



Cisco Data Intelligence Platform on Cisco UCS M6 with Cloudera Data Platform Private Cloud Base

Deployment Guide for Cisco Data Intelligence
Platform with Cloudera Data Platform Private Cloud
Data Base and Spark 3 with NVIDIA RAPIDS for GPU-
powered Data Science

Published: January 2023



In partnership 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>.

Executive Summary

Today, leading enterprises utilize artificial intelligence/machine learning (AI/ML) to discover insights hidden in massive amounts of data through data processing and data engineering. As enterprises are adopting newer AI/ML enabled use cases to support problem solving and progress toward business intelligence goal through revolution of increased computing power, vast amounts of data storage and better algorithms are not enough to drive AI/ML enabled business challenges.

Data scientists are utilizing data sets on a magnitude and scale never seen before, implementing use cases such as transforming supply chain models, responding to increased levels of fraud, predicting customer churn, and developing new product lines. To be successful, data scientists need the tools and underlying processing power to train, evaluate, iterate, and retrain their models to obtain highly accurate results. The sheer size of the data to be processed and analyzed has a direct impact on the cost and speed at which companies can train and operate their AI/ML models with dynamic scalability. Data set size can also heavily influence where to deploy infrastructure—whether in a public, private, or hybrid cloud.

Cloudera Private Cloud enables unified data fabric with broad set of tools and management capability for data analytics and AI/ML use cases along with secure user access and data governance through:

- **Cloudera Data Platform Private Cloud Base (CDP PvC Base)** - provides storage and supports the traditional data lake environments. It also introduced Apache Ozone, the next generation of filesystem for data lake
- **Cloudera Data Platform Private Cloud Data Services (CDP PvC DS)** - provides personas (such as data analyst, data scientist, data engineer) driven data services from private and hybrid data lakes.

[Cisco Data Intelligence Platform](#) (CDIP) is thoughtfully designed private cloud for data lake. It supports data intensive workloads with Cloudera Data Platform Private Cloud Base and compute rich (AI/ML) and compute intensive workloads with Cloudera Data Platform Private Cloud Data Services. CDIP further provides storage consolidation with Apache Ozone on Cisco UCS infrastructure enables an object store implementation to support several new use cases and higher scale, which is fully managed by Cisco Intersight. Cisco Intersight simplifies management and moves management of computing resources from network to the cloud.

This CVD implements CDIP with cloud advantage in mind for private and hybrid cloud. It is based on Cisco UCS M6 family of servers which support 3rd Generation Intel Xeon Scalable family processors with PCIe Gen 4 capabilities. These servers include the following.

- **The Cisco UCS C240 M6 Server for Storage (Apache Ozone and HDFS)** - Extends the capabilities of the Cisco UCS rack server portfolio supporting more than 43 percent more cores per socket and 33 percent more memory when compared with the previous generation.
- **The Cisco UCS X-Series with Cisco Intersight** - A modular system managed from the cloud. It is designed to meet the needs of modern applications and improve operational efficiency, agility, and scale through an adaptable, future-ready, and modular design.

Furthermore, with Cisco Intersight you get all the benefits of SaaS delivery and full life cycle management of network and compute. This empowers you to analyze, update, fix, and automate your environment in ways that were not possible before.

This CVD explains the implementation of Cloudera Data Platform Private Cloud Base (CDP PvC) 7.1.8 with CDS 3.3 powered by Apache Spark and NVIDIA RAPIDS for GPU powered data science at scale.

CDIP with Cloudera Data Platform enables customers to independently scale storage and computing resources as needed while offering an exabyte scale with low total cost of ownership (TCO). It offers future-proof architecture with the latest technologies provided by Cloudera.

Solution Overview

This chapter contains the following:

- [Audience](#)
- [Purpose of this Document](#)
- [What's New in this Release?](#)

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

This solution implements Cloudera Data Platform Private Cloud Base (CDP PvC Base) and Cloudera Data Platform Private Cloud Data Services (CDP PvC DS) on Cisco Data Intelligence Platform (CDIP) architecture, a world-class platform specifically designed for demanding workloads that is both easy to scale and easy to manage, even as the requirements grow to thousands of servers and petabytes of storage.

Today, many companies recognize the immense potential of big data and machine learning technologies. It is also evident that everyday enormous amount of data is being ingested in on-premises or cloud enabled data lakes with very high velocity. It is quite apparent that IT leaders are challenged in finding ways, how to maximize the ROI of their data, extract valuable insights, and make informed business decisions to gain competitive edge. Furthermore, Apps have transformed into whole new thinking of IT. Apps are becoming the “business” from just supporting the business functions. As a result, modernizing apps, adopting cloud-native architectures, creating micro-services, and utilizing advanced analytics using AI/ML frameworks are becoming de-facto standards for digital transformation. Amid those challenges, siloed monolithic apps and data are further slowing down the pace of innovation and limiting their transformation journey towards modern digitization.

Corporations are leveraging new capabilities, building out departments and increasing hiring. However, these efforts have a new set of challenges:

- Making the data available to the diverse set of engineers (Data engineers, analysts, data scientists) 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 Data Intelligence Platform that includes computing, storage, connectivity, capabilities built on Cisco Unified Computing System (Cisco UCS) infrastructure, using Cisco UCS C-Series and S-Series Rack Servers and unified management with Cisco Intersight to help companies manage the entire infrastructure from a single pane of glass along with Cloudera Data Platform to provide the software for fast ingest of data and managing and processing exabyte scale data being collected. This architecture is specifically designed for performance and linear scalability for big data and machine learning workload.

Audience

The intended audience for this document includes, but is not limited to, sales engineers, field consultants, professional services, IT managers, IT engineers, partners, and customers who are interested in learning about and deploying the Cloudera Data Platform Private Cloud (CDP PvC) on the Cisco Data Intelligence Platform on Cisco UCS M6 Rack-Mount servers and Cisco UCS X-Series for digital transformation through cloud-native modern data analytics and AI/ML.

Purpose of this Document

This document describes the architecture, installation, configuration, and validated use cases for the Cisco Data Intelligence Platform using Cloudera Data Platform Private Cloud Base and NVIDIA RAPIDS on Cisco UCS M6 Rack-Mount servers. A reference architecture is provided to configure the Cloudera Data Platform on Cisco UCS C240 M6 with Nvidia A100 GPU.

What's New in this Release?

This solution extends the portfolio of Cisco Data Intelligence Platform (CDIP) architecture with Cloudera Data Platform Private Cloud Base (CDP PvC Base), a state-of-the-art platform, providing a data cloud for demanding workloads that is easy to deploy, scale and manage. Furthermore, as the enterprise's requirements and needs changes overtime, the platform can grow to thousands of servers, at exabytes of storage and tens of thousands of cores to process this data.

The following will be implemented in this validated design:

- Cisco Intersight to configure and manage Cisco Infrastructure
- Data lake provided by Cloudera Data Platform Private Cloud Base on Cisco UCS servers
- CDS 3.3 powered by Apache Spark with GPU support
- NVIDIA RAPIDS to accelerate ETL and ML workflows without any code change

In this release, you will be exploring Cloudera Data Platform Private Cloud Base with Cloudera Data Science (CDS) with GPU support as an add-on service that enables RAPIDS Accelerator for Apache Spark.

Solution Summary

This chapter contains the following:

- [Cisco Data Intelligence Platform](#)
- [Reference Architecture](#)

This CVD details the process of installing CDP Private Cloud Base including the installation of CDS 3.3 powered by Apache Spark and NVIDIA GPU with RAPIDS accelerator and configuration details of the cluster.

Cisco Data Intelligence Platform

Cisco Data Intelligence Platform (CDIP) is a cloud-scale architecture, primarily for a private cloud data lake 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 provides the following:

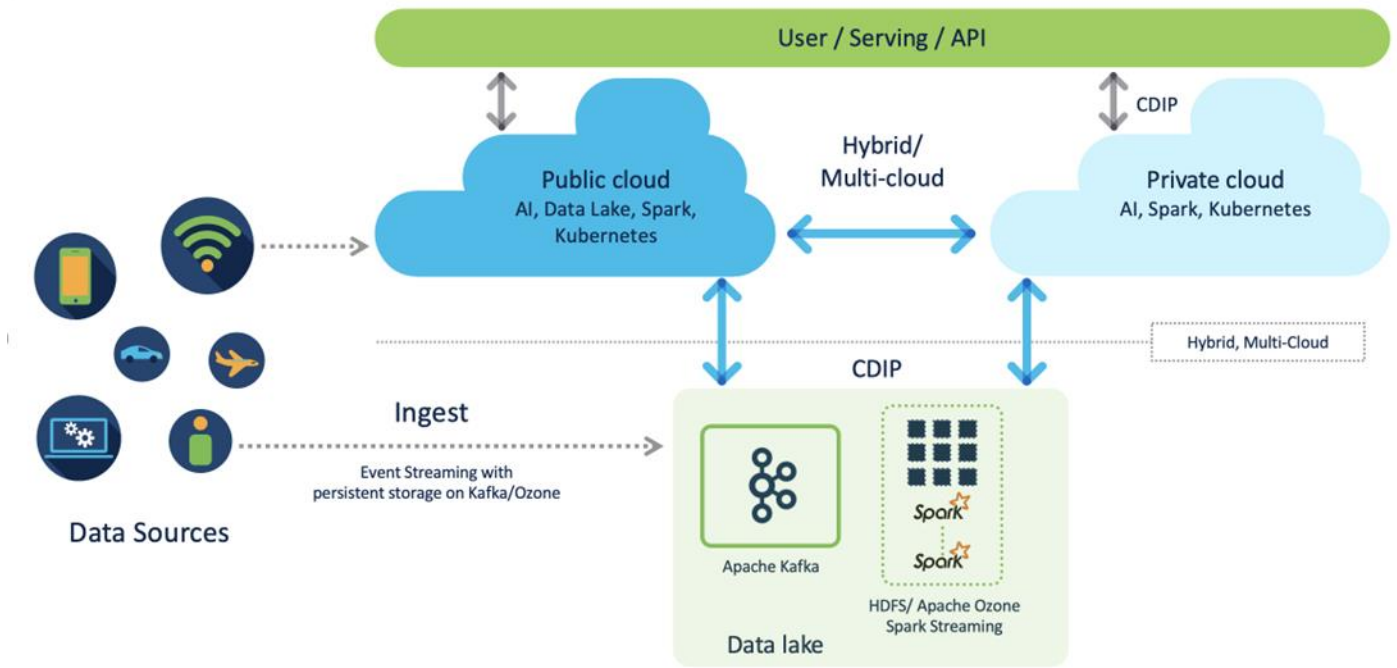
- Extremely fast data ingest, and data engineering done at the data lake.
- AI compute farm allowing for easy to manage different types of personas to work on AI/ML frameworks while achieving auto-scalability for different compute types (GPU, CPU, FPGA) to work on this data for further analytics.

Note: Cloudera Private Cloud Data Services 1.4 supports GPU only for Cloudera Machine Learning (CML). Cloudera Data Engineering (CDE) will support GPU in future release.

- 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. Next-generation Apache Ozone filesystem for storage in a data lake.
- Seamlessly scale the architecture to thousands of nodes with a single pane of glass management using Cisco Intersight and Cisco Application Centric Infrastructure (ACI).

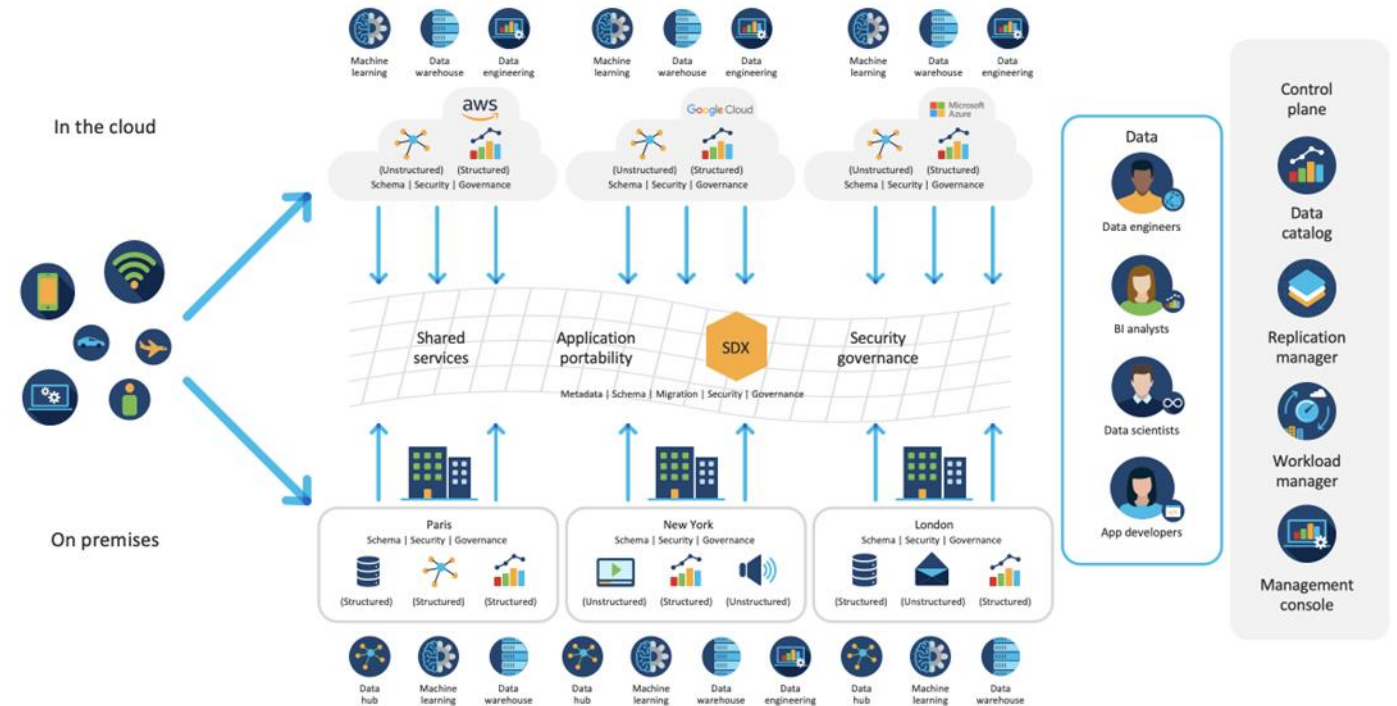
Cisco Data Intelligence Platform caters to the evolving architecture bringing together a fully scalable infrastructure with centralized management and fully supported software stack (in partnership with industry leaders in the space) to each of these three independently scalable components of the architecture including data lake, AI/ML and Object stores.

Figure 1. Cisco Data Intelligence Platform (CDIP) - Evolution of Data Lake to Hybrid Cloud



CDIP offers private cloud which enables it to become a hybrid cloud for the data lakes and apps which provides unified user experiences with common identity, single API framework that stretches from private cloud to public cloud, auto-scales when app demand grows. Further, implement tighter control over sensitive data with data governance and compliance, and integrate common data serving layer for data analytics, business intelligence, AI inferencing, and so on.

Figure 2. CDIP - Hybrid Cloud Architecture



CDIP with CDP private cloud is built to meet the needs of enterprises for their hybrid cloud with unmatched choices such as any data, any analytics, and engineering anywhere. This solution includes:

- **Flexibility** to run workload anywhere for quick and easy insights.
- **Security** that is consistent across all clouds provided by Cloudera's SDX. Write centrally controlled compliance and governance policies once and apply everywhere, enabling safe, secure, and compliant end-user access to data and analytics.
- **Performance and scale** to optimize TCO across your choices. It brings unparalleled scale and performance to your mission-critical applications while securing future readiness for evolving data models.
- **Single pane of glass** visibility for your infrastructure and workloads. Register multi-cloud, including public and private in a single management console and launch virtual analytic workspaces or virtual warehouses within each environment as needed.
- **Secure data and workload migration** to protect your enterprise data and deliver it where is needed. Securely manage data and meta-data migration across all environments.
- **Unified and multi-function Analytics** for cloud-native workloads whether real-time or batch. Integrates data management and analytics experiences across the entire data lifecycle for data anywhere.
- **Hybrid and multi-cloud data warehouse** service for all modern, self-service, and advanced analytics use cases, at scale.
- **Track and Audit everything** across entire ecosystem of CDIP deployments.

CDIP with CDP Private Cloud Hybrid Uses Cases

With the increasing hybrid cloud adoption due to increasing data volume and variety, CDIP addresses use cases that caters to the needs of today's demand of hybrid data platforms, such as the following:

- **Hybrid Workload** - Offload workload on-premises to cloud or vice-versa as per the requirements or auto-scale during peak hours due to real-time urgency or seasonality Cloudera Replication Manager and Cloudera Workload Manager
- **Hybrid Pipelines** - Implement and optimize data pipelines for easier management. Automate and orchestrate your data pipelines as per demand or where it is needed the most. Implement secure data exchange between choice of your cloud and on-premises data hub at scale
- **Hybrid Data Integration** - Integrate data sources among clouds. Simplify application development or ML model training that needs on-premises data sources or cloud-native data stores
- **Hybrid DevOps** - Accelerate development with dev sandboxes in the cloud, however, production runs on-premises
- **Hybrid Data Applications** - Build applications that runs anywhere for cost, performance, and data residency

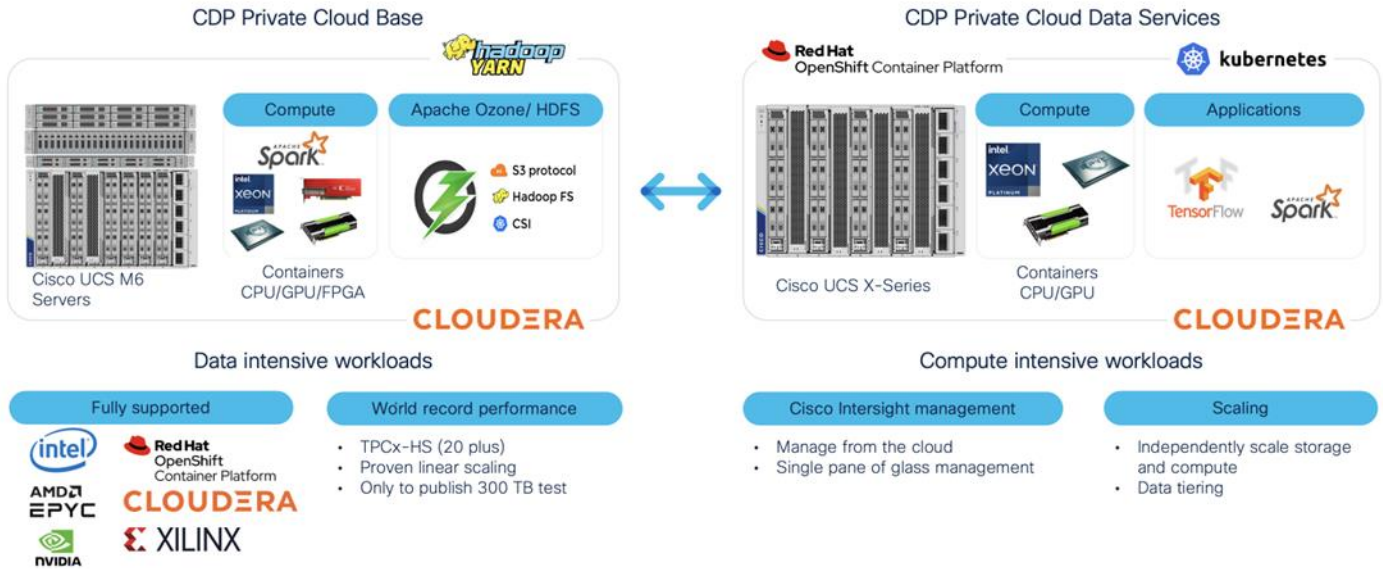
Cisco Data Intelligence Platform with Cloudera Data Platform

Cisco developed numerous industry leading Cisco Validated Designs (reference architectures) in the area of Big Data, compute farm with Kubernetes (CVD with RedHat OpenShift Container Platform) and Object store.

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

- Data lake enabled through CDP PvC Base
- Private Cloud with compute on Kubernetes can be enabled through CDP Private Cloud Data Services
- Exabyte storage enabled through Apache Ozone

Figure 3. Cisco Data Intelligent Platform with Cloudera Data Platform



This architecture can start from a single rack ([Figure 4](#)) and scale to thousands of nodes with a single pane of glass management with Cisco Application Centric Infrastructure (ACI) ([Figure 5](#)).

Figure 4. Cisco Data Intelligence Platform with Cloudera Data Platform Private Cloud Data Services

**2 x Cisco UCS
Fabric Interconnect**



**25/100Gb
connection from
each server**

**Data Intensive Workload
(CDP PvC Base)**

- 11 x Cisco UCS C240 M6 with CDP PvC Base
- 3 x CDP mgmt. node
- 8 x CDP data node with NVIDIA GPU

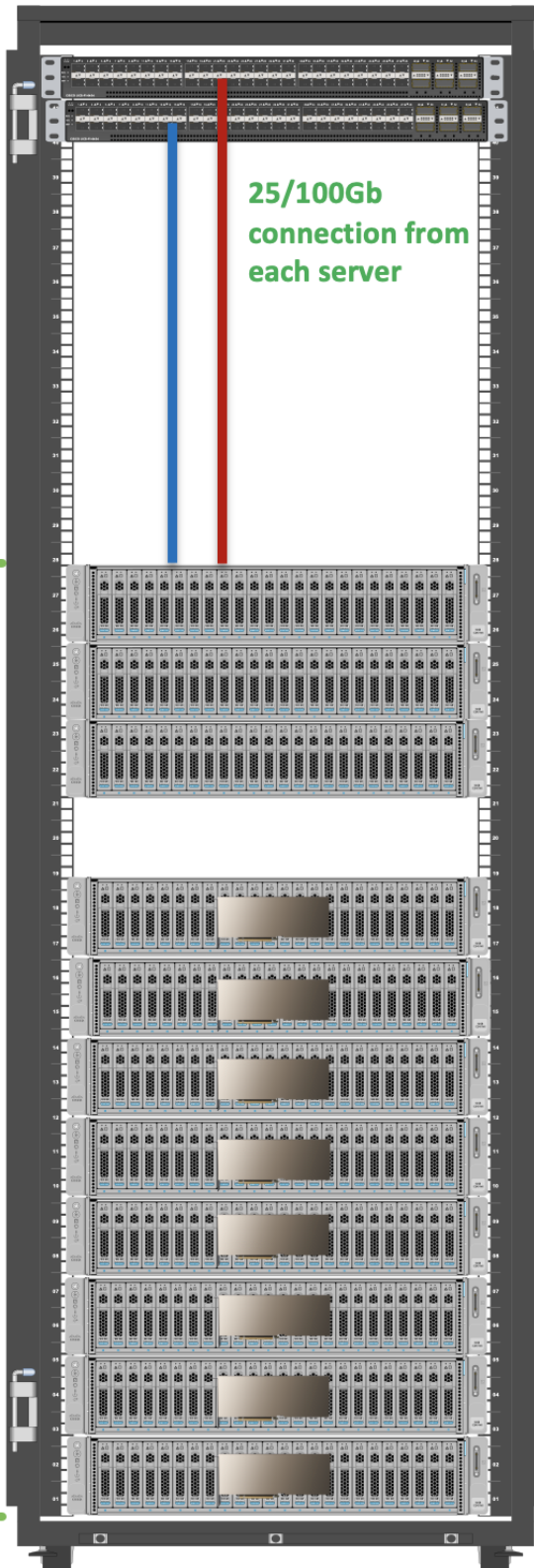
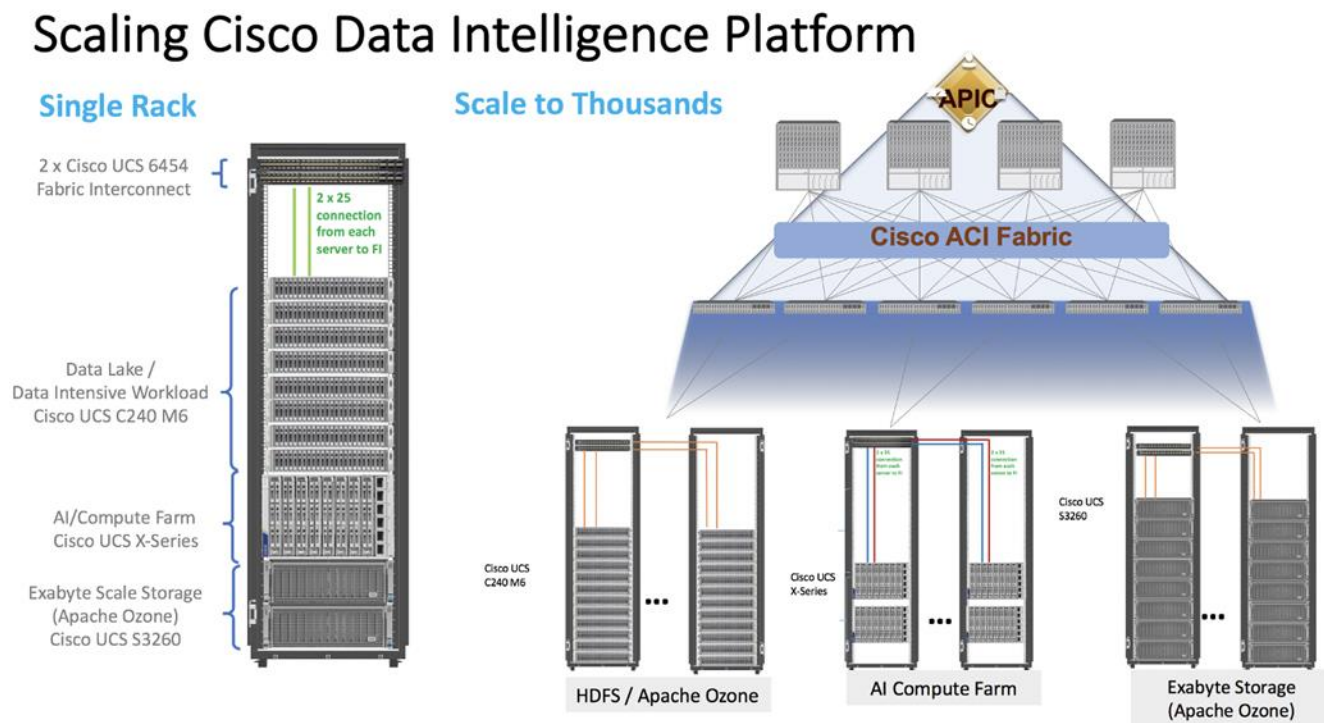


Figure 5. Cisco Data Intelligent Platform at Scale



Reference Architecture

Cisco Data Intelligence Platform reference architectures are carefully designed, optimized, and tested with the leading big data and analytics software distributions to achieve a balance of performance and capacity to address specific application requirements. You can deploy these configurations as is or use them as templates for building custom configurations. You can scale your solution as your workloads demand, including expansion to thousands of servers using Cisco Nexus 9000 Series Switches. The configurations vary in disk capacity, bandwidth, price, and performance characteristics.

Data Lake (CDP PvC Base) Reference Architecture

[Table 1](#) lists the CDIP with CDP PvC data lake and dense storage with Apache Ozone reference architecture.

Table 1. Cisco Data Intelligence Platform with CDP Private Cloud Base (Apache Ozone) Configuration on Cisco UCS M6

| | High Performance | Performance | Capacity |
|---------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Server | 16 x Cisco UCS C240 M6SN Rack Servers with small-form-factor (SFF) drives | 16 x Cisco UCS C240 M6 Rack Servers with small-form-factor (SFF) drives | 16 x Cisco UCS C240 M6 Rack Servers with large-form-factor (LFF) drives |
| CPU | 2 x 3 rd Gen Intel Xeon Scalable Processors 6338 processors (2 x 32 cores, at 2.0 GHz) | 2 x 3 rd Gen Intel Xeon Scalable Processors 6338 processors (2 x 32 cores, at 2.0 GHz) | 2 x 3 rd Gen Intel Xeon Scalable Processors 6338 processors (2 x 32 cores, at 2.0 GHz) |
| Memory | 16 x 32 GB RDIMM DRx4 3200 MHz (512 GB) | 16 x 32 GB RDIMM DRx4 3200 MHz (512 GB) | 16 x 32 GB RDIMM DRx4 3200 MHz (512 GB) |
| Boot | M.2 with 2 x 960-GB SSDs | M.2 with 2 x 960-GB SSDs | M.2 with 2 x 960-GB SSDs |
| Storage | 24 x 6.4TB 2.5in U2 NVMe and 2 | 24 x 2.4TB 12G SAS 10K RPM SFF HDD (4K) (or 24 x 7.6TB | 16 x 16TB 12G SAS 7.2K RPM LFF HDD(4K) and 2 x 3.2TB |

| | High Performance | Performance | Capacity |
|------------------------------|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| | x 3.2TB NVMe | Enterprise Value 12G SATA SSDs) and 2 x 3.2TB NVMe | NVMe |
| Virtual Interface Card (VIC) | Cisco UCS VIC 1467 (4 x 10/25G) Cisco UCS VIC 1477 (2 x 40/100G) Cisco UCS VIC 15428 (4 x 10/25/50G) | Cisco UCS VIC 1467 (4 x 10/25G) Cisco UCS VIC 1477 (2 x 40/100G) Cisco UCS VIC 15428 (4 x 10/25/50G) | Cisco UCS VIC 1467 (4 x 10/25G) Cisco UCS VIC 1477 (2 x 40/100G) Cisco UCS VIC 15428 (4 x 10/25/50G) |
| Storage Controller | NA | Cisco 12-Gbps SAS modular RAID controller with 4-GB flash-based write cache (FBWC) or Cisco 12-Gbps modular SAS host bus adapter (HBA) | Cisco 12-Gbps SAS modular RAID controller with 4-GB FBWC or Cisco 12-Gbps modular SAS host bus adapter (HBA) |
| Network Connectivity | Cisco UCS 6400 or 6500 Fabric Interconnect | Cisco UCS 6400 or 6500 Fabric Interconnect | Cisco UCS 6400 or 6500 Fabric Interconnect |
| GPU | NVIDIA GPU A100 | NVIDIA GPU A100 | NVIDIA GPU A100 |

Note: The reference architecture highlighted here is the sizing guide for Apache Ozone based deployment. When sizing data lake for HDFS, Cloudera doesn't support exceeding 100 TB per data node and drives larger than 8 TB. For more information, visit HDFS and Ozone section in CDP PvC Base hardware requirement: <https://docs.cloudera.com/cdp-private-cloud-base/7.1.7/installation/topics/cdpdc-runtime.html>

Compute Farm (CDP PvC DS) Reference Architecture

[Table 2](#) lists the CDIP with CDP PvC DS configuration for master and worker nodes with RHOC reference architecture.

Table 2. Cisco Data Intelligence Platform with CDP Private Cloud Data Services configuration

| | High Core Option |
|--------------------|----------------------------------------------------------------------------------------------------------------------|
| Servers | Cisco UCS X-Series 9508 chassis with X210C Blades (Up to 8 Per chassis) |
| CPU | 2 x 3 rd Gen Intel Xeon Scalable Processors 6338 processors (2 x 32 cores, at 2.0 GHz) |
| Memory | 16 x 64GB RDIMM DRx4 3200 MHz (1TB) |
| Boot | M.2 with 2 x 960GB SSD |
| Storage | 4 x 3.2TB 2.5in U2 NVMe* (Red Hat OpenShift Container Storage (RHOCS)/Portworx [2 drives], Local storage [2 drives]) |
| VIC | Cisco UCS VIC 14425 4x25G mLOM or Cisco UCS VIC 15231 2x100/200G mLOM |
| Storage controller | Cisco UCS X210c Compute Node compute pass through controller |

| High Core Option | |
|----------------------|--------------------------------------------|
| Network connectivity | Cisco UCS 6400 or 6500 Fabric Interconnect |
| GPU (optional) | Cisco UCS X440p with NVIDIA A100 GPU |

Figure 6. Cisco Data Intelligent Platform with CDP PvC – Reference Architecture

| Component | Configuration | Configuration | X-Series X210C |
|------------------|-----------------------------|-----------------------------|-----------------------------|
| Compute | 2 x 6330 (28C/2.0GHz) | 2 x 6330 (28C/2.0GHz) | 2 x 6330 (28C/2.0GHz) |
| Network | 5th Gen FI 6536 / VIC 15428 | 5th Gen FI 6536 / VIC 15428 | 5th Gen FI 6536 / VIC 15231 |
| Memory | 32G x 16 (512G) | 32G x 16 (512G) | 32G x 16 (512G) |
| Drives (Storage) | 10 x 2.4TB 10krpm SFF HDD | 4 x 3.8TB NVMe | 2 x 1.9TB NVMe |
| OS Drives | 2 x M.2 with 960GB | 2 x M.2 with 960GB | 2 x M.2 with 960GB |

Figure 7. Cisco Data Intelligent Platform with CDP PvC – Reference Architecture

| Component | Configuration (C240 M6) | Configuration (X-Series) |
|--------------------------|-----------------------------------------------------------------------------------------------------|-------------------------------------------------|
| Compute | 2 x 6338 (32C/2.0GHz) | 2 x 6338 (32C/2.0GHz) |
| Network | 5th Gen FI 6536 / VIC 15428 (4 x 50G mLOM) | 5th Gen FI 6536 / VIC 15231 (2 x 100/200G mLOM) |
| Memory | 32G x 16 (512G) | 64G x 16 (1024G) |
| Drives (Storage) | 24 x 2.4TB SFF or 16x16TB LFF HDD or 24 x 7.6 SSD/NVMe drives 2 x 3.8TB NVMe (Ozone metadata) | Up to 15.3 TB NVMe X 6 |
| OS Drives | 2 x M.2 with 960GB | 2 x M.2 with 960GB |
| GPU for AI/ML (optional) | NVidia A100 | Cisco UCS X440p with NVidia A100 |

Note: NVMe storage capacity and quantity needs to be updated based on the dataset requirement. For more information, visit CDP PvC DS with RHOCP hardware requirements: <https://docs.cloudera.com/cdp-private-cloud-data-services/1.3.4/installation/topics/cdppvc-installation-openshift-requirements.html>

Note: This deployment guide was tested with Cisco UCS Fabric Interconnect 6454 connected to Cisco UCS C240 M6 server with mLOM Cisco UCS VIC 1467.

As illustrated in [Figure 4](#), this CVD was designed with the following:

- Cisco UCS C240 M6 Rack Server with one NVIDIA A100 GPU Installed per node
- Cloudera Data Private Cloud Base 7.1.8
- CDS 3.3 powered by Apache Spark
- NVIDIA RAPIDS libraries for accelerated data science

Note: This deployment guide was tested with one Nvidia A100 GPU install per Cisco UCS C240 M6 server. Additionally, two more Nvidia A100 GPU can be installed per node with total three GPU node. For more details and GPU installation requirement on Cisco UCS C240 M6 visit:

<https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/c240m6-sff-specsheet.pdf>

Technology Overview

This chapter contains the following:

- [Cisco Data Intelligence Platform](#)
- [Cisco Unified Computing System](#)
- [Cisco UCS Fabric Interconnect](#)
- [Cloudera Data Platform \(CDP\)](#)
- [Cloudera Data Warehouse \(CDW\)](#)
- [Cloudera Data Engineering](#)

Cisco Data Intelligence Platform

This section describes the components used to build Cisco Data Intelligence Platform, a highly scalable architecture designed to meet a variety of scale-out application demands with seamless data integration and management integration capabilities.

Cisco Data Intelligence Platform powered by Cloudera Data Platform delivers:

- Latest generation of CPUs from Intel (3rd generation Intel Scalable family, with Ice Lake CPUs).
- Cloud scale and fully modular architecture where big data, AI/compute farm, and massive storage tiers work together as a single entity and each CDIP component can also scale independently to address the IT issues in the modern data center.
- World record Hadoop performance both for MapReduce and Spark frameworks published at [TPCx-HS benchmark](#).
- AI compute farm offers different types of AI frameworks and compute types (GPU, CPU, FPGA) to work data for analytics.
- A massive storage tier enables to gradually retire data and quick retrieval when needed on a storage dense sub-systems with a lower \$/TB providing a better TCO.
- Data compression with FPGA, offload compute-heavy compression tasks to FPGA, relieve CPU to perform other tasks, and gain significant performance.
- Seamlessly scale the architecture to thousands of nodes.
- Single pane of glass management with Cisco Intersight.
- ISV Partner ecosystem – Top notch ISV partner ecosystem, offering best of the breed end-to-end validated architectures.
- Pre-validated and fully supported platform.
- Disaggregate Architecture supports separation of storage and compute for a data lake.
- Container Cloud, Kubernetes, compute farm backed by the industry leading container orchestration engine and offers the very first container cloud plugged with data lake and object store.

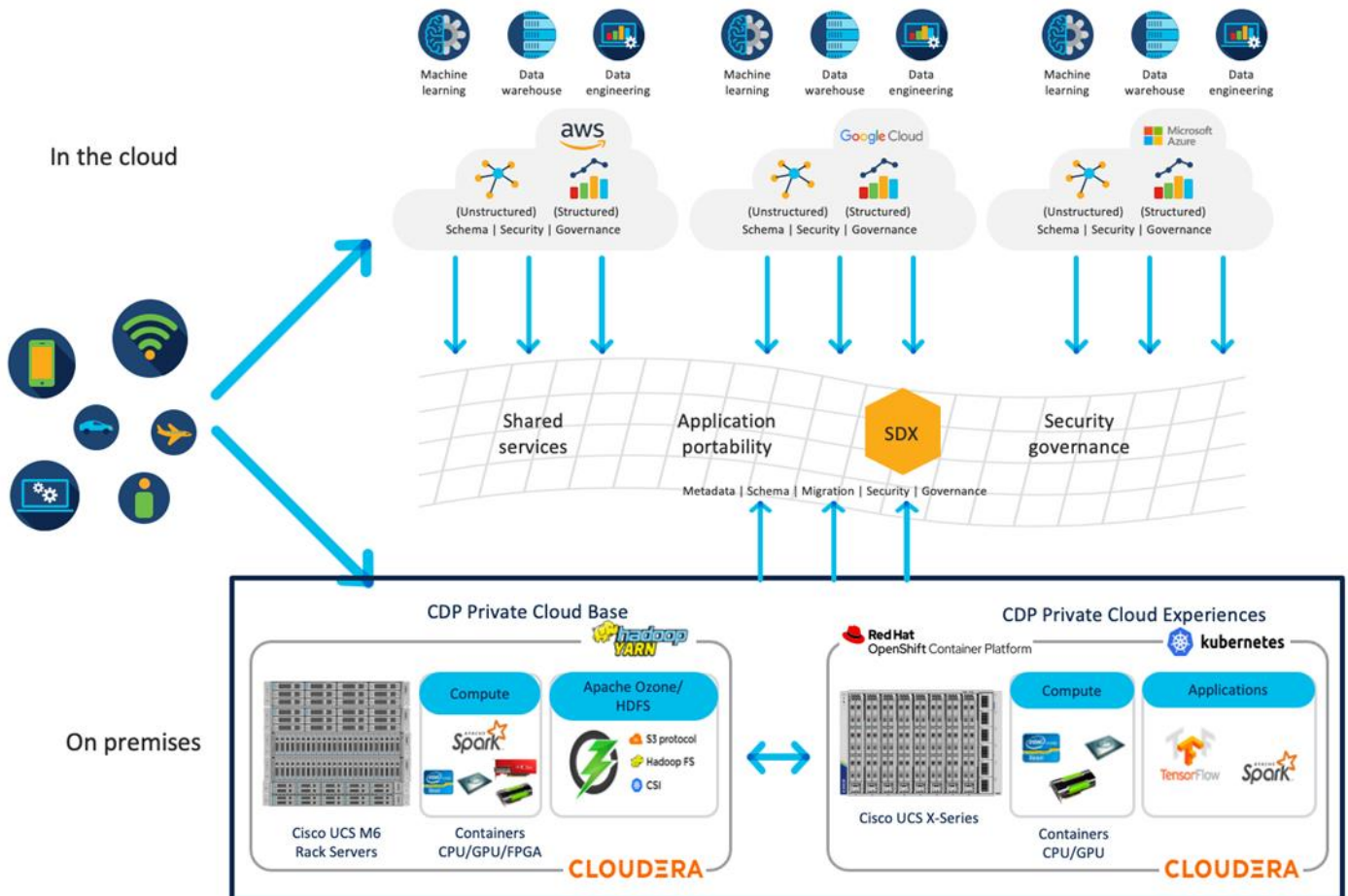
CDIP with CDP Hybrid Cloud Architecture

Cisco Data Intelligent Platform (CDIP) with Cloudera Data Platform (CDP) integrates different domains, such as specific layers of compute infrastructure between on-premises environments and public clouds. Integrations

can include moving a Kubernetes-based application to establish secure connectivity, user access, or policies per workloads between environments. These hybrid cloud architecture frameworks and operating models are better defined with the more encompassing term hybrid IT, which also includes multi-cloud scenarios enabling distributed nature of the infrastructure that can assure elasticity, scalability, performance, and efficiency as well as bring apps closer to their intended users with ability to cloud burst.

Red Hat OpenShift or Embedded Container Service (ECS) being the preferred container cloud platform for CDP private cloud and so is for CDIP, is the market leading Kubernetes powered container platform. This combination is the first enterprise data cloud with a powerful hybrid architecture that decouples compute and storage for greater agility, ease-of-use, and more efficient use of private and multi-cloud infrastructure resources. With Cloudera's Shared Data Experience (SDX), security and governance policies can be easily and consistently enforced across data and analytics in private as well as multi-cloud deployments. This hybridity will open myriad opportunities for seamless portability of workloads and applications for multi-function integration with other frameworks such as streaming data, batch workloads, analytics, data pipelining/engineering, and machine learning.

Figure 8. CDIP with CDP PvC - Hybrid Cloud Architecture



Cloud Native Architecture for Data Lake and AI

Cisco Data Intelligence Platform with CDP private cloud accelerates the process of becoming cloud-native for your data lake and AI/ML workloads. By leveraging Kubernetes powered container cloud, enterprises can now quickly break the silos in monolithic application frameworks and embrace a continuous innovation of micro-

services architecture with CI/CD approach. With cloud-native ecosystem, enterprises can build scalable and elastic modern applications that extends the boundaries from private cloud to hybrid.

Containerization

Hadoop 3.0 introduced production-ready Docker container support on YARN with GPU isolation and scheduling. This created plethora of opportunities for modern applications, such as micro-services and distributed applications frameworks comprised of 1000s of containers to execute AI/ML algorithms on peta bytes of data with ease and in a speedy fashion.

Docker support in Apache Hadoop 3 can be leveraged by Apache Spark for addressing long standing challenges related to software dependencies to be installed on all hosts where Spark executors run in the cluster. By converting Spark application's on YARN side by side in docker containers with custom packages, users can bring their own versions of python, libraries, without heavy involvement of admins and have an efficient solution with docker image layer caching.

