3.4.5+cdh6.2.0",
                    "version": "3.4.5-cdh6.2.0"
                },
                {
                    "name": "sentry",
                    "pkg_release": "967373",
                    "pkg_version": "2.1.0+cdh6.2.0",
                    "version": "2.1.0-cdh6.2.0"
                },
                {
                    "name": "solr",
                    "pkg_release": "967373",
                    "pkg_version": "7.4.0+cdh6.2.0",
                    "version": "7.4.0-cdh6.2.0"
                },
                {
                    "name": "impala",
                    "pkg_release": "967373",
                    "pkg_version": "3.2.0+cdh6.2.0",
                    "version": "3.2.0-cdh6.2.0"
                },
                {
                    "name": "hadoop-httpfs",
                    "pkg_release": "967373",
                    "pkg_version": "3.0.0+cdh6.2.0",
                    "version": "3.0.0-cdh6.2.0"
                },
                {
                    "name": "oozie",
                    "pkg_release": "967373",
                    "pkg_version": "5.1.0+cdh6.2.0",
                    "version": "5.1.0-cdh6.2.0"
                },
                {
                    "name": "hive-hcatalog",
                    "pkg_release": "967373",
                    "pkg_version": "2.1.1+cdh6.2.0",
                    "version": "2.1.1-cdh6.2.0"
                },
                {
                    "name": "hive",
                    "pkg_release": "967373",
                    "pkg_version": "2.1.1+cdh6.2.0",
                    "version": "2.1.1-cdh6.2.0"
                },
                {
                    "name": "hadoop-mapreduce",
                    "pkg_release": "967373",
                    "pkg_version": "3.0.0+cdh6.2.0",
                    "version": "3.0.0-cdh6.2.0"
                },
                {
                    "name": "parquet",
                    "pkg_release": "967373",
                    "pkg_version": "1.9.0+cdh6.2.0",
                    "version": "1.9.0-cdh6.2.0"
                },
                {
                    "name": "sqoop",
                    "pkg_release": "967373",
                    "pkg_version": "1.4.7+cdh6.2.0",
                    "version": "1.4.7-cdh6.2.0"
                },
                {
                    "name": "hadoop-hdfs",
                    "pkg_release": "967373",
                    "pkg_version": "3.0.0+cdh6.2.0",
                    "version": "3.0.0-cdh6.2.0"
                },
                {
                    "name": "hadoop",
                    "pkg_release": "967373",
                    "pkg_version": "3.0.0+cdh6.2.0",
                    "version": "3.0.0-cdh6.2.0"
                },
                {
                    "name": "pig",
                    "pkg_release": "967373",
                    "pkg_version": "0.17.0+cdh6.2.0",
                    "version": "0.17.0-cdh6.2.0"
                },
                {
                    "name": "kudu",
                    "pkg_release": "967373",
                    "pkg_version": "1.9.0+cdh6.2.0",
                    "version": "1.9.0-cdh6.2.0"
                },
                {
                    "name": "spark",
                    "pkg_release": "967373",
                    "pkg_version": "2.4.0+cdh6.2.0",
                    "version": "2.4.0-cdh6.2.0"
                },
                {
                    "name": "flume-ng",
                    "pkg_release": "967373",
                    "pkg_version": "1.9.0+cdh6.2.0",
                    "version": "1.9.0-cdh6.2.0"
                },
                {
                    "name": "hadoop-kms",
                    "pkg_release": "967373",
                    "pkg_version": "3.0.0+cdh6.2.0",
                    "version": "3.0.0-cdh6.2.0"
                },
                {
                    "name": "kafka",
                    "pkg_release": "967373",
                    "pkg_version": "2.1.0+cdh6.2.0",
                    "version": "2.1.0-cdh6.2.0"
                },
                {
                    "name": "kite",
                    "pkg_release": "967373",
                    "pkg_version": "1.0.0+cdh6.2.0",
                    "version": "1.0.0-cdh6.2.0"
                },
                {
                    "name": "hadoop-yarn",
                    "pkg_release": "967373",
                    "pkg_version": "3.0.0+cdh6.2.0",
                    "version": "3.0.0-cdh6.2.0"
                }
            ],
            "hash": "e9c8328d8c370517c958111a3db1a085ebace237",
            "parcelName": "CDH-6.2.0-1.cdh6.2.0.p0.967373-el7.parcel",
            "replaces": "IMPALA, SOLR, SPARK, KAFKA, IMPALA_KUDU, KUDU, SPARK2"
        }
    ]
}
```

4. Copy /tmp/clouderarepo/CDH6.2.0parcels to the admin node (rhel01):

```
#scp -r /tmp/clouderarepo/CDH6.2.0parcels/ rhel01:/var/www/html/
```

5. Verify that these files are accessible by visiting the URL http://10.13.1.31/CDH6.2.0parcels/ in admin node.

Set Up the MariaDB Database for Cloudera Manager

You will set up the following for Cloudera Manager:

- Install the MariaDB Server
- Configure and Start the MariaDB Server
- Install the MariaDB/MySQL JDBC Driver
- Create Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server

Install the MariaDB Server

To use a MariaDB database, follow these steps:

1. In the admin node where Cloudera Manager will be installed, use the following command to install the mari-adb/mysql server.

```
#yum -y install mariadb-server
```

2. To configure and start the MySQL Server, stop the MariaDB server if it is running.

```
#service mariadb stop
```

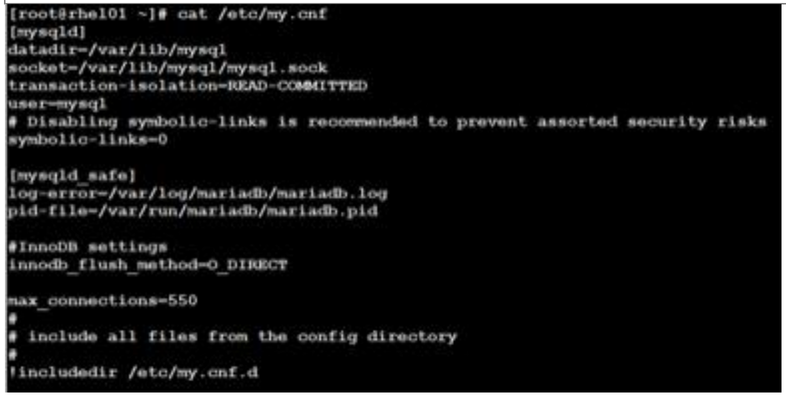
3. Move the old InnoDB log if exists.

4. Move files /var/lib/mysql/ib_logfile0 and /var/lib/mysql/ib_logfile1 out of/var/lib/mysql/ to a backup location.

```
#mv /var/lib/mysql/ib_logfile0 /root/ib_logfile0.bkp
#mv /var/lib/mysql/ib_logfile1 /root/ib_logfile1.bkp
```

5. Determine the location of the option file, my.cnf and edit/add following lines:

```
#vi /etc/my.cnf
[mysqld]
transaction-isolation = READ-COMMITTED
# InnoDB settings
innodb_flush_method = O_DIRECT
max_connections = 550
```



```
[root@rhel01 ~]# cat /etc/my.cnf
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
transaction-isolation=READ-COMMITTED
user=mysql
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid

#InnoDB settings
innodb_flush_method=O_DIRECT

max_connections=550
#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
```

The max_connections need to be increased based on number of nodes and applications. Please follow the recommendations as mentioned in the Cloudera document: https://www.cloudera.com/documentation/enterprise/6/6.2/topics/install_cm_mariadb.html#install_cm_mariadb.

6. Make sure MySQL Server starts at boot:

```
# systemctl enable mariadb.service
```

7. Start the MySQL Server:

```
# systemctl start mariadb.service
```

8. Set the MySQL root password on admin node (rhel01)

```
#cd /usr/bin/
#mysql_secure_installation
```

Install the MySQL JDBC Driver

Install the JDBC driver on the Cloudera Manager Server host, as well as any hosts that run the Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server roles.

To install the MySQL JDBC driver, follow these steps:

1. From a host connected to the Internet, download the MySQL JDBC Driver and transfer it to the admin node. Download the MySQL JDBC driver from the URL http://www.mysql.com/downloads/connector/j/5.1.html.

2. Copy mysql-connector-java-5.1.47.tar.gz to admin node(rhel01)

```
#scp mysql-connector-java-5.1.47.tar.gz rhel01:/root/
```

3. Log in to the admin node and extract the file:

```
#tar xzvf mysql-connector-java-5.1.47.tar.gz
```

4. Create the /usr/share/java/ directory on the admin node (rhel01):

```
#mkdir -p /usr/share/java/
```

5. Go to the mysql-connector-java-5.1.47 directory on the admin node (rhel1) and copy mysql-connector-java-5.1.47-bin.jar to /usr/share/java/:

```
#cd mysql-connector-java-5.1.47
#cp mysql-connector-java-5.1.47-bin.jar /usr/share/java/mysql-connector-java.jar
```

6. Creating Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Navigator Audit Server, and Navigator Metadata Server:

7. In the admin node Log into MySQL as the root user:

```
#mysql -u root -p
```

8. Enter the password that was previously supplied.

9. Create databases for the Activity Monitor, Reports Manager and Hive Metastore Server using the command below:

```
MariaDB [(none)]> create database scm DEFAULT CHARACTER SET utf8;
MariaDB [(none)]> create database amon DEFAULT CHARACTER SET utf8;
MariaDB [(none)]> create database rman DEFAULT CHARACTER SET utf8;
MariaDB [(none)]> create database metastore DEFAULT CHARACTER SET utf8;
MariaDB [(none)]> create database nav DEFAULT CHARACTER SET utf8;
MariaDB [(none)]> create database navms DEFAULT CHARACTER SET utf8;
MariaDB [(none)]> create database sentry DEFAULT CHARACTER SET utf8;
MariaDB [(none)]> create database oozie DEFAULT CHARACTER SET utf8;
MariaDB [(none)]> create database hue DEFAULT CHARACTER SET utf8;
MariaDB [(none)]> grant all on scm.*TO 'root'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on rman.*TO 'root'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on metastore.*TO 'root'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on amon.*TO 'root'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on nav.*TO 'root'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on navms.*TO 'root'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on sentry.*TO 'root'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all privileges on oozie.*TO 'root'@'%' IDENTIFIED BY
'password';
MariaDB [(none)]> grant all on hue.*TO 'root'@'%' IDENTIFIED BY 'password';

MariaDB [(none)]> grant all on scm.*TO 'scm'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on rman.*TO 'rman'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on metastore.*TO 'hive'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on amon.*TO 'amon'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on nav.*TO 'nav'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on navms.*TO 'navms'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all on sentry.*TO 'root'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> grant all privileges on oozie.*TO 'oozie'@'%' IDENTIFIED BY
'password';
MariaDB [(none)]> grant all on hue.*TO 'hue'@'%' IDENTIFIED BY 'password';
MariaDB [(none)]> quit
```

Cloudera Manager Installation

The following sections describe how to install Cloudera Manager and then using Cloudera Manager to install CDH 6.2.0.

Set Up the Cloudera Manager Server Database

The Cloudera Manager Server Database stores information about service and host configurations.

Install Cloudera Manager

Cloudera Manager, an end to end management application, is used to install and configure CDH. During CDH Installation, Cloudera Manager's Wizard will help to install Hadoop services on all nodes using the following procedure:

- Discovery of the cluster nodes
- Configure the Cloudera parcel or package repositories
- Install Hadoop, Cloudera Manager Agent (CMA) and Impala on all the cluster nodes.
- Install the Oracle JDK if it is not already installed across all the cluster nodes.
- Assign various services to nodes.
- Start the Hadoop services

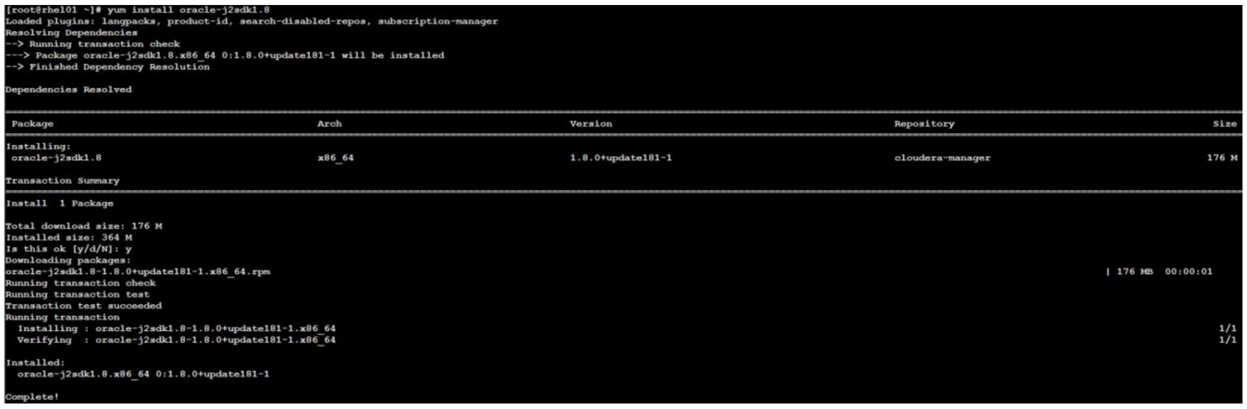
To install Cloudera Manager, follow these steps:

1. Update the repo files to point to local repository.

```
#rm -f /var/www/html/clouderarepo/*.repo
#cp /etc/yum.repos.d/c*.repo /var/www/html/clouderarepo/
```

2. Install the Oracle Java Development Kit on the Cloudera Manager Server host.

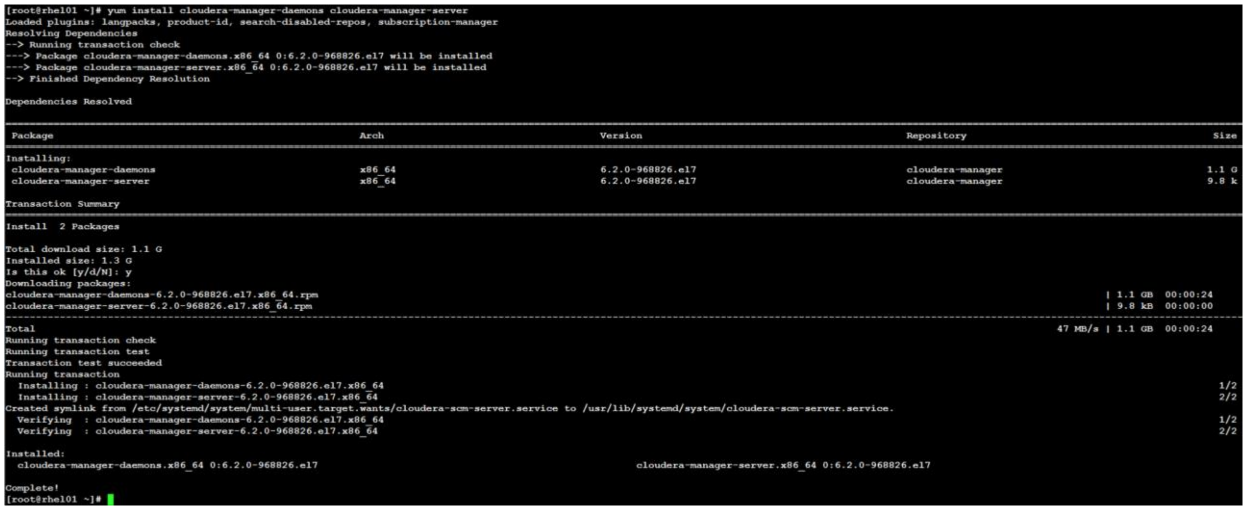
```
#yum install oracle-j2sdk1.8
```



```
[root@rhel01 ~]# yum install oracle-j2sdk1.8
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
---> Package oracle-j2sdk1.8.x86_64 0:1.8.0+update181-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                          Arch                       Version                          Repository                   Size
================================================================================================================================
Installing:
 oracle-j2sdk1.8                  x86_64                     1.8.0+update181-1                cloudera-manager            176 M

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

Total download size: 176 M
Installed size: 364 M
Is this ok [y/d/N]: y
Downloading packages:
oracle-j2sdk1.8-1.8.0+update181-1.x86_64.rpm                                                         | 176 MB  00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : oracle-j2sdk1.8-1.8.0+update181-1.x86_64                                                                  1/1
  Verifying  : oracle-j2sdk1.8-1.8.0+update181-1.x86_64                                                                  1/1

Installed:
  oracle-j2sdk1.8.x86_64 0:1.8.0+update181-1

Complete!
```

3. Install the Cloudera Manager Server packages either on the host where the database is installed, or on a host that has access to the database:

```
#yum install cloudera-manager-daemons cloudera-manager-server
```



```
[root@rhel01 ~]# yum install cloudera-manager-daemons cloudera-manager-server
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
---> Package cloudera-manager-daemons.x86_64 0:6.2.0-968826.el7 will be installed
---> Package cloudera-manager-server.x86_64 0:6.2.0-968826.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                            Arch                   Version                        Repository                     Size
================================================================================================================================
Installing:
 cloudera-manager-daemons           x86_64                 6.2.0-968826.el7               cloudera-manager              1.1 G
 cloudera-manager-server            x86_64                 6.2.0-968826.el7               cloudera-manager              9.8 k

Transaction Summary
================================================================================================================================
Install  2 Packages

Total download size: 1.1 G
Installed size: 1.3 G
Is this ok [y/d/N]: y
Downloading packages:
cloudera-manager-daemons-6.2.0-968826.el7.x86_64.rpm                                                 | 1.1 GB  00:00:24
cloudera-manager-server-6.2.0-968826.el7.x86_64.rpm                                                  | 9.8 kB  00:00:00
--------------------------------------------------------------------------------------------------------------------------------
Total                                                                                       47 MB/s | 1.1 GB  00:00:24
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : cloudera-manager-daemons-6.2.0-968826.el7.x86_64                                                          1/2
  Installing : cloudera-manager-server-6.2.0-968826.el7.x86_64                                                           2/2
Created symlink from /etc/systemd/system/multi-user.target.wants/cloudera-scm-server.service to /usr/lib/systemd/system/cloudera-scm-server.service.
  Verifying  : cloudera-manager-daemons-6.2.0-968826.el7.x86_64                                                          1/2
  Verifying  : cloudera-manager-server-6.2.0-968826.el7.x86_64                                                           2/2

Installed:
  cloudera-manager-daemons.x86_64 0:6.2.0-968826.el7                  cloudera-manager-server.x86_64 0:6.2.0-968826.el7

Complete!
[root@rhel01 ~]#
```

Prepare a Cloudera Manager Server External Database

To prepare a Cloudera Manager Server external database, follow these steps:

1. Run the scm_prepare_database.sh script on the host where the Cloudera Manager Server package is installed (rhel1) admin node:

```
# cd /opt/cloudera/cm/schema/
#./scm_prepare_database.sh mysql scm root <password>
#./scm_prepare_database.sh mysql amon root <password>
```

```
#./scm_prepare_database.sh mysql rman root <password>
#./scm_prepare_database.sh mysql metastore root <password>
#./scm_prepare_database.sh mysql nav root <password>
#./scm_prepare_database.sh mysql navms root <password>
#./scm_prepare_database.sh mysql sentry root <password>
#./scm_prepare_database.sh mysql oozie root <password>
#./scm_prepare_database.sh mysql hue root <password>
```

2. Verify the database connectivity using the following command.

```
[root@rhel01 ~]# mysql -u root -p
mysql> connect scm
mysql> connect amon
mysql> connect rman
mysql> connect metastore
mysql> connect nav
mysql> connect navms
mysql> connect sentry
mysql> connect oozie
mysql> connect hue
```

The MySQL External database setup is complete.

Start the Cloudera Manager Server

To start the Cloudera Manager Server, follow these steps:

1. Start the Cloudera Manager Server:

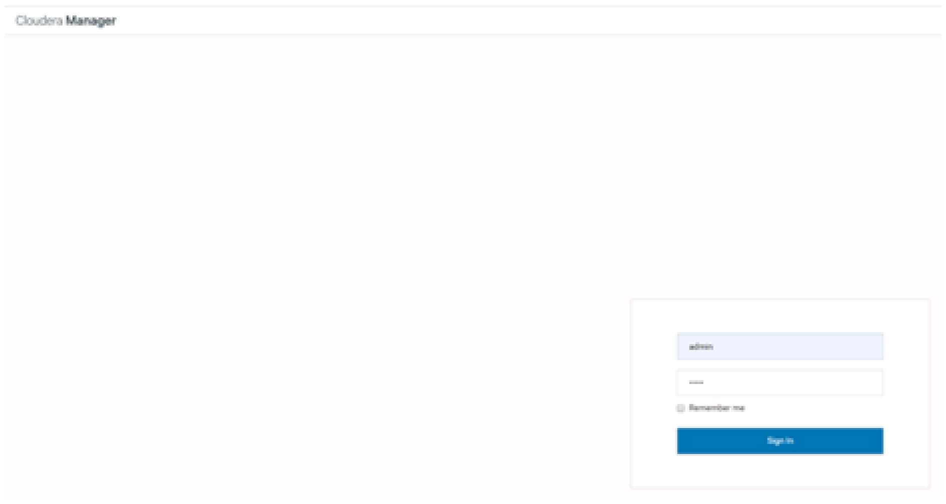
```
#service cloudera-scm-server start
Access the Cloudera Manager using the URL, http://10.13.1.31:7180 to verify that the
server is up.
```

2. Once the installation of Cloudera Manager is complete, install CDH5 using the Cloudera Manager Web interface.

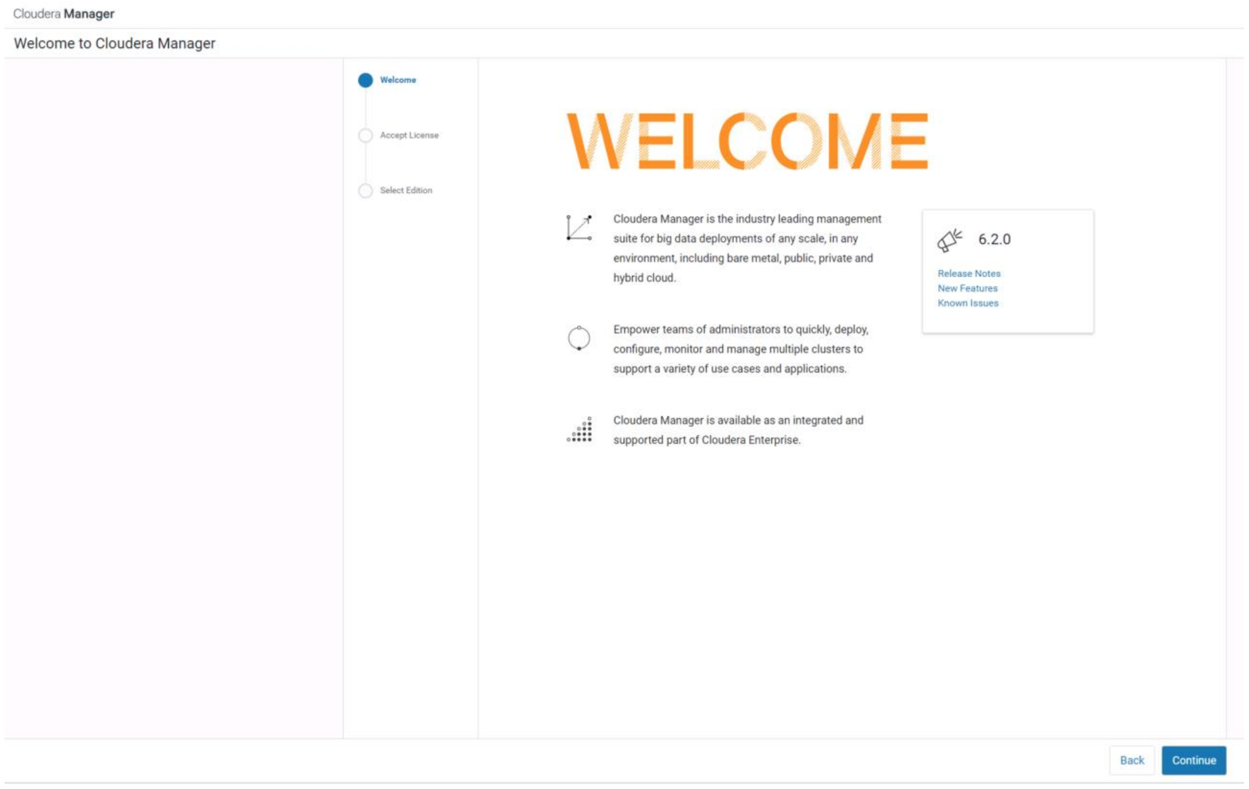
Install Cloudera Enterprise Data Hub (CDH6)

To install the Cloudera Enterprise Data Hub, follow these steps:

1. Log into the Cloudera Manager. Enter " admin" for both the Username and Password fields.



2. Click Continue.



Cloudera Manager is the industry leading management suite for big data deployments of any scale, in any environment, including bare metal, public, private and hybrid cloud.

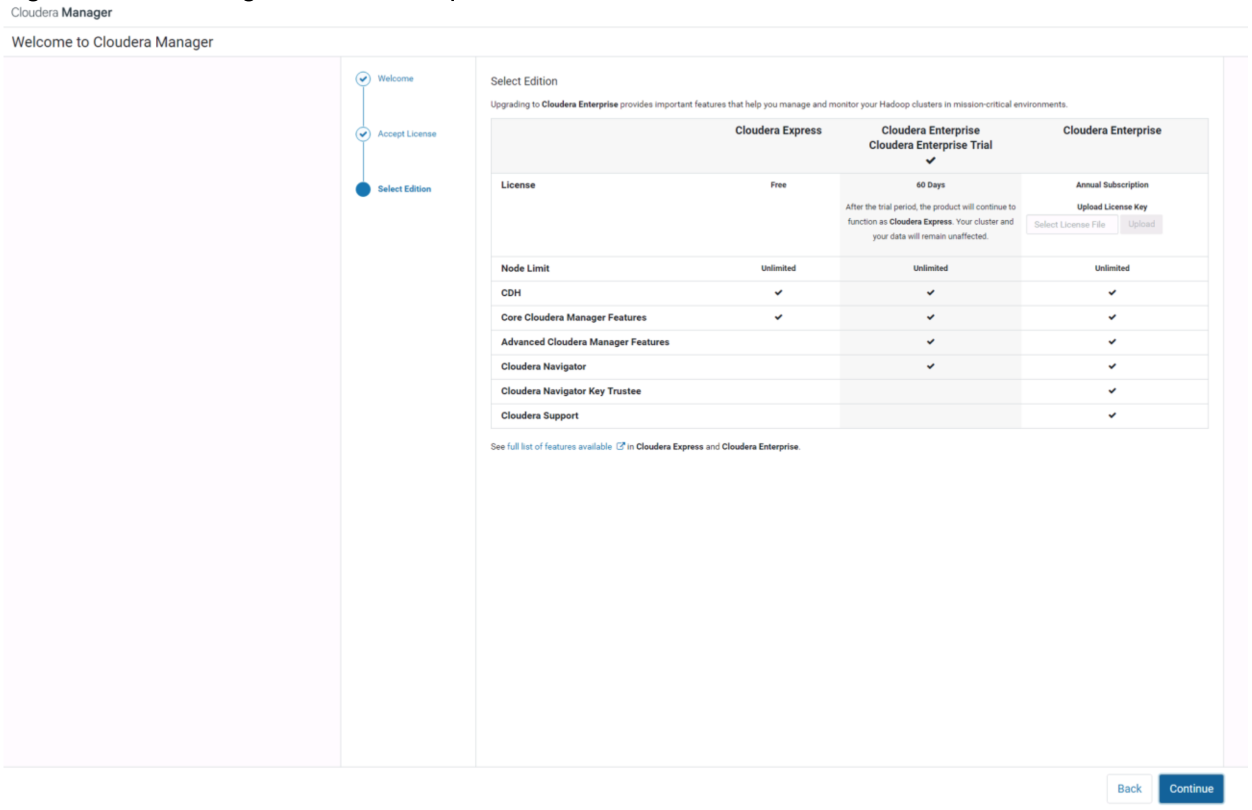
Empower teams of administrators to quickly, deploy, configure, monitor and manage multiple clusters to support a variety of use cases and applications.

Cloudera Manager is available as an integrated and supported part of Cloudera Enterprise.

3. If you do not have a Cloudera license, select Cloudera Enterprise Data Hub Trial Edition. If you do have a Cloudera license, Click "Upload License" and select your license.

4. Based on your requirements, choose the appropriate Cloudera Editions for the Installation.

Figure 62 Installing Cloudera Enterprise



Cloudera Manager

Welcome to Cloudera Manager

Welcome

Accept License

Select Edition

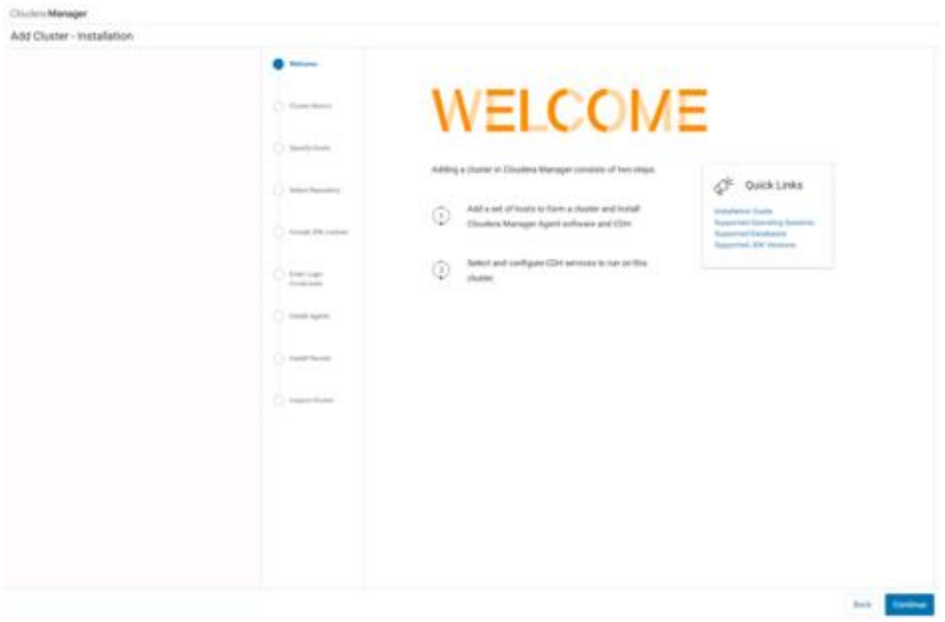
Select Edition

Upgrading to **Cloudera Enterprise** provides important features that help you manage and monitor your Hadoop clusters in mission-critical environments.

	Cloudera Express	Cloudera Enterprise Cloudera Enterprise Trial ✓	Cloudera Enterprise
License	Free	60 Days After the trial period, the product will continue to function as **Cloudera Express**. Your cluster and your data will remain unaffected.	Annual Subscription Upload License Key Select License File Upload
Node Limit	Unlimited	Unlimited	Unlimited
CDH	✓	✓	✓
Core Cloudera Manager Features	✓	✓	✓
Advanced Cloudera Manager Features		✓	✓
Cloudera Navigator		✓	✓
Cloudera Navigator Key Trustee			✓
Cloudera Support			✓

See full list of features available in **Cloudera Express** and **Cloudera Enterprise**.

5. Click Continue.



Cloudera Manager

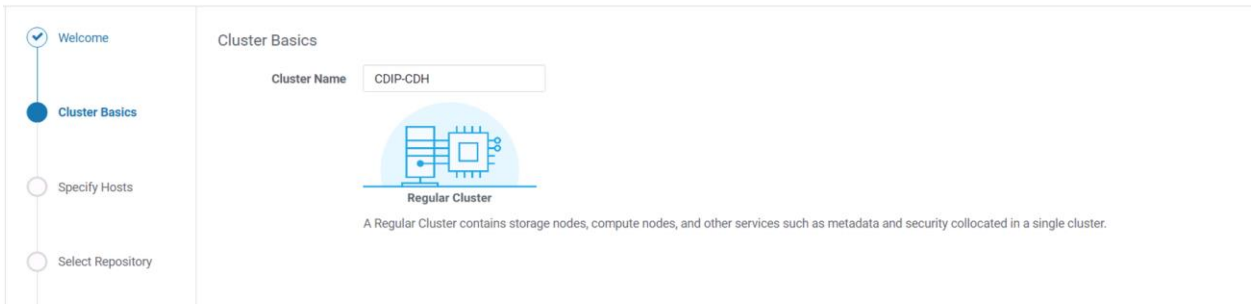
Add Cluster - Installation

WELCOME

Adding a cluster in Cloudera Manager consists of two steps.

1. Add a set of hosts to form a cluster and install Cloudera Manager Agent software and CDH
2. Select and configure CDH services to run on this cluster.

6. Enter a name for the Cluster.



Cluster Basics

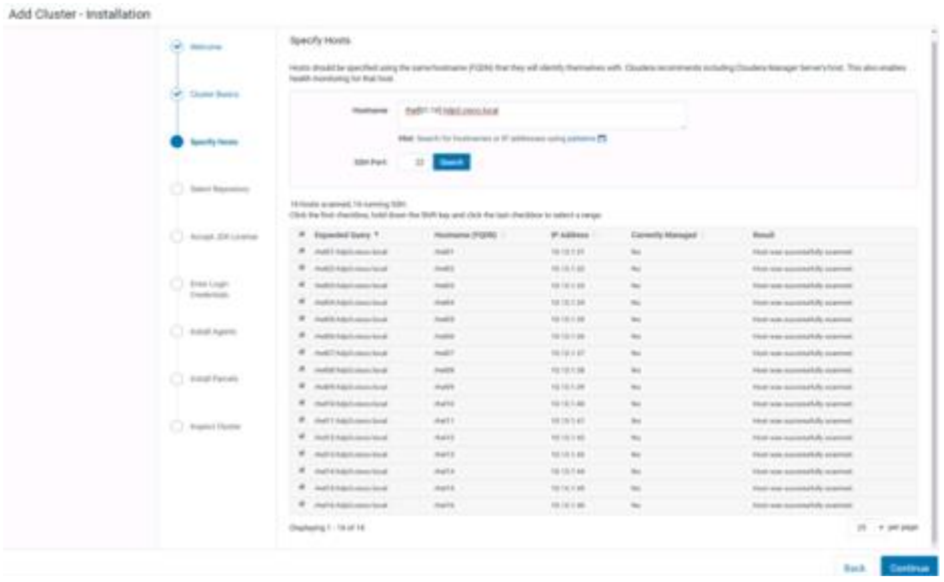
Cluster Name: CDIP-CDH

A Regular Cluster contains storage nodes, compute nodes, and other services such as metadata and security collocated in a single cluster.

7. Specify the hosts that are part of the cluster using their IP addresses or hostname. The figure below shows a pattern that specifies the IP addresses range.

