

Cisco Data Intelligence Platform with Hortonworks and Modernizing with NVMe Hybrid Storage

Deployment Guide for the Cisco UCS Integrated Infrastructure for Hortonworks 3.1.0 with Intel NVMe and HDD in Hybrid Storage

Updated: February 27, 2020

Published: December 6, 2019

Partnered with:



About the Cisco Validated Design Program

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

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

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

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

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

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

© 2020 Cisco Systems, Inc. All rights reserved.

Table of Contents

Executive Summary	6
Solution Overview	7
Introduction.....	7
Audience	7
Purpose of this Document	7
What's New in this Release?	8
What's Next?	8
Solution Summary	8
Cisco Data Intelligence Platform	8
Modernizing Hadoop with NVMe	10
Reference Architecture	10
Data Lake Reference Architecture	10
Technology Overview	13
Cisco UCS Integrated Infrastructure for Big Data and Analytics	13
Cisco UCS Manager.....	13
Cisco UCS 6300 Series Fabric Interconnects	13
Cisco UCS C-Series Rack-Mount Servers	14
Cisco UCS C240 M5 Rack-Mount Server	14
Cisco UCS Virtual Interface Cards (VICs).....	15
Intel P4510 Series Data Center NVMe.....	16
Hortonworks Data Platform.....	17
Apache Ambari.....	17
HDP for Data Access	17
Red Hat Ansible Automation.....	18
Solution Design	19
Requirements	19
Physical Topology	19
Port Configuration on Fabric Interconnect.....	19
Server Configuration and Cabling for Cisco UCS C240 M5.....	19
Software Distributions and Versions.....	20
Deployment Hardware and Software	22
Cisco Unified Computing System Configuration	22
Configure Cisco UCS Fabric Interconnect	22
Configure Fabric Interconnects for a Cluster Setup	22
Configure Cisco UCS Manager	24
Create Service Profile Template	53

Install Red Hat Enterprise Linux 7.6.....	60
Post OS Install Configuration	71
Configure /etc/hosts	71
Set Up Passwordless Login	72
Create a Red Hat Enterprise Linux (RHEL) 7.6 Local Repository	73
Create the Red Hat Repository Database.....	74
Set Up Ansible	74
Install httpd.....	77
Set Up All Nodes to use the RHEL Repository	77
Upgrade the Cisco Network Driver for VIC1387	78
Set Up JAVA	79
Enable Syslog.....	80
Set the ulimit	81
Disable SELinux.....	82
Set TCP Retries.....	82
Disable the Linux Firewall	83
Disable IPv6 Defaults.....	83
Disable Swapping.....	83
Disable Memory Overcommit	84
Disable Transparent Huge Pages	84
NTP Configuration	85
Install Megaraid StorCLI	86
Configure the Filesystem for NameNodes and DataNodes	87
Delete Partitions	89
Cluster Verification	90
Install HDP 3.1.0.....	92
Prerequisites for HDP Installation	92
Hortonworks Repository	92
Downgrade Snappy on All Nodes	95
HDP Installation.....	95
Install and Setup Ambari Server on rhel1	95
Setup Ambari Server On Admin Node (Rhel1)	101
Launch the Ambari Server	103
Create the Cluster.....	104
Select Version.....	105
Select Hosts.....	105
Hostname Pattern Expressions	107

Confirm Hosts	108
Choose Services.....	108
Assign Masters.....	109
Assign Slaves and Clients	110
Customize Services	111
HDFS.....	115
MapReduce2.....	117
YARN.....	117
Configure the HDFS NameNode High Availability.....	119
HBase	119
Zookeeper	119
Storm	119
Ambari Metrics	120
Accumulo	121
Atlas	121
Kafka	121
Knox.....	121
SmartSense	122
Spark.....	122
Review	122
Deploy.....	123
Summary of the Installation Process.....	124
High Availability for HDFS NameNode and YARN ResourceManager	125
Configure the HDFS NameNode High Availability.....	125
Configure the YARN ResourceManager HA	134
Configure NVMe as YARN Local Directory	137
Summary	139
For More Information	139
Bill of Materials	140
Appendix.....	144
Configure Data Drives on Name Node and Data Nodes.....	144
Configure Data Drives on Name Nodes	145
Configure Data Drives on Data Nodes.....	146
About the Authors.....	147
Acknowledgements	147

Executive Summary

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

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

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

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

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

This CVD implements Cisco UCS Integrated Infrastructure using Hortonworks 3.1.0 and offers hybrid storage model which includes Intel Non-Volatile Memory Express (NVMe) to host Hadoop temp data along with hard disk drives (HDD) for Hadoop Distributed File System (HDFS). As a result, this reference architecture helps achieve improved performance with fewer nodes and maintain better TCO.

Solution Overview

Introduction

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

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

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

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

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

Audience

The intended audience of this document includes sales engineers, field consultants, professional services, IT managers, partner engineering and customers who want to deploy the Hortonworks 3.1.0 on the Cisco UCS Integrated Infrastructure for Big Data and Analytics with Intel NVMe (Cisco UCS M5 Rack-Mount servers).

Purpose of this Document

This document describes the architecture and deployment procedures for Hortonworks 3.1.0 on a 28-node Cisco UCS C240 M5 cluster based on Cisco UCS Integrated Infrastructure for Big Data and Analytics.

This document describes the architecture and step by step guidelines of deployment procedures for Cisco Data Intelligence Platform using Hortonworks 3.1.0 on Cisco UCS C240 M5 with Intel NVMe and Spinning HDDs in hybrid configuration.

What's New in this Release?

This CVD implements the following:

- Data Lake with Hortonworks 3.1.0 on Cisco UCS Integrated Infrastructure with hybrid storage model consisting Intel NVMe and HDD for Big Data and Analytics
- Installation and setup of the above through Hortonworks Ambari
- Integration of Intel NVMe with CDH for separating Temp or Spark MMap files to NVMe and HDFS to HDD.

What's Next?

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

- Cisco Boot optimized M.2 Raid Controller for hardware RAID
- 4th Generation FI
- All NVMe solution for Cisco UCS Integrated Infrastructure for Big Data and Analytics

Solution Summary

This CVD details the process of installing Hortonworks 3.1.0 and the configuration details of the cluster. The current version of Cisco UCS Integrated Infrastructure for Big Data and Analytics offers the following configurations depending on the compute and storage requirements.

Cisco Data Intelligence Platform

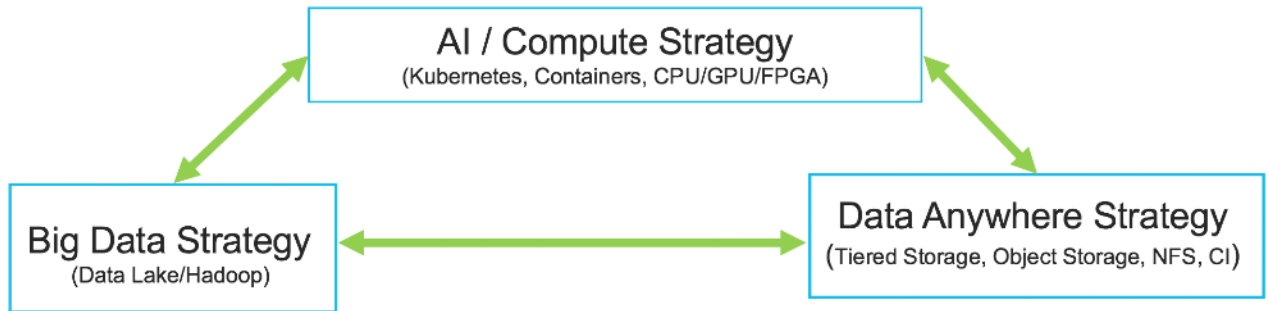
Cisco Data Intelligence Platform (CDIP) is a cloud scale architecture which brings together big data, AI/compute farm, and storage tiers to work together as a single entity while also being able to scale independently to address the IT issues in the modern data center. This architecture allows for:

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

Cisco Data Intelligence Platform caters to the evolving architecture bringing together a fully scalable infrastructure with centralized management and fully supported software stack (in partnership with industry leaders in the space)

to each of these three independently scalable components of the architecture including data lake, AI/ML and Object stores.

Figure 1 Cisco Data Intelligent Platform

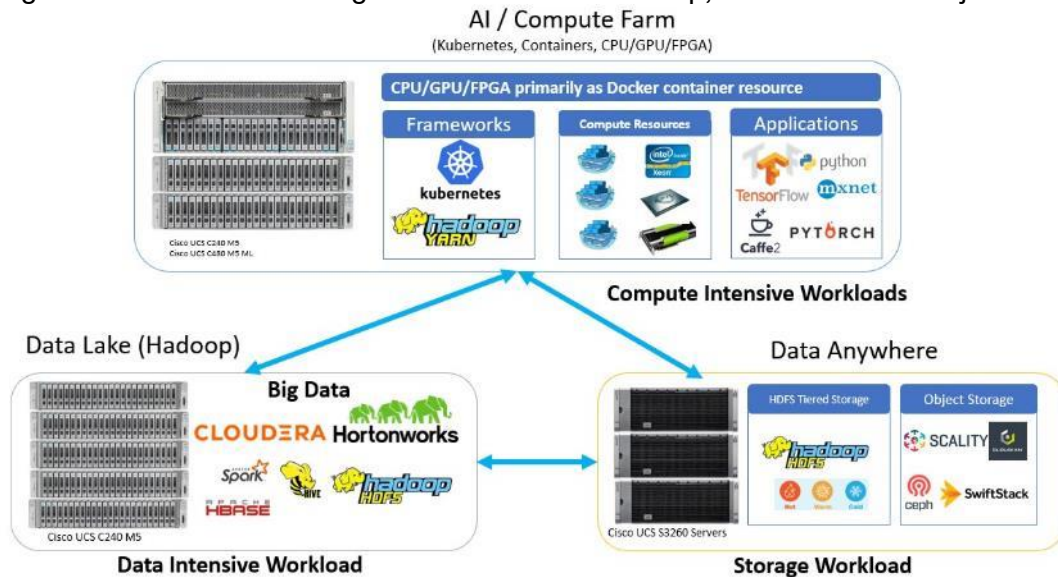


Cisco has developed numerous industry leading Cisco Validated Designs (reference architectures) in the area of Big Data (CVDs with Cloudera, Hortonworks and MapR), compute farm with Kubernetes (CVD with Red Hat OpenShift) and Object store (Scality, SwiftStack, Cloudian, and others).

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

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

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



This architecture can start from a single rack and scale to thousands of nodes with a single pane of glass management with Cisco Application Centric Infrastructure (ACI). More details can be found in [Scaling the Solution for Cisco Data Intelligence Platform](#).

Modernizing Hadoop with NVMe

NVMe (Non-Volatile Memory express) is a host controller interface and storage protocol created to accelerate the transfer of data and low latency by reducing IO bottleneck to bring performance improvement compare to its predecessors SAS or SATA. This characteristics of NVMe make them perfect candidate to manage huge data streams in parallel at same time minimize latency for data intensive workloads. NVMe helps achieve important business insights by unlocking parallel access to the storage from real time streams whether that is data sent from Internet of Things (IoT) to Big Data lake.

Enterprises built and utilized Big Data Analytics solutions architected with legacy hard disk drives (HDD) to collect, process and run analysis in real-time or batch processing on massive data to make agile business decisions. The cost advantage of traditional HDD over high-performance flash drives and additional speed, efficiency and reduced latency was not perceived as valuable since majority of analytics was batch processed. Technology such as machine learning, IoT devices, autonomous vehicle etc. where data collection and analysis from various sensors and devices needs to be processed in real-time and store in the data lake. As the quantity of the data grows more and more applications and services will be drawn where the data is stored which requires modernizing existing Hadoop architecture.

Traditionally, Hadoop HDFS and YARN temporary files are stored on HDD hence creating contention for running applications and MapReduce or Spark jobs on large datasets creating overlap on multiple data flows generated by YARN. In order to cater growing datasets requirement in enterprises and achieve better performance, this solution separates temporary data for MapReduce or even Spark and MMap files for Spark which is very IO intensive on NVMe from HDFS data.

Reference Architecture

Table 1 lists the reference architecture configuration details for the data lake. AI/ML components of the data lake and tiered storage are explained in detailed in the [Cisco Data Intelligence Platform with Hortonworks Data Platform 3.1 and Cloudera Data Science Workbench 1.5](#) published CVD.

Data Lake Reference Architecture

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

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

	Performance
Servers	28 x Cisco UCS C240 M5 Rack Servers with small-form-factor (SFF) drives
CPU	2 x 2 nd Gen Intel® Xeon® Scalable 6230 processors (2 x 20 cores, at 2.1 GHz)
Memory	12 x 32GB DDR4 (384 GB)
Boot	M.2 with 2 x 240-GB SSDs
Storage	24 x 2.4TB 10K rpm SFF SAS HDDs and 2 x 8TB Intel P4500 NVMe High Performance Value Endurance
Virtual interface card (VIC)	40 Gigabit Ethernet (Cisco UCS VIC 1387 or Cisco UCS VIC 1497)
Storage controller	Cisco 12-Gbps SAS modular RAID controller with 4-GB flash-based write cache (FBWC)

	Performance
Network connectivity	Cisco UCS 6332 Fabric Interconnect



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

As illustrated in Figure 3, a 28-node cluster with Rack#1 hosting 16 Cisco UCS C240 M5 server and Rack#2 hosting 12 Cisco UCS C240 M5 server. Each link in the figure represents a 40 Gigabit Ethernet link from each of the sixteen servers directly connected to a Fabric Interconnect. Every server is connected to both Fabric Interconnects.

Optionally a 30-node starter cluster with Rack#1 hosting 16 Cisco UCS C240 M5 servers and Rack#2 hosting six Cisco UCS 240 M5 servers for AI/Compute workload and four Cisco UCS S3260 Storage server for tiered storage as shown in Figure 4.

Figure 3 8x Cisco UCS S3260 Chassis with 2 Server Nodes Each (Object Storage Nodes)

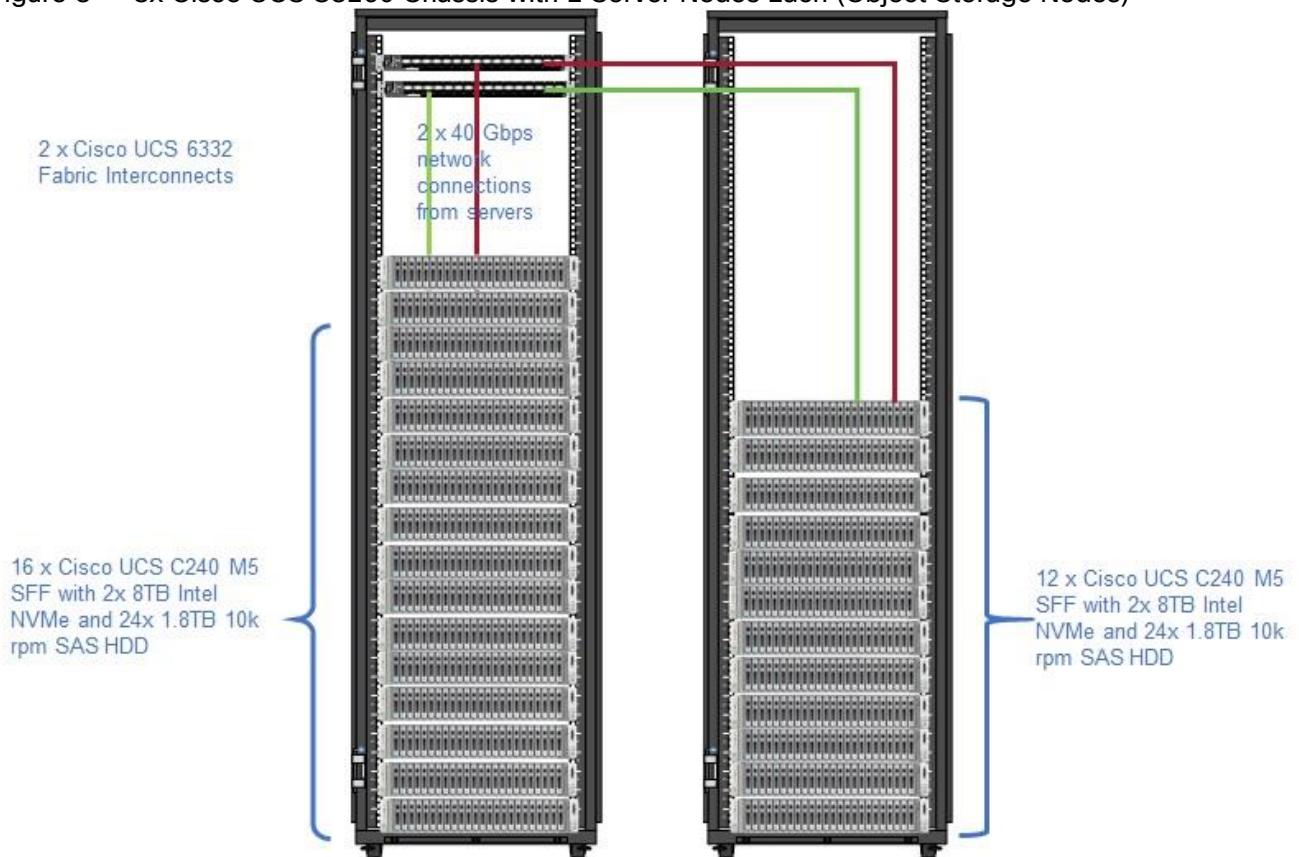
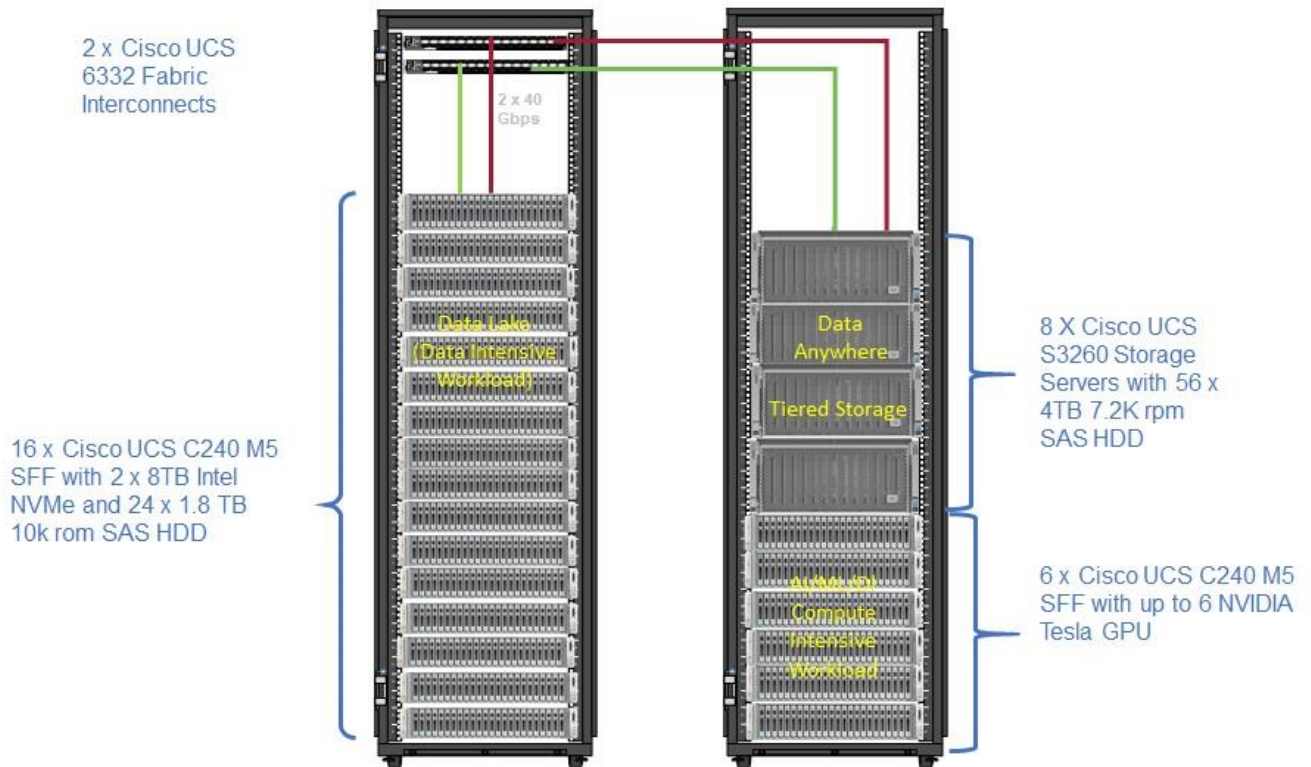


Figure 4 Cisco Data Intelligence Platform with Hortonworks – 30 Node Configuration (Optional)



An alternate configuration for cases where more GPU capacity is needed. Four of the Cisco UCS C240 M5 servers from the previous configuration in 0 are replaced with Cisco UCS C480 M5 ML M5 server which support up to eight V100 MXM GPUs.



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



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

Technology Overview

Cisco UCS Integrated Infrastructure for Big Data and Analytics

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

Cisco UCS Manager

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

Key Features

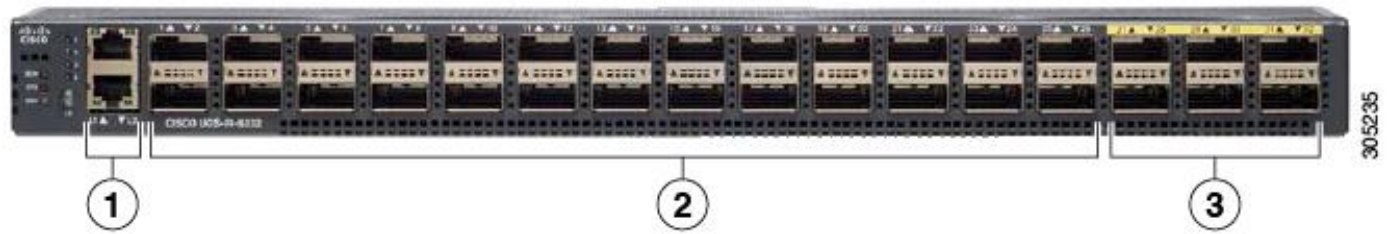
- Supports Cisco UCS B-Series Blade and Cisco UCS C-Series Rack Servers, the Cisco UCS C3260 storage server, Cisco UCS Mini, and the Cisco HyperFlex hyperconverged infrastructure.
- Programmatically controls server, network, and storage resources, with a unified, policy-driven management, so they can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

Cisco UCS 6300 Series Fabric Interconnects

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

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

**Figure 5 Cisco UCS 6332 UP 32 -Port Fabric Interconnect
FI 6332
(Front view)**



Cisco UCS C-Series Rack-Mount Servers

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

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

Cisco UCS C240 M5 Rack-Mount Server

The Cisco UCS C240 M5 Rack-Mount Server (Figure 6) is a 2-socket, 2-Rack-Unit (2RU) rack server offering industry-leading performance and expandability. It supports a wide range of storage and I/O-intensive infrastructure workloads, from big data and analytics to collaboration. Cisco UCS C-Series Rack Servers can be deployed as standalone servers or as part of a Cisco Unified Computing System (Cisco UCS) managed environment to take advantage of Cisco’s standards-based unified computing innovations that help reduce customers’ Total Cost of Ownership (TCO) and increase their business agility.

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

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

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

Figure 6 Cisco UCS C240 M5 Rack-Mount Server



Cisco UCS Virtual Interface Cards (VICs)

Cisco UCS Virtual Interface Cards (VIC) are unique to Cisco. Cisco UCS Virtual Interface Cards incorporate next-generation converged network adapter (CNA) technology from Cisco which reduces the number of network adapters, cables, and switches needed and radically simplifies the network, reducing complexity. Cisco VICs can support 256 Express (PCIe) virtual devices, either virtual Network Interface Cards (vNICs) or virtual Host Bus Adapters (vHBAs), with a high rate of I/O Operations Per Second (IOPS), support for lossless Ethernet, and 10/25/40/100-Gbps connection to servers. The PCIe Generation 3 x16 interface helps ensure optimal bandwidth to the host for network-intensive applications, with a redundant path to the fabric interconnect. Cisco VICs support NIC teaming with fabric failover for increased reliability and availability. In addition, it provides a policy-based, stateless, agile server infrastructure for your data center.

Cisco VIC 1497

The Cisco VIC 1497 (Figure 8) is a dual-port Quad Small Form-Factor (QSFP28) mLOM card designed for the M5 generation of Cisco UCS C-Series Rack Servers. The card supports 40/100-Gbps Ethernet or FCoE. The card can present PCIe standards-compliant interfaces to the host, and these can be dynamically configured as NICs or HBAs.

Figure 7 Cisco VIC 1497**Cisco UCS VIC 1387**

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

Figure 8 Cisco UCS VIC 1387

Intel P4510 Series Data Center NVMe

The Intel® SSD DC P4510 Series drives built on NVMe specification 1.2 PCIe with the increased density of Intel 64-layer 3D NAND and enhanced firmware features. The 8TB DC P4510 part of the reference architecture as shown in figure 3, is built to handle read-intensive workloads and beyond which supports optimized storage efficiency while enabling data center to do more per server and minimize service disruptions. The DC P4510 creates greater Quality of Service, bandwidth, and Performance. It significantly increases server agility and utilization and accelerates applications across a wide range of workloads to lead data centers through their evolving transformation.

Key Benefits

- Optimized for storage efficiency across a range of workloads
- Manageability to maximize IT efficiency

- Industry-leading reliability and security
- Designed for today's modern data centers

Hortonworks Data Platform

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

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

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

Apache Ambari

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

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

The following are the core benefits that Hadoop operators get with Ambari:

- Simplified Installation, Configuration and Management. Easily and efficiently create, manage and monitor clusters at scale. Takes the guesswork out of configuration with [Smart Configs](#) and Cluster Recommendations. Enables repeatable, automated cluster creation with [Ambari Blueprints](#).
- Centralized Security Setup. Reduce the complexity to administer and configure cluster security across the entire platform. Helps automate the setup and configuration of advanced cluster security capabilities such as Kerberos and [Apache Ranger](#).
- Full Visibility into Cluster Health. Ensure your cluster is healthy and available with a holistic approach to monitoring. Configures predefined alerts – based on operational best practices – for cluster monitoring. Captures and visualizes critical operational metrics – using [Grafana](#) – for analysis and troubleshooting. Integrated with [Hortonworks SmartSense](#) for proactive issue prevention and resolution.
- Highly Extensible and Customizable. Fit Hadoop seamlessly into your enterprise environment. Highly extensible with [Ambari Stacks](#) for bringing custom services under management, and with [Ambari Views](#) for customizing the Ambari Web UI.

HDP for Data Access

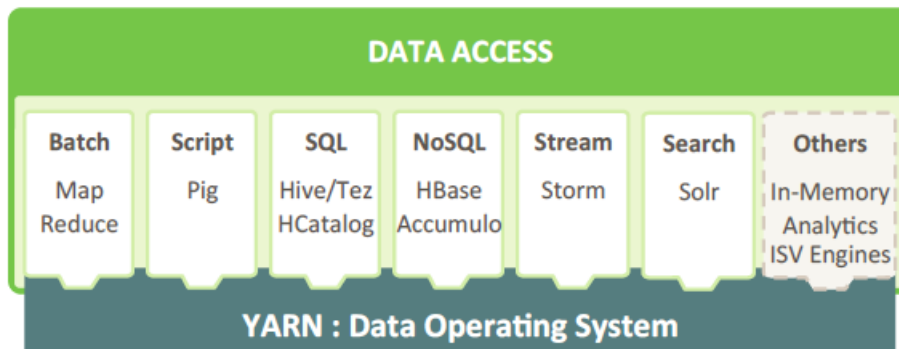
With YARN at its foundation, HDP provides a range of processing engines that allow users to interact with data in multiple and parallel ways, without the need to stand up individual clusters for each data set/application. Some applications require batch while others require interactive SQL or low-latency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

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

As shown in Table 2 , HDP Enterprise natively provides for the following data access types:

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

Table 2 YARN



Red Hat Ansible Automation

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



For more information about Ansible, go to:

<https://www.redhat.com/en/technologies/management/ansible>

Solution Design

Requirements

This CVD describes architecture and deployment procedures for Hortonworks 3.1.0 on a 28-node cluster based on Cisco UCS Integrated Infrastructure for Big Data and Analytics. The solution details how to configure Hortonworks 3.1.0 on the infrastructure and all its dependencies. In addition, it also details the configuration for Hortonworks Dataflow for various use cases.

The cluster configuration consists of the following:

- Two Cisco UCS 6332UP Fabric Interconnects
- 28 Cisco UCS C240 M5 Rack-Mount servers
- Two Cisco R42610 standard racks
- Four Vertical Power distribution units (PDUs) (Country Specific)

Physical Topology

Each rack consists of two vertical PDUs. The first rack consists of two Cisco UCS 6332UP Fabric Interconnects, 16 Cisco UCS C240 M5 Rack Servers connected to each of the vertical PDUs for redundancy; thereby, ensuring availability during power source failure. The second rack consists of 12 Cisco UCS C240 M5 Servers connected to each of the vertical PDUs for redundancy; thereby, ensuring availability during power source failure, like the first rack.

Port Configuration on Fabric Interconnect

Table 3 lists the port configuration on Cisco UCS FI 6332 Fabric Interconnect.

Table 3 Port Configuration on Fabric Interconnect

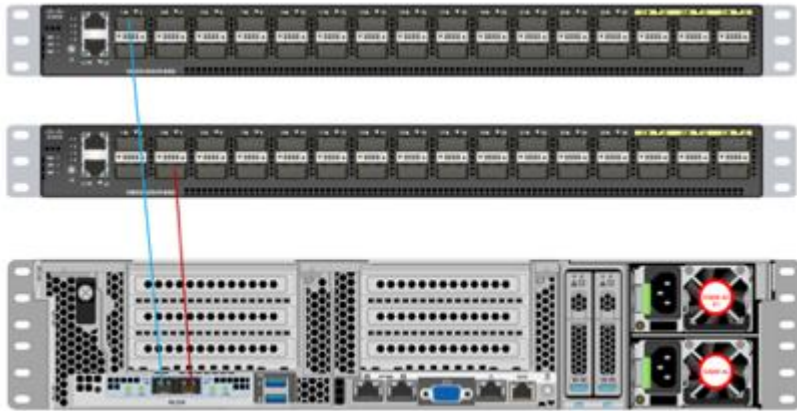
Port Type	Port Number
Server	1-26
Network	29-32

Server Configuration and Cabling for Cisco UCS C240 M5

The Cisco UCS C240 M5 rack server is equipped with 2 x Intel Xeon 2nd Gen Scalable Family Processor 6230 (2 x 20 cores, 2.1 GHz), 384 GB of memory, Cisco UCS Virtual Interface Card 1337, Cisco 12-Gbps SAS Modular Raid Controller with 4-GB FBWC, 24 x 2.4 TB 10K rpm SFF SAS HDDs and 2 x 8TB Cisco 2.5" U.2 8TB Intel P4500 NVMe High Perf. Value Endurance, M.2 with 2 x 240-GB SSDs for Boot.

Figure 9 illustrates the port connectivity between the Cisco UCS FI 6332 and Cisco UCS C240 M5 Rack Server. Twenty-two Cisco UCS C240 M5 servers are installed in this configuration.

Figure 9 Fabric topology for Cisco UCS C240 M5 Rack Server



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

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

Software Distributions and Versions

The software distributions required versions are listed in Table 4 .

Table 4 Software distribution and Version

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

Layer	Component	Version or Release
	Cisco UCS Manager	4.0(4c)
	HDP	3.1.0



The latest drivers can be downloaded from here:

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



Support for the Intel 2nd generation scalable family processor is added in UCSM version 4.0.4a.

Deployment Hardware and Software

Cisco Unified Computing System Configuration

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

Configure Cisco UCS Fabric Interconnect

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

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

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

```
#connect local-mgmt
```



```
#erase config
```



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

Configure Fabric Interconnects for a Cluster Setup

To configure the Cisco UCS Fabric Interconnects, follow this step:

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

Configure Fabric Interconnect A

To configure Fabric Interconnect A, follow these steps:

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

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

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

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

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

4. Enter the cluster name for the system name.
5. Enter the Mgmt0 IPv4 address.
6. Enter the Mgmt0 IPv4 netmask.
7. Enter the IPv4 address of the default gateway.
8. Enter the cluster IPv4 address.

To configure DNS, answer y.

9. Enter the DNS IPv4 address.

Answer y to set up the default domain name.

10. Enter the default domain name.

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

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

Configure Fabric Interconnect B

To configure Fabric Interconnect B, follow these steps:

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

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

2. Enter the admin password that was configured for the first Fabric Interconnect.
3. Enter the Mgmt0 IPv4 address.
4. Answer yes to save the configuration.
5. Wait for the login prompt to confirm that the configuration has been saved.

For more information about configuring Cisco UCS 6332 Series Fabric Interconnect, go to:

https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/ucs-manager/GUI-User-Guides/Getting-Started/4-0/b_UCSM_Getting_Started_Guide_4_0.html

Log into Cisco UCS Manager

To log into Cisco UCS Manager, follow these steps:

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

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

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



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

Configure Cisco UCS Manager

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

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

12. Configure VLAN

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

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

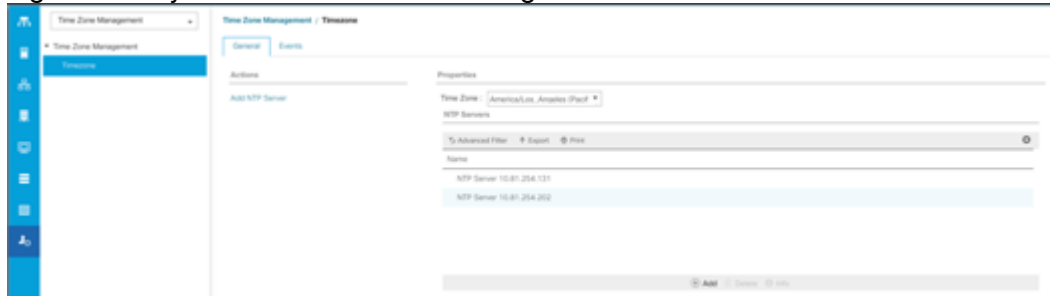
Details for each step are discussed in the following sections.

Synchronize Cisco UCSM to NTP

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

1. In Cisco UCS Manager, in the navigation pane, click the Admin tab.
2. Select All > Time zone Management.
3. In the Properties pane, select the appropriate time zone in the Time zone menu.
4. Click Save Changes and then click OK.
5. Click Add NTP Server.
6. Enter the NTP server IP address and click OK.
7. Click OK to finish.
8. Click Save Changes.

Figure 10 Synchronize Cisco UCS Manager to NTP



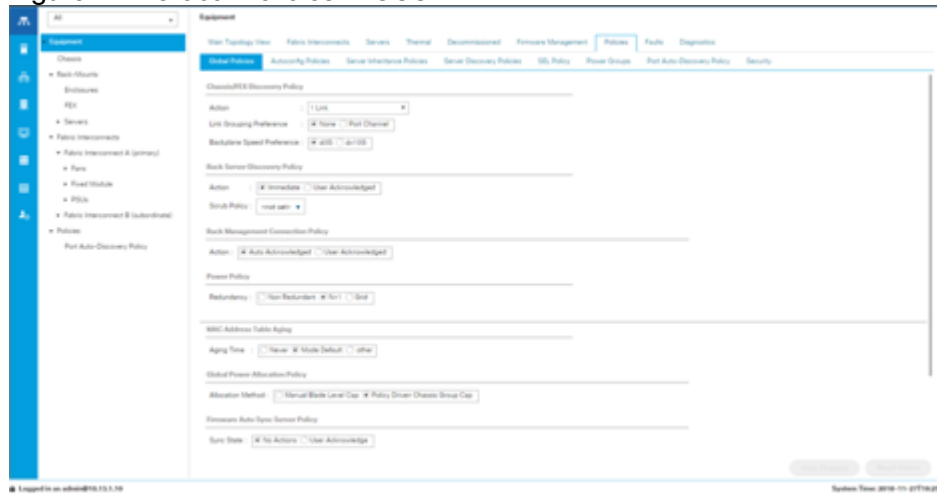
Configure Global Policies

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

To configure the global policies, follow this step:

1. In Cisco UCS Manager; Go to Equipment > Policies (right pane) > Global Policies as shown in Figure 11.

Figure 11 Global Policies in UCSM



Configure Server Ports

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

1. Go to Equipment > Fabric Interconnects > Fabric Interconnect A > Fixed Module > Ethernet Ports.
2. Select the ports (for this solution ports are 1-28) which are connected to the Cisco UCS VIC 1387 on Cisco UCS C240 M5 rack server.
3. Right-click and select Configure as Server Port.

Figure 12 Configure Server Port on Cisco UCS Manager Fabric Interconnect for Server/Chassis Discovery

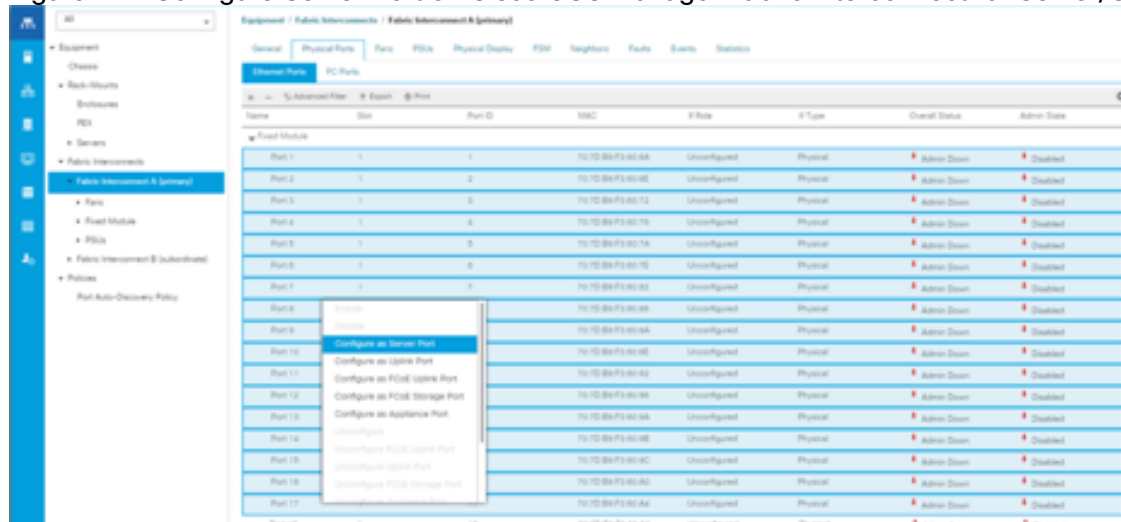
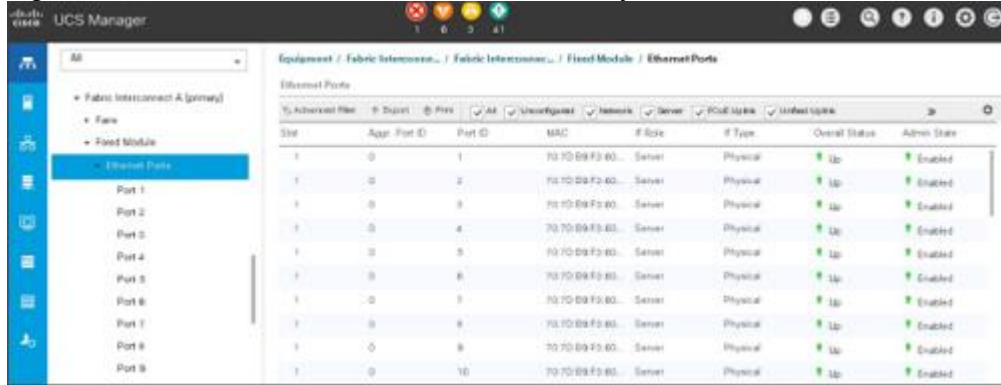


Figure 13 Ports Status after the Server Discovery



Configure Uplink Ports

Configure Network Ports to connect to the datacenter network switch.

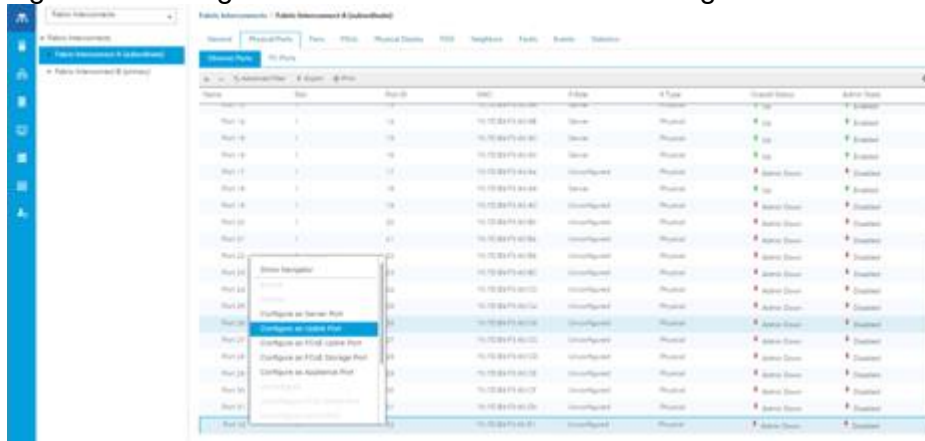


In our solution study we connected to Nexus 9000 series switch.

To configure Network ports, follow these steps:

1. Go to Equipment > Fabric Interconnects > Fabric Interconnect A > Fixed Module > Ethernet Ports.
2. Select the ports (for this solution ports are 29-32) which are connected to the Cisco Nexus 9000 series switch for northbound network connectivity.
3. Right-click and select Configure as Network Port.