Cisco Unified Computing System

Cisco Unified Computing System (Cisco UCS) is a next-generation data center platform that integrates computing, networking, storage access, and virtualization resources into a cohesive system designed to reduce total cost of ownership and increase business agility. The system integrates a low-latency, lossless 10-100 Gigabit Ethernet unified network fabric with enterprise-class, x86-architecture servers. The system is an integrated, scalable, multi-chassis platform with a unified management domain for managing all resources.

Cisco UCS Differentiators

Cisco Unified Computing System is revolutionizing the way servers are managed in the datacenter. The following are the unique differentiators of Cisco Unified Computing System and Cisco UCS Manager:

- **Embedded Management**—In Cisco UCS, the servers are managed by the embedded firmware in the Fabric Inter-connects, eliminating the need for any external physical or virtual devices to manage the servers.
- **Unified Fabric**—In Cisco UCS, from blade server chassis or rack servers to FI, there is a single Ethernet cable used for LAN, SAN, and management traffic. This converged I/O results in reduced cables, SFPs and adapters - reducing capital and operational expenses of the overall solution.
- **Auto Discovery**—By simply inserting the blade server in the chassis or connecting the rack server to the fabric interconnect, discovery and inventory of compute resources occurs automatically without any management intervention. The combination of unified fabric and auto-discovery enables the wire-once architecture of Cisco UCS, where compute capability of Cisco UCS can be extended easily while keeping the existing external connectivity to LAN, SAN, and management networks.

Cisco UCS Manager

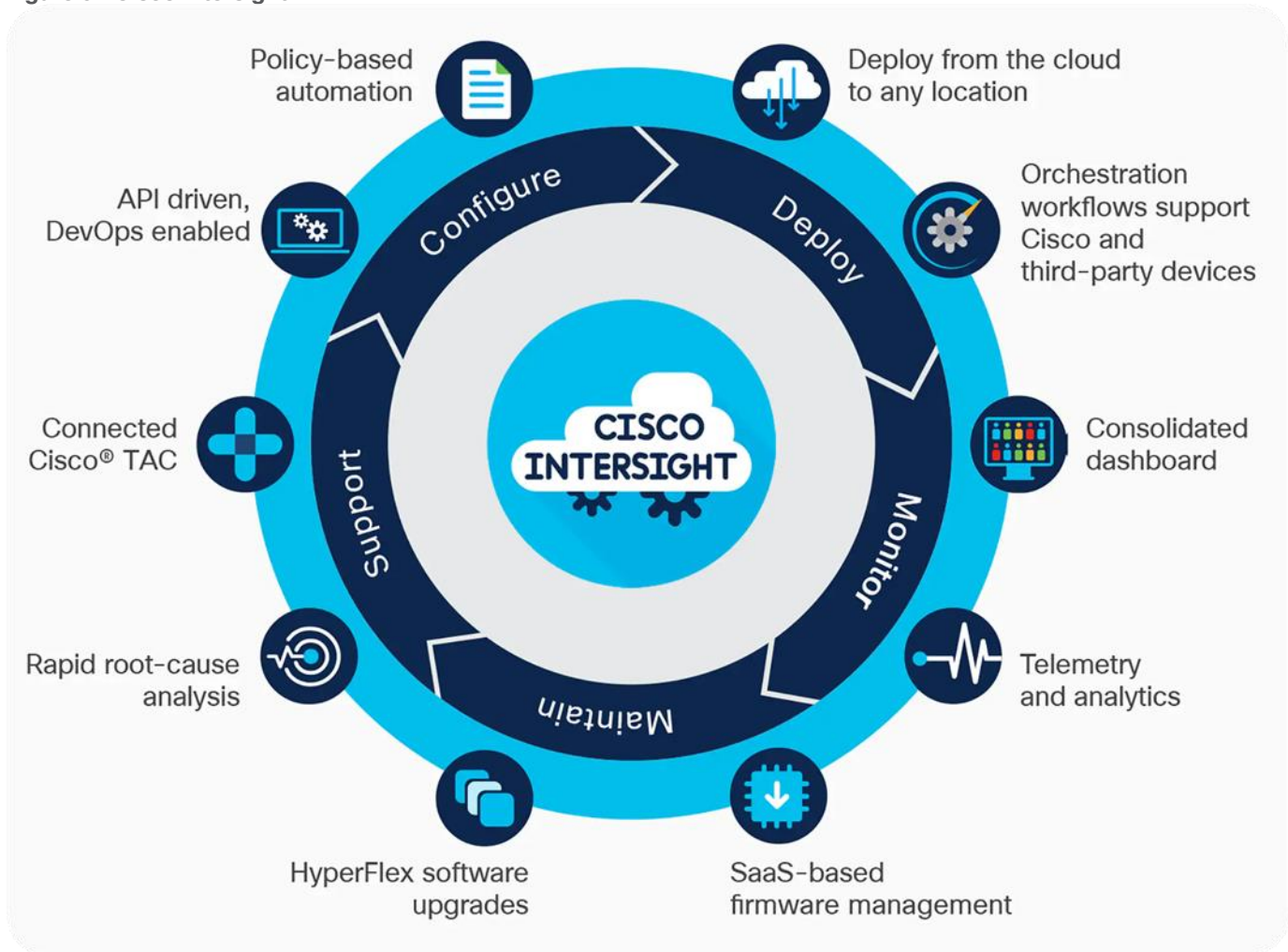
Cisco UCS Manager (UCSM) provides unified, integrated management for all software and hardware components in Cisco UCS. Using Cisco Single Connect technology, it manages, controls, and administers multiple chassis for thousands of virtual machines. Administrators use the software to manage the entire Cisco Unified Computing System as a single logical entity through an intuitive graphical user interface (GUI), a command-line interface (CLI), or a through a robust application programming interface (API).

Cisco Intersight

Cisco Intersight is a lifecycle management platform for your infrastructure, regardless of where it resides. In your enterprise data center, at the edge, in remote and branch offices, at retail and industrial sites—all these locations present unique management challenges and have typically required separate tools. Cisco Intersight Software as

a Service (SaaS) unifies and simplifies your experience of the Cisco Unified Computing System (Cisco UCS) and Cisco HyperFlex systems. See [Figure 9](#).

Figure 9. Cisco Intersight

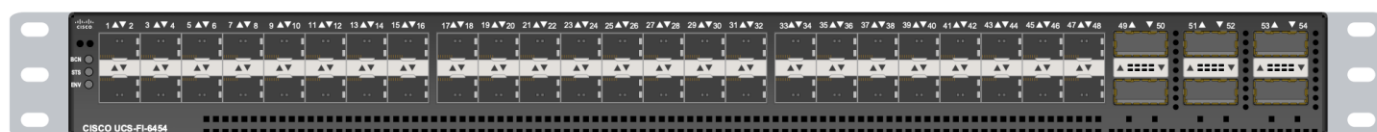


Cisco UCS Fabric Interconnect

The Cisco UCS Fabric Interconnect (FI) is a core part of the Cisco Unified Computing System, providing both network connectivity and management capabilities for the system. Depending on the model chosen, the Cisco UCS Fabric Interconnect offers line-rate, low-latency, lossless 10/25/40/100 Gigabit Ethernet, Fibre Channel over Ethernet (FCoE) and Fibre Channel connectivity. Cisco UCS Fabric Interconnects provide the management and communication backbone for the Cisco UCS C-Series, B-Series and X-Series Blade Servers, and 9508 Series Blade Server Chassis. All servers and chassis, and therefore all blades, attached to the Cisco UCS Fabric Interconnects become part of a single, highly available management domain. In addition, by supporting unified fabrics, the Cisco UCS Fabric Interconnects provide both the LAN and SAN connectivity for all servers within its domain.

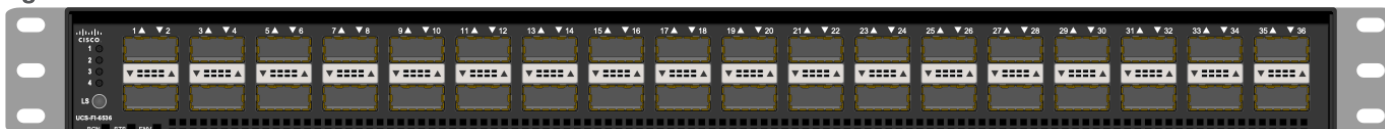
The Cisco UCS 6454 54-Port Fabric Interconnect ([Figure 10](#)) is a One-Rack-Unit (1RU) 10/25/40/100 Gigabit Ethernet, FCoE, and Fibre Channel switch offering up to 3.82 Tbps throughput and up to 54 ports. The switch has 28 10/25-Gbps Ethernet ports, 4 1/10/25-Gbps Ethernet ports, 6 40/100-Gbps Ethernet uplink ports, and 16 unified ports that can support 10/25-Gbps Ethernet ports or 8/16/32-Gbps Fibre Channel ports. All Ethernet ports are capable of supporting FCoE.

Figure 10. Cisco UCS 6454 Fabric Interconnect



The Cisco UCS 6536 36-Port Fabric Interconnect ([Figure 11](#)) is a One-Rack-Unit (1RU) 10/25/40/100 Gigabit Ethernet, FCoE, and Fibre Channel switch offering up to 7.42 Tbps throughput and up to 36 ports. The switch has 32 40/100-Gbps Ethernet ports and 4 unified ports that can support 40/100-Gbps Ethernet ports or 16 Fiber Channel ports after break-out at 8/16/32-Gbps FC speeds. The 16 FC ports after breakout can either operate as an FC uplink port or as an FC storage port. The switch supports 2 1-Gbps speed after breakout and all 36 ports can breakout for 10/25-Gbps Ethernet connectivity. All Ethernet ports are capable of supporting FCoE.

Figure 11. Cisco UCS 6536 Fabric Interconnect



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 M6 Rack-Mount Server

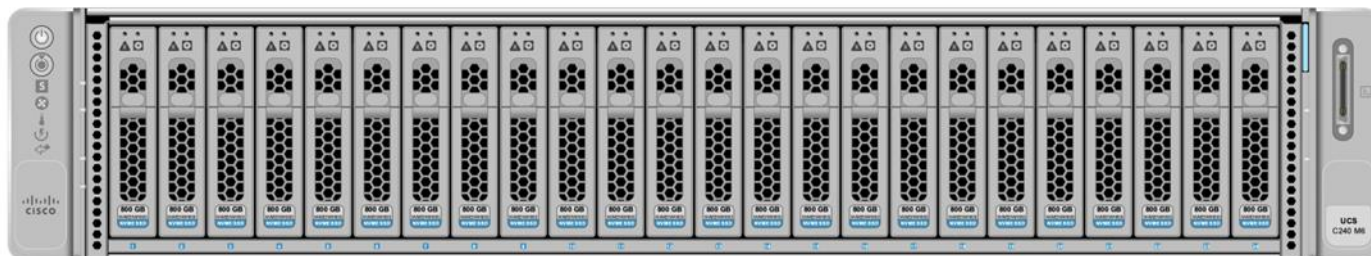
The Cisco UCS C240 M6 Rack Server ([Figure 12](#)) is well-suited for a wide range of storage and I/O-intensive applications such as big data analytics, databases, collaboration, virtualization, consolidation, and high-performance computing in its two-socket, 2RU form factor.

The Cisco UCS C240 M6 Server extends the capabilities of the Cisco UCS rack server portfolio with 3rd Gen Intel Xeon Scalable Processors supporting more than 43 percent more cores per socket and 33 percent more memory when compared with the previous generation.

You can deploy the Cisco UCS C-Series rack servers as standalone servers or as part of the Cisco Unified Computing System managed by Cisco Intersight, or Intersight Managed Mode to take advantage of Cisco standards-based unified computing innovations that can help reduce your total cost of ownership (TCO) and increase your business agility.

These improvements deliver significant performance and efficiency gains that will improve your application performance. The Cisco UCS C240 M6 Rack Server delivers outstanding levels of expandability and performance.

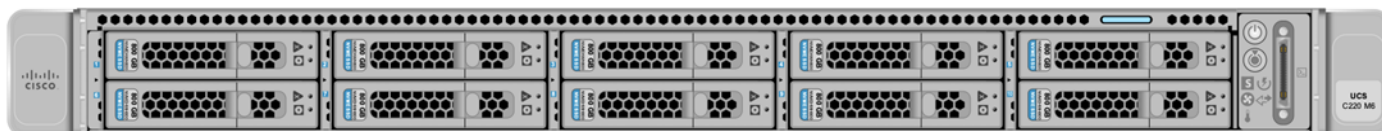
Figure 12. Cisco UCS C240 M6



The Cisco UCS C220 M6 Rack Server ([Figure 13](#)) is the most versatile general-purpose infrastructure and application server in the industry. This high-density, 1RU, 2-socket rack server delivers industry-leading performance and efficiency for a wide range of workloads, including virtualization, collaboration, and bare-metal applications. You can deploy the Cisco UCS C-Series Rack Servers as standalone servers or as part of the Cisco Unified Computing System managed by Cisco Intersight, Cisco UCS Manager, or Intersight Managed Mode to take advantage of Cisco standards-based unified computing innovations that can help reduce your Total Cost of Ownership (TCO) and increase your business agility.

The Cisco UCS C220 M6 Rack Server extends the capabilities of the Cisco UCS rack server portfolio. The Cisco UCS C220 M6 Rack Server delivers outstanding levels of expandability and performance.

Figure 13. Cisco UCS C220 M6



Cisco UCS X-Series Modular System

The Cisco UCS X-Series with Cisco Intersight is a modular system managed from the cloud. It is designed to meet the needs of modern applications and improve operational efficiency, agility, and scale through an adaptable, future-ready, modular design.

Designed to deploy and automate hybrid cloud environments:

- Simplify with cloud-operated infrastructure
- Simplify with an adaptable system designed for modern applications
- Simplify with a system engineered for the future

Figure 14. Cisco UCS X9508 Chassis front and rear view

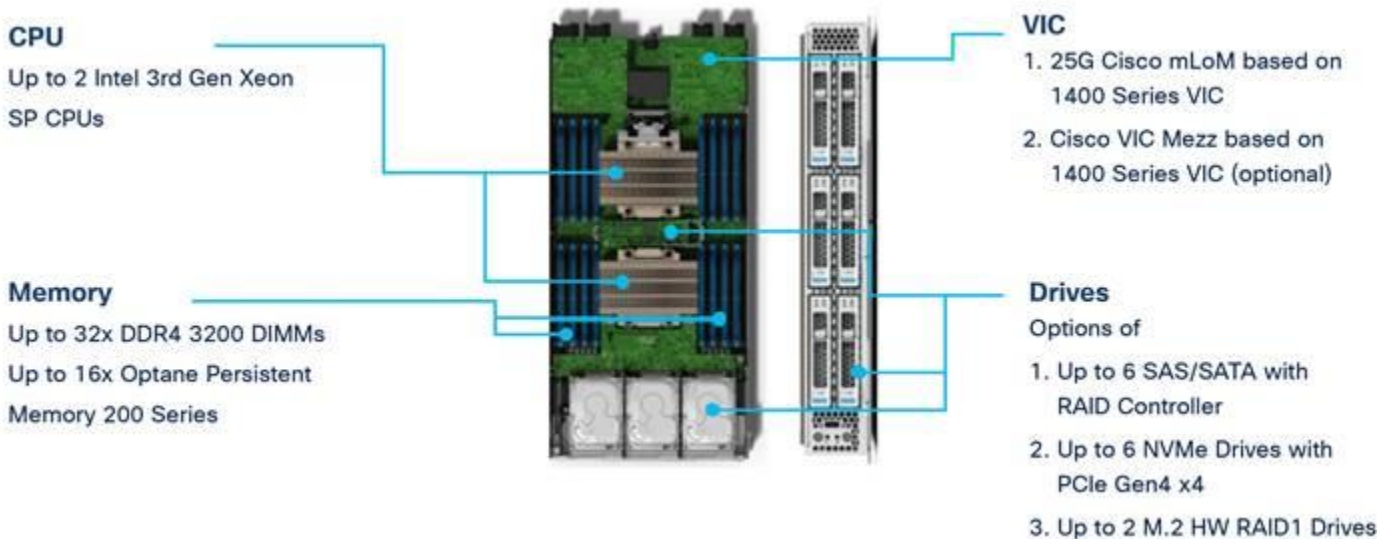


For more details, go to: <https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-x-series-modular-system/x9508-specsheet.pdf>

Cisco UCS X210c Compute Node

The Cisco UCS X210c M6 Compute Node is the first computing device to integrate into the Cisco UCS X-Series Modular System. Up to eight compute nodes can reside in the 7-Rack-Unit (7RU) Cisco UCS X9508 Chassis, offering one of the highest densities of compute, IO, and storage per rack unit in the industry.

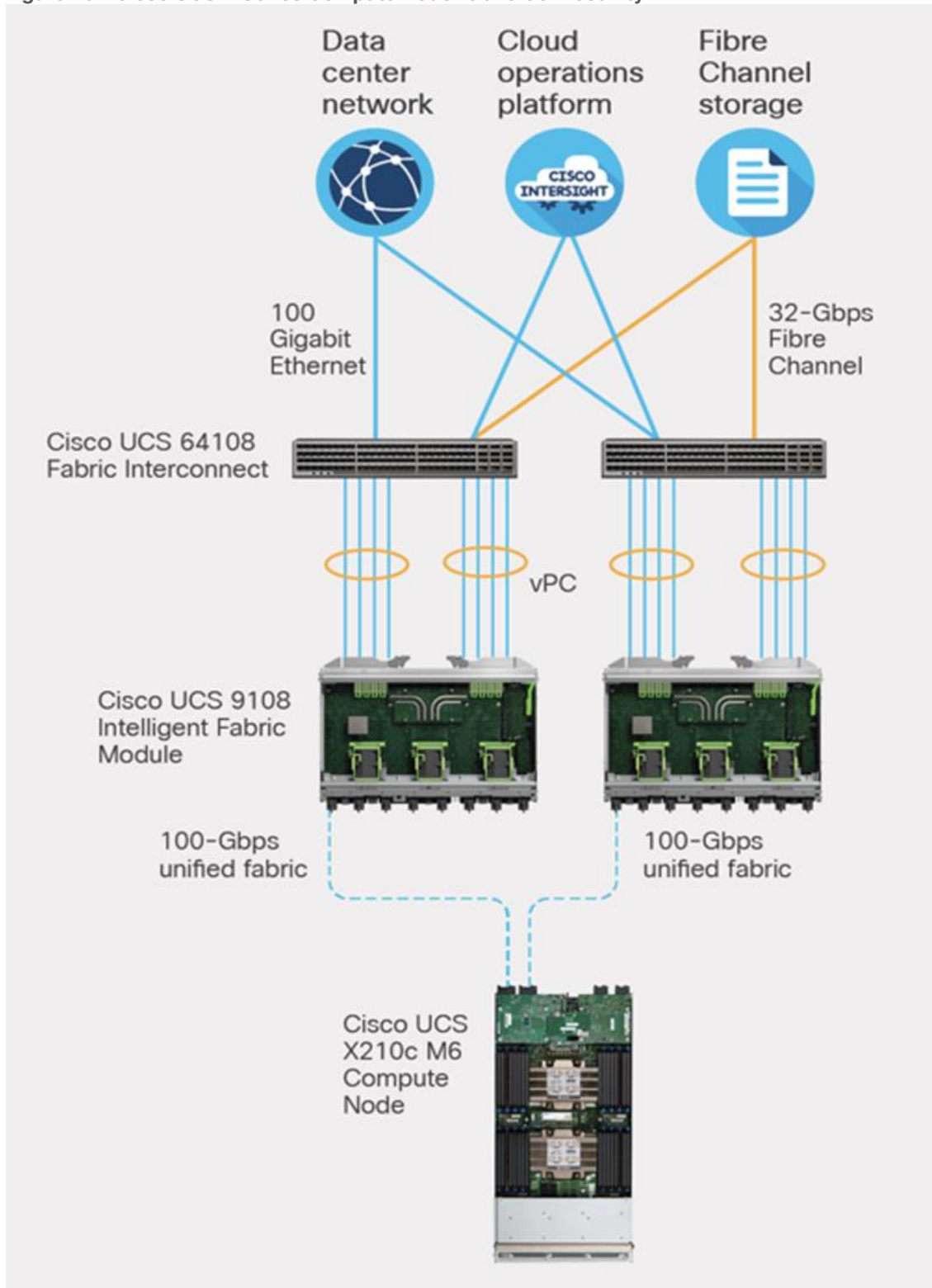
Figure 15. Cisco UCS X210c M6 Compute Node



Unified Fabric Connectivity

A unified fabric interconnects all devices in the system. It securely carries all traffic to the fabric interconnects where it can be broken out into IP networking, Fibre Channel SAN, and management connectivity.

Figure 16. Cisco UCS X Series Compute Node Fabric Connectivity



Cisco UCS X440p PCIe Node

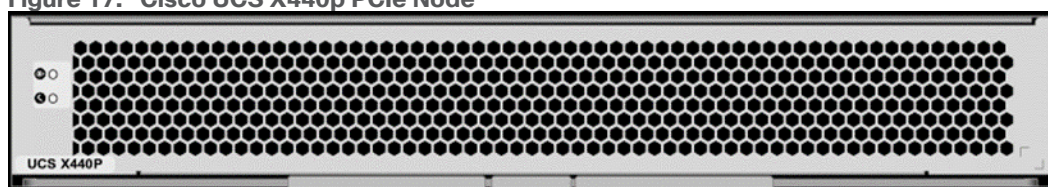
The Cisco UCS X440p PCIe Node ([Figure 17](#)) is the first PCIe resource node to integrate into the Cisco UCS X-Series Modular System. The Cisco UCS X9508 Chassis has eight node slots, up to four of which can be X440p PCIe nodes when paired with a Cisco UCS X210c M6 Compute Node. The Cisco UCS X440p PCIe Node

supports two x16 full-height, full-length dual slot PCIe cards, or four x8 full-height, full-length single slot PCIe cards and requires both Cisco UCS 9416 X-Fabric modules for PCIe connectivity. This provides up to 16 GPUs per chassis to accelerate your applications with the Cisco UCS X440p Nodes. If your application needs even more GPU acceleration, up to two additional GPUs can be added on each Cisco UCS X210c compute node.

Benefits include:

- Accelerate more workloads with up to four GPUs
- Make it easy to add, update, and remove GPUs to Cisco UCS X210c M6 Compute Nodes
- Get a zero-cable solution for improved reliability and ease of installation
- Have industry standard PCIe Gen 4 connections for compatibility

Figure 17. Cisco UCS X440p PCIe Node

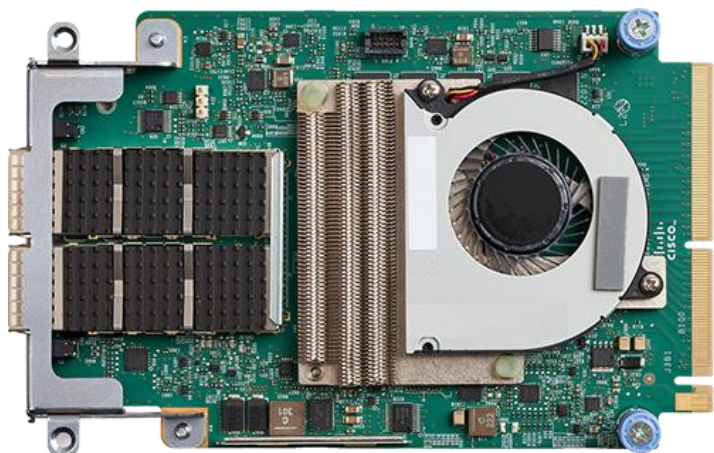


Cisco UCS Virtual Interface Cards

The Cisco UCS Virtual Interface Card (VIC) extends the network fabric directly to both servers and virtual machines so that a single connectivity mechanism can be used to connect both physical and virtual servers with the same level of visibility and control. Cisco® VICs provide complete programmability of the Cisco UCS I/O infrastructure, with the number and type of I/O interfaces configurable on demand with a zero-touch model.

Cisco VICs support Cisco SingleConnect technology, which provides an easy, intelligent, and efficient way to connect and manage computing in your data center. Cisco SingleConnect unifies LAN, SAN, and systems management into one simplified link for rack servers, blade servers, and virtual machines. This technology reduces the number of network adapters, cables, and switches needed and radically simplifies the network, reducing complexity. Cisco VICs can support 512 PCI 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 10/25/50/100/200-Gbps connection to servers. The PCIe Generation 4 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.

Figure 18. Cisco UCS VIC 15238



For more details go to: <https://www.cisco.com/c/en/us/products/interfaces-modules/unified-computing-system-adapters/index.html>

Ready for a Hybrid Cloud World

The Cisco Intersight cloud operations platform is the force that transforms the Cisco UCS X-Series Modular System from a set of components into a flexible server platform to propel your most important workloads.

The Cisco UCS X-Series with Intersight is built with a common purpose: to make hardware think like software so that you can easily adapt to a rapidly changing world. Through server profiles, Intersight defines the identity, connectivity, and I/O configuration of your servers and automates the entire infrastructure lifecycle. It's easy to imagine how, as more features are released, the modular system supports a pool of I/O resources: banks of nonvolatile memory, GPU accelerators, specialized ASICs, and massive amounts of NVMe storage. Just as the chassis and Cisco UCS X-Fabric technology are designed to incorporate a constant flow of new capabilities, Cisco Intersight is designed to automatically integrate those technologies into servers along with a constant flow of new, higher-level management capabilities. Software as a service (SaaS) meets modular, infrastructure as code, and the line between hardware and software dissolves.

In its [FutureScape: Worldwide IT Industry 2020 Predictions report](#), IDC predicts that, by 2023, 300 percent more applications will run in the data center and edge locations, 500 million digital applications and services will be developed using cloud-native approaches, and more than 40 percent of new enterprise IT infrastructure will be deployed at the edge. This means that you need a consistent operational approach for all of your infrastructure, wherever it is deployed. With Cisco Intersight and the Cisco UCS X-Series you can:

- Define desired system configurations based on policies that use pools of resources provided by the Cisco UCS X-Series. Let Cisco Intersight assemble the components and set up everything from firmware levels to which I/O devices are connected. Infrastructure is code, so your IT organization can use the Cisco Intersight GUI, and your DevOps teams can use the Intersight API, the Intersight Service for HashiCorp Terraform, or the many API bindings from languages such as Python and PowerShell.
- Deploy from the cloud to any location. Anywhere the cloud reaches, Cisco Intersight can automate your IT processes. We take the guesswork out of implementing new services with a curated set of services we bundle with the Intersight Kubernetes Service, for example.
- Visualize the interdependencies between software components and how they use the infrastructure that supports them with Intersight Workload Optimizer.

-
- Optimize your workload by analyzing runtime performance and make resource adjustments and workload placements to keep response time within a desired range. If your first attempt at matching resources to workloads doesn't deliver the results you need, you can reshape the system quickly and easily. Cisco Intersight facilitates deploying workloads into your private cloud and into the public cloud. Now one framework bridges your core, cloud, and edge infrastructure, managing infrastructure and workloads wherever they are deployed.
 - Maintain your infrastructure with a consolidated dashboard of infrastructure components regardless of location. Ongoing telemetry and analytics give early detection of possible failures. Reduce risk of configuration drift and inconsistent configurations through automation with global policy enforcement.
 - Support your infrastructure with AI-driven root-cause analysis and automated case support for the always-connected Cisco Technical Assistance Center (Cisco TAC). Intersight watches over you when you update your solution stack, helping to prevent incompatible hardware, firmware, operating system, and hypervisor configurations.

Modular Management Architecture

Cisco Intersight is a unified, secure, modular platform that consists of a set of services that bridge applications and infrastructure to meet your specific needs, including:

- Intersight Infrastructure Service
Manage your infrastructure lifecycle, including Cisco data center products, Cisco converged infrastructure solutions, and third-party endpoints
- Intersight Workload Optimizer
Revolutionize how you manage application resources across any environment with real-time, full-stack visibility to help ensure performance and better cost control
- Intersight Kubernetes Service
Simplify Kubernetes with automated lifecycle management across your multi-cloud environment
- Intersight Virtualization Service
Deploy and manage virtual machines on premises or in the cloud
- Intersight Cloud Orchestrator
Standardize application lifecycle management across multiple clouds

Cisco Intersight

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

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

Intersight recommendation engine provides actionable intelligence for IT operations management. The insights are driven by expert systems and best practices from Cisco.

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

Cisco Intersight provides the following features for ease of operations and administration for the IT staff:

- Connected TAC
- Security Advisories
- Hardware Compatibility List (HCL)

To learn more about all the features of Cisco Intersight, go to:

<https://www.cisco.com/c/en/us/products/servers-unified-computing/intersight/index.html>

Connected TAC

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

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

Figure 19. Cisco Intersight: Connected TAC

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

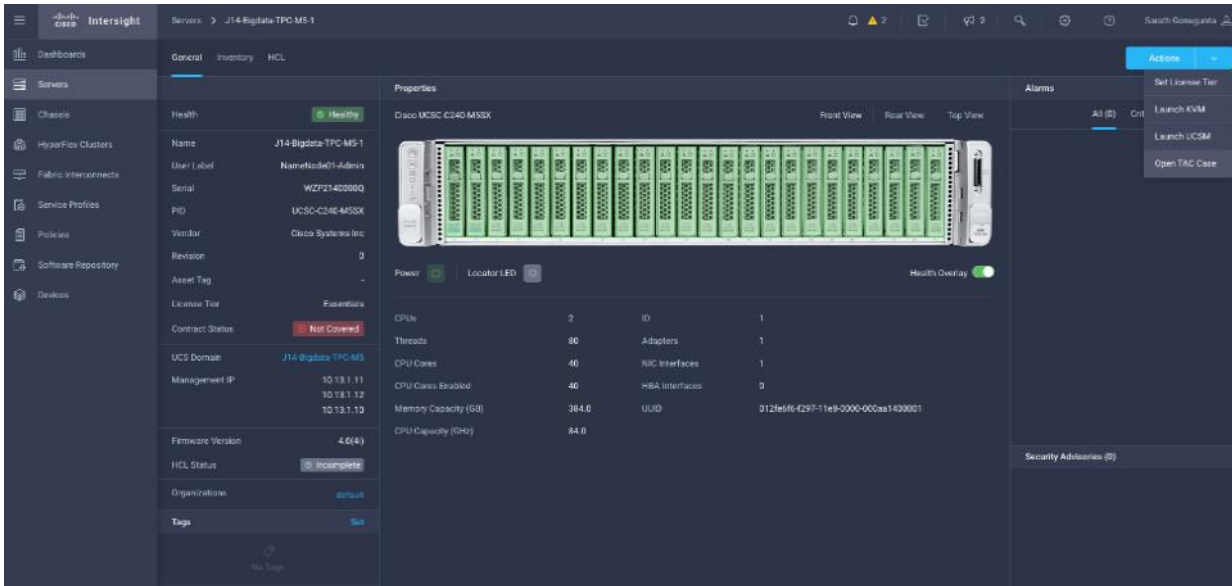


Procedure 1. Enable Connected TAC

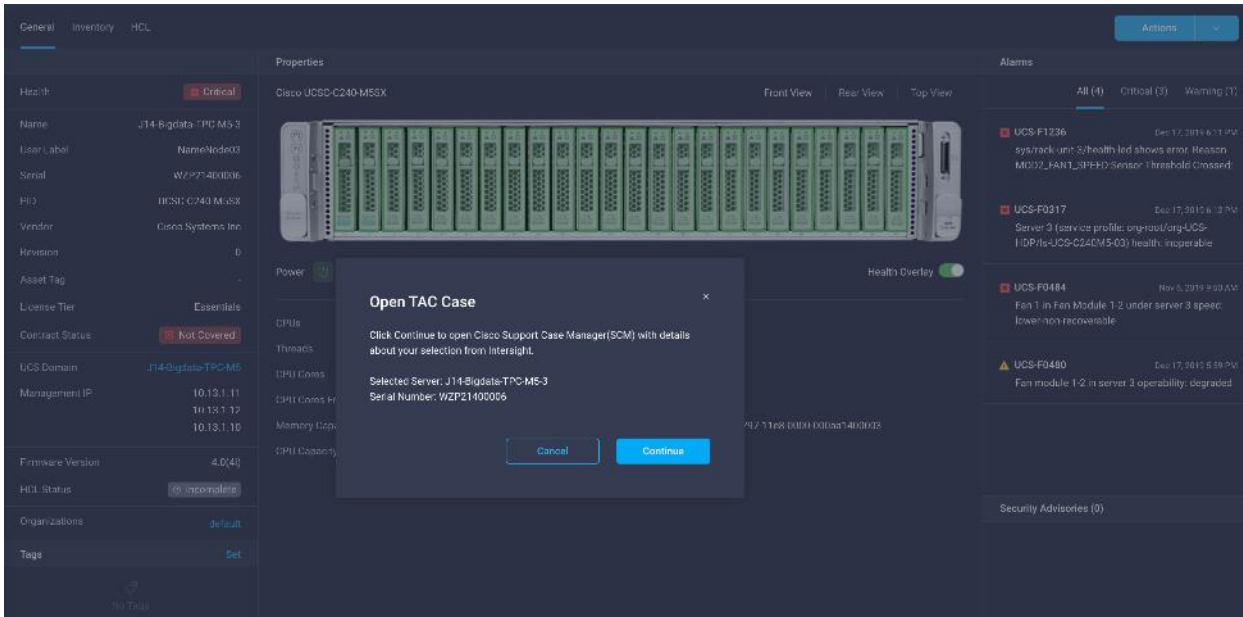
Step 1. Log into intersight.com.

Step 2. Click the Servers tab. Go to Server > Actions tab. From the drop-down list, click Open TAC Case.

Step 3. Click Open TAC Case to launch the Cisco URL for the support case manager where associated service contracts for Server or Fabric Interconnect is displayed.



Step 4. Click Continue.



Step 5. Follow the procedure to Open TAC Case.

Cisco Intersight Integration for HCL

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

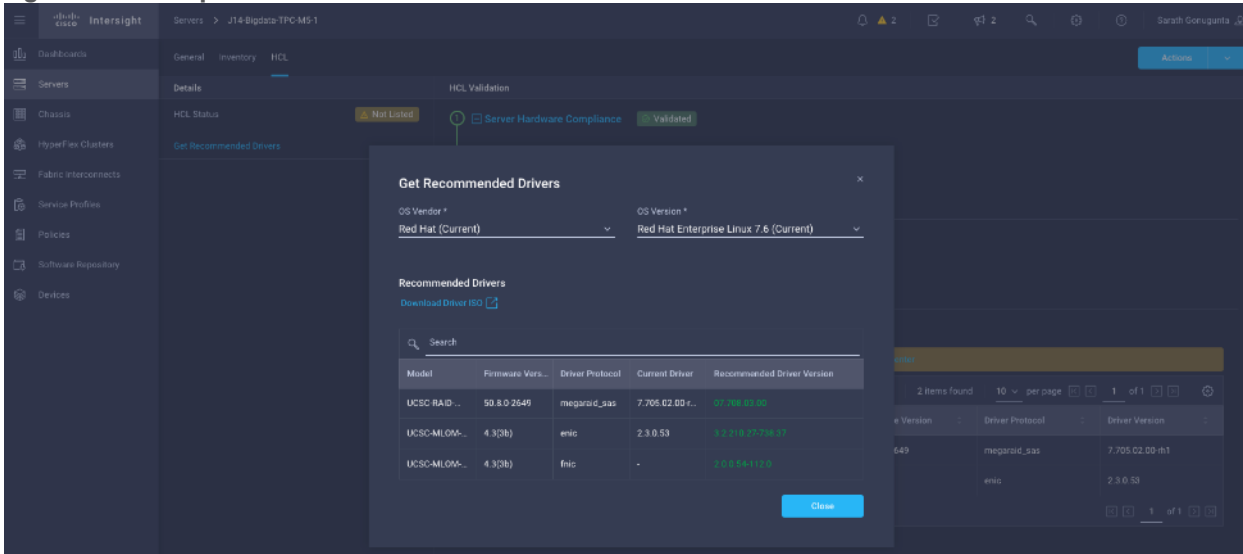
You can use Cisco UCS Tools, a host utility vSphere Installation Bundle (VIB), or OS Discovery Tool, an open-source script to collect OS and driver information to evaluate HCL compliance.

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

For more information, go to:

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

Figure 20. Example of HCL Status and OS Driver Recommendation



Advisories (PSIRTs)

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

Figure 21. Intersight Dashboard

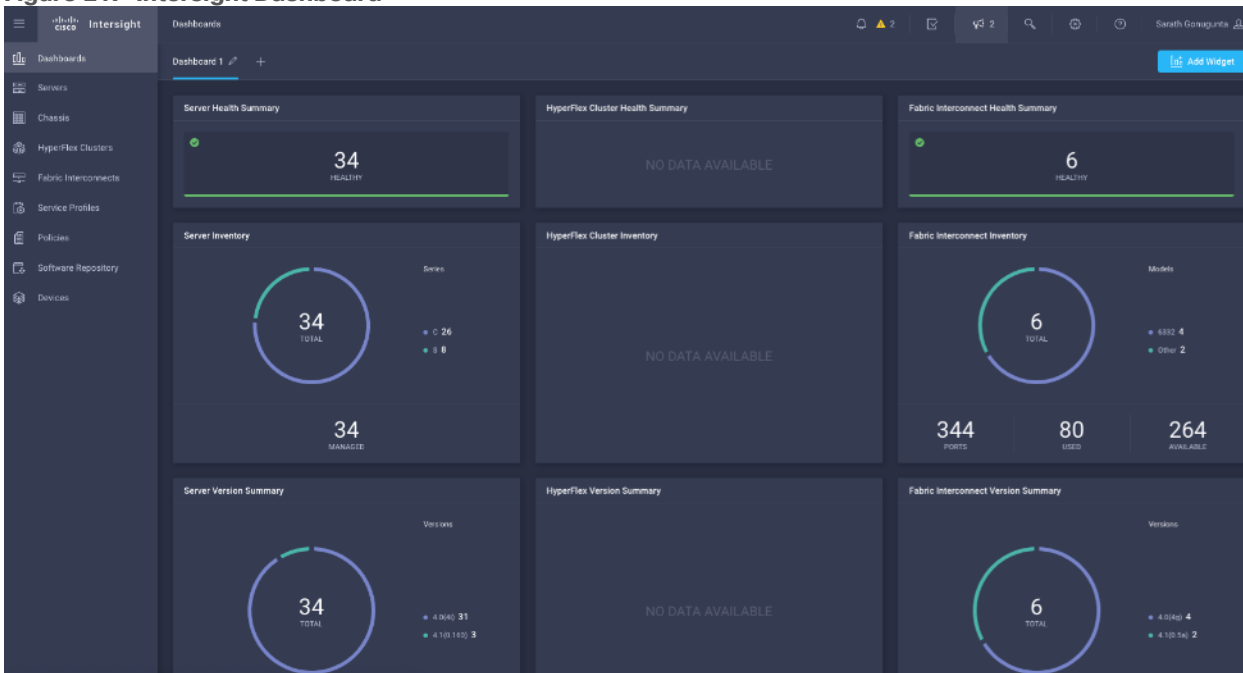
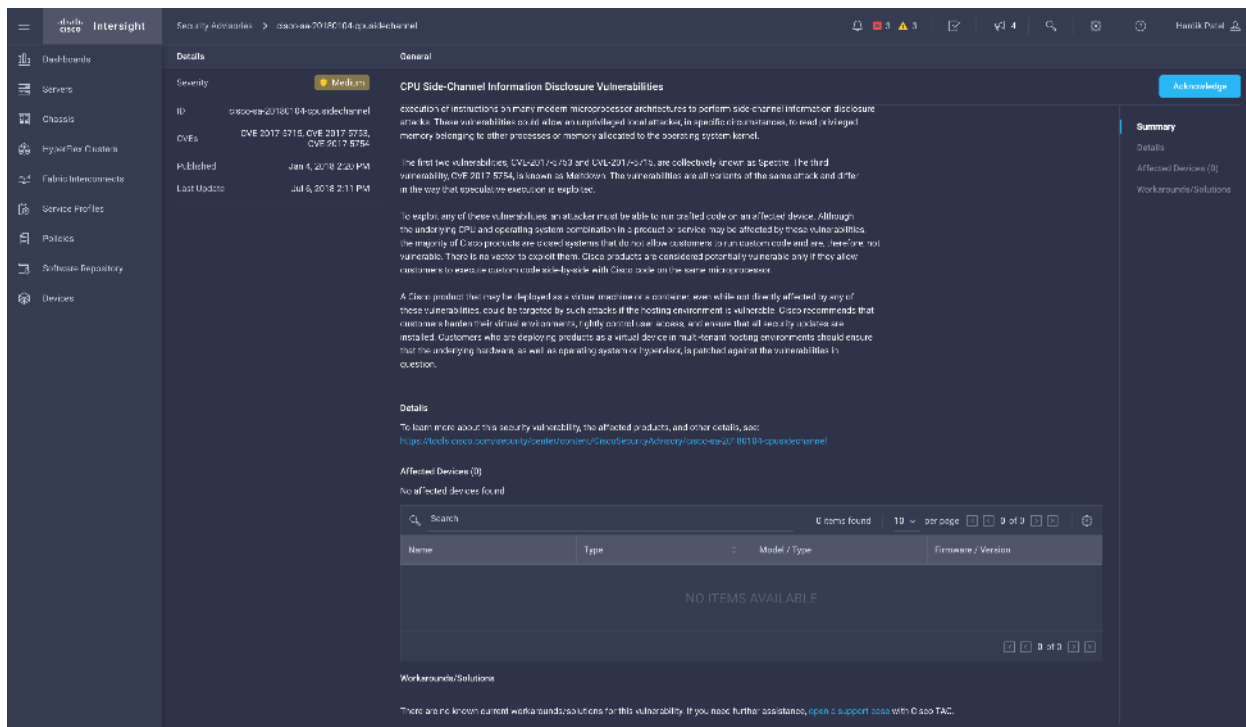
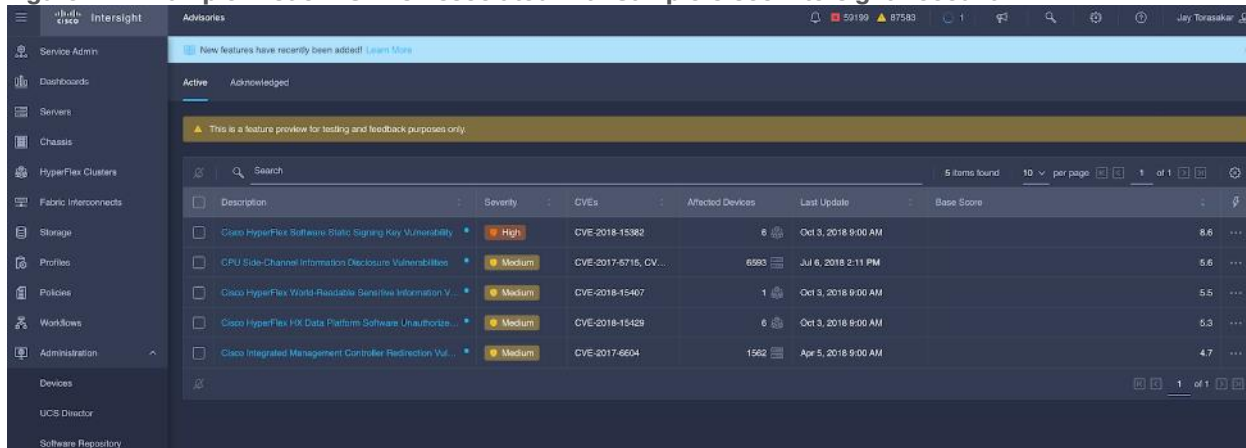


Figure 22. Example: List of PSIRTs Associated with Sample Cisco Intersight Account



Cloudera Data Platform (CDP)

Cloudera Data Platform Private Cloud (CDP PvC) is the on-premises version of Cloudera Data Platform. CDP Private Cloud delivers powerful analytic, transactional, and machine learning workloads in a hybrid data platform, combining the agility and flexibility of public cloud with the control of the data center. With a choice of traditional as well as elastic analytics and scalable object storage, CDP Private Cloud modernizes traditional monolithic cluster deployments into a powerful and efficient platform.

