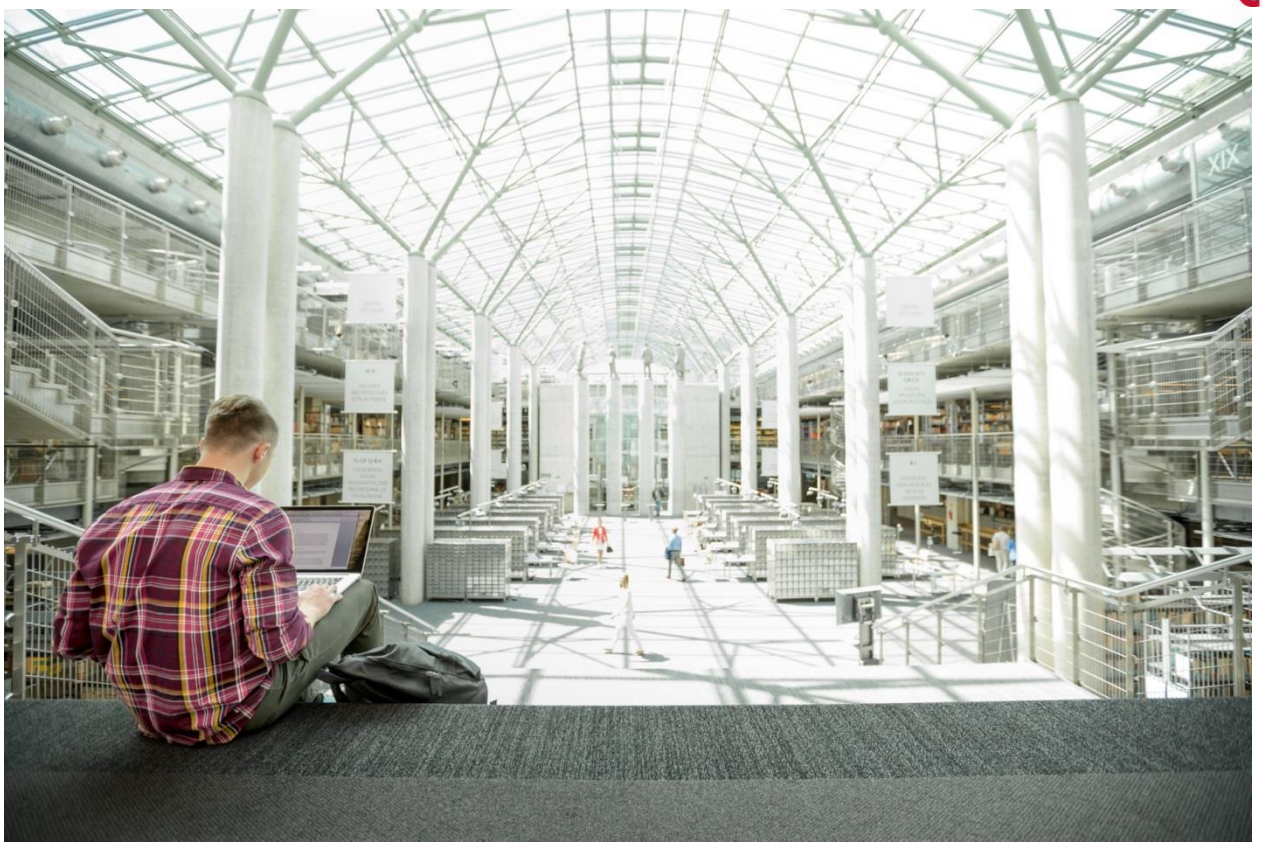


Enterprise Data Lake with Cisco UCS S3260 Storage Server and Hortonworks Data Platform

Last Updated: October 30, 2016

Cisco Validated Design

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

Data is being generated on 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. Business Data Lakes centered around Hadoop-based storage are becoming common. Once there is ability to store and load these different types of datasets the next step is to gain business value by processing, analyzing gaining insights and taking action on data.

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) 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. 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 Cisco 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 Hortonworks Data Platform is a tested, reliable method for big data systems. Together, they power the next generation architecture for taking full advantage of the power of data to accelerate decision-making and innovation.

Solution Overview

Introduction

Massive amounts of information and data are being generated every second. Big Data is not just some abstract concept but it has become a priority for many organizations. This solution unlocks the value of big data while maximizing existing investments.

Apache Hadoop is the most popular big data framework. The technology is evolving rapidly to enable decisions while working large volumes of data. A solution that can work effectively to enable processing of multiple petabytes of data to get actionable insights is needed.

The Cisco UCS S3260 Storage Server is specifically designed to address this with its modular design and unique capabilities. It is designed with the flexibility to handle both high-capacity and high-performance workloads.

Solution

This solution is a simple and linearly scalable architecture that provides data processing on HortonwWorks Data Platform that enables data processing with a centrally managed automated Hadoop deployment, providing all the benefits of the Cisco Unified Computing System (UCS).

Some of the features of this solution include:

- Infrastructure for both big data and large scale analytics
- Simplified infrastructure management via the Cisco UCS Manager
- Architectural scalability - linear scaling based on network, storage and compute requirements.
- Usage of Hortonworks Data Platform (HDP) for comprehensive cluster monitoring and management

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 to power data processing on dense compute and storage servers for big data deployments.

Audience

This document describes the architecture and deployment procedures for Hortonworks Data Platform (HDP) on an Enterprise data lake using 8 Cisco UCS S3260 Storage Servers with two C3x60 M4 server nodes each as worker nodes, and 3 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 Hortonworks Data Platform (HDP) on the Cisco Unified Computing System (UCS) using Cisco UCS S3260 Storage Servers.

Solution Summary

This CVD describes in detail the process for installing Hortonworks 2.5.0 including the configuration details of the cluster.

The configuration using Cisco UCS S3260 Storage Servers as data nodes and Cisco UCS C240 M4 Rack Servers as management nodes, is shown in Error! Reference source not found.This configuration upports 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 Chassis, 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])	• 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

Scaling/Performance Options

The server supports upto 28 LFF disk drives of various capacities. The base reference configuration consists of 24 LFF drives each with 4 TB capacity. The storage capacities can be extended as shown in Table 2 below.

Table 2 Disk Drive Options

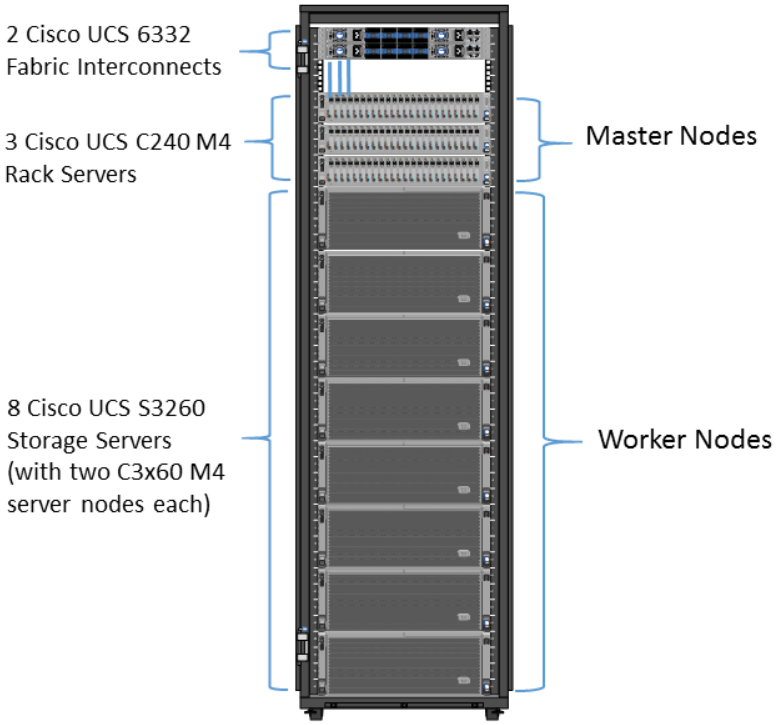
HDD Type	Capacity with 24 disk drives	Capacity with 28 disk drives
4 TB 7200 −rpm LFF SAS drives	1.54 PetaBytes	1.8 PetaBytes
6 TB 7200 −rpm LFF SAS drives	2.38 PetaBytes	2.68 PetaBytes
8 TB 7200 −rpm LFF SAS drives	3.07 PetaBytes	3.58 PetaBytes
10 TB 7200 ~prm LFF SAS drives	3.84 PetaBytes	4.48 PetaBytes

Technology Overview

Reference Architecture

Figure 1 shows the base configuration of the enterprise data lake cluster. It is comprised of 16 data nodes (worker nodes) using 8 Cisco UCS S3260 Storage Servers and 3 management nodes (master nodes) using Cisco UCS C240 M4 Rack Servers.

Figure 1 Reference Architecture



Note: This CVD describes the installation process of HDP 2.5.0 for a 16 node cluster (with three management nodes).

Table 3 Components

Component	Description
Connectivity	2 x Cisco UCS 6332 32-Port Fabric Interconnects
HDP Nodes	8 x Cisco UCS S3260 Storage Servers 3 x Cisco UCS C240 M4 Rack Servers Hadoop NameNode/Secondary NameNode and Resource Manager and Data Nodes.

	Spark Executors are collocated on a Data Node. *Please refer to the Service Assignment section for specific service assignment and configuration details.

Cisco UCS S3260 Storage Server for Big Data and Analytics with HDP

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.

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 (supports 4TB, 6TB, 8TB and 10TB) 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 Server

Cisco UCS C240 M4 High-Density Rack Server (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. 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 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 shows the Cisco UCS C240 M4 Rack Server.

Figure 4 Cisco UCS C240 M4 Rack Server

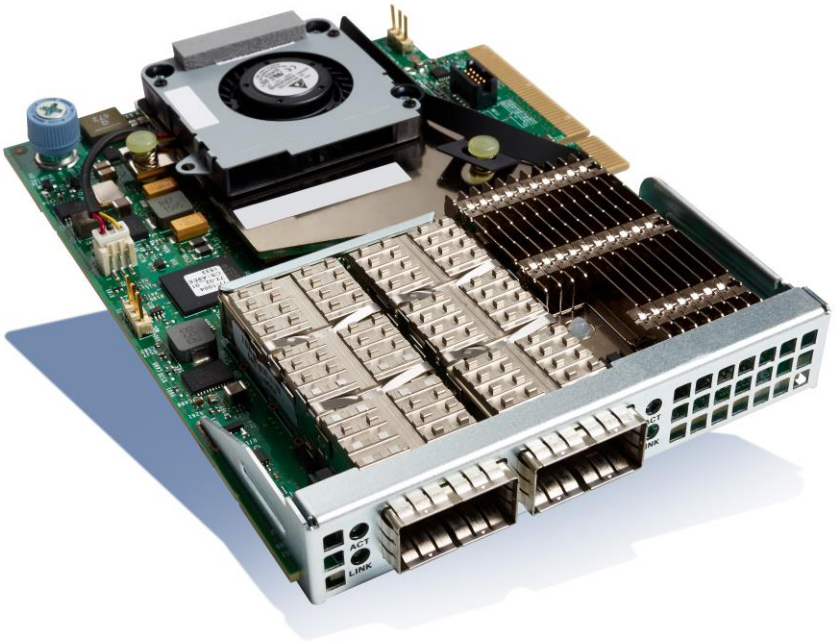
Figure 5 Back View of Cisco UCS C240 M4 Rack Server

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.

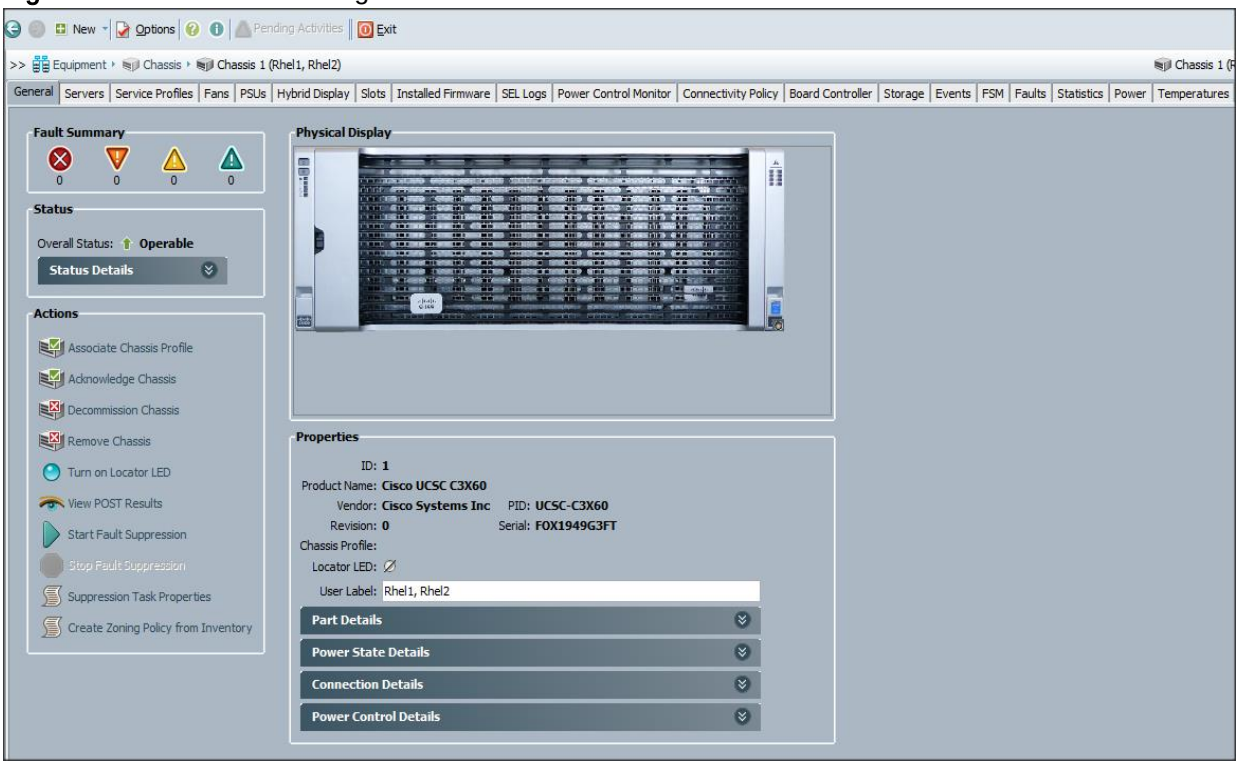
Figure 7 Cisco UCS 6332 32-Port Fabric Interconnect

Cisco UCS Manager

Cisco UCS Manager resides within the Cisco UCS 6300 Series Fabric Interconnect. It makes the system self-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



Hortonworks (HDP 2.5.0)

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

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

HDP for Data Access

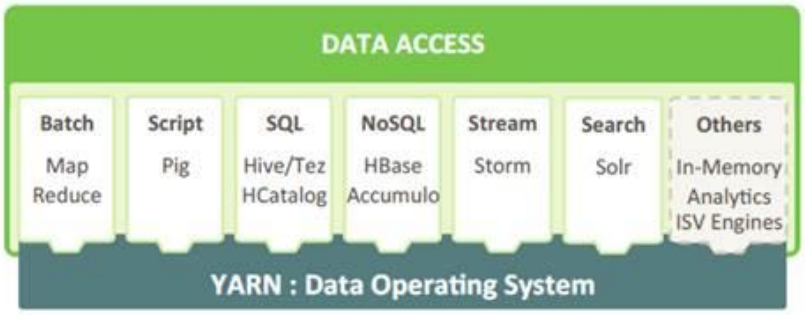
With YARN at its foundation, HDP provides a range of processing engines that allow users to interact with data in multiple and parallel ways, without the need to stand up individual clusters for each data set/application. Some applications require batch while others require interactive SQL or low-latency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

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

As shown in Figure 9, HDP Enterprise natively provides for the following data access types:

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

Figure 9 YARN



HDP Cluster Operations

HDP delivers a comprehensive set of completely open operational capabilities that provide both visibilities into cluster health as well as the ability to manage, monitor and configure resources.

- Apache Ambari – is a completely open framework to provision, manage and monitor Apache Hadoop clusters. It provides a simple, elegant UI that allows you to image a Hadoop cluster.
- Apache Oozie - provides a critical scheduling capability to organize and schedule jobs within Enterprise Hadoop across all data access points.

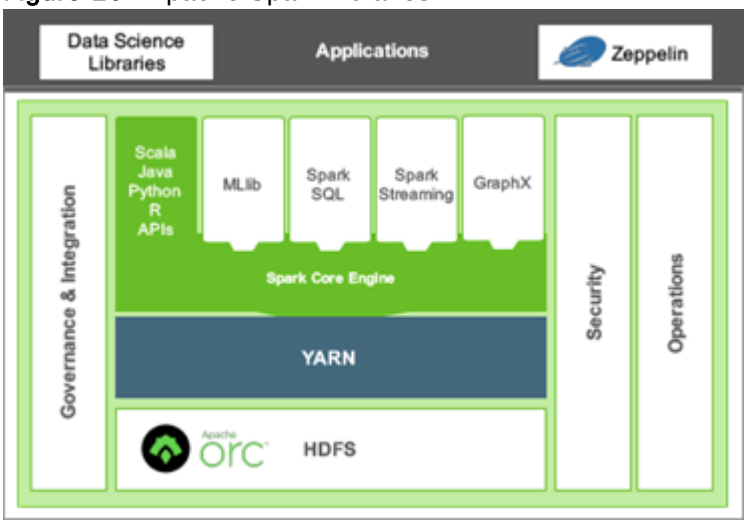
Apache Spark

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. The Cisco Unified Computing System 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.

Apache Spark enhances the existing big data environments by adding new capabilities to Hadoop or other big data deployments. The platform unifies a broad range of capabilities–batch processing, real-time stream processing, advanced analytic capabilities, and interactive exploration–that can intelligently optimize applications

Spark provides programmers with an application interface centered on a data structure called the resilient distributed dataset (RDD), a read-only set of data items distributed over a cluster of machines that is maintained in a fault-tolerant way.

Figure 10 Apache Spark Libraries



As shown in Figure 10 above, Apache Spark has a number of libraries:

- Apache Spark SQL/DataFrame API for querying structured data inside Spark programs.

- Apache Spark streaming offers Spark's core API enabling 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 can access diverse data sources including HDFS, Cassandra, HBase, and S3. 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-based placement policies, and enabling security by running on Kerberos enabled clusters.

Some common use cases that are popular in the field with Apache Spark are shown in Table 4.

Table 4 Apache Spark Use Cases

Insurance	Optimize claims reimbursement processes by using Spark's machine learning capabilities to process and analyze all claims.
Healthcare	Build a Patient Care System using Spark Core, Streaming and SQL.
Retail	Use Spark to analyze point-of-sale data and coupon usage.
Internet	Use Spark's ML capability to identify fake profiles and enhance product matches to show their customers.
Banking	Use a machine learning model to predict the profile of retail banking customers for certain financial products.
Government	Analyze spending across geography, time and category.
Scientific Research	Analyze earthquake events by time, depth and geography to predict future events.
Investment Banking	Analyze intra-day stock prices to predict future price movements.
Geospatial Analysis	Analyze Uber trips by time and geography to predict future demand and pricing.
Twitter Sentiment Analysis	Analyze large volumes of Tweets to determine positive, negative or neutral sentiment for specific organizations and products.
Airlines	Build a model for predicting airline travel delays.
Devices	Predict likelihood of a building exceeding threshold temperatures.

Solution Design

Requirements

This CVD describes the architecture and deployment procedures to install Hortonworks (HDP 2.5.0) on eight Cisco UCS S3260 Storage Servers each with two C3x60 M4 server nodes each as Hadoop data nodes, and three Cisco UCS C240 M4 Rack servers as Hadoop Management nodes for Big Data and Analytics. The solution goes into detail configuring HDP 2.5.0 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 racks
- Two Vertical Power distribution units (PDUs) (Country Specific)

Rack and PDU Configuration

Each rack consists of two vertical PDUs. The rack consists of 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.

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

Table 5 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

Position	Devices
27	Server
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

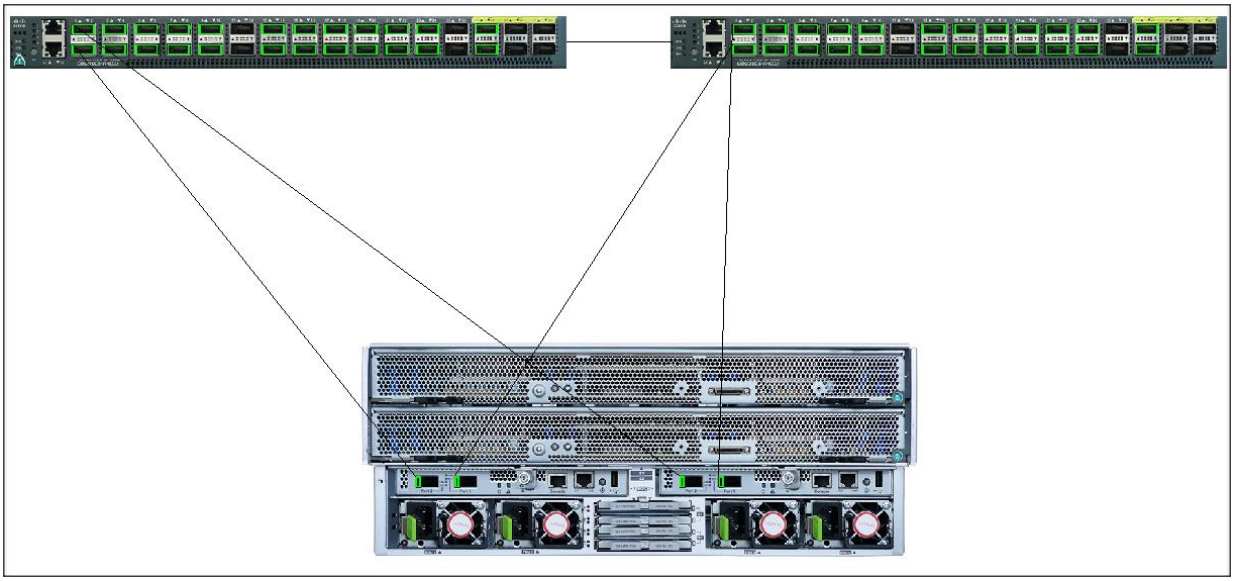
Port Type	Port Number
Network	32
Server	1-19

Server Configuration and Cabling for 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 11 illustrates the port connectivity between the Fabric Interconnect, and Cisco UCS S3260 Storage Server Chassis. Eight Cisco UCS S3260 Storage Server Chassis are used in single rack configurations.

Figure 11 Fabric Topology for Cisco UCS S3260 Storage Server with UCSC-C3K-M4 SRB server blades



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

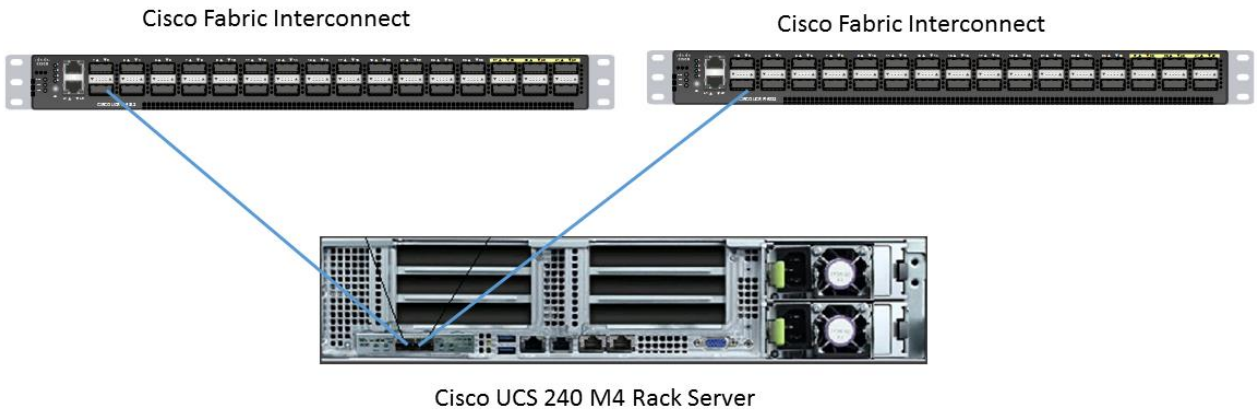
Figure 11 depicts a connectivity between Cisco UCS S3260 Storage Server chassis and Cisco UCS 6300 Fabric Interconnect. Each chassis has two C3x60 M4 server nodes. Each link in the figure represents a 40 Gigabit Ethernet link from Cisco UCS S3260 Storage Server chassis connecting to a Fabric Interconnect. Every chassis is connected to both Fabric Interconnects represented with dual link

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 12). 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.

Figure 12 Fabric Topology for Cisco UCS C240 M4 Rack Server



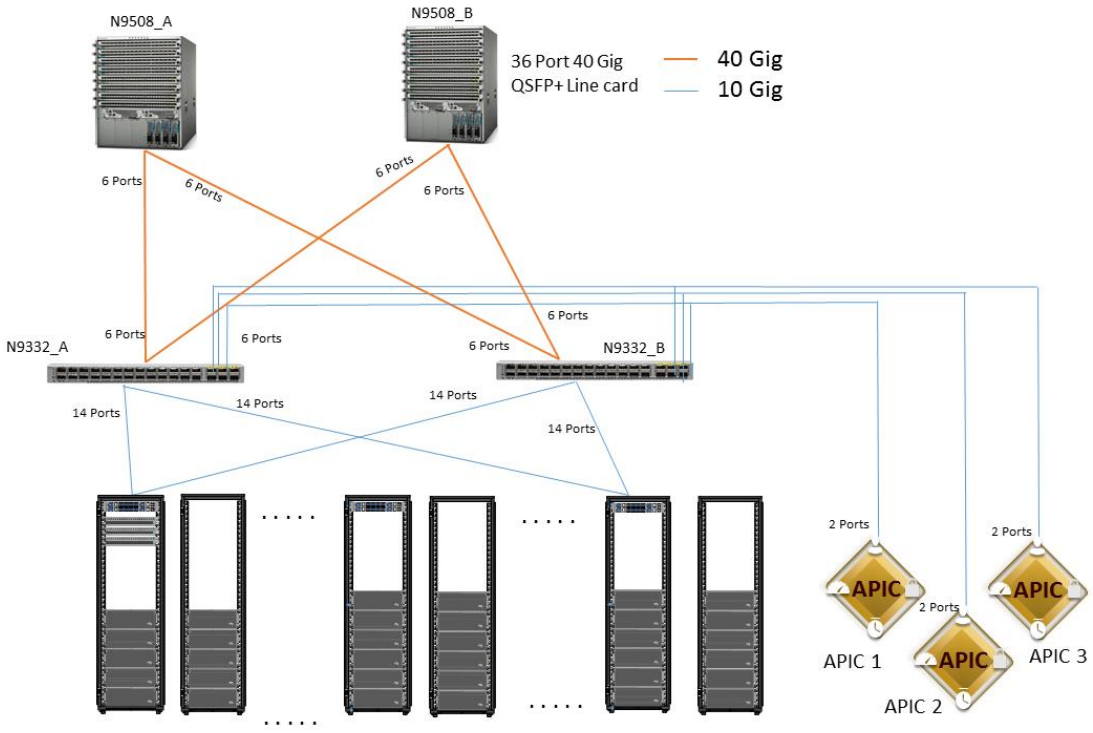
Cisco UCS S3260 Storage Server Scaling with Cisco Application Centric Infrastructure (ACI)

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 Interconnect which are interconnected through 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 and two Nexus 9332PQ as the leaf switches and three APIC-L1 as an APIC appliance.

ACI Scaling Diagram



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 6 below.

Table 6 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

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 7 Leaf to Fabic 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.

Hortonworks (HDP 2.5.0)

The Hortonworks Distribution for Apache Hadoop version used is 2.5.0. For more information visit http://www.hortonworks.com

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

Table 8 The software version 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)
	Cisco UCS Manager	3.1(2b)

Layer	Component	Version or Release
	Hortonworks (HDP)	2.5.0

The latest drivers can be downloaded from 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 : 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.

Performing 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 `A` 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:

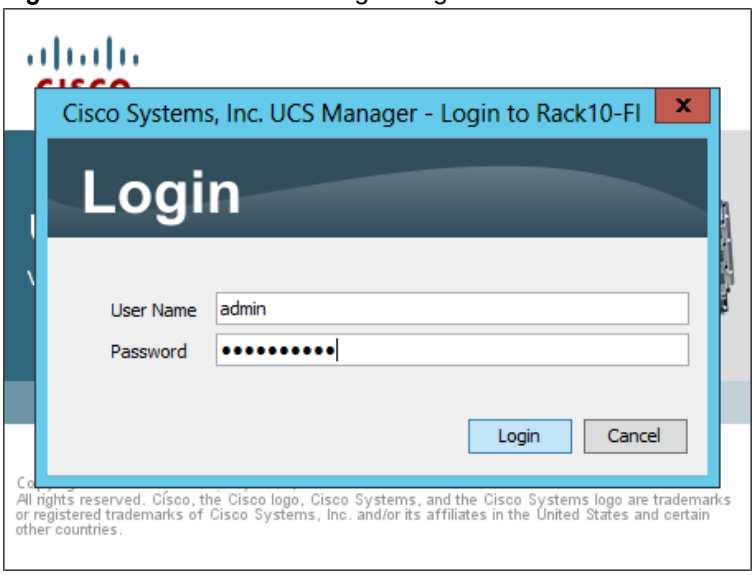
http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-6300-series-fabric-interconnects/index.html

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

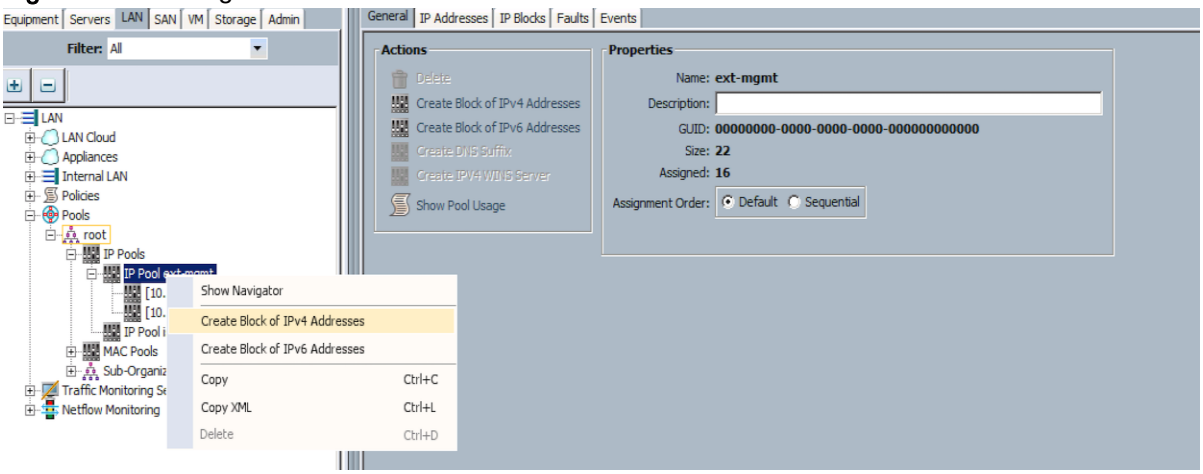


Adding a Block of IP Addresses for KVM Access

These steps provide details for creating a block of KVM IP addresses for server access in the Cisco UCS environment.

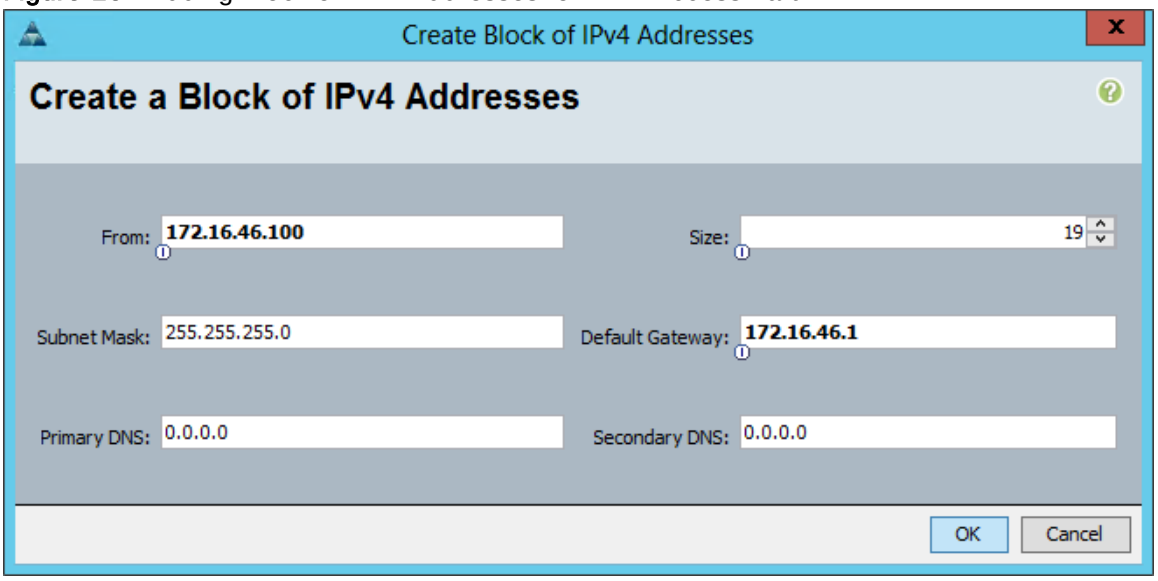
1. Select the `LAN` tab at the top of the left window, Figure 14.
2. Select `Pools > IpPools > Ip Pool ext-mgmt.`
3. Right-click `IP Pool ext-mgmt.`
4. Select `Create Block of IPv4 Addresses.`

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)

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 Port

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.

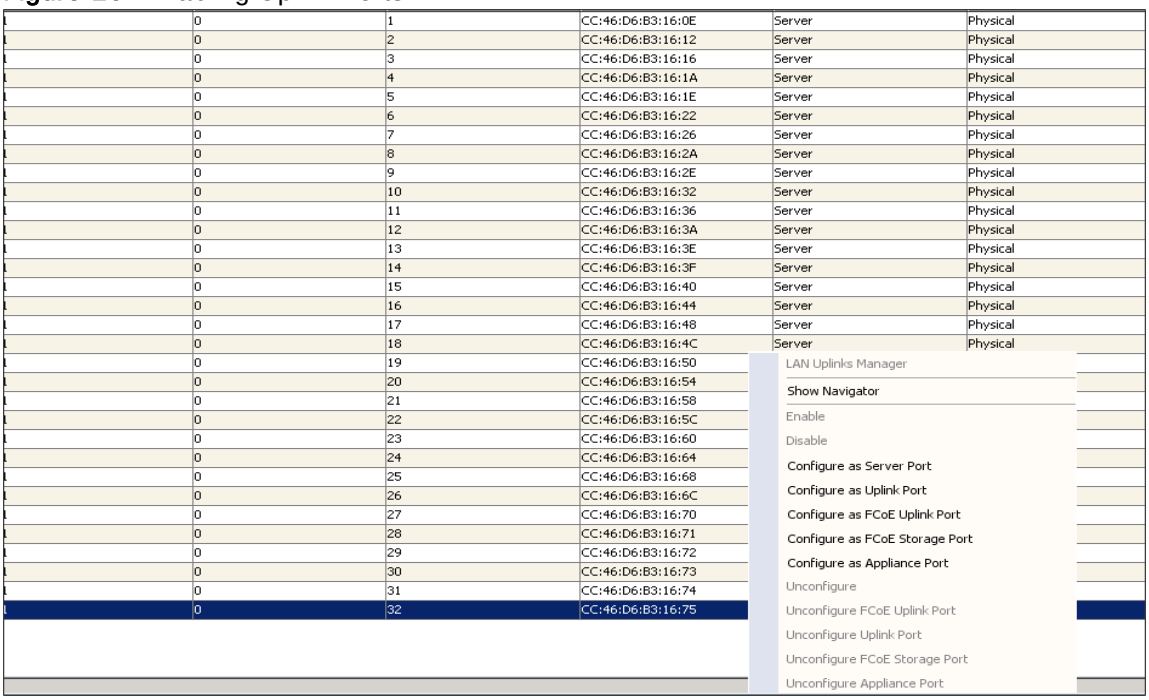
4. Select port 32 that are connected to the uplink switch, right-click, and then select Configure as Uplink Port. (Figure 16)

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

VLANs are configured as in shown in Table 9.

Table 9 VLAN Configurations

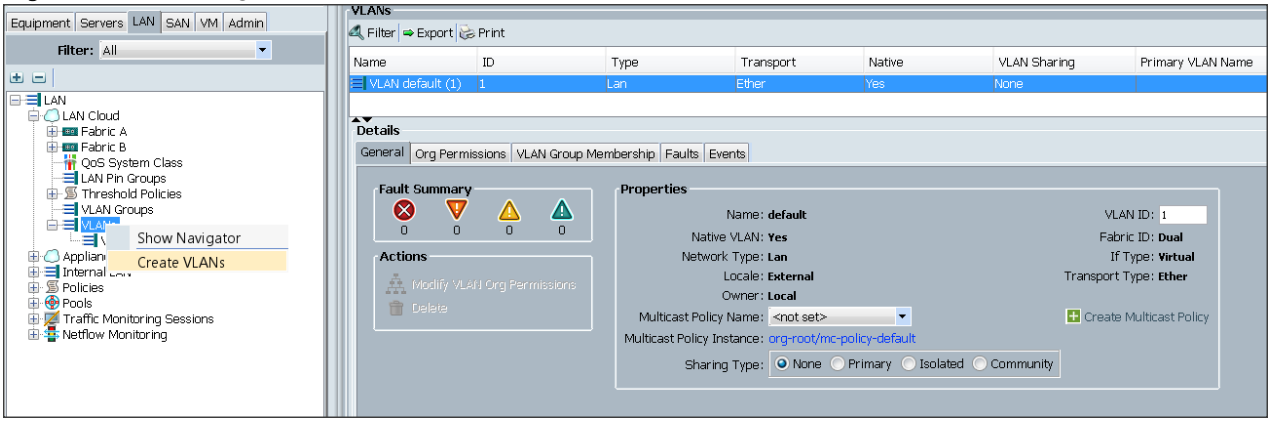
VLAN	NIC Port	Function
VLAN76	Eth0	Management & Data Traffic

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:

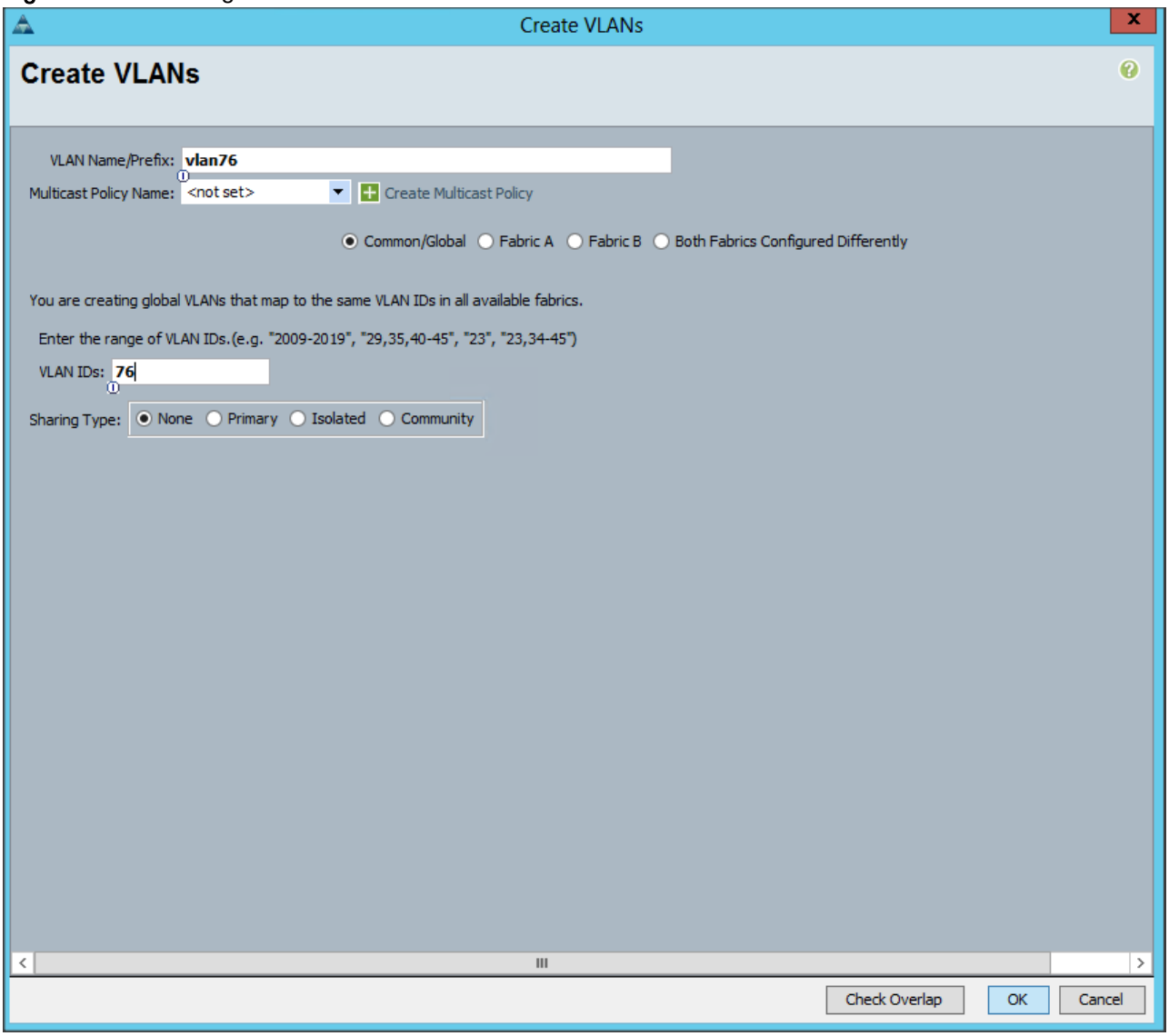
1. Select the `LAN` tab in the left pane in the UCSM GUI.
2. Select LAN > LAN Cloud > VLANs.
3. Right-click the `VLANs` under the root organization.
4. Select `Create VLANs` to create the VLAN. (Figure 17)

Figure 17 Creating a VLAN



5. Enter vlan76 for the `VLAN Name`. (Figure 18)
6. Keep multicast policy as <not set>.
7. Select `Common/Global` for vlan76.
8. Enter 76 in the `VLAN IDs` field for the Create VLAN IDs.
9. Click `OK` and then, click `Finish`.
10. Click `OK` in the success message box.

Figure 18 Creating VLAN for Data



11. Click `OK` and then, click `Finish`.

Enabling Server Ports

To enable server ports, complete the following steps:

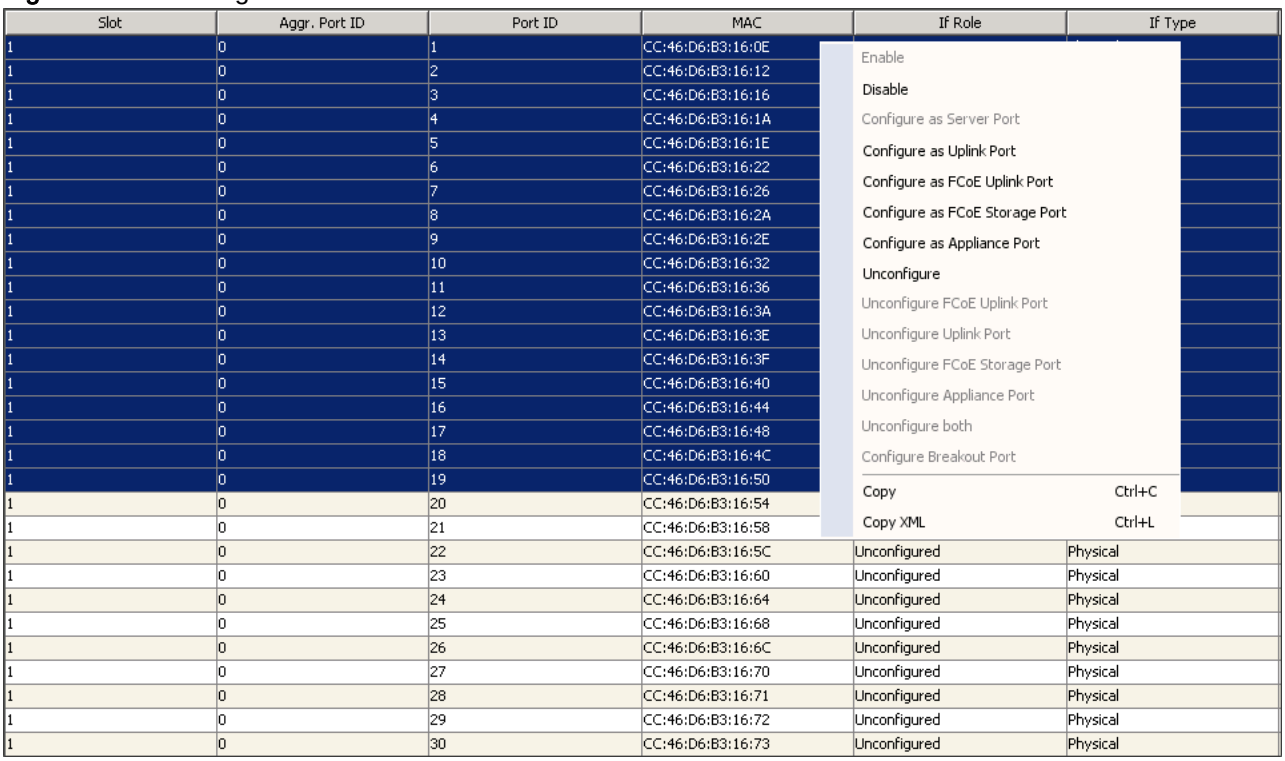
1. Select the `Equipment` tab on the top left of the window.
2. Select `Equipment > Fabric Interconnects > Fabric Interconnect A (primary) > Fixed Module`
3. Click the Ethernet Ports section.
4. Select all the ports that are connected to the Servers right-click them, and select `Configure as a Server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

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 19 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.

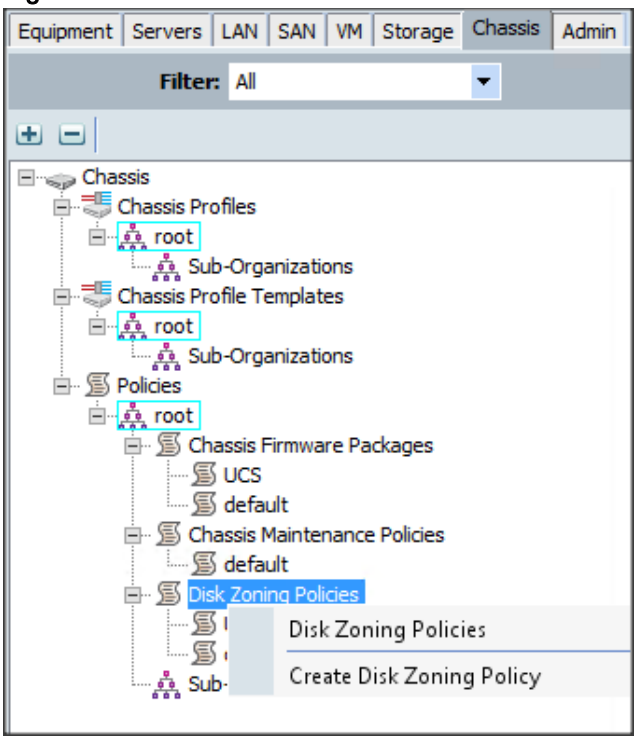
Creating Disk Zoning Policy

1. Click the Chassis tab on UCS Manager on the top left menu (Figure 20).

2. Expand Policies→Root→Disk Zoning Policies

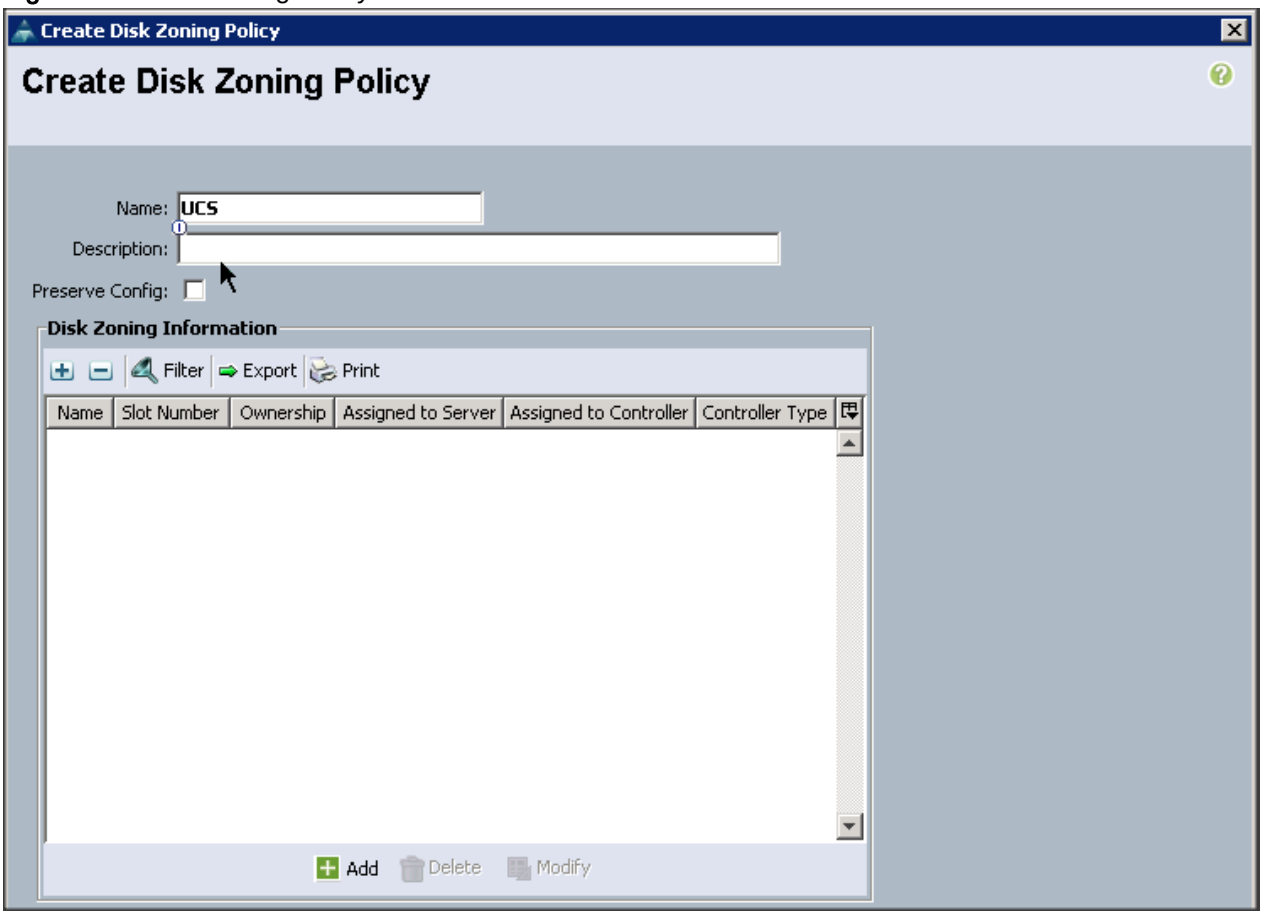
3. Right click on the Disk Zoning Policies and click Create Disk Zoning Policy.

Figure 20 Chassis Profile Screen



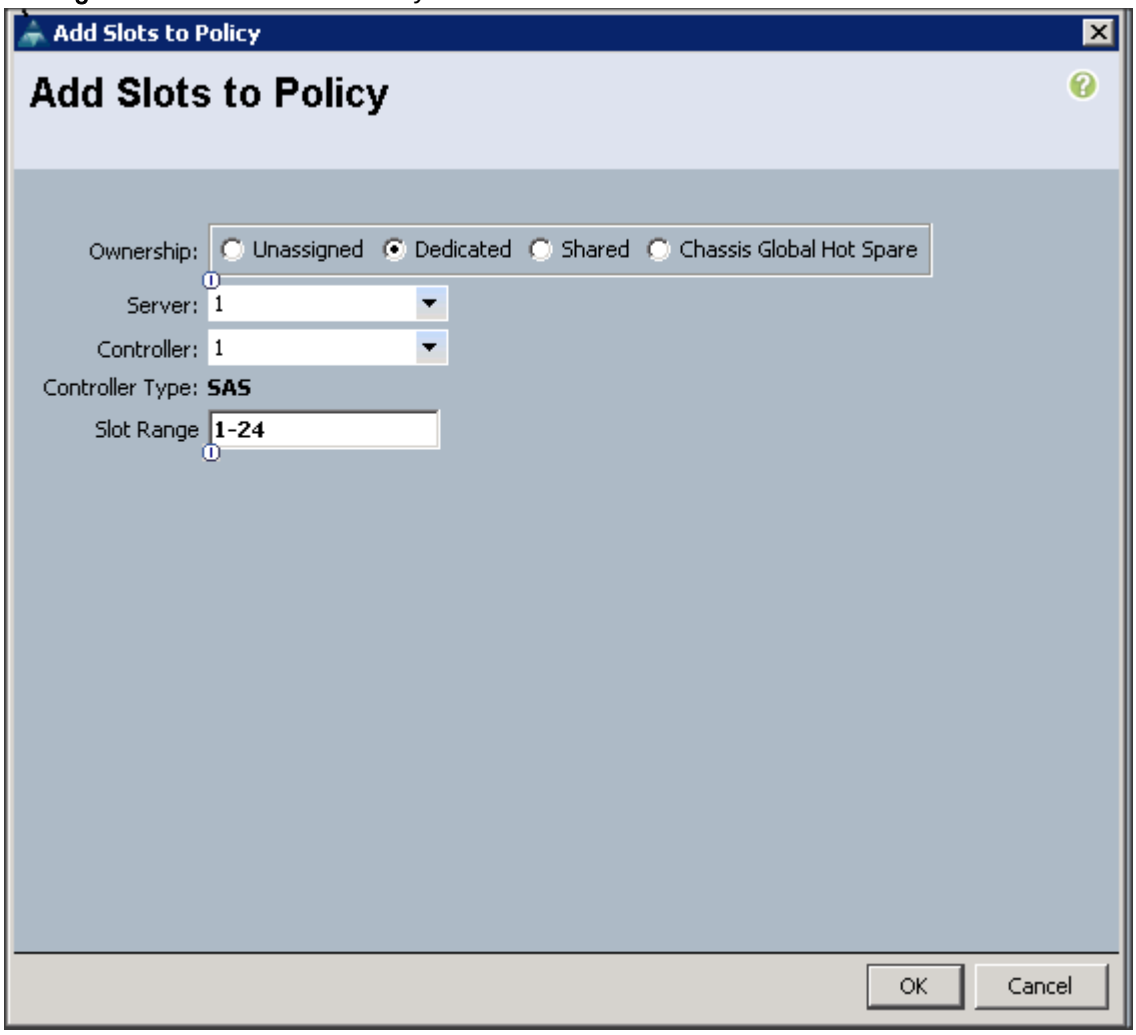
4. On Create Disk Zoning Policy windows enter the Name UCS and click “+” to create the Disk Zoning. (Figure 21)

Figure 21 Disk Zoning Policy Screen



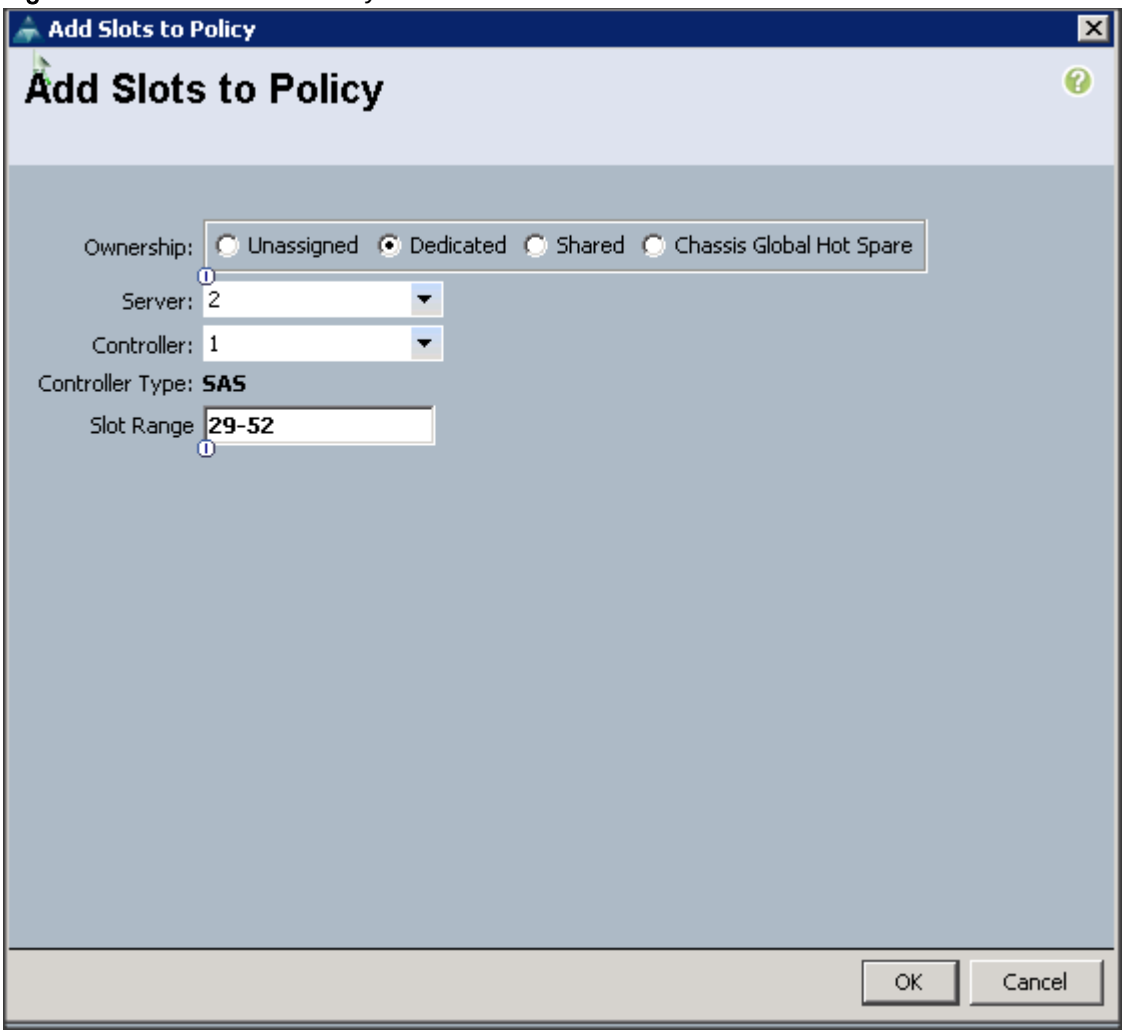
5. In the Add Slots to Policy window (Figure 22), select the “Dedicated” radio button. From the server drop down list choose “1”, from the controller drop down list choose “1”, in the slot range enter 1-24 and click “OK”.

Figure 22 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 23)

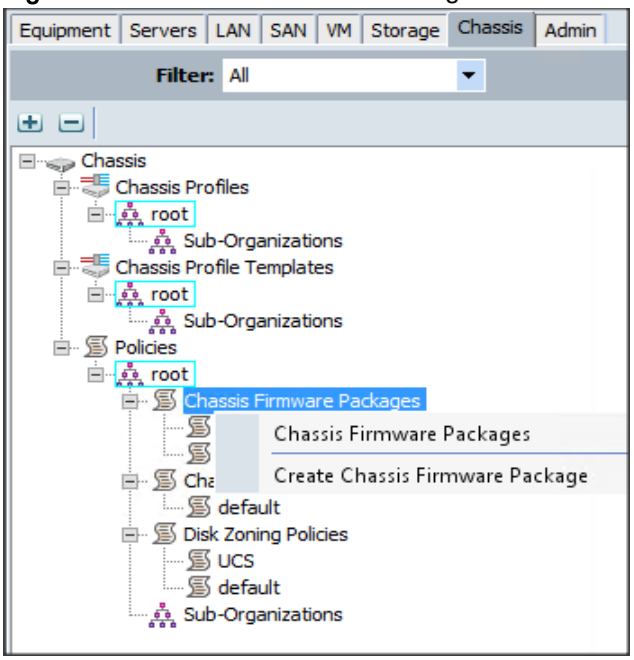
Figure 23 Add Slots to Policy Screen 2



Creating Chassis Firmware Package Policy

1. Right click on the Chassis Firmware Packages and click “Create Chassis Firmware Packages”. (Figure 24)

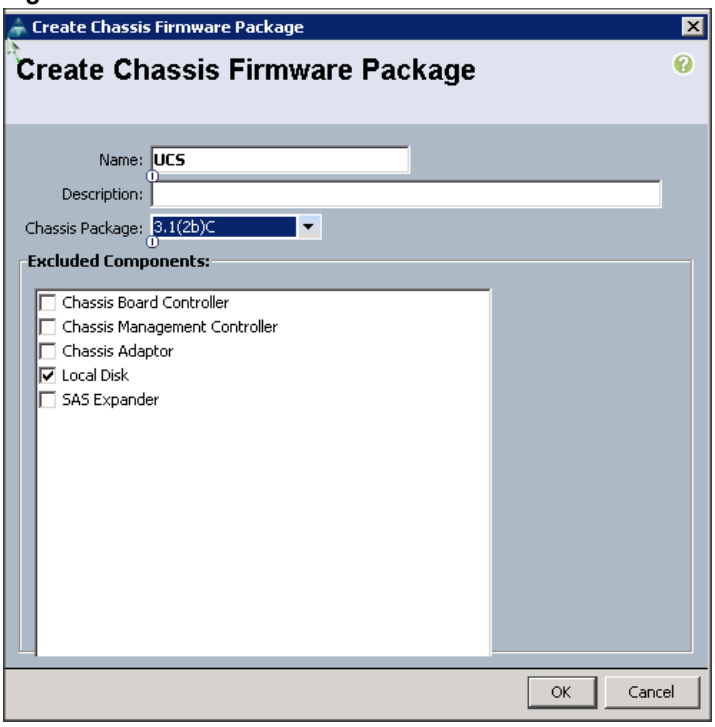
Figure 24 Chassis Firmware Packages



2. In Create Chassis Firmware Package window enter UCS as the Name. (Figure 25)

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

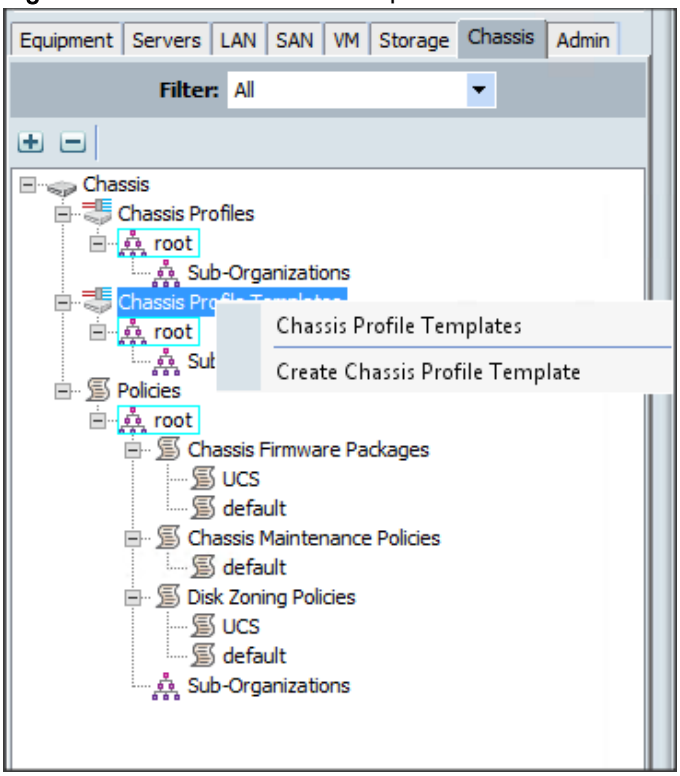
Figure 25 Create Chassis Firmware Screen



Creating Chassis Profiles from Template

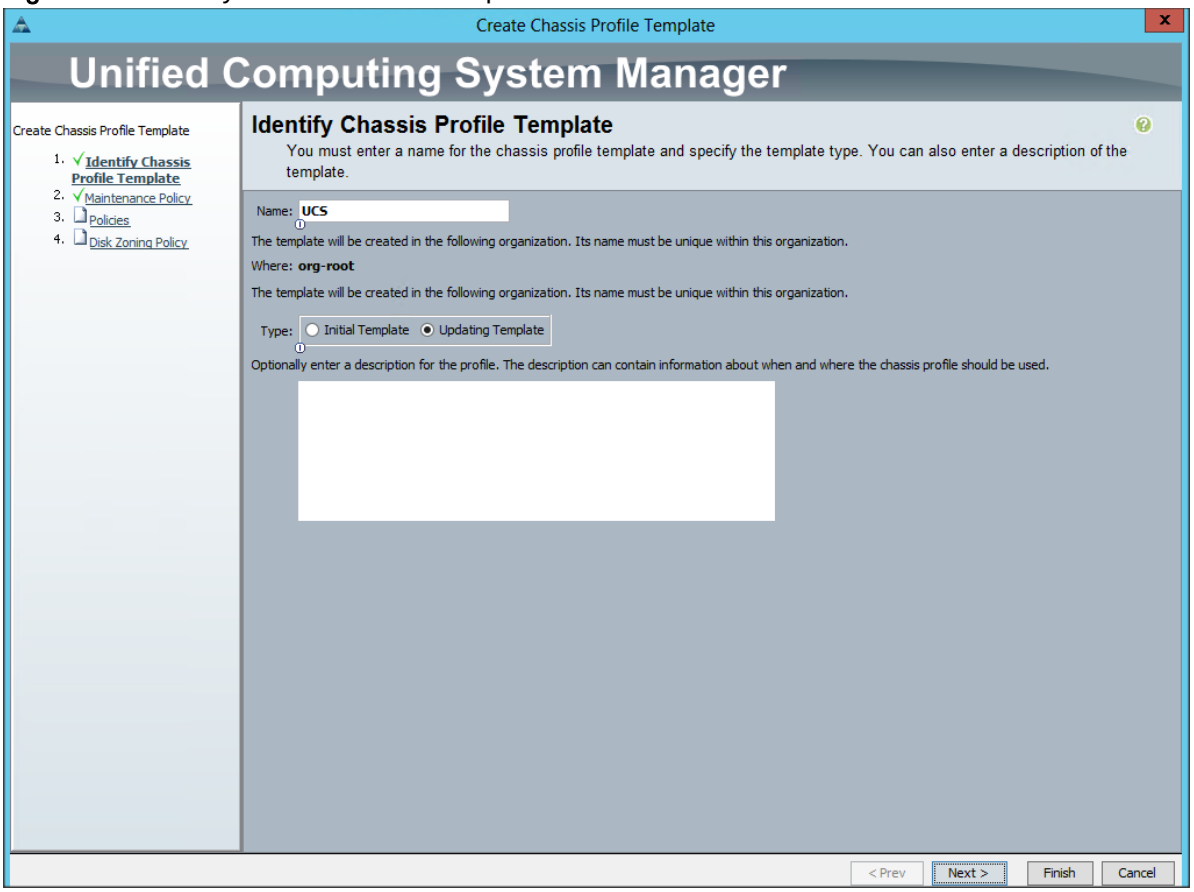
1. Under Chassis Profile Template, right click and click “Create Chassis Profile Templates” (Figure 26).