Figure 14 Configure Network Port on Cisco UCS Manager Fabric Interconnect



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

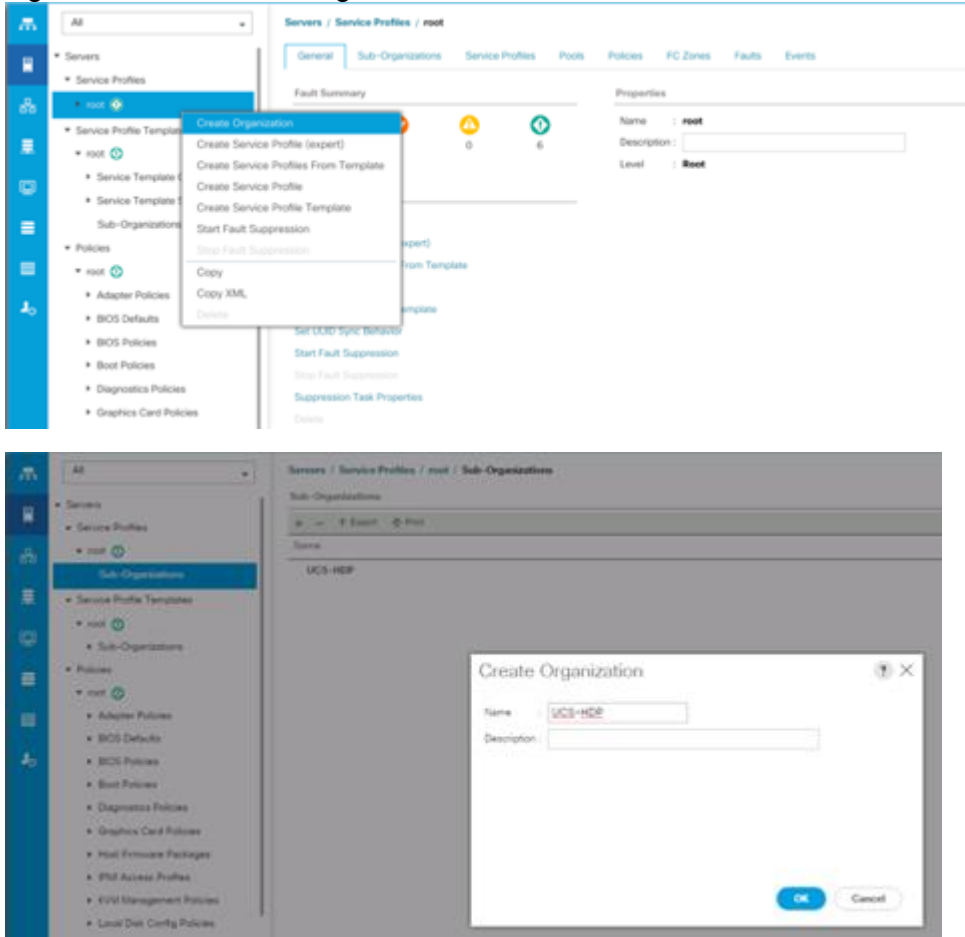
Create New Organization

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

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

2. Select root > Sub-Organization.
3. Right-click Sub-Organization.
4. Enter the name of the Organization.
5. Click OK.

Figure 15 Create New Organization



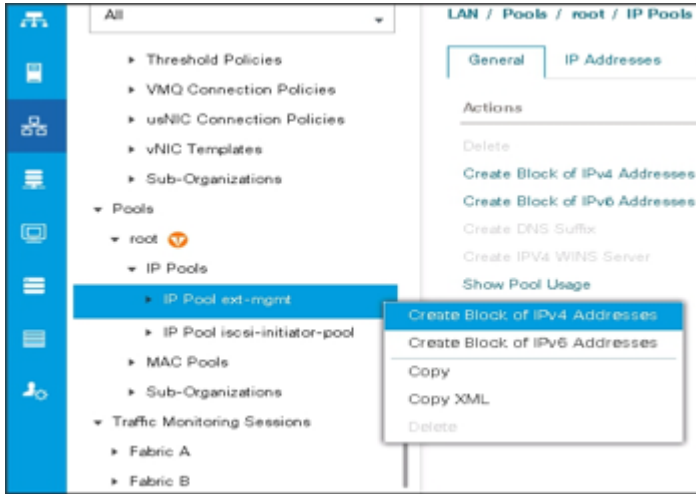
Cisco UCS Manager pools and policies required for this solution were created under new “UCS-HDP” Organization created.

Configure IP, UUID, Server and MAC Pools

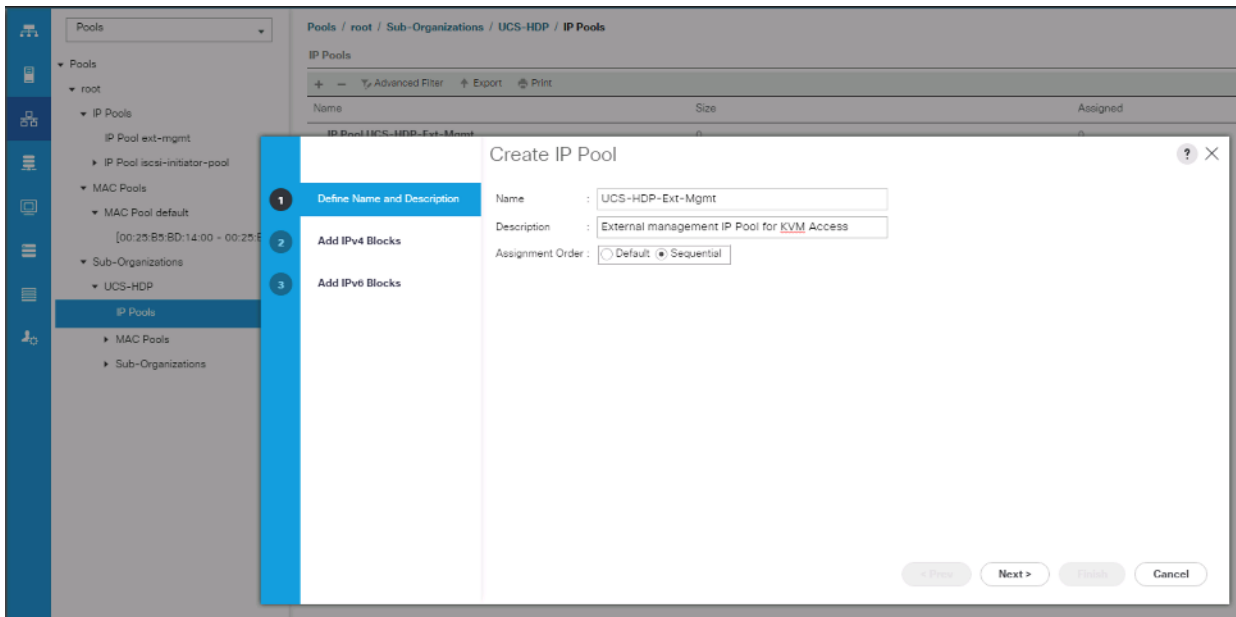
IP Pool Creation

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

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



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



4. Click Add IPv4 Block.

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



UUID Suffix Pool Creation

To configure the necessary universally unique identifier (UUID) suffix pool for the Cisco UCS environment, follow these steps:

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

Figure 16 UUID Suffix Pool Creation

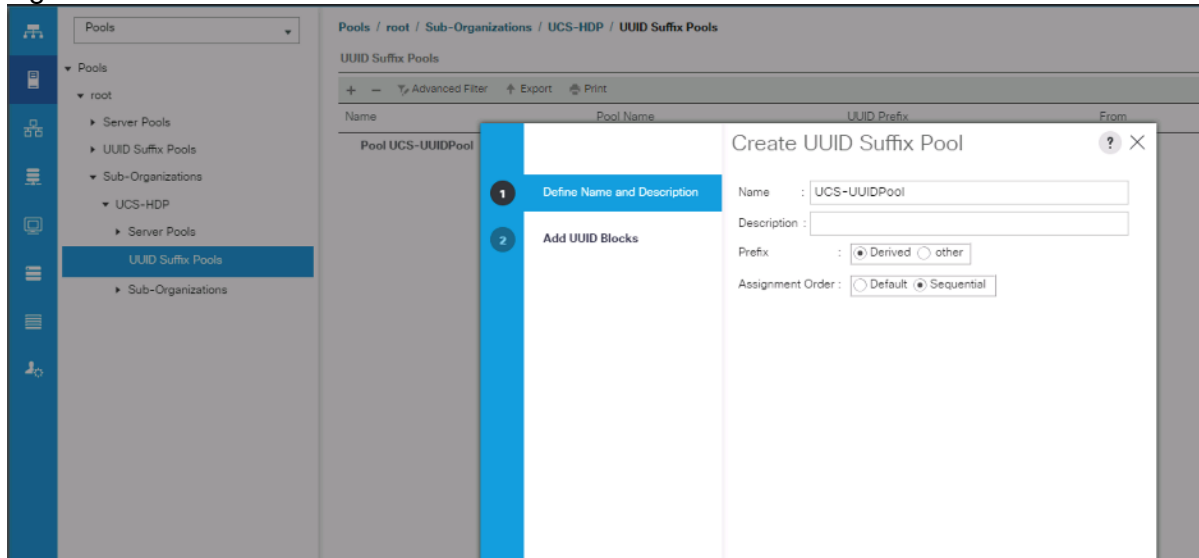
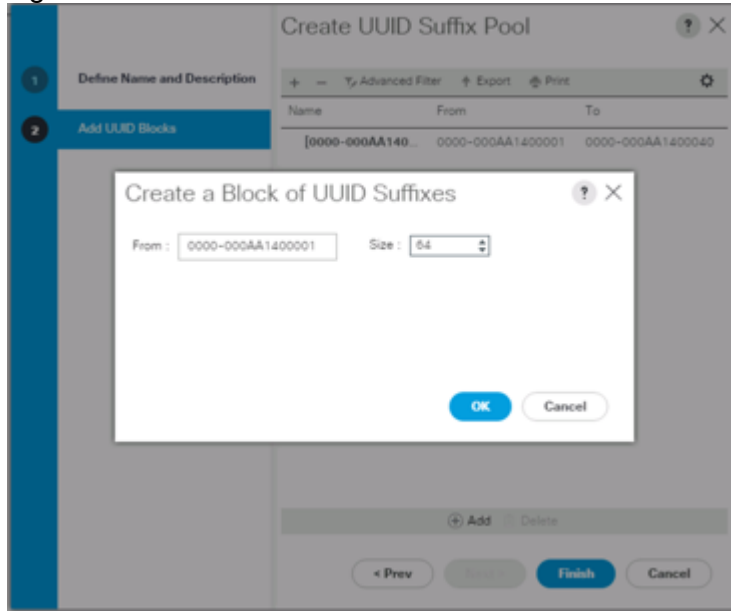


Figure 17 Create a Block of UUID Suffixes



Server Pool Creation

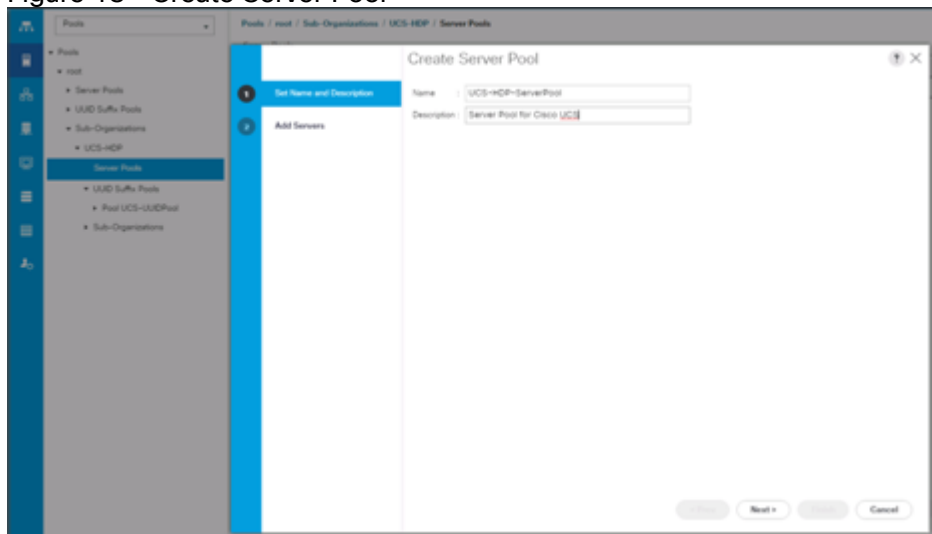
To configure the necessary server pool for the Cisco UCS environment, follow these steps:



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

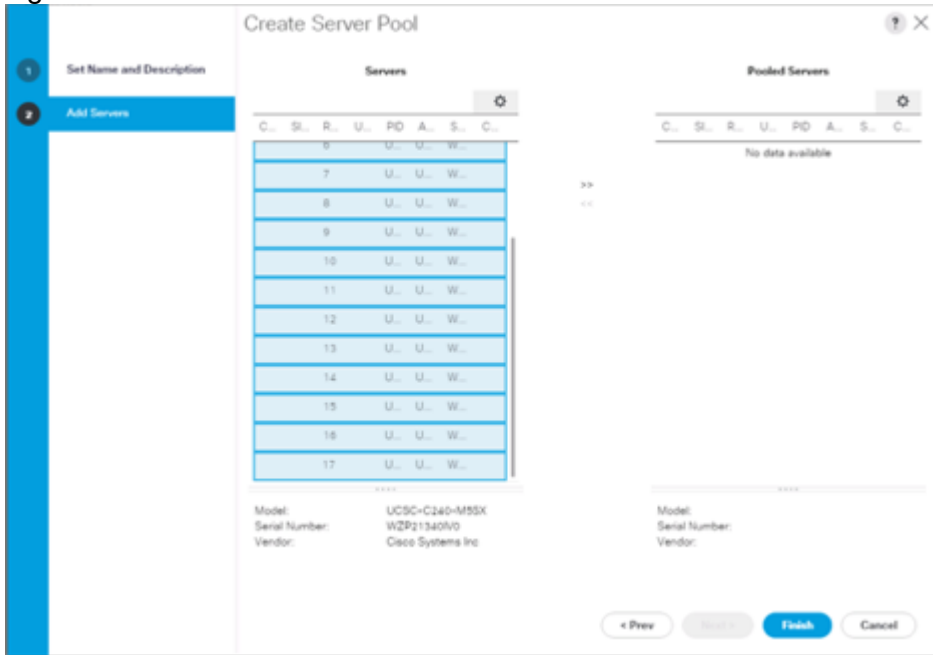
1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Pools > root > Sub-Organization > UCS-HDP> right-click Server Pools > Select Create Server Pool.
3. Enter name of the server pool.
4. Optional: Enter a description for the server pool then click Next.

Figure 18 Create Server Pool

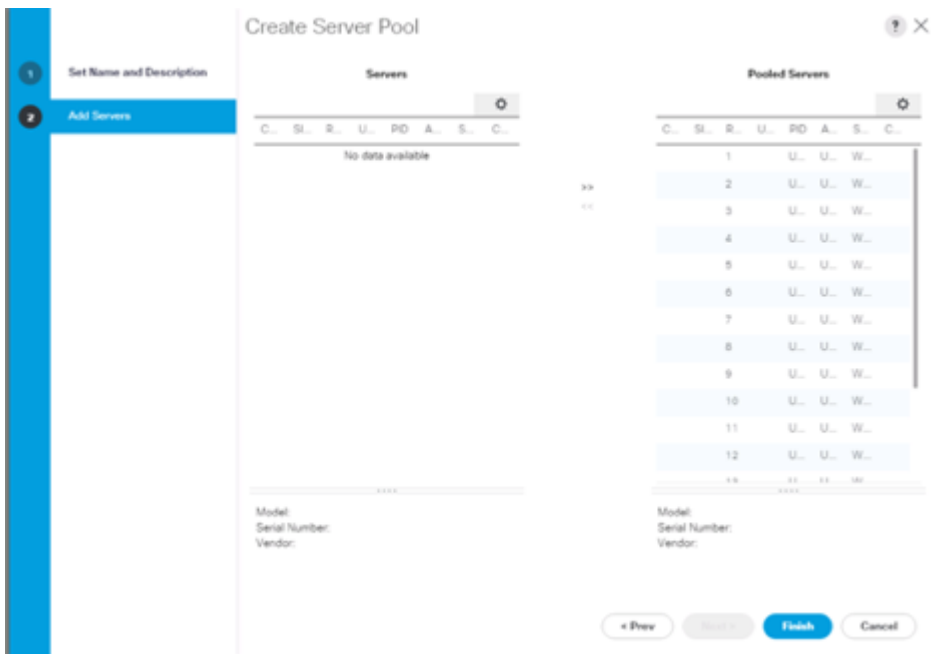


5. Select servers to be used for the deployment and click > to add them to the server pool. In our case we added thirty servers in this server pool.
6. Click Finish and then click OK.

Figure 19 Add Server in the Server Pool



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

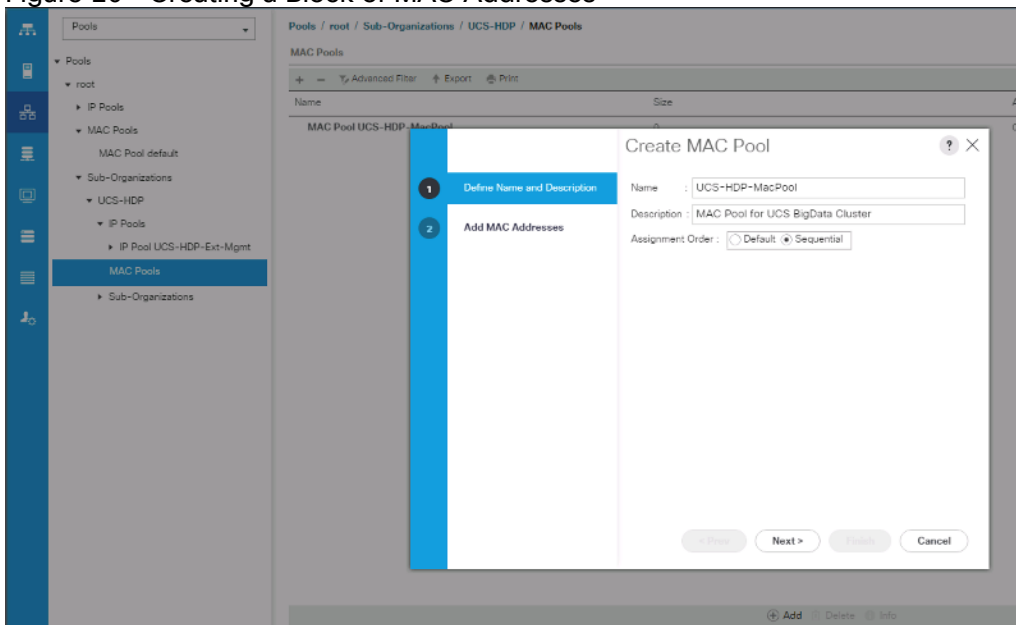


MAC Pool Creation

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

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

Figure 20 Creating a Block of MAC Addresses

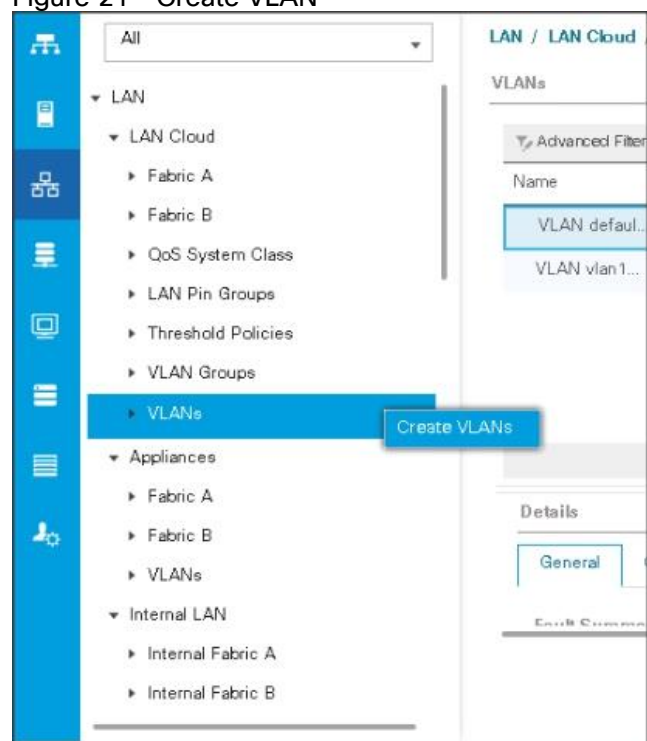


Configure VLAN

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

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

Figure 21 Create VLAN



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

Figure 22 Create VLANs

Create VLANs ? X

VLAN Name/Prefix :

Multicast Policy Name : [Create Multicast Policy](#)

Common/Global Fabric A Fabric B Both Fabrics Configured Differently

You are creating global VLANs that map to the same VLAN IDs in all available fabrics. Enter the range of VLAN IDs (e.g. "2009-2019", "29,35,40-45", "23", "23,34-45")

VLAN IDs :

Sharing Type : None Primary Isolated Community

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

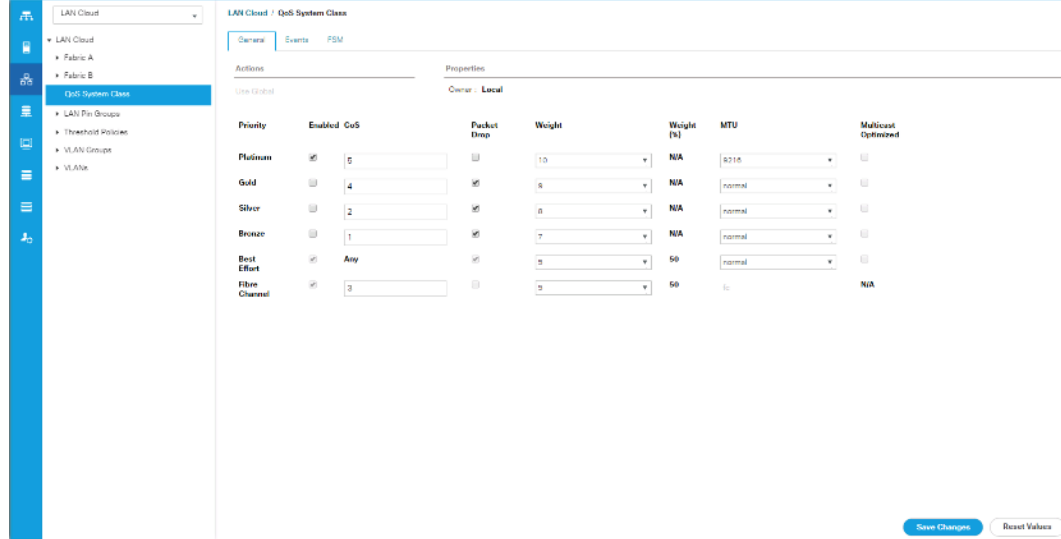
To configure jumbo frames and enable quality of service in the Cisco UCS fabric, follow these steps:

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



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

Figure 23 Configure System Class QoS on Cisco UCS Fabric Interconnects

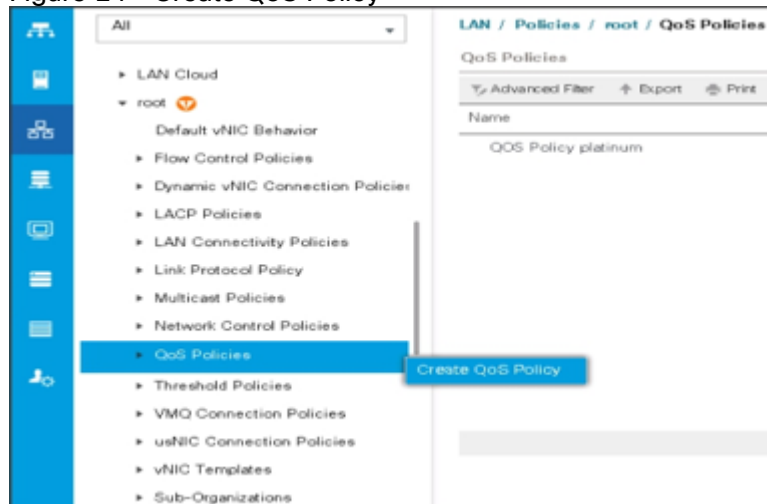


Create QoS Policies

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

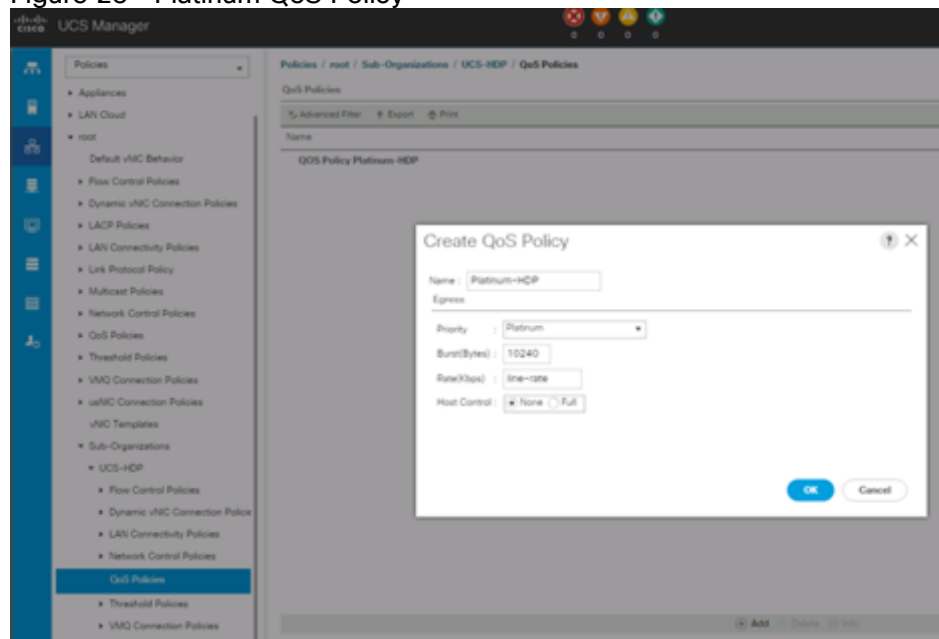
1. Select LAN tab in the left pane in the Cisco UCS Manager GUI.
2. Select LAN > Policies > root > UCS-HDP > QoS Policies.
3. Right-click QoS Policies.
4. Select Create QoS Policy.

Figure 24 Create QoS Policy



We created a Platinum class QoS policy for this solution.

Figure 25 Platinum QoS Policy

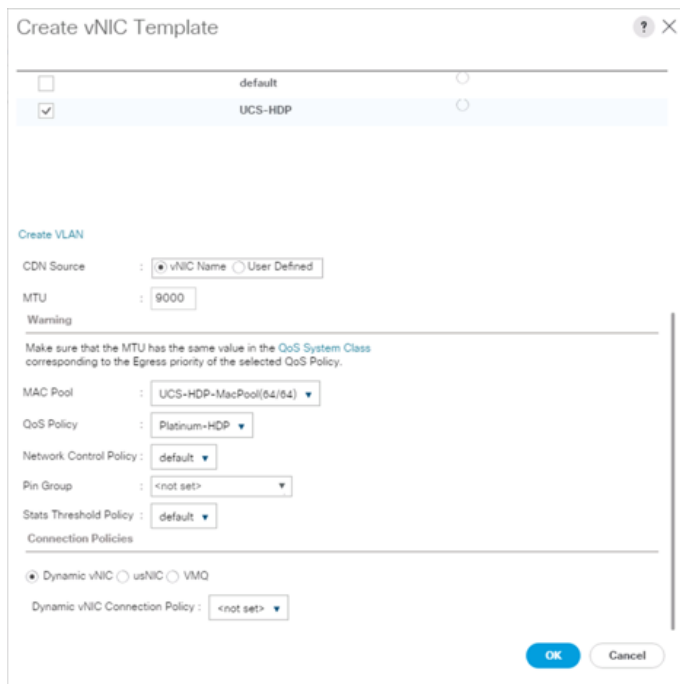
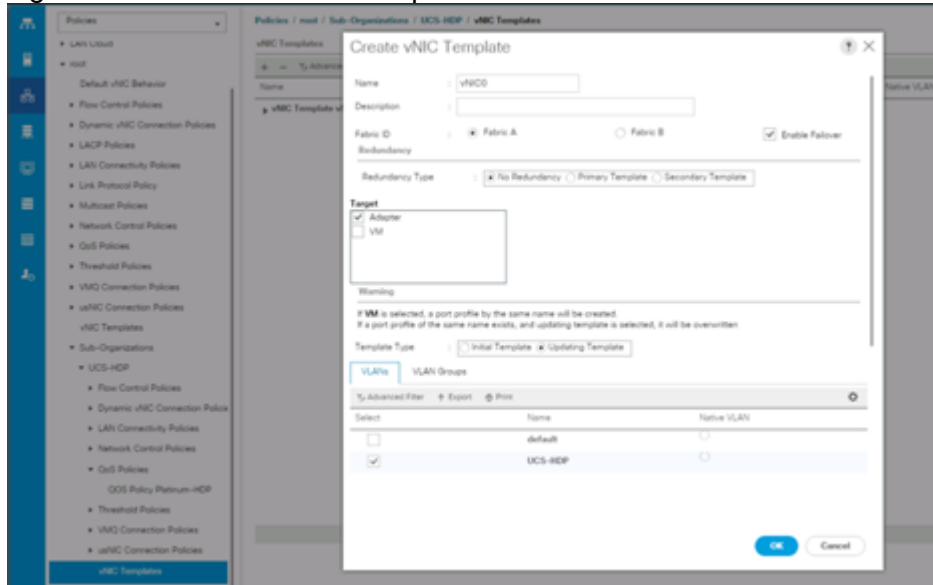


Create vNIC Templates

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

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

Figure 26 Create the vNIC Template



Create Host Firmware Package

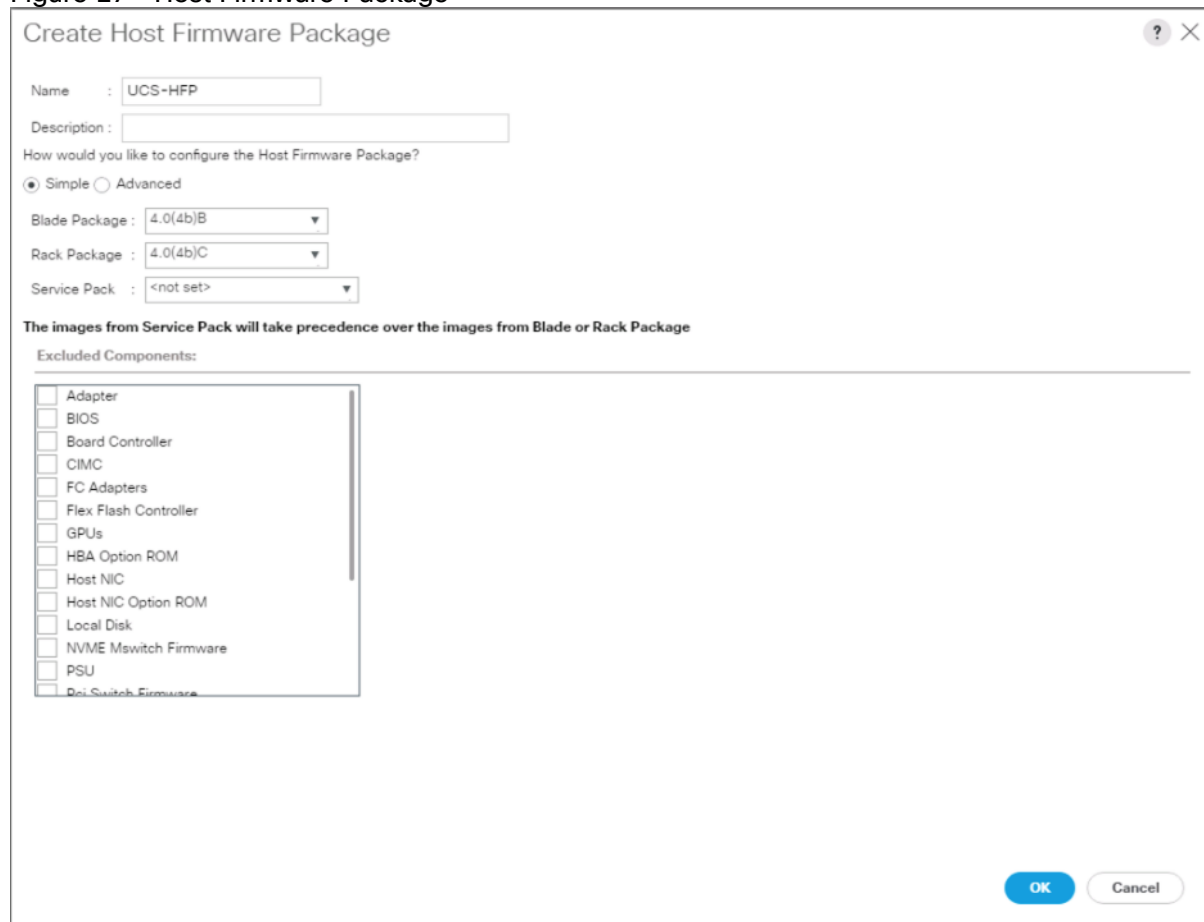
Firmware management policies allow the administrator to select the corresponding packages for a given server configuration. These policies often include packages for adapter, BIOS, board controller, FC adapters, host bus adapter (HBA) option ROM, and storage controller properties.

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

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

3. Right-click Host Firmware Packages.
4. Select Create Host Firmware Package.
5. Enter name of the host firmware package.
6. Leave Simple selected.
7. Select the version.
8. Click OK to create the host firmware package.

Figure 27 Host Firmware Package



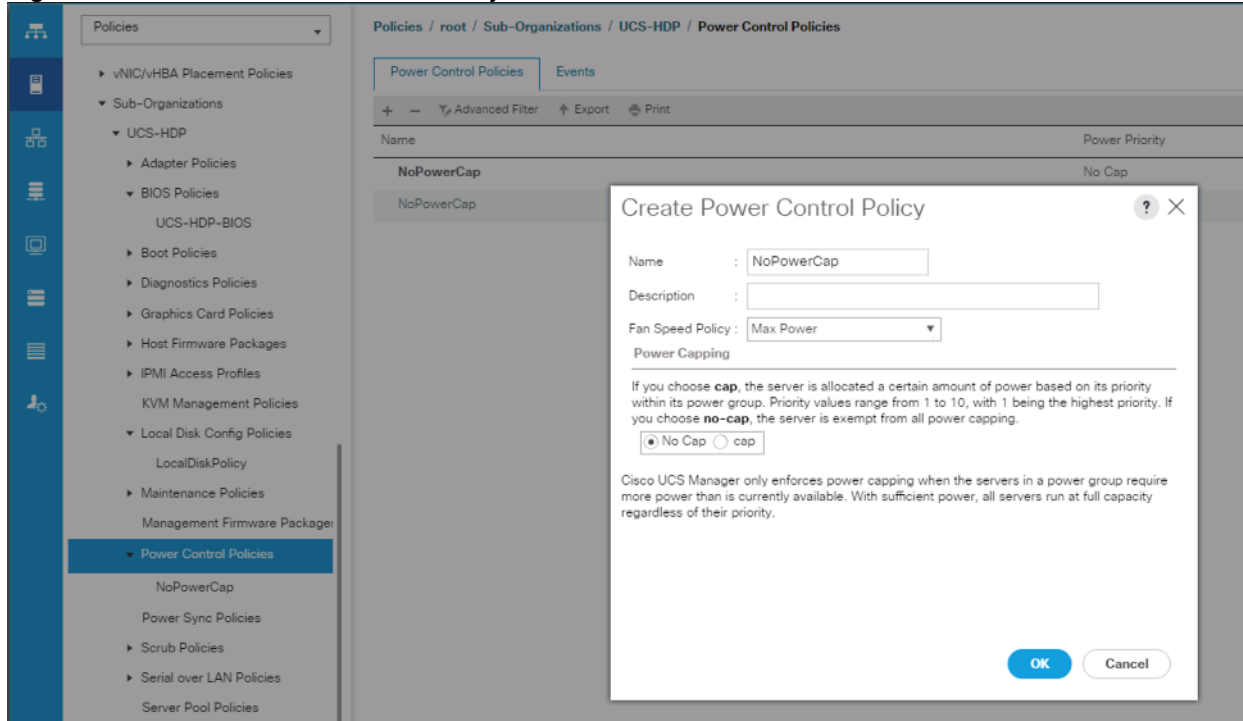
Create Power Control Policy

To create a power control policy for the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root > Sub-Organization > UCS-HDP > Power Control Policies.
3. Right-click Power Control Policies.
4. Select Create Power Control Policy.

5. Select Fan Speed Policy as “Max Power”.
6. Enter NoPowerCap as the power control policy name.
7. Change the power capping setting to No Cap.
8. Click OK to create the power control policy.

Figure 28 Create Power Control Policy

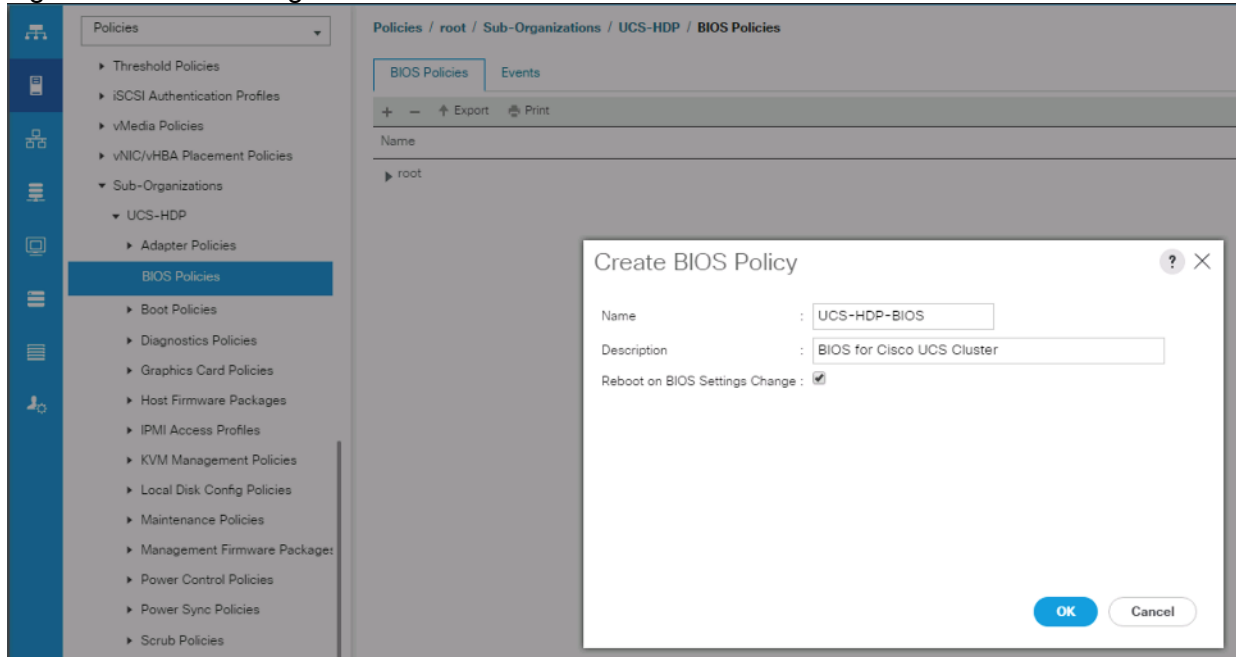


Create Server BIOS Policy

To create a server BIOS policy for the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root > Sub-Organization > UCS-HDP > BIOS Policies.
3. Right-click BIOS Policies.
4. Select Create BIOS Policy.
5. Enter C240M5-BIOS as the BIOS policy name.