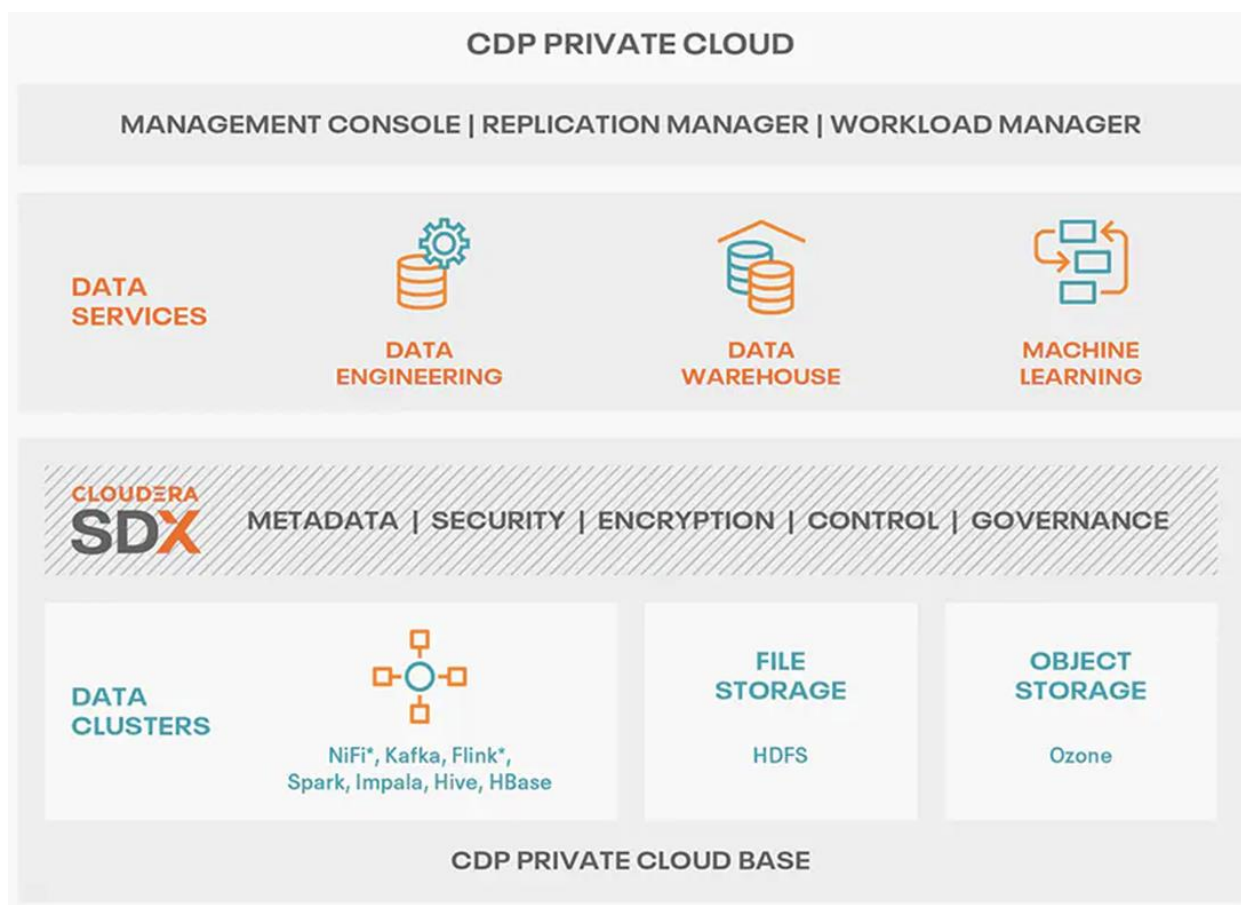
An integral part of CDP Hybrid Cloud, CDP Private Cloud provides the first step for data center customers toward true data and workload mobility, managed from a single pane of glass and with consistent data security and governance across all clouds, public and private.

With CDP Private Cloud, organizations benefit from:

- **Unified Distribution:** CDP offers rapid time to value through simplified provisioning of easy-to-use, self-service analytics enabling onboarding of new use cases at higher velocity.

- Hybrid & On-prem: Hybrid and multi-cloud experience, on-prem it offers best performance, cost, and security. It is designed for data centers with optimal infrastructure.
- Management: It provides consistent management and control points for deployments.
- Consistency: Security and governance policies can be configured once and applied across all data and workloads.
- Portability: Policies stickiness with data, even if it moves across all supported infrastructure.
- Improved cost efficiency with optimized resource utilization and the decoupling of compute and storage, lowering data center infrastructure costs up to 50%.
- Predictable performance thanks to workload isolation and perfectly managed multi-tenancy, eliminating the impact of spikes on critical workloads and resulting missed SLAs and SLOs.

Figure 23. Cloudera Data Platform Private Cloud



Cloudera Data Platform Private Cloud Base (CDP PvC Base)

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

CDP Private Cloud Base supports a variety of hybrid solutions where compute tasks are separated from data storage and where data can be accessed from remote clusters, including workloads created using CDP Private

Cloud Data Services. This hybrid approach provides a foundation for containerized applications by managing storage, table schema, authentication, authorization, and governance.

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

Cloudera Data Platform Private Cloud Data Services (CDP PvC DS)

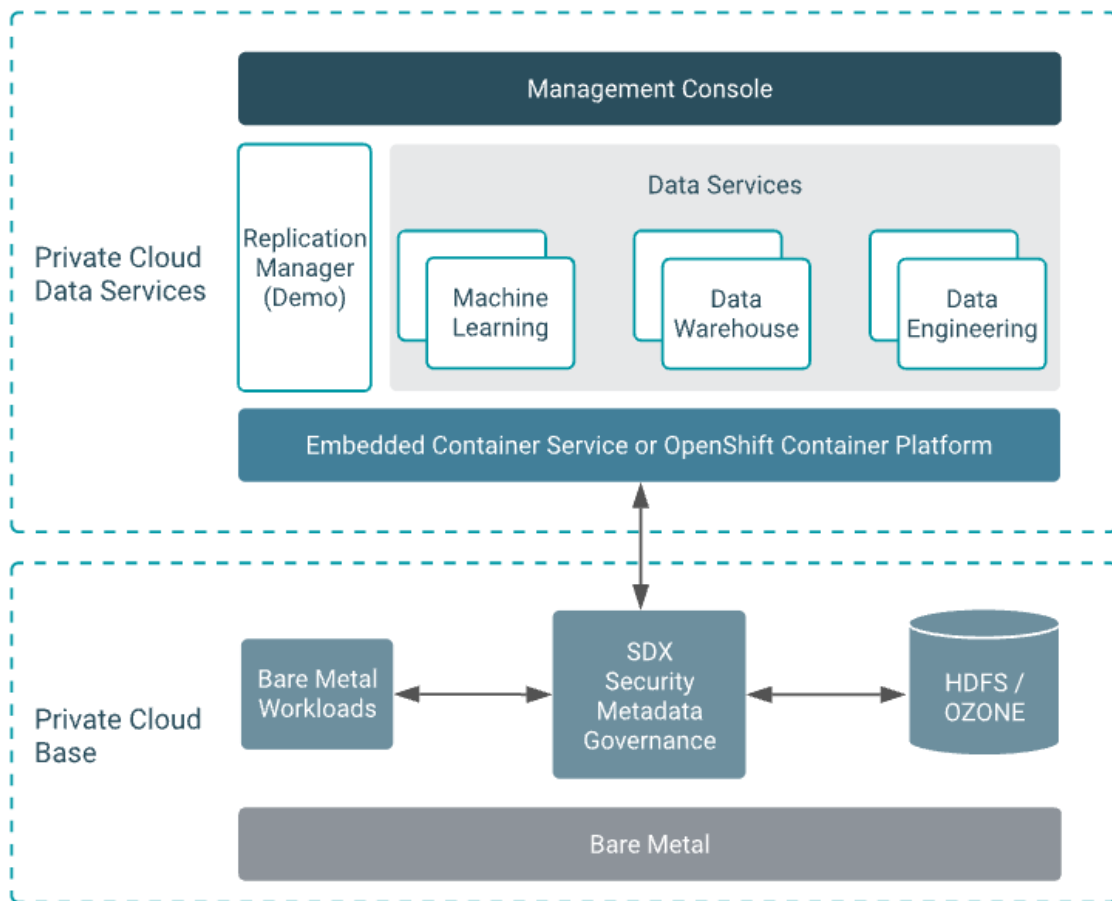
Cloudera Data Platform (CDP) Private Cloud ([Figure 24](#)) is the newest on-prem offering of CDP that brings many of the benefits of the public cloud deployments to the on-prem CDP deployments.

CDP Private Cloud provides a disaggregation of compute and storage and allows independent scaling of compute and storage clusters. Using containerized applications deployed on Kubernetes, CDP Private Cloud brings both agility and predictable performance to analytic applications. CDP Private Cloud gets unified security, governance, and metadata management through Cloudera Shared Data Experience (SDX), which is available on a CDP Private Cloud Base cluster.

CDP Private Cloud users can rapidly provision and deploy Cloudera Data Engineering (CDE), Cloudera Data Warehousing (CDW) and Cloudera Machine Learning (CML) services through the Management Console, and easily scale them up or down as required.

A CDP Private Cloud deployment requires you to have a Private Cloud Base cluster and a RedHat OpenShift Kubernetes cluster. The OpenShift cluster is set up on a Bare Metal deployment. The Private Cloud deployment process involves configuring the Management Console on the OpenShift cluster, registering an environment by providing details of the Data Lake configured on the Base cluster, and then creating the workloads.

Figure 24. Cloudera Data Platform Private Cloud Data Services (CDP PvC DS)



Cloudera Shared Data Experience (SDX)

SDX is a fundamental part of Cloudera Data Platform architecture, unlike other vendors’ bolt-on approaches to security and governance. Independent from compute and storage layers, SDX delivers an integrated set of security and governance technologies built on metadata and delivers persistent context across all analytics as well as public and private clouds. Consistent data context simplifies the delivery of data and analytics with a multi-tenant data access model that is defined once and seamlessly applied everywhere.

SDX reduces risk and operational costs by delivering consistent data context across deployments. IT can deploy fully secured and governed data lakes faster, giving more users access to more data, without compromise.

Key benefit and feature of SDX includes:

- **Insightful metadata** - Trusted, reusable data assets and efficient deployments need more than just technical and structural metadata. CDP’s Data Catalog provides a single pane of glass to administer and discover all data, profiled, and enhanced with rich metadata that includes the operational, social, and business context, and turns data into valuable information.
- **Powerful security** - Eliminate business and security risks and ensure compliance by preventing unauthorized access to sensitive or restricted data across the platform with full auditing. SDX enables organizations to establish multi-tenant data access with ease through standardization and seamless enforcement of granular, dynamic, role- and attribute-based security policies on all clouds and data centers.

- **Full encryption** - Enjoy ultimate protection as a fundamental part of your CDP installation. Clusters are deployed and automatically configured to use Kerberos and for encrypted network traffic with Auto-TLS. Data at rest, both on-premises and in the cloud, is protected with enterprise-grade cryptography, supporting best practice tried and tested configurations.
- **Hybrid control** - Meet the ever-changing business needs to balance performance, cost, and resilience. Deliver true infrastructure independence. SDX enables it all with the ability to move data, together with its context, as well as workloads between CDP deployments. Platform operational insight into aspects like workload performance deliver intelligent recommendations for optimal resource utilization.
- **Enterprise-grade governance** - Prove compliance and manage the complete data lifecycle from the edge to AI and from ingestion to purge with data management across all analytics and deployments. Identify and manage sensitive data, and effectively address regulatory requirements with unified, platform-wide operations, including data classification, lineage, and modeling.

CDP Private Cloud Management Console

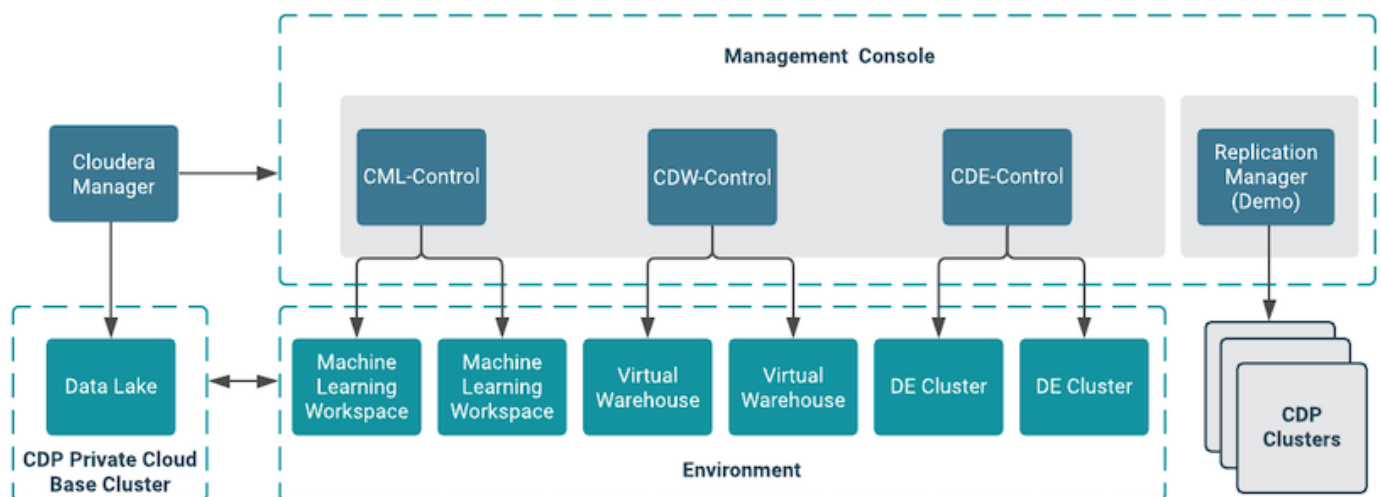
The Management Console is a service used by CDP administrators to manage environments, users, and services.

The Management Console allows you to:

- Enable user access to CDP Private Cloud Data Services, onboard and set up authentication for users, and determine access rights for the various users to the available resources.
- Register an environment, which represents the association of your user account with compute resources using which you can manage and provision workloads such as Data Warehouse and Machine Learning. When registering the environment, you must specify a Data Lake residing on the Private Cloud base cluster to provide security and governance for the workloads.
- View information about the resources consumed by the workloads for an environment.
- Collect diagnostic information from the services for troubleshooting purposes.

[Figure 25](#) shows a basic architectural overview of the CDP Private Cloud Management Console.

Figure 25. CDP Private Cloud Management Console



Apache Ozone

Apache Ozone is a scalable, redundant, and distributed object store for Hadoop. Apart from scaling to billions of objects of varying sizes, Ozone can function effectively in containerized environments such as Kubernetes and YARN. Applications using frameworks like Apache Spark, YARN, and Hive work natively without any modifications. Ozone is built on a highly available, replicated block storage layer called Hadoop Distributed Data Store (HDDS).

Ozone is a scale-out architecture with minimal operational overheads and long-term maintenance efforts. Ozone can be co-located with HDFS with single security and governance policies for easy data exchange or migration and also offers seamless application portability. Ozone enables separation of compute and storage via the S3 API as well as similar to HDFS, it also supports data locality for applications that choose to use it.

Apache Ozone is a scalable, redundant, and distributed object store for Hadoop. Apart from scaling to billions of objects of varying sizes, Ozone can function effectively in containerized environments such as Kubernetes and YARN. Applications using frameworks like Apache Spark, YARN, and Hive work natively without any modifications. Apache Ozone is built on a highly available, replicated block storage layer called Hadoop Distributed Data Store (HDDS).

Apache Ozone consists of volumes, buckets, and keys:

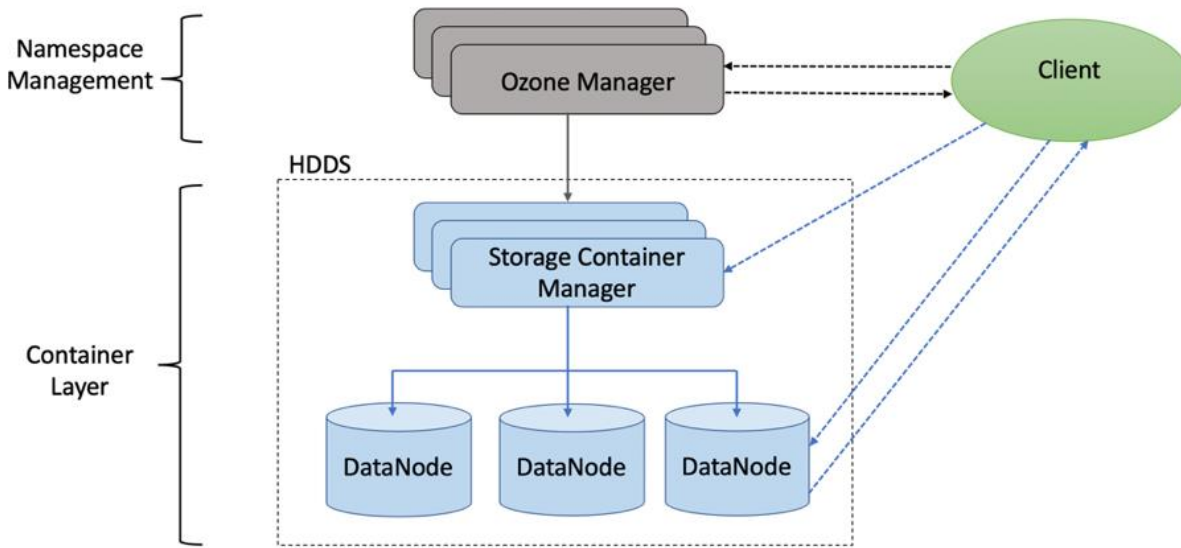
- Volumes are similar to user accounts. Only administrators can create or delete volumes.
- Buckets are similar to directories. A bucket can contain any number of keys, but buckets cannot contain other buckets.
- Keys are similar to files. Each key is part of a bucket, which, in turn, belongs to a volume. Ozone stores data as keys inside these buckets.

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

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

Apache Ozone is a distributed key-value store that can manage both small and large files alike. While HDFS provides POSIX-like semantics, Apache Ozone looks and behaves like an Object Store.

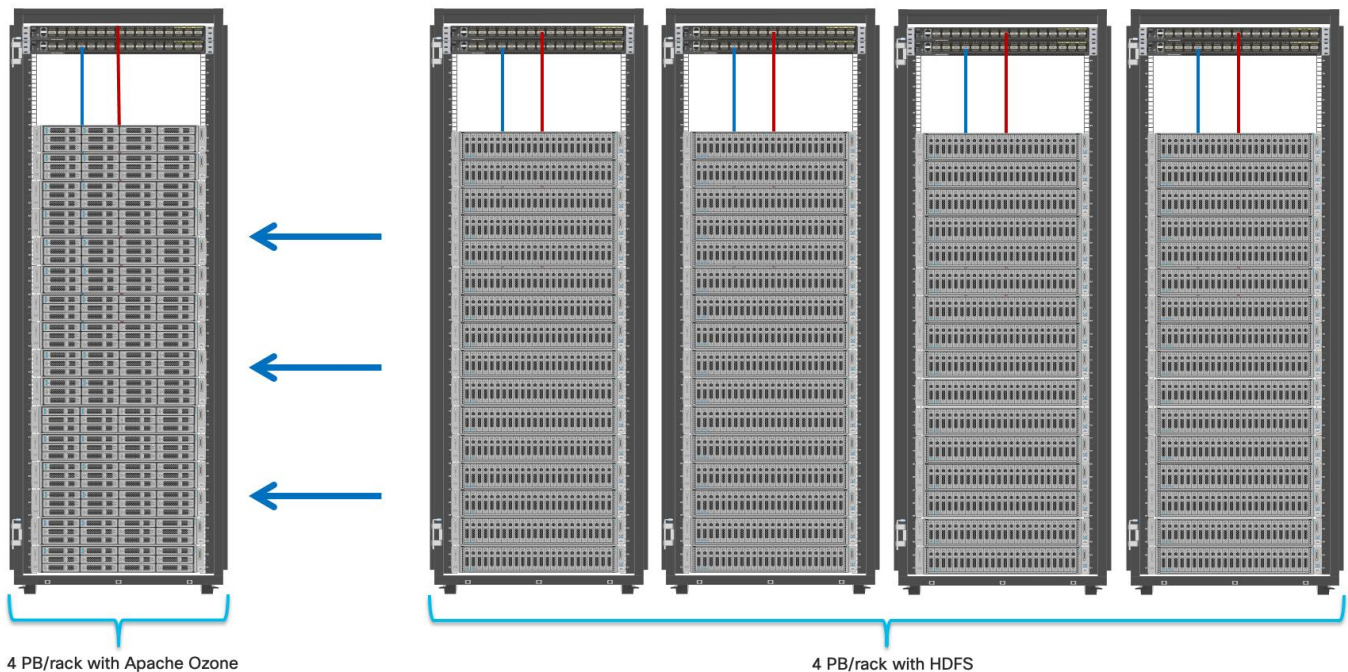
Figure 26. Basic Architecture for Apache Ozone



Apache Ozone has the following cost savings and benefits due to storage consolidation:

- Lower Infrastructure cost
- Lower software licensing and support cost
- Lower lab footprint
- Newer additional use cases with support for HDFS and S3 and billions of objects supporting both large and small files in a similar fashion.

Figure 27. Data Lake Consolidation with Apache Ozone



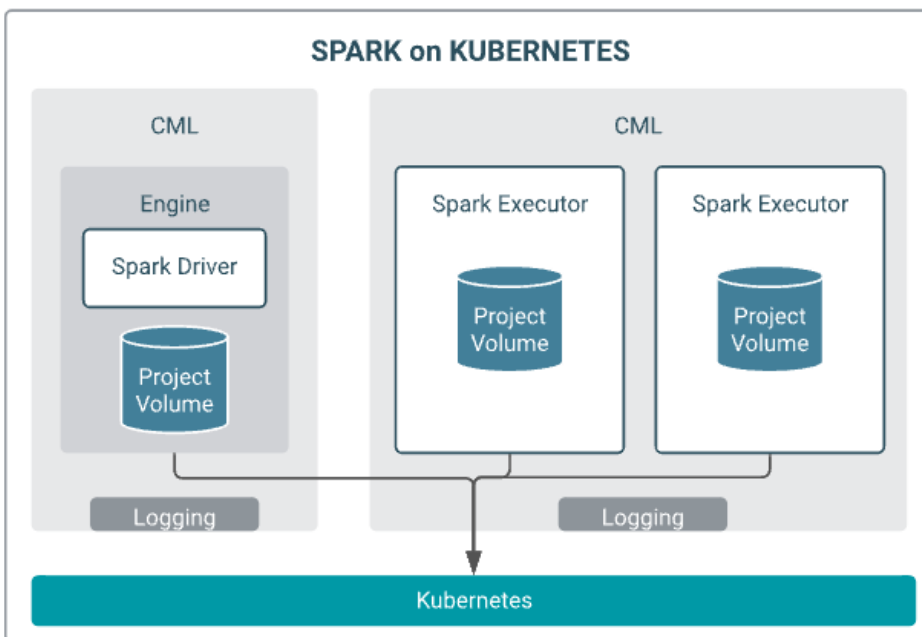
For more information about Apache Ozone, go to: <https://blog.cloudera.com/apache-ozone-and-dense-data-nodes/>

Apache Spark 3.0

Apache Spark 3.0 delivered many new capabilities, performance gains, and extended compatibility for the Spark ecosystem such as accelerator-aware scheduling, adaptive query execution, dynamic partition pruning, join hints, new query explain, better ANSI compliance, observable metrics, new UI for structured streaming, new UDAF and built-in functions, new unified interface for Pandas UDF, and various enhancements in the built-in data sources.

Spark is no longer limited just to CPU for its workload, it now offers GPU isolation and pooling GPUs from different servers to accelerated compute. To easily manage the deep learning environment, YARN launches the Spark 3.0 applications with GPU. Spark 3.0 introduces new shuffle service for Spark on Kubernetes that will allow dynamic scale up and down. Spark 3.0 also supports GPU support with pod level isolation for executors which makes scheduling more flexible on a cluster with GPUs. This prepares the other workloads, such as Machine Learning and ETL, to be accelerated by GPU for Spark Workloads. [Cisco Blog on Apache Spark 3.0](#)

Figure 28. Spark on Kubernetes



Solution Design

This chapter contains the following:

- [Requirements](#)
- [Solution Prerequisites](#)
- [Cloudera Data Platform Private Cloud Base Requirements](#)

This CVD explains the architecture and deployment procedures for Cloudera Data Platform Private Cloud on a 11-node cluster using Cisco UCS Integrated Infrastructure for Big Data and Analytics. The solution provides the details to configure CDP PvC on the bare metal RHEL infrastructure.

This CVD was designed with the following:

- Cisco Intersight managed Cisco UCS C240 M6 Rack Server with NVIDIA A100 GPU Installed per node
- Cloudera Data Private Cloud Base 7.1.8
- CDS 3.3 powered by Apache Spark
- NVIDIA RAPIDS libraries for accelerated data science

Requirements

Physical Components

[Table 3](#) lists the required physical components and hardware.

Table 3. CDIP with CDP PvC Base with CDS 3.3 System hardware Components

| Component | Hardware |
|----------------------|------------------------------------------|
| Fabric Interconnects | 2 x Cisco UCS 64108 Fabric Interconnects |
| Servers | 11 x Cisco UCS C240 M6 Rack Server |

Software Components

[Table 4](#) lists the software components and the versions required for a single cluster of the Cohesity Helios Platform running in Cisco UCS, as tested, and validated in this document.

Table 4. Software Distributions and Firmware Versions

| Layer | Component | Version or Release |
|----------|-------------------------------------------|--------------------|
| Compute | Cisco UCS C240 M6 Rack Server | 4.2.2f |
| Network | Cisco UCS Fabric Interconnect 64108 | 4.2.2c |
| | Cisco UCS VIC 1467 | 5.2(2b) |
| | Cloudera Data Platform Private Cloud Base | 7.1.8 |
| Software | Cloudera Manager | 7.7.3 |
| | CDS | 3.3.7180 |
| | Postgres | 14.5 |

| Layer | Component | Version or Release |
|-------|----------------------------------------------------------|--------------------|
| | Hadoop (Includes YARN and HDFS) | 3.1.1.7.1.8.0-801 |
| | Spark | 2.4.8.7.1.8.0-801 |
| | Red Hat Enterprise Linux Server (CDP Private Cloud Base) | 8.6 |

Note: The Cisco latest drivers can be downloaded here: <https://software.cisco.com/download/home>.

Note: Please check the CDP PvC requirements and supported versions for information about hardware, operating system, and database requirements, as well as product compatibility matrices, here: <https://supportmatrix.cloudera.com/> and here: <https://docs.cloudera.com/cdp-private-cloud-upgrade/latest/release-guide/topics/cdpdc-requirements-supported-versions.html>

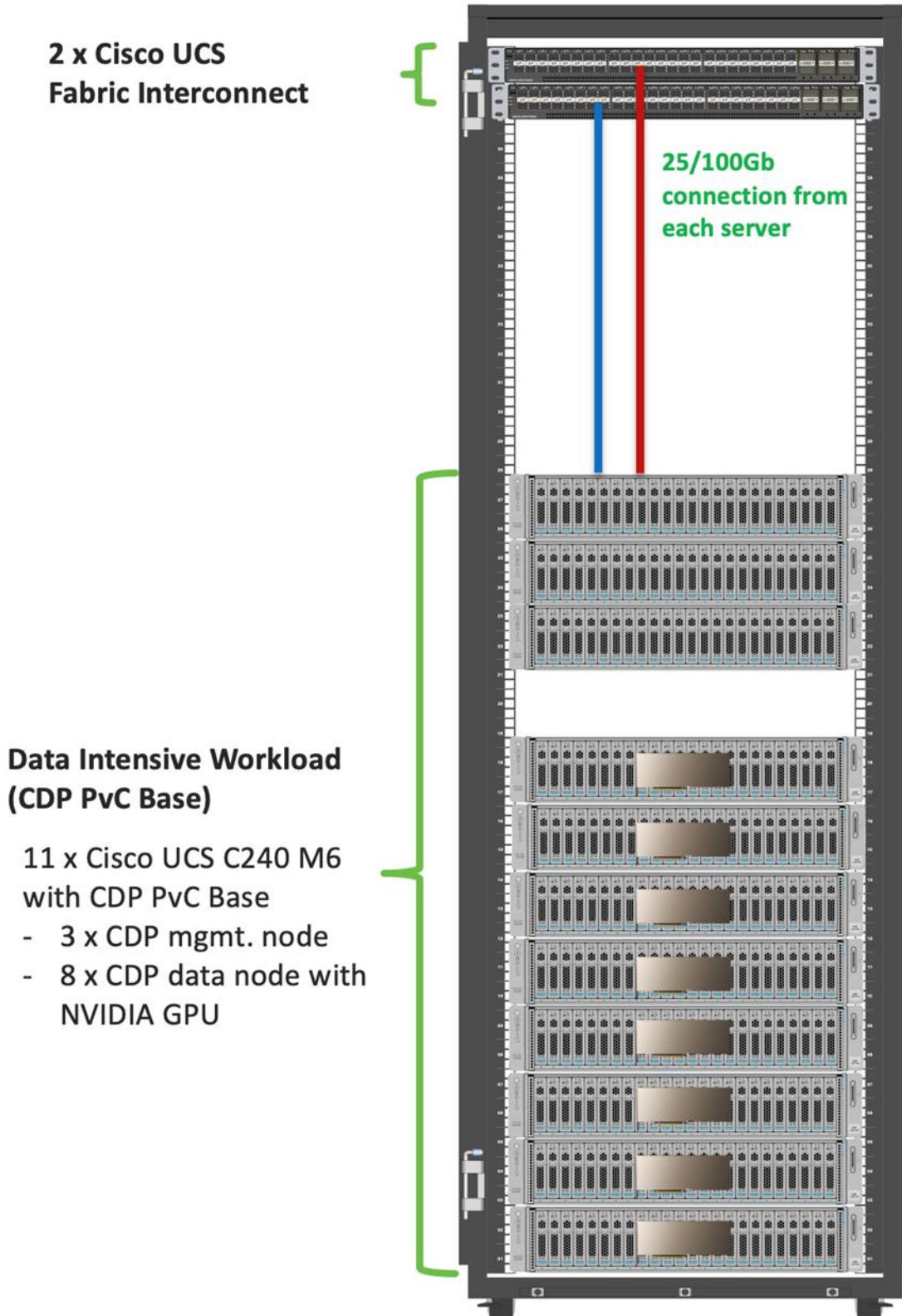
Note: For Cloudera Private Cloud Base and Experiences versions and supported features, go to: <https://docs.cloudera.com/cdp-private-cloud-base/7.1.8/runtime-release-notes/topics/rt-pvc-runtime-component-versions.html>

Note: For Cloudera Private Cloud Base requirements and supported version, go to: <https://docs.cloudera.com/cdp-private-cloud-base/7.1.8/installation/topics/cdpdc-requirements-supported-versions.html>

Physical Topology

Single rack consists of two vertical PDUs and two Cisco UCS Fabric Interconnect with 11 x Cisco UCS C240 M6 Rack Servers connected to each of the vertical PDUs for power redundancy. This ensures availability during power source failure. [Figure 29](#) illustrates four 25 Gigabit Ethernet link from each server connected to both Fabric Interconnects. (Port 0-1 connected to FI - A and port 2-3 connected to FI - B).

Figure 29. Cisco Data Intelligence Platform with Cloudera Data Platform Private Cloud Base



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

Note: Intel Virtual RAID on CPU (Intel VROC) configured RAID 1 for NVMe drives to provide business continuity for ozone metadata in case of hardware failure. For more details, see Intel Virtual RAID on CPU (Intel VROC) section in <https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/c240m6-sff-specsheet.pdf>.

Note: NVMe drives are configured to store ozone metadata and ozone mgmt. configuration for the master/mgmt. nodes and storage/data nodes.

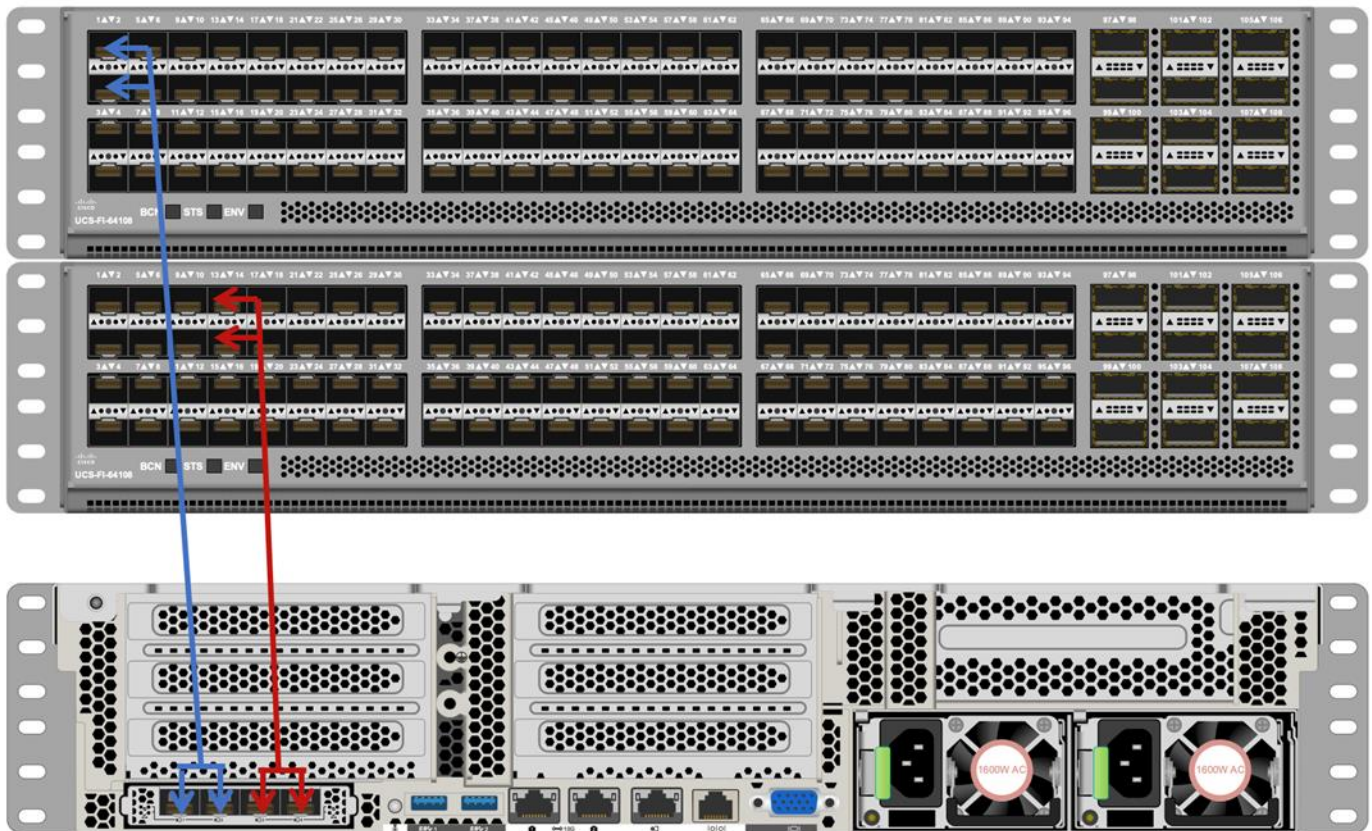
Note: The "hybrid" mixed compute Data Nodes use NVMe for both ozone metadata and shuffle (spark, mr, tez) + caching (llap). They should mount ozone partitions across both drives as RAID1 (800GB), with the remaining space used for shuffle/cache as independent JBOD partitions.

Note: Minimum starter configuration is 3 master nodes and 8 Data Nodes. This will support erasure coding rs(6,3) in the future. Additional Data Nodes can be added in increments of 1 to increase storage.

Logical Topology

Figure 30 shows the logical topology

Figure 30. Logical Topology



- Cisco UCS 64108 Fabric Interconnects provide network connectivity.
- The Cisco UCS C240 M6 rack server connects to fabric interconnects using Cisco UCS VIC 1467, where two or four 25 Gigabit Ethernet ports can connect to the appropriate FI.

Solution Prerequisites

There are many platform dependencies to enable Cloudera Data Platform Private Cloud Data Services running on RedHat OpenShift Container Platform. The containers need to access data stored on HDFS in Cloudera Data Platform Private Cloud Base in a fully secure manner.

The following are the prerequisites needed to enable this solution:

- Network requirements
- Security requirements
- Operating System requirements
- Cloudera requirements

Network Requirements

Cloudera Base cluster that houses HDFS storage and Cloudera Private Cloud compute-only clusters should be reachable with no more than a 3:1 oversubscription to be able to read from and write to the base HDFS cluster. The recommended network architecture is Spine-Leaf between the spine and leaf switches. Additional routing hops should be avoided in production and ideally both HDFS/Ozone storage and Cloudera Private Cloud Data Services are on the same network.

For more information, go to: <https://docs.cloudera.com/cdp-private-cloud-upgrade/latest/release-guide/topics/cdpdc-networking-security-requirements.html>

Cloudera Data Platform Private Cloud Requirements

NTP

Both CDP PvC Base and CDP PvC DS cluster should have their time synched with the NTP Clock time from same the NTP source. Also make sure, Active Directory server where Kerberos is setup for data lake and for other services must also be synched with same NTP source.

JDK 11

The cluster must be configured with JDK 11, JDK8 is not supported. You can use Oracle, OpenJDK 11.04, or higher. JAVA 11 is a JKS requirement and must be met. In this CVD we used Oracle JDK 11.0.9.

Kerberos

Kerberos must be configured using an Active Directory (AD) or MIT KDC. The Kerberos Key Distribution Center (KDC) will use the domain's Active Directory service database as its account database. An Active Directory server is recommended for default Kerberos implementations and will be used in the validation of this solution. Kerberos will be enabled for all services in the cluster.

Note: Red Hat IPA/Identity Management is currently not supported.

Database Requirements

Cloudera Manager and Runtime come packaged with an embedded PostgreSQL database for use in non-production environments. The embedded PostgreSQL database is not supported in production environments. For production environments, you must configure your cluster to use dedicated external databases.

For detailed information about supported database go to: <https://supportmatrix.cloudera.com/>

Configure Cloudera Manager with TLS/SSL

TLS/SSL provides privacy and data integrity between applications communicating over a network by encrypting the packets transmitted between endpoints (ports on a host, for example). Configuring TLS/SSL for any system typically involves creating a private key and public key for use by server and client processes to negotiate an encrypted connection at runtime. In addition, TLS/SSL can use certificates to verify the trustworthiness of keys presented during the negotiation to prevent spoofing and mitigate other potential security issues.

Setting up Cloudera clusters to use TLS/SSL requires creating private key, public key, and storing these securely in a keystore, among other tasks. Although adding a certificate to the keystore may be the last task in the process, the lead time required to obtain a certificate depends on the type of certificate you plan to use for the cluster.

For detailed information on encrypting data in transit, go to: <https://docs.cloudera.com/cdp-private-cloud-base/7.1.7/security-encrypting-data-in-transit/topics/cm-security-guide-ssl-certs.html>

The Auto-TLS feature automates all the steps required to enable TLS encryption at a cluster level. Using Auto-TLS, you can let Cloudera manage the Certificate Authority (CA) for all the certificates in the cluster or use the company's existing CA. In most cases, all the necessary steps can be enabled easily via the Cloudera Manager UI. This feature automates the following processes when Cloudera Manager is used as a Certificate Authority:

- Creates the root Certificate Authority or a Certificate Signing Request (CSR) for creating an intermediate Certificate Authority to be signed by company's existing Certificate Authority (CA)
- Generates the CSRs for hosts and signs them

Configuring TLS Encryption for Cloudera Manager Using Auto-TLS for detailed information:

<https://docs.cloudera.com/cdp-private-cloud-base/7.1.7/security-encrypting-data-in-transit/topics/cm-security-how-to-configure-cm-tls.html>

Manually Configuring TLS Encryption for Cloudera Manager for detailed information:

<https://docs.cloudera.com/cdp-private-cloud-base/7.1.7/security-encrypting-data-in-transit/topics/cm-security-how-to-configure-cm-tls.html>

TLS uses JKS-format (Java KeyStore)

Cloudera Manager Server, Cloudera Management Service, and many other CDP services use JKS formatted key-stores and certificates. Java 11 is required for JKS.

Licensing Requirements

The cluster must be setup with a license with entitlements for installing Cloudera Private Cloud. 60 days evaluation license for Cloudera Data Platform Private Cloud Base does not allow you to set up CDP Private Cloud Data Services.

Cisco UCS Install and Configure

This chapter contains the following:

- [Install Cisco UCS](#)

This section details the Cisco Intersight deployed Cisco UCS C240 M6 rack server connected to Cisco UCS Fabric Interconnect 64108 as part of the infrastructure build out. The racking, power, and installation of the Cisco UCS Rack Server for Cloudera Private Cloud Base can be found at [Cisco Data Intelligence Platform design zone](#) page. For detailed installation information, refer to the [Cisco Intersight Managed Mode Configuration Guide](#).

