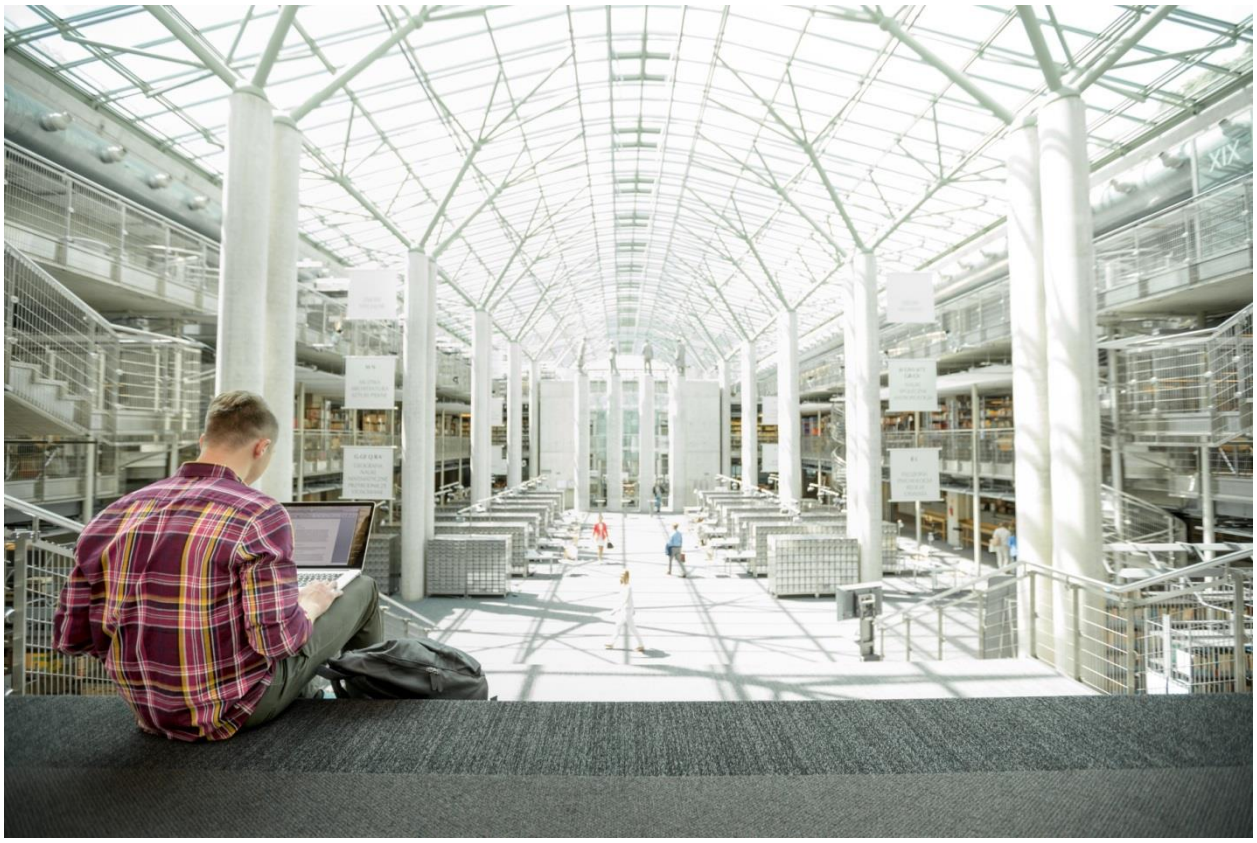


Cisco UCS S3260 Storage Server with IBM BigInsights

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

About Cisco Validated Designs 2
Executive Summary 8
Solution Overview 9
Solution 9
Audience 9
Solution Summary 10
Scaling/Performance Options 10
Data Storage for Big Data Processing 11
Reference Architecture 11
Technology Overview 13
Reference Architecture 13
Cisco UCS S3260 Storage Server with IBM BigInsights 13
Cisco UCS S3260 Storage Servers 13
Cisco UCS 6300 Series Fabric Interconnects 15
Cisco UCS C-Series Rack Mount Servers 16
Cisco UCS Virtual Interface Cards (VICs) 17
Cisco UCS Manager 18
IBM BigInsights for Apache Hadoop: A complete Hadoop Platform 19
IBM IOP 20
Integration with Apache Spark 20
IBM BigSQL 21
IBM Text Analytics 21
BigSheets 22
Big R 22
Enterprise Management 23
Built-in Security 23
Solution Design 24
Requirements 24
Rack and PDU Configuration 24
Port Configuration on Fabric Interconnects 25
Server Configuration and Cabling for UCS S3260 Storage Server 25

Server Configuration and Cabling for Cisco UCS C240 M4 Rack Server 26
Cisco UCS S3260 Storage Server Scaling with Cisco Application Centric Infrastructure (ACI) 27
Software Distributions and Versions 32
IBM BigInsights with Apache Hadoop 32
Red Hat Enterprise Linux (RHEL) 32
Software Versions 32
Fabric Configuration 33
Initial Setup of Cisco UCS 6332 Fabric Interconnects 34
Configure Fabric Interconnect A 34
Configure Fabric Interconnect B 34
Logging Into Cisco UCS Manager 35
Adding a Block of IP Addresses for KVM Access 36
Enabling Uplink Ports 37
Configuring VLANs 38
Enabling Server Ports 40
Creating Chassis Profile 41
Creating Disk Zoning Policy 41
Creating Chassis Firmware Package Policy 45
Creating Chassis Profiles from Template 47
Associating Chassis Profiles to Individual Chassis 52
Creating a Storage Profile for Boot Drives 54
Creating Pools for Service Profile Templates 57
Creating MAC Address Pools 57
Creating a Server Pool 59
Creating Policies for Service Profile Templates 61
Creating Host Firmware Package Policy 61
Creating QoS Policies 62
Platinum Policy 62
Setting Jumbo Frames 64
Creating the Local Disk Configuration Policy 65
Creating Server BIOS Policy 66
Creating the Boot Policy 69
Creating Power Control Policy 71

Creating a Service Profile Template 73
Configuring the Storage Provisioning for the Template 74
Configuring Network Settings for the Template 76
Configuring the vMedia Policy for the Template 82
Configuring Server Boot Order for the Template 83
Configuring Server Assignment for the Template 85
Configuring Operational Policies for the Template 86
Creating Service Profile Templates for Hadoop Management Nodes 89
Creating an Organization 89
Cloning the Template for Hadoop Management Nodes 89
Creating Service Profile from Template 93
Installing Red Hat Enterprise Linux 7.2 on Management Nodes 95
Installing Red Hat Enterprise Linux 7.2 on Data Nodes 117
Post OS Install Configuration 139
Setting Up Password-less Login 139
Configuring /etc/hosts 140
Creating a Red Hat Enterprise Linux (RHEL) 7.2 Local Repo 141
Creating the Red Hat Repository Database 142
Setting up ClusterShell 143
Installing httpd 145
Set Up all Nodes to Use the RHEL Repository 145
Configure DNS 146
Upgrading the Cisco Network Driver for VIC1387 147
Installing xfsprogs 148
NTP Configuration 148
Enabling Syslog 149
Setting ulimit 150
Disabling SELinux 151
Set TCP Retries 151
Disabling the Linux Firewall 152
Disable Swapping 152
Disable Transparent Huge Pages 152
Disable IPv6 Defaults 153

Configuring Data Drives on Name Node and Other Management Nodes 153
Configuring Data Drives on Data Nodes 154
Configuring the Filesystem for NameNodes and DataNodes 155
Cluster Verification 157
Installing IBM BigInsights 162
Pre-Requisites for IOP Installation 162
Set Up the Ambari server 165
Ambari Server Components Set up 166
Creating a Cluster 167
Select a Stack 168
IOP Installation 168
Confirm Hosts 169
Choose Services 170
Assign Masters 171
Assign Slaves and Clients 173
Customize Services 174
HDFS 175
YARN 175
HDFS JVM Settings 175
MapReduce2 175
Hive 176
HBase 177
Oozie 177
Knox 178
Review 179
Summary of Install Process 180
Enabling High Availability 181
HDFS High Availability 181
Setting Up Resource Manager (Yarn) High Availability 189
BigSQL Installation 192
How to Acquire BigInsights Value-added Services 192
Prechecks and Preconditions 193
Bill of Materials 206

About the Authors 211
Acknowledgements 211

Executive Summary

Digital transformation in today's world is accelerating data generation by massive numbers. And with all this data generated organizations are transforming how the business activities, processes, and models leverage this information to get actionable insights from their data. Data storage with retrieval on demand for processing is required for customers. Customers need to achieve the lowest cost for storage, easy access to data while keeping the costs of operation low.

The Cisco® UCS S3260 Storage Server is specifically designed to address this challenge. This next-generation high-density storage system 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 modular design of the Cisco UCS S3260 Storage Server protects the company's 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.

IBM BigInsights for Apache Hadoop is a platform for the analysis and visualization of Internet-scale data volumes. BigInsights is powered by Apache Hadoop, an open source distributed computing platform. BigInsights helps organizations quickly build and deploy custom analytics and workloads to capture insights from big data that can then be integrated into existing databases, data warehouses, and business intelligence infrastructures.

The Cisco UCS S3260 Storage Server for Big Data and Analytics with IBM BigInsights is a tested and dependable deployment model for enterprise quality Hadoop based systems. This system is designed with easy storage, rapid access, operational excellence and security empowerment in mind.

Solution Overview

Enormous amount of information is 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 IBM Big Insights 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.
- Provides advanced analytics built on Hadoop technology to meet big data analysis requirements.
- Integrates with IBM and other information solutions to help enhance data manipulation and management tasks.

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.

Audience

This document describes the architecture and deployment procedures of IBM BigInsights for Apache Hadoop (16 data nodes with 3 master/management nodes) based on Cisco UCS Integrated Infrastructure for Big Data. The intended audience of this document includes, but is not limited to sales engineers, field consultants, professional services, IT managers, partner engineering and customers who want to deploy IBM BigInsights for Apache Hadoop on the Cisco UCS Integrated Infrastructure for Big Data.

Solution Summary

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 with Cisco UCS C240 M4 servers for management. This architecture is specifically designed for performance and linear scalability for big data workloads.

The configuration using Cisco UCS S3260 Storage Servers as data nodes and Cisco UCS C240 M4 rack servers as management nodes is shown in Table 1. This configuration supports the massive scalability that big data enterprise deployments demand. This architecture can scale to thousands of servers with Cisco Nexus 9000 Series Switches.

Table 1 lists the Configuration Details of the Cisco UCS S3260 Storage Server Configuration.

Table 1 **Cisco UCS S3260 Storage Server Configuration Details**

Cisco UCS S3260 Storage Server Data Nodes	Cisco UCS C240 M4 Management Nodes
Connectivity: 2 Cisco UCS 6332 Fabric Interconnects	
8 Cisco UCS S3260 Storage Servers, each with two nodes, each node with:	3 Cisco UCS C240 M4 Rack Servers each with:
• 2 Intel Xeon® processor E5-2680 v4 CPUs (14 cores on each CPU)	• 2 Intel Xeon® processor E5-2680 v4 CPUs (14 cores on each CPU)
• 256 GB of memory	• 256 GB of memory
• Cisco 12-Gbps SAS Modular RAID Controller with 4-GB FBWC	• Cisco 12-Gbps SAS Modular RAID Controller with 2-GB FBWC
• 24 x 8-TB 7, 200-rpm LFF SAS drives (3.07 petabytes [PB] total)	• 8 x 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)
• 2 x 480-GB 6-Gbps 2.5-inch enterprise value SATA SSD drives for boot	• 2 x 240-GB 6-Gbps 2.5-inch enterprise value SATA SSD drives for boot

Scaling/Performance Options

Depending on the storage and performance requirements, any of the following HDDs can be used with an optional expansion to 24 drives per node. Table 2 lists the Hard Drives and their Expansion Storage Values.

Table 2 **Hard Drives with Storage and Expansion Storage Values**

HDD	Total Storage (24 drives)	Expansion (with 28 drives)
4 TB 7200 –rpm LFF SAS drives	1.54 PetaBytes	1.8 PetaBytes

HDD	Total Storage (24 drives)	Expansion (with 28 drives)
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 ~rpm LFF SAS drives	3.84 PetaBytes	4.48 PetaBytes

Data Storage for Big Data Processing

Companies have realized the power of big data and are now collecting more data than ever before. They need to get value from this data, often in real-time. Sensors, IoT devices, social network data and online transactions are just a few examples. They are all generating data continuously, 24x7. This data needs to be captured, stored, monitored and processed quickly in order to make informed, data-driven decisions in real-time.

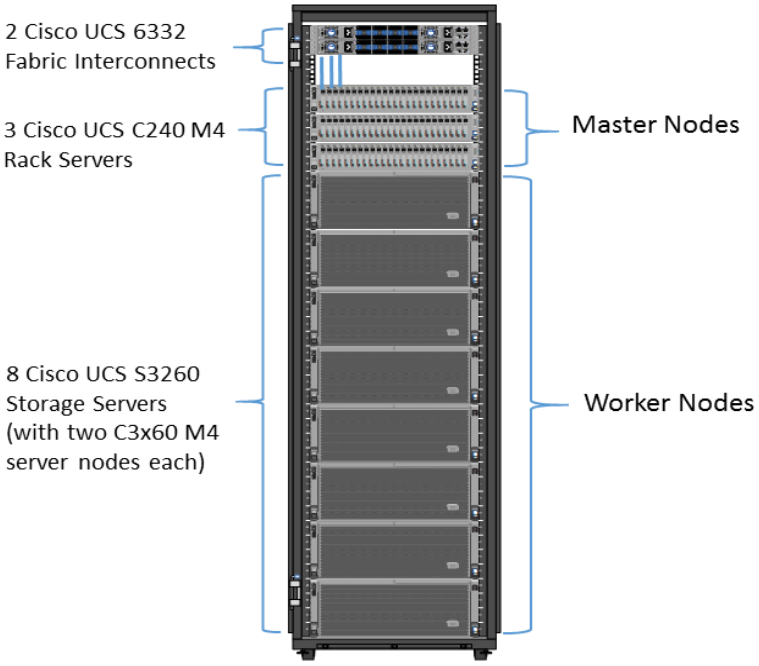
In addition, real-time streaming data needs to be sent to the company's enterprise-wide data store where it can be used for traditional analysis and reporting, data discovery and as input to sophisticated machine-learning algorithms.

The next section details the relevant reference architectures for meeting these real world challenges.

Reference Architecture

Figure 1 shows the base configuration of a high-availability cluster. It is comprised of 16 data nodes using Cisco UCS S3260 Storage Servers and 3 management nodes using Cisco UCS C240 M4 rack servers.

Figure 1 Reference Architecture



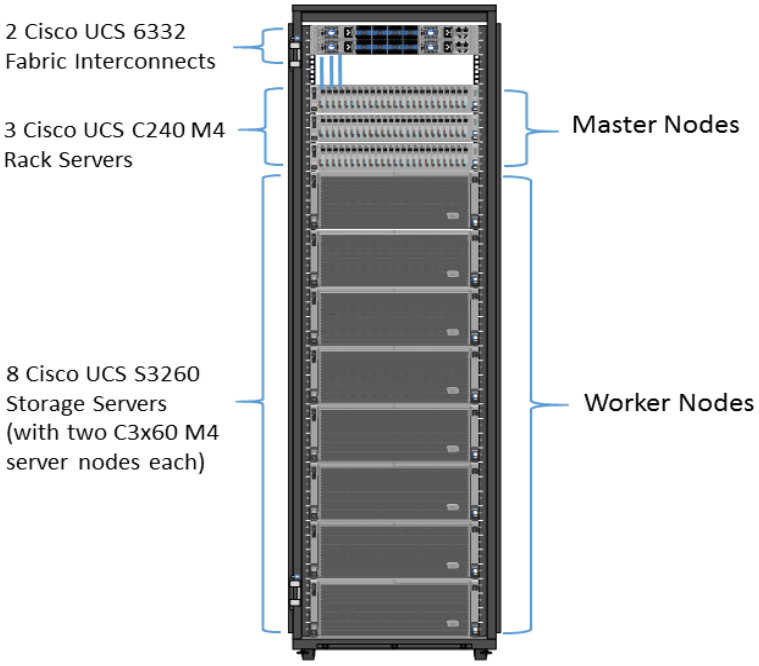
Note: This CVD describes the installation process of IBM BigInsights 4.2 for a cluster (3 management nodes for high-availability + 16 data nodes).

Table 3 **Components for a BigInsights 19 Node Cluster**

Component	Description
Connectivity	2 x Cisco UCS 6332 32-Port Fabric Interconnects
Big Insights Nodes	3 x Cisco UCS C240 M4 Rack Server 8 x Cisco UCS S3260 Storage 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.

Technology Overview

Reference Architecture



Cisco UCS S3260 Storage Server with IBM BigInsights

This solution is based on the Cisco Unified Computing System (Cisco UCS) infrastructure using Cisco UCS 6300 Series Fabric Interconnects, Cisco UCS S3260 Storage Servers, and Cisco UCS C240 M4 Rack Servers. This architecture is specifically designed for performance and linear scalability for big data workloads and is built using the following components:

Cisco UCS S3260 Storage Servers

The Cisco UCS S3260 Storage Server 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.

Figure 2 Cisco UCS S3260 Storage Server

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 a 4-GB cache and a host bus adapter (HBA) controller. For each server, there are two internal solid-state-disk (SSD) drives for booting.

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

The Cisco UCS S3260 Storage Server chassis has 56 top-load LFF HDDs with a maximum capacity of 10 TB per HDD and can be mixed with up to 24 SSDs with maximum capacity of 3.2 TB per SSD.

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 6300 Series Fabric Interconnects

Cisco UCS 6300 Series Fabric Interconnects 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 4 Cisco UCS 6332 UP 32-Port Fabric Interconnect

Cisco UCS C-Series Rack Mount Servers

Cisco UCS C-Series C220 M4 High-Density Rack Servers (Small Form Factor Disk Drive Model) (Figure 5), and Cisco UCS C240 M4 High-Density Rack Servers (Small Form Factor Disk Drive Model) (Figure 6), 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 v3 and 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 servers provides 24 DIMM (PCIe) 3.0 slots and can support up to 1.5 TB of main memory (128 or 256 GB is typical for big data applications). It can support a range of disk drive and SSD options; twenty-four Small Form Factor (SFF) disk drives, plus two (optional) internal SATA boot drives, for a total of 26 internal drives are supported in the Performance-optimized option, or twelve Large Form Factor (LFF) disk drives option 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 1227 (VICs), are 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 5 Cisco UCS C220 M4 Rack Server (Small Form Factor Disk Drive Model)

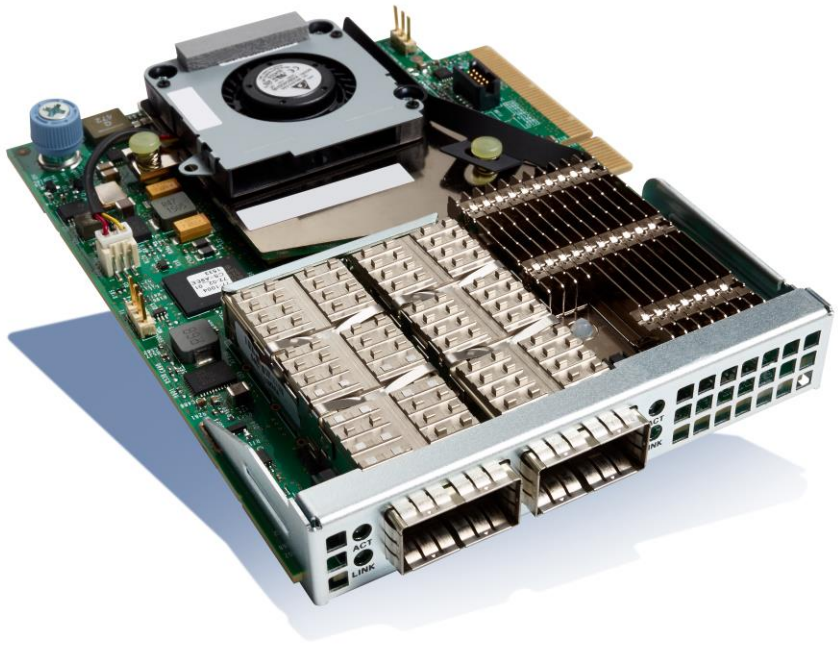
Figure 6 Cisco UCS C240 M4 Rack Server

Cisco UCS Virtual Interface Cards (VICs)

Cisco UCS Virtual Interface Cards (VICs) are unique to Cisco. Cisco UCS Virtual Interface Cards incorporate next-generation converged network adapter (CNA) technology from Cisco, and offer dual 10-Gbps ports designed for use with Cisco UCS C-Series Rack-Mount Servers. Optimized for virtualized networking, these cards deliver high performance and bandwidth utilization, and support up to 256 virtual devices. The Cisco UCS Virtual Interface Card (VIC) 1227 is a dual-port, enhanced Small Form-Factor Pluggable (SFP+), 10 Gigabit Ethernet, and Fiber Channel over Ethernet (FCoE)-capable, PCI Express (PCIe) modular LAN on motherboard (mLOM) adapter. It is designed exclusively for the M4 generation of Cisco UCS C-Series Rack Servers and the C3160 dense storage servers.

The Cisco UCS Virtual Interface Card 1387 (Figure 7) 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 7 Cisco UCS VIC 1387

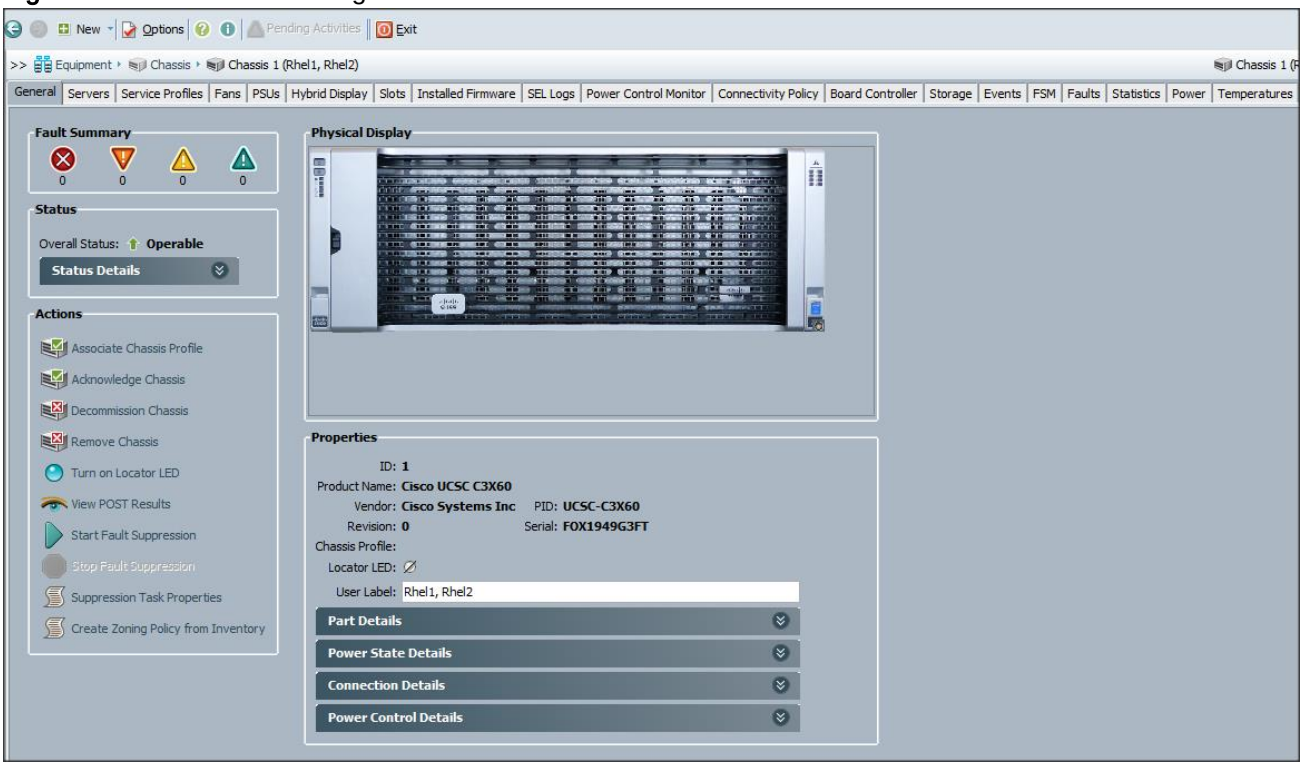


Cisco UCS Manager

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), 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 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 case Cisco UCS Manager will automatically detect the failed drives and replace with one of the available hot spare drives, rebuild it and make it available to use within the Chassis.

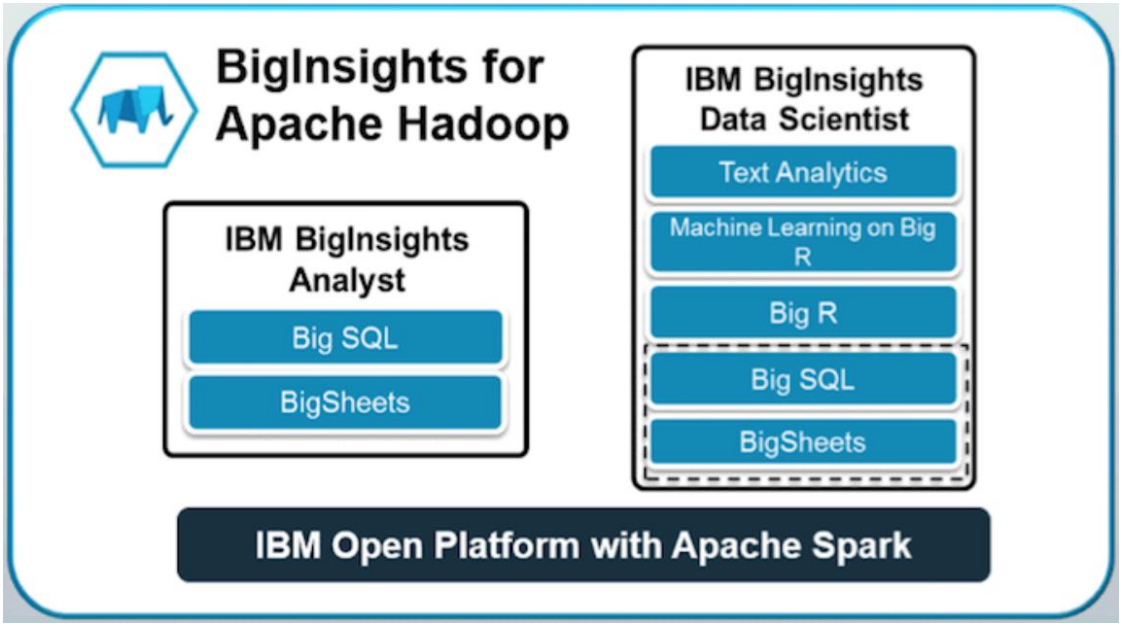
Figure 8 Cisco UCS Manager



IBM BigInsights for Apache Hadoop: A complete Hadoop Platform

With IBM BigInsights 4.2 (Figure 9), IBM provides the full range of analytics for Hadoop, Spark and SQL over an open and flexible platform. This platform combines batch processing, SQL, streaming and complex analytics seamlessly for any application to handle a wide range of data processing scenarios.

Figure 9 IBM BigInsights



IBM BigInsights, built on IBM IOP, is designed with analytics, operational excellence and security empowerment in mind.

IBM BigInsights with Apache Hadoop at the core:

- Provides advanced analytics built on Hadoop technology (IBM BigInsights Data Scientist module) to meet big data analysis requirements.
- Designed for performance and usability (IBM BigInsights Analyst module) through performance optimized capabilities, visualization, rich developer tools and powerful analytic functions.
- Delivers management, security and reliability features (IBM Enterprise Management module) to support large-scale deployments and help speed time to value.
- Integrates with IBM and other information solutions to help enhance data manipulation and management tasks.

IBM BigInsights with Apache Hadoop provides a scalable, flexible, integrated platform that makes it easy to manage rapidly increasing volumes and varieties of data in any enterprise. Industry-leading IBM BigInsights with Apache Hadoop products and solutions enable to deploy and manage Apache Hadoop and related projects, manipulate and analyze data, and keep that data secure and protected.

IBM IOP

IBM Open Platform (IOP) with Apache Hadoop and Apache Spark is IBM's big data platform. IOP is built on 100% open source Apache Ecosystem components. It is designed for flexible and efficient analytics and operations.

Key features of IBM Open Platform with Apache Hadoop include:

- Ambari operational framework for provisioning, managing & monitoring Apache Hadoop clusters.
- Native support for rolling upgrades for Hadoop services.
- Support for long-running applications within YARN:
  - Higher cluster utilization.
  - Lower operational costs.
  - Reduced data motion.

Integration with Apache Spark

IBM IOP includes integration with Apache Spark 1.6.1. The benefits include fast processing from the Spark core, near real-time analytics with Spark streaming, built-in machine learning libraries that are highly extensible using Spark MLlib, querying of unstructured data and more value from free-form text analytics with Spark SQL, and graph computation/graph analytics with Spark GraphX.

IBM BigSQL

IBM BigSQL is the ultimate platform for RDBMS off-load and consolidation, featuring standard compliant SQL, as well as support many vendor specific extensions.

It is faster and easier to offload old data from existing enterprise data warehouses or data marts to free up capacity while preserving most of the familiar SQL from those platforms. BigSQL's SQL engine for Hadoop can work with Hive, HBase, and Spark concurrently for best in class analytic capabilities.

As with any RDBMS, performance is a critical factor, and this is certainly the case for BigSQL. One of the most significant improvements in Big SQL 4.2 is improved out-of-the-box performance, including more partitioning capabilities, better default execution plans.

Key Features of IBM BigSQL include:

- Easy installation/administration via Apache Ambari.
- Improved performance with concurrent query processing/partition options/optimized default configurations.
- Statistics collection and measurement.
- Enhanced security via Impersonation support to allow a service user to securely access data in Hadoop on behalf of another user.
- Metadata Integration: automatic synchronization of Big SQL metadata with Hive.
- Resource Management: More optimal distribution of resources for high demand (enterprise) environments.
- BigSQL disaster recovery:
- Support for Online backup of BigSQL metastore + data (local tables) and offline restore on remote DR site.
- Regular backup/restored configured to meet user's *recovery* window requirements.

IBM Text Analytics

IBM Text Analytics is a powerful system for extracting structured information from unstructured and semi-structured text by defining rules to create extractors. It includes an all-new powerful web-based Visual Text Analytics Framework allowing developers to easily build high-quality applications that can process text in multiple written languages and derive insights from large amounts of native textual data in various formats.

Cisco UCS Integrated Infrastructure for Big Data and Analytics offers several configurations to meet a variety of computing and storage requirements.

BigSheets

BigSheets turns Do It Yourself Analytics into a reality for analysts by going beyond structured database management into unstructured data management. Seeing the whole picture will help all levels of business make better decisions.

BigSheets provides a web-based, spreadsheet-style view into collections of files in Hadoop. Users can perform data transformations, filtering and visualizations at massive scale. No coding is required because BigSheets translates the spreadsheet actions into MapReduce to leverage the computational resources of the Hadoop cluster. This helps analysts discover value in data quickly and easily.

BigSheets is an extension of the mashup paradigm that:

- Integrates gigabytes, terabytes, or petabytes of unstructured data from web-based repositories.
- Collects a wide range of unstructured web data stemming from user-defined seed URLs.
- Extracts and enriches data using the unstructured information management architecture selected (LanguageWare, OpenCalais, etc.).
- Lets users explore and visualize this data in specific, user defined contexts (such as ManyEyes).

Some of BigSheets benefits include:

- Provides business users with a new approach to keep pace with data escalation. By taking the structure to the data, this helps mine petabytes of data without additional storage requirements.
- BigSheets provides business users with a new approach that allows them to break down data into consumable, situation-specific frames of reference. This enables organizations to translate untapped, unstructured, and often unknown web data into actionable intelligence.
- Leverage all the compute resources of the Hadoop cluster to drive insights and visualizations with BigSheets right on the cluster–no extraction required.

Big R

The IBM BigInsights Data Scientist module includes Big R. Big R enables data scientists to run native R functions to explore, visualize, transform, and model big data right from within the R environment. Data scientists can run scalable machine learning algorithms with a wide class of algorithms and growing R-like syntax for new algorithms & customize existing algorithms. BigInsights for Apache Hadoop running Big R can use the entire cluster memory, spill to disk and run thousands of models in parallel.

Big R provides a new processing engine, and enables automatic tuning of machine learning performance over massive data sets in Hadoop clusters. Big R can be used for comprehensive data analysis, hiding some of the complexity of manually writing MapReduce jobs.

Benefits of Big R include:

- End-to-end integration with open source R.
- Transparent execution on Hadoop.
- Seamless access to rich and scalable machine learning algorithms provided in Big R.
- Text analytics to extract meaningful information from unstructured data.

Enterprise Management

The IBM BigInsights Enterprise Management module provides a comprehensive web-based interface included in BigInsights that simplifies cluster management, service management, job management and file management. Administrators and users can share the same interface, launching applications and viewing a variety of configurable reports and dashboards.

Built-in Security

IBM IOP now supports Apache Ranger. It provides a centralized security platform for managing authorization, access control, auditing and data protection. Another new feature in Big SQL 4.2 is the support of Impersonation. Impersonation is the ability to allow a service user to securely access data in Hadoop on behalf of another user. In Big SQL, impersonation can be enabled at the global level to enable impersonation of connected users for actions on Hadoop tables.

Solution Design

Requirements

This CVD describes architecture and deployment procedures for IBM BigInsights (4.2.0) on eight Cisco UCS S3260 Storage Server chassis each with two C3X60 M4 server nodes as Hadoop data nodes, and three Cisco UCS C240 M4 as Hadoop Management nodes. The solution goes into detail configuring IOP 4.2 on the infrastructure.

The cluster configuration consists of the following:

- Two Cisco UCS 6332 Fabric Interconnects
- Three Cisco UCS C240 M4 servers
- Eight Cisco UCS S3260 Storage Server Chassis (2 Server nodes on each chassis)
- 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 C240M4 Servers. Each chassis is connected to two vertical PDUs for redundancy; thereby, ensuring availability during power source failure.

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

Table 4 **Rack Configuration**

Position	Devices
42	Cisco UCS FI 6332
41	Cisco UCS FI 6332
40	Unused
39	Unused
38	Cisco UCS C240M4 Rack Server
37	
36	Cisco UCS C240M4 Rack Server
35	
34	Cisco UCS C240M4 Rack Server
33	
32	Cisco UCS S3260 Storage Server
31	
30	

29	
8	Cisco UCS S3260 Storage Server
27	
26	
25	
24	Cisco UCS S3260 Storage Server
23	
22	
21	
20	Cisco UCS S3260 Storage Server
19	
18	
17	
16	Cisco UCS S3260 Storage Server
15	
14	
13	
12	Cisco UCS S3260 Storage Server
11	
10	
9	
8	Cisco UCS S3260 Storage Server
7	
6	
5	
4	Cisco UCS S3260 Storage Server
3	
2	
1	

Port Configuration on Fabric Interconnects

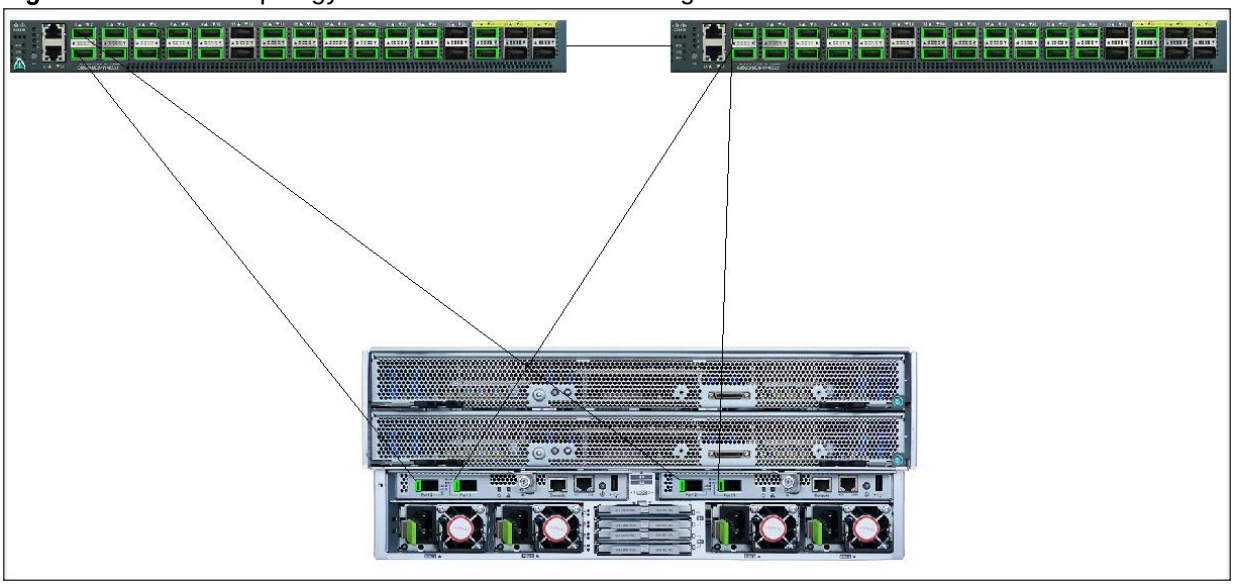
Port Type	Port Number
Network	32
Server	1 to 19

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 10 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 10 Fabric Topology for Cisco UCS S3260 Storage Server with UCSC-C3K-M4SRB 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

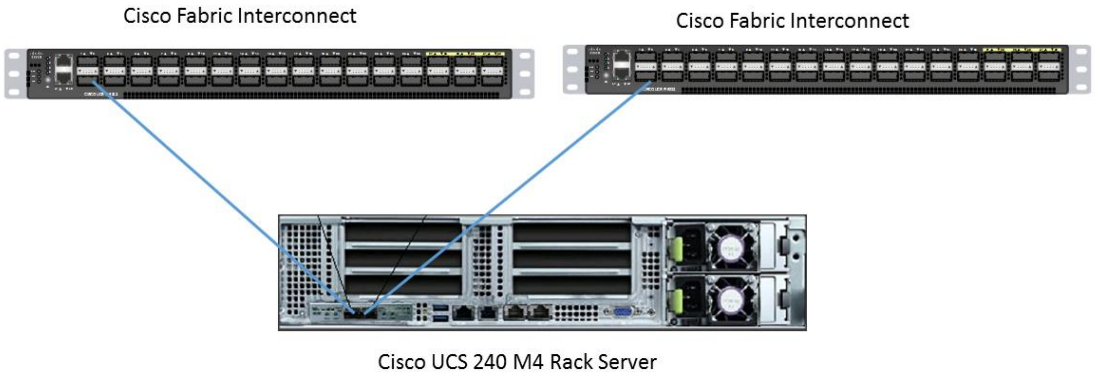
Figure 10 depicts the connectivity between Cisco UCS S3260 Storage Server chassis and Cisco UCS 6300 Fabric Interconnects. Each chassis has two C3x60 M4 server nodes. Each link in the figure represents a 40 Gigabit Ethernet link from the Cisco UCS S3260 Storage Server chassis connecting to a Fabric Interconnect.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

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

Server Configuration and Cabling for Cisco UCS C240 M4 Rack Server

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

Figure 11 Fabric Topology for Cisco UCS C240 M4 Rack Server

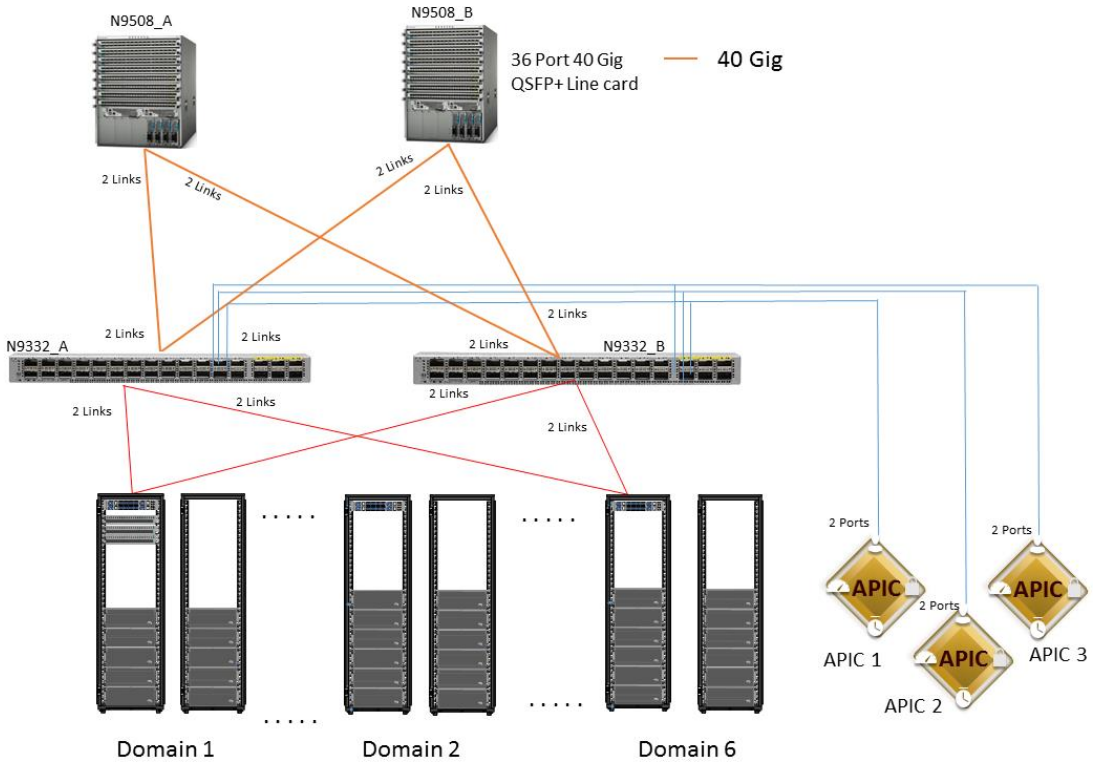


Cisco UCS S3260 Storage Server Scaling with Cisco Application Centric Infrastructure (ACI)

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

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

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



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 5 below.

Table 5 **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 6 **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 software distributions required versions are listed below.

IBM BigInsights with Apache Hadoop

For more information visit https://www.ibm.com/support/knowledgecenter/en/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.welcome.doc/doc/welcome.html.

Red Hat Enterprise Linux (RHEL)

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

Software Versions

The software versions tested and validated in this document are shown in Table 7.

Table 7 **Software Versions**

Layer	Component	Version or Release
Compute (Chassis)	Board Controller	1.0.14
System IO Controller	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

Layer	Component	Version or Release
	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
	LSI MegaRAID SAS Driver	06.810.10.00
Software	Red Hat Enterprise Linux Server	7.2 (x86_64)
	Cisco UCS Manager	3.1(2b)
	Big Insights	4.2

Note: The latest drivers can be downloaded from the link below: https://software.cisco.com/download/release.html?mdfid=283862063&flowid=25886&softwareid=283853158&release=1.5.7d&relind=AVAILABLE&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 Cisco UCS Manager using virtual IP address of using the web browser.
- Launch Cisco UCS Manager.
- Enable server, uplink and appliance ports.
- Start discovery process.
- Create pools and polices for service profile template.
- Create Service Profile template and 64 Service profiles.
- Associate Service Profiles to servers.

Initial Setup of Cisco UCS 6332 Fabric Interconnects

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

Configure Fabric Interconnect A

1. Connect to the console port on the first Cisco UCS 6332 Fabric Interconnect.
2. At the prompt to enter the configuration method, enter console to continue.
3. If asked to either perform a new setup or restore from backup, enter setup to continue.
4. Enter y to continue to set up a new Fabric Interconnect.
5. Enter y to enforce strong passwords.
6. Enter the password for the admin user.
7. Enter the same password again to confirm the password for the admin user.
8. When asked if this fabric interconnect is part of a cluster, answer y to continue.
9. Enter 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.

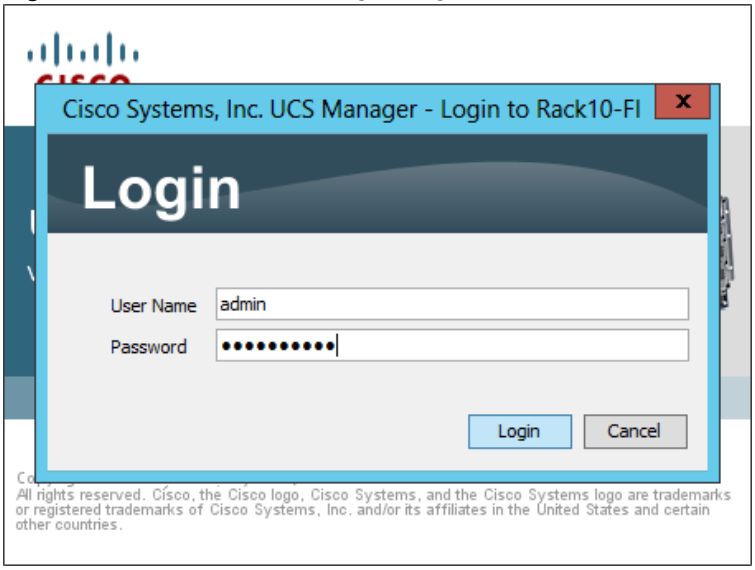
For more information on configuring Cisco UCS 6200 Series Fabric Interconnect, see: http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/gui/config/guide/2.0/b_UCSM_GUI_Configuration_Guide_2_0_chapter_0100.html.

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 12)

Figure 12 Cisco UCS Manager Login Screen

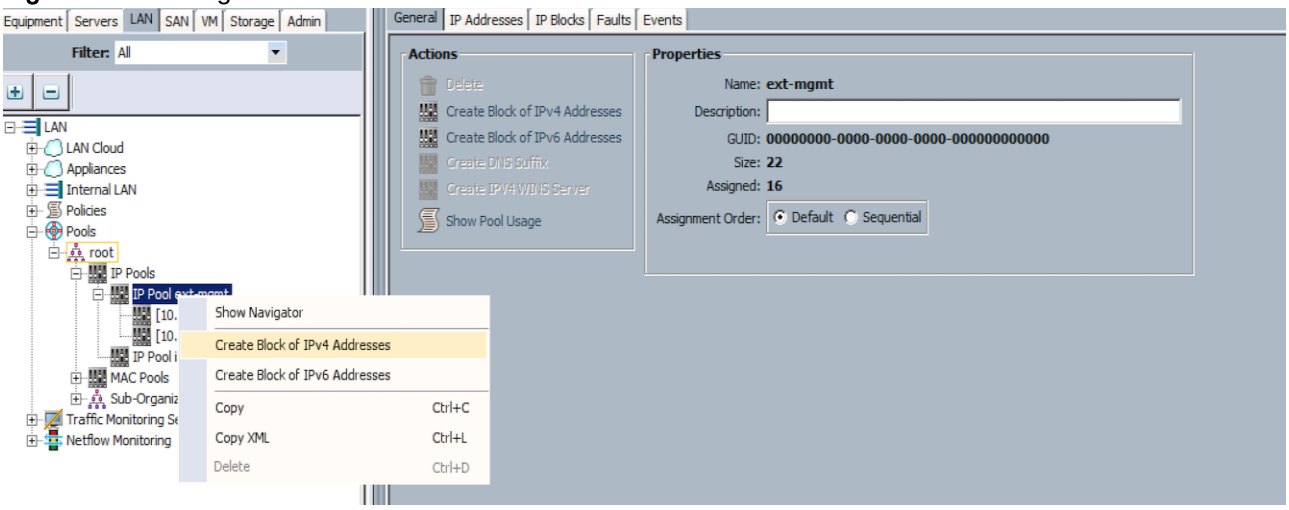


Adding a Block of IP Addresses for KVM Access

To create a block of KVM IP addresses for server access in the Cisco UCS environment, complete the following steps, as shown in Figure 13:

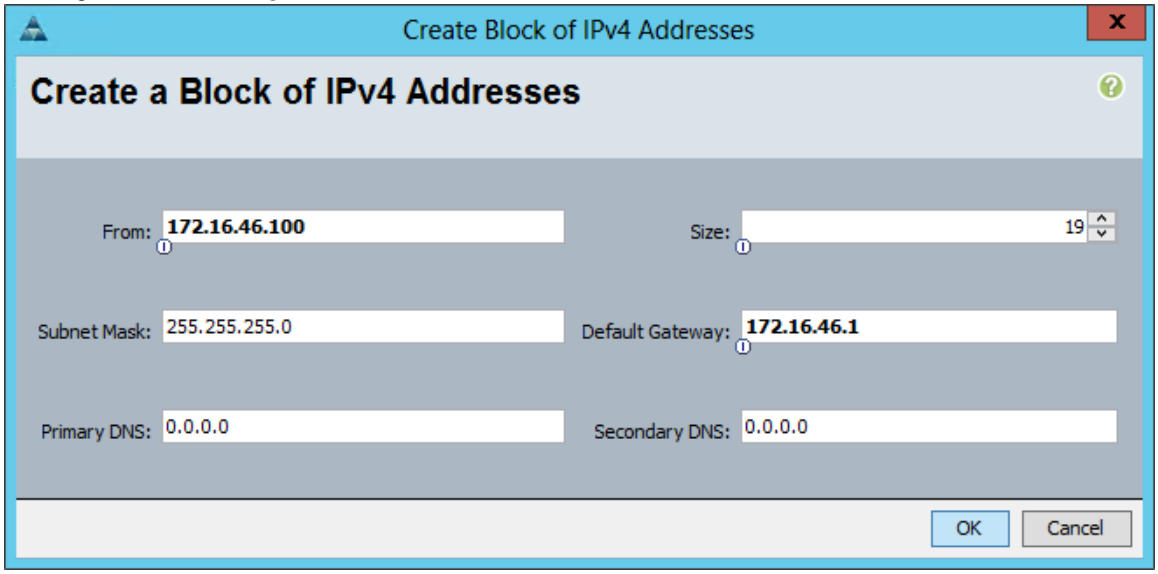
1. Select the `LAN` tab at the top of the left window.
2. Select `Pools > IpPools > Ip Pool ext-mgmt.`
3. Right-click `IP Pool ext-mgmt.`
4. Select `Create Block of IPv4 Addresses.`

Figure 13 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 14).

Figure 14 Adding a Block of IPv4 Addresses for KVM Access Part 2



6. Click OK to create the IP block.

7. Click OK in the message box.

Enabling Uplink Ports

To enable uplinks ports, complete the following steps, as shown in Figure 15:

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

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

3. Expand the Unconfigured Ethernet Ports section.

4. Select port 1 that is connected to the uplink switch, right-click, then select Reconfigure > Configure as Uplink Port.

5. Select Show Interface and select 10GB for Uplink Connection.

6. A pop-up window appears to confirm the selection. Click Yes then OK to continue.

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

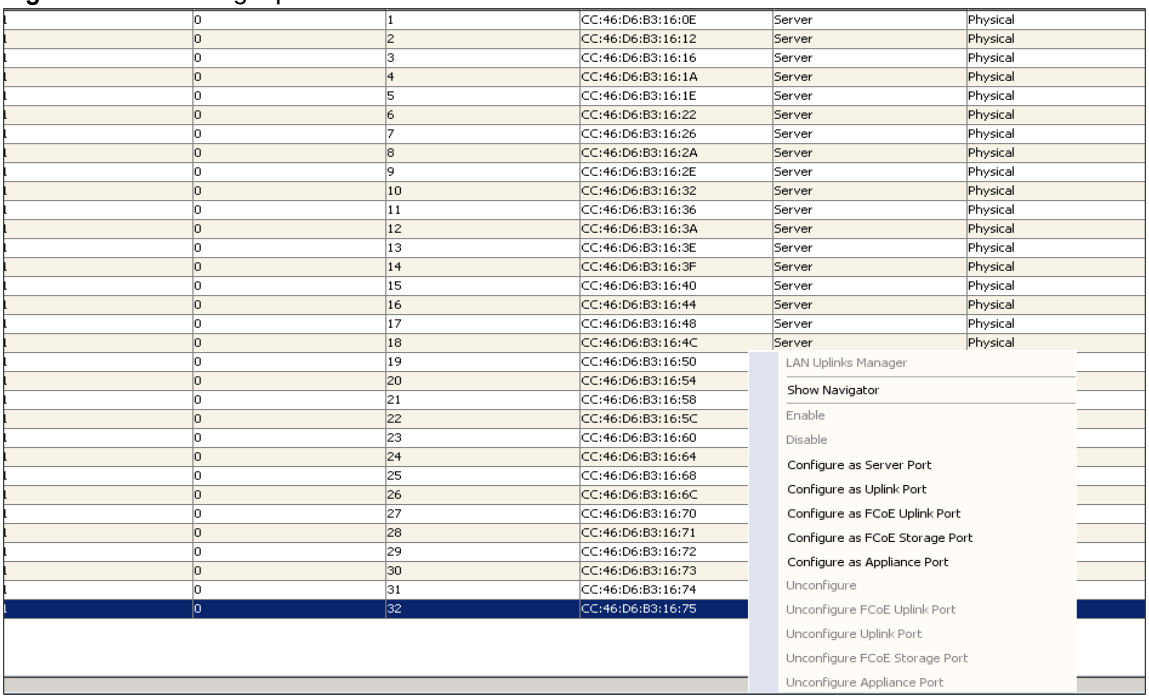
8. Expand the Unconfigured Ethernet Ports section.

9. Select port number 1, which is connected to the uplink switch, right-click, then select Reconfigure > Configure as Uplink Port.

10. Select Show Interface and select 10GB for Uplink Connection.

11. A pop-up window appears to confirm the selection. Click Yes then OK to continue.

Figure 15 Enabling Uplink Ports



Configuring VLANs

VLANs are configured as in shown in Table 8.

Table 8 **VLAN Configurations**

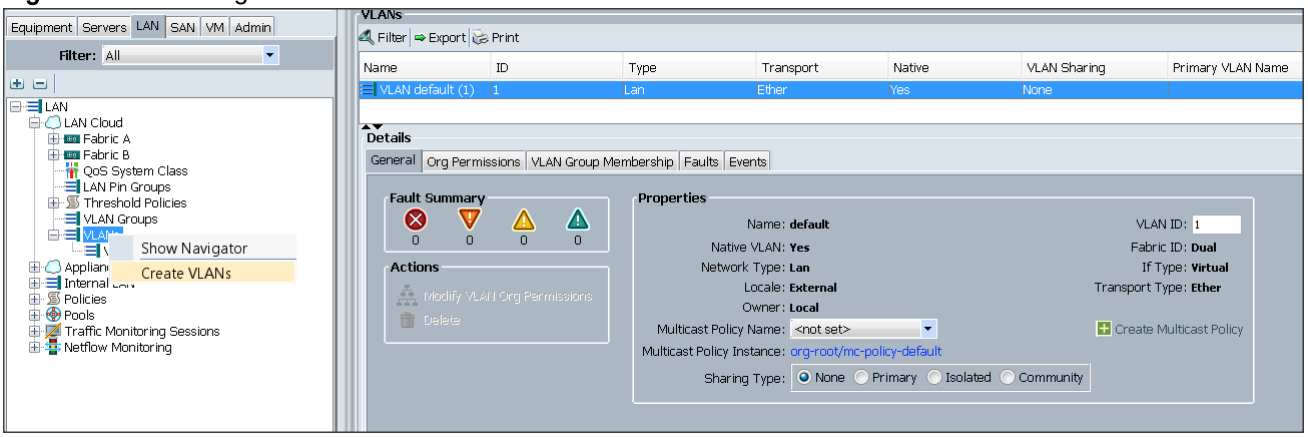
VLAN	NIC Port	Function
VLAN76	eth0	Management & Data Traffic

The NIC will carry 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, as shown in Figure 16:

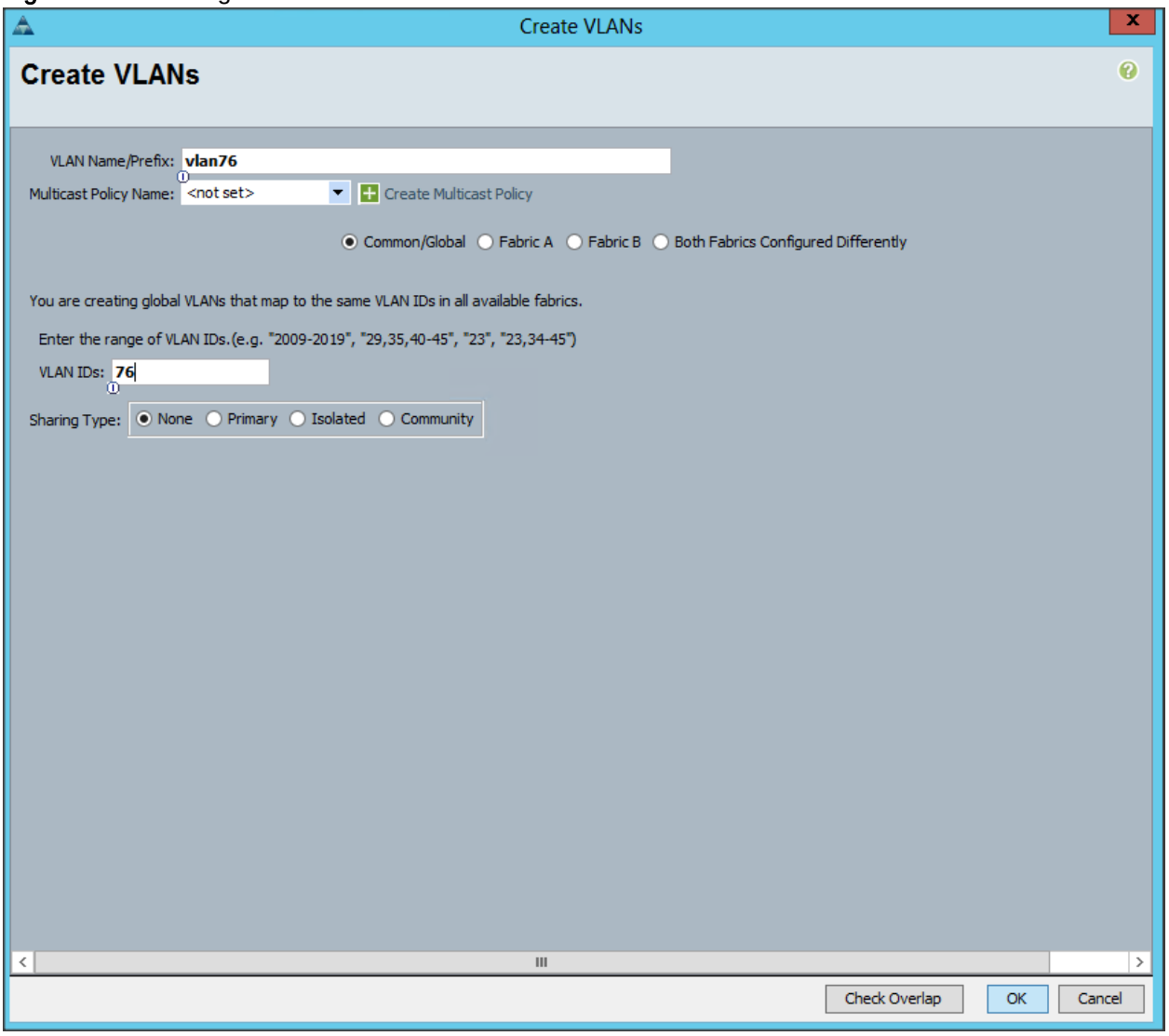
1. Select the `LAN` tab in the left pane in the Cisco UCSM GUI.
2. Select LAN > LAN Cloud > VLANs.
3. Right-click the `VLANs` under the root organization.
4. Select `Create VLANs` to create the VLAN.

Figure 16 Creating a VLAN



5. Enter vlan76 for the `VLAN Name`. (Figure 17)
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 17 Creating a VLAN for DATA



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

Enabling Server Ports

To enable server ports, complete the following steps:

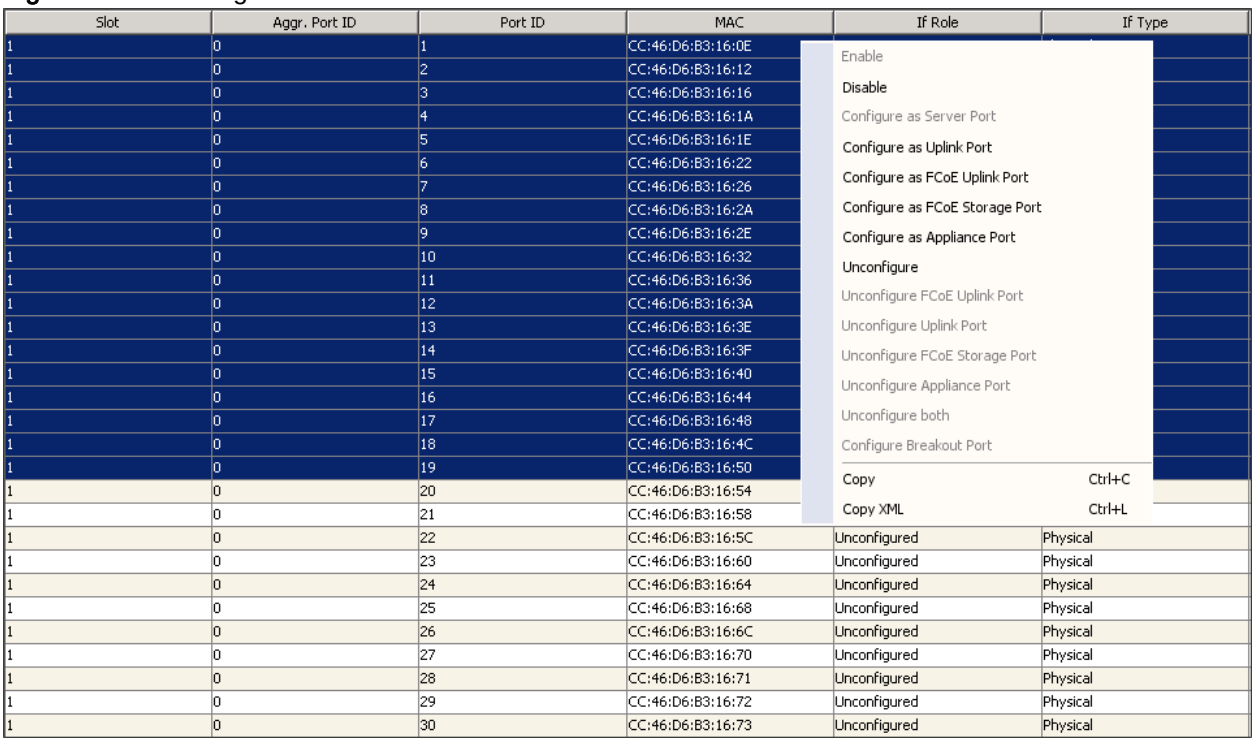
1. Select the `Equipment` tab on the top left of the window (Figure 18).
2. Select `Equipment > Fabric Interconnects > Fabric Interconnect A (primary) > Fixed Module.`
3. Click the Ethernet Ports section.
4. Select all the ports that are connected to the Servers right-click them, and select `Reconfigure > Configure as a Server Port.`

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 18 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.

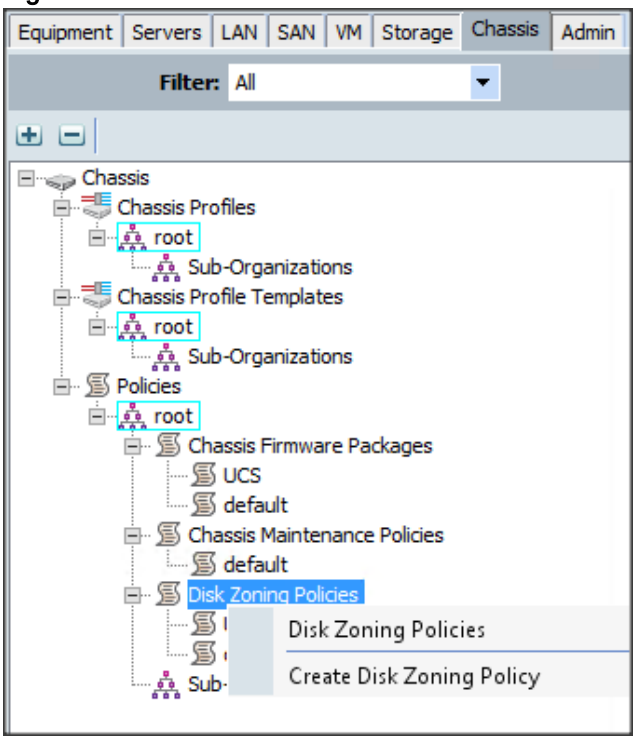
Creating Disk Zoning Policy

1. Click the Chassis tab on UCS Manager on the top left menu (Figure 19).

2. Expand Policies→Root→Disk Zoning Policies

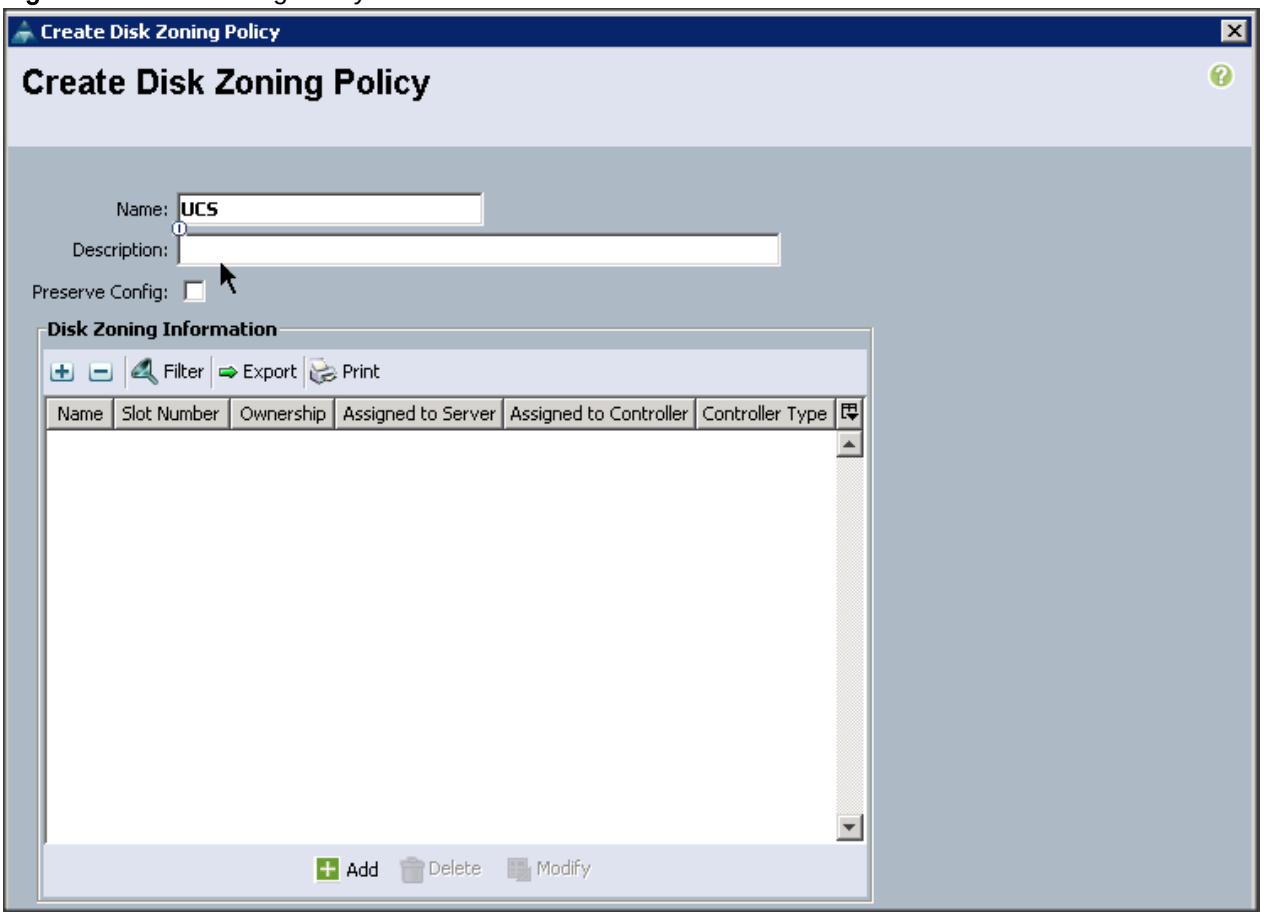
3. Right click on the Disk Zoning Policies and click Create Disk Zoning Policy.

