

Cisco Data Intelligence Platform on Cisco UCS C4200 with Cloudera Data Platform

Design and Deployment Guide for Modernizing Data Lake with All Flash Cisco UCS C4200 Platform with Cisco UCS C125 M5 on Cloudera Data Platform Private Cloud Base 7.1.3 Managed by Cisco Intersight

Published: October 2020



In partnership with:

CLOUDERA

About the Cisco Validated Design Program

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

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

ALL DESIGNS, SPECIFICATIONS, STATEMENTS, INFORMATION, AND RECOMMENDATIONS (COLLECTIVELY, "DESIGNS") IN THIS MANUAL ARE PRESENTED "AS IS," WITH ALL FAULTS. CISCO AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THE DESIGNS, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Inter-network Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unified Computing System (Cisco UCS), Cisco UCS B-Series Blade Servers, Cisco UCS C-Series Rack Servers, Cisco UCS S-Series Storage Servers, Cisco UCS Manager, Cisco UCS Management Software, Cisco Unified Fabric, Cisco Application Centric Infrastructure, Cisco Nexus 9000 Series, Cisco Nexus 7000 Series, Cisco Prime Data Center Network Manager, Cisco NX-OS Software, Cisco MDS Series, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, Giga-Drive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, Power-Panels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries. Lisa.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0809R)

© 2020 Cisco Systems, Inc. All rights reserved.

Contents

| | |
|--|-----|
| Executive Summary | 4 |
| Solution Overview | 6 |
| Technology Overview | 18 |
| Solution Design | 27 |
| Deployment Hardware and Software | 35 |
| Summary | 166 |
| Bill of Materials | 167 |
| Appendix..... | 169 |
| About the Author..... | 193 |
| Feedback | 194 |

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](#).

Hadoop enables data engineering, providing very fast ingestion of data and Extract, Transform, and Load (ETL) processing. In a data-intensive workload, computing moves to the data to enable faster, distributed processing of the data. Building a data pipeline that receives data flows from different data sources at higher velocities, performs ETL on this data so that it lands in HDFS, and then makes it available for the serving layer either for real-time streaming or batch processing, is an extremely I/O intensive operation.

Cisco UCS C4200 platform is primarily positioned for All SSDs and the AMD Rome EPYC 2 series CPUs with highest core density for x86 architecture provenly ideal platform for the Hadoop data and compute intensive applications. Given the higher server density within 2RU form factor and the additional network capabilities running as single socket for Hadoop, Cisco UCS C125 M5 Rack Server Node is predominantly an excellent server architecture for Hadoop data lake on Cloudera Data Platform deployment. This CVD implements the data lake tier of the Cisco Data Intelligence Platform with Cloudera Data Platform Data Center (CDP-Private Cloud Base). CDP combines the best of both worlds, such as Cloudera Enterprise Data Hub and Hortonworks Data Platform Enterprise along with new features and enhancements across the stack. This unified distribution is a scalable and customizable platform where end user can securely run many types of workloads.

Furthermore, this CVD with CDP-Private Cloud Base sets the foundation for CDP private cloud which offers cloud-like user experience with self-service portal where users can efficiently search, curate, and share data, enabling access to trusted data and analytics in a secured manner.

This solution offers cohesive platform for both IT and data scientists by providing a scalable infrastructure for IT while also providing application platform for data scientists.

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 Cloudera Data Platform Private Cloud Base on 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.

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 or 6454 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 Data Platform Data Center 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, design choices, and deployment procedures for Cisco Data Intelligence Platform using Cloudera Data Platform DC on Cisco UCS C125 M Rack Server Node.

This document also serves as a step-by-step guide on how to deploy CDP Private Cloud Base on 48 node cluster of Cisco UCS C125 M5 Rack Server Nodes.

What's New in this Release?

This solution extends the portfolio of Cisco Data Intelligence Platform (CDIP) architecture with Cloudera Data Platform Data Center, a state-of-the-art platform, providing a data cloud for demanding workloads that is easy to deploy, scale and manage. Furthermore, as the enterprise's requirements and needs changes overtime, the platform can grow to thousands of servers, hence providing peta bytes of storage.

The following design consideration will be implemented in this validated design:

- Modernizing the Data Lake with All flash C4200 platform with single socket AMD based server nodes using Cloudera Data Platform Datacenter for Big Data and Analytics
- Cisco Intersight deployed UCSM managed cluster with some feature limitations (outlined in the Intersight deployment sections)

What's Next?

This CVD showcases the Hadoop cluster deployment using Cisco Intersight (partially covered features) and UCS manager. This solution can also be fully deployed using Cisco Intersight in the future. Additional Cisco UCS features will be added to the Appendix. Some of the industry driven platforms and services include the following:

- Cloudera Data Platform Private Cloud
- Apache Ozone - Object Store for the dis-aggregated compute and storage
- A fully integrated CDP on CDIP with
 - Data lake enabled through fully supported production grade CDP Private Cloud Base
 - AI/ML enabled through CDP Private Cloud using REDHAT OpenShift
 - Exabyte storage enabled through Apache Ozone

Solution Summary

This CVD details the process of installing Cloudera Data Platform Data Center 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.

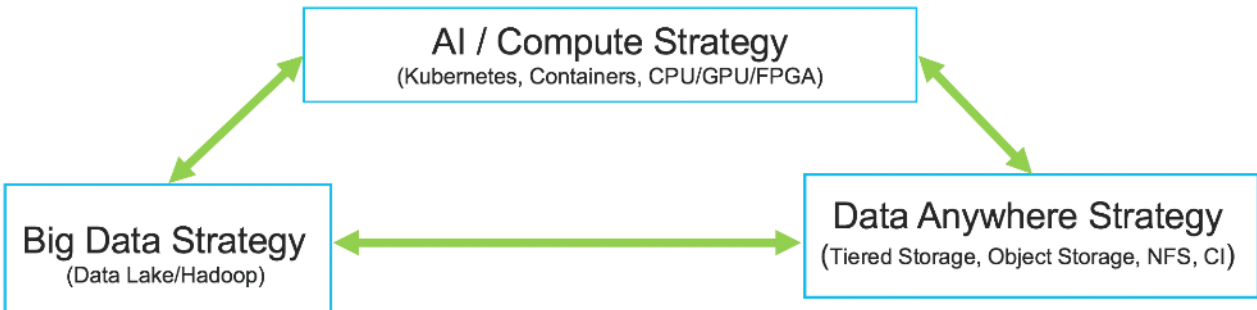
Cisco Data Intelligence Platform

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

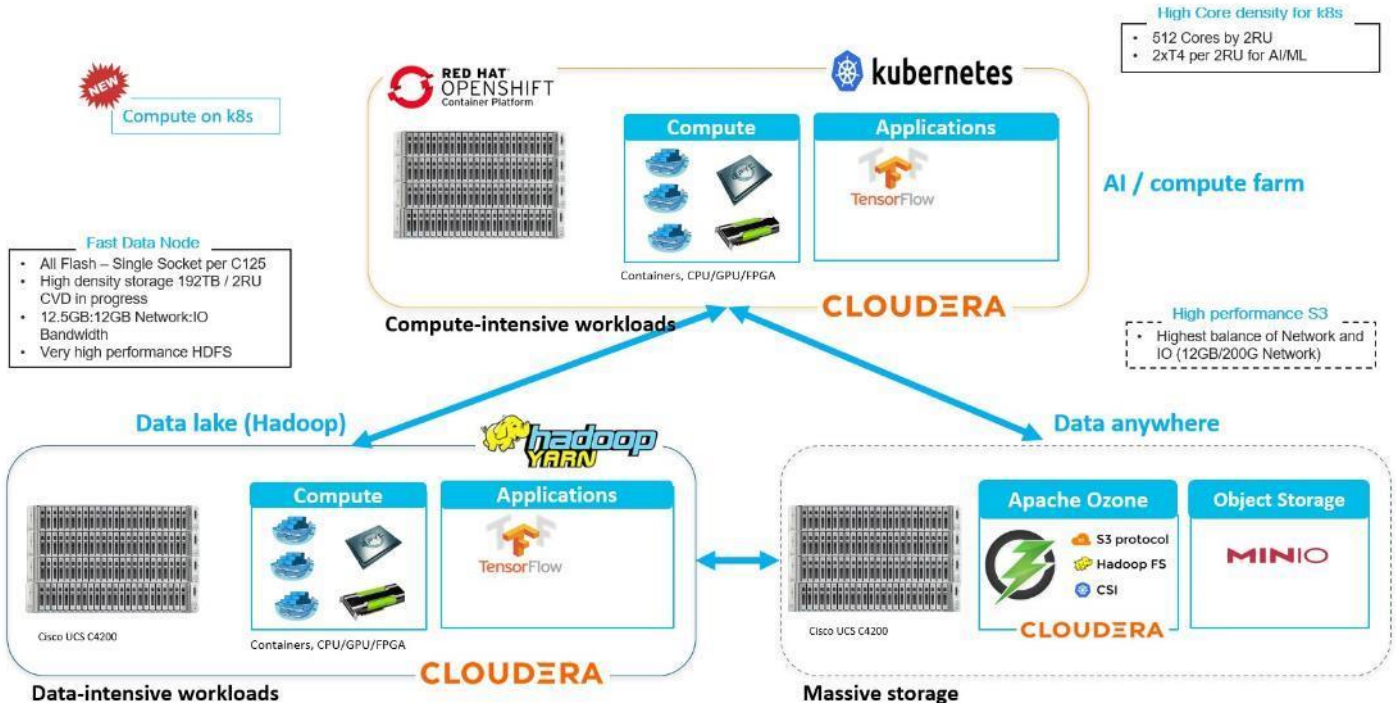


Cisco has developed numerous industry leading Cisco Validated Designs (reference architectures) in the area of Big Data (CVDs with Cloudera, Hortonworks and MapR), compute farm with Kubernetes (CVD with RedHat OpenShift) and Object store (Apache Ozone, MinIO, SwiftStack, and others).

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
- CDIP with CDP 7.1.3 with CDSW or CML (powered by Kubernetes/OpenShift) and Tiered Storage with Ozone Object Store

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

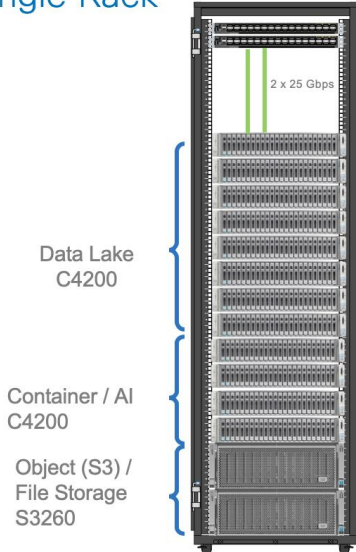


This architecture can start from a single rack and scale to thousands of nodes with a single pane of glass management with Cisco Application Centric Infrastructure (ACI).

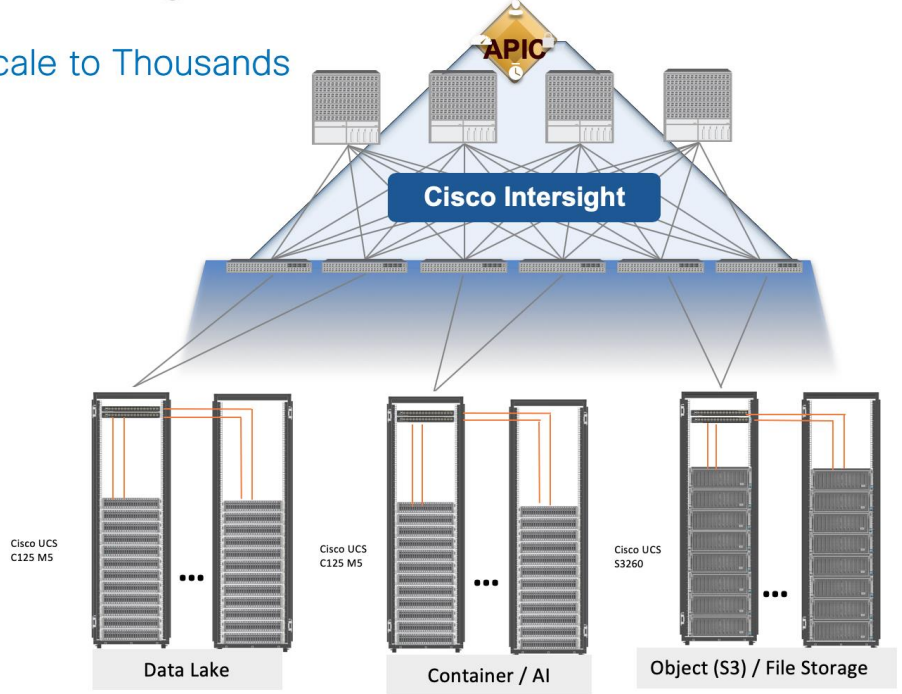
Figure 3. Cisco Data Intelligent Platform at Scale

Cisco Data Intelligent Platform

Can Start Small
- Single Rack

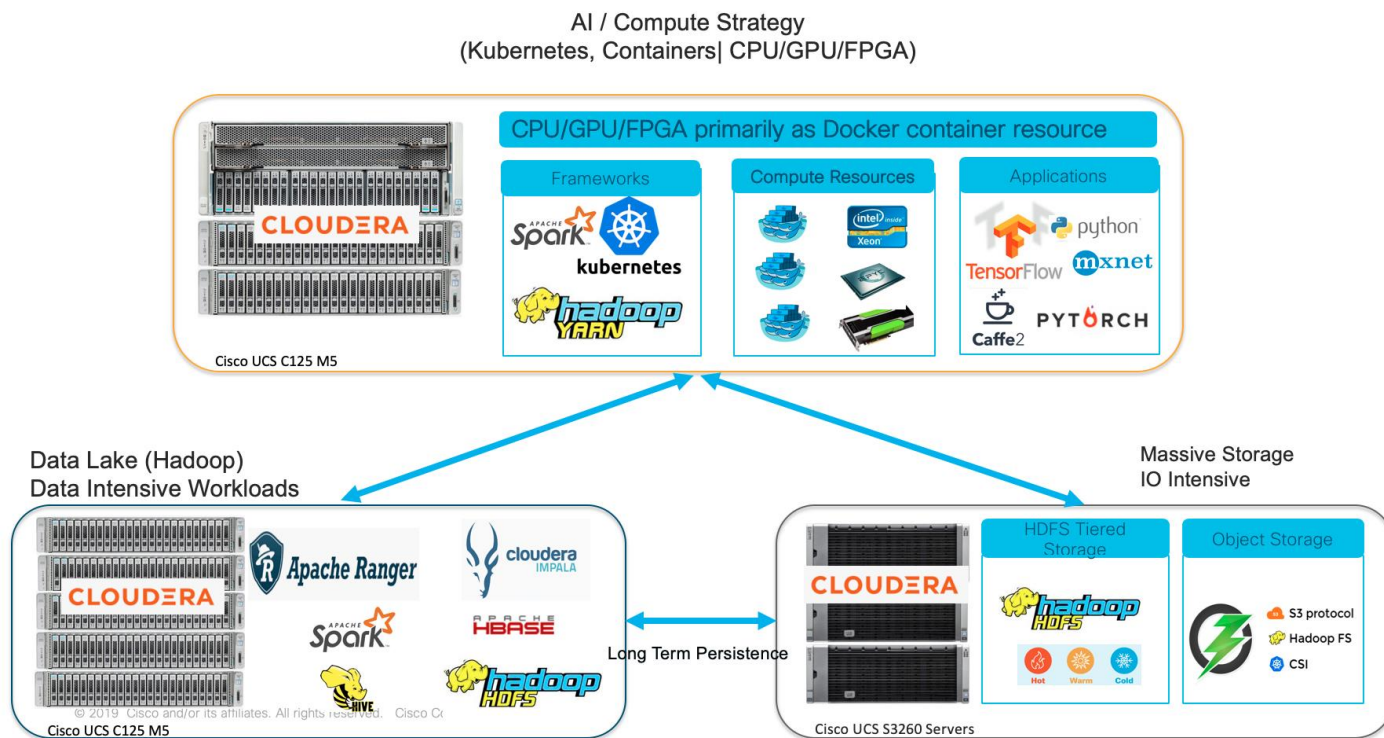


Scale to Thousands



CDP PVC Base on CDIP

Figure 4. Cloudera Data Platform on Cisco Data Intelligent Platform



A CDIP architecture can fully be enabled by Cloudera Data Platform with the following components:

- Data lake enabled through CDP Private Cloud Base
- AI/ML enabled through CDP Private Cloud and
- Exabyte storage enabled through Apache Ozone

Reference Architecture

The reference architecture for Cisco UCS C4200 Series Chassis with Cisco UCS C125 M5 Rack Server Nodes powered by AMD EPYC processors and Hortonworks' big data distribution is optimally designed and tested to help ensure a balance between performance and capacity. It can scale out to meet big data and analytics requirements. It can expand to thousands of servers with Cisco Nexus® 9000 Series Switches using the Cisco® Application Policy Infrastructure Controller (APIC) with a leaf-and-spine design using the Cisco Application Centric Infrastructure (Cisco ACI™) platform. This next generation infrastructure can be deployed to meet a wide variety of computing, storage, and connectivity options.

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

| | Superior Performance | Performance | Superior Performance | Performance |
|------------------------------|---|--|--|---|
| Servers | 9 x Cisco UCS C4200 Series Chassis Each with 4 x Cisco UCS C125 M5 Rack Server Nodes | 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 | 1 x AMD EPYC 2 7532 Processor (1 x 32 cores, 2.4 GHz) | 2 x 2 nd Gen Intel® Xeon® Scalable 6230 processors (2 x 20 cores, at 2.1 GHz) | 2 x 2 nd Gen Intel Xeon Scalable 6230 processors (2 x 20 cores, at 2.1 GHz) | 2 x 2 nd Gen Intel Xeon Processor Scalable Family 5220 (2 x 18 cores, 2.2 GHz) |
| Memory | 8 x 64 GB RDIMMs (512 GB) | 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 SATA SSDs | M.2 with 2 x 240-GB SSDs | M.2 with 2 x 240-GB SSDs | 2 x 240-GB SATA SSDs |
| Storage | 6 x 7.6 TB Enterprise Value SATA SSD | 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 per server node |
| Virtual interface card (VIC) | 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) or |
| Storage controller | Cisco 12-Gbps SAS 9460-8i RAID Controller 2GB cache (FBWC) | 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 | 2 x Cisco UCS 6454 Fabric Interconnect | Cisco UCS 6332 Fabric Interconnect or Cisco UCS 6454 Fabric | Cisco UCS 6332 Fabric Interconnect or Cisco UCS 6454 Fabric Interconnect | Cisco UCS 6332 Fabric Interconnect |

| | Superior Performance | Performance | Superior Performance | Performance |
|----------------|----------------------|--|--|-------------|
| | | Interconnect | | |
| GPU (optional) | NA | Up to 2 x NVIDIA Tesla V100 with 32 GB memory each Or Up to 6 x NVIDIA Tesla T4 with 16 GB memory each | 2 x NVIDIA Tesla V100 with 32 GB memory each Or Up to 6 x NVIDIA Tesla T4 with 16 GB memory each | NA |



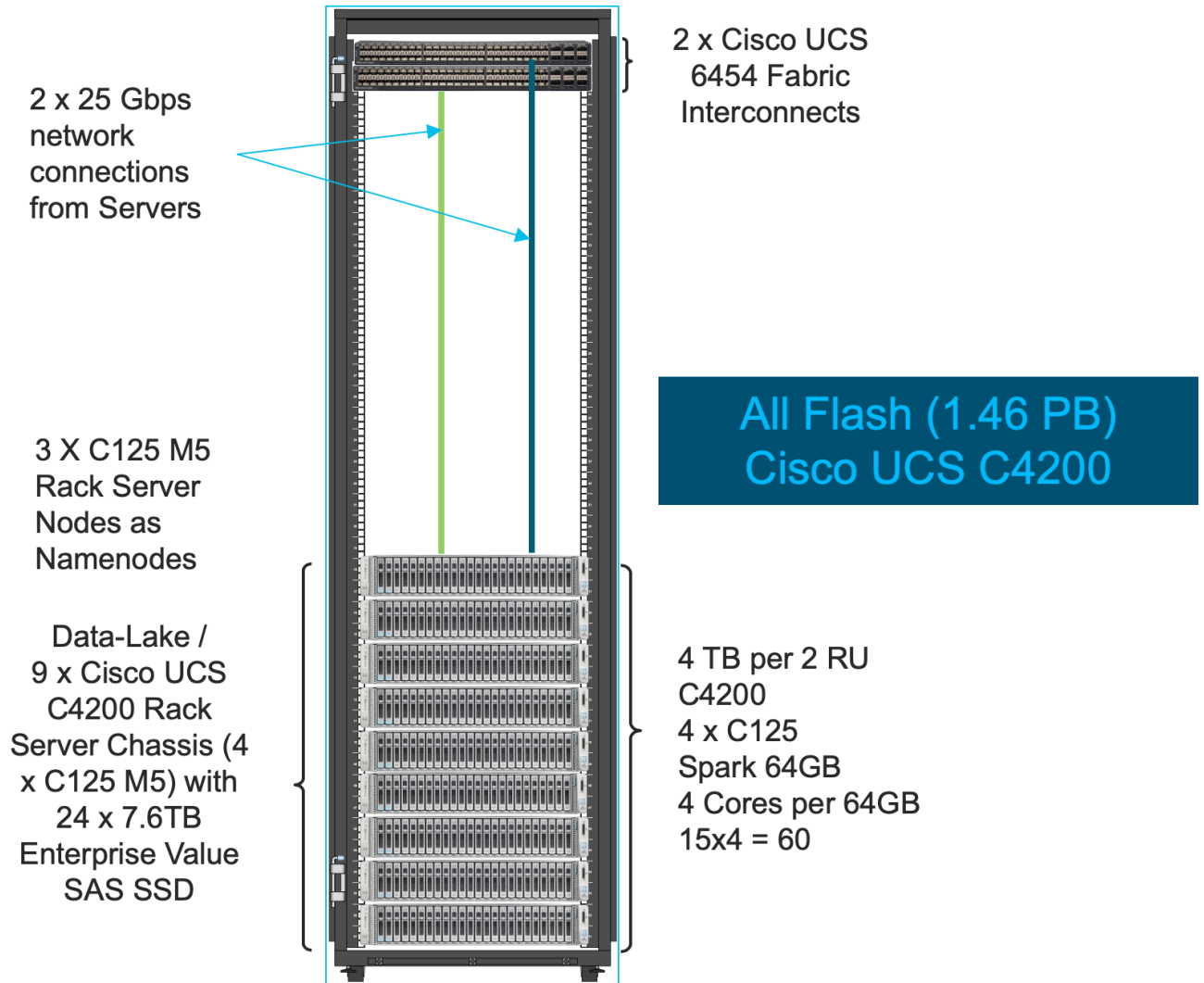
The above mentioned non-Superior Performance configuration models can also be deployed with the 4th Generation Cisco UCS 6454 Fabric Interconnect with 25G VIC.



In this architecture, we deployed Cisco UCS C125 M5 Rack Server Nodes with Intersight capabilities.

As illustrated in [Figure 5](#), a 9 node cluster with Rack#1 hosting 32 x Cisco UCS M5 Rack Server Node as data nodes and 3 x Cisco UCS C125 M5 Rack Server Node as name nodes. Each link in the figure represents a 25 Gigabit Ethernet link from each of the 48 servers directly connected to a Fabric Interconnect.

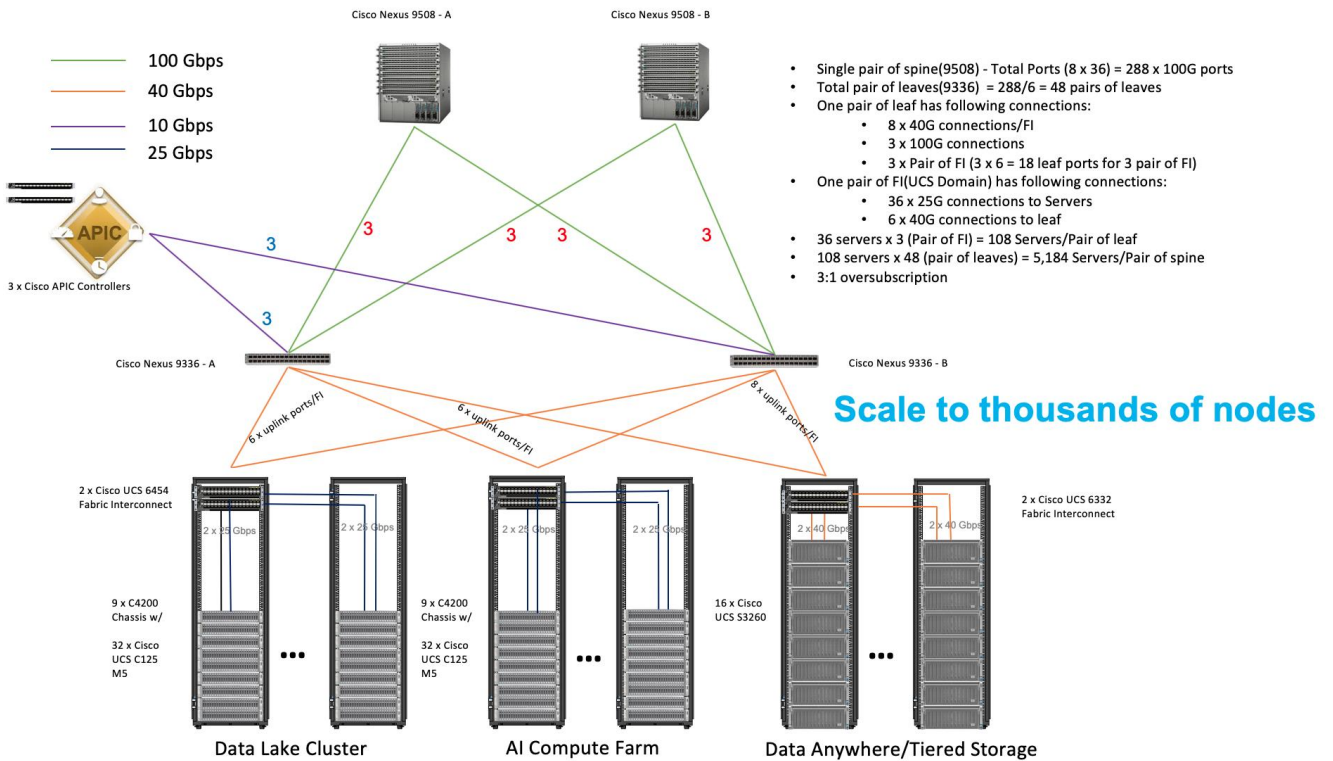
Figure 5. Cisco Data Intelligence Platform with Cloudera Data Platform Data Center - Data Lake



Scaling the Solution

[Figure 6](#) illustrates how to scale the solution. Each pair of Cisco UCS 6454 Fabric Interconnects has 36 Cisco UCS C125 M5 servers connected to it. This allows for six uplinks from each Fabric Interconnect to the Cisco Nexus 9336 switch. Three pairs of Cisco UCS 6454 FI's can connect to a single switch with four uplink ports each. With 36 servers per FI, a total of 108 servers can be supported. Additionally, this solution can scale to thousands of nodes with the Nexus 9500 series family of switches.

Figure 6. Scaling the Solution



In the reference architectures discussed, 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

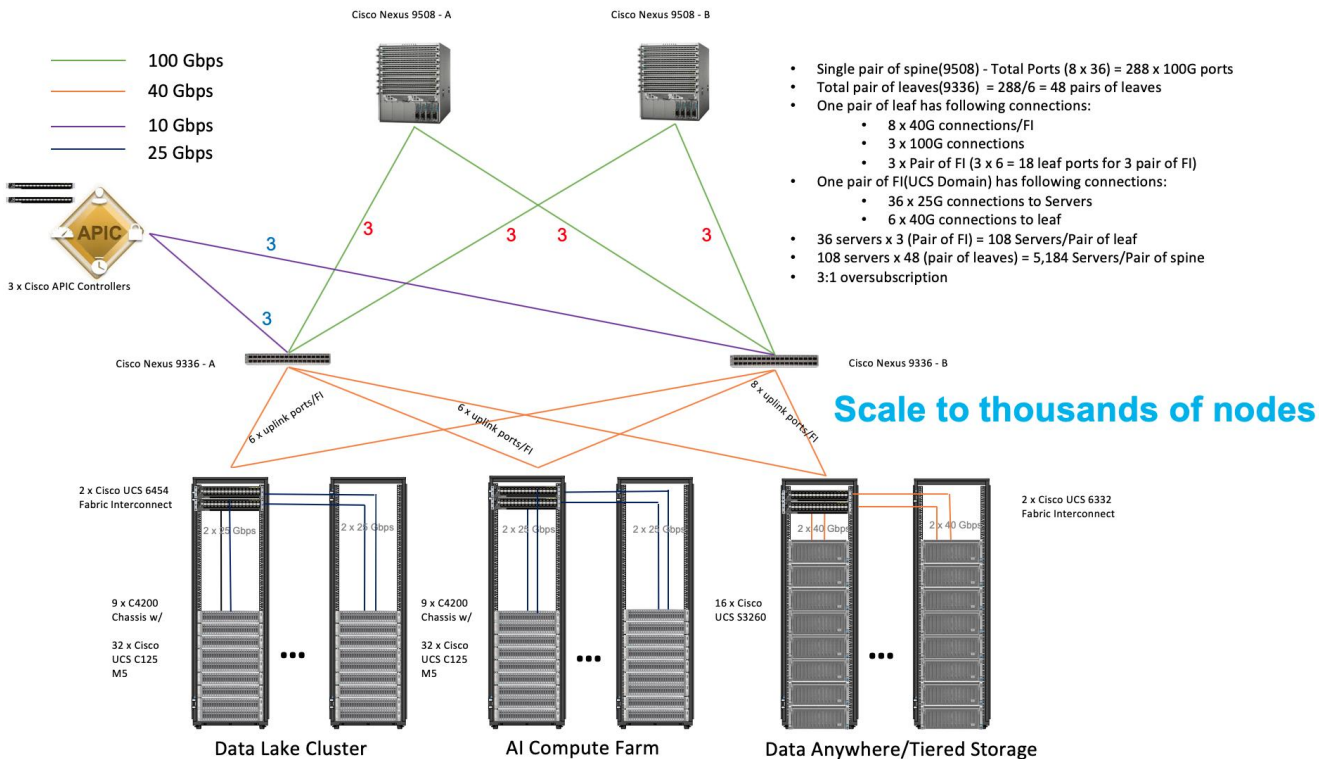
In the following scenarios, the goal is to populate up to a maximum of 200 leaf nodes in a Cisco ACI domain. Not all cases reach that number because they use the Cisco Nexus® 9508 Switch for this sizing and not the Cisco Nexus 9516 Switch.

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, 36 x Cisco UCS C125 M5 Servers are connected to a pair of Cisco UCS 6454 Fabric Interconnects (with 48 x 25-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 a pair of 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



Scaled Architecture with 2:1 Oversubscription with Cisco ACI

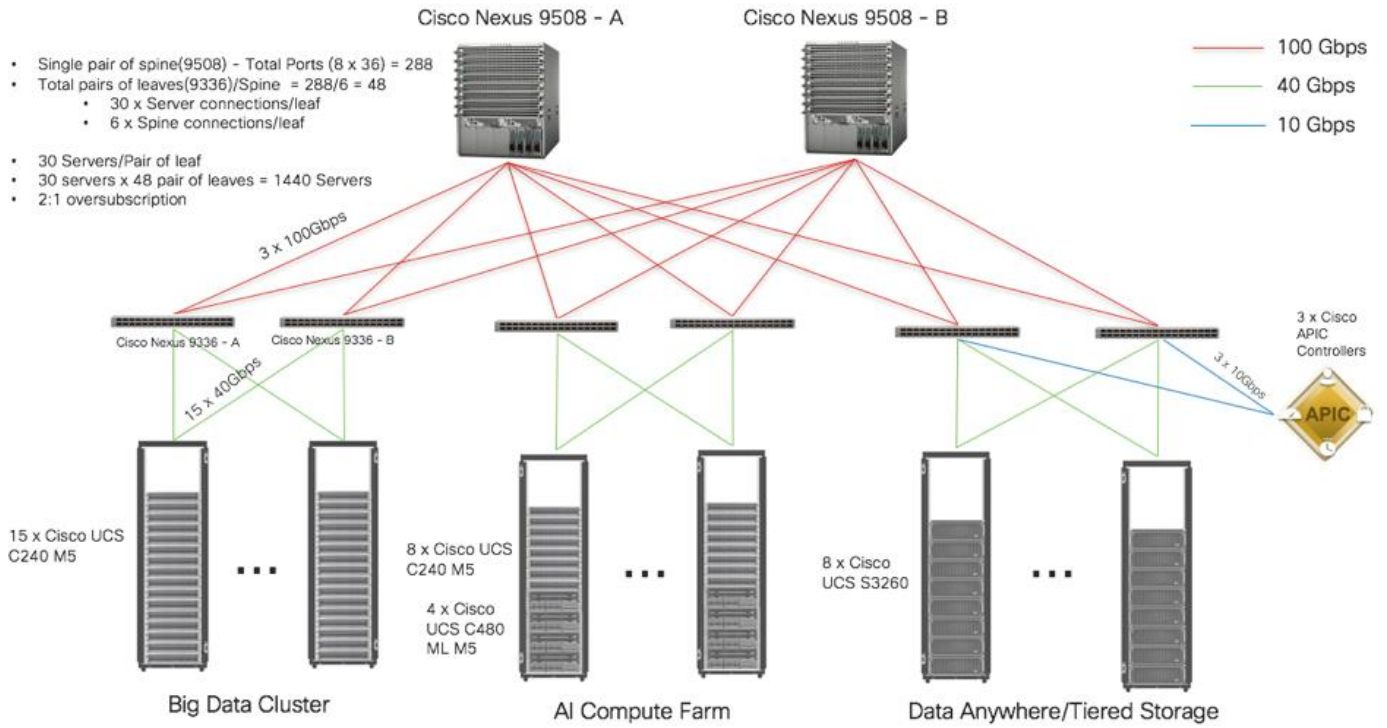
In the scenario discussed here and shown in [Figure 8](#), the Cisco Nexus 9508 Switch with the Cisco N9K-X9736C-FX line card can support up to 36 x 100-Gbps ports, each and 8 such line cards.

Here, for the 2:1 oversubscription, 30 Cisco UCS C125 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 C125 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 to 1440 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



Technology Overview

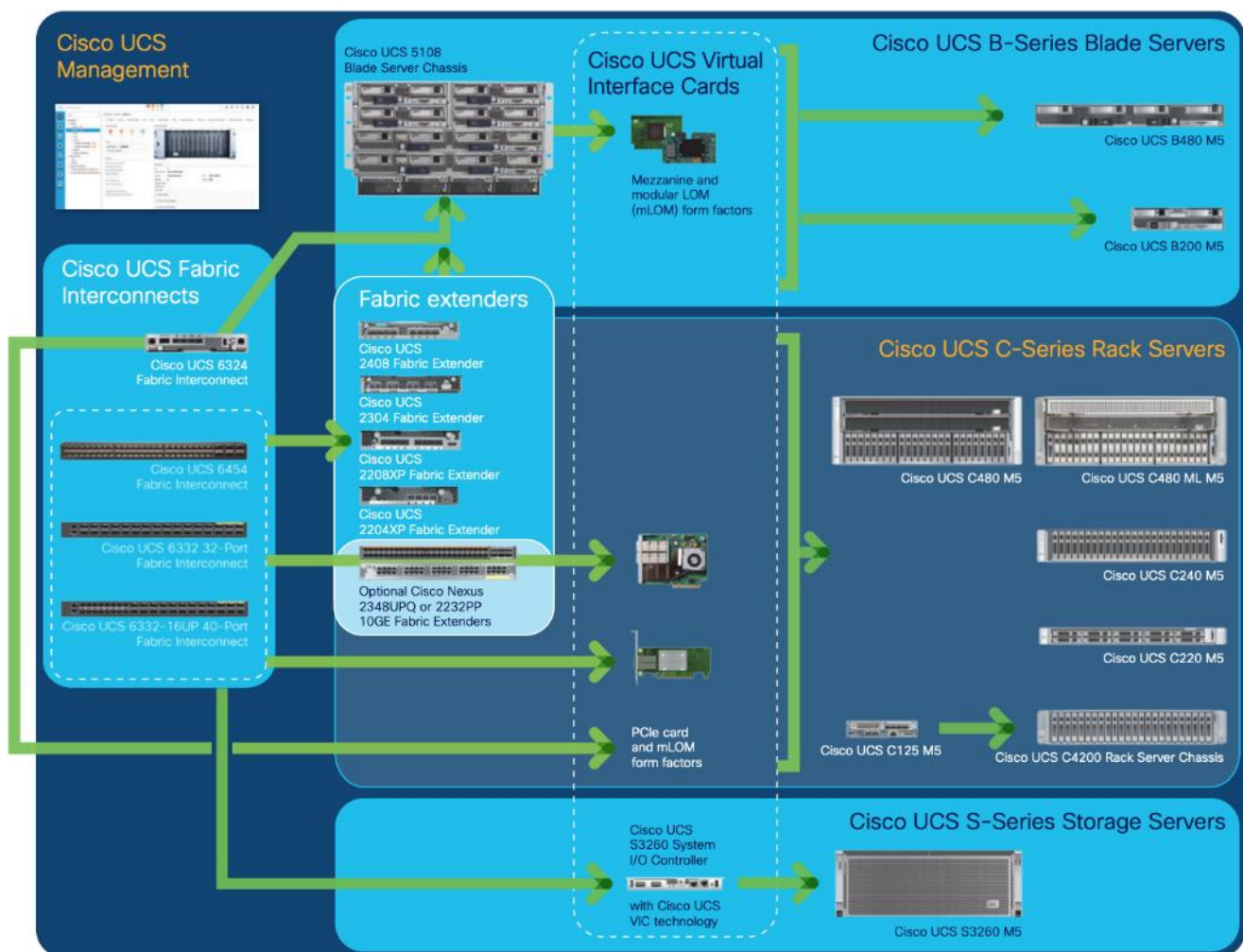
Cisco UCS Integrated Infrastructure for Big Data and Analytics

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

Cisco Unified Computing System™ (Cisco UCS®) is a next-generation data center platform that unites computing, networking, storage access, and virtualization resources into a cohesive system designed to reduce Total Cost of Ownership (TCO) and increase business agility. The system integrates a low-latency, lossless 10/25/40/100 Gigabit Ethernet unified network fabric with enterprise-class, x86-architecture servers. The system is an integrated, scalable, multi-chassis platform in which all resources participate in a unified management domain ([Figure 9](#)).

Figure 9. Cisco UCS Component Hierarchy



Cisco Intersight

Cisco Intersight is Cisco's systems management platform that delivers intuitive computing through cloud-powered intelligence. This platform offers a more intelligent level of management that enables IT organizations to analyze, simplify, and automate their environments in ways that were not possible with prior generations of tools. This capability empowers organizations to achieve significant savings in Total Cost of Ownership (TCO) and to deliver applications faster, so they can support new business initiatives.

Cisco Intersight is a Software as a Service (SaaS) infrastructure management which provides a single pane of glass management of CDIP infrastructure in the data center. Cisco Intersight scales easily, and frequent updates are implemented without impact to operations. Cisco Intersight Essentials enables customers to centralize configuration management through a unified policy engine, determine compliance with the Cisco UCS Hardware Compatibility List (HCL), and initiate firmware updates. Enhanced capabilities and tight integration with Cisco TAC enables more efficient support. Cisco Intersight automates uploading files to speed troubleshooting. The Intersight recommendation engine provides actionable intelligence for IT operations management. The insights are driven by expert systems and best practices from Cisco.

Cisco Intersight offers flexible deployment either as Software as a Service (SaaS) on Intersight.com or running on your premises with the Cisco Intersight virtual appliance. The virtual appliance provides users with the benefits of Cisco Intersight while allowing more flexibility for those with additional data locality and security requirements.

Figure 10. Cisco Intersight



Cisco Intersight has the following:

- Connected TAC
- Security Advisories
- Hardware Compatibility List (HCL) and much more

To learn more about all the features of Intersight go to: <https://www.cisco.com/c/en/us/products/servers-unified-computing/intersight/index.html>

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 applica-

tion-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

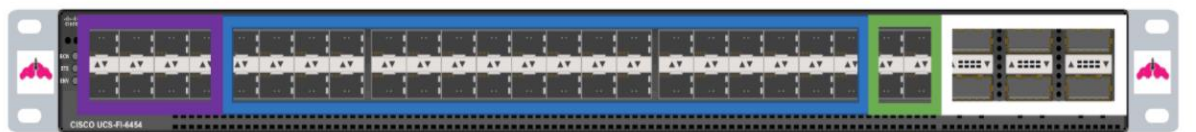
- Supports Cisco UCS B-Series Blade and Cisco UCS C-Series Rack Servers, the Cisco UCS C3260 storage server, Cisco UCS Mini, and the Cisco HyperFlex hyperconverged infrastructure.
- Programmatically controls server, network, and storage resources, with a unified, policy-driven management, so they can 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 6400 Series Fabric Interconnect

The Cisco UCS 6454 provides the management and communication backbone for the Cisco UCS B-Series Blade Servers, Cisco UCS 5108 B-Series Server Chassis, Cisco UCS Managed C-Series Rack Servers, and Cisco UCS S-Series Storage Servers, providing both network connectivity and management capabilities for the system ([Figure 12](#)).

From a networking perspective, the Cisco UCS 6454 uses a cut-through architecture, supporting deterministic, low-latency, line-rate 10/25/40/100 Gigabit Ethernet ports, switching capacity of 3.82 Tbps, and 160 Gbps bandwidth between FI 6454 and IOM 2208 per 5108 blade chassis, independent of packet size and enabled services. The product family supports Cisco® low-latency, lossless 10/25/40/100 Gigabit Ethernet unified network fabric capabilities, which increase the reliability, efficiency, and scalability of Ethernet networks. The Fabric Interconnect supports multiple traffic classes over a lossless Ethernet fabric from the server through the Fabric Interconnect. Significant TCO savings come from an FCoE optimized server design in which Network Interface Cards (NICs), Host Bus Adapters (HBAs), cables, and switches can be consolidated.

Figure 11. Cisco UCS 6454 Fabric Interconnect



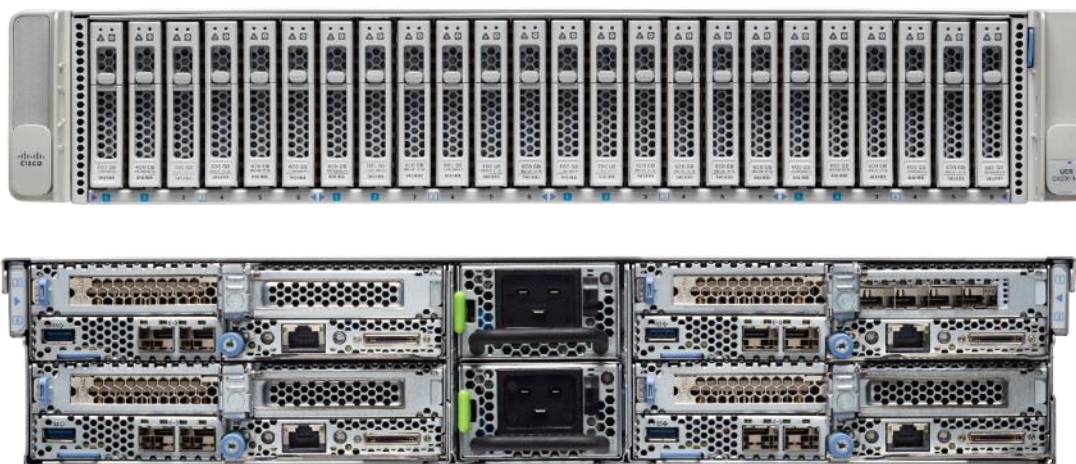
Ethernet (10/25) or Fibre Channel (8/16/32) Ethernet/FCoE Ports (10/25) Uplink Ports (40/100) Ethernet/FCoE Ports (1/10/25)

Cisco UCS C4200 Series Chassis

The Cisco UCS C4200 Series Chassis is a modular, density optimized rack-server chassis that supports:

- Up to four Cisco UCS C125 M5 Rack Server Nodes and up to 256 cores per chassis with AMD EPYC processors: It is excellent for environments requiring dense computing form factors and high core densities, such as scale-out, computing-intensive, general service provider, and BareMetal applications.
- 24 small-form-factor (SFF) drives: The drive bays are allocated so that each rack server node has access to six SAS and SATA HDD or solid-state disks (SSDs) or up to four SSDs and two Non-Volatile Memory Express (NVMe) drives.

Figure 12. Cisco UCS C4200 Series Chassis



Cisco UCS C125 M5 Rack Server Node

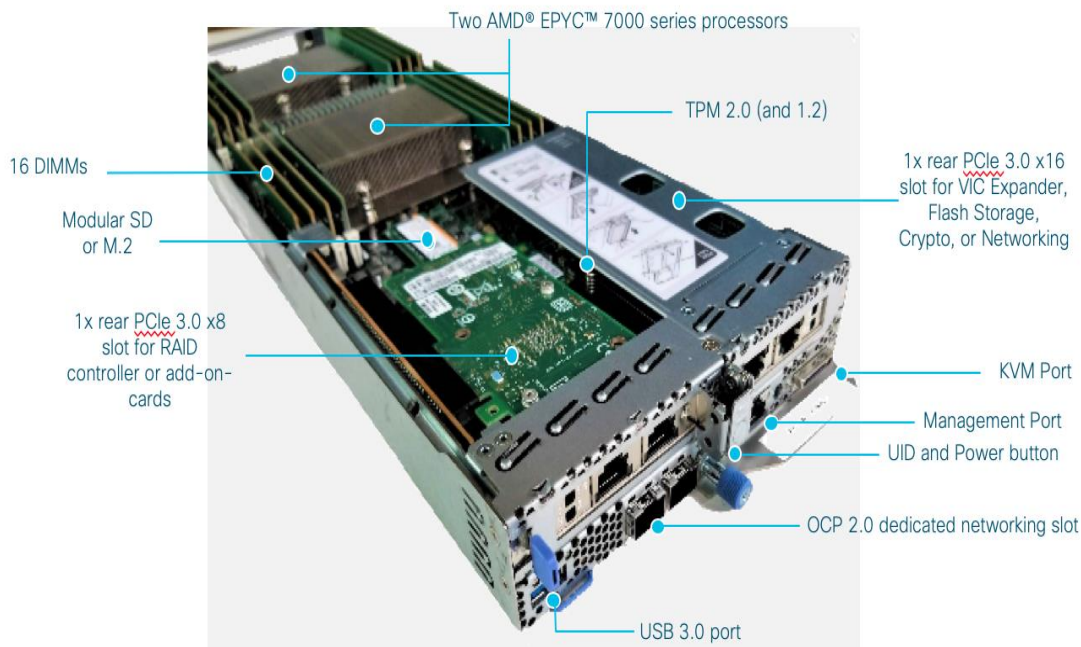
Cisco UCS C125 M5 Rack Server Node has the greatest number of cores commercially available in a multi-node system. It offers 2 sockets per node and from 8 to 32 cores per processor, with the support of the AMD EPYC 7000 series processors, 16 DIMM slots for 2666-MHz DDR4 DIMMs, and capacity points of up to 128 GB per slot for a total of 2 TB per socket. It offers up to 2 half-height and half-length PCI Express (PCIe) 3.0 slots and an optional M.2 SSD module. The C125 supports either SAS RAID through a PCIe 12-Gbps SAS storage controller card or SATA directly from the AMD EPYC processor. The node also includes a dedicated internal LAN mezzanine slot based on the Open Compute Project (OCP) 2.0 standard, supporting networking speeds of up to 100 Gbps. Additionally, a fourth-generation Cisco PCIe virtual interface card (VIC) can be added in the x16 PCIe 3.0 slot.

Figure 13. Cisco U125 Rack Server Node



AMD EPYC 7000 Series Processor

Designed from the foundation for a new generation of solutions, AMD EPYC server processors implement a philosophy of choice without restriction. Choose the number of cores that meet your needs without sacrificing key features such as memory and I/O. Each EPYC processor can have from 8 to 32 cores with access to an exceptional amount of I/O and memory regardless of the number of cores in use. The processors include 128 PCIe Generation 3 lanes and support for up to 2 TB of high-speed memory per socket. The innovative AMD EPYC architecture provides outstanding performance. I/O intensive workloads can use the plentiful I/O bandwidth with the right number of cores, helping organizations avoid overpaying for unneeded power. And computing-intensive workloads can make use of fully loaded core counts, dual sockets, and plenty of memory.



Samsung enterprise SSDs

Samsung enterprise SSDs offer a simple yet versatile and comprehensive selection of enterprise data storage and caching options suitable for nearly any application. They are engineered to provide high throughput and a consistent rate of I/O operations per second (IOPS). They offer large capacities of up to 3.84 TB, making them well suited for enterprise storage solutions for businesses seeking to enhance performance and cost effectiveness by upgrading their current servers or workstations from hard-disk drives (HDDs). These newer SSDs dramatically reduce latency and improve the IOPS rate.

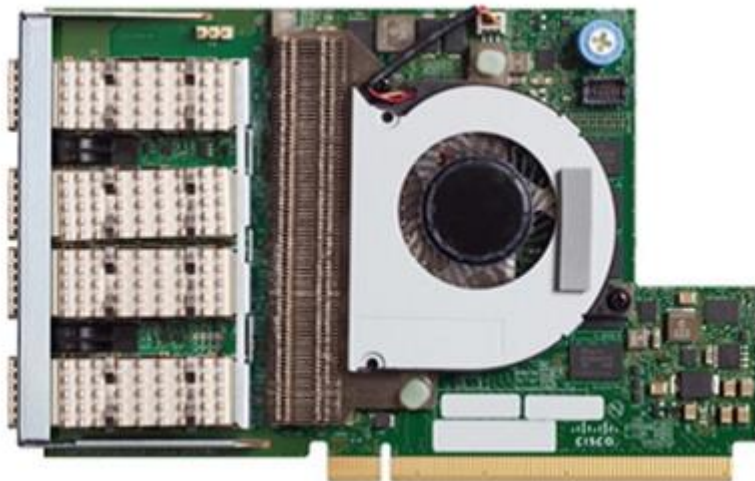
Cisco UCS Virtual Interface Cards (VICs)

This section describes the available Cisco VICs.

Cisco UCS VIC 1457

The Cisco UCS VIC 1457 ([Figure 14](#)) is a quad-port Small Form-Factor Pluggable (SFP28) mLOM card designed for the M5 generation of Cisco UCS C-Series Rack Servers. The card supports 10/25-Gbps Ethernet or FCoE. The card can present PCIe standards-compliant interfaces to the host, and these can be dynamically configured as either NICs or HBAs.

Figure 14. Cisco UCS VIC 1457



Cloudera Data Platform (CDP) Private Cloud Base

CDP Private Cloud Base is an on-premise version of Cloudera Data Platform.

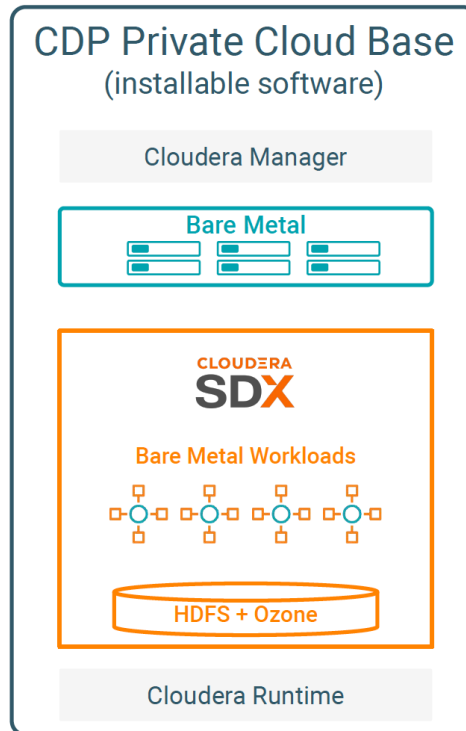
This new product combines the best of Cloudera Enterprise Data Hub and Hortonworks Data Platform Enterprise along with new features and enhancements across the stack. This unified distribution is a scalable and customizable platform where you can securely run many types of workloads.

CDP PVC Base supports a variety of hybrid solutions where compute tasks are separated from data storage and where data can be accessed from remote clusters, including workloads created using CDP Private Cloud Experiences. This hybrid approach provides a foundation for containerized applications by managing storage, table schema, authentication, authorization, and governance.

CDP PVC Base is comprised of a variety of components such as Apache HDFS, Apache Hive 3, Apache HBase, and Apache Impala, along with many other components for specialized workloads. You can select any combination of these services to create clusters that address your business requirements and workloads. Several pre-configured packages of services are also available for common workloads.

Figure 15. Cloudera Data Platform Private Cloud Base Overview

- Similar architecture to CDH and HDP
- Formerly known as CDP Data Center



CDP PVC - Base

Take the two best open-source data analytics platforms, fuse them together, add new capabilities, and we get CDP Data Center.



+ New Features

CDP Data Center

- ✓ First CDP product on-premises
- ✓ 20+ components
- ✓ Highly customizable

Apache Ozone

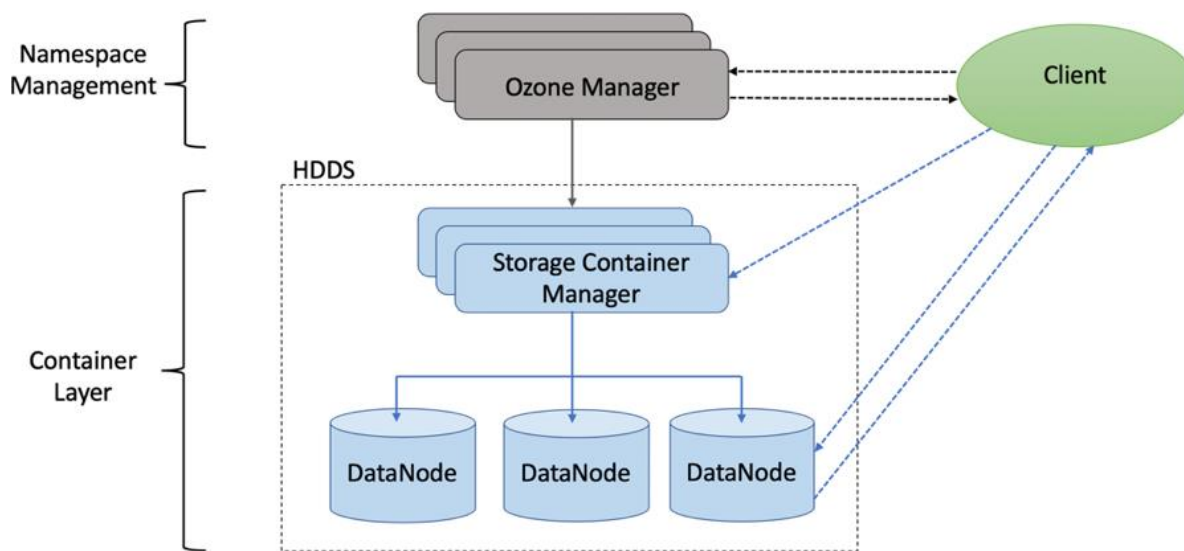
Ozone is a scalable, redundant, and distributed object store optimized for big data workloads. Apart from scaling to billions of objects of varying sizes, Ozone can function effectively in containerized environments such as Kubernetes and YARN.

Ozone consists of three important storage elements: volumes, buckets, and keys. Each key is part of a bucket, which, in turn, belongs to a volume. Only an administrator can create volumes. Depending on their requirements, users can create buckets in volumes. Ozone stores data as keys inside these buckets.

When a key is written to Ozone, the associated data is stored on the DataNodes in chunks called blocks. Therefore, each key is associated with one or more blocks. Within the DataNodes, a series of unrelated blocks is stored in a container, allowing many blocks to be managed as a single entity.

Ozone separates management of namespaces and storage, helping it to scale effectively. Ozone Manager manages the namespaces while Storage Container Manager handles the containers.

Figure 16. Basic Architecture for Ozone



Ozone is available for technical preview and considered to be under development. Do not use this component in your production systems.

Red Hat Ansible Automation

This solution uses Red Hat Ansible Automation for all pre and post deployment steps for automating repeatable tasks to maintain consistency.

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 Data Platform Data Center on a 48-node cluster based on Cisco UCS Integrated Infrastructure for Big Data and Analytics. The solution provides the details to configure CDP PVC Base on the infrastructure.

The cluster configuration consists of the following:

- 2 Cisco UCS 6454 Fabric Interconnects
- 9 Cisco UCS C4200 Series Chassis
- 1 Cisco R42610 standard racks
- 2 Vertical Power distribution units (PDUs) (Country Specific)

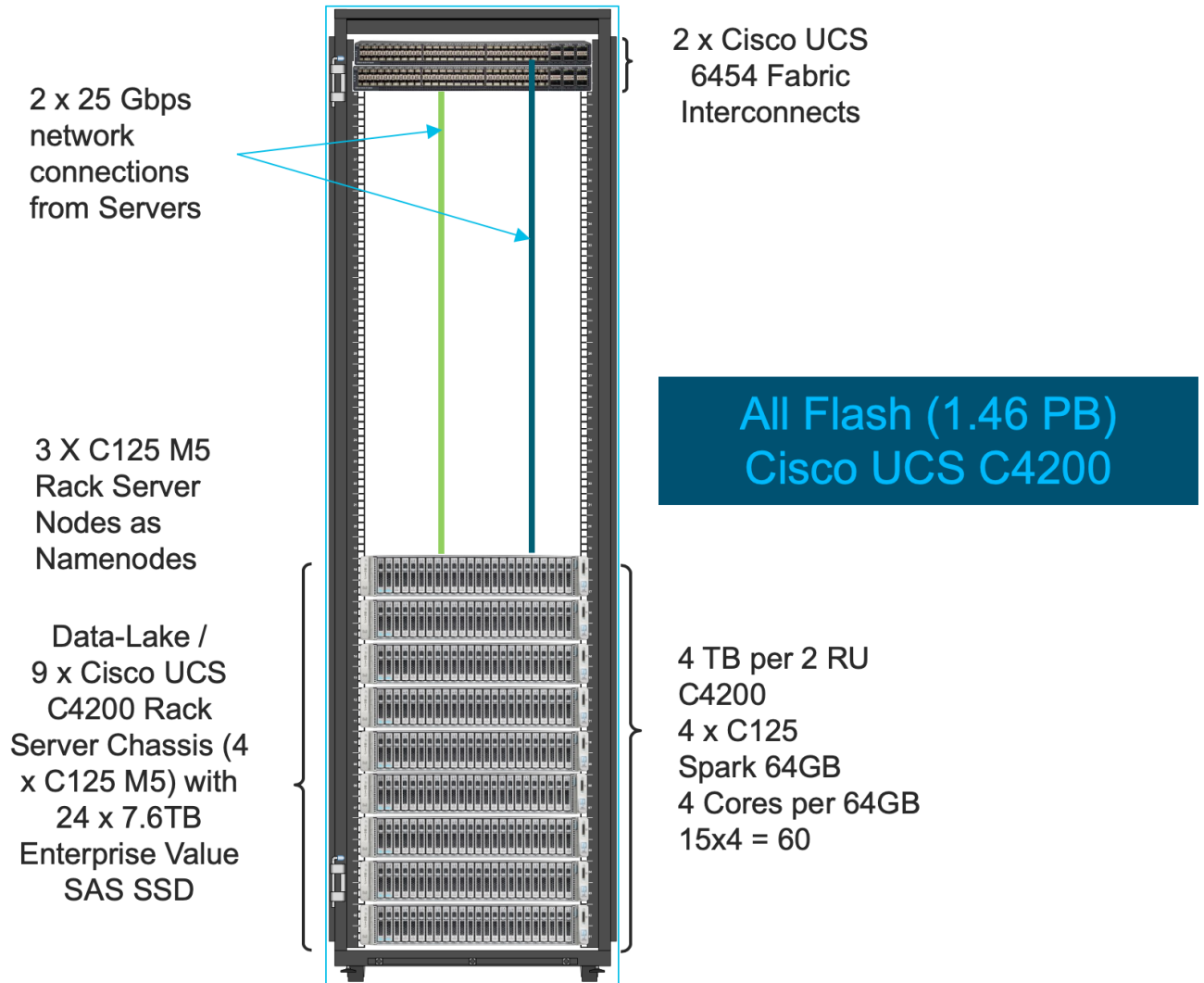
Physical Topology

Each rack consists of two vertical PDUs. The first rack consists of two Cisco UCS 6454 Fabric Interconnects, 48 Cisco UCS C125 M5 Rack Server Node connected to each of the vertical PDUs for redundancy; thereby, ensuring availability during power source failure. [Figure17](#) represents a 25 Gigabit Ethernet link from each server is connected to both Fabric Interconnects.



Please contact your Cisco representative for country-specific information.

Figure 17. Cisco Data Intelligence Platform - 9 Node Configuration with CDP Private Cloud Base



Port Configuration on Fabric Interconnect

Table 2 lists the port configuration on Cisco UCS FI 6454 Fabric Interconnect.