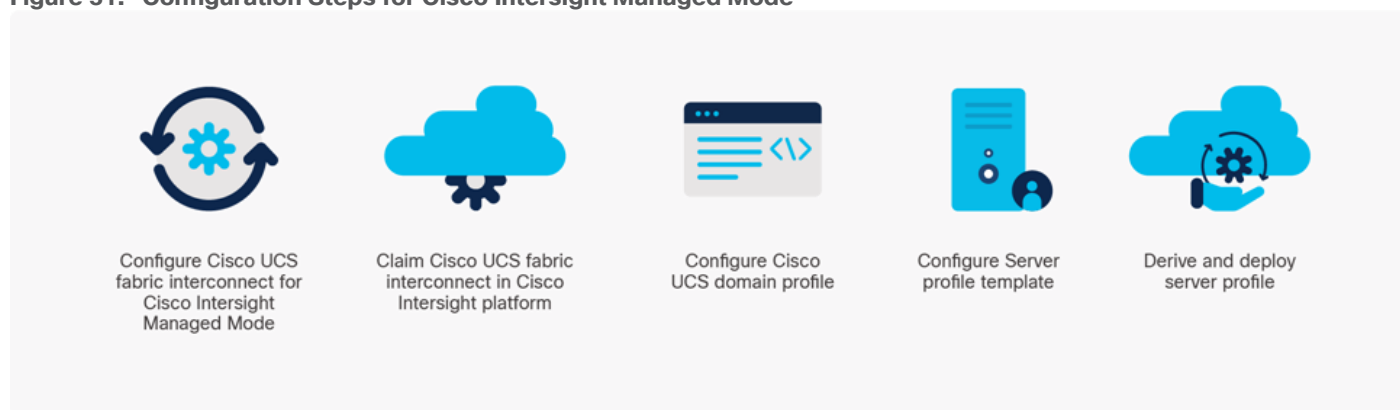
Install Cisco UCS

This subject contains the following procedures:

- [Claim a Cisco UCS Fabric Interconnect in the Cisco Intersight Platform](#)
- [Configure Cisco Intersight Pools and Policies](#)
- [Cisco Intersight Storage Policy Creation](#)

Cisco Intersight Managed Mode standardizes policy and operation management for Cisco UCS X-Series. The compute nodes in Cisco UCS X-Series are configured using server profiles defined in Cisco Intersight. These server profiles derive all the server characteristics from various policies and templates. At a high level, configuring Cisco UCS using Intersight Managed Mode consists of the steps shown in [Figure 31](#).

Figure 31. Configuration Steps for Cisco Intersight Managed Mode



During the initial configuration, for the management mode the configuration wizard enables customers to choose whether to manage the fabric interconnect through Cisco UCS Manager or the Cisco Intersight platform.

Procedure 1. Cisco UCS Fabric Interconnect Configuration in the Cisco Intersight Managed Mode

Step 1. Enter the Express setup IP address from Fabric Interconnect serial console in to the web browser.

Figure 32. Cisco UCS Fabric Interconnect Express Setup



Step 2. [Figure 33](#) shows the dialog during initial configuration of Cisco UCS FIs for setting up IMM. Select Intersight. Click Submit.

Figure 33. Cisco UCS Fabric Interconnect Initial Setup

Fabric Interconnect Initial Setup



Step 3. Enter details for Fabric setup as shown in [Figure 34](#).

Figure 34. Cisco UCS Fabric Interconnect Setup

Fabric Interconnect Initial Setup - Intersight Managed Mode

Basic Settings

Fabric Setup: Fabric A Fabric B

IPv4
 IPv6

System setup

Enforce strong password?: Yes No

System name:

Admin Password: Confirm Admin password:

Mgmt IP Address: . . .

Mgmt IP Netmask: . . .

Default Gateway: . . .

DNS Server IP: . . .

Domain Name :

After successful configuration; there will be message as shown in [Figure 35](#).

Figure 35. Successful completion of Cisco UCS Fabric Interconnect Setup for FI - A

Fabric Interconnect configuration is submitted successfully.
The configuration process is going to take upto 5 minutes. Monitor the console output to track the progress of configuration.
Once the configuration is complete, this page will be redirected to the configured IP address.
Please proceed further after 5 minutes.

Step 4. For seconds Fabric Interconnect; since Fabric Interconnect A is already configured successfully. Select Enable Clustering and enter password for FI - A as shown in [Figure 36](#).

Figure 36. Cisco UCS Fabric Interconnect Setup for FI - B
Fabric Interconnect Initial Setup

Basic Settings

Installer has detected the presence of a peer Fabric Interconnect. This Fabric Interconnect will be added to the cluster. If this is correct, Please provide admin password of the other Fabric Interconnect.

Cluster and Fabric setup

Enable clustering
 Standalone mode

Switch Fabric: Fabric A Fabric B

System setup

Admin Password of Peer:

Step 5. Enter mgmt. IP address for FI - B.

Figure 37. Cisco UCS Fabric Interconnect Setup for FI - B
Fabric Interconnect Initial Setup - Intersight Managed Mode

Basic Settings

System setup

Peer FI is IPv4 Cluster enabled. Please Provide Local Fabric Interconnect Mgmt IP Address

Mgmt IP Address: 10 . 4 . 1 . 10

Submit Reset

After successful configuration; prompt will display message as shown in [Figure 38](#).

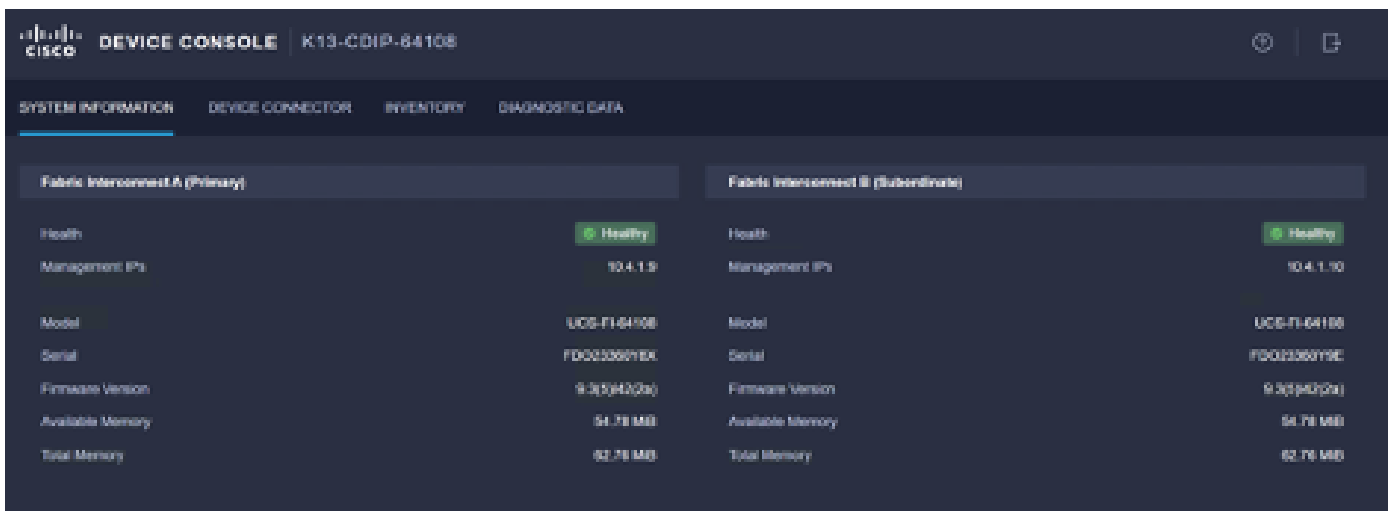
Figure 38. Cisco UCS Fabric Interconnect Setup for FI - B

Fabric Interconnect configuration is submitted successfully.
The configuration process is going to take upto 5 minutes. Monitor the console output to track the progress of configuration.
Once the configuration is complete, this page will redirected to the configured IP address.
Please proceed further after 5 minutes.

Step 6. Login to FI - A via entering <https://<fi-a>> in the web browser. Enter username and password.



Step 7. Review system information tab.



Step 8. Go to Device Connector and copy Device ID and Claim Code.



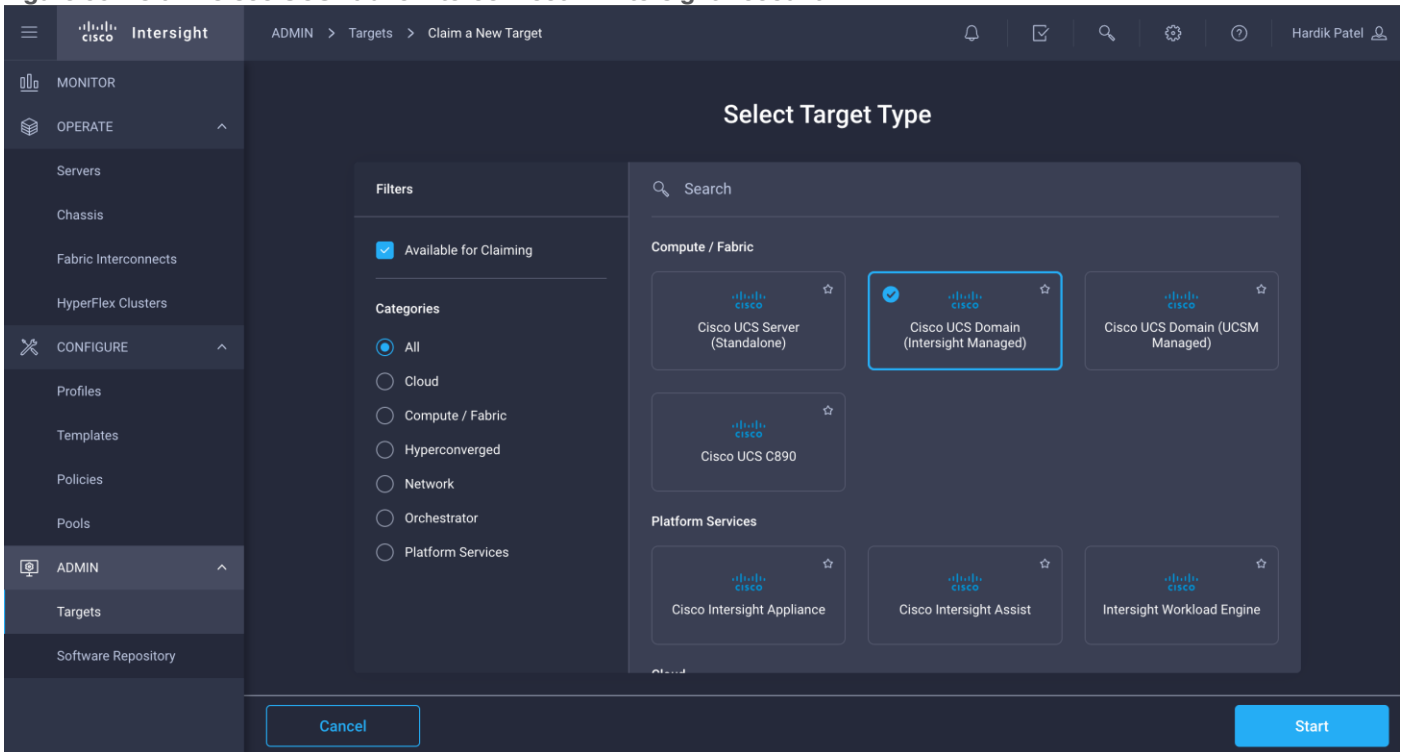
Procedure 2. Claim a Cisco UCS Fabric Interconnect in the Cisco Intersight Platform

Note: After setting up the Cisco UCS fabric interconnect for Cisco Intersight Managed Mode, FIs can be claimed to a new or an existing Cisco Intersight account. When a Cisco UCS fabric interconnect is successfully added to the Cisco Intersight platform, all subsequent configuration steps are completed in the Cisco Intersight portal.

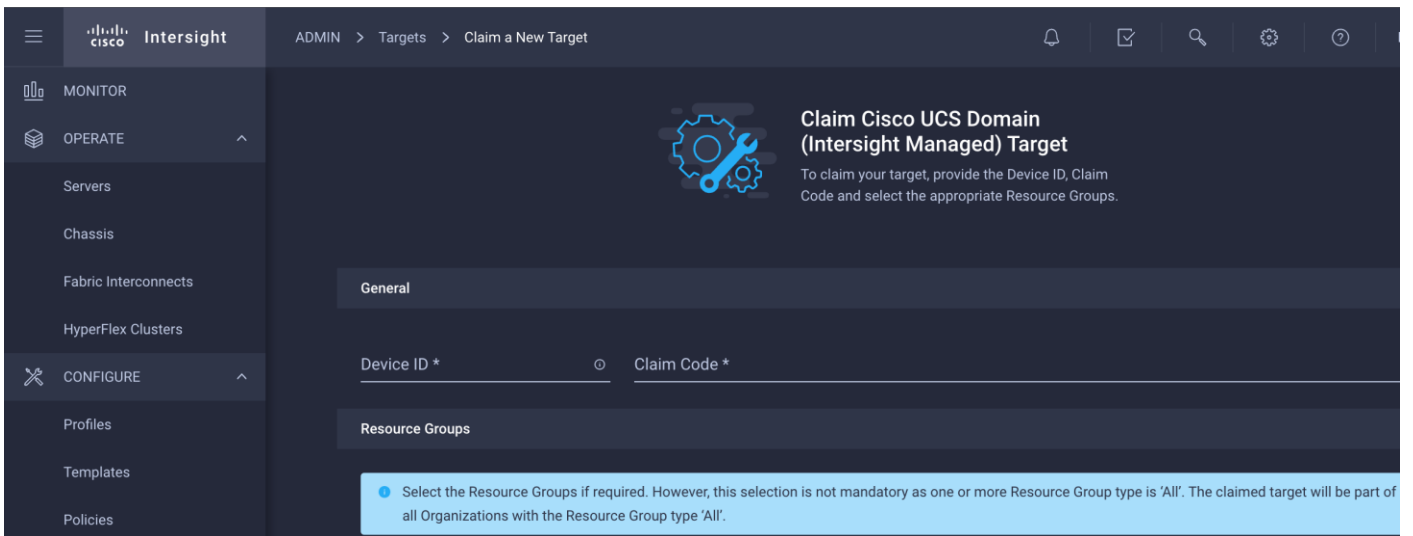
Step 1. To claim FI in IMM node, go to Targets > Claim a New Target.

Step 2. Select Cisco UCS Domain (Intersight Managed).

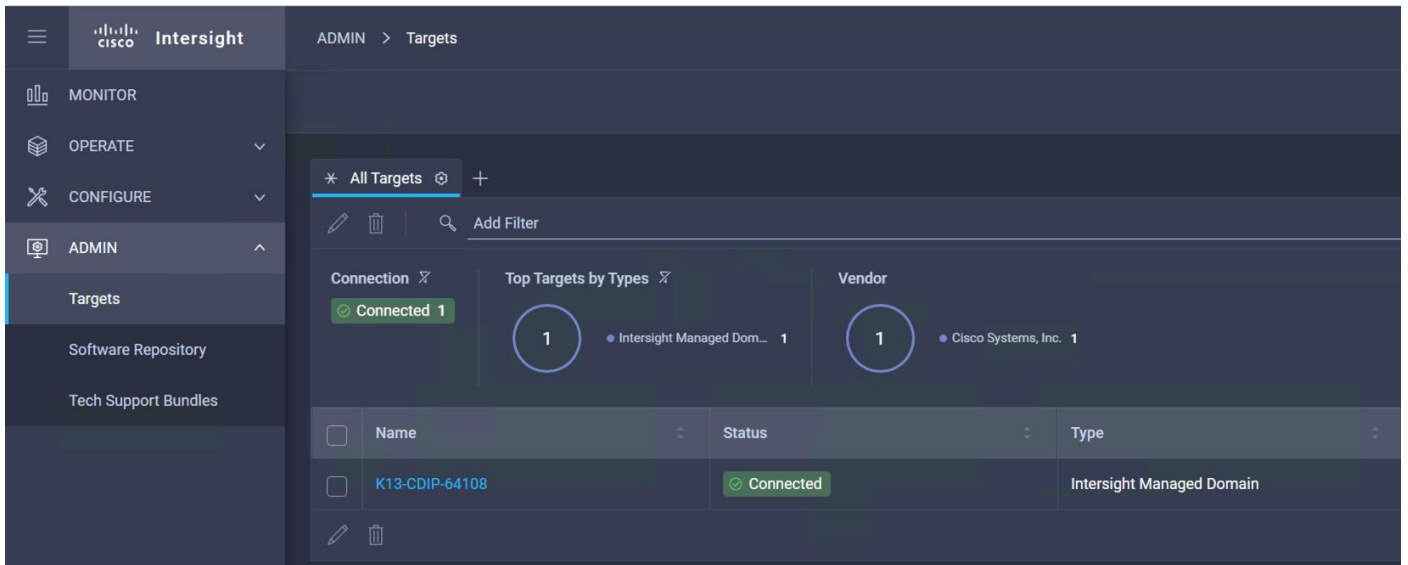
Figure 39. Claim Cisco UCS Fabric Interconnect in Intersight Account



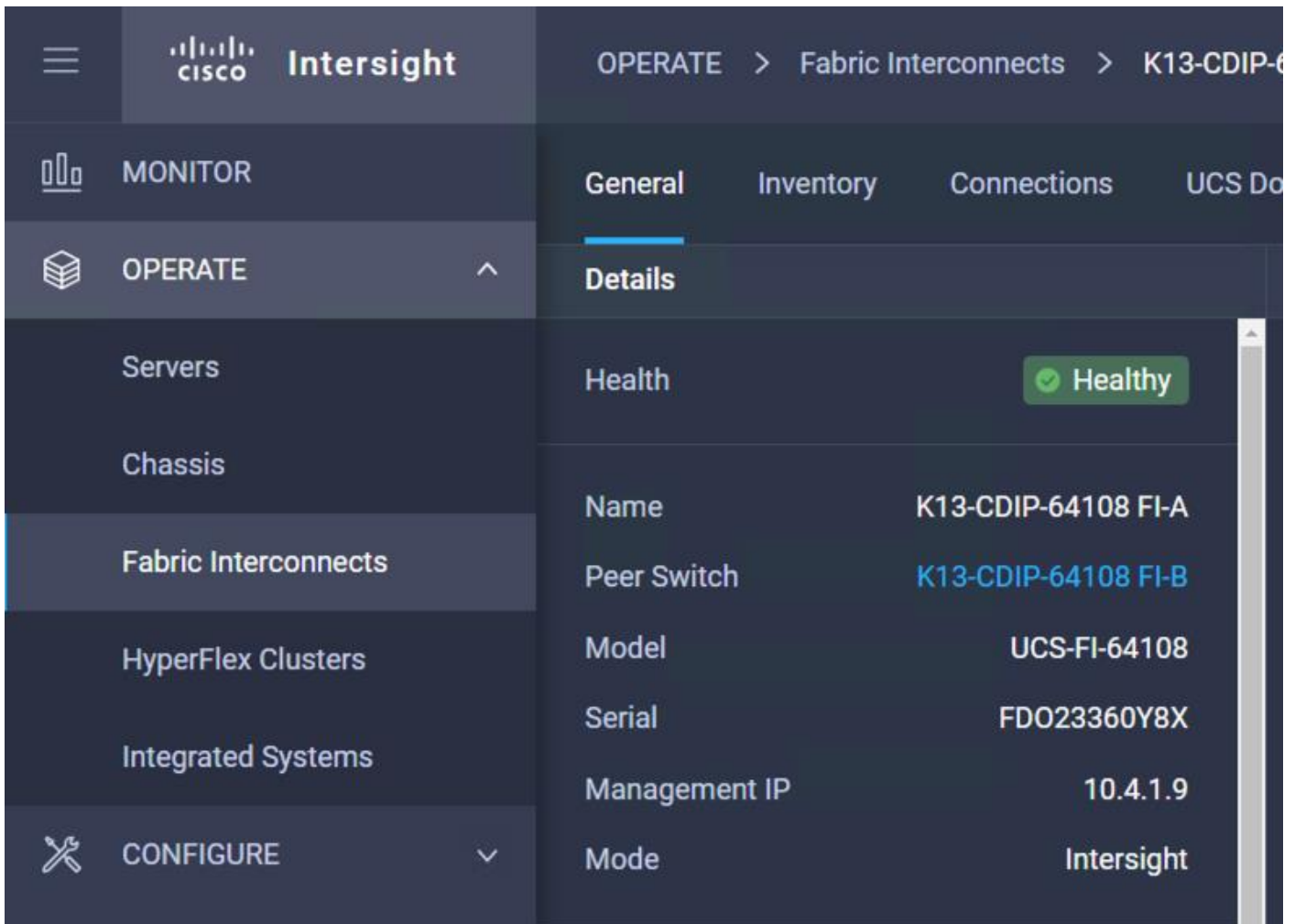
Step 3. Enter Device ID and Claim Code from one of the FI to be claimed. Click Claim.



Step 4. Review the newly claimed Cisco UCS Domain.



Step 5. Cisco UCS fabric interconnect in OPERATE tab shows details and Management Mode as shown below:

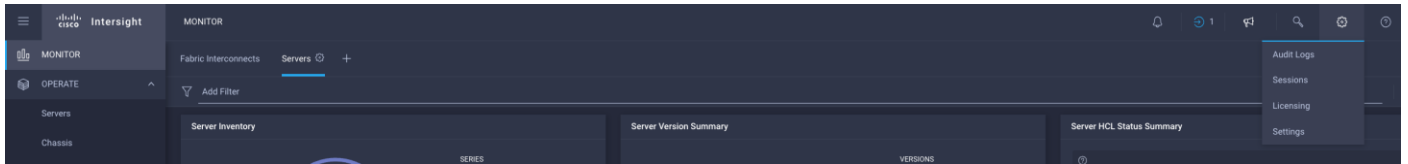


Step 6. Cisco UCS fabric interconnect Device Console WebUI > Device Connector tab shows claimed account name as shown below:

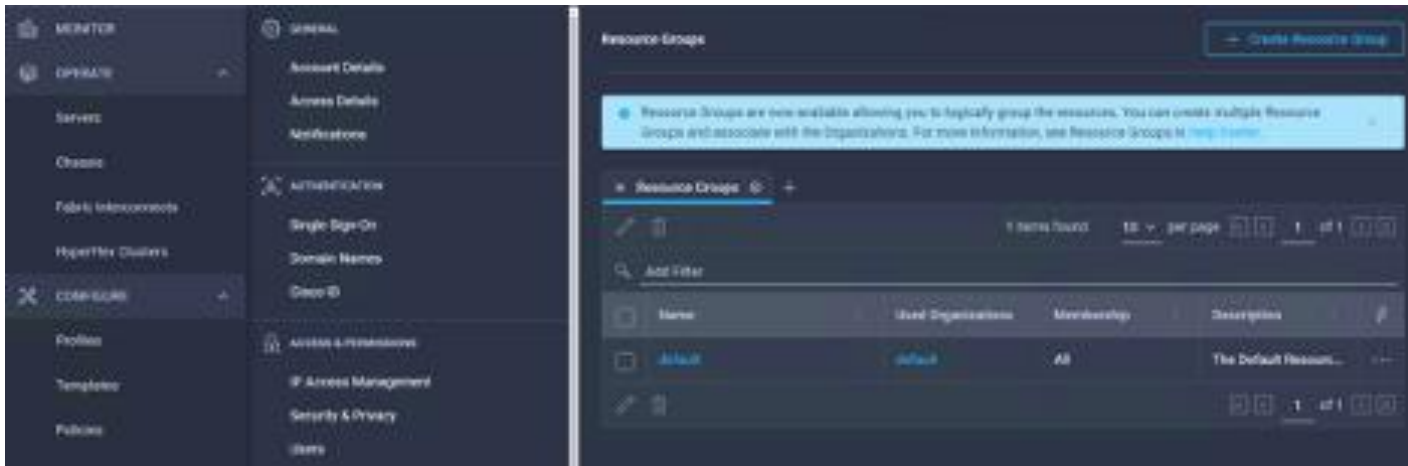


Procedure 3. Configure Cisco Intersight Account settings

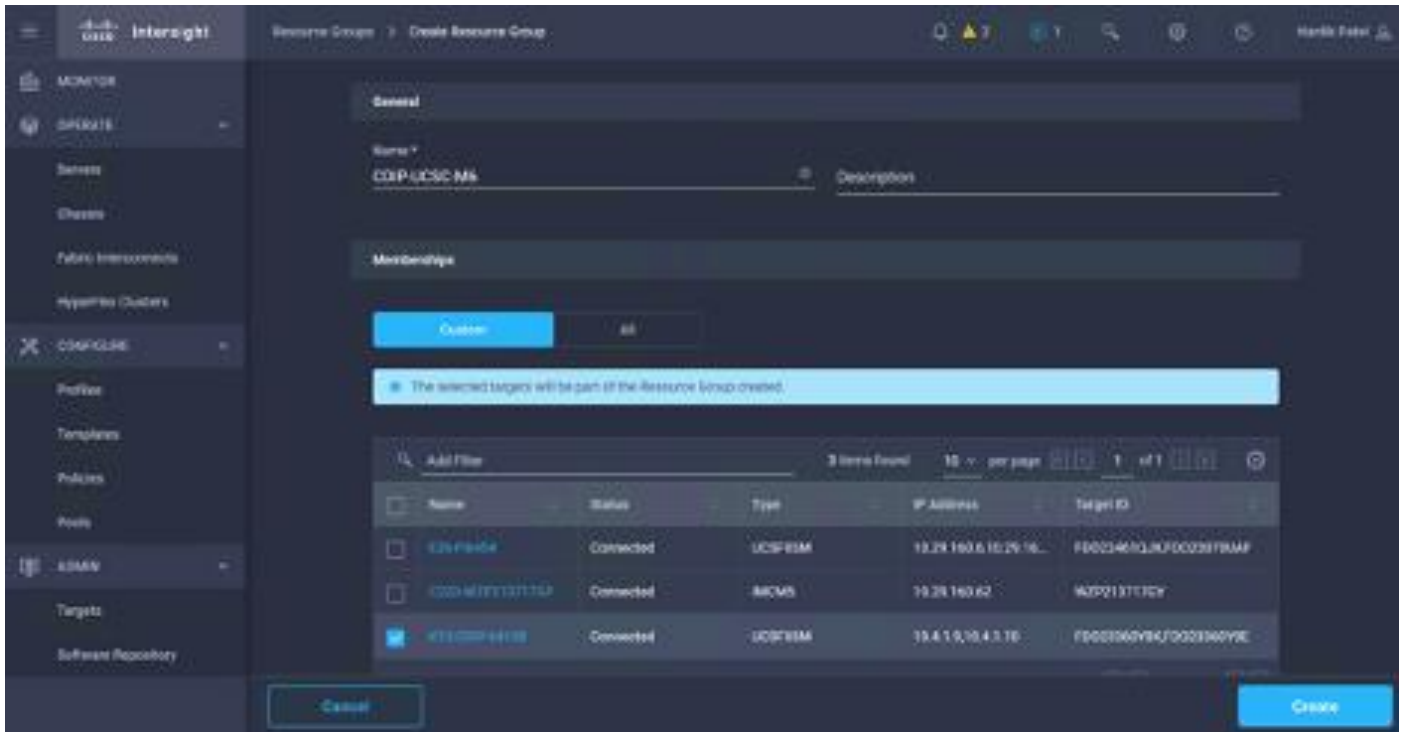
Step 1. To configure or display account specific parameters or edit license subscription; click on the gear icon on top right corner of Intersight Web console. For more details: https://intersight.com/help/saas/features/cisco_intersight/settings



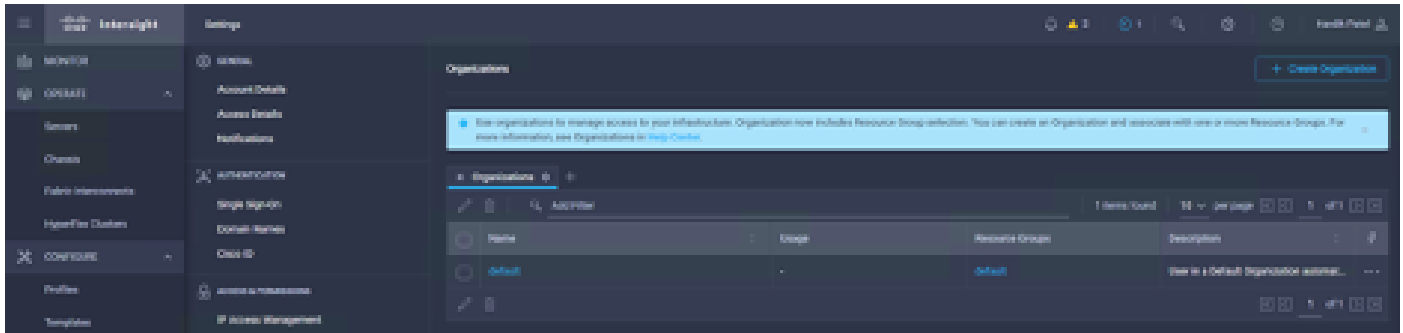
Step 2. In access & Permissions section, select Resource Group. Create New resource group, for new Cisco UCS FI domain.



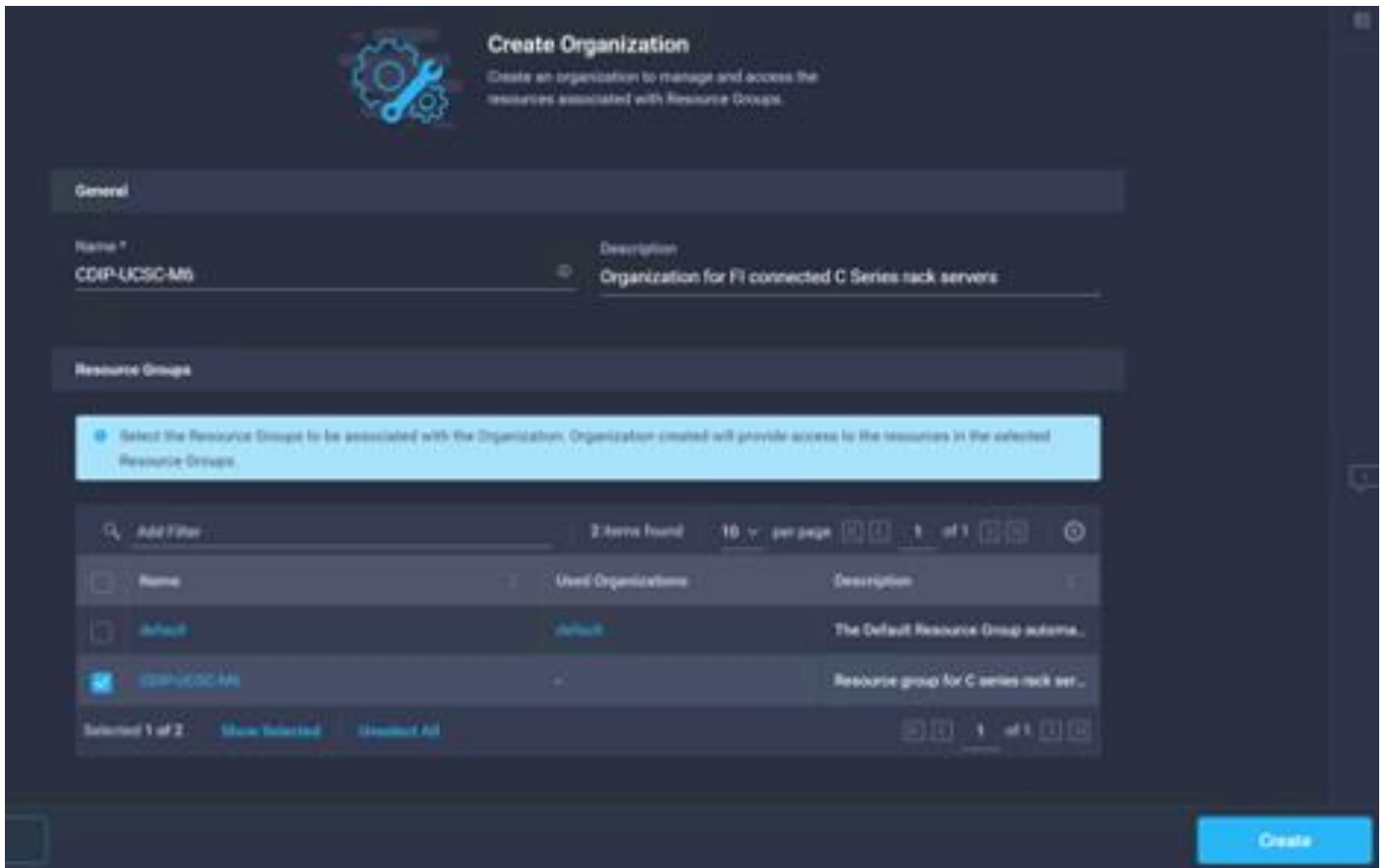
Step 3. Select Target to be part of the resource group, click Create.



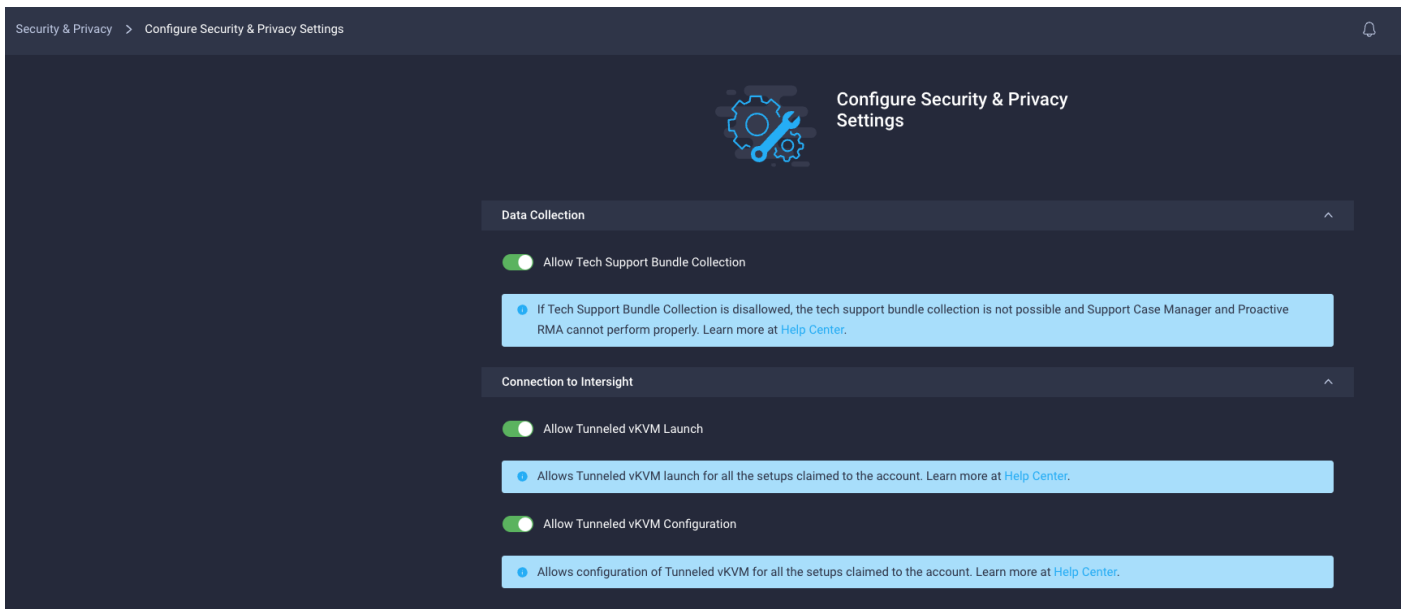
Step 4. In access & Permissions section, select Organizations; click Create Organization



Step 5. Enter details for new Organization creation. Click Create.



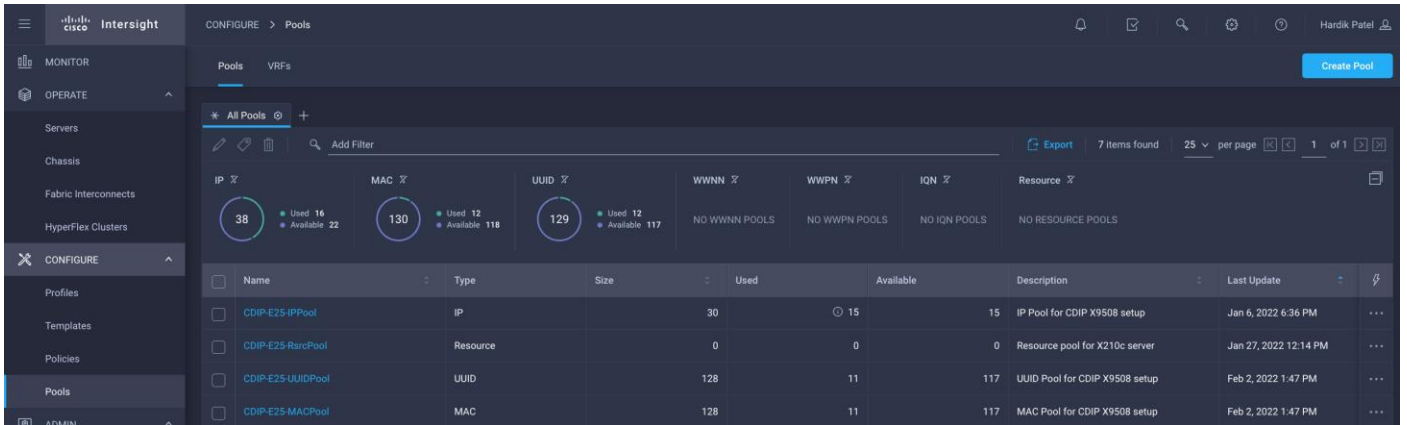
Step 6. In security and Privacy settings click on Configure to enable allow Tunneled vKVM Launch and configuration. Click Save.



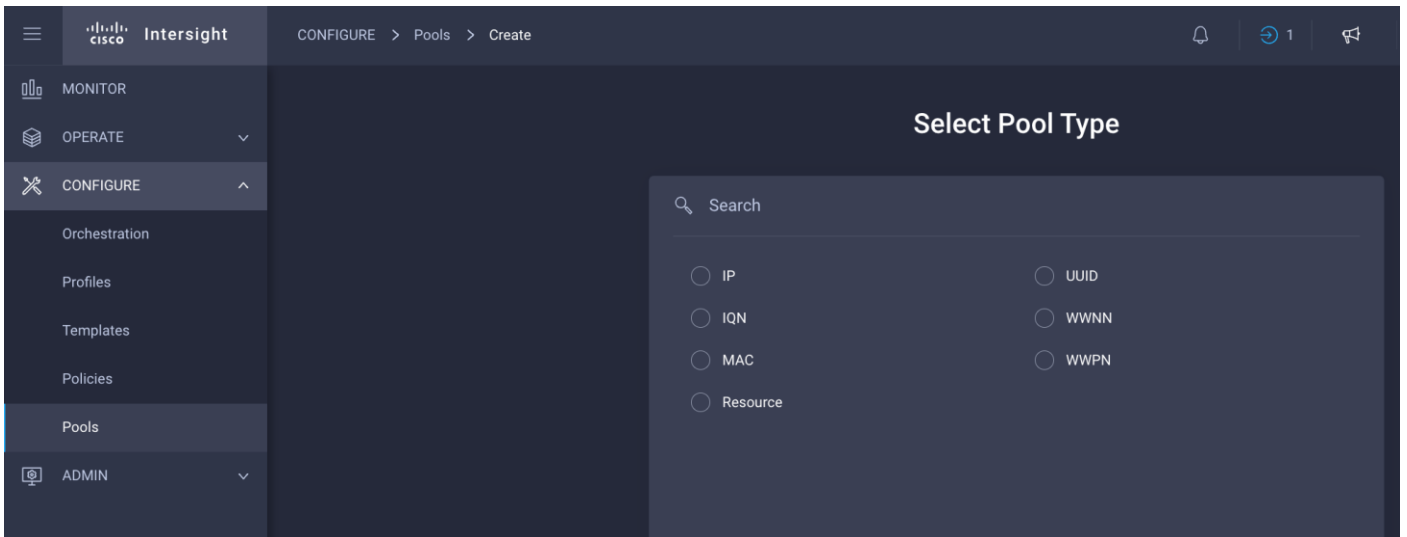
Procedure 4. Configure Cisco Intersight Pools, Policies and Profiles

Note: Cisco Intersight requires different pools and policies which can be created at the time of profile creation or can be pre-populated and attached to the profile.

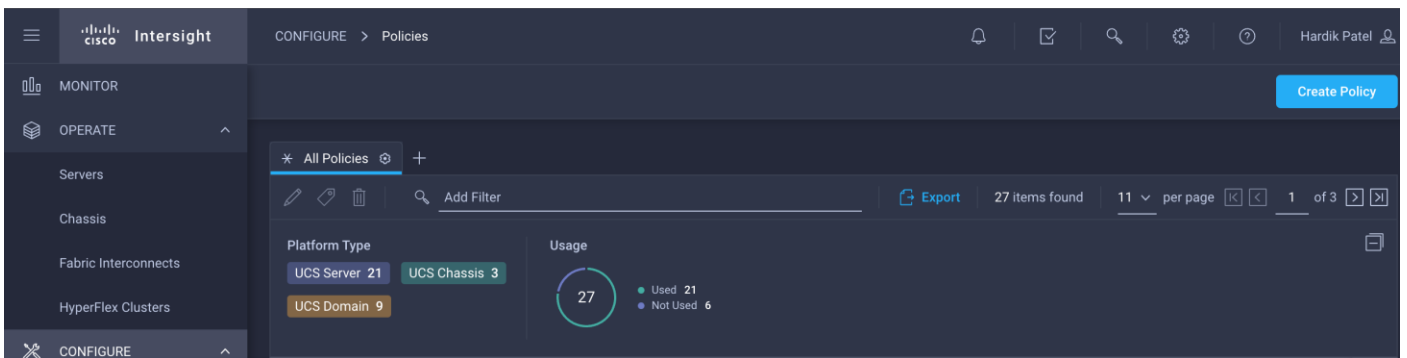
Step 1. To create the required set of pools, go to Configure > Pools. Click Create Pool.



Step 2. Select one of the pool type creation and provide a range for the pool creation.



Step 3. To create the required set of policies, go to Configure > Policies. Click Create Policy.