Figure 19 Chassis Profile Screen



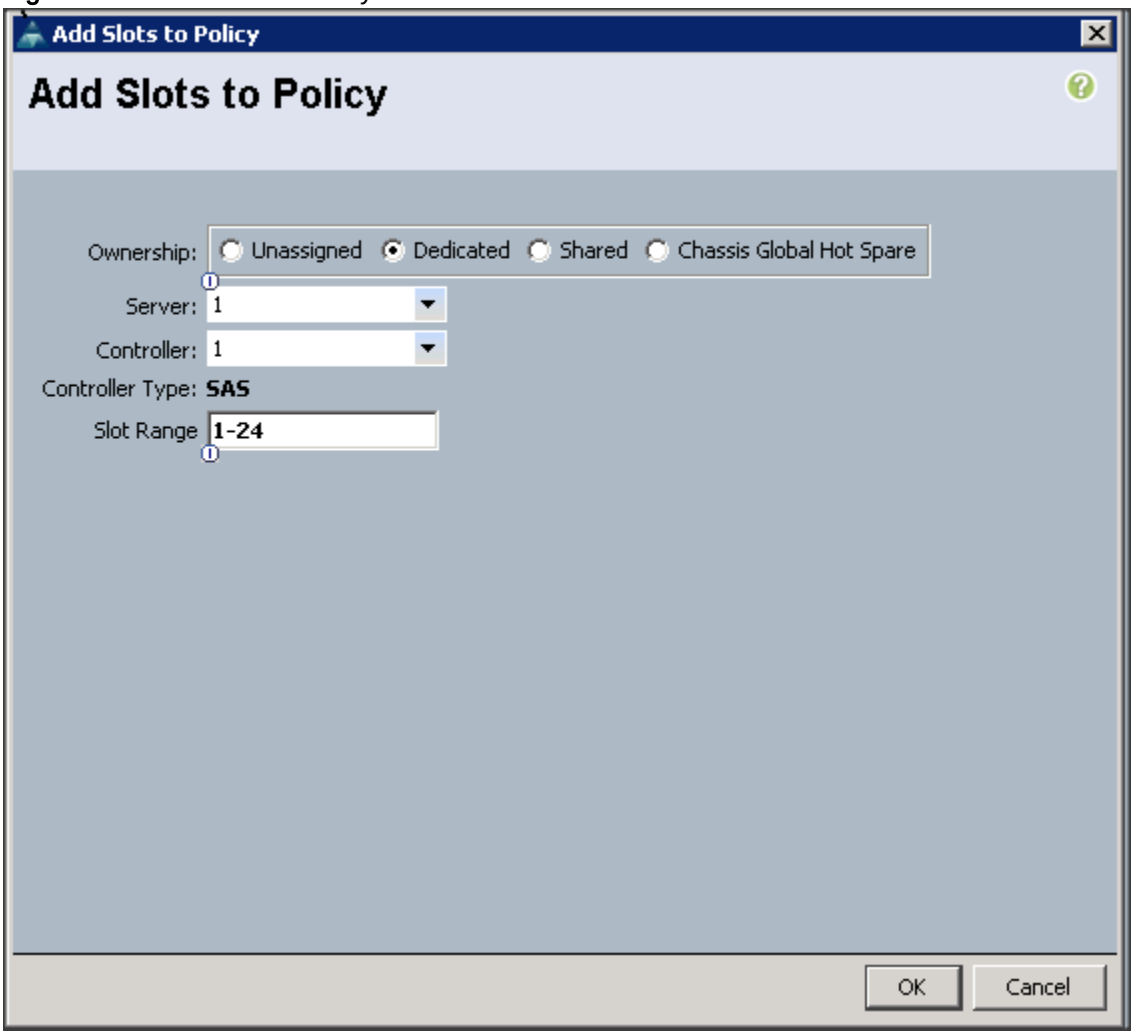
4. On Create Disk Zoning Policy windows enter the Name UCS and click “+” to create the Disk Zoning. (Figure 20)

Figure 20 Disk Zoning Policy Screen



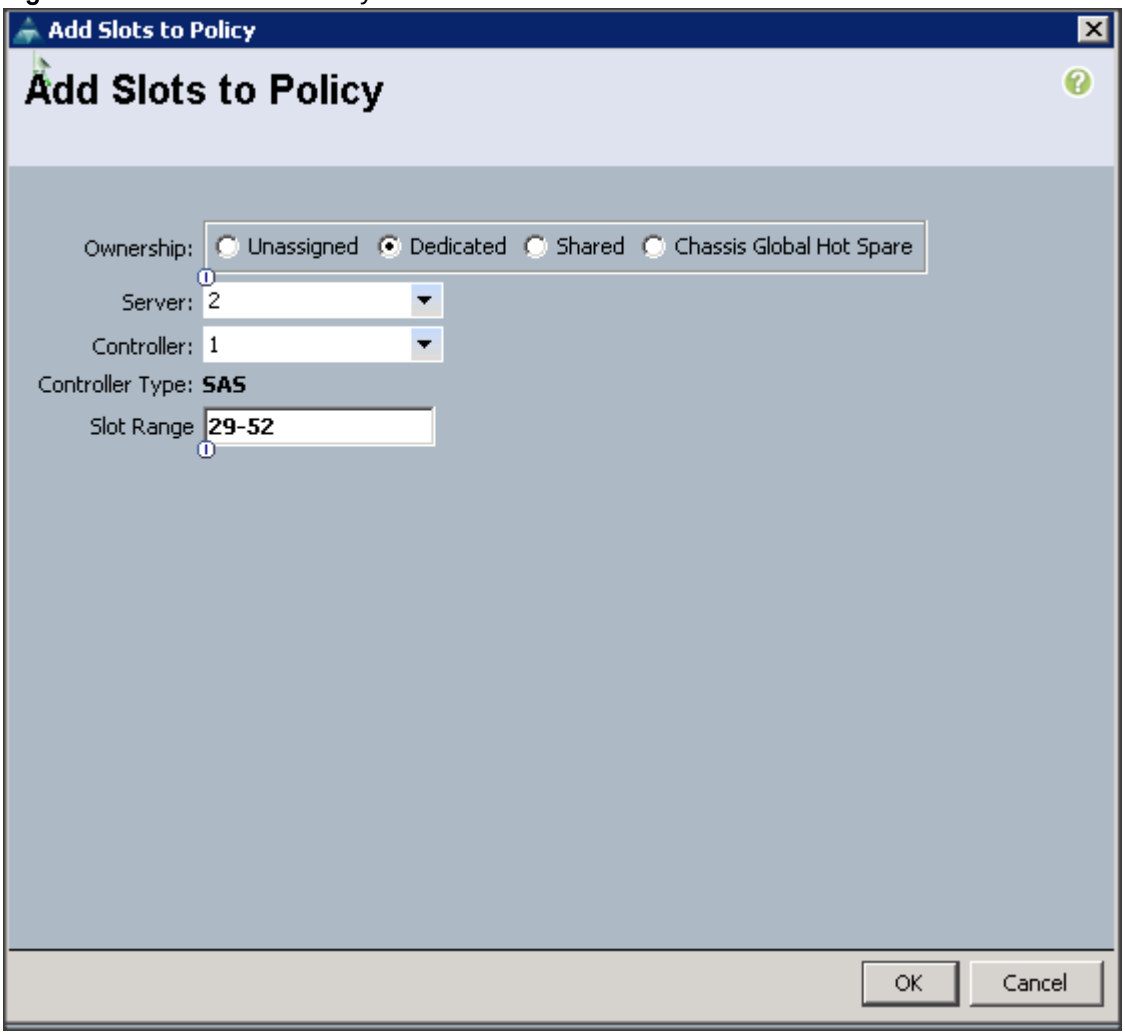
5. In the Add Slots to Policy window (Figure 21), select the “Dedicated” radio button. From the server drop down list choose “1”, from the controller drop down list choose “1”, in the slot range enter 1-24 and click “OK”.

Figure 21 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 22)

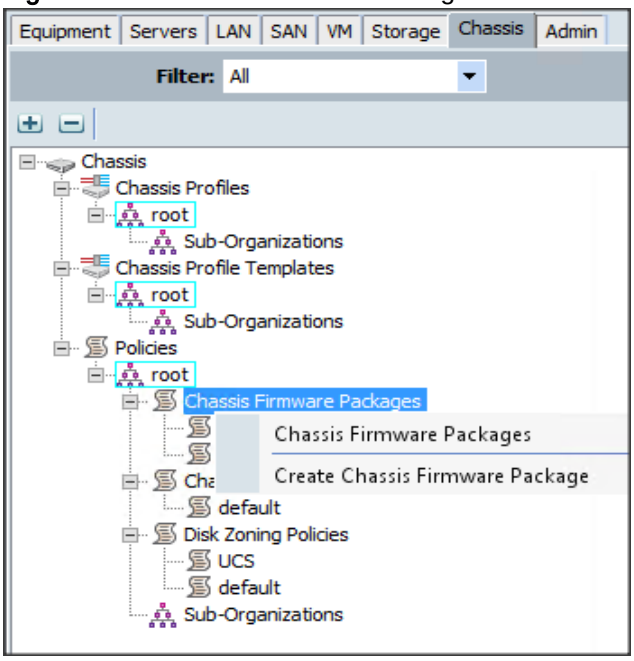
Figure 22 Add Slots to Policy Screen 2



Creating Chassis Firmware Package Policy

1. Right click on the Chassis Firmware Packages and click “Create Chassis Firmware Packages”. (Figure 23)

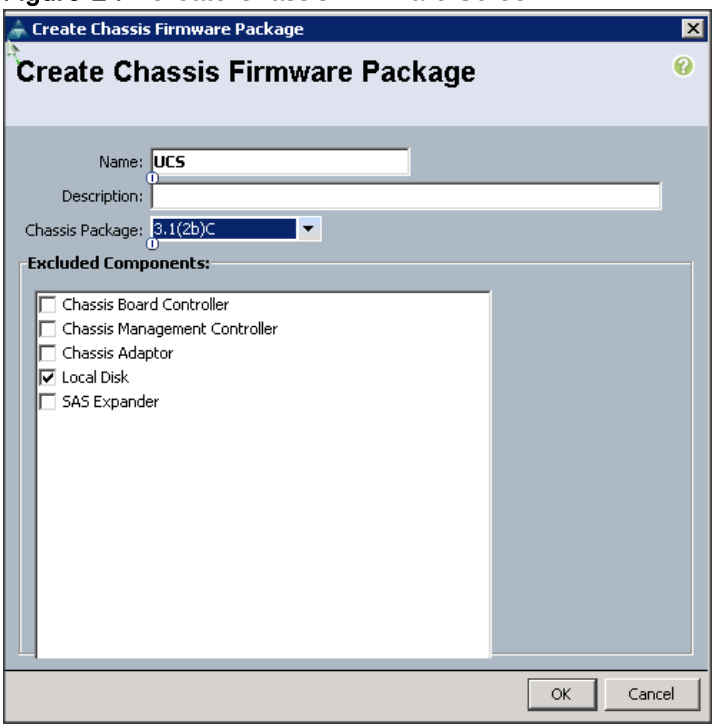
Figure 23 Chassis Firmware Packages



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.

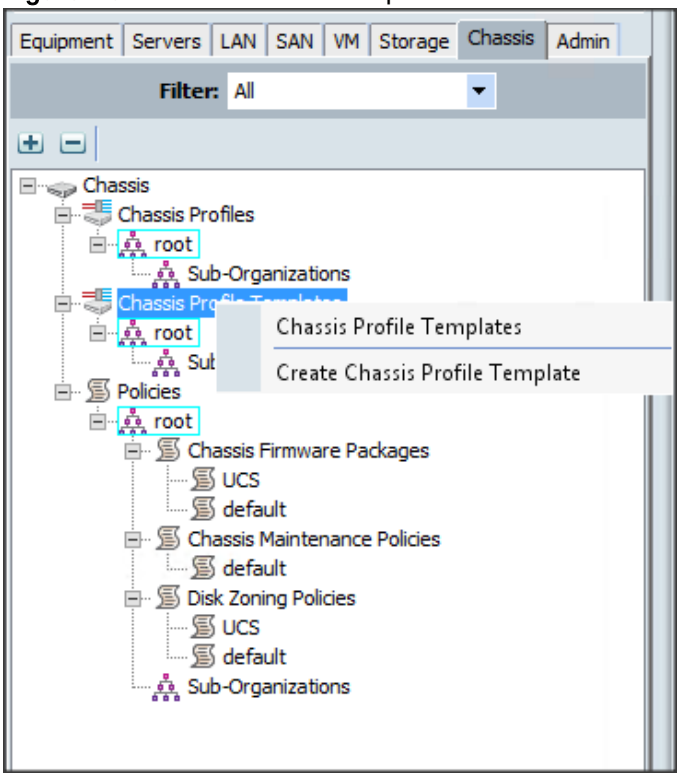
Figure 24 Create Chassis Firmware Screen



Creating Chassis Profiles from Template

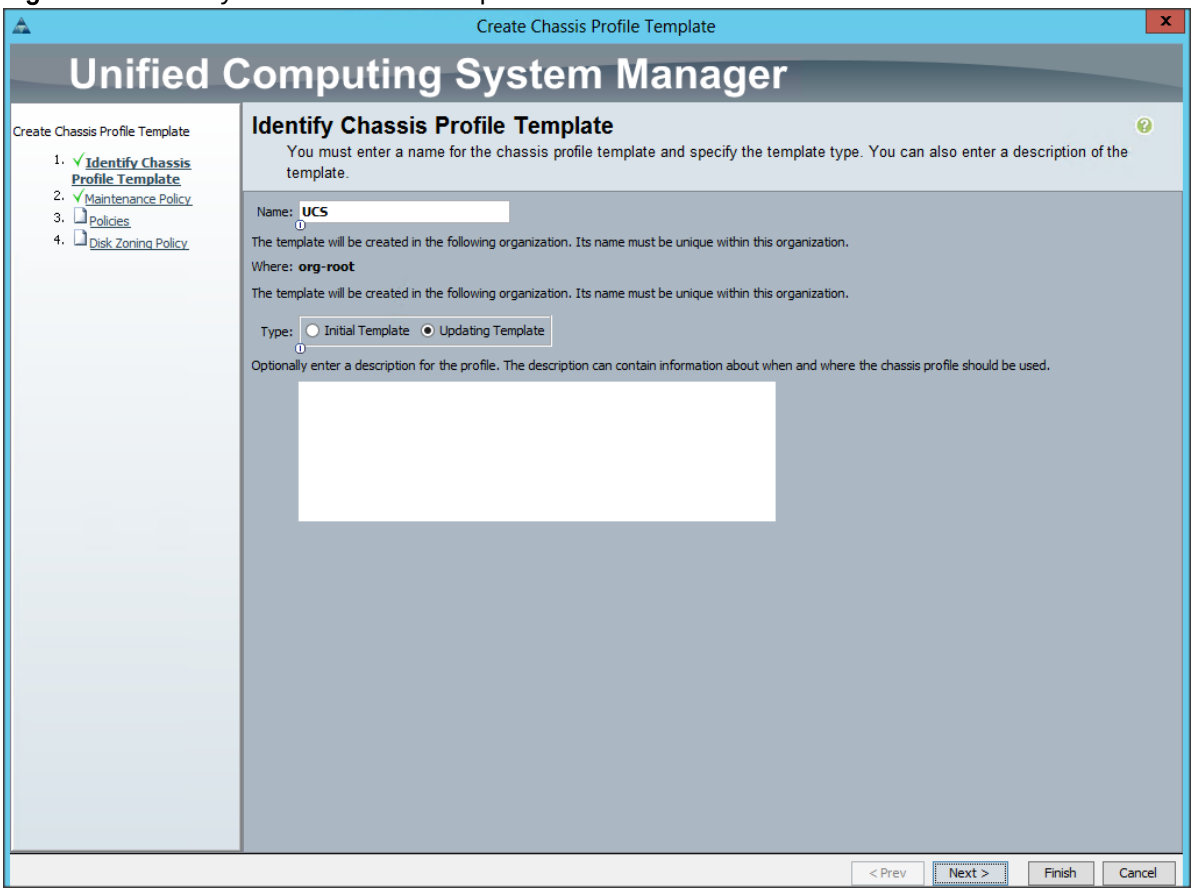
1. Under Chassis Profile Template, right click and click “Create Chassis Profile Templates” (Figure 25).

Figure 25 Chassis Profile Templates



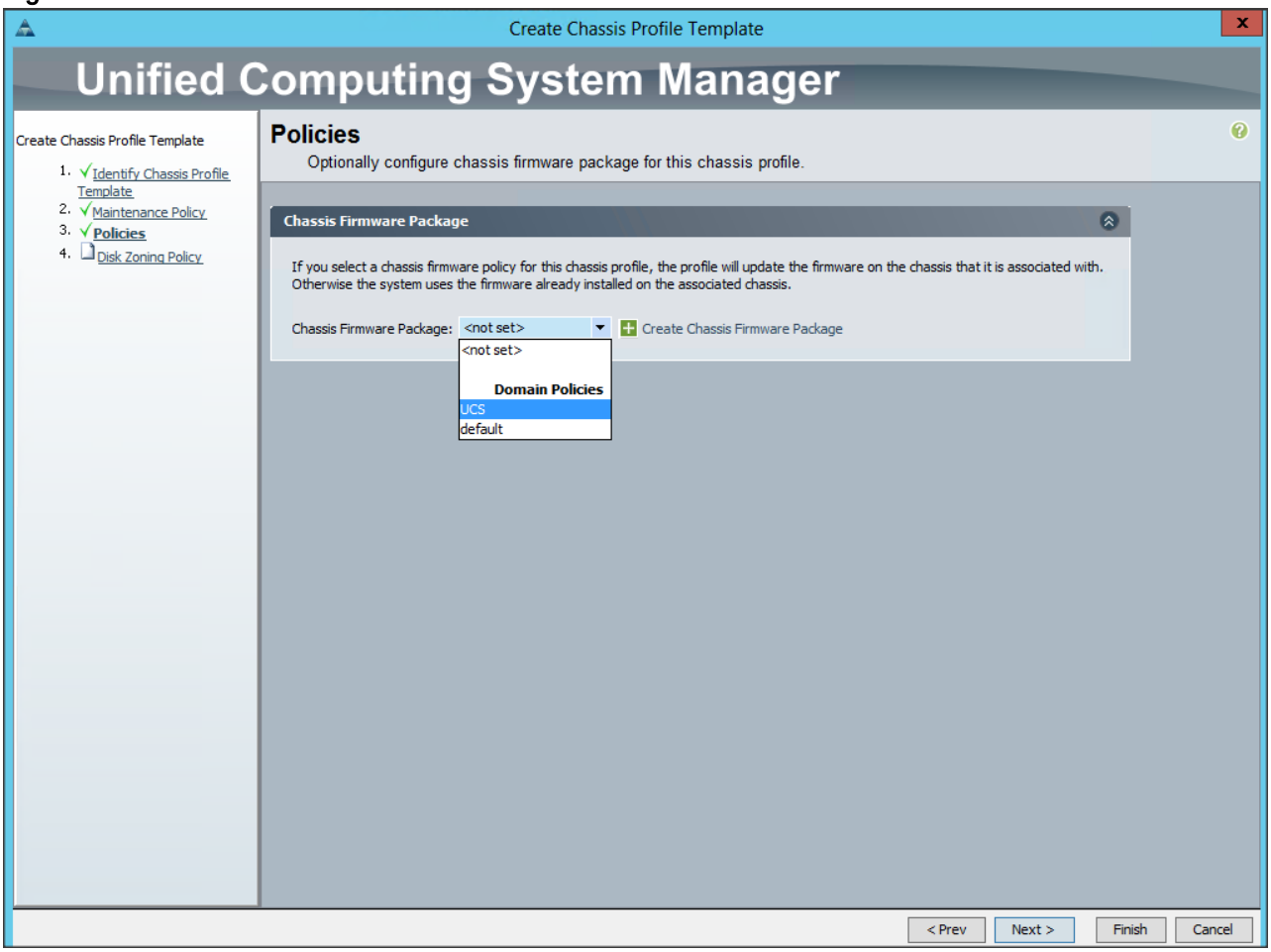
2. Enter the Name “UCS” and select “Updating Template” as the type, and click Next and Next again. (Figure 26)

Figure 26 Identify Chassis Profile Template



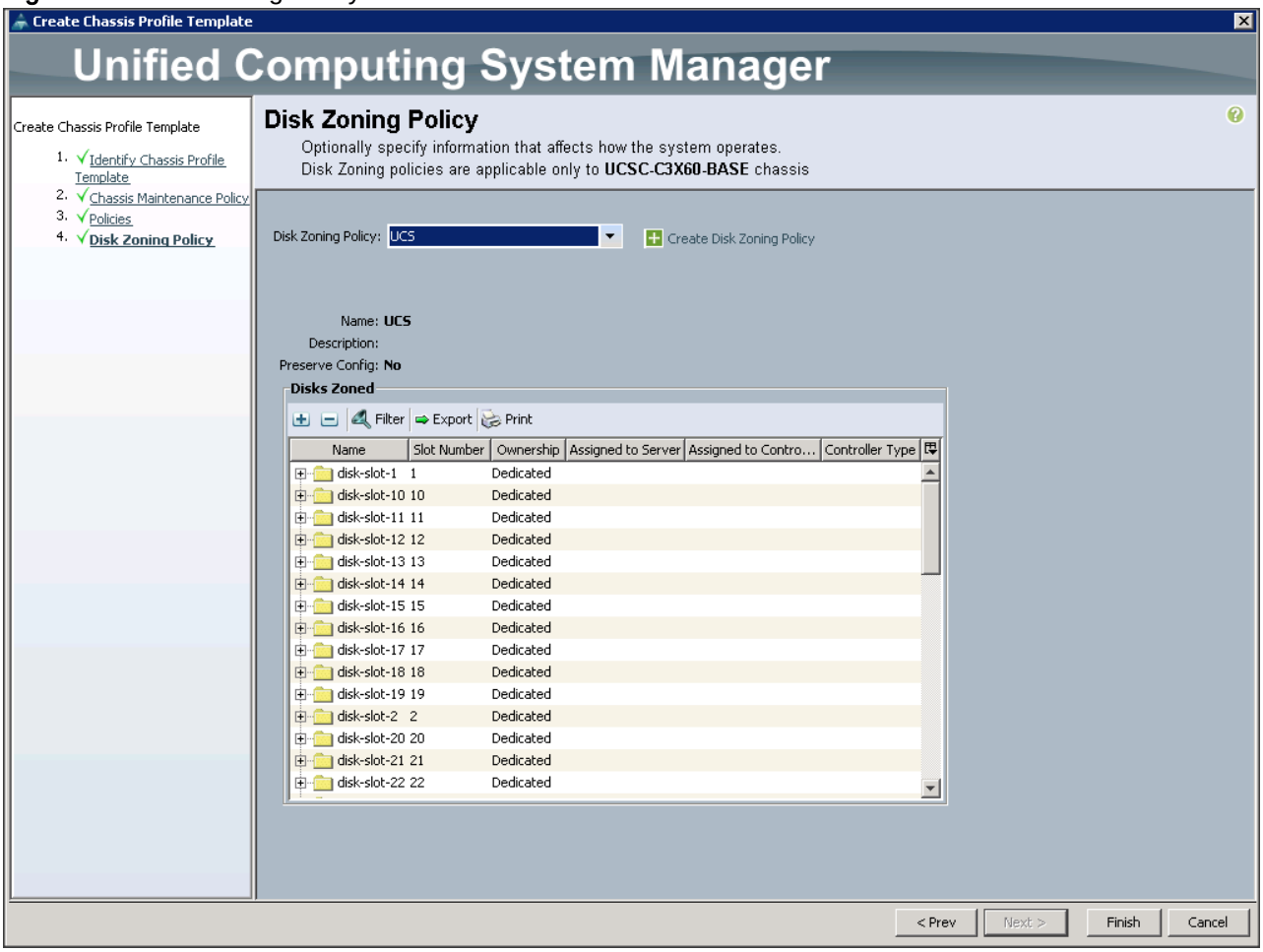
3. From the Chassis Firmware Package drop down list choose UCS and click Next. (Figure 27)

Figure 27 UCS Policies



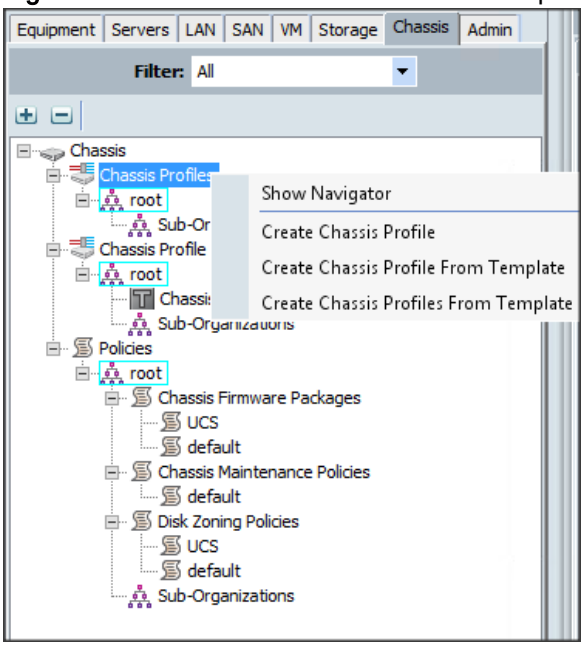
4. From the Disk Zoning Policy drop down list choose UCS and click Finish. (Figure 28)

Figure 28 Disk Zoning Policy



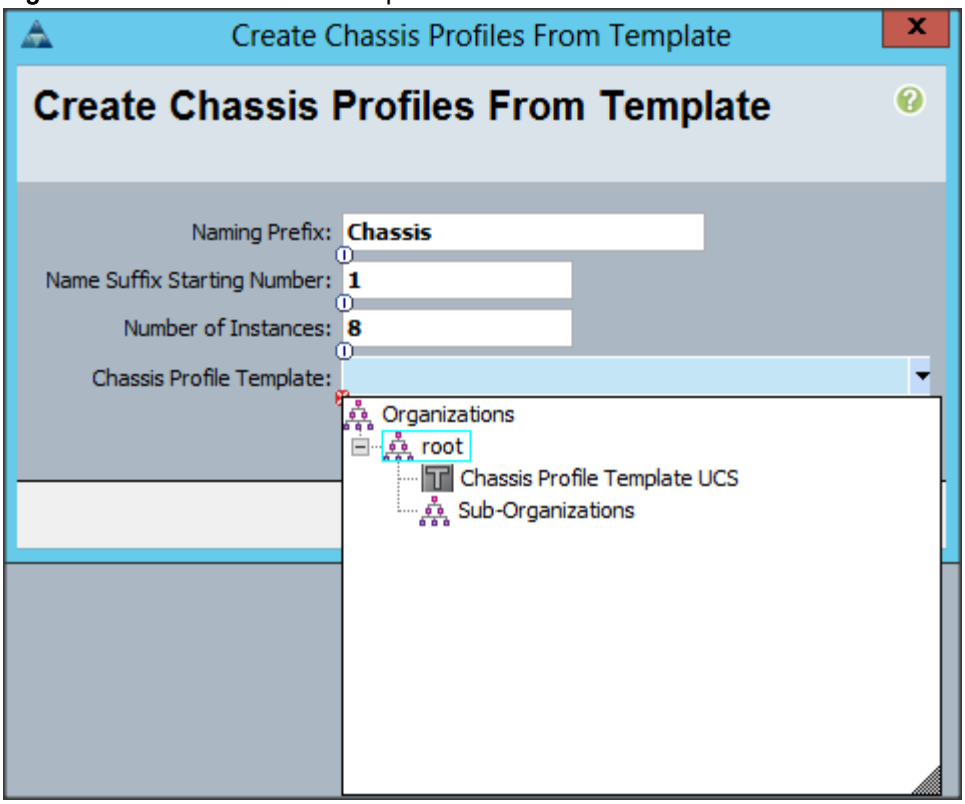
5. Right click on the Chassis Profiles and click “Create Chassis Profile from Templates” (Figure 29).

Figure 29 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 30)

Figure 30 Chassis Profile Template UCS

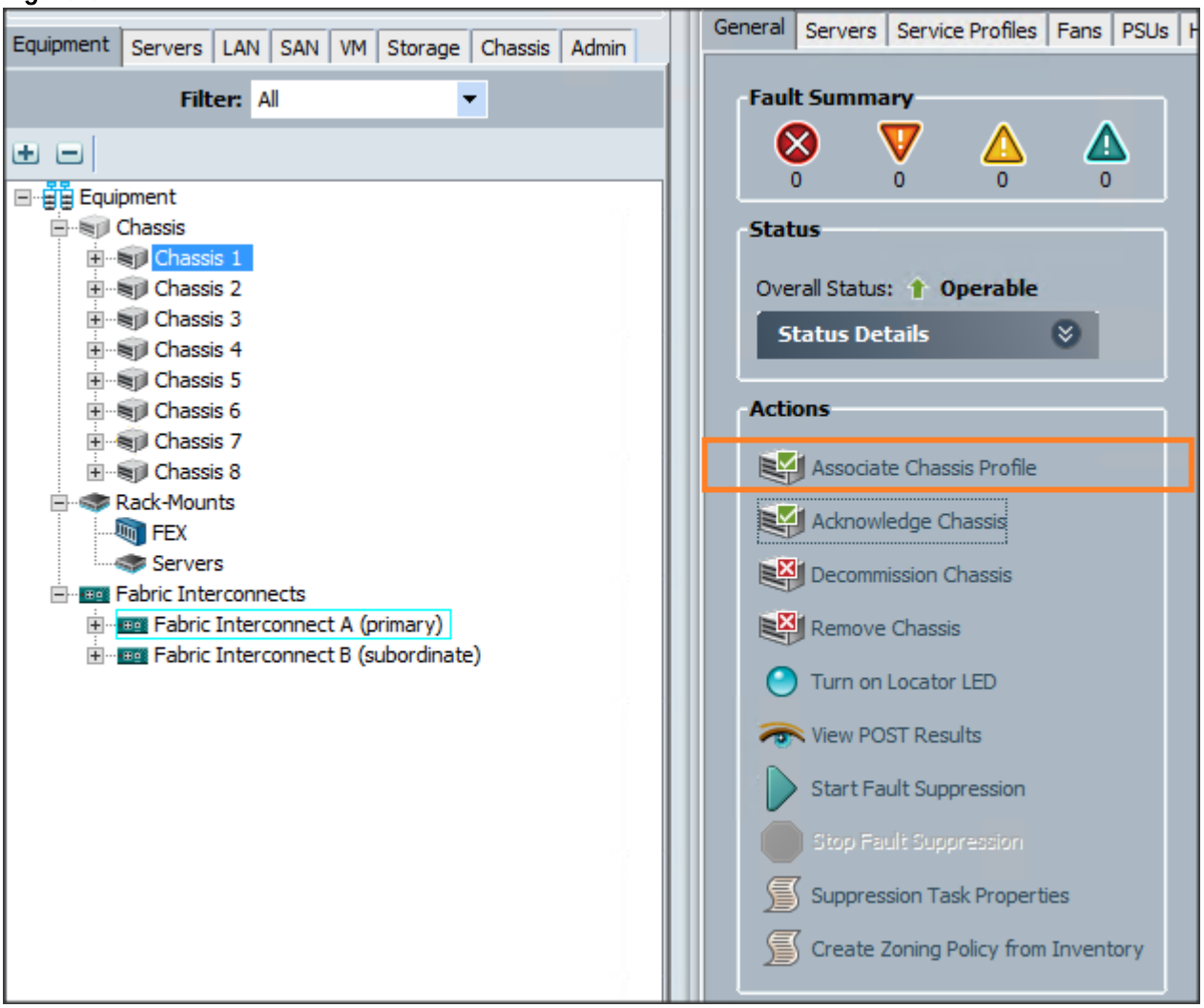


Associating Chassis Profiles to Individual Chassis

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

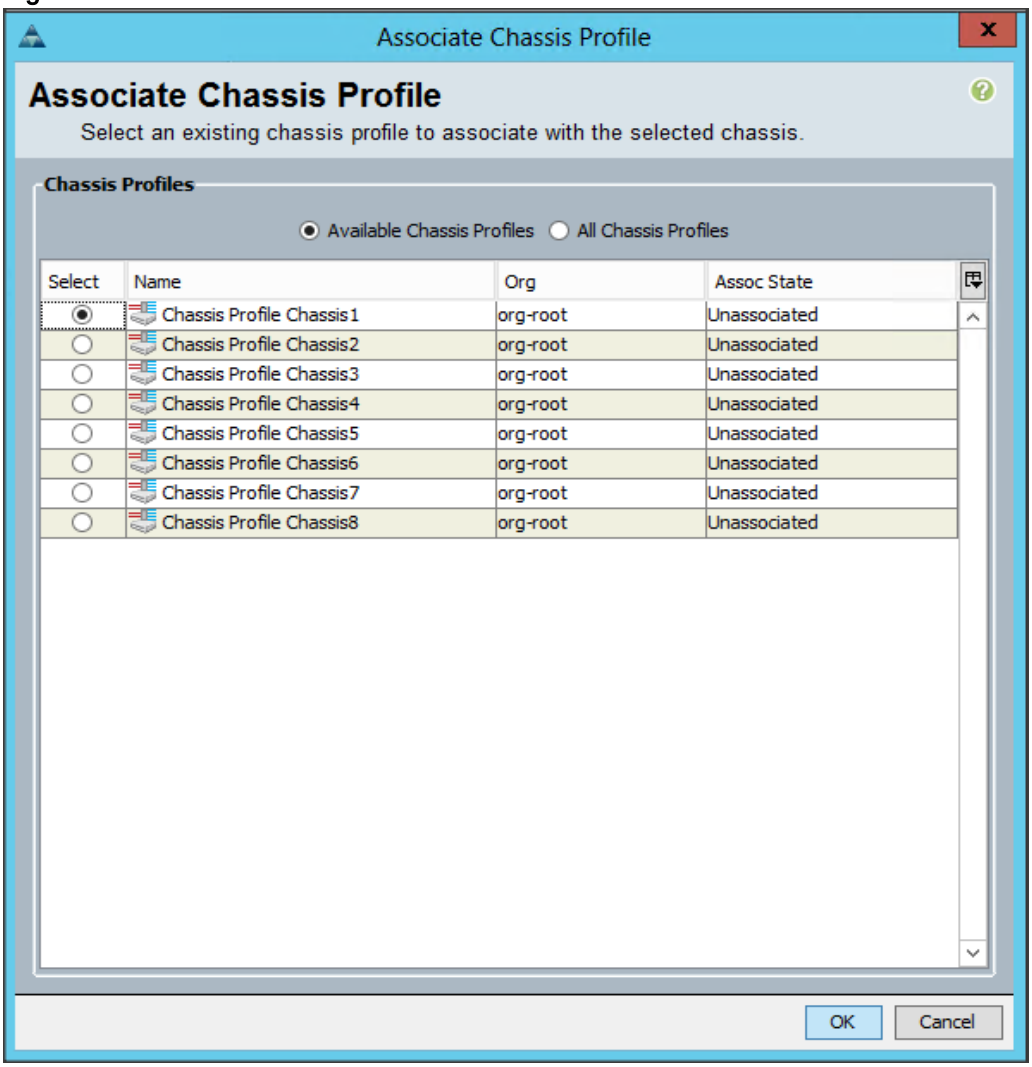
2. Select the Chassis and click Associate Chassis Profile. (Figure 31)

Figure 31 Associate Chassis Profiles



3. Select “Chassis Profile Chassis 1” and click “OK”. (Figure 32)

Figure 32 Associate Chassis Profile

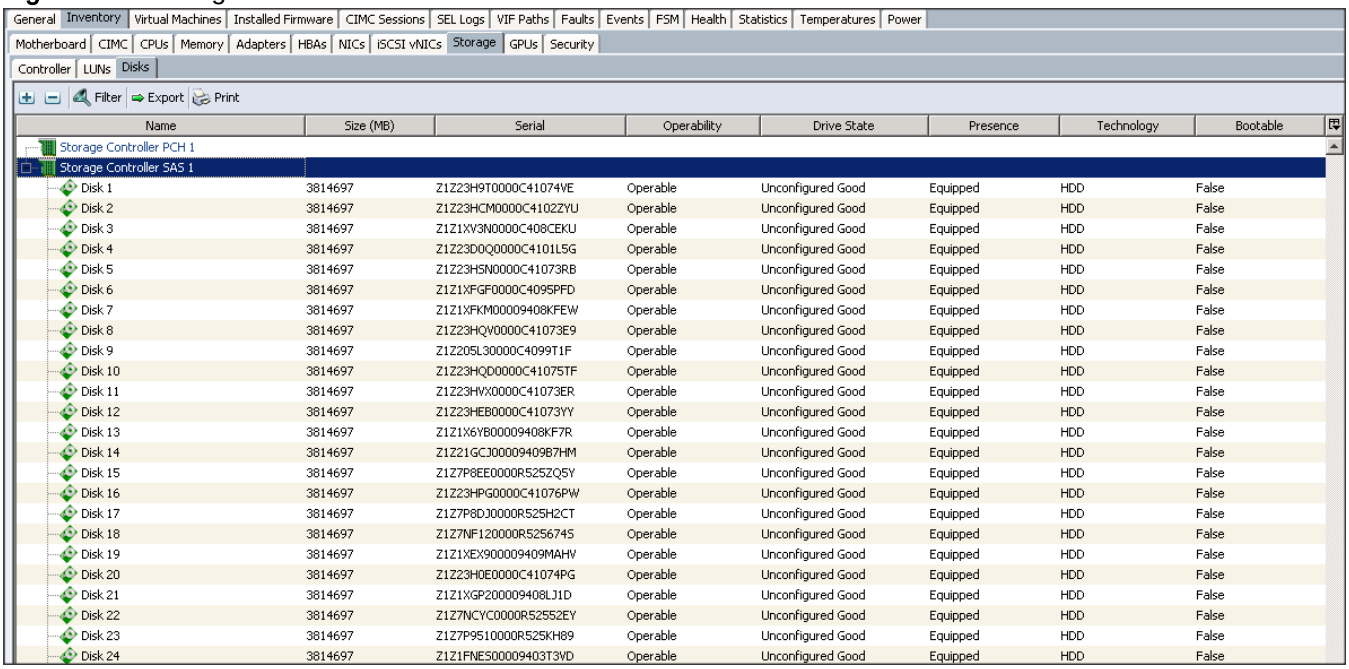


4. Repeat steps 2 and 3 for the rest of the chassis.

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 33)

Figure 33 Storage Controller SAS 1

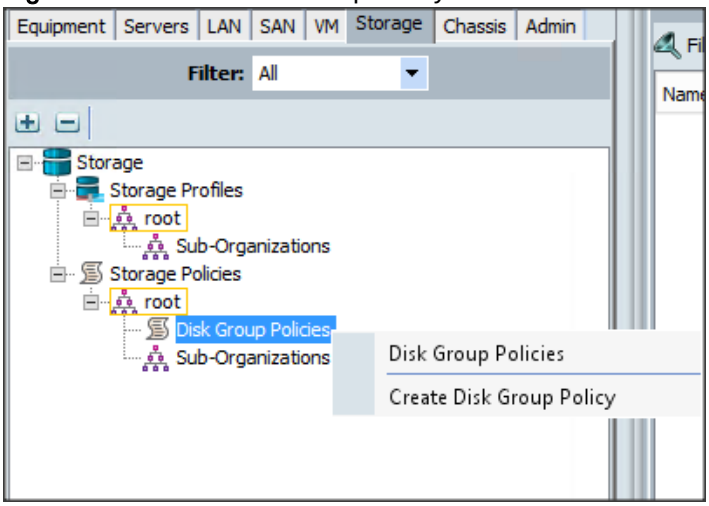


Name	Size (MB)	Serial	Operability	Drive State	Presence	Technology	Bootable
Storage Controller PCH 1							
Storage Controller SAS 1							
Disk 1	3814697	Z1Z23H9T0000C41074VE	Operable	Unconfigured Good	Equipped	HDD	False
Disk 2	3814697	Z1Z23HCM0000C4102ZYU	Operable	Unconfigured Good	Equipped	HDD	False
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	Z1Z7NF120000R525674S	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 34)

Figure 34 Create Disk Group Policy



2. In the Create Disk Policy window, configure the following parameters and click OK. (Figure 35)

    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

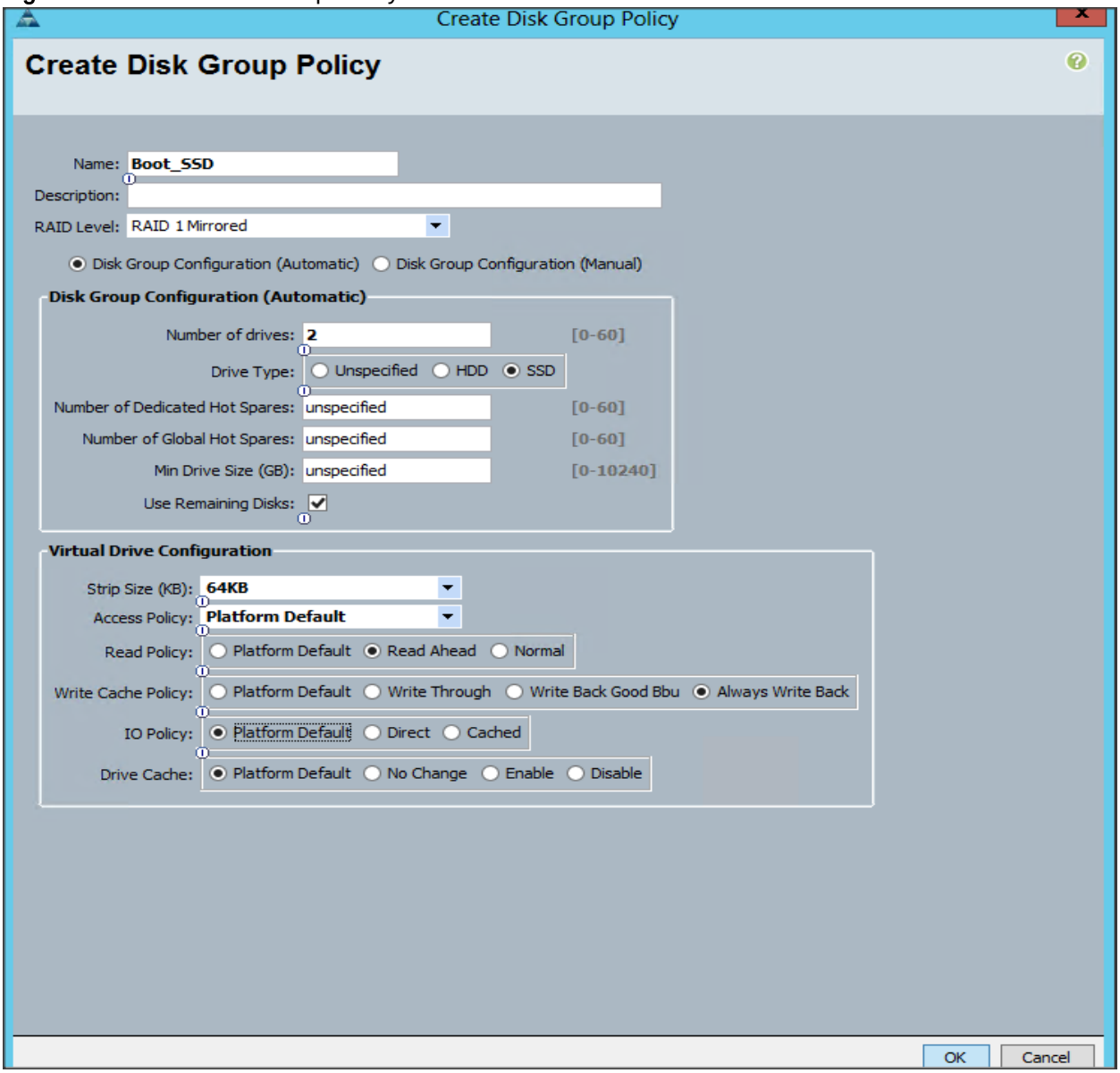
h. Access Policy = Platform Default

i. Read Policy = Read Ahead

j. Write Cache Policy = Always Write Back

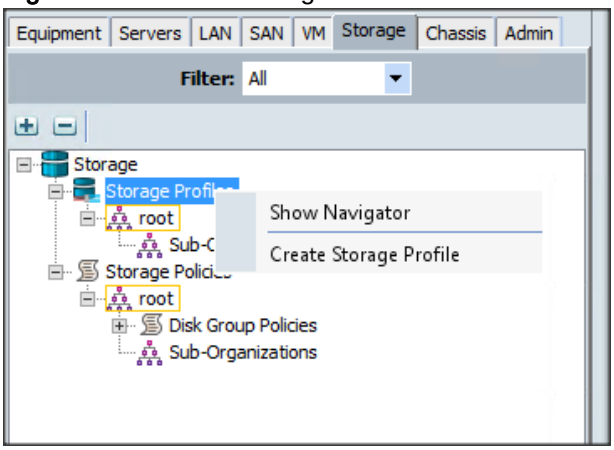
k. IO Policy and Drive Cache = Platform Default

Figure 35 Create Disk Group Policy



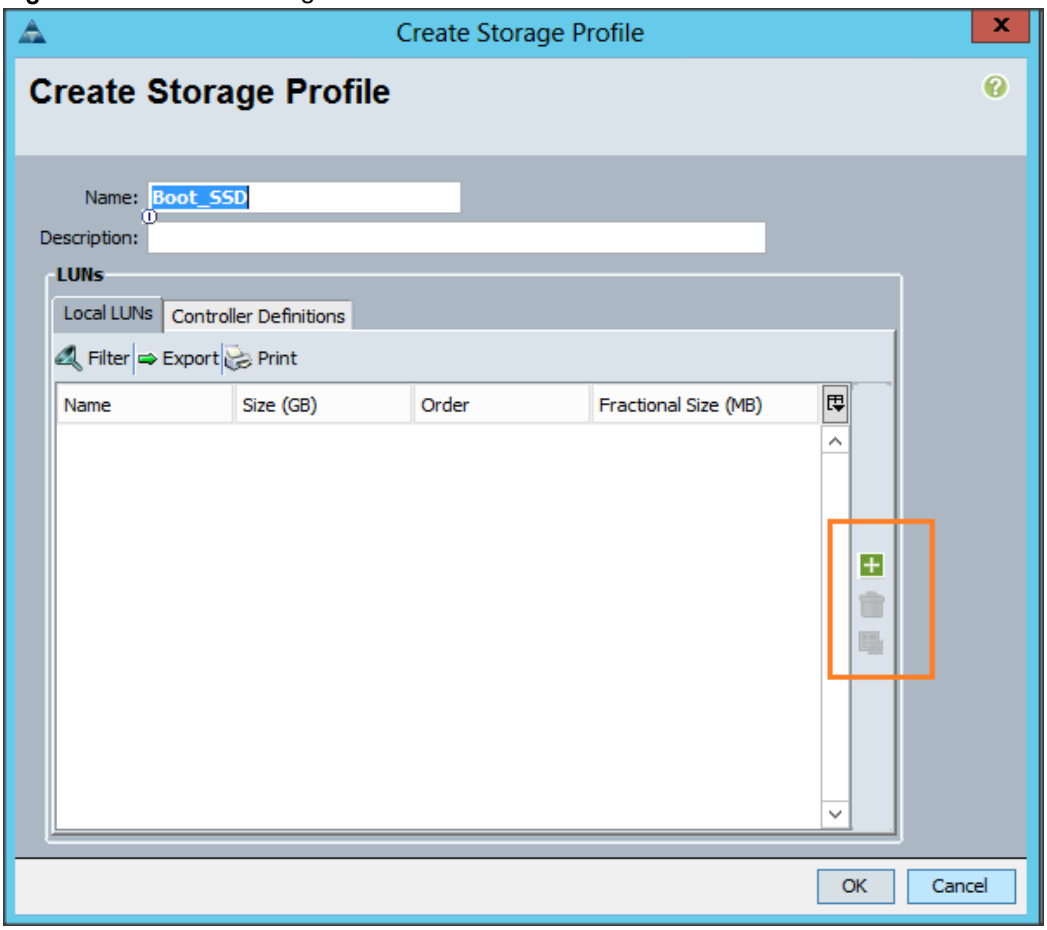
3. Click on the Storage tab. Right click on Storage Profile and click Create Storage Profile. (Figure 36)

Figure 36 Create Storage Profile



4. Enter “Boot_SSD” in the name field. Under Local LUNs click “+” to add local lun. (Figure 37)

Figure 37 Create Storage Profile

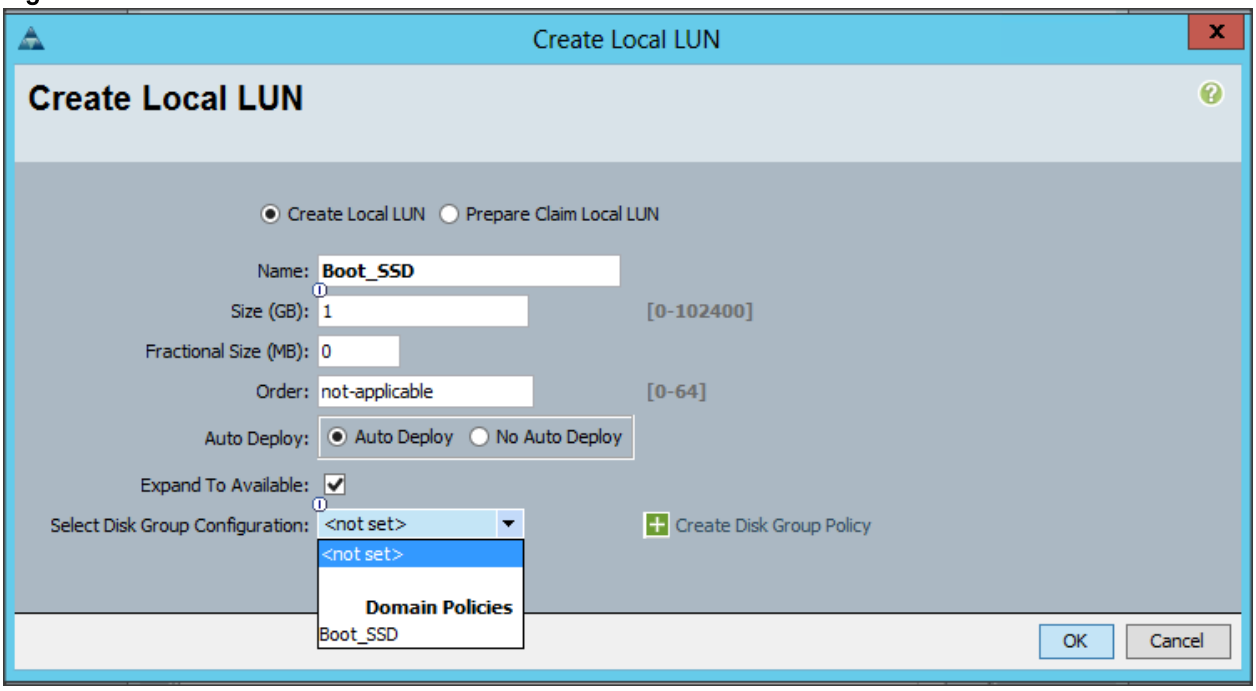


5. In the Create Local LUN window, enter the name “Boot_SSD”. (Figure 38)

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 38 Create Local LUN



Creating Pools for Service Profile Templates

Creating MAC Address Pools

To create MAC address pools, complete the following steps (Figure 39):

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.

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 40).
10. Specify a size of the MAC address pool, which is sufficient to support the available server resources.
11. Click `OK`.

Figure 39 Creating a MAC Pool

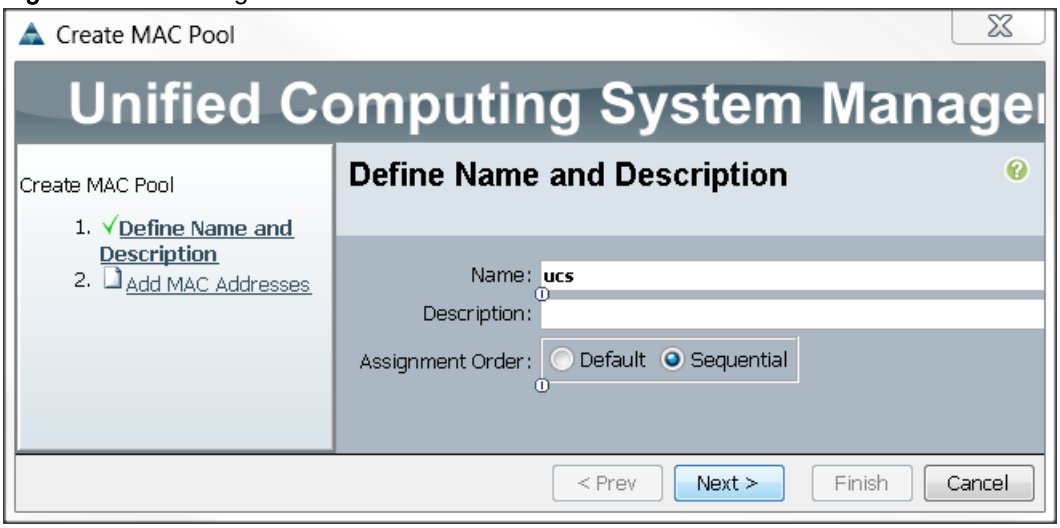
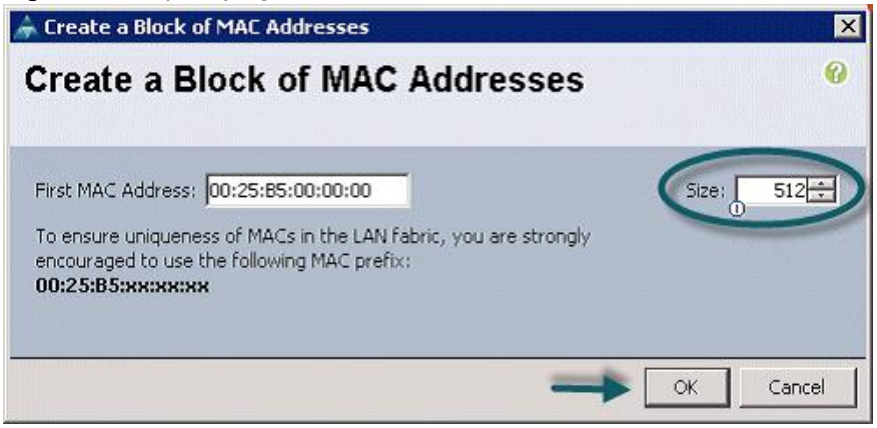
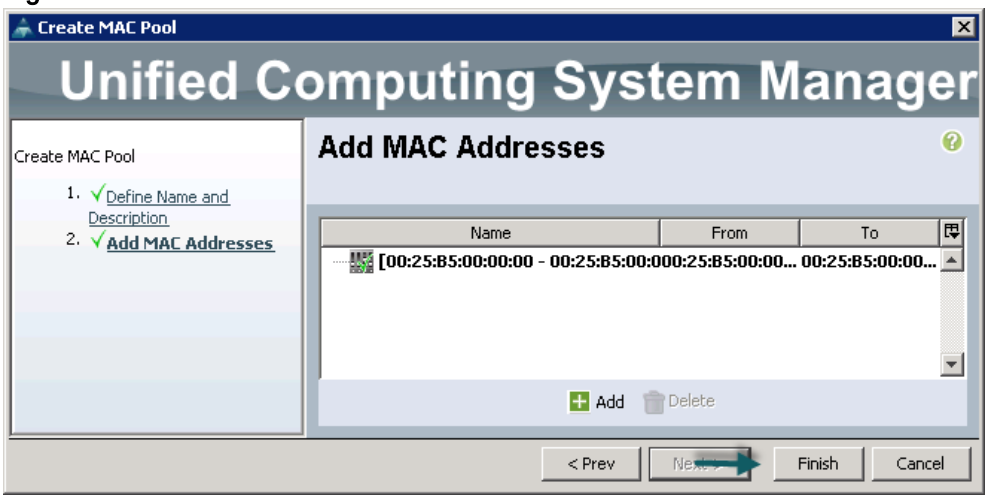


Figure 40 Specifying First MAC Address and Size



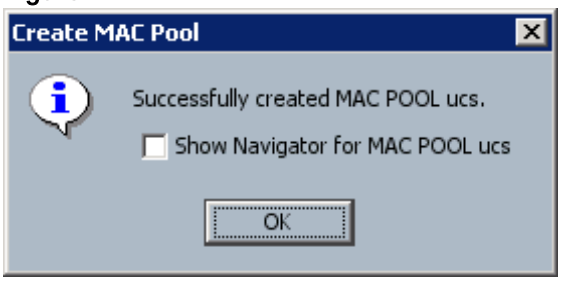
12. Click `Finish` (Figure 41).

Figure 41 Add MAC Addresses



13. When the message box displays, click `OK` (Figure 42).

Figure 42 Create MAC Pool



Creating a Server Pool

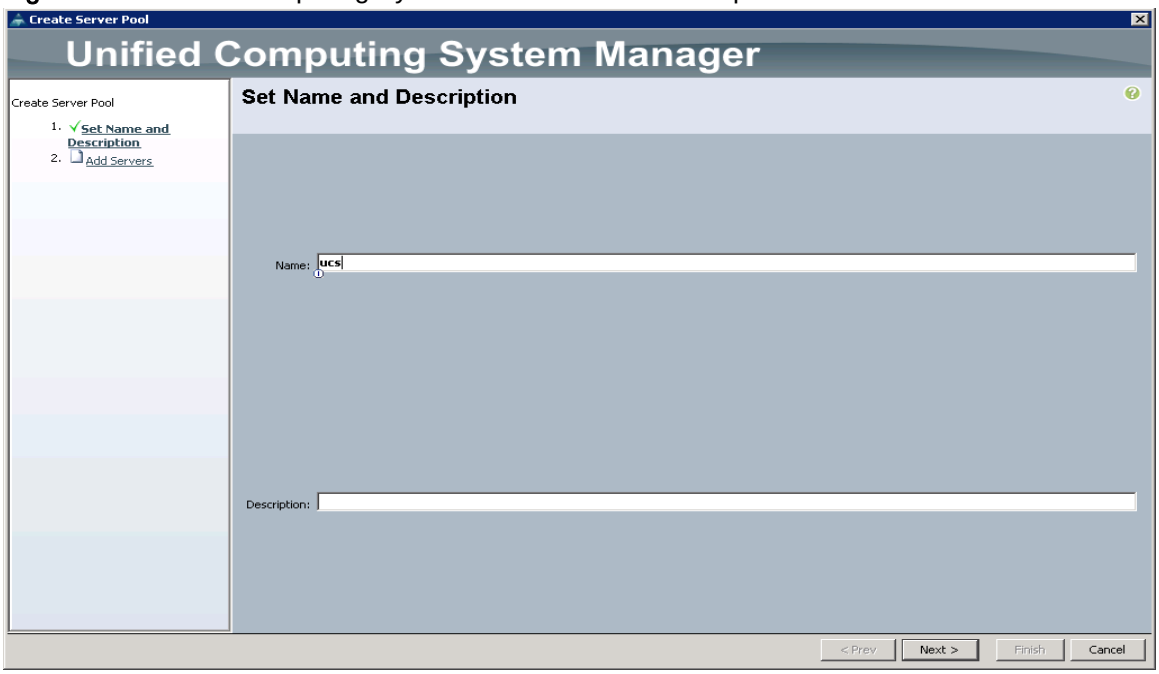
A server pool contains a set of servers. These servers typically share the same characteristics. Those characteristics can be their location in the chassis, or an attribute such as server type, amount of memory, local storage, type of CPU, or local drive configuration. A server can be manually assigned to a server pool, or server pool policies and server pool policy qualifications can be used 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 Cisco UCS Manager GUI.
2. Select Pools > root.
3. Right-click the `Server Pools`.
4. Select Create Server Pool.
5. Enter the 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 Unified Computing System/Set Name and Description

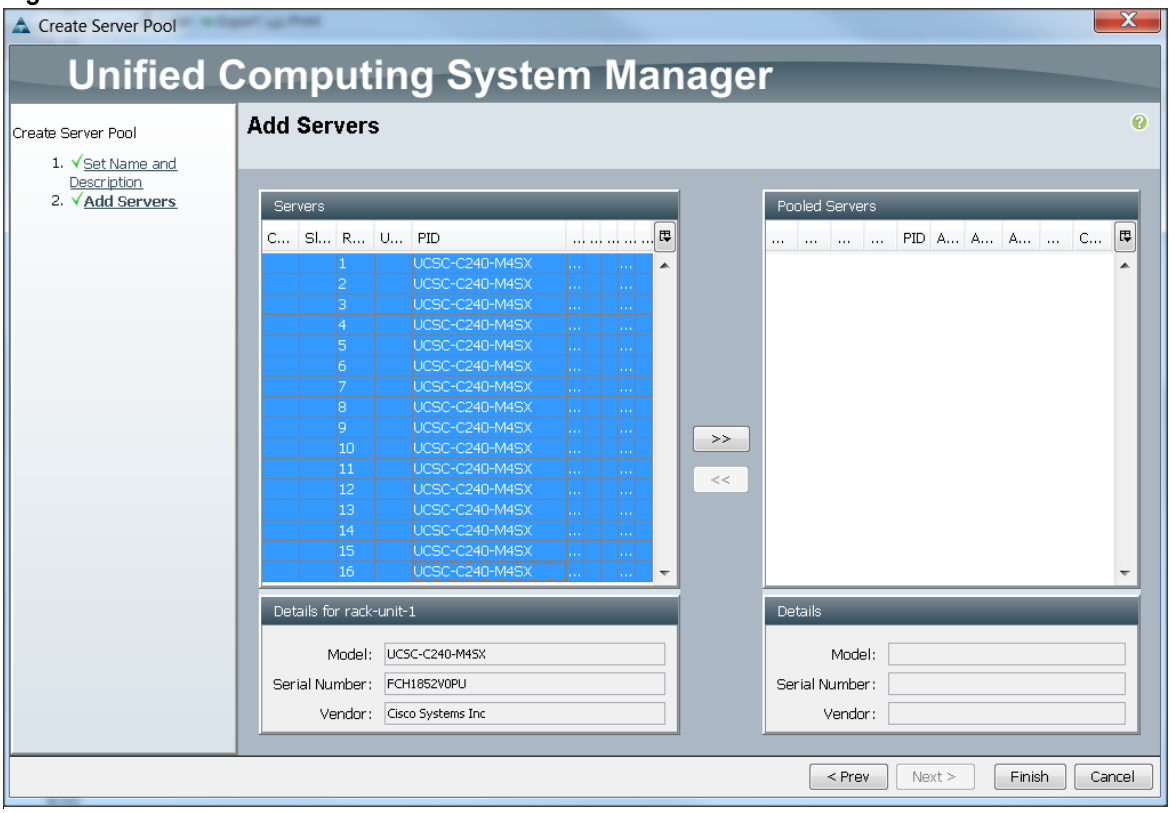


8. Select all the Cisco UCS C240M4SX servers to be added to the server pool that was previously created (ucs), then `Click >>` to add them to the pool (Figure 44).

9. Click `Finish`.

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

Figure 44 Add Servers to UCS



Creating Policies for Service Profile Templates

Creating Host Firmware Package Policy

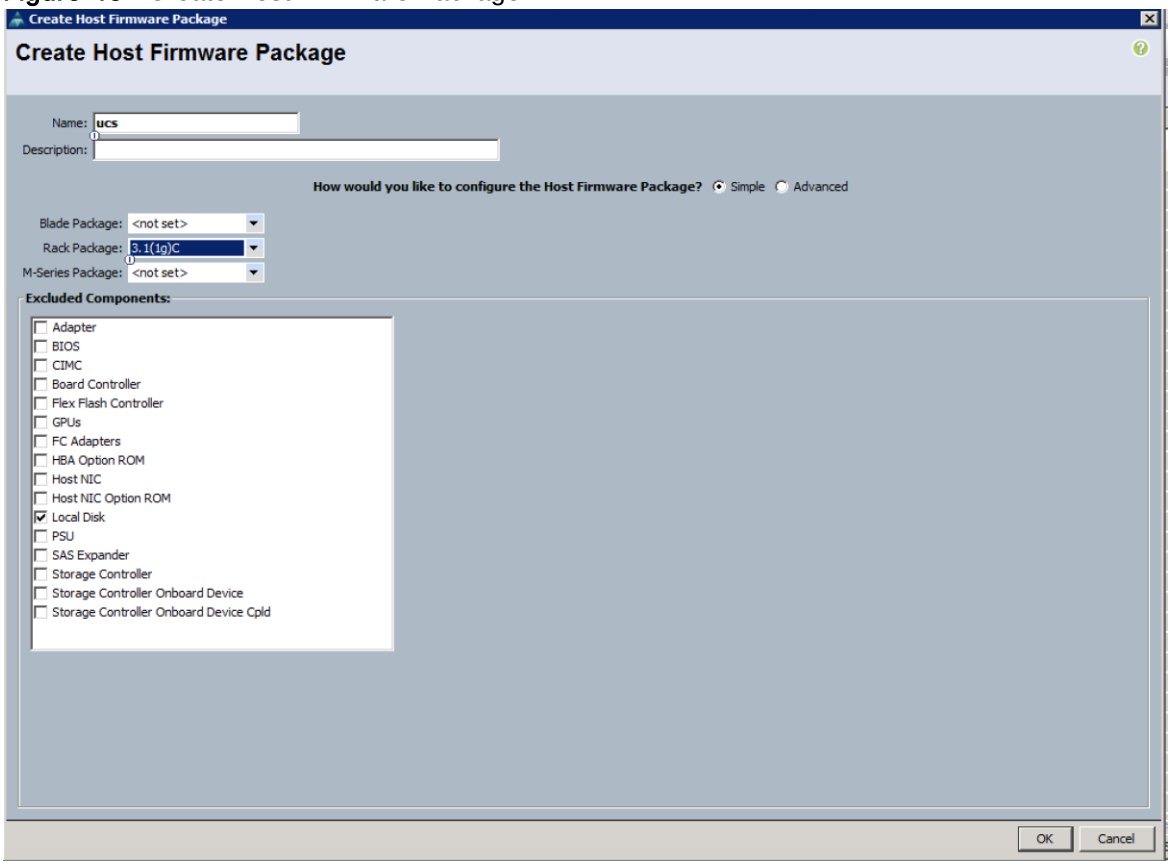
Firmware management policies allow the administrator to select the corresponding packages for a given server configuration. These include adapters, BIOS, board controllers, FC adapters, HBA options, and storage controller properties as applicable.

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 Cisco UCS Manager GUI.
2. Select Policies > root.
3. Right-click Host Firmware Packages.
4. Select Create Host Firmware Package (Figure 45).
5. Enter the required Host Firmware package name (ucs).
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 45 Create Host Firmware Package



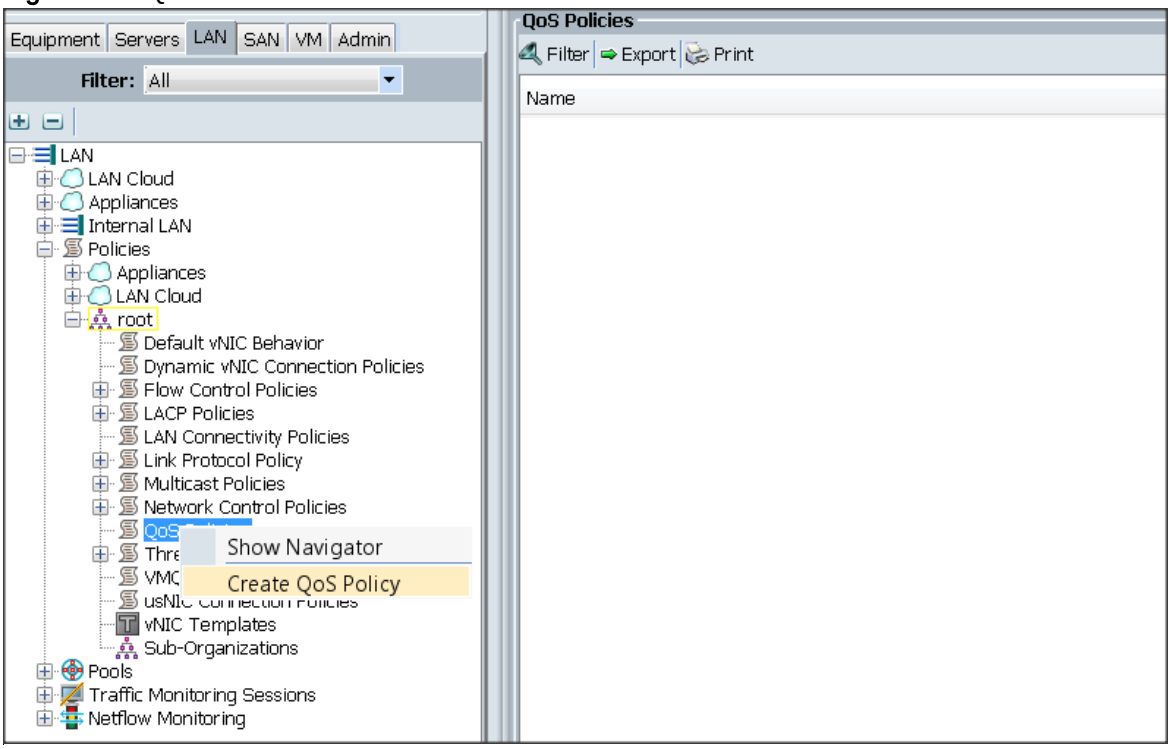
Creating QoS Policies

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

Platinum Policy

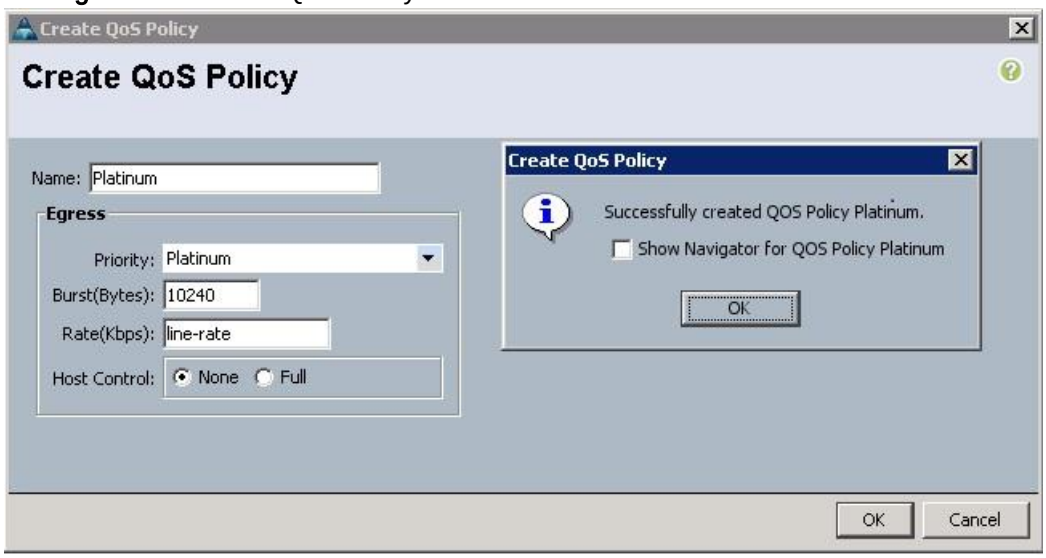
1. Select the LAN tab in the left pane in the Cisco UCS Manager GUI (Figure 46).
2. Select Policies > root.
3. Right-click QoS Policies.
4. Select Create QoS Policy.

Figure 46 QoS Policies



5. Enter `Platinum` as the name of the policy.
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 47).

Figure 47 Create QoS Policy

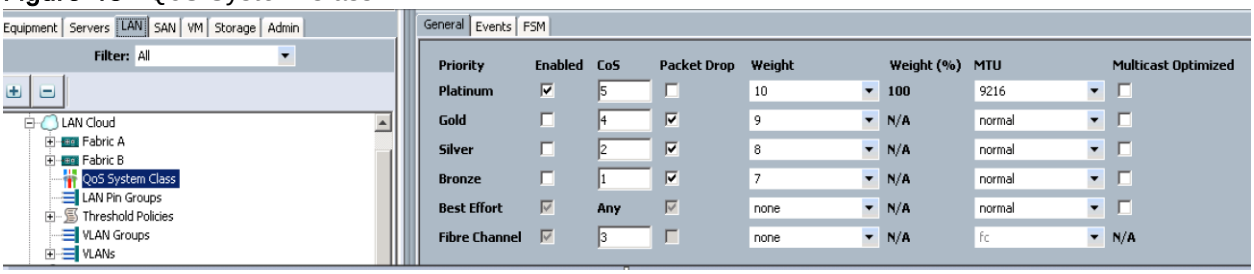


Setting Jumbo Frames

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

1. Select the `LAN` tab in the left pane in the Cisco UCSM GUI.
2. Select `LAN Cloud > QoS System Class` (Figure 48).
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 48 QoS System Class

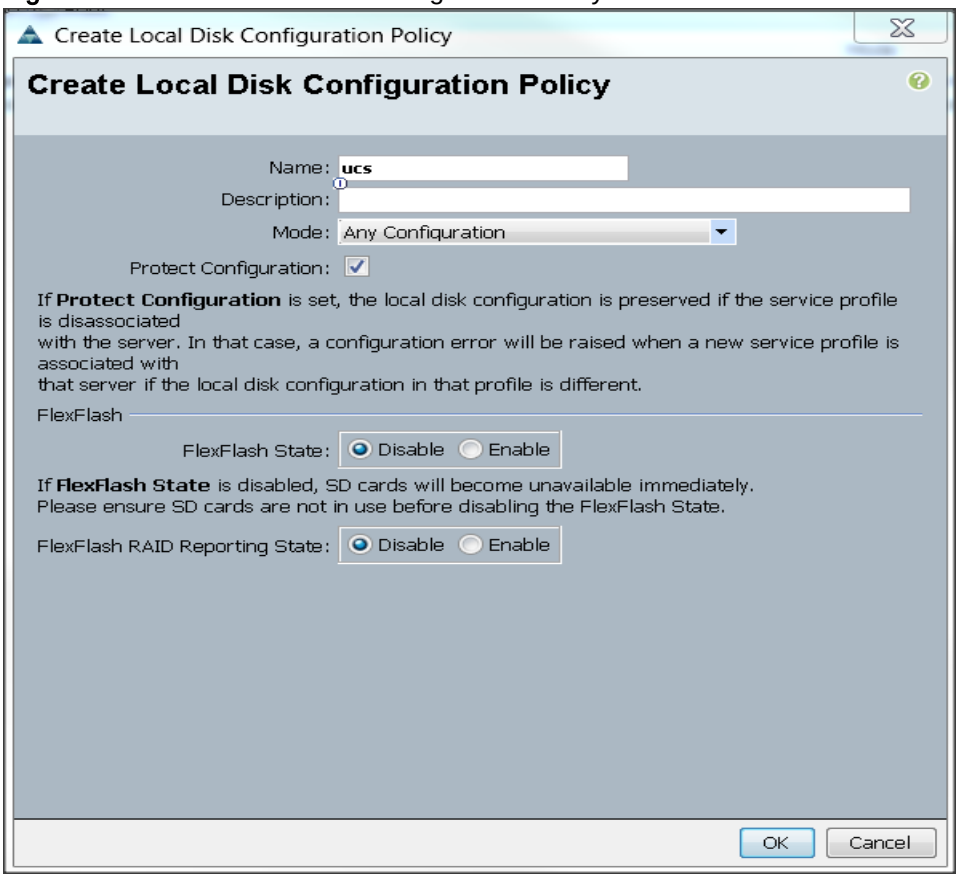


Creating the Local Disk Configuration Policy

To create a 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 Cisco UCS Manager GUI.
2. Go to `Policies > root`.
3. Right-click `Local Disk Config Policies`.
4. Select `Create Local Disk Configuration Policy`.
5. Enter `ucs` as the local disk configuration policy name (Figure 49).
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 49 Create Local Disk Configuration Policy



Creating 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:

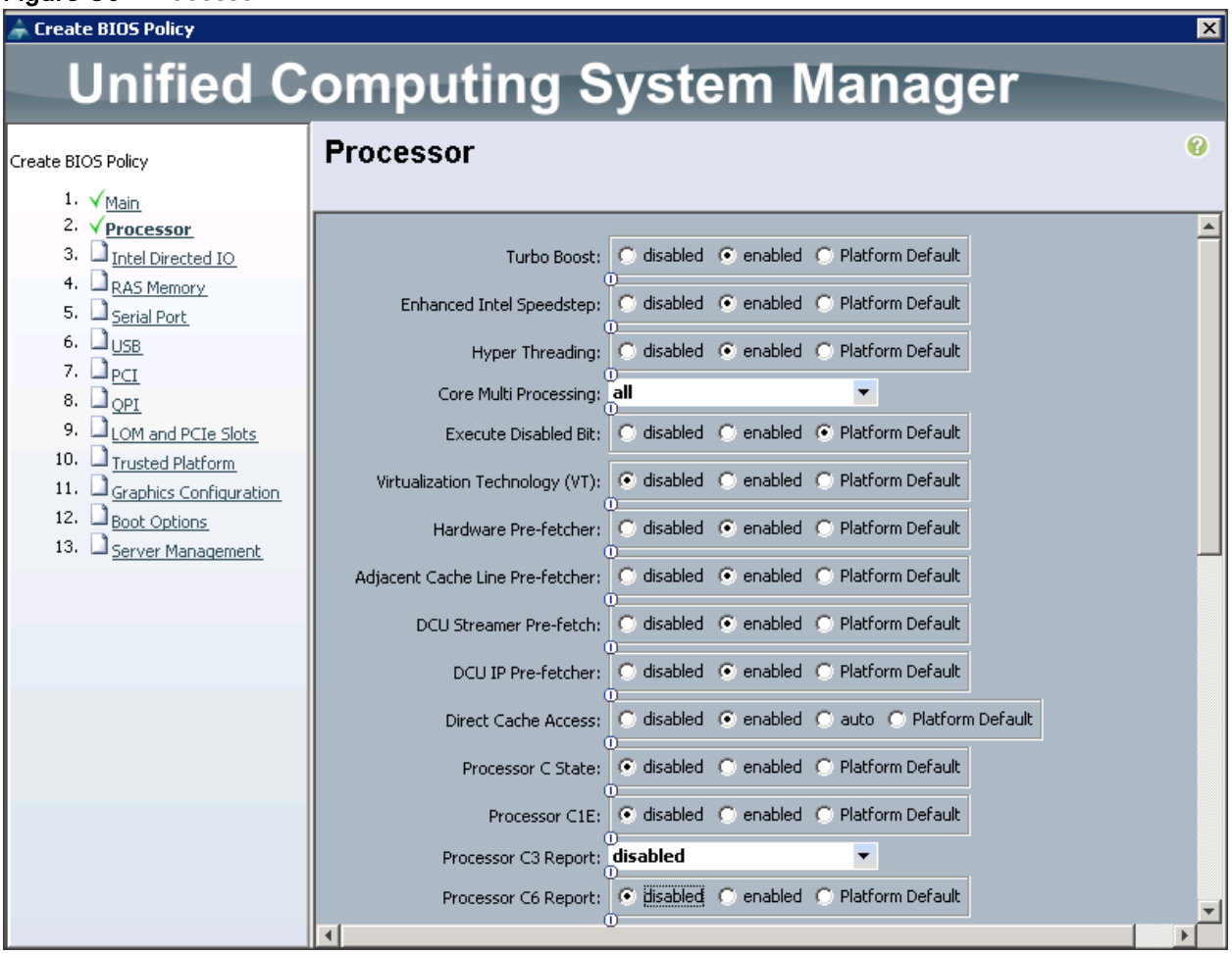
1. Select the `Servers` tab in the left pane in the Cisco UCS Manager GUI.
2. Select Policies > root.
3. Right-click `BIOS Policies`.
4. Select Create BIOS Policy.

5. Enter the preferred BIOS policy `name` (`ucs`).

6. Change the BIOS settings as shown below.

7. Changes only need to be made in the `Processor` (Error! Reference source not found.) `and RAS Memory settings` (Figure 51).

Figure 50 Processor



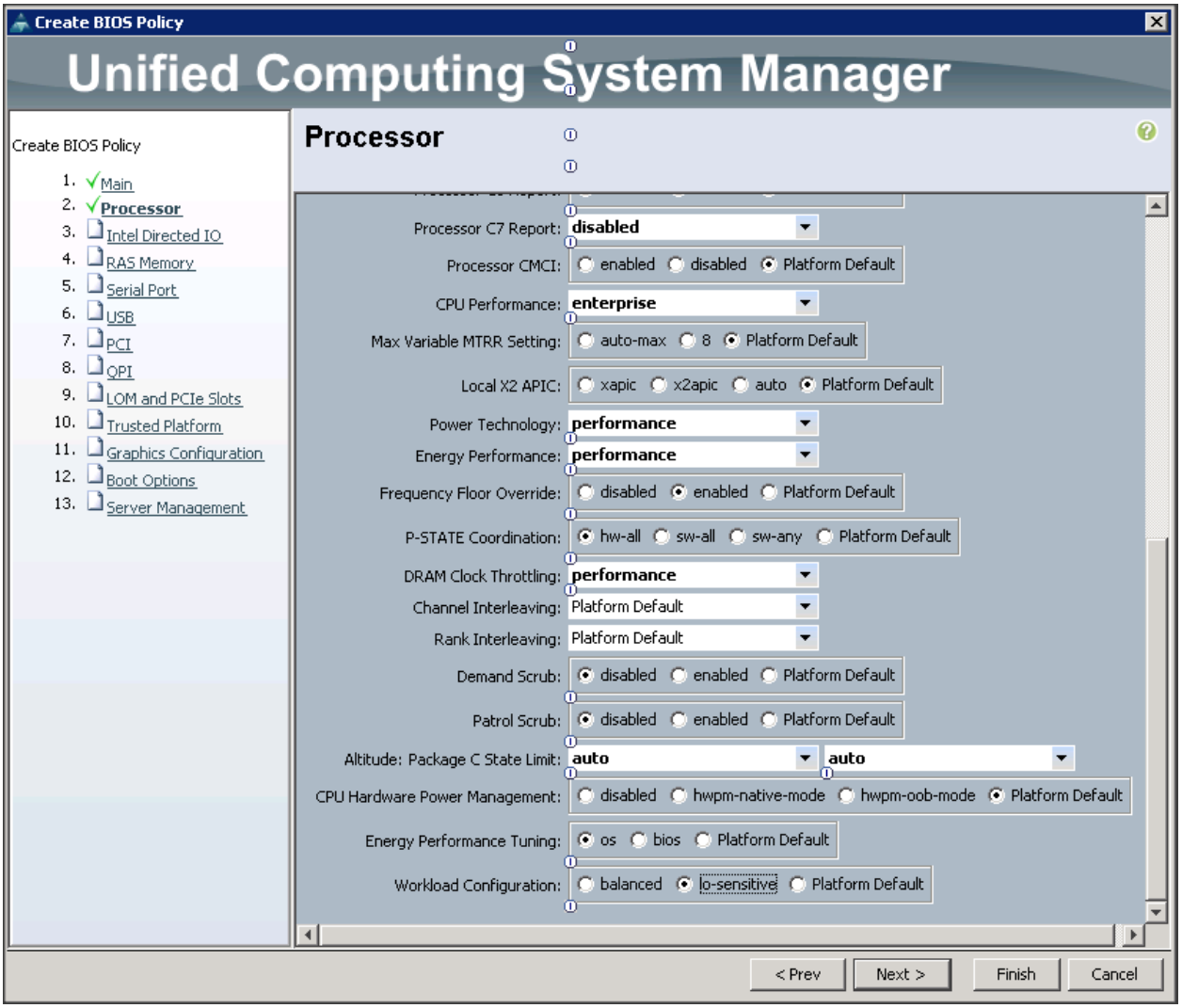
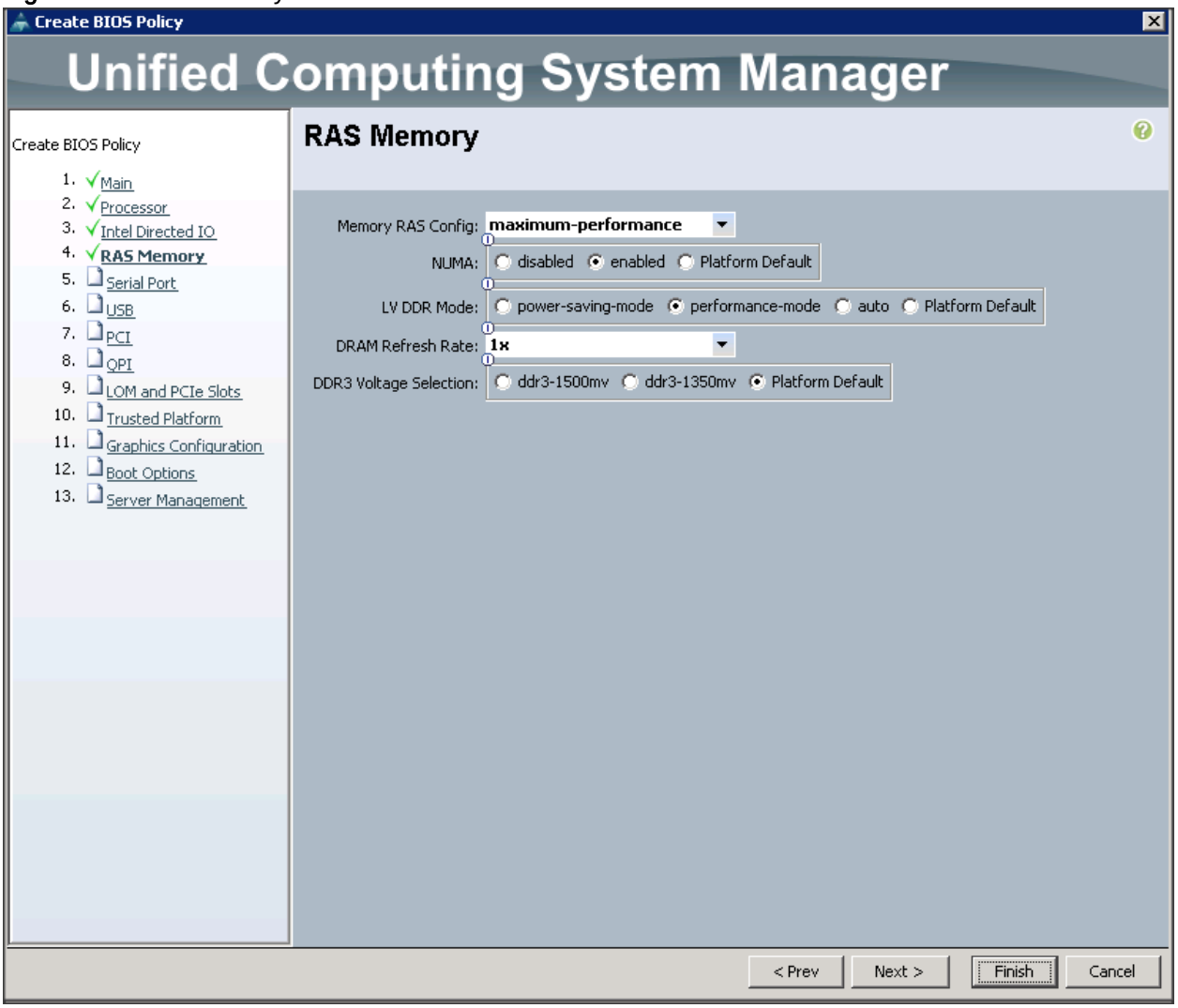

Create BIOS Policy
Unified Computing System Manager
Create BIOS Policy
1. Main
2. Processor
3. Intel Directed IO
4. RAS Memory
5. Serial Port
6. USB
7. PCI
8. QPI
9. LOM and PCIe Slots
10. Trusted Platform
11. Graphics Configuration
12. Boot Options
13. Server Management
Processor
Processor C7 Report: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

Figure 51 RAS Memory

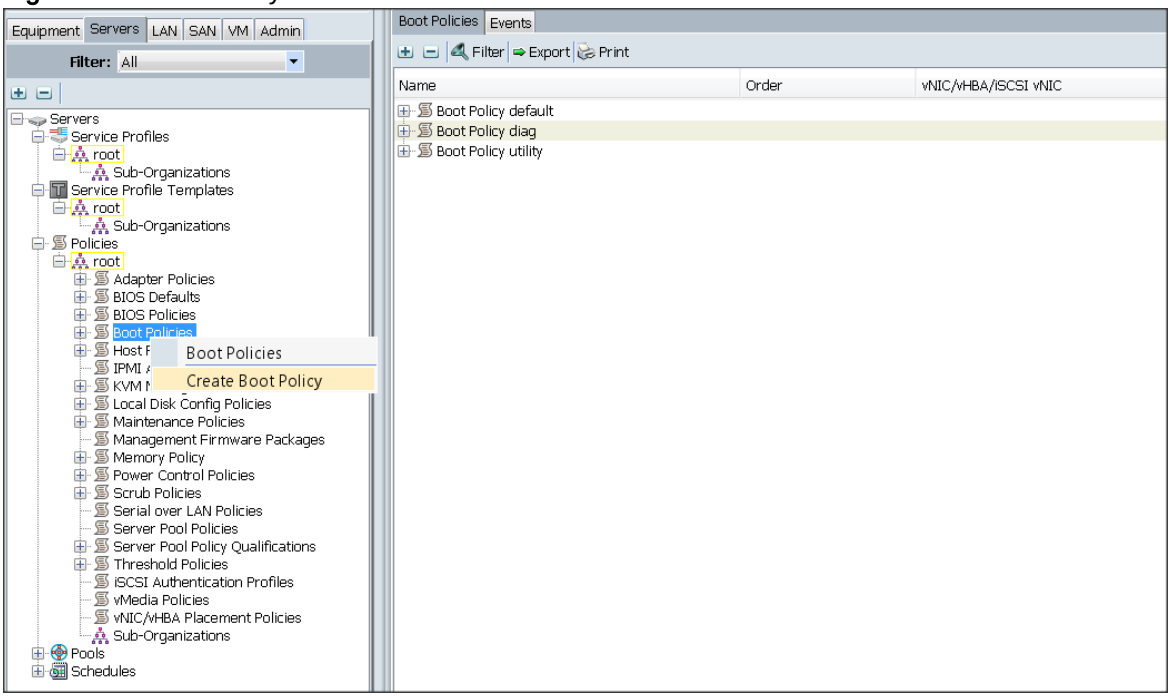


Creating the Boot Policy

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 Cisco UCS Manager GUI.
2. Select Policies > root.
3. Right-click the `Boot Policies`.
4. Select Create Boot Policy (Figure 52).

Figure 52 Boot Policy Events



5. Enter `ucs` for the boot policy name.
6. (Optional) enter a description for the boot policy.
7. Keep the Reboot on Boot Order Change check box unchecked.
8. Keep Enforce vNIC/vHBA/iSCSI Name check box checked.
9. Keep Boot Mode Default (Legacy).
10. Expand Local Devices 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. (Error! Reference source not found.)

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

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

Figure 53 Add Local LUN Image Path

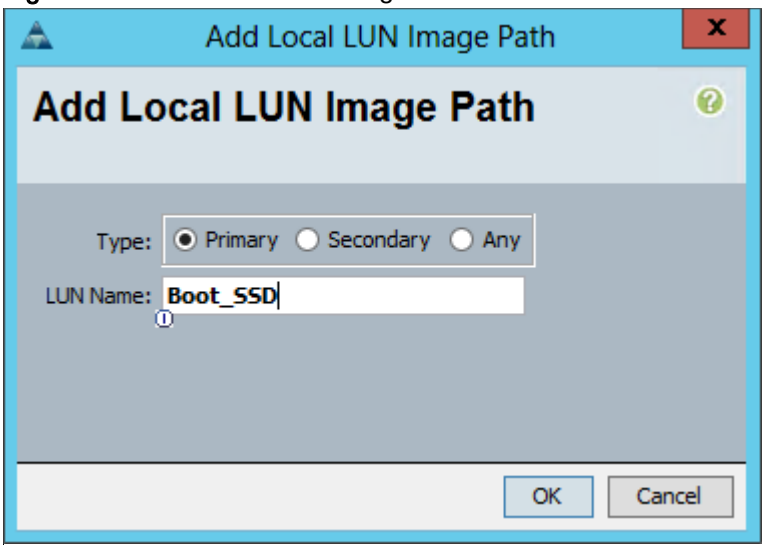
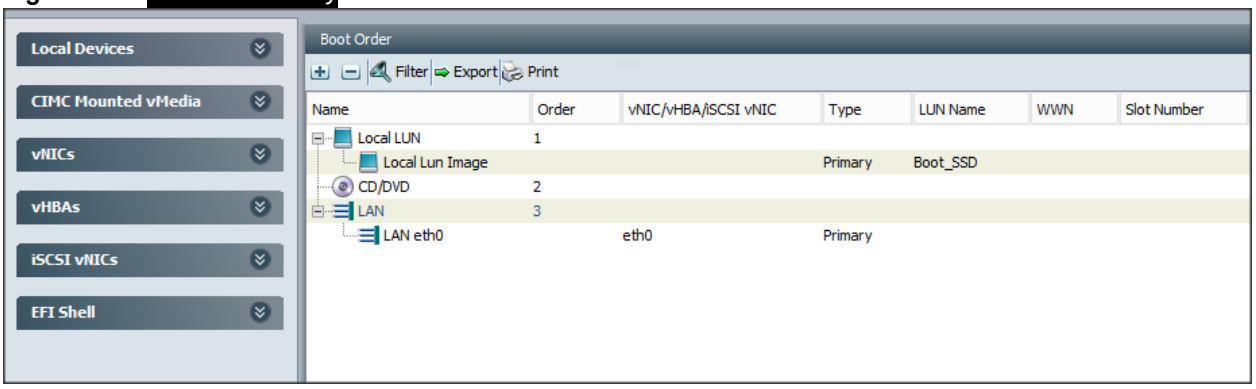


Figure 54 Add Boot Policy

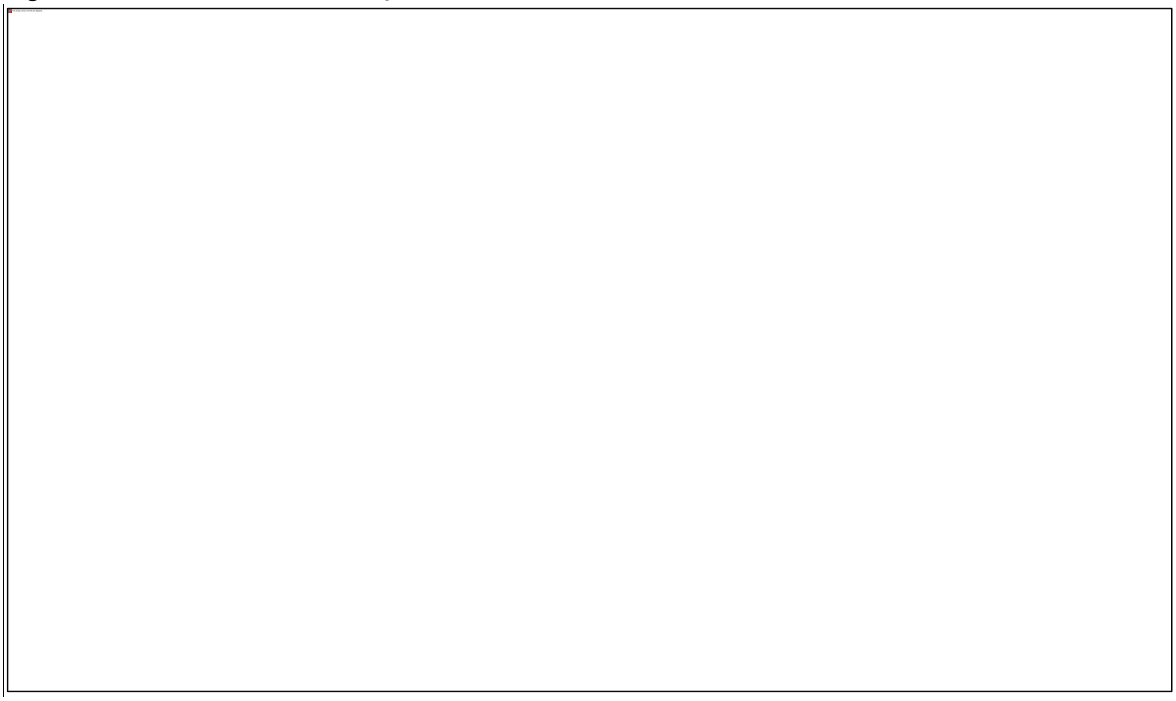


Creating Power Control Policy

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

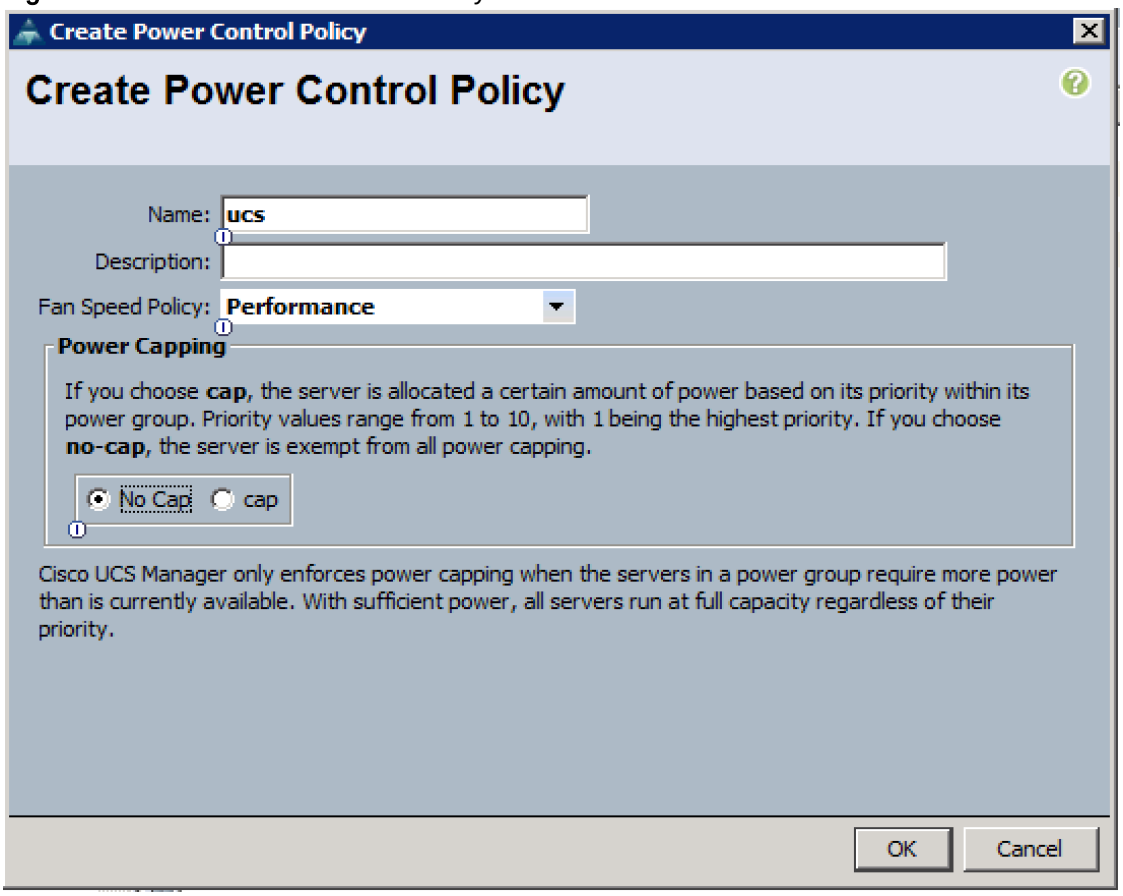
1. Select the `Servers` tab in the left pane in the Cisco UCS Manager GUI.
2. Select Policies > root.
3. Right-click the Power Control Policies (Figure 55).
4. Select Create Power Control Policy.

Figure 55 Power Control Policy



5. Enter `ucs` for the Power Control policy name (Figure 56).
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 56 Create Power Control Policy

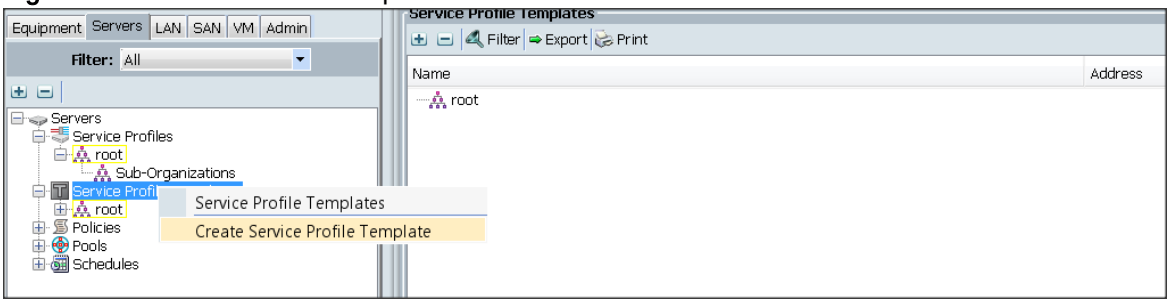


Creating a Service Profile Template

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

1. Select the `Servers` tab in the left pane in the Cisco UCSM GUI.
2. Right-click `Service Profile Templates` (Figure 57).
3. Select `Create Service Profile Template`.

Figure 57 Service Profile Template

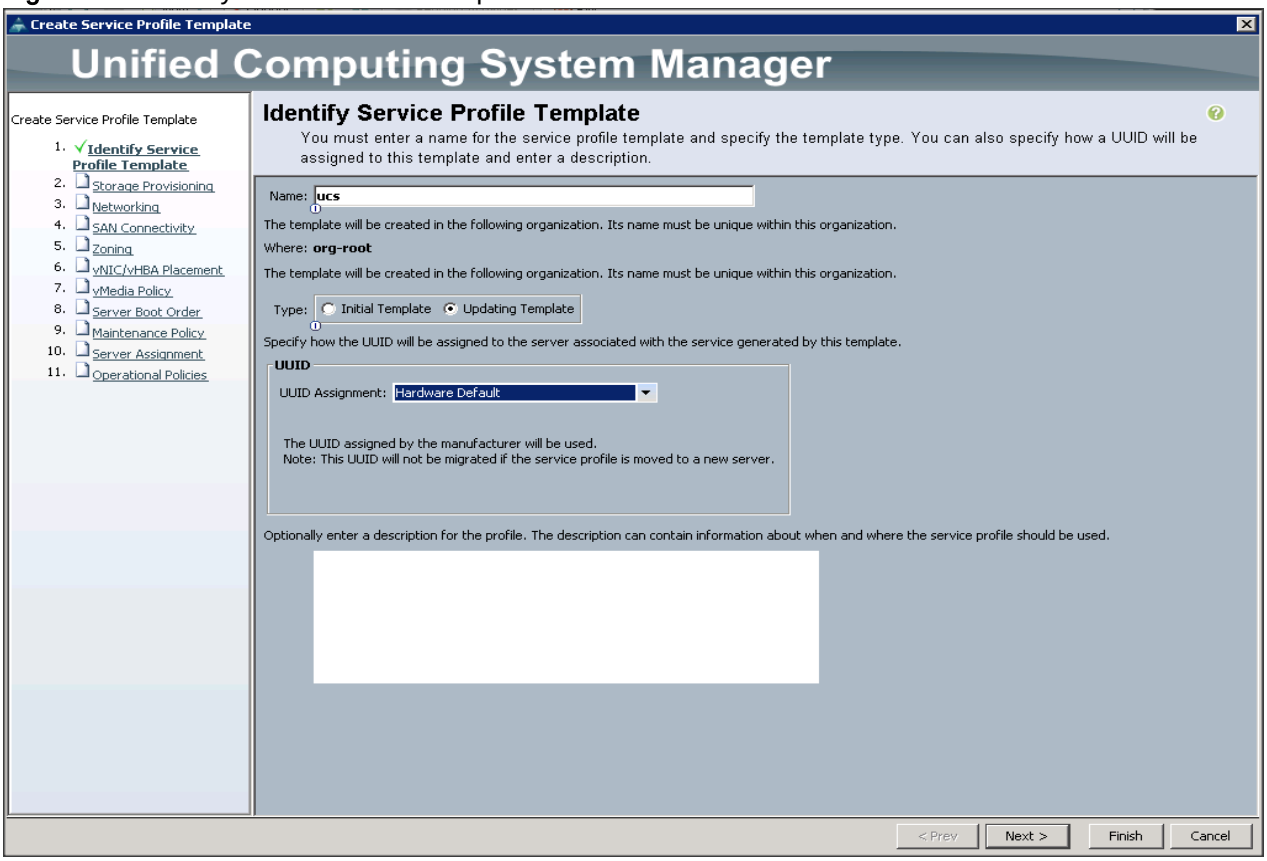


The Create Service Profile Template window appears.

To identify the service profile template, complete the following steps (Figure 58):

4. Name the service profile template `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 58 Identify Service Profile Template

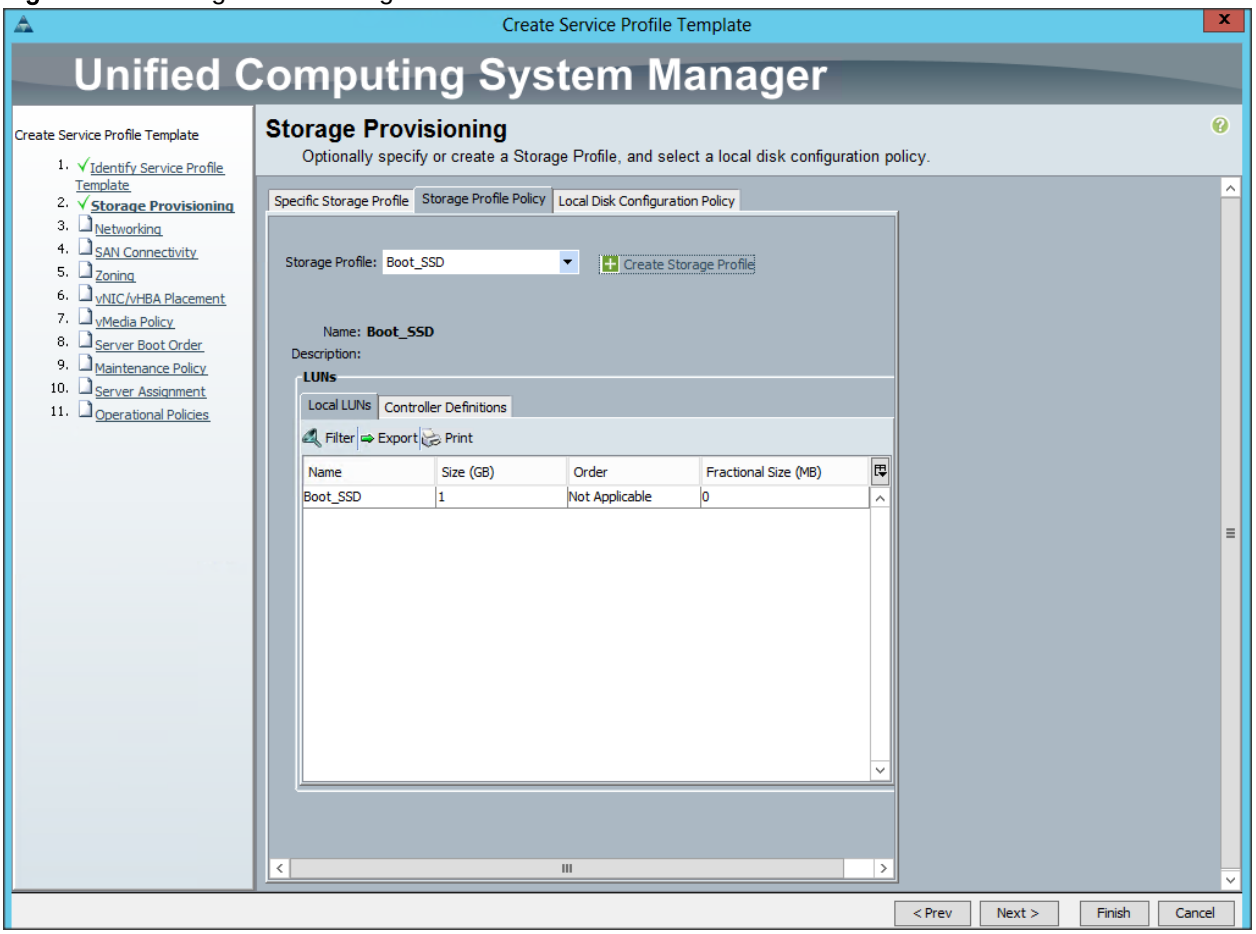


Configuring the Storage Provisioning for the Template

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 59)
2. Click `Next` to continue to the next section.

Figure 59 Storage Provisioning



4. Go to the Local Disk Configuration Policy tab, and select `ucs` for the Local Storage. Figure 60. Click `Next` to continue to the next section

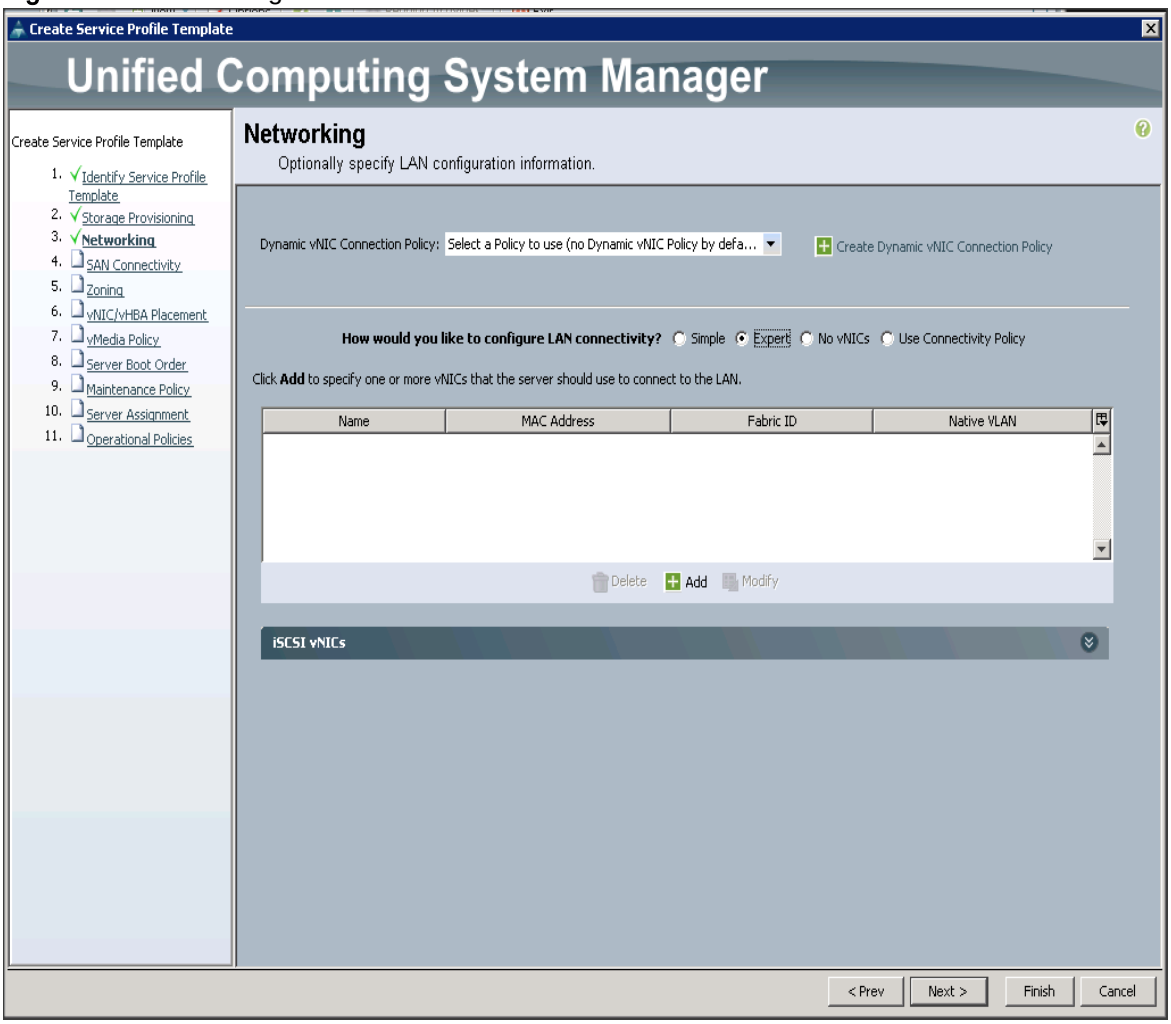
Figure 60 Storage Provisioning /Local Storage

5. 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 set at the default (Figure 61).
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.

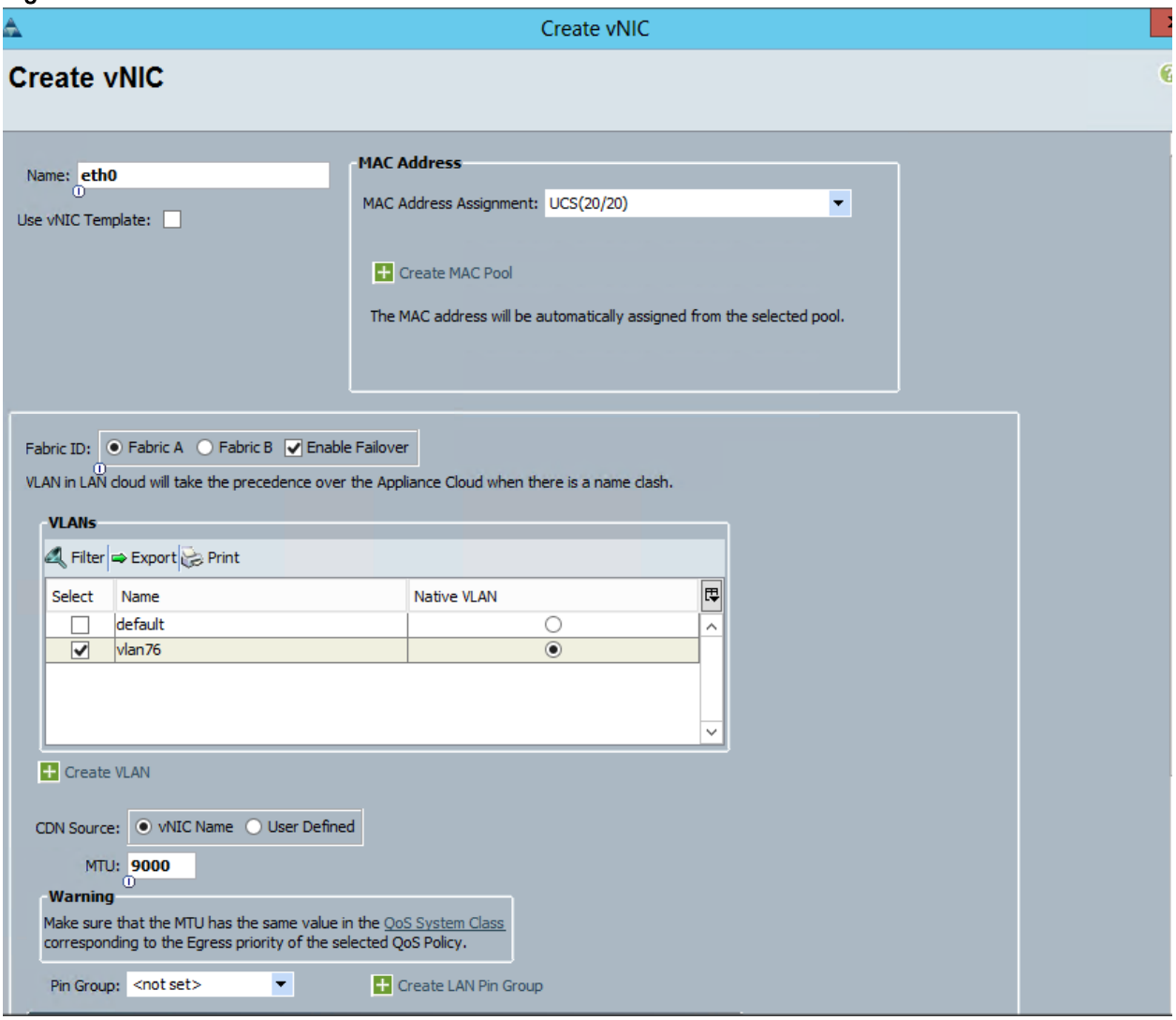
Figure 61 Networking



4. The Create vNIC window displays (Figure 62). Name the vNIC `eth0`.
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 `VLAN19` check box for VLANs and select the `Native VLAN radio button`.
8. Select MTU size 9000.
9. Select the adapter policy Linux.
10. Select the QoS Policy Platinum.
11. Keep the Network Control Policy set to Default.

12. Click OK.

Figure 62 Create vNIC



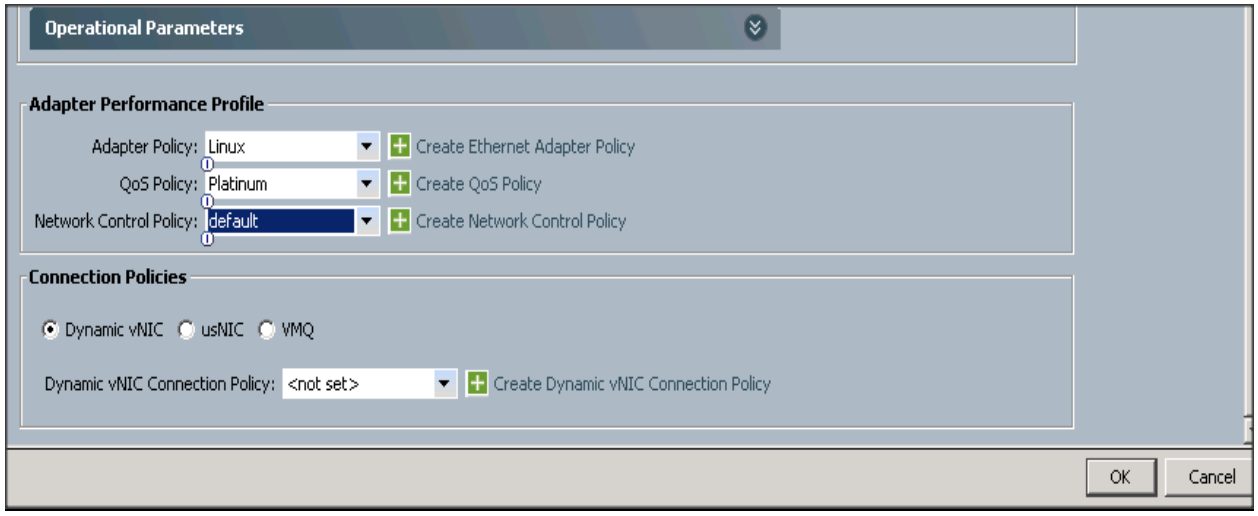
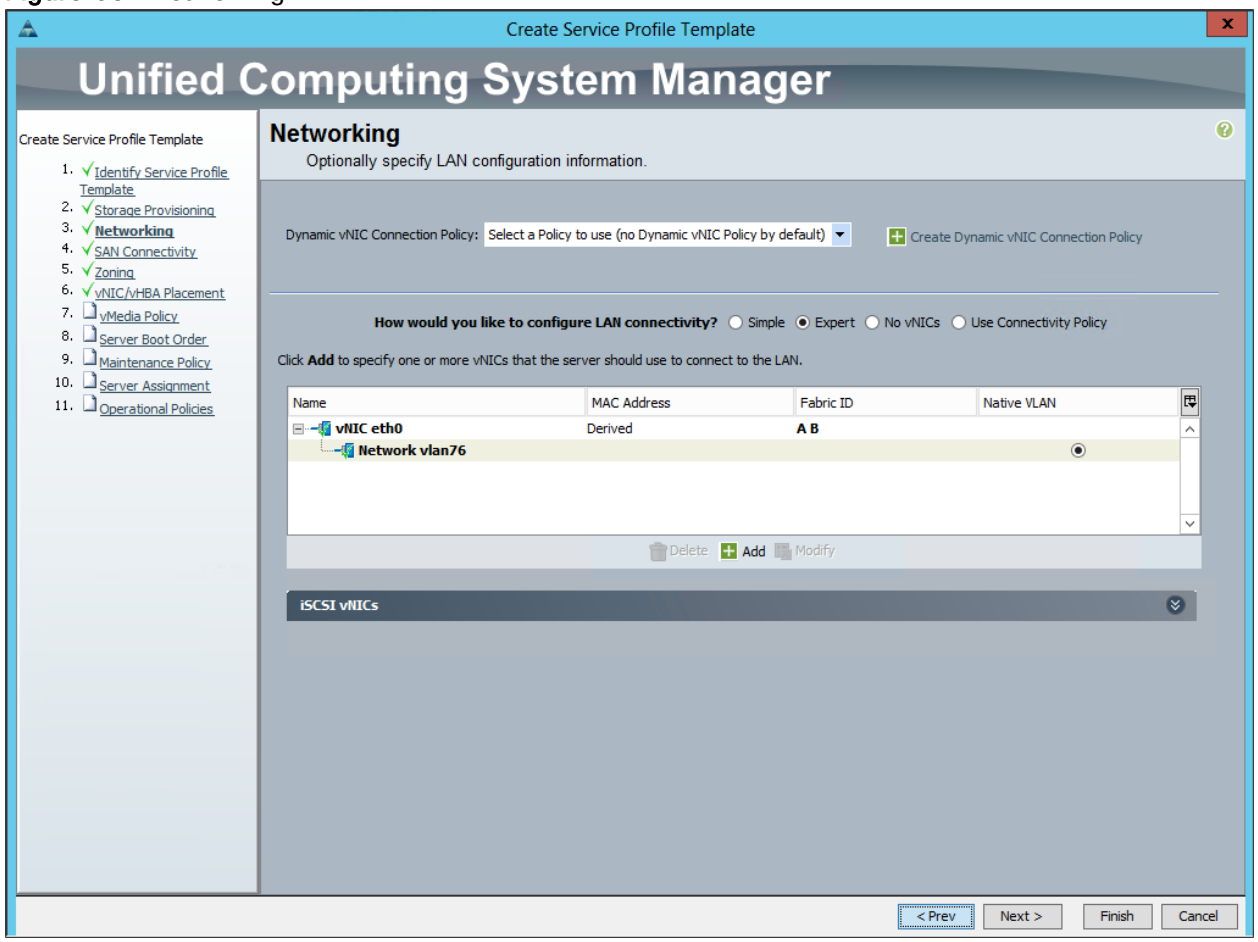


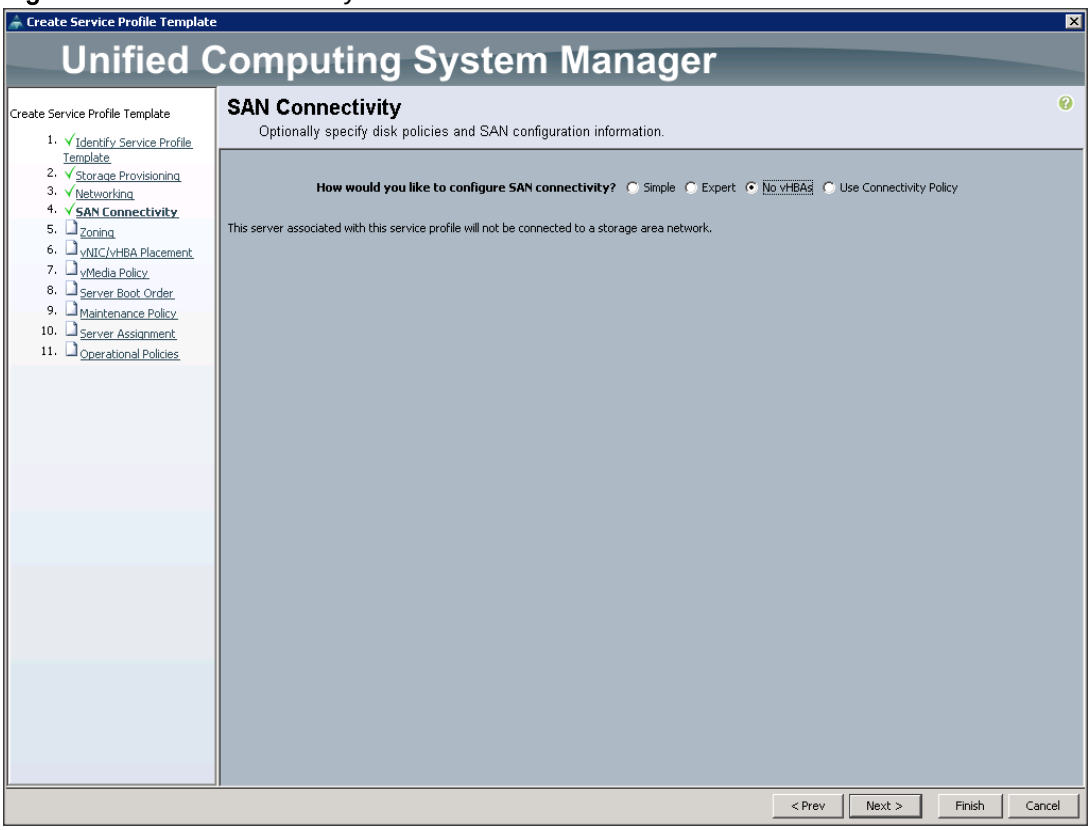
Figure 63 Networking LAN



13. Click `Next` to continue with SAN Connectivity (Figure 63).

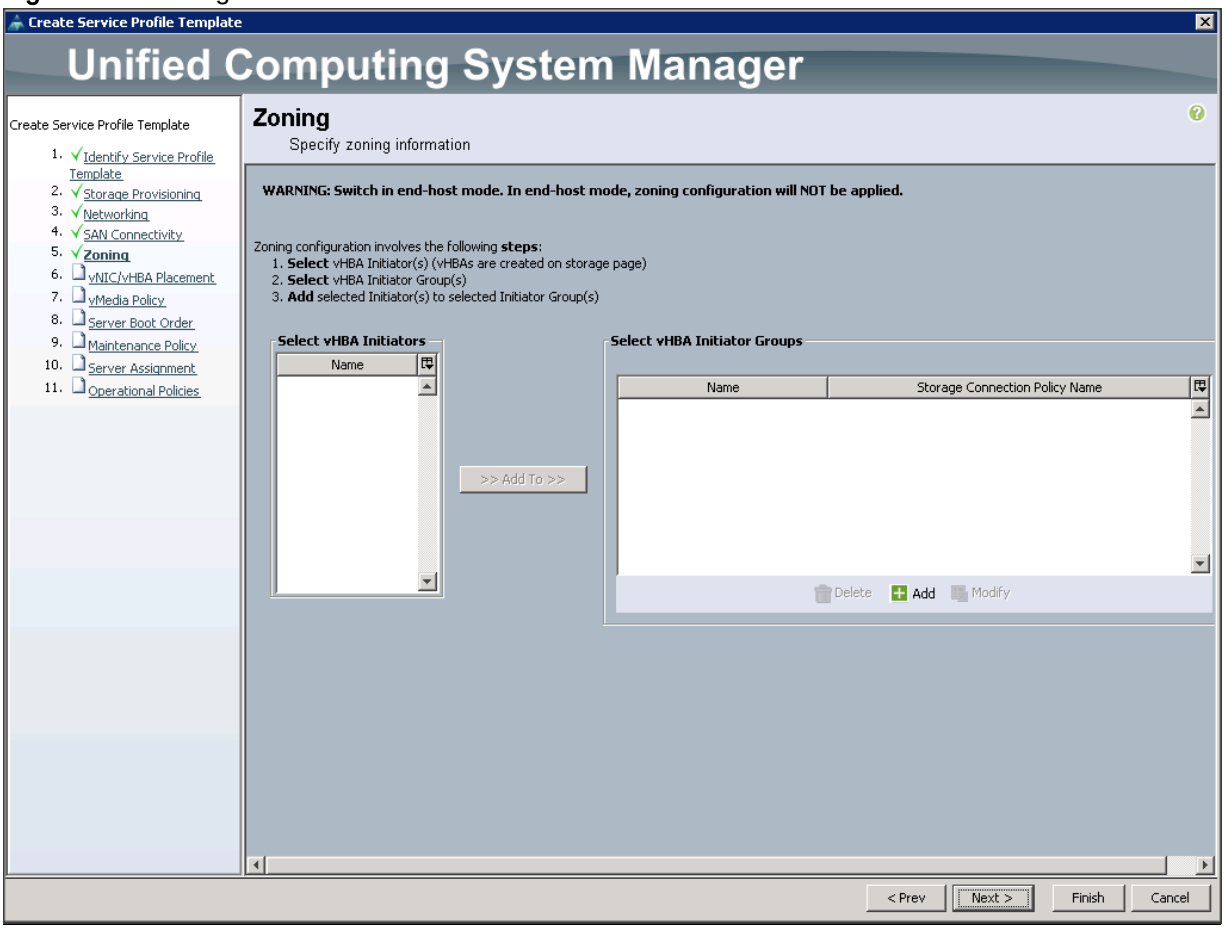
14. Select no vHBAs for How would you like to configure SAN Connectivity? (Figure 64)

Figure 64 SAN Connectivity



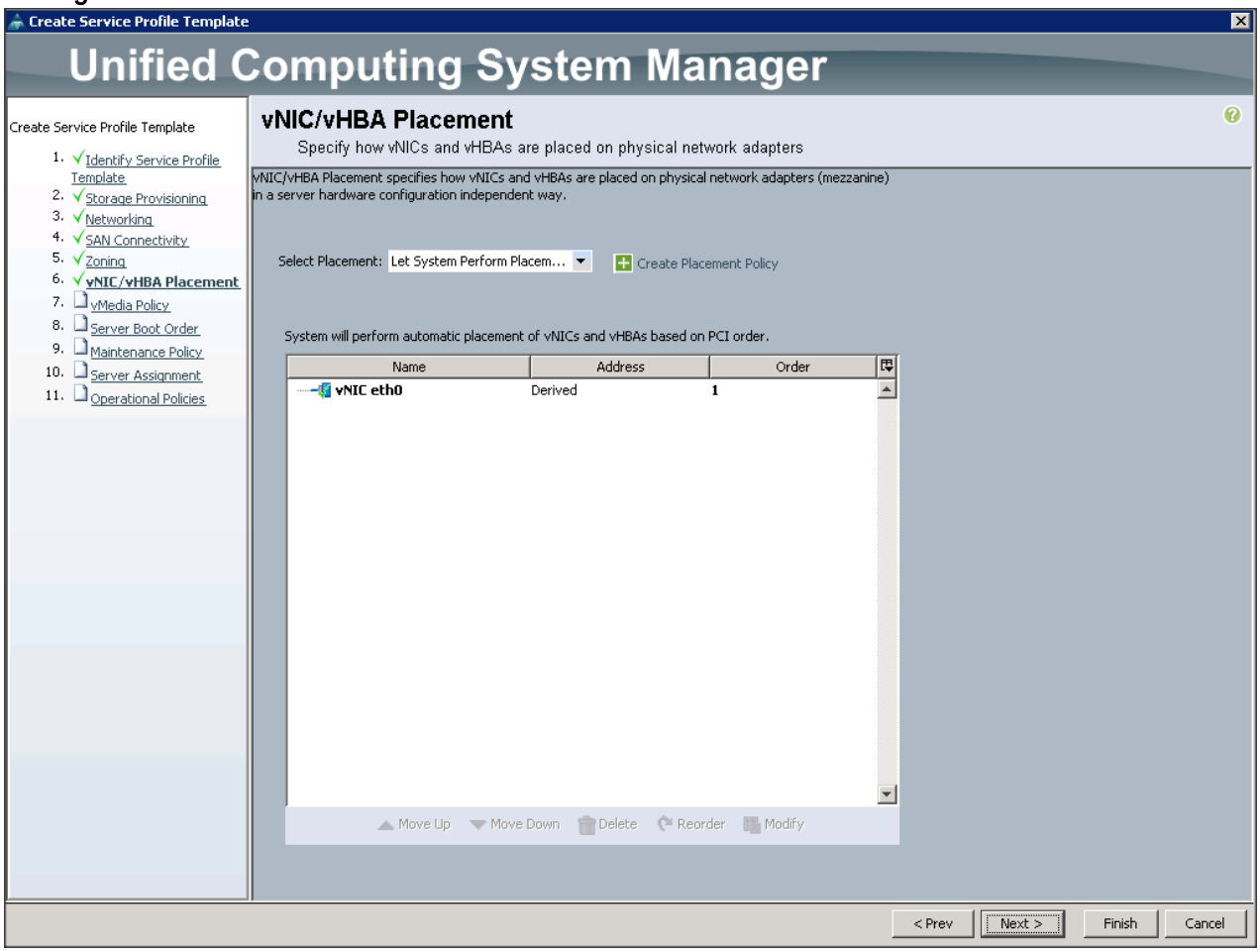
15. Click `Next` to continue with Zoning (Figure 65).

Figure 65 Zoning



16. Click `Next` to continue with vNIC/vHBA placement. (Figure 66)

Figure 66 vNIC/vHBA Placement

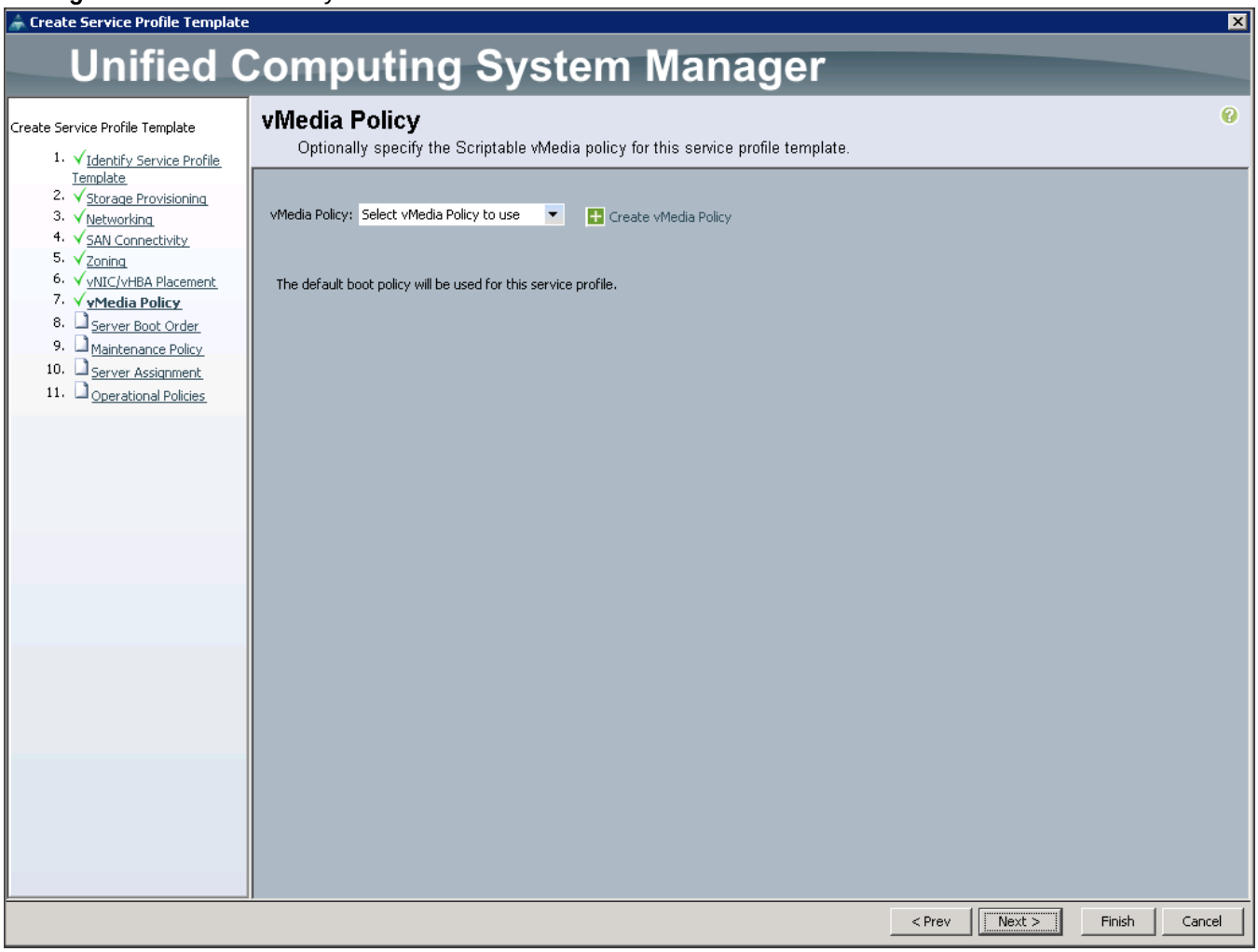


17. Click `Next` to configure vMedia Policy.

Configuring the vMedia Policy for the Template

1. Click `Next` once the vMedia Policy window appears to go to the next section.(Figure 67)

Figure 67 vMedia Policy

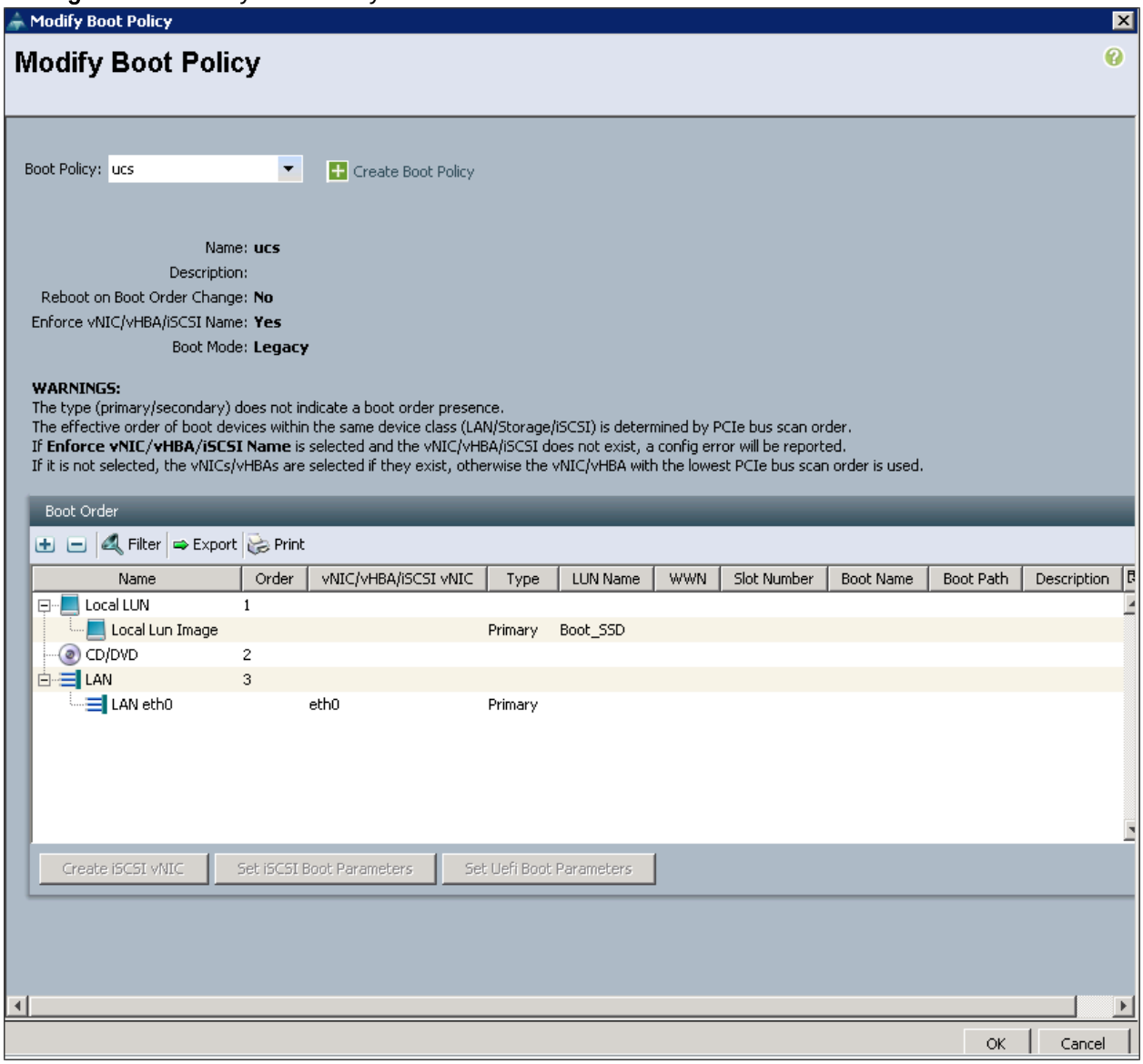


Configuring Server Boot Order for the Template

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

1. Select `ucs` in the Boot Policy name field. (Figure 68)
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.

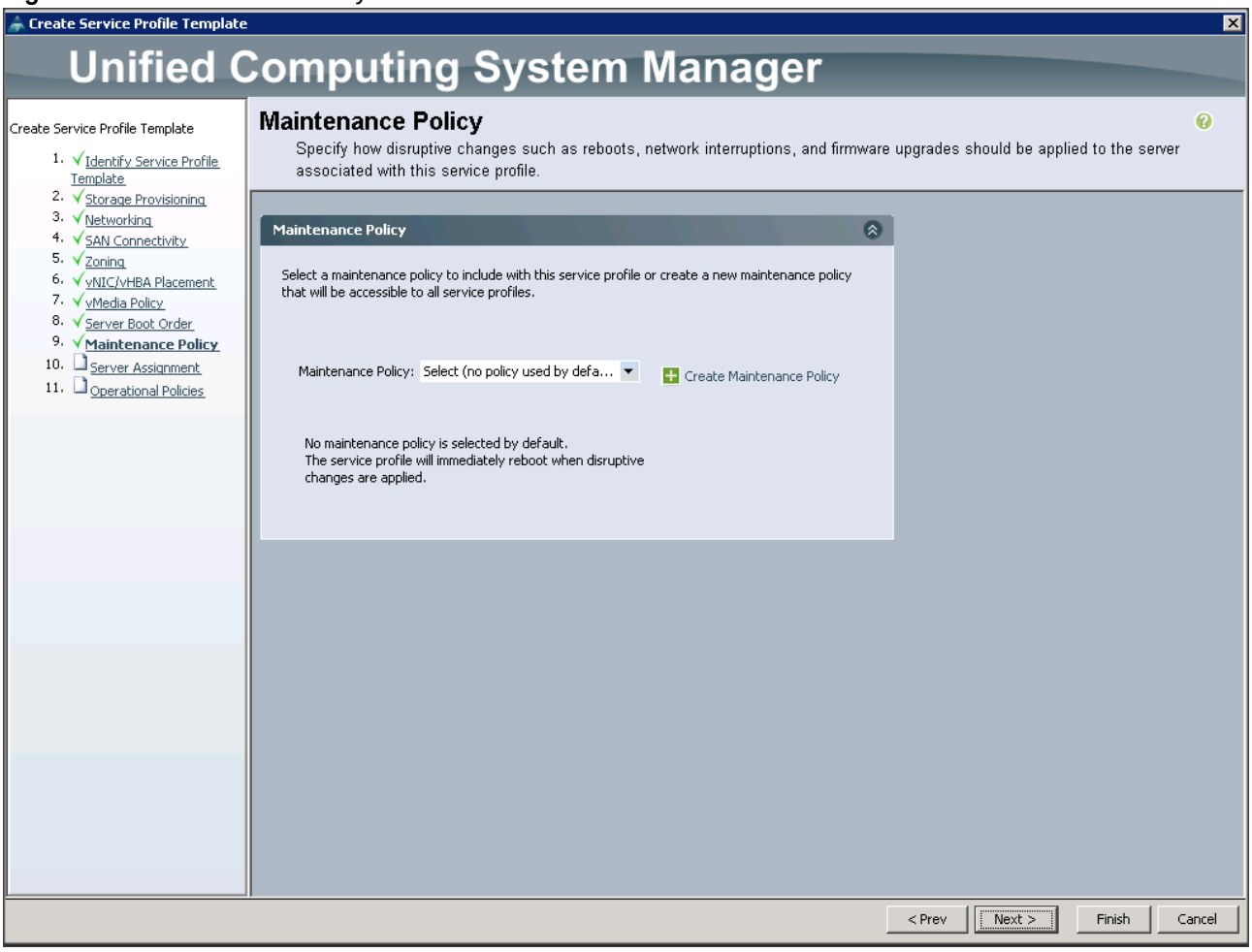
Figure 68 Modify Boot Policy



6. In the Maintenance Policy window, apply the maintenance policy (Figure 69).

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

Figure 69 Maintenance Policy

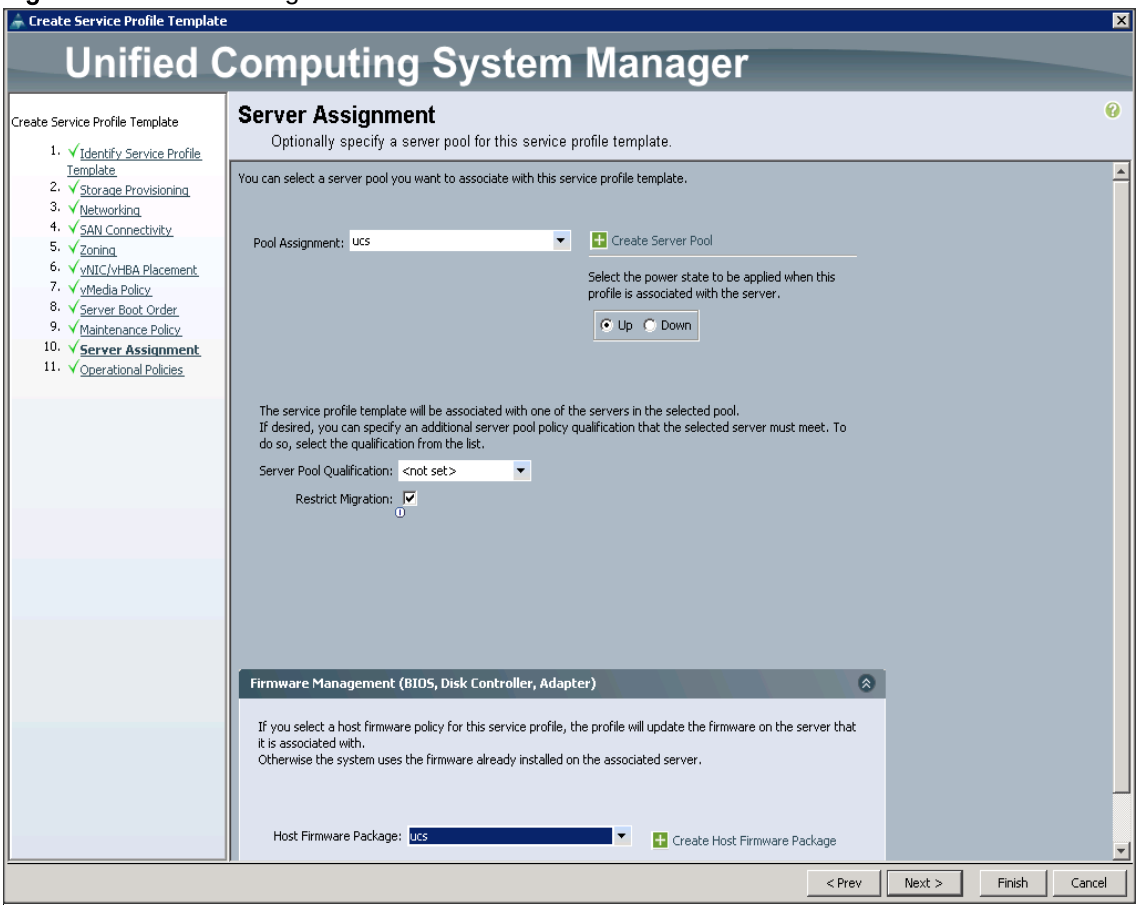


Configuring Server Assignment for the Template

In the Server Assignment window (Figure 70), to assign the servers to the pool, complete the following steps:

1. Select `ucs` for the Pool Assignment field.
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 70 Server Assignment

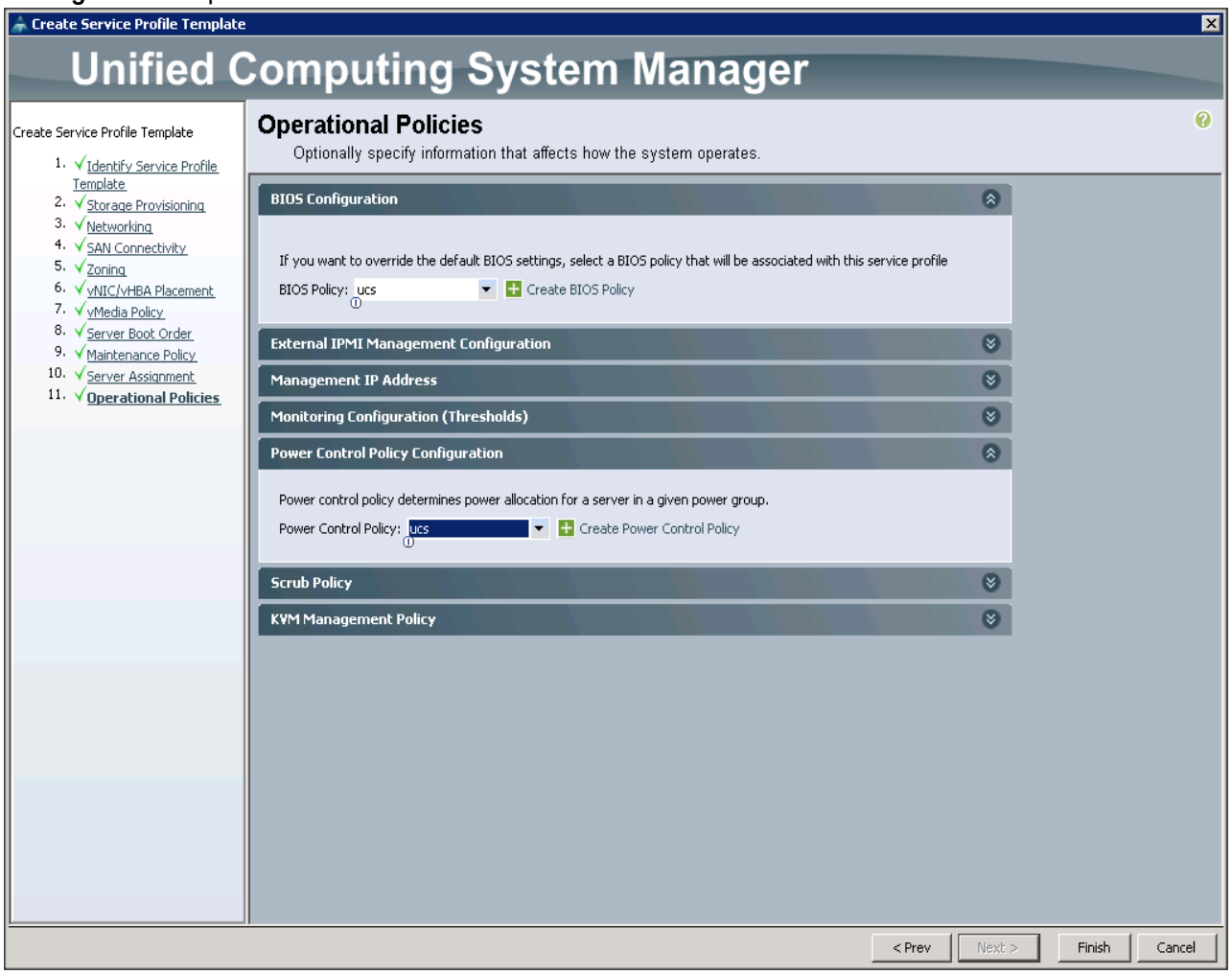


Configuring Operational Policies for the Template

In the Operational Policies Window (Figure 71), complete the following steps:

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

Figure 71 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 Cisco UCS Manager GUI.
6. Go to `Service Profile Templates > root.`
7. Right-click `Service Profile Templates ucs.`
8. Select `Create Service Profiles From Template`. (Figure 72)

Figure 72 Create Service Profiles From Template

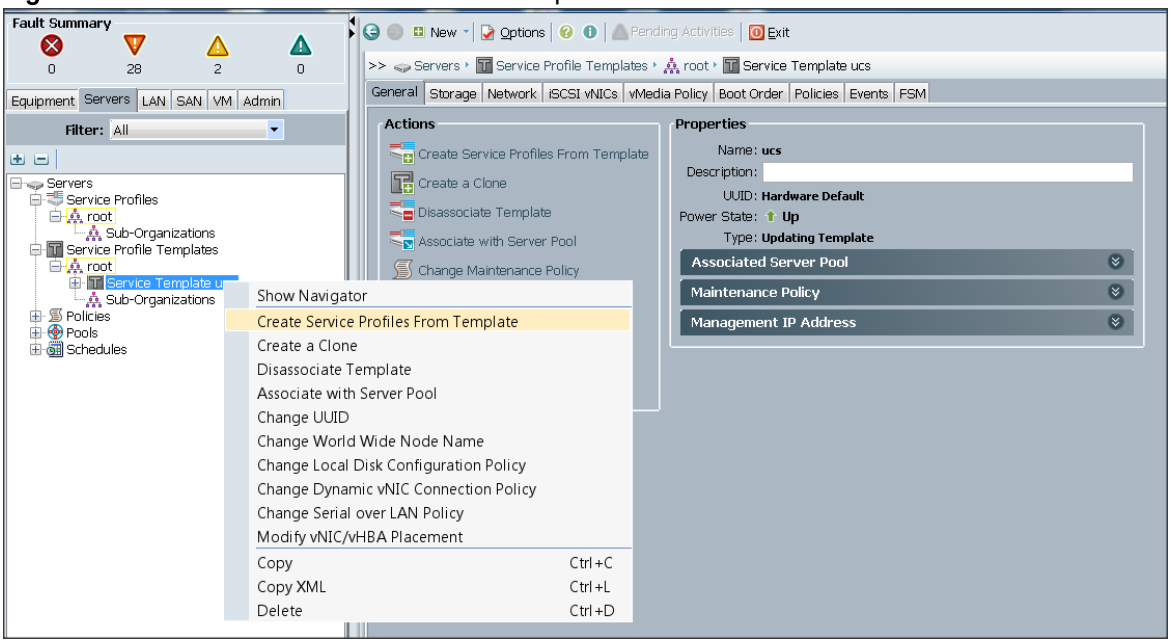
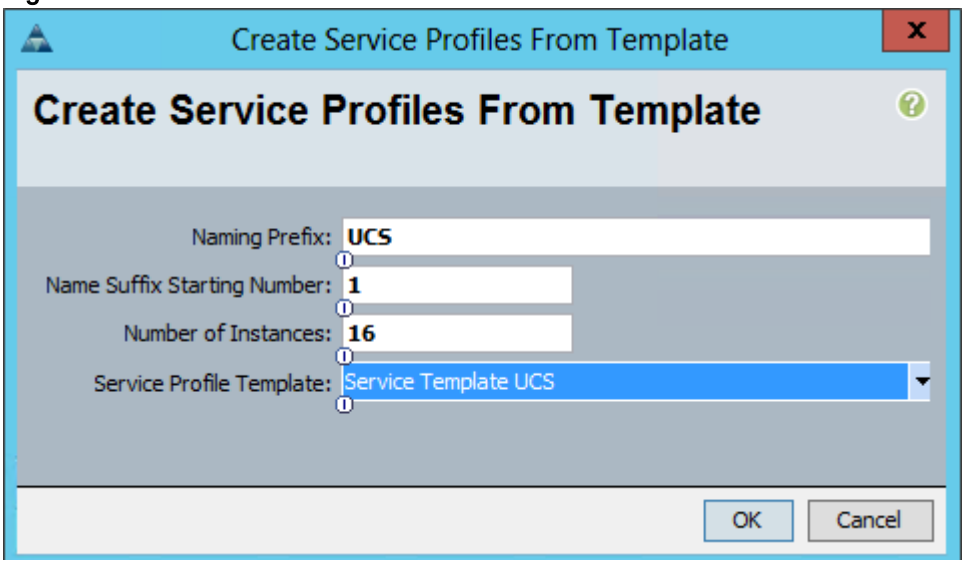


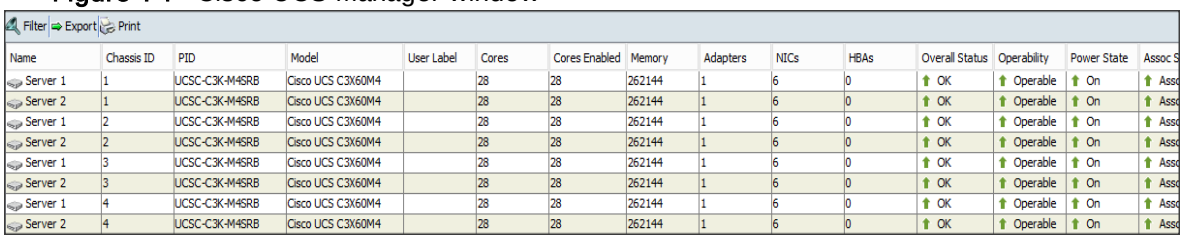
Figure 73 Service Profile Name



Association of the Service Profiles will take place automatically. Click OK. (Figure 73)

The final Cisco UCS Manager window is shown below in Figure 74.

Figure 74 Cisco UCS Manager window



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.

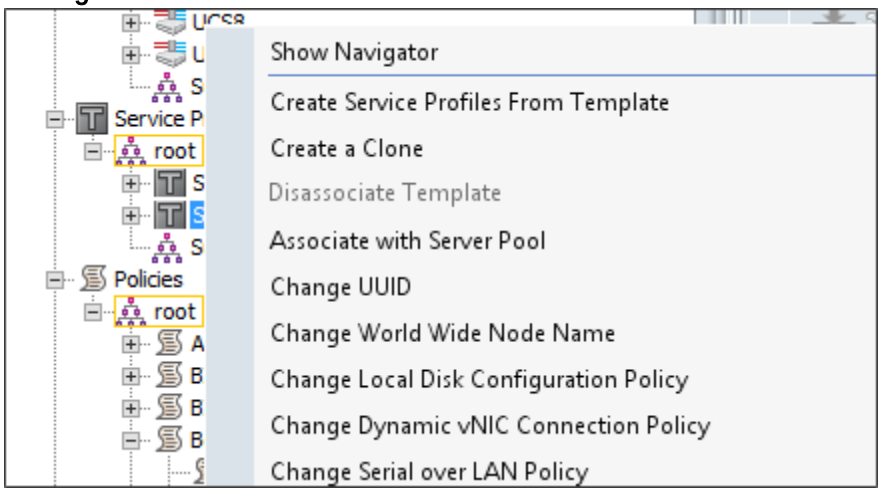
To configure an organization within the Cisco UCS Manager GUI, complete the following steps:

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

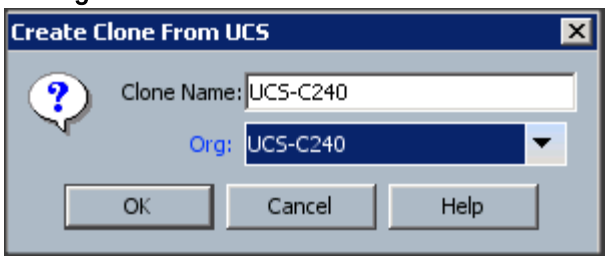
1. Click on Servers tab, go to Service Profile Template → root.
2. Right click on the existing template UCS and click Create a Clone. (Figure 75)

Figure 75 Create a Clone



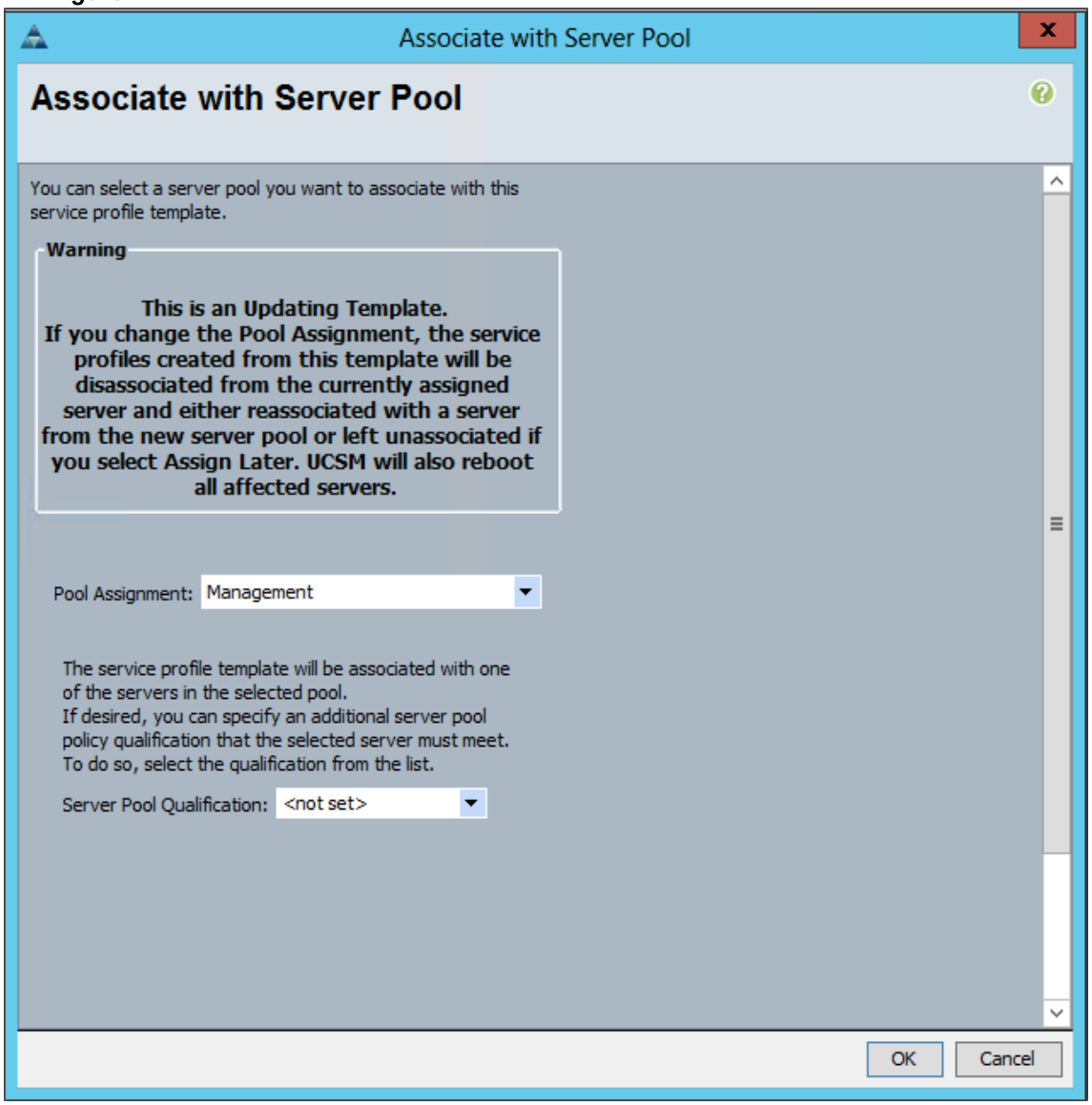
3. In the Clone Name, enter UCS-C240 and from the Org drop down list choose UCS-C240 and click OK. (Figure 76)

Figure 76 Create a Clone



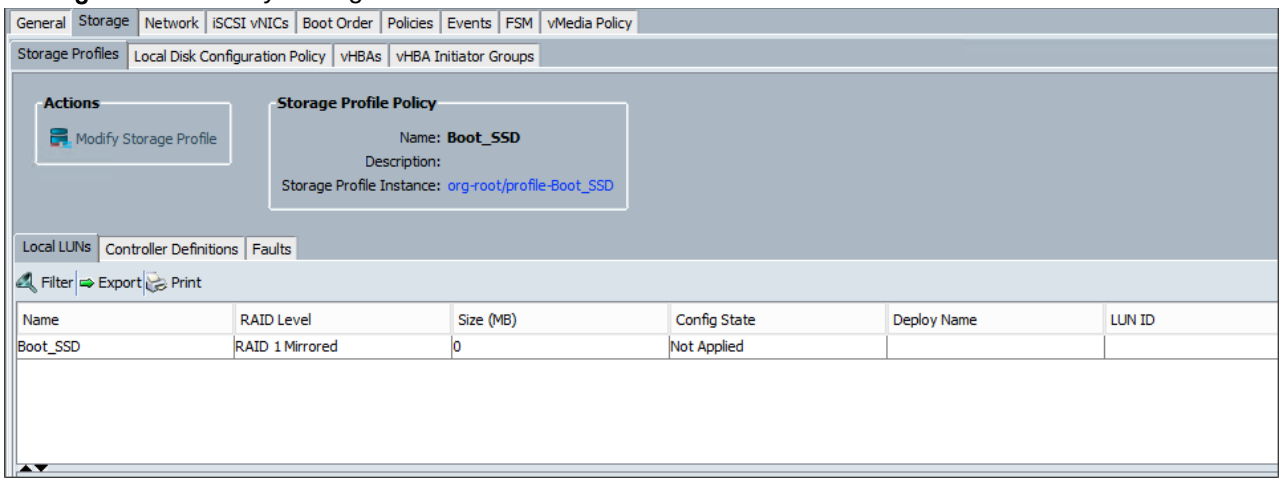
4. Go to root → Sub-Organization → UCS-C240 and select the Service Template UCS-C240.
5. In the right window general tab click Associate with Server pool. (Figure 77)
6. In the Pool Assignment drop down list choose Management and click OK.

Figure 77 Associate with Server Pool



7. In the right window select the Storage tab and click Modify Storage Profile. (Figure 78)

Figure 78 Modify Storage Profile



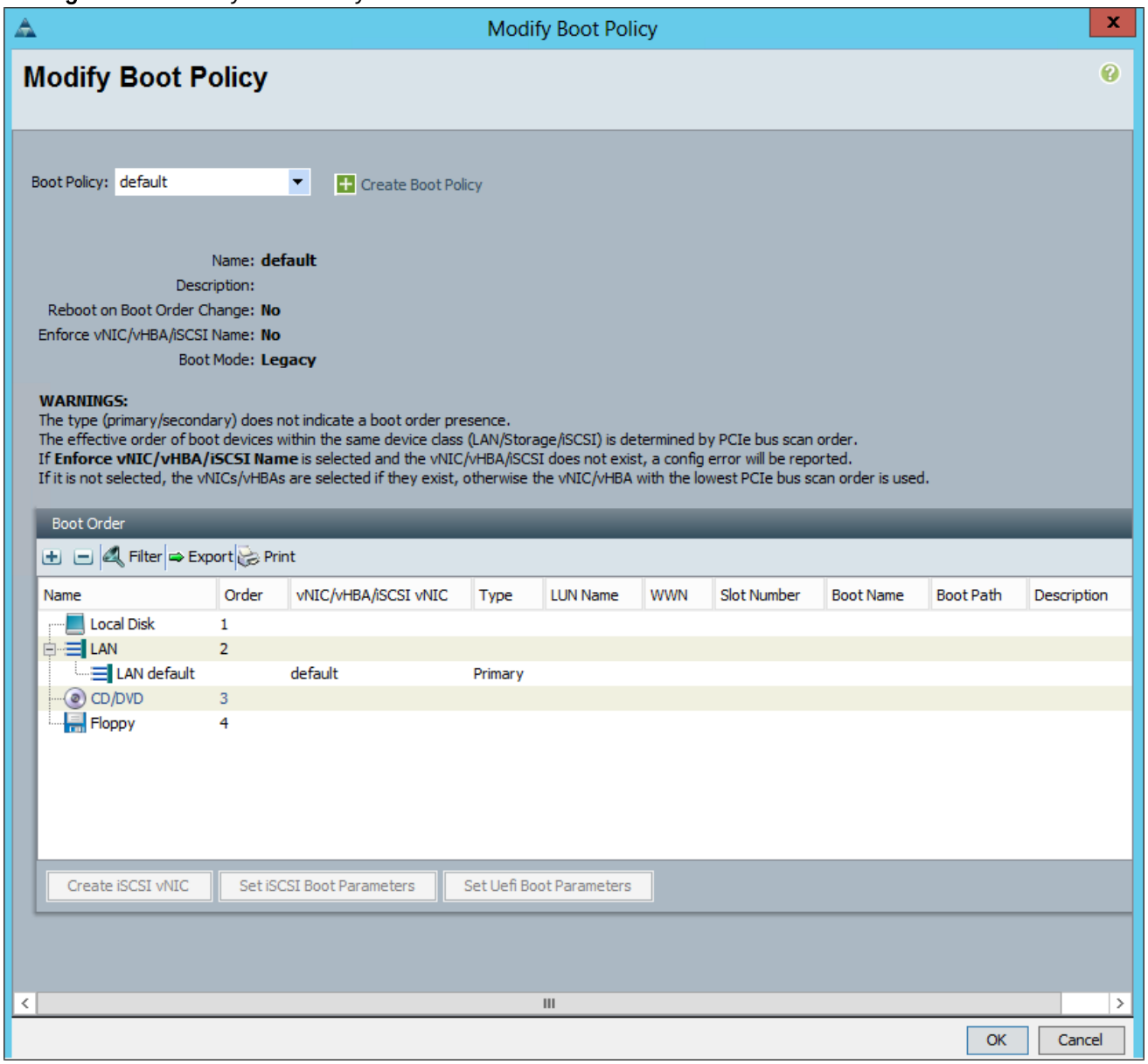
8. From the Storage profile drop down list choose No Storage Profile and click OK. (Figure 79)

Figure 79 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 80)

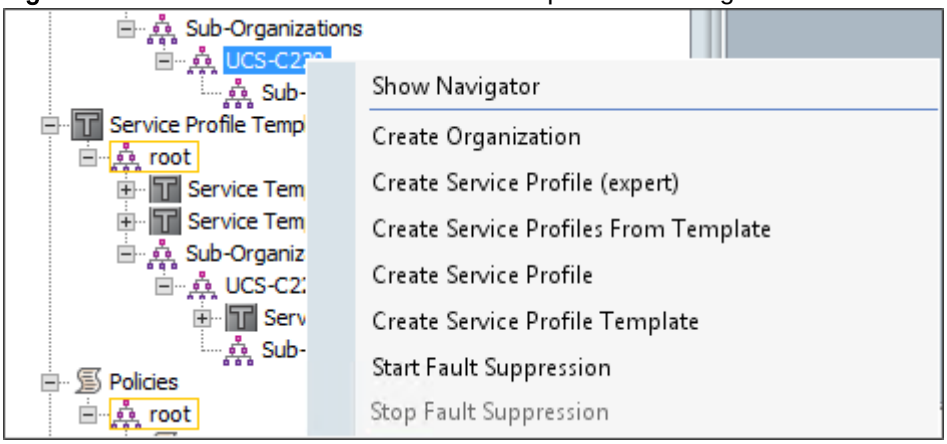
Figure 80 Modify Boot Policy



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 81)

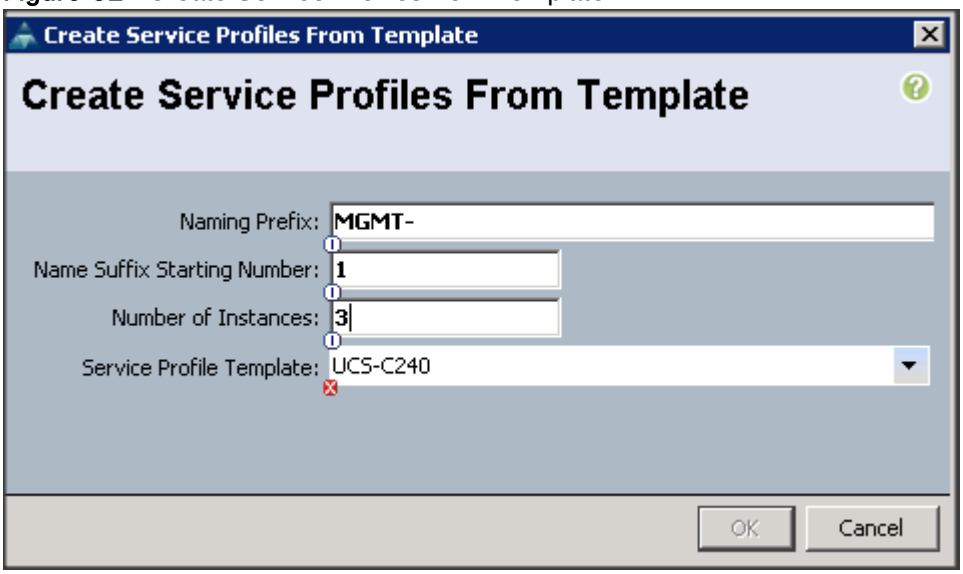
Figure 81 Create Service Profiles from Template for Management nodes



In the Create Service Profiles from Template screen: (Figure 82)

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 82 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

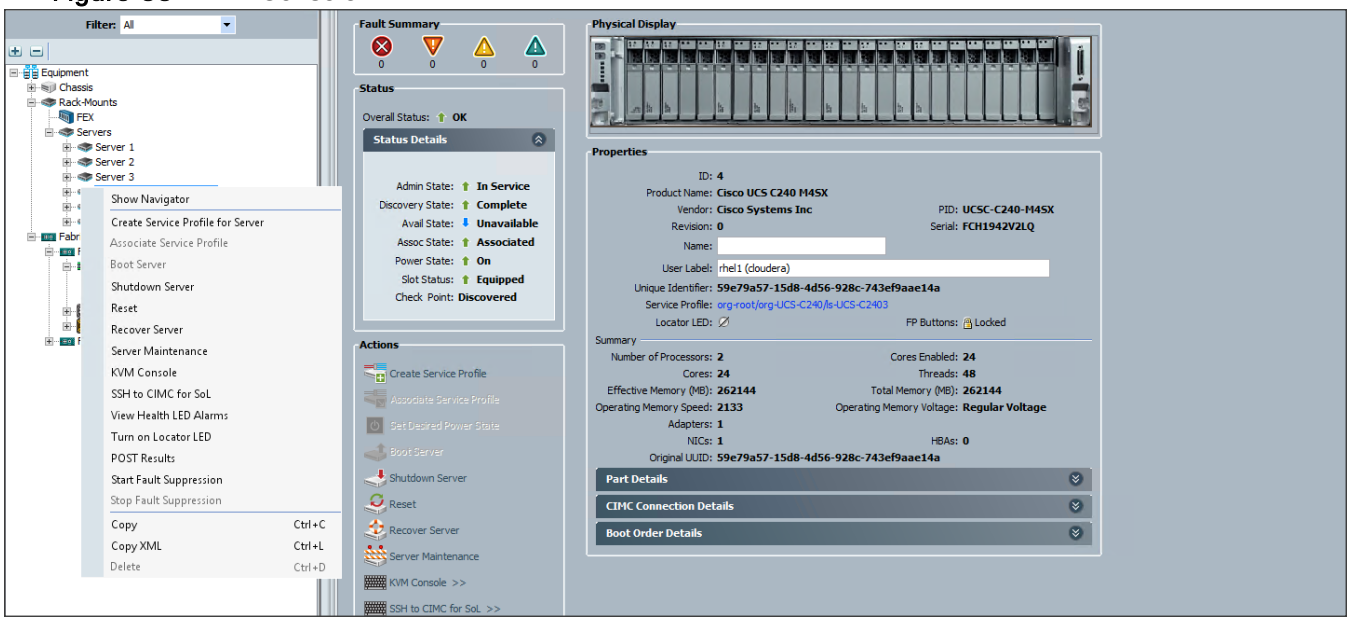
The following section provides detailed procedures for installing Red Hat Enterprise Linux 7.2 using Software RAID (OS based Mirroring) on Cisco UCS C240 M4 Rack Servers. There are multiple ways to install the Red Hat Linux operating system. The installation procedure described in this deployment guide 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 83.
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 83 KVM Console

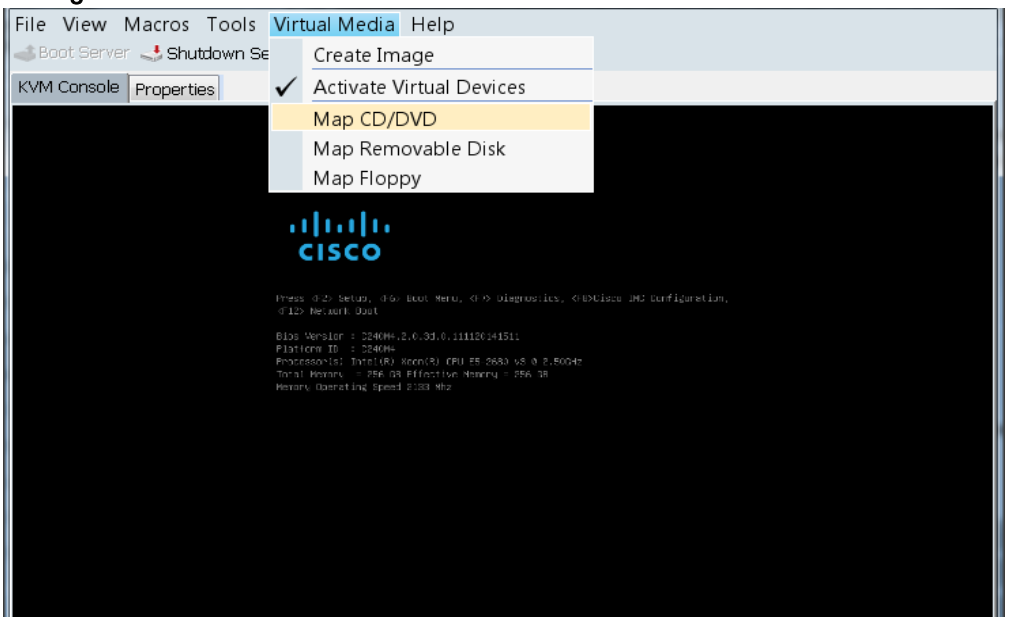


6. Click the Activate Virtual Devices found in the Virtual Media tab. (Figure 84)

Figure 84 Virtual Media Tab

7. In the KVM window (Figure 85), select the Virtual Media tab and click the Map CD/DVD.

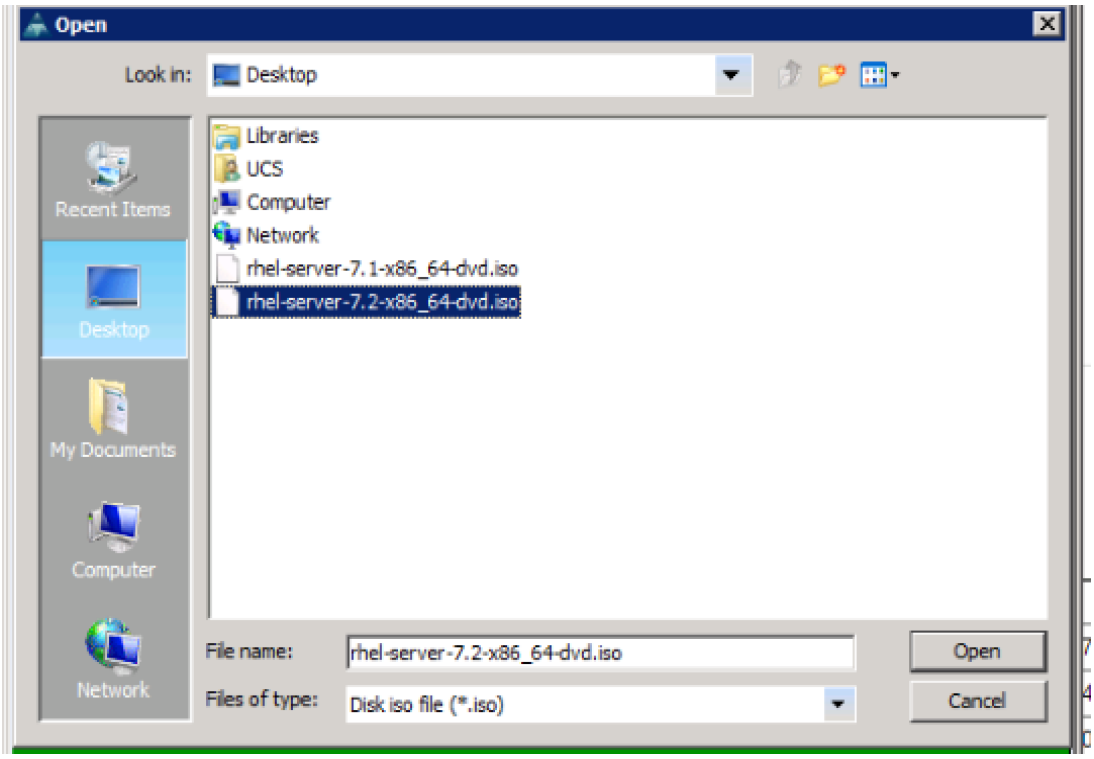
Figure 85 KVM Console Window



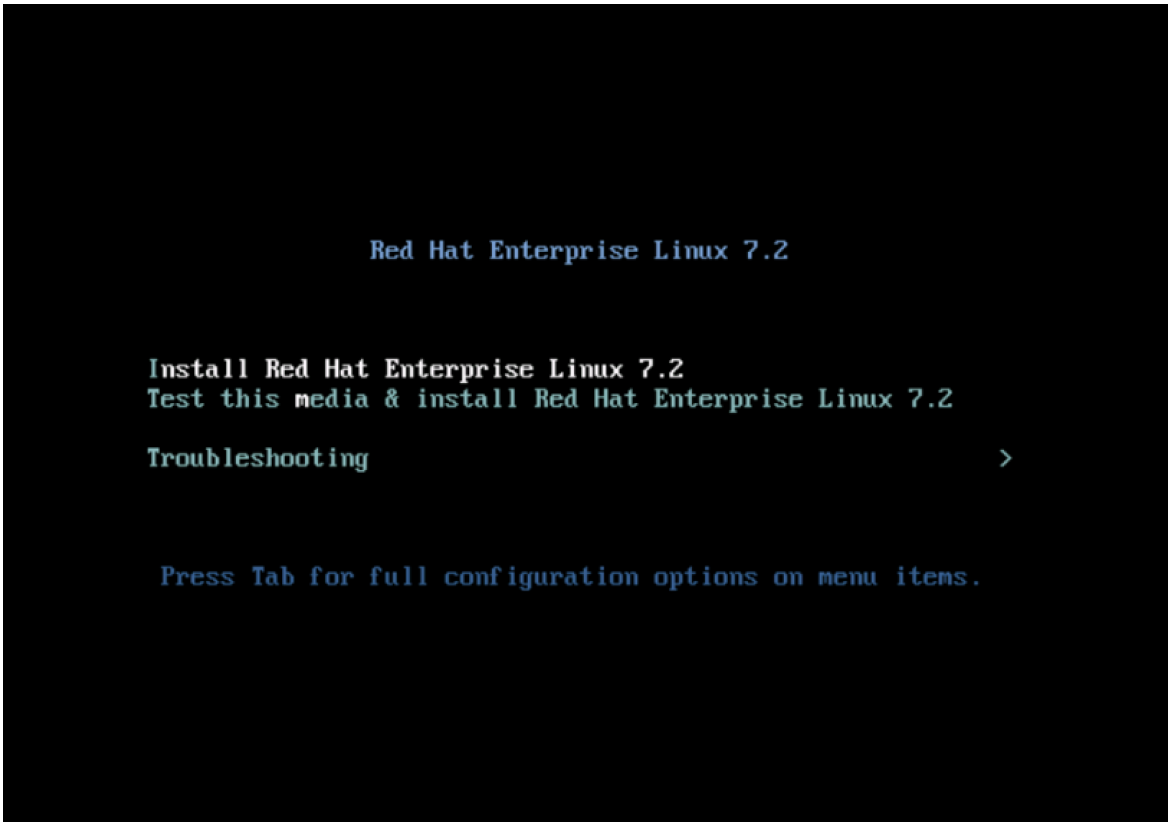
8. Browse to the Red Hat Enterprise Linux Server 7.2 installer ISO image file.

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.



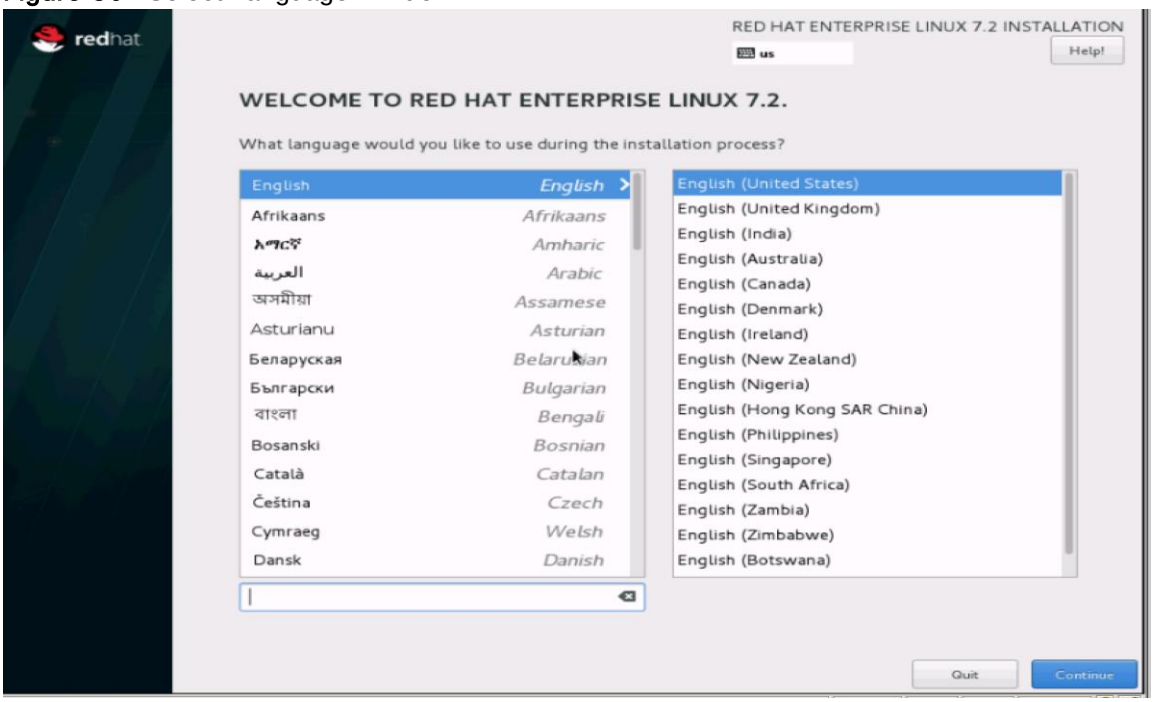
10. In the KVM window, select the KVM tab to monitor during boot.
11. In the KVM window, select the Macros > Static Macros > Ctrl-Alt-Del button in the upper left corner.
12. Click OK.
13. Click OK to reboot the system.
14. On reboot, the machine detects the presence of the Red Hat Enterprise Linux Server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



16. Skip the Media test and start the installation.

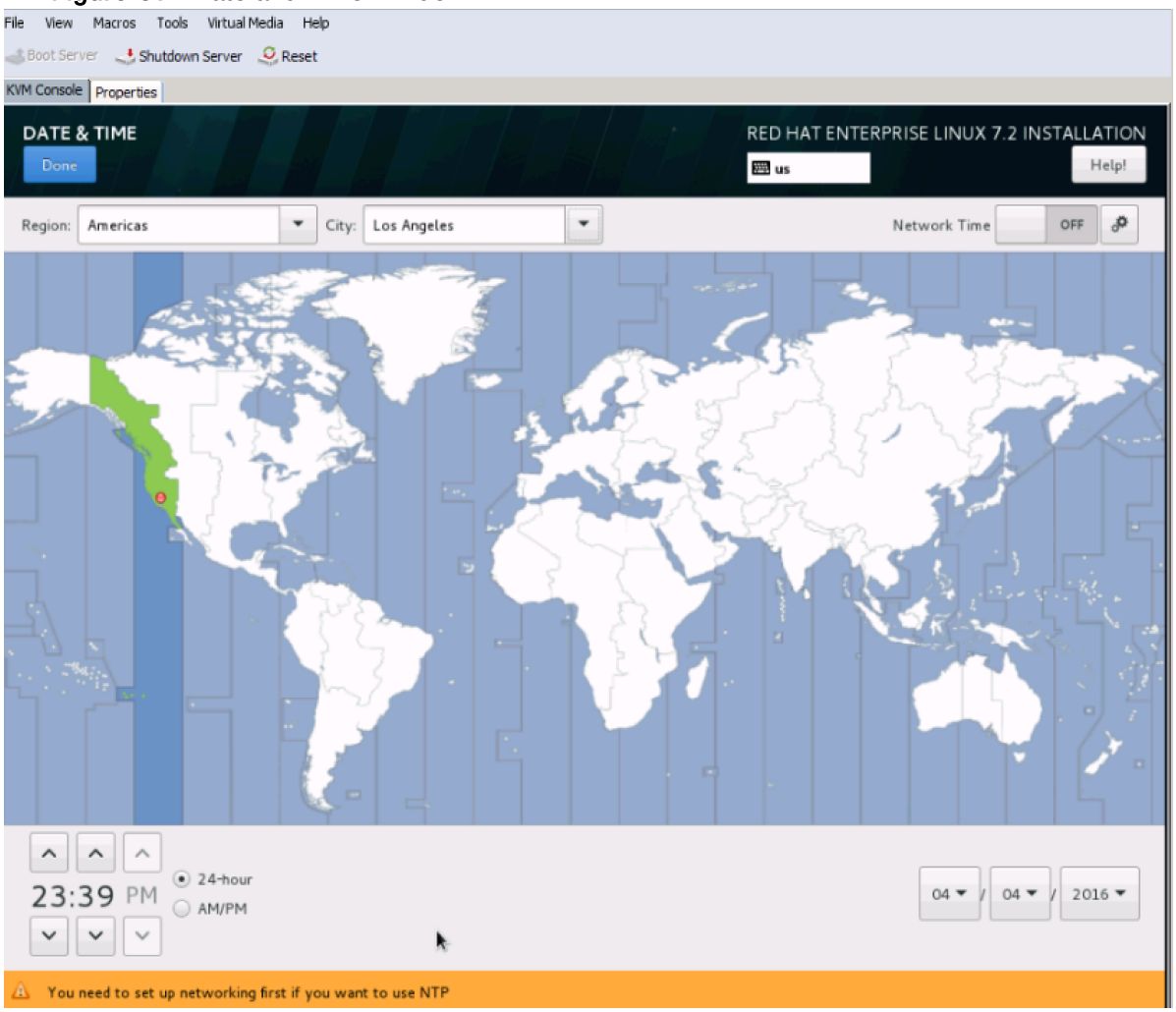
17. Select language of installation (Figure 86), and click Continue.

Figure 86 Select Language Window



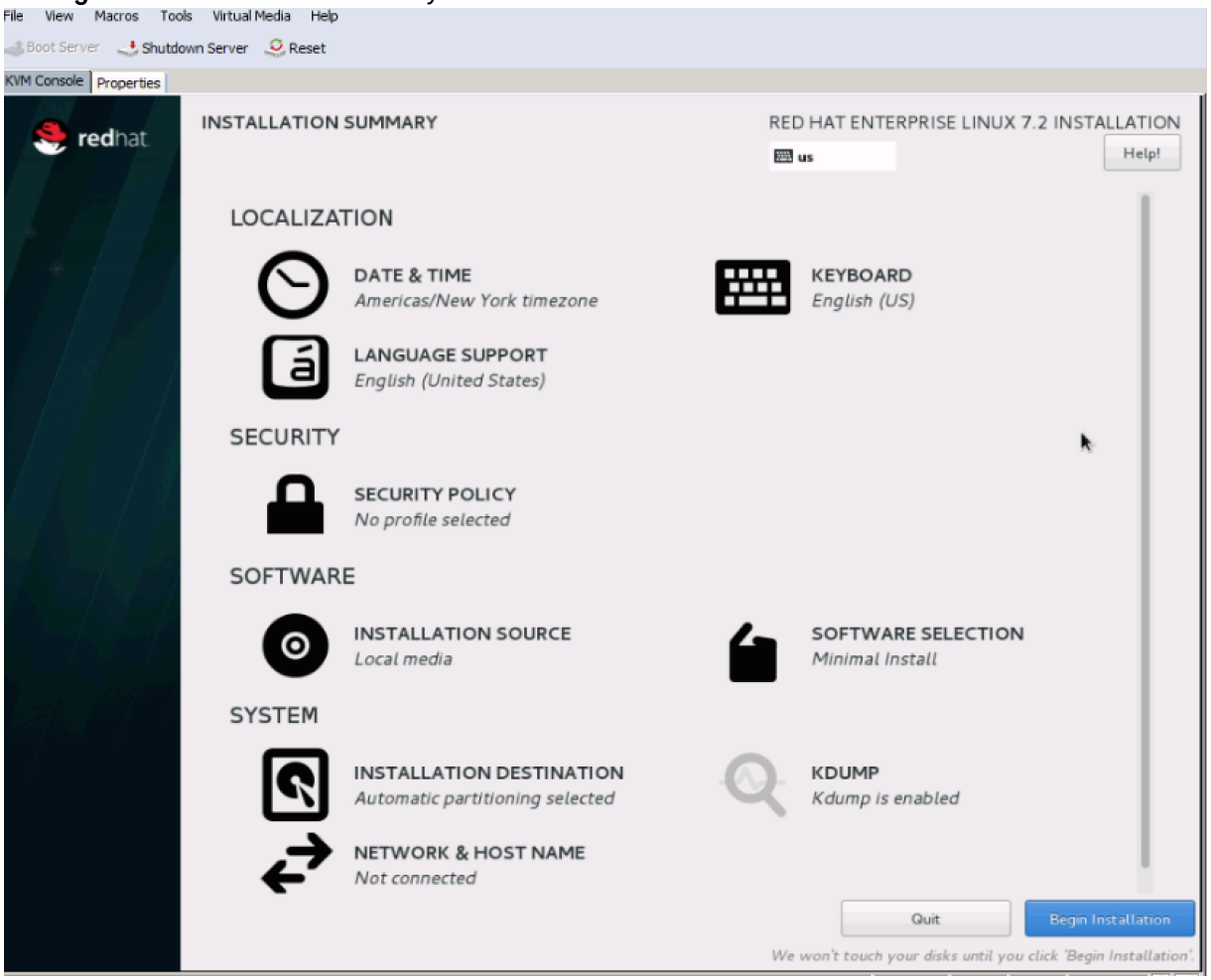
18. Select Date and time as shown in Figure 87.

Figure 87 Date and Time Window



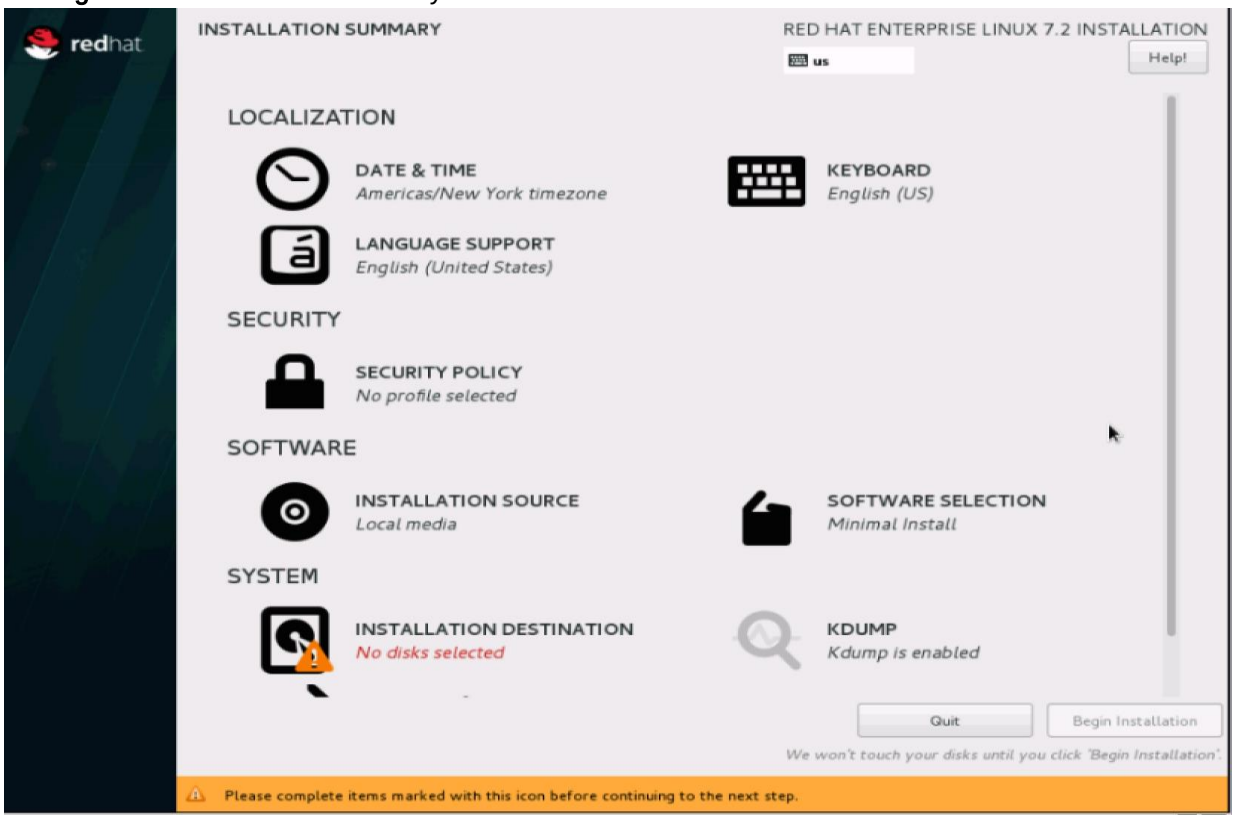
19. Select the location on the map, set the time and click Done.

Figure 88 Installation Summary Window



20. Click on Installation Destination, shown above in Figure 88.

Figure 89 Installation Summary Window

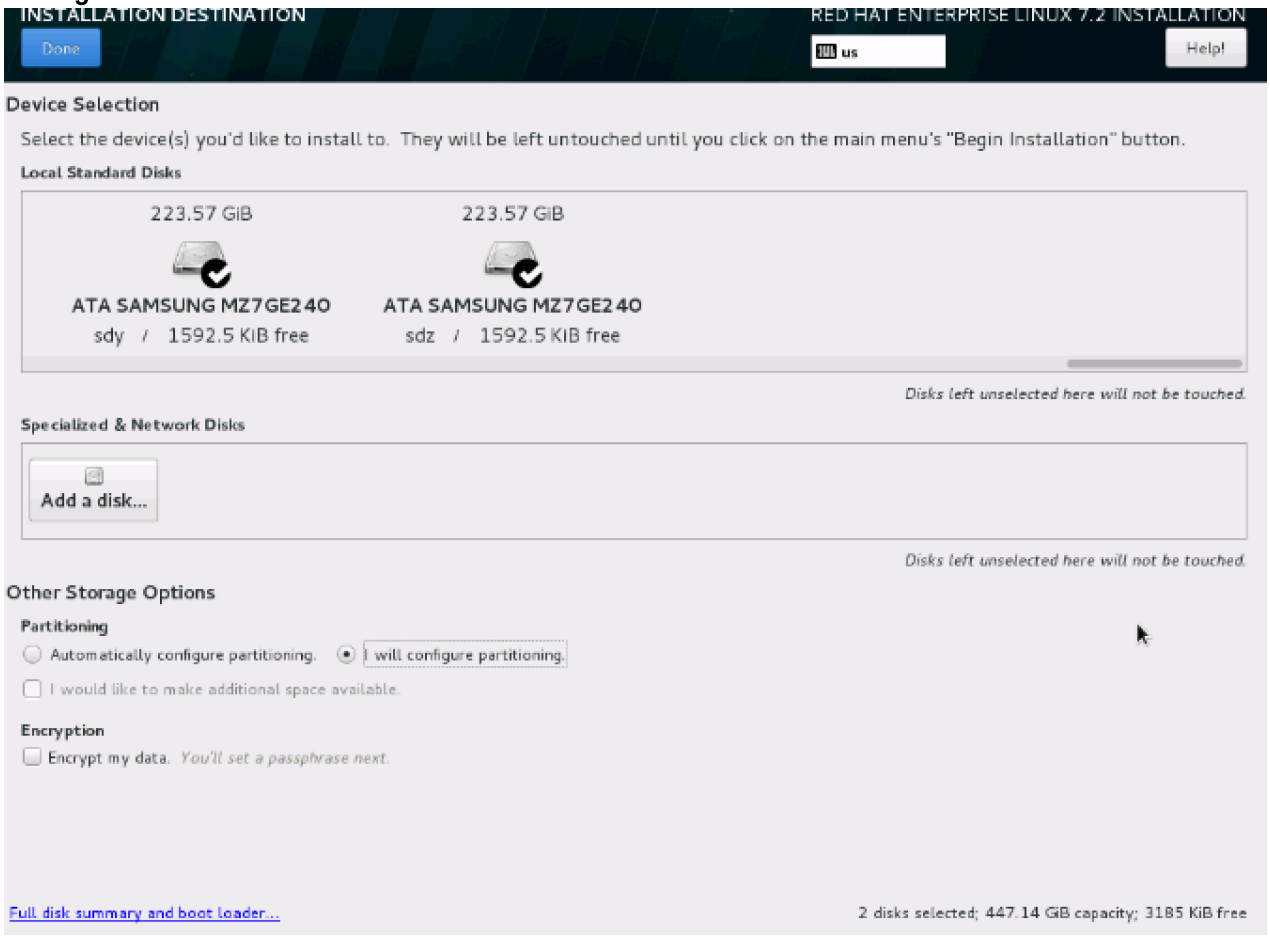


A Caution symbol appears next to Installation Destination as shown in Figure 89 above.

21. This opens the Installation Destination window displaying the boot disks. This is shown in Figure 90 below.

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

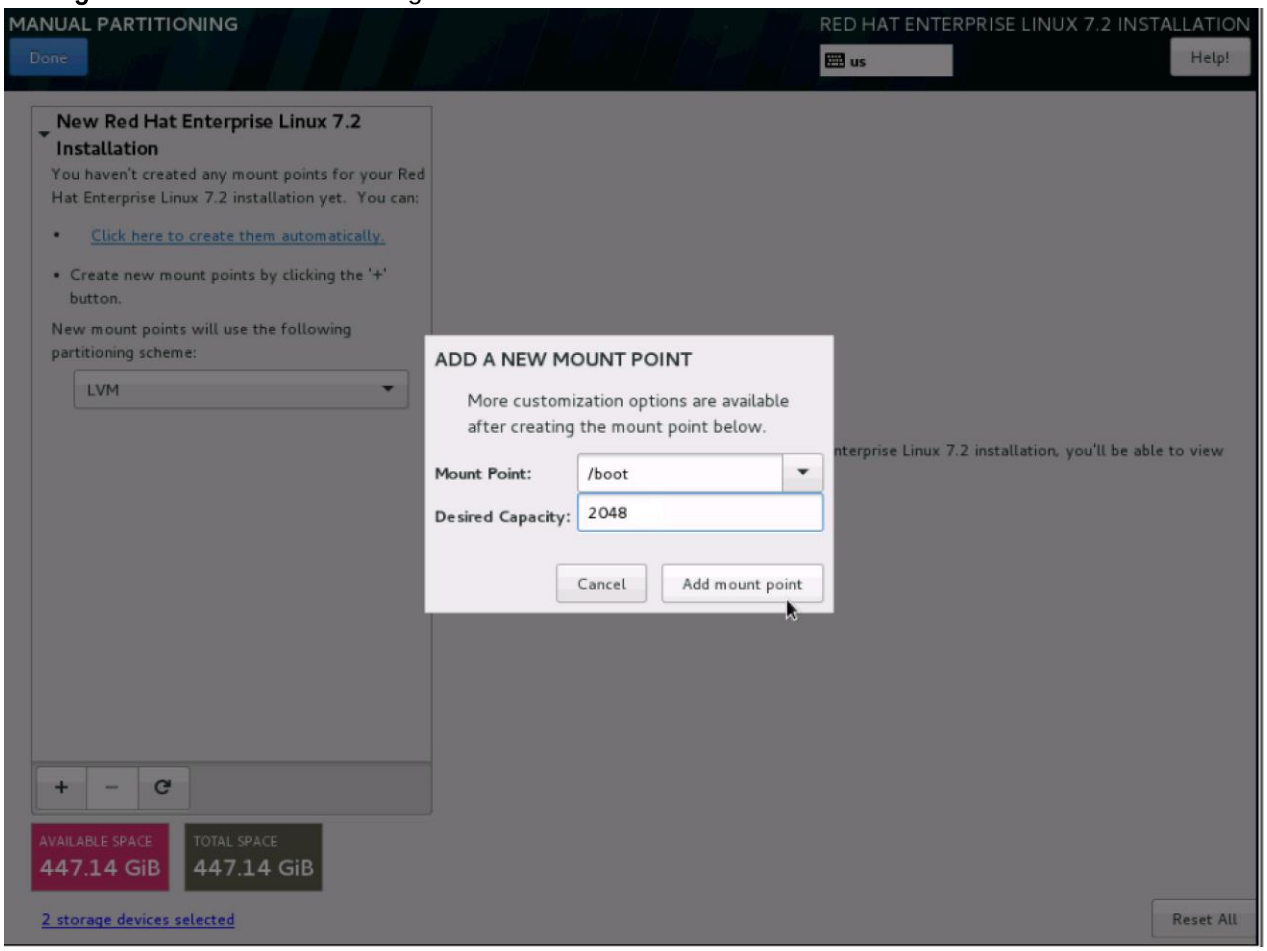
Figure 90 Installation Destination Window



This opens the new window for creating the partitions, as shown in Figure 91.

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

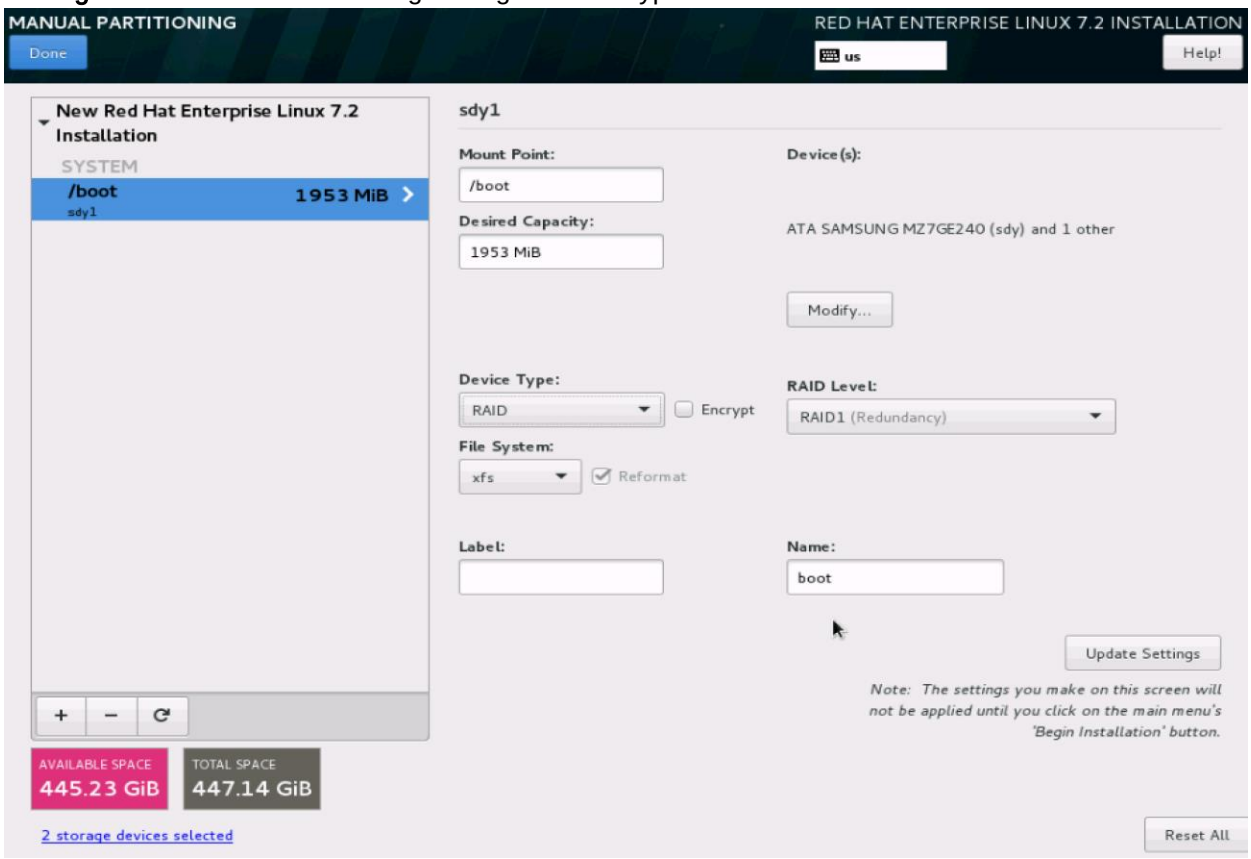
Figure 91 Manual Partitioning



24. Click Add Mount Point to add the partition.

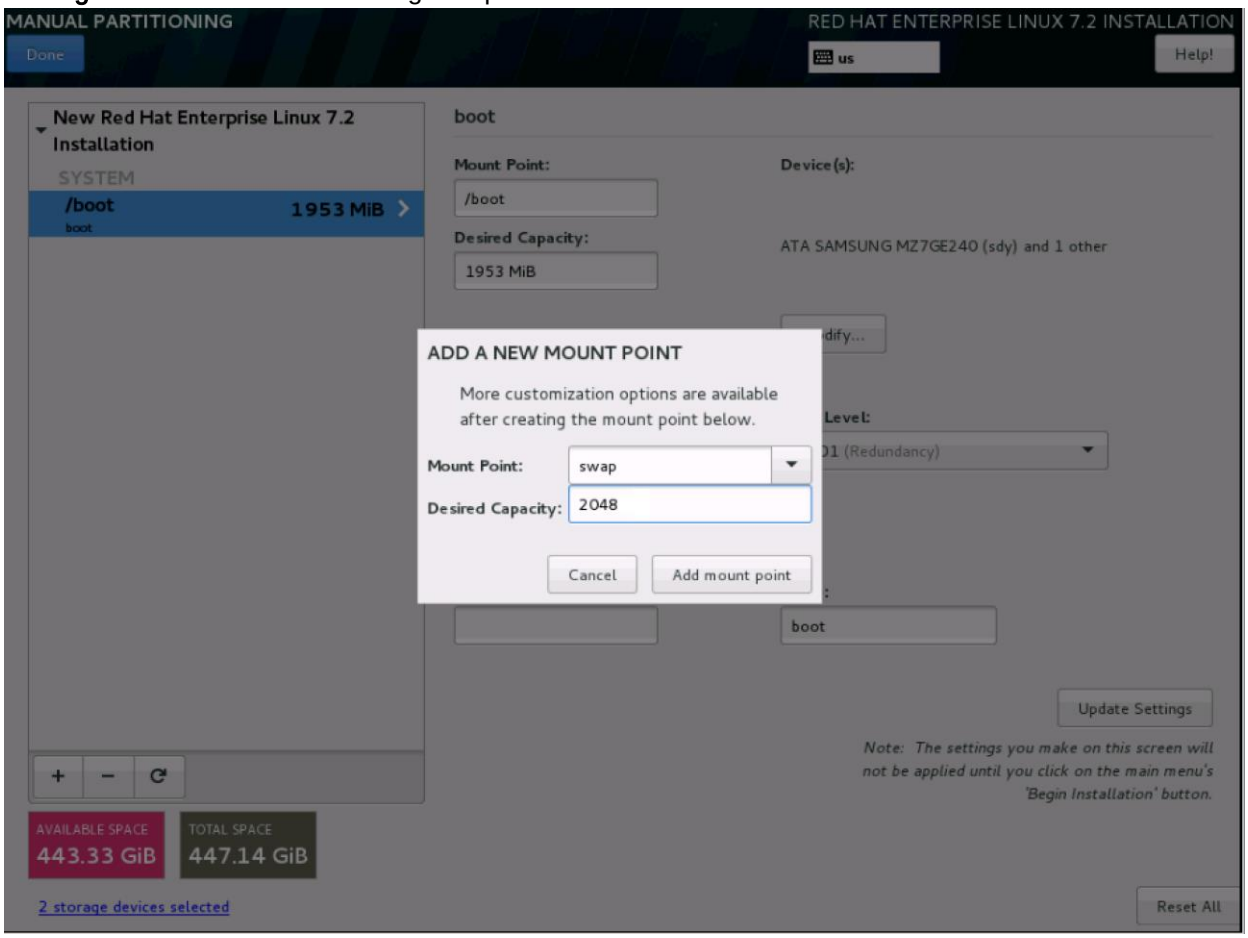
The screen refreshes to show the added Mount Point (Figure 92).

Figure 92 Manual Partitioning/Change Device Type



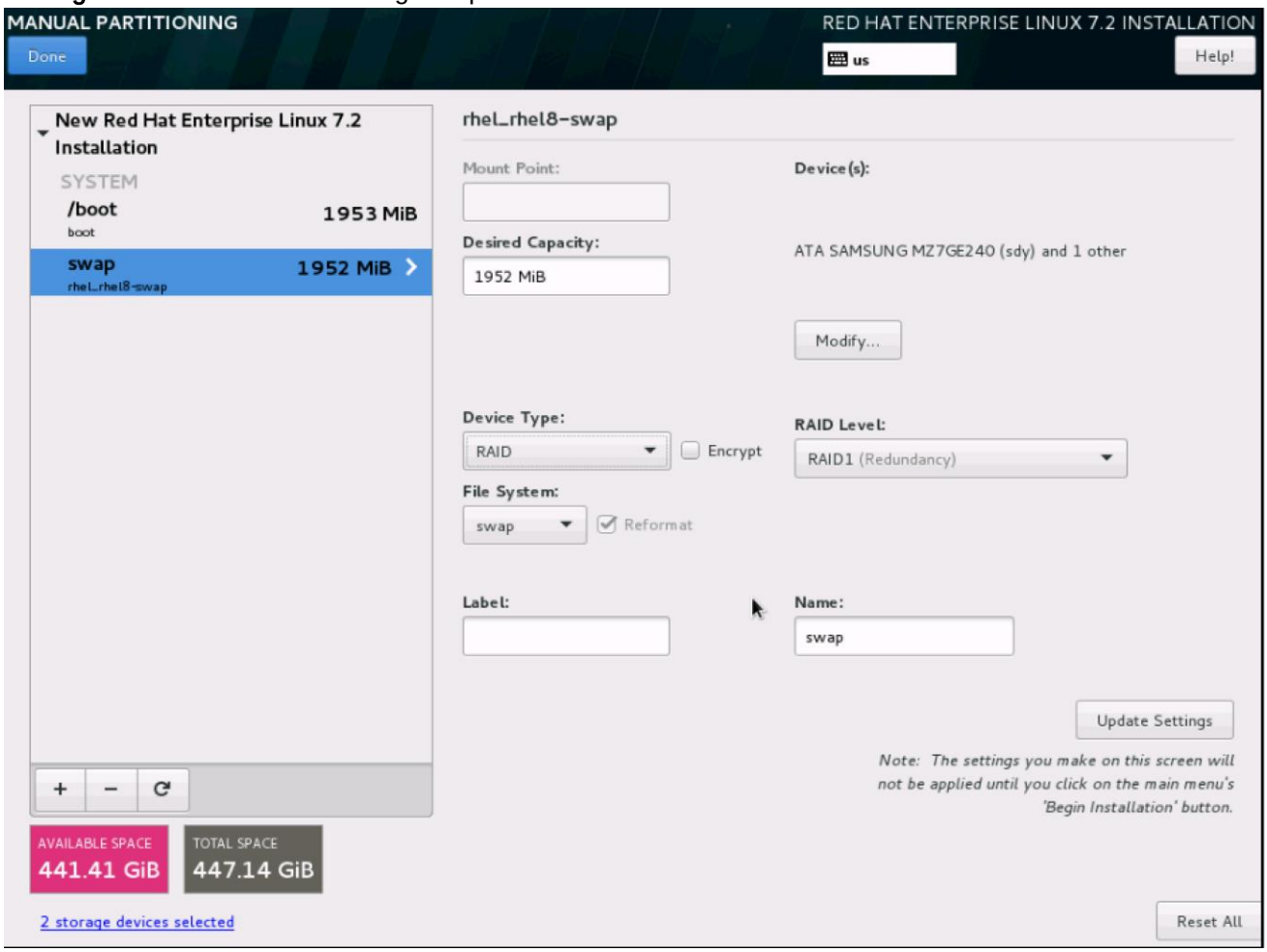
25. Change the Device type to RAID and make sure the RAID Level is RAID1 (Redundancy).
26. Click on Update Settings to save the changes.
27. Click on the + sign to create the swap partition of size 2048 MB as shown in Figure 93 below.

Figure 93 Manual Partitioning/Swap



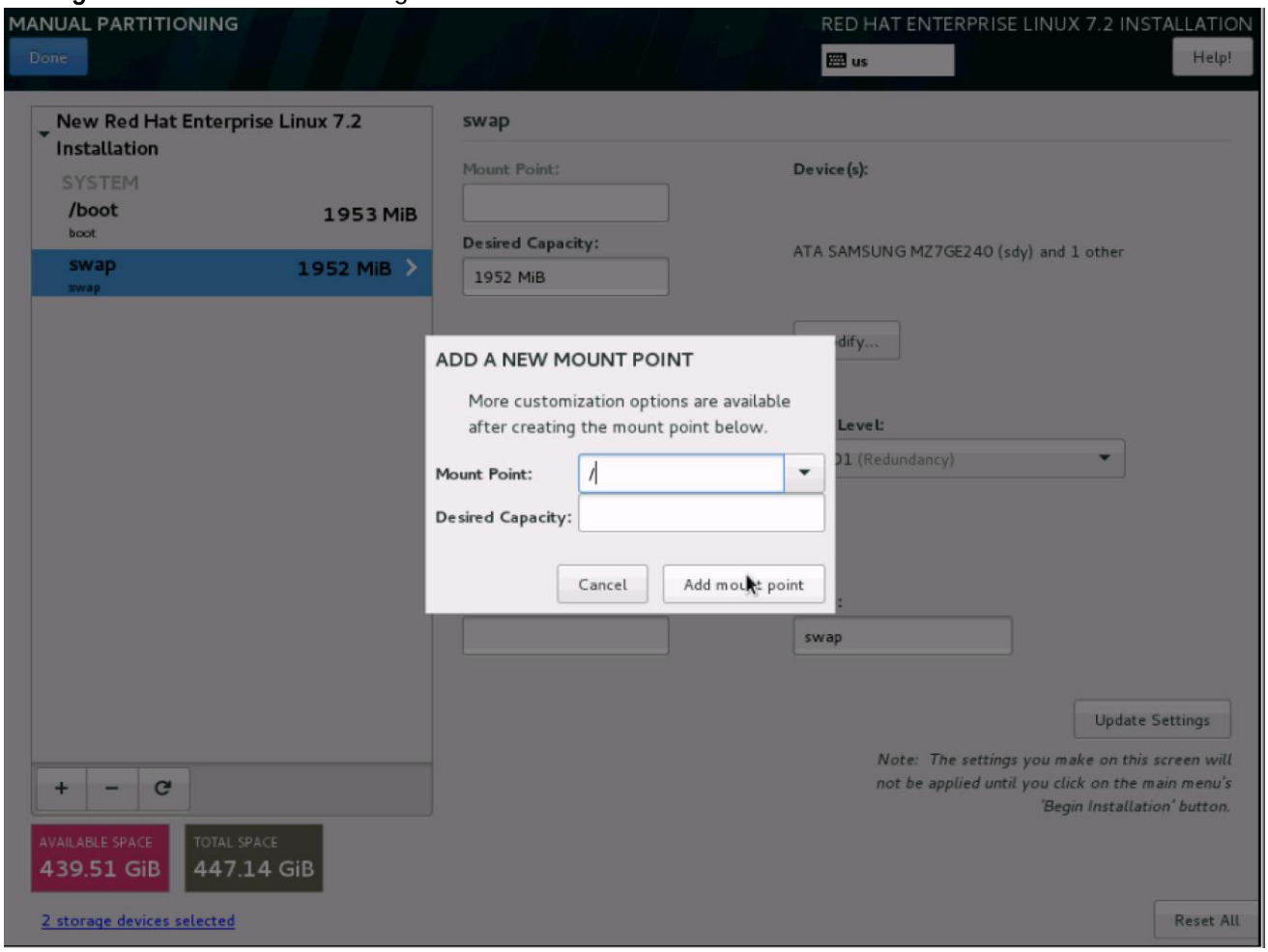
28. Change the Device type to RAID and RAID level to RAID1 (Redundancy) and click on Update Settings.

Figure 94 Manual Partitioning/Swap



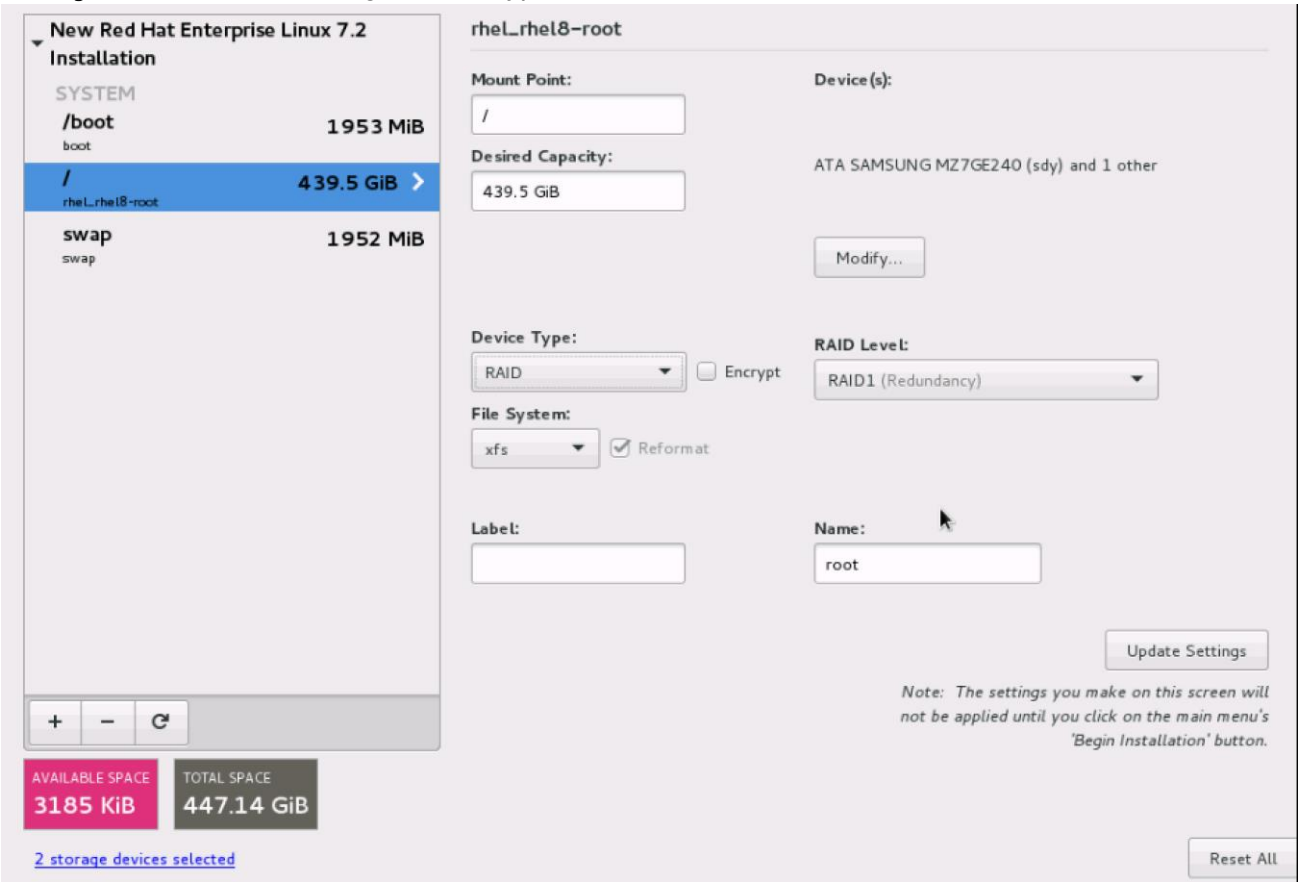
- Click + to add the / partition. The size can be left empty so it uses the remaining capacity and click Add Mountpoint. (Figure 95).

Figure 95 Manual Partitioning/Add A New Mount Point



29. In the next window (Figure 96), change the Device type to RAID and RAID level to RAID1 (Redundancy). Click Update Settings.

Figure 96 Partition/Change Device Type to RAID

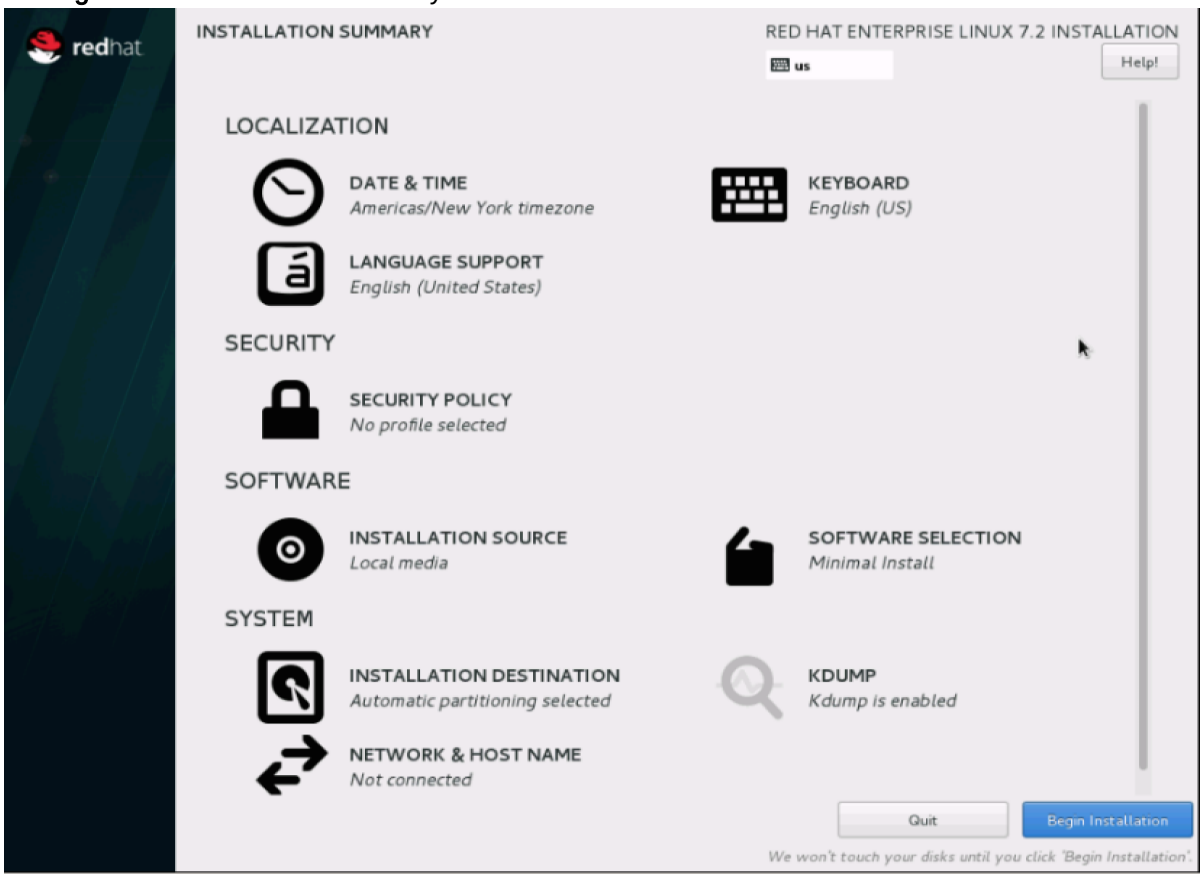


30. Click Done to go back to the main screen and continue the Installation.

The Installation screen opens (Figure 97).

31. Click on Software Selection.

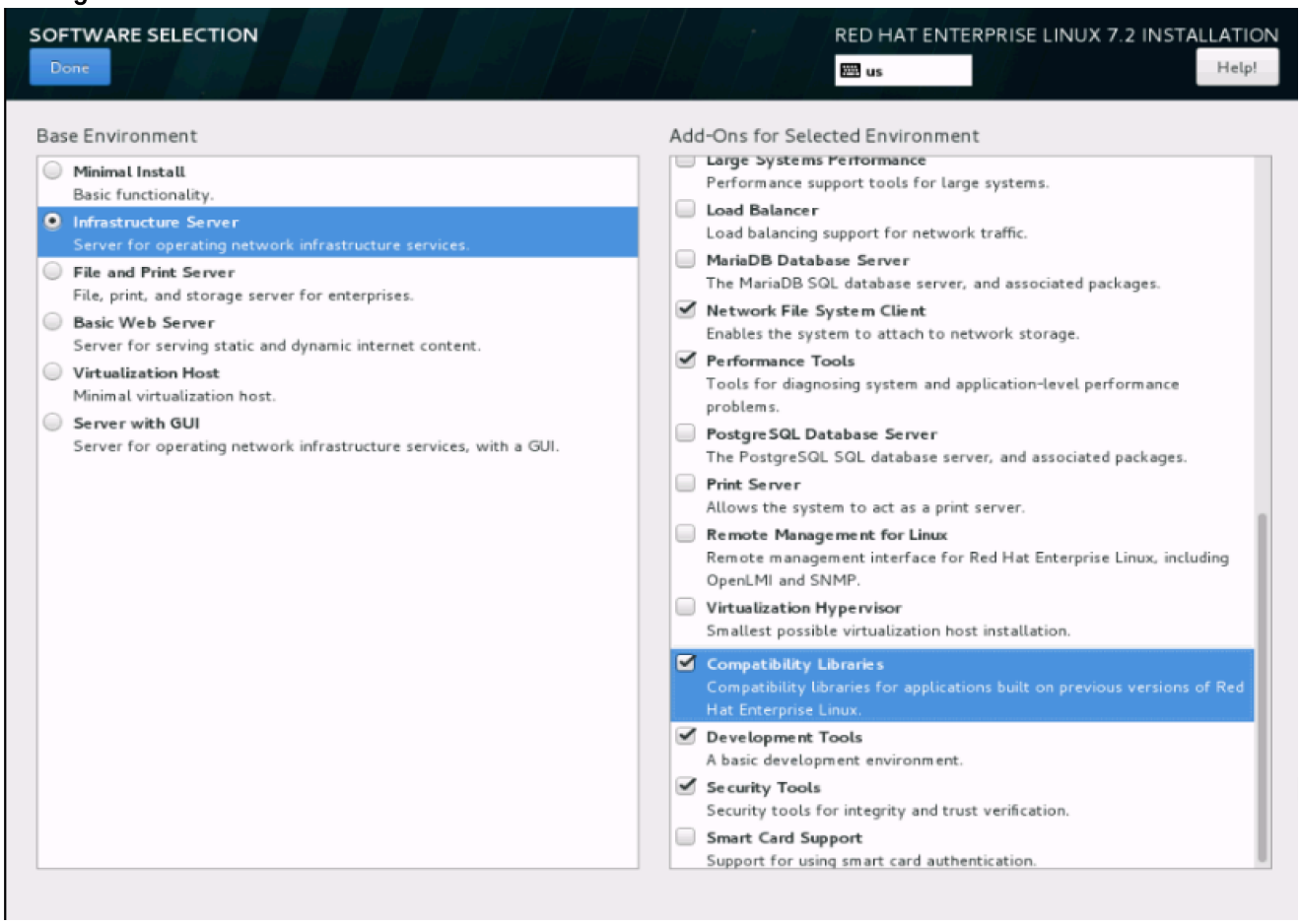
Figure 97 Installation Summary Window



The Software Selection screen opens (Figure 98).

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

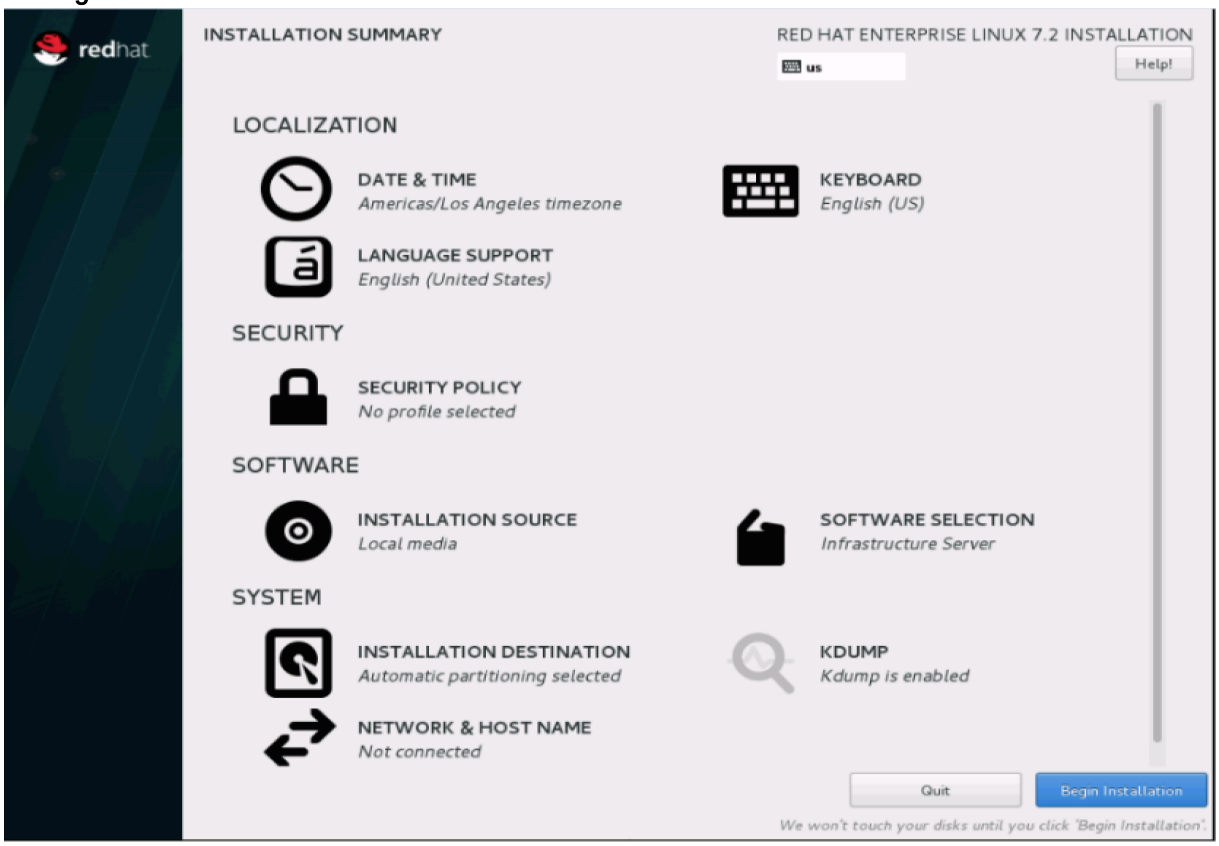
Figure 98 Software Selection



The Installation Summary window returns (Figure 99).

33. Click on Network and Hostname.

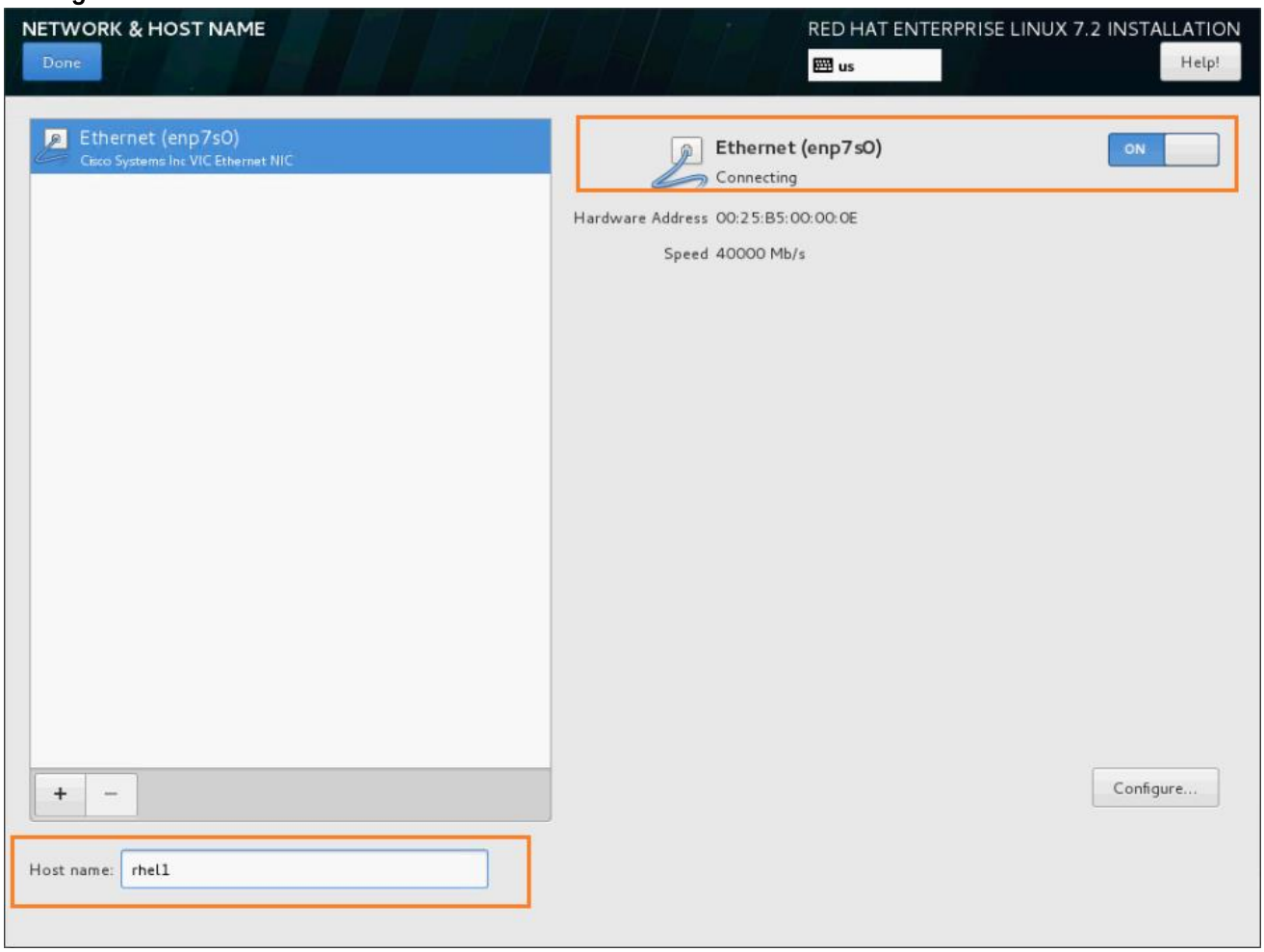
Figure 99 Network and Host Name Window



Configure Hostname and Networking for the Host (Figure 100).

34. Type in the hostname as shown below.

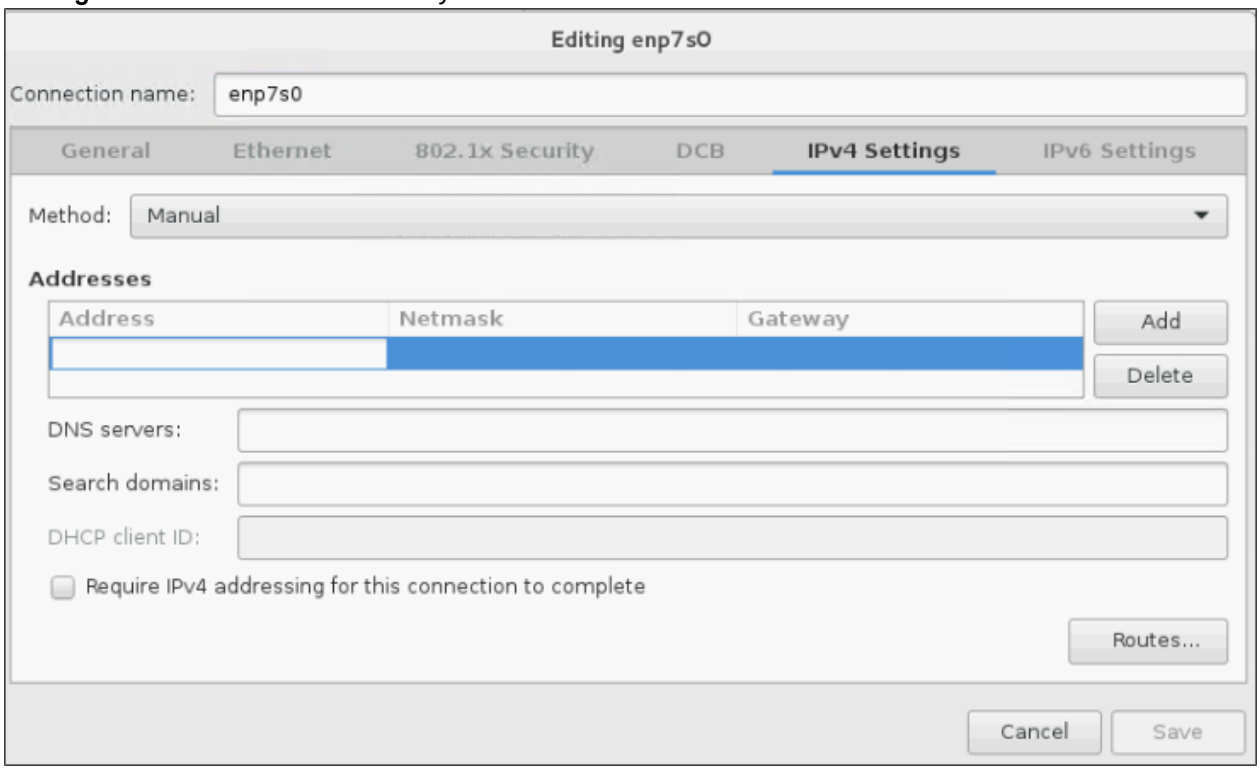
Figure 100 Network and Host Name



35. Click on Configure to open the Network Connectivity window (Figure 101).

36. Click on IPV4Settings.

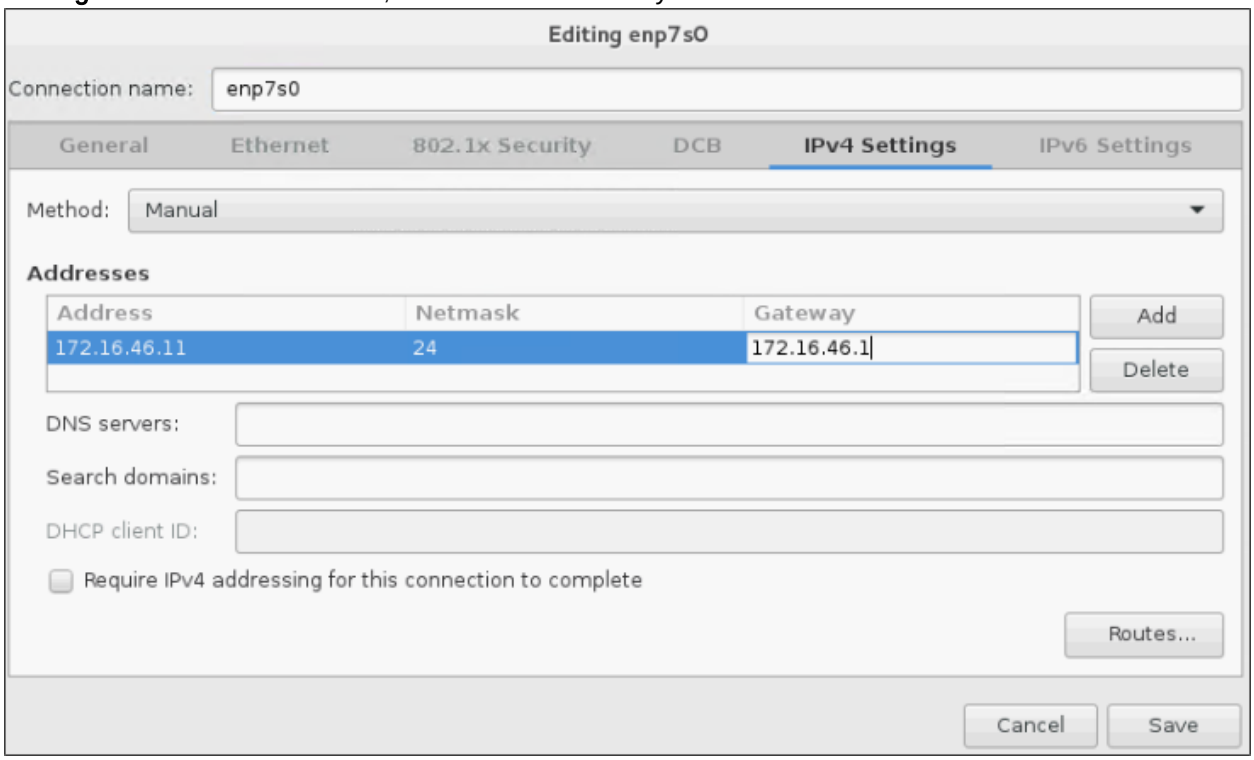
Figure 101 Network Connectivity Window



37. Change the Method to Manual and click Add. Figure 102 shows the Add Details pop up window.

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

Figure 102 Add IP Address, Netmask and Gateway Details



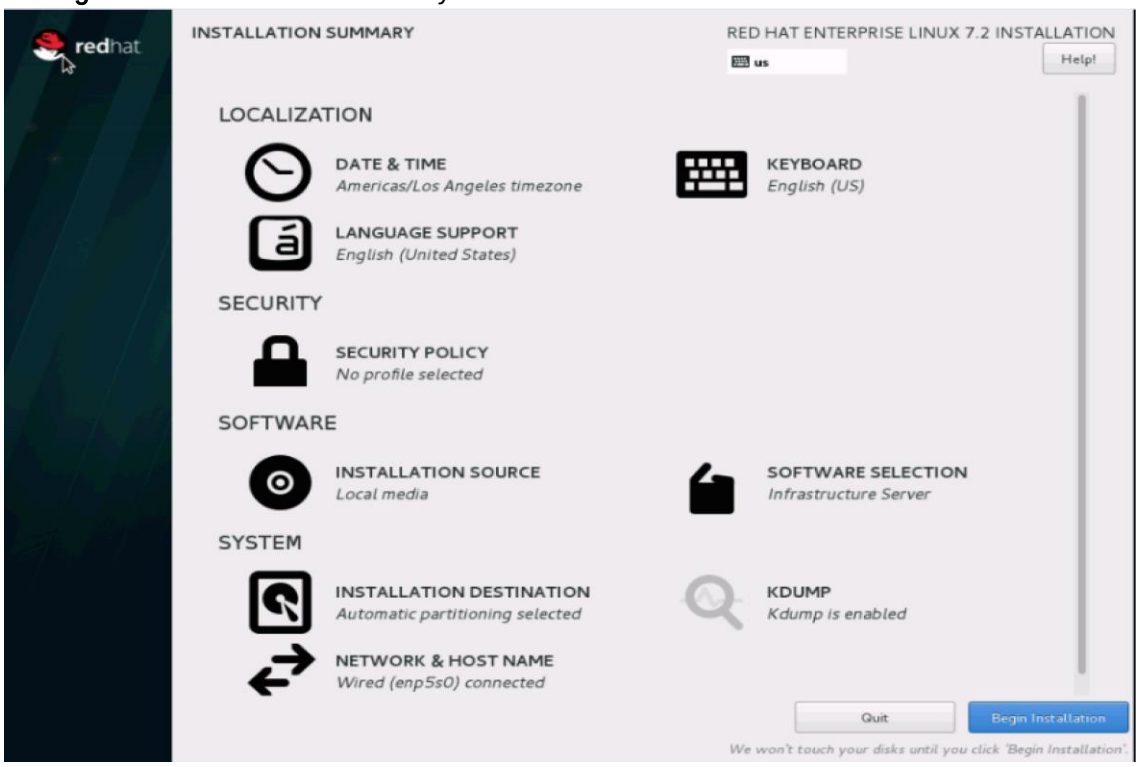
39. Click Save.

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

The Installation Summary window opens (Figure 103).

41. Click Begin Installation in the main menu.

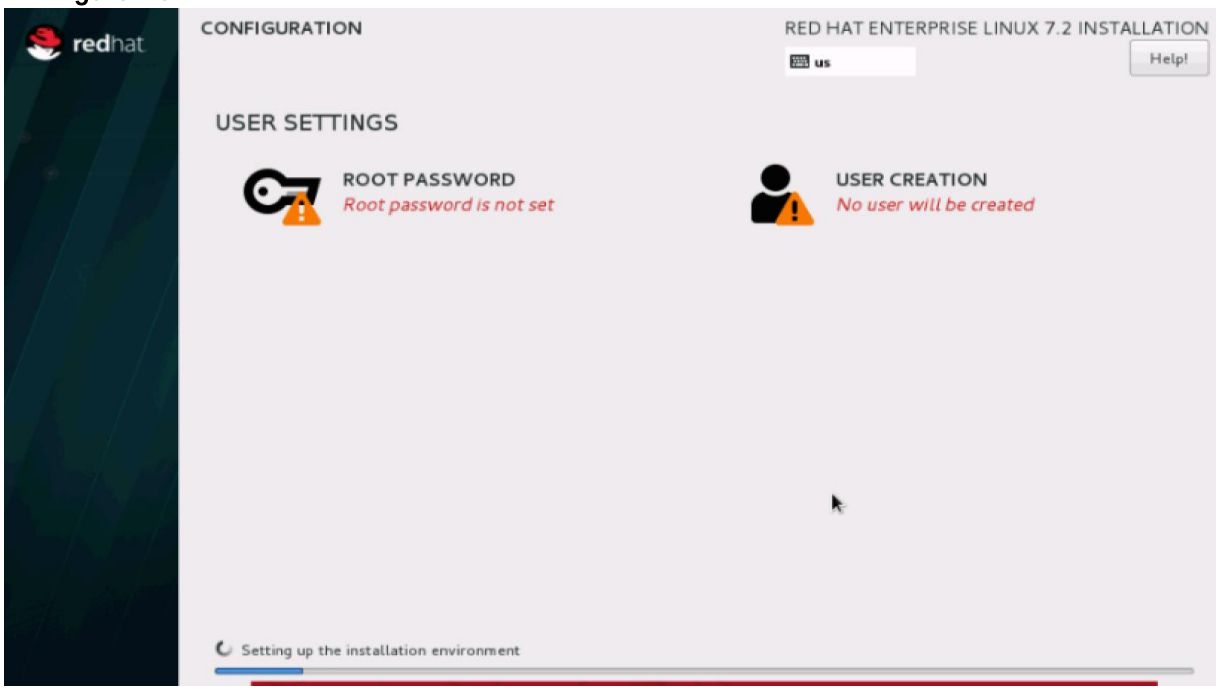
Figure 103 Installation Summary Window



A new window opens (Figure 104).

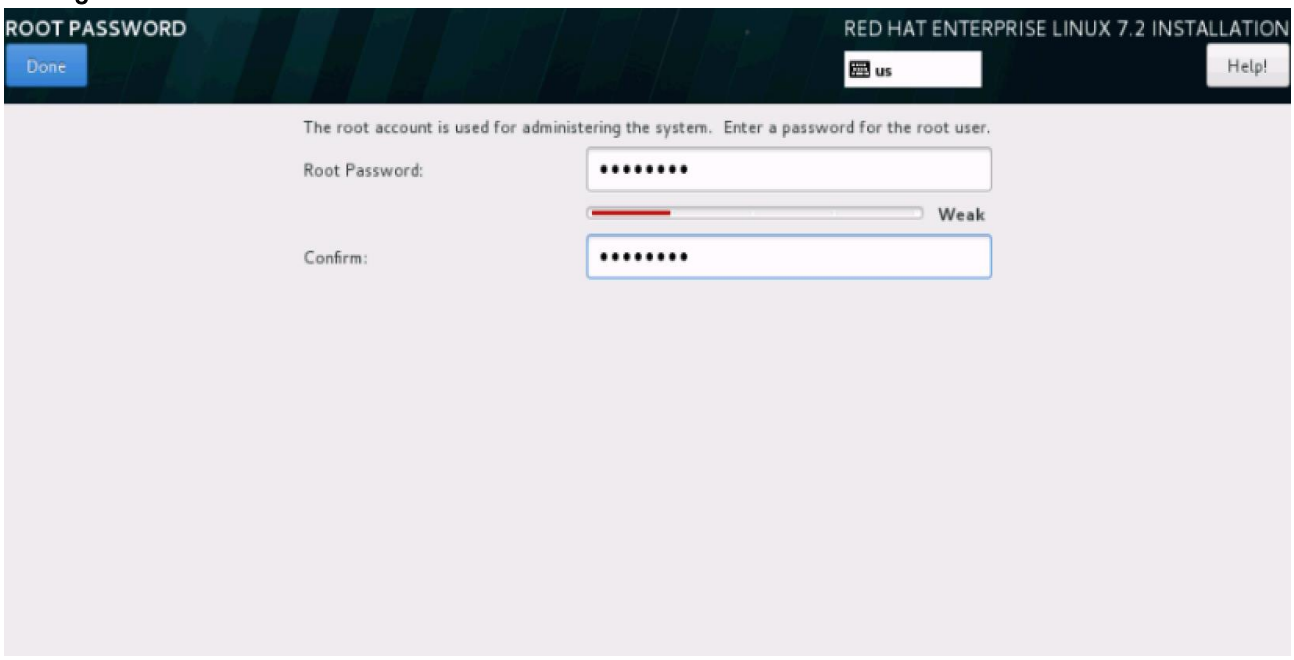
42. Select Root Password in the User Settings.

Figure 104 Select Root Password



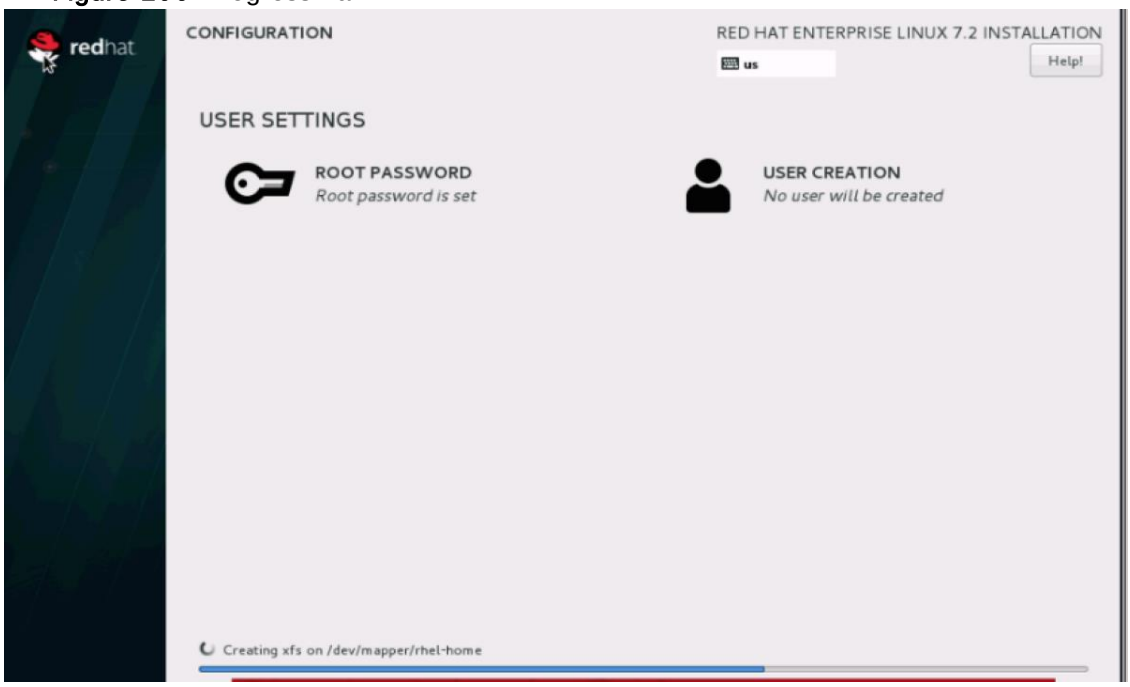
43. On the next screen (Figure 105), enter the Root Password and click done.

Figure 105 Enter the Root Password



A progress window will open (Figure 106).

Figure 106 Progress Bar



44. Once the installation is complete reboot the system.

45. Repeat steps 1 to 43 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

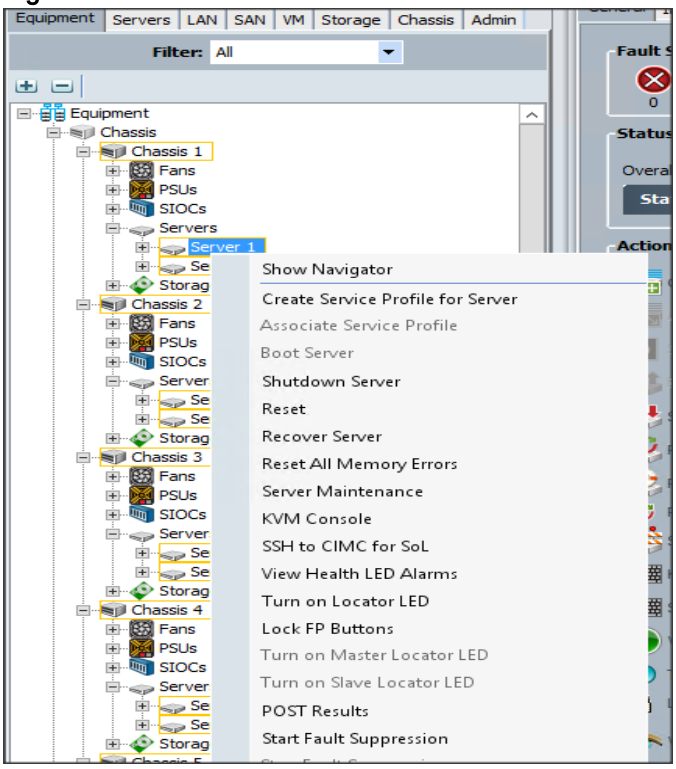
The following section provides detailed procedures for installing Red Hat Enterprise Linux 7.2 on Cisco UCS S3260 Storage Servers. There are multiple ways to install the Red Hat Linux operating system. The installation procedure described in this deployment guide uses KVM console and virtual media from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

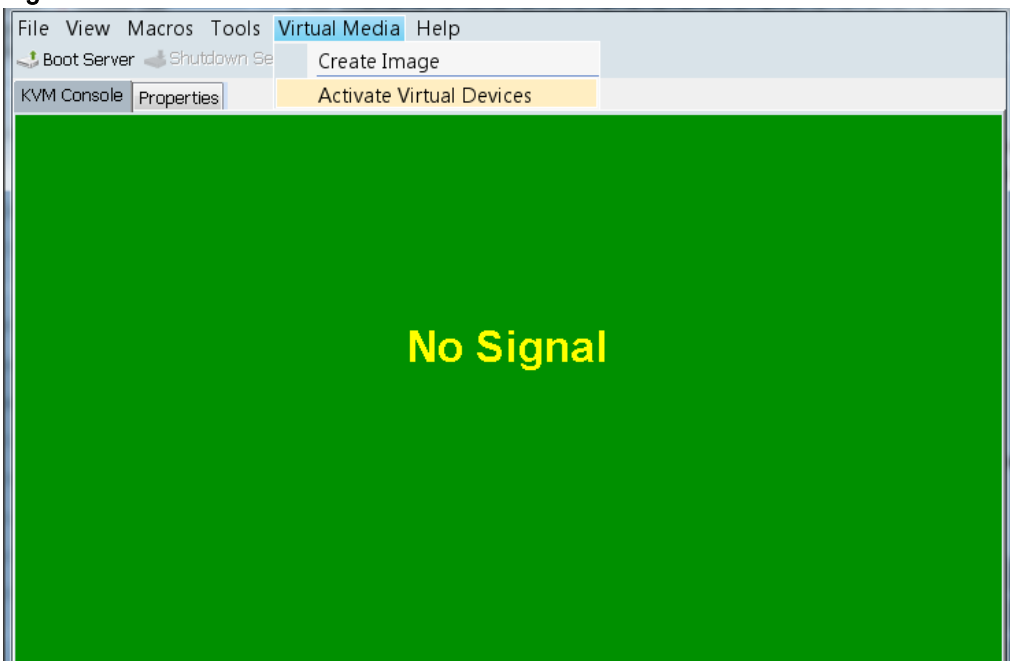
1. Log in to the Cisco UCS 6332 Fabric Interconnect and launch the Cisco UCS Manager application.
2. Select the Equipment tab.
3. In the navigation pane expand Chassis and then Servers.
4. Right click on the server and select KVM Console. (Figure 107)
5. In the KVM window, select the Virtual Media tab.

Figure 107 KVM Console



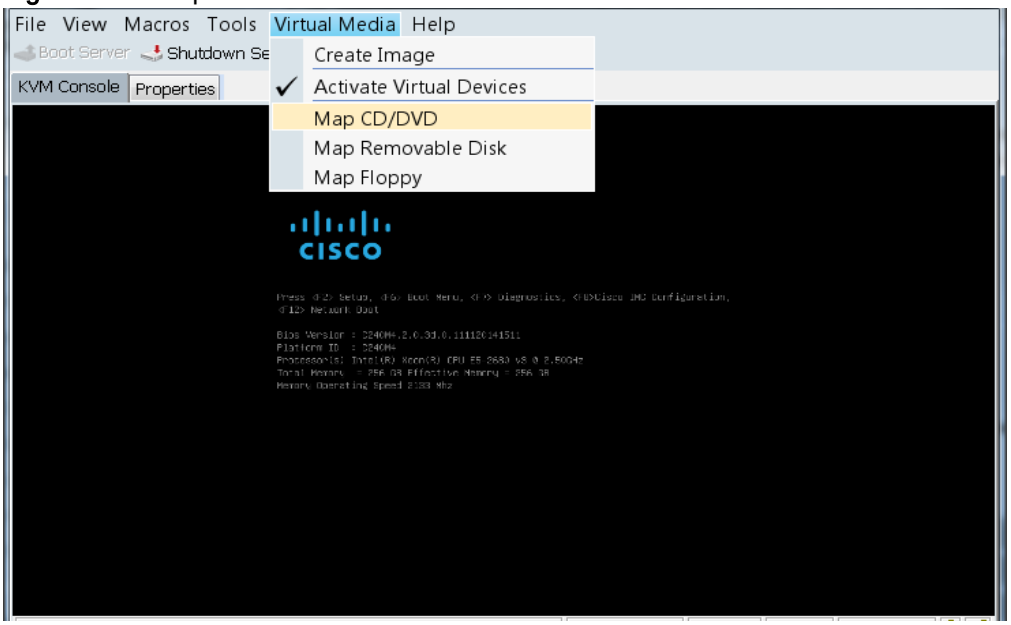
6. Click the Activate Virtual Devices found in the Virtual Media tab. (Figure 108)

Figure 108 Activate Virutal Devices



7. In the KVM window, select the Virtual Media tab and click the Map CD/DVD. (Figure 109)

Figure 109 Map CD/DVD

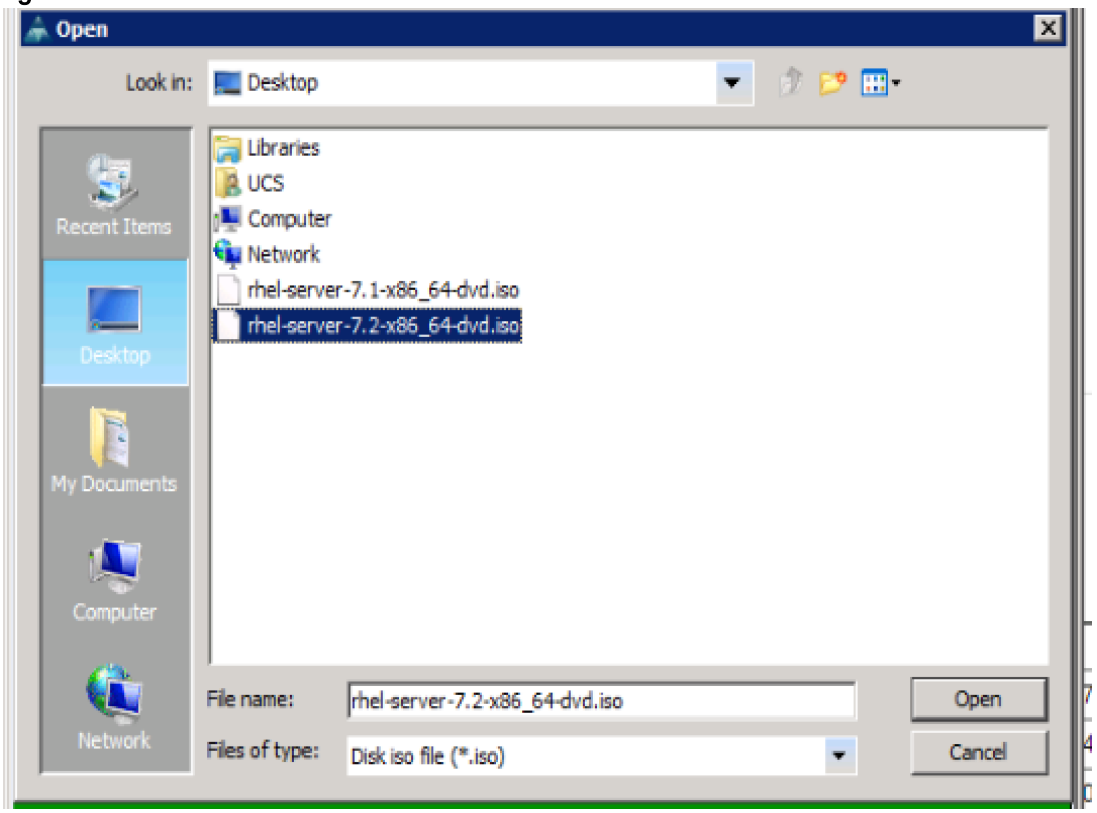


8. Browse to the Red Hat Enterprise Linux Server 7.2 installer ISO image file.

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

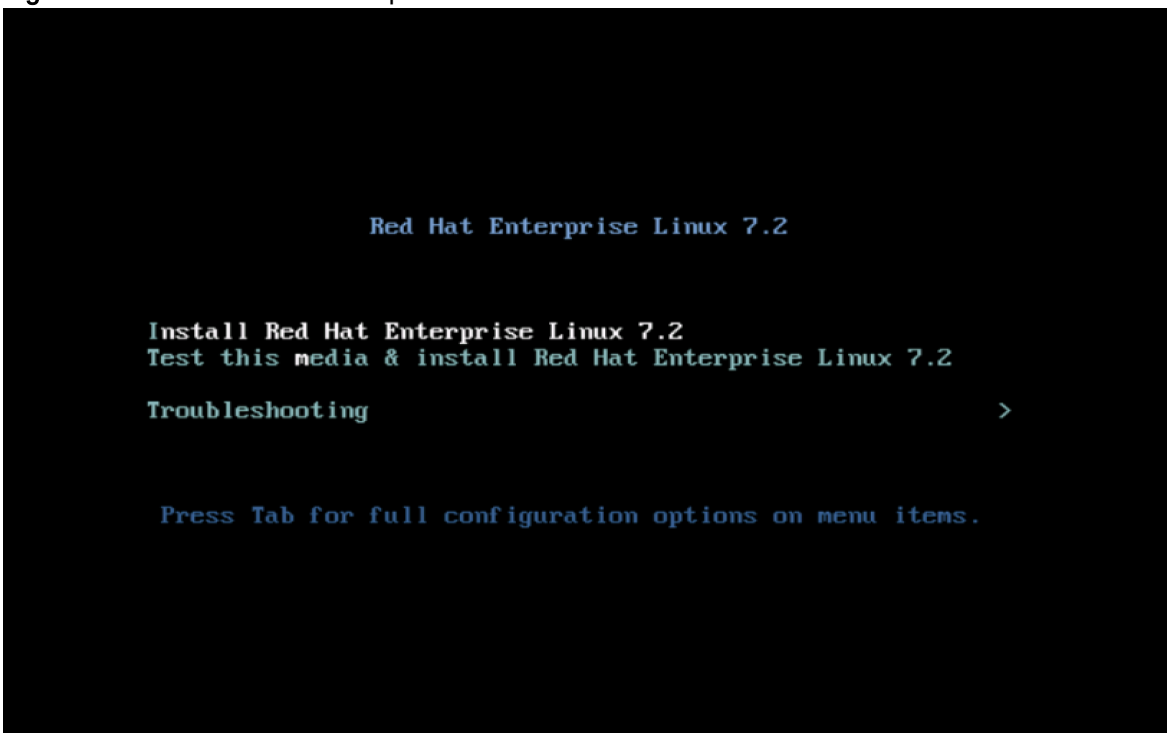
9. Click Open to add the image to the list of virtual media. (Figure 110)

Figure 110 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.

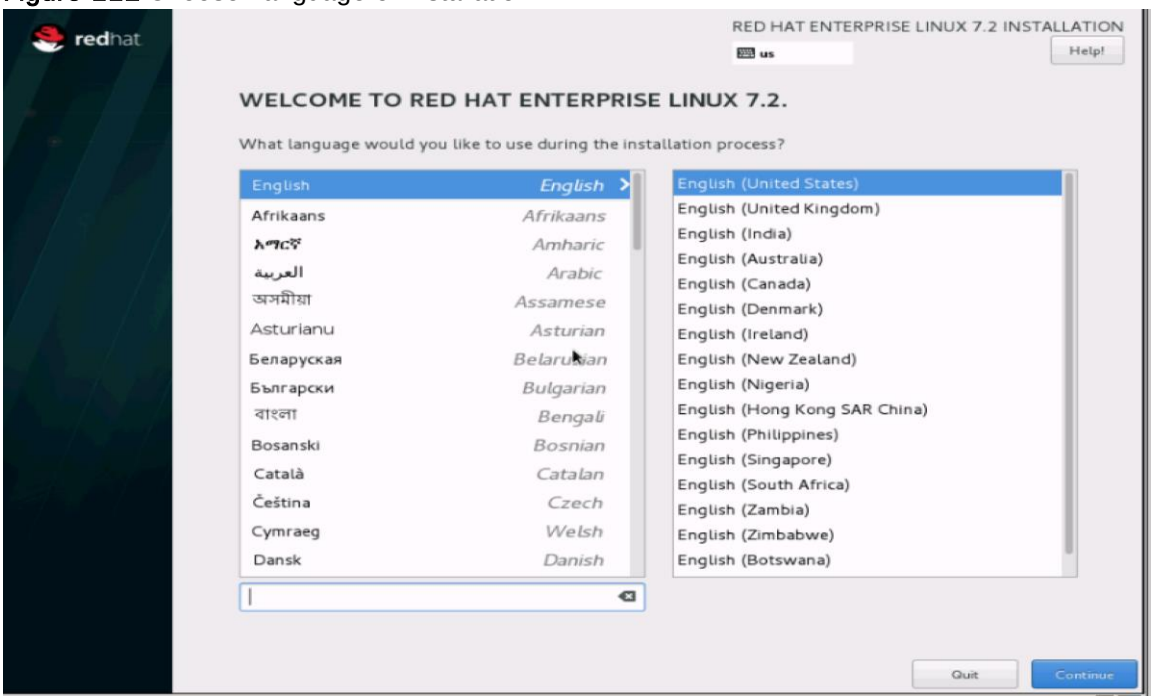
Figure 111 Install Red Hat Enterprise Linux 7.2



16. Skip the Media test and start the installation. (Figure 111)

17. Select language of installation and click Continue. (Figure 112)

Figure 112 Choose Language of Installation



18. Select Date and Time, (Figure 113) which pops up another window as shown below in Figure 114.

Figure 113 Date and Time

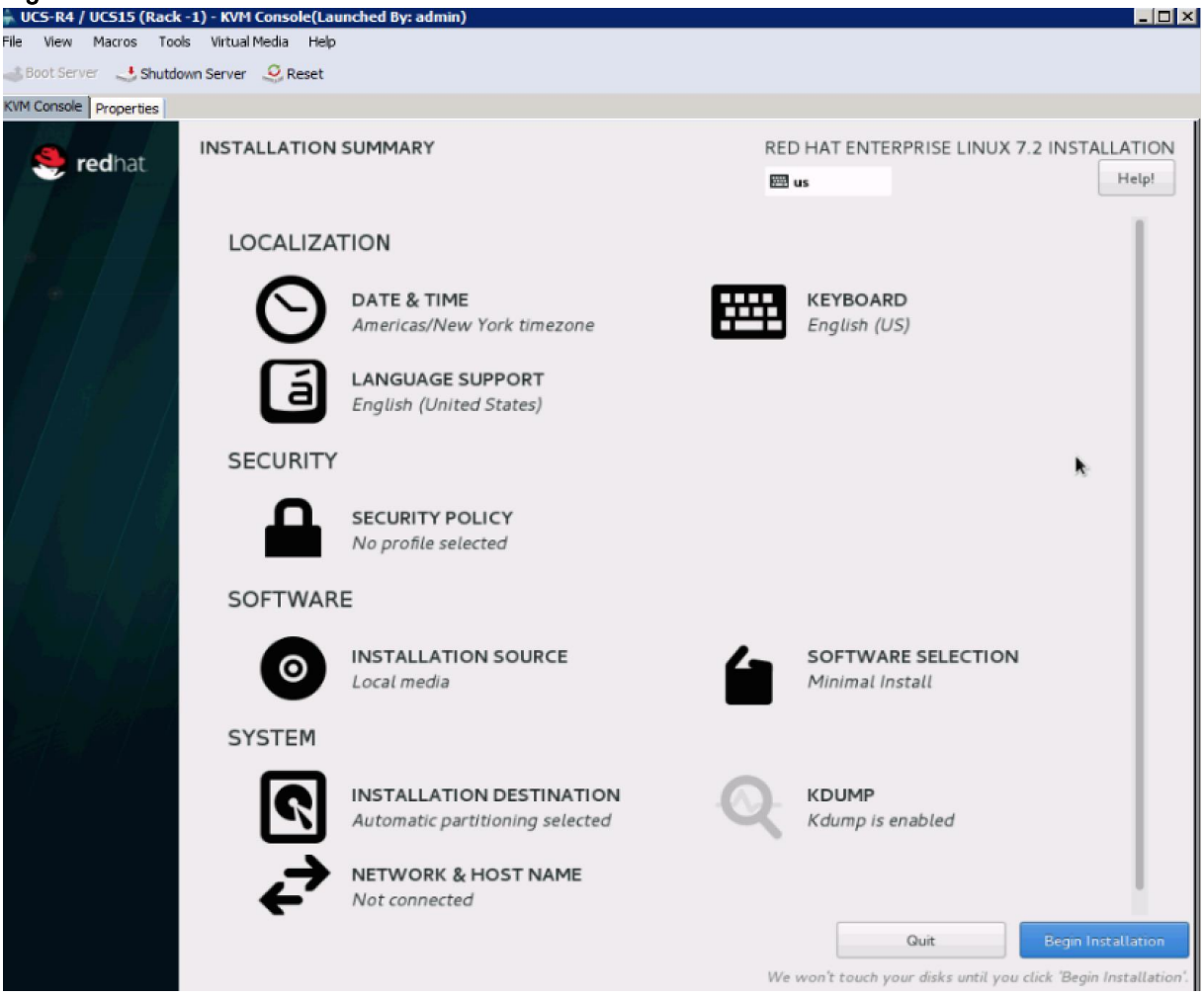
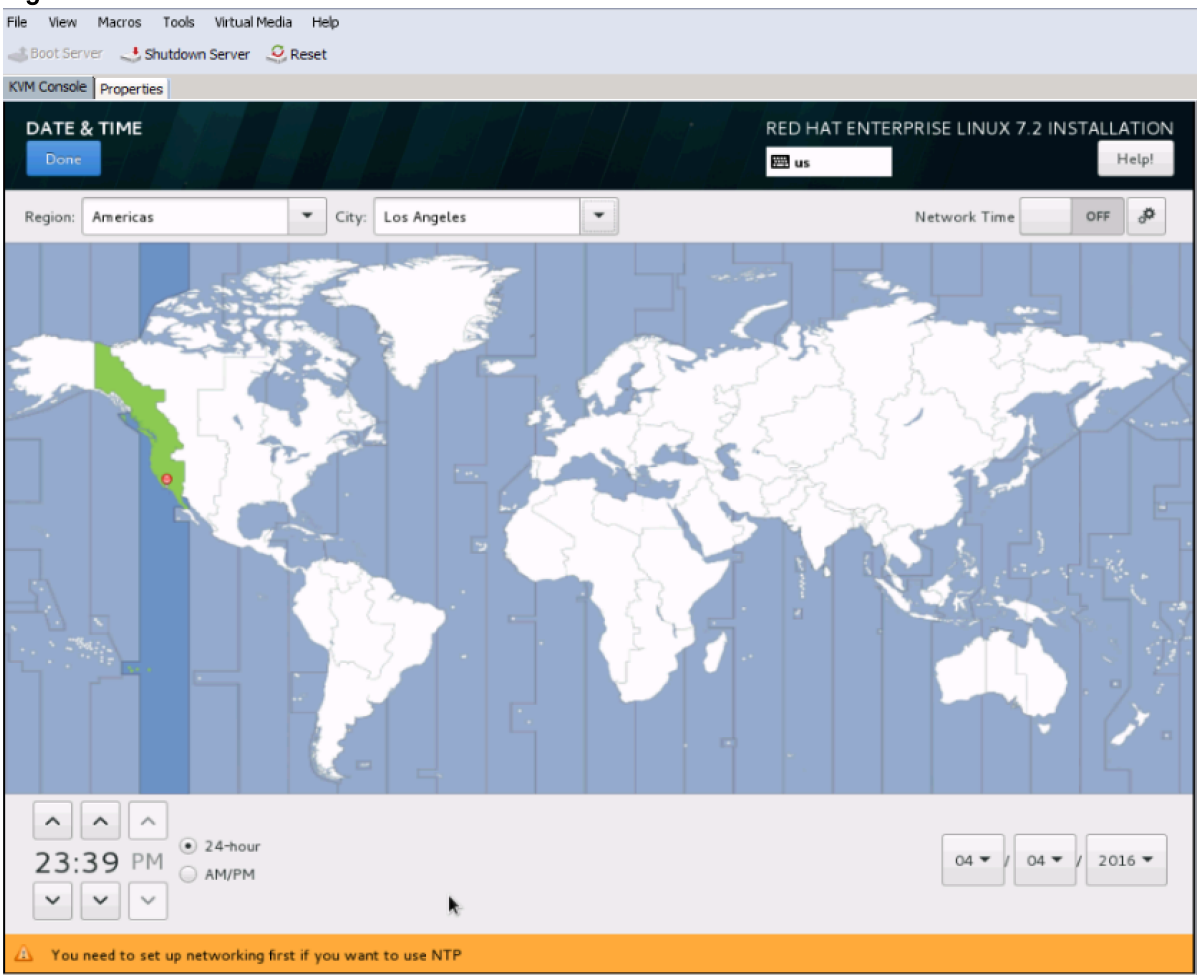


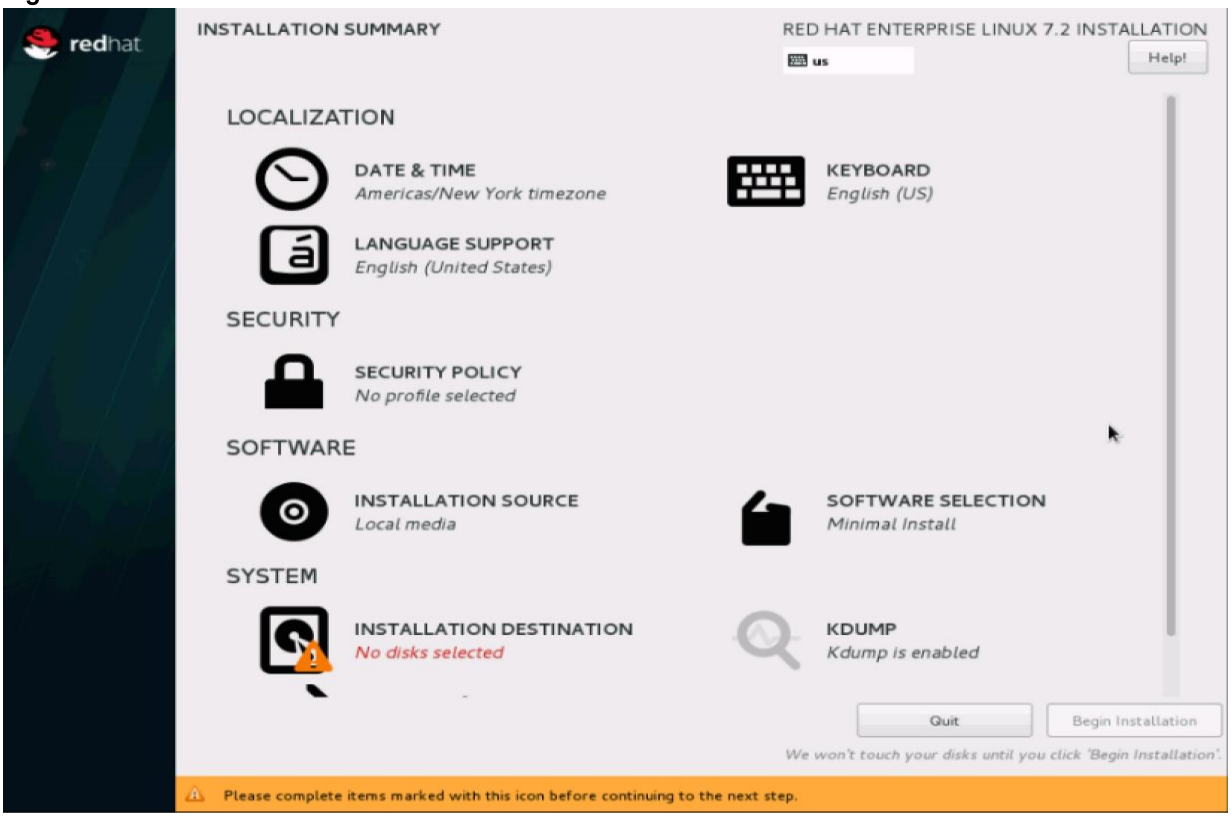
Figure 114 Choose Time Zone



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

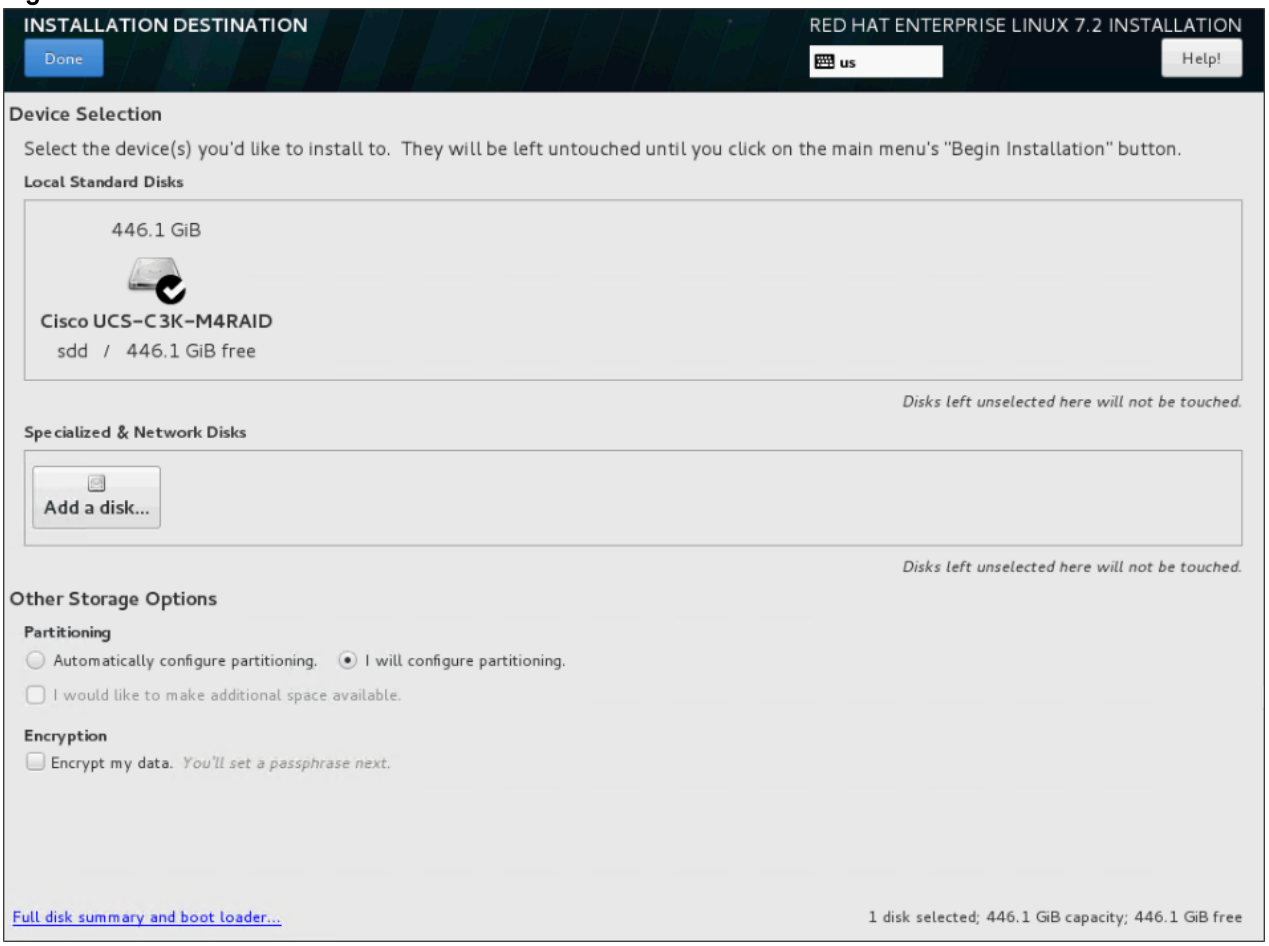
20. Click on Installation Destination. (Figure 115)

Figure 115 Installation Destination



21. This opens a new window with the boot disks. Make the selection, and choose I will configure partitioning. Click Done. (Figure 116)

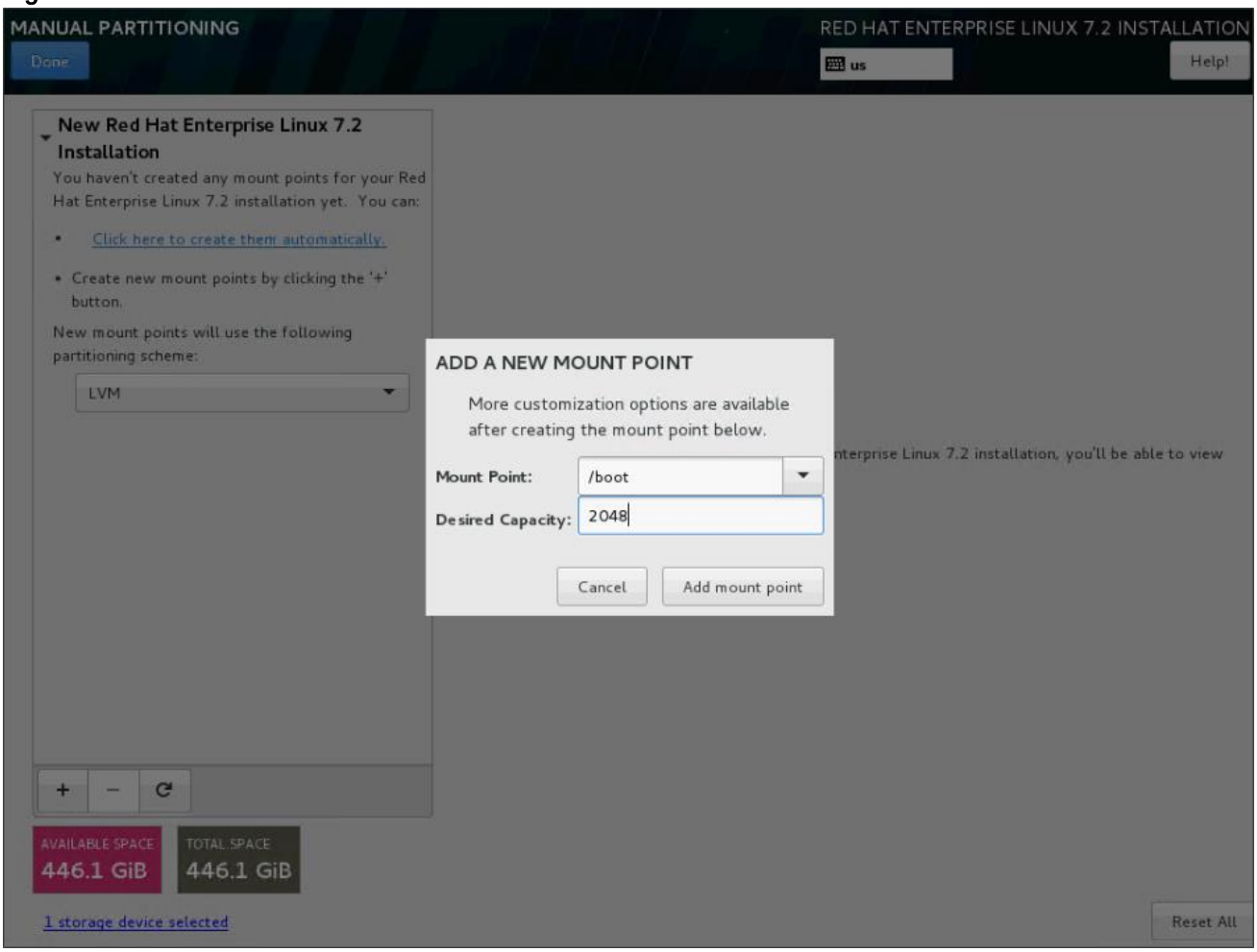
Figure 116 Installation Destination



22. This opens the new window for creating the partitions. (Figure 117) Click on the + sign to add a new partition as shown below, boot partition of size 2048 MB.

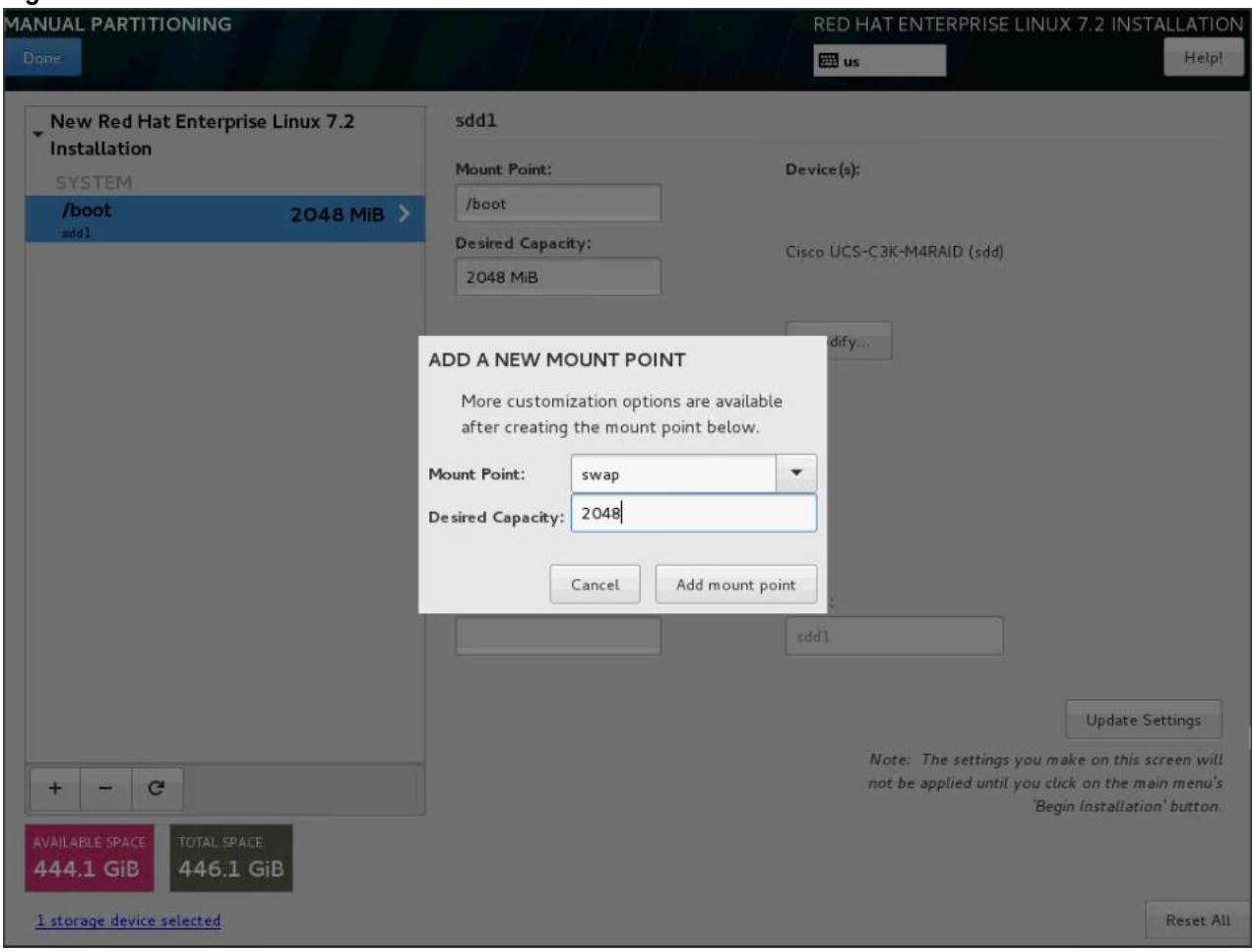
23. Click Add MountPoint to add the partition.

Figure 117 Add a New Mount Point



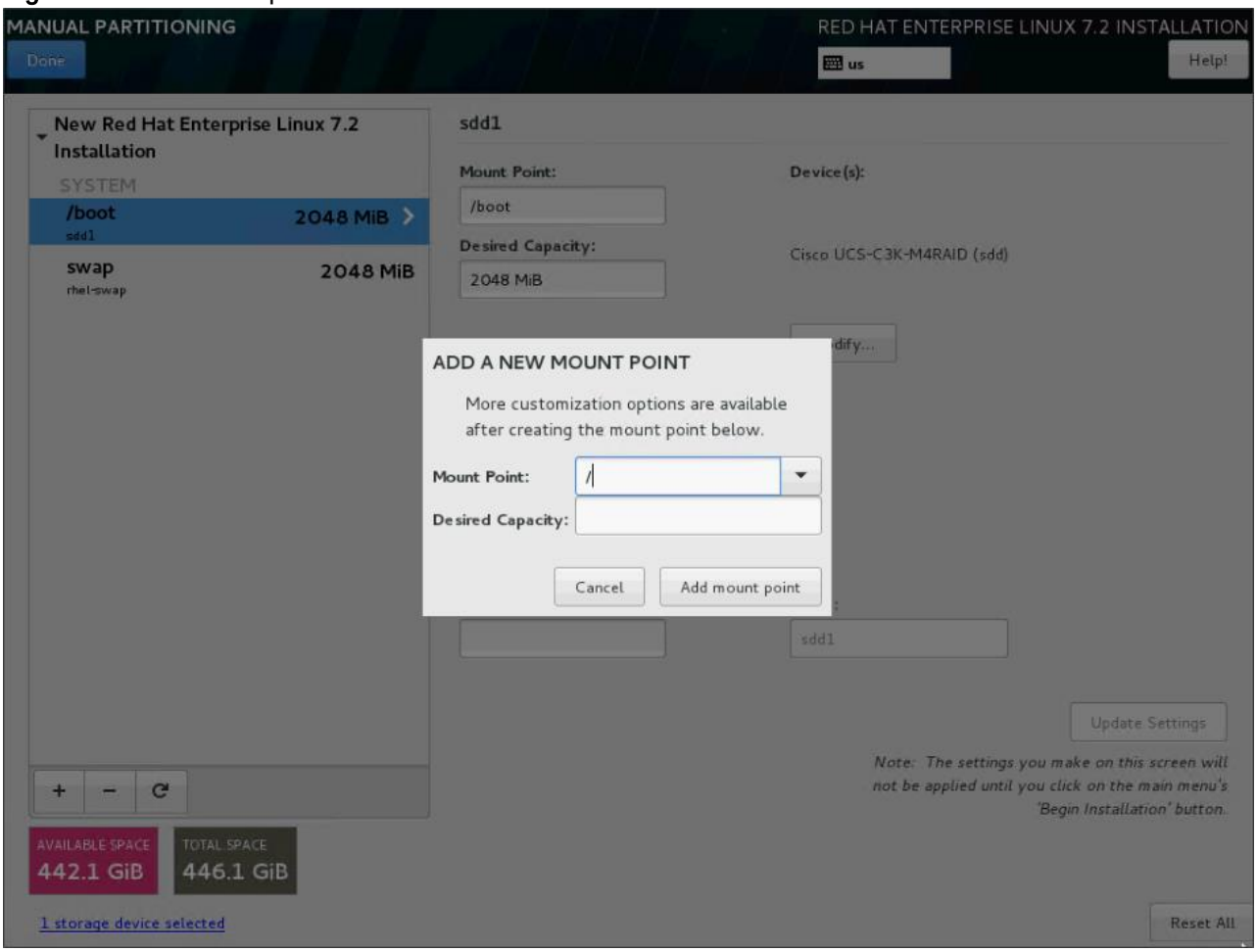
24. Click on the + sign to create the swap partition of size 2048 MB as shown below. (Figure 118)

Figure 118 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 119)

Figure 119 Add a swap

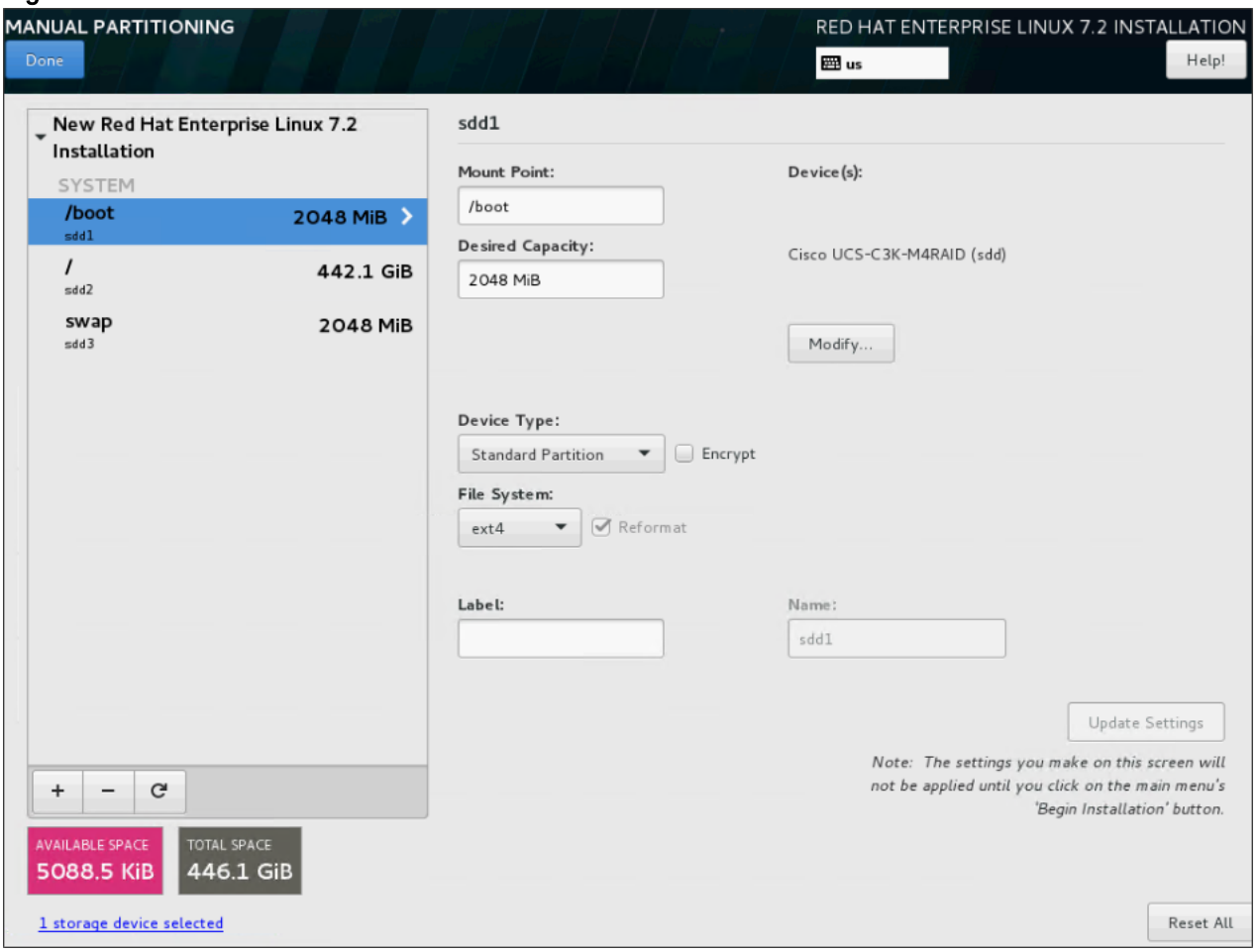


26. Select /boot partition and change the Device Type to Standard Partition and the file system to ext4. (Figure 120)

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.

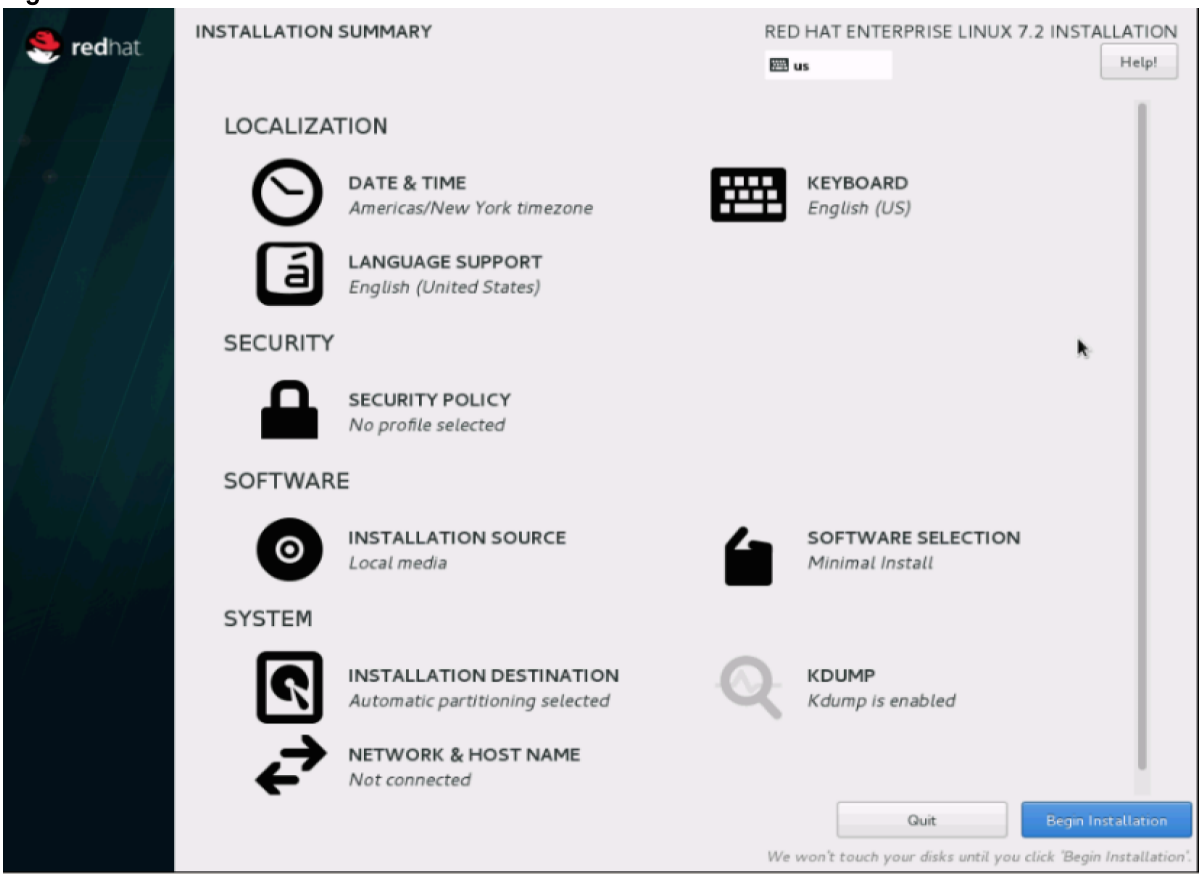
Figure 120 Standard Partition



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

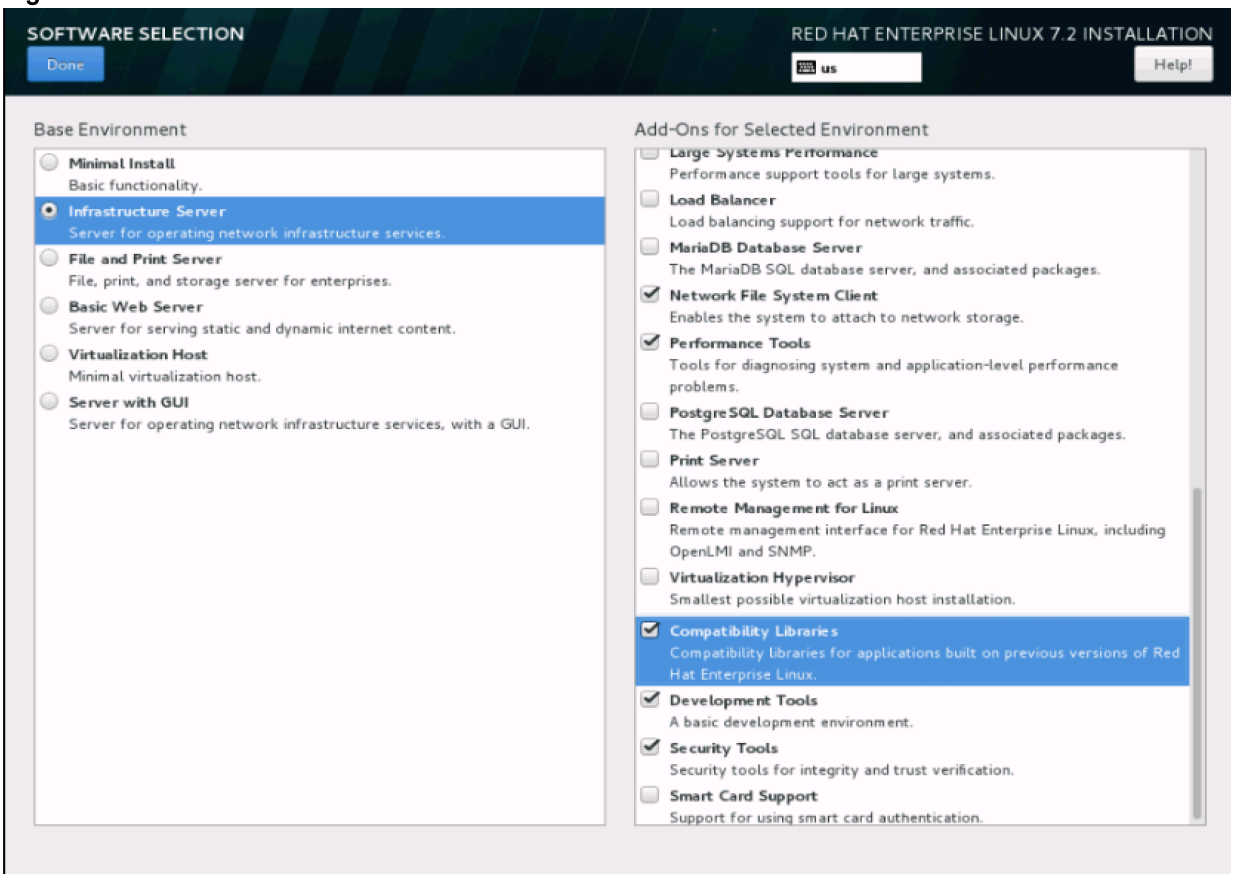
30. Click on Software Selection. (Figure 121)

Figure 121 Software Selection



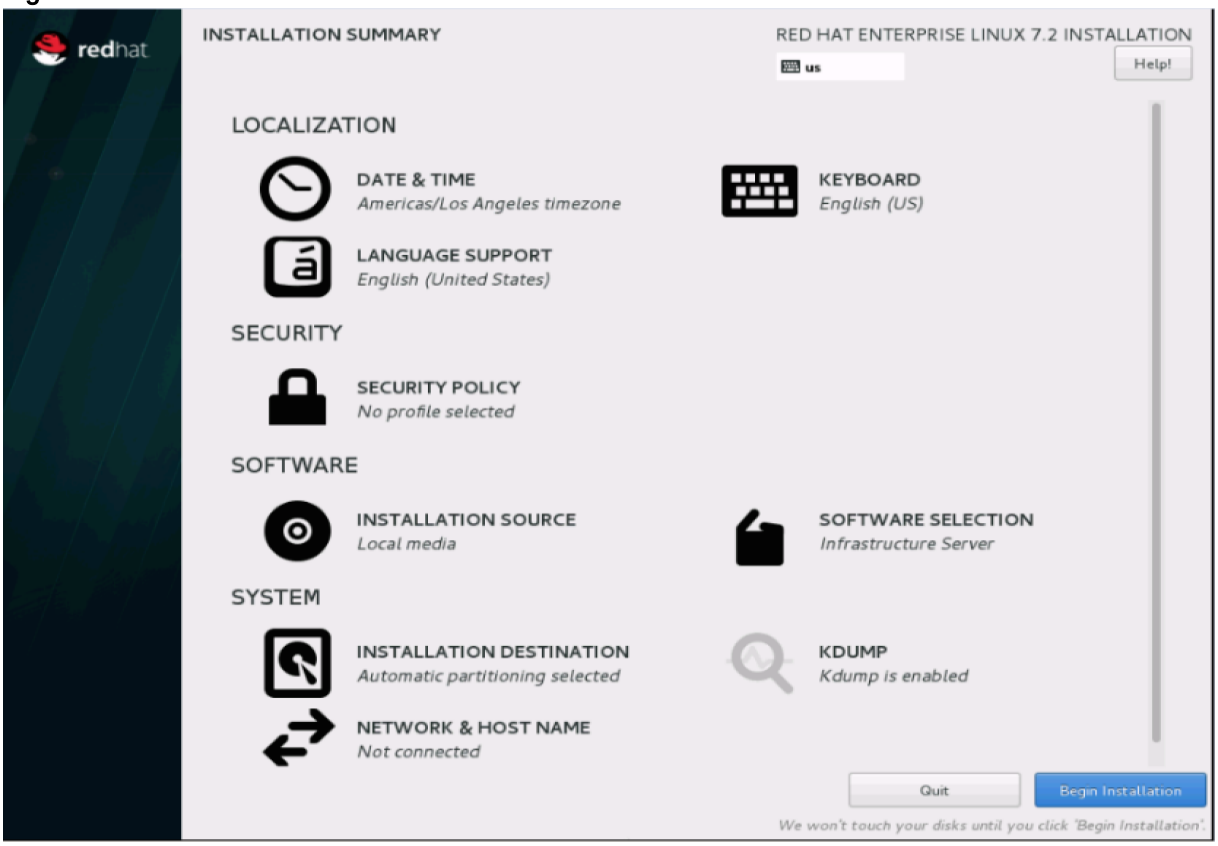
31. Select Infrastructure Server and select the Add-Ons as noted below. Click Done. (Figure 122)

Figure 122 Infrastructure Server



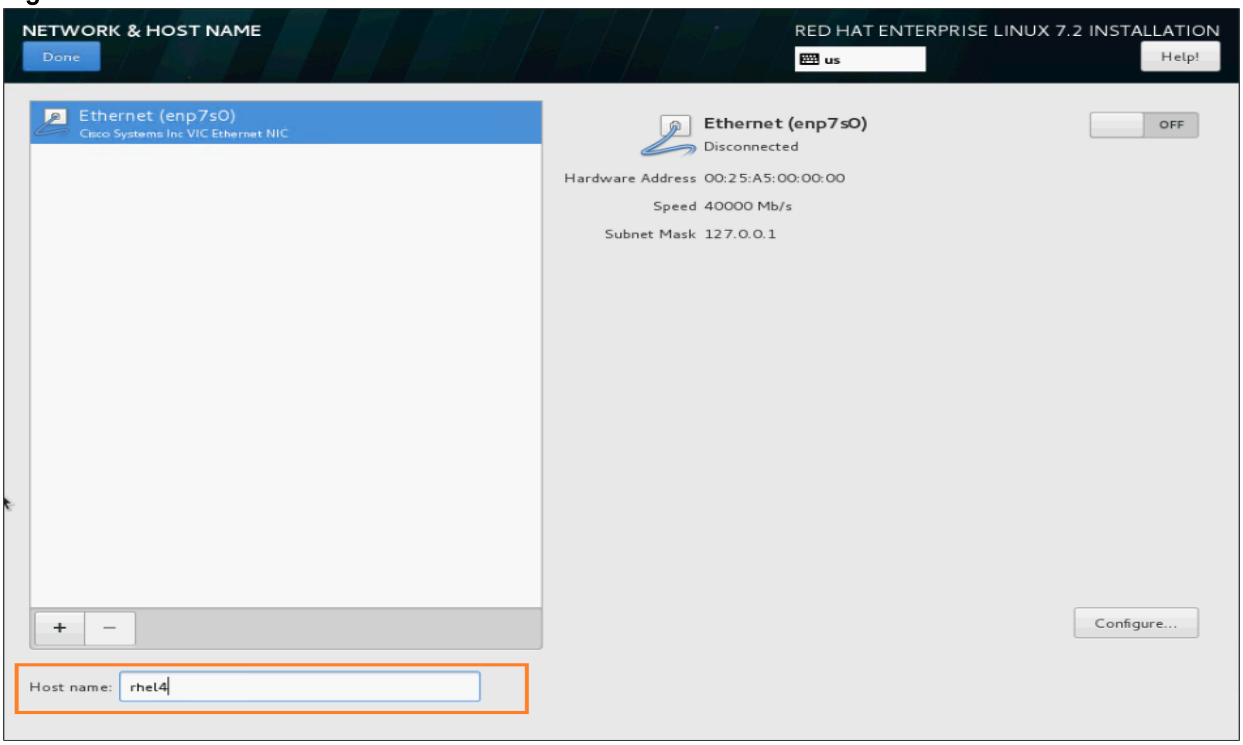
32. Click on Network and Hostname and configure Hostname and Networking for the Host. (Figure 123)

Figure 123 Network and Hostname



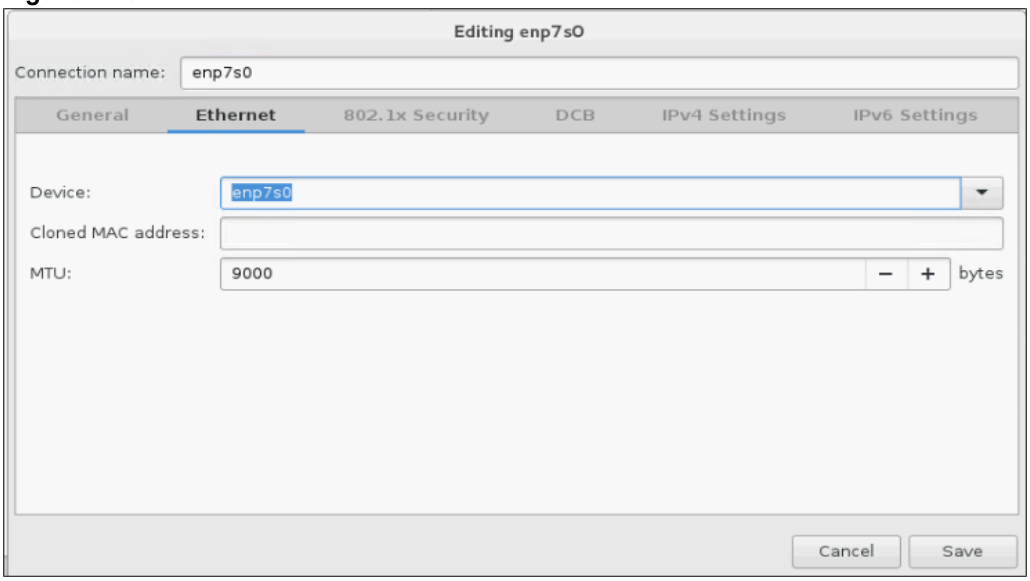
33. Type in the hostname as shown below. (Figure 124)

Figure 124 Add the Host Name



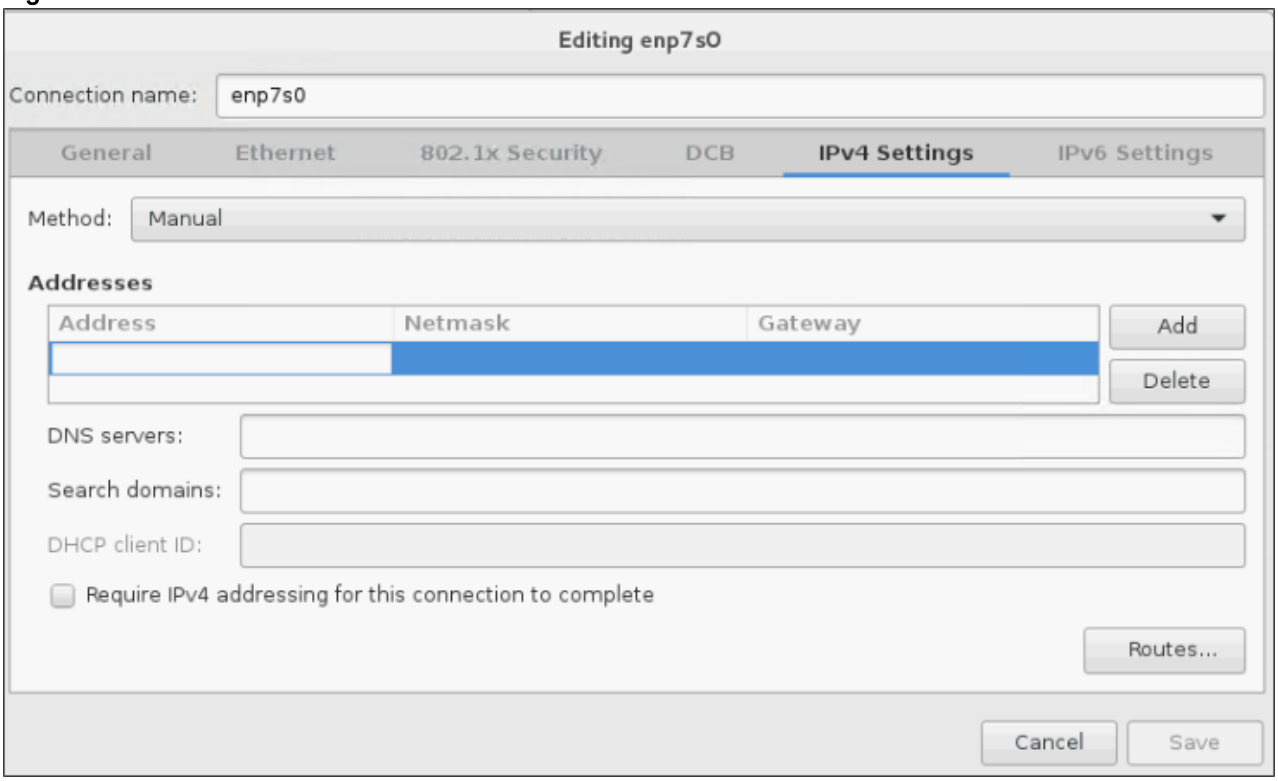
34. Click on Configure to open the Network Connectivity window. Click on Ethernet. (Figure 125)

Figure 125 Add Ethernet



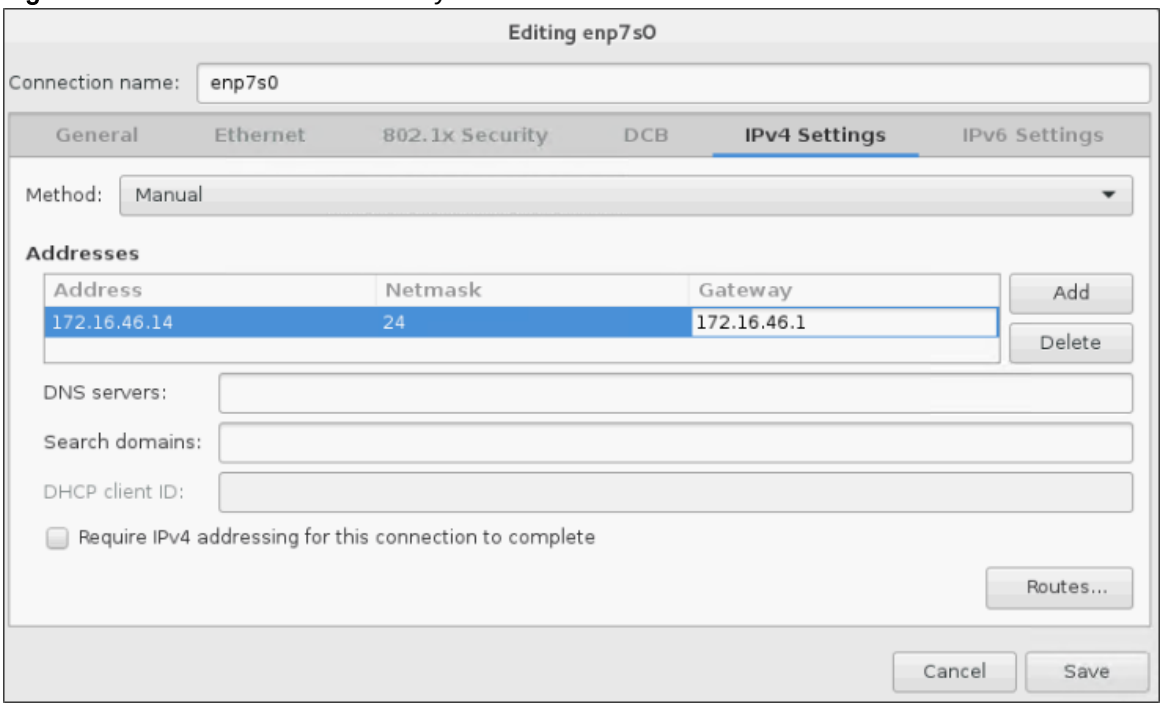
35. Click on IPv4 Settings and change the Method to Manual and click Add to enter the IP Address, Netmask and Gateway details. (Figure 126)

Figure 126 Add IP Details



36. Enter the desired IP address, Netmask and Gateway and click Save. (Figure 127)

Figure 127 Manual IP Address Entry



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

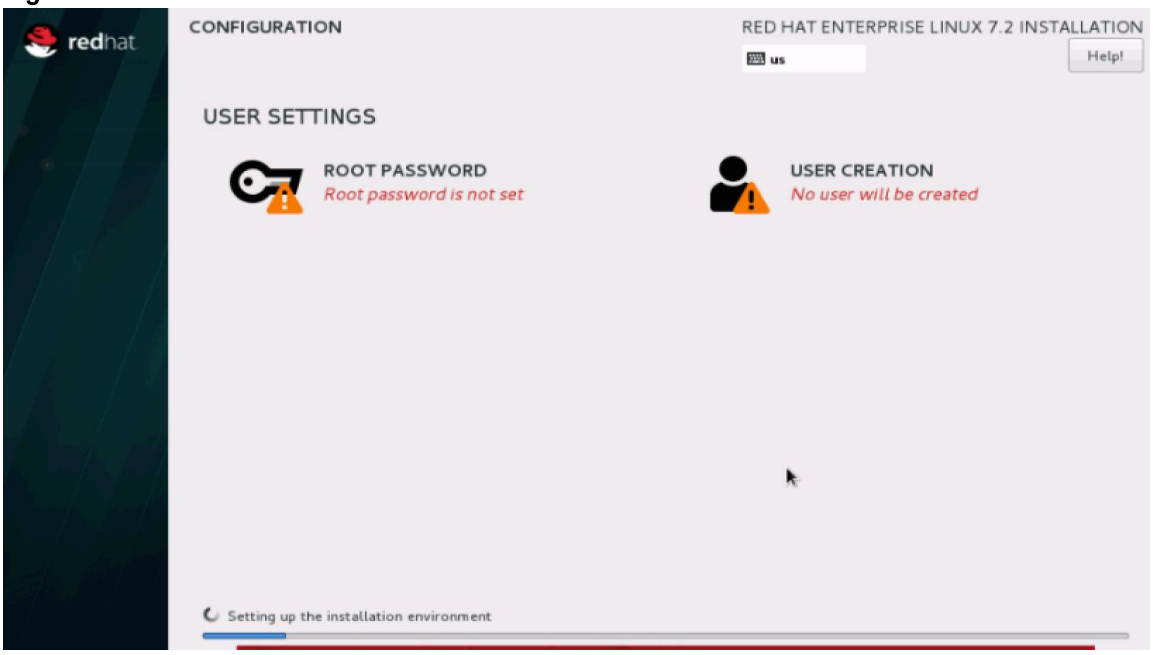
menu.

38. Click Begin Installation in the main menu. (Figure 128)

Figure 128 Begin Installation

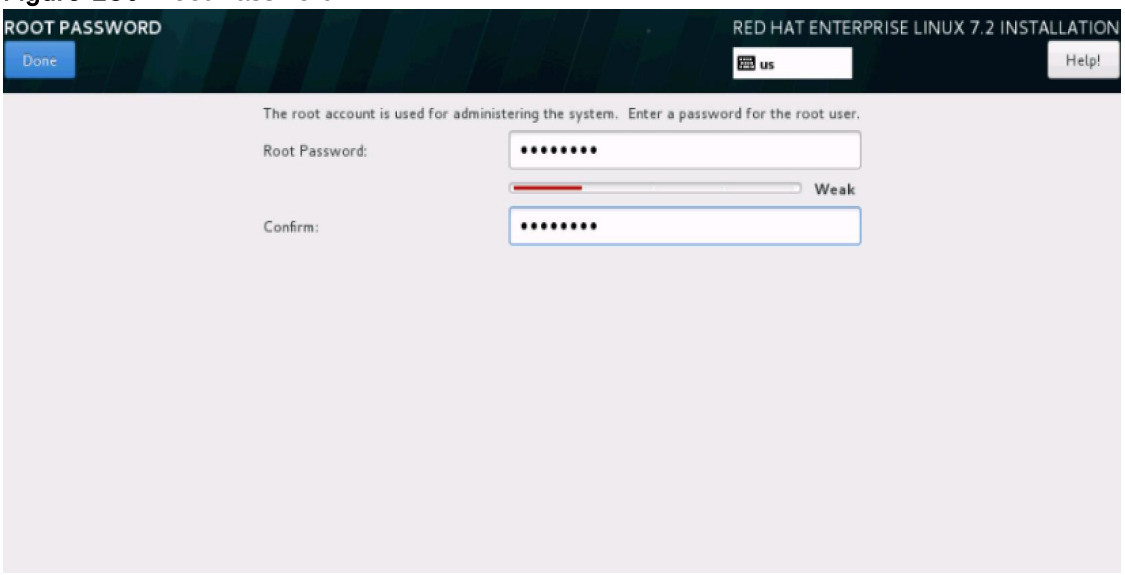
39. Select Root Password in the User Settings. (Figure 129)

Figure 129 Root Password



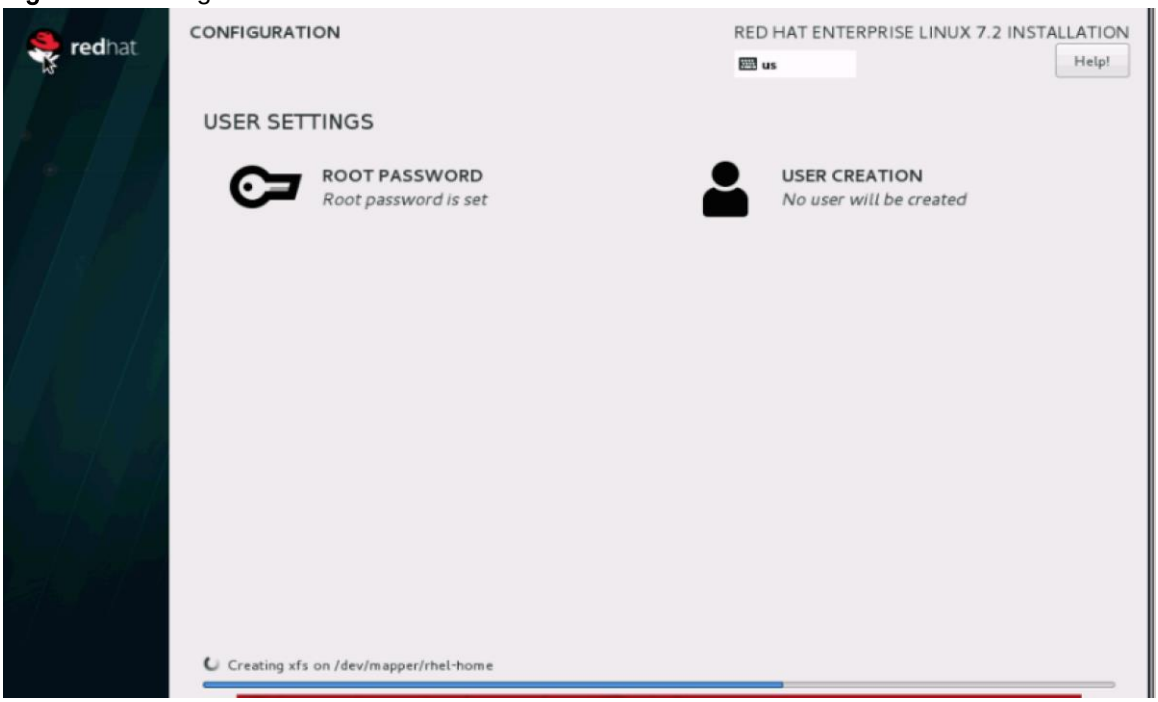
40. Enter the Root Password and click done. (Figure 130)

Figure 130 Root Password



The Installation Progress window displays the process. (Figure 131)

Figure 131 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 9.

Table 9 **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

Hostname	eth0
rhel7	172.16.46.17
rhel8	172.16.46.18
rhel9	172.16.46.19
rhel10	172.16.46.20
rhel11	172.16.46.21
rhel12	172.16.46.22
rhel13	172.16.46.23
rhel14	172.16.46.24
rhel15	172.16.46.25
rhel16	172.16.46.26
rhel17	172.16.46.27
rhel18	172.16.46.28
rhel19	172.16.46.29

Post OS Install Configuration

Choose one of the nodes of the cluster or a separate node as the Admin Node for management such as 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).

2. `#ssh 172.16.46.11`

3. 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 +      |
|       = = .     |
|        = .      |
|       .         |
|                 |
+-----------------+
```

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

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

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

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

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

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

```
vi ~/.ssh/config

ServerAliveInterval 99

StrictHostKeyChecking no
```

10. 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 \
localhost4.localdomain4

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

172.16.46.11    rhel1

172.16.46.12    rhel2

172.16.46.13    rhel3

172.16.46.14    rhel4

172.16.46.15    rhel5

172.16.46.16    rhel6

172.16.46.17    rhel7

172.16.46.18    rhel8

172.16.46.19    rhel9
```

```
172.16.46.20    rhel10
172.16.46.21    rhel11
172.16.46.22    rhel12
172.16.46.23    rhel13
172.16.46.24    rhel14
172.16.46.25    rhel15
172.16.46.26    rhel16
172.16.46.27    rhel17
172.16.46.28    rhel18
172.16.46.29    rhel19
```

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

To create a repository using 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.

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

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

4. Alternatively, if access to a Red Hat ISO Image is available, copy the ISO file to rhel1.

5. And login back to rhel1 and create the mount directory.

```
#scp rhel-server-7.2-x86_64-dvd.iso rhel1:/root/

#mkdir -p /mnt/rheliso

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

6. 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
```

7. Now on rhel1 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://10.4.1.31/rhelrepo
gpgcheck=0
enabled=1
```

8. 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.

9. 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

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: langpacks, product-id, search-disabled-repos, subscription-
              : manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
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:
```

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 ClusterShell (clush) and install it on rhel1. ClusterShell is available from EPEL (Extra Packages for Enterprise Linux) repository.

```
#wget
http://rpm.pbone.net/index.php3/stat/4/idpl/31529309/dir/redhat_el_7/com/c
lustershell-1.7-1.el7.noarch.rpm.html

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

2. Login to rhel1 and install cluster shell.

3. `#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 ~]#
```

4. 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.

5. For 64 node cluster as in our CVD, set groups file as follows,

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

```
Complete!
[root@rhel1 ~]# vi /etc/clustershell/groups.d/local.cfg
[root@rhel1 ~]#
```

`all: rhel[1-19]`

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

Note: ClusterShell will not work if not ssh to the machine earlier (as it must be in the known_hosts file), for instance, as in the case below for rhel<host>.

```
[root@rhel1 ~]# ssh rhel2
The authenticity of host 'rhel2 (10.4.1.32)' can't be established.
ECDSA key fingerprint is 12:90:ec:f5:a2:45:23:5c:d5:30:66:d7:87:ee:1f:55.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'rhel2' (ECDSA) to the list of known hosts.
```

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
```

```
[root@rhel1 ~]# cat /etc/httpd/conf/httpd.conf | grep ServerName
# ServerName gives the name and port that the server uses to identify itself.
ServerName      10.4.1.31:80
[root@rhel1 ~]#
```

3. Start httpd

```
#service httpd start

#chkconfig httpd on
```

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.

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

4. Purge the yum caches after this

```
#clush -a -B yum clean all
#clush -a -B yum repolist
```

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

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

Configure 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 if setting up dnsmasq on the Admin node. If not 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 the 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` to 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

7. 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`.

8. 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).

9. 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"
```

10. 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
```

11. 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 the 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 the 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
```

3. `keys /etc/ntp/keys`

4. 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
```

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

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

6. Run the following script 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"
```

7. 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. One of the following commands should be used to confirm that the service is properly configured:

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

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

```
[root@rhel1 ~]# clush -B -a rsyslogd -v
---------------
rhel [1-64] (64)
---------------
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. This value should be set 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.

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.

SELinux can be disabled by editing `/etc/selinux/config` and changing the `SELINUX` line to `SELINUX=disabled`. The following command will disable `SELINUX` on all nodes:

```
#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.

Reboot the machine, if needed for SELinux to be disabled incase it does not take effect. It can checked using

```
#clush -a -b sestatus
```

Set TCP Retries

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

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

Disabling the Linux Firewall

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

```
#clush -a -b " firewall-cmd --zone=public --add-port=80/tcp --permanent"

#clush -a -b "firewall-cmd --reload"

#clush –a –b "systemctl stop firewalld.service"

#clush –a -b  "systemctl disable firewalld.service"
```

Disable Swapping

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

```
#clush -a -b "echo never > /sys/kernel/mm/transparent_hugepage/enabled"

#clush -a -b "echo never > /sys/kernel/mm/transparent_hugepage/defrag"
```

1. The commands above must be run for every reboot, so copy these commands to /etc/rc.local so they are executed automatically for every reboot.

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

```
#rm –f /root/thp_disable

#echo "echo never > /sys/kernel/mm/transparent_hugepage/enabled" >>

/root/thp_disable

#echo "echo never > /sys/kernel/mm/transparent_hugepage/defrag " >>

/root/thp_disable
```

3. Copy file to each node:

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

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

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

Disable IPv6 Defaults

1. Disable IPv6 as the addresses used are IPv4:

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

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

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

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

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

Configuring Data Drives on Name Node and Other Management Nodes

This section describes steps to configure non-OS disk drives as RAID1 using StorCli commands as described below. All the drives are going to be part of a single RAID1 volume. This volume can be used for staging any client data to be loaded to HDFS. This volume won't be used for HDFS data.

1. From the website download storcli:
   http://www.lsi.com/downloads/Public/RAID%20Controllers/RAID%20Controllers%20Common%20Files/1.14.12_StorCLI.zip

2. Extract the zip file and copy storcli-1.14.12-1.noarch.rpm from the Linux directory.

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

```
#scp storcli-1.14.12-1.noarch.rpm rhel1:/root/
```

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

```
#clush -a -b -c /root/storcli-1.14.12-1.noarch.rpm --dest=/root/
```

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

```
#clush -a -b "rpm -ivh storcli-1.14.12-1.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
install.log  libstorelibir-2.so  libstorelibir-2.so.14.07-0  storcli64
[root@rhel1 storcli]# cp storcli64 /root/
```

7. Copy storcli64 to all the nodes using the following commands:

```
#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,$1:5,$1:6,$1:7,$1:8,$1:9,$1:10,$1:11,$1:12,$1:13,$1
:14,$1:15,$1:16,$1:17,$1:18,$1:19,$1:20,$1:21,$1:22,$1:23,$1:24] wb ra
nocachedbadbbu strpsz1024 -a0
```

The script above requires enclosure ID as a parameter.

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 the 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
```

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

Configuring Data Drives on Data Nodes

This section describes steps to configure non-OS disk drives as individual RAID0 volumes using StorCli commands as described below. These volumes are going to be used for HDFS Data.

1. Issue the following command from the admin node to create the virtual drives with individual RAID 0 configurations on all the data nodes.

```
#clush –w rhel[3-64] -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 the 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 Datanode. OS boot partition is going to be skipped. All drives are going to be 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
  #disks_count=`lsblk -id | grep sd | wc -l`
  #if [ $disks_count -eq 24 ]; then
  # echo "Found 24 disks"
  #else
  # echo "Found $disks_count disks. Expecting 24. Exiting.."
  # exit 1
  #fi
  [[ "-x" == "${1}" ]] && set -x && set -v && shift 1
  count=1
  for X in /sys/class/scsi_host/host?/scan
  do
  echo '- - -' > ${X}
  done
  for X in /dev/sd?
```

```
do

echo "========"

echo $X

echo "========"

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

]]

then

echo "$X bootable - skipping."

continue

else

Y=${X##*/}1

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

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

(( $? )) && continue

#Identify UUID

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

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

(( $? )) && continue

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

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

(( $? )) && continue

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

((count++))

fi

done
```

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

```
#chmod 755 /root/driveconf.sh
#clush –a –B –c /root/driveconf.sh
```

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

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

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

```
#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 partition is to be deleted and run `fdisk` to delete 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 Integrated Infrastructure for Big Data and
Analytics \ Cluster Verification === ${NC}"

echo ""

echo ""

echo -e "${green} ==== System Information  ==== ${NC}"

echo ""

echo ""

echo -e "${green}System ${NC}"

clush -a -B " `which dmidecode` |grep -A2 '^System Information'"

echo ""

echo ""

echo -e "${green}BIOS ${NC}"

clush -a -B " `which dmidecode` | grep -A3 '^BIOS I'"

echo ""

echo ""

echo -e "${green}Memory ${NC}"

clush -a -B "cat /proc/meminfo | grep -i ^memt | uniq"

echo ""

echo ""

echo -e "${green}Number of Dimms ${NC}"

clush -a -B "echo -n 'DIMM slots: ';  dmidecode |grep -c \
'^[[:space:]]*Locator:'"

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

clush -a -B " dmidecode | awk '/Memory Device$/,/^$/ {print}' |\grep -e
'^Mem' -e Size: -e Speed: -e Part | sort -u | grep -v -e 'NO \ DIMM' -e 'No
Module Installed' -e Unknown"

echo ""

echo ""

# probe for cpu info #

echo -e "${green}CPU ${NC}"

clush -a -B "grep '^model name' /proc/cpuinfo | sort -u"
```

```
echo ""

clush -a -B "`which lscpu` | grep -v -e op-mode -e ^Vendor -e family -e\
Model: -e Stepping: -e BogoMIPS -e Virtual -e ^Byte -e '^NUMA node(s)'"

echo ""

echo ""

# probe for nic info #

echo -e "${green}NIC ${NC}"

clush -a -B "ls /sys/class/net | grep ^enp | \xargs -l  `which ethtool` |
grep -e ^Settings -e Speed"

echo ""

clush -a -B "`which lspci` | grep -i ether"

echo ""

echo ""

# probe for disk info #

echo -e "${green}Storage ${NC}"

clush -a -B "echo 'Storage Controller: '; `which lspci` | grep -i -e \ raid -
e storage -e lsi"

echo ""

clush -a -B "dmesg | grep -i raid | grep -i scsi"

echo ""

clush -a -B "lsblk -id | awk '{print \$1,\$4}'|sort | nl"

echo ""

echo ""


echo -e "${green} ================ Software  ====================== ${NC}"

echo ""

echo ""

echo -e "${green}Linux Release ${NC}"

clush -a -B "cat /etc/*release | uniq"

echo ""

echo ""

echo -e "${green}Linux Version ${NC}"

clush -a -B "uname -srvm | fmt"
```

```
echo ""

echo ""

echo -e "${green}Date ${NC}"

clush -a -B date

echo ""

echo ""

echo -e "${green}NTP Status ${NC}"

clush -a -B "ntpstat 2>&1 | head -1"

echo ""

echo ""

echo -e "${green}SELINUX ${NC}"

clush -a -B "echo -n 'SElinux status: '; grep ^SELINUX= \/etc/selinux/config
2>&1"

echo ""

echo ""

clush -a -B "echo -n 'CPUspeed Service: ';  cpupower frequency-info \ status
2>&1"

#clush -a -B "echo -n 'CPUspeed Service: '; `which chkconfig` --list \
cpuspeed 2>&1"

echo ""

echo ""

echo -e "${green}Java Version${NC}"

clush -a -B 'java -version 2>&1; echo JAVA_HOME is ${JAVA_HOME:-Not \ De-
fined!}'

echo ""

echo ""

echo -e "${green}Hostname LoOKup${NC}"

clush -a -B " ip addr show"

echo ""

echo ""

echo -e "${green}Open File Limit${NC}"

clush -a -B 'echo -n "Open file limit(should be >32K): "; ulimit -n'

# MapR related RPMs
```

```
clush -a -B 'rpm -qa | grep -i nfs |sort'


clush -a -B 'rpm -qa | grep -i nfs |sort'

clush -a -B 'echo Missing RPMs: ; for each in make patch redhat-lsb
irqbalance  syslinux hdparm sdparm dmidecode nc; do rpm -q $each | grep "is
not installed";  done'

clush -a -B "ls -d /opt/mapr/* | head"

# mapr login for hadoop

clush -a -B 'echo "mapr login for Hadoop "; getent passwd mapr'

clush -a -B 'echo "Root login "; getent passwd root'

exit
```

NOTE: Please install pciutils if not installed, to run the script correctly.

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 IBM BigInsights

IBM BigInsights provides the power of open source, IBM innovations and rich developer tools – and puts the full range of analytics for Hadoop, Spark, and SQL into the hands of big data analytics teams. IBM Open Platform (IOP) with Apache Spark and Apache Hadoop is IBM's big data platform.

Pre-Requisites for IOP Installation

The IBM Open Platform with Apache Spark and Apache Hadoop uses the repository-based Ambari installer.

To download the IBM Open Platform rpm package file go to:
http://www.ibm.com/support/docview.wss?uid=swg24042361

An IBM ID is required for the download.

Additional components from BigInsights like BigSQL, BigSheets etc. can also be downloaded at this site.

1. Log on to the admin server rhel1 and run the following command to install the nc package:

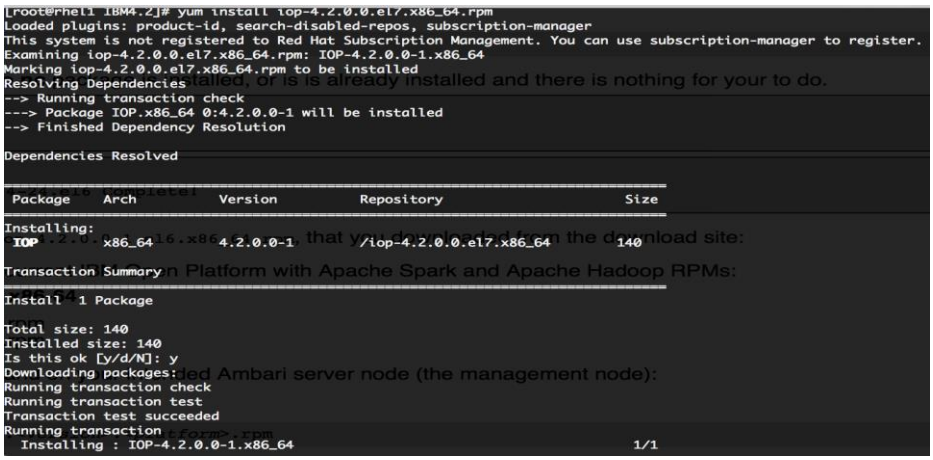
```
#clush –a –b yum install -y nc
```

2. Copy the downloaded rpm for IOP from the download site to the admin node:

```
#scp iop-4.2.0.0-1.el7.x86_64.rpm rhel1:/root
```

3. Install the rpm file on the admin node:

```
# yum install iop-4.2.0.0-1.el7.x86_64.rpm
```



4. From a host connected to the internet, download the required IBM repository files and transfer to the admin node:

```
#mkdir -p /tmp/IBM
```

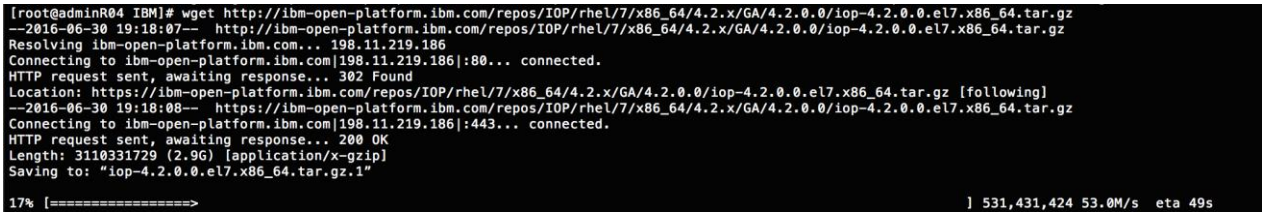
```
#cd /tmp/IBM/
```

5. To download the Ambari repo go to:

#wget http://ibm-open-platform.ibm.com/repos/Ambari/rhel/7/x86_64/2.2.x/GA/2.2.0/ambari-2.2.0.el7.x86_64.tar.gz

6. Download the IOP repo at:

#wget http://ibm-open-platform.ibm.com/repos/IOP/rhel/7/x86_64/4.2.x/GA/4.2.0.0/iop-4.2.0.0.el7.x86_64.tar.gz



7. Download the IOP-UTILS repo at:

#wget http://ibm-open-platform.ibm.com/repos/IOP-UTILS/rhel/7/x86_64/1.2/iop-utils-1.2.0.0.el7.x86_64.tar.gz

8. Copy the repository directory to the admin node:

9. `#scp -r /tmp/IBM/ rhel1:/var/www/html /repos`

10. Extract the files login to rhel1:

```
#cd /var/www/html/repos

#tar -zxvf ambari-2.2.0.el7.x86_64.tar.gz

#tar -zxvf iop-4.2.0.0.el7.x86_64.tar.gz
```

```
#tar –zxvf iop-utils-1.2.0.0.el7.x86_64.tar.gz
```

11. Update the ambari.repo file to use the local repository set up above.

```
#vi /etc/yum.repos.d/ambari.repo

[BI_AMBARI-2.2.0]

name=ambari-2.2.0

baseurl=http:// 172.16.46.11/repos/Ambari/RHEL7/x86_64/2.2.0/

enabled=1

gpgcheck=0
```

12. Update the baseurl with the webserver's address where the local repo is available as shown above.

13. Clean the yum cache on all nodes so that the right packages from the remote repository are seen by the local yum.

```
# clush –a –b yum clean all
```

```
rhel1
----------------
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.
Cleaning repos: BI_AMBARI-2.2.0 rhel7.2
Cleaning up everything
```

14. Install the Ambari server on the admin management node, using the following command:

```
#yum install ambari-server
```

```
[root@rhel1 ~]# yum install ambari-server
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.
BI_AMBARI-2.2.0                                                                                    | 2.9 kB  00:00:00
rhel7.2                                                                                            | 2.9 kB  00:00:00
(1/2): BI_AMBARI-2.2.0/primary_db                                                                  | 9.8 kB  00:00:00
(2/2): rhel7.2/primary_db                                                                          | 3.6 MB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package ambari-server.x86_64 0:2.2.0_IBM-000000 will be installed
--> Processing Dependency: postgresql-server >= 8.1 for package: ambari-server-2.2.0_IBM-000000.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: postgresql-server-9.2.13-1.el7_1.x86_64
--> Processing Dependency: postgresql(x86-64) = 9.2.13-1.el7_1 for package: postgresql-server-9.2.13-1.el7_1.x86_64
--> Processing Dependency: libpq.so.5()(64bit) for package: postgresql-server-9.2.13-1.el7_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                 2.2.0_IBM-000000                 BI_AMBARI-2.2.0                410 M
Installing for dependencies:
 postgresql                       x86_64                 9.2.13-1.el7_1                   rhel7.2                        3.0 M
 postgresql-libs                  x86_64                 9.2.13-1.el7_1                   rhel7.2                        230 k
 postgresql-server                x86_64                 9.2.13-1.el7_1                   rhel7.2                        3.8 M

Transaction Summary
================================================================================================================================
Install  1 Package (+3 Dependent packages)

Total download size: 417 M
Installed size: 475 M
Is this ok [y/d/N]: y
Downloading packages:
ambari-server-2.2.0_IBM-000000.x86_64.rpm                                                          | 410 MB  00:00:04
--------------------------------------------------------------------------------------------------------------------------------
Total                                                                                   87 MB/s | 417 MB  00:00:04
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : postgresql-libs-9.2.13-1.el7_1.x86_64                                                                1/4
  Installing : postgresql-9.2.13-1.el7_1.x86_64                                                                     2/4
```

15. Update the value for the baseurl in the file below:

```
/var/lib/ambari-server/resources/stacks/BigInsights/4.2/repos/repoinfo.xml
```

with the urls for the local repository as shown below:

```
#vi /var/lib/ambari-
server/resources/stacks/BigInsights/4.2/repos/repoinfo.xml

<reposinfo>

  <mainrepoid>IOP-4.2</mainrepoid>

  <os family="redhat7">

    <repo>

<baseurl>http://10.4.1.31/repos/IOP/RHEL7/x86_64/4.2.0.0/</baseurl>

      <repoid>IOP-4.2</repoid>

      <reponame>IOP</reponame>

    </repo>

    <repo>

<baseurl>http://10.4.1.31/repos/IOP-UTILS/rhel/7/x86_64/1.2/</baseurl>

      <repoid>IOP-UTILS-1.2</repoid>

      <reponame>IOP-UTILS</reponame>

    </repo>

  </os>

</reposinfo>
```

16. Edit the file `/etc/ambari-server/conf/ambari.properties` to update the JDK location from the IOP-UTILS repository as shown below:

```
#vi /etc/ambari-server/conf/ambari.properties
openjdk1.8.url=http:// 172.16.46.11/repos/IOP-UTILS/rhel/7/x86_64/1.2/openjdk/jdk-1.8.0.tar.gz
```

Set Up the Ambari server

1. Run the following command and accept the default settings:

```
#ambari-server setup
```

```
[root@rhel1 ~]# ambari-server setup
Using python  /usr/bin/python2
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...
Redirecting to /bin/systemctl status  iptables.service

Checking JDK...
[1] OpenJDK 1.8.0
[2] Custom JDK
================================================================================
Enter choice (1): 1
Downloading JDK from http://   10.4.1.31  /repos/IOP-UTILS/rhel/7/x86_64/1.2/openjdk/jdk-1.8.0.tar.gz to /var/lib/ambari-server/r
jdk-1.8.0.tar.gz... 100% (57.6 MB of 57.6 MB)
Successfully downloaded JDK distribution to /var/lib/ambari-server/resources/jdk-1.8.0.tar.gz
Installing JDK to /usr/jdk64/
Successfully installed JDK to /usr/jdk64/
Completing setup...
Configuring database...
Enter advanced database configuration [y/n] (n)? n
Configuring database...
Default properties detected. Using built-in database.
Configuring ambari database...
Checking PostgreSQL...
Running initdb: This may take upto a minute.
Initializing database ... OK


About to start PostgreSQL
Configuring local database...
Connecting to local database...done.
Configuring PostgreSQL...
Restarting PostgreSQL
Extracting system views...
ambari-admin-2.2.0_IBM_000000-SNAPSHOT.jar
.....
```

2. Start the Ambari server, using the following command:

```
#ambari-server start
```

3. Ensure that the postgresql service, which is used by Ambari, starts automatically on reboot.

4. Run the following command on the admin node:

```
#systemctl start postgresql.service

#systemctl enable postgresql.service
```

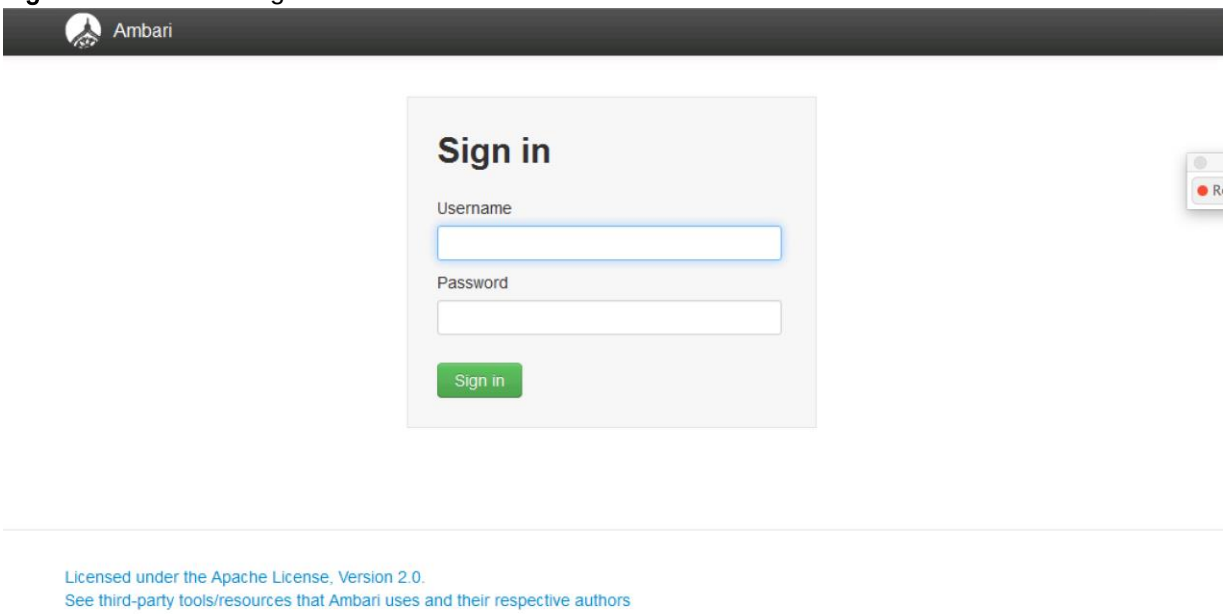
Ambari Server Components Set up

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

2. Point the browser to http:// 172.16.46.11:8080.

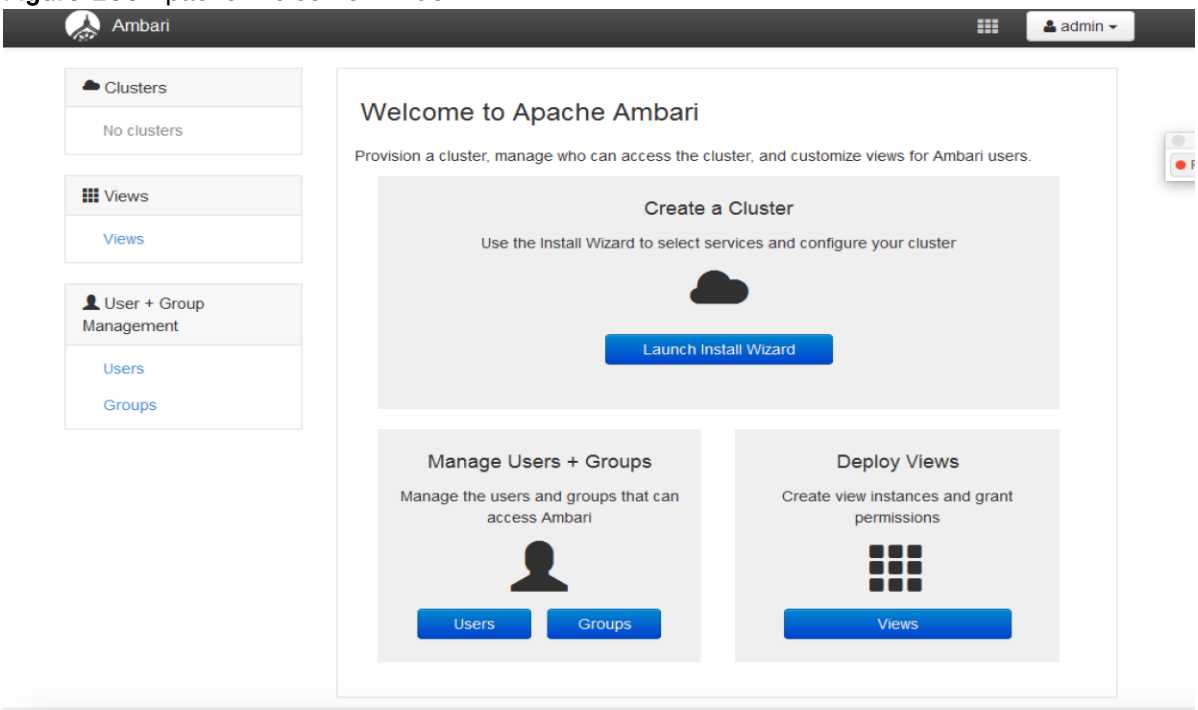
3. Log in to the Ambari Server using the default username/password: admin/admin. This can be changed at a later period of time. (Figure 132)

Figure 132 Ambari Login Window



Once logged in, the "Welcome to Apache Ambari" window appears. (Figure 133)

Figure 133 Apache Welcome Window



Creating a Cluster

To create a cluster, complete the following steps.

1. Click the Launch Install Wizard button.

2. On the Get started page type “Cisco_BI42” as the name for the cluster in the text box. Click the Next button. (Figure 134)

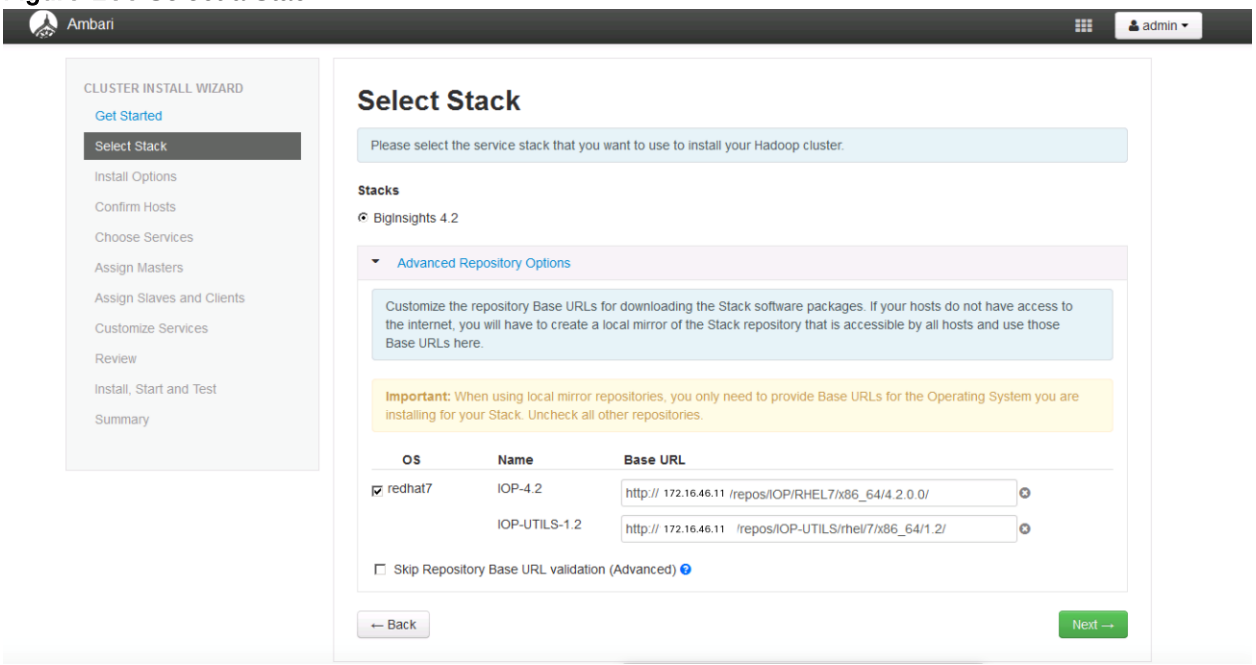
Figure 134 Get Started Window

Select a Stack

1. On the following screen, select the BigInsights 4.2 stack. (Figure 135)

2. Expand “Advanced Repository Options” to check that it is pointed to the local repository on the admin node.

Figure 135 Select a Stack



IOP Installation

In order to build up the cluster, the install wizard needs to know general information about how the cluster has to be set up. This requires providing the Fully Qualified Domain Name (FQDN) of each of the

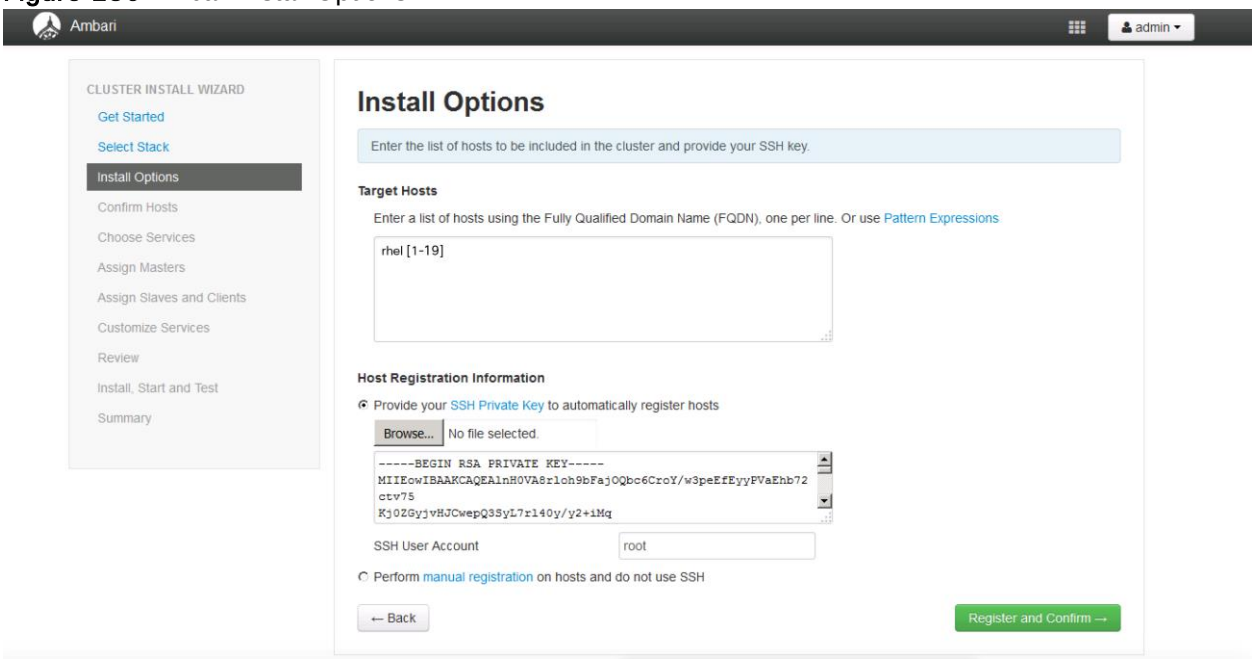
hosts. The wizard also needs to access the private key file that was created in Set Up Password-less SSH. It uses these to locate all the hosts in the system and to access and interact with them securely.

1. Use the Target Hosts text box to enter the list of host names, one per line. Ranges inside brackets can be used to indicate larger sets of hosts. (Figure 136)

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.

Figure 136 Ambari Install Options



4. Click on register and confirm to continue with the installation.

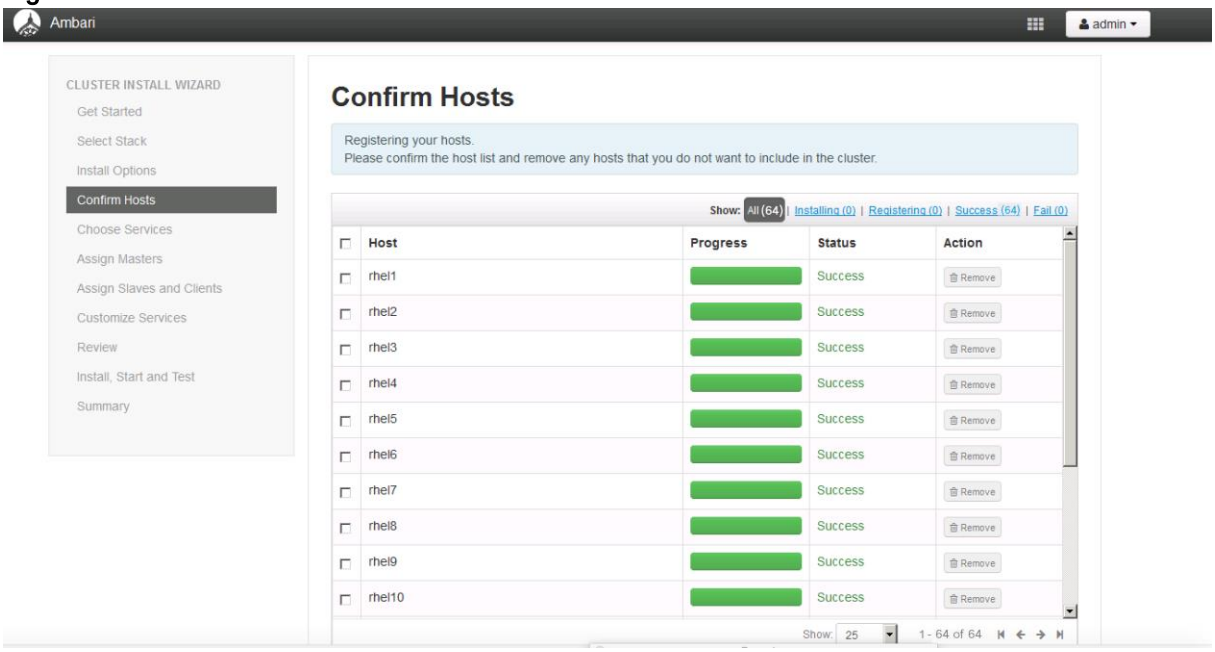
Confirm Hosts

This screen ensures 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. (Figure 137)

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

1. To remove a single host, click the small white Remove button in the Action column. When the lists of hosts are confirmed, click Next.

Figure 137 Confirm Hosts



Host checks (all hosts are in order).

Choose Services

IOP is made up of a number of components.

1. Select All to preselect all items. (Figure 138)
2. When done, click Next.

Figure 138 Choose Services

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. 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 139)

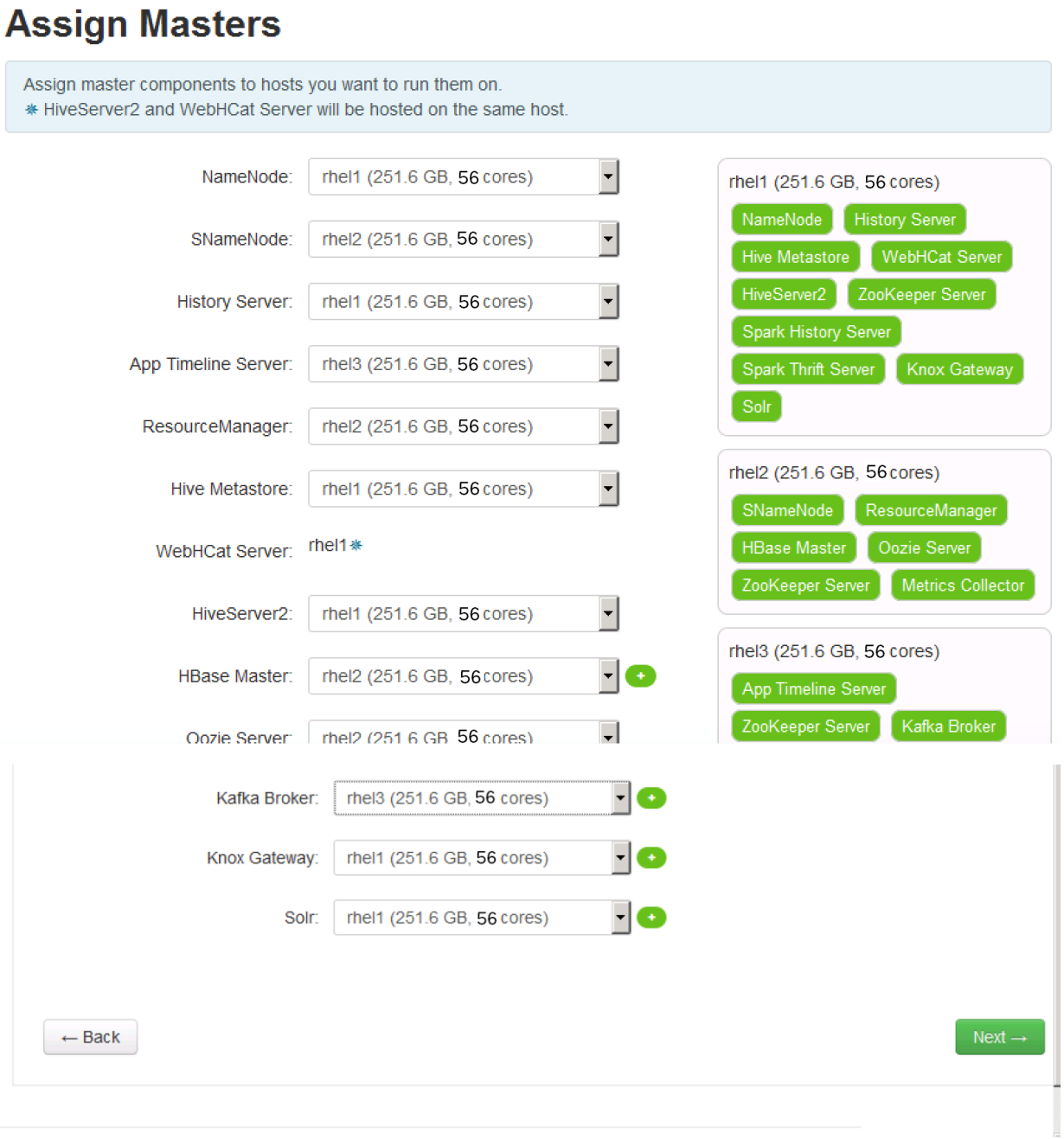
Reconfigure the service assignments to match the values in Table 10.

Table 10 **Service Name and Host**

Service Name	Host
NameNode	rhel1
SNameNode	rhel2
HistoryServer	rhel1
App Timeline Server	rhel2
ResouceManager	rhel2
Hive Metastore	rhel1
WebHCat Server	rhel1
HiveServer2	rhel1
HBase Master	rhel2
Oozie Server	rhel1
Zookeeper	rhel1, rhel2, rhel3
Spark History Server	rhel1
Kafka Broker	rhel3
Knox Gateway	rhel1
Metrics Collector	rhel1

There is one broker by default, add more based on requirements.

Figure 139 Assign Masters



Click the Next button.

Assign Slaves and Clients

The Ambari install wizard attempts to assign the slave components (DataNodes, NFSGateway, NodeManager, RegionServers, Flume, Spark Thrift Server and Client) to appropriate hosts in the cluster. Reconfigure the service assignment to match the example shown in Figure 140 and Table 11 below:

1. Assign DataNode, NodeManager, RegionServer, and Flume on nodes rhel3- rhel64.

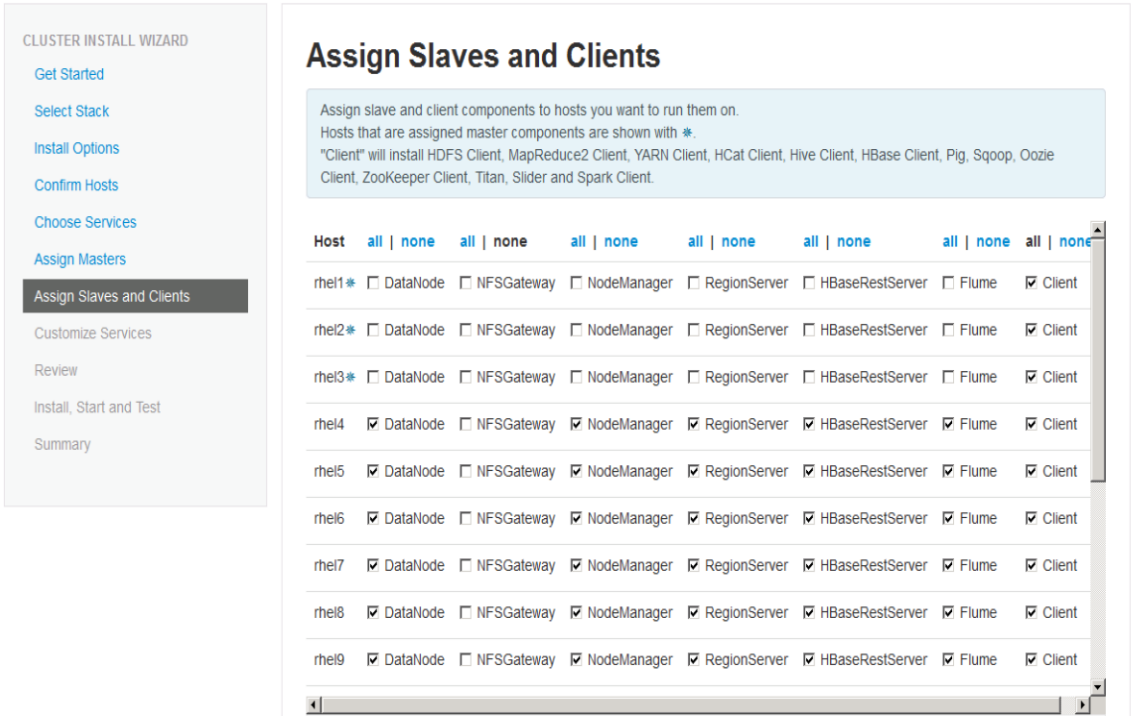
2. Assign Client to all nodes.

3. Click the Next button.

Table 11 **Assign Client Services and Hosts**

Client Service Name	Host
DataNode	rhel4-rhel19
NFSGateway	rhel1
NodeManager	rhel4-rhel19
RegionServer	rhel4-rhel19
Flume	rhel4-rhel19
Spark Thrift Server	rhel1
Client	All nodes, rhel1-rhel19

Figure 140 Assign Slaves and Clients



Customize Services

This section describes 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, memory and service level settings for each component and service level tuning.

HDFS

In Ambari, choose the HDFS Service tab and use the “Search” box on top to filter for the properties mentioned in the table to update their values to those shown in Table 13.

YARN

1. In Ambari, choose the YARN Service from the tab and use the “Search” box on top to filter for the properties mentioned in Table 12 below to update their values.

2. Update the following YARN configurations:

Table 12 **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

HDFS JVM Settings

1. Update the HDFS configurations in Ambari as shown in Table 13.

Table 13 **HDFS Values**

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

MapReduce2

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

2. Update the following MapReduce 2 configurations:

Table 14 **MapReduce 2 Values**

Property Name	Value
Default virtual memory for a job's map-task	4096
Default virtual memory for a job's reduce-task	8192
Sort allocation memory (mapreduce.task.sort.io.mb)	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

Hive

Choose Hive Service from the tab (Figure 141) and select the Advanced tab and make the changes below:

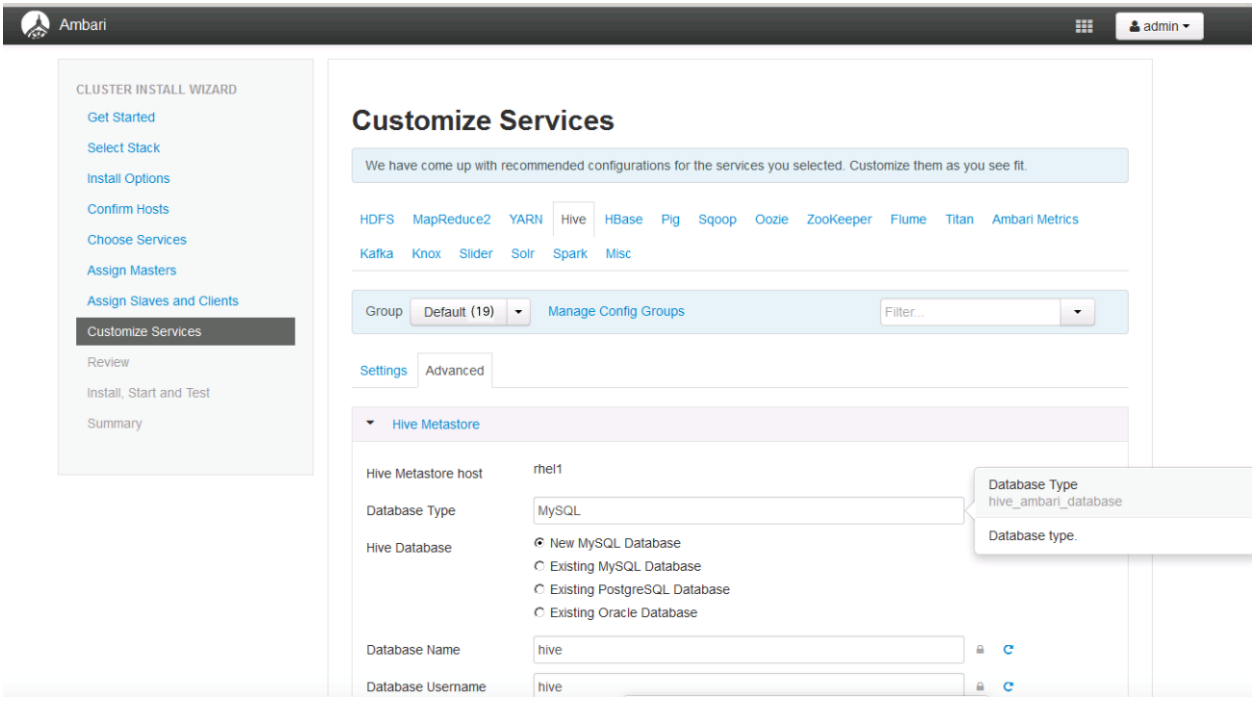
1. Select New MySQL Database

   Database Name: hive

   Database Username: hive

2. Enter the Hive database password as per organizational policy.

Figure 141 Customize Hive Services



HBase

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

Update the following HBASE configurations:

Table 15 **HBase Values**

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

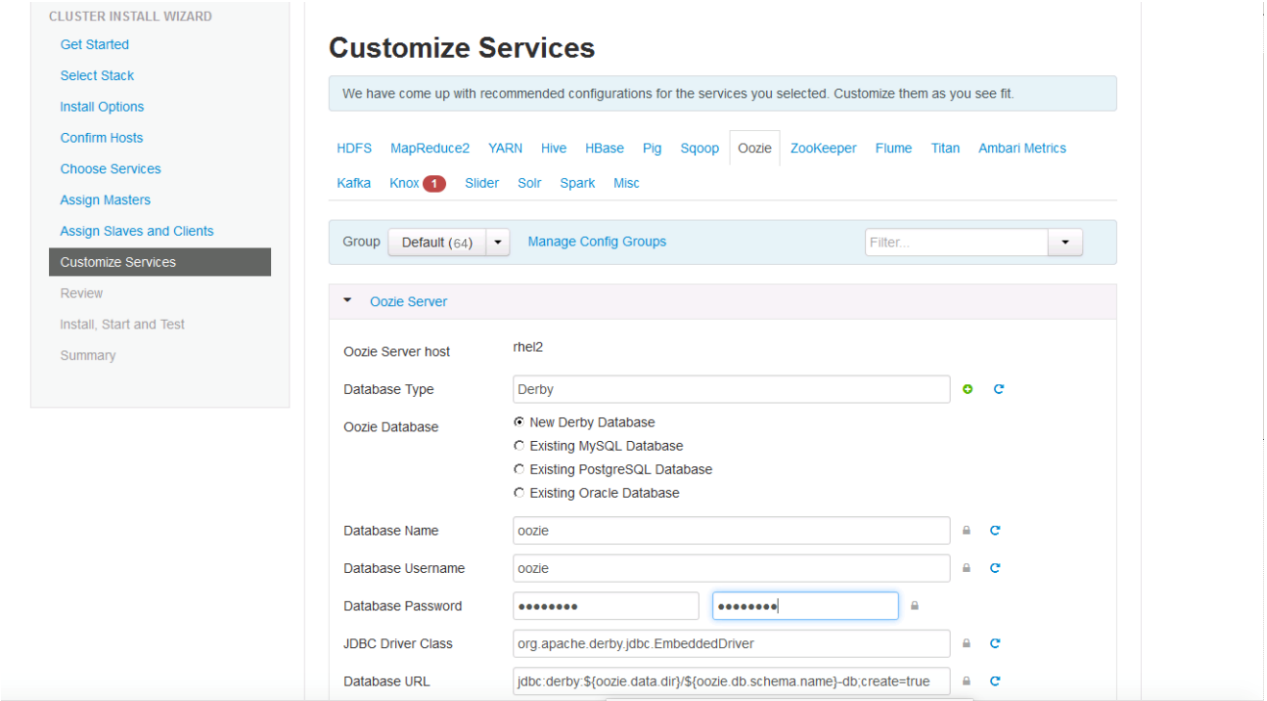
Note: If HBase is not running, keep the default value of 1024 for the Java Heap size for HBase RegionServers and HBase Master.

Oozie

Similarly, under the Oozie tab (Figure 142), change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

1. Select New Derby Database
2. Name the database oozie
3. Name the username oozie

Figure 142 Customize Oozie Services

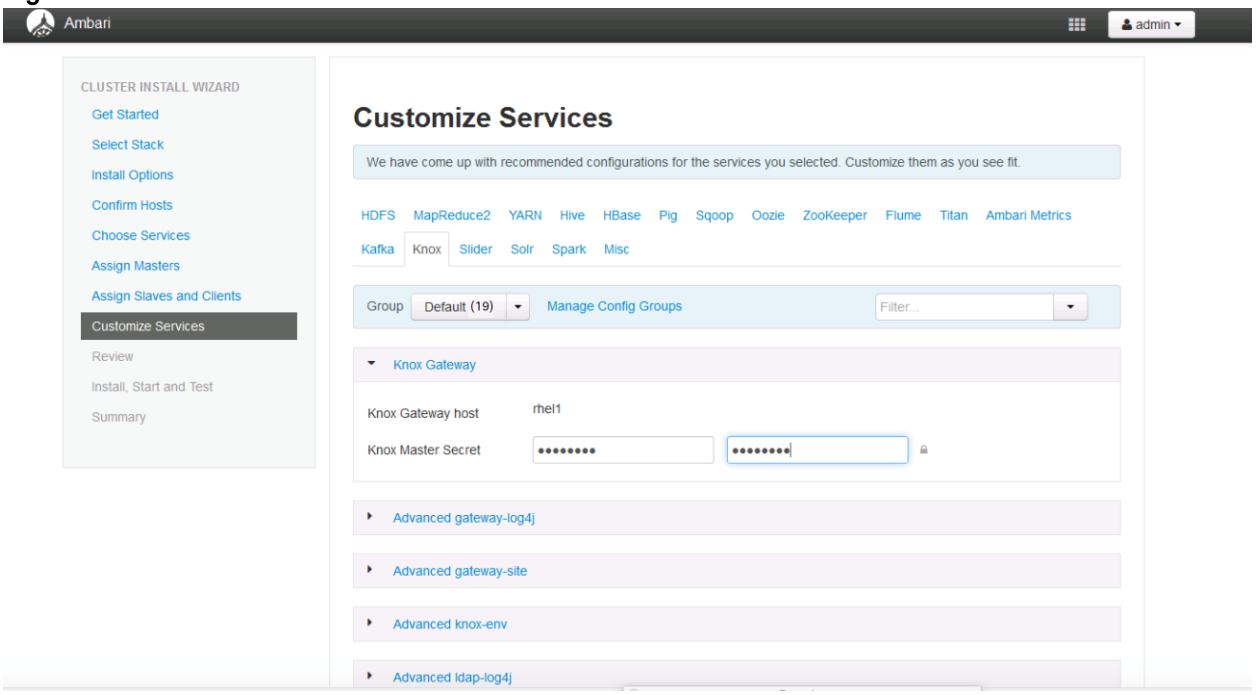


Knox

Choose Knox Service from the tab (Figure 143) and expand the Knox gateway tab and make the changes below:

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

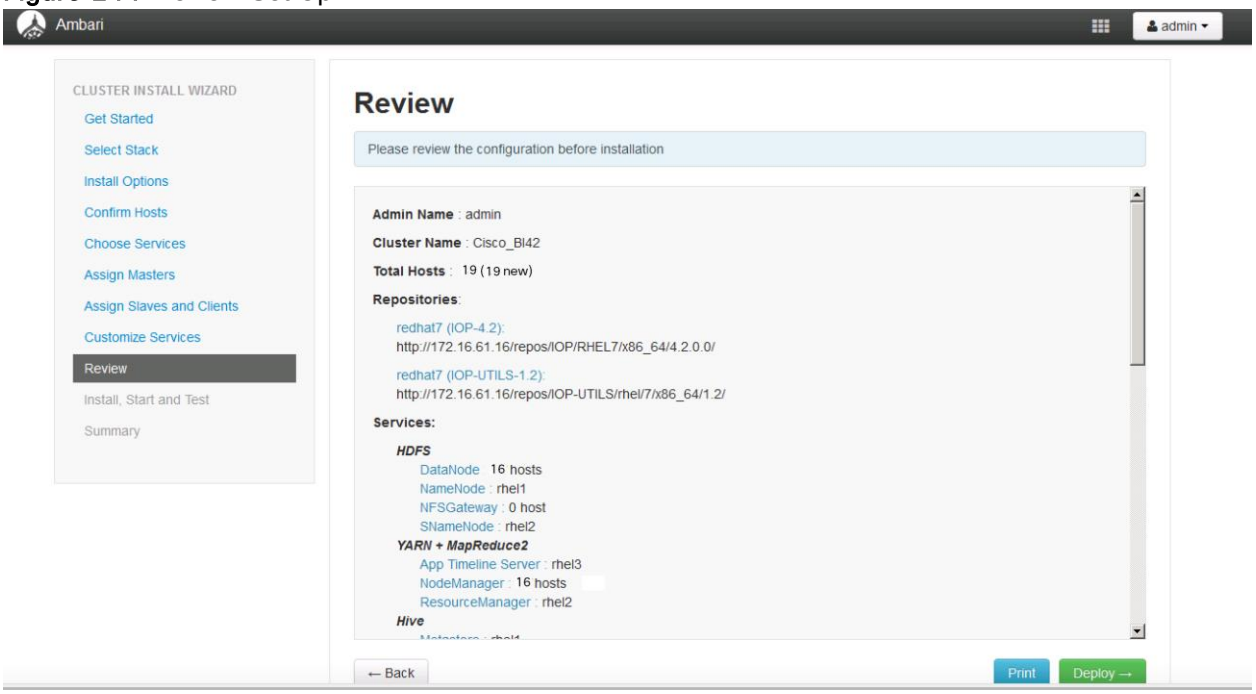
Figure 143 Customize Knox Services



Review

The assignments that have been made are displayed. Check to ensure everything is correct before clicking on Deploy button. If any changes are to be made, use the left navigation bar to return to the appropriate screen. (Figure 144)

Figure 144 Review Set Up



The progress of the install is shown on the screen. 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.

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

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

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

4. To see a larger version of the log contents, click the Open icon. To copy the contents to the clip-board, use the Copy icon.

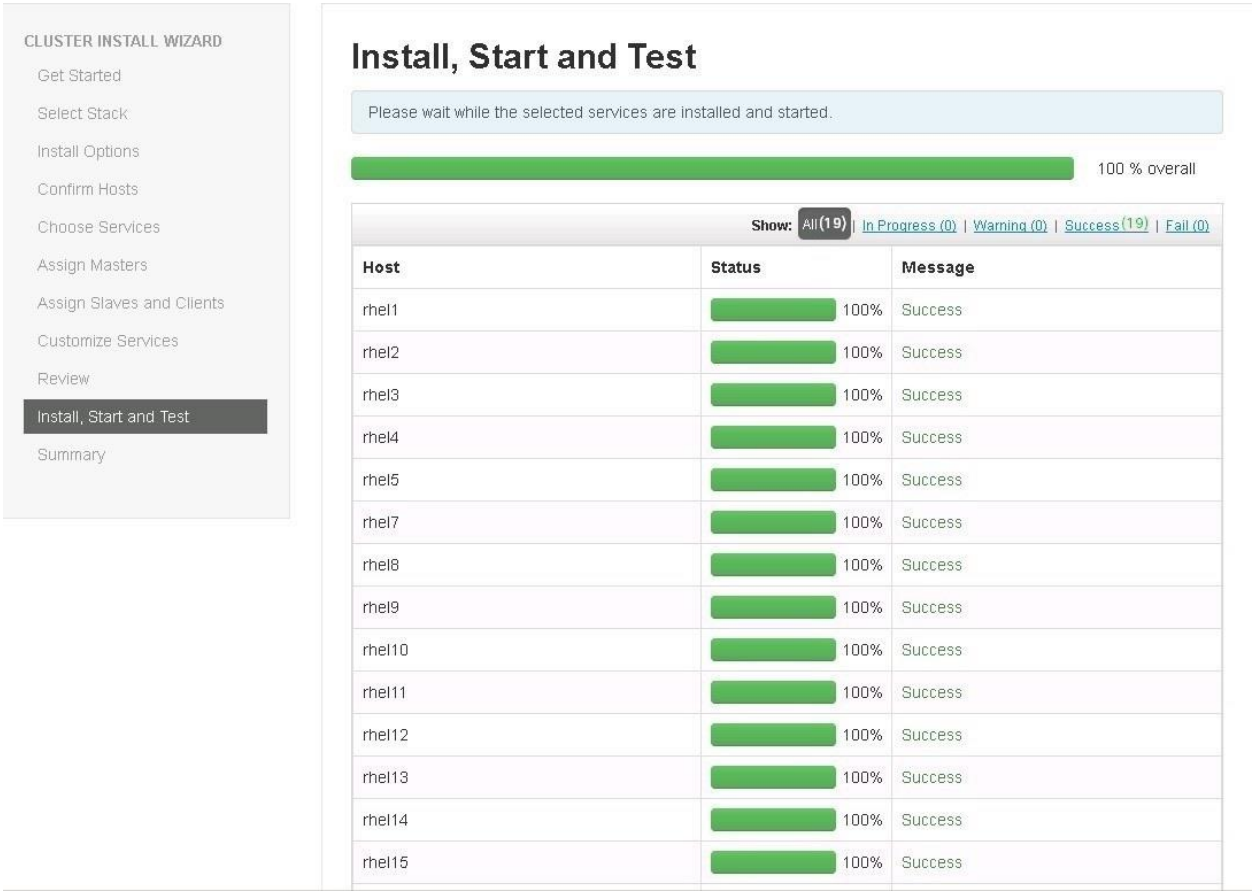
Depending on which components are installing, the entire process may take 10 or more minutes. (Figure 145)

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

Summary of Install Process

The Summary page shows a summary of the accomplished tasks. Click Complete.

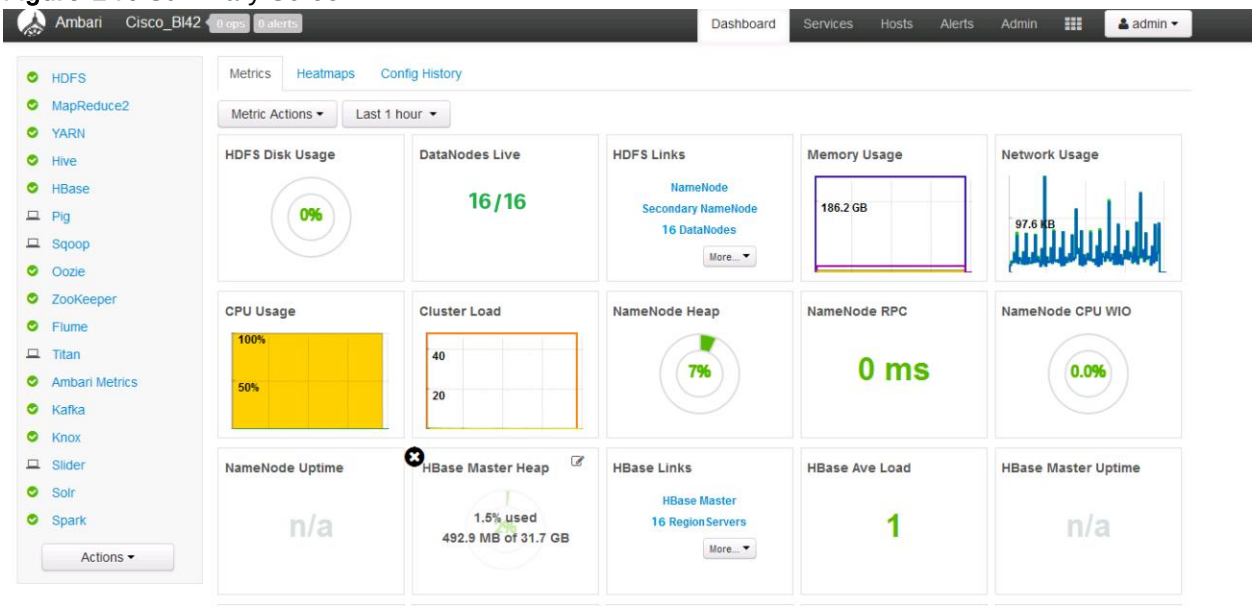
Figure 145 Progress Screen



The Summary page shows a summary of the accomplished tasks. (Figure 146)

Click Complete to finish the installation.

Figure 146 Summary Screen



Enabling High Availability

The idea behind providing High Availability is to ensure the availability of all critical services:

- Without any SLA violations
- Planned or scheduled outages
- Mask unplanned outages (crashes, etc.)

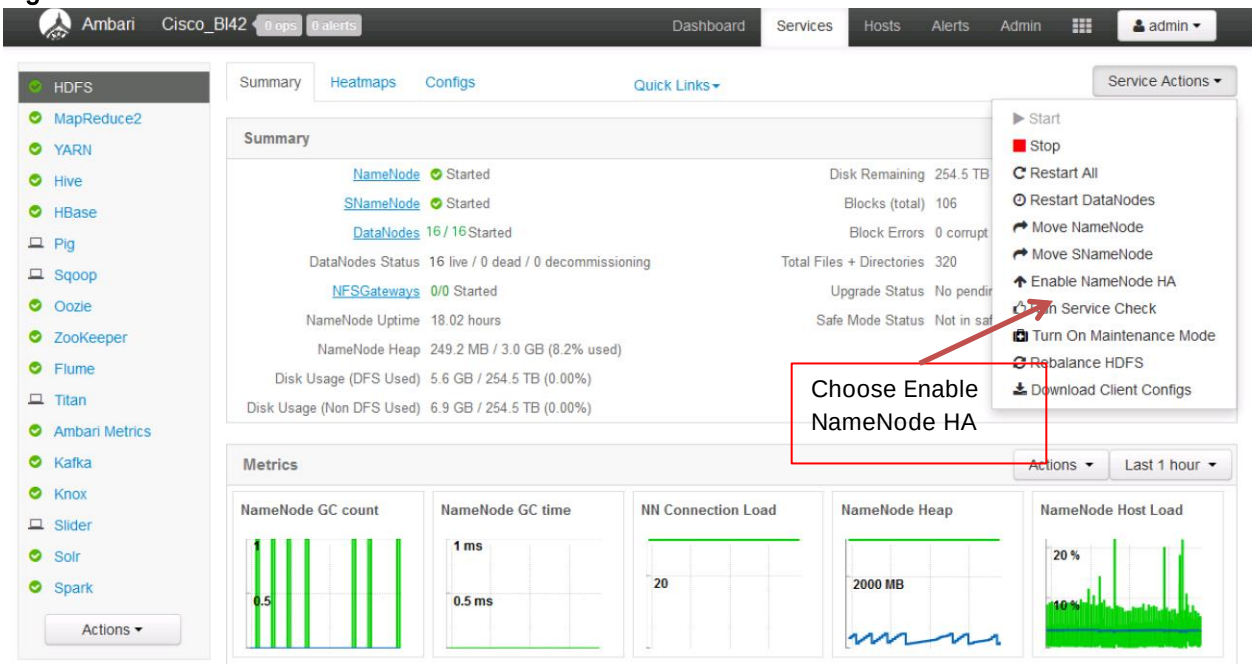
HDFS High Availability

The HDFS HA feature provides the option of running two NameNodes in the same cluster. These NameNodes are referred to as: the Active and the Standby. The Standby NameNode, allowing a fast failover to a new NameNode during a graceful administrator-initiated failover for the purpose of planned maintenance or in the event of machine crashes (unplanned outage).

To setup NameNode high availability, complete the following steps:

1. From the Ambari dashboard, click HDFS Service (Figure 147).
2. Select Service Actions > Enable NameNode HA.

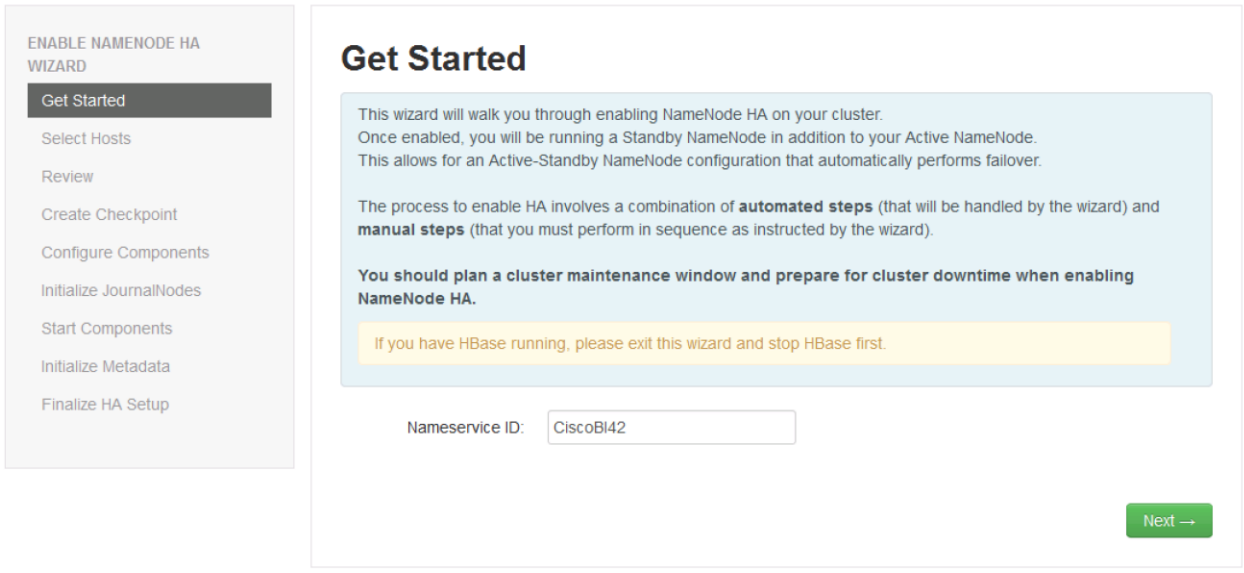
Figure 147 HDFS HA Service



The Enable NameNode HA wizard (0) opens the Enable HA NameNode Wizard. (Figure 148)

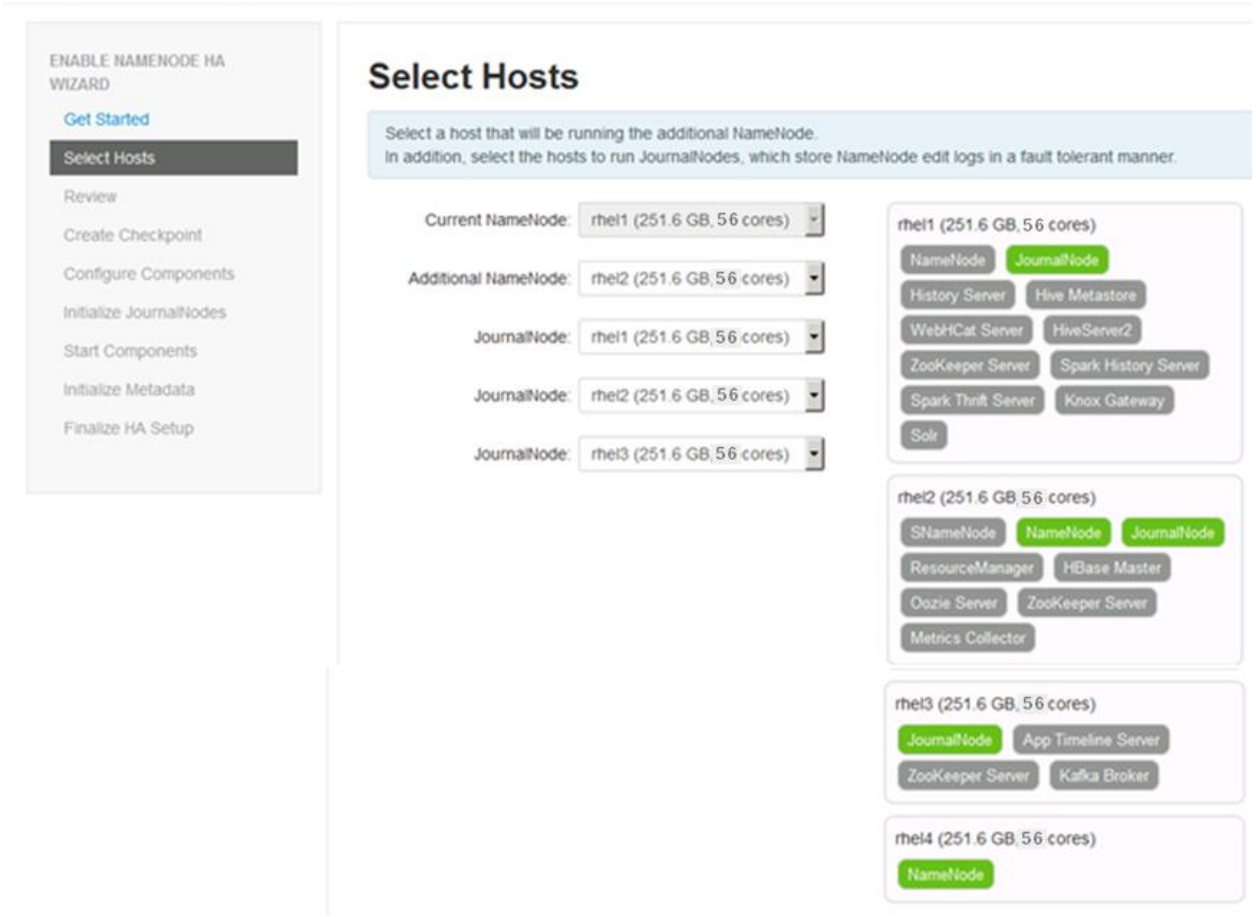
Figure 148 Get Started Window

Enable NameNode HA Wizard



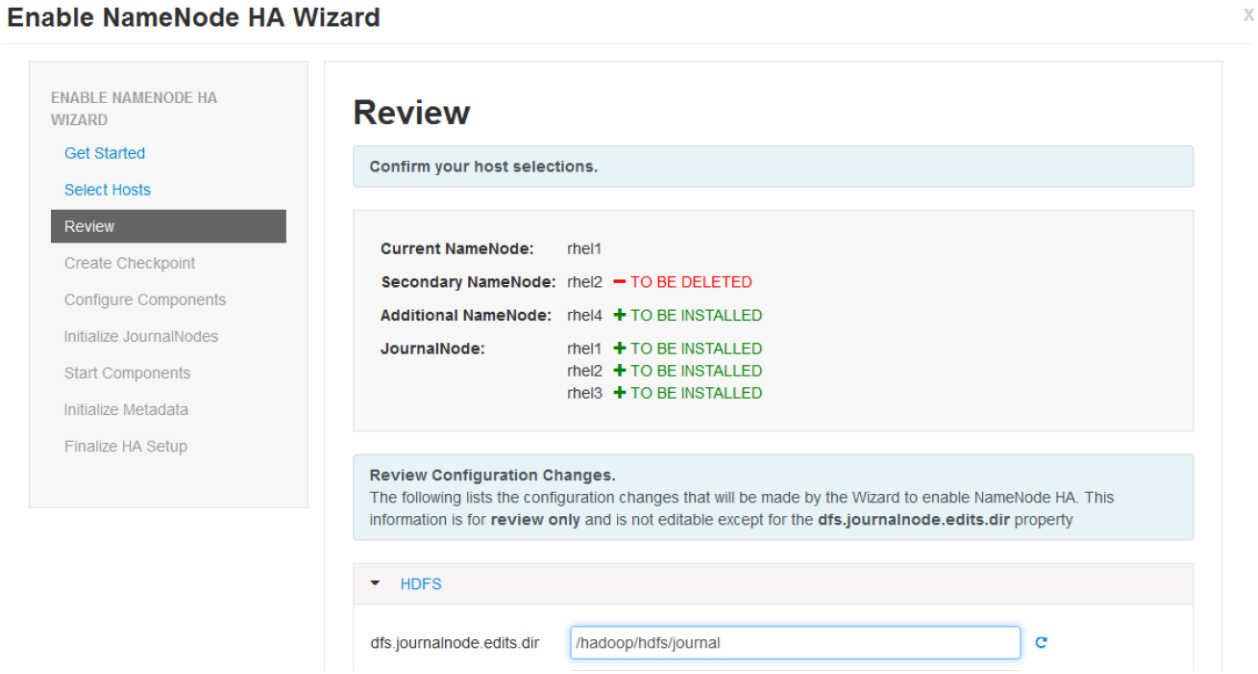
3. On the Select Hosts page, select which hosts to use for additional NameNodes and JournalNodes. (Figure 149)

Figure 149 Select Hosts For NameNode



4. On the Review page (Figure 150), confirm the selections. To change any values, click Back, to continue, click Next.

Figure 150 Review Host Selections



5. Create a checkpoint on the NameNode with some manual steps on the Linux server (rhel1). (Figure 151)

Figure 151 Create a Check Point on the NameNode

6. Log in to the current NameNode host.

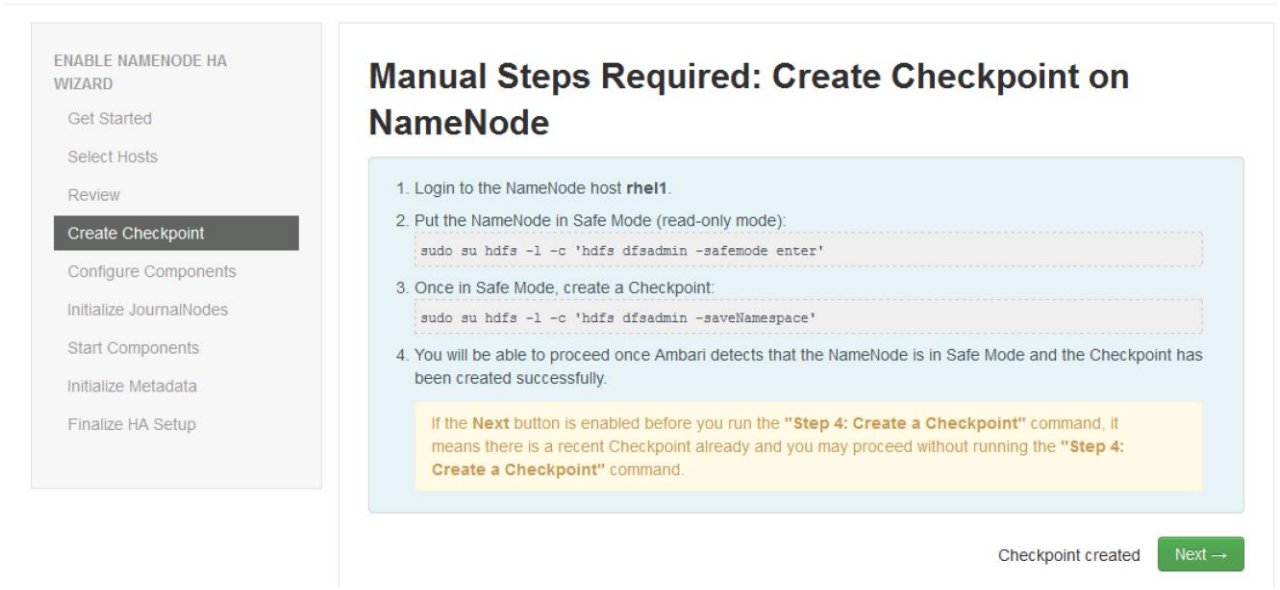
7. Run the following command to put the NameNode into safe mode.

```
[root@rhel1 ~]#
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs dfsadmin -safemode enter'
Safe mode is ON
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs dfsadmin -saveNamespace'
Save namespace successful
```

8. Create the checkpoint.

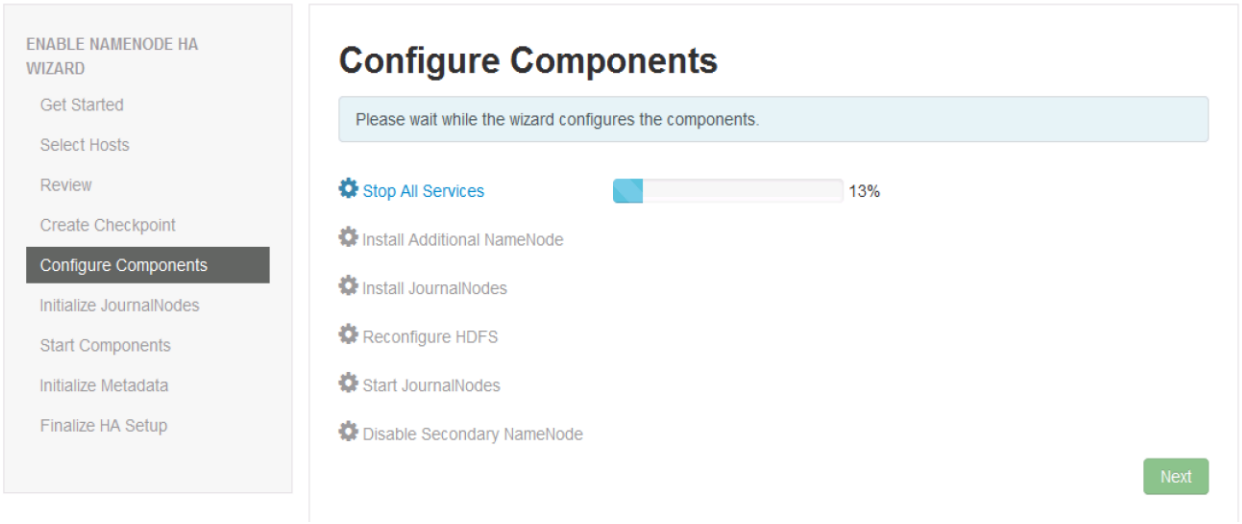
9. Return to the Ambari web interface. When Ambari detects success, click Next. (Figure 152)

Figure 152 Create a Check Point on the NameNode



10. See the progress bars on the Configure Components page, as the wizard configures the components. When all the configuration steps are complete, click Next. (Figure 153)

Figure 153 Configure Components



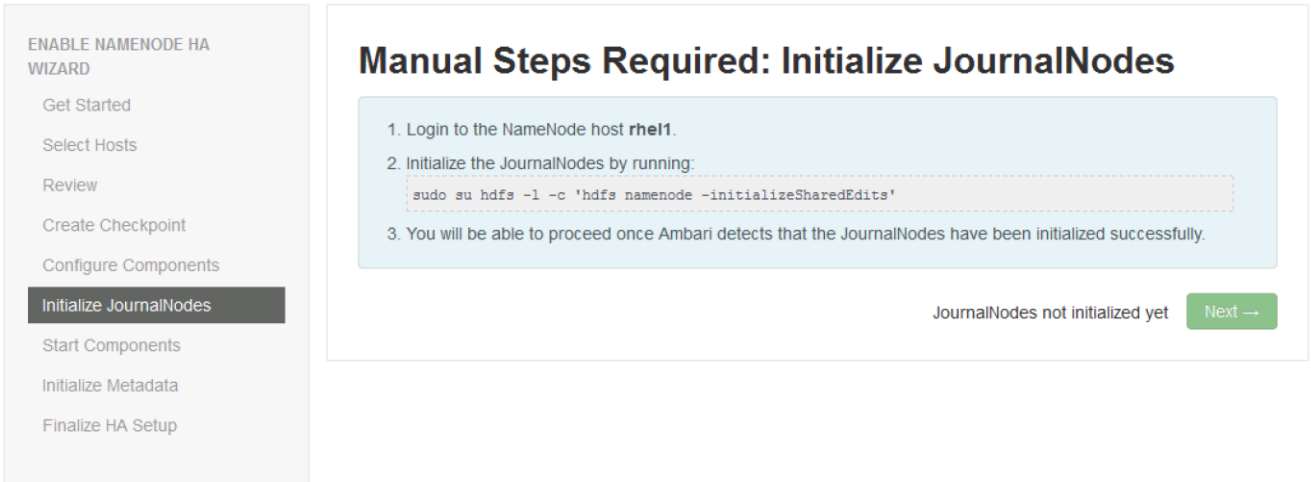
11. Initialize the JournalNodes with the manual step below (Figure 154)

12. Log in to the current NameNode host.

13. Run the following command to initialize the JournalNodes:

```
sudo su hdfs -l -c "hdfs namenode -initializeSharedEdits"
```

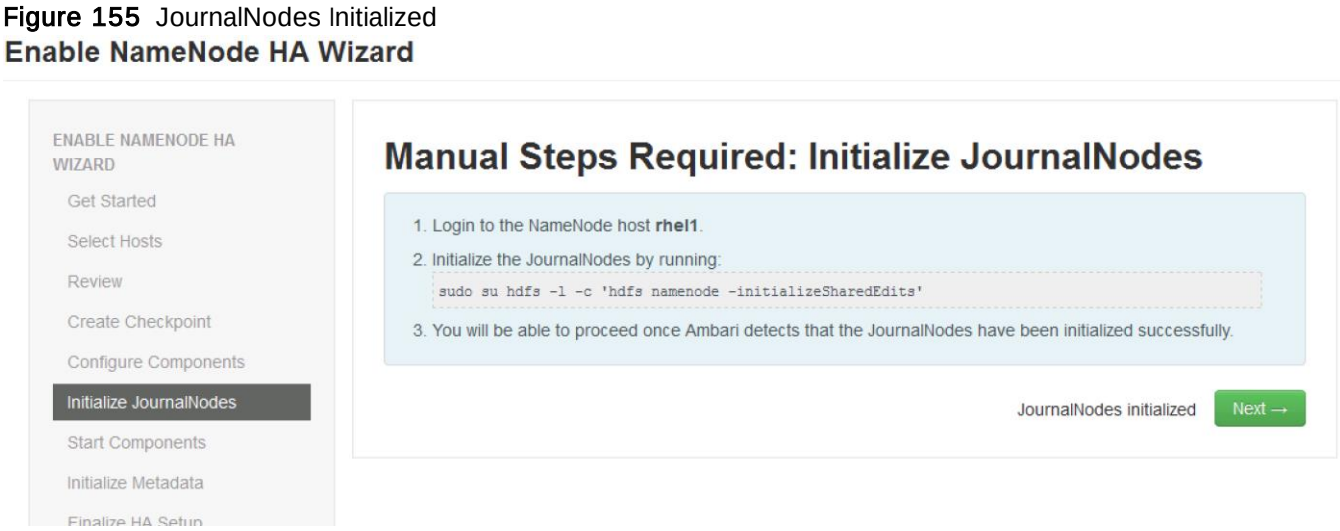
Figure 154 Initialize JournalNodes



```
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs namenode -initializeSharedEdits'
16/07/01 16:16:16 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = rhel1/10.4.1.31
STARTUP_MSG:   args = [-initializeSharedEdits]
STARTUP_MSG:   version = 2.7.2-IBM-12
STARTUP_MSG:   classpath = /usr/iop/4.2.0.0/hadoop/conf:/usr/iop/4.2.0.0/hadoop/lib/ojdbc6.jar:/usr/iop/4.2.0.0/hadoop/lib/jersey-json-1.9.jar:/usr/iop/4.2.0.0/hadoop/lib/ranger-hdfs-plugin-shim-0.5.2-IBM-2.jar:/usr/iop/4.2.0.0/hadoop/lib/jersey-server-1.9.jar:/usr/iop/4.2.0.0/hadoop/lib/ranger-plugin-classloader-0.5.2-IBM-2.jar:/usr/iop/4.2.0.0/hadoop/lib/jets3t-0.9.0.jar:/usr/iop/4.2.0.0/hadoop/lib/ranger-yarn-plugin-shim-0.5.2-IBM-2.jar:/usr/iop/4.2.0.0/hadoop/lib/activation-1.1.jar:/usr/iop/4.2.0.0/hadoop/lib/json-simple-1.1.jar:/usr/iop/4.2.0.0/hadoop/lib/apacheds-i18n-2.0.0-M15.jar:/usr/iop/4.
```

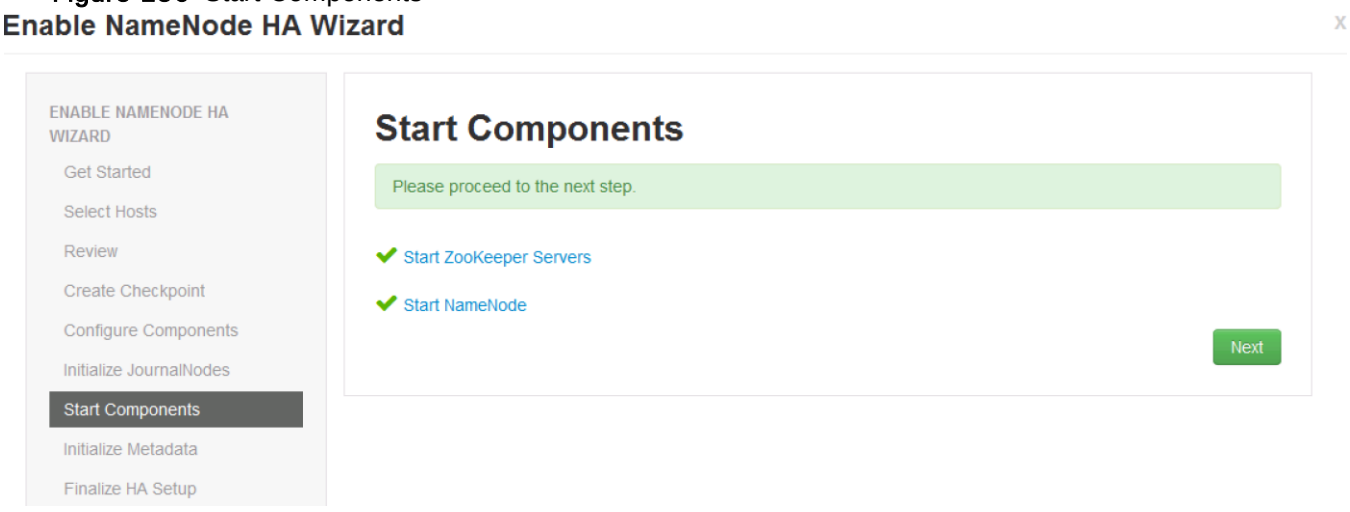
14. Return to the Ambari web interface. When Ambari detects success, click Next. (Figure 155)

Figure 155 JournalNodes Initialized



15. On the start Components page (Figure 156), the wizard starts the Zookeeper servers and the NameNode. Once done, click Next.

Figure 156 Start Components



To initialize NameNode HA Metadata, complete the following steps at the NameNode host (rhel1) (Figure 157):

1. Initialize the metadata for NameNode (automatic failover):