Table 2. Port Configuration on Fabric Interconnect

| Port Type | Port Number |
|-----------|-------------|
| Server | 1-48 |
| Network | 49-54 |

Server Configuration and Cabling for Cisco UCS C125 M5 Rack Server Node

The Cisco UCS C4200 chassis is a modular architecture consisting of the following modules:

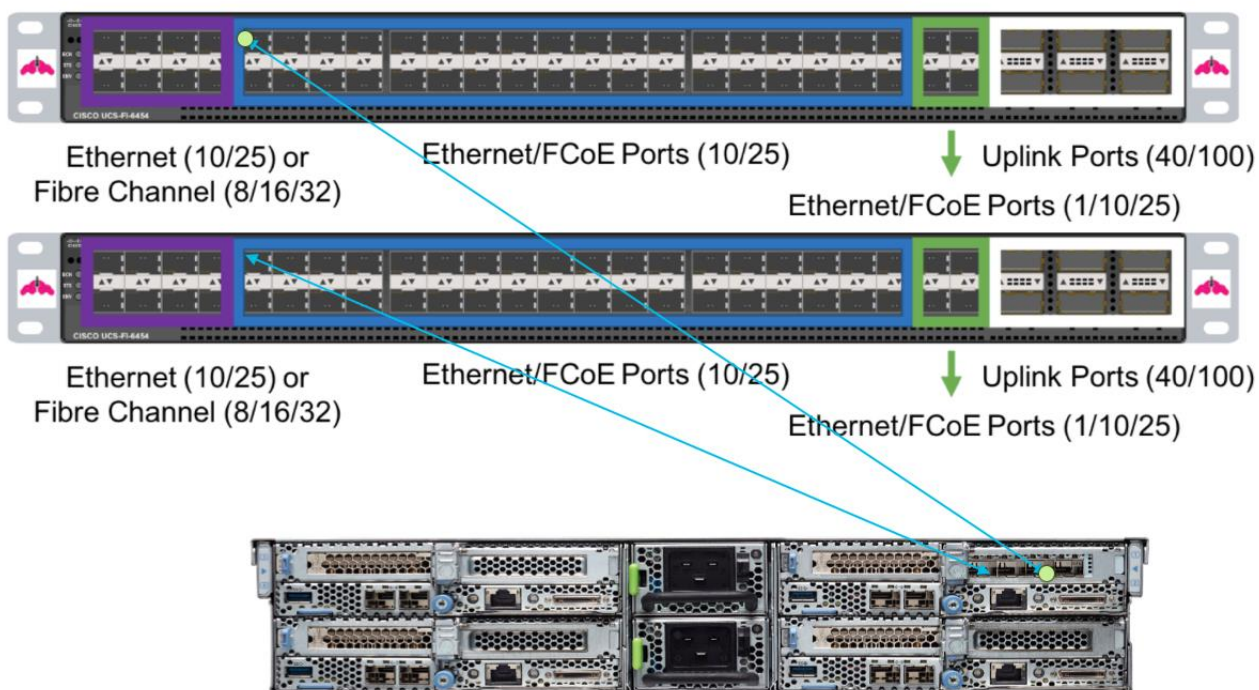
- Base Chassis: 24 SFF drive bays segmented into four groups of six direct attach drives (one group per node slot), four rear slots supporting C125 M5 server node, four redundant hot-pluggable fans, two 2400W AC high-line redundant power supplies, and a rail mounting kit.
- Server Node: Each C125 M5 has two sockets supporting the AMD Epyc 7000 Processors up to 180W TDP, 16 DIMM slots for 2666 MHz DDR4 DIMMs and capacity points up to 64GB, up to 2 half-height/half-length PCI Express (PCIe) 3.0 slots, and optional M.2/SD module. The C125 supports either SAS RAID via a PCIe 12G SAS storage controller card or SATA direct from the AMD Epyc CPU. The node also includes a dedicated internal LAN mezzanine slot based on the OCP 2.0 standard supporting networking speeds up to 100Gbps. Additionally, installation of a 4th generation Cisco PCIe Virtual Interface Card (VIC) can be added in the x16 PCIe 3.0 slot.

[Figure 18](#) illustrates the port connectivity between the Cisco UCS FI 6454 and Cisco UCS C125 M5 Rack Server Nodes. 13 Cisco UCS C125 M5 Rack Server Nodes are installed in this configuration.

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

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

Figure 18. Fabric Topology for Cisco UCS C125 M5 Rack Server Node



Software Distributions and Firmware Versions

The software distributions required versions are listed in [Table 3](#).

Table 3. Software Distribution and Version

| Layer | Component | Version or Release |
|----------|-----------------------------------|--------------------|
| Compute | Cisco UCS C125 M5 | C125 M5.4.1(2a)C |
| Network | Cisco UCS 6454 | UCS 4.1(2a) A |
| | Cisco UCS VIC1455 Firmware | 5.1(2d) |
| Storage | Cisco 12G Modular Raid Controller | 51.10.0-3151 |
| Software | Red Hat Enterprise Linux Server | 7.8 |
| | Cisco UCS Manager | 4.1(2a)A |
| | Cloudera CDP Private Cloud Base | 7.1.3 |
| | Hadoop | 3.1 |
| | Spark | 2.4 |

The latest drivers can be downloaded from the link below:

[https://software.cisco.com/download/home/283862063/type/283853158/release/4.1\(2a\)](https://software.cisco.com/download/home/283862063/type/283853158/release/4.1(2a))

Cisco Intersight

Cisco Intersight provides the following features to assist the IT staff with an ease of operations and administration.

Connected TAC

Connected TAC is an automated transmission of technical support files to the Cisco® Technical Assistance Center (TAC) for accelerated troubleshooting.

Cisco Intersight enables Cisco TAC to automatically generate and upload Tech Support Diagnostic files when a Service Request is opened. If you have devices that are connected to Intersight but not claimed, Cisco TAC can only check the connection status and will not be permitted to generate Tech Support files. When enabled, this feature works in conjunction with the Smart Call Home service and with an appropriate service contract. Devices that are configured with Smart Call Home and claimed in Intersight can use Smart Call Home to open a Service Request and have Intersight collect Tech Support diagnostic files.

Figure 19. Cisco Intersight: Connected TAC

Cisco Intersight + Cisco TAC + Smart Call Home = Proactive resolution



To enable Connected TAC, follow these steps:

1. Log into intersight.com.
2. Click the Servers tab. Select Server > Actions tab. From the drop-down list, select Open TAC Case.
3. Clicking “Open TAC Case” launches the Cisco URL for Support case manager where associated service contracts for Server or Fabric Interconnect is displayed.

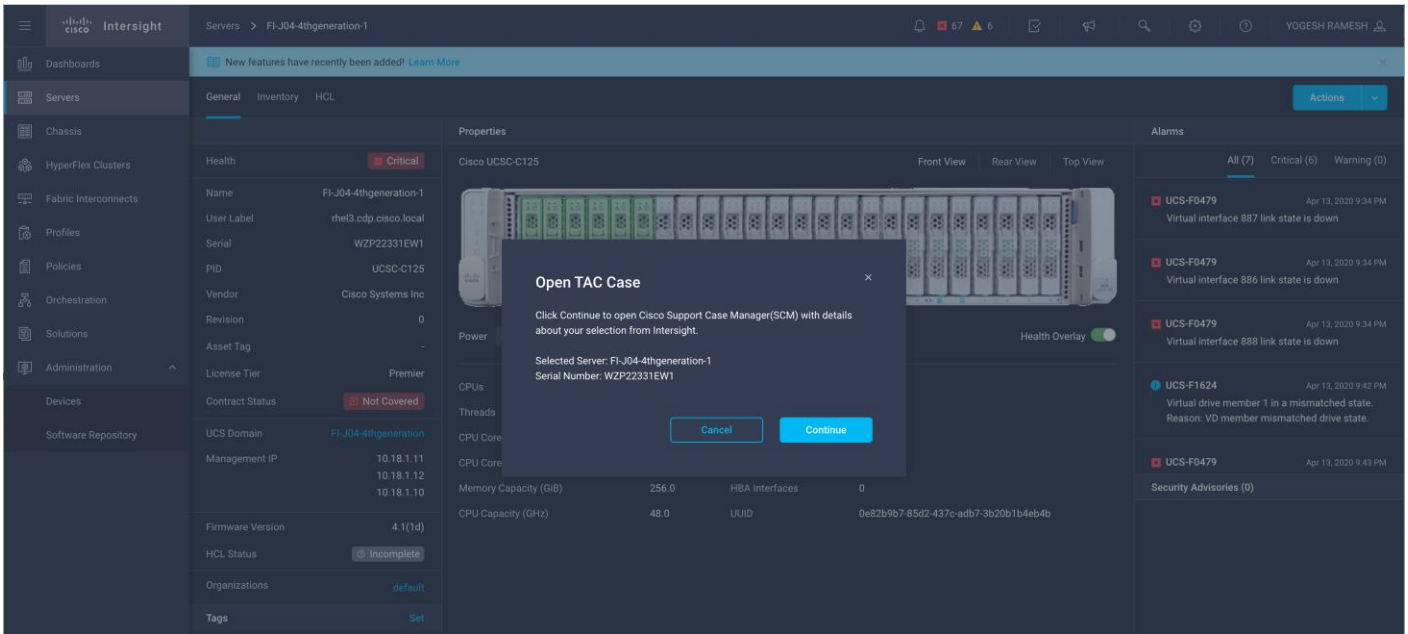
The screenshot displays the Cisco Intersight interface for a server. The main content area shows the following details:

- Health:** Critical
- Properties:** Cisco UCS-C125
- Name:** FI-J04-4thgeneration-1
- User Label:** rhe13.cdp.cisco.local
- Serial:** WZP22331EW1
- PID:** UCSC-C125
- Vendor:** Cisco Systems Inc
- Revision:** 0
- Asset Tag:** -
- License Tier:** Premier
- Contract Status:** Not Covered
- UCS Domain:** FI-J04-4thgeneration
- Management IP:** 10.18.1.11, 10.18.1.12, 10.18.1.10
- Firmware Version:** 4.1(1d)
- HCL Status:** Incomplete
- Organizations:** default
- Tags:** Set

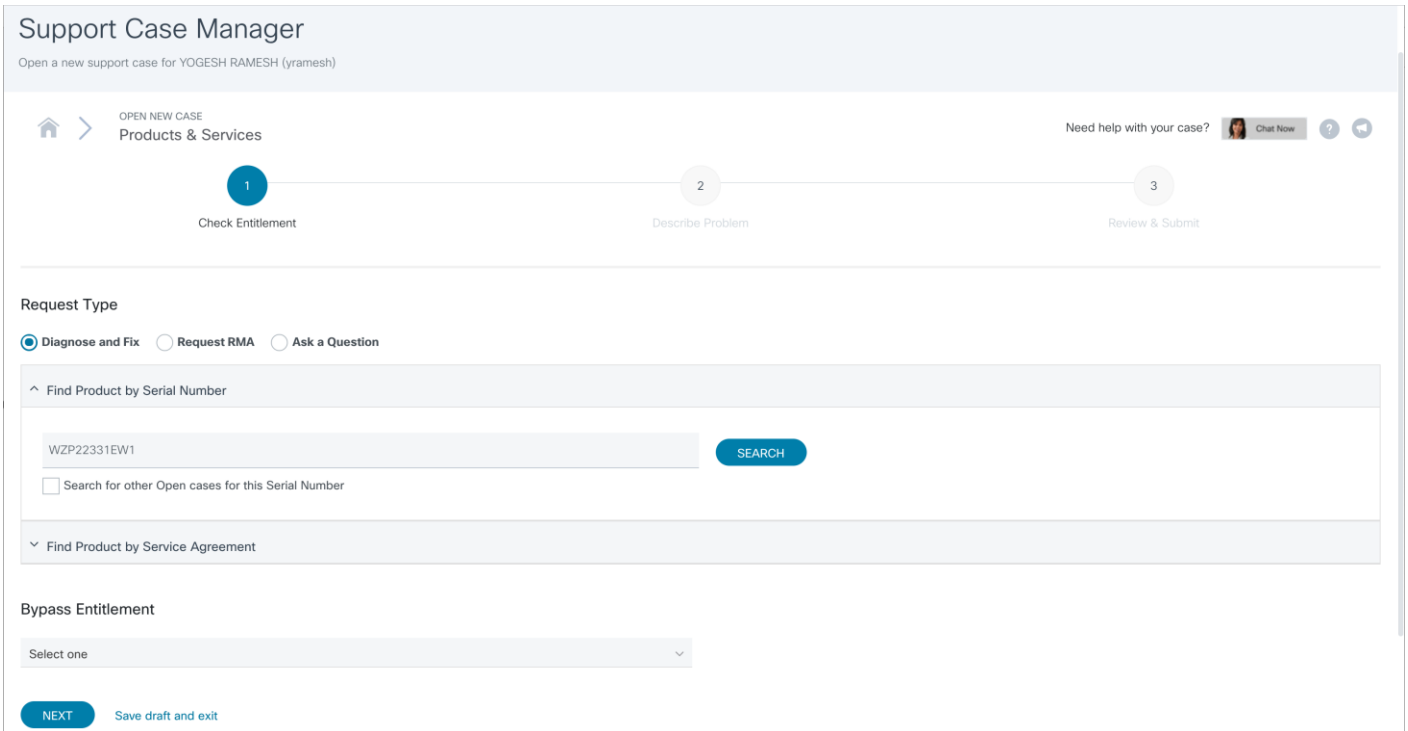
The Alarms section on the right shows the following alerts:

- UCS-F0479: Virtual interface 887 link state is down
- UCS-F0479: Virtual interface 886 link state is down
- UCS-F0479: Virtual interface 888 link state is down
- UCS-F1624: Virtual drive member 1 in a mismatched state. Reason: VD member mismatched drive state.
- UCS-F0479: Virtual interface 886 link state is down

4. Click Continue.



5. Follow the process to the Open TAC Case.



Cisco Intersight Integration for HCL

Cisco Intersight evaluates the compatibility of your Cisco UCS and Cisco HyperFlex systems to check if the hardware and software have been tested and validated by Cisco or Cisco partners. Cisco Intersight reports validation issues after checking the compatibility of the server model, processor, firmware, adapters, operating system, and drivers, and displays the compliance status with the Hardware Compatibility List (HCL).

You can use Cisco UCS Tools, a host utility vSphere Installation Bundle (VIB), or OS Discovery Tool, an open source script to collect OS and driver information to evaluate HCL compliance.

In Intersight, you can view the HCL compliance status in the dashboard (as a widget), the Servers table view, and the Server details page. Below is the server details page.



For more information, go to:

[https://www.intersight.com/help/features#compliance_with_hardware_compatibility_list_\(hcl\)](https://www.intersight.com/help/features#compliance_with_hardware_compatibility_list_(hcl))

Figure 20. Example of HCL Status and Driver Recommendation for RHEL 7.8

The screenshot shows the Cisco Intersight interface for a server named 'FI-J04-4thgeneration-7'. The 'HCL' (Hardware Compatibility List) tab is active, showing the following status:

- Server Hardware Compliance: Validated
- Server Software Compliance: Incomplete
- Missing Operating System information: Learn more at Help Center
- Adapter Compliance: Incomplete

The 'Adapter Compliance' section contains a table with 4 items found:

| Model | Hardware Status | Software Status | Firmware Version | Driver Protocol | Driver Version |
|---------------------------|-----------------|-----------------|------------------|-----------------|----------------|
| FCH SATA Controller [AHC] | - | - | | | |
| UCSC-SAS9460-8i | - | - | 50.8.0-2549 | | |
| FCH SATA Controller [AHC] | - | - | | | |
| UCSC-PCIE-C25Q-04 | - | - | 5.0(3e) | | |

Advisories (PSIRTs)

Cisco Intersight sources critical security advisories from the Cisco Security Advisory service to alert users about the endpoint devices that are impacted by the advisories and deferrals. These alerts are displayed as Advisories in Intersight. The Cisco Security Advisory service identifies and monitors and updates the status of the advisories to provide the latest information on the impacted devices, the severity of the advisory, the impacted products, and any available workarounds. If there are no known workarounds, you can open a support case with Cisco TAC for further assistance. A select list of the security advisories is shown in Intersight under Advisories.

Figure 21. Intersight Dashboard

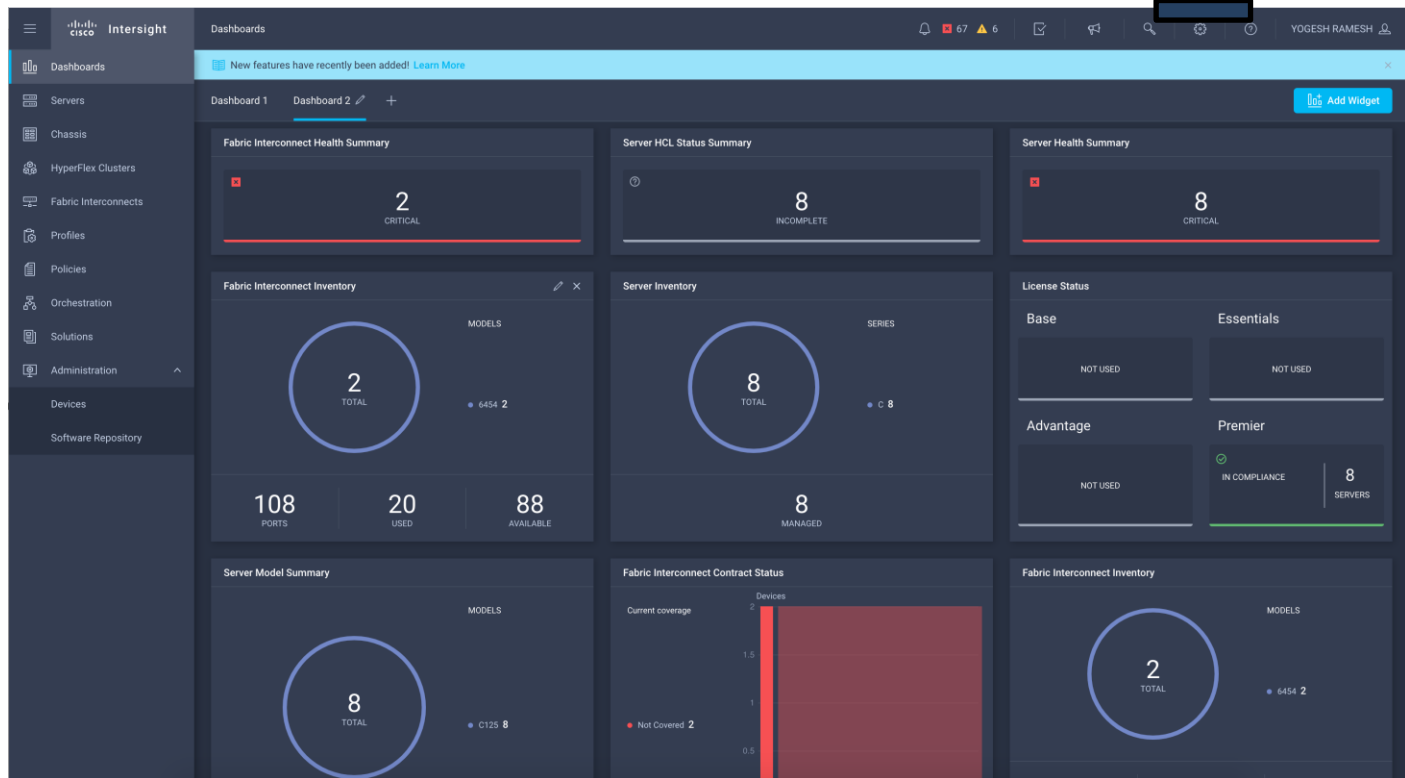
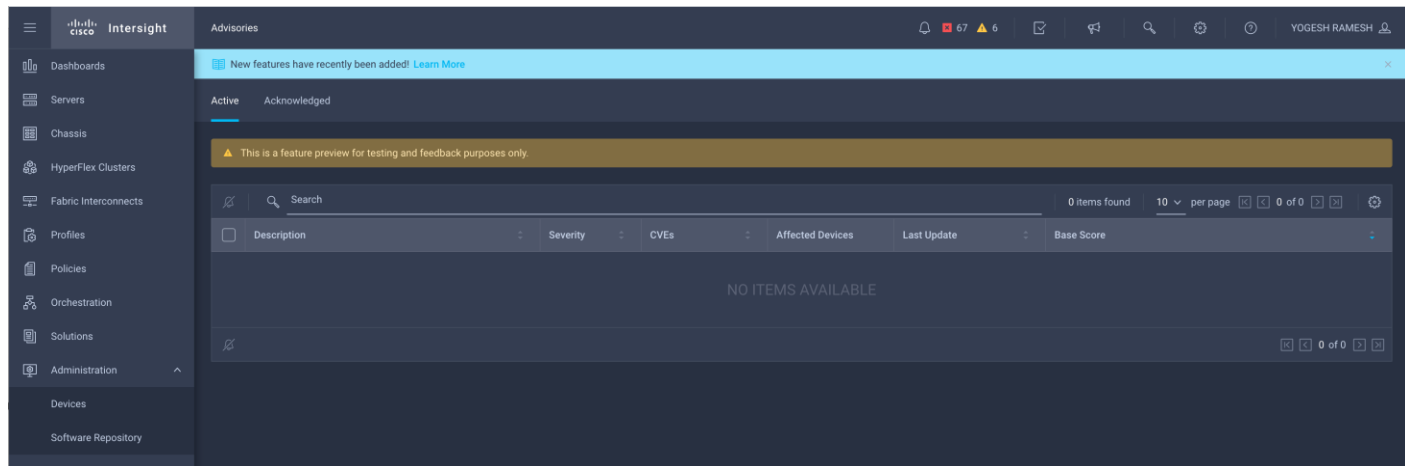


Figure 22. Example: List of PSIRTs Associated with Sample Intersight Account



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 4.1(1b). To upgrade the Cisco UCS Manager software and the Cisco UCS 6454 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:
 - a. The management Ethernet port (mgmt0) is connected to an external hub, switch, or router.
 - b. The L1 ports on both fabric interconnects are directly connected to each other.
 - c. 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 6454 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 6454 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 6454 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 6454 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 into the Cisco UCS Manager.

Upgrade Cisco UCS Manager Software to Version 4.1(2a)A

This document assumes you're using Cisco UCS 4.1(2a). Refer to the [Cisco UCS 4.1 Release](#) (upgrade Cisco UCS Manager software and Cisco UCS 6454 Fabric Interconnect software to version 4.1(2a)). Also, make sure the Cisco UCS C-Series version 4.1(2c) software bundles are installed on the Fabric Interconnects.

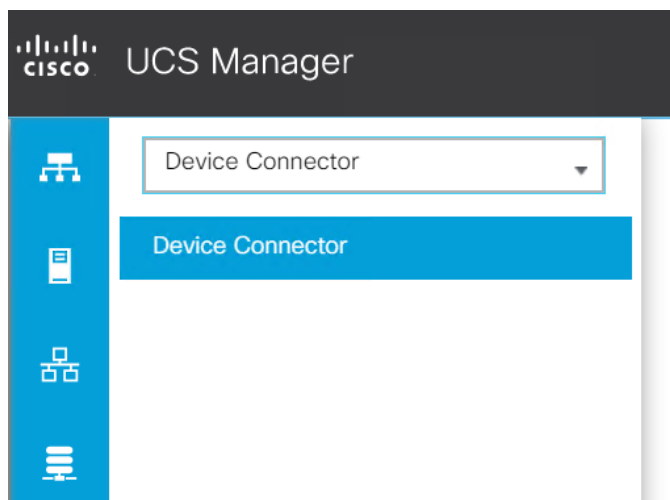


Upgrading Cisco UCS firmware is beyond the scope of this document. However for complete Cisco UCS Install and Upgrade Guides, go to: <https://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-manager/products-installation-guides-list.html>

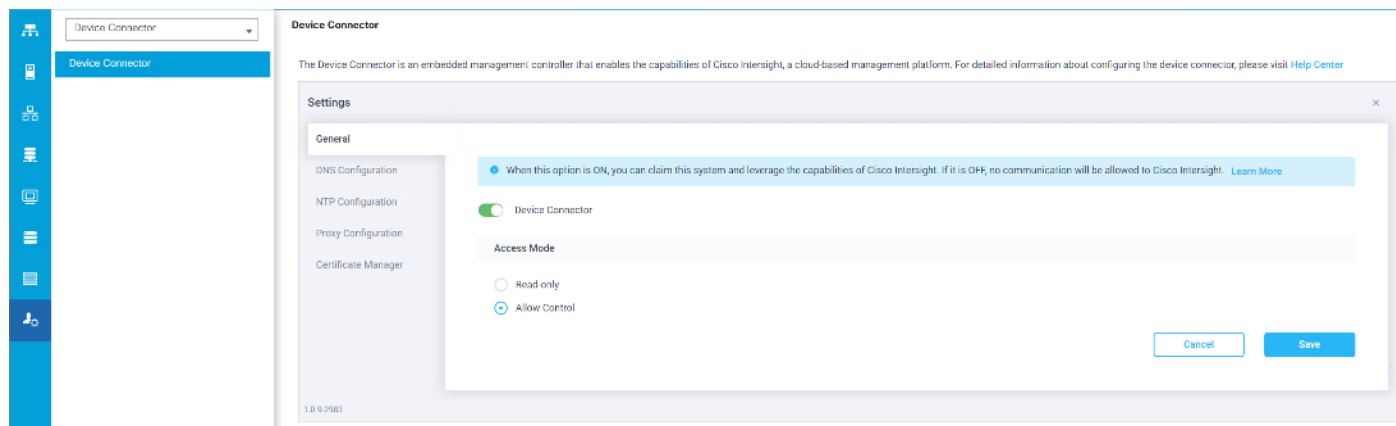
Register Cisco UCS Manager with Intersight

To register UCSM with Intersight, follow these steps:

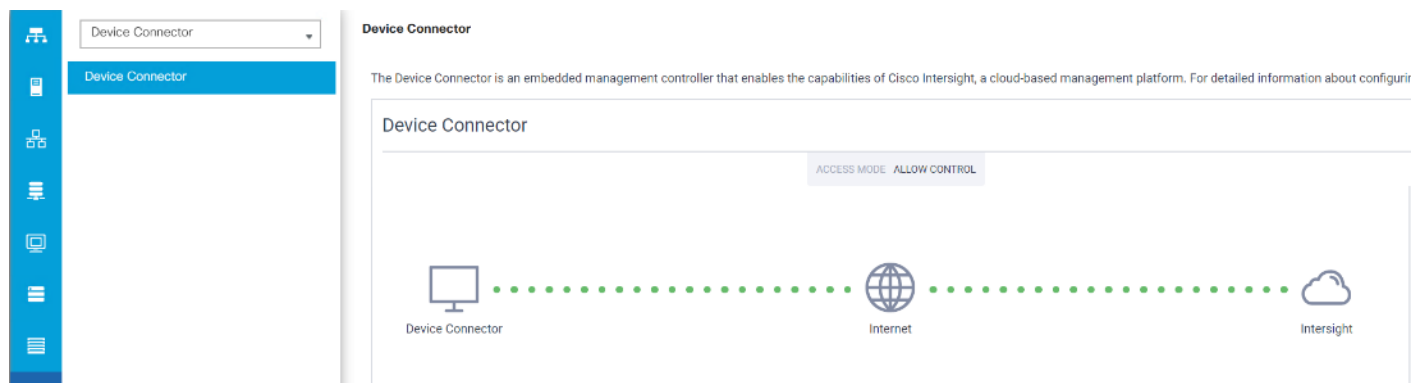
1. Log into the WebUI for Cisco UCS Manager and click the Admin tab. Select Device Connector from the drop-down list. Click Settings.



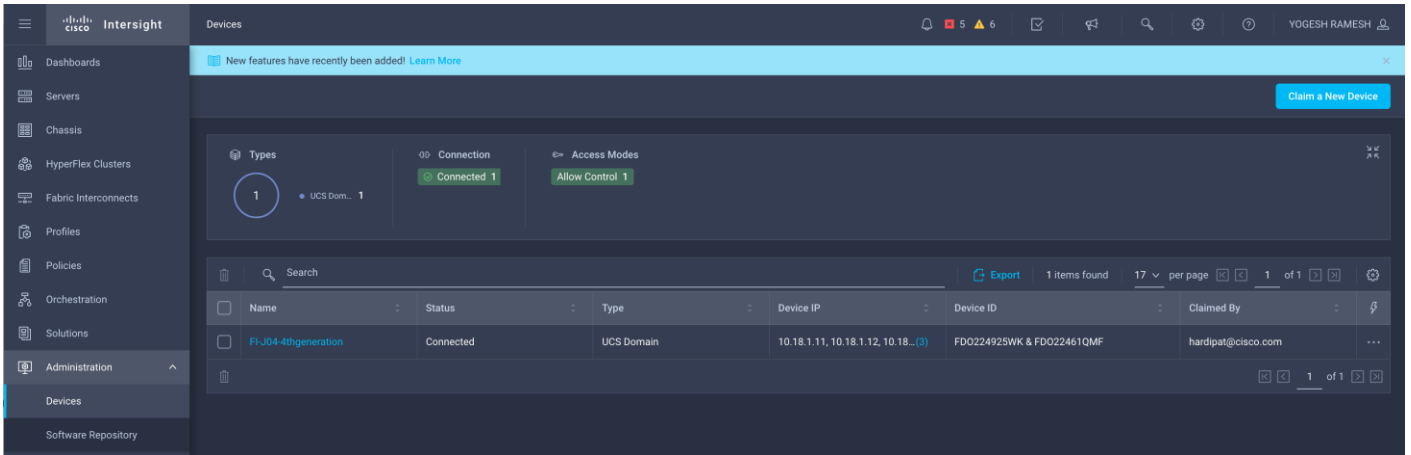
2. Enable Device Connector. Select Allow Control in Access Mode.



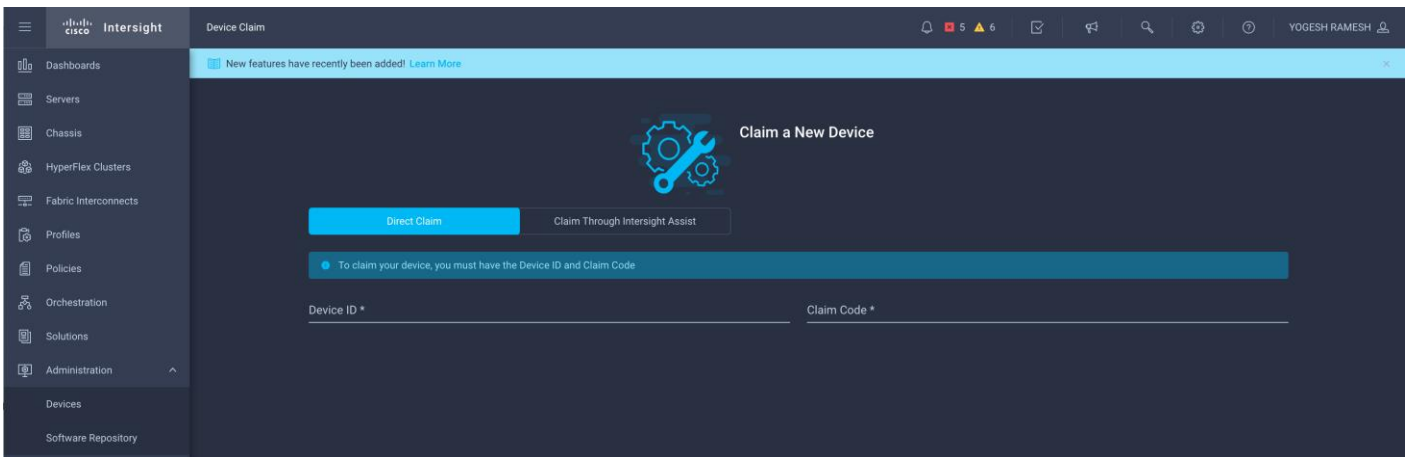
3. Complete the steps for the DNS configuration, NTP Configuration and Proxy Configuration as applicable. Click Save.
4. Make sure UCSM can communicate to Intersight.



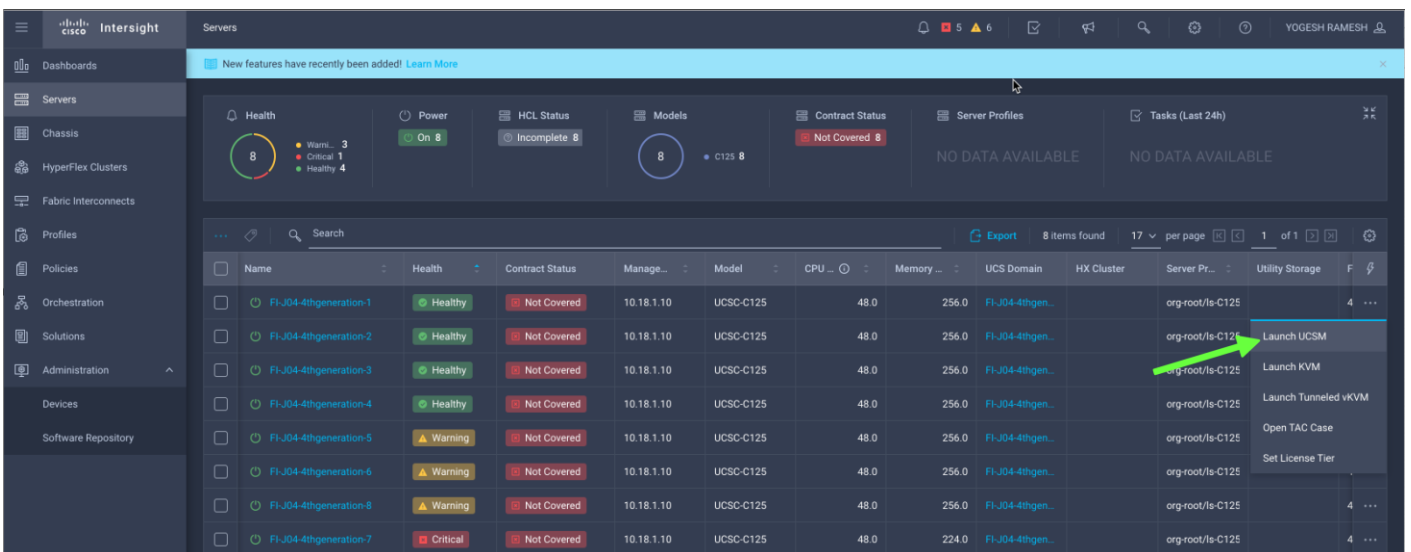
5. Copy Device Claim ID from right side of Device Connector screen.
6. Log into intersight.com
7. Select Devices tab and click Claim a New Device.



8. Enter Device ID and Device Claim Code copied from Cisco UCS Manager. Click Claim.



9. Once Claimed, UCSM can be launched directly from Intersight.



For more information, go to: [Claiming a Device](#)

Configure Cisco UCS Manager through Intersight

To configure Cisco UCS Manager, follow these high-level steps:

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.

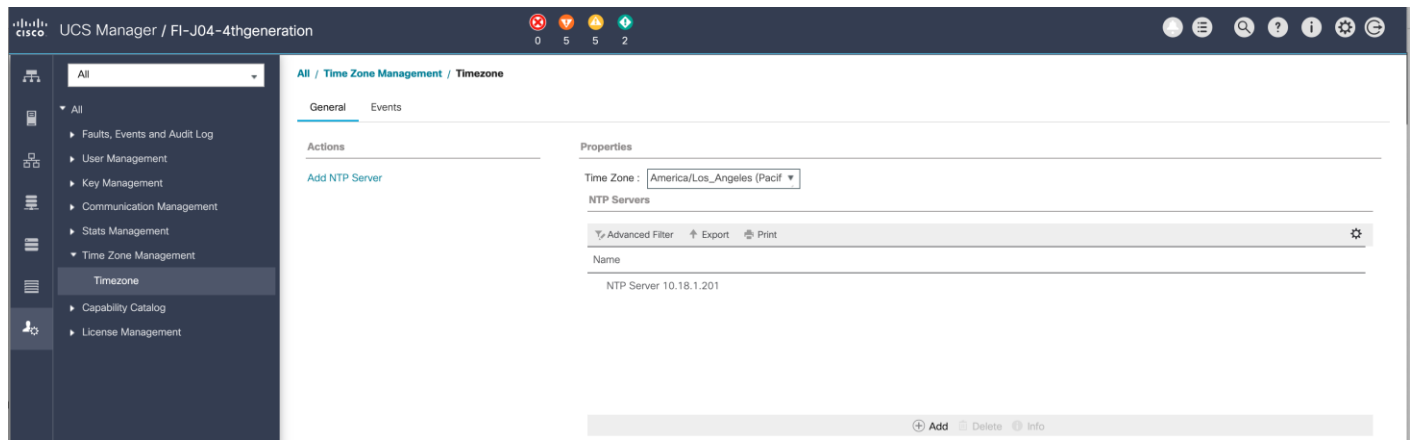
Synchronize Cisco UCSM to NTP

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 23. Synchronize Cisco UCS Manager to NTP



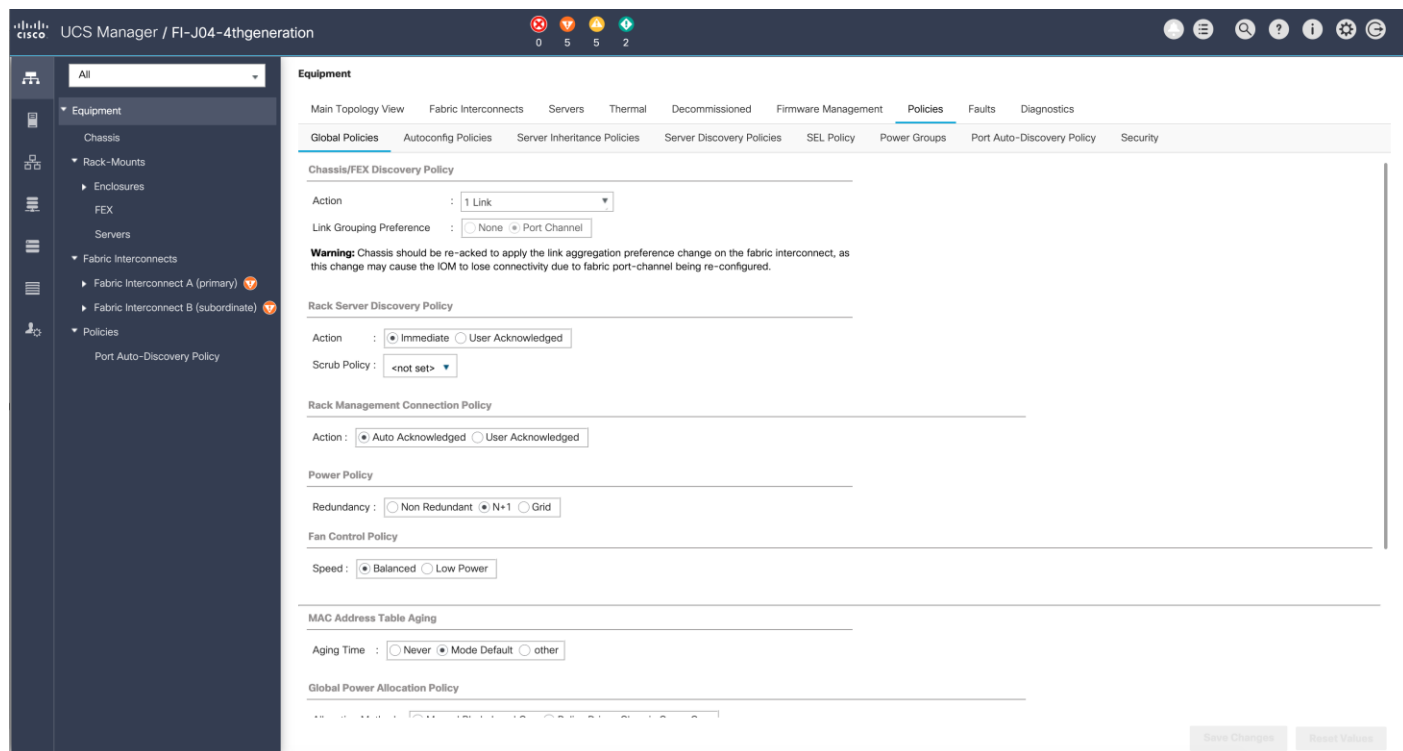
Configure Global Policies

The rack server /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; Configure Global Policy. Go to Equipment > Policies > Global Policies.

Figure 24. Global Policies in UCSM

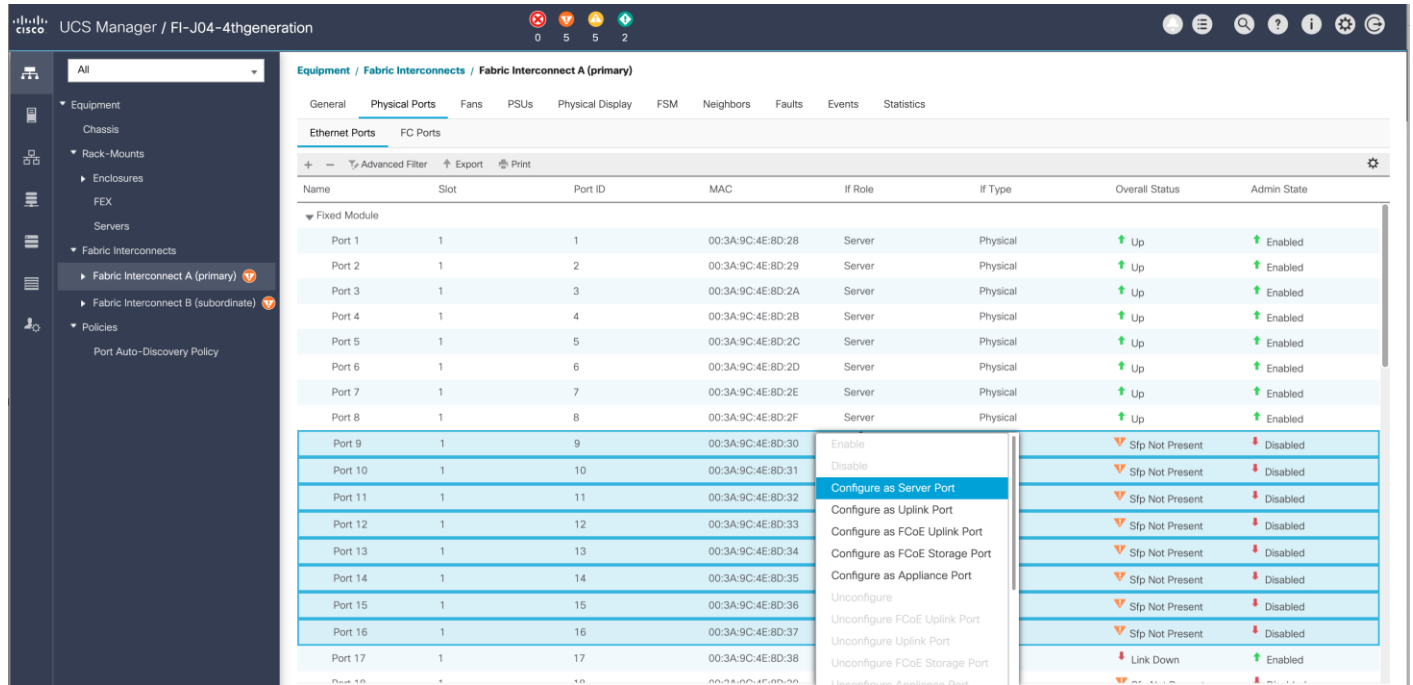


Configure Server Ports

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 1-48) which are connected to the Cisco UCS VIC 1455 on Cisco UCS C125 M5 rack server node.
3. Right-click and select Configure as Server Port.

Figure 25. Configure Server Port on Cisco UCS Manager Fabric Interconnect for Server/Chassis Discovery

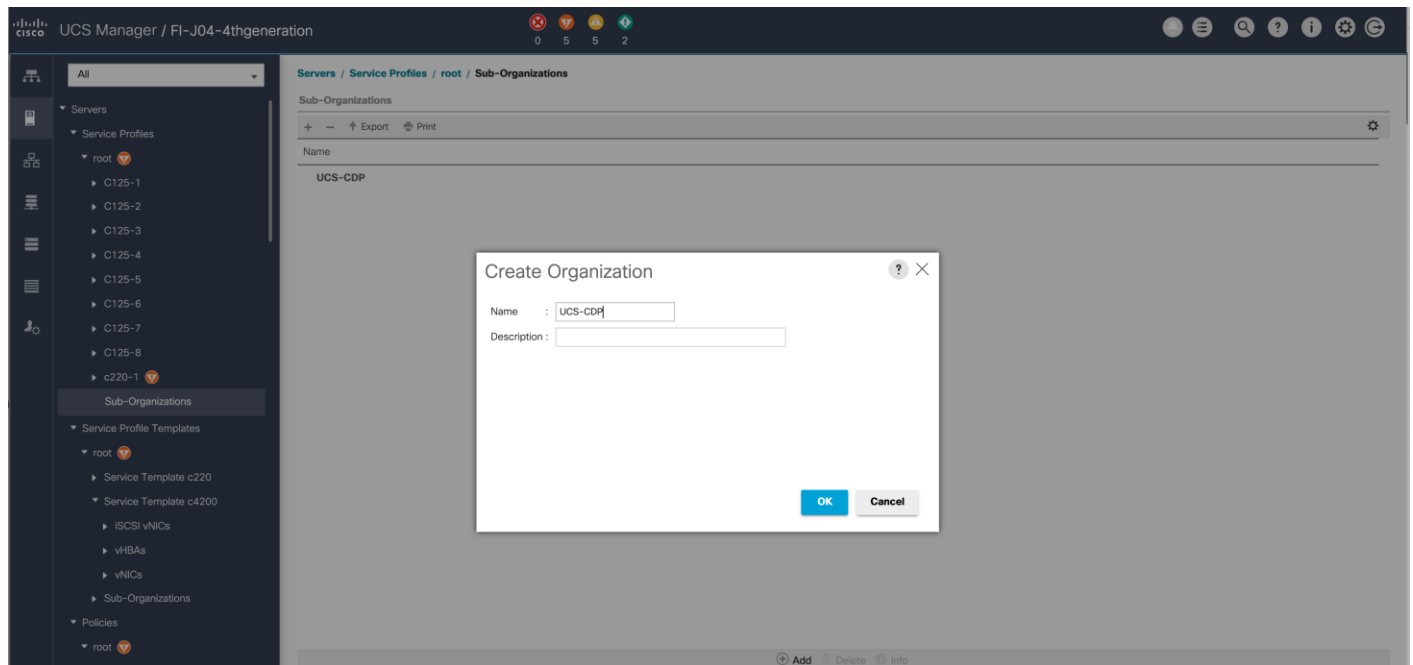



Create New Organization

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 and select Create Sub-Organization.
4. Enter the name of the Organization.
5. Click OK.

Figure 26. Create New Organization under Sub-Organizations



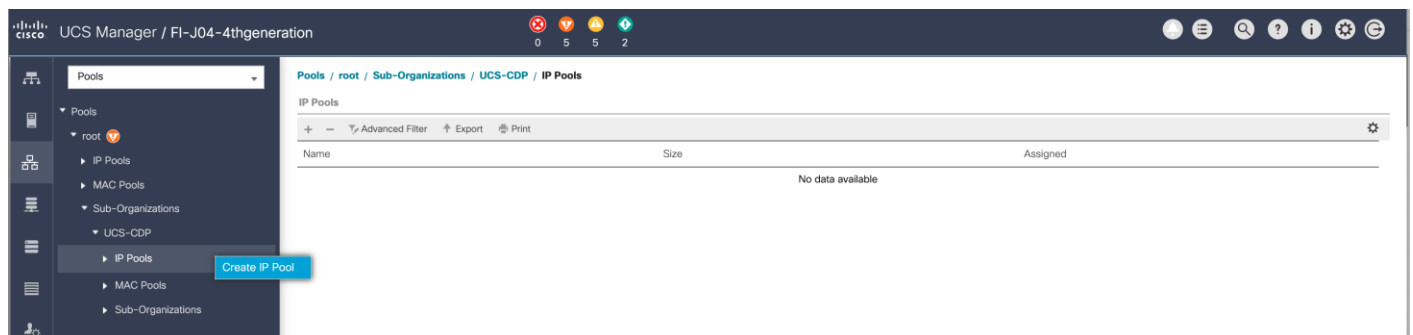
 The Cisco UCS Manager pools and policies required for this solution were created under the newly created “UCS-CDP” Organization.

Configure IP, UUID, Server and MAC Pools

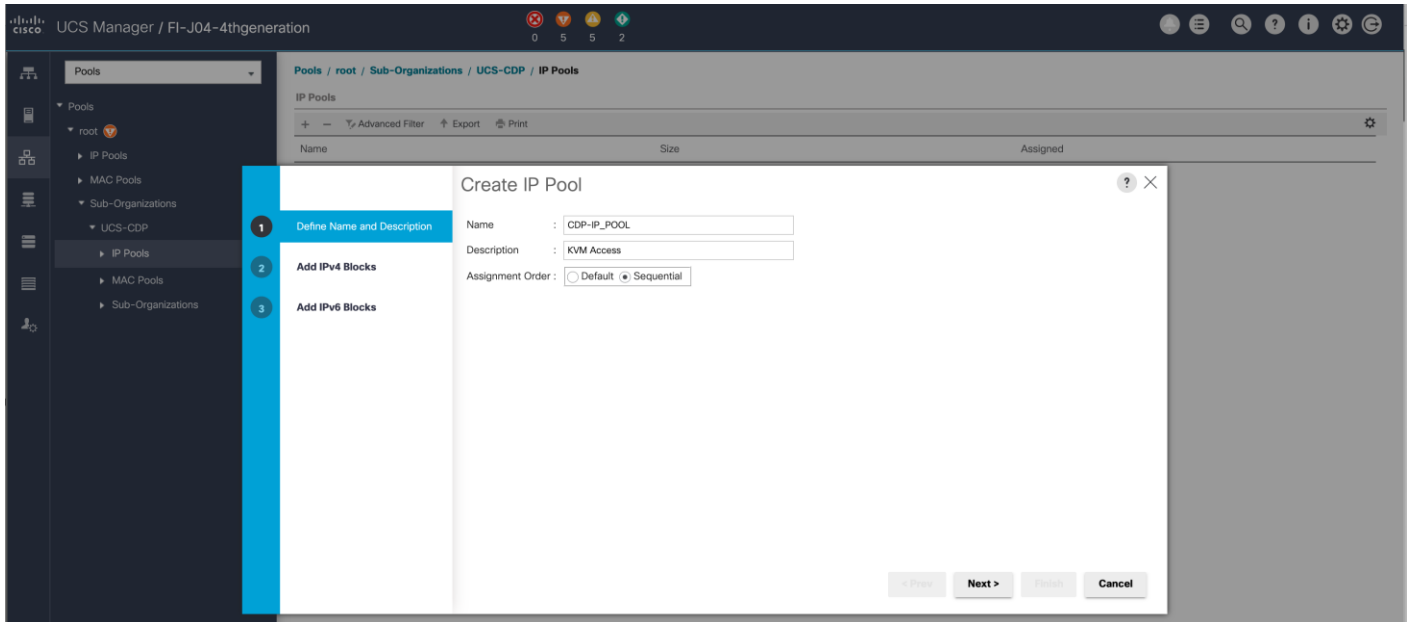
IP Pool Creation

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:

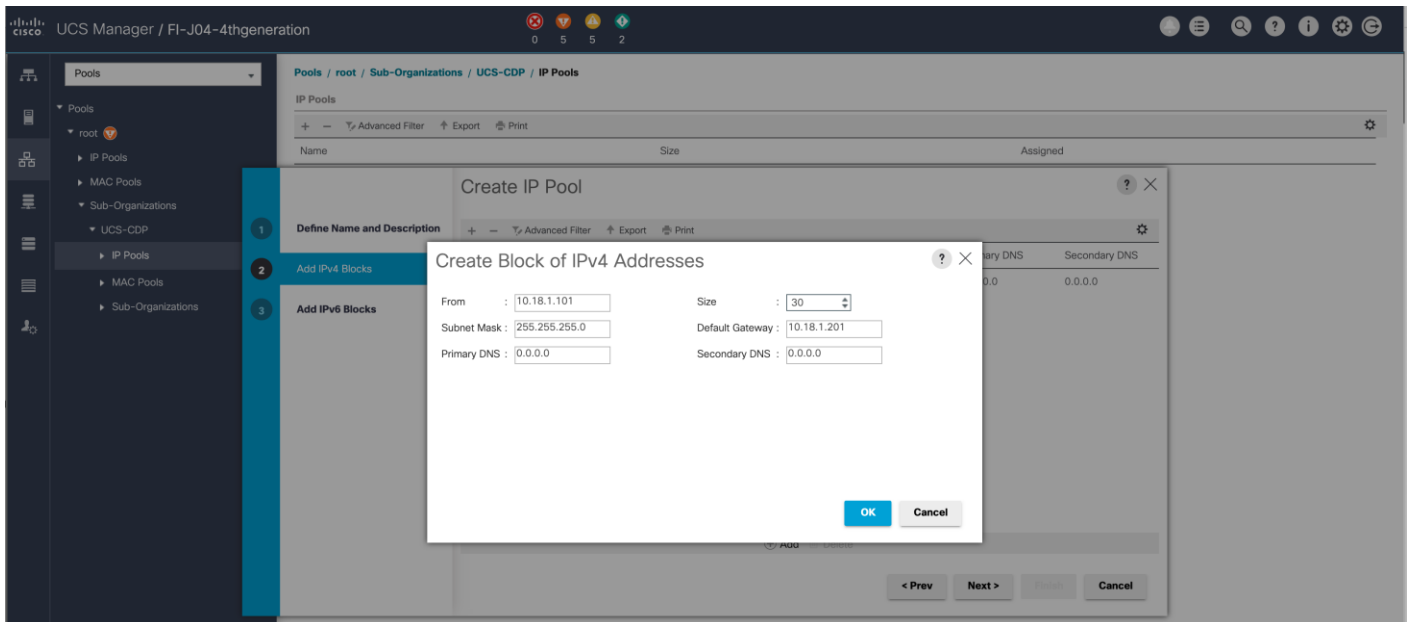
1. In Cisco UCS Manager, in the navigation pane, click the LAN tab.
2. Select Pools > root > Sub-Organizations > UCS-CDP > IP Pools > click Create IP Pool.



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



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 gateway information as shown below.



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-CDP.
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 27. UUID Suffix Pool Creation

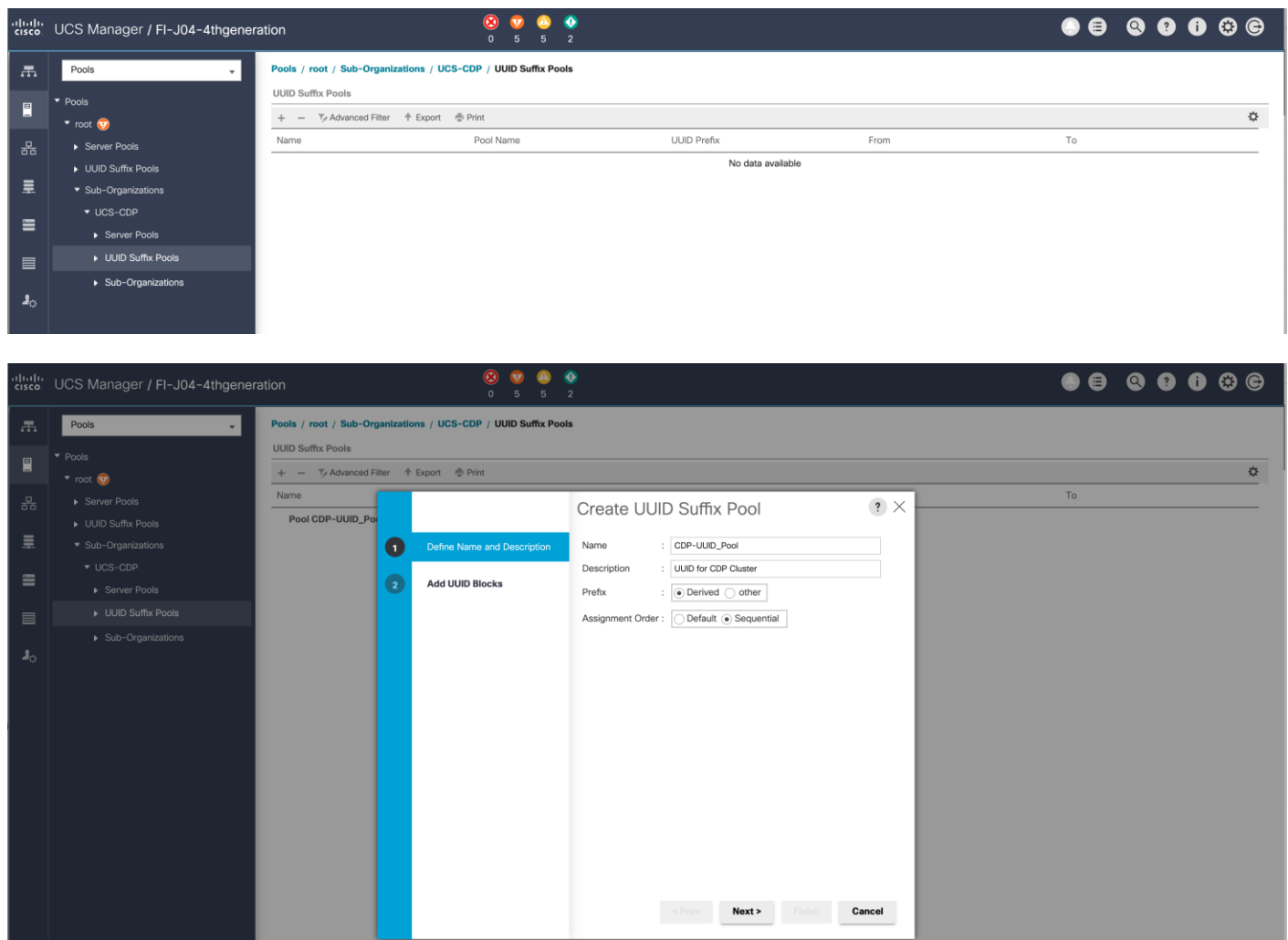
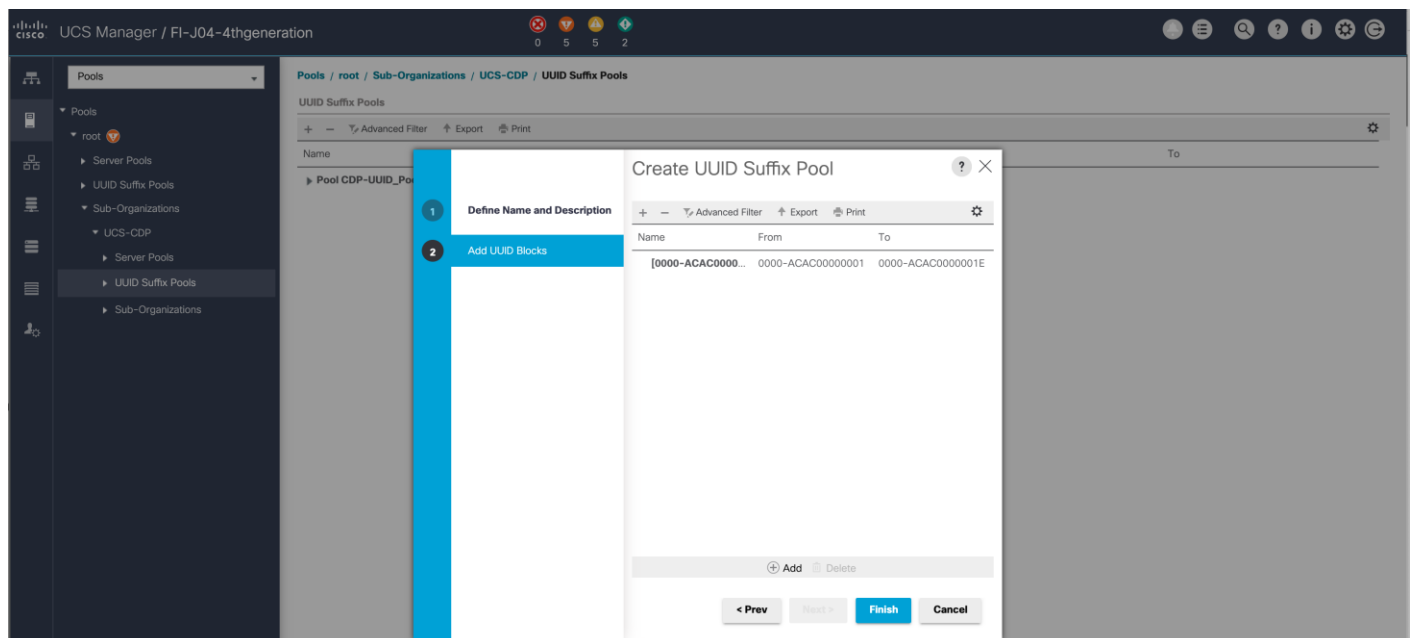
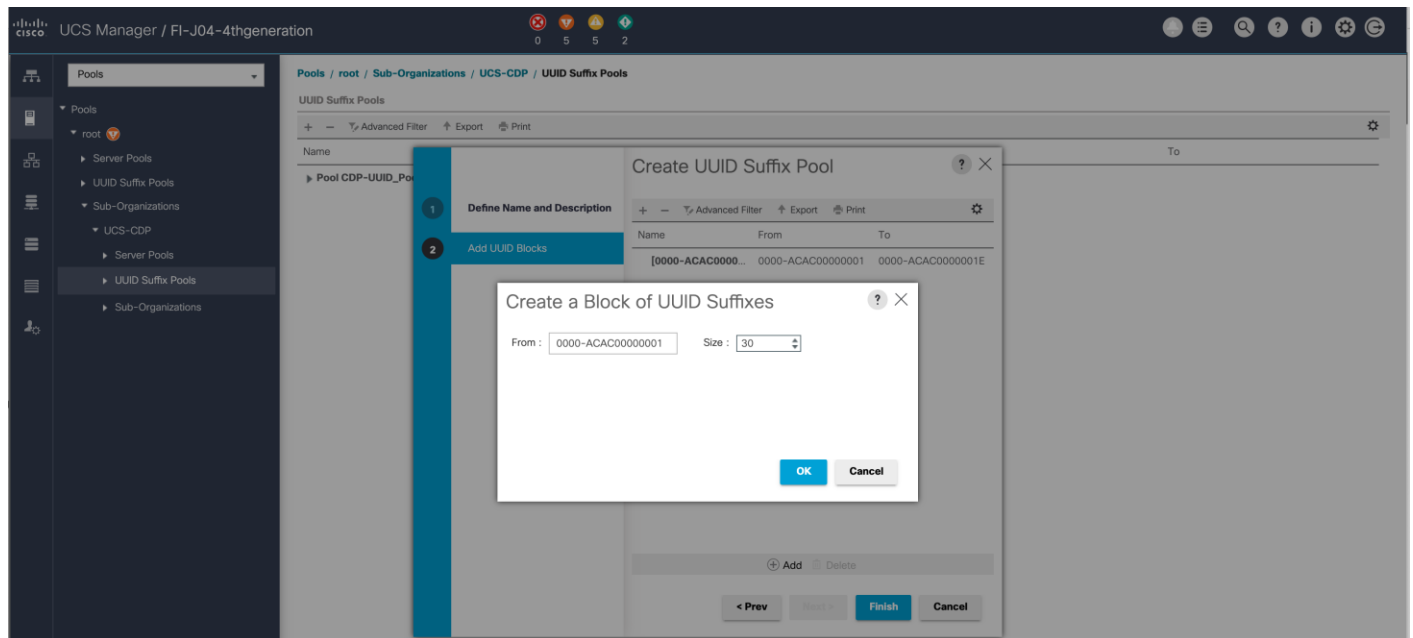


Figure 28. Create a Block of UUID Suffixes



Server Pool Creation

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

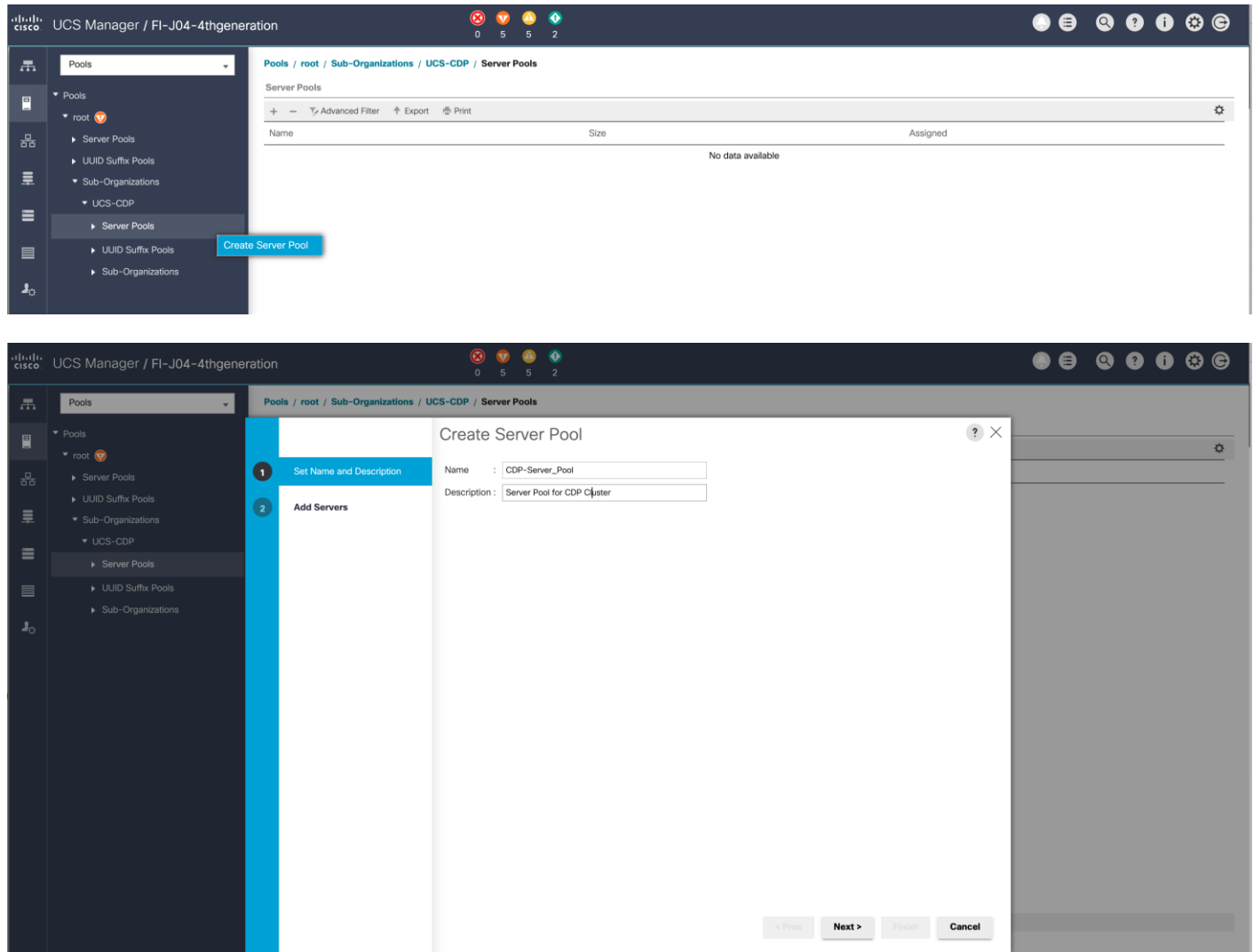


Consider creating unique server pools to achieve the granularity that is required in your environment.