Figure 29 BIOS Configuration



Policies / root / Sub-Organizations / TPC-BDA / BIOS Policies / BDA-BIOS

Main | **Advanced** | Boot Options | Server Management | Events

Processor | Intel Directed IO | RAS Memory | Serial Port | USB | PCI | QPI | LOM and PCIe Slots | Trusted Platform | Graphics Configuration

Advanced Filter | Export | Print

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

Channel Interleaving	Auto
IMC Inteleave	Platform Default
Memory Interleaving	Platform Default
Rank Interleaving	Platform Default
Sub NUMA Clustering	Platform Default
Local X2 Apic	Platform Default
Max Variable MTRR Setting	Platform Default
P STATE Coordination	HW ALL
Package C State Limit	Platform Default
Autonomous Core C-state	Platform Default
Processor C State	Disabled
Processor C1E	Disabled
Processor C3 Report	Disabled
Processor C6 Report	Disabled
Processor C7 Report	Disabled
Processor CMC1	Platform Default
Power Technology	Performance

Advanced Filter ↑ Export Print



BIOS Setting	Value
Energy Performance	Performance
ProcessorEppProfile	Performance
Adjacent Cache Line Prefetcher	Enabled
DCU IP Prefetcher	Enabled
DCU Streamer Prefetch	Enabled
Hardware Prefetcher	Enabled
UPI Prefetch	Enabled
LLC Prefetch	Enabled
XPT Prefetch	Enabled
Core Performance Boost	Platform Default
Downcore control	Platform Default
Global C-state Control	Platform Default
L1 Stream HW Prefetcher	Platform Default
L2 Stream HW Prefetcher	Platform Default
Determinism Slider	Platform Default
IOMMU	Platform Default
Bank Group Swap	Platform Default
Bank Group Swap	Platform Default
Chipselect Interleaving	Platform Default
Configurable TDP Control	Platform Default
AMD Memory Interleaving	Platform Default
AMD Memory Interleaving Size	Platform Default
SMEE	Platform Default
SMT Mode	Platform Default
SVM Mode	Platform Default
Demand Scrub	Enabled
Patrol Scrub	Enabled
Workload Configuration	Platform Default

Policies / root / Sub-Organizations / TPC-BDA / BIOS Policies / BDA-BIOS

Main | **Advanced** | Boot Options | Server Management | Events

Processor | Intel Directed IO | **RAS Memory** | Serial Port | USB | PCI | QPI | LOM and PCIe Slots | Trusted Platform | Graphics Configuration

Advanced Filter | Export | Print

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



Cisco UCS M5 Server Performance Tuning guide:

https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-b-series-blade-servers/whitepaper_c11-740098.pdf



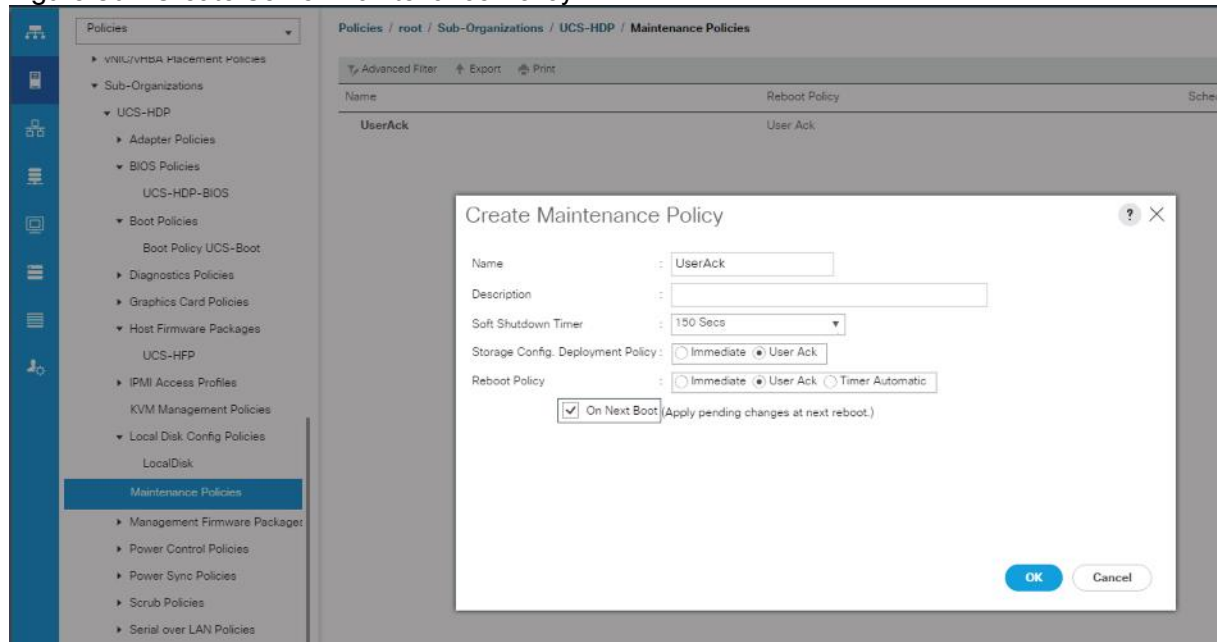
BIOS settings can have a significant performance impact, depending on the workload and the applications. The BIOS settings listed in this section is for configurations optimized for best performance which can be adjusted based on the application, performance, and energy efficiency requirements.

Configure Maintenance Policy

To update the default Maintenance Policy, follow these steps:

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

Figure 30 Create Server Maintenance Policy

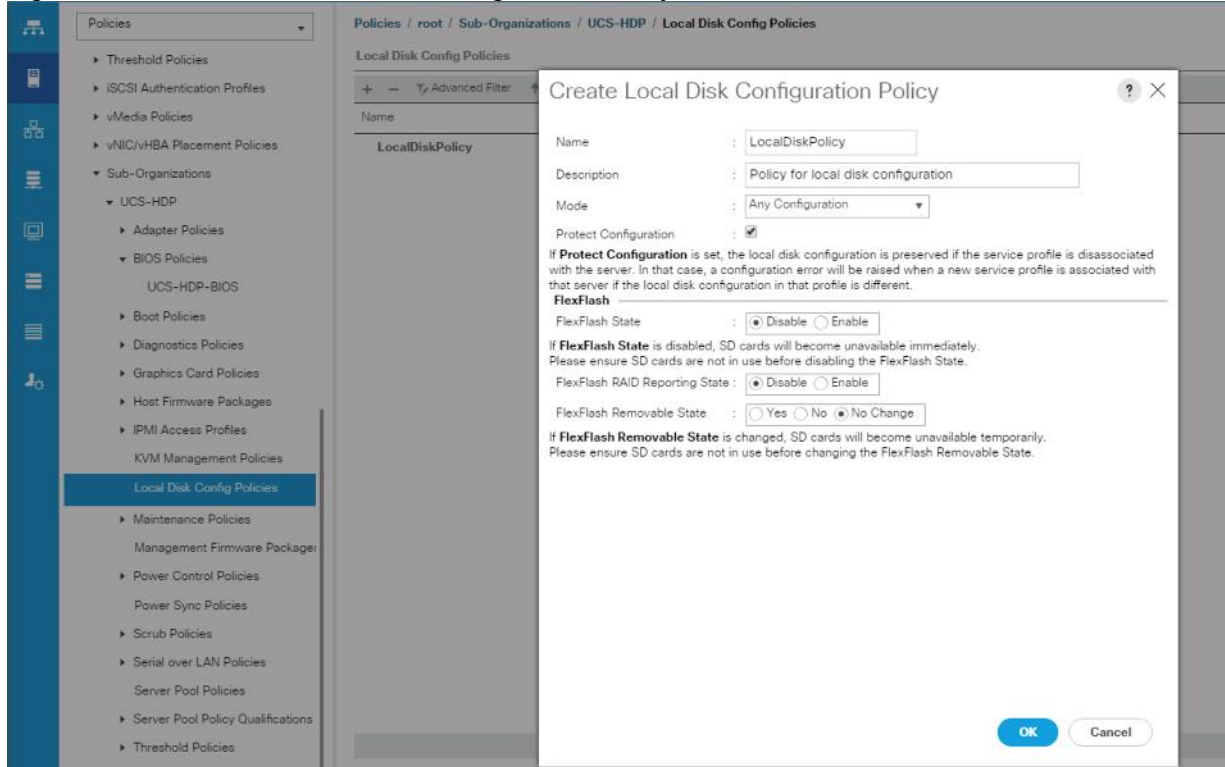


Create the Local Disk Configuration Policy

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

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

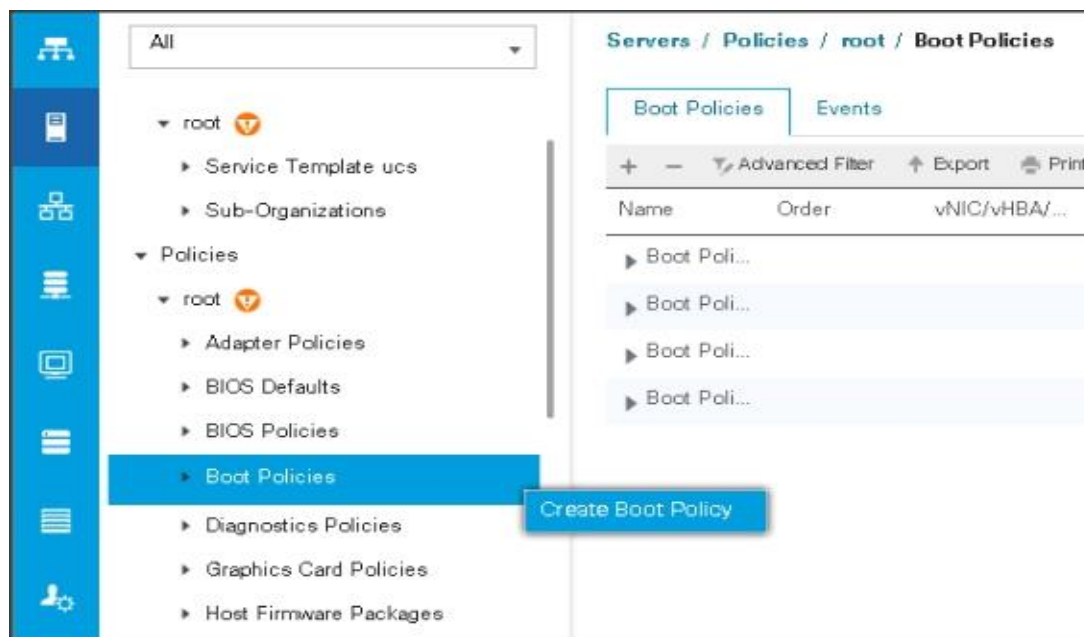
Figure 31 Create the Local Disk Configuration Policy



Create Boot Policy

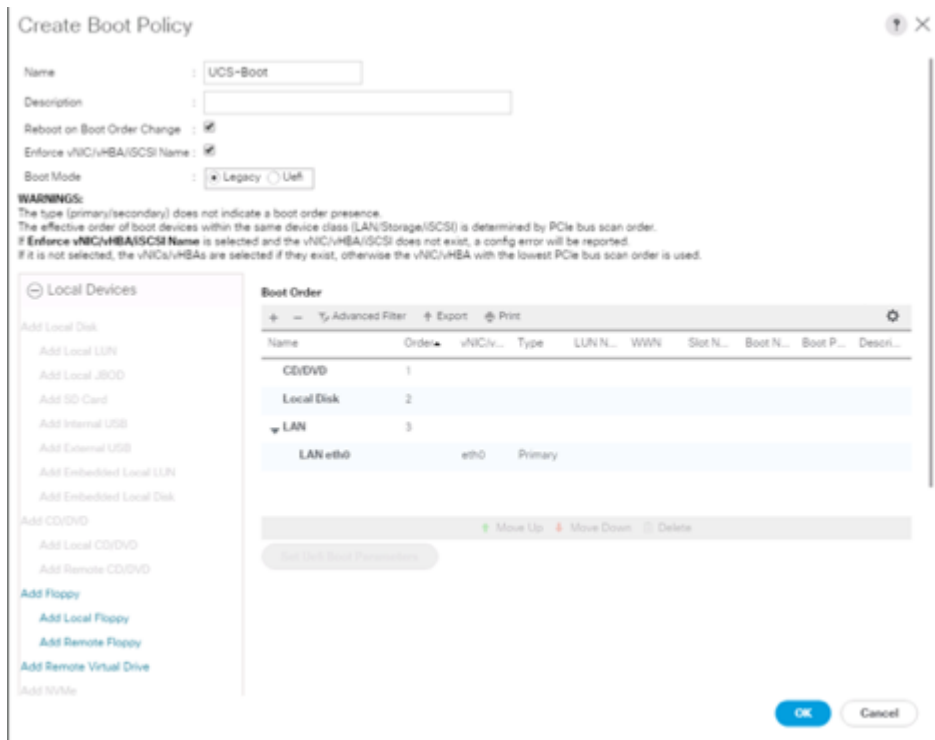
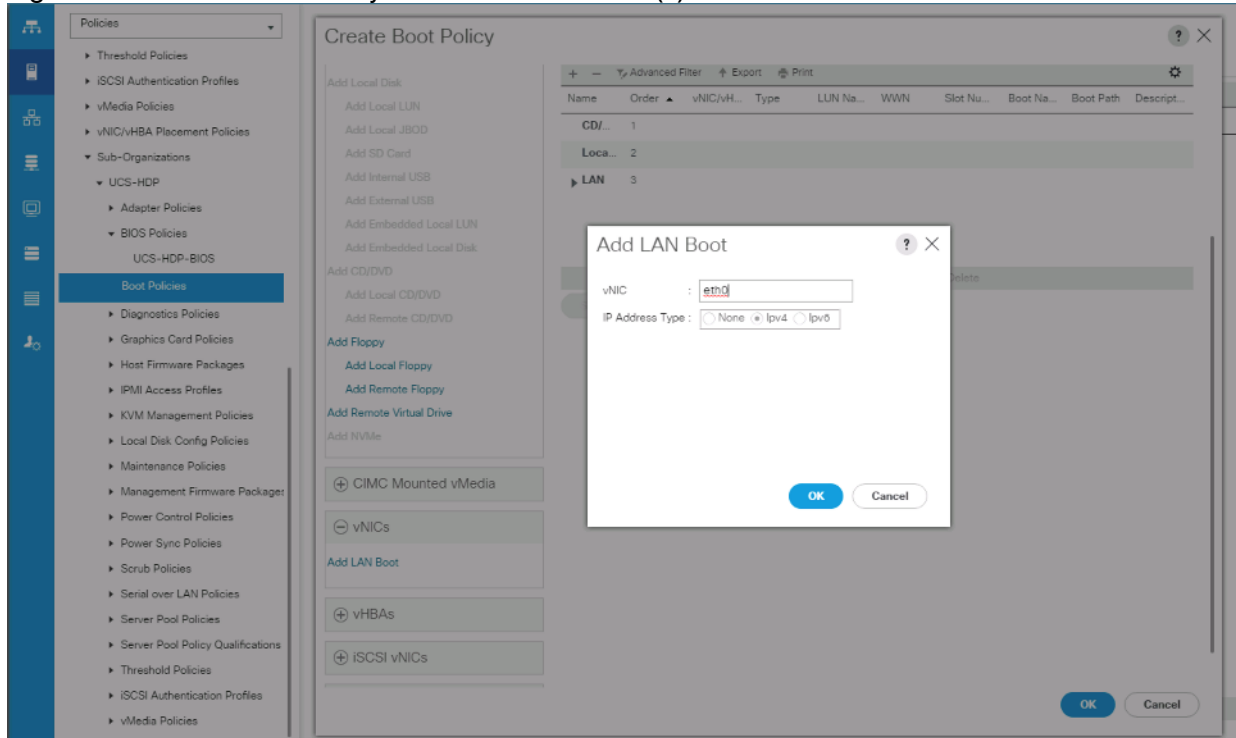
To create boot policies within the Cisco UCS Manager GUI, follow these steps:

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



5. Enter ucs for the boot policy name.
6. (Optional) enter a description for the boot policy.
7. Keep the Reboot on Boot Order Change check box unchecked.
8. Keep Enforce vNIC/vHBA/iSCSI Name check box checked.
9. Keep Boot Mode Default (Legacy).
10. Expand Local Devices > Add CD/DVD and select Add Local CD/DVD.
11. Expand Local Devices and select Add Local Disk.
12. Expand vNICs and select Add LAN Boot and enter eth0.
13. Click OK to add the Boot Policy.
14. Click OK.

Figure 32 Create Boot Policy for Cisco UCS Server(s)



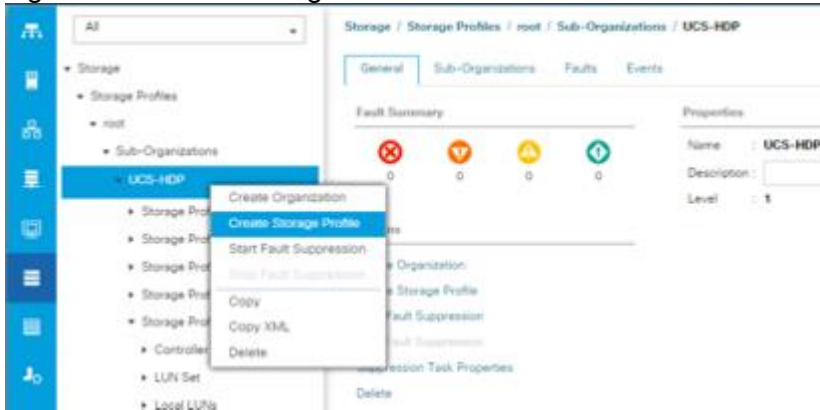
Create Storage Profile for Individual RAID0

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

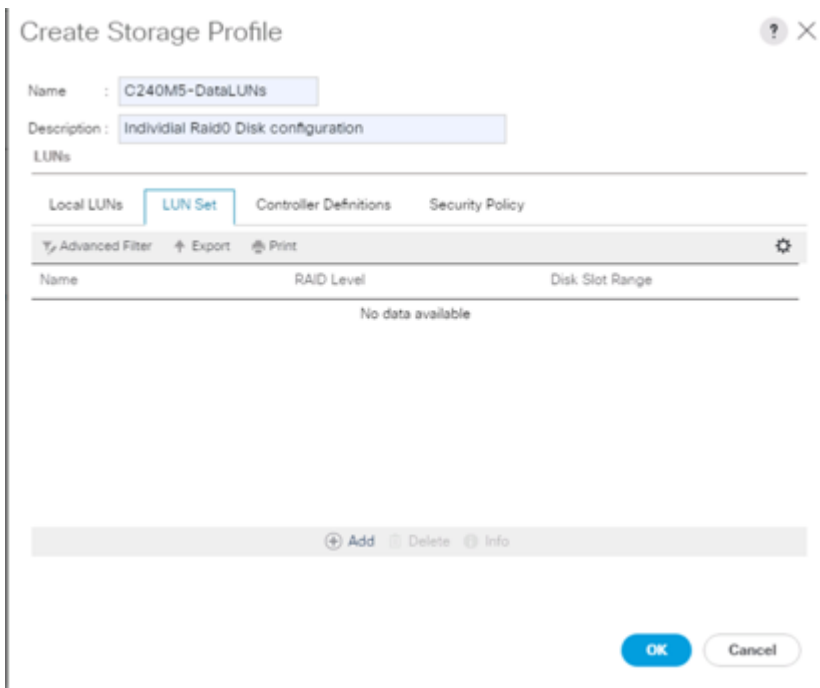
1. On the UCSM navigation page on the left-hand side select Storage tab.

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

Figure 33 Create Storage Profile



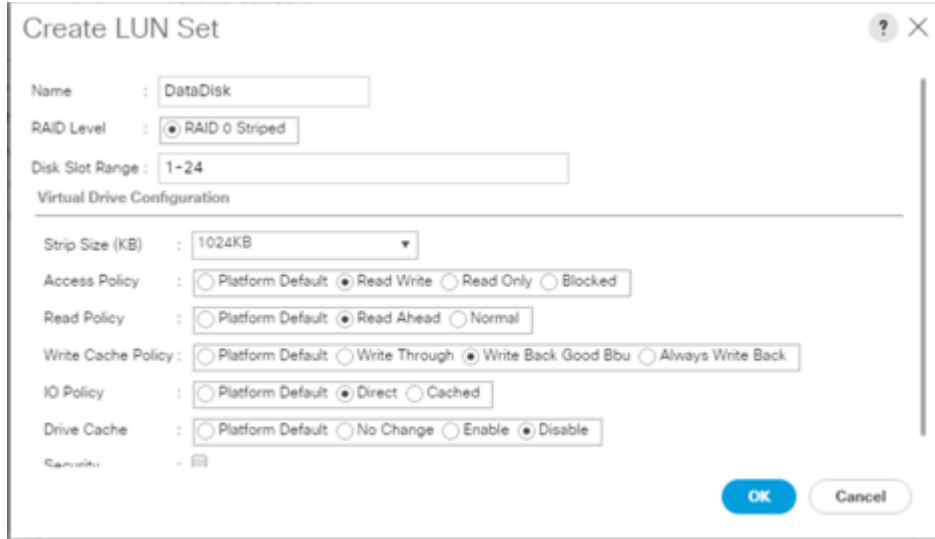
3. Enter a name for the Storage Profile and click the LUN Set tab.
4. Click Add.



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

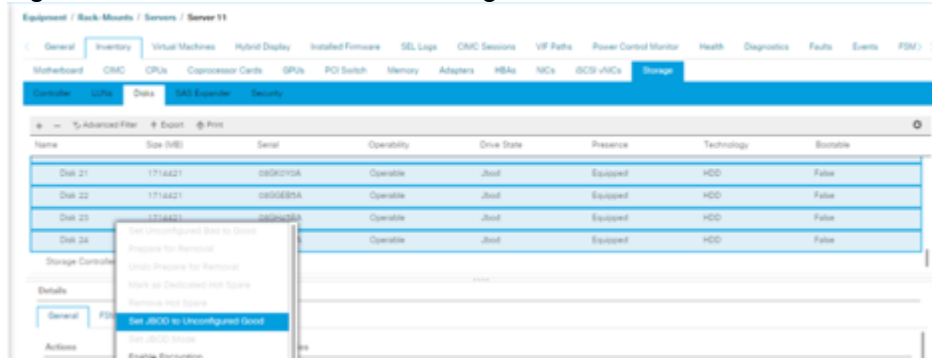
5. Select the properties for the LUN set:
 - a. Enter a name for LUN set.
 - b. Disk Slot Range – 1 – 24/26/56 (Depends on number of drives installed in a server).
 - c. Enter Virtual Drive configuration:

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



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

Figure 34 Set JBOD Disks to Unconfigured Good



Create Storage Policy and Storage Profile

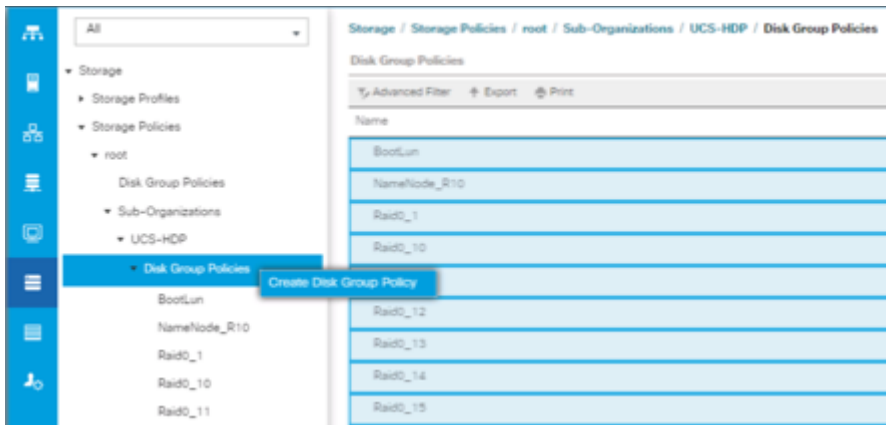
To create a Storage Profile with multiple RAID LUNs, create Storage Policies and attach them to a Storage Profile.

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

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



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



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

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

5. Disk Group Configuration.



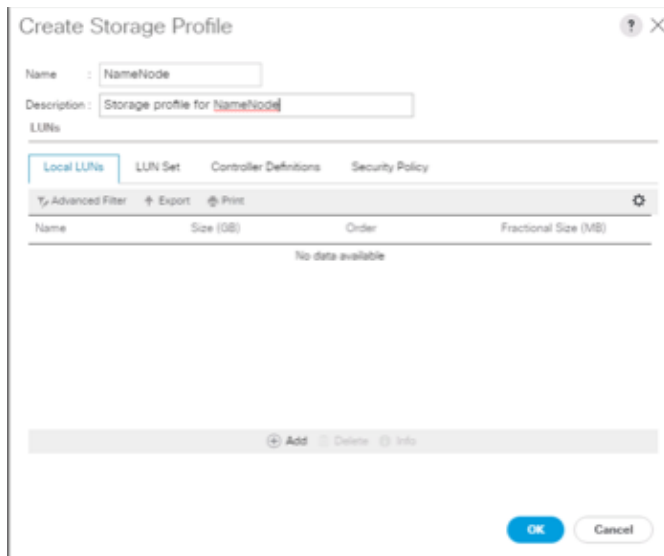
6. Virtual Drive Configuration.



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

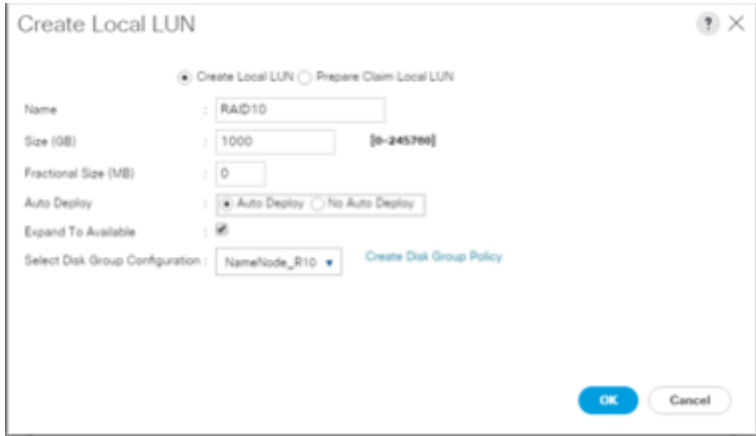


8. Enter a name for the Storage profile and click Add.



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

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



For Cisco UCS S3260, we created a Storage Profile with a Storage Policy to create a Boot LUN and attached it to a Storage Profile as shown above. The LUN set policy for an individual server node (server node 1 and server node 2) to create an individual RAID0 is shown in Figure 35.

Figure 35 Storage Policy to Configure Boot LUN for S3260 Server Node(s)

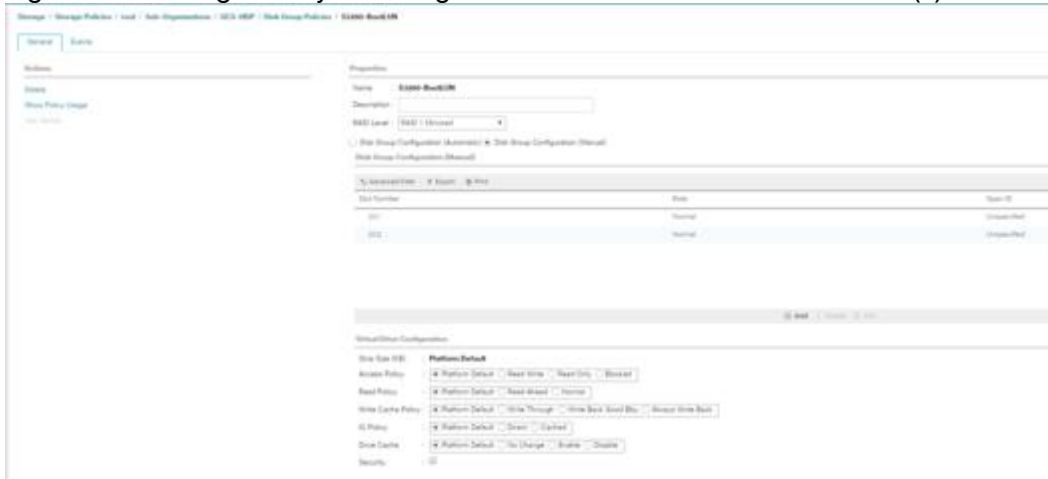


Figure 36 Storage Profile to Configure Individual RAID 0 on Server Node 1: Disk Slot 1-28

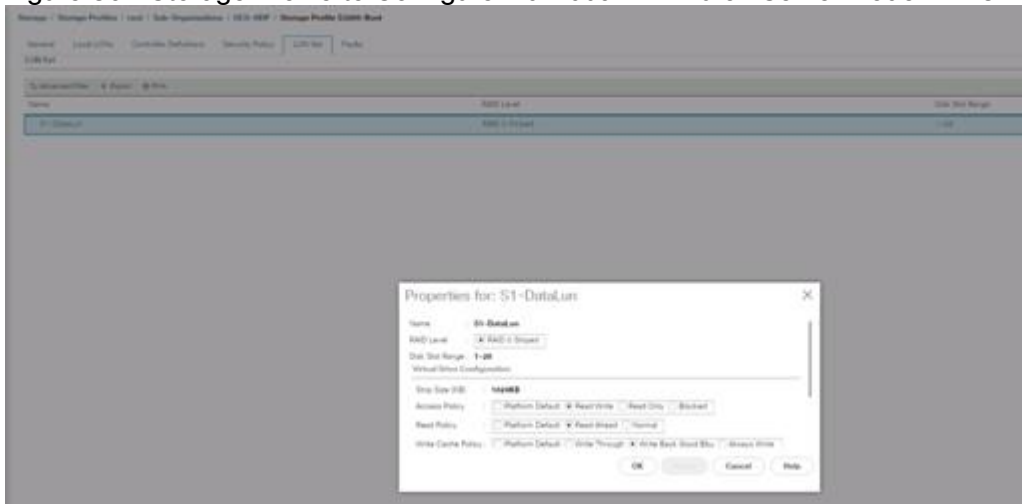
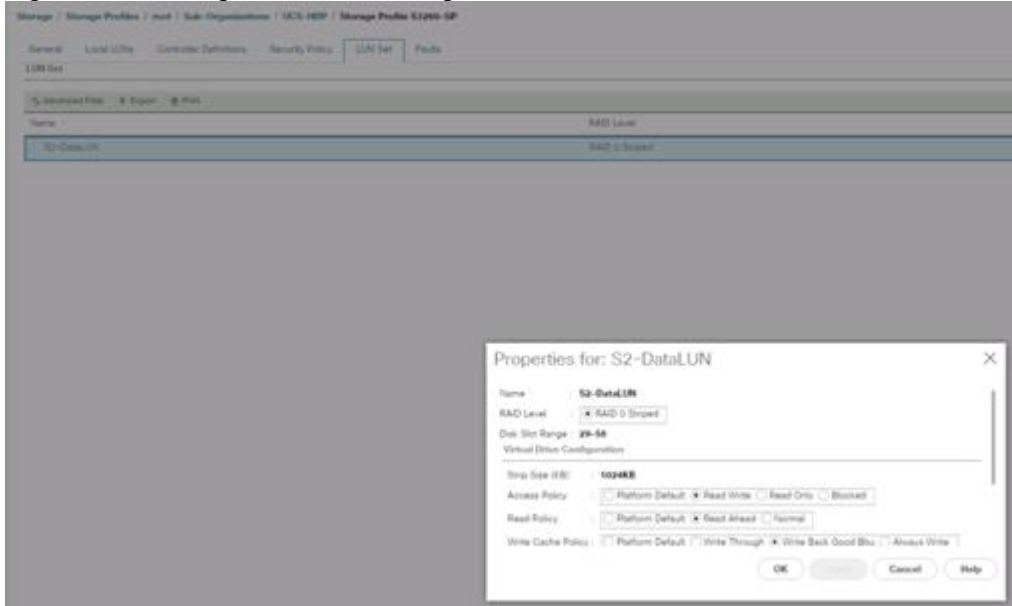


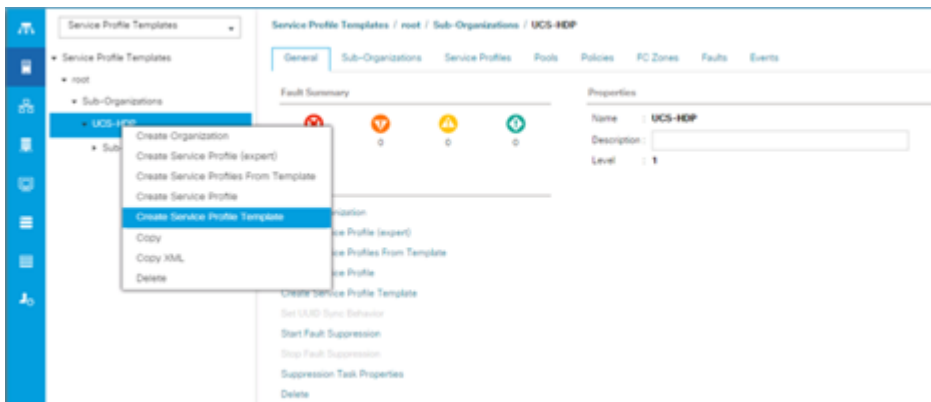
Figure 37 Storage Profile to Configure Individual RAID 0 on Server Node 1: Disk Slot 29-56



Create Service Profile Template

To create a service profile template, follow these steps:

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



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

Create Service Profile Template

You must enter a name for the service profile template and specify the template type. You can also specify how a UUID will be assigned to the template and enter a description.

Name:

The template will be created in the following organization. Its name must be unique within this organization.
Where: **org-root/org-UCS-HDP**

The template will be created in the following organization. Its name must be unique within this organization.
Type: Initial Template Updating Template

Specify how the UUID will be assigned to the server associated with the service generated by the template.
UUID

UID Assignment:

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

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

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

Create Service Profile Template

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

Specific Storage Profile:

Local Storage:

Create Local Disk Configuration Policy

Mode: **Any Configuration**

Protect Configuration: **Yes**

If **Protect Configuration** is set, the local disk configuration is preserved if the service profile is disassociated with the server. In that case, a configuration error will be raised when a new service profile is associated with that server if the local disk configuration in that profile is different.

FlexFlash

FlexFlash State: **Disable**

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

FlexFlash RAID Reporting State: **Disable**

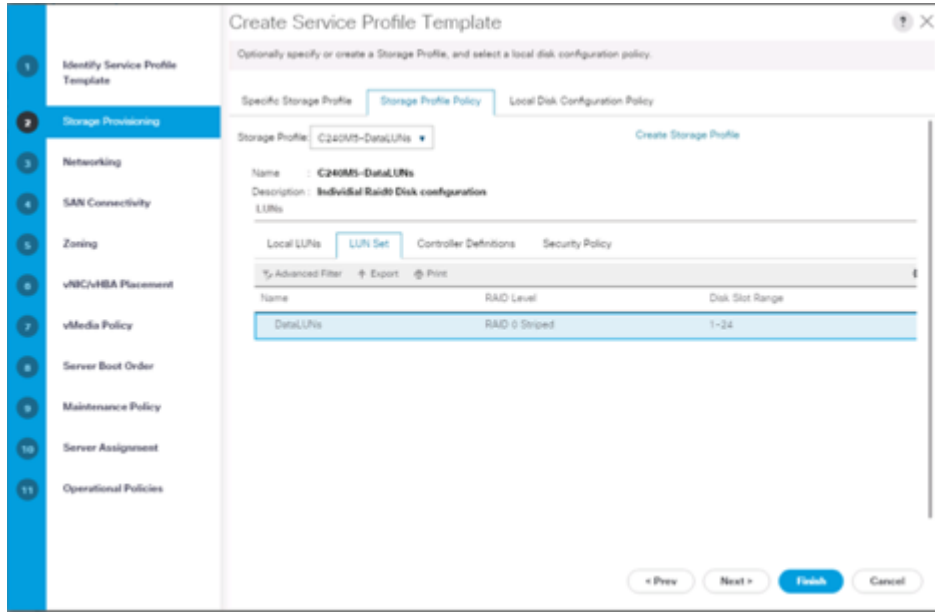
FlexFlash Removable State: **No Change**

If **FlexFlash Removable State** is changed, SD cards will become unavailable temporarily. Please ensure SD cards are not in use before changing the FlexFlash Removable State.

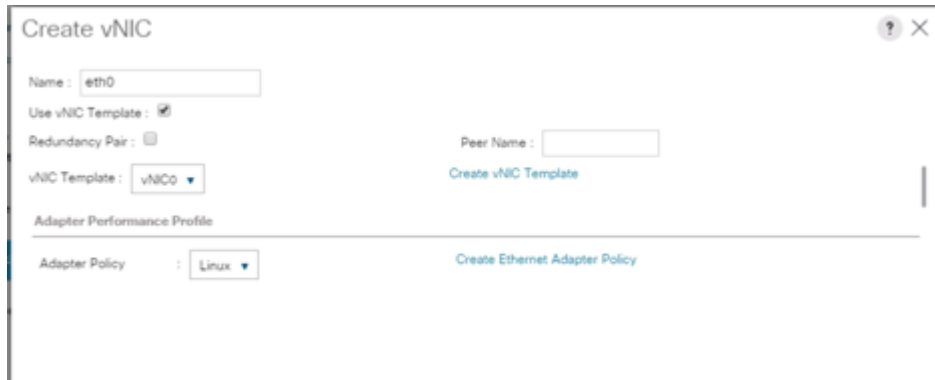
4. On Storage Profile Policy; select Storage Profile to attach with the server.




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

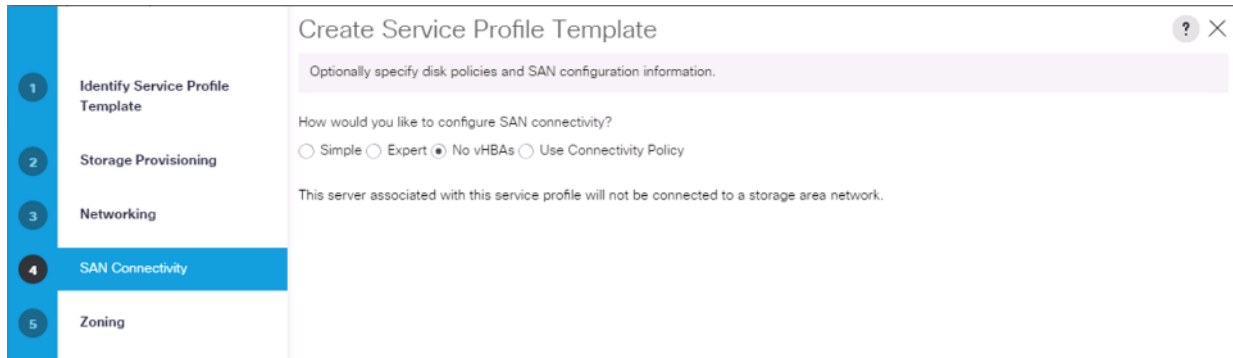


5. In the networking window, select Expert and click Add to create vNICs. Add one or more vNICs that the server should use to connect to the LAN.
6. In the create vNIC menu as vNIC name.
7. Select vNIC Template as vNIC0 and Adapter Policy as Linux.

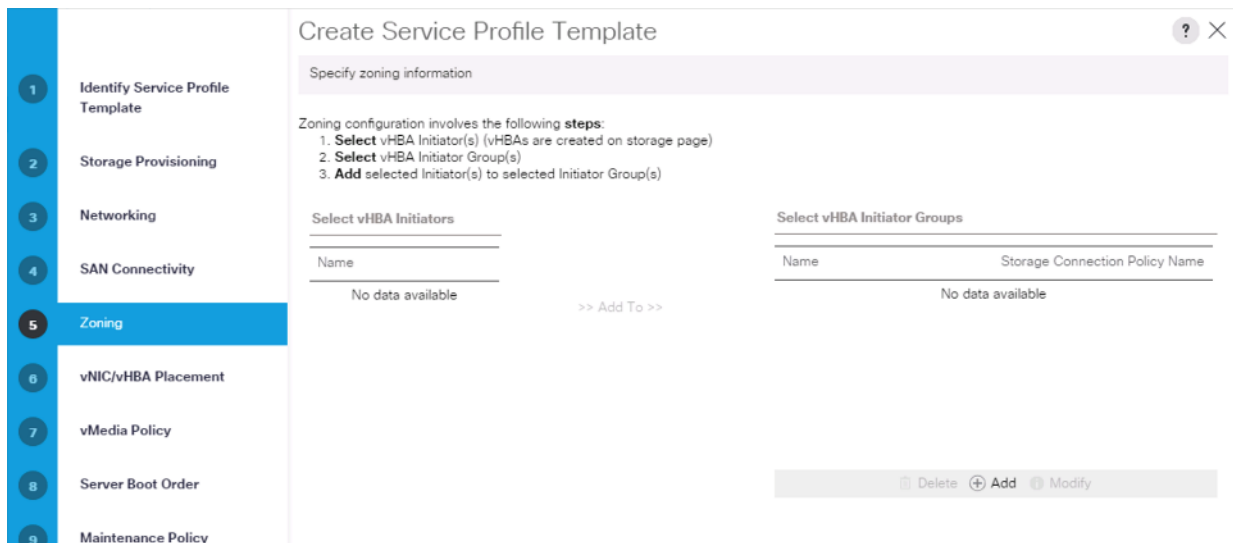


 Optionally, Network Bonding can be setup on the vNICs for each host for redundancy as well as for increased throughput.

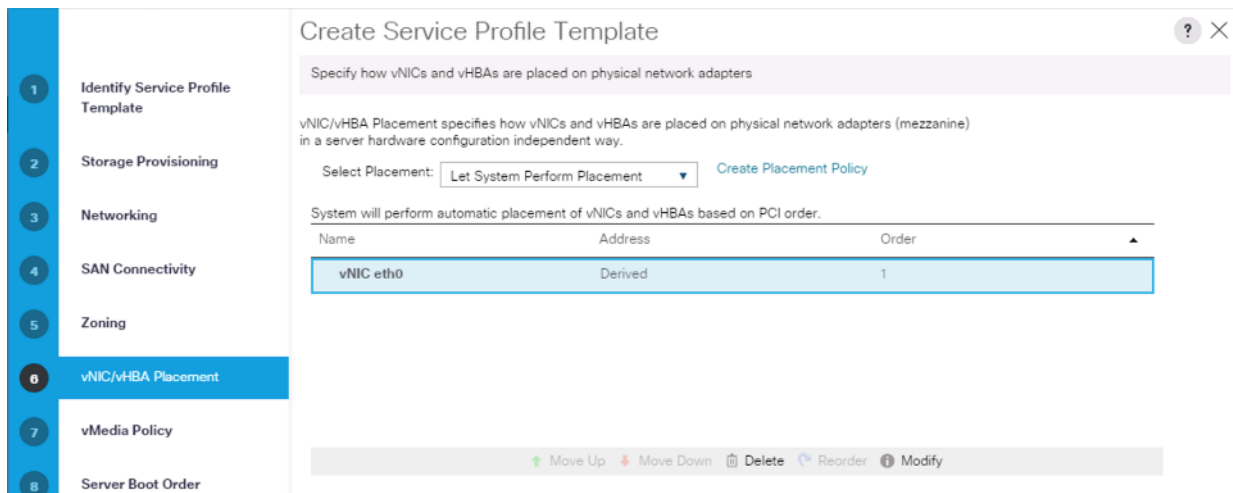
8. In the SAN Connectivity menu, select no vHBAs.



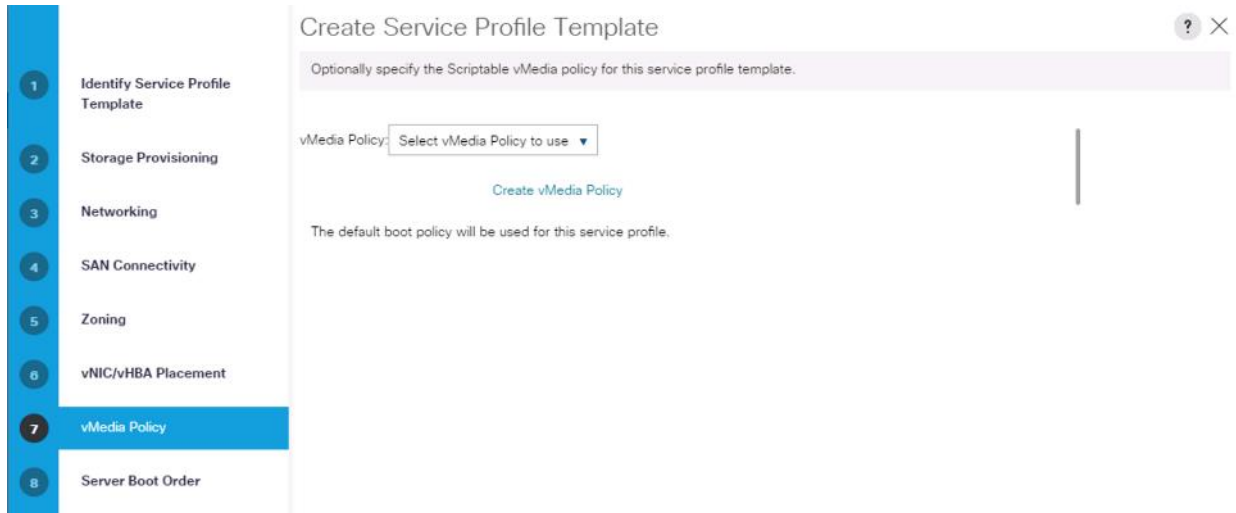
9. Click Next on the Zoning tab.



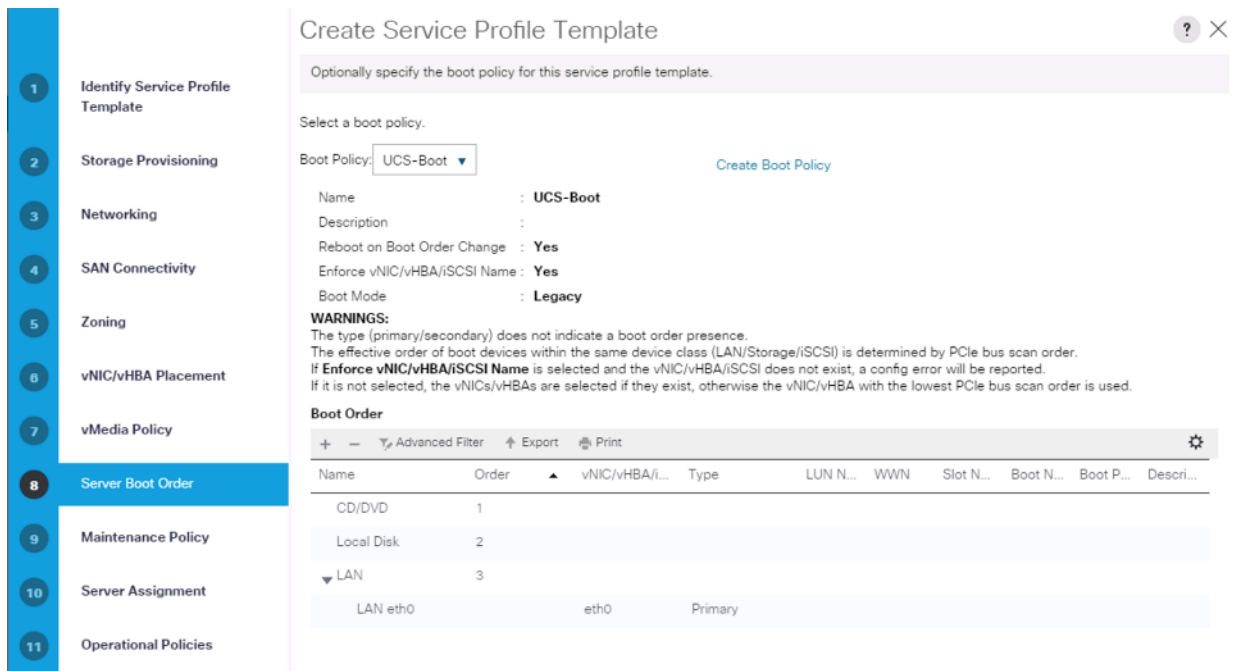
10. Select Let System Perform Placement for vNIC/vHBA Placement. Click Next.



11. Click Next on the vMedia Policy tab.



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




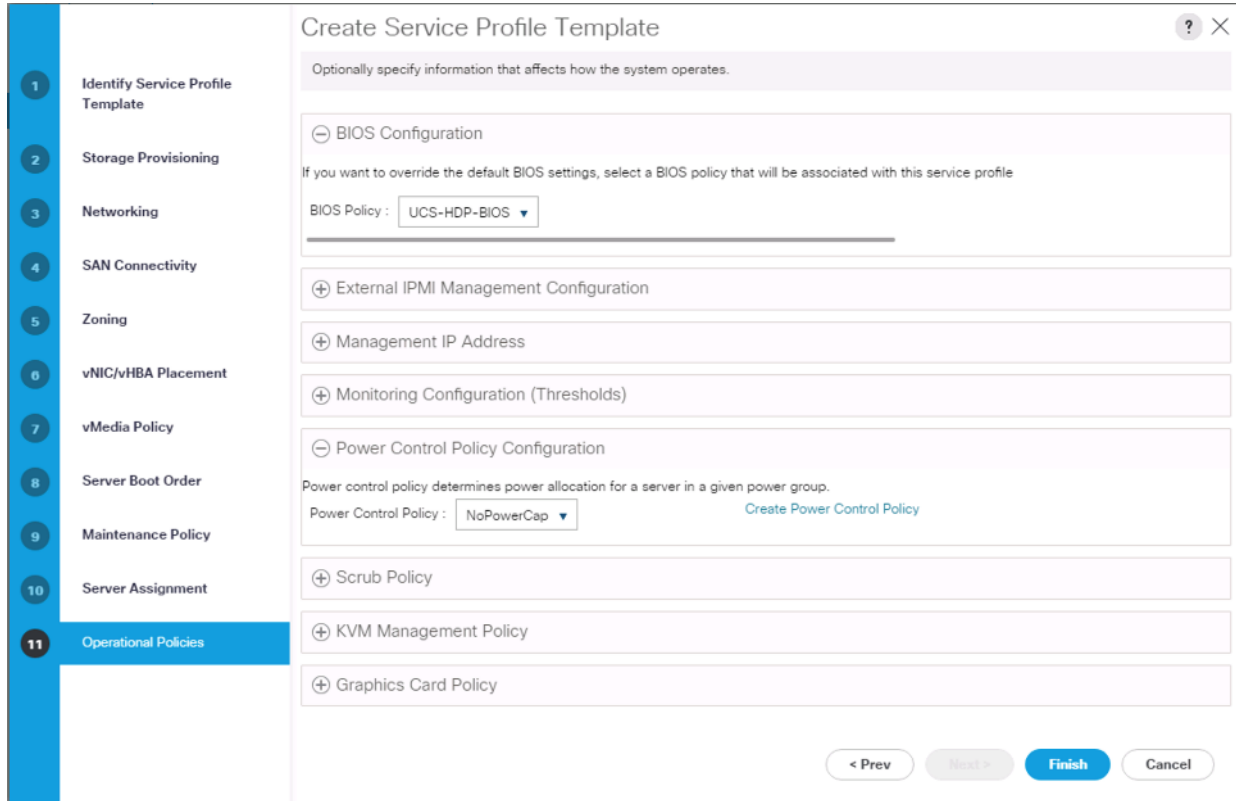
13. Select UserAck maintenance policy, which requires user acknowledgement prior rebooting server when making changes to policy or pool configuration tied to a service profile.

The screenshot shows the 'Create Service Profile Template' interface. On the left is a navigation sidebar with steps 1 through 7. Step 1, 'Identify Service Profile Template', is selected. The main content area is titled 'Maintenance Policy' and contains the following text: 'Specify how disruptive changes such as reboots, network interruptions, and firmware upgrades should be applied to the server associated with this service profile.' Below this, it says 'Select a maintenance policy to include with this service profile or create a new maintenance policy that will be accessible to all service profiles.' The 'Maintenance Policy' dropdown is set to 'UserAck'. A 'Create Maintenance Policy' link is visible. A summary table shows: Name: UserAck, Description: (empty), Soft Shutdown Timer: 150 Secs, Storage Config. Deployment Policy: User Ack, and Reboot Policy: User Ack.

14. Select the Server Pool policy to automatically assign a service profile to a server that meets the requirements for server qualification based on the pool configuration. Select Power state when the Service Profile is associated to server.
15. On the same page you can configure “Host firmware Package Policy” which helps to keep the firmware in sync when associated to server.

The screenshot shows the 'Create Service Profile Template' interface with steps 1 through 11 in the sidebar. Step 10, 'Server Assignment', is selected. The main content area includes: 'Optionally specify a server pool for this service profile template.' 'You can select a server pool you want to associate with this service profile template.' 'Pool Assignment' dropdown is set to 'UCS-HDP-ServerPool'. 'Select the power state to be applied when this profile is associated with the server.' Radio buttons for 'Up' and 'Down' are shown. 'The service profile template will be associated with one of the servers in the selected pool. If desired, you can specify an additional server pool policy qualification that the selected server must meet. To do so, select the qualification from the list.' 'Server Pool Qualification' dropdown is set to '<not set>'. 'Restrict Migration' checkbox is checked. The 'Firmware Management (BIOS, Disk Controller, Adapter)' section contains: 'If you select a host firmware policy for this service profile, the profile will update the firmware on the server that it is associated with. Otherwise the system uses the firmware already installed on the associated server.' 'Host Firmware Package' dropdown is set to 'UCS-MFP'. A 'Create Host Firmware Package' link is visible.

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



16. Click Finish to create the Service Profile template.

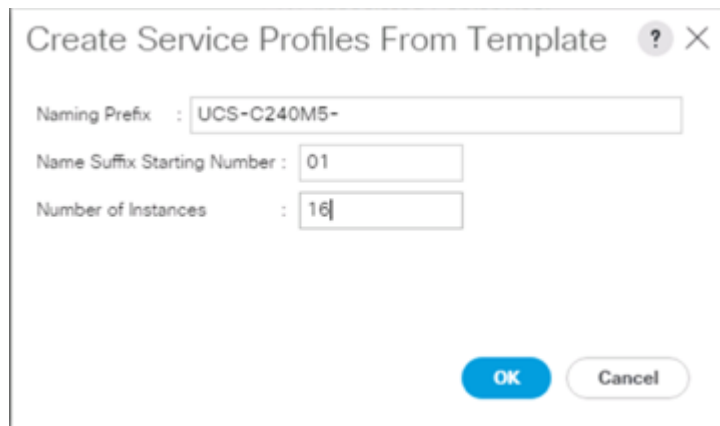
Create Service Profiles from Template

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

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

Figure 38 Create Service Profile from Template






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

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

Install Red Hat Enterprise Linux 7.6

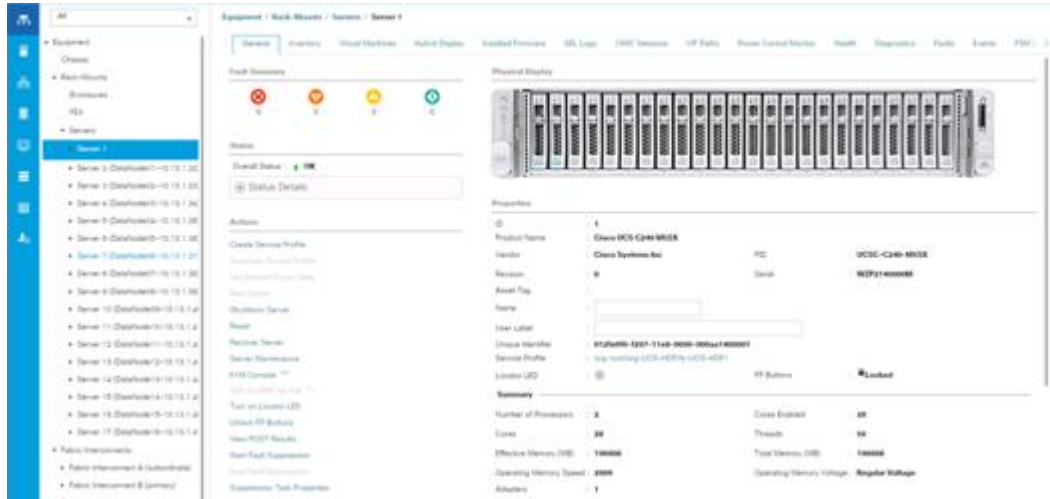
This section provides detailed procedures for installing Red Hat Enterprise Linux Server using Software RAID (OS based Mirroring) on Cisco UCS C240 M5 servers. There are multiple ways to install the RHEL operating system. The installation procedure described in this deployment guide uses KVM console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



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

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

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

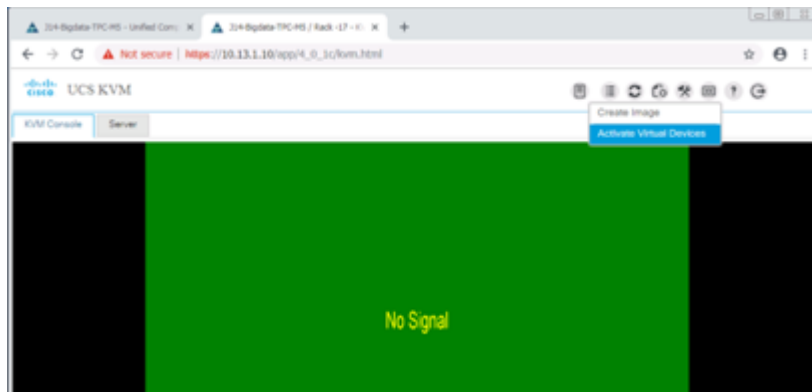


6. Click the link to launch the KVM console.

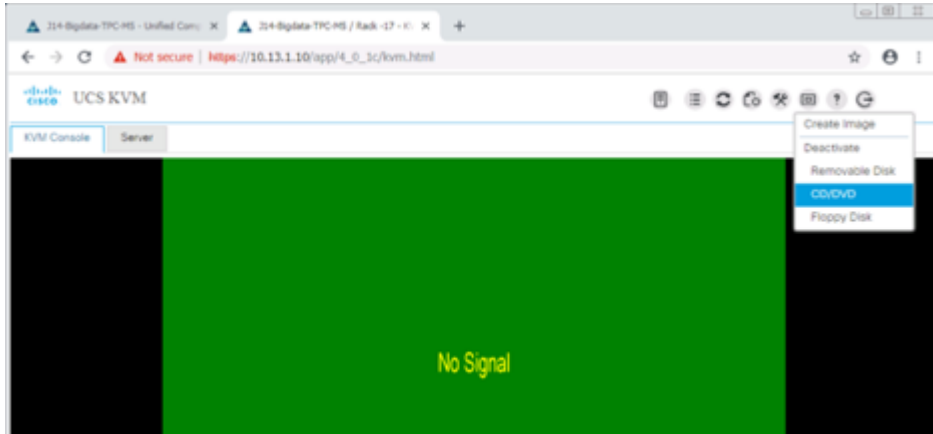
KVM server certificate has been accepted. Click this link to continue loading the KVM client application:
https://10.13.1.10/app/4_0_1c/kvm.html?&kvmIpAddr=10.13.1.166

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

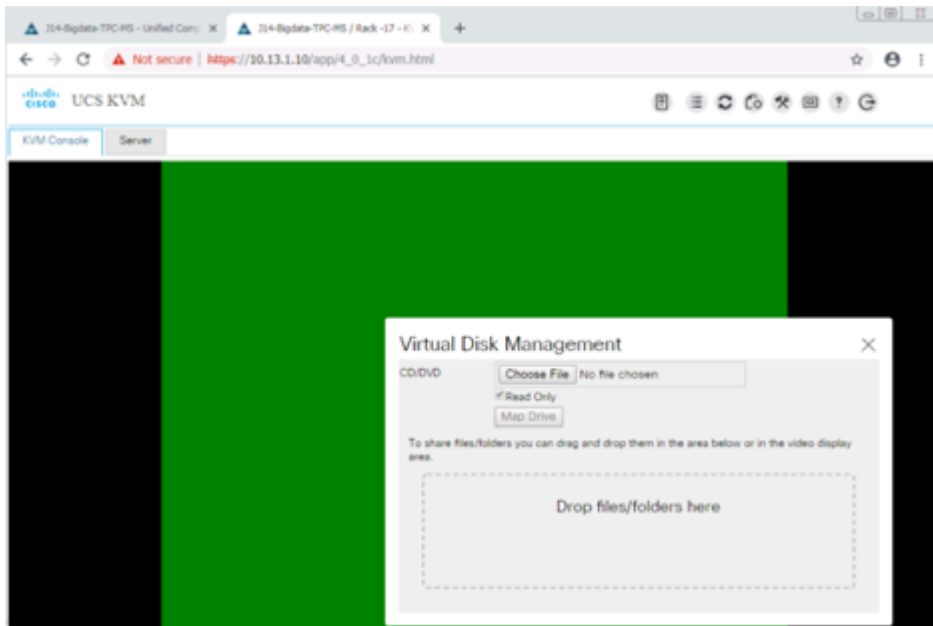
8. Click the Activate Virtual Devices found in Virtual Media tab.



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



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



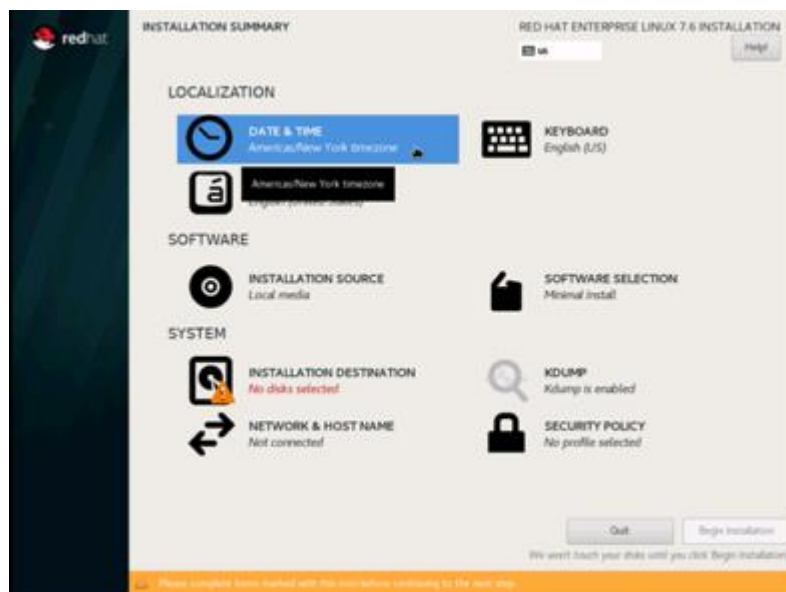
11. Browse to the Red Hat Enterprise Linux 7.6 installer ISO image file.



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



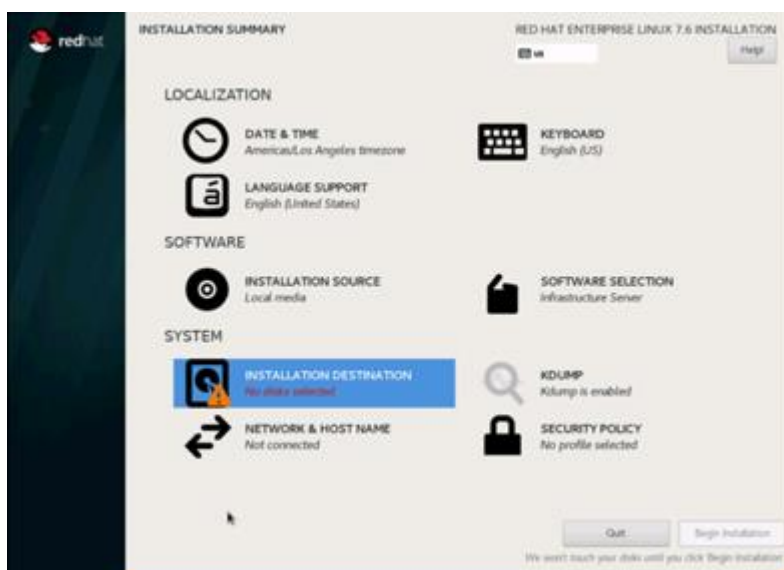
12. Click Open to add the image to the list of virtual media.
13. Select the Installation option from Red Hat Enterprise Linux 7.6.
14. Select the language for the installation and click Continue.
15. Select date and time, which pops up another window as shown below.



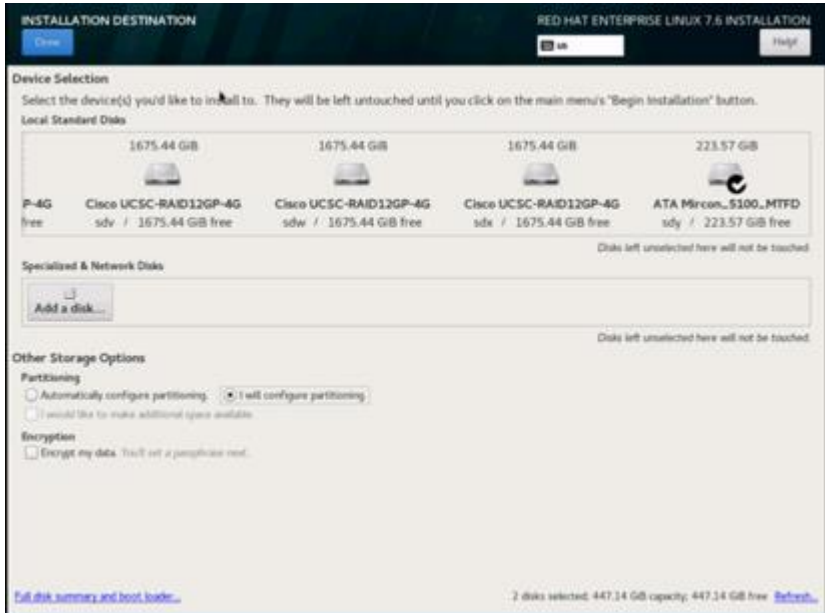
16. Select the location on the map, set the time, and click Done.



17. Click Installation Destination.

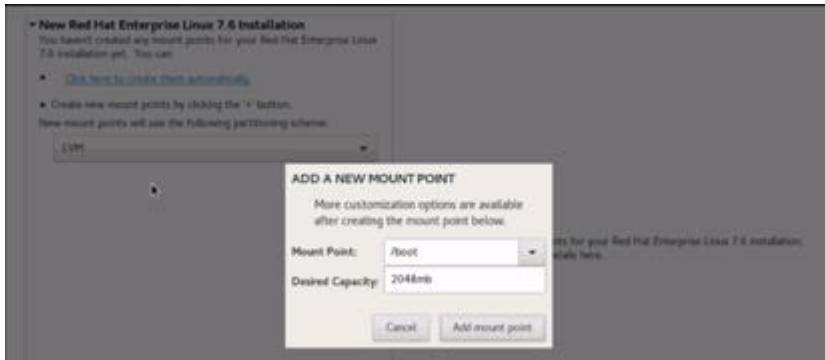


18. This opens a new window with the boot disks. Make the selection and choose "I will configure partitioning". Click Done. We selected two M.2 SATA SSDs.

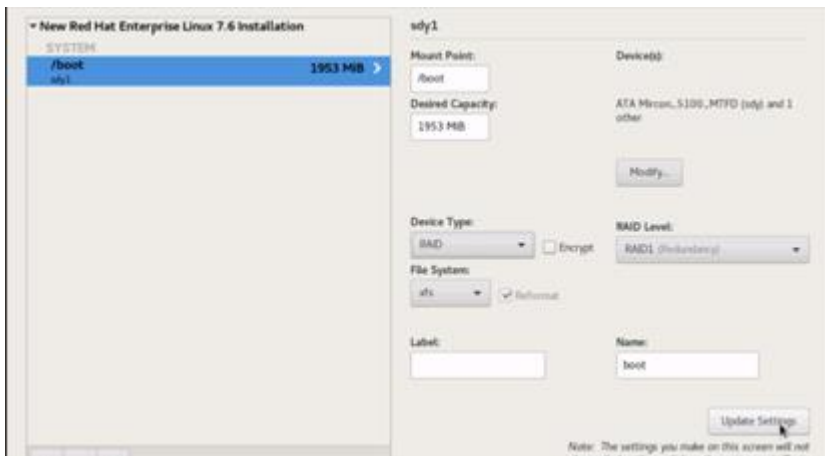


19. This opens a window to create the partitions. Click the + sign to add a new partition as shown below with a boot partition size 2048 MB.

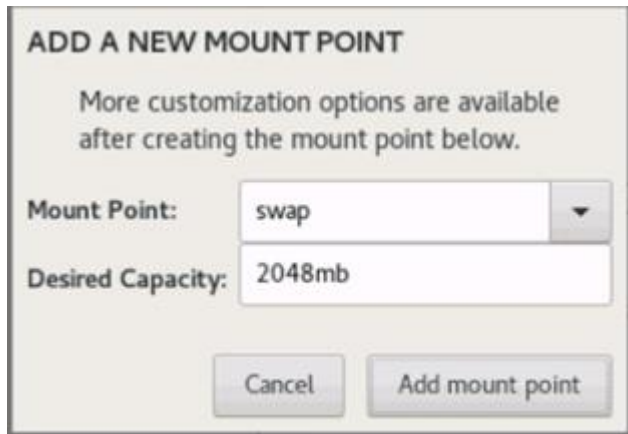
20. Click Add Mount Point to add the partition.



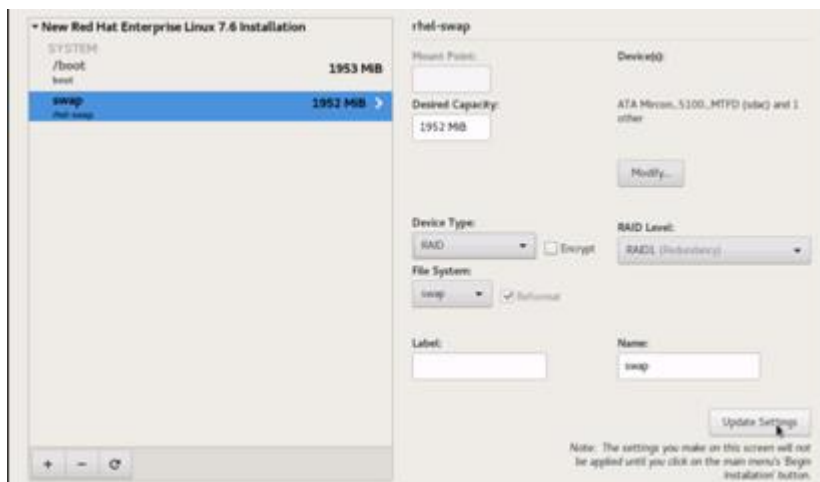
21. Change the device type to RAID and make sure the RAID level is RAID1 (redundancy) and click Update Settings to save the changes.



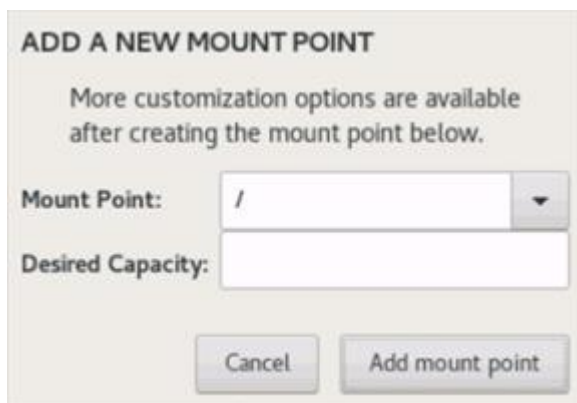
22. Click the + sign to create the swap partition of size 2048 MB. Click Add Mount Point.



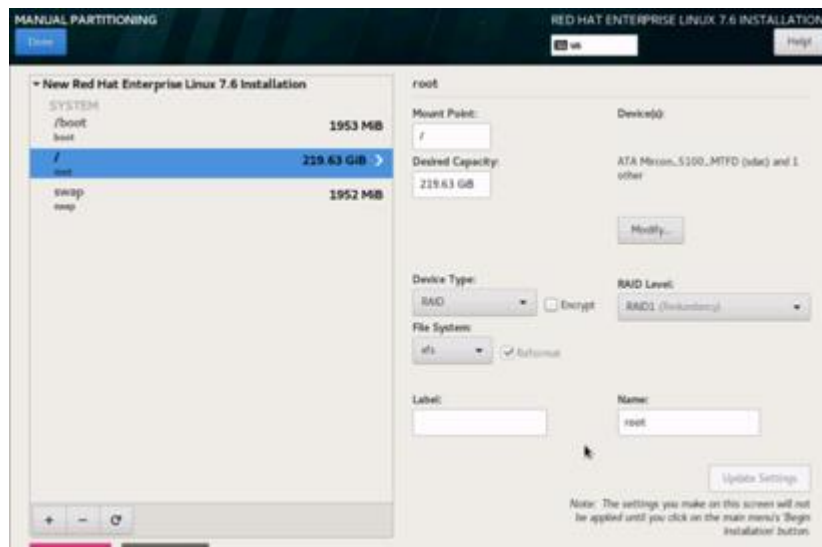
23. Change the Device type to RAID and RAID level to RAID1 (Redundancy) and click Update Settings.



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

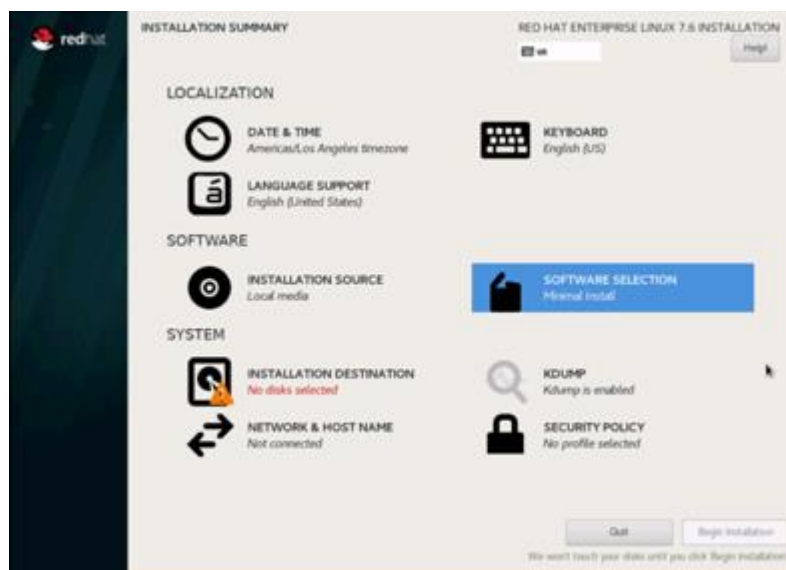


25. Change the Device type to RAID and RAID level to RAID1 (Redundancy). Click Update Settings.



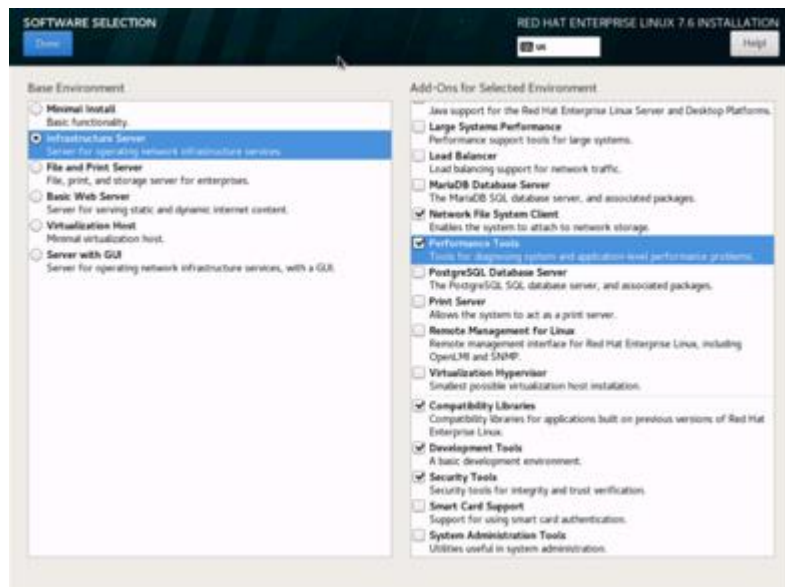
26. Click Done to go back to the main screen and continue the Installation.

27. Click Software Selection.

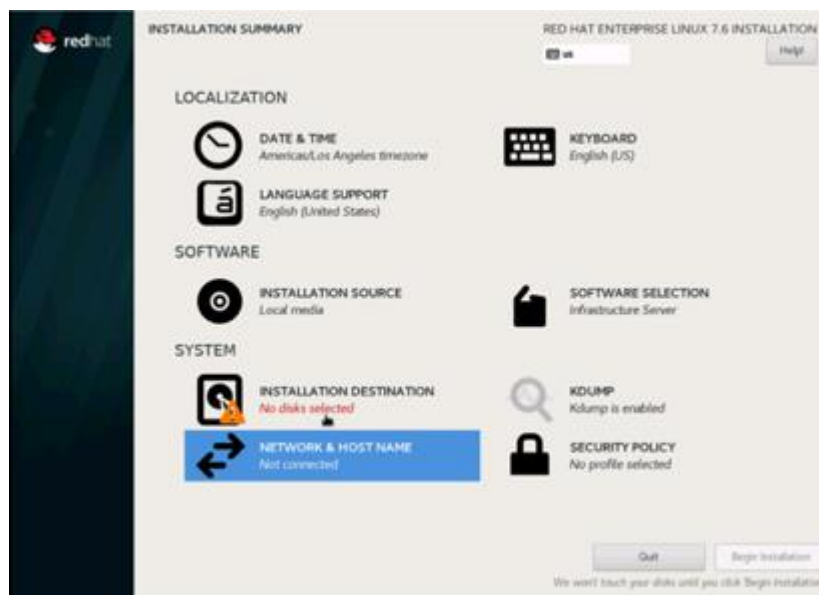


28. Select Infrastructure Server and select the Add-Ons as noted below then click Done:

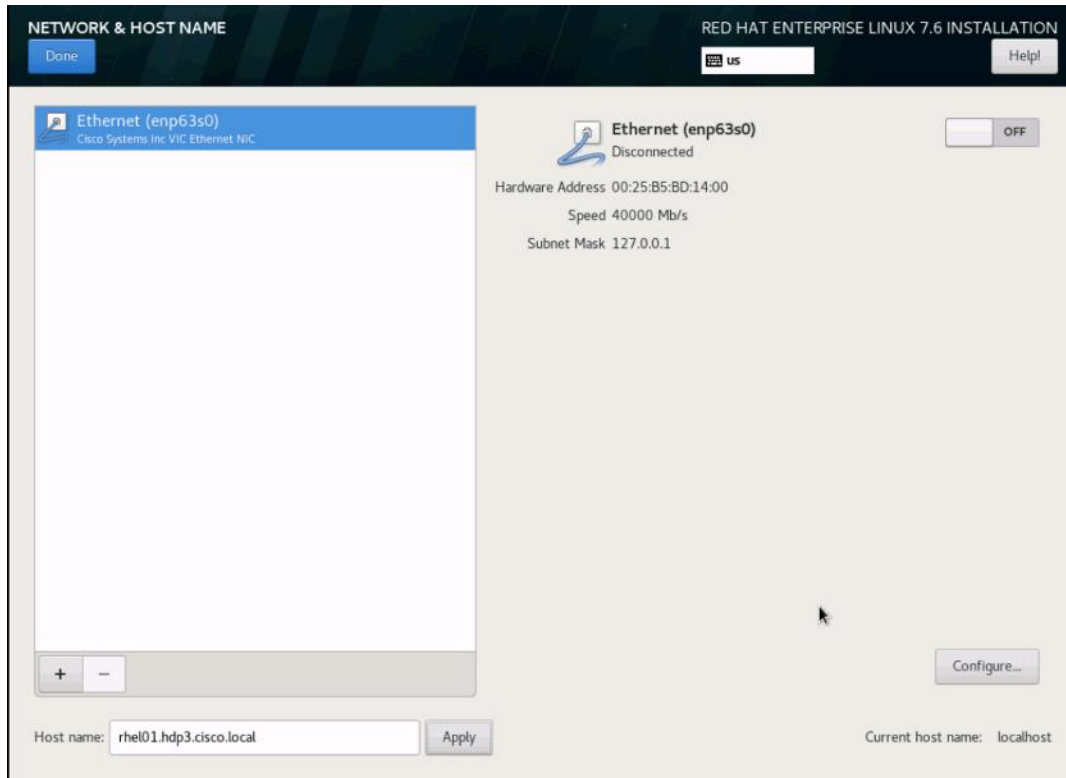
- a. Network File System Client
- b. Performance Tools
- c. Compatibility Libraries
- d. Development Tools
- e. Security Tools



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

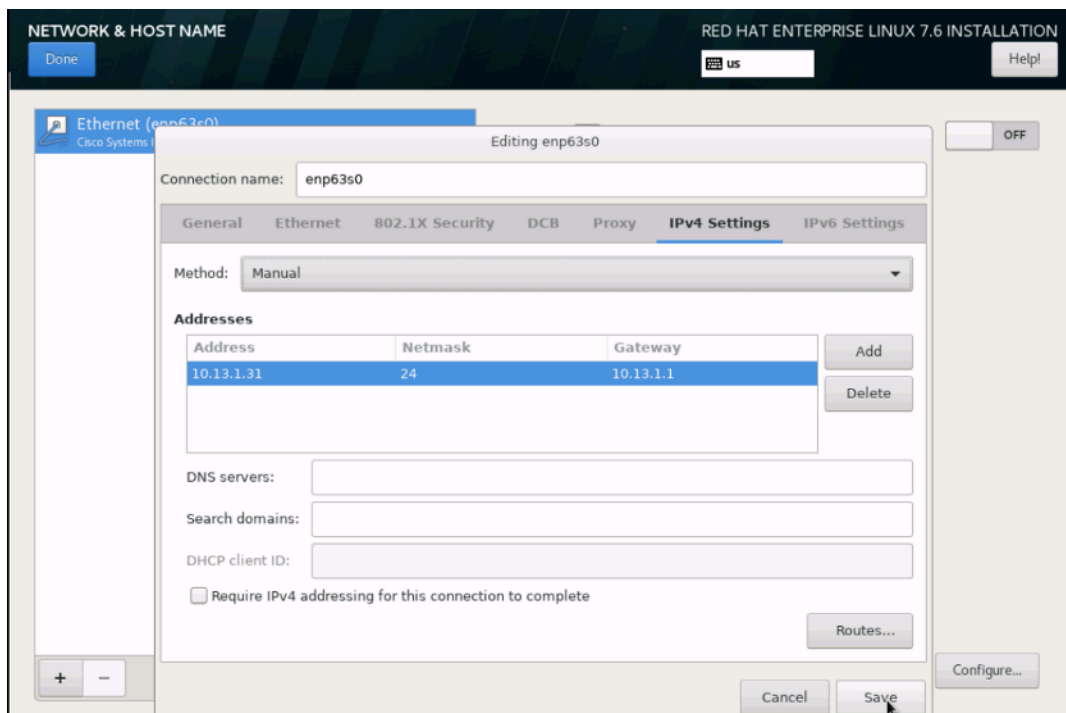


30. Type in the hostname as shown below.



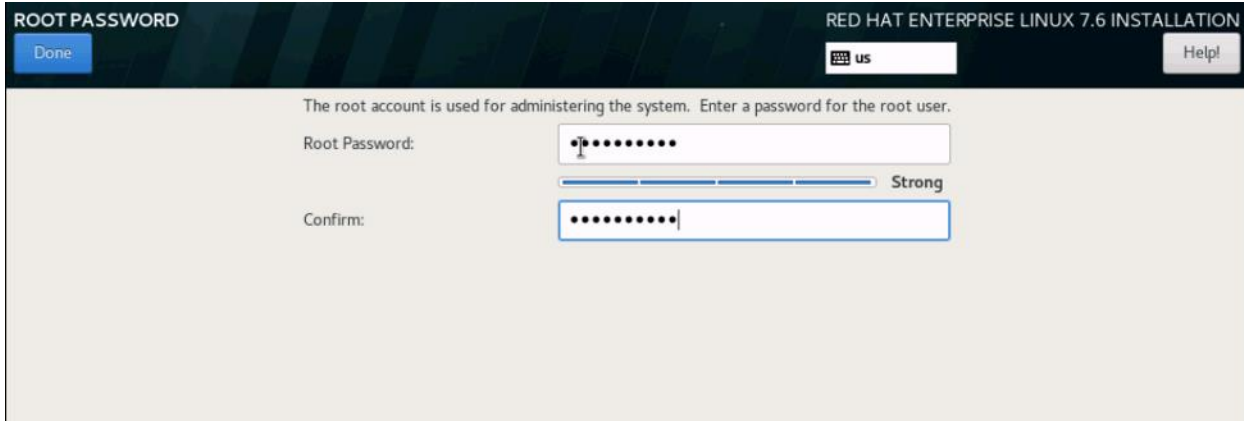
31. Click Configure to open the Network Connectivity window. Click IPv4 Settings.

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



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

34. Click Begin Installation in the main menu.
35. Select Root Password in the User Settings.
36. Enter the Root Password and click Done.



37. Once the installation is complete reboot the system.
38. Repeat steps 1 to 37 to install Red Hat Enterprise Linux 7.6 on Servers 2 through 30.



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

The hostnames and their corresponding IP addresses are shown in Table 5 .

Table 5 Hostname and IP address

Hostname	Eth0
rhel01	10.15.1.31
rhel02	10.15.1.32
rhel03	10.15.1.33
rhel04	10.15.1.34
rhel05	10.15.1.35
.....
Rhel29	10.15.1.59
Rhel30	10.15.1.60



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



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

Post OS Install Configuration

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

Configure /etc/hosts

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



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

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

```
ORIGIN hdp3.cisco.local
rhel01  A 10.15.1.31
rhel02  A 10.15.1.32
rhel03  A 10.15.1.33
...
...
rhel29  A 10.15.1.59
rhel30  A 10.15.1.60
```

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

1. Log into the Admin Node (rhel01).

```
#ssh 10.15.1.31
```

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

3. On Admin Node (rhel01):

```
[root@rhel01 ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.15.1.31  rhel01  rhel01.hdp3.cisco.local
10.15.1.32  rhel02  rhel02.hdp3.cisco.local
10.15.1.33  rhel03  rhel03.hdp3.cisco.local
10.15.1.34  rhel04  rhel04.hdp3.cisco.local
10.15.1.35  rhel05  rhel05.hdp3.cisco.local
10.15.1.36  rhel06  rhel06.hdp3.cisco.local
10.15.1.37  rhel07  rhel07.hdp3.cisco.local
10.15.1.38  rhel08  rhel08.hdp3.cisco.local
10.15.1.39  rhel09  rhel09.hdp3.cisco.local
10.15.1.40  rhel10  rhel10.hdp3.cisco.local
10.15.1.41  rhel11  rhel11.hdp3.cisco.local
```

```

10.15.1.42 rhel12 rhel12.hdp3.cisco.local
10.15.1.43 rhel13 rhel13.hdp3.cisco.local
10.15.1.44 rhel14 rhel14.hdp3.cisco.local
10.15.1.45 rhel15 rhel15.hdp3.cisco.local
10.15.1.46 rhel16 rhel16.hdp3.cisco.local
10.15.1.47 rhel17 rhel17.hdp3.cisco.local
10.15.1.48 rhel18 rhel18.hdp3.cisco.local
10.15.1.49 rhel19 rhel19.hdp3.cisco.local
10.15.1.50 rhel20 rhel20.hdp3.cisco.local
10.15.1.51 rhel21 rhel21.hdp3.cisco.local
10.15.1.52 rhel22 rhel22.hdp3.cisco.local
10.15.1.53 rhel23 rhel23.hdp3.cisco.local
10.15.1.54 rhel24 rhel24.hdp3.cisco.local
10.15.1.55 rhel25 rhel25.hdp3.cisco.local
10.15.1.56 rhel26 rhel26.hdp3.cisco.local
10.15.1.57 rhel27 rhel27.hdp3.cisco.local
10.15.1.58 rhel28 rhel28.hdp3.cisco.local

```

Set Up Passwordless Login

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

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

1. Log into the Admin Node (rhel01).

```
#ssh 10.15.1.31
```

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

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

Figure 39 ssh-keygen

```

[root@rhel01 ~]# ssh-keygen -N '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
Created directory '/root/.ssh'.
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:YAnkweN48qxdy6BxLIpH/H5D2UYOU/Dd7+M0k4Dm6aw root@rhel01.hdp3.cisco.local
The key's randomart image is:
+---[RSA 2048]-----+
|
|  o+ ..          |
| ..+o o.. .     |
| o B..+... .    |
|+ * =.o... .    |
|+O +  BS . .    |
|+ * . oo+ . . o |
| . o . .o + o   |
| . oo + .       |
| .Eoo .o       |
+---[SHA256]-----+

```

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

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


4. Enter yes for Are you sure you want to continue connecting (yes/no)?
5. Enter the password of the remote host.

```

root@rhel01:~# for i in {1..30};do sudo "copying rhel01.hq3.cisco.local" ssh-copy-id -i ~/.ssh/id_rsa.pub root@rhel01.hq3.cisco.local;done;
copying rhel01.hq3.cisco.local
/ssh/rsa/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'rhel01.hq3.cisco.local [10.15.1.31]' can't be established.
RSA key fingerprint is 88A354:06d952d109f27f9a9e570c18c6e846d010109e.
RSA key fingerprint is MD5:13:0e:a8:08:aa:70:f4:d8:83:68:8d:3d:3e:88:bd:53.
Are you sure you want to continue connecting (yes/no)? yes
/ssh/rsa/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/ssh/rsa/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@rhel01.hq3.cisco.local's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'root@rhel01.hq3.cisco.local'"
and check to make sure that only the key(s) you wanted were added.

copying rhel02.hq3.cisco.local
/ssh/rsa/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'rhel02.hq3.cisco.local [10.15.1.32]' can't be established.
RSA key fingerprint is 88A354:06d952d109f27f9a9e570c18c6e846d010109e.
RSA key fingerprint is MD5:aa:84:70:ae:5e:28:0f:f2:a4:0e:0e:6a:8d:77:d0:38.
Are you sure you want to continue connecting (yes/no)? yes
/ssh/rsa/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/ssh/rsa/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@rhel02.hq3.cisco.local's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'root@rhel02.hq3.cisco.local'"
and check to make sure that only the key(s) you wanted were added.

```

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

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

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

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

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

2. Copy the contents of the Red Hat DVD to /var/www/html/rhelrepo
3. Alternatively, if you have access to a Red Hat ISO Image, Copy the ISO file to rhel01.
4. Log back into rhel01 and create the mount directory.

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

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

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

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

```
# vi /var/www/html/rhelrepo/rheliso.repo
[rhel7.6]
name=Red Hat Enterprise Linux 7.6
baseurl=http://10.15.1.31/rhelrepo
gpgcheck=0
enabled=1
```

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

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



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

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



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

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

Create the Red Hat Repository Database

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

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

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

- Run “createrepo” on the RHEL repository to create the repo database on admin node.

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

Figure 40 createrepo

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

Set Up Ansible

To set up Ansible, follow these steps:

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

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



For more information about downloading and installing the Ansible engine, go to: <https://access.redhat.com/articles/3174981>

- Run the following command to install ansible:

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

3. Verify Ansible installation by running the following commands:

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

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

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

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

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

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

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

```
rhel22.hdp3.cisco.local  
rhel23.hdp3.cisco.local  
rhel24.hdp3.cisco.local  
rhel25.hdp3.cisco.local  
rhel26.hdp3.cisco.local  
rhel27.hdp3.cisco.local  
rhel28.hdp3.cisco.local
```

```
[nodes]
```

```
rhel101.hdp3.cisco.local  
rhel102.hdp3.cisco.local  
rhel103.hdp3.cisco.local  
rhel104.hdp3.cisco.local  
rhel105.hdp3.cisco.local  
rhel106.hdp3.cisco.local  
rhel107.hdp3.cisco.local  
rhel108.hdp3.cisco.local  
rhel109.hdp3.cisco.local  
rhel110.hdp3.cisco.local  
rhel111.hdp3.cisco.local  
rhel112.hdp3.cisco.local  
rhel113.hdp3.cisco.local  
rhel114.hdp3.cisco.local  
rhel115.hdp3.cisco.local  
rhel116.hdp3.cisco.local  
rhel117.hdp3.cisco.local  
rhel118.hdp3.cisco.local  
rhel119.hdp3.cisco.local  
rhel120.hdp3.cisco.local  
rhel121.hdp3.cisco.local  
rhel122.hdp3.cisco.local  
rhel123.hdp3.cisco.local  
rhel124.hdp3.cisco.local  
rhel125.hdp3.cisco.local  
rhel126.hdp3.cisco.local  
rhel127.hdp3.cisco.local  
rhel128.hdp3.cisco.local
```

5. Verify host group by running the following commands. Figure 41 shows the outcome of the ping command.

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

Figure 41 Ansible - Ping Hosts

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

Install httpd

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

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



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

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

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

```
# vi /etc/httpd/conf/httpd.conf
ServerName 10.15.1.31:80
```

3. Start httpd:

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

Set Up All Nodes to use the RHEL Repository

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



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

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

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

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

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

3. Purge the yum caches:

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



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

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

Upgrade the Cisco Network Driver for VIC1387

To upgrade the Cisco network driver for VIC1387, follow these steps:

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

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

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

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

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

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

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

- 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\RH7\RH7.6\kmod-megaraid_sas-07.708.03.00_el7.6-2.x86_64.rpm
- Copy the rpm on all nodes of the cluster using the following Ansible commands. For this example, the rpm is assumed to be in present working directory of rhel01:

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

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

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

Set Up JAVA

To setup JAVA, follow these steps:



CDH 6 requires JAVA 8.

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

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

- Extract and Install JDK on all nodes:

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

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

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

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

- Make the two java scripts created above executable:

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

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

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

7. Setup Java Alternatives

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

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

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

9. Setup JAVA_HOME on all nodes.

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

10. Display JAVA_HOME on all nodes.

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

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

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

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

11. Display current java -version.

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

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

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

Enable Syslog

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

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


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

Set the ulimit

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

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

To set `ulimit`, follow these steps:

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

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

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

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

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

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

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



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

Disable SELinux



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

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

To disable SELinux, follow these steps:

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

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



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

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

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

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

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

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

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

Set TCP Retries

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

To set TCP retries, follow these steps:



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

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

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

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

```
# ansible nodes -m command -a "sysctl -p"
```

Disable the Linux Firewall



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

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

Disable IPv6 Defaults

To disable IPv6 defaults, follow these steps:

1. Run the following command:

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

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

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

Disable Swapping

To disable swapping, follow these steps:

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

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

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

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

Disable Memory Overcommit

To disable Memory Overcommit, follow these steps:

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

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

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

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

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

Disable Transparent Huge Pages

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

To disable Transparent Huge Pages, follow these steps:

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

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

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

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

3. Copy file to each node:

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

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

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

NTP Configuration

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

To configure NTP, follow these steps:

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



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

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

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

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

```
# vi /root/ntp.conf
server 10.15.1.31
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
restrict -6 ::1
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```

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

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

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

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

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

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

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

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

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



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

7. Install the Chrony service:

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

8. Activate the Chrony service at boot:

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

9. Start the Chrony service:

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

Install Megaraid StorCLI

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

<https://docs.broadcom.com/docs/12352476>

To install StorCLI, follow these steps:

1. Download StorCLI: <https://www.broadcom.com/support/download-search/?pg=&pf=&pn=&po=&pa=&dk=storcli>.
2. Extract the .zip file and copy storcli-1.23.02-1.noarch.rpm from the linux directory.
3. Download StorCLI and its dependencies and transfer to Admin node:

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

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

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

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

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

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

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

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

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



The Cisco UCS Manager configuration explains the steps to deploy the required storage configuration via Storage Policy and Storage Profile attached to Service Profile Template for NameNode(s), Management Node(s), GPU Node(s) and DataNode(s). To configure Storage with StorCLI, go to section [Configure Data Drives on Name Node and Data Nodes](#), in the Appendix.

Configure the Filesystem for NameNodes and DataNodes

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

To configure the filesystem for NameNodes and DataNodes, follow these steps:

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

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

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



The following script creates partition and disk label for front facing 24 HDD disk.



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

```
#vi /root/driveconf.sh
#!/bin/bash
[[ "-x" == "${1}" ]] && set -x && set -v && shift 1
count=1
for X in /sys/class/scsi_host/host?/scan
do
echo '- - -' > ${X}
done
for X in /dev/sd?
do
list+=$(echo $X " ")
done
for X in /dev/sd??
do
list+=$(echo $X " ")
done
```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

6. On the admin create following scripts to configure NVMe disks partition and filesystem for DataNodes only.



The following script creates partition and disk label for rear facing 2 NVMe disk.

```

#vi nvme1.sh
echo "Formatting and Mounting Drive => /dev/nvme0n1"

```



```

/sbin/mkfs.xfs -f /dev/nvme0n1
(( $? )) && continue

#Identify UUID
UUID=`blkid /dev/nvme0n1 | cut -d " " -f2 | cut -d "=" -f2 | sed 's/"//g'`

echo "Make Directory /data/nvme01"
/bin/mkdir -p /data/nvme01
(( $? )) && continue

echo "UUID of /dev/nvme0n1 = ${UUID}, mounting nvme0n1 using UUID on /data/nvme01"
/bin/mount -t xfs -o inode64,noatime -U ${UUID} /data/nvme01
(( $? )) && continue

echo "Creating fstab entry ${UUID} /data/nvme01 xfs inode64,noatime 0 0"
echo "UUID=${UUID} /data/nvme01 xfs inode64,noatime 0 0" >> /etc/fstab

done

#vi nvme2.sh
echo "Formatting and Mounting Drive => /dev/nvme1n1"
/sbin/mkfs.xfs -f /dev/nvme1n1
(( $? )) && continue

#Identify UUID
UUID=`blkid /dev/nvme1n1 | cut -d " " -f2 | cut -d "=" -f2 | sed 's/"//g'`

echo "Make Directory /data/nvme02"
/bin/mkdir -p /data/nvme02
(( $? )) && continue

echo "UUID of /dev/nvme1n1 = ${UUID}, mounting nvme1n1 using UUID on /data/nvme02"
/bin/mount -t xfs -o inode64,noatime -U ${UUID} /data/nvme02
(( $? )) && continue

echo "Creating fstab entry ${UUID} /data/nvme02 xfs inode64,noatime 0 0"
echo "UUID=${UUID} /data/nvme02 xfs inode64,noatime 0 0" >> /etc/fstab

done

```

Delete Partitions

To delete a partition, follow these steps:

1. Run the mount command (`\mount`) to identify which drive is mounted to which device `/dev/sd<?>`
2. `umount` the drive for which partition is to be deleted and run `fdisk` to delete as shown below.



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

```

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

```

Cluster Verification

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

To verify a cluster, follow these steps:



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

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

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

```

echo ""
echo ""
# probe for nic info #
echo -e "${green}NIC ${NC}"
clush -a -B "`which ifconfig` | egrep ' (^e|^p)' | awk '{print \$1}' | \ xargs -l
`which ethtool` | grep -e ^Settings -e Speed"
echo ""
clush -a -B "`which lspci` | grep -i ether"
echo ""
echo ""
# probe for disk info #
echo -e "${green}Storage ${NC}"
clush -a -B "echo 'Storage Controller: '; `which lspci` | grep -i -e \ raid -e
storage -e lsi"
echo ""
clush -a -B "dmesg | grep -i raid | grep -i scsi"
echo ""
clush -a -B "lsblk -id | awk '{print \$1,\$4}'|sort | nl"
echo ""
echo ""

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

```

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

2. Change permissions to executable:

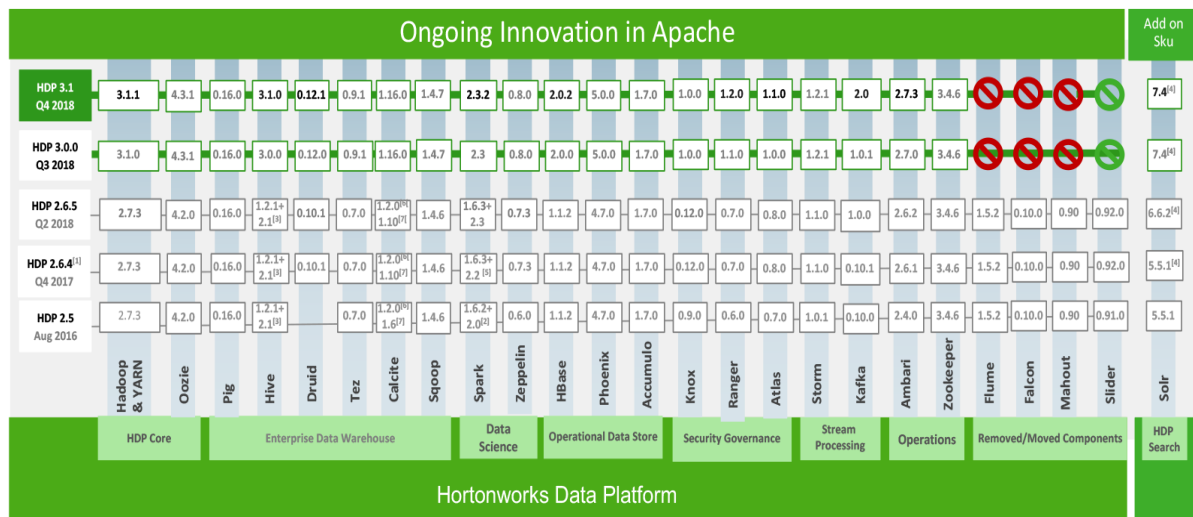
```
# chmod 755 cluster_verification.sh
```

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

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

Install HDP 3.1.0

HDP is an enterprise grade, hardened Hadoop distribution. HDP combines Apache Hadoop and its related projects into a single tested and certified package. HDP 3.1.0 components are depicted in below. This section details how to install HDP 3.1.0 on the cluster.



[1] HDP 2.6 - Shows current Apache branches being used. Final component version subject to change based on Apache release process.
 [2] Spark 1.6.3+ Spark 2.1 - HDP 2.6 supports both Spark 1.6.3 and Spark 2.1 as GA.
 [3] Hive 2.1 is GA within HDP 2.6.
 [4] Apache Solr is available as an add-on product HDP Search.
 [5] Spark 2.2 is GA

Prerequisites for HDP Installation

This section details the prerequisites for the HDP installation, such as setting up HDP repositories.

Hortonworks Repository

To set up the Hortonworks repository, follow these steps:

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



If the admin node is connected to the internet via outbound NAT, repositories can be downloaded directly into the admin node.

```
# mkdir -p /tmp/Hortonworks/
# cd /tmp/Hortonworks
```

2. Download Hortonworks HDP repo:

```
# wget http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/HDP-3.1.0.0-centos7-rpm.tar.gz
--2018-10-13 11:02:02-- http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/HDP-3.1.0.0-centos7-rpm.tar.gz
Resolving public-repo-1.hortonworks.com (public-repo-1.hortonworks.com)...
13.35.121.86, 13.35.121.14, 13.35.121.127, ...
Connecting to public-repo-1.hortonworks.com (public-repo-1.hortonworks.com)|13.35.121.86|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8964079720 (8.3G) [application/x-tar]
Saving to: 'HDP-3.1.0.0-centos7-rpm.tar.gz'

100%[=====] 8,964,079,720 50.3MB/s  in 2m 42s

2018-10-13 11:04:44 (52.9 MB/s) - 'HDP-3.1.0.0-centos7-rpm.tar.gz' saved
[8964079720/8964079720]
```

3. Download Hortonworks HDP-Utils repo:

```
# wget http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/centos7/HDP-UTILS-1.1.0.22-centos7.tar.gz
--2018-10-13 11:05:30-- http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/centos7/HDP-UTILS-1.1.0.22-centos7.tar.gz
Resolving public-repo-1.hortonworks.com (public-repo-1.hortonworks.com)...
13.35.121.86, 13.35.121.127, 13.35.121.14, ...
Connecting to public-repo-1.hortonworks.com (public-repo-1.hortonworks.com)|13.35.121.86|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 90606616 (86M) [application/x-tar]
Saving to: 'HDP-UTILS-1.1.0.22-centos7.tar.gz'

100%[=====] 90,606,616 45.0MB/s  in 1.9s

2018-10-13 11:05:33 (45.0 MB/s) - 'HDP-UTILS-1.1.0.22-centos7.tar.gz' saved
[90606616/90606616]
```

4. Download HDP-GPL repo:

```
# wget http://public-repo-1.hortonworks.com/HDP-GPL/centos7/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-centos7-gpl.tar.gz
--2018-10-13 12:10:45-- http://public-repo-1.hortonworks.com/HDP-GPL/centos7/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-centos7-gpl.tar.gz
Resolving public-repo-1.hortonworks.com (public-repo-1.hortonworks.com)...
13.35.121.120, 13.35.121.127, 13.35.121.86, ...
Connecting to public-repo-1.hortonworks.com (public-repo-1.hortonworks.com)|13.35.121.120|:80... connected.
```

```

HTTP request sent, awaiting response... 200 OK
Length: 162054 (158K) [application/x-tar]
Saving to: 'HDP-GPL-3.1.0.0-centos7-gpl.tar.gz'

100%[=====
=====>] 162,054      --.-K/s   in 0.1s

2018-10-13 12:10:45 (1.27 MB/s) - 'HDP-GPL-3.1.0.0-centos7-gpl.tar.gz' saved
[162054/162054]

```

5. Download the Ambari repo:

```

# wget http://public-repo-
1.hortonworks.com/ambari/centos7/2.x/updates/2.7.1.0/ambari-2.7.1.0-centos7.tar.gz

```

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

```

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

```

7. Extract the tar ball:

```

[root@rhel1 Hortonworks]# tar -zxvf HDP-3.1.0.0-centos7-rpm.tar.gz
[root@rhel1 Hortonworks]# tar -zxvf HDP-UTILS-1.1.0.22-centos7.tar.gz
[root@rhel1 Hortonworks]# tar -zxvf HDP-GPL-3.1.0.0-centos7-gpl.tar.gz
[root@rhel1 Hortonworks]# tar -zxvf ambari-2.7.1.0-centos7.tar.gz

```

8. Create HDP repo with the following contents:

```

[root@rhel1]# cat /etc/yum.repos.d/hdp.repo
[HDP-3.1.0.0]
name= Hortonworks Data Platform Version - HDP-3.1.0.0
baseurl= http://rhel1.hdp3.cisco.com/Hortonworks/HDP/centos7/3.1.0.0-187
gpgcheck=0
enabled=1
priority=1

[HDP-GPL-3.1.0.0]
name=Hortonworks GPL Version - HDP-GPL-3.1.0.0
baseurl= http://rhel1.hdp3.cisco.com/Hortonworks/HDP-GPL/centos7/3.1.0.0-187
gpgcheck=0
enabled=1
priority=1

[HDP-UTILS-1.1.0.22]
name=Hortonworks Data Platform Utils Version - HDP-UTILS-1.1.0.22
baseurl= http://rhel1.hdp3.cisco.com/Hortonworks/HDP-UTILS/centos7/1.1.0.22
gpgcheck=0
enabled=1
priority=1

```



To verify the files, go to: <http://rhel1.hdp3.cisco.com/Hortonworks>.

9. Create the Ambari repo:

```

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

[Updates-ambari-2.7.1.0]
name=ambari-2.7.1.0 - Updates
baseurl=http://rhel1.hdp3.cisco.com/Hortonworks/ambari/centos7/2.7.1.0-169
gpgcheck=0
enabled=1
priority=1
From the admin node copy the repo files to /etc/yum.repos.d/ of all the nodes of the
cluster:
# ansible nodes -m copy -a "src=/etc/yum.repos.d/hdp.repo dest=/etc/yum.repos.d/."
# ansible nodes -m copy -a "src=/etc/yum.repos.d/ambari.repo
dest=/etc/yum.repos.d/."

```

Downgrade Snappy on All Nodes

Downgrade snappy on all data nodes by running this command from admin node:

```
# ansible all -m command -a "yum -y downgrade snappy"
```

HDP Installation

To install HDP, complete the following the steps:

Install and Setup Ambari Server on rhel1

1. Run the following command in rhel1 to install ambari-server:

```

#yum -y install ambari-server
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-
manager to register.
Resolving Dependencies
--> Running transaction check
---> Package ambari-server.x86_64 0:2.7.1.0-169 will be installed
--> Processing Dependency: postgresql-server >= 8.1 for package: ambari-server-
2.7.1.0-169.x86_64
--> Running transaction check
---> Package postgresql-server.x86_64 0:9.2.23-3.el7_4 will be installed
--> Processing Dependency: postgresql-libs(x86-64) = 9.2.23-3.el7_4 for package:
postgresql-server-9.2.23-3.el7_4.x86_64
--> Processing Dependency: postgresql(x86-64) = 9.2.23-3.el7_4 for package:
postgresql-server-9.2.23-3.el7_4.x86_64
--> Processing Dependency: libpq.so.5() (64bit) for package: postgresql-server-
9.2.23-3.el7_4.x86_64
--> Running transaction check
---> Package postgresql.x86_64 0:9.2.23-3.el7_4 will be installed
---> Package postgresql-libs.x86_64 0:9.2.23-3.el7_4 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
=====

```

Package Repository	Arch	Size	Version
Installing:			
ambari-server	x86_64		2.7.1.0-169
Updates-ambari-2.7.1.0		353 M	
Installing for dependencies:			
postgresql	x86_64		9.2.23-
3.el7_4	rhel7.6		3.0 M
postgresql-libs	x86_64		9.2.23-
3.el7_4	rhel7.6		234 k
postgresql-server	x86_64		9.2.23-
3.el7_4	rhel7.6		3.8 M
Transaction Summary			
=====			
Install 1 Package (+3 Dependent packages)			
Total download size: 360 M			
Installed size: 452 M			
Downloading packages:			
(1/4): postgresql-libs-9.2.23-3.el7_4.x86_64.rpm			
234 kB 00:00:00			
(2/4): postgresql-9.2.23-3.el7_4.x86_64.rpm			
3.0 MB 00:00:00			
(3/4): postgresql-server-9.2.23-3.el7_4.x86_64.rpm			
3.8 MB 00:00:00			
(4/4): ambari-server-2.7.1.0-169.x86_64.rpm			
353 MB 00:00:03			

Total			
109 MB/s 360 MB 00:00:03			
Running transaction check			
Running transaction test			
Transaction test succeeded			
Running transaction			
Installing : postgresql-libs-9.2.23-3.el7_4.x86_64			
1/4			
Installing : postgresql-9.2.23-3.el7_4.x86_64			
2/4			
Installing : postgresql-server-9.2.23-3.el7_4.x86_64			
3/4			
Installing : ambari-server-2.7.1.0-169.x86_64			
4/4			
Verifying : postgresql-9.2.23-3.el7_4.x86_64			
1/4			
Verifying : postgresql-libs-9.2.23-3.el7_4.x86_64			
2/4			
Verifying : postgresql-server-9.2.23-3.el7_4.x86_64			
3/4			
Verifying : ambari-server-2.7.1.0-169.x86_64			
4/4			
Installed:			


```
ambari-server.x86_64 0:2.7.1.0-169
```

Dependency Installed:

```
postgresql.x86_64 0:9.2.23-3.e17_4          postgresql-libs.x86_64 0:9.2.23-3.e17_4
          postgresql-server.x86_64 0:9.2.23-3.e17_4
```

Complete!

Install PostgreSQL in rhel2

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

The rhel2 hosts the Hive and Oozie services and Ambari Server is installed on rhel1. To install, follow these steps:

2. Log into rhel2.

3. Install Red Hat Package Manager (RPM) according to the requirements of your operating system:

```
yum install https://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-redhat96-9.6-3.noarch.rpm
```

4. Install PostgreSQL version 9.5 or later:

```
yum install postgresql96-server postgresql96-contrib postgresql96
```

5. Initialize the database as shown in the below figure by running the following command:

```
/usr/pgsql-9.6/bin/postgresql96-setup initdb
```

```
[root@rhel2 ~]#
[root@rhel2 ~]#
[root@rhel2 ~]# /usr/pgsql-9.6/bin/postgresql96-setup initdb
Initializing database ... OK
[root@rhel2 ~]#
```

6. Start PostgreSQL:

```
# systemctl enable postgresql-9.6.service
# systemctl start postgresql-9.6.service
```

7. Open `/var/lib/pgsql/9.6/data/postgresql.conf` and update to the following:

```
listen_addresses = '*'
```

8. Update these files on rhel2 in the location chosen to install the databases for Hive, Oozie and Ambari, using the host ip addresses:

```
[root@rhel2 ~]# cat /var/lib/pgsql/9.6/data/pg_hba.conf|tail -n 20
# TYPE  DATABASE        USER            ADDRESS                 METHOD
# "local" is for Unix domain socket connections only
#local  all             all             peer
# IPv4 local connections:
#host   all             all             127.0.0.1/32           ident
# IPv6 local connections:
host   all             all             ::1/128                ident
# Allow replication connections from localhost, by a user with the
```

```
# replication privilege.
#local replication postgres peer
#host replication postgres 127.0.0.1/32 ident
#host replication postgres ::1/128 ident

local all postgres peer
local all all md5
host all postgres,hive,oozie 10.16.1.32/24 md5
host all ambari 10.16.1.31/24 md5

[root@rhel2 ~]#
```



Before adding new entries, comment the old entries as mentioned above.

9. Restart PostgreSQL:

```
# systemctl stop postgresql-9.6.service
# systemctl start postgresql-9.6.service
```

10. Run the following:

```
sudo -u postgres psql
```

```
[root@rhel2 ~]#
[root@rhel2 ~]# sudo -u postgres psql
could not change directory to "/root": Permission denied
psql (9.6.10)
Type "help" for help.

postgres=# \q
[root@rhel2 ~]#
```



For more information about setting up PostgreSQL, go to:

https://docs.hortonworks.com/HDPDocuments/Ambari-2.7.1.0/bk_ambari-installation/content/install-postgres.html

Create Database for Ambari

To create the database for Ambari, follow these steps:

1. Run the following commands mentioned below in bold to create and prepare database for Ambari:

```
[root@rhel2 ~]# sudo -u postgres psql
could not change directory to "/root": Permission denied
psql (9.6.10)
Type "help" for help.

postgres=# \dt
No relations found.
postgres=#
postgres=#
postgres=# create database ambari;
CREATE DATABASE
postgres=# create user ambari with password 'bigdata';
CREATE ROLE
```

```

postgres=# grant all privileges on database ambari to ambari;
GRANT
postgres=# \connect ambari;
You are now connected to database "ambari" as user "postgres".
ambari=# create schema ambari authorization ambari;
CREATE SCHEMA
ambari=# alter schema ambari owner to ambari;
ALTER SCHEMA
ambari=# alter role ambari set search_path to 'ambari', 'public';
ALTER ROLE
ambari=# \q

```

2. Restart PostgreSQL:

```
[[root@rhel2 ~]# systemctl restart postgresql-9.6.service
```

3. Verify the ambari user by logging into psql:

```
# psql -U ambari -d ambari
```

```

[root@rhel2 ~]#
[root@rhel2 ~]# psql -U ambari -d ambari
psql (9.6.10)
Type "help" for help.

ambari=> █

```

4. Load the Ambari Server database schema:



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

5. Find the Ambari-DDL-Postgres-CREATE.sql file in the /var/lib/ambari-server/resources/ directory of the Ambari Server host after you have installed Ambari Server.

```
Copy /var/lib/ambari-server/resources/ from rhel1 to rhel2:/tmp/.
[root@rhel1 ~]# scp -r /var/lib/ambari-server/resources/* rhel2:/tmp/
```

6. Run the following command to launch the Ambari-DDL-Postgres-CREATE.sql script:

```

[root@rhel2 tmp]# cd /tmp

[root@rhel2 tmp]# psql -U ambari -d ambari
Password for user ambari:
psql (9.6.10)
Type "help" for help.

ambari=> \i Ambari-DDL-Postgres-CREATE.sql
CREATE TABLE
CREATE TABLE
CREATE TABLE
..... OUTPUT OMITTED ----

```

7. Check the table is created by running \dt command:

```
[root@rhel2 ~]# psql -U ambari -d ambari
psql (9.6.10)
Type "help" for help.

ambari=>
ambari=>
ambari=> \dt

      List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
ambari | adminpermission | table | ambari
ambari | adminprincipal | table | ambari
ambari | adminprincipaltype | table | ambari
ambari | adminprivilege | table | ambari
ambari | adminresource | table | ambari
ambari | adminresourcetype | table | ambari
ambari | alert_current | table | ambari
ambari | alert_definition | table | ambari
ambari | alert_group | table | ambari
ambari | alert_group_target | table | ambari
ambari | alert_grouping | table | ambari
ambari | alert_history | table | ambari
```

8. Restart PostgreSQL:

```
[root@rhel2 ~]# systemctl restart postgresql-9.6.service
```

Create Database for Hive

Run the following command as shown in bold to create and prepare database for Hive:

```
[root@rhel2 ~]# sudo -u postgres psql
could not change directory to "/root": Permission denied
psql (9.6.10)
Type "help" for help.

postgres=# create database hive;
CREATE DATABASE
postgres=# create user hive with password 'bigdata';
CREATE ROLE
postgres=# grant all privileges on database hive to hive;
GRANT
postgres=# \q
[root@rhel2 ~]#
```

Create Database for Oozie

Run the following command to create and prepare database for Oozie:

```
[root@rhel2 ~]# sudo -u postgres psql
could not change directory to "/root": Permission denied
psql (9.6.10)
Type "help" for help.

postgres=# create database oozie;
CREATE DATABASE
postgres=# create user oozie with password 'bigdata';
CREATE ROLE
postgres=# grant all privileges on database oozie to oozie;
GRANT
postgres=# \q
```

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

Setup Ambari Server On Admin Node (Rhel1)

To setup the Ambari server, follow these steps:

1. Install the PostgreSQL JDBC driver:

```
[root@rhel1 2.7.1.0-169]# yum -y install postgresql-jdbc*
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-
manager to register.
Resolving Dependencies
--> Running transaction check
---> Package postgresql-jdbc.noarch 0:9.2.1002-5.el7 will be installed
--> Processing Dependency: jpackage-utils for package: postgresql-jdbc-9.2.1002-
5.el7.noarch
--> Processing Dependency: java for package: postgresql-jdbc-9.2.1002-5.el7.noarch
--> Running transaction check
---> Package java-1.8.0-openjdk.x86_64 1:1.8.0.161-2.b14.el7 will be installed
```

2. Configure Ambari server to use the JDBC driver for connectivity to Ambari database in PostgreSQL:

```
[root@rhel1 2.7.1.0-169]# ambari-server setup --jdbc-db=postgres --jdbc-
driver=/usr/share/java/postgresql-jdbc.jar
Using python /usr/bin/python
Setup ambari-server
Copying /usr/share/java/postgresql-jdbc.jar to /var/lib/ambari-
server/resources/postgresql-jdbc.jar
If you are updating existing jdbc driver jar for postgres with postgresql-jdbc.jar.
Please remove the old driver jar, from all hosts. Restarting services that need the
driver, will automatically copy the new jar to the hosts.
JDBC driver was successfully initialized.
Ambari Server 'setup' completed successfully.
```

3. Setup Ambari Server by running the following command:

```
[root@rhel1 ~]# ambari-server setup -j $JAVA_HOME
Using python /usr/bin/python
Setup ambari-server
Checking SELinux...
SELinux status is 'disabled'
Customize user account for ambari-server daemon [y/n] (n)? n
Adjusting ambari-server permissions and ownership...
Checking firewall status...
Checking JDK...
WARNING: JAVA_HOME /usr/java/jdk1.8.0_181-amd64 must be valid on ALL hosts
WARNING: JCE Policy files are required for configuring Kerberos security. If you
plan to use Kerberos, please make sure JCE Unlimited Strength Jurisdiction Policy
Files are valid on all hosts.
Check JDK version for Ambari Server...
JDK version found: 8
Minimum JDK version is 8 for Ambari. Skipping to setup different JDK for Ambari
Server.
Checking GPL software agreement...
```

```

GPL License for LZO: https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html
Enable Ambari Server to download and install GPL Licensed LZO packages [y/n] (n)? y
Completing setup...
Configuring database...
Enter advanced database configuration [y/n] (n)? y
Configuring database...
=====
Choose one of the following options:
[1] - PostgreSQL (Embedded)
[2] - Oracle
[3] - MySQL / MariaDB
[4] - PostgreSQL
[5] - Microsoft SQL Server (Tech Preview)
[6] - SQL Anywhere
[7] - BDB
=====
Enter choice (4):
Hostname (rhel2.hdp3.cisco.com):
Port (5432):
Database name (ambari):
Postgres schema (ambari):
Username (ambari):
Enter Database Password (bigdata):
Configuring ambari database...
Configuring remote database connection properties...
WARNING: Before starting Ambari Server, you must run the following DDL against the
database to create the schema: /var/lib/ambari-server/resources/Ambari-DDL-Postgres-
CREATE.sql
Proceed with configuring remote database connection properties [y/n] (y)? y
Extracting system views...
.....
Adjusting ambari-server permissions and ownership...
Ambari Server 'setup' completed successfully.

```

4. Start the Ambari Server:

```
[root@rhell ~]# ambari-server start
```

```

[root@rhell ~]# ambari-server start
Using python /usr/bin/python
Starting ambari-server
Ambari Server running with administrator privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Ambari database consistency check started...
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.....
Server started listening on 8080

DB configs consistency check: no errors and warnings were found.
Ambari Server 'start' completed successfully.
[root@rhell ~]# ^C
[root@rhell ~]# █

```

5. To check status of Ambari Server, run the following command:

```
# ambari-server status
```

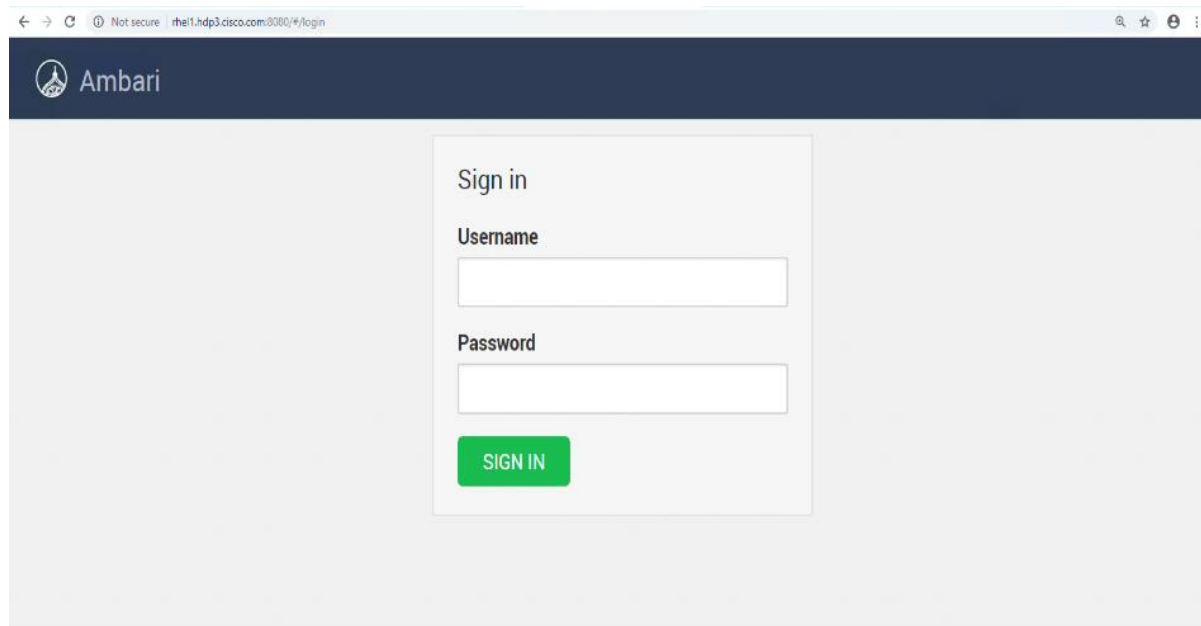
```
[root@rhell ~]# ambari-server status
Using python /usr/bin/python
Ambari-server status
Ambari Server running
Found Ambari Server PID: 65658 at: /var/run/ambari-server/ambari-server.pid
[root@rhell ~]#
```

Launch the Ambari Server

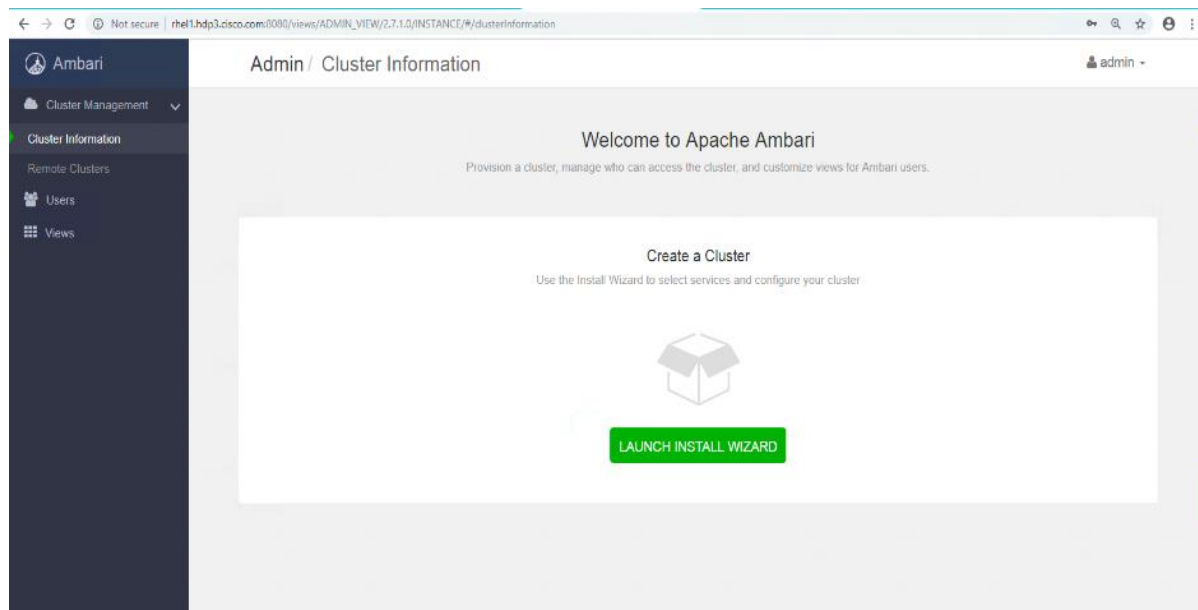
When the Ambari service starts, access the Ambari Install Wizard through the browser. To launch the Ambari server, follow these steps:

1. Point the browser to <http://<ip address for rhel1>:8080> or <http://rhel1.hdp3.cisco.com:8080>

The Ambari Login screen opens.



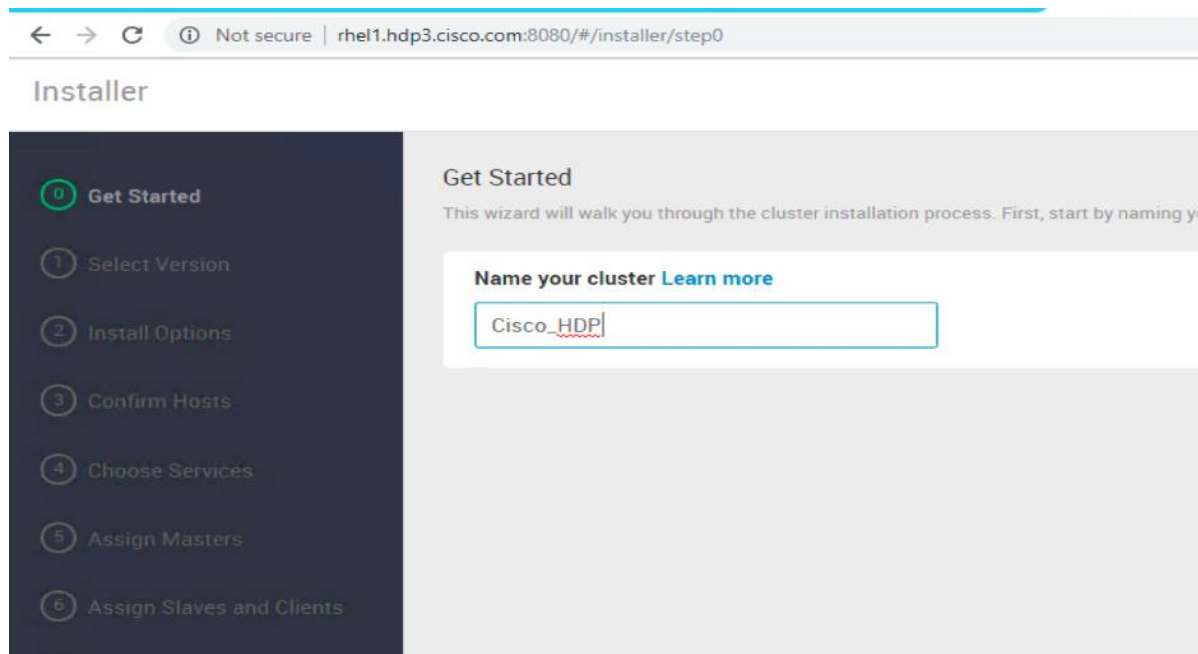
2. Log into the Ambari Server using the default username/password: admin/admin. This can be changed later.
3. When logged in the "Welcome to Apache Ambari" window opens.



Create the Cluster

To create the cluster, follow these steps:

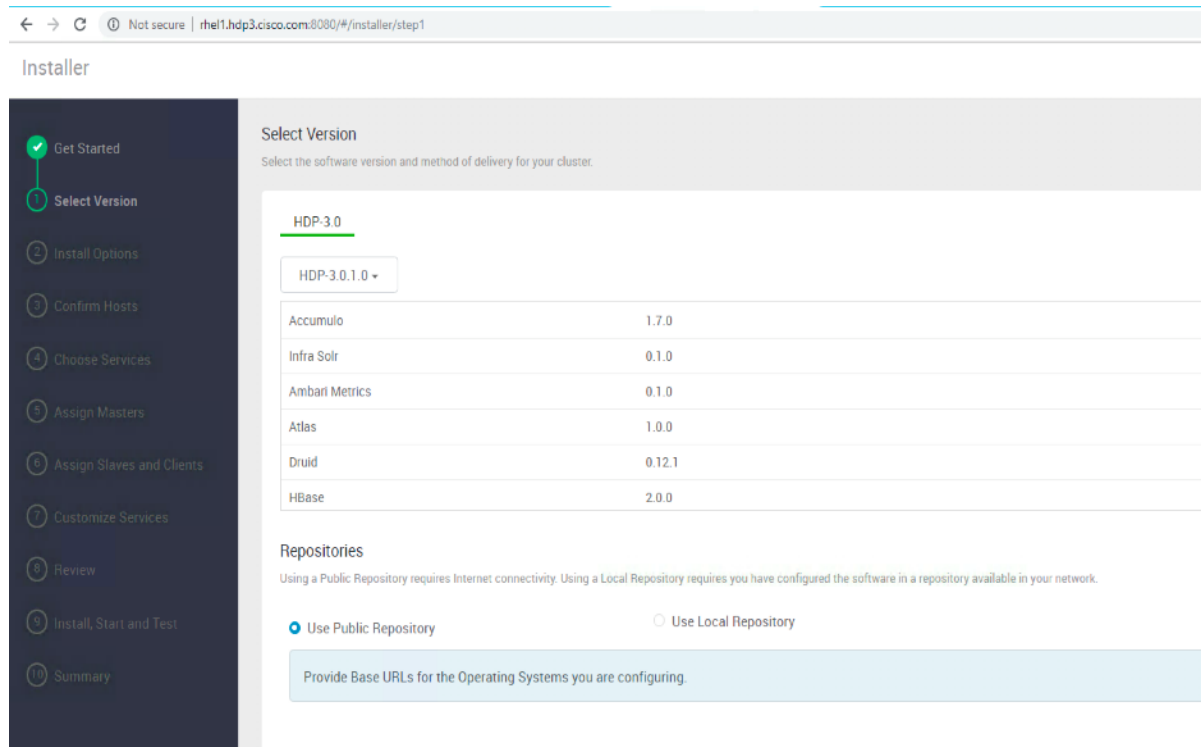
1. To create a cluster, click “LAUNCH INSTALL WIZARD.”
2. From the Get started page type “Cisco_HDP” for the name for the cluster.
3. Click Next.



Select Version

To select the version, follow these steps:

1. In the Select Version section, choose the HDP 3.1.0. version.



Installer

Select Version
Select the software version and method of delivery for your cluster.

HDP-3.0

HDP-3.0.1.0

Accumulo	1.7.0
Infra Solr	0.1.0
Ambari Metrics	0.1.0
Atlas	1.0.0
Druid	0.12.1
HBase	2.0.0

Repositories
Using a Public Repository requires Internet connectivity. Using a Local Repository requires you have configured the software in a repository available in your network.

Use Public Repository Use Local Repository

Provide Base URLs for the Operating Systems you are configuring.

2. Under Repositories, select “Use Local Repository.”
3. Update the Red Hat 7 HDP-3.0 URL to <http://rhel1.hdp3.cisco.com/Hortonworks/HDP/centos7/3.1.0.0-187/>
4. Update the Red Hat 7 HDP-3.0-GPL URL to <http://rhel1.hdp3.cisco.com/Hortonworks/HDP-GPL/centos7/3.1.0.0-187/>
5. Update the Red Hat 7 HDP-UTILS-1.1.0.22 to <http://rhel1.hdp3.cisco.com/Hortonworks/HDP-UTILS/centos7/1.1.0.22/>

	HDP-3.0	http://rhel1.hdp3.cisco.com/Hortonworks/HDP/centos7/3.0.1.0-187/
redhat7	HDP-3.0-GPL	http://rhel1.hdp3.cisco.com/Hortonworks/HDP-GPL/centos7/3.0.1.0-187/
	HDP-UTILS-1.1.0.22	http://rhel1.hdp3.cisco.com/Hortonworks/HDP-UTILS/centos7/1.1.0.22/



Make sure there are no trailing spaces after the URLs.

Select Hosts

To build up the cluster, you need to provide the general information about how you want to set up the cluster. This requires providing the Fully Qualified Domain Name (FQDN) of each of the hosts. You also need to provide access

to the private key file that was created in Set Up Password-less SSH; this is used 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 also be used to indicate larger sets of hosts.
2. Select the option Provide your SSH Private Key in the Ambari cluster install wizard:

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

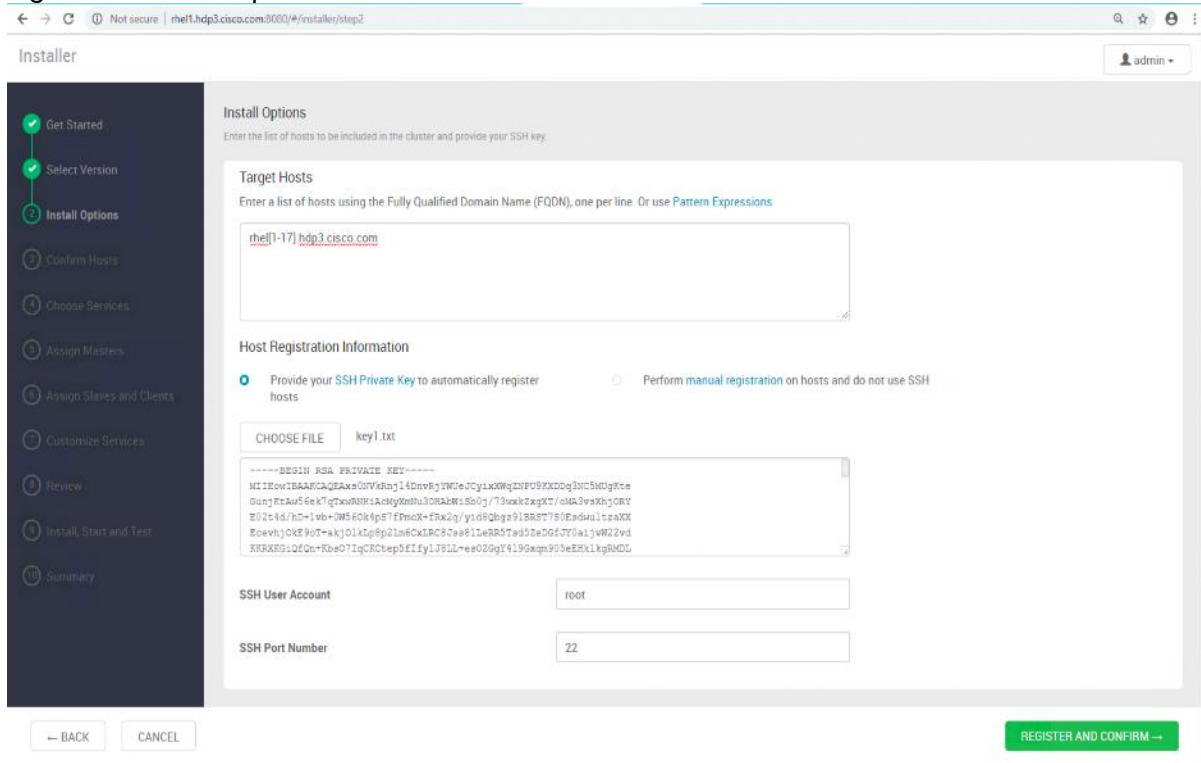


Make sure there is no extra white space after the text-----END RSA PRIVATE KEY-----

```
[root@rhell ~]# cat /root/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCAQEAYDOIRbk4mBZrIzc0/gOM2iYT2h4vXkIXA/uvQVPthFreUdgt
Zehw/Qtdk7meeqhgqsHmb1CriF0m6SxvPEXW2cGoAx75hZwtUDIR3Qlvk6oYUmdW
BKq5TMfUMKfD7tknkGkg5N+YHsPCoNILLz/Wqc0lhZ0tiCmrxeRnPGS1JY74/Db
AOBewMuNajAoVppPD6cLGF6/NKORpEDUnCuwe5pCRV5tko+gzBeBF5oeCS6Ya6I7
ns0HplJXV0mv23SNUwl3cswbqLdrr3atG6YrieVrmmr/PlrKmp192tzQ1mHZMBqG
w1RJTILjygWogp5g7NqBGeM7sX4V6omzv4vmzwIBIwKCAQEAg4+UEI+o2PjKVCuX
2h+XEwMUXCJ3KoNEYBpr2nj7KxckYas/8oLN6B1pYR0UB3X2Y2Vc6hBwuLI+JDMk
hrGNMALqWdjthU1oyX/9HDlmlDyTo9k8LvPY2q8zqvHnJ+3Jis192Dspc01xRRxQ
wnpofjAmlCDx5Wxp4MZyX9HynCcKmhFefobLys6glox84eHW1y6boxU1dh7hsQ
pck+xpDfW1shYFbvckTUCHUAezF4+uBT5F0PMiD7PwzrvbXKA65ABuezv9gg2/I1
FekIkRvbosniFbBUi2ZOS1uN/gsaZgmsQ9gTarJlV8zMy6K31LETc0ck12LzHRX2
5sEx6wKBgQD9CiKc0HFiuLrQWW5cLTDJU8wzTiNK4M91Qb2LohfFuZfluiA13Ref
yiL9MjE3A5Mnn9pcRxxMmXXPF4t9iuLh3+3tCsr1TzPml4WT+Fipa9sh+3JZ2HKgm
pCquAEdoFRK4oP3/yYQg95gie2SC9sB0z6zVohdyNUvnkiMb9vwi3wKBgQDKiyTi
Yu421owsYKfz7YjomjRKUFaH4CKtnyJy1SM3wFPRnZJd4BUAmQ0Datxr2tW4si+4
t88M8XS6FHGHymSqrL0tYzMLmmwUtjCLN2QfQSeg1NovekXxXL0iUzel8PL3ZOH
Aebj0/GLQ3SF/PgWmOkCwnTajov/x1dBdiSQEQKBGEERPbmX8UVF3Nz9ZyVqtMYO
09KtsU3Ex52x0ad1Vpht5TSSmolkvO6TEE+8cw4lfzX5j+vXwxh+bjozBj30/Dwc
GGGbrQbrkKscs5HLL3Z5+QqtWepB4hiQnUKvnVVHP1QMJA6S53YxCdz7KHlypnqQ
bkWQFKhw2QEiUivDKuR1AoGASzr/EkiAtufFb5GdbjOn4V3Y6Gb7ky3DvNS1BhSm
rk7ADAdTnzX5N23L08gAf9Tws+ppfx+zTfNI NOMFmNY1Y9EpyJs0S/1adLEOrOwu
sC8J8bu/5RNWk8z+z9s5zwUrd5txT2cY1J8t1KQgtWYUPxoVoe/ccfENA5LP872S
xnsCgYAFRE4SbB416p9miRl+gNCiihm9N+FmHmMcP/y80QL/MoAYoHB1Tn8cwVu
l+sju4bWGUzvnGMWxwpeU5zVBra+yShh309IwjP/1kpCNWz7CX+/uI6FY+slZxTr
t5P/AvhOvUKMhRFjXFQoY5yqNUkasvIu6S8Q1unl8N2IhEgw1g==
-----END RSA PRIVATE KEY-----
```

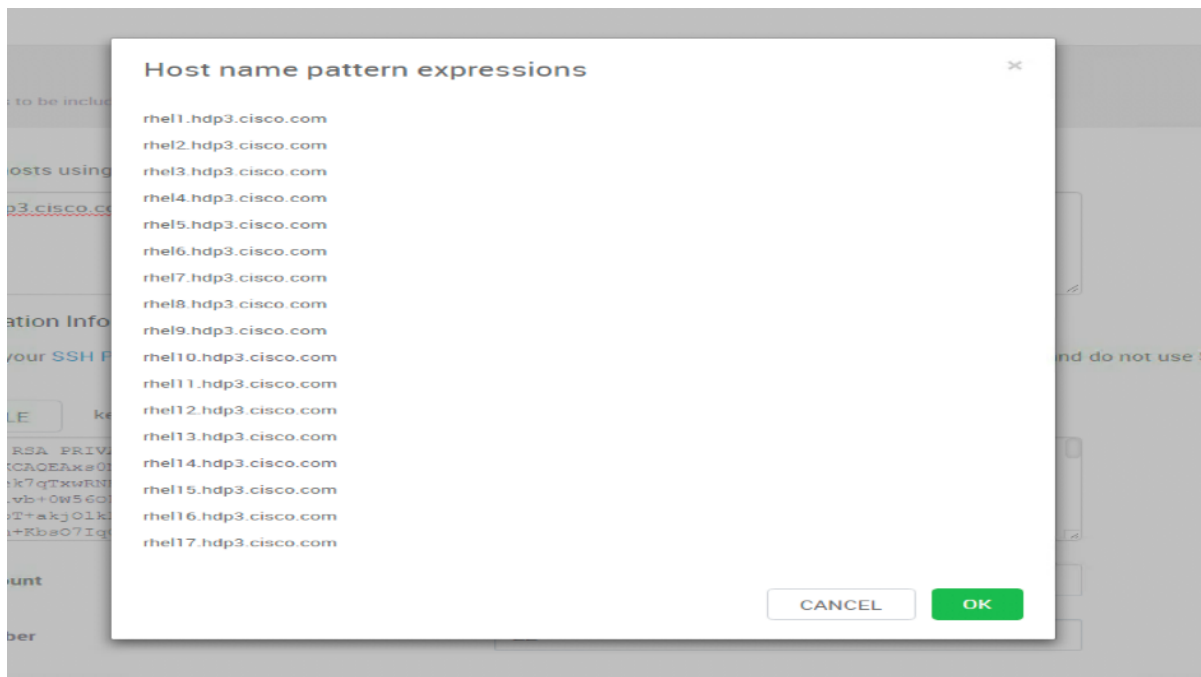
3. Click the Register and Confirm to continue.

Figure 42 Install Options



Hostname Pattern Expressions

1. Click OK on the Host Name Pattern Expressions popup.



Confirm Hosts

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

To confirm host, follow these steps:

1. If any host was selected in error, remove it by selecting the appropriate checkboxes and clicking the grey Remove Selected button.
2. To remove a single host, click the small white Remove button in the Action column.
3. When the list of hosts is confirmed, click Next.

The screenshot shows the Ambari installer interface at the 'Confirm Hosts' step. The browser address bar shows 'rhe11.hdp3.cisco.com:8080/#/installer/step3'. The page title is 'Installer' and the user is logged in as 'admin'. The main content area is titled 'Confirm Hosts' and contains the instruction: 'Please confirm the host list and remove any hosts that you do not want to include in the cluster.' Below this is a table with columns for Host, Progress, Status, and Action. The table lists 11 hosts, all with a 'Success' status and a green progress bar. The 'Action' column contains a small white 'Remove' button for each host. Below the table, there is a yellow bar indicating '16 Other Registered Hosts' and a blue bar with a loading spinner and the text 'Please wait while the hosts are being checked for potential problems...'. At the bottom of the interface, there are 'BACK' and 'CANCEL' buttons on the left, and a green 'NEXT' button on the right.

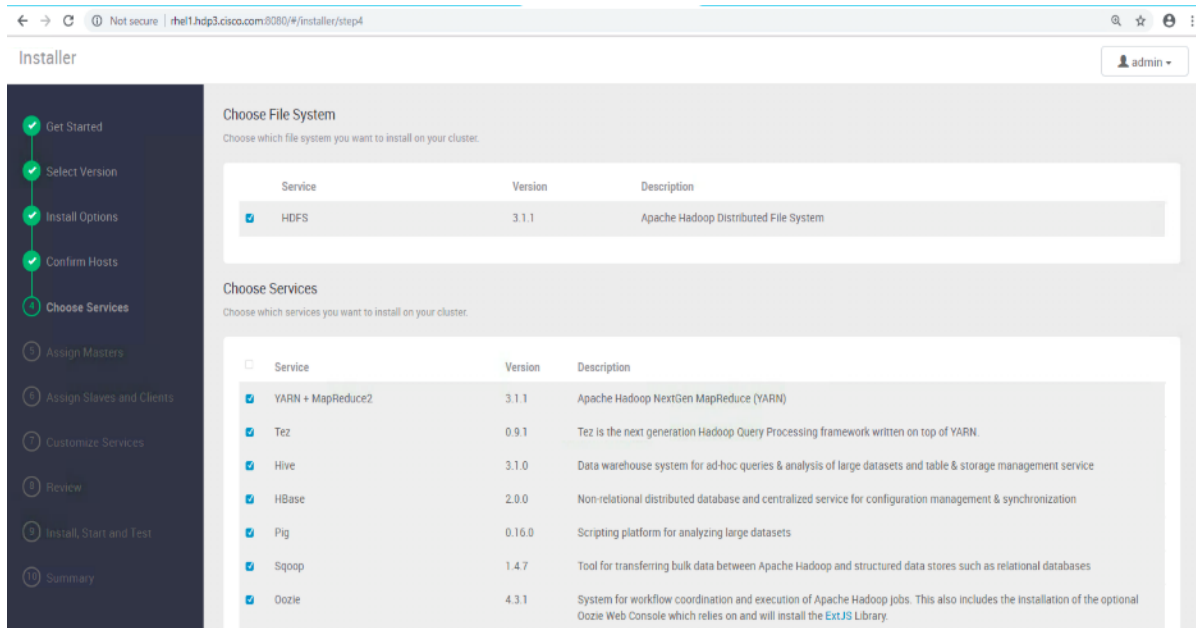
Host	Progress	Status	Action
<input type="checkbox"/> rhe11.hdp3.cisco.com	<div style="width: 100%; height: 10px; background-color: green;"></div>	Success	<input type="button" value="Remove"/>
<input type="checkbox"/> rhe12.hdp3.cisco.com	<div style="width: 100%; height: 10px; background-color: green;"></div>	Success	<input type="button" value="Remove"/>
<input type="checkbox"/> rhe13.hdp3.cisco.com	<div style="width: 100%; height: 10px; background-color: green;"></div>	Success	<input type="button" value="Remove"/>
<input type="checkbox"/> rhe14.hdp3.cisco.com	<div style="width: 100%; height: 10px; background-color: green;"></div>	Success	<input type="button" value="Remove"/>
<input type="checkbox"/> rhe15.hdp3.cisco.com	<div style="width: 100%; height: 10px; background-color: green;"></div>	Success	<input type="button" value="Remove"/>
<input type="checkbox"/> rhe16.hdp3.cisco.com	<div style="width: 100%; height: 10px; background-color: green;"></div>	Success	<input type="button" value="Remove"/>
<input type="checkbox"/> rhe17.hdp3.cisco.com	<div style="width: 100%; height: 10px; background-color: green;"></div>	Success	<input type="button" value="Remove"/>
<input type="checkbox"/> rhe18.hdp3.cisco.com	<div style="width: 100%; height: 10px; background-color: green;"></div>	Success	<input type="button" value="Remove"/>
<input type="checkbox"/> rhe19.hdp3.cisco.com	<div style="width: 100%; height: 10px; background-color: green;"></div>	Success	<input type="button" value="Remove"/>
<input type="checkbox"/> rhe10.hdp3.cisco.com	<div style="width: 100%; height: 10px; background-color: green;"></div>	Success	<input type="button" value="Remove"/>
<input type="checkbox"/> rhe11.hdp3.cisco.com	<div style="width: 100%; height: 10px; background-color: green;"></div>	Success	<input type="button" value="Remove"/>

Choose Services

HDP is made up of several components. Go to Hortonworks [Understand the Basics](#) for more information.

To choose services, follow these steps:

1. Select all to preselect all items.
2. When you have made your selections, click Next.



Assign Masters

The Ambari installation wizard attempts to assign the master nodes for various services that have been selected to appropriate hosts in the cluster, as listed in Table 6. The right column shows the current service assignments by host, with the hostname and its number of CPU cores and amount of RAM indicated.

1. Reconfigure the service assignments to match Table 6 shown below.

Table 6 Reconfigure the Service Assignments

Service Name	Host
NameNode	rhel1, rhel3 (HA)
SNameNode	rhel2
History Server	rhel2
App Timeline Server	rhel2
Resource Manager	rhel2, rhel3 (HA)
Hive Metastore	rhel2
WebHCat Server	rhel2
HiveServer2	rhel2
HBase Master	rhel2
Oozie Server	rhel1
Zookeeper	rhel1, rhel2, rhel3
Spark History Server	rhel2
SmartSense HST Server	rhel1

Service Name	Host
Grafana	rhel1
Atlas Metadata Server	rhel2
Metrics Collector	rhel1

2. Click Next.

Assign Slaves and Clients

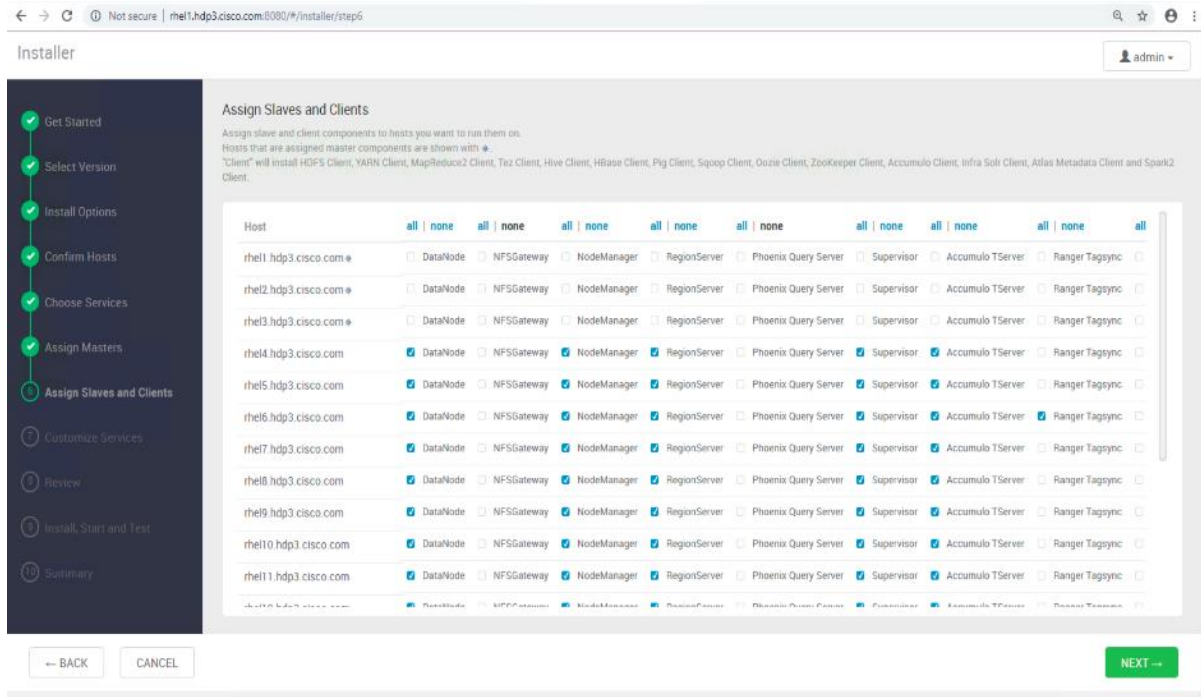
The Ambari install wizard attempts to assign the slave components (DataNodes, NFSGateway, NodeManager, RegionServers, Supervisor, and Client) to appropriate hosts in the cluster.

To assign slaves and clients, follow these steps:

1. Reconfigure the service assignment to match the values shown in Table 7 .
2. Assign DataNode, NodeManager, RegionServer, and Supervisor on nodes rhel3- rhel28.
3. Assign Client to all nodes.
4. Click Next.

Table 7 Services and Hostnames

Client Service Name	Host
DataNode	rhel4-rhel16; rhel24-rhel31
NFSGateway	rhel1
NodeManager	rhel4-rhel16; rhel24-rhel31
RegionServer	rhel4-rhel13; rhel24-rhel31
Supervisor	rhel4-rhel13; rhel24-rhel31
Client	All nodes, rhel1-rhel31
Submarine Nodes	rhel13-rhel16
CDSW Nodes	rhel18-rhel23



Customize Services

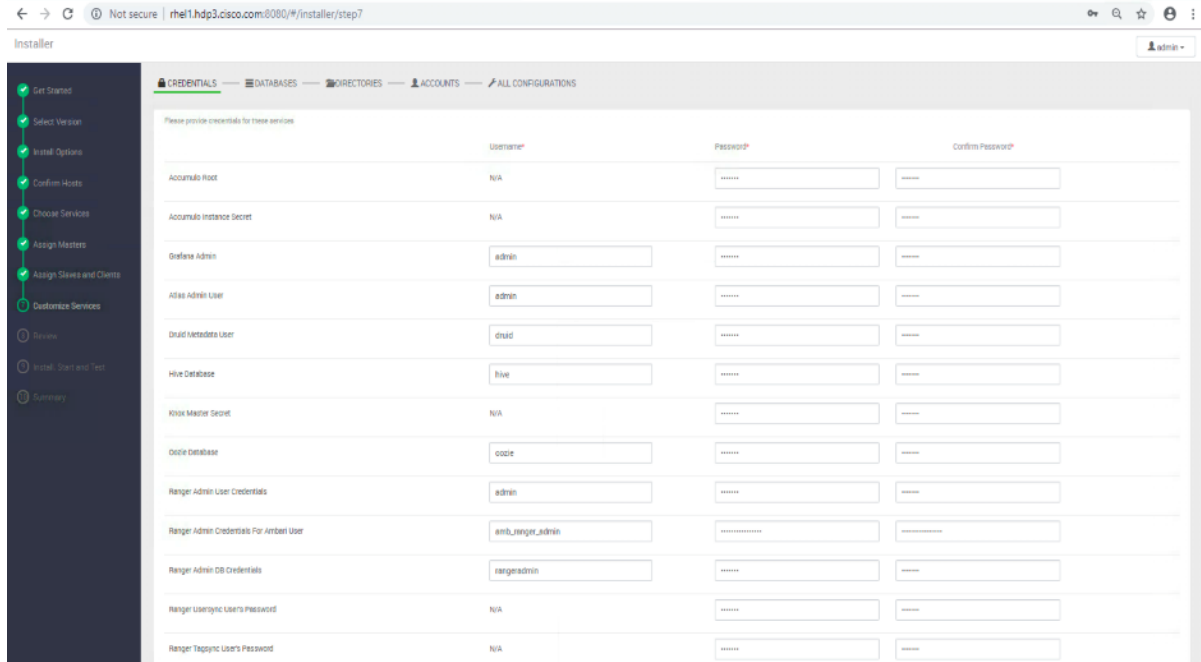
This section shows the tabs that manage configuration settings for Hadoop components. The wizard attempts to set reasonable defaults for each of the options here, but this can be modified to meet specific requirements. The following sections provide configuration guidance that should be refined to meet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

The following changes need to be made:

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

Credentials

Specify the credentials as per your organizational policies for services in CREDENTIALS tab as shown below:



Databases

In the DATABASES tab, to configure the database for DRUID, HIVE, OOZIE, and RANGER, follow these steps:

1. Configure DRUID as shown below:

DRUID META DATA STORAGE
Druid Metadata storage database name

Druid Metadata storage type

Metadata storage user

Metadata storage password

Metadata storage hostname

Metadata storage port

Metadata storage connector uri

2. Change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

druid_log_dir

Druid PID dir

3. Configure HIVE:

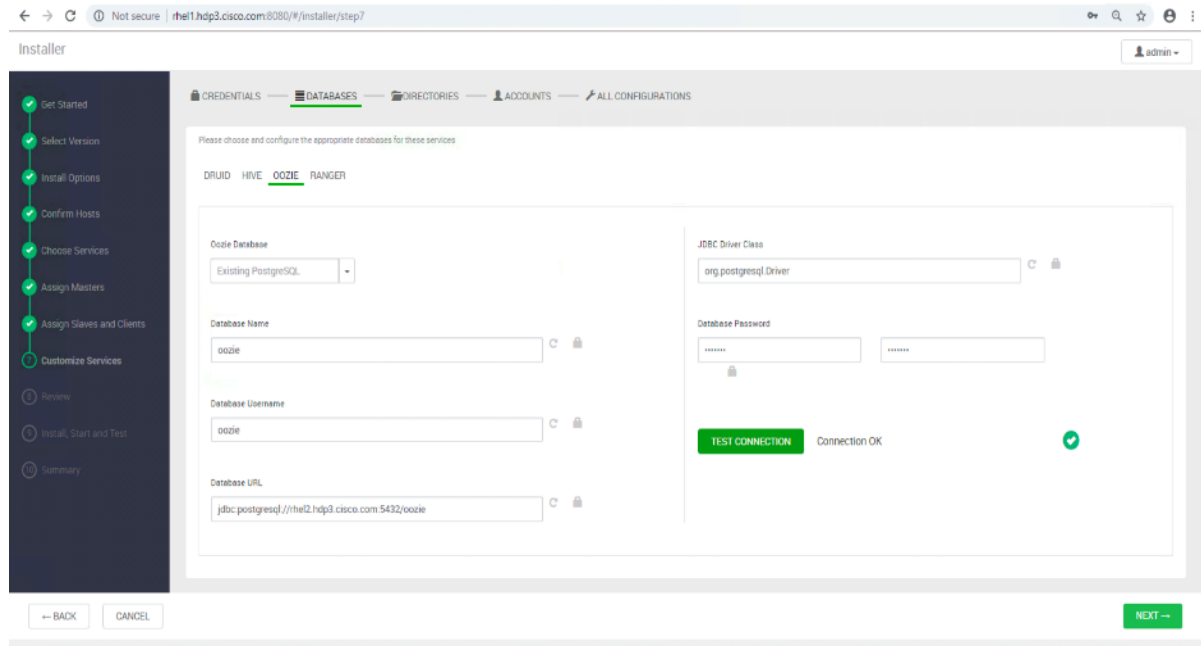
- a. Select Existing PostgreSQL database in the Hive Database drop-down list.
- b. Enter Database Name as hive
- c. Enter Database Username as hive
- d. Enter Database URL as jdbc:postgresql://rhel2.hdp3.cisco.com/hive
- e. Enter Database password (use the password created during hive database setup in earlier steps; for example, bigdata)
- f. Click TEST CONNECTION to verify the connectivity
- g. In Advanced tab, Change the default log location by filtering the Log Dir property and modifying the /var prefix to /data/disk1.
- h. Change the WebHCat log directory by filtering the Log Dir property and modifying the /var prefix to /data/disk1.

The screenshot displays the Ambari Installer interface for configuring HIVE. The 'Databases' tab is active, showing configuration options for HIVE. The 'Hive Database' dropdown is set to 'Existing PostgreSQL'. The 'Hive Database Type' is 'postgres'. The 'JDBC Driver Class' is 'org.postgresql.Driver'. The 'Database Name' is 'hive', the 'Database Username' is 'hive', and the 'Database URL' is 'jdbc:postgresql://rhel2.hdp3.cisco.com:5432/hive'. A 'TEST CONNECTION' button is present, and the status indicates 'Connection OK' with a green checkmark. A sidebar on the left shows the installation progress, and a top navigation bar includes 'CREDENTIALS', 'DATABASES', 'DIRECTORIES', 'ACCOUNTS', and 'ALL CONFIGURATIONS'.

4. Configure OOZIE:

- a. Select Existing PostgreSQL database in the Hive Database drop-down list.
- b. Enter Database Name as oozie.
- c. Enter Database Username as oozie.
- d. Enter Database URL as jdbc:postgresql://rhel2.hdp3.cisco.com/oozie.
- e. Enter Database password (use the password created during hive database setup in earlier steps; for example, bigdata).
- f. Click TEST CONNECTION to verify the connectivity.

- g. In Advanced tab, Change the default log location by filtering the Log Dir property and modifying the /var prefix to /data/disk1.



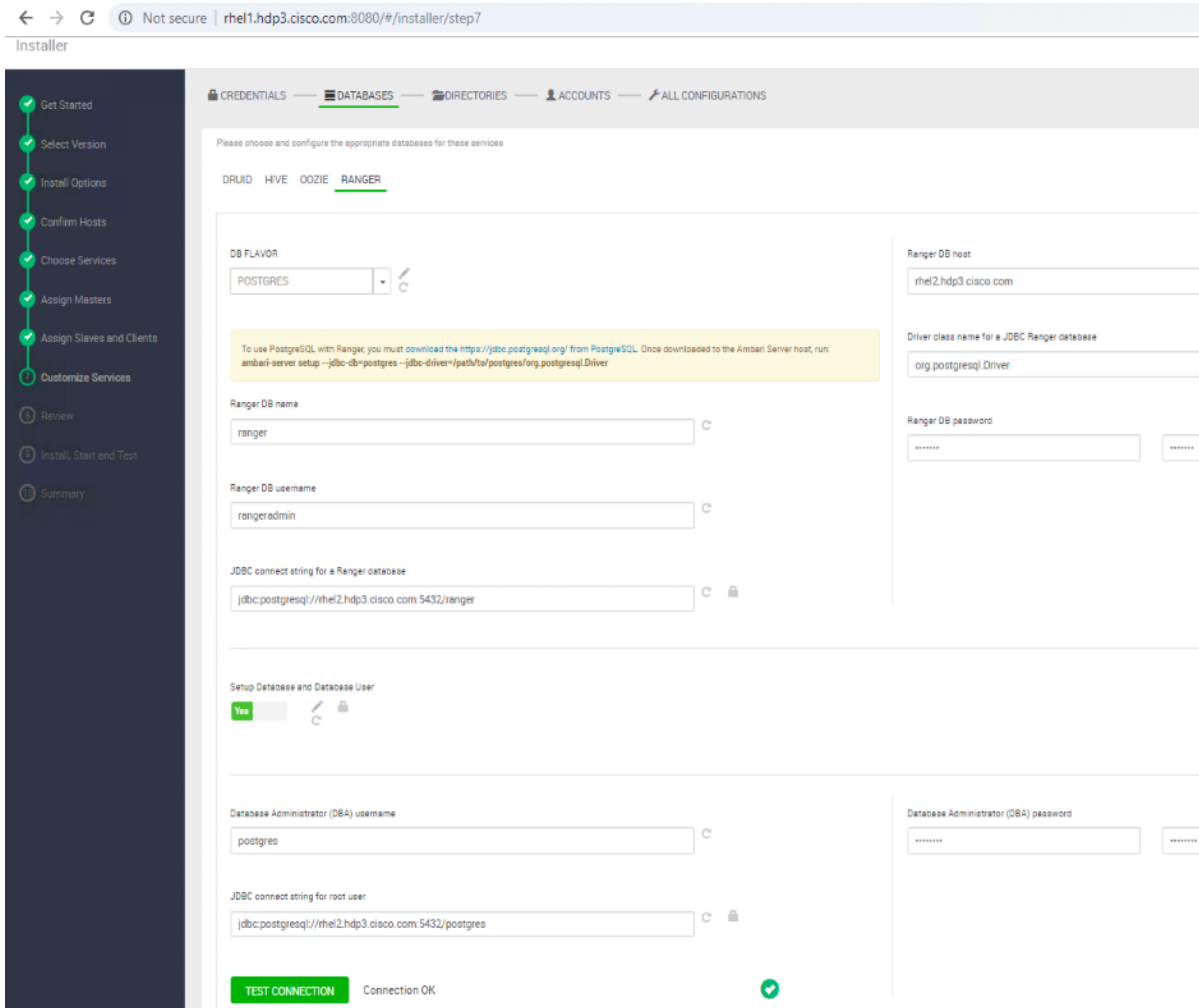
Advanced oozie-env

Oozie Log Dir

/data/disk1/log/oozie

5. Configure RANGER:

- a. Select POSTGRES from DB FLAVOR drop-down list.
- b. Provide a Ranger DB name. For example, ranger.
- c. Provide Ranger DB Username such as rangeradmin.
- d. Enter JDBC connect string for a Ranger Database as jdbc:postgresql://rhel2.dhp3.cisco.com:54323/ranger.
- e. Ranger DB Host as rhel2.hdp3.cisco.com.
- f. Enter Ranger DB password. (Ranger database is not previously created. provide password string that would be configured in the DB. For example, bigdata).
- g. Select "Yes" for Setup Database and Database User.
- h. Enter "postgres" in Database Administrator (DBA) username.
- i. Enter Database Administrator (DBA) password.
- j. Enter jdbc:postgresql://rhel2.hdp3.cisco.com:5432/postgres in JDBC connect string for root user.
- k. Update Ranger Admin Log Dir from /var to /data/disk1.



HDFS

1. In Ambari, select the HDFS Service tab and use the “Search” box to filter for the properties mentioned in Table 8 and update their values.
2. Update the following HDFS configurations in Ambari.

Table 8 HDFS Configurations in Ambari

Property Name	Value
NameNode Java Heap Size	4096
Hadoop maximum Java heap size	4096
DataNode maximum Java heap size	4096
Datanode failed disk toleration	5

3. Change the default log location by filtering the Log Dir property and modifying the /var prefix to /data/disk1.

Installer

admin

CREDENTIALS DATABASES DIRECTORIES ACCOUNTS ALL CONFIGURATIONS

HDFS YARN MAPREDUCE2 TEZ HIVE HBASE PIG SQOOP OOZIE ZOOKEEPER STORM ACCUMULO INFRA SOLR AMBARI METRICS ATLAS KAFKA KN

Default (17) Filter...

SETTINGS ADVANCED

NameNode

NameNode directories

/data/disk1/hadoop/hdfs/namenode

NameNode Java heap size

4096

NameNode Server threads

1400

Minimum replicated blocks %

100%

DataNode

DataNode directories

/data/disk1/hadoop/hdfs/data
/data/disk2/hadoop/hdfs/data
/data/disk3/hadoop/hdfs/data
/data/disk4/hadoop/hdfs/data
/data/disk5/hadoop/hdfs/data
/data/disk6/hadoop/hdfs/data
/data/disk7/hadoop/hdfs/data
/data/disk8/hadoop/hdfs/data

DataNode failed disk tolerance

3

DataNode maximum Java heap size

4096

NameNode

NameNode directories

/data/disk1/hadoop/hdfs/namenode

NameNode Java heap size

4096 MB

NameNode Server threads

1400

DataNode

DataNode directories

/data/disk1/hadoop/hdfs/data/data/disk2/hadoop/hdfs/data/data/disk3/hadoop/hdfs/data/data/disk4/hadoop/hdfs/data/data/disk5/hadoop/hdfs/data/data/disk6/hadoop/hdfs/data/data/disk7/hadoop/hdfs/data/data/disk8

DataNode failed disk tolerance

3

DataNode maximum Java heap size

4096 MB

General

WebHDFS enabled

Hadoop maximum Java heap size

4096

MB

Advanced hadoop-env

Hadoop PID Dir Prefix: /var/run/hadoop

Hadoop Root Logger: INFO,RFA

Hadoop Log Dir Prefix: /data/disk1/log/hadoop

MapReduce2

1. In Ambari, choose the MapReduce Service tab and update the values as shown below.
2. Under the MapReduce2 tab, change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

SETTINGS **ADVANCED**

MapReduce

MapReduce Framework

Map Memory: 4GB

Reduce Memory: 8GB

Sort Allocation Memory: 2047MB

MapReduce AppMaster

AppMaster Memory: 4GB

Advanced mapred-env

Mapreduce Log Dir Prefix: /data/disk1/log/hadoop-mapreduce

Mapreduce PID Dir Prefix: /var/run/hadoop-mapreduce

YARN

1. In Ambari, select the YARN Service, and update the following as shown in Table 9 .

Table 9 YARN Configuration

Property Name	Value
ResourceManager Java heap size	4096
NodeManager Java heap size	2048
YARN Java heap size	4096

Resource Manager

ResourceManager Java heap size MB

Node Manager

NodeManager Java heap size MB

Application Timeline Server

General

YARN Java heap size MB

- Under YARN tab, change the default log location by filtering the Log Dir property and modifying the /var prefix to /data/nvme01 and /data/nvme02.

YARN NodeManager Local directories

YARN NodeManager Log directories



YARN requires other configurations such as config group, node labeling, enabling docker runtime, CPU/GPU scheduling and isolation, and so on, which can be found in section [High Availability for HDFS NameNode and YARN ResourceManager](#).



High Availability for NameNode and YARN Resource Manager can be configured using Ambari or also on non-Ambari clusters. This deployment guide explains the configuration of high availability using Ambari – Use the Ambari wizard interface to configure HA of the components.

The Ambari web UI provides a wizard-driven user experience that allows to configure high availability of the components in many Hortonworks Data Platform (HDP) stack services. The high availability of the components is achieved by setting up primary and secondary components. If the primary component fails or becomes unresponsive, services failover to secondary component. After configuring the high availability for a service, Ambari enables you to manage and disable (roll back) the high availability of components within that service.

Configure the HDFS NameNode High Availability

The HDFS NameNode high availability feature enables you to run redundant NameNodes in the same cluster in an Active/Passive configuration with a hot standby. This eliminates the NameNode as a potential single point of failure (SPOF) in an HDFS cluster. With the release of Hadoop 3.0, you can configure more than one backup NameNode.

Prior to the release of Hadoop 2.0, the NameNode represented a single point of failure (SPOF) in an HDFS cluster. Each cluster had a single NameNode, and if that machine or process became unavailable, the cluster would be unavailable until the NameNode was either restarted or brought up on a separate machine. This situation impacted the total availability of the HDFS cluster in two major ways:

- In the case of an unplanned event such as a machine crash, the cluster would be unavailable until an operator restarted the NameNode.
- Planned maintenance events such as software or hardware upgrades on the NameNode machine would result in periods of cluster downtime.

HDFS NameNode High Availability avoids this by facilitating either a fast failover to one or more backup NameNodes during machine crash, or a graceful administrator-initiated failover during planned maintenance.



Secondary NameNode is not required in high availability configuration because the Standby node also performs the tasks of the Secondary NameNode.

HBase

Under the HBase tab, change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

HBase Log Dir Prefix

Zookeeper

Under the Zookeeper tab, change the default log location by filtering the Log Dir property and modifying the /var prefix to /data/disk1.

Advanced zookeeper-env

Zookeeper Log Dir

Zookeeper PID Dir

Storm

Under the Storm tab, change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

Advanced storm-env

Storm Log directory	<input type="text" value="/data/disk1/log/storm"/>
Storm PID directory	<input type="text" value="/var/run/storm"/>

Ambari Metrics

1. Choose the Ambari Metrics Service and expand the General tab and make the changes shown below.
2. Enter the Grafana Admin password as per organizational policy.
3. Change the default log location for Metrics Collector, Metrics Monitor and Metrics Grafana by finding the Log Dir property and modifying the /var prefix to /data/disk1.
4. Change the default data dir location for Metrics Grafana by finding the data Dir property and modifying the /var prefix to /data/disk1.

General

Metrics Service operation mode	<input type="text" value="embedded"/>
Metrics Collector log dir	<input type="text" value="/data/disk1/log/ambari-metrics-collector"/>
Metrics Collector pid dir	<input type="text" value="/var/run/ambari-metrics-collector"/>
Metrics Monitor log dir	<input type="text" value="/data/disk1/log/ambari-metrics-monitor"/>
Metrics Monitor pid dir	<input type="text" value="/var/run/ambari-metrics-monitor"/>
Grafana Admin Username	<input type="text" value="admin"/>
Grafana Admin Password	<input type="password" value="....."/> <input type="password" value="....."/>

Advanced ams-grafana-env

Metrics Grafana data dir	<input type="text" value="/data/disk1/lib/ambari-metrics-grafana"/>
Metrics Grafana log dir	<input type="text" value="/data/disk1/log/ambari-metrics-grafana"/>
Metrics Grafana pid dir	<input type="text" value="/var/run/ambari-metrics-grafana"/>

Advanced ams-hbase-env

HBase Log Dir Prefix	<input type="text" value="/data/disk1/log/ambari-metrics-collector"/>
----------------------	---

Accumulo

Select Accumulo Service and change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

Advanced accumulo-env

Accumulo Log Dir	<input type="text" value="/data/disk1/log/accumulo"/>
------------------	---

Atlas

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

Advanced atlas-env

Metadata Data directory	<input type="text" value="/var/lib/atlas/data"/>
Metadata Log directory	<input type="text" value="/data/disk1/log/atlas"/>
Metadata PID directory	<input type="text" value="/var/run/atlas"/>

Kafka

Under the Kafka tab, change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

Advanced kafka-env

Kafka Log directory	<input type="text" value="/data/disk1/log/kafka"/>
---------------------	--

Knox

1. Select the Knox Service tab and expand the Knox gateway tab and make the changes shown below.
2. Enter the Knox Master Secret password as per organizational policy.
3. For Knox, change the gateway port to 8444 to ensure no conflicts with local HTTP server.

Knox Gateway

Knox Gateway host	<input type="text" value="rhel1.hdp3.cisco.com"/>
Knox Master Secret	<input type="password" value="....."/> <input type="password" value="....."/>

Advanced gateway-site

gateway.port

8444

SmartSense

The SmartSense account requires the Hortonworks support subscription. Subscribers can populate the properties as shown below:

SmartSense Account	Local Storage
Customer account name <input type="text" value="unspecified"/>	Bundle storage directory <input type="text" value="/var/lib/smartsense/hst-server/data"/>
SmartSense ID <input type="text" value="unspecified"/>	Server temporary data directory <input type="text" value="/var/lib/smartsense/hst-server/tmp"/>
Notification Email <input type="text" value="unspecified"/>	Agent temporary data directory <input type="text" value="/var/lib/smartsense/hst-agent/data/tmp"/>
Enable Flex Subscription <input type="checkbox"/> No	

Spark

Select the Spark tab, change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

Advanced livy2-env

Livy2 Log directory

/data/disk1/log/livy2

Livy2 PID directory

/var/run/livy2

Advanced spark2-env

Spark Log directory

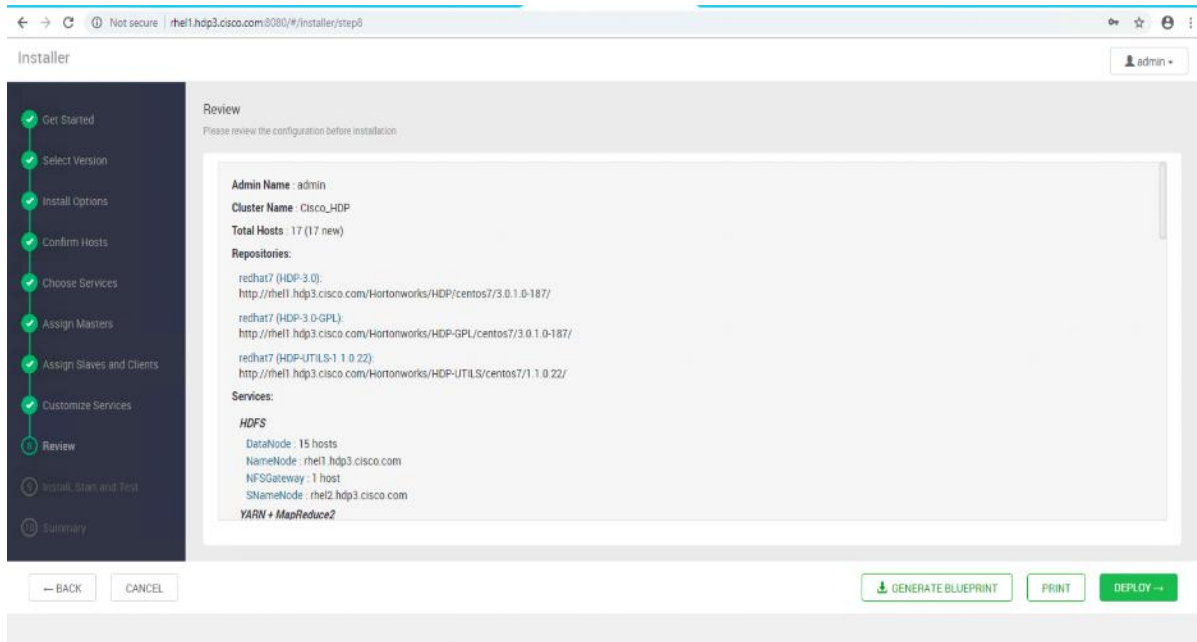
/data/disk1/log/spark2

Spark PID directory

/var/run/spark2

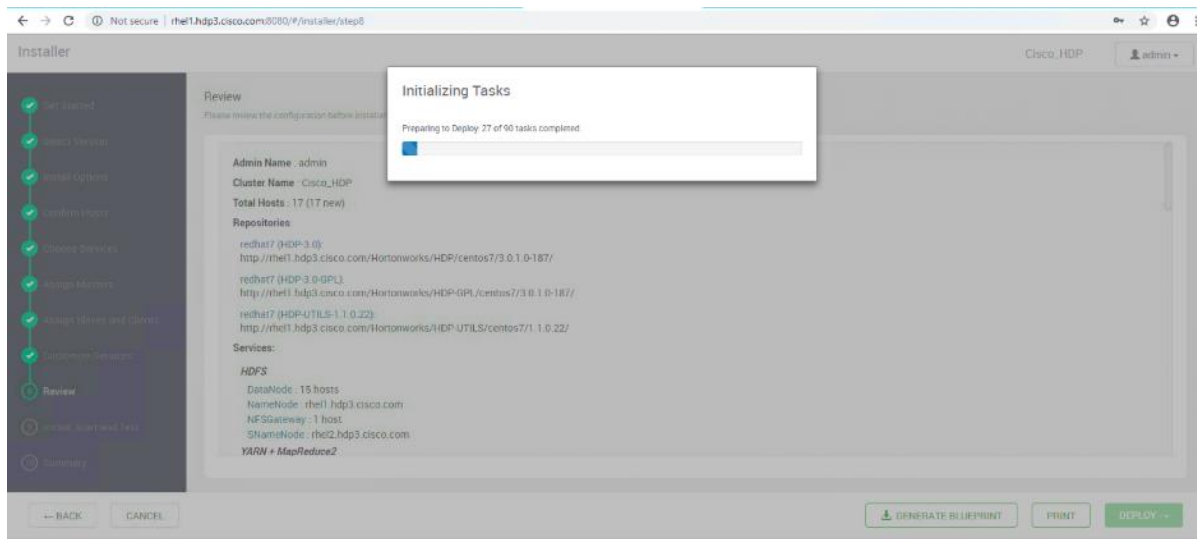
Review

The assignments that have been made are displayed. Check to make sure all is correct before clicking the Deploy button. If any changes are necessary, use the left navigation bar to return to the appropriate screen.

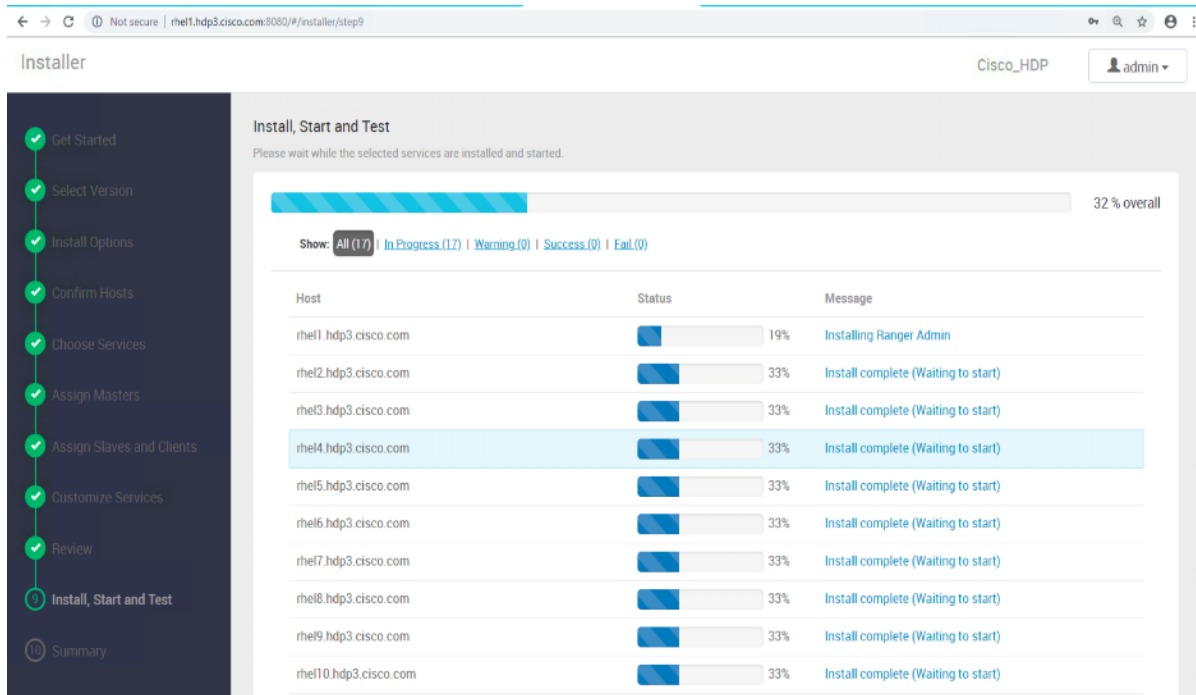


Deploy

1. When the review is complete, click DEPLOY.

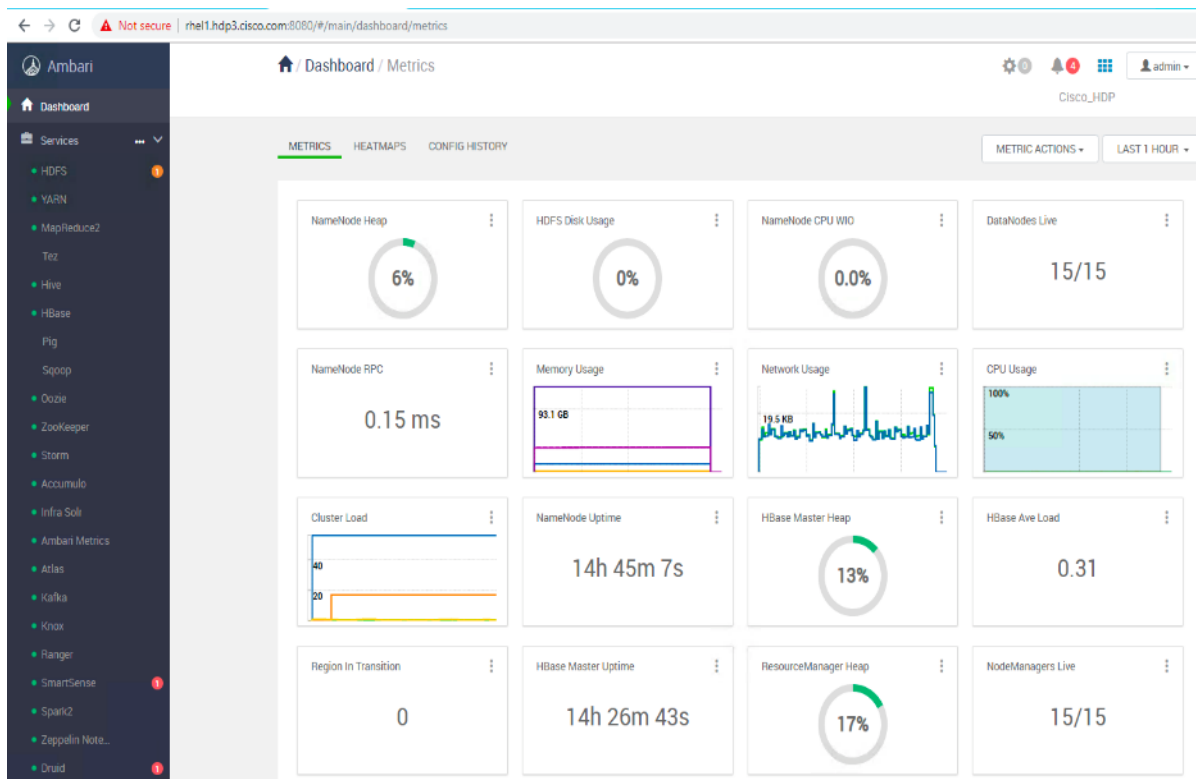


2. Follow the onscreen installation process. Watch for warnings and failures by clicking the link as shown below:



Summary of the Installation Process

1. On the Summary page click “COMPLETE.”



High Availability for HDFS NameNode and YARN ResourceManager

High availability for NameNode and YARN Resource Manager can be configured using Ambari or also on non-Ambari clusters. This deployment guide covers the configuration of high availability using Ambari – Use the Ambari wizard interface to configure HA of the components.

The Ambari web UI provides a wizard-driven user experience that allows to configure high availability of the components in many Hortonworks Data Platform (HDP) stack services. The high availability of the components is achieved by setting up primary and secondary components. If the primary component fails or becomes unresponsive, services failover to secondary component. After configuring the high availability for a service, Ambari enables you to manage and disable (roll back) the high availability of components within that service.

Configure the HDFS NameNode High Availability

The HDFS NameNode high availability feature enables you to run redundant NameNodes in the same cluster in an Active/Passive configuration with a hot standby. This eliminates the NameNode as a potential single point of failure (SPOF) in an HDFS cluster. With the release of Hadoop 3.0, you can configure more than one backup NameNode.

Prior to the release of Hadoop 2.0, the NameNode represented a single point of failure (SPOF) in an HDFS cluster. Each cluster had a single NameNode, and if that machine or process became unavailable, the cluster would be unavailable until the NameNode was either restarted or brought up on a separate machine. This situation impacted the total availability of the HDFS cluster in two major ways:

- In the case of an unplanned event such as a machine crash, the cluster would be unavailable until an operator restarted the NameNode.
- Planned maintenance events such as software or hardware upgrades on the NameNode machine would result in periods of cluster downtime.

HDFS NameNode HA avoids this by facilitating either a fast failover to one or more backup NameNodes during machine crash, or a graceful administrator-initiated failover during planned maintenance.



Secondary NameNode is not required in high availability configuration because the Standby node also performs the tasks of the Secondary NameNode.

Active NameNode honors all the client requests and the Standby NameNode acts as a backup. The Standby NameNode keeps its state synchronized with Active NameNode through a group of JournalNodes (JNs). When the Active node performs any namespace modification, the Active node durably logs a modification record to a majority of these JNs. The Standby node reads the edits from the JNs and continuously watches the JNs for changes to the edit log.

Prerequisites for NameNode High Availability

The following are the prerequisites for NameNode high availability:

- NameNode Machine: Hardware for Active and Standby node should be identical.
- JournalNodes Machine: JournalNode daemon is relatively lightweight, therefore it can be co-located on machines with other Hadoop daemons; it is typically located on the management nodes.
- There MUST be at least three JournalNodes, because the edit log modifications must be written to a majority of JNs. This allows the system tolerate failure of a single machine. You may also run more than three JournalNodes, but in order to increase the number of failures that the system can tolerate, you must run an odd number of JNs (3, 5, 7, and so on).

- ZooKeeper Machines: For automatic failover capability, an existing Zookeeper cluster must exist. The Zookeeper service can also co-exist with other Hadoop daemons.
- In HA Cluster, the Standby NameNode also performs the checkpoints of the namespace state, therefore do not deploy a Secondary NameNode, CheckpointNode, or BackupNode in a high availability cluster.

Deploy the NameNode High Availability Cluster

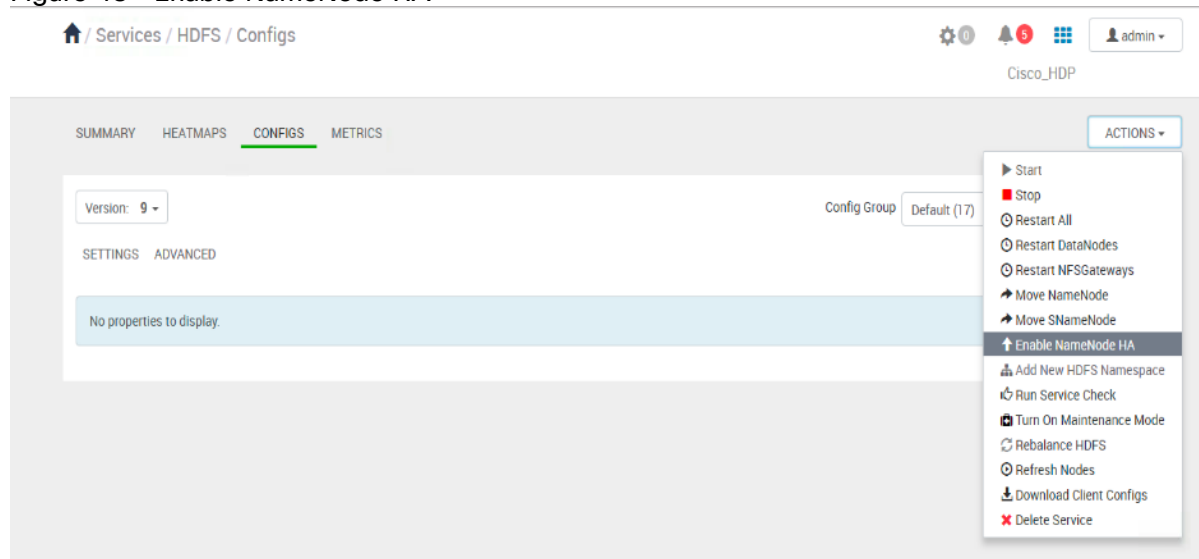
To deploy the NameNode high availability on an Ambari managed cluster, follow these steps:



High availability cannot accept HDFS cluster names that include underscore (_).

1. Log into Ambari. Click HDFS > CONFIGS. Click the ACTIONS drop-down list and click Enable NameNode HA to launch the wizard.

Figure 43 Enable NameNode HA



2. Step 1 launches the Enable NameNode HA wizard. On the Get Started page, specify the Nameservice ID as shown below. Click Next.

Figure 44 Enable NameNode HA Wizard – Get Started

Enable NameNode HA Wizard X

Get Started

This wizard will walk you through enabling NameNode HA on your cluster. Once enabled, you will be running a Standby NameNode in addition to your Active NameNode. This allows for an Active-Standby NameNode configuration that automatically performs failover. The process to enable HA involves a combination of **automated steps** (that will be handled by the wizard) and **manual steps** (that you must perform in sequence as instructed by the wizard). **You should plan a cluster maintenance window and prepare for cluster downtime when enabling NameNode HA.**

If you have HBase running, please exit this wizard and stop HBase first.

Nameservice ID:

NEXT →

3. On the Select Hosts page, select the Additional NameNode and JournalNode. Click Next.

Figure 45 Enable NameNode HA Wizard – Select Hosts

Enable NameNode HA Wizard X

Select Hosts

Select a host that will be running the additional NameNode.
In addition, select the hosts to run JournalNodes, which store NameNode edit logs in a fault tolerant manner.

Current NameNode:

Additional NameNode:

JournalNode:

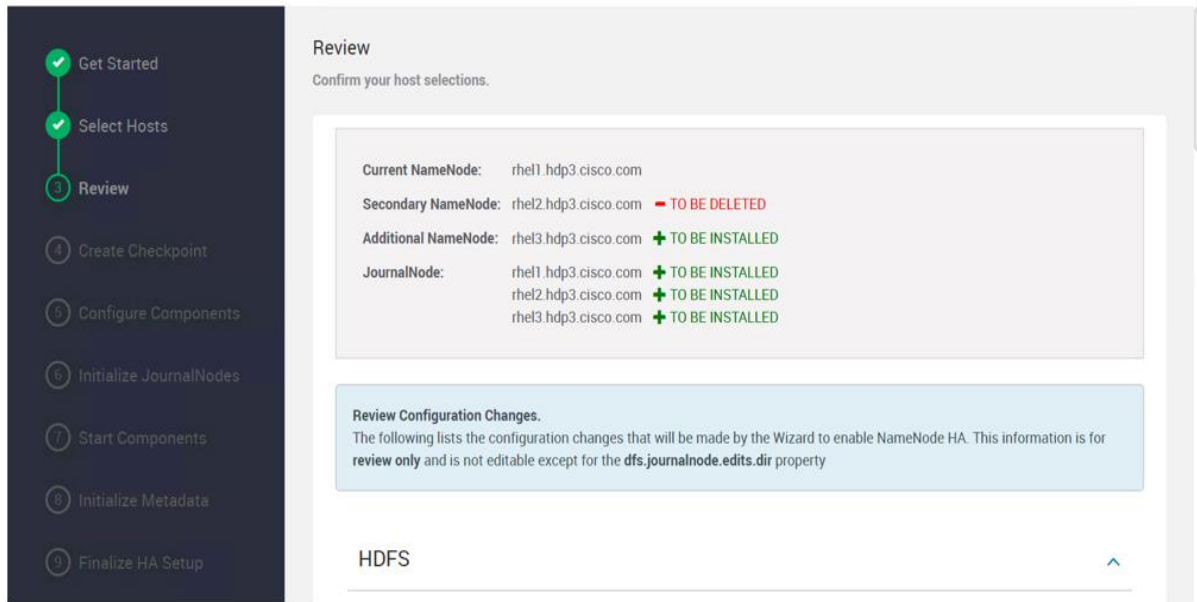
JournalNode:

JournalNode:

+

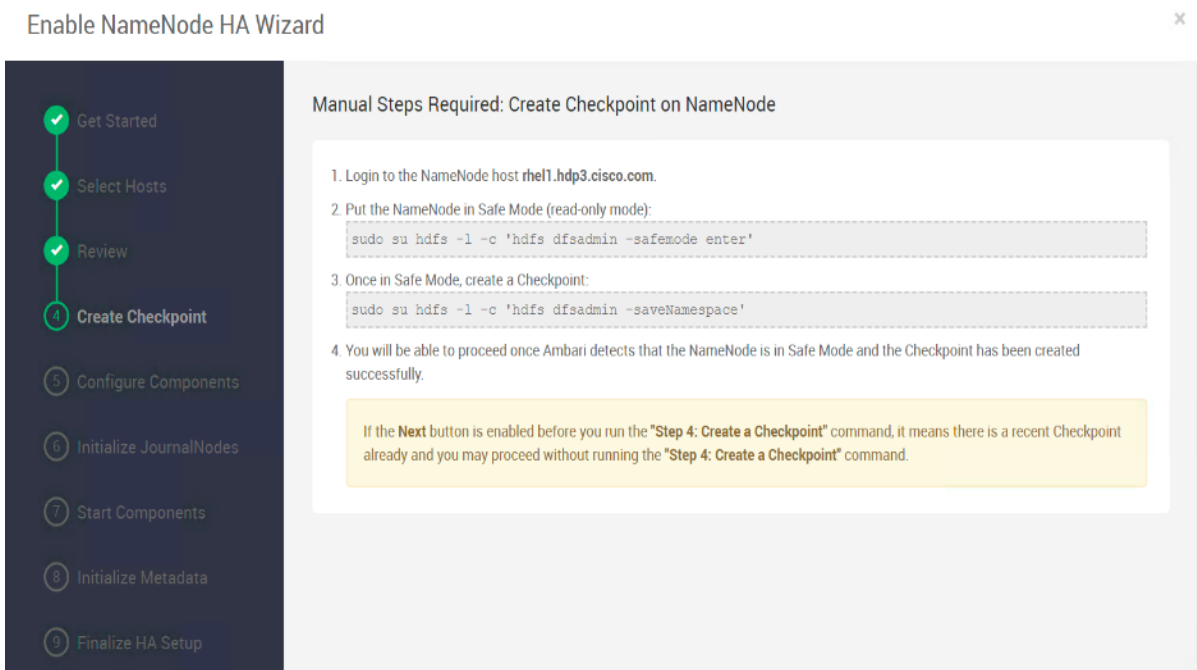
4. On the Review page, confirm the selection. To change any values, click Back, or to continue click Next.

Figure 46 Enable NameNode HA Wizard – Review



5. Create a checkpoint on the NameNode on the linux server (rhel1.hdp3.cisco.com) as shown below:

Figure 47 Enable NameNode HA Wizard – Create Checkpoint



6. SSH to current NameNode, rhel1.hdp3.cisco.com and run the following commands:

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



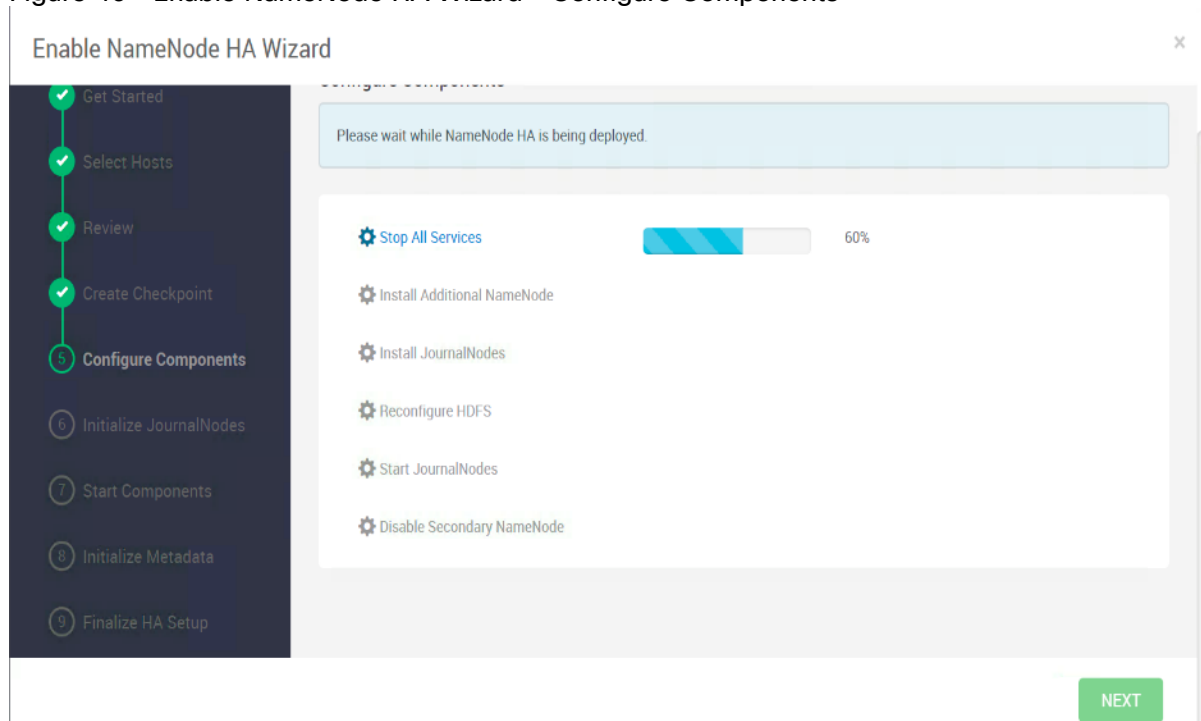
```
[root@rhell ~]# sudo su hdfs -l -c 'hdfs dfsadmin -saveNamespace'
Save namespace successful
[root@rhell ~]#
```

Figure 48 Current NameNode – Safe Mode and Create Checkpoint Command

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

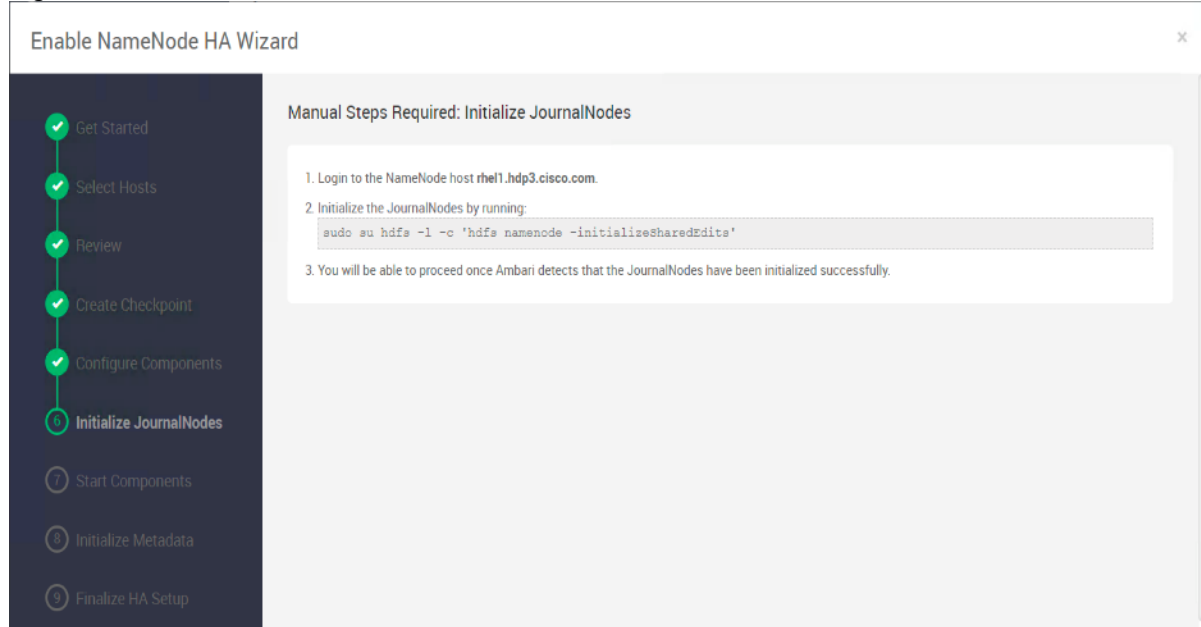
7. Return to the Ambari web UI, verify that the Checkpoint was created. Click Next.
8. See the progress bar on the Configure Components page. When the configuration steps are completed, click Next.

Figure 49 Enable NameNode HA Wizard – Configure Components



9. Initialize the JournalNodes as shown below:

Figure 50 Enable NameNode HA Wizard – Initialize JournalNodes



10. SSH to the current NameNode, for example rhel1.hdp3.cisco.com.

11. Run the following command:

```
[root@rhell ~]# sudo su hdfs -l -c 'hdfs namenode -initializeSharedEdits'
```

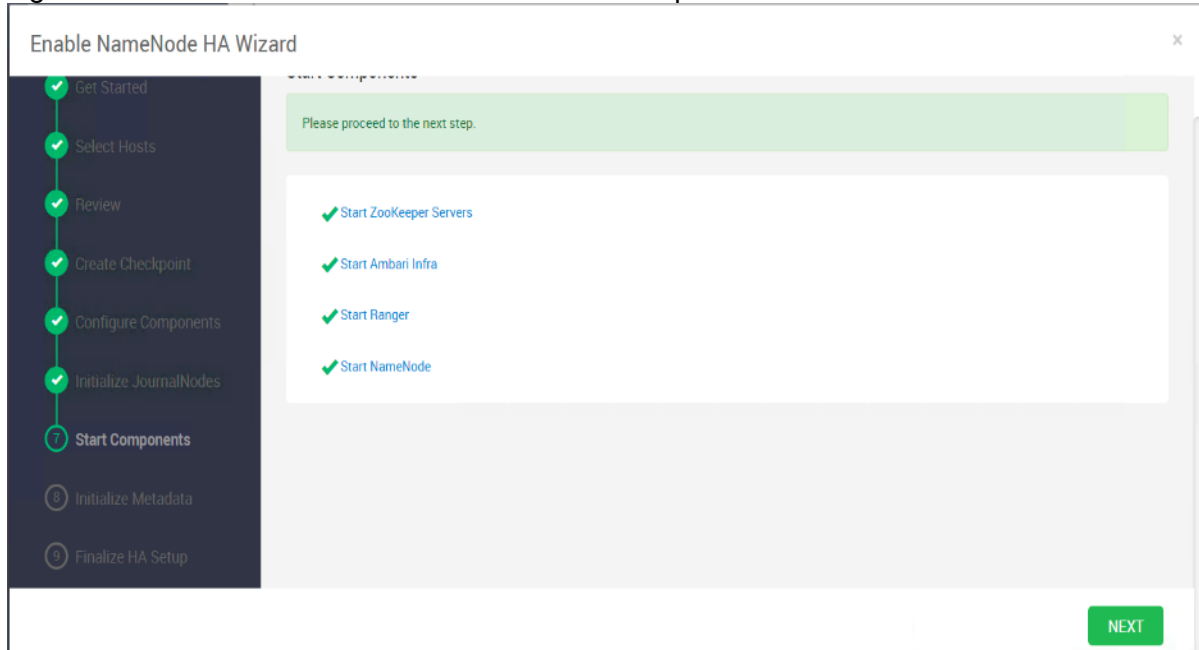
Figure 51 Initialize JournalNodes

```
[root@rhell ~]#
[root@rhell ~]# sudo su hdfs -l -c 'hdfs namenode -initializeSharedEdits'
19/01/15 12:43:27 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = rhell/10.16.1.31
STARTUP_MSG: args = [-initializeSharedEdits]
STARTUP_MSG: version = 3.1.1.3.0.1.0-187
STARTUP_MSG: classpath = /usr/hdp/3.0.1.0-187/hadoop/conf:/usr/hdp/3.0.1.0-187/hadoop/lib/nimbus-jose-jwt-4.41.1.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/ranger-hdfs-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/jersey-server-1.19.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/ranger-er-plugin-classloader-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/jersey-servlet-1.19.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/ranger-yarn-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/jetty-util-9.3.19.v20170502.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/accessors-smart-1.2.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/protobuf-java-2.5.0.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/asm-5.0.4.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/
```

12. Return to the Ambari UI, when Ambari detects success, click Next.

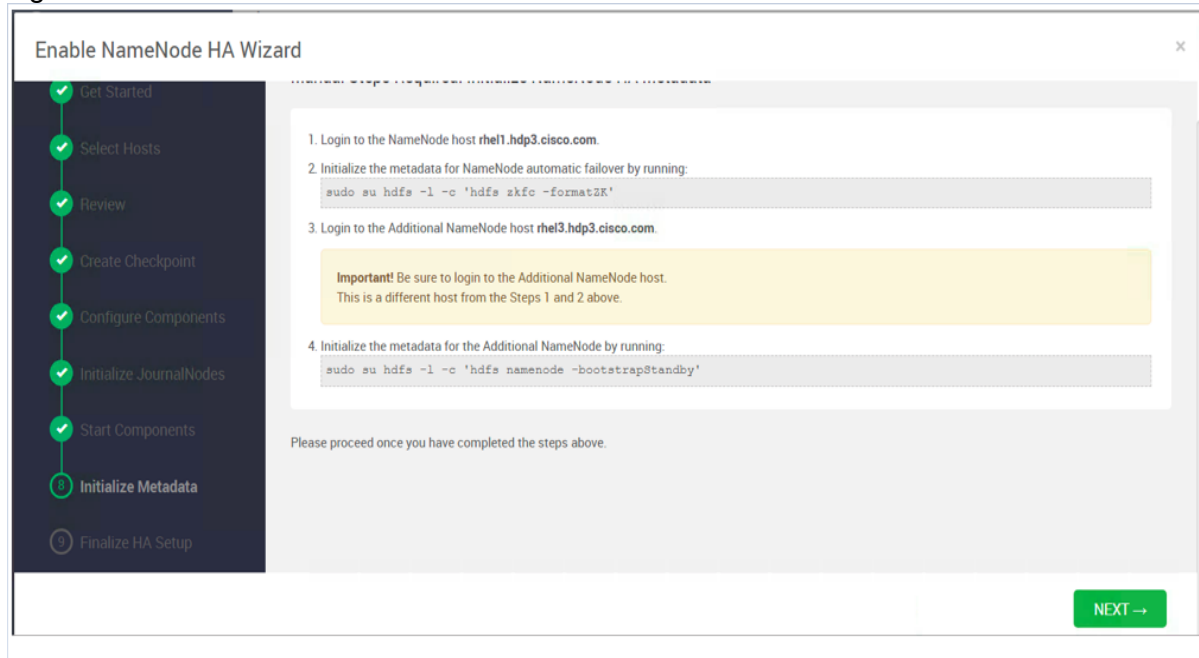
13. On the Start Components page, when completed, click Next.

Figure 52 Enable NameNode HA Wizard – Start Components



14. On the Initialize Metadata page, add the information as shown below:

Figure 53 Enable NameNode HA Wizard – Initialize Metadata



15. SSH to rhel1.hdp3.cisco.com and run the following command:

```
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs zkfc -formatZK'
```

Figure 54 Initialize the Metadata for NameNode

```
[root@rhel1 ~]#
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs zkfc -formatZK'
19/01/15 12:49:24 INFO tools.DFSZKFailoverController: STARTUP_MSG:
/*****
STARTUP_MSG: Starting DFSZKFailoverController
STARTUP_MSG: host = rhel1/10.16.1.31
STARTUP_MSG: args = [-formatZK]
STARTUP_MSG: version = 3.1.1.3.0.1.0-187
STARTUP_MSG: classpath = /usr/hdp/3.0.1.0-187/hadoop/conf:/usr/hdp/3.0.1.0-187/hadoop/lib/nimbus-jose-jwt-4.41.1.jar:/usr
/hdp/3.0.1.0-187/hadoop/lib/ranger-hdfs-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/jersey-server-1.1
9.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/ranger-plugin-classloader-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/jerse
y-servlet-1.19.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/ranger-yarn-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoo
p/lib/jetty-util-9.3.19.v20170502.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/accessors-smart-1.2.jar:/usr/hdp/3.0.1.0-187/hadoop/1
```

16. SSH to an additional NameNode, for example, rhel3.hdp3.cisco.com and run the following command:

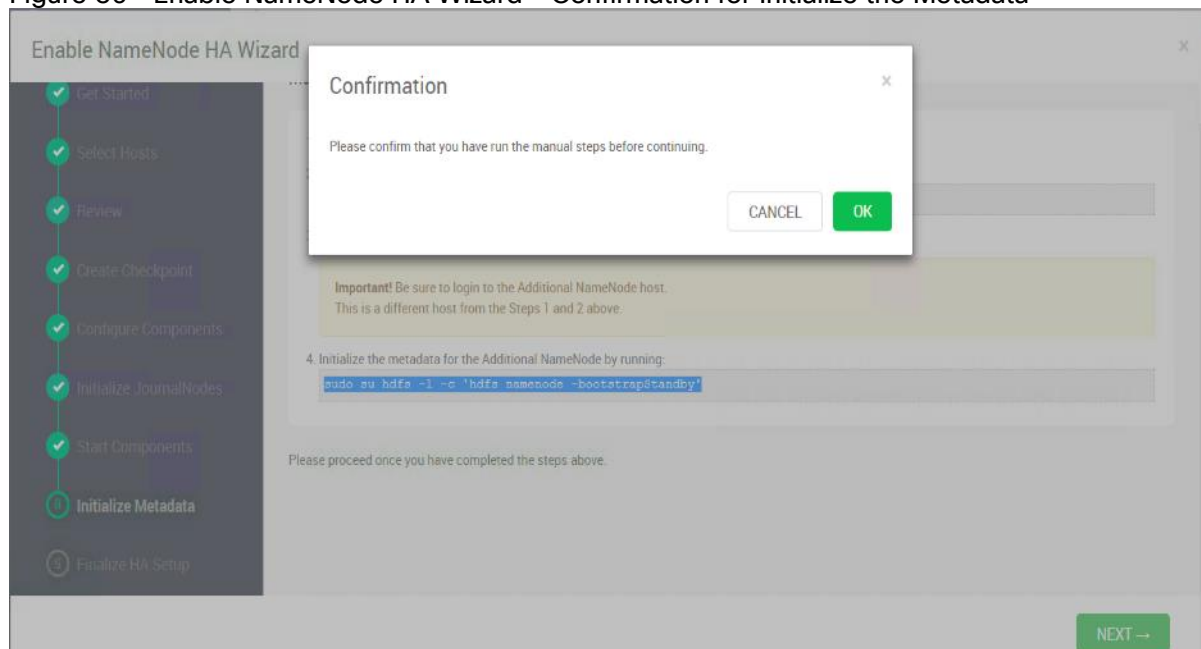
```
[root@rhel3 ~]# sudo su hdfs -l -c 'hdfs namenode -bootstrapStandby'
```

Figure 55 Initialize the Metadata for Additional NameNode

```
[root@rhel13 ~]#
[root@rhel13 ~]#
[root@rhel13 ~]# sudo su hdfs -l -c 'hdfs namenode -bootstrapStandby'
19/01/17 14:34:59 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = rhel13/10.16.1.33
STARTUP_MSG: args = [-bootstrapStandby]
STARTUP_MSG: version = 3.1.1.3.0.1.0-187
STARTUP_MSG: classpath = /usr/hdp/3.0.1.0-187/hadoop/conf:/usr/hdp/3.0.1.0-187/
1.0-187/hadoop/lib/ranger-hdfs-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/
1.0-187/hadoop/lib/ranger-plugin-classloader-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.
/3.0.1.0-187/hadoop/lib/ranger-yarn-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.
r:/usr/hdp/3.0.1.0-187/hadoop/lib/accessors-smart-1.2.jar:/usr/hdp/3.0.1.0-187/h
87/hadoop/lib/asm-5.0.4.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/re2j-1.1.jar:/usr/hdp
1.0-187/hadoop/lib/jul-to-slf4j-1.7.25.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/commo
```

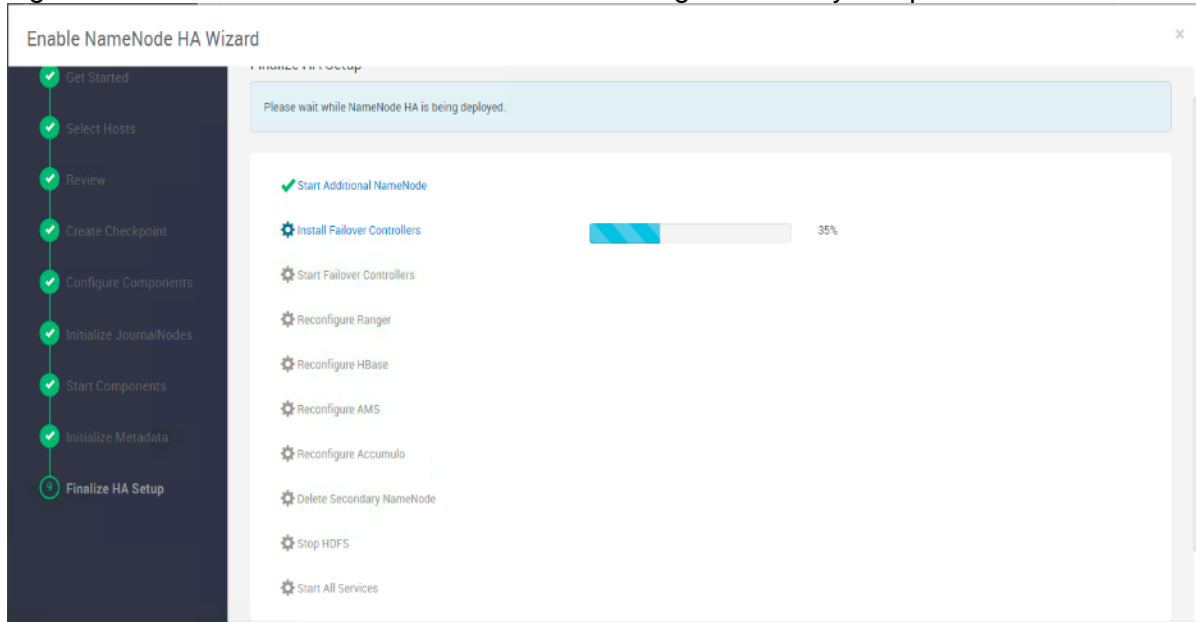
17. Return to the Ambari web UI, click NEXT. Click OK on the confirmation pop-up window. Make sure the initialization of metadata was performed in NameNode and an additional NameNode as mentioned in step 15 and 16.

Figure 56 Enable NameNode HA Wizard - Confirmation for Initialize the Metadata



18. On the Finalize HA Setup page, you can see the progress bar as the high availability completes.

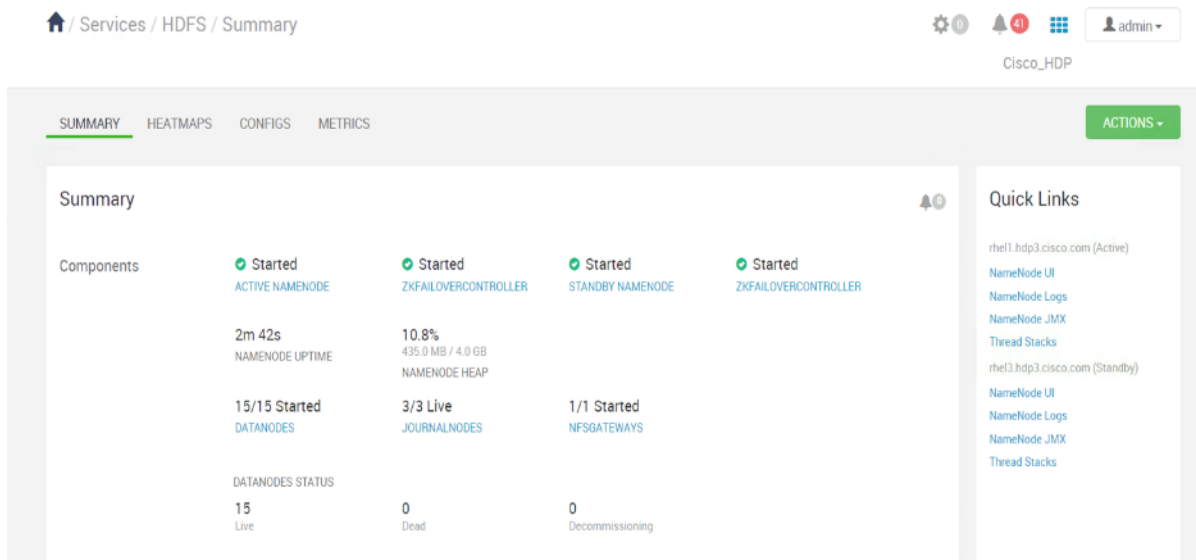
Figure 57 Enable NameNode HA Wizard - Finalize High Availability Setup



19. Click COMPLETE when done.

20. Click HDFS > SUMMARY tab, verify the Active and Standby NameNode. The Quick Links pane also shows that rhel1.hdp3.cisco.com is running the Active NameNode and rhel3.hdp3.cisco.com is running in Standby NameNode.

Figure 58 Ambari - HDFS - Summary Information



Configure the YARN ResourceManger HA

This section provides instructions on setting up the ResourceManager (RM) HA feature in a HDFS cluster. The Active and Standby ResourceManagers embed the ZooKeeper based ActiveStandbyElector to determine which RM should be active.

Prerequisites for ResourceManager HA

The following are the prerequisites for ResourceManager HA:

- The servers where Active and Standby RMs are run should have identical hardware.
- For automated failover configurations, there must be an existing Zookeeper cluster available. The ZooKeeper service nodes can be co-located with the other Hadoop daemons.



At least three ZooKeeper servers must be running.

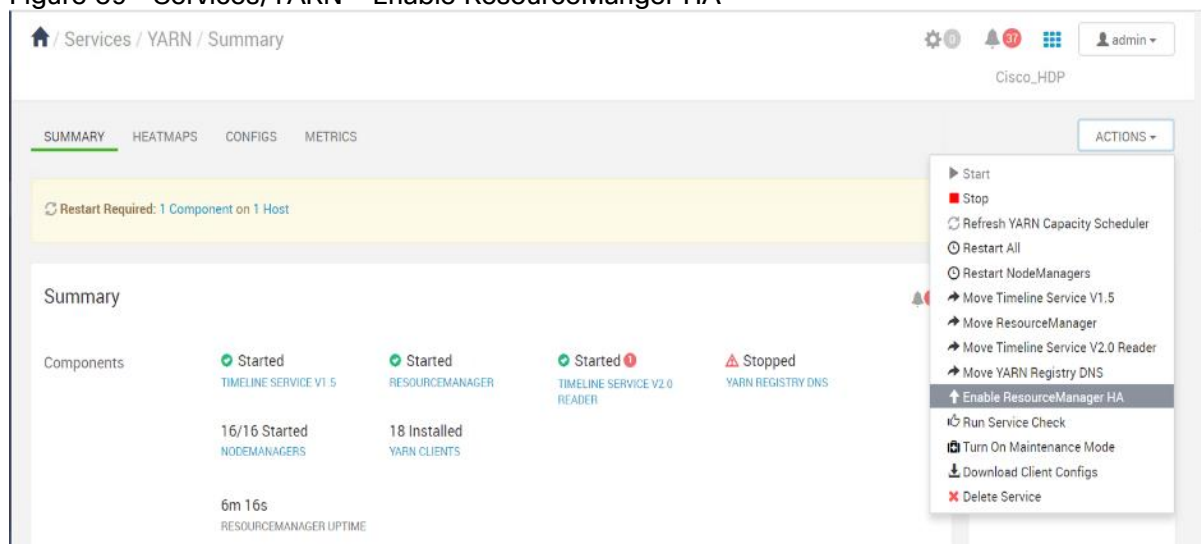
Deploy the ResourceManager HA

ResourceManager HA can be configured manually or through Ambari. These instructions are based on configuring ResourceManager HA using Ambari.

To setup ResourceManager HA, follow these steps:

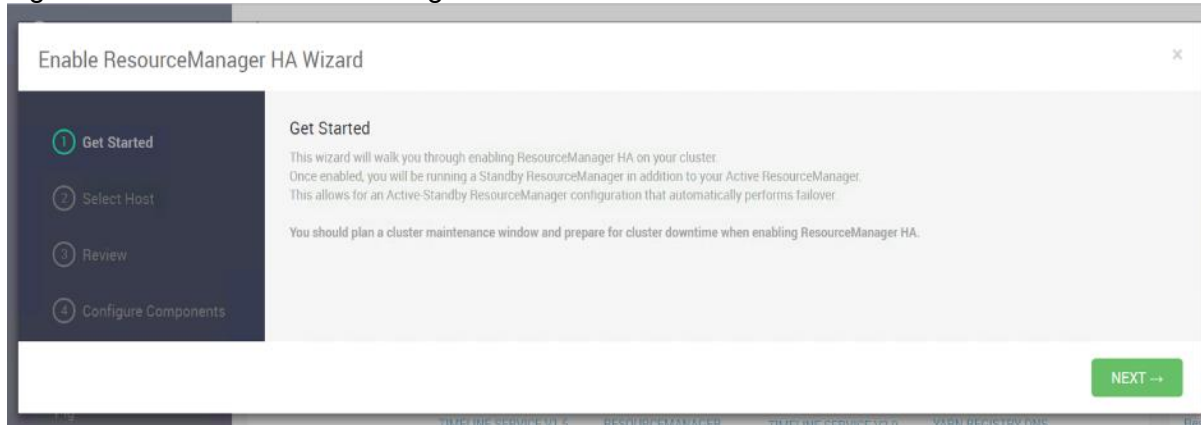
1. From the Ambari web UI, click Services > YARN. Click the ACTIONS drop-down list and select Enable ResourceManager HA.

Figure 59 Services/YARN - Enable ResourceManger HA



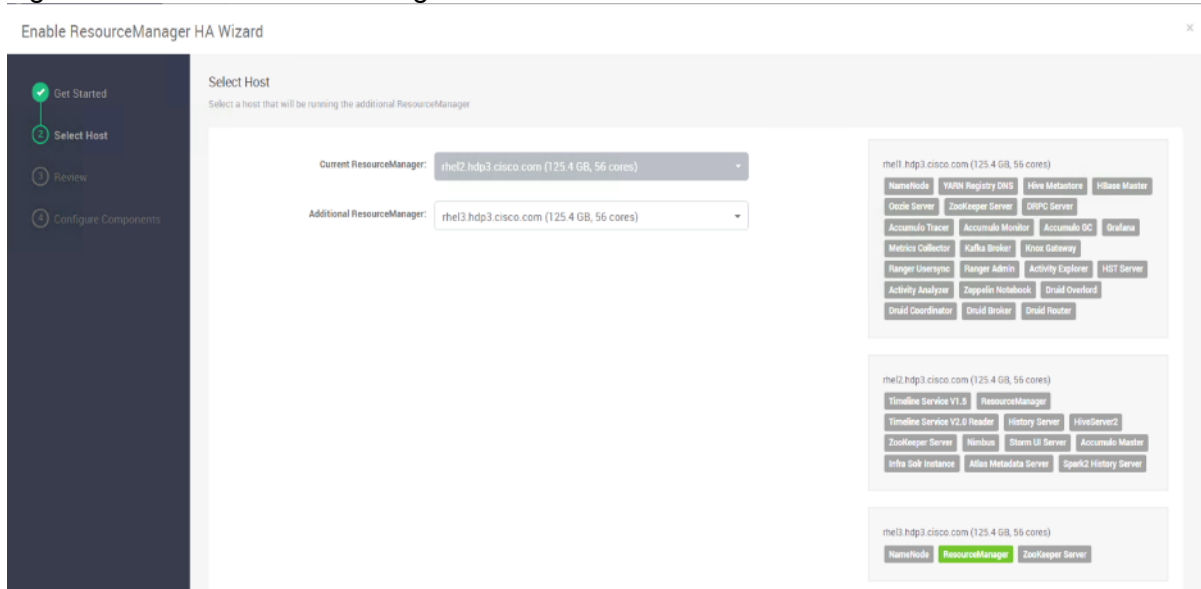
2. This launches the ResourceManager HA wizard as shown below. Click NEXT.

Figure 60 Enable ResourceManager HA Wizard – Get Started



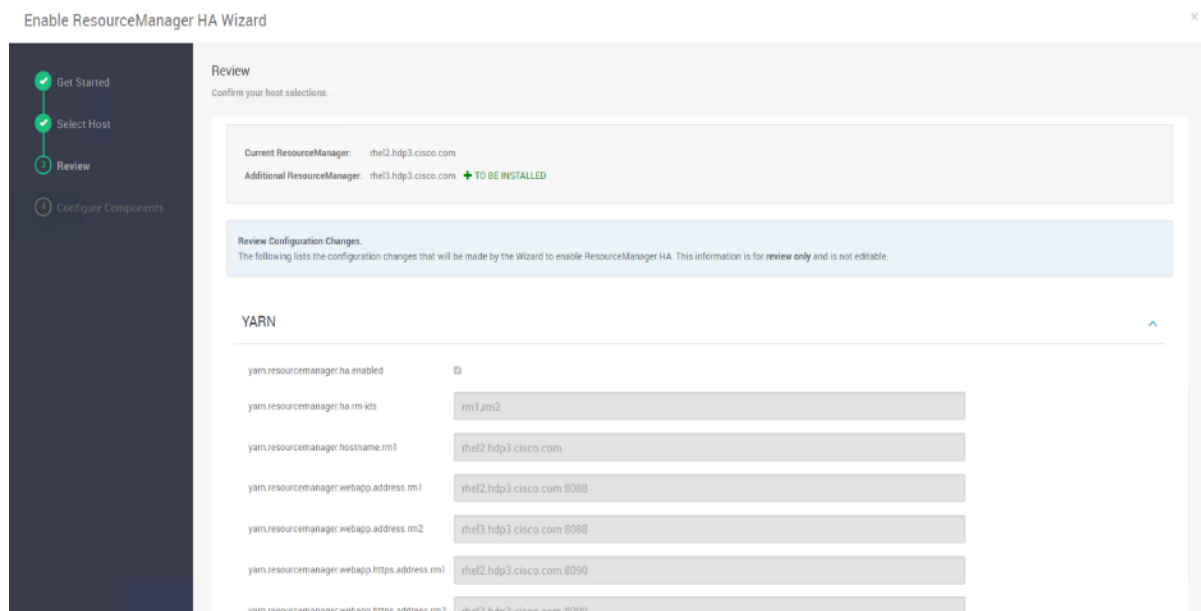
3. From the Select Host page, select the host for Additional Resource Manager. Click NEXT.

Figure 61 Enable ResourceManger HA Wizard – Select Host



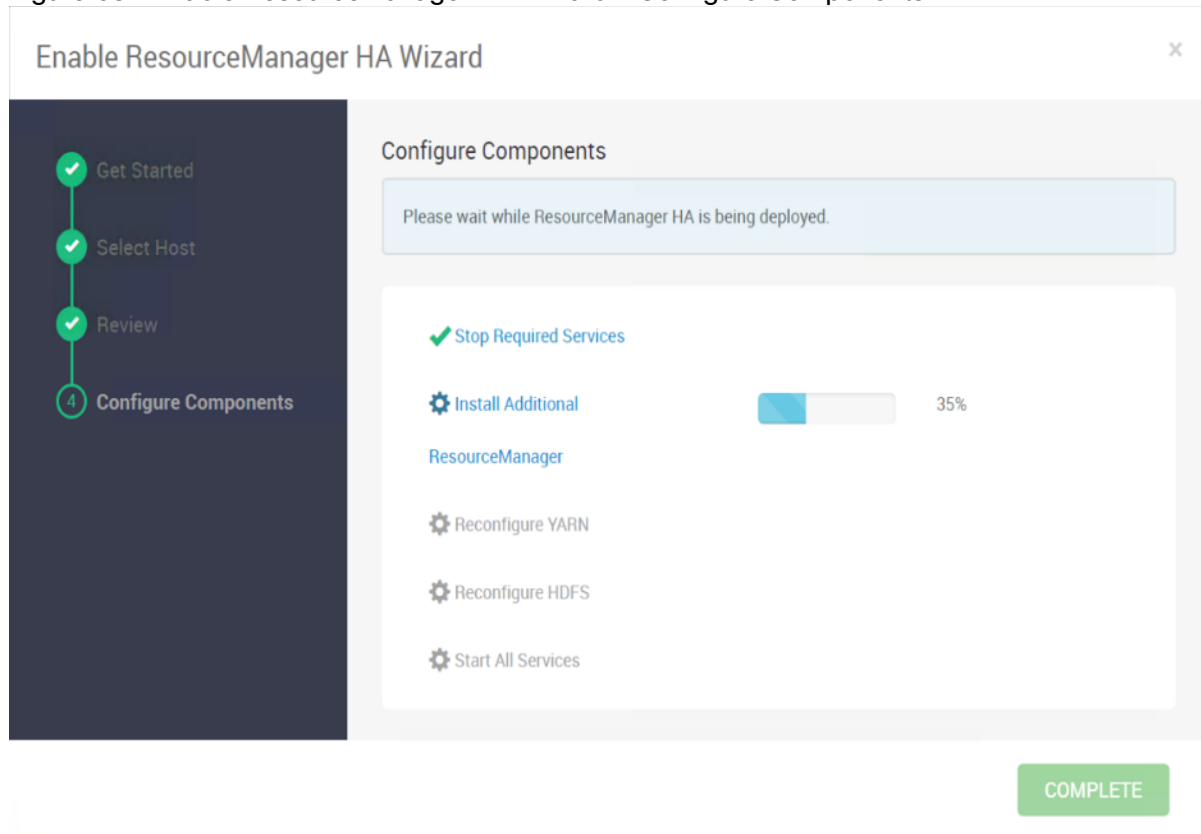
4. Proceed to the Review page.

Figure 62 Enable ResourceManager HA Wizard - Review



- The Configure Components page shows the progress bar as the Additional ResourceManager is being deployed.

Figure 63 Enable ResourceManager HA Wizard - Configure Components



- Click COMPLETE when done.



It was observed that in certain circumstances, services might fail to restart. Click COMPLETE and re-start the services in Ambari dashboard.

- Verify the ResourceManger HA setup by clicking Services > YARN > SUMMARY tab. The Quick Links pane identifies Active and Standby ResourceManager.

Figure 64 Services/YARN/Summary – Verify ResourceManger HA

The screenshot shows the Ambari dashboard for Services > YARN > Summary. The page has a navigation bar with tabs for SUMMARY, HEATMAPS, CONFIGS, and METRICS. The SUMMARY tab is active. The main content area is titled 'Summary' and contains several status cards:

- Timeline Service V1.5:** Started (green checkmark)
- Active ResourceManager:** Started (green checkmark)
- Standby ResourceManager:** Started (green checkmark)
- Timeline Service V2.0 Reader:** Started (green checkmark with a red exclamation mark)
- YARN Registry DNS:** Started (green checkmark)
- NodeManagers:** 16/16 Started
- YARN Clients:** 18 Installed
- ResourceManager Uptime:** 13m
- NodeManagers Status:** 16 Active, 0 Lost, 0 Unhealthy, 0 Rebooted

On the right side, there is a 'Quick Links' pane with a notification bell icon. It lists links for two nodes: 'rhe2.hdp3.cisco.com (Active)' and 'rhe3.hdp3.cisco.com (Standby)'. For each node, the links include ResourceManager UI, ResourceManager logs, ResourceManager JMX, and Thread Stacks. An 'ACTIONS' button is visible in the top right corner of the dashboard.



Configure NVMe as YARN Local Directory

To configure YARN local directory on NVMe disks, follow these steps:

- Click on cluster > YARN > Configuration tab, filter properties for dirs.
- Modify disk labels specific to NVMe as per the format and partition performed earlier for following properties:
 - yarn.nodemanager.local-dir
 - yarn.nodemanager.log-dirs



YARN NodeManager Local directories

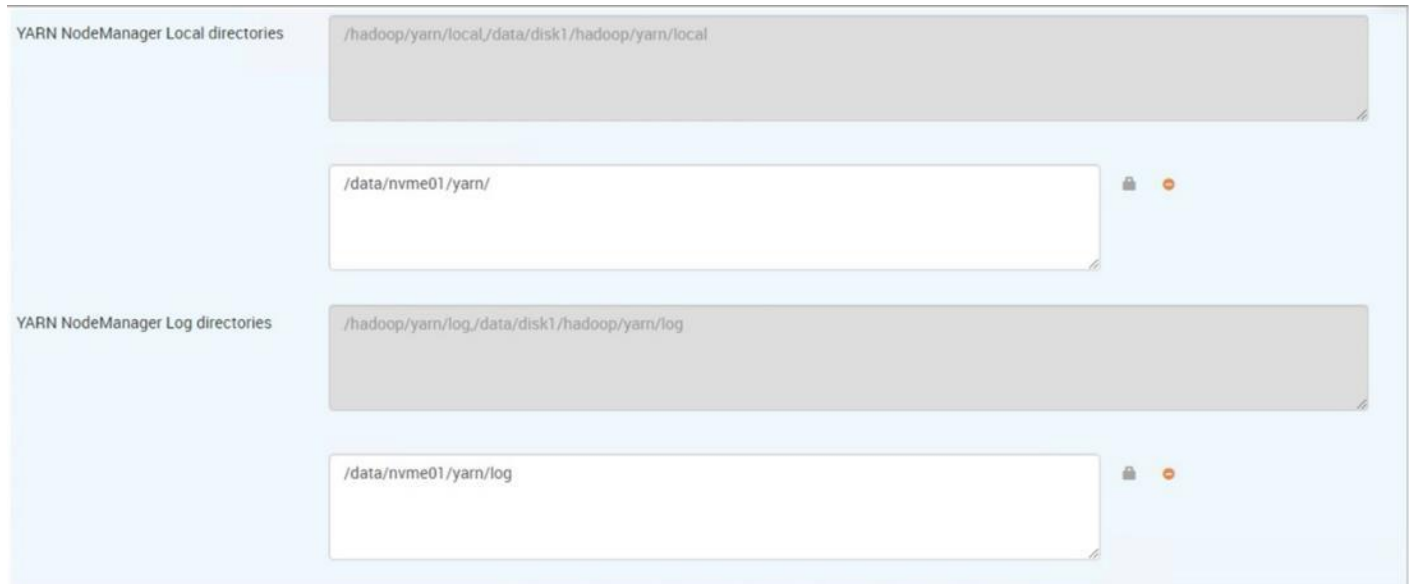
/hadoop/yarn/local/data/disk1/hadoop/yarn/local

/data/nvme01/yarn/  

YARN NodeManager Log directories

/hadoop/yarn/log/data/disk1/hadoop/yarn/log

/data/nvme01/yarn/log  

The image shows a configuration interface for YARN NodeManager. It is divided into two main sections: 'YARN NodeManager Local directories' and 'YARN NodeManager Log directories'. Each section contains a list of directory paths. The first section has two entries: a greyed-out path '/hadoop/yarn/local/data/disk1/hadoop/yarn/local' and an active path '/data/nvme01/yarn/' with a lock icon and a status indicator. The second section also has two entries: a greyed-out path '/hadoop/yarn/log/data/disk1/hadoop/yarn/log' and an active path '/data/nvme01/yarn/log' with a lock icon and a status indicator.

Summary

When building an infrastructure to enable this modernized architecture which could scale to thousands of nodes, operational efficiency can't be an afterthought.

To bring in seamless operation of the application at this scale, you need:

- Infrastructure automation of Cisco UCS servers with service profiles and Cisco Data Center network automation with application profiles with Cisco ACI
- Centralized Management and Deep telemetry and Simplified granular trouble-shooting capabilities and Multi-tenancy allowing application workloads including containers, micro-services, with the right level of security and SLA for each workload.
- Cisco UCS with Cisco Intersight and Cisco ACI can enable this cloud scale architecture deployed and managed with ease

For More Information

For additional information, see the following resources:

- To find out more about Cisco UCS big data solutions, see <http://www.cisco.com/go/bigdata>.
- To find out more about Cisco UCS big data validated designs, see http://www.cisco.com/go/bigdata_design
- To find out more about Cisco UCS AI/ML solutions, see <http://www.cisco.com/go/ai-compute>
- To find out more about Cisco ACI solutions, see <http://www.cisco.com/go/aci>
- To find out more about Cisco validated solutions based on Software Defined Storage, see <https://www.cisco.com/c/en/us/solutions/data-center-virtualization/software-defined-storage-solutions/index.html>

Bill of Materials

This section provides the BOM for the 16 Nodes Hadoop Base Rack. See Table 10 for the BOM for the Hadoop Base rack, Table 11 for the BOM for Expansion Rack, Table 12 for software components. Table 13 lists Hortonworks SKUs available from Cisco.

Table 10 Bill of Materials for Cisco UCS C240M5SX Hadoop Nodes Base Rack

Part Number	Description	Qty
UCSC-C240-M5SX	Cisco UCS C240 M5 2U rack server w/o CPU, Memory, drives, PCIe cards or power supply	16
CON-OSP-C240M5A2	SNTC 24X7X40S UCS C240 M5 A2	16
UCS-CPU-I6230	2.1 GHz 6230/125W 20C/27.5MB Cache/DDR4 2933MHz	32
UCS-MR-X32G2RT-H	32GB DDR4-2933-MHz RDIMM/2Rx4/1.2v	192
UCSC-PCI-1-C240M5	Riser 1 including 3 PCIe slots (x8, x16, x8); slot 3 required CPU2	16
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	16
UCSC-PSU1-1600W	Cisco UCS 1600W AC Power Supply for Rack Server	32
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	32
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 & C240 M4 & M5 rack servers	16
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	16
UCSC-HS-C240M5	Heat sink for UCS C240 M5 rack servers 150W CPUs & below	32
UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	416
UCSC-PCI-2B-240M5	Cisco Riser 2 option 2B with 3 PCIe slots (x8, x16, x8) plus 1 NVMe connector (controls two rear SFF NVMe drives) and supports a GPU.	16
UCSC-RNVME-240M5=	The Cisco C240 M5 Rear NVMe Cable Kit consists of a rear NVMe cable and backplane	16
CBL-SC-MR12GM5P	Super Cap cable for UCSC-RAID-M5HD	16
UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	16
UCSC-RAID-M5HD	Cisco 12G Modular RAID controller with 4GB cache	16
UCS-SP-FI6332-2X	UCS SP Select 2 x 6332 FI	1
UCS-SP-FI6332	(Not sold standalone) UCS 6332 1RU FI/12 QSFP+	2
CON-OSP-SPFI6332	ONSITE 24X7X4 (Not sold standalone) UCS 6332 1RU FI/No PSU/3	2

Part Number	Description	Qty
UCS-PSU-6332-AC	UCS 6332 Power Supply/100-240VAC	4
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	4
QSFP-H40G-CU3M	40GBASE-CR4 Passive Copper Cable, 3m	16
QSFP-40G-SR-BD	QSFP40G BiDi Short-reach Transceiver	8
N10-MGT015	UCS Manager v3.2(1)	2
UCS-ACC-6332	UCS 6332 Chassis Accessory Kit	2
UCS-FAN-6332	UCS 6332 Fan Module	8
QSFP-H40G-CU3M=	40GBASE-CR4 Passive Copper Cable, 3m	32
UCS-HD24TB10K4KN	2.4 TB 12G SAS 10K RPM SFF HDD (4K)	384
UCSC-NVMEHW-I8000	Cisco 2.5" U.2 8TB Intel P4500 NVMe High Perf. Value Endurance	32
UCS-M2-240GB	240 GB M.2 SATA SSD	32
UCS-M2-HWRAID	Cisco Boot optimized M.2 Raid controller	16

Table 11 Bill of Materials for Hadoop Nodes Expansion Rack

Part Number	Description	Qty
UCSC-C240-M5SX	Cisco UCS C240 M5 2U rack server w/o CPU, Memory, drives, PCIe cards or power supply	12
CON-OSP-C240M5A2	SNTC 24X7X4OS UCS C240 M5 A2	12
UCS-CPU-6230	2.1 GHz 6230/125W 20C/27.5MB Cache/DDR4 2933MHz	24
UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	144
UCSC-PCI-1-C240M5	Riser 1 including 3 PCIe slots (x8, x16, x8); slot 3 required CPU2	12
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	12
UCSC-PSU1-1600W	Cisco UCS 1600W AC Power Supply for Rack Server	24
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	24
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 & C240 M4 & M5 rack servers	12
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	12
UCSC-HS-C240M5	Heat sink for UCS C240 M5 rack servers 150W CPUs and below	24
UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	312

Part Number	Description	Qty
UCSC-PCI-2B-240M5	Cisco Riser 2 option 2B with 3 PCIe slots (x8, x16, x8) plus 1 NVMe connector (controls two rear SFF NVMe drives) and supports a GPU.	12
UCSC-RNVME-240M5=	The Cisco C240 M5 Rear NVMe Cable Kit consists of a rear NVMe cable and backplane	24
CBL-SC-MR12GM5P	Super Cap cable for UCSC-RAID-M5HD	12
UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	12
UCSC-RAID-M5HD	Cisco 12G Modular RAID controller with 4GB cache	12
UCS-HD24TB10K4KN	2.4 TB 12G SAS 10K RPM SFF HDD (4K)	288
UCSC-NVMEHW-I8000	Cisco 2.5" U.2 8TB Intel P4500 NVMe High Perf. Value Endurance	24
UCS-M2-240GB	240 GB M.2 SATA SSD	24
UCS-M2-HWRAID	Cisco Boot optimized M.2 Raid controller	12

Table 12 Red Hat Enterprise Linux License

Part Number	Description	Qty
RHEL-2S2V-3A	Red Hat Enterprise Linux	30
CON-ISV1-EL2S2V3A	3-year Support for Red Hat Enterprise Linux	30

Table 13 Hortonworks SKU's Available at Cisco

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

Cisco TOP SKU	Cisco PID with Duration	Product Name
UCS-BD-CEADN-BZ=	UCS-BD-CEADN-BZ-3Y	Hortonworks Enterprise Analytical Database Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEADN-GD=	UCS-BD-CEADN-GD-3Y	Hortonworks Enterprise Analytical Database Edition, Node License, Gold Support - 3 Year
UCS-BD-CEDHN-BZ=	UCS-BD-CEDHN-BZ-3Y	Hortonworks Enterprise Data Hub Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEDHN-GD=	UCS-BD-CEDHN-GD-3Y	Hortonworks Enterprise Data Hub Edition, Node License, Gold Support - 3 Year

Appendix

Configure Data Drives on Name Node and Data Nodes

This section describes the steps needed to configure non-OS disk drives as RAID1 using the StorCli command. All drives are part of a single RAID1 volume. This volume can be used for staging any client data to be loaded to HDFS. This volume will not be used for HDFS data.

To configure data drives on Name node and other nodes, if the drive state shows up as JBOD, creating RAID in the subsequent steps will fail with the error *“The specified physical disk does not have the appropriate attributes to complete the requested command.”*

To configure data drives on Name and Data nodes, follow these steps:

1. If the drive state shows up as JBOD, it can be converted into Unconfigured Good using Cisco UCSM or storcli64 command. Following steps should be performed if the state is JBOD.
2. Get the enclosure id as follows:

```
ansible all -m shell -a "./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'"
```

```
[root@rhel01 ~]# ansible all -m shell -a "./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'"
rhel06.hdp3.cisco.local | CHANGED | rc=0 >>
 24 Enclosure Device ID: 66
 24 Enclosure position: 0

rhel04.hdp3.cisco.local | CHANGED | rc=0 >>
 24 Enclosure Device ID: 66
 24 Enclosure position: 0

rhel08.hdp3.cisco.local | CHANGED | rc=0 >>
 24 Enclosure Device ID: 66
 24 Enclosure position: 0
```



It is observed that some earlier versions of storcli64 complains about above mentioned command as if it is deprecated. In this case, please use `“./storcli64 /c0 show all| awk '{print $1}'| sed -n '/[0-9]:[0-9]/p|awk '{print substr($1,1,2)}'|sort -u”` command to determine enclosure id.



In case of S3260 use `-a0` and `-a1` or `c0` and `c1` as there are two controller per node.

3. Convert to unconfigured good:

```
ansible datanodes -m command -a "./storcli64 /c0 /e66 /sall set good force"
```

4. Verify status by running the following command:

```
# ansible datanodes -m command -a "./storcli64 /c0 /e66 /sall show"
```

5. Run this script as root user on rhel01 to rhel3 to create the virtual drives for the management nodes:


```
#vi /root/raid1.sh
./storcli64 -cfgldadd
r1[$1:1,$1:2,$1:3,$1:4,$1:5,$1:6,$1:7,$1:8,$1:9,$1:10,$1:11,$1:12,$1:13,$1:14,$1:15,
$1:16,$1:17,$1:18,$1:19,$1:20,$1:21,$1:22,$1:23,$1:24] wb ra nocachedbadbbu
strpsz1024 -a0
```



The script (above) requires enclosure ID as a parameter.

6. Run the following command to get enclosure id:

```
#!/storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c
| awk '{print $2}'
#chmod 755 raid1.sh
```

7. Run MegaCli script:

```
#!/raid1.sh <EnclosureID> obtained by running the command above
WB: Write back
RA: Read Ahead
NoCachedBadBBU: Do not write cache when the BBU is bad.
Strpsz1024: Strip Size of 1024K
```



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

8. Run the following command. State should change to Online:

```
ansible namenodes -m command -a " ./storcli64 /c0 /e66 /sall show"
```

9. State can also be verified in UCSM as show below in Equipment>Rack-Mounts>Servers>Server # under Inventory/Storage/Disk tab:

Name	Size (MB)	Serial	Operability	Disk State	Presence	Technology	Buildable
Storage Controller SAS 1							
Disk 1	1718655	S3Z017N20000000044020	Operable	Online	Equipped	HDD	False
Disk 2	1718655	S3Z020070000000240030	Operable	Online	Equipped	HDD	False
Disk 3	1718655	S3Z010P20000000010010	Operable	Online	Equipped	HDD	False
Disk 4	1718655	S3Z012200000000000000	Operable	Online	Equipped	HDD	False
Disk 5	1718655	S3Z017H00000000000000	Operable	Online	Equipped	HDD	False
Disk 6	1718655	S3Z014S00000000000000	Operable	Online	Equipped	HDD	False
Disk 7	1718655	S3Z010000000000000000	Operable	Online	Equipped	HDD	False

Configure Data Drives on Name Nodes

To configure non-OS disk drives as RAID 1 volume using storcli command, follow this step:

1. Run the following script as root user on RHEL01 to RHEL03 (any NameNode) to create the virtual drive for the management node(s).

```
#vi /root/raid1.sh
```

```
#ansible namenodes -m command -a "./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,$1:25,$1:26] WB RA direct
NoCachedBadBBU strpsz1024 -a0"
```

Configure Data Drives on Data Nodes

To configure non-OS disk drives as individual RAID0 volumes using storcli command, follow this step. These volumes will be used for HDFS Data.

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

```
[root@rhel01 ~]# ansible datanodes -m command -a "./storcli64 -cfgeachdskraid0 WB RA
direct NoCachedBadBBU strpsz1024 -a0"

rhel7.hdp3.cisco.local | SUCCESS | rc=0 >>
Adapter 0: Created VD 0
Configured physical device at Encl-66:Slot-7.
Adapter 0: Created VD 1
Configured physical device at Encl-66:Slot-6.
Adapter 0: Created VD 2
Configured physical device at Encl-66:Slot-8.
Adapter 0: Created VD 3
Configured physical device at Encl-66:Slot-5.
Adapter 0: Created VD 4
Configured physical device at Encl-66:Slot-3.
Adapter 0: Created VD 5
Configured physical device at Encl-66:Slot-4.
Adapter 0: Created VD 6
Configured physical device at Encl-66:Slot-1.
Adapter 0: Created VD 7
Configured physical device at Encl-66:Slot-2.
..... Omitted Ouput
24 physical devices are Configured on adapter 0.

Exit Code: 0x00
```



The command (above) will not override existing configurations. To clear and reconfigure existing configurations, refer to the Embedded MegaRAID Software Users Guide available at www.broadcom.com.

About the Authors

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

Yogesh Ramesh is a Big Data Solutions Architect at Computing Systems Product Group. He is part of the solution engineering team focusing on big data infrastructure, solutions, and performance.

Acknowledgements

For their support and contribution to the design, validation, and creation of this Cisco Validated Design, the author would like to thank:

- Karthik Kulkarni, Architect, Computing Systems Product Group, Cisco Systems, Inc.
- Muhammad Afzal, Architect, Computing Systems Product Group, Cisco Systems, Inc.