Cisco UCS Domain Profile

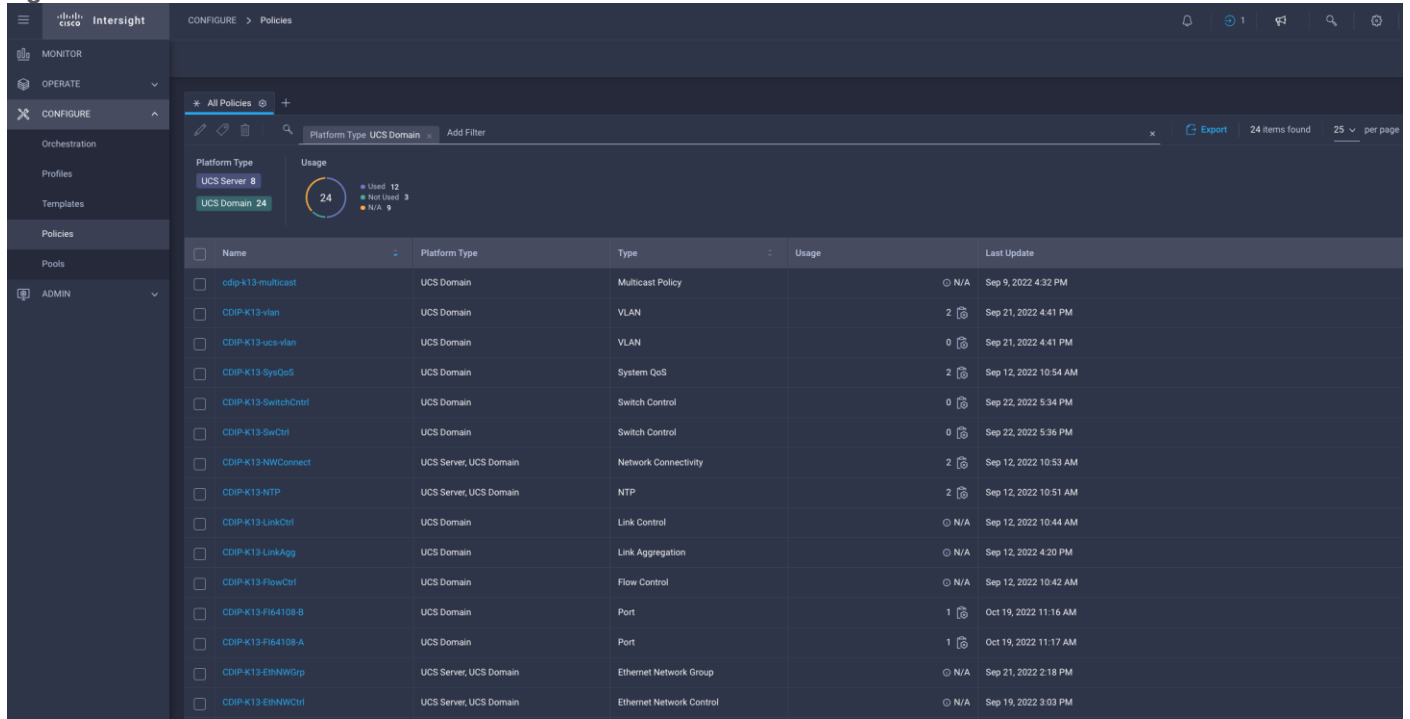
A Cisco UCS domain profile configures a pair of fabric interconnect through reusable policies, allows configuration of the ports and port channels, and configures the VLANs to be used in the network. It defines the characteristics of and configures the ports on the fabric interconnects. One Cisco UCS domain profile can be assigned to one fabric interconnect domain, and the Cisco Intersight platform supports the attachment of one port policy per Cisco UCS domain profile.

Some of the characteristics of the Cisco UCS domain profile environment are:

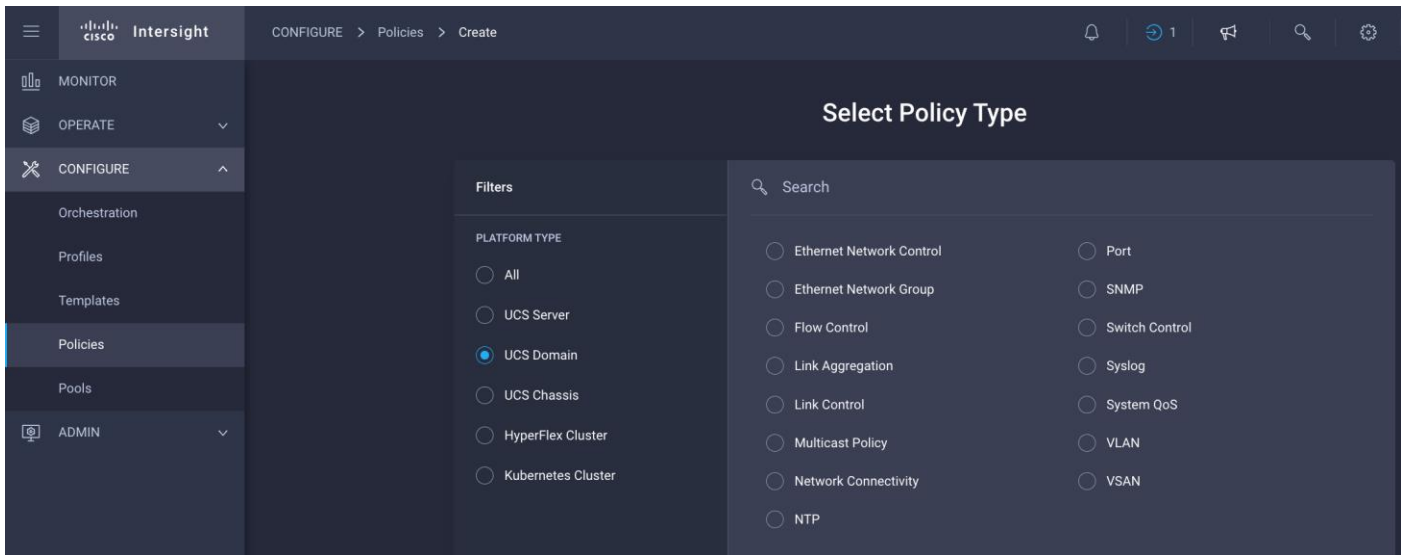
- A single domain profile is created for the pair of Cisco UCS fabric interconnects.
- Unique port policies are defined for the two fabric interconnects.
- The VLAN configuration policy is common to the fabric interconnect pair because both fabric interconnects are configured for same set of VLANs.
- The Network Time Protocol (NTP), network connectivity, and system Quality-of-Service (QoS) policies are common to the fabric interconnect pair.

After the Cisco UCS domain profile has been successfully created and deployed, the policies including the port policies are pushed to Cisco UCS fabric interconnects. Cisco UCS domain profile can easily be cloned to install additional Cisco UCS systems. When cloning the UCS domain profile, the new UCS domains utilize the existing policies for consistent deployment of additional UCS systems at scale.

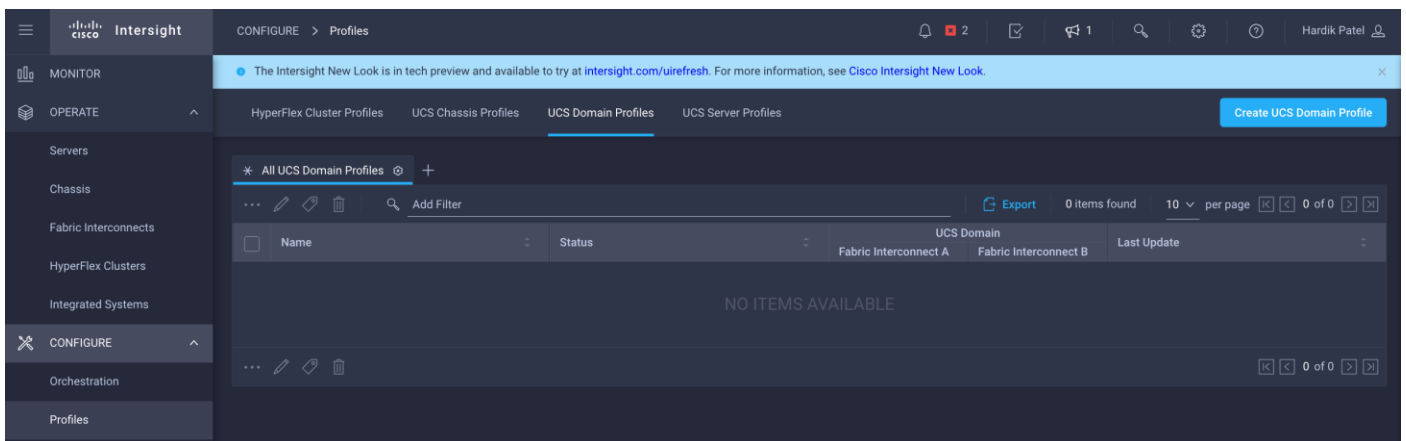
Figure 40. Cisco UCS Domain Policies



Step 1. Create policies for UCS Domain which will be applied to fabric interconnects.



Step 2. Go to Configure > Profiles. Click Create UCS Domain Profile.



Step 3. Click Start.


Create UCS Domain Profile

A UCS domain profile streamlines fabric interconnect assignment, port, and fabric interconnect configuration to eliminate failures caused by inconsistent configuration.

UCS Domain Assignment

Create a Fabric Interconnect pair and assign to a domain profile immediately or later.



 [About UCS Domain Profile Creation](#)

Do not show this page again

Start >

Step 4. Select organization, add name, description, and tag for the UCS Domain Profile.



Step 1

General

Add a name, description and tag for the UCS domain profile.

Organization *

CDIP-UCSC-M6



Name *

CDIP-K13-FI64108



Set Tags

Description

<= 1024

Step 5. Select UCS Domain to assign UCS Domain Profile.



Step 2 UCS Domain Assignment

Choose to assign a fabric interconnect pair to the profile now or later.

Assign Now

Assign Later

1 Choose to assign a fabric interconnect pair now or later. If you choose Assign Now, select a pair that you want to assign and click Next . If you choose Assign Later, click Next to proceed to policy selection.

Show Assigned

Search: Add Filter ⚙️

| | Domain Name | Fabric Interconnect A | | | Fabric Interconnect B | | |
|----------------------------------|----------------|-----------------------|-------------|----------------|-----------------------|-------------|----------------|
| | | Model | Serial | Bundle Version | Model | Serial | Bundle Version |
| <input checked="" type="radio"/> | K13-CDIP-64108 | UCS-FI-64108 | FD023360Y8X | | UCS-FI-64108 | FD023360Y9E | |

Step 6. Select policy for VLAN and VSAN configuration as applicable.



Step 3 VLAN & VSAN Configuration

Create or select a policy for the fabric interconnect pair.

Fabric Interconnect A 1 of 2 Policies Configured

VLAN Configuration

× | ✎ | CDIP-K13-vlan 📄

VSAN Configuration

Select Policy 📄

Fabric Interconnect B 1 of 2 Policies Configured

VLAN Configuration

× | ✎ | CDIP-K13-vlan 📄

VSAN Configuration

Select Policy 📄

Step 7. Sample VLAN policy configuration. Configure VLAN policy as required.



Step 2 Policy Details

Add policy details

This policy is applicable only for UCS Domains

VLANs

Add VLANs

Show VLAN Ranges

2 items found | 25 per page | 1 of 1

Add Filter

| <input type="checkbox"/> | VLAN ID | Name | Sharing Type | Primary VL... | Multicast Policy | Auto Allow On Up... | ⚡ |
|--------------------------|---------|-----------------|--------------|---------------|---------------------|---------------------|-----|
| <input type="checkbox"/> | 1 | default | None | | | Yes | ... |
| <input type="checkbox"/> | 4 | cdip-k13-vlan_4 | None | | cdip-k13-multica... | Yes | ... |

1 of 1

Set Native VLAN ID

VLAN ID

1

Step 8. Select Ports Configuration for FI - A and FI - B.



Step 4

Ports Configuration

Create or select a port policy for the fabric interconnect pair.

- 1 Configure ports by creating or selecting a policy.

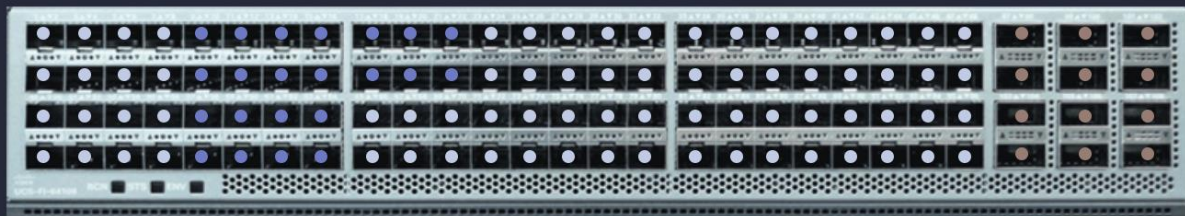
Fabric Interconnect A Configured

Fabric Interconnect B Configured

Ports Configuration

Selected Policy CDIP-K13-FI64108-B

Ports | Port Channels




● Ethernet Uplink Port Channel ● Server ● Unconfigured

Step 9. Port Configuration policy creation allows to configure port roles based on the requirement i.e. Server ports, uplink port, Port channel configuration, Unified ports, and breakout options.

Fabric Interconnect B Configured

Ports Configuration Selected Policy CDIP-K13-FI64108-B

Ports Port Channels



Legend: ● Ethernet Uplink Port Channel ● Server ● Unconfigured

| Port Type | | Port Channel Type | |
|-----------|-----|-------------------|---|
| Ethernet | 108 | Ethernet Uplink | 1 |

| Port Role | | Port Channel Role | |
|--------------|----|-------------------|----|
| Server | 22 | Ethernet Uplink | 12 |
| Unconfigured | 74 | | |

Step 10. Create Port role as Server for ports connected to Cisco UCS servers.



Configure (8 Ports)

Configuration

Selected Ports

Port 1, Port 2, Port 3, Port 4, Port 5, Port 6, Port 7, Port 8

Role

Server



i N9K-C93180YC-FX3 requires CI74 FEC for 25G speed ports. Learn more at [Help Center](#).

FEC **i**



Auto



CI74



Manual Chassis/Server Numbering **i**

Step 11. Create Ethernet Uplink Port Channel for ports connected to pair of Nexus 9000 switch. Create or assign policies to attach with Ethernet Uplink Port Channel.

- Flow Control
- Link Aggregation
- Link Control
- Ethernet Network Group

Note: The Ethernet Network Group Policy specifies a set of VLANs to allow on the uplink port. The specified VLAN set must be either identical or disjoint from those specified on other uplink interfaces. Ensure that the VLANs are defined in the VLAN Policy, and 'Auto Allow on Uplinks' option is disabled. Note, default VLAN-1 is auto allowed and can be specified as the native VLAN.

Figure 41. Cisco UCS Port Channel configuration for Fabric Interconnect A

i The combined maximum number of Ethernet Uplink, FCoE Uplink, and Appliance port channels permitted is 12 and the maximum number of FC port channels permitted is 4.

Role
Ethernet Uplink Port Channel ▼

Port Channel ID * ⓘ Admin Speed ▼ ⓘ
142 1 - 256 Auto

Ethernet Network Group ⓘ
[Select Policy](#)

Flow Control
Selected Policy CDIP-K13-FlowCtrl 👁 | ✕

Link Aggregation
Selected Policy CDIP-K13-LinkAgg 👁 | ✕

Link Control
Selected Policy CDIP-K13-LinkCtrl 👁 | ✕

Select Member Ports

i FC or Ethernet ports with unconfigured role are available for port channel creation.

Figure 42. Cisco UCS Port Channel configuration for Fabric Interconnect B

The screenshot displays the configuration page for a Port Channel in Cisco UCS. At the top, a 'Configuration' header is visible. A light blue notification box contains the text: 'The combined maximum number of Ethernet Uplink, FCoE Uplink, and Appliance port channels permitted is 12 and the maximum number of FC port channels permitted is 4.' Below this, the 'Role' is set to 'Ethernet Uplink Port Channel'. The 'Port Channel ID' is '141' (range 1-256) and the 'Admin Speed' is 'Auto'. Under 'Ethernet Network Group', there is a 'Select Policy' button. The 'Flow Control' section shows 'Selected Policy' as 'CDIP-K13-FlowCtrl'. The 'Link Aggregation' section shows 'Selected Policy' as 'CDIP-K13-LinkAgg'. The 'Link Control' section shows 'Selected Policy' as 'CDIP-K13-LinkCtrl'. A 'Select Member Ports' section is partially visible at the bottom. A final light blue notification box at the bottom states: 'FC or Ethernet ports with unconfigured role are available for port channel creation.'

Step 12. Select compute and management policies to be associated with fabric interconnects in UCS Domain configuration step.

CONFIGURE > Edit UCS Domain Profile (CDIP-K13-FI64108)

Step 5
UCS Domain Configuration
Select the compute and management policies to be associated with the fabric interconnect.

Show Attached Policies (3)

Management 2 of 4 Policies Configured

| | |
|----------------------|------------------------------------------------------------------------------------|
| NTP | x edit CDIP-K13-NTP copy |
| Syslog | Select Policy copy |
| Network Connectivity | x edit CDIP-K13-NWConnect copy |
| SNMP | Select Policy copy |

Network 1 of 2 Policies Configured

| | |
|----------------|---------------------------------------------------------------------------------|
| System QoS * | x edit CDIP-K13-SysQoS copy |
| Switch Control | Select Policy copy |

Progress

- General
- UCS Domain Assignment
- VLAN & VSAN Configuration
- Ports Configuration
- UCS Domain Configuration**
- Summary

Step 13. System QoS policy with below configuration was deployed.

Figure 43. QoS Policy to be attached with Cisco UCS Domain Profile



Step 2

Policy Details

Add policy details

ⓘ This policy is applicable only for UCS Domains

Configure Priorities

| | | | | |
|---------------------------------------------------|------------|--------------|--------------------------------------------------------|-------------|
| <input checked="" type="checkbox"/> Platinum | CoS 5 | Weight 10 | <input type="checkbox"/> Allow Packet Drops | MTU 9216 |
| | 0 - 6 | 0 - 10 | | 1500 - 9216 |
| <input type="checkbox"/> Gold | | | | |
| <input type="checkbox"/> Silver | | | | |
| <input type="checkbox"/> Bronze | | | | |
| <input checked="" type="checkbox"/> Best Effort | CoS Any | Weight 5 | <input checked="" type="checkbox"/> Allow Packet Drops | MTU 1500 |
| | 0 - 6 | 0 - 10 | | 1500 - 9216 |
| <input checked="" type="checkbox"/> Fibre Channel | CoS 3 | Weight 5 | <input type="checkbox"/> Allow Packet Drops | MTU 2240 |
| | 0 - 6 | 0 - 10 | | 1500 - 9216 |

Step 14. Review UCS Domain profile summary. Click Deploy.

Progress

- General
- UCS Domain Assignment
- VLAN & VSAN Configuration
- Ports Configuration
- UCS Domain Configuration
- Summary

Step 6 Summary

Review the UCS domain profile details, resolve configuration errors and deploy the profile.

General

Name: CDIP-K13-FI64108 Status: OK

Organization: CDIP-UCSC-M6

| Fabric Interconnect | Model | Serial | Requires Reboot |
|---------------------|--------------|-------------|-----------------|
| K13-CDIP-64108 FI-A | UCS-FI-64108 | FD023360Y8X | No |
| K13-CDIP-64108 FI-B | UCS-FI-64108 | FD023360Y9E | No |

[Ports Configuration](#)
 [VLAN & VSAN Configuration](#)
 [UCS Domain Configuration](#)
 [Errors / Warnings](#)

< Back
Close
Deploy

Step 15. After successful deployment of domain profile chassis and/or server discovery will start according to connection between Cisco UCS hardware.

Figure 44. Cisco UCS X9508 Chassis tab in Intersight Managed Mode

Intersight OPERATE > Chassis > E26-FI6454-1

MONITOR | **OPERATE** | **CONFIGURE** | **ADMIN**

Chassis Details | Inventory | Connections

Health Healthy

Properties UCS-9508 Front View | Rear View

Details

- Name: E26-FI6454-1
- Serial: FOX2501P0C4
- Model: UCSX-9508
- Revision: 0
- Part Number: 68-6847-03
- Management Mode: Intersight
- Contract Status: Not Covered
- UCS Domain: E26-FI6454
- Chassis Profile: CDIP-E25-X9508-01
- Contract Coverage
- Contract Status: Not Covered
- Organization: default ITZ-CDIP-Test
- Tags: Set

Locator LED **Health Overlay**

Figure 45. Cisco Intersight Servers tab reporting Cisco UCS X210c M6 Compute Node

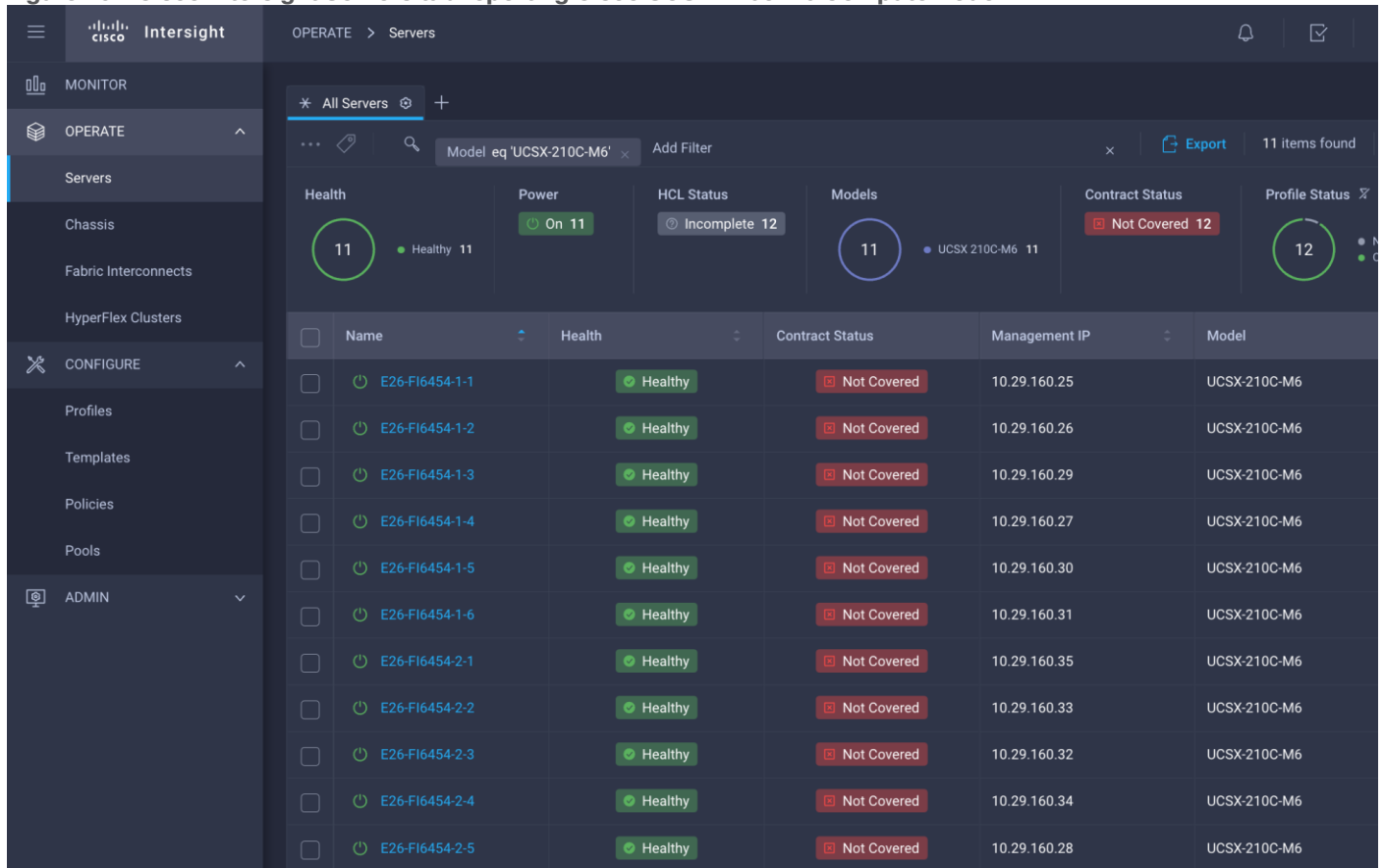
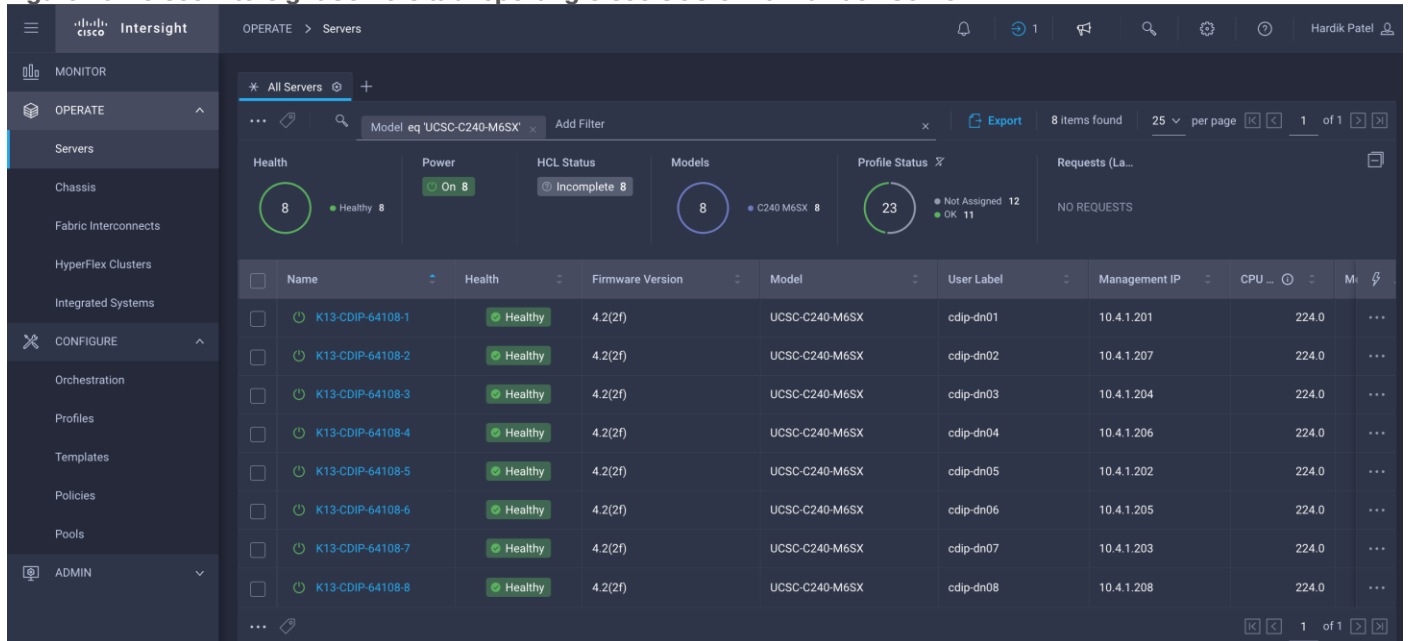


Figure 46. Cisco Intersight Servers tab reporting Cisco UCS C240 M6 Rack Server

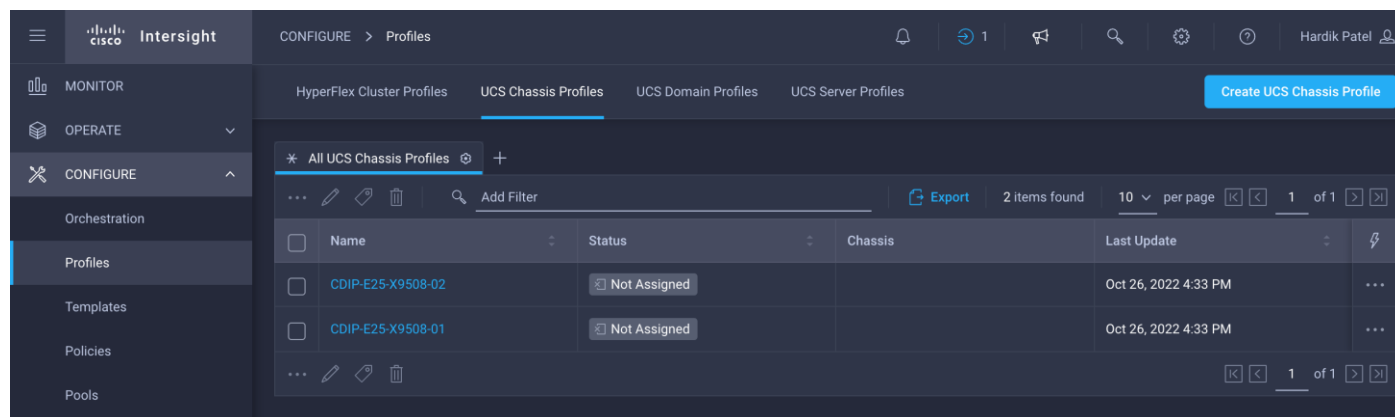


Cisco UCS Chassis Profile

The Cisco UCS X9508 Chassis and Cisco UCS X210c M6 Compute Nodes are automatically discovered when the ports are successfully configured using the domain profile, as shown in the following figures.

Procedure 1. Create UCS Chassis profile

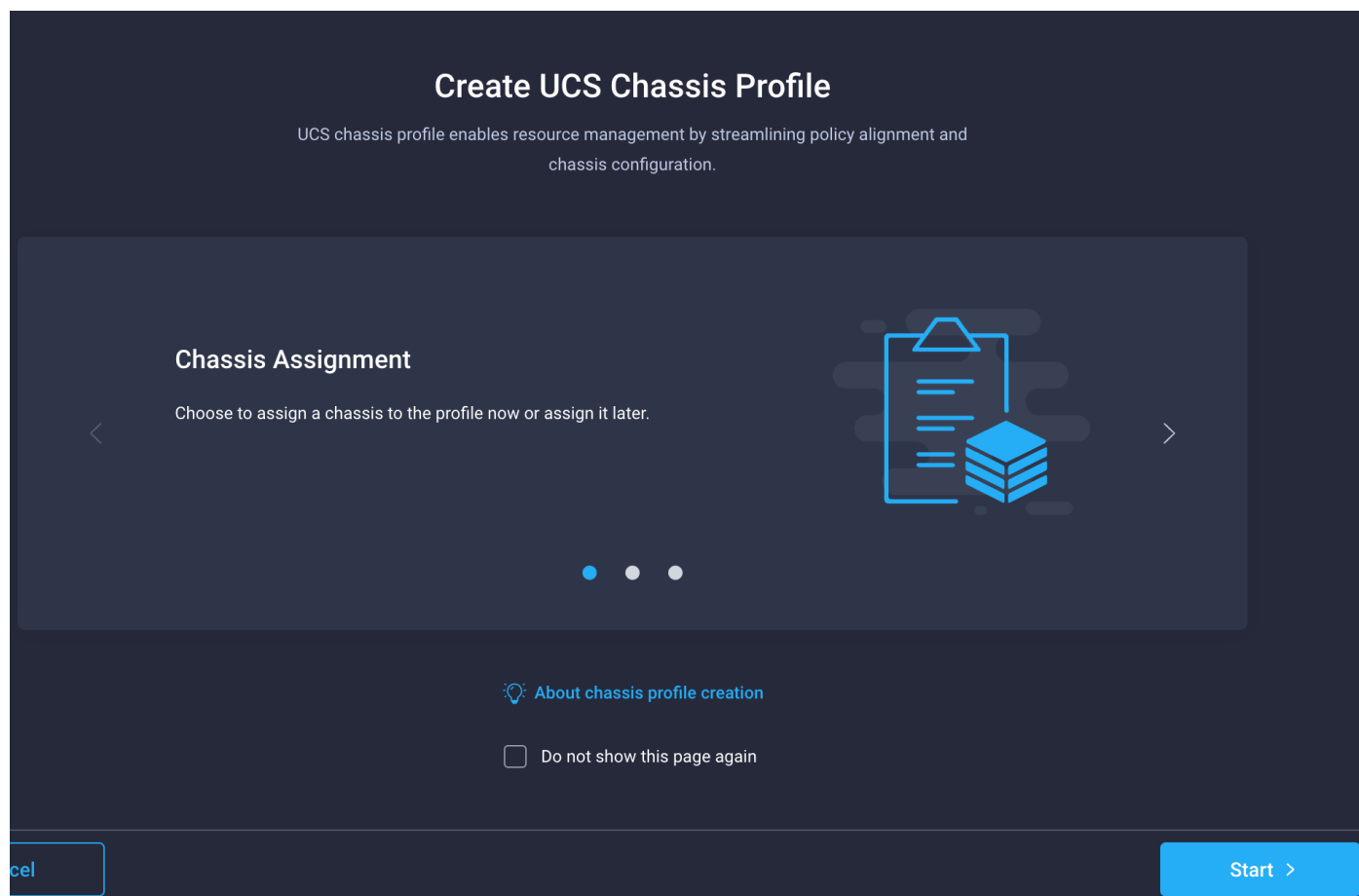
Step 1. To create UCS Chassis profile, go to Configure > Profiles > UCS Chassis Profiles. Click Create UCS Chassis Profile.



The screenshot shows the Cisco Intersight interface for configuring UCS Chassis Profiles. The breadcrumb navigation is 'CONFIGURE > Profiles'. The main content area is titled 'UCS Chassis Profiles' and contains a table with the following data:

| Name | Status | Chassis | Last Update |
|-------------------|--------------|---------|----------------------|
| CDIP-E25-X9508-02 | Not Assigned | | Oct 26, 2022 4:33 PM |
| CDIP-E25-X9508-01 | Not Assigned | | Oct 26, 2022 4:33 PM |

Step 2. Click Start.



The screenshot shows the 'Create UCS Chassis Profile' wizard. The title is 'Create UCS Chassis Profile'. The subtitle reads: 'UCS chassis profile enables resource management by streamlining policy alignment and chassis configuration.' The main content area is titled 'Chassis Assignment' and contains the text: 'Choose to assign a chassis to the profile now or assign it later.' There is a 'Start >' button at the bottom right.

Step 3. Select organization, enter name, tags, and description for the UCS Chassis profile.

CONFIGURE > Create UCS Chassis Profile

Progress

- 1 General
- 2 Chassis Assignment
- 3 Chassis Configuration
- 4 Summary

Step 1
General
Enter a name, description and tag for the chassis profile.

Organization *
CDIP-UCSC-M6

Name *
UCSX-Chassis

Set Tags
CDIP AAE25X9508 Enter a tag in the key:value format.


Description
≤ 1024

Step 4. Select chassis configuration policies.

Step 3
Chassis Configuration
Create or select existing policies that you want to associate with this chassis profile.

| | |
|------------|----------------------|
| IMC Access | CDIP-K13-IMCAccess |
| Power | CDIP-K13-PowerPolicy |
| SNMP | |
| Thermal | K13-Thermal |

Step 5. Review the chassis profile summary.



Step 4

Summary

Verify details of the chassis profile and policies, resolve errors if any, and deploy.




General

| | | | |
|------------------|--------------|--------|----------------------------------------------------------------------------|
| Organization | CDIP-UCSC-M6 | Status | Not Assigned |
| Name | UCSX-Chassis | Tags | |
| Assigned Chassis | - | | |

Tags

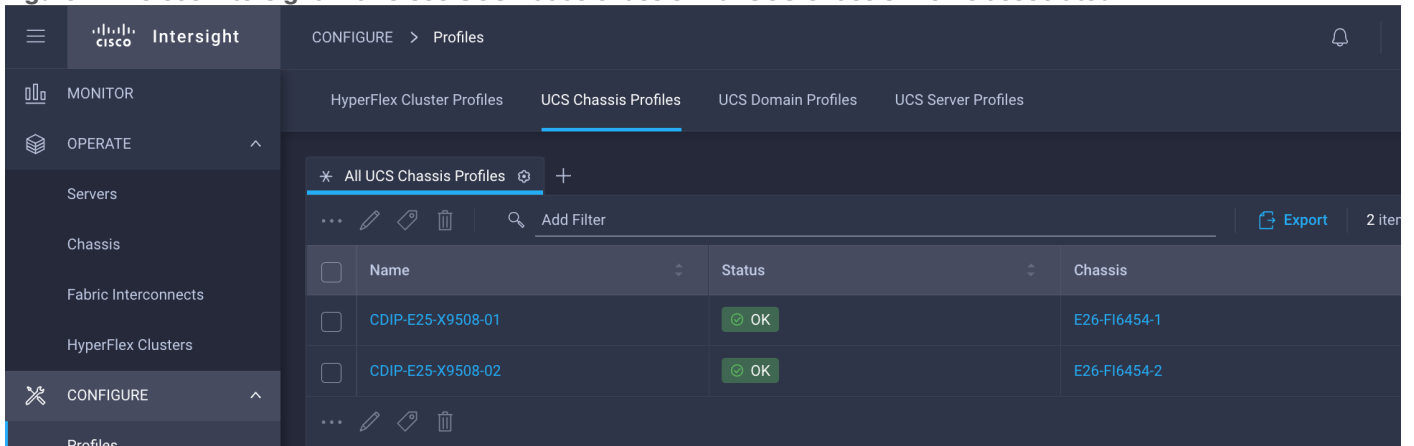
CDIP AAE25X9508

Chassis Configuration
Errors/Warnings (0)

| | |
|------------|-----------------------------------------------------------------------------------------------------------|
| IMC Access | CDIP-K13-IMCAccess  |
| Power | CDIP-K13-PowerPolicy  |
| Thermal | K13-Thermal  |

Step 6. Click Deploy Chassis Profile

Figure 47. Cisco Intersight with Cisco UCS X9508 chassis with UCS Chassis Profile associated



The screenshot shows the Cisco Intersight interface. The left sidebar has 'CONFIGURE' selected. The main area shows 'CONFIGURE > Profiles' with tabs for 'HyperFlex Cluster Profiles', 'UCS Chassis Profiles', 'UCS Domain Profiles', and 'UCS Server Profiles'. The 'UCS Chassis Profiles' tab is active, displaying a table with the following data:

| Name | Status | Chassis |
|-------------------|--------|--------------|
| CDIP-E25-X9508-01 | OK | E26-F16454-1 |
| CDIP-E25-X9508-02 | OK | E26-F16454-2 |

Cisco UCS Server Profile

In Cisco Intersight, a Server Profile enables resource management by streamlining policy alignment, and server configuration. After creating Server Profiles, you can edit, clone, deploy, attach to a template, create a template, detach from template, or unassign them as required. From the Server Profiles table view, you can select a profile to view details in the Server Profiles Details view.

Procedure 1. Create Cisco Intersight Policy

Create BIOS Policy

Step 1. Go to Configure > Policies > Create Policy.

Step 2. Select policy type as BIOS.

The screenshot shows the 'Select Policy Type' screen. On the left, under 'Filters', the 'PLATFORM TYPE' section has radio buttons for 'All', 'UCS Server' (selected), 'UCS Domain', 'UCS Chassis', 'HyperFlex Cluster', and 'Kubernetes Cluster'. The main area has a search bar and a list of policy types with radio buttons: Adapter Configuration, BIOS (selected), Boot Order, Certificate Management, Device Connector, Ethernet Adapter, Ethernet Network, Ethernet Network Control, iSCSI Static Target, LAN Connectivity, LDAP, Local User, Network Connectivity, NTP, Persistent Memory, and Power.

Step 3. Enter Add a name, description, and tag for the BIOS Policy. Click Next.



Step 1 General

Add a name, description and tag for the policy.

Organization *

CDIP-UCSC-M6



Name *

CDIP-BIOS

Set Tags

Description

<= 1024

Step 4. Edit BIOS options by click on + sign and edit required value for each of the BIOS settings. Sample BIOS configuration is shown below:

CONFIGURE > Policies > BIOS > CDIP-K13-BIOS > Edit

Step 2
Policy Details
Add policy details

All Platforms | UCS Server (Standalone) | UCS Server (FI-Attached)

▲ The BIOS settings will be applied only on next host reboot.

- + Boot Options
- + Intel Directed IO
- + LOM And PCIe Slots
- + Main
- + Memory
- + PCI
- + Power And Performance

CONFIGURE > Policies > BIOS > Create

All Platforms | UCS Server (Standalone) | UCS Server (FI-Attached)

▲ The BIOS settings will be applied only on next host reboot.