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

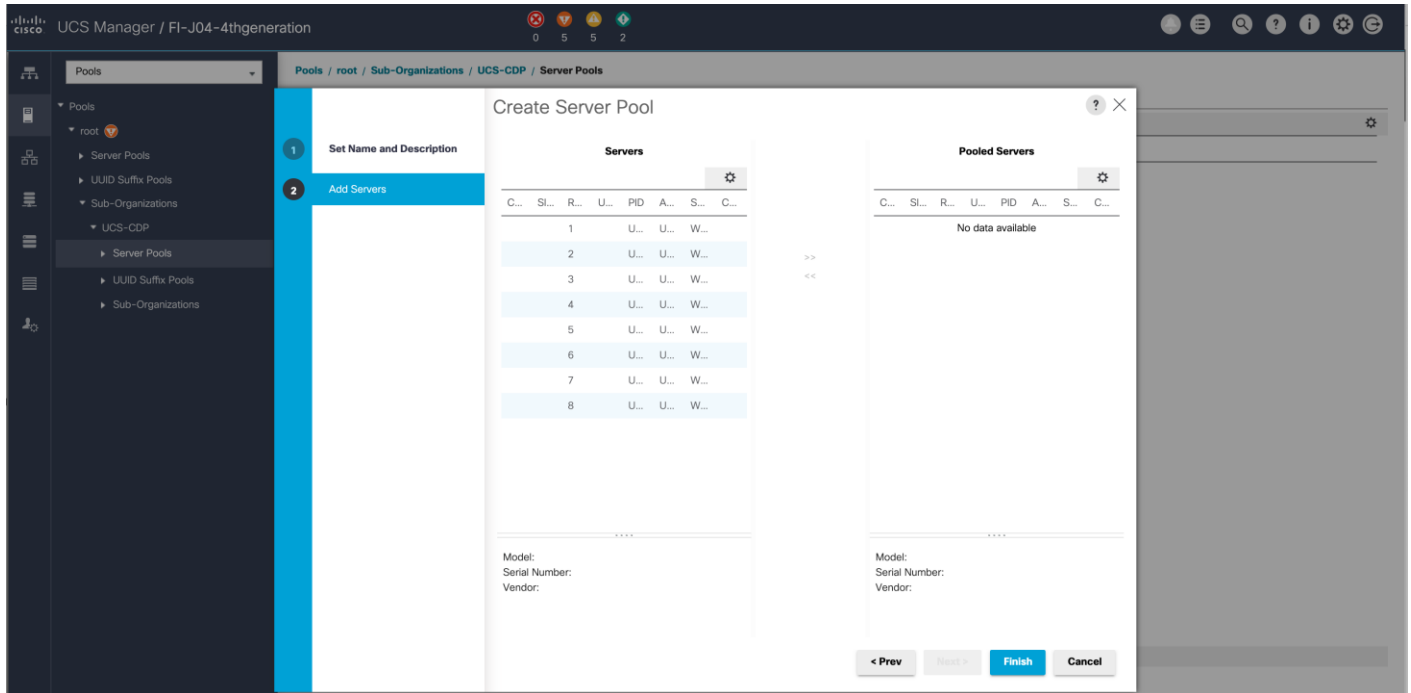
2. Select Pools > root > Sub-Organization > UCS-CDP > right-click Server Pools > Select Create Server Pool.
3. Enter name of the server pool.
4. Optional: Enter a description for the server pool then click Next.

Figure 29. Create Server Pool

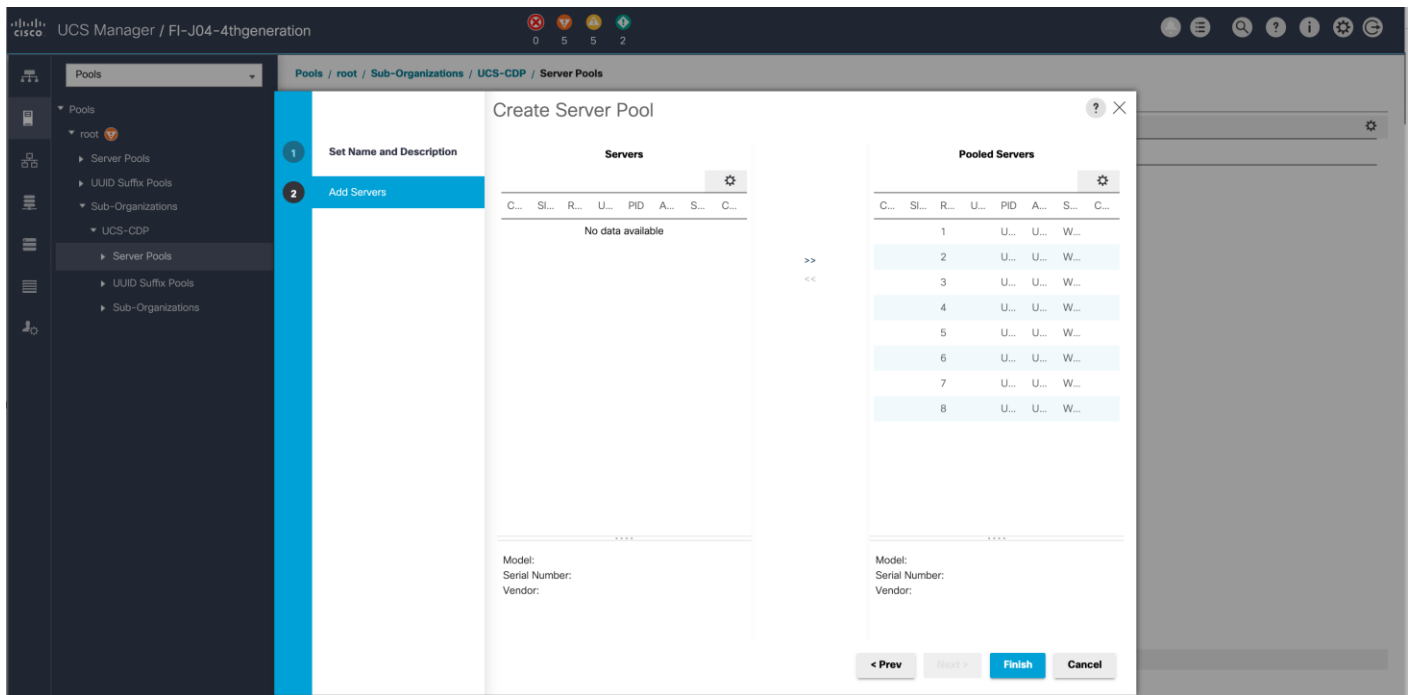


5. Select servers to be used for the deployment and click > to add them to the server pool. In our case we added thirty servers in this server pool.
6. Click Finish and then click OK.

Figure 30. Add Server in the Server Pool



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

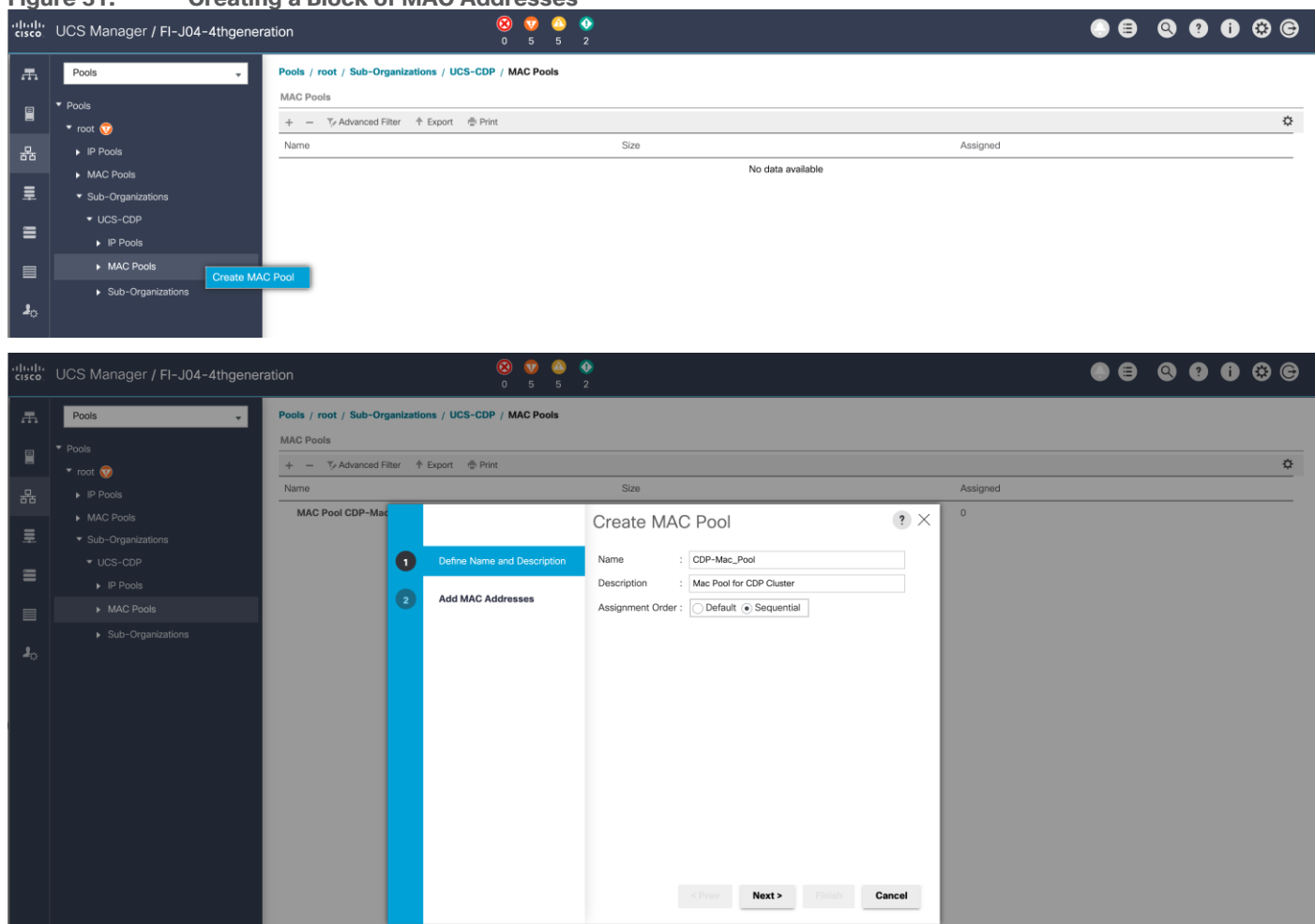


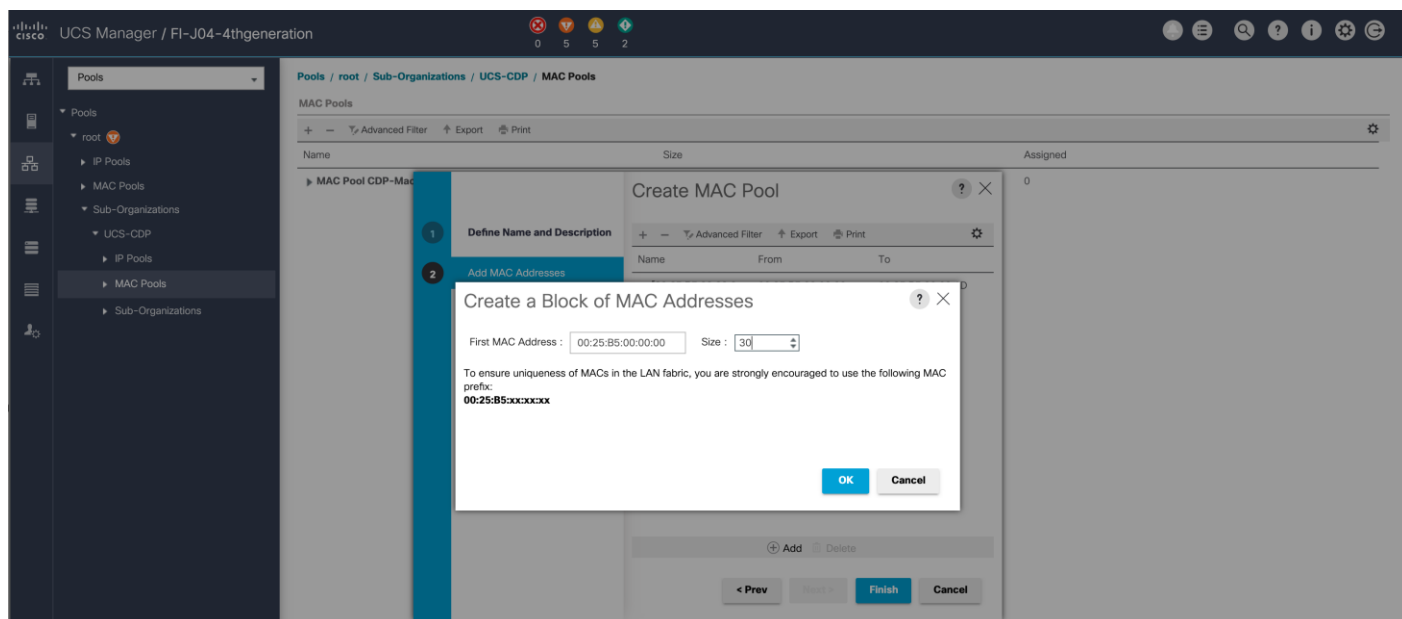
MAC Pool Creation

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-CDP> 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 31. Creating a Block of MAC Addresses



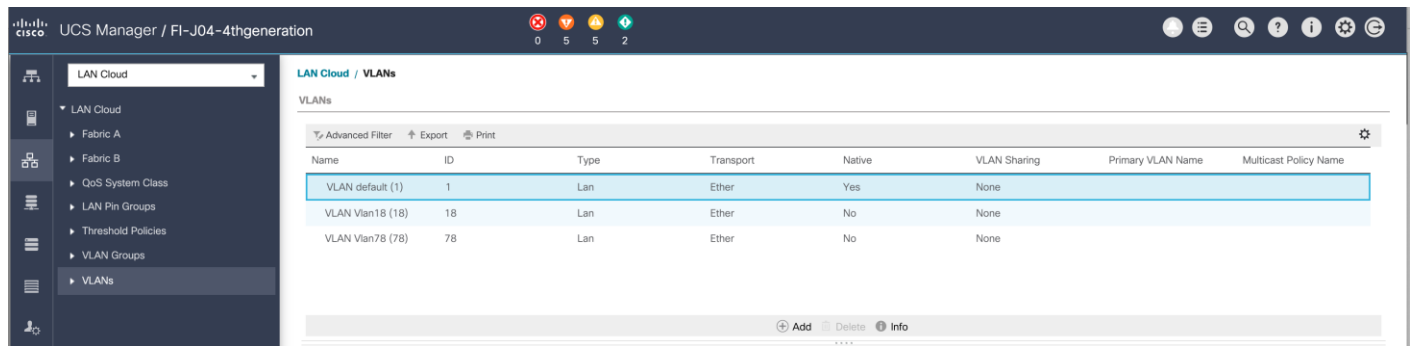


Configure VLAN

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

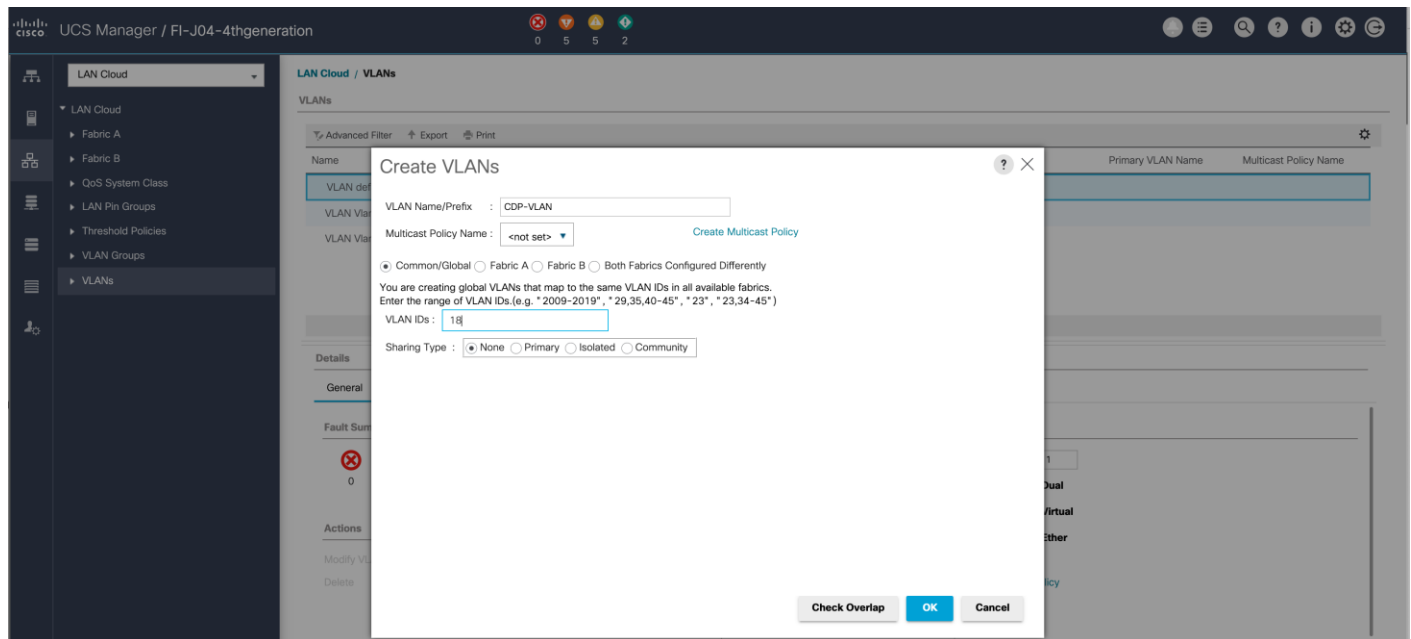
1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select LAN > LAN Cloud.
3. Right-click VLANs.
4. Select Create VLANs.
5. Enter the name of the VLAN to be used.
6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter <VLAN Number> as the ID of the VLAN ID.
8.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

Figure 32. Create VLAN



The NIC will carry the data traffic from VLAN18. 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 33. Create VLANs



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

Figure 34. Configure System Class QoS on Cisco UCS Fabric Interconnects

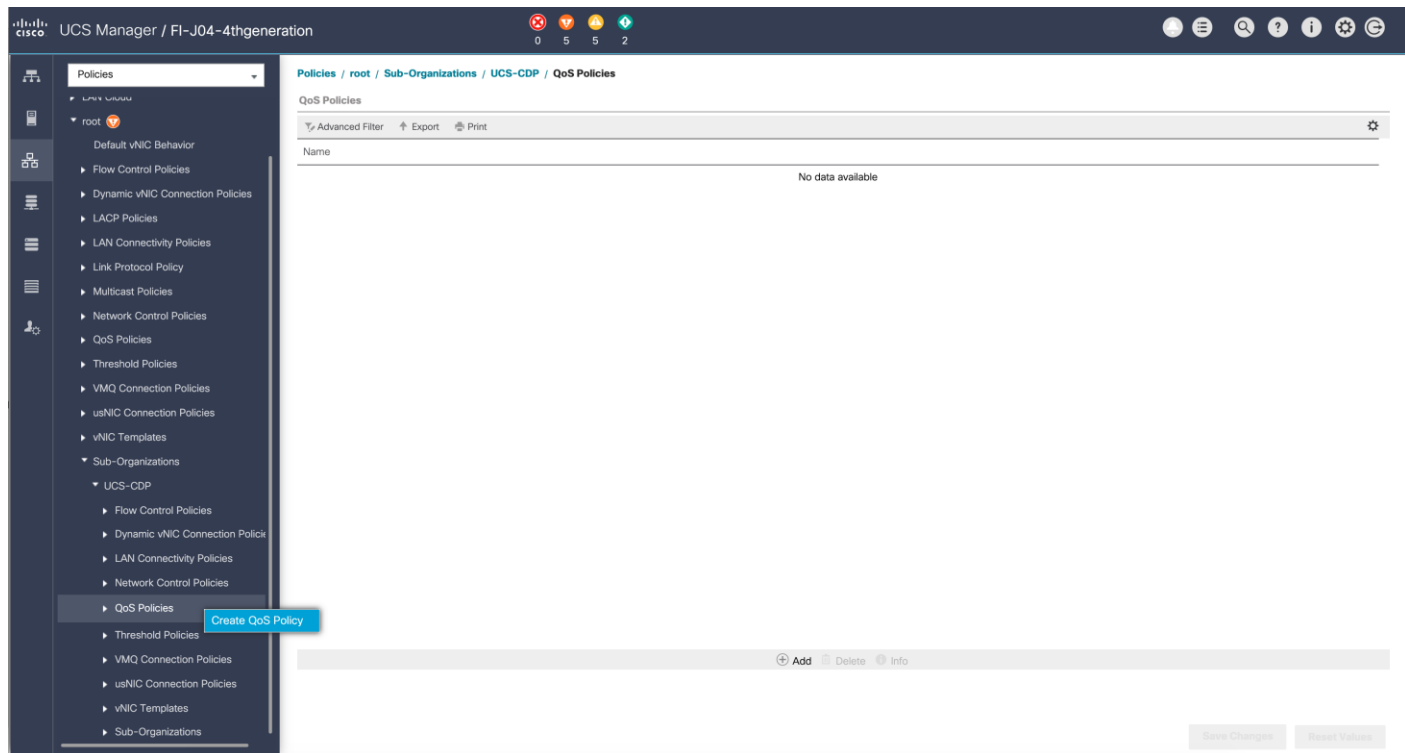
| Priority | Enabled | CoS | Packet Drop | Weight | Weight (%) | MTU | Multicast Optimized |
|---------------|-------------------------------------|-----|-------------------------------------|--------|------------|--------|--------------------------|
| Platinum | <input checked="" type="checkbox"/> | 5 | <input type="checkbox"/> | 10 | 100 | 9216 | <input type="checkbox"/> |
| Gold | <input type="checkbox"/> | 4 | <input checked="" type="checkbox"/> | 9 | N/A | normal | <input type="checkbox"/> |
| Silver | <input type="checkbox"/> | 2 | <input checked="" type="checkbox"/> | 8 | N/A | normal | <input type="checkbox"/> |
| Bronze | <input type="checkbox"/> | 1 | <input checked="" type="checkbox"/> | 7 | N/A | normal | <input type="checkbox"/> |
| Best Effort | <input checked="" type="checkbox"/> | Any | <input checked="" type="checkbox"/> | 5 | N/A | normal | <input type="checkbox"/> |
| Fibre Channel | <input checked="" type="checkbox"/> | 3 | <input type="checkbox"/> | 5 | N/A | 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 Cisco UCS Manager GUI.
2. Select LAN > Policies > root > UCS-CDP > QoS Policies.
3. Right-click QoS Policies.
4. Select Create QoS Policy.

Figure 35. Create QoS Policy




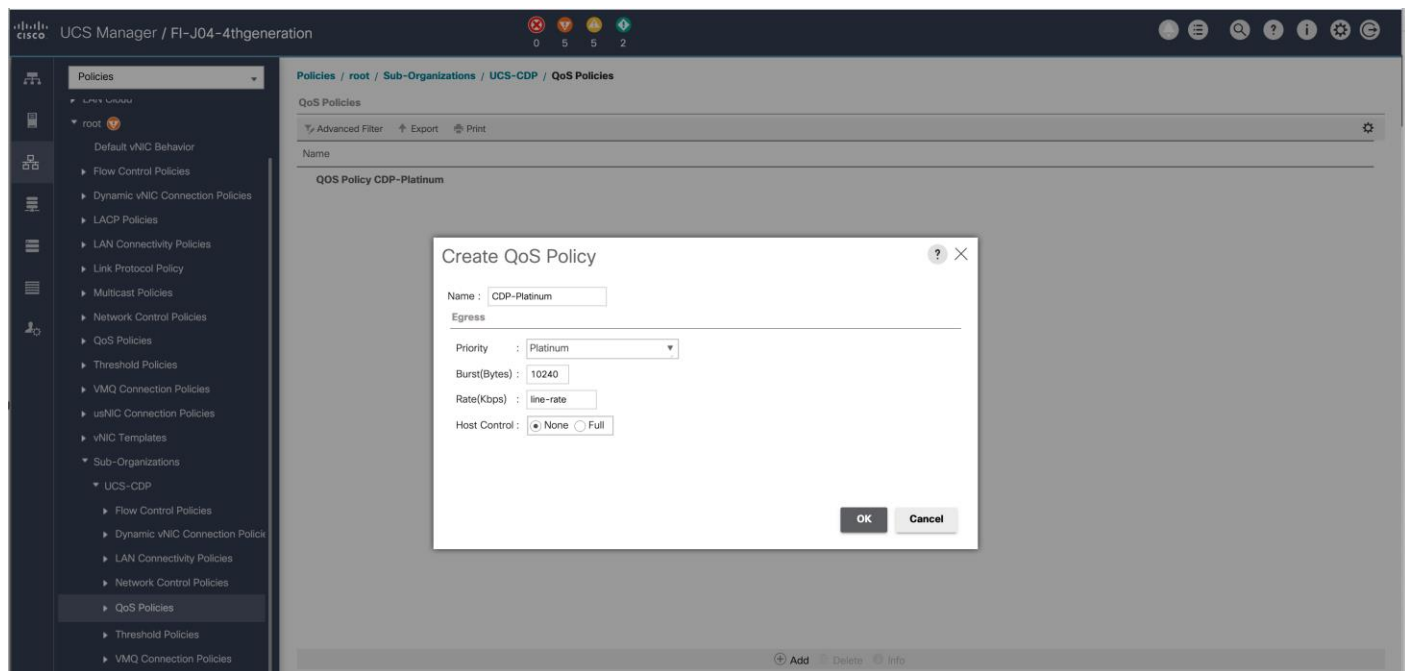
 We created a Platinum class QoS policy for this solution.

Figure 36. Platinum QoS Policy

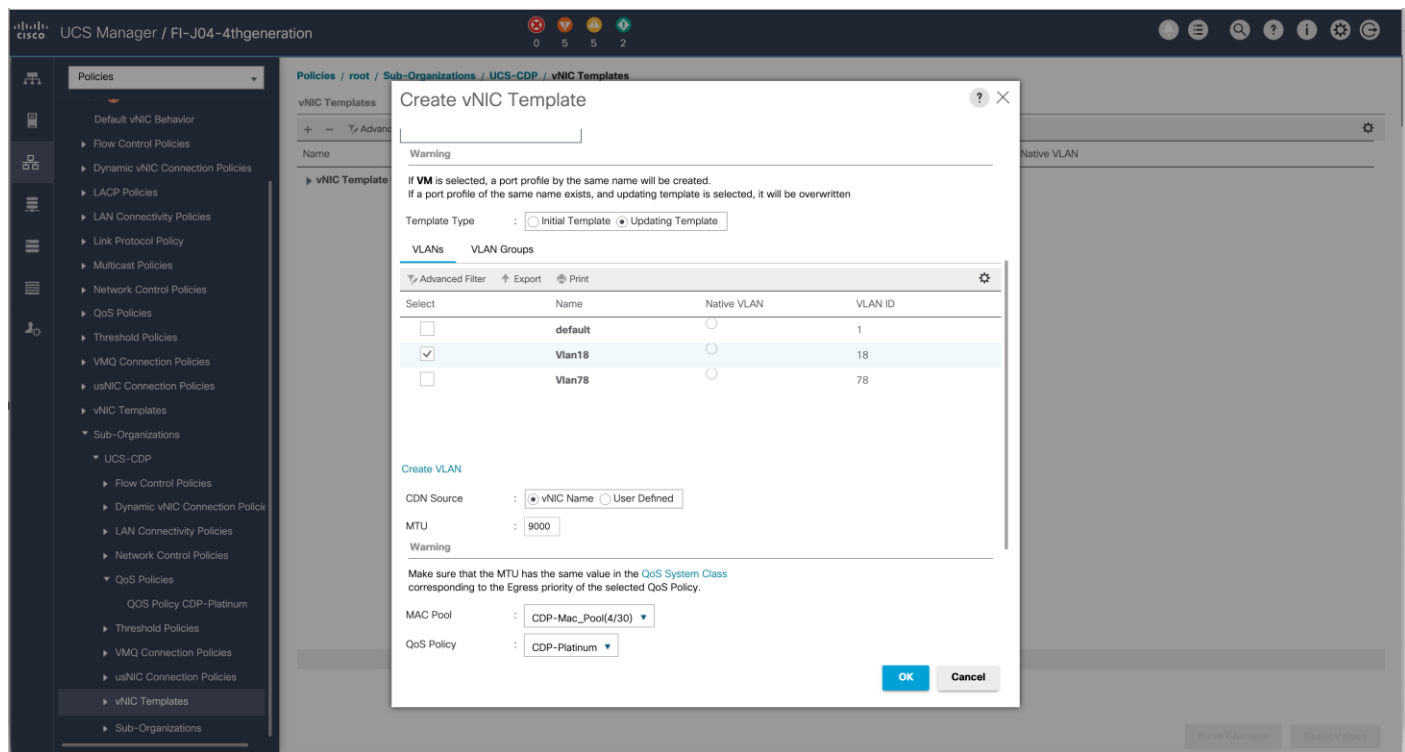
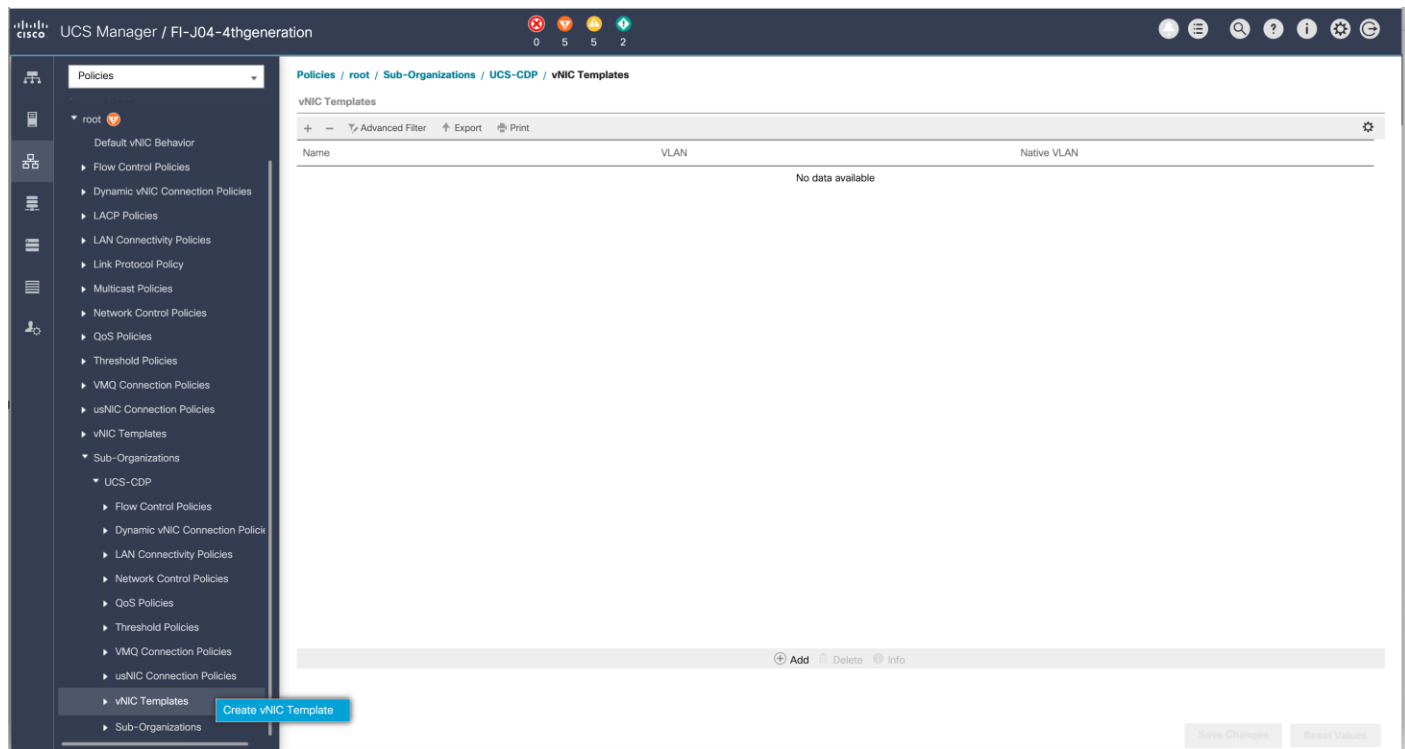


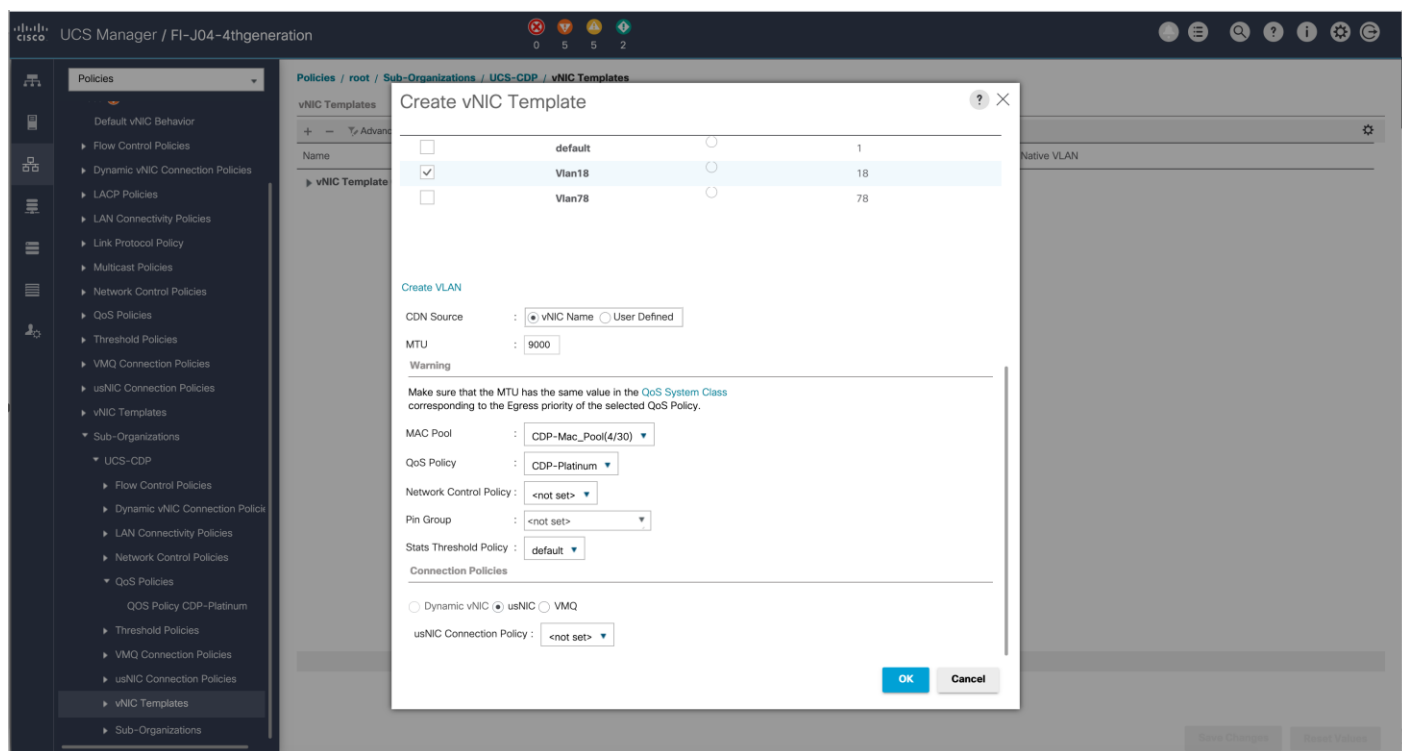
Create vNIC Templates

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-CDP> 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 37. Create the vNIC Template





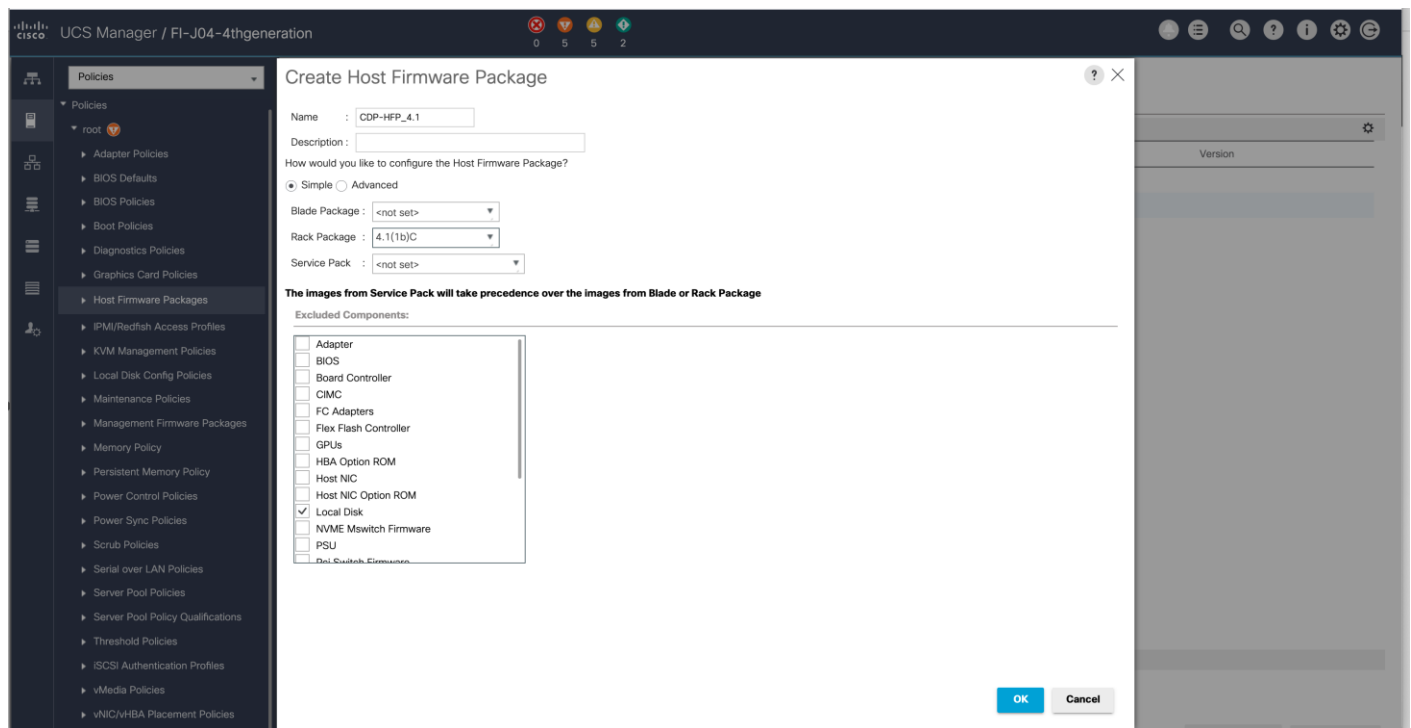
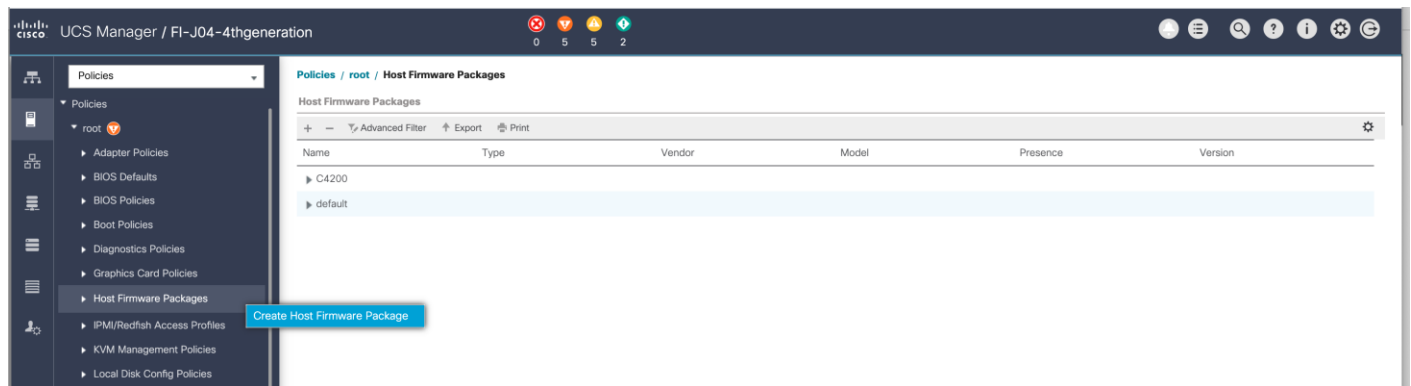
Create Host Firmware Package

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-CDP> 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 38. Host Firmware Package



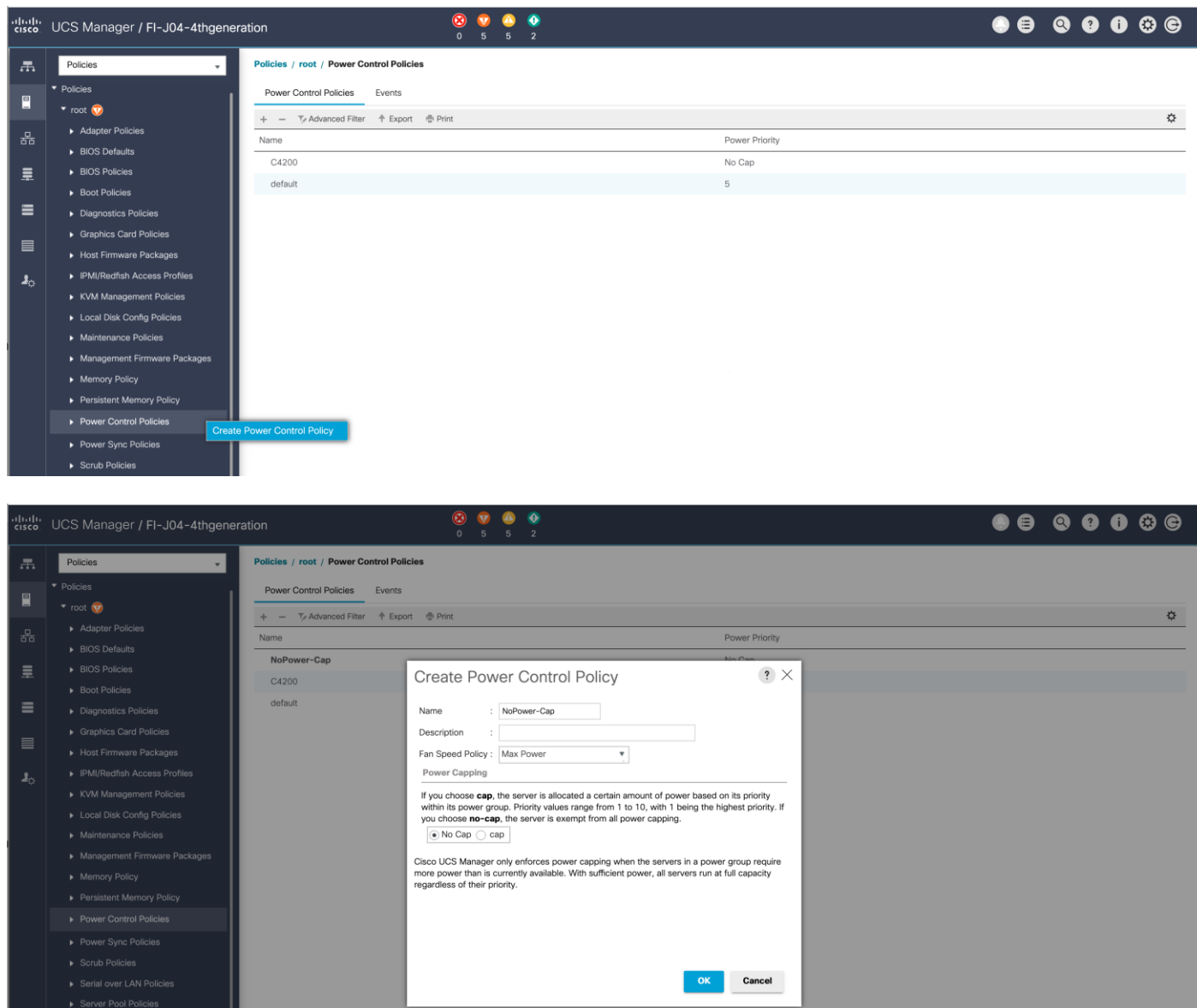
Create Power Control Policy

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-CDP> 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 39. Create Power Control Policy



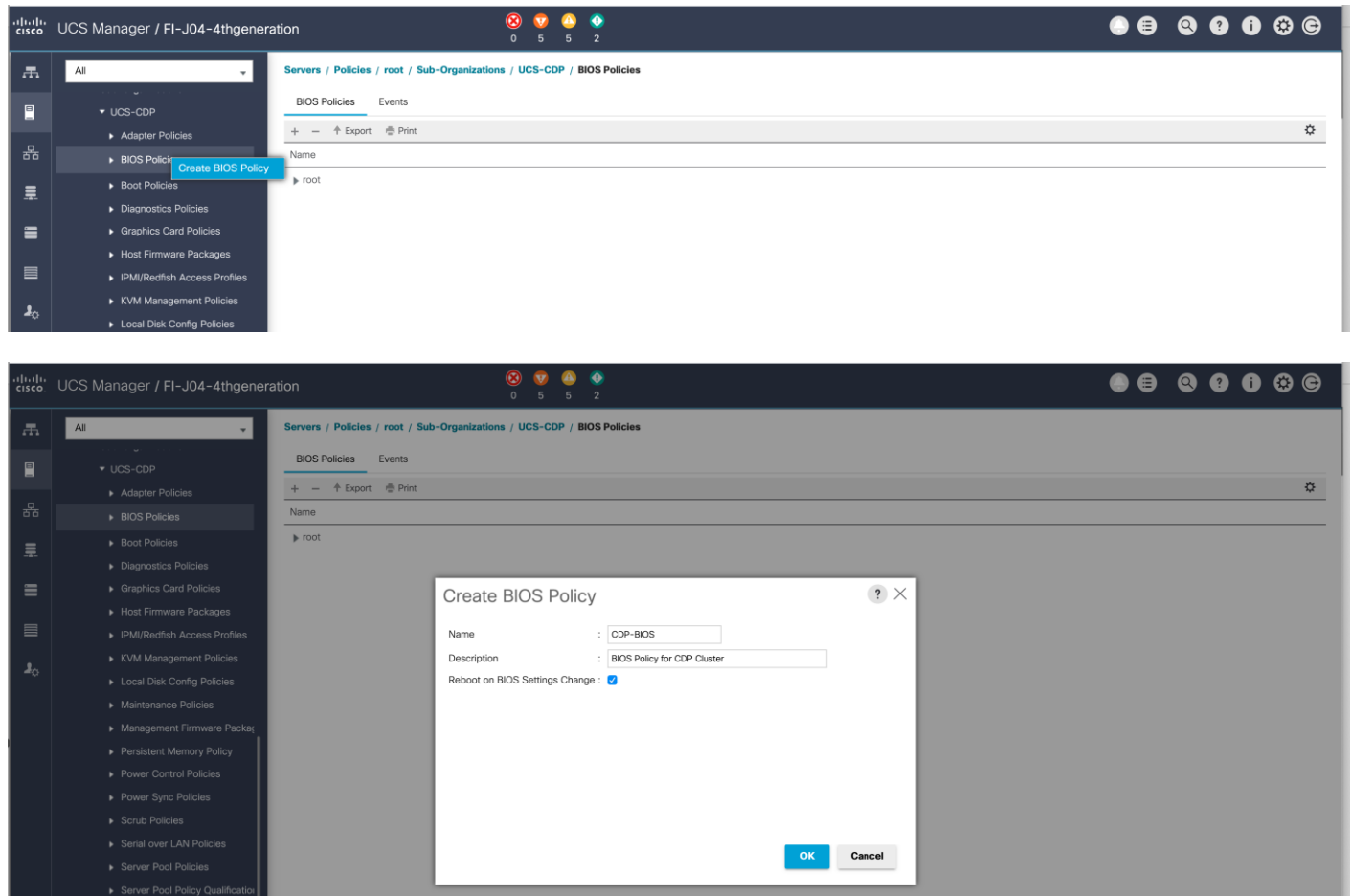
Create Server BIOS Policy

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-CDP > BIOS Policies.

3. Right-click BIOS Policies.
4. Select Create BIOS Policy.
5. Enter the BIOS policy name.

Figure 40. BIOS Configuration



The screenshot shows the Cisco UCS Manager interface for configuring BIOS settings. The breadcrumb path is: Servers / Policies / root / Sub-Organizations / UCS-CDP / BIOS Policies / CDP-BIOS. The 'Advanced' tab is selected, showing a list of BIOS settings. The table below lists the settings and their current values:

| BIOS Setting | Value |
|------------------------------|------------------|
| XPT Prefetch | Platform Default |
| Core Performance Boost | Platform Default |
| Downcore control | Platform Default |
| Global C-state Control | Enabled |
| L1 Stream HW Prefetcher | Platform Default |
| L2 Stream HW Prefetcher | Platform Default |
| Determinism Slider | Power |
| IOMMU | Disabled |
| Bank Group Swap | Platform Default |
| Chipselect Interleaving | Platform Default |
| Configurable TDP Control | Platform Default |
| AMD Memory Interleaving | Auto |
| AMD Memory Interleaving Size | Platform Default |
| SMEE | Platform Default |
| SMT Mode | Platform Default |
| SVM Mode | Platform Default |
| Demand Scrub | Platform Default |



For more information, go to: [“Hadoop Tuning Guide for AMD EPYC Processor Based Servers”](#)



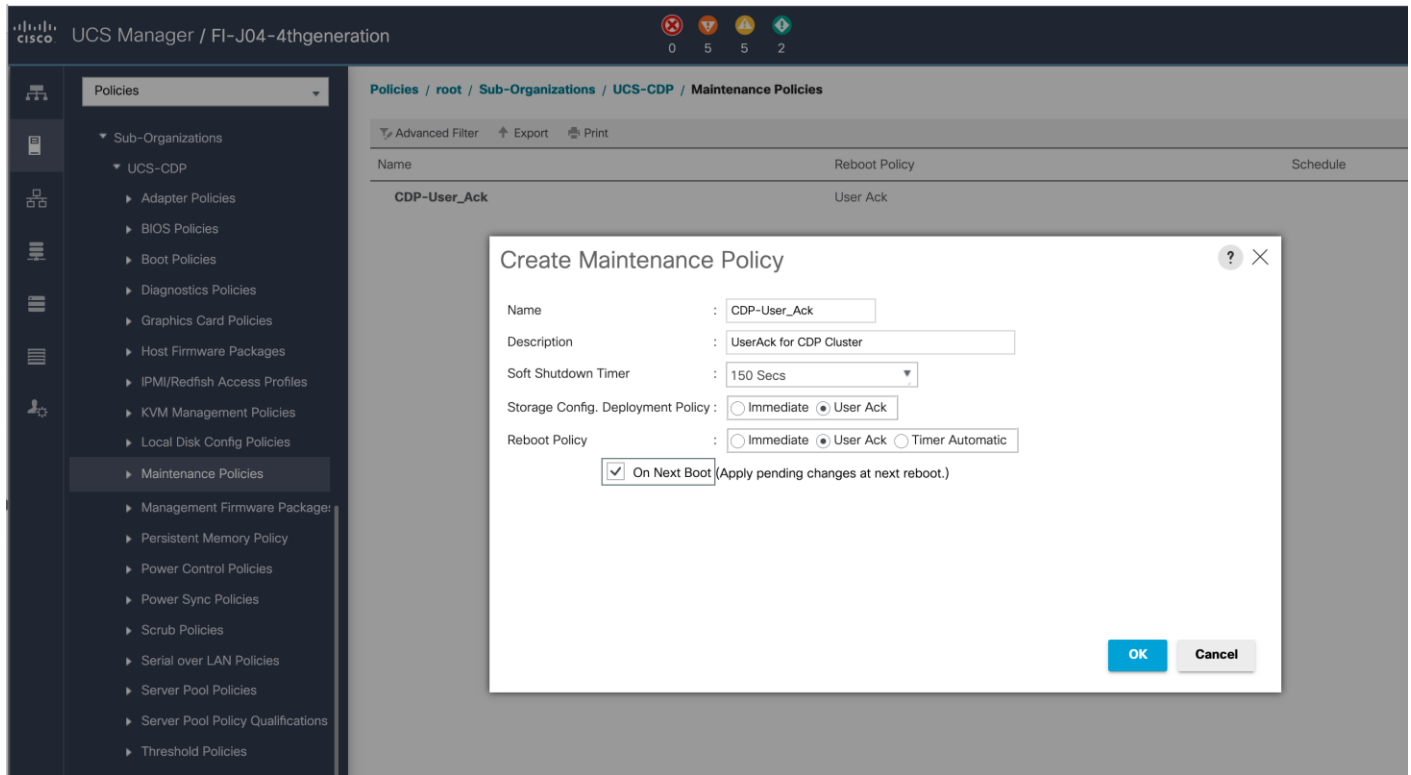
BIOS settings can have a significant performance impact, depending on the workload and the applications. The BIOS settings listed in this section is for configurations optimized for best performance which can be adjusted based on the application, performance, and energy efficiency requirements.

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-CDP> 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 41. Create Server Maintenance Policy

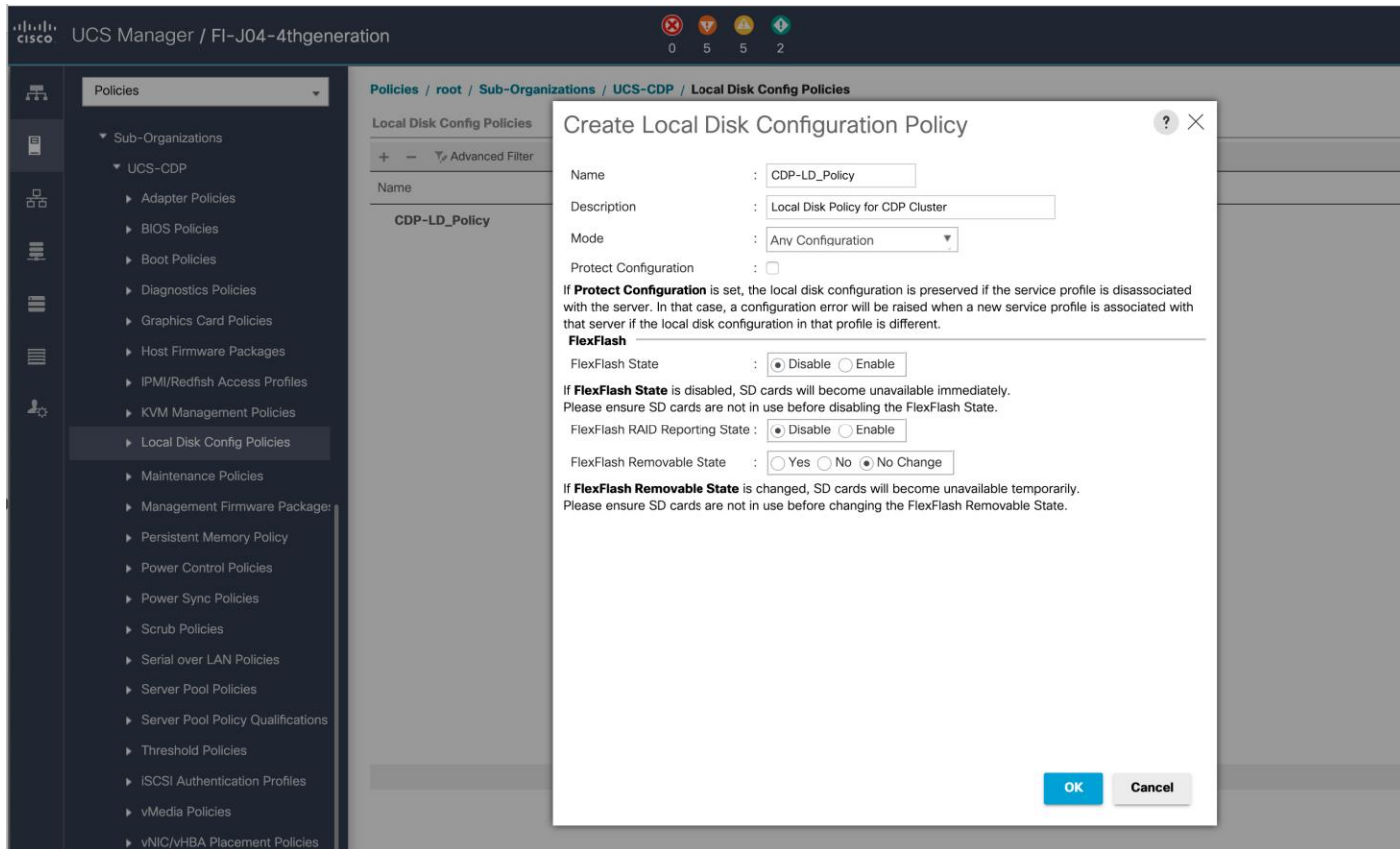


Create the Local Disk Configuration Policy

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

1. Select the Servers tab in the Cisco UCS Manager GUI.
2. Select Policies > root > Sub-Organization > UCS-CDP > 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 42. Create the Local Disk Configuration Policy



Create Boot Policy

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

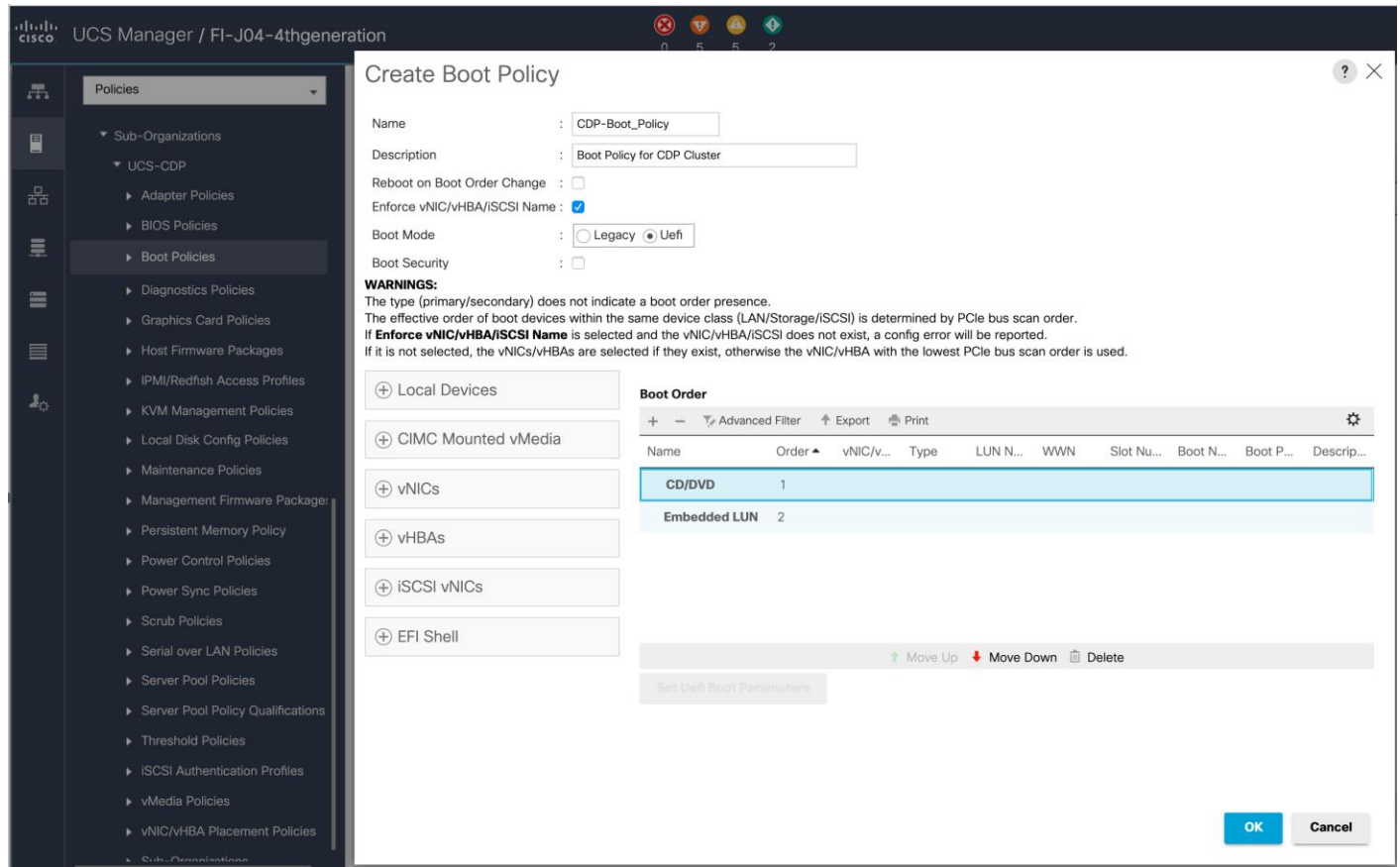
1. Select the Servers tab in the Cisco UCS Manager GUI.
2. Select Policies > root.
3. Right-click the Boot Policies.
4. Select Create Boot Policy.

The screenshot shows the Cisco UCS Manager interface for a system named 'J14-Bigdata-TPC-M5'. The breadcrumb navigation path is 'Policies / root / Sub-Organizations / UCS-BDA-CDP / Boot Policies'. The left-hand navigation pane shows a tree structure under 'Policies' with sub-organizations and various policy categories. The 'Boot Policies' category is selected, and a 'Create Boot Policy' button is visible. The main content area shows the 'Boot Policies' tab with a table header and some action buttons like 'Advanced Filter', 'Export', and 'Print'.

| Name | Order | vNIC/vHBA/iSCSI v... | Type |
|------|-------|----------------------|------|
|------|-------|----------------------|------|

5. Enter "CDP-Boot_Policy" 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 43. Create Boot Policy for Cisco UCS Server(s)



Create Storage Profile for Individual RAID0 for Data Nodes

To create the storage profile for the individual RAID0, follow these steps:

1. On the UCSM navigation page, select the Storage tab.
2. From the Storage Profiles drop-down list, right-click and select Create Storage Profile.
3. Enter a name for the Storage Profile and click the LUN Set tab.
4. Click Add.
5. Select the properties for the LUN set:
 - a. Enter a name for LUN set. Disk Slot Range – 1 – 6 (Depends on number of drives installed in a server).
 - b. 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

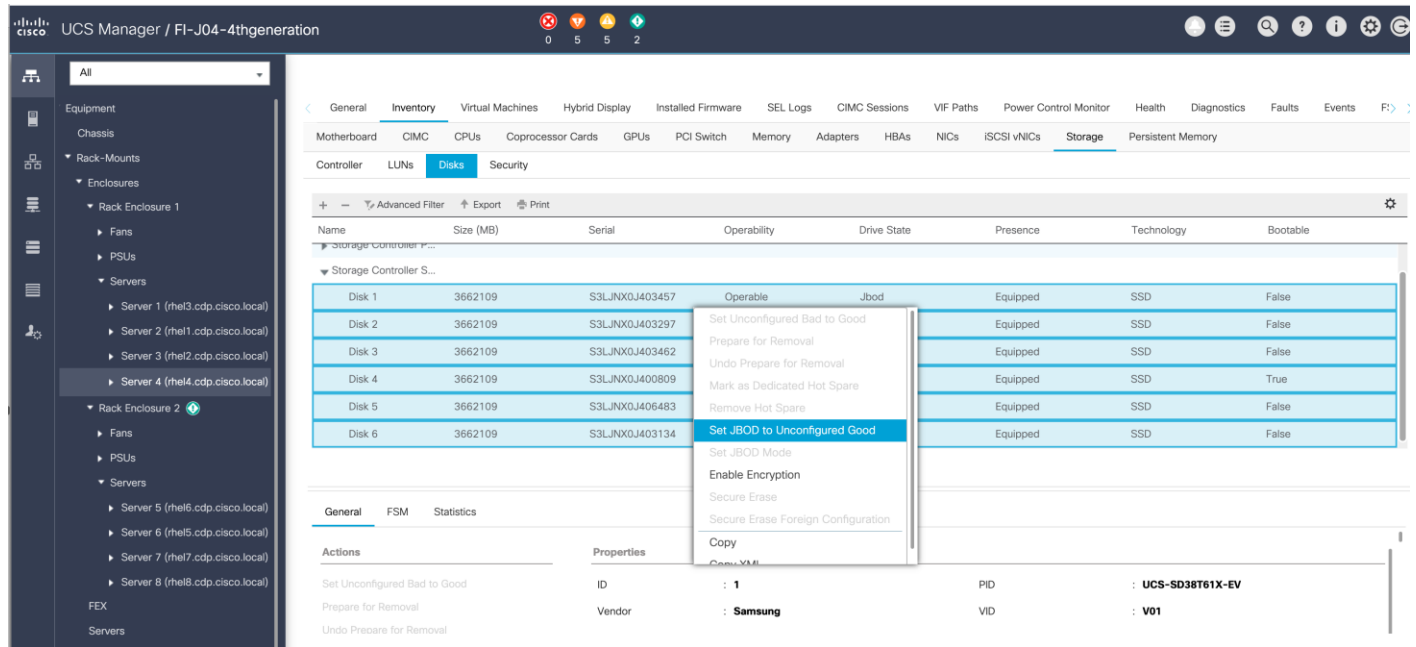
The screenshot displays the UCS Manager interface for a FI-J04-4thgeneration server. The main window is titled 'Storage / Storage Profiles / root / Sub-Organizations / UCS-CDP'. A 'Create Storage Profile' dialog is open, showing a 'LUN Set' configuration. Overlaid on this is a 'Create LUN Set' dialog with the following settings:

- RAID Level: RAID 0 Striped
- Disk Slot Range: 1-6
- Strip Size (KB): 1024KB
- Access Policy: Read Write
- Read Policy: Read Ahead
- Write Cache Policy: Write Back Good Bbu
- IO Policy: Direct
- Drive Cache: 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 44. Set JBOD Disks to Unconfigured Good



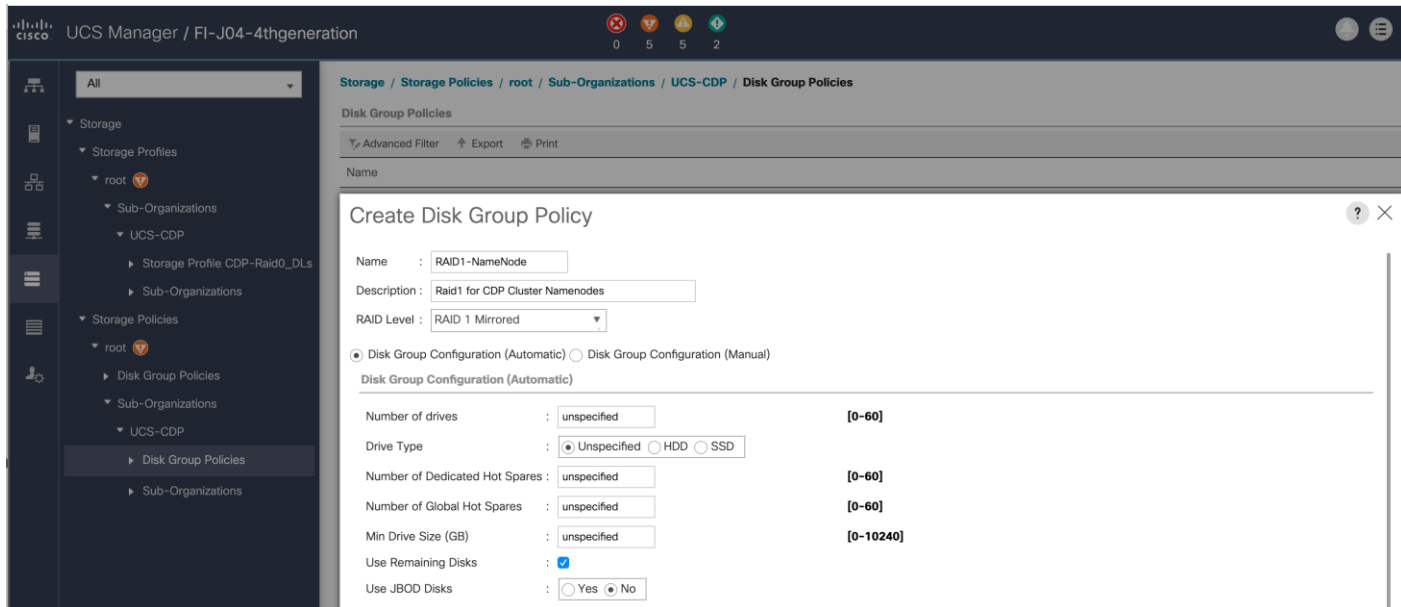
Create Storage Policy and Storage Profile RAID1 for Name Nodes

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 and select “Storage Policies”.
2. From the Storage Policies drop-down list, select and right-click “Disk Group Policies”. Select “Create Disk Group Policy”.
3. Enter a name for Disk Group Policy and select RAID level.
4. Select “Disk Group Configuration” (Automatic/Manual).

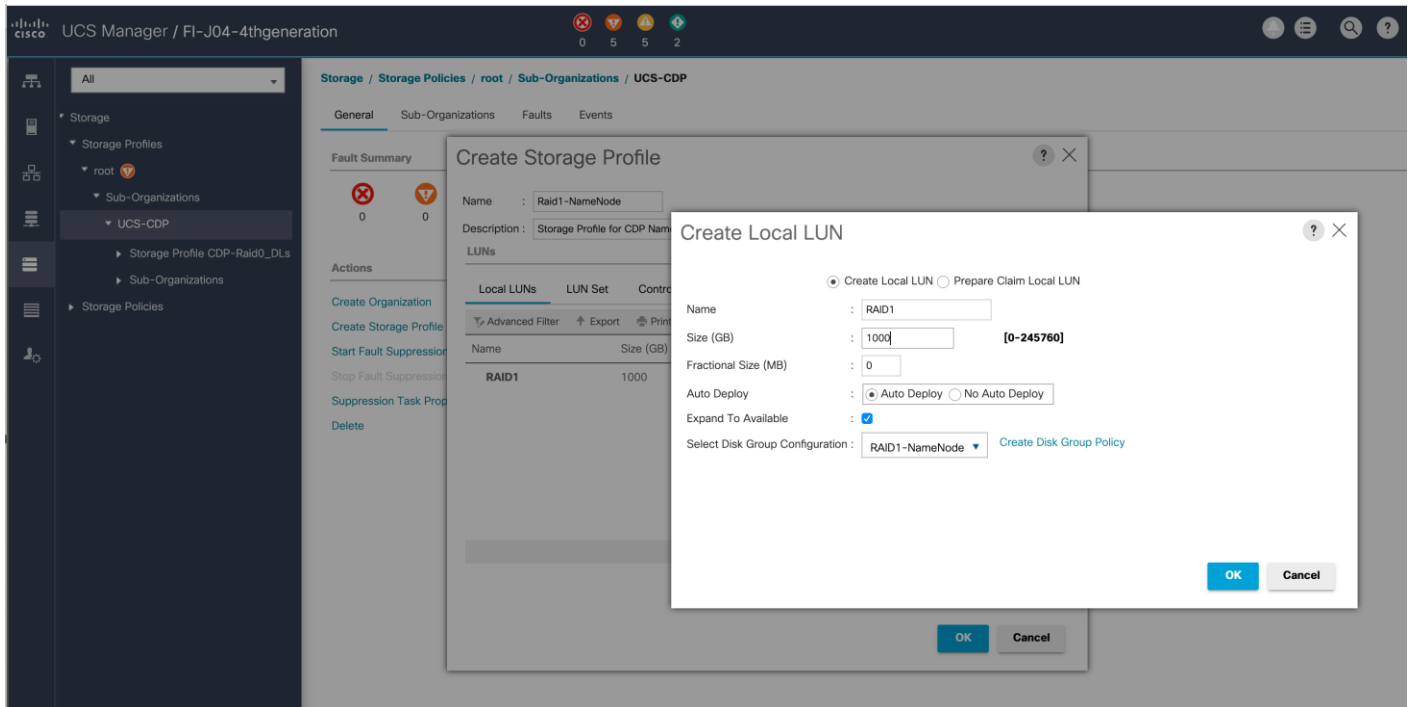
Figure 45. Disk Group Configuration.



5. Virtual Drive Configuration.



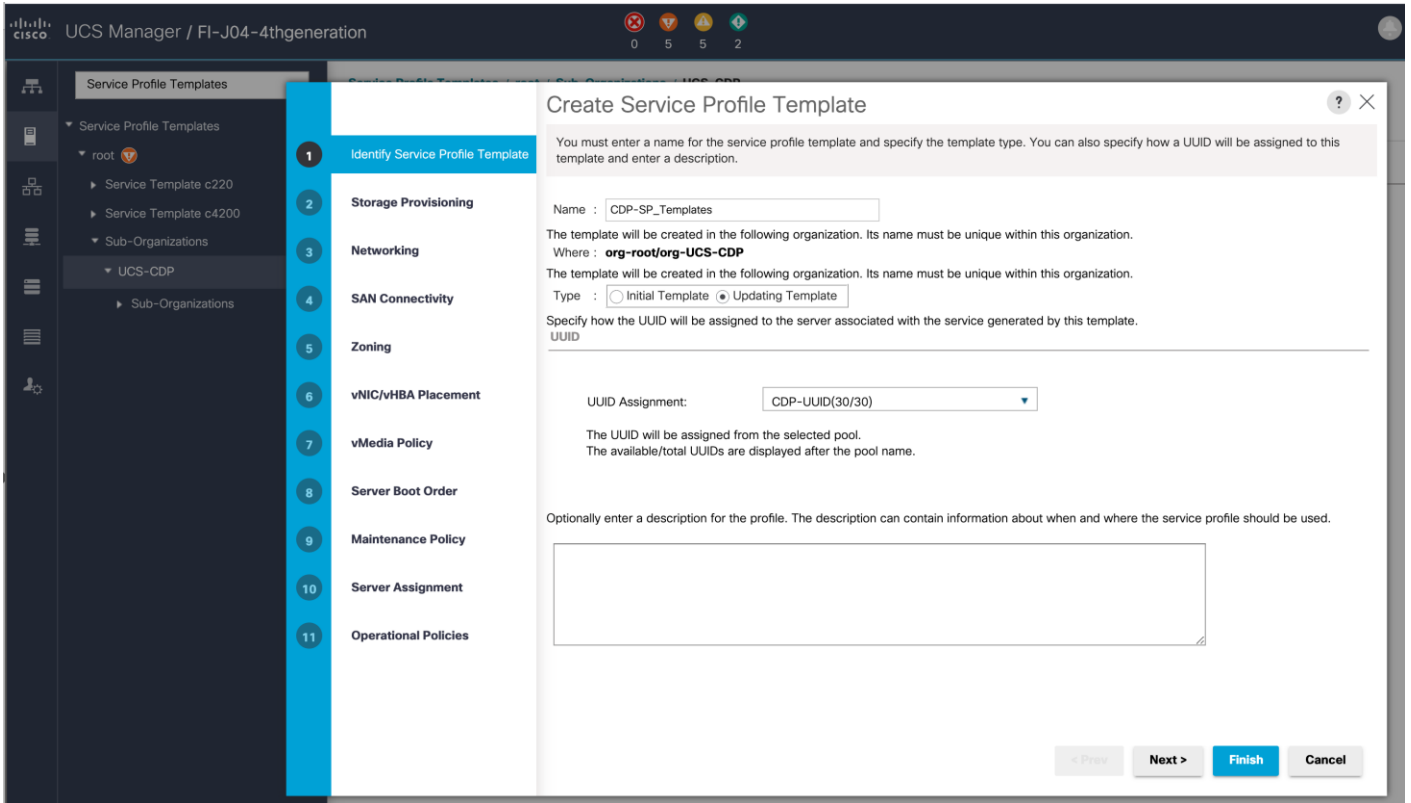
6. Select Storage Profiles, right-click and select Create Storage Profile
7. Enter a name for the Storage profile and click Add.
8. Enter a Local LUN name and select Auto Deploy.
9. Check the box for Expand to Available and from the drop-down list select the storage policy you want to attach with the Storage Profile. Click OK.



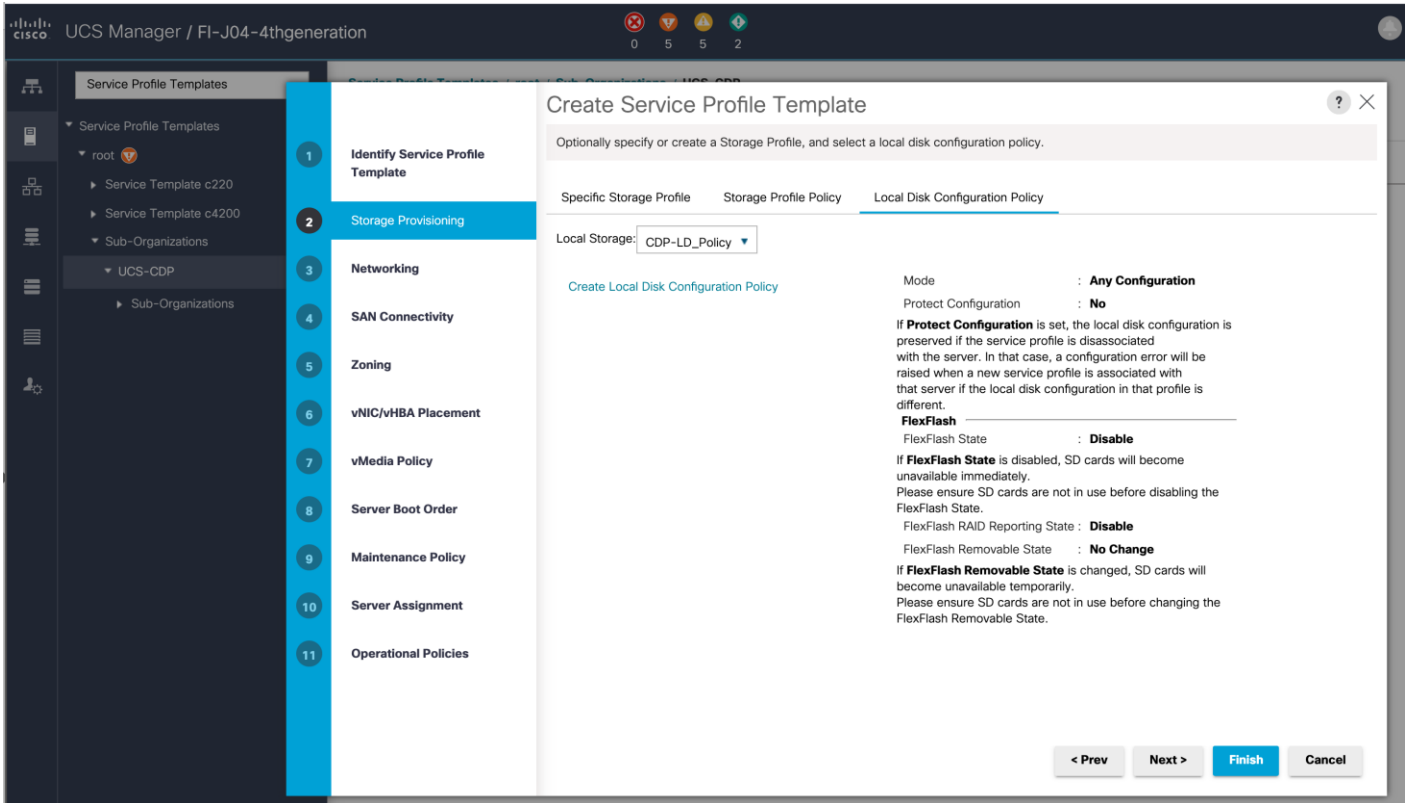
Create Service Profile Template

To create a service profile template, follow these steps:

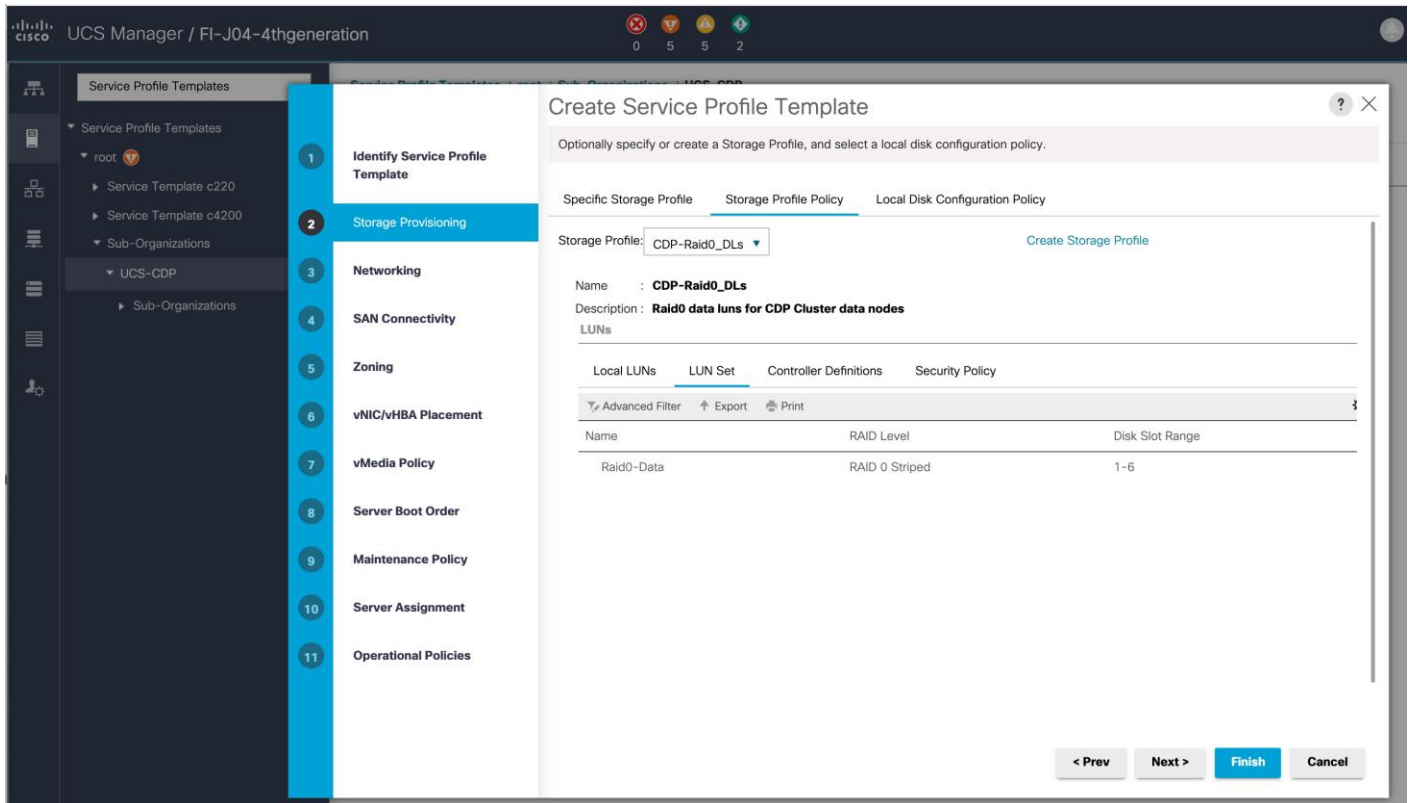
1. In the Cisco UCS Manager, go to Servers > Service Profile Templates > root Sub Organization > UCS-CDP> and right-click "Create Service Profile Template" as shown below.
2. Enter the Service Profile Template name, Updating Template as type of template and select the UUID Pool that was created earlier. Click Next.



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



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



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 the vNIC Template CDP-vNIC0 and the Adapter Policy CDP-Linux:
 - a. Tx Q / Rx Q - 8
 - b. Enable RSS on the server adapter setting
 - c. Completion Q 16
 - d. Interrupts 32
 - e. Tx / Rx ring sizes = 4096/4096

Create Ethernet Adapter Policy



Name : CDP-Linux

Description : Adapter Policy for the CDP Cluster Nodes

Resources

Pooled : Disabled Enabled

Transmit Queues : 8 [1-1000]

Ring Size : 4096 [64-4096]

Receive Queues : 8 [1-1000]

Ring Size : 4096 [64-4096]

Completion Queues : 16 [1-2000]

Interrupts : 32 [1-1024]

Options

Transmit Checksum Offload : Disabled Enabled

Receive Checksum Offload : Disabled Enabled

TCP Segmentation Offload : Disabled Enabled

TCP Large Receive Offload : Disabled Enabled

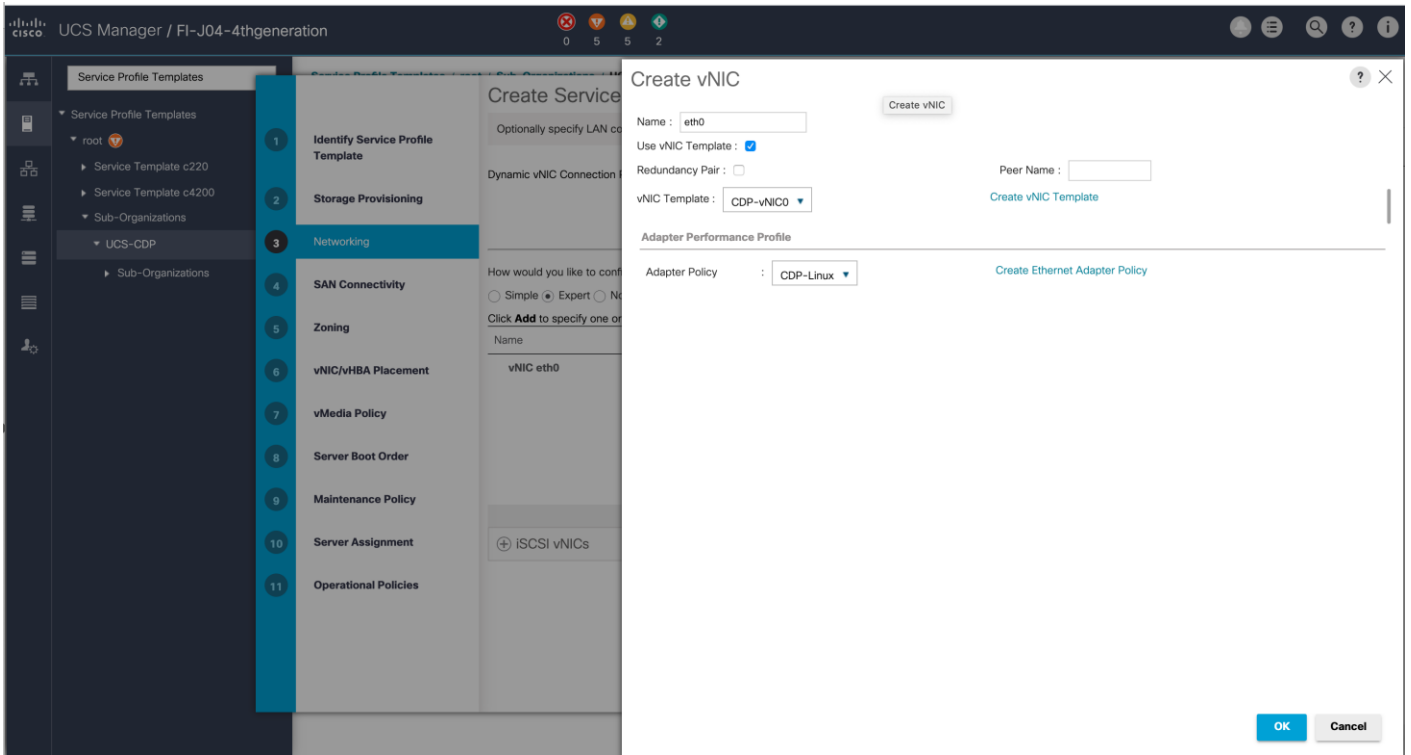
Receive Side Scaling (RSS) : Disabled Enabled

Accelerated Receive Flow Steering : Disabled Enabled

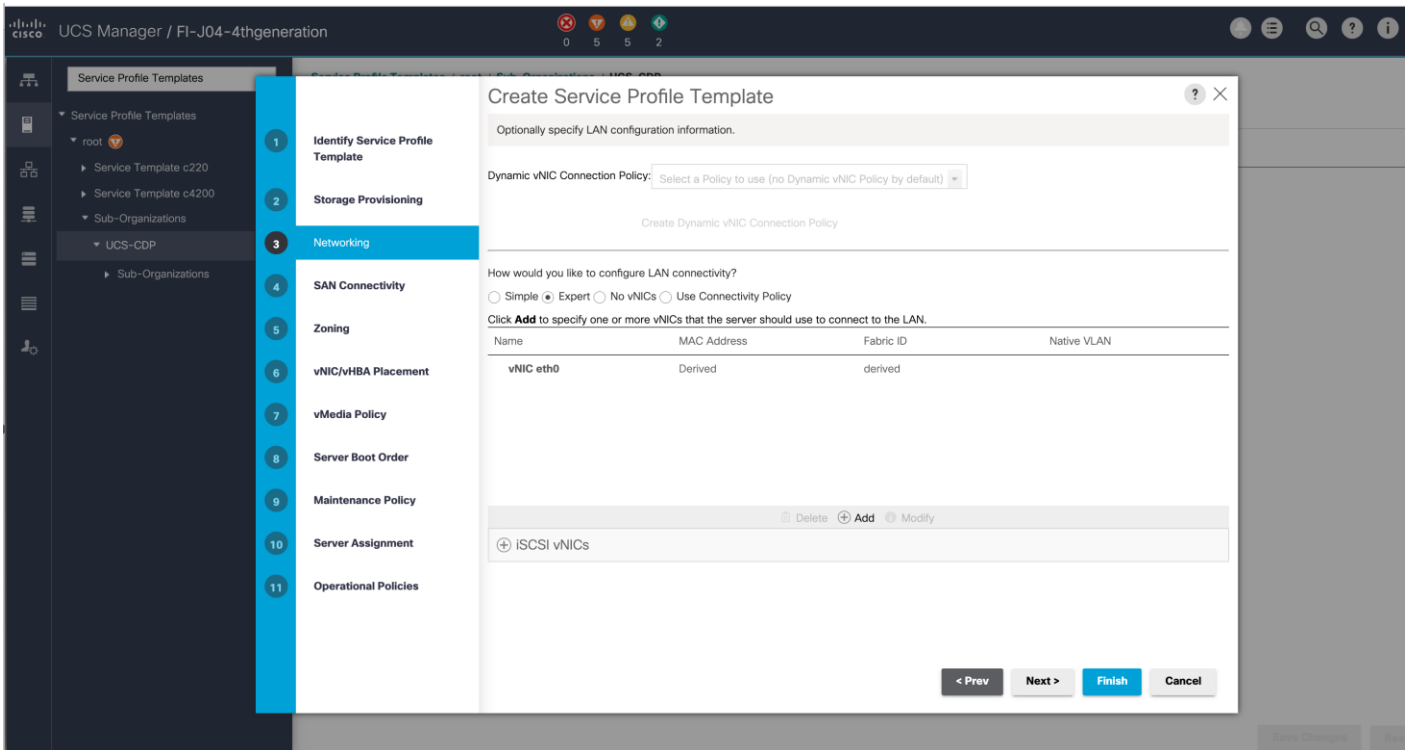
Network Virtualization using Generic Routing Encapsulation : Disabled Enabled

OK

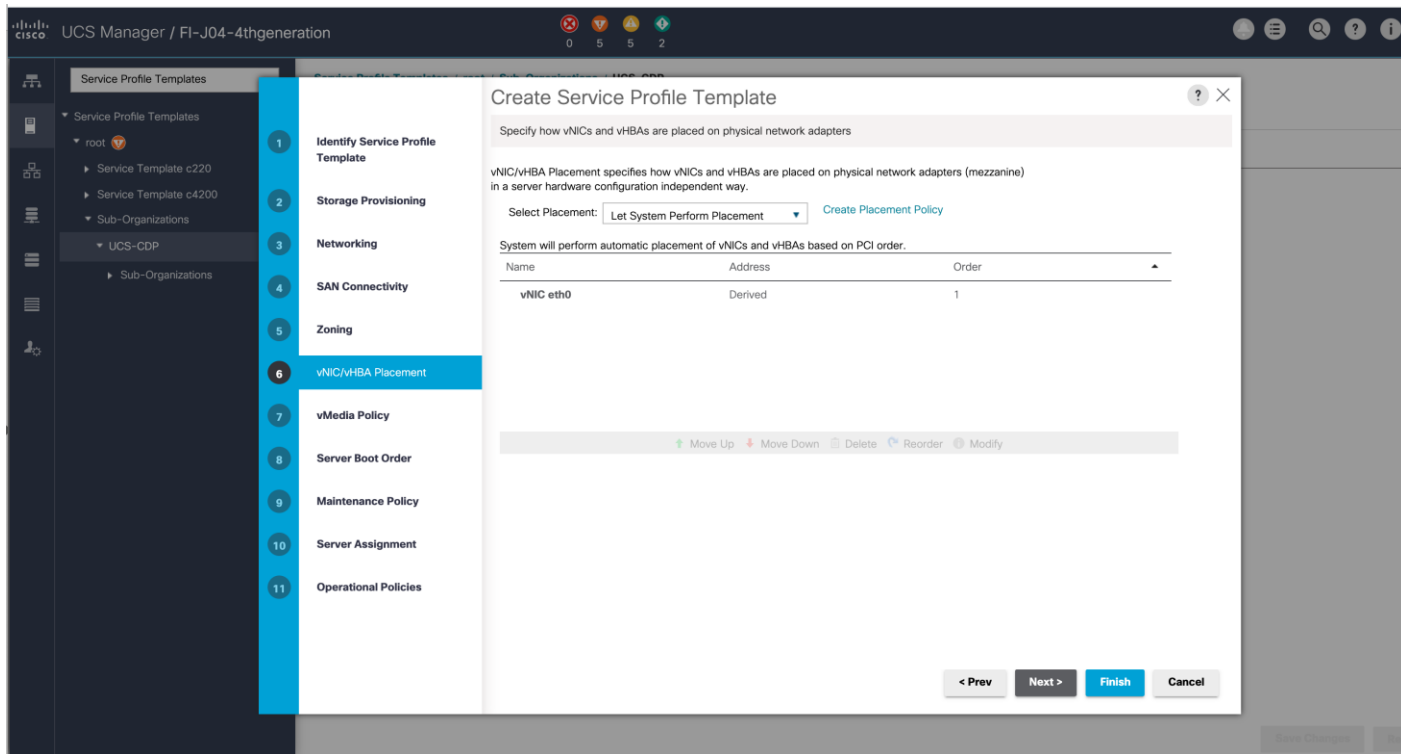
Cancel



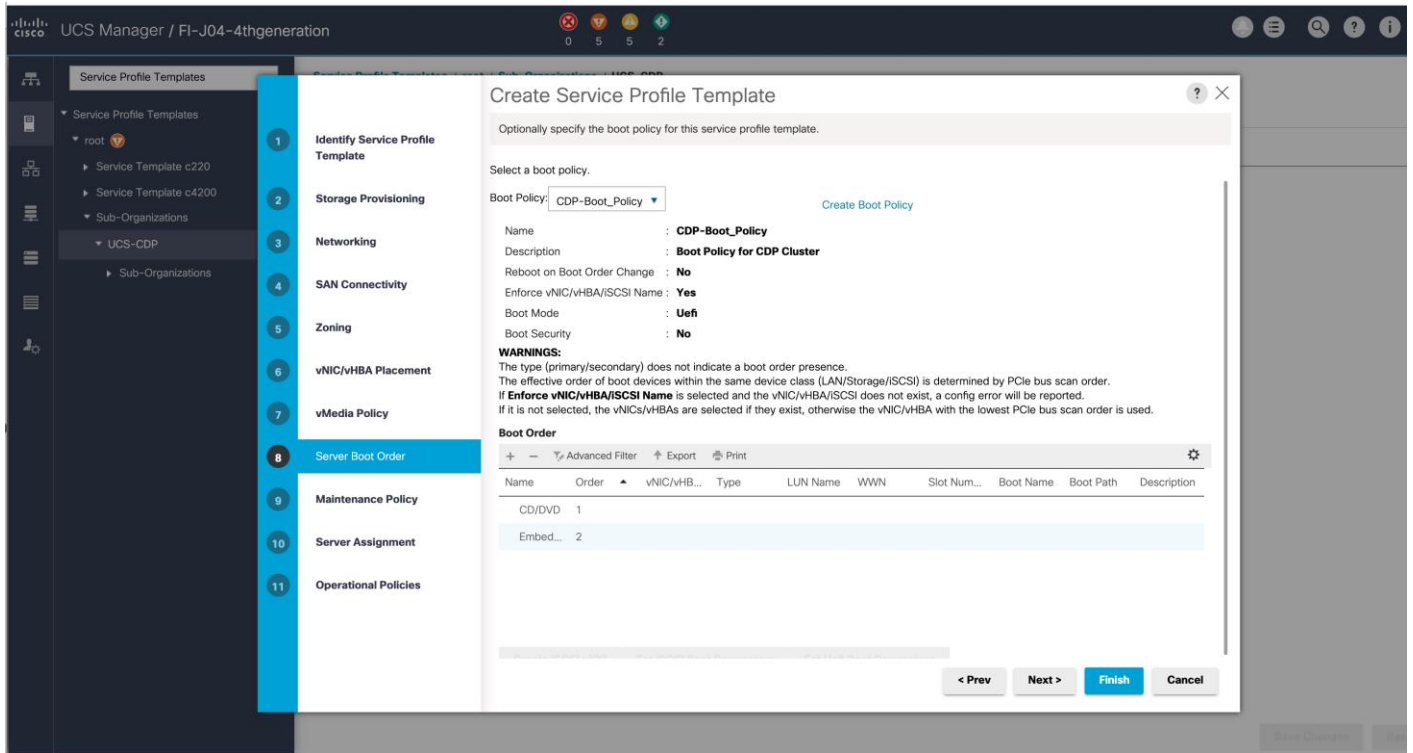
Optionally, Network Bonding can be setup on the vNICs for each host for redundancy as well as for increased throughput.



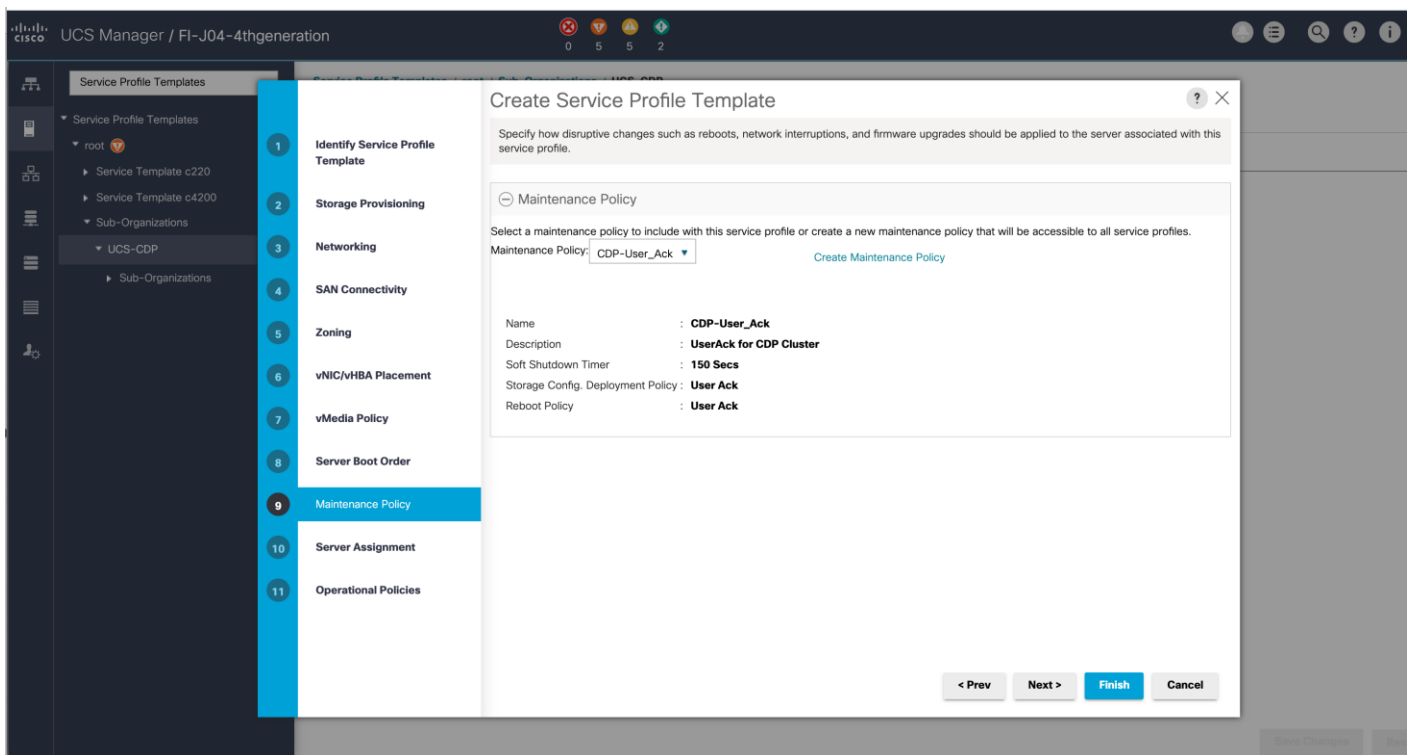
8. In the SAN Connectivity menu, select no vHBAs.
9. Click Next in the Zoning tab (no change) and Click Next.
10. Select Let System Perform Placement for vNIC/vHBA Placement. Click Next.



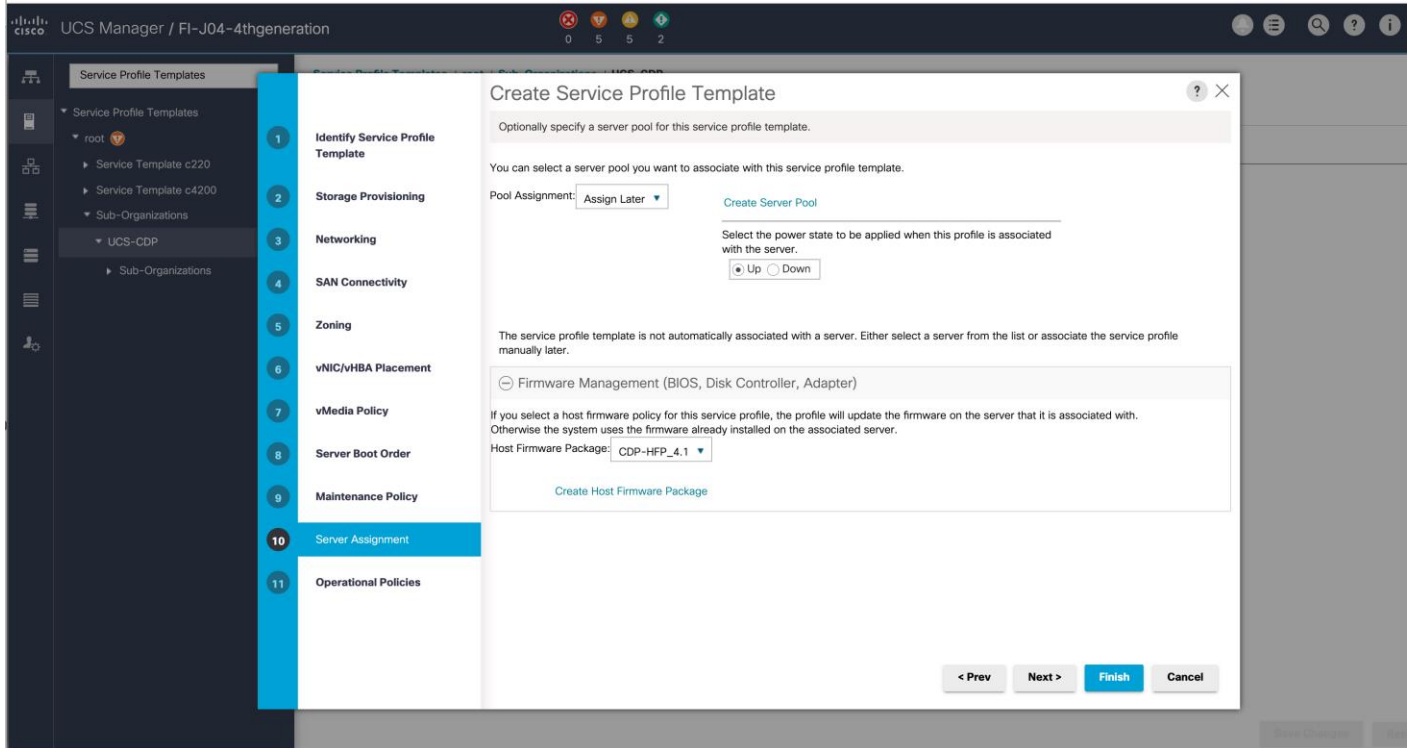
11. Click Next in the vMedia Policy tab (no change) and click Next.
12. Select Boot Policy in the Server Boot Order tab.



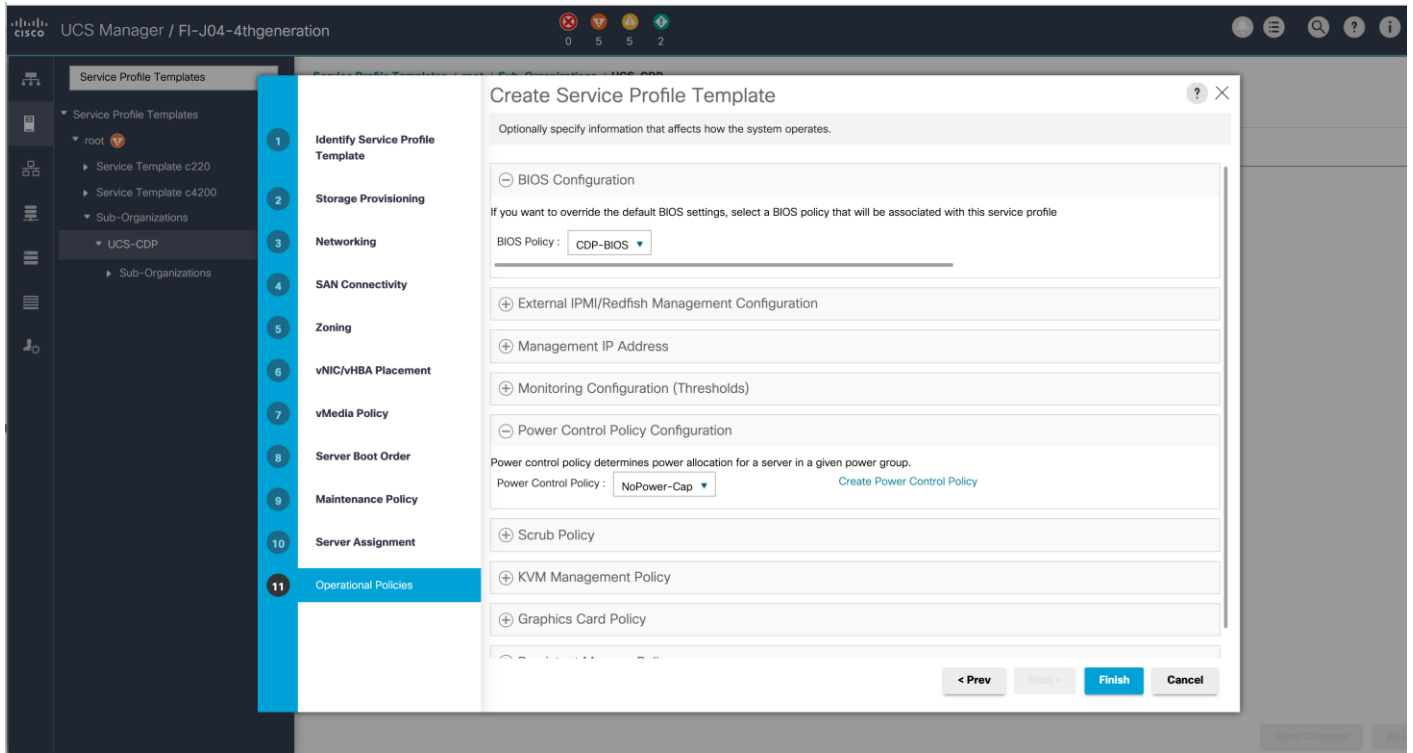
13. Select UserAck maintenance policy, which requires user acknowledgement prior rebooting server when making changes to policy or pool configuration tied to a service profile.



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.



On the Operational Policy page, the BIOS policy for a Cisco UCS C125 M5 Rack server node with the Power Control Policy set to “NoPowerCap” for maximum performance.



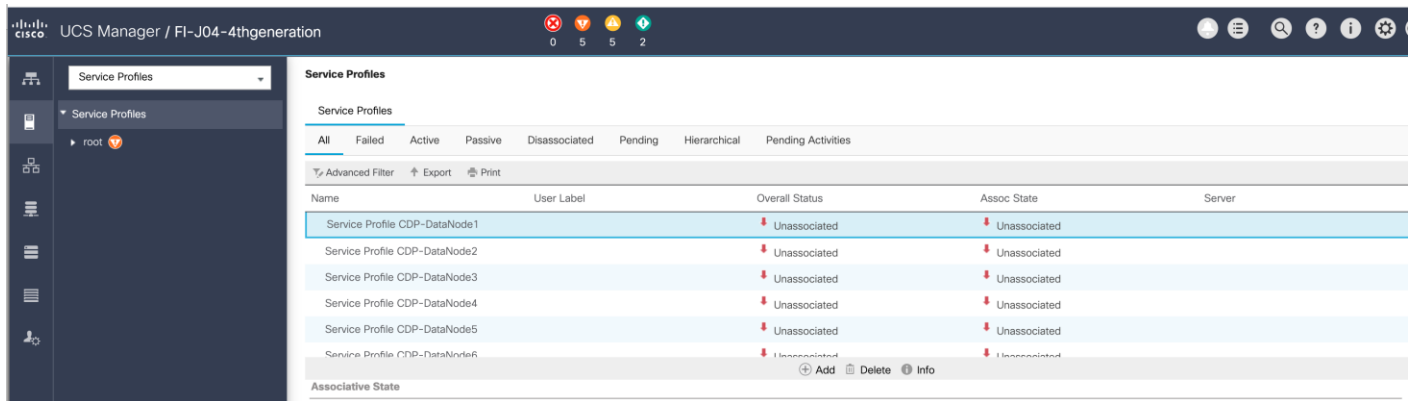
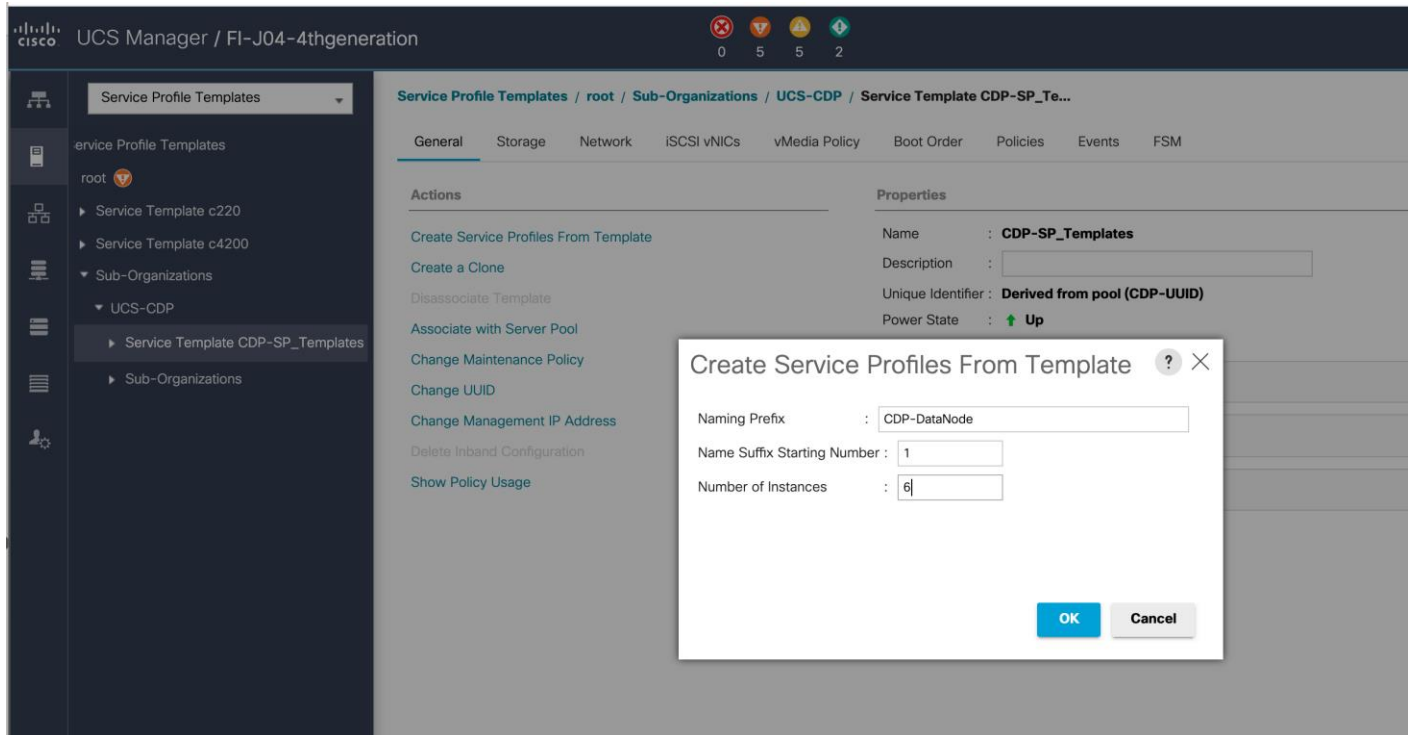
16. Click Finish to create the Service Profile template.

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 46. Create Service Profile from Template



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

- Repeat these steps to create service profile template(s) and service profile(s) according to different deployment scenario.

Install Red Hat Enterprise Linux 7.8

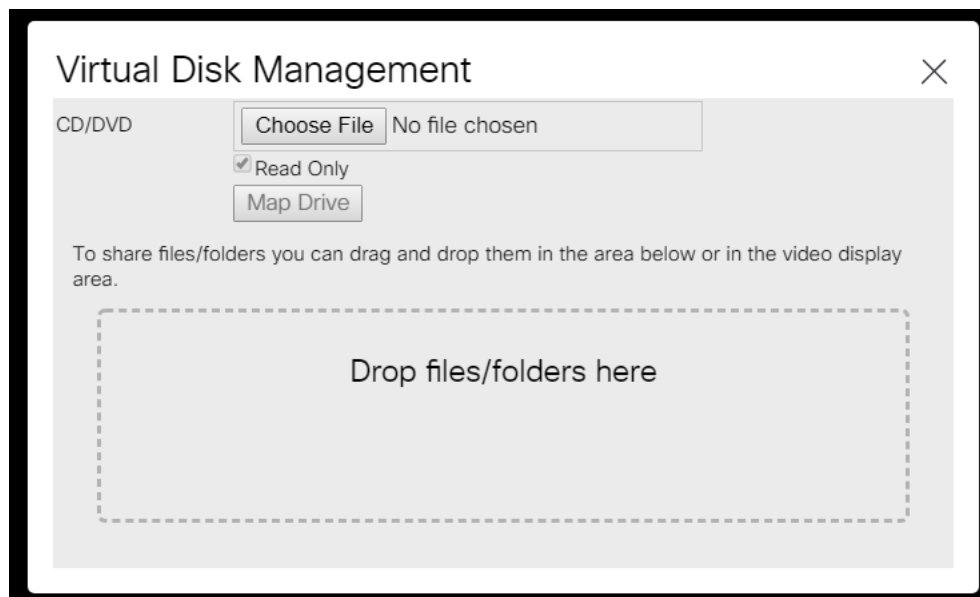
This section provides detailed procedures for installing Red Hat Enterprise Linux Server using Software RAID (OS based Mirroring) on Cisco UCS C125 M5 servers. There are multiple ways to install the RHEL operating system. The installation procedure described in this deployment guide uses KVM console and virtual media from Cisco UCS Manager.

To install the Red Hat Enterprise Linux 7.8 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.
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.



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



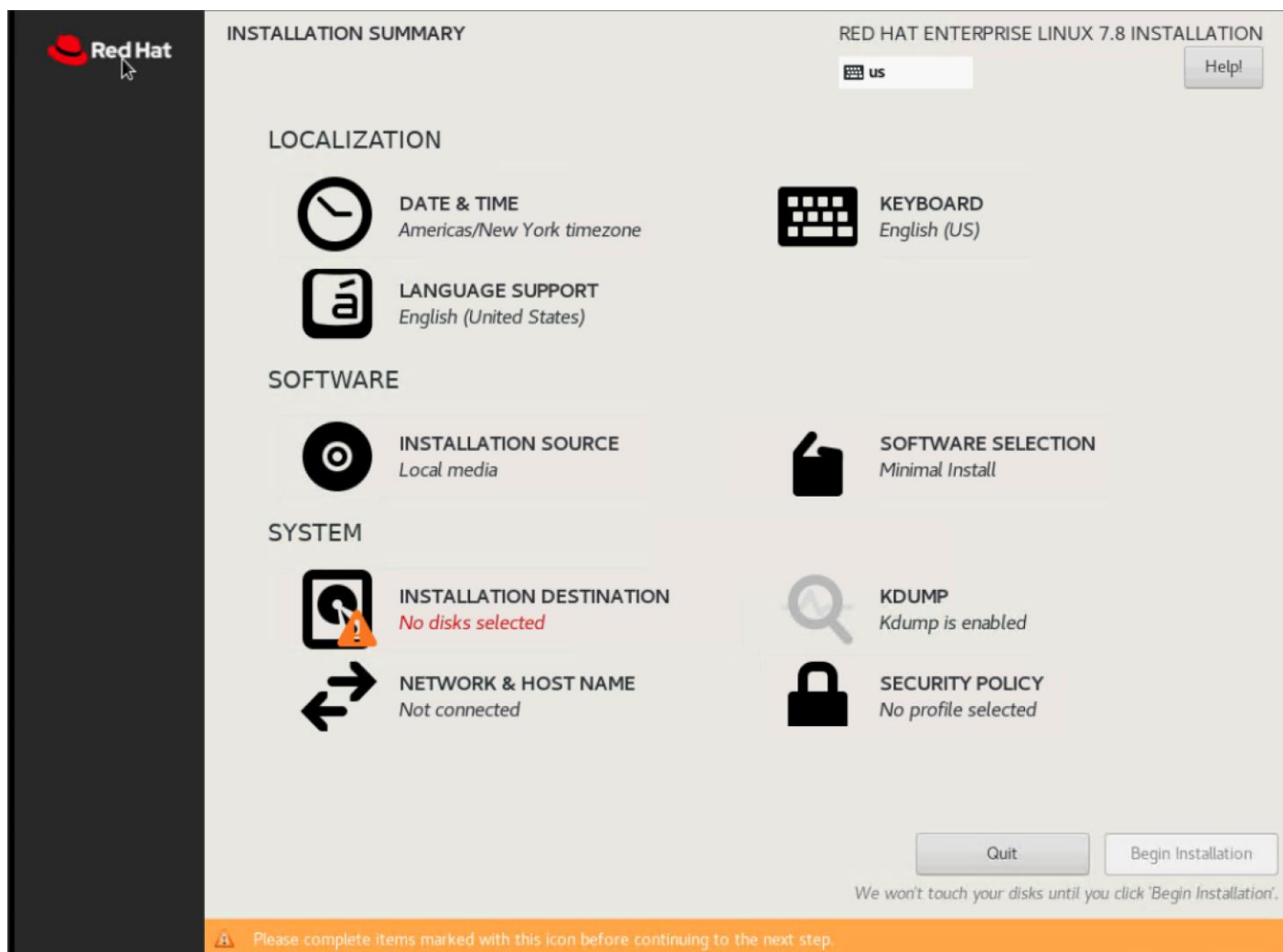
10. Select Map Drive in the Virtual Disk Management windows.
11. Browse to the Red Hat Enterprise Linux 7.8 installer ISO image file.



The Red Hat Enterprise Linux 7.8 Server DVD is assumed to be on the client machine.

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

13. Select the Installation option from Red Hat Enterprise Linux 7.8.
14. Select the language for the installation and click Continue.
15. Select date and time, which pops up another window as shown below.

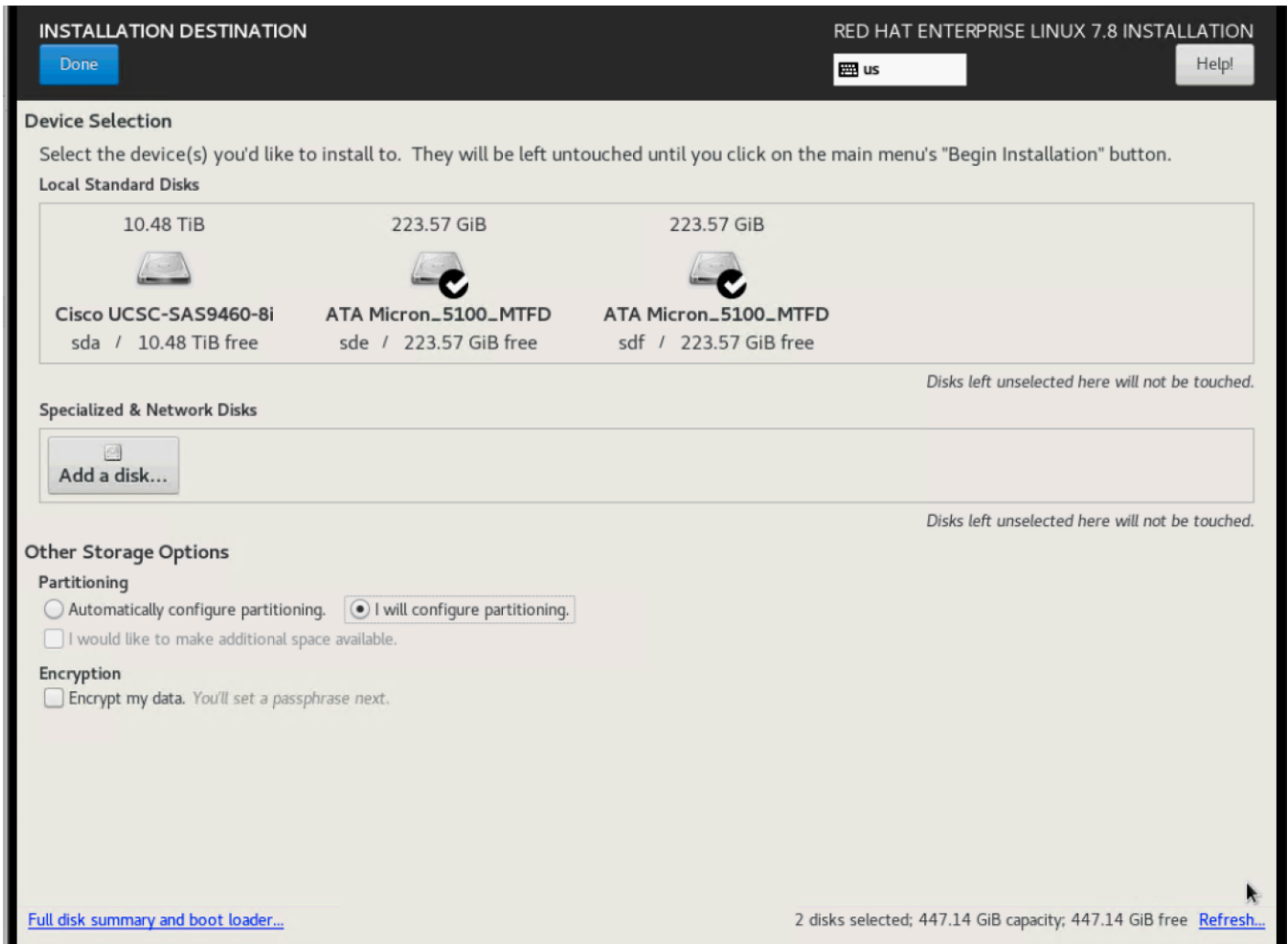


16. Select the location on the map, set the time, and click Done.



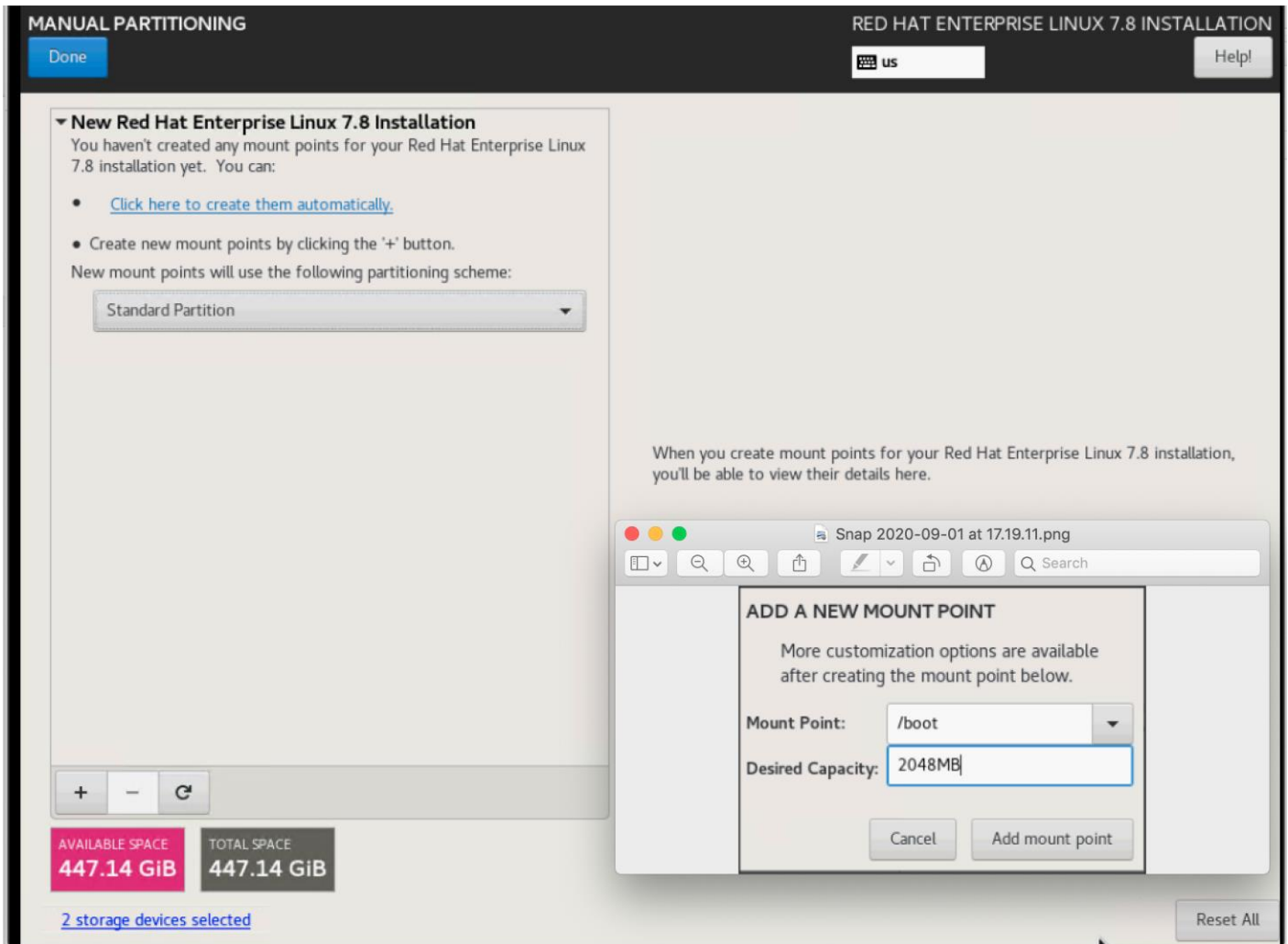
17. Click Installation Destination.

18. This opens a new window with the boot disks. Select a device and choose "I will configure partitioning". Click Done. We selected two M.2 SATA SSDs.



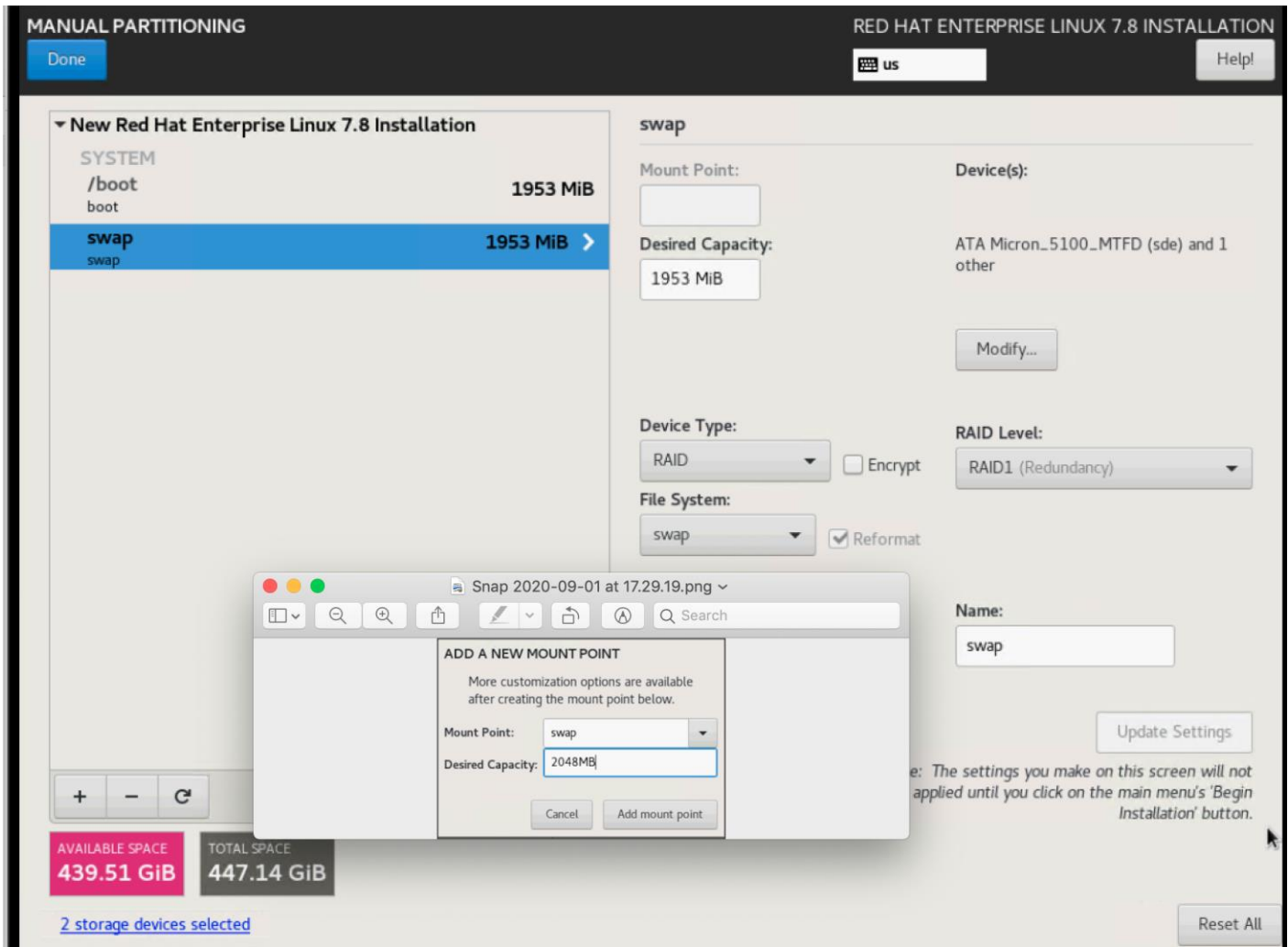
19. This opens a window to create the partitions. Click the + sign to add a new partition as shown below with a boot partition size 2048 MB.

20. Click Add Mount Point to add the partition.



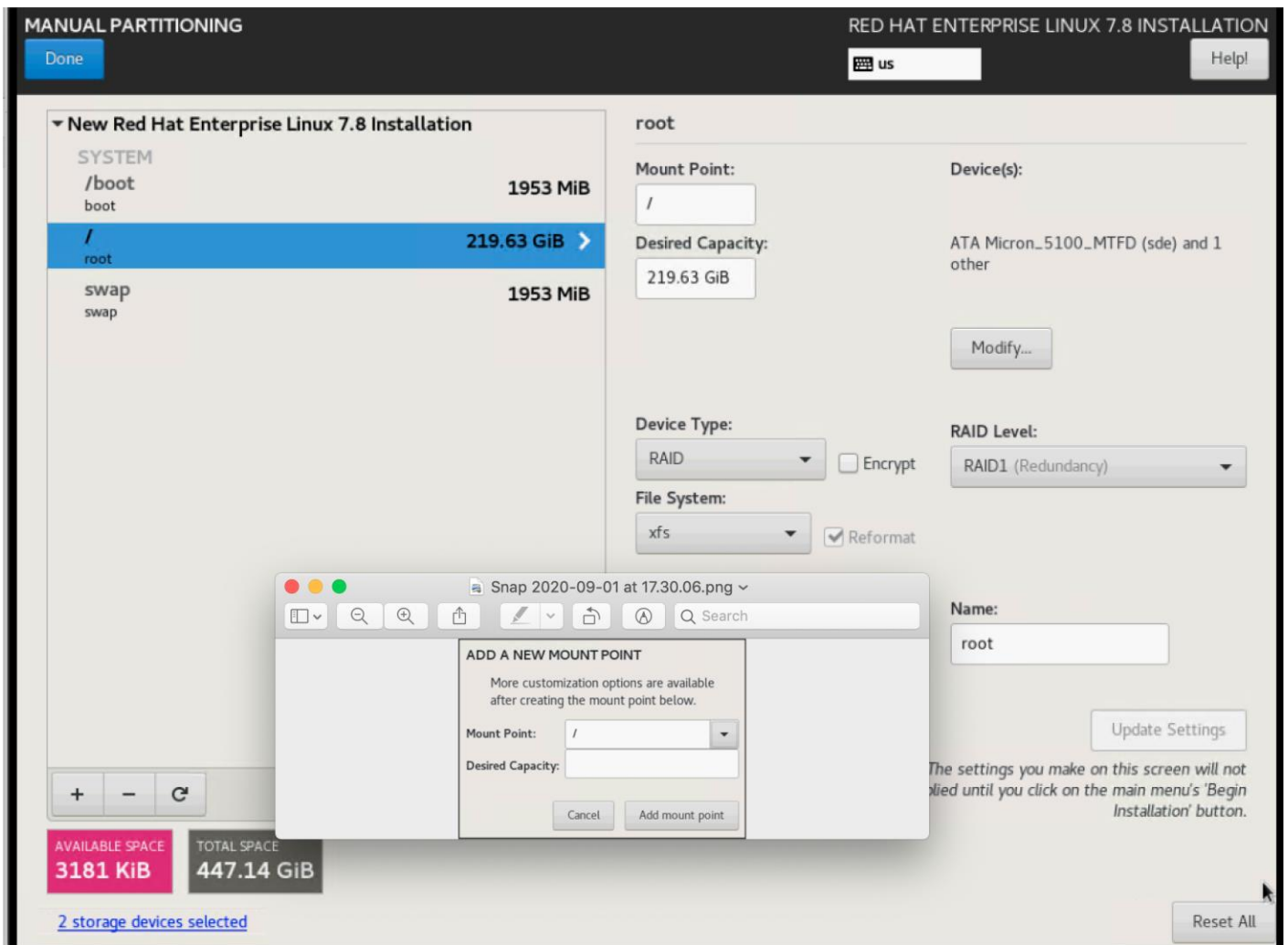
21. Change the device type to RAID and make sure the RAID level is RAID1 (redundancy) and click Update Settings to save the changes.

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



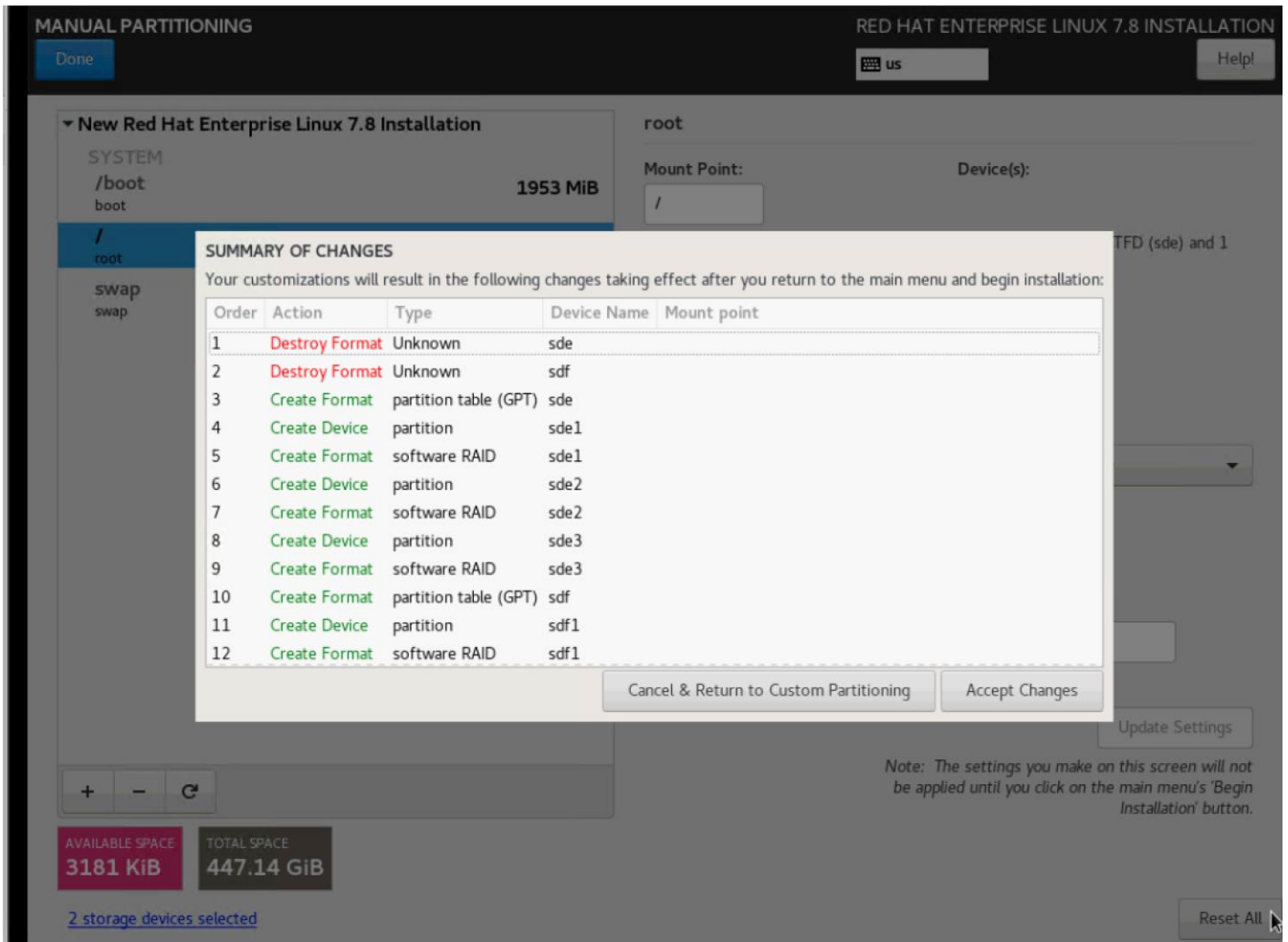
23. Change the Device type to RAID and RAID level to RAID1 (Redundancy) and click Update Settings.

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



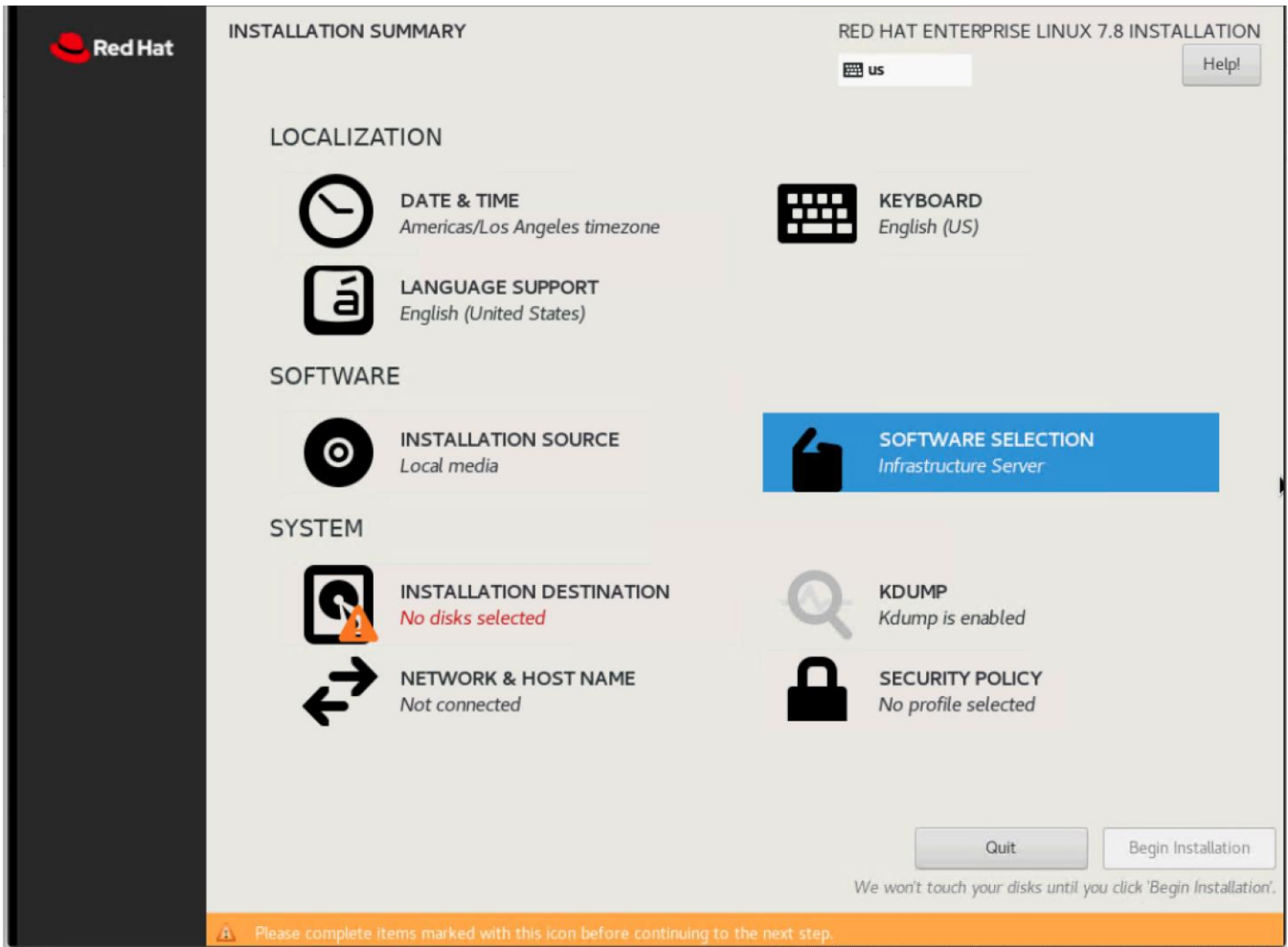
25. Change the Device type to RAID and RAID level to RAID1 (Redundancy). Click Update Settings

26. Click Accept Changes.



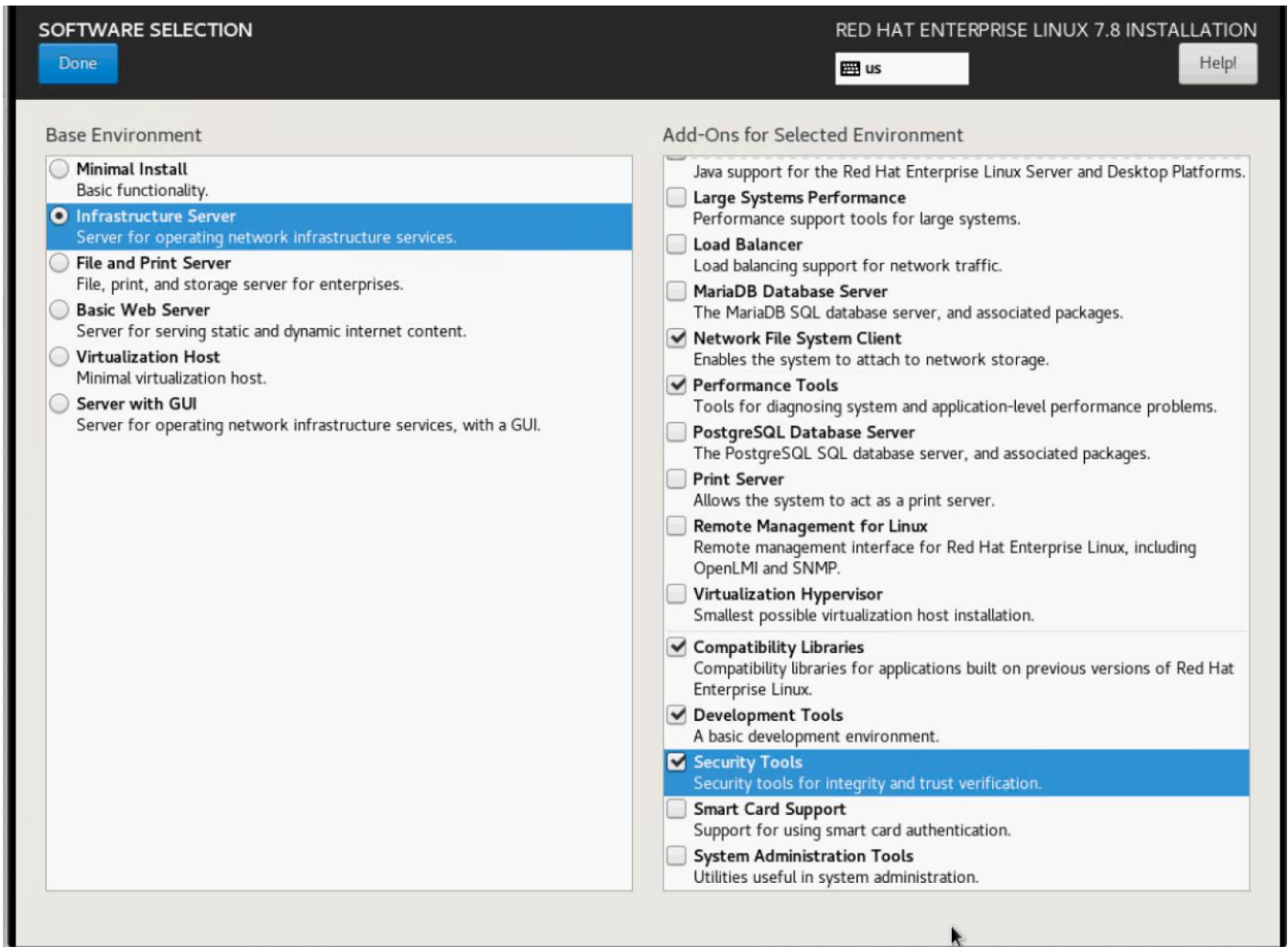
27. Click Done to return to the main screen and continue the Installation.

28. Click Software Selection.

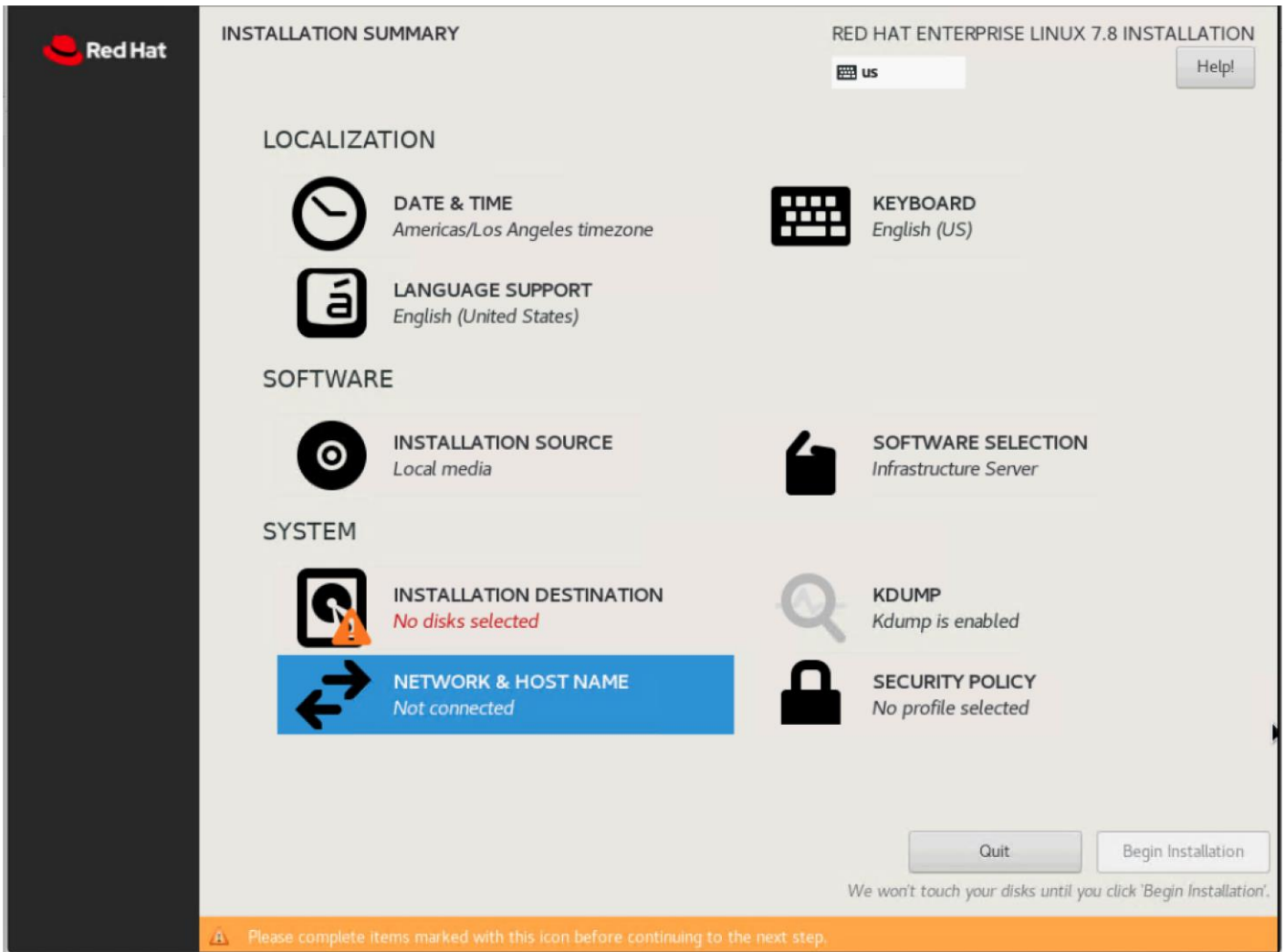


29. Select Infrastructure Server and select the Add-Ons as noted below, then click Done:

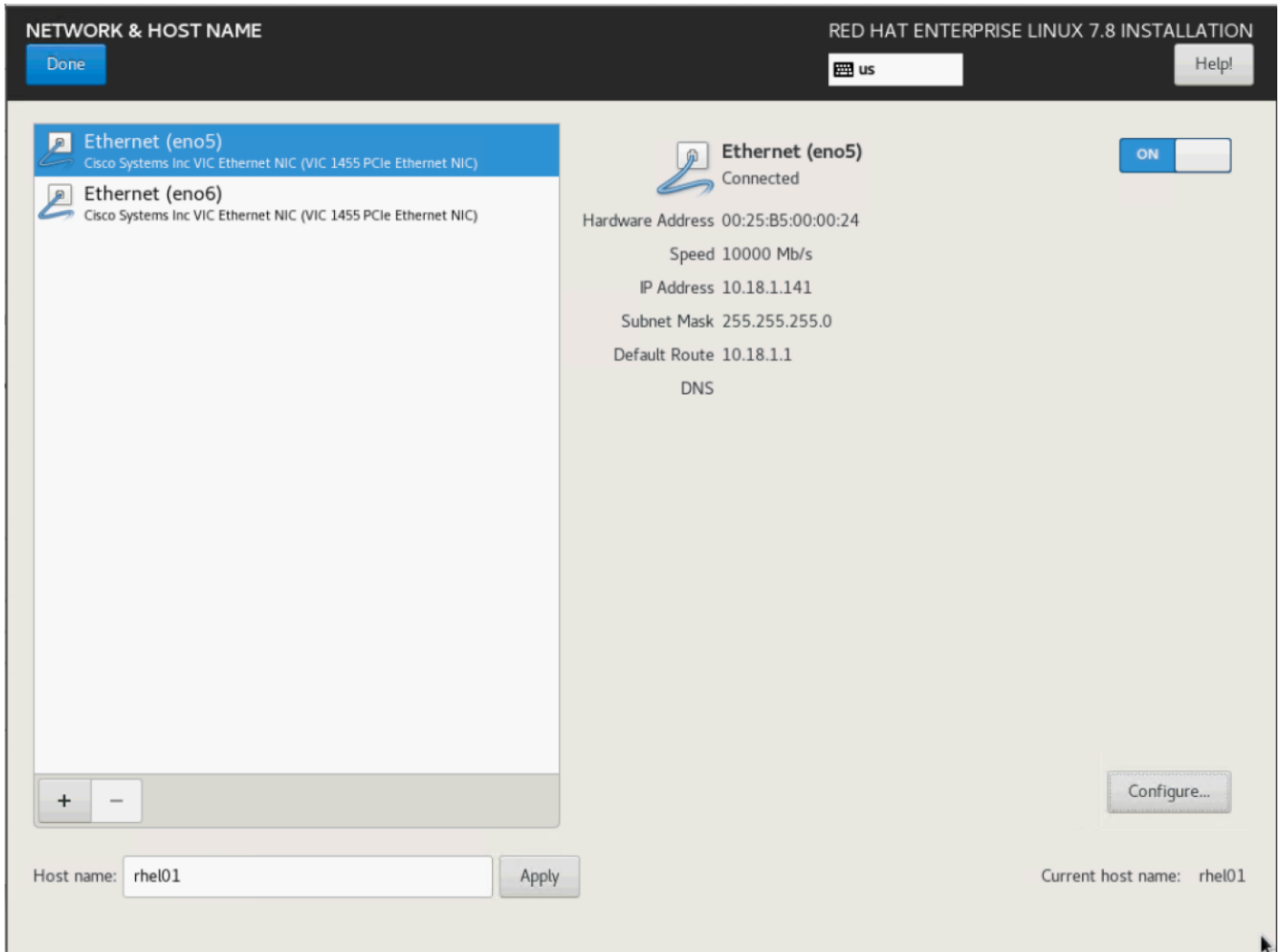
- a. Network File System Client
- b. Performance Tools
- c. Compatibility Libraries
- d. Development Tools
- e. Security Tools



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



31. Type in the hostname as shown below.

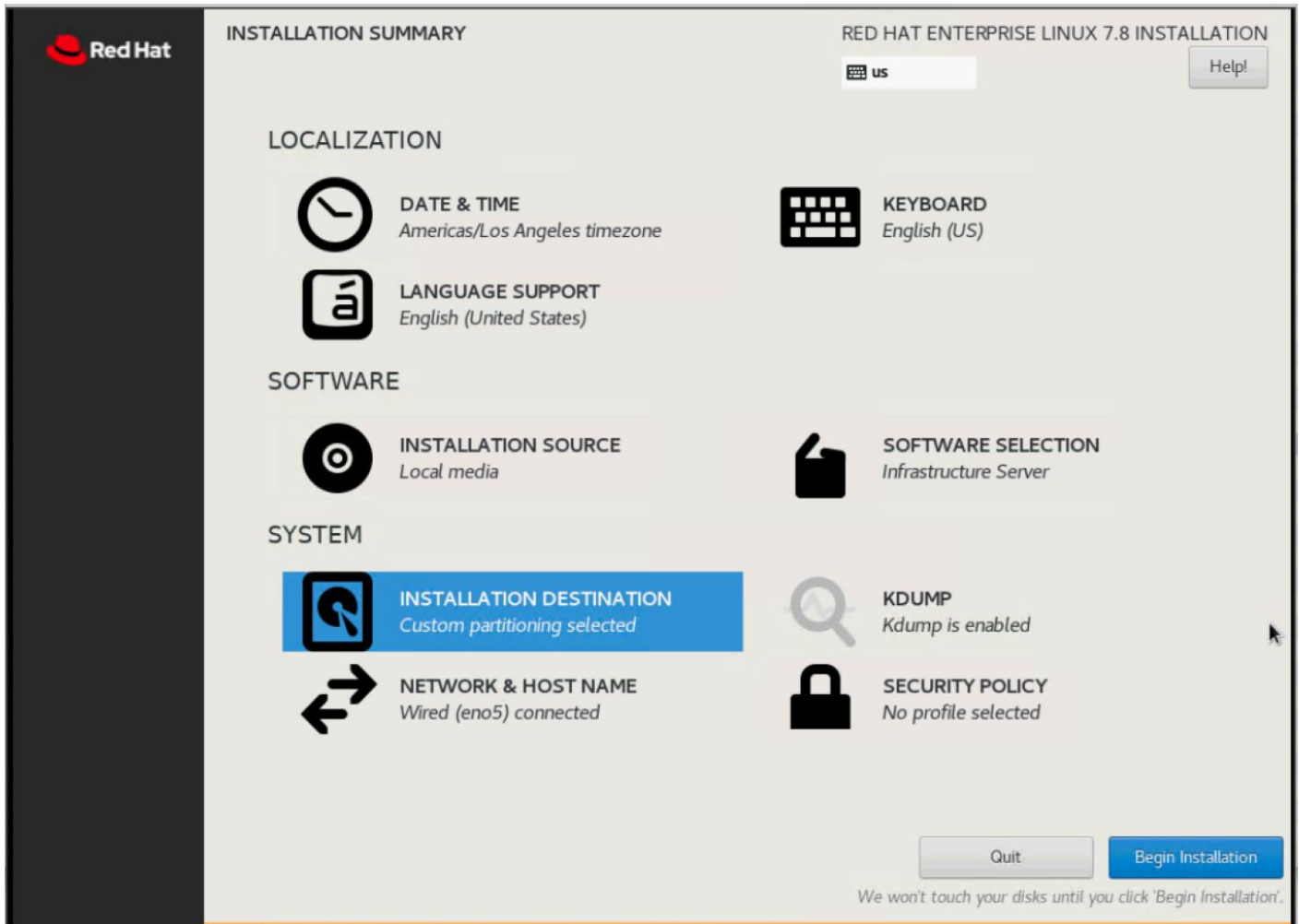


32. Click Configure to open the Network Connectivity window. Click IPv4 Settings.

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

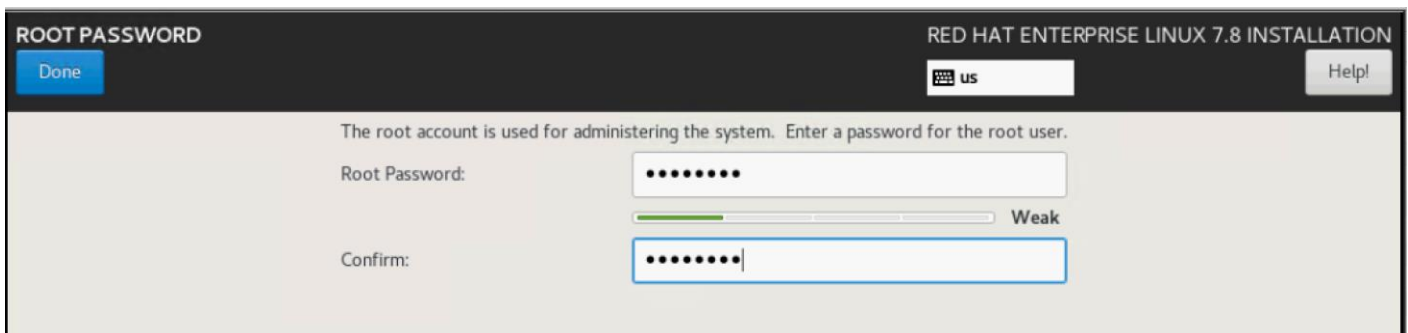
34. Click Save, update the hostname, and turn Ethernet ON. Click Done to return to the main menu.

35. Click Begin Installation in the main menu.

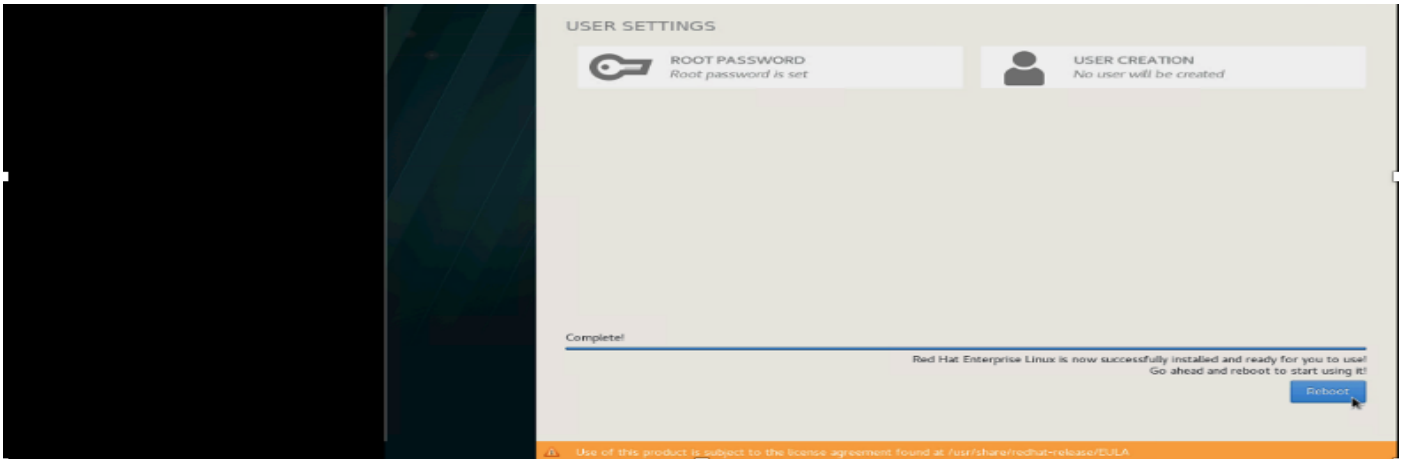


36. Select Root Password in the User Settings.

37. Enter the Root Password and click Done.



38. Once the installation is complete, reboot the system.



39. Repeat steps 1 to 38 to install Red Hat Enterprise Linux 7.8 on rest of the Server Nodes (2-48).



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



Go to the Appendix, section [Configure Cisco Boot Optimized M.2 RAID Controller](#) for Installation steps for Cisco Boot Optimized M.2 RAID Controller.

The hostnames and their corresponding IP addresses are shown in [Table 4](#).

Table 4. Hostname and IP address

| Hostname | Eth0 |
|----------|-------------|
| rhel1 | 10.18.1.131 |
| rhel2 | 10.18.1.132 |
| rhel3 | 10.18.1.133 |
| rhel4 | 10.18.1.134 |
| rhel5 | 10.18.1.135 |
| | |
| rhel47 | 10.18.1.177 |
| rhel48 | 10.18.1.178 |



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 CDP PVC Base installation, Ansible, creating a local Red Hat repo, and others. In this document, we used rhel1 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 cdp.cisco.local
rhel1  A 10.18.1.131
rhel2  A 10.18.1.132
rhel3  A 10.18.1.133
...
...
rhel47 A 10.18.1.177
rhel48 A 10.18.1.178
```

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

1. Log into the Admin Node (rhel1).

```
#ssh 10.18.1.131
```

2. Populate the host file with IP addresses and corresponding hostnames on the Admin node (rhel1) and other nodes as follows:

3. On Admin Node (rhel1):

```
[root@rhel1 ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.18.1.131 rhel1.cdp.cisco.local
10.18.1.132 rhel2.cdp.cisco.local
10.18.1.133 rhel3.cdp.cisco.local
10.18.1.134 rhel4.cdp.cisco.local
10.18.1.135 rhel5.cdp.cisco.local
10.18.1.136 rhel6.cdp.cisco.local
10.18.1.137 rhel7.cdp.cisco.local
10.18.1.138 rhel8.cdp.cisco.local
```

```
10.18.1.139 rhel9.cdp.cisco.local
10.18.1.140 rhel10.cdp.cisco.local
10.18.1.141 rhel11.cdp.cisco.local
10.18.1.142 rhel12.cdp.cisco.local
10.18.1.143 rhel13.cdp.cisco.local
10.18.1.144 rhel14.cdp.cisco.local
10.18.1.145 rhel15.cdp.cisco.local
10.18.1.146 rhel16.cdp.cisco.local
10.18.1.147 rhel17.cdp.cisco.local
10.18.1.148 rhel18.cdp.cisco.local
10.18.1.149 rhel19.cdp.cisco.local
.
.
.
10.18.1.177 rhel47.cdp.cisco.local
10.18.1.178 rhel48.cdp.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 (rhel1).

```
#ssh 10.18.1.131
```

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 47. ssh-keygen

```
[root@rhel1 ansible]# ssh-keygen -N '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
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:j+IdDaaxUBH2ciy/c4M0YcDPHgOoRWrsb8NGnaaj28s root@rhel1
The key's randomart image is:
+---[RSA 2048]-----+
|  ...  .=.  |
| ..o  ...=  |
| .=  .++*   |
| +   ..+*+. |
| .  ..+.oS  |
| + o. * O   |
|   O  + * =  |
| * o. o + .  |
| o.E. . .   |
+-----[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..48}; do echo "copying rhel$i.cdp.cisco.local"; ssh-copy-id -i ~/.ssh/id_rsa.pub
root@rhel$i.cdp.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.

Create a Red Hat Enterprise Linux (RHEL) 7.8 Local Repository

To create a repository using RHEL DVD or ISO on the admin node (in this deployment rhel1 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 7.8 local repository, follow these steps:

1. Log into rhel1. 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 rhel1.
4. Log back into rhel1 and create the mount directory.

```
# scp rhel-server-7.8-x86_64-dvd.iso rhel1:/root/
```



```
# mkdir -p /mnt/rheliso
# mount -t iso9660 -o loop /root/rhel-server-7.8-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 rhel1 create a .repo file to enable the use of the yum command.

```
# vi /var/www/html/rhelrepo/rheliso.repo
[rhel7.8]
name=Red Hat Enterprise Linux 7.8
baseurl=http://10.18.1.131/rhelrepo
gpgcheck=0
enabled=1
```

7. Copy rheliso.repo file from /var/www/html/rhelrepo to /etc/yum.repos.d on rhel1.

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



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

8. To make use of repository files on rhel1 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 (rhel1) using the repo (such as httpd, create-repo, and so on.)

```
# vi /etc/yum.repos.d/rheliso.repo
[rhel7.8]
name=Red Hat Enterprise Linux 7.8
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 (rhel1). 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 .
```

```
[root@rhel1 rhelrepo]# createrepo .
Spawning worker 0 with 3763 pkgs
Workers Finished
Gathering worker results

Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete
```

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@rhel1 ~]# cat /etc/ansible/hosts
[admin]
rhel1.cdp.cisco.local

[namenodes]
rhel1.cdp.cisco.local
rhel2.cdp.cisco.local
rhel3.cdp.cisco.local

[datanodes]
rhel4.cdp.cisco.local
rhel5.cdp.cisco.local
```

```
rhel6.cdp.cisco.local
rhel7.cdp.cisco.local
rhel8.cdp.cisco.local
rhel9.cdp.cisco.local
rhel10.cdp.cisco.local
rhel11.cdp.cisco.local
rhel12.cdp.cisco.local
rhel13.cdp.cisco.local
rhel14.cdp.cisco.local
rhel15.cdp.cisco.local
rhel16.cdp.cisco.local
rhel17.cdp.cisco.local
rhel18.cdp.cisco.local
rhel19.cdp.cisco.local
rhel20.cdp.cisco.local
rhel21.cdp.cisco.local
rhel22.cdp.cisco.local
rhel23.cdp.cisco.local
rhel24.cdp.cisco.local
rhel25.cdp.cisco.local
.
.
Rhel47.cdp.cisco.local
Rhel48.cdp.cisco.local
```

```
[nodes]
rhel1.cdp.cisco.local
rhel2.cdp.cisco.local
rhel3.cdp.cisco.local
rhel4.cdp.cisco.local
rhel5.cdp.cisco.local
rhel6.cdp.cisco.local
rhel7.cdp.cisco.local
rhel8.cdp.cisco.local
rhel9.cdp.cisco.local
rhel10.cdp.cisco.local
rhel11.cdp.cisco.local
rhel12.cdp.cisco.local
rhel13.cdp.cisco.local
rhel14.cdp.cisco.local
rhel15.cdp.cisco.local
rhel16.cdp.cisco.local
rhel17.cdp.cisco.local
rhel18.cdp.cisco.local
rhel19.cdp.cisco.local
rhel20.cdp.cisco.local
rhel21.cdp.cisco.local
rhel22.cdp.cisco.local
rhel23.cdp.cisco.local
rhel24.cdp.cisco.local
rhel25.cdp.cisco.local
.
.
.
Rhel47.cdp.cisco.local
Rhel48.cdp.cisco.local
```

5. Verify host group by running the following commands. Error! Reference source not found. shows the outcome of the ping command.

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

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 10.18.1.131:80
```

3. Start httpd:

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

Disable the Linux Firewall



The default Linux firewall settings are 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.

To disable the Linux firewall, run the following:

```
# 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 nodes 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/
```

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 namenodes -a "sestatus"
rhel1.cdp.cisco.local | CHANGED | rc=0 >>
SELinux status:          disabled

rhel2.cdp.cisco.local | CHANGED | rc=0 >>
SELinux status:          disabled

rhel3.cdp.cisco.local | CHANGED | rc=0 >>
SELinux status:          disabled
```

Upgrade the Cisco Network Driver for VIC1455

The latest Cisco Network driver is required for performance and updates. The latest drivers can be downloaded from this link: [https://software.cisco.com/download/home/283862063/type/283853158/release/4.1\(2a\)](https://software.cisco.com/download/home/283862063/type/283853158/release/4.1(2a))

In the ISO image, the required driver `kmod-enic-4.0.0.11-802.43.rhel17u8.x86_64.rpm` can be located at `\Network\Cisco\VIC\RHEL\RHEL7.8\`.

To upgrade the Cisco Network Driver for VIC1455, follow these steps:

From a node connected to the Internet, download, extract and transfer `kmod-enic-.rpm` to `rhel1` (admin node).

To upgrade the Cisco Network driver for VIC1455, follow these steps:

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 `rhel1`:

```
[root@rhell ~]# ansible all -m copy -a "src=/root/kmod-enic-4.0.0.11-802.43.rhel7u8.x86_64.rpm dest=/root/."
```

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

```
[root@rhell ~]# ansible all -m yum -a "name=/root/kmod-enic-4.0.0.11-802.43.rhel7u8.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@rhell ~]# 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: \RHEL7.8\kmod-megaraid_sas-07.710.06.00_el7.8-1.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 rhel1:

```
[root@rhell ~]# ansible all -m copy -a "src=/root/kmod-megaraid_sas-07.710.06.00_el7.8-1.x86_64.rpm
dest=/root/."
```

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

```
[root@rhell ~]# ansible all -m yum -a "name=/root/kmod-megaraid_sas-07.710.06.00_el7.8-1.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@rhell ~]# ansible all -m shell -a "modinfo megaraid_sas | head -5"
```

Set Up JAVA

To setup JAVA, follow these steps:



CDP Private Cloud Base requires JAVA 8.

1. Download `jdk-8u241-linux-x64.rpm` and src the rpm to admin node (rhel1) from this 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-8u241-linux-x64.rpm dest=/root/."
```

3. Extract and Install JDK all nodes:

```
# ansible all -m command -a "rpm -ivh jdk-8u241-linux-x64.rpm"
```

4. Create the following files `java-set-alternatives.sh` and `java-home.sh` on admin node (rhel1):

```
# 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 $item /usr/java/jdk1.8.0_241-amd64/bin/$item 9
  alternatives --set $item /usr/java/jdk1.8.0_241-amd64/bin/$item
done

# vi java-home.sh
export JAVA_HOME=/usr/java/jdk1.8.0_241-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@rhell ~]# 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"
rhell1.cdp.cisco.local | CHANGED | rc=0 >>
java - status is manual.
link currently points to /usr/java/jdk1.8.0_241-amd64/bin/java

rhel4.cdp.cisco.local | CHANGED | rc=0 >>
java - status is manual.
link currently points to /usr/java/jdk1.8.0_241-amd64/bin/java

rhel5.cdp.cisco.local | CHANGED | rc=0 >>
java - status is manual.
link currently points to /usr/java/jdk1.8.0_241-amd64/bin/java
```

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"
rhel19.cdp.cisco.local | CHANGED | rc=0 >>
/usr/java/jdk1.8.0_241-amd64
```

11. Display current java -version.

```
# ansible all -m command -a "java -version"
rhel20.cdp.cisco.local | CHANGED | rc=0 >>
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b11, 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. To set the ulimit on Redhat, edit “`/etc/security/limits.conf`” on admin node `rhel1` and add the following lines:

```
# vi /etc/security/limits.conf
root soft nofile 64000
root hard nofile 64000
```

2. Copy the `/etc/security/limits.conf` file from admin node (`rhel1`) 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:

```
# cat /etc/pam.d/su
#%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.

Set TCP Retries

Adjusting the `tcp_retries` parameter for the system network enables faster detection of failed nodes. Given the advanced networking features of Cisco 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 `rhel1` and add the following lines:

```
net.ipv4.tcp_retries2=5
Copy the /etc/sysctl.conf file from admin node (rhel1) 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=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"
```

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"
rhel28.cdp.cisco.local | CHANGED | rc=0 >>
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
```

```

# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.tcp_retries2=5
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
vm.swappiness = 0
vm.overcommit_memory = 0

#Network Tunings
net.ipv4.conf.default.rp_filter=1
net.ipv4.tcp_timestamps=0
net.ipv4.tcp_sack = 1
net.core.netdev_max_backlog = 25000
net.core.rmem_max = 2147483647
net.core.wmem_max = 2147483647
net.core.rmem_default = 33554431
net.core.wmem_default = 33554432
net.core.optmem_max = 33554432
net.ipv4.tcp_rmem =8192 33554432 2147483647
net.ipv4.tcp_wmem =8192 33554432 2147483647
net.ipv4.tcp_low_latency=1
net.ipv4.tcp_adv_win_scale=1
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1

net.ipv4.conf.all.arp_filter=1
net.ipv4.tcp_retries2=5
net.ipv6.conf.lo.disable_ipv6 = 1
net.core.somaxconn = 65535

#memory cache settings
vm.swappiness=1
vm.overcommit_memory=0
vm.dirty_background_ratio=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.d/rc.local`:

```
# ansible nodes -m shell -a "cat /root/thp_disable >> /etc/rc.d/rc.local"
# ansible nodes -m shell -a "chmod +x /etc/rc.d/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 (`rhel1`). 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.

To configure NTP, follow these steps:

```
# 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 10.18.1.131
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 (`rhel1`):

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

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

```
# ansible all -m service -a "name=ntpd state=stopped"
# ansible all -m command -a "ntpdate rhell.cdp.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"
```



Alternatively, the new Chrony service can be installed, which 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
```

10. 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 rhell:/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](#).

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 in section [Delete Partitions](#).

```
#!/bin/bash
#disks_count=`lsblk -id | grep sd | wc -l`
#if [ $disks_count -eq 24 ]; then
# echo "Found 24 disks"
#else
# echo "Found $disks_count disks. Expecting 24. Exiting.."
# exit 1
#fi
[[ "-x" == "${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
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}"
/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 -U ${UUID} /data/disk${count}
(( $? )) && continue
echo "UUID=${UUID} /data/disk${count} xfs inode64,noatime 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 (rhel1).

```
#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 Data Platform Private Cloud Base

This section provides instructions for installing Cloudera software, including Cloudera Manager, Cloudera Runtime, and other managed services, in a production environment.

Review the [Cloudera Production Installation: Before You Install](#) steps prior to the production installation of Cloudera Manager, Cloudera Runtime, and other managed services, review the Cloudera Data Platform 7 Requirements and Supported Versions, in addition to the Cloudera Data Platform Release Notes.

Prerequisites for CDP PVC Base Installation

This section details the prerequisites for the CDP Private Cloud Base installation, such as setting up Cloudera Repo.

Cloudera Manager Repository

To setup the Cloudera Manager Repository, follow these steps:

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/cloudera-repos/
```

2. Download Cloudera Manager Repository:

```
#cd /tmp/cloudera-repos/  
# wget https://archive.cloudera.com/cm7/7.1.3/redhat7/yum/cloudera-manager.repo  
# reposync --config=./cloudera-manager.repo --repoid=cloudera-manager  
# wget https://archive.cloudera.com/cm7/7.1.3/allkeys.asc
```



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/cloudera-repos/ rhel1:/var/www/html/  
# scp allkeys.asc rhel1:/var/www/html/cloudera-repos/cm7/
```

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

```
#cd /var/www/html/cloudera-repos/  
#createrepo --baseurl http://10.18.1.140/cloudera-repos/cm7/ /var/www/html/cloudera-repos/cm7/
```



Go to: <http://10.18.1.131/cloudera-repos/cm7/> to verify the files.

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

```
# vi /var/www/html/cloudera-repos/cm7/cloudera-repo.repo  
# cat /var/www/html/cloudera-repos/cm7/cloudera-repo.repo  
[cloudera-repo]  
name=Cloudera Manager 7.1.3  
baseurl=http://10.18.1.140/cloudera-repos/cm7/  
gpgcheck=0  
enabled=1
```

7. Copy the file `cloudera-repo.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/cloudera-repos/cm7/cloudera-repo.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-repo.repo dest=/etc/yum.repos.d/."
```

Set Up the Local Parcels for CDP PVC Base 7.1.3

From a host connected the internet, download CDP PVC Base 7.1.3 parcels that are meant for RHEL7.8 from the URL: <https://archive.cloudera.com/cdh7/7.1.3.0/parcels/> and place them in the directory `/var/www/html/cloudera-repos/` of the Admin node.

The following are the required files for RHEL7.8:

- [CDH-7.1.3-1.cdh7.1.3.p0.4992530-el7.parcel](#)
- [CDH-7.1.3-1.cdh7.1.3.p0.4992530-el7.parcel.sha256](#)
- `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/cloudera-repos/CDH7.1.3.0parcels
```

2. Download parcels:

```
#cd /tmp/cloudera-repos/CDH7.1.3.0parcels
# wget https://archive.cloudera.com/cdh7/7.1.3.0/parcels/CDH-7.1.3-1.cdh7.1.3.p0.4992530-el7.parcel
# wget https://archive.cloudera.com/cdh7/7.1.3.0/parcels/CDH-7.1.3-1.cdh7.1.3.p0.4992530-el7.parcel.sha256
# wget https://archive.cloudera.com/cdh7/7.1.3.0/parcels/manifest.json
```

3. Copy `/tmp/cloudera-repos/CDH7.1.3.0parcels` to the admin node (rhel1):

```
# scp -r /tmp/cloudera-repos/CDH7.1.3.0parcels rhel1:/var/www/html/cloudera-repos/
# chmod -R ugo+rX /var/www/html/cloudera-repos/cdh7.1.3.0parcels
```

4. Verify that these files are accessible by visiting the URL <http://10.18.1.140/cloudera-repos/cdh7.1.3.0parcels/> in admin node.

5. Download Sqoop Connectors.

```
# mkdir -p /tmp/cloudera-repos/sqoop-connectors
# wget --recursive --no-parent --no-host-directories https://archive.cloudera.com/sqoop-connectors/parcels/latest/ -P /tmp/cloudera-repos/
```

6. Copy `/tmp/cloudera-repos/sqoop-connectors` to the admin node (rhel1).

```
# scp -r /tmp/cloudera-repos/sqoop-connectors rhel1:/var/www/html/cloudera-repos/
# sudo chmod -R ugo+rX /var/www/html/cloudera-repos/sqoop-connectors
```

Install and Configure Database for Cloudera Manager

You will set up the following for Cloudera Manager:

- Install the PostgreSQL Server

- Installing the psycopg2 Python Package
- Configure and Start the PostgreSQL Server

Install PostgreSQL Server

To Install the PostgreSQL packages on the PostgreSQL server, follow these steps:

1. In the admin node where Cloudera Manager will be installed, use the following command to install PostgreSQL server.

```
#yum -y install postgresql-server
```

2. Install psycopg2 Python package 2.7.5 or higher if lower version is installed.

```
# yum install -y python-pip
# pip install psycopg2==2.7.5 --ignore-installed
```



Check installing dependencies for hue:

https://docs.cloudera.com/documentation/enterprise/upgrade/topics/ug_cdh_upgrade_hue_psycopg2.html

Configure and Start PostgreSQL Server

To configure and start the PostgreSQL Server, follow these steps:

1. stop PostgreSQL server if it is running:

```
# systemctl stop postgresql.service
```



Take a backup of the existing database.



By default, PostgreSQL only accepts connections on the loopback interface. You must reconfigure PostgreSQL to accept connections from the fully qualified domain names (FQDN) of the hosts hosting the services for which you are configuring databases. If you do not make these changes, the services cannot connect to and use the database on which they depend.

2. Make sure that LC_ALL is set to en_US.UTF-8 and initialize the database as follows::

```
# echo 'LC_ALL="en_US.UTF-8"' >> /etc/locale.conf
# sudo su -l postgres -c "postgresql-setup initdb"
```

3. To enable MD5 authentication, edit `/var/lib/pgsql/data/pg_hba.conf` by adding the following line.

```
# host all all 127.0.0.1/32 md5
```



The host line specifying md5 authentication shown above must be inserted before this ident line: `# host all 127.0.0.1/32 ident`



Failure to do so may cause an authentication error when running the `scm_prepare_database.sh` script. You can modify the contents of the `md5` line shown above to support different configurations. For example, if you want to access PostgreSQL from a different host, replace `127.0.0.1` with your IP address and update `postgresql.conf`, which is typically found in the same place as `pg_hba.conf`, to include: `# listen_addresses = '*'`

4. Configure settings to ensure your system performs as expected. Update these settings in the `/var/lib/pgsql/data/postgresql.conf` file. Settings vary based on cluster size and resources as follows:

```
max_connection - 100
shared_buffers - 1024 MB
wal_buffers - 16 MB
checkpoint_segments - 128
checkpoint_completion_target - 0.9
```



Refer to the Cloudera Data Platform Data Center Installation guide, Configuration and Starting the PostgreSQL Server section for more details: <https://docs.cloudera.com/cdp-private-cloud-base/latest/installation/topics/cdpdc-install-configure-databases.html>

5. Start the PostgreSQL Server and configure to start at boot.

```
# systemctl start postgresql
# systemctl enable postgresql
```

Databases for CDP

Create databases and service accounts for components that require database.

We created databases for the following components:

- Cloudera Manager Server
- Cloudera Management Service Roles: Activity Monitor, Reports Manager, Hive Metastore Server, Data Analytics Studio, Ranger, hue, and oozie.
- The databases must be configured to support the PostgreSQL UTF8 character set encoding.
- Record the values you enter for database names, usernames, and passwords. The Cloudera Manager installation wizard requires this information to correctly connect to these databases.

To create databases for CDP, follow these steps:

1. In the admin node, connect to PostgreSQL:

```
# sudo -u postgres psql
```

2. Create databases using the command below:

```
CREATE ROLE scm LOGIN PASSWORD 'password';
CREATE DATABASE scm OWNER scm ENCODING 'UTF8';

CREATE ROLE amon LOGIN PASSWORD 'password';
CREATE DATABASE amon OWNER amon ENCODING 'UTF8';
```

```
CREATE ROLE rman LOGIN PASSWORD 'password';
CREATE DATABASE rman OWNER rman ENCODING 'UTF8';

CREATE ROLE hue LOGIN PASSWORD 'password';
CREATE DATABASE hue OWNER hue ENCODING 'UTF8';

CREATE ROLE hive LOGIN PASSWORD 'password';
CREATE DATABASE metastore OWNER hive ENCODING 'UTF8';

CREATE ROLE nav LOGIN PASSWORD 'password';
CREATE DATABASE nav OWNER nav ENCODING 'UTF8';

CREATE ROLE navms LOGIN PASSWORD 'password';
CREATE DATABASE navms OWNER navms ENCODING 'UTF8';

CREATE ROLE oozie LOGIN PASSWORD 'password';
CREATE DATABASE oozie OWNER oozie ENCODING 'UTF8';

CREATE ROLE rangeradmin LOGIN PASSWORD 'password';
CREATE DATABASE ranger OWNER rangeradmin ENCODING 'UTF8';

CREATE ROLE das LOGIN PASSWORD 'password';
CREATE DATABASE das OWNER das ENCODING 'UTF8';

ALTER DATABASE metastore SET standard_conforming_strings=off;
ALTER DATABASE oozie SET standard_conforming_strings=off;
```



For detailed information about Apache Ranger specific configuration for PostgreSQL, go to: [Configuring a PostgreSQL Database for Ranger](#)

Cloudera Manager Installation

The following sections describe how to install Cloudera Manager and then using Cloudera Manager to install CDP PVC Base 7.1.3.

Install Cloudera Manager

Cloudera Manager, an end-to-end management application, is used to install and configure CDP Private Cloud Base. During CDP Installation, Cloudera Manager's Wizard will help to install Hadoop services and any other role(s)/service(s) 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 or Open JDK if it is not already installed across all the cluster nodes.
- Assign various services to nodes.
- Start the Hadoop services



For more information about CDP Private Cloud Base, see the [JAVA requirements](#).

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.

```
# ansible nodes -m shell -a "yum install -y java-1.8.0-openjdk-devel"
```



Please see the CDP PVC Base documentation for more information: [Manually Installing OpenJDK and Oracle JDK](#)

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 -y cloudera-manager-agent cloudera-manager-daemons cloudera-manager-server
```

Set Up the Cloudera Manager Server Database

The Cloudera Manager Server Database includes a script that can create and configure a database for itself.

The script can:

- Create the Cloudera Manager Server database configuration file.
- (PostgreSQL) Create and configure a database for Cloudera Manager Server to use.
- (PostgreSQL) Create and configure a user account for Cloudera Manager Server.



For database requirements, see the CDP PVC Base documentation: [Install and Configure Databases](#)

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 postgresql scm scm <password>
# ./scm_prepare_database.sh postgresql amon amon <password>
# ./scm_prepare_database.sh postgresql rman rman <password>
# ./scm_prepare_database.sh postgresql hue hue <password>
# ./scm_prepare_database.sh postgresql metastore hive <password>
# ./scm_prepare_database.sh postgresql oozie oozie<password>
# ./scm_prepare_database.sh postgresql das das <password>
# ./scm_prepare_database.sh postgresql ranger rangeradmin <password>
```

Start the Cloudera Manager Server

To start the Cloudera Manager Server, follow these steps:

1. Start the Cloudera Manager Server:

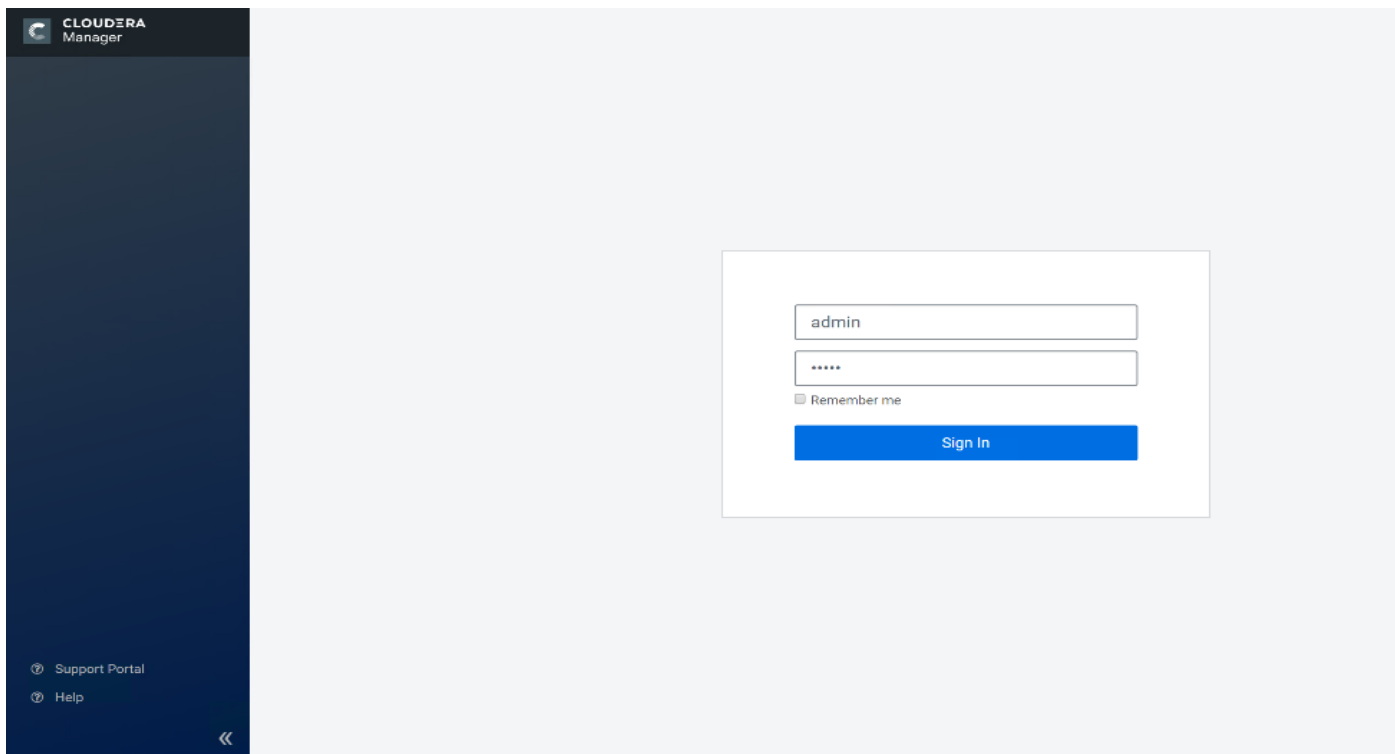
```
#systemctl start cloudera-scm-server
```

2. Access the Cloudera Manager using the URL, <http://10.18.1.140:7180> to verify that the server is up.
3. Once the installation of Cloudera Manager is complete, install CDP-Private Cloud Base 7.1.3 using the Cloudera Manager Web interface.

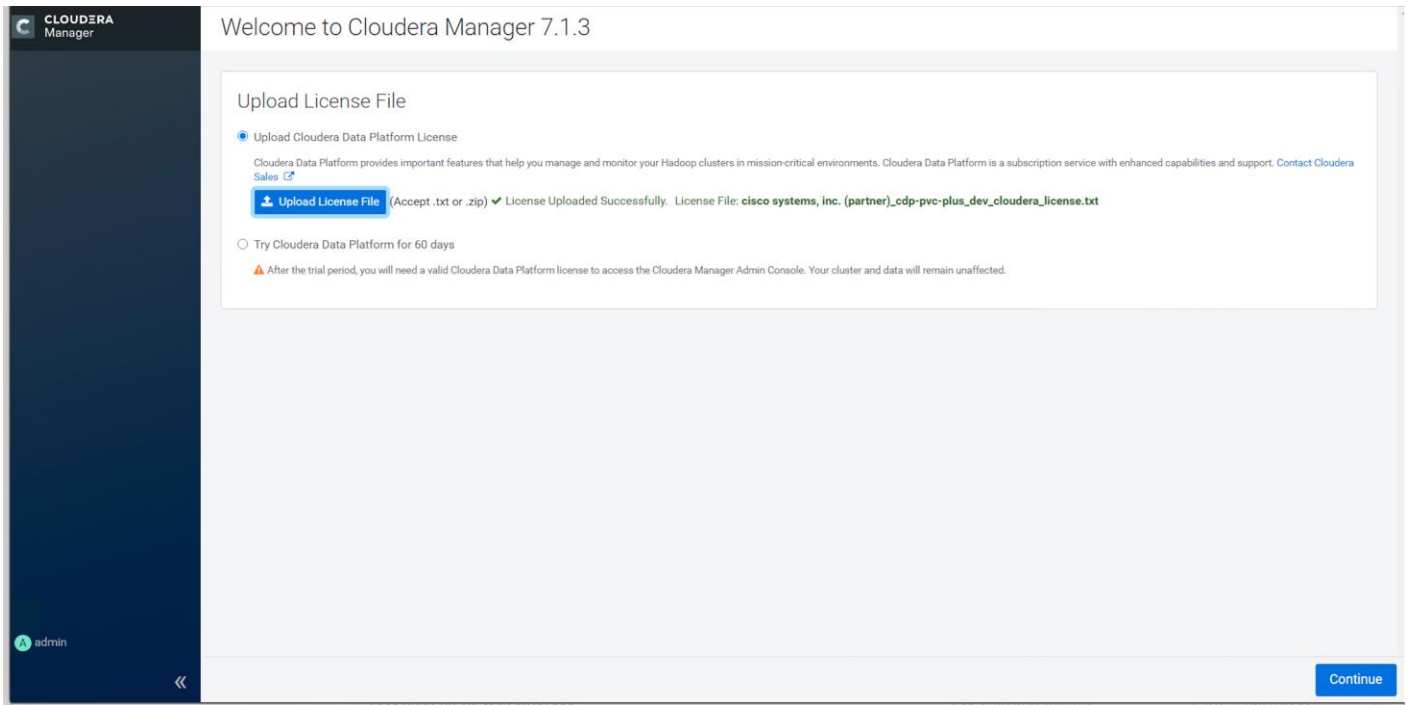
Install Cloudera Data Platform Private Cloud Base (CDP-Private Cloud Base 7.1.3)

To install the Cloudera Data Platform Data Center, follow these steps:

1. Log into the Cloudera Manager. Enter "admin" for both the Username and Password fields.



2. Upload license file. Click Continue after successfully uploading license for CDP Private Cloud Base.



3. Click Continue on the Welcome screen.

CLOUDERA
Manager

Add Cluster - Installation

- 1 Welcome
- 2 Cluster Basics
- 3 Specify Hosts
- 4 Select Repository
- 5 Select JDK
- 6 Enter Login Credentials
- 7 Install Agents
- 8 Install Parcels
- 9 Inspect Cluster

WELCOME

! Auto-TLS is currently not enabled. This means the over-the-wire communication is insecure. Click [here to setup Enable Auto-TLS](#).

! A KDC is currently not configured. This means you cannot create Kerberized clusters. Kerberized clusters are required for the Ranger, Atlas, and Kudu services. Click [here to setup a KDC](#).

Adding a cluster in Cloudera Manager consists of two steps.

- 1 Add a set of hosts to form a cluster and install Cloudera Runtime and the Cloudera Manager Agent software.
- 2 Select and configure the services to run on this cluster.

Quick Links

- [Installation Guide](#)
- [Operating System Requirements](#)
- [Database Requirements](#)
- [JDK Requirements](#)

Parcels
Running Commands
Support
admin

Back Continue

4. Click Continue on Enable Auto-TLS.

CLUSTERA
Manager

Enable Auto-TLS

1 **Generate CA**
2 Remaining Steps

Generate CA

This wizard helps you enable Auto-TLS. Ensure that you have installed the Cloudera Manager Agent package on the Cloudera Manager Server host.

Note: You will need to restart The Cloudera Manager Server, the Cloudera Management service, and all clusters to complete this process.

Trusted CA Certificates Location
The full file path containing a concatenated list of CA certificates in pem format that will be imported into the Auto-TLS truststore and distributed to all hosts. This file should be readable by the 'cloudera-scm' user.

Enable TLS for All existing and future clusters
 Future clusters only
Cloudera Management service, agent to server communication, Cloudera Manager admin console will automatically be enabled with TLS.

Cloudera Manager needs to distribute the certificates to all the hosts over ssh.

SSH Username 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

Password

SSH Port

Cancel ← Back Next →

CLUSTERA
Manager

Enable Auto-TLS

1 **Generate CA**
2 **Remaining Steps**

Remaining Steps

Note: Now you must **restart** the Cloudera Manager server from the command line manually.

```
$ ssh my_cloudera_manager_server_host
$ systemctl restart cloudera-scm-server
$ tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log
```

Wait until the cloudera-manager-server.log shows the message Started Jetty server and then click Continue
Afterwards, you must **restart** the Cloudera Management Service and finally **restart** any clusters that are stale.

Cancel ← Back Finish →

CLOUDERA
Manager

Add Cluster - Installation

- 1 Welcome
- 2 Cluster Basics
- 3 Specify Hosts
- 4 Select Repository
- 5 Select JDK
- 6 Enter Login Credentials
- 7 Install Agents
- 8 Install Parcels
- 9 Inspect Cluster

Parcels
Running Commands
Support
admin

WELCOME

✔ AutoTLS has already been enabled.

⚠ A KDC is currently not configured. This means you cannot create Kerberized clusters. Kerberized clusters are required for Ranger, Atlas, and services that depend on them. Click [here](#) to setup a KDC.

Adding a cluster in Cloudera Manager consists of two steps.

- 1 Add a set of hosts to form a cluster and install Cloudera Runtime and the Cloudera Manager Agent software.
- 2 Select and configure the services to run on this cluster.

Quick Links

- [Installation Guide](#)
- [Operating System Requirements](#)
- [Database Requirements](#)
- [JDK Requirements](#)

Back Continue

5. Enter name for the Cluster.

CLOUDERA
Manager


Add Cluster - Installation

- 1 Welcome
- 2 Cluster Basics
- 3 Specify Hosts
- 4 Select Repository
- 5 Select JDK
- 6 Enter Login Credentials
- 7 Install Agents
- 8 Install Parcels
- 9 Inspect Cluster

Parcels
Running Commands
Support
admin

Cluster Basics

Cluster Name



Regular Cluster

A Regular Cluster contains storage nodes, compute nodes, and other services such as metadata and security collocated in a single cluster.

Back Continue

6. Specify the hosts that are part of the cluster using their IP addresses or hostname.

```
10.18.1.[131-178] or rhel[1-48].cdp.cisco.local
```

7. After the IP addresses or hostnames are entered, click Search.

The screenshot shows the Cloudera Manager interface for adding a cluster. The left sidebar contains a navigation menu with steps: Welcome, Cluster Basics, Specify Hosts (selected), Select Repository, Select JDK, Enter Login Credentials, Install Agents, Install Parcels, and Inspect Cluster. The main content area is titled 'Specify Hosts' and includes a search box for Hostname (containing '10.18.1.[140-147]') and an SSH Port field (containing '22'). Below the search fields, a table displays the results of a scan for 8 hosts, all of which were successfully scanned and are currently not managed. The table has columns for 'Expanded Query', 'Hostname (FQDN)', 'IP Address', 'Currently Managed', and 'Result'. At the bottom right, there are 'Back' and 'Continue' buttons.

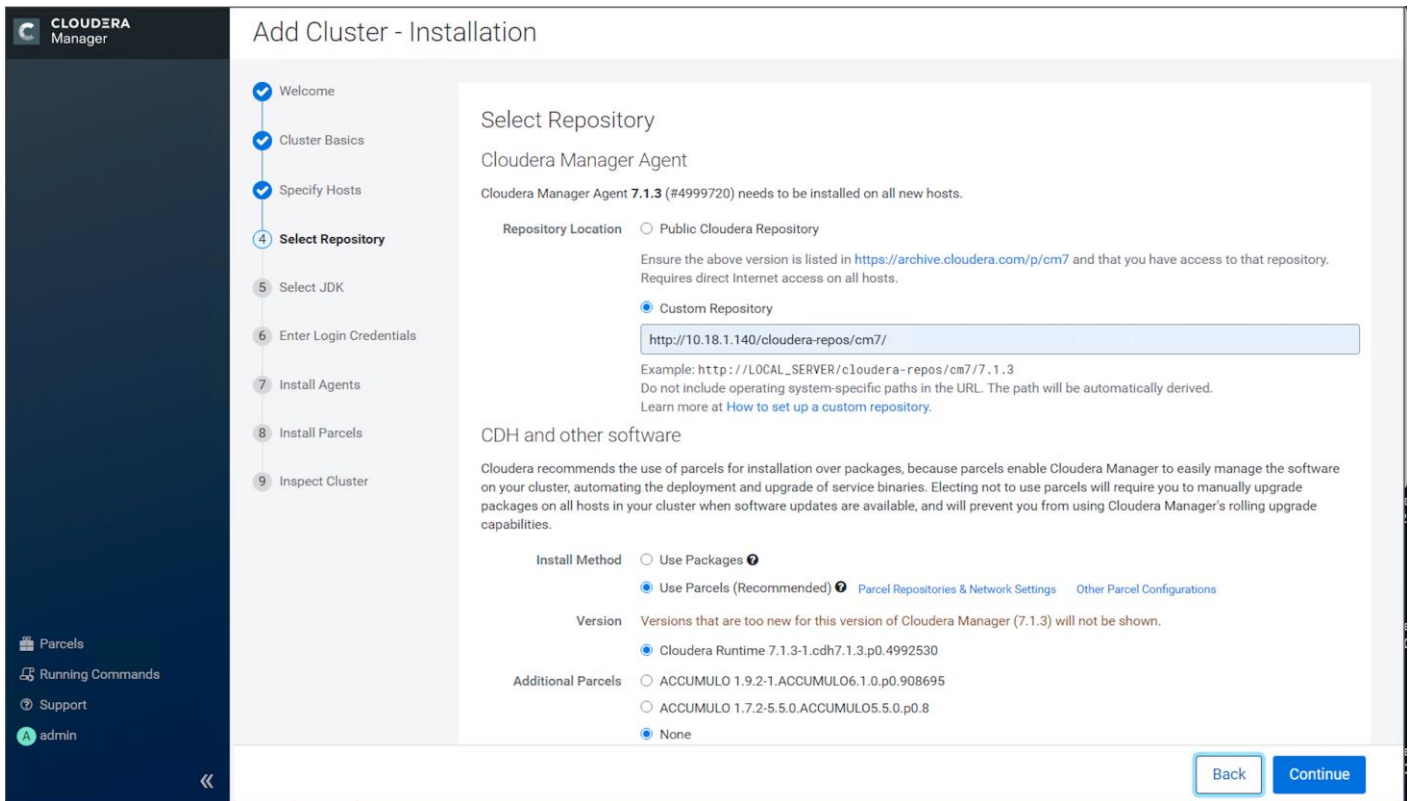
| Expanded Query | Hostname (FQDN) | IP Address | Currently Managed | Result | |
|-------------------------------------|-----------------|------------------------|-------------------|--------|--------------------------------|
| <input checked="" type="checkbox"/> | 10.18.1.140 | rhel10.cdp.cisco.local | 10.18.1.140 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | 10.18.1.141 | rhel11.cdp.cisco.local | 10.18.1.141 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | 10.18.1.142 | rhel12.cdp.cisco.local | 10.18.1.142 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | 10.18.1.143 | rhel13.cdp.cisco.local | 10.18.1.143 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | 10.18.1.144 | rhel14.cdp.cisco.local | 10.18.1.144 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | 10.18.1.145 | rhel15.cdp.cisco.local | 10.18.1.145 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | 10.18.1.146 | rhel16.cdp.cisco.local | 10.18.1.146 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | 10.18.1.147 | rhel17.cdp.cisco.local | 10.18.1.147 | No | Host was successfully scanned. |

8. Cloudera Manager will "discover" the nodes in the cluster. Verify that all desired nodes have been found and selected for installation.

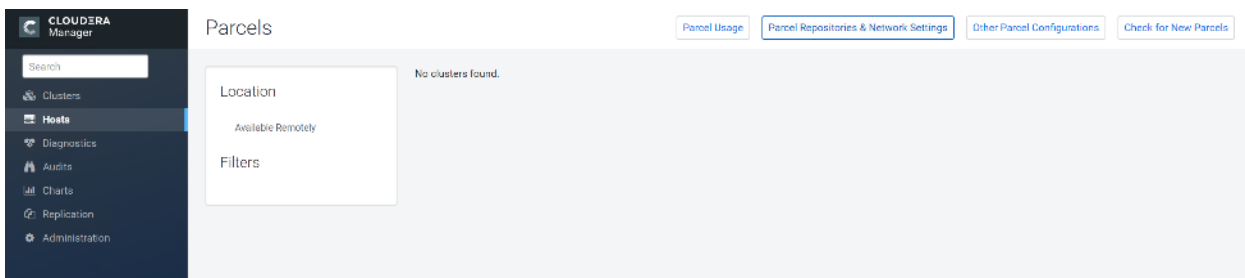
Edit the Cloudera Data Platform Data Center Parcel Settings to Use the CDP 7.1.3 Parcels

To edit the CDP Private Cloud Base Parcel settings, follow these steps:

1. Add custom repository path for Cloudera Manager local repository created.



2. On the Cloudera Manager installation wizard, click Parcels.
3. Click Parcel Repositories and Network Settings.



4. Click to remove the entire remote repository URLs and add the URL to the location where we kept the CDP PVC Base 7.1.3 parcels such as <http://10.18.1.140/cloudera-repos/7.1.3.0parcels/>

Parcel Repository & Network Settings

Cloudera Manager checks the connection to the configured parcel repository URLs. A valid license is required to access most Cloudera parcel repositories. Last Updated: Sep 11, 12:36:28 PM PDT

> 7/7 URL(s) - The repository was successfully accessed and the manifest downloaded and validated. (HTTP Status: 200)

Remote Parcel Repository URLs

Enable Automatic Authentication for Cloudera Repositories

HTTP authentication username override for Cloudera Repositories

HTTP authentication password override for Cloudera Repositories

Proxy Server

Proxy Port

Proxy User

Reason for change: Modified Remote Parcel Repository URLs

5. Click Save Changes to finish the configuration.
6. Click Continue on the confirmation page.
7. For the method of installation, select the Use Parcels (Recommended) radio button.
8. For the CDP PVC Base version, select the Cloudera Runtime 7.1.3-1.cdh7.1.3.p0.4992530-e17- 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://10.18.1.140/cloudera-repos/cm7> and click Continue.
11. Select the appropriate option for JDK.

CLUSTER
CLUSTER MANAGER

Add Cluster - Installation

- Welcome
- Cluster Basics
- Specify Hosts
- Select Repository
- 5 Select JDK**
- 6 Enter Login Credentials
- 7 Install Agents
- 8 Install Parcels
- 9 Inspect Cluster

Parcels
Running Commands
Support
admin

Select JDK

| | |
|------------------------------|-----------------------------------|
| Selected Version | Cloudera Runtime 7.1 |
| Supported JDK Version | OpenJDK 8, 11 or Oracle JDK 8, 11 |

[More details on supported JDK version.](#)

If you plan to use JDK 11, you will need to install it manually on all hosts and then select the **Manually manage JDK** option below.

- Manually manage JDK**
Please ensure that a supported JDK is already installed on all hosts. You will need to manage installing the unlimited strength JCE policy file, if necessary.
- Install a Cloudera-provided version of OpenJDK**
By proceeding, Cloudera will install a supported version of OpenJDK version 8.
- Install a system-provided version of OpenJDK**
By proceeding, Cloudera will install the default version of OpenJDK version 8 provided by the Operating System.

Back Continue



We selected the Manually Manager JDK option as shown in the screenshot above.

12. Provide SSH login credentials for the cluster and click Continue.

CLUSTER
Add Cluster - Installation

Welcome

Cluster Basics

Specify Hosts

Select Repository

Select JDK

6 Enter Login Credentials

7 Install Agents

8 Install Parcels

9 Inspect Cluster

Parcels

Recent Commands

Support

admin

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:

Number of Simultaneous Installations:
(Running a large number of installations at once can consume large amounts of network bandwidth and other system resources)

Back Continue

The installation of the local Cloudera repository and using parcels begins.

CLUSTER
Add Cluster - Installation

Welcome

Cluster Basics

Specify Hosts

Select Repository

Select JDK

Enter Login Credentials

7 Install Agents

8 Install Parcels

9 Inspect Cluster

Parcels

Running Commands 1

Support

admin

Install Agents

Installation in progress.

0 of 7 host(s) completed successfully. [Abort Installation](#)

| Hostname | IP Address | Progress | Status |
|------------------------|-------------|----------------------------------|---|
| rhel11.cdp.cisco.local | 10.18.1.141 | <div style="width: 100%;"></div> | C Installing cloudera-manager-agent package... Details |
| rhel12.cdp.cisco.local | 10.18.1.142 | <div style="width: 100%;"></div> | C Installing cloudera-manager-agent package... Details |
| rhel13.cdp.cisco.local | 10.18.1.143 | <div style="width: 100%;"></div> | C Installing cloudera-manager-agent package... Details |
| rhel14.cdp.cisco.local | 10.18.1.144 | <div style="width: 100%;"></div> | C Installing cloudera-manager-agent package... Details |
| rhel15.cdp.cisco.local | 10.18.1.145 | <div style="width: 100%;"></div> | C Installing cloudera-manager-agent package... Details |
| rhel16.cdp.cisco.local | 10.18.1.146 | <div style="width: 100%;"></div> | C Installing cloudera-manager-agent package... Details |
| rhel17.cdp.cisco.local | 10.18.1.147 | <div style="width: 100%;"></div> | C Installing cloudera-manager-agent package... Details |

Back Continue

The screenshot shows the Cloudera Manager interface during the 'Add Cluster - Installation' process. The left sidebar contains a navigation menu with the following items: Welcome, Cluster Basics, Specify Hosts, Select Repository, Select JDK, Enter Login Credentials, Install Agents, **8 Install Parcels** (highlighted), and 9 Inspect Cluster. Below the menu are links for Parcels, Running Commands, Support, and an admin user icon. The main content area is titled 'Install Parcels' and displays the progress of parcel installation. A message states: 'The selected parcels are being downloaded and installed on all the hosts in the cluster.' Below this, a progress bar for 'Cloudera Runtime 7.1.3-1.odh7...' is shown with the following metrics: Downloaded: 100%, Distributed: 8/8 (57 MiB/s), Unpacked: 8/8, and Activated: 8/8. The progress bar is filled with green segments. At the bottom right of the main area are 'Back' and 'Continue' buttons.

13. Run the 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.

CLOUDERA Manager

Add Cluster - Installation

- Welcome
- Cluster Basics
- Specify Hosts
- Select Repository
- Select JDK
- Enter Login Credentials
- Install Agents
- Install Parcels
- 9 Inspect Cluster**

Parcels
Recent Commands
Support
admin

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](#)

You can use this tool to evaluate the network performance between hosts, such as ping latency.

Ping Timeout: Seconds
Amount of time after which the inspector reports a failure.

Ping Count:
Number of times the inspector pings each host.

Ping Packet Size: Bytes
Size of the test packet sent when pinging the hosts.

Status ✔ Last Run in 3 minutes Duration 13.93s [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 in 3 minutes Duration 10.29s [Show Inspector Results](#) [Run Again](#) [More](#)

[Back](#) [Continue](#)

15. Select services that need to be started on the cluster.

CLOUDERA Manager

Add Cluster - Configuration

- 1 Select Services**
- 2 Assign Roles
- 3 Setup Database
- 4 Enter Required Parameters
- 5 Review Changes
- 6 Command Details
- 7 Summary

Parcels
Running Commands
Support
admin

Select Services

Choose a combination of services to install.

Data Engineering
Process, develop, and serve predictive models.
Services: HDFS, YARN, YARN Queue Manager, Ranger, Atlas, Hive, Hive on Tez, Spark, Oozie, Hue, and Data Analytics Studio

Data Mart
Browse, query, and explore your data in an interactive way.
Services: HDFS, Ranger, Atlas, Hive, Impala, and Hue

Operational Database
Real-time insights for modern data-driven business.
Services: HDFS, Ranger, Atlas, and HBase

Custom Services
Choose your own services. Services required by chosen services will automatically be included.

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.

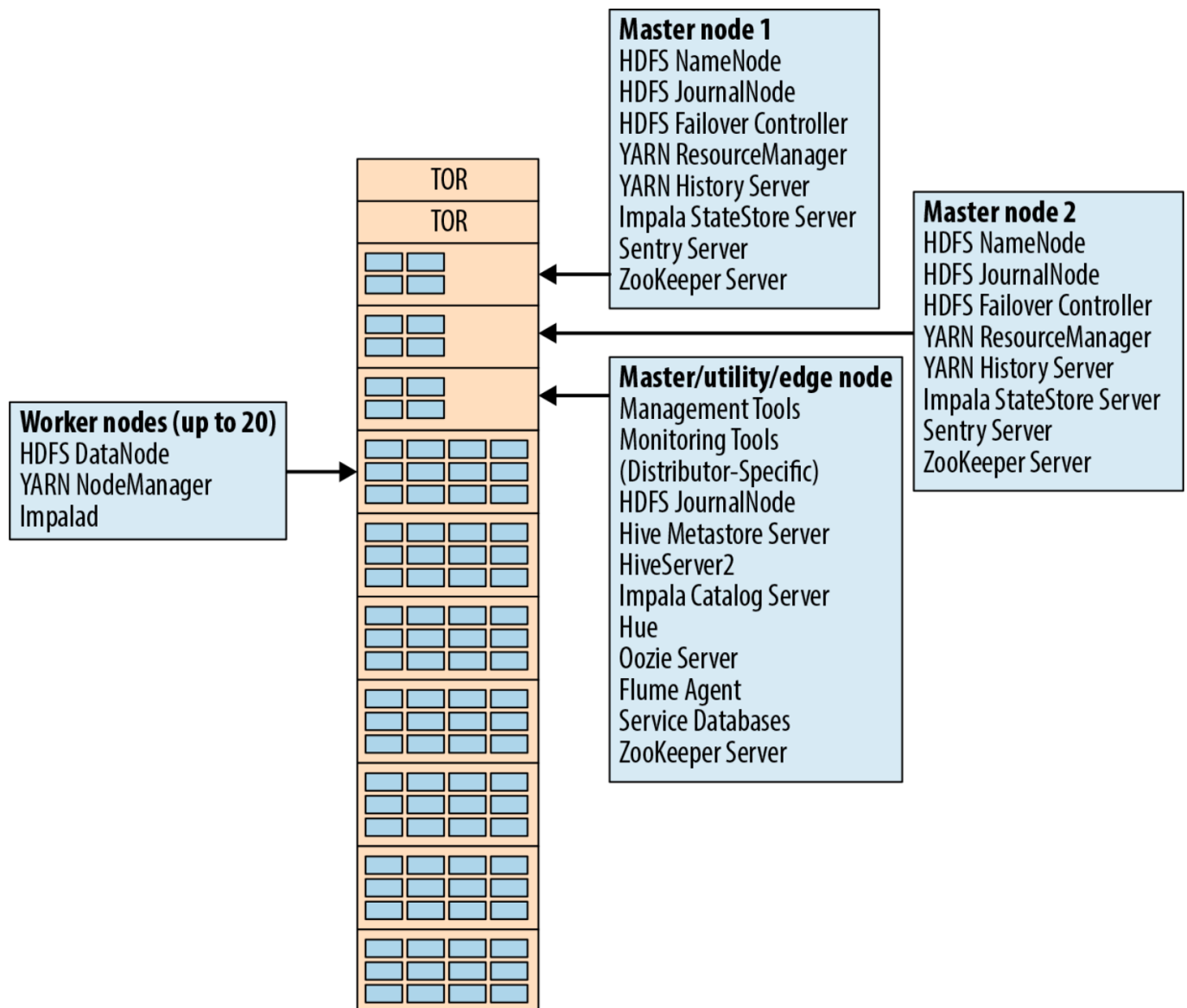
[Back](#) [Continue](#)



Custom Services was chosen for this POC.

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.

We define a small cluster as not exceeding 20 worker nodes. Small clusters are typically implemented on a single rack, but if Hadoop services are deployed redundantly, it is perfectly feasible to distribute even small clusters on multiple racks. In this example there are three management nodes, of which one also collocates utility and edge cluster roles.



17. Reconfigure the service assignment to match [Table 5](#).

Table 5. Service/Role Assignment

| Service Name | Host |
|------------------------------|--|
| NameNode | Rhel1, rhel2 (HA) |
| HistoryServer | Rhel2 |
| JournalNodes | Rhel1, rhel2, rhel3 |
| ResourceManager | Rhel1, rhel2 (HA) |
| Hue Server | Rhel3 |
| HiveMetastore Server | Rhel3 |
| HiveServer2 | Rhel3 |
| HBase Master | Rhel1, rhel2 |
| Oozie Server | Rhel3 |
| ZooKeeper | Rhel1, rhel2, rhel3 |
| DataNode | Rhel4 to rhel48 |
| NodeManager | Rhel4 to rhel48 |
| RegionServer | Rhel4 to rhel48 |
| Sqoop Server | Rhel3 |
| Impala Catalog Server Daemon | Rhel3 |
| Impala State Store | Rhel1 |
| Impala Daemon | Rhel4 to rhel48 |
| Solr Server | Rhel4 (can be installed on all hosts if needed, if there is a search use case) |
| Spark History Server | Rhel1, rhel2 |

| Service Name | Host |
|-----------------|-----------------|
| Spark Executors | Rhel4 to rhel48 |

Figure 48. Assign Roles in Cloudera Manager; Cluster Creation Wizard Example

CLUSTER
CLouDERA Manager

Add Cluster - Configuration

- 1 Select Services
- 2 Assign Roles**
- 3 Setup Database
- 4 Enter Required Parameters
- 5 Review Changes
- 6 Command Details
- 7 Summary

Parcels
Running Commands
Support
admin

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](#)

HDFS

- NameNode x 1 New: rhel10.cdp.cisco.local
- SecondaryNameNode x 1 New: rhel11.cdp.cisco.local
- Balancer x 1 New: rhel10.cdp.cisco.local
- HttpFS: Select hosts
- NFS Gateway: Select hosts
- DataNode x 5 New: rhel[13-17].cdp.cisco.local

Hive

- Gateway x 8 New: rhel[10-17].cdp.cisco.local
- Hive Metastore Server x 1 New: rhel10.cdp.cisco.local
- WebHCat Server: Select hosts
- HiveServer2 x 2 New: rhel[11-12].cdp.cisco.local

Hive on Tez

- Gateway x 8 New: rhel[10-17].cdp.cisco.local
- HiveServer2 x 3 New: rhel[10-12].cdp.cisco.local

Hue

- Hue Server x 1 New: rhel10.cdp.cisco.local
- Load Balancer x 2 New: rhel[11-12].cdp.cisco.local

Back Continue

CLOUDERA Manager

Impala

- Impala StateStore x 1 New:
- Impala Catalog Server x 1 New:
- Impala Daemon x 5 New:

Cloudera Management Service

- Service Monitor x 1 New:
- Activity Monitor x 1 New:
- Host Monitor x 1 New:
- Reports Manager x 1 New:
- Event Server x 1 New:
- Alert Publisher x 1 New:
- Telemetry Publisher:

Oozie

- Oozie Server x 1 New:

Ozone

- Storage Container Manager x 1 New:
- Ozone Manager x 3 New:
- Ozone Recon x 1 New:
- Ozone DataNode x 5 New:
- S3 Gateway x 1 New:
- Ozone Prometheus:
- Gateway x 8 New:

Ranger

- Ranger Admin x 1 New:
- Ranger Usersync x 1 New:
- Ranger Tagsync x 1 New:

Parcels

Running Commands

Support

admin

Back Continue

CLOUDERA Manager

Gateway x 8 New:

Ranger

- Ranger Admin x 1 New:
- Ranger Usersync x 1 New:
- Ranger Tagsync x 1 New:

Solr

- Solr Server x 1 New:

Spark

- History Server x 1 New:
- Gateway x 8 New:

Tez

- Gateway x 8 New:

YARN

- ResourceManager x 1 New:
- JobHistory Server x 1 New:
- NodeManager x 5 New:

ZooKeeper

- Server x 1 New:

Parcels

Running Commands

Support

admin

Back Continue

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 high availability recommendation defined in [Table 5](#).

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.

The screenshot displays the 'Setup Database' configuration page in CLOUDERA Manager. The page is titled 'Add Cluster - Configuration' and shows a progress bar with steps: Select Services, Assign Roles, Setup Database (current), Enter Required Parameters, Review Changes, Command Details, and Summary. The 'Setup Database' section is expanded, showing configuration for several services. Each service configuration includes a 'Type' dropdown (all set to PostgreSQL), a 'Database Hostname' field (all set to localhost), a 'Database Name' field, and a 'Username' field. A 'Test Connection' button is located at the bottom right of the configuration area. The services listed are Activity Monitor, Reports Manager, Oozie Server, Ranger, Hive, Data Analytics Studio, and Hue. Each service configuration is marked as 'Successful'.

| Service | Type | Database Hostname | Database Name | Username |
|-----------------------|------------|-------------------|---------------|-------------|
| Activity Monitor | PostgreSQL | localhost | amon | amon |
| Reports Manager | PostgreSQL | localhost | rman | rman |
| Oozie Server | PostgreSQL | localhost | oozie | oozie |
| Ranger | PostgreSQL | localhost | ranger | rangeradmin |
| Hive | PostgreSQL | localhost | hive | hive |
| Data Analytics Studio | PostgreSQL | localhost | das | das |
| Hue | PostgreSQL | localhost | hue | hue |

4. Review and customize the configuration changes based on your requirements.

CLUSTER
CLOUDERA
Manager

Add Cluster - Configuration

- Select Services
- Assign Roles
- Setup Database
- 4 Enter Required Parameters**
- Review Changes
- Command Details
- Summary

Enter Required Parameters

| | | |
|---|----------------------------|---|
| Ranger Admin User Initial Password rangeradmin_user_password | Ranger (Service-Wide) Undo | ? |
| Ranger Usersync User Initial Password rangerusersync_user_password | Ranger (Service-Wide) Undo | ? |
| Ranger Tagsync User Initial Password rangertagsync_user_password | Ranger (Service-Wide) Undo | ? |
| Ranger KMS Keyadmin User Initial Password keyadmin_user_password | Ranger (Service-Wide) Undo | ? |

Parcels
Running Commands
Support
admin

Back Continue

CLUSTER
CLOUDERA
Manager

Add Cluster - Configuration

- Select Services
- Assign Roles
- Setup Database
- Enter Required Parameters
- 5 Review Changes**
- Command Details
- Summary

Review Changes

| | | |
|--|---|---|
| Enable Kerberos Authentication kerberos.auth.enabled | <input type="checkbox"/> CDP-PVT-BASE1 > Data Analytics Studio (Service-Wide) | ? |
| Hive Session Parameters das_hive_session_params | CDP-PVT-BASE1 > Data Analytics Studio (Service-Wide) | ? |
| Hive Secure Session Parameters das_hive_secure_session_params | CDP-PVT-BASE1 > Data Analytics Studio (Service-Wide) | ? |
| Additional Eventprocessor Java Options das_eventprocessor_java_opts | CDP-PVT-BASE1 > Data Analytics Studio Eventprocessor Default Group -Xmx4096m | ? |
| Additional Eventprocessor Classpath data_analytics_studio_ep_additional_classpath | CDP-PVT-BASE1 > Data Analytics Studio Eventprocessor Default Group | ? |
| Additional Webapp Java Options das_webapp_java_opts | CDP-PVT-BASE1 > Data Analytics Studio Webapp Server Default Group -Xmx4096m | ? |
| Additional Webapp Classpath data_analytics_studio_webapp_additional_classpath | CDP-PVT-BASE1 > Data Analytics Studio Webapp Server Default Group | ? |
| DAS User Authentication data_analytics_studio_user_authentication | CDP-PVT-BASE1 > Data Analytics Studio Webapp Server Default Group DEFAULT | ? |
| Knox SSO Endpoint data_analytics_studio_webapp_knox_sso_url | CDP-PVT-BASE1 > Data Analytics Studio Webapp Server Default Group | ? |

Parcels
Running Commands
Support
admin

Back Continue

5. Click Continue to start running the cluster services.

CLOUDERA Manager

Add Cluster - Configuration

- Select Services
- Assign Roles
- Setup Database
- Enter Required Parameters
- Review Changes
- Command Details**
- Summary

Parcels
Running Commands
Support
admin

First Run Command

Status: ✔ Finished Context: [CDP-PVT-BASE1](#) Sep 15, 12:01:56 PM 9m

Successfully completed 13 steps.

✔ Completed 1 of 1 step(s).

Show All Steps Show Only Failed Steps Show Only Running Steps

| | | |
|--|---------------------|--------|
| Run a set of services for the first time Successfully completed 13 steps. | Sep 15, 12:01:56 PM | 9m |
| Execute 10 steps in sequence Successfully completed 13 steps. | Sep 15, 12:02:01 PM | 8.9m |
| Ensuring that the expected software releases are installed on hosts. | Sep 15, 11:51:26 AM | 5.01s |
| Execute 7 steps in parallel | Sep 15, 11:51:31 AM | 33.53s |
| Execute 3 steps in parallel | Sep 15, 12:02:01 PM | 31.37s |
| Execute 16 steps in parallel | Sep 15, 12:02:33 PM | 59.18s |
| Execute 7 steps in parallel | Sep 15, 12:03:32 PM | 44.37s |
| Execute 3 steps in parallel | Sep 15, 12:04:16 PM | 34.78s |
| Start Ranger Ranger | Sep 15, 12:04:51 PM | 23.36s |
| Execute 4 steps in parallel | Sep 15, 12:05:15 PM | 5.3m |
| Start Hive on Tez Hive on Tez | Sep 15, 12:10:32 PM | 23.74s |
| Verifying successful startup of services | Sep 15, 12:10:56 PM | 43ms |

Rows per page: 25 1 - 10 of 10

[Back](#) [Continue](#)

6. Hadoop services are installed, configured, and now running on all the nodes of the cluster. Click Finish to complete the installation.

CLOUDERA Manager

Add Cluster - Configuration

- Select Services
- Assign Roles
- Setup Database
- Enter Required Parameters
- Review Changes
- Command Details
- Summary**

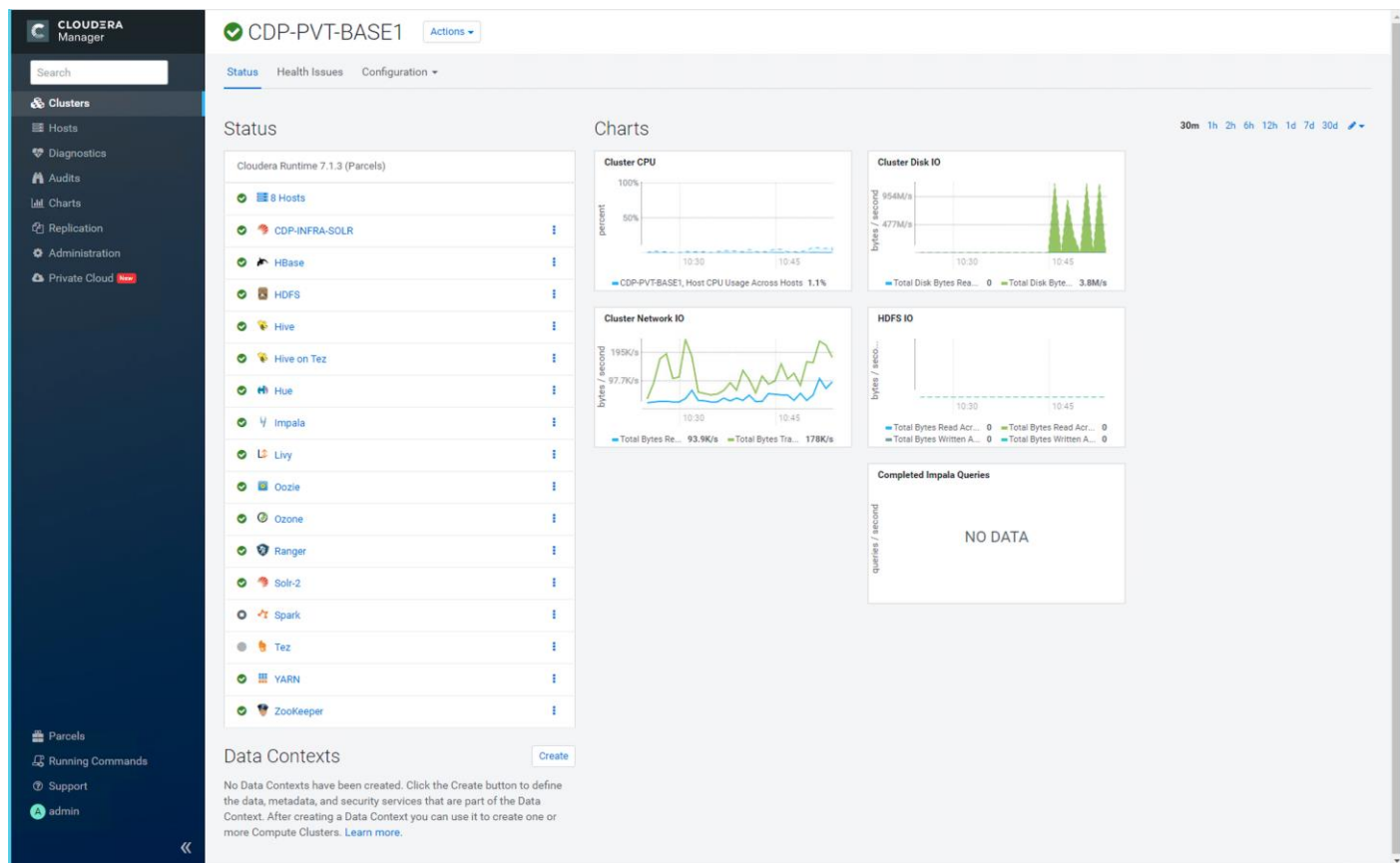
Parcels
Recent Commands
Support
admin

Summary

✔ The services are installed, configured, and running on your cluster.

[Back](#) [Finish](#)

7. Cloudera Manager now displays the status of all Hadoop services running on the cluster.



Securing CDP

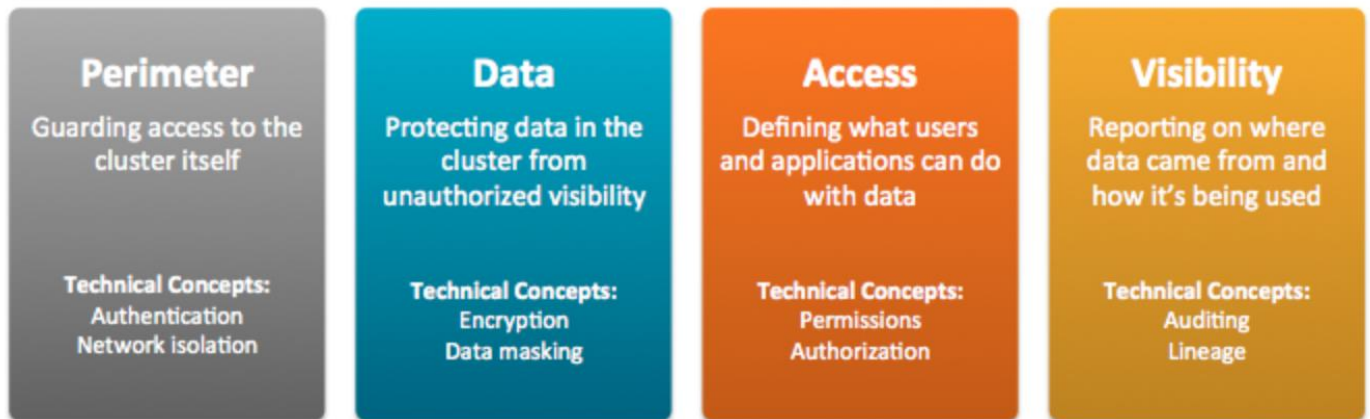
As a system designed to support vast amounts and types of data, Cloudera clusters must meet ever-evolving security requirements imposed by regulating agencies, governments, industries, and the general public. Cloudera clusters comprise both Hadoop core and ecosystem components, all of which must be protected from a variety of threats to ensure the confidentiality, integrity, and availability of all the cluster's services and data.



Its highly recommended to consult professional services team while deploying the security features like Authentication, Authorization, and Audit.

Security Requirements

Goals for data management systems, such as confidentiality, integrity, and availability, require that the system be secured across several dimensions. These can be characterized in terms of both general operational goals and technical concepts, as shown in the figure below:



Perimeter Access to the cluster must be protected from a variety of threats coming from internal and external networks and from a variety of actors. Network isolation can be provided by proper configuration of firewalls, routers, subnets, and the proper use of public and private IP addresses, for example. Authentication mechanisms ensure that people, processes, and applications properly identify themselves to the cluster and prove they are who they say they are, before gaining access to the cluster.

Data Data in the cluster must always be protected from unauthorized exposure. Similarly, communications between the nodes in the cluster must be protected. Encryption mechanisms ensure that even if network packets are intercepted or hard-disk drives are physically removed from the system by bad actors, the contents are not usable.

Access Access to any specific service or item of data within the cluster must be specifically granted. Authorization mechanisms ensure that once users have authenticated themselves to the cluster, they can only see the data and use the processes to which they have been granted specific permission.

Visibility Visibility means that the history of data changes is transparent and capable of meeting data governance policies. Auditing mechanisms ensure that all actions on data and its lineage—source, changes over time, and so on—are documented as they occur.

Hadoop Security Architecture

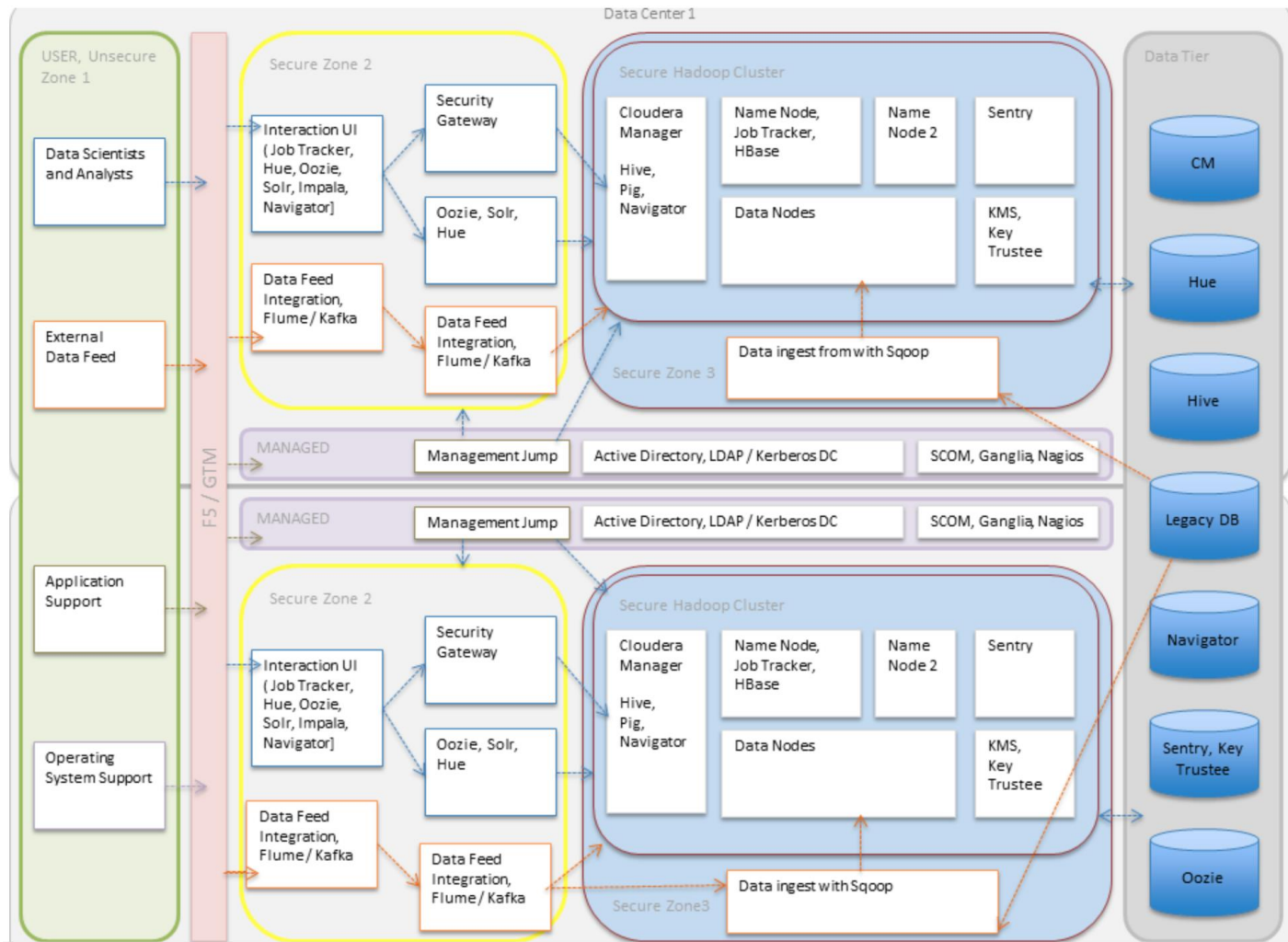
The figure below is an example of some of the many components at work in a production Cloudera enterprise cluster. The figure highlights the need to secure clusters that may ingest data from both internal and external data feeds, and across possibly multiple datacenters. Securing the cluster requires applying authentication and access controls throughout these many inter- and intra-connections, as well as to all users who want to query, run jobs, or even view the data held in the cluster.

External data streams are authenticated by mechanisms in place for Flume and Kafka. Data from legacy databases is ingested using Sqoop. Data scientists and BI analysts can use interfaces such as Hue to work with data on Impala or Hive, for example, to create and submit jobs. Kerberos authentication can be leveraged to protect all these interactions.

Encryption can be applied to data at-rest using transparent HDFS encryption with an enterprise-grade Key Trustee Server. Cloudera also recommends using Navigator Encrypt to protect data on a cluster associated with the Cloudera Manager, Cloudera Navigator, Hive and HBase metastores, and any log files or spills.

Authorization policies can be enforced using Sentry (for services such as Hive, Impala, and Search) as well as HDFS Access Control Lists.

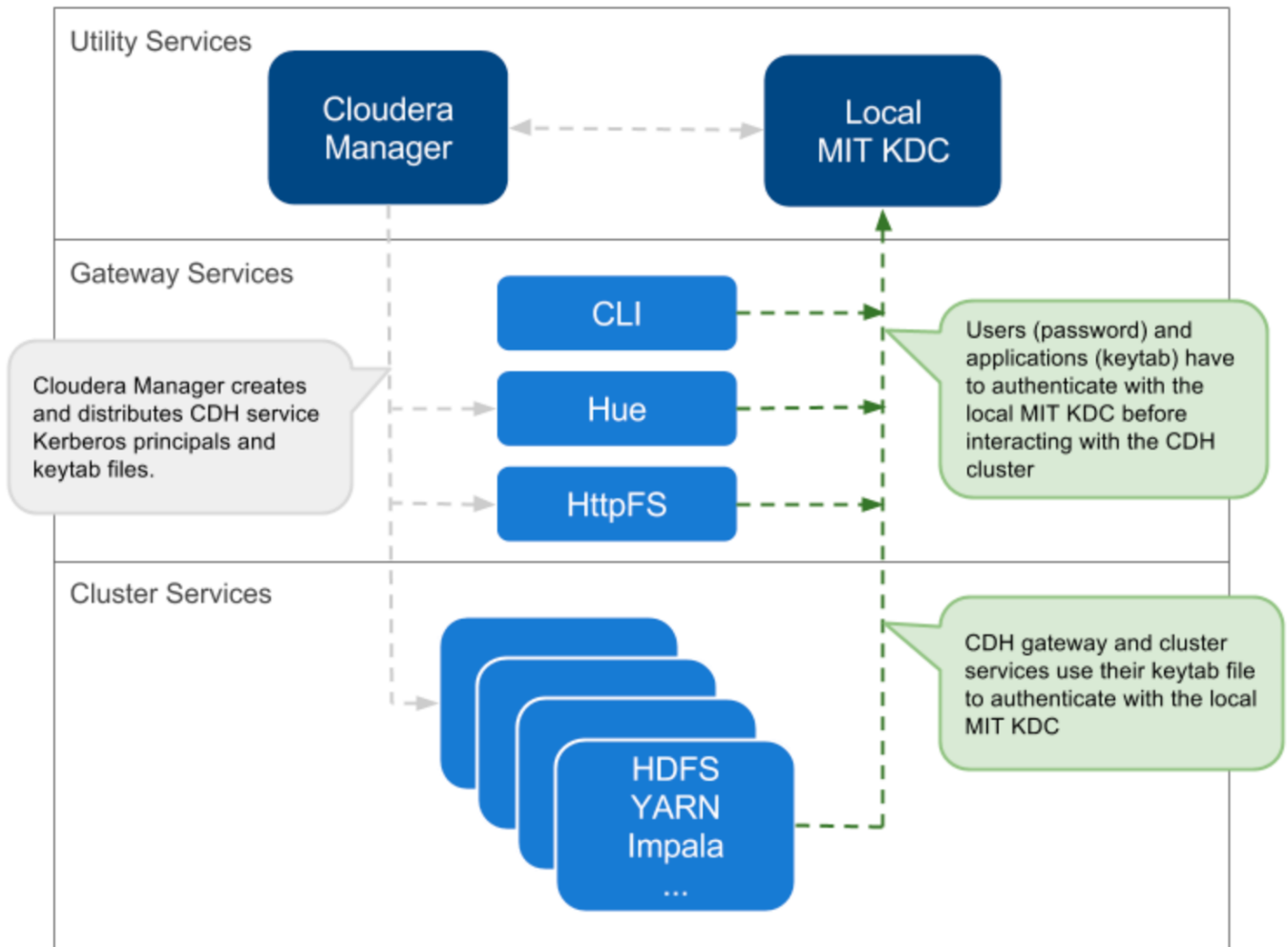
Auditing capabilities can be provided by using Cloudera Navigator.



Enable Kerberos Authentication for CDP

Cloudera Manager provides a wizard for integrating your organization's Kerberos instance with your cluster to provide authentication services.

Kerberos must already be deployed in your organization and the Kerberos key distribution center (KDC) must be ready to use, with a realm established. If you are using Red Hat Identity Management/FreeIPA, all of your cluster hosts must already be joined to the IPA domain. For Hue and Oozie, the Kerberos realm must support renewable tickets. Cloudera does not provide a Kerberos implementation. Cloudera clusters can be configured to use Kerberos for authentication, either MIT Kerberos or Microsoft Server Active Directory Kerberos, specifically the Key Distribution Center or KDC. The Kerberos instance must be setup and operational before you can configure the cluster to use it.



Before integrating Kerberos with your cluster, configure TLS encryption between CM Server and all CM Agent host systems in the cluster. During the Kerberos integration process, CM Server sends keytab files to the CM Agent hosts, and TLS encrypts the network communication, so these files are protected. For detailed instruction see [Configuring TLS Encryption for Cloudera Manager and CDH Using Auto-TLS](#).



For clusters deployed using Cloudera Manager Server, Cloudera recommends using the Kerberos configuration wizard available through the Cloudera Manager Admin Console. For more information, see [Enabling Kerberos Authentication Using the Wizard](#).

Enable Encryption for CDP

Encryption is a process that uses digital keys to encode various components—text, files, databases, passwords, applications, or network packets, for example—so that only the appropriate entity (user, system process, and so on) can decode (decrypt) the item and view, modify, or add to the data. Cloudera provides encryption mechanisms to protect data persisted to disk or other storage media (*data at rest encryption* or simply, data encryption) and as it moves over the network (*data in transit encryption*).

Protecting Data At-Rest

Protecting data at rest typically means encrypting the data when it is stored on disk and letting authorized users and processes—and only authorized users and processes—to decrypt the data when needed for the application or task at hand. With data-at-rest encryption, encryption keys must be distributed and managed, keys should be rotated or changed on a regular basis (to reduce the risk of having keys compromised), and many other factors complicate the process.

However, encrypting data alone may not be sufficient. For example, administrators and others with sufficient privileges may have access to personally identifiable information (PII) in log files, audit data, or SQL queries. Depending on the specific use case—in hospital or financial environment, the PII may need to be redacted from all such files, to ensure that users with privileges on the logs and queries that might contain sensitive data are nonetheless unable to view that data when they should not.

Cloudera provides complementary approaches to encrypting data at rest, and provides mechanisms to mask PII in log files, audit data, and SQL queries.

Protecting Data In-Transit

For data-in-transit, implementing data protection and encryption is relatively easy. Wire encryption is built into the Hadoop stack, such as SSL, and typically does not require external systems. This data-in-transit encryption is built using session-level, one-time keys, by means of a session handshake with immediate and subsequent transmission. Thus, data-in-transit avoids much of the key management issues associated with data-at-rest due to the temporal nature of the keys, but it does rely on proper authentication; a certificate compromise is an issue with authentication but can compromise wire encryption. As the name implies, data-in-transit covers the secure transfer and intermediate storage of data. This applies to all process-to-process communication, within the same node or between nodes. There are three primary communication channels:

HDFS Transparent Encryption: Data encrypted using [HDFS Transparent Encryption](#) is protected end-to-end. Any data written to and from HDFS can only be encrypted or decrypted by the client. HDFS does not have access to the unencrypted data or the encryption keys. This supports both, at-rest encryption as well as in-transit encryption.

Data Transfer: The first channel is data transfer, including the reading and writing of data blocks to HDFS. Hadoop uses a SASL-enabled wrapper around its native direct TCP/IP-based transport, called DataTransportProtocol, to secure the I/O streams within a DIGEST-MD5 envelope. This procedure also employs secured HadoopRPC (see Remote Procedure Calls) for the key exchange. The HttpFS REST interface, however, does not provide secure communication between the client and HDFS, only secured authentication using SPNEGO.

For the transfer of data between DataNodes during the shuffle phase of a MapReduce job (that is, moving intermediate results between the Map and Reduce portions of the job), Hadoop secures the communication channel with HTTP Secure (HTTPS) using Transport Layer Security (TLS). For more information, see [Using YARN with a secure cluster](#).

Remote Procedure Calls: The second channel is system calls to remote procedures (RPC) to the various systems and frameworks within a Hadoop cluster. Like data transfer activities, Hadoop has its own native protocol for RPC, called HadoopRPC, which is used for Hadoop API client communication, intra-Hadoop services communication, as well as monitoring, heartbeats, and other non-data, non-user activity. HadoopRPC is SASL-enabled for secured transport and defaults to Kerberos and DIGEST-MD5 depending on the type of communication and security settings.

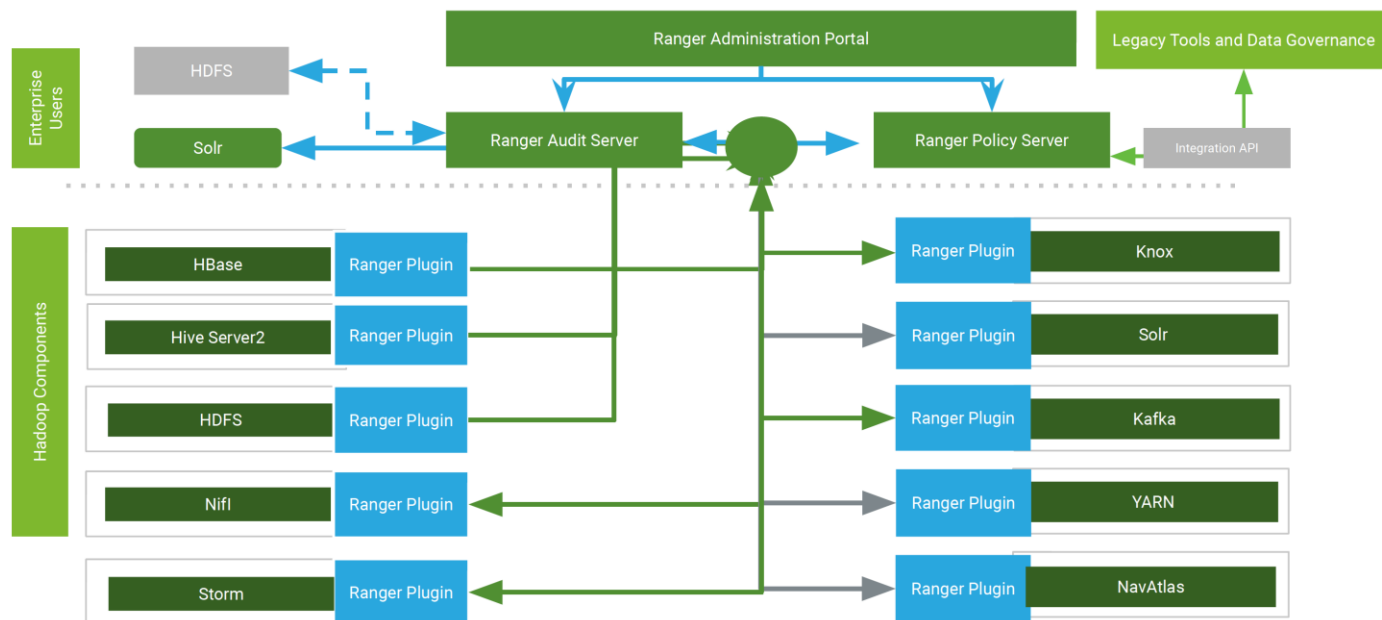
User Interfaces: The third channel includes the various web-based user interfaces within a Hadoop cluster. For secured transport, the solution is straightforward; these interfaces employ HTTPS.

Enabling Ranger for Authorization and Audit for CDP

Authorization is one of the fundamental security requirements of any computing environment. Its goal is to ensure that only the appropriate people or processes can access, view, use, control, or change specific resources, services, or data. In any cluster deployed to meet specific workloads using various CDH components (Hive, HDFS, Impala, and so on), different authorization mechanisms can ensure that only authorized users or processes can access data, systems, and other resources as needed. Ideally, authorization mechanisms can leverage the authentication mechanisms, so that when users login to a system—a cluster, for example—they are transparently authorized based on their identity across the system for the applications, data, and other resources they are authorized to use.

For example, Cloudera CDH clusters can be configured to leverage the user and group accounts that exist in the organization's Active Directory (or other LDAP-accessible directory) instance.

Ranger Architecture



For more information about the various configurations and integrations, go to: [Using Ranger to Provide Authorization in CDP](#).

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 NameNodes (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 counterpart 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.

Rack-Aware Replica Placement

Data is stored in blocks on the HDFS filesystem. Data blocks are replicated, three times by default, for data durability and high availability. Block replicas are distributed across the cluster's racks and nodes using a rack placement policy. NameNode uses a rack-aware replica placement policy to improve data reliability, durability, and network bandwidth utilization. Using the rack-aware replica placement policy replicas of a block are placed on different DataNodes in the cluster on different racks so that failure of a single DataNode or rack does not make all replicas of a block unavailable. Block replicas on other DataNodes/racks are used in the event of failure of a DataNode or a rack. The rack-aware replica placement policy is implemented by the NameNode.

Using rack-aware replica placement improves data reliability, availability, and network utilization. To get maximum performance, it is important to configure rack-aware policy in CDP so that it knows the topology of your network. Network locations such as hosts and racks are represented in a tree, which reflects the network "distance" between locations. HDFS will use the network location to be able to place block replicas more intelligently to trade off performance and resilience. When placing jobs on hosts, CDP will prefer within-rack transfers (where there is more bandwidth available) to off-rack transfers; the MapReduce and YARN schedulers use network location to determine where the closest replica is as input to a map task. These computations are performed with the assistance of rack awareness scripts. Cloudera Manager includes internal rack awareness scripts, but you must specify the racks where the hosts in your cluster are located. If your cluster contains more than 10 hosts, Cloudera recommends that you specify the rack for each host. HDFS, MapReduce, and YARN will automatically use the racks you specify.

Customers can avoid complete data loss of data when a worker node goes down. No duplicated replicas are on the same node or nodes under the same rack. First replica is on the local rack or on nodes under the same rack.

Cloudera Manager supports nested rack specifications. For example, you could specify the rack `/rack3`, or `/group5/rack3` to indicate the third rack in the fifth group. All hosts in a cluster must have the same number of path components in their rack specifications.

To specify racks for hosts, follow these steps:

1. Click the Hosts tab.
2. Check the checkboxes next to the host(s) for a particular rack, such as all hosts for `/rack123`.
3. Click Actions for Selected (n) > Assign Rack, where n is the number of selected hosts.
4. Enter a rack name or ID that starts with a slash / such as `/rack123` or `/aisle1/rack123`, and then click Confirm.
5. Optionally, restart the affected services. Rack assignments are not automatically updated for running services.

To configure the rack awareness on a Hadoop cluster, follow these steps:

1. Using Cloudera Manager, configure the following in safety valves:

a. In HDFS - Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml:

```
<property>
<name>net.topology.impl</name>
<value> org.apache.hadoop.net.NetworkTopologyWithNodeGroup</value>
</property>
<property>
<name>net.topology.nodegroup.aware</name>
<value> true</value>
</property>
<property>
<name> dfs.block.replicator.classname</name>
<value>org.apache.hadoop.hdfs.server.blockmanagement.BlockPlacementPolicyWithNodeGroup</value>
</property>
```

2. In YARN Service MapReduce Advanced Configuration Snippet (Safety Valve) mapred.xml, add the following properties and values:

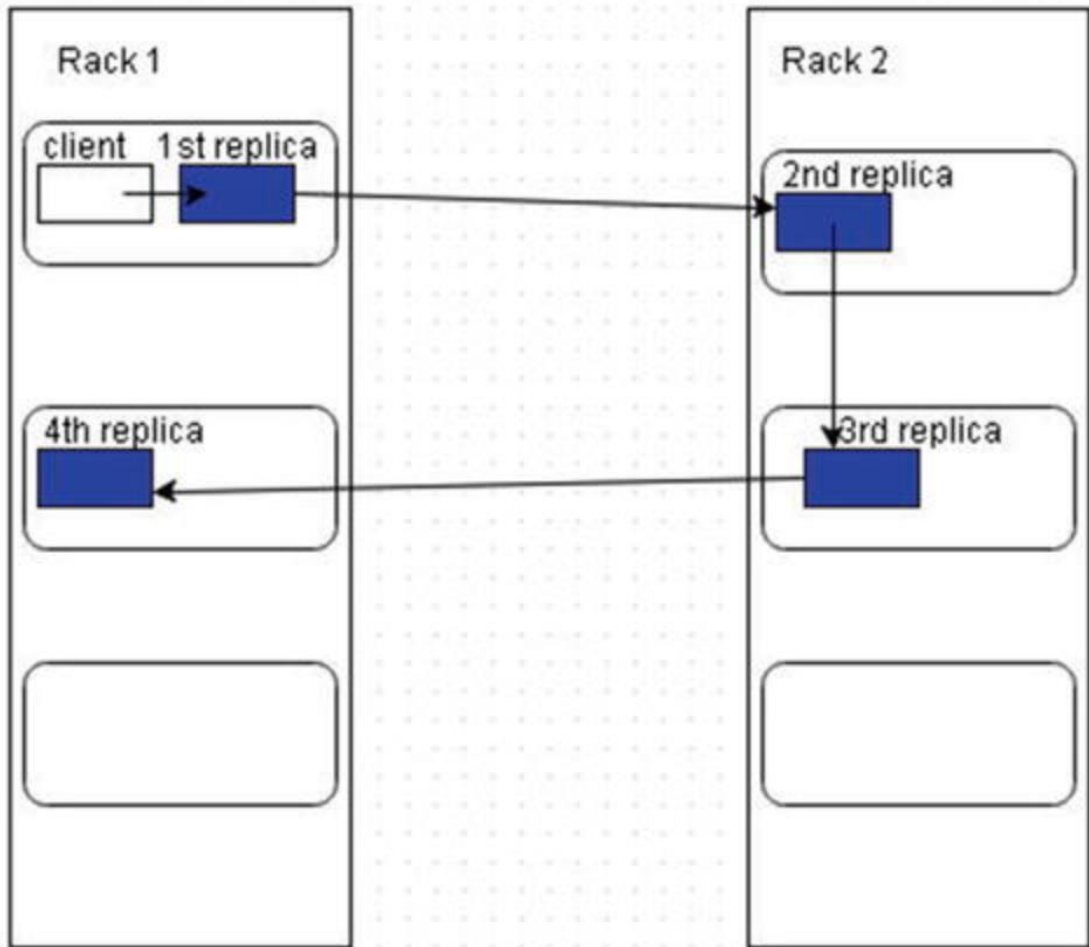
```
<property>
<name>mapred.jobtracker.nodegroup.aware</name>
<value>true</value>
</property>
<property>
<name>mapred.task.cache.levels</name>
<value>3</value>
</property>
```

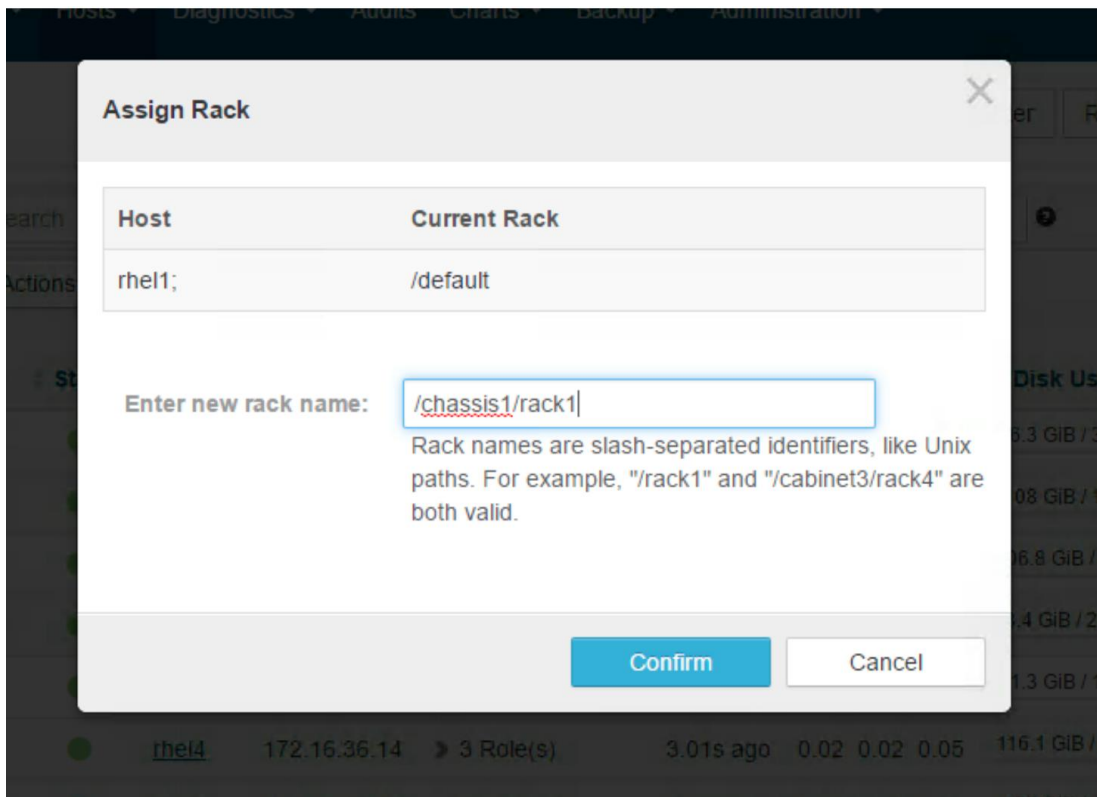
To set rack location of hosts, follow these steps:



For more information, see [Setting Racks for Hosts](#).

1. Select each node from the Hosts page and then assign a rack, following the format of /chassis\$ID/rack\$ID:





Follow the rack name format for each worker node in the Hadoop cluster.

Hadoop uses the rack information to place replica blocks on redundant racks.

After adding the safety valves and the rack names for each server, follow these steps:

1. Stop the cluster.
2. Deploy client config.
3. Start Zookeeper.
4. Start HDFS.
5. Start all the other services.

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

if 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://docs.cloudera.com/content/www/en-us/documentation/enterprise/6/6.3/PDF/cloudera-administration.pdf>

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:

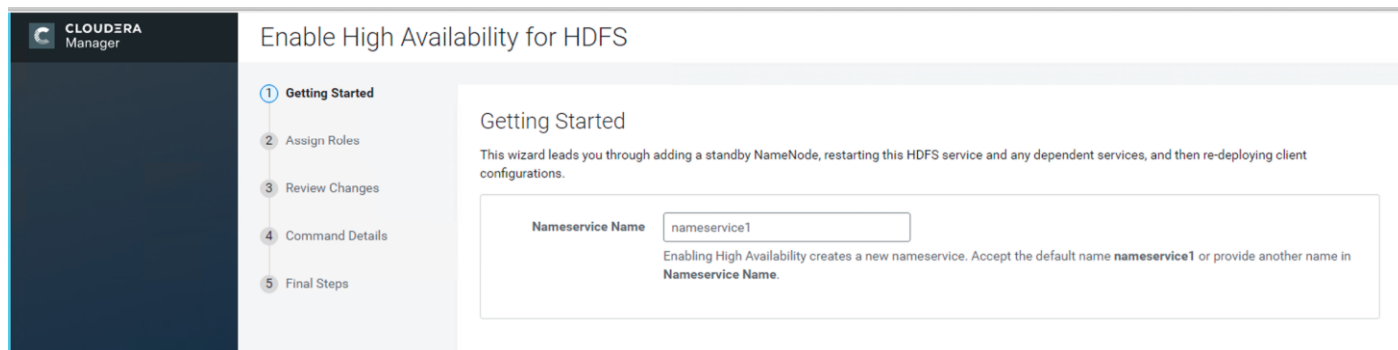
1. Log into the admin node (rhel1) and create the Edit directory for the JournalNodes(rhel10-12):

```
# ansible namenodes -m shell -a "mkdir -p /data/disk1/jn"
# ansible namenodes -m shell -a "chmod 77 /data/disk1/jn"
```

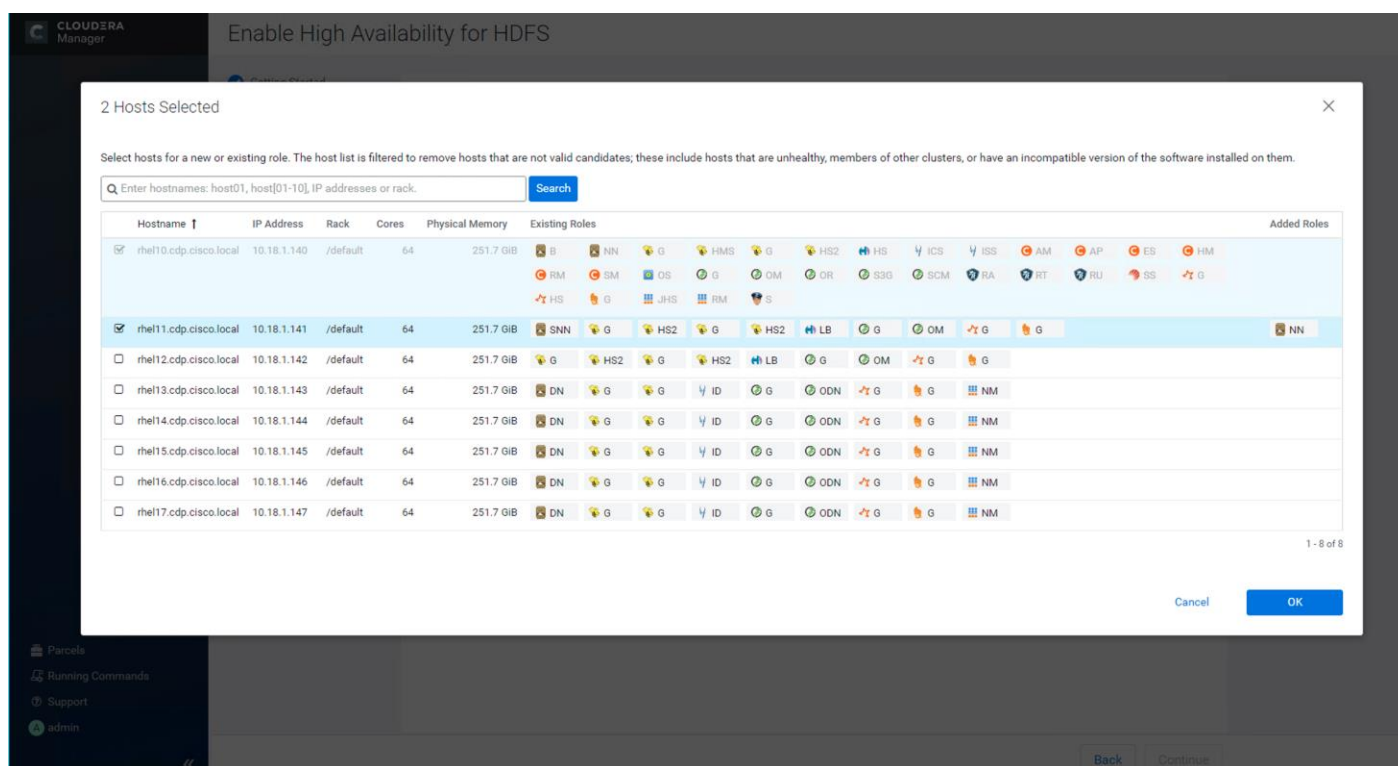
2. Log into the Cloudera manager and go to the HDFS service.
3. Select Actions> Enable High Availability. A screen showing the hosts that are eligible to run a standby NameNode and the JournalNodes displays.

The screenshot shows the Cloudera Manager interface for the HDFS service on a NameNode (Active). The 'Actions' menu is open, showing options like 'Start this NameNode', 'Stop this NameNode', 'Restart this NameNode', 'Enter Maintenance Mode', 'Format', 'Refresh Node List', 'Enter Safemode', 'Leave Safemode', 'Create /tmp Directory', 'Save Namespace', 'Finalize Metadata Upgrade', 'Bootstrap Standby NameNode', 'Initialize Shared Edits Directory', 'Monitor Offline', 'Create Ranger Audit Directory', 'Enable High Availability', 'List Open Files (lsdf)', 'Collect Stack Traces (jstack)', 'Heap Dump (jmap)', and 'Heap Histogram (jmap-histo)'. The 'Enable High Availability' option is highlighted. The main content area shows 'Health Tests' (14 Good, 2 Disabled) and 'Health History' (8 Became Good, 1 Became Good, 5 Became Good, 1 Became Good, 6 Became Disabled). There are also several performance graphs for Transactions, Average Edit Log Sync Time, RPC Workload Summary, and Queue Length.

- Specify a name for the nameservice or accept the default name nameservice1 and click Continue.



- In the NameNode Hosts field, click Select a host. The host selection dialog displays.
- Check the checkbox next to the hosts (rhe11) where the standby NameNode is to be set up and click OK.



- In the JournalNode Hosts field, click Select hosts. The host selection dialog displays.
- 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.

3 Hosts Selected

Select hosts for a new or existing role. The host list is filtered to remove hosts that are not valid candidates; these include hosts that are unhealthy, members of other clusters, or have an incompatible version of the software installed on them.

Enter hostnames: host01, host[01-10], IP addresses or rack.

Tip: Click the first checkbox, hold down the Shift key and click the last checkbox to select a range.

| Hostname | IP Address | Rack | Cores | Physical Memory | Existing Roles | Added Roles |
|--|-------------|----------|-------|-----------------|--|-------------|
| <input checked="" type="checkbox"/> rhe110.cdp.cisco.local | 10.18.1.140 | /default | 64 | 251.7 GiB | B, NN, G, HMS, G, HS2, HS, ICS, ISS, AM, AP, ES, HM, JN, RM, SM, OS, G, OM, OR, S3G, SCM, RA, RT, RU, SS, G, HS, G, JHS, RM, S | JN |
| <input checked="" type="checkbox"/> rhe111.cdp.cisco.local | 10.18.1.141 | /default | 64 | 251.7 GiB | SNN, G, HS2, G, HS2, LB, G, OM, G, G | NN, JN |
| <input checked="" type="checkbox"/> rhe112.cdp.cisco.local | 10.18.1.142 | /default | 64 | 251.7 GiB | G, HS2, G, HS2, LB, G, OM, G, G | JN |
| <input type="checkbox"/> rhe113.cdp.cisco.local | 10.18.1.143 | /default | 64 | 251.7 GiB | DN, G, G, ID, G, ODN, G, G, NM | |
| <input type="checkbox"/> rhe114.cdp.cisco.local | 10.18.1.144 | /default | 64 | 251.7 GiB | DN, G, G, ID, G, ODN, G, G, NM | |
| <input type="checkbox"/> rhe115.cdp.cisco.local | 10.18.1.145 | /default | 64 | 251.7 GiB | DN, G, G, ID, G, ODN, G, G, NM | |
| <input type="checkbox"/> rhe116.cdp.cisco.local | 10.18.1.146 | /default | 64 | 251.7 GiB | DN, G, G, ID, G, ODN, G, G, NM | |
| <input type="checkbox"/> rhe117.cdp.cisco.local | 10.18.1.147 | /default | 64 | 251.7 GiB | DN, G, G, ID, G, ODN, G, G, NM | |

1 - 8 of 8

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

Getting Started

2 Assign Roles

Review Changes

Command Details

Final Steps

Assign Roles

NameNode Hosts: rhe110.cdp.cisco.local (Cu...), rhe111.cdp.cisco.local

JournalNode Hosts: rhe[10-12].cdp.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 JournalNode edits directory into the fields for each JournalNode host.

The screenshot shows the Cloudera Manager interface for configuring High Availability for HDFS. The left sidebar shows the navigation menu with 'Review Changes' selected. The main content area is titled 'Review Changes' and contains a table of configuration parameters for the 'Service HDFS'.

Set the following configuration values for your new role(s). Required values are marked with *.

| Parameter | Group | Value | Description |
|---|--------|---|--|
| Service HDFS | | | |
| NameNode Data Directories* dfs.namenode.name.dir | rhel10 | /data/disk1/dfs/nn Inherited from: NameNode Default Group | Determines where on the local file system the NameNode should store the name table (fsimage). For redundancy, enter a comma-delimited list of directories to replicate the name table in all of the directories. Typical values are /data/N/dfs/nn where N=1..3. |
| | rhel11 | /data/disk1/dfs/nn Inherited from: NameNode Default Group | |
| JournalNode Edits Directory* dfs.journalnode.edits.dir | rhel10 | <input type="text" value="/data/disk1/jn"/> Reset to empty default value | Directory on the local file system where NameNode edits are written. |
| | rhel11 | <input type="text" value="/data/disk1/jn"/> Reset to empty default value | |
| | rhel12 | <input type="text" value="/data/disk1/jn"/> Reset to empty default value | |

Extra Options

- Force initialize the ZooKeeper ZNode for autofailover. Any previous ZNode used for this nameservice will be overwritten.
- Clear any existing data present in name directories of Standby NameNode.
Make sure you have backed up any existing data in the name directories of Standby NameNode.
- Clear any existing data present in the JournalNode edits directory for this nameservice.
Make sure you have backed up any existing data in the edits directory on **all** hosts running JournalNodes.

Buttons: Back, Continue



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 not to 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.

Enable High Availability for HDFS

- ✓ Getting Started
- ✓ Assign Roles
- ✓ Review Changes
- 4 Command Details**
- 5 Final Steps

Enable High Availability Command

Status 🔄 Running Context [HDFS](#) 📅 Sep 21, 10:31:54 PM Abort

✓ **Completed 12 of 20 step(s).**

Show All Steps Show Only Failed Steps Show Only Running Steps

| | | | | |
|---|---|--|---------------------|--------|
| ✓ | Check that name directories for the new Standby NameNode either do not exist or are writable and empty. Can optionally clear directories. | rhe111.cdp.cisco.local | Sep 21, 10:31:54 PM | 2.05s |
| > | Check that edits directories for the nameservice either do not exist or are writable and empty. Can optionally clear directories. | | Sep 21, 10:31:56 PM | 2.57s |
| > | Stop hdfs and its dependent services | CDP-PVT-BASE1 | Sep 21, 10:31:58 PM | 5.4m |
| ✓ | Creating roles to enable High Availability. | | Sep 21, 10:37:25 PM | 26ms |
| ✓ | Deleting the SecondaryNameNode role. The checkpoint directories of the SecondaryNameNode will not be deleted. | | Sep 21, 10:37:25 PM | 32ms |
| > | Configuring NameNodes and the HDFS service to enable High Availability. | | Sep 21, 10:37:25 PM | 1ms |
| ✓ | Initializing High Availability state in ZooKeeper. | Failover Controller (rhe110) | Sep 21, 10:37:25 PM | 18.56s |
| > | Starting the JournalNodes | | Sep 21, 10:37:44 PM | 23.46s |
| > | ⚠️ Formatting the name directories of the current NameNode. If the name directories are not empty, this is expected to fail. Failed to format NameNode. | NameNode (rhe110) | Sep 21, 10:38:07 PM | 19.25s |
| > | Initializing shared edits directory of NameNodes. | NameNode (rhe110) | Sep 21, 10:38:27 PM | 19.8s |
| ✓ | Starting the NameNode that will be transitioned to active mode | NameNode (rhe110) | Sep 21, 10:38:46 PM | 22.64s |

Back Continue

- 📦 Parcels
- 🔧 Running Commands **1**
- 🔗 Support
- 👤 admin



| Task | Target | Time | Duration |
|--|-----------------------------|---------------------|----------|
| Configuring NameNodes and the HDFS service to enable High Availability. | | Sep 21, 10:37:25 PM | 1ms |
| Initializing High Availability state in ZooKeeper. | Failover Controller (rhe10) | Sep 21, 10:37:25 PM | 18.54s |
| Starting the JournalNodes | | Sep 21, 10:37:44 PM | 23.46s |
| Formatting the name directories of the current NameNode. If the name directories are not empty, this is expected to fail. Failed to format NameNode. | NameNode (rhe10) | Sep 21, 10:38:07 PM | 19.25s |
| Initializing shared edits directory of NameNodes. | NameNode (rhe10) | Sep 21, 10:38:27 PM | 19.8s |
| Starting the NameNode that will be transitioned to active mode NameNode (rhe10). | NameNode (rhe10) | Sep 21, 10:38:46 PM | 22.64s |
| Waiting for the Active NameNode to start up. | NameNode (rhe10) | Sep 21, 10:39:09 PM | 4.77s |
| Bootstrapping Standby NameNode by initializing its name directories. | NameNode (rhe11) | Sep 21, 10:39:14 PM | 12.55s |
| Starting Standby NameNode | NameNode (rhe11) | Sep 21, 10:39:26 PM | 22.65s |
| Starting the Failover Controller on the host of the Active NameNode. | Failover Controller (rhe10) | Sep 21, 10:39:49 PM | 22.63s |
| Starting the Failover Controller on the host of the Standby NameNode. | Failover Controller (rhe11) | Sep 21, 10:40:12 PM | 22.63s |
| Waiting for the Standby NameNode to start up. | NameNode (rhe11) | Sep 21, 10:40:34 PM | 4.33s |
| Creating HDFS /tmp directory if not already created. Command (Create /tmp Directory (1546336589)) has failed | NameNode (rhe10) | Sep 21, 10:40:39 PM | 3.1m |
| Start hdfs and its dependent services | CDP-PVT-BASE1 | Sep 21, 10:43:47 PM | 2.7m |
| Deploying configurations for clients of services in this cluster. | CDP-PVT-BASE1 | Sep 21, 10:46:27 PM | 19.48s |

Rows per page: 25 | 1 - 20 of 20

Back Continue



If the directories are not empty, formatting the name directory is expected to fail.



If the directories are already created, creating hdfs /tmp directory is expected to fail.

14. In the next screen, additional steps are suggested by the Cloudera Manager to update the Hue and Hive metastore. Click Finish.

Enable High Availability for HDFS

Getting Started

Assign Roles

Review Changes

Command Details

Final Steps

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.

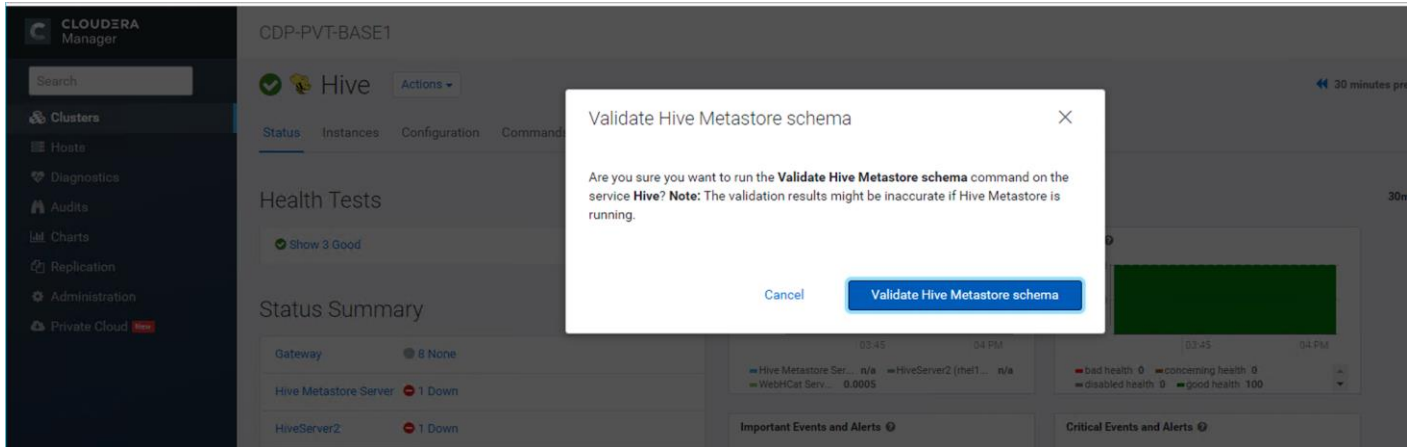
The following subsections explain configuring Hue and Hive for High Availability as needed.

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. Select Actions> Validate the Metastore schema.

The screenshot displays the Cloudera Manager interface for a Hive service instance. The left sidebar shows navigation options like Clusters, Hosts, Diagnostics, Audits, Charts, Replication, Administration, and Private Cloud. The main content area shows the Hive service status as 'Good' with a 'Show 3 Good' indicator. The 'Actions' dropdown menu is open, listing various operations such as Start, Stop, Restart, Add Role Instances, Rename, Enter Maintenance Mode, Deploy Client Configuration, Create Hive User Directory, Create Hive Warehouse Directory, Create Hive Warehouse External Directory, Create Hive Metastore database tables, Create Hive Sys database, **Validate Hive Metastore schema** (highlighted), Update Hive Metastore NameNodes, Upgrade Hive Metastore Database Schema, Create Ranger Plugin Audit Directory, and Download Client Configuration. The right side of the interface features several charts: 'CPU Cores Used' (line chart), 'Health' (bar chart showing 100% good health), 'Important Events and Alerts' (empty chart), 'Critical Events and Alerts' (empty chart), and 'Hive Metastore Server Canary Duration' (line chart showing 3.05s).

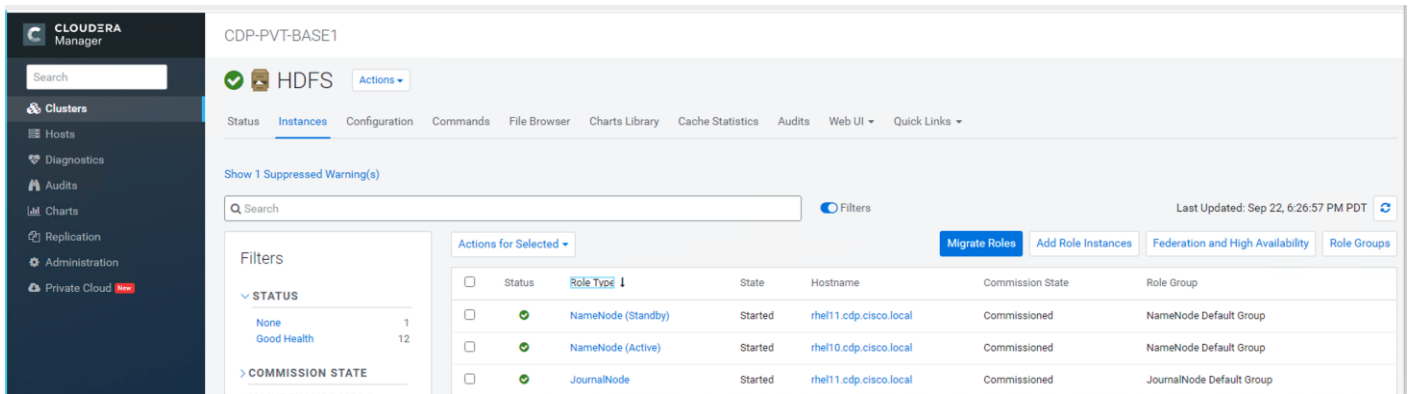


- Restart the Hue and Impala services if stopped prior to updating the Metastore.

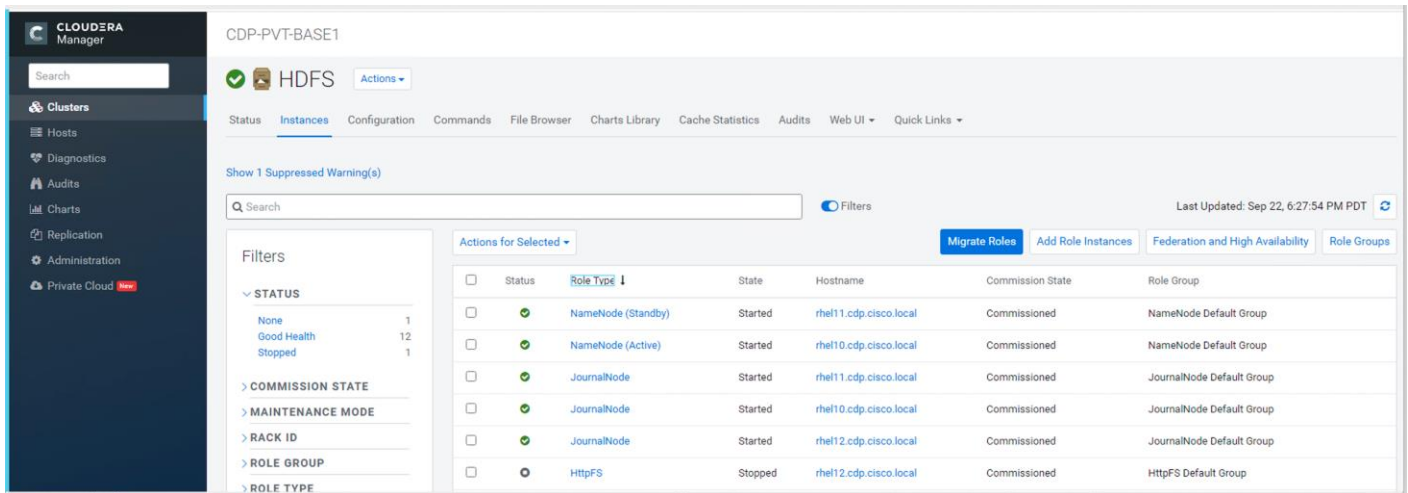
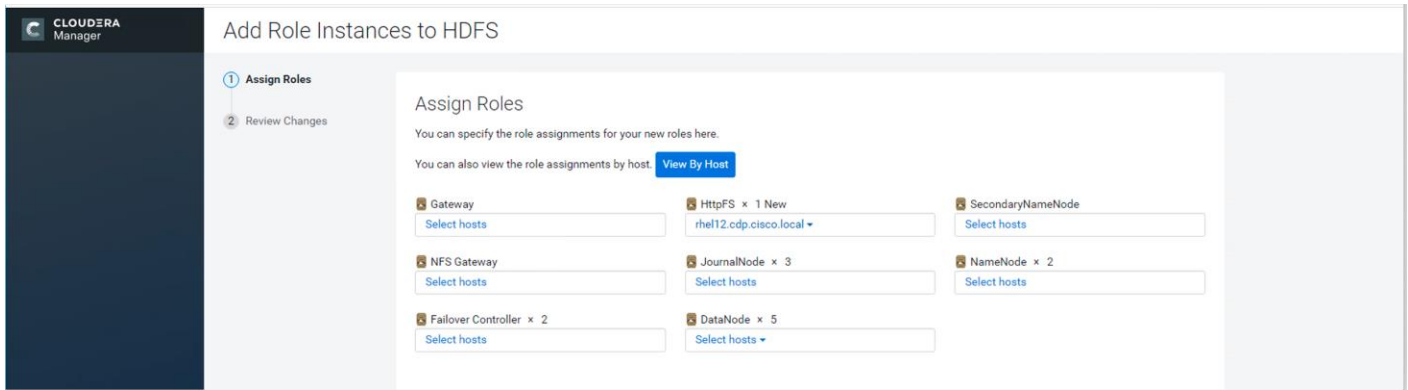
Configure Hue to Work with HDFS High Availability

To configure Hue to work with HDFS High Availability, follow these steps:

- Go to the HDFS service.
- Click the Instances tab.
- Click Add Role Instances.



- Select the text box below the HttpFS field. The Select Hosts dialog displays.
- Select the host on which to run the role and click OK.
- Click Continue.
- 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.

CLUSTER: CDP-PVT-BASE1

Service: Hue

Configuration: HDFS Web Interface Role (webhdfs_url)

| SCOPE | |
|-------------------------|----|
| Hue (Service-Wide) | 68 |
| Hue Server | 48 |
| Kerberos Ticket Renewer | 30 |
| Load Balancer | 36 |

Role: Hue (Service-Wide) HttpFS (rhe112)

Other options: NameNode (rhe111), NameNode (rhe110)

Note: HTTPFS role is recommended for Web interface if HDFS is HA or federated.

Warning: 'NoneType' object has no attribute 'timeoutout'

Job Name: DistCp

Source: \$(nameNode1)/path/to/input.txt

Destination: \$(nameNode2)/path/to/output.txt

Variables: (empty)

Query History | Saved Queries (You don't have any saved queries.)



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 information 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:

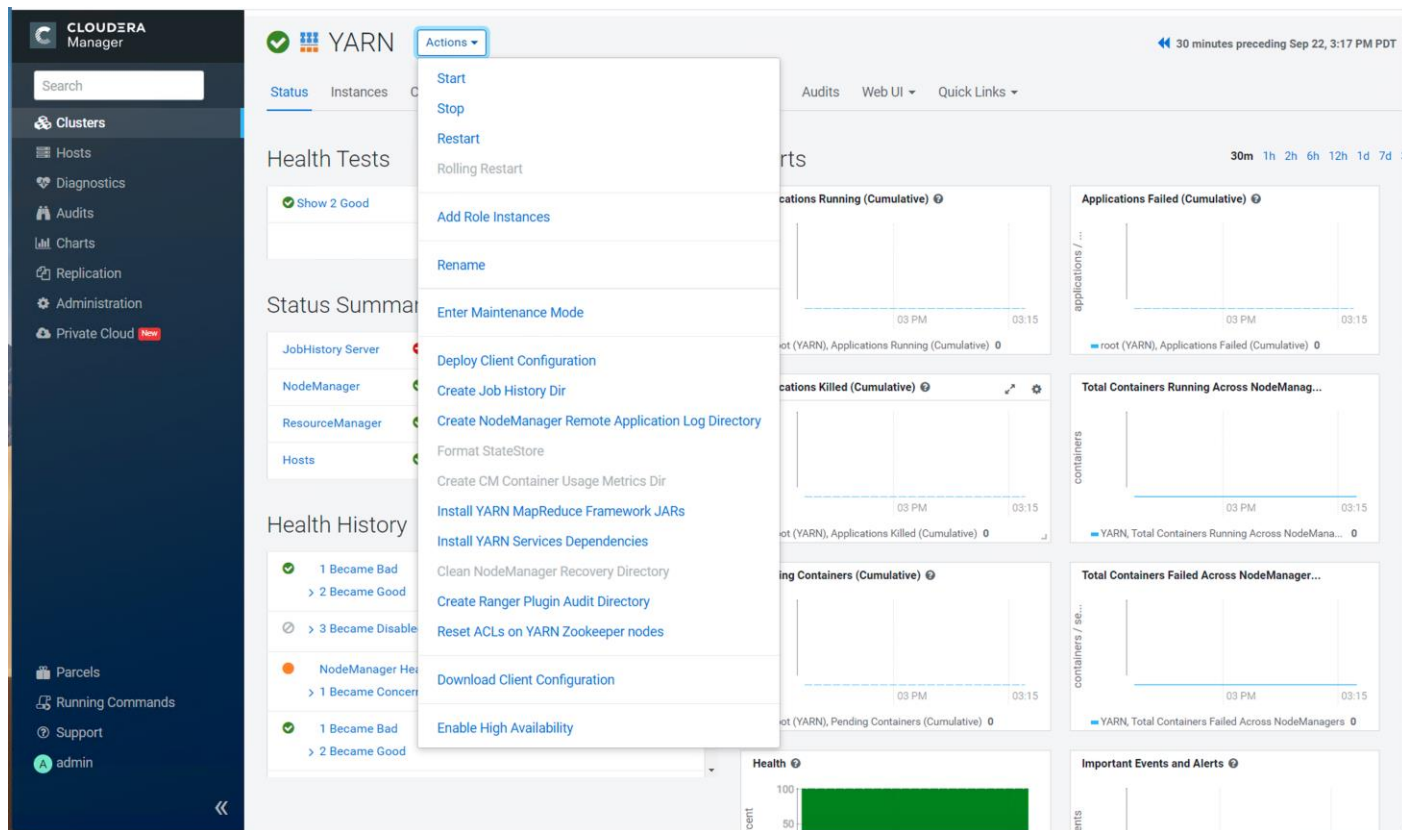
https://www.cloudera.com/documentation/enterprise/latest/topics/cdh_hag_rm_ha_config.html#xd_583c10bfd326ba--43d5fd93-1410993f8c2--7f77

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.

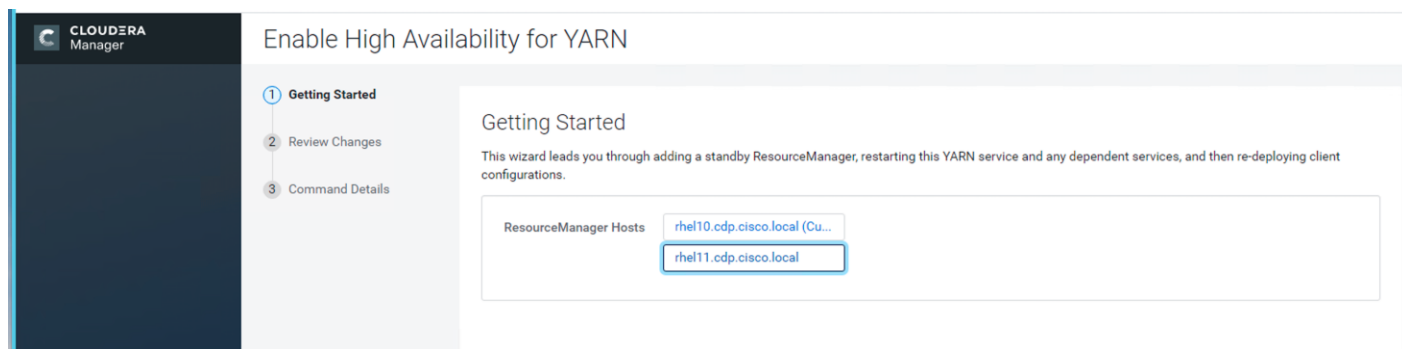


A screen showing the hosts that are eligible to run a standby ResourceManager displays.



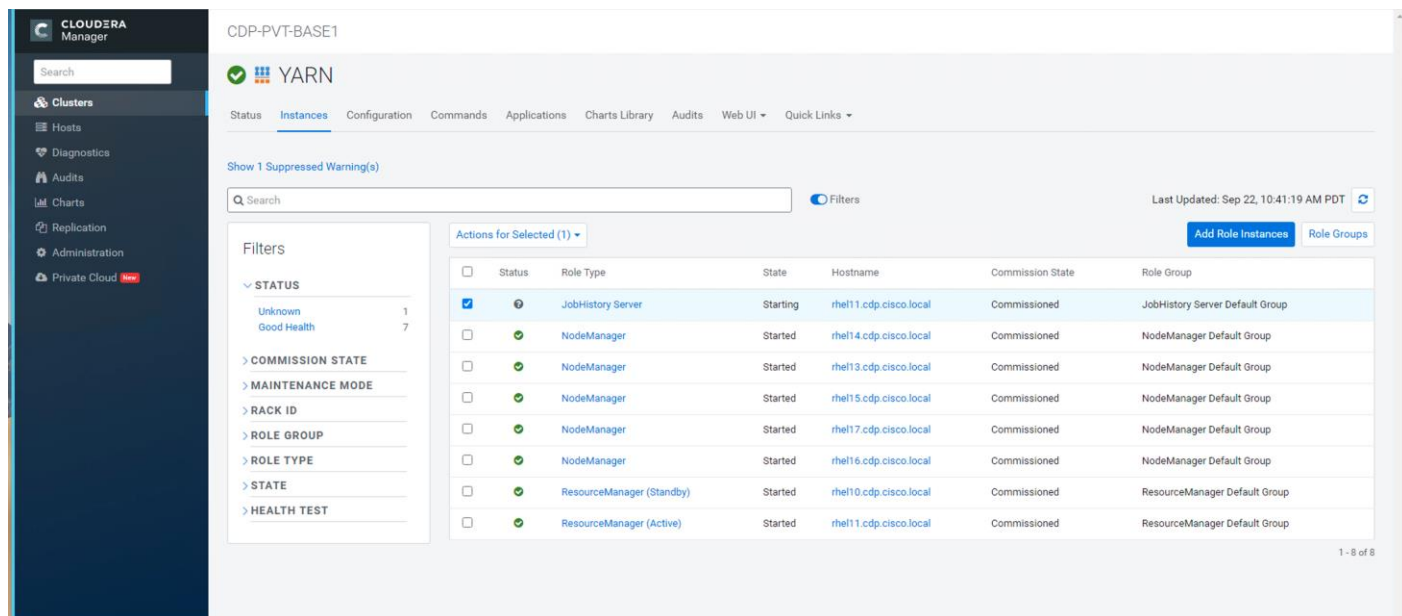
The host where the current ResourceManager is running is not available as a choice.

3. Select the host (rhel11) where the standby ResourceManager is to be installed and click Continue.



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.

4. Click Finish once the installation is completed successfully and verify the RM roles.



Configure Yarn (MR2 Included) and HDFS Services

The parameters in [Table 6](#) and [Table 7](#) 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 6. YARN

| Service | Value |
|--|--|
| mapreduce.output.fileoutputformat.compress.type | BLOCK |
| mapreduce.output.fileoutputformat.compress.codec | org.apache.hadoop.io.compress.DefaultCodec |
| mapreduce.map.output.compress.codec | org.apache.hadoop.io.compress.SnappyCodec |
| mapreduce.map.output.compress | True |
| zlib.compress.level | BEST_SPEED |
| mapreduce.task.io.sort.factor | 64 |
| mapreduce.map.sort.spill.percent | 0.9 |
| mapreduce.reduce.shuffle.parallelcopies | 20 |

| Service | Value |
|---|--------------------------------|
| yarn.nodemanager.resource.memory-mb | 320GB |
| yarn.nodemanager.resource.cpu-vcores | 64 |
| yarn.scheduler.maximum-allocation-vcores | 64 |
| yarn.scheduler.maximum-allocation-mb | 320GB |
| mapreduce.task.io.sort.mb | 2047 |
| mapreduce.job.reduce.slowstart.completedmap | 0.8 |
| yarn.app.mapreduce.am.resource.cpu-vcores | 1 |
| mapreduce.map.memory.mb | 5G |
| mapreduce.reduce.memory.mb | 5G |
| mapreduce.job.heap.memory-mb.ratio | 0.8 |
| mapreduce.job.shuffle.merge.percent | 0.95 |
| mapreduce.job.shuffle.input.buffer.percent | 0.7 |
| mapreduce.job.reduce.input.buffer.percent | 0.7 |
| mapreduce.input.fileinputformat.split.minsize | 4096000000 |
| mapreduce.ifile.readahead.bytes | 16777216 |
| mapreduce.inmem.merge.threshold | 0 |
| Enable Optimized Map-side Output Collector | Enable - Gateway Default Group |

Table 7. HDFS

| Service | Value |
|---------------------------------------|--------|
| dfs.datanode.failed.volumes.tolerated | 6 |
| dfs.datanode.du.reserved | 50 GiB |

| Service | Value |
|---|----------|
| 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.

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 $\max(384, 0.10 * \text{spark.executor.memory})$.
- 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.default.parallelism=10000
spark.driver.memoryOverhead=4096
spark.executor.memoryOverhead=4096
spark.executor.extraJavaOptions=-XX:+UseParallelGC -XX:ParallelGCThreads=4
spark.shuffle.file.buffer=1024k
spark.broadcast.compress=true
spark.shuffle.compress=true
spark.io.compression.codec=org.apache.spark.io.SnappyCompressionCodec
spark.io.compression.snappy.blockSize=512k
```

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 transform-

mation 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 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 disks 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.kryo.serializer.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

The following are the 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 the 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."
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 achieve a 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.
- CDP on CIDP delivers new approach to data where machine learning intelligently auto scale workloads up and down for more cost-effective use of private cloud infrastructure.

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 Data Intelligence Platform, see <https://www.cisco.com/c/dam/en/us/products/servers-unified-computing/ucs-c-series-rack-servers/solution-overview-c22-742432.pdf>
- 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>
- Cloudera Data Platform Data Center 7.1.3 release note, see <https://docs.cloudera.com/cdp-private-cloud-base/7.1.3/concepts-cloudera-manager.html>

Bill of Materials

This section provides the BoM for the 12 Nodes Hadoop Base Rack. See [Table 8](#) for BOM for the Hadoop Base rack and [Table 9](#) for Red Hat Enterprise Linux License.

Table 8. Bill of Materials for Cisco UCS C4200-SFF Hadoop Nodes Base Rack

| Part Number | Description | Quantity |
|--|--|---------------|
| UCSC-C4200-SFF | UCS C4200 Chassis 24 SFF HDD/SSD SAS/SATA | 12 |
| UCSC-C125 | UCS C125 Base Compute Node Tray | 48 |
| UCSC-RAID-C125KIT | UCS C125 9460-8i RAID kit | 48 |
| UCSC-PSU3-2400W | Cisco UCS 2400W AC Power Supply | 24 |
| UCSC-RAILB-C4200 | UCS C4200 Rack Rail | 24 |
| UCS-SD38T61X-EV / UCS-SD76TBMS4-EV | 7.6TB 2.5-inch Enterprise Value 6G SATA SSD | 288 |
| UCSC-PCIE-C25Q-04 | Cisco UCS VIC 1455 Quad Port 10/25G SFP28 CNA PCIE | 48 |
| UCS-M2-240GB | 240GB SATA M.2 | 96 |
| UCS-MR-X64G2RW | 64GB RDIMM DRx4-3200-MHz RDIMM/2Rx4/1.2v | |
| [8 x 64 GB DDR4 (512GB) per Server Node] | 384 | |
| UCSC-SAS9460-8I | Cisco 12G 9460-8i Raid controller with 2GB cache | 48 |
| UCS-FI-6454-U | UCS Fabric Interconnect 6454 Configured model: UCS 6454 1RU FI, with no PSU, with 54 ports and includes 18x10/25-Gbps and 2x40/100-Gbps port licenses | UCS-FI-6454-U |
| RHEL-2S2V-3A | Red Hat Enterprise Linux (1-2 CPU, 1-2 VN); 3-Yr Support Req | 48 |
| CON-ISV1-EL2S2V3A | ISV 24X7 RHEL Server 2Socket-OR-2Virtual; ANNUAL List Price | 48 |

| Part Number | Description | Quantity |
|--------------------|---|----------|
| CAB-N5K6A-NA | Power Cord, 200/240V 6A North America | 24 |
| RACK2-UCS2 | Cisco R42612 standard rack, w/side panels | 1 |
| CON-SNT-RCK2UCS2 | SNTC 8X5XNBD, Cisco R42612 standard rack, w side panels | 1 |
| UCS-L-6400-25GC | | |
| | UCS 6400 series ONLY Fabric Int 1 Port 10/25 Gbps/FC port license C-direct only (used to connect directly from FI 6454 to C220, C240, C460, C480, and/or C4200) | 2 |
| CON-3ETOP-SPFI6454 | 3YR ETSP 24X7X4OS (Not sold standalone) UCS 6454 FI | 1 |
| UCS-PSU-6332-AC | UCS 6332/6454 Power Supply/100-240VAC | 4 |
| UCS-SID-INFR-BD | Big Data and Analytics Platform (Hadoop/IoT/ITOA/AI/ML) | 48 |
| UCS-SID-WKL-BD | Big Data and Analytics (Hadoop/IoT/ITOA) | 48 |



NameNode was configured with 6 x 7.6TB SAS SSDs.

Table 9. Red Hat Enterprise Linux License

| Part Number | Description | Quantity |
|-------------------|---|----------|
| RHEL-2S2V-3A | Red Hat Enterprise Linux | 48 |
| CON-ISV1-EL2S2V3A | 3-year Support for Red Hat Enterprise Linux | 48 |



For Cloudera Data Platform Private Cloud Base software licensing requirement, contact [Cloudera Data Platform software - Sales](#)

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:

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

The screenshot shows the UCS Manager interface. On the left is a navigation menu with icons for Home, Storage, Sub-Organizations, and UCS-HDP. The 'Disk Group Policies' menu item is highlighted, and a blue callout box with the text 'Create Disk Group Policy' is positioned over it. The main content area shows the breadcrumb path: Storage / Storage Policies / root / Sub-Organizations / UCS-HDP / Disk Group Policies. Below this, there are options for 'Advanced Filter', 'Export', and 'Print'. A table lists existing policies: BootLun, CDSW-R10, NameNode_R10, and S3260-BootLUN.

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

The 'Create Disk Group Policy' dialog box is shown. It has a title bar with a question mark and a close button. The 'Name' field contains 'Boot-M2-HWRaid' and the 'Description' field contains 'Boot policy for Cisco UCS M.2 HW Raid controller'. The 'RAID Level' is set to 'RAID 1 Mirrored'. There are two radio buttons: 'Disk Group Configuration (Automatic)' and 'Disk Group Configuration (Manual)', with the latter being selected. Below this is a table with columns 'Slot Number', 'Role', and 'Span ID', which is currently empty with the text 'No data available'. At the bottom, there are 'Add', 'Delete', and 'Info' buttons. The 'Virtual Drive Configuration' section shows 'Strip Size (KB)' set to 'Platform Default'. 'OK' and 'Cancel' buttons are at the bottom right.



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 |
| Storage Controller SATA 2 | | | | | | | |

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

Create Disk Group Policy

Name :

Description :

RAID Level :

Disk Group Configuration (Automatic)

Disk Group Configuration (Manual)

Advanced Filter Export Print

| Slot Number |
|-------------|
| 253 |

Virtual Drive Configuration

Strip Size (KB) :

OK Cancel

Create Local Disk Configuration Reference

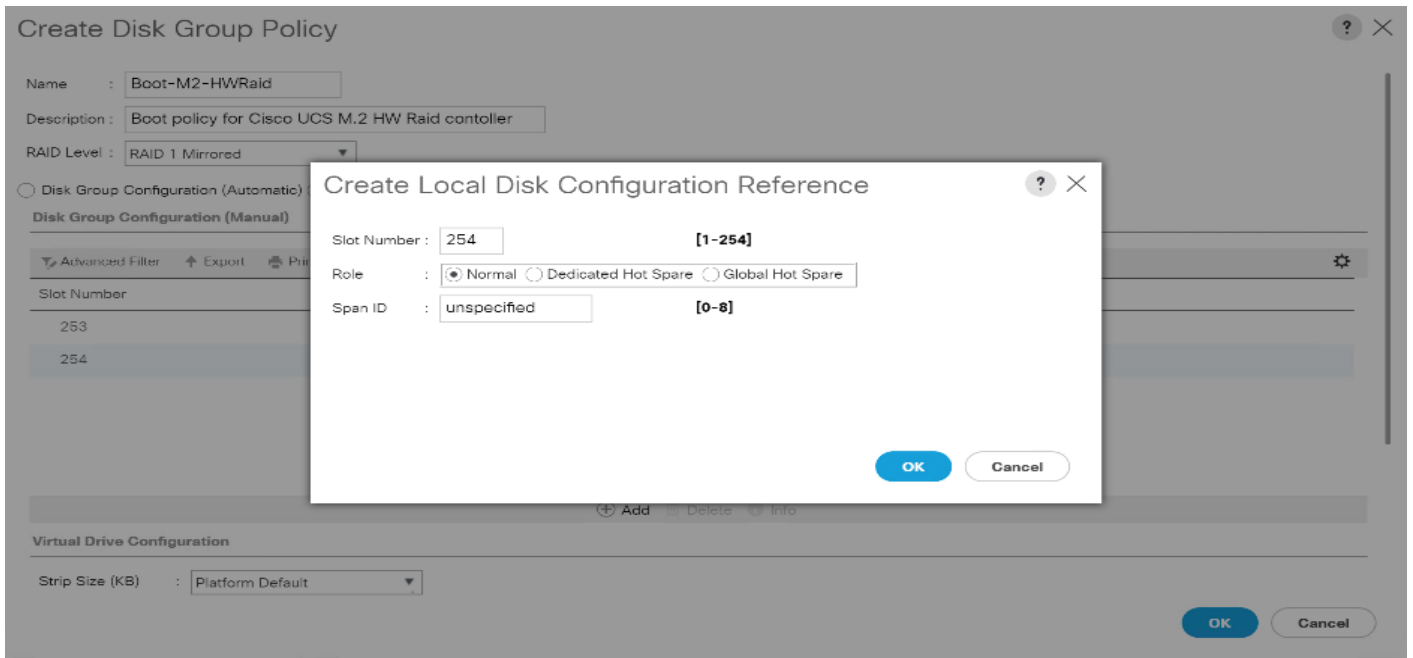
Slot Number : [1-254]

Role : Normal Dedicated Hot Spare Global Hot Spare

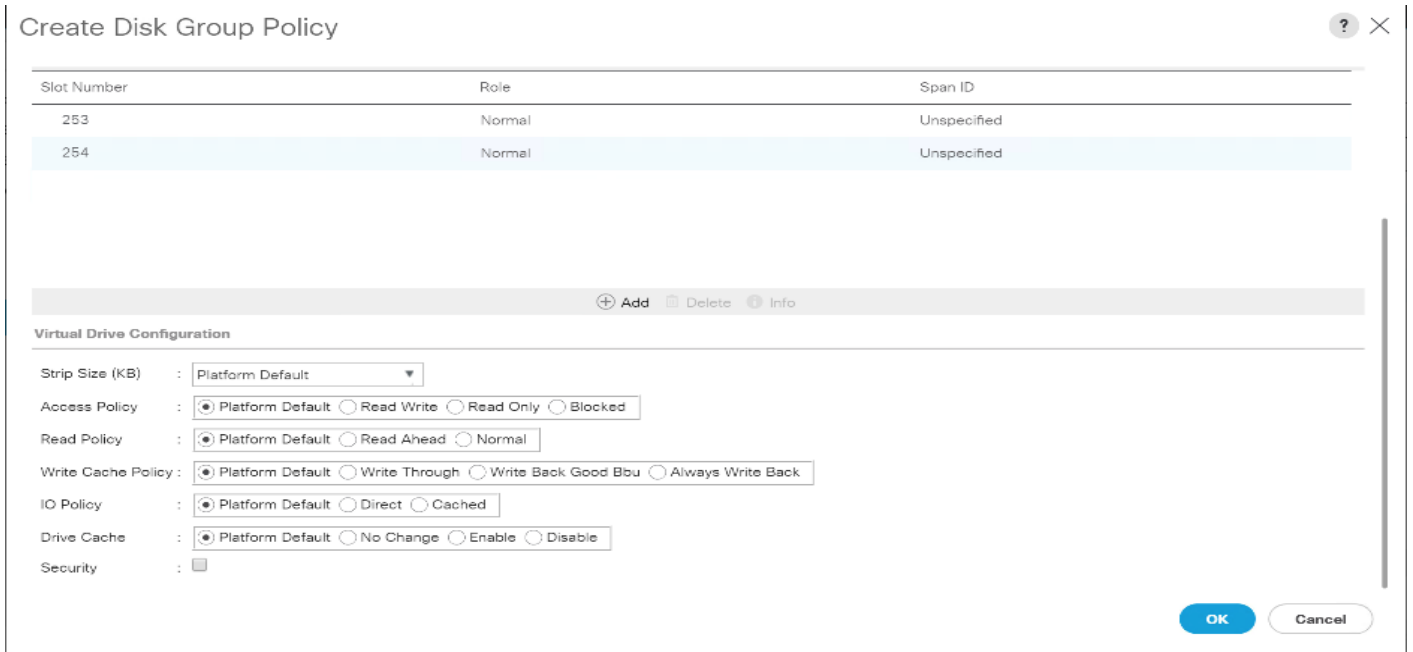
Span ID : [0-8]

OK Cancel

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



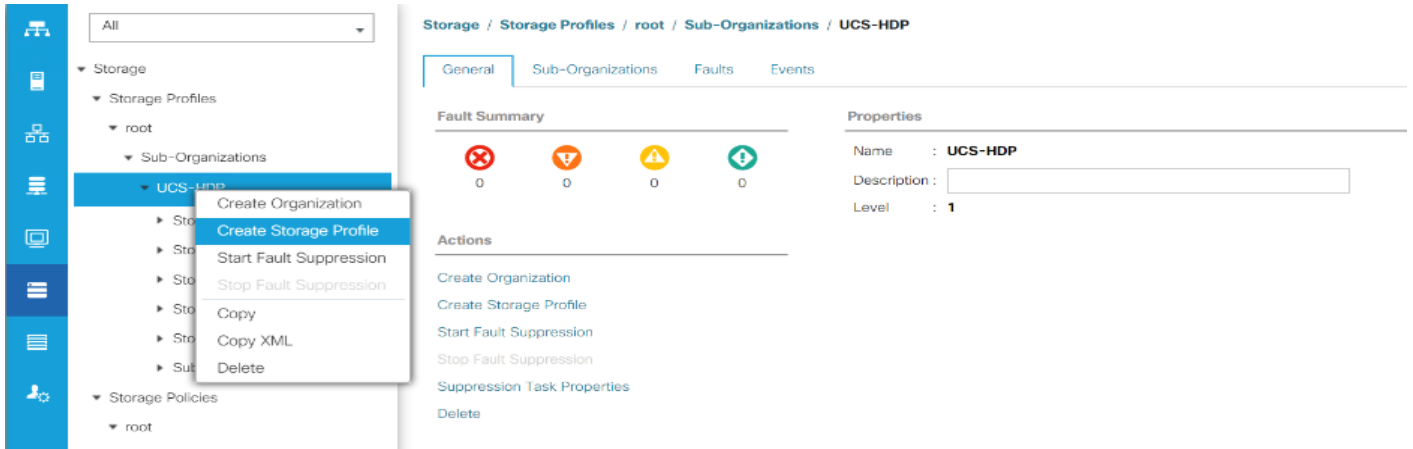
5. In Virtual Drive Configuration section leave all option as Platform Default. Click OK.



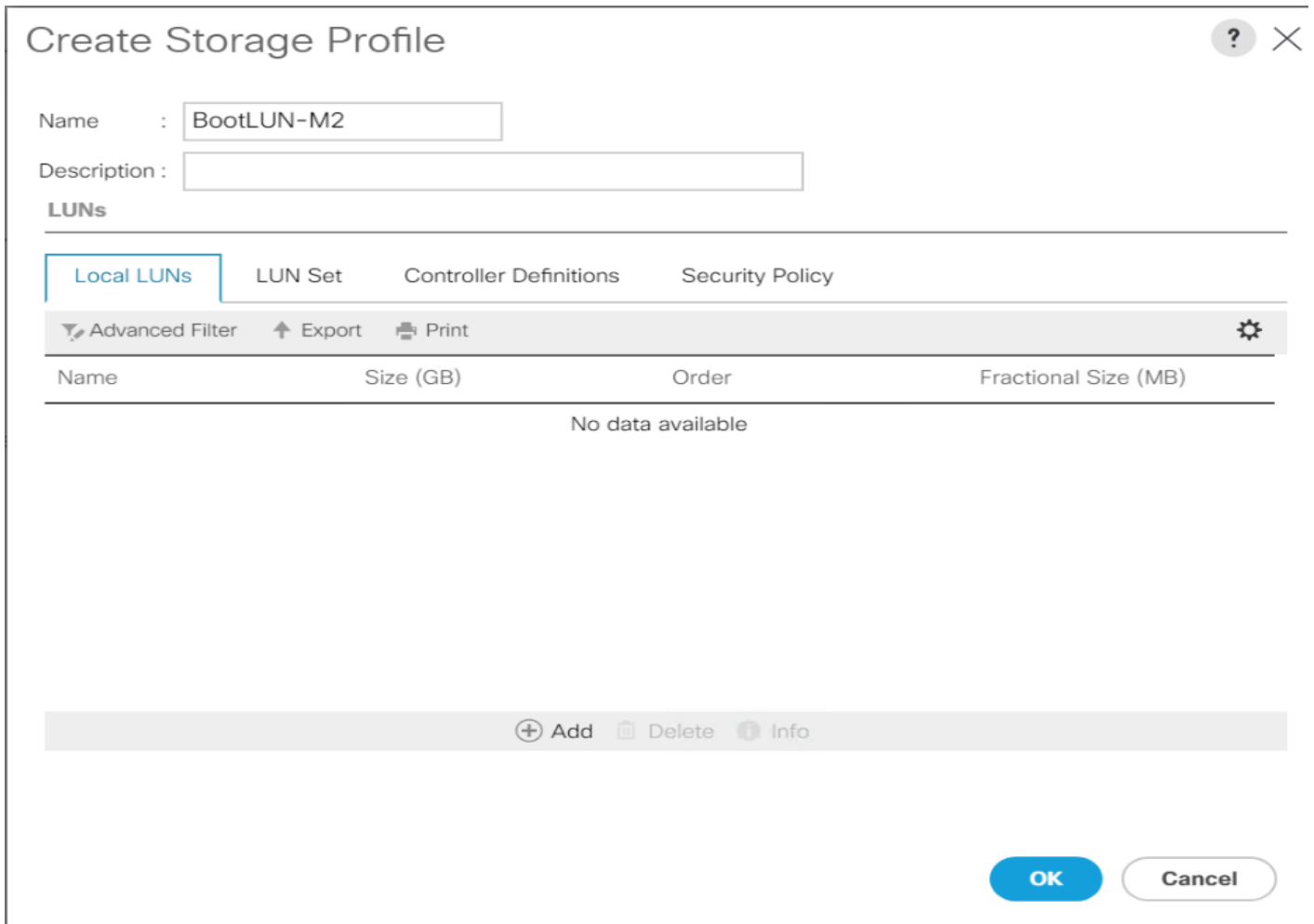
Configure Storage Profile

To configure the storage profile, follow these steps:

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



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



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 :

Size (GB) : **[0-245760]**

Fractional Size (MB) :

Auto Deploy : Auto Deploy No Auto Deploy

Expand To Available :

Select Disk Group Configuration : [Create Disk Group Policy](#)

OK

Cancel

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 49. Example of the Service Profile Associated to a Cisco UCS C125 M5 Server with Cisco UCS-M2-HWRAID and 2 240GB M.2 SATA SSD Installed

General | **Storage** | Network | iSCSI vNICs | vMedia Policy | Boot Order | Virtual Machines | FC Zones | Policies | Server Details | CIMC Sessions | FSM | VIF >

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**
 Storage Profile Instance : org-root/org-UCS-BDA/profile-BootLUN-M2

Local LUNs | LUN Set | Controller Definitions | Security Policy | Faults

Advanced Filter | Export | Print

| 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

Actions

Set LUN Name
 Rename Referenced LUN
 Set Online
 Set Undeployed
 Claim Orphaned LUN

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 50. Example of Virtual Drive Created from 2 M.2 SATA SSD

General | Inventory | Virtual Machines | Hybrid Display | Installed Firmware | SLL Logs | CIMC Sessions | VIF Paths | Power Control Monitor | Health | Diagnostics | Faults | Events | FSM | Statistics | Temperatures | Power

Motherboard | CIMC | CPUs | Coprocessor Cards | GPUs | PG Switch | Memory | Adapters | HBAs | NICs | iSCSI vNICs | **Storage** | Persistent Memory

Controller | LUNs | **Disks** | SAS Expander | Security

Advanced Filter | Export | Print

| Name | Size (MB) | Raid Type | Config State | Deploy Action | Operability | Presence | Bootable |
|---------------------------|-----------|-----------------|--------------|---------------|-------------|----------|----------|
| Storage Controller PCH 6 | | | | | | | |
| Storage Controller SAS 1 | | | | | | | |
| Storage Controller SATA 2 | | | | | | | |
| Virtual Drive BootLUN-M2 | 228672 | RAID 1 Mirrored | Applied | No Action | Operable | Equipped | True |
| Storage Controller SATA 7 | | | | | | | |

Actions

Rename
 Delete
 Set Transport Ready
 Hide Virtual Drive
 Clear Transport Ready
 Unhide Virtual Drive
 Secure Virtual Drive

Properties

Virtual Drive Name : **BootLUN-M2** | Size (MB) : **228672**
 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**
 Rotatable : **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 : **sysback-ctrl-11**

General Inventory Virtual Machines Hybrid Display Installed Firmware SPI Logs CIMC Sessions VFP Paths Power Control Monitor Health Diagnostics Faults Events FSM Statistics Temperatures Power

Motherboard CIMC CPUs Coprocessor Cards OPUS PCI Switch Memory Adapters HBAs NICs iSCSI vMCS Storage Persistent Memory

General Logs Disks RAID Firmware Security

Advanced Filter Export Print

| 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 | 173619C49ABE | Operable | Online | Equipped | SSD | False |
| Disk 254 | 228936 | 1719170523BE | Operable | Online | Equipped | SSD | False |
| Storage Controller SATA 3 | | | | | | | |

General FSM Statistics

Actions

- Set Unconfigured Bad to Good
- Prepare for Removal
- Undo Prepare for Removal
- Set JBOD Mode
- Mark as Dedicated Hot Spare
- Remove Hot Spares
- Set JBOD to Unconfigured Good
- Enable Encryption
- Secure Erase
- Secure Erase Foreign Configuration
- Turn on Locator LED

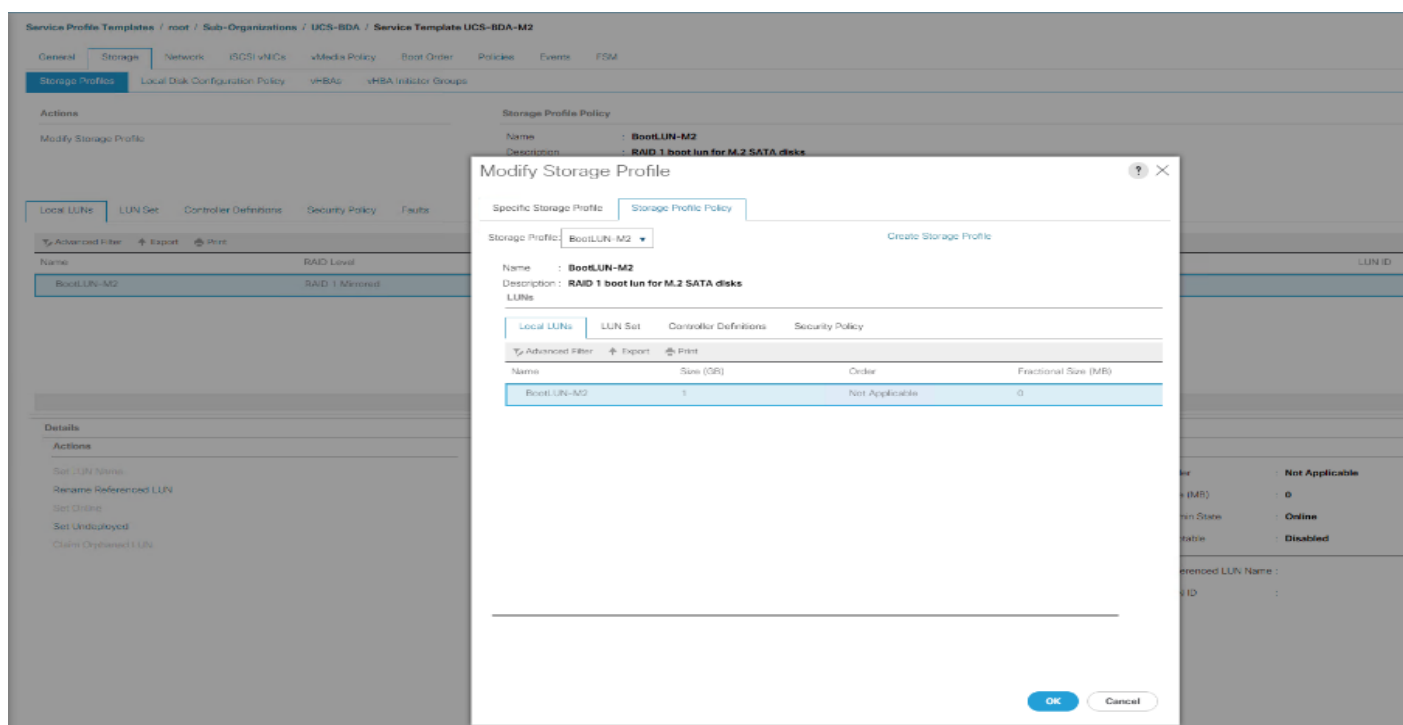
Properties

| | | | |
|---------------------|-------------------------|-----------------------|----------------|
| ID | : 253 | PID | : UCS-M2-240GB |
| Vendor | : Micron | VID | : V01 |
| Serial | : 173619C49ABE | 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 | | |
| 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:

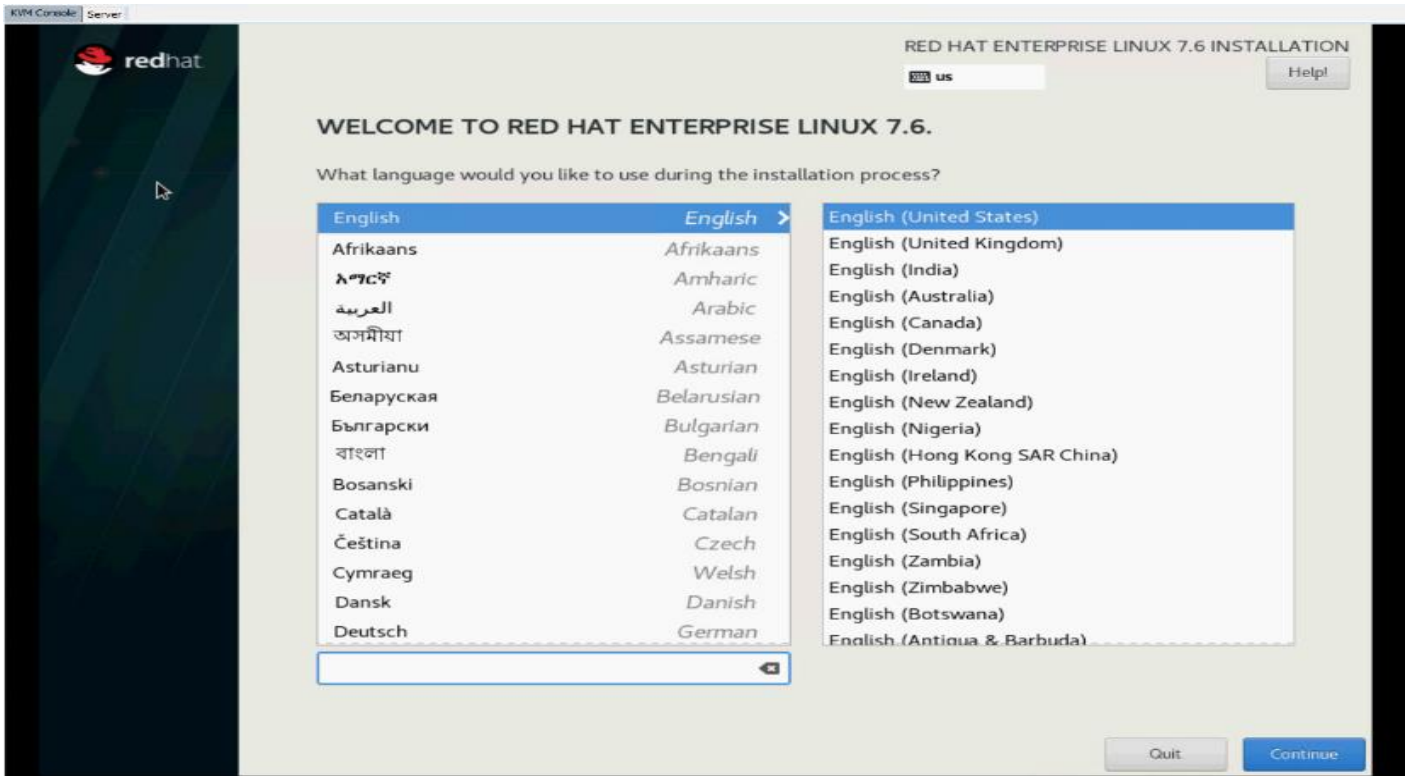
1. Go to Service Profile Template.
2. Select Storage tab in Service Profile Template.
3. Select Storage Profile Tab. Click Modify Storage Profile.
4. From the Storage Profile drop-down list, select Storage Profile for Cisco Boot Optimized RAID Controller.
5. If updating a Service Profile Template, once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



Install RHEL 7.8 on Cisco Optimized M.2 RAID Controller

To install Red Hat Enterprise Linux 7.8 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:

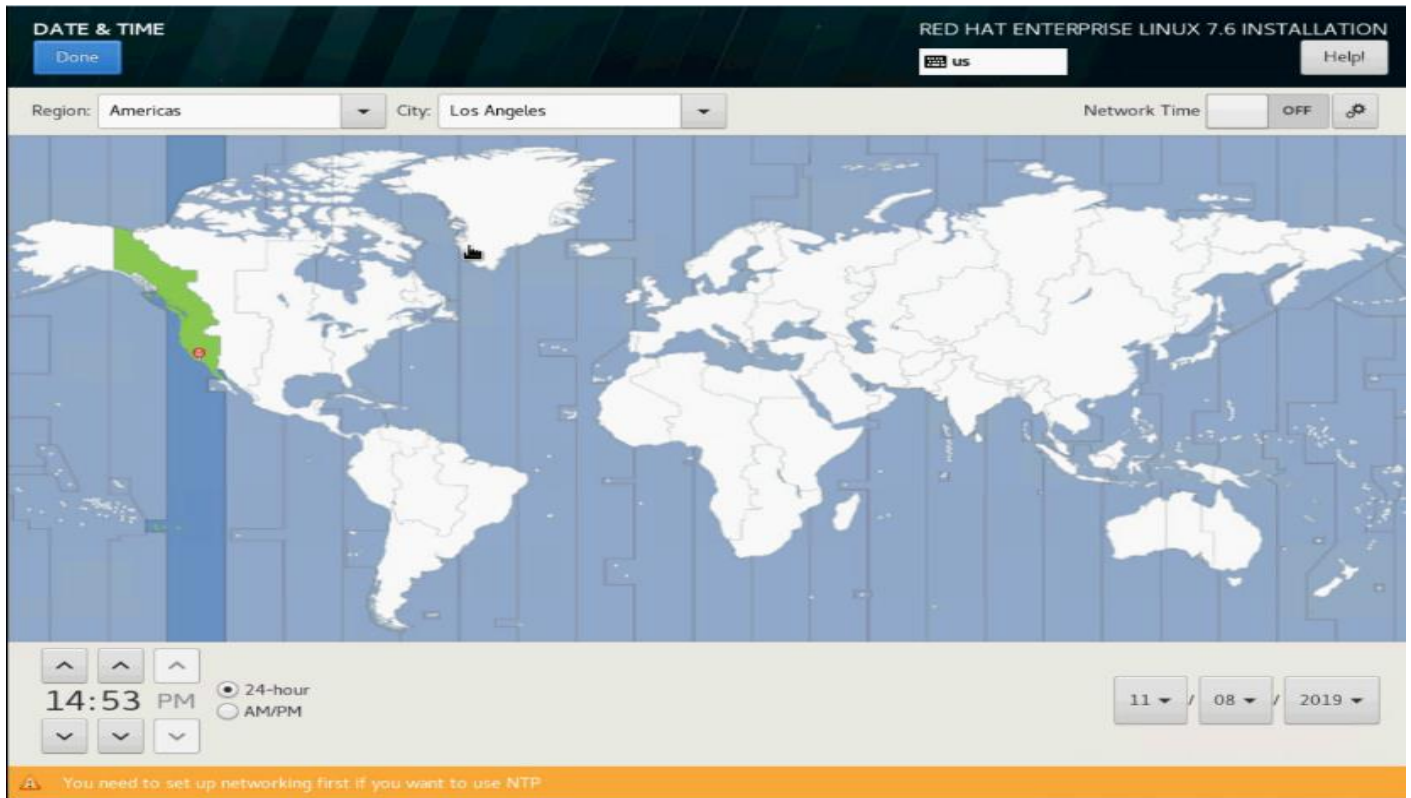
1. On the Welcome screen, select a language and click Continue.



2. Select DATE & TIME.



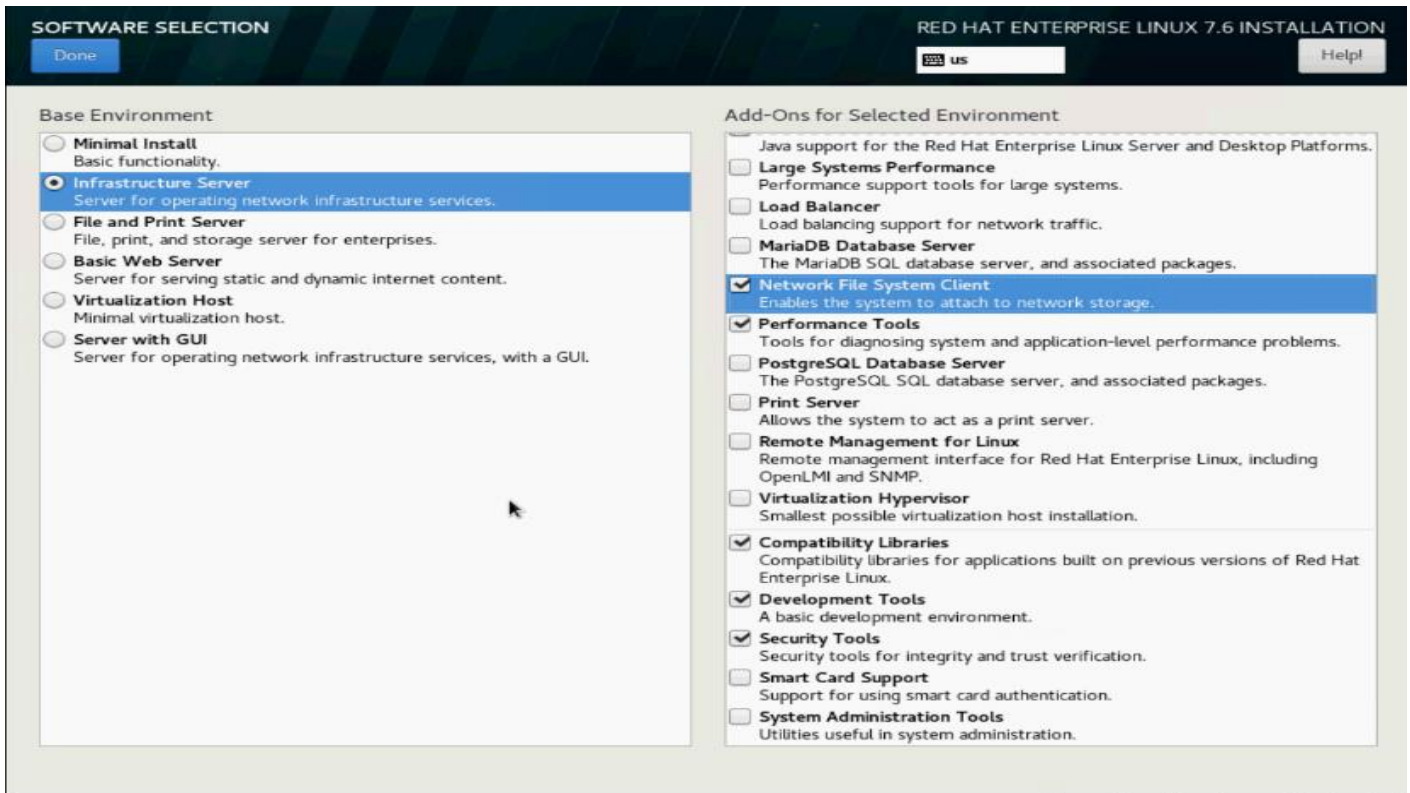
3. Select region and City.



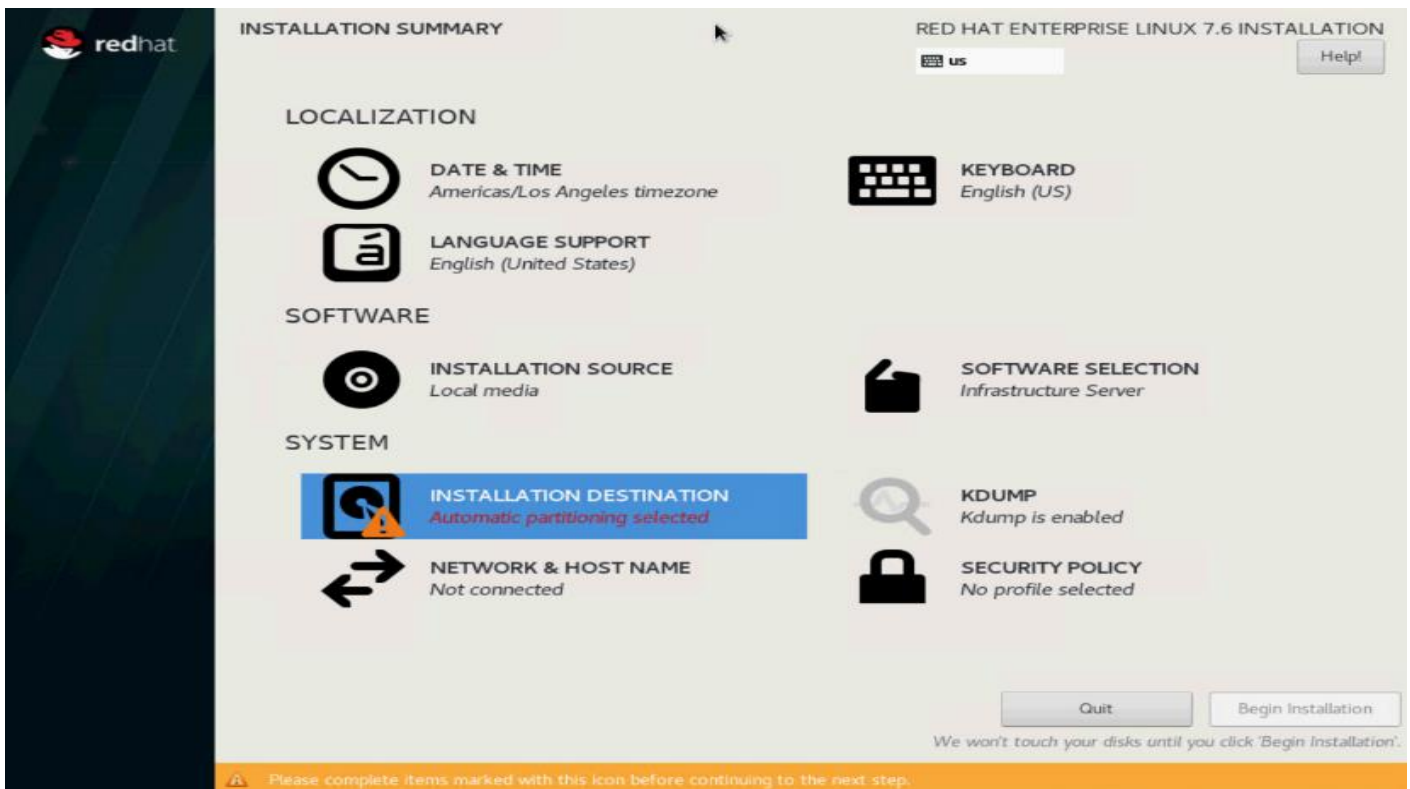
4. Select SOFTWARE SELECTION.



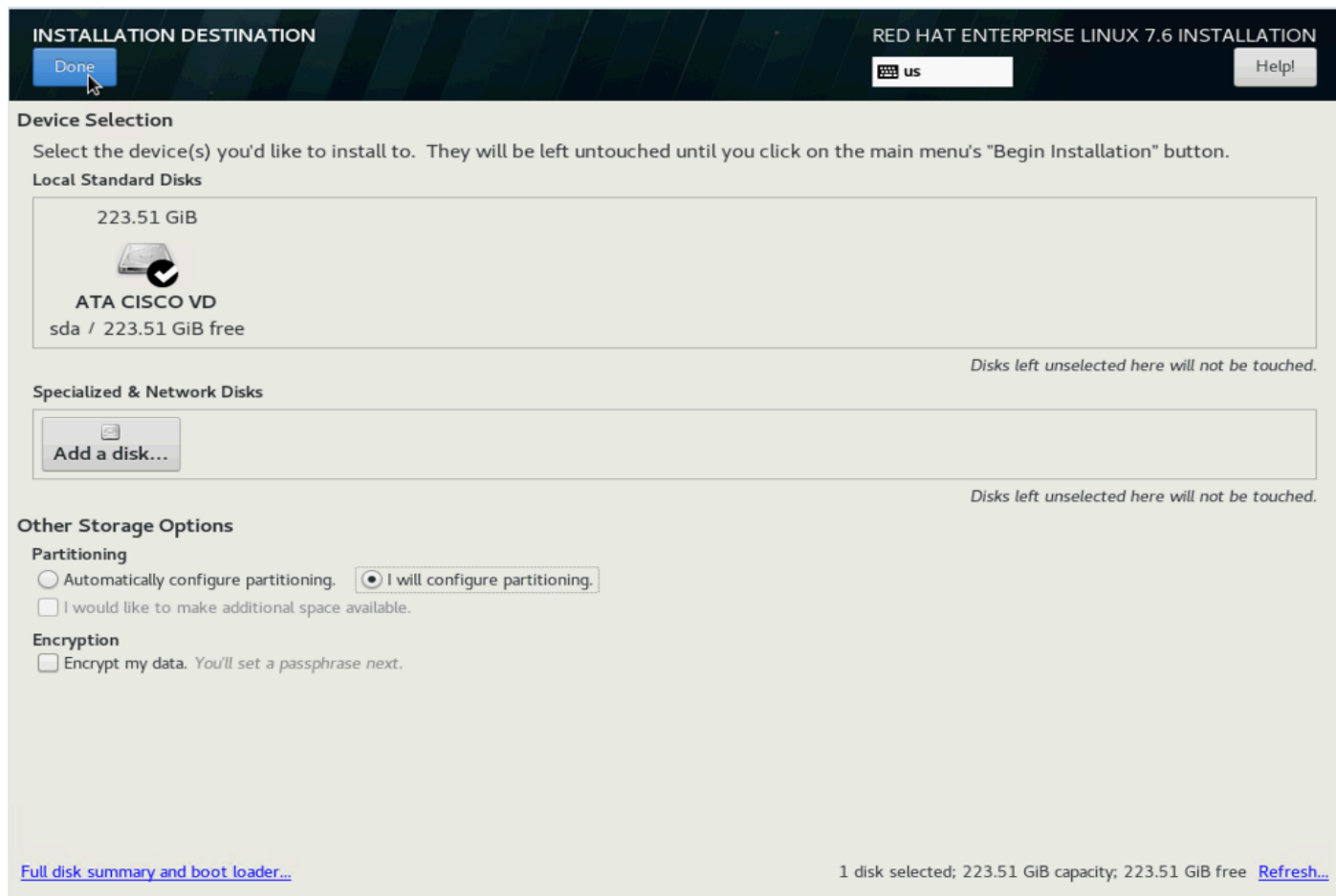
5. Select Infrastructure Server in Base Environment. For Add-Ons for the selected environment, choose:
 - a. Network File System Client
 - b. Performance Tools
 - c. Compatibility Libraries
 - d. Development Tools
 - e. Security Tools



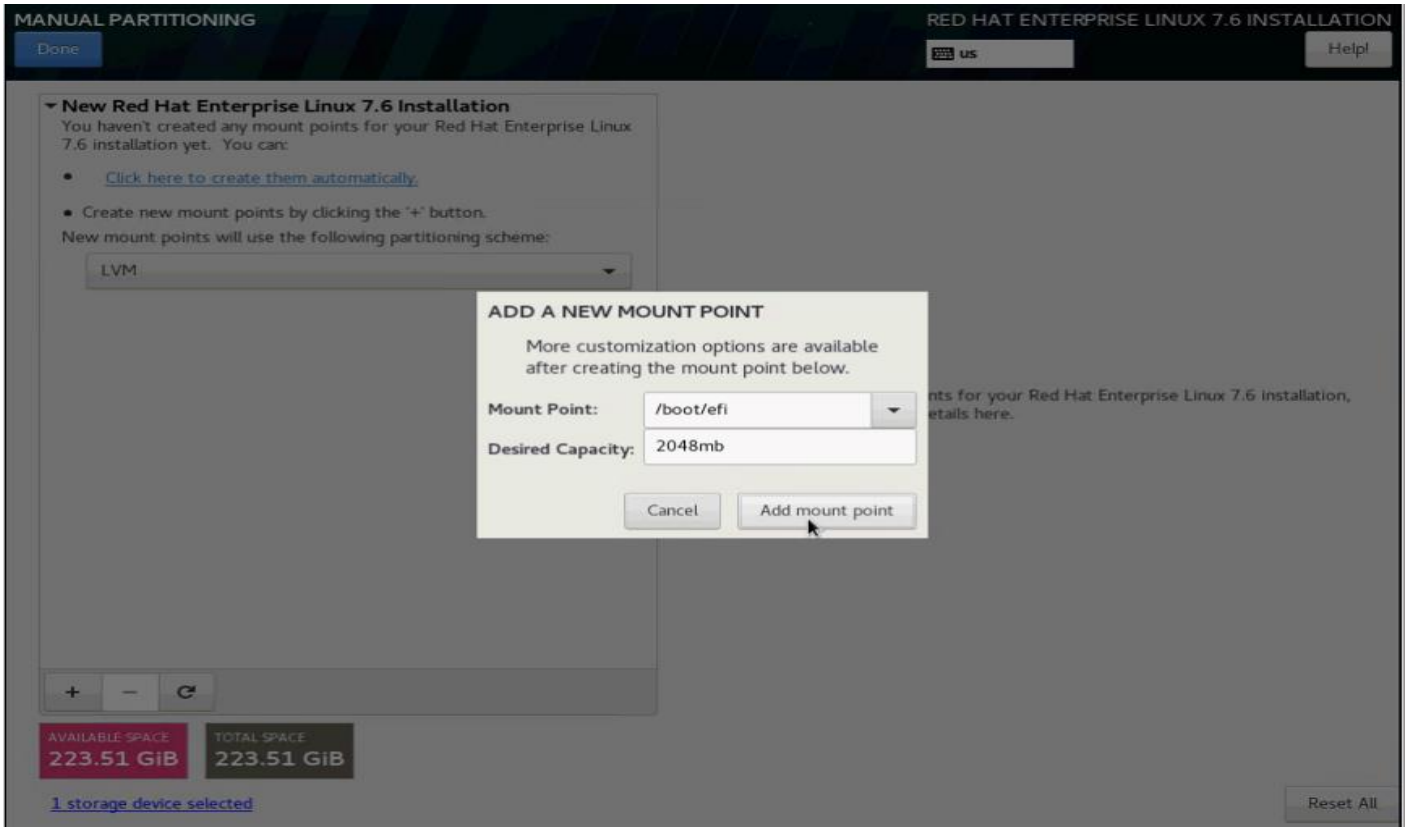
6. Select Installation Destination.



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



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.



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

Table 10. Mount Point and Desired Capacity for RHEL Installation

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

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

MANUAL PARTITIONING RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

Done us Help

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

1 storage device selected

rhel_rhel10-root

Mount Point: /

Device(s): ATA CISCO VD (sda)

Desired Capacity: 217.78 GiB

Device Type: LVM Encrypt

File System: xfs Reformat

Volume Group: rhel_rhel10 (0 B free)

Label:

Name: root

Update Settings

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

Reset All

11. Click Accept Changes.

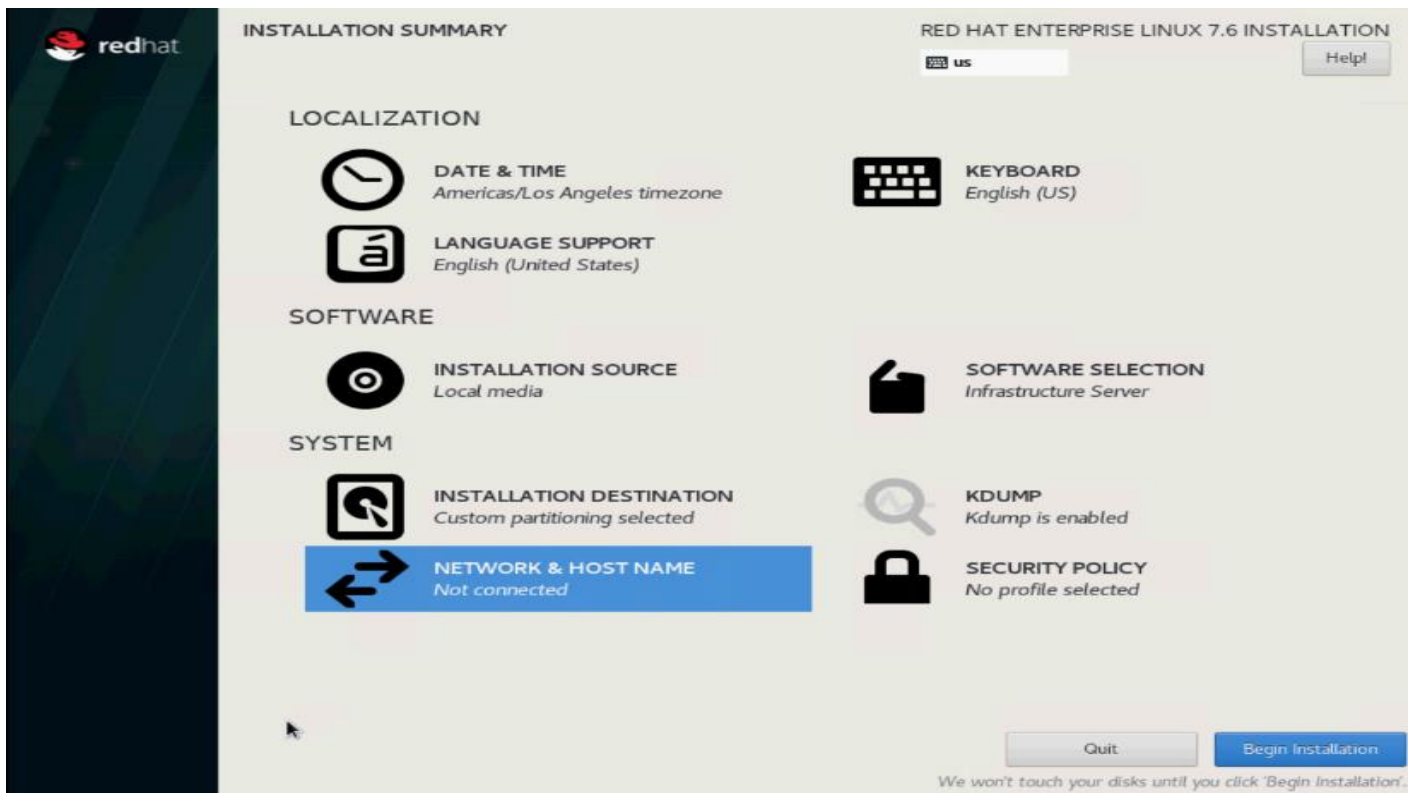
SUMMARY OF CHANGES

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

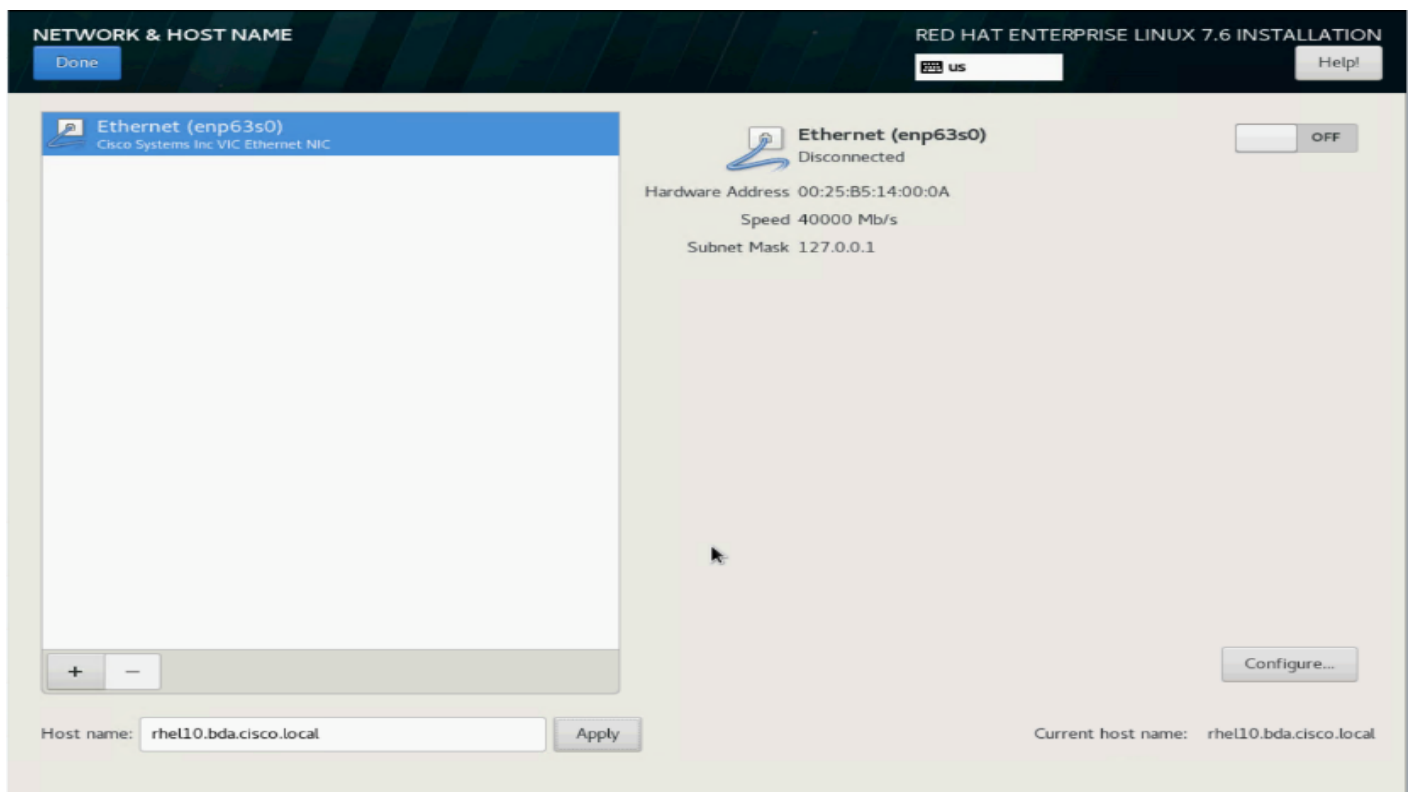
| Order | Action | Type | Device Name | Mount point |
|-------|----------------|-----------------------|------------------|-------------|
| 1 | Destroy Format | Unknown | sda | |
| 2 | Create Format | partition table (GPT) | sda | |
| 3 | Create Device | partition | sda1 | |
| 4 | Create Format | EFI 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 | lvm lv | rhel_rhel10-swap | |
| 11 | Create Format | swap | rhel_rhel10-swap | |
| 12 | Create Device | lvm lv | rhel_rhel10-root | |

Cancel & Return to Custom Partitioning Accept Changes

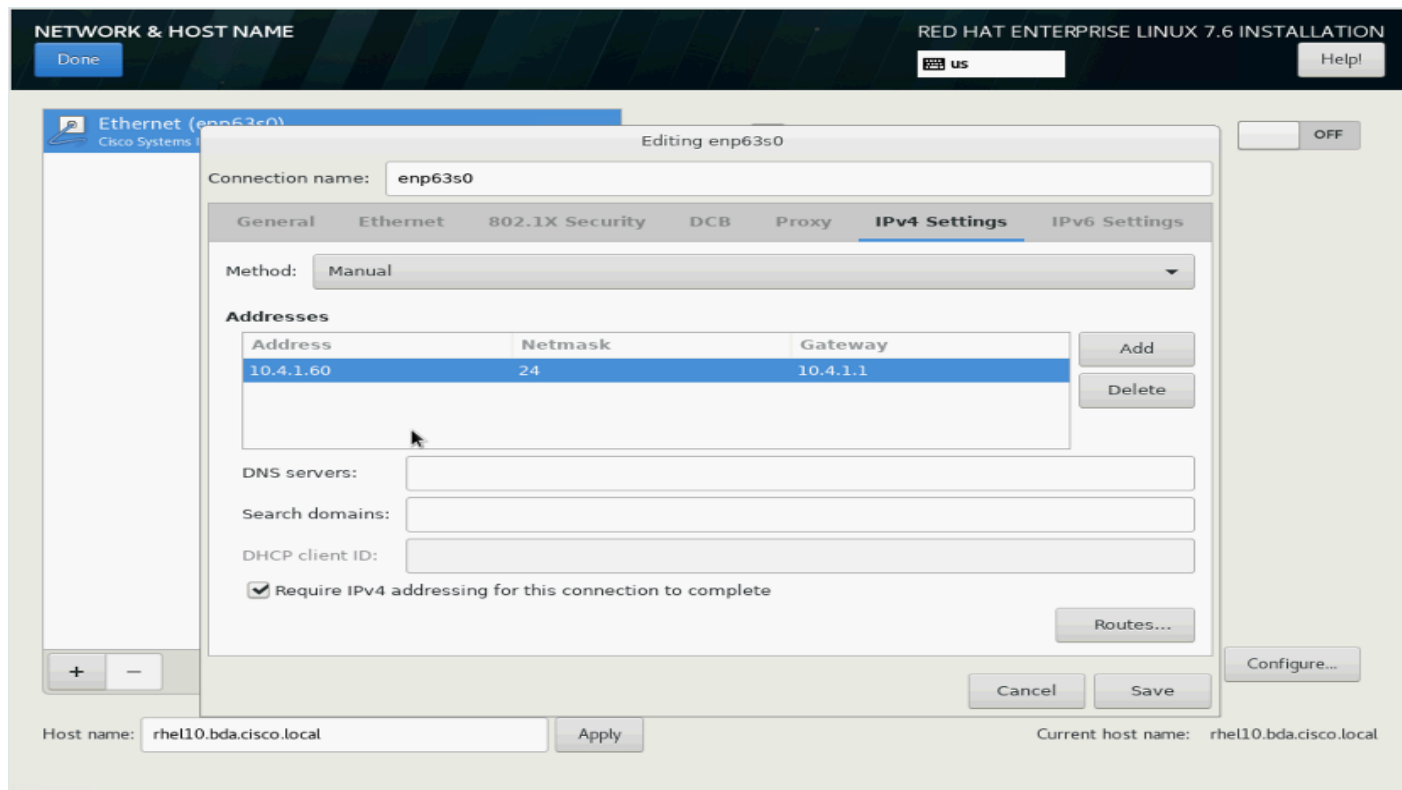
12. Click NETWORK & HOST NAME.



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



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



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

NETWORK & HOST NAME RED HAT ENTERPRISE LINUX 7.6 INSTALLATION

Done us Help

Ethernet (enp63s0)
Cisco Systems Inc VIC Ethernet NIC

+ -

Ethernet (enp63s0) ON
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

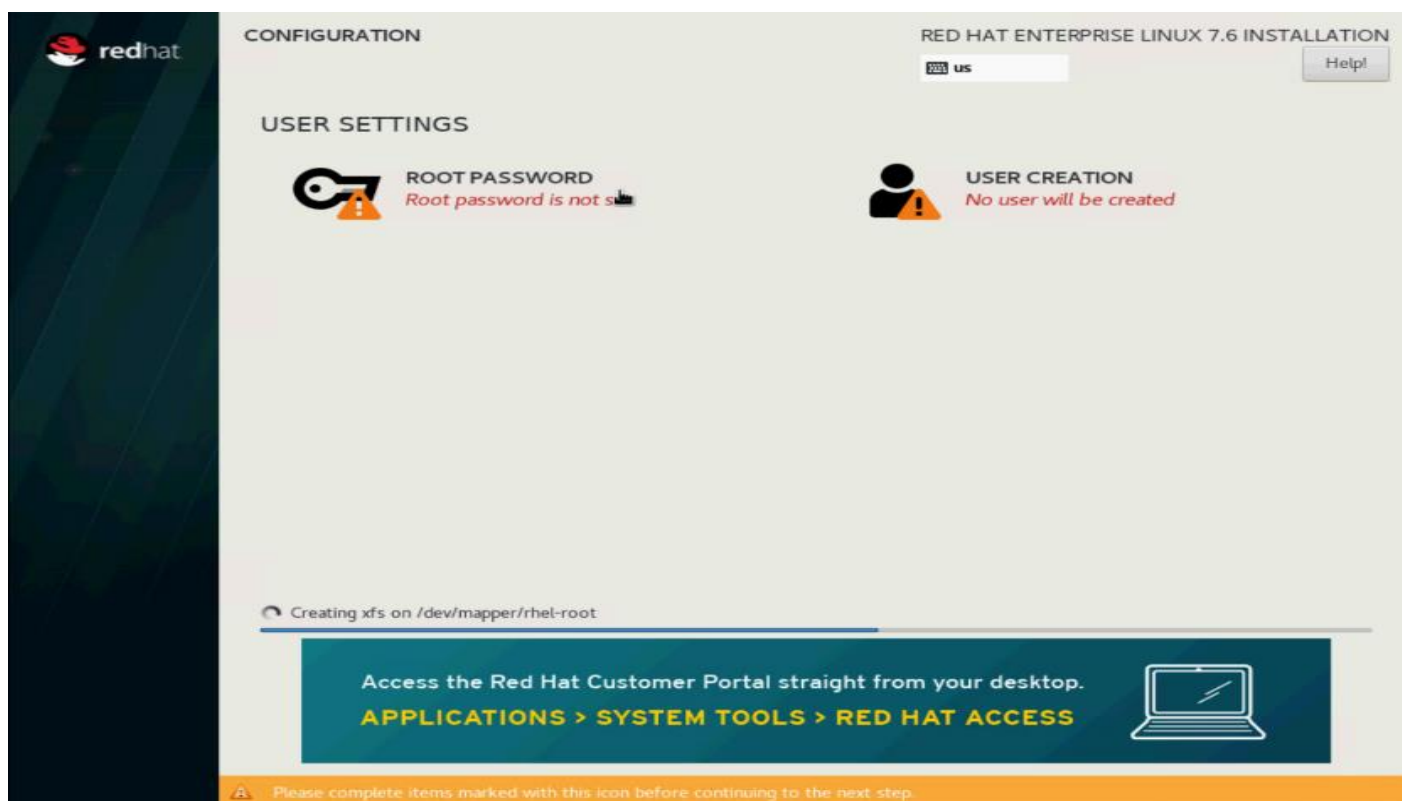
Configure...

Host name: Apply Current host name: rhel10.bda.cisco.local

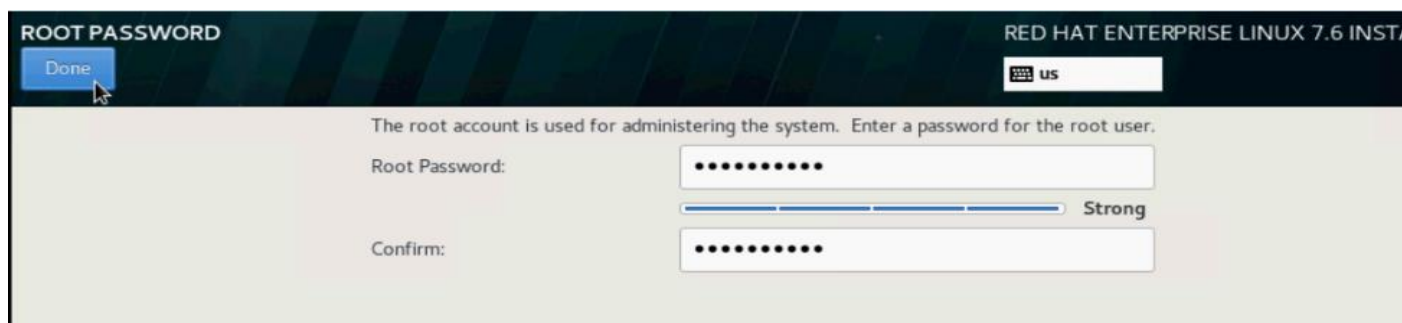
16. Click Begin Installation.



17. Click ROOT PASSWORD.



18. Enter Root Password. Click Done.



19. Reboot when the 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@rhe101 ~]# ansible all -m shell -a "./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'"
rhe106.hdp3.cisco.local | CHANGED | rc=0 >>
 24 Enclosure Device ID: 66
 24 Enclosure position: 0
rhe104.hdp3.cisco.local | CHANGED | rc=0 >>
 24 Enclosure Device ID: 66
 24 Enclosure position: 0
rhe108.hdp3.cisco.local | CHANGED | rc=0 >>
 24 Enclosure Device ID: 66
 24 Enclosure position: 0
```



It has been observed that some earlier versions of storcli64 complains about the 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` since 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 rhe1 to rhe3 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.

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 0 | | | | | | | |
| Storage Controller SAG 1 | | | | | | | |
| Disk 1 | 1719555 | S1Z0176a2000003444C2a | Operable | Online | Equipped | HDD | False |
| Disk 2 | 1719555 | S1Z029876000003444C2a | Operable | Online | Equipped | HDD | False |
| Disk 3 | 1719555 | S1Z0116P70000003444C2a | Operable | Online | Equipped | HDD | False |
| Disk 4 | 1719555 | S1Z012300000003444C2a | Operable | Online | Equipped | HDD | False |
| Disk 5 | 1719555 | S1Z0127400000003444C2a | Operable | Online | Equipped | HDD | False |
| Disk 6 | 1719555 | S1Z0116P70000003444C2a | Operable | Online | Equipped | HDD | False |
| Disk 7 | 1719555 | S1Z0116P70000003444C2a | 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@rhell1 ~]# ansible datanodes -m command -a "./storcli64 -cfgeachdiskraid0 WB RA direct NoCachedBadBBU strpsz1024 -a0"
```

```

rhel7.cdp.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.

Hadoop Tuning Guide for AMD EPYC Processors Based Servers

Introduction

The Apache Hadoop software library is a framework that allows for distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering either local computation and storage or both. When implemented in a cluster, the software has the built-in resiliency to handle everything from a failed drive to an entirely failed server. Hadoop uses these techniques instead of relying on hardware to delivery high availability. With such a wide diversity of applications and environments in which Hadoop runs there is not a golden rule for tuning the cluster. Different Hadoop distribution vendors expose different settings through their management software. This guide will help provide suggestions as to which parameters will have the most impact on the performance of Hadoop clusters. The categories of parameters discussed in this paper include Basic Input Output System (BIOS), Linux OS, Hadoop Distributed File System (HDFS), and Yet Another Resource Manager (YARN).

It is highly recommended to review the Hadoop Tuning Guide for AMD EPYC™ Processor Based Servers: <https://developer.amd.com/wp-content/resources/56419.PDF>



The concepts disclosed in this document while tested on Cloudera CDH 5.15 should apply to Hadoop distributions based upon Hadoop 2.x /3.x, including Cloudera CDP.

About the Author

Yogesh Ramesh, Big Data Solutions Architect, Cisco Systems, Inc.

Yogesh Ramesh 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.

Feedback

For comments and suggestions about this guide and related guides, join the discussion on [Cisco Community](https://cs.co/en-cvds) at <https://cs.co/en-cvds>.

Americas Headquarters

Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters

Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters

Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at <https://www.cisco.com/go/offices>.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)