Figure 26 Chassis Profile Templates



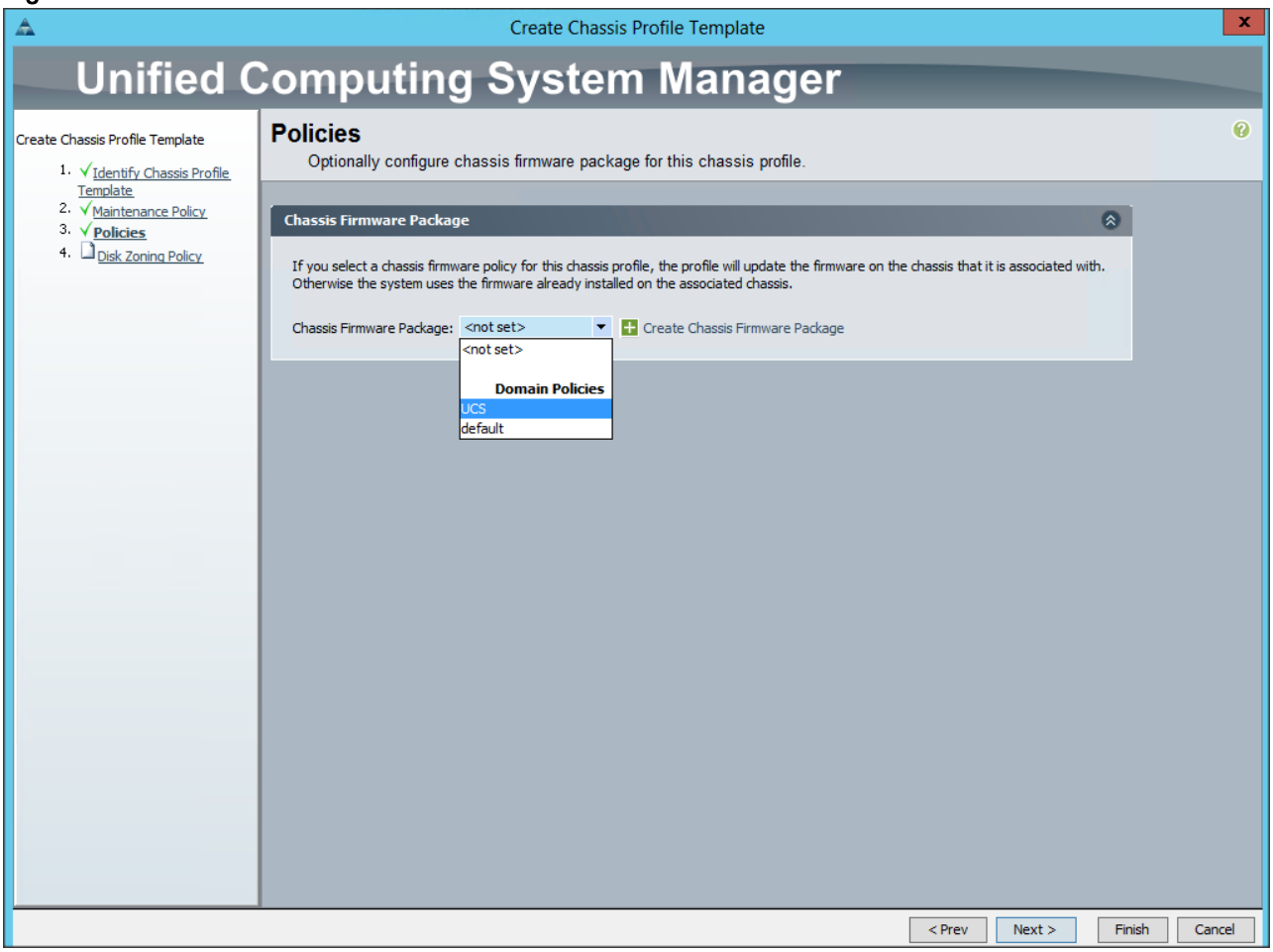
2. Enter the Name “UCS” and select “Updating Template” as the type, and click Next and Next again. (Figure 27)

Figure 27 Identify Chassis Profile Template



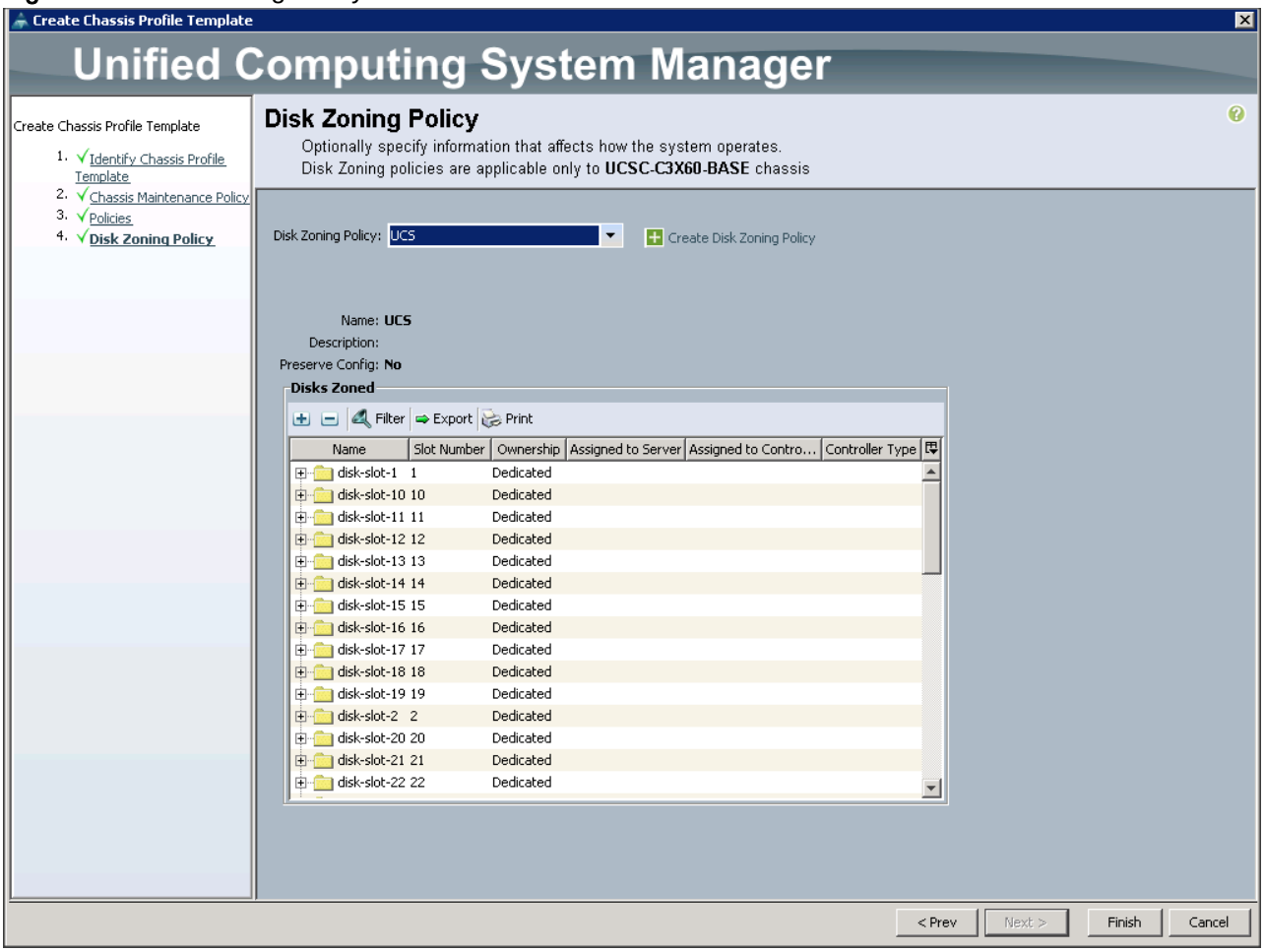
3. From the Chassis Firmware Package drop down list choose UCS and click Next. (Figure 28)

Figure 28 UCS Policies



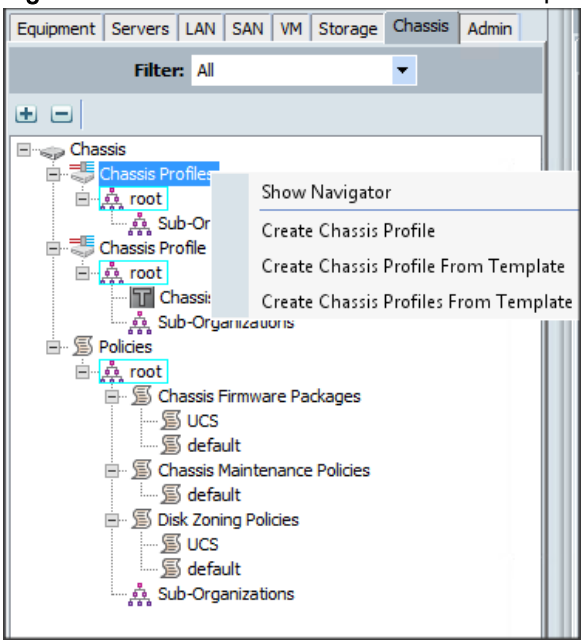
4. From the Disk Zoning Policy drop down list choose UCS and click Finish. (Figure 29)

Figure 29 Disk Zoning Policy



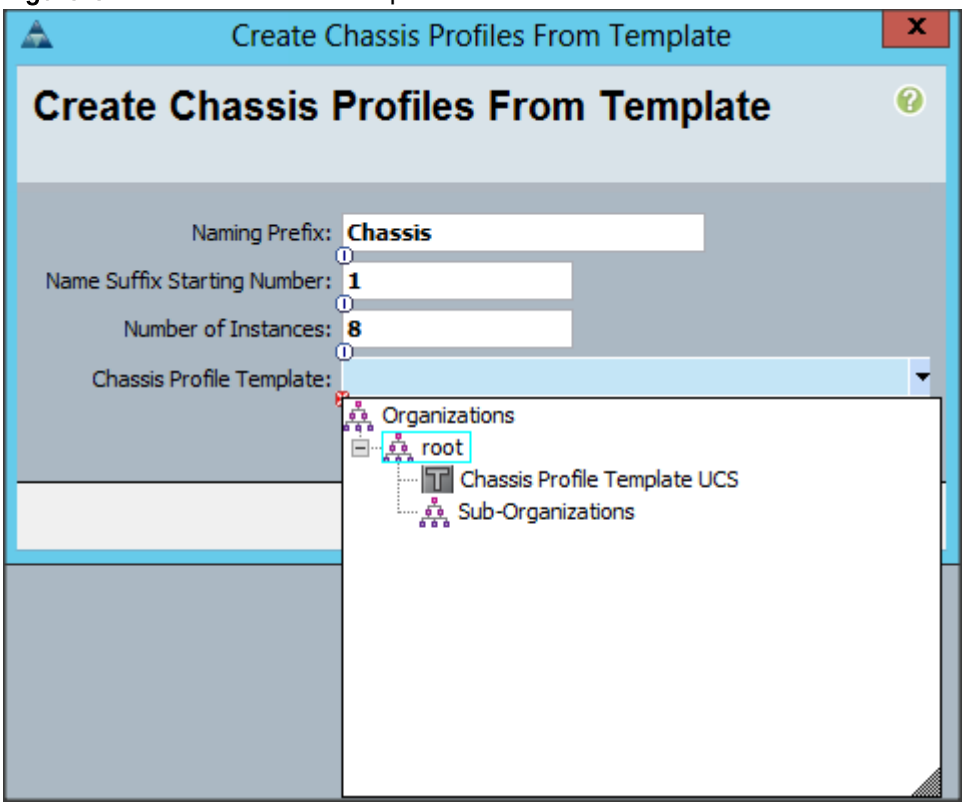
5. Right click on the Chassis Profiles and click “Create Chassis Profile from Templates” (Figure 30).

Figure 30 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 31)

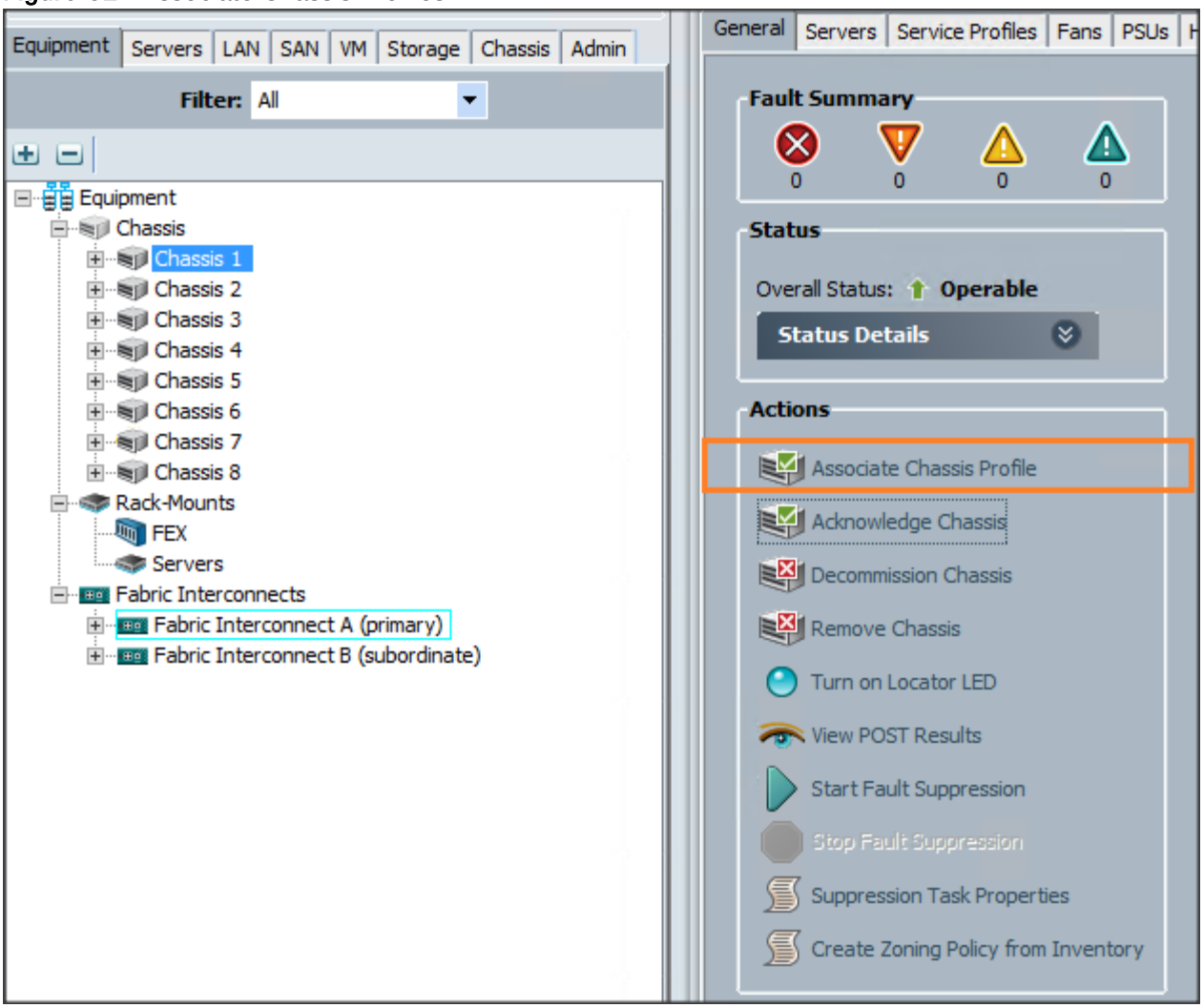
Figure 31 Chassis Profile Template UCS



Associating Chassis Profiles to Individual Chassis

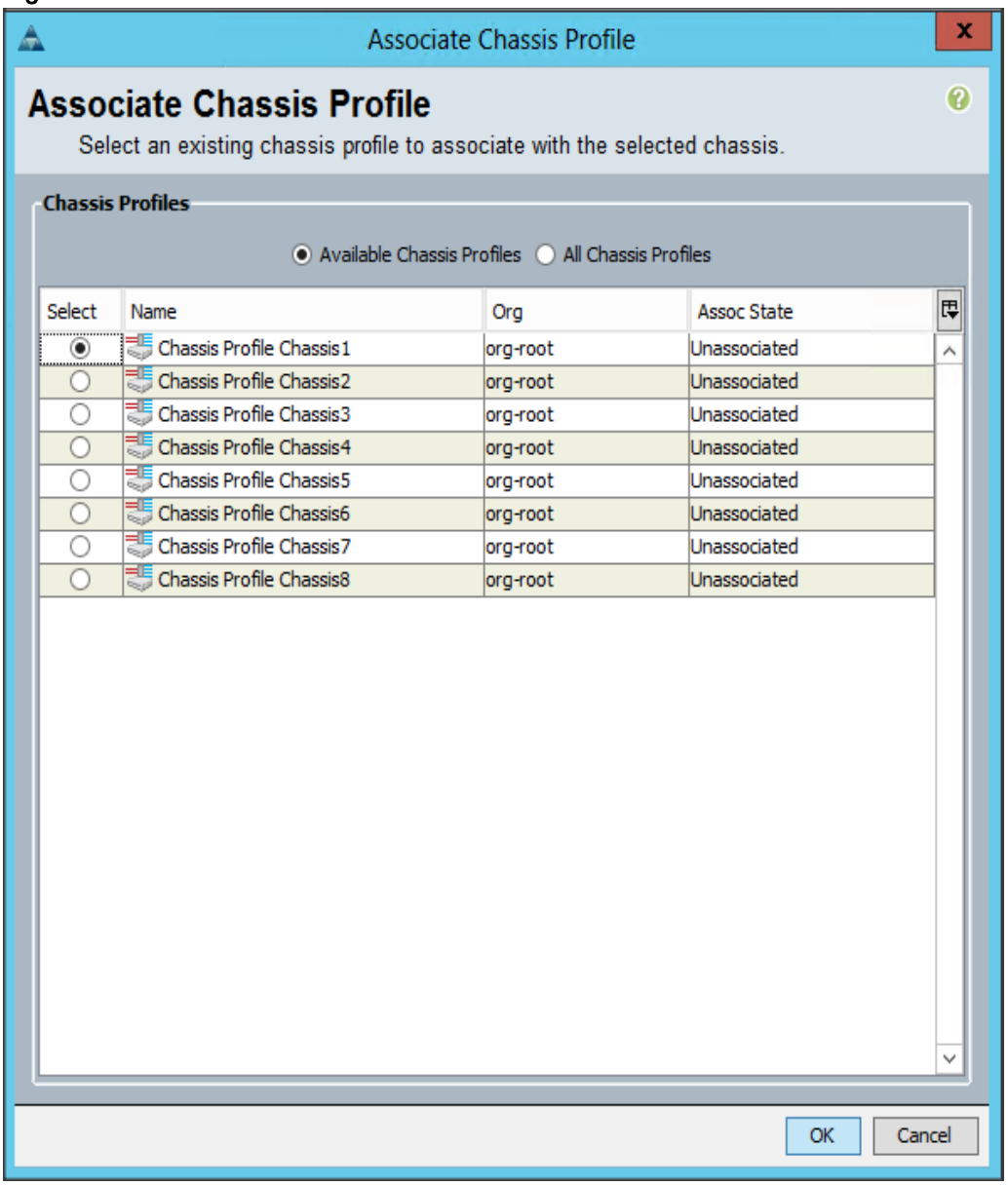
1. On the Cisco UCS Manager UI select the Equipment tab. Under Equipment expand Chassis.
2. Select the Chassis and click Associate Chassis Profile. (Figure 32)

Figure 32 Associate Chassis Profiles



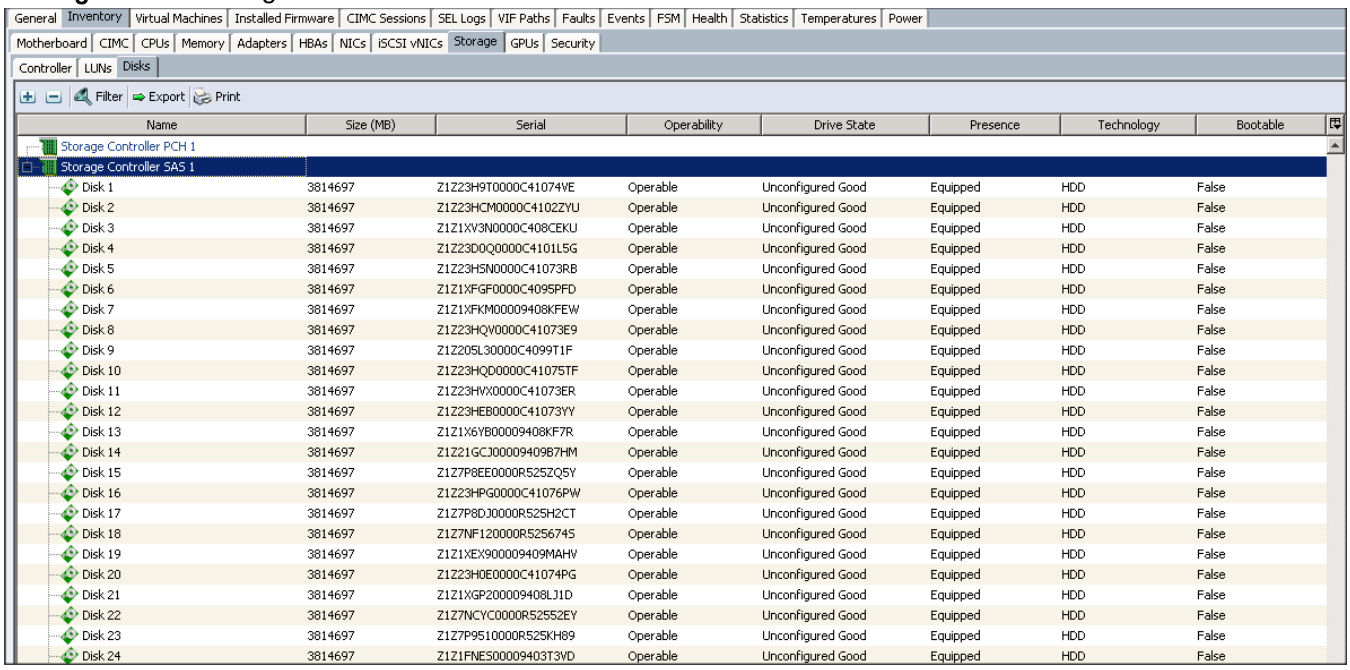
3. Select “Chassis Profile Chassis 1” and click “OK”. (Figure 33)

Figure 33 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 34)

Figure 34 Storage Controller SAS 1

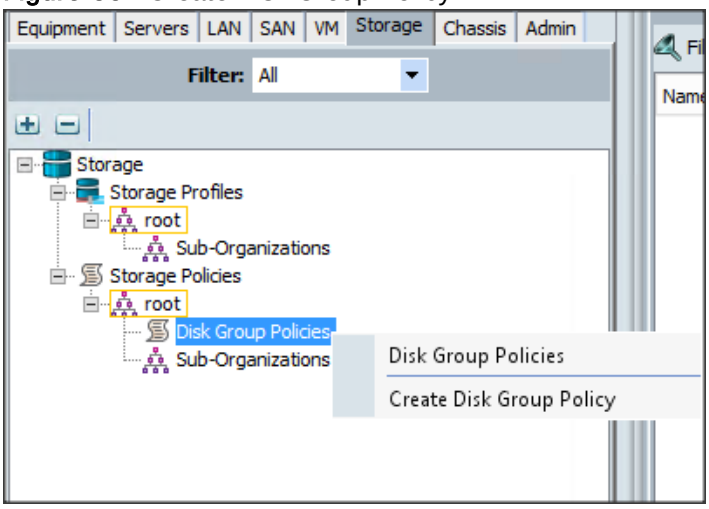


Name	Size (MB)	Serial	Operability	Drive State	Presence	Technology	Bootable
Storage Controller PCH 1							
Storage Controller SAS 1							
Disk 1	3814697	Z1Z23H9T0000C41074VE	Operable	Unconfigured Good	Equipped	HDD	False
Disk 2	3814697	Z1Z23HCM0000C4102ZYU	Operable	Unconfigured Good	Equipped	HDD	False
Disk 3	3814697	Z1Z1XV3N0000C408CEKU	Operable	Unconfigured Good	Equipped	HDD	False
Disk 4	3814697	Z1Z23D0Q0000C4101L5G	Operable	Unconfigured Good	Equipped	HDD	False
Disk 5	3814697	Z1Z23H5N0000C41073RB	Operable	Unconfigured Good	Equipped	HDD	False
Disk 6	3814697	Z1Z1XFGF0000C4095PFD	Operable	Unconfigured Good	Equipped	HDD	False
Disk 7	3814697	Z1Z1XFKM00009408KFEW	Operable	Unconfigured Good	Equipped	HDD	False
Disk 8	3814697	Z1Z23HQV0000C41073E9	Operable	Unconfigured Good	Equipped	HDD	False
Disk 9	3814697	Z1Z205L30000C4099T1F	Operable	Unconfigured Good	Equipped	HDD	False
Disk 10	3814697	Z1Z23HQD0000C41075TF	Operable	Unconfigured Good	Equipped	HDD	False
Disk 11	3814697	Z1Z23HVX0000C41073ER	Operable	Unconfigured Good	Equipped	HDD	False
Disk 12	3814697	Z1Z23HEB0000C41073YY	Operable	Unconfigured Good	Equipped	HDD	False
Disk 13	3814697	Z1Z1X6YB00009408KF7R	Operable	Unconfigured Good	Equipped	HDD	False
Disk 14	3814697	Z1Z21GCJ00009409B7HM	Operable	Unconfigured Good	Equipped	HDD	False
Disk 15	3814697	Z1Z7P8EE0000R525ZQ5Y	Operable	Unconfigured Good	Equipped	HDD	False
Disk 16	3814697	Z1Z23HPG0000C41076PW	Operable	Unconfigured Good	Equipped	HDD	False
Disk 17	3814697	Z1Z7P8DJ0000R525H2CT	Operable	Unconfigured Good	Equipped	HDD	False
Disk 18	3814697	Z1Z7NF120000R5256745	Operable	Unconfigured Good	Equipped	HDD	False
Disk 19	3814697	Z1Z1XEX900009409MAHV	Operable	Unconfigured Good	Equipped	HDD	False
Disk 20	3814697	Z1Z23H0E0000C41074PG	Operable	Unconfigured Good	Equipped	HDD	False
Disk 21	3814697	Z1Z1XGP200009408LJ1D	Operable	Unconfigured Good	Equipped	HDD	False
Disk 22	3814697	Z1Z7NCYC0000R52552EY	Operable	Unconfigured Good	Equipped	HDD	False
Disk 23	3814697	Z1Z7P9510000R525KH89	Operable	Unconfigured Good	Equipped	HDD	False
Disk 24	3814697	Z1Z1FNES00009403T3VD	Operable	Unconfigured Good	Equipped	HDD	False

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 35)

Figure 35 Create Disk Group Policy



2. In the Create Disk Policy window, configure the following parameters and click OK. (Figure 36)
   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

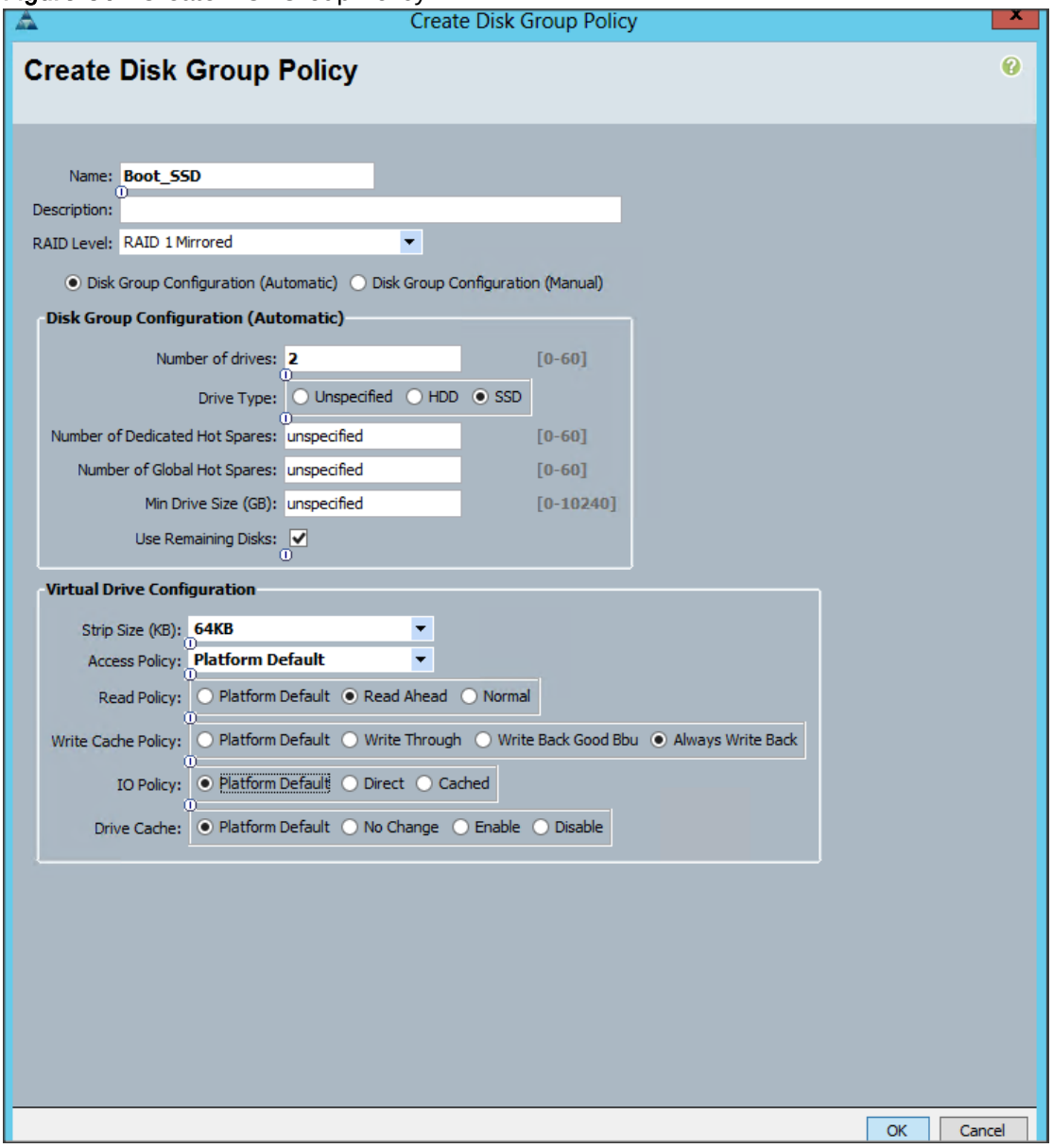
h. Access Policy = Platform Default

i. Read Policy = Read Ahead

j. Write Cache Policy = Always Write Back

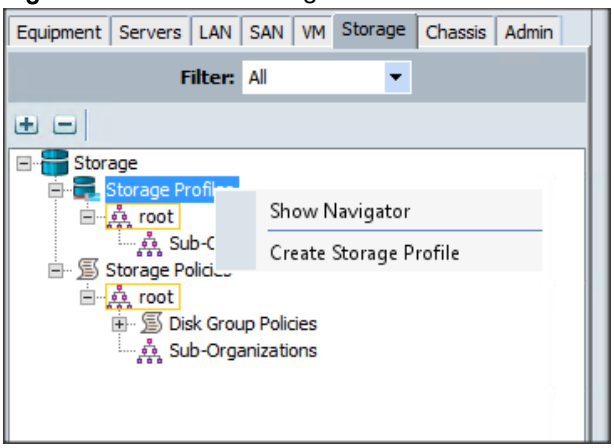
k. IO Policy and Drive Cache = Platform Default

Figure 36 Create Disk Group Policy



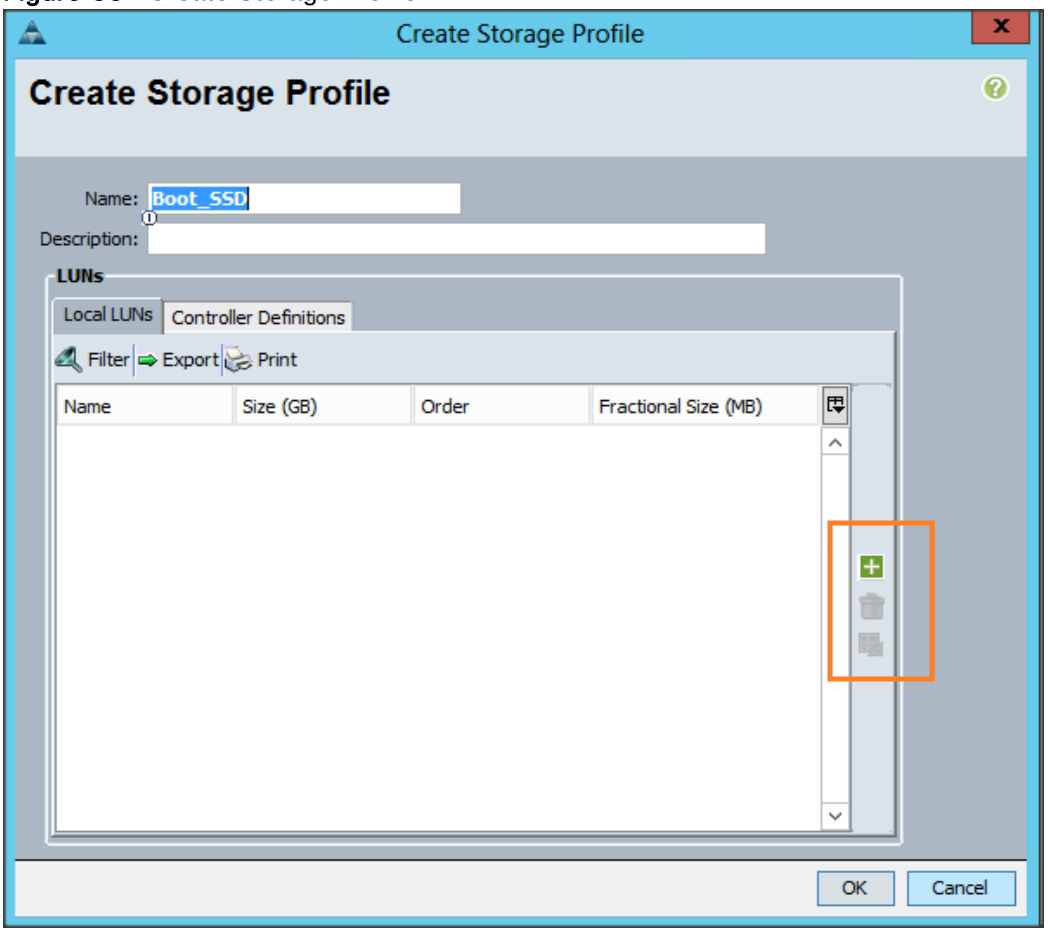
3. Click on the Storage tab. Right click on Storage Profile and click Create Storage Profile. (Figure 37)

Figure 37 Create Storage Profile



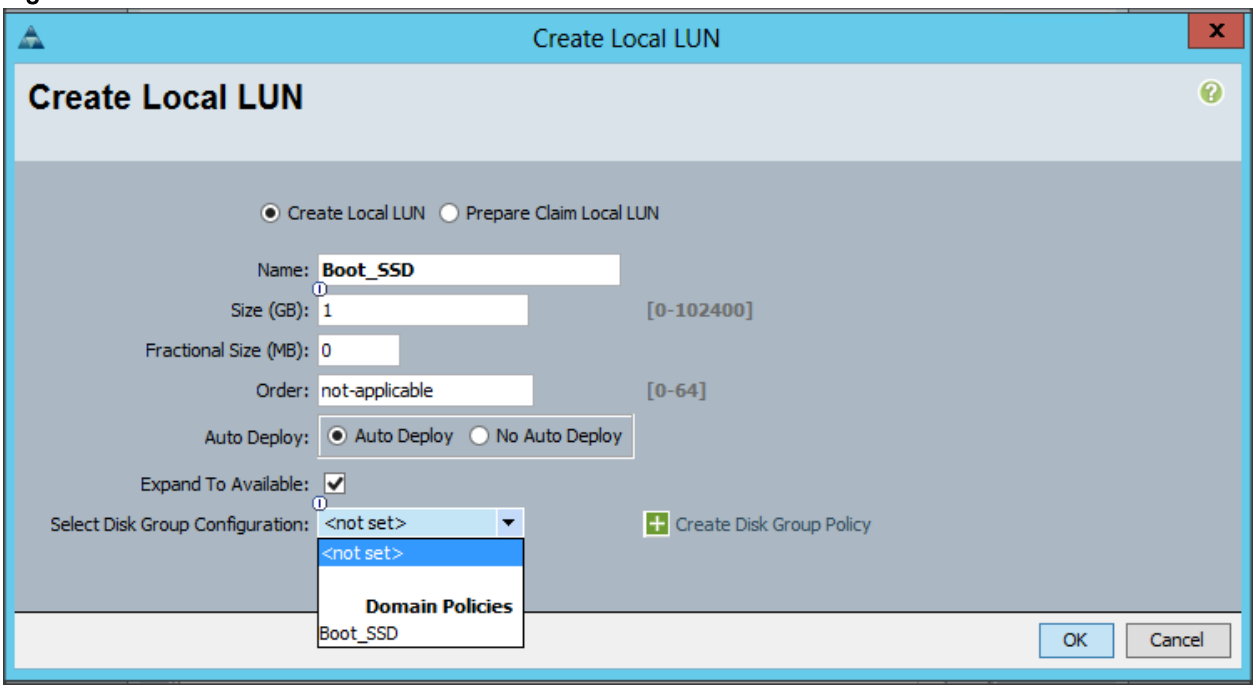
4. Enter “Boot_SSD” in the name field. Under Local LUNs click “+” to add local lun. (Figure 38)

Figure 38 Create Storage Profile



5. In the Create Local LUN window, enter the name “Boot_SSD”. (Figure 39)
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 39 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 `ucs` for the `name` of the MAC pool. (Figure 40)
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 41)
10. Specify a size of the MAC address pool, which is sufficient to support the available server resources.

11. Click `OK`.

Figure 40 Define Name and Description of MAC Pool

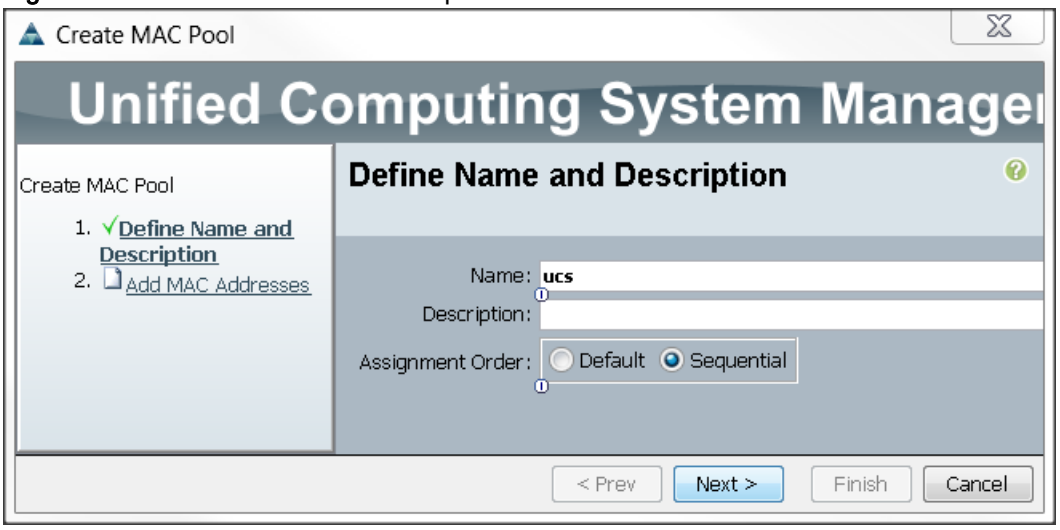
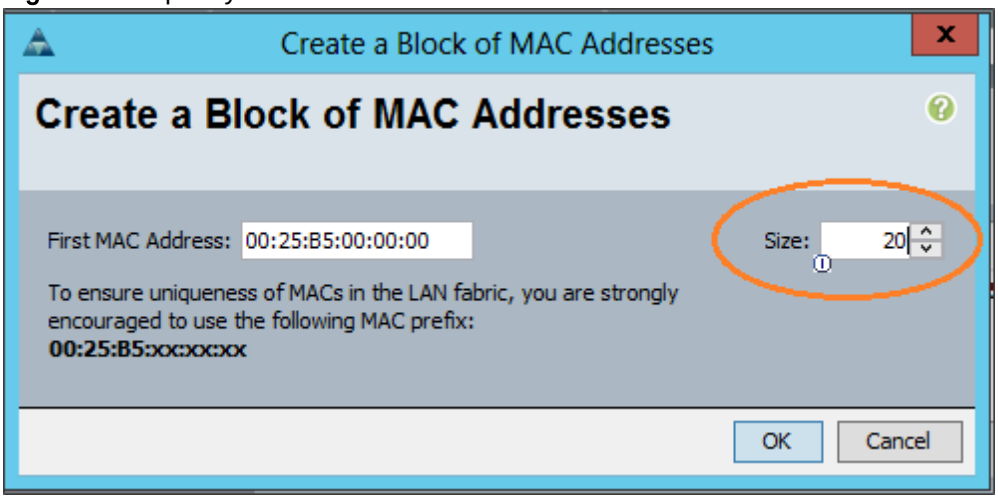
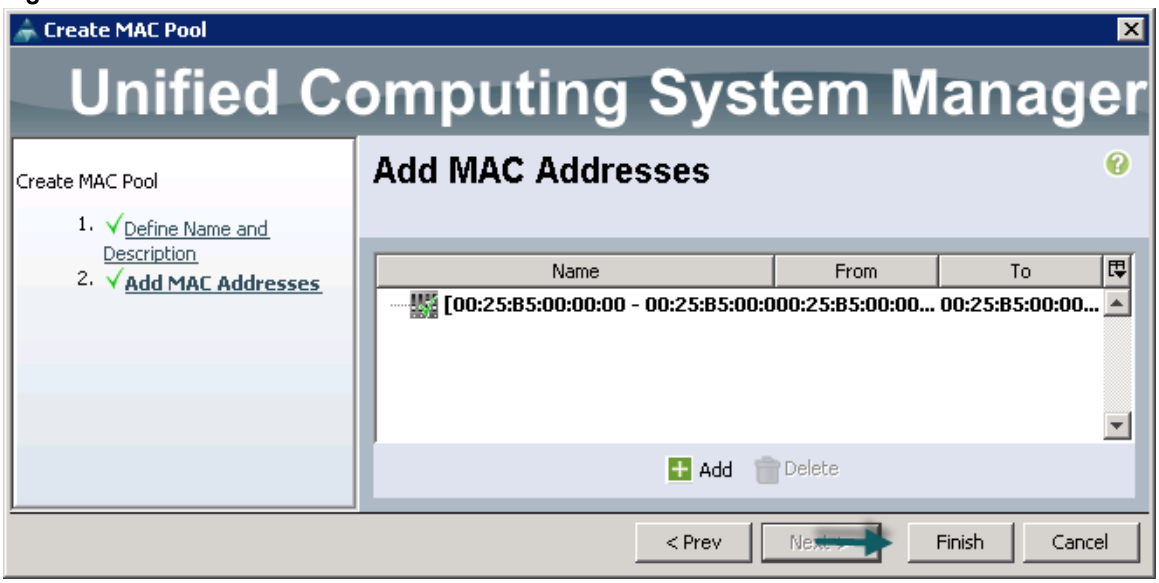


Figure 41 Specify first MAC Address and Size



12. Click `Finish`. (Figure 42)

Figure 42 Add MAC Addresses



13. When the message box displays, click `OK`.

Creating a Server Pool

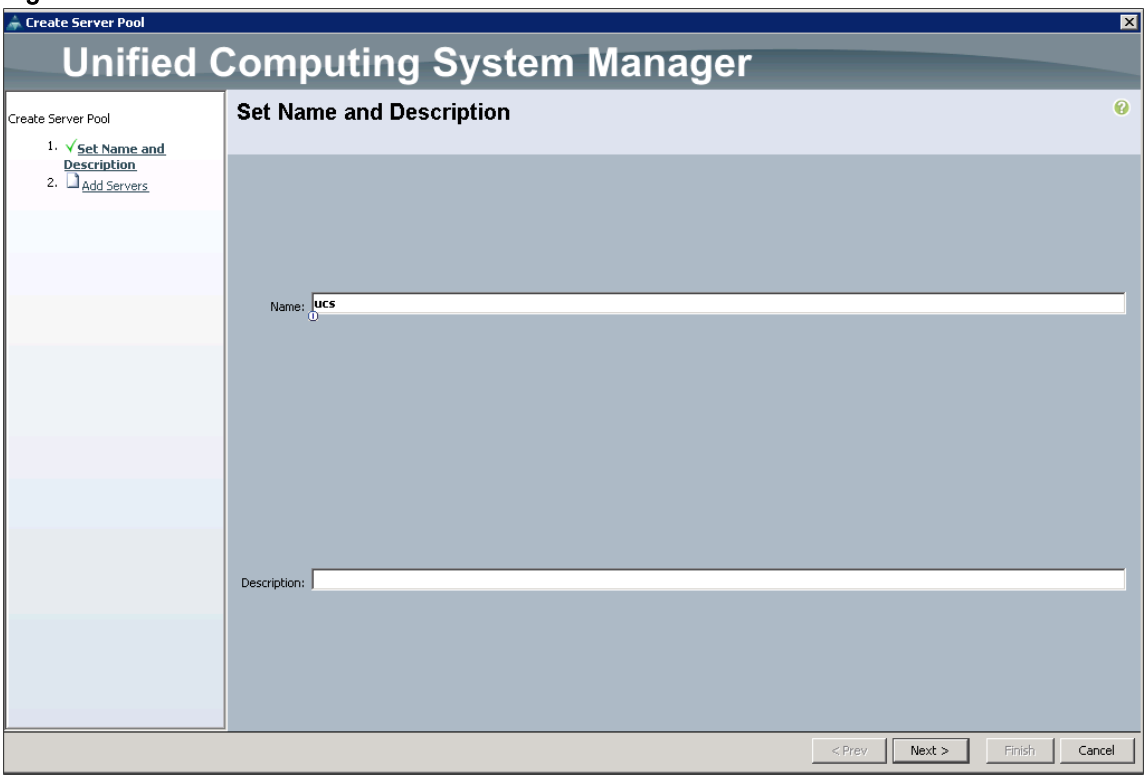
A server pool contains a set of servers. These servers typically share the same characteristics. Those characteristics can be their location in the chassis, or an attribute such as server type, amount of memory, local storage, type of CPU, or local drive configuration. You can manually assign a server 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 43)
6. (Optional) enter a description for the organization.

7. Click `Next >` to add the servers.

Figure 43 Name the Server Pool

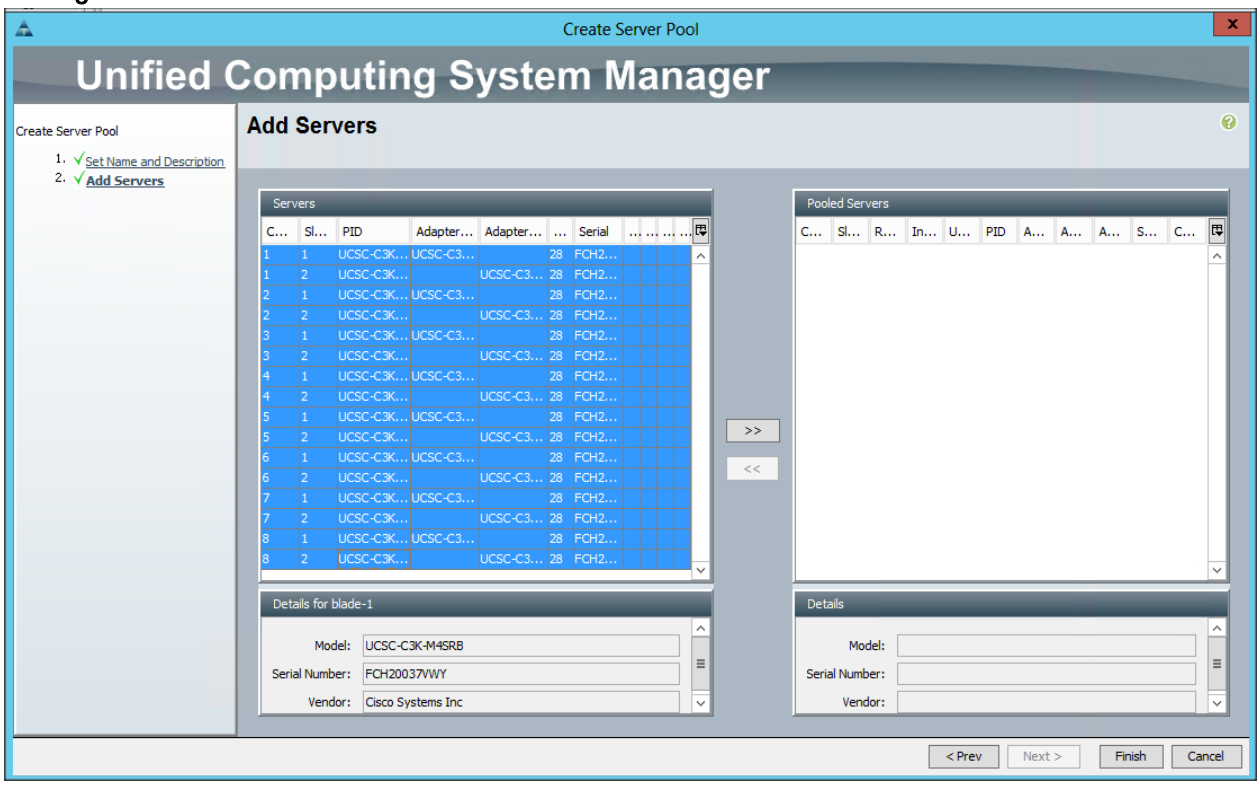


8. Select all the Cisco UCS S3260 Storage Server to be added to the server pool that was previously created (ucs), then `Click >>` to add them to the pool. (Figure 44)

9. Click `Finish`.

10. Click `OK` and then click `Finish`.

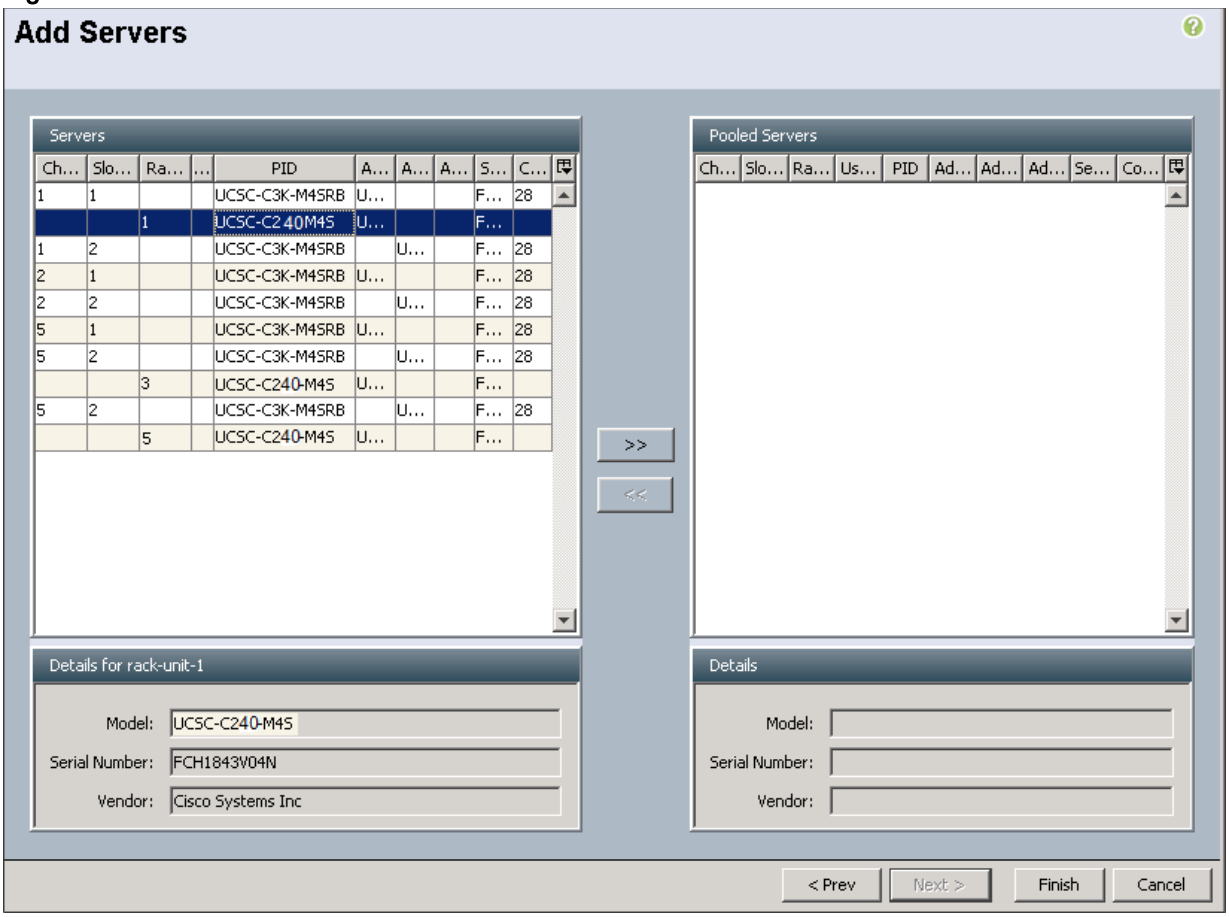
Figure 44 Add Server



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 45)

Figure 45 Add Servers Window



13. Click `Finish`.
14. Click `OK` and then click `Finish`

Creating Policies for Service Profile Templates

Creating Host Firmware Package Policy

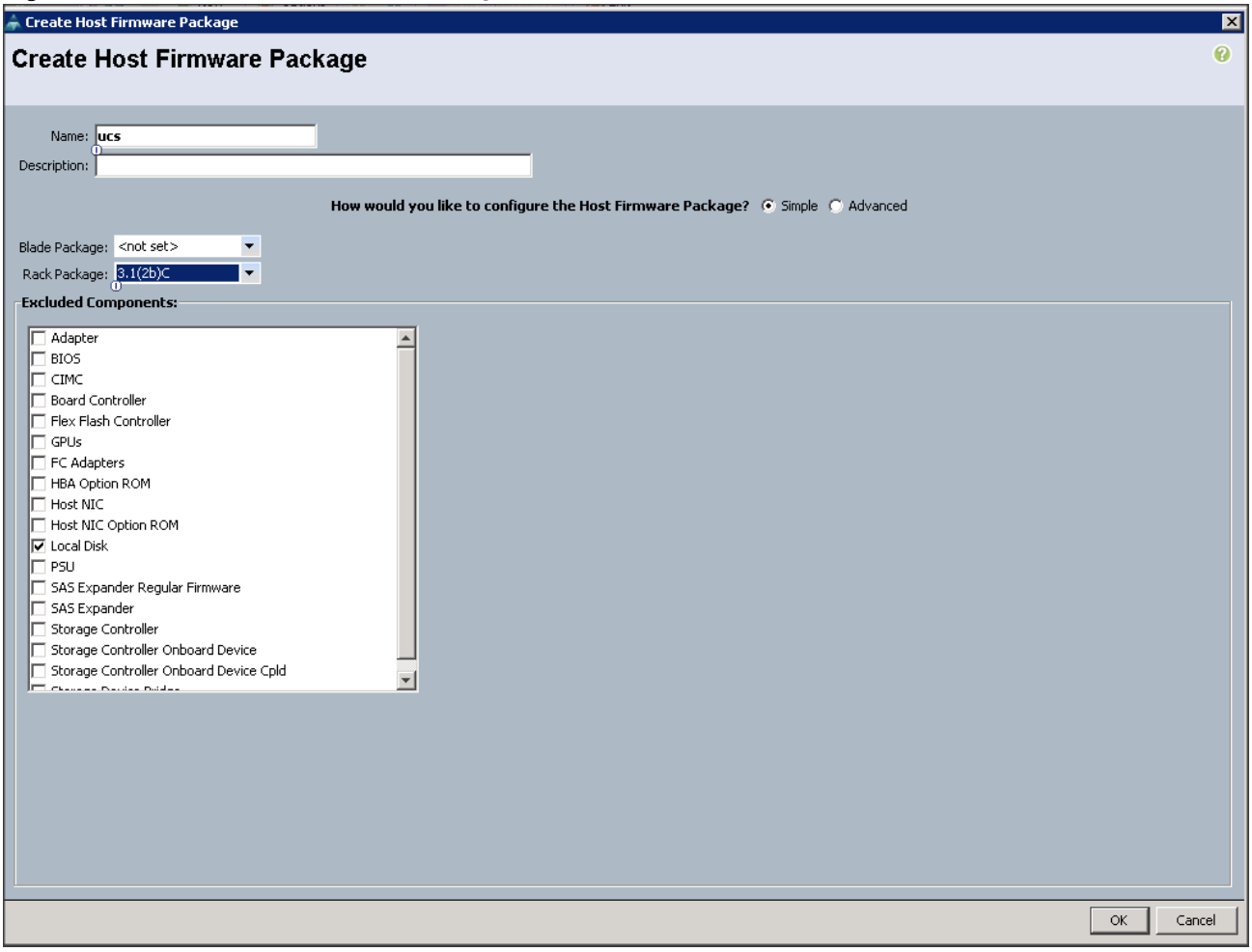
Firmware management policies allow the administrator to select the corresponding packages for a given server configuration. These include adapters, BIOS, board controllers, FC adapters, HBA options, and storage controller properties as applicable.

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 46)
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 46 Create Host Firmware Package Screen



Creating QoS Policies

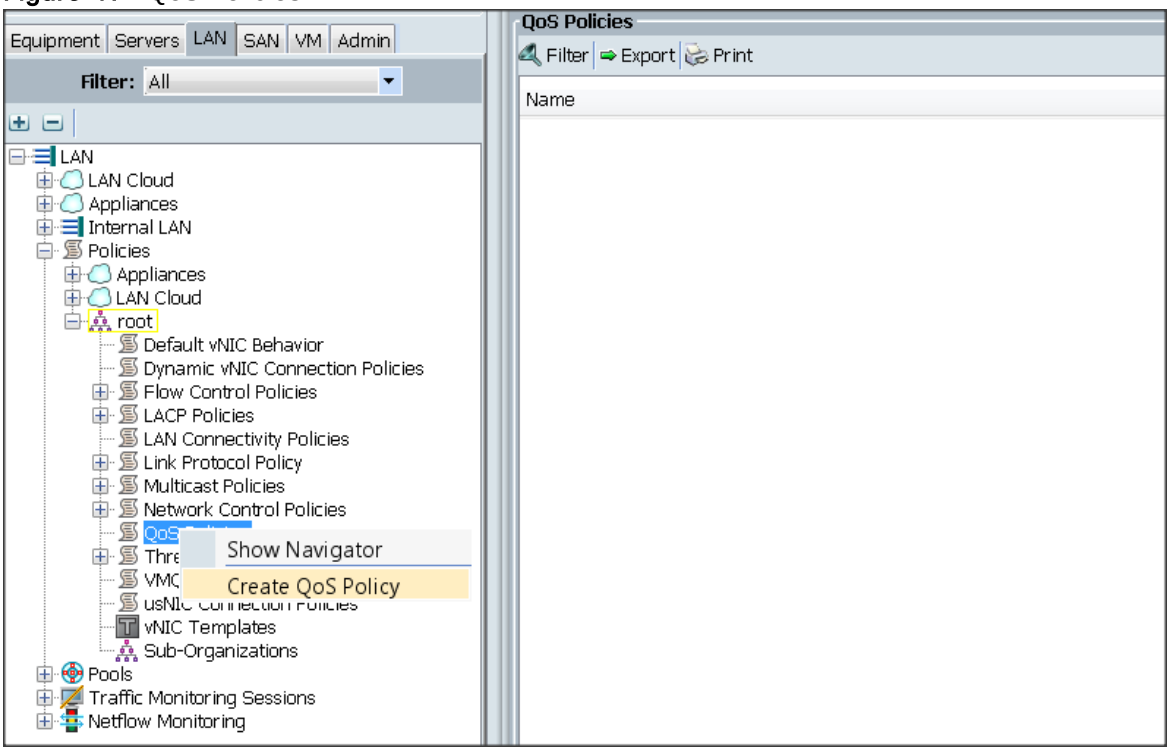
To create the QoS policy for a given server configuration using the Cisco UCS Manager GUI, complete the following steps:

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 47)

Figure 47 QoS Policies



5. Enter `Platinum` as the name of the policy. (Figure 48)
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.

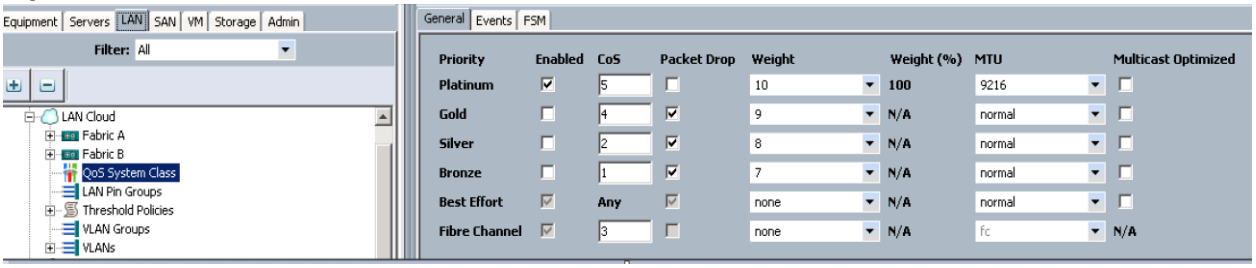
Figure 48 Create QoS Policy

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 49)
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 49 LAN General

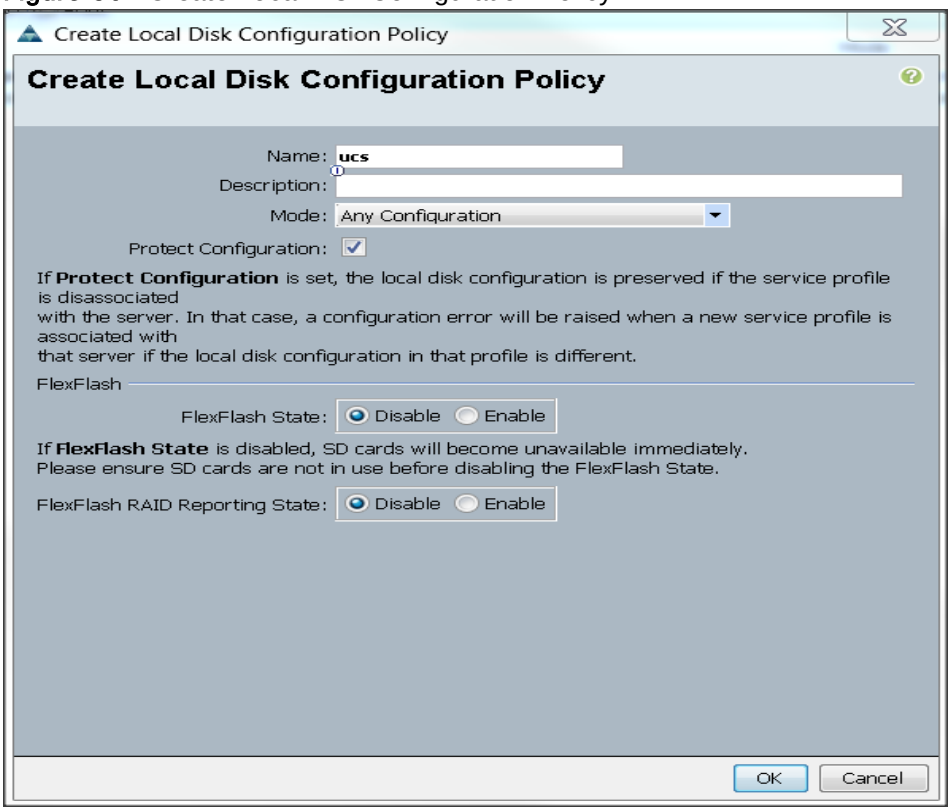


Creating the Local Disk Configuration Policy

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 50)
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 50 Create Local Disk Configuration Policy



Creating a Server BIOS Policy

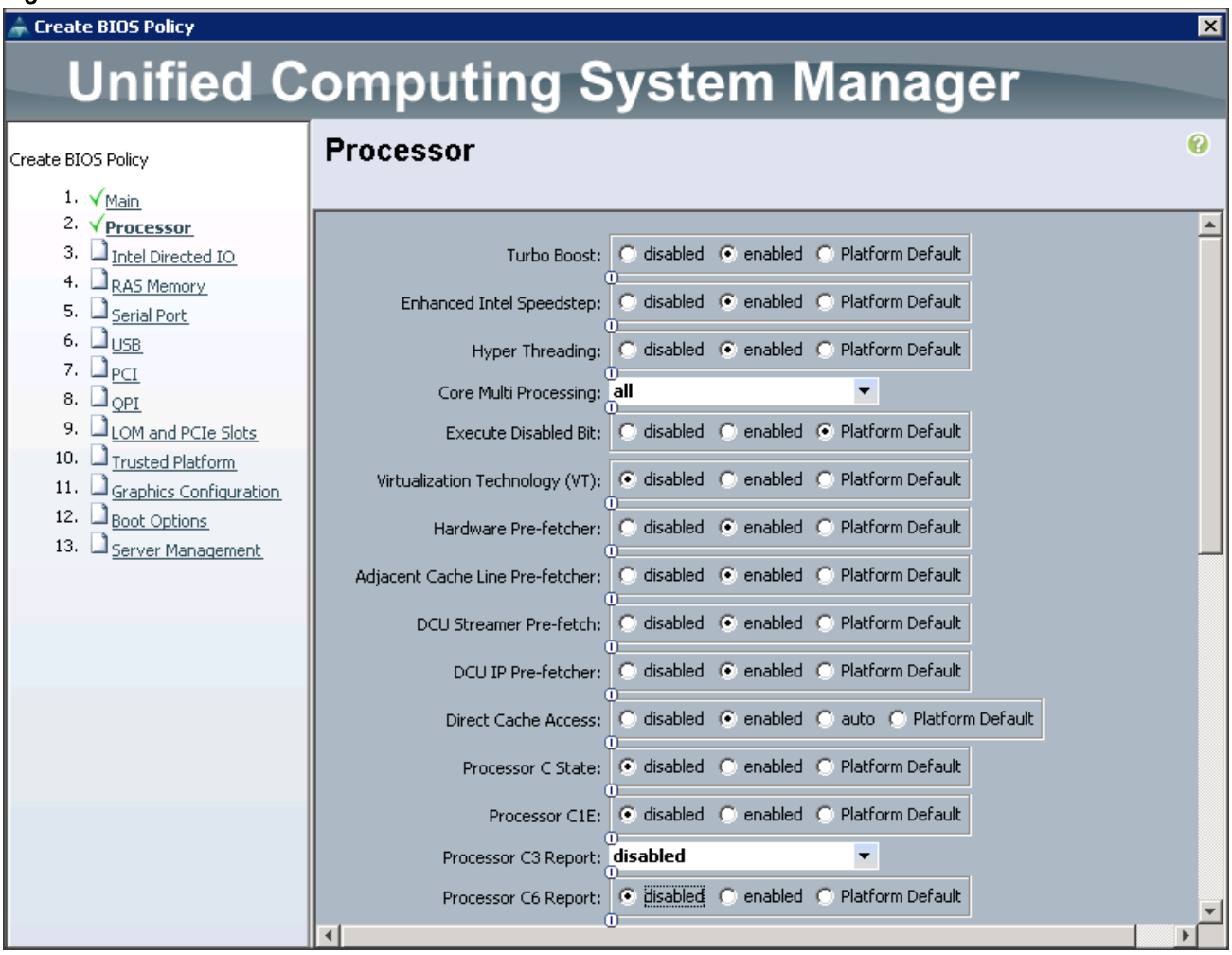
The BIOS policy feature in Cisco UCS automates the BIOS configuration process. The traditional method of setting the BIOS is manually, and is often error-prone. By creating a BIOS policy and assigning the policy to a server or group of servers, can enable transparency within the BIOS settings configuration.

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` (`ucs`).
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 51) and RAS Memory settings (Figure 52).

Figure 51 Processor



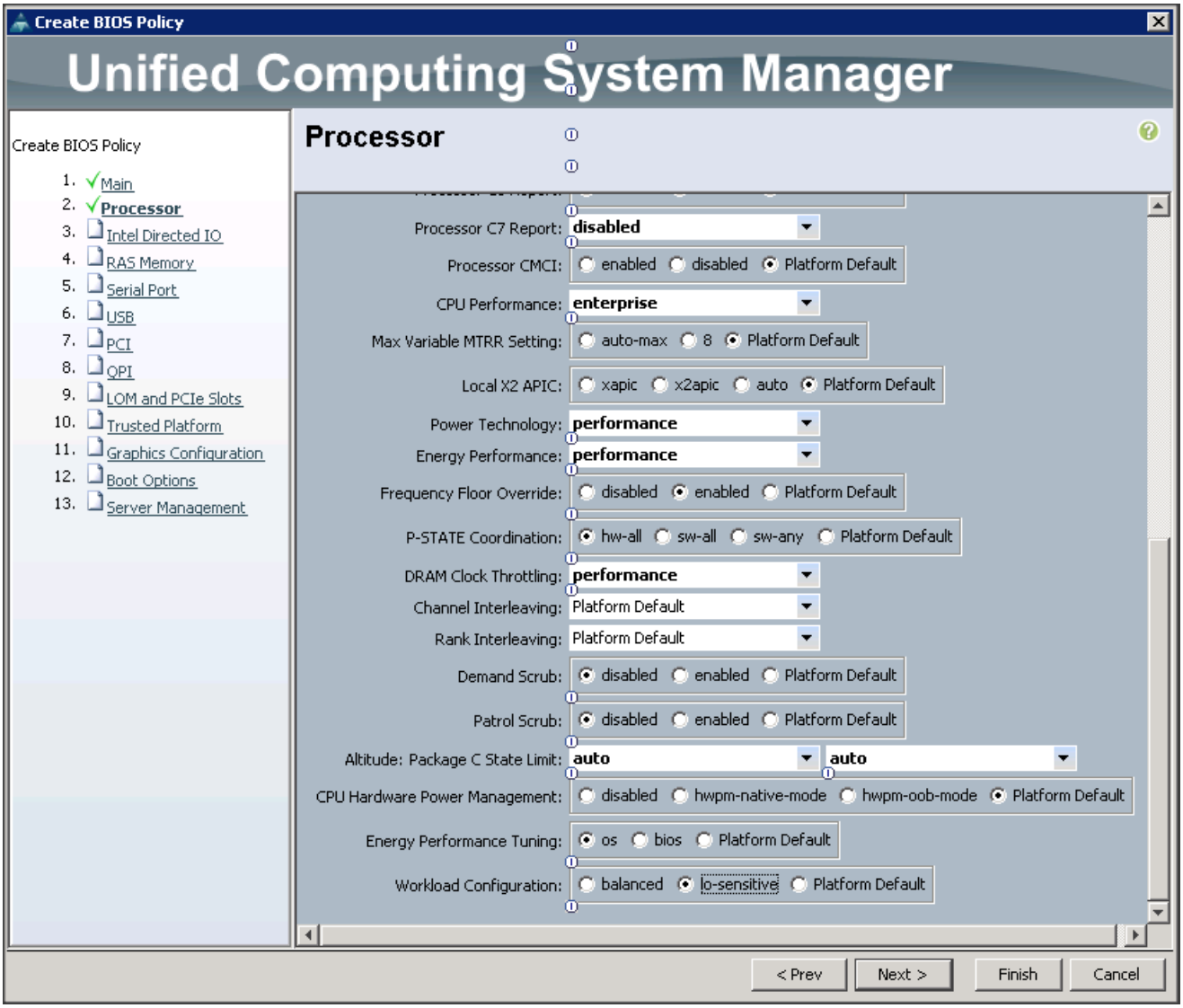
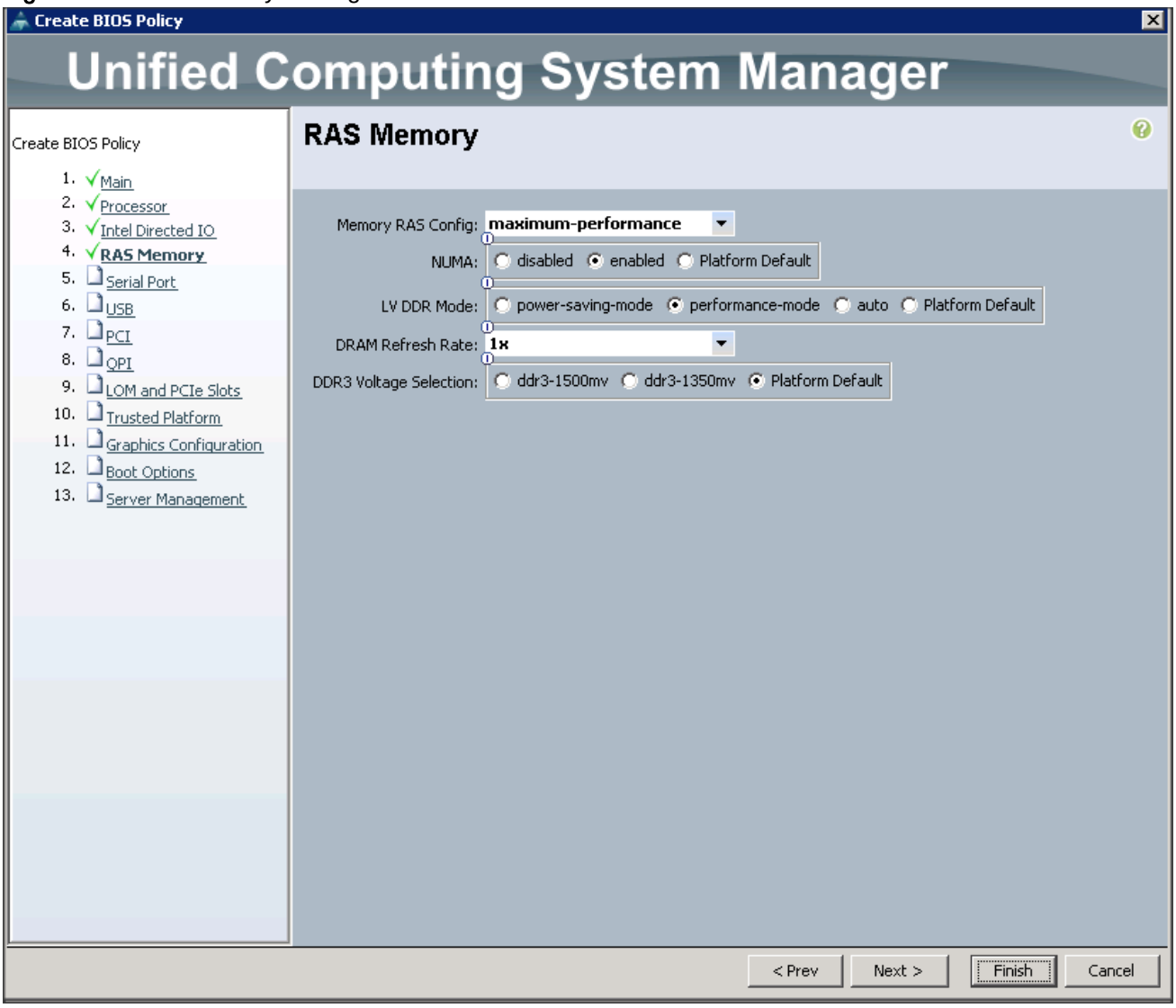

Create BIOS Policy
Unified Computing System Manager
Create BIOS Policy
1. Main
2. Processor
3. Intel Directed IO
4. RAS Memory
5. Serial Port
6. USB
7. PCI
8. QPI
9. LOM and PCIe Slots
10. Trusted Platform
11. Graphics Configuration
12. Boot Options
13. Server Management
Processor
Processor C7 Report: 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 52 RAS Memory Settings

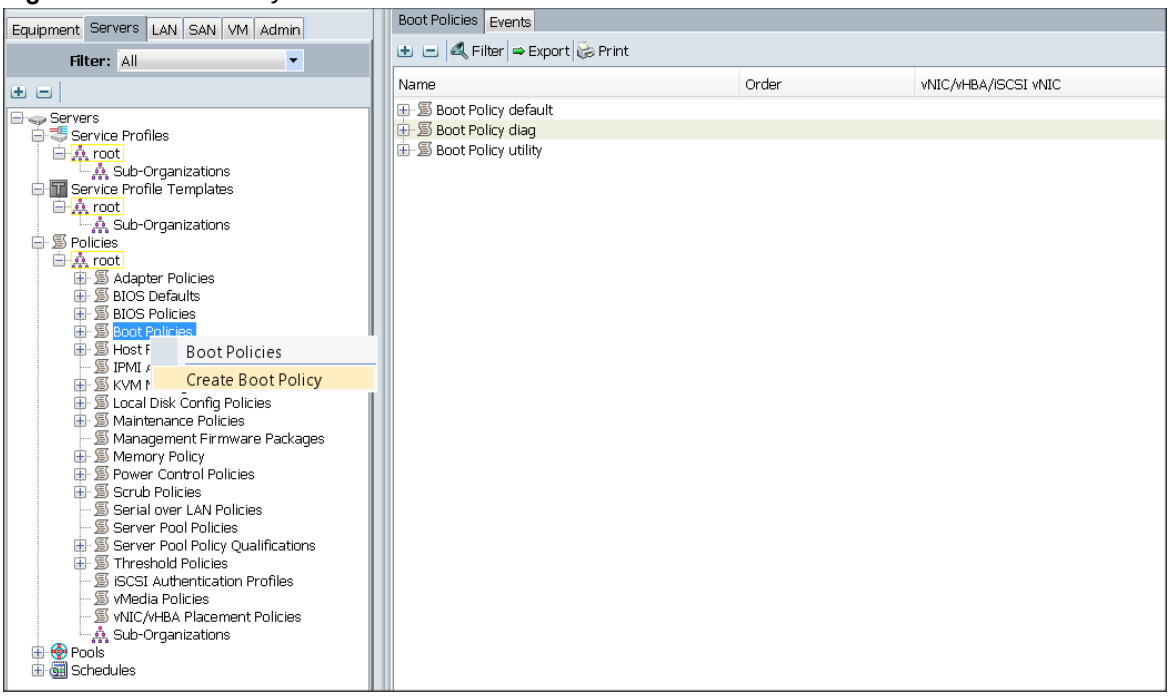


Creating the Boot Policy

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 53)

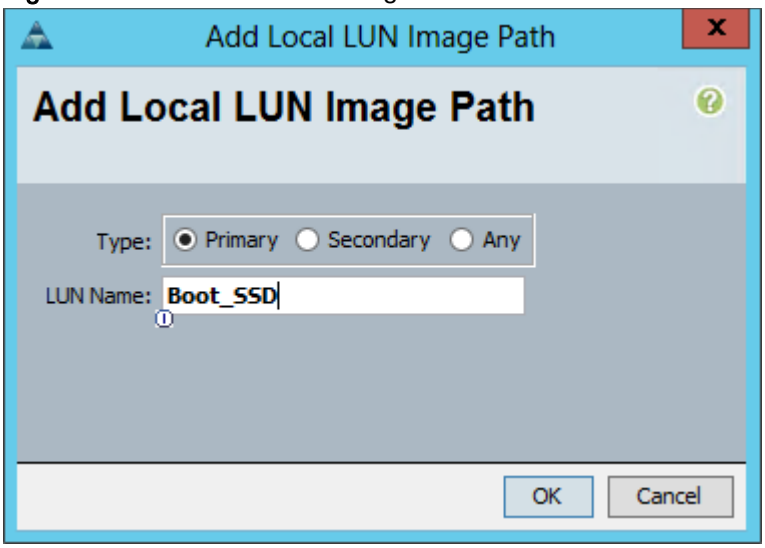
Figure 53 Boot Policy Screen



5. Enter `ucs` 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 54)

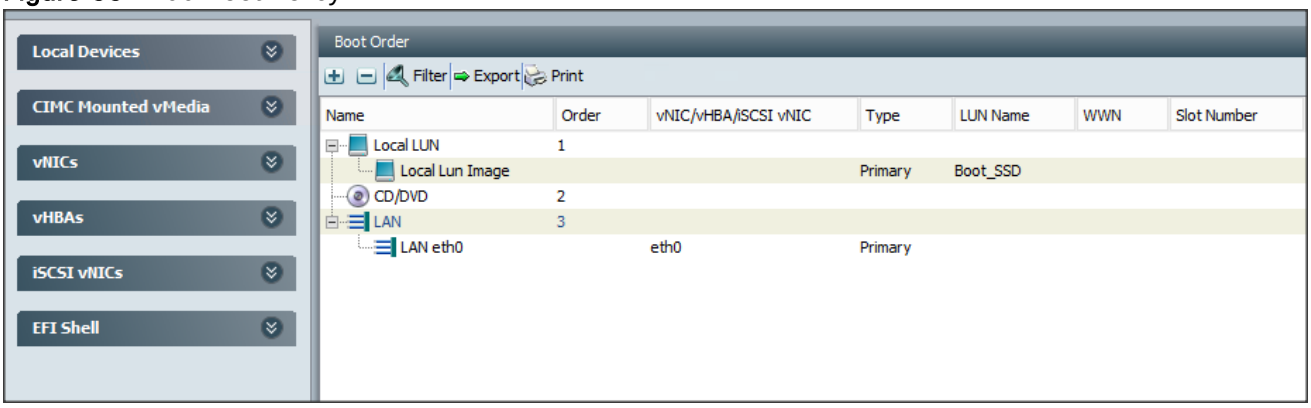
Note: The LUN name must match with the LUN name created earlier.

Figure 54 Add Local LUN Image Path



12. Expand Local Devices > Add CD/DVD and select Add Local CD/DVD. (Figure 55)
13. Expand `vNICs` and select `Add LAN Boot` and enter `eth0`.
14. Click `OK` to add the Boot Policy.
15. Click `OK`.

Figure 55 Add Boot Policy

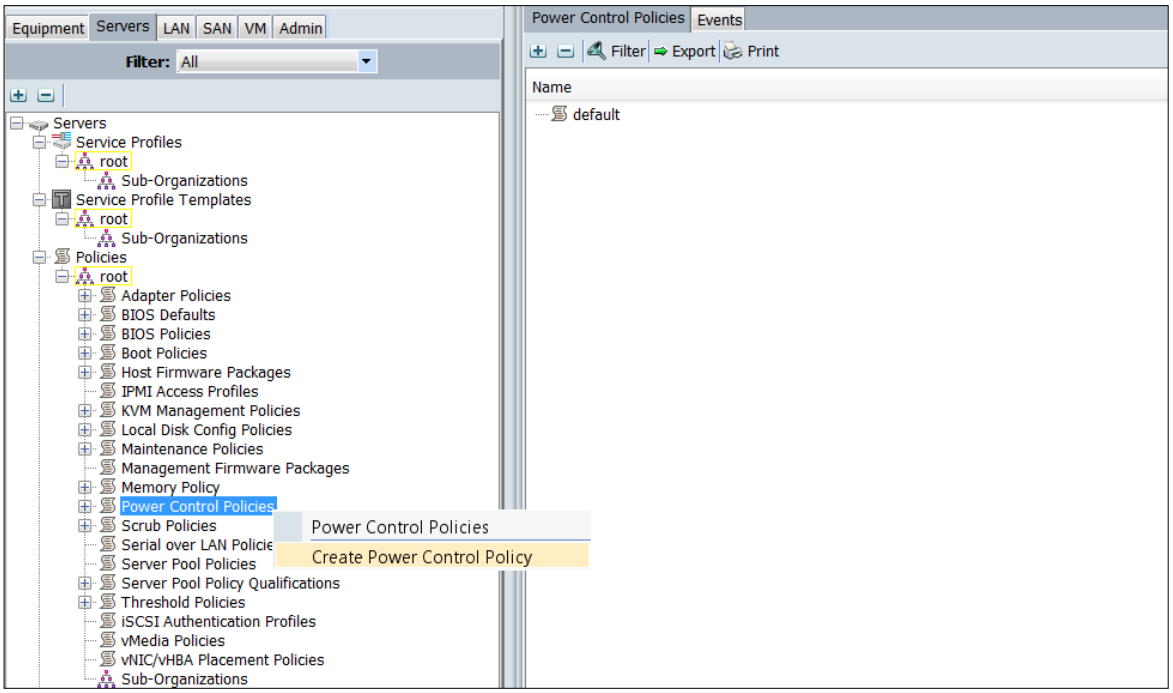


Creating Power Control Policy

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

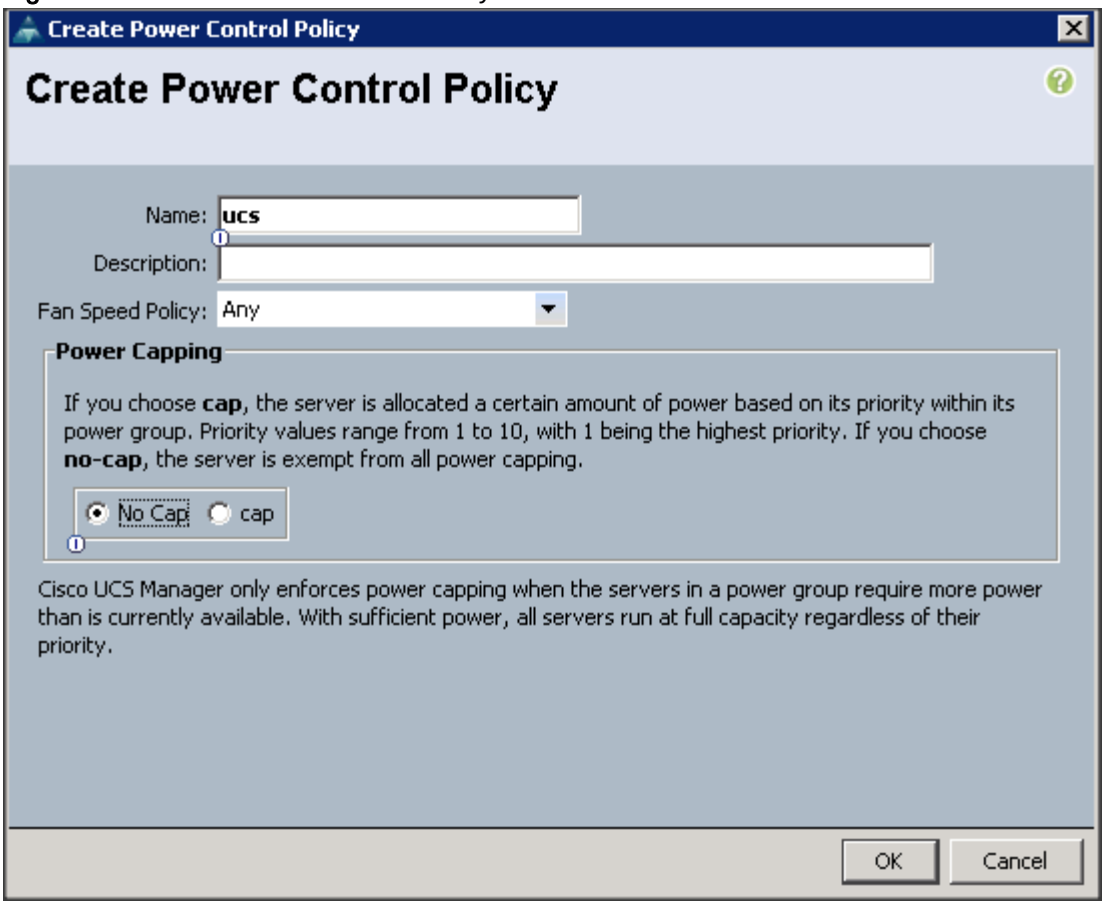
1. Select the `Servers` tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click the Power Control Policies. (Figure 56)
4. Select Create Power Control Policy.

Figure 56 Power Control Policy



5. Enter `ucs` as the Power Control policy `name`. (Figure 57)
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 57 Create Power Control Policy Screen

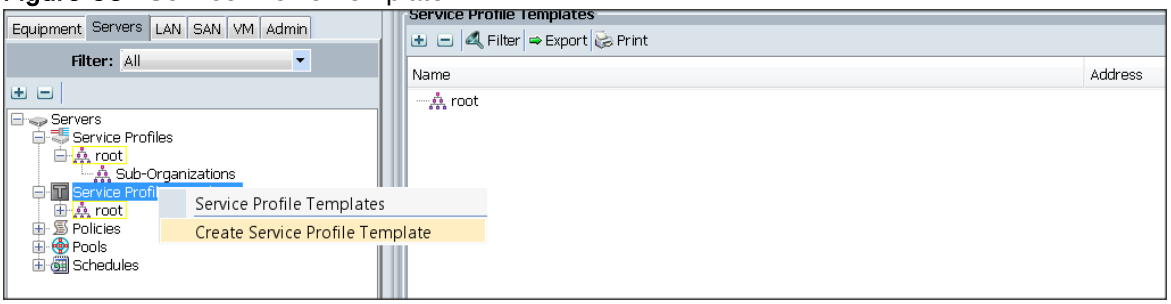


Creating a Service Profile Template

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 58)

Figure 58 Service Profile Template

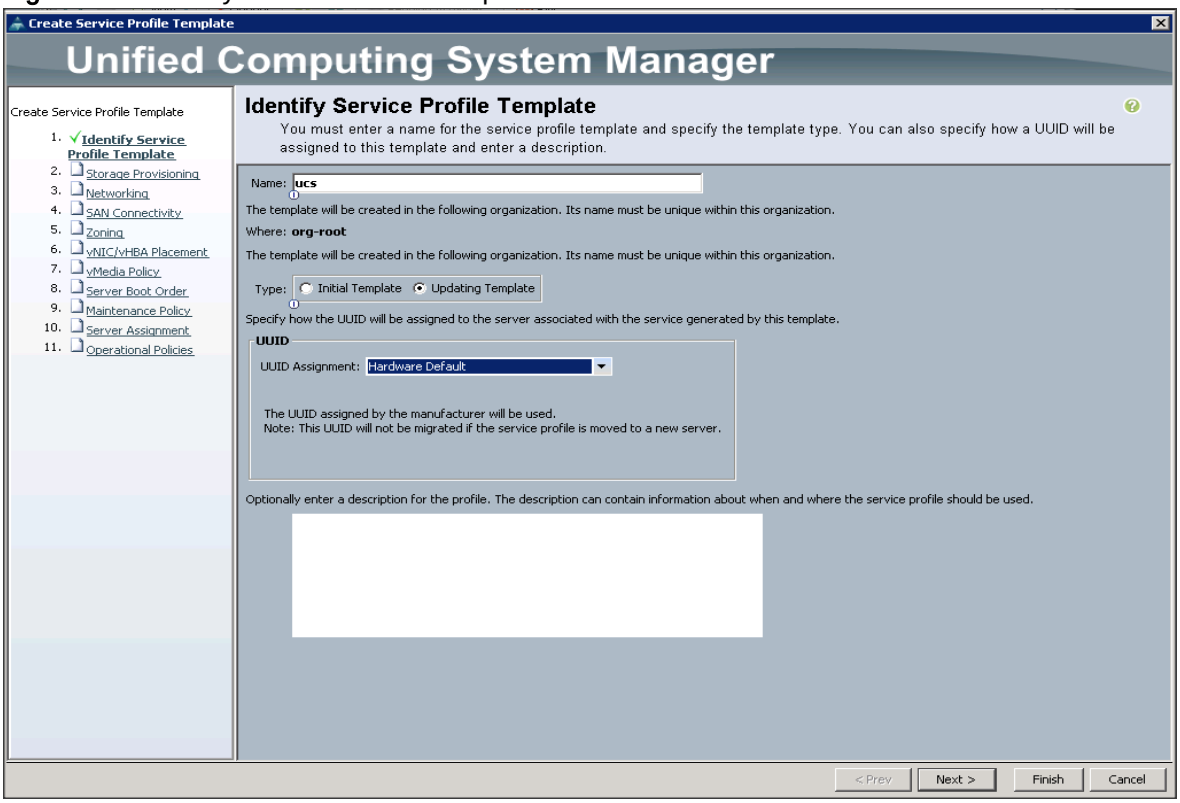


The Create Service Profile Template window appears. (Figure 59)

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 59 Identify Service Profile Template

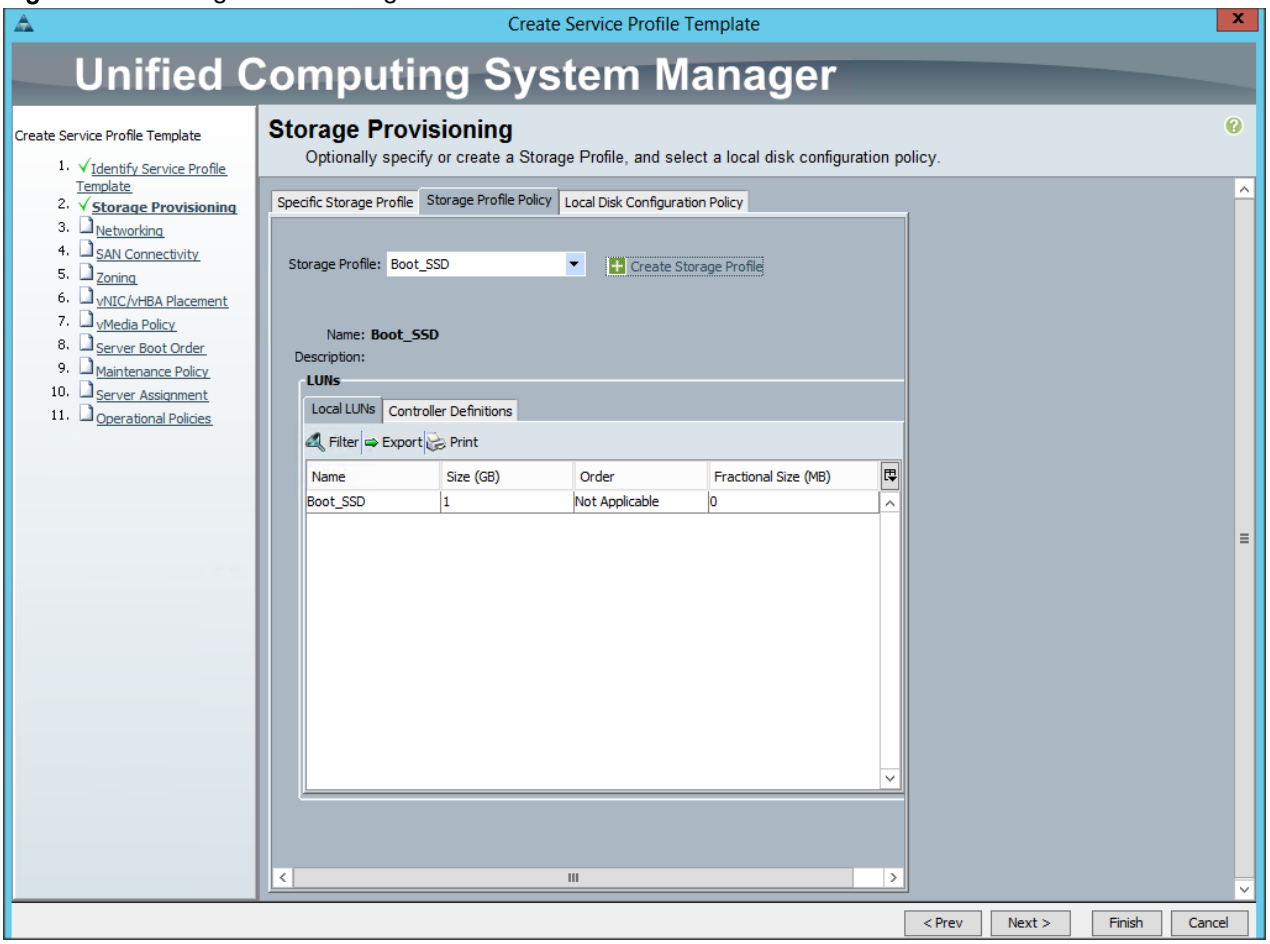


Configuring the Storage Provisioning for the Template

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 60)

Figure 60 Storage Provisioning



7. Go to the Local Disk Configuration Policy tab, and select `ucs` for the Local Storage. (Figure 61)

8. Click `Next` to continue to the next section.

Figure 61 Storage Provisioning /Local Storage

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

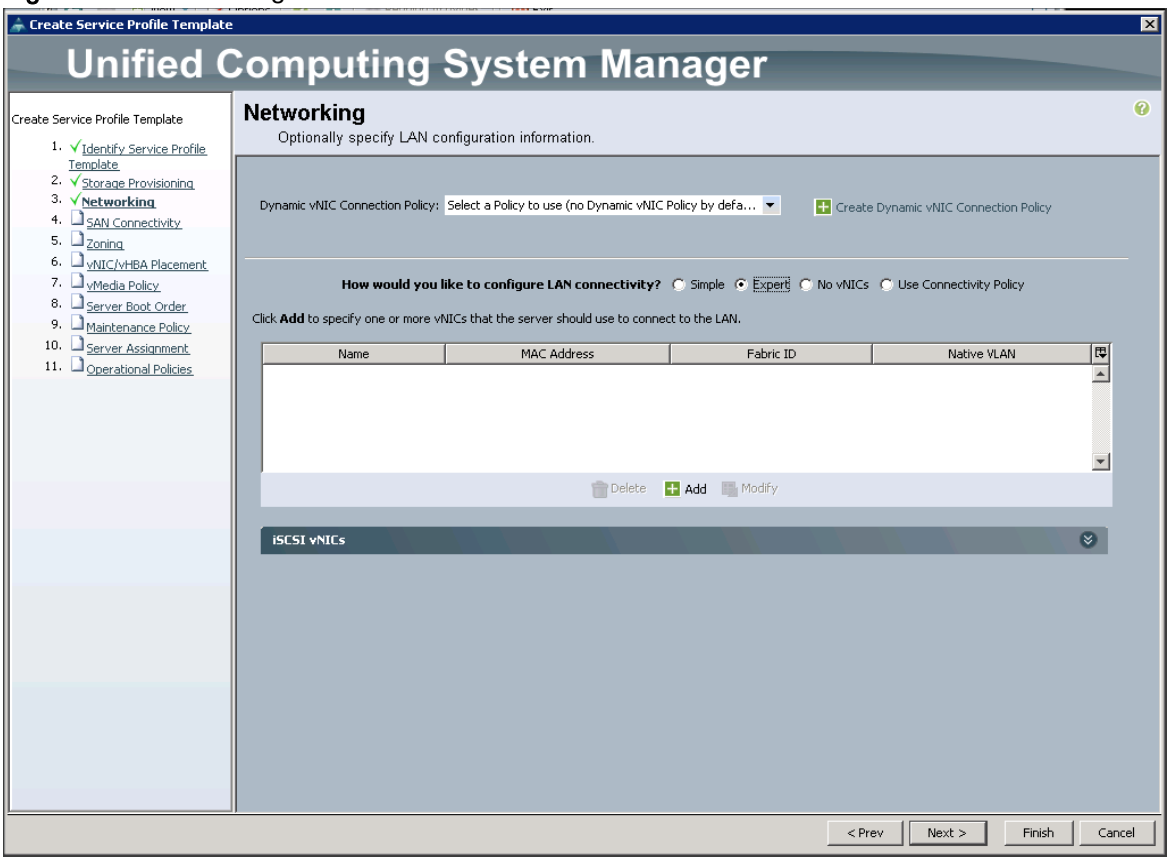
Configuring Network Settings for the Template

1. Keep the `Dynamic vNIC Connection Policy` field at the default. (Figure 62)

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 62 Networking



4. The Create vNIC window displays. Name the vNIC `eth0`. (Figure 63)
5. Select `ucs` 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 63 Create vNIC

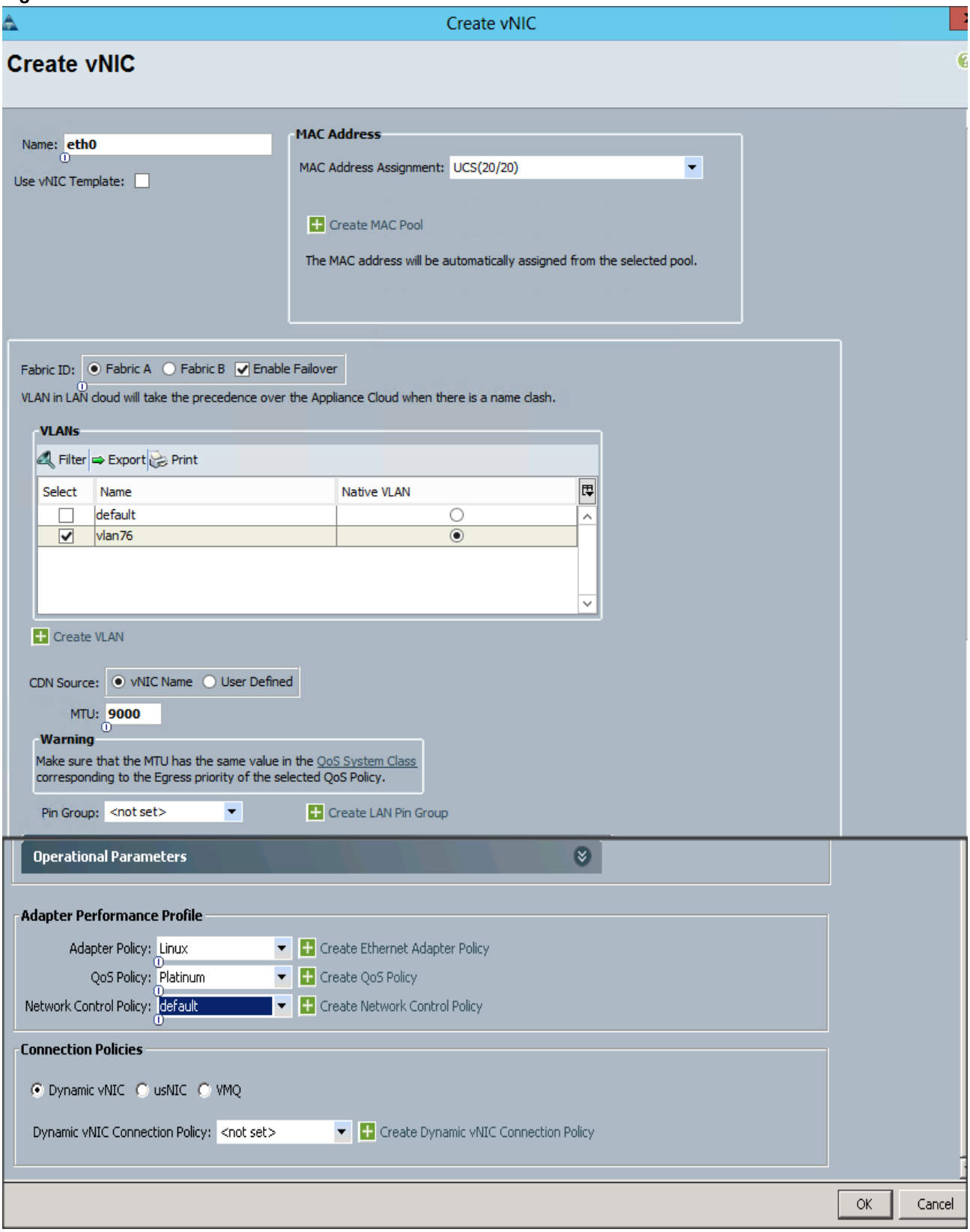
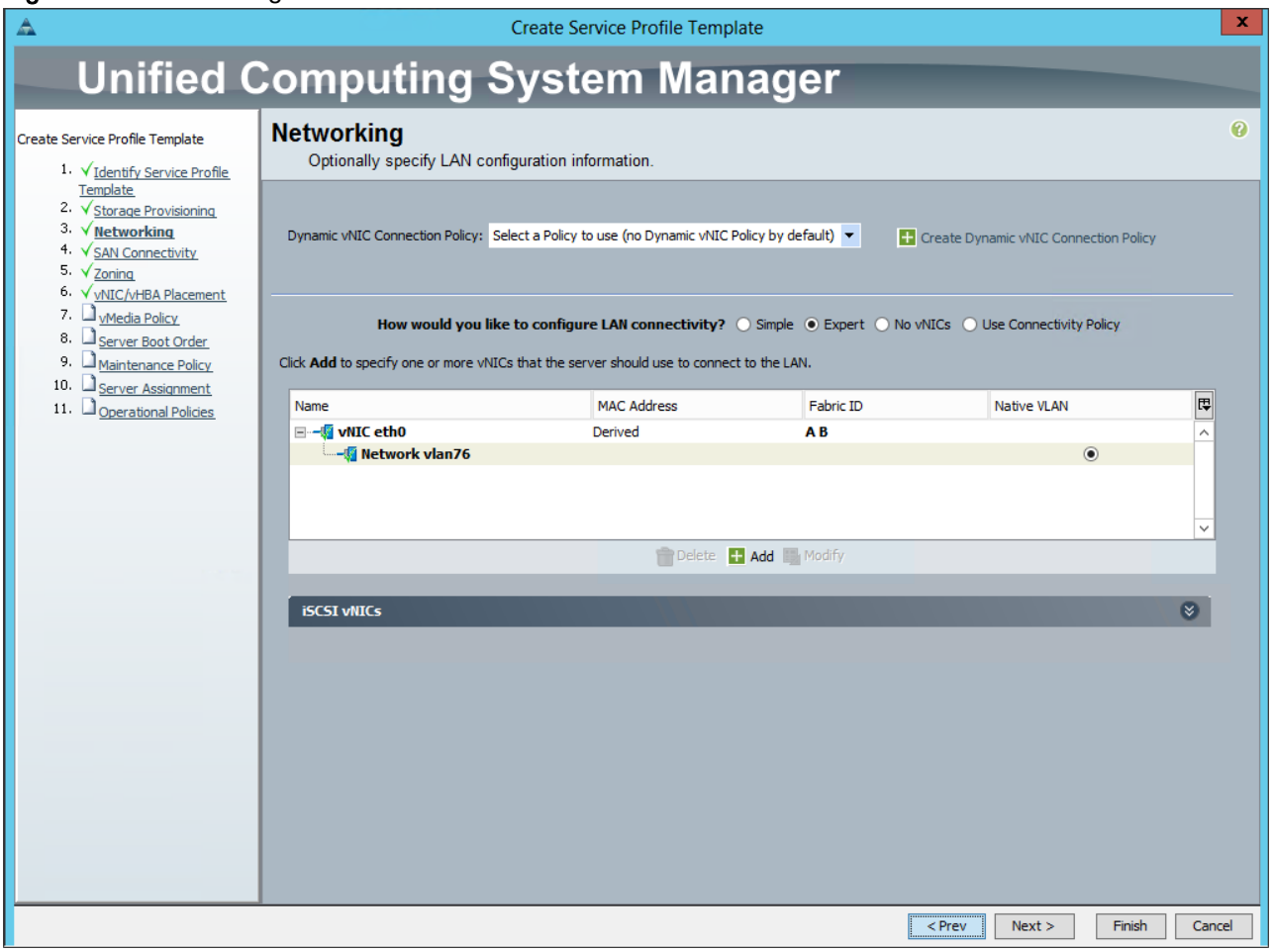


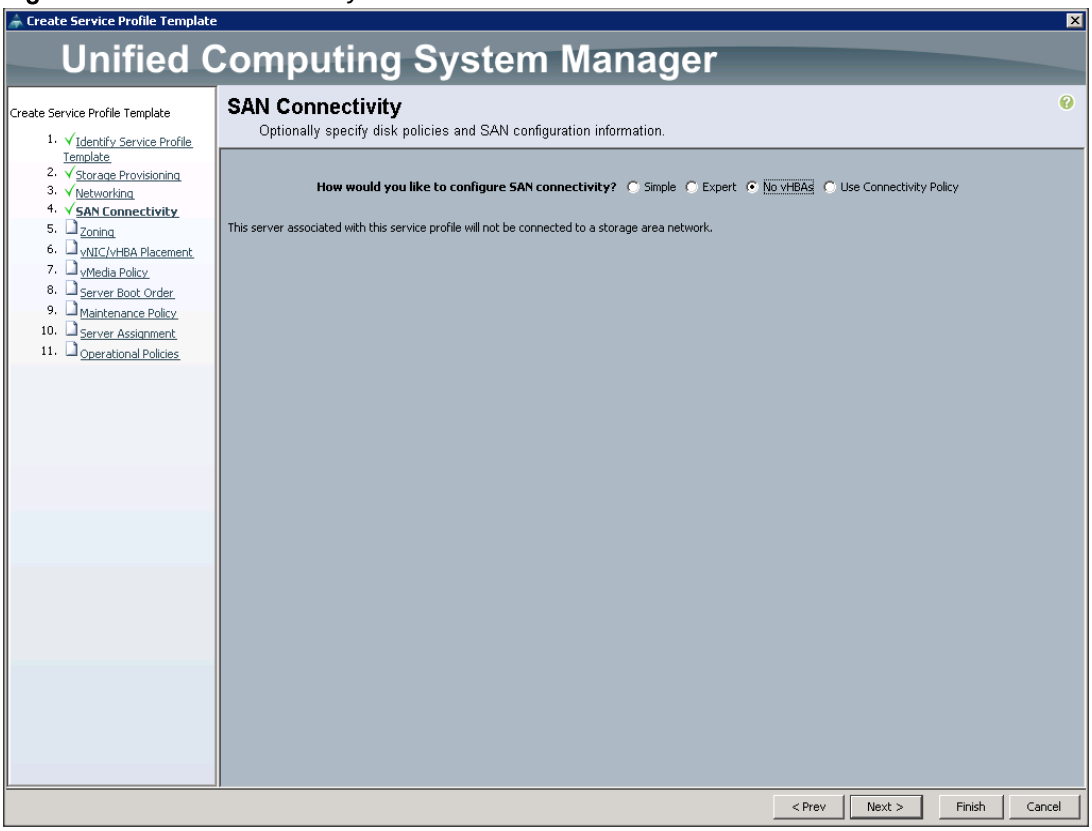
Figure 64 Networking LAN



13. Click `Next` to continue with SAN Connectivity. (Figure 64)

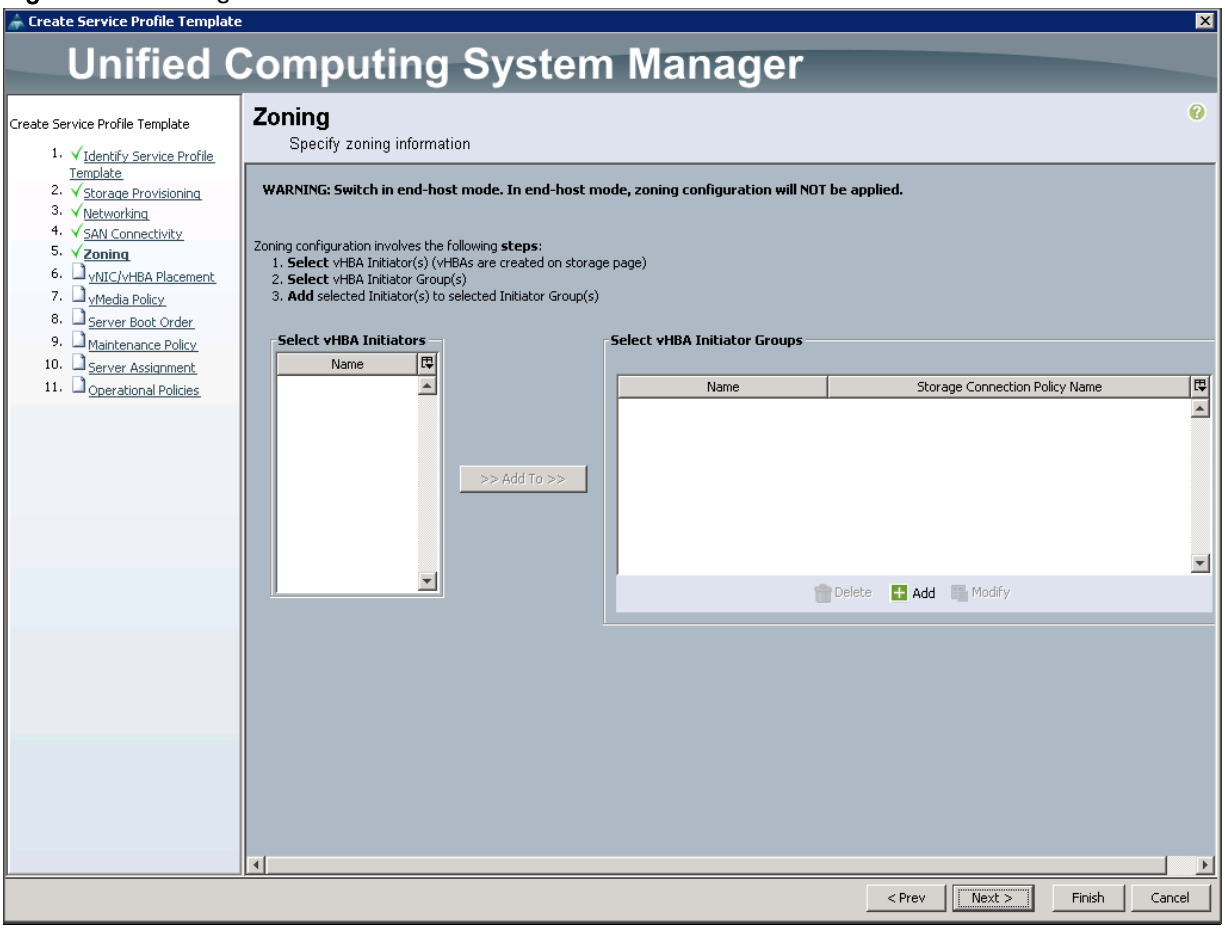
14. Select no vHBAs for How would you like to configure SAN Connectivity? (Figure 65)

Figure 65 SAN Connectivity



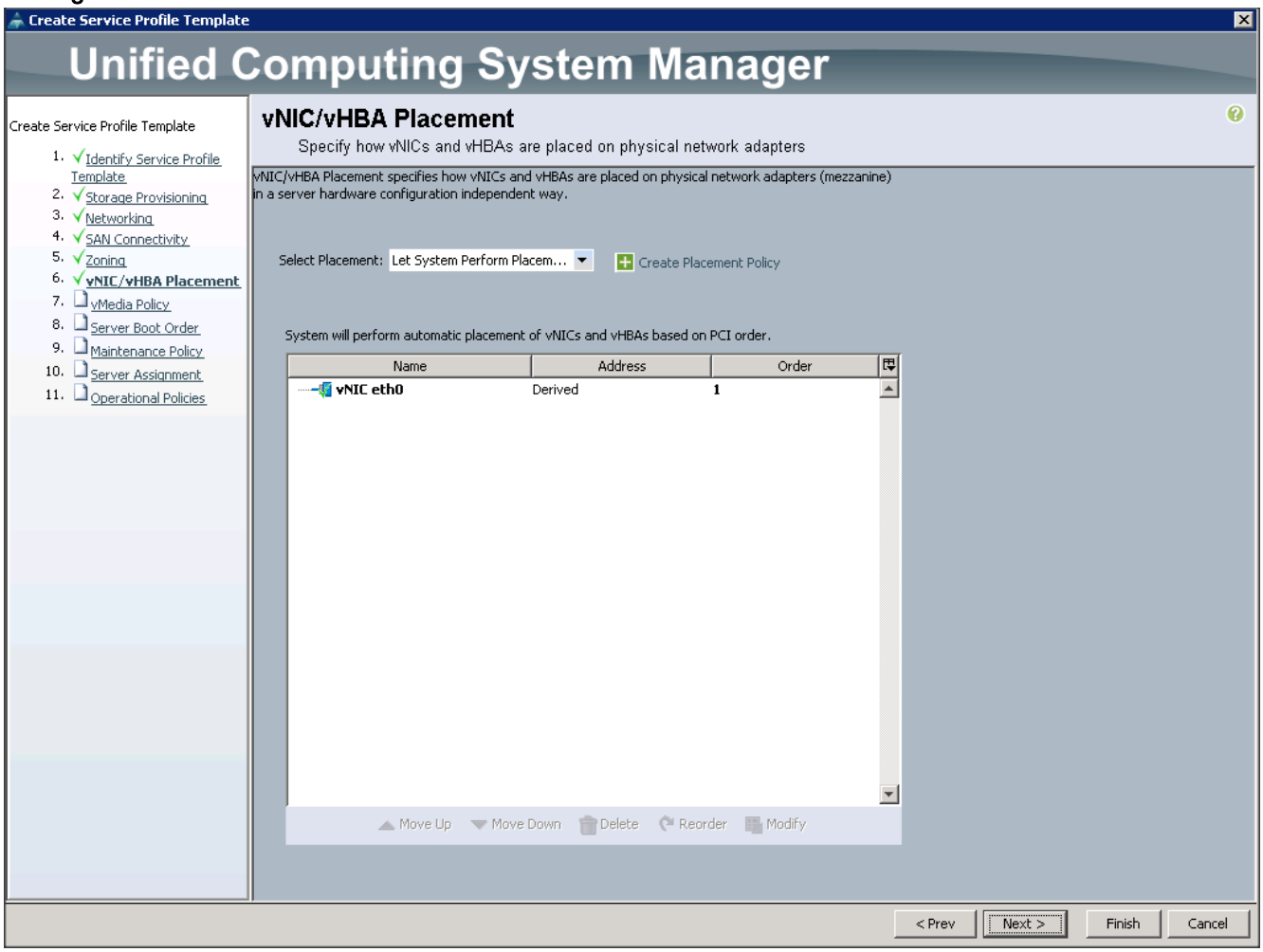
15. Click `Next` to continue with Zoning. (Figure 66)

Figure 66 Zoning



16. Click `Next` to continue with vNIC/vHBA placement. (Figure 67)

Figure 67 vNIC/vHBA Placement

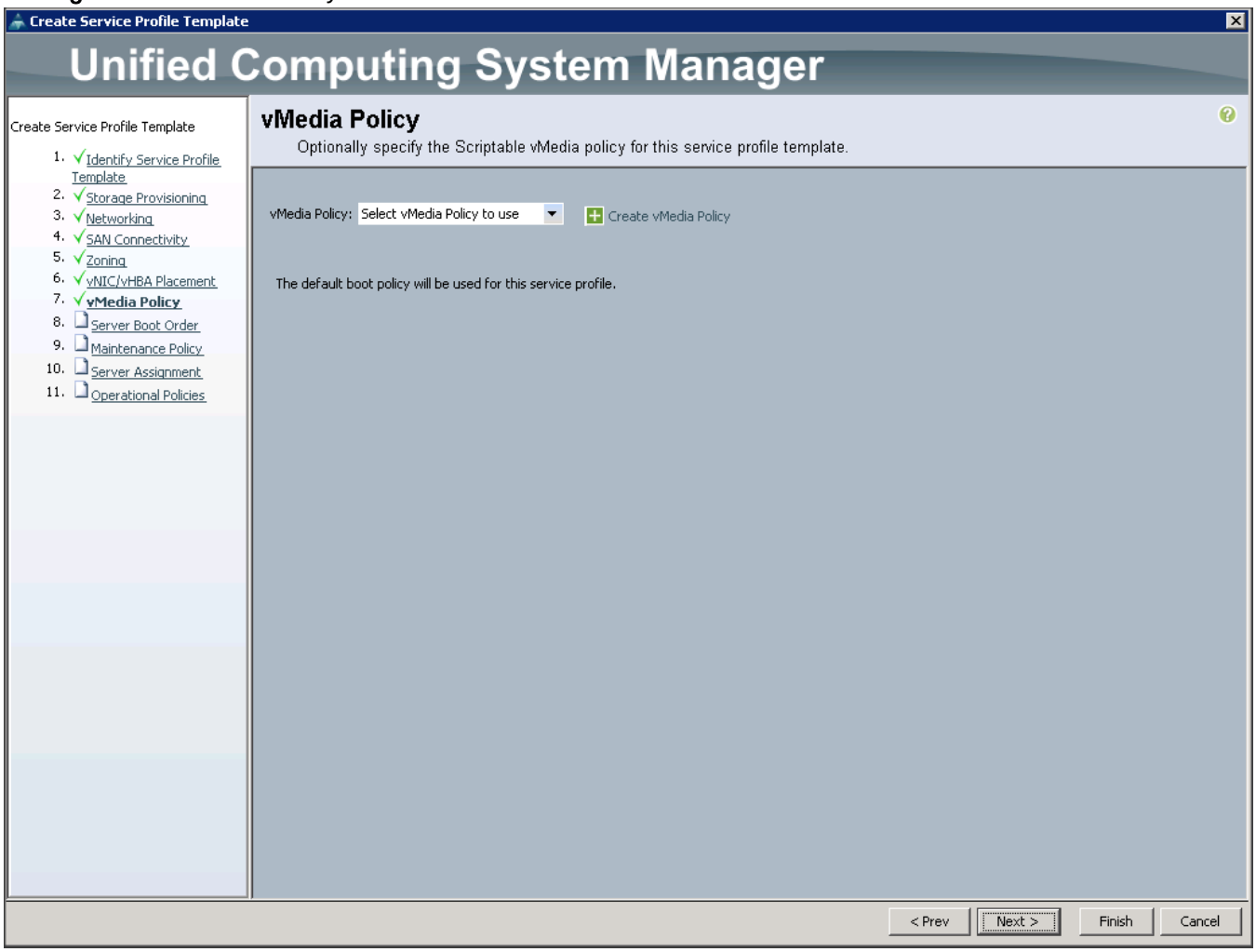


17. Click `Next` to configure vMedia Policy.

Configuring the vMedia Policy for the Template

1. Click `Next`. When the vMedia Policy window appears, to go to the next section. (Figure 68)

Figure 68 vMedia Policy

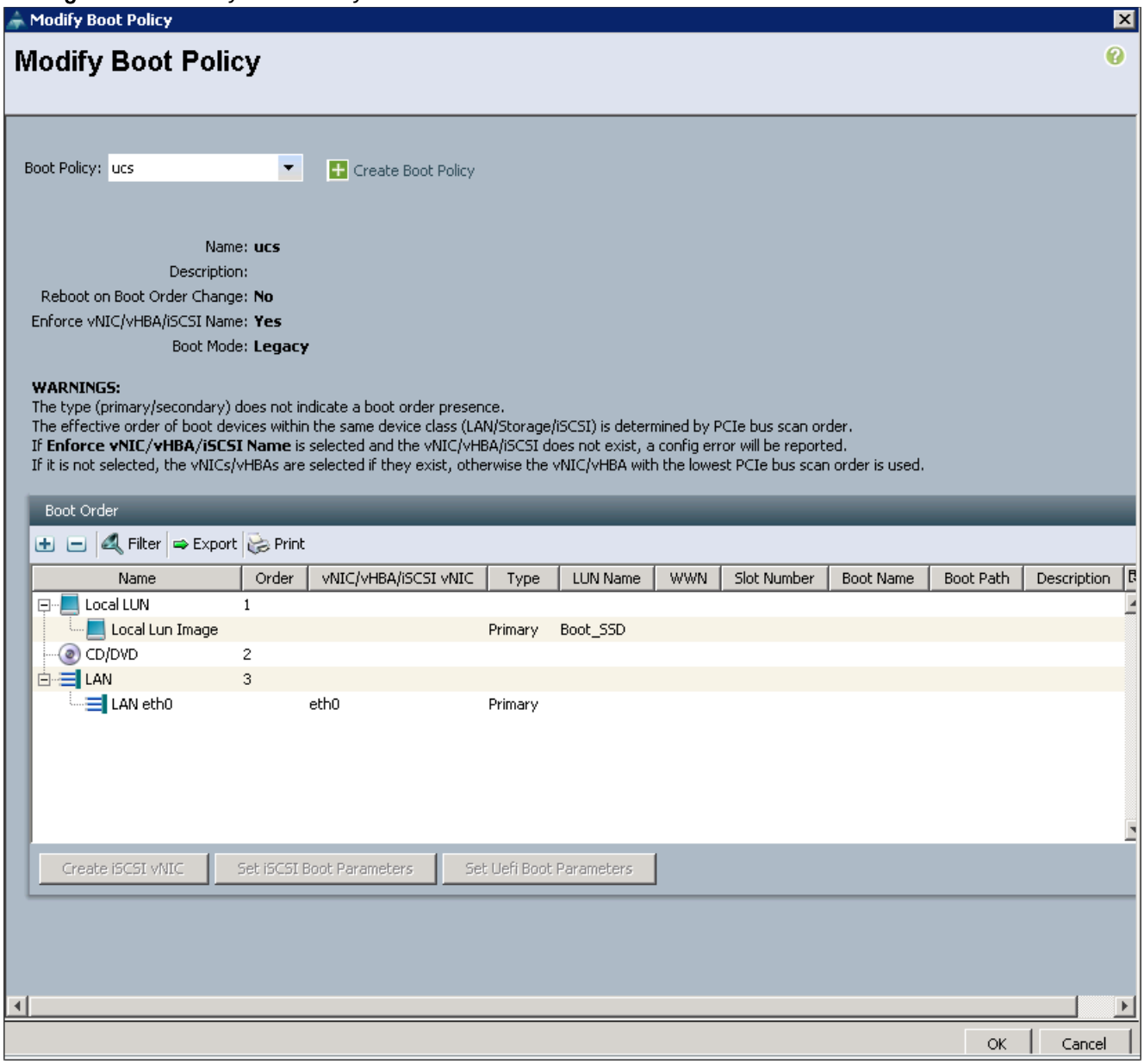


Configuring Server Boot Order for the Template

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

1. Select `ucs` in the Boot Policy name field. (Figure 69)
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.

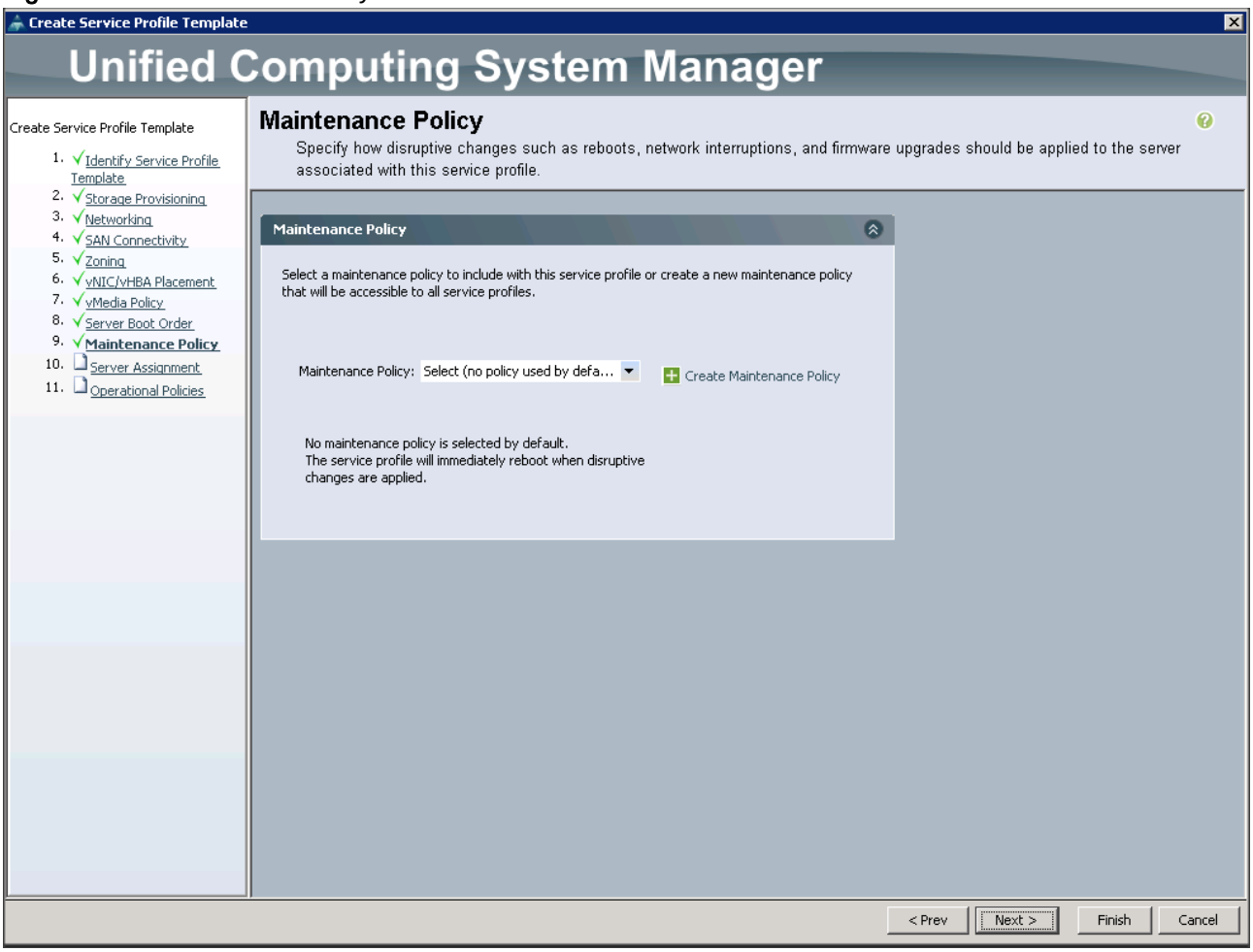
Figure 69 Modify Boot Policy



6. In the Maintenance Policy window, apply the maintenance policy. (Figure 70)

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

Figure 70 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 71)
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.

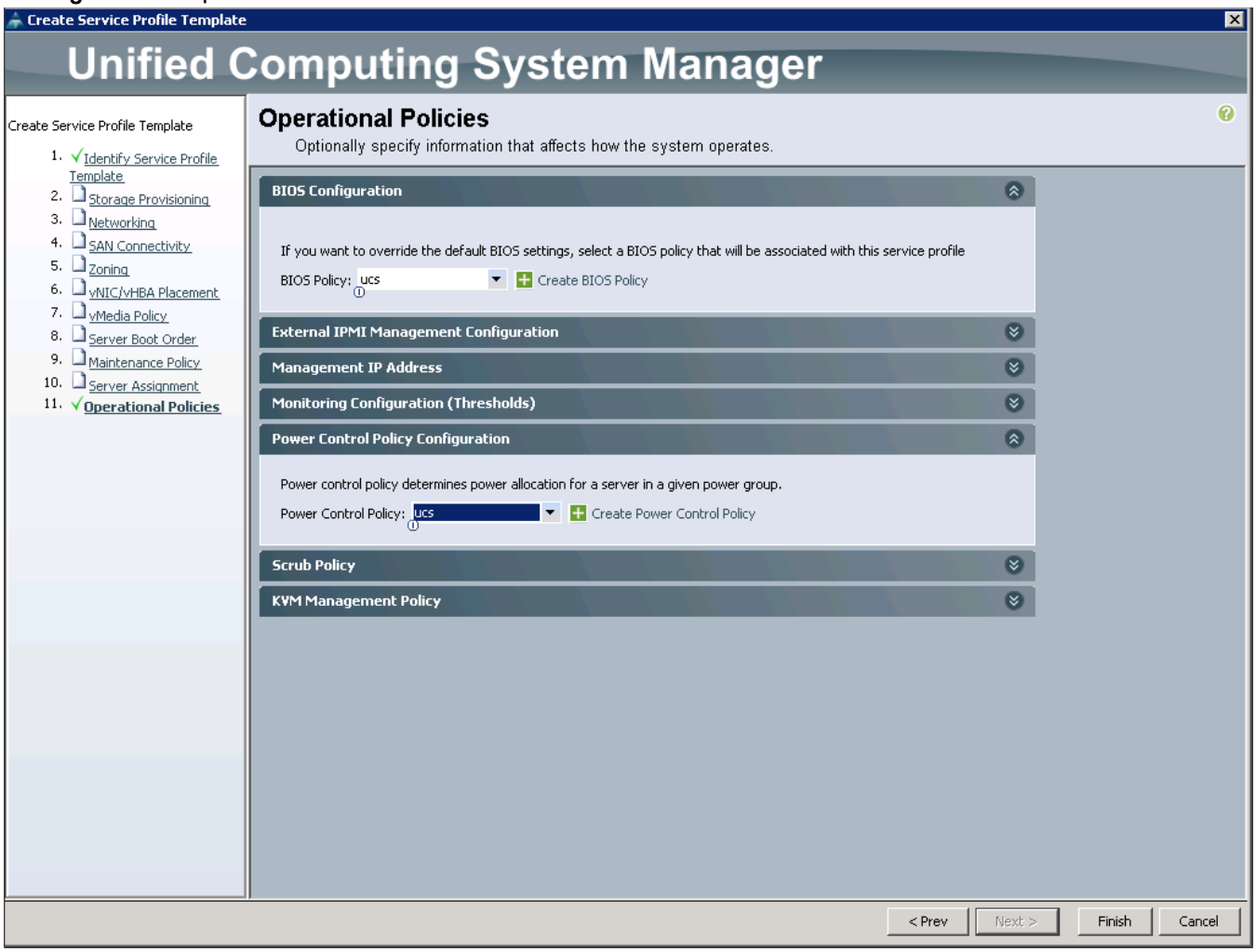
Figure 71 Server Assignment

Configuring Operational Policies for the Template

In the Operational Policies Window (Figure 72), complete the following steps:

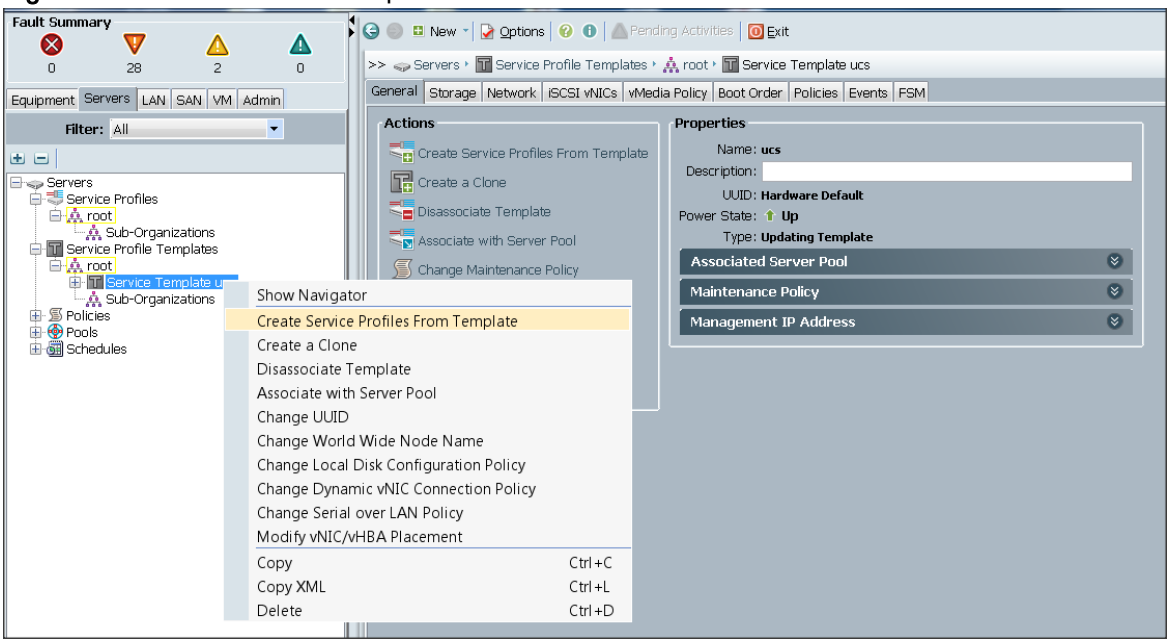
1. Select `ucs` in the `BIOS Policy` field.
2. Select `ucs` in the Power Control Policy field.

Figure 72 Operational Policies



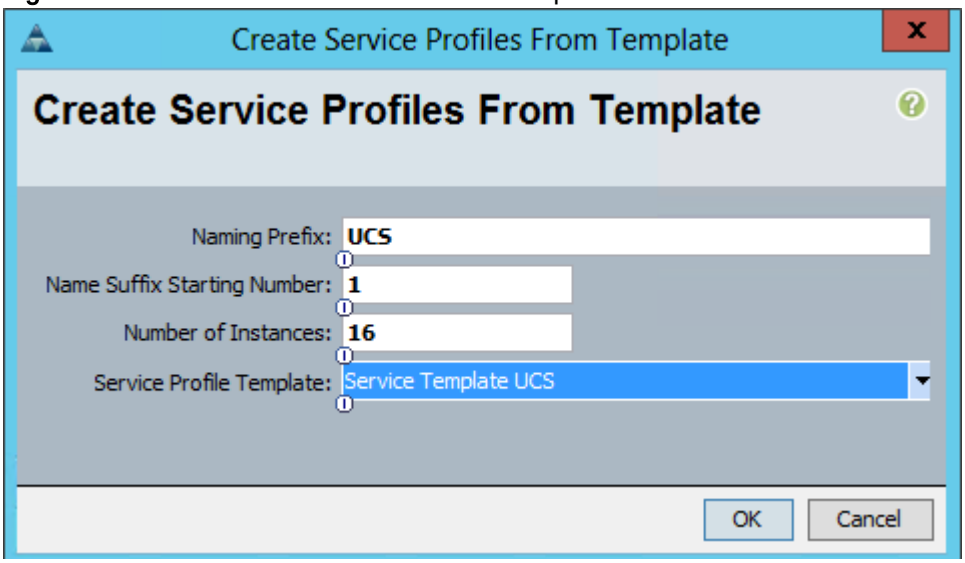
3. Click `Finish` to create the Service Profile template.
4. Click `OK` in the pop-up window to proceed.
5. Select the `Servers` tab in the left pane of the UCS Manager GUI. (Figure 73)
6. Go to `Service Profile Templates > root.`
7. Right-click `Service Profile Templates ucs.`
8. Select `Create Service Profiles From Template.`

Figure 73 Service Profile Templates



The Create Service Profiles from Template window appears. (Figure 74)

Figure 74 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 below.

Figure 75 UCS Manager Server Setup

Filter | Export | Print

Name	Chassis ID	PID	Model	User Label	Cores	Cores Enabled	Memory	Adapters	NICs	HBAs	Overall Status	Operability	Power State	Assoc S
Server 1	1	UCSC-C3K-M4SRB	Cisco UCS C3X60M4		28	28	262144	1	6	0	OK	Operable	On	Asso
Server 2	1	UCSC-C3K-M4SRB	Cisco UCS C3X60M4		28	28	262144	1	6	0	OK	Operable	On	Asso
Server 1	2	UCSC-C3K-M4SRB	Cisco UCS C3X60M4		28	28	262144	1	6	0	OK	Operable	On	Asso
Server 2	2	UCSC-C3K-M4SRB	Cisco UCS C3X60M4		28	28	262144	1	6	0	OK	Operable	On	Asso
Server 1	3	UCSC-C3K-M4SRB	Cisco UCS C3X60M4		28	28	262144	1	6	0	OK	Operable	On	Asso
Server 2	3	UCSC-C3K-M4SRB	Cisco UCS C3X60M4		28	28	262144	1	6	0	OK	Operable	On	Asso
Server 1	4	UCSC-C3K-M4SRB	Cisco UCS C3X60M4		28	28	262144	1	6	0	OK	Operable	On	Asso
Server 2	4	UCSC-C3K-M4SRB	Cisco UCS C3X60M4		28	28	262144	1	6	0	OK	Operable	On	Asso

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.

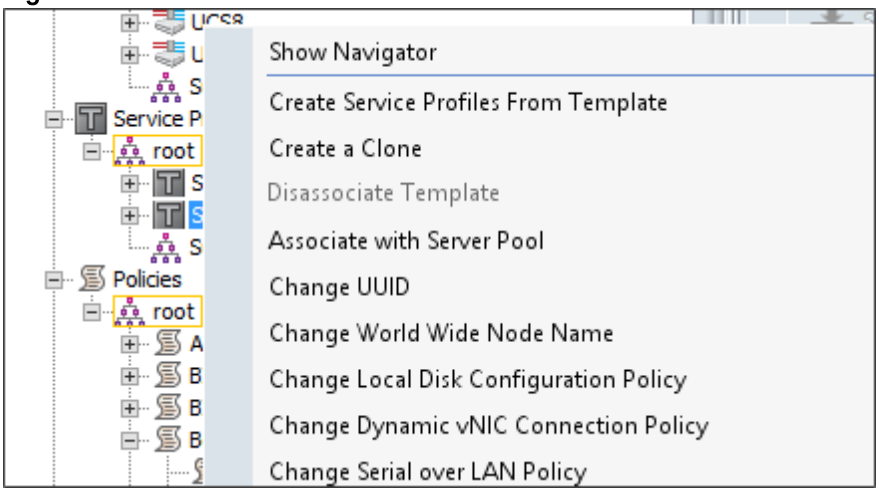
To configure an organization within the Cisco UCS Manager GUI, complete the following steps:

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

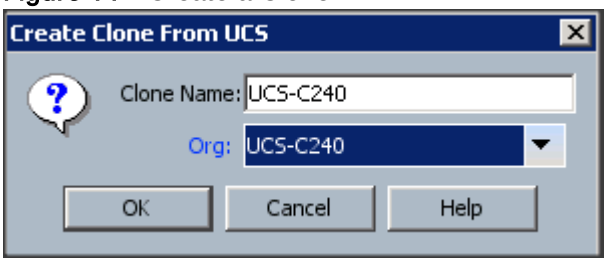
1. Click on Servers tab, go to Service Profile Template → root.
2. Right click on the existing template UCS and click Create a Clone. (Figure 76)

Figure 76 Create a Clone



3. In the Clone Name, enter UCS-C240 and from the Org drop down list choose UCS-C240 and click OK. (Figure 77)

Figure 77 Create a Clone

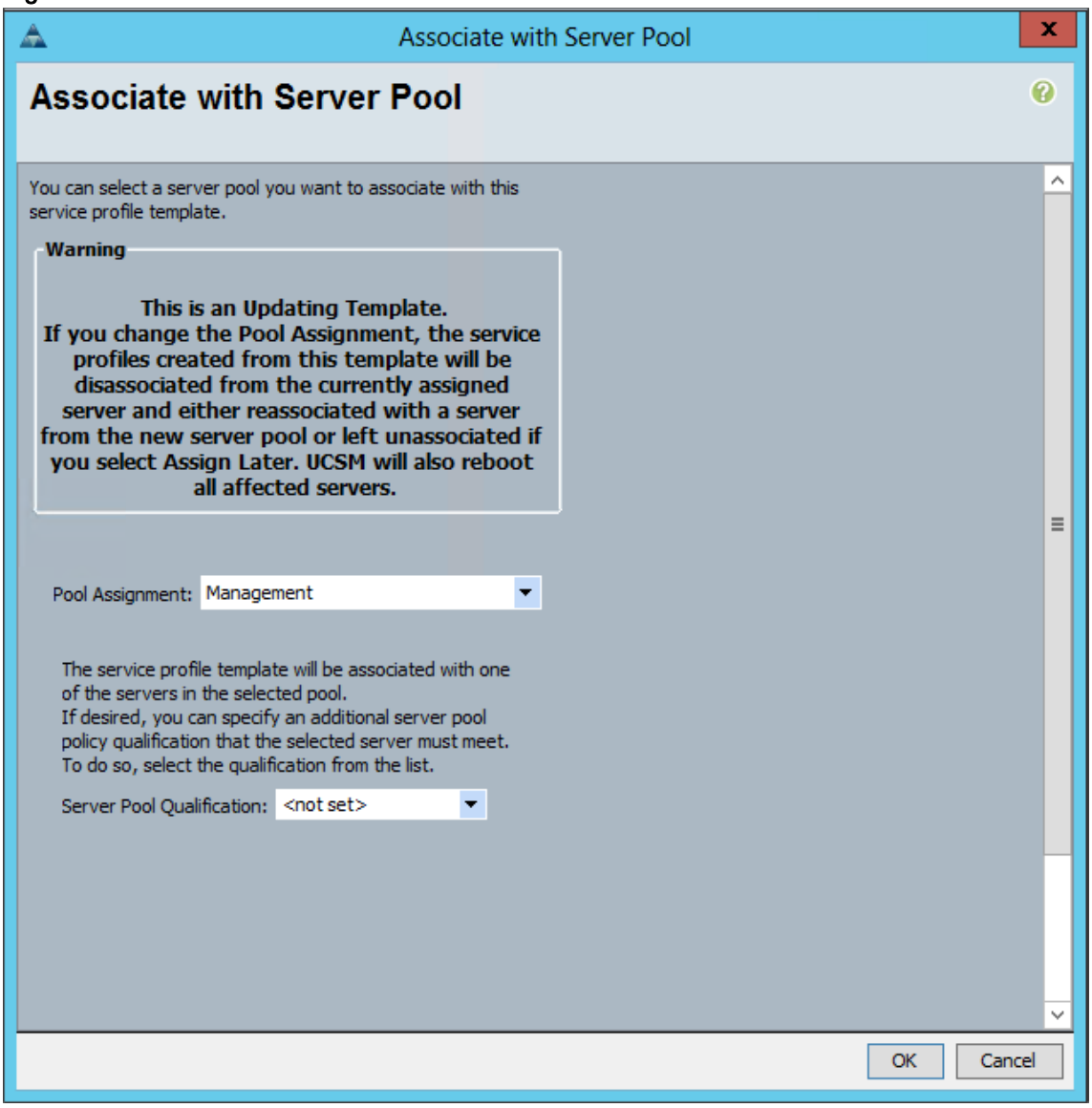


4. Go to root → Sub-Organization → UCS-C240 and select the Service Template UCS-C240.

5. In the right window general tab click Associate with Server pool. (Figure 78)

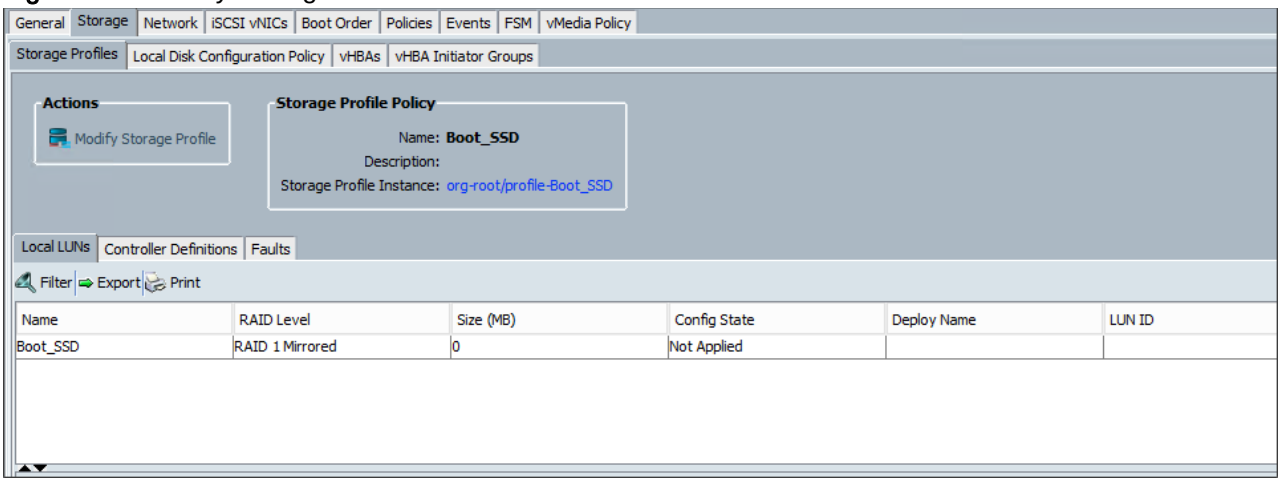
6. In the Pool Assignment drop down list choose Management and click OK.

Figure 78 Associate with Server Pool



7. In the right window select the Storage tab and click Modify Storage Profile. (Figure 79)

Figure 79 Modify Storage Profile



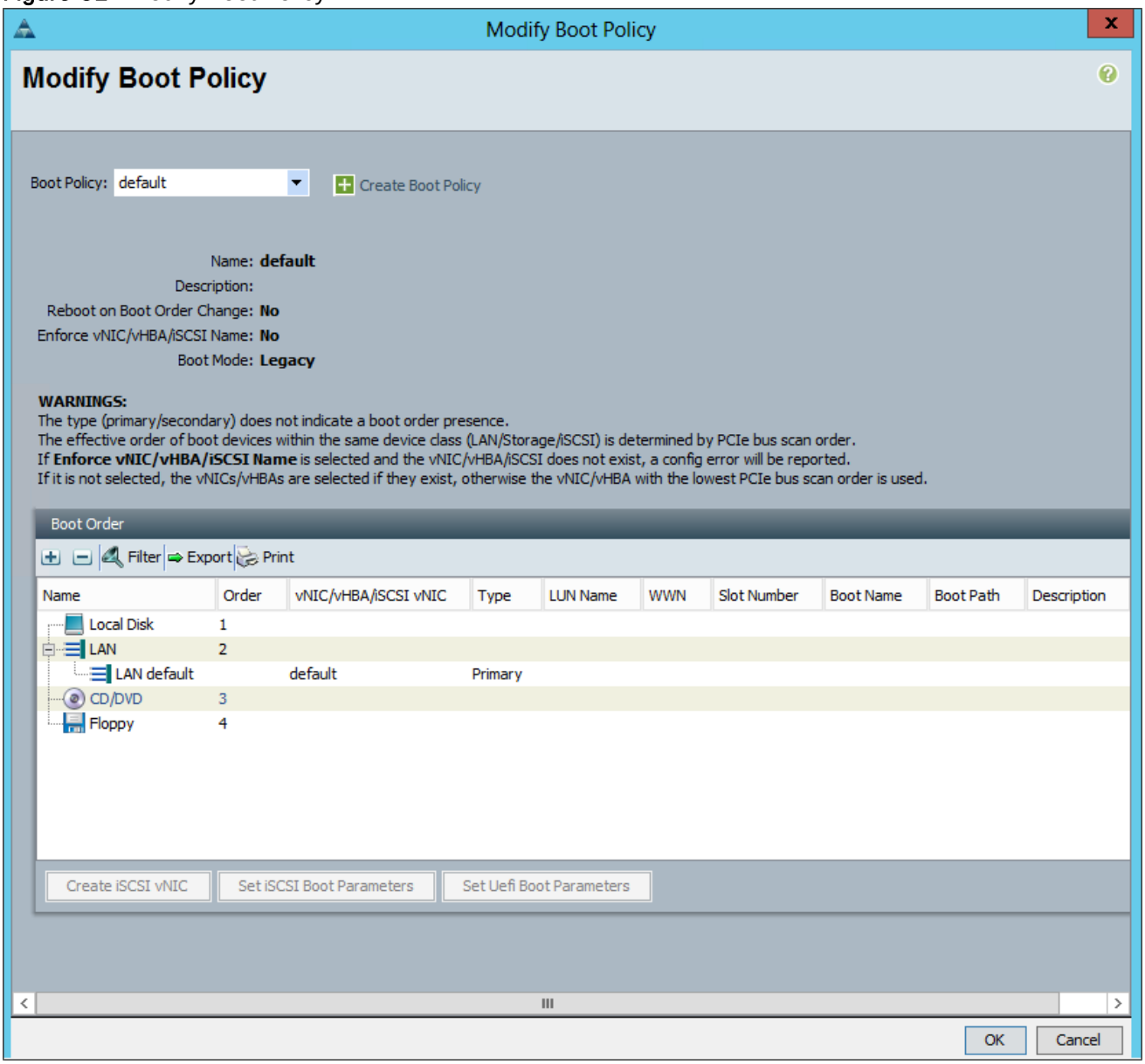
8. From the Storage profile drop down list choose No Storage Profile and click OK. (Figure 80)

Figure 80 No Storage Profile

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

10. From the Boot Policy drop down list choose Default and click OK. (Figure 81)

Figure 81 Modify Boot Policy

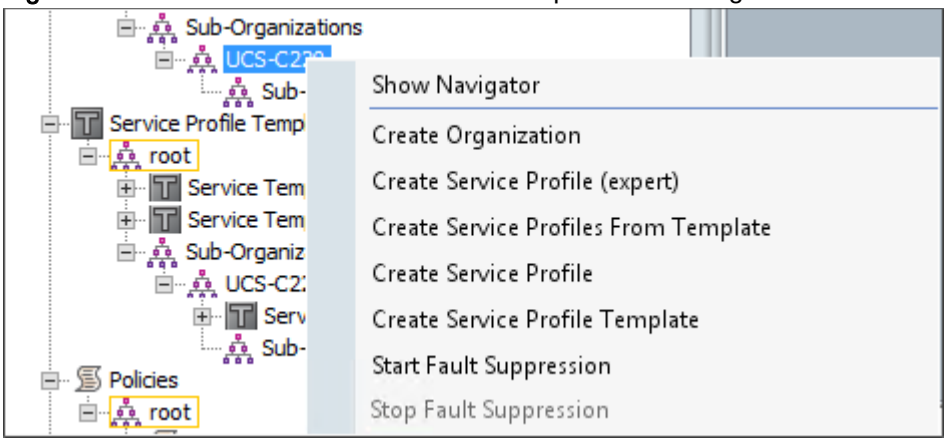


Creating Service Profile from Template

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

2. Right click and select Create Service Profiles from Template. (Figure 82)

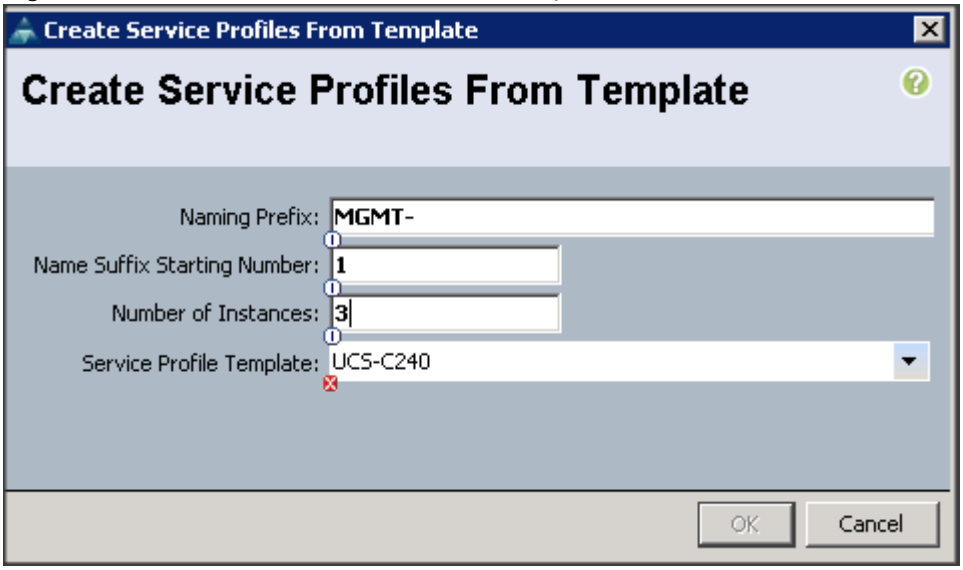
Figure 82 Create Service Profiles from Template for Management nodes



In the Create Service Profiles from Template screen: (Figure 83)

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

Figure 83 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 on Management Nodes

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

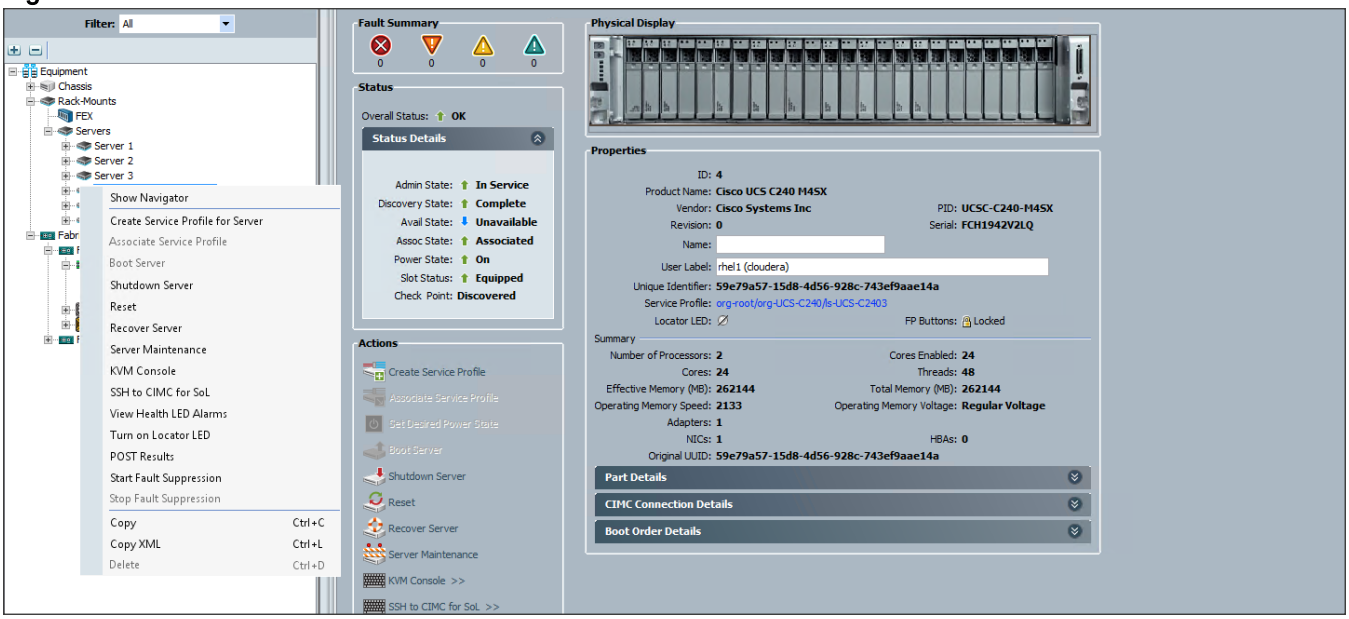
install the Red Hat Linux operating system. The installation procedure described in this deployment guide uses KVM console and virtual media from Cisco UCS Manager.

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

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

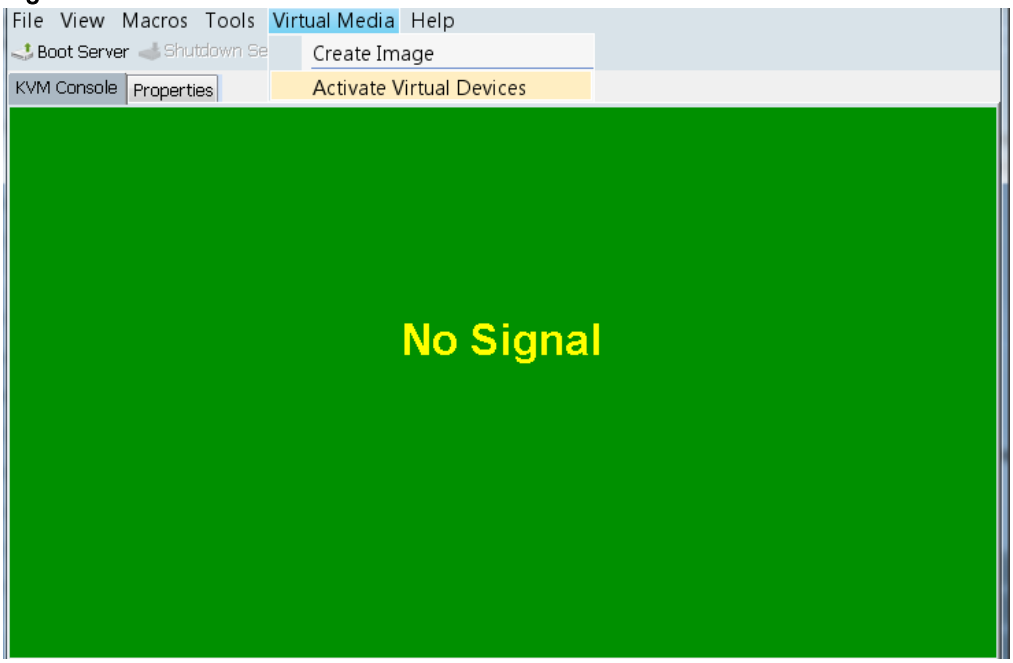
1. Log in to the Cisco UCS 6332 Fabric Interconnect and launch the Cisco UCS Manager application.
2. Select the `Equipment` tab as shown in Figure 84.
3. In the navigation pane expand `Rack-Mounts` and then `Servers`.
4. Right click on the server and select `KVM Console`.
5. In the KVM window, select the `Virtual Media` tab.

Figure 84 KVM Console



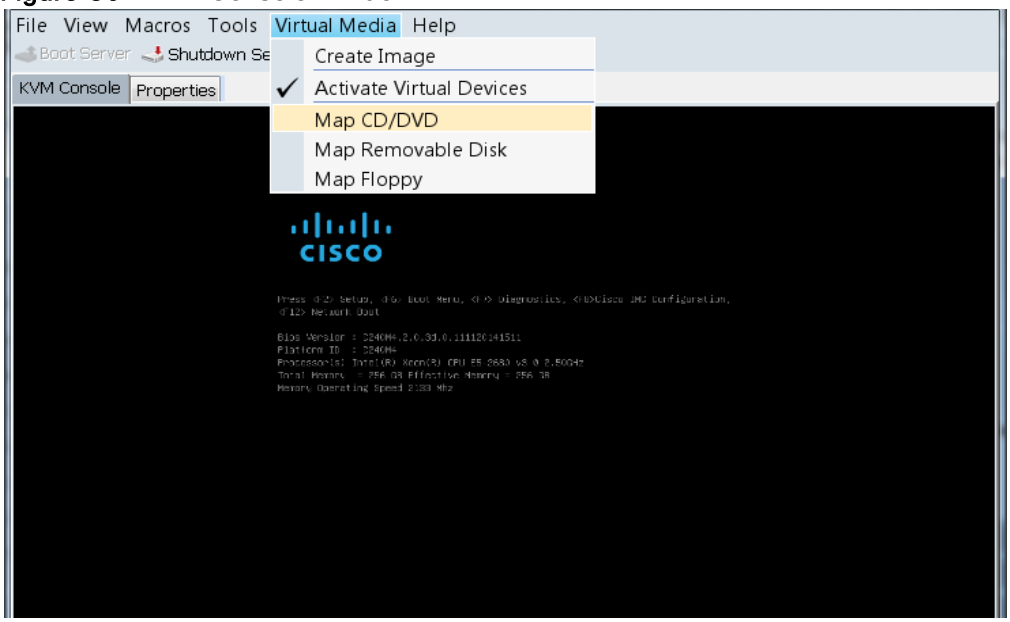
6. Click the `Activate Virtual Devices` found in the Virtual Media tab. (Figure 85)

Figure 85 Virtual Media Tab



7. In the KVM window (Figure 86), select the Virtual Media tab and click the `Map CD/DVD`.

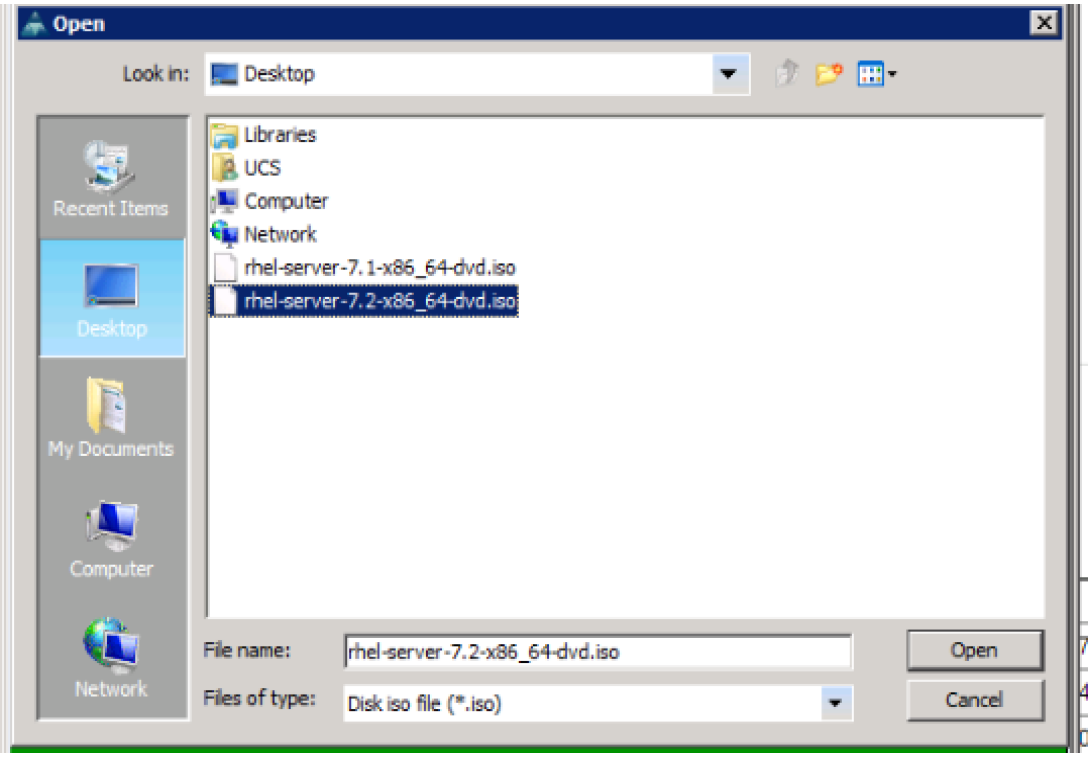
Figure 86 KVM Console Window



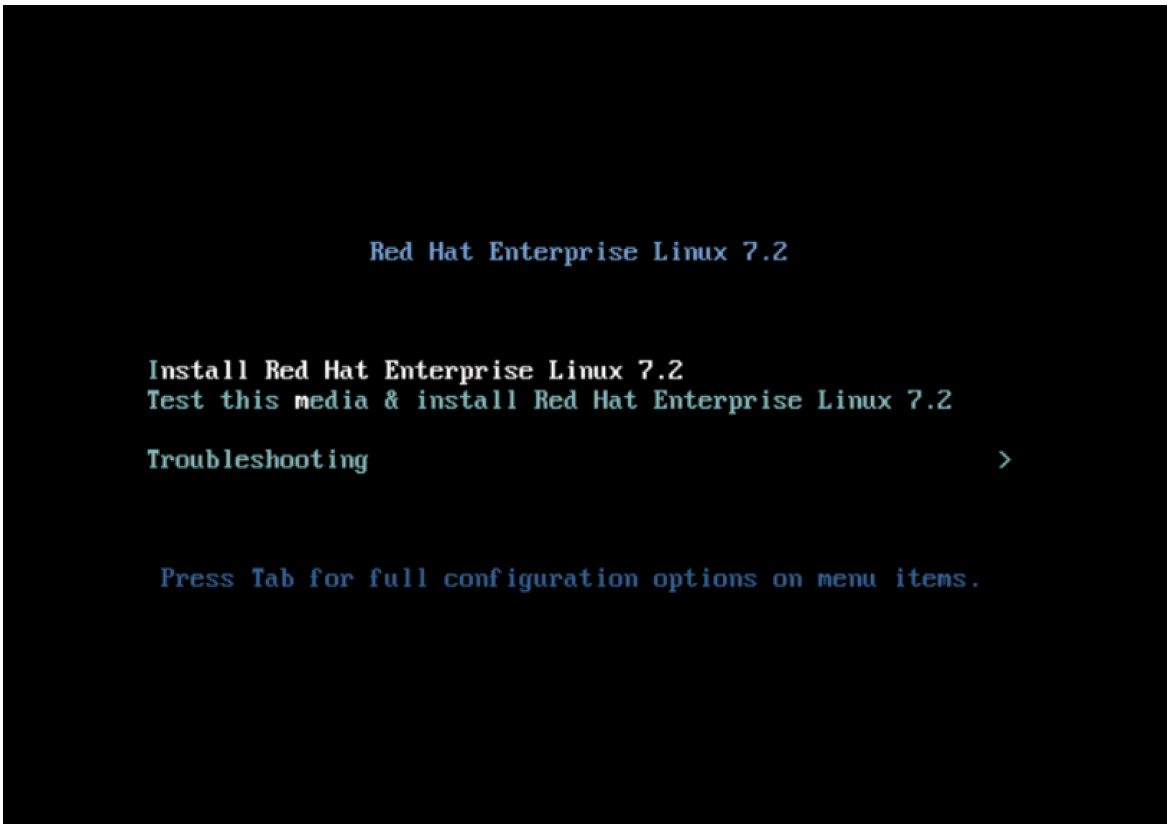
8. Browse to the Red Hat Enterprise Linux Server 7.2 installer ISO image file.

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.



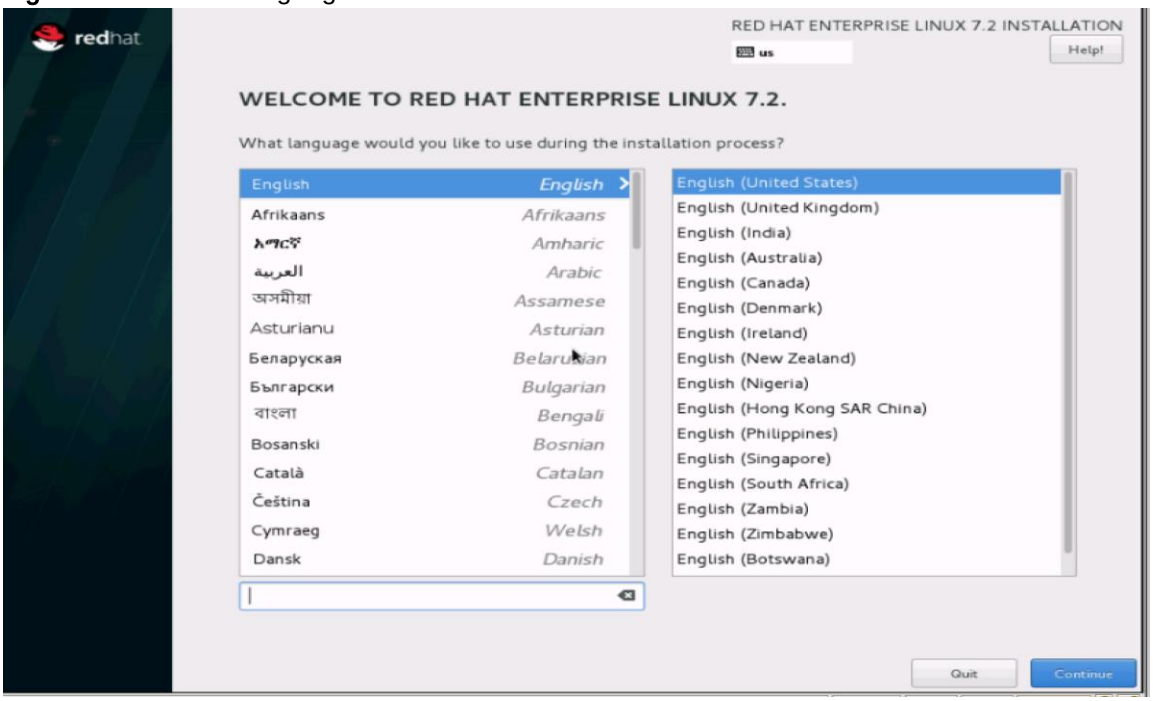
10. In the KVM window, select the KVM tab to monitor during boot.
11. In the KVM window, select the `Macros > Static Macros > Ctrl-Alt-Del` button in the upper left corner.
12. Click `OK`.
13. Click `OK` to reboot the system.
14. On reboot, the machine detects the presence of the Red Hat Enterprise Linux Server 7.2 install media.
15. Select the Install or Upgrade an Existing System.



16. Skip the Media test and start the installation.

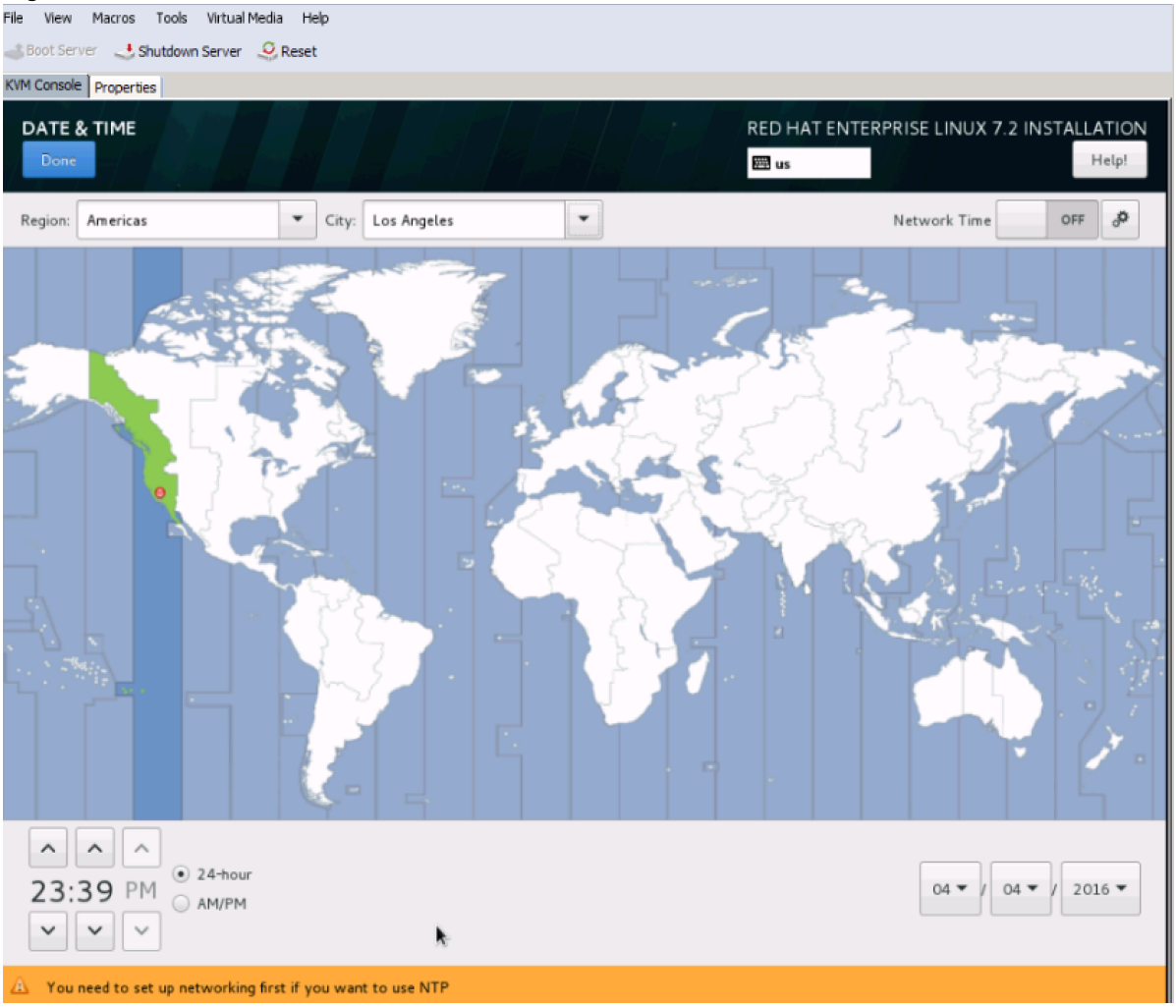
17. Select language of installation (Figure 87), and click Continue.

Figure 87 Select Language Window



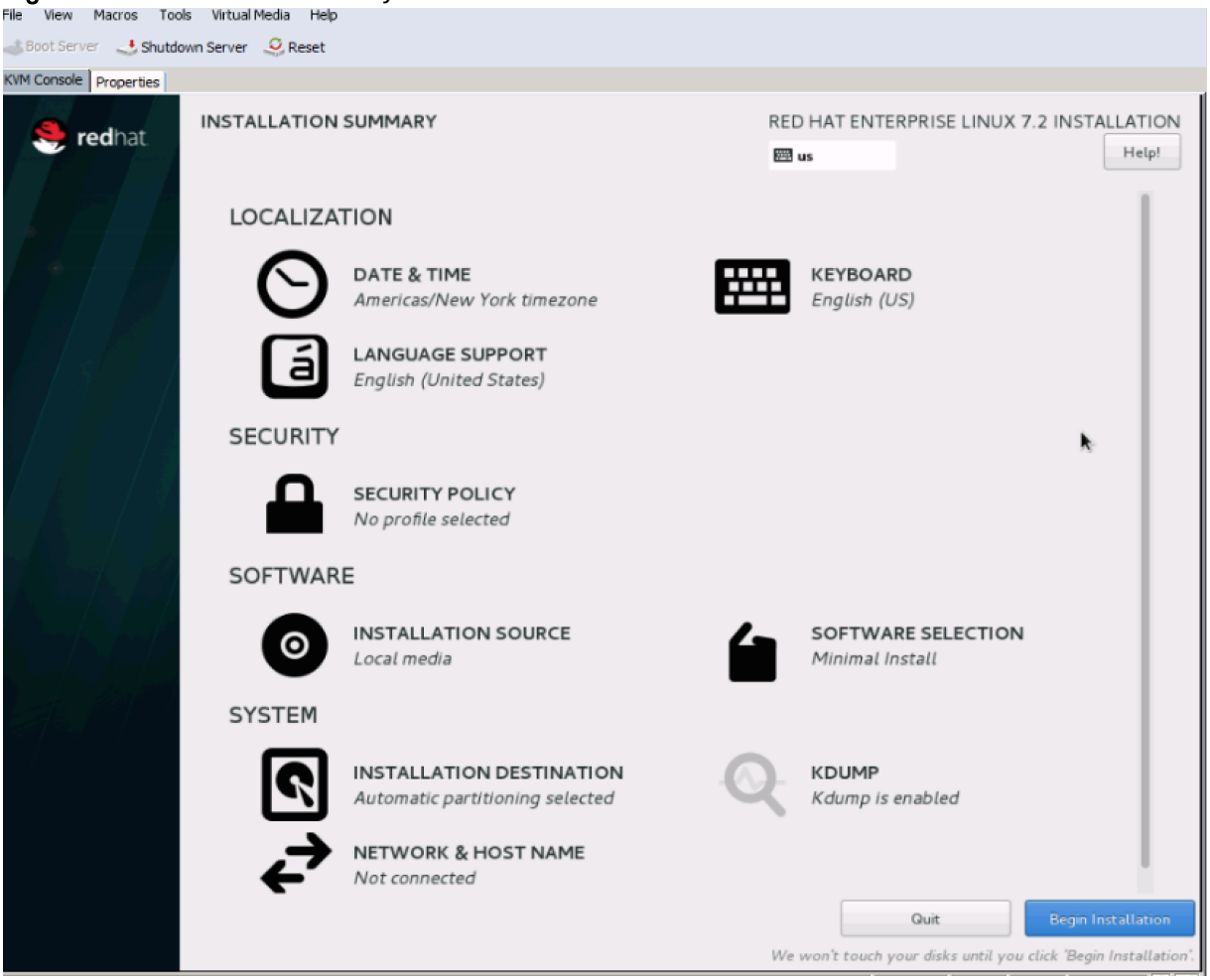
18. Select Date and time as shown in Figure 88.

Figure 88 Date and Time Window



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

Figure 89 Installation Summary Window



20. Click on Installation Destination, shown above in Figure 89.

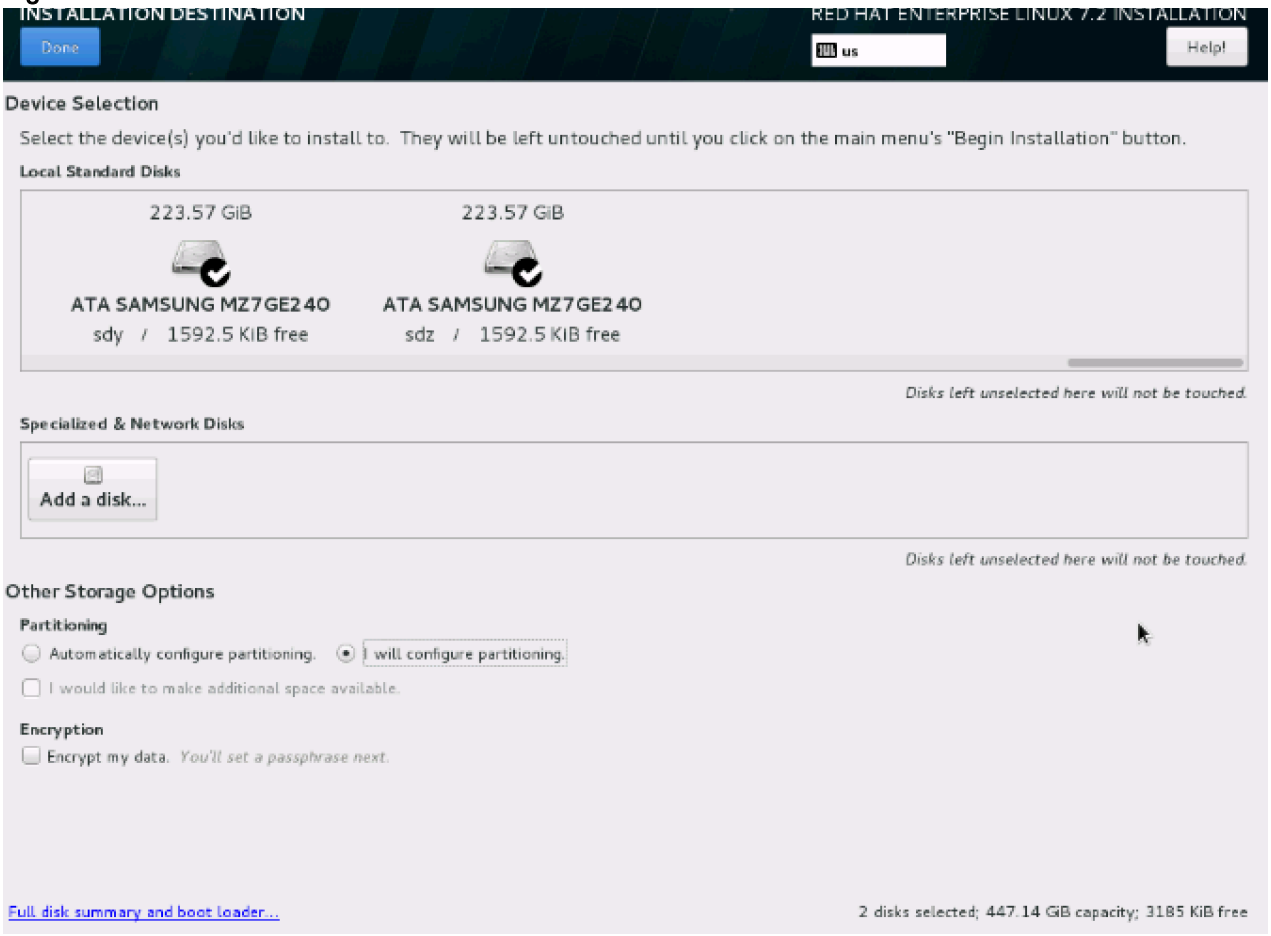
Figure 90 Installation Summary Window

A Caution symbol appears next to Installation Destination as shown in Figure 90 above.

21. This opens the Installation Destination window displaying the boot disks. This is shown in Figure 91 below.

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

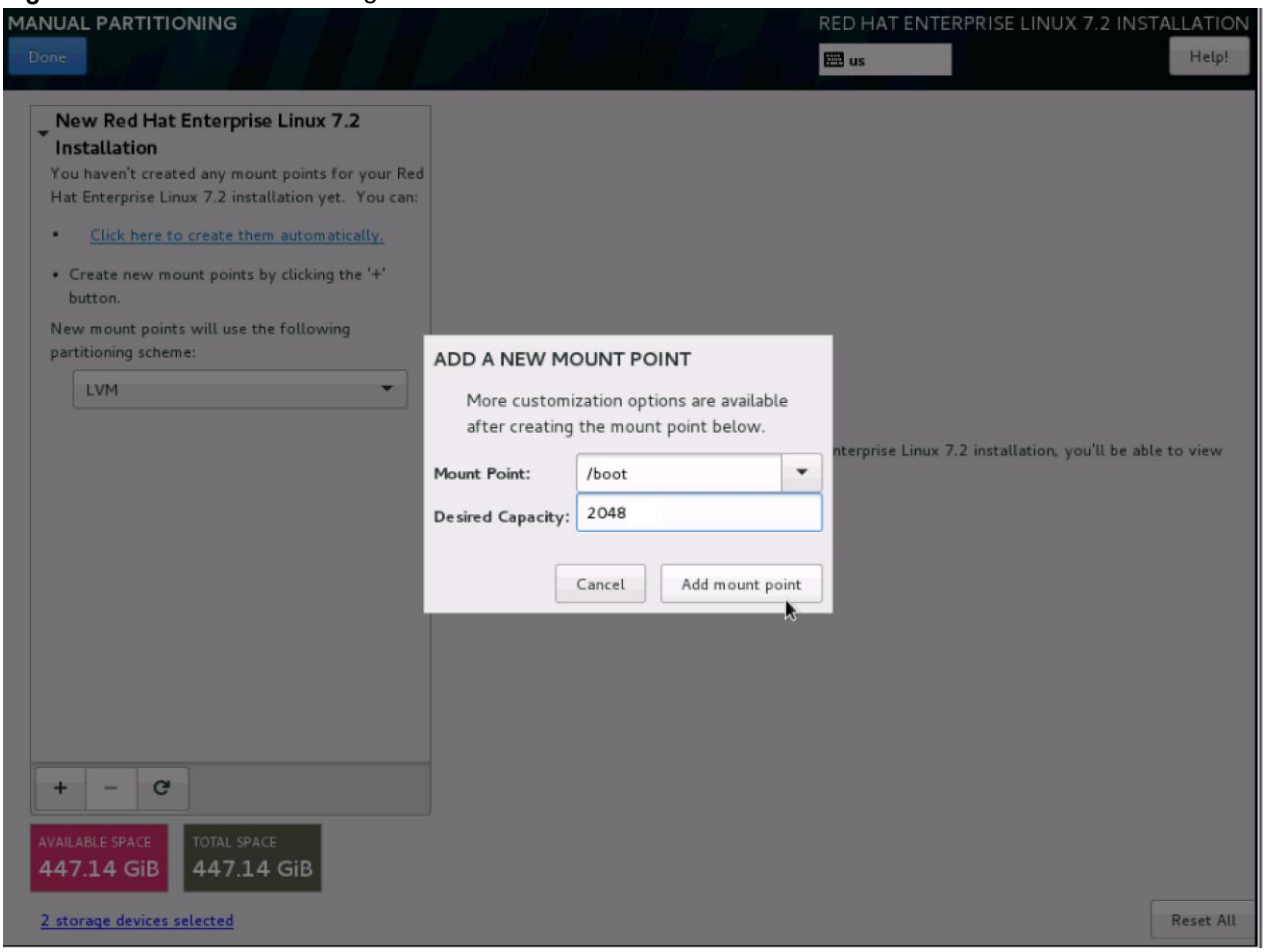
Figure 91 Installation Destination Window



This opens the new window for creating the partitions, as shown in Figure 92.

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

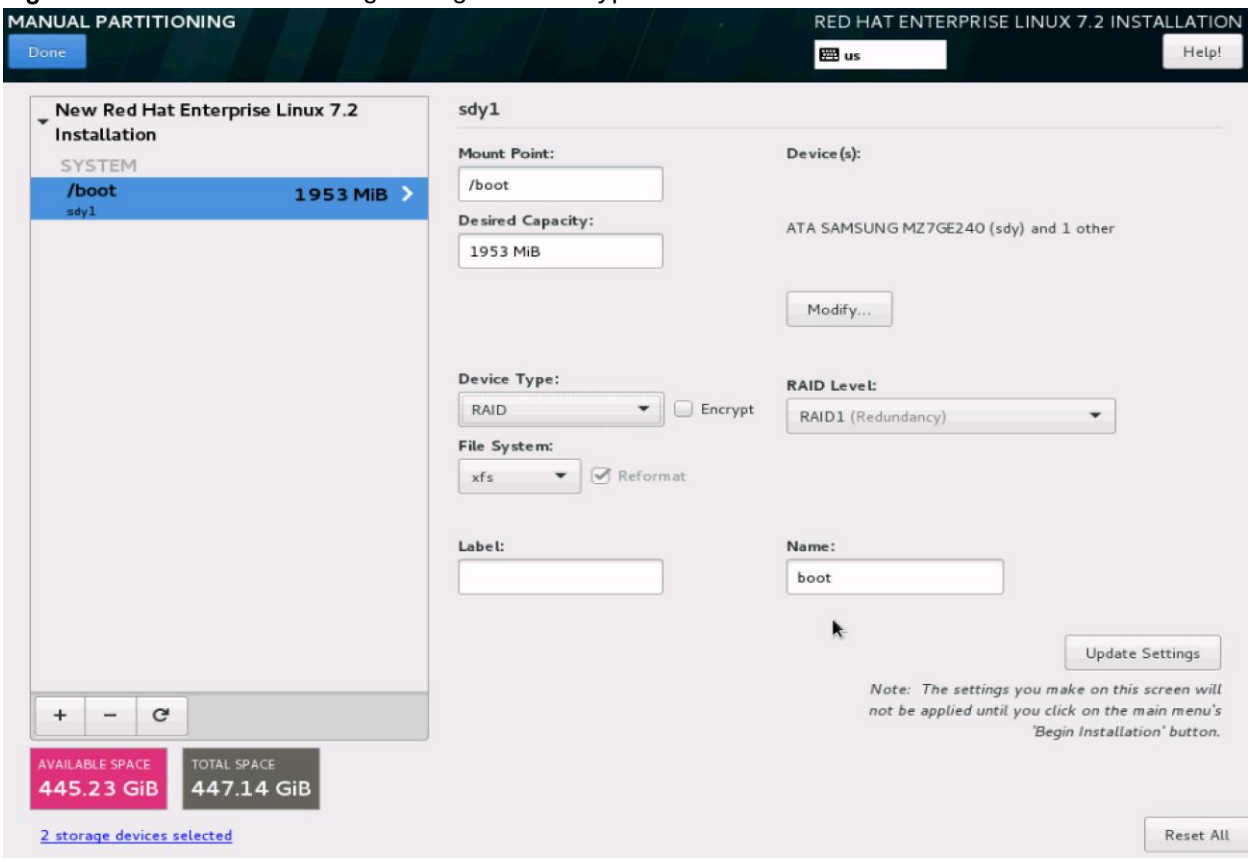
Figure 92 Manual Partitioning



24. Click Add Mount Point to add the partition.

The screen refreshes to show the added Mount Point (Figure 93).

Figure 93 Manual Partitioning/Change Device Type

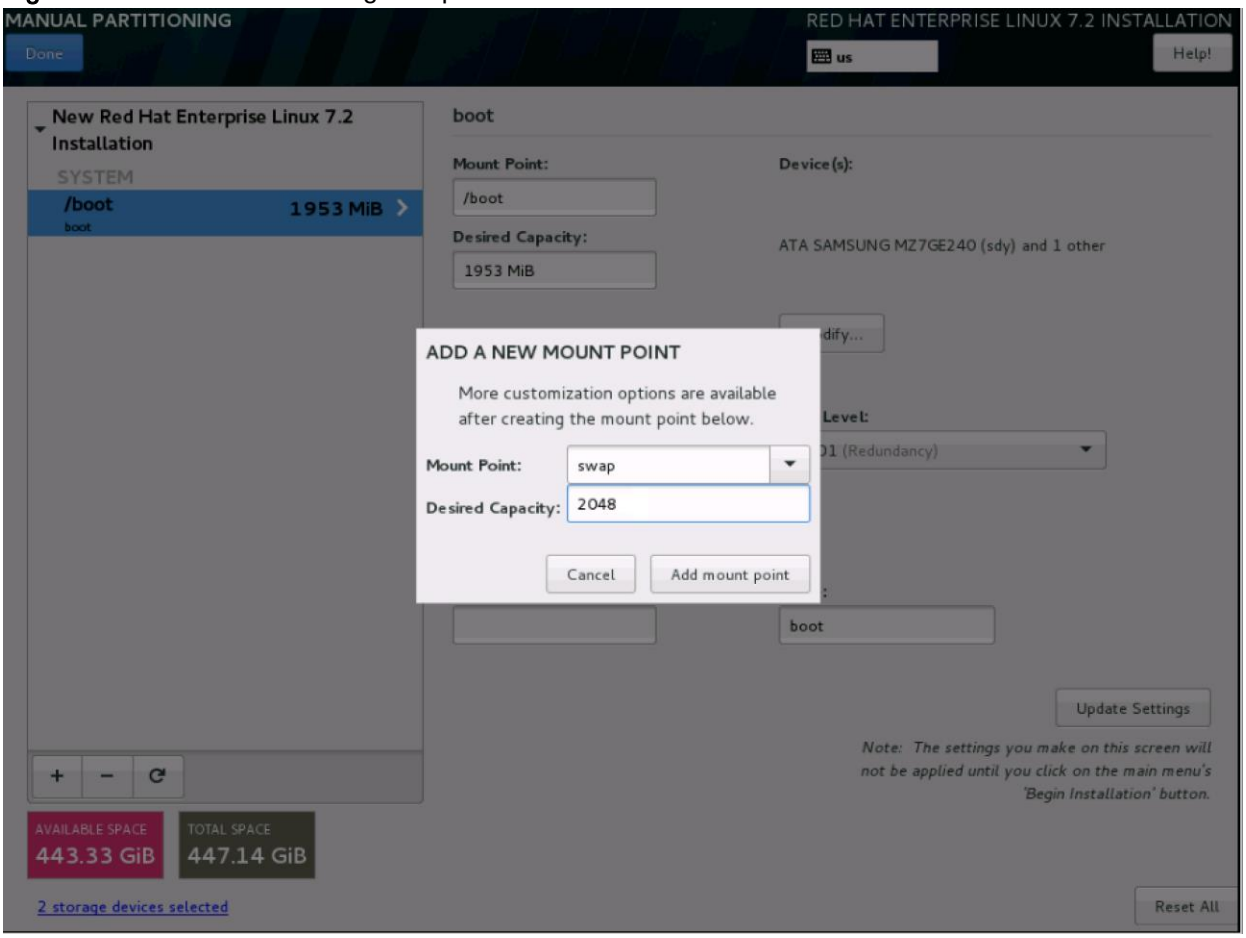


25. Change the Device type to RAID and make sure the RAID Level is RAID1 (Redundancy).

26. Click on Update Settings to save the changes.

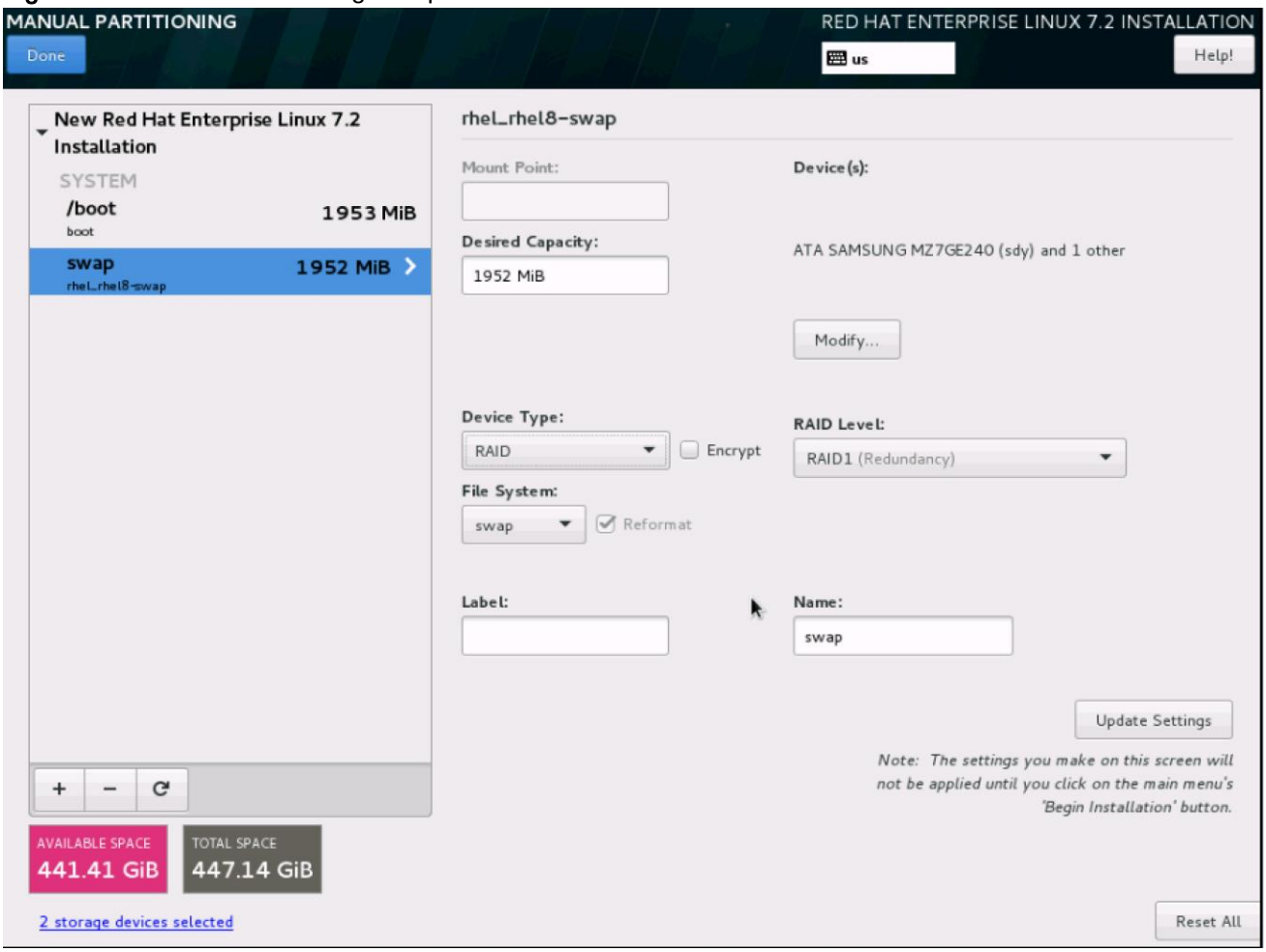
27. Click on the + sign to create the swap partition of size 2048 MB as shown in Figure 94 below.

Figure 94 Manual Partitioning/Swap



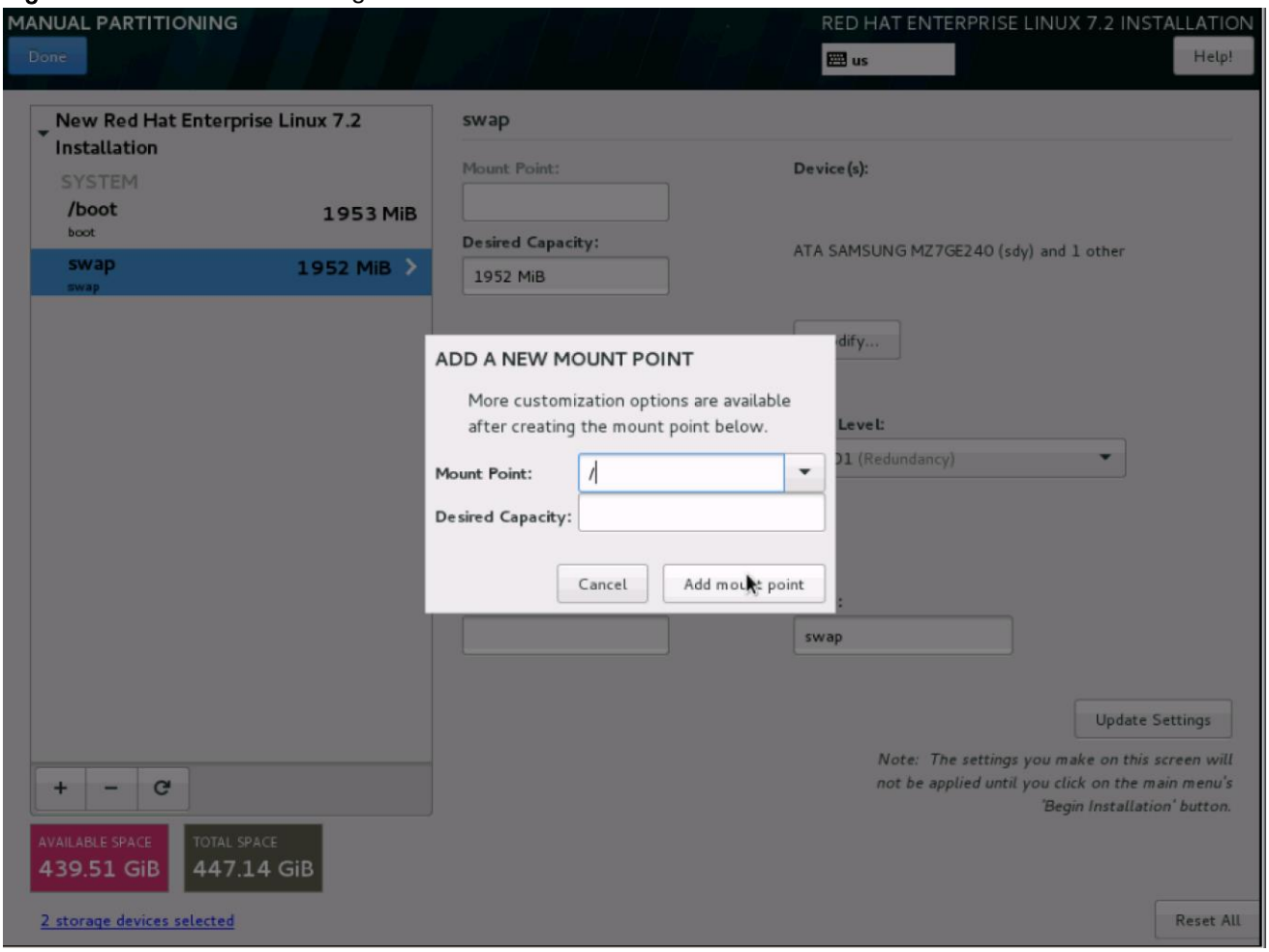
28. Change the Device type to RAID and RAID level to RAID1 (Redundancy) and click on Update Settings.

Figure 95 Manual Partitioning/Swap



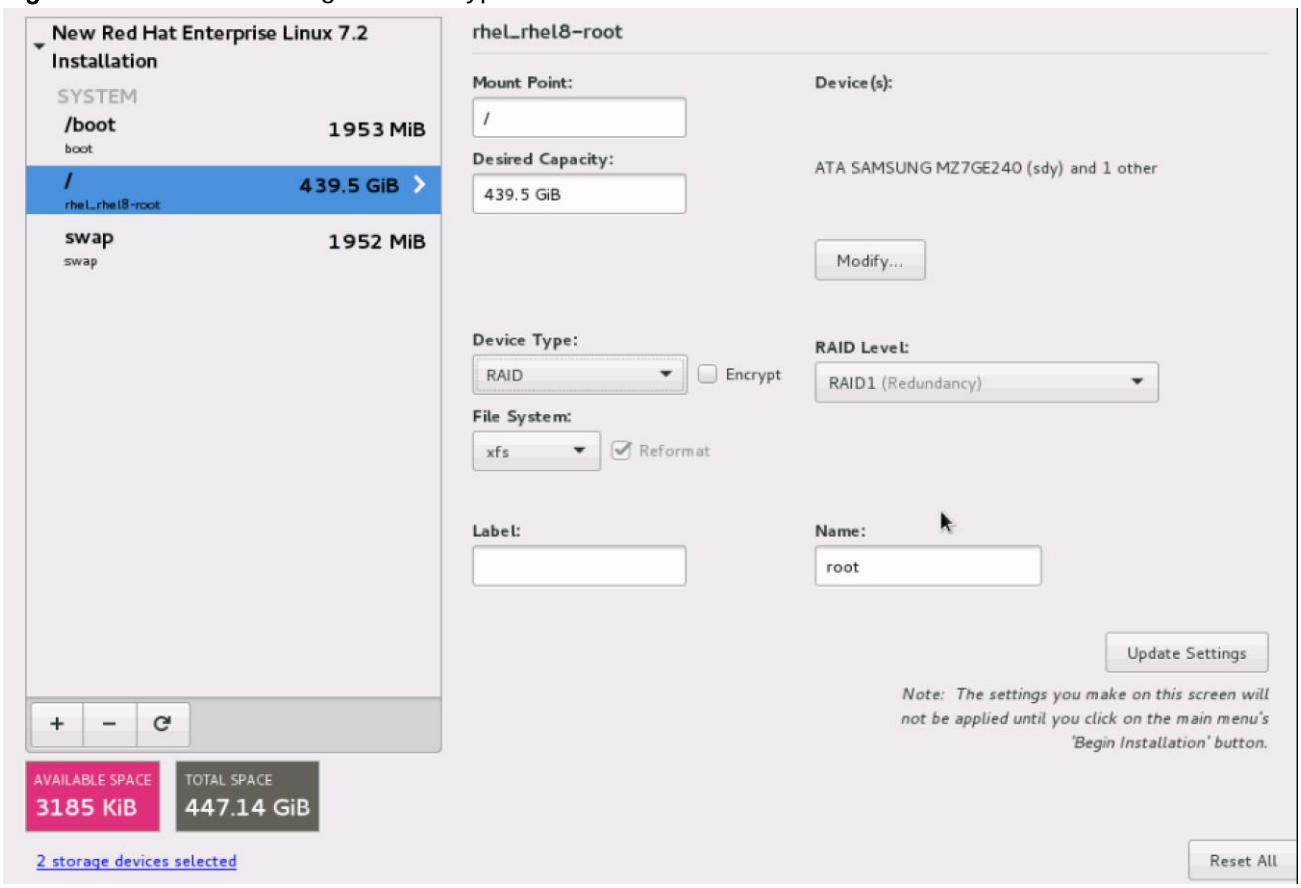
29. Click + to add the / partition. The size can be left empty so it uses the remaining capacity and click Add Mountpoint. (Figure 96).

Figure 96 Manual Partitioning/Add A New Mount Point



30. In the next window (Figure 97), change the Device type to RAID and RAID level to RAID1 (Redundancy). Click Update Settings.

Figure 97 Partition/Change Device Type to RAID

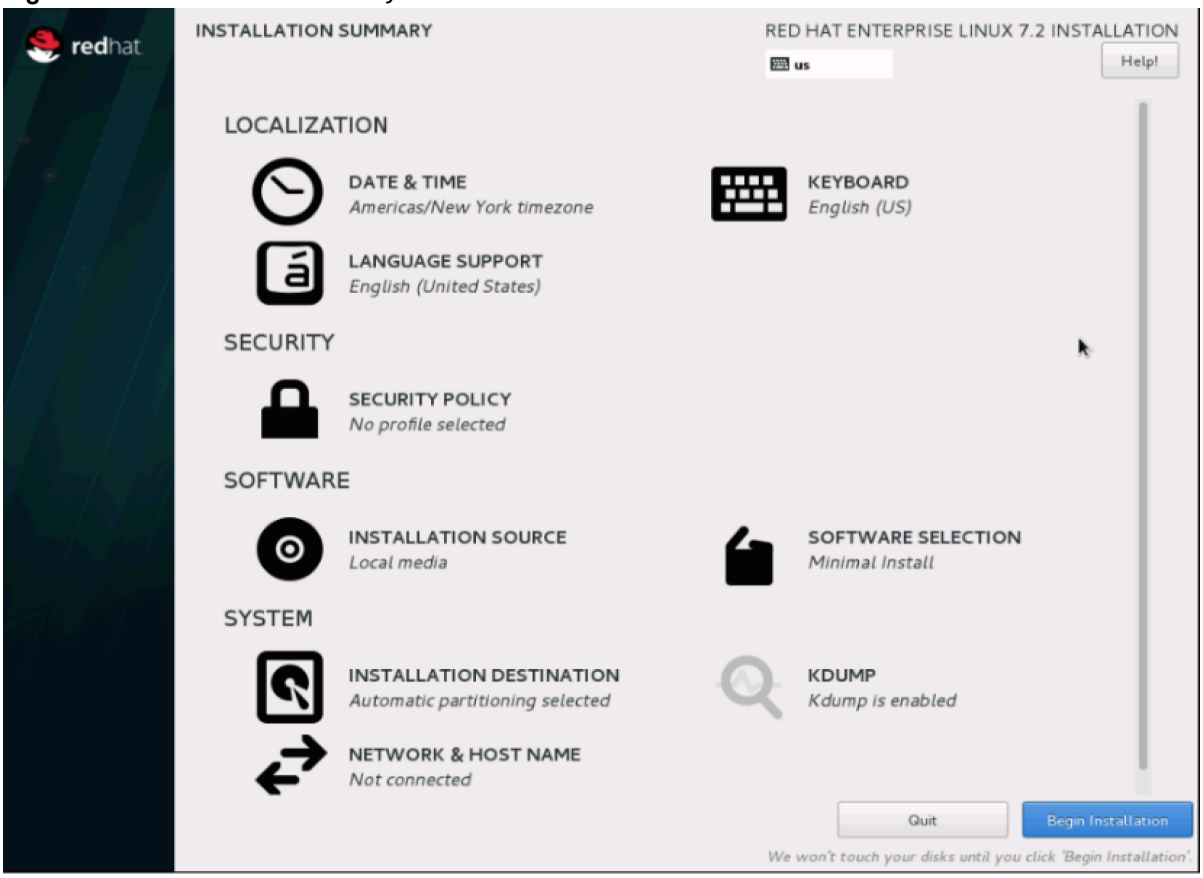


31. Click Done to go back to the main screen and continue the Installation.

The Installation screen opens (Figure 98).

32. Click on Software Selection.

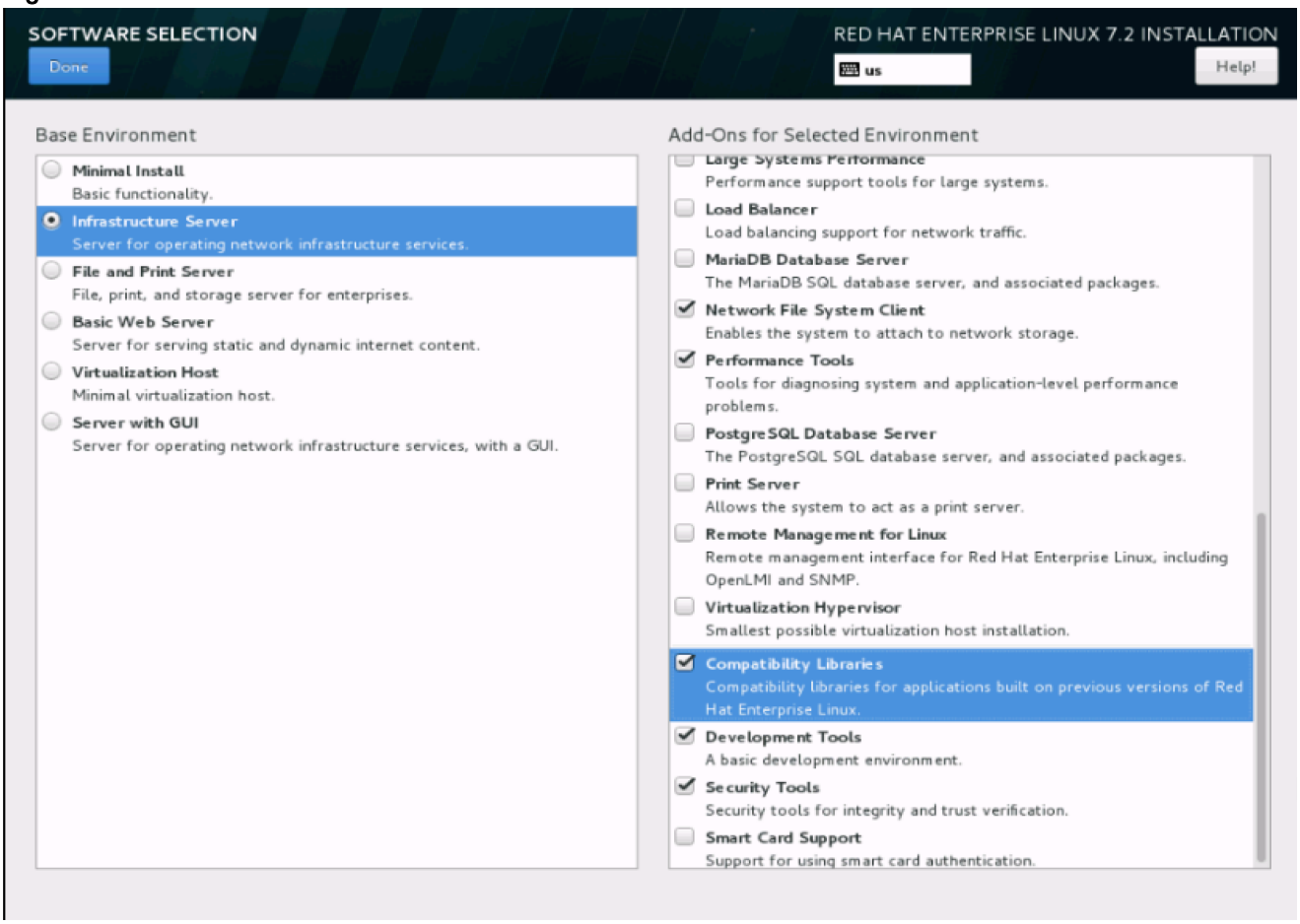
Figure 98 Installation Summary Window



The Software Selection screen opens (Figure 99).

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

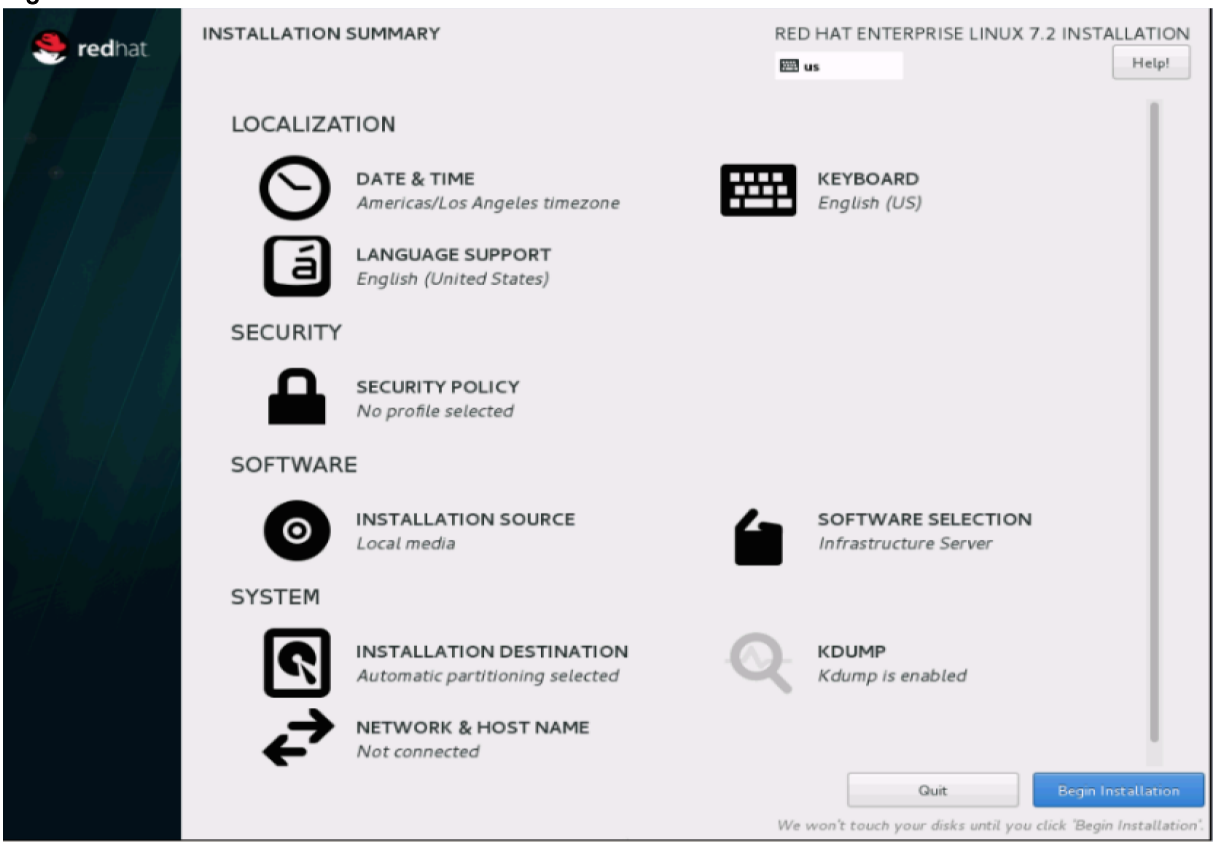
Figure 99 Software Selection



The Installation Summary window returns (Figure 100).

34. Click on Network and Hostname.

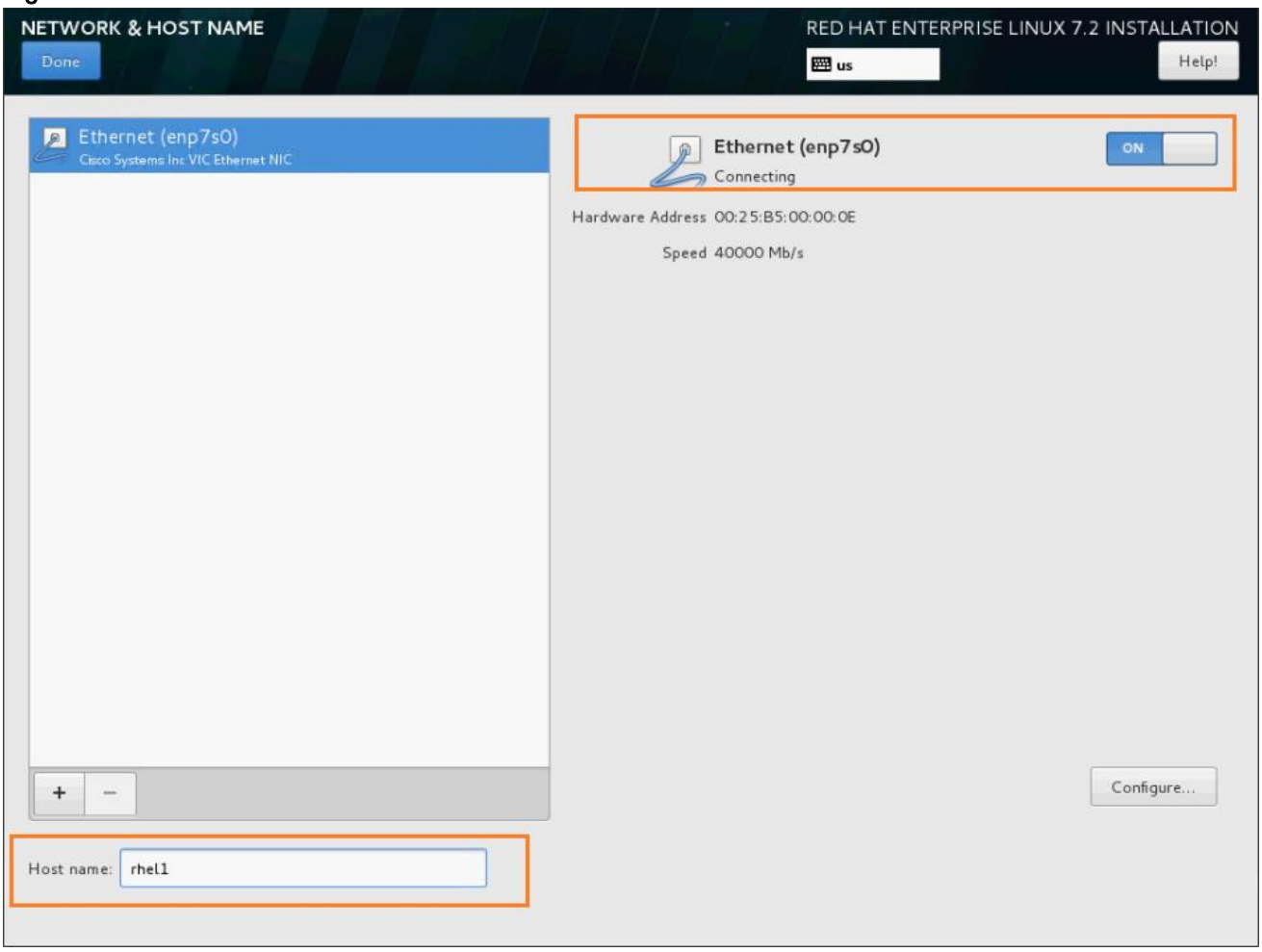
Figure 100 Network and Host Name Window



Configure Hostname and Networking for the Host (Figure 101).

35. Type in the hostname as shown below.

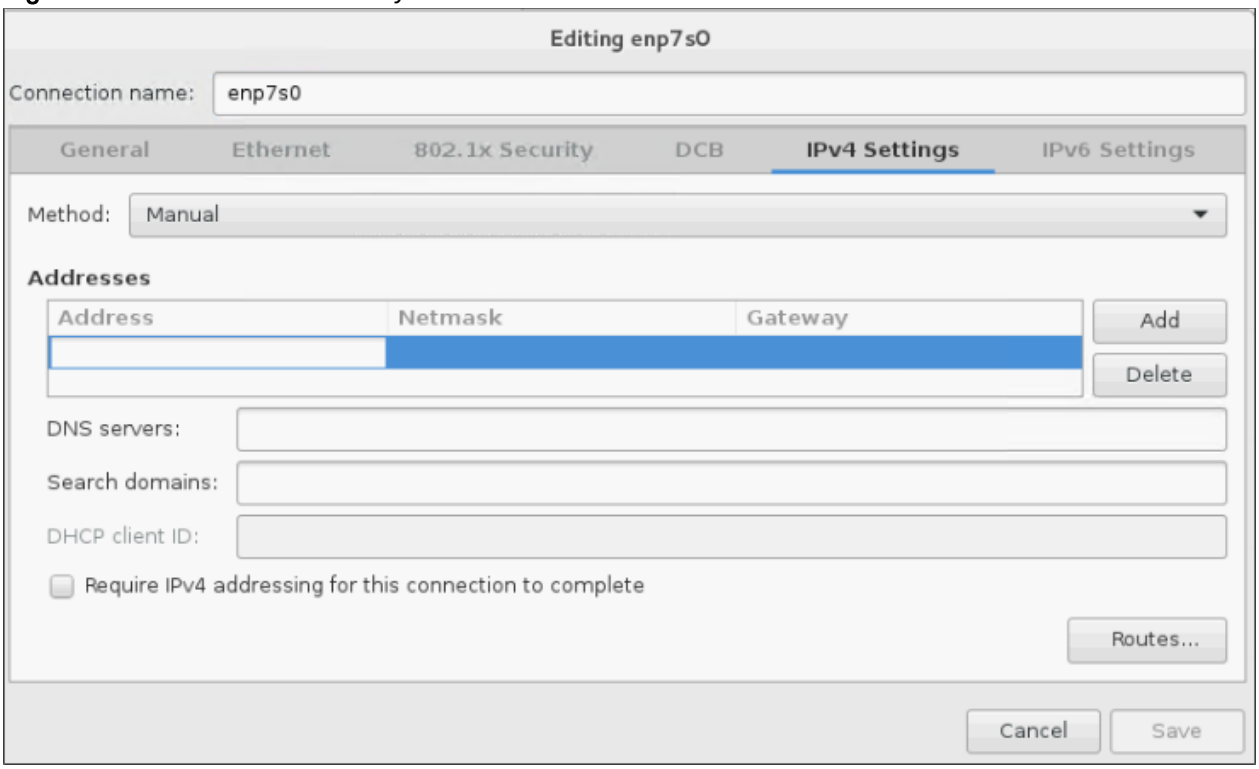
Figure 101 Network and Host Name



36. Click on Configure to open the Network Connectivity window (Figure 102).

37. Click on IPV4Settings.

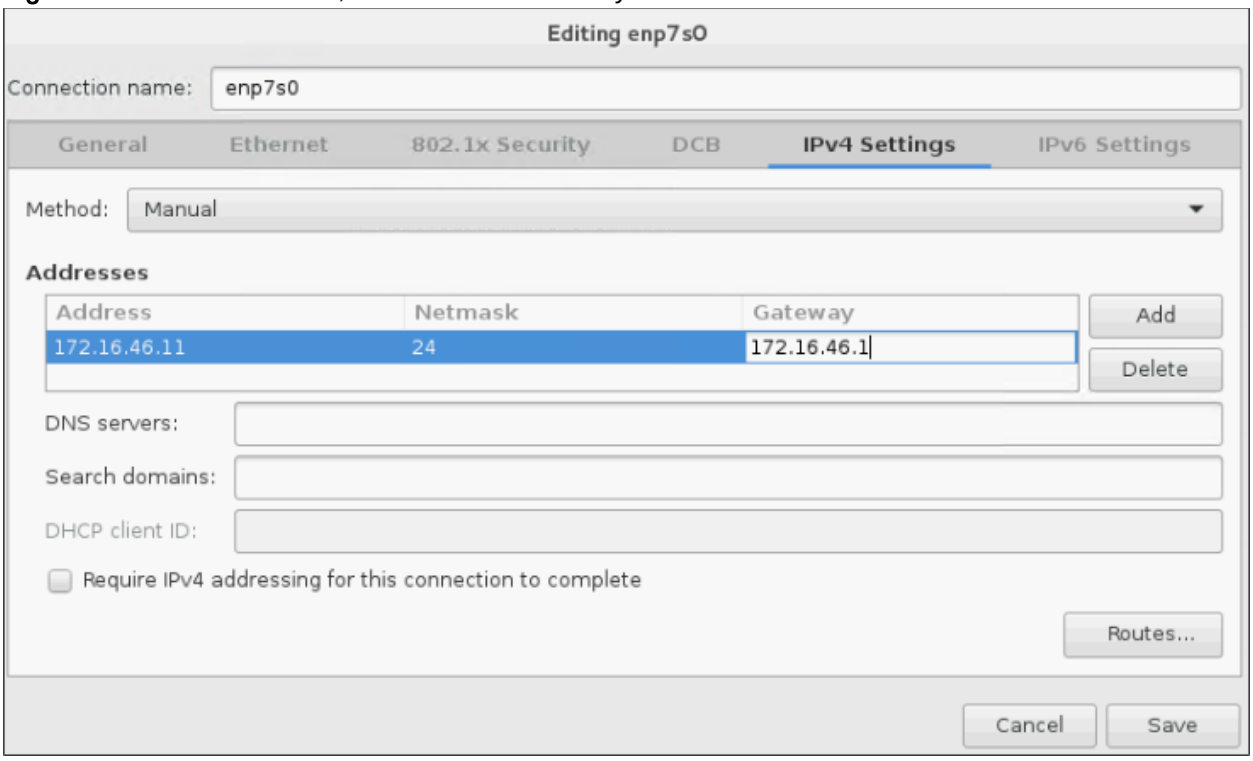
Figure 102 Network Connectivity Window



38. Change the Method to Manual and click Add. Figure 103 shows the Add Details pop up window.

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

Figure 103 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 104).

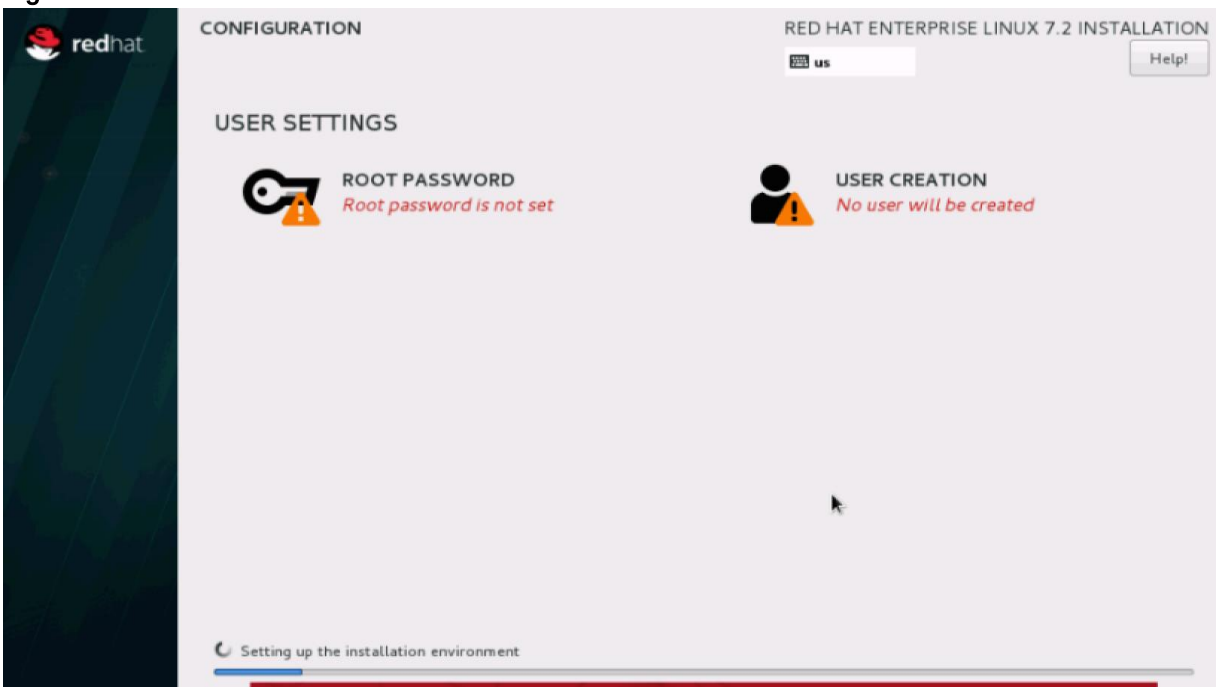
42. Click Begin Installation in the main menu.

Figure 104 Installation Summary Window

A new window opens (Figure 105).

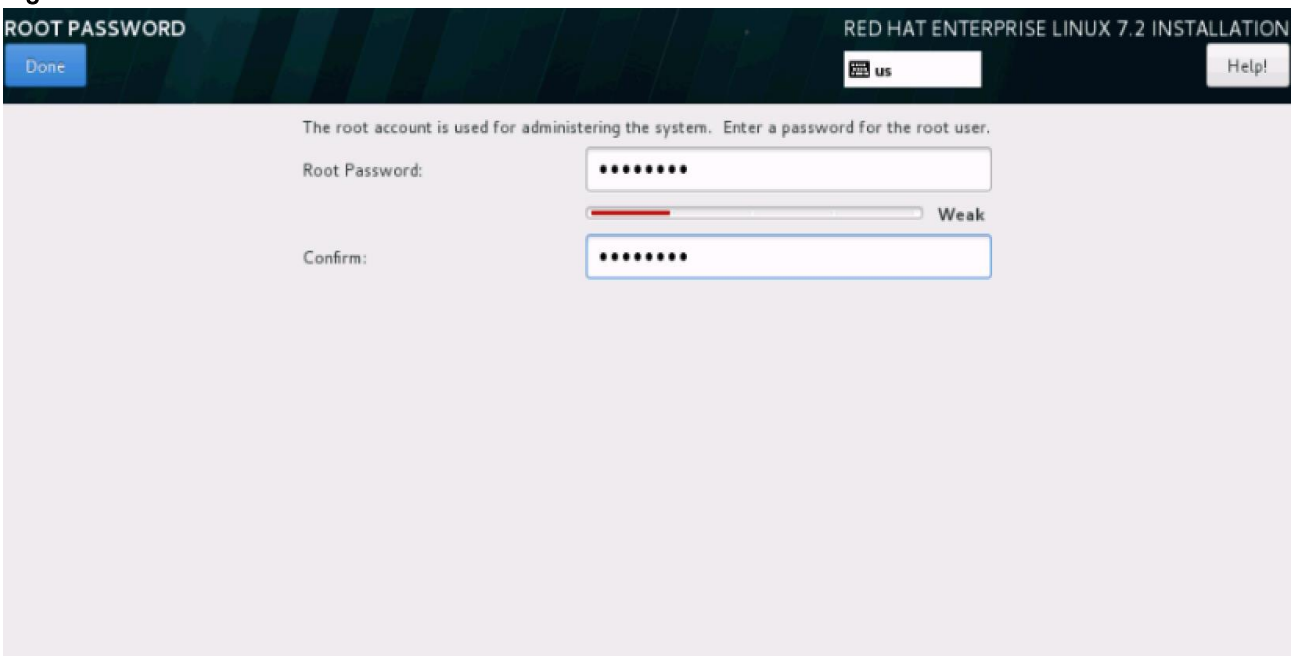
43. Select `Root Password` in the User Settings.

Figure 105 Select Root Password



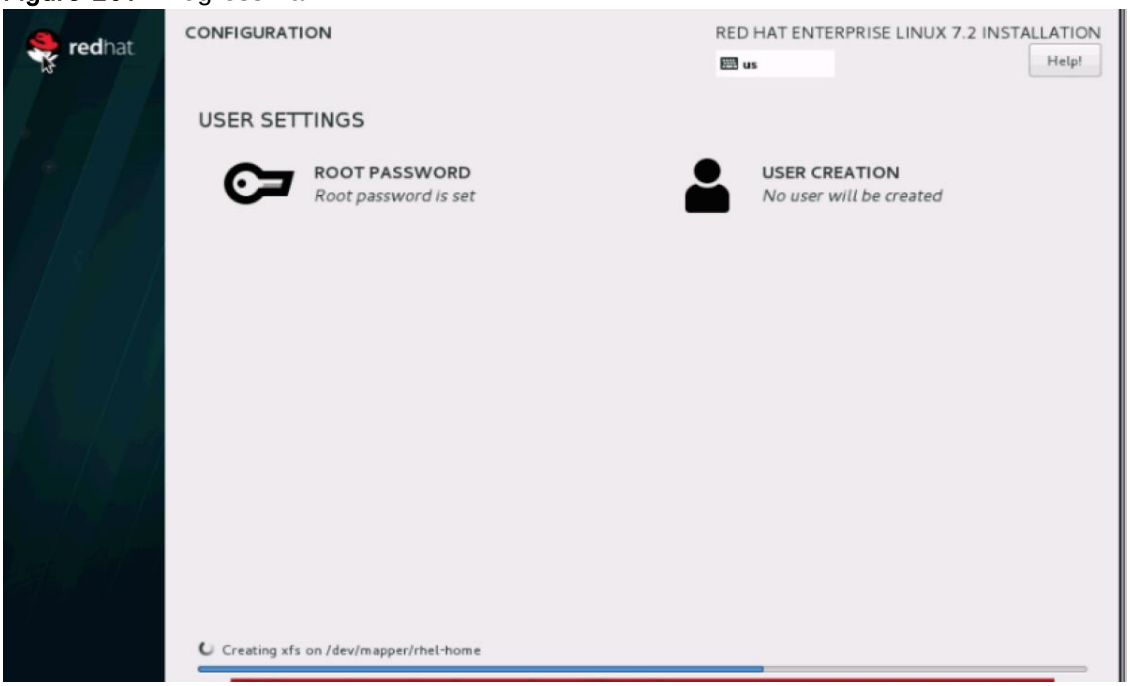
44. On the next screen (Figure 106), enter the `Root Password` and click `done`.

Figure 106 Enter the Root Password



A progress window will open (Figure 107).

Figure 107 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

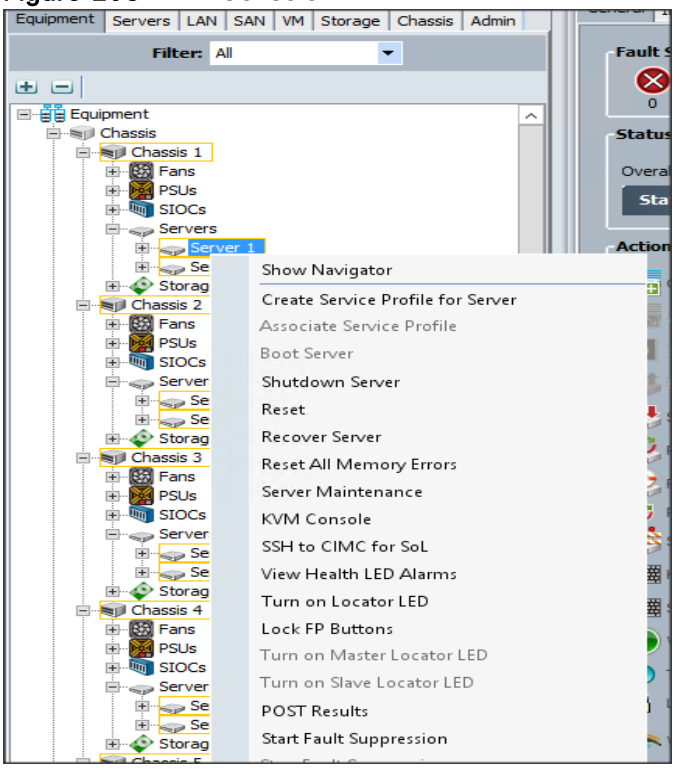
The following section provides detailed procedures for installing Red Hat Enterprise Linux 7.2 on Cisco UCS S3260 Storage Servers. There are multiple ways to install the Red Hat Linux operating system. The installation procedure described in this deployment guide uses KVM console and virtual media from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

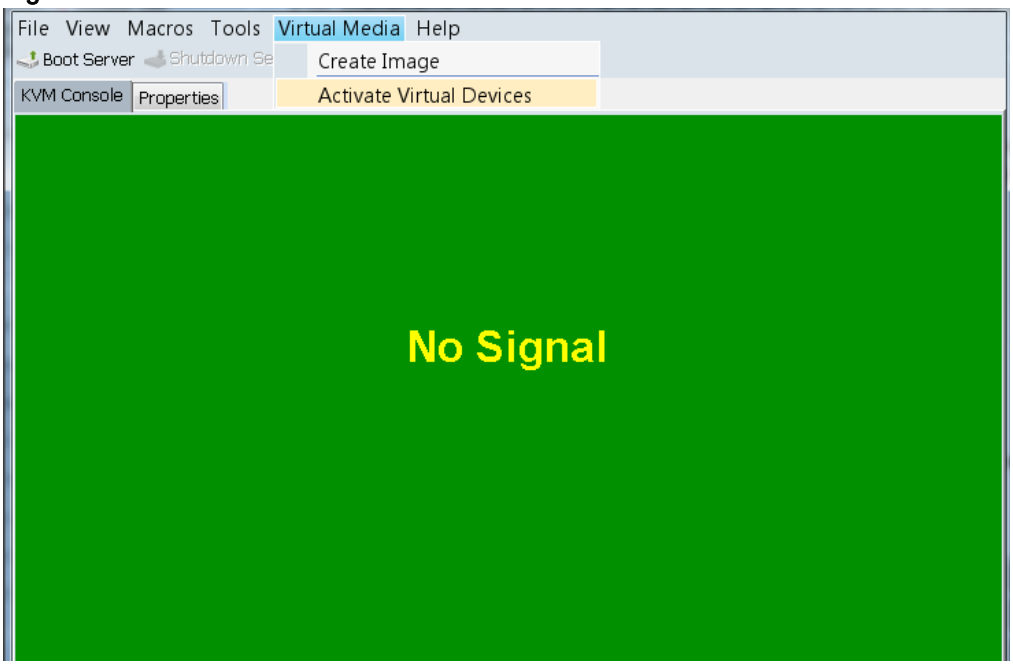
1. Log in to the Cisco UCS 6332 Fabric Interconnect and launch the Cisco UCS Manager application.
2. Select the `Equipment` tab.
3. In the navigation pane expand `Chassis` and then `Servers`.
4. Right click on the server and select `KVM Console`. (Figure 108)
5. In the KVM window, select the `Virtual Media` tab.

Figure 108 KVM Console



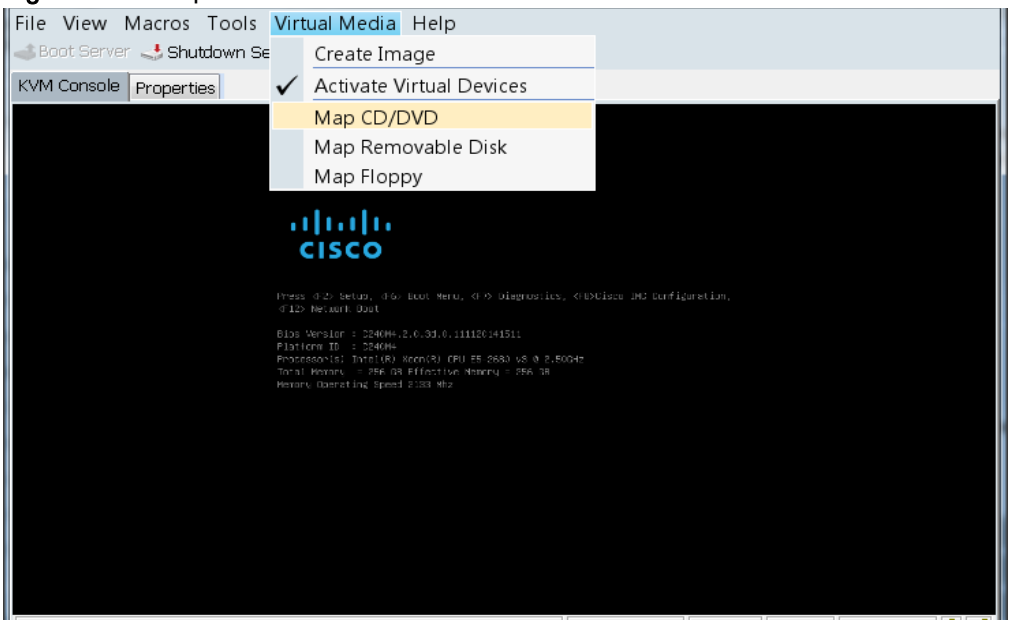
6. Click the `Activate Virtual Devices` found in the Virtual Media tab. (Figure 109)

Figure 109 Activate Virutal Devices



7. In the KVM window, select the Virtual Media tab and click the `Map CD/DVD`. (Figure 110)

Figure 110 Map CD/DVD

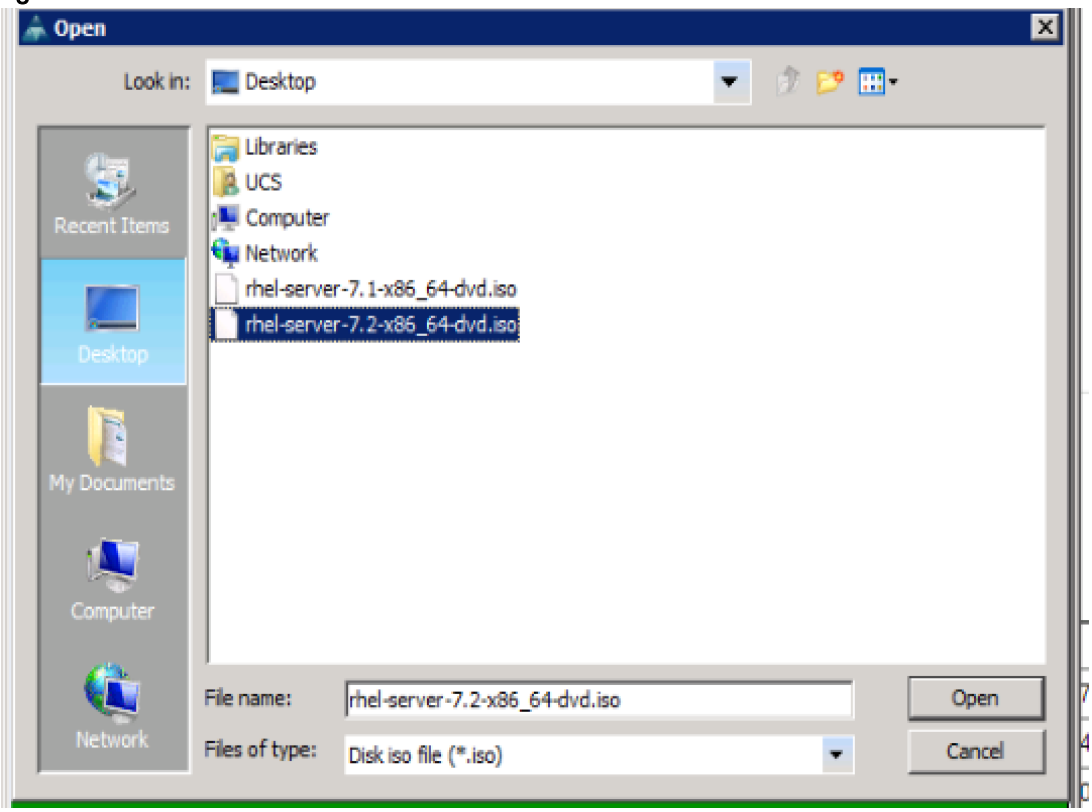


8. Browse to the Red Hat Enterprise Linux Server 7.2 installer ISO image file.

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

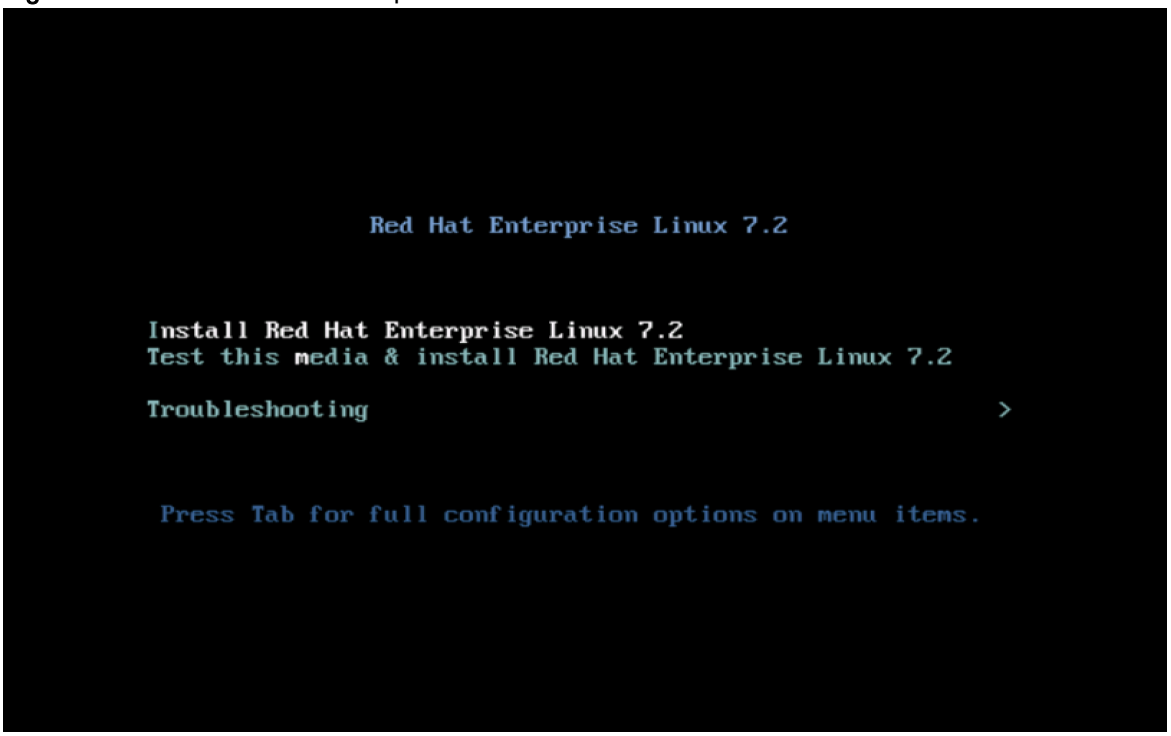
9. Click Open to add the image to the list of virtual media. (Figure 111)

Figure 111 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. Select the Install or Upgrade an Existing System.

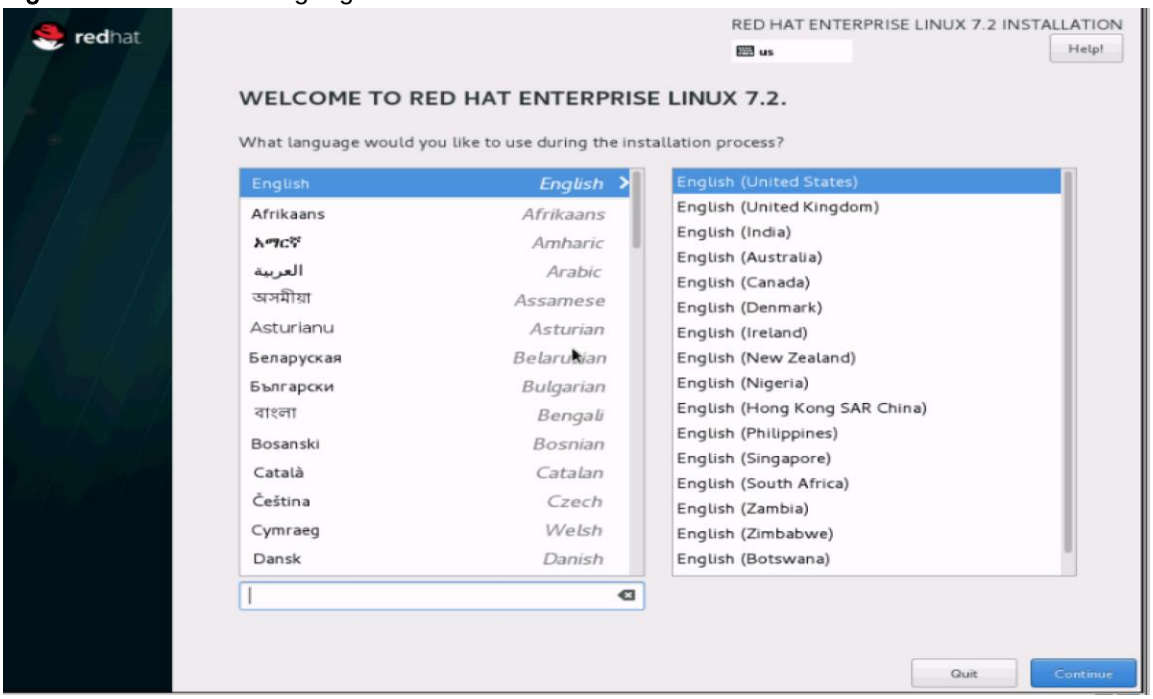
Figure 112 Install Red Hat Enterprise Linux 7.2



16. Skip the Media test and start the installation. (Figure 112)

17. Select language of installation and click Continue. (Figure 113)

Figure 113 Choose Language of Installation



18. Select Date and Time, (Figure 114) which pops up another window as shown below in Figure 115.

Figure 114 Date and Time

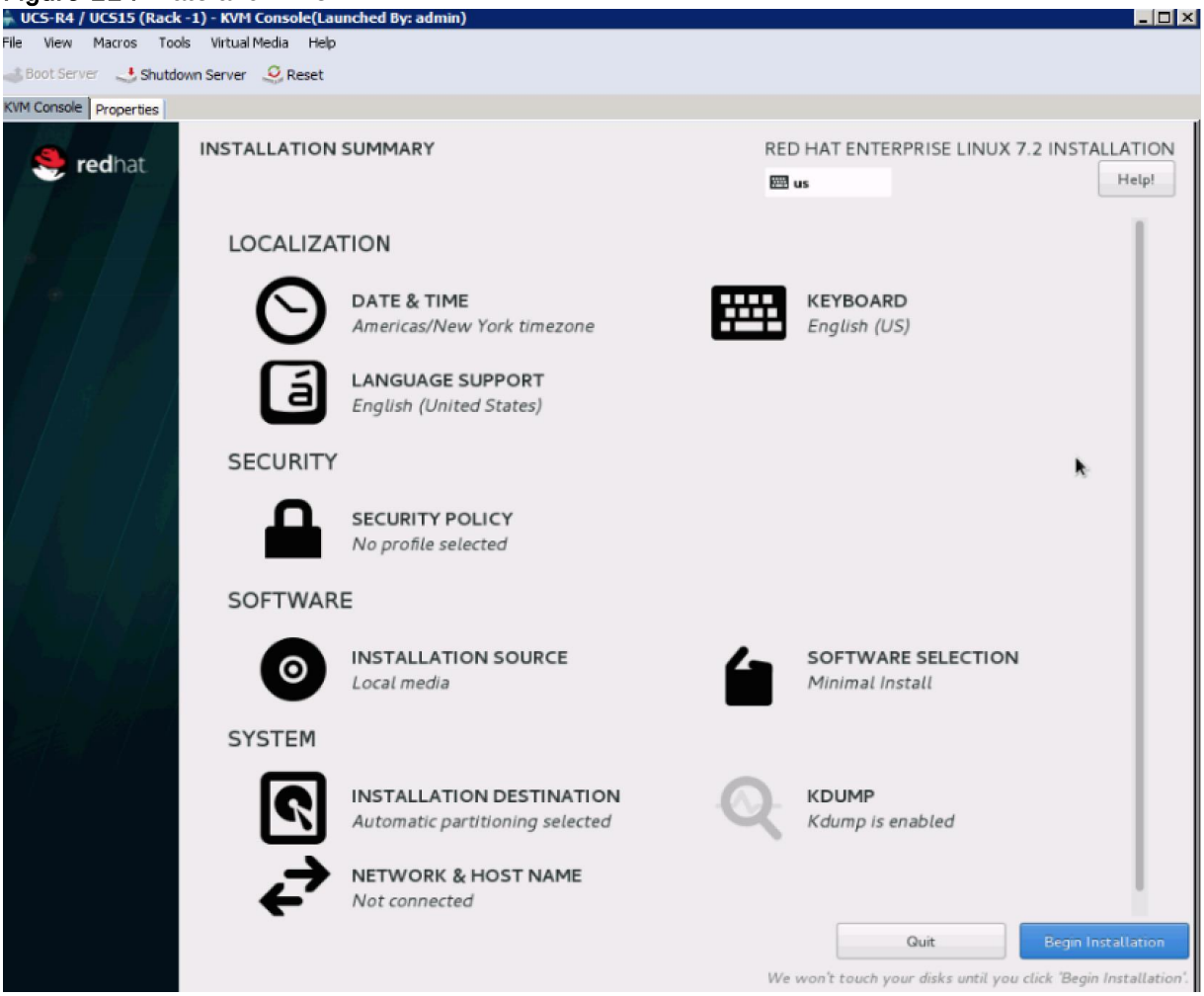
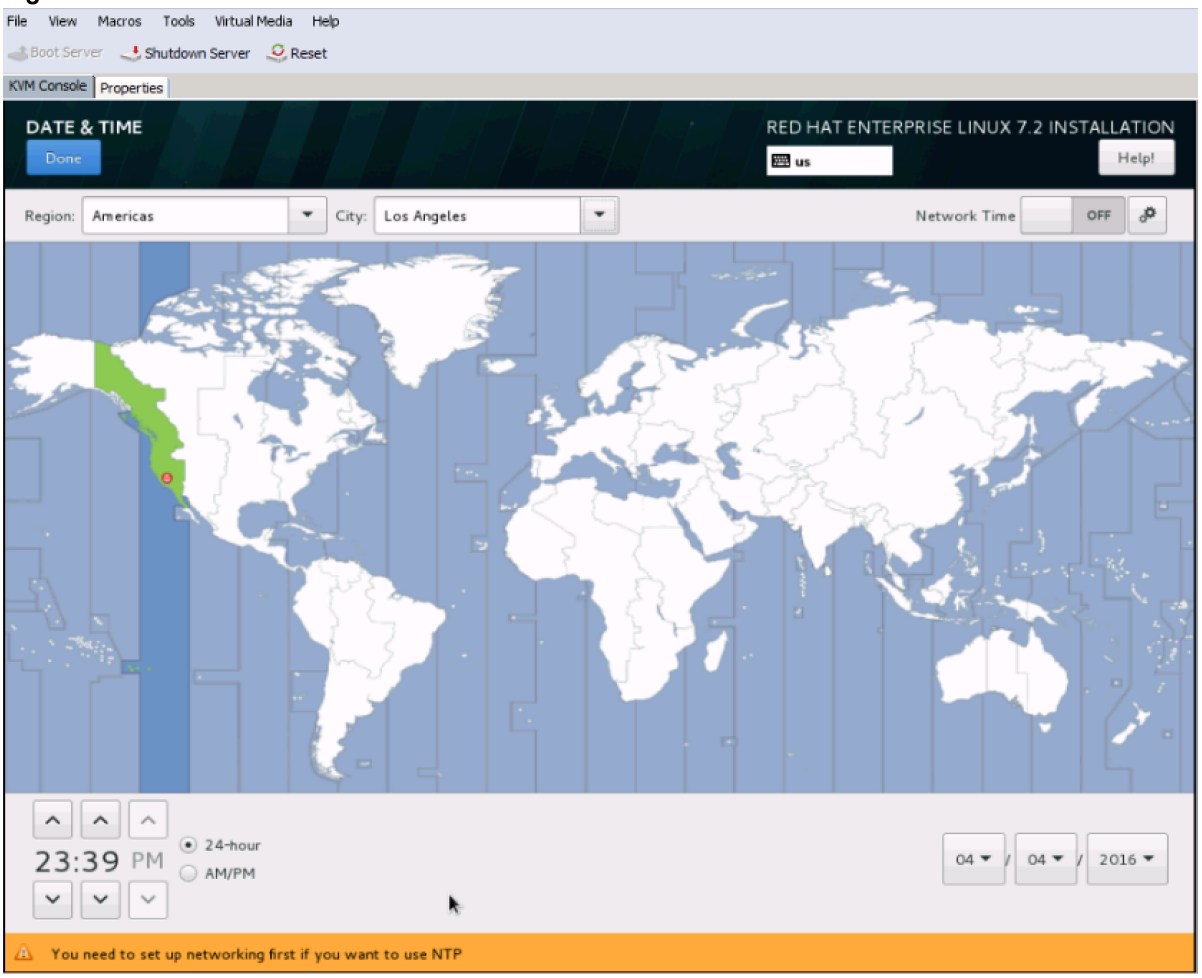


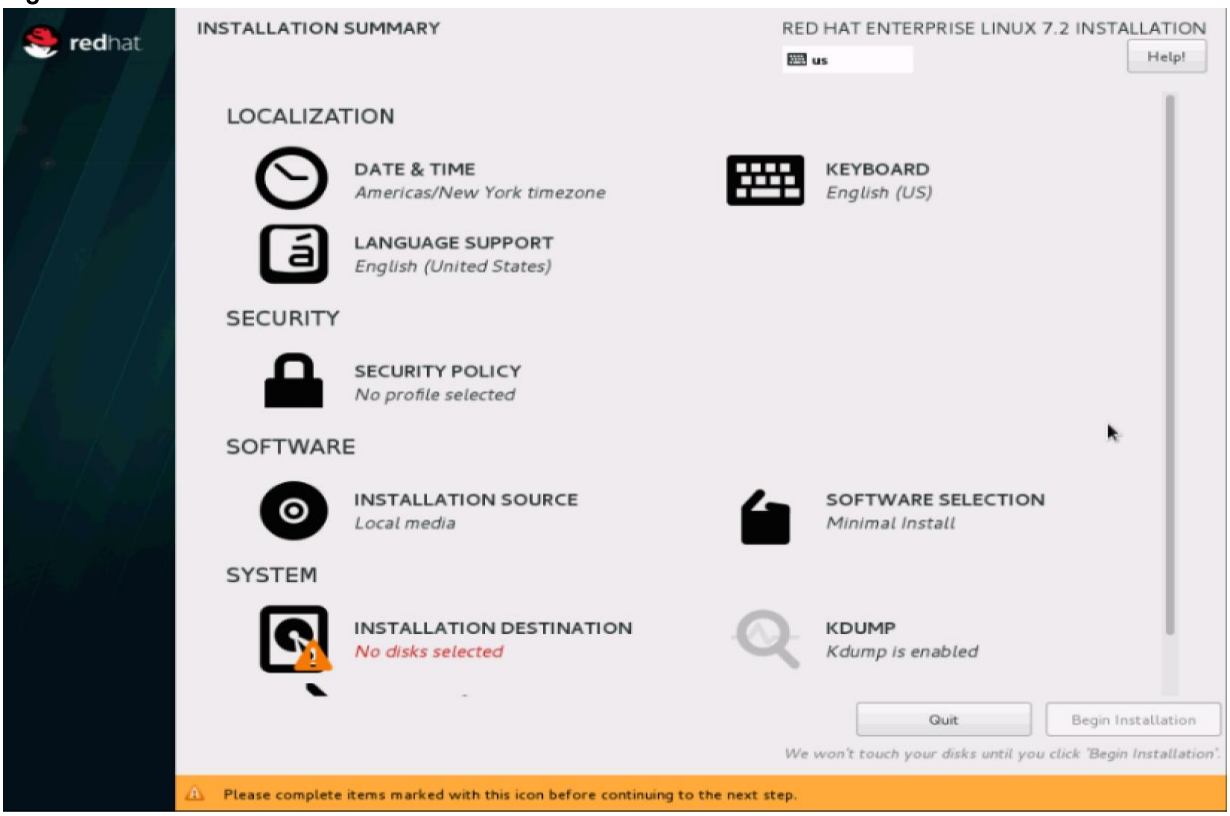
Figure 115 Choose Time Zone



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

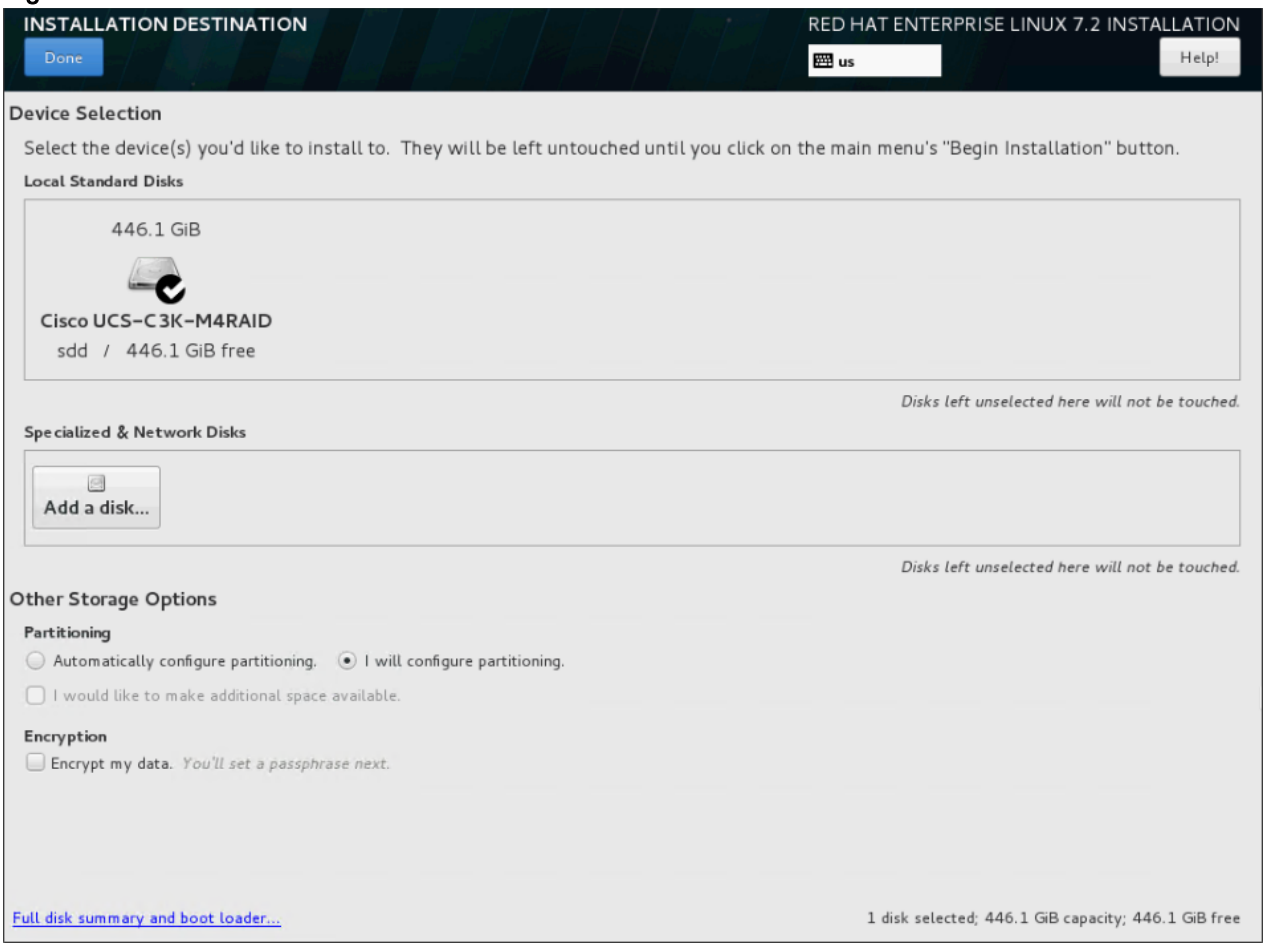
20. Click on Installation Destination. (Figure 116)

Figure 116 Installation Destination



21. This opens a new window with the boot disks. Make the selection, and choose I will configure partitioning. Click Done. (Figure 117)

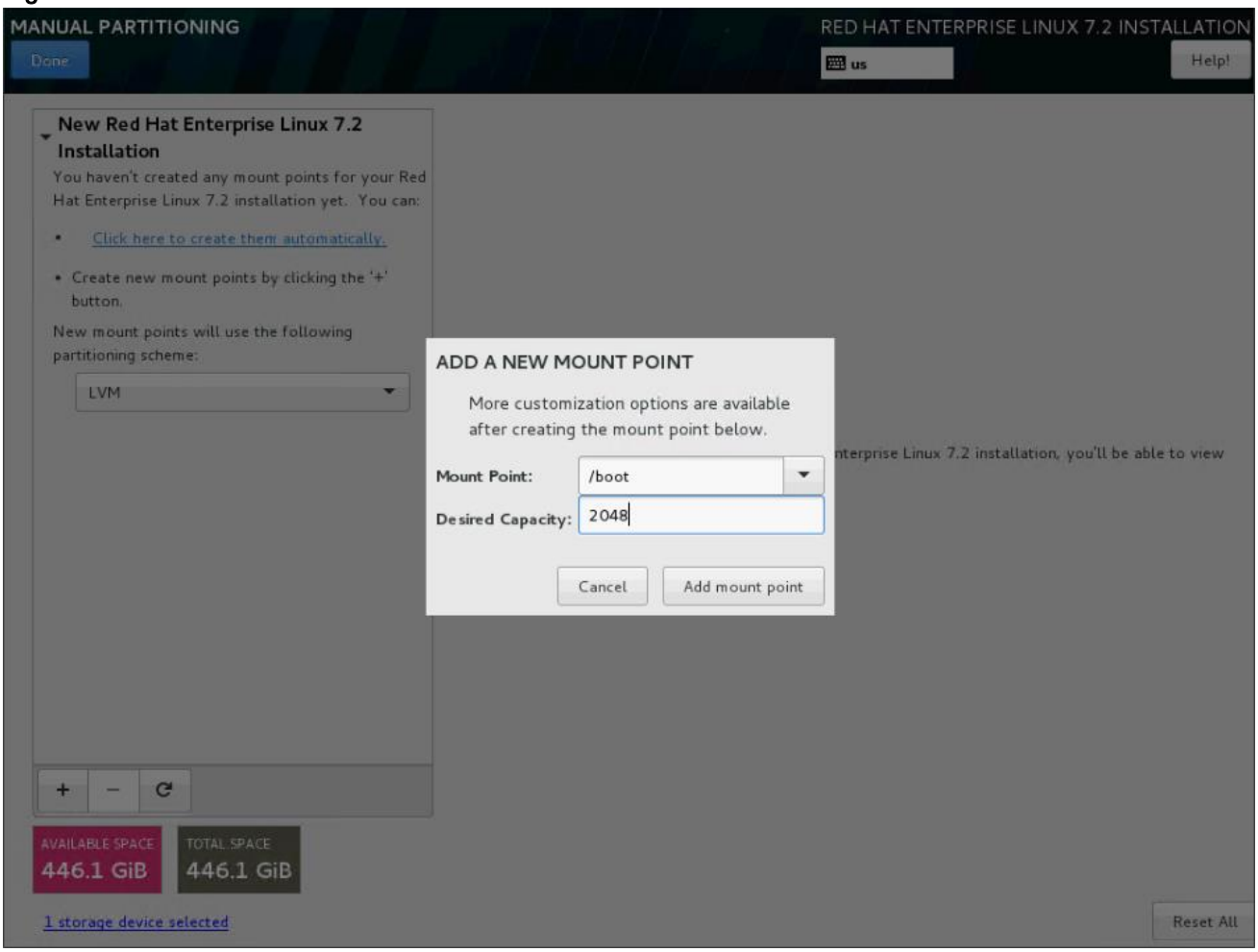
Figure 117 Installation Destination



22. This opens the new window for creating the partitions. (Figure 118) Click on the + sign to add a new partition as shown below, boot partition of size 2048 MB.