— Boot Options

| | |
|-------------------------------------------------|--------------------------------------------------|
| Number of Retries platform-default | Cool Down Time (sec) platform-default |
| Boot Option Retry platform-default | IPV4 HTTP Support platform-default |
| IPV4 PXE Support platform-default | IPV6 HTTP Support platform-default |
| IPV6 PXE Support platform-default | Network Stack platform-default |
| Onboard SCU Storage Support platform-default | Onboard SCU Storage SW Stack platform-default |
| Power ON Password platform-default | P-SATA Mode platform-default |
| SATA Mode platform-default | VMD Enablement enabled |

CONFIGURE > Policies > BIOS > Create

Progress

- 1 General
- 2 Policy Details

| | |
|--------------------------------------------------|----------------------------------------------------|
| platform-default | platform-default |
| AMD Memory Interleaving Size platform-default | SEV-SNP Support platform-default |
| CKE Low Policy platform-default | CR QoS platform-default |
| CR FastGo Config platform-default | DCPMM Firmware Downgrade platform-default |
| DRAM Refresh Rate 1x | DRAM SW Thermal Throttling platform-default |
| eADR Support platform-default | Low Voltage DDR Mode platform-default |
| Memory Bandwidth Boost platform-default | Memory Refresh Rate platform-default |
| Memory Size Limit in GiB * platform-default | Memory Thermal Throttling Mode platform-default |
| Mirroring Mode platform-default | NUMA Optimized platform-default |
| NVM Performance Setting platform-default | Operation Mode platform-default |

CONFIGURE > Policies > BIOS > CDIP-K13-BIOS > Edit

Progress

- 1 General
- 2 Policy Details

Processor

| | |
|----------------------------------------------|-------------------------------------------------|
| Adjacent Cache Line Prefetcher enabled | Altitude platform-default |
| Autonomous Core C State platform-default | CPU Autonomous C State platform-default |
| Boot Performance Mode platform-default | Burst and Postponed Refresh platform-default |
| APBDIS platform-default | Downcore Control platform-default |
| Streaming Stores Control platform-default | Fixed SOC P-State platform-default |
| DF C-States platform-default | CCD Control platform-default |
| CPU Downcore control platform-default | CPU SMT Mode platform-default |

CONFIGURE > Policies > BIOS > CDIP-K13-BIOS > Edit

Progress

- 1 General
- 2 Policy Details

| | | | |
|-----------------------------------|------------------|--------------------------------|------------------|
| ACPI SRAT L3 Cache As NUMA Domain | platform-default | Channel Interleaving | platform-default |
| Cisco xGMI Max Speed | platform-default | Closed Loop Thermal Throttling | platform-default |
| Processor CMC1 | platform-default | Config TDP | platform-default |
| Configurable TDP Level | platform-default | Core Multi Processing | all |
| Energy Performance | performance | Frequency Floor Override | enabled |
| CPU Performance | enterprise | Power Technology | performance |
| Demand Scrub | disabled | Direct Cache Access Support | enabled |
| DRAM Clock Throttling | Performance | Energy Efficient Turbo | enabled |

CONFIGURE > Policies > BIOS > CDIP-K13-BIOS > Edit

Progress

- 1 General
- 2 Policy Details

| | | | |
|---------------------------|------------------|----------------------------------------|------------------|
| Energy Performance Tuning | platform-default | Enhanced Intel Speedstep(R) Technology | enabled |
| Processor EPP Enable | platform-default | EPP Profile | platform-default |
| Execute Disable Bit | platform-default | Local X2 Apic | platform-default |
| Hardware Prefetcher | enabled | CPU Hardware Power Management | platform-default |
| IMC Interleaving | platform-default | Intel Dynamic Speed Select | platform-default |
| Intel HyperThreading Tech | enabled | Intel Speed Select | platform-default |
| Intel Turbo Boost Tech | enabled | Intel(R) VT | disabled |
| I/O Error Enable | platform-default | DCU IP Prefetcher | enabled |

CONFIGURE > Policies > BIOS > CDIP-K13-BIOS > Edit

Progress

- 1 General
- 2 Policy Details

| | | | |
|---------------------------|------------------|--------------------------|------------------|
| KTI Prefetch | enabled | LLC Prefetch | enabled |
| Intel Memory Interleaving | platform-default | Package C State Limit | platform-default |
| Patrol Scrub | disabled | Patrol Scrub Interval * | platform-default |
| Processor CTE | disabled | Processor C3 Report | disabled |
| Processor C6 Report | disabled | CPU C State | disabled |
| P-STATE Coordination | HW ALL | Power Performance Tuning | platform-default |
| UPI Link Frequency Select | platform-default | Rank Interleaving | platform-default |
| Single PCTL | platform-default | SMT Mode | platform-default |

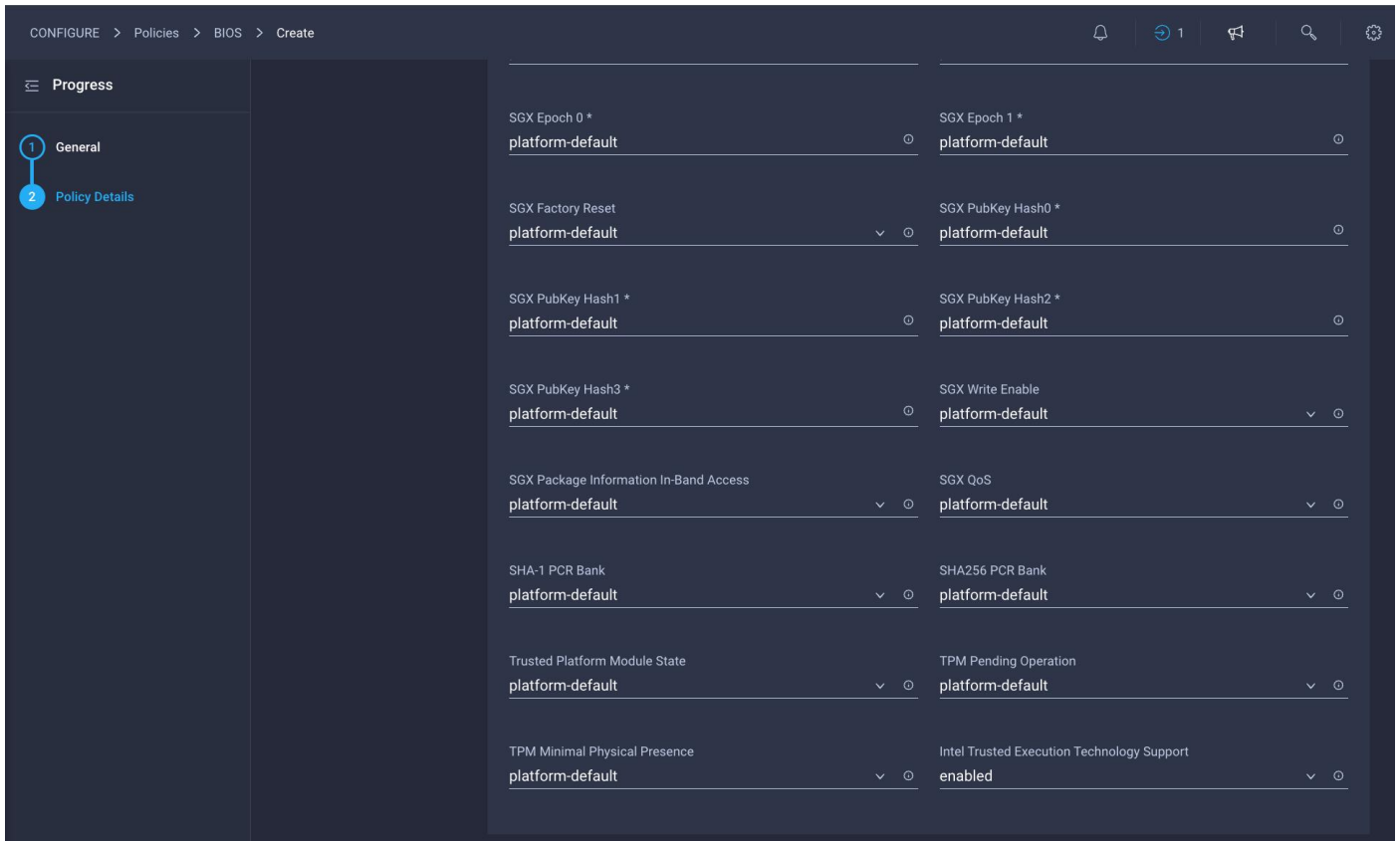
CONFIGURE > Policies > BIOS > CDIP-K13-BIOS > Edit

Progress

- 1 General
- 2 Policy Details

| | | | |
|------------------------|------------------|--------------------------|------------------|
| Single PCTL | platform-default | SMT Mode | platform-default |
| Sub Numa Clustering | platform-default | DCU Streamer Prefetch | enabled |
| SVM Mode | platform-default | Uncore Frequency Scaling | platform-default |
| Workload Configuration | platform-default | XPT Prefetch | enabled |

- + QPI
- + Serial Port
- + Server Management
- + Trusted Platform
- + USB



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

Note: For more information, go to: [Performance Tuning Guide](#).

Procedure 2. Create Boot Order Policy

Step 1. Go to Configure > Policies > Create Policy.

Step 2. Select policy type as Boot Order.

Select Policy Type

Filters

PLATFORM TYPE

- All
- UCS Server
- UCS Domain
- UCS Chassis
- HyperFlex Cluster
- Kubernetes Cluster

Search

- Adapter Configuration
- Add-ons
- Auto Support
- Backup Configuration
- BIOS
- Boot Order
- Certificate Management
- Container Runtime
- Local User
- Multicast Policy
- Network CIDR
- Network Configuration
- Network Connectivity
- Node IP Ranges
- Node OS Configuration
- NTP

Step 3. Add a name, description, and tag for the Boot Order policy. Click Next.

Step 4. Configure UEFI Boot Mode with Enable Secure Boot. Enable Local Disk with M.2 drive installed in “MSTOR-RAID” slot and CIMC Mapped DVD. Additional boot devices can be added, or boot order can be adjusted as required.

Note: UEFI Boot Mode with Enable Secure Boot required Trusted Execution Technology (TXT) Support Enabled in BIOS policy.

Configured Boot Mode ⓘ

Unified Extensible Firmware Interface (UEFI) Legacy

Enable Secure Boot ⓘ

Add Boot Device | ▾

| Local Disk (m2-hwboot) Enabled 🗑️ ⬆️ ⬇️ | |
|-------------------------------------------------------------------------------------------------------------------|--------------------------|
| Device Name * m2-hwboot ⓘ | Slot MSTOR-RAID ⓘ |
| Bootloader Name ⓘ | Bootloader Description ⓘ |
| Bootloader Path ⓘ | |

| Virtual Media (vMedia-kvm) Enabled 🗑️ ⬆️ ⬇️ | |
|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| Device Name * vMedia-kvm ⓘ | |
| | Sub-Type CIMC MAPPED DVD ⬇️ ⓘ |

Procedure 3. Create Virtual Media Policy

- Step 1.** Go to Configure > Policies > Create Policy.
- Step 2.** Select policy type as Virtual Media.

Select Policy Type

| Filters | Search |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>PLATFORM TYPE</p> <p><input checked="" type="radio"/> All</p> <p><input type="radio"/> UCS Server</p> <p><input type="radio"/> UCS Domain</p> <p><input type="radio"/> UCS Chassis</p> <p><input type="radio"/> HyperFlex Cluster</p> <p><input type="radio"/> Kubernetes Cluster</p> | <p><input type="radio"/> External iSCSI Storage</p> <p><input type="radio"/> FC Zone</p> <p><input type="radio"/> Fibre Channel Adapter</p> <p><input type="radio"/> Fibre Channel Network</p> <p><input type="radio"/> Fibre Channel QoS</p> <p><input type="radio"/> Flow Control</p> <p><input type="radio"/> HTTP Proxy</p> <p><input type="radio"/> Http Proxy Policy</p> <p><input type="radio"/> IMC Access</p> <p><input type="radio"/> IPMI Over LAN</p> <p><input type="radio"/> iSCSI Adapter</p> <p><input type="radio"/> iSCSI Boot</p> <p><input type="radio"/> iSCSI Static Target</p> <p><input type="radio"/> Kubernetes Version</p> <p><input type="radio"/> LAN Connectivity</p> <p><input type="radio"/> LDAP</p> <p><input type="radio"/> Link Aggregation</p> <p><input type="radio"/> Link Control</p> <p><input type="radio"/> SMTP</p> <p><input type="radio"/> SNMP</p> <p><input type="radio"/> SSH</p> <p><input type="radio"/> Storage</p> <p><input type="radio"/> Storage Configuration</p> <p><input type="radio"/> Switch Control</p> <p><input type="radio"/> Syslog</p> <p><input type="radio"/> System QoS</p> <p><input type="radio"/> Thermal</p> <p><input type="radio"/> Trusted Certificate Authorities</p> <p><input type="radio"/> UCSM Configuration</p> <p><input type="radio"/> vCenter</p> <p><input type="radio"/> Virtual KVM</p> <p><input type="radio"/> Virtual Machine Infra Config</p> <p><input type="radio"/> Virtual Machine Instance Type</p> <p><input checked="" type="radio"/> Virtual Media</p> <p><input type="radio"/> VLAN</p> <p><input type="radio"/> VSAN</p> |

Step 3. Enter name for vMedia Policy.



Step 1

General

Add a name, description and tag for the policy.

Organization *

CDIP-UCSC-M6



Name *

CDIP-vMedia

Set Tags

Description

<= 1024

Step 4. Click Add Virtual Media. Select Virtual Media Type and protocol. Enter required field value.

Add Virtual Media

Virtual Media Type ⓘ

CDD HDD

NFS

CIFS

HTTP/HTTPS

Name *

rhel8.6 ⓘ

File Location *

http://10.4.1.7/rheliso/rhel-8.6-x86_64-dvd.iso ⓘ

Mount Options ⓘ

Username

root ⓘ

Password

.....



Cancel

Add

Procedure 4. Create Virtual KVM Policy

Step 1. Go to Configure > Policies > Create Policy.

Step 2. Select policy type as IMC Access.

Step 3. Enable In-Band or Out-Of-Band Configuration and select IP Pool to assign as range of IP address for Virtual KVM access.

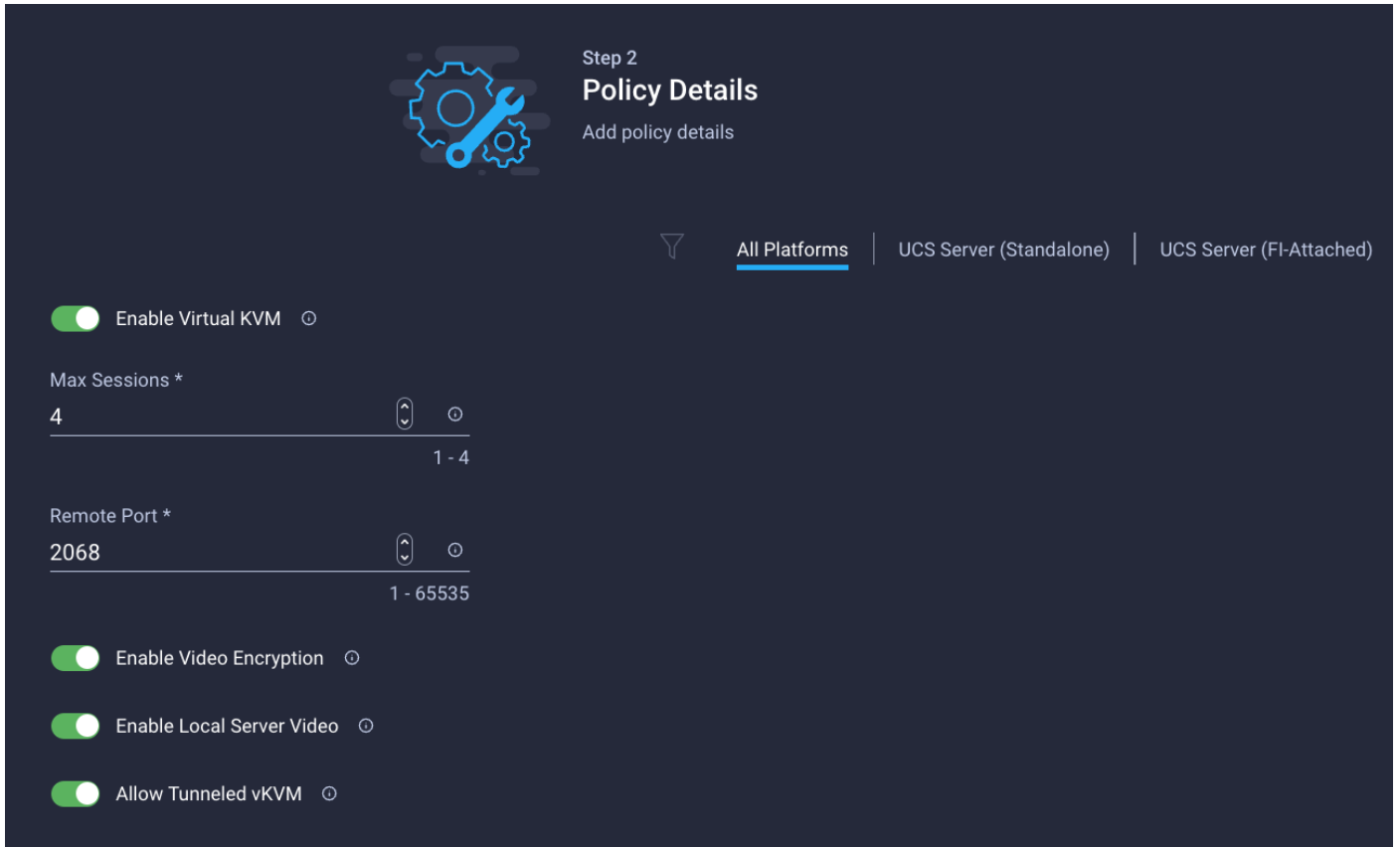
The screenshot displays the 'Policy Details' configuration interface. At the top, it indicates 'Step 2' and 'Add policy details'. A navigation bar shows 'All Platforms' selected, with 'UCS Server (FI-Attached)' and 'UCS Chassis' as options. A light blue warning box contains the text: 'A minimum of one configuration must be enabled. Policies like SNMP, vMedia and Syslog are currently not supported via Out-Of-Band and will require an In-Band IP to be configured. Check here for more info, [Help Centre](#)'. Below this, there are two configuration rows: 'In-Band Configuration' with a toggle switch set to 'Enabled', and 'Out-Of-Band Configuration' with a toggle switch set to 'Enabled'. At the bottom, the 'IP Pool' is set to 'CDIP-K13-IPPool', with an eye icon and a close icon next to it.

Procedure 5. Create Virtual KVM Policy

Step 1. Go to Configure > Policies > Create Policy.

Step 2. Select policy type as Virtual KVM.

Step 3. Virtual KVM Policy configuration.



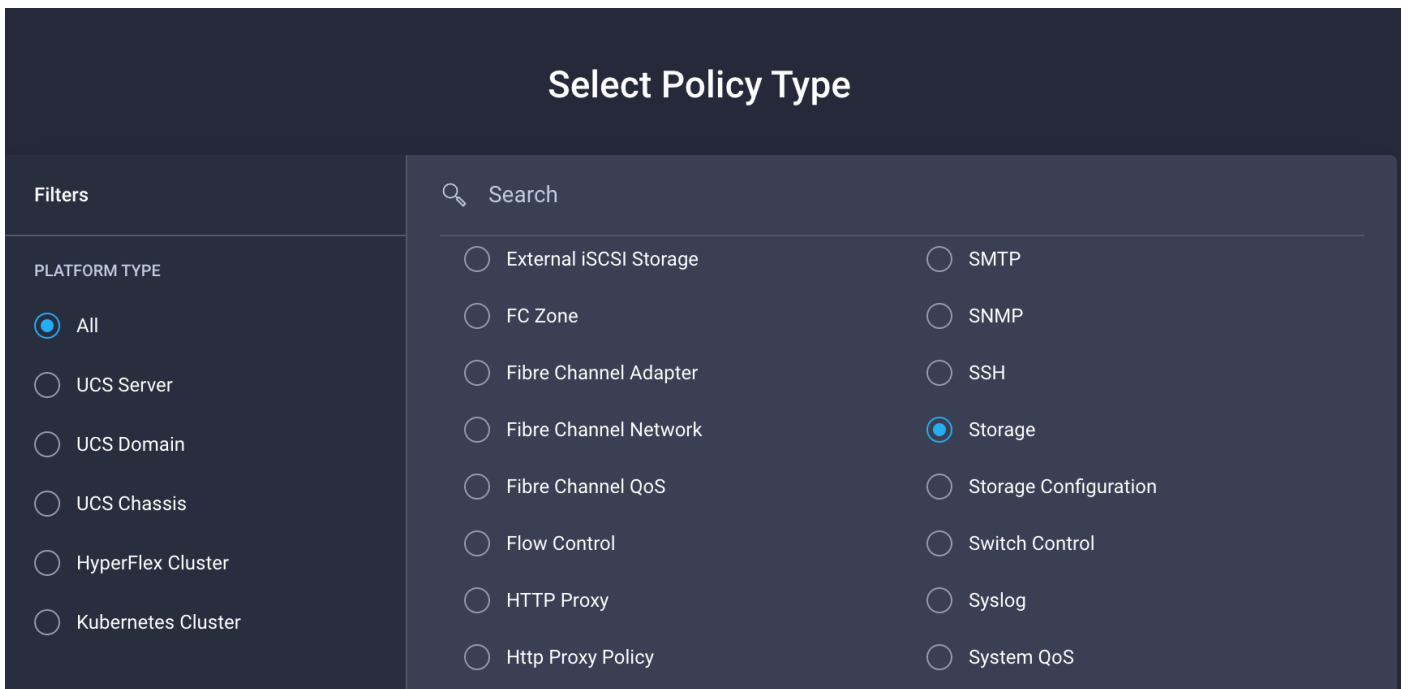
Step 2
Policy Details
Add policy details

Enable Virtual KVM ⓘ
 Max Sessions *
 4 1 - 4
 Remote Port *
 2068 1 - 65535
 Enable Video Encryption ⓘ
 Enable Local Server Video ⓘ
 Allow Tunneled vKVM ⓘ

Procedure 6. Create Storage Policy

Step 1. Go to Configure > Policies > Create Policy.

Step 2. Select policy type as Storage.



Select Policy Type

Filters

SEARCH Search

All
 UCS Server
 UCS Domain
 UCS Chassis
 HyperFlex Cluster
 Kubernetes Cluster

External iSCSI Storage
 FC Zone
 Fibre Channel Adapter
 Fibre Channel Network
 Fibre Channel QoS
 Flow Control
 HTTP Proxy
 Http Proxy Policy
 SMTP
 SNMP
 SSH
 Storage
 Storage Configuration
 Switch Control
 Syslog
 System QoS

Step 3. Enter name for the storage policy.



Step 1

General

Add a name, description and tag for the policy.

Organization *

CDIP-UCSC-M6



Name *

CDIP-DN-Storage

Set Tags

Description

<= 1024

Step 4. Enable JBOD drives for virtual drive creation, select state of the unused drive. Enable configuration for M.2 RAID configuration, MRAID/RAID Controller configuration or MRAID/RAID Single Drive RAID0 Configuration as applicable.

Step 5. Enable M.2 configuration and select Slot of the M.2 RAID controller for virtual drive creation as “MSTOR-RAID-1 (MSTOR-RAID)”

Step 6. Enter the details for data node/storage node configuration according to disk slot populated in the server. Please refer to the server inventory > storage controllers > RAID controller > Physical Drives for disk slot details.

Figure 48. Recommended virtual drive configuration for HDDs

The screenshot shows a configuration interface for virtual drive settings. At the top, there are navigation tabs: 'All Platforms' (selected), 'UCS Server (Standalone)', and 'UCS Server (FI-Attached)'. Below this, the 'General Configuration' section includes a toggle for 'Use JBOD drives for Virtual Drive creation' (checked) and a dropdown for 'Unused Disks State' set to 'No Change'. The 'M.2 RAID Configuration' section has a toggle for 'Enable' (checked) and a dropdown for 'Slot of the M.2 RAID controller for virtual drive creation' set to 'MSTOR-RAID-1 (MSTOR-RAID)'. The 'MRAID/RAID Controller Configuration' section has a toggle for 'Enable' (unchecked). The 'MRAID/RAID Single Drive RAID0 Configuration' section has a toggle for 'Enable' (checked). Below this, 'Drive Slots' is set to '1-24,101-104'. The 'Strip Size' is '1MiB', 'Access Policy' is 'Read Write', and 'Read Policy' is 'Always Read Ahead'. Finally, 'Write Policy' is 'Write Back Good BBU' and 'Disk Cache' is 'Disabled'. Each dropdown menu has a small circular icon to its right.

General Configuration

Use JBOD drives for Virtual Drive creation ⓘ

Unused Disks State
No Change ▼ ⓘ

M.2 RAID Configuration Enable

Slot of the M.2 RAID controller for virtual drive creation
MSTOR-RAID-1 (MSTOR-RAID) ▼ ⓘ

MRAID/RAID Controller Configuration Enable

MRAID/RAID Single Drive RAID0 Configuration Enable

Drive Slots
1-24,101-104 ⓘ

Strip Size: 1MiB ▼ ⓘ Access Policy: Read Write ▼ ⓘ Read Policy: Always Read Ahead ▼ ⓘ

Write Policy: Write Back Good BBU ▼ ⓘ Disk Cache: Disabled ▼ ⓘ

Figure 49. Recommended virtual drive configuration for SSDs

The screenshot shows the UCS Manager configuration interface for virtual drive configuration for SSDs. The page is dark-themed and has a navigation bar at the top with three tabs: "All Platforms" (selected), "UCS Server (Standalone)", and "UCS Server (FI-Attached)".

The configuration is organized into several sections:

- General Configuration:**
 - Use JBOD drives for Virtual Drive creation:** A toggle switch is turned on (green).
 - Unused Disks State:** A dropdown menu is set to "No Change".
- M.2 RAID Configuration:**
 - A toggle switch is turned on (green) and labeled "Enable".
 - Slot of the M.2 RAID controller for virtual drive creation:** A dropdown menu is set to "MSTOR-RAID-1 (MSTOR-RAID)".
- MRAID/RAID Controller Configuration:**
 - A toggle switch is turned off (grey) and labeled "Enable".
- MRAID/RAID Single Drive RAID0 Configuration:**
 - A toggle switch is turned on (green) and labeled "Enable".
- Drive Slots:** A dropdown menu is set to "1-24,101-104".
- Strip Size:** A dropdown menu is set to "64KiB".
- Access Policy:** A dropdown menu is set to "Read Write".
- Read Policy:** A dropdown menu is set to "No Read Ahead".
- Write Policy:** A dropdown menu is set to "Write Through".
- Disk Cache:** A dropdown menu is set to "Unchanged".

Step 7. Create storage policy for master/mgmt node.

M.2 RAID Configuration Enable

Slot of the M.2 RAID controller for virtual drive creation
 MSTOR-RAID-1 (MSTOR-RAID) ⊙

MRAID/RAID Controller Configuration Enable

Global Hot Spares ⊙

[Add Drive Group](#)

| <input type="checkbox"/> | Drive Group Name | RAID Level | Number of Spans | Dedicated Hot Spares | Drive Array Spans |
|--------------------------|------------------|------------|-----------------|----------------------|-----------------------------------------------|
| <input type="checkbox"/> | Mgmt-NN-R1 | RAID1 | | | { 1-24 } i |

[Add Virtual Drive](#)

| <input type="checkbox"/> | Virtual Drive Name | Drive Group | Size (MiB) | Expand to Available | Set as Boot Drive | |
|--------------------------|--------------------|-------------|------------|---------------------|-------------------|---|
| <input type="checkbox"/> | Mgmt-NN-R1 | Mgmt-NN-R1 | - | Yes | No | ⋮ |

Procedure 7. Create Ethernet Adapter Policy


Step 1. Go to Configure > Policies > Create Policy.

Step 2. Select policy type as Ethernet Adapter.

Step 3. Add policy details as follows:

- Interrupts - 11
- Receive Queue Count - 8
- Receive Ring Size - 4096
- Transmit Queue Count - 8
- Transmit Ring Size - 4096
- Completion Queue Count - 12

◦ Receive Side Scaling - Enabled



Step 2 Policy Details

Add policy details

[All Platforms](#) | [UCS Server \(Standalone\)](#) | [UCS Server \(FI-Attached\)](#)

- Enable Virtual Extensible LAN
- Enable Network Virtualization using Generic Routing Encapsulation
- Enable Accelerated Receive Flow Steering
- Enable Precision Time Protocol
- Enable Advanced Filter
- Enable Interrupt Scaling
- Enable GENEVE Offload

RoCE Settings

- Enable RDMA over Converged Ethernet

Interrupt Settings

| | | |
|------------|----------------|---------------------|
| Interrupts | Interrupt Mode | Interrupt Timer, us |
| 11 | MSIx | 125 |
| 1 - 1024 | | 0 - 65535 |

Interrupt Coalescing Type

Min

Receive

Receive Queue Count

8



1 - 1000

Receive Ring Size

4096



64 - 16384

Transmit

Transmit Queue Count

4



1 - 1000

Transmit Ring Size

4096



64 - 16384

Completion

Completion Queue Count

12



1 - 2000

Completion Ring Size

1



1 - 256

Uplink Failback Timeout (seconds)

5



0 - 600

Receive

Receive Queue Count

8



1 - 1000

Receive Ring Size

4096



64 - 16384

Transmit

Transmit Queue Count

4



1 - 1000

Transmit Ring Size

4096



64 - 16384

Completion

Completion Queue Count

12



1 - 2000

Completion Ring Size

1



1 - 256

Uplink Failback Timeout (seconds)

5



0 - 600

TCP Offload

Enable Tx Checksum Offload ⓘ

Enable Rx Checksum Offload ⓘ

Enable Large Send Offload ⓘ

Enable Large Receive Offload ⓘ

Receive Side Scaling

Enable Receive Side Scaling ⓘ

Enable IPv4 Hash ⓘ

Enable IPv6 Extensions Hash ⓘ

Enable IPv6 Hash ⓘ

Enable TCP and IPv4 Hash ⓘ

Enable TCP and IPv6 Extensions Hash ⓘ

Enable TCP and IPv6 Hash ⓘ

Enable UDP and IPv4 Hash ⓘ

Enable UDP and IPv6 Hash ⓘ

Procedure 8. Create LAN Connect Policy

Step 1. Go to Configure > Policies > Create Policy.

Step 2. Select policy type as LAN Connectivity.

The screenshot shows a dark-themed interface titled "Select Policy Type". On the left, there is a "Filters" section with a "PLATFORM TYPE" category. Under this category, several radio buttons are listed: "All", "UCS Server" (which is selected), "UCS Domain", "UCS Chassis", "HyperFlex Cluster", and "Kubernetes Cluster". To the right of the filters is a search bar and a grid of 30 radio button options. The "LAN Connectivity" option is selected and highlighted with a blue background. The other options include Adapter Configuration, BIOS, Boot Order, Certificate Management, Device Connector, Ethernet Adapter, Ethernet Network, Ethernet Network Control, Ethernet Network Group, Ethernet QoS, FC Zone, Fibre Channel Adapter, Fibre Channel Network, Fibre Channel QoS, IMC Access, IPMI Over LAN, iSCSI Adapter, iSCSI Boot, iSCSI Static Target, LDAP, Local User, Network Connectivity, NTP, Persistent Memory, Power, SAN Connectivity, SD Card, Serial Over LAN, SMTP, SNMP, SSH, Storage, Syslog, Virtual KVM, and Virtual Media. At the bottom right of the interface, there is a blue "Start" button.

Step 3. Enter policy name and select Target Platform as UCS Server (FI-Attached).



Step 1

General

Add a name, description and tag for the policy.

Organization *

CDIP-UCSC-M6



Name *

K13-LanConnect

Target Platform ⓘ

UCS Server (Standalone) UCS Server (FI-Attached)

Set Tags

Description

≤ 1024

Step 4. Click Add vNIC.

Step 2
Policy Details
Add policy details

Enable Azure Stack Host QoS

IQN

None | Pool | Static

This option ensures the IQN name is not associated with the policy

vNIC Configuration

Manual vNICs Placement | Auto vNICs Placement

For manual placement option you need to specify placement for each vNIC. Learn more at [Help Center](#)

Add vNIC | Graphic vNICs Editor

0 items found | 25 per page | 0 of 0


Add Filter

| | Name | Slot ID | Switch ID | PCI Order | Failover | Pin Group | MAC Pool |
|--------------------|------|---------|-----------|-----------|----------|-----------|----------|
| NO ITEMS AVAILABLE | | | | | | | |

Step 5. Enter or select an existing policy for vNIC creation (the screenshot shows placement with mLOM Cisco UCS VIC 1467):

- vNIC name
- select MAC Pool
- Placement
- Consistent Device Naming (CDN)
- Failover – Enabled
- Ethernet Network Group Policy
- Ethernet Network Control Policy
- Ethernet QoS

◦ Ethernet Adapter



Edit vNIC

General

Name * ⊙ Pin Group Name ⌵ ⊙

MAC

Pool Static

MAC Pool * ⊙

Selected Pool 👁 ✕

Placement

Simple Advanced

Slot ID * ⊙ PCI Link ⌵ ⊙
0 - 1

Switch ID * ⌵ ⊙

PCI Order ⌵ ⊙

Consistent Device Naming (CDN)

Source
vNIC Name ▼ ⓘ

Failover

Enabled ⓘ

Ethernet Network Group Policy * ⓘ
Selected Policy CDIP-K13-EthNWGrp 👁 | ✕

Ethernet Network Control Policy * ⓘ
Selected Policy CDIP-K13-EthNWCtrl 👁 | ✕

Ethernet QoS * ⓘ
Selected Policy CDIP-K13-EthQoS 👁 | ✕

Ethernet Adapter * ⓘ
Selected Policy CDIP-K13-EthAdapter 👁 | ✕

iSCSI Boot ⓘ
[Select Policy](#) 📄

Connection

Disabled ⓘ

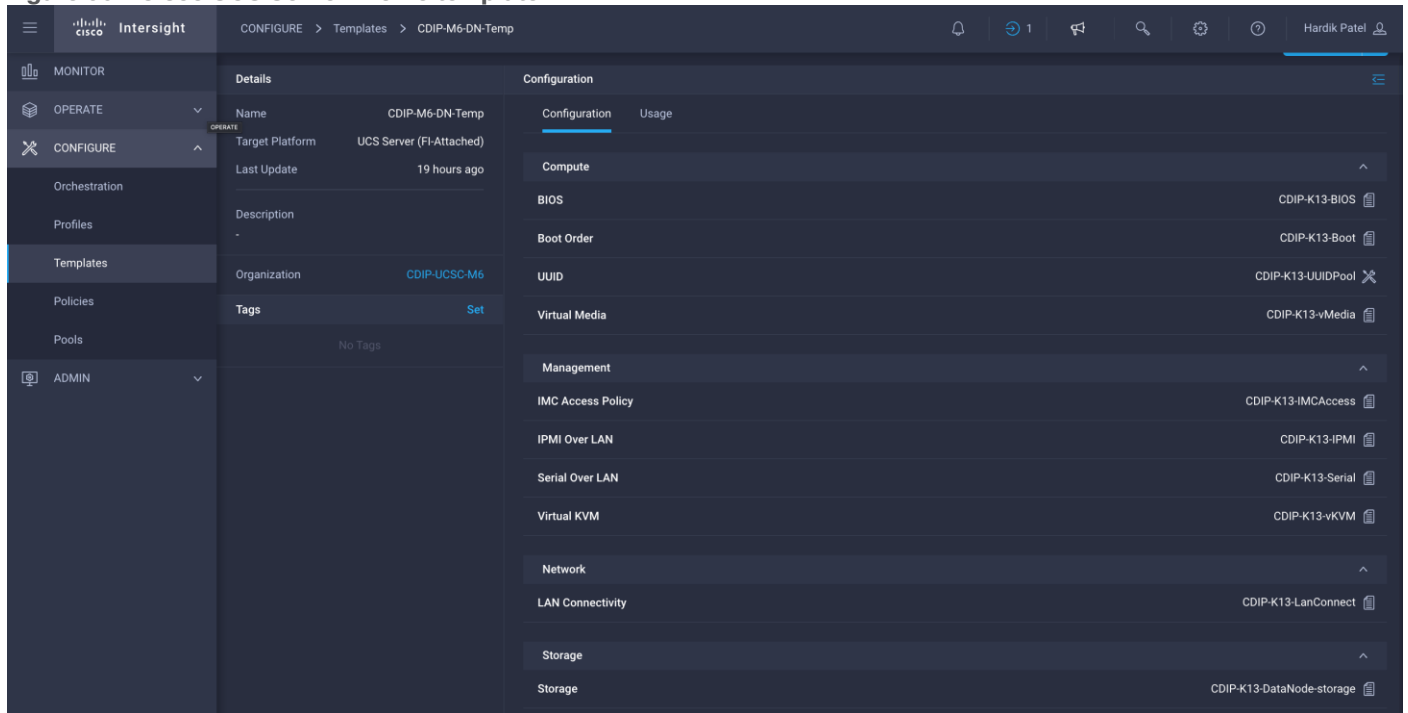
usNIC

VMQ

Server Profile Template

A server profile template enables resource management by simplifying policy alignment and server configuration. A server profile template is created using the server profile template wizard. The server profile template wizard groups the server policies into the following four categories to provide a quick summary view of the policies that are attached to a profile:

Figure 50. Cisco UCS Server Profile template



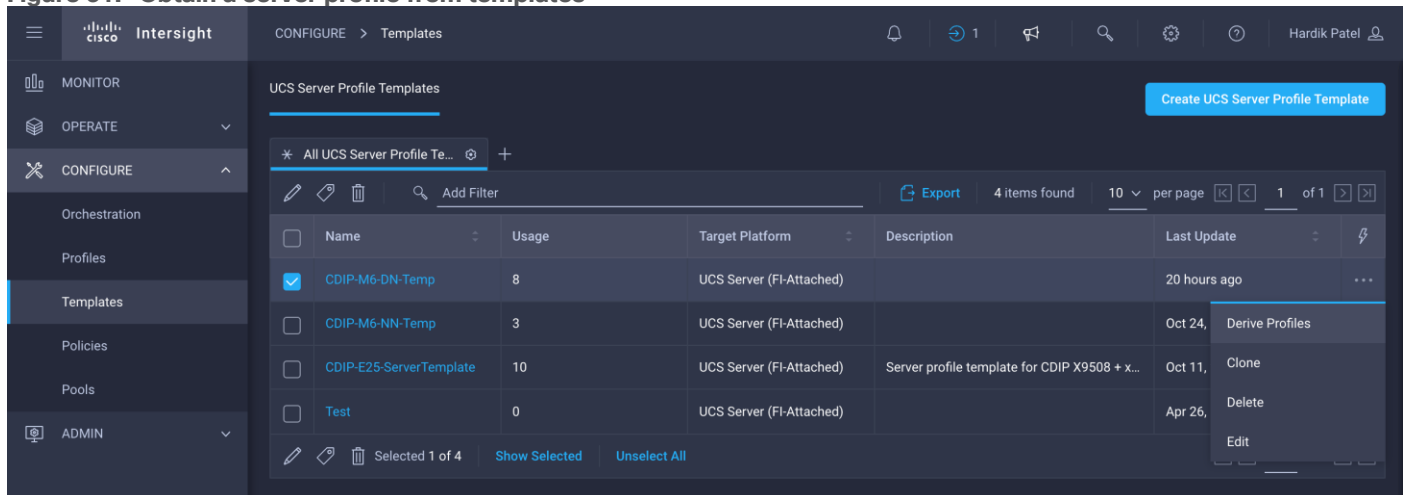
Obtain and Deploy Server Profiles from the Cisco Intersight Server Profile Template

The Cisco Intersight server profile allows server configurations to be deployed directly on the server based on policies defined in the server profile template. After a server profile template has been successfully created, server profiles can be derived from the template and associated with the Cisco UCS Servers as shown in [Figure 51](#).

Procedure 1. Obtain and deploy the server profiles

Step 1. Go to Configure > Templates > Select existing Server Profile Template. Click Derive Profiles.

Figure 51. Obtain a server profile from templates



Step 2. Select Server Assignment to derive profiles from template. Select Assign Now, Assign Server from a Resource Pool or Assign Later.

Progress

- 1 General
- 2 Details
- 3 Summary



Step 1
General

Select the server(s) that need to be assigned to profile(s) or specify the number of profiles that you want to derive and assign the servers later.

UCS Server Profile Template

Name **CDIP-M6-DN-Temp** Organization **CDIP-UCSC-M6**
Target Platform **UCS Server (FI-Attached)**

Server Assignment

Assign Now Assign Server from a Resource Pool Assign Later

11 items found 25 per page 1 of 1

Add Filter

| <input type="checkbox"/> | Name | User Label | Health | Model | UCS Domain | Serial Nu... |
|-------------------------------------|------------------|------------|---------|-----------------|----------------|--------------|
| <input checked="" type="checkbox"/> | K13-CDIP-6410... | cdip-dn01 | Healthy | UCSC-C240-M6... | K13-CDIP-64108 | WZP26220Q37 |
| <input checked="" type="checkbox"/> | K13-CDIP-6410... | cdip-dn02 | Healthy | UCSC-C240-M6... | K13-CDIP-64108 | WZP26200FV5 |

Cancel

Next >



Step 2

Details

Edit the description, tags, and auto-generated names of the profiles.

General

Organization *

CDIP-UCSC-M6

Target Platform

UCS Server (FI-Attached)

Description

<= 1024

Set Tags

Derive

Profile Name Prefix

CDIP-M6-Datanode

Digits Count

1



>= 1

Start Index for Suffix

1



>= 0

1 Name *

CDIP-M6-Datanode1

2 Name *

CDIP-M6-Datanode2

3 Name *

CDIP-M6-Datanode3

4 Name *

CDIP-M6-Datanode4

Step 3. Review Derive Profile from the template. Click Derive.

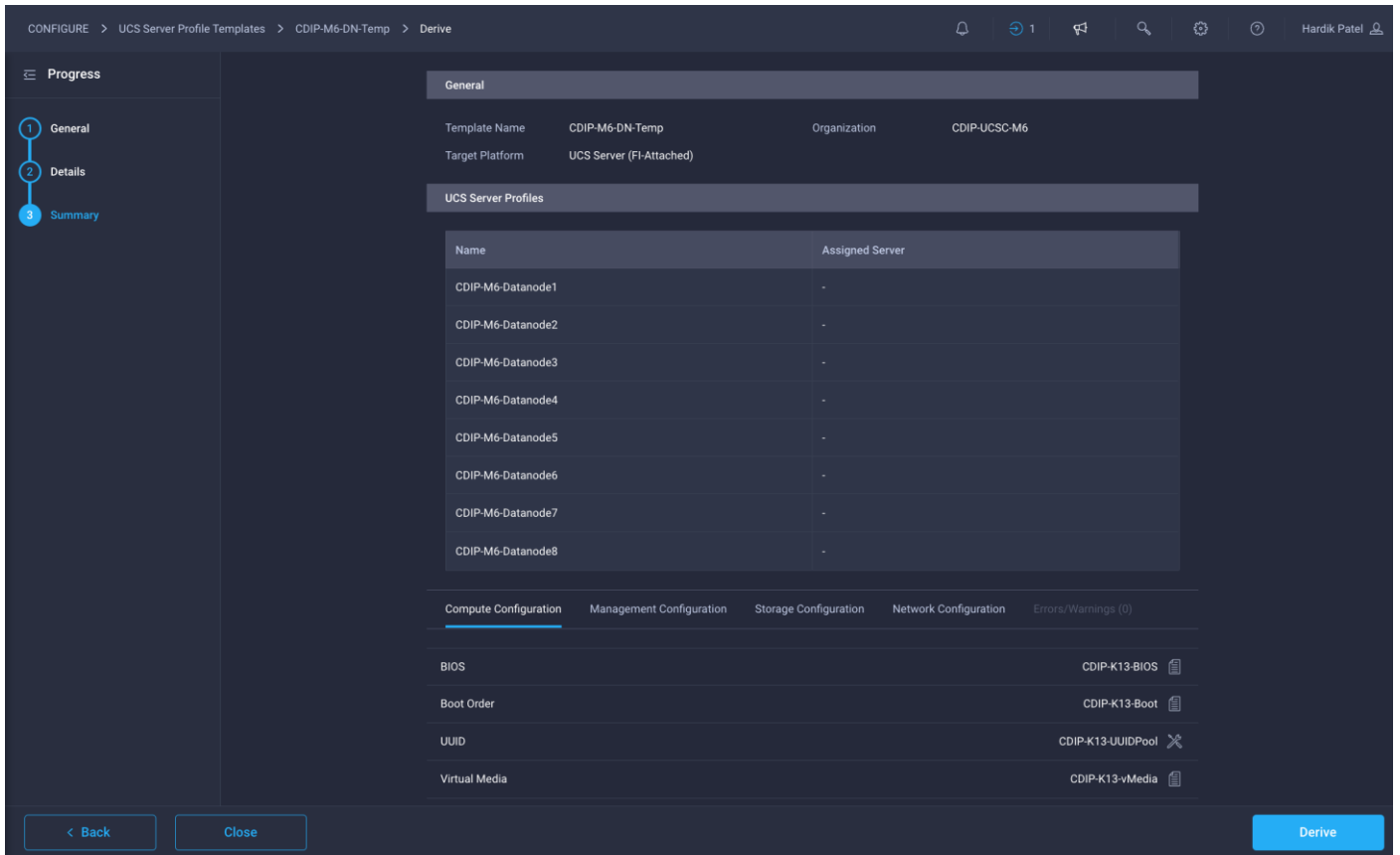
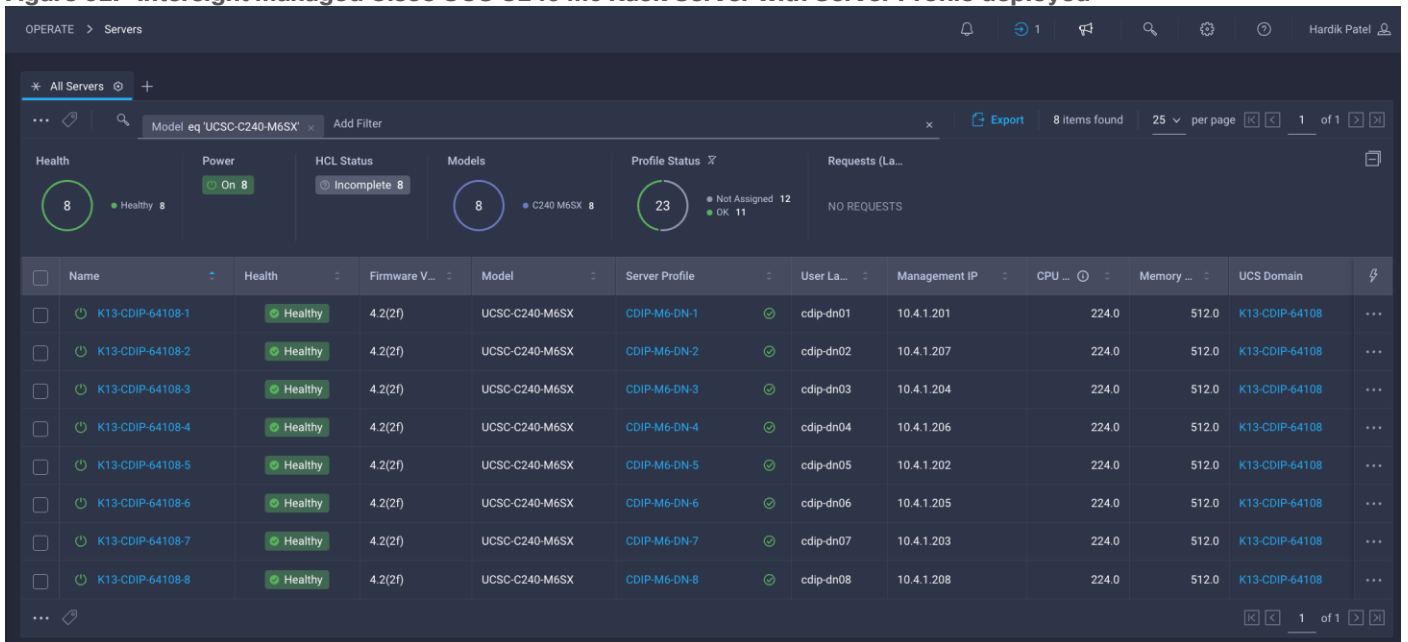


Figure 52. Intersight Managed Cisco UCS C240 M6 Rack Server with Server Profile deployed



Install Red Hat Enterprise Linux 8.6

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.