```
sudo su hdfs -l -c 'hdfs zkfc -formatZK'
```

2. Login to the additional NameNode host (rhel4):

3. Run the following command to initialize the metadata for the additional NameNode:

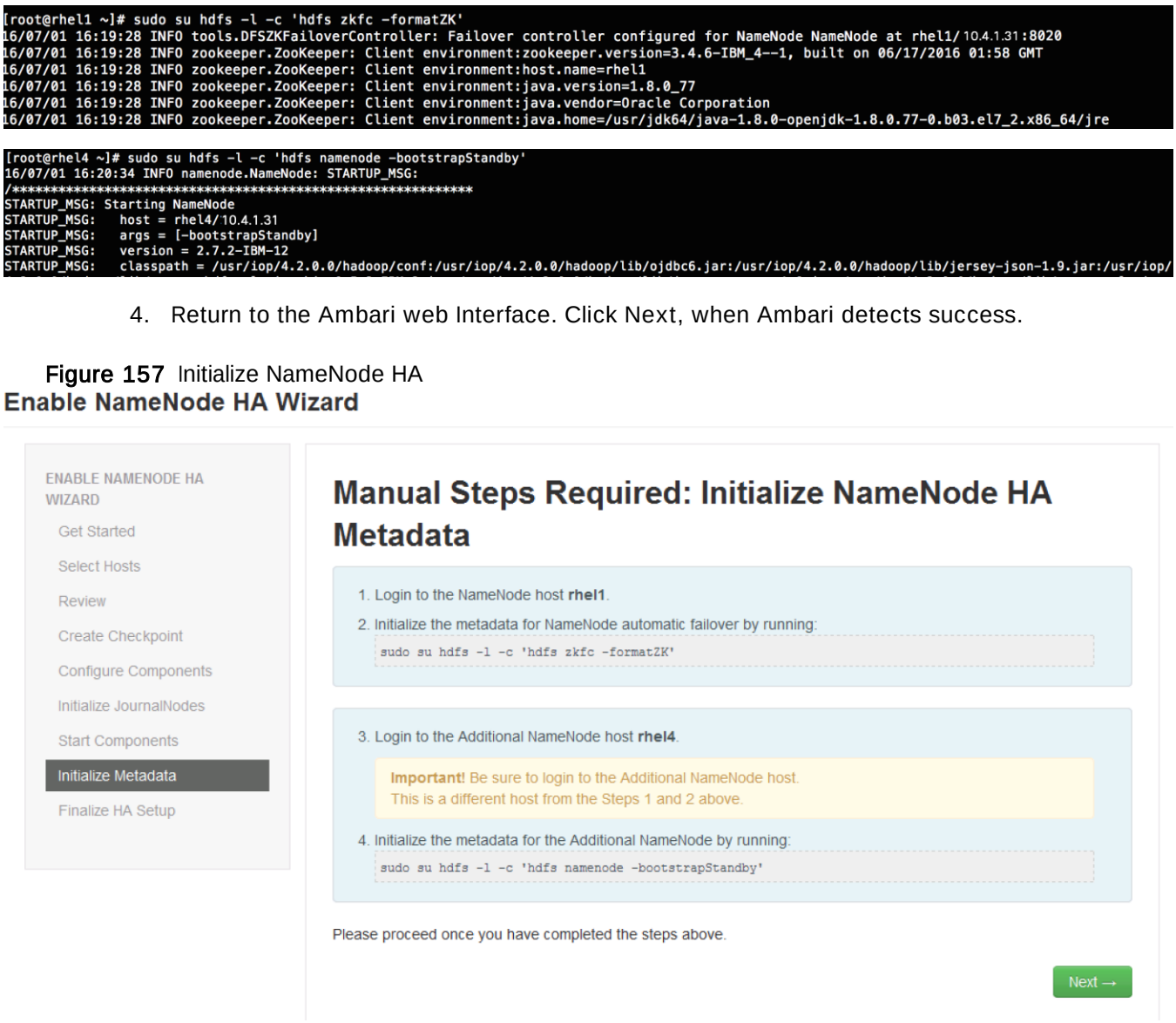
```
sudo su -l hdfs -c 'hdfs namenode -bootstrapStandby'
```

```
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs zkfc -formatZK'
16/07/01 16:19:28 INFO tools.DFSZKFailoverController: Failover controller configured for NameNode NameNode at rhel1/10.4.1.31:8020
16/07/01 16:19:28 INFO zookeeper.ZooKeeper: Client environment:zookeeper.version=3.4.6-IBM_4--1, built on 06/17/2016 01:58 GMT
16/07/01 16:19:28 INFO zookeeper.ZooKeeper: Client environment:host.name=rhel1
16/07/01 16:19:28 INFO zookeeper.ZooKeeper: Client environment:java.version=1.8.0_77
16/07/01 16:19:28 INFO zookeeper.ZooKeeper: Client environment:java.vendor=Oracle Corporation
16/07/01 16:19:28 INFO zookeeper.ZooKeeper: Client environment:java.home=/usr/jdk64/java-1.8.0-openjdk-1.8.0.77-0.b03.el7_2.x86_64/jre
```

```
[root@rhel4 ~]# sudo su hdfs -l -c 'hdfs namenode -bootstrapStandby'
16/07/01 16:20:34 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = rhel4/10.4.1.31
STARTUP_MSG:   args = [-bootstrapStandby]
STARTUP_MSG:   version = 2.7.2-IBM-12
STARTUP_MSG:   classpath = /usr/iop/4.2.0.0/hadoop/conf:/usr/iop/4.2.0.0/hadoop/lib/ojdbc6.jar:/usr/iop/4.2.0.0/hadoop/lib/jersey-json-1.9.jar:/usr/iop/
```

4. Return to the Ambari web Interface. Click Next, when Ambari detects success.

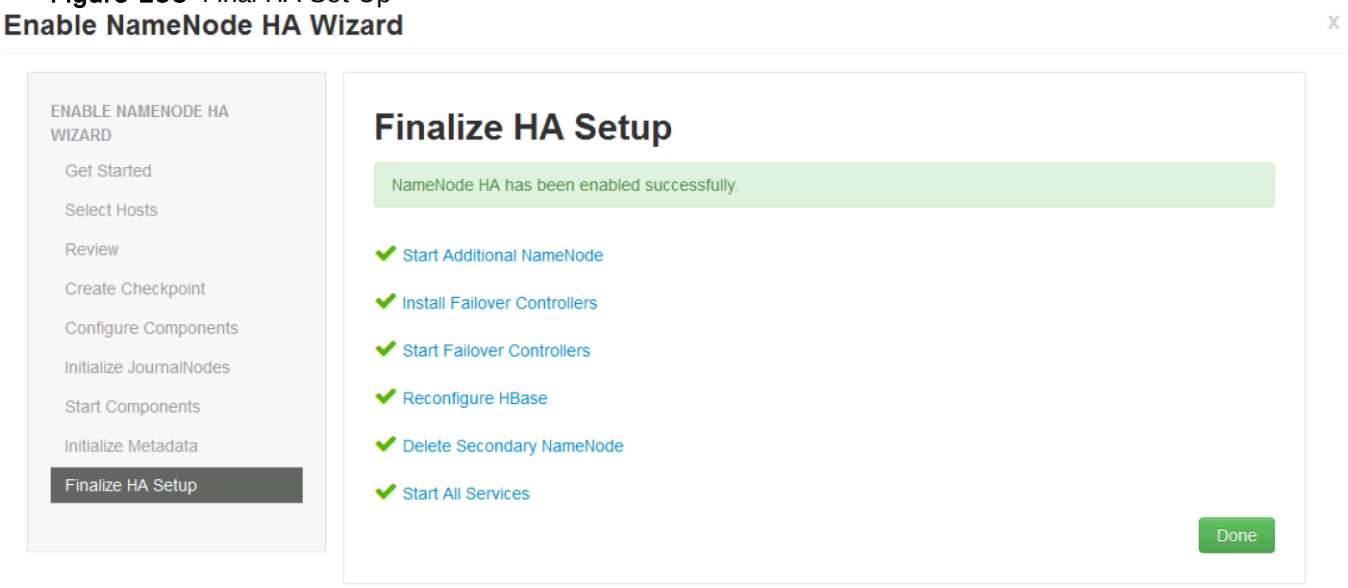
Figure 157 Initialize NameNode HA



On the Finalize HA setup page (Figure 158), the user can see the progress bars as the wizard completes the High Availability setup.

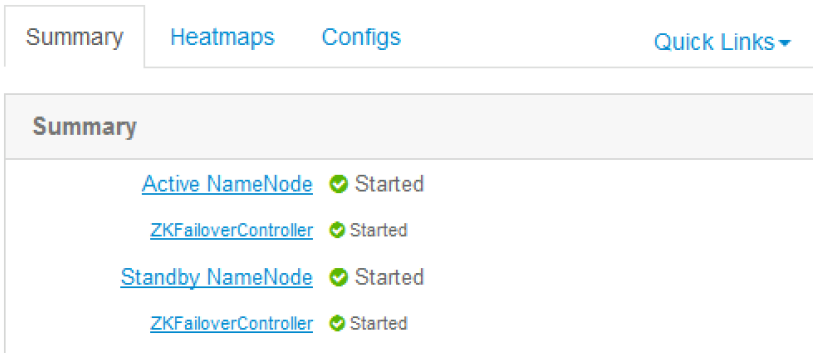
5. When complete click Done.

Figure 158 Final HA Set Up



6. Once the setup is complete, the Ambari web interface will reload. Please confirm to see if any services need to be restarted. (Figure 158)

Figure 159 Ambari Summary



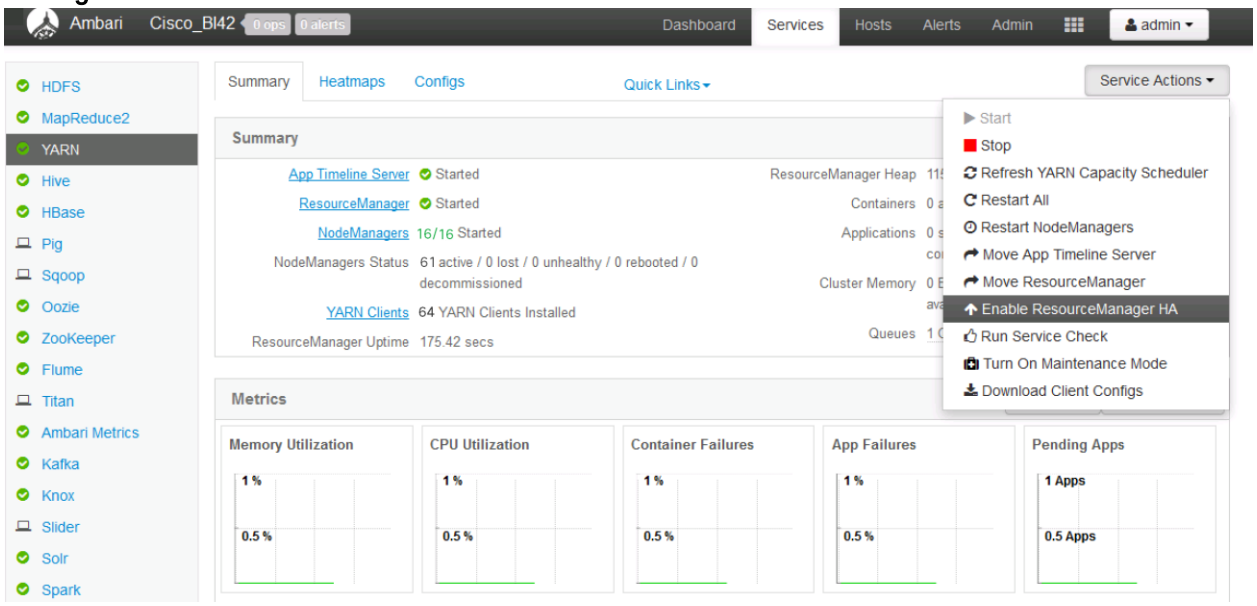
Setting Up Resource Manager (Yarn) High Availability

The process of setting up High Availability for Yarn, sets up a standby resource manager so that if active resource manager goes down, standby takes over and Yarn can continue to function.

Note: At least three Zookeeper servers must be running.

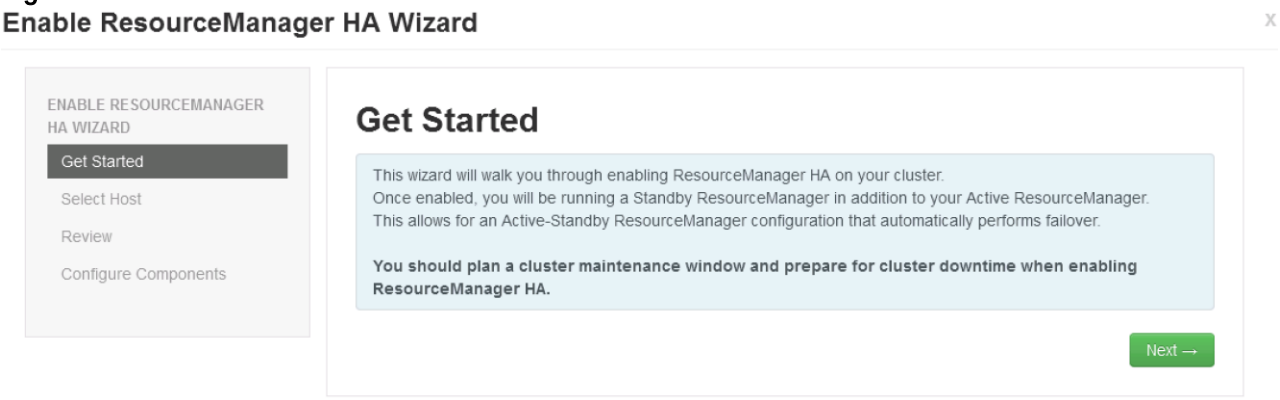
1. From the Ambari web interface, click on the Yarn service (Figure 160).

Figure 160 YARN



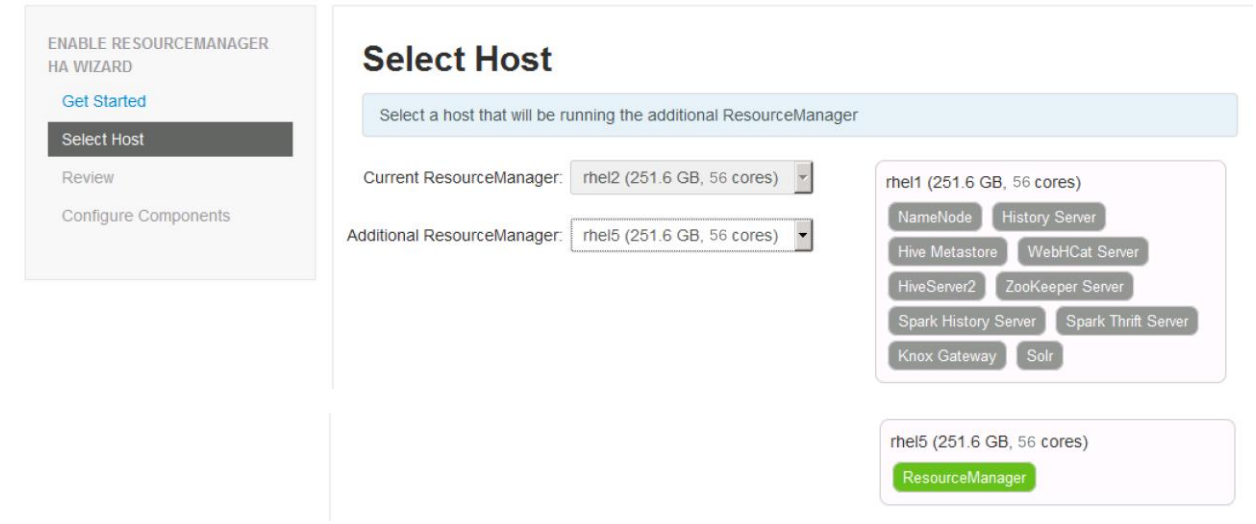
This will open the Resource Manager HA setup wizard (Figure 161):

Figure 161 HA Wizard



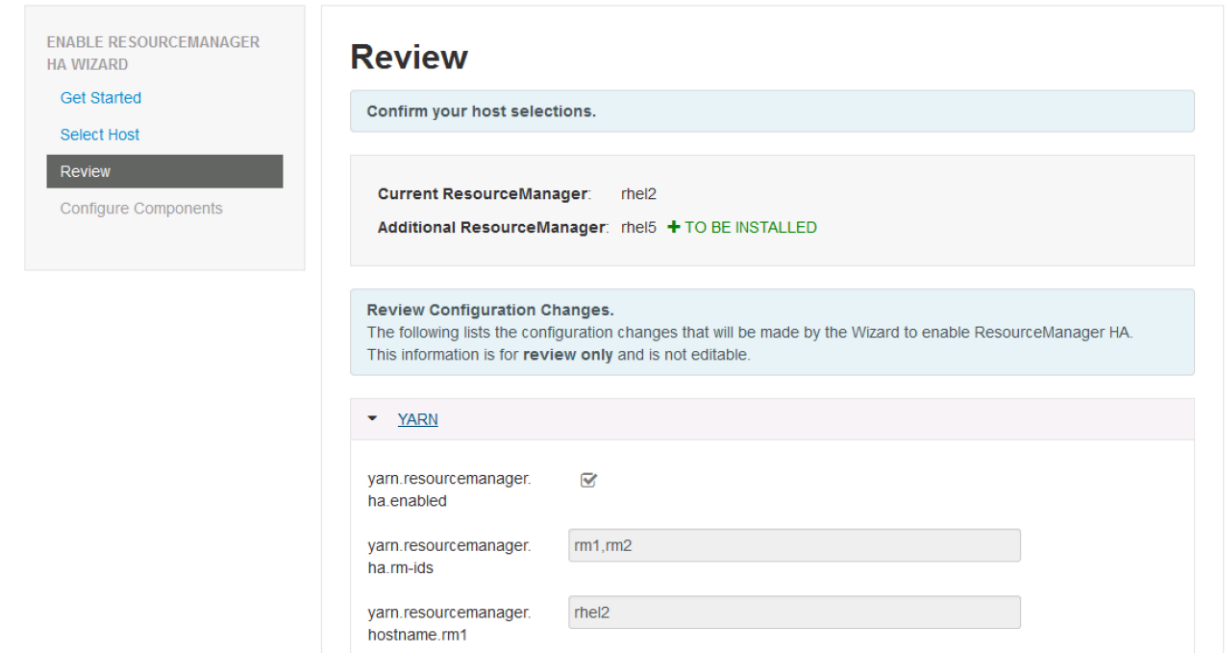
2. From the Select Hosts page, choose the host for Additional Resource Manager (Figure 162).

Figure 162 Select Host for ResourceManager



Proceed to the Review screen (Figure 163).

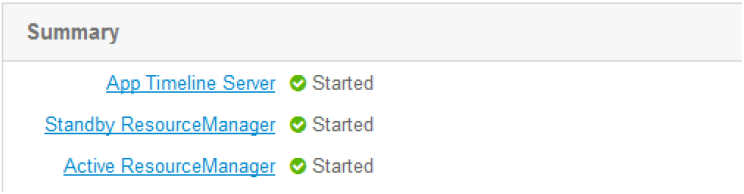
Figure 163 HA Wizard



The Configure Components screen (Figure 164) shows the progress bars as the Additional Resource Manager is being deployed.

Figure 164 Configure Components
Enable ResourceManager HA Wizard

Once complete, the Ambari web interface reloads and informs the user to restart the required services. The active and standby ResourceManagers show they are successfully restarted.



Note: Below are the steps for installing BigSQL. This was done separately so the user may notice a difference in the number of nodes, but all the above steps still apply.

BigSQL Installation

How to Acquire BigInsights Value-added Services

For complete download information (including part numbers and SKUs), go to:

http://www-01.ibm.com/support/docview.wss?uid=swg24042361

In the context of this document, the user will obtain “biginsights_4.2.0.0.el7.x86_64.bin” from the Passport Advantage portal. This file is used to deploy BigInsights 4.2 over Redhat Enterprise Linux 7.

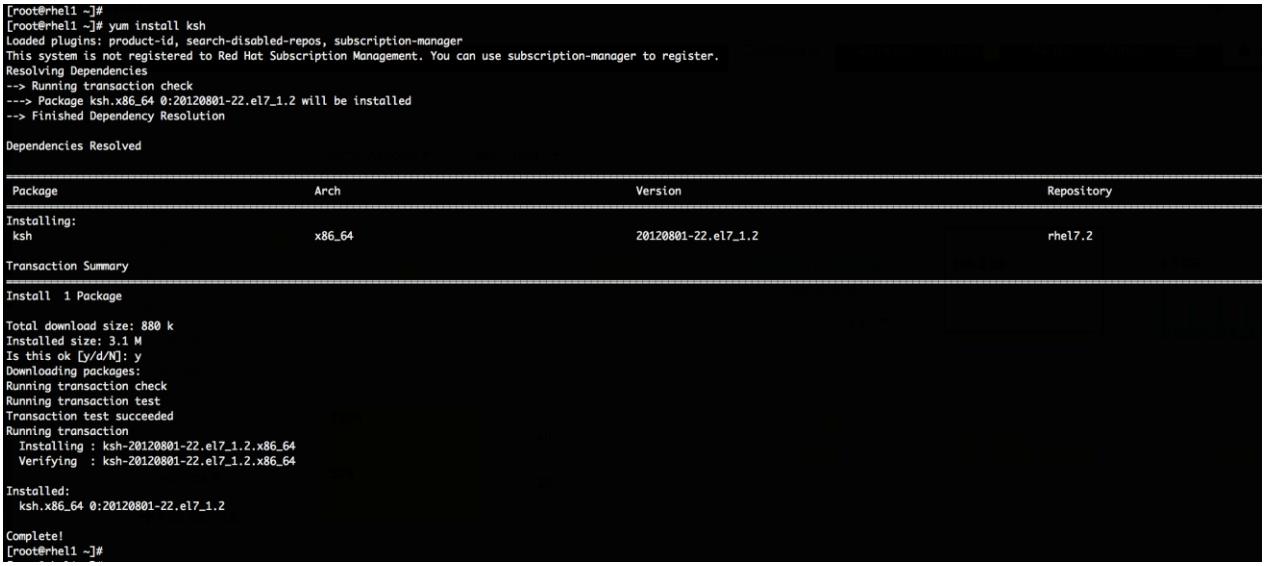
When the user runs the bin file, they will be prompted with the option to download the RPM packages. Then the user may proceed to download the RPMs tar-ball.

Prechecks and Preconditions

It is important to take care of the following precheck criteria before starting with the BigSQL installation on the cluster. To take care of the prechecks and preconditions, complete the following steps:

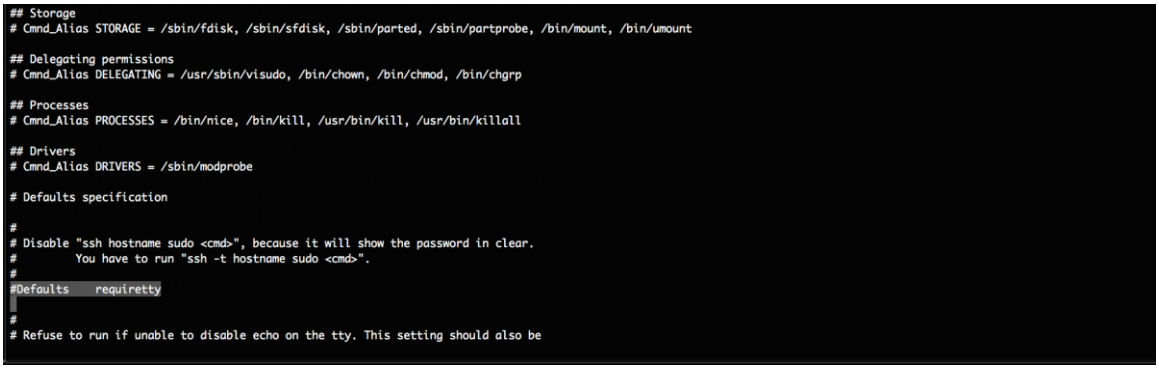
1. Install KornShell (KSH):

Note: It is important that KSH be installed on all the nodes.



2. Disable requiretty: open the file /etc/sudoers and comment out the line “default requiretty”

```
visudo –f /etc/sudoers
```



The changes shown above need to be made on all the nodes that will be part of the BigSQL installation.

3. The storage subsystem should have disks of sector size 512 Bytes.

Important Caveat/Note: The storage disks on the nodes being used for the BigSQL installation should have the sector size of 512 bytes. Otherwise, the user will receive the error "SQLCODE-902".

4. All the server nodes should have Fully Qualified Domain Names, example, rhel1.cisco.com.

To install BigSQL complete the following steps:

1. Untar the downloaded BigInsights 4.2 package. The “BigInsights-Valuepacks” appears.

```
[root@rhel1 BigInsights-Valuepacks]#
[root@rhel1 BigInsights-Valuepacks]#
[root@rhel1 BigInsights-Valuepacks]#
[root@rhel1 BigInsights-Valuepacks]#
[root@rhel1 BigInsights-Valuepacks]# pwd
/root/BigInsights-Valuepacks
[root@rhel1 BigInsights-Valuepacks]# ls
RHEL7
[root@rhel1 BigInsights-Valuepacks]#
```

2. Create the folder:

```
/var/www/html/repos/valueadds
```

3. Copy the downloaded RPMs to this folder.

```
# cp /root/BigInsights-Valuepacks/RHEL7/x86_64/4.2.0.0/*
/var/www/html/repos/valueadds/
```

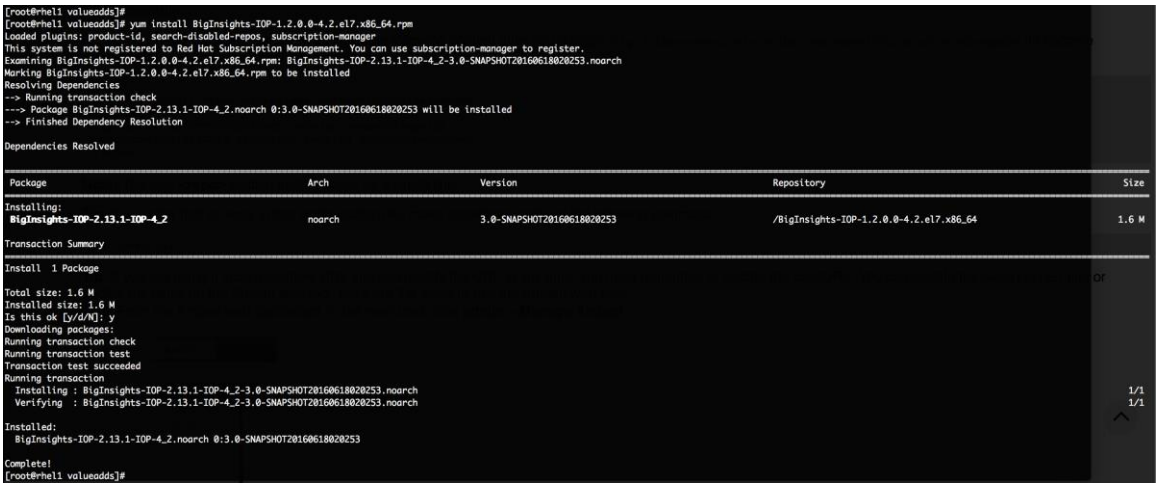
```
[root@rhel1 BigInsights-Valuepacks]#
[root@rhel1 BigInsights-Valuepacks]#
[root@rhel1 BigInsights-Valuepacks]# ls RHEL7/x86_64/4.2.0.0/
BigInsights-IOP-1.2.0.0-4.2.el7.x86_64.rpm  BigR-SystemML-5.12.rpm                  BI-Value-Add-Upgrade-IOP-1.2.0.0-4.2.el7.x86_64.rpm     text-analytics-runtime-4.12.1.46-rpm.rpm
biginsights-tools-8.24.rpm                  bigsheets-distrib-5.21.0.288.rpm       database-migrator-4.6.rpm                               text-analytics-web-tooling-3.12.5.112.rpm
BigR-4.12.1.22.rpm                          bigsheets-patch-distrib-5.21.0.288.rpm db2luw-rhel7-x86_64-11.1.0.0-s160613-db2rpm-beijing.rpm web-ui-framework-2.14.rpm
BigR-BigSQL1-3.10.rpm                       bigsql-dist_4_2_0_0-5.78.2.437-el7-x86_64.rpm dsm-2.1.0.0-N20160606_1925.noarch.rpm
BigR-Jaql-3.9.0.5.rpm                       bigsql-samples_4_2_0_0-5.78.2.437.rpm  jsqsh-4.8.rpm
[root@rhel1 BigInsights-Valuepacks]#
[root@rhel1 BigInsights-Valuepacks]# cp RHEL7/x86_64/4.2.0.0/* /var/www/html/repos/valueadds/
```

4. Run createrepo in the `/var/www/html/repos/valueadds` folder.

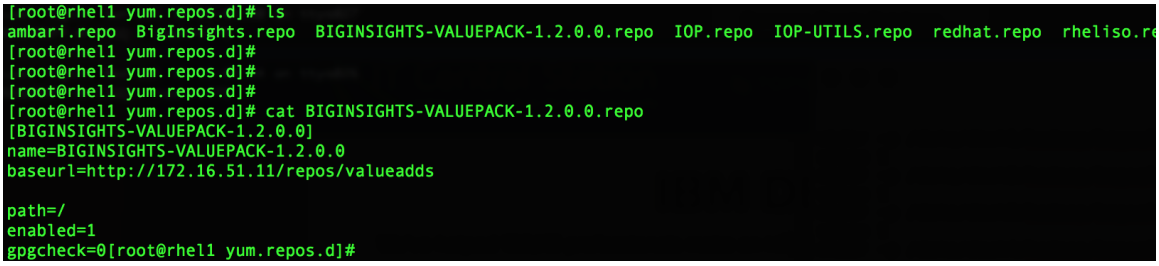
```
[root@rhel1 valueadds]#
[root@rhel1 valueadds]#
[root@rhel1 valueadds]# createrepo .
```

```
[root@rhel1 valueadds]# ls
BigInsights-IOP-1.2.0.0-4.2.el7.x86_64.rpm     BI-Value-Add-Upgrade-IOP-1.2.0.0-4.2.el7.x86_64.rpm
biginsights-tools-8.24.rpm                     database-migrator-4.6.rpm
BigR-4.12.1.22.rpm                             db2luw-rhel7-x86_64-11.1.0.0-s160613-db2rpm-beijing.rpm
BigR-BigSQL1-3.10.rpm                          dsm-2.1.0.0-N20160606_1925.noarch.rpm
BigR-Jaql-3.9.0.5.rpm                          jsqsh-4.8.rpm
BigR-SystemML-5.12.rpm                         repodata
bigsheets-distrib-5.21.0.288.rpm               text-analytics-runtime-4.12.1.46-rpm.rpm
bigsheets-patch-distrib-5.21.0.288.rpm         text-analytics-web-tooling-3.12.5.112.rpm
bigsql-dist_4_2_0_0-5.78.2.437-el7-x86_64.rpm  web-ui-framework-2.14.rpm
bigsql-samples_4_2_0_0-5.78.2.437.rpm
[root@rhel1 valueadds]#
```

5. Run "`yum install BigInsights-IOP-1.2.0.0-4.2.el7.x86_64.rpm`"



6. Check for BIGINSIGHTS-VALUEPACK-1.2.0.0.repo under `/etc/yum.repos.d`



7. Update /var/lib/ambari-server/resources/stacks/BigInsights/4.2/repos/repoinfo.xml file to include:

```
<repo>

<baseurl>http://172.16.51.11/repos/valueadds</baseurl><repoid>BIGINSIGHTS-
VALUEPACK-1.2.0.0</repoid>

<reponame>BIGINSIGHTS-VALUEPACK-1.2.0.0</reponame>

</repo>
```

8. Restart the Ambari server, to reflect the new services to be installed.

9. Issue the command:

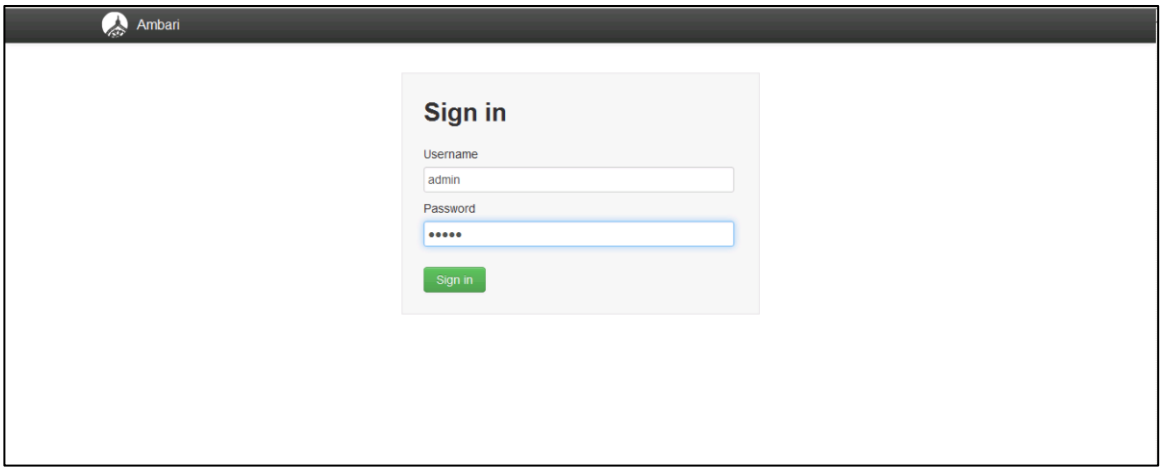
```
service ambari-server restart.
```

```
[root@rhel1 ~]# service ambari-server restart
Using python  /usr/bin/python2
Restarting ambari-server
Using python  /usr/bin/python2
Stopping ambari-server
Ambari Server stopped
Using python  /usr/bin/python2
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 ~]#
```

Once the Ambari server has restarted, the user can see the new services on the web interface.

10. Login to the Ambari server again. (Figure 165)

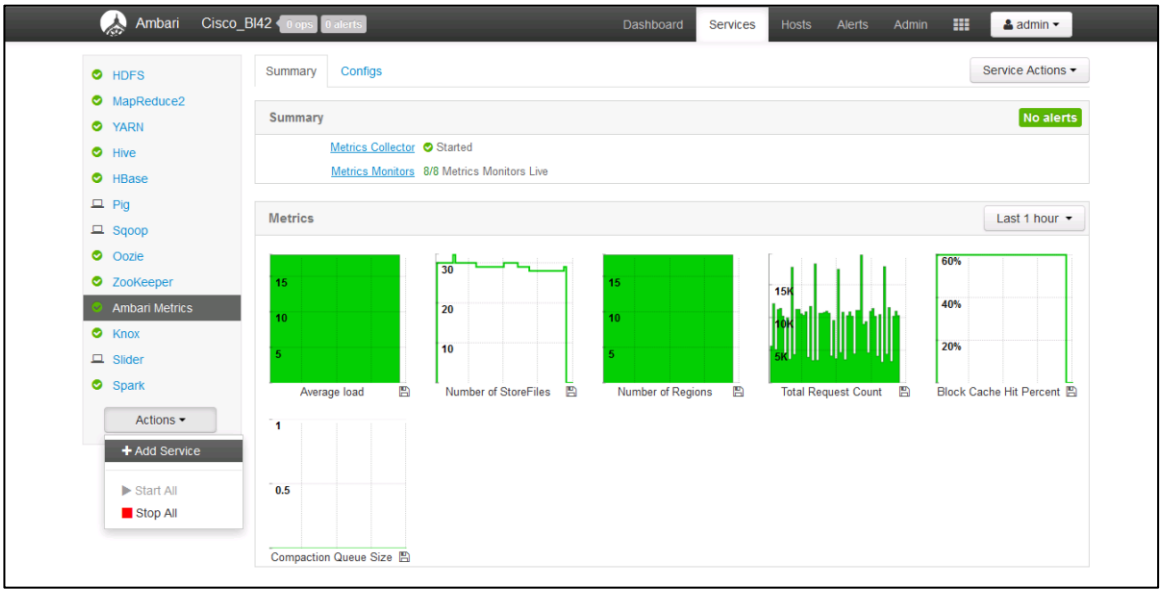
Figure 165 Ambari Login Screen



11. Proceed to install the BigSQL service.

12. From the Actions tab, choose “Add Service”, (Figure 166).

Figure 166 Add Service

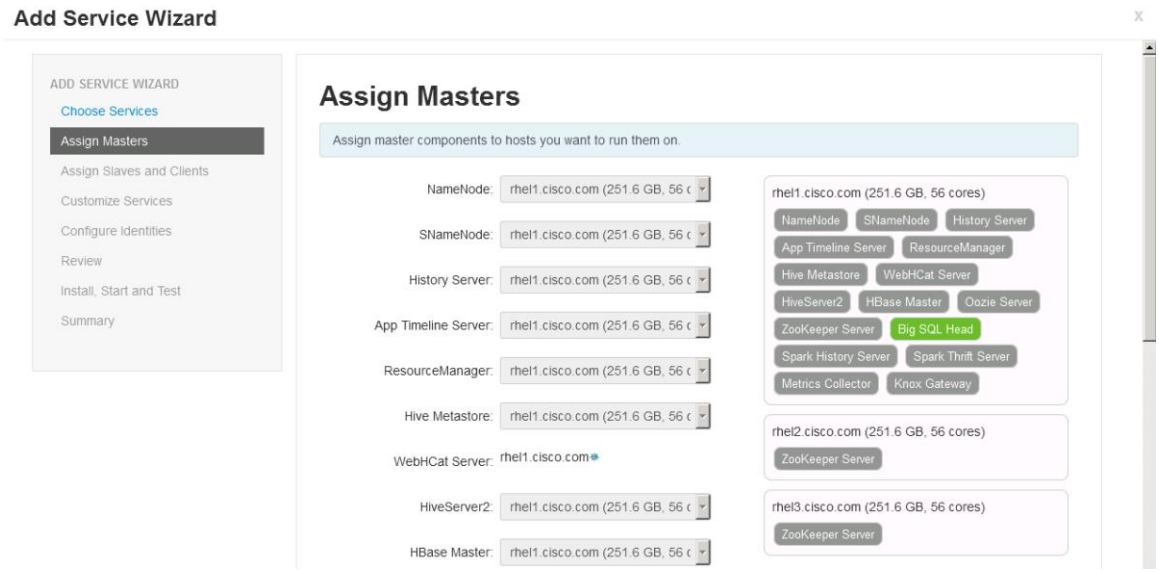


13. From the “Add Service Wizard”, choose “BigInsights BigSQL” service and click Next. (Figure 167)

Figure 167 Add Service Wizard

14. Choose the node on which to install the BigSQL head, (rhel1.cisco.com in this installation example). (Figure 168)

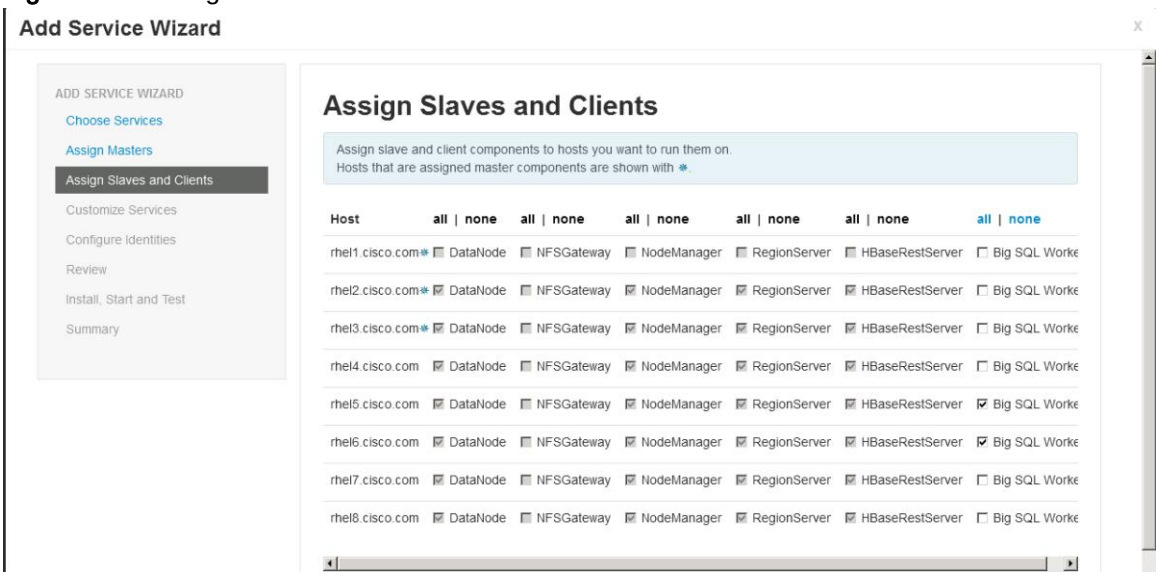
Figure 168 Assign Masters



15. Click Next.

16. Proceed to choose nodes to assign BigSQL worker nodes. (Figure 169)

Figure 169 Assign Slaves and Clients



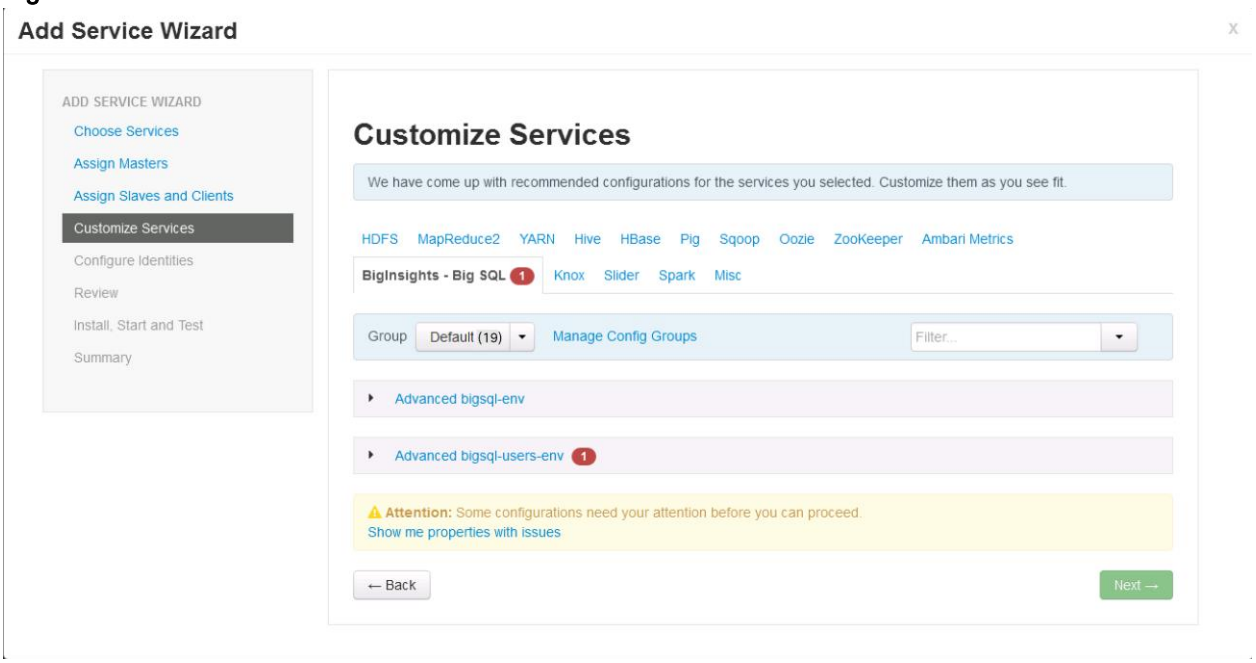
(rhel5.cisco.com and rhel6.cisco.com are shown in this example).

17. After selecting the worker nodes, click Next.

The user will be prompted to customize the service and set up the environment. Any issues and open items will be pointed out now.

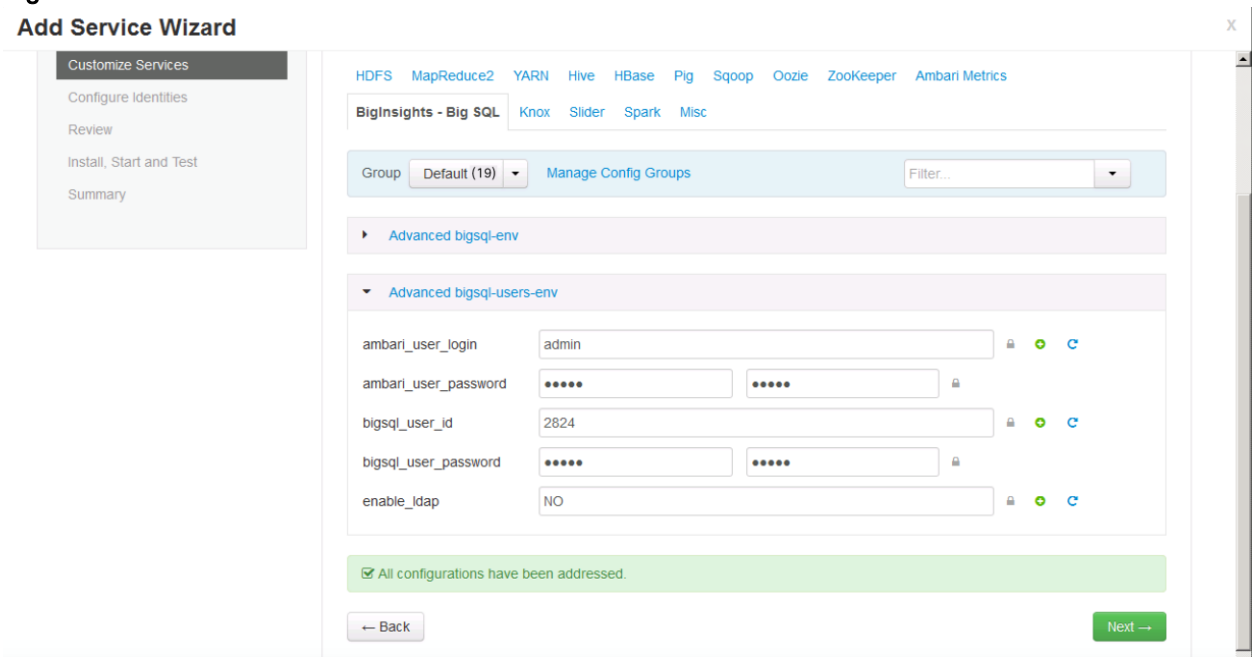
For e.g. Bigsql_user_password is missing here and the user will be prompted to enter the password. (Figure 170)

Figure 170 Customize Services



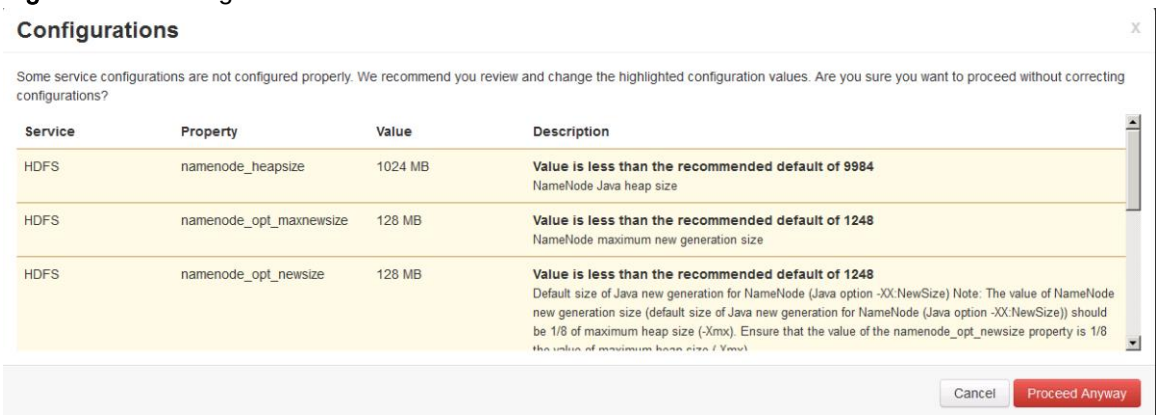
18. Once the password is set, click Next. (Figure 171)

Figure 171 Add Passwords



If prompted to make configuration changes based on Ambari recommendations, make the appropriate changes or click “proceed anyway”, (Figure 172).

Figure 172 Configuration Confirmation Screen

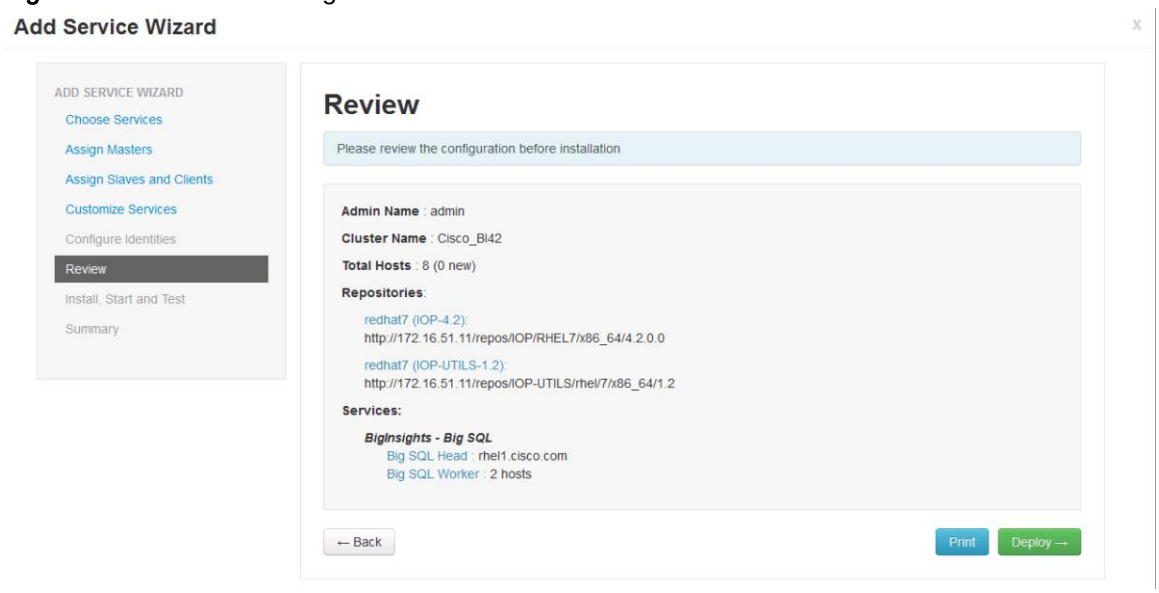


19. Confirm the head node and worker nodes.

20. Click “Deploy” on the review screen.

21. Click “Deploy” to start the install process. (Figure 173)

Figure 173 Review Confirguration



22. Click “Deploy” to start the install process.

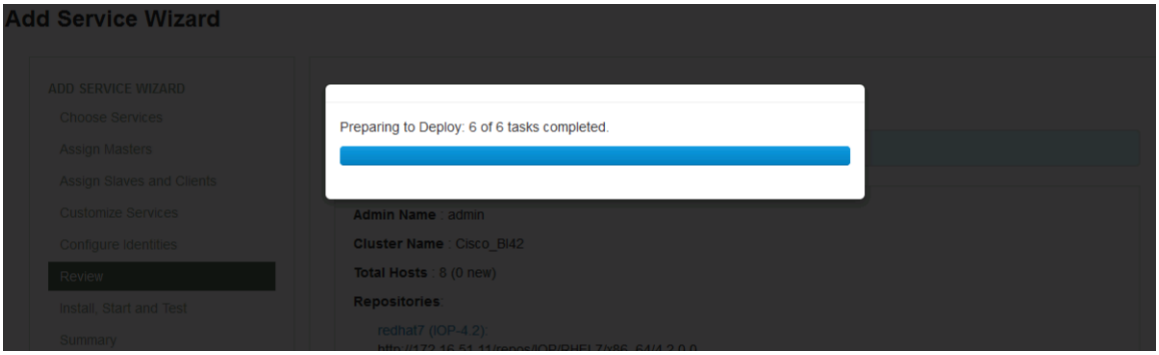
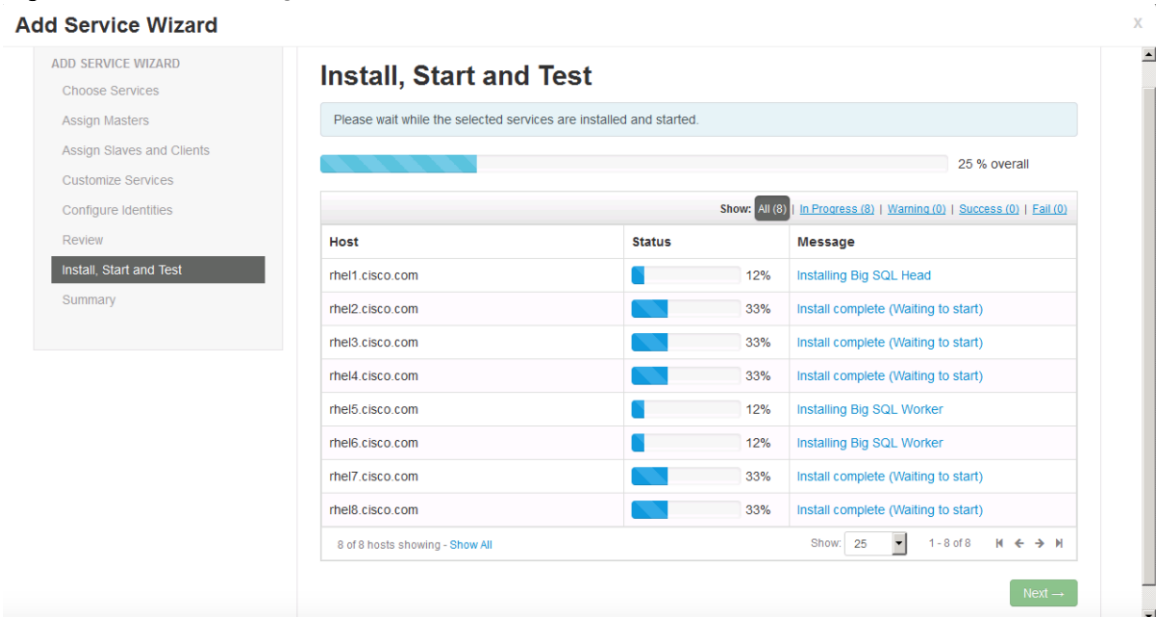
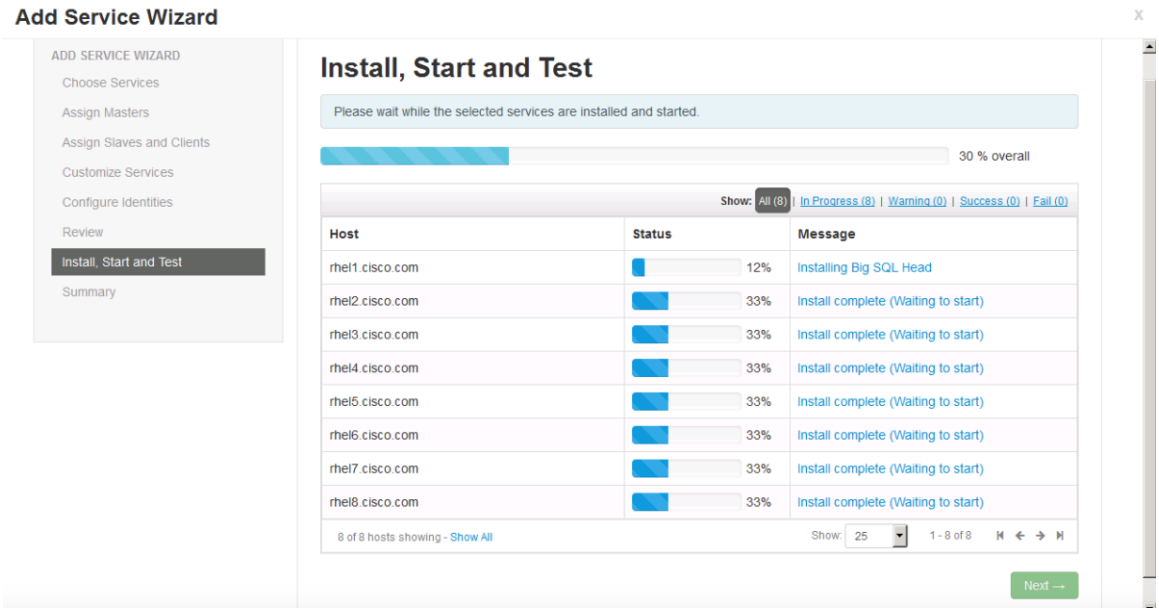


Figure 174 Install Progess Screen

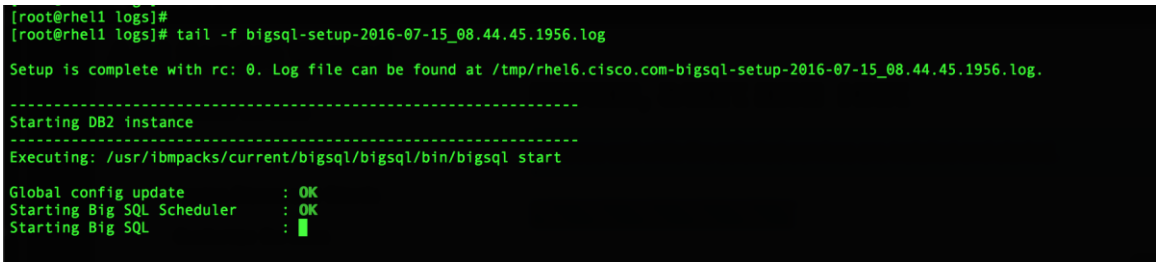


23. Install progress is shown in Figure 174 and Figure 175.

Figure 175 Install Process Screen 2

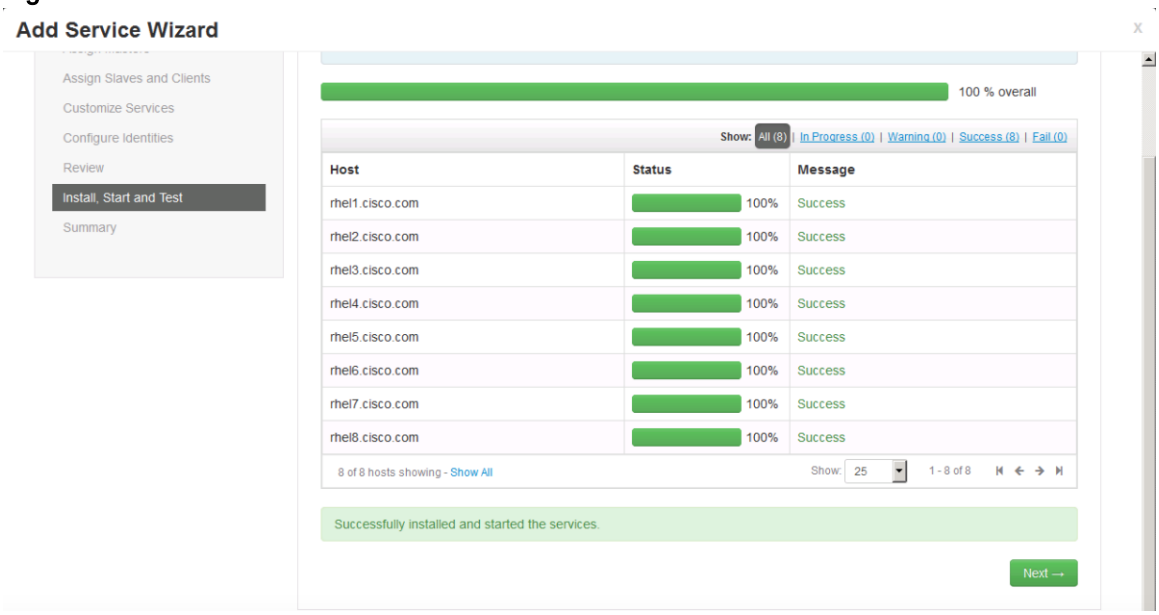


24. Check and track progress through the log file under `/tmp/bigsql/logs/`:



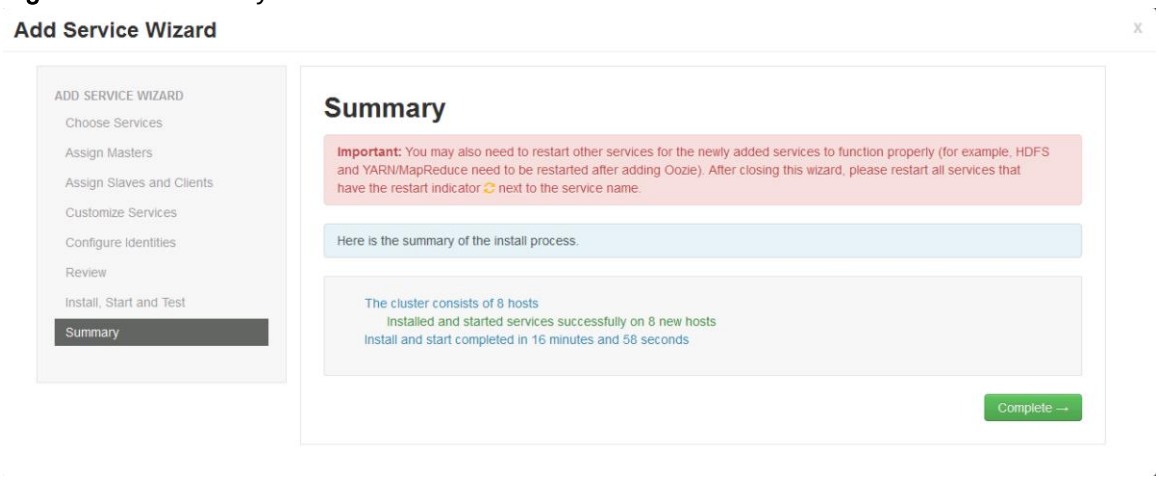
25. Once complete, the installation will show success. (Figure 176)

Figure 176 Successful Installation



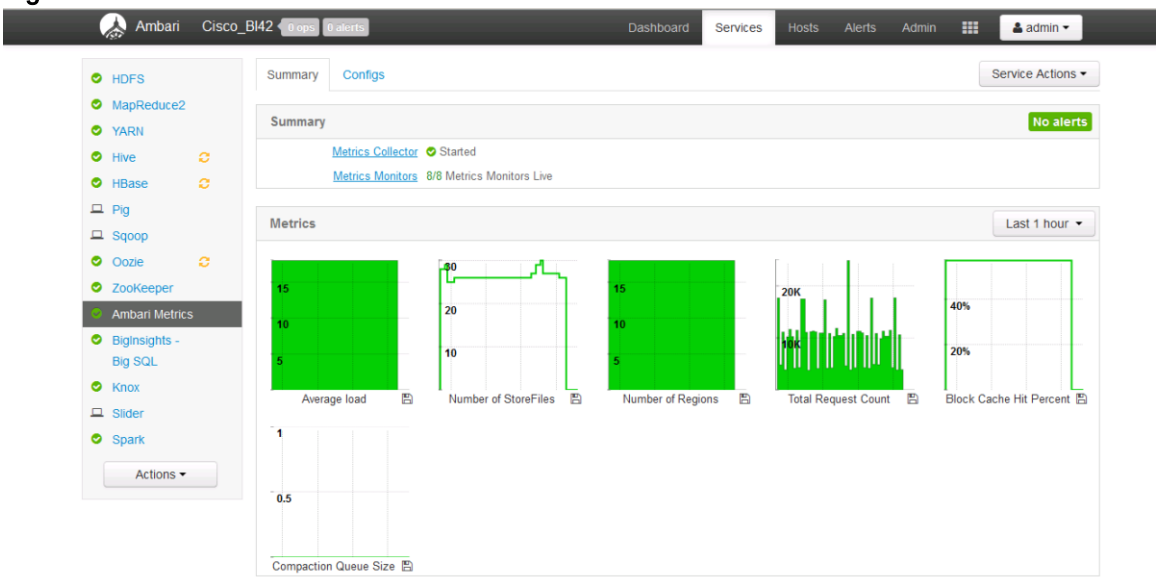
After the successful installation, the user will be prompted for the warning to restart other services for the BigSQL service to work properly. (Figure 177)

Figure 177 Summary



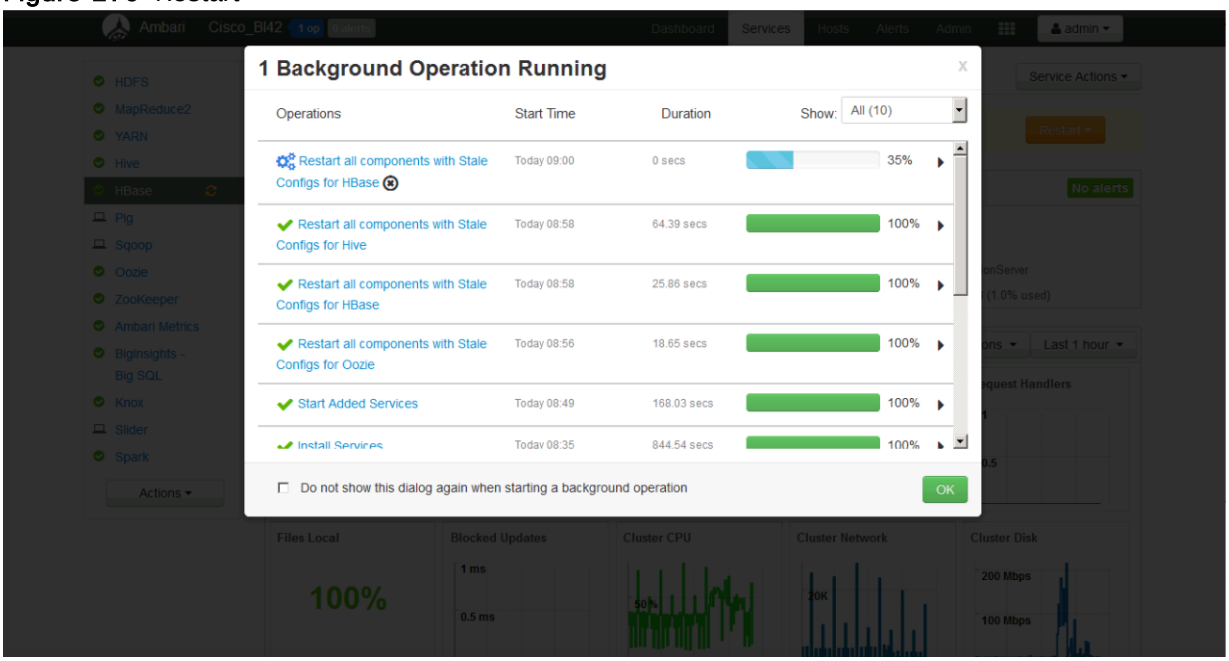
26. Click “Complete” to go back to the Ambari screen with the BigSQL service successfully installed and some services requiring a restart. (Figure 178)

Figure 178 Ambari Dashboard



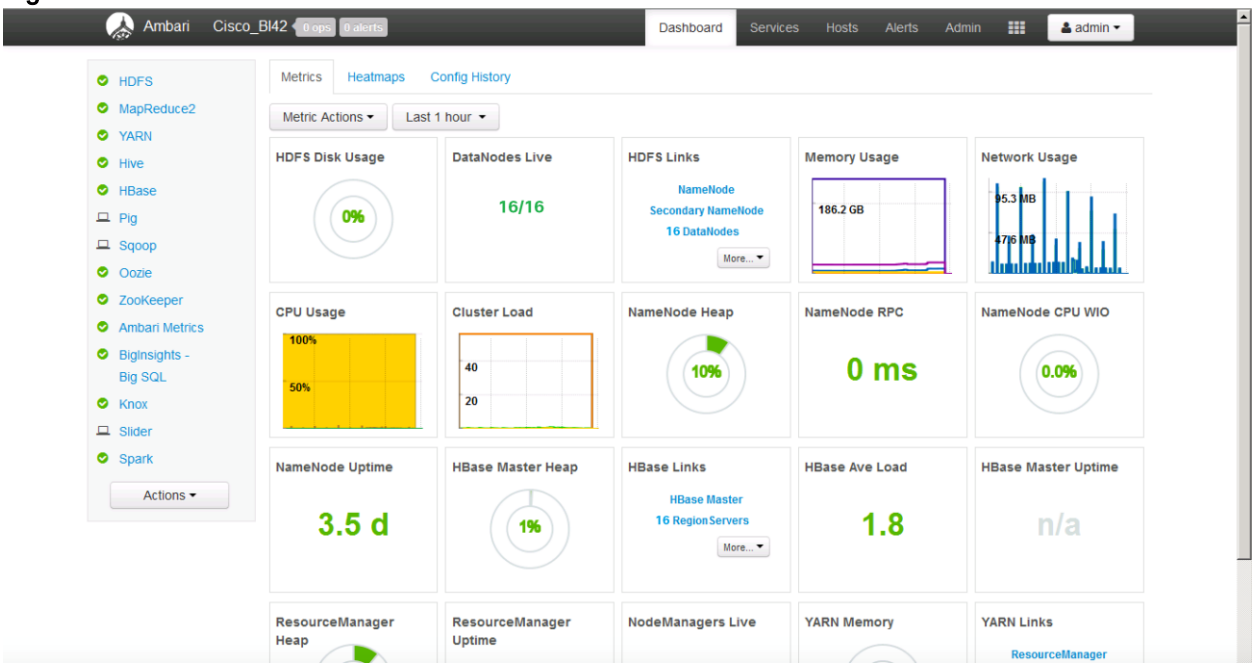
27. Restart all services Hive, Hbase etc.

Figure 179 Restart



Once all services are restarted, the Ambari Service will display the installation details. (Figure 180)

Figure 180 Installation Details



Bill of Materials

This section provides the BOM for the 16 nodes. See Table 16 Bill of Materials for the Cisco UCS Fabric Interconnect 6332, Table 17 Bill of Materials for the Cisco UCS C240M4 Rack Server, Table 18 for the Cisco UCS S3260 Storage Server Base Rack, Table 19 Bill of Materials for Cisco UCS S3260 Storage Server Capacity Rack, and Table 20 for software components.

Table 16 **Bill of Materials for Cisco UCS Fabric Interconnect 6332**

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

Table 17 **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 18 **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 19 **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 20 **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

About the Authors

Rajesh Shroff, Big Data Solutions Architect, Data Center Solutions Group, Cisco Systems, Inc.
Rajesh is a big data infrastructure and performance engineer, with focus on solutions and emerging trends in big data and analytics.

Acknowledgements

- Chinmayi Narasimhadevara, Big Data Software Engineer, Data Center Solutions Group, Cisco Systems, Inc.
- Mike Ahern, Big Data Performance, IBM
- Kathy A. McKnight, Senior Technical Staff Member, Big SQL and DB2 Development, IBM
- Barbara Dixon, Technical Writer, Data Center Solutions Group, Cisco Systems, Inc.
- Karthik Karupasamy, Big Data TME, Data Center Solutions Group, Cisco Systems Inc.