```
10.13.1.[31-60] or rhel[01-30]
```

8. After the IP addresses or hostnames are entered, click Search.



Add Cluster - Installation

Specify Hosts

[illegible]

	Expanded Query	Hostname (FQDN)	IP Address	Currently Managed	Result
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

9. Cloudera Manager will " discover" the nodes in the cluster. Verify that all desired nodes have been found and selected for installation.

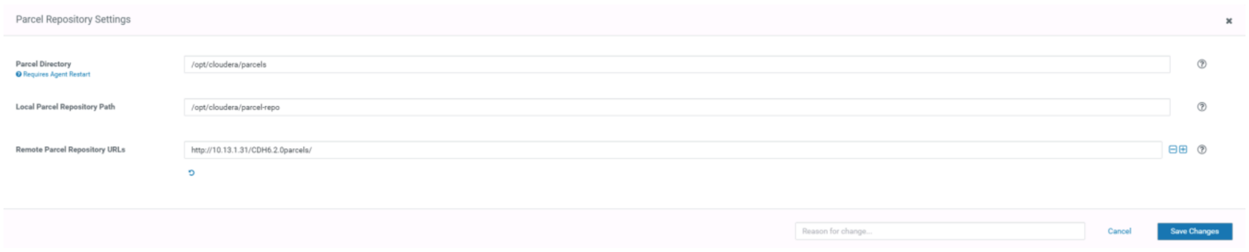
Edit the Cloudera Enterprise Parcel Settings to Use the CDH 5.13.0 Parcels

To edit the Cloudera Enterprise Parcel settings, follow these steps:

1. Open another tab in the same browser window and visit the URL: http://10.13.1.31:7180/cmf/parcel/status for modifying the parcel settings.

2. Click Configuration.

3. Click to remove the entire remote repository URLs and add the URL to the location where we kept the CDH 6.2.0 parcels i.e. http://10.13.1.31/CDH6.2.0parcels/.


Parcel Repository Settings

Parcel Directory (Requires Agent Restart)	/opt/cloudera/parcels
Local Parcel Repository Path	/opt/cloudera/parcel-repo
Remote Parcel Repository URLs	http://10.13.1.31/CDH6.2.0parcels/

4. Click Save Changes to finish the configuration.

5. Navigate back to the Cloudera installation home page, http://10.13.1.31:7180.

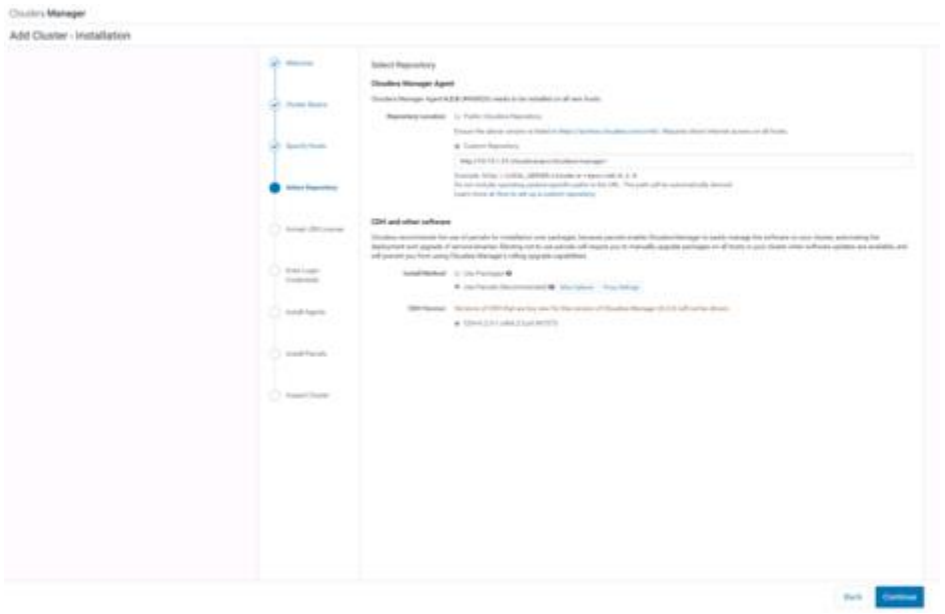
6. Click Continue on the confirmation page.

7. For the method of installation, select the Use Parcels (Recommended) radio button.

8. For the CDH version, select the CDH-6.2.0-1.cdh6.2.0.p0.967373 radio button.

9. For the specific release of Cloudera Manager, select the Custom Repository radio button.

10. Enter the URL for the repository within the admin node. http://10.13.1.50/clouderarepo/cloudera-manager and click Continue.


Cloudera Manager

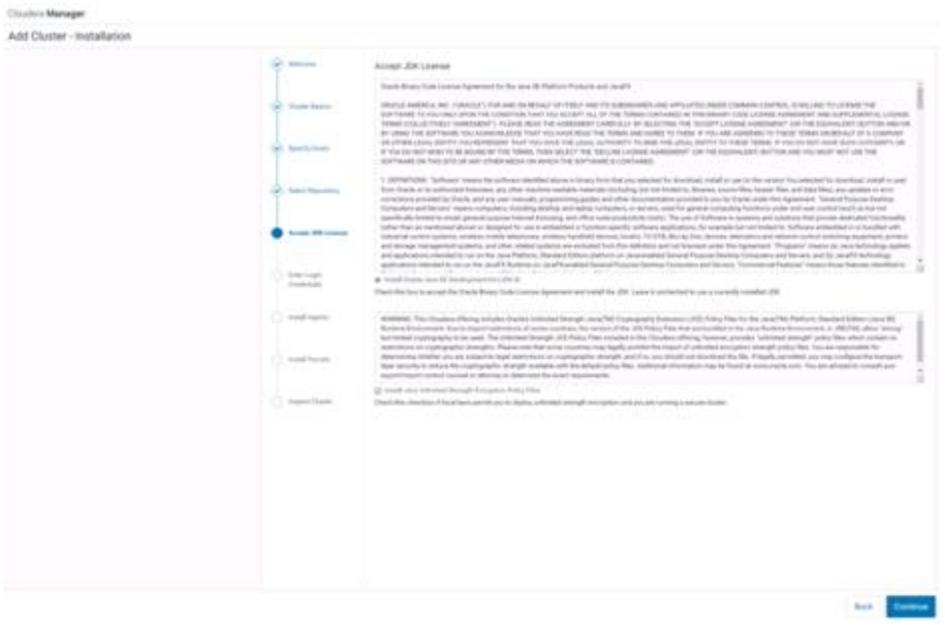
Add Cluster - Installation

Select Repository

Cloudera Manager Agent

CDH and other software

11. Select Install Oracle Java SE Development Kit 9JDK 8). Click Continue.



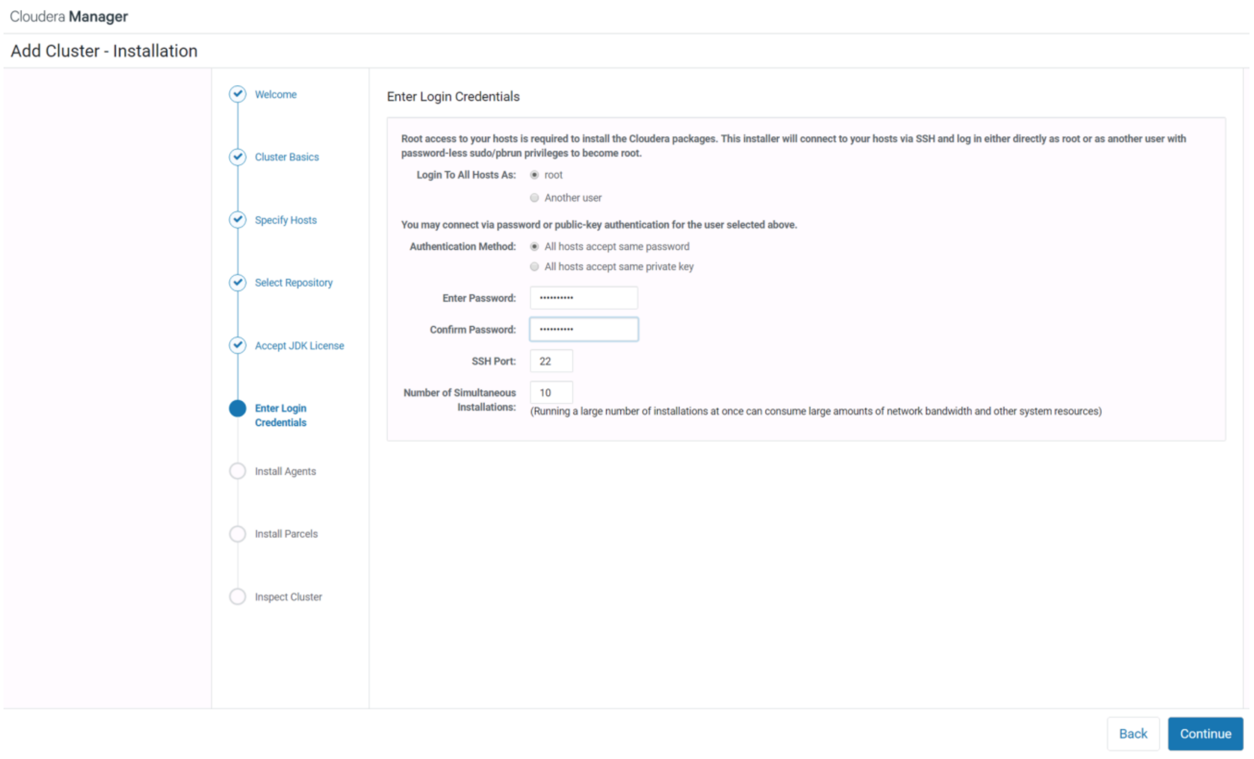
Cloudera **Manager**

Add Cluster - Installation

Accept JDK License

[illegible]

12. Provide SSH login credentials for the cluster and click Continue.



Cloudera **Manager**

Add Cluster - Installation

Enter Login Credentials

Root access to your hosts is required to install the Cloudera packages. This installer will connect to your hosts via SSH and log in either directly as root or as another user with password-less sudo/pbrun privileges to become root.

Login To All Hosts As: [✓ root] Another user

You may connect via password or public-key authentication for the user selected above.

Authentication Method: [✓ All hosts accept same password] All hosts accept same private key

Enter Password: ••••••••••

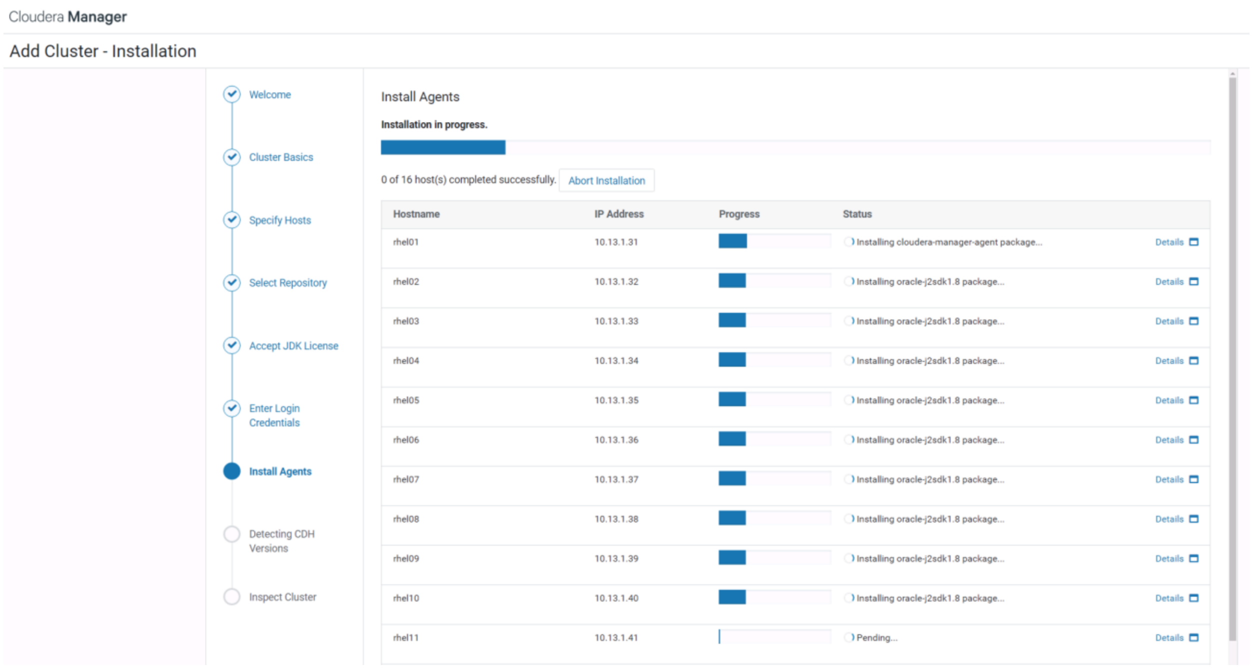
Confirm Password: ••••••••••

SSH Port: 22

Number of Simultaneous Installations: 10

(Running a large number of installations at once can consume large amounts of network bandwidth and other system resources)

Installation using parcels begins.

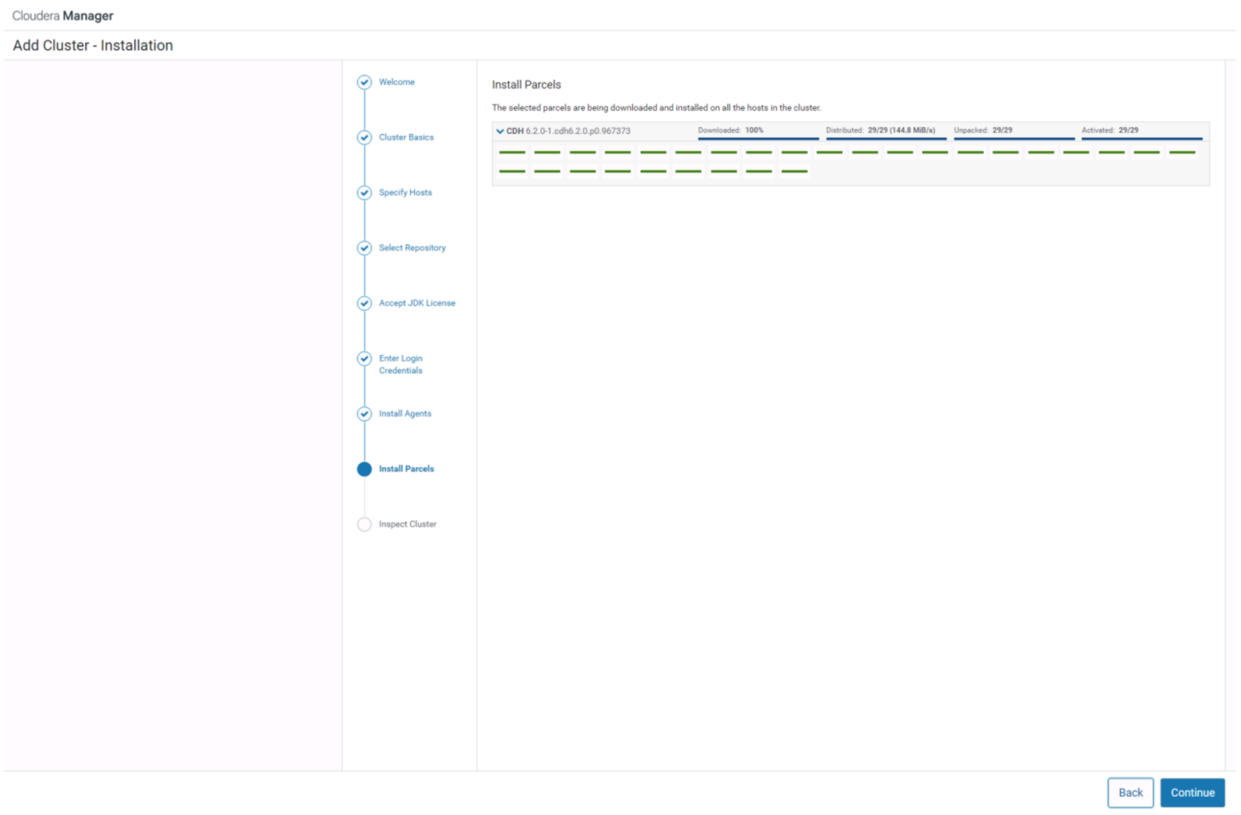


Install Agents

Installation in progress.

0 of 16 host(s) completed successfully.

Hostname	IP Address	Progress	Status
rhel01	10.13.1.31		Installing cloudera-manager-agent package...
rhel02	10.13.1.32		Installing oracle-j2sdk1.8 package...
rhel03	10.13.1.33		Installing oracle-j2sdk1.8 package...
rhel04	10.13.1.34		Installing oracle-j2sdk1.8 package...
rhel05	10.13.1.35		Installing oracle-j2sdk1.8 package...
rhel06	10.13.1.36		Installing oracle-j2sdk1.8 package...
rhel07	10.13.1.37		Installing oracle-j2sdk1.8 package...
rhel08	10.13.1.38		Installing oracle-j2sdk1.8 package...
rhel09	10.13.1.39		Installing oracle-j2sdk1.8 package...
rhel10	10.13.1.40		Installing oracle-j2sdk1.8 package...
rhel11	10.13.1.41		Pending...



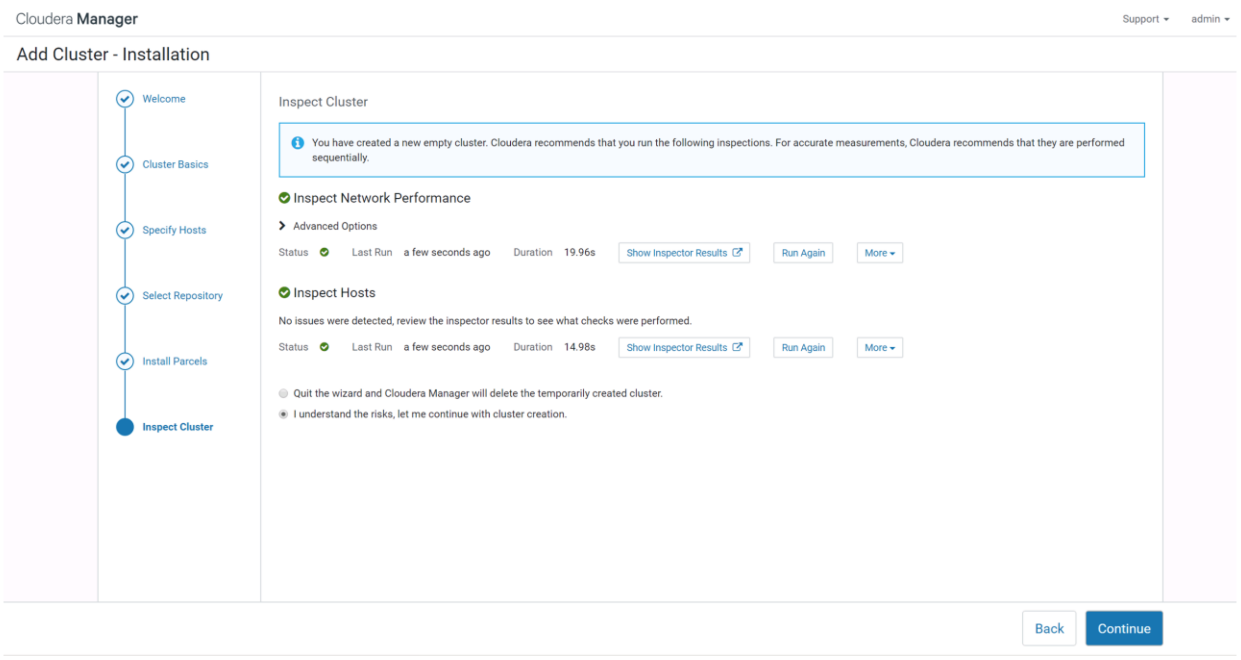
Install Parcels

The selected parcels are being downloaded and installed on all the hosts in the cluster.

CDH 6.2.0-1.cdh6.2.0.p0.967373 Downloaded: 100% Distributed: 29/29 (144.8 MiB/s) Unpacked: 29/29 Activated: 29/29

13. Run inspect the hosts and network performance test through Cloudera Manager on which it has just performed the installation.

14. Review and verify the summary. Click Continue.



Inspect Cluster

You have created a new empty cluster. Cloudera recommends that you run the following inspections. For accurate measurements, Cloudera recommends that they are performed sequentially.

Inspect Network Performance

Advanced Options

Status ✓ Last Run a few seconds ago Duration 19.96s Show Inspector Results Run Again More

Inspect Hosts

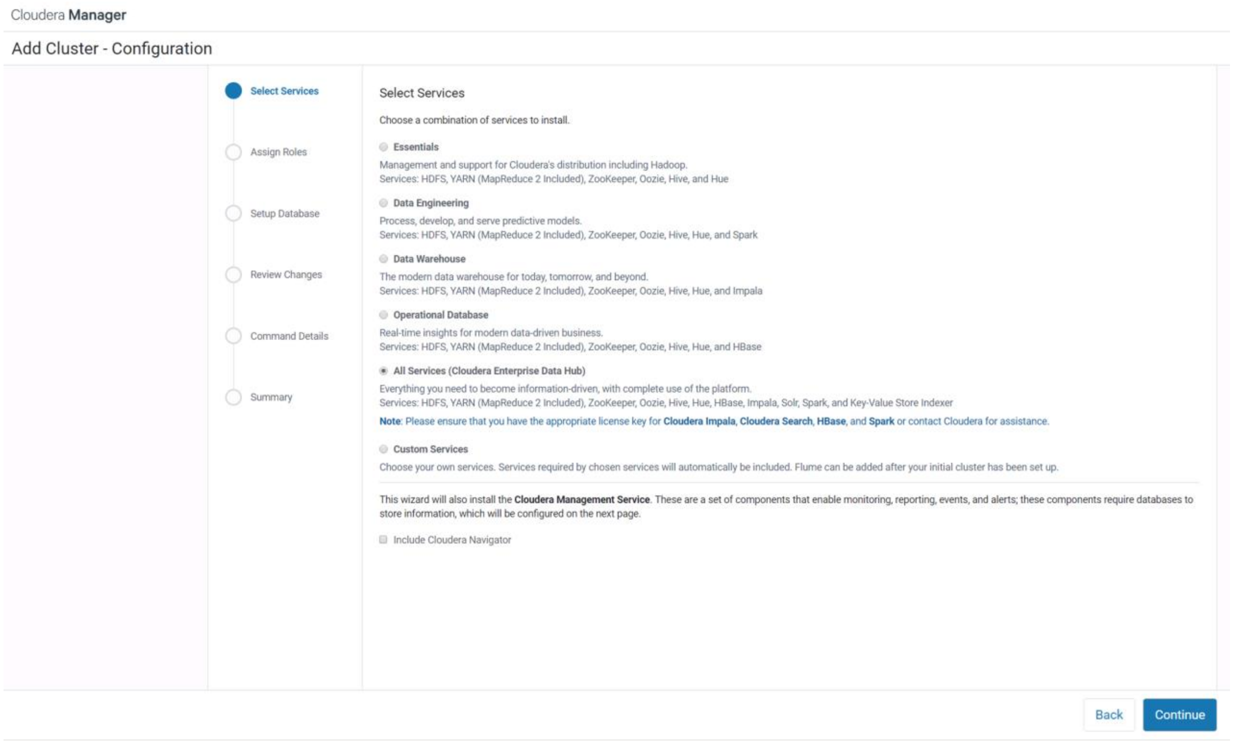
No issues were detected, review the inspector results to see what checks were performed.

Status ✓ Last Run a few seconds ago Duration 14.98s Show Inspector Results Run Again More

Quit the wizard and Cloudera Manager will delete the temporarily created cluster.

[✓ I understand the risks, let me continue with cluster creation.]

15. Select services that need to be started on the cluster.



Select Services

Choose a combination of services to install.

Essentials
Management and support for Cloudera's distribution including Hadoop.
Services: HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, and Hue

Data Engineering
Process, develop, and serve predictive models.
Services: HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Spark

Data Warehouse
The modern data warehouse for today, tomorrow, and beyond.
Services: HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Impala

Operational Database
Real-time insights for modern data-driven business.
Services: HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and HBase

[✓ All Services (Cloudera Enterprise Data Hub)]
Everything you need to become information-driven, with complete use of the platform.
Services: HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, HBase, Impala, Solr, Spark, and Key-Value Store Indexer
Note: Please ensure that you have the appropriate license key for Cloudera Impala, Cloudera Search, HBase, and Spark or contact Cloudera for assistance.

Custom Services
Choose your own services. Services required by chosen services will automatically be included. Flume can be added after your initial cluster has been set up.

This wizard will also install the Cloudera Management Service. These are a set of components that enable monitoring, reporting, events, and alerts; these components require databases to store information, which will be configured on the next page.

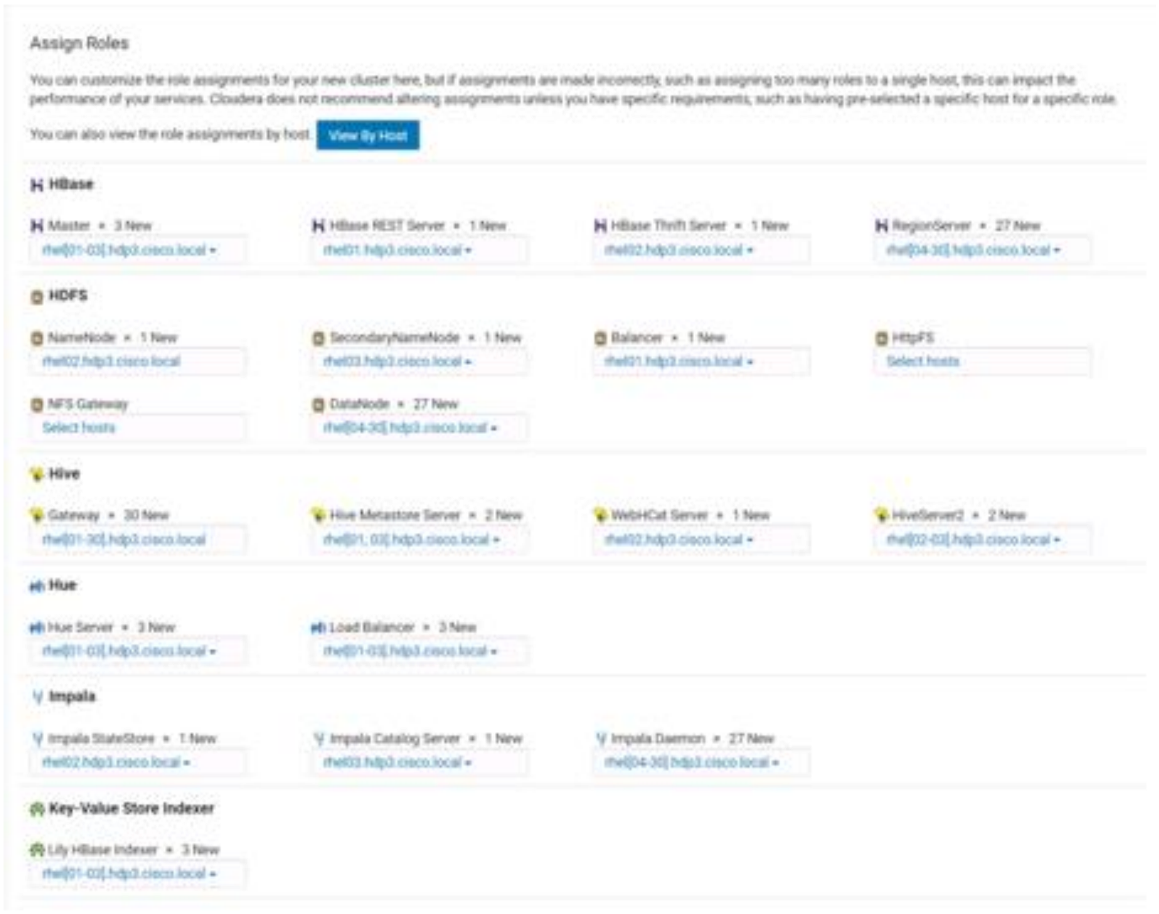
Include Cloudera Navigator

16. This is a critical step in the installation. Inspect and customize the role assignments of all the nodes based on your requirements and click Continue.

17. Reconfigure the service assignment to match Table 9

Table 9 Service/Role Assignment

Service Name	Host
NameNode	Rhel01, rhel02 (HA)
HistoryServer	Rhel01
JournalNodes	Rhel01, rhel02, rhel03
ResourceManager	Rhel02, rhel03 (HA)
Hue Server	Rhel02
HiveMetastore Server	Rhel01
HiveServer2	Rhel02
HBase Master	Rhel02
Oozie Server	Rhel01
ZooKeeper	Rhel01, rhel02, rhel03
DataNode	Rhel04 to rhel30
NodeManager	Rhel04 to rhel30
RegionServer	Rhel04 to rhel30
Sqoop Server	Rhel01
Impala Catalog Server Daemon	Rhel01
Impala State Store	Rhel02
Impala Daemon	Rhel04 to rhel30
Solr Server	Rhel04 (can be installed on all hosts if needed, if there is a search use case)
Spark History Server	Rhel01
Spark Executors	Rhel04 to rhel30



Assign Roles

You can customize the role assignments for your new cluster here, but if assignments are made incorrectly, such as assigning too many roles to a single host, this can impact the performance of your services. Cloudera does not recommend altering assignments unless you have specific requirements, such as having pre-selected a specific host for a specific role.

You can also view the role assignments by host. View By Host

HBase

Master × 3 New
rhel[01-03].hdp3.cisco.local ▾

HBase REST Server × 1 New
rhel01.hdp3.cisco.local ▾

HBase Thrift Server × 1 New
rhel02.hdp3.cisco.local ▾

RegionServer × 27 New
rhel[04-30].hdp3.cisco.local ▾

HDFS

NameNode × 1 New
rhel02.hdp3.cisco.local

SecondaryNameNode × 1 New
rhel03.hdp3.cisco.local ▾

Balancer × 1 New
rhel01.hdp3.cisco.local ▾

HttpFS
Select hosts

NFS Gateway
Select hosts

DataNode × 27 New
rhel[04-30].hdp3.cisco.local ▾

Hive

Gateway × 30 New
rhel[01-30].hdp3.cisco.local

Hive Metastore Server × 2 New
rhel[01, 03].hdp3.cisco.local ▾

WebHCat Server × 1 New
rhel02.hdp3.cisco.local ▾

HiveServer2 × 2 New
rhel[02-03].hdp3.cisco.local ▾

Hue

Hue Server × 3 New
rhel[01-03].hdp3.cisco.local ▾

Load Balancer × 3 New
rhel[01-03].hdp3.cisco.local ▾

Impala

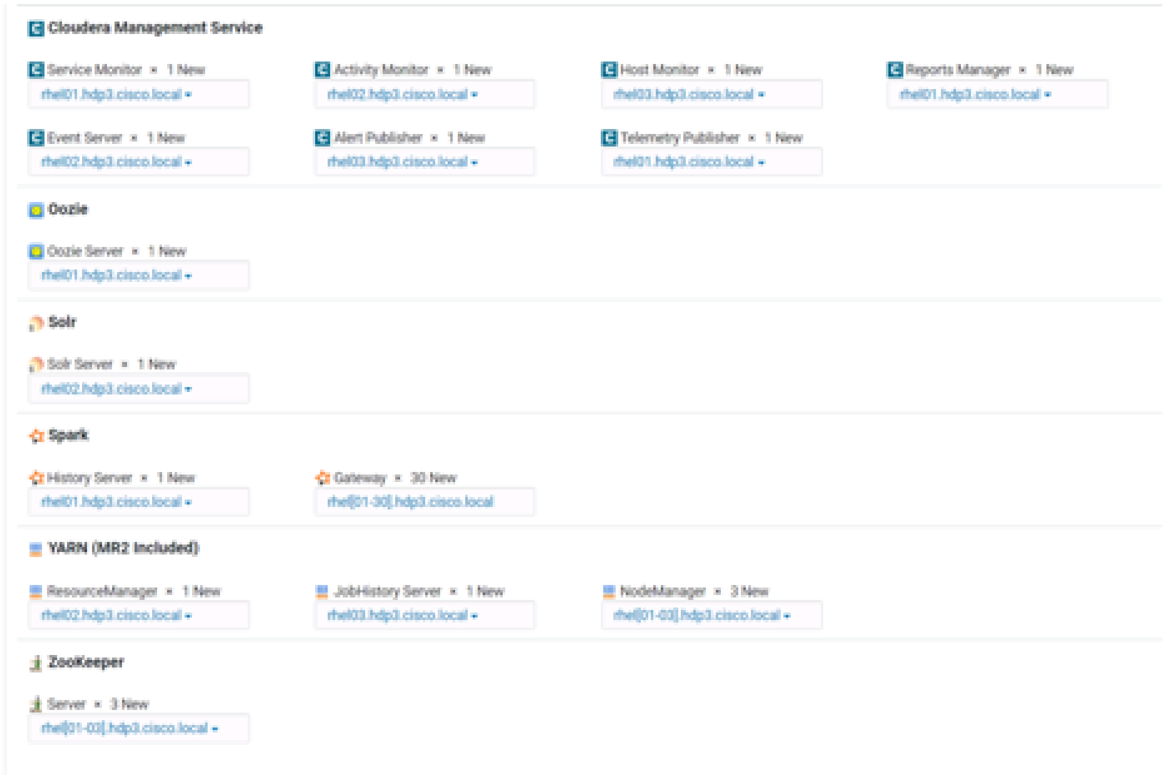
Impala StateStore × 1 New
rhel02.hdp3.cisco.local ▾

Impala Catalog Server × 1 New
rhel03.hdp3.cisco.local ▾

Impala Daemon × 27 New
rhel[04-30].hdp3.cisco.local ▾

Key-Value Store Indexer

Lily HBase Indexer × 3 New
rhel[01-03].hdp3.cisco.local ▾



Cloudera Management Service

Service Monitor × 1 New
rhel01.hdp3.cisco.local ▾

Activity Monitor × 1 New
rhel02.hdp3.cisco.local ▾

Host Monitor × 1 New
rhel03.hdp3.cisco.local ▾

Reports Manager × 1 New
rhel01.hdp3.cisco.local ▾

Event Server × 1 New
rhel02.hdp3.cisco.local ▾

Alert Publisher × 1 New
rhel03.hdp3.cisco.local ▾

Telemetry Publisher × 1 New
rhel01.hdp3.cisco.local ▾

Oozie

Oozie Server × 1 New
rhel01.hdp3.cisco.local ▾

Solr

Solr Server × 1 New
rhel02.hdp3.cisco.local ▾

Spark

History Server × 1 New
rhel01.hdp3.cisco.local ▾

Gateway × 30 New
rhel[01-30].hdp3.cisco.local

YARN (MR2 Included)

ResourceManager × 1 New
rhel02.hdp3.cisco.local ▾

JobHistory Server × 1 New
rhel03.hdp3.cisco.local ▾

NodeManager × 3 New
rhel[01-03].hdp3.cisco.local ▾

ZooKeeper

Server × 3 New
rhel[01-03].hdp3.cisco.local ▾

Set Up the Database

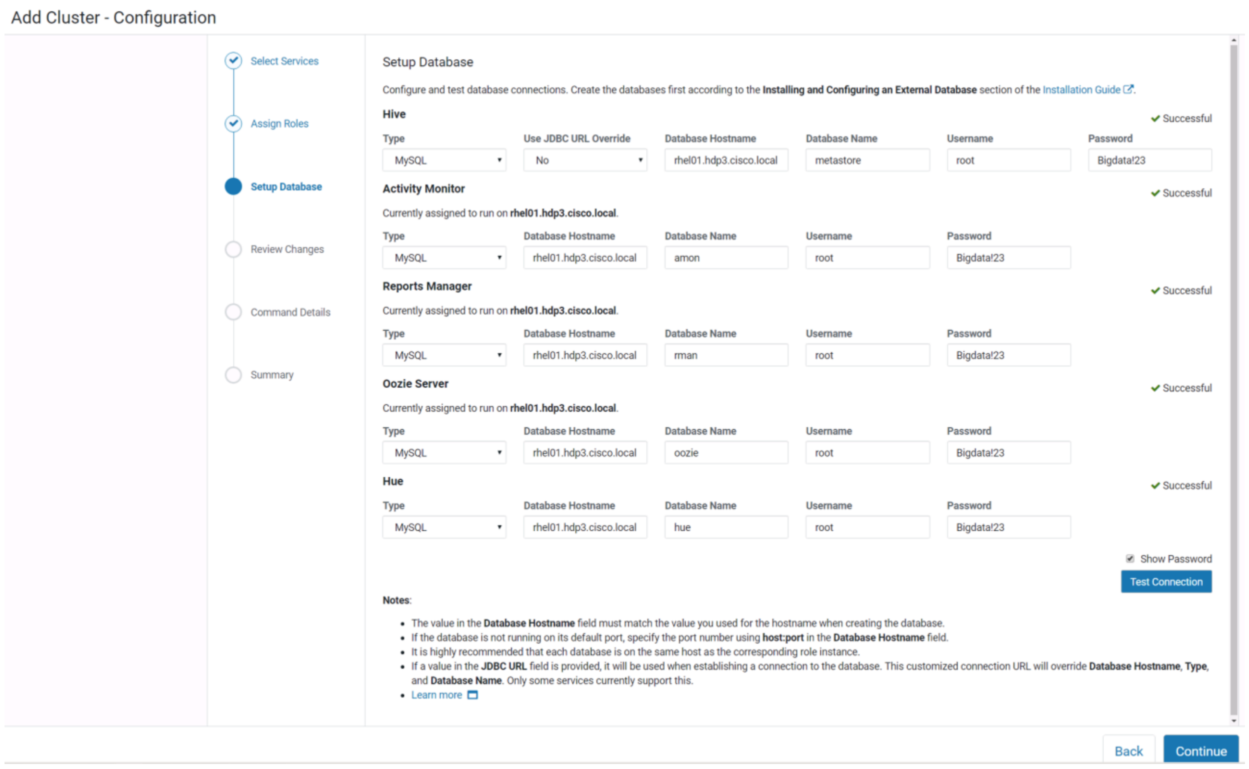
The role assignment recommendation above is for clusters of up to 64 servers. For clusters larger than 64 nodes, use the HA recommendation defined in Table 9

To set up the database, follow these steps:

1. In the Database Host Name sections use port 3306 for TCP/IP because connection to the remote server always uses TCP/IP.

2. Enter the Database Name, username and password that were used during the database creation stage earlier in this document.

3. Click Test Connection to verify the connection and click Continue.


Add Cluster - Configuration

Setup Database

Configure and test database connections. Create the databases first according to the **Installing and Configuring an External Database** section of the Installation Guide.

Hive ✓ Successful

Type	Use JDBC URL Override	Database Hostname	Database Name	Username	Password
MySQL	No	rhel01.hdp3.cisco.local	metastore	root	Bigdata!23

Activity Monitor ✓ Successful

Currently assigned to run on **rhel01.hdp3.cisco.local**.

Type	Database Hostname	Database Name	Username	Password
MySQL	rhel01.hdp3.cisco.local	amon	root	Bigdata!23

Reports Manager ✓ Successful

Currently assigned to run on **rhel01.hdp3.cisco.local**.

Type	Database Hostname	Database Name	Username	Password
MySQL	rhel01.hdp3.cisco.local	rman	root	Bigdata!23

Oozie Server ✓ Successful

Currently assigned to run on **rhel01.hdp3.cisco.local**.

Type	Database Hostname	Database Name	Username	Password
MySQL	rhel01.hdp3.cisco.local	oozie	root	Bigdata!23

Hue ✓ Successful

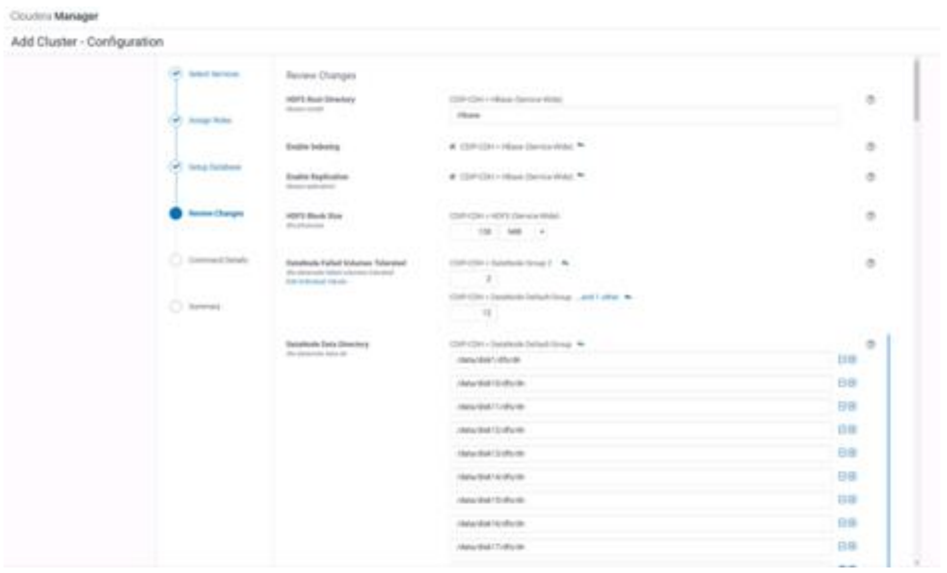
Type	Database Hostname	Database Name	Username	Password
MySQL	rhel01.hdp3.cisco.local	hue	root	Bigdata!23

✓ Show Password

Notes:

- The value in the **Database Hostname** field must match the value you used for the hostname when creating the database.
- If the database is not running on its default port, specify the port number using **host:port** in the **Database Hostname** field.
- It is highly recommended that each database is on the same host as the corresponding role instance.
- If a value in the **JDBC URL** field is provided, it will be used when establishing a connection to the database. This customized connection URL will override **Database Hostname**, **Type**, and **Database Name**. Only some services currently support this.
- Learn more

4. Review and customize the configuration changes based on your requirements.

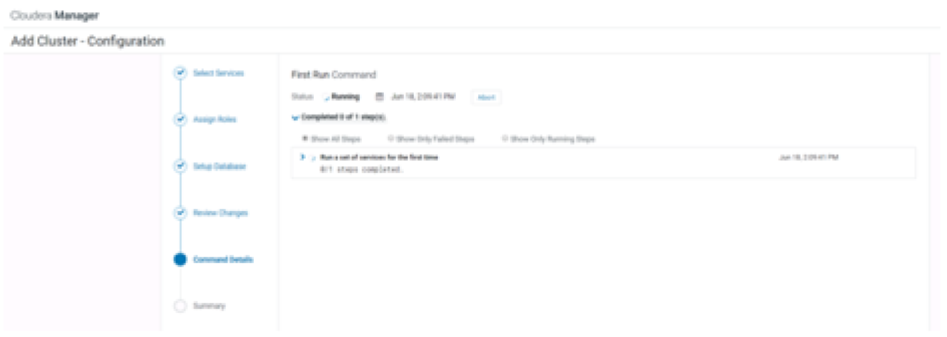


Cloudera Manager

Add Cluster - Configuration

Review Changes

5. Click Continue to start running the cluster services.



Cloudera Manager

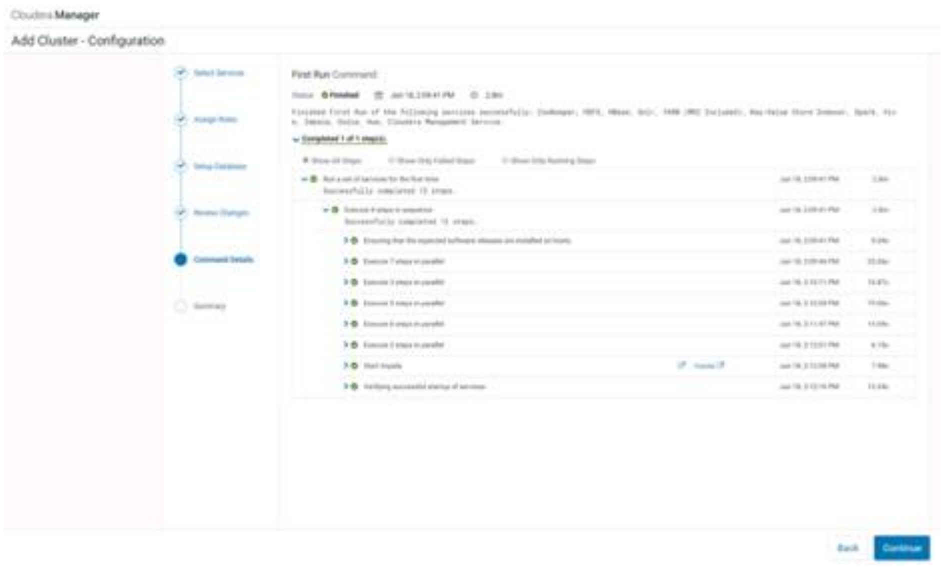
Add Cluster - Configuration

First Run Command

Status Running Jun 18, 2:09:41 PM

Completed 0 of 1 step(s).

Run a set of services for the first time Jun 18, 2:09:41 PM



Cloudera Manager

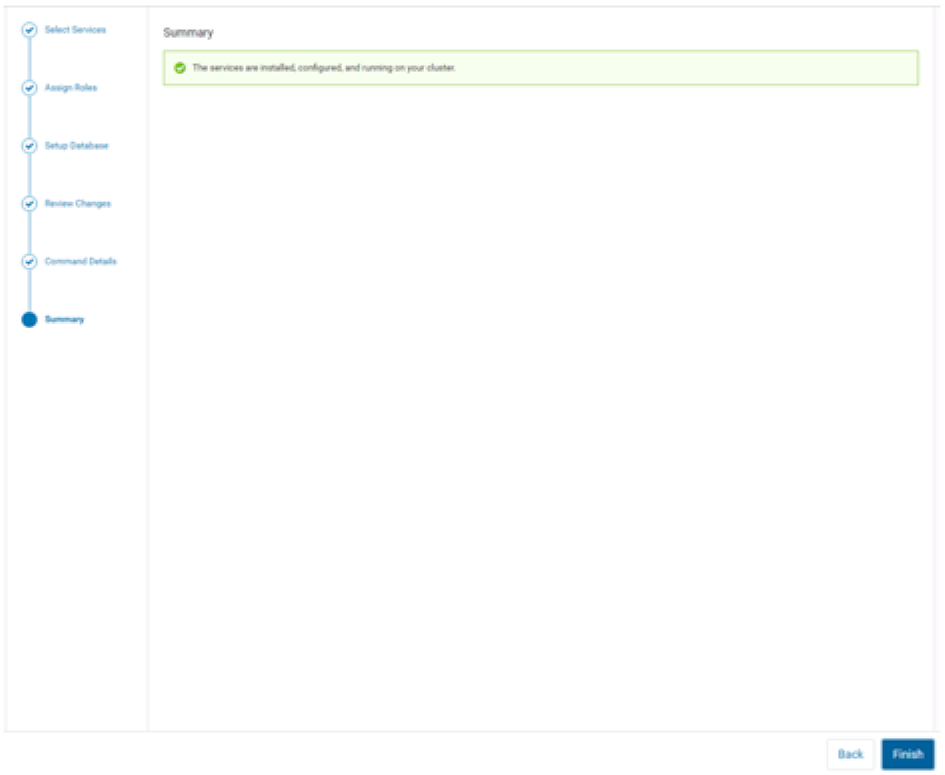
Add Cluster - Configuration

First Run Command

Status Finished

Completed 1 of 1 step(s).

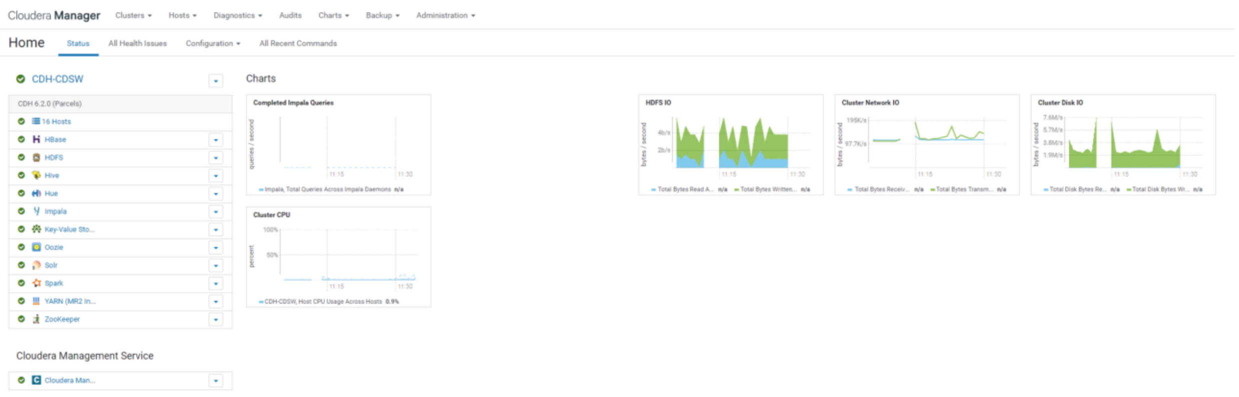
6. Hadoop services are installed, configured, and now running on all the nodes of the cluster. Click Finish to complete the installation.



Summary

The services are installed, configured, and running on your cluster.

7. Cloudera Manager now displays the status of all Hadoop services running on the cluster.



Scale the Cluster

The role assignment recommendation above is for cluster with at least 64 servers and in High Availability. For smaller cluster running without High Availability the recommendation is to dedicate one server for NameNode and a second server for secondary name node and YARN Resource Manager. For larger clusters larger than 28 nodes the recommendation is to dedicate one server each for name node, YARN Resource Manager and one more for running both NameNode (High Availability) and Resource Manager (High Availability) as in the table (no Secondary NameNode when in High Availability).

For production clusters, it is recommended to set up NameNode and Resource manager in High Availability mode.

This implies that there will be at least 3 master nodes, running the NameNode, YARN Resource manager, the failover counter-part being designated to run on another node and a third node that would have similar capacity as the other two nodes.

All the three nodes will also need to run zookeeper and quorum journal node services. It is also recommended to have a minimum of 7 DataNodes in a cluster. Please refer to the next section for details on how to enable HA.

Enable High Availability

Setting up High Availability is done after the Cloudera Installation is completed.

HDFS High Availability

The HDFS High Availability feature provides the option of running two NameNodes in the same cluster, in an Active/Passive configuration. These are referred to as the Active NameNode and the Standby NameNode. Unlike the Secondary NameNode, the Standby NameNode is a hot standby, allowing a fast failover to a new NameNode in the case that a machine crashes, or a graceful administrator-initiated failover for the purpose of planned maintenance. There cannot be more than two NameNodes.

For more information go to:
https://www.cloudera.com/documentation/enterprise/latest/topics/cdh_hag_hdfs_ha_enabling.html

Set Up HDFS High Availability

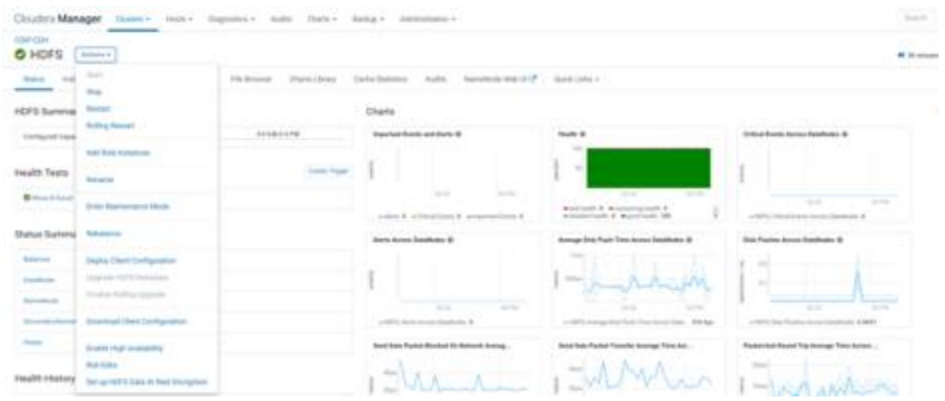
The Enable High Availability workflow leads through adding a second (standby) NameNode and configuring JournalNodes. During the workflow, Cloudera Manager creates a federated namespace. To set up HDFS High Availability, follow these steps:

1. Log into the admin node (rhel01) and create the Edit directory for the JournalNode:

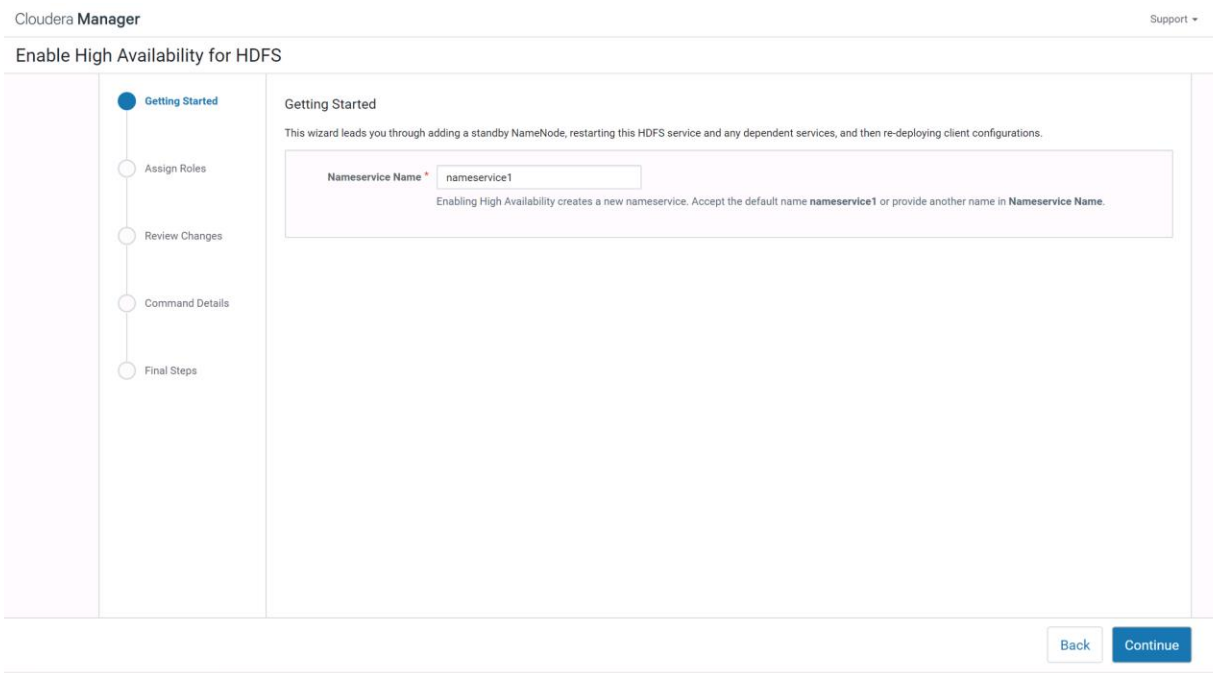
```
# ansible namenodes -m shell -a "mkdir -p /data/disk1/namenode-edits"
# ansible namenodes -m shell -a "chmod 77 /data/disk1/namenode-edits"
```

2. Log into the Cloudera manager and go to the HDFS service.

3. Select Actions> Enable High Availability. A screen showing the hosts that are eligible to run a standby NameNode and the JournalNodes displays.



4. Specify a name for the nameservice or accept the default name nameservice1 and click Continue.



Getting Started

This wizard leads you through adding a standby NameNode, restarting this HDFS service and any dependent services, and then re-deploying client configurations.

Nameservice Name * nameservice1

Enabling High Availability creates a new nameservice. Accept the default name **nameservice1** or provide another name in **Nameservice Name**.

5. In the NameNode Hosts field, click Select a host. The host selection dialog displays.

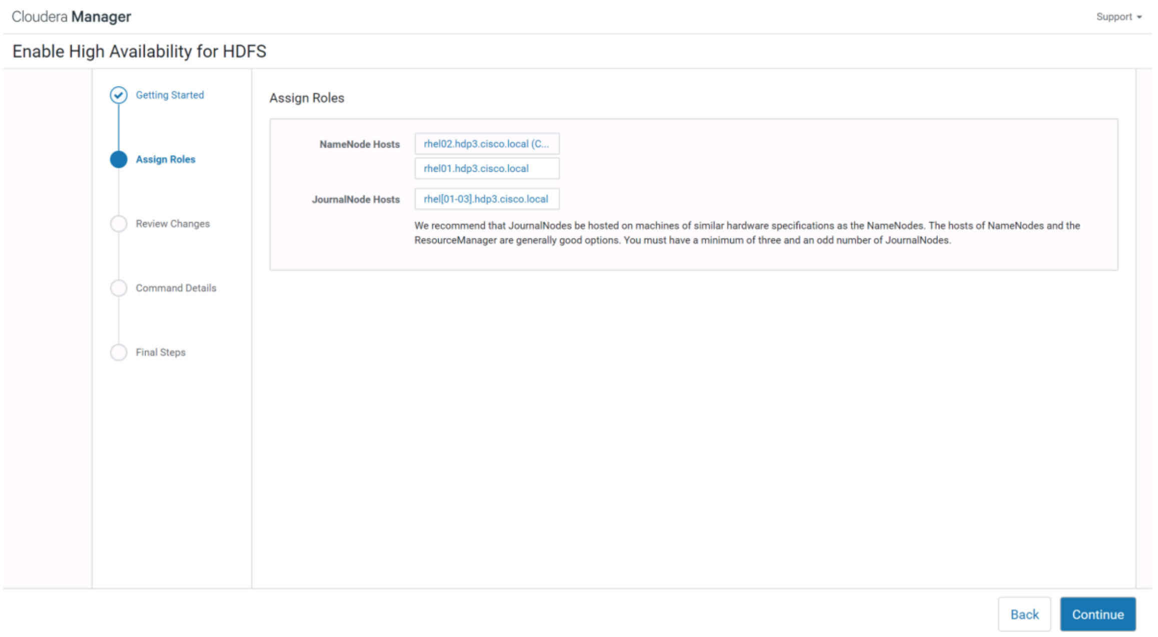
6. Check the checkbox next to the hosts (rhel2) where the standby NameNode is to be set up and click OK.

7. In the JournalNode Hosts field, click Select hosts. The host selection dialog displays.

8. Check the checkboxes next to an odd number of hosts (a minimum of three) to act as JournalNodes and click OK. We used the same nodes for the Zookeeper nodes.

9. Click Continue.

The standby NameNode cannot be on the same host as the active NameNode, and the host that is chosen should have the same hardware configuration (RAM, disk space, number of cores, and so on) as the active NameNode.



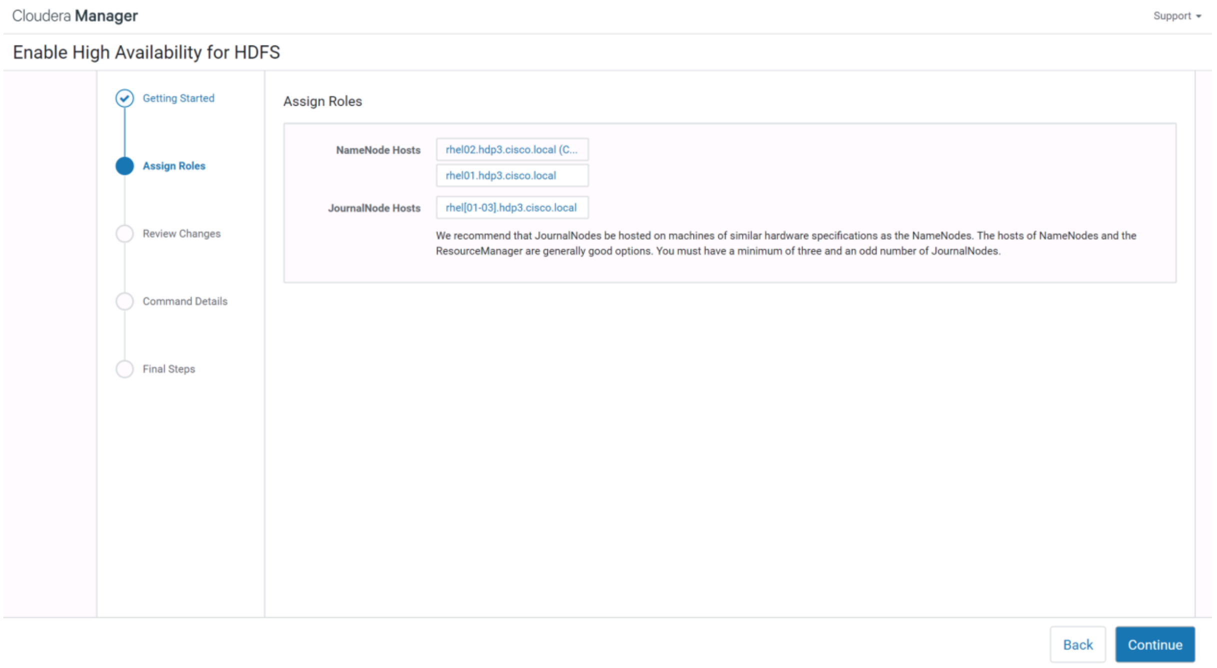
Enable High Availability for HDFS

Assign Roles

NameNode Hosts: rhel02.hdp3.cisco.local (C...; rhel01.hdp3.cisco.local

JournalNode Hosts: rhel[01-03].hdp3.cisco.local

We recommend that JournalNodes be hosted on machines of similar hardware specifications as the NameNodes. The hosts of NameNodes and the ResourceManager are generally good options. You must have a minimum of three and an odd number of JournalNodes.



Enable High Availability for HDFS

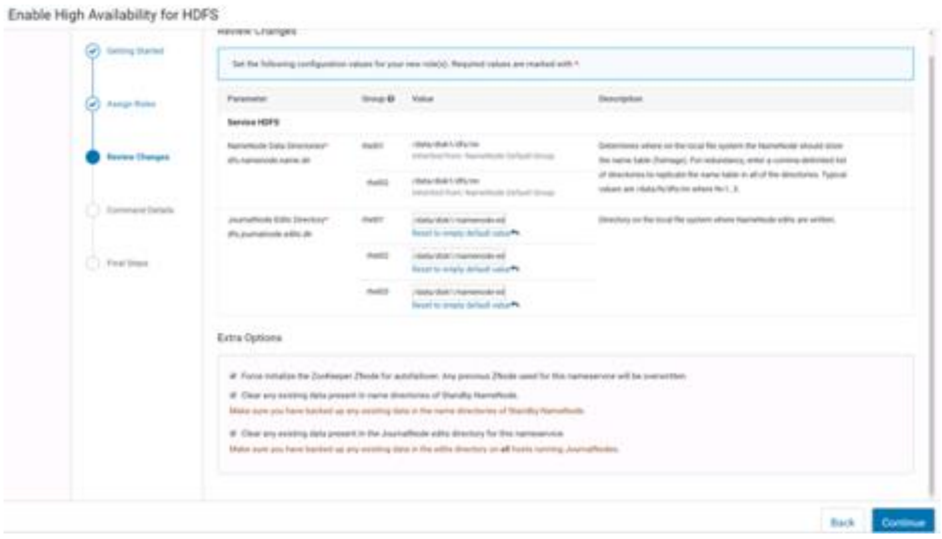
Assign Roles

NameNode Hosts: rhel02.hdp3.cisco.local (C...; rhel01.hdp3.cisco.local

JournalNode Hosts: rhel[01-03].hdp3.cisco.local

We recommend that JournalNodes be hosted on machines of similar hardware specifications as the NameNodes. The hosts of NameNodes and the ResourceManager are generally good options. You must have a minimum of three and an odd number of JournalNodes.

10. In the JournalNode Edits Directory property, enter a directory location created earlier in step 1 for the Journal-Node edits directory into the fields for each JournalNode host.



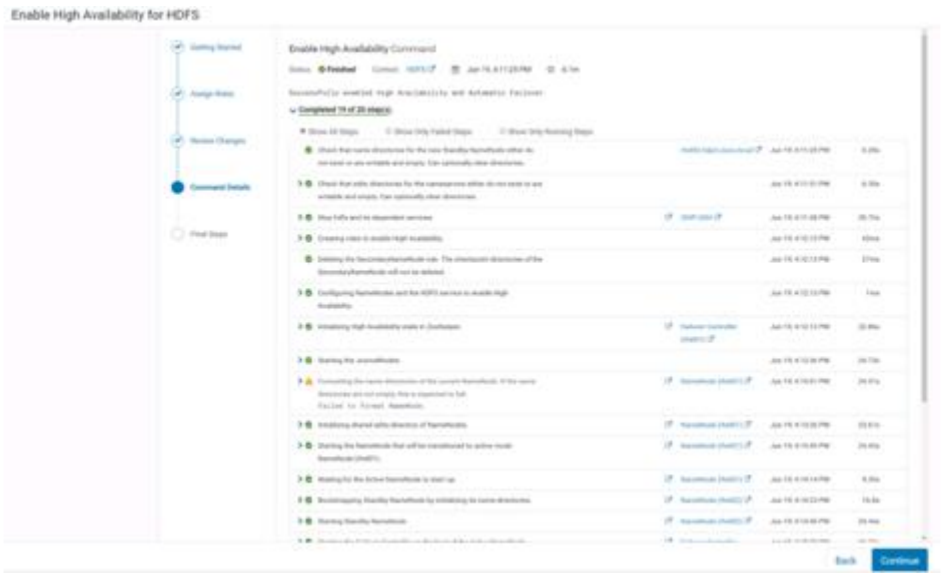
> **The directories specified should be empty and must have the appropriate permissions.**

11. Extra Options: Decide whether Cloudera Manager should clear existing data in ZooKeeper, Standby NameNode, and JournalNodes. If the directories are not empty (for example, re-enabling a previous HA configuration), Cloudera Manager will not automatically delete the contents—select to delete the contents by keeping the default checkbox selection. The recommended default is to clear the directories.

> **If you choose to not configure any of the extra options, the data should be in sync across the edits directories of the JournalNodes and should have the same version data as the NameNodes.**

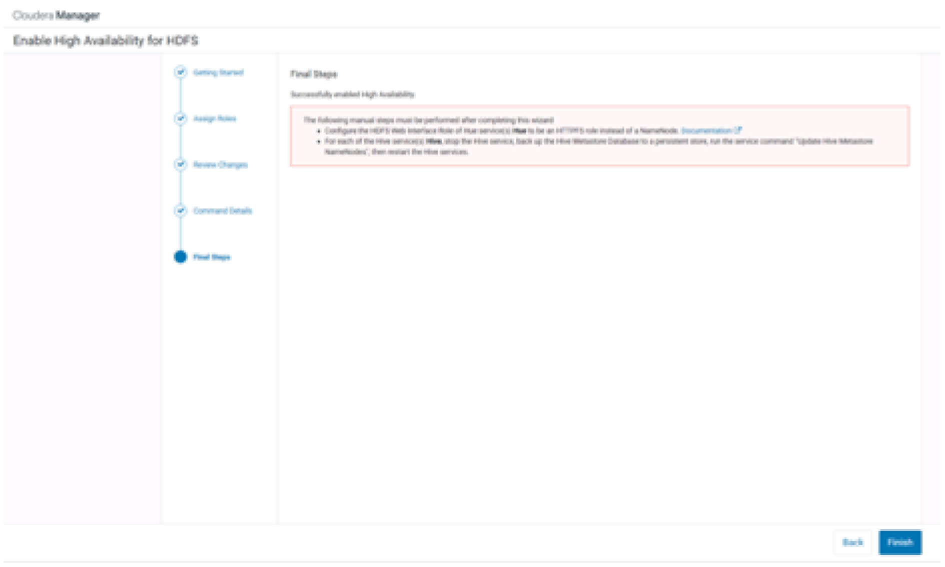
12. Click Continue.

13. Cloudera Manager executes a set of commands that will stop the dependent services, delete, create, and configure roles and directories as appropriate, create a nameservice and failover controller, and restart the dependent services and deploy the new client configuration.



Formatting the name directory is expected to fail, if the directories are not empty.

14. In the next screen additional steps are suggested by the Cloudera Manager to update the Hue and Hive metastore. Click Finish.



Final Steps

Successfully enabled High Availability.

The following manual steps must be performed after completing this wizard:

- Configure the HDFS Web Interface Role of Hue service(s) **Hue** to be an HTTPFS role instead of a NameNode. Documentation
- For each of the Hive service(s) **Hive**, stop the Hive service, back up the Hive Metastore Database to a persistent store, run the service command "Update Hive Metastore NameNodes", then restart the Hive services.

The following subsections explain configuring Hue and Hive for High Availability as needed.

Configure Hive Metastore to Use HDFS High Availability

To configure the Hive Megastore to use HDFS High Availability, follow these steps:

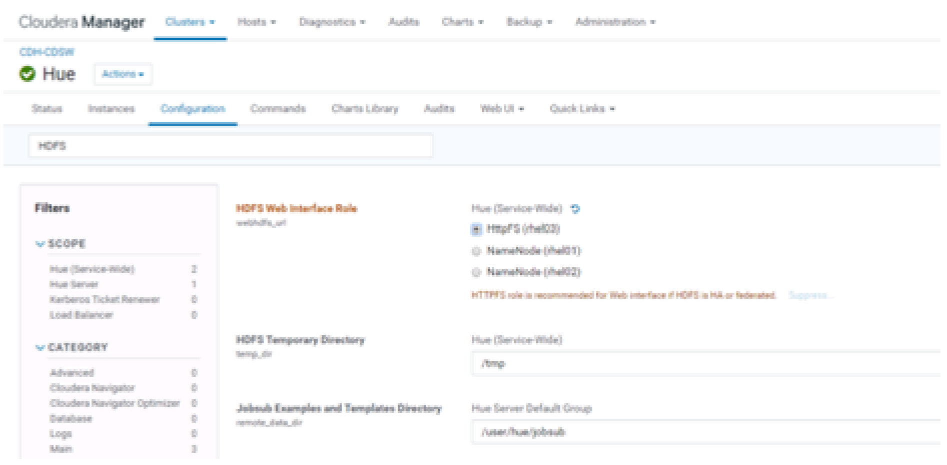
1. Go the Hive service.
2. Select Actions > Stop.
3. Click Stop to confirm the command.
4. Back up the Hive metastore database (if any existing data is present).
5. Select Actions> Update Hive Metastore NameNodes and confirm the command.
6. Select Actions> Start.
7. Restart the Hue and Impala services if stopped prior to updating the Metastore.

Configure Hue to Work with HDFS High Availability

To configure Hue to work with HDFS High Availability, follow these steps:

1. Go to the HDFS service.
2. Click the Instances tab.
3. Click Add Role Instances.

4. Select the text box below the HttpFS field. The Select Hosts dialog displays.
5. Select the host on which to run the role and click OK.
6. Click Continue.
7. Check the checkbox next to the HttpFS role and select Actions for Selected> Start.
8. After the command has completed, go to the Hue service.
9. Click the Configuration tab.
10. Locate the HDFS Web Interface Role property or search for it by typing its name in the Search box.
11. Select the HttpFS role that was just created instead of the NameNode role and save your changes.
12. Restart the Hue service.



HDFS

HDFS Web Interface Role
webhdfs_url
Hue (Service-Wide)
(●) HttpFS (rhel03)
( ) NameNode (rhel01)
( ) NameNode (rhel02)
HTTPFS role is recommended for Web interface if HDFS is HA or federated.

HDFS Temporary Directory
temp_dir
Hue (Service-Wide)
/tmp

Jobsub Examples and Templates Directory
remote_data_dir
Hue Server Default Group
/user/hue/jobsub

Please refer to the High Availability section in the Cloudera Management document: https://www.cloudera.com/documentation/enterprise/6/6.2/topics/admin_ha.html for more details on setting up High Availability for other components like Impala, Oozie, and so on.

YARN High Availability

The YARN Resource Manager (RM) is responsible for tracking the resources in a cluster and scheduling applications (for example, MapReduce jobs). Before CDH 5, the RM was a single point of failure in a YARN cluster. The RM high availability (HA) feature adds redundancy in the form of an Active/Standby RM pair to remove this single point of failure. Furthermore, upon failover from the Standby RM to the Active, the applications can resume from their last check-pointed state; for example, completed map tasks in a MapReduce job are not re-run on a subsequent attempt. This allows events such the following to be handled without any significant performance effect on running applications.

- Unplanned events such as machine crashes.
- Planned maintenance events such as software or hardware upgrades on the machine running the ResourceManager.

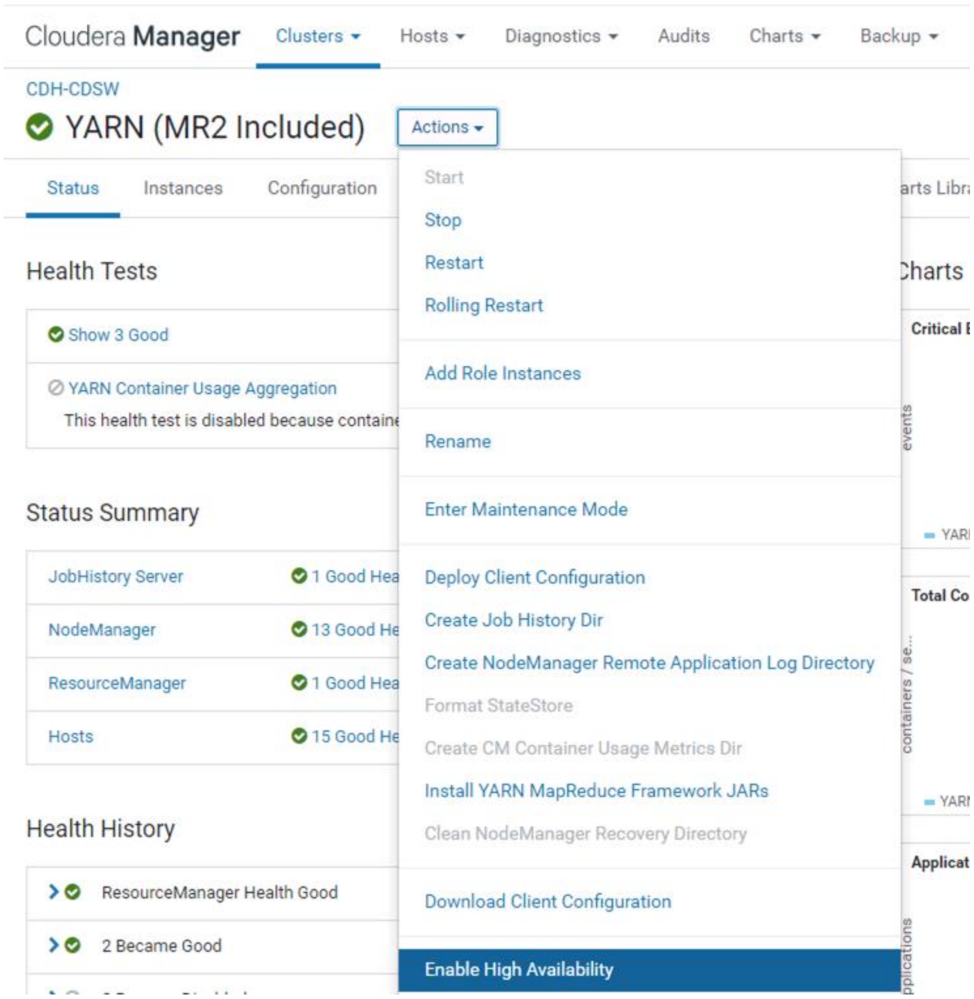
For more information, go to:
https://www.cloudera.com/documentation/enterprise/latest/topics/cdh_hag_rm_ha_config.html#xd_583c10bfdbd326ba--43d5fd93-1410993f8c2--7f77

Set Up YARN High Availability

To set up YARN high availability, follow these steps:

1. Log into the Cloudera manager and go to the YARN service.

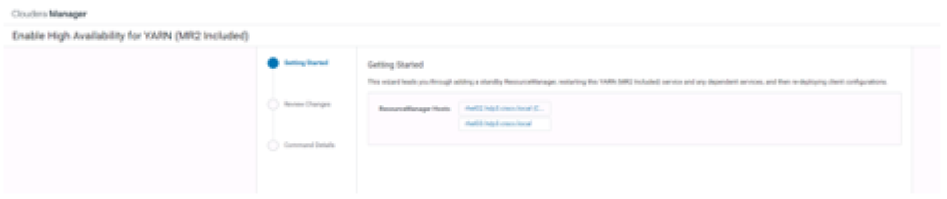
2. Select Actions> Enable High Availability.



3. A screen showing the hosts that are eligible to run a standby ResourceManager displays.

4. The host where the current ResourceManager is running is not available as a choice.

5. Select the host (rhel3) where the standby ResourceManager is to be installed and click Continue.



Getting Started

This wizard leads you through adding a standby ResourceManager, restarting the YARN (MR2 Included) service and any dependent services, and then re-deploying client configurations.

ResourceManager Hosts

6. Cloudera Manager proceeds to execute a set of commands that stop the YARN service, add a standby ResourceManager, initialize the ResourceManager high availability state in ZooKeeper, restart YARN, and redeploy the relevant client configurations.

7. Click Finish once the installation is completed successfully.

Configure Yarn (MR2 Included) and HDFS Services

The parameters in Table 10 and Table 11 are used for Cisco UCS Integrated Infrastructure for Big Data and Analytics Performance Optimized cluster configuration described in this document. These parameters are to be changed based on the cluster configuration, number of nodes and specific workload.

Table 10 YARN

Service	Value
mapreduce.map.memory.mb	3GiB
mapreduce.reduce.memory.mb	3GiB
mapreduce.map.java.opts.max.heap	2560 MiB
yarn.nodemanager.resource.memorymb	180 GiB
yarn.nodemanager.resource.cpu-vcores	32
yarn.scheduler.minimum-allocation-mb	4 GiB
yarn.scheduler.maximum-allocation-mb	180 GiB
yarn.scheduler.maximum-allocation-vcores	48
mapreduce.task.io.sort.mb	256 MiB

Table 11 HDFS

Service	Value
dfs.datanode.failed.volumes.tolerated	6
dfs.datanode.du.reserved	50 GiB
dfs.datanode.data.dir.perm	755
Java Heap Size of Namenode in Bytes	2628 MiB
dfs.namenode.handler.count	54
dfs.namenode.service.handler.count	54
Java Heap Size of Secondary namenode in Bytes	2628 MiB

Configure Spark

The two main resources that Spark (and YARN) are dependent on are CPU and memory. Disk and network I/O, play a part in Spark performance as well, but neither Spark nor YARN currently can actively manage them. Every Spark executor in any application has the same fixed number of cores and same fixed heap size. The number of cores can be specified with the **executor-cores** flag when invoking spark-submit, spark-shell, and pyspark from the command line, or by setting the **spark.executor.cores** property in the **spark-defaults.conf** file or in the **SparkConf** object.

And the heap size can be controlled with the executor-memory flag or the spark.executor.memory property. The cores property controls the number of concurrent tasks an executor can run, executor-cores = 5 mean that each executor can run a maximum of five tasks at the same time. The memory property impacts the amount of data Spark can cache, as well as the maximum sizes of the shuffle data structures used for grouping, aggregations, and joins.

The num-executors command-line flag or spark.executor.instances configuration property control the number of executors requested. Dynamic Allocation can be enabled from CDH5.4 instead setting the spark.dynamicAllocation.enabled to true. Dynamic allocation enables a Spark application to request executors when there is a backlog of pending tasks and free up executors when idle.

Asking for five executor cores will result in a request to YARN for five virtual cores. The memory requested from YARN is a little more complex for a couple reasons:

- `executor-memory/spark.executor.memory` controls the executor heap size, but JVMs can also use some memory off heap, for example for VM overhead, interned Strings and direct byte buffers. The value of the `spark.yarn.executor.memoryOverhead` property is added to the executor memory to determine the full memory request to YARN for each executor. It defaults to max `(384, 0.10 * spark.executor.memory)`.
- YARN may round the requested memory up a little. YARN's `yarn.scheduler.minimum-allocation-mb and yarn.scheduler.increment-allocation-mb` properties control the minimum and increment request values respectively.
- The application master is a non-executor container with the special capability of requesting containers from YARN, takes up resources of its own that must be budgeted in. In *yarn-client* mode, it defaults to a 1024MB and one vcore. In *yarn-cluster* mode, the application master runs the driver, so it's often useful to add its resources with the –driver-memory and –driver-cores properties.
- Running executors with too much memory often results in excessive garbage collection delays. 64GB is a rough guess at a good upper limit for a single executor.
- A good estimate is that at most five tasks per executor can achieve full write throughput, so it's good to keep the number of cores per executor around that number.
- Running tiny executors (with a single core and just enough memory needed to run a single task, for example) throws away the benefits that come from running multiple tasks in a single JVM. For example, broadcast variables need to be replicated once on each executor, so many small executors will result in many more copies of the data.

Tune Resource Allocation for Spark

Below is an example of configuring a Spark application to use as much of the cluster as possible, we are using an example cluster with 16 nodes running NodeManagers, each equipped with 56 cores and 256GB of memory.

yarn.nodemanager.resource.memory-mb and yarn.nodemanager. resource.cpu-vcores should be set to 180 * 1024 = 184320 (megabytes) and 48 respectively.