This chapter contains the following:

- [Install Red Hat Enterprise Linux \(RHEL\) 8.6](#)
- [Post OS Install](#)

Note: In this study, Red Hat Enterprise Linux version 8.6 DVD/ISO was utilized for OS the installation via CIMC mapped vMedia on Cisco UCS C240 M6 Rack Server.

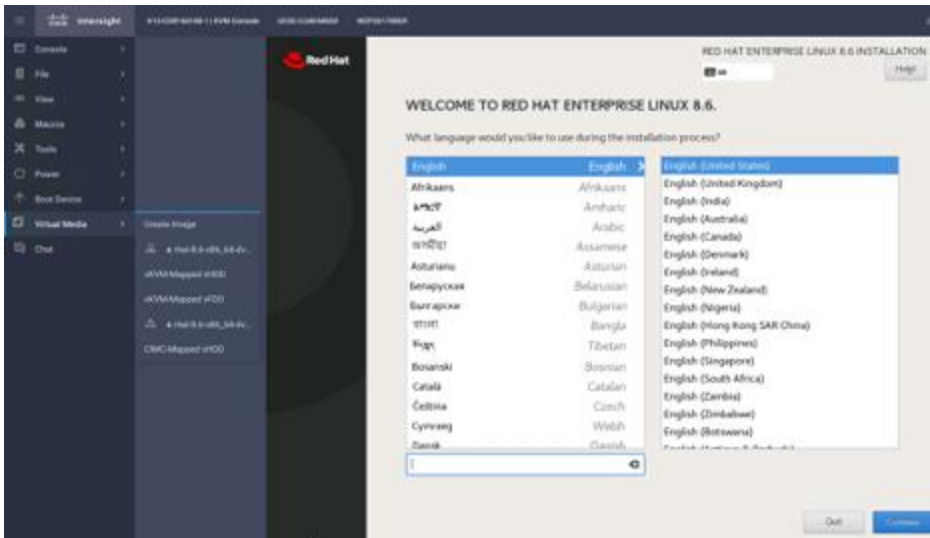
Procedure 1. Install Red Hat Enterprise Linux (RHEL) 8.6

Step 1. Log into the Cisco Intersight.

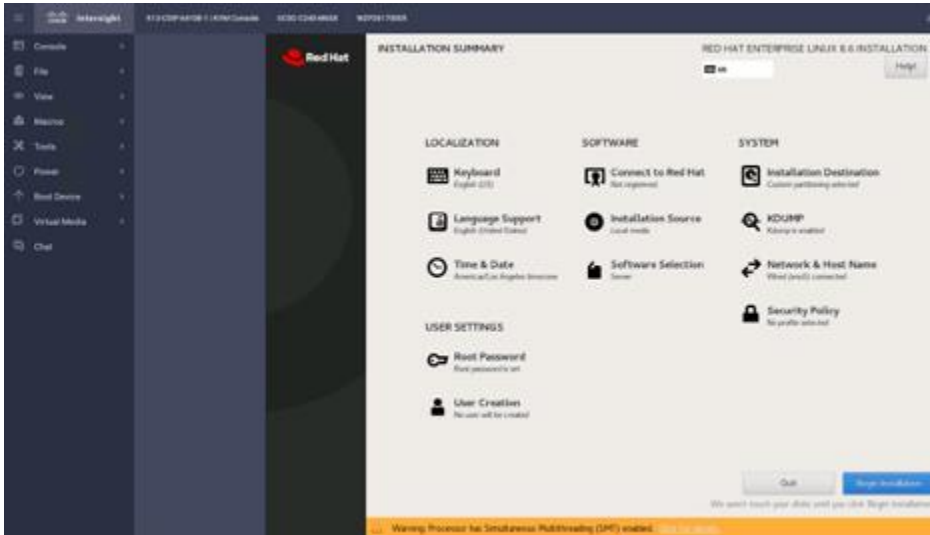
Step 2. Go to Operate > Servers > Click the ellipses and select Launch vKVM or Tunneled vKVM.

| Name | Health | Firmware V... | Model | Server Profile | User LA... | Management IP | CPU ... | Memory ... | UCS Domain | Actions |
|------------------|---------|---------------|----------------|----------------|------------|---------------|---------|------------|----------------|--------------------------|
| K13-CDIP-64108-1 | Healthy | 4.2(2f) | UCSC-C240-M6SX | CDIP-M6-DN-1 | cdip-dn01 | 10.4.1.201 | 224.0 | 512.0 | K13-CDIP-64108 | Power |
| K13-CDIP-64108-2 | Healthy | 4.2(2f) | UCSC-C240-M6SX | CDIP-M6-DN-2 | cdip-dn02 | 10.4.1.207 | 224.0 | 512.0 | K13-CDIP-64108 | System |
| K13-CDIP-64108-3 | Healthy | 4.2(2f) | UCSC-C240-M6SX | CDIP-M6-DN-3 | cdip-dn03 | 10.4.1.204 | 224.0 | 512.0 | K13-CDIP-64108 | Profile |
| K13-CDIP-64108-4 | Healthy | 4.2(2f) | UCSC-C240-M6SX | CDIP-M6-DN-4 | cdip-dn04 | 10.4.1.206 | 224.0 | 512.0 | K13-CDIP-64108 | Install Operating System |
| K13-CDIP-64108-5 | Healthy | 4.2(2f) | UCSC-C240-M6SX | CDIP-M6-DN-5 | cdip-dn05 | 10.4.1.202 | 224.0 | 512.0 | K13-CDIP-64108 | Upgrade Firmware |
| K13-CDIP-64108-6 | Healthy | 4.2(2f) | UCSC-C240-M6SX | CDIP-M6-DN-6 | cdip-dn06 | 10.4.1.205 | 224.0 | 512.0 | K13-CDIP-64108 | Launch vKVM |
| K13-CDIP-64108-7 | Healthy | 4.2(2f) | UCSC-C240-M6SX | CDIP-M6-DN-7 | cdip-dn07 | 10.4.1.203 | 224.0 | 512.0 | K13-CDIP-64108 | Launch Tunneled vKVM |
| K13-CDIP-64108-8 | Healthy | 4.2(2f) | UCSC-C240-M6SX | CDIP-M6-DN-8 | cdip-dn08 | 10.4.1.208 | 224.0 | 512.0 | K13-CDIP-64108 | Open TAC Case |

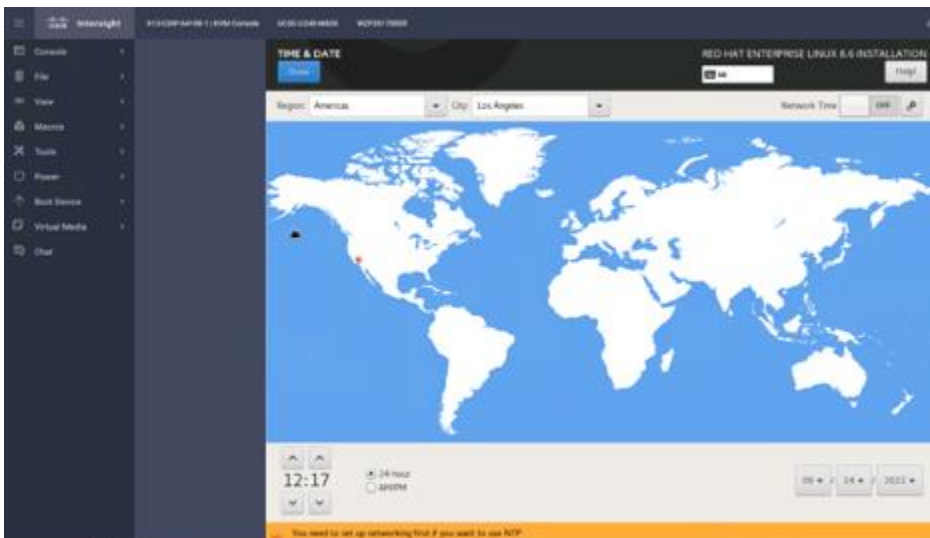
Step 3. From the virtual KVM console check virtual media tab for the image in use. Click Continue on the Welcome screen for RHEL 8.6 installation.



Step 4. Select Time & Date.

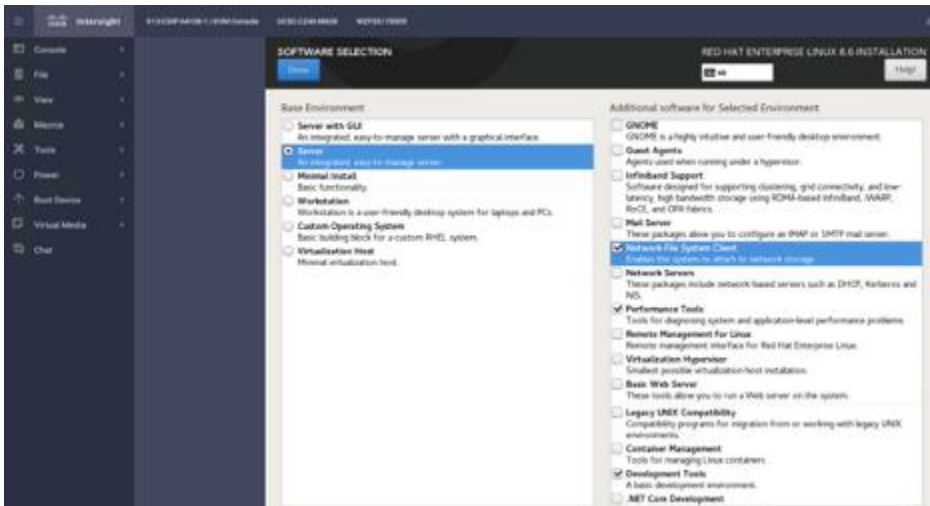


Step 5. Select Region and City.

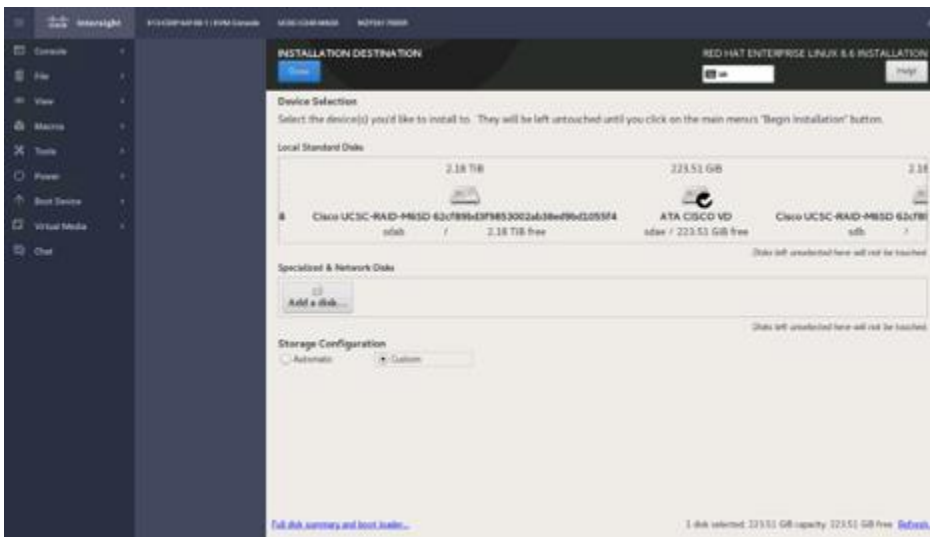


Step 6. Select Software Selection. Select “Server for the Bare Environment” and add the required software:

- Network File System Client
- Performance Tools
- Development Tools
- Security Tools
- System Tools

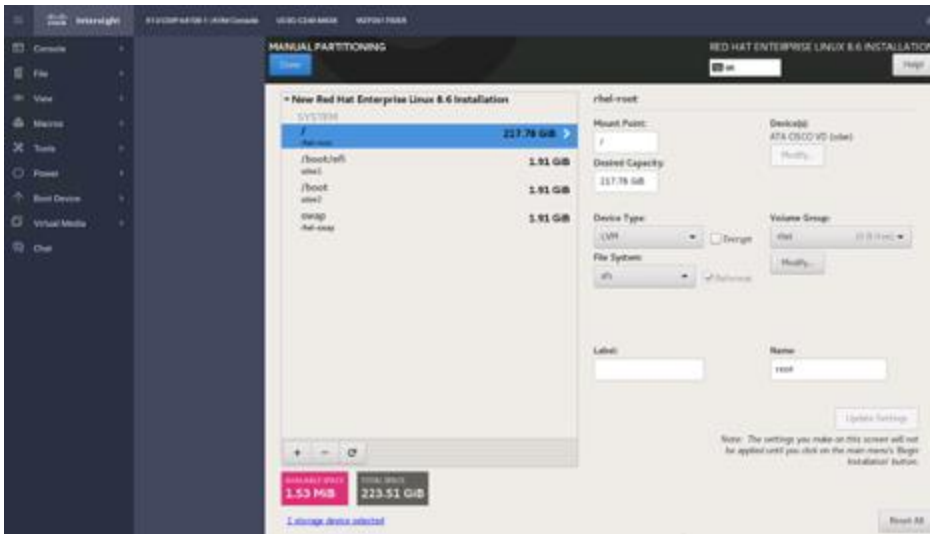


Step 7. Click Installation Destination > Select storage device ATA CISCO VD (M.2 Hardware RAID controller provisioned RAID 1 virtual disk). Select Custom storage configuration. Click Done.

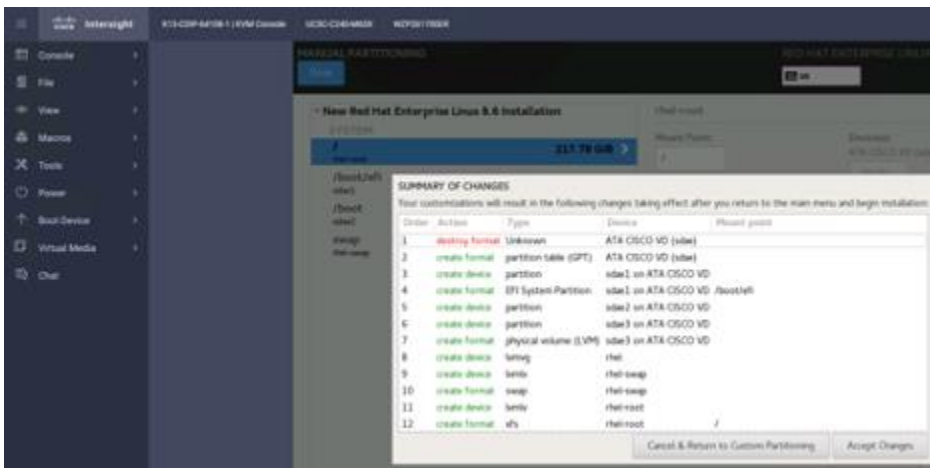


Step 8. Click the + sign to add new mount point. Click Done after creating the new mount points as follows:

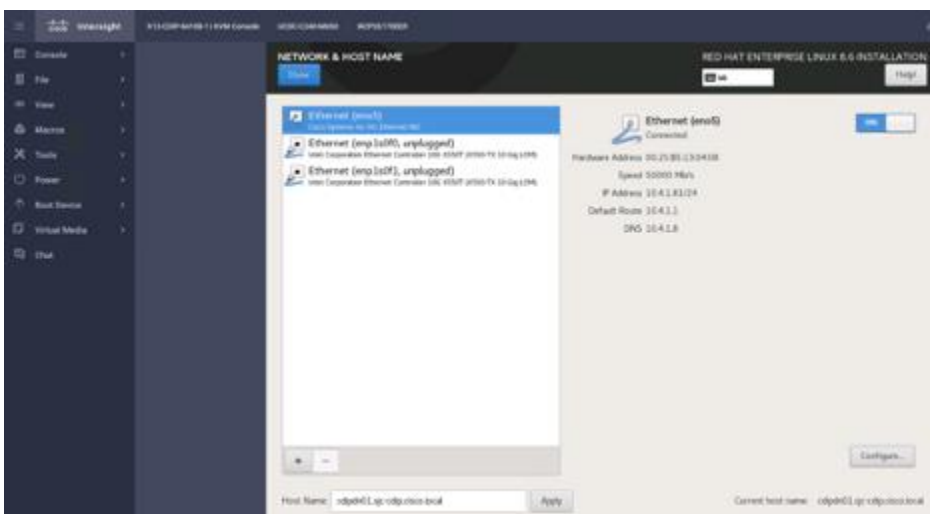
- /boot/efi - capacity 2048mb
- /boot - capacity 2048mb
- Swap - capacity 2048mb
- / - capacity blank (which will allocates remaining capacity)



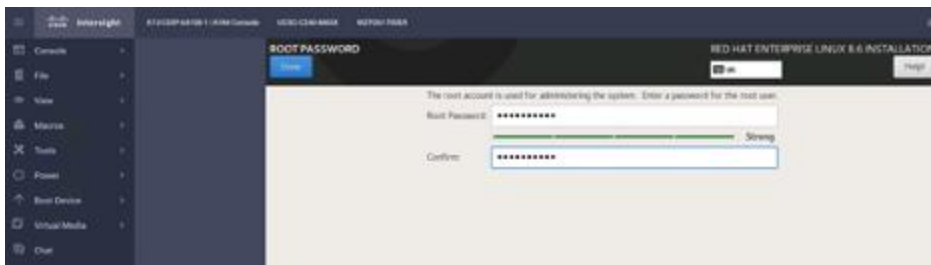
Step 9. Click Accept Changes.



Step 10. Select Network & Host Name. Enter host name and configure network adapter with static IP address.



Step 11. Select Root Password. Enter the root password and confirm.



Step 12. Click Begin Installation.

Step 13. Reboot after successful OS installation.

Post OS Installation

Choose one of the nodes of the cluster or a separate node as the Admin Node for management, such as CDP DC installation, Ansible, creating a local Red Hat repo, and others. In this document, we configured cdipnn01 for this purpose.

Procedure 1. Configure /etc/hosts

Step 1. Setup /etc/hosts on the Admin node; this is a pre-configuration to setup DNS as shown in the next section.

Note: For the purpose of simplicity, /etc/hosts file is configured with hostnames in all the nodes. However, in large scale production grade deployment, DNS server setup is highly recommended.

Step 2. To create the host file on the admin node, follow these steps:

Step 3. Log into the Admin Node (cdipnn01).

```
# ssh 10.4.1.90
```

Step 4. Populate the host file with IP addresses and corresponding hostnames on the Admin node (cdipnn01) and other nodes as follows:

```
vi /etc/hosts
10.4.1.81      cdipdn01.sjc-cdip.cisco.local  cdipdn01
10.4.1.82      cdipdn02.sjc-cdip.cisco.local  cdipdn02
10.4.1.83      cdipdn03.sjc-cdip.cisco.local  cdipdn03
10.4.1.84      cdipdn04.sjc-cdip.cisco.local  cdipdn04
10.4.1.85      cdipdn05.sjc-cdip.cisco.local  cdipdn05
10.4.1.86      cdipdn06.sjc-cdip.cisco.local  cdipdn06
10.4.1.87      cdipdn07.sjc-cdip.cisco.local  cdipdn07
10.4.1.88      cdipdn08.sjc-cdip.cisco.local  cdipdn08
10.4.1.90      cdipnn01.sjc-cdip.cisco.local  cdipnn01
10.4.1.89      cdipnn02.sjc-cdip.cisco.local  cdipnn02
10.4.1.91      cdipnn03.sjc-cdip.cisco.local  cdipnn03
```

Procedure 2. Set Up Password-less 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.

Enable the passwordless login across all the nodes when Red Hat Linux is installed across all the nodes in the cluster.

Step 1. Log into the Admin Node (cdipnn01).

```
# ssh 10.4.1.90
```

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

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

Step 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..03}; do echo "copying cdipnn$i.sjc-cdip.cisco.local"; ssh-copy-id -i ~/.ssh/id_rsa.pub
root@cdipnn$i.sjc-cdip.cisco.local; done;
# for i in {01..09}; do echo "copying cdipdn$i.sjc-cdip.cisco.local"; ssh-copy-id -i ~/.ssh/id_rsa.pub
root@cdipdn$i.sjc-cdip.cisco.local; done;
```

Step 4. Enter yes for Are you sure you want to continue connecting (yes/no)?

Step 5. Enter the password of the remote host.

Procedure 3. Create a Red Hat Enterprise Linux (RHEL) 8.6 Local Repository

To create a repository using RHEL DVD or ISO on the admin node, create a directory with all the required RPMs, run the “createrepo” command and then publish the resulting repository.

Note: Based on this repository file, yum requires httpd to be running on rhelnn01 for other nodes to access the repository.

Note: This step is required to install software on Admin Node (rhelnn01) using the repo (such as httpd, create-repo, and so on.)

Step 1. Log into cdipnn01.

Step 2. Copy RHEL 8.6 iso from remote repository

```
# scp rhel-8.6-x86_64-dvd.iso cdipnn01:/root/
```

Step 3. Create a directory that would contain the repository.

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

Step 4. Create mount point to mount RHEL ISO

```
# mkdir -p /mnt/rheliso
# mount -t iso9660 -o loop /root/rhel-8.4-x86_64-dvd.iso /mnt/rheliso/
```

Step 5. Copy the contents of the RHEL 8.6 ISO to /var/www/html/rhelrepo

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

Step 6. Create a .repo file to enable the use of the yum command on cdipnn01

```
# vi /var/www/html/rhelrepo/rheliso.repo
[rhel8.6]
name= Red Hat Enterprise Linux 8.6
baseurl=http://10.4.1.90/rhelrepo
gpgcheck=0
enabled=1
```

Step 7. Copy the rheliso.repo file from /var/www/html/rhelrepo to /etc/yum.repos.d on cdipnn01.

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

Step 8. To make use of repository files on rhelnn01 without httpd, edit the baseurl of repo file /etc/yum.repos.d/rheliso.repo to point repository location in the file system.

```
# vi /etc/yum.repos.d/rheliso.repo
[rhel8.6]
name=Red Hat Enterprise Linux 8.6
baseurl=file:///var/www/html/rhelrepo
gpgcheck=0
enabled=1
```

Procedure 4. Create the Red Hat Repository Database

Step 1. Install the createrepo package on admin node (rhelnn01). Use it to regenerate the repository database(s) for the local copy of the RHEL DVD contents.

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

Step 2. Run createrepo on the RHEL repository to create the repo database on admin node.

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

Procedure 5. Set up Ansible

Step 1. Install ansible-core

```
# yum install -y ansible-core
# ansible --version
ansible [core 2.12.2]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.8/site-packages/ansible
  ansible collection location = /root/.ansible/ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.8.12 (default, Sep 16 2021, 10:46:05) [GCC 8.5.0 20210514 (Red Hat 8.5.0-3)]
  jinja version = 2.10.3
  libyaml = True
```

Step 2. 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.

```
# vi /etc/ansible/hosts

[admin]
cdipnn01.sjc-cdip.cisco.local

[namenodes]
cdipnn01.sjc-cdip.cisco.local
cdipnn02.sjc-cdip.cisco.local
cdipnn03.sjc-cdip.cisco.local

[datanodes]
cdipdn01.sjc-cdip.cisco.local
cdipdn02.sjc-cdip.cisco.local
cdipdn03.sjc-cdip.cisco.local
cdipdn04.sjc-cdip.cisco.local
cdipdn05.sjc-cdip.cisco.local
cdipdn06.sjc-cdip.cisco.local
cdipdn07.sjc-cdip.cisco.local
cdipdn08.sjc-cdip.cisco.local

[nodes]
cdipnn01.sjc-cdip.cisco.local
cdipnn02.sjc-cdip.cisco.local
cdipnn03.sjc-cdip.cisco.local
cdipdn01.sjc-cdip.cisco.local
cdipdn02.sjc-cdip.cisco.local
cdipdn03.sjc-cdip.cisco.local
cdipdn04.sjc-cdip.cisco.local
cdipdn05.sjc-cdip.cisco.local
cdipdn06.sjc-cdip.cisco.local
cdipdn07.sjc-cdip.cisco.local
cdipdn08.sjc-cdip.cisco.local
```

Step 3. Verify host group by running the following commands.

```
# ansible nodes -m ping
```

Procedure 6. Install httpd

Setting up the RHEL repository on the admin node requires httpd.

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

Note: 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
```

Step 2. Edit httpd.conf file; add ServerName and make the necessary changes to the server configuration file:

```
# vi /etc/httpd/conf/httpd.conf
ServerName 10.4.1.90:80
```

Step 3. Start httpd service.

```
# systemctl start httpd
# systemctl enable httpd
# chkconfig httpd on
```

Procedure 7. Disable the Linux Firewall

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

```
# 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"
# ansible all -m command -a "chkconfig firewalld off"
```

Procedure 8. Set Up All Nodes to use the RHEL Repository

Step 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/."
```

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

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

Step 3. Purge the yum caches:

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

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

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

Procedure 9. Disable SELinux

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

Step 1. SELinux can be disabled by editing /etc/selinux/config and changing the SELINUX line to SELINUX=disabled. To disable SELinux, follow these steps:

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

Note: This command may fail if SELinux is already disabled. This requires reboot to take effect.

Step 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"
```

Procedure 10. Upgrade Cisco UCS VIC Driver

The latest Cisco Network driver is required for performance and updates. The latest drivers can be downloaded from the link: [https://software.cisco.com/download/home/283862063/type/283853158/release/4.2\(2d\)](https://software.cisco.com/download/home/283862063/type/283853158/release/4.2(2d))

In the ISO image, the required driver can be located at \Network\Cisco\VIC\RHEL\RHEL8.6\kmod-enic-4.2.0.28-877.22.rhel8u6.x86_64.rpm

To upgrade the Cisco Network Driver for VIC1457, follow these steps:

Step 1. From a node connected to the Internet, download, extract, and transfer kmod-enic-.rpm to rhelnn01 (admin node).

Step 2. 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 cdipnn01:

```
# ansible all -m copy -a "src=/root/kmod-enic-4.2.0.28-877.22.rhel8u6.x86_64.rpm dest=/root/."
```

Step 3. Use the yum module to install the enic driver rpm file on all the nodes through Ansible:

```
# ansible all -m yum -a "name=/root/kmod-enic-4.2.0.28-877.22.rhel8u6.x86_64.rpm state=present"
```

Step 4. 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:

```
# ansible all -m shell -a "modinfo enic | head -5"
cdipdn02.sjc-cdip.cisco.local | CHANGED | rc=0 >>
filename:      /lib/modules/4.18.0-372.9.1.el8.x86_64/extra/enic/enic.ko
version:       4.2.0.28-877.22
retpoline:     Y
license:       GPL v2
author:        Scott Feldman <scofeldm@cisco.com>
```

Procedure 11. Setup JAVA

Note: Please review JAVA requirement in CDP Private Cloud Base Requirements and Supported Versions sections: <https://docs.cloudera.com/cdp-private-cloud-upgrade/latest/release-guide/topics/cdpdc-java-requirements.html>

Download JDK 11 and copy the rpm to admin node:

<https://www.oracle.com/java/technologies/downloads/#license-lightbox>

Step 1. Copy JDK rpm to all nodes:

```
# ansible nodes -m copy -a "src=/root/jdk-11.0.10_linux-x64_bin.rpm dest=/root/."
```

Step 2. Extract and Install JDK all nodes:

```
# ansible all -m command -a "rpm -ivh jdk-11.0.10_linux-x64_bin.rpm"
```

Step 3. Create the following files java-set-alternatives.sh and java-home.sh on admin node.

```
# 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 $item /usr/java/jdk-11.0.16/bin/$item 9
alternatives --set $item /usr/java/jdk-11.0.16/bin/$item
done
# vi java-home.sh
export JAVA_HOME=/usr/java/jdk-11.0.16
```

Step 4. Make the two java scripts created above executable:

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

Step 5. 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"
```

Step 6. Setup Java Alternatives:

```
# ansible all -m shell -a "/root/java-set-alternatives.sh"
```

Step 7. 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"
```

Step 8. Setup JAVA_HOME on all nodes.

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

Step 9. Display JAVA_HOME on all nodes.

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

Step 10. Display current java -version.

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

```
# java -version
java version "11.0.16" 2022-07-19 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.16+11-LTS-199)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.16+11-LTS-199, mixed mode)
# echo $JAVA_HOME
/usr/java/jdk-11.0.16
```

Procedure 12. 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.

Step 1. 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"
```

Procedure 13. Set 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.

Step 1. For setting the ulimit on Red Hat, edit /etc/security/limits.conf on admin node rhelnn01 and add the following lines:

```
# vi /etc/security/limits.conf
* soft nfile 1048576
* hard nfile 1048576
```

Step 2. Copy the /etc/security/limits.conf file from admin node (rhelnn01) to all the nodes using the following command:

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

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

```
##PAM-1.0
auth        required      pam_env.so
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
```

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

Procedure 14. Set TCP Retries

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

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

Step 1. Edit the file `/etc/sysctl.conf` and on admin node `rhelnn01` and add the following lines:

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

Step 2. Copy the `/etc/sysctl.conf` file from admin node to all the nodes using the following command:

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

Step 3. Load the settings from default `sysctl` file `/etc/sysctl.conf` by running the following command:

```
# ansible nodes -m command -a "sysctl -p"Start and enable xinetd, dhcp and vsftpd service.
```

Procedure 15. Disable IPv6 Defaults

Step 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"
```

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

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

Procedure 16. Disable Swapping

Step 1. Run the following on all nodes.

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

Step 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"
```

Procedure 17. Disable Memory Overcommit

Step 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"
```

Step 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"
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.tcp_retries2=5
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
vm.swappiness=0
vm.overcommit_memory=0
```

Procedure 18. Disable Transparent Huge Pages

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

Step 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"
```

Step 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
```

Step 3. Copy file to each node:

```
# ansible nodes -m copy -a "src=/root/thp_disable dest=/root/thp_disable"
Append the content of file thp_disable to /etc/rc.d/rc.local:
# ansible nodes -m shell -a "cat /root/thp_disable >> /etc/rc.d/rc.local"
# ansible nodes -m shell -a "chmod +x /etc/rc.d/rc.local"
```

Procedure 19. Configure Chrony

Step 1. edit `/etc/chrony.conf` file.

```
# vi /etc/chrony.conf
server 10.4.1.7 iburst
driftfile /var/lib/chrony/drift
makestep 1.0 3
rtcsync
allow 10.4.1.0/24
keyfile /etc/chrony.keys
leapsectz right/UTC
logdir /var/log/chrony
```

Step 2. Copy `chrony.conf` file from the admin node to the `/etc` of all nodes by running command below:

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

Step 3. Start Chrony service.

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

Procedure 20. Install Megaraid StorCLI

This procedure 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>

Step 1. Download StorCLI: <https://www.broadcom.com/support/download-search/?pg=&pf=&pn=&po=&pa=&dk=storcli>.

Step 2. Extract the .zip file and copy storcli-007.2203.0000.0000-1.noarch.rpm from the linux directory.

Step 3. Download StorCLI and its dependencies and transfer to Admin node:

```
# scp storcli-007.2203.0000.0000-1.noarch.rpmsrhelnn01:/root/
```

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

```
# ansible all -m copy -a "src=/root/storcli-007.2203.0000.0000-1.noarch.rpm dest=/root/."
```

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

```
# ansible all -m shell -a "rpm -ivh storcli-007.2203.0000.0000-1.noarch.rpm"
```

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

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

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

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

Note: The Cisco UCS Intersight Storage policy configuration explains the steps to deploy the required storage configuration attached to Server Profile(s).

Procedure 21. Configure FileSystem for Name Nodes and Data Nodes

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.

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

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

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

Note: This script assumes there are no partitions already existing on the data volumes. If there are partitions, delete them before running the script. This process is in section Delete Partitions.

Note: Cloudera recommends two NVMe drives for the Ozone master nodes and Ozone data nodes in Raid 1 but in case of SSDs are installed for Ozone metadata which will require the run partition script below with edits so that Raid 1 based virtual drive volume created out of two SSDs can be presented separately as /ozone/metadata partition for example.

```
#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

# vi driveconfig_nvme.sh
#!/bin/bash
echo "Formatting and Mounting Drive => /dev/md126"
/sbin/mkfs.xfs -f /dev/md126
(( $? )) && continue

#Identify UUID
UUID=`blkid /dev/md126 | cut -d " " -f2 | cut -d "=" -f2 | sed 's//g'`

echo "Make Directory /ozone/metata"
/bin/mkdir -p /ozone/metata
(( $? )) && continue

echo "UUID of /dev/md126 = ${UUID}, mounting md126 using UUID on /ozone/metadadata"
/bin/mount -t xfs -o inode64,noatime -U ${UUID} /temp/nvme1
(( $? )) && continue

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

done

```

Step 2. Run the following command to copy driveconf.sh to all the nodes:

```

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

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

```

Step 3. Run the following command from the admin node to run the script across all data nodes:

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

Step 4. Run the following from the admin node to list the partitions and mount points:


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

Procedure 22. Delete Partitions

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

Note: 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<?>
```

Procedure 23. Verify Cluster

This procedure explains how 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.

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

```
#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 Veri-fication
=== ${NC}"
echo ""
echo ""
echo -e "${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 ${NC}"
clush -a -B "grep '^model name' /proc/cpuinfo | sort -u"
echo ""
clush -a -B " `which lscpu` | grep -v -e op-mode -e ^Vendor -e family -e \ Model: -e Stepping: -e Bo-goMIPS -e
Virtual -e ^Byte -e ^NUMA node(s)'"
echo ""
echo ""
# probe for nic info #
echo -e "${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 ${NC}"
clush -a -B "uname -srvn | fmt"
echo ""
echo ""
echo -e "${green}Date ${NC}"
clush -a -B date
echo ""
echo ""
echo -e "${green}NTP Status ${NC}"
clush -a -B "ntpstat 2>&l | head -1"
echo ""
echo ""
echo -e "${green}SELINUX ${NC}"
clush -a -B "echo -n 'SELinux status: '; grep ^SELINUX= \ /etc/selinux/config 2>&l"
echo ""
echo ""
clush -a -B "echo -n 'CPUspeed Service: '; `which service` cpuspeed \ status 2>&l"
clush -a -B "echo -n 'CPUspeed Service: '; `which chkconfig` --list \ cpuspeed 2>&l"
echo ""
echo ""
echo -e "${green}Java Version${NC}"
clush -a -B 'java -version 2>&l; 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'

```

Step 1. Change permissions to executable:

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

Step 2. Run the Cluster Verification tool from the admin node. This can be run before starting Hadoop to identify any discrepancies in Post OS Configuration between the servers or during troubleshooting of any cluster / Hadoop issues:

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

Install Cloudera Data Platform Private Cloud Base

This chapter contains the following:

- [Cloudera Runtime](#)
- [Additional Tools](#)
- [Cloudera Data Platform Private Cloud Base Requirements](#)
- [Enable AutoTLS](#)
- [Enable Kerberos](#)
- [Install CDP Private Cloud Base](#)
- [Install CDS 3.3 Powered by Apache Spark](#)

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

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

Cloudera Runtime

Cloudera Runtime is the core open-source software distribution within CDP Private Cloud Base. Cloudera Runtime includes approximately 50 open-source projects that comprise the core distribution of data management tools within CDP. Cloudera Runtime components are documented in this library. See Cloudera Runtime Component Versions for a list of these components. For more information review Cloudera Runtime Release notes: <https://docs.cloudera.com/cdp-private-cloud-base/7.1.8/runtime-release-notes/index.html>

Additional Tools

CDP Private Cloud Base also includes the following tools to manage and secure your deployment:

- Cloudera Manager allows you to manage, monitor, and configure your clusters and services using the Cloudera Manager Admin Console web application or the Cloudera Manager API.
- Apache Atlas provides a set of metadata management and governance services that enable you to manage CDP cluster assets.
- Apache Ranger manages access control through a user interface that ensures consistent policy administration in CDP clusters.

For more details review, [Cloudera Private Cloud Base Installation guide](#).

Cloudera Data Platform Private Cloud Base Requirements

Refer to the [CDP Private Cloud Base Requirements and Supported Versions](#) for information about hardware, operating system, and database requirements, as well as product compatibility matrices.

Refer Cloudera Manager release note for new feature and support: <https://docs.cloudera.com/cdp-private-cloud-base/7.1.8/manager-release-notes/topics/cm-whats-new-773.html>

Procedure 1. Setup Cloudera Manager Repository

Note: These steps require a cloudera username and password to access <https://archive.cloudera.com/p/cm7/>

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

```
# mkdir -p /var/www/html/cloudera-repos/cm7.7.3/
```

Step 2. Download Cloudera Manager Repository:

```
# cd /var/www/html/cloudera-repos/cm7.7.3/
# wget https://<username>:<password>@archive.cloudera.com/p/cm7/7.7.3/redhat8/yum/cloudera-manager-trial.repo
# wget https://<username>:<password>@archive.cloudera.com/p/cm7/7.7.3/allkeys.asc
```

Step 3. Edit cloudera-manager-trial.repo file baseurl and gpgkey with username and password provided by Cloudera and edit URL to match repository location.

```
# vi cloudera-manager-trial.repo
[cloudera-manager]
name=Cloudera Manager 7.7.3
baseurl=https://<username>:<password>@archive.cloudera.com/p/cm7/7.7.3/redhat8/yum/
gpgkey=https://<username>:<password>@archive.cloudera.com/p/cm7/7.7.3/redhat8/yum/RPM-GPG-KEY-cloudera
gpgcheck=1
enabled=1
autorefresh=0
type=rpm-md
```

Step 4. Create directory to download cloudera manager agent, daemon, and server files

```
# mkdir -p /var/www/html/cloudera-repos/cm7.7.3/cloudera-manager/RPMS/x86_64
# cd /var/www/html/cloudera-repos/cm7.7.3/cloudera-manager/RPMS/x86_64/

# wget https://<username>:<password>@archive.cloudera.com/p/cm7/7.7.3/redhat8/yum/RPMS/x86_64/cloudera-
manager-agent-7.7.3-32839716.el8.x86_64.rpm
# wget https://<username>:<password>@archive.cloudera.com/p/cm7/7.7.3/redhat8/yum/RPMS/x86_64/cloudera-
manager-daemons-7.7.3-32839716.el8.x86_64.rpm
# wget https://<username>:<password>@archive.cloudera.com/p/cm7/7.7.3/redhat8/yum/RPMS/x86_64/cloudera-
manager-server-7.7.3-32839716.el8.x86_64.rpm
# wget https://<username>:<password>@archive.cloudera.com/p/cm7/7.7.3/redhat8/yum/RPMS/x86_64/cloudera-
manager-server-db-2-7.7.3-32839716.el8.x86_64.rpm
# wget https://<username>:<password>@archive.cloudera.com/p/cm7/7.7.3/redhat8/yum/RPMS/x86_64/openjdk8-
8.0+232_9-cloudera.x86_64.rpm

# ls -l /var/www/html/cloudera-repos/cm7.7.3/cloudera-manager/RPMS/x86_64
total 1857868
-rw-r--r-- 1 root root 50416076 Oct 13 07:10 cloudera-manager-agent-7.7.3-32839716.el8.x86_64.rpm
-rw-r--r-- 1 root root 1747532156 Oct 13 07:10 cloudera-manager-daemons-7.7.3-32839716.el8.x86_64.rpm
-rw-r--r-- 1 root root 17840 Oct 13 07:10 cloudera-manager-server-7.7.3-32839716.el8.x86_64.rpm
-rw-r--r-- 1 root root 15076 Oct 13 07:10 cloudera-manager-server-db-2-7.7.3-32839716.el8.x86_64.rpm
-rw-r--r-- 1 root root 104465615 Oct 13 07:10 openjdk8-8.0+232_9-cloudera.x86_64.rpm
```

Step 5. Run createrepo command to create local repository.

```
# createrepo --baseurl http://10.4.1.90/cloudera-repos/cm7.7.3/ /var/www/html/cloudera-repos/cm7.7.3/
```

Note: In a web browser please check and verify cloudera manager repository created by entering baseurl <http://10.4.1.90/cloudera-repos/cm7.7.3/>

Step 6. Create the cloudera manager repo file as below:

```
# cd /var/www/html/cloudera-repos/cm7.7.3/
# vi cloudera-manager.repo
[cloudera-manager]
name=Cloudera Manager 7.7.3
```

```
baseurl=http://10.4.1.90/cloudera-repos/cm7.7.3/
gpgcheck=0
enabled=1

# chmod -R ugo+rX /var/www/html/cloudera-repos/cm7.7.3/
```

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

```
# cp /var/www/html/cloudera-repos/cm7.7.3/cloudera-manager.repo /etc/yum.repos.d/cloudera-manager.repo
```

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

```
# ansible all -m copy -a "src=/etc/yum.repos.d/cloudera-manager.repo dest=/etc/yum.repos.d/cloudera-manager.repo"
```

Procedure 2. Set Up the Local Parcels for CDP PvC Base 7.1.8

Step 1. From a host connected the internet, download CDP PvC Base 7.1.8 parcels for RHEL8 from the URL: <https://archive.cloudera.com/p/cdh7/7.1.8.0/parcels/> and place them in the directory /var/www/html/cloudera-repos/cdh7.1.8.0/ of the admin node.