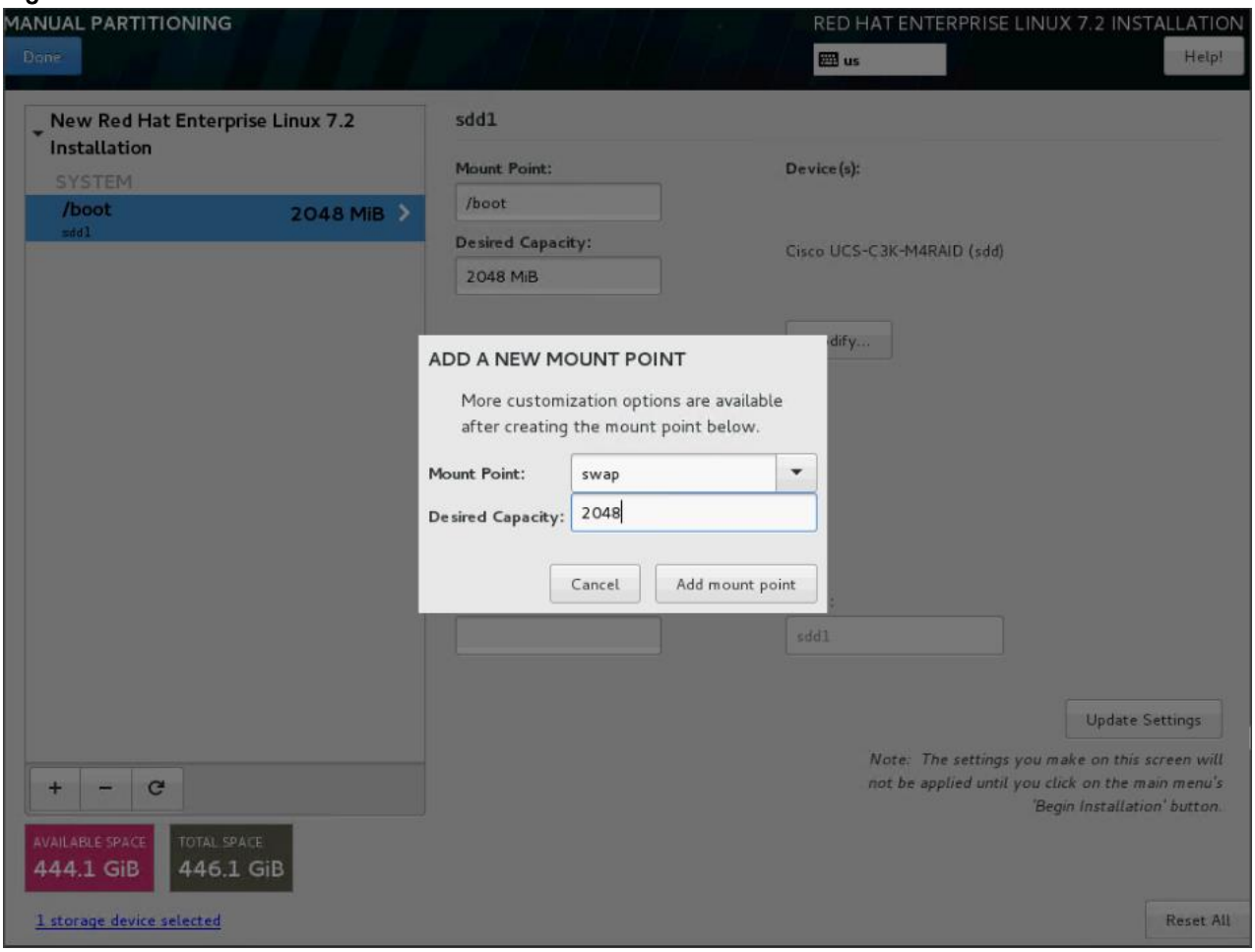
23. Click Add MountPoint to add the partition.

Figure 118 Add a New Mount Point



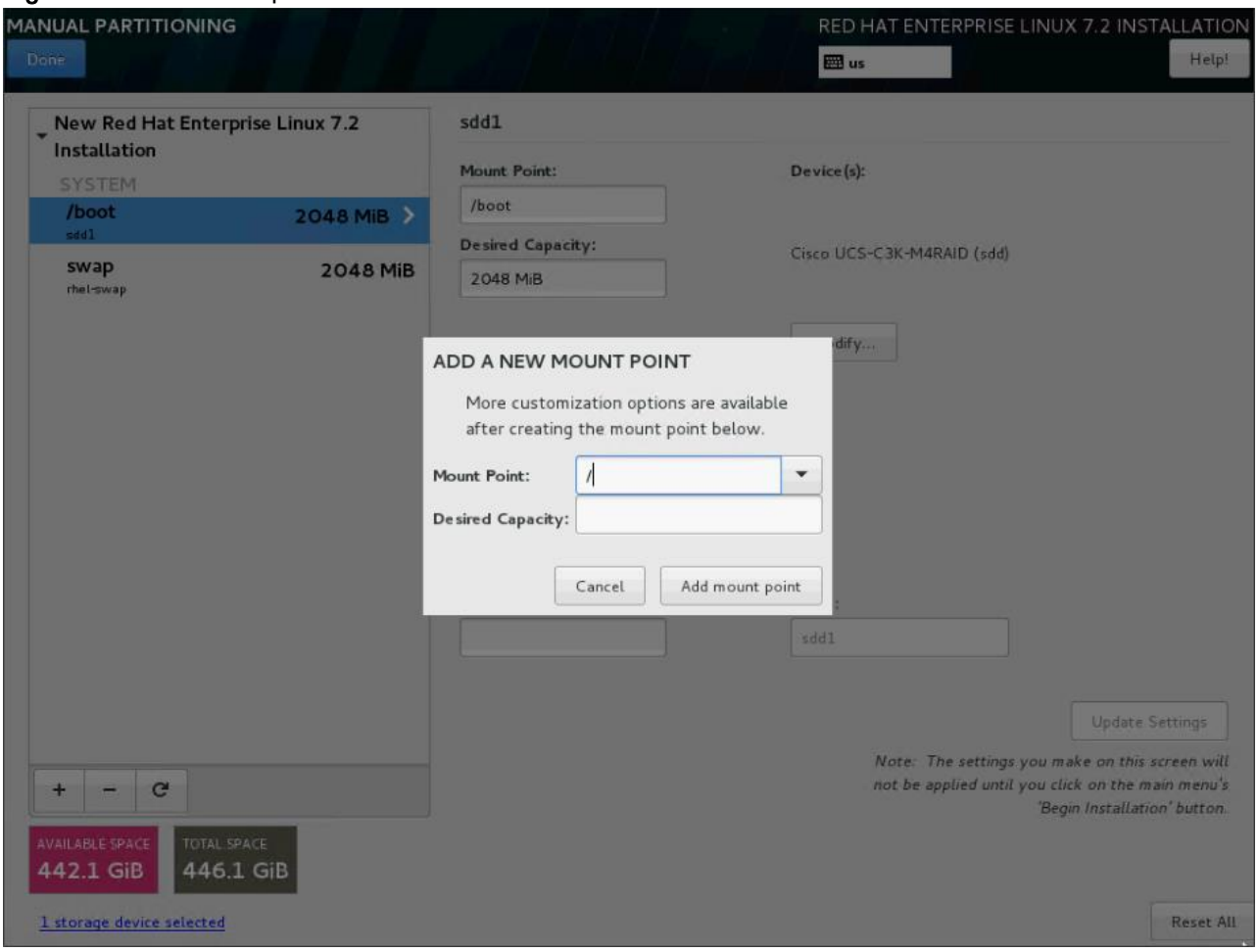
24. Click on the + sign to create the swap partition of size 2048 MB as shown below. (Figure 119)

Figure 119 Add a New Mount Point



25. Click + to add the / partition. The size can be left empty so it uses the remaining capacity and click Add Mountpoint. (Figure 120)

Figure 120 Add a swap

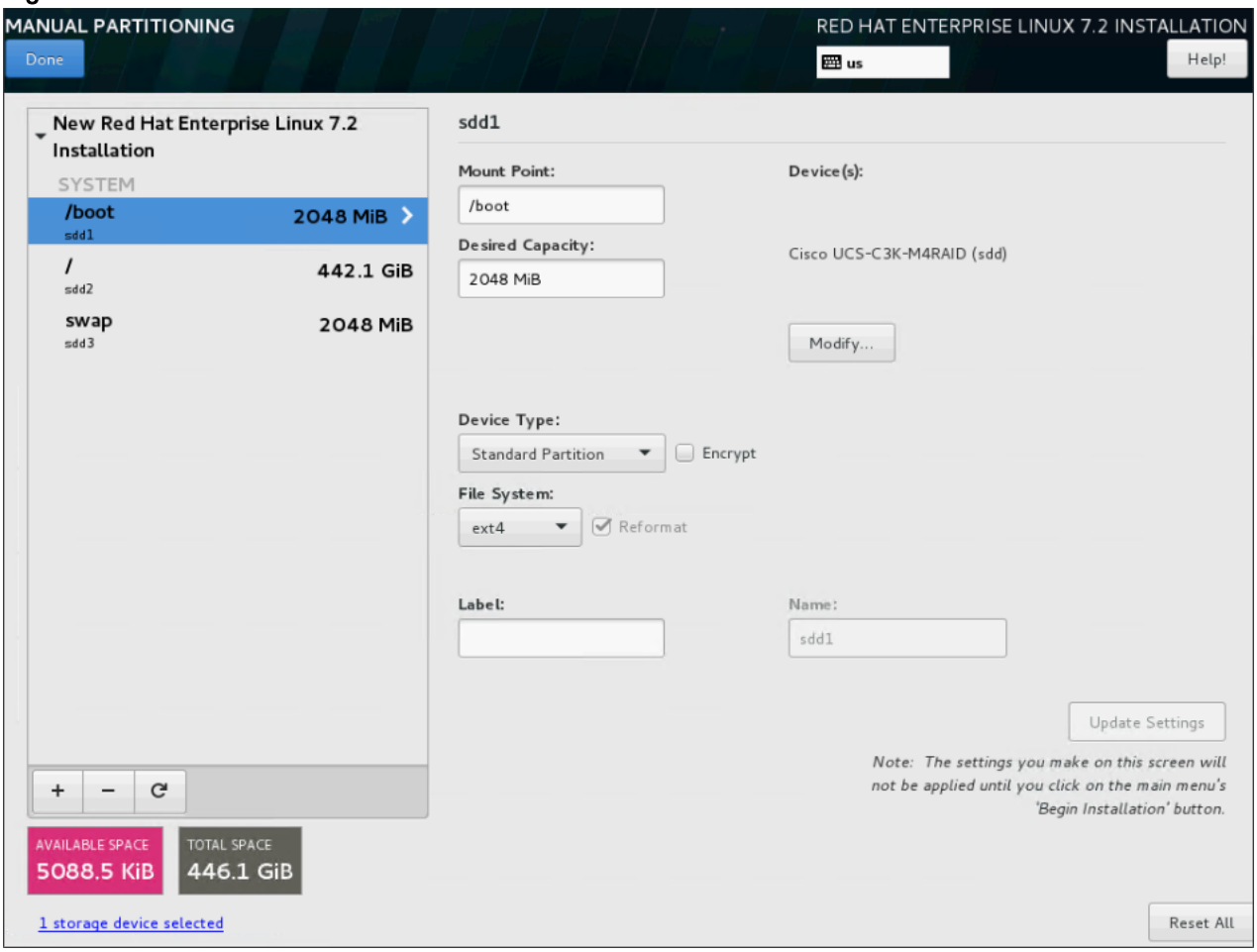


26. Select /boot partition and change the Device Type to Standard Partition and the file system to ext4. (Figure 121)

27. Select “/” partition and change the Device Type to Standard Partition and the file system to ext4.

28. Select “swap” partition and change the Device Type to Standard Partition.

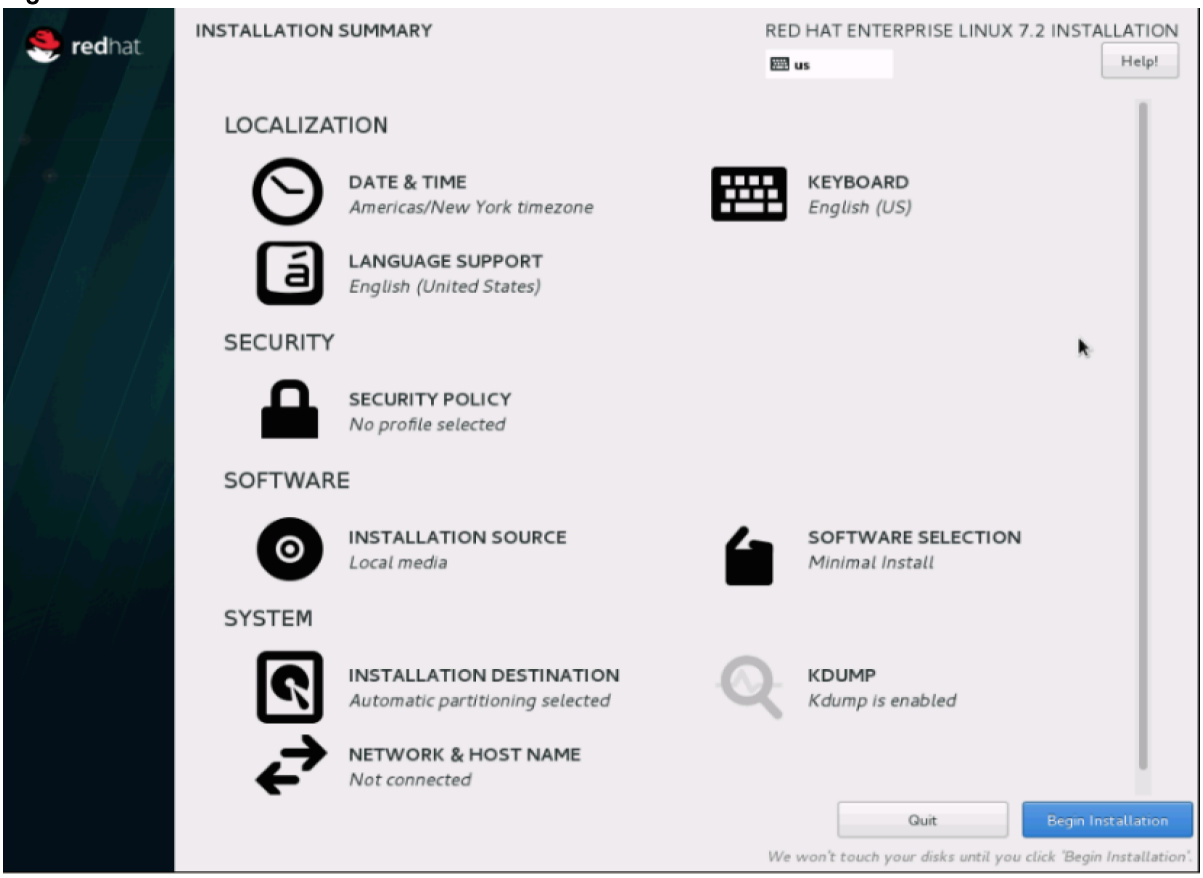
Figure 121 Standard Partition



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

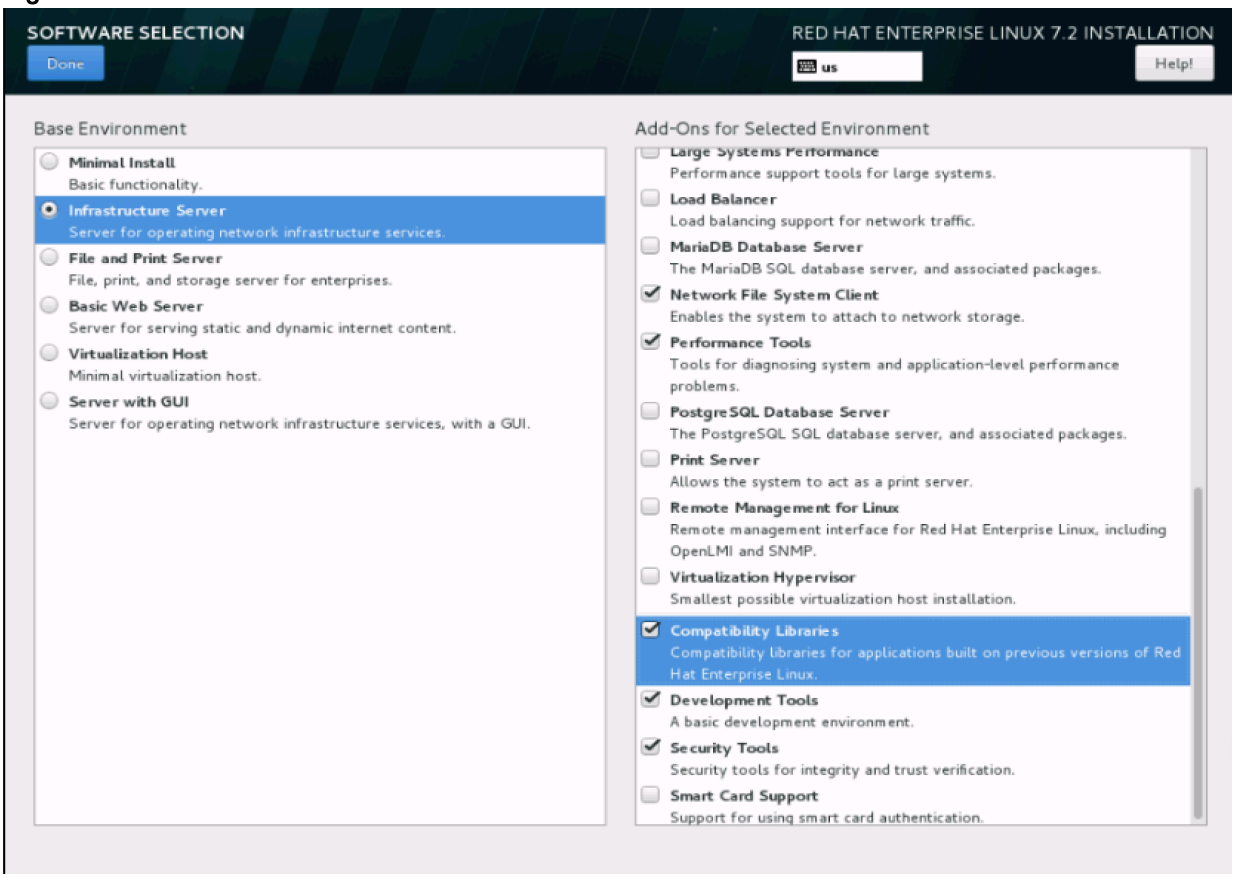
30. Click on Software Selection. (Figure 122)

Figure 122 Software Selection



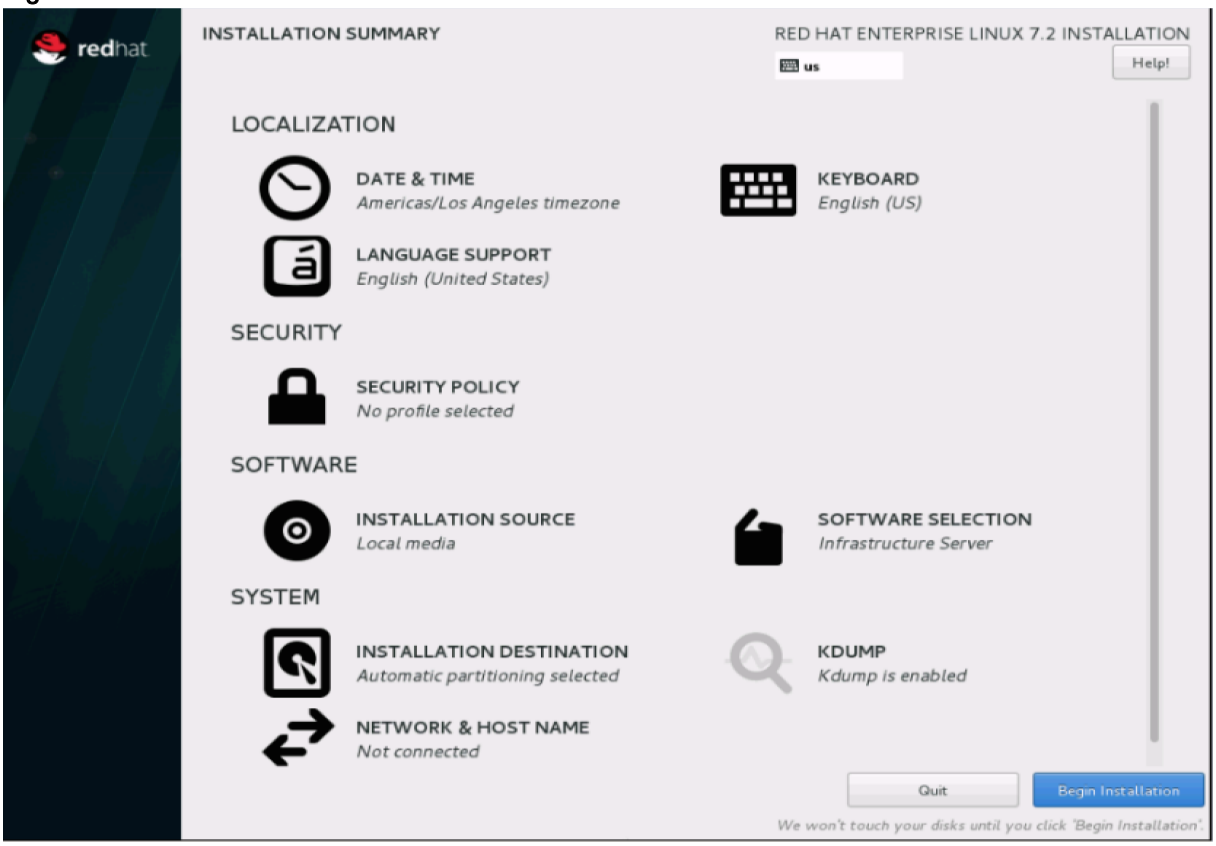
31. Select Infrastructure Server and select the Add-Ons as noted below. Click Done. (Figure 123)

Figure 123 Infrastructure Server



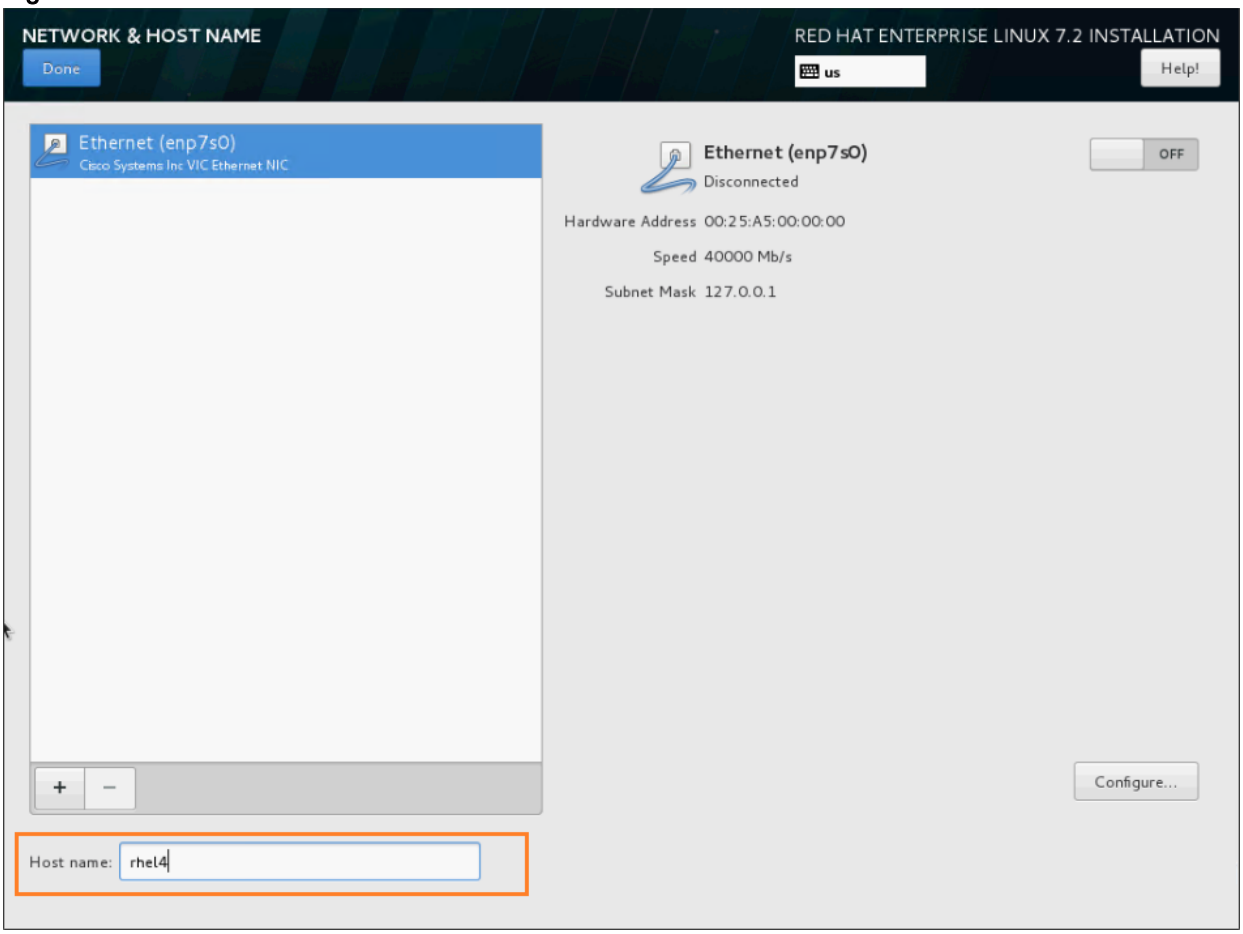
32. Click on Network and Hostname and configure Hostname and Networking for the Host. (Figure 124)

Figure 124 Network and Hostname



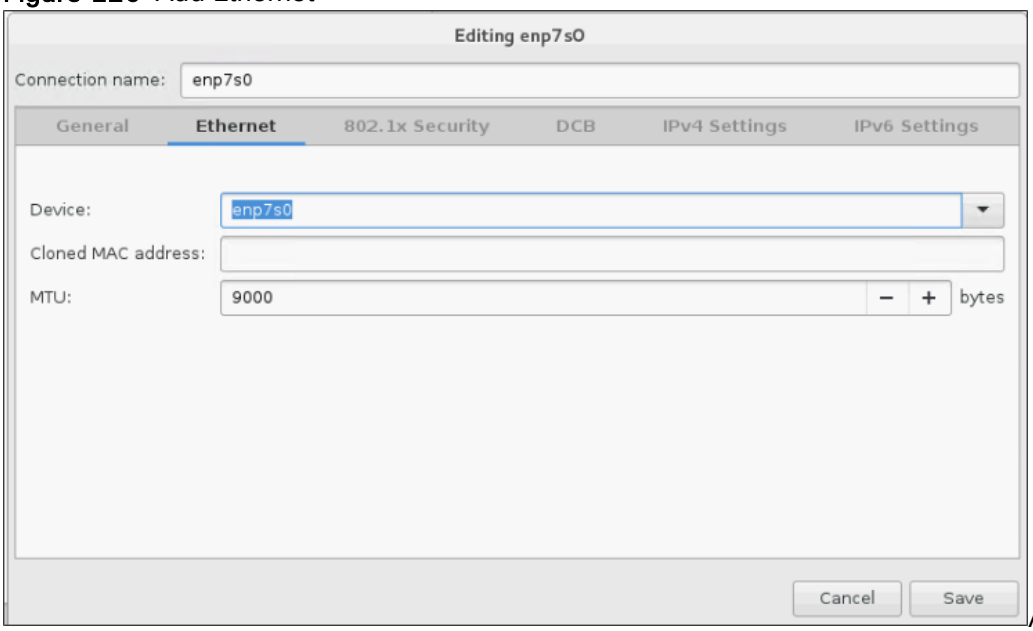
33. Type in the hostname as shown below. (Figure 125)

Figure 125 Add the Host Name



34. Click on Configure to open the Network Connectivity window. Click on Ethernet. (Figure 126)

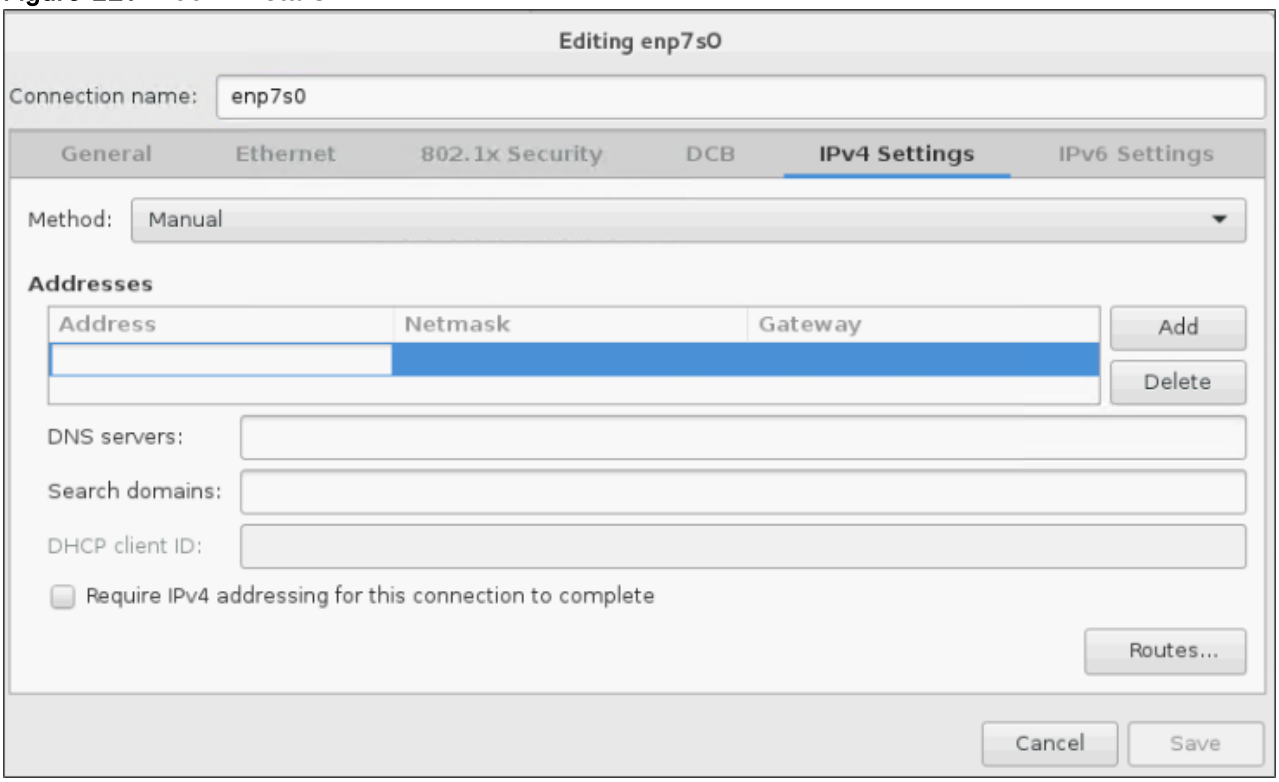
Figure 126 Add Ethernet



A

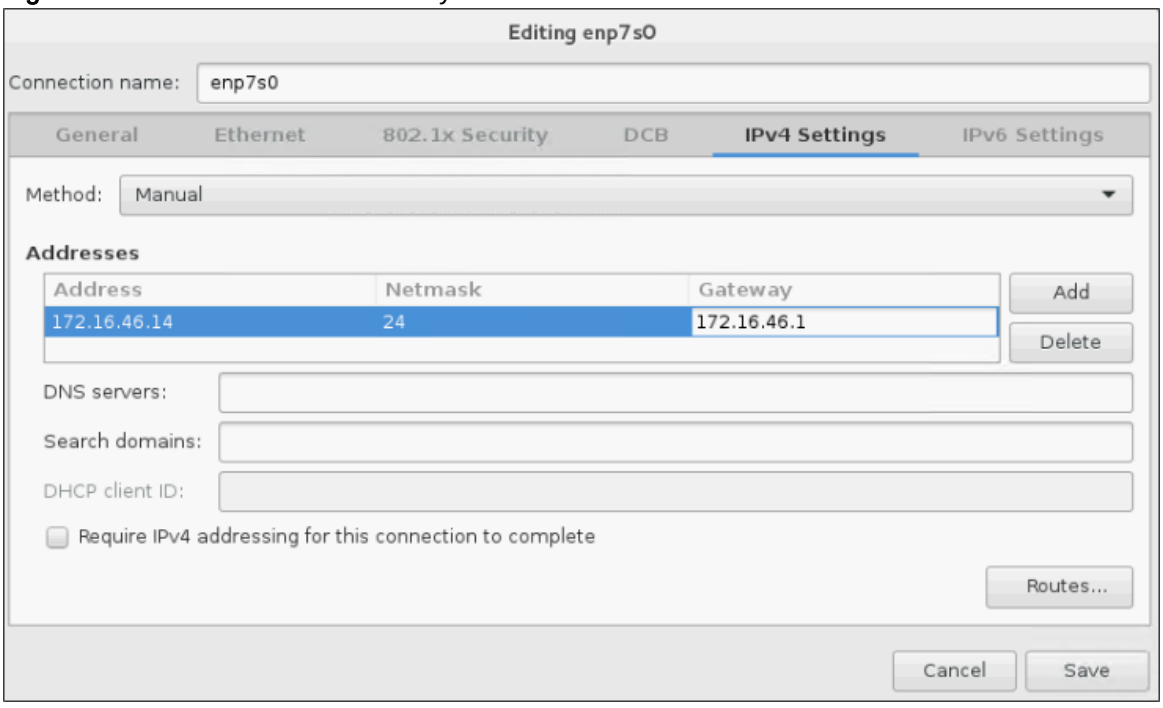
35. Click on IPv4 Settings and change the Method to Manual and click Add to enter the IP Address, Netmask and Gateway details. (Figure 127)

Figure 127 Add IP Details



36. Enter the desired IP address, Netmask and Gateway and click Save. (Figure 128)

Figure 128 Manual IP Address Entry



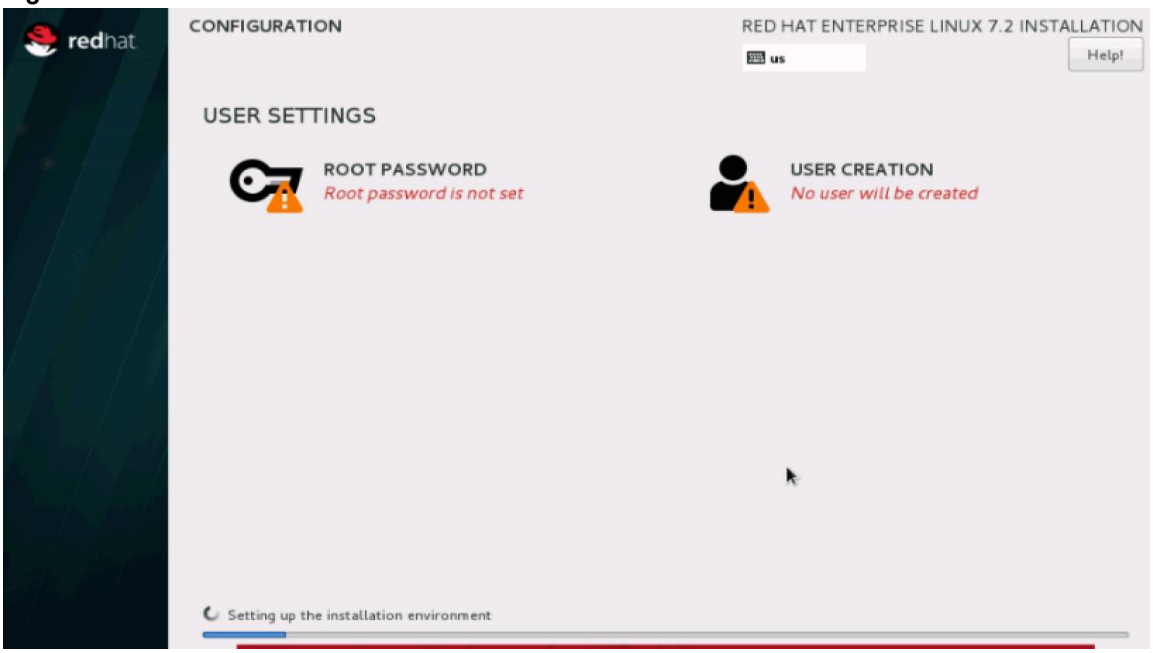
37. Click Save, update the hostname and turn Ethernet ON. Click Done to return to the main menu.

38. Click Begin Installation in the main menu. (Figure 129)

Figure 129 Begin Installation

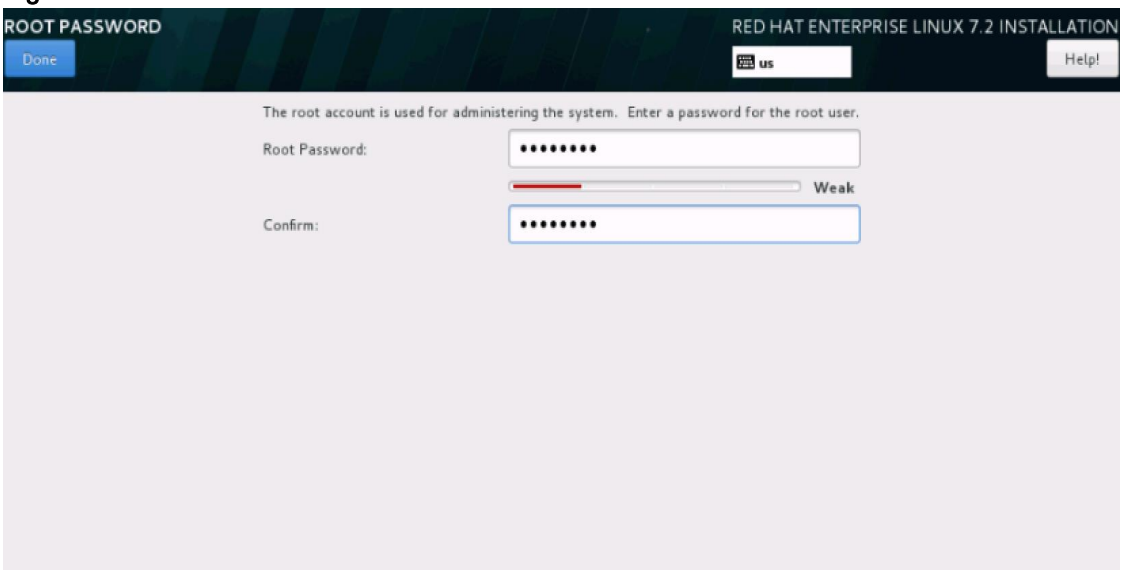
39. Select `Root Password` in the User Settings. (Figure 130)

Figure 130 Root Password



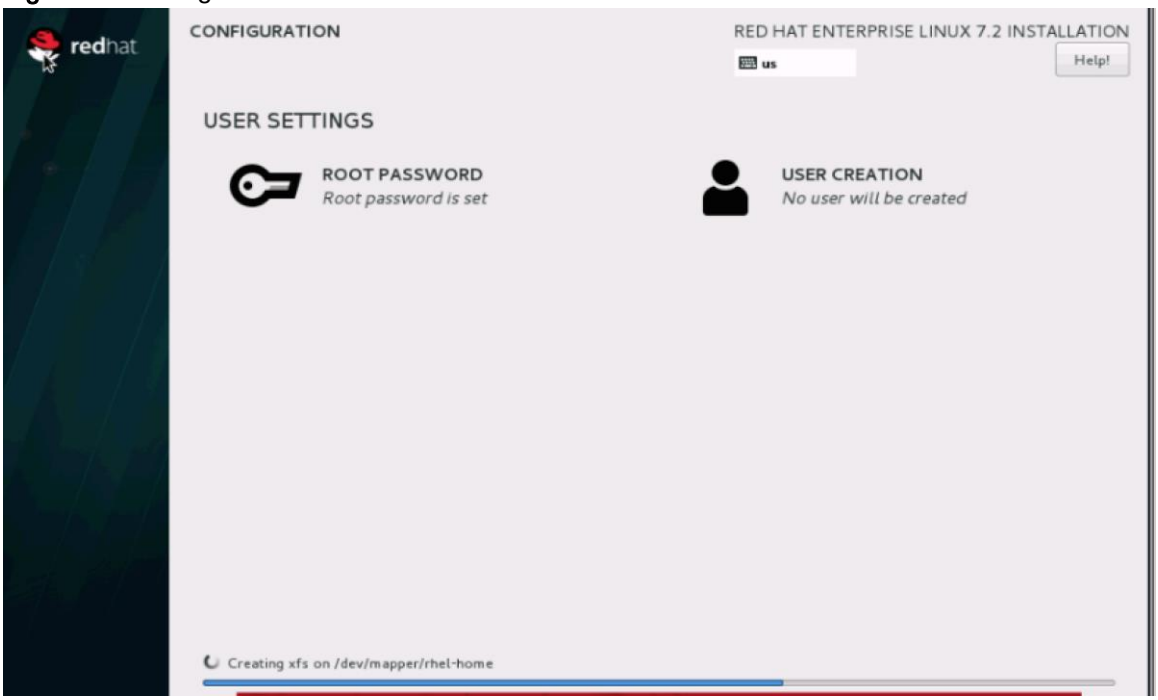
40. Enter the `Root Password` and click `done`. (Figure 131)

Figure 131 Root Password



The Installation Progress window displays the process. (Figure 132)

Figure 132 Progess Screen



41. Once the installation is complete reboot the system.

42. 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 1.

Table 1 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

Hostname	eth0
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

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.

```
[root@rhel1 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
87:78:ad:cc:56:0b:52:e4:0a:86:19:23:cb:27:5e:ed root@rhel1
The key's randomart image is:
+--[ RSA 2048]----+
|. o     .        |
|.o =.  o         |
|.oooo.  o        |
|. +... + o       |
| .   E+ S +      |
|       = = .     |
|        = .      |
|       .         |
|                 |
+-----------------+
```

1. 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/
```

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

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

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

```
vi ~/.ssh/config

ServerAliveInterval 99
StrictHostKeyChecking no
```

4. 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:

2. On Admin Node (rhel1)

```
#vi /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 \ lo-
calhost4.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
```

```
[root@rhel1 ~]# yum -y install createrepo
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
rhel7.2                                                                                  | 4.1 kB  00:00:00
(1/2): rhel7.2/group_gz                                                                  | 136 kB  00:00:00
(2/2): rhel7.2/primary_db                                                                | 3.6 MB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package createrepo.noarch 0:0.9.9-23.el7 will be installed
--> Processing Dependency: deltarpm for package: createrepo-0.9.9-23.el7.noarch
--> Processing Dependency: python-deltarpm for package: createrepo-0.9.9-23.el7.noarch
--> Running transaction check
---> Package deltarpm.x86_64 0:3.6-3.el7 will be installed
---> Package python-deltarpm.x86_64 0:3.6-3.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================================
 Package                              Arch                          Version                          Repository                      Size
================================================================================================================================================
Installing:
 createrepo                           noarch                        0.9.9-23.el7                     rhel7.2                         92 k
Installing for dependencies:
 deltarpm                             x86_64                        3.6-3.el7                        rhel7.2                         82 k
 python-deltarpm                      x86_64                        3.6-3.el7                        rhel7.2                         31 k

Transaction Summary
================================================================================================================================================
Install  1 Package (+2 Dependent packages)

Total download size: 205 k
Installed size: 553 k
Downloading packages:
------------------------------------------------------------------------------------------------------------------------------------------------
Total                                                                                    68 MB/s | 205 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : deltarpm-3.6-3.el7.x86_64                                                                                  1/3
  Installing : python-deltarpm-3.6-3.el7.x86_64                                                                           2/3
  Installing : createrepo-0.9.9-23.el7.noarch                                                                             3/3
rhel7.2/productid                                                                        | 1.6 kB  00:00:00
  Verifying  : python-deltarpm-3.6-3.el7.x86_64                                                                           1/3
  Verifying  : deltarpm-3.6-3.el7.x86_64                                                                                  2/3
  Verifying  : createrepo-0.9.9-23.el7.noarch                                                                             3/3

Installed:
  createrepo.noarch 0:0.9.9-23.el7

Dependency Installed:
  deltarpm.x86_64 0:3.6-3.el7                   python-deltarpm.x86_64 0:3.6-3.el7

Complete!
[root@rhel1 ~]#
```

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.

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/clustersh
ell/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.

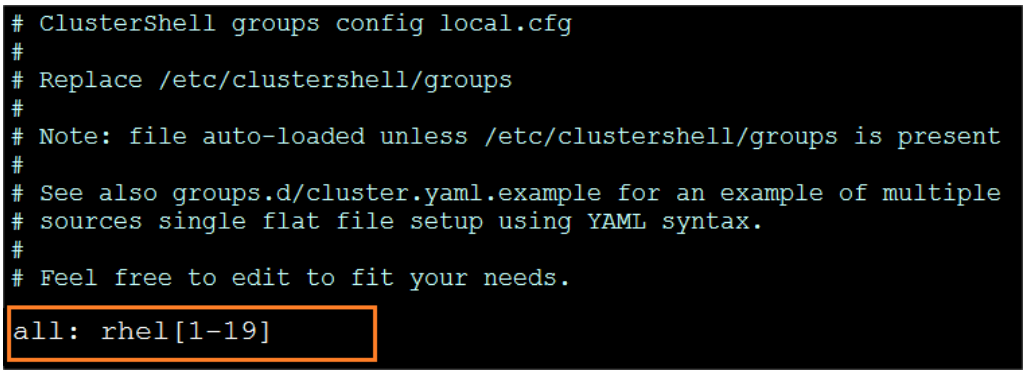
```
#yum –y install clustershell-1.71.el7.noarch.rpm
```

```
[root@rhel1 ~]# yum -y install clustershell-1.7-1.el7.noarch.rpm
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Examining clustershell-1.7-1.el7.noarch.rpm: clustershell-1.7-1.el7.noarch
Marking clustershell-1.7-1.el7.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package clustershell.noarch 0:1.7-1.el7 will be installed
--> Processing Dependency: PyYAML for package: clustershell-1.7-1.el7.noarch
--> Running transaction check
---> Package PyYAML.x86_64 0:3.10-11.el7 will be installed
--> Processing Dependency: libyaml-0.so.2()(64bit) for package: PyYAML-3.10-11.el7.x86_64
--> Running transaction check
---> Package libyaml.x86_64 0:0.1.4-11.el7_0 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                  Arch                 Version                      Repository                                   Size
================================================================================================================================
Installing:
 clustershell             noarch               1.7-1.el7                    /clustershell-1.7-1.el7.noarch              1.8 M
Installing for dependencies:
 PyYAML                   x86_64               3.10-11.el7                  rhel7.2                                     153 k
 libyaml                  x86_64               0.1.4-11.el7_0               rhel7.2                                      55 k

Transaction Summary
================================================================================================================================
Install  1 Package (+2 Dependent packages)

Total size: 2.0 M
Total download size: 208 k
Installed size: 2.5 M
Downloading packages:
--------------------------------------------------------------------------------------------------------------------------------
Total                                                                                    98 MB/s | 208 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : libyaml-0.1.4-11.el7_0.x86_64                                                                              1/3
  Installing : PyYAML-3.10-11.el7.x86_64                                                                                  2/3
  Installing : clustershell-1.7-1.el7.noarch                                                                              3/3
  Verifying  : libyaml-0.1.4-11.el7_0.x86_64                                                                              1/3
  Verifying  : clustershell-1.7-1.el7.noarch                                                                              2/3
  Verifying  : PyYAML-3.10-11.el7.x86_64                                                                                  3/3

Installed:
  clustershell.noarch 0:1.7-1.el7

Dependency Installed:
  PyYAML.x86_64 0:3.10-11.el7                       libyaml.x86_64 0:0.1.4-11.el7_0

Complete!
[root@rhel1 ~]#
```

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:

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

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

```
#yum –y install httpd
```

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

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

ServerName 172.16.46.11:80
```

3. 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 "service firewalld stop"

#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:

1. Disable Network manager on all nodes.

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

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

2. 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.

3. Install and Start dnsmasq on Admin node.

```
#service dnsmasq start

#chkconfig dnsmasq on
```

4. 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.

5. 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

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
```

5. 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
```

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.local`.

   ```
   #clush -a -b “cat /root/thp_disable >> /etc/rc.local”
   ```

Disable IPv6 Defaults

5. 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"
```

6. 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.

1. To download storcli go to: http://docs.avagotech.com/docs/1.19.04_StorCLI.zip
2. Extract the zip file and copy storcli-1.19.04-1.noarch.rpm from the linux directory.
3. Download storcli and its dependencies and transfer to Admin node.

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

4. 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/
```

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

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

6. 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
```

7. Copy storcli64 to all the nodes:

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

8. 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.

9. 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
```

10. 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
```

(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.

3. Change the mode to include execution privileges.

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

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

5. 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}'
```

6. 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.

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.

7. 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 reconfigure 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.

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

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
```

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

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

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

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

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

5. 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.

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

7. `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 In-
stalled' -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
```

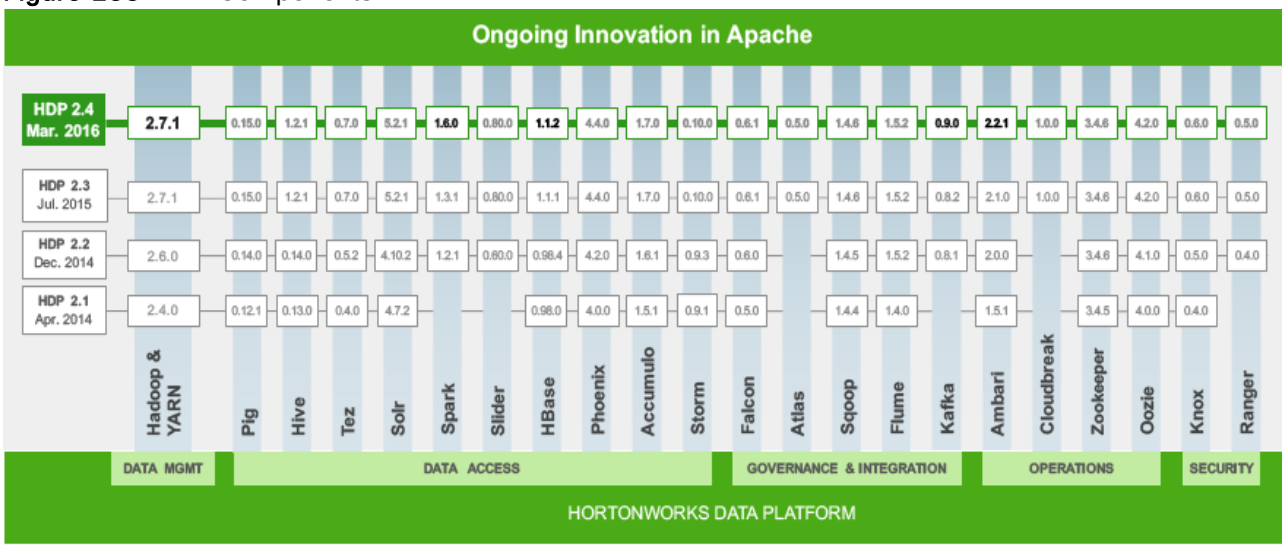
3. Run the Cluster Verification tool from the admin node. This can be run before starting Hadoop to identify any discrepancies in Post OS Configuration between the servers or during troubleshooting of any cluster / Hadoop issues.

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

Installing HDP 2.5

HDP is an enterprise grade, hardened Hadoop distribution. HDP combines Apache Hadoop and its related projects into a single tested and certified package. HPD 2.5 components are depicted in Figure 133 below. The following section goes in to detail on how to install HDP 2.5 on the cluster.

Figure 133 HDP Components



Pre-Requisites for HDP

This section details the pre-requisites for HDP Installation such as setting up of HDP Repositories.

Hortonworks Repo

1. From a host connected to the Internet, download the Hortonworks repositories as shown below and transfer to the admin node.

```
mkdir -p /tmp/Hortonworks

cd /tmp/Hortonworks/
```

2. Download the Hortonworks HDP Repo:

wget http://public-repo-1.hortonworks.com/HDP/centos7/2.x/updates/2.5.0.0/HDP-2.5.0.0-centos7-rpm.tar.gz

```
[root@Rack08-JB Hortonworks]# wget http://public-repo-1.hortonworks.com/HDP/centos7/2.x/updates/2.5.0.0/HDP-2.5.0
.0-centos7-rpm.tar.gz
--2016-09-01 14:43:25--  http://public-repo-1.hortonworks.com/HDP/centos7/2.x/updates/2.5.0.0/HDP-2.5.0.0-centos7
-rpm.tar.gz
Resolving public-repo-1.hortonworks.com... 54.230.144.14, 54.230.144.149, 54.230.144.169, ...
Connecting to public-repo-1.hortonworks.com|54.230.144.14|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5340741705 (5.0G) [application/x-tar]
Saving to: "HDP-2.5.0.0-centos7-rpm.tar.gz"

 0% [                                                                    ] 7,568,169   3.09M/s
```

3. Download Hortonworks HDP-Utils Repo:

   wget http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.21/repos/centos7/HDP-UTILS-1.1.0.21-centos7.tar.gz

```
[root@Rack08-JB Hortonworks]# wget http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.21/repos/centos7/HDP-UTIL
S-1.1.0.21-centos7.tar.gz
--2016-09-01 14:44:41--  http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.21/repos/centos7/HDP-UTILS-1.1.0.21
-centos7.tar.gz
Resolving public-repo-1.hortonworks.com... 54.230.144.14, 54.230.144.73, 54.230.144.196, ...
Connecting to public-repo-1.hortonworks.com|54.230.144.14|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 780929800 (745M) [application/x-gzip]
Saving to: "HDP-UTILS-1.1.0.21-centos7.tar.gz"

 0% [                                                                    ] 4,761,947   2.11M/s
```

4. Download Ambari Repo:

   wget http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.4.0.1/AMBARI-2.4.0.1-centos7.tar.gz

```
[root@Rack08-JB Hortonworks]# wget http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.4.0.1/AMBARI
-2.4.0.1-centos7.tar.gz
--2016-09-01 14:45:19--  http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.4.0.1/AMBARI-2.4.0.1-c
entos7.tar.gz
Resolving public-repo-1.hortonworks.com... 54.230.144.188, 54.230.144.197, 54.230.144.149, ...
Connecting to public-repo-1.hortonworks.com|54.230.144.188|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1385798830 (1.3G) [application/x-tar]
Saving to: "AMBARI-2.4.0.1-centos7.tar.gz"

 0% [                                                                    ] 1,236,941   1.20M/s
```

5. Copy the repository directory to the admin node:

   ```
   scp -r /tmp/Hortonworks/ rhel1:/var/www/html
   ```

6. Login to rhel1.

7. Extract the files:

   ```
   cd /var/www/html/Hortonworks

   tar -zxvf HDP-2.5.0.0-centos7-rpm.tar.gz

   tar -zxvf HDP-UTILS-1.1.0.21-centos7.tar.gz

   tar -zxvf AMBARI-2.4.0.1-centos7.tar.gz
   ```

8. Create the `hdp.repo` file with following contents:

   ```
   vi /etc/yum.repos.d/hdp.repo
   ```

```
[HDP-2.5.0.0]

name= Hortonworks Data Platform Version - HDP-2.5.0.0

baseurl= http://rhel1/Hortonworks/HDP/centos7

gpgcheck=0

enabled=1

priority=1


[HDP-UTILS-1.1.0.20]

name=Hortonworks Data Platform Utils Version - HDP-UTILS-1.1.0.21

baseurl= http://rhel1/Hortonworks/HDP-UTILS-1.1.0.21/repos/centos7

gpgcheck=0

enabled=1

priority=1
```

```
[root@rhel1 ~]# vi /etc/yum.repos.d/hdp.repo
[root@rhel1 ~]# cat /etc/yum.repos.d/hdp.repo
[HDP-2.5.0.0]
name= Hortonworks Data Platform Version - HDP-2.5.0.0
baseurl= http://rhel1/Hortonworks/HDP/centos7
gpgcheck=0
enabled=1
priority=1

[HDP-UTILS-1.1.0.21]
name=Hortonworks Data Platform Utils Version - HDP-UTILS-1.1.0.21
baseurl= http://rhel1/Hortonworks/HDP-UTILS-1.1.0.21/repos/centos7
gpgcheck=0
enabled=1
priority=1
```

9. Create the Ambari repo file with following contents:

```
vi /etc/yum.repos.d/ambari.repo
[ambari-2.4.0.1-1]

name=ambari-2.4.0.1-1

baseurl=http://rhel1/Hortonworks/AMBARI-2.4.0.1/centos7/2.4.0.1-1

gpgcheck=0
```

```
enabled=1

priority=1
```

```
[root@rhel1 ~]# vi /etc/yum.repos.d/ambari.repo
[root@rhel1 ~]# cat /etc/yum.repos.d/ambari.repo
[ambari-2.4.0.1-1]
name=ambari-2.4.0.1-1
baseurl=http://rhel1/Hortonworks/AMBARI-2.4.0.1/centos7/2.4.0.1-1
gpgcheck=0
enabled=1
priority=1
```

10. 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/hdp.repo --dest=/etc/yum.repos.d/
clush -a -b -c /etc/yum.repos.d/ambari.repo --dest=/etc/yum.repos.d/
```

HDP Installation

To install HDP, complete the following the steps:

Install and Setup Ambari Server on rhel1

```
yum -y install ambari-server
```

```
[root@rhel1 ~]# yum -y install ambari-server
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscriptio
n-manager to register.
HDP-2.5.0.0                                                | 2.9 kB  00:00:00
HDP-UTILS-1.1.0.21                                         | 2.9 kB  00:00:00
ambari-2.4.0.1-1                                           | 2.9 kB  00:00:00
(1/3): HDP-2.5.0.0/primary_db                              |  69 kB  00:00:00
(2/3): HDP-UTILS-1.1.0.21/primary_db                       |  34 kB  00:00:00
(3/3): ambari-2.4.0.1-1/primary_db                         | 8.3 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package ambari-server.x86_64 0:2.4.0.1-1 will be installed
--> Processing Dependency: postgresql-server >= 8.1 for package: ambari-server-2.4.0.1-1.
x86_64
--> Running transaction check
---> Package postgresql-server.x86_64 0:9.2.13-1.el7_1 will be installed
--> Processing Dependency: postgresql-libs(x86-64) = 9.2.13-1.el7_1 for package: postgres
ql-server-9.2.13-1.el7_1.x86_64
--> Processing Dependency: postgresql(x86-64) = 9.2.13-1.el7_1 for package: postgresql-se
rver-9.2.13-1.el7_1.x86_64
--> Processing Dependency: libpq.so.5()(64bit) for package: postgresql-server-9.2.13-1.el
7_1.x86_64
--> Running transaction check
---> Package postgresql.x86_64 0:9.2.13-1.el7_1 will be installed
---> Package postgresql-libs.x86_64 0:9.2.13-1.el7_1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch         Version               Repository          Size
================================================================================
Installing:
 ambari-server          x86_64       2.4.0.1-1             ambari-2.4.0.1-1    646 M
Installing for dependencies:
```

External Database PostgreSQL Installation

The PostgreSQL database will be used by Ambari, Hive and Oozie services.

In this installation rhel1 will host the Hive, Oozie and Ambari services.

1. Login to rhel1 and perform the following steps.

```
yum -y install postgresql-*
```

```
[root@rhel1 ~]# yum -y install postgresql-*
```

```
postgresql-setup initdb

/bin/systemctl start postgresql.service

systemctl enable postgresql
```

```
[root@rhel1 ~]# postgresql-setup initdb
Initializing database ... OK

[root@rhel1 ~]# /bin/systemctl start  postgresql.service
[root@rhel1 ~]# systemctl enable postgresql
Created symlink from /etc/systemd/system/multi-user.target.wants/postgresql.service to /usr/lib/
/system/postgresql.service.
```

```
service postgresql status
```

```
[root@rhel1 ~]# service postgresql status
Redirecting to /bin/systemctl status  postgresql.service
● postgresql.service - PostgreSQL database server
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; vendor preset:
   Active: active (running) since Fri 2016-09-02 14:30:39 PDT; 1min 20s ago
 Main PID: 21155 (postgres)
   CGroup: /system.slice/postgresql.service
           ├─21155 /usr/bin/postgres -D /var/lib/pgsql/data -p 5432
           ├─21156 postgres: logger process
           ├─21158 postgres: checkpointer process
           ├─21159 postgres: writer process
           ├─21160 postgres: wal writer process
           ├─21161 postgres: autovacuum launcher process
           └─21162 postgres: stats collector process

Sep 02 14:30:38 rhel1 systemd[1]: Starting PostgreSQL database server...
Sep 02 14:30:38 rhel1 pg_ctl[21152]: LOG:  could not bind IPv6 socket: Cannot assign r
Sep 02 14:30:38 rhel1 pg_ctl[21152]: HINT:  Is another postmaster already running on p
Sep 02 14:30:39 rhel1 systemd[1]: Started PostgreSQL database server.
Hint: Some lines were ellipsized, use -l to show in full.
```

2. Update these files on rhel1 in the location chosen to install the databases for Hive, Oozie and Ambari, using the host ip addresses.

```
vi /var/lib/pgsql/data/pg_hba.conf
```

```
[root@rhel1 ~]# vi /var/lib/pgsql/data/pg_hba.conf
[root@rhel1 ~]# cat /var/lib/pgsql/data/pg_hba.conf | tail -n 20



# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            ident
# IPv6 local connections:
host    all             all             ::1/128                 ident
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local   replication     postgres                                peer
#host    replication     postgres        127.0.0.1/32            ident
#host    replication     postgres        ::1/128                 ident
local   all     postgres        peer
local   all     all     md5
host    all     postgres,hive,oozie     172.16.46.11/24 md5
host    all     ambari  172.16.46.11/24 md5
```

```
vi /var/lib/pgsql/data/postgresql.conf

search for listen_address and replace with (*)
```

```
[root@rhel1 ~]# vi /var/lib/pgsql/data/postgresql.conf
[root@rhel1 ~]# cat /var/lib/pgsql/data/postgresql.conf | grep listen_add
listen_addresses = '*'          # what IP address(es) to listen on;
```

3. Issue the following command to verify postgres has been installed correctly.

```
sudo -u postgres psql
```

```
[root@rhel1 ~]# sudo -u postgres psql
could not change directory to "/root"
psql (9.2.13)
Type "help" for help.

postgres=#
```

http://docs.hortonworks.com/HDPDocuments/Ambari-2.2.2.0/bk_ambari_reference_guide/content/_using_ambari_with_postgresql.html

Ambari

To set up PostgreSQL to be used with Ambari, complete the following steps:

1. Connect to PostgreSQL database admin utility using

```
sudo -u postgres psql
```

2. Create the Ambari database.

```
CREATE DATABASE ambari;
```

3. Create a user for Ambari with password “Cisco_123”.

```
CREATE USER ambari WITH PASSWORD 'Cisco_123';
```

4. Grant all privileges on the Ambari database to the Ambari user

```
GRANT ALL PRIVILEGES ON DATABASE ambari TO ambari;
```

5. Connect to Ambari database

```
\connect ambari
```

6. Create the Ambari schema authorization to Ambari user.

```
CREATE SCHEMA ambari AUTHORIZATION ambari;
```

7. Change the Ambari schema owner to Ambari user.

```
ALTER SCHEMA ambari OWNER TO ambari;
```

8. Alter the Ambari role, set search_path to 'ambari', 'public'.

```
ALTER ROLE ambari SET search_path to 'ambari','public';
```

9. Log out using `\q`

```
[root@rhel1 ~]# sudo -u postgres psql
could not change directory to "/root"
psql (9.2.13)
Type "help" for help.

postgres=# CREATE DATABASE ambari;
CREATE DATABASE
postgres=# CREATE USER ambari WITH PASSWORD 'Cisco_123';
CREATE ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE ambari TO ambari;
GRANT
postgres=# \connect ambari
You are now connected to database "ambari" as user "postgres".
ambari=# CREATE SCHEMA ambari AUTHORIZATION ambari;
CREATE SCHEMA
ambari=# ALTER SCHEMA ambari OWNER TO ambari;
ALTER SCHEMA
ambari=# ALTER ROLE ambari SET search_path to 'ambari','public';
ALTER ROLE
ambari=# \q
```

```
[root@rhel1 ~]# service postgresql restart
```

```
 [root@rhel1 ~]# psql -U ambari -h rhel1 -d ambari
```

10. Please enter the password.

```
[root@rhel1 ~]# psql -U ambari -h rhel1 -d ambari
Password for user ambari:
psql (9.2.13)
Type "help" for help.

ambari=>
```

Pre-load the Ambari database schema into your PostgreSQL database using the schema script.

11. To Load the Ambari Server database schema, find the `Ambari-DDL-Postgres-CREATE.sql` file in the `/var/lib/ambari-server/resources/` directory of the Ambari Server host after you have installed Ambari Server.

```
cd /var/lib/ambari-server/resources/
```

```
 psql -U ambari -h rhel1 -d ambari
```

12. Enter the password ‘Cisco_123’

```
\connect ambari
```

```
\i Ambari-DDL-Postgres-CREATE.sql;
```

```
[root@rhel1 ~]# cd /var/lib/ambari-server/resources/
[root@rhel1 resources]# psql -U ambari -h rhel1 -d ambari
Password for user ambari:
psql (9.2.13)
Type "help" for help.

ambari=> \connect ambari
You are now connected to database "ambari" as user "ambari".
ambari=> \i Ambari-DDL-Postgres-CREATE.sql;
psql:Ambari-DDL-Postgres-CREATE.sql:25: NOTICE:  CREATE TABLE / PR
"pk_stack" for table "stack"
psql:Ambari-DDL-Postgres-CREATE.sql:25: NOTICE:  CREATE TABLE / UN
tack" for table "stack"
CREATE TABLE
psql:Ambari-DDL-Postgres-CREATE.sql:32: NOTICE:  CREATE TABLE / PR
"pk_extension" for table "extension"
psql:Ambari-DDL-Postgres-CREATE.sql:32: NOTICE:  CREATE TABLE / UN
xtension" for table "extension"
CREATE TABLE
```

13. Check the table is created by running \dt command.

```
ambari=> \dt
                  List of relations
 Schema |             Name              | Type  | Owner
--------+-------------------------------+-------+--------
 ambari | adminpermission               | table | ambari
 ambari | adminprincipal                | table | ambari
 ambari | adminprincipaltype            | table | ambari
 ambari | adminprivilege                | table | ambari
 ambari | adminresource                 | table | ambari
 ambari | adminresourcetype             | table | ambari
 ambari | alert_current                 | table | ambari
 ambari | alert_definition              | table | ambari
 ambari | alert_group                   | table | ambari
 ambari | alert_group_target            | table | ambari
```

14. Log out with \q.

Hive

To set up PostgreSQL to be used with Hive, complete the following steps:

1. Connect to PostgreSQL database admin utility using

```
sudo -u postgres psql
```

2. Create the Hive database.

```
CREATE DATABASE hive;
```

3. Create a hive user with the password "Cisco_123".

```
CREATE USER hive WITH PASSWORD 'Cisco_123';
```

4. Grant all privileges on the Hive database to the hive user

```
GRANT ALL PRIVILEGES ON DATABASE hive TO hive;
```

5. Log out using `\q`

```
[root@rhel1 ~]# sudo -u postgres psql
could not change directory to "/root"
psql (9.2.13)
Type "help" for help.

postgres=# CREATE DATABASE hive;
CREATE DATABASE
postgres=# CREATE USER hive WITH PASSWORD 'Cisco_123';
CREATE ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE hive TO hive;
GRANT
```

Oozie

To set up PostgreSQL to be used with Oozie, complete the following steps:

1. Connect to PostgreSQL database admin utility using

```
sudo -u postgres psql
```

2. Create the O0zie database.

```
CREATE DATABASE oozie;
```

3. Create a user for Oozie and grant it permissions.

4. Create a Oozie user with the password "Cisco_123".

```
CREATE USER oozie WITH PASSWORD 'Cisco_123';
```

5. Grant all privileges on the Oozie database to the Oozie user

```
GRANT ALL PRIVILEGES ON DATABASE oozie TO oozie;
```

6. Log out using `\q`

```
[root@rhel1 ~]# sudo -u postgres psql
could not change directory to "/root"
psql (9.2.13)
Type "help" for help.

postgres=# CREATE DATABASE oozie;
CREATE DATABASE
postgres=# CREATE USER oozie WITH PASSWORD 'Cisco_123';
CREATE ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE oozie TO oozie;
GRANT
```

7. Connect to the admin node (rhel1) and run the commands described below.

```
yum -y install postgresql-jdbc*

ambari-server setup --jdbc-db=postgres --jdbc-
driver=/usr/share/java/postgresql-jdbc.jar
```

```
[root@rhel1 ~]# yum -y install postgresql-jdbc*
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manage
egister.
Package postgresql-jdbc-9.2.1002-5.el7.noarch already installed and latest version
Nothing to do
[root@rhel1 ~]# ambari-server setup --jdbc-db=postgres --jdbc-driver=/usr/share/java/postgresql-j
r
Using python  /usr/bin/python
Setup ambari-server
Copying /usr/share/java/postgresql-jdbc.jar to /var/lib/ambari-server/resources
If you are updating existing jdbc driver jar for postgres with postgresql-jdbc.jar. Please remove
ld driver jar, from all hosts. Restarting services that need the driver, will automatically copy t
w jar to the hosts.
JDBC driver was successfully initialized.
Ambari Server 'setup' completed successfully.
```

Downgrade Snappy on All Nodes

To downgrade snappy on all data nodes, run the following command from admin node:

```
clush -a -b yum -y downgrade snappy
```

Setting Up the Ambari Server on the Admin Node (rhel1)

```
ambari-server setup -j $JAVA_HOME
```

Note: Enter the advanced database configuration option - Yes and choose option 4 for the existing PostgreSQL Database selection.

```
[root@rhel1 ~]# ambari-server setup -j $JAVA_HOME
Using python  /usr/bin/python
Setup ambari-server
Checking SELinux...
SELinux status is 'disabled'
Customize user account for ambari-server daemon [y/n] (n)? n
Adjusting ambari-server permissions and ownership...
Checking firewall status...
Checking JDK...
WARNING: JAVA_HOME /usr/java/jdk1.8.0_91 must be valid on ALL hosts
WARNING: JCE Policy files are required for configuring Kerberos security. If you plan t
lease make sure JCE Unlimited Strength Jurisdiction Policy Files are valid on all hosts
Completing setup...
Configuring database...
Enter advanced database configuration [y/n] (n)? y
Configuring database...
================================================================================
Choose one of the following options:
[1] - PostgreSQL (Embedded)
[2] - Oracle
[3] - MySQL / MariaDB
[4] - PostgreSQL
[5] - Microsoft SQL Server (Tech Preview)
[6] - SQL Anywhere
[7] - BDB
================================================================================
Enter choice (1): 4
Hostname (localhost): rhel1
Port (5432):
Database name (ambari):
Postgres schema (ambari):
Username (ambari):
Enter Database Password (bigdata):
Re-enter password:
Configuring ambari database...
Configuring remote database connection properties...
```

```
Configuring ambari database...
Configuring remote database connection properties...
WARNING: Before starting Ambari Server, you must run the following DDL against the
he schema: /var/lib/ambari-server/resources/Ambari-DDL-Postgres-CREATE.sql
Proceed with configuring remote database connection properties [y/n] (y)? y
Extracting system views...
......ambari-admin-2.4.0.1.1.jar
.....
Adjusting ambari-server permissions and ownership...
Ambari Server 'setup' completed successfully.
```

1. Starting the Ambari Server

```
ambari-server start
```

```
[root@rhel1 ~]# ambari-server start
Using python  /usr/bin/python
Starting ambari-server
Ambari Server running with administrator privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start....................
Ambari Server 'start' completed successfully.
```

2. Confirm Ambari Server startup

```
ps -aef | grep ambari-server
```

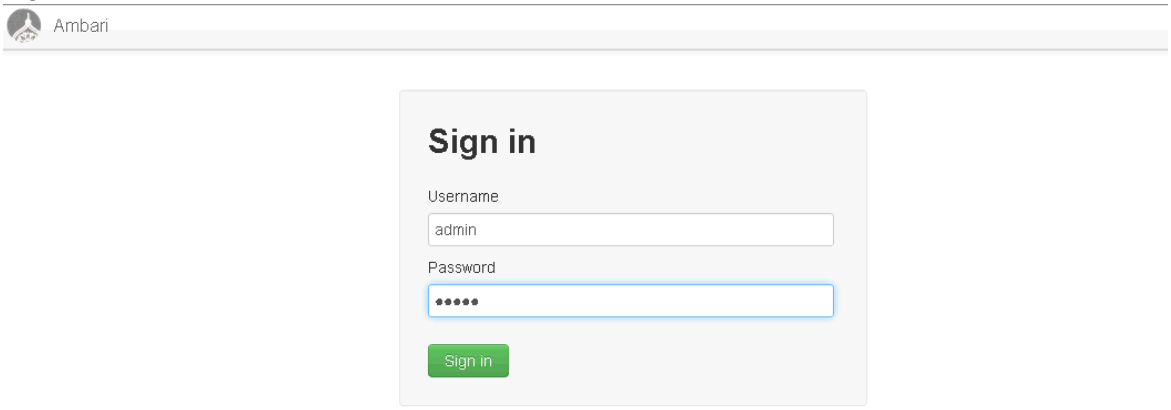
```
[root@rhel1 ~]# ambari-server start
Using python  /usr/bin/python
Starting ambari-server
Ambari Server running with administrator privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start....................
Ambari Server 'start' completed successfully.
[root@rhel1 ~]# ps -ef | grep ambari-server
root      5057     1 13 18:31 pts/4    00:00:06 /usr/java/jdk1.8.0_91/bin/java
-server -XX:NewRatio=3 -XX:+UseConcMarkSweepGC -XX:-UseGCOverheadLimit -XX:CMSI
nitiatingOccupancyFraction=60 -Dsun.zip.disableMemoryMapping=true -Xms512m -Xmx
2048m -Djava.security.auth.login.config=/etc/ambari-server/conf/krb5JAASLogin.c
onf -Djava.security.krb5.conf=/etc/krb5.conf -Djavax.security.auth.useSubjectCr
edsOnly=false -cp /etc/ambari-server/conf:/usr/lib/ambari-server/*:/usr/share/j
ava/postgresql-jdbc.jar org.apache.ambari.server.controller.AmbariServer
root      5133  4269  0 18:31 pts/4    00:00:00 grep --color=auto ambari-serve
```

Logging into the Ambari Server

Once the Ambari service has been started, access the Ambari Install Wizard through the browser.

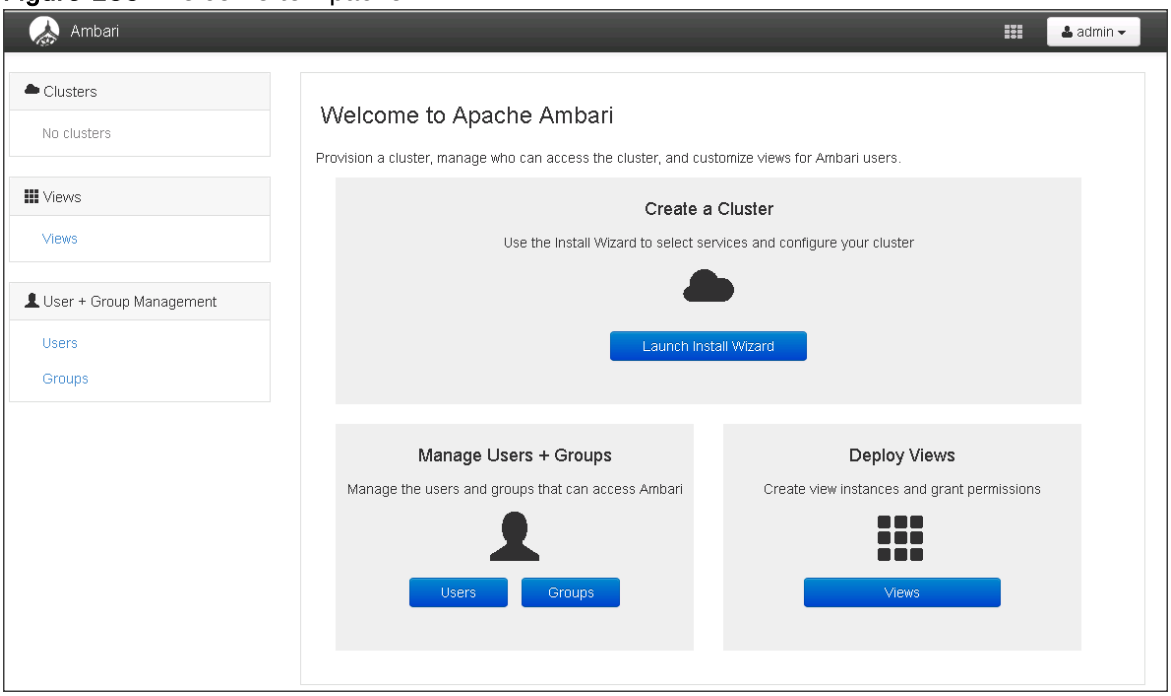
1. Point the browser to http://<ip address for rhel1>:8080.
2. The Ambari Login screen will open (Figure 134).
3. Log in to the Ambari Server using the default username/password: admin/admin. This can be changed at a later period of time.

Figure 134 Ambari Login Window



Once logged in, the “Welcome to Apache Ambari” window appears (Figure 135).

Figure 135 Welcome to Apache



Creating a Cluster

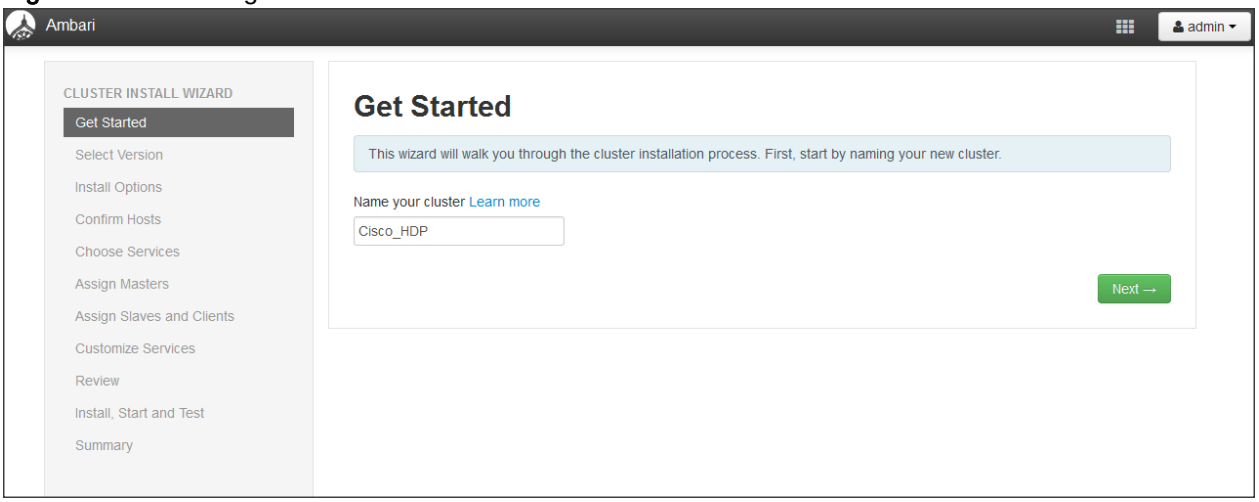
To create a cluster, complete the following steps:

1. Click the button under “Create a Cluster” to launch the install wizard as shown in Figure 135 above.

2. On the “Get Started” page (Figure 136) type “Cisco_HDP” as the name for the cluster.

3. Click Next.

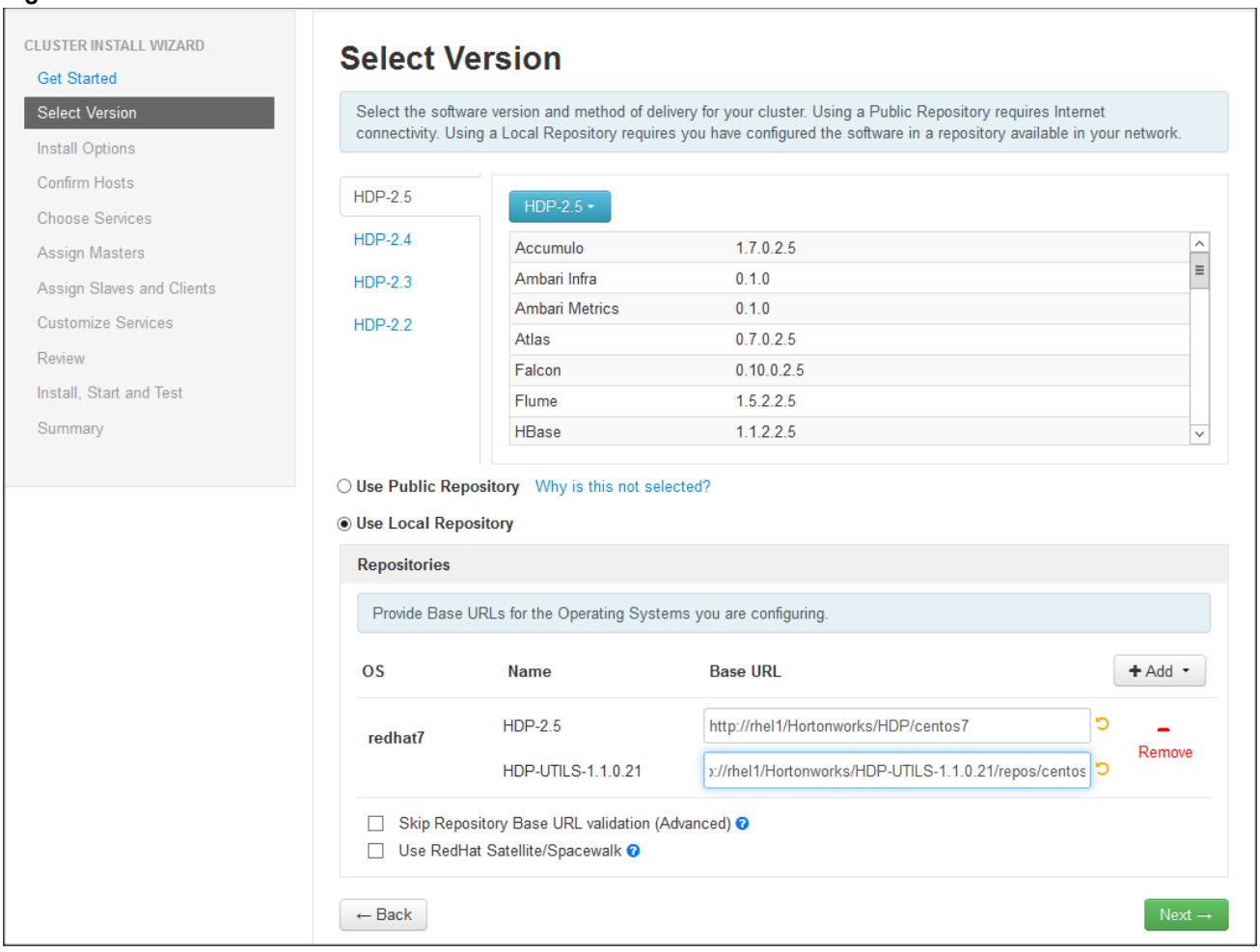
Figure 136 Creating a Cluster



Select a Version

1. In the next screen (Figure 137), select the HDP 2.5 stack.
2. Select “ Use Local Repository”.
3. Select the RedHat 7 checkbox.
4. Provide the base URL’s for the operating systems
5. Update the Redhat 7 HDP-2.5.0.0 URL to http://rhel1/Hortonworks/HDP/centos7
6. Update the Redhat 7 HDP-UTILS-1.1.0.21 URL to http://rhel1/Hortonworks/HDP-UTILS-1.1.0.21/repos/centos7
7. Select a Version. (Figure 137)

Figure 137 Select Version



Note: Make sure there are no trailing spaces after the URLs.

HDP Installation

To build up the cluster, the install wizard needs to know general information about how the cluster is to be set up. This requires providing the Fully Qualified Domain Name (FQDN) of each of the hosts. The wizard also needs to access the private key file that was created in *Set Up Password-less SSH*. It uses these to locate all the hosts in the system and to access and interact with them securely.

Figure 138 below shows the install wizard window.

1. Use the Target Hosts text box to enter the list of host names, one per line. Ranges inside brackets can also be used to indicate larger sets of hosts.

2. Select the option Provide your SSH Private Key in the Ambari cluster install wizard.

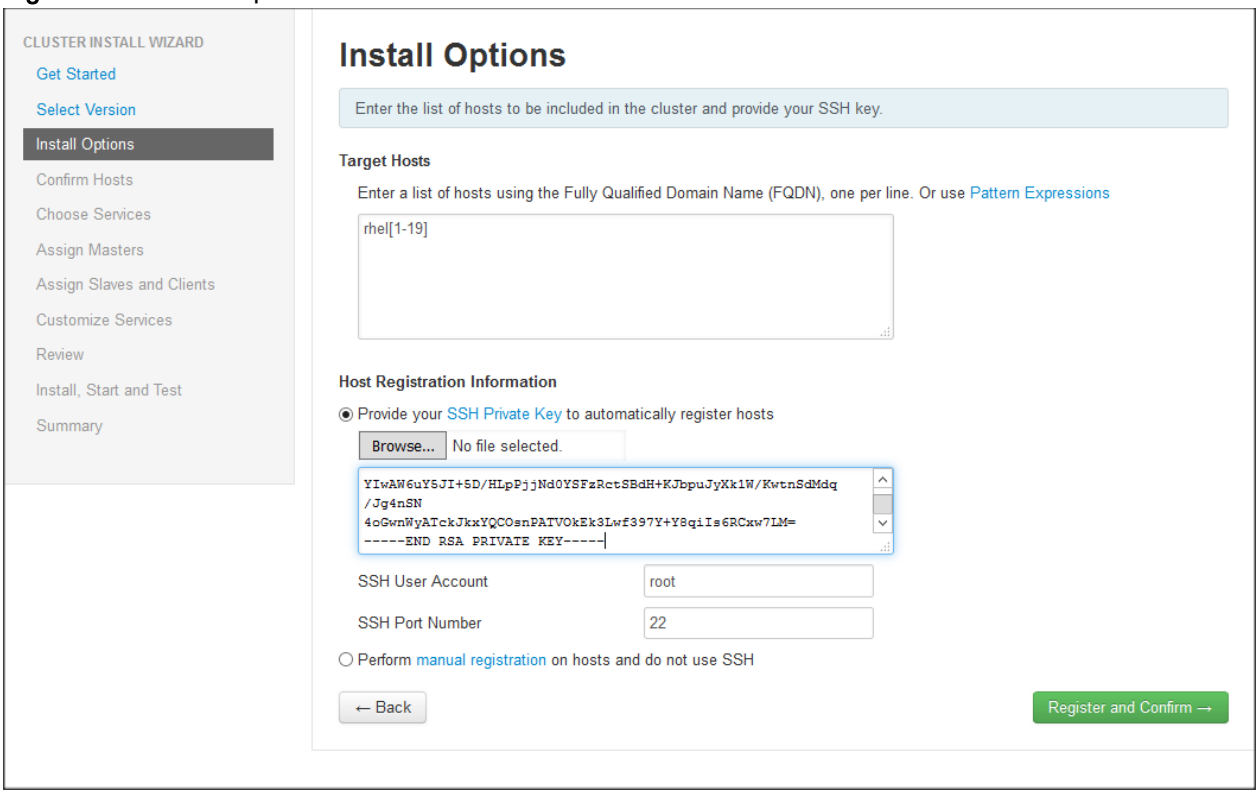
3. Copy the contents of the file /root/.ssh/id_rsa on rhel1 and paste it in the text area provided by the Ambari cluster install wizard.

Note: Make sure there is no extra white space after the text-----END RSA PRIVATE KEY-----

```
[root@rhel1 ~]# cat /root/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCAQEAyDOIRbk4mBZrizc0/gOM2iYT2h4vxkIxA/uvQVPthFreUdgT
Zehw/Qtdk7meeqhgqsHmb1CriF0m6SxvPEXW2cGoAx75hZwTuDIR3Qlvk6oYUmDW
BKq5TMfUMKfD7tknkGkg5N+YHsPCoNILlz/WqcO1hZZOtiCmrxeRnPGSlJY74/Db
A0BewMuNajAoVppPD6cLGF6/NKORpEDUnCuwe5pCRV5tko+gzBeBF5oeCS6Ya6I7
nS0HplJXV0Mv23SNUwl3cswbqLdrr3atG6YRieVrmmr/PlrKMp192tzQ1mHZMBqG
w1RJTILjygW0gp5g7NQBGeM7sX4V60mzv4vmzwIBIwKCAQEAg4+UEI+o2PjKVCuX
2h+XEwMUXCJ3KoNEyBpr2nj7KxckYas/8oLN6B1pYROUB3X2YZVc6hBwuLI+JDMk
hrGNMALqwDjtHUlOyX/9HDlmlDyTo9k8LvPY2q8zqvHnJ+3Jisi92Dspc01xRRxQ
wnpofjAm1CDx5WXp4MZYX9HynCcKmheFefobLys6gloxd84eHW1y6b0xU1dh7hsQ
pcK+xpdFW1sHYFbvckTuCHUAezF4+uBT5F0PMiD7PwzrvbXKA65ABuezv9gg2/I1
PekIkRvbosniFbBUi2ZOS1uN/gsaZgmSQ9gTarJlV8zMy6K31LETcOckl2LZHRX2
5sEx6wKBgQD9CiKc0HFiulrQWW5cLTDJU8wzTiNK4M9lQb2LOhfFuZfluiAl3Ref
yiL9MjE3A5Mnn9pcRxMmXXPF4t9iuLh3+3tCsr1TzPml4WT+Fipa9sh+3JZ2HKgm
pCquAEdoFRK4oP3/yYQg95gie2SC9sBOz6zVohdyNUvnkiMb9vwi3wKBgQDKiyTi
Yu421OwsYKfZ7YjomjRKUFaH4CKtnyJy1SM3wFPRnZJd4BUaMq0DaTxr2tW4si+4
t88M8XS6FHGHymSqRtL0tYzMlmmwUtjCLNZQfqSeg1NovekXxXL0iUzel8PL3ZOH
AeBj0/GLQ3SF/PGWMokCwNtaJoV/xldBdIsqEQKBgEERPBmx8UVF3NZ9ZYVqtMYO
O9KtsU3Ex52x0ad1VpHt5TsSmo1kvO6TEE+8cw4lfZx5j+vXwxh+bjozBj30/Dwc
GGGbrQbrkKscs5HLL3Z5+QqtwEpB4hiQnUKvnVVHP1QMJA6S53YxCdz7KHlypnqq
bkWQFKhW2QEIUivDKuRlAoGASzr/EkIAtUfFb5GdbjOn4V3Y6Gb7kY3DvNS1BhSm
rk7ADAdTnzX5NZ3LO8gAf9TwS+ppfx+zTfNInOMFmNYlY9EpyJs0S/1adLE0roWu
sC8J8bu/5RNWk8z+z9s5zwUrd5txT2cY1J8t1KQGtWyUPxoVoe/ccfENA5LP872S
xnsCgYAFRE4SbB416p9miiR1+gNCiihM9N+FmHMmcP/y80QL/MoAYoHB1Tn8cwVu
l+sju4bWGUZvnGMWXwpEU5zVBra+yShh309IwjP/1kpCNWz7CX+/uI6FY+slZxTr
t5P/AvhOvUKMhRFjXFQoY5yqNUkasvIu6S8Q1unl8N2IhEgw1g==
-----END RSA PRIVATE KEY-----
```

4. Click the Register and Confirm button to continue.

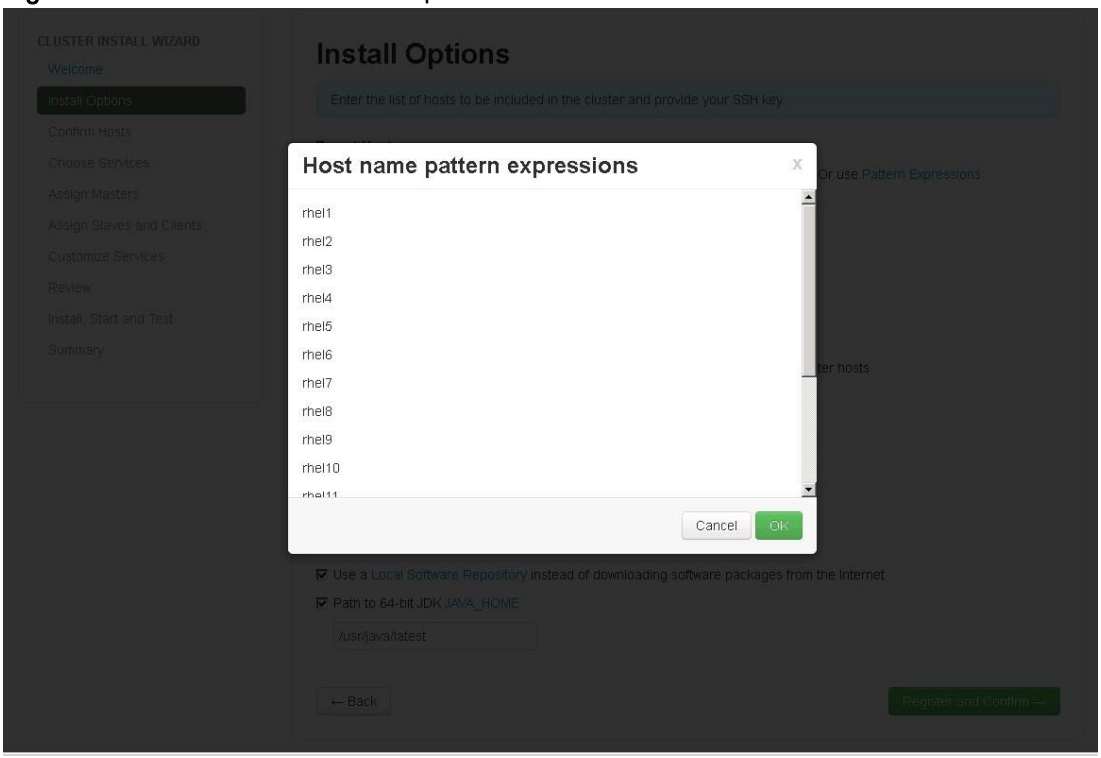
Figure 138 Install Options



Hostname Pattern Expressions

1. Click OK on the Host Name Pattern Expressions popup (Figure 139).

Figure 139 Host Name Pattern Expressions



Confirm Hosts

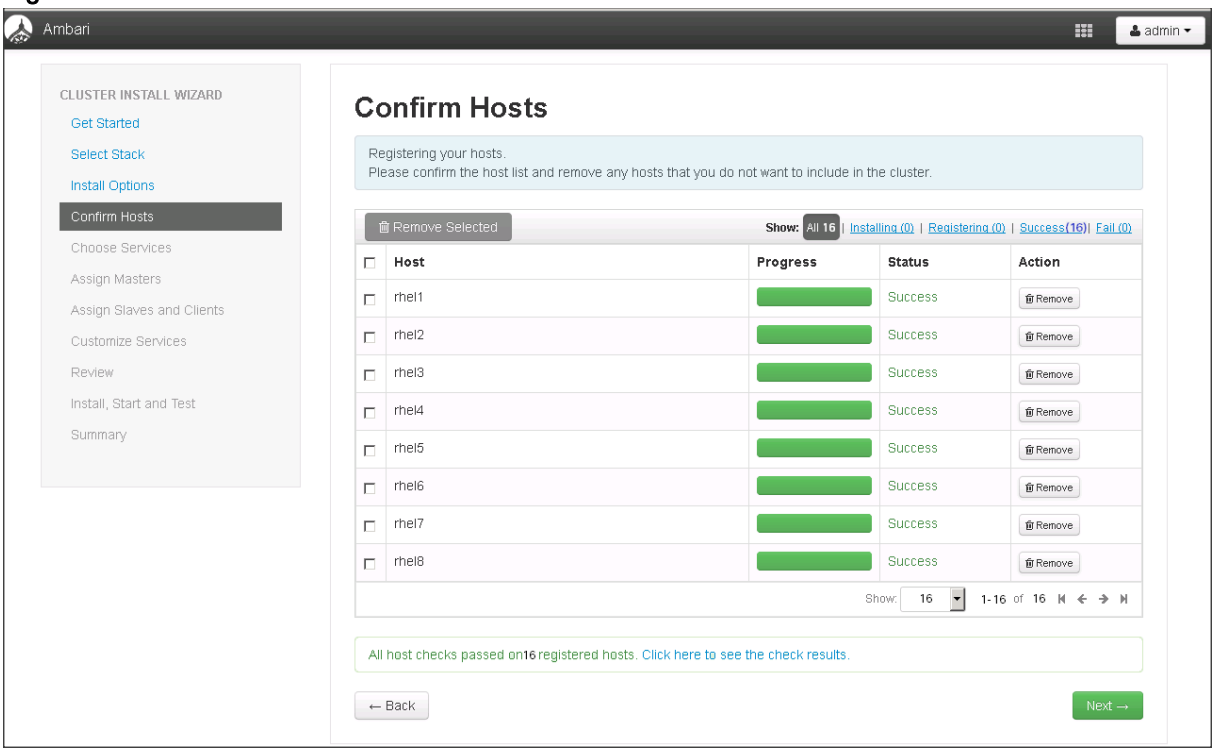
Figure 140 shows the Confirm Hosts screen This helps ensure that Ambari has located the correct hosts for the cluster and checks those hosts to make sure they have the correct directories, packages, and processes to continue the install.

1. If any host was selected in error, remove it by selecting the appropriate checkboxes and clicking the grey Remove Selected button.

2. To remove a single host, click the small white Remove button in the Action column.

3. When the list of hosts is confirmed, click Next.

Figure 140 Confirm Hosts



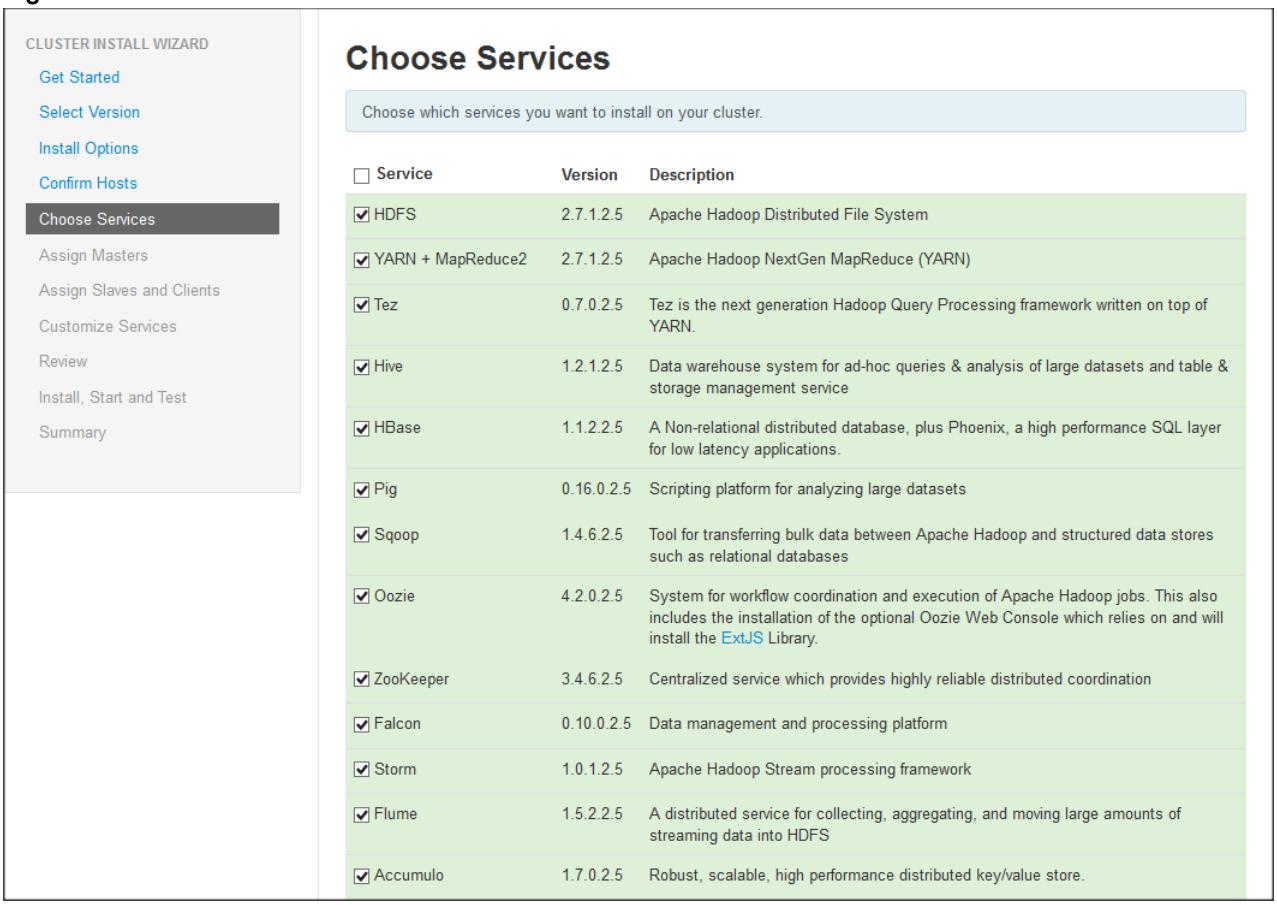
Choose Services

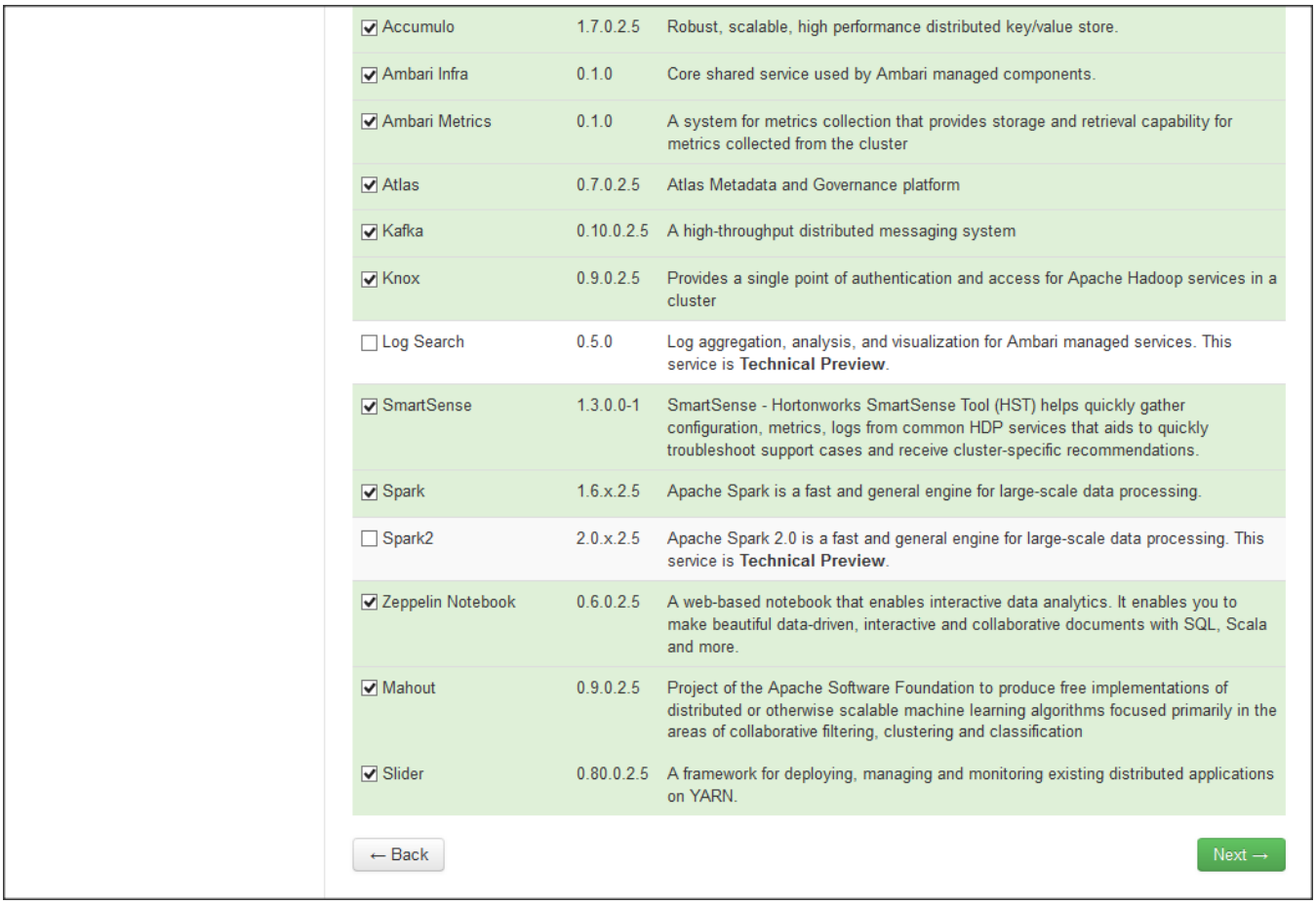
HDP is made up of a number of components. See Understand the Basics for more information. The services are listed in Figure 141 below.

1. Select all to preselect all items.

2. When you have made your selections, click Next.

Figure 141 Choose Services



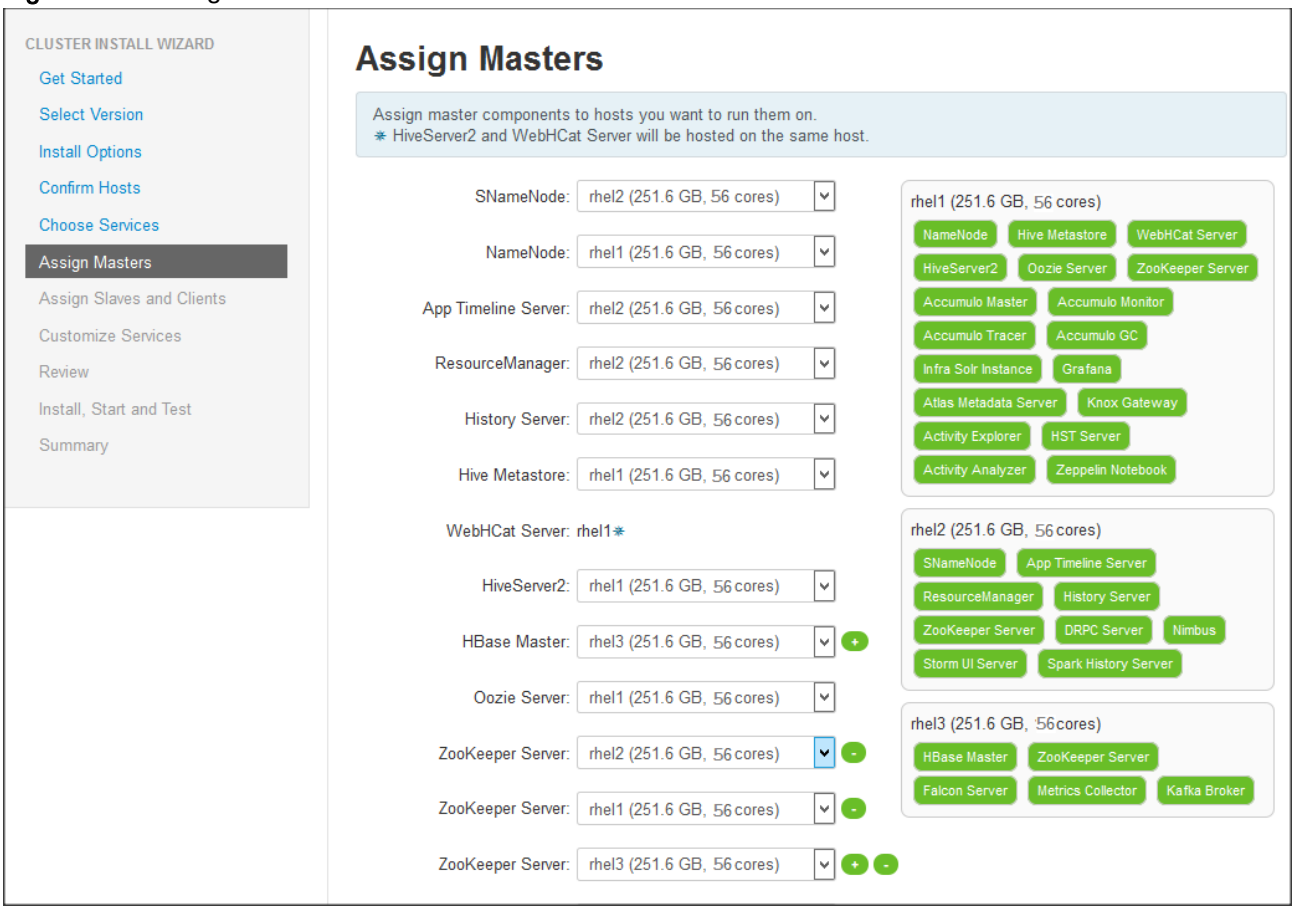


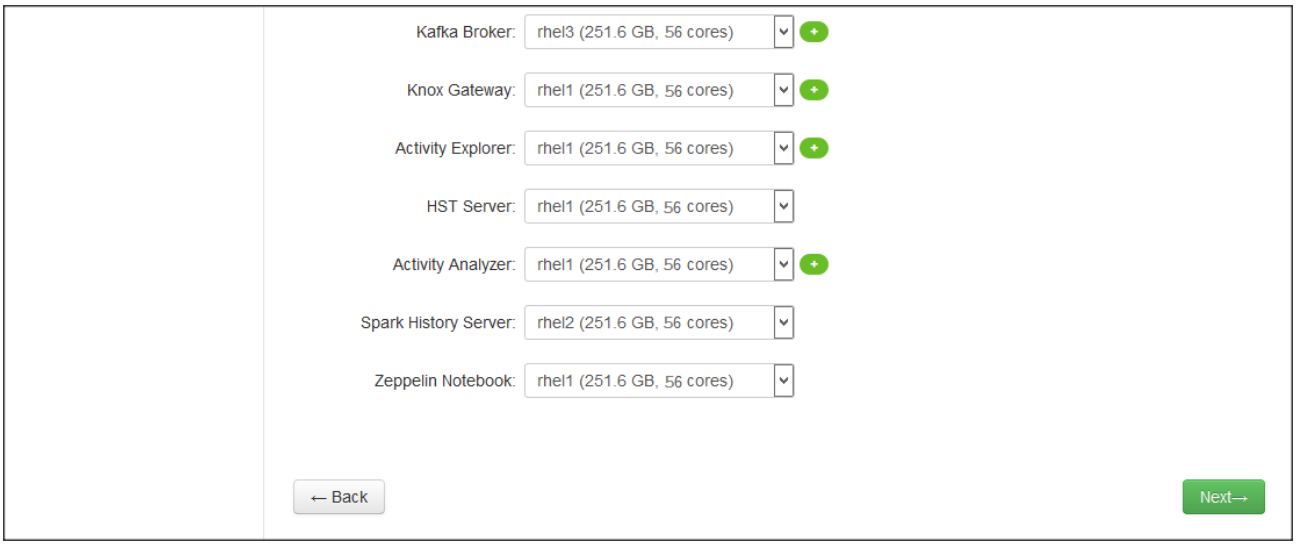
☑ Accumulo	1.7.0.2.5	Robust, scalable, high performance distributed key/value store.
☑ Ambari Infra	0.1.0	Core shared service used by Ambari managed components.
☑ Ambari Metrics	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
☑ Atlas	0.7.0.2.5	Atlas Metadata and Governance platform
☑ Kafka	0.10.0.2.5	A high-throughput distributed messaging system
☑ Knox	0.9.0.2.5	Provides a single point of authentication and access for Apache Hadoop services in a cluster
☐ Log Search	0.5.0	Log aggregation, analysis, and visualization for Ambari managed services. This service is **Technical Preview**.
☑ SmartSense	1.3.0.0-1	SmartSense - Hortonworks SmartSense Tool (HST) helps quickly gather configuration, metrics, logs from common HDP services that aids to quickly troubleshoot support cases and receive cluster-specific recommendations.
☑ Spark	1.6.x.2.5	Apache Spark is a fast and general engine for large-scale data processing.
☐ Spark2	2.0.x.2.5	Apache Spark 2.0 is a fast and general engine for large-scale data processing. This service is **Technical Preview**.
☑ Zeppelin Notebook	0.6.0.2.5	A web-based notebook that enables interactive data analytics. It enables you to make beautiful data-driven, interactive and collaborative documents with SQL, Scala and more.
☑ Mahout	0.9.0.2.5	Project of the Apache Software Foundation to produce free implementations of distributed or otherwise scalable machine learning algorithms focused primarily in the areas of collaborative filtering, clustering and classification
☑ Slider	0.80.0.2.5	A framework for deploying, managing and monitoring existing distributed applications on YARN.

Assign Masters

The Ambari install wizard attempts to assign the master nodes for various services that have been selected to appropriate hosts in the cluster, as shown in Figure 142. The right column shows the current service assignments by host, with the hostname and its number of CPU cores and amount of RAM indicated.

Figure 142 Assign Masters





3. Reconfigure the service assignments to match Table 10 shown below.

Table 10 Reconfigure the service assignments

Service Name	Host
NameNode	rhel1
SNameNode	rhel2
History Server	rhel2
App Timeline Server	rhel2
Resource Manager	rhel2
Hive Metastore	rhel1
WebHCat Server	rhel1
HiveServer2	rhel1
HBase Master	rhel3
Oozie Server	rhel1
Zookeeper	rhel1, rhel2, rhel3
Falcon Server	rhel3
DRPC Server	rhel2
Nimbus	rhel2
Storm UI Server	rhel2
Accumulo Master	rhel1
Accumulo Monitor	rhel1
Accumulo Tracer	rhel1
Accumulo GC	rhel1
Infra Solr Instance	rhel1
Grafana	rhel1
Kafka Broker	rhel1
Accumulo GC	rhel1
Atlas Metadata Server	rhel1
Knox Gateway	rhel1
Metrics Collector	rhel3
Activity Explorer	rhel1
HST server	rhel1
Activity Analyser	rhel1
Spark history server	rhel2

Service Name	Host
Zeppelin Notebook	rhel1

Note: On a small cluster (<16 nodes), consolidate all master services to run on a single node. For large clusters (> 64 nodes), deploy master services across 3 nodes.

4. Click Next.

Assign Slaves and Clients

The Ambari install wizard attempts to assign the slave components (DataNodes, NFSGateway, NodeManager, RegionServers, Phoenix Query Server, Supervisor, Flume, Accumulo TServer, Spark Thrift Server and Client) to appropriate hosts in the cluster as shown in Figure 143.

1. Reconfigure the service assignment to match the values shown in Table 11 below:

2. Assign DataNode, NodeManager, RegionServer, Supervisor and Flume on nodes rhel3- rhel19.

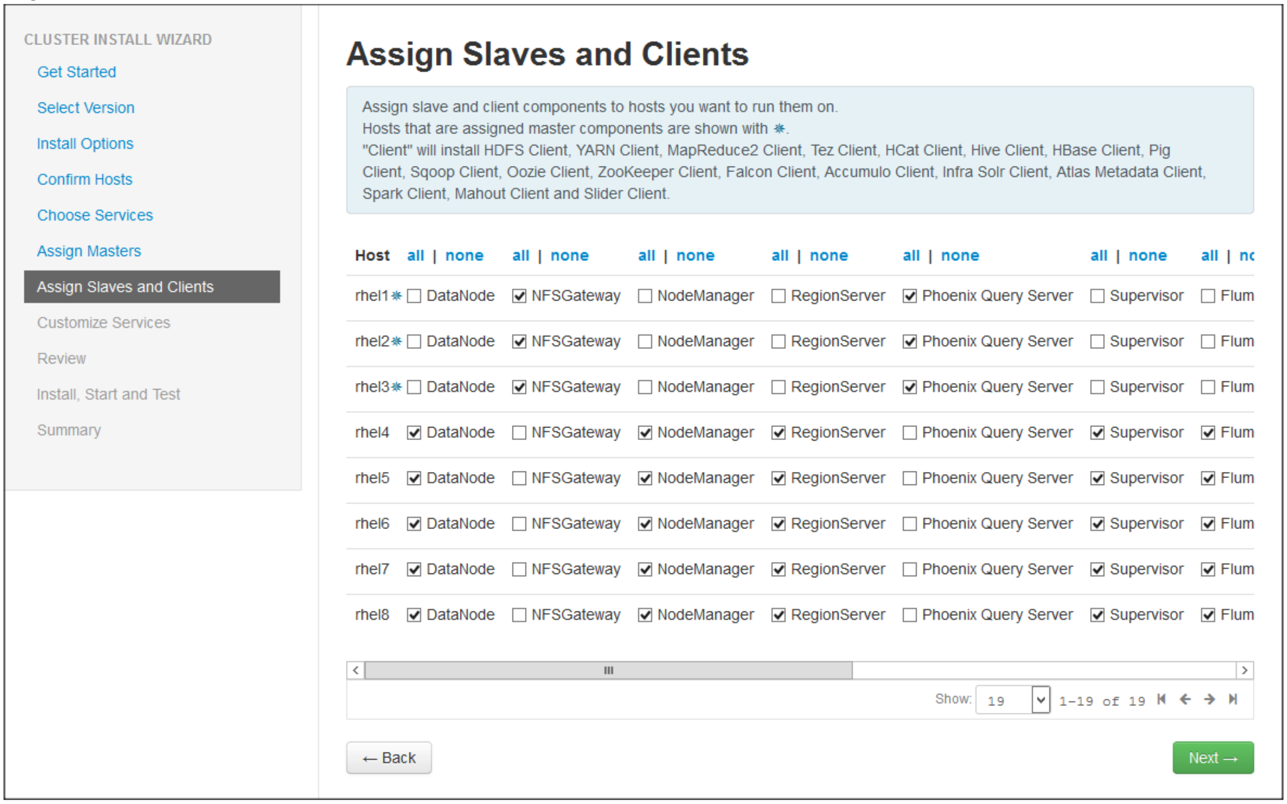
3. Assign Client to all nodes.

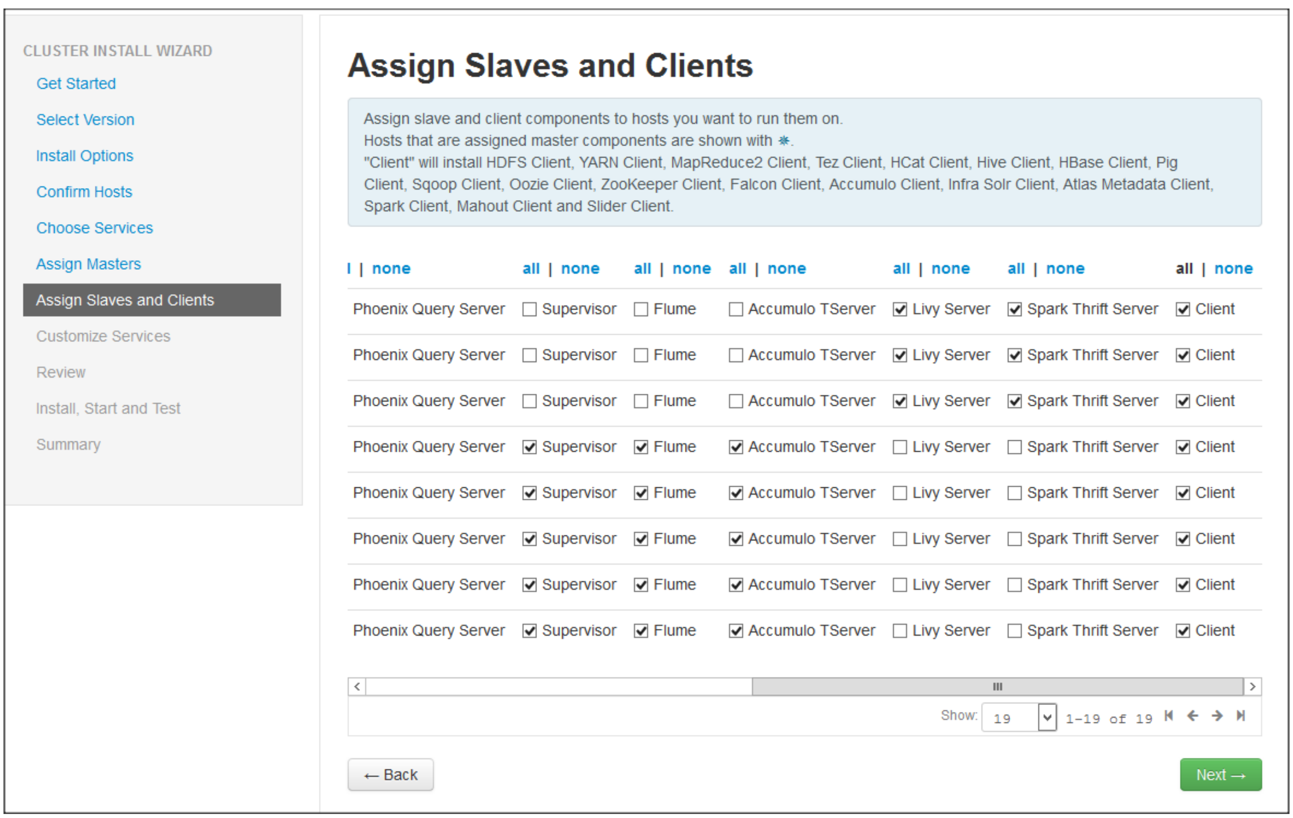
4. Click the Next button.

Table 11 Services and Hostnames

Client Service Name	Host
DataNode	Rhel4-rhel19
NFSGateway	rhel1
NodeManager	rhel3-rhel19
RegionServer	rhel3-rhel19
Phoenix Query Server	rhel1
Supervisor	rhel3-rhel19
Flume	rhel3-rhel19
Accumulo TServer	rhel3-rhel19
Spark Thrift Server	rhel1
Client	All nodes, rhel1-rhel19

Figure 143 Assign Slaves and Clients





Customize Services

This section as shown in Figure 144 displays the tabs that manage configuration settings for Hadoop components. The wizard attempts to set reasonable defaults for each of the options here, but this can be modified to meet specific requirements. The following sections provide configuration guidance that should be refined to meet specific use case requirements.

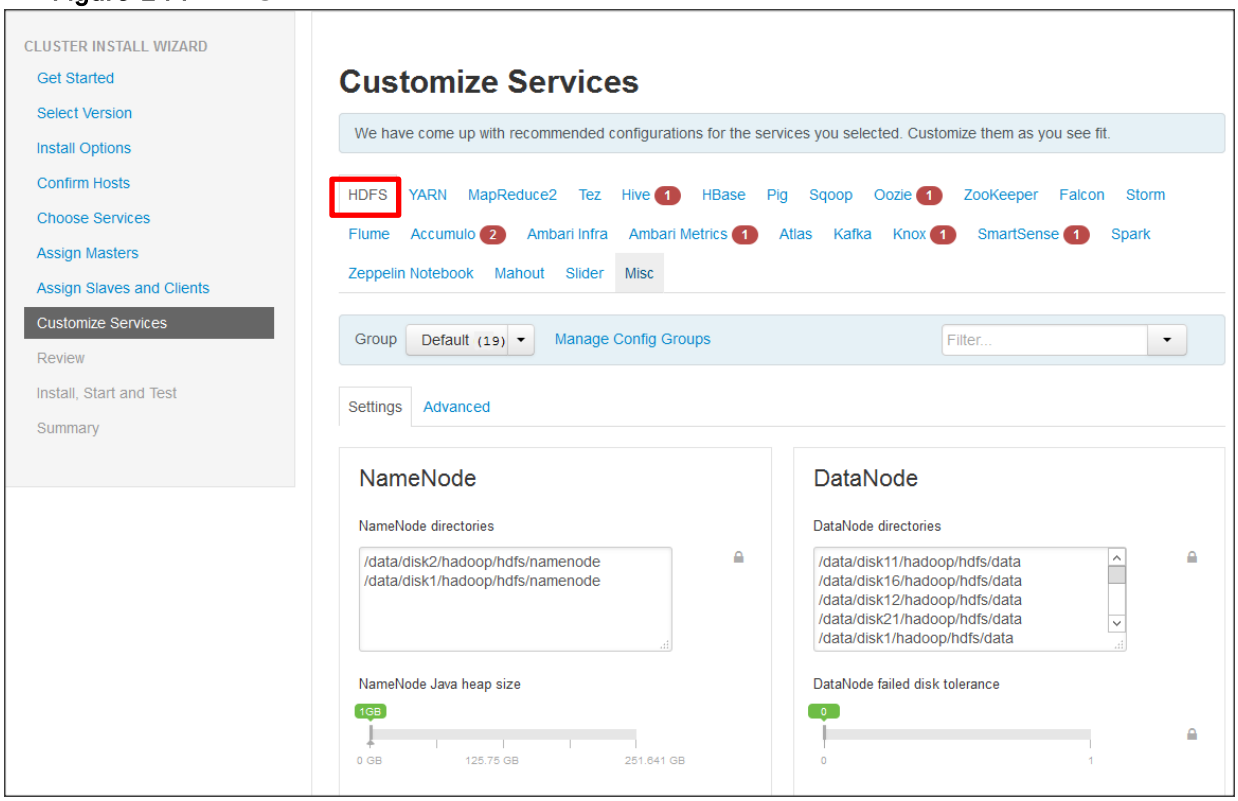
The following changes are to be made:

- Memory and service level settings for each component and service level tuning.
- Customize the log locations of all the components to ensure growing logs do not cause the SSDs to run out of space.

HDFS

1. In Ambari, choose the HDFS Service tab. (Figure 144) and use the “Search” box on top to filter for the properties mentioned in Table 12 and update their values.

Figure 144 HDFS



HDFS JVM Settings

1. Update the following HDFS configurations in Ambari.

Table 12 HDFS Configurations in Ambari

Property Name	Value
NameNode Java Heap Size	4096

Hadoop maximum Java heap size	4096
DataNode maximum Java heap size	4096
Datanode Volumes Failure Toleration	3

YARN

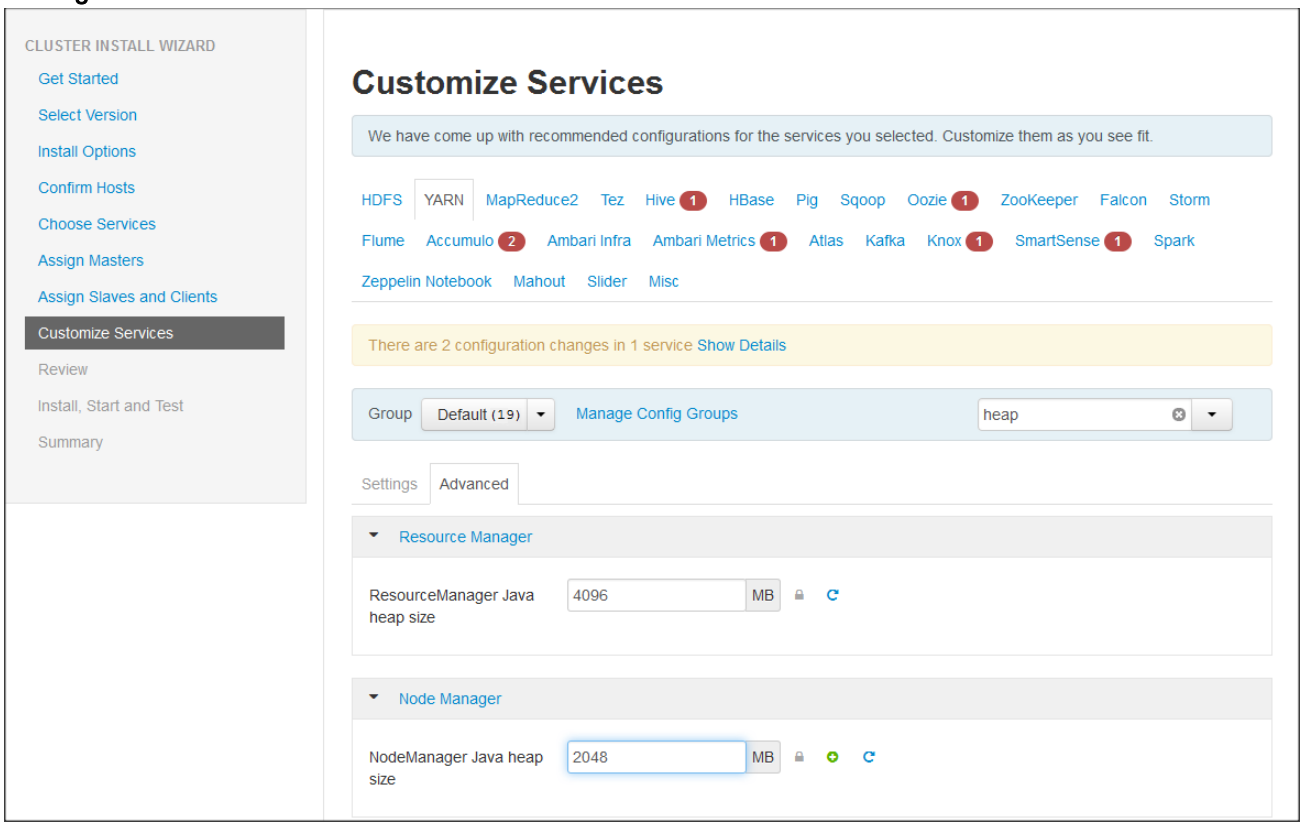
In Ambari, choose the YARN Service from the tab as shown in Table 13, and use the “Search” box on top to filter for the properties mentioned in Figure 145 below to update their values.

Update the following YARN configurations.

Table 13 YARN Configuration Values

Property Name	Value
ResourceManager Java heap size	4096
NodeManager Java heap size	2048
yarn.nodemanager.resource.memory-mb	184320
YARN Java heap size	4096
yarn.scheduler.minimum-allocation-mb	4096
yarn.scheduler.maximum-allocation-mb	184320

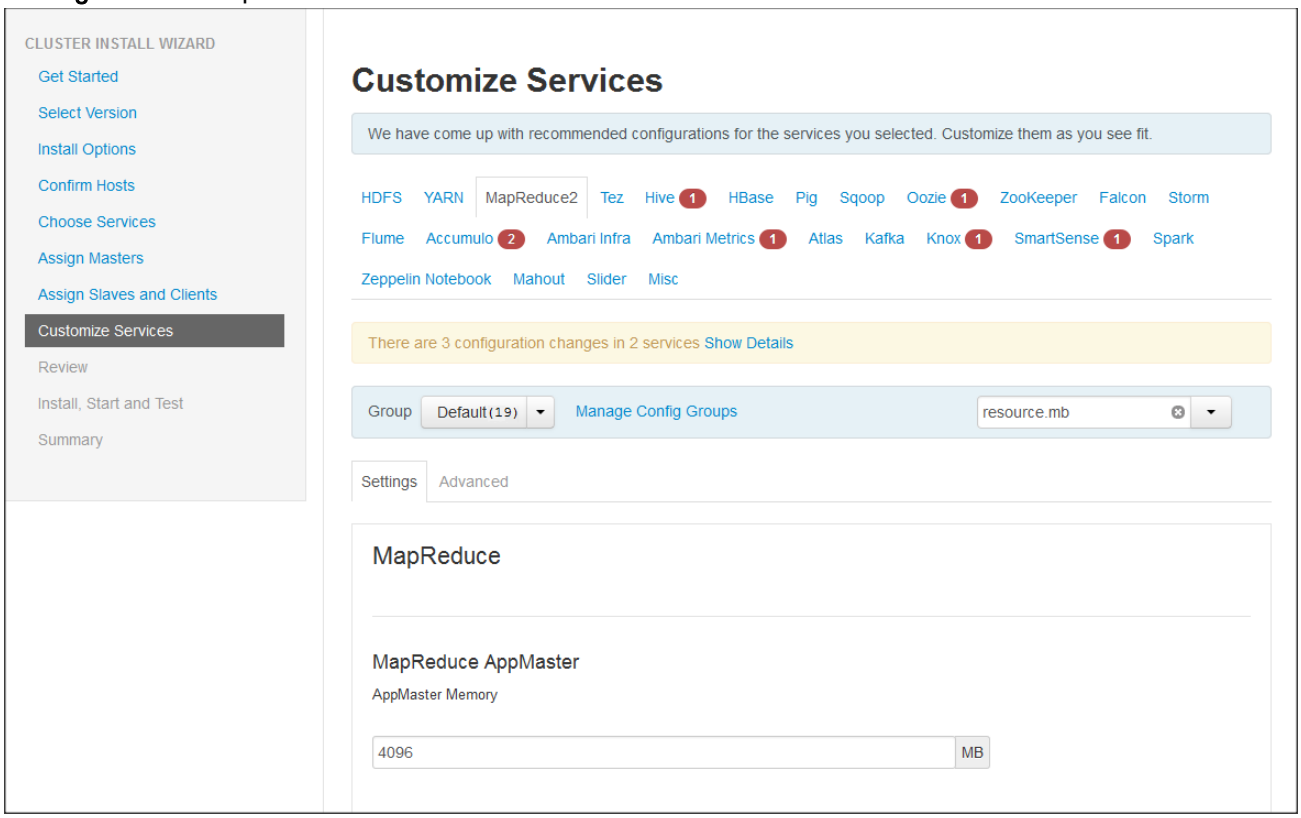
Figure 145 Customize Services Screen



MapReduce2

Figure 146 shows the MapReduce2 Tab.

Figure 146 MapReduce Tab



1. In Ambari, choose the MapReduce Service tab and use the “Search” box on top to filter for the properties mentioned in Table 14 and update their values.

2. Update the following MapReduce configurations.

Table 14 MapReduce Configurations

Property Name	Value
Default virtual memory for a job's map-task	4096
Default virtual memory for a job's reduce-task	8192
Map-side sort buffer memory	1638
yarn.app.mapreduce.am.resource.mb	4096
mapreduce.map.java.opts	-Xmx3276m
mapreduce.reduce.java.opts	-Xmx6552m
yarn.app.mapreduce.am.command-opts	-Xmx6552m

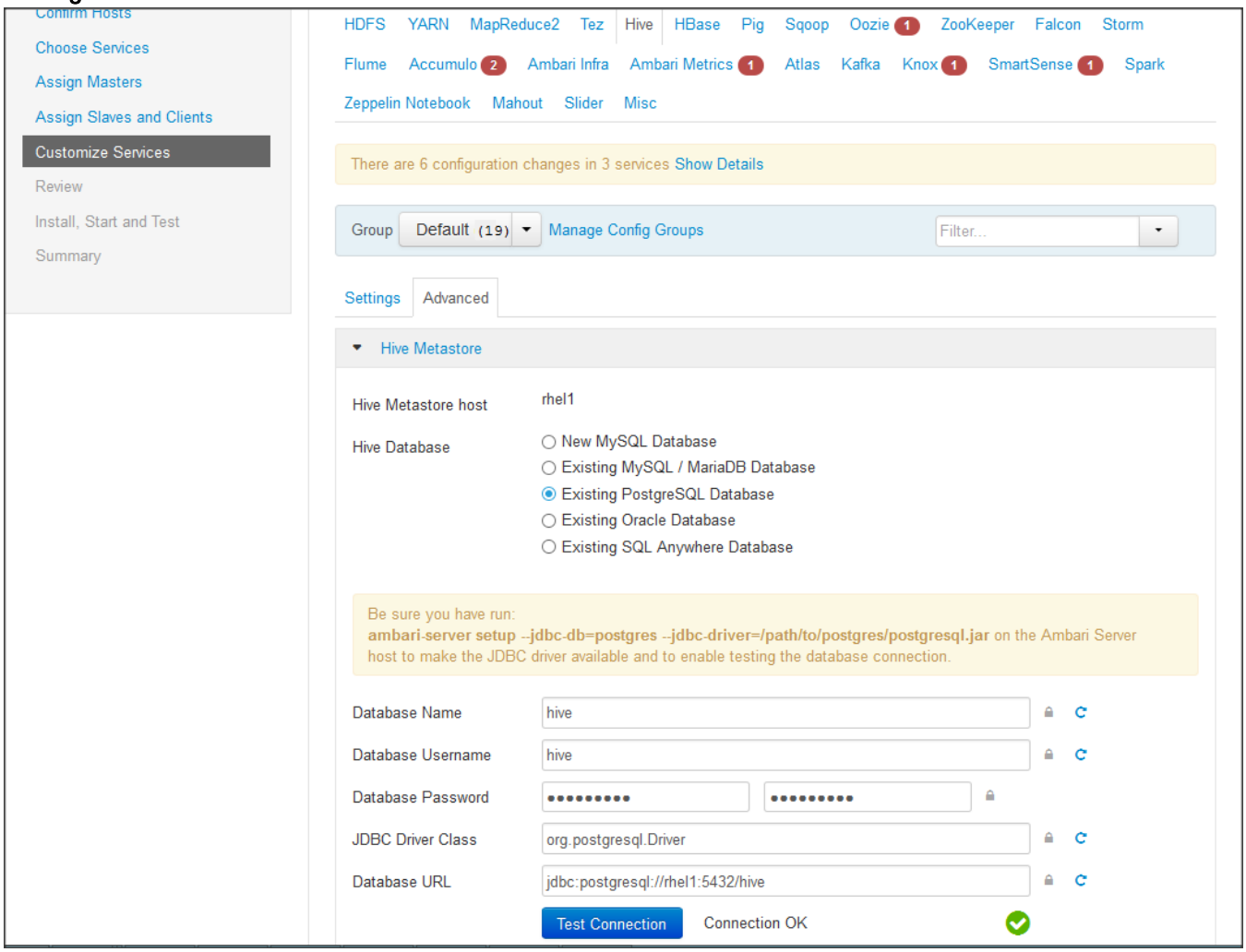
Tez

No changes are required.

Hive

1. Choose Hive Service from the tab, as shown in Figure 147.
2. Select the advanced tab and make the changes below:
3. Select Existing PostgreSQL Database.
4. Database Name hive.
5. Database Username hive.
6. Enter the Hive database password as per organizational policy.
7. Database password Cisco_123.
8. Please test connection.

Figure 147 Hive Tab



HBase

In Ambari, choose HBase Service from the tab (Figure 148) and use the “Search” box on top to filter for the properties mentioned in Table 15 to update their values.

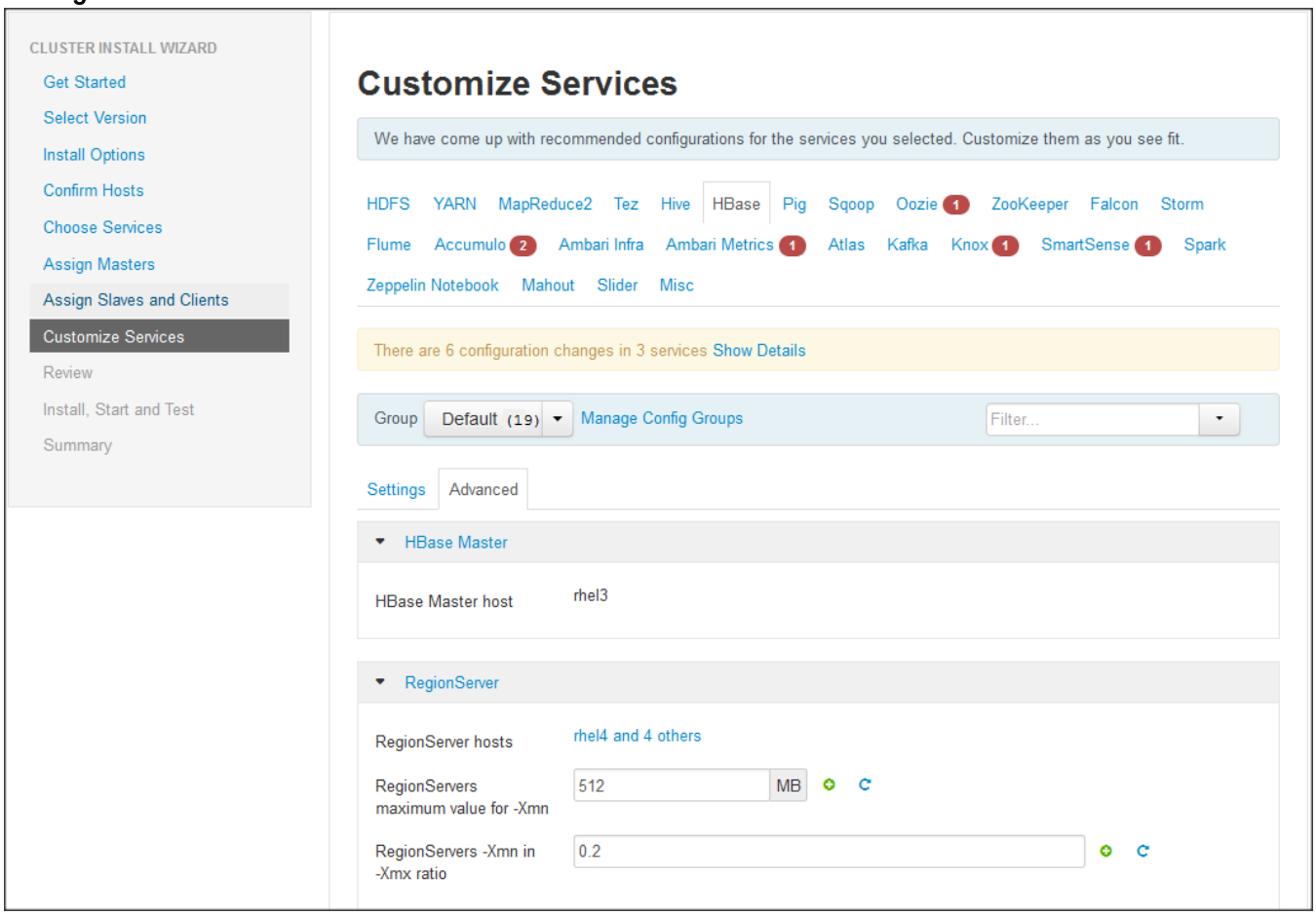
1. Update the following HBase configurations:

Table 15 HBASE Configurations

Property Name	Value
HBase Master Maximum Java Heap Size	4096
HBase RegionServers Maximum Java Heap Size	16384

Note: If you are not running HBase, keep the default value of 1024 for Java Heap size for HBase RegionServers and HBase Master

Figure 148 HBase



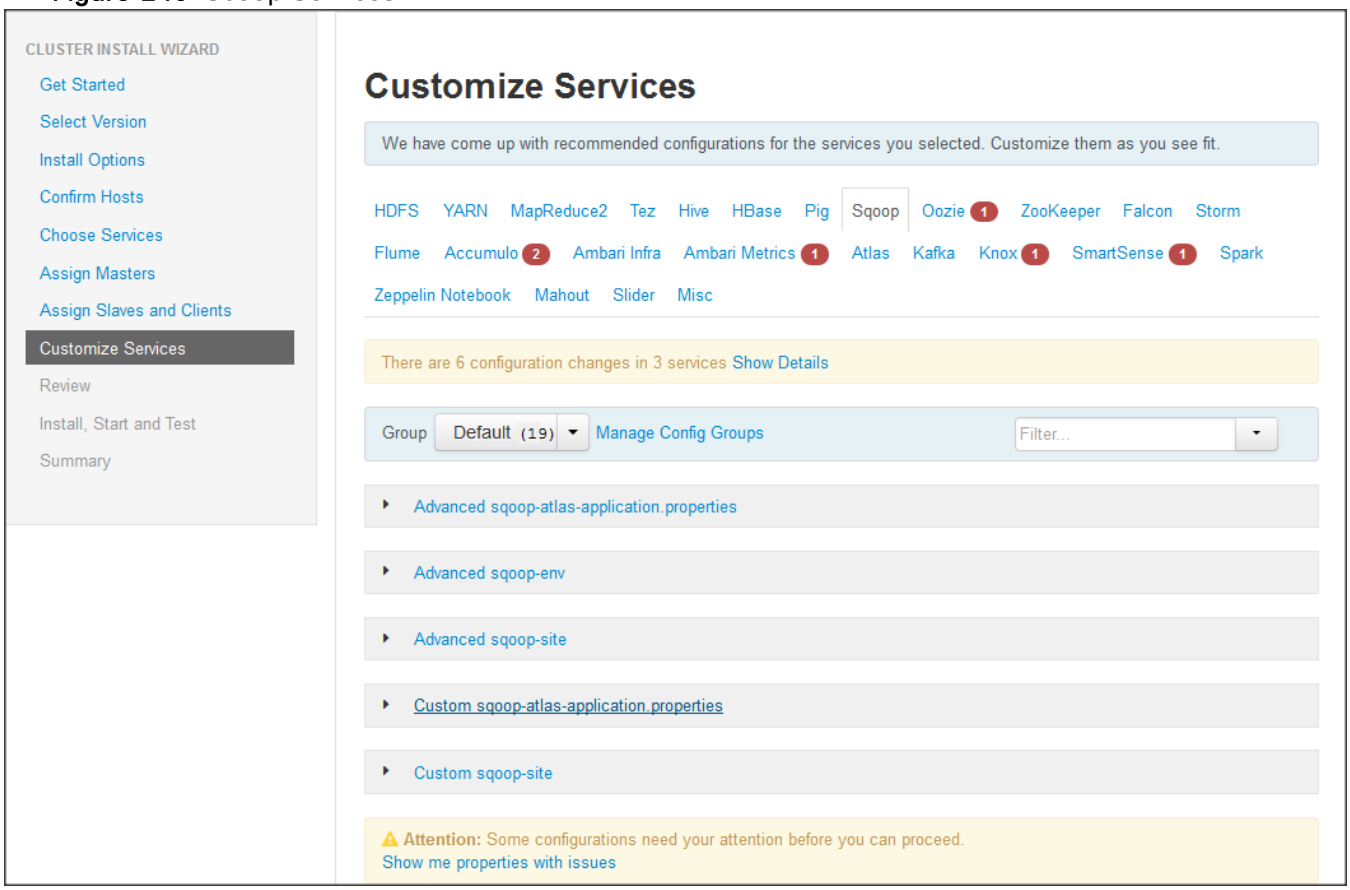
Pig

No changes are required.

Sqoop

No changes are required.

Figure 149 Scoop Services

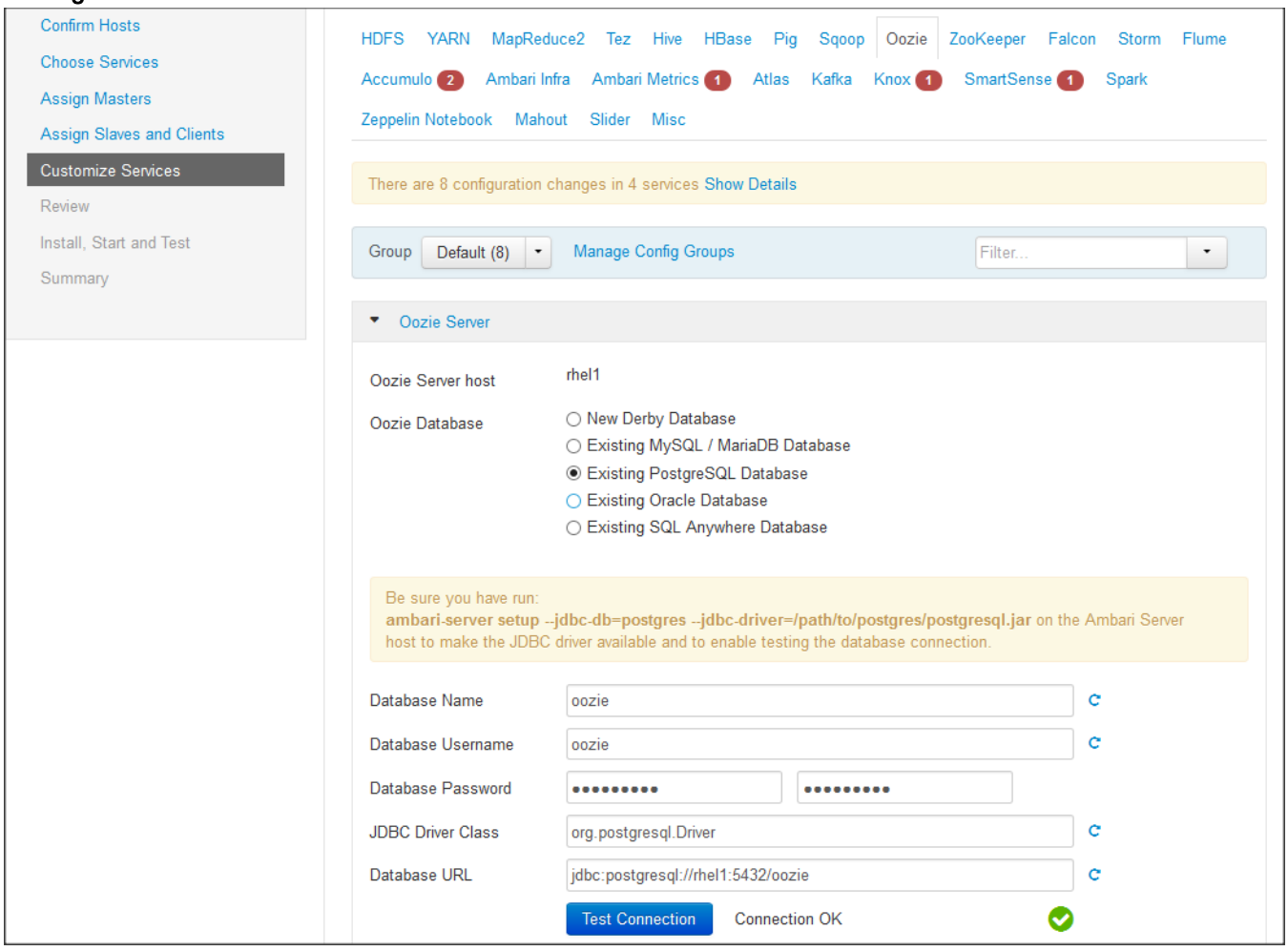


Oozie

Similarly, under the Oozie tab, (Figure 150), change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

1. Select Existing PostgreSQL Database.
2. Database Name oozie.
3. Database Username oozie. Enter the oozie database password as per organizational policy.
4. Database password is Cisco_123.
5. Please test the connection.

Figure 150 Oozie Tab



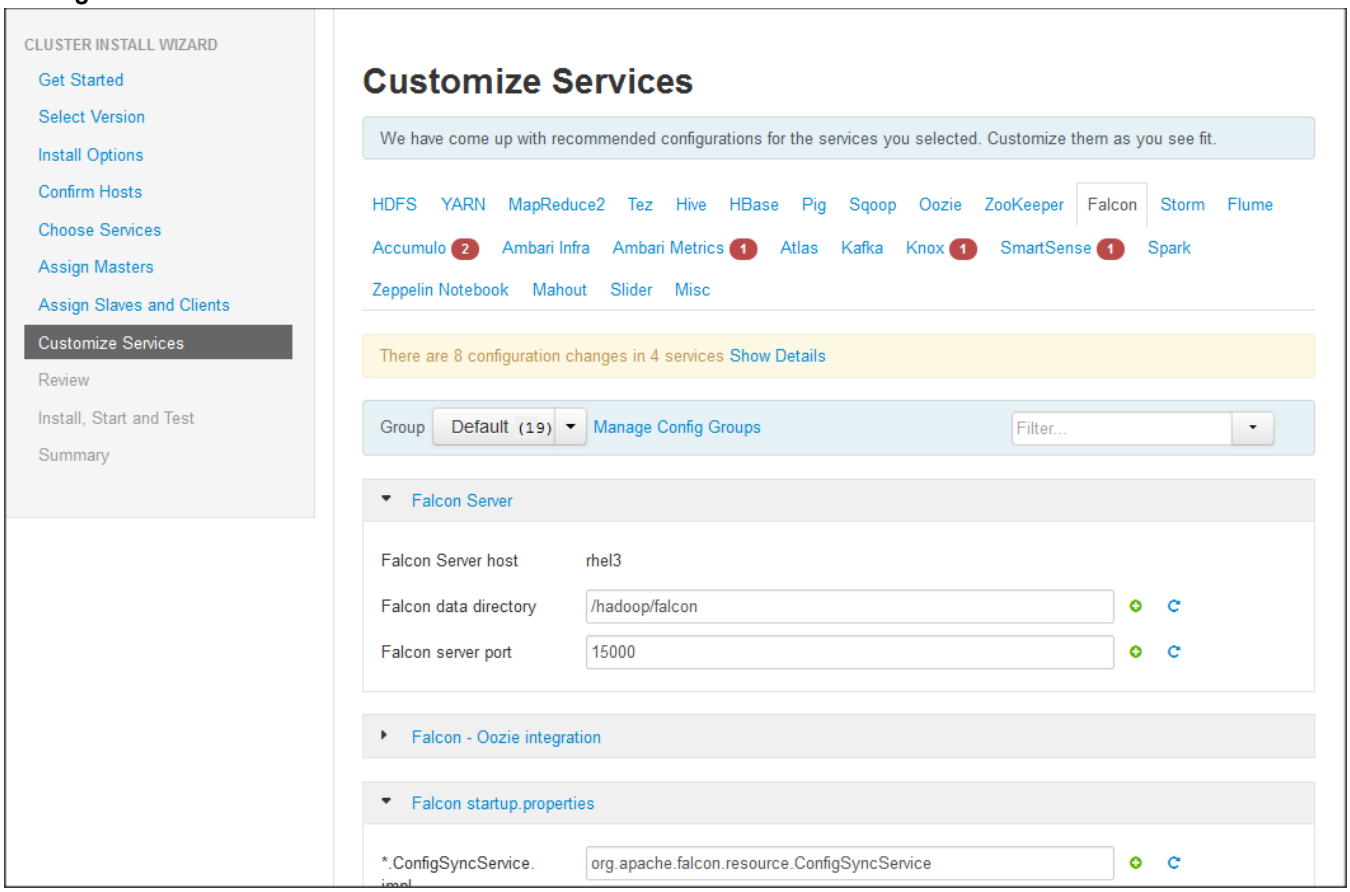
Zookeeper

No changes required

Falcon

No changes required

Figure 151 Falcon Tab



Storm

No changes required

Flume

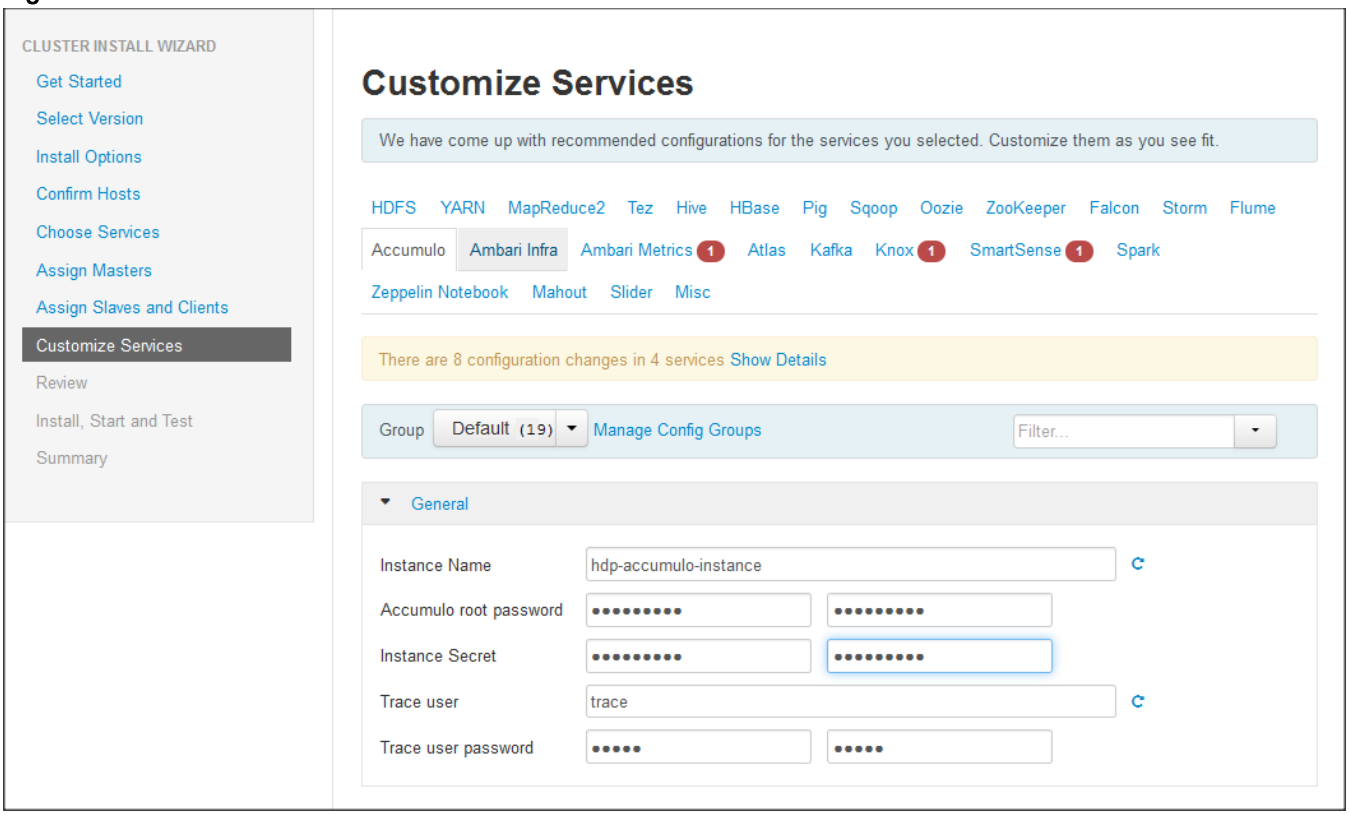
No changes required

Accumulo

Choose Accumulo Service from the tab and expand the General tab and make the changes below:

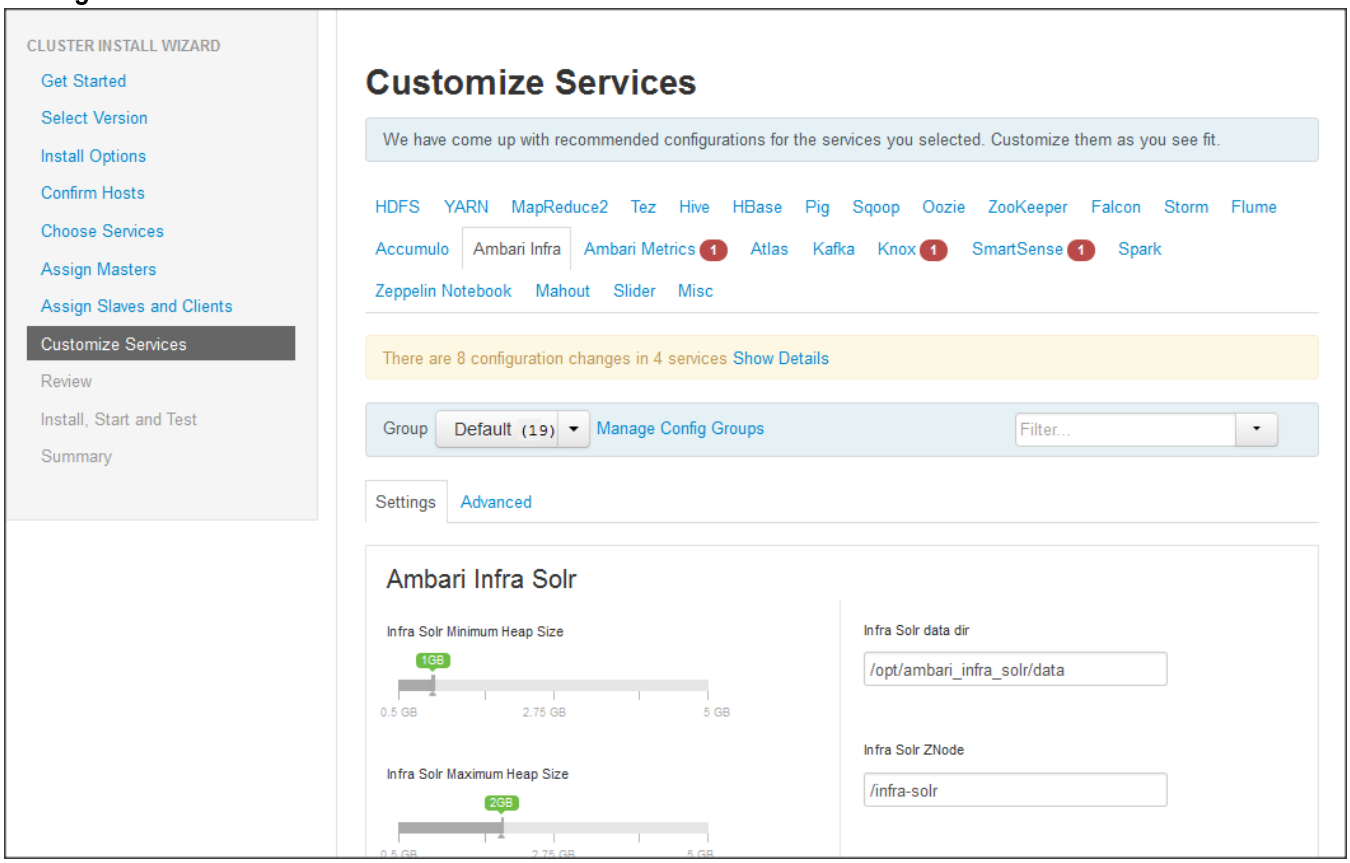
1. Enter the Accumulo root password as per organizational policy. (Figure 152)
2. Enter the Accumulo instance Secret password as per organizational policy.

Figure 152 Accumulo



Ambari Infra

Figure 153 Ambari Infra

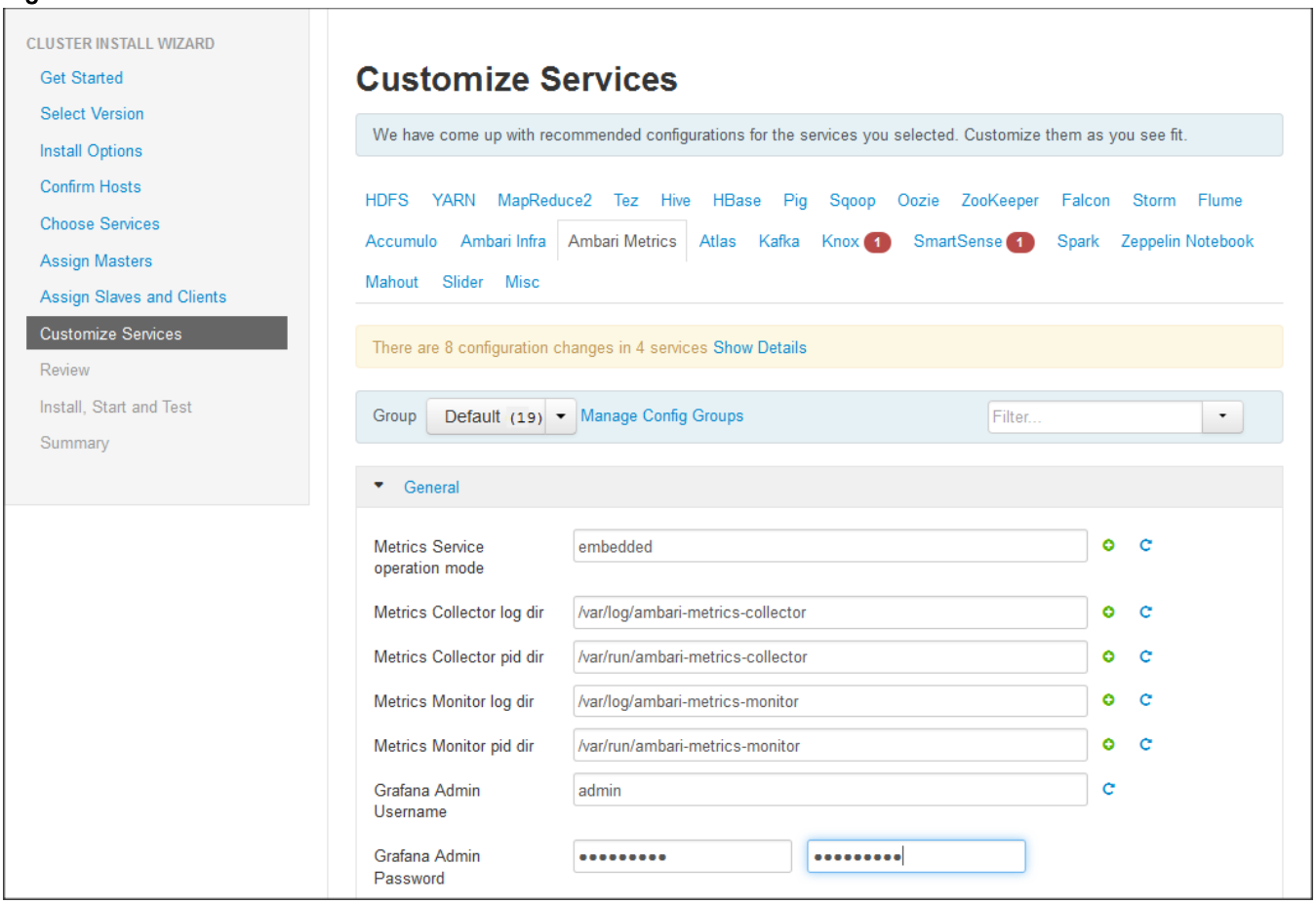


Ambari Metrics

1. Choose the Ambari Metrics Service, (Figure 154), from the tab and expand the general tab and make the changes below:

2. Enter the Grafana Admin password as per organizational policy.

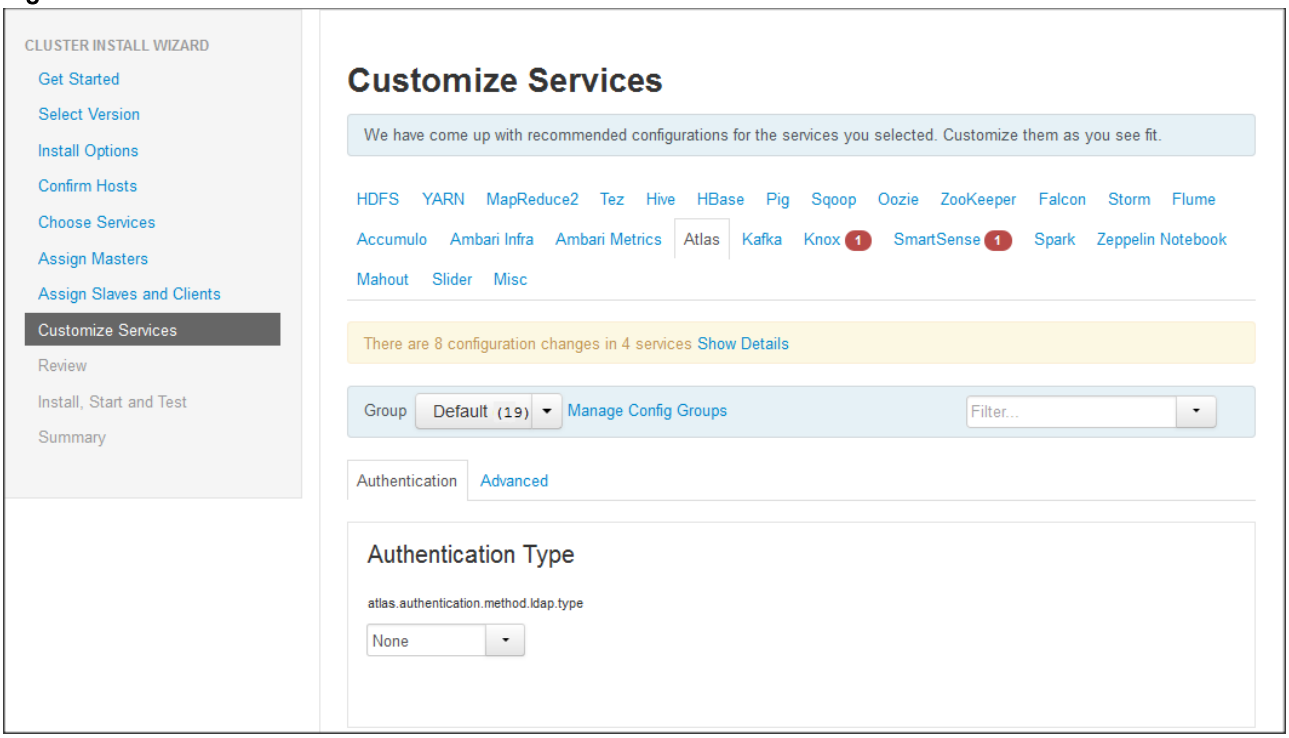
Figure 154 Ambari Metrics



Atlas

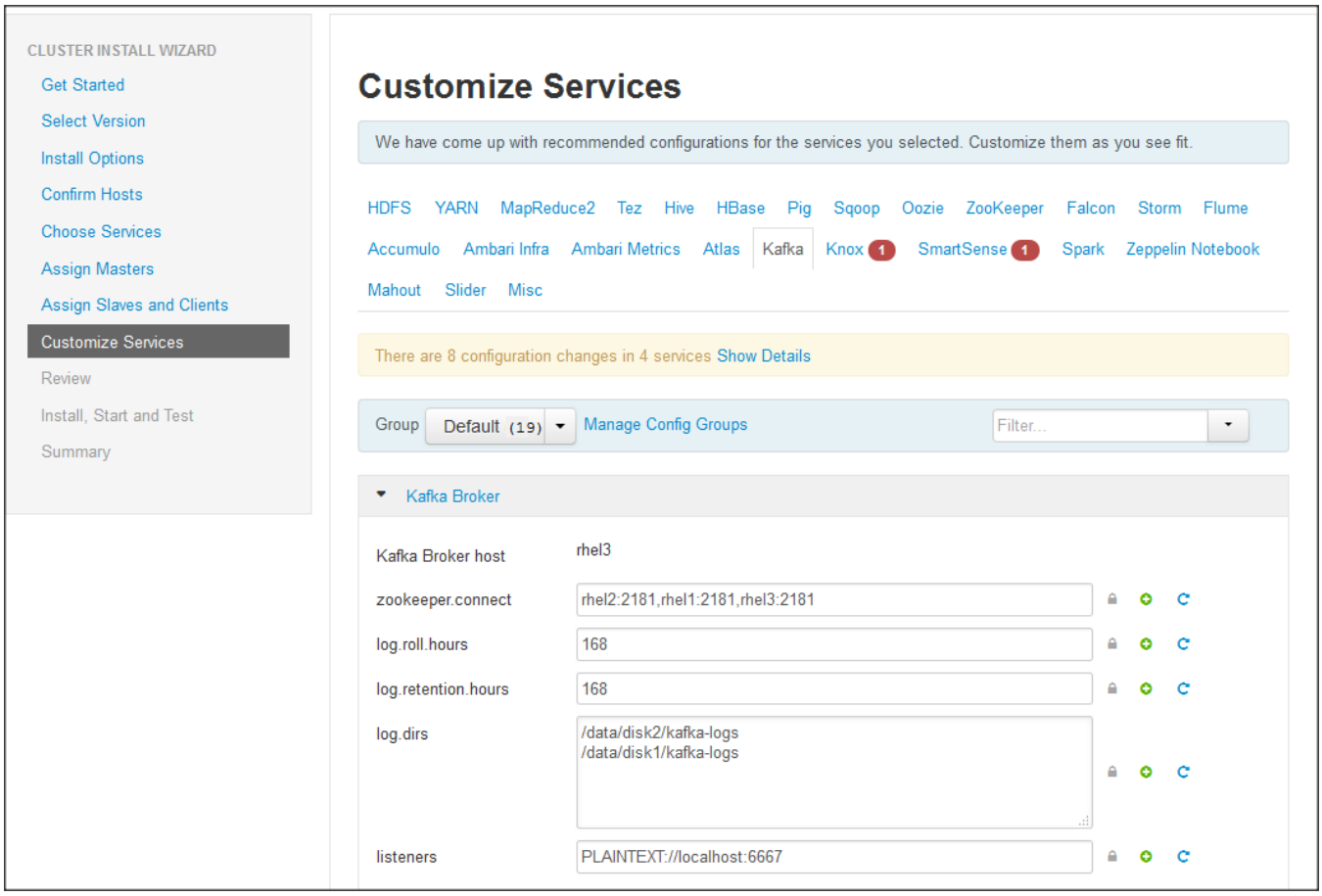
Under the Atlas tab, (Figure 155), change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

Figure 155 Atlas Tab



Kafka

1. Under the Kafka tab (Error! Reference source not found.), No change required.



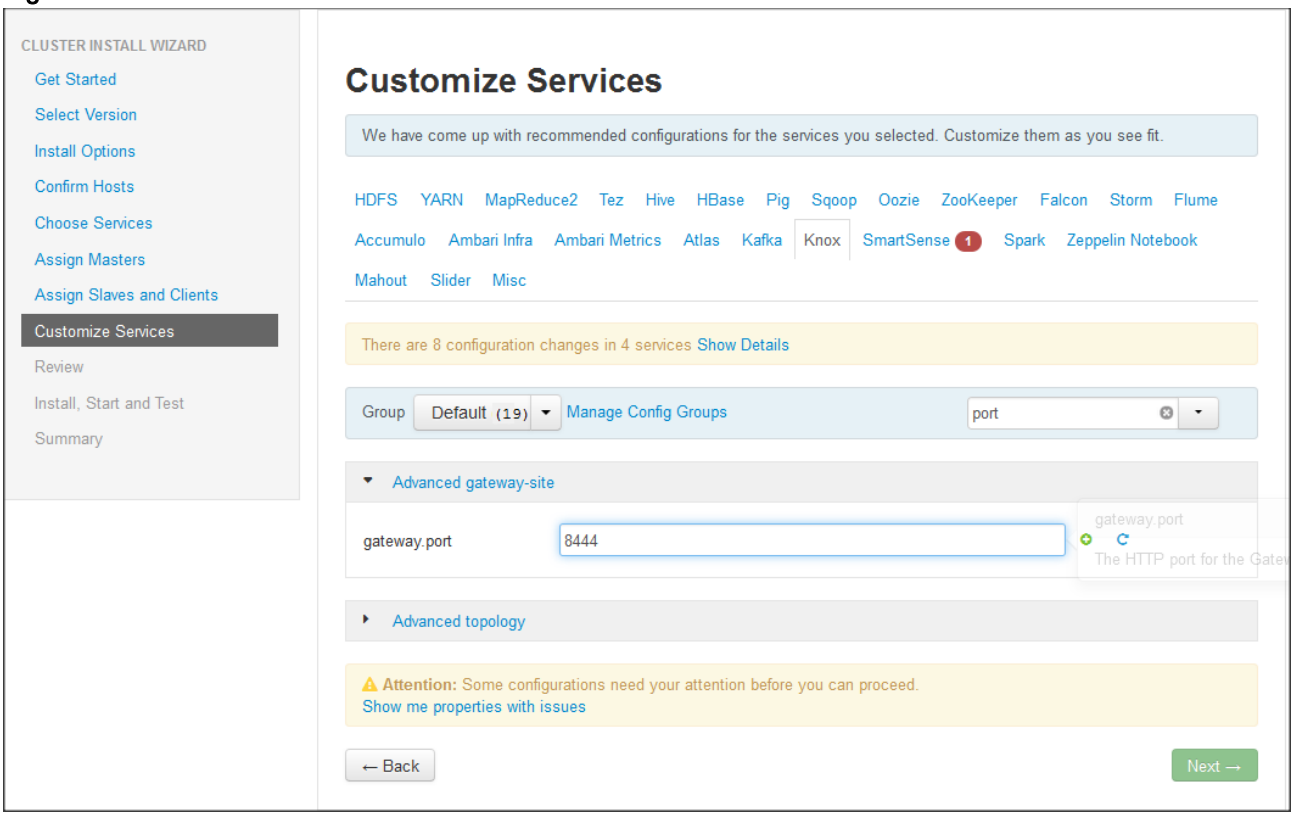
Knox

1. Choose Knox Service, (Figure 156), from the tab and expand the Knox gateway tab and make the changes below:

2. Enter the Knox Master Secret password as per organizational policy.

3. For Knox, change the gateway port to 8444 to ensure no conflicts with local HTTP server.

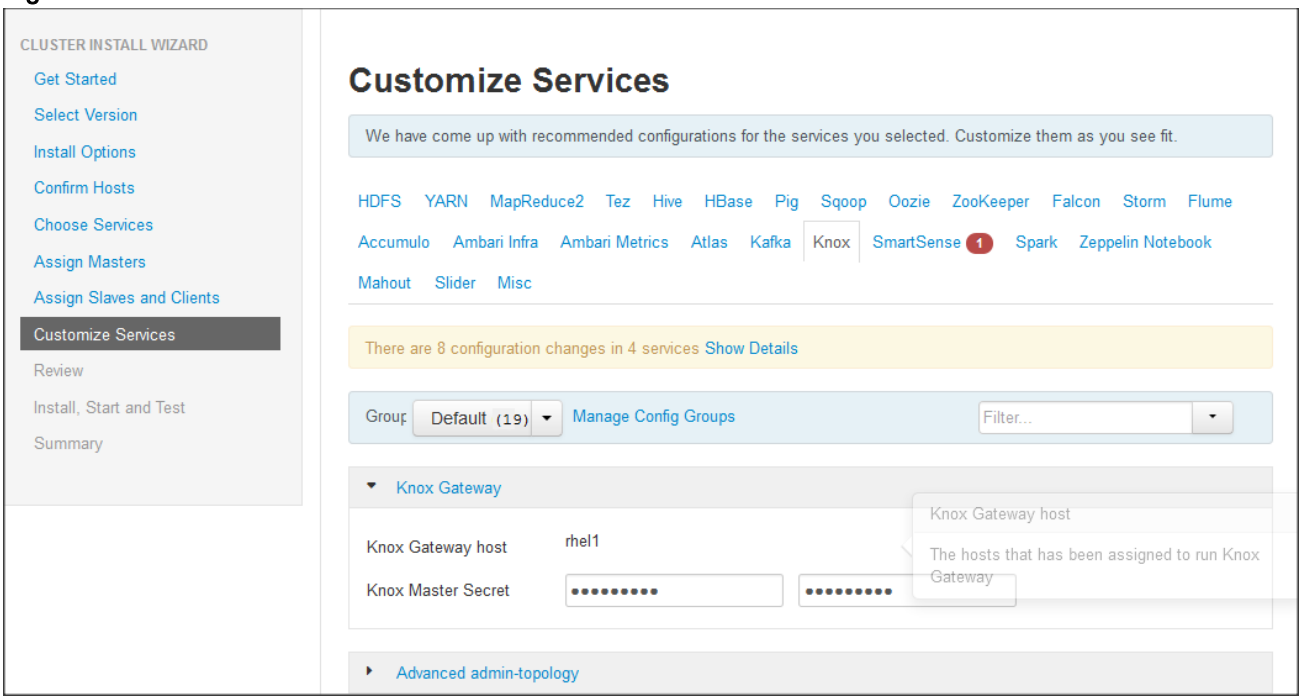
Figure 156 Knox Service



SmartSense

Figure 157 shows the SmartSense tab. This requires the Hortonworks support subscription. Subscribers can populate the properties below.

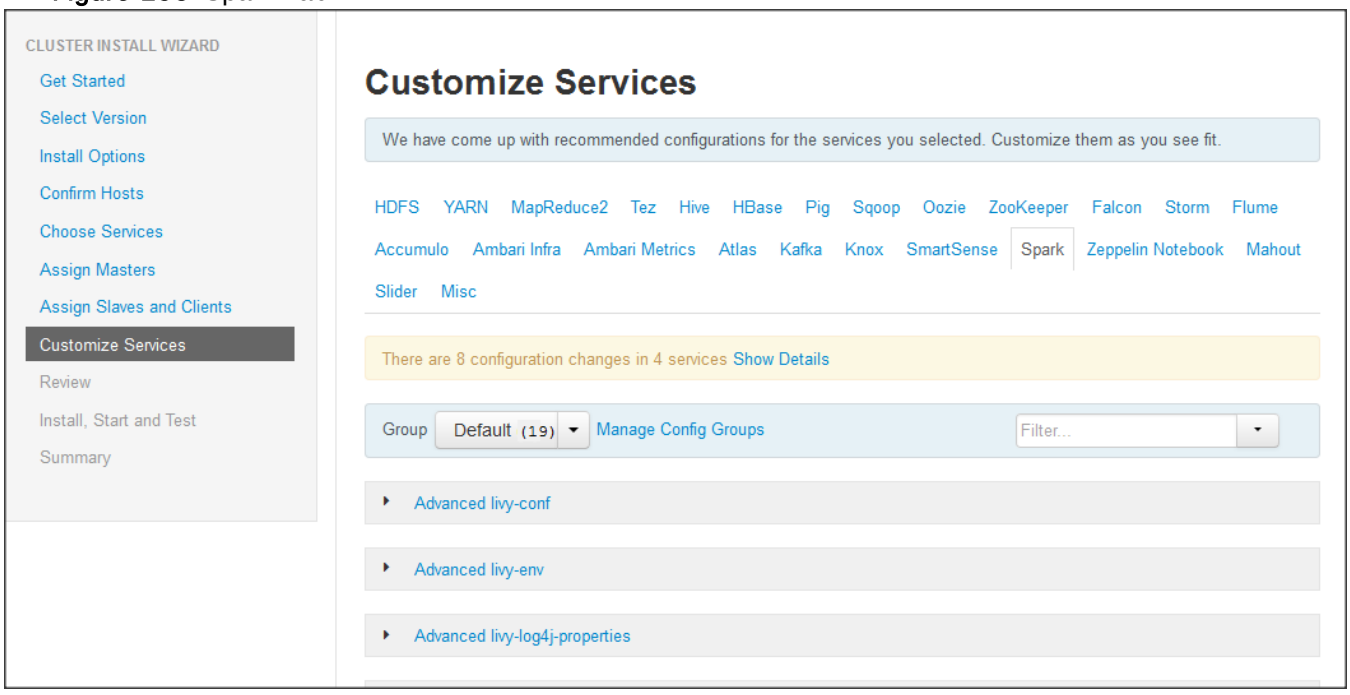
Figure 157 SmartSense



Spark

1. Select the Spark tab, (Figure 158), change the default log location by finding the Log Dir property and modifying the `/var` prefix to `/data/disk1`.

Figure 158 Spark Tab



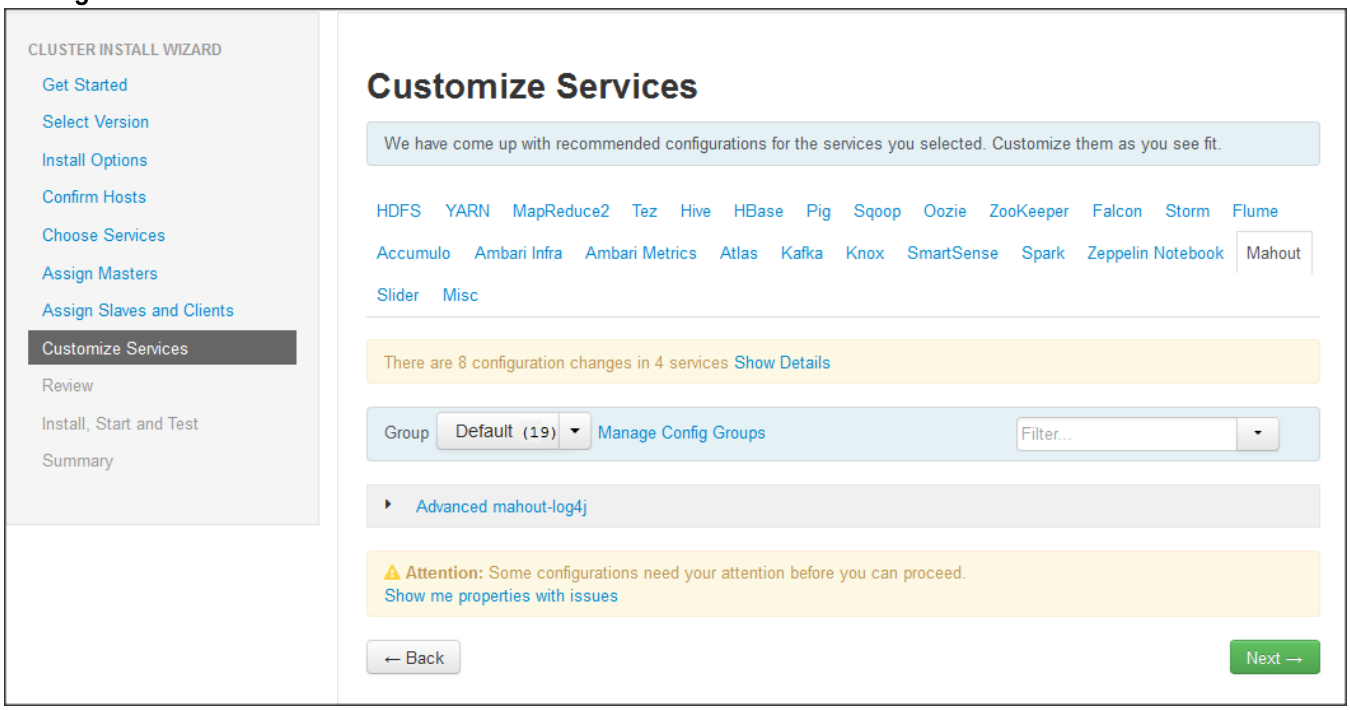
Zeppelin Notebook

No changes required

Mahout

No changes are required.

Figure 159 Mahout Tab



Slider

No changes are required.

Misc

No changes are required.

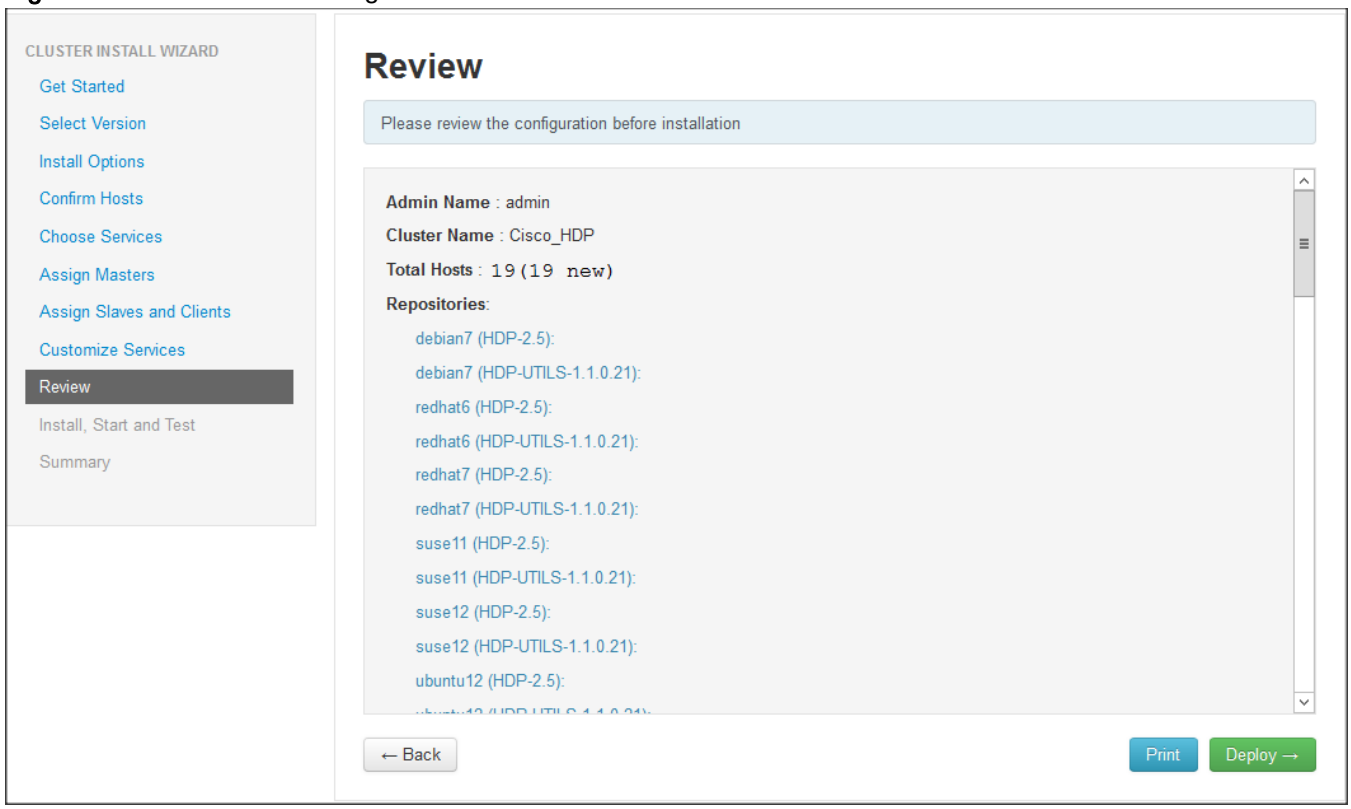
Review

The assignments that have been made are displayed, (Figure 160). Check to ensure everything is correct before clicking on the Deploy button. If any changes are to be made, use the left navigation bar to return to the appropriate screen.

Deploy

1. Once the review is complete, click the Deploy button.

Figure 160 Review the Configuration



The progress of the install is shown on the screen as shown in Figure 161. Each component is installed and started and a simple test is run on the component. The next screen displays the overall status of the install in the progress bar at the top of the screen and a host-by-host status in the main section.

2. To see specific information on what tasks have been completed per host, click the link in the Message column for the appropriate host.

3. In the Tasks pop-up, click the individual task to see the related log files.

4. Select filter conditions by using the Show dropdown list.

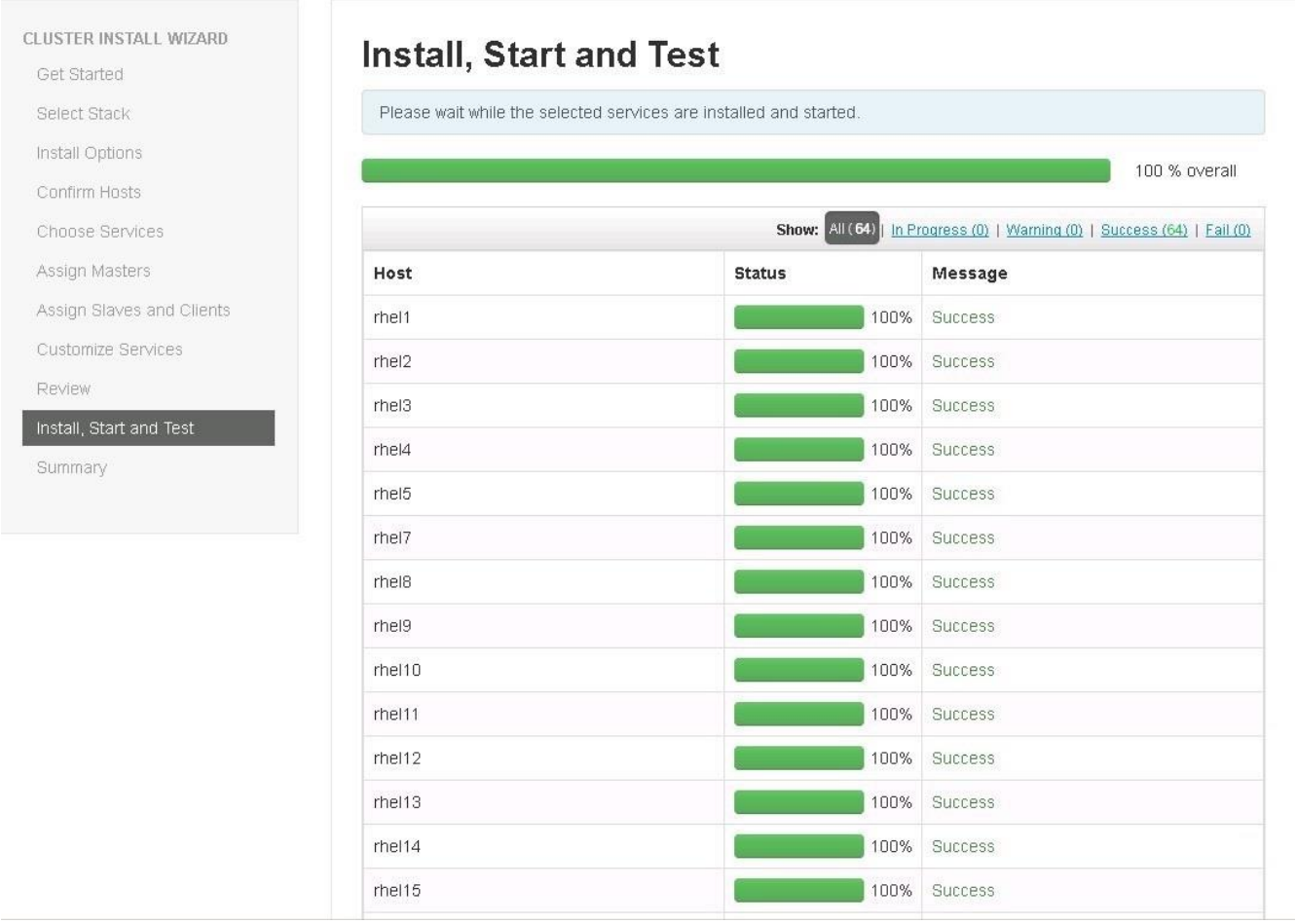
5. To see a larger version of the log contents, click the Open icon or to copy the contents to the clipboard, use the Copy icon.

   Depending on which components are installing, the entire process may take 10 or more minutes.

6. When successfully installed and started the service appears, click Next.

Figure 161 displays the install progress screen.

Figure 161 Install Progress Screen

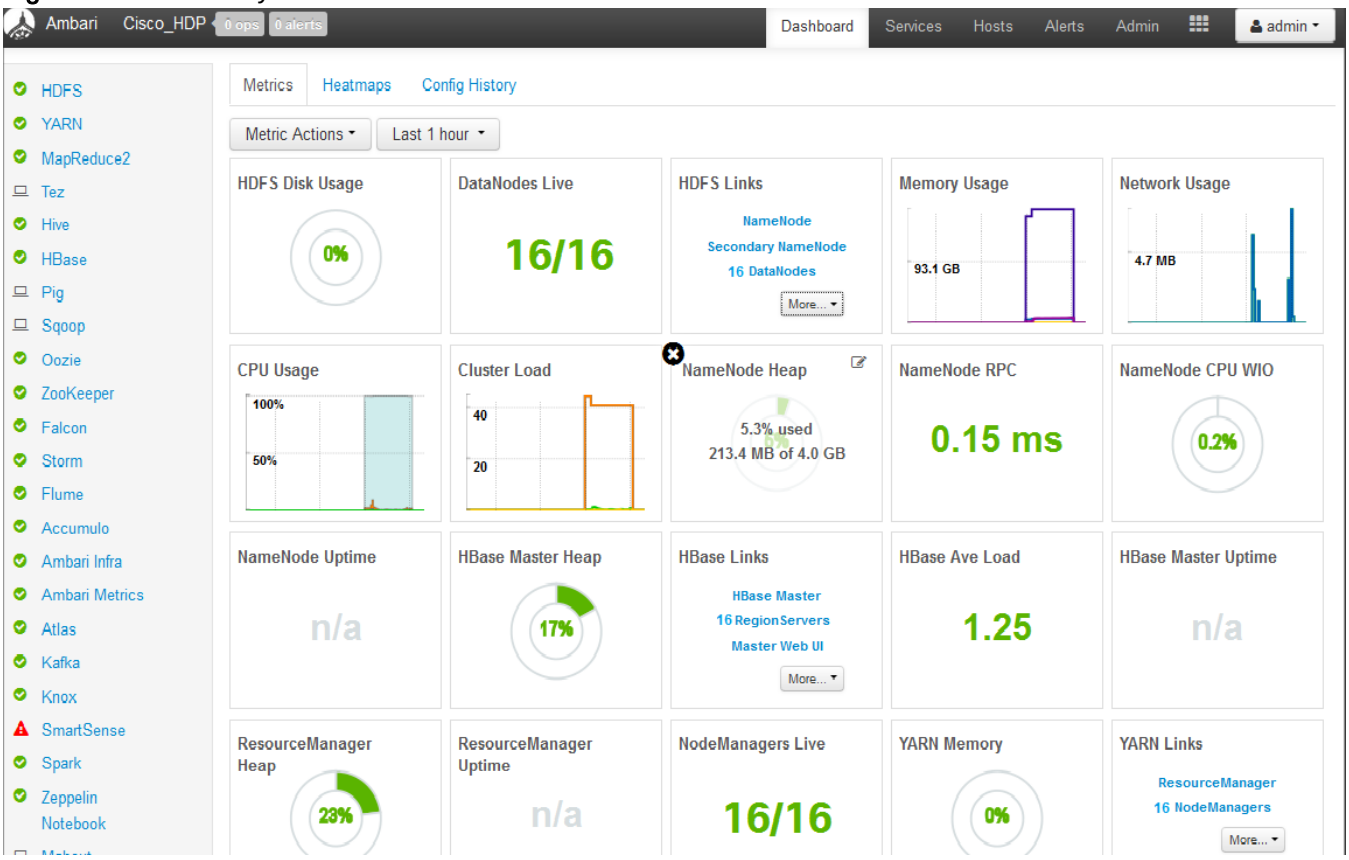


Summary of the Installation Process

Figure 162 shows a summary of the accomplished tasks.

Click Complete.

Figure 162 Summary Screen



Bill of Materials

This section provides the BOM for the 16 nodes. See Error! Reference source not found. for Bill of Materials for the Cisco UCS Fabric Interconnect, Table 16 Bill of Materials for Cisco UCS C240M4 Rack Server, Table 17 for the Cisco UCS S3260 Storage Server Base Rack, Table 18 Bill of Materials for Cisco UCS S3260 Storage Server Capacity Rack, Table 20 for software components.

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 16 Bill of Materials for Cisco UCS C240M4 Rack Server

Part Number	Description	Quantity

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 17 Bill of Materials for Cisco UCS S3260 Storage Server Base Rack

Part Number	Description	Quantity
UCSS-S3260	Cisco UCS S3260 Base Chassis w/4x, 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 18 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
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 6TB, 8TB and 10TB Disk drives also.

Table 19 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 20 Hortonworks Software

Hortonworks Software edition needed for this CVD		
Hortonworks 2.5	UCS-BD-CEDHC-BZ=	19
Hortonworks 2.5	UCS-BD-CEDHC-GD=	19

SKUS for Hortonworks Subscription

Cisco PID (TOP level)	Cisco Subscription PID	Description
UCS-BD-HDP-JSS=	UCS-BD-HDP-JSS-6M	HDP Data Platform Jumpstart Subscription - Up to 16 Nodes – 1 E sponse - 6 Months - sold to new customers only - max. Qty. to b tomer
UCS-BD-HDP-ENT-ND=	UCS-BD-ENT-ND-1Y	HDP Enterprise Subscription - 4 Nodes - 24x7 Sev 1 Response - - min of 3 SKUs required for new customers
UCS-BD-HDP-ENT-ND=	UCS-BD-ENT-ND-2Y	HDP Enterprise Subscription - 4 Nodes - 24x7 Sev 1 Response - - min of 3 SKUs required for new customers
UCS-BD-HDP-ENT-ND=	UCS-BD-ENT-ND-3Y	HDP Enterprise Subscription - 4 Nodes - 24x7 Sev 1 Response - - min of 3 SKUs required for new customers
UCS-BD-HDP-EPL-ND=	UCS-BD-EPL-ND-1Y	HDP Enterprise Plus Subscription - 4 Nodes - 24x7 Sev 1 Respon - min of 3 SKUs required for new customers
UCS-BD-HDP-EPL-ND=	UCS-BD-EPL-ND-2Y	HDP Enterprise Plus Subscription - 4 Nodes - 24x7 Sev 1 Respon - min of 3 SKUs required for new customers
UCS-BD-HDP-EPL-ND=	UCS-BD-EPL-ND-3Y	HDP Enterprise Plus Subscription - 4 Nodes - 24x7 Sev 1 Respon - min of 3 SKUs required for new customers

About the Authors

Manan Trivedi is a Big Data Solutions Architect in the Data Center Solutions Group, Cisco Systems Inc. Manan is part of the Big Data solution engineering team focusing on big data infrastructure and performance.

Ali Bajwa, Principal Partner Solutions Engineer, Technology Alliances Team, Hortonworks, Inc.
Ali is a Senior Partner Solutions Engineer at Hortonworks and works as part of the Technology Alliances team. His focus is to evangelize and assist partners integrate with Hortonworks Data Platform.

Acknowledgements

· Karthik Kulkarni, Big Data Solutions Architect, Data Center Solutions Group, Cisco Systems Inc.

· Barbara Dixon, Technical Writer, Data Center Solutions Group, Cisco Systems, Inc.