```
spark.executor.instances/num-executors = 63
spark.executor.cores/--executor-cores = 5
spark.executor.memory/--executor-memory = 41G
```

This configuration results in four executors on all nodes except for the one with the AM, which will have three executors.

```
executor-memory is derived as (180/4 executors per node) = 45; 45 * 0.10 = 4.5 45 –
4.5 ~ 40.
For taking care of long running processes use 2G for the spark driver
spark.driver.memory = 2G
```

Submit a Job

```
--driver -memory 2G –executor -memory 40G --num-executors 63 --executor-cores 5 --
properties-file /opt/cloudera/parcels/CDH/etc/spark/conf.dist/spark-defaults.conf
```

In yarn-cluster mode, the local directories used by the Spark executors and the Spark driver will be the local directories configured for YARN (Hadoop YARN config yarn.nodemanager.local-dirs). If the user specifies spark.local.dir, it will be ignored.

In yarn-client mode, the Spark executors will use the local directories configured for YARN while the Spark driver will use those defined in spark.local.dir. The Spark driver does not run on the YARN cluster in yarn-client mode, only the Spark executors do.

```
spark.local.dir /tmp (Directory to use for "scratch" space in Spark, including map
output files and RDDs that get stored on disk. This should be on a fast, local disk
in your system).
```

Every Spark stage has several tasks, each of which processes data sequentially. In tuning Spark jobs, this parallelism number is the most important parameter in determining performance. The number of tasks in a stage is the same as the number of partitions in the last RDD in the stage. The number of partitions in an RDD is the same as the number of partitions in the RDD on which it depends, with a couple exceptions: the coalesce transformation allows creating an RDD with fewer partitions than its parent RDD, the union transformation creates an RDD with the sum of its parents' number of partitions, and Cartesian creates an RDD with their product.