Step 2. Create directory to download CDH parcels.

```
# mkdir -p /var/www/html/cloudera-repos/cdh7.1.8.0/
```

Step 3. Download CDH parcels as highlighted below:

```
# wget https://<username>:<password>@archive.cloudera.com/p/cdh7/7.1.8.0/parcels/CDH-7.1.8-1.cdh7.1.8.p0.30990532-el8.parcel
# wget https://<username>:<password>@archive.cloudera.com/p/cdh7/7.1.8.0/parcels/CDH-7.1.8-1.cdh7.1.8.p0.30990532-el8.parcel.sha1
# wget https://<username>:<password>@archive.cloudera.com/p/cdh7/7.1.8.0/parcels/CDH-7.1.8-1.cdh7.1.8.p0.30990532-el8.parcel.sha256
# wget https://<username>:<password>@archive.cloudera.com/p/cdh7/7.1.8.0/parcels/KEYTRUSTEE_SERVER-7.1.8.0-1.keytrustee7.1.8.0.p0.30990532-el8.parcel
# wget https://<username>:<password>@archive.cloudera.com/p/cdh7/7.1.8.0/parcels/KEYTRUSTEE_SERVER-7.1.8.0-1.keytrustee7.1.8.0.p0.30990532-el8.parcel.sha
# wget https://<username>:<password>@archive.cloudera.com/p/cdh7/7.1.8.0/parcels/KEYTRUSTEE_SERVER-7.1.8.0-1.keytrustee7.1.8.0.p0.30990532-el8.parcel.sha1
# wget https://<username>:<password>@archive.cloudera.com/p/cdh7/7.1.8.0/parcels/KEYTRUSTEE_SERVER-7.1.8.0-1.keytrustee7.1.8.0.p0.30990532-el8.parcel.sha256
# wget https://<username>:<password>@archive.cloudera.com/p/cdh7/7.1.8.0/parcels/manifest.json

# chmod -R ugo+rX /var/www/html/cloudera-repos/cdh7.1.8.0/
```

Note: In a web browser please check and verify cloudera manager repository created by entering baseurl: <http://10.4.1.90/cloudera-repos/cdh7.1.8.0/>

Procedure 3. Set Up the Local Parcels for CDS 3.3 powered by Apache Spark

Cloudera Service Descriptors (CSD) file for CDS 3.3 is available in Cloudera Manager for CDP 7.1.8.

Step 1. From a host connected the internet, download CDS 3.3 Powered by Apache Spark parcels for RHEL8 from the URL: <https://archive.cloudera.com/p/spark3/3.3.7180.0/parcels/> and place them in the directory [/var/www/html/cloudera-repos/ spark3-3.3.7180/](http://var/www/html/cloudera-repos/spark3-3.3.7180/) of the admin node.

Note: Although Spark 2 and Spark 3 can coexist in the same CDP Private Cloud Base cluster, you cannot use multiple Spark 3 versions simultaneously. All clusters managed by the same Cloudera Manager Server must use exactly the same version of CDS 3.3 Powered by Apache Spark.

Step 2. Create directory to download CDH parcels.

```
# mkdir -p /var/www/html/cloudera-repos/spark3-3.3.7180/
```

Step 3. Download CDH parcels as highlighted below:

```
# wget https://<username>:<password>@archive.cloudera.com/p/spark3/3.3.7180.0/parcels/manifest.json
# wget https://<username>:<password>@archive.cloudera.com/p/spark3/3.3.7180.0/parcels/SPARK3-
3.3.0.3.3.7180.0-274-1.p0.31212967-e18.parcel
# wget https://<username>:<password>@archive.cloudera.com/p/spark3/3.3.7180.0/parcels/SPARK3-
3.3.0.3.3.7180.0-274-1.p0.31212967-e18.parcel.sha1

# chmod -R ugo+rX /var/www/html/cloudera-repos/spark3-3.3.7180/
```

Step 4. In a web browser please check and verify cloudera manager repository created by entering baseurl: <http://10.4.1.90/cloudera-repos/spark3-3.3.7180/>

Procedure 4. Install Python 3.8 on RHEL8 for Hue

Certain services, such as Hue, in CDP 7.1.8 and higher use Python 3.8. You must install Python 3.8 on all the hosts running the affected services after you have installed Cloudera Manager and before adding the services to your cluster.

Note: Installing Python 3.8 is mandatory if you want to use Hue.

Note: Ubuntu 20 comes preinstalled with Python 3.8. You must install Python 3.8 manually on CentOS 7, RHEL 8, SLES 12, and Ubuntu 18.

Step 1. Install the following packages before installing Python 3.8

```
# sudo dnf install -y gcc openssl-devel bzip2-devel libffi-devel
```

Step 2. Download Python 3.8 and decompress the package by running the following commands:

```
# cd /opt/
# curl -O https://www.python.org/ftp/python/3.8.12/Python-3.8.12.tgz
# tar -zxvf Python-3.8.14.tgz
```

Step 3. Go to decompressed Python directory

```
# cd /opt/Python-3.8.14/
```

Step 4. Install Python 3.8 as follows:

```
./configure --enable-optimizations --enable-shared
```

Note: By default, Python could be installed in any one of the following locations. If you are installing Python 3.8 in any other location, then you must specify the path using the `--prefix` option.

```
/usr/bin
/usr/local/python38/bin
/usr/local/bin
/opt/rh/rh-python38/root/usr/bin
```

Note: The `--enable-shared` option is used to build a shared library instead of a static library.

```
echo $LD_LIBRARY_PATH
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib/
cd /usr/local/bin/
ls -ll
```

Step 5. Build Python 3.8 as follows:

```
# make
```

Step 6. Run the following command to put the compiled files in the default location or in the custom location that you specified using the `--prefix` option:

```
# make install
```

Step 7. Copy the shared compiled library files (`libpython3.8.so`) to the `/lib64/` directory:

```
# cp --no-clobber ./libpython3.8.so* /lib64/
```

Step 8. Change the permissions of the libpython3.8.so files as follows:

```
# chmod 755 /lib64/libpython3.8.so*
```

Step 9. If you see an error such as error while loading shared libraries: libpython3.8.so.1.0: cannot open shared object file: No such file or directory, then run the following command:

```
# export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib/
```

Step 10. (For Hue) If you have installed Python 3.8 at a custom location, then you must append the custom path in Cloudera Manager > Clusters > Hue > Configuration > Hue Service Environment Advanced Configuration Snippet (Safety Valve) separated by colon (:) as follows:

Key: PATH

Value: [***CUSTOM-INSTALL-PATH***]:/usr/local/sbin:/usr/local/bin:/usr/sbin:

Step 11. Check Python version

```
# ansible nodes -m command -a "python3 --version"
cdipdn01.sjc-cdip.cisco.local | CHANGED | rc=0 >>
Python 3.8.12
cdipdn02.sjc-cdip.cisco.local | CHANGED | rc=0 >>
Python 3.8.12
```

Procedure 5. Install and Configure Database for Cloudera Manager

Cloudera Manager uses various databases and datastores to store information about the Cloudera Manager configuration, as well as information such as the health of the system, or task progress.

Please review [Database Requirement for CDP PvC Base](#).

This procedure highlights the installation and configuration steps with PostgreSQL. Please review Install and Configure Databases for CDP Private Cloud Base for more details: <https://docs.cloudera.com/cdp-private-cloud-base/7.1.8/installation/topics/cdpdc-install-configure-databases.html>

Step 1. Install PostgreSQL packages

```
# sudo dnf install -y https://download.postgresql.org/pub/repos/yum/reporepms/EL-8-x86_64/pgdg-redhat-repo-latest.noarch.rpm
# sudo dnf -qy module disable postgresql
# sudo dnf -y install postgresql14 postgresql14-server postgresql14-libs postgresql14-devel
```

Step 2. Backup existing database.

Note: If you already have a PostgreSQL database set up, you can skip to the section Configuring and Starting the PostgreSQL Server to verify that your PostgreSQL configurations meet the requirements for Cloudera Manager.

Step 3. Make sure that the data directory, which by default is `/var/lib/postgresql/data/`, is on a partition that has sufficient free space.

Note: Cloudera Manager supports the use of a custom schema name for the Cloudera Manager Server database, but not the Runtime component databases (such as Hive and Hue). For more information, see Schemas in the PostgreSQL documentation. By default, PostgreSQL only accepts connections on the loopback interface. You must reconfigure PostgreSQL to accept connections from the fully qualified domain names (FQDN) of the hosts hosting the services for which you are configuring databases. If you do not make these changes, the services cannot connect to and use the database on which they depend.

Step 4. Installing the psycopg2 Python package for PostgreSQL-backed Hue.

Note: If you are installing Runtime 7 and using PostgreSQL as a backend database for Hue, then you must install the 2.9.3 version of the psycopg2 package on all Hue hosts. The psycopg2 package is automatically installed as a dependency of Cloudera Manager Agent, but the version installed is often lower than 2.9.3

```
Install the psycopg2 package dependencies for RHEL 8 by running the following commands:  
# yum install -y xmlsec1 xmlsec1-openssl  
  
Add the location of the installed postgresql-devel package to the PATH environment variable by running the following command:  
# export PATH=/usr/pgsql-[*DB-VERSION*/bin:$PATH  
  
Install the psycopg2 package by running the following command:  
# pip3.8 install psycopg2==2.9.5
```

Step 5. Make sure that LC_ALL is set to en_US.UTF-8 and initialize the database as follows:

```
# echo 'LC_ALL="en_US.UTF-8"' >> /etc/locale.conf
```

Step 6. Initialize the database

```
# sudo /usr/pgsql-14/bin/postgresql-14-setup initdb
```

Step 7. To enable MD5 authentication, edit /var/lib/pgsql/14/data/pg_hba.conf by adding the following line:

```
# vi /var/lib/pgsql/14/data/pg_hba.conf  
host all all 127.0.0.1/32 md5
```

Step 8. Configure settings to ensure your system performs as expected. Update these settings in the /var/lib/pgsql/14/data/postgresql.conf file. Settings vary based on cluster size and resources as follows:

```
# vi /var/lib/pgsql/14/data/postgresql.conf  
listen_addresses = '*' # what IP address(es) to listen on;  
max_connections = 1000 # (change requires restart)  
shared_buffers = 1024MB # min 128kB  
wal_buffers = 16MB # min 32kB, -1 sets based on shared_buffers  
max_wal_size = 6GB  
min_wal_size = 512MB  
checkpoint_completion_target = 0.9 # checkpoint target duration, 0.0 - 1.0  
standard_conforming_strings = off
```

Note: Settings vary based on cluster size and resources.

Step 9. Start the PostgreSQL Server and configure to start at boot.

```
# systemctl start postgresql-14.service  
# systemctl enable postgresql-14.service
```

Step 10. Install and configure Postgres JDBC Drive

```
# yum install -y postgresql-jdbc*  
# cp /usr/share/java/postgresql-jdbc.jar /usr/share/java/postgresql-connector-java.jar  
# ls /usr/share/java/postgresql-connector-java.jar  
# chmod 644 /usr/share/java/postgresql-connector-java.jar
```

Step 11. Create databases and service accounts for components that require databases. Following components requires databases:

- Cloudera Manager Server
- Cloudera Management Service roles
- Data Analytics Studio (DAS) Supported with PostgreSQL only.
- Hue
- Hive metastore
- Oozie
- Data Analytics Studio

- Schema Registry
- Streams Messaging Manager

Note: The databases must be configured to support the PostgreSQL UTF8 character set encoding.

Note: Record the values you enter for database names, usernames, and passwords. The Cloudera Manager installation wizard requires this information to correctly connect to these databases.

```
# sudo -u postgres psql

CREATE ROLE scm LOGIN PASSWORD 'Password';
CREATE DATABASE scm OWNER scm ENCODING 'UTF8';
GRANT ALL PRIVILEGES ON DATABASE scm TO scm;

CREATE ROLE amon LOGIN PASSWORD 'Password';
CREATE DATABASE amon OWNER amon ENCODING 'UTF8';
GRANT ALL PRIVILEGES ON DATABASE amon TO amon;

CREATE ROLE rman LOGIN PASSWORD 'Password';
CREATE DATABASE rman OWNER rman ENCODING 'UTF8';
GRANT ALL PRIVILEGES ON DATABASE rman TO rman;

CREATE ROLE hue LOGIN PASSWORD 'Password';
CREATE DATABASE hue OWNER hue ENCODING 'UTF8';
GRANT ALL PRIVILEGES ON DATABASE hue TO hue;

CREATE ROLE hive LOGIN PASSWORD 'Password';
CREATE DATABASE metastore OWNER hive ENCODING 'UTF8';
GRANT ALL PRIVILEGES ON DATABASE metastore TO hive;

CREATE ROLE oozie LOGIN PASSWORD 'Password';
CREATE DATABASE oozie OWNER oozie ENCODING 'UTF8';
GRANT ALL PRIVILEGES ON DATABASE oozie TO oozie;

CREATE ROLE rangeradmin LOGIN PASSWORD 'Password';
CREATE DATABASE ranger OWNER rangeradmin ENCODING 'UTF8';
GRANT ALL PRIVILEGES ON DATABASE ranger TO rangeradmin;

CREATE ROLE registry LOGIN PASSWORD 'Password';
CREATE DATABASE registry OWNER registry ENCODING 'UTF8';
GRANT ALL PRIVILEGES ON DATABASE registry TO registry;

CREATE ROLE streamsmgmr LOGIN PASSWORD 'Password';
CREATE DATABASE streamsmgmr OWNER streamsmgmr ENCODING 'UTF8';
GRANT ALL PRIVILEGES ON DATABASE streamsmgmr TO streamsmgmr;

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

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

Note: If you plan to use Apache Ranger, please visit [Configuring a PostgreSQL Database for Ranger or Ranger KMS](#) for instructions on creating and configuring the Ranger database.

Note: If you plan to use Schema Registry or Streams Messaging Manager, please visit [Configuring the Database for Streaming Components](#) for instructions on configuring the database.

The following procedures describes how to install Cloudera Manager and then using Cloudera Manager to install Cloudera Data Platform Private Cloud Base 7.1.8.

Procedure 6. Install Cloudera Manager

Cloudera Manager, an end-to-end management application, is used to install and configure CDP PvC Base. During CDP Installation, Cloudera Manager's Wizard will help to install Hadoop services and any other role(s)/service(s) on all nodes using the following procedure:

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

Note: Please see the [JAVA requirements](#) for CDP PvC Base.

Step 1. Install the Cloudera Manager Server packages by running following command:

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

Step 2. Run the scm_prepare_database.sh script to check and prepare Cloudera Manager Server and the database connection.

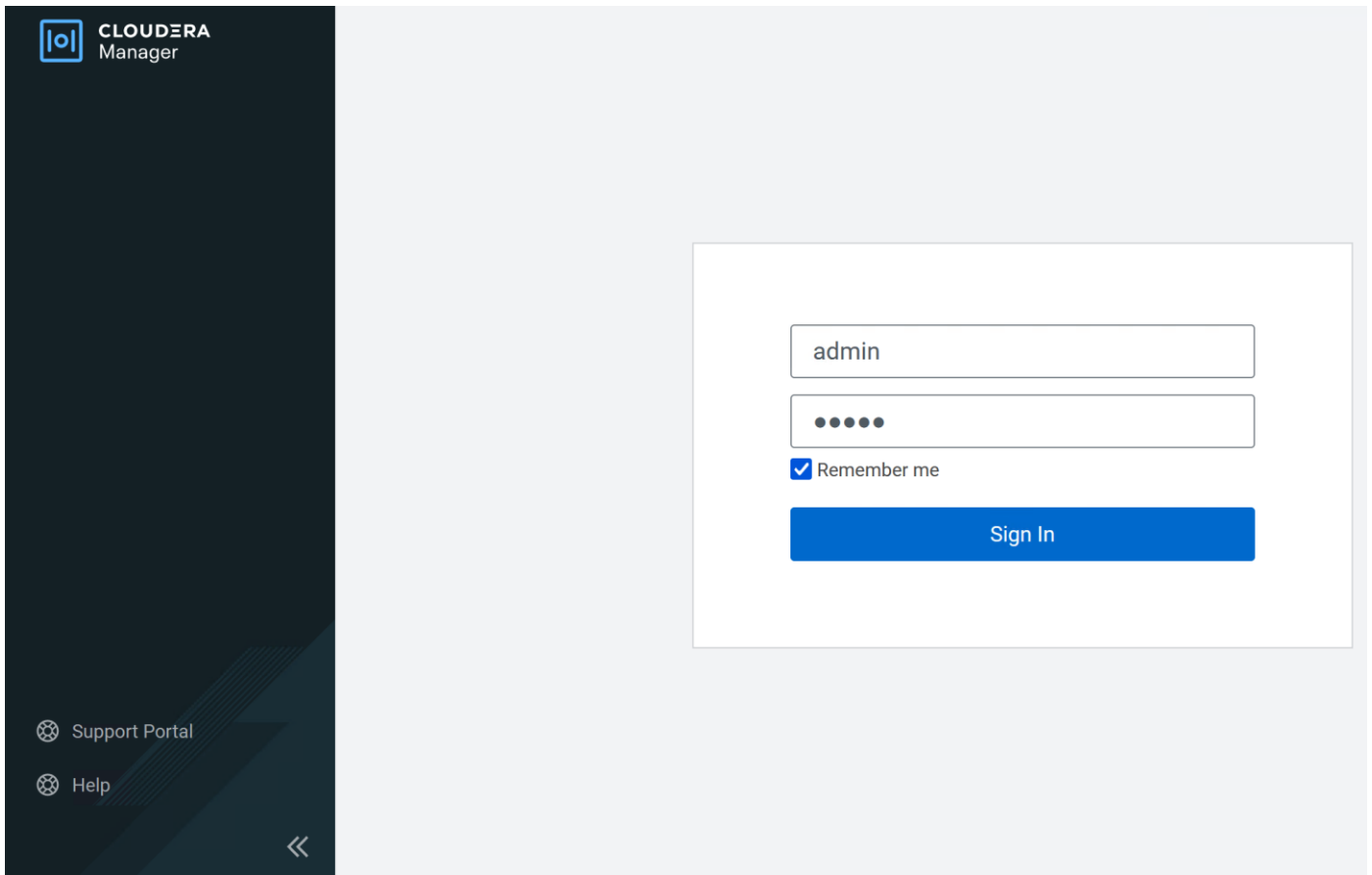
```
# cd /opt/cloudera/cm/schema/  
# ./scm_prepare_database.sh postgresql registry registry <Password>  
# ./scm_prepare_database.sh postgresql streamsmgmr streamsmgmr <Password>  
# ./scm_prepare_database.sh postgresql amon amon <Password>  
# ./scm_prepare_database.sh postgresql rman rman <Password>  
# ./scm_prepare_database.sh postgresql hue hue <Password>  
# ./scm_prepare_database.sh postgresql metastore hive <Password>  
# ./scm_prepare_database.sh postgresql oozie oozie <Password>  
# ./scm_prepare_database.sh postgresql das das <Password>  
# ./scm_prepare_database.sh postgresql ranger rangeradmin <Password>  
# ./scm_prepare_database.sh postgresql scm scm <Password>
```

Step 3. Start the Cloudera Manager Server:

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

Step 4. Access the Cloudera Manager WebUI using the URL, http://<cm_ip_address>:7180

Note: Default username and password for Cloudera Manager is admin/admin.



Procedure 7. Enable AutoTLS

Auto-TLS is managed using the certmanager utility, which is included in the Cloudera Manager Agent software, and not the Cloudera Manager Server software. You must install the Cloudera Manager Agent software on the Cloudera Manager Server host to be able to use the utility. You can use certmanager to manage auto-TLS on a new installation. For more information, go to: [Configuring TLS Encryption for Cloudera Manager Using Auto-TLS](#)

Step 1. The certmanager syntax is as follows:

```
# cd /opt/cloudera/cm-agent/bin/  
# export JAVA_HOME=/usr/java/jdk-11.0.16; /opt/cloudera/cm-agent/bin/certmanager setup --configure-services  
INFO:root:Logging to /var/log/cloudera-scm-agent/certmanager.log
```

Step 2. The certificates, keystores, and password files generated by auto-TLS are stored in `/var/lib/cloudera-scm-agent/agent-cert` on each Cloudera Manager Agent.

```
# cd /var/lib/cloudera-scm-agent/agent-cert/  
[root@rhelnn01 agent-cert]# ls -l  
total 12  
-rw-r--r-- 1 cloudera-scm cloudera-scm 1233 Oct 27 17:47 cm-auto-global_truststore.jks  
-rw----- 1 cloudera-scm cloudera-scm 4354 Oct 27 17:47 cm-auto-host_keystore.jks
```

Step 3. Restart Cloudera Manager Server.

```
# systemctl restart cloudera-scm-server  
# systemctl status cloudera-scm-server -l
```

Procedure 8. Enable Kerberos

Cloudera Manager provides a wizard for integrating your organization's Kerberos with your cluster to provide authentication services. Cloudera Manager clusters can be integrated with MIT Kerberos, Red Hat Identity

Management (or the upstream FreeIPA), or Microsoft Active Directory. For more information, see [Enable Kerberos Authentication for CDP](#).

Note: In our lab, we configured Active-Directory based Kerberos authentication. We presume that Active Directory is pre-configured with OU, user(s) and proper authentication is setup for Kerberos Authentication. LDAP users and bind users are expected to be in the same branch/OU.

Note: Before integrating Kerberos with your cluster, configure TLS encryption between Cloudera Manager Server and all Cloudera Manager Agent host systems in the cluster. During the Kerberos integration process, Cloudera Manager Server sends keytab files to the Cloudera Manager Agent hosts, and TLS encrypts the network communication, so these files are protected.

Note: For Active Directory, you must have administrative privileges to the Active Directory instance for initial setup and for on-going management, or you will need to have the help of your AD administrator prior to and during the integration process. For example, administrative access is needed to access the Active Directory KDC, create principals, and troubleshoot Kerberos TGT/TGS-ticket-renewal and take care of any other issues that may arise.

Step 1. In Cloudera manager console select setup a KDC. Click Continue.

Add Private Cloud Base Cluster

Private Cloud Base Cluster
Add a cluster to provide storage and metadata for a compute cluster or to run workloads that benefit from data locality.
Selected

✔ AutoTLS has already been enabled.

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

Adding a cluster in Cloudera Manager consists of two steps.

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

Quick Links

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

← Back Continue →

Step 2. Select Active Directory as shown below.

Setup KDC for this Cloudera Manager

- 1 Getting Started
- 2 Enter KDC Information
- 3 Manage krb5.conf
- 4 Enter Account Credentials
- 5 Command Details

Getting Started

ⓘ This wizard walks you through the steps to configure Cloudera Manager for Kerberos authentication.

Before using the wizard, ensure that you have performed the following steps:

1. Read the [documentation](#) about enabling Kerberos.
2. Set up a working KDC (Key Distribution Center) and specify the **KDC Type**:

KDC Type ⓘ

MIT KDC

Active Directory

Red Hat IPA

[Undo](#)

3. Configure the KDC to have **non-zero ticket lifetime and renewal lifetime**. Clusters will not work properly if tickets are not renewable.
4. Configure the KDC to have an account that has **permissions to create other accounts**.
5. Install OpenLdap client libraries on the **Cloudera Manager Server host** if you want to use Active Directory.

6.

```
# RHEL / CentOS
$ yum install openldap-clients krb5-workstation krb5-libs
```

If Red Hat IPA is used as the KDC

```
$ yum install freeipa-client
```

```
# SUSE
$ zypper install openldap2-client krb5-client
```

If Red Hat IPA is used as the KDC

```
$ zypper install freeipa-client
```

```
# Ubuntu
$ apt-get install ldap-utils krb5-user
```

If Red Hat IPA is used as the KDC

```
$ apt-get install freeipa-client
```

I have completed all the above steps.

Cancel
← Back
Continue →

Step 3. As recommended, install the following in all Cloudera Manager hosts by running the following command. Once completed, click the checkbox “I have completed all the above steps” and click Continue.

```
# ansible all -m command -a "yum install -y openldap-clients krb5-workstation krb5-libs"
```

Step 4. Enter KDC information for this Cloudera Manager. Use [Table 5](#) as an example to fill-in the KDC setup information.

Table 5. KDC Setup components and their corresponding value

| Component | Value |
|-------------------------------------------------------------|------------------------------------------------|
| Kerberos Security Realm | SJC-CDIP.CISCO.LOCAL |
| KDC Server Host | winjb-vlan4.sjc-cdip.cisco.local |
| KDC Admin Server Host | winjb-vlan4.sjc-cdip.cisco.local |
| Domain Name(s) | sjc2-cdip.cisco.local |
| Active Directory Suffix | OU=cdip_kerberos,DC=sjc-cdip,DC=cisco,DC=local |
| Active Directory Delete Accounts on Credential Regeneration | Select |

Setup KDC for this Cloudera Manager

- 1 Getting Started
- 2 Enter KDC Information
- 3 Manage krb5.conf
- 4 Enter Account Credentials
- 5 Command Details

Enter KDC Information

Specify information about the KDC. The properties below are used by Cloudera Manager to generate principals for daemons running on the cluster.

| | | |
|-------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|--|
| Kerberos Encryption Types <small>krb_enc_types</small> | <input type="text" value="rc4-hmac"/> | |
| Kerberos Security Realm <small>default_realm</small> <small>security_realm</small> | <input type="text" value="SJC-CDIP.CISCO.LOCAL"/> | |
| KDC Server Host <small>kdc</small> <small>kdc_host</small> | <input type="text" value="winjb-vlan4.sjc-cdip.cisco.local"/> | |
| KDC Admin Server Host <small>admin_server</small> <small>kdc_admin_host</small> | <input type="text" value="winjb-vlan4.sjc-cdip.cisco.local"/> | |
| Domain Name(s) <small>krb_domain</small> | <input type="text" value="sjc2-cdip.cisco.local"/> | |
| Active Directory Suffix <small>ad_kdc_domain</small> | <input type="text" value="OU=cdip_kerberos,DC=sjc-cdip,DC=cisco,DC=local"/> | |
| Active Directory Delete Accounts on Credential Regeneration <small>ad_delete_on_regenerate</small> | <input checked="" type="checkbox"/> | |
| Active Directory Set Encryption Types <small>ad_set_encryption_types</small> | <input type="checkbox"/> | |

Cancel
← Back
Continue →

Note: In this setup, we used Kerberos authentication with Active Directory (AD). Setting up AD is beyond the scope of this document.

Step 5. Check the box for Manage “krb5.conf” through Cloudera Manager. This will install krb5.conf file in all the hosts selected for data lake.

Setup KDC for this Cloudera Manager

- 1 Getting Started
- 2 Enter KDC Information
- 3 Manage krb5.conf
- 4 Enter Account Credentials
- 5 Command Details

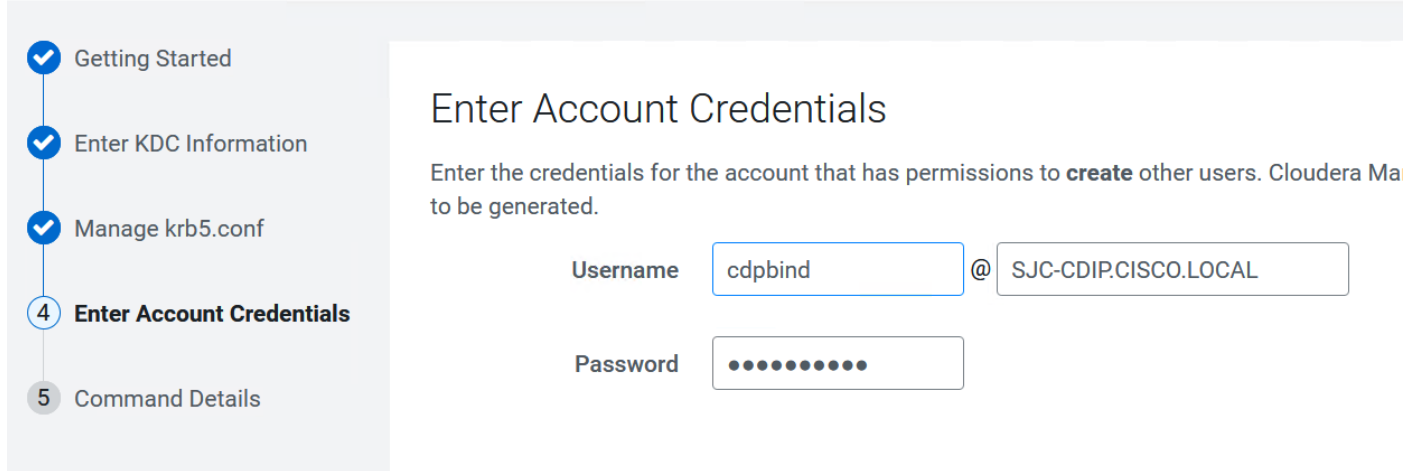
Manage krb5.conf

Specify the properties needed for generating the krb5.conf file for the cluster. You can use the Advanced Configuration Snippet to specify configuration of an advanced KDC setup, for example, with cross-realm authentication.

| | | |
|----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|--|
| krb5.conf file path <small>Requires Server Restart</small> <small>krb_krb5_conf_path</small> | <input type="text" value="/etc/krb5.conf"/> | |
| Manage krb5.conf through Cloudera Manager <small>Requires Server Restart</small> <small>krb_manage_krb5_conf</small> | <input checked="" type="checkbox"/> | |
| Kerberos Ticket Lifetime <small>ticket_lifetime</small> <small>krb_ticket_lifetime</small> | <input type="text" value="1"/> day(s) | |
| Kerberos Renewable Lifetime <small>renew_lifetime</small> <small>krb_renew_lifetime</small> | <input type="text" value="7"/> day(s) | |
| DNS Lookup KDC <small>dns_lookup_kdc</small> <small>krb_dns_lookup_kdc</small> | <input type="checkbox"/> | |
| Forwardable Tickets <small>forwardable</small> <small>krb_forwardable</small> | <input checked="" type="checkbox"/> | |

Step 6. Enter account credentials for the bind user which you have created in AD. This credential will be used to create service accounts in AD. In our lab setup, “cdpbind” user is created in AD for this purpose. Click Continue.

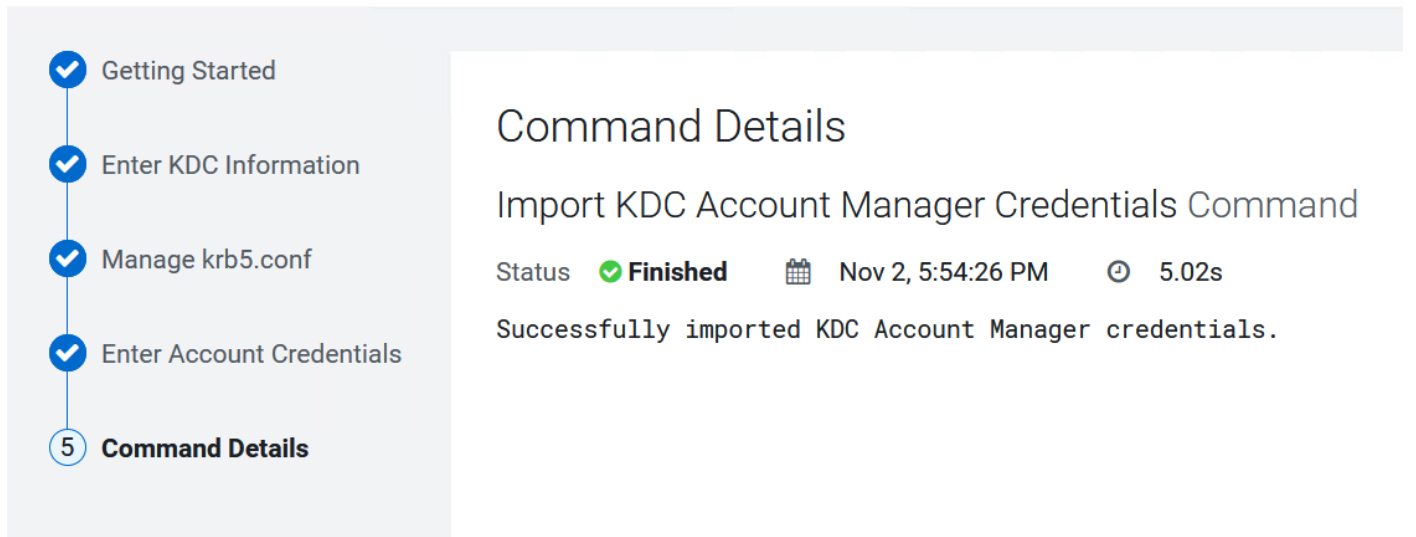
Setup KDC for this Cloudera Manager



The screenshot shows the 'Enter Account Credentials' step in the Cloudera Manager wizard. On the left, a vertical progress bar lists five steps: 'Getting Started', 'Enter KDC Information', 'Manage krb5.conf', '4 Enter Account Credentials' (the current step), and '5 Command Details'. The main content area is titled 'Enter Account Credentials' and contains the instruction: 'Enter the credentials for the account that has permissions to create other users. Cloudera Ma to be generated.' Below this, there are two input fields: 'Username' with the value 'cdpbind' and a domain dropdown menu set to 'SJC-CDIP.CISCO.LOCAL', and 'Password' which is currently masked with dots.

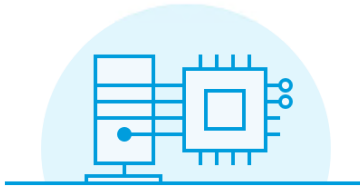
Step 7. Click Finish to complete the KDC setup.

Setup KDC for this Cloudera Manager



The screenshot shows the 'Command Details' step in the Cloudera Manager wizard. On the left, the progress bar now shows '5 Command Details' as the active step. The main content area is titled 'Command Details' and displays the command: 'Import KDC Account Manager Credentials Command'. Below the command, the status is shown as 'Status ✔ Finished' with a calendar icon, the time 'Nov 2, 5:54:26 PM', and a clock icon with '5.02s'. A message below reads: 'Successfully imported KDC Account Manager credentials.'

Once the KDC set up is completed, the Cloudera Manager wizard for adding a cluster will reflect the following:



Private Cloud Base Cluster

Add a cluster to provide storage and metadata for a compute cluster or to run workloads that benefit from data locality.

Selected

✓ AutoTLS has already been enabled.

✓ The KDC is already set up. You can now create Kerberized clusters.

Adding a cluster in Cloudera Manager consists of two steps.

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

💡 Quick Links

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

Procedure 9. Install Cloudera Private Cloud Base using the Cloudera Manager WebUI

Step 1. Upload Cloudera Data Platform license file. Click Continue.

Welcome to Cloudera Manager 7.7.3

Upload License File

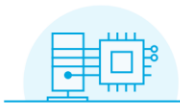
Upload Cloudera Data Platform License

Cloudera Data Platform provides important features that help you manage and monitor your environments. Cloudera Data Platform is a subscription service with enhanced capabilities

[Upload License File](#) (Accept .txt or .zip) ✔ License Uploaded Successfully.

Step 2. Enable AutoTLS and Setup KDC to create Kerberized cluster. (Please review procedure 7 and 8 in this section)

Add Private Cloud Base Cluster



Private Cloud Base Cluster
Add a cluster to provide storage and metadata for a compute cluster or to run workloads that benefit from data locality.
[Selected](#)

✔ AutoTLS has already been enabled.

✔ The KDC is already set up. You can now create Kerberized clusters.

Adding a cluster in Cloudera Manager consists of two steps.

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

📍 Quick Links

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

[← Back](#)

[Continue →](#)

Step 3. Verify Kerberos configuration.

```
# kinit cdpbind@SJC-CDIP.CISCO.LOCAL
Password for cdpbind@SJC-CDIP.CISCO.LOCAL:
# klist
```

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: cdpbind@SJC-CDIP.CISCO.LOCAL
```

```
Valid starting      Expires          Service principal
11/03/2022 15:16:38  11/04/2022 01:16:38  krbtgt/SJC-CDIP.CISCO.LOCAL@SJC-CDIP.CISCO.LOCAL
renew until 11/10/2022 14:16:33
```

Step 4. Enter a cluster name. Click Continue.

Add Private Cloud Base Cluster

1 Cluster Basics

2 Specify Hosts

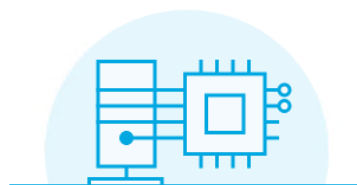
3 Select Repository

4 Install Parcels

5 Inspect Cluster

Cluster Basics

Cluster Name



Base Cluster

A Base Cluster contains storage nodes, compute nodes, and network nodes.

Step 5. Specify the hosts that are part of the cluster using their IP addresses or hostname. The figure below shows a pattern that specifies the IP addresses range. Cloudera Manager will "discover" the nodes based to add in the cluster. Verify that all desired nodes have been found and selected for installation.

```
10.4.1.[81-88] or cdipdn[01-08].sjc-cdip.cisco.local
10.4.1.[89-91] or cdipnn[01-03].sjc-cdip.cisco.local
```

Add Private Cloud Base Cluster

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

Specify Hosts

Hosts should be specified using the same hostname (FQDN) that they will identify themselves with.

Hostname

Hint: Search for hostnames or IP addresses using [pattern](#)

SSH Port

11 hosts scanned, 11 running SSH.

| <input checked="" type="checkbox"/> | Expanded Query | Hostname (FQDN) ↑ | IP Address | Currently Managed | Result |
|-------------------------------------|-------------------------------|-------------------------------|------------|-------------------|--------------------------------|
| <input checked="" type="checkbox"/> | cdipdn01.sjc-cdip.cisco.local | cdipdn01.sjc-cdip.cisco.local | 10.4.1.81 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | cdipdn02.sjc-cdip.cisco.local | cdipdn02.sjc-cdip.cisco.local | 10.4.1.82 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | cdipdn03.sjc-cdip.cisco.local | cdipdn03.sjc-cdip.cisco.local | 10.4.1.83 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | cdipdn04.sjc-cdip.cisco.local | cdipdn04.sjc-cdip.cisco.local | 10.4.1.84 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | cdipdn05.sjc-cdip.cisco.local | cdipdn05.sjc-cdip.cisco.local | 10.4.1.85 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | cdipdn06.sjc-cdip.cisco.local | cdipdn06.sjc-cdip.cisco.local | 10.4.1.86 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | cdipdn07.sjc-cdip.cisco.local | cdipdn07.sjc-cdip.cisco.local | 10.4.1.87 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | cdipdn08.sjc-cdip.cisco.local | cdipdn08.sjc-cdip.cisco.local | 10.4.1.88 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | cdipnn01.sjc-cdip.cisco.local | cdipnn01.sjc-cdip.cisco.local | 10.4.1.90 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | cdipnn02.sjc-cdip.cisco.local | cdipnn02.sjc-cdip.cisco.local | 10.4.1.89 | No | Host was successfully scanned. |
| <input checked="" type="checkbox"/> | cdipnn03.sjc-cdip.cisco.local | cdipnn03.sjc-cdip.cisco.local | 10.4.1.91 | No | Host was successfully scanned. |

Rows per page: 25 1 - 11 of 11

Step 6. Enter Custom Repository or Cloudera Repository to install Cloudera Manager Agent on all nodes in the cluster.

Add Private Cloud Base Cluster

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

Select Repository

Cloudera Manager Agent

Cloudera Manager Agent 7.7.3 (#32839716) needs to be installed on all new hosts.

Repository Location Cloudera Repository (Requires direct Internet access on all hosts.) Custom Repository

Example: `http://LOCAL_SERVER/cloudera-repos/cm7/7.7.3`

Do not include operating system-specific paths in the URL. The path will be automatically derived.

Learn more at [How to set up a custom repository.](#)

Step 7. In other software section, select Use Parcels and click Parcel Repository & Network Settings to provide custom Parcels location to be installed.

Other Software

Cloudera recommends the use of parcels for installation over packages deployment and upgrade of service binaries. Electing not to use parcels will prevent you from using Cloudera Manager's rolling upgrade capabilities.

Install Method

Use Packages

Use Parcels (Recommended)

[? Parcel Repositories & Network Settings](#)

Step 8. Enter custom repository URL for CDH7 and CDS 3.3 parcels. Click Verify and Save. Close the Parcel Repository & Network Settings wizard.

Parcel Repository & Network Settings

Cloudera Manager checks the connection to the configured parcel repository URLs. A valid license is required to access most Cloudera parcel repositories.

>  2/2 URL(s) - The repository was successfully accessed and the manifest downloaded and validated. (HTTP Status: 200)

Remote Parcel Repository URLs

 [remote_parcel_repo_urls](#)



Enable Automatic Authentication for Cloudera Repositories



 [remote_repo_auth](#)

Step 9. Select the parcels for installation.

Other Software

Cloudera recommends the use of parcels for installation over packages, because parcels enable Cloudera Manager to easily manage the software on your cluster, automating the deployment and upgrade of service binaries. Electing not to use parcels will require you to manually upgrade packages on all hosts in your cluster when software updates are available, and will prevent you from using Cloudera Manager's rolling upgrade capabilities.

Install Method Use Packages
 Use Parcels (Recommended)
[Parcel Repositories & Network Settings](#) [Other Parcel Configurations](#)

CDH Version **Versions that are too new for this version of Cloudera Manager (7.7.3) will not be shown.**
 Cloudera Runtime 7.1.8-1.cdh7.1.8.p0.30990532

Additional Parcels SPARK3 3.3.0.3.3.7180.0-274-1.p0.31212967
 None

Step 10. Click Continue.

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

Select Repository

Cloudera Manager Agent

Cloudera Manager Agent 7.7.3 (#32839716) needs to be installed on all new hosts.

Repository Location Cloudera Repository (Requires direct Internet access on all hosts.)
 Custom Repository

Example: http://LOCAL_SERVER/cloudera-repos/cm7/7.7.3
Do not include operating system-specific paths in the URