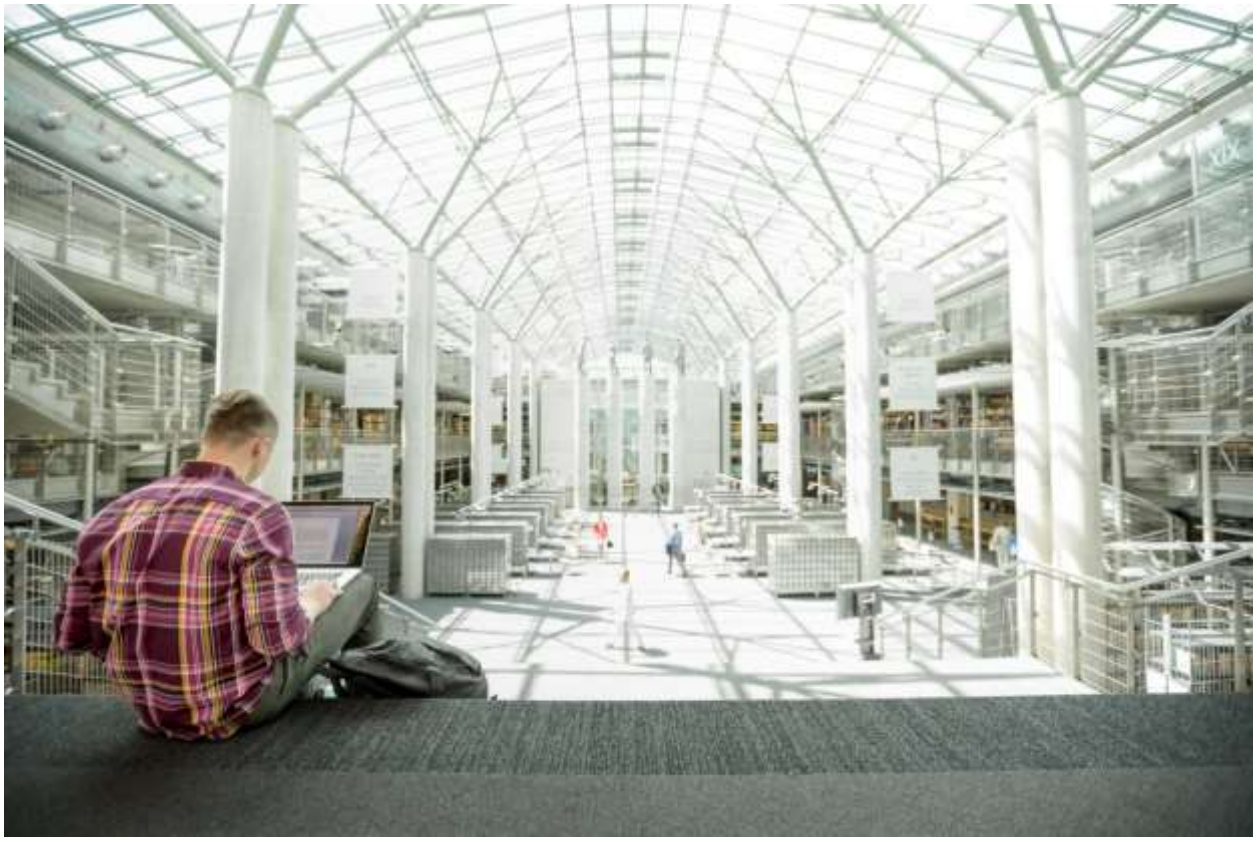


Cisco UCS S3260 Storage Server with Cloudera Enterprise

Last Updated: March 29, 2017

About Cisco Validated Designs

The CVD program consists of systems and solutions designed, tested, and documented to facilitate faster, more reliable, and more predictable customer deployments. For more information visit http://www.cisco.com/go/designzone.

Table of Contents

Executive Summary

Big data is now a significant element in many industries and gaining meaningful traction across multiple industries. Organizations heavily using big data technologies include healthcare, education, and energy along with financial, utility, advertising, retail, public sector, and manufacturing. Apache Hadoop, a technology developed to handle large volumes of data of any format, is much more efficient than traditional enterprise data warehouses. Because Hadoop is designed to run on industry standard hardware with infinite scalability, the cost savings are significant. Overall savings increase as organizations' data volumes grow. Cloudera and Cisco offer an industry-leading solution for enterprise Hadoop deployments.

The Cisco® UCS S3260 Storage Server is the latest addition to the highly successful Cisco Unified Computing System™ (Cisco UCS®) reference architecture for big data. This server provides up to 600 terabytes (TB) raw storage in only four rack units (4RU), providing the best dollar-per-terabyte value while delivering superior computing performance and a balanced core-to-spindle ratio. The Cisco® UCS S3260 Storage Server provides superior performance at a lower total cost and fewer servers mean less rack space, fewer OS and software licenses, and less networking equipment to purchase and maintain, and lower power and cooling costs. The UCS S3260 Storage Server is specifically designed to process huge volumes of data with high performance.

The modular design of the Cisco® UCS S3260 Storage Server protects your long-term technology investment. The computing, storage, and network components can be upgraded independently as technology advances. There is no need to replace the entire server; simply upgrade an individual component.

It complements Cisco UCS Integrated Infrastructure for Big Data and Analytics, a highly scalable architecture for big data systems that includes computing, storage, and networking resources fully managed through Cisco UCS Manager and linearly scalable to thousands of nodes using Cisco Nexus® 9000 Series Switches and the Cisco Application Centric Infrastructure (Cisco ACI™) platform.

The Cisco UCS S3260 Storage Server for Big Data and Analytics with Cloudera Enterprise is a tested, dependable deployment model for Hadoop-based big data systems. Together, they offer a fast and predictable path for businesses to unlock the value of big data

Solution Overview

Introduction

Big data is now a significant element in many industries and gaining meaningful traction across multiple industries. Organizations heavily using big data technologies include healthcare, education, and energy along with financial, utility, advertising, retail, public sector, and manufacturing. But how do you put all of this information to work for you?

Apache Hadoop, a technology developed to handle large volumes of data of any format, is much more efficient than traditional enterprise data warehouses. Because Hadoop is designed to run on industry standard hardware with infinite scalability, the cost savings are significant. Overall savings increase as organizations' data volumes grow.

Cloudera and Cisco offer an industry-leading solution for enterprise Hadoop deployments. A solution optimized to enable effective distributed parallel processing of huge amounts of data so companies can extract the most value out of their data is required. The Cisco UCS S3260 Storage Server is an application-influenced server and is designed specifically for data-intensive workloads.

Audience

This document describes the architecture and deployment procedures for Cloudera Enterprise 5.8 on an enterprise data hub using 8 Cisco UCS S3260 Storage Servers with two C3x60 M4 server nodes each as worker nodes, and three Cisco UCS C240 M4 Rack Servers as master nodes. The intended audience for this document includes, but is not limited to, sales engineers, field consultants, professional services, IT managers, partner engineering, and customers who want to deploy Cloudera Enterprise 5.8 on the Cisco Unified Computing System (UCS) using Cisco UCS S3260 Storage Servers.

Solution Summary

This CVD describes in detail the process of installing Cloudera Enterprise 5.8 and the configuration details of the cluster. It also details application configuration for Apache Spark and the libraries it provides.

The configuration using Cisco UCS S3260 Storage Servers as data nodes and Cisco UCS C240 M4 Rack Servers as management nodes, is shown in Table 1 This configuration supports the massive scalability that big data enterprise deployments demand.

Table 1 Reference Architecture - Configuration Details

Connectivity: Two Cisco UCS 6332 Fabric Interconnects	
Eight Cisco UCS S3260 Storage Server, each with two C3x60 M4 server nodes, each server node with:	Three Cisco UCS C240 M4 Rack Servers each with:

Two Intel Xeon processor E5-2680 v4 CPUs (14 cores on each CPU)	Two Intel Xeon processor E5-2680 v4 CPUs (14 cores on each CPU)
256 GB of memory	256 GB of memory
Cisco UCS-C3K-M4RAID SAS Modular RAID Controller with 4-GB FBWC	Cisco 12-Gbps SAS Modular RAID Controller with 2-GB FBWC
Twenty-four 4-TB 7,200-rpm LFF SAS drives (96 terabytes [TB])(1.56 petabytes[PB] total)	Twelve 1.2-TB 10,000-rpm SFF SAS drives
Cisco UCS VIC 1387 (with 2 x 40 Gigabit Ethernet QSFP ports)	Cisco UCS VIC 1387 (with 2 x 40 Gigabit Ethernet QSFP ports)
Two 480-GB 6-Gbps 2.5-inch enterprise value SATA SSD drives for boot	Two 240-GB 6-Gbps 2.5-inch enterprise value SATA SSD drives for boot

Data Storage for Big Data

Data is being generated at an unprecedented scale. More data is being collected more quickly and stored longer than ever before. Traditional transactional data is being supplemented with data from high-speed, real-time streaming systems and then stored for long periods of time both for archival and regulatory purposes. Sensors, internet-of-things (IoT) devices, social media, online transactions, and other sources are all generating data that needs to be efficiently captured, processed, and stored.

In order to satisfy the business and functional requirements, applications built on these platforms must reliably process data with no data loss, at large scale, while retrieving data efficiently to meet the SLA requirements.

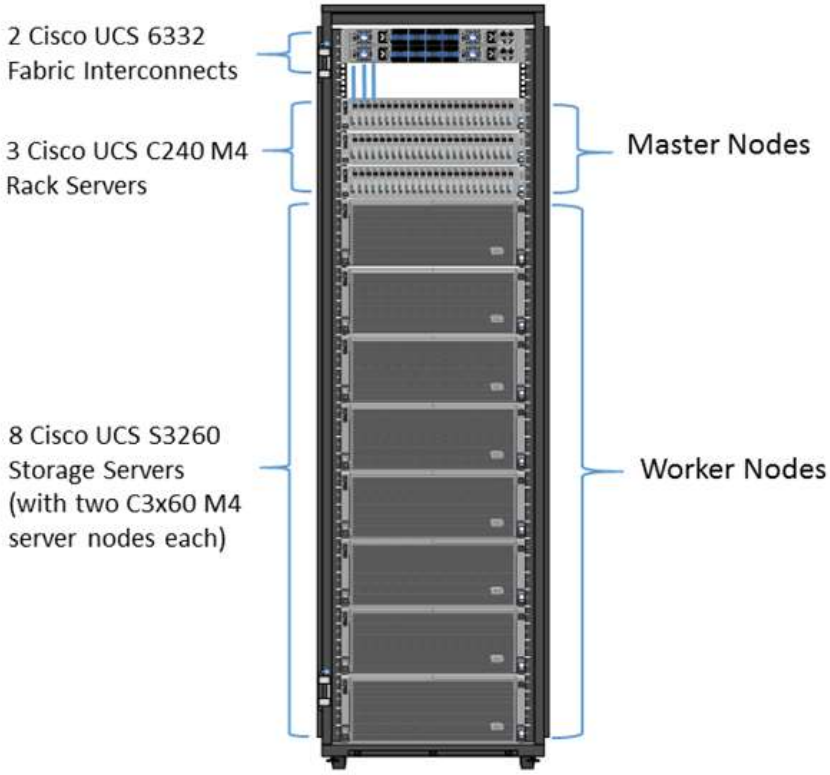
In order to do this, the system must be designed well and the software properly tuned. The next section details the relevant reference architecture for deploying Cloudera Enterprise on the Cisco Unified Computing System (UCS) using Cisco UCS S3260 Storage Servers.

Technology Overview

Reference Architecture

Figure 1 shows the base configuration of a high-availability Cloudera Enterprise cluster. It is comprised of 16 data nodes using 8 Cisco UCS S3260 Storage Servers (with two C3x60 M4 server nodes each) and 3 management nodes using Cisco UCS C240 M4 Rack Servers.

Figure 1 Reference Architecture



Note: This CVD describes the installation process of Cloudera Enterprise 5.8 (three master nodes for high-availability + 16 worker nodes).

Cisco UCS S3260 Storage Server for Big Data and Analytics with Cloudera Enterprise

This solution is based on the Cisco Unified Computing System (Cisco UCS) infrastructure using Cisco UCS 6300 Series Fabric Interconnects, and Cisco UCS S3260 Storage Servers. This architecture is specifically designed for high performance and linear scalability for big data workloads and is built using the following components:

Cisco UCS S3260 Storage Server

The Cisco UCS S3260 Storage Server (Figure 2) is a high-density modular storage server designed to deliver efficient, industry-leading storage for data-intensive workloads. The Cisco UCS S3260 Storage Server is a modular chassis with dual server nodes (two servers per chassis) and up to 60 large-form-factor (LFF) drives in a 4RU form factor. The server uses dual Intel® Xeon® Processor E5-2600 v4 Series CPUs and supports up to 512 GB of main memory and a range of hard-disk-drive (HDD) options. It comes with a pass-through controller or a RAID card with 4 GB cache and host bus adapter (HBA) controller, and up to two internal solid-state-disk (SSD) drives for boot, as shown in Figure 3 below.

Figure 2 Cisco UCS S3260 Storage Server

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

Figure 3 Cisco UCSC S3260 Storage Server Chassis –Back view, showing Two Servers

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

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

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

Cisco UCS C240 M4 Rack Servers

Cisco UCS C240 M4 High-Density Rack Servers (Small Form Factor Disk Drive Model), are enterprise-class systems that support a wide range of computing, I/O, and storage-capacity demands in compact designs. For this CVD three Cisco UCS C240 servers each with 12 drives are used for the master nodes.

Cisco UCS C-Series Rack-Mount Servers are based on the Intel Xeon® E5-2600 v4 series processor family that delivers the best combination of performance, flexibility, and efficiency gains, with 12-Gbps SAS throughput. The Cisco UCS C240 M4 Rack servers provide 24 DIMMs slots and can support up to 1.5 TB of main memory, (128 or 256 GB is typical for Big Data applications). It can support a range of disk drive and SSD options; twenty-four Small Form Factor (SFF) disk drives plus two (optional) internal SATA boot drives, for a total of 26 internal drives, are supported in the Performance Optimized option. Twelve Large Form Factor (LFF) disk drives, plus two (optional) internal SATA boot drives, for a total of 14 internal drives, are supported in the Capacity Optimized option, along with 2x1 Gigabit Ethernet embedded LAN-on-motherboard (LOM) ports. Cisco UCS Virtual Interface Cards 1387 (VICs), designed for the M4 generation of Cisco UCS C-Series Rack Servers, are optimized for high-bandwidth and low-

latency cluster connectivity, with support for up to 256 virtual devices, that are configured on demand through Cisco UCS Manager. Figure 4 show the Cisco UCS C240 M4 Rack Server. The back view is shown in Figure 5.

Figure 4 Cisco UCS C240 M4 Rack Server

Figure 5 Back View of Cisco UCS C240 M4 Rack Server

Cisco UCS VIC 1387

Cisco UCS Virtual Interface Cards (VICs) are unique to Cisco. The Cisco UCS VIC 1387 incorporates next-generation converged network adapter (CNA) technology from Cisco, and offers dual 40-Gbps ports designed for use with Cisco UCS Rack-Mount Servers. Optimized for virtualized networking, this card delivers high performance and bandwidth utilization, and supports up to 256 virtual devices.

The Cisco UCS VIC 1387 (Figure 6) offers dual-port, Enhanced Quad, Small Form-Factor Pluggable (QSFP) 40 Gigabit Ethernet and Fiber Channel over Ethernet (FCoE), in a modular-LAN-on-motherboard (mLOM) form factor. The mLOM slot can be used to install a Cisco VIC without consuming a PCIe slot providing greater I/O expandability.

Figure 6 Cisco UCS VIC 1387

Cisco UCS 6300 Series Fabric Interconnects

Cisco UCS 6300 Series Fabric Interconnects as shown in Figure 7, provide high-bandwidth, low-latency connectivity for servers, with Cisco UCS Manager providing integrated, unified management for all connected devices. The Cisco UCS 6300 Series Fabric Interconnects are a core part of Cisco UCS, providing low-latency, lossless 40 Gigabit Ethernet, Fibre Channel over Ethernet (FCoE), and Fibre Channel functions with management capabilities for systems deployed in redundant pairs.

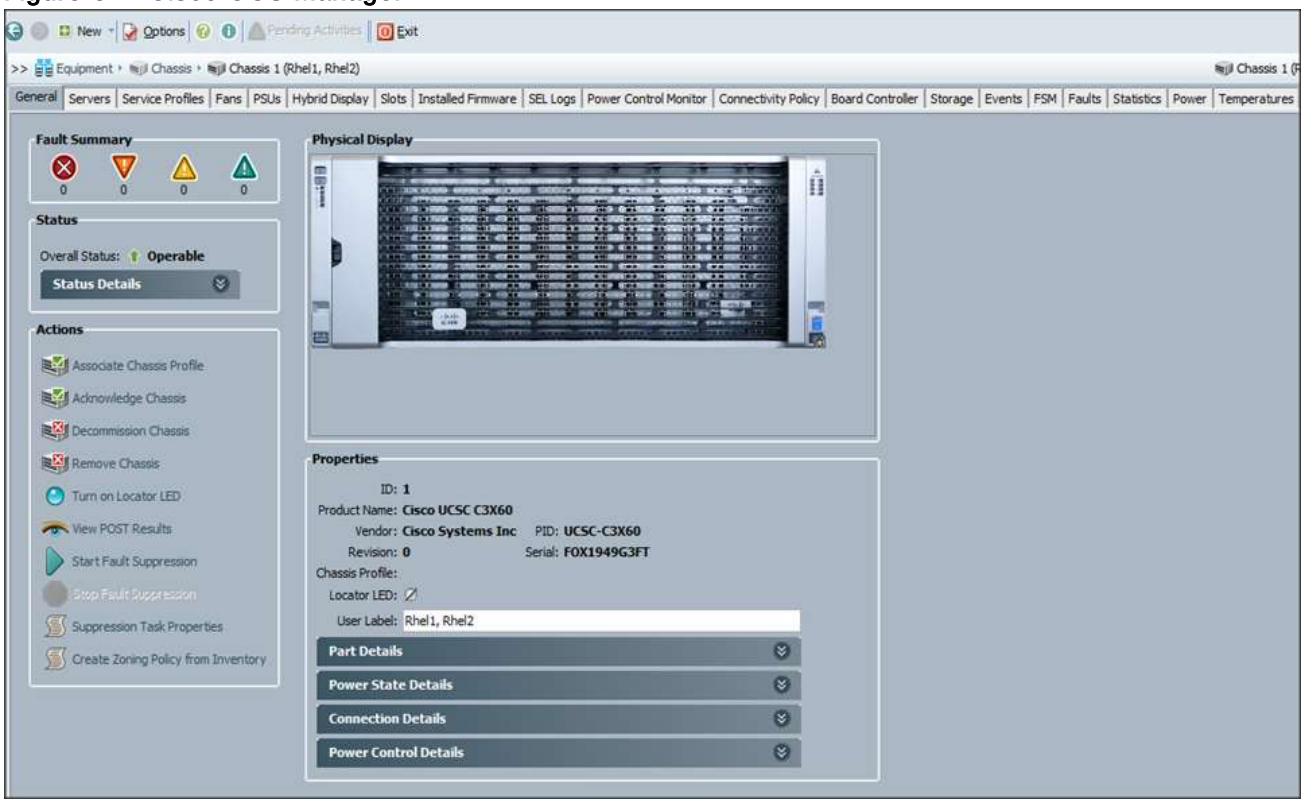
Cisco Fabric Interconnects offer the full active-active redundancy, performance, and exceptional scalability needed to support the large number of nodes that are typical in clusters serving big data applications. Cisco UCS Manager enables rapid and consistent server configuration using service profiles and automates ongoing system maintenance activities such as firmware updates across the entire cluster as a single operation. Cisco UCS Manager also offers advanced monitoring with options to raise alarms and send notifications about the health of the entire cluster.

Figure 7 Cisco UCS 6332 32-Port Fabric Interconnect

Cisco UCS Manager resides within the Cisco UCS 6300 Series Fabric Interconnect. It makes the system errors-aware and self-integrating, managing all of the system components as a single logical entity. Cisco UCS Manager can be accessed through an intuitive graphical user interface (GUI), as shown in Figure 8, a command-line interface (CLI), or an XML application-programming interface (API). Cisco UCS Manager uses service profiles to define the personality, configuration, and connectivity of all resources within Cisco UCS, radically simplifying provisioning of resources so the process takes minutes instead of days. This simplification allows IT departments to shift their focus from constant maintenance to strategic business initiatives.

The new Cisco UCS Manager has smart capabilities such as predictive drive failure and rebuild. With the integration with Cisco UCS S3260 Storage Server, Cisco UCS Manager can be configured to have hot spare drives in case of any drive failure. In such a case, Cisco UCS Manager will automatically detect the failed drives and replace it with one of the available hot spare drives, rebuild it and make it available to use within the Chassis.

Figure 8 Cisco UCS Manager



Cloudera Enterprise 5.8

Hadoop is a new type of data platform: one place to store unlimited data and access that data with multiple frameworks, all within the same platform. However, all too often, enterprises struggle to turn this new technology into real business value.

Powered by the world's most popular Hadoop distribution, only Cloudera Enterprise (Figure 9) makes Hadoop fast, easy, and secure so you can focus on results, not the technology.

Fast for Business - Only Cloudera Enterprise enables more insights for more users, all within a single platform. With the most powerful open source tools and the only active data optimization designed for Hadoop, you can move from big data to results faster. Key features include:

- In-Memory Data Processing: The most experience with Apache Spark
- Fast Analytic SQL: The lowest latency and best concurrency for BI with Apache Impala
- Native Search: Complete user accessibility built-into the platform with Apache Solr
- Active Data Optimization: Cloudera Navigator Optimizer (limited beta) helps tune data and workloads for peak performance with Hadoop

Easy to Manage - Hadoop is a complex, evolving ecosystem of open source projects. Only Cloudera Enterprise makes it simple so you can run at scale, across a variety of environments, all while meeting SLAs. Key features include:

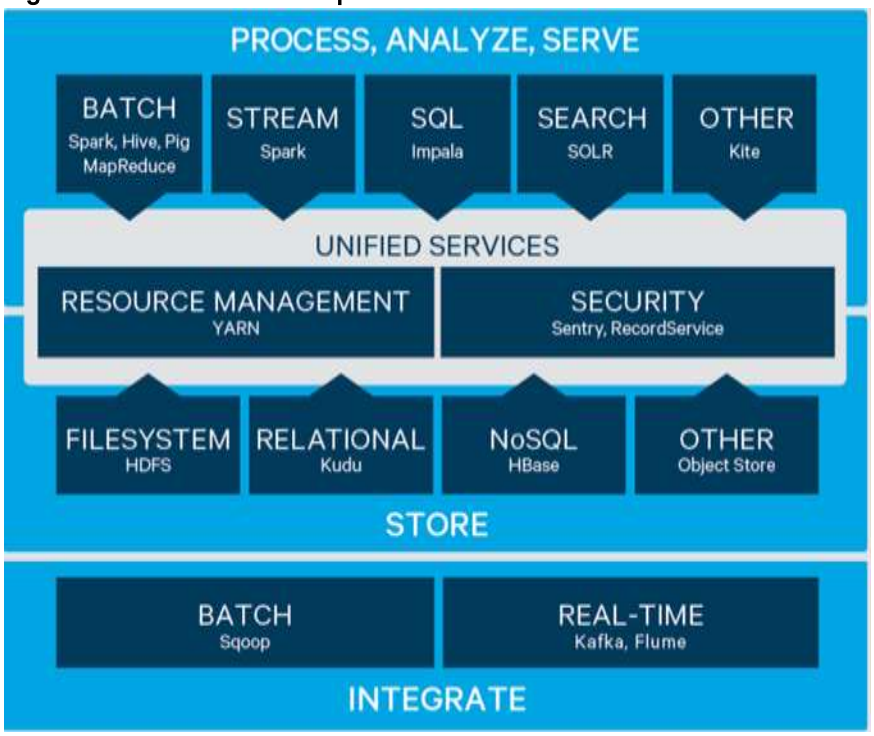
- Powerful Cluster Operations: Cloudera Manager is the Hadoop administration tool trusted by the professionals
- Hybrid Cloud Operations: Only Cloudera Director provides dynamic cluster management across all the major cloud environments
- Expert Support: Dedicated help and predictive care, just a click away
- Open Source Leadership: Constant open source development and curation, with the most rigorous testing, for trusted innovation

Secure without Compromise - The potential of big data is huge, but not at the expense of security. Cloudera Enterprise is the only Hadoop platform to achieve compliance with its comprehensive security and governance. Key features include:

- Enterprise Encryption and Key Management: Protect everything with Navigator Encrypt and Navigator Key Trustee
- Uniform Access Policy Enforcement: Uniformly manage and enforce role-based access controls across the entire platform with Apache Sentry and RecordService
- Automated Data Management: The only full-stack audit, lineage, discovery, and lifecycle management for Hadoop with Cloudera Navigator
- Secure Operations: The only separation of duties to protect production environments and built-in log and query redaction to protect sensitive information

Please refer to http://www.cloudera.com/products.html for more details.

Figure 9 Cloudera Enterprise



Apache Spark

Apache Spark is a fast and general-purpose engine for large-scale data processing. Running Spark on Cisco UCS Platform customers can accelerate streaming, interactive queries, machine learning, and batch workloads, and deliver more insights in less time.

Traditional servers are not designed to support the massive scalability, performance, and efficiency requirements of Big Data solutions. These outdated and siloed computing solutions are difficult to integrate with network and storage resources, and are time-consuming to deploy and expensive to operate. Cisco UCS Integrated Infrastructure for Big Data and Analytics takes a different approach, combining computing, networking, storage access, and management capabilities into a unified, fabric-based architecture that is optimized for big data workloads.

Spark's key advantage is speed, with an advanced DAG execution engine that supports cyclic data-flow and in-memory computing. Applications can be developed using built-in, high-level Apache Spark operations, or they can interact with applications like Python, R, and Scala REPLS, or Java. These various options allow users to quickly and easily build new applications and explore data faster.

Spark provides programmers with any application interface, centered on a data structure called the resilient distributed dataset (RDD), a read-only multiset of data items distributed over a cluster of machines that is maintained in a fault-tolerant way. Calculations are performed and results are delivered only when needed, and results can be configured to persist in memory, allowing Apache Spark to deliver a new level of computing efficiency and computation performance to Big Data deployments.

Spark has a number of libraries:

- Spark SQL/DataFrame API for querying structured data inside Spark programs.
- Spark Streaming offers Spark's core API that is able to perform real-time processing of streaming data, including web server log files, social media, and messaging queues.
- MLLib to take advantage of machine-learning algorithms and accelerate application performance across clusters.

Spark runs on Hadoop, stands alone, or in the cloud. It can access diverse data sources including HDFS, HBase. Spark with YARN is an optimal way to schedule and run Spark jobs on a Hadoop cluster alongside a variety of other data-processing frameworks, leveraging existing clusters using queue placement policies, and enabling security by running on Kerberos-enabled clusters.

Some common use cases popular in the field with Apache Spark:

- Simpler, faster, ETL − Data can be processed into the required format by avoiding intermediate writes to disk, and cleaned and aggregated in-memory before the final disk write.
- Real-time actions − Anomalous behaviors detected in real-time, and downstream actions are processed accordingly. For example; credit card transactions occurring in a different location generating actions for fraud alert, IOT sensors transmitting device failure data, etc.

- Data enrichment − Live data is enriched with more information by joining it with cached static datasets, allowing for a more comprehensive features set in real-time.
- Exploratory analytics − Events related to a specific time-window can be grouped together and analyzed. This sample data can be used by Data Scientists to update machine-learning models using tools like Python, etc. within Spark.
- Streaming data with analytics − The same code for streaming analytic operations can be used for batch, to compute over both the stream and historical data. This reduces moving parts and helps increase the productivity, consistency, and maintainability of analytic procedures. Spark is compatible with the rest of the streaming data ecosystem, supporting data sources including Flume, Kafka, ZeroMQ, and HDFS.
- Model building and machine learning − Spark is a big data tool that data scientists find easy to use which makes it ideal for building models for analytical purposes. Offline model building which needs data transfer from a Hadoop environment can be avoided now that Spark is used for model building and deployment.

Solution Design

Requirements

This CVD describes the architecture and deployment procedures to install Cloudera (Cloudera Enterprise 5.8.0) on eight Cisco UCS S3260 Storage Servers each with two C3x60 M4 server nodes each as Hadoop data nodes, and three Cisco UCS C240M4 Rack servers as Hadoop Management nodes for Big Data and Analytics. The solution goes into detail configuring Cloudera Enterprise 5.8.0 on the infrastructure.

The cluster configuration consists of the following:

- Two Cisco UCS 6332 Fabric Interconnects
- Three Cisco UCS C240 M4 Rack Servers
- Eight Cisco UCS S3260 Storage Servers with two C3x60 M4 server nodes each
- One Cisco R42610 standard rack
- Two Vertical Power distribution units (PDUs) (Country Specific)

Rack and PDU Configuration

Each rack contains two vertical PDUs, two Cisco UCS 6332 Fabric Interconnects, eight Cisco UCS S3260 Storage Servers with two C3x60 M4 server nodes each and three Cisco UCS C240 M4 Rack Servers. Each chassis is connected to two vertical PDUs for redundancy; ensuring availability during power source failure. The Rack Configuration is shown in Table 2

Note: Please contact your Cisco representative for country specific information.

Table 2 Rack Configuration

Position	Devices
42	Cisco UCS FI 6332
41	Cisco UCS FI 6332
40	Unused
39	Unused
38	Cisco UCS C240 M4 Rack Server
37	
36	Cisco UCS C240 M4 Rack Server
35	
34	Cisco UCS C240 M4 Rack Server
33	
32	Cisco UCS S3260 Storage Server
31	
30	
29	

28	Cisco UCS S3260 Storage Server
27	
26	
25	
24	Cisco UCS S3260 Storage Server
23	
22	
21	
20	Cisco UCS S3260 Storage Server
19	
18	
17	
16	Cisco UCS S3260 Storage Server
15	
14	
13	
12	Cisco UCS S3260 Storage Server
11	
10	
9	
8	Cisco UCS S3260 Storage Server
7	
6	
5	
4	Cisco UCS S3260 Storage Server
3	
2	
1	

Port Configuration on Fabric Interconnects

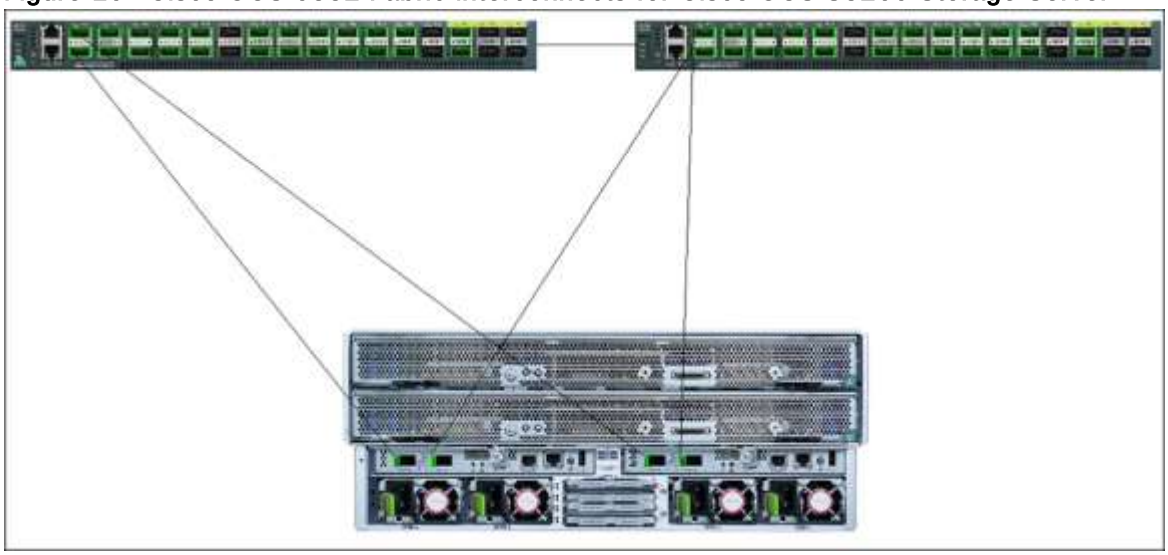
Port Type	Port Number
Network	32
Server	1-19

Server Configuration and Cabling for Cisco UCS S3260 Storage Server

The Cisco UCS S3260 Storage Server Chassis is equipped with two C3x60 M4 server nodes each and four 480 GB SATA SSDs. Each server node is equipped with two Intel Xeon® E5-2680 v4 processors, 256 GB of memory and a Cisco UCS-C3K-M4RAID SAS Modular RAID Controller with 4-GB FBWC.

Figure 10 illustrates the port connectivity between the Cisco UCS 6332 Fabric Interconnect, and Cisco UCS S3260 Storage Server Chassis. Eight Cisco UCS S3260 Storage Server Chassis are used in single rack configurations.

Figure 10 Cisco UCS 6332 Fabric Interconnects for Cisco UCS S3260 Storage Server



For more information on physical connectivity illustrations and cluster setup, see:

http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c-series_integration/ucsm3-1/b_C-Series-Integration_UCSM3-1/b_C-Series-Integration_UCSM3-1_chapter_010.html

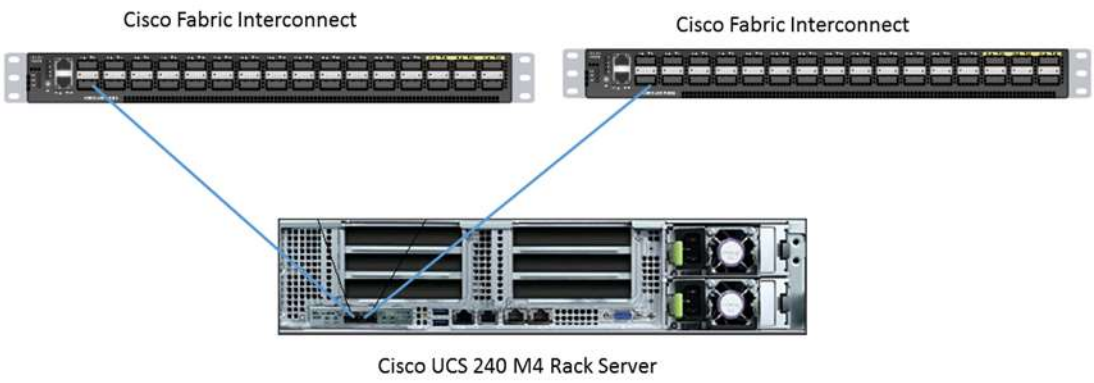
Figure 10 depicts the connectivity between Cisco UCS S3260 Storage Server chassis and Cisco UCS 6300 Fabric Interconnects. Each chassis has two C3x60 M4 server nodes. Each link in the figure represents a 40 Gigabit Ethernet link from the Cisco UCS S3260 Storage Server chassis connecting to a Fabric Interconnect. Every chassis is connected to both Fabric Interconnects represented with dual links.

Since each chassis will have two server nodes, the top server node works with the left SIOC and the bottom server node works with right SIOC (as show in Figure 10). Similarly, for the boot drives, the top two SSD slots are assigned for server node 1 and the bottom two SSD slots are assigned for server node 2.

Server Configuration and Cabling for Cisco UCS C240 M4 Rack Server

Each Cisco UCS C240M4 Rack Server is equipped with two Intel Xeon® E5-2680 v4 processors, 256 GB of memory and a Cisco 12-Gbps SAS Modular RAID Controller with 2-GB FBWC. The Fabric Topology for the Cisco UCS C240 M4 Rack Server is shown in Figure 11.

Figure 11 Fabric Topology for Cisco UCS C240 M4 Rack Server



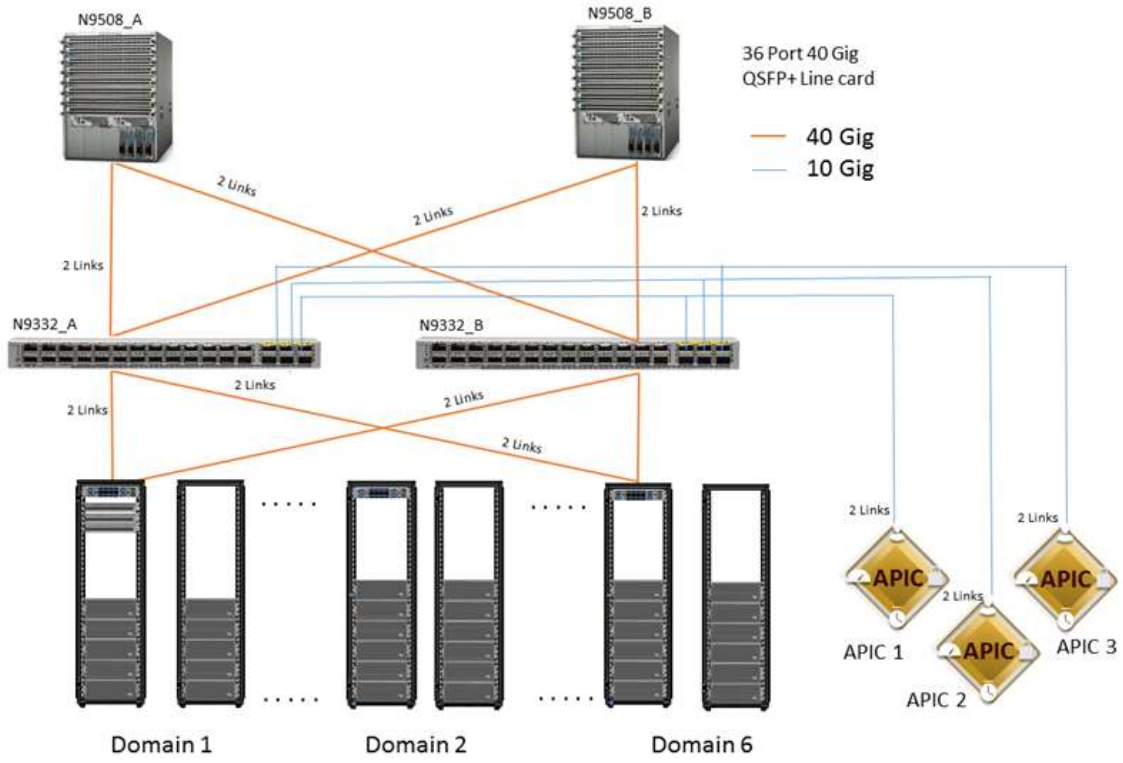
Cisco UCS S3260 Storage Server Scaling with Cisco Application Centric Infrastructure (ACI)

The system architecture includes the Cisco UCS S3260 Storage Server chassis. Each Fabric Interconnect domain can have 12 chassis under a single pair of Fabric Interconnects which are interconnected through the Cisco Application Centric Infrastructure (ACI) Fabric.

The ACI Fabric consists of three major components: the Application Policy Infrastructure Controller (APIC), spine switches, and leaf switches. These three components handle both the application of network policy and the delivery of packets.

The system architecture can be scaled up linearly and consists of 1 domain (1 pair of FIs) connecting to ACI having two Nexus 9508 switches acting as a Spine, two Nexus 9332PQ as the leaf switches, and three APIC-L1 as an APIC appliance. Figure 12

Figure 12 System Architecture



The following explains the system architecture for the base rack:

- The 8 Cisco UCS S3260 Storage Server chassis are rack mounted and connected to a pair of Fabric Interconnect representing a domain through 40GE link (4x40GE link to a pair of FI)
- Multiple such domains can be connected to a pair of ACI leaf switches. Here 40GE x 4 links from each FI are connected to leaf switches. This is done through a virtual port-channel of 2 links connected to each of the Nexus 9332.
- Nexus 9332 receives the 4x40GE from each pair of Fabric Interconnect as a vPC (Virtual Port-Channel), i.e., 2 ports coming from each single FI as an uplink to the leaf. There are 2 vPC for the 1 domain in each of 9332 connecting to a single pair of FIs
- Each leaf is connected to each Spine via 2 x 40 Gig connectivity cables.
- The three APIC's are connected to two leaves (Nexus 9332) via 10 gig SFP cable.

Six UCS domains can be connected to a pair of Leaf switches, this will accommodate up to 70 Cisco UCS S3260 Storage Servers.

- 1 pair of FI can connect up to 12 chassis
- 1 pair of Leaf switch can connect up to 6 pair of FI
- 1 Pair of Line card can connect up to 9 pair of leaf switches.

Further scaling can be done based on the requirement and is explained in Table 3 below.

Table 3 Spine to Leaf Connectivity

Spine	Line Card Pair	Ports Used	POD	Chassis	Leaf
N9508_A	Line Card 1 Line Card 1	1-2 3-4	1	70	9332_1A 9332_1B
	Line Card 1 Line Card 1	5-6 7-8	2	154	9332_2A 9332_2B
	Line Card 1 Line Card 1	9-10 11-12	3	238	9332_3A 9332_3B
	Line Card 1 Line Card 1	13-14 15-16	4	322	9332_4A 9332_4B
	Line Card 1 Line Card 1	17-18 19-20	5	406	9332_5A 9332_5B
	....	....	....	....	....
	Line Card 1 Line Card 1	33-34 35-36	9	742	9332_9A 9332_9B
	...	...	...	...	...
	Line Card 8 Line Card 8	1-2 3-4	64	5362	9332_64A 9332_64B
	Line Card 8 Line Card 8	1-2 3-4	65	5446	9332_65A 9332_65B
	Line Card 8 Line Card 8	1-2 3-4	66	5530	9332_66A 9332_66B
	Line Card 8 Line Card 8	1-2 3-4	67	5614	9332_67A 9332_67B
	Line Card 8 Line Card 8	5-6 7-8	68	5698	9332_68A 9332_68B
	....	....	....	....	....
	Line Card 8 Line Card 8	9-10 11-12	72	6034	9332_72A 9332_72B

Spine	Line Card Pair	Ports Used	POD	Chassis	Leaf
N9508_B	Line Card 1 Line Card 1	1-2 3-4	1	70	9332_1A 9332_1B
	Line Card 1 Line Card 1	5-6 7-8	2	154	9332_2A 9332_2B
	Line Card 1 Line Card 1	9-10 11-12	3	238	9332_3A 9332_3B
	Line Card 1 Line Card 1	13-14 15-16	4	322	9332_4A 9332_4B
	Line Card 1 Line Card 1	17-18 19-20	5	406	9332_5A 9332_5B
	....	....	....	....	....
	Line Card 1 Line Card 1	33-34 35-36	9	742	9332_9A 9332_9B
	...	...	...	...	...
	Line Card 8 Line Card 8	1-2 3-4	64	5362	9332_64A 9332_64B
	Line Card 8 Line Card 8	1-2 3-4	65	5446	9332_65A 9332_65B
	Line Card 8 Line Card 8	1-2 3-4	66	5530	9332_66A 9332_66B
	Line Card 8 Line Card 8	1-2 3-4	67	5614	9332_67A 9332_67B
	Line Card 8 Line Card 8	5-6 7-8	68	5698	9332_68A 9332_68B
	....	....	....	....	....
	Line Card 8 Line Card 8	9-10 11-12	72	6034	9332_72A 9332_72B

Table 4 Leaf to Fabric Interconnect Connectivity

LeafF	Ports Used	FI Domain	Chassis
9332_1A	1-4	Domain-1	1-10
9332_1A	5-8	Domain-2	11-22
9332_1A	9-12	Domain-3	23-34
9332_1A	13-16	Domain-4	35-46
9332_1A	17-20	Domain-5	47-58
9332_1A	21-24	Domain-6	59-70
9332_1A	25-27	APIC	
	28-31	Uplink to Spine	
	32	Unused	
9332_2A	1-4	Domain-1	1-10
9332_2A	5-8	Domain-2	11-22
9332_2A	9-12	Domain-3	23-34
9332_2A	13-16	Domain-4	35-46
9332_2A	17-20	Domain-5	47-58
9332_2A	21-24	Domain-6	59-70
9332_2A	25-27	APIC	
	28-31	Uplink to Spine	
	32	Unused	

Based on the system architecture above, only 6 UCS FI Domains can be connected to the first pair of leaf switches due to the port restrictions, as the leaf switch needs to connect three APIC Appliances, providing the scalability up to 70 chassis (10 chassis and 3 management nodes for the first domain and 12 chassis in each additional FI Domain). Each additional leaf pair can have up to 7 UCS FI Domain, providing the scalability up to 84 chassis (12 chassis in each FI Domain). The Cisco UCS S3260 Storage Server can be scaled up to 742 chassis with just a pair of line cards on the Nexus 9508 spine switch. Nexus 9508 can have up to 8 linecards, and with all 8 linecards being used for scaling can connect up to 6034 chassis providing a massive storage solution for the industry.

The architecture above has 4 unused ports in each FI, these ports can either be used as an uplink to Leaf switches or can be connected to external appliances. Most Hadoop distributions

require more than 3 management nodes in case the data nodes exceed more than 100. In that case these unused ports can be used to connect additional management nodes.

If the scaling is performed beyond the pair of leaf switches, it is recommended to connect APIC in three different leaf switches for maximum redundancy.

Note: This example shows a sample scaling capability using ACI a production implementation might vary based on the customer's network throughput requirements. Please reach out to a Cisco representative for your specific requirements.

Software Distributions and Versions

The required software distribution versions are listed below.

Cloudera Enterprise 5.8.0

Cloudera Enterprise version used is 5.8.0. For more information visit https://www.cloudera.com/documentation/enterprise/release-notes/topics/cdh_vd_cdh5_maven_repo_58x.html - concept_s1z_m5f_x5

Red Hat Enterprise Linux (RHEL)

The operating system supported is Red Hat Enterprise Linux 7.2. For more information visit http://www.redhat.com.

Software Versions

The software versions tested and validated in this document are shown in Table 5

Table 5 Software Versions

Layer	Component	Version or Release
Compute (Chassis) System IO Controller	Board Controller	1.0.14
	Chassis Management Controller	2.0(13aS4)
	Shared Adapter	4.1(2a)
	SAS Expander	04.08.01.B073
Compute (Server Nodes)	BIOS	C3x60M4.2.0.13c
	Board Controller	2.0
	CIMC Controller	2.0(13e)
Network	Cisco UCS 6332	3.1(2b)
	Kernel	5.0(3)N2(3.12b)
	Driver	2.3.0.30
Storage	Storage Controller SAS	29.00.1-0042
	Driver	06.810.10.00
Software	Red Hat Enterprise Linux Server	7.2 (x86_64)

Layer	Component	Version or Release
	Cisco UCS Manager	3.1(2b)
	Cloudera CDH	5.8.0

Note: The latest drivers can be downloaded from the link: https://software.cisco.com/download/release.html?mdfid=283862063&release=2.0(13)&relind=AVAILABLE&flowid=25886&softwareid=283853158&rellifecycle=&reltype=latest

Note: The latest supported RAID controller driver is already included with the RHEL 7.2 operating system.

Fabric Configuration

This section provides details for configuring a fully redundant, highly available Cisco UCS 6332 fabric configuration.

- Initial setup of the Fabric Interconnect A and B.
- Connect to UCS Manager using virtual IP address of using the web browser.
- Launch UCS Manager.
- Enable server, uplink and appliance ports.
- Start discovery process.
- Create pools and polices for service profile template.
- Create chassis and storage profile.
- Create Service Profile template and 16 Service profiles.
- Associate Service Profiles to servers.

Initial Setup of Cisco UCS 6332 Fabric Interconnects

This section describes the initial setup of the Cisco UCS 6332 Fabric Interconnects A and B.

Configure Fabric Interconnect A

1. Connect to the console port on the first Cisco UCS 6332 Fabric Interconnect.
2. At the prompt to enter the configuration method, enter console to continue.
3. If asked to either perform a new setup or restore from backup, enter setup to continue.
4. Enter y to continue to set up a new Fabric Interconnect.
5. Enter y to enforce strong passwords.

6. Enter the password for the admin user.
7. Enter the same password again to confirm the password for the admin user.
8. When asked if this fabric interconnect is part of a cluster, answer y to continue.
9. Enter n for the switch fabric.
10. Enter the cluster name for the system name.
11. Enter the Mgmt0 IPv4 address.
12. Enter the Mgmt0 IPv4 netmask.
13. Enter the IPv4 address of the default gateway.
14. Enter the cluster IPv4 address.
15. To configure DNS, answer y
16. Enter the DNS IPv4 address.
17. Answer y to set up the default domain name.
18. Enter the default domain name.
19. Review the settings that were printed to the console, and if they are correct, answer yes to save the configuration.
20. Wait for the login prompt to make sure the configuration has been saved.

Configure Fabric Interconnect B

1. Connect to the console port on the second Cisco UCS 6332 Fabric Interconnect.
2. When prompted to enter the configuration method, enter console to continue.
3. The installer detects the presence of the partner Fabric Interconnect and adds this fabric interconnect to the cluster. Enter y to continue the installation.
4. Enter the admin password that was configured for the first Fabric Interconnect.
5. Enter the Mgmt0 IPv4 address.
6. Answer yes to save the configuration.
7. Wait for the login prompt to confirm that the configuration has been saved.

For more information on Cisco UCS 6300 Series Fabric Interconnect, see:

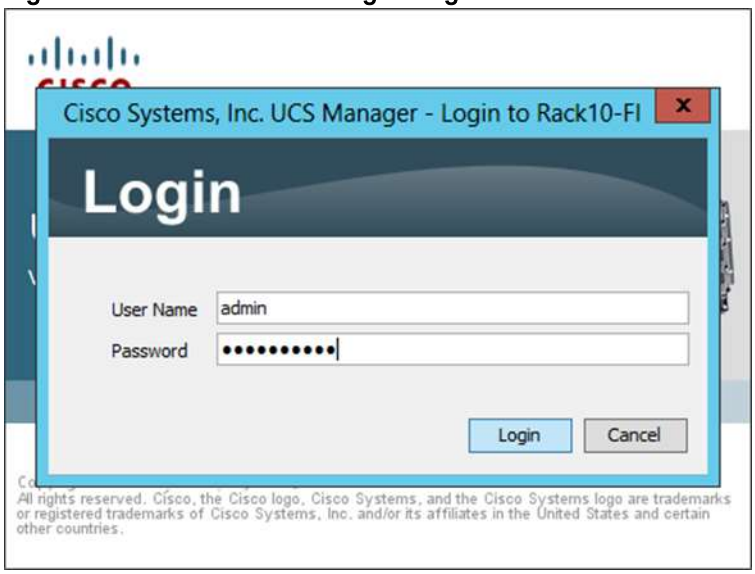
http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-6300-series-fabric-interconnects/index.html

Logging Into Cisco UCS Manager

To login to Cisco UCS Manager, complete the following steps:

1. Open a Web browser and navigate to the Cisco UCS 6332 Fabric Interconnect cluster address.
2. Click the Launch link to download the Cisco UCS Manager software.
3. If prompted to accept security certificates, accept as necessary.
4. When prompted, enter admin for the username and enter the administrative password.
5. Click Login to log in to the Cisco UCS Manager, Figure 13

Figure 13 Cisco UCS Manager Login Screen

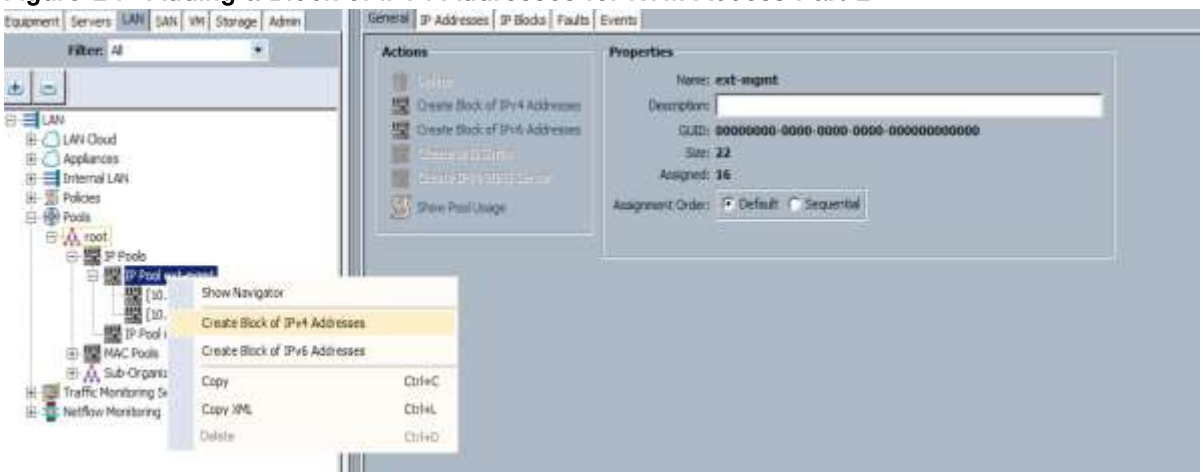


Adding a Block of IP Addresses for KVM Access

To create a block of KVM IP addresses for server access in the Cisco UCS environment, complete the following steps:

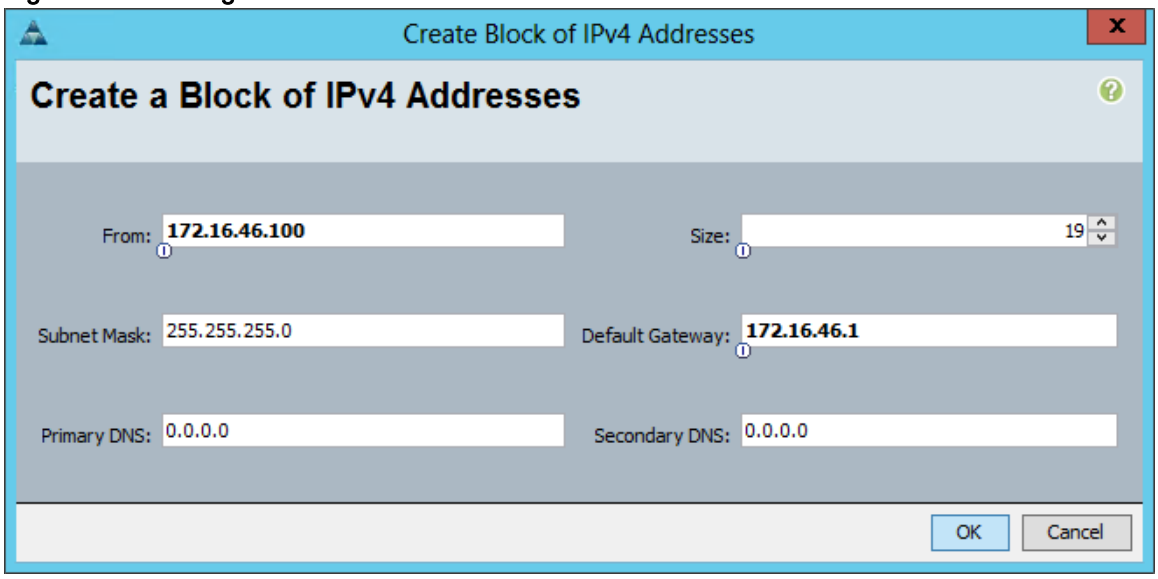
1. Select the LAN tab at the top of the left window, Figure 14
2. Select Pools > IpPools > Ip Pool ext-mgmt.
3. Right-click IP Pool ext-mgmt.
4. Select Create Block of IPv4 Addresses.

Figure 14 Adding a Block of IPv4 Addresses for KVM Access Part 1



5. Enter the starting IP address of the block and number of IPs needed, as well as the subnet and gateway information.

Figure 15 Adding Block of IPv4 Addresses for KVM Access Part 2



6. Click OK to create the IP block.

7. Click OK in the message box.

Enabling Uplink Ports

To enable uplinks ports, complete the following steps:

1. Select the Equipment tab on the top left of the window.

2. Select Equipment > Fabric Interconnects > Fabric Interconnect A (primary) > Fixed Module.

3. Click Ethernet Ports section.

4. Select port 32 that are connected to the uplink switch, right-click, and then select Configure as Uplink Port. (Figure 16)

5. Select Equipment > Fabric Interconnects > Fabric Interconnect B (subordinate) > Fixed Module.

6. Click Ethernet Ports section.

7. Select port 32 that is connected to the uplink switch, right-click, and then select Configure as Uplink Port.

Figure 16 Enabling Uplink Ports

Configuring VLAN

The VLANs are configured as in shown in Table 6

Table 6 VLAN Configurations

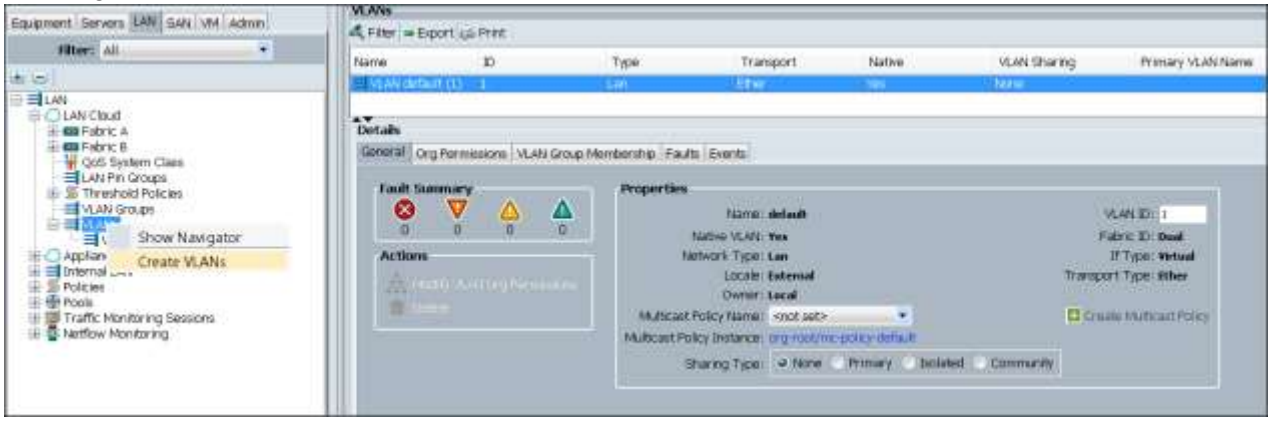
VLAN	NIC Port	Function
VLAN76	Eth0	Management & Data Traffic

The NIC will carry both the management and the data traffic from VLAN76. 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.

To configure VLANs in the Cisco UCS Manager GUI, complete the following steps:

1. Select the LAN tab in the left pane in the UCSM GUI.
2. Select LAN > LAN Cloud > VLANs.
3. Right-click the VLANs under the root organization.
4. Select Create VLANs to create the VLAN. (0)

Creating a VLAN



Enter vlan76 for the VLAN Name. (0)

Keep multicast policy as <not set>.

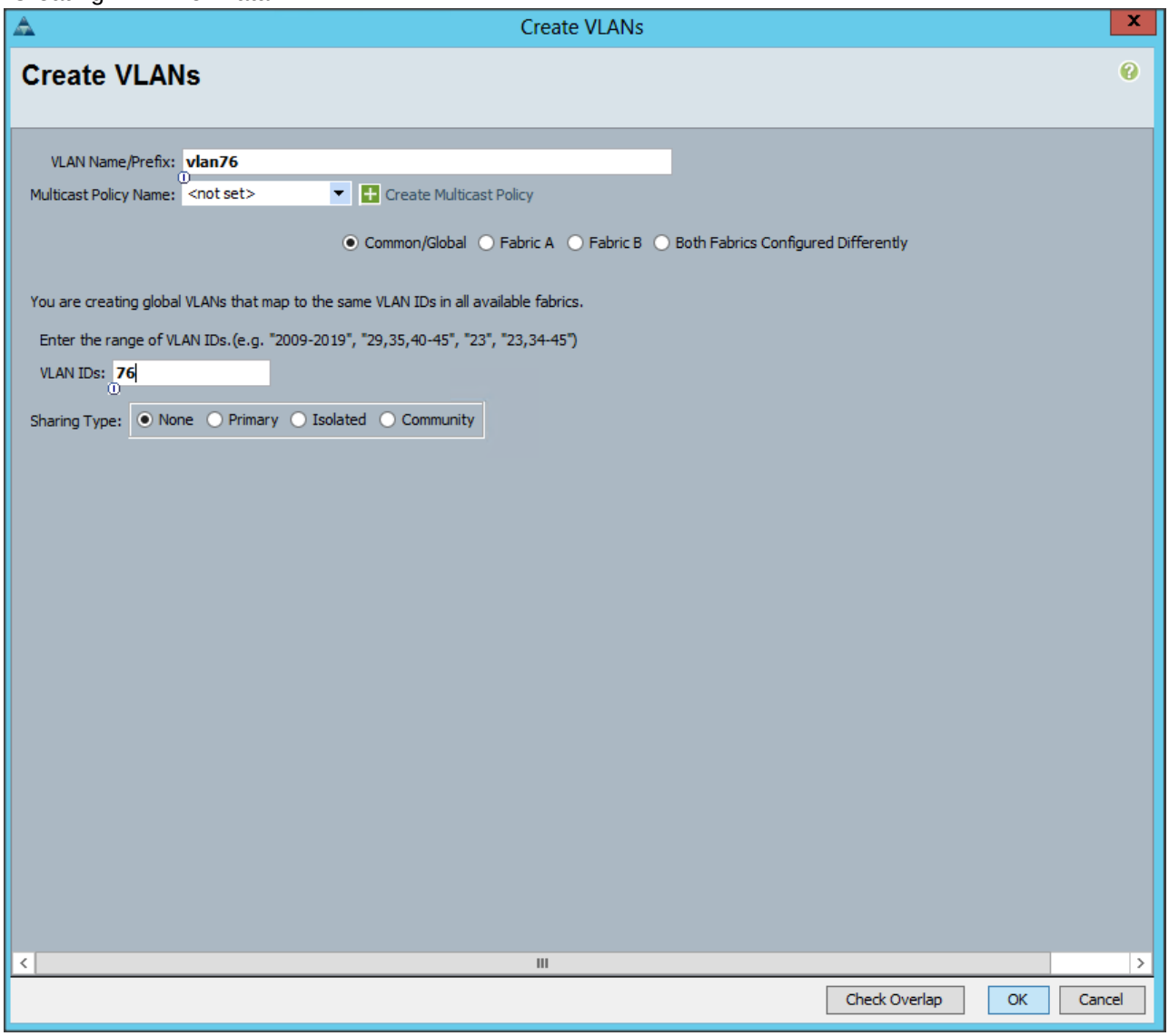
Select Common/Global for vlan76.

Enter 76 in the VLAN IDs field for the Create VLAN IDs.

Click OK and then, click Finish

Click OK in the success message box.

Creating VLAN for Data



Click OK and then, click Finish

Enabling Server Ports

To enable server ports, complete the following steps:

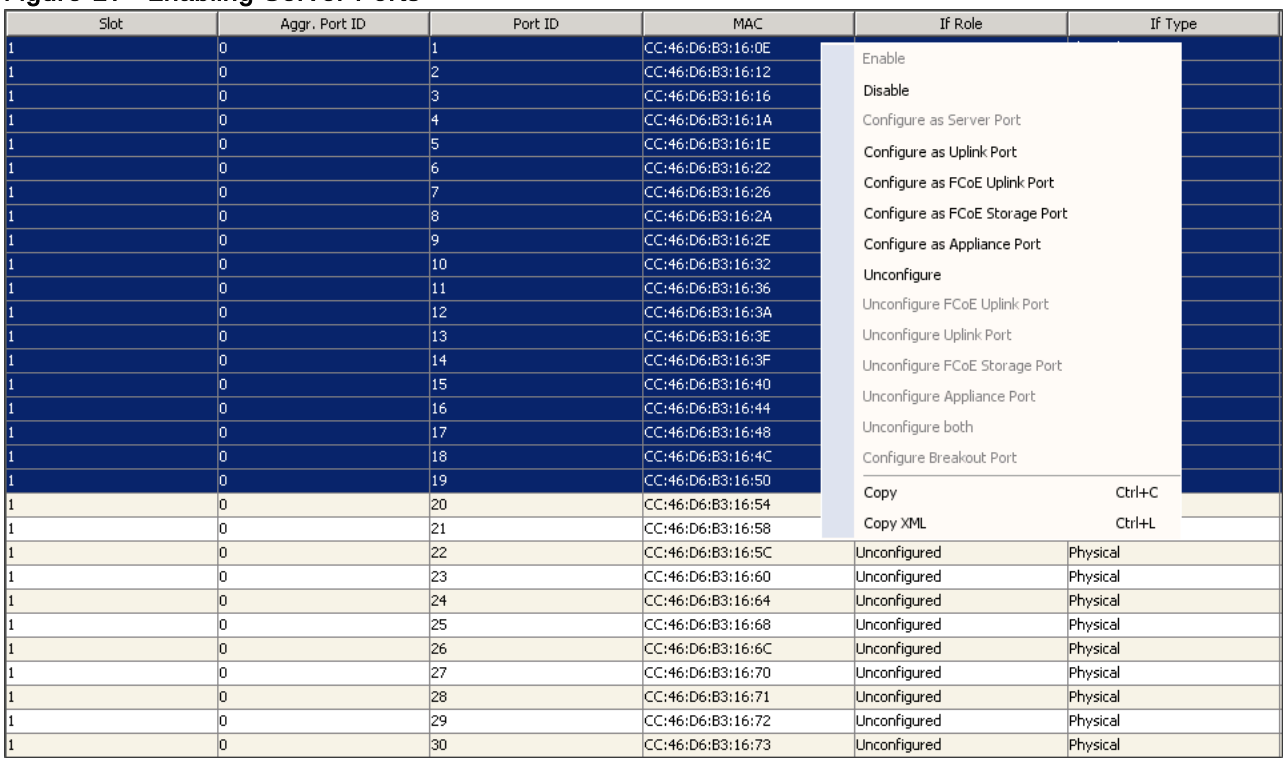
1. Select the Equipment tab on the top left of the window.
2. Select Equipment > Fabric Interconnects > Fabric Interconnect A (primary) > Fixed Module
3. Click the Ethernet Ports section.
4. Select all the ports that are connected to the Servers right-click them, and select Configure as a Server Port. (In this case it is ports 1-19). (Figure 17)

5. Select Equipment > Fabric Interconnects > Fabric Interconnect B (subordinate) > Fixed Module.

6. Click Ethernet Ports section.

7. Select all the ports that are connected to the Servers right-click them, and select `Configure as a Server Port` (In this case it is ports 1-19).

Figure 17 Enabling Server Ports



Creating Chassis Profile

Chassis profile is required to assign the number of drives to the particular server nodes and also to upgrade the chassis firmware.

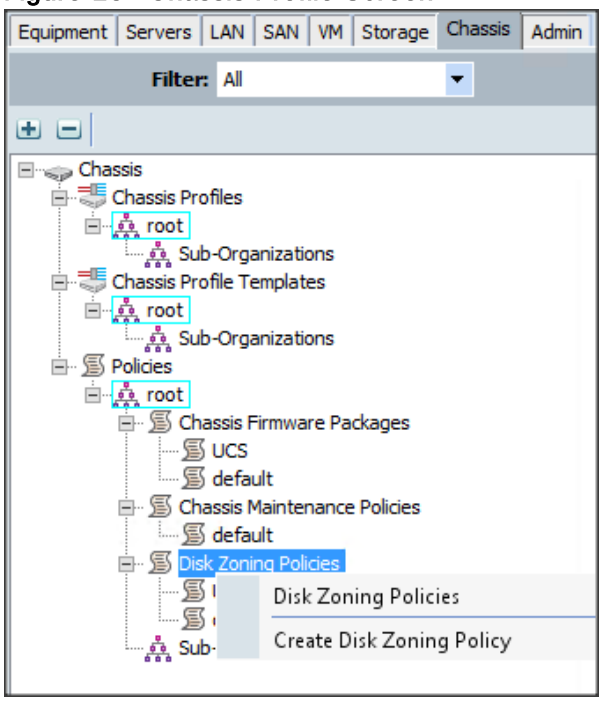
Creating Disk Zoning Policy

1. Click the Chassis tab on UCS Manager on the top left menu. Figure 18

2. Expand Policies→Root→Disk Zoning Policies

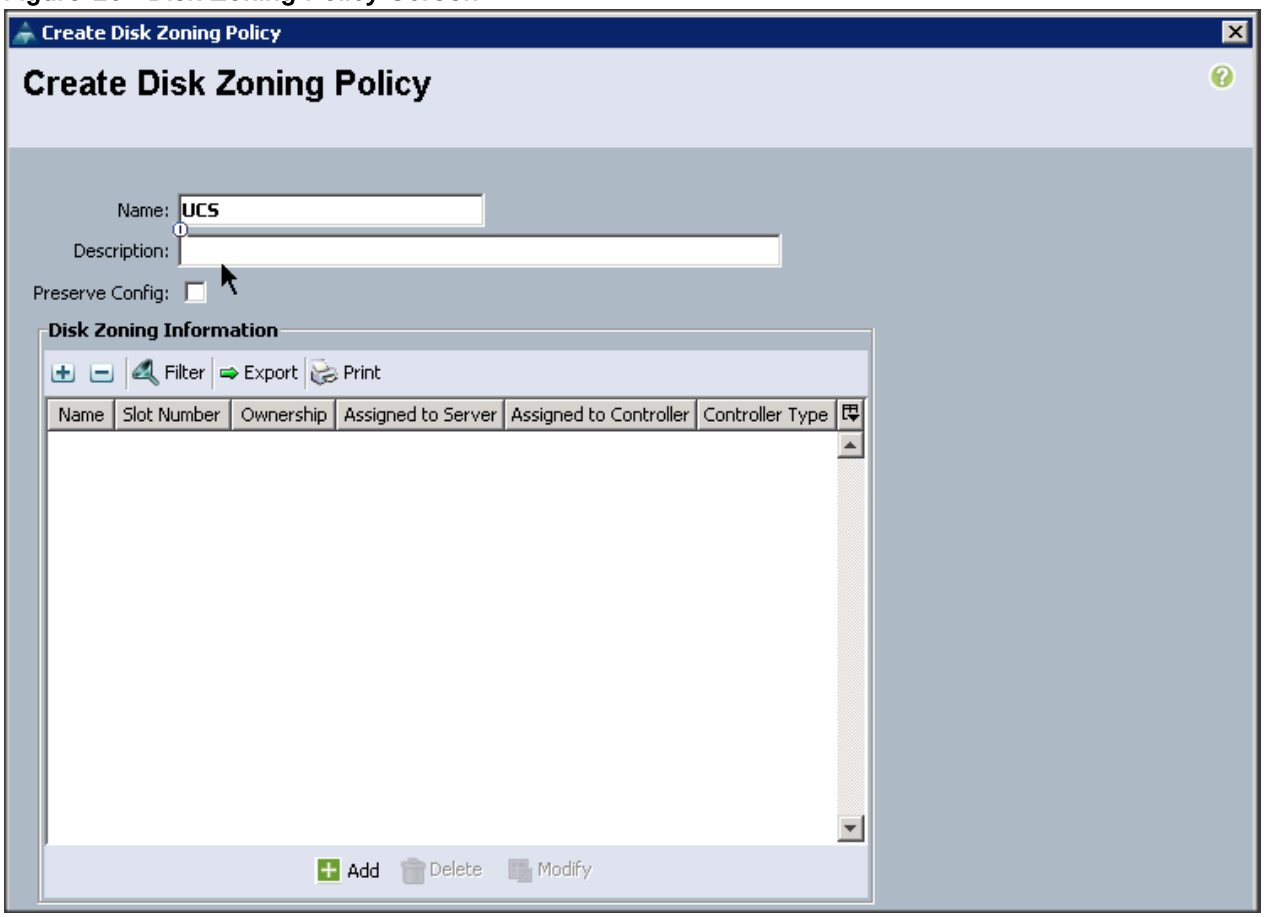
3. Right click on the Disk Zoning Policies and click Create Disk Zoning Policy.

Figure 18 Chassis Profile Screen



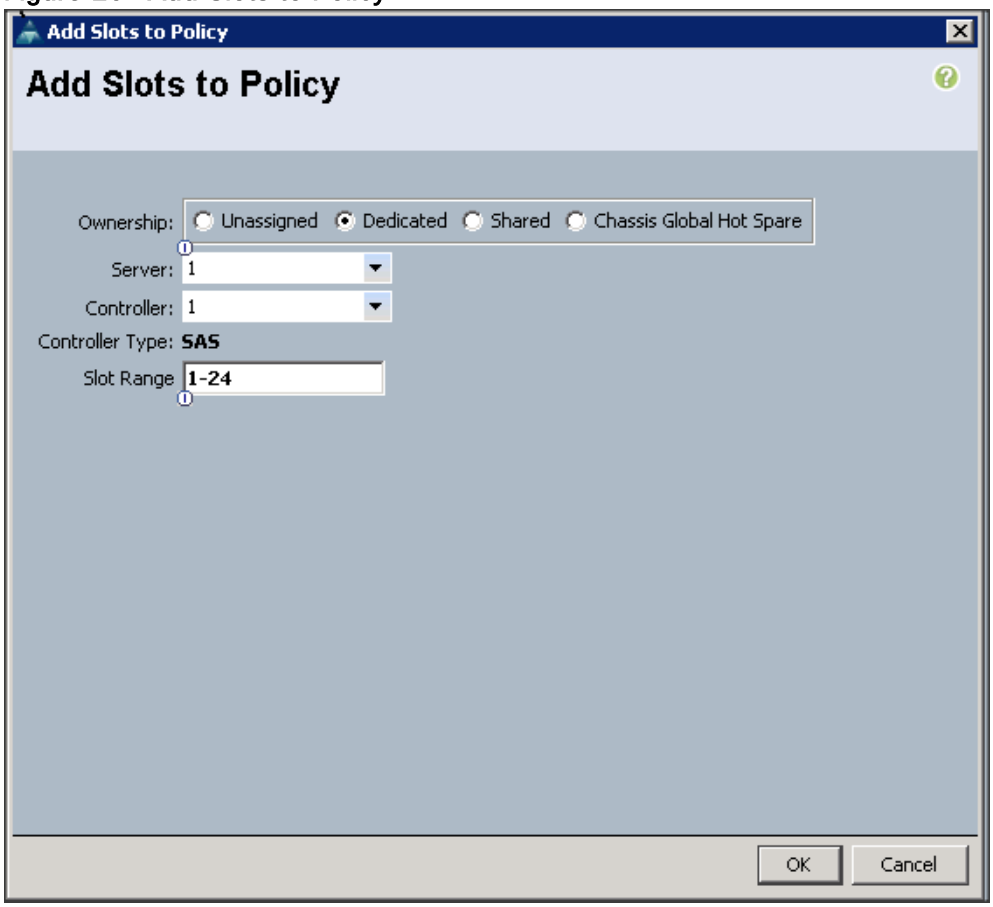
4. On Create Disk Zoning Policy windows enter the Name UCS and click "+" to create the Disk Zoning. Figure 19

Figure 19 Disk Zoning Policy Screen



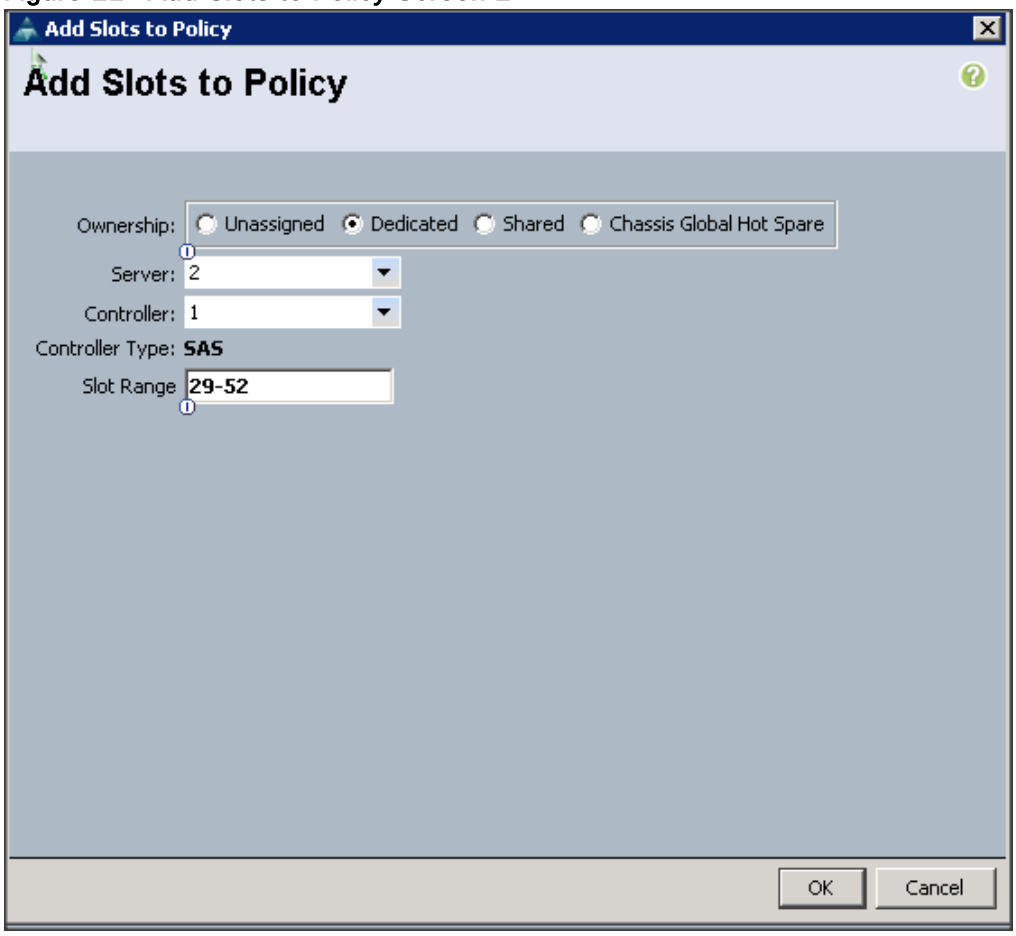
5. In the Add Slots to Policy window (Figure 20), select the "Dedicated" radio button. From the server drop down list choose "1", from the controller drop down list choose "1", in the slot range enter 1-24 and click "OK".

Figure 20 Add Slots to Policy



6. Click "+" again and In Add Slots to Policy window, select the "Dedicated" radio button. From the server drop down list choose "2", from the controller drop down list choose "1", in the slot range enter 29-52 and click "OK". (Figure 21)

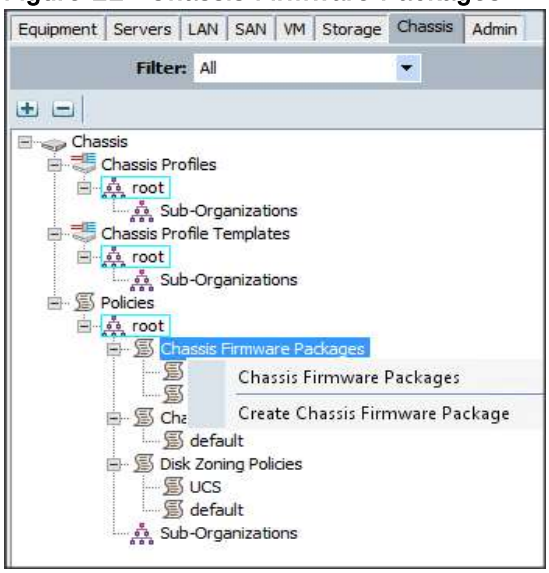
Figure 21 Add Slots to Policy Screen 2



Creating Chassis Firmware Package Policy

1. Right click on the Chassis Firmware Packages and click "Create Chassis Firmware Packages". (Figure 22)

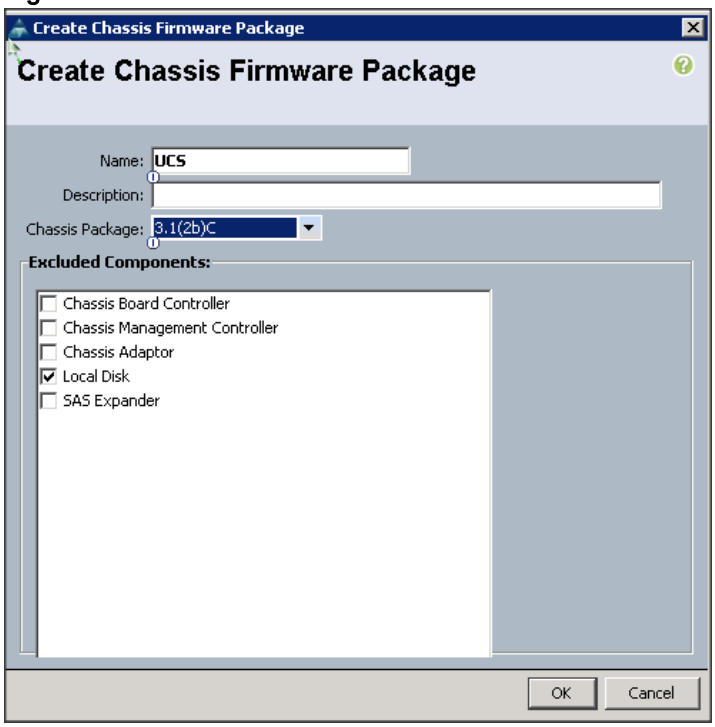
Figure 22 Chassis Firmware Packages



2. In Create Chassis Firmware Package window enter UCS as the Name. (Figure 23)

3. From the Chassis Packages drop down list choose the appropriate package and click OK.

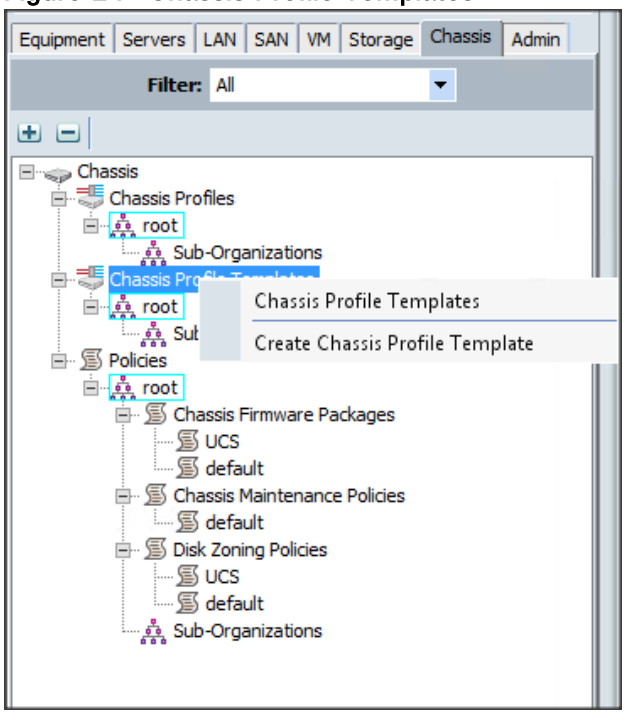
Figure 23 Create Chassis Firmware Screen



Creating Chassis Profiles from Template

1. Under Chassis Profile Template, right click and click "Create Chassis Profile Templates" (Figure 24).

Figure 24 Chassis Profile Templates

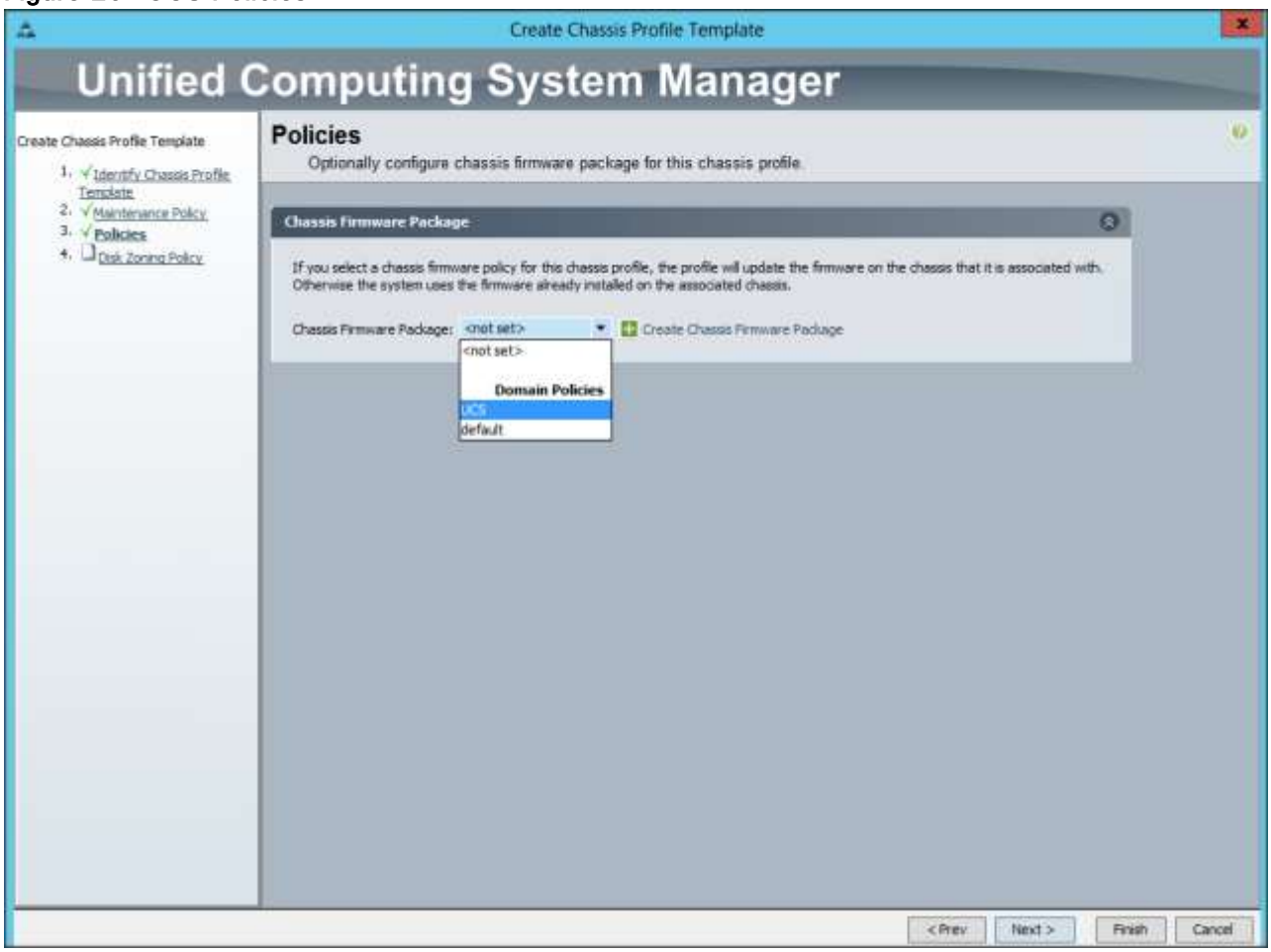


2. Enter the Name "UCS" and select "Updating Template" as the type, and click Next and Next again. (Figure 25)

Figure 25 Identify Chassis Profile Template

3. From the Chassis Firmware Package drop down list choose UCS and click Next. (Figure 26)

Figure 26 UCS Policies

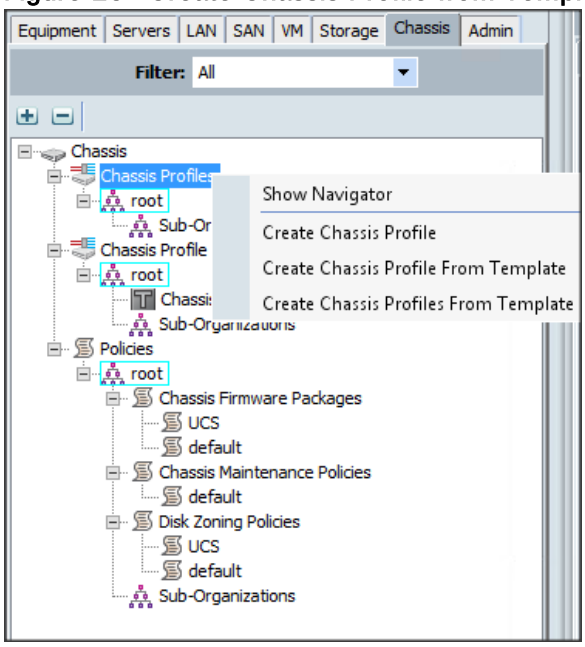


4. From the Disk Zoning Policy drop down list choose UCS and click Finish. (Figure 27)

Figure 27 Disk Zoning Policy

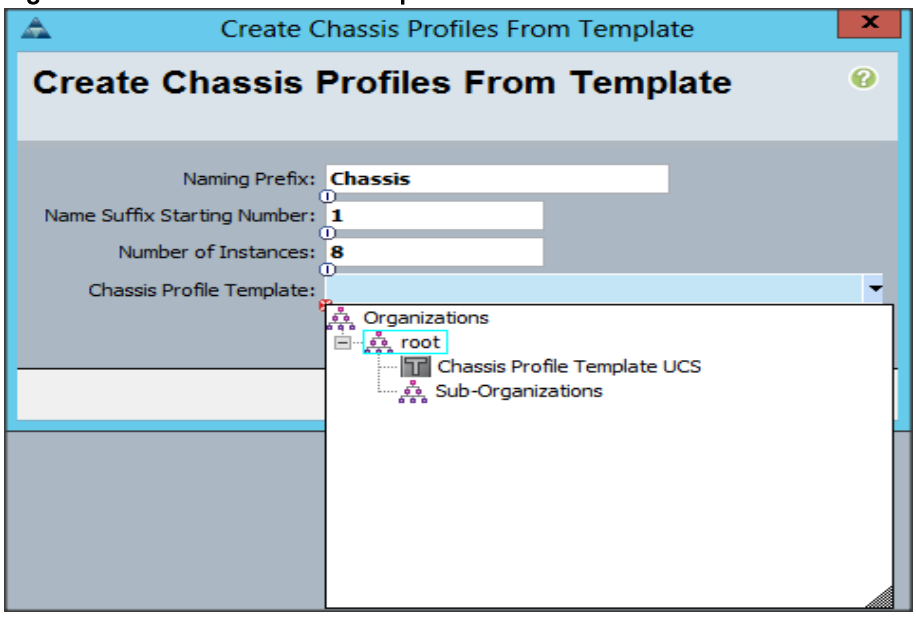
5. Right click on the Chassis Profiles and click “Create Chassis Profile from Templates” (Figure 28).

Figure 28 Create Chassis Profile from Templates



6. Enter Chassis as the Naming Prefix, the Number of Instances is "8" and from the Chassis Profile Template drop down list choose "Chassis Profile Template UCS" and click OK. (Figure 29)

Figure 29 Chassis Profile Template UCS

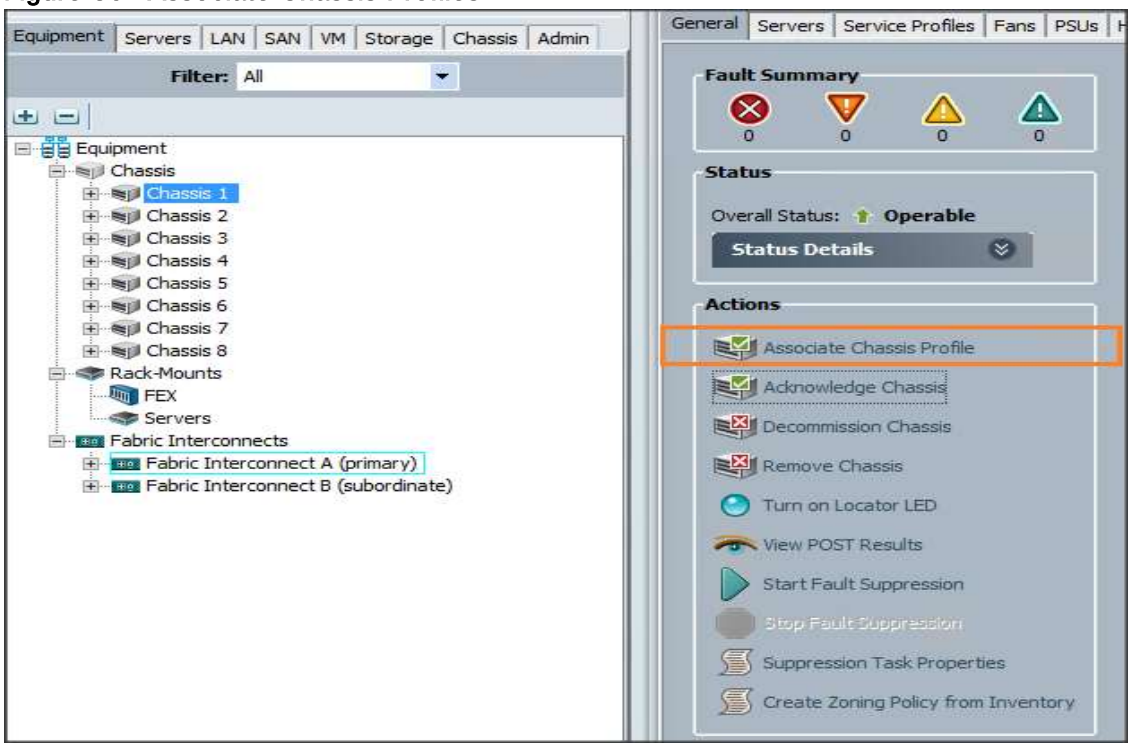


Associating Chassis Profiles to Individual Chassis

1. On the Cisco UCS Manager UI select the Equipment tab. Under Equipment expand Chassis.

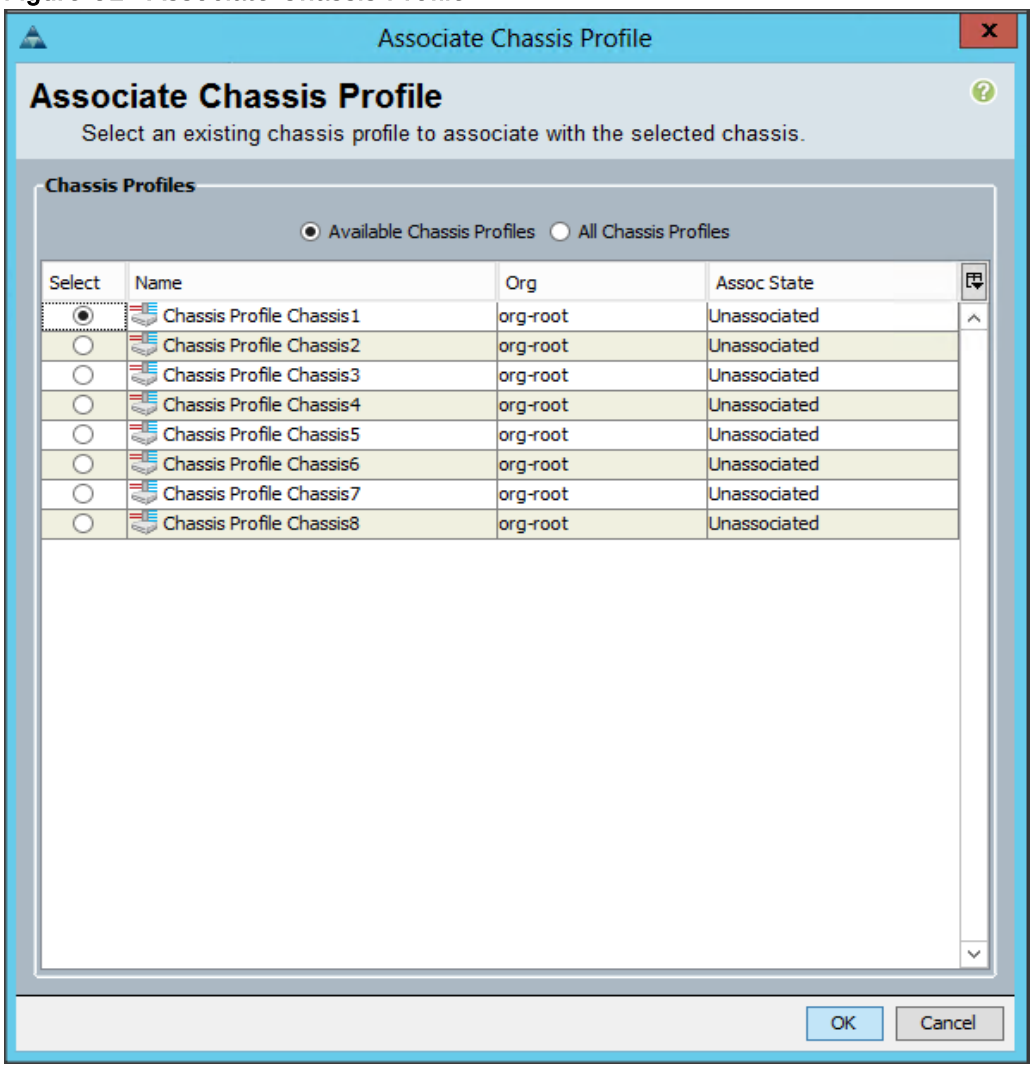
2. Select the Chassis and click Associate Chassis Profile. (Figure 30)

Figure 30 Associate Chassis Profiles



3. Select “Chassis Profile Chassis 1” and click “OK”. (Figure 31)

Figure 31 Associate Chassis Profile

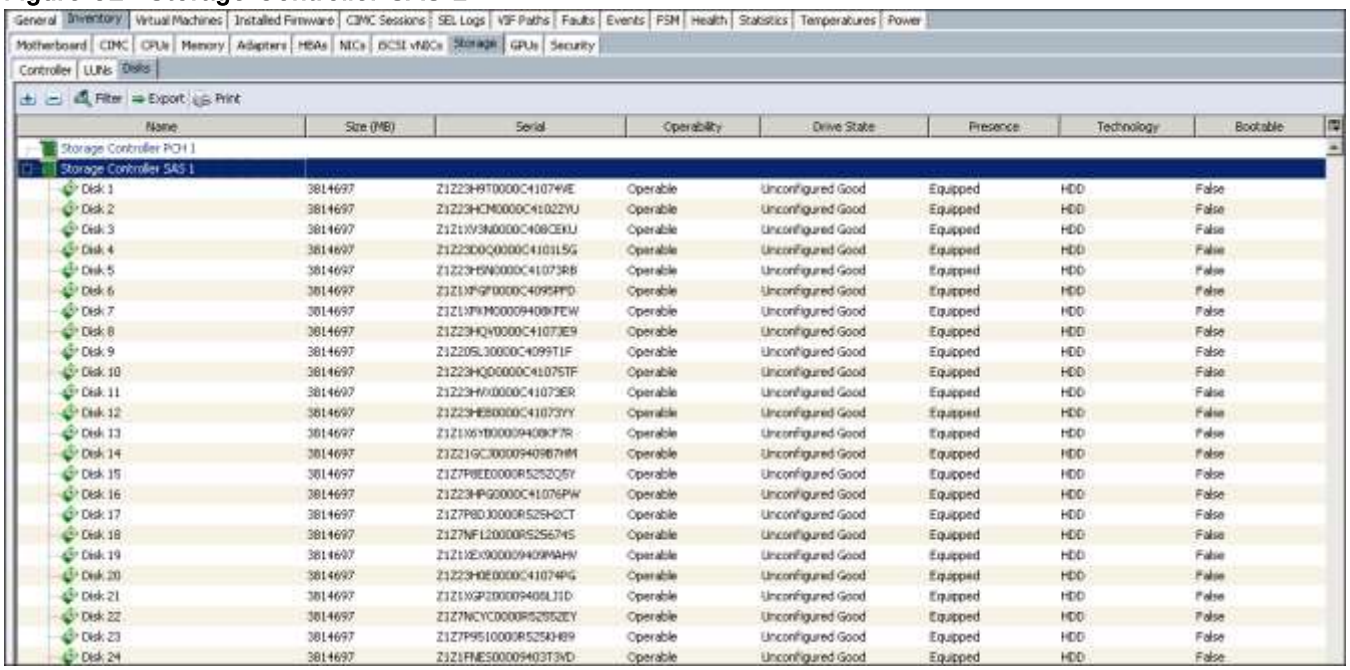


4. Repeat steps 2 and 3 for the rest of the chassis.

5. Once the chassis profile is associated, only 24 disks will be assigned to each server node.

6. To verify that, go to Equipment→Chassis→ 1→Server 1. Click on the Inventory→Storage→Disks. Expand Storage controller SAS 1. (Figure 32)

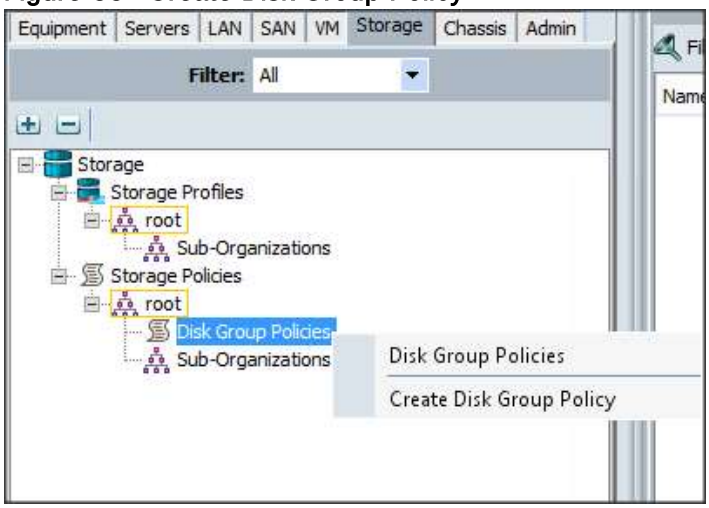
Figure 32 Storage Controller SAS 1



Creating a Storage Profile for Boot Drives

1. Go to Storage and expand Storage→ Storage Policies. Right click on Disk Group Policies and click Create Disk Group Policies. (Figure 33)

Figure 33 Create Disk Group Policy



2. In the Create Disk Policy window, configure the following parameters and click OK. (Figure 34)

a. Name = “Boot_SSD”

b. RAID Level = RAID 1 Mirrored

c. Disk Group Configuration=Automatic

d. Number of Drives=2

e. Drive Type= SSD

f. Use Remaining Disks = checked

g. Strip Size = 64 KB

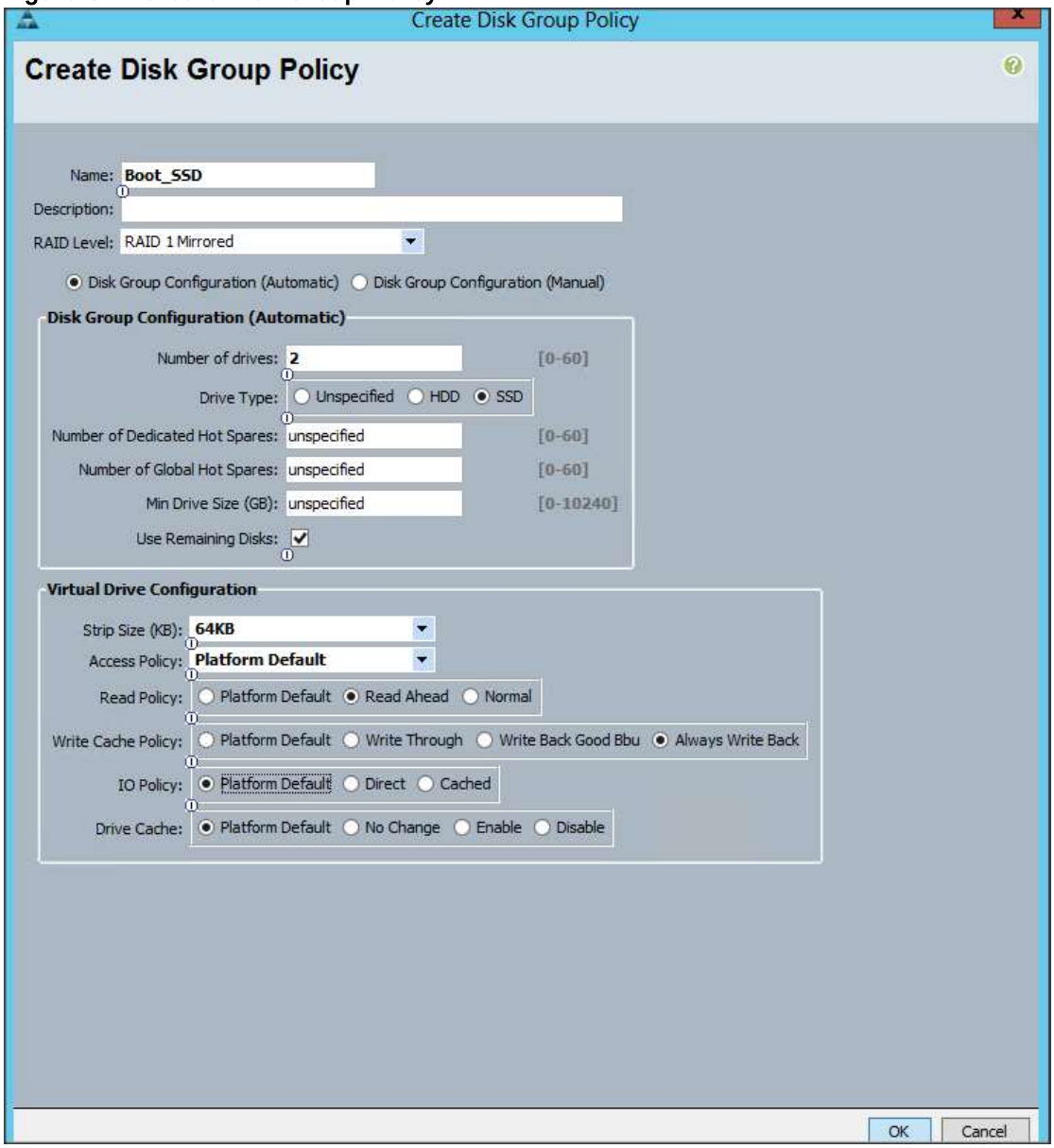
h. Access Policy = Platform Default

i. Read Policy = Read Ahead

j. Write Cache Policy = Always Write Back

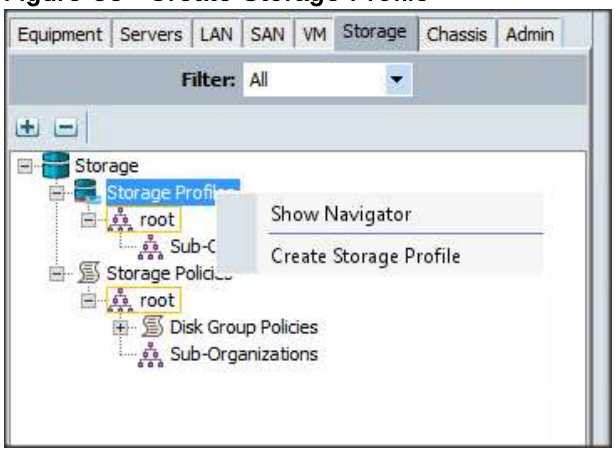
a. IO Policy and Drive Cache = Platform Default

Figure 34 Create Disk Group Policy



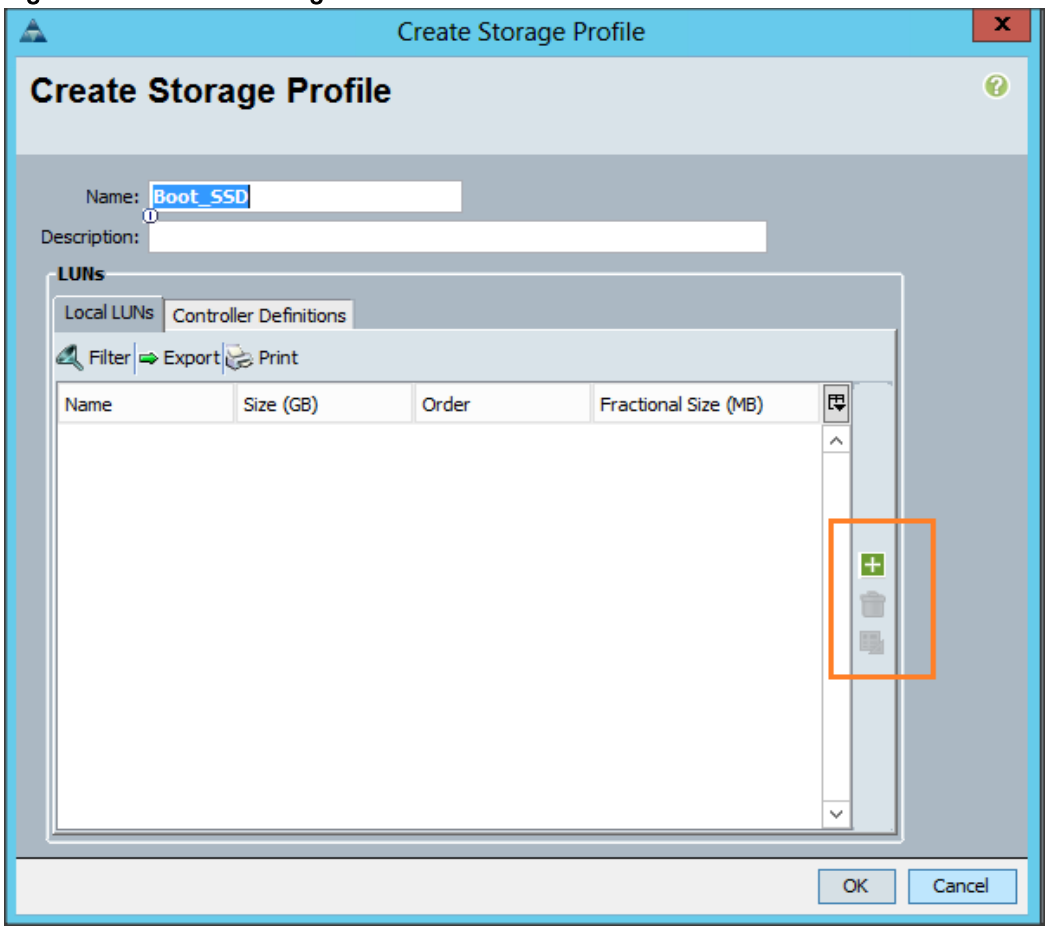
3. Click on the Storage tab. Right click on Storage Profile and click Create Storage Profile. (Figure 35)

Figure 35 Create Storage Profile



4. Enter "Boot_SSD" in the name field. Under Local LUNs click "+" to add local lun. (Figure 36)

Figure 36 Create Storage Profile

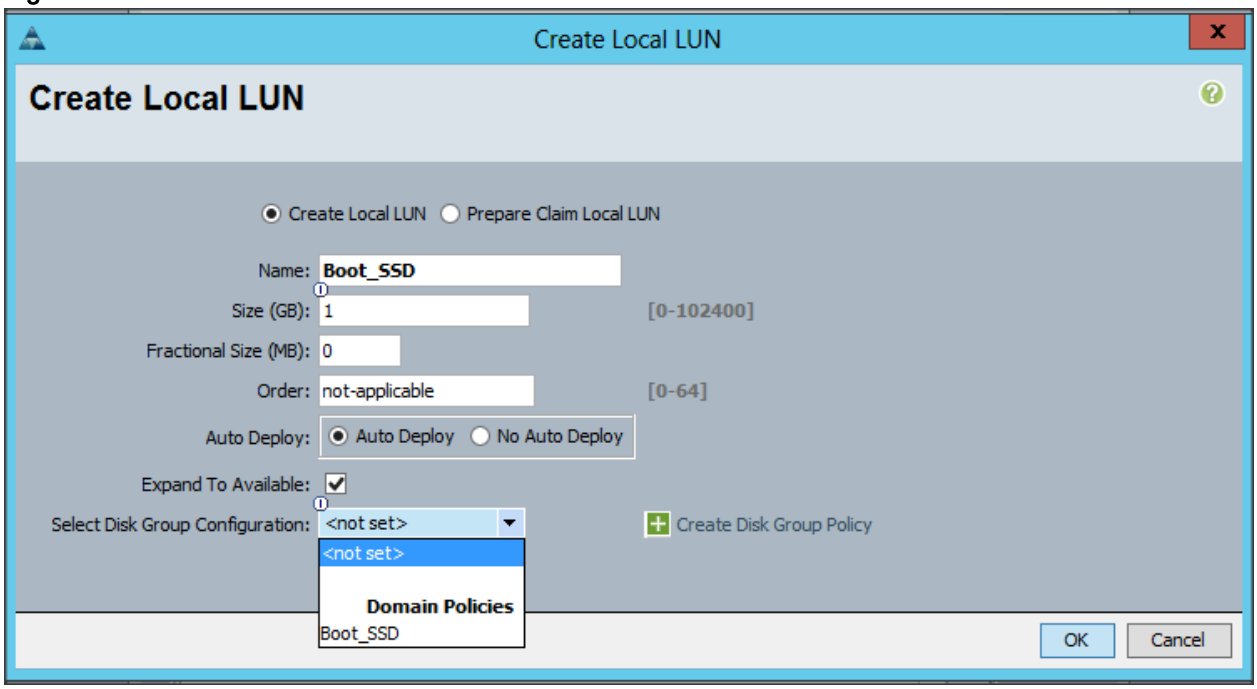


5. In the Create Local LUN window, enter the name "Boot_SSD". (Figure 37)

6. Check the “Expand to Available” checkbox to use all available space.

7. Under Select Disk Group Configuration drop down list choose “Boot_SSD” created earlier and click “OK” and “OK” again to complete the configuration.

Figure 37 Create Local LUN



Creating Pools for Service Profile Templates

Creating MAC Address Pools

To create MAC address pools, complete the following steps:

1. Select the LAN tab on the left of the window.

2. Select Pools > root.

3. Right-click MAC Pools under the root organization.

4. Select Create MAC Pool to create the MAC address pool. Enter ..ool for the ..me of the MAC pool. (Figure 38)

5. (Optional) Enter a description of the MAC pool.

6. Select Assignment Order Sequential.

7. Click Next.

8. Click Add

9. Specify a starting MAC address. (Figure 39)

10. Specify a size of the MAC address pool, which is sufficient to support the available server resources.

11. Click OK

Figure 38 Define Name and Description of MAC Pool

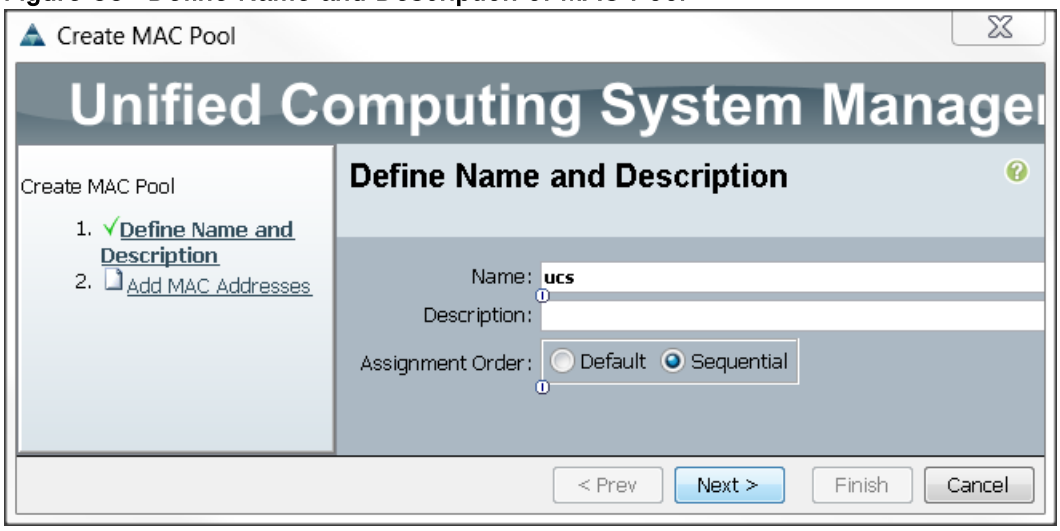
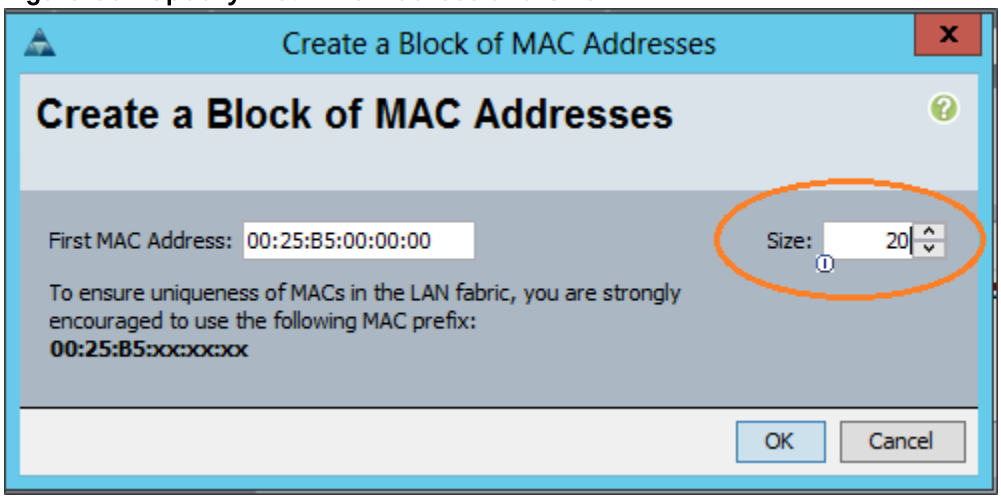
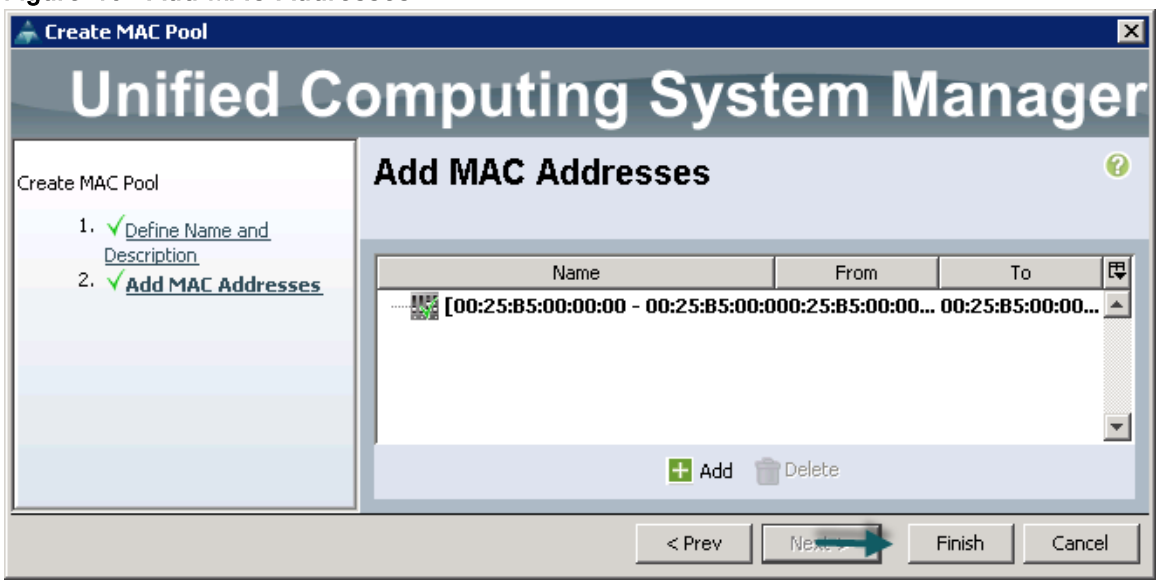


Figure 39 Specify first MAC Address and Size

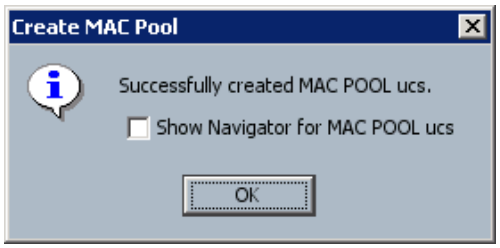


12. Click Finish. (Figure 40)

Figure 40 Add MAC Addresses



13. When the message box displays, click OK



Creating a Server Pool

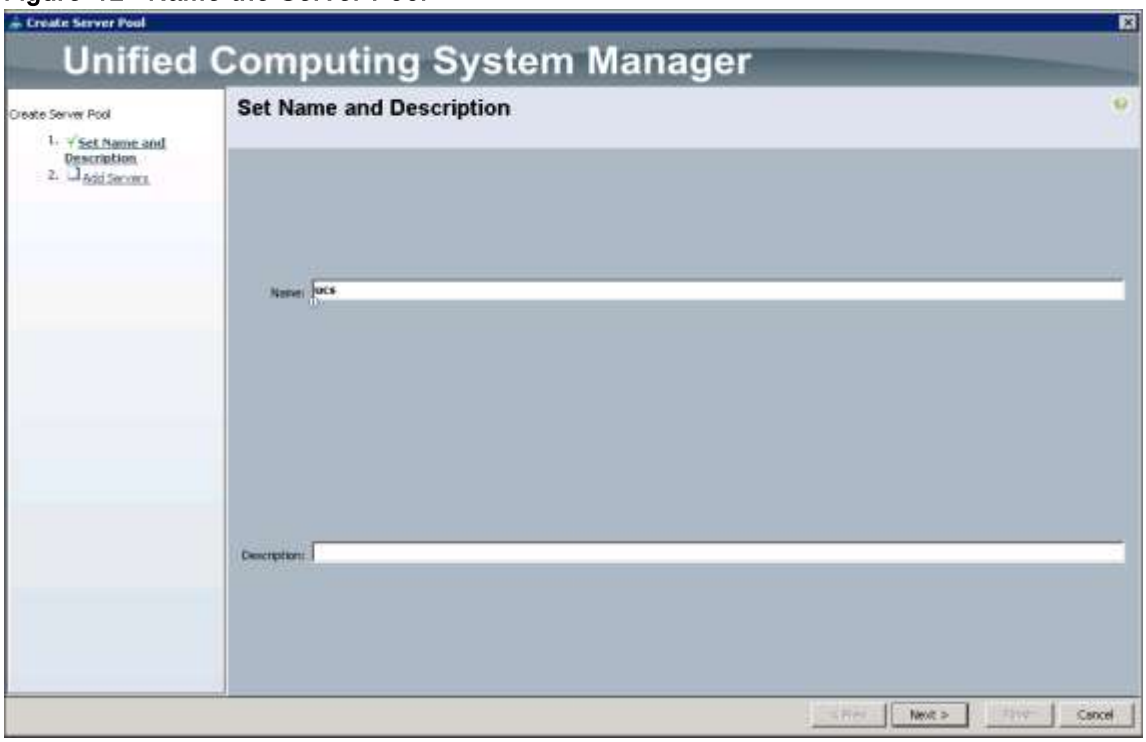
A server pool contains a set of servers. These servers typically share the same characteristics. Those characteristics can be their location in the chassis, or an attribute such as server type, amount of memory, local storage, type of CPU, or local drive configuration. You can manually assign a server to a server pool, or use server pool policies and server pool policy qualifications to automate the assignment

To configure the server pool within the Cisco UCS Manager GUI, complete the following steps:

1. Select the Servers tab in the left pane in the UCS Manager GUI.
2. Select Pools > root.
3. Right-click the Server Pools
4. Select Create Server Pool.
5. Enter your required name ucs for the Server Pool in the name text box. (Figure 41)
6. (Optional) enter a description for the organization.

7. Click Next > to add the servers.

Figure 41 Name the Server Pool



8. Select all the Cisco UCS S3260 Storage Server to be added to the server pool that was previously created (ucs), then Click >> to add them to the pool. (Figure 42)

9. Click Finish

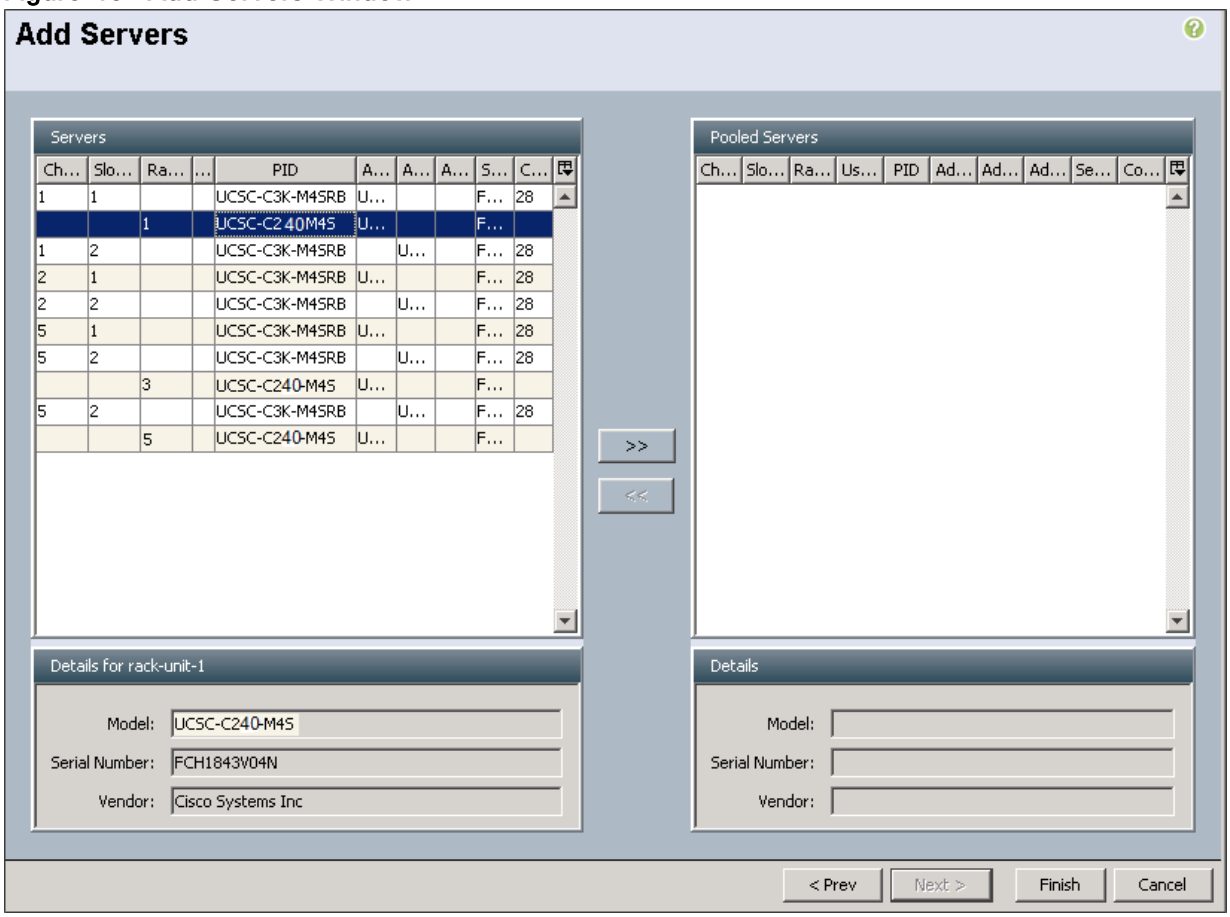
10. Click OK and then click Finish

Figure 42 Add Server

11. Repeat steps 1 through 7 to create another server pool named Management.

12. Select three Cisco UCS C240 M4 Rack Servers to be added to the server pool named Management, then `Click >>` to add them to the pool. (Figure 43)

Figure 43 Add Servers Window



13. Click `Finish`

14. Click OK and then click `Finish`

Creating Policies for Service Profile Templates

Creating Host Firmware Package Policy

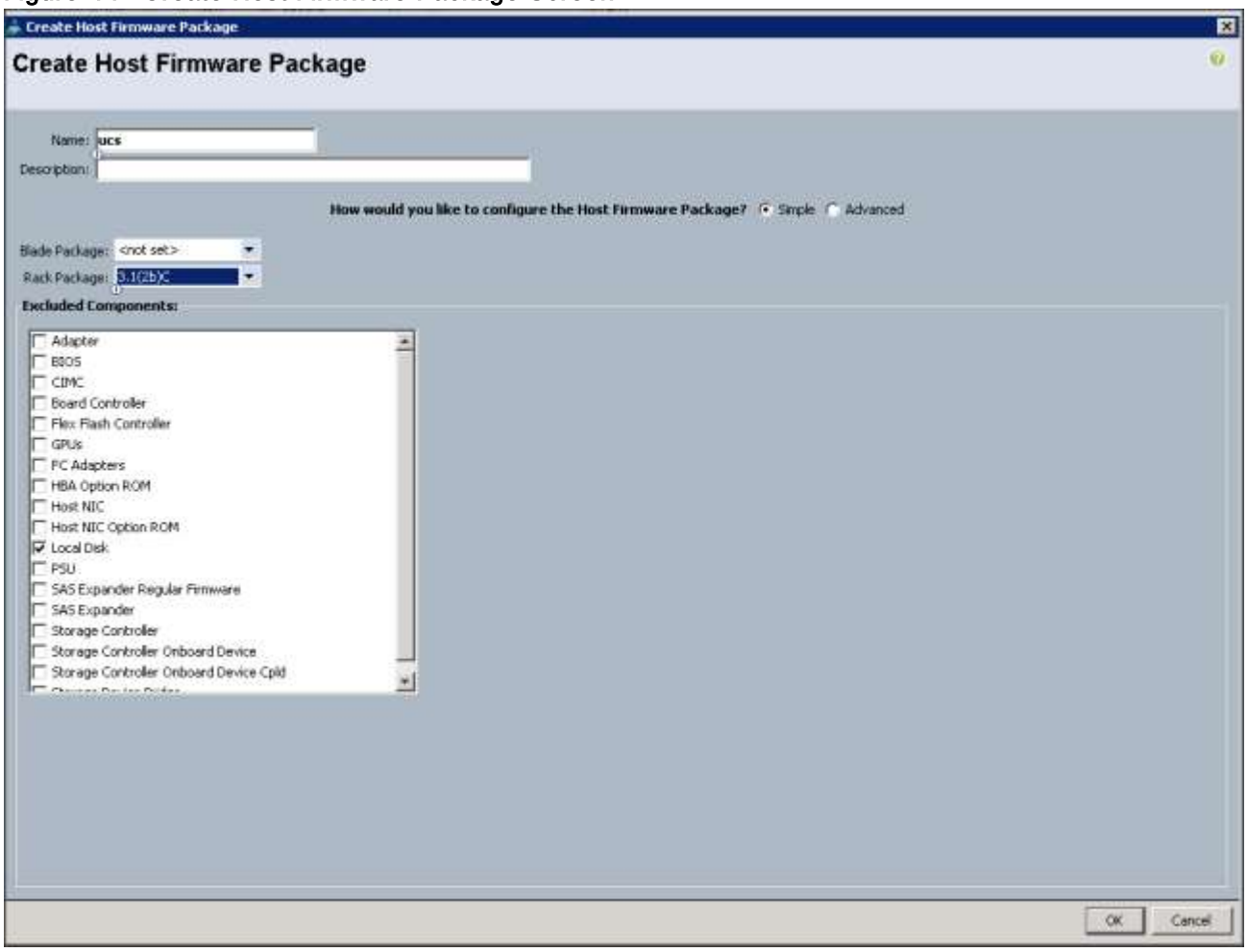
Firmware management policies allow the administrator to select the corresponding packages for a given server configuration. These include adapters, BIOS, board controllers, FC adapters, HBA options, and storage controller properties as applicable.

To create a firmware management policy for a given server configuration using the Cisco UCS Manager GUI, complete the following steps:

1. Select the `Servers` tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click Host Firmware Packages.

4. Select Create Host Firmware Package.
5. Enter the required Host Firmware package name (ucs). (Figure 44)
6. Select `Simple` radio button to configure the Host Firmware package.
7. Select the appropriate Rack package that has been installed.
8. Click `OK` to complete creating the management firmware package
9. Click `OK`

Figure 44 Create Host Firmware Package Screen



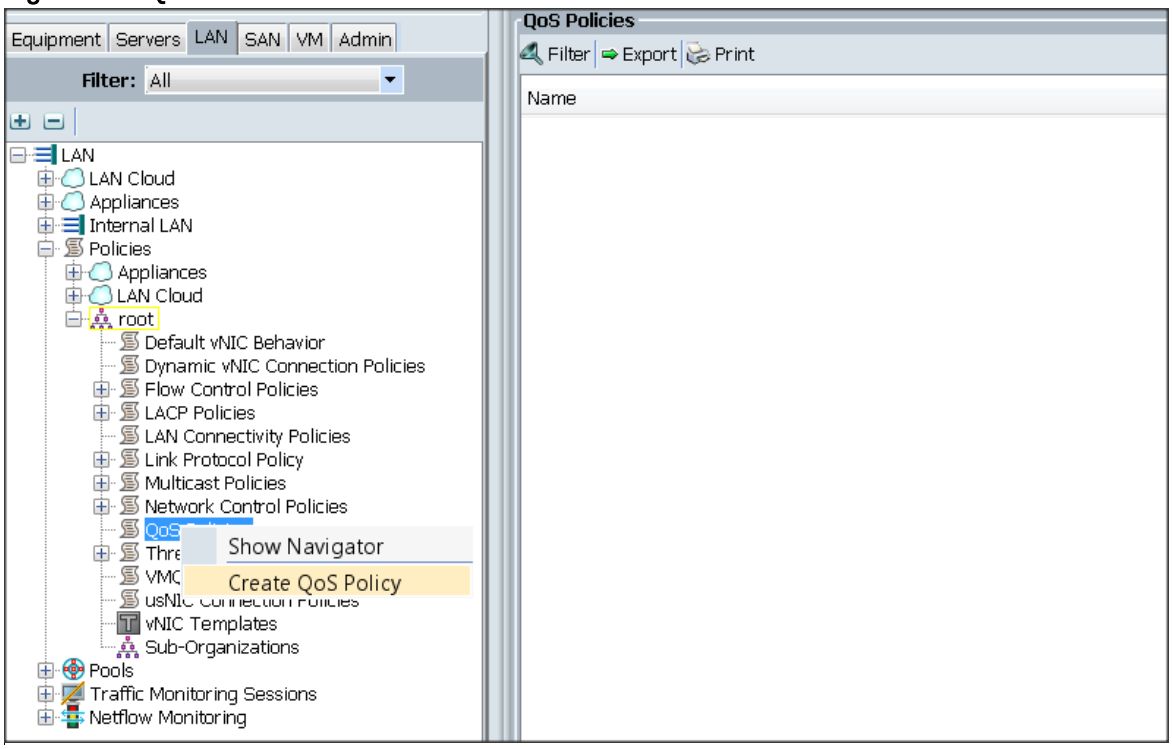
Creating QoS Policies

To create the QoS policy for a given server configuration using the Cisco UCS Manager GUI, complete the following steps:

Platinum Policy

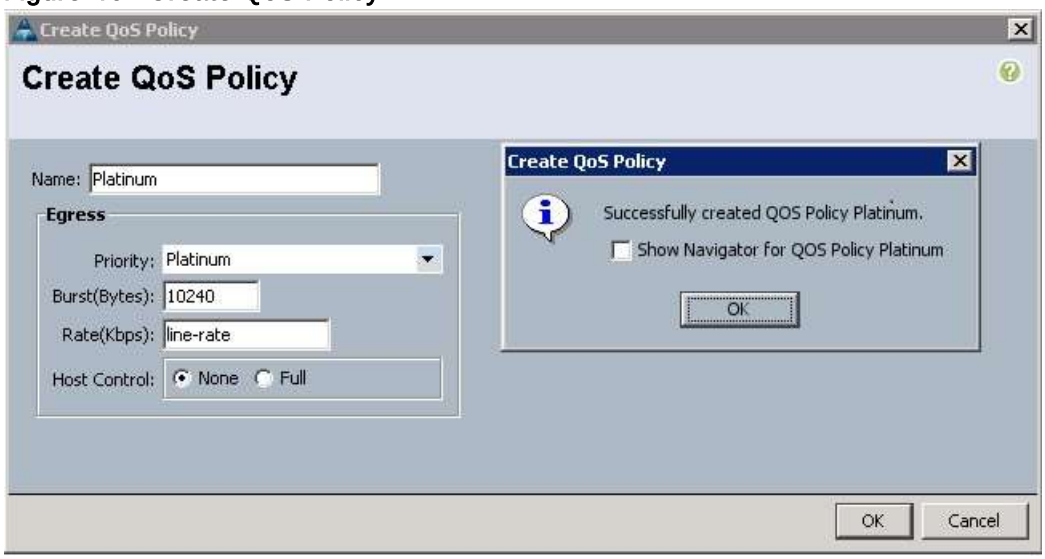
1. Select the LAN tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click QoS Policies.
4. Select Create QoS Policy. (Figure 45)

Figure 45 QoS Policies



5. Enter `Platinum` as the name of the policy. (Figure 46)
6. Select `Platinum` from the drop down menu.
7. Keep the `Burst(Bytes)` field set to default (10240).
8. Keep the `Rate(Kbps)` field set to default (line-rate).
9. Keep `Host Control` radio button set to default (none).
10. Once the pop-up window appears, click OK to complete the creation of the Policy.

Figure 46 Create QoS Policy

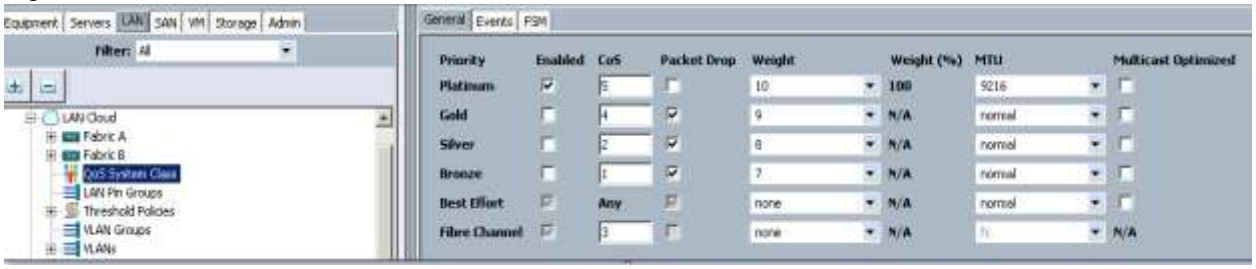


Setting Jumbo Frames

To set Jumbo frames and enable QoS, complete the following steps:

1. Select the LAN tab in the left pane in the UCSM GUI.
2. Select LAN Cloud > QoS System Class. (Figure 47)
3. In the right pane, select the General tab
4. In the Platinum row, enter 9216 for MTU.
5. Check the Enabled Check box next to Platinum
6. In the Best Effort row, select none for weight.
7. In the Fiber Channel row, select none for weight.
8. Click Save Changes.
9. Click OK

Figure 47 LAN General

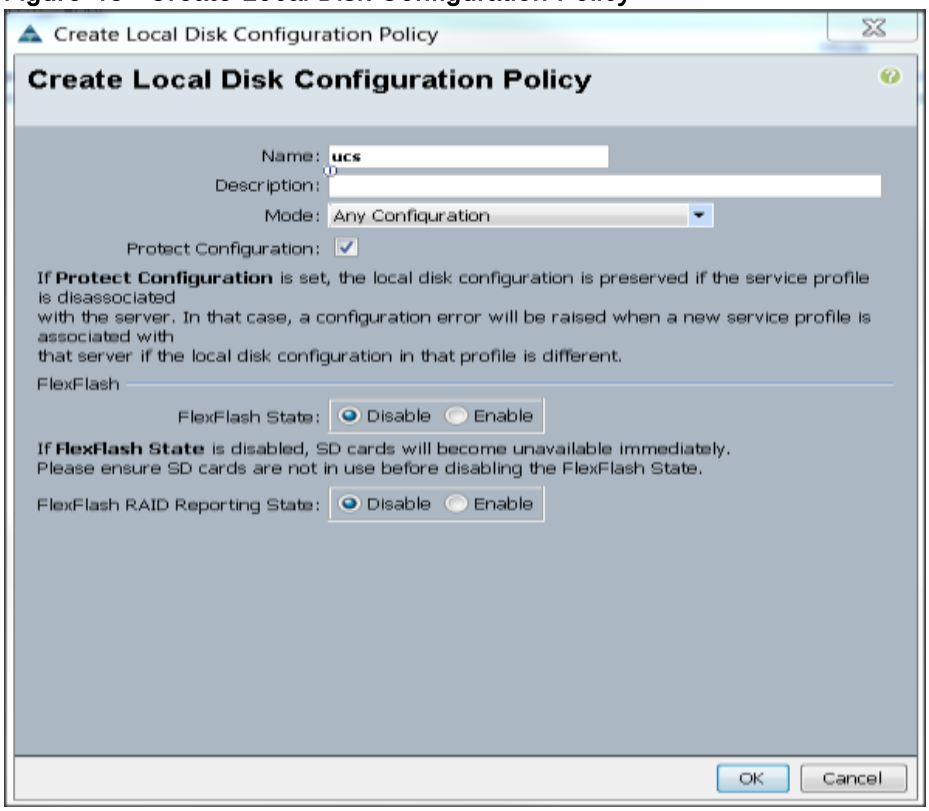


Creating the Local Disk Configuration Policy

To create the local disk configuration policy in the Cisco UCS Manager GUI, complete the following steps:

1. Select the `Servers` tab on the left pane in the UCS Manager GUI.
2. Go to Policies > root.
3. Right-click Local Disk Configuration Policies.
4. Select Create Local Disk Configuration Policy.
5. Enter `ucs` as the local disk configuration policy name. (Figure 48)
6. Change the Mode to Any Configuration. Check the Protect Configuration box.
7. Keep the `FlexFlash State` field as `default` (Disable).
8. Keep the `FlexFlash RAID Reporting` State field as `default` (Disable).
9. Click `OK` to complete the creation of the Local Disk Configuration Policy.
10. Click `OK`.

Figure 48 Create Local Disk Configuration Policy



Creating a Server BIOS Policy

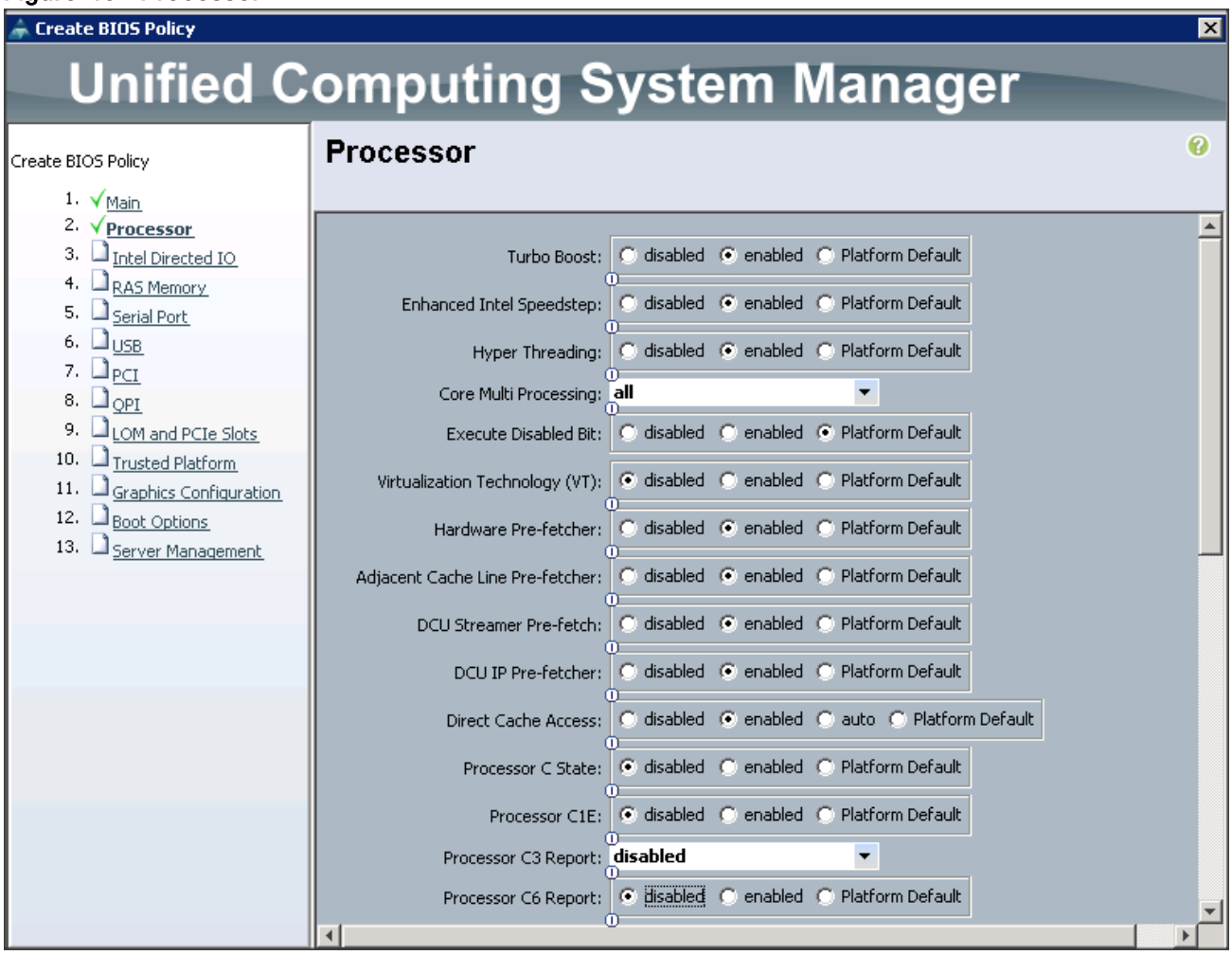
The BIOS policy feature in Cisco UCS automates the BIOS configuration process. The traditional method of setting the BIOS is manually, and is often error-prone. By creating a BIOS policy and assigning the policy to a server or group of servers, can enable transparency within the BIOS settings configuration.

Note: *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.

To create a server BIOS policy using the Cisco UCS Manager GUI, complete the following steps:

1. Select the `Servers` tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click `BIOS Policies`
4. Select Create BIOS Policy.
5. Enter your preferred BIOS policy name (...ce).
6. Change the BIOS settings as shown in the following figures.
7. The only changes that need to be made are in the Processor (Figure 49) and RAS Memory settings (Figure 50).

Figure 49 Processor



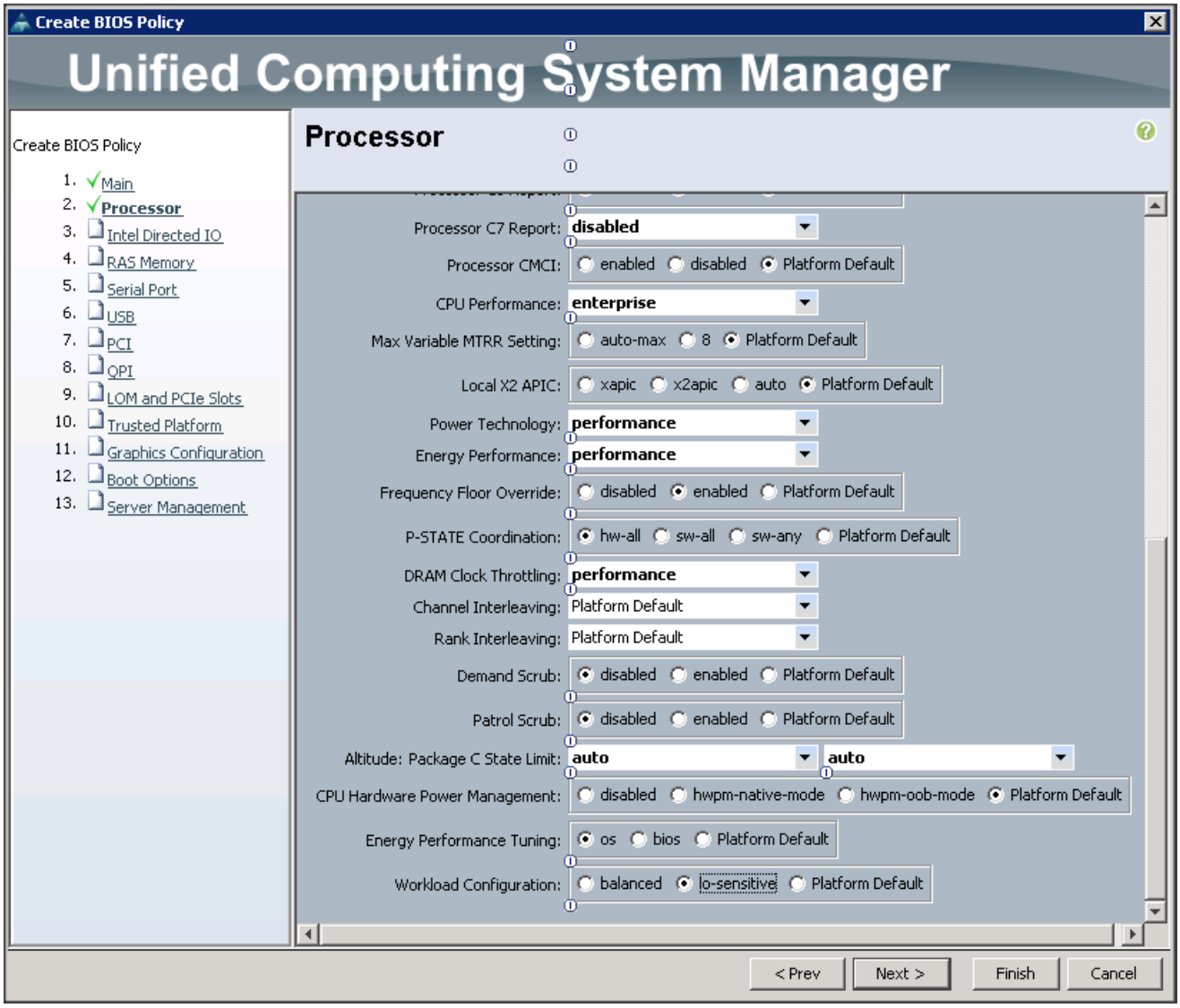
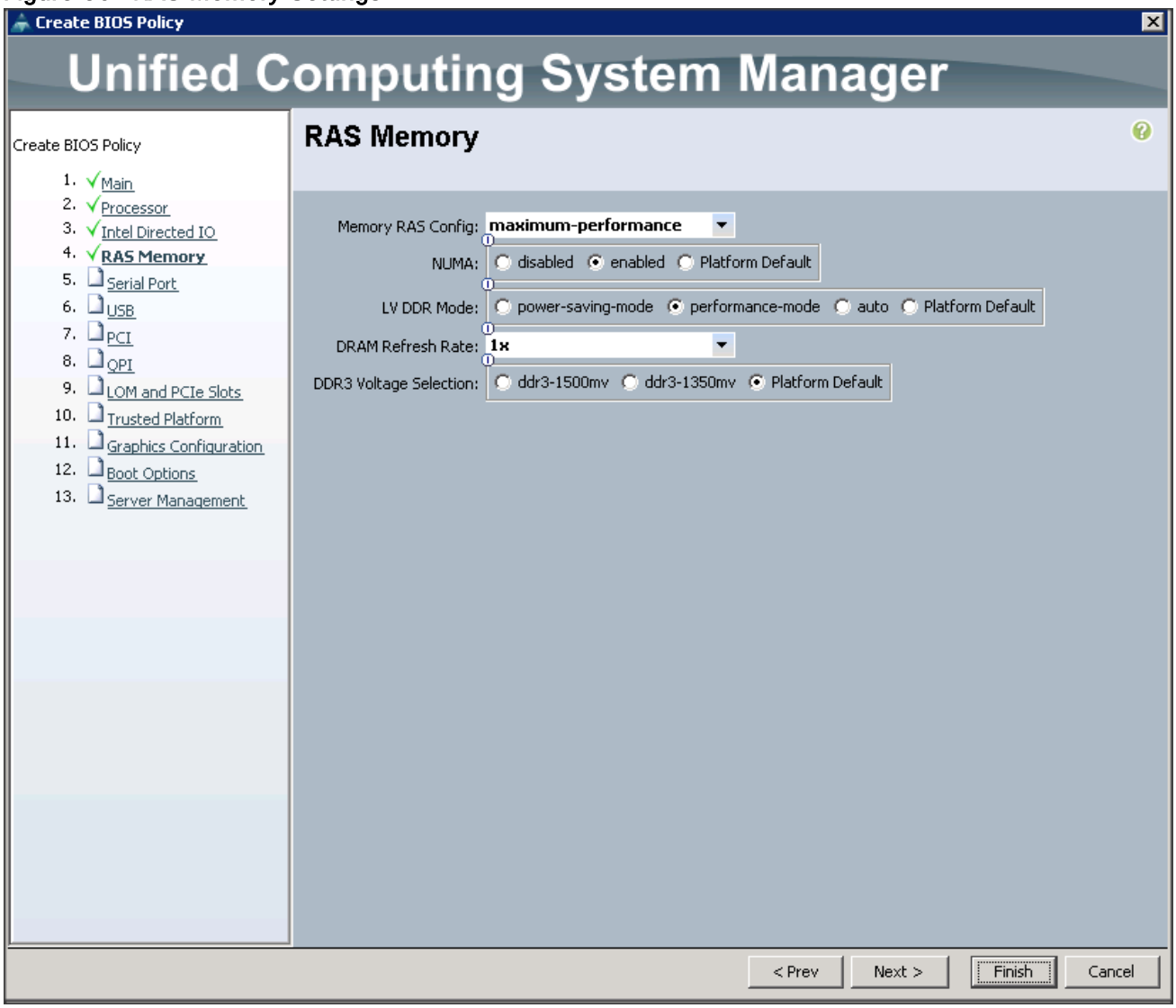

Create BIOS Policy
Unified Computing System Manager
Create BIOS Policy
1. Main
2. Processor
3. Intel Directed IO
4. RAS Memory
5. Serial Port
6. USB
7. PCI
8. QPI
9. LOM and PCIe Slots
10. Trusted Platform
11. Graphics Configuration
12. Boot Options
13. Server Management
Processor
Processor C7 Report: disabled
Processor CMCI: enabled disabled Platform Default
CPU Performance: enterprise
Max Variable MTRR Setting: auto-max 8 Platform Default
Local X2 APIC: xapic x2apic auto Platform Default
Power Technology: performance
Energy Performance: performance
Frequency Floor Override: disabled enabled Platform Default
P-STATE Coordination: hw-all sw-all sw-any Platform Default
DRAM Clock Throttling: performance
Channel Interleaving: Platform Default
Rank Interleaving: Platform Default
Demand Scrub: disabled enabled Platform Default
Patrol Scrub: disabled enabled Platform Default
Altitude: Package C State Limit: auto auto
CPU Hardware Power Management: disabled hwpm-native-mode hwpm-oob-mode Platform Default
Energy Performance Tuning: os bios Platform Default
Workload Configuration: balanced lo-sensitive Platform Default
< Prev
Next >
Finish
Cancel

Figure 50 RAS Memory Settings

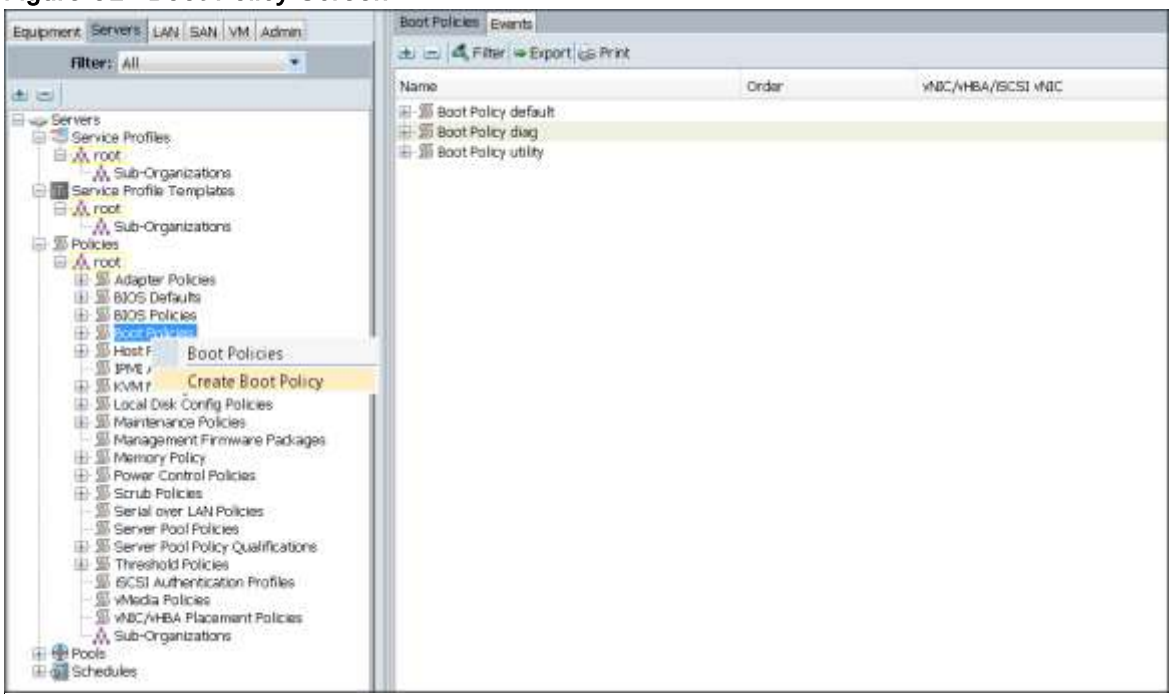


Creating the Boot Policy

To create boot policies within the Cisco UCS Manager GUI, complete the following steps:

1. Select the `Servers` tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click the `Boot Policies`.
4. Select Create Boot Policy. (Figure 51)

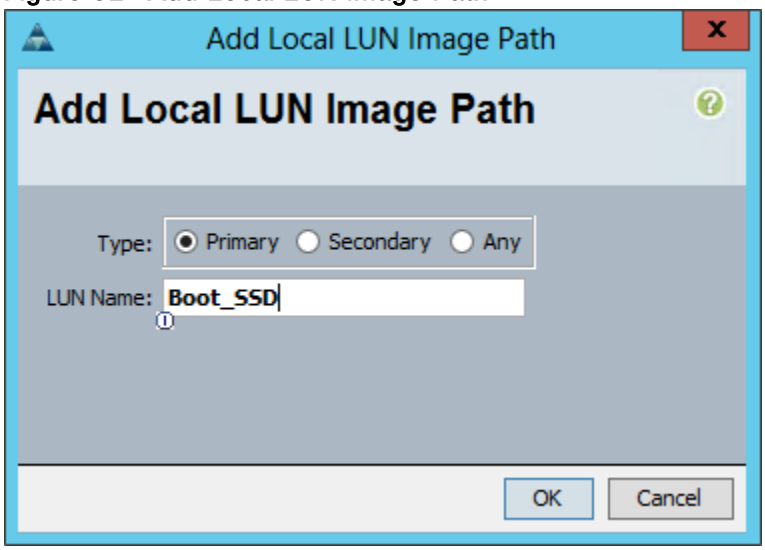
Figure 51 Boot Policy Screen



5. Enter ..ce as 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 and select Add Local Lun.
11. In the Add Local LUN Image Path window, select Primary and enter the Name "Boot_SSD" that was created earlier during storage profile creation step. (Figure 52)

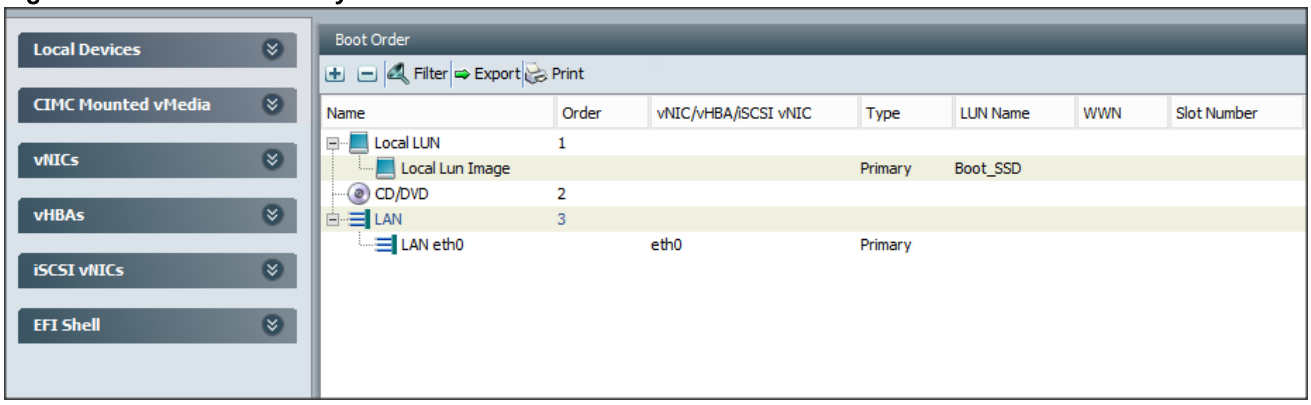
Note: The LUN name must match with the LUN name created earlier.

Figure 52 Add Local LUN Image Path



12. Expand Local Devices > Add CD/DVD and select Add Local CD/DVD. (Figure 53)
13. Expand vNICs and select Add LAN Boot and enter eth0
14. Click OK to add the Boot Policy.
15. Click OK

Figure 53 Add Boot Policy



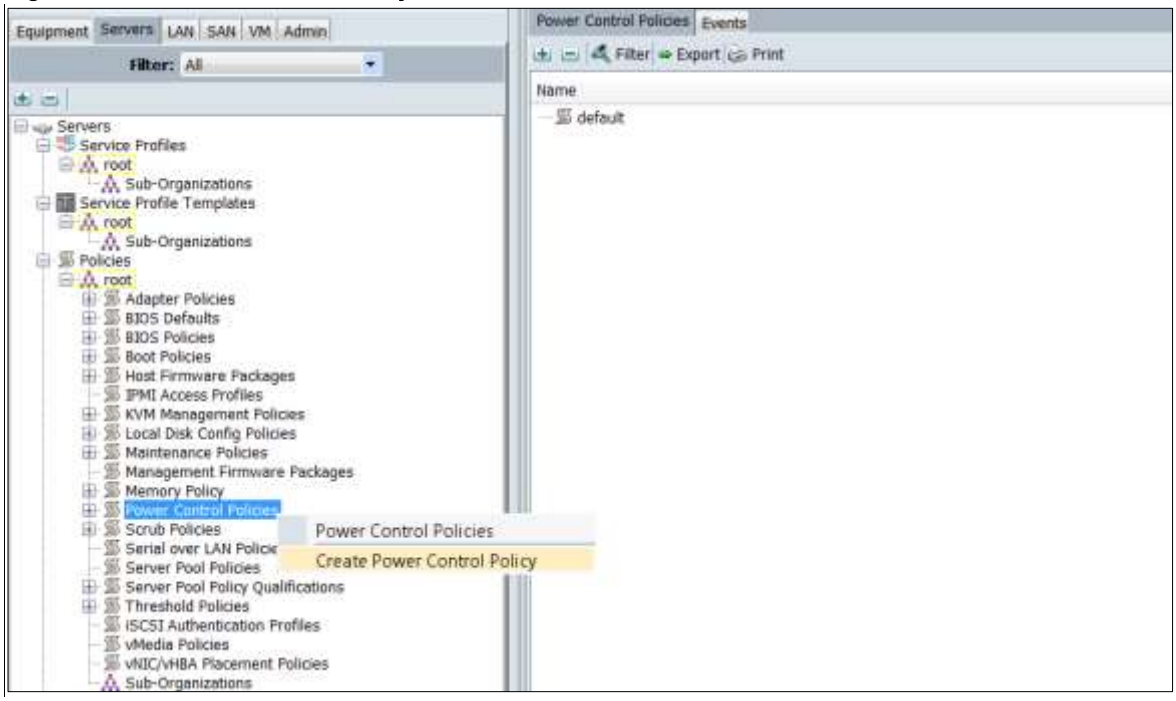
Creating Power Control Policy

To create Power Control policies within the Cisco UCS Manager GUI, complete the following steps:

1. Select the Servers tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click the Power Control Policies. (Figure 54)

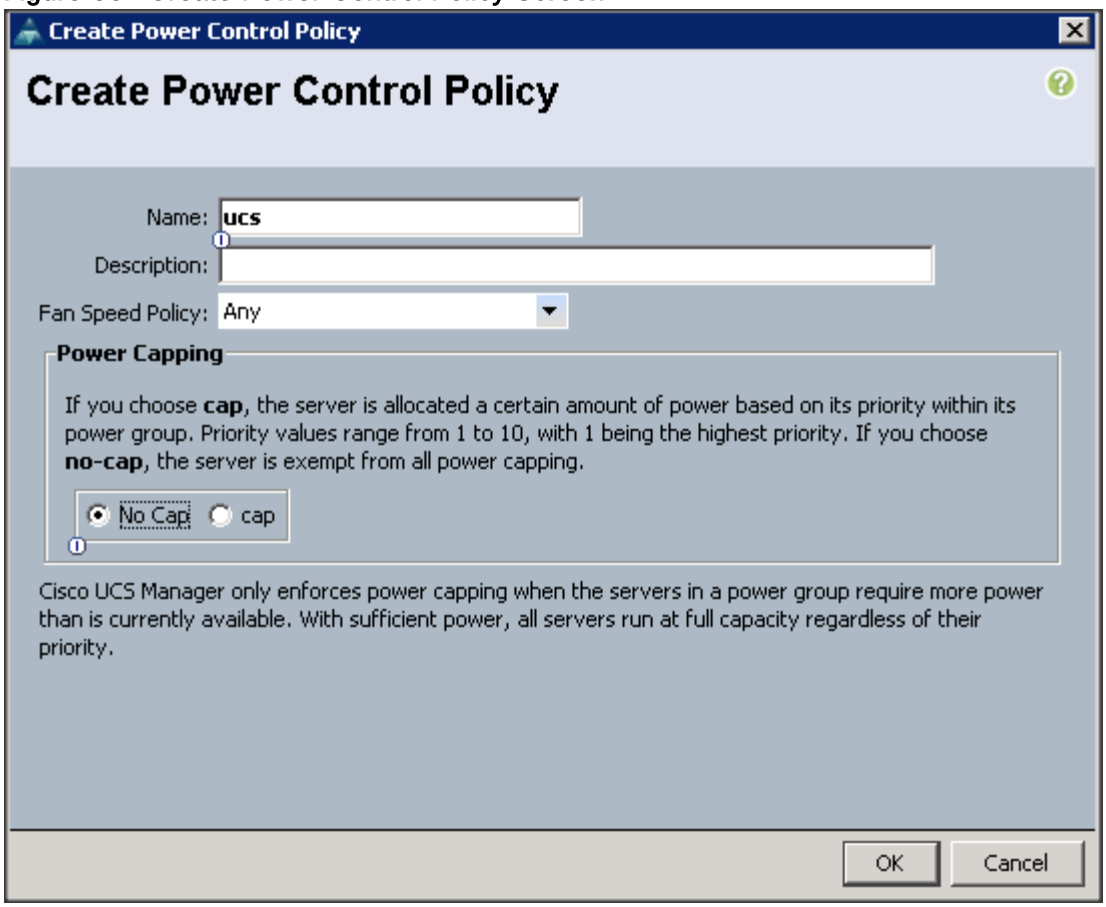
4. Select Create Power Control Policy.

Figure 54 Power Control Policy



5. Enter ..ce as the Power Control policy ..ame. (Figure 55)
6. (Optional) enter a description for the boot policy.
7. Select Performance for Fan Speed Policy.
8. Select `No cap` for `Power Capping` selection.
9. Click OK to create the Power Control Policy.
10. Click OK

Figure 55 Create Power Control Policy Screen

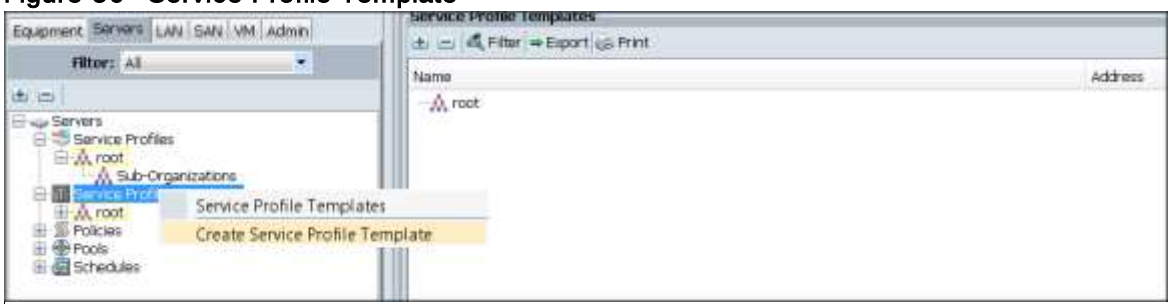


Creating a Service Profile Template

To create a Service Profile Template, complete the following steps:

1. Select the `Servers` tab in the left pane in the UCSM GUI.
2. Right-click Service Profile Templates.
3. Select `Create Service Profile Template`. (Figure 56)

Figure 56 Service Profile Template

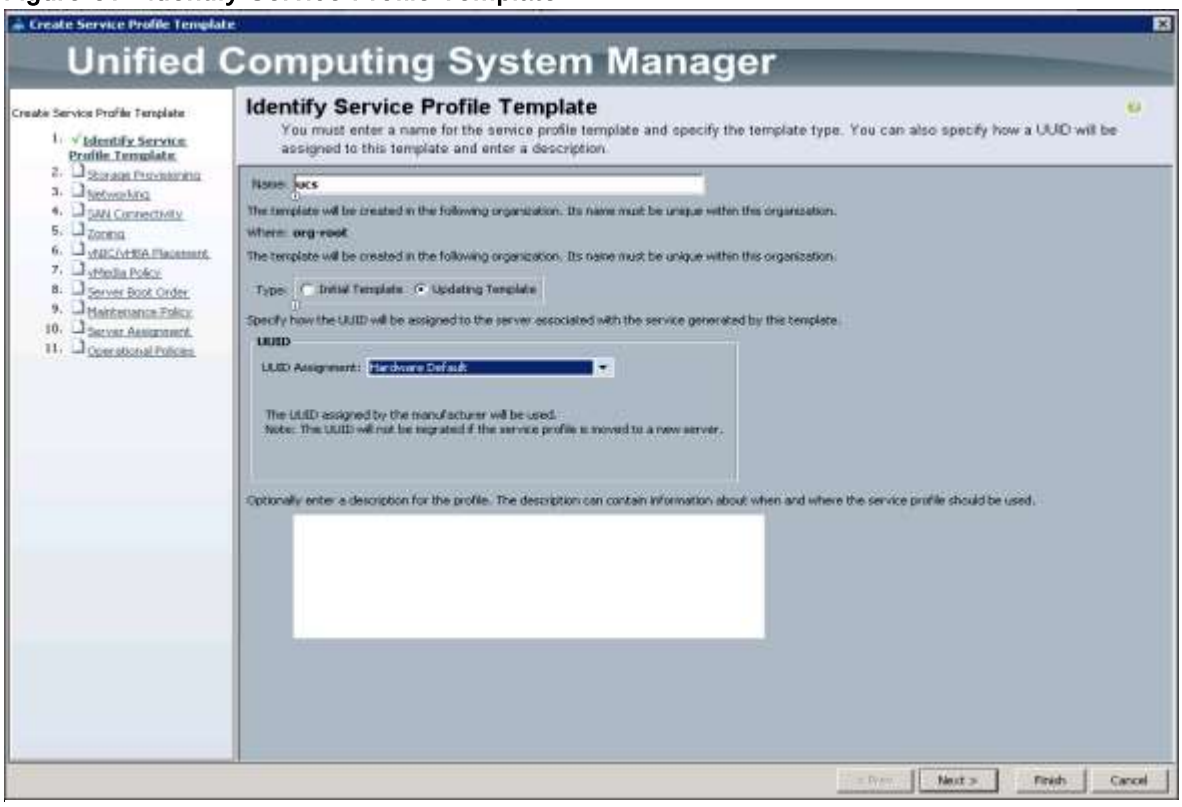


The Create Service Profile Template window appears. (Figure 57)

To identify the service profile template, complete the following steps:

4. Name the service profile template as ucs. Select the `Updating Template` radio button.
5. In the UUID section, select `Hardware Default` as the UUID pool.
6. Click Next to continue to the next section.

Figure 57 Identify Service Profile Template



Configuring the Storage Provisioning for the Template

To configure storage policies, complete the following steps:

1. Go to Storage Profile Policy tab, and select Boot_SSD from the drop down list. (Figure 58)

Figure 58 Storage Provisioning

2. Go to the Local Disk Configuration Policy tab, and select ..... for the Local Storage. (Figure 59)

3. Click Next .. to continue to the next section.

Figure 59 Storage Provisioning /Local Storage

4. Click Next once the Networking window appears, then to go to the next section.

Configuring Network Settings for the Template

1. Keep the `Dynamic vNIC Connection Policy` field at the default. (Figure 60)
2. Select the `Expert` radio button for the option, " How would you like to configure LAN connectivity?"
3. Click `Add` to add a vNIC to the template.

Figure 60 Networking

4. The Create vNIC window displays. Name the vNIC eth0. (Figure 61)
5. Select Auto in the Mac Address Assignment pool.
6. Select the Fabric A radio button and check the Enable failover check box for the Fabric ID.
7. Check the VLAN76 check box for VLANs and select the Native VLAN radio button
8. Select MTU size as 9000
9. Select adapter policy as Linux.
10. Select QoS Policy as Platinum.
11. Keep the Network Control Policy as Default.
12. Click OK

Figure 61 Create vNIC

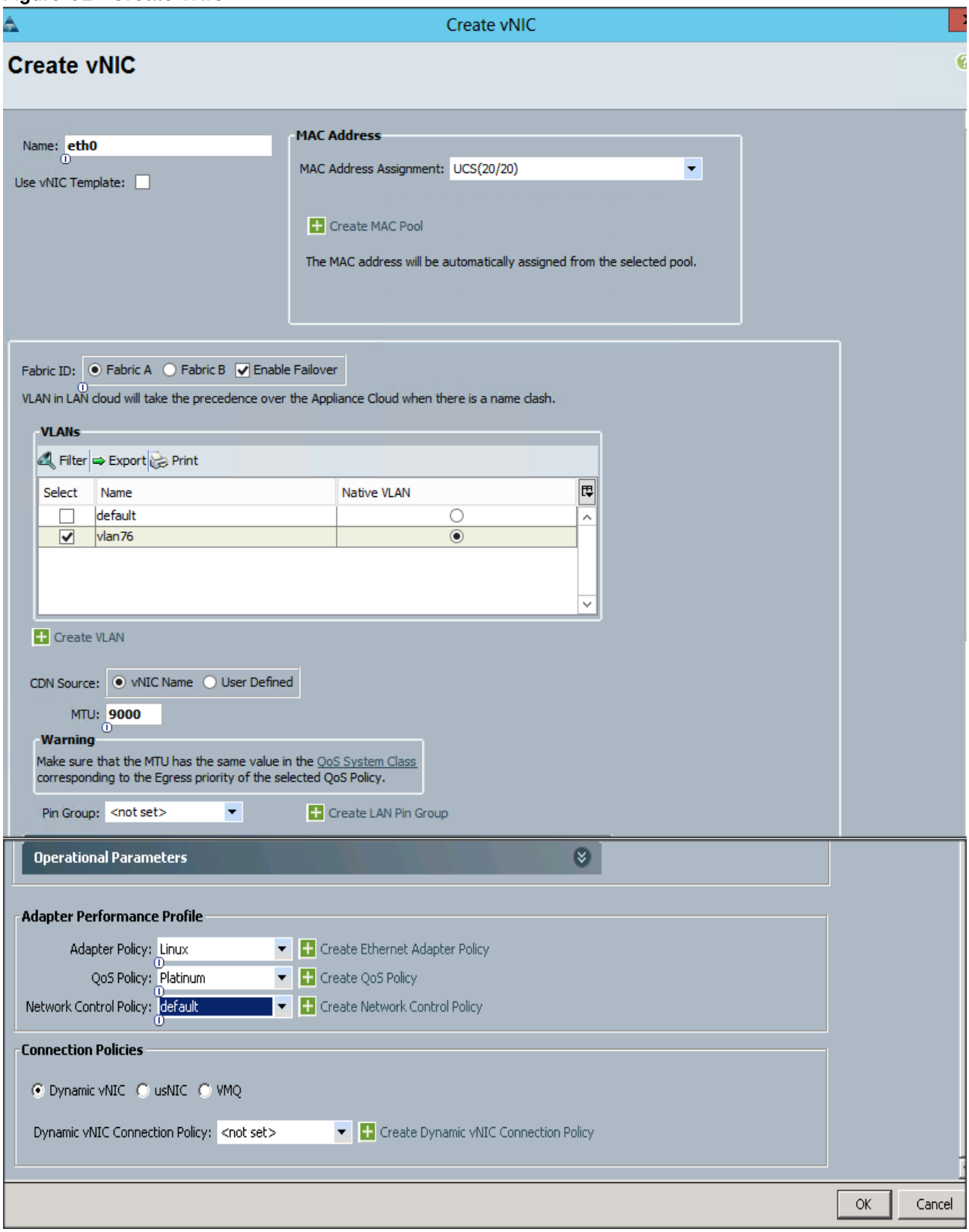
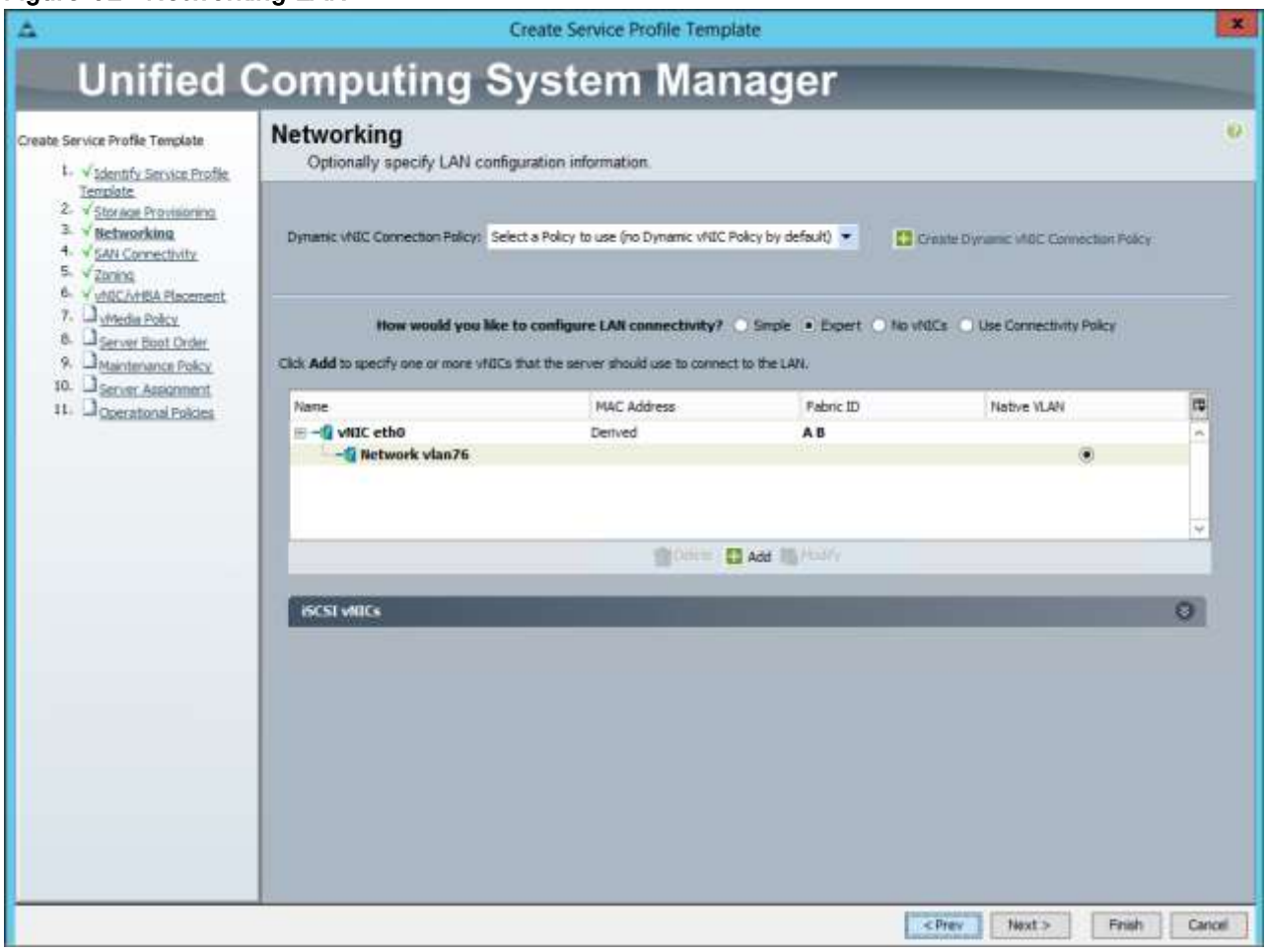


Figure 62 Networking LAN



13. Click Next to continue with SAN Connectivity. (Figure 62)

14. Select no vHBAs for How would you like to configure SAN Connectivity? (Figure 63)

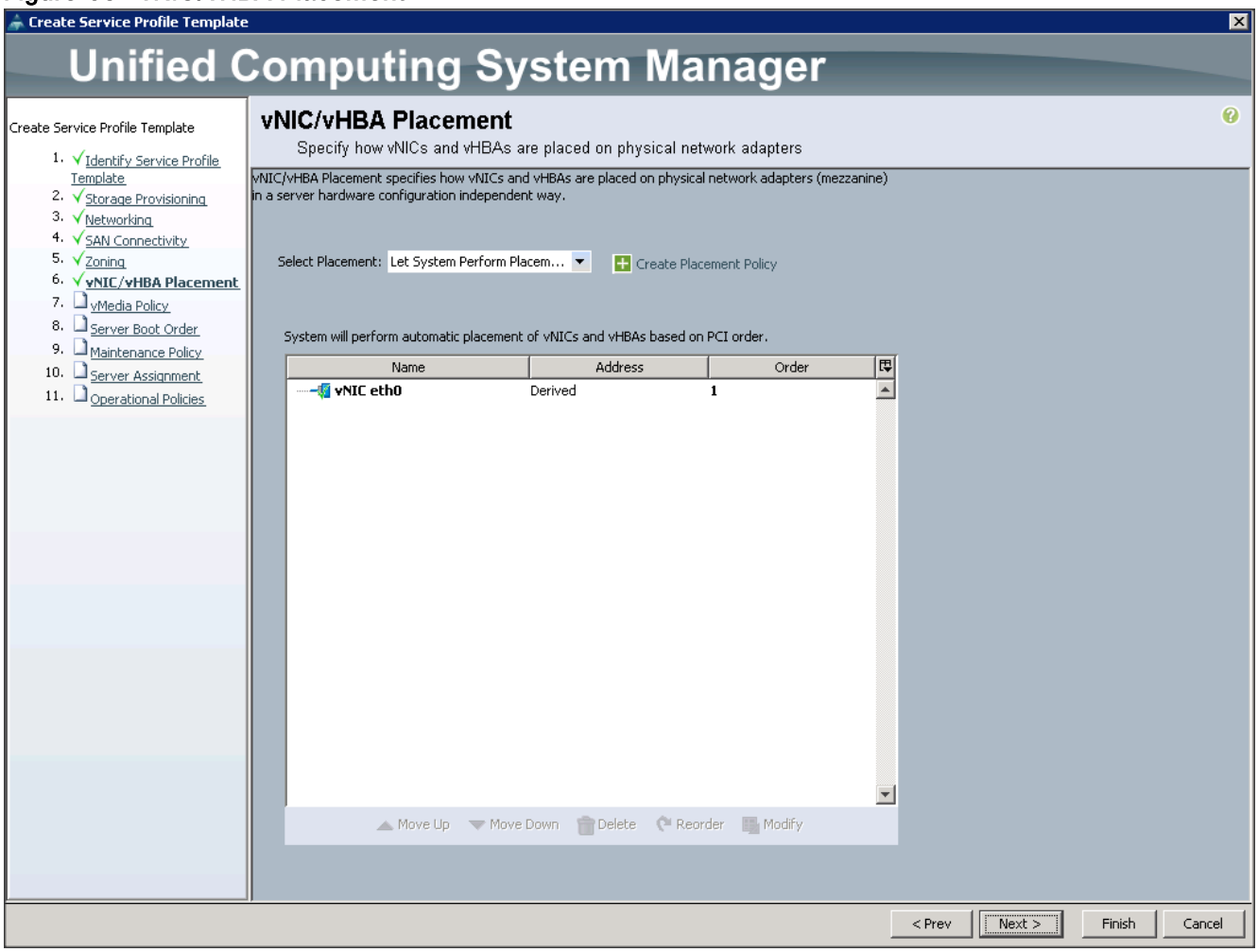
Figure 63 SAN Connectivity

15. Click Next to continue with Zoning. (Figure 64)

Figure 64 Zoning

16. Click Next to continue with vNIC/vHBA placement. (Figure 65)

Figure 65 vNIC/vHBA Placement

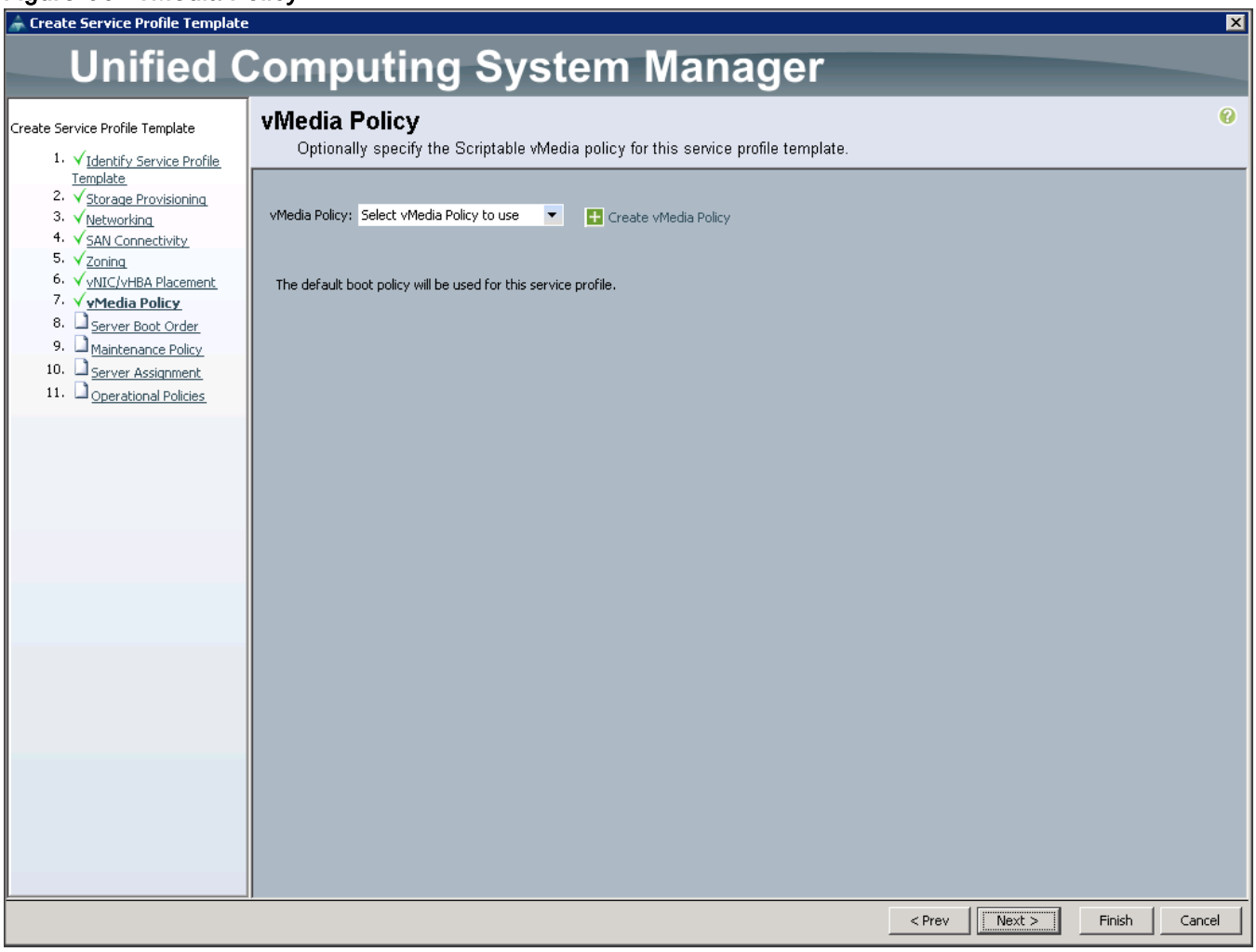


17. Click Next to configure vMedia Policy.

Configuring the vMedia Policy for the Template

1. Click Next. When the vMedia Policy window appears, to go to the next section. (Figure 66)

Figure 66 vMedia Policy

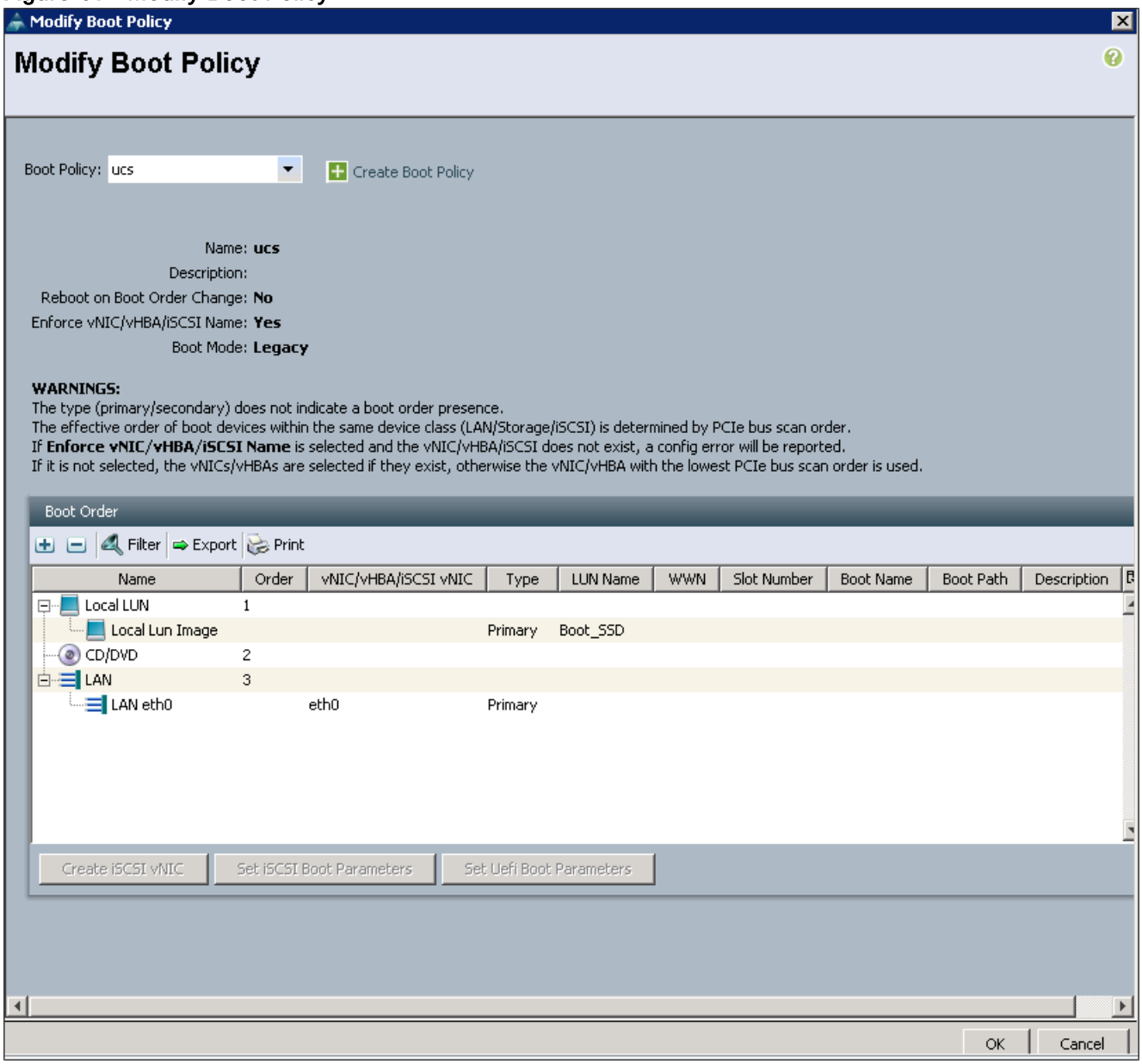


Configuring Server Boot Order for the Template

To set the boot order for the servers, complete the following steps:

1. Select default in the Boot Policy name field. (Figure 67)
2. Review to make sure that all of the boot devices were created and identified.
3. Verify that the boot devices are in the correct boot sequence.
4. Click OK
5. Click Next to continue to the next section.

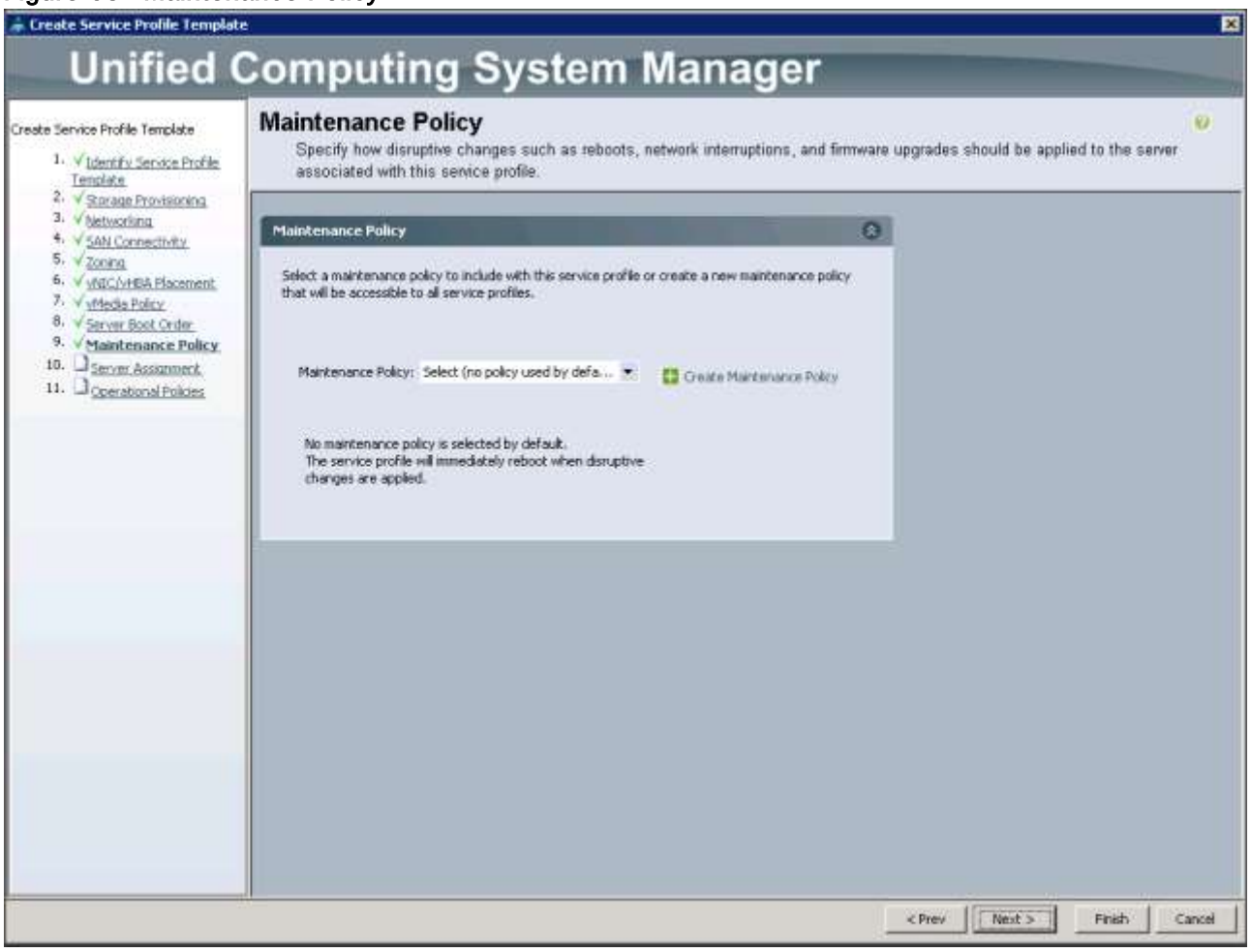
Figure 67 Modify Boot Policy



6. In the Maintenance Policy window, apply the maintenance policy. (Figure 68)

7. Keep the Maintenance policy at no policy used by default. Click Next to continue to the next section.

Figure 68 Maintenance Policy

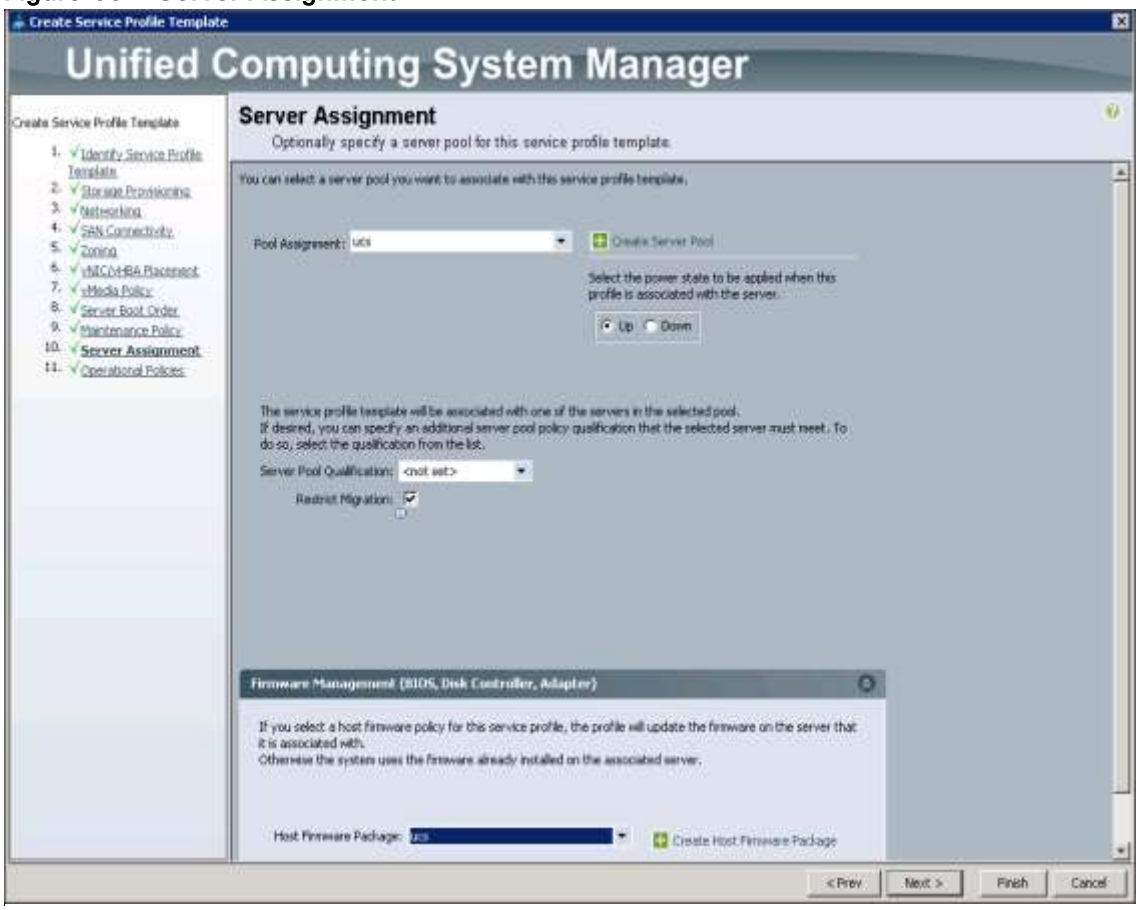


Configuring Server Assignment for the Template

In the Server Assignment window, to assign the servers to the pool, complete the following steps:

1. Select ucs for the Pool Assignment field. (Figure 69)
2. Select the power state to be Up.
3. Keep the Server Pool Qualification field set to <not set>.
4. Check the Restrict Migration check box.
5. Select ucs in Host Firmware Package.

Figure 69 Server Assignment

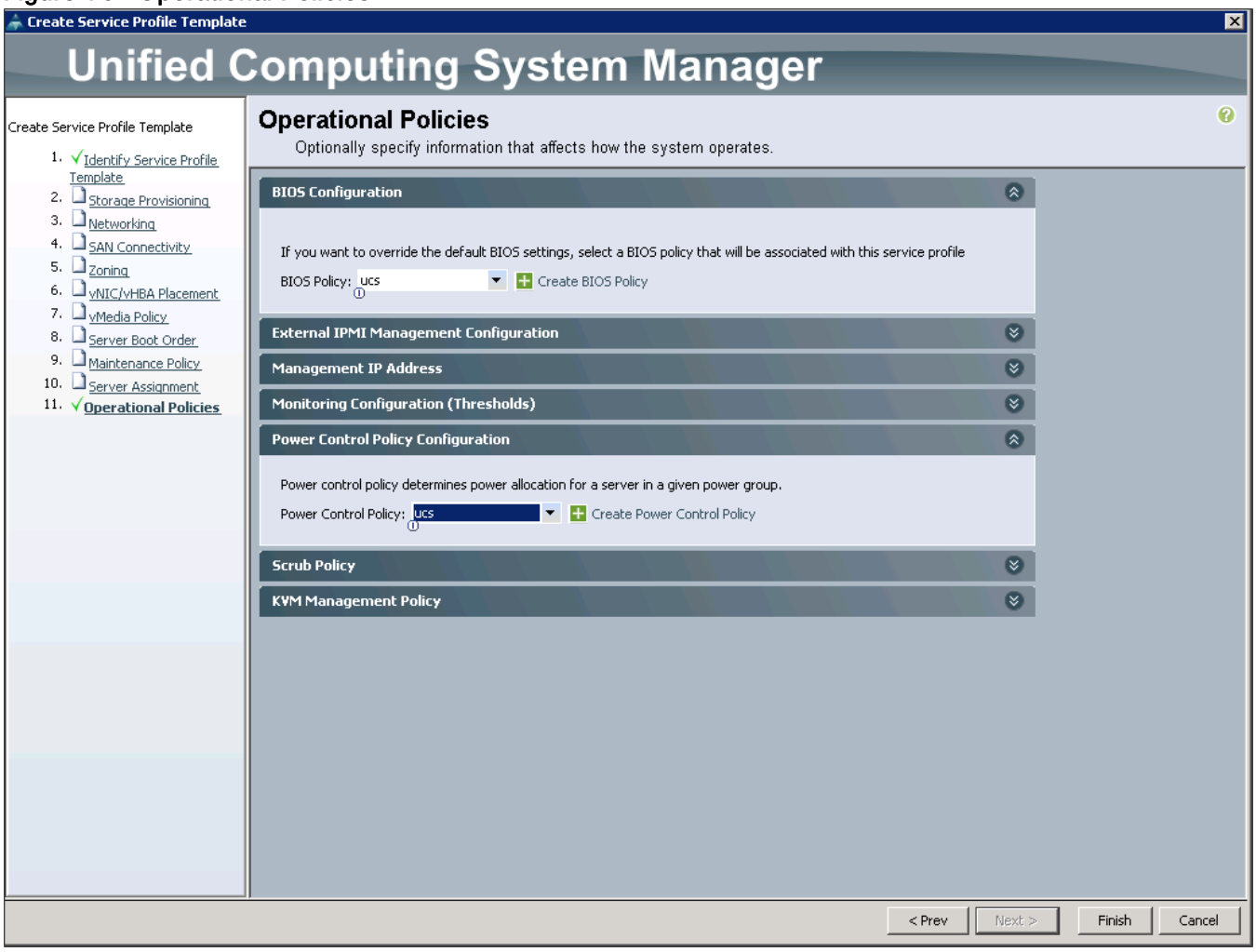


Configuring Operational Policies for the Template

In the Operational Policies Window (Figure 70), complete the following steps:

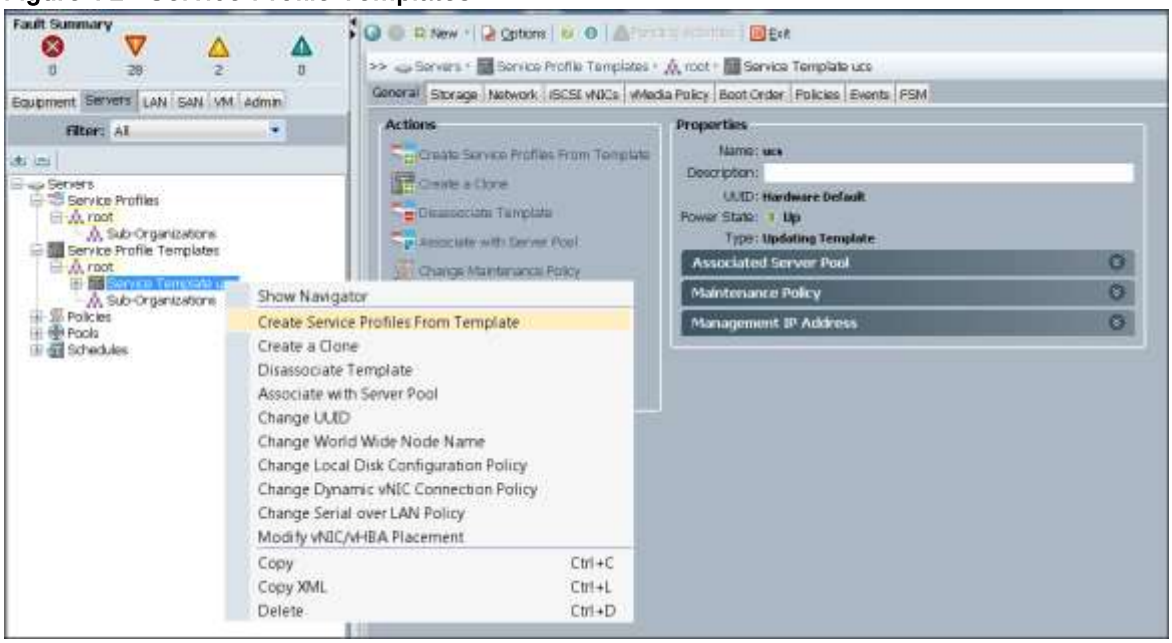
1. Select ucs in the BIOS Policy field.
2. Select ucs in the Power Control Policy field.

Figure 70 Operational Policies



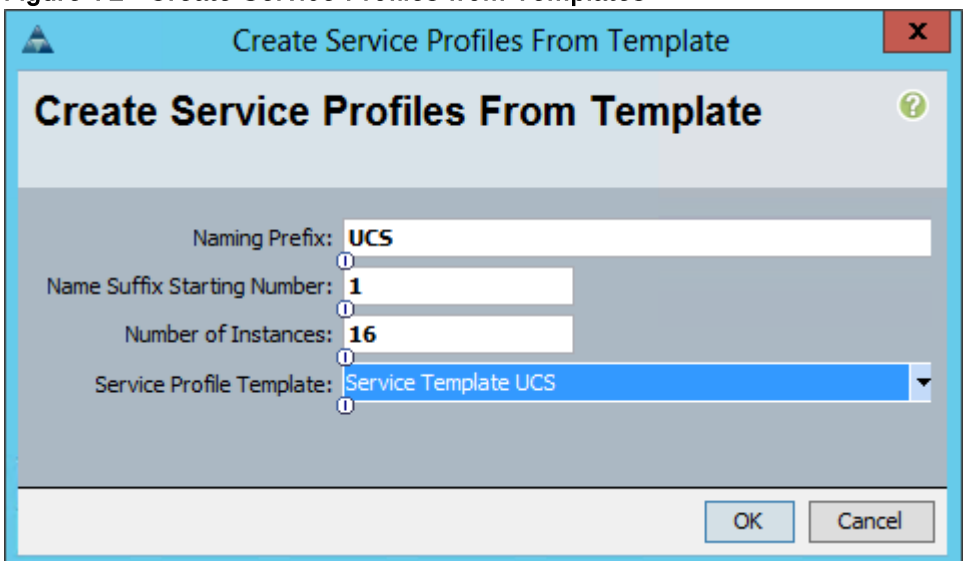
3. Click `Finish` to create the Service Profile template.
4. Click `OK` in the pop-up window to proceed.
5. Select the `Servers` tab in the left pane of the UCS Manager GUI. (Figure 71)
6. Go to Service Profile Templates > root.
7. Right-click Service Profile Templates ucs.
8. Select Create Service Profiles From Template.

Figure 71 Service Profile Templates



The Create Service Profiles from Template window appears. (Figure 72)

Figure 72 Create Service Profiles from Templates

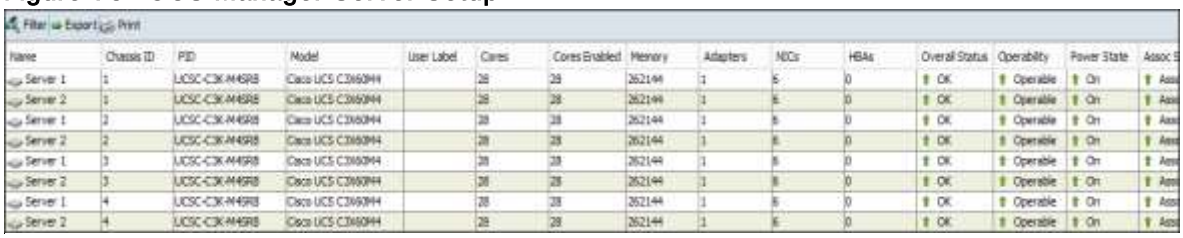


9. Click OK.

Association of the Service Profiles will take place automatically.

The final Cisco UCS Manager window is shown in Figure 73 below.

Figure 73 UCS Manager Server Setup



Creating Service Profile Templates for Hadoop Management Nodes

Creating an Organization

Organizations are used as a means to arrange and restrict access to various groups within the IT organization, thereby enabling multi-tenancy of the compute resources. This document does not assume the use of Organizations; however the necessary steps are provided for future reference.

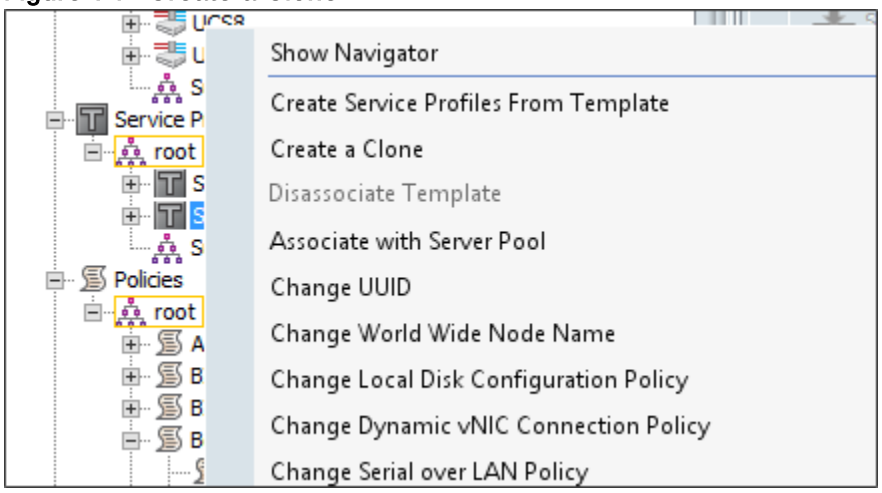
To configure an organization within the Cisco UCS Manager GUI, complete the following steps:

1. Click on Servers tab, go to Service Profile Template → root.
2. Right click on root and select Create Organization from the options.
3. Enter UCS-C240 as the name for the organization.
4. Click Ok.

Cloning the Template for Hadoop Management Nodes

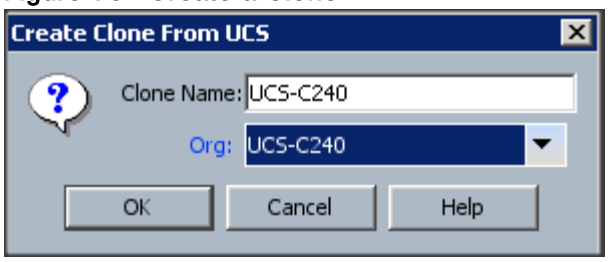
5. Click on Servers tab, go to Service Profile Template → root.
6. Right click on the existing template UCS and click Create a Clone. (Figure 74)

Figure 74 Create a Clone



7. In the Clone Name, enter UCS-C240 and from the Org drop down list choose UCS-C240 and click OK. (Figure 75)

Figure 75 Create a Clone

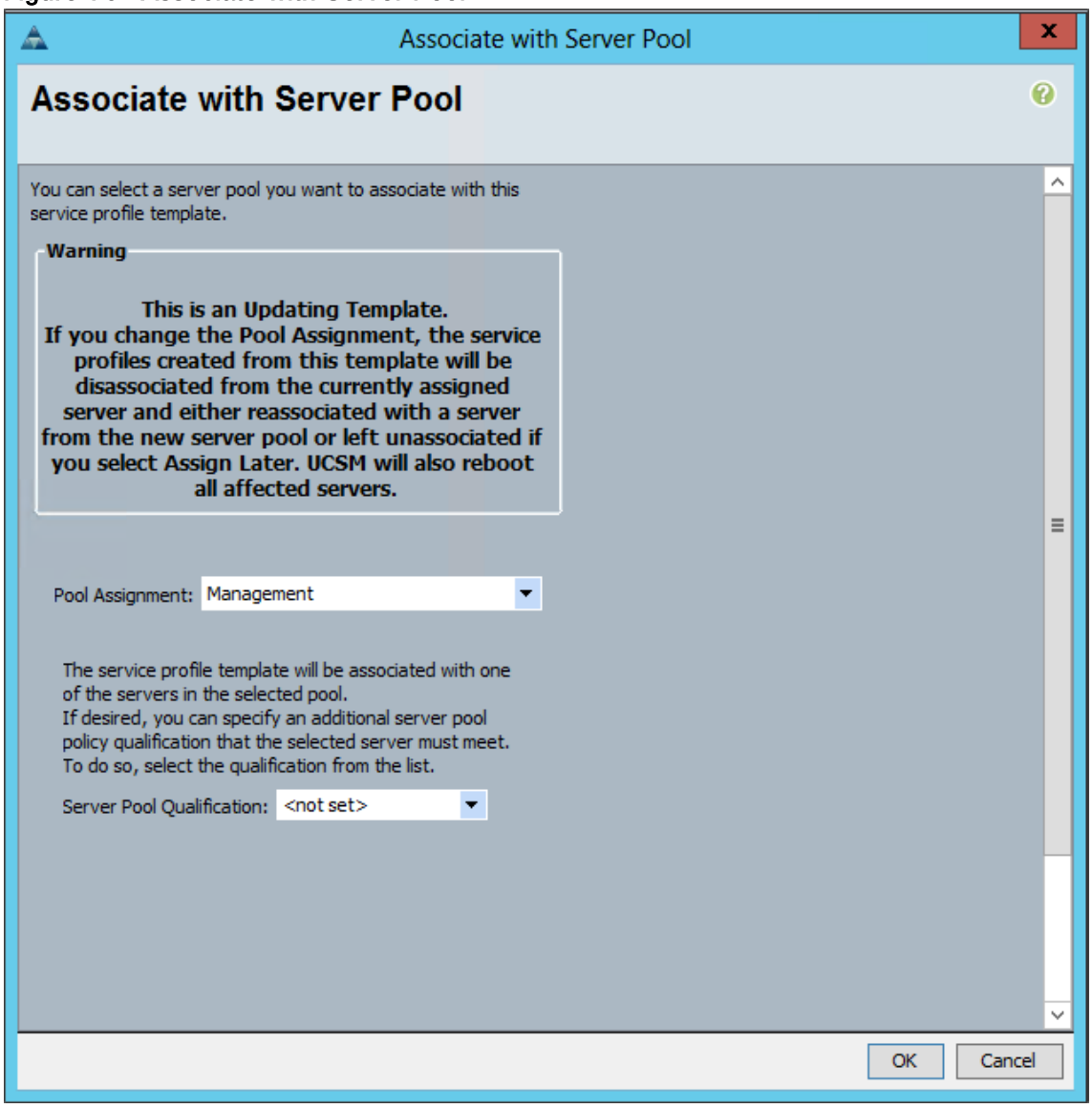


8. Go to root → Sub-Organization → UCS-C240 and select the Service Template UCS-C240.

9. In the right window general tab click Associate with Server pool. (Figure 76)

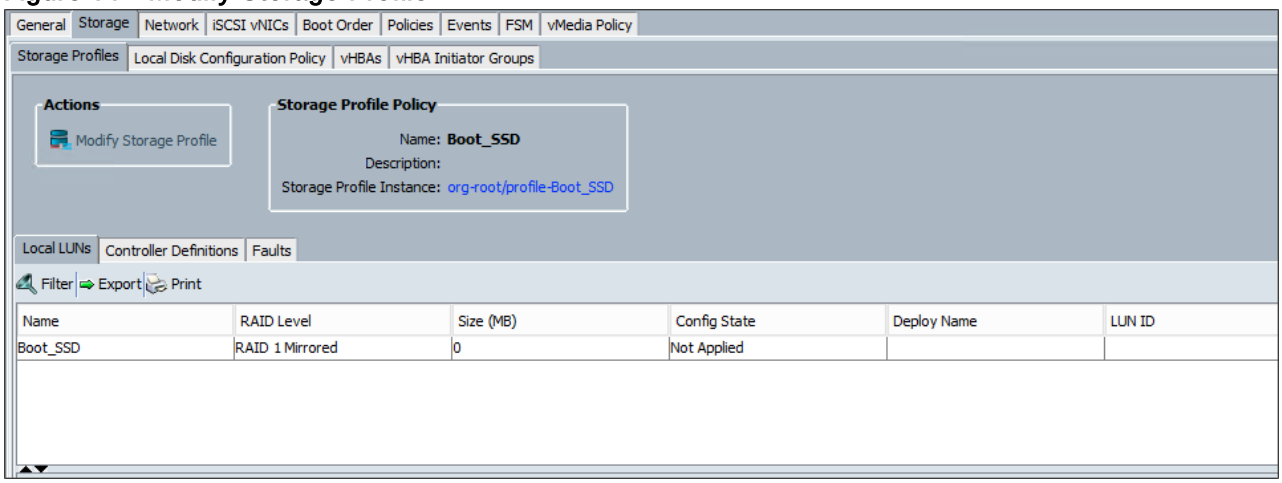
10. In the Pool Assignment drop down list choose Management and click OK.

Figure 76 Associate with Server Pool



11. In the right window select the Storage tab and click Modify Storage Profile. (Figure 77)

Figure 77 Modify Storage Profile



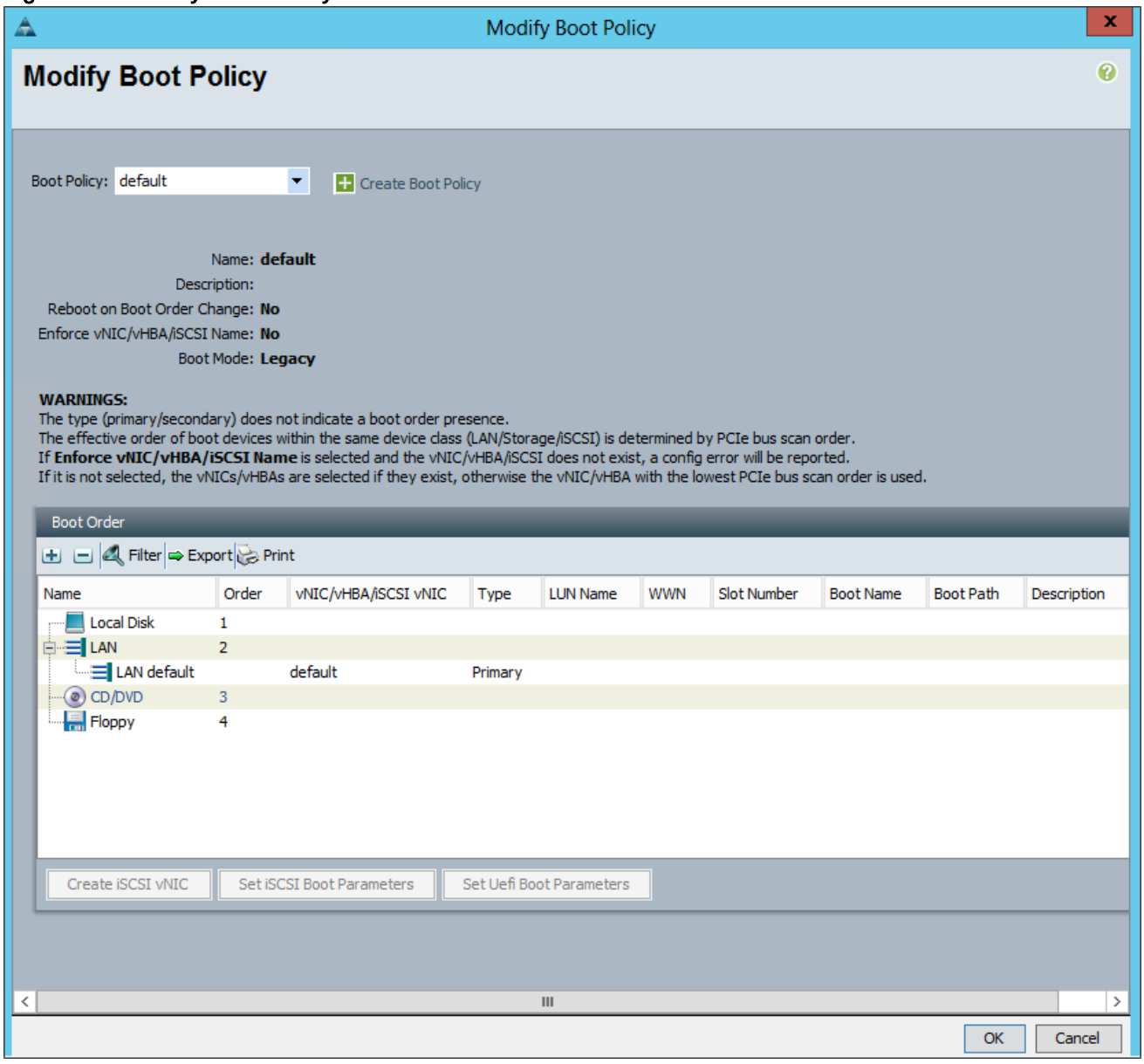
12. From the Storage profile drop down list choose No Storage Profile and click OK. (Figure 78)

Figure 78 No Storage Profile

13. Select the Boot Order tab and click Modify Boot Policy.

14. From the Boot Policy drop down list choose Default and click OK. (Figure 79)

Figure 79 Modify Boot Policy

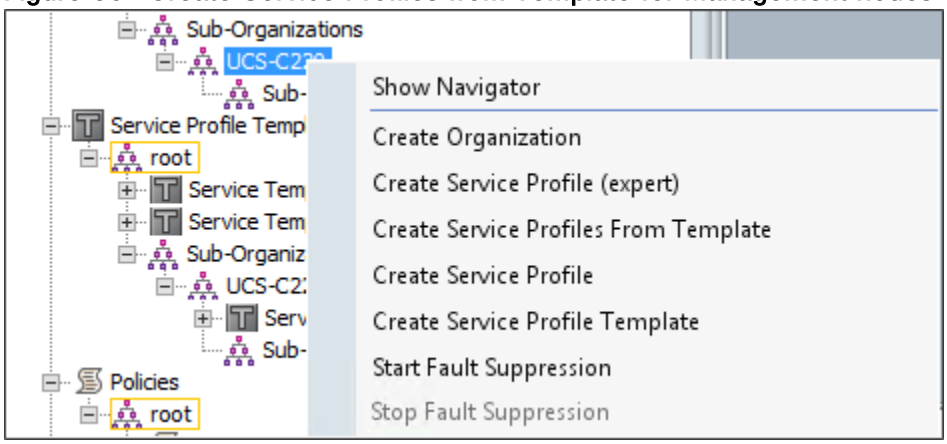


Creating Service Profile from Template

1. Go to Servers → Service Profiles → root → Sub-Organization → UCS-C240.

2. Right click and select Create Service Profiles from Template. (Figure 80)

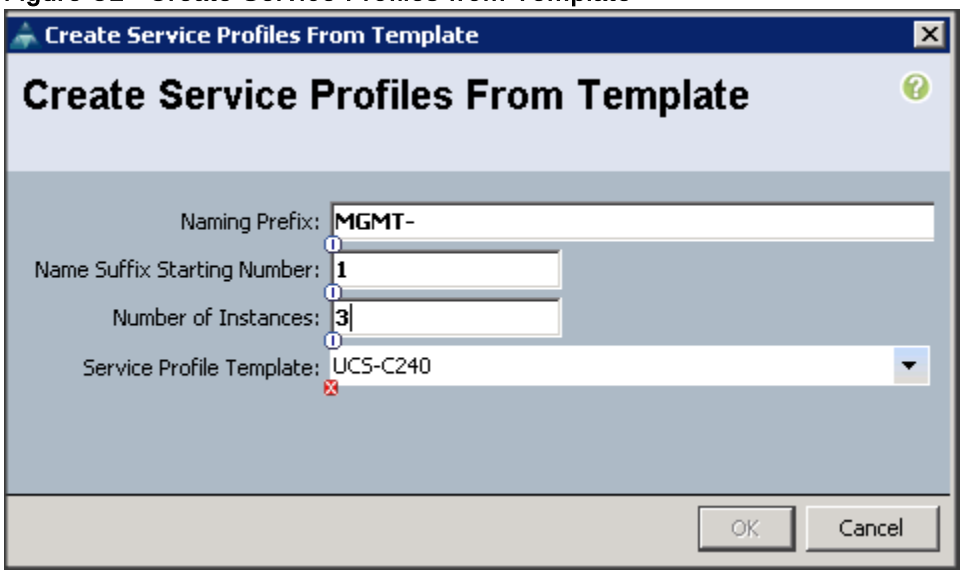
Figure 80 Create Service Profiles from Template for Management nodes



In the Create Service Profiles from Template screen: (Figure 81)

- Naming Prefix enter MGMT-
- Name Suffix Starting Number 1
- Number of Instances 3
- Service Profile Template UCS-C240 and click OK.

Figure 81 Create Service Profiles from Template



The service profile will be applied to the three Management UCS-C240 M4 Rack Server nodes.

Installing Red Hat Enterprise Linux 7.2

The following section provides detailed procedures for installing Red Hat Enterprise Linux 7.2 using Software RAID (OS based Mirroring) on Cisco UCS C240 M4 Rack Servers. There are multiple ways to install the Red Hat Linux operating system. The installation procedure described in this deployment guide uses KVM console and virtual media from Cisco UCS Manager.

Note: This requires RHEL 7.2 DVD/ISO for the installation.

Installing Red Hat Enterprise Linux 7.2 on Management Nodes

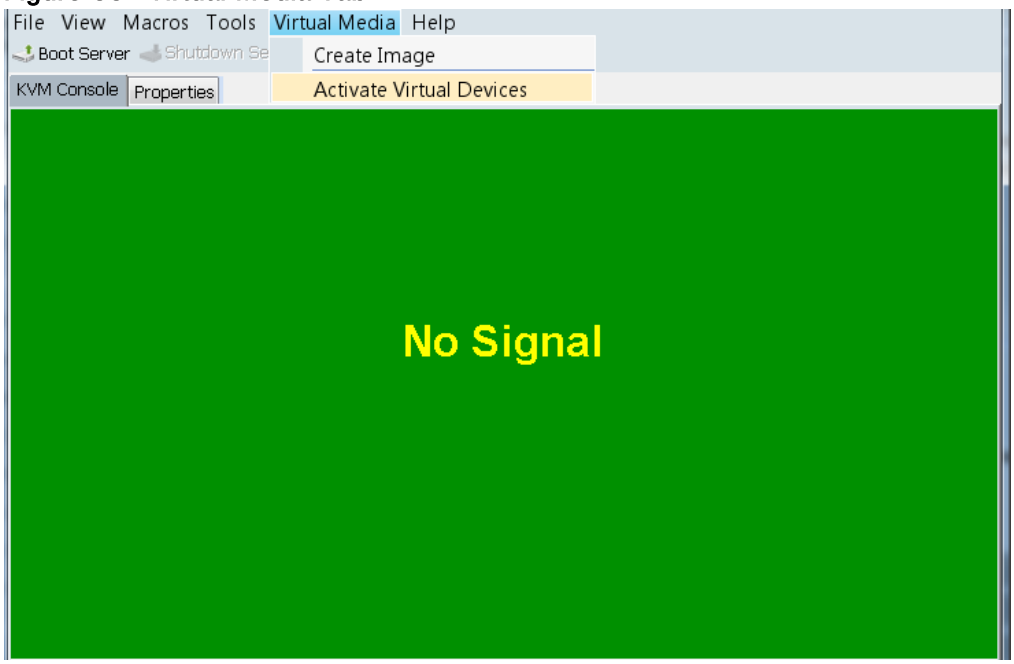
To install the Red Hat Linux 7.2 operating system, complete the following steps:

1. Log in to the Cisco UCS 6332 Fabric Interconnect and launch the Cisco UCS Manager application.
2. Select the `Equipment` tab as shown in Figure 82
3. In the navigation pane expand `Rack-Mounts` and then `Servers`
4. Right click on the server and select `KVM Console`
5. In the KVM window, select the `Virtual Media` tab.

Figure 82 KVM Console

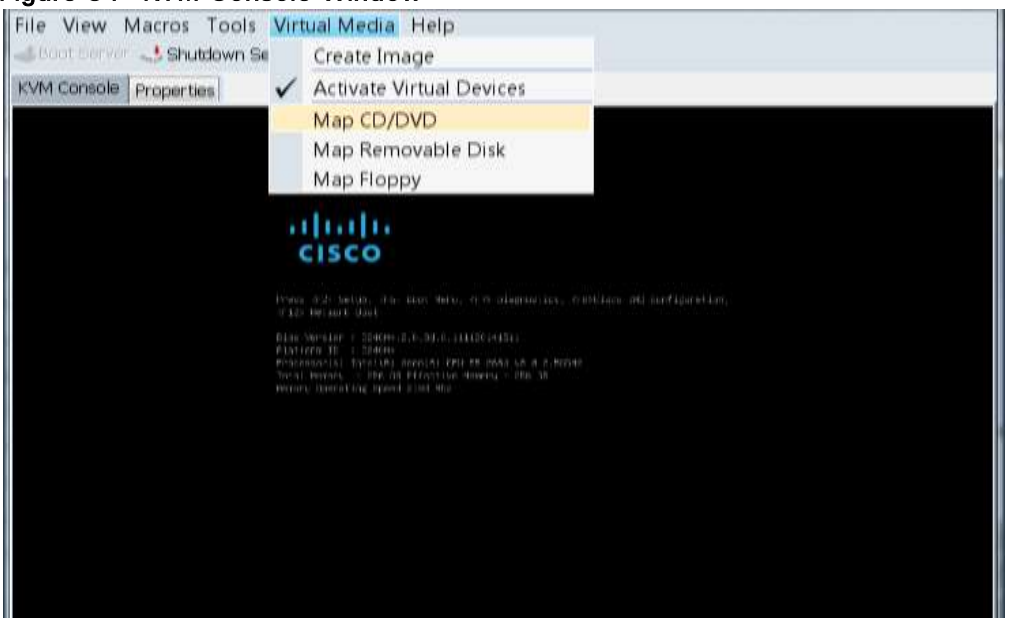
6. Click the `Activate Virtual Devices` found in the Virtual Media tab. (Figure 83)

Figure 83 Virtual Media Tab



7. In the KVM window (Figure 84), select the Virtual Media tab and click the `Map CD/DVD`

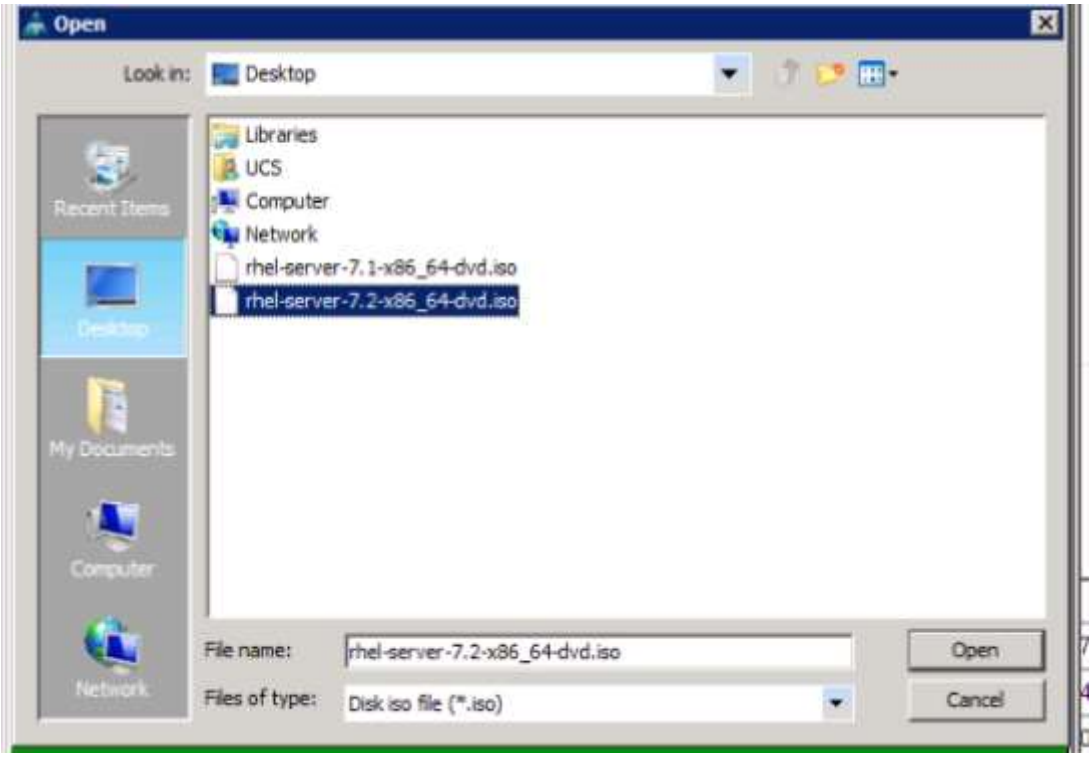
Figure 84 KVM Console Window



8. Browse to the Red Hat Enterprise Linux Server 7.2 installer ISO image file.

Note: The Red Hat Enterprise Linux 7.2 DVD is assumed to be on the client machine.

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



10. In the KVM window, select the KVM tab to monitor during boot.

11. In the KVM window, select the `Macros > Static Macros > Ctrl-Alt-Del` button in the upper left corner.

12. Click OK

13. Click OK to reboot the system.

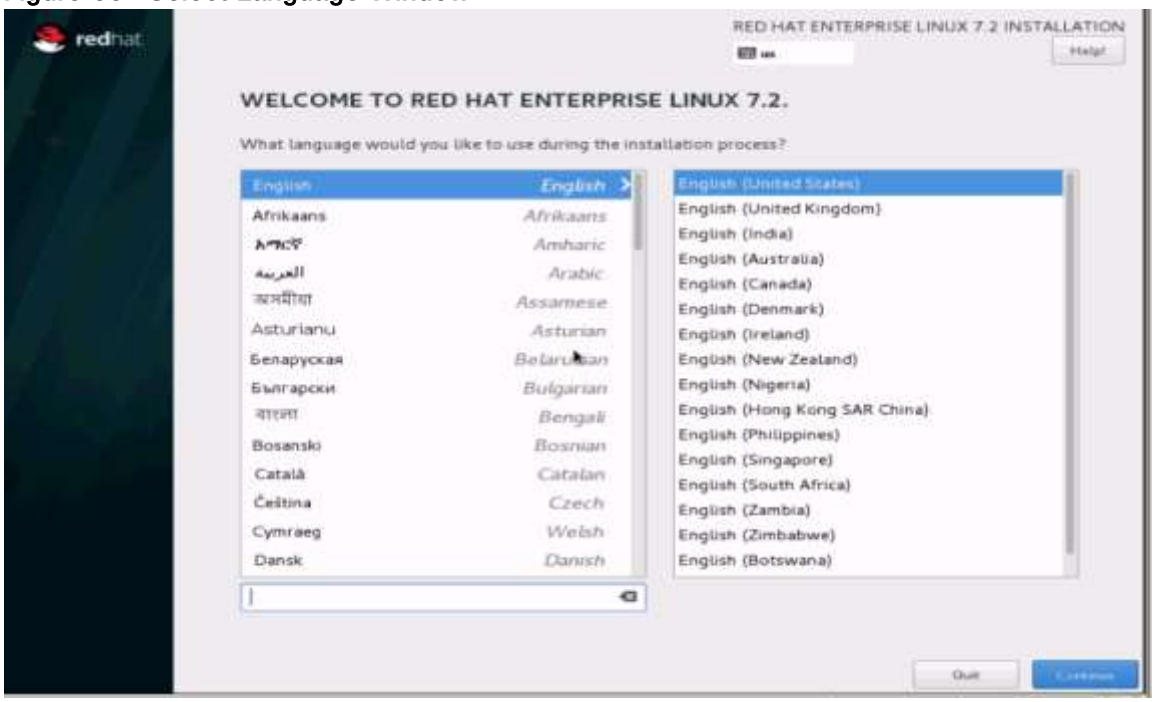
14. On reboot, the machine detects the presence of the Red Hat Enterprise Linux Server 7.2 install media.

15. Select the Install or Upgrade an Existing System.

16. Skip the Media test and start the installation.

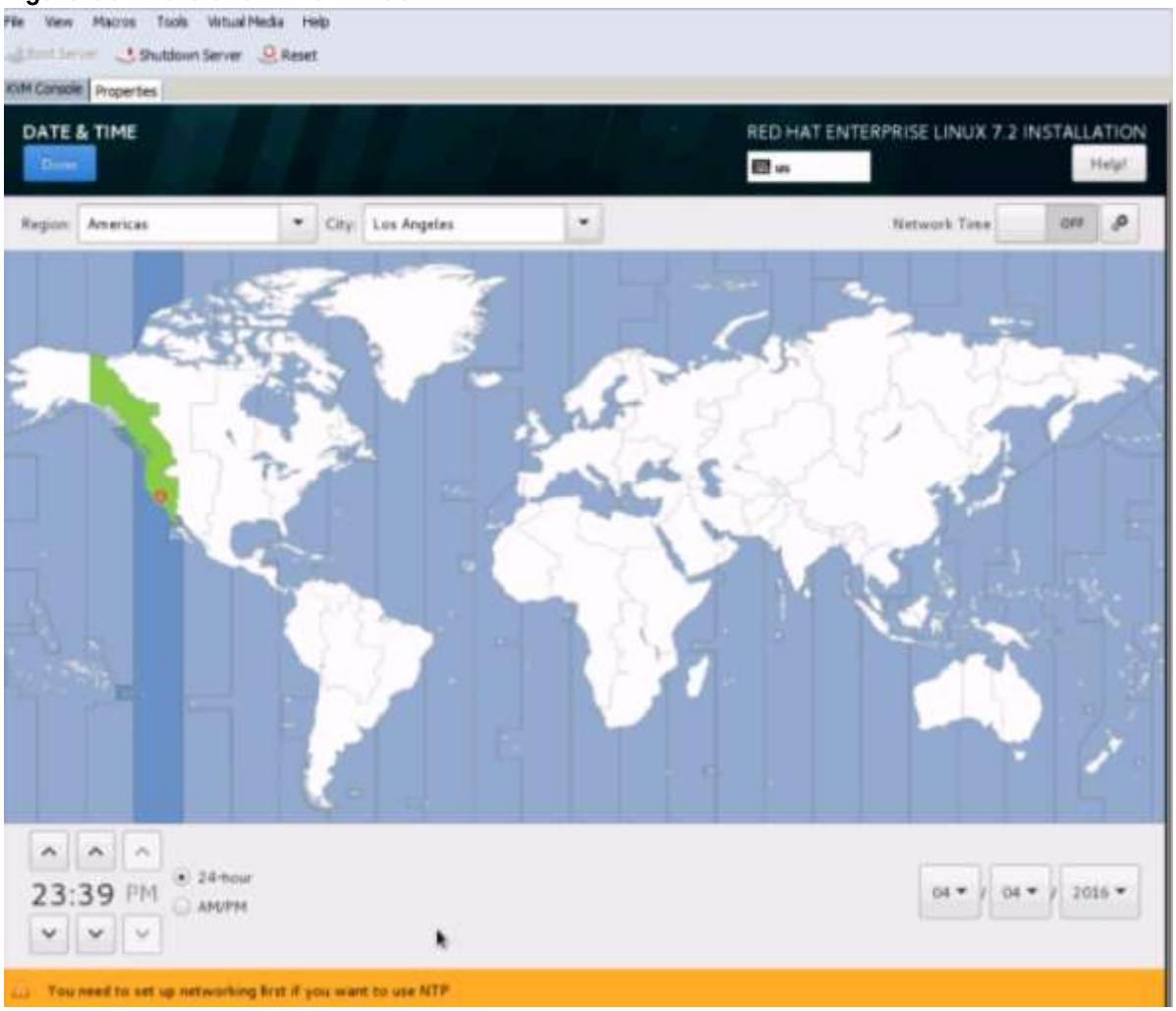
17. Select language of installation (Figure 85), and click Continue.

Figure 85 Select Language Window



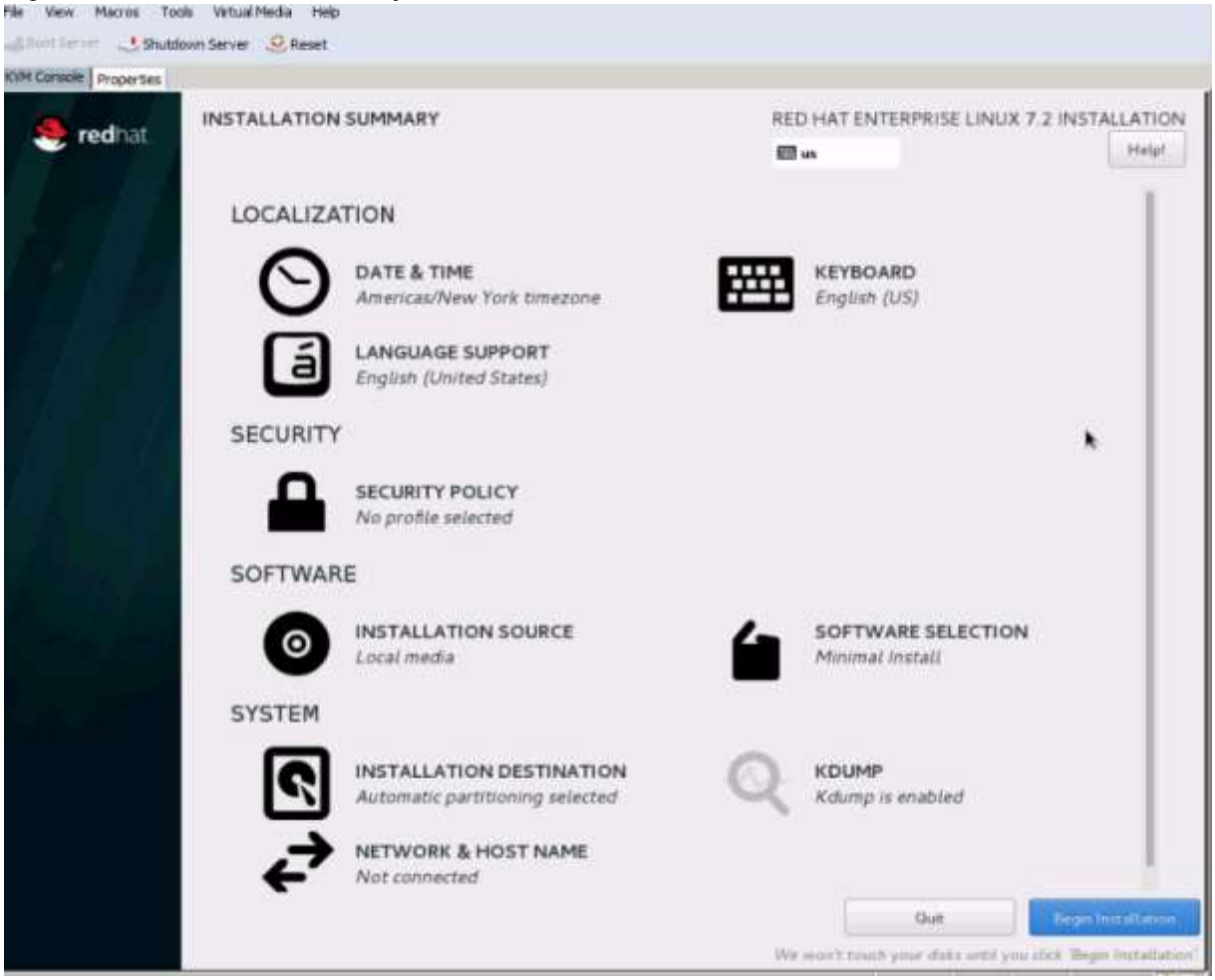
18. Select Date and time as shown in Figure 86

Figure 86 Date and Time Window



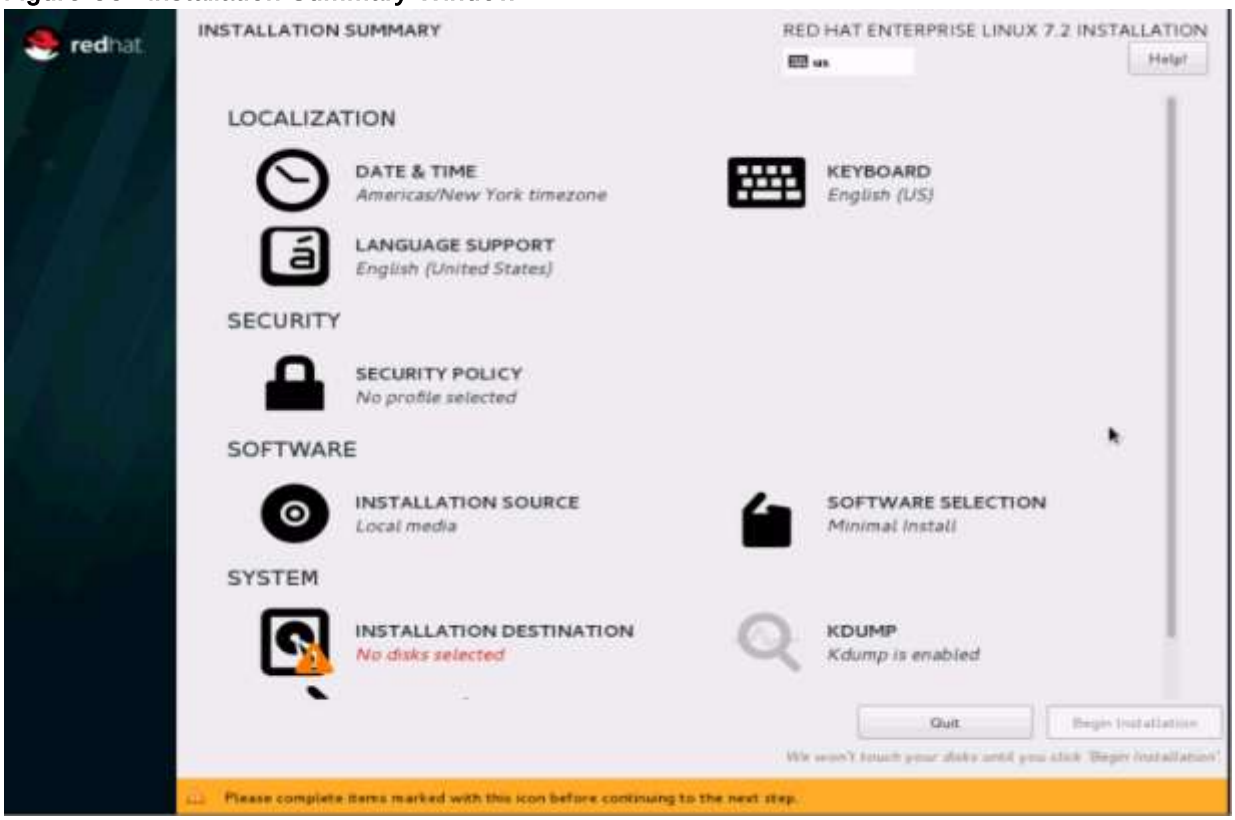
19. Select the location on the map, set the time and click Done

Figure 87 Installation Summary Window



20. Click on Installation Destination, shown above in Figure 87

Figure 88 Installation Summary Window

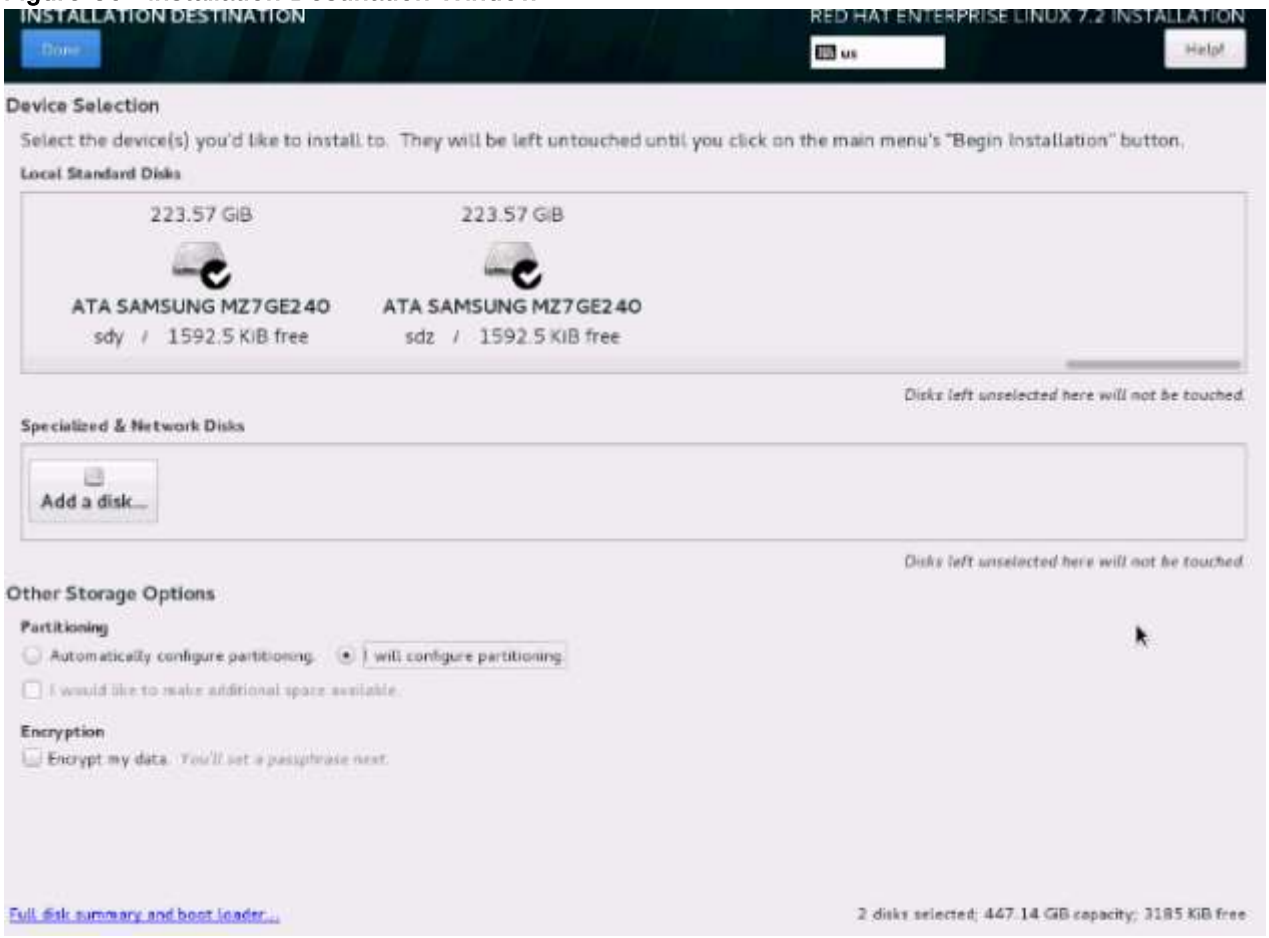


A Caution symbol appears next to Installation Destination as shown in Figure 88 above.

21. This opens the Installation Destination window displaying the boot disks. This is shown in Figure 89 below.

22. Make the selection, and choose "I will configure partitioning." Click Done.

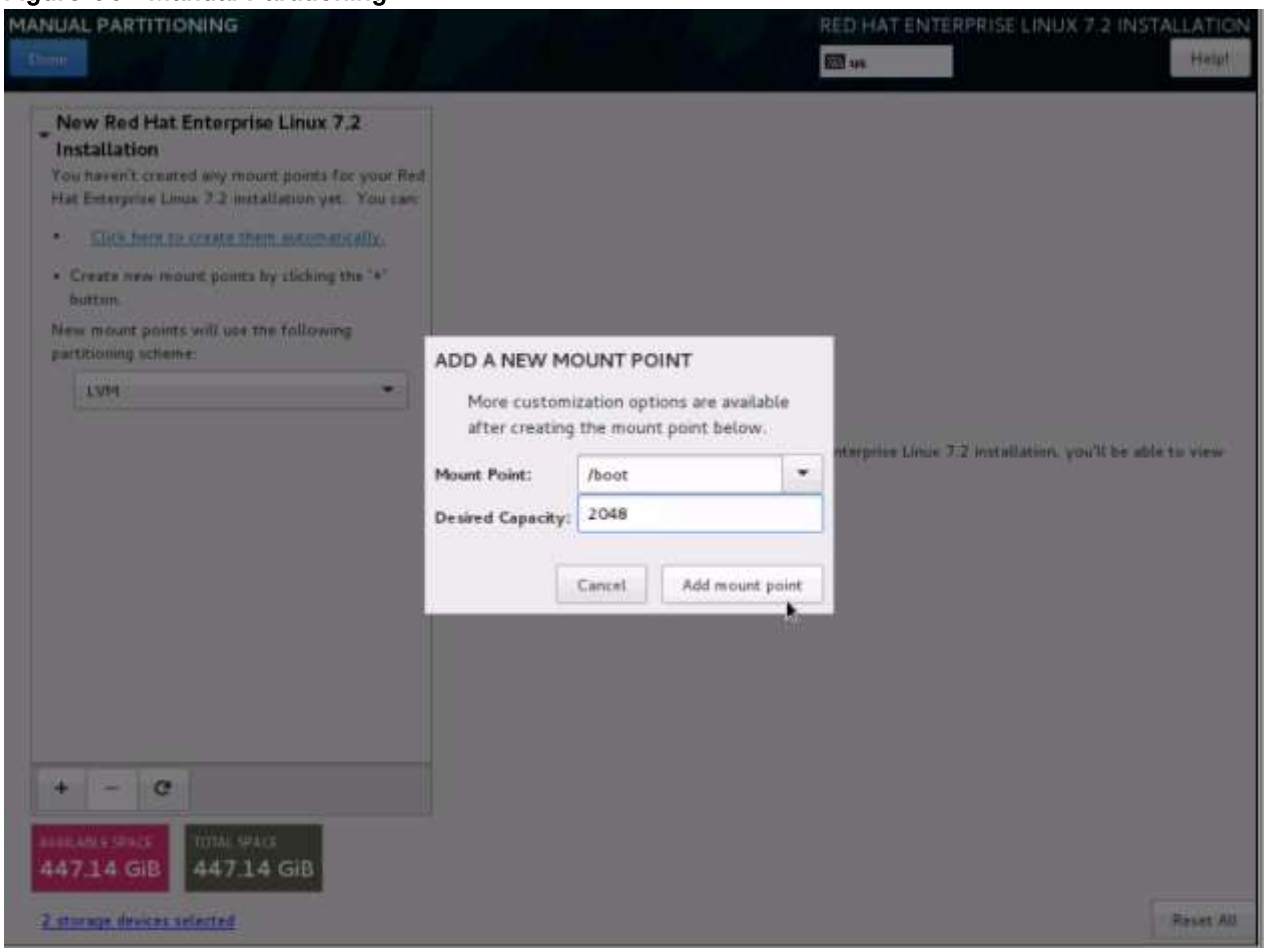
Figure 89 Installation Destination Window



This opens the new window for creating the partitions, as shown in Figure 90.

23. Click on the + sign to add a new partition as shown below, boot partition of size 2048 MB.

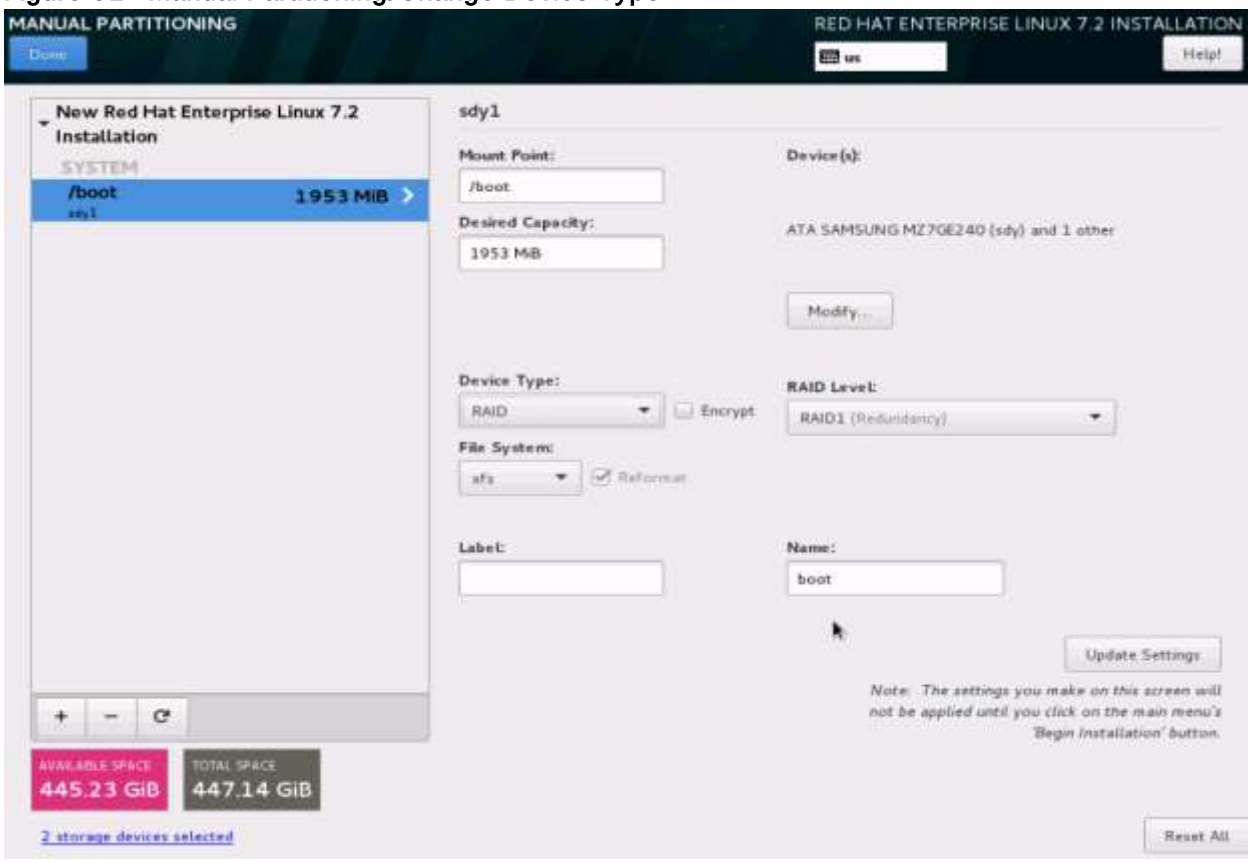
Figure 90 Manual Partitioning



24. Click Add Mount Point to add the partition.

The screen refreshes to show the added Mount Point (Figure 91).

Figure 91 Manual Partitioning/Change Device Type

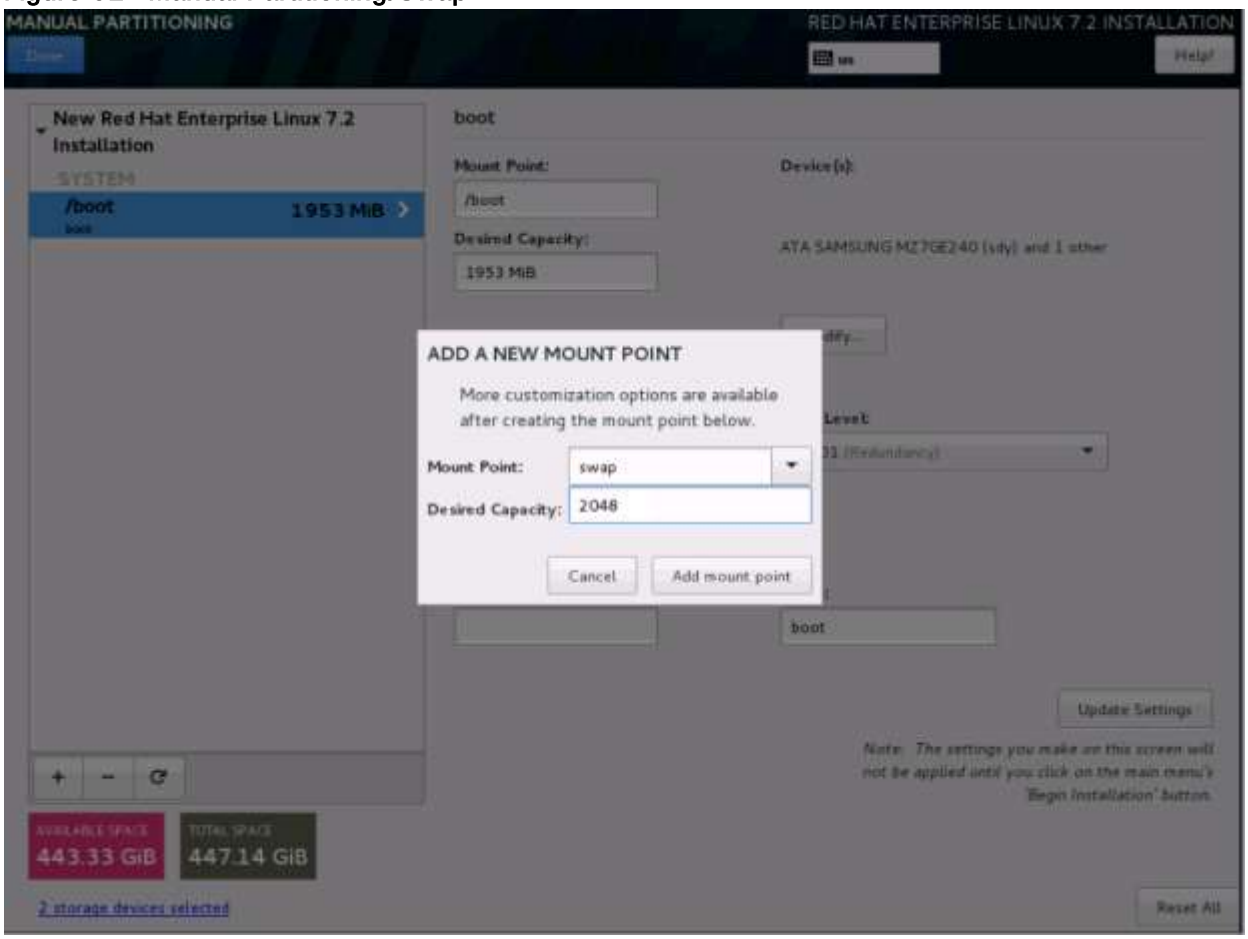


25. Change the Device type to RAID and make sure the RAID Level is RAID1 (Redundancy).

26. Click on Update Settings to save the changes.

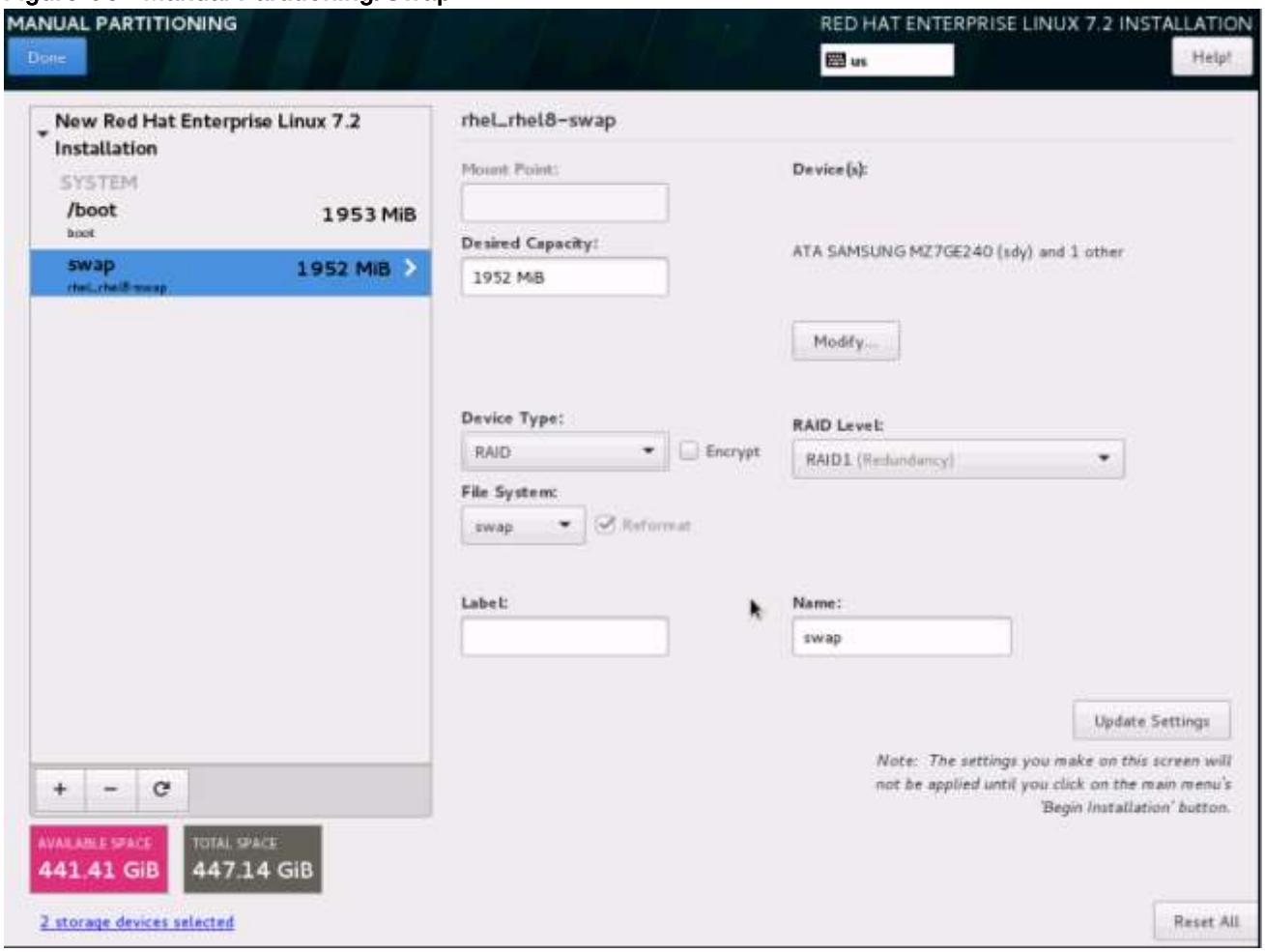
27. Click on the + sign to create the swap partition of size 2048 MB as shown in Figure 92 below.

Figure 92 Manual Partitioning/Swap



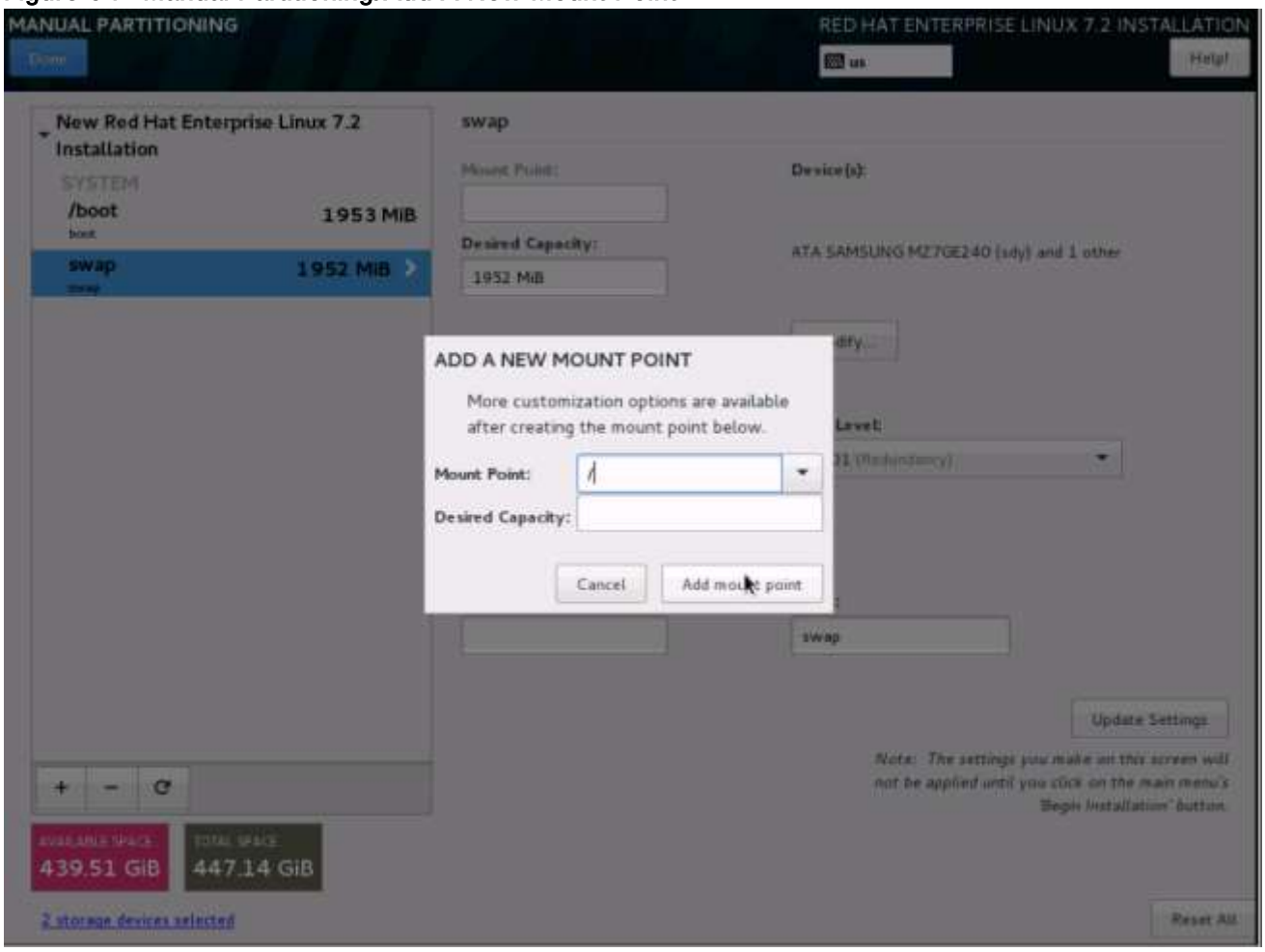
28. Change the Device type to RAID and RAID level to RAID1 (Redundancy) and click on Update Settings.

Figure 93 Manual Partitioning/Swap



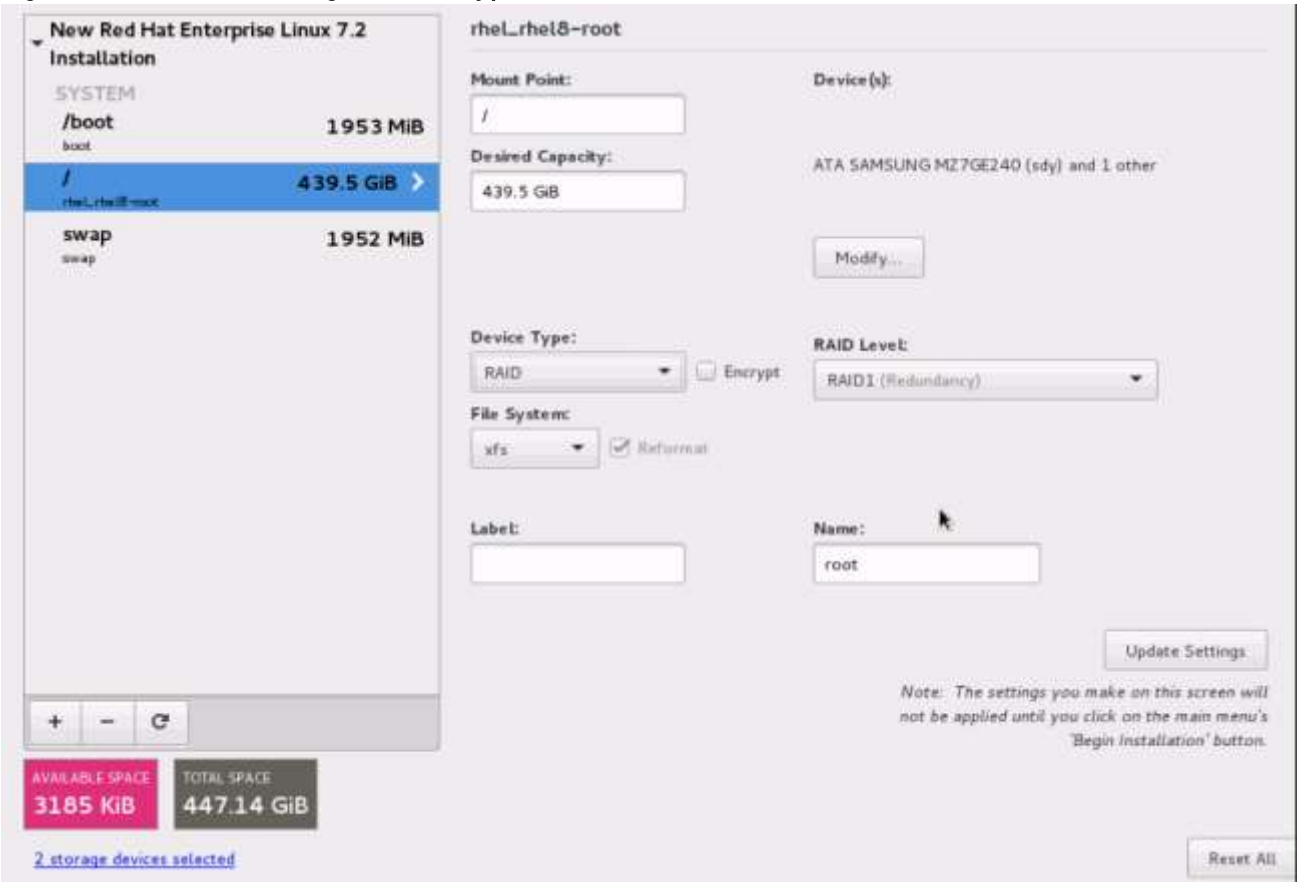
29. Click + to add the / partition. The size can be left empty so it uses the remaining capacity and click Add Mountpoint. (Figure 94).

Figure 94 Manual Partitioning/Add A New Mount Point



30. In the next window (Figure 95), change the Device type to RAID and RAID level to RAID1 (Redundancy). Click Update Settings.

Figure 95 Partition/Change Device Type to RAID

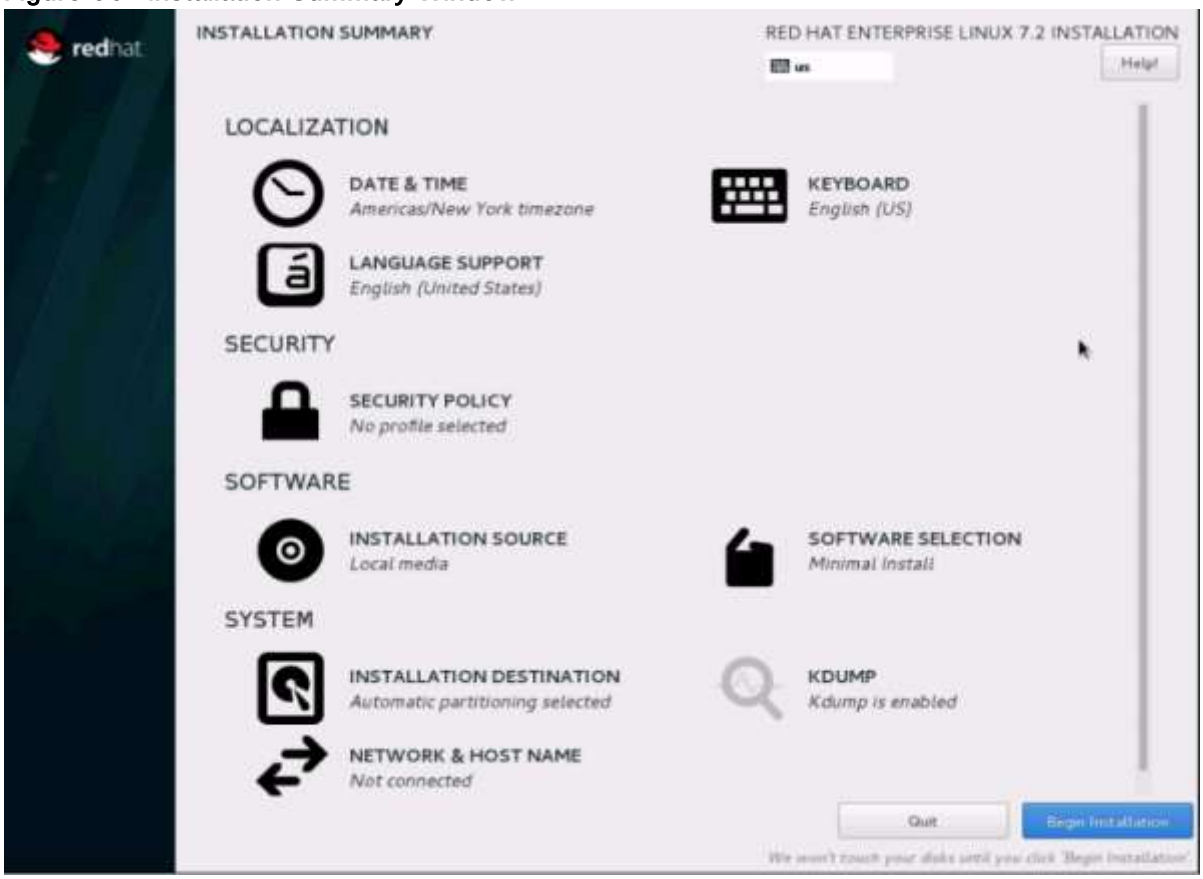


31. Click Done to go back to the main screen and continue the Installation.

The Installation screen opens (Figure 96).

32. Click on Software Selection.

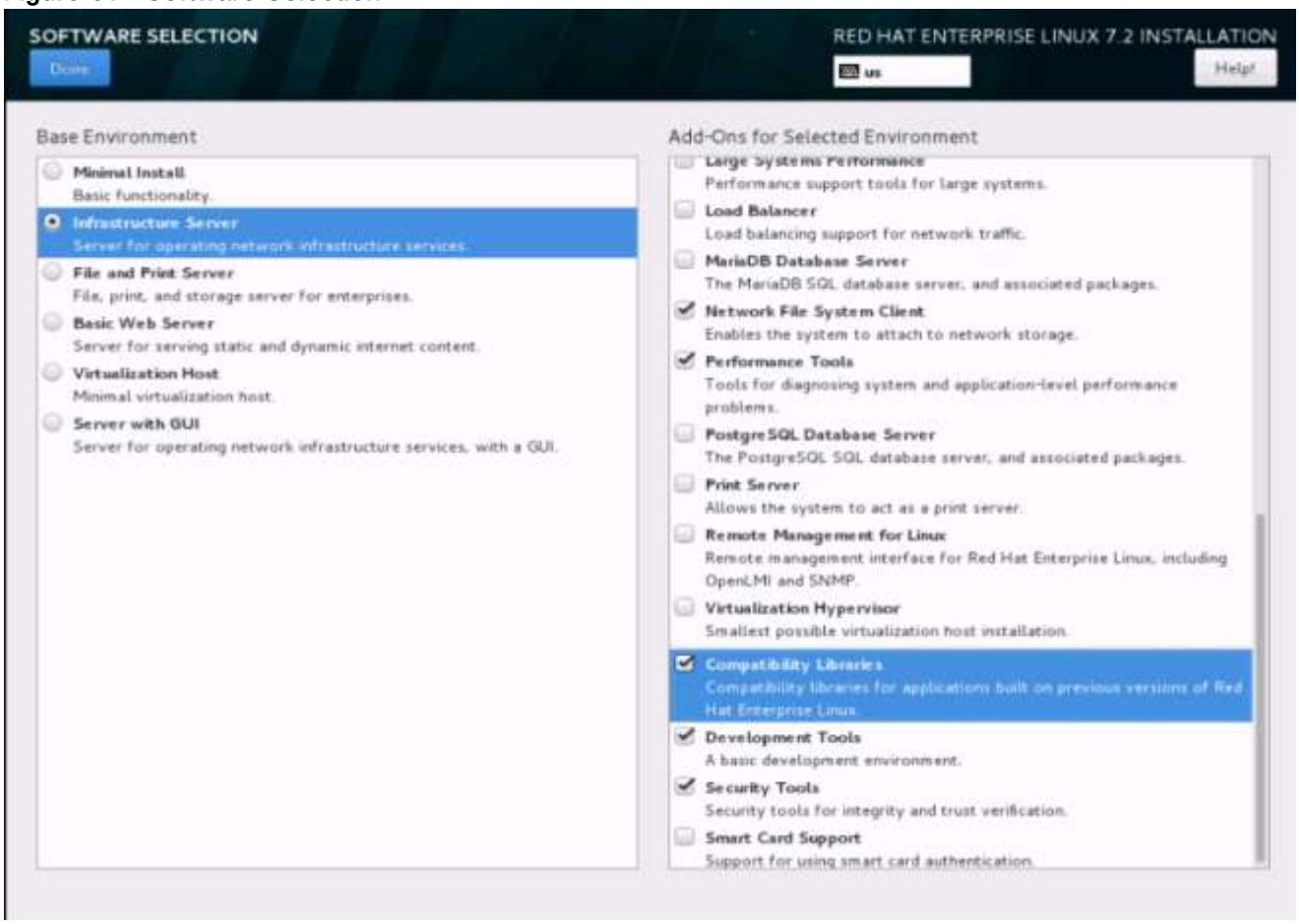
Figure 96 Installation Summary Window



The Software Selection screen opens (Figure 97).

33. Select Infrastructure Server and select the Add-Ons as noted below. Click Done.

Figure 97 Software Selection



The Installation Summary window returns (Figure 98).

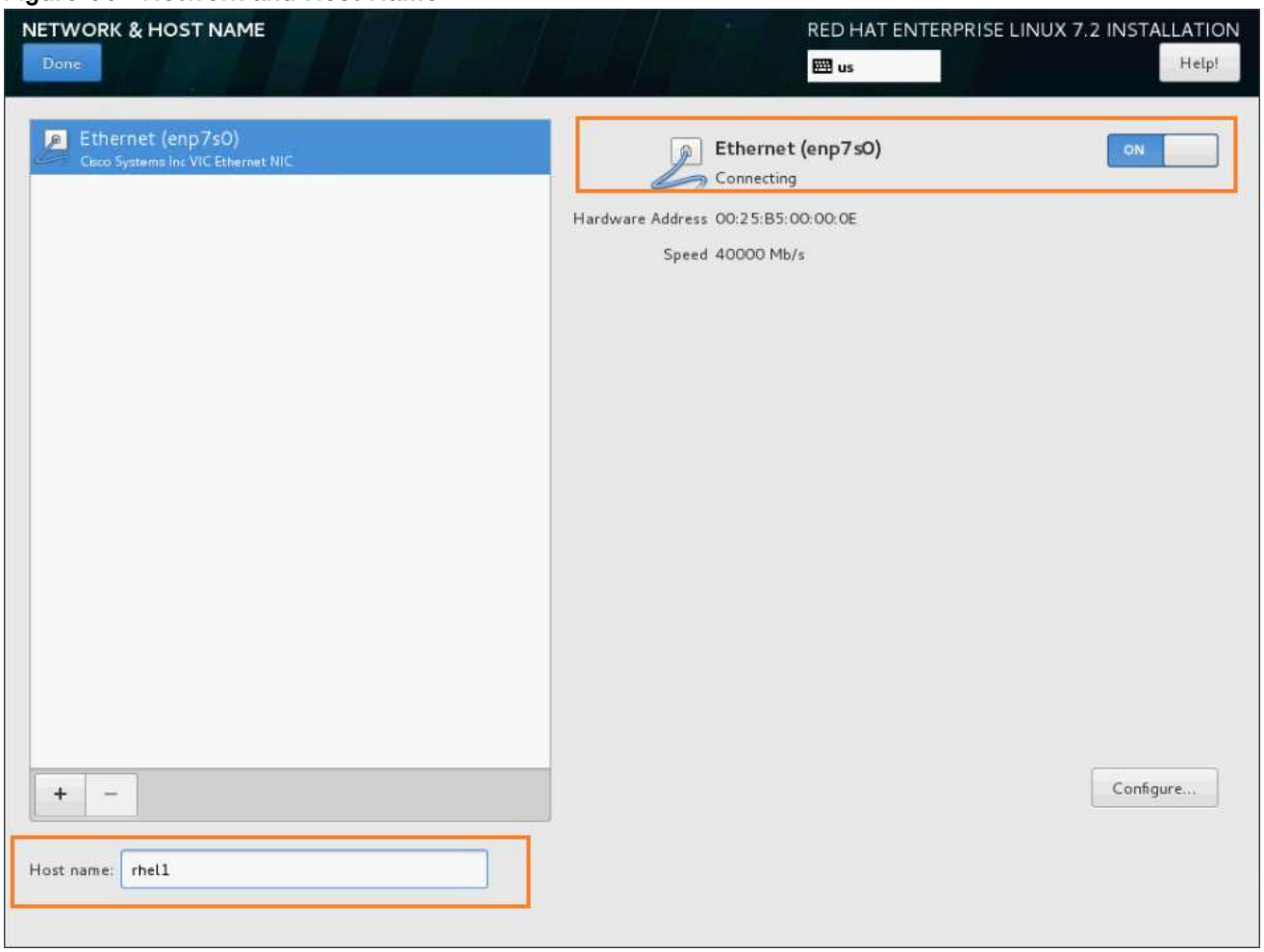
34. Click on Network and Hostname.

Figure 98 Network and Host Name Window

Configure Hostname and Networking for the Host.

35. Type in the hostname as shown below. (Figure 99)

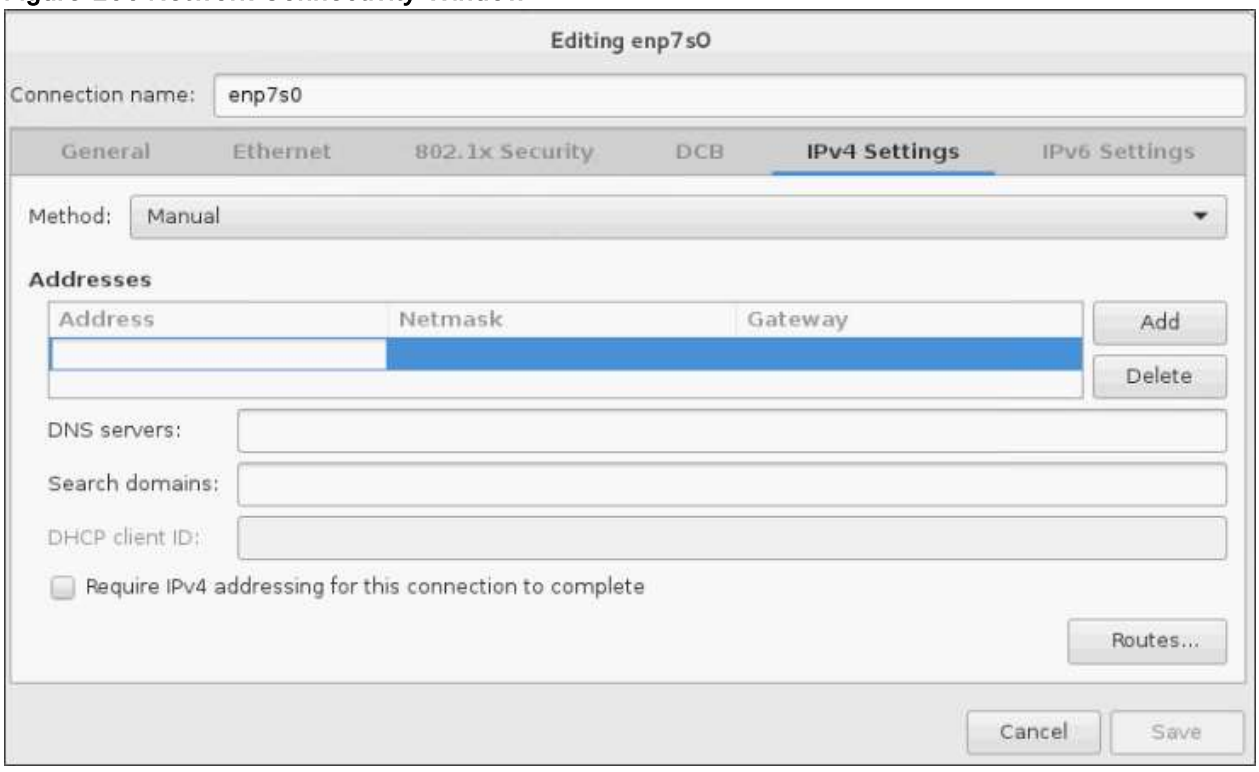
Figure 99 Network and Host Name



36. Click on Configure to open the Network Connectivity window (Figure 100).

37. Click on IPV4Settings.

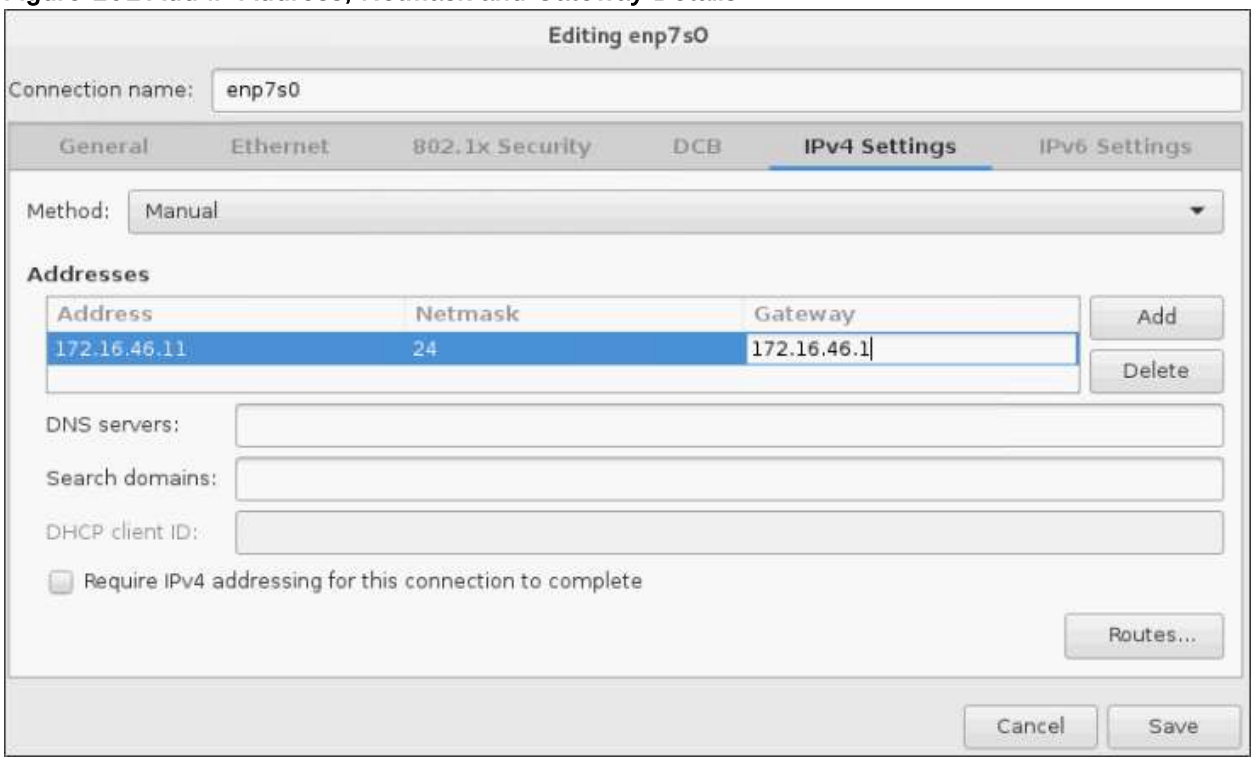
Figure 100 Network Connectivity Window



38. Change the Method to Manual and click Add. Figure 101 shows the Add Details pop up window.

39. Enter the IP Address, Netmask and Gateway details. Click Add after each addition.

Figure 101 Add IP Address, Netmask and Gateway Details



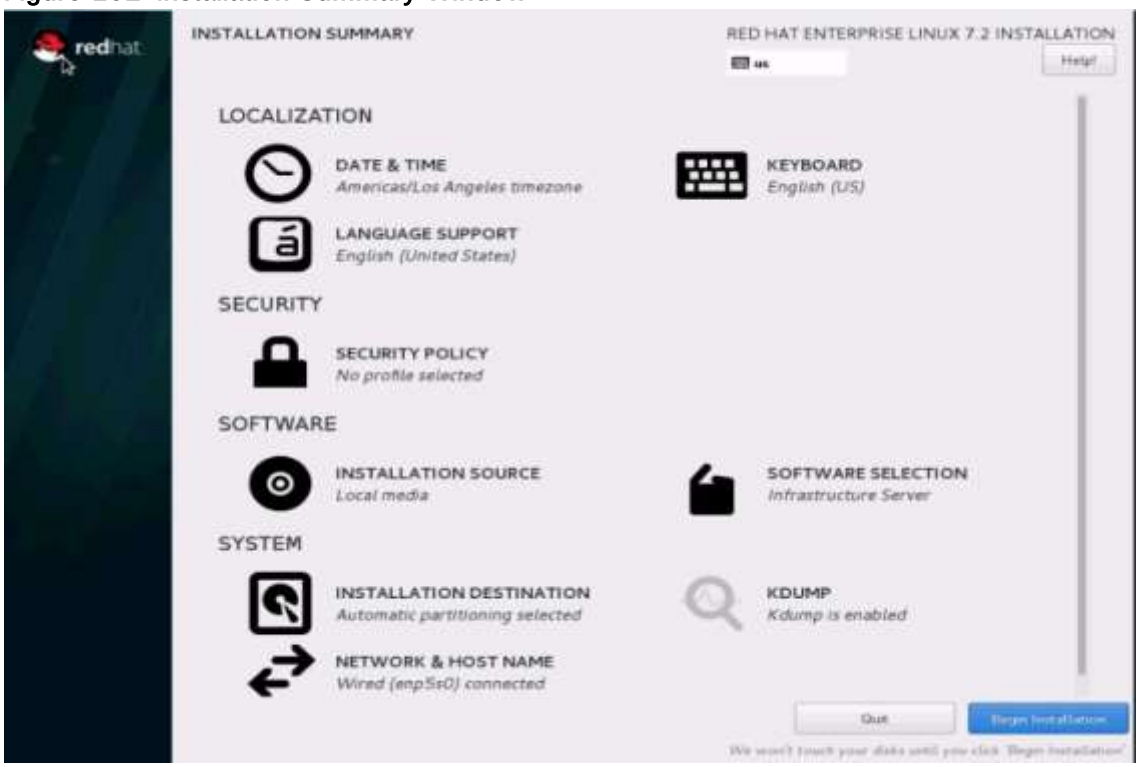
40. Click Save.

41. Update the hostname and turn Ethernet ON. Click Done to return to the main menu.

The Installation Summary window opens (Figure 102).

42. Click Begin Installation in the main menu.

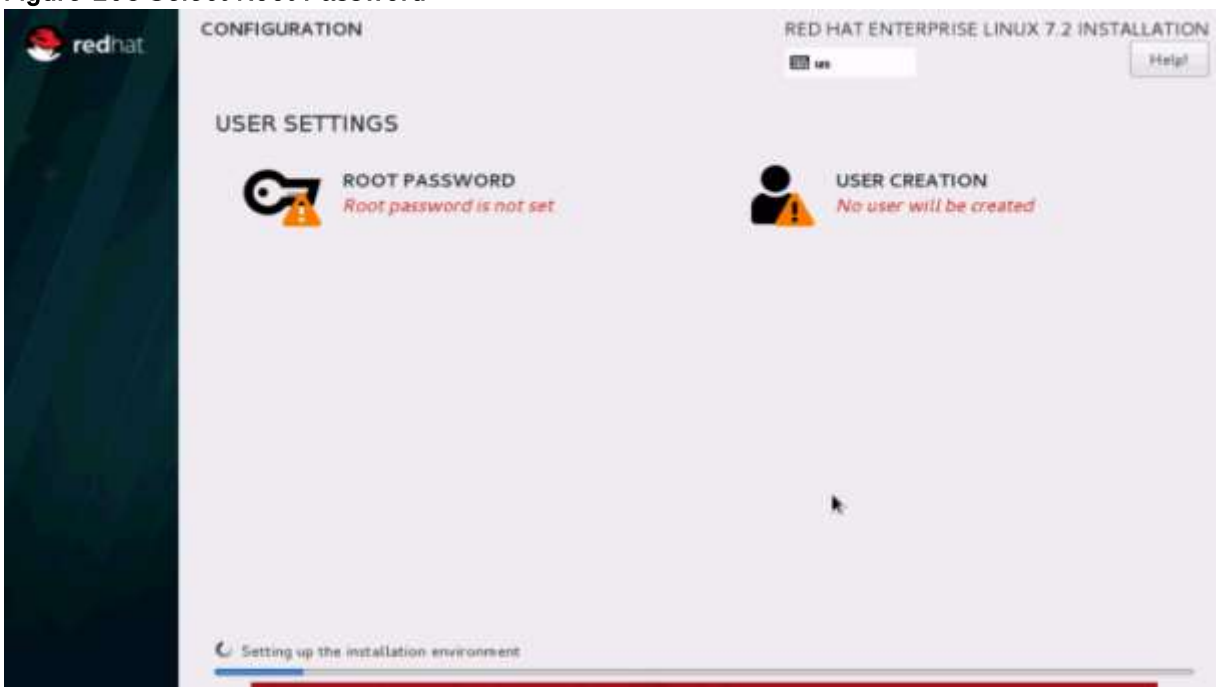
Figure 102 Installation Summary Window



A new window opens (Figure 103).

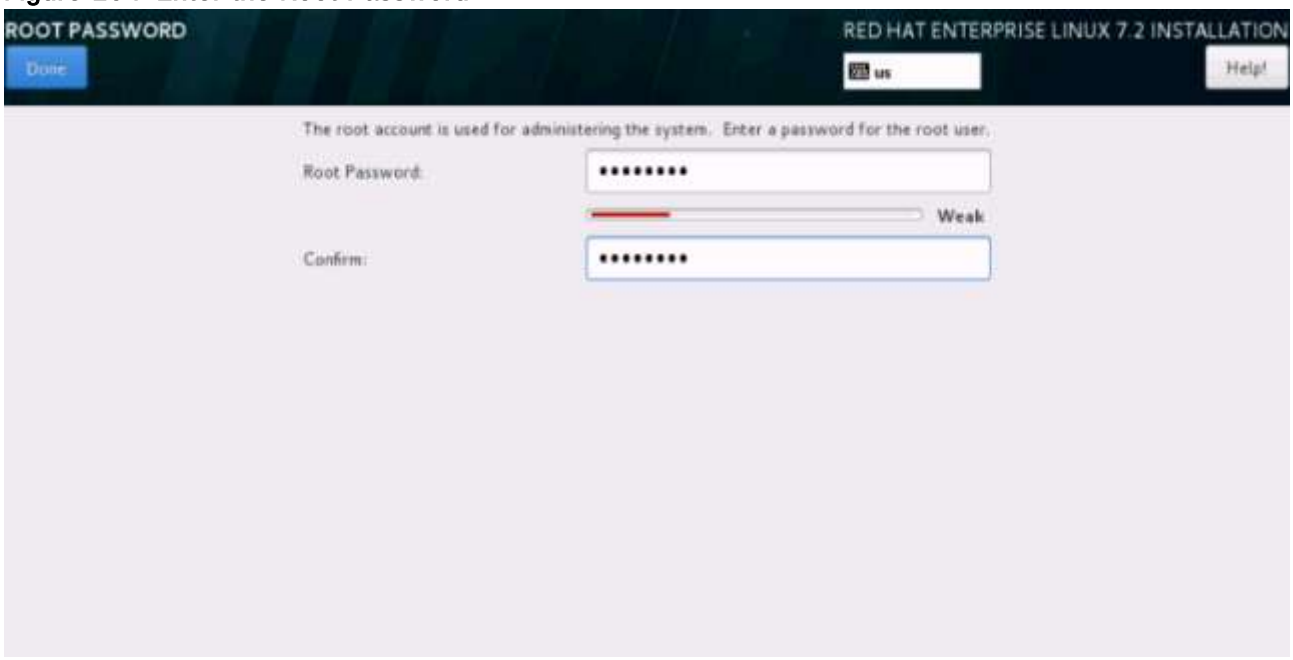
43. Select `Root Password` in the User Settings.

Figure 103 Select Root Password



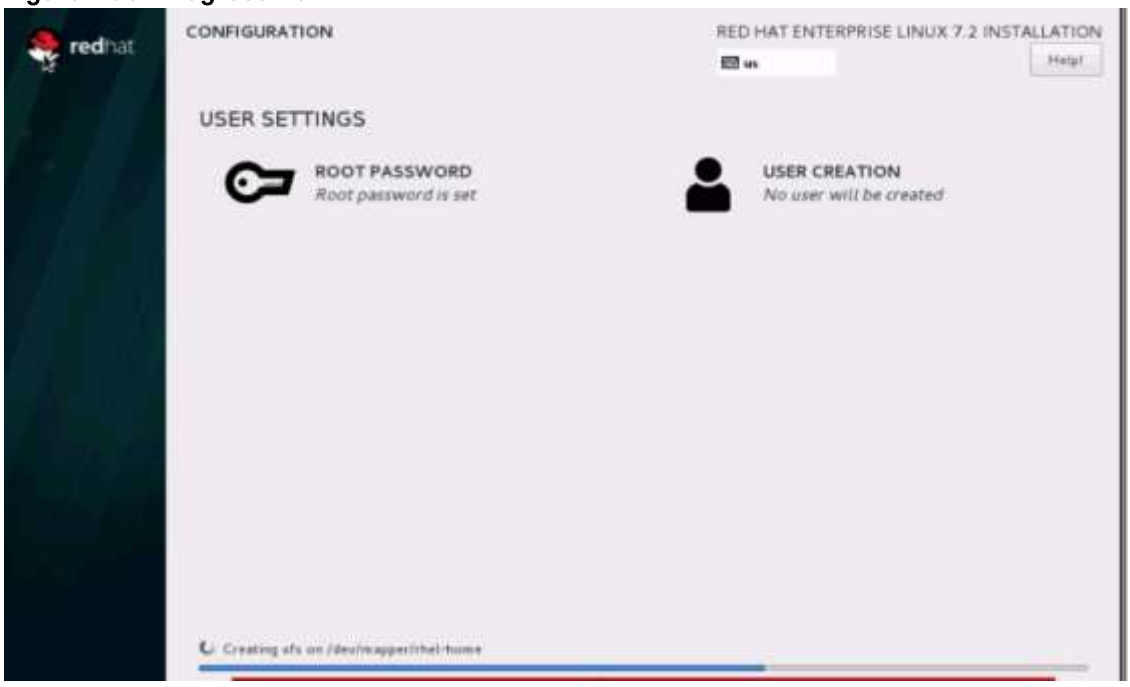
44. On the next screen (Figure 104), enter the `Root Password` and click Done

Figure 104 Enter the Root Password



A progress window will open (Figure 105).

Figure 105 Progress Bar



45. Once the installation is complete reboot the system.

46. Repeat steps 1 to 45 to install Red Hat Enterprise Linux 7.2 on other Management Nodes.

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

Installing Red Hat Enterprise Linux 7.2 on Data Nodes

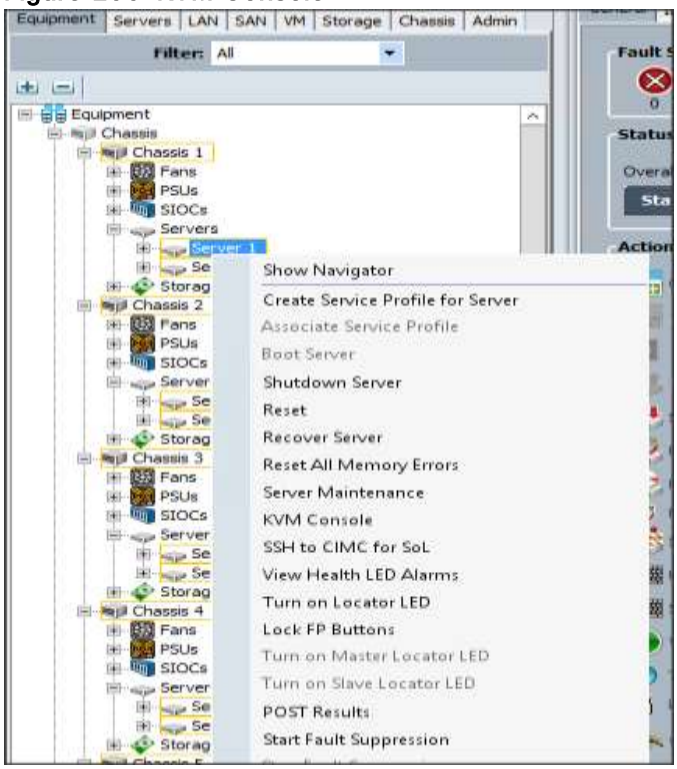
The following section provides detailed procedures for installing Red Hat Enterprise Linux 7.2 on Cisco UCS S3260 Storage Servers. There are multiple ways to install the Red Hat Linux operating system. The installation procedure described in this deployment guide uses KVM console and virtual media from Cisco UCS Manager.

Note: This requires RHEL 7.2 DVD/ISO for the installation

To install the Red Hat Linux 7.2 operating system, complete the following steps:

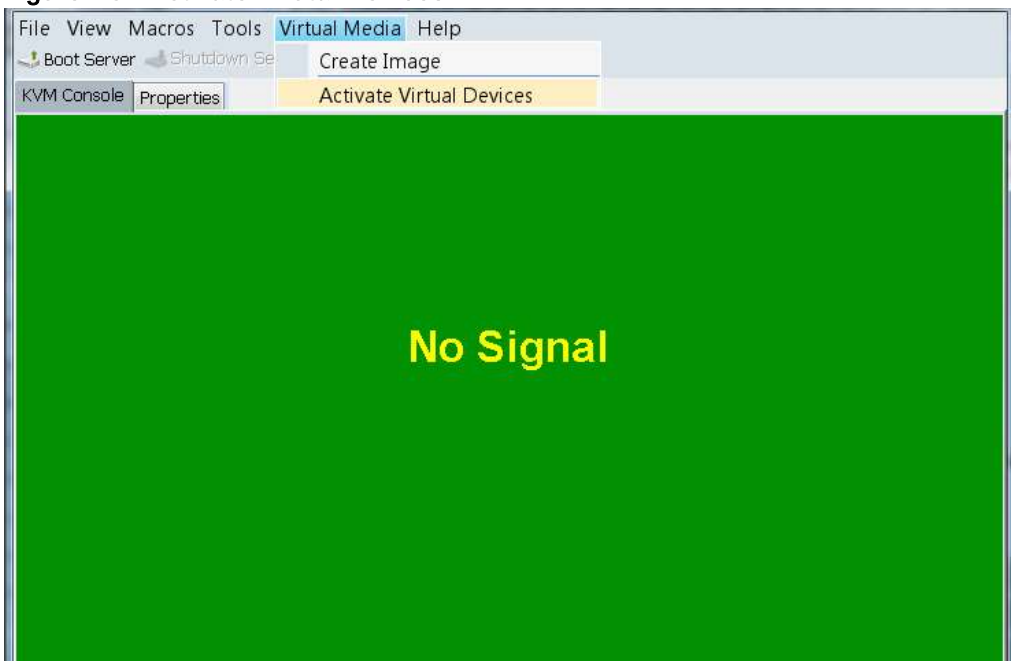
1. Log in to the Cisco UCS 6332 Fabric Interconnect and launch the Cisco UCS Manager application.
2. Select the `Equipment` tab.
3. In the navigation pane expand `Chassis` and then `Servers`
4. Right click on the server and select `KVM Console`. (Figure 106)
5. In the KVM window, select the `Virtual Media` tab.

Figure 106 KVM Console



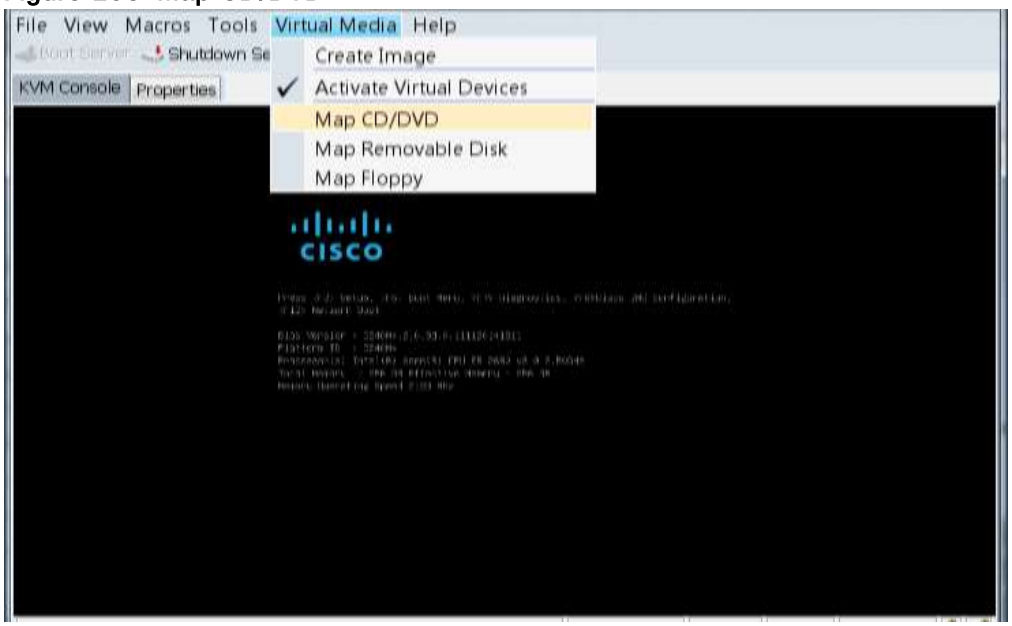
6. Click the `Activate Virtual Devices` found in the Virtual Media tab. (Figure 107)

Figure 107 Activate Virutal Devices



7. In the KVM window, select the Virtual Media tab and click the `Map CD/DVD`. (Figure 108)

Figure 108 Map CD/DVD

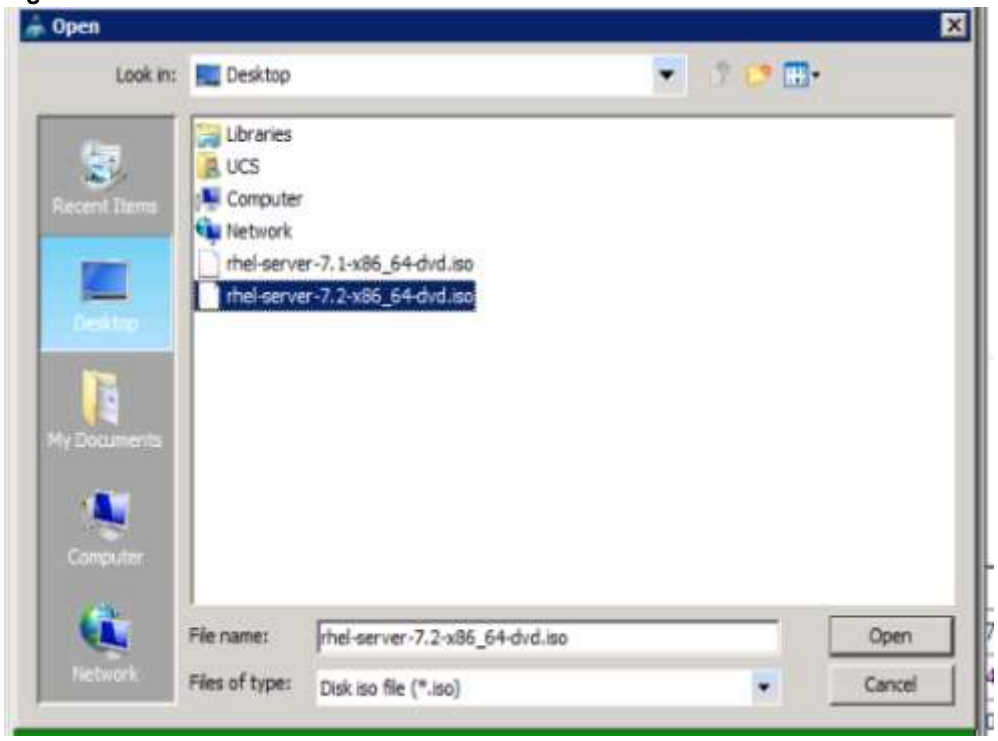


8. Browse to the Red Hat Enterprise Linux Server 7.2 installer ISO image file.

Note: The Red Hat Enterprise Linux 7.2 DVD is assumed to be on the client machine.

9. Click Open to add the image to the list of virtual media. (Figure 109)

Figure 109 Select the rhel-server



10. In the KVM window, select the KVM tab to monitor during boot.

11. In the KVM window, select the `Macros > Static Macros > Ctrl-Alt-Del` button in the upper left corner.

12. Click `OK`

13. Click `OK` to reboot the system.

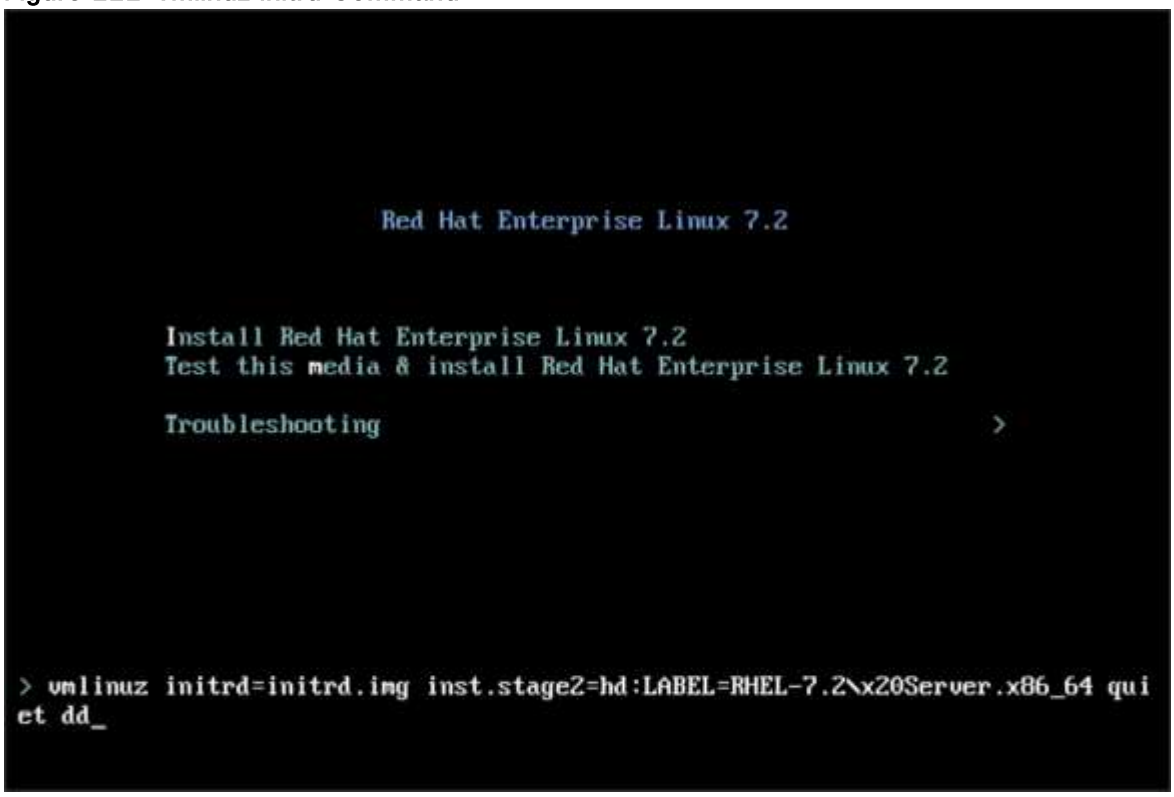
14. On reboot, the machine detects the presence of the Red Hat Enterprise Linux Server 7.2 install media.

15. When you see the Install Red Hat screen, press tab for full configuration options. Figure 110

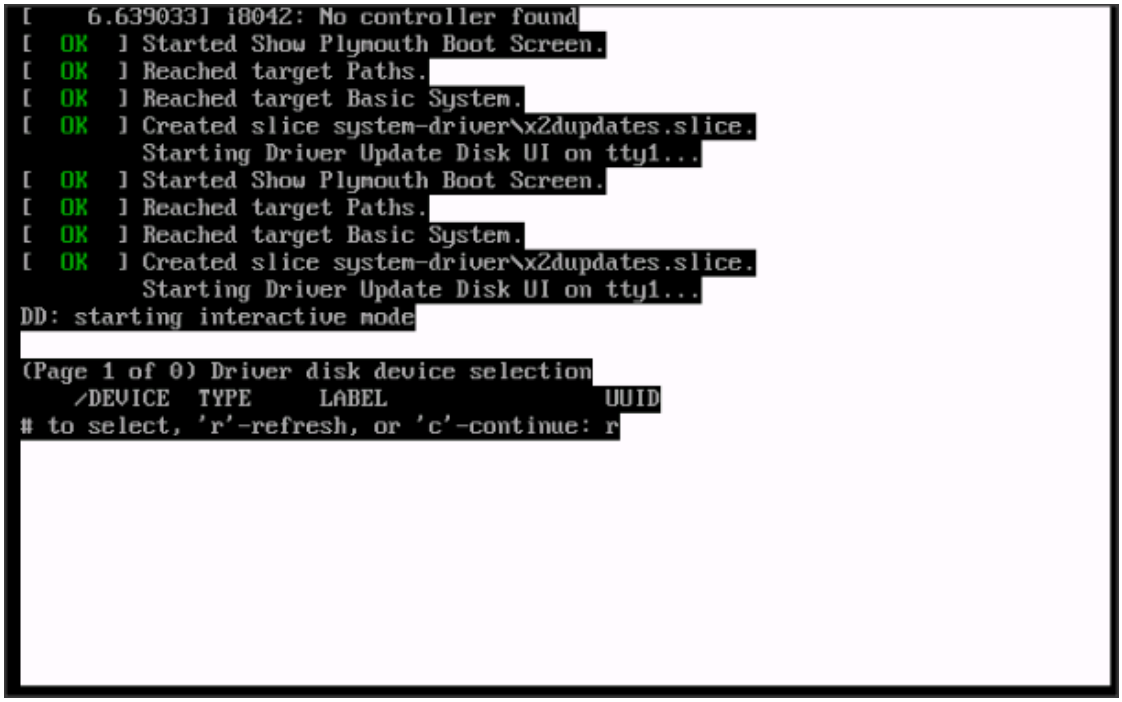
Figure 110 Install Red Hat Enterprise Linux 7.2

16. The `vmlinuz initrd` command pops up, Figure 111

Figure 111 vmlinuz initrd Command



17. Add a space, type dd, then press enter.



18. Type r and press enter. Now verify the Redhat 7.2 iso is mounted.

19. Next, unmount the Redhat 7.2 iso from the virtual media, and mount the Megaraid 7.2 iso image. Figure 112

Figure 112 Virtual Media

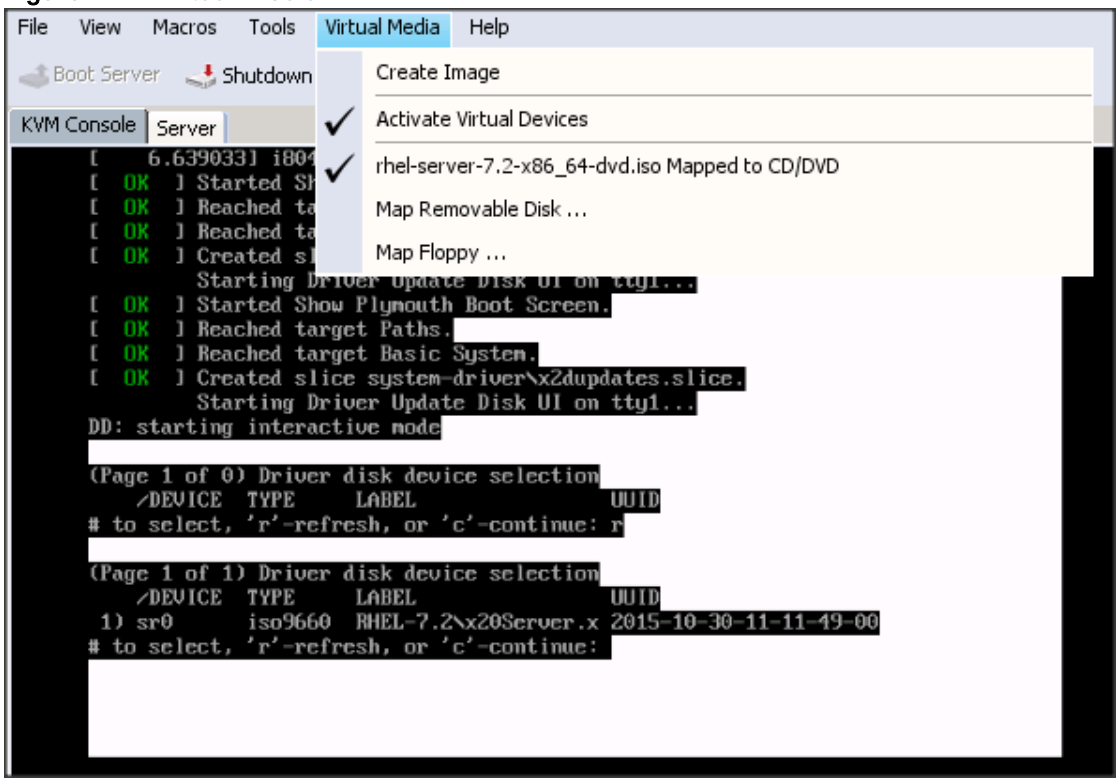
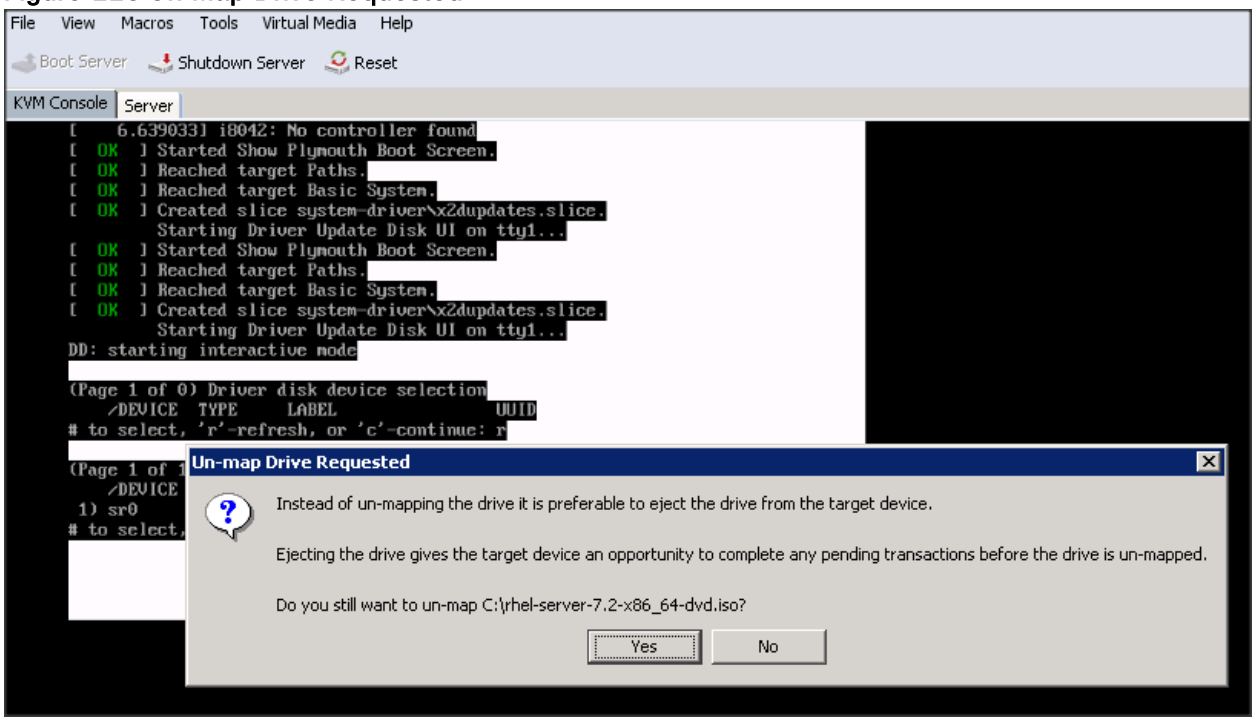
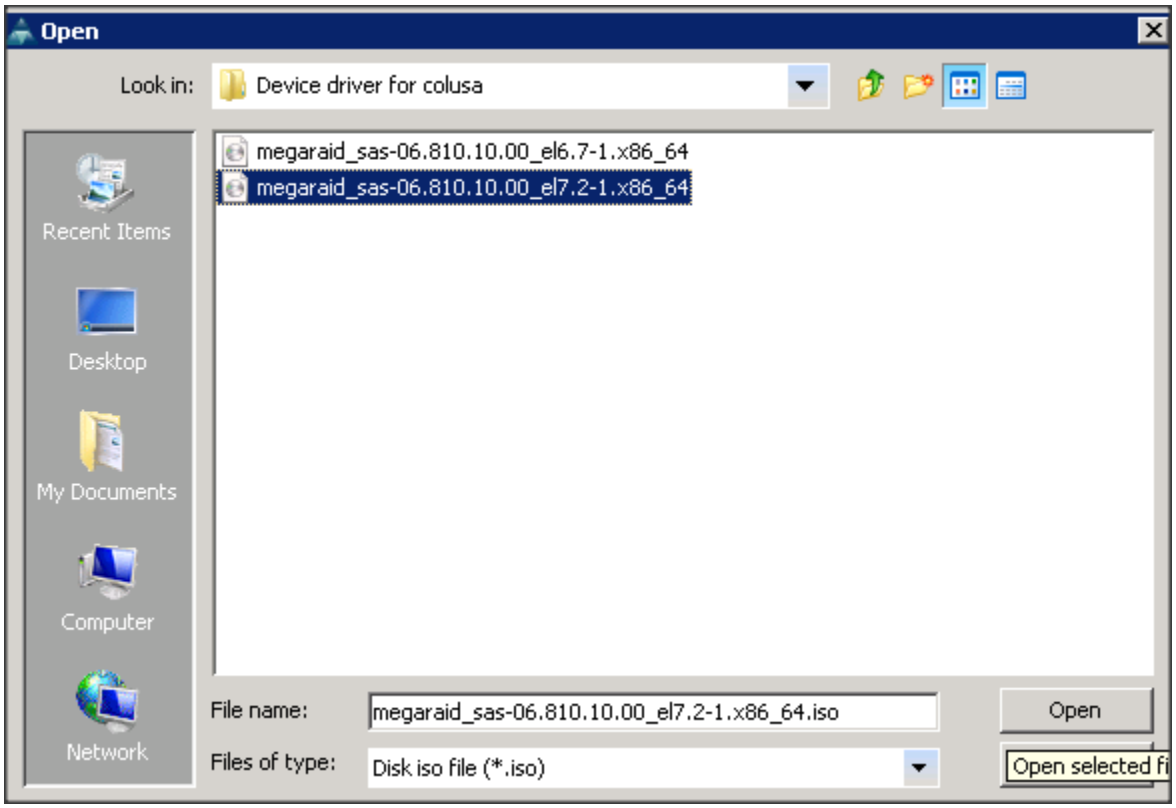


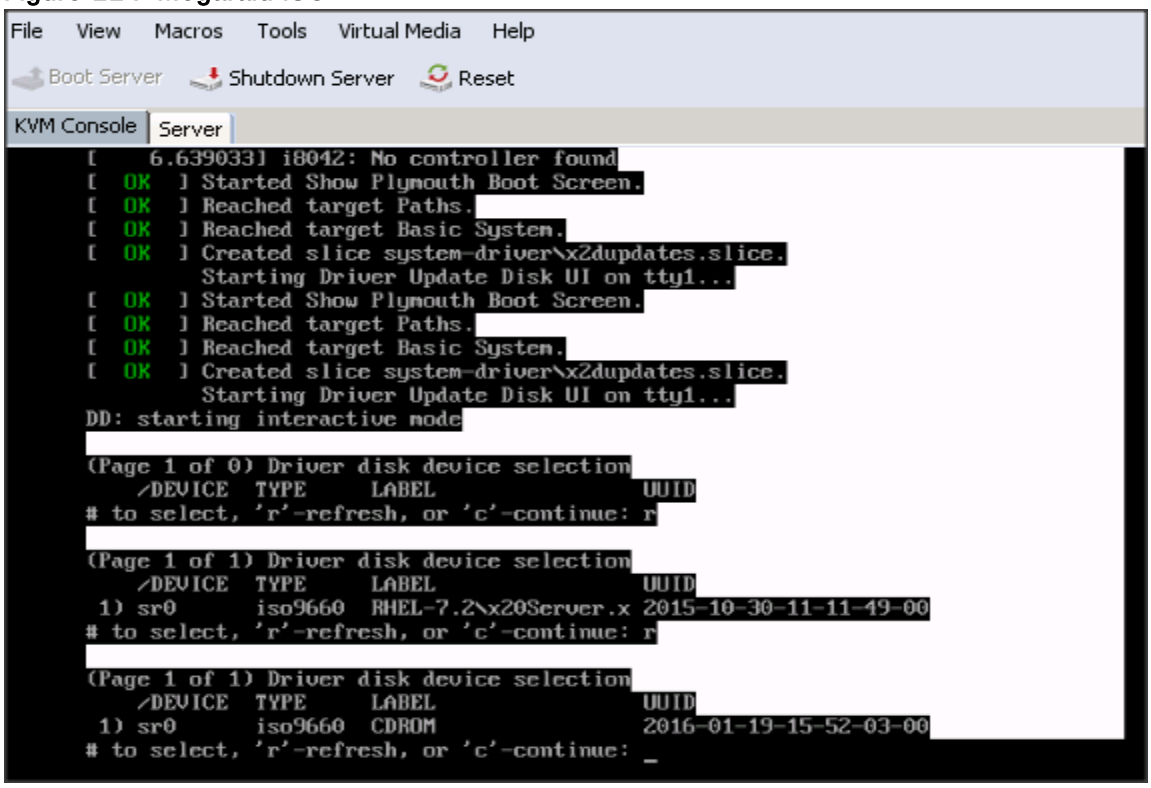
Figure 113 Un Map Drive Requested





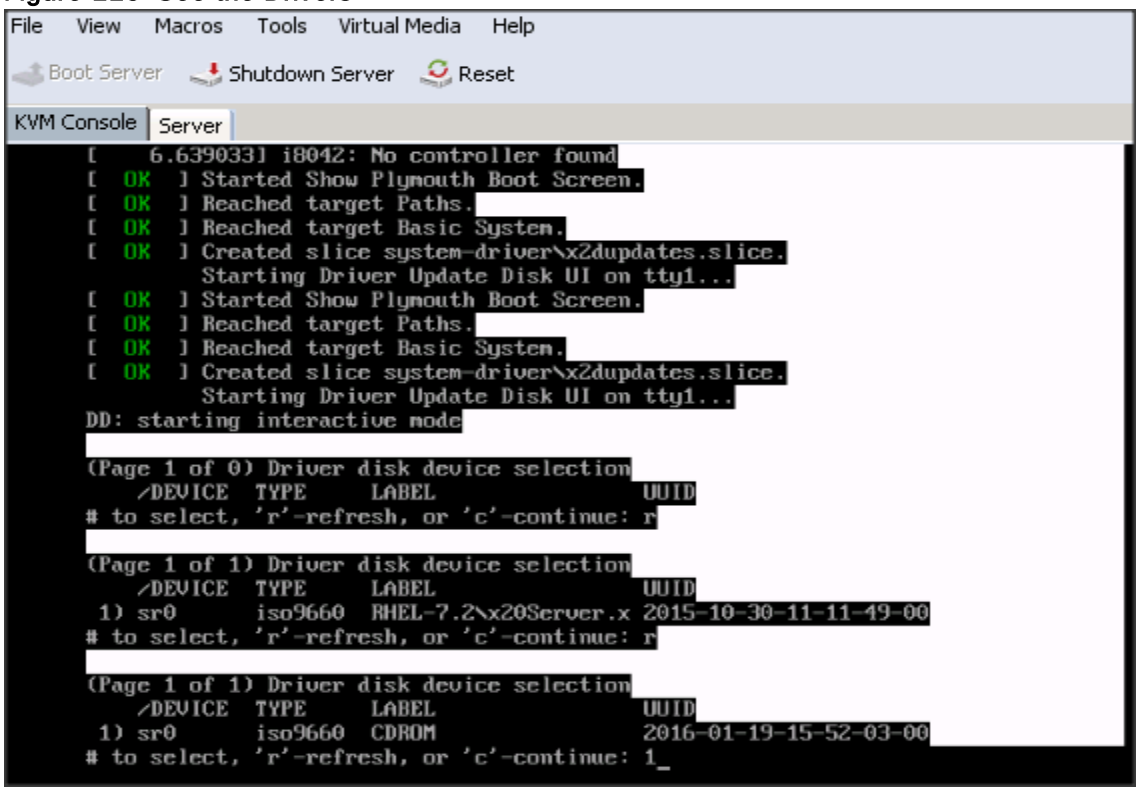
20. After mounting the Megaraid iso, type r again to see the megaraid iso. Figure 114

Figure 114 Megaraid ISO



21. Type ' and press enter to see the drivers. Figure 115

Figure 115 See the Drivers



22. Type ' to check the box and select the drivers. Press enter. Figure 116

Figure 116 Check the box

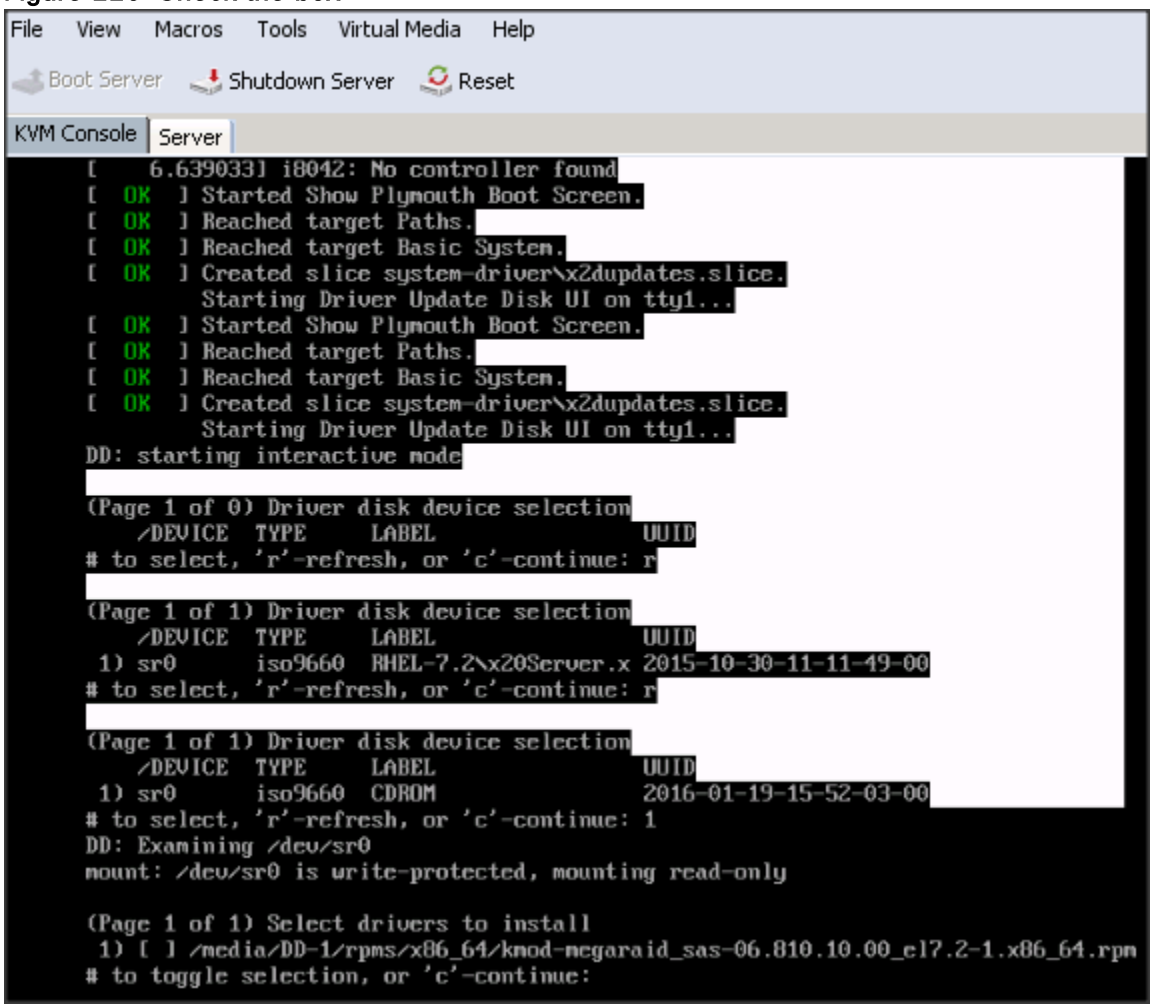
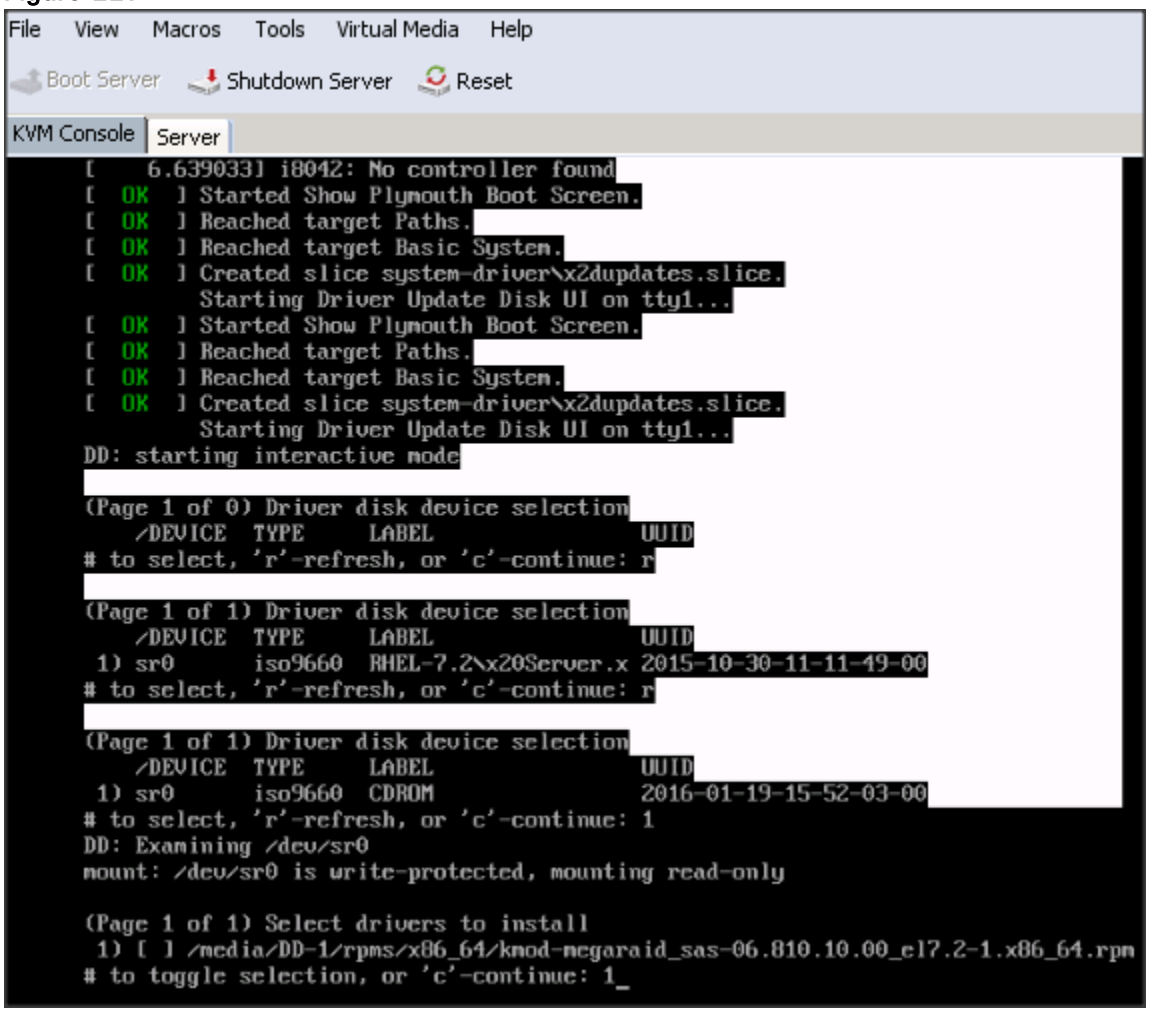


Figure 117



23. Once the box is checked, type :: to continue, and press enter.

Figure 118 Megaraid is mounted

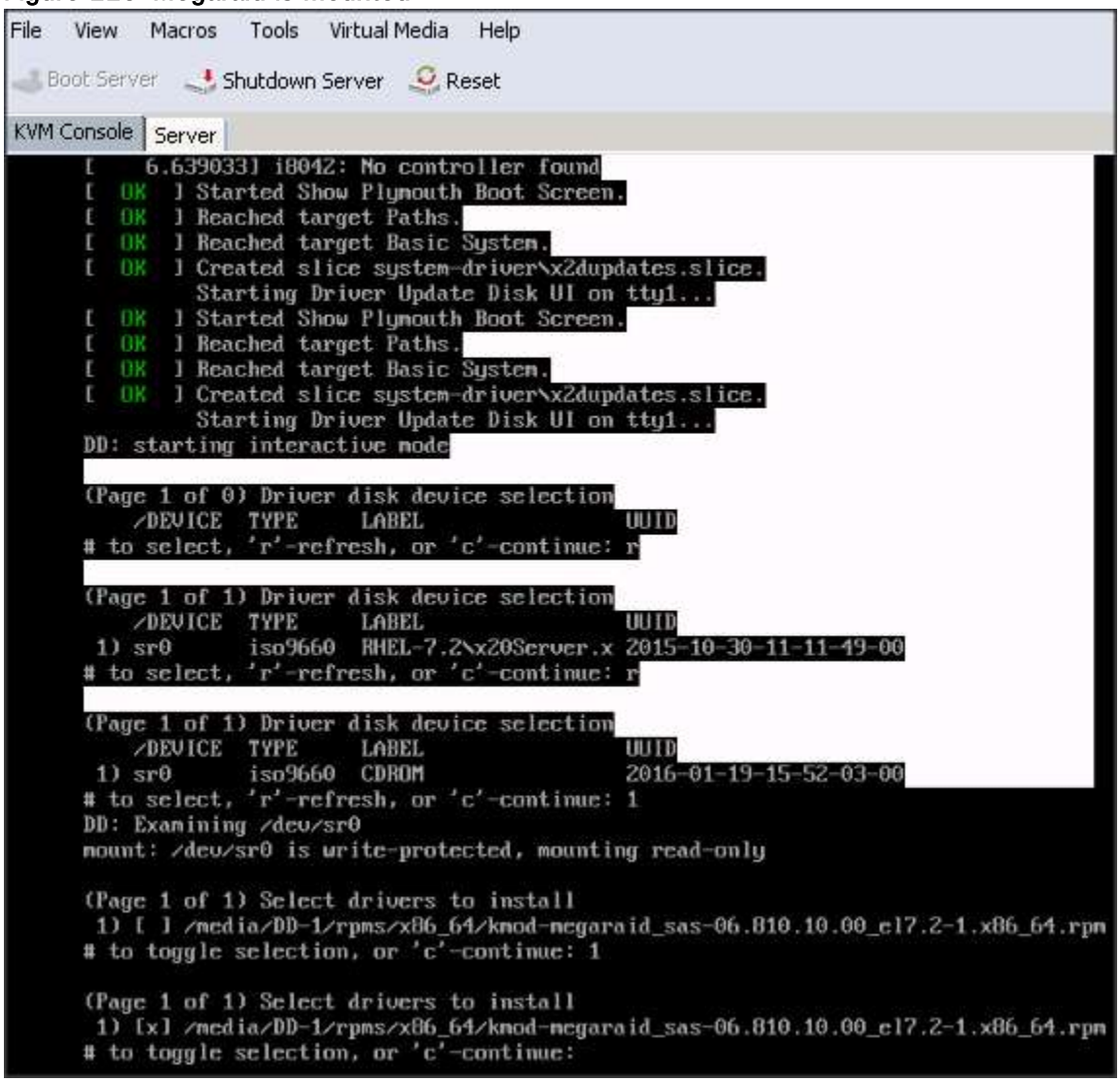
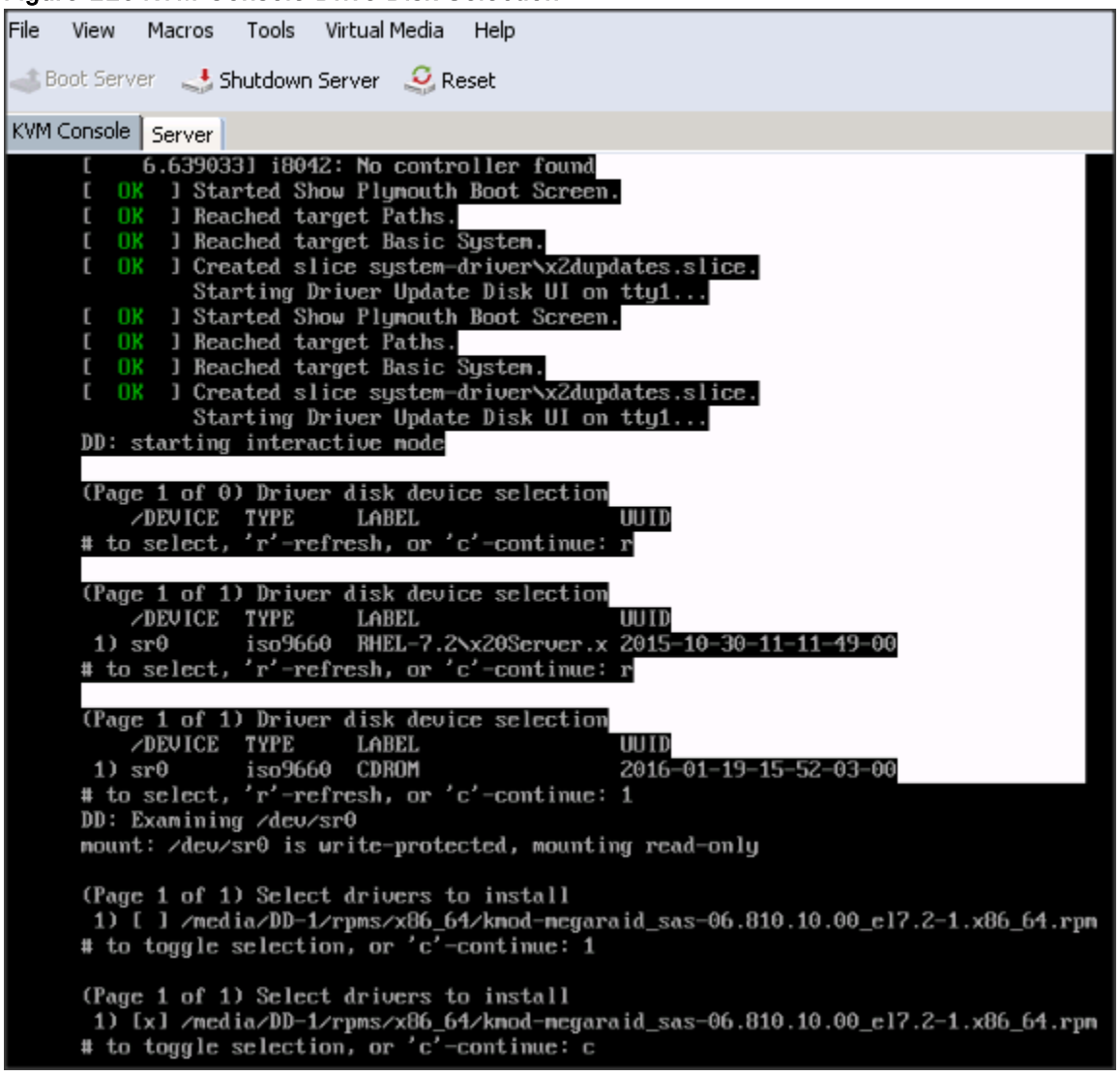
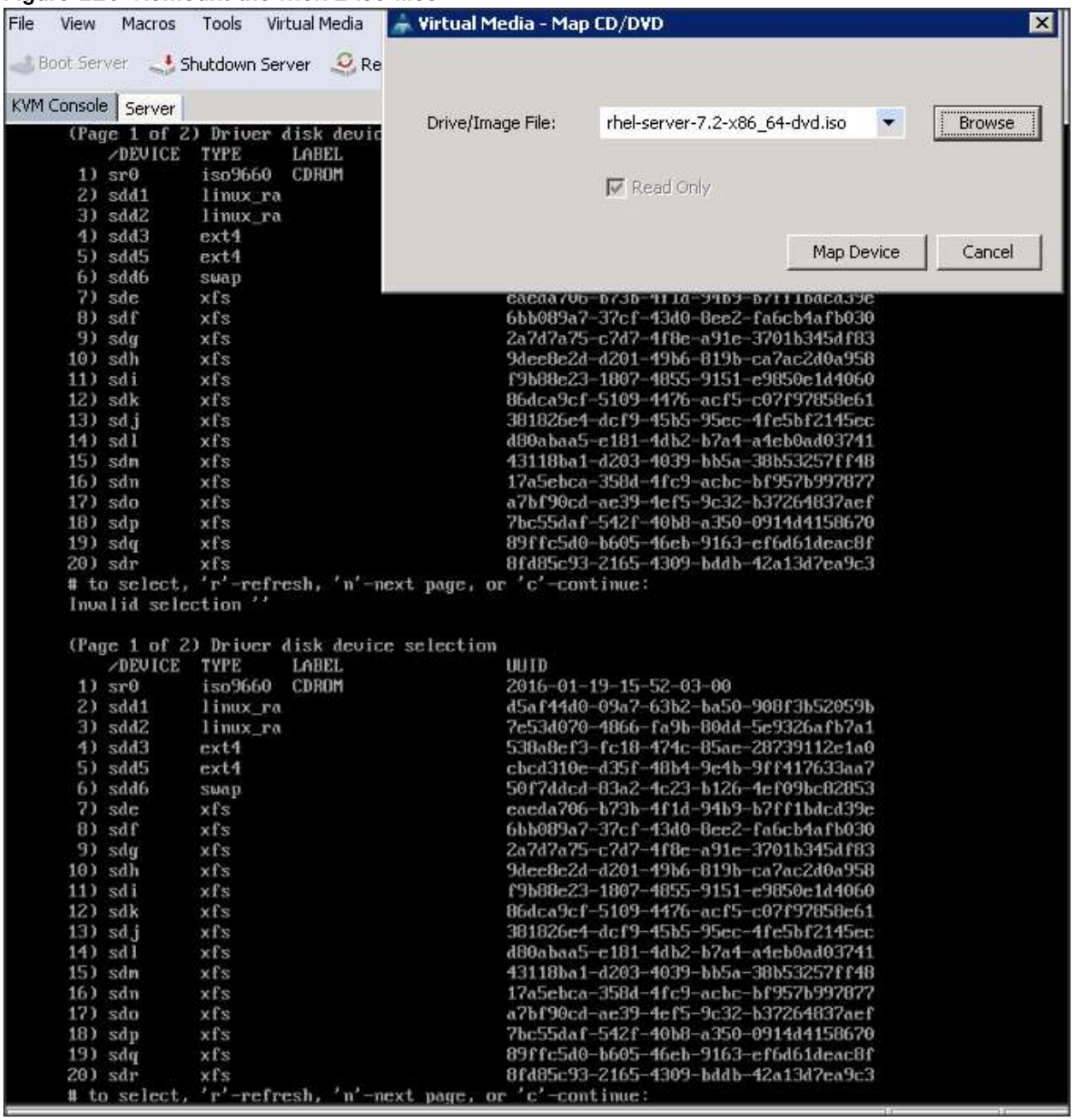


Figure 119 KVM Console Drive Disk Selection



24. After the driver extraction, unmount the megaraid driver from the virtual media and re-mount the Redhat 7.2 iso image. Figure 120

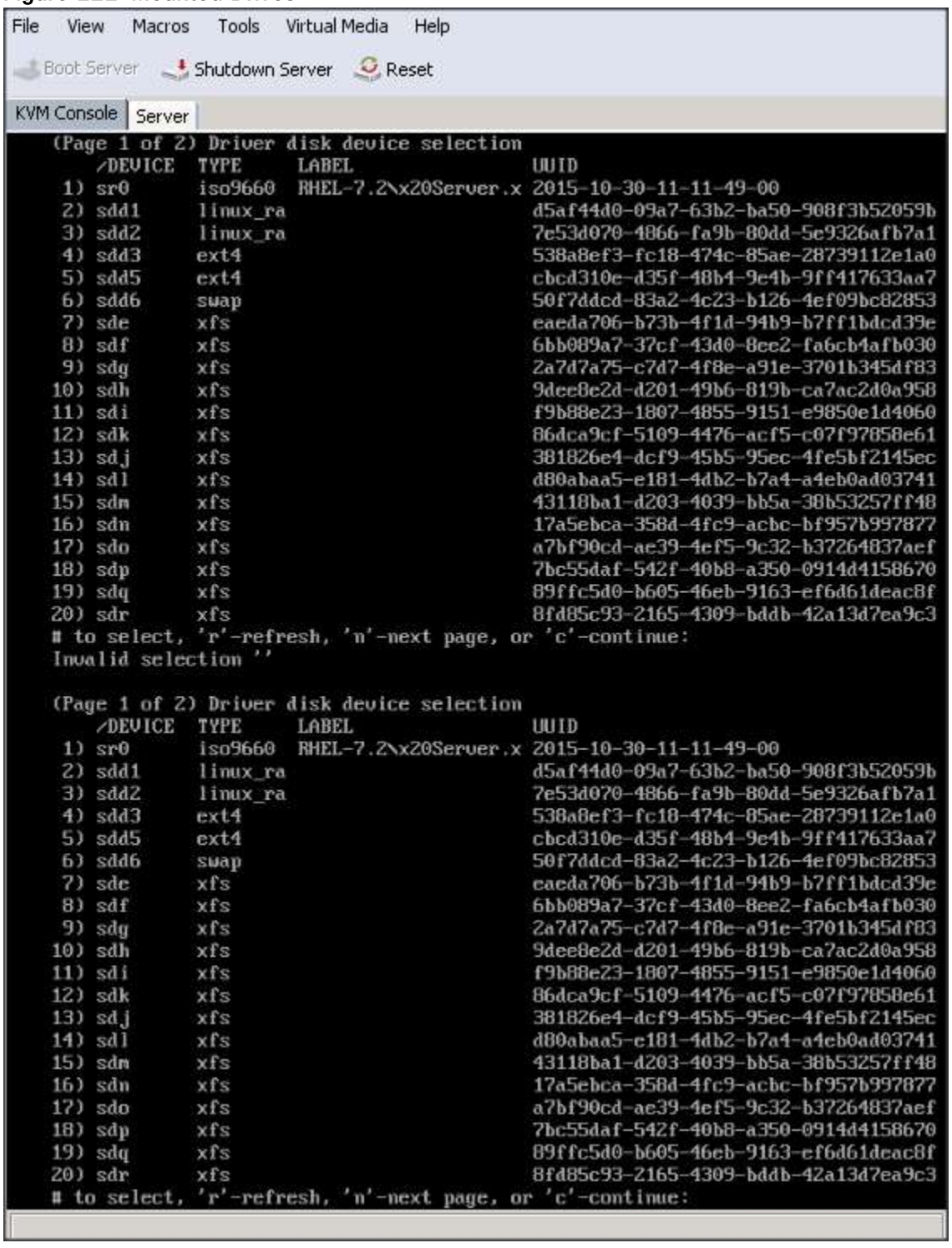
Figure 120 Remount the rhel72 iso files



25. After mounting the Redhat 7.2 iso image in the virtual media , type r to confirm the iso in the command shell.

26. Verify the Redhat 7.2 iso image and type :, type :: to continue the installation of the Redhat 7.2. Figure 121

Figure 121 Mounted Drives



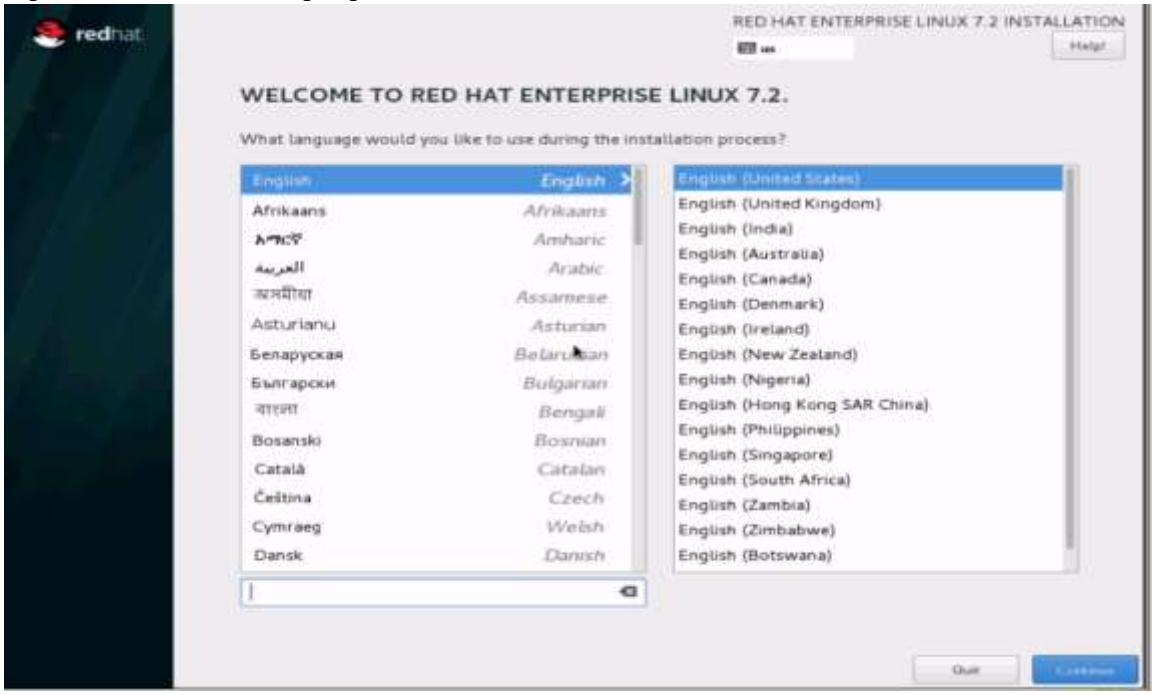
27. Now enter :: to continue the installation process.

28. Skip the Media test and start the installation. Figure 122

Figure 122 Installation Screen

29. Select language of installation and click Continue. (Figure 123)

Figure 123 Choose Language of Installation



30. Select Date and Time, (Figure 124) which pops up another window as shown below in Figure 125

Figure 124 Date and Time

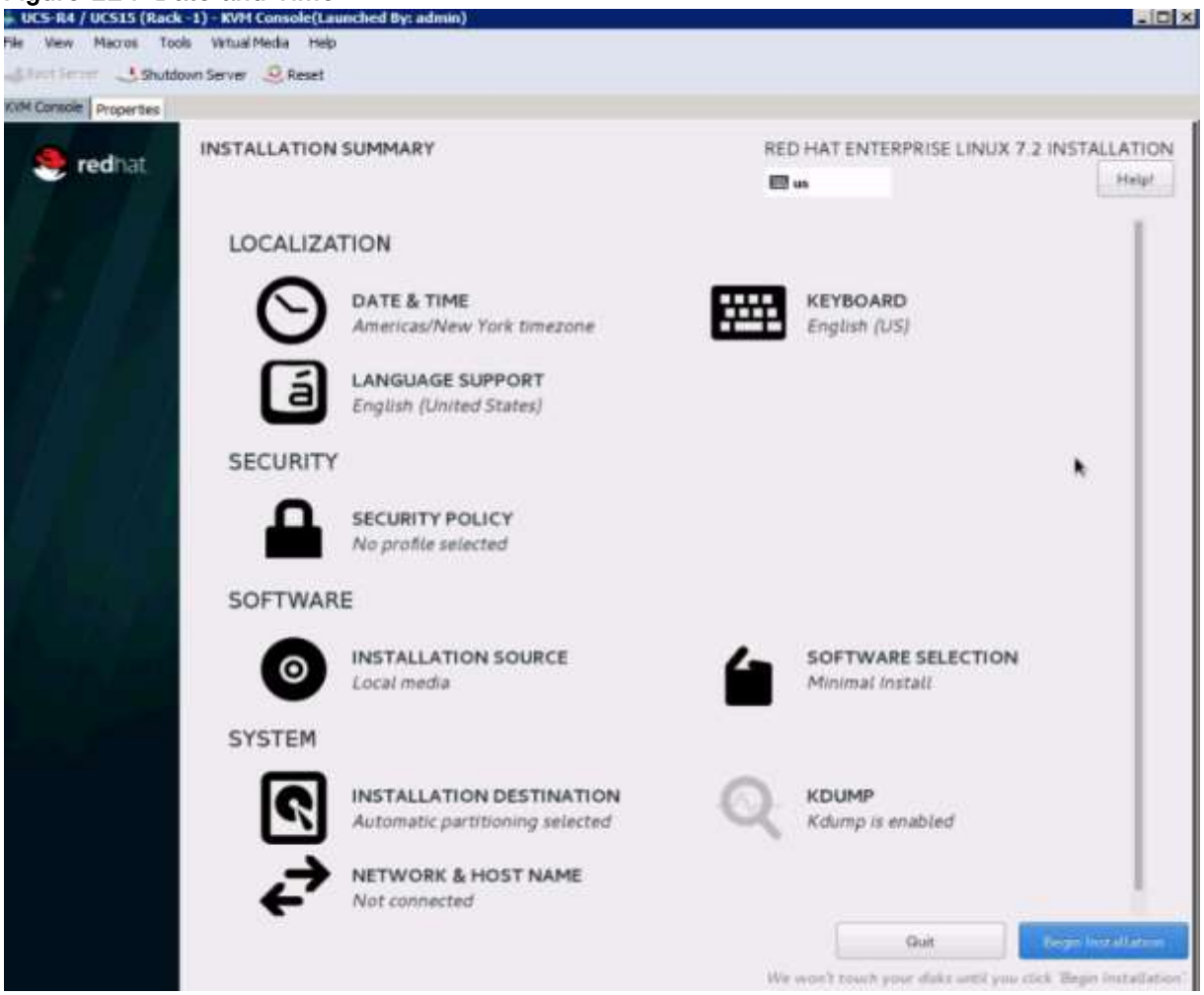
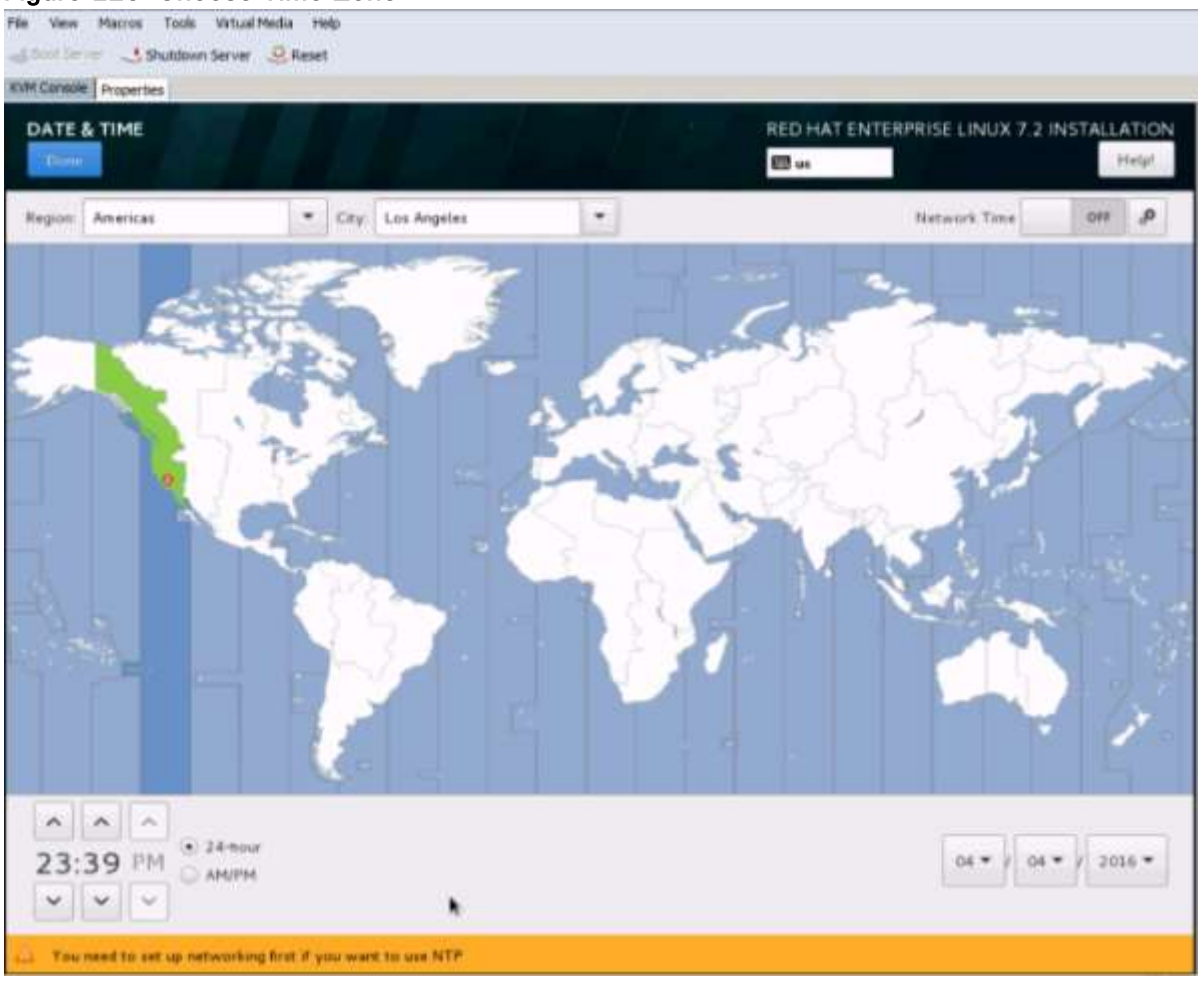


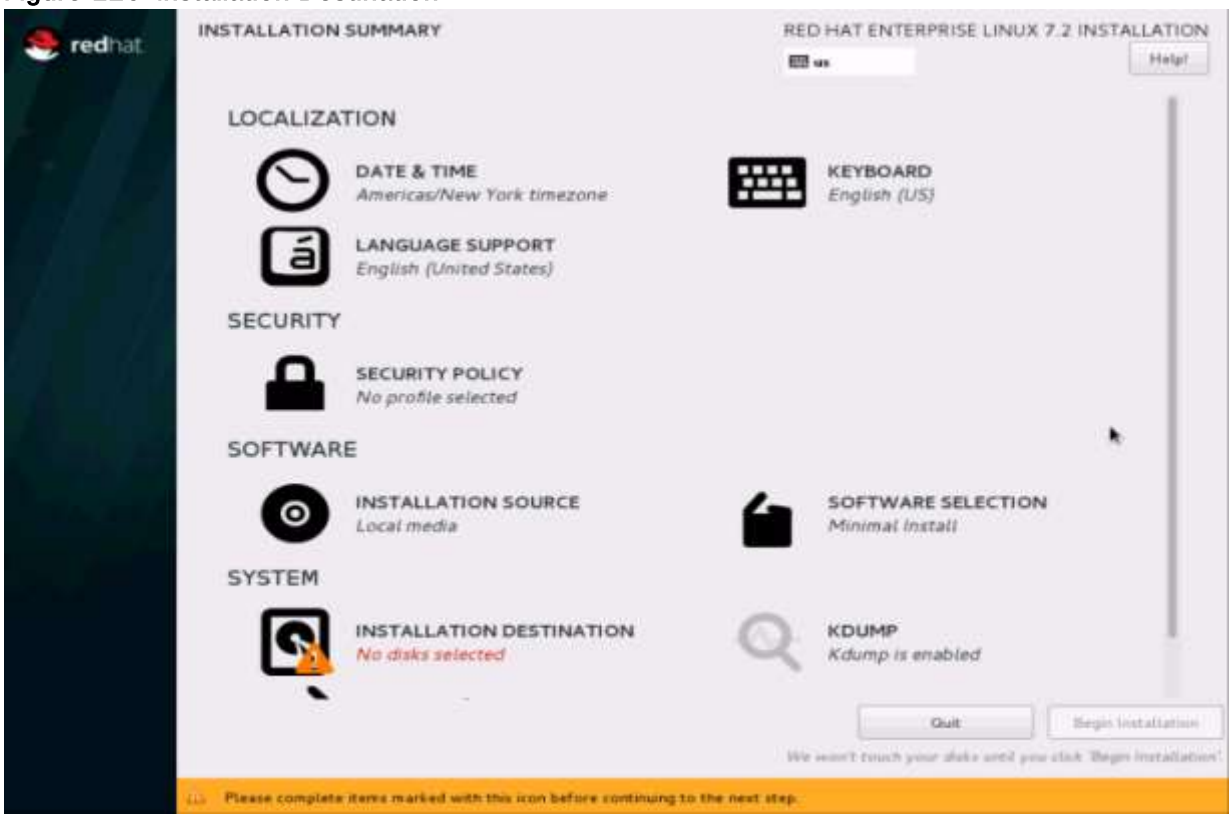
Figure 125 Choose Time Zone



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

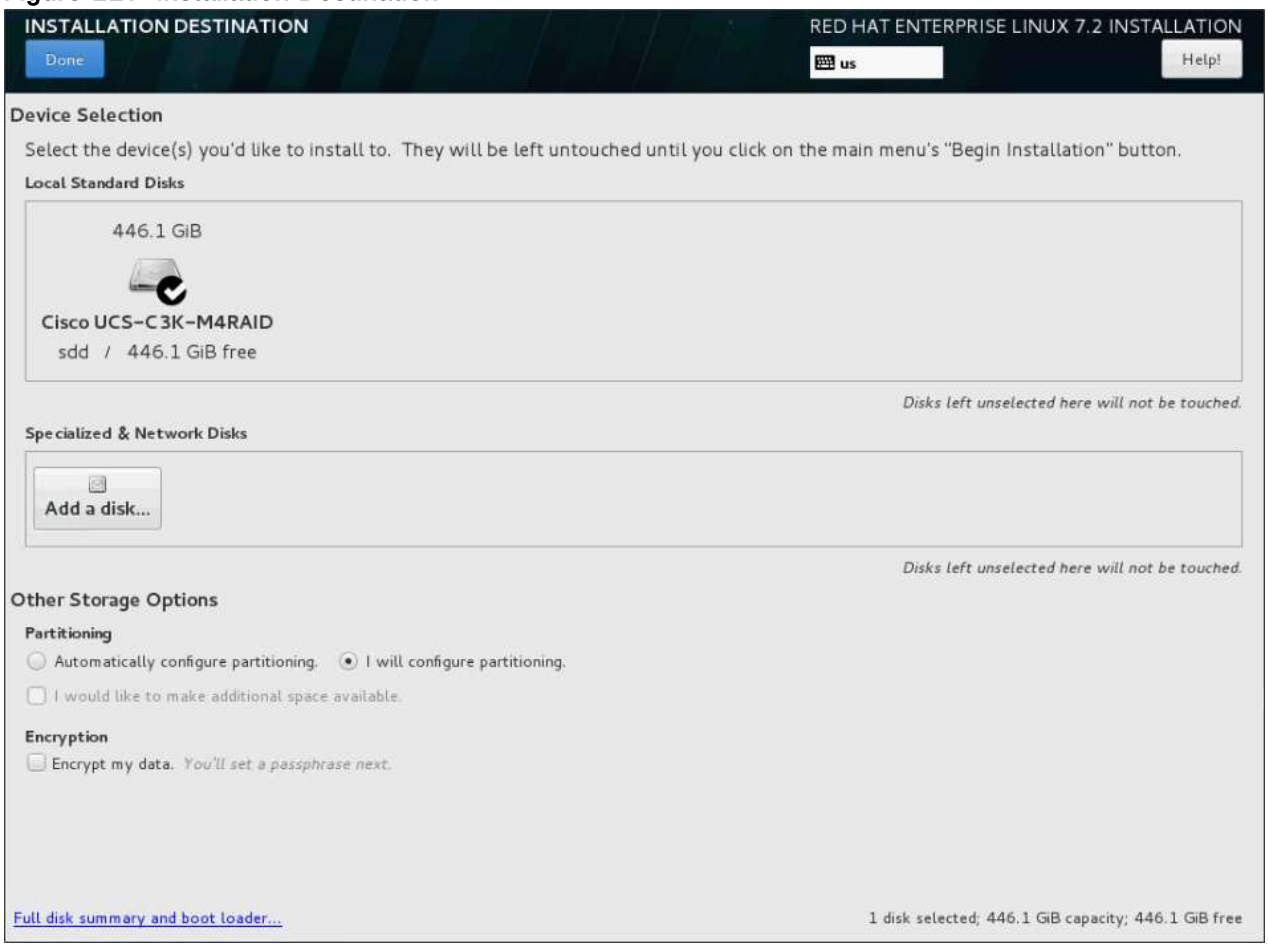
32. Click on Installation Destination. (Figure 126)

Figure 126 Installation Destination



33. This opens a new window with the boot disks. Make the selection, and choose I will configure partitioning. Click Done. (Figure 127)

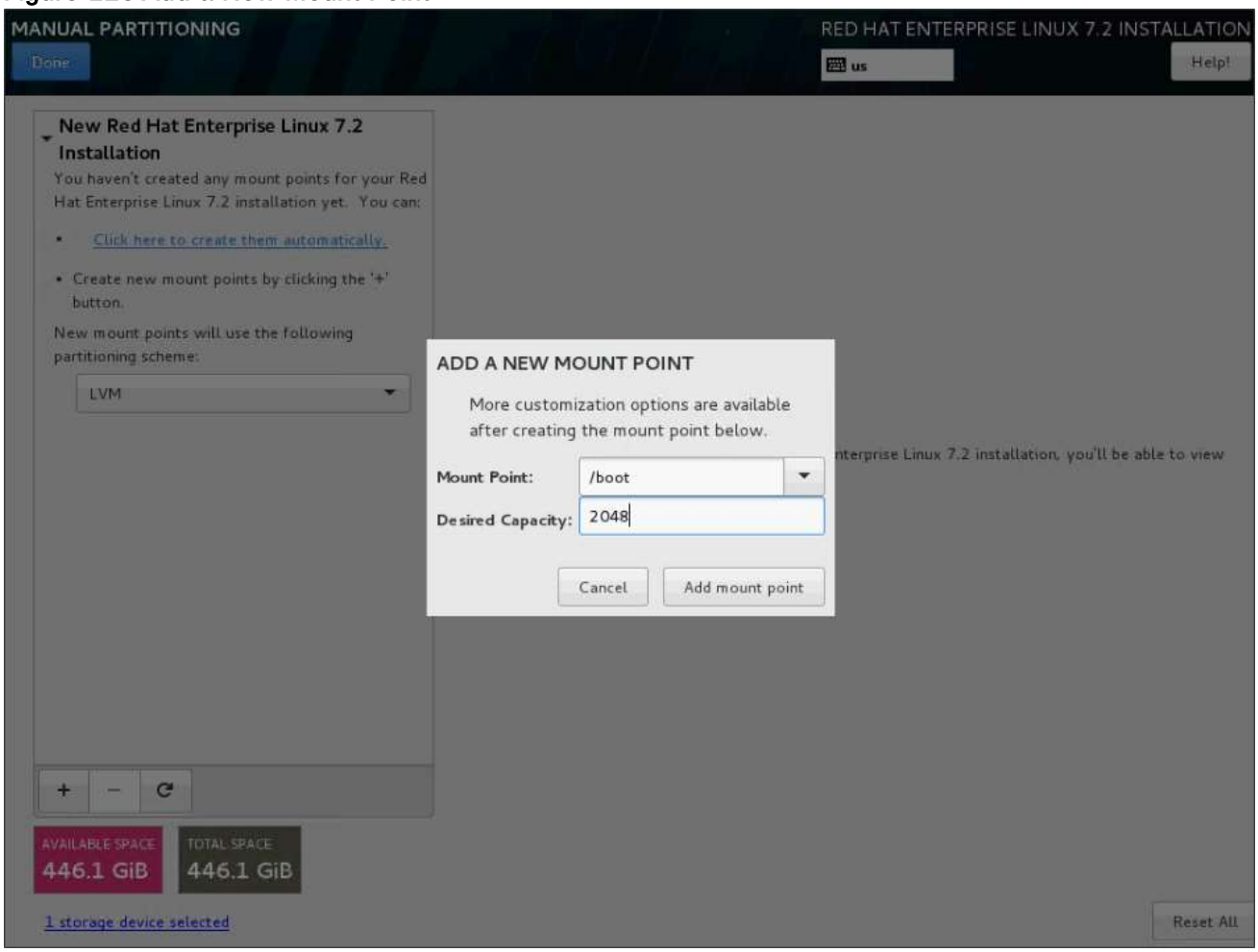
Figure 127 Installation Destination



34. This opens the new window for creating the partitions. (Figure 128) Click on the + sign to add a new partition as shown below, boot partition of size 2048 MB.

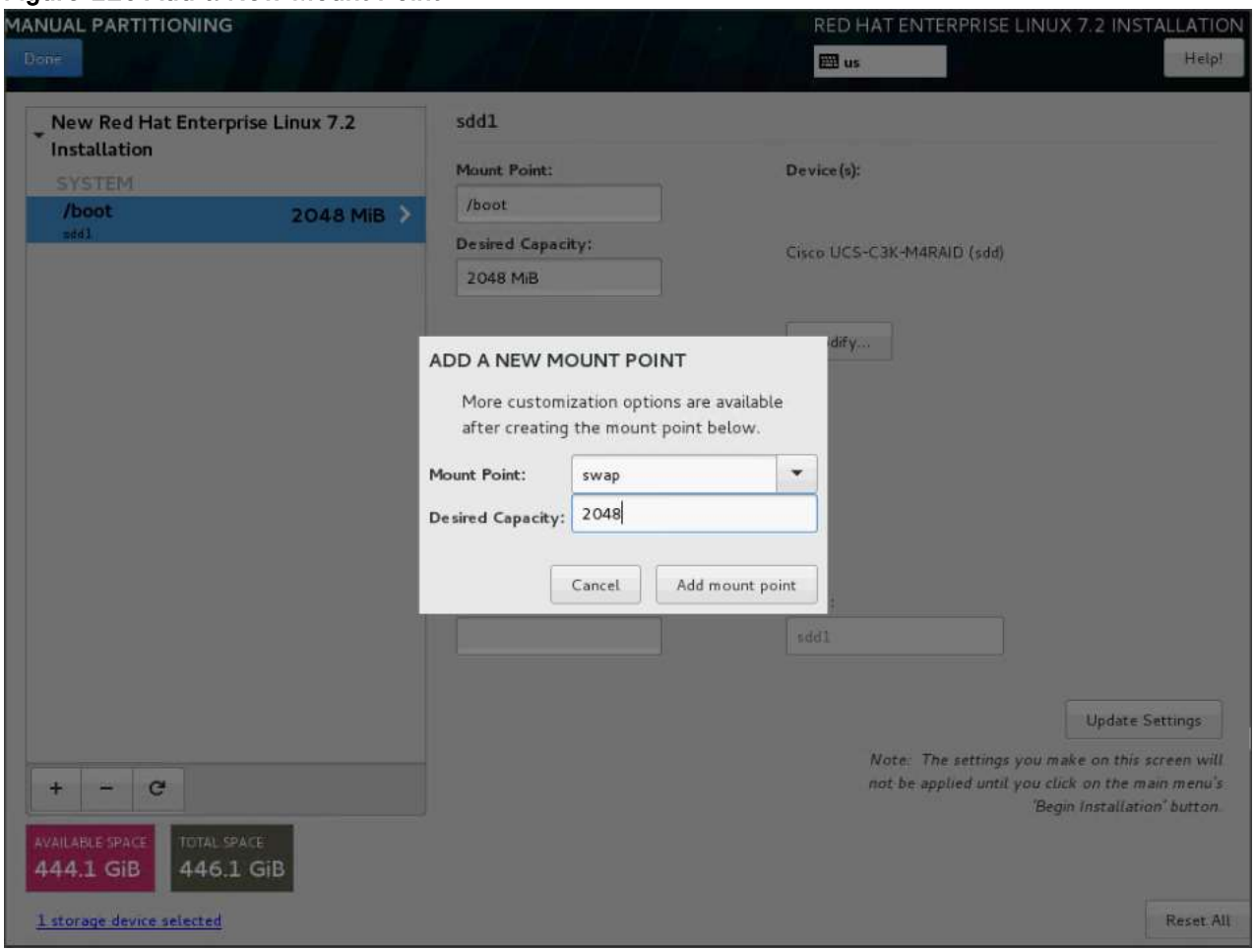
35. Click Add MountPoint to add the partition.

Figure 128 Add a New Mount Point



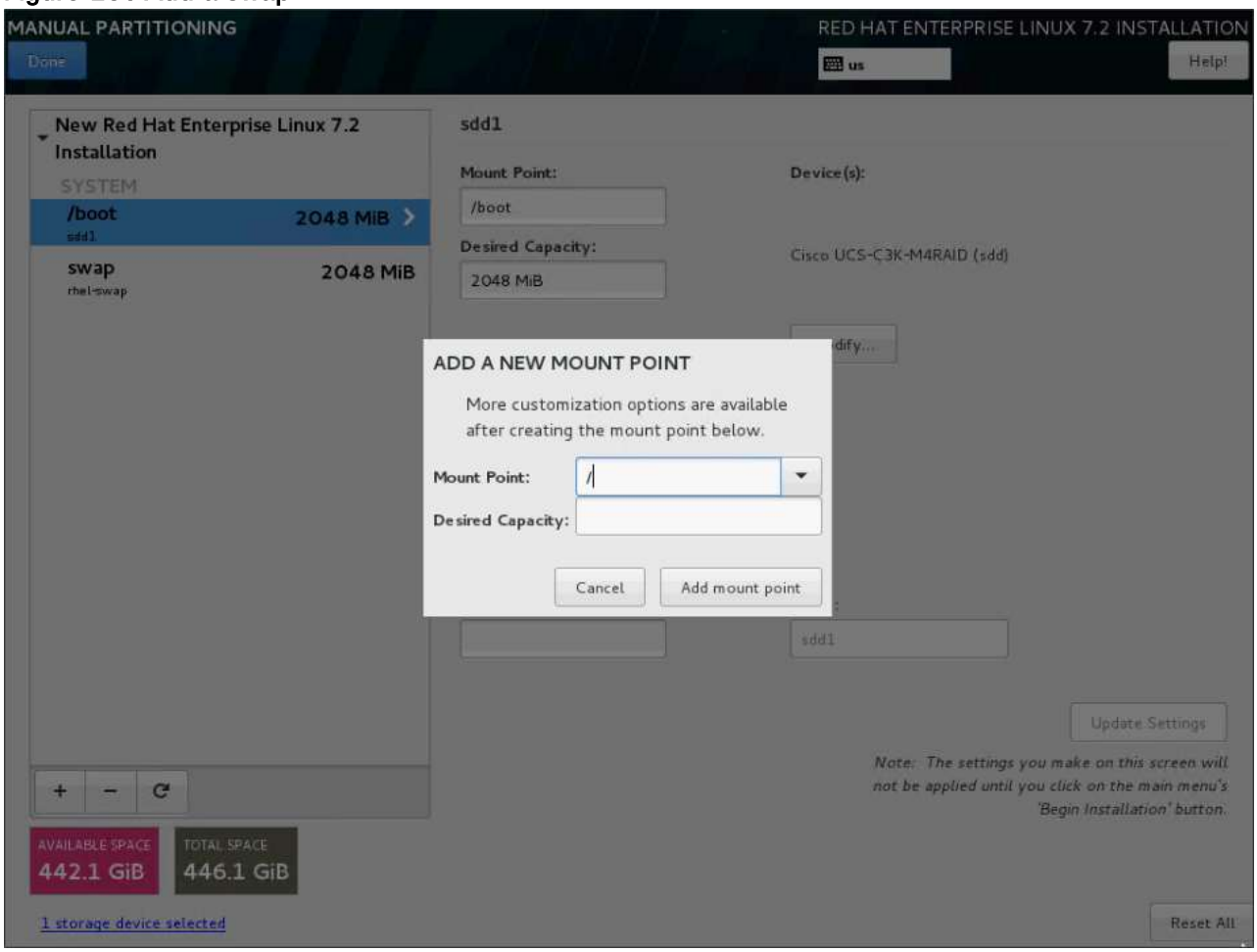
36. Click on the + sign to create the swap partition of size 2048 MB as shown below. (Figure 129)

Figure 129 Add a New Mount Point



37. Click + to add the / partition. The size can be left empty so it uses the remaining capacity and click Add Mountpoint. (Figure 130)

Figure 130 Add a swap

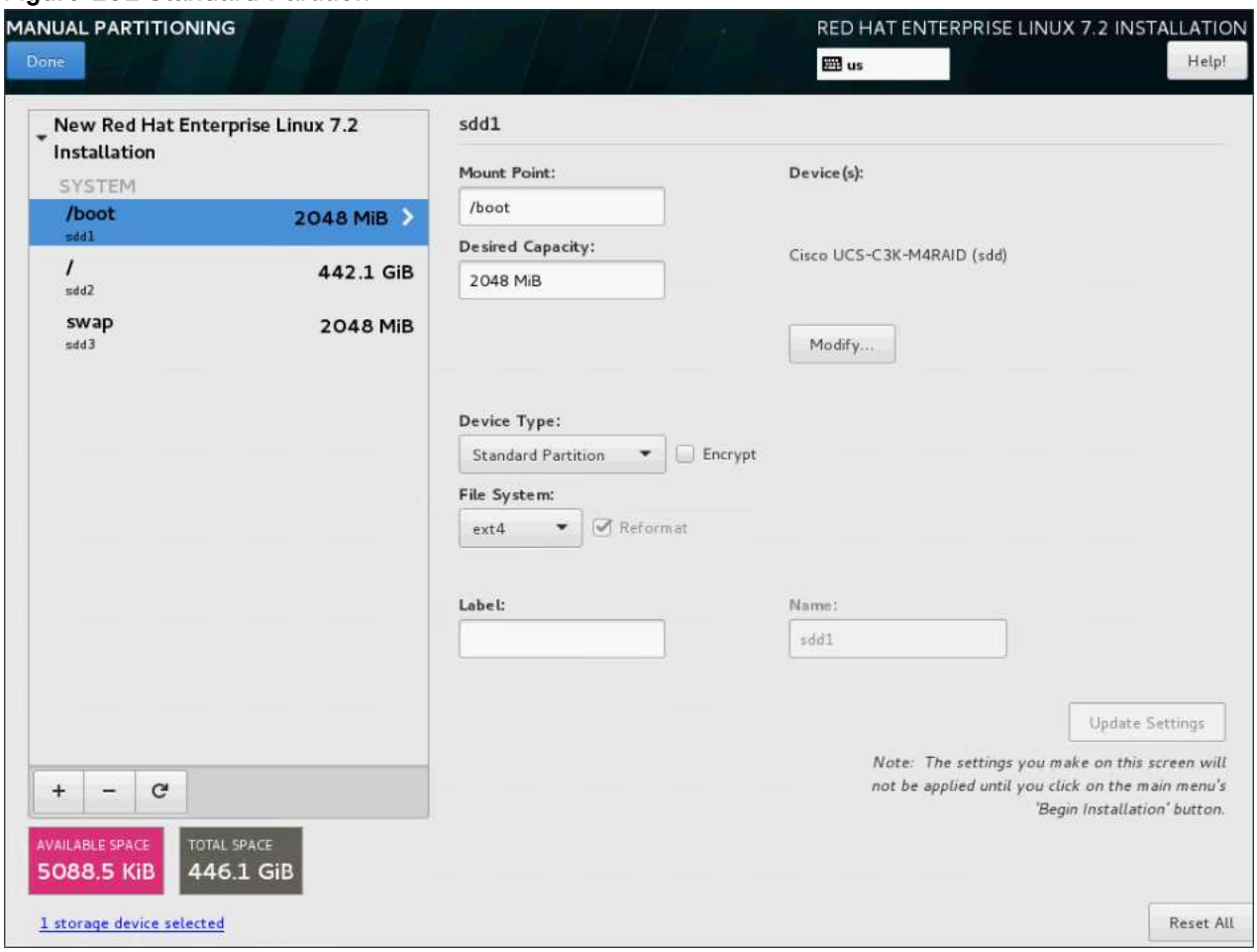


38. Select /boot partition and change the Device Type to Standard Partition and the file system to ext4. (Figure 131)

39. Select "/" partition and change the Device Type to Standard Partition and the file system to ext4.

40. Select "swap" partition and change the Device Type to Standard Partition.

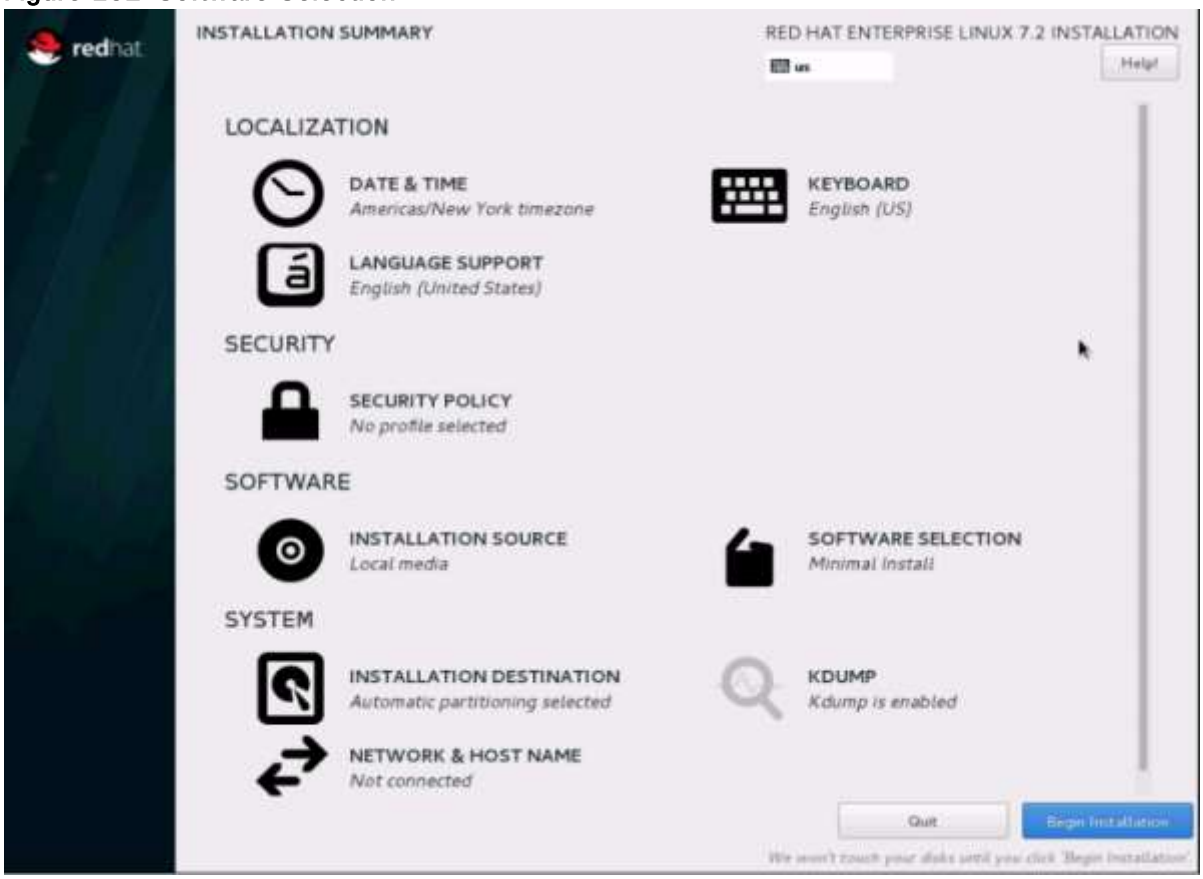
Figure 131 Standard Partition



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

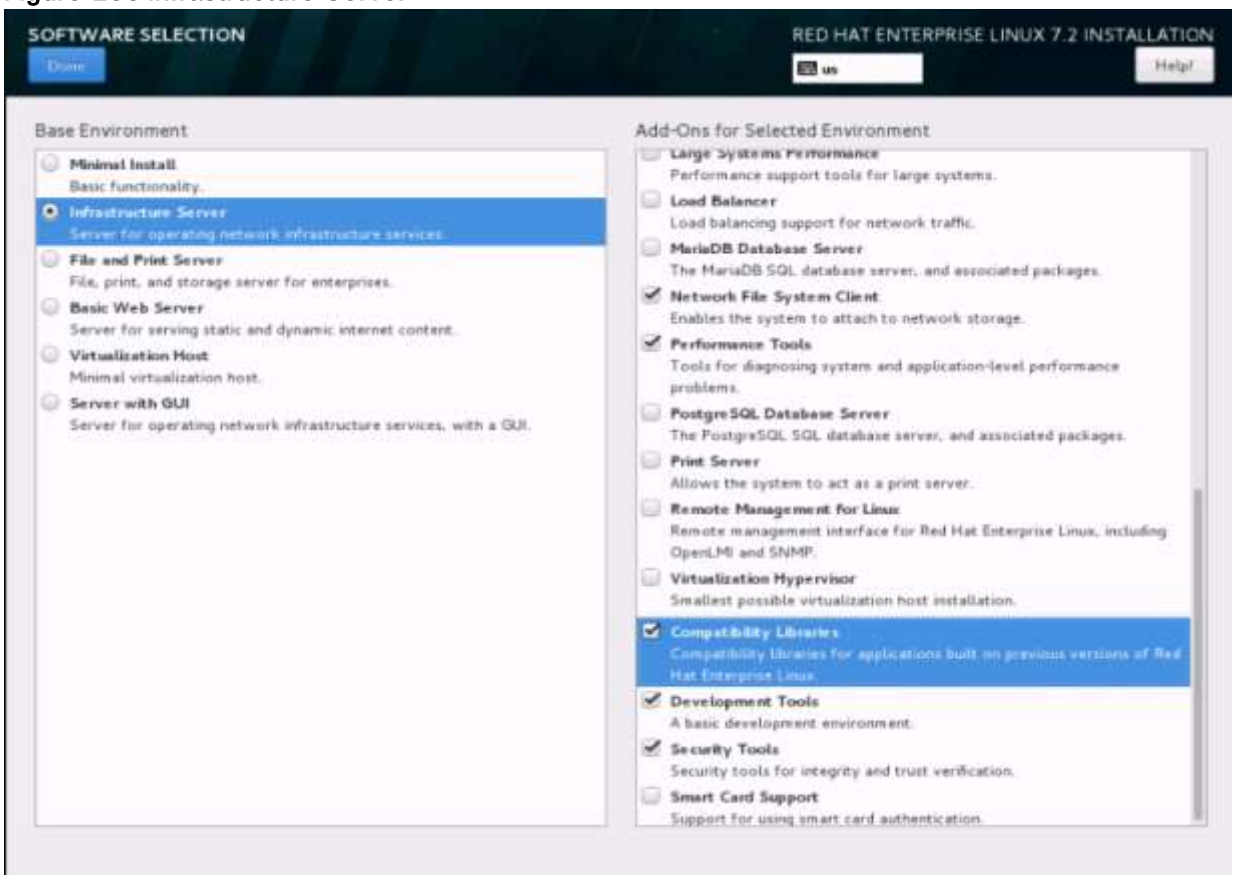
42. Click on Software Selection. (Figure 132)

Figure 132 Software Selection



43. Select Infrastructure Server and select the Add-Ons as noted below. Click Done. (Figure 133)

Figure 133 Infrastructure Server

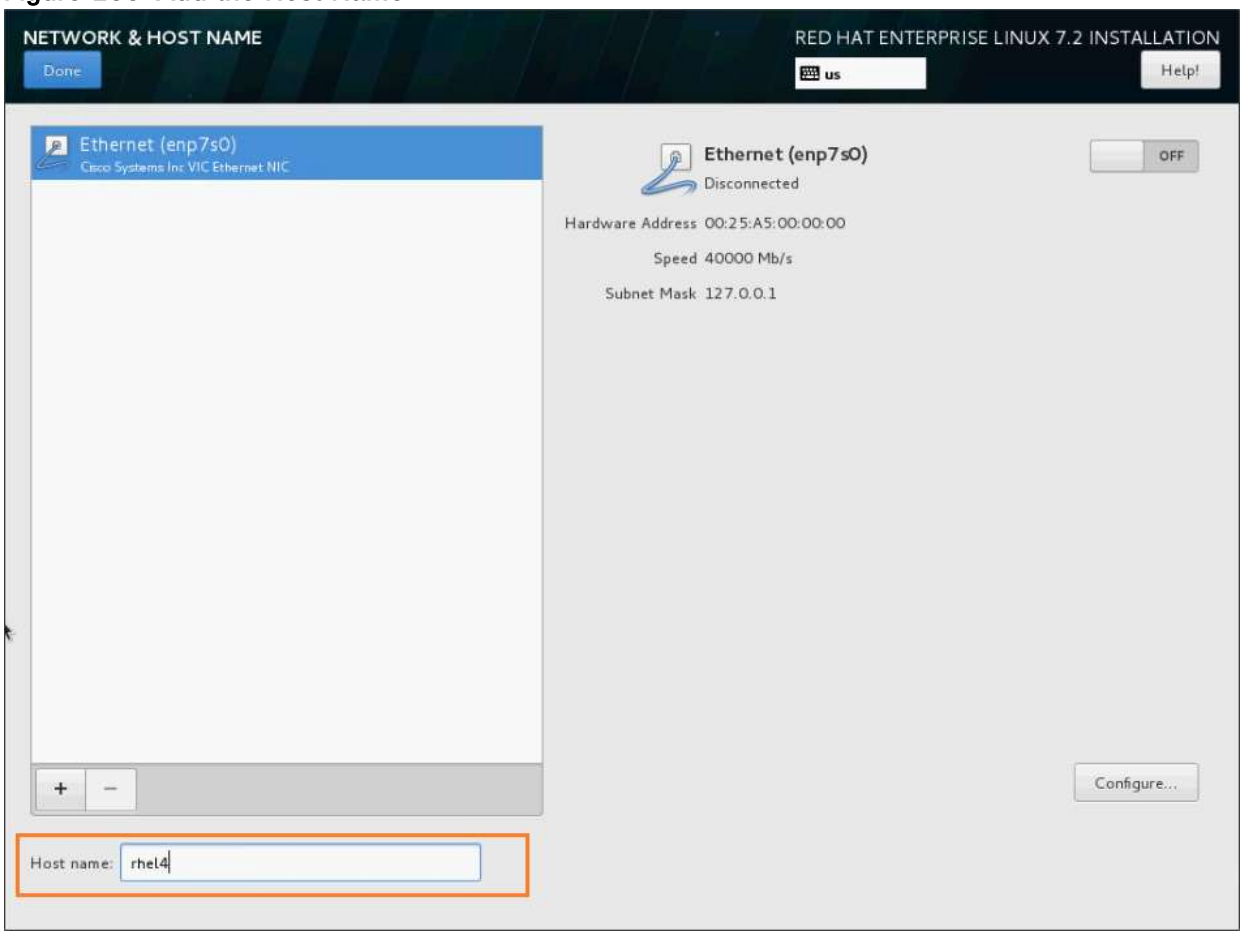


44. Click on Network and Hostname and configure Hostname and Networking for the Host. (Figure 134)

Figure 134 Network and Hostname

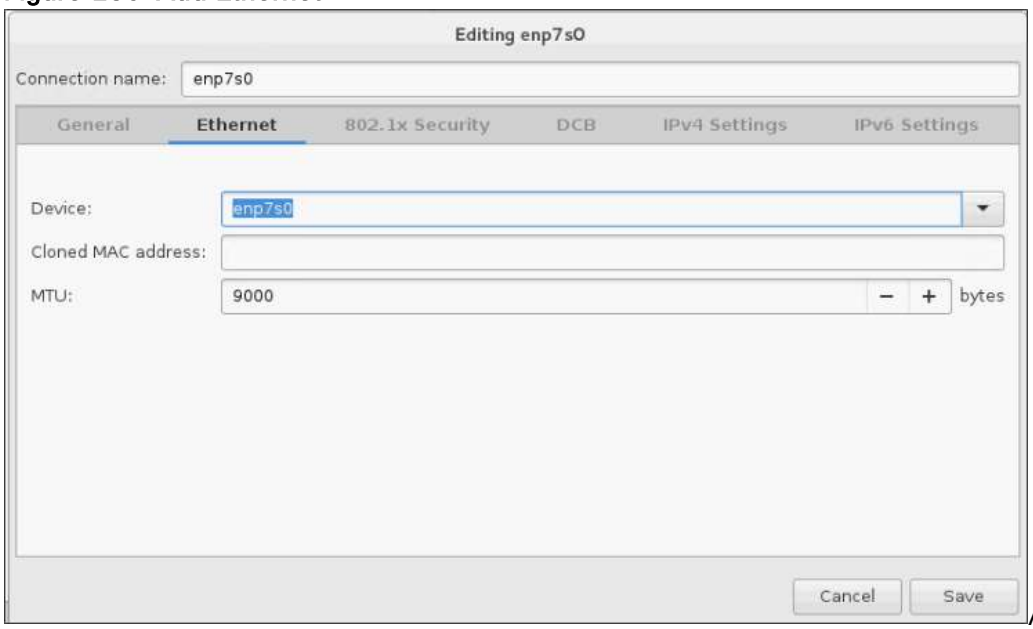
45. Type in the hostname as shown below. (Figure 135)

Figure 135 Add the Host Name



46. Click on Configure to open the Network Connectivity window. Click on Ethernet. (Figure 136)

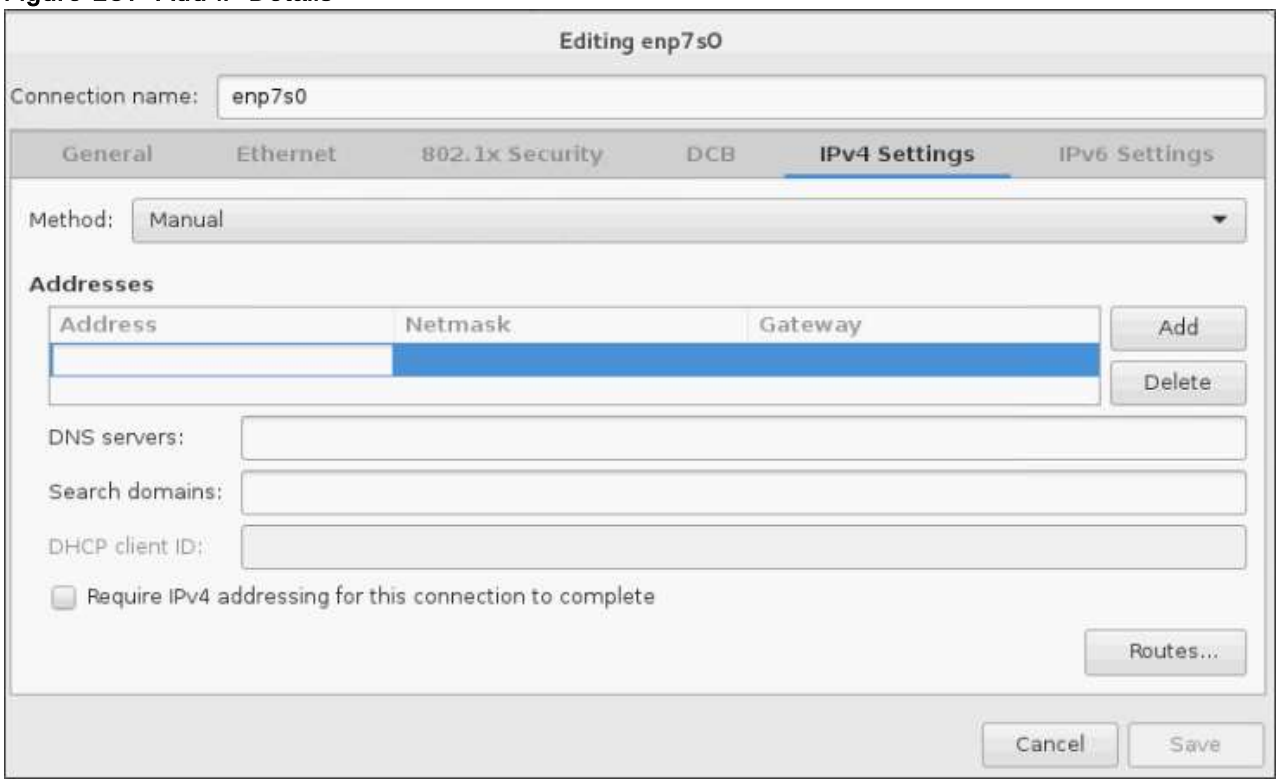
Figure 136 Add Ethernet



A

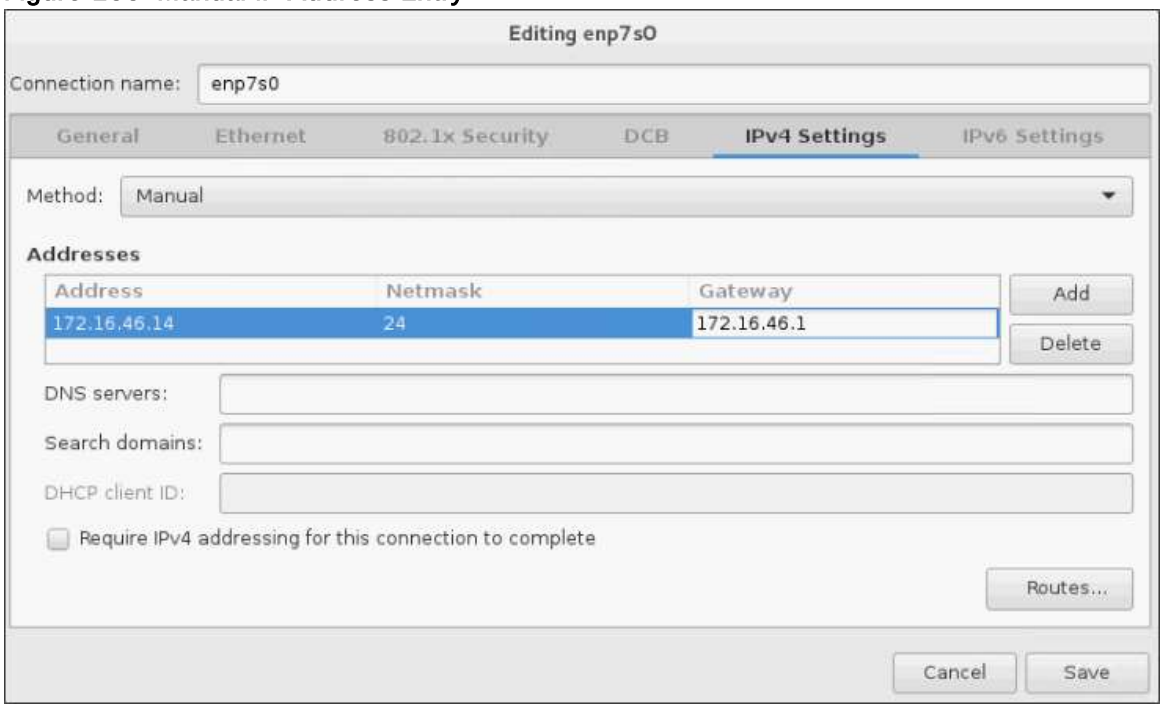
47. Click on IPv4 Settings and change the Method to Manual and click Add to enter the IP Address, Netmask and Gateway details. (Figure 137)

Figure 137 Add IP Details



48. Enter the desired IP address, Netmask and Gateway and click Save. (Figure 138)

Figure 138 Manual IP Address Entry



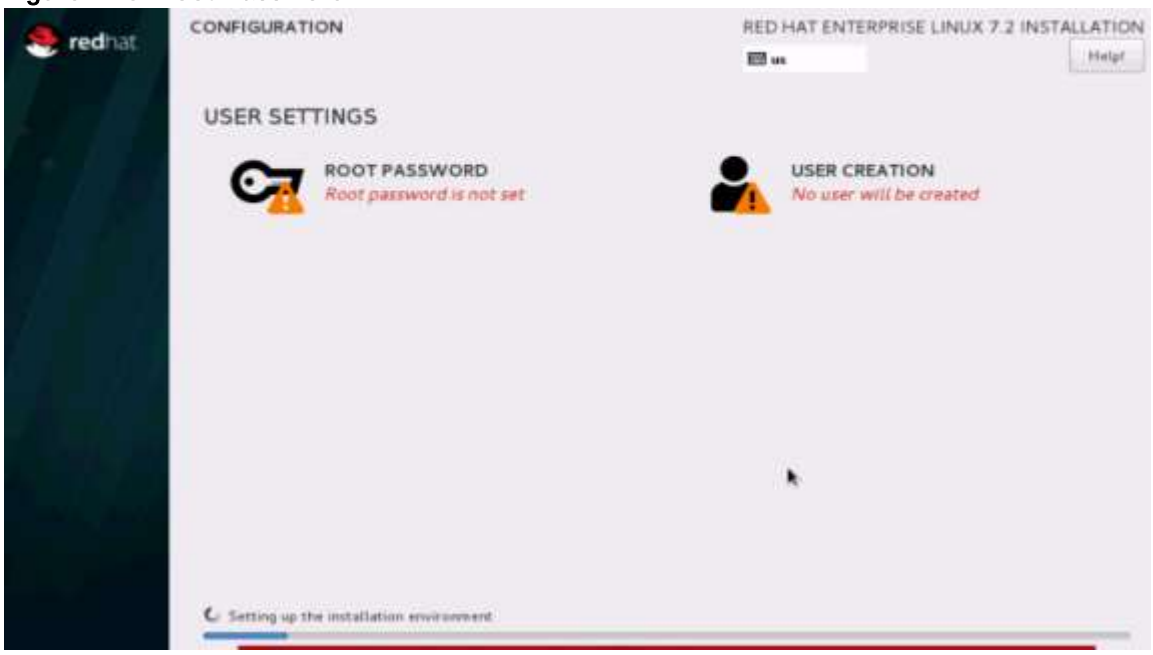
49. Click Save, update the hostname and turn Ethernet ON. Click Done to return to the main menu.

50. Click Begin Installation in the main menu. (Figure 139)

Figure 139 Begin Installation

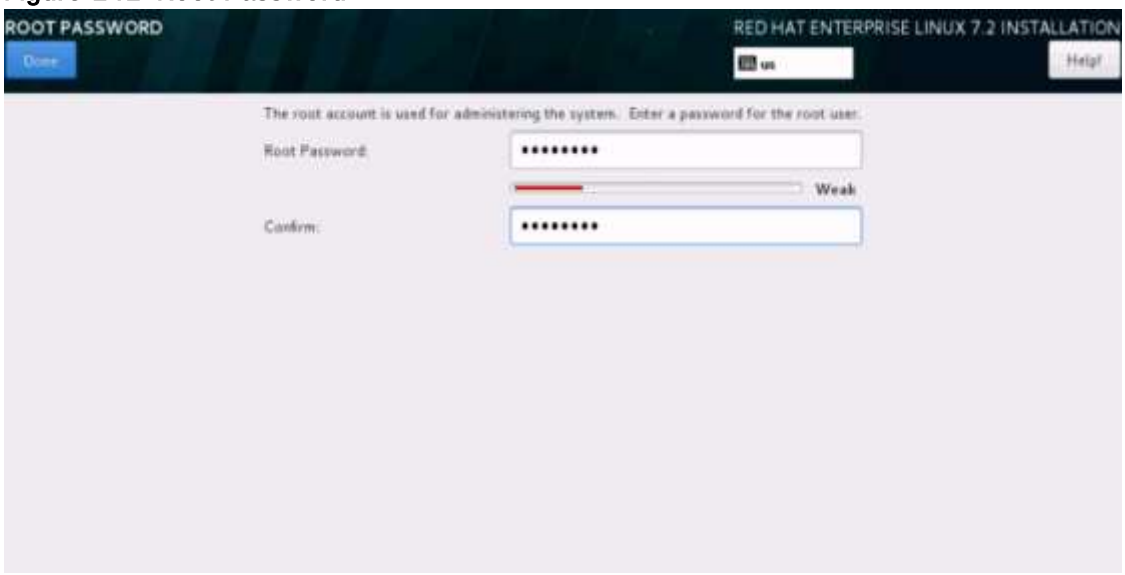
51. Select `Root Password` in the User Settings. (Figure 140)

Figure 140 Root Password



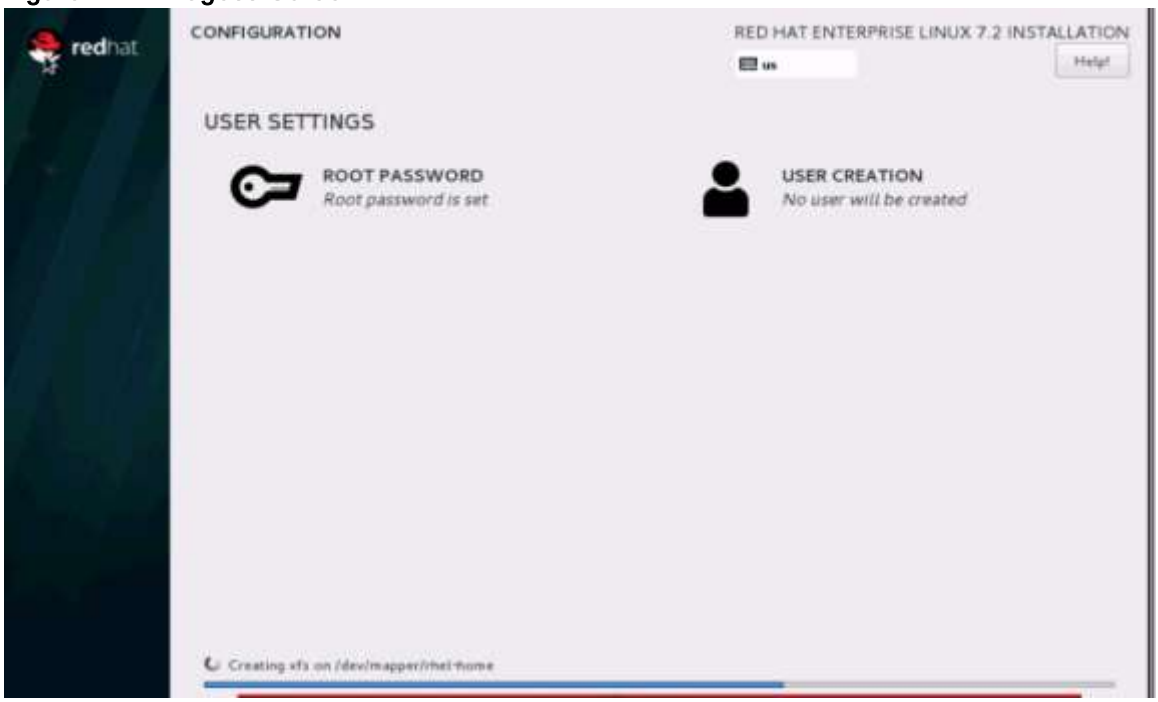
52. Enter the `Root Password` and click Done. (Figure 141)

Figure 141 Root Password



The Installation Progress window displays the process. (Figure 142)

Figure 142 Progess Screen



53. Once the installation is complete reboot the system.

54. Repeat steps 1 to 40 to install Red Hat Enterprise Linux 7.2 on rest of the Data Nodes.

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

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

Table 7 Hostnames and IP Addresses

Hostname	eth0
rhel1	172.16.46.11
rhel2	172.16.46.12
rhel3	172.16.46.13
rhel4	172.16.46.14
rhel1	172.16.46.15
rhel6	172.16.46.16
rhel7	172.16.46.17
rhel8	172.16.46.18
rhel9	172.16.46.19
rhel10	172.16.46.20
rhel11	172.16.46.21
rhel12	172.16.46.22
rhel13	172.16.46.23
rhel14	172.16.46.24
rhel15	172.16.46.25
rhel16	172.16.46.26
rhel17	172.16.46.27
rhel18	172.16.46.28
rhel19	172.16.46.29

Note: Cloudera does not support multi-homed configurations, so please assign only one network to each node.

Post OS Install Configuration

Choose one of the nodes of the cluster or a separate node as the Admin Node for management such as CDH installation, cluster parallel shell, creating a local Red Hat repo and others. In this document, we use rhel1 for this purpose.

Setting Up Password-less Login

To manage all of the clusters nodes from the admin node, password-less login needs to be setup. It assists in automating common tasks with clustershell (clush, a cluster wide parallel shell), and shell-scripts without having to use passwords.

Once Red Hat Linux is installed across all the nodes in the cluster, follow the steps below in order to enable password-less login across all the nodes.

1. Login to the Admin Node (rhel1).

```
#ssh 172.16.46.11
```

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

3. Download sshpass to the node connected to the internet and copy it to the admin node (rhel1) using the command

wget ftp://195.220.108.108/linux/dag/redhat/el6/en/x86_64/dag/RPMS/sshpass-1.05-1.el6.rf.x86_64.rpm

```
scp sshpass-1.05-1.el6.x86_64.rpm rhel1:/root/
```

4. Log in to the admin node and Install the rpm using the command

```
yum -y install sshpass-1.05-1.el6.x86_64.rpm
```

5. Create a file under /.ssh/config and enter the following lines

```
vi ~/.ssh/config

ServerAliveInterval 99
StrictHostKeyChecking no
```

6. Then 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-host's `.ssh/authorized_keys`

```
#for IP in {11..29}; do echo -n "$IP -> "; sshpass -p secret123 ssh-copy-id -i ~/.ssh/id_rsa.pub 172.16.46.$IP; done
```

Configuring /etc/hosts

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

To create the host file on the admin node, complete the following steps:

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

On Admin Node (rhel1)

```
#vi /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 \
localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 \ localhost6.localdomain6

172.16.46.11    rhel1

172.16.46.12    rhel2

172.16.46.13    rhel3

172.16.46.14    rhel4

172.16.46.15    rhel5

172.16.46.16    rhel6

172.16.46.17    rhel7

172.16.46.18    rhel8

172.16.46.19    rhel9

172.16.46.20    rhel10

172.16.46.21    rhel11

172.16.46.22    rhel12

172.16.46.23    rhel13
```

```
172.16.46.24    rhel14
172.16.46.25    rhel15
172.16.46.26    rhel16
172.16.46.27    rhel17
172.16.46.28    rhel18
172.16.46.29    rhel19
```

Creating a Red Hat Enterprise Linux (RHEL) 7.2 Local Repo

To create a repository using the 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.

1. Log on to rhel1. Create a directory that would contain the repository.

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

2. Create the mount directory for redhat iso.

```
#mkdir -p /mnt/rheliso
```

```
#mount -t iso9660 -o loop /root/rhel-server-7.2-x86_64-dvd.iso
/mnt/rheliso/
```

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

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

```
[root@rhel1 ~]# mkdir -p /var/www/html/rhelrepo
[root@rhel1 ~]# mkdir -p /mnt/rheliso
[root@rhel1 ~]# mount -t iso9660 -o loop /root/rhel-server-7.2-x86_64-dvd.iso /mnt/rheliso/
mount: /dev/loop0 is write-protected, mounting read-only
[root@rhel1 ~]# cp -r /mnt/rheliso/* /var/www/html/rhelrepo
```

4. Now create a .repo file to enable the use of the yum command.

```
#vi /var/www/html/rhelrepo/rheliso.repo
```

```
[rhel7.2]
```

```
name=Red Hat Enterprise Linux 7.2
```

```
baseurl=http://172.16.46.11/rhelrepo
```

```
gpgcheck=0
```

```
enabled=1
```

5. Now 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/
```

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

6. 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.

Note: This step is needed to install software on Admin Node (rhel1) using the repo (such as httpd, create-repo, etc.)

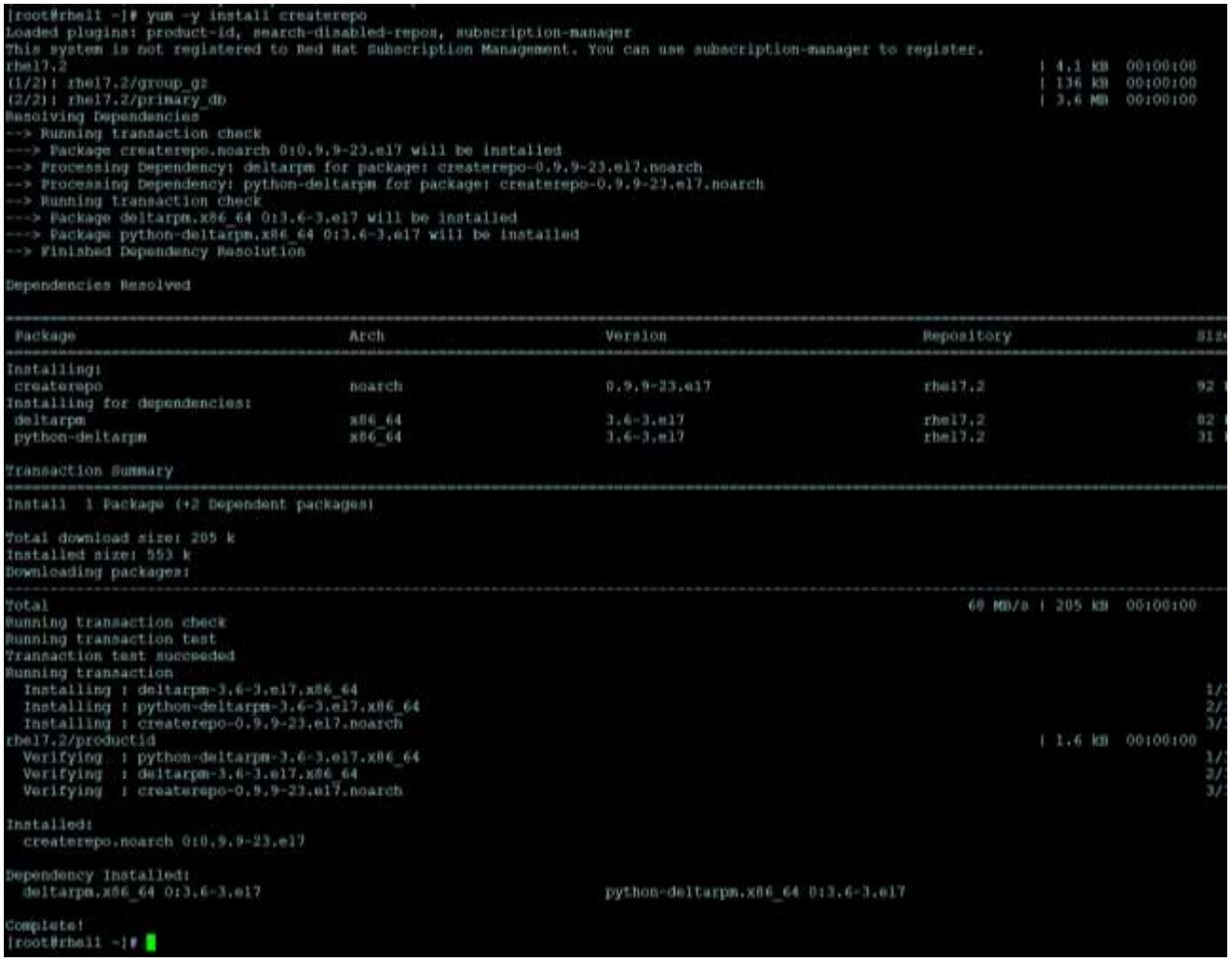
```
#vi /etc/yum.repos.d/rheliso.repo
[rhel7.2]
name=Red Hat Enterprise Linux 7.2
baseurl=file:///var/www/html/rhelrepo
gpgcheck=0
enabled=1
```

Creating the Red Hat Repository Database

To create a Red Hat Repository Database, complete the following 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
```

Setting up ClusterShell

ClusterShell (or clush) is the cluster-wide shell that runs commands on several hosts in parallel. To set up ClusterShell, complete the following steps:

1. From the system connected to the Internet download Cluster shell (clush) and copy and install it on rhel1. Cluster shell is available from EPEL (Extra Packages for Enterprise Linux) repository.

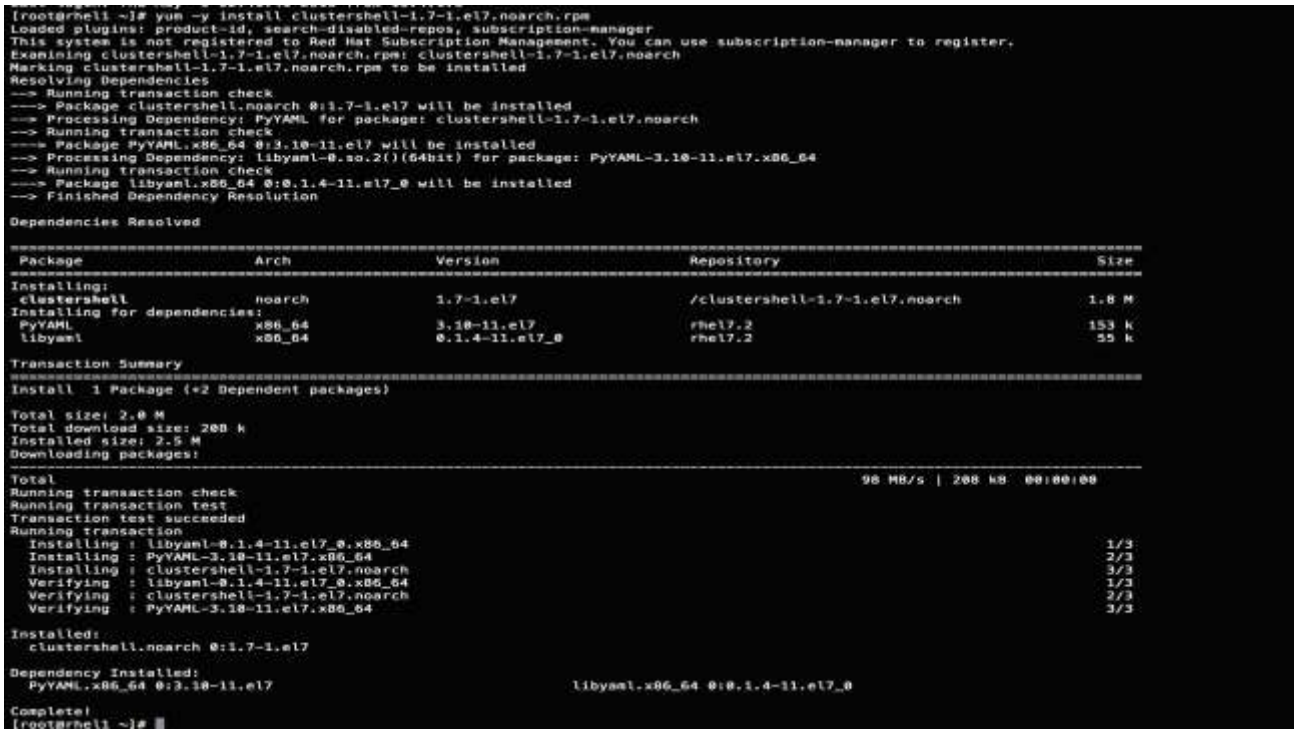
wget
ftp://ftp.pbone.net/mirror/ftp.sourceforge.net/pub/sourceforge/c/cl/clustershell/clustershell/1.7/clustershell-1.7-1.el7.noarch.rpm

```
[root@Rack10-JB tmp]# wget ftp://ftp.pbone.net/mirror/ftp.sourceforge.net/pub/sourceforge/c/cl/clusters
hell/clustershell/1.7/clustershell-1.7-1.el7.noarch.rpm
--2016-07-20 09:19:11--  ftp://ftp.pbone.net/mirror/ftp.sourceforge.net/pub/sourceforge/c/cl/clustershe
ll/clustershell/1.7/clustershell-1.7-1.el7.noarch.rpm
           => "clustershell-1.7-1.el7.noarch.rpm.1"
Resolving ftp.pbone.net... 85.14.85.4
Connecting to ftp.pbone.net|85.14.85.4|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD (1) /mirror/ftp.sourceforge.net/pub/sourceforge/c/cl/clustershell/cluster
shell/1.7 ... done.
==> SIZE clustershell-1.7-1.el7.noarch.rpm ... 371336
==> PASV ... done.    ==> RETR clustershell-1.7-1.el7.noarch.rpm ... done.
Length: 371336 (363K) (unauthoritative)

100%[======================================================================>] 371,336      348K/s   in 1.0s

2016-07-20 09:19:15 (348 KB/s) - "clustershell-1.7-1.el7.noarch.rpm.1" saved [371336]
```

```
#scp clustershell-1.7-1.el7.noarch.rpm rhel1:/root/
```

2. Login to rhel1 and install cluster shell.

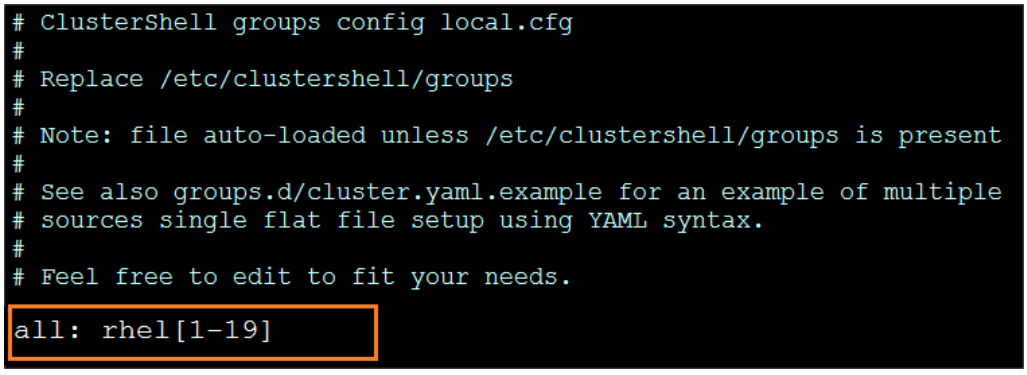
```
#yum -y install clustershell-1.71.el7.noarch.rpm
```



3. Edit /etc/clustershell/groups.d/local.cfg file to include hostnames for all the nodes of the cluster. This set of hosts is taken when running clush with the '-a' option.

4. For a 19 node cluster as in our CVD, set groups file as follows:

```
#vi /etc/clustershell/groups.d/local.cfg
```




```
all: rhel[1-19]
```

Note: For more information and documentation on ClusterShell, visit https://github.com/cea-hpc/clustershell/wiki/UserAndProgrammingGuide.

Installing httpd

Setting up RHEL repo on the admin node requires httpd. To set up RHEL repository on the admin node, complete the following steps:

5. 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
```

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

```
#vi /etc/httpd/conf/httpd.conf

ServerName 172.16.46.11:80
```

7. Start httpd:

```
#service httpd start

#chkconfig httpd on

Disabling the Linux Firewall
```

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

```
#clush –a –b “systemctl stop firewalld”

#clush –a –b “systemctl disable firewalld”
```

Disabling SELinux

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

1. To disable SELinux, edit /etc/selinux/config and change the SELINUX line to:

```
SELINUX=disabled.
```

2. To disable SELINUX on all nodes, use the following command:

```
#clush -a -b "sed –i 's/SELINUX=enforcing/SELINUX=disabled/g'
/etc/selinux/config"
```

```
[root@rhel1 ~]# clush -a -b "sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config "
```

```
#clush –a –b "setenforce 0"
```

Note: The command above may fail if SELinux is already disabled.

3. Reboot the machine to disable SELinux, if does not take effect. Check it using:

```
#clush –a –b sestatus
```

Set Up all Nodes to Use the RHEL Repository

Note: Based on this repo 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.

```
#clush –w rhel[2-19] -c /var/www/html/rhelrepo/rheliso.repo --
dest=/etc/yum.repos.d/
```

```
[root@rhel1 ~]# clush -w rhel[2-19] -c /var/www/html/rhelrepo/rheliso.repo --dest=/etc/yum.repos.d/
```

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

```
#clush –w rhel[2-19] –c /etc/hosts --dest=/etc/hosts
```

3. Purge the yum caches.

```
#clush –a –B yum clean all

#clush –a –B yum repolist
```

Note: While the suggested configuration is to disable SELinux, if for any reason SELinux needs to be enabled on the cluster, run the following to make sure that httpd is able to read the Yum repofiles.

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

Configuring DNS

This section details setting up DNS using dnsmasq as an example based on the /etc/hosts configuration setup in the earlier section.

To create the host file across all the nodes in the cluster, complete the following steps:

4. Disable Network manager on all nodes.

```
#clush –a –b service NetworkManager stop

#clush –a –b chkconfig NetworkManager off
```

5. Update /etc/resolv.conf file to point to Admin Node.

```
#vi /etc/resolv.conf

nameserver 172.16.46.11
```

Note: This step is needed to set up dnsmasq on the Admin node. Otherwise this file should be updated with the correct nameserver.

Note: Alternatively, #systemctl start NetworkManager.service can be used to start the service. #systemctl stop NetworkManager.service can be used to stop the service. Use #systemctl disable NetworkManager.service to stop a service from being automatically started at boot time.

6. Install and Start dnsmasq on Admin node.

```
#service dnsmasq start

#chkconfig dnsmasq on
```

7. Deploy /etc/resolv.conf from the admin node (rhel1) to all the nodes via the following clush command:

```
#clush -a -B -c /etc/resolv.conf
```

Note: A clush copy without –dest copies to the same directory location as the source-file directory.

8. Ensure DNS is working fine by running the following command on Admin node and any data-node

```
[root@rhel2 ~]# nslookup rhel1

Server:     172.16.46.11

Address:    172.16.46.11#53

Name: rhel1

Address: 172.16.46.11
```

Note: yum install –y bind-utils will need to be run for nslookup utility to run.

Upgrading the Cisco Network Driver for VIC1387

The latest Cisco Network driver is required for performance and updates. To download the latest drivers, go to the link below:

https://software.cisco.com/download/release.html?mdfid=283862063&release=2.0(13)&relind=AVAILABLE&flowid=25886&softwareid=283853158&rellifecycle=&reltype=latest

Note that the C-Series and S-Series servers use the same drivers.

1. In the ISO image, the required driver kmod-enic-2.3.0.30-rhel7u2.el7.x86_64.rpm can be located at \Network\Cisco\VIC\RHEL\RHEL7.2.

2. From a node connected to the Internet, download, extract and transfer kmod-enic-2.3.0.30-rhel7u2.el7.x86_64.rpm to rhel1 (admin node).

3. Install the rpm on all nodes of the cluster using the following clush commands. For this example the rpm is assumed to be in present working directory of rhel1.

```
[root@rhel1 ~]# clush -a -b -c kmod-enic-2.3.0.30-
rhel7u2.el7.x86_64.rpm

[root@rhel1 ~]# clush -a -b "rpm –ivh kmod-enic-2.3.0.30-
rhel7u2.el7.x86_64.rpm"
```

4. Ensure 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@rhel1 ~]# clush -a -B "modinfo enic | head -5"
```

```
[root@rhel1 ~]# modinfo enic
filename:       /lib/modules/2.6.32-573.el6.x86_64/extra/enic/enic.ko
version:        2.3.0.30
```

Also it is recommended to download the kmod-megaraid driver for higher performance , the RPM can be found in the same package at \Storage\LSI\Cisco_Storage_12G_SAS_RAID_controller\RHEL\RHEL7.2

Installing xfsprogs

From the admin node rhel1 run the command below to Install `xfsprogs` on all the nodes for xfs filesystem.

```
#clush -a -B yum -y install xfsprogs
```

```
[root@rhel1 ~]# clush -a -B yum -y install xfsprogs
```

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.

```
#clush –a –b "yum –y install ntp"
```

Note: 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 172.16.46.11

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 command in the admin node (rhel1)

```
#for SERVER in {12..29}; do scp /root/ntp.conf
172.16.46.$SERVER:/etc/ntp.conf; done
```

```
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
ntp.conf                                      100%  136     0.1KB/s   00:00
```

Note: Instead of the above for loop, this could be run as a clush command with "-w"option.

```
#clush -w rhel[2-19]  -b -c /root/ntp.conf --dest=/etc
```

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

```
#clush -a -b "service ntpd stop"

#clush -a -b "ntpdate rhel1"

#clush -a -b "service ntpd start"
```

5. Ensure restart of NTP daemon across reboots

```
#clush -a -b "systemctl enable ntpd"
```

Enabling 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.

To confirm that the service is properly configured, use one of the following commands:

```
#clush -B -a rsyslogd -v

#clush -B -a service rsyslog status
```

```
[root@rhel1 ~]# clush -B -a rsyslogd -v
---------------
rhel[1-19] (19)
---------------
rsyslogd 7.4.7, compiled with:
        FEATURE_REGEXP:                         Yes
        FEATURE_LARGEFILE:                      No
        GSSAPI Kerberos 5 support:              Yes
        FEATURE_DEBUG (debug build, slow code): No
        32bit Atomic operations supported:      Yes
        64bit Atomic operations supported:      Yes
        Runtime Instrumentation (slow code):    No
        uuid support:                           Yes

See http://www.rsyslog.com for more information.
```

Setting 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. Set the value to 64000 on every node.

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

1. For setting the ulimit on Redhat, edit /etc/security/limits.conf on admin node rhel1 and add the following lines:

```
root soft nofile 64000
```

```
root hard nofile 64000
```

```
[root@rhel1 ~]# cat /etc/security/limits.conf | grep 64000
root soft nofile 64000
root hard nofile 64000
```

2. Copy the /etc/security/limits.conf file from admin node (rhel1) to all the nodes using the following command.

```
#clush -a -b -c /etc/security/limits.conf --dest=/etc/security/
```

```
[root@rhel1 ~]# clush -a -b -c /etc/security/limits.conf --dest=/etc/security/
```

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

```
#%PAM-1.0
auth            sufficient      pam_rootOK.so
# Uncomment the following line to implicitly trust users in the "wheel"
group.
#auth           sufficient      pam_wheel.so trust use_uid
# Uncomment the following line to require a user to be in the "wheel"
group.
#auth           required        pam_wheel.so use_uid
auth            include         system-auth
account         sufficient      pam_succeed_if.so uid = 0 use_uid quiet
account         include         system-auth
password        include         system-auth
session         include         system-auth
session         optional        pam_xauth.so
```

Note: 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

Adjust the tcp_retries parameter for the system network to enable faster detection of failed nodes. Given the advanced networking features of UCS, this is a safe and recommended change (failures observed at the operating system layer are most likely serious rather than transitory). On each node, setting 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 add the following lines:

```
net.ipv4.tcp_retries2=5
```

2. Copy the /etc/sysctl.conf file from admin node (rhel1) to all the nodes using the following command:

```
#clush -a -b -c /etc/sysctl.conf --dest=/etc/
```

3. Load the settings from default sysctl file /etc/sysctl.conf by running.

```
#clush -B -a sysctl -p
```

```
[root@rhel1 ~]# clush -B -a sysctl -p
---------------
rhel[1-16] (16)
---------------
net.ipv4.tcp_retries2 = 5
```

Disable Swapping

1. To reduce Swapping, run the following on all nodes. The variable vm.swappiness defines how often swap should be used, 60 is the default.

```
#clush -a -b " echo 'vm.swappiness=1' >> /etc/sysctl.conf"
```

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

```
#clush -a -b "sysctl -p"
```

Disable Transparent Huge Pages

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

1. To run the following commands for every reboot, copy these commands to /etc/rc.local so they are executed automatically for every reboot.

```
#clush -a -b "echo never > /sys/kernel/mm/transparent_hugepage/enabled”
#clush -a -b "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 the following file to each node:

```
#clush -a -b -c /root/thp_disable
```

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

```
#clush -a -b "cat /root/thp_disable >> /etc/rc.d/rc.local"
```

5. Change the permissions of the rc.local file

```
#chmod u+x /etc/rc.d/rc.local
```

6. Start the rc.local service if not started already

```
#systemctl start rc-local
```

Disable IPv6 Defaults

1. Disable IPv6 as the addresses used are IPv4.

```
#clush -a -b "echo 'net.ipv6.conf.all.disable_ipv6 = 1' >>
/etc/sysctl.conf"

#clush -a -b "echo 'net.ipv6.conf.default.disable_ipv6 = 1' >>
/etc/sysctl.conf"

#clush -a -b "echo 'net.ipv6.conf.lo.disable_ipv6 = 1' >>
/etc/sysctl.conf"
```

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

```
#clush -a -b "sysctl -p"
```

```
[root@rhel1 ~]# clush -a -b sysctl -p
---------------
rhel[1-19] (19)
---------------
net.ipv4.tcp_retries2 = 5
vm.swappiness = 1
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Configuring RAID1 on Hadoop Management Nodes

Configure non-OS disk drives as RAID1 using StorCli commands as described below. The first four disk drives are going to be part of a single RAID1 volume. This volume will be used for HDFS Metadata. This section describes in detail the RAID configuration of disk drives for HDFS Name Node Metadata.

3. To download storcli go to:

http://docs.avagotech.com/docs/1.19.04_StorCLI.zip

4. Extract the zip file and copy storcli-1.19.04-1.noarch.rpm from the linux directory.

5. Download storcli and its dependencies and transfer to Admin node.

```
#scp storcli-1.19.04-1.noarch.rpm rhel1:/root/
```

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

```
#clush -a -b -c /root/ storcli-1.19.04-1.noarch.rpm --dest=/root/
```

7. Run the following command to install storcli on all the nodes

```
#clush -a -b "rpm -ivh storcli-1.19.04-1.noarch.rpm"
```

8. Run the below command to copy storcli64 to root directory.

```
#cd /opt/MegaRAID/storcli/

#cp storcli64 /root/
```

```
[root@rhel1 ~]# cd /opt/MegaRAID/storcli/
[root@rhel1 storcli]# ls
storcli64
```

9. Copy storcli64 to all the nodes:

```
#clush -a -b -c /root/storcli64 --dest=/root/
```

10. Run the following script as root user on rhel1 to rhel3 to create the virtual drives for the management nodes.

```
#vi /root/raid1.sh

./storcli64 -cfgldadd r1[$1:1,$1:2,$1:3,$1:4] wb ra nocachedbadbbu
strpsz1024 -a0
```

The script above requires Enclosure ID as a parameter.

11. Run the following command to get the enclosure id.

```
#./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' |
sort | uniq -c | awk '{print $2}'

#chmod 755 raid1.sh
```

12. Run MegaCli script as follows:./

```
#./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
```

Configuring the Virtual Drive (RAID10) for DB Filesystem on Hadoop Management Node

This section describes configuring the remaining 8 disk drives as a RAID10 DB file system with read-ahead cache enabled and write cache enabled while battery is present.

1. Create a script named raid10.sh on the admin node and copy it over to all Master/Management Nodes.

```
vi /root/raid10.sh
```

2. Paste the following contents into the file and save it.

```
/opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid10 drives=$1:5-12
pdperarray=4 WB ra direct Strip=1024
```

3. Please add/remove drives based on your configuration.

Note: Do not execute this script on the Data Nodes.

Note: This script must be executed manually on each of the Management nodes. The script takes the EnclosureID as Input, which would be different on different Management servers.

4. Change the mode to include execution privileges.

```
chmod +x /root/raid10.sh
```

5. Copy the script over to all the Management nodes.

6. The script above requires enclosure ID as a parameter. Run the following command to get EnclosureID on each Management node.

```
/opt/MegaRAID/storcli/storcli64 pdlist -a0 | grep Enc | grep -v 252 |
awk '{print $4}' | sort | uniq -c | awk '{print $2}'
```

7. Run the script to create a single RAID10 volume as follows:

```
./raid10.sh <EnclosureID>
```

Note: The command above will not override any existing configuration. To clear and reconfigure existing configurations refer to Embedded MegaRAID Software Users Guide available at www.lsi.com.

Cloudera recommends the following disk configuration for the master nodes.

- At least 10 physical disks in the following configuration
- 2 x RAID1 OS (Root disk)
- 4 x RAID 10 (DB filesystems)
- 2 x RAID 1 HDFS NameNode metadata
- 1 x JBOD - ZooKeeper
- 1 x JBOD - Quorum JournalNode

Configuring Data Drives on Data Nodes

Configure non-OS disk drives as individual RAID0 volumes using the StorCli command as described below. These volumes will be used for HDFS Data.

1. To create virtual drives with individual RAID 0 configurations on all the data nodes, from the admin node, issue the following command:

```
#clush –w rhel[4-19] -B ./storcli64 -cfgeachdskraid0 WB RA direct
NoCachedBadBBU strpsz1024 –a0

WB: Write back

RA: Read Ahead

NoCachedBadBBU: Do not write cache when the BBU is bad.

Strpsz1024: Strip Size of 1024K
```

Note: The command above will not override existing configurations. To clear and re-configure existing configurations refer to Embedded MegaRAID Software Users Guide available at www.lsi.com.

Configuring the Filesystem for NameNodes and DataNodes

The following script will format and mount the available volumes on each node whether it is a Namenode or a Data node. The OS boot partition is skipped. All drives are mounted based on their UUID as /data/disk1, /data/disk2, and so on.

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

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.

Note: The script assumes there are no partitions already existing on the data volumes. If there are partitions, delete them before running the script. This process is documented in the "Note" section at the end of the section.

```
#vi /root/driveconf.sh

#!/bin/bash

 [[ "-x" == "${1}" ]] && set -x && set -v && shift 1

count=1

for X in /sys/class/scsi_host/host?/scan

do

echo '- - -' > ${X}

done

for X in /dev/sd?

do

list+=$(echo $X " ")

done


for X in /dev/sd??

do

list+=$(echo $X " ")

done


for X in $list

do

echo "========"

echo $X

echo "========"

if [[ -b ${X} && `/sbin/parted -s ${X} print quit|/bin/grep -c boot` -
ne 0

]]

then

echo "$X bootable - skipping."

continue

else

Y=${X##*/}1

echo "Formatting and Mounting Drive => ${X}"
```

```
/sbin/mkfs.xfs -f ${X}

(( $? )) && continue

#Identify UUID

UUID=`blkid ${X} | cut -d " " -f2 | cut -d "=" -f2 | sed 's/"//g'`

/bin/mkdir  -p  /data/disk${count}

(( $? )) && continue

echo "UUID of ${X} = ${UUID}, mounting ${X} using UUID on
/data/disk${count}"

/bin/mount  -t xfs  -o inode64,noatime,nobarrier -U ${UUID}
/data/disk${count}

(( $? )) && continue

echo "UUID=${UUID} /data/disk${count} xfs inode64,noatime,nobarrier 0
0" >> /etc/fstab

((count++))

fi

done
```

2. Copy driveconf.sh to all the nodes with the following command:

```
#chmod 755 /root/driveconf.sh

#clush -a -B -c /root/driveconf.sh
```

3. From the admin node run the following script across all data nodes:

```
#clush -a -B /root/driveconf.sh
```

4. To list the partitions and mount points, run the following from the admin node

```
#clush -a -B df -h

#clush -a -B mount

#clush -a -B cat /etc/fstab
```

Note: In-case there is a need to delete any partitions, it can be done so using the following.

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

6. umount the drive for which the partition is to be deleted, and run fdisk to delete it as shown below.

Note: Care should be taken **not to delete the OS partition** as 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 describes 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.

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 C3260 Storage Server 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: '; dmidecode |grep -c \
'^[[:space:]]*Locator:'"

clush -a -B "echo -n 'DIMM count is: '; dmidecode | grep \Size| grep -c
"MB""

clush -a -B " 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 "ls /sys/class/net | grep ^enp | \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: '; cpupower frequency-info \
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 \
De-fined!}'

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'

exit
```

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
```

Installing Cloudera

Cloudera Enterprise is an enterprise grade, hardened Hadoop distribution. Cloudera Enterprise offers Apache Hadoop and its ecosystem into a single tested and certified product. It offers the latest innovations from the open source community with the testing and quality expected from enterprise quality software.

Pre-Requisites for Cloudera Enterprise Installation

This section details the prerequisites for Cloudera Enterprise installation such as setting up Cloudera Enterprise Repo.

Cloudera Manager Repository

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

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

2. Download Cloudera Manager Repository.

```
#cd /tmp/clouderarepo/
```

```
#wget http:/ /archive.cloudera.com/cm5/redhat/7/x86_64/cm/cloudera-manager.repo
```

```
[root@Rack10-JB clouderarepo]# wget http://archive.cloudera.com/cm5/redhat/7/x86_64/cm/cloudera-manager
.repo
--2016-07-21 10:05:17--  http://archive.cloudera.com/cm5/redhat/7/x86_64/cm/cloudera-manager.repo
Resolving archive.cloudera.com... 151.101.52.167
Connecting to archive.cloudera.com|151.101.52.167|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 290
Saving to: "cloudera-manager.repo.1"

100%[======================================================>] 290         --.-K/s   in 0s

2016-07-21 10:05:17 (49.0 MB/s) - "cloudera-manager.repo.1" saved [290/290]
```

```
# reposync --config=./cloudera-manager.repo --repoid=cloudera-manager
```

```
[root@Rack10-JB clouderarepo]# reposync --config=./cloudera-manager.repo --repoid=cloudera-manager
[cloudera-manager: 1     of 7     ] Downloading RPMS/x86_64/cloudera-manager-agent-5.8.0-1.cm580.p0.42.
el7.x86_64.rpm
cloudera-manager-agent-5.8.0-1.cm580.p0.42.el7.x86_64.rpm                    | 7.8 MB     00:00
[cloudera-manager: 2     of 7     ] Downloading RPMS/x86_64/cloudera-manager-daemons-5.8.0-1.cm580.p0.4
2.el7.x86_64.rpm
cloudera-manager-daemons-5.8.0-1.cm580.p0.42.el7.x86_64.rpm                  | 515 MB     00:15
[cloudera-manager: 3     of 7     ] Downloading RPMS/x86_64/cloudera-manager-server-5.8.0-1.cm580.p0.42
.el7.x86_64.rpm
cloudera-manager-server-5.8.0-1.cm580.p0.42.el7.x86_64.rpm                   | 8.3 kB     00:00
[cloudera-manager: 4     of 7     ] Downloading RPMS/x86_64/cloudera-manager-server-db-2-5.8.0-1.cm580.
p0.42.el7.x86_64.rpm
cloudera-manager-server-db-2-5.8.0-1.cm580.p0.42.el7.x86_64.rpm              | 9.9 kB     00:00
[cloudera-manager: 5     of 7     ] Downloading RPMS/x86_64/enterprise-debuginfo-5.8.0-1.cm580.p0.42.el
7.x86_64.rpm
enterprise-debuginfo-5.8.0-1.cm580.p0.42.el7.x86_64.rpm                      |  29 MB     00:01
[cloudera-manager: 6     of 7     ] Downloading RPMS/x86_64/jdk-6u31-linux-amd64.rpm
jdk-6u31-linux-amd64.rpm                                                     |  68 MB     00:02
[cloudera-manager: 7     of 7     ] Downloading RPMS/x86_64/oracle-j2sdk1.7-1.7.0+update67-1.x86_64.rpm
oracle-j2sdk1.7-1.7.0+update67-1.x86_64.rpm                                  | 135 MB     00:05
```

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

3. Run the following command to move the RPMs

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

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

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

```
#cd /var/www/html/clouderarepo/

#createrepo --baseurl http://172.16.46.11/clouderarepo/cloudera-
manager/

/var/www/html/clouderarepo/cloudera-manager
```

Note: Visit http://172.16.46.11/clouderarepo/ to verify the files.

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

```
#vi /var/www/html/clouderarepo/cloudera-manager/cloudera-manager.repo

[cloudera-manager]

name=Cloudera Manager

baseurl=http://172.16.46.11/clouderarepo/cloudera-manager/

gpgcheck=0

enabled=1
```

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

```
#cp /var/www/html/clouderarepo/cloudera-manager/cloudera-manager.repo
/etc/yum.repos.d/
```

8. From the admin node copy the repo files to /etc/yum.repos.d/ of all the nodes of the cluster.

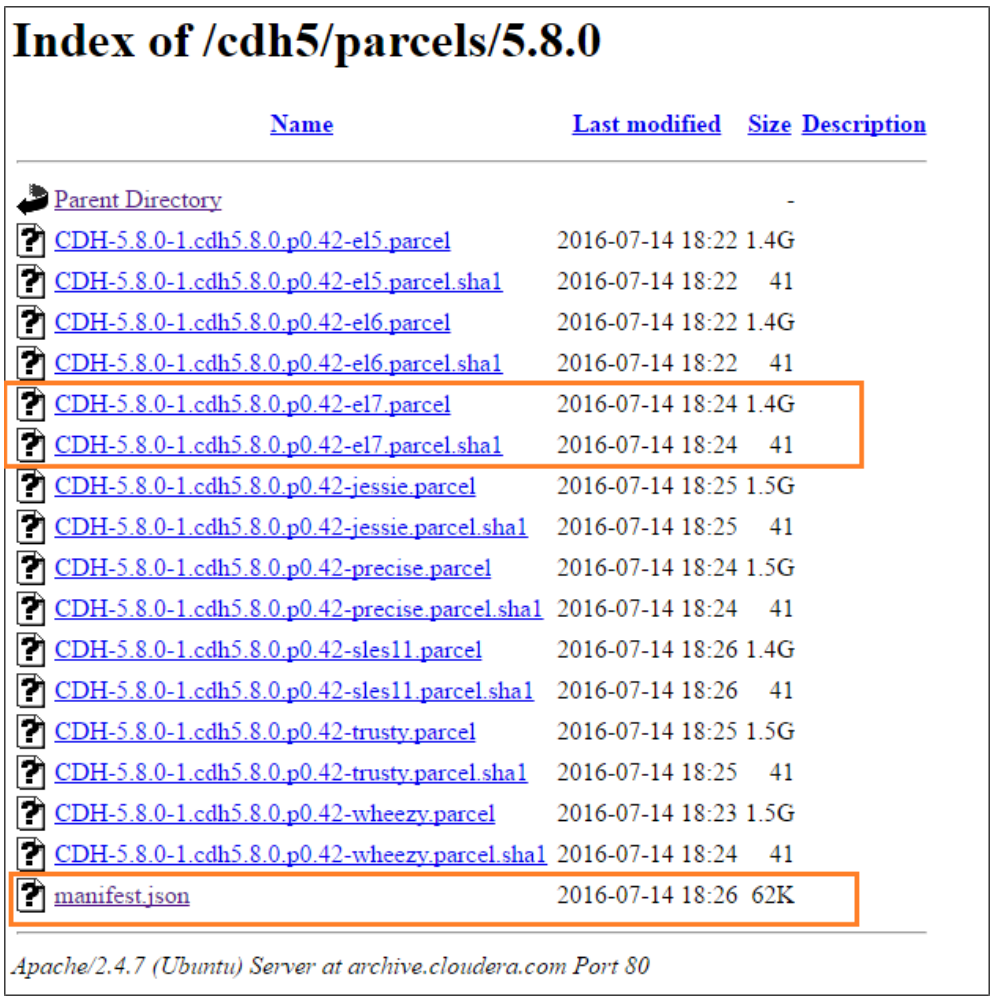
```
#clush -a -B -c /etc/yum.repos.d/cloudera-manager.repo
```

Setting up the Local Parcels for Cloudera Enterprise 5.8.0

From a host connected the internet, download the appropriate Cloudera Enterprise 5.8.0 parcels that are meant for RHEL7.2 from the URL: http://archive.cloudera.com/cdh5/parcels/ and place them in the directory "/var/www/html/CDH5.8.0_Parcel" of the Admin node.

The following list shows the relevant files for RHEL7.2, as shown in the figure below:

- CDH-5.8.0-1.cdh5.8.0.p0.42-el7.parcel
- CDH-5.8.0-1.cdh5.8.0.p0.42-el7.parcel.sha1 and
- manifest.json



Index of /cdh5/parcels/5.8.0

Name	Last modified	Size	Description
Parent Directory		-	
CDH-5.8.0-1.cdh5.8.0.p0.42-el5.parcel	2016-07-14 18:22	1.4G	
CDH-5.8.0-1.cdh5.8.0.p0.42-el5.parcel.sha1	2016-07-14 18:22	41	
CDH-5.8.0-1.cdh5.8.0.p0.42-el6.parcel	2016-07-14 18:22	1.4G	
CDH-5.8.0-1.cdh5.8.0.p0.42-el6.parcel.sha1	2016-07-14 18:22	41	
CDH-5.8.0-1.cdh5.8.0.p0.42-el7.parcel	2016-07-14 18:24	1.4G	
CDH-5.8.0-1.cdh5.8.0.p0.42-el7.parcel.sha1	2016-07-14 18:24	41	
CDH-5.8.0-1.cdh5.8.0.p0.42-jessie.parcel	2016-07-14 18:25	1.5G	
CDH-5.8.0-1.cdh5.8.0.p0.42-jessie.parcel.sha1	2016-07-14 18:25	41	
CDH-5.8.0-1.cdh5.8.0.p0.42-precise.parcel	2016-07-14 18:24	1.5G	
CDH-5.8.0-1.cdh5.8.0.p0.42-precise.parcel.sha1	2016-07-14 18:24	41	
CDH-5.8.0-1.cdh5.8.0.p0.42-sles11.parcel	2016-07-14 18:26	1.4G	
CDH-5.8.0-1.cdh5.8.0.p0.42-sles11.parcel.sha1	2016-07-14 18:26	41	
CDH-5.8.0-1.cdh5.8.0.p0.42-trusty.parcel	2016-07-14 18:25	1.5G	
CDH-5.8.0-1.cdh5.8.0.p0.42-trusty.parcel.sha1	2016-07-14 18:25	41	
CDH-5.8.0-1.cdh5.8.0.p0.42-wheezy.parcel	2016-07-14 18:23	1.5G	
CDH-5.8.0-1.cdh5.8.0.p0.42-wheezy.parcel.sha1	2016-07-14 18:24	41	
manifest.json	2016-07-14 18:26	62K	

Apache/2.4.7 (Ubuntu) Server at archive.cloudera.com Port 80

Downloading Parcels

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/CDH5.8.0_Parcel
```

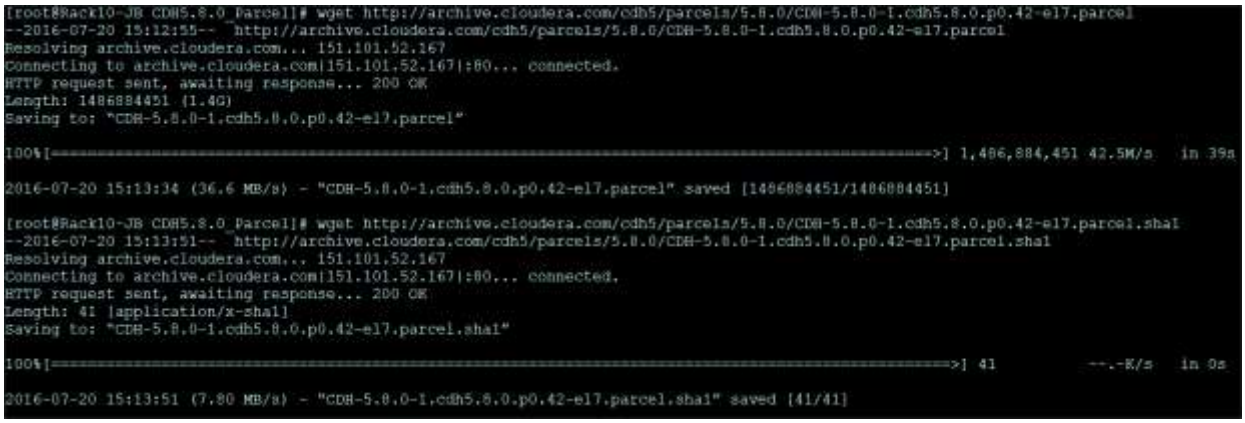
2. Download parcels:

#cd /tmp/CDH5.8.0_Parcels

wget http://archive.cloudera.com/cdh5/parcels/5.8.0/CDH-5.8.0-1.cdh5.8.0.p0.42-el7.parcel

wget http://archive.cloudera.com/cdh5/parcels/5.8.0/CDH-5.8.0-1.cdh5.8.0.p0.42-el7.parcel.sha1

wget http://archive.cloudera.com/cdh5/parcels/5.8.0/manifest.json



3. Now edit the /tmp/clouderarepo/CDH5.8_Parcels/manifest.json file and remove the scripts that are not meant for RHEL7.2. Below is that script which can be copy and pasted.

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

```
[root@rhel1 CDH5.8.0_Parcel]# cat manifest.json
{
 "lastUpdated": 14685205130000,
 "parcels": [
     {
         "parcelName": "CDH-5.8.0-1.cdh5.8.0.p0.42-el7.parcel",
```

```
"components": [
    {
        "pkg_version": "0.7.0+cdh5.8.0+0",
        "pkg_release": "1.cdh5.8.0.p0.73",
        "name": "bigtop-tomcat",
        "version": "6.0.44-cdh5.8.0"
    },
    {
        "pkg_version": "0.11.0+cdh5.8.0+91",
        "pkg_release": "1.cdh5.8.0.p0.77",
        "name": "crunch",
        "version": "0.11.0-cdh5.8.0"
    },
    {
        "pkg_version": "1.6.0+cdh5.8.0+50",
        "pkg_release": "1.cdh5.8.0.p0.75",
        "name": "flume-ng",
        "version": "1.6.0-cdh5.8.0"
    },
    {
        "pkg_version": "2.6.0+cdh5.8.0+1601",
        "pkg_release": "1.cdh5.8.0.p0.93",
        "name": "hadoop-0.20-mapreduce",
        "version": "2.6.0-cdh5.8.0"
    },
    {
        "pkg_version": "2.6.0+cdh5.8.0+1601",
        "pkg_release": "1.cdh5.8.0.p0.93",
        "name": "hadoop",
        "version": "2.6.0-cdh5.8.0"
    },
    {
```

```
        "pkg_version": "2.6.0+cdh5.8.0+1601",
        "pkg_release": "1.cdh5.8.0.p0.93",
        "name": "hadoop-hdfs",
        "version": "2.6.0-cdh5.8.0"
    },
    {
        "pkg_version": "2.6.0+cdh5.8.0+1601",
        "pkg_release": "1.cdh5.8.0.p0.93",
        "name": "hadoop-httpfs",
        "version": "2.6.0-cdh5.8.0"
    },
    {
        "pkg_version": "2.6.0+cdh5.8.0+1601",
        "pkg_release": "1.cdh5.8.0.p0.93",
        "name": "hadoop-kms",
        "version": "2.6.0-cdh5.8.0"
    },
    {
        "pkg_version": "2.6.0+cdh5.8.0+1601",
        "pkg_release": "1.cdh5.8.0.p0.93",
        "name": "hadoop-mapreduce",
        "version": "2.6.0-cdh5.8.0"
    },
    {
        "pkg_version": "2.6.0+cdh5.8.0+1601",
        "pkg_release": "1.cdh5.8.0.p0.93",
        "name": "hadoop-yarn",
        "version": "2.6.0-cdh5.8.0"
    },
    {
        "pkg_version": "1.2.0+cdh5.8.0+160",
        "pkg_release": "1.cdh5.8.0.p0.80",
```

```
    "name": "hbase",
    "version": "1.2.0-cdh5.8.0"
},
{
    "pkg_version": "1.5+cdh5.8.0+64",
    "pkg_release": "1.cdh5.8.0.p0.75",
    "name": "hbase-solr",
    "version": "1.5-cdh5.8.0"
},
{
    "pkg_version": "1.1.0+cdh5.8.0+610",
    "pkg_release": "1.cdh5.8.0.p0.77",
    "name": "hive",
    "version": "1.1.0-cdh5.8.0"
},
{
    "pkg_version": "1.1.0+cdh5.8.0+610",
    "pkg_release": "1.cdh5.8.0.p0.77",
    "name": "hive-hcatalog",
    "version": "1.1.0-cdh5.8.0"
},
{
    "pkg_version": "3.9.0+cdh5.8.0+2512",
    "pkg_release": "1.cdh5.8.0.p0.88",
    "name": "hue",
    "version": "3.9.0-cdh5.8.0"
},
{
    "pkg_version": "2.6.0+cdh5.8.0+0",
    "pkg_release": "1.cdh5.8.0.p0.111",
    "name": "impala",
    "version": "2.6.0-cdh5.8.0"
```

```
},
{
    "pkg_version": "1.0.0+cdh5.8.0+136",
    "pkg_release": "1.cdh5.8.0.p0.73",
    "name": "kite",
    "version": "1.0.0-cdh5.8.0"
},
{
    "pkg_version": "1.0.0+cdh5.8.0+0",
    "pkg_release": "1.cdh5.8.0.p0.73",
    "name": "llama",
    "version": "1.0.0-cdh5.8.0"
},
{
    "pkg_version": "0.9+cdh5.8.0+27",
    "pkg_release": "1.cdh5.8.0.p0.71",
    "name": "mahout",
    "version": "0.9-cdh5.8.0"
},
{
    "pkg_version": "4.1.0+cdh5.8.0+291",
    "pkg_release": "1.cdh5.8.0.p0.83",
    "name": "oozie",
    "version": "4.1.0-cdh5.8.0"
},
{
    "pkg_version": "1.5.0+cdh5.8.0+174",
    "pkg_release": "1.cdh5.8.0.p0.71",
    "name": "parquet",
    "version": "1.5.0-cdh5.8.0"
},
{
```

```
    "pkg_version": "0.12.0+cdh5.8.0+83",
    "pkg_release": "1.cdh5.8.0.p0.71",
    "name": "pig",
    "version": "0.12.0-cdh5.8.0"
},
{
    "pkg_version": "1.5.1+cdh5.8.0+244",
    "pkg_release": "1.cdh5.8.0.p0.83",
    "name": "sentry",
    "version": "1.5.1-cdh5.8.0"
},
{
    "pkg_version": "4.10.3+cdh5.8.0+423",
    "pkg_release": "1.cdh5.8.0.p0.79",
    "name": "solr",
    "version": "4.10.3-cdh5.8.0"
},
{
    "pkg_version": "1.6.0+cdh5.8.0+205",
    "pkg_release": "1.cdh5.8.0.p0.74",
    "name": "spark",
    "version": "1.6.0-cdh5.8.0"
},
{
    "pkg_version": "1.99.5+cdh5.8.0+38",
    "pkg_release": "1.cdh5.8.0.p0.72",
    "name": "sqoop2",
    "version": "1.99.5-cdh5.8.0"
},
{
    "pkg_version": "1.4.6+cdh5.8.0+65",
    "pkg_release": "1.cdh5.8.0.p0.69",
```

```
                    "name": "sqoop",
                    "version": "1.4.6-cdh5.8.0"
                },
                {
                    "pkg_version": "0.9.0+cdh5.8.0+17",
                    "pkg_release": "1.cdh5.8.0.p0.68",
                    "name": "whirr",
                    "version": "0.9.0-cdh5.8.0"
                },
                {
                    "pkg_version": "3.4.5+cdh5.8.0+94",
                    "pkg_release": "1.cdh5.8.0.p0.76",
                    "name": "zookeeper",
                    "version": "3.4.5-cdh5.8.0"
                }
            ],
            "replaces": "IMPALA, SOLR, SPARK",
            "hash": "26f281689fc24bde3ed0a34fb895417a88834fa3"
        }
]
 }
```


4. Copy /tmp/CDH5.8.0_Parcels to the admin node (rhel1)

```
#scp -r /tmp/CDH5.8.0_Parcels/ rhel1:/var/www/html/
```

5. Verify that these files are accessible by visiting the URL http://172.16.46.11/CDH5.8.0_Parcels/ in admin node.

Setting Up the MariaDB Database for Cloudera Manager

- Install the MariaDB Server
- Configure and Start the MariaDB Server
- Install the MariaDB/MySQL JDBC Driver

- Create Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server

Installing the MariaDB Server

To use a MariaDB database, complete the following steps:

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

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

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

```
#systemctl stop mariadb.service
```

3. Move the old InnoDB log if exists.

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

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

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

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

```
[root@rhel1 ~]# vi /etc/my.cnf
[root@rhel1 ~]# cat /etc/my.cnf
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
transaction-isolation = READ-COMMITTED
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

# InnoDB settings
innodb_flush_method = O_DIRECT

max_connections = 550
```

Note: The max_connections need to be increased based on number of nodes and applications. Please follow the recommendations as mentioned in the Cloudera document http://www.cloudera.com/documentation/enterprise/latest/topics/install_cm_mariadb.html - install_cm_mariadb_config

6. Ensure MySQL Server starts at boot:

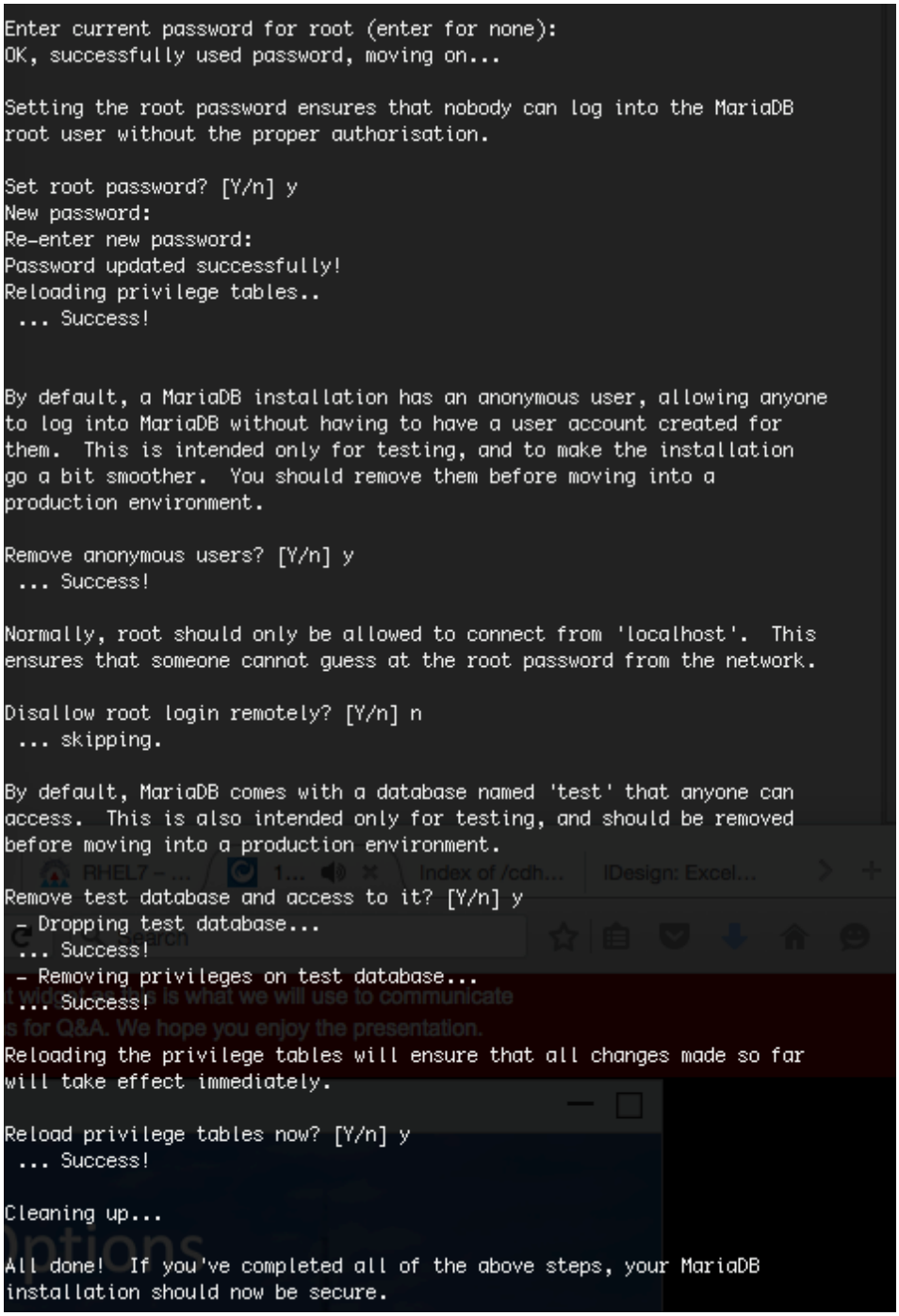
```
#systemctl enable mariadb.service
```

7. Start the MySQL Server:

```
#systemctl start mariadb.service
```

8. Set the MySQL root password on admin node (rhel1)

```
#cd /usr/bin/
#mysql_secure_installation
```



Installing the MySQL JDBC Driver

Install the JDBC driver on the Cloudera Manager Server host, as well as hosts which run the Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server roles.

1. From a host connected to the Internet, download the MySQL JDBC Driver and transfer it to the admin node. Download the MySQL JDBC driver from the URL http://www.mysql.com/downloads/connector/j/5.1.html

2. Copy mysql-connector-java-5.1.39.tar.gz to admin node(rhel1)

```
#scp mysql-connector-java-5.1.39.tar.gz rhel1:/root/
```

3. Log in to the admin node and extract the file:

```
#tar xzvf mysql-connector-java-5.1.39.tar.gz
```

4. Create the /usr/share/java/ directory on the admin node (rhel1)

```
#mkdir -p /usr/share/java/
```

5. Go to the mysql-connector-java-5.1.39 directory on the admin node (rhel1) and copy mysql-connector-java-5.1.39-bin.jar to /usr/share/java/

```
#cd mysql-connector-java-5.1.39

#cp mysql-connector-java-5.1.39-bin.jar /usr/share/java/mysql-
connector-java.jar
```

Creating Databases for Servers

To create databases for Activity Monitor, Reports Manager, Hive Metastore Server, Navigator Audit Server and Navigator Metadata Server

1. In the admin node Log into MySQL as the root user:

```
#mysql -u root -p
```

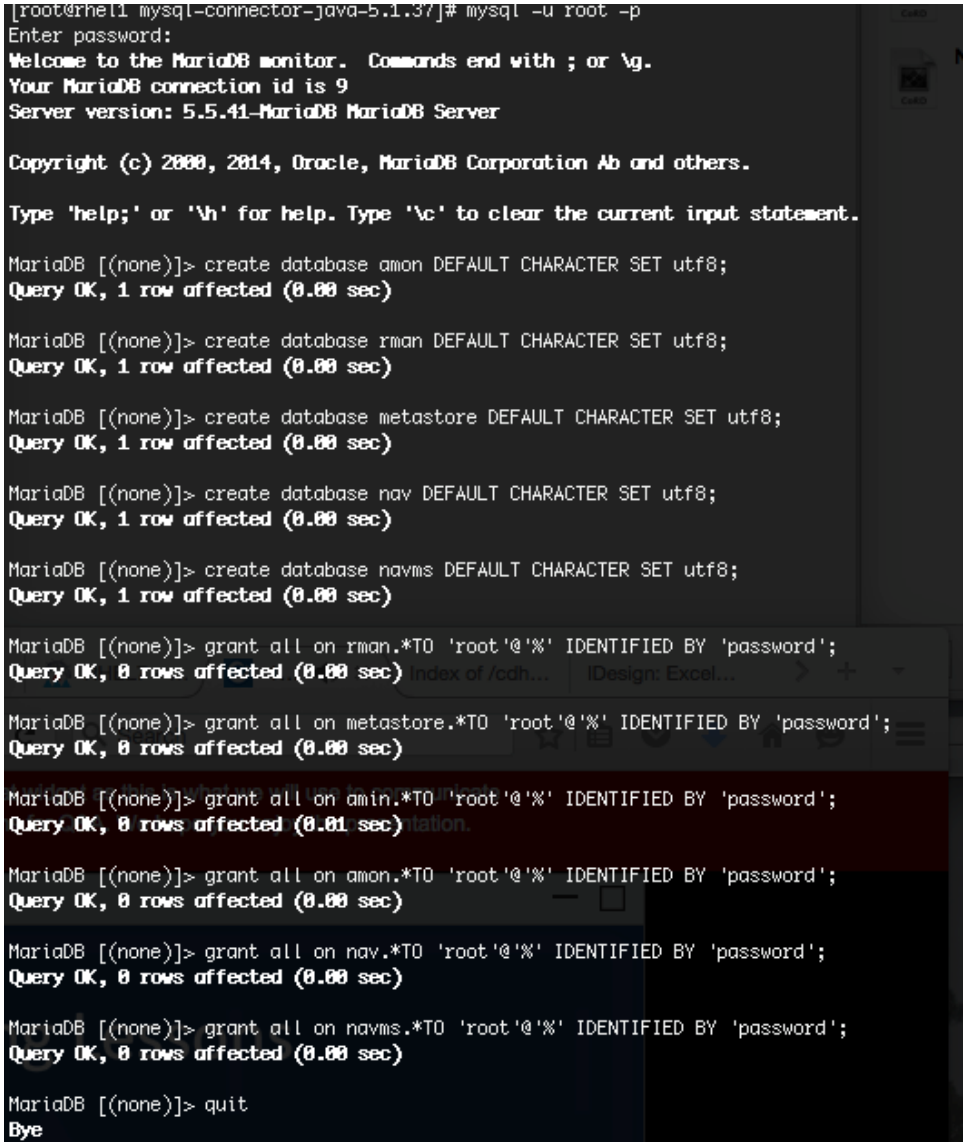
2. Enter the password that was supplied in step 8 above.

Enter password:

3. Create databases for the Activity Monitor, Reports Manager and Hive Metastore Server using the command below

```
MariaDB [(none)]> create database amon DEFAULT CHARACTER SET utf8;

MariaDB [(none)]> create database hue DEFAULT CHARACTER SET utf8;

MariaDB [(none)]> create database rman DEFAULT CHARACTER SET utf8;

MariaDB [(none)]> create database metastore DEFAULT CHARACTER SET utf8;

MariaDB [(none)]> create database nav DEFAULT CHARACTER SET utf8;

MariaDB [(none)]> create database navms DEFAULT CHARACTER SET utf8;

MariaDB [(none)]> create database sentry DEFAULT CHARACTER SET utf8;

MariaDB [(none)]> create database oozie DEFAULT CHARACTER SET utf8;
```

```
MariaDB [(none)]> grant all on rman.*TO 'root'@'%' IDENTIFIED
BY'password';

MariaDB [(none)]> grant all on hue.*TO 'root'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all on metastore.*TO 'root'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all on amon.*TO 'root'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all on nav.*TO 'root'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all on navms.*TO 'root'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all on sentry.*TO 'root'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all privileges on oozie.* to root@'%'
IDENTIFIED BY ‘password’;

MariaDB [(none)]> grant all on rman.*TO 'rman'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all on hue.*TO 'hue'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all on metastore.*TO 'hive'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all on amon.*TO 'amon'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all on nav.*TO 'nav'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all on navms.*TO 'navms'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all on sentry.*TO 'root'@'%' IDENTIFIED BY
'password';

MariaDB [(none)]> grant all privileges on oozie.* to oozie@'%'
IDENTIFIED BY ‘password’;
```



Installing Cloudera Manager

The following section describes installation of Cloudera Manager first and then using Cloudera Manager to install Cloudera Enterprise 5.8.

Setting Up the Cloudera Manager Server Database

The Cloudera Manager Server Database stores information about service and host configurations.

Installing Cloudera Manager

Cloudera Manager, an end to end management application, is used to install and configure Cloudera Enterprise. During Cloudera Enterprise Installation, Cloudera Manager's Wizard will help to install Hadoop services on all nodes using the following procedure:

- Discovery of the cluster nodes
- Configure the Cloudera parcel or package repositories
- Install Hadoop, Cloudera Manager Agent (CMA) and Impala on all the cluster nodes.
- Install the Oracle JDK if it is not already installed across all the cluster nodes.
- Assign various services to nodes.
- Start the Hadoop services.

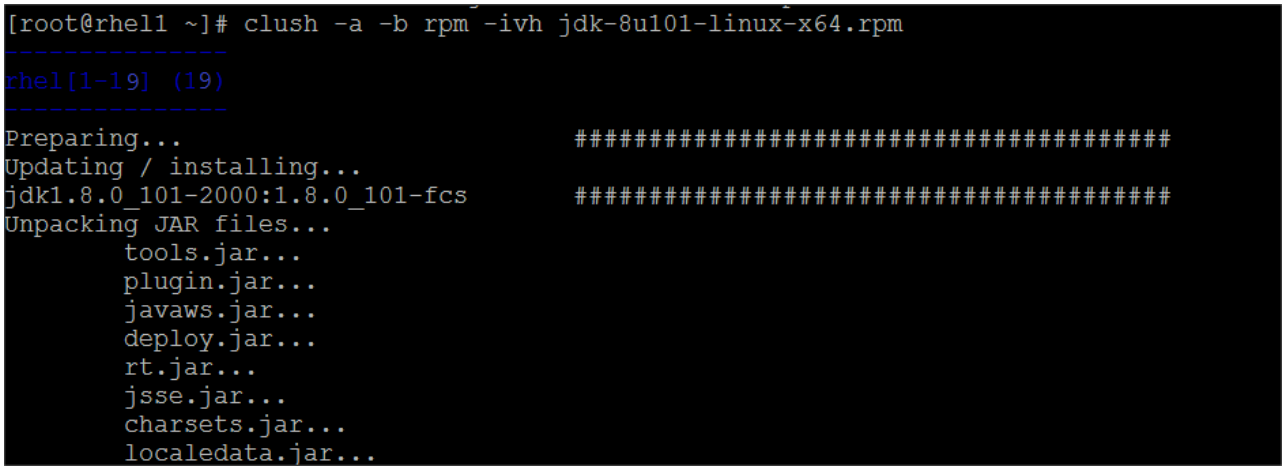
To install Cloudera Manager, complete the following steps:

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 all the host. (download the JDK1.8)

```
#clush -a -b rpm -ivh jdk-8u101-linux-x64.rpm
```



3. Install the Cloudera Manager Server packages either on the host where the database is installed, or on a host that has access to the database.

```
#yum install cloudera-manager-daemons cloudera-manager-server
```

Preparing a Cloudera Manager Server External Database

1. Run the scm_prepare_database.sh script on the host where the Cloudera Manager Server package is installed (rhel1) admin node.

```
cd /usr/share/cmf/schema
./scm_prepare_database.sh mysql amon root <password>
./scm_prepare_database.sh mysql hue root <password>
./scm_prepare_database.sh mysql rman root <password>
./scm_prepare_database.sh mysql metastore root <password>
./scm_prepare_database.sh mysql nav root <password>
./scm_prepare_database.sh mysql navms root <password>
./scm_prepare_database.sh mysql sentry root <password>
./scm_prepare_database.sh mysql oozie root <password>
```

2. Verify the database connectivity using the following command.

```
[root@rhel1 ~]# mysql –u root –p
mysql> connect amon
mysql> connect rman
mysql> connect metastore
mysql> connect nav
mysql> connect navms
mysql> connect sentry
mysql> connect oozie
```

```
[root@rhel1 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.1.71 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> connect amon
Connection id:    15
Current database: amon

mysql> connect rman
Connection id:    16
Current database: rman

mysql> connect metastore
Connection id:    17
Current database: metastore
```

The MySQL External database setup is complete.

Starting the Cloudera Manager Server

1. Start the Cloudera Manager Server

```
#service cloudera-scm-server start
```

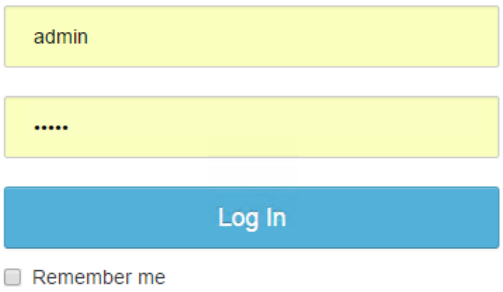
2. Access the Cloudera Manager using the URL, http://172.16.46.11:7180 to verify that the server is up.

3. Once the installation of Cloudera Manager is complete, install Cloudera Enterprise 5 using the Cloudera Manager Web interface.

Installing Cloudera Enterprise

To install the Cloudera Enterprise Data Hub, complete the following steps:

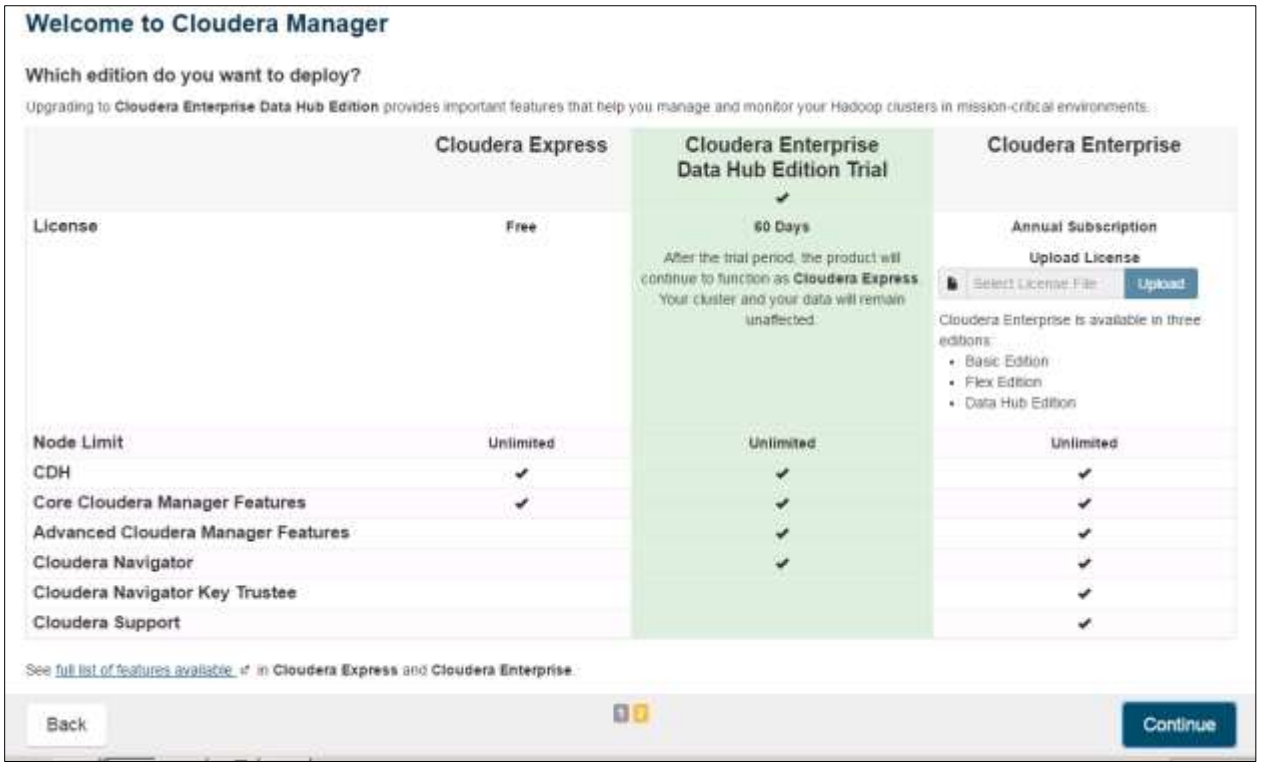
1. Login to the Cloudera Manager. Enter " admin" for both the Username and Password fields.



2. If you do not have a Cloudera license, select Cloudera Enterprise Data Hub Trial Edition. If you do have a Cloudera license, Click “Upload License” and select your license.

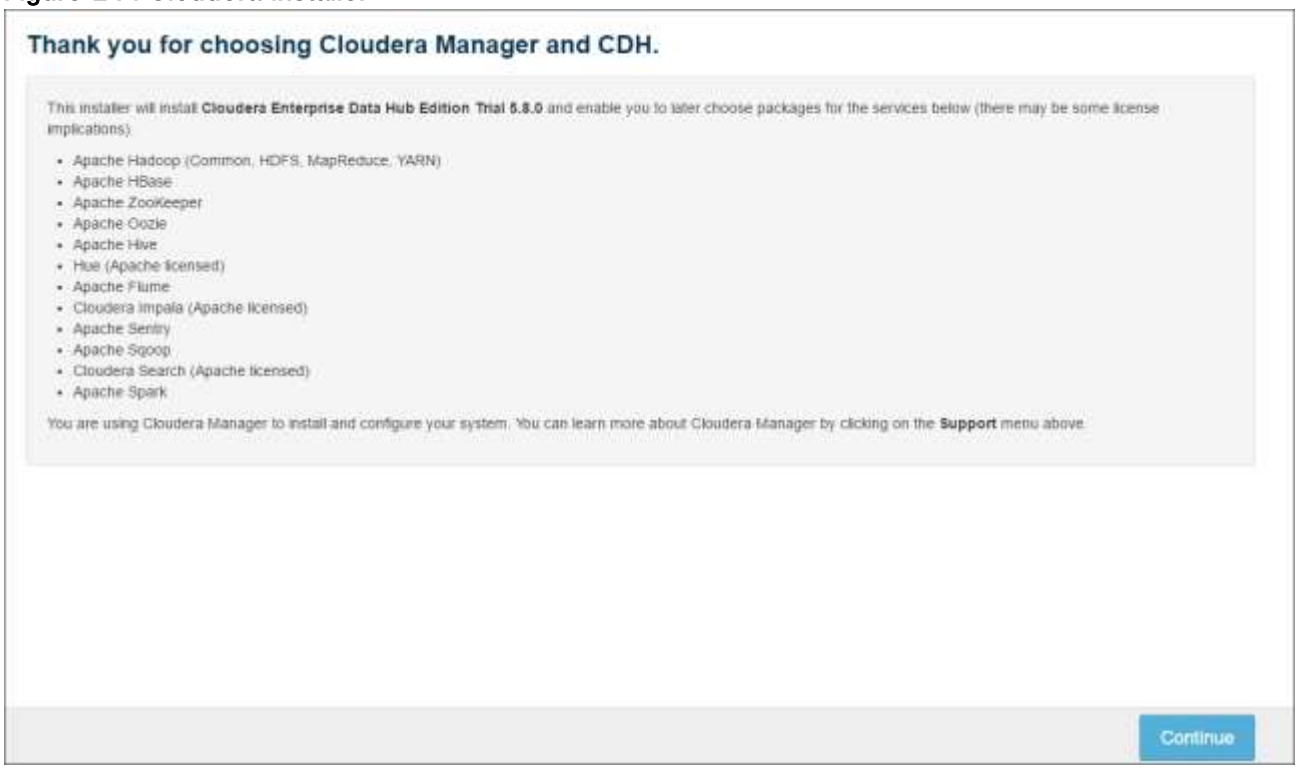
3. Based on requirement, choose appropriate Cloudera Editions for the Installation. (Figure 143)

Figure 143 Installing Cloudera Enterprise



4. Click Continue on the confirmation page. (Figure 144)

Figure 144 Cloudera Installer



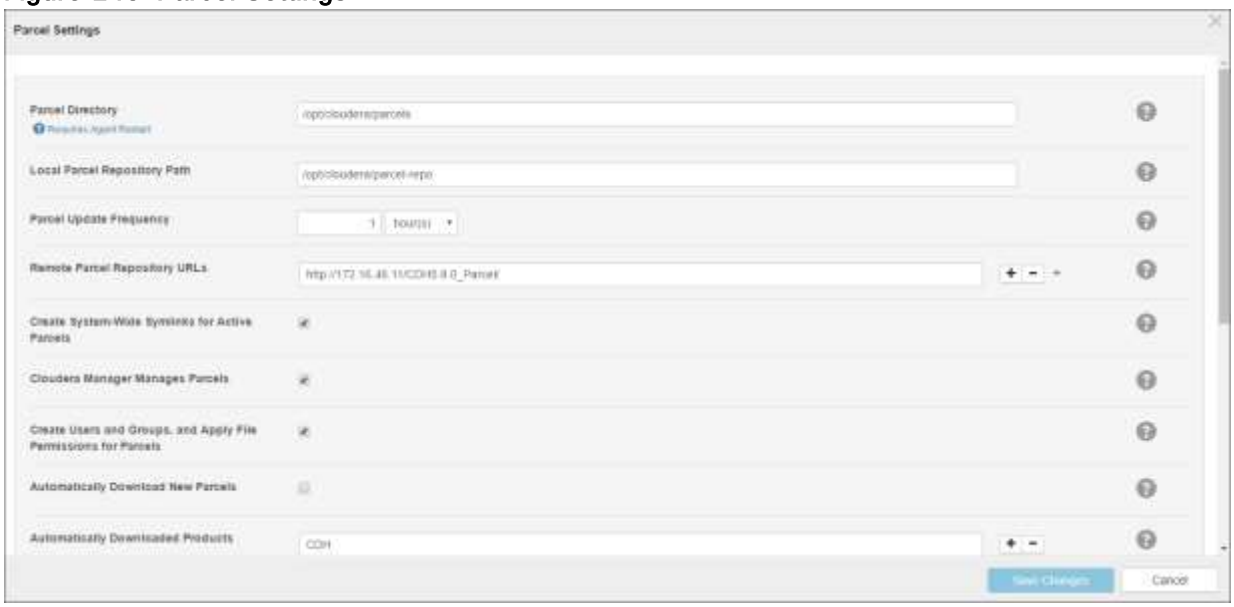
Edit the Cloudera Enterprise Parcel Settings to Use the CDH 5.8.0 Parcels

5. Open another tab in the same browser window and visit the URL: http://172.16.46.11:7180/cmf/parcel/status to modify the parcel settings.

6. Click Configuration on this page.

7. Click to remove the entire remote repository URLs, and add the URL to the location where we kept the CDH 5.8.0 parcels i.e. http://172.16.46.11/CDH5.8.0_Parcels/ (Figure 145)

Figure 145 Parcel Settings

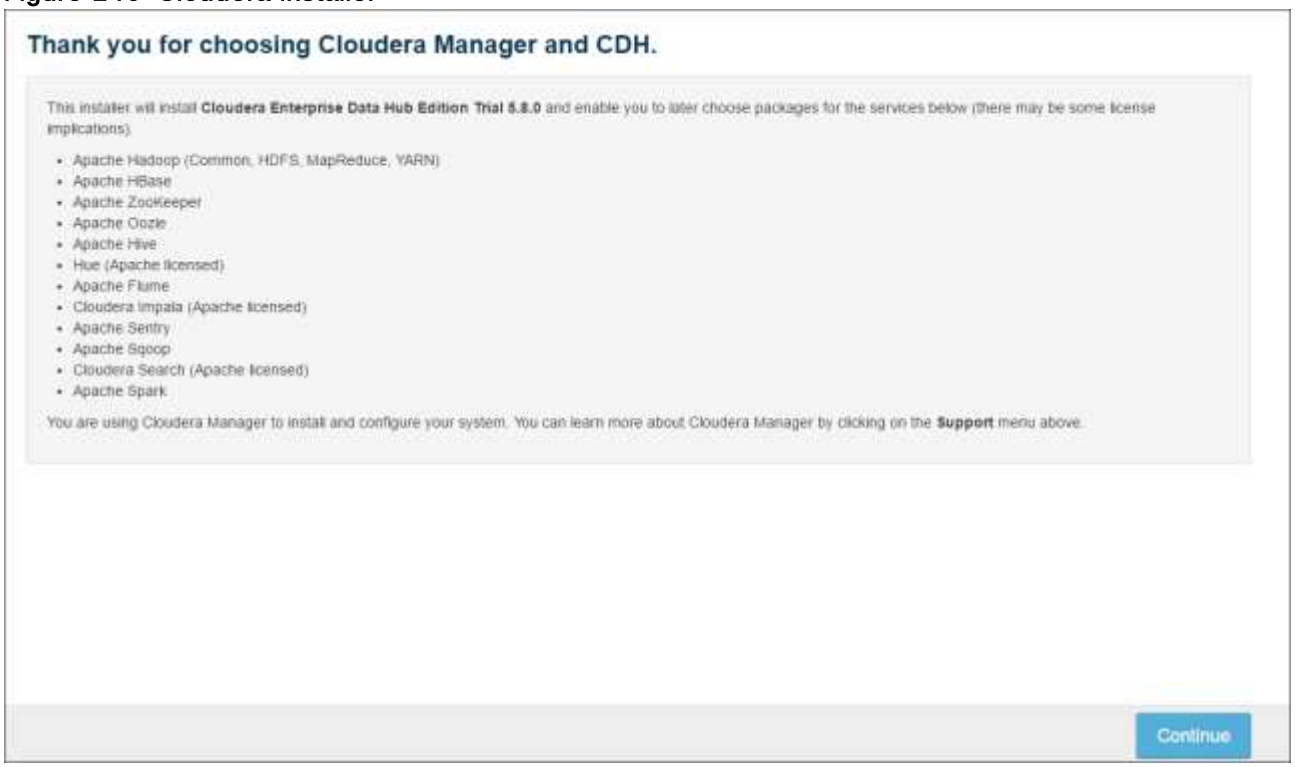


8. Click Save Changes to finish the configuration.

9. Navigate back to the Cloudera installation home page i.e. http://172.16.46.11:7180

10. Click Continue on the confirmation page. (Figure 146)

Figure 146 Cloudera Installer



11. Specify the hosts that are part of the cluster using their IP addresses or hostname. The figure below shows use of a pattern to specify the IP addresses range.

```
172.16.46.[11-19] or rhel[1-19]
```

12. After the IP addresses or hostnames are entered, click Search. (Figure 147)

Figure 147 Searching for Cluster Nodes

13. Cloudera Manager will "discover" the nodes in the cluster. Verify that all desired nodes have been found and selected for installation. (Figure 148)

Figure 148 Specify Hosts for the Cluster

14. Click Continue.

15. For the method of installation, select the Use Parcels (Recommended) radio button. (Figure 149)

16. For the CDH version, select the CDH5.8.0-1.cdh5.8.0.p0.42 radio button.

17. For the specific release of Cloudera Manager, select the Custom Repository radio button.

18. Enter the URL for the repository within the admin node. http://172.16.46.11/clouderarepo/cloudera-manager as in Figure 149, and click Continue as in Figure 150

Figure 149 Cluster Installation

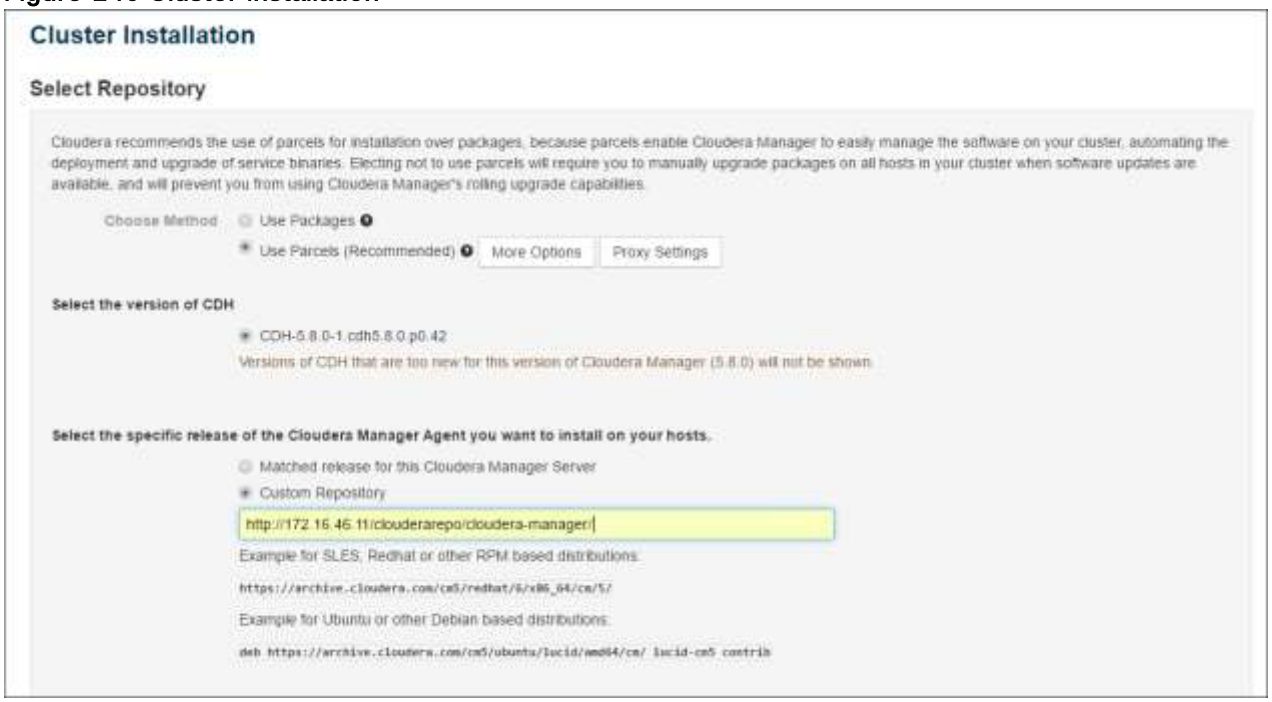
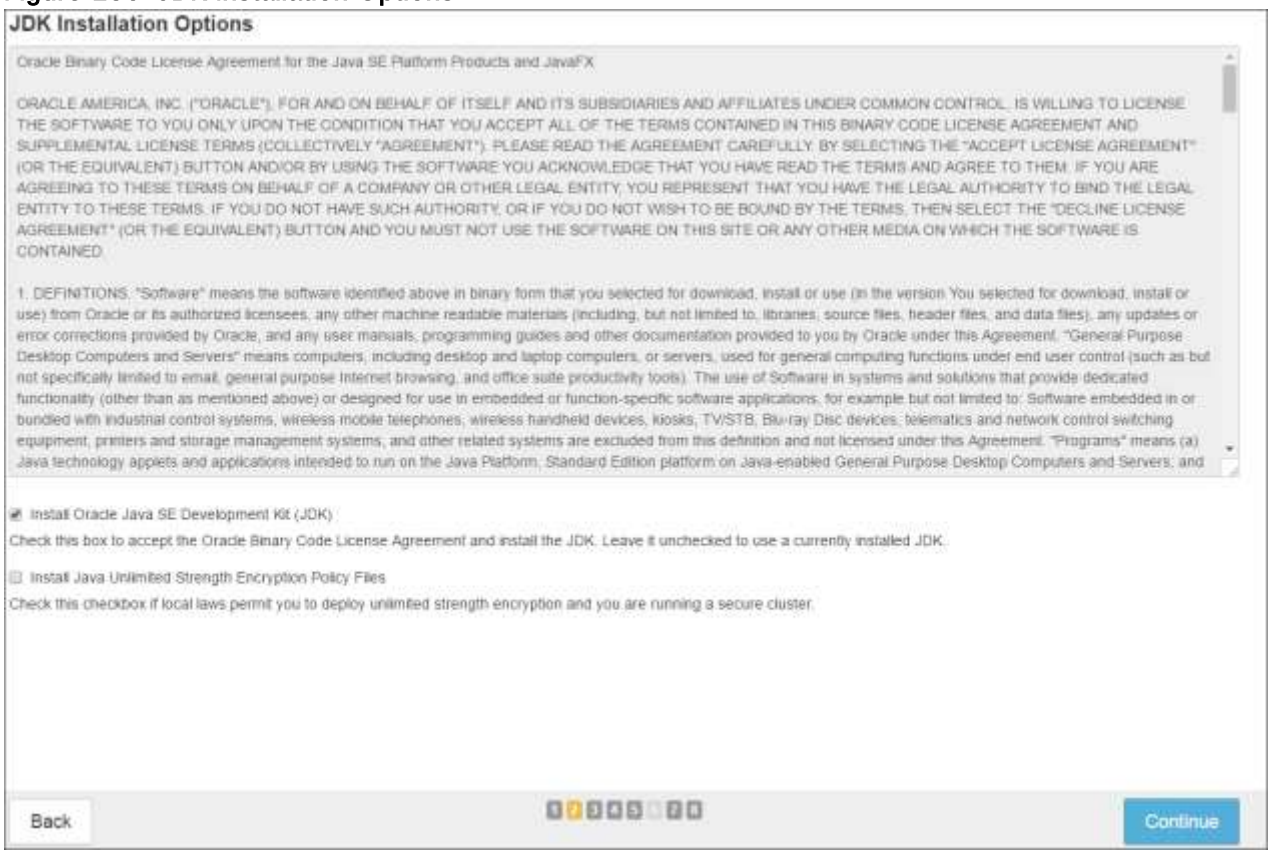
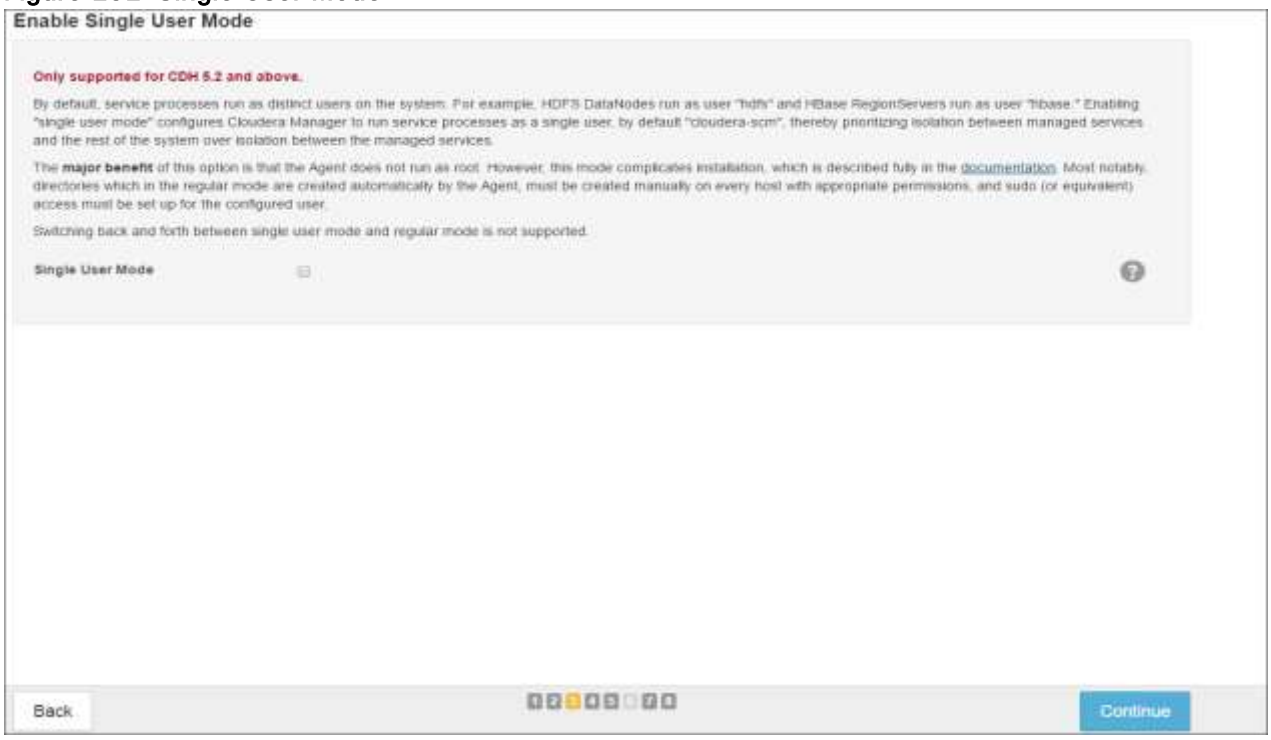


Figure 150 JDK Installation Options



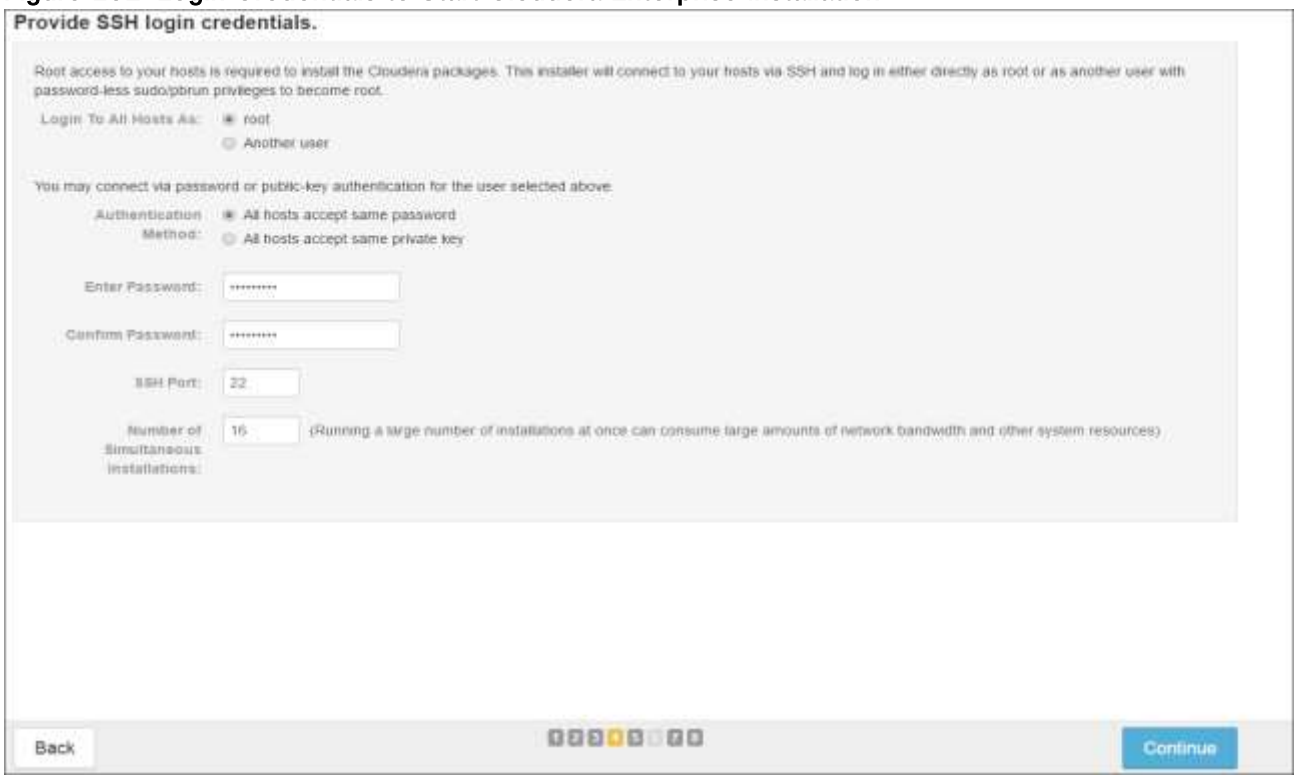
19. Click Continue as in Figure 151

Figure 151 Single User Mode



20. Provide SSH login credentials for the cluster and click Continue. (Figure 152)

Figure 152 Login Credentials to Start Cloudera Enterprise Installation



21. Installation using parcels begins. (Figure 153)

Figure 153 Cluster Installation Confirmation

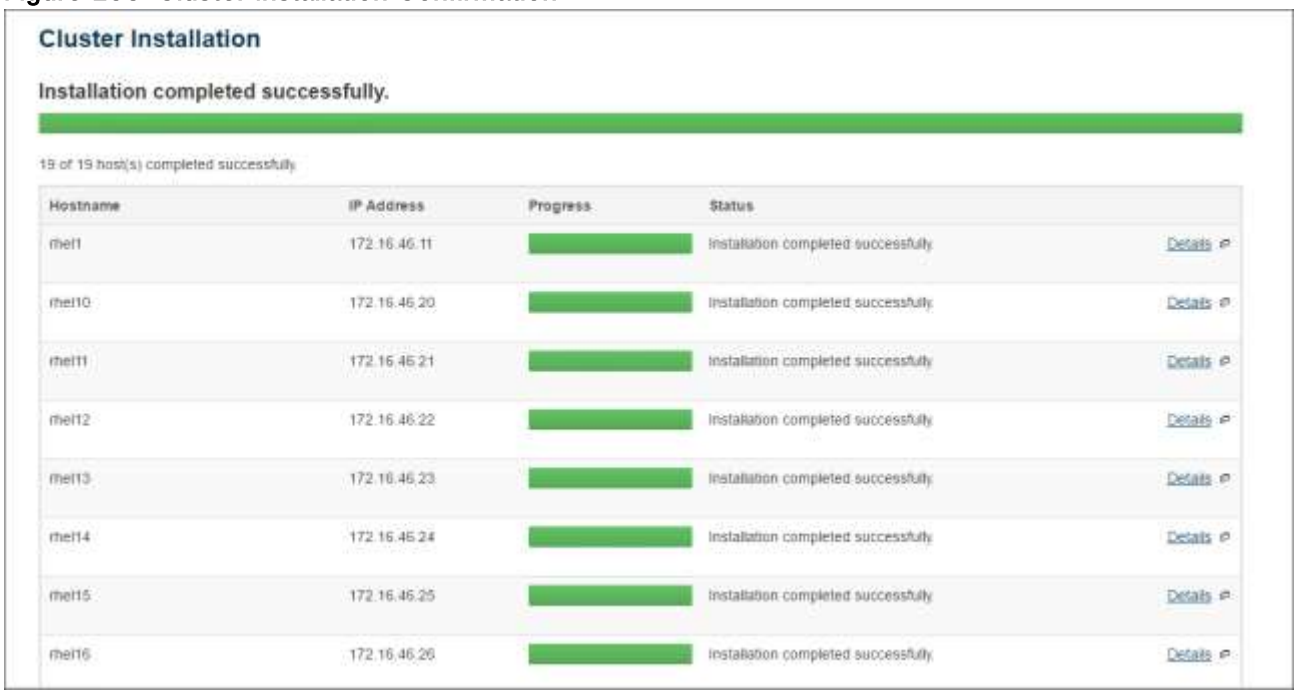


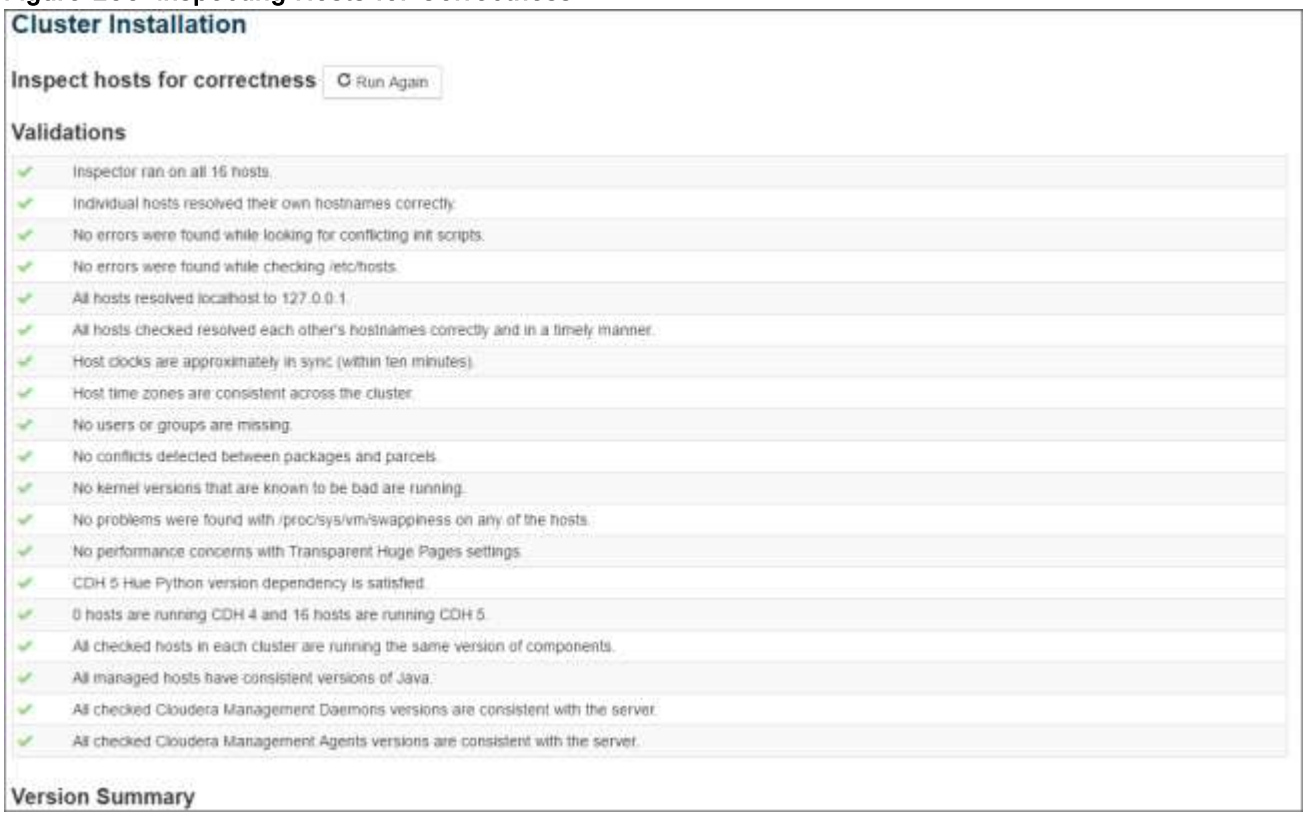
Figure 154 Installing Selected Parcels

22. Once the installation is completed successfully, click Continue to select the required services.

23. Wait for Cloudera Manager to inspect the hosts on which it has just performed the installation.

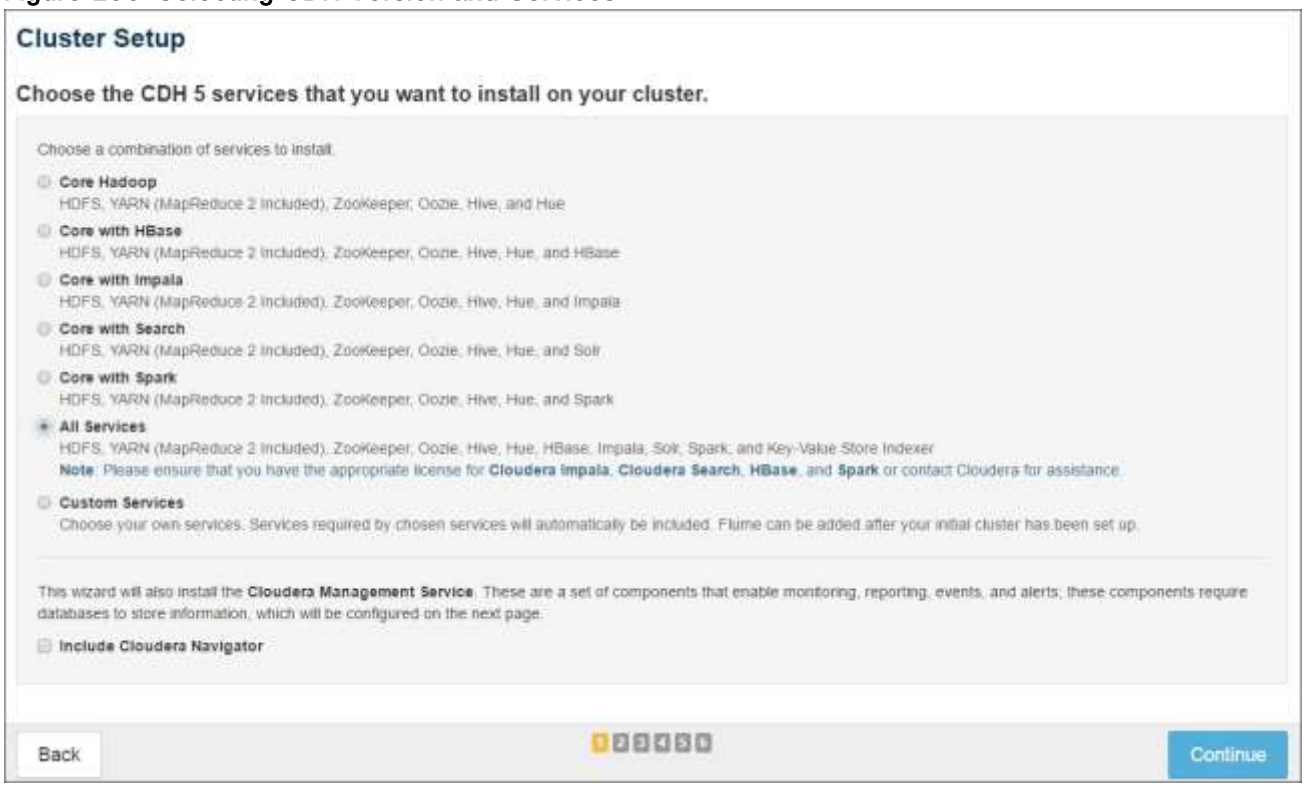
24. Review and verify the summary. Click Continue. (Figure 155)

Figure 155 Inspecting Hosts for Correctness



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

Figure 156 Selecting CDH Version and Services



26. This is one of the critical steps in the installation. Inspect and customize the role assignments of all the nodes based on your requirements and click Continue. (Figure 157)

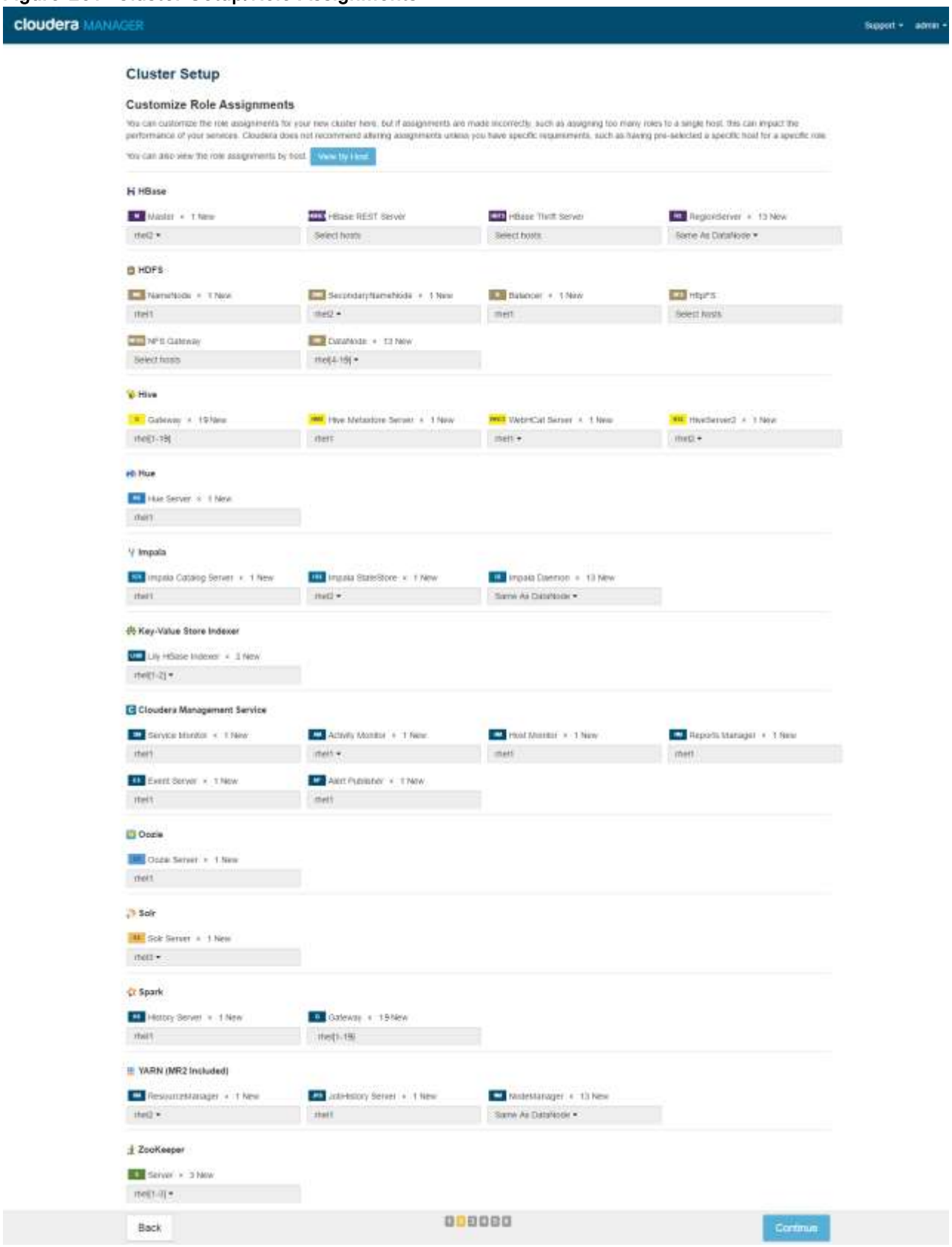
27. Reconfigure the service assignment to match Table 8 below.

Table 8 Service Assignments

Service Name	Host
NameNode	rhel1, rhel2 (HA)
HistoryServer	rhel1
JournalNodes	rhel1,rhel2,rhel3
ResouceManager	rhel2, rhel3(HA)
Hue Server	rhel2
HiveMetastore Server	rhel1
HiveServer2	rhel2
HBase Master	rhel2
Oozie Server	rhel1
ZooKeeper	rhel1, rhel2, rhel3

Service Name	Host
DataNode	rhel4 to rhel19
NodeManager	rhel4 to rhel19
RegionServer	rhel4 to rhel19
Sqoop Server	rhel1
Impala Catalog Server Daemon	rhel1
Impala State Store	rhel2
Impala Daemon	rhel4 to rhel19
Spark History Server	rhel1
Spark Executors	rhel4 to rhel19

Figure 157 Cluster Setup/Role Assignments



Setting Up the Database

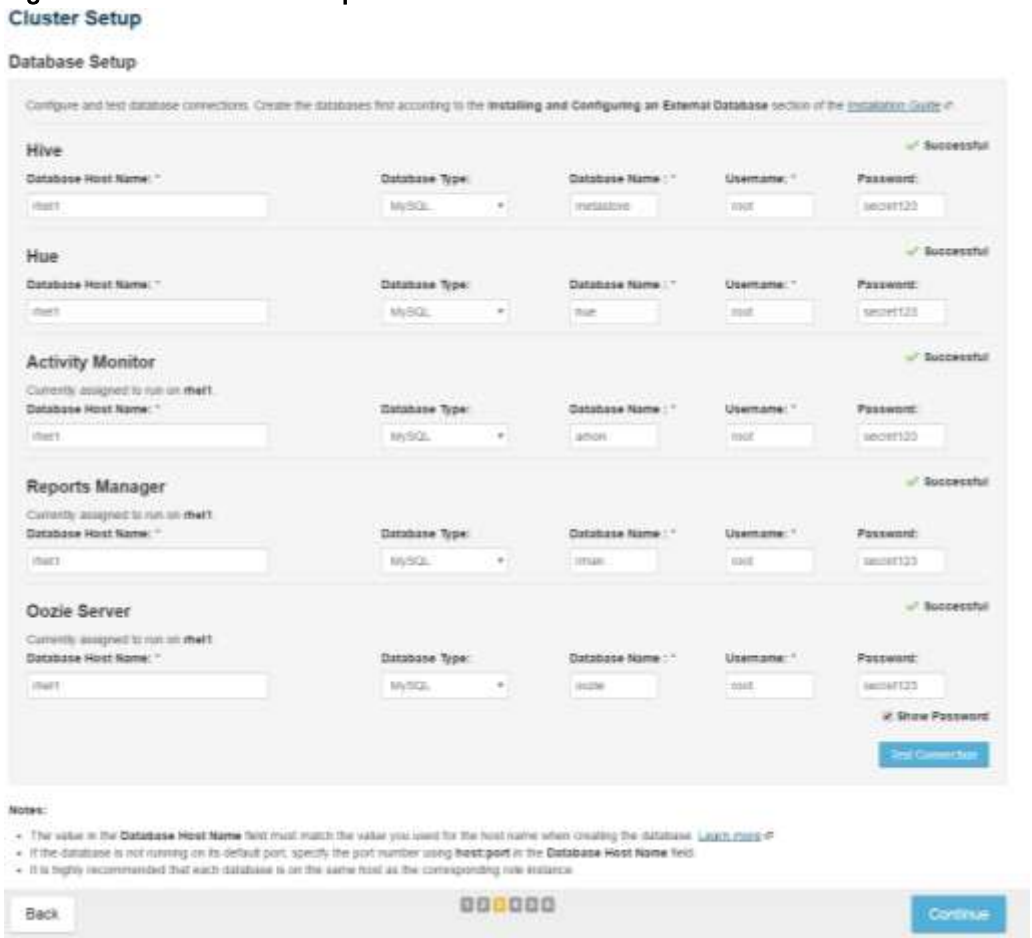
The role assignment recommendation above is for clusters of up to 64 servers. For clusters larger than 64 nodes, use the HA recommendation defined in Table 8 above.

1. In the Database Host Name sections use port 3306 for TCP/IP because connection to the remote server always uses TCP/IP. (Figure 158)

2. Enter the Database Name, username and password that were used during the database creation stage earlier in this document.

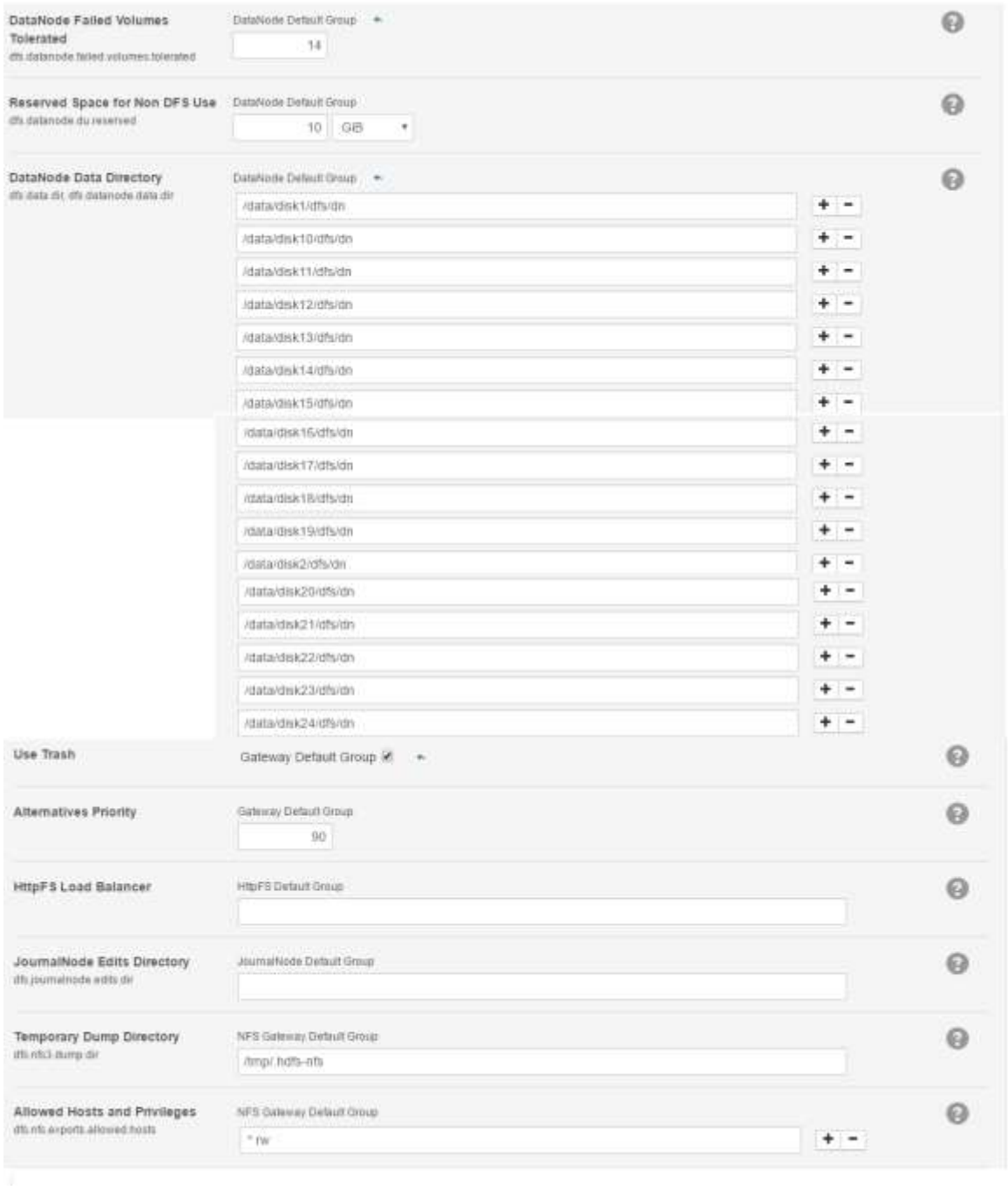
3. Click Test Connection to verify the connection and click Continue.

Figure 158 Database Setup



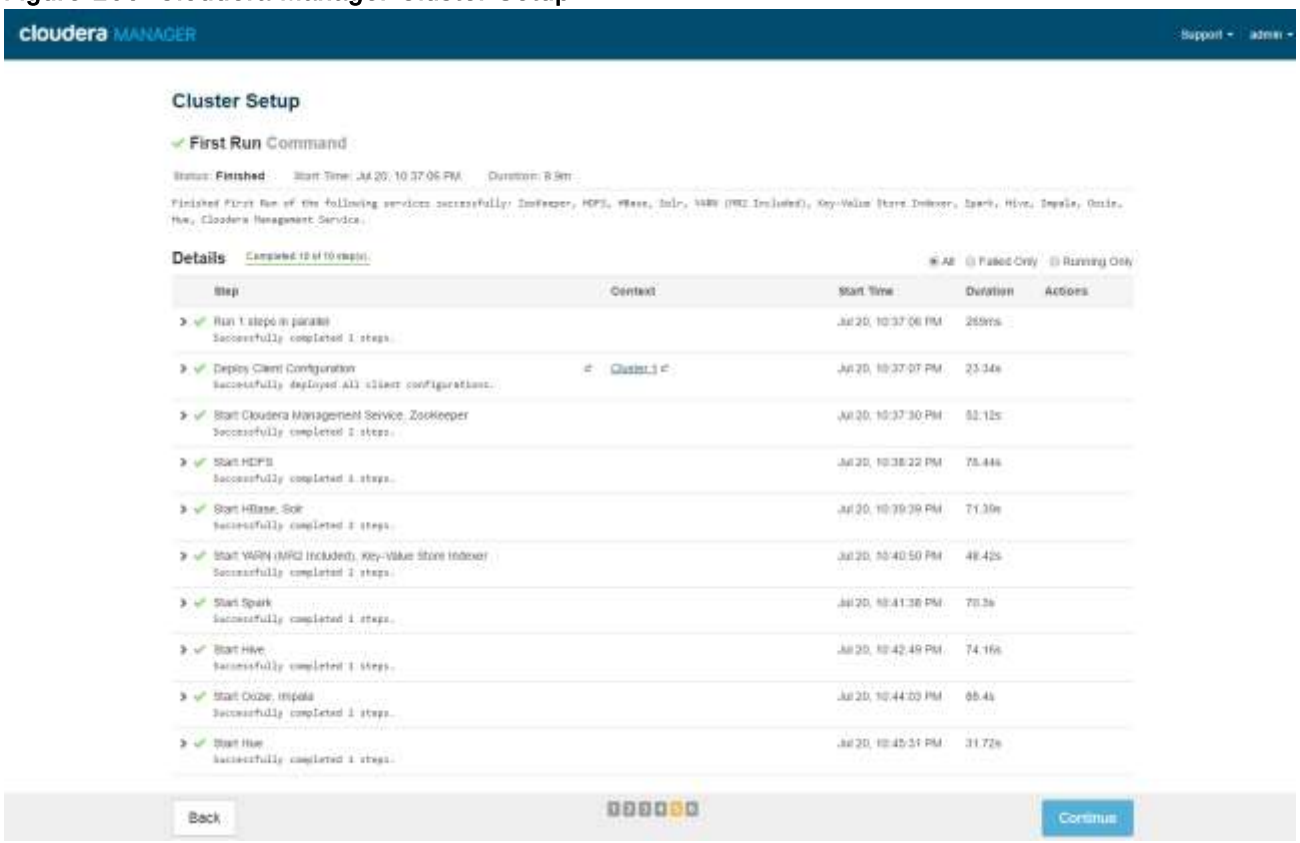
4. Review and customize the configuration changes based on your requirements. (Figure 159)

Figure 159 Review the Configuration Changes Part1



5. Click Continue to start running the cluster services. (Figure 160)

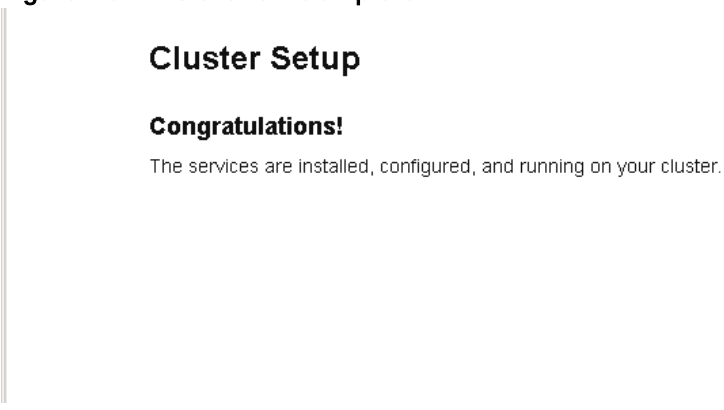
Figure 160 Cloudera Manager Cluster Setup



Starting the Cluster Services

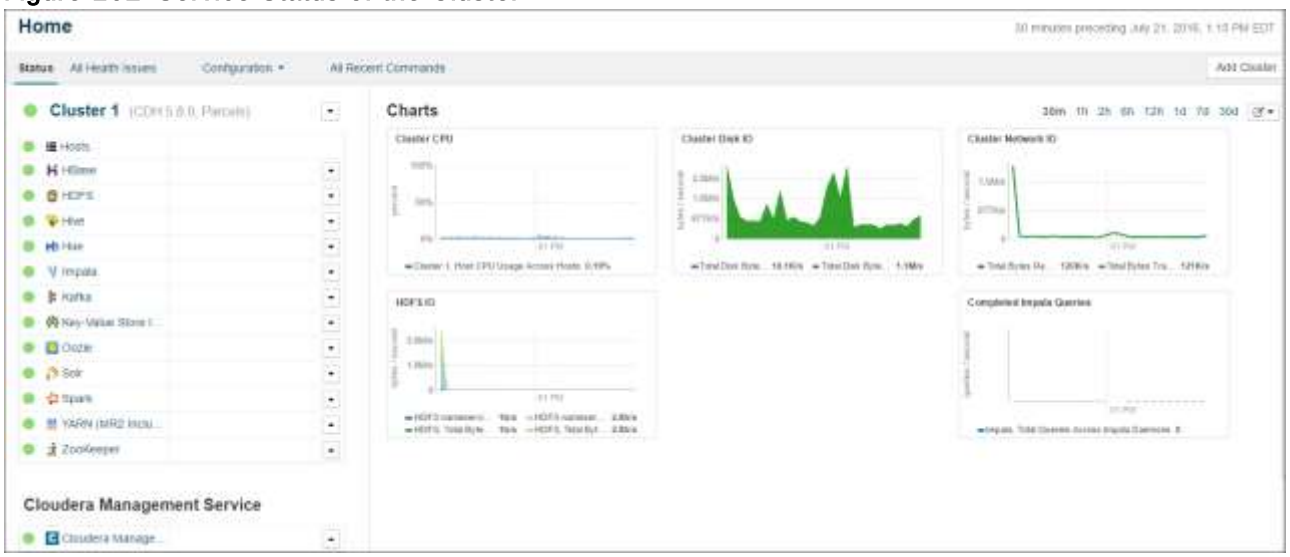
1. Hadoop services are installed, configured and now running on all the nodes of the cluster. Click Finish to complete the installation. (Figure 161)

Figure 161 Installation Complete



Cloudera Manager now displays the status of all Hadoop services running on the cluster. (Figure 162)

Figure 162 Service Status of the Cluster



Scaling the Cluster

The role assignment recommendation above is for a cluster with at least 16 servers and in High Availability (HA). For smaller clusters running without HA, the recommendation is to dedicate one server for the NameNode and a second server for a secondary name node and YARN Resource Manager. For clusters larger than 64 nodes, the recommendation is to dedicate one server each for NameNode, YARN Resource Manager and one more for running both NameNode (HA) and Resource Manager (HA) as in the table (no Secondary NameNode when in HA).

For production clusters it is recommended to set up NameNode and Resource Manager in HA mode. This implies that there will be at least 3 master nodes, running the NameNode, YARN Resource Manager, the fail-over counter-part being designated to run on another node and a third node that would have similar capacity as the other two nodes.

All three nodes will also need to run ZooKeeper and Quorum journal node services. It is also recommended to have a minimum of 5 Data Nodes in a cluster. Please refer to the next section for details on how to enable HA.

Rack-Aware Replica Placement using Hadoop Virtualization Extensions

Using rack-aware replica placement extends the Hadoop topology awareness mechanism and refines the data-locality related policies to further improve data reliability, availability and network utilization. HVE (Hadoop Virtualization Extensions) allow Hadoop clusters full awareness of the topology they are running. It is recommended to enable this specifically as there two nodes that share a single chassis in this configuration and it can cause failure or affect data locality group between hosts in this non-virtualized environment.

This needs to be done before data is loaded into HDFS, if not pre-existing HDFS will have to copied back in for HVE to apply to them.

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.

To configure rack awareness on a Hadoop cluster, complete the following steps:

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

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

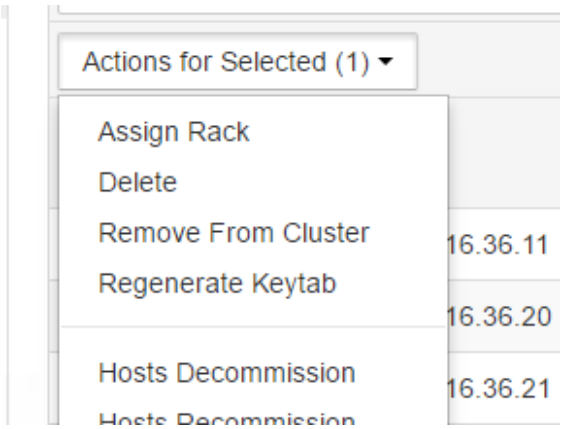
```
<property>

<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.BlockPlacementPoli
cyWithNodeGroup</value>

</property>
```

3. 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 Setting Racks for Hosts:

4. Select each node from the Hosts page and then assign a rack, following the format of /rack$ID/chassis$ID (shown below)



Note: 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.

- Stop the cluster
- Deploy client config
- Start ZooKeeper
- Start HDFS
- Start all the other services.

Enabling High Availability

Note: Setting up HA is done after the Cloudera Installation is completed.

HDFS High Availability

The HDFS HA feature provides the option of running two NameNodes in the same cluster, in an Active/Passive configuration. These are referred to as the Active NameNode and the Standby NameNode. Unlike the Secondary NameNode, the Standby NameNode is a hot standby, allowing a fast failover to a new NameNode in the case that a machine crashes, or a graceful administrator-initiated failover for the purpose of planned maintenance. There cannot be more than two NameNodes.

For more information, go to:

http://www.cloudera.com/documentation/enterprise/latest/topics/cdh_hag_hdfs_ha_intro.html - topic_2_1

Setting Up HDFS HA

The Enable High Availability workflow leads through adding a second (standby) NameNode and configuring JournalNodes. During the workflow, Cloudera Manager creates a federated namespace.

1. Log in to the admin node (rhel1) and create the Edit directory for the JournalNode

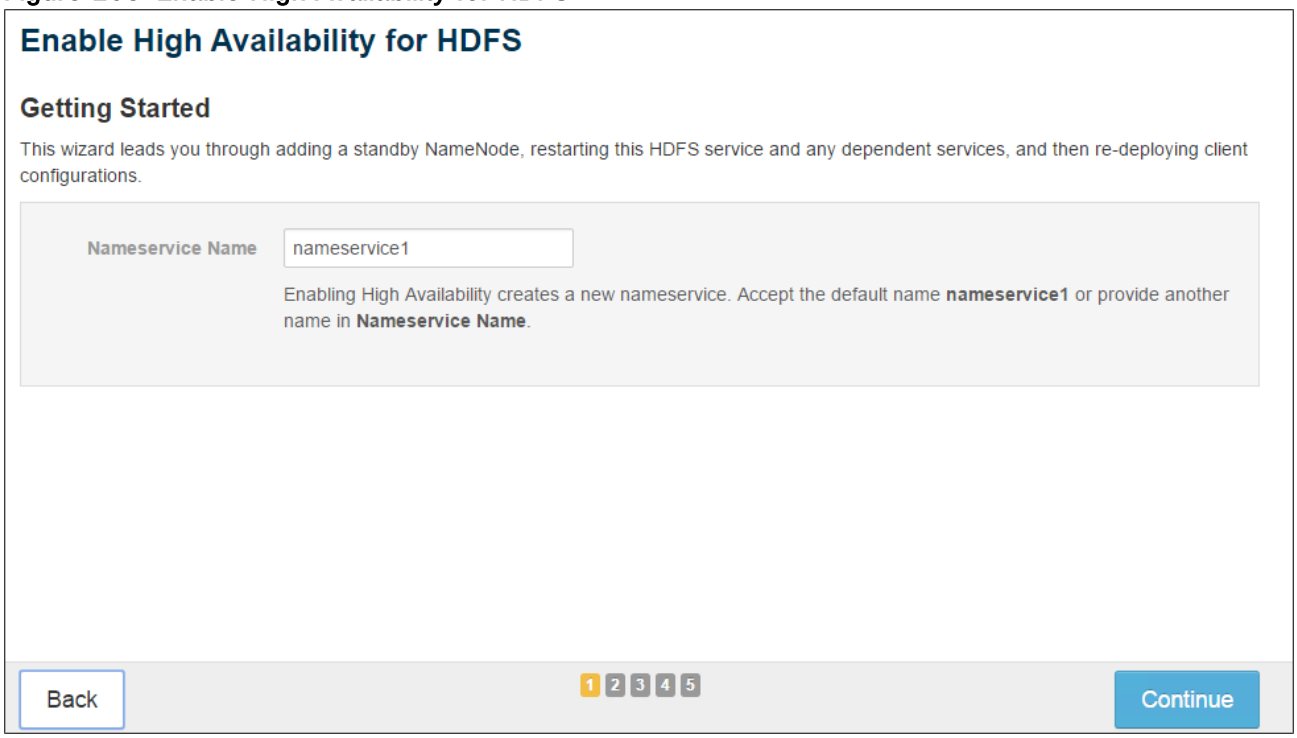
```
#clush -w rhel[1-3] mkdir -p /data/disk1/namenode-edits

#clush -w rhel[1-3] chmod 777 /data/disk1/namenode-edits
```

```
[root@rhel1 ~]# clush -w rhel[1-3] mkdir -p /data/disk1/namenode-edits
[root@rhel1 ~]# clush -w rhel[1-3] chmod 77 /data/disk1/namenode-edits
```

2. Log in to the Cloudera manager and go to the HDFS service.

3. In the top right corner select Actions> Enable High Availability. A screen showing the hosts that are eligible to run a standby NameNode and the JournalNodes displays. (Figure 163)

Figure 163 Enable High Availability for HDFS



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

5. In the NameNode Hosts field, click Select a host. The host selection dialog displays.

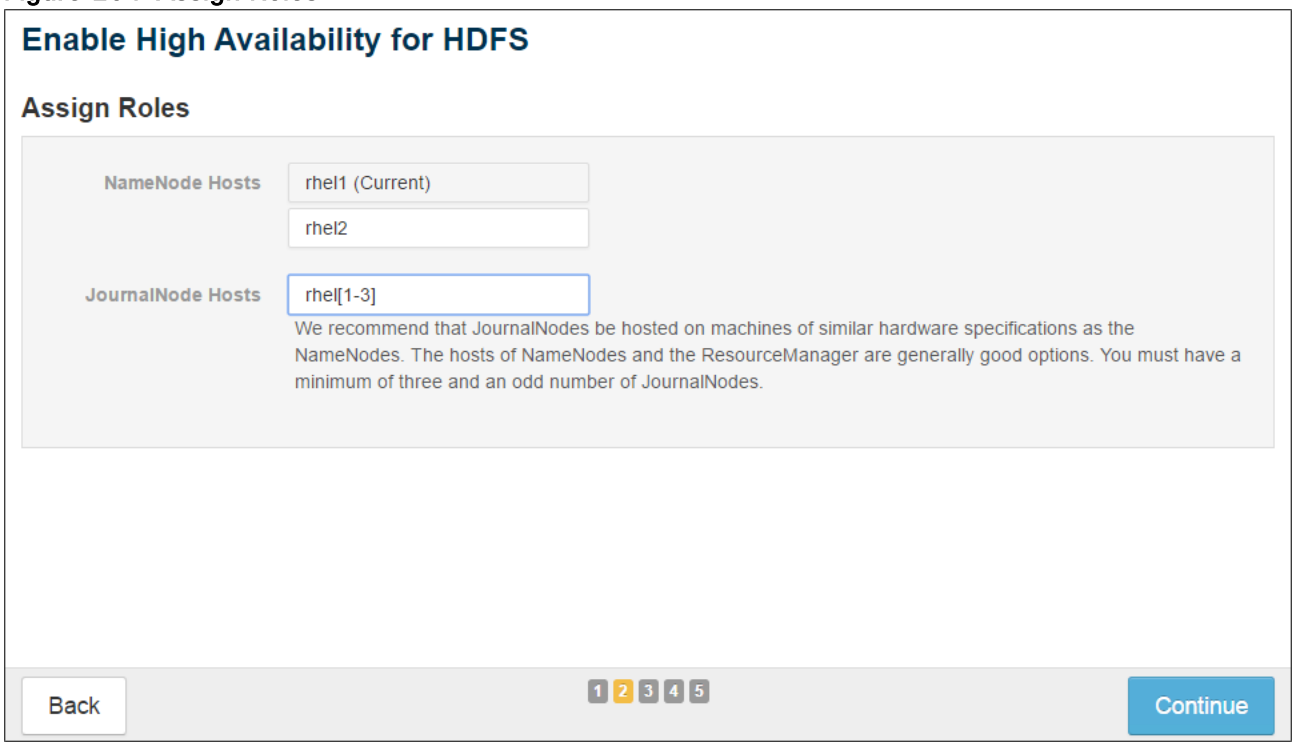
6. Check the checkbox next to the hosts (rhel2) where the standby NameNode is to be set up and click OK.

Note: 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.

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

8. Check the checkboxes next to an odd number of hosts (a minimum of three) to act as JournalNodes and click **OK**. Here we are using the same nodes as ZooKeeper nodes.

Figure 164 Assign Roles

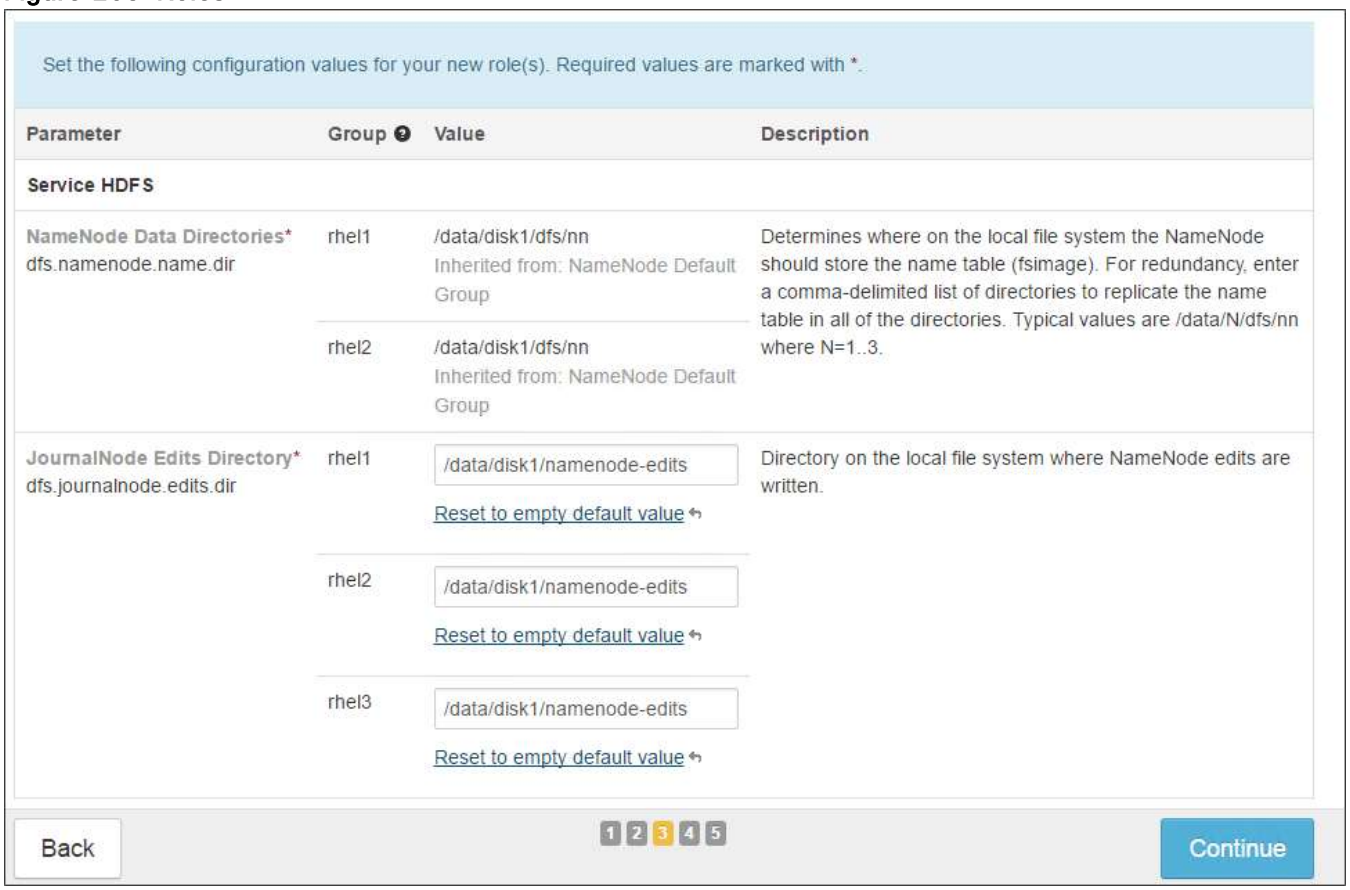


Note: JournalNodes should be hosted on hosts with similar hardware specification as the NameNodes. It is recommended that each JournalNode is put on the same hosts as the active and standby NameNodes, and the third JournalNode on ResourceManager node.

9. Click Continue.

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. (Figure 165)

Figure 165 Roles



Note: The directories specified should be empty, and must have the appropriate permissions.

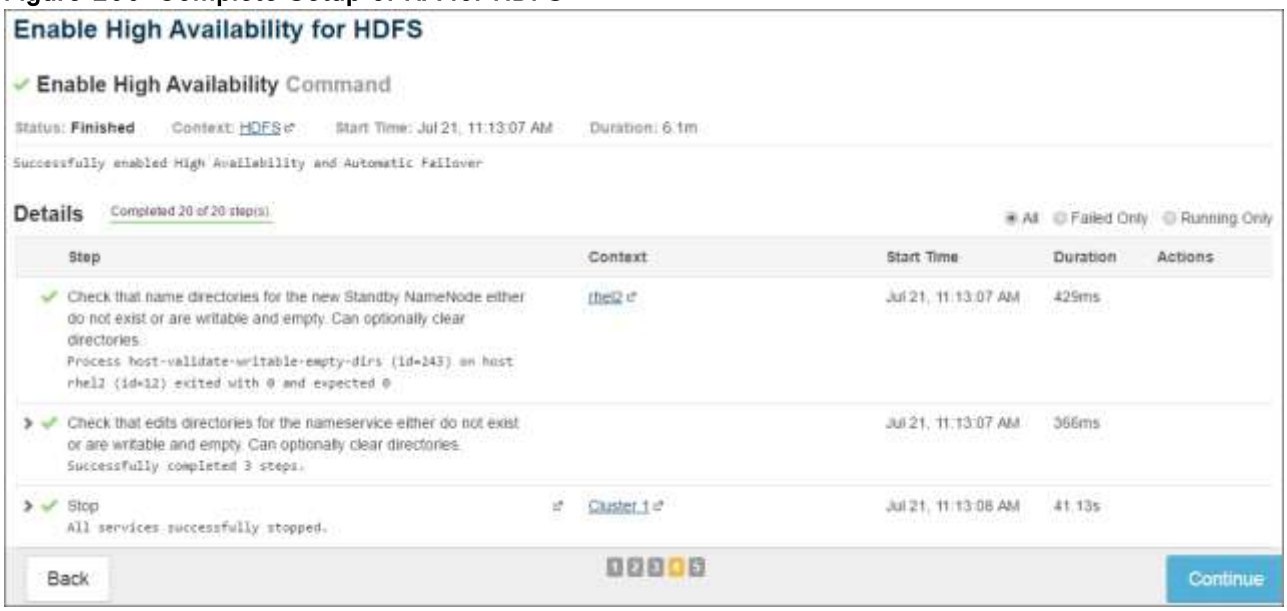
Extra Options: Decide whether Cloudera Manager should clear existing data in ZooKeeper, Standby NameNode, and JournalNodes. If the directories are not empty (for example, re-enabling a previous HA configuration), Cloudera Manager will not automatically delete the contents. Select to delete the contents by keeping the default checkbox selection. The recommended default is to clear the directories.

Note: If chosen not to do so, the data should be in sync across the edits directories of the JournalNodes and should have the same version data as the NameNodes.

11. Click Continue.

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. (Figure 166)

Figure 166 Complete Setup of HA for HDFS



Note: Formatting of the name directory is expected to fail, if the directories are not empty.

Figure 167 HA for HDFS Enabled

12. In the next screen additional steps are suggested by the Cloudera Manager to update the Hue and Hive Metastore. Click finish for Figure 167 shown above.

Note: The following subsections cover configuring Hue and Hive for HA as needed.

Federation and HA

13. In the Cloudera Manager, click on Home> HDFS> Instances to see NameNodes in High Availability. ()

Configuring Hive Metastore to Use HDFS HA

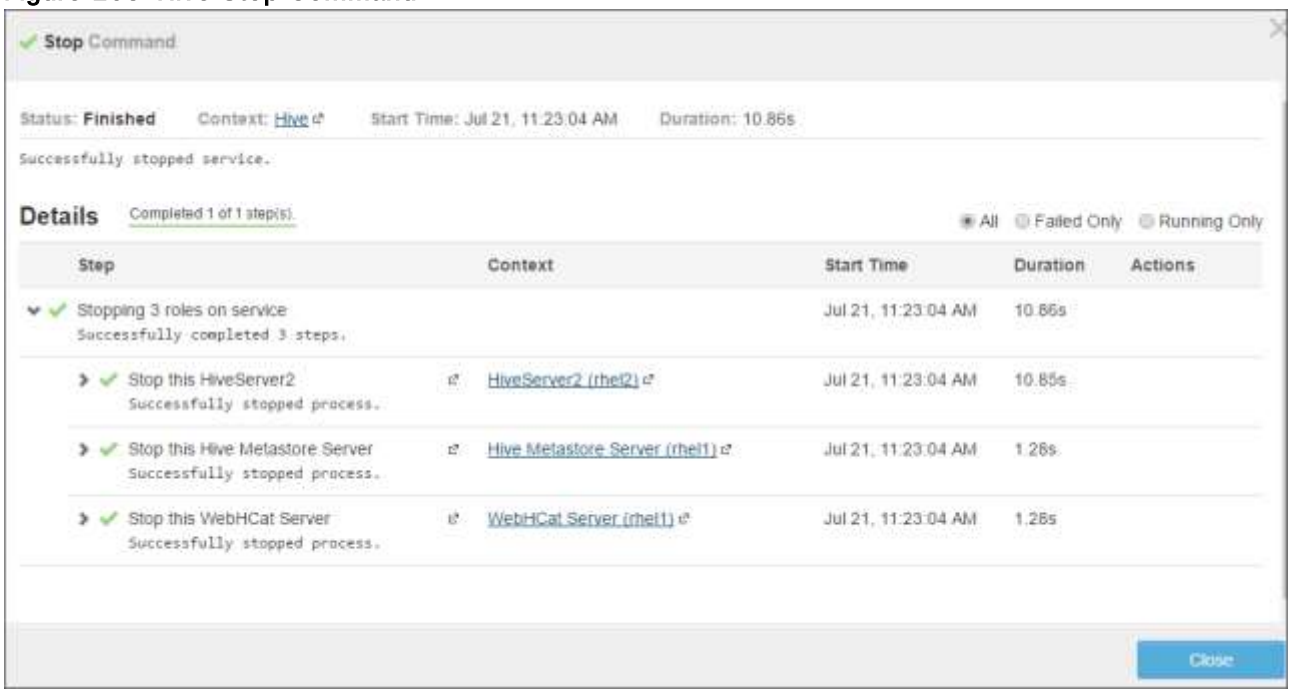
The Hive Metastore can be configured to use HDFS high availability.

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

- Go the Hive service.

- Select Actions> Stop. (Figure 168)
- Click Stop to confirm the command.

Figure 168 Hive Stop Command



15. Back up the Hive Metastore database (if any existing data is present)
16. Select Actions> Update Hive Metastore NameNodes and confirm the command.
17. Select Actions> Start.

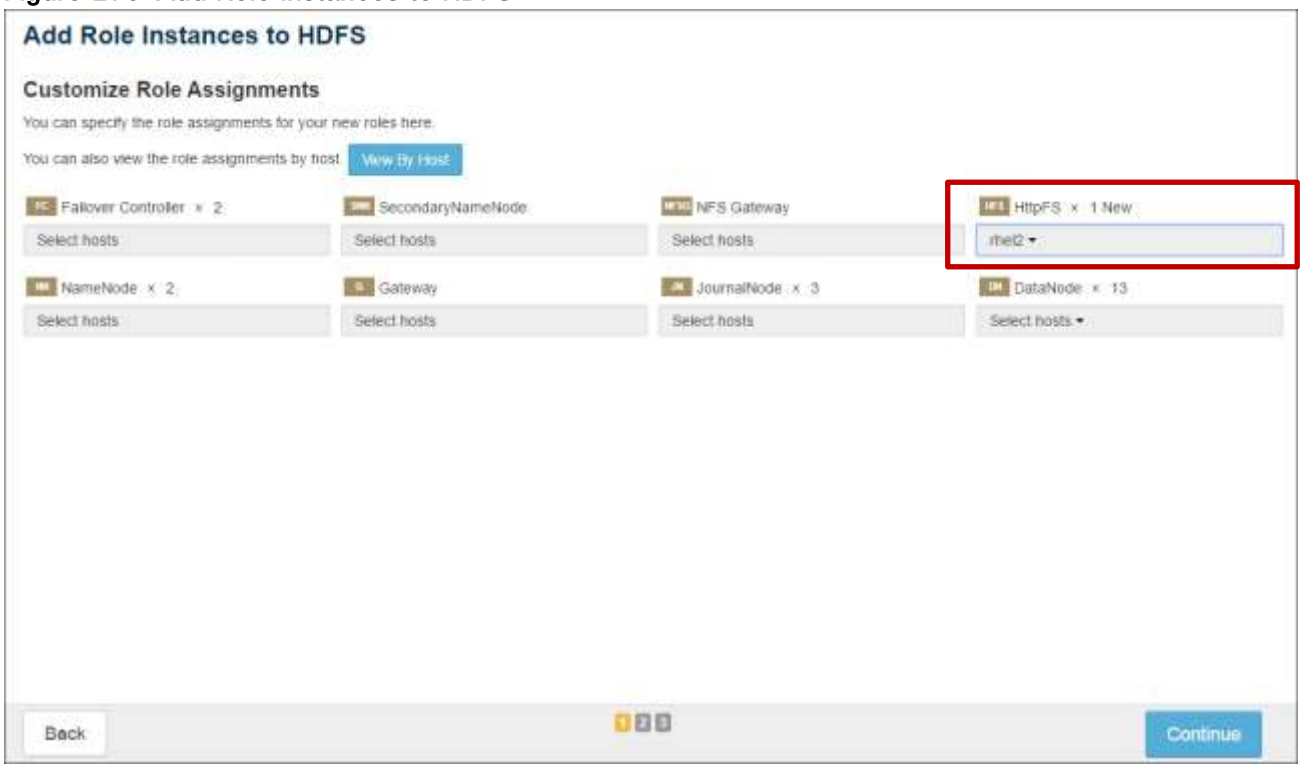
Configuring Hue to Work with HDFS HA

1. Go to the HDFS service.
2. Click the Instances tab.
3. Click Add Role Instances. (Figure 169)

Figure 169 Role Instances

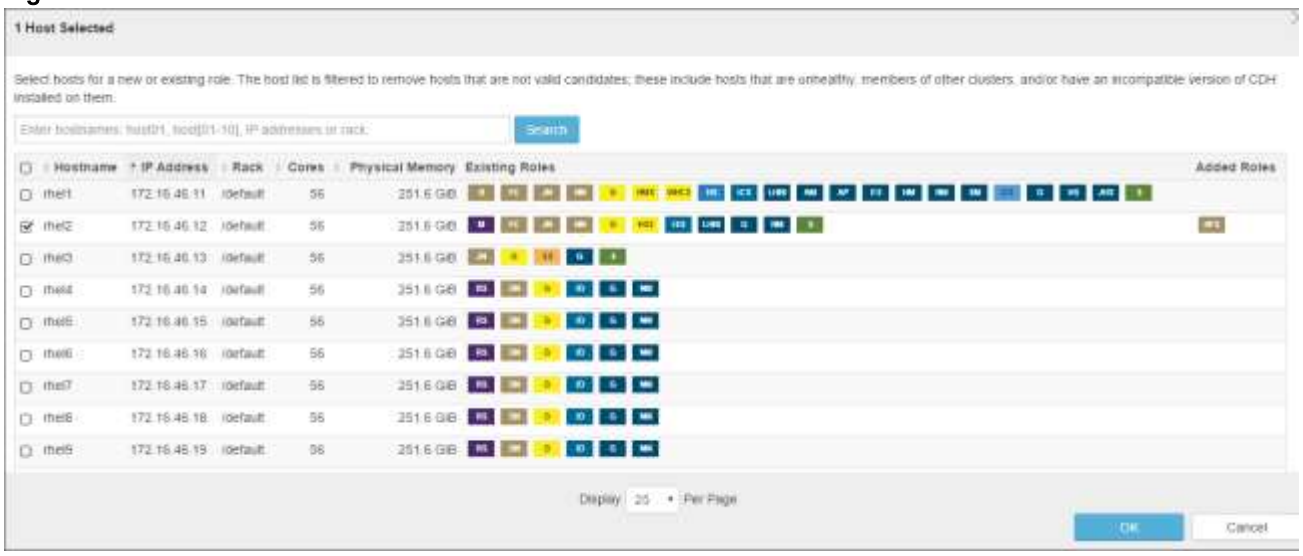
4. Select the text box below the HttpFS field. (Figure 170) The Select Hosts dialog displays.

Figure 170 Add Role Instances to HDFS



5. Select the host on which to run the role and click OK. (Figure 171)

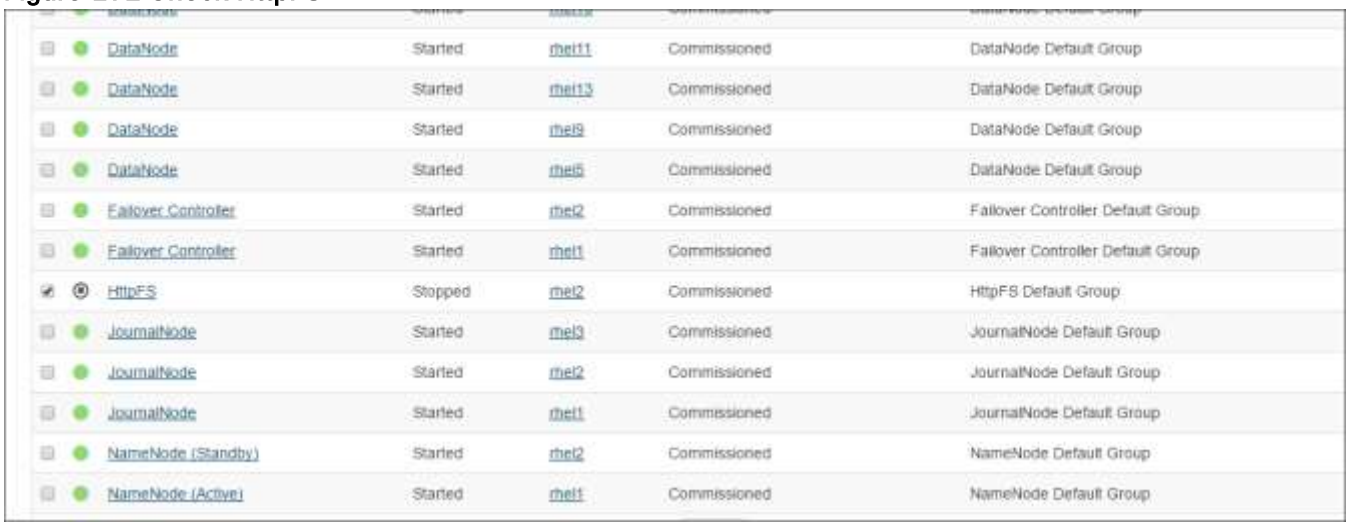
Figure 171 Select the Host to Run the Role



6. Click Continue.

7. Check the checkbox next to the HttpFS role and select Actions for Selected> Start. (Figure 172)

Figure 172 Check HttpFS



DataNode	Started	rhel11	Commissioned	DataNode Default Group
DataNode	Started	rhel13	Commissioned	DataNode Default Group
DataNode	Started	rhel9	Commissioned	DataNode Default Group
DataNode	Started	rhel5	Commissioned	DataNode Default Group
Failover Controller	Started	rhel2	Commissioned	Failover Controller Default Group
Failover Controller	Started	rhel1	Commissioned	Failover Controller Default Group
HttpFS	Stopped	rhel2	Commissioned	HttpFS Default Group
JournalNode	Started	rhel3	Commissioned	JournalNode Default Group
JournalNode	Started	rhel2	Commissioned	JournalNode Default Group
JournalNode	Started	rhel1	Commissioned	JournalNode Default Group
NameNode (Standby)	Started	rhel2	Commissioned	NameNode Default Group
NameNode (Active)	Started	rhel1	Commissioned	NameNode Default Group

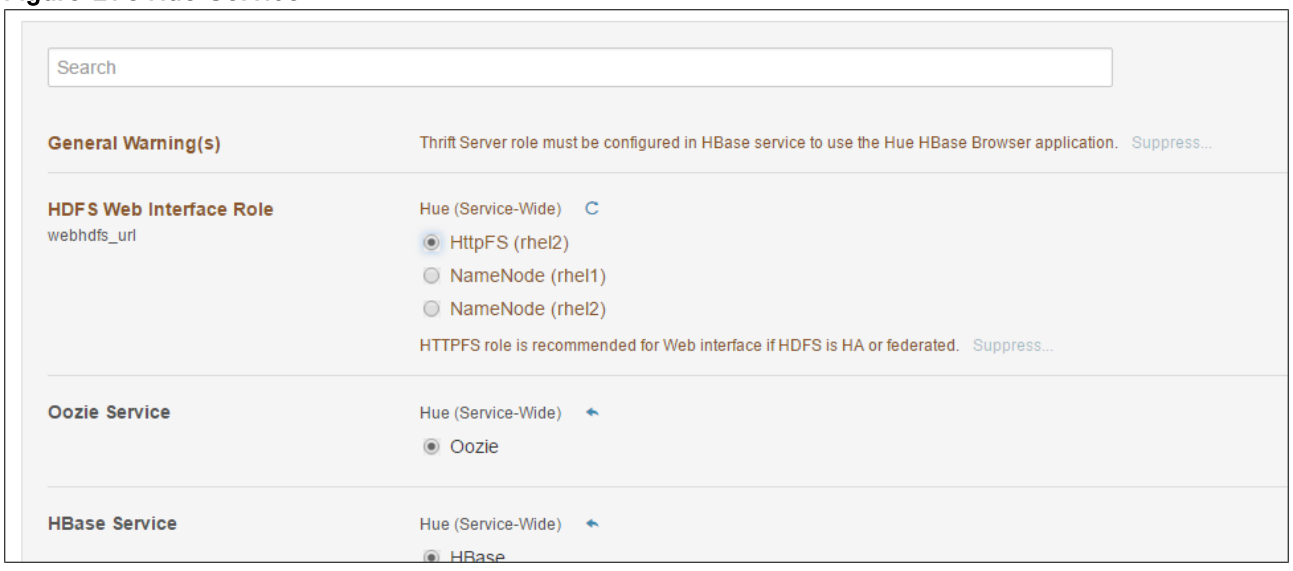
8. After the command has completed, go to the Hue service.

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. (Figure 173)

Figure 173 Hue Service



12. Restart the Hue service.

Note: Refer to the Cloudera website: http://www.cloudera.com/documentation/enterprise/5-3-x/topics/cdh_hag_hdfs_ha_cdh_components_config.html - concept_rj1_hsq_bp for further details on setting up HA for other components like Impala, Oozie etc.

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 please go to:
http://www.cloudera.com/documentation/enterprise/latest/topics/cdh_hag_rm_ha_config.html - xd_583c10bfdbd326ba--43d5fd93-1410993f8c2--7f77

Setting up YARN HA

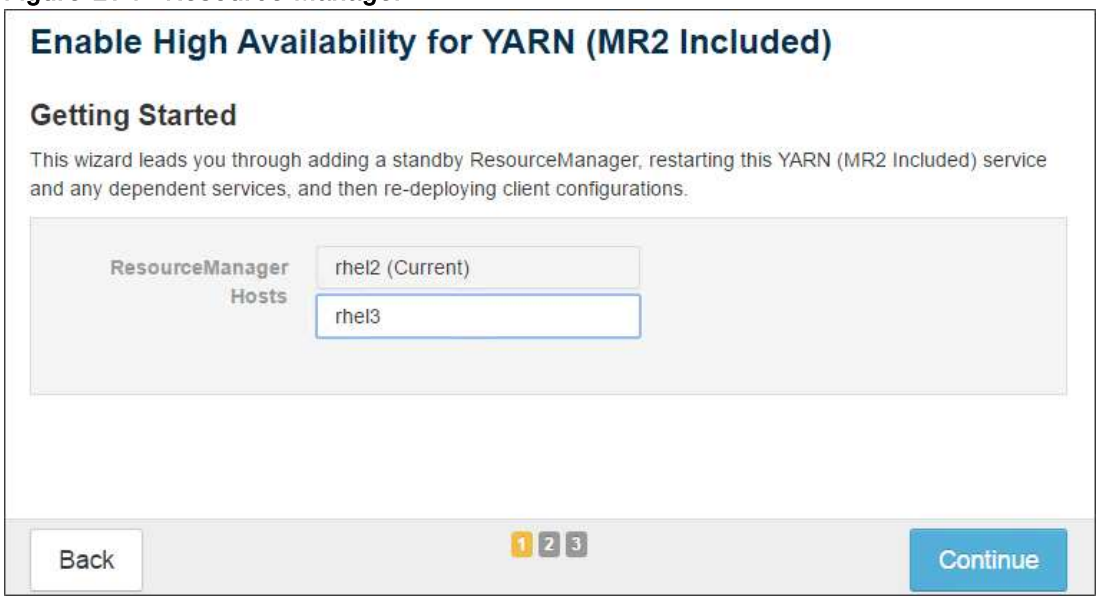
1. Log in to the Cloudera manager and go to the YARN service.
2. Select Actions> Enable High Availability.

A screen showing the hosts that are eligible to run a standby ResourceManager displays. (Figure 174)

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

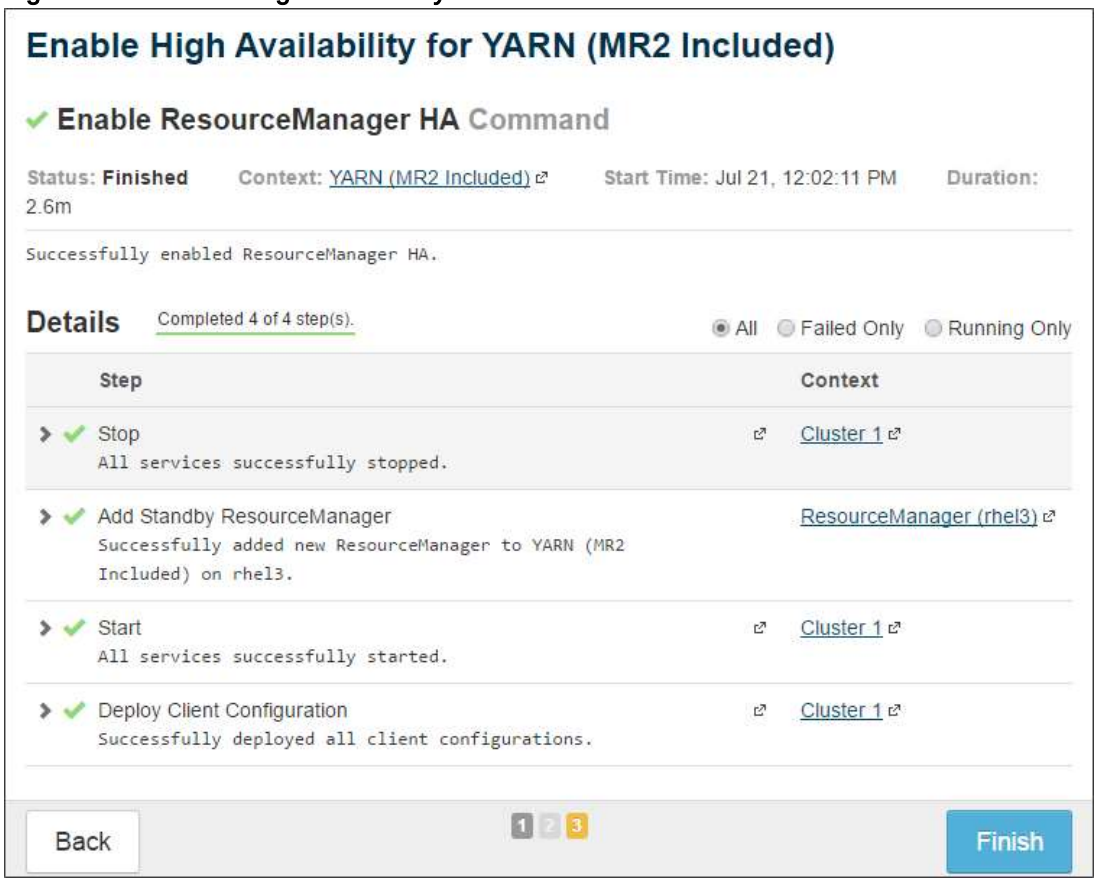
3. Select the host (rhel3) where the standby ResourceManager is to be installed, and click Continue.

Figure 174 Resource Manager



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. (Figure 175)

Figure 175 Enable High Availability for HA



4. Click **Finish** once the installation is completed successfully.

Configuring Yarn (MR2 Included) and HDFS Services

The parameters in Table 9 are used for Cisco UCS S3260 Storage Server 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 9 YARN

Service	Value
mapreduce.map.memory.mb	3GiB
mapreduce.reduce.memory.mb	3GiB
mapreduce.map.java.opts.max.heap	2560 MiB
yarn.nodemanager.resource.memorymb	180 GiB
yarn.nodemanager.resource.cpu-vcores	32
yarn.scheduler.minimum-allocation-mb	4 GiB
yarn.scheduler.maximum-allocation-mb	180 GiB
yarn.scheduler.maximum-allocation-vcores	48
mapreduce.task.io.sort.mb	256 MiB

Table 10 HDFS

dfs.datanode.failed.volumes.tolerated	6
dfs.datanode.du.reserved	50 GiB
dfs.datanode.data.dir.perm	755
Java Heap Size of Namenode in Bytes	2628 MiB
dfs.namenode.handler.count	54
dfs.namenode.service.handler.count	54
Java Heap Size of Secondary namenode in Bytes	2628 MiB

Configuring Spark

The two main resources that Spark (and YARN) are dependent on are *CPU* and *memory*. Disk and network I/O, of course, play a part in Spark performance as well, but neither Spark nor YARN currently can actively manage them. Every Spark executor in any application has the same fixed number of cores and same fixed heap size. The number of cores can be specified with the

executor-cores flag when invoking spark-submit, spark-shell, and pyspark from the command line, or by setting the spark.executor.cores property in the spark-defaults.conf file or in the SparkConf object.

And the heap size can be controlled with the executor-memory flag or the spark.executor.memory property. The cores property controls the number of concurrent tasks an executor can run, executor-cores = 5 mean that each executor can run a maximum of five tasks at the same time. The memory property impacts the amount of data Spark can cache, as well as the maximum sizes of the shuffle data structures used for grouping, aggregations, and joins.

The num-executors command-line flag or spark.executor.instances configuration property control the number of executors requested. Dynamic Allocation can be enabled from CDH5.4 instead setting the spark.dynamicAllocation.enabled to true. Dynamic allocation enables a Spark application to request executors when there is a backlog of pending tasks and free up executors when idle.

Asking for five executor cores will result in a request to YARN for five virtual cores. The memory requested from YARN is a little more complex for a couple reasons:

- executor-memory/spark.executor.memory controls the executor heap size, but JVMs can also use some memory off heap, for example for VM overhead, interned Strings and direct byte buffers. The value of the spark.yarn.executor.memoryOverhead property is added to the executor memory to determine the full memory request to YARN for each executor. It defaults to max (384, 0.10 * spark.executor.memory).
- YARN may round the requested memory up a little. YARN's yarn.scheduler.minimum-allocation-mb and yarn.scheduler.increment-allocation-mb properties control the minimum and increment request values respectively.
- The application master is a non-executor container with the special capability of requesting containers from YARN, takes up resources of its own that must be budgeted in. In *yarn-client* mode, it defaults to a 1024MB and one vcore. In *yarn-cluster* mode, the application master runs the driver, so it's often useful to add its resources with the –driver-memory and –driver-cores properties.
- Running executors with too much memory often results in excessive garbage collection delays. 64GB is a rough guess at a good upper limit for a single executor.
- A good estimate is that at most five tasks per executor can achieve full write throughput, so it's good to keep the number of cores per executor around that number.
- Running tiny executors (with a single core and just enough memory needed to run a single task, for example) throws away the benefits that come from running multiple tasks in a single JVM. For example, broadcast variables need to be replicated once on each executor, so many small executors will result in many more copies of the data.

Tuning Resource Allocations for Spark

An example of configuring a Spark application to use as much of the cluster as possible is shown below. We are using an example cluster with 16 nodes running NodeManagers, each equipped with 56 cores and 256 GB of memory.

yarn.nodemanager.resource.memory-mb and yarn.nodemanager. resource.cpu-vcores should be set to 180 * 1024 = 184320 (megabytes) and 48 respectively. And use the following when submitting a spark job.

- spark.executor.instances/num-executors = 63
- spark.executor.cores/--executor-cores = 5
- spark.executor.memory/--executor-memory = 41G

This configuration results in four executors on all nodes except for the one with the AM, which will have three executors.

- executor-memory is derived as (180/4 executors per node) = 45; 45 * 0.10 = 4.5 45 − 4.5 ~ 40.

For taking care of long running processes use 2G for the spark driver:

- spark.driver.memory = 2G

For Submitting a Job

Use the following command for submitting and running a Spark 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 a number of tasks, each of which processes data sequentially. In tuning Spark jobs, this parallelism number is the most important parameter in determining performance. The number of tasks in a stage is the same as the number of partitions in the last RDD in the stage. The number of partitions in an RDD is the same as the number of partitions in the RDD on which it depends, with a couple exceptions: the coalesce transformation allows creating an RDD with fewer partitions than its parent RDD, the union transformation creates an RDD with the sum of its parents' number of partitions, and Cartesian creates an RDD with their product.

RDDs produced by a file have their partitions determined by the underlying MapReduce InputFormat that's used. Typically there will be a partition for each HDFS block being read. Partitions for RDDs produced by parallelize come from the parameter given by the user, or spark.default.parallelism if none is given.

The primary concern is that the number of tasks will be too small. If there are fewer tasks than slots available to run them in, the stage won't be taking advantage of all the CPU available.

If the stage in question is reading from Hadoop, your options are:

- Use the repartition transformation, which will trigger a shuffle.
- Configure your InputFormat to create more splits.
- Write the input data out to HDFS with a smaller block size.

If the stage is getting its input from another stage, the transformation that triggered the stage boundary will accept a numPartitions argument.

The most straightforward way to tune the number of partitions is experimentation: Look at the number of partitions in the parent RDD and then keep multiplying that by 1.5 until performance stops improving.

In contrast with MapReduce for Spark when in doubt, it's almost always better to be on the side of a larger number of tasks (and thus partitions).

Shuffle Performance Improvement

spark.shuffle.compress true (compress map output files)

spark.broadcast.compress true(compress broadcast variables before sending them)

spark.io.compression.codec org.apache.spark.io.SnappyCompressionCodec (codec used to compress internal data such as RDD partitions, broadcast variables and shuffle outputs)

spark.shuffle.spill.compress true (Whether to compress data spilled during shuffles.)

spark.shuffle.io.numConnectionsPerPeer 4 (Connections between hosts are reused in order to reduce connection buildup for large clusters. For clusters with many hard disks and few hosts, this may result in insufficient concurrency to saturate all disks, and so users may consider increasing this value.)

spark.shuffle.file.buffer 64K (Size of the in-memory buffer for each shuffle file output stream. These buffers reduce the number of disk seeks and system calls made in creating intermediate shuffle file).

Improving Serialization Performance

Serialization plays an important role in the performance of any distributed application. Often, this will be the first thing that should be tuned to optimize a Spark application.

spark.serializer org.apache.spark.serializer.KryoSerializer (when speed is necessary)

spark.kryo.referenceTracking false

spark.kryoserializer.buffer 2000 (If the objects are large, may need to increase the size further to fit the size of the object being deserialized).

SparkSQL is ideally suited for mixed procedure jobs where SQL code is combined with Scala, Java, or Python programs. In general the SparkSQL command line interface is used for single user operations and ad hoc queries.

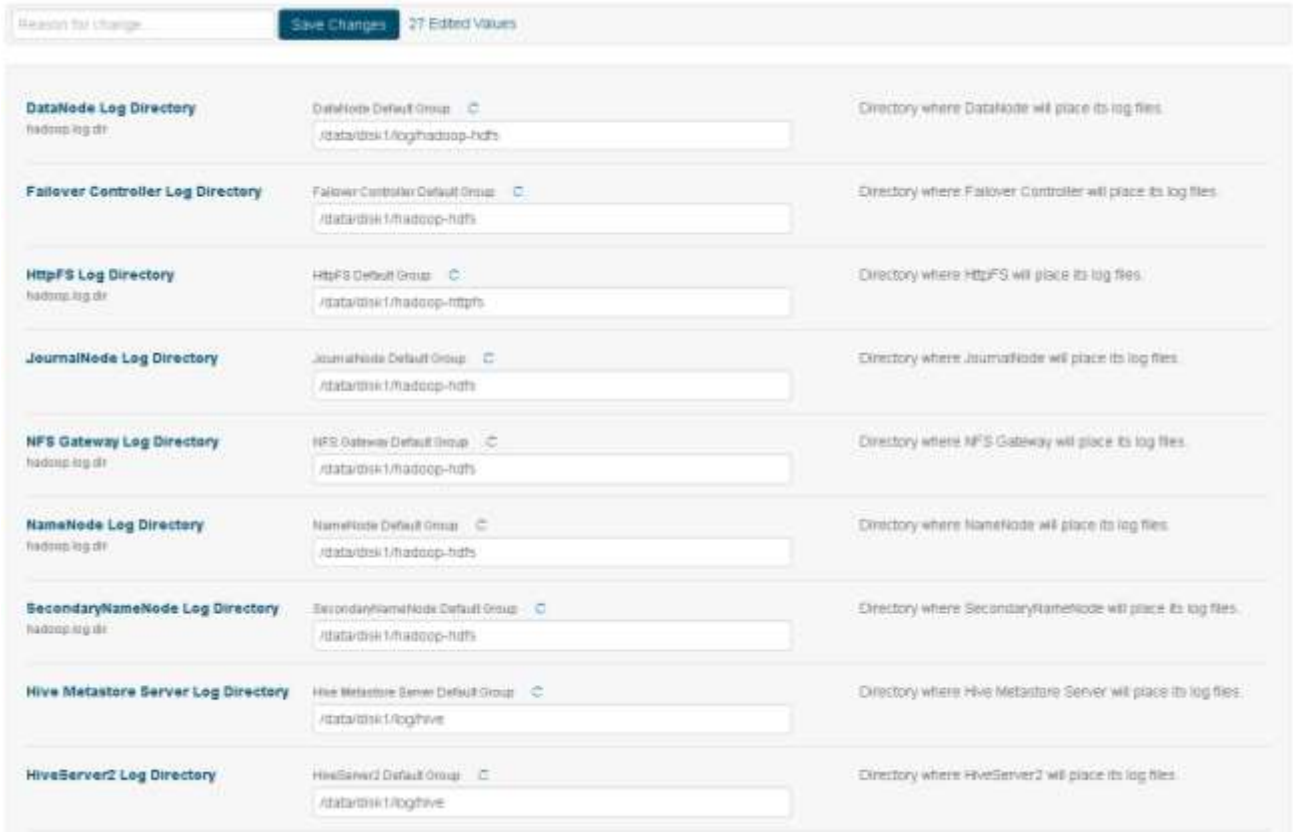
For multi-user SparkSQL environments, it is recommended to use a Thrift server connected via JDBC.

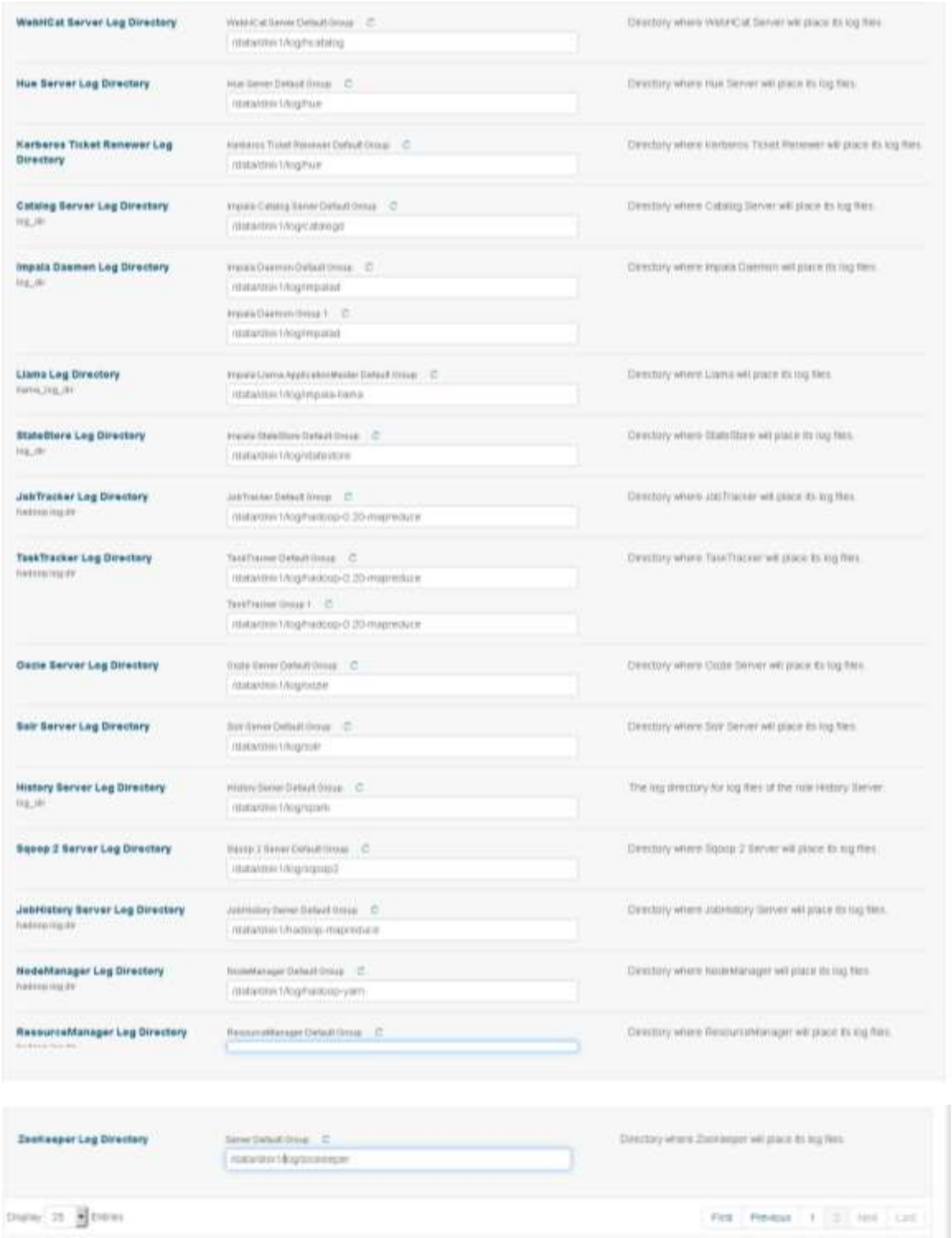
Changing the Log Directory for All Applications

To change the default log from the /var prefix to /data/disk1, complete the following steps:

1. Log into the Cloudera home page and click My Clusters.
2. From the configuration drop down menu select “All Log Directories”

Figure 176 Changes





3. Click Save Changes.

Bill of Materials

This section provides the BOM for the 16 nodes. See Table 11 Bill of Materials for the Cisco UCS Fabric Interconnect 6332, Table 12 Bill of Materials for the Cisco UCS C240M4 Rack Server, Table 13 for the Cisco UCS S3260 Storage Server Base Rack, Table 14 Bill of Materials for Cisco UCS S3260 Storage Server Capacity Rack, and Table 15 and Table 16 for software components. Table 17 lists Cloudera SKUs available from Cisco.

Table 11 Bill of Materials for Cisco UCS Fabric Interconnect 6332

Part Number	Description	Quantity
UCS-FI-6332UP-UPG	UCS 6332UP 2RU Fabric Int/No PSU/48 UP/ 18p LIC	2
CON-SNT-FI6332UP	SMARTNET 8X5XNBD UCS 6332UP 2RU Fabric Int/2 PSU/4 Fans	2
SFP-H40GB-CU3M	40GBASE-CU SFP+ Cable 3 Meter	8
UCS-ACC-6296UP	UCS 6296UP Chassis Accessory Kit	2
UCS-PSU-6296UP-AC	UCS 6296UP Power Supply/100-240VAC	4
N10-MGT014	UCS Manager v3.1	2
UCS-L-6200-10G-C	2rd Gen FI License to connect C-direct only	62
UCS-BLKE-6200	UCS 6200 Series Expansion Module Blank	6
UCS-FAN-6296UP	UCS 6296UP Fan Module	8
CAB-N5K6A-NA	Power Cord 200/240V 6A North America	4
UCS-FI-E16UP	UCS 6200 16-port Expansion module/16 UP/ 8p LIC	4
RACK-UCS2	Cisco R42610 standard rack w/side panels	1
RP208-30-1P-U-2=	Cisco RP208-30-U-2 Single Phase PDU 20x C13 4x C19 (Country Specific)	2
CON-UCW3-RPDUX	UC PLUS 24X7X4 Cisco RP208-30-U-X Single Phase PDU 2x (Country Specific)	6

Table 12 Bill of Materials for Cisco UCS C240M4 Rack Server

Part Number	Description	Quantity

UCS-SL-CPA4-P2	Performance Optimized Option 2 Cluster	1
UCSC-C240-M4SX	UCS C240 M4 SFF 24 HD w/o CPU, memory, HD, PCIe, PS, rail kit w/expander	3
UCSC-MRAID12G	Cisco 12G SAS Modular Raid Controller	3
UCSC-MRAID12G-2GB	Cisco 12Gbps SAS 2GB FBWC Cache module (Raid 0/1/5/6)	3
UCSC-MLOM-CSC-02	Cisco UCS VIC1387 VIC MLOM - Dual Port 40Gb Ethernet QSFP ports	3
CAB-9K12A-NA	Power Cord 125VAC 13A NEMA 5-15 Plug North America	6
UCSC-PSU2V2-1200W	1200W/800W V2 AC Power Supply for 2U C-Series Servers	6
UCSC-RAILB-M4	Ball Bearing Rail Kit for C240 M4 rack servers	3
UCSC-HS-C240M4	Heat Sink for UCS C240 M4 Rack Server	6
UCSC-SCCBL240	Supercap cable 250mm	3
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	6
UCS-MR-1X161RV-A	16GB DDR4-2400-MHz RDIMM/PC4-19200/single rank/x4/1.2v	48
UCS-HD18TB10KS4K	1.2 TB 12G SAS 10K rpm SFF HDD (4K)	36
UCS-SD240GBKS4-EB	240 GB 2.5 inch Enterprise Value 6G SATA SSD (BOOT)	6
UCSC-PCI-1C-240M4	Right PCI Riser Bd (Riser 1) 2onbd SATA bootdrvs+ 2PCI slts	3

Table 13 Bill of Materials for Cisco UCS S3260 Storage Server Base Rack

Part Number	Description	Quantity
UCSC-C3260	Cisco UCS C3260 Base Chassis w/4x PSU, SSD, Railkit	8
CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	32
UCS-C3K-HD4TB	UCS C3000 4TB NL-SAS 7200 RPM 12Gb HDD w Carrier- Top Load	48

UCSC-C3160-BEZEL	Cisco UCS C3160 System Bezel	8
UCSC-C3X60-RAIL	UCS C3X60 Rack Rails Kit	8
UCSC-PSU1-1050W	UCS C3X60 1050W Power Supply Unit	32
UCSC-C3K-M4SRB	UCS C3000 M4 Server Node for Intel E5-2600 v4	8
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	16
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	64
UCS-C3K-M4RAID	Cisco UCS C3000 RAID Controller M4 Server w 4G RAID Cache	8
UCSC-HS-C3X60	Cisco UCS C3X60 Server Node CPU Heatsink	16
UCSC-C3K-M4SRB	UCS C3000 M4 Server Node for Intel E5-2600 v4	8
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	16
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	64
UCS-C3K-M4RAID	Cisco UCS C3000 RAID Controller M4 Server w 4G RAID Cache	8
UCSC-HS-C3X60	Cisco UCS C3X60 Server Node CPU Heatsink	16
UCS-S3260-42HD4	Cisco UCS C3X60 Three row of drives containing 42 x 4TB (Tot	8
UCS-C3K-HD4TB	UCS C3000 4TB NL-SAS 7200 RPM 12Gb HDD w Carrier- Top Load	336
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	8
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	8
UCS-C3X60-G2SD48	UCSC C3X60 480GB Boot SSD (Gen 2)	32

Table 14 Bill of Materials for Cisco UCS S3260 Storage Server Capacity Rack

Part Number	Description	Quantity
UCSC-C3260	Cisco UCS C3260 Base Chassis w/4x PSU, SSD, Railkit	8

Part Number	Description	Quantity
CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	32
UCSC-C3X60-HD8TB	UCSC 3X60 8TB NL-SAS 7.2K Helium HDD with HDD Carrier	48
UCSC-C3160-BEZEL	Cisco UCS C3160 System Bezel	8
UCSC-C3X60-RAIL	UCS C3X60 Rack Rails Kit	8
UCSC-PSU1-1050W	UCS C3X60 1050W Power Supply Unit	32
UCSC-C3K-M4SRB	UCS C3000 M4 Server Node for Intel E5-2600 v4	8
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	16
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	64
UCS-C3K-M4RAID	Cisco UCS C3000 RAID Controller M4 Server w 4G RAID Cache	8
UCSC-HS-C3X60	Cisco UCS C3X60 Server Node CPU Heatsink	16
UCSC-C3K-M4SRB	UCS C3000 M4 Server Node for Intel E5-2600 v4	8
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	16
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	64
UCS-C3K-M4RAID	Cisco UCS C3000 RAID Controller M4 Server w 4G RAID Cache	8
UCSC-HS-C3X60	Cisco UCS C3X60 Server Node CPU Heatsink	16
UCSC-C3X60-42HD8	UCS C3X60 3 rows of 8TB NL-SAS7200 RPM SAS-3 (42Total) 336TB	8
UCSC-C3X60-HD8TB	UCSC 3X60 8TB NL-SAS 7.2K Helium HDD with HDD Carrier	336
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	8
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	8
UCS-C3X60-G2SD48	UCSC C3X60 480GB Boot SSD (Gen 2)	32

Note: Both Cisco UCS S3260 Storage Server Basic Rack and Cisco UCS S3260 Storage Server Capacity Rack Bundle comes with 24 x 4TB Disk Drives, supports up to 28 x 4 TB.

Table 15 Red Hat Enterprise Linux License

Red Hat Enterprise Linux		
RHEL-2S2V-3A	Red Hat Enterprise Linux	19
CON-ISV1-EL2S2V3A	3 year Support for Red Hat Enterprise Linux	19

Table 16 Cloudera Software

Cloudera Software edition needed for this CVD		
Cloudera Enterprise Flex Edition	UCS-BD-CEDHC-BZ=	19
Cloudera Enterprise Data Hub Edition	UCS-BD-CEDHC-GD=	19

Table 17 Cloudera SKU's Available at Cisco

Cisco TOP SKU	Cisco PID with Duration	Product Name
UCS-BD-CEBN-BZ=	UCS-BD-CEBN-BZ-3Y	Cloudera Enterprise Basic Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEBN-BZI=	UCS-BD-CEBN-BZI-3Y	Cloudera Enterprise Basic Edition + Indemnification, Node License, Bronze Support - 3 Year
UCS-BD-CEBN-GD=	UCS-BD-CEBN-GD-3Y	Cloudera Enterprise Basic Edition, Node License, Gold Support - 3 Year
UCS-BD-CEBN-GDI=	UCS-BD-CEBN-GDI-3Y	Cloudera Enterprise Basic Edition + Indemnification, Node License, Gold Support - 3 Year
UCS-BD-CEDEN-BZ=	UCS-BD-CEDEN-BZ-3Y	Cloudera Enterprise Data Engineering Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEDEN-GD=	UCS-BD-CEDEN-GD-3Y	Cloudera Enterprise Data Engineering Edition, Node License, Gold Support - 3 Year
UCS-BD-CEODN-BZ=	UCS-BD-CEODN-BZ-3Y	Cloudera Enterprise Operational Database Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEODN-GD=	UCS-BD-CEODN-GD-2Y	Cloudera Enterprise Operational Database Edition, Node License, Gold Support - 2 Year
UCS-BD-CEODN-GD=	UCS-BD-CEODN-GD-3Y	Cloudera Enterprise Operational Database Edition, Node License, Gold Support - 3 Year
UCS-BD-CEADN-BZ=	UCS-BD-CEADN-BZ-3Y	Cloudera Enterprise Analytical Database Edition, Node License, Bronze Support - 3 Year

Cisco TOP SKU	Cisco PID with Duration	Product Name
UCS-BD-CEADN-GD=	UCS-BD-CEADN-GD-3Y	Cloudera Enterprise Analytical Database Edition, Node License, Gold Support - 3 Year
UCS-BD-CEDHN-BZ=	UCS-BD-CEDHN-BZ-3Y	Cloudera Enterprise Data Hub Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEDHN-GD=	UCS-BD-CEDHN-GD-3Y	Cloudera Enterprise Data Hub Edition, Node License, Gold Support - 3 Year
UCS-BD-CEBC-BZ=	UCS-BD-CEBC-BZ-3Y	Cloudera Enterprise Basic Edition, Capacity License, Bronze Support - 3 Year
UCS-BD-CEBC-BZI=	UCS-BD-CEBC-BZI-3Y	Cloudera Enterprise Basic Edition + Indemnification, Capacity License, Bronze Support - 3 Year
UCS-BD-CEBC-GD=	UCS-BD-CEBC-GD-3Y	Cloudera Enterprise Basic Edition, Capacity License, Gold Support - 3 Year
UCS-BD-CEBC-GDI=	UCS-BD-CEBC-GDI-3Y	Cloudera Enterprise Basic Edition + Indemnification, Capacity License, Gold Support - 3 Year
UCS-BD-CEDEC-BZ=	UCS-BD-CEDEC-BZ-3Y	Cloudera Enterprise Data Engineering Edition, Capacity License, Bronze Support - 3 Year
UCS-BD-CEDEC-GD=	UCS-BD-CEDEC-GD-3Y	Cloudera Enterprise Data Engineering Edition, Capacity License, Gold Support - 3 Year
UCS-BD-CEODC-BZ=	UCS-BD-CEODC-BZ-3Y	Cloudera Enterprise Operational Database Edition, Capacity License, Bronze Support - 3 Year
UCS-BD-CEODC-GD=	UCS-BD-CEODC-GD-3Y	Cloudera Enterprise Operational Database Edition, Capacity License, Gold Support - 3 Year
UCS-BD-CEADC-BZ=	UCS-BD-CEADC-BZ-3Y	Cloudera Enterprise Analytical Database Edition, Capacity License, Bronze Support - 3 Year
UCS-BD-CEADC-GD=	UCS-BD-CEADC-GD-3Y	Cloudera Enterprise Analytical Database Edition, Capacity License, Gold Support - 3 Year
UCS-BD-CEDHC-BZ=	UCS-BD-CEDHC-BZ-3Y	Cloudera Enterprise Data Hub Edition, Capacity License, Bronze Support - 3 Year
UCS-BD-CEDHC-GD=	UCS-BD-CEDHC-GD-3Y	Cloudera Enterprise Data Hub Edition, Capacity License, Gold Support - 3 Year

About Authors

Chinmayi Narasimhadevara, Big Data Software Engineer, Data Center Solutions Group, Cisco Systems, Inc. Chinmayi's focus areas are solutions and emerging trends in Big Data and Analytics related technologies and infrastructure in the Data Center.

Acknowledgements

- Manan Trivedi, Big Data Solutions Engineer, Data Center Solutions Group, Cisco Systems Inc.
- Karthik Kulkarni, Big Data Solutions Architect, Data Center Solutions Group, Cisco Systems Inc.
- Amrit Kharel, Network Engineer, Data Center Solutions Group, Cisco Systems Inc.
- Dwai Lahari, Senior Solutions Architect at Cloudera.
- Alex Moundalexis, Senior Solutions Architect, Cloudera.
- Rick Hallihan, Solutions Architect, Cloudera.
- Barbara Dixon, Technical Writer, Data Center Solutions Group, Cisco Systems, Inc.
- Michael Kaleta, Director, Business Development, Cloudera
- Sandi Lii, Sr. Director, Partner Marketing, Cloudera