```
RDDs produced by a file have their partitions determined by the underlying MapReduce
InputFormat that's used. Typically there will be a partition for each HDFS block
being read. Partitions for RDDs produced by parallelize come from the parameter
given by the user, or spark.default.parallelism if none is given.
```

The primary concern is that the number of tasks will be too small. If there are fewer tasks than slots available to run them in, the stage won't be taking advantage of all the CPU available.

If the stage in question is reading from Hadoop, your options are:

- Use the repartition transformation, which will trigger a shuffle.
- Configure your InputFormat to create more splits.
- Write the input data out to HDFS with a smaller block size.

If the stage is getting its input from another stage, the transformation that triggered the stage boundary will accept a numPartitions argument.

The most straightforward way to tune the number of partitions is experimentation: Look at the number of partitions in the parent RDD and then keep multiplying that by 1.5 until performance stops improving.

In contrast with MapReduce for Spark when in doubt, it is almost always better to be on the side of a larger number of tasks (and thus partitions).

Shuffle Performance Improvement

`spark.shuffle.compress true` (compress map output files)

spark.broadcast.compress true (compress broadcast variables before sending them)

spark.io.compression.codec org.apache.spark.io.SnappyCompressionCodec (codec used to compress internal data such as RDD partitions, broadcast variables and shuffle outputs)

spark.shuffle.spill.compress true (Whether to compress data spilled during shuffles.)

spark.shuffle.io.numConnectionsPerPeer 4 (Connections between hosts are reused in order to reduce connection buildup for large clusters. For clusters with many hard disks and few hosts, this may result in insufficient concurrency to saturate all disks, and so users may consider increasing this value.)

spark.shuffle.file.buffer 64K (Size of the in-memory buffer for each shuffle file output stream. These buffers reduce the number of disk seeks and system calls made in creating intermediate shuffle file)

Improve Serialization Performance

Serialization plays an important role in the performance of any distributed application. Often, this will be the first thing that should be tuned to optimize a Spark application.

`spark.serializer org.apache.spark.serializer.KryoSerializer` (when speed is necessary)

`spark.kryo.referenceTracking false`

`spark.kryoserializer.buffer 2000` (If the objects are large, may need to increase the size further to fit the size of the object being deserialized).

SparkSQL is ideally suited for mixed procedure jobs where SQL code is combined with Scala, Java, or Python programs. In general, the SparkSQL command line interface is used for single user operations and ad hoc queries.

For multi-user SparkSQL environments, it is recommended to use a Thrift server connected via JDBC.

Spark SQL Tuning

Below are some guidelines for Spark SQL tuning:

To compile each query to Java bytecode on the fly, turn on sql.codegen. This can improve performance for large queries but can slow down very short queries.

`spark.sql.codegen true`

`spark.sql.unsafe.enabled true`

Configuration of in-memory caching can be done using the `setConf` method on `SQLContext` or by running `SET key=value` commands using SQL.

`spark.sql.inMemoryColumnarStorage.compressed true` (will automatically select a compression codec for each column based on statistics of the data)

`spark.sql.inMemoryColumnarStorage.batchSize 5000` (Controls the size of batches for columnar caching. Larger batch sizes can improve memory utilization and compression, but risk OOMs when caching data)

The columnar nature of the ORC format helps avoid reading unnecessary columns, but it is still possible to read unnecessary rows. ORC avoids this type of overhead by using predicate push-down with three levels of built-in indexes within each file: file level, stripe level, and row level. This combination of indexed data and columnar storage reduces disk I/O significantly, especially for larger datasets where I/O bandwidth becomes the main bottleneck for performance.

By default, ORC predicate push-down is disabled in Spark SQL. To obtain performance benefits from predicate push-down, enable it explicitly, as follows:

`spark.sql.orc.filterPushdown=true`

In SparkSQL to automatically determine the number of reducers for joins and groupbys, use the parameter,

`spark.sql.shuffle.partitions 200, (default value is 200)`

This property can be put into `hive-site.xml` to override the default value.

Set log to WARN in log4j.properties to reduce log level.

Running Thrift server and connecting to spark-sql through beeline is the recommended option for multi-session testing.

Compression for Hive

Set the following Hive parameters to compress the Hive output files using Snappy compression:

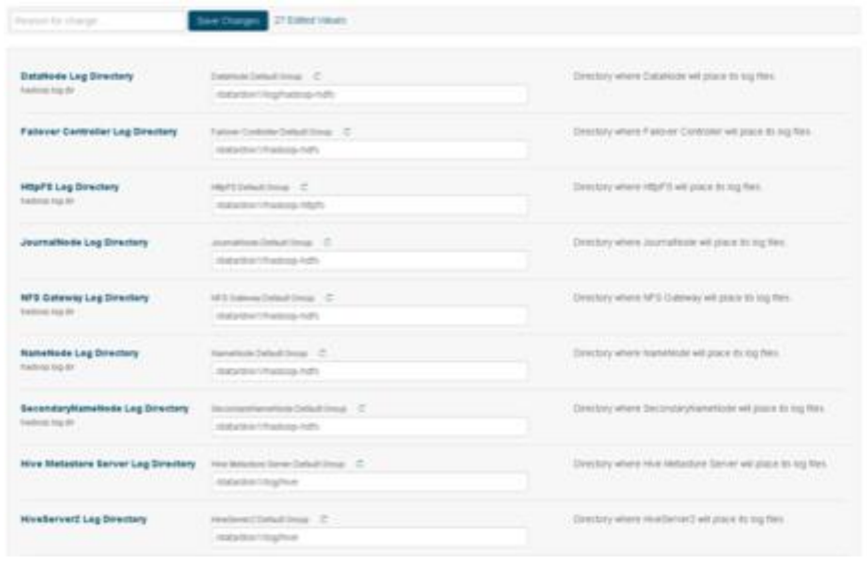
```
hive.exec.compress.output=true
hive.exec.orc.default.compress=SNAPPY
```

Change the Log Directory for All Applications

To change the default log from the /var prefix to /data/disk1, follow these steps:

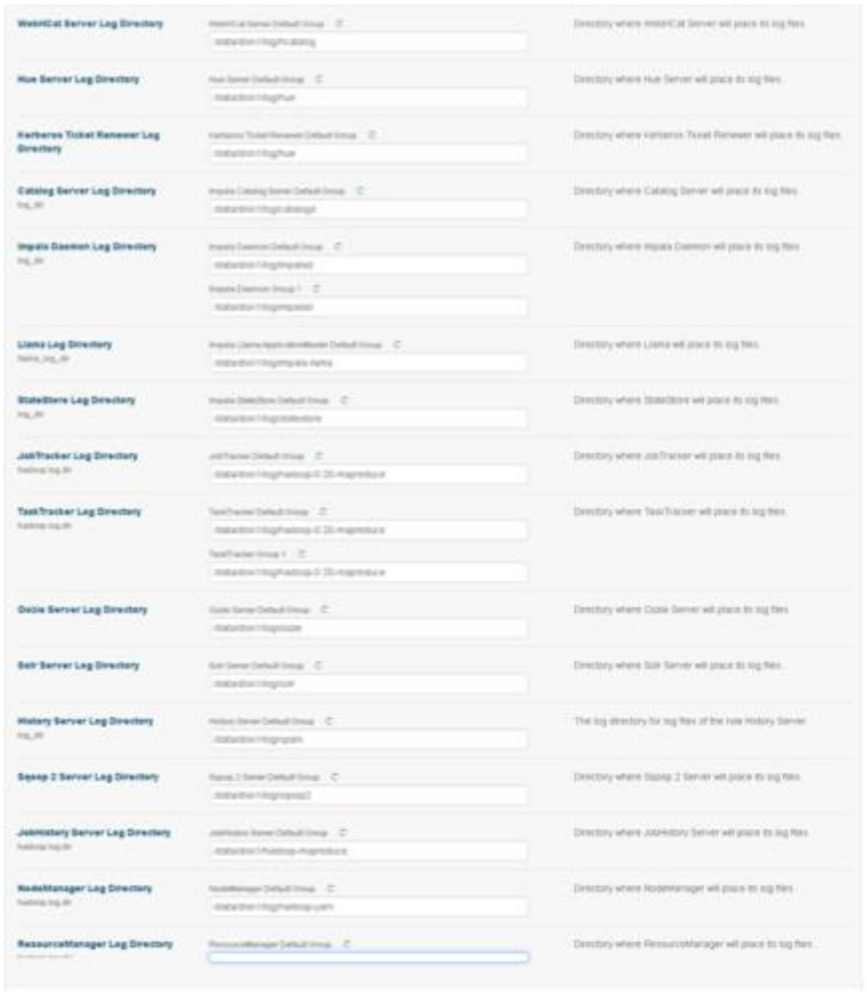
1. Log into the cloudera home page and click My Clusters.

2. From the configuration drop-down list select "All Log Directories."

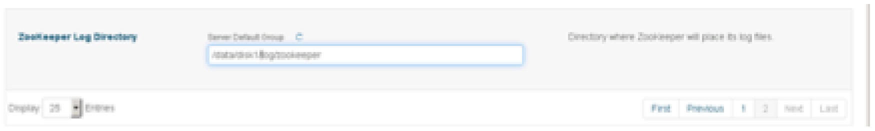


Reason for change | Save Changes | 27 Edited Values

DataNode Log Directory hadoop.log.dir	DataNode Default Group /data/disk1/log/hadoop-hdfs	Directory where DataNode will place its log files.
Failover Controller Log Directory	Failover Controller Default Group /data/disk1/hadoop-hdfs	Directory where Failover Controller will place its log files.
HttpFS Log Directory hadoop.log.dir	HttpFS Default Group /data/disk1/hadoop-httpfs	Directory where HttpFS will place its log files.
JournalNode Log Directory	JournalNode Default Group /data/disk1/hadoop-hdfs	Directory where JournalNode will place its log files.
NFS Gateway Log Directory hadoop.log.dir	NFS Gateway Default Group /data/disk1/hadoop-hdfs	Directory where NFS Gateway will place its log files.
NameNode Log Directory hadoop.log.dir	NameNode Default Group /data/disk1/hadoop-hdfs	Directory where NameNode will place its log files.
SecondaryNameNode Log Directory hadoop.log.dir	SecondaryNameNode Default Group /data/disk1/hadoop-hdfs	Directory where SecondaryNameNode will place its log files.
Hive Metastore Server Log Directory	Hive Metastore Server Default Group /data/disk1/log/hive	Directory where Hive Metastore Server will place its log files.
HiveServer2 Log Directory	HiveServer2 Default Group /data/disk1/log/hive	Directory where HiveServer2 will place its log files.



WebHCat Server Log Directory	WebHCat Server Default Group /data/disk1/log/hcatalog	Directory where WebHCat Server will place its log files.
Hue Server Log Directory	Hue Server Default Group /data/disk1/log/hue	Directory where Hue Server will place its log files.
Kerberos Ticket Renewer Log Directory	Kerberos Ticket Renewer Default Group /data/disk1/log/hue	Directory where Kerberos Ticket Renewer will place its log files.
Catalog Server Log Directory log_dir	Impala Catalog Server Default Group /data/disk1/log/catalogd	Directory where Catalog Server will place its log files.
Impala Daemon Log Directory log_dir	Impala Daemon Default Group /data/disk1/log/impalad Impala Daemon Group 1 /data/disk1/log/impalad	Directory where Impala Daemon will place its log files.
Llama Log Directory llama_log_dir	Impala Llama ApplicationMaster Default Group /data/disk1/log/impala-llama	Directory where Llama will place its log files.
StateStore Log Directory log_dir	Impala StateStore Default Group /data/disk1/log/statestore	Directory where StateStore will place its log files.
JobTracker Log Directory hadoop.log.dir	JobTracker Default Group /data/disk1/log/hadoop-0.20-mapreduce	Directory where JobTracker will place its log files.
TaskTracker Log Directory hadoop.log.dir	TaskTracker Default Group /data/disk1/log/hadoop-0.20-mapreduce TaskTracker Group 1 /data/disk1/log/hadoop-0.20-mapreduce	Directory where TaskTracker will place its log files.
Oozie Server Log Directory	Oozie Server Default Group /data/disk1/log/oozie	Directory where Oozie Server will place its log files.
Solr Server Log Directory	Solr Server Default Group /data/disk1/log/solr	Directory where Solr Server will place its log files.
History Server Log Directory log_dir	History Server Default Group /data/disk1/log/spark	The log directory for log files of the role History Server.
Sqoop 2 Server Log Directory	Sqoop 2 Server Default Group /data/disk1/log/sqoop2	Directory where Sqoop 2 Server will place its log files.
JobHistory Server Log Directory hadoop.log.dir	JobHistory Server Default Group /data/disk1/hadoop-mapreduce	Directory where JobHistory Server will place its log files.
NodeManager Log Directory hadoop.log.dir	NodeManager Default Group /data/disk1/log/hadoop-yarn	Directory where NodeManager will place its log files.
ResourceManager Log Directory hadoop.log.dir	ResourceManager Default Group 	Directory where ResourceManager will place its log files.



ZooKeeper Log Directory	Server Default Group /data/disk1/log/zookeeper	Directory where ZooKeeper will place its log files.

Display 25 Entries | First Previous 1 2 Next Last

3. Click Save.

Summary

When building an infrastructure to enable this modernized architecture which could scale to thousands of nodes, operational efficiency can't be an afterthought.

To bring in seamless operation of the application at this scale, you need:

- Infrastructure automation of Cisco UCS servers with service profiles and Cisco Data Center network automation with application profiles with Cisco ACI
- Centralized Management and Deep telemetry and Simplified granular trouble-shooting capabilities and · Multi-tenancy allowing application workloads including containers, micro-services, with the right level of security and SLA for each workload.
- Cisco UCS with Cisco Intersight and Cisco ACI can enable this cloud scale architecture deployed and managed with ease

For More Information

For additional information, see the following resources:

- To find out more about Cisco UCS big data solutions, see http://www.cisco.com/go/bigdata.
- To find out more about Cisco UCS big data validated designs, see http://www.cisco.com/go/bigdata_design
- To find out more about Cisco UCS AI/ML solutions, see http://www.cisco.com/go/ai-compute
- To find out more about Cisco ACI solutions, see http://www.cisco.com/go/aci
- To find out more about Cisco validated solutions based on Software Defined Storage, see https://www.cisco.com/c/en/us/solutions/data-center-virtualization/software-defined-storage-solutions/index.html

Bill of Materials

This section provides the BOM for the 16 Nodes Hadoop Base Rack, 6 Nodes Data Science (AI/ML) Rack and 8 Nodes Hadoop tiered storage. See Table 12 for BOM for the Hadoop Base rack, Table 13 for BOM for Data Science(AI/ML) Expansion Rack, Table 14 for BOM for Hadoop Tiered Storage. Table 15 , Table 16 , and Table 17 for software components. Table 18 lists Cloudera SKUs available from Cisco.

If UCS-SP-CPA4-P2 is added to the BOM, all the required components for only 16 servers are automatically added. If not, you can pick each of the individual components that are specified after this and build the BOM manually.

Table 12 Bill of Materials for Cisco UCS C240M5SX Hadoop Nodes Base Rack

Part Number	Description	Qty
UCS-SP-C240M5-A2	SP C240 M5SX w/2x6132,6x32GB mem,VIC1387	16
CON-OSP-C240M5A2	SNTC 24X7X4OS UCS C240 M5 A2	16
UCS-CPU-I6230	2.1 GHz 6230/125W 20C/27.5MB Cache/DDR4 2933MHz	32
UCS-MR-X32G2RT-H	32GB DDR4-2933-MHz RDIMM/2Rx4/1.2v	192
UCSC-PCI-1-C240M5	Riser 1 including 3 PCIe slots (x8, x16, x8); slot 3 required CPU2	16
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	16
UCSC-PSU1-1600W	Cisco UCS 1600W AC Power Supply for Rack Server	32
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	32
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 & C240 M4 & M5 rack servers	16
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	16
UCSC-HS-C240M5	Heat sink for UCS C240 M5 rack servers 150W CPUs & below	32
UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	416
UCSC-PCIF-240M5	C240 M5 PCIe Riser Blanking Panel	16
CBL-SC-MR12GM5P	Super Cap cable for UCSC-RAID-M5HD	16
UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	16
UCSC-RAID-M5HD	Cisco 12G Modular RAID controller with 4GB cache	16
UCS-SP-FI6332-2X	UCS SP Select 2 x 6332 FI	1
UCS-SP-FI6332	(Not sold standalone) UCS 6332 1RU FI/12 QSFP+	2
CON-OSP-SPFI6332	ONSITE 24X7X4 (Not sold standalone) UCS 6332 1RU FI/No PSU/3	2

Part Number	Description	Qty
UCS-PSU-6332-AC	UCS 6332 Power Supply/100-240VAC	4
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	4
QSFP-H40G-CU3M	40GBASE-CR4 Passive Copper Cable, 3m	16
QSFP-40G-SR-BD	QSFP40G BiDi Short-reach Transceiver	8
N10-MGT015	UCS Manager v3.2(1)	2
UCS-ACC-6332	UCS 6332 Chassis Accessory Kit	2
UCS-FAN-6332	UCS 6332 Fan Module	8
QSFP-H40G-CU3M=	40GBASE-CR4 Passive Copper Cable, 3m	32
UCS-HD24TB10K4KN	2.4 TB 12G SAS 10K RPM SFF HDD (4K)	572

Table 13 Bill of Materials for Hadoop Nodes Expansion Rack

Part Number	Description	Qty
UCS-SP-C240M5-A2	SP C240 M5SX w/2x6132,6x32GB mem,VIC1387	6
CON-OSP-C240M5A2	SNTC 24X7X4OS UCS C240 M5 A2	6
UCS-CPU-6132	2.6 GHz 6132/140W 14C/19.25MB Cache/DDR4 2666MHz	16
UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	48
UCSC-PCI-1-C240M5	Riser 1 including 3 PCIe slots (x8, x16, x8); slot 3 required CPU2	8
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	8
UCSC-PSU1-1600W	Cisco UCS 1600W AC Power Supply for Rack Server	16
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	16
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 & C240 M4 & M5 rack servers	8
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	8
UCSC-HS-C240M5	Heat sink for UCS C240 M5 rack servers 150W CPUs and below	16
UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	208
UCSC-PCIF-240M5	C240 M5 PCIe Riser Blanking Panel	8
CBL-SC-MR12GM5P	Super Cap cable for UCSC-RAID-M5HD	8
UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	8
UCSC-RAID-M5HD	Cisco 12G Modular RAID controller with 4GB cache	8

Part Number	Description	Qty
UCS-SP-H1P8TB-4X	UCS SP 1.8 TB 12G SAS 10K RPM SFF HDD (4K) 4Pk	48
UCS-SP-H1P8TB	1.8 TB 12G SAS 10K RPM SFF HDD (4K)	192
UCS-SP-HD-1P8T-2	1.8TB 12G SAS 10K RPM SFF HDD (4K) 2 Pack	8
UCS-SP-HD-1P8T	SP 1.8TB 12G SAS 10K RPM SFF HDD (4K)	16

Table 14 Bill of Materials for Hadoop Nodes Tiered Storage Rack

Part Number	Description	Qty
UCSS-S3260	Cisco UCS S3260 Storage Server Base Chassis	4
CON-OSP-UCSS3260	SNTC 24X7X4OS, Cisco UCS S3260 Storage Server Base Chassis	4
UCSC-PSU1-1050W	Cisco UCS 1050W AC Power Supply for Rack Server	16
CAB-N5K6A-NA	Power Cord, 200/240V 6A North America	16
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	4
UCSC-C3X60-RAIL	UCS C3X60 Rack Rails Kit	4
UCSS-S3260-BBEZEL	Cisco UCS S3260 Bezel	4
N20-BBLKD-7MM	UCS 7MM SSD Blank Filler	8
N20-BKVM	KVM local IO cable for UCS servers console port	8
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	4
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	4
UCS-S3260-M5SRB	UCS S3260 M5 Server Node for Intel Scalable CPUs	4
UCS-S3260-DRAID	UCS S3260 Dual Raid based on LSI 3316	4
UCS-S3260-M5HS	UCS S3260 M5 Server Node HeatSink	8
UCS-S3260-M5SRB	UCS S3260 M5 Server Node for Intel Scalable CPUs	4
UCS-S3260-DRAID	UCS S3260 Dual Raid based on LSI 3316	4
UCS-S3260-M5HS	UCS S3260 M5 Server Node HeatSink	8
UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	48
UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	48
UCS-C3K-HD4TB	UCS C3000 4TB NL-SAS 7200 RPM 12Gb HDD w Carrier-Top Load	224

Part Number	Description	Qty
UCS-S3260-G3SD48	UCS S3260 480G Boot SSD (Micron 6G SATA)	16
UCS-S3260-56HD4	Cisco UCS C3X60 Four row of drives containing 56 x 4TB	4
UCS-CPU-I5220	Intel 5220 2.2GHz/125W 18C/24.75MB DCP DDR4 2666 MHz	8
UCS-CPU-I5220	Intel 5220 2.2GHz/125W 18C/24.75MB DCP DDR4 2666 MHz	8
RACK2-UCS2	Cisco R42612 standard rack, w/side panels	1
CON-SNT-RCK2UCS2	SNTC 8X5XNBD, Cisco R42612 standard rack, w side panels	1

Table 15 Bill of Materials for CDSW Nodes Rack

Part Number	Description	Qty
UCS-SP-C240M5-A2	SP C240 M5SX w/2x6132,6x32GB mem, VIC1387	6
CON-OSP-C240M5A2	SNTC 24X7X4OS UCS C240 M5 A2	6
UCS-CPU-I6230	2.1 GHz 6230/125W 20C/27.5MB Cache/DDR4 2933MHz	12
UCS-MR-X32G2RT-H	32GB DDR4-2933-MHz RDIMM/2Rx4/1.2v	72
UCSC-PCI-1-C240M5	Riser 1 including 3 PCIe slots (x8, x16, x8); slot 3 required CPU2	6
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	6
UCSC-PSU1-1600W	Cisco UCS 1600W AC Power Supply for Rack Server	12
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	12
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220, C240 M4 and M5 rack servers	6
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	6
UCSC-HS-C240M5	Heat sink for UCS C240 M5 rack servers 150W CPUs & below	12
UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	156
CBL-SC-MR12GM5P	Super Cap cable for UCSC-RAID-M5HD	6
UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	6
UCSC-RAID-M5HD	Cisco 12G Modular RAID controller with 4GB cache	6
UCSC-PCI-2A-240M5	Riser 2A including 3 PCIe slots (x8, x16, x16) supports GPU	6
UCS-SP-SD-1P6TB-4X	UCS SP 1.6TB 2.5inch Ent. Perf 12G SAS SSD	6

Part Number	Description	Qty
	(10Xendurance)4Pk	
UCS-SP-SD-1P6TB	1.6TB 2.5inch Ent. Performance 12G SAS SSD (10X endurance)	24
UCSC-GPU-T4-16=	NVIDIA Tesla T4 16GB	12
CON-SNT-CGPUT416	SNTC-24X7X4OS NVIDIA T4 PCIE 75W 16GB	12

Table 16 Red Hat Enterprise Linux License

Part Number	Description	Qty
RHEL-2S2V-3A	Red Hat Enterprise Linux	30
CON-ISV1-EL2S2V3A	3 year Support for Red Hat Enterprise Linux	30

Table 17 Cloudera Data Science Workbench Software

UCS-BD-CDSWB=	UCS-BD-CDSWB-1Y	Cloudera Data Science Work Bench, 10-user pack - 1 Year
UCS-BD-CDSWB=	UCS-BD-CDSWB-2Y	Cloudera Data Science Work Bench, 10-user pack - 2 Year
UCS-BD-CDSWB=	UCS-BD-CDSWB-3Y	Cloudera Data Science Work Bench, 10-user pack - 3 Year

Table 18 Cloudera SKU's Available at Cisco

Cisco TOP SKU	Cisco PID with Duration	Product Name
UCS-BD-CEBN-BZ=	UCS-BD-CEBN-BZ-3Y	Cloudera Enterprise Basic Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEBN-BZI=	UCS-BD-CEBN-BZI-3Y	Cloudera Enterprise Basic Edition + Indemnification, Node License, Bronze Support - 3 Year
UCS-BD-CEBN-GD=	UCS-BD-CEBN-GD-3Y	Cloudera Enterprise Basic Edition, Node License, Gold Support - 3 Year
UCS-BD-CEBN-GDI=	UCS-BD-CEBN-GDI-3Y	Cloudera Enterprise Basic Edition + Indemnification, Node License, Gold Support - 3 Year
UCS-BD-CEDEN-BZ=	UCS-BD-CEDEN-BZ-3Y	Cloudera Enterprise Data Engineering Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEDEN-GD=	UCS-BD-CEDEN-GD-3Y	Cloudera Enterprise Data Engineering Edition, Node License, Gold Support - 3 Year
UCS-BD-CEODN-BZ=	UCS-BD-CEODN-BZ-3Y	Cloudera Enterprise Operational Database Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEODN-GD=	UCS-BD-CEODN-GD-2Y	Cloudera Enterprise Operational Database Edition, Node License, Gold Support - 2 Year
UCS-BD-CEODN-GD=	UCS-BD-CEODN-GD-3Y	Cloudera Enterprise Operational Database Edition, Node License, Gold Support - 3 Year
UCS-BD-CEADN-	UCS-BD-CEADN-BZ-3Y	Cloudera Enterprise Analytical Database Edition, Node License,

Cisco TOP SKU	Cisco PID with Duration	Product Name
BZ=		Bronze Support - 3 Year
UCS-BD-CEADN-GD=	UCS-BD-CEADN-GD-3Y	Cloudera Enterprise Analytical Database Edition, Node License, Gold Support - 3 Year
UCS-BD-CEDHN-BZ=	UCS-BD-CEDHN-BZ-3Y	Cloudera Enterprise Data Hub Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEDHN-GD=	UCS-BD-CEDHN-GD-3Y	Cloudera Enterprise Data Hub Edition, Node License, Gold Support - 3 Year

Appendix

Configure Cisco Boot Optimized M.2 RAID Controller

Beginning with 4.0(4a), Cisco UCS Manager supports Cisco boot optimized M.2 RAID controller (UCS-M2-HWRAID), which is based on Marvell® 88SE92xx PCIe to SATA 6Gb/s controller.

The following M.2 drives are managed by the Cisco boot optimized M.2 RAID controller:

- 240GB M.2 6G SATA SSD
- 960GB M.2 6G SATA SSD

The Cisco boot optimized M.2 RAID controller supports only RAID1/JBOD (default - JBOD) mode and only UEFI boot mode.

The following are the limitations of the Cisco boot optimized M.2 RAID controller:

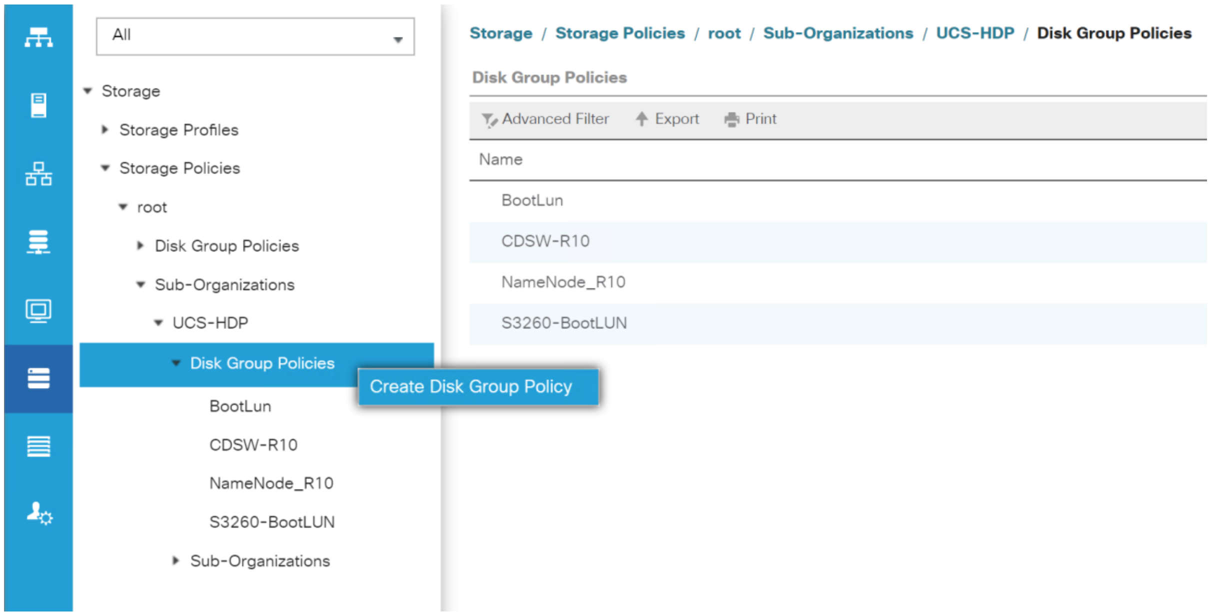
- Existing LUN migration is not supported.
- Local Disk Configuration policy is not supported.
- Entire disk capacity is used while creating single LUN.
- LUN is created using the Local LUN tab (see Configuring Local LUNs) under storage profile and not using the controller definitions.
- You cannot mix different capacity M.2 drives.

To create a `Disk Group Policy` and `Storage Profile Policy` to be attach with `Service Profile` for Cisco Optimized M.2 RAID Controller follow the steps in the following sections.

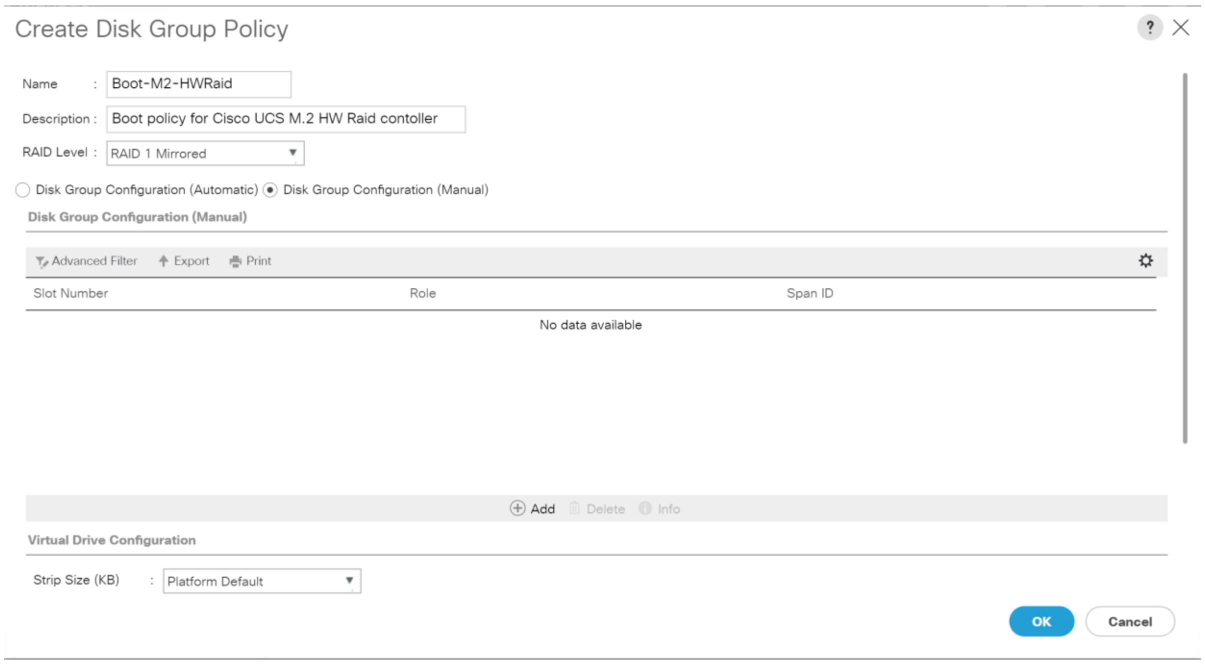
Configure Disk Group Policy

To configure the disk group policy, follow these steps:

4. In the UCSM WebUI, Go to storage tab. In the Storage Policy section, right-click Disk Group Policies. Click Create Disk Group Policy.



5. Enter a name and description for the new Disk Group Policy. Select Manual Disk Group Configuration. Click Add.


Create Disk Group Policy

Name : Boot-M2-HWRaid

Description : Boot policy for Cisco UCS M.2 HW Raid contoller

RAID Level : RAID 1 Mirrored

Disk Group Configuration (Automatic) [✓ Disk Group Configuration (Manual)]

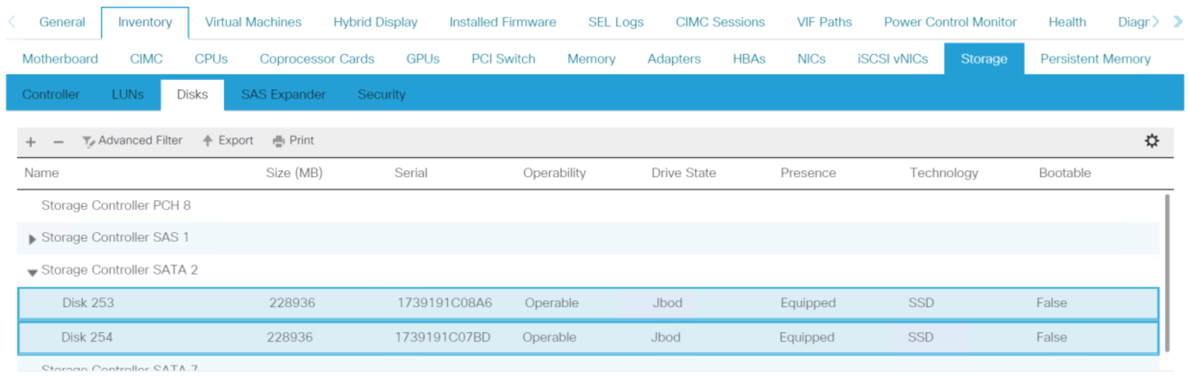
Disk Group Configuration (Manual)

Slot Number	Role	Span ID
No data available		

Virtual Drive Configuration

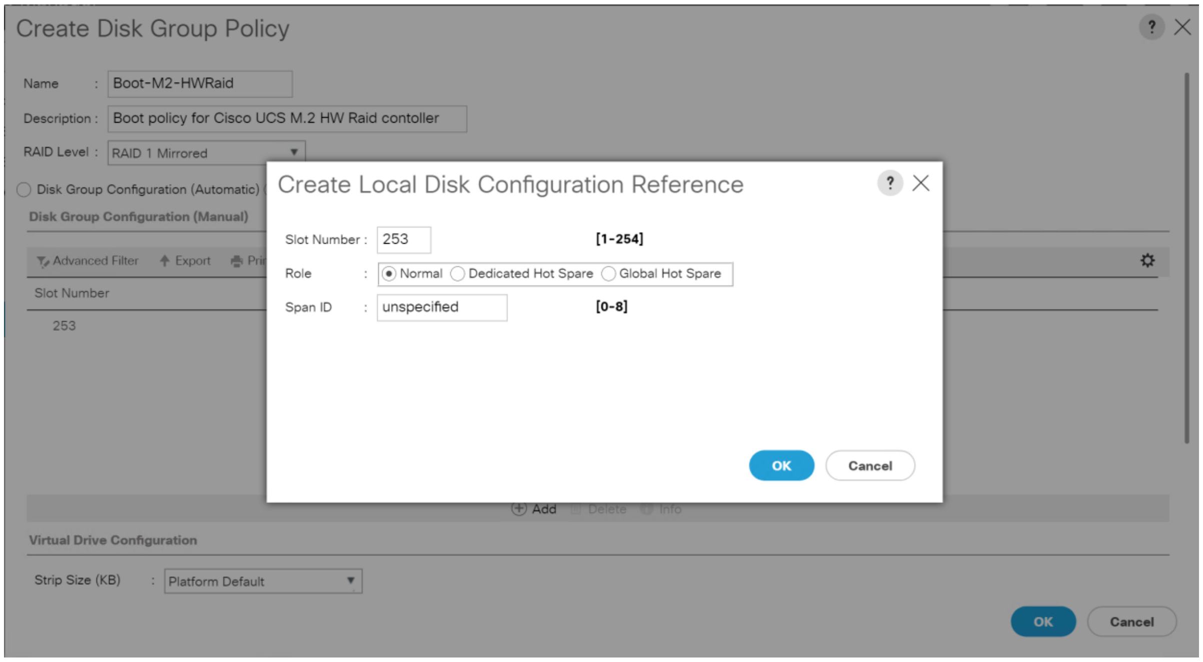
Strip Size (KB) : Platform Default

M.2 disks are allocated Disk slot Number 253 and 254.



Name	Size (MB)	Serial	Operability	Drive State	Presence	Technology	Bootable
Storage Controller PCH 8							
Storage Controller SAS 1							
Storage Controller SATA 2							
Disk 253	228936	1739191C08A6	Operable	Jbod	Equipped	SSD	False
Disk 254	228936	1739191C07BD	Operable	Jbod	Equipped	SSD	False

6. Enter Slot Number 253 for the first disk. Click OK.



Create Disk Group Policy

Name : Boot-M2-HWRaid

Description : Boot policy for Cisco UCS M.2 HW Raid contoller

RAID Level : RAID 1 Mirrored

Disk Group Configuration (Automatic)

Disk Group Configuration (Manual)

Slot Number
253

Virtual Drive Configuration

Strip Size (KB) : Platform Default

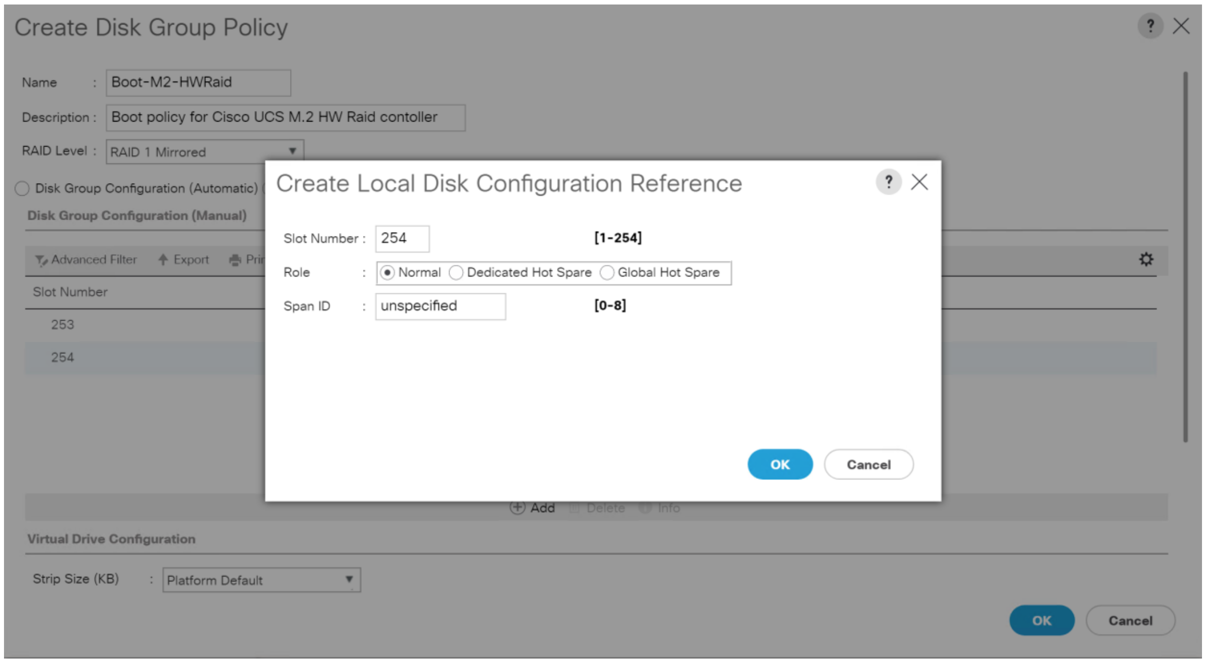
Create Local Disk Configuration Reference

Slot Number : 253 [1-254]

Role : [✓ Normal] Dedicated Hot Spare Global Hot Spare

Span ID : unspecified [0-8]

7. Click Add to add second disk, enter Slot Number 254.



Create Disk Group Policy

Name : Boot-M2-HWRaid

Description : Boot policy for Cisco UCS M.2 HW Raid contoller

RAID Level : RAID 1 Mirrored

Disk Group Configuration (Automatic)

Disk Group Configuration (Manual)

Slot Number
253
254

Create Local Disk Configuration Reference

Slot Number : 254 [1-254]

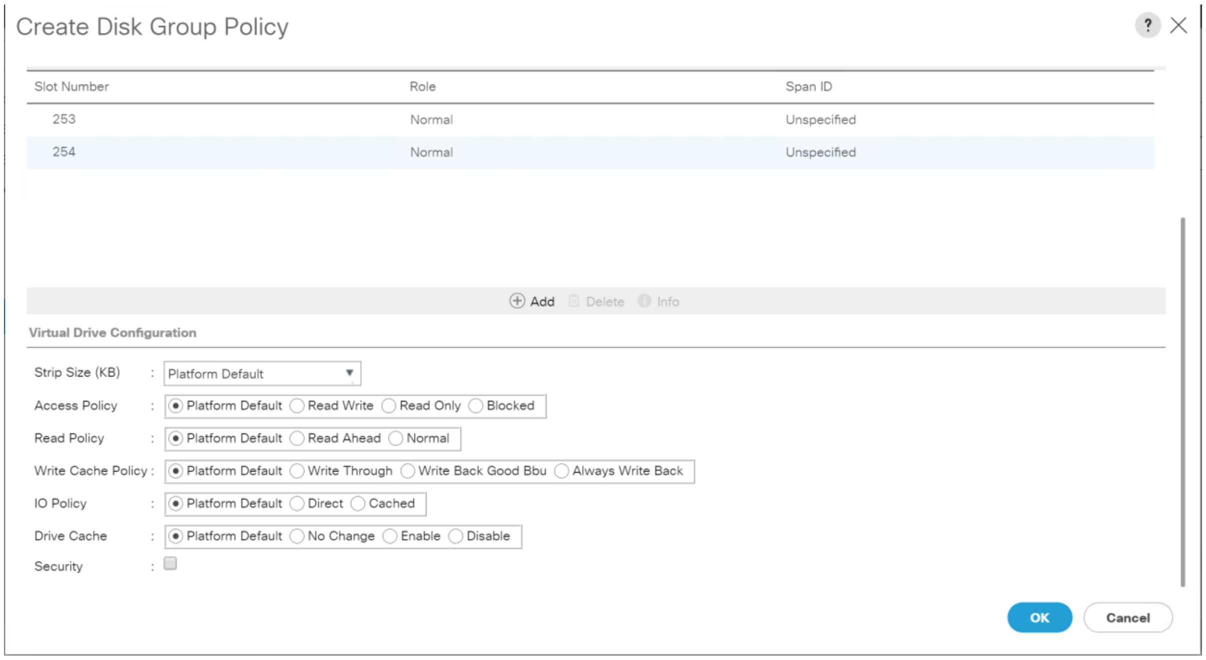
Role : [✓ Normal] Dedicated Hot Spare Global Hot Spare

Span ID : unspecified [0-8]

Virtual Drive Configuration

Strip Size (KB) : Platform Default

8. In Virtual Drive Configuration section leave all option as Platform Default. Click OK.



Create Disk Group Policy

Slot Number	Role	Span ID
253	Normal	Unspecified
254	Normal	Unspecified

Virtual Drive Configuration

Strip Size (KB) : Platform Default

Access Policy : [✓ Platform Default] Read Write Read Only Blocked

Read Policy : [✓ Platform Default] Read Ahead Normal

Write Cache Policy : [✓ Platform Default] Write Through Write Back Good Bbu Always Write Back

IO Policy : [✓ Platform Default] Direct Cached

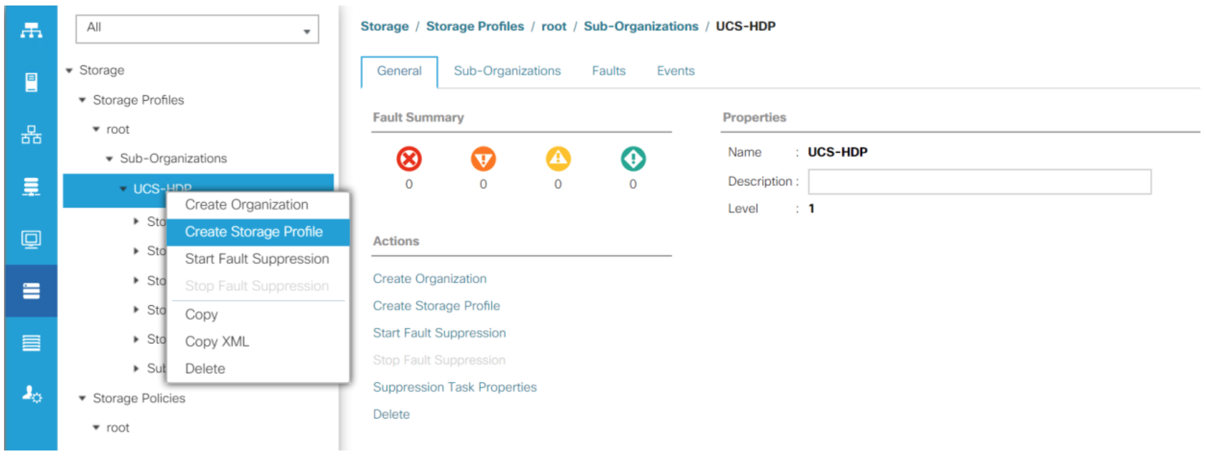
Drive Cache : [✓ Platform Default] No Change Enable Disable

Security : ☐

Configure Storage Profile

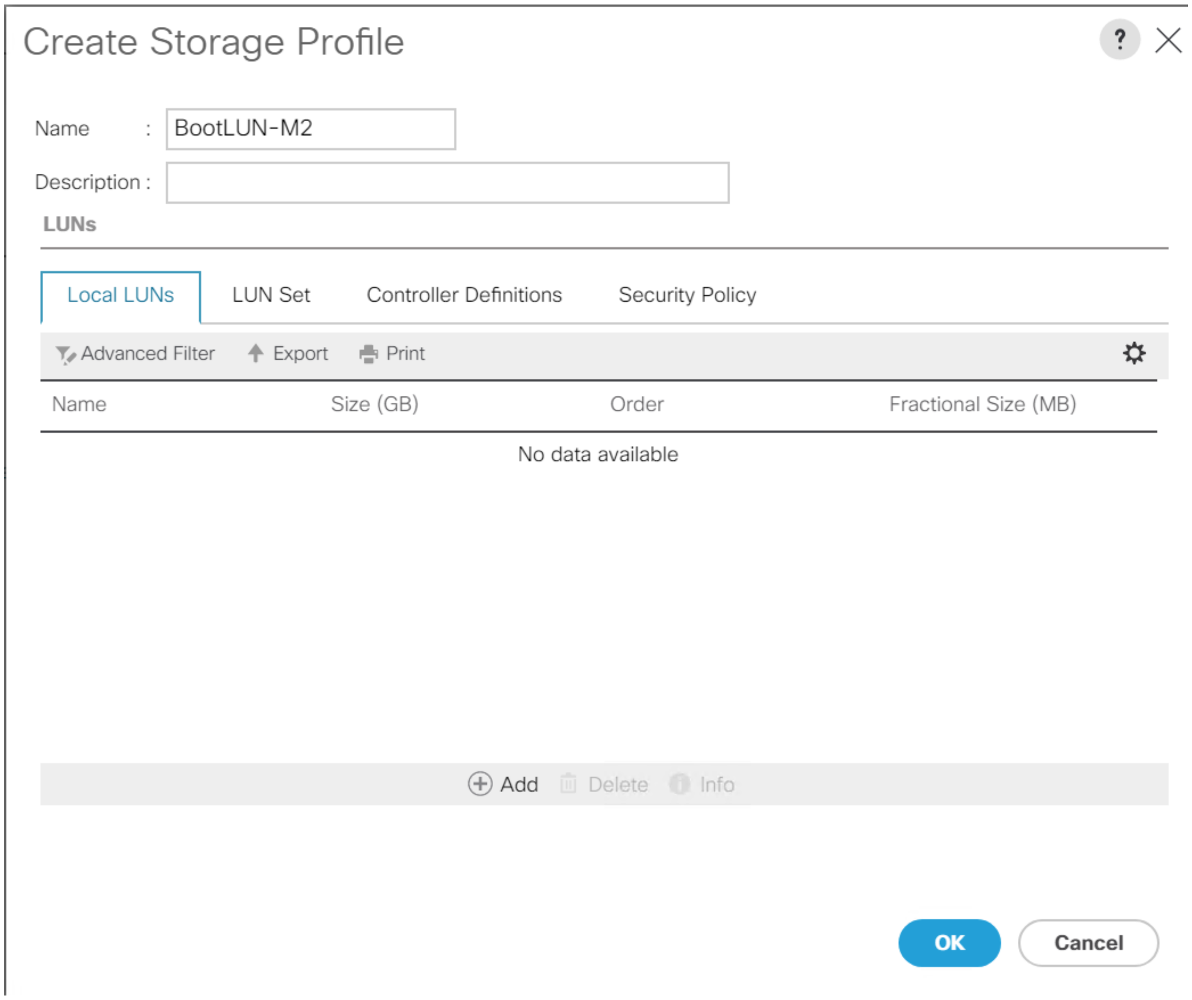
To configure the storage profile, follow these steps:

9. In the Storage Profiles section, select Storage Profiles. Right-click and select Create Storage Profile.



Name : UCS-HDP
Description :
Level : 1

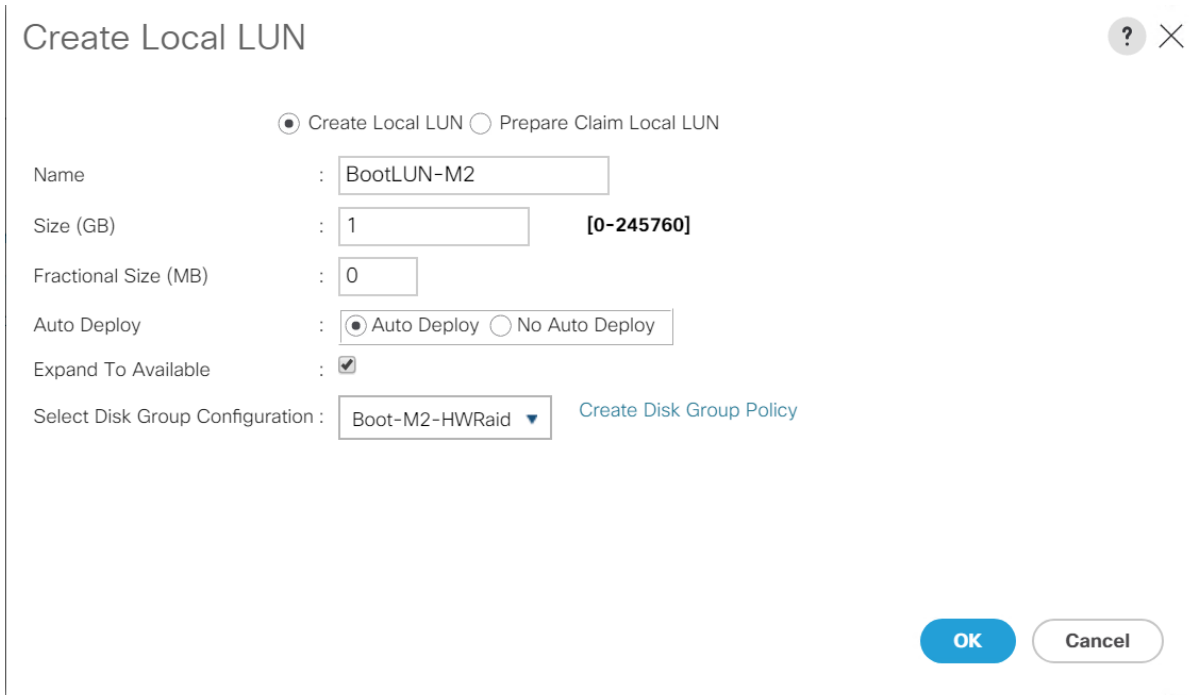
10. Enter a name for the Storage Profile. Click Add.



Name : BootLUN-M2
Description :

Name	Size (GB)	Order	Fractional Size (MB)
No data available			

11. Enter a name for the Local LUN to be created, Click Auto Deploy, check the box for Expand to Available, and from the drop-down list for Disk Group Configuration, select RAID 1 Disk Group Policy created for M.2 SATA Disks. Click OK.


Create Local LUN

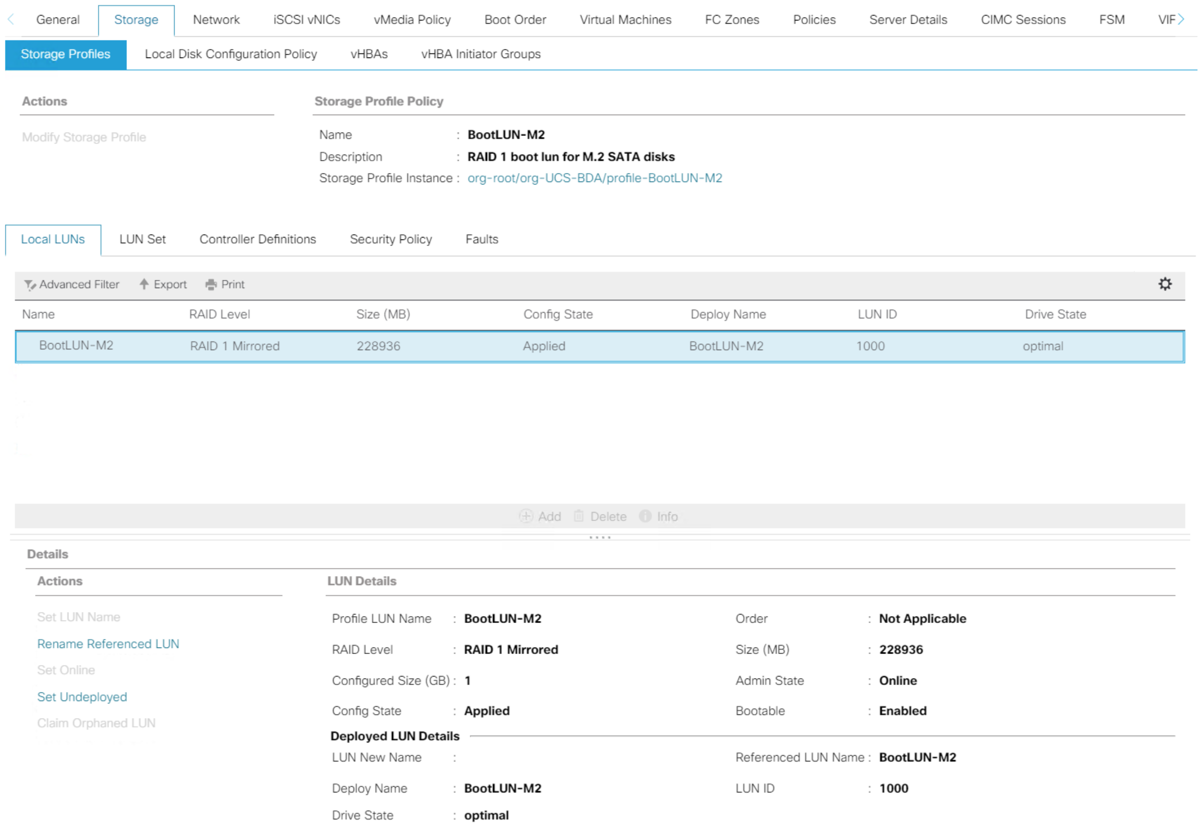
[✓ Create Local LUN] Prepare Claim Local LUN

Name :	BootLUN-M2
Size (GB) :	1 [0-245760]
Fractional Size (MB) :	0
Auto Deploy :	[✓ Auto Deploy] No Auto Deploy
Expand To Available :	✓
Select Disk Group Configuration :	Boot-M2-HWRaid

12. Attach a Storage Profile created to a Service profile or create a new Service Profile.

13. Go to the Storage tab on the Service Profile, select Storage Profile. Select Modify Storage Profile. Select Storage Profile created for M.2 SATA Disks.

Figure 63 Example of the Service Profile Associated to a Cisco UCS C240 M5 server with Cisco UCS-M2-HWRAID and 2 240GB M.2 SATA SSD Installed



Storage Profile Policy

Name	BootLUN-M2
Description	RAID 1 boot lun for M.2 SATA disks
Storage Profile Instance	org-root/org-UCS-BDA/profile-BootLUN-M2

Name	RAID Level	Size (MB)	Config State	Deploy Name	LUN ID	Drive State
BootLUN-M2	RAID 1 Mirrored	228936	Applied	BootLUN-M2	1000	optimal

Details

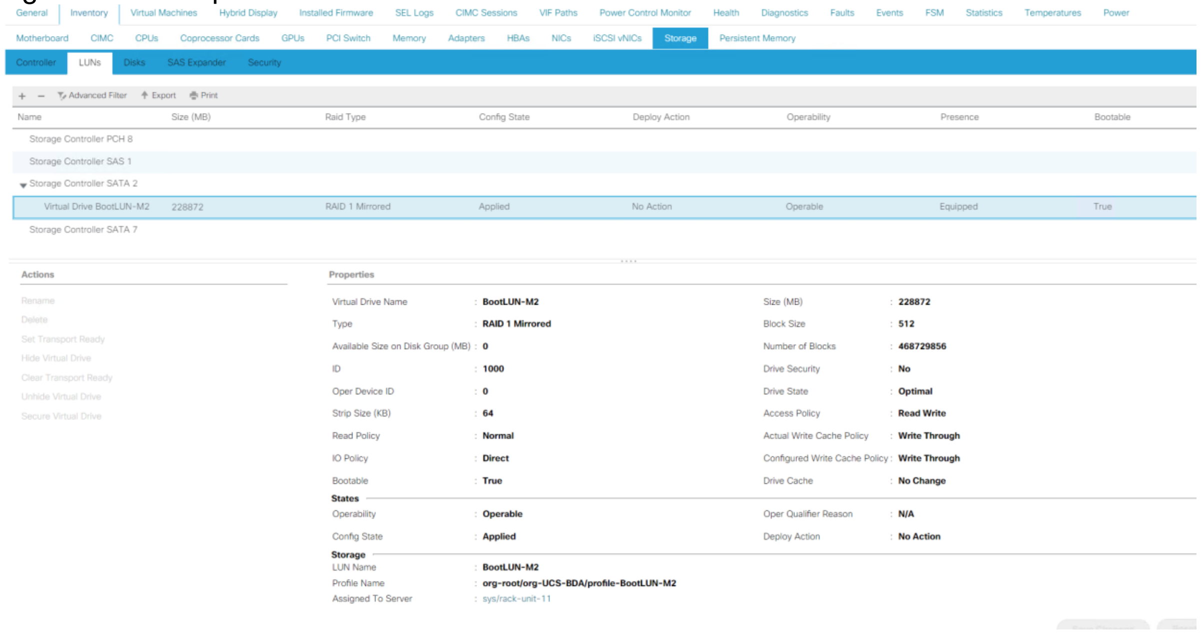
LUN Details

Profile LUN Name	BootLUN-M2	Order	Not Applicable
RAID Level	RAID 1 Mirrored	Size (MB)	228936
Configured Size (GB)	1	Admin State	Online
Config State	Applied	Bootable	Enabled

Deployed LUN Details

LUN New Name		Referenced LUN Name	BootLUN-M2
Deploy Name	BootLUN-M2	LUN ID	1000
Drive State	optimal		

Figure 64 Example of Virtual Drive Created from 2 M.2 SATA SSD



Name	Size (MB)	Raid Type	Config State	Deploy Action	Operability	Presence	Bootable
Storage Controller PCH 8							
Storage Controller SAS 1							
Storage Controller SATA 2							
Virtual Drive BootLUN-M2	228872	RAID 1 Mirrored	Applied	No Action	Operable	Equipped	True
Storage Controller SATA 7							

Properties

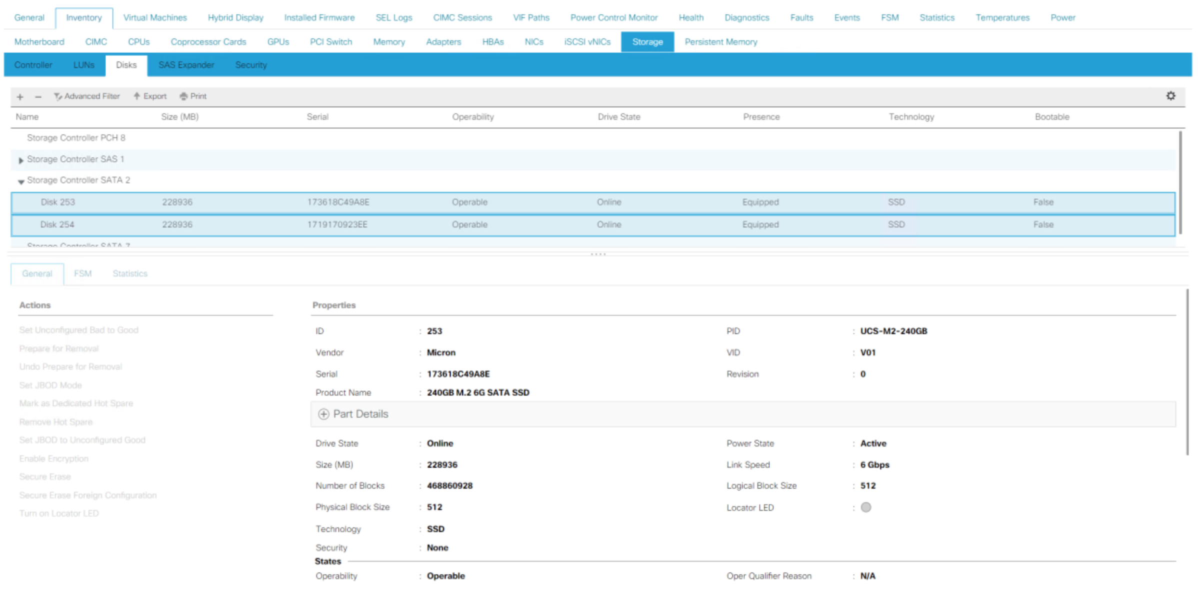
Virtual Drive Name	BootLUN-M2	Size (MB)	228872
Type	RAID 1 Mirrored	Block Size	512
Available Size on Disk Group (MB)	0	Number of Blocks	468729856
ID	1000	Drive Security	No
Oper Device ID	0	Drive State	Optimal
Strip Size (KB)	64	Access Policy	Read Write
Read Policy	Normal	Actual Write Cache Policy	Write Through
IO Policy	Direct	Configured Write Cache Policy	Write Through
Bootable	True	Drive Cache	No Change

States

Operability	Operable	Oper Qualifier Reason	N/A
Config State	Applied	Deploy Action	No Action

Storage

LUN Name	BootLUN-M2
Profile Name	org-root/org-UCS-BDA/profile-BootLUN-M2
Assigned To Server	sys/rack-unit-11



Name	Size (MB)	Serial	Operability	Drive State	Presence	Technology	Bootable
Storage Controller PCH 8							
Storage Controller SAS 1							
Storage Controller SATA 2							
Disk 253	228936	173618C49A8E	Operable	Online	Equipped	SSD	False
Disk 254	228936	1719170923EE	Operable	Online	Equipped	SSD	False

Actions

Set Unconfigured Bad to Good
Prepare for Removal
Undo Prepare for Removal
Set JBOD Mode
Mark as Dedicated Hot Spare
Remove Hot Spare
Set JBOD to Unconfigured Good
Enable Encryption
Secure Erase
Secure Erase Foreign Configuration
Turn on Locator LED

Properties

ID	: 253	PID	: UCS-M2-240GB
Vendor	: Micron	VID	: V01
Serial	: 173618C49A8E	Revision	: 0
Product Name	: 240GB M.2 6G SATA SSD		

Part Details

Drive State	: Online	Power State	: Active
Size (MB)	: 228936	Link Speed	: 6 Gbps
Number of Blocks	: 468860928	Logical Block Size	: 512
Physical Block Size	: 512	Locator LED	: ●
Technology	: SSD		
Security	: None		

States

Operability	: Operable	Oper Qualifier Reason	: N/A

Apply Storage Profile in Service Profile Template

To create a new Service Profile template or update an existing template for Service Profile to attach a newly created Storage Profile for Cisco Boot Optimized RAID Controller, follow these steps:

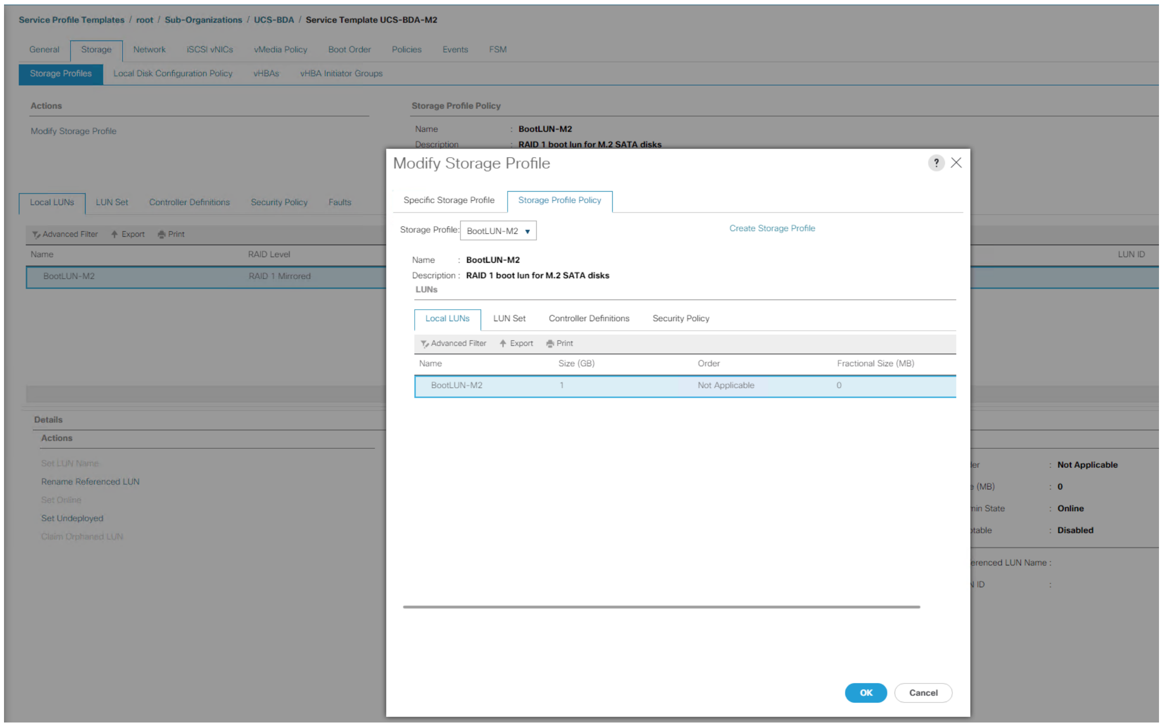
14. Go to Service Profile Template.

15. Select Storage tab in Service Profile Template.

16. Select Storage Profile tab. Click Modify Storage Profile.

17. From the Storage Profile drop-down list, select Storage Profile for Cisco Boot Optimized RAID Controller.

18. If updating a Service Profile Template, once saved the changes in the configuration change in the Service Profile Template and are automatically applied to all Service Profile binded with the template.



Service Profile Templates / root / Sub-Organizations / UCS-BDA / Service Template UCS-BDA-M2

General | Storage | Network | iSCSI vNICs | vMedia Policy | Boot Order | Policies | Events | FSM

Storage Profiles | Local Disk Configuration Policy | vHBAs | vHBA Initiator Groups

Actions: Modify Storage Profile

Storage Profile Policy
Name : BootLUN-M2
Description : RAID 1 boot lun for M.2 SATA disks

Local LUNs | LUN Set | Controller Definitions | Security Policy | Faults

Name	RAID Level	LUN ID
BootLUN-M2	RAID 1 Mirrored	

Details

Actions: Set LUN Name, Rename Referenced LUN, Set Online, Set Undeployed, Claim Orphaned LUN

Order : Not Applicable
Fractional Size (MB) : 0
Admin State : Online
Bootable : Disabled
Referenced LUN Name :
LUN ID :

Modify Storage Profile

Specific Storage Profile | Storage Profile Policy

Storage Profile: BootLUN-M2 | Create Storage Profile

Name : BootLUN-M2
Description : RAID 1 boot lun for M.2 SATA disks

LUNs

Local LUNs | LUN Set | Controller Definitions | Security Policy

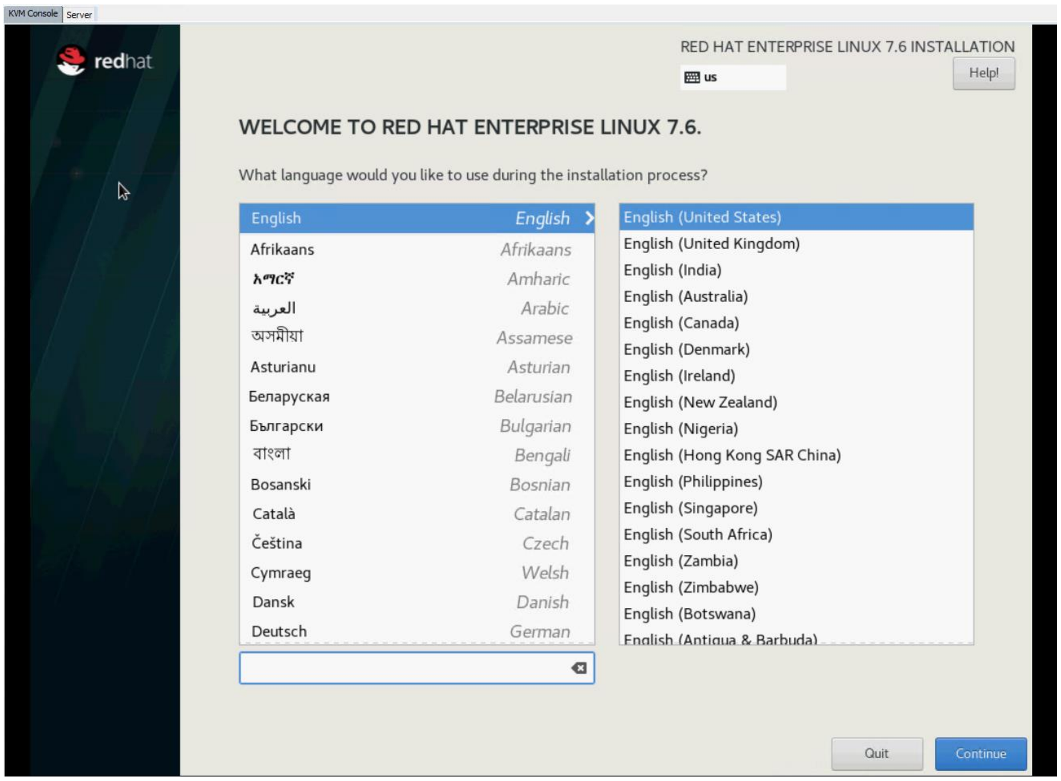
Name	Size (GB)	Order	Fractional Size (MB)
BootLUN-M2	1	Not Applicable	0

OK | Cancel

Install RHEL 7.6 on Cisco Optimized M.2 RAID Controller

To install Red Hat Enterprise Linux 7.6 OS on Cisco UCS server with Virtual Drive created from Cisco Optimized M.2 RAID Controller (UCS-M2-HWRAID) in UEFI Boot Mode, follow these steps:

19. On the Welcome screen, select a language and click Continue.



RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

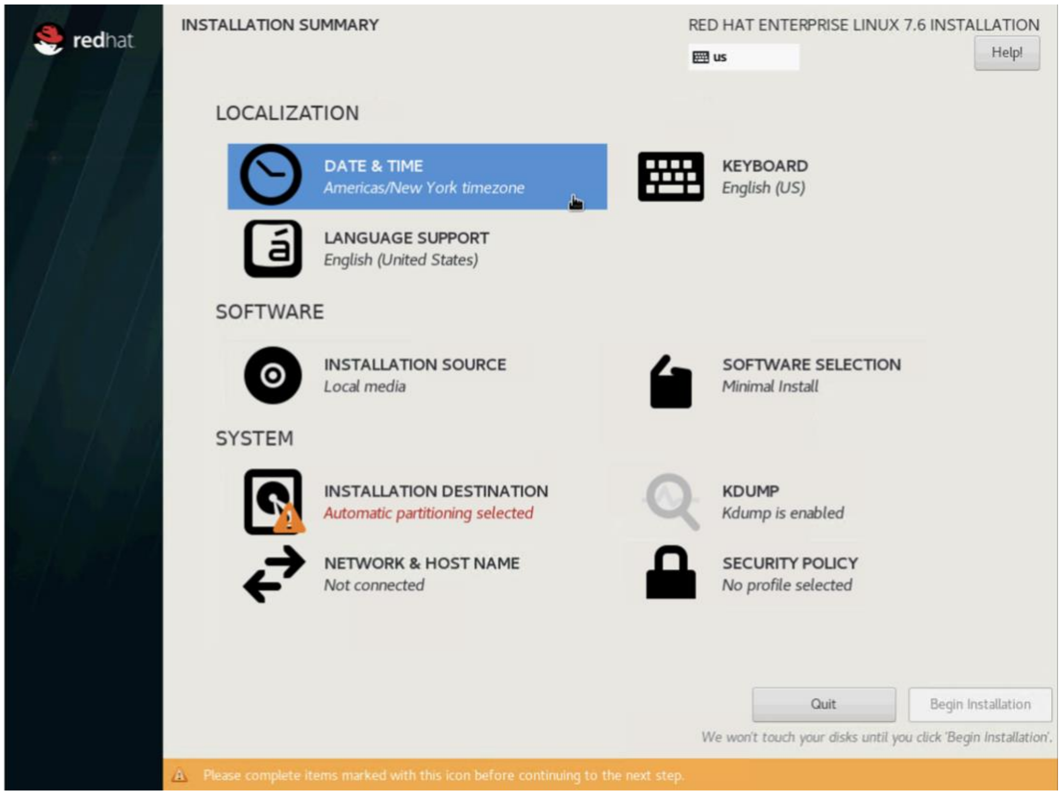
WELCOME TO RED HAT ENTERPRISE LINUX 7.6.

What language would you like to use during the installation process?

English	English
Afrikaans	Afrikaans
አማርኛ	Amharic
العربية	Arabic
অসমীয়া	Assamese
Asturianu	Asturian
Беларуская	Belarusian
Български	Bulgarian
বাংলা	Bengali
Bosanski	Bosnian
Català	Catalan
Čeština	Czech
Cymraeg	Welsh
Dansk	Danish
Deutsch	German

English (United States)
English (United Kingdom)
English (India)
English (Australia)
English (Canada)
English (Denmark)
English (Ireland)
English (New Zealand)
English (Nigeria)
English (Hong Kong SAR China)
English (Philippines)
English (Singapore)
English (South Africa)
English (Zambia)
English (Zimbabwe)
English (Botswana)
English (Antigua & Barbuda)

20. Select DATE & TIME.



INSTALLATION SUMMARY

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

LOCALIZATION

DATE & TIME
Americas/New York timezone

KEYBOARD
English (US)

LANGUAGE SUPPORT
English (United States)

SOFTWARE

INSTALLATION SOURCE
Local media

SOFTWARE SELECTION
Minimal Install

SYSTEM

INSTALLATION DESTINATION
Automatic partitioning selected

KDUMP
Kdump is enabled

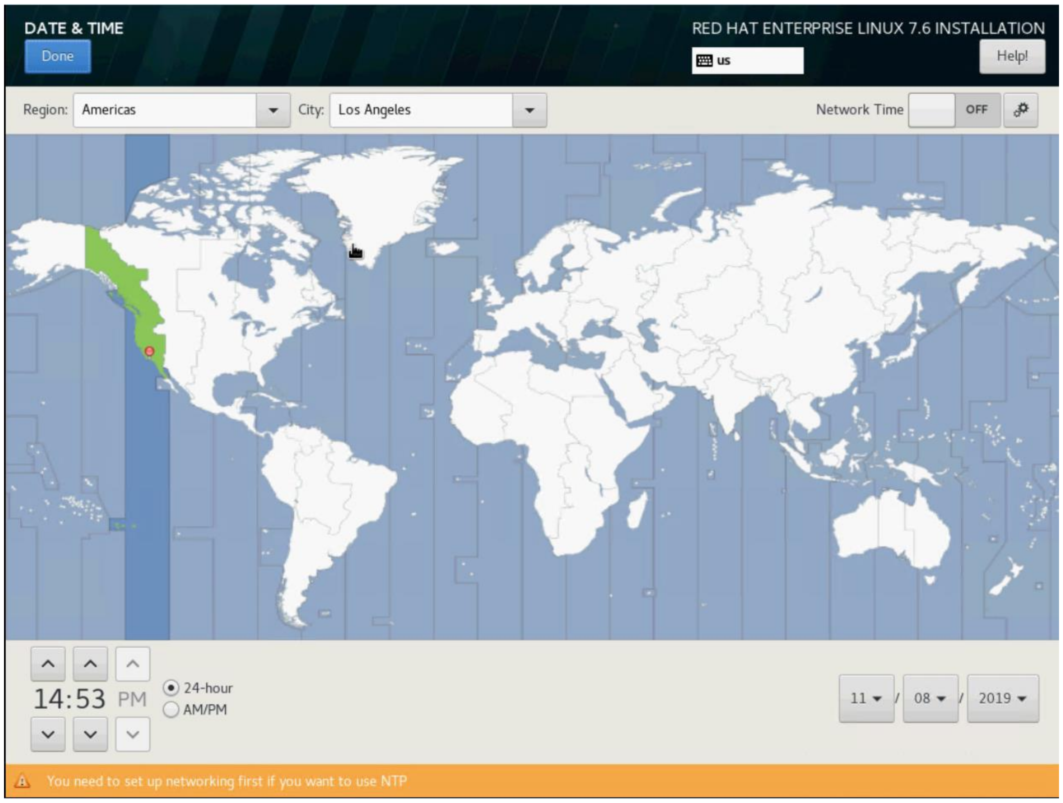
NETWORK & HOST NAME
Not connected

SECURITY POLICY
No profile selected

We won't touch your disks until you click 'Begin Installation'.

Please complete items marked with this icon before continuing to the next step.

21. Select region and City.


DATE & TIME

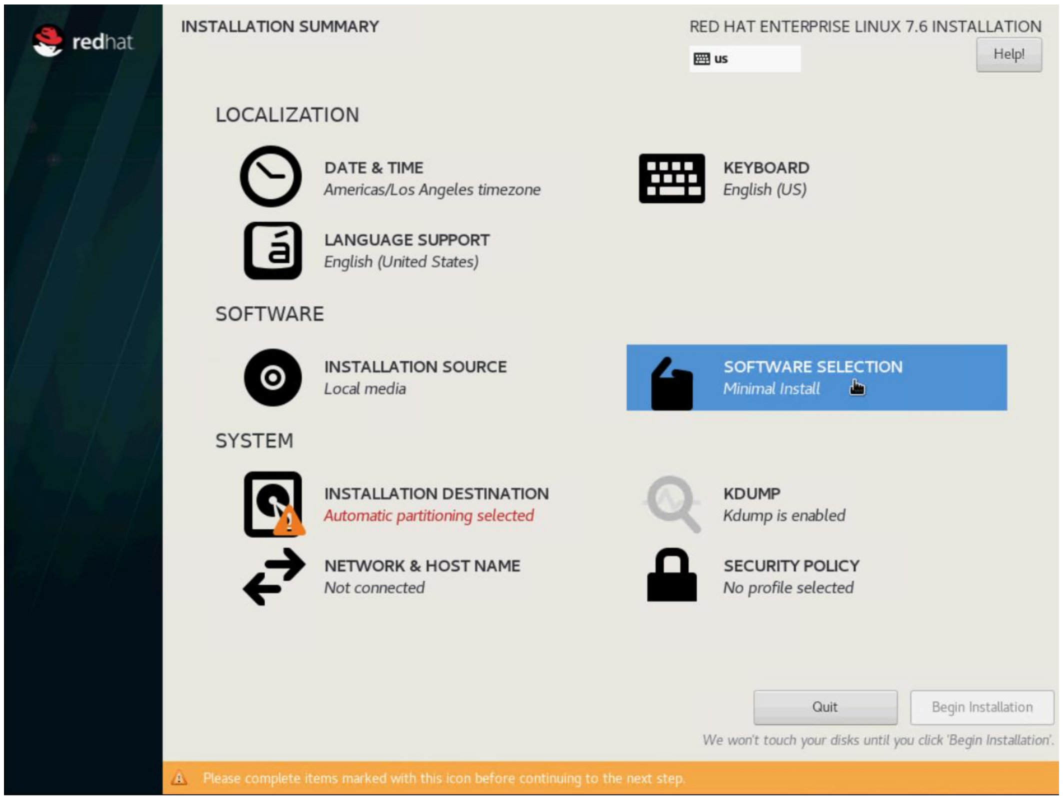
RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

Region: Americas City: Los Angeles Network Time OFF

14:53 PM ◉ 24-hour ○ AM/PM 11 / 08 / 2019

You need to set up networking first if you want to use NTP

22. Select SOFTWARE SELECTION.


INSTALLATION SUMMARY

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

LOCALIZATION

DATE & TIME
Americas/Los Angeles timezone

KEYBOARD
English (US)

LANGUAGE SUPPORT
English (United States)

SOFTWARE

INSTALLATION SOURCE
Local media

SOFTWARE SELECTION
Minimal Install

SYSTEM

INSTALLATION DESTINATION
Automatic partitioning selected

KDUMP
Kdump is enabled

NETWORK & HOST NAME
Not connected

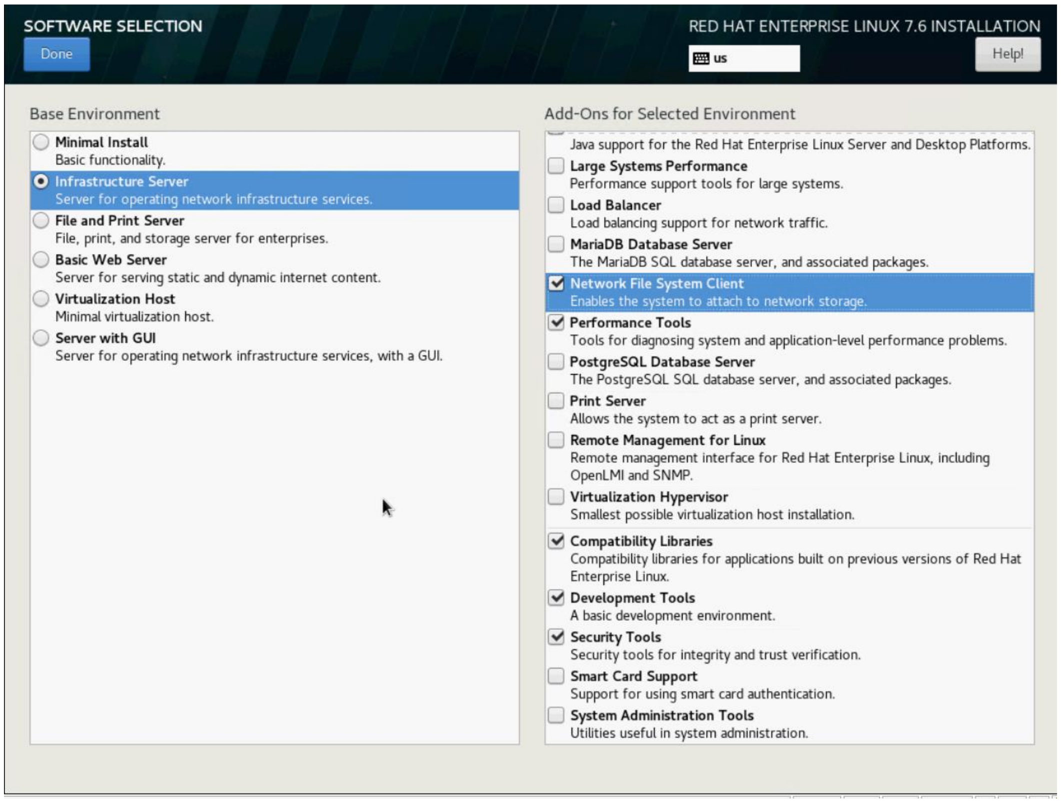
SECURITY POLICY
No profile selected

We won't touch your disks until you click 'Begin Installation'.

Please complete items marked with this icon before continuing to the next step.

23. Select Infrastructure Server in Base Environment. For Add-Ons for the selected environment, select:

- Network File System Client
- Performance Tools
- Compatibility Libraries
- Development Tools
- Security Tools



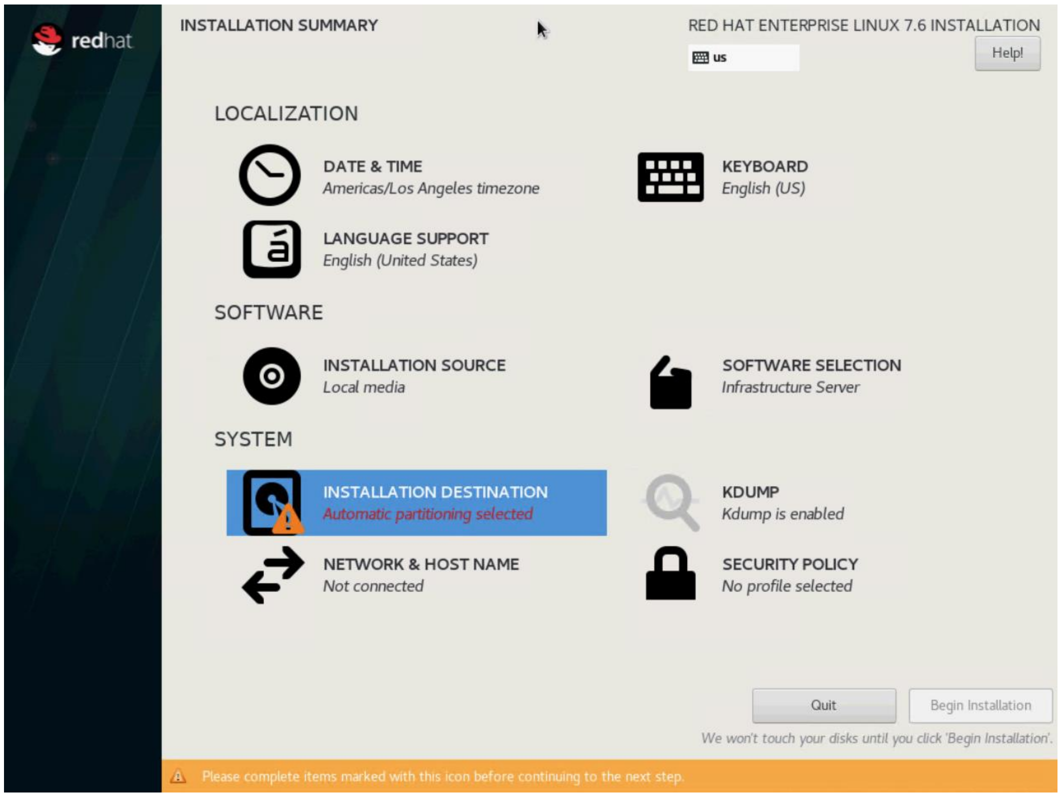
Base Environment

- ○ **Minimal Install** Basic functionality.
- ● **Infrastructure Server** Server for operating network infrastructure services.
- ○ **File and Print Server** File, print, and storage server for enterprises.
- ○ **Basic Web Server** Server for serving static and dynamic internet content.
- ○ **Virtualization Host** Minimal virtualization host.
- ○ **Server with GUI** Server for operating network infrastructure services, with a GUI.

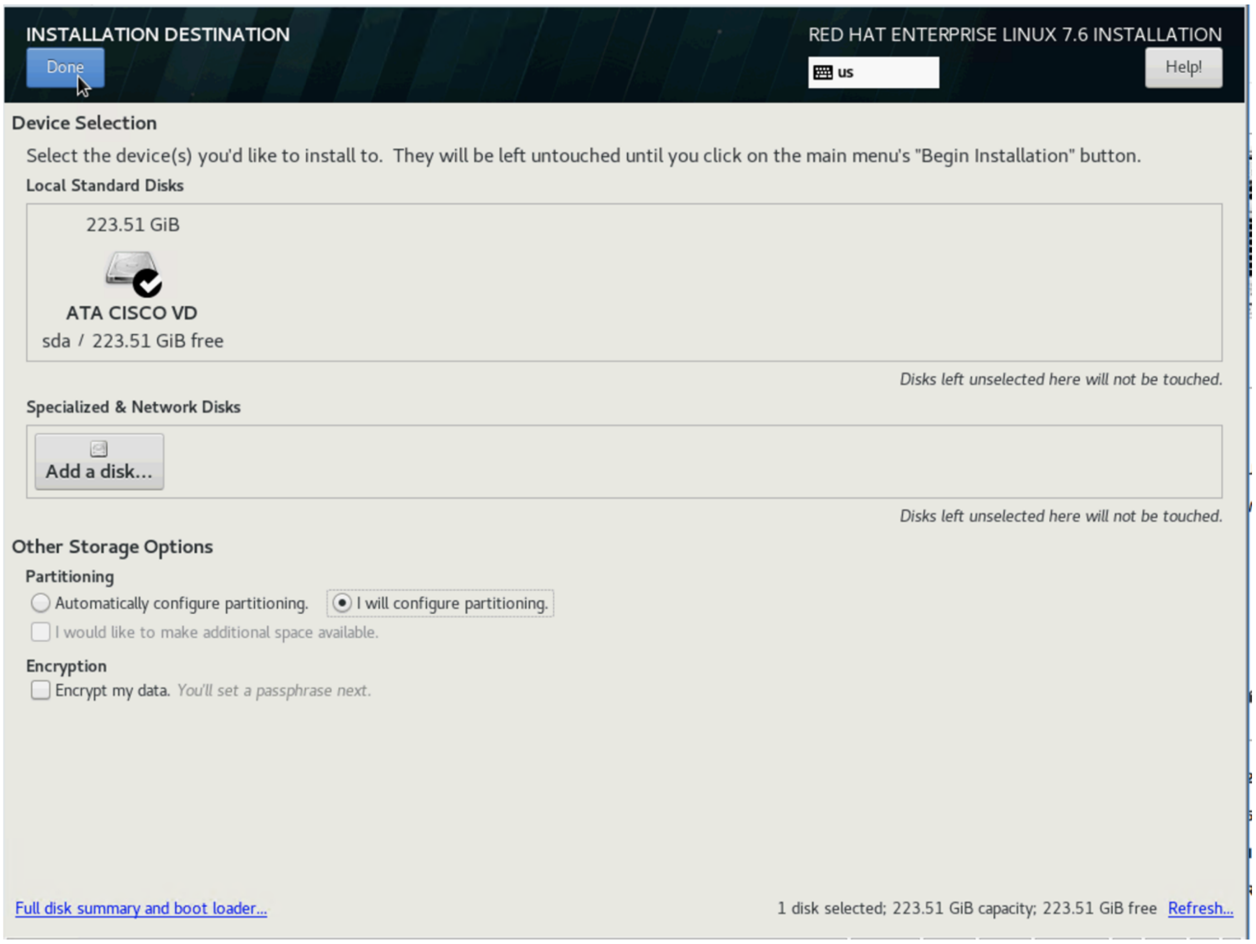
Add-Ons for Selected Environment

- Java support for the Red Hat Enterprise Linux Server and Desktop Platforms.
- ☐ **Large Systems Performance** Performance support tools for large systems.
- ☐ **Load Balancer** Load balancing support for network traffic.
- ☐ **MariaDB Database Server** The MariaDB SQL database server, and associated packages.
- ✓ **Network File System Client** Enables the system to attach to network storage.
- ✓ **Performance Tools** Tools for diagnosing system and application-level performance problems.
- ☐ **PostgreSQL Database Server** The PostgreSQL SQL database server, and associated packages.
- ☐ **Print Server** Allows the system to act as a print server.
- ☐ **Remote Management for Linux** Remote management interface for Red Hat Enterprise Linux, including OpenLMI and SNMP.
- ☐ **Virtualization Hypervisor** Smallest possible virtualization host installation.
- ✓ **Compatibility Libraries** Compatibility libraries for applications built on previous versions of Red Hat Enterprise Linux.
- ✓ **Development Tools** A basic development environment.
- ✓ **Security Tools** Security tools for integrity and trust verification.
- ☐ **Smart Card Support** Support for using smart card authentication.
- ☐ **System Administration Tools** Utilities useful in system administration.

24. Select Installation Destination.



25. Select the Virtual Drive created from M.2 SATA SSDs. Select I will Configure Partitioning. Click DONE.



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

223.51 GiB

ATA CISCO VD

sda / 223.51 GiB free

Disks left unselected here will not be touched.

Specialized & Network Disks

Disks left unselected here will not be touched.

Other Storage Options

Partitioning

[ ] Automatically configure partitioning. [✓ I will configure partitioning.]

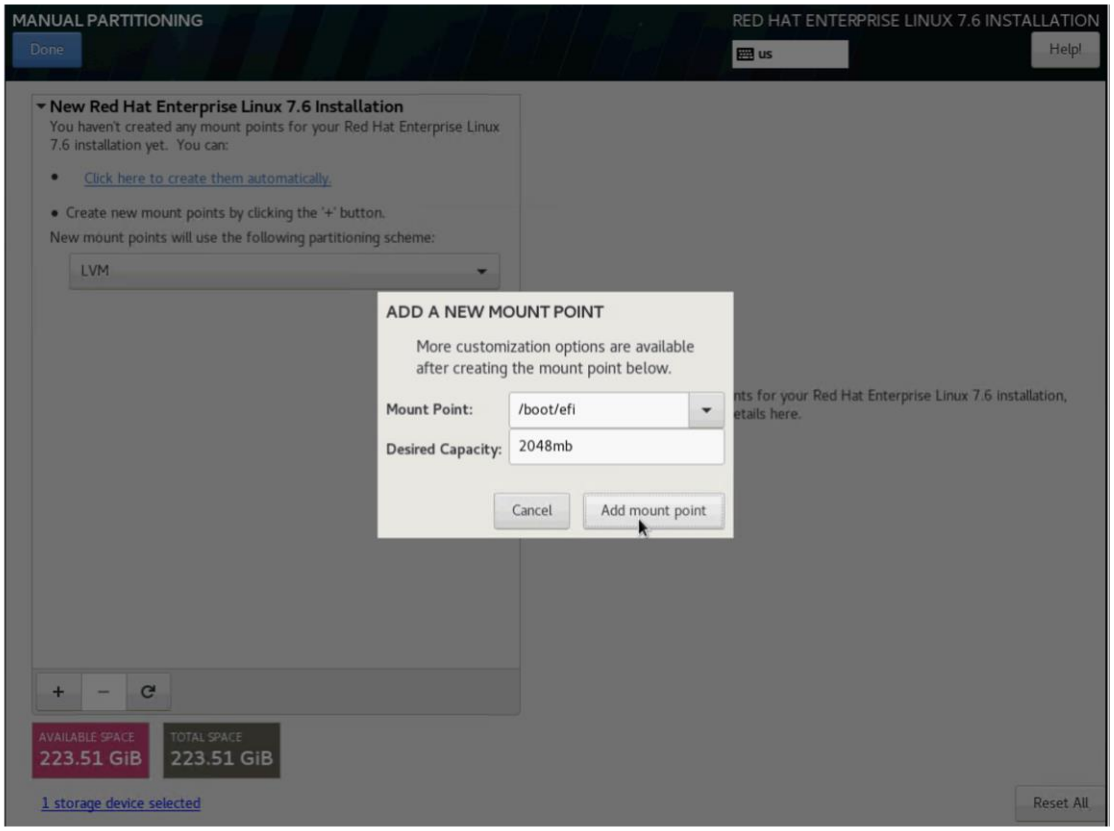
[ ] I would like to make additional space available.

Encryption

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

1 disk selected; 223.51 GiB capacity; 223.51 GiB free

26. Click the + button to add manual configuration to install Red Hat Enterprise Linux 7.6. Enter /boot/efi as mount point and 2048mb as Desired Capacity.


MANUAL PARTITIONING

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

New Red Hat Enterprise Linux 7.6 Installation
You haven't created any mount points for your Red Hat Enterprise Linux 7.6 installation yet. You can:

- Click here to create them automatically.
- Create new mount points by clicking the '+' button.

New mount points will use the following partitioning scheme:
LVM

ADD A NEW MOUNT POINT
More customization options are available after creating the mount point below.
Mount Point: /boot/efi
Desired Capacity: 2048mb

nts for your Red Hat Enterprise Linux 7.6 installation,
etails here.

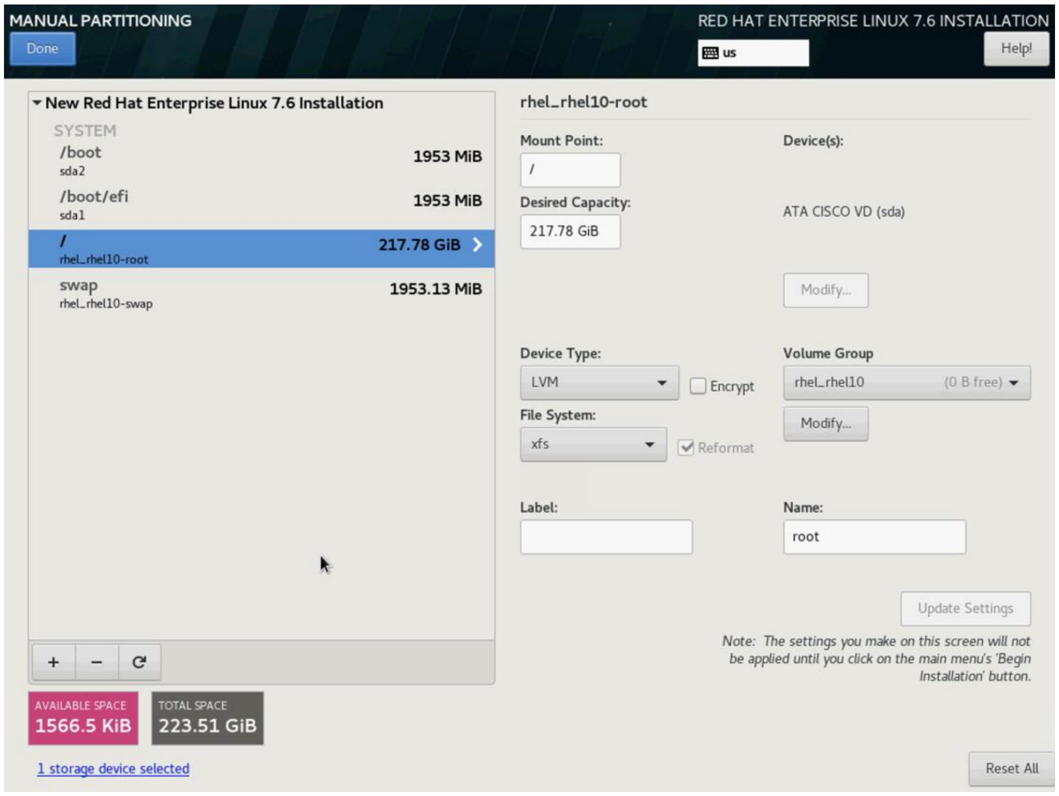
AVAILABLE SPACE 223.51 GiB
TOTAL SPACE 223.51 GiB
1 storage device selected

27. Click the + button for the following mountpoint and desired capacity as shown below.

Table 19 Mount Point and Desired Capacity for RHEL Installation

Mountpoint	Desired Capacity
/boot/efi	2048mb
/boot	2048mb
Swap	2048mb
/	

28. Verify the mount points and desired capacity. Click DONE.



New Red Hat Enterprise Linux 7.6 Installation
SYSTEM

/boot sda2	1953 MiB
/boot/efi sda1	1953 MiB
/ rhel_rhel10-root	217.78 GiB
swap rhel_rhel10-swap	1953.13 MiB

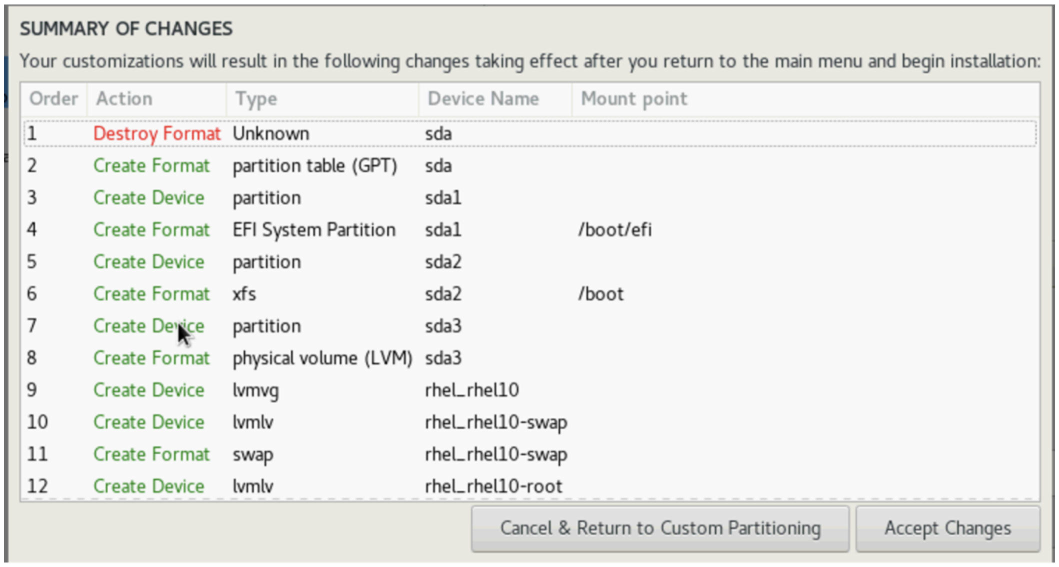
rhel_rhel10-root

Mount Point: /
Desired Capacity: 217.78 GiB
Device(s): ATA CISCO VD (sda)
Device Type: LVM
Encrypt
File System: xfs
[✓ Reformat]
Volume Group: rhel_rhel10 (0 B free)
Label:
Name: root

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

AVAILABLE SPACE 1566.5 KiB
TOTAL SPACE 223.51 GiB
1 storage device selected

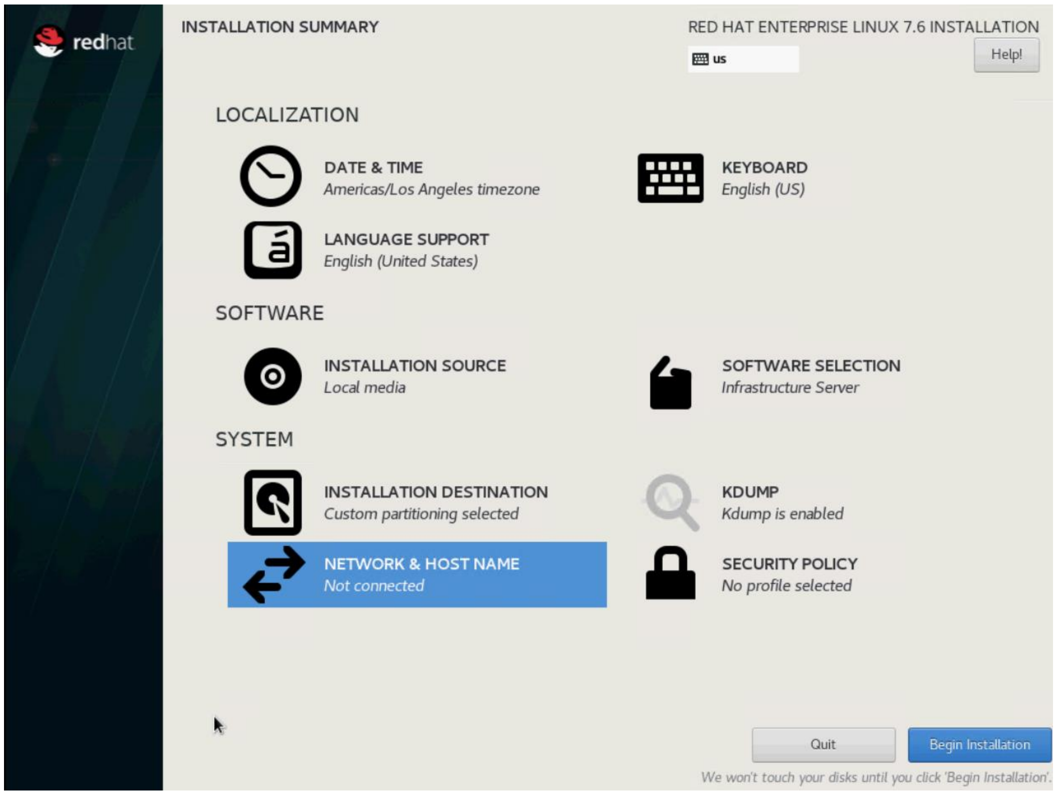
29. Click Accept Changes.


SUMMARY OF CHANGES

Your customizations will result in the following changes taking effect after you return to the main menu and begin installation:

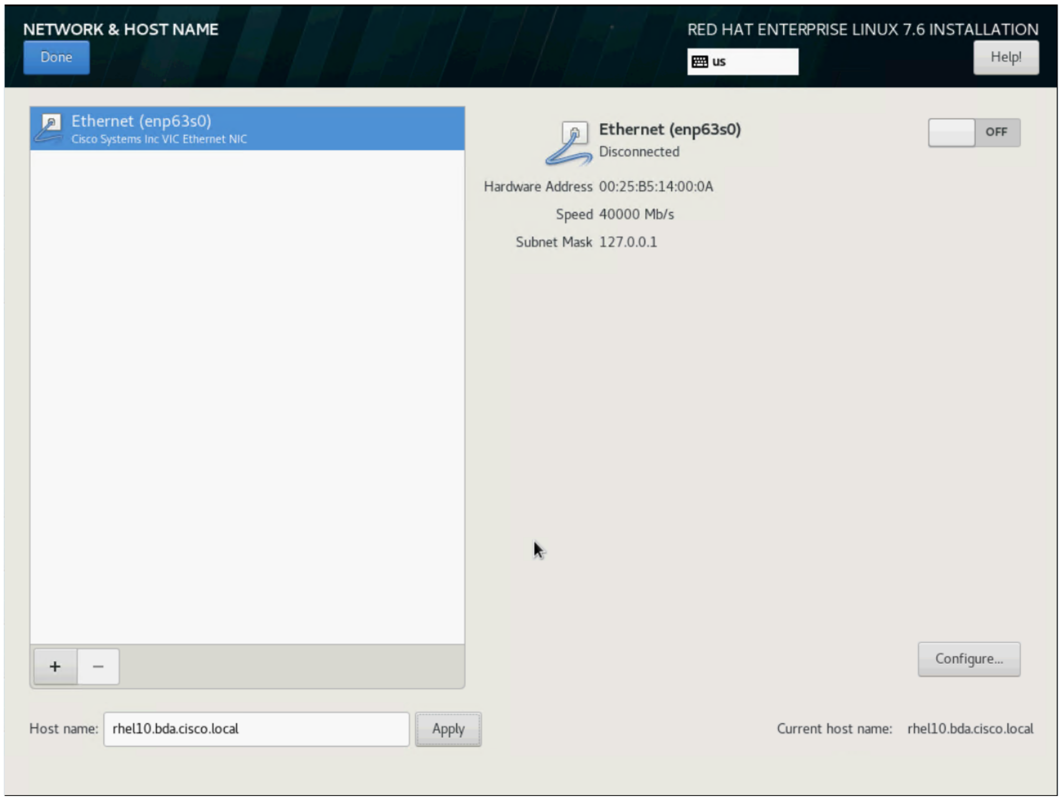
Order	Action	Type	Device Name	Mount point
1	Destroy Format	Unknown	sda	
2	Create Format	partition table (GPT)	sda	
3	Create Device	partition	sda1	
4	Create Format	EFI System Partition	sda1	/boot/efi
5	Create Device	partition	sda2	
6	Create Format	xfs	sda2	/boot
7	Create Device	partition	sda3	
8	Create Format	physical volume (LVM)	sda3	
9	Create Device	lvmvg	rhel_rhel10	
10	Create Device	lvmlv	rhel_rhel10-swap	
11	Create Format	swap	rhel_rhel10-swap	
12	Create Device	lvmlv	rhel_rhel10-root	

30. Click NETWORK & HOST NAME.



We won't touch your disks until you click 'Begin Installation'.

31. Enter Host name, click Apply. Select Configure.


NETWORK & HOST NAME
Done
RED HAT ENTERPRISE LINUX 7.6 INSTALLATION
us
Help!
Ethernet (enp63s0)
Cisco Systems Inc VIC Ethernet NIC
OFF
+
−
Configure...
Apply

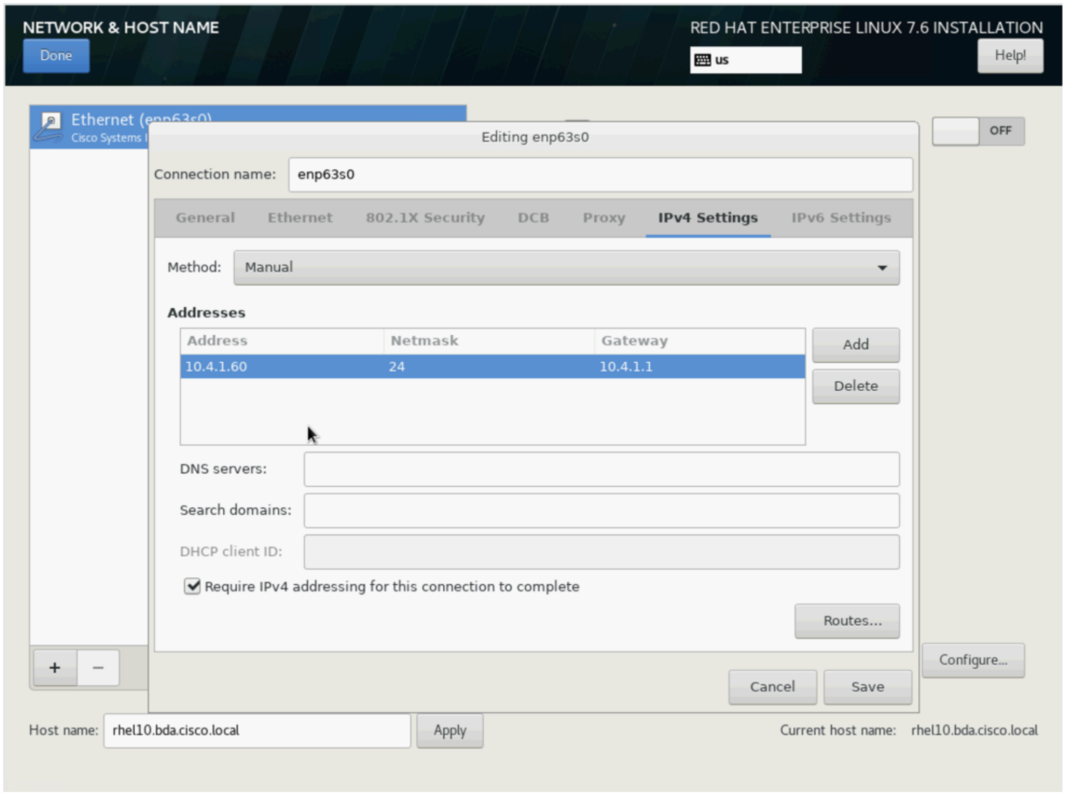
Ethernet (enp63s0)
Disconnected

Hardware Address 00:25:B5:14:00:0A
Speed 40000 Mb/s
Subnet Mask 127.0.0.1

Host name: rhel10.bda.cisco.local

Current host name: rhel10.bda.cisco.local

32. Select IPv4 Settings, Enter IP Address, Netmask and Gateway. Click Save.



Editing enp63s0

Connection name: enp63s0

Method: Manual

Addresses

Address	Netmask	Gateway
10.4.1.60	24	10.4.1.1

DNS servers:

Search domains:

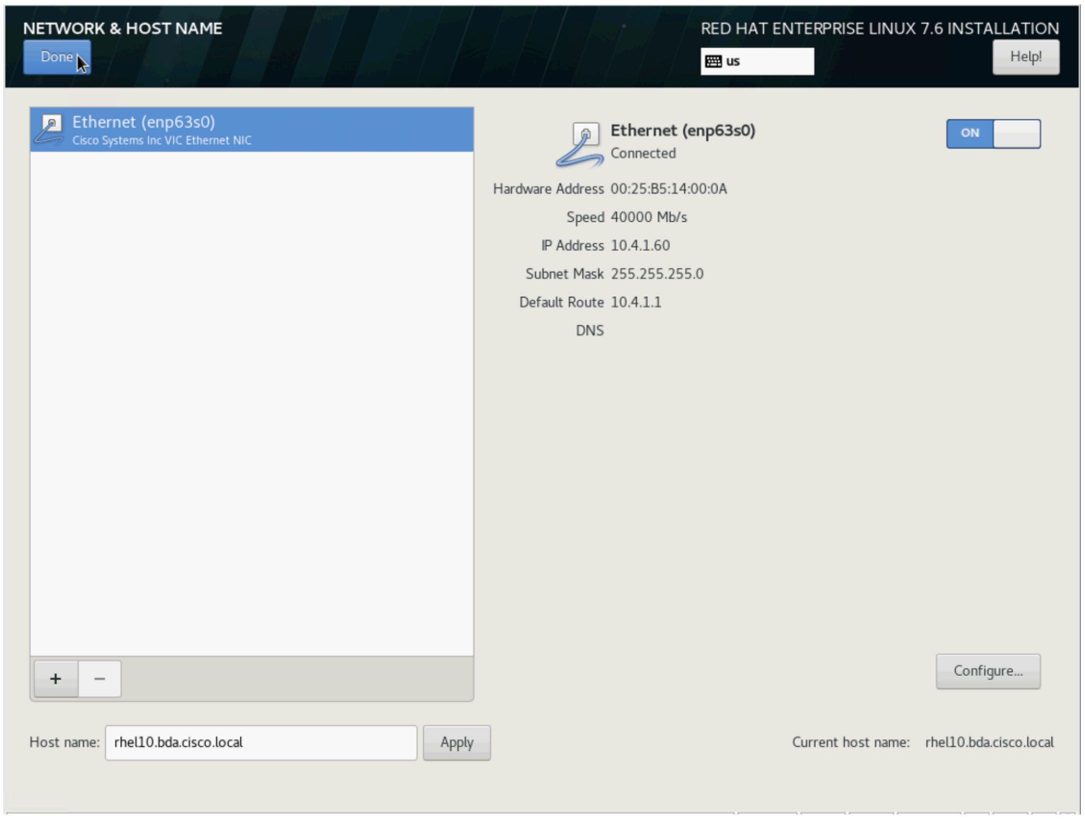
DHCP client ID:

☑ Require IPv4 addressing for this connection to complete

Host name: rhel10.bda.cisco.local

Current host name: rhel10.bda.cisco.local

33. Click OFF to turn ON the network adapter. Click Done.



Ethernet (enp63s0)
Cisco Systems Inc VIC Ethernet NIC

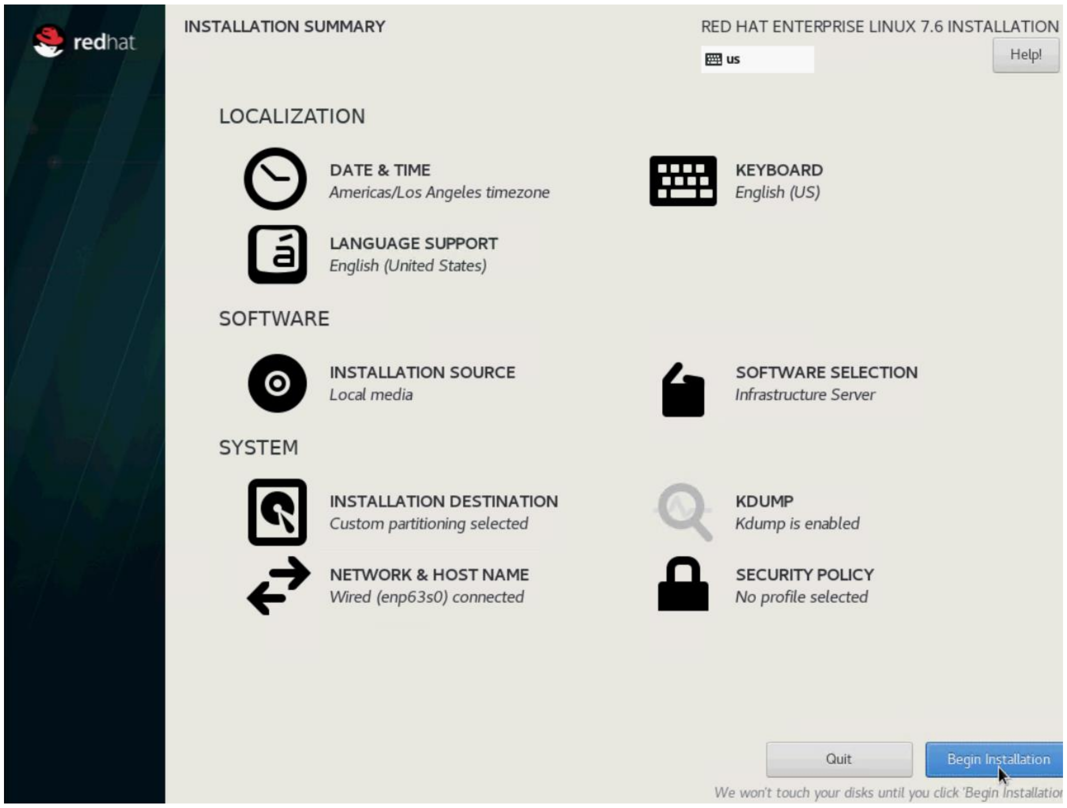
Ethernet (enp63s0)
Connected

Hardware Address	00:25:B5:14:00:0A
Speed	40000 Mb/s
IP Address	10.4.1.60
Subnet Mask	255.255.255.0
Default Route	10.4.1.1
DNS	

Host name: rhel10.bda.cisco.local

Current host name: rhel10.bda.cisco.local

34. Click Begin Installation.



INSTALLATION SUMMARY

RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

LOCALIZATION

DATE & TIME
Americas/Los Angeles timezone

KEYBOARD
English (US)

LANGUAGE SUPPORT
English (United States)

SOFTWARE

INSTALLATION SOURCE
Local media

SOFTWARE SELECTION
Infrastructure Server

SYSTEM

INSTALLATION DESTINATION
Custom partitioning selected

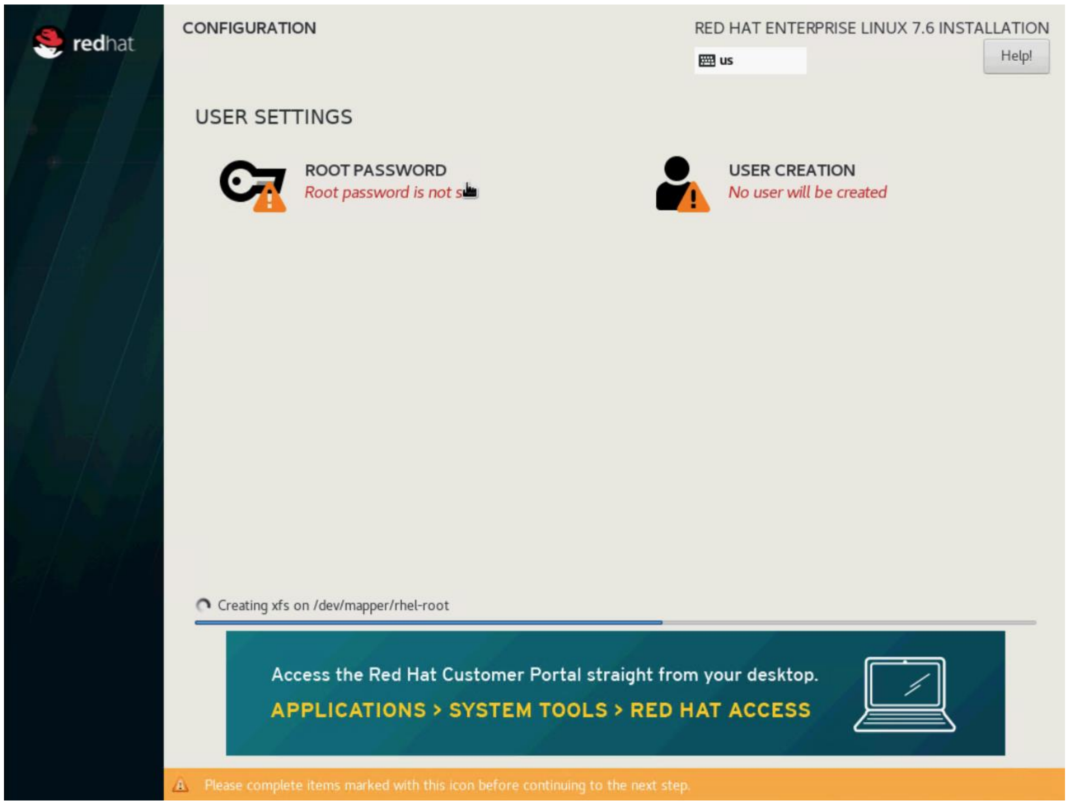
KDUMP
Kdump is enabled

NETWORK & HOST NAME
Wired (enp63s0) connected

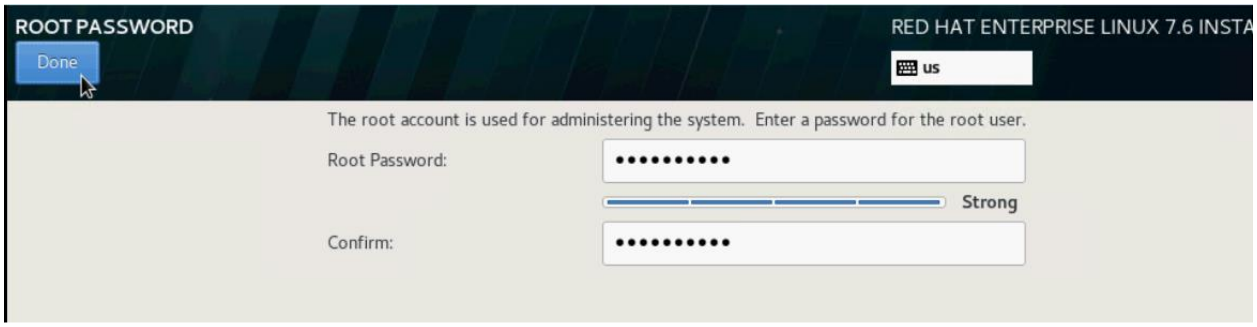
SECURITY POLICY
No profile selected

We won't touch your disks until you click 'Begin Installatio

35. Click ROOT PASSWORD.


redhat
CONFIGURATION
RED HAT ENTERPRISE LINUX 7.6 INSTALLATION
us
Help!
USER SETTINGS
ROOT PASSWORD
Root password is not s
USER CREATION
No user will be created
Creating xfs on /dev/mapper/rhel-root
Access the Red Hat Customer Portal straight from your desktop.
APPLICATIONS > SYSTEM TOOLS > RED HAT ACCESS
Please complete items marked with this icon before continuing to the next step.

36. Enter Root Password. Click Done.



The root account is used for administering the system. Enter a password for the root user.

Root Password: ••••••••••

Strong

Confirm: ••••••••••

37. Reboot when installation process completes.

Configure Data Drives on Name Node and Other Management Nodes

This section describes the steps needed to configure non-OS disk drives as RAID1 using the StorCli command. All drives are part of a single RAID1 volume. This volume can be used for staging any client data to be loaded to HDFS. This volume will not be used for HDFS data.

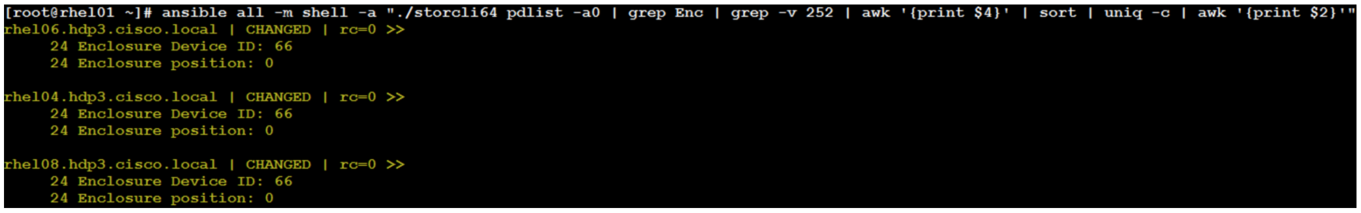
To configure data drives on the Name node and other nodes, if the drive state displays as JBOD, creating RAID in the subsequent steps will fail with the error "*The specified physical disk does not have the appropriate attributes to complete the requested command*."

To configure data drive on the Name node and other management nodes, follow these steps:

1. If the drive state shows up as JBOD, it can be converted into Unconfigured Good using Cisco UCSM or storcli64 command. The following steps should be performed if the state is JBOD.

2. Get the enclosure id as follows:

```
ansible all -m shell -a "./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk
'{print $4}' | sort | uniq -c | awk '{print $2}'"
```



```
[root@rhel01 ~]# ansible all -m shell -a "./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'"
rhel06.hdp3.cisco.local | CHANGED | rc=0 >>
     24 Enclosure Device ID: 66
     24 Enclosure position: 0

rhel04.hdp3.cisco.local | CHANGED | rc=0 >>
     24 Enclosure Device ID: 66
     24 Enclosure position: 0

rhel08.hdp3.cisco.local | CHANGED | rc=0 >>
     24 Enclosure Device ID: 66
     24 Enclosure position: 0
```

It is observed that some earlier versions of storcli64 complains about above mentioned command as if it is deprecated. In this case, please use "`./storcli64 /c0 show all| awk '{print $1}'| sed -n '/[0-9]:[0-9]/p'|awk '{print substr($1,1,2)}'|sort -u`" command to determine enclosure id.

With S3260, use -a0 and -a1 or c0 and c1 as there are two controller per node.

3. Convert to unconfigured good:

```
ansible datanodes -m command -a "./storcli64 /c0 /e66 /sall set good force"
```

4. Verify status by running the following command:

```
# ansible datanodes -m command -a "./storcli64 /c0 /e66 /sall show"
```

5. Run this script as root user on rhel01 to rhel3 to create the virtual drives for the management nodes:

```
#vi /root/raid1.sh
./storcli64 -cfgldadd
r1[$1:1,$1:2,$1:3,$1:4,$1:5,$1:6,$1:7,$1:8,$1:9,$1:10,$1:11,$1:12,$1:13,$1:14,$1:15,
$1:16,$1:17,$1:18,$1:19,$1:20,$1:21,$1:22,$1:23,$1:24] wb ra nocachedbadbbu
strpsz1024 -a0
```

The script (above) requires enclosure ID as a parameter.

6. Run the following command to get enclosure id:

```
#./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c
| awk '{print $2}'
#chmod 755 raid1.sh
```

7. Run MegaCli script:

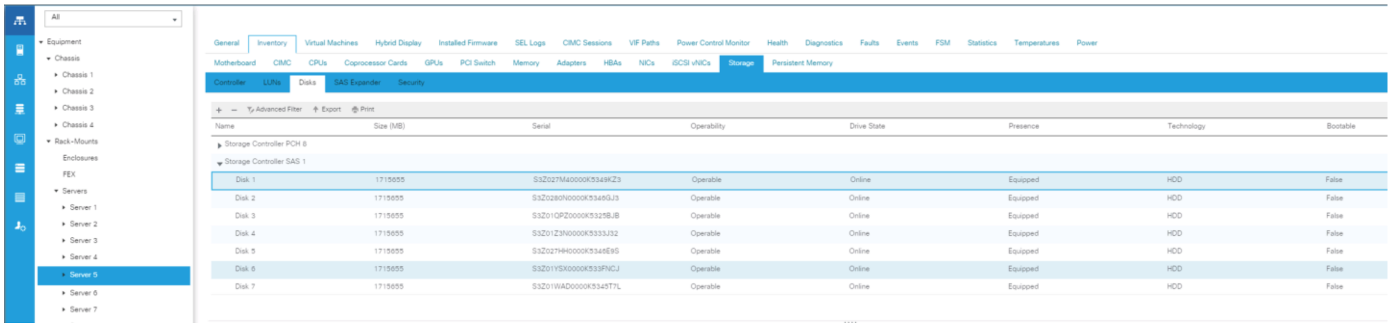
```
#./raid1.sh <EnclosureID> obtained by running the command above
WB: Write back
RA: Read Ahead
NoCachedBadBBU: Do not write cache when the BBU is bad.
Strpsz1024: Strip Size of 1024K
```

The command (above) will not override any existing configuration. To clear and reconfigure existing configurations refer to Embedded MegaRAID Software Users Guide available: [www.broadcom.com.](www.broadcom.com)

8. Run the following command. State should change to Online:

```
ansible namenodes -m command -a "./storcli64 /c0 /e66 /sall show"
```

9. State can also be verified in UCSM as show below in Equipment>Rack-Mounts>Servers>Server # under Inventory/Storage/Disk tab:



Name	Size (MB)	Serial	Operability	Drive State	Presence	Technology	Bootable
Storage Controller PCH 8							
Storage Controller SAS 1							
Disk 1	1715655	S3Z027M40000K5349KZ3	Operable	Online	Equipped	HDD	False
Disk 2	1715655	S3Z0280N0000K5346GJ3	Operable	Online	Equipped	HDD	False
Disk 3	1715655	S3Z01QPZ0000K5325BJB	Operable	Online	Equipped	HDD	False
Disk 4	1715655	S3Z01Z3N0000K5333J32	Operable	Online	Equipped	HDD	False
Disk 5	1715655	S3Z027HH0000K5346E9S	Operable	Online	Equipped	HDD	False
Disk 6	1715655	S3Z01YSX0000K533FNCJ	Operable	Online	Equipped	HDD	False
Disk 7	1715655	S3Z01WAD0000K5345T7L	Operable	Online	Equipped	HDD	False

Configure Data Drives on Data Nodes

To configure non-OS disk drives as individual RAID0 volumes using StorCli command, follow this step. These volumes will be used for HDFS Data.

1. Issue the following command from the admin node to create the virtual drives with individual RAID 0 configurations on all the data nodes:

```
[root@rhel01 ~]# ansible datanodes -m command -a "./storcli64 -cfgeachdskraid0 WB RA
direct NoCachedBadBBU strpsz1024 -a0"

rhel7.hdp3.cisco.local | SUCCESS | rc=0 >>
Adapter 0: Created VD 0
Configured physical device at Encl-66:Slot-7.
Adapter 0: Created VD 1
Configured physical device at Encl-66:Slot-6.
Adapter 0: Created VD 2
Configured physical device at Encl-66:Slot-8.
Adapter 0: Created VD 3
Configured physical device at Encl-66:Slot-5.
Adapter 0: Created VD 4
Configured physical device at Encl-66:Slot-3.
Adapter 0: Created VD 5
Configured physical device at Encl-66:Slot-4.
Adapter 0: Created VD 6
Configured physical device at Encl-66:Slot-1.
Adapter 0: Created VD 7
```

```
Configured physical device at Encl-66:Slot-2.
…… Omitted Ouput
24 physical devices are Configured on adapter 0.

Exit Code: 0x00
```

The command (above) will not override existing configurations. To clear and reconfigure existing configurations, refer to the Embedded MegaRAID Software Users Guide available at www.broadcom.com.

Cloudera Data Science Workbench (CDSW)

CDSW has prerequisites, one of which is CUDA. CUDA itself, also has prerequisites. The order of installation is:

1. CUDA prerequisites
2. CUDA
3. CDSW prerequisites
4. CDSW

Install Prerequisites for CUDA

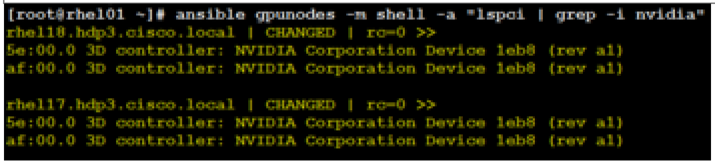
Details for the CUDA installation can be found here: http://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html.

These commands are run as root or sudo.

List GPUs and CPUs installed:

`lspci | grep -i nvidia`

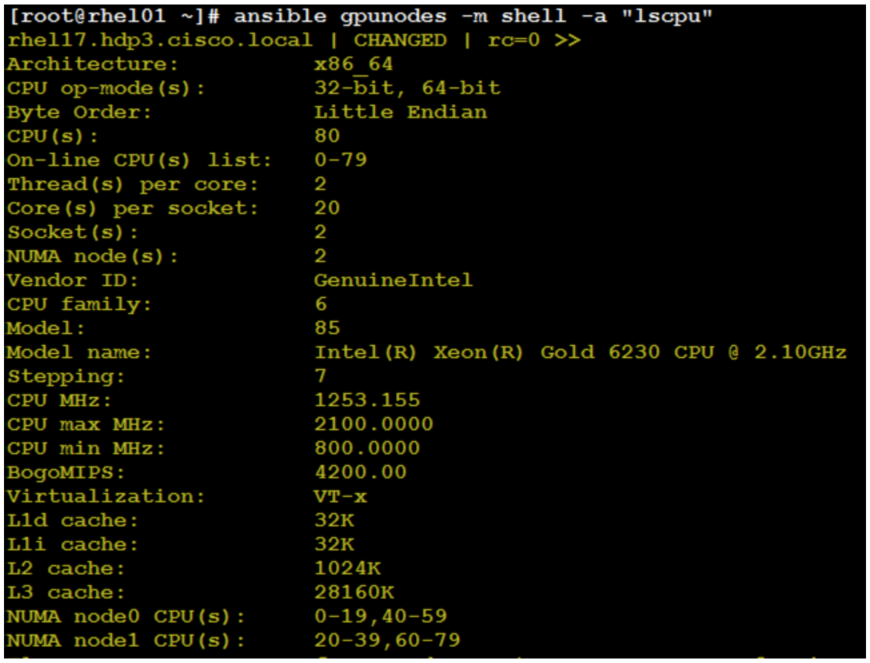
```
# ansible gpunodes -m shell -a "lspci | grep -i nvidia"
```



```
[root@rhel01 ~]# ansible gpunodes -m shell -a "lspci | grep -i nvidia"
rhel18.hdp3.cisco.local | CHANGED | rc=0 >>
5e:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)
af:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)

rhel17.hdp3.cisco.local | CHANGED | rc=0 >>
5e:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)
af:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)
```

`#lscpu`

ansible gpunodes -m shell -a " lscpu"

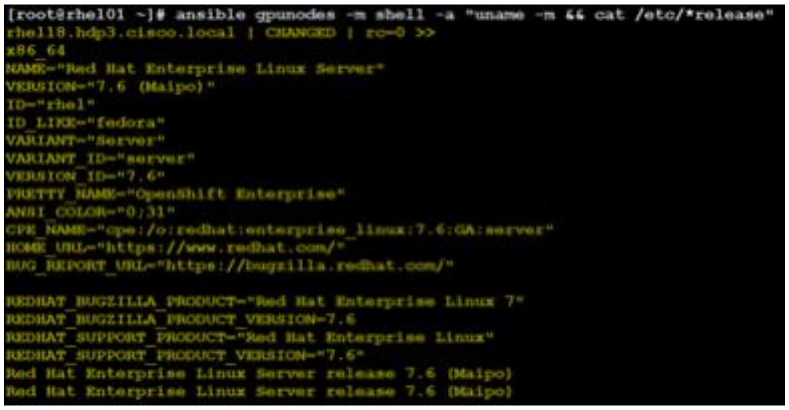


```
[root@rhel01 ~]# ansible gpunodes -m shell -a "lscpu"
rhel17.hdp3.cisco.local | CHANGED | rc=0 >>
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                80
On-line CPU(s) list:   0-79
Thread(s) per core:    2
Core(s) per socket:    20
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 85
Model name:            Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz
Stepping:              7
CPU MHz:               1253.155
CPU max MHz:           2100.0000
CPU min MHz:           800.0000
BogoMIPS:              4200.00
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              1024K
L3 cache:              28160K
NUMA node0 CPU(s):     0-19,40-59
NUMA node1 CPU(s):     20-39,60-79
```

Verify Supported Version of Linux

Make sure Red Hat Enterprise Linux OS running is supported.

```
# ansible gpunodes -m shell -a "uname -m && cat /etc/*release"
```



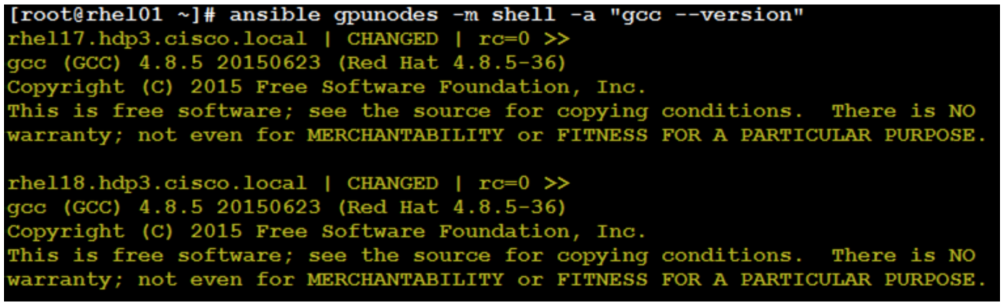
```
[root@rhel01 ~]# ansible gpunodes -m shell -a "uname -m && cat /etc/*release"
rhel18.hdp3.cisco.local | CHANGED | rc=0 >>
x86_64
NAME="Red Hat Enterprise Linux Server"
VERSION="7.6 (Maipo)"
ID="rhel"
ID_LIKE="fedora"
VARIANT="Server"
VARIANT_ID="server"
VERSION_ID="7.6"
PRETTY_NAME="OpenShift Enterprise"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:redhat:enterprise_linux:7.6:GA:server"
HOME_URL="https://www.redhat.com/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"

REDHAT_BUGZILLA_PRODUCT="Red Hat Enterprise Linux 7"
REDHAT_BUGZILLA_PRODUCT_VERSION=7.6
REDHAT_SUPPORT_PRODUCT="Red Hat Enterprise Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="7.6"
Red Hat Enterprise Linux Server release 7.6 (Maipo)
Red Hat Enterprise Linux Server release 7.6 (Maipo)
```

GCC Installation

Make sure gcc is installed in the system:

```
# gcc –version
```

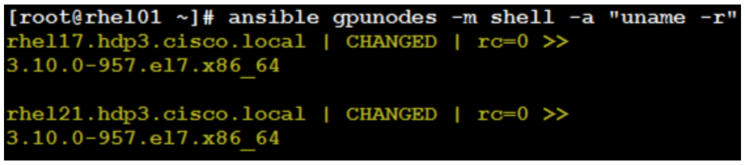
```
# ansible gpunodes -m shell -a "gcc --version"
```


```
[root@rhel01 ~]# ansible gpunodes -m shell -a "gcc --version"
rhel17.hdp3.cisco.local | CHANGED | rc=0 >>
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

rhel18.hdp3.cisco.local | CHANGED | rc=0 >>
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Install Kernel Headers and Installation Packages

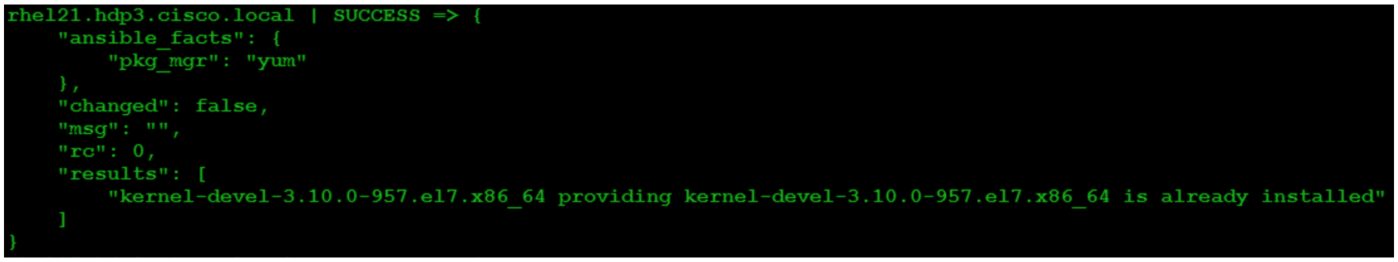
The CUDA Driver requires that the kernel headers and development packages for the running version of the kernel are installed at the time of the driver installation, as well as whenever the driver is rebuilt. For example, if your system is running kernel version 3.17.4-301, the 3.17.4-301 kernel headers and development packages must also be installed.

```
# ansible gpunodes -m shell -a "uname -r"
```

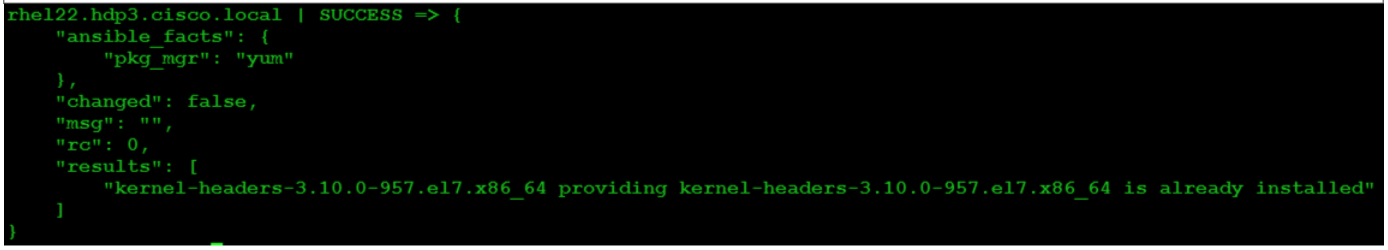

```
[root@rhel01 ~]# ansible gpunodes -m shell -a "uname -r"
rhel17.hdp3.cisco.local | CHANGED | rc=0 >>
3.10.0-957.el7.x86_64

rhel21.hdp3.cisco.local | CHANGED | rc=0 >>
3.10.0-957.el7.x86_64
```

Run the following command to Install:

```
# ansible gpunodes -m  yum -a "name=kernel-devel-$(uname -r) state=present"
```


```
rhel21.hdp3.cisco.local | SUCCESS => {
    "ansible_facts": {
        "pkg_mgr": "yum"
    },
    "changed": false,
    "msg": "",
    "rc": 0,
    "results": [
        "kernel-devel-3.10.0-957.el7.x86_64 providing kernel-devel-3.10.0-957.el7.x86_64 is already installed"
    ]
}
```

```
# ansible gpunodes -m  yum -a "name=kernel-headers-$(uname -r) state=present"
```


```
rhel22.hdp3.cisco.local | SUCCESS => {
    "ansible_facts": {
        "pkg_mgr": "yum"
    },
    "changed": false,
    "msg": "",
    "rc": 0,
    "results": [
        "kernel-headers-3.10.0-957.el7.x86_64 providing kernel-headers-3.10.0-957.el7.x86_64 is already installed"
    ]
}
```

Install dkms

The NVIDIA driver RPM packages depend on other external packages, such as DKMS. Those packages are only available on third-party repositories, such as EPEL.

http://rpmfind.net/linux/rpm2html/search.php?query=dkms for RHEL 7.x

RHEL 7.x http://rpmfind.net/linux/epel/7/x86_64/Packages/d/dkms-2.7.1-1.el7.noarch.rpm

```
#wget http://rpmfind.net/linux/epel/7/x86_64/Packages/d/dkms-2.7.1-1.el7.noarch.rpm
```

Copy dkms rpm to all the GPU servers

Install dkms with yum install

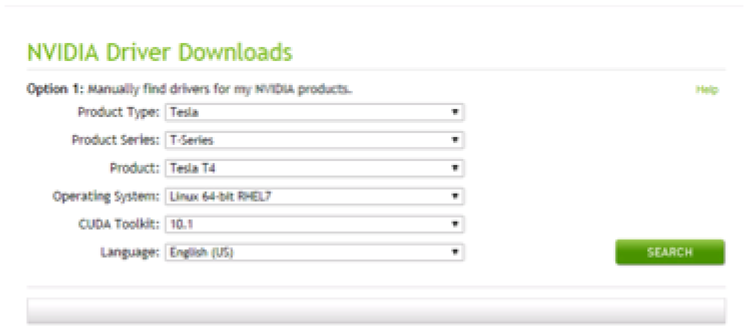
```
# ansible gpunodes -m command -a "yum install -y /root/dkms-2.7.1-1.el7.noarch.rpm"
```

Install NVIDIA GPU Drivers

To install the NVIDIA GPU drivers, follow these steps:

1. Download this NVIDIA GPU driver from http://www.nvidia.com/Download/index.asp?lang=en-us
2. For the NVIDIA driver download, select the product type, Series, Product, OS, and CUDA toolkit.

For this deployment, select 9.2 for CUDA Toolkit.

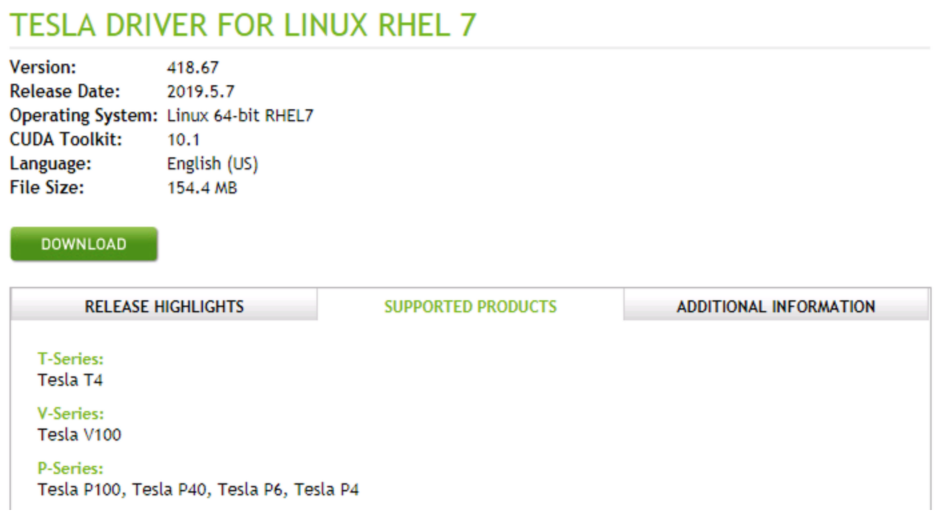


NVIDIA Driver Downloads

Option 1: Manually find drivers for my NVIDIA products.

Product Type:	Tesla
Product Series:	T-Series
Product:	Tesla T4
Operating System:	Linux 64-bit RHEL7
CUDA Toolkit:	10.1
Language:	English (US)

3. Click SEARCH. The selected driver is shown below.



TESLA DRIVER FOR LINUX RHEL 7

Version:	418.67
Release Date:	2019.5.7
Operating System:	Linux 64-bit RHEL7
CUDA Toolkit:	10.1
Language:	English (US)
File Size:	154.4 MB

T-Series:
Tesla T4

V-Series:
Tesla V100

P-Series:
Tesla P100, Tesla P40, Tesla P6, Tesla P4

4. Click Download. The download link can be captured by right-clicking AGREE & DOWNLOAD as shown below.

```
# wget http://us.download.nvidia.com/tesla/ 418.67/nvidia-diag-driver-local-repo-
rhel7-418.67-1.0-1.x86_64.rpm
```

5. Copy the .rpm file in all the GPU nodes as shown below.

```
# ansible gpunodes -m copy -a "src=/root/nvidia-diag-driver-local-repo-rhel7-418.67-
1.0-1.x86_64.rpm dest=/root/."
```

6. Install the driver by running the following command:

```
# ansible gpunodes -m command -a "rpm -ivh /root/nvidia-diag-driver-local-repo-
rhel7-418.67-1.0-1.x86_64.rpm "
```

Install CUDA

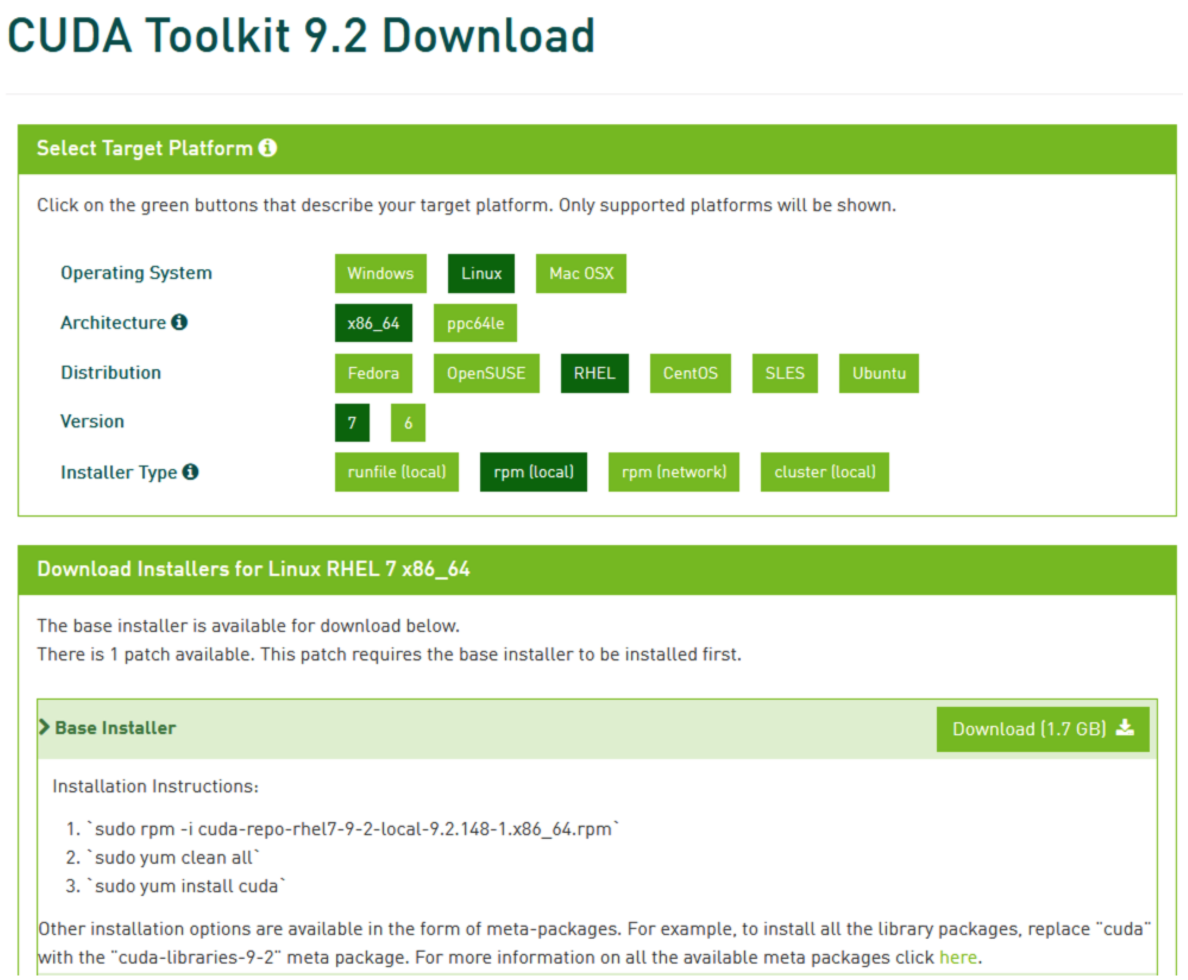
To install CUDA, follow this step:

1. Download CUDA 9.2

TensorFlow needs CUDA; make sure that version of CUDA is supported by TensorFlow before installing CUDA. Earlier versions of CUDA are here: https://developer.nvidia.com/cuda-toolkit-archive.

CUDA version 9.2 is available here:

https://developer.nvidia.com/cuda-92-download-archive?target_os=Linux&target_arch=x86_64&target_distro=RHEL&target_version=7&target_type=rpmlocal



CUDA Toolkit 9.2 Download

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows [✓ Linux] Mac OSX
Architecture	[✓ x86_64] ppc64le
Distribution	Fedora OpenSUSE [✓ RHEL] CentOS SLES Ubuntu
Version	[✓ 7] 6
Installer Type	runfile (local) [✓ rpm (local)] rpm (network) cluster (local)

Download Installers for Linux RHEL 7 x86_64

The base installer is available for download below.
There is 1 patch available. This patch requires the base installer to be installed first.

Base Installer — Download (1.7 GB)

Installation Instructions:

1. `sudo rpm -i cuda-repo-rhel7-9-2-local-9.2.148-1.x86_64.rpm`
2. `sudo yum clean all`
3. `sudo yum install cuda`

Other installation options are available in the form of meta-packages. For example, to install all the library packages, replace "cuda" with the "cuda-libraries-9-2" meta package. For more information on all the available meta packages click here.

This CVD is showing sample steps to install CUDA 9.2 to use TensorFlow 1.14. Please use the appropriate CUDA as needed.

```
#wget https://developer.nvidia.com/compute/cuda/9.2/Prod2/local_installers/cuda-repo-rhel7-9-2-local-9.2.148-1.x86_64.rpm
# ansible gpunodes -m copy -a "src=/root/cuda-repo-rhel7-9-2-local-9.2.148-1.x86_64.rpm dest=/root/."
# ansible gpunodes -m shell -a "rpm -i cuda-repo-rhel7-9-2-local-9.2.148-1.x86_64.rpm "
# ansible gpunodes -m shell -a "yum clean all"
# ansible gpunodes -m shell -a "yum -y install cuda"
```

Enable optional RPM repository if run into issue: subscription-manager repos --enable=rhel-7-server-optional-rpms

Download and Setup NVIDIA CUDA Deep Neural Network Library (cuDNN)

Download cuDNN 7.2.1

Download cuDNN from https://developer.nvidia.com/cudnn for the same CUDA version.

Copy it into all the GPU servers.

```
# wget https://developer.nvidia.com/compute/machine-
learning/cudnn/secure/v7.2.1/prod/9.2_20180806/cudnn-9.2-linux-x64-v7.2.1.38.tgz
# ansible gpunodes -m copy -a "src=/root/cudnn-9.2-linux-x64-v7.2.1.38 dest=/root/."
# tar -xzvf cudnn-9.2-linux-x64-v7.2.1.38
# ansible gpunodes -m shell -a "cp /root/cuda/include/cudnn.h /usr/local/cuda-
9.2/include"
# ansible gpunodes -m shell -a "cp /root/cuda/lib64/libcudnn* /usr/local/cuda-
9.2/lib64"
# ansible gpunodes -m shell -a "chmod a+r /usr/local/cuda-9.2/include/cudnn.h
/usr/local/cuda-9.2/lib64/libcudnn*"
```

Post Installation Steps

1. Add CUDA in PATH and LD_LIBRARY_PATH variable in all the GPU nodes.

2. The PATH and LD_LIBRARY_PATH variable needs to include /usr/local/<cuda 8 or 9 path>/bin:

```
# ansible gpunodes -m shell -a "export PATH=/usr/local/cuda-
9.2/bin${PATH:+:${PATH}}"
# ansible gpunodes -m shell -a "export LD_LIBRARY_PATH=/usr/local/cuda-
9.2/lib\${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}"
```

3. Reboot the GPU nodes.

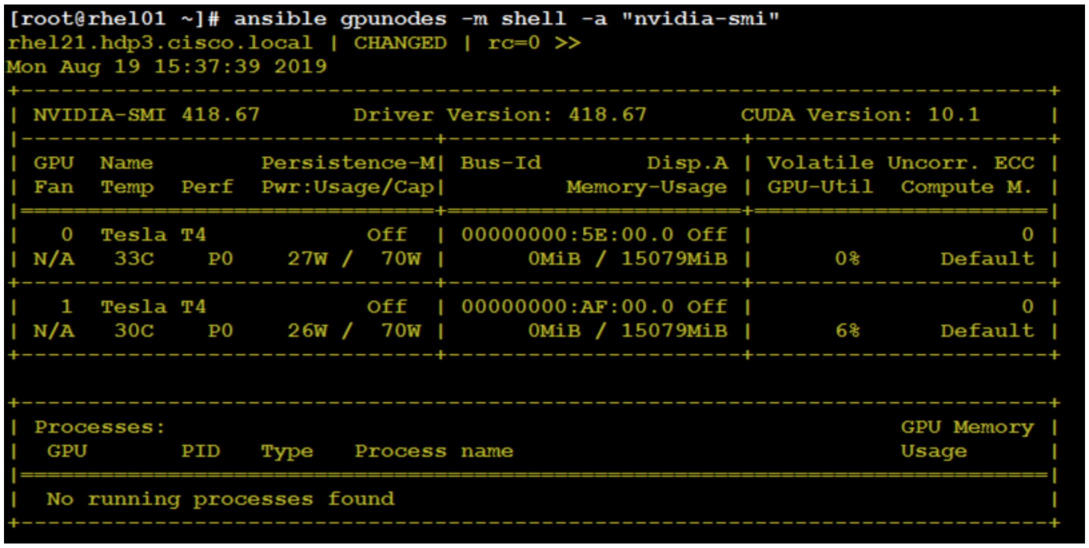
```
# ansible gpunodes -a "/sbin/reboot"
```

For more information, refer to: https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#post-installation-actions.

Verify Drivers

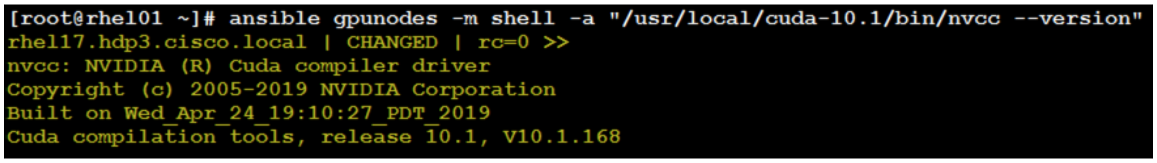
Restart the servers to verify the drivers have been installed:

```
# ansible gpunodes -m shell -a "nvidia-smi"
Or on specific Node
# ansible rhel20.hdp3.cisco.local -m shell -a "nvidia-smi"
```



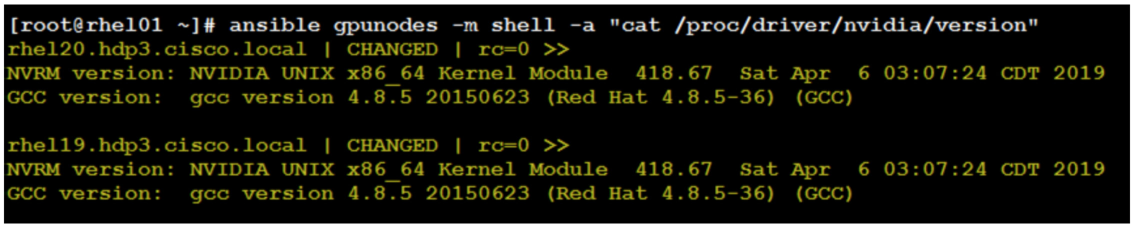
```
[root@rhel01 ~]# ansible gpunodes -m shell -a "nvidia-smi"
rhel21.hdp3.cisco.local | CHANGED | rc=0 >>
Mon Aug 19 15:37:39 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.67       Driver Version: 418.67       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| 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   33C    P0    27W /  70W |      0MiB / 15079MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            Off  | 00000000:AF:00.0 Off |                    0 |
| N/A   30C    P0    26W /  70W |      0MiB / 15079MiB |      6%      Default |
+-------------------------------+----------------------+----------------------+

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

ansible gpunodes -m shell -a " /usr/local/cuda-9.2/bin/nvcc --version"



```
[root@rhel01 ~]# ansible gpunodes -m shell -a "/usr/local/cuda-10.1/bin/nvcc --version"
rhel17.hdp3.cisco.local | CHANGED | rc=0 >>
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Wed_Apr_24_19:10:27_PDT_2019
Cuda compilation tools, release 10.1, V10.1.168
```

ansible gpunodes -m shell -a " cat /proc/driver/nvidia/version"



```
[root@rhel01 ~]# ansible gpunodes -m shell -a "cat /proc/driver/nvidia/version"
rhel20.hdp3.cisco.local | CHANGED | rc=0 >>
NVRM version: NVIDIA UNIX x86_64 Kernel Module  418.67  Sat Apr  6 03:07:24 CDT 2019
GCC version:  gcc version 4.8.5 20150623 (Red Hat 4.8.5-36) (GCC)

rhel19.hdp3.cisco.local | CHANGED | rc=0 >>
NVRM version: NVIDIA UNIX x86_64 Kernel Module  418.67  Sat Apr  6 03:07:24 CDT 2019
GCC version:  gcc version 4.8.5 20150623 (Red Hat 4.8.5-36) (GCC)
```

ansible gpunodes -m shell -a " export NVIDIA_DRIVER_VERSION=418.67"

Installation Prerequisites for CDSW

Follow the steps in this section on all CDSW nodes:

For more information, go to: https://www.cloudera.com/documentation/data-science-work-bench/latest/topics/cdsw_requirements_supported_versions.html#cdsw_requirements_supported_versions and https://www.cloudera.com/documentation/data-science-workbench/1-6-x/topics/cdsw_install.html

Set Up a Wildcard DNS Subdomain

Cloudera Data Science Workbench uses subdomains to provide isolation for user-generated HTML and JavaScript, and routing requests between services. To set up subdomains for Cloudera Data Science Workbench, configure your DNS server with an A record for a wildcard DNS name such as `*.cdsw.<your_domain>.com` for the master host, and a second A record for the root entry of `cdsw.<your_domain>.com`.

You can also use a wildcard CNAME record if it is supported by your DNS provider.

This Wildcard DNS subdomain need to be used by the jump host/edge node or the bastion server as well. For more information, go to: https://www.cloudera.com/documentation/data-science-workbench/1-6-x/topics/cdsw_install.html#wildcard_dns

For the Solution validation, we had installed DNS server on windows 2012 R2 and setup DNS wild card configuration on the same.

To set up a wildcard DNS subdomain, follow these steps:

1. With dnsmasq, to add a wildcard DNS subdomain, complete the following steps only for the master host.
2. Update /etc/dnsmasq.conf to enable Wildcard entry:

```
address=/cdsw/10.13.1.50
```

```
#service dnsmasq restart
```

3. Test the working of wildcard DNS with dig or nslookup:

```
#nslookup *.cdsw.<your domain>.com
```

```
#dig cdsw.<your_domain>.com
```

```
#dig *.cdsw.<your domain>.com
```

For more information, go to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_install.html#pre_install.

Supported JDK Version

The entire CDH cluster, including Cloudera Data Science Workbench gateway nodes, must use Oracle JDK. OpenJDK is not supported by CDH, Cloudera Manager, or Cloudera Data Science Workbench.

Spark 2.2 which is needed by CDSW to run Spark jobs (on Hadoop nodes) requires JDK 1.8. On CSD-based deployments, Cloudera Manager automatically detects the path and version of Java installed on Cloudera Data Science Workbench gateway hosts.

To upgrade your entire CDH cluster to JDK 1.8, see Upgrading to Oracle JDK 1.8.

For more details, go to: https://www.cloudera.com/documentation/enterprise/release-notes/topics/rn_consolidated_pcm.html#pcm_jdk

IP Tables and Security on CDSW Nodes

Disable all pre-existing `iptables` rules. While Kubernetes makes extensive use of iptables, it is difficult to predict how pre-existing iptables rules will interact with the rules inserted by Kubernetes. Therefore, Cloudera recommends you use the following commands to disable all pre-existing rules before you proceed with the installation.

```
yum -y install iptables-services
yum -y install initscripts
systemctl stop firewalld
systemctl mask firewalld
systemctl disable firewalld
systemctl enable iptables
systemctl start iptables
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -t nat -F
iptables -t mangle -F
iptables -F
iptables -X
service iptables save
```

For more information about Cloudera Data Science Workbench 1.6.x Requirements and Supported Platforms, go to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_requirements_supported_versions.html

Please refer to section Disable SELinux if SELinux is enabled.

Configure Block Devices

Docker Block Device

The Cloudera Data Science Workbench installer will format and mount Docker on each gateway host. Make sure there is no important data stored on these devices. *Do not mount these block devices prior to installation.*

Application Block Device or Mount Point

The master host on Cloudera Data Science Workbench requires at least 500 GB for database and project storage. This recommended capacity is contingent on the expected number of users and projects on the cluster. While large data files should be stored on HDFS, it is not uncommon to find gigabytes of data or libraries in individual projects. Running out of storage will cause the application to fail. Make sure you continue to carefully monitor disk space usage and I/O using Cloudera Manager.

To enable data resilience, enable this drive as RAID1 of SSDs (using commands as shown in configuring namenode).

Cloudera Data Science Workbench will store all application data at `/var/lib/cdsw`. In a CSD-based deployment, this location is not configurable. Cloudera Data Science Workbench will assume the system administrator has formatted and mounted one or more block devices to `/var/lib/cdsw`.

Regardless of the application data storage configuration you choose, /var/lib/cdsw must be stored on a separate block device (the RAID1 of SSDs created for this).

Download and Install CDSW with Cloudera Manager

To download and install CDSW, follow these steps:

1. Download the Cloudera Data Science Workbench Custom Service Descriptor (CSD) CLOUDERA_DATA_SCIENCE_WORKBENCH-1.5.0.jar

For more information, refer to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_install_parcel.html#csd.

Cloudera CDSW 1.5 supports CDH 6.1.x or higher and RHEL 7.2, 7.3, 7.4, 7.5. Nodes part of the CDSW configuration in the cluster were configured with RHEL 7.5

For visit, https://www.cloudera.com/documentation/data-science-workbench/1-5-x/topics/cdsw_requirements_supported_versions.html for complete list of requirements and supported platforms.

```
Log on to the Cloudera Manager Server host, and place the CSD
CLOUDERA_DATA_SCIENCE_WORKBENCH-1.5.0.jar file under /opt/cloudera/csd, which is the
default location for CSD files.
Once cloudera Manager is copied under /opt/cloudera/csd set the file ownership
to cloudera-scm:cloudera-scm with permission 644
# scp CLOUDERA_DATA_SCIENCE_WORKBENCH-CDH6-1.5.0.jar rhel01:/opt/cloudera/csd/
# chown cloudera-scm:cloudera-scm CLOUDERA_DATA_SCIENCE_WORKBENCH-CDH6-1.5.0.jar
# chmod 644 CLOUDERA_DATA_SCIENCE_WORKBENCH-CDH6-1.5.0.jar
```

2. Restart cloudera-management service

```
service cloudera-scm-server restart
```

3. Log into the Cloudera Manager Admin Console and restart the Cloudera Management Service.

   a. Select Clusters > Cloudera Management Service.

   b. Select Actions > Restart.

   c. From the machine with internet access, download Cloudera CDSW parcel from https://archive.cloudera.com/cdsw1/1.5.0/parcels/

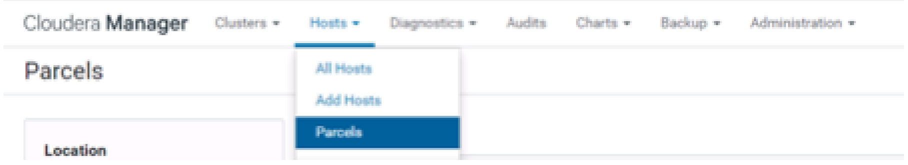
```
#  wget https://archive.cloudera.com/cdsw1/1.5.0/parcels/CDSW-1.5.0.p1.849870-
el7.parcel
#  wget https://archive.cloudera.com/cdsw1/1.5.0/parcels/CDSW-1.5.0.p1.849870-
el7.parcel.sha
#  wget https://archive.cloudera.com/cdsw1/1.5.0/parcels/manifest.json
```

   d. Copy downloaded CDSW parcels into the server hosting Cloudera parcels copy all these files to CDSW folder and move it to /var/www/html/CDSW

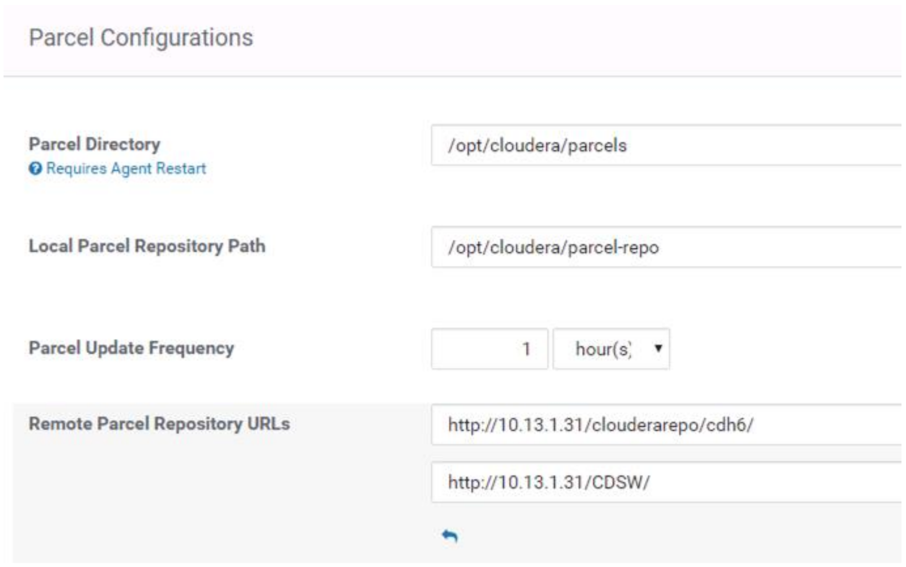
```
# mkdir -p /var/www/html/CDSW
# scp CDSW-1.5.0.* rhel01:/var/www/html/CDSW/
# scp manifest.json rhel01:/var/www/html/CDSW/
```

e. Log into the Cloudera Manager Admin Console.

```
Click Hosts > Parcels in the main navigation bar.
```



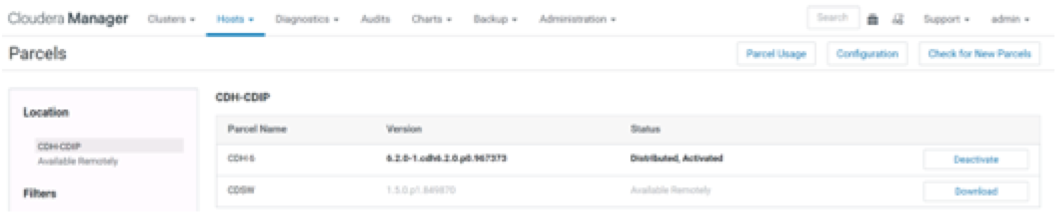
4. Click Configuration to configure the Cloudera Data Science Workbench parcel. Add URL to access CDSW local repository.



Parcel Configurations

Parcel Directory (Requires Agent Restart)	/opt/cloudera/parcels
Local Parcel Repository Path	/opt/cloudera/parcel-repo
Parcel Update Frequency	1 hour(s)
Remote Parcel Repository URLs	http://10.13.1.31/clouderarepo/cdh6/
	http://10.13.1.31/CDSW/

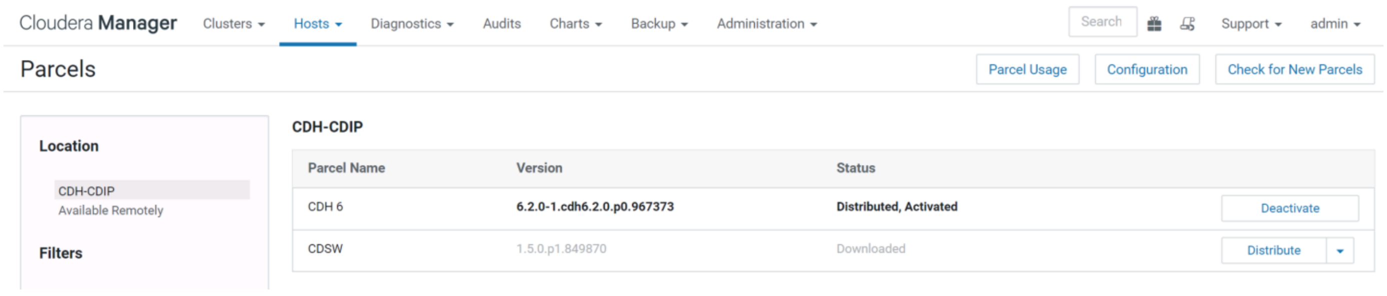
5. Click Download to download parcel to all nodes in the cluster.



CDH-CDIP

Parcel Name	Version	Status	
CDH 6	6.2.0-1.cdh6.2.0.p0.967373	Distributed, Activated	Deactivate
CDSW	1.5.0.p1.849870	Available Remotely	Download

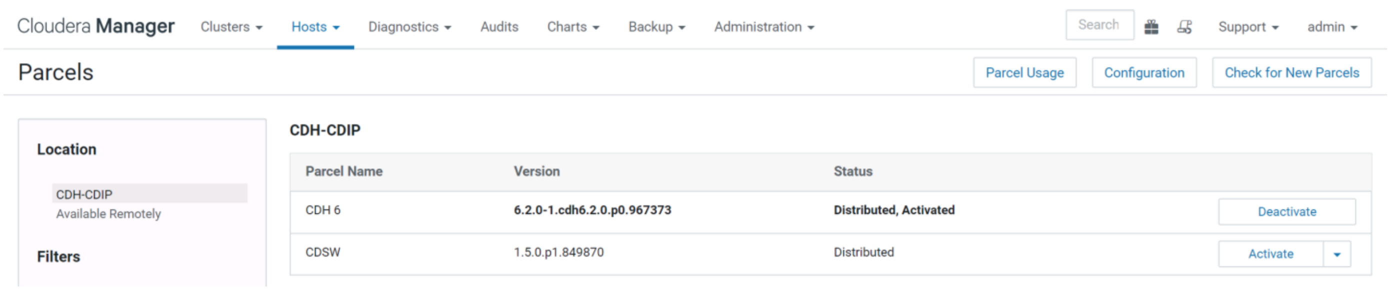
6. Distribute the parcel to the hosts in your cluster, and then activate the parcel.



CDH-CDIP

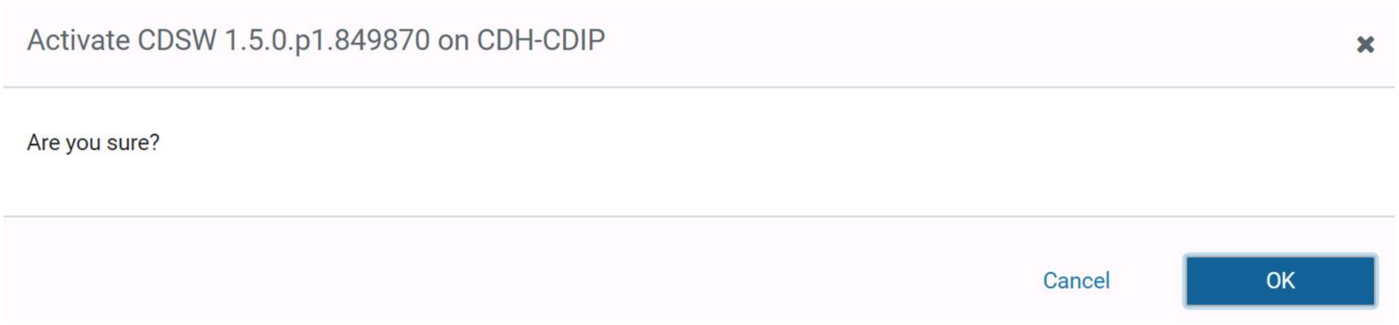
Parcel Name	Version	Status	
CDH 6	6.2.0-1.cdh6.2.0.p0.967373	Distributed, Activated	Deactivate
CDSW	1.5.0.p1.849870	Downloaded	Distribute

7. Once distributed, click Activate.



Parcel Name	Version	Status	
CDH 6	6.2.0-1.cdh6.2.0.p0.967373	Distributed, Activated	Deactivate
CDSW	1.5.0.p1.849870	Distributed	Activate

8. Click OK.



Activate CDSW 1.5.0.p1.849870 on CDH-CDIP

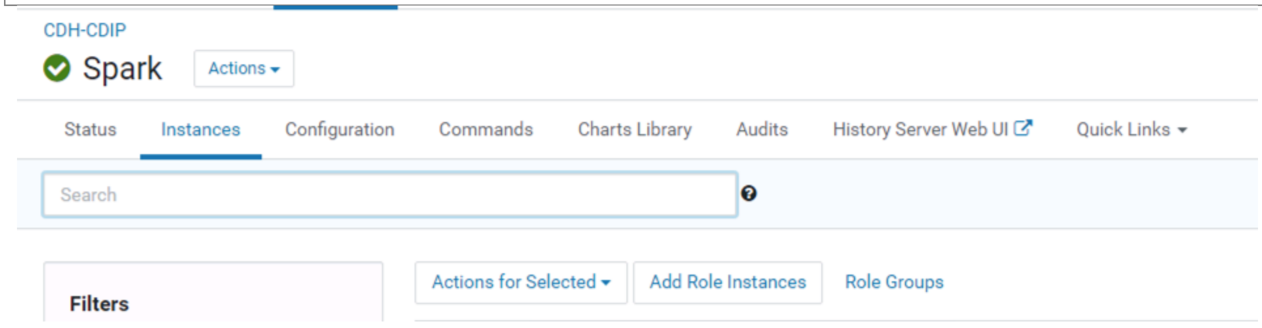
Are you sure?

Add Apache Spark 2 Service

From CDH version 6 and later, Apache Spark 2 is packaged within CDH and can no longer be installed separately.

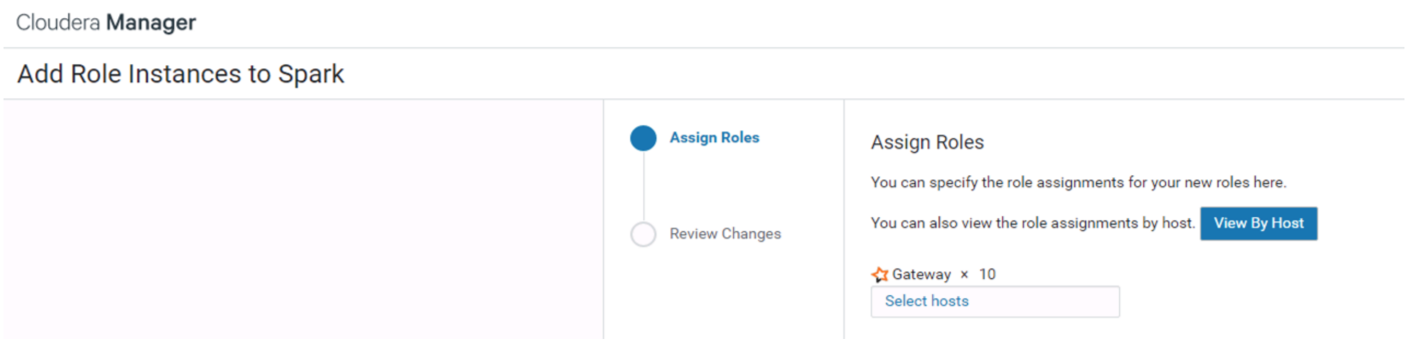
To find out which version of Spark 2 ships with your version of CDH 6, see the [CDH 6 Packaging Information]guide.

```
In Cloudera Manager WebUI console for the cluster, Select Spark. Click Add Role
Instances.
```



To add Apache Spark 2 Service, follow these steps:

1. Select host(s) to add Spark gateway, click Continue.



Cloudera **Manager**

Add Role Instances to Spark

Assign Roles

Review Changes

Assign Roles

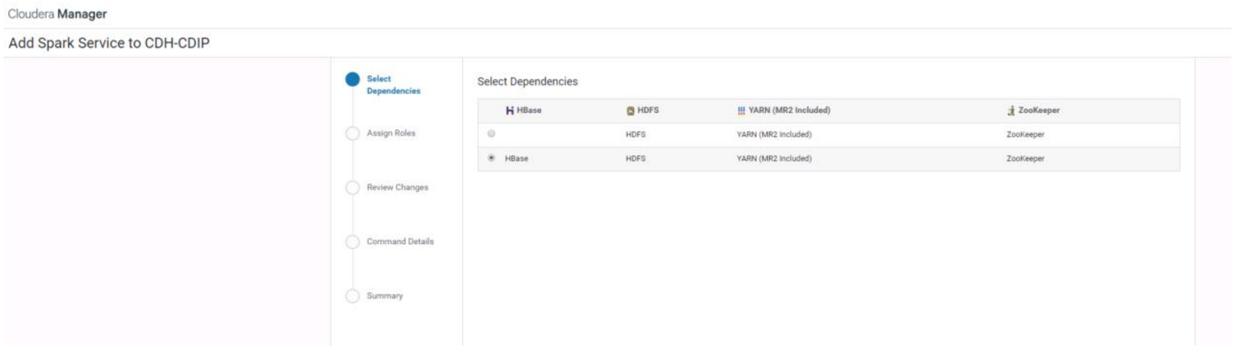
You can specify the role assignments for your new roles here.

You can also view the role assignments by host. View By Host

Gateway × 10

Select hosts

2. Select Dependencies.



Cloudera **Manager**

Add Spark Service to CDH-CDIP

Select Dependencies

Assign Roles

Review Changes

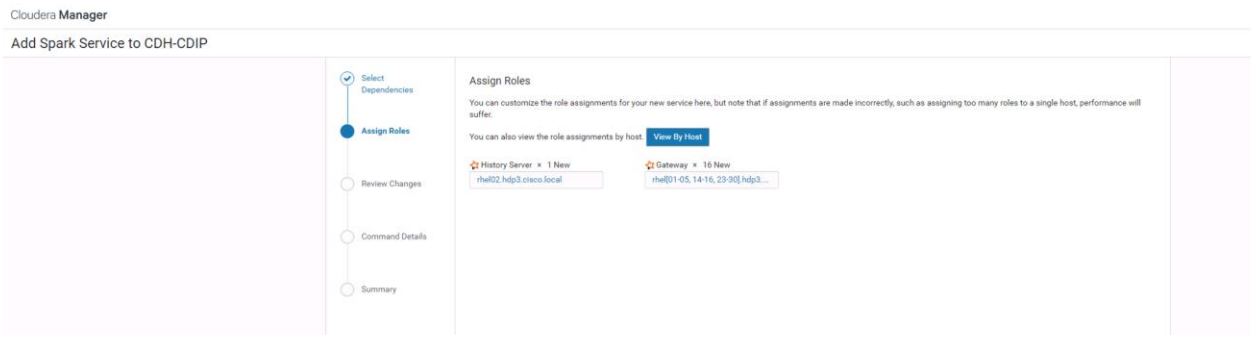
Command Details

Summary

Select Dependencies

	HBase	HDFS	YARN (MR2 Included)	ZooKeeper
○		HDFS	YARN (MR2 Included)	ZooKeeper
●	HBase	HDFS	YARN (MR2 Included)	ZooKeeper

3. If Spark was not added part of the cluster creation process. Click `Add Service` and Assign Spark History Server and gateway roles as shown in the screenshot below and complete adding Spark service to cluster.



Cloudera **Manager**

Add Spark Service to CDH-CDIP

Select Dependencies

Assign Roles

Review Changes

Command Details

Summary

Assign Roles

You can customize the role assignments for your new service here, but note that if assignments are made incorrectly, such as assigning too many roles to a single host, performance will suffer.

You can also view the role assignments by host. View By Host

History Server × 1 New

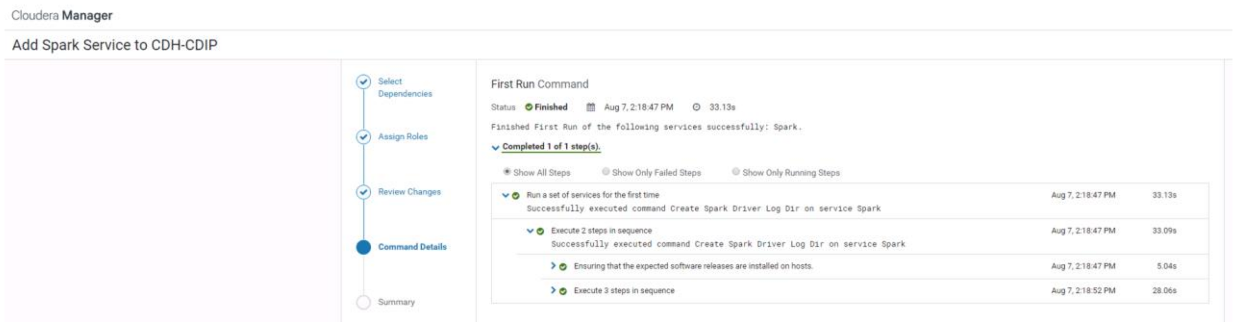
rhel02.hdp3.cisco.local

Gateway × 16 New

rhel[01-05, 14-16, 23-30].hdp3....

We selected one history server and six NVidia GPU installed nodes which will be used later as part of the CDSW service as Spark Gateway.

4. Cloudera Manager runs the command to configure history server and Spark gateway role and start them on nodes assigned.



Cloudera **Manager**

Add Spark Service to CDH-CDIP

First Run Command

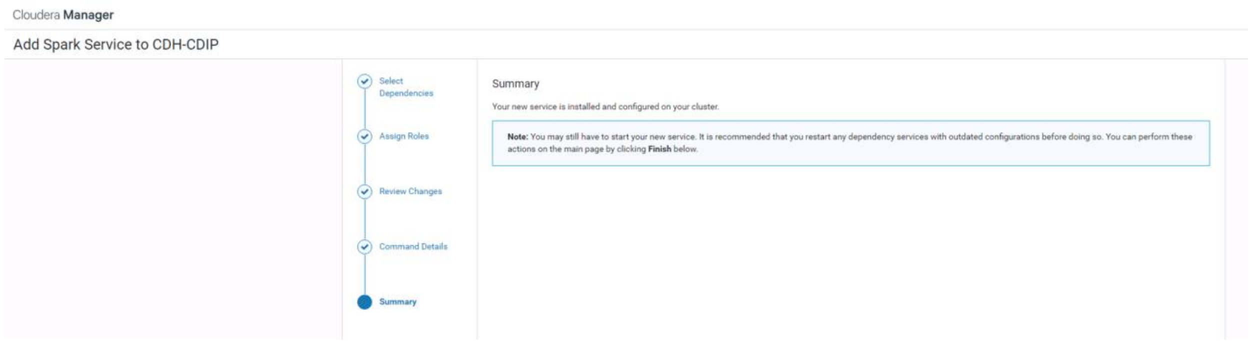
Status **Finished** Aug 7, 2:18:47 PM 33.13s

```
Finished First Run of the following services successfully: Spark.
```

Completed 1 of 1 step(s).

```
Run a set of services for the first time                                      Aug 7, 2:18:47 PM   33.13s
Successfully executed command Create Spark Driver Log Dir on service Spark
  Execute 2 steps in sequence                                                 Aug 7, 2:18:47 PM   33.09s
  Successfully executed command Create Spark Driver Log Dir on service Spark
    Ensuring that the expected software releases are installed on hosts.      Aug 7, 2:18:47 PM   5.04s
    Execute 3 steps in sequence                                               Aug 7, 2:18:52 PM   28.06s
```

5. Review the Summary. Click Finish.



Cloudera **Manager**

Add Spark Service to CDH-CDIP

Summary

Your new service is installed and configured on your cluster.

Note: You may still have to start your new service. It is recommended that you restart any dependency services with outdated configurations before doing so. You can perform these actions on the main page by clicking **Finish** below.

6. Select Deploy Client Configuration.

Add the Cloudera Data Science Workbench Service

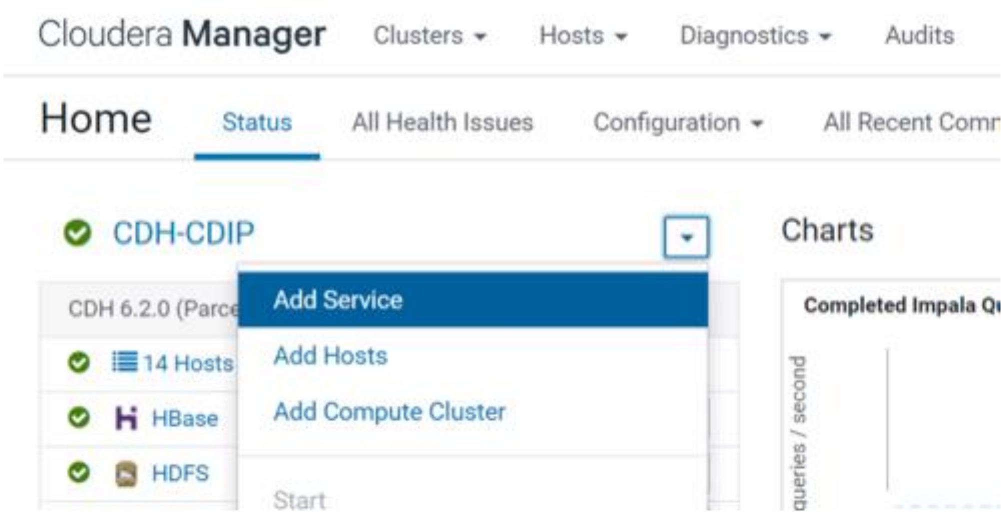
To install CDSW, follow these steps:

For more information, go to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_install_parcel.html.

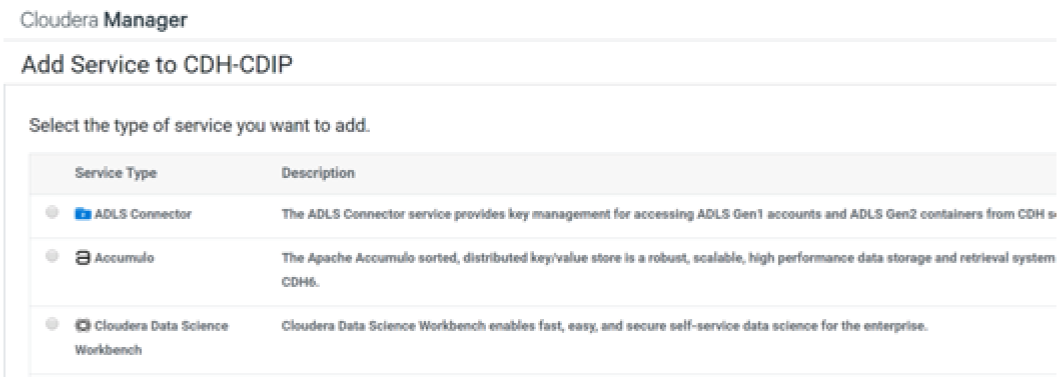
---

1. Log into the Cloudera Manager Admin Console.

2. On the Home > Status tab, right-click the cluster name and select Add a Service to launch the wizard. A list of services will be displayed.



3. Select the Cloudera Data Science Workbench service and click Continue.



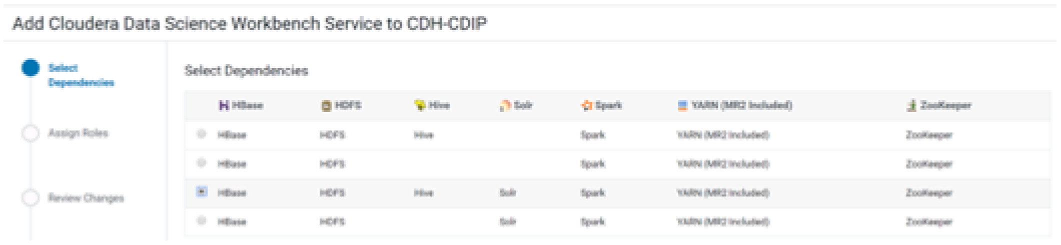
Cloudera Manager

Add Service to CDH-CDIP

Select the type of service you want to add.

	Service Type	Description
○	ADLS Connector	The ADLS Connector service provides key management for accessing ADLS Gen1 accounts and ADLS Gen2 containers from CDH s
○	Accumulo	The Apache Accumulo sorted, distributed key/value store is a robust, scalable, high performance data storage and retrieval system CDH6.
○	Cloudera Data Science Workbench	Cloudera Data Science Workbench enables fast, easy, and secure self-service data science for the enterprise.

4. Select Dependencies. Click Continue.

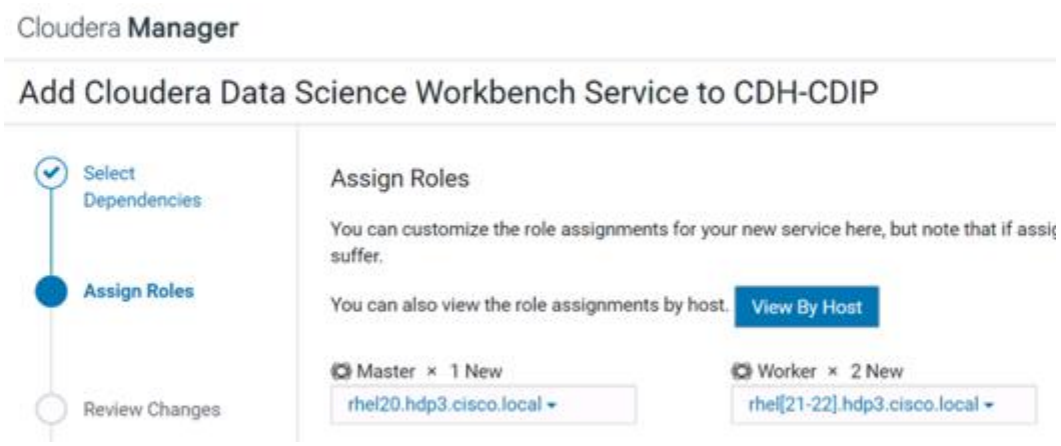


Add Cloudera Data Science Workbench Service to CDH-CDIP

Select Dependencies

	HBase	HDFS	Hive	Solr	Spark	YARN (MR2 Included)	ZooKeeper
○	HBase	HDFS	Hive		Spark	YARN (MR2 Included)	ZooKeeper
○	HBase	HDFS			Spark	YARN (MR2 Included)	ZooKeeper
●	HBase	HDFS	Hive	Solr	Spark	YARN (MR2 Included)	ZooKeeper
○	HBase	HDFS		Solr	Spark	YARN (MR2 Included)	ZooKeeper

5. Assign the Master and Worker roles to the gateway hosts. You must assign the Cloudera Data Science Work-bench Master role to one gateway host, and optionally, assign the Worker role to one or more gate-way hosts.


Cloudera Manager

Add Cloudera Data Science Workbench Service to CDH-CDIP

Select Dependencies

Assign Roles

Review Changes

Assign Roles

You can customize the role assignments for your new service here, but note that if assig
suffer.

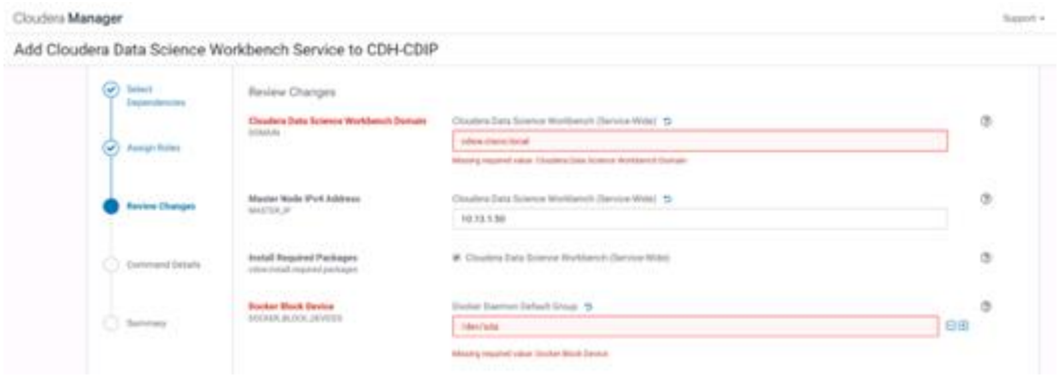
You can also view the role assignments by host. View By Host

Master × 1 New: rhel20.hdp3.cisco.local

Worker × 2 New: rhel[21-22].hdp3.cisco.local

Do not assign Master and Worker to the same host; even on single node deployments, the Master can perform the functions of a Worker as needed.

6. Configure the following parameters and click Continue.


Cloudera Manager

Add Cloudera Data Science Workbench Service to CDH-CDIP

Review Changes

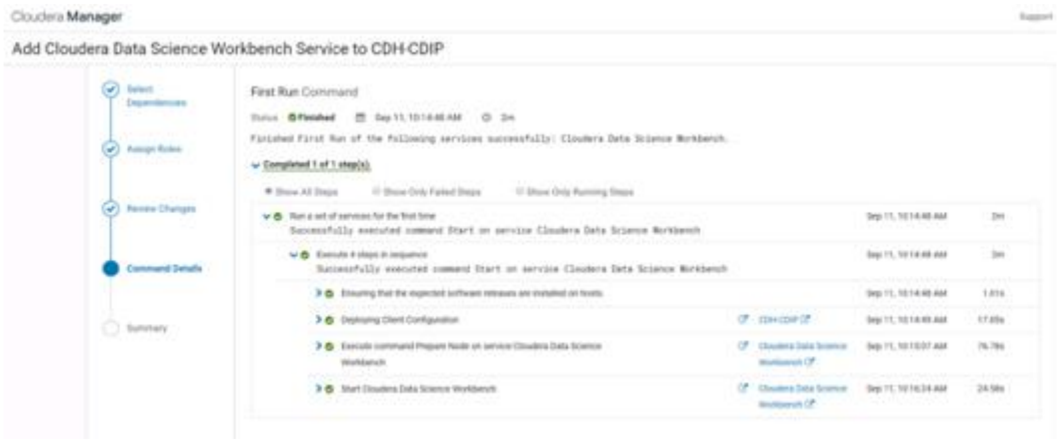
Cloudera Data Science Workbench Domain (DOMAIN) — Cloudera Data Science Workbench (Service-Wide): cdsw.cisco.local — Missing required value: Cloudera Data Science Workbench Domain

Master Node IPv4 Address (MASTER_IP) — Cloudera Data Science Workbench (Service-Wide): 10.13.1.50

Install Required Packages (cdsw.install.required.packages) — ✓ Cloudera Data Science Workbench (Service-Wide)

Docker Block Device (DOCKER_BLOCK_DEVICES) — Docker Daemon Default Group: /dev/sdc — Missing required value: Docker Block Device

7. The wizard will now begin a First Run of the Cloudera Data Science Workbench service. This includes deploying client configuration for HDFS, YARN and Spark 2, installing the package dependencies on all hosts, and formatting the Docker block device. The wizard will also assign the Application role to the host running Master, and the Docker Daemon role to all the Cloudera Data Science Workbench gateway hosts.


Cloudera Manager

Add Cloudera Data Science Workbench Service to CDH-CDIP

First Run Command

Status: Finished — Sep 11, 10:14:48 AM — 2m

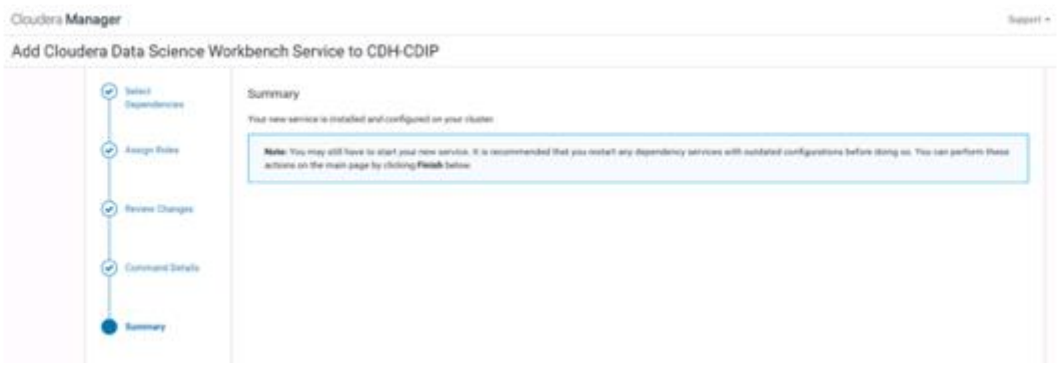
```
Finished First Run of the following services successfully: Cloudera Data Science Workbench.
```

Completed 1 of 1 step(s).

Step	Context	Date	Duration
Run a set of services for the first time — Successfully executed command Start on service Cloudera Data Science Workbench		Sep 11, 10:14:48 AM	2m
Execute 4 steps in sequence — Successfully executed command Start on service Cloudera Data Science Workbench		Sep 11, 10:14:48 AM	2m
Ensuring that the expected software releases are installed on hosts.		Sep 11, 10:14:48 AM	1.01s
Deploying Client Configuration	CDH-CDIP	Sep 11, 10:14:49 AM	17.85s
Execute command Prepare Node on service Cloudera Data Science Workbench	Cloudera Data Science Workbench	Sep 11, 10:15:07 AM	76.79s
Start Cloudera Data Science Workbench	Cloudera Data Science Workbench	Sep 11, 10:16:24 AM	24.58s

8. Once the First Run command has completed successfully, click Finish to go back to the Cloudera Manager home page.

CDSW will take 10-15 minutes to come online for the first time; they can follow the Docker/Master/Application process logs to track progress.



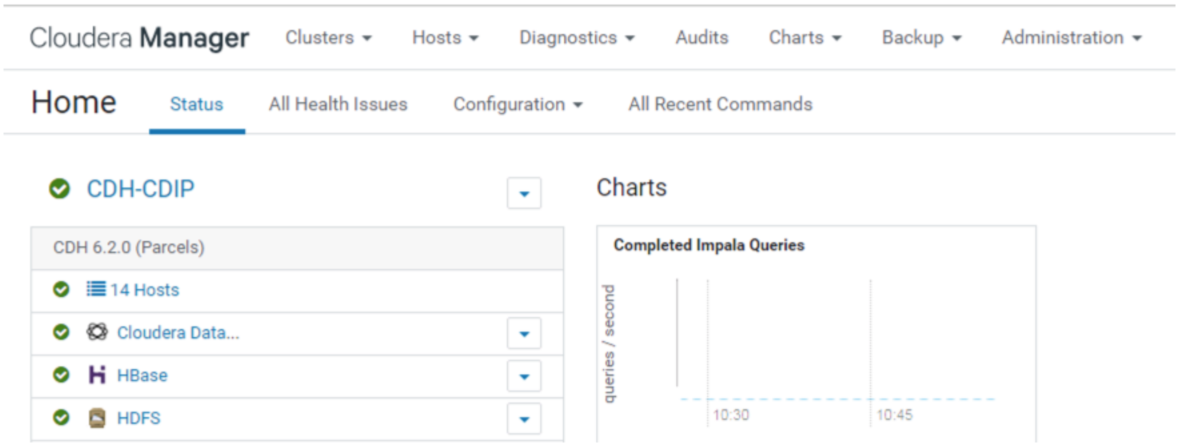
Add Cloudera Data Science Workbench Service to CDH-CDIP

Summary

Your new service is installed and configured on your cluster.

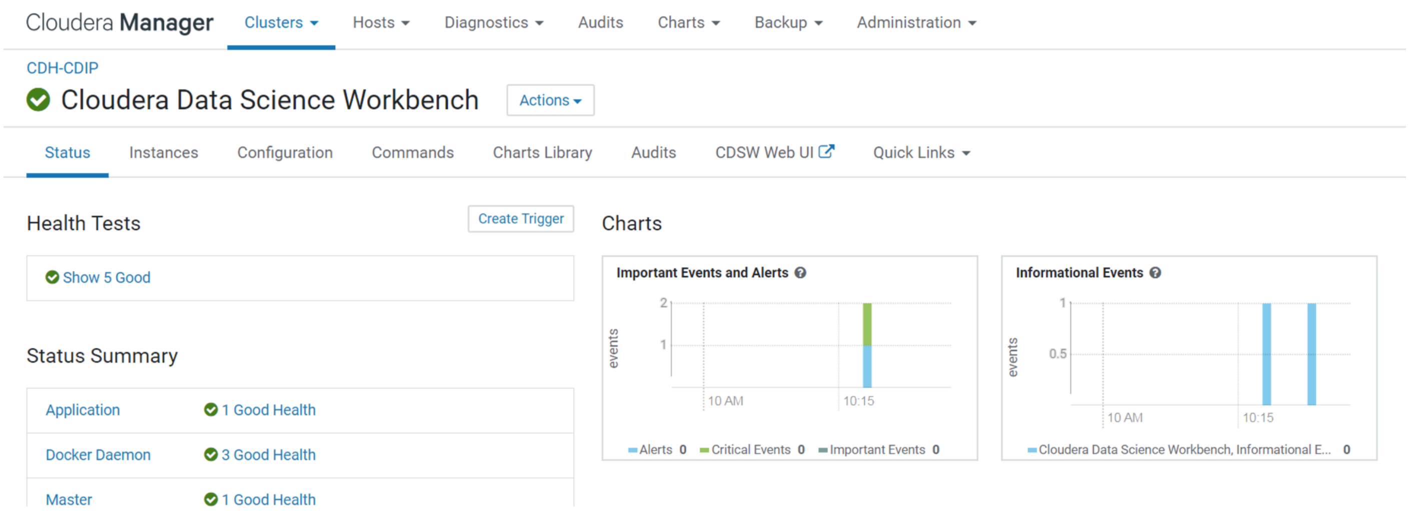
Note: You may still have to start your new service. It is recommended that you restart any dependency services with outdated configurations before doing so. You can perform these actions on the main page by clicking **Finish** below.

The Cloudera Manager Web Console reporting CDSW service configured on Cluster is shown below:



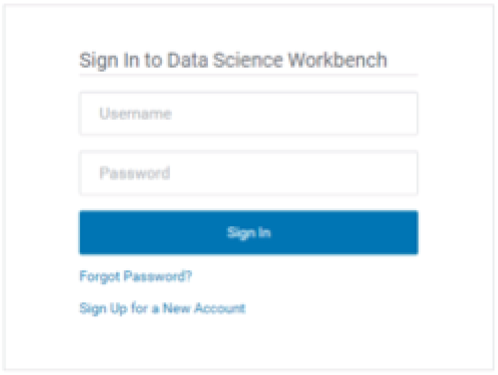
Create the Administrator Account

After your installation is complete, set up the initial administrator account. Go to the Cloudera Data Science Workbench web application or via CDSW WebUI from Cloudera Manager.



You must access Cloudera Data Science Workbench from the Cloudera Data Science Workbench Domain configured when setting up the service, and not the hostname of the master node. Visiting the hostname of the master node will result in a 404 error.

The first account that you create becomes the site administrator. You may now use this account to create a new project and start using the workbench to run data science workloads.



Non-Kerberized Clusters

In this CVD, we did not enable Kerberos.

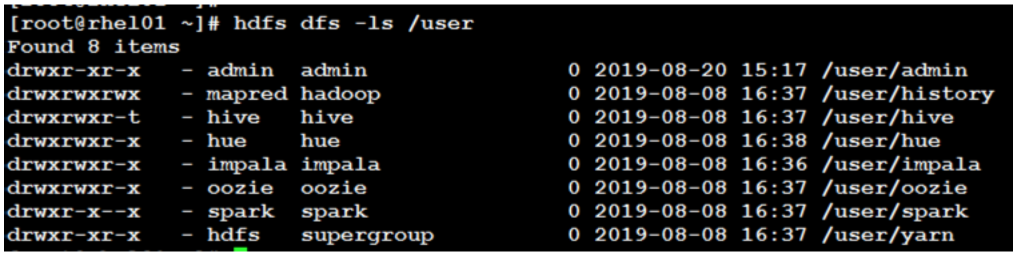
To enable Kerberos, refer to: [https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_kerberos.html.](https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_kerberos.html)

```
To disable Kerberos, delete the file /etc/krb5.conf on the CDSW nodes.
```

1. For a non-kerberized cluster, by default, your Hadoop username will be set to your Cloudera Data Science Workbench login username. Override this default and set an alternative HADOOP_USER_NAME ("admin" in this case which was the admin user created).

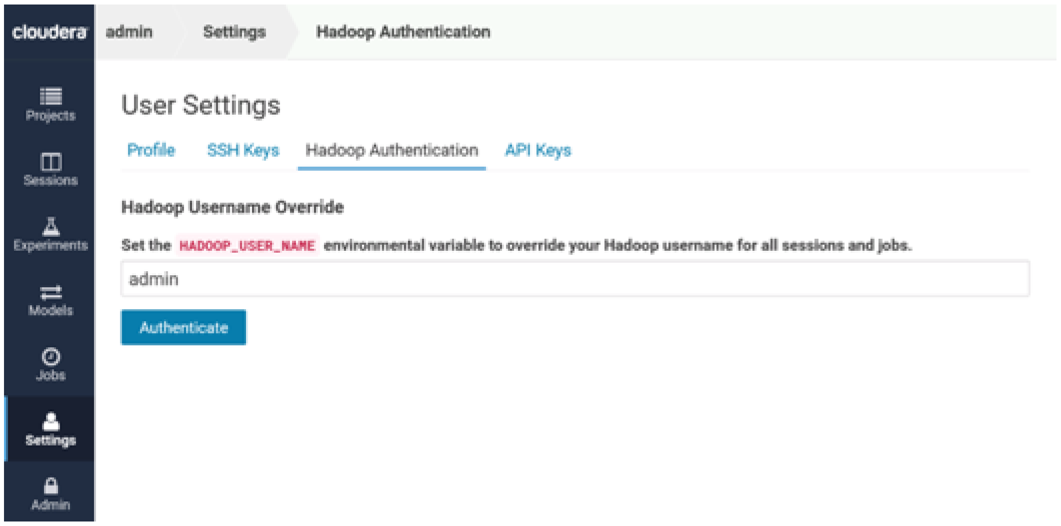
2. Create User admin in hdfs:

```
# sudo -u hdfs hdfs dfs -mkdir /user/admin
# sudo -u hdfs hdfs dfs -chown admin:admin /user/admin
```



```
[root@rhel01 ~]# hdfs dfs -ls /user
Found 8 items
drwxr-xr-x   - admin  admin            0 2019-08-20 15:17 /user/admin
drwxrwxrwx   - mapred hadoop           0 2019-08-08 16:37 /user/history
drwxrwxr-t   - hive   hive             0 2019-08-08 16:37 /user/hive
drwxrwxr-x   - hue    hue              0 2019-08-08 16:38 /user/hue
drwxrwxr-x   - impala impala           0 2019-08-08 16:36 /user/impala
drwxrwxr-x   - oozie  oozie            0 2019-08-08 16:37 /user/oozie
drwxr-x--x   - spark  spark            0 2019-08-08 16:37 /user/spark
drwxr-xr-x   - hdfs   supergroup       0 2019-08-08 16:37 /user/yarn
```

3. Go to the Settings on Cloudera Data Science WorkBench, Click Hadoop Authentication > Hadoop Username Override. Enter admin. Click Authenticate.



User Settings

Hadoop Username Override

Set the HADOOP_USER_NAME environmental variable to override your Hadoop username for all sessions and jobs.

admin

4. Restart CDSW from Cloudera Manager.

Use GPUs for Cloudera Data Science Workbench Workloads

A GPU is a specialized processor that can be used to accelerate highly parallelized computationally intensive workloads. Because of their computational power, GPUs have been found to be particularly well-suited to deep learning workloads. Ideally, CPUs and GPUs should be used in tandem for data engineering and data science workloads. A typical machine learning workflow involves data preparation, model training, model scoring, and model fitting. You can use existing general-purpose CPUs for each stage of the workflow, and optionally accelerate the math-intensive steps with the selective application of special-purpose GPUs. For example, GPUs allow you to accelerate model fitting using frameworks such as TensorFlow, PyTorch, Keras, MXNet, and Microsoft Cognitive Toolkit (CNTK).

By enabling GPU support, data scientists can share GPU resources available on Cloudera Data Science Workbench nodes. Users can request a specific number of GPU instances, up to the total number available on a node, which are then allocated to the running session or job for the duration of the run. Projects can use isolated versions of libraries, and even different CUDA and cuDNN versions via Cloudera Data Science Workbench's extensible engine feature.

For more information, go to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_gpu.html.

Enable GPU with CDSW

- Cloudera Data Science Workbench only supports CUDA-enabled NVIDIA GPU cards.
- Cloudera Data Science Workbench does not support heterogeneous GPU hardware in a single deployment.
- Cloudera Data Science Workbench does not include an engine image that supports NVIDIA libraries. Create your own custom CUDA-capable engine image using the instructions described in this topic.
- Cloudera Data Science Workbench does not install or configure the NVIDIA drivers on the Cloudera Data Science Workbench gateway hosts. These depend on your GPU hardware and will have to be installed by your system administrator. The steps provided in this topic are generic guidelines that will help you evaluate your setup.
- The instructions described in this topic require Internet access. If you have an airgapped deployment, you will be required to manually download and load the resources onto your hosts.
- For a list of known issues associated with this feature, refer Known Issues - GPU Support.

This section provides instructions about creating your own custom CUDA-capable engine image.

To enable Docker containers to use the GPUs, the previously installed NVIDIA driver libraries must be consolidated in a single directory named after the <driver_version> and mounted into the containers. This is done using the nvidia-docker package, which is a thin wrapper around the Docker CLI and a Docker plugin.

```
The following sample steps demonstrate how to use nvidia-docker to set up the
directory structure for the drivers so that they can be easily consumed by the
Docker containers that will leverage the GPU. Perform these steps on all nodes with
GPU hardware installed.
```

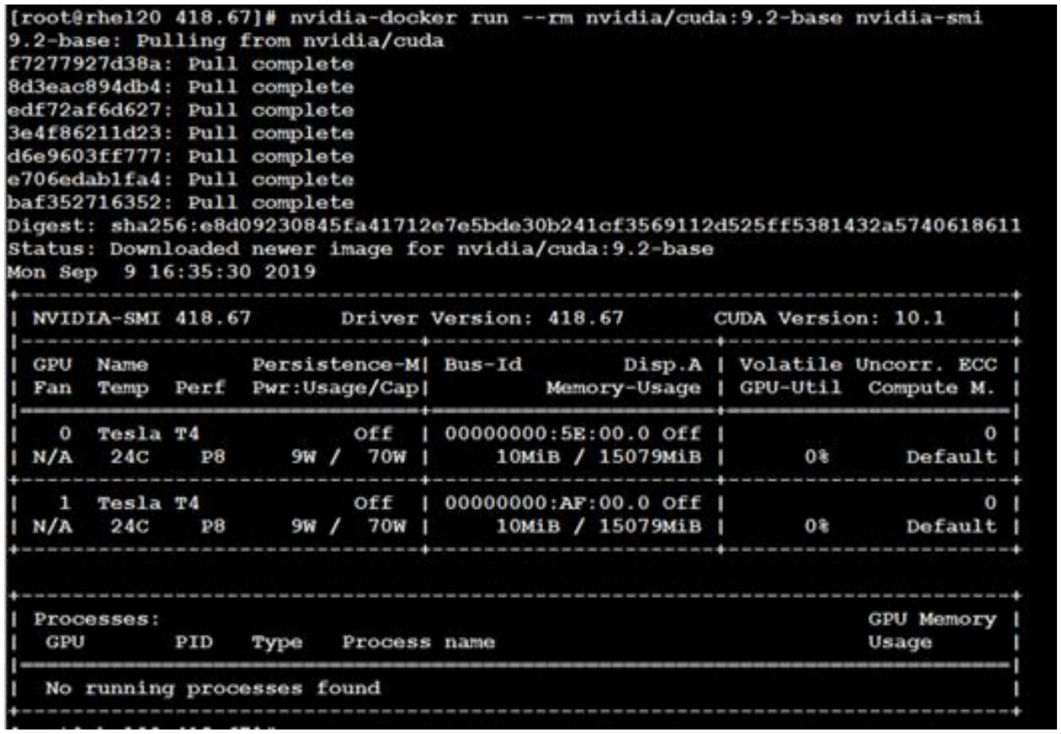
1. Download and install nvidia-docker:

```
# wget https://github.com/NVIDIA/nvidia-docker/releases/download/v1.0.1/nvidia-
docker-1.0.1-1.x86_64.rpm
# scp nvidia-docker-1.0.1-1.x86_64.rpm rhel01:/root/
Ansible cdswgpunodes -m copy -a "src=/root/nvidia-docker-1.0.1-1.x86_64.rpm
dest=/root/."
# ansible cdswgpunodes -m command -a "yum install -y nvidia-docker-1.0.1-
1.x86_64.rpm"
```

CDSW 1.5 requires nvidia-docker 1.0.

2. Run a small container to create the Docker volume structure:

```
#nvidia-docker run --rm nvidia/cuda:9.2-base nvidia-smi
```



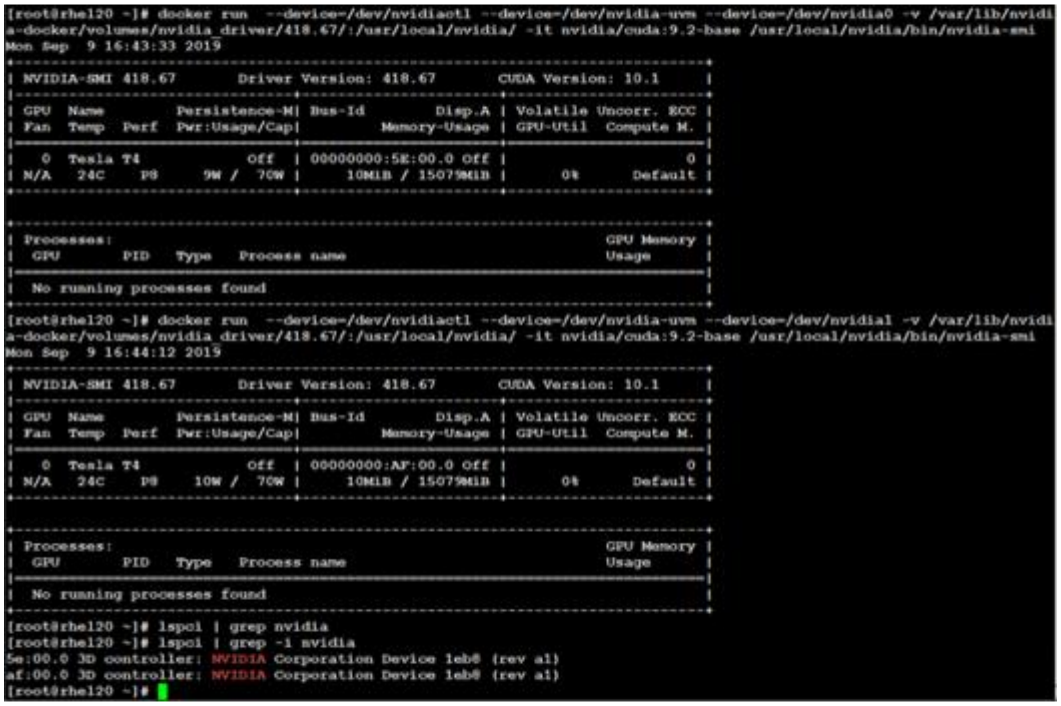
```
[root@rhel20 418.67]# nvidia-docker run --rm nvidia/cuda:9.2-base nvidia-smi
9.2-base: Pulling from nvidia/cuda
f7277927d38a: Pull complete
8d3eac894db4: Pull complete
edf72af6d627: Pull complete
3e4f86211d23: Pull complete
d6e9603ff777: Pull complete
e706edab1fa4: Pull complete
baf352716352: Pull complete
Digest: sha256:e8d09230845fa41712e7e5bde30b241cf3569112d525ff5381432a5740618611
Status: Downloaded newer image for nvidia/cuda:9.2-base
Mon Sep  9 16:35:30 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.67       Driver Version: 418.67       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   24C    P8     9W /  70W |     10MiB / 15079MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            Off  | 00000000:AF:00.0 Off |                    0 |
| N/A   24C    P8     9W /  70W |     10MiB / 15079MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

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

3. Use the following Docker command to verify that Cloudera Data Science Workbench can access the GPU:

```
#vi nvidia-gpu-access.sh

mkdir /var/lib/nvidia-docker/

mkdir /var/lib/nvidia-docker/volumes/

mkdir /var/lib/nvidia-docker/volumes/nvidia_driver/

mkdir /var/lib/nvidia-docker/volumes/nvidia_driver/418.67

mkdir /var/lib/nvidia-docker/volumes/nvidia_driver/418.67/bin

mkdir /var/lib/nvidia-docker/volumes/nvidia_driver/418.67/lib64

cp /usr/bin/nvidia* /var/lib/nvidia-docker/volumes/nvidia_driver/418.67/bin

cp /usr/lib64/libcuda* /var/lib/nvidia-docker/volumes/nvidia_driver/418.67/lib64

cp /usr/lib64/libnvidia* /var/lib/nvidia-
docker/volumes/nvidia_driver/418.67/lib64
```

Please replace the version of NVidia driver to one that matches your environment. Run “nvidia-smi” to capture the running version of NVidia driver.

```
# docker run --net host \

--device=/dev/nvidiactl \

--device=/dev/nvidia-uvm \

--device=/dev/nvidia0 \
-v /var/lib/nvidia-docker/volumes/nvidia_driver/418.67/:/usr/local/nvidia/ \
-it nvidia/cuda \
usr/local/nvidia/bin/nvidia-smi
```

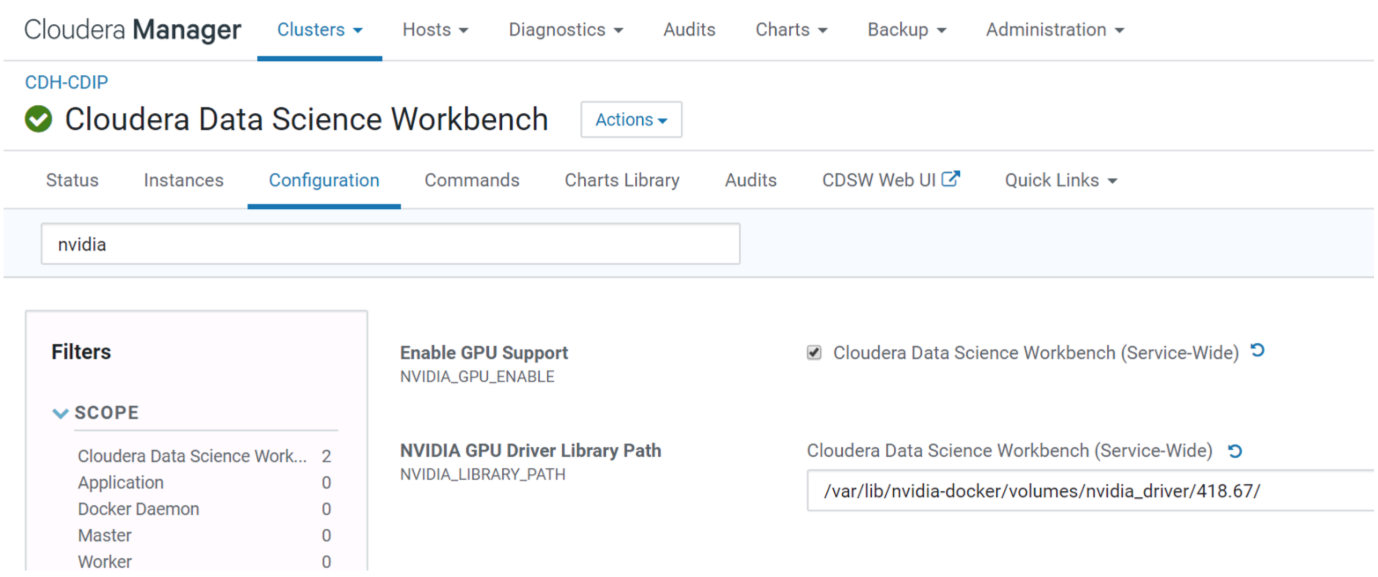


```
[root@rhel20 ~]# docker run  --device=/dev/nvidiactl --device=/dev/nvidia-uvm --device=/dev/nvidia0 -v /var/lib/nvidi
a-docker/volumes/nvidia_driver/418.67/:/usr/local/nvidia/ -it nvidia/cuda:9.2-base /usr/local/nvidia/bin/nvidia-smi
Mon Sep  9 16:43:33 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.67       Driver Version: 418.67       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   24C    P8     9W /  70W |      10MiB / 15079MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
[root@rhel20 ~]# docker run  --device=/dev/nvidiactl --device=/dev/nvidia-uvm --device=/dev/nvidia1 -v /var/lib/nvidi
a-docker/volumes/nvidia_driver/418.67/:/usr/local/nvidia/ -it nvidia/cuda:9.2-base /usr/local/nvidia/bin/nvidia-smi
Mon Sep  9 16:44:12 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.67       Driver Version: 418.67       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:AF:00.0 Off |                    0 |
| N/A   24C    P8    10W /  70W |      10MiB / 15079MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
[root@rhel20 ~]# lspci | grep nvidia
[root@rhel20 ~]# lspci | grep -i nvidia
5e:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)
af:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)
[root@rhel20 ~]#
```

Enable GPU Support in Cloudera Data Science Workbench

To enable GPU support for a CSD deployment, follow these steps:

1. Go to the CDSW service in Cloudera Manager. Click Configuration.
2. Search for the `Enable GPU Support` property and click the check box to enable it.



nvidia

Cloudera Data Science Work...	2
Application	0
Docker Daemon	0
Master	0
Worker	0

Enable GPU Support
NVIDIA_GPU_ENABLE
☑ Cloudera Data Science Workbench (Service-Wide)

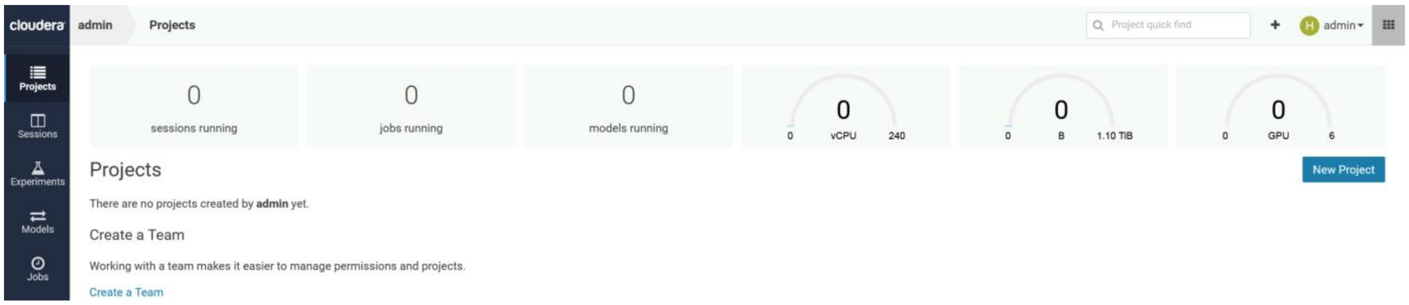
NVIDIA GPU Driver Library Path
NVIDIA_LIBRARY_PATH
Cloudera Data Science Workbench (Service-Wide)
/var/lib/nvidia-docker/volumes/nvidia_driver/418.67/

3. Click Save Changes.

4. Restart the CDSW service in Cloudera Manager.

Test Cloudera Data Science Workbench for GPUs

Once Cloudera Data Science Workbench has successfully restarted, if NVIDIA drivers have been installed on the Cloudera Data Science Workbench hosts, Cloudera Data Science Workbench will now be able to detect the GPUs available on its hosts.



Projects

There are no projects created by **admin** yet.

Create a Team

Working with a team makes it easier to manage permissions and projects.

Create a Team

Create a Custom CUDA-Capable Engine Image

The base engine image (`docker.repository.cloudera.com/cdsw/engine:8`) that ships with Cloudera Data Science Workbench will need to be extended with CUDA libraries to make it possible to use GPUs in jobs and sessions.

For more information, go to [https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_gpu.html.](https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_gpu.html)

Designate a Server for Docker and Start the Registry

To designate a server for Docker and start the registry, follow these steps:

Designate a server in the cluster for the Docker registry. Minimal resources are required, but sufficient disk space is needed to store the images and metadata. Docker must be installed and running.

Optional: By default, data will only be persisted within the container. If you would like to persist the data on the host, you can customize the bind mounts using the -v option.

1. Create /var/lib/registry folder:

```
# mkdir /var/lib/registry
```

2. Configure Docker to allow pulling from this insecure registry. Modify /etc/docker/daemon.json on all nodes in the cluster to include the following configuration options:

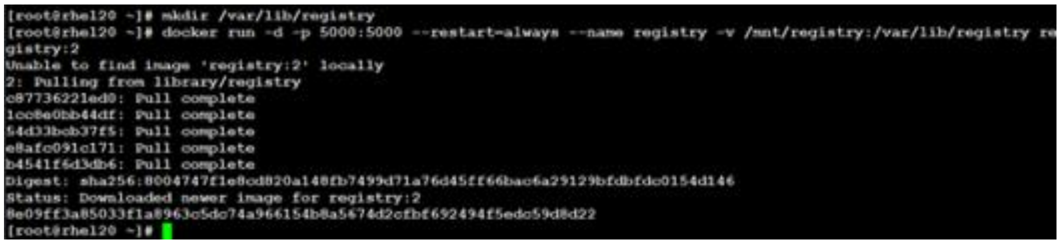
```
# vi /etc/docker/daemon.json

{


    "live-restore" : true,
    "debug" : true,

    "insecure-registries" : ["rhel20.hdp3.cisco.local:5000"]
}
```

3. Restart Docker on all nodes.

Run a Local Registry

To start the registry container, run the following command:

```
# docker run -d -p 5000:5000 --restart=always --name registry -v
/mnt/registry:/var/lib/registry registry:2
```



```
[root@rhel20 ~]# mkdir /var/lib/registry
[root@rhel20 ~]# docker run -d -p 5000:5000 --restart=always --name registry -v /mnt/registry:/var/lib/registry re
gistry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry
c87736221ed0: Pull complete
1cc8e0bb44df: Pull complete
54d33bcb37f5: Pull complete
e8afc091c171: Pull complete
b4541f6d3db6: Pull complete
Digest: sha256:8004747f1e8cd820a148fb7499d71a76d45ff66bac6a29129bfdbfdc0154d146
Status: Downloaded newer image for registry:2
8e09ff3a85033f1a8963c5dc74a966154b8a5674d2cfbf692494f5edc59d8d22
[root@rhel20 ~]#
```

Verify the registry container by running docker ps command:

```
# docker ps -a
```

Test the Docker Registry

To test the docker registry, follow the steps:

1. Pull the docker image.

```
# docker pull nvidia/cuda:9.2-base
```

2. Tag the image

```
# docker tag nvidia/cuda:9.2-base rhel20.hdp3.cisco.local:5000/cdip-demo-base
```

3. Push the image into private registry.

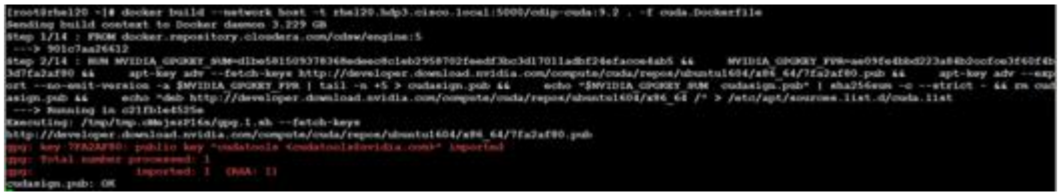
```
# docker push rhel20.hdp3.cisco.local:5000/cdip-demo-base
```

For more information about creating a local docker registry, go to: https://docs.docker.com/registry/deploying/#copy-an-image-from-docker-hub-to-your-registry

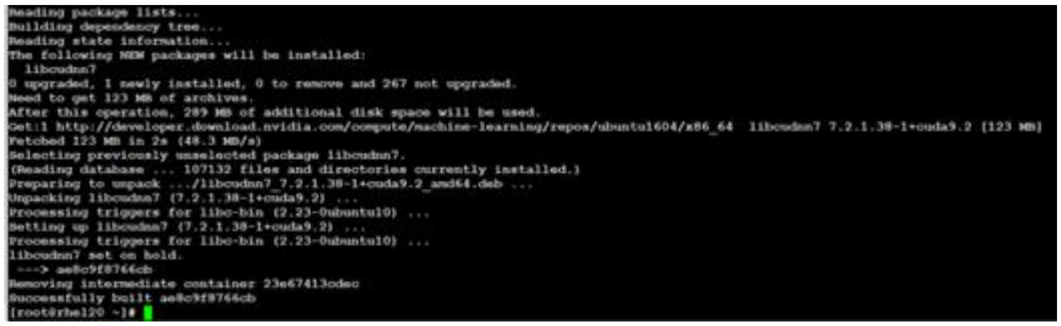
The following sample Dockerfile illustrates an engine on top of which machine learning frameworks such as TensorFlow and PyTorch can be used. This Dockerfile uses a deep learning library from NVIDIA called NVIDIA CUDA Deep Neural Network (cuDNN). Make sure you check with the machine learning framework that you intend to use in order to know which version of cuDNN is needed. As an example, TensorFlow 1.14.0 uses CUDA 10.0 and requires cuDNN 7.4.

To create the cuda.Dockerfile, run the following command:

```
FROM  docker.repository.cloudera.com/cdsw/engine:5

RUN
NVIDIA_GPGKEY_SUM=d1be581509378368edeec8c1eb2958702feedf3bc3d17011adbf24efacce4ab5
&& \
    NVIDIA_GPGKEY_FPR=ae09fe4bbd223a84b2ccfce3f60f4b3d7fa2af80 && \
    apt-key adv --fetch-keys
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.p
ub && \
    apt-key adv --export --no-emit-version -a $NVIDIA_GPGKEY_FPR | tail -n +5 >
cudasign.pub && \
    echo "$NVIDIA_GPGKEY_SUM  cudasign.pub" | sha256sum -c --strict - && rm
cudasign.pub && \
    echo "deb
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64 /" >
/etc/apt/sources.list.d/cuda.list

ENV CUDA_VERSION 9.2.148
LABEL com.nvidia.cuda.version="${CUDA_VERSION}"

ENV CUDA_PKG_VERSION 9-2=$CUDA_VERSION-1
RUN apt-get update && apt-get install -y --no-install-recommends \
        cuda-cudart-$CUDA_PKG_VERSION && \
    ln -s cuda-9.2 /usr/local/cuda && \
    rm -rf /var/lib/apt/lists/*

RUN echo "/usr/local/cuda/lib64" >> /etc/ld.so.conf.d/cuda.conf && \
    ldconfig

RUN echo "/usr/local/nvidia/lib" >> /etc/ld.so.conf.d/nvidia.conf && \
    echo "/usr/local/nvidia/lib64" >> /etc/ld.so.conf.d/nvidia.conf

ENV PATH /usr/local/nvidia/bin:/usr/local/cuda/bin:${PATH}
ENV LD_LIBRARY_PATH /usr/local/nvidia/lib:/usr/local/nvidia/lib64

RUN echo "deb http://developer.download.nvidia.com/compute/machine-
learning/repos/ubuntu1604/x86_64 /" > /etc/apt/sources.list.d/nvidia-ml.list

ENV CUDNN_VERSION 7.2.1.38
LABEL com.nvidia.cudnn.version="${CUDNN_VERSION}"

RUN apt-get update && apt-get install -y --no-install-recommends \
            libcudnn7=$CUDNN_VERSION-1+cuda9.2 && \
    apt-mark hold libcudnn7 && \
    rm -rf /var/lib/apt/lists/*
You can build a custom engine image from cuda.Dockerfile using the following sample
command:
 # docker build --network host -t rhel20.hdp3.cisco.local:5000/cdip-cuda:9.2 . -f
cuda.Dockerfile
```

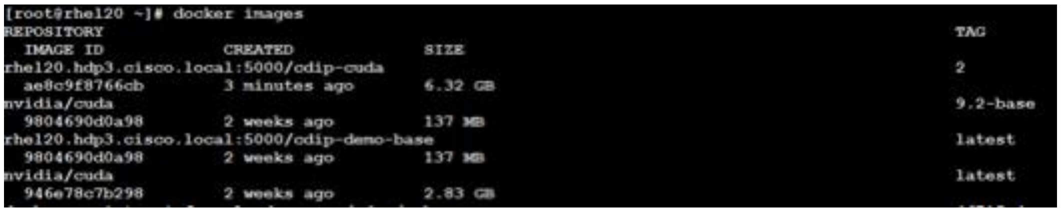


```
[root@rhel20 ~]# docker build --network host -t rhel20.hdp3.cisco.local:5000/cdip-cuda:9.2 . -f cuda.Dockerfile
Sending build context to Docker daemon 3.229 GB
Step 1/14 : FROM docker.repository.cloudera.com/cdsw/engine:5
 ---> 901c7aa26612
Step 2/14 : RUN NVIDIA_GPGKEY_SUM=d1be581509378368edeec8c1eb2958702feedf3bc3d17011adbf24efacce4ab5 &&     NVIDIA_GPGKEY_FPR=ae09fe4bbd223a84b2ccfce3f60f4b
3d7fa2af80 &&     apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub &&     apt-key adv --exp
ort --no-emit-version -a $NVIDIA_GPGKEY_FPR | tail -n +5 > cudasign.pub &&     echo "$NVIDIA_GPGKEY_SUM  cudasign.pub" | sha256sum -c --strict - && rm cud
asign.pub &&     echo "deb http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64 /" > /etc/apt/sources.list.d/cuda.list
 ---> Running in c21fb1e4525e
Executing: /tmp/tmp.dWejzzP16a/gpg.1.sh --fetch-keys
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub
gpg: key 7FA2AF80: public key "cudatools <cudatools@nvidia.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
cudasign.pub: OK
```



```
Reading package lists...
Building dependency tree...
Reading state information...
The following NEW packages will be installed:
  libcudnn7
0 upgraded, 1 newly installed, 0 to remove and 267 not upgraded.
Need to get 123 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Get:1 http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1604/x86_64  libcudnn7 7.2.1.38-1+cuda9.2 [123 MB]
Fetched 123 MB in 2s (48.3 MB/s)
Selecting previously unselected package libcudnn7.
(Reading database ... 107132 files and directories currently installed.)
Preparing to unpack .../libcudnn7_7.2.1.38-1+cuda9.2_amd64.deb ...
Unpacking libcudnn7 (7.2.1.38-1+cuda9.2) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
Setting up libcudnn7 (7.2.1.38-1+cuda9.2) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
libcudnn7 set on hold.
 ---> ae8c9f8766cb
Removing intermediate container 23e67413cdec
Successfully built ae8c9f8766cb
[root@rhel20 ~]#
```

```
docker images
```



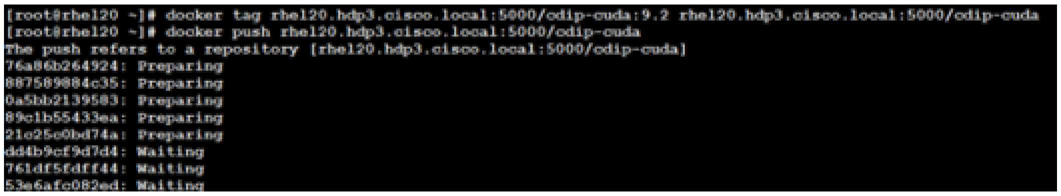
```
[root@rhel20 ~]# docker images
REPOSITORY                                                                                  TAG
  IMAGE ID            CREATED             SIZE
rhel20.hdp3.cisco.local:5000/cdip-cuda                                                      2
  ae8c9f8766cb        3 minutes ago       6.32 GB
nvidia/cuda                                                                                 9.2-base
  9804690d0a98        2 weeks ago         137 MB
rhel20.hdp3.cisco.local:5000/cdip-demo-base                                                 latest
  9804690d0a98        2 weeks ago         137 MB
nvidia/cuda                                                                                 latest
  946e78c7b298        2 weeks ago         2.83 GB
```

1. Tag the image as localhost:5000/<image name>. This creates an additional tag for the existing image. When the first part of the tag is a hostname and port, Docker interprets this as the location of a registry, when pushing.

```
 # docker tag rhel20.hdp3.cisco.local:5000/cdip-cuda:9.2
rhel20.hdp3.cisco.local:5000/cdip-cuda
```

2. Push the image to the local registry running at localhost:5000:

```
# docker push rhel20.hdp3.cisco.local:5000/cdip-cuda
```



```
[root@rhel20 ~]# docker tag rhel20.hdp3.cisco.local:5000/cdip-cuda:9.2 rhel20.hdp3.cisco.local:5000/cdip-cuda
[root@rhel20 ~]# docker push rhel20.hdp3.cisco.local:5000/cdip-cuda
The push refers to a repository [rhel20.hdp3.cisco.local:5000/cdip-cuda]
76a86b264924: Preparing
887589884c35: Preparing
0a5bb2139583: Preparing
89c1b55433ea: Preparing
21c25c0bd74a: Preparing
dd4b9cf9d7d4: Waiting
761df5fdff44: Waiting
53e6afc082ed: Waiting
```

Allocate GPUs for Sessions and Jobs

Once Cloudera Data Science Workbench has been enabled to use GPUs, a site administrator must whitelist the CUDA-capable engine image created in the previous step. Site administrators can also set a limit on the maximum number of GPUs that can be allocated per session or job.

To allocate GPUs for sessions and jobs, follow these steps:

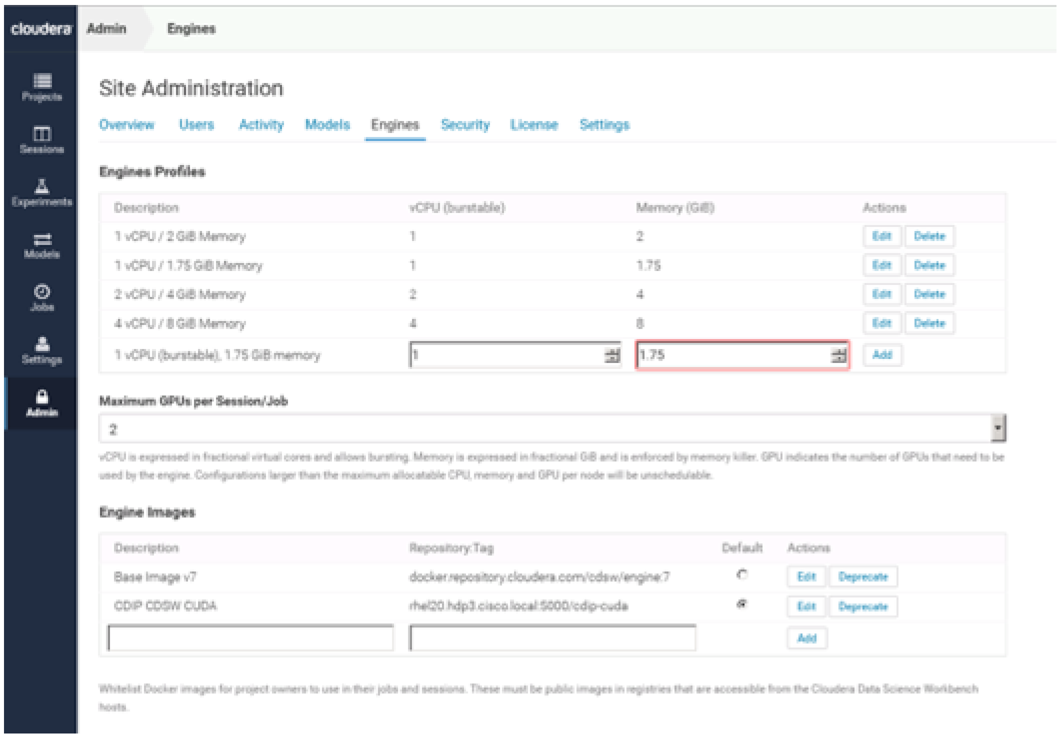
1. Sign into Cloudera Data Science Workbench as a site administrator.

2. Click Admin.

3. Go to the Engines tab.

4. From the Maximum GPUs per Session/Job drop-down list, select the maximum number of GPUs that can be used by an engine.

5. Under Engine Images, add the custom CUDA-capable engine image created in the previous step. This white-lists the image and allows project administrators to use the engine in their jobs and sessions. Enter description and Tag created for repository. Click Add.



Site Administration

Engines Profiles

Description	vCPU (burstable)	Memory (GiB)	Actions
1 vCPU / 2 GiB Memory	1	2	Edit Delete
1 vCPU / 1.75 GiB Memory	1	1.75	Edit Delete
2 vCPU / 4 GiB Memory	2	4	Edit Delete
4 vCPU / 8 GiB Memory	4	8	Edit Delete
1 vCPU (burstable), 1.75 GiB memory	1	1.75	Add

Maximum GPUs per Session/Job

2

vCPU is expressed in fractional virtual cores and allows bursting. Memory is expressed in fractional GiB and is enforced by memory killer. GPU indicates the number of GPUs that need to be used by the engine. Configurations larger than the maximum allocatable CPU, memory and GPU per node will be unschedulable.

Engine Images

Description	Repository:Tag	Default	Actions
Base Image v7	docker.repository.cloudera.com/cdsw/engine:7	○	Edit Deprecate
CDIP CDSW CUDA	rhel20.hdp3.cisco.local:5000/cdip-cuda	●	Edit Deprecate
			Add

Whitelist Docker images for project owners to use in their jobs and sessions. These must be public images in registries that are accessible from the Cloudera Data Science Workbench hosts.

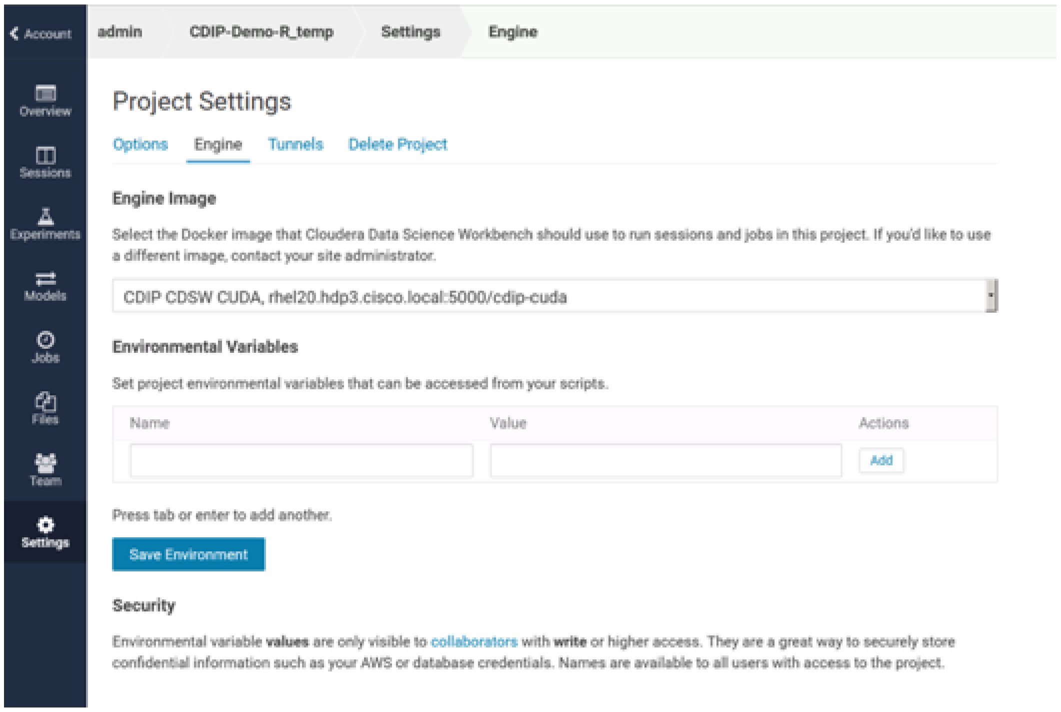
Project administrators can now whitelist the CUDA engine image to make it available for sessions and jobs within a project by following these steps:

1. Go to Projects Overview.

2. Click any existing project or create a new project and choose Settings.

3. Go to the Engines tab.

4. Under Engine Image, add the CUDA-capable engine image.



Project Settings

Options Engine Tunnels Delete Project

Engine Image

Select the Docker image that Cloudera Data Science Workbench should use to run sessions and jobs in this project. If you'd like to use a different image, contact your site administrator.

CDIP CDSW CUDA, rhel20.hdp3.cisco.local:5000/cdip-cuda

Environmental Variables

Set project environmental variables that can be accessed from your scripts.

Name	Value	Actions
		Add

Press tab or enter to add another.

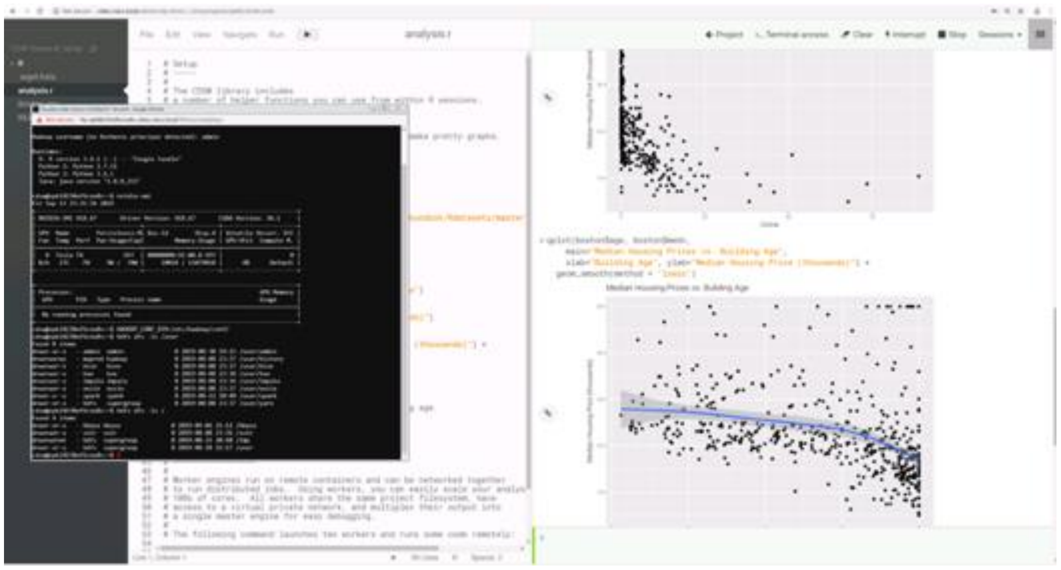
Save Environment

Security

Environmental variable **values** are only visible to collaborators with **write** or higher access. They are a great way to securely store confidential information such as your AWS or database credentials. Names are available to all users with access to the project.

5. Go to the Sessions tab and click an existing session or create a new session and allocate resources.

6. Click Terminal Access.



About the Authors

Hardik Patel, Big Data Solutions Architect, Cisco Systems, Inc.

Hardik Patel is a Big Data Solutions Architect at Computing Systems Product Group. He is part of the solution engineering team focusing on big data infrastructure, solutions, and performance.

Acknowledgements

For their support and contribution to the design, validation, and creation of this Cisco Validated Design, the author would like to thank:

- Karthik Kulkarni, Architect, Computing Systems Product Group, Cisco Systems, Inc.
- Muhammad Afzal, Architect, Computing Systems Product Group, Cisco Systems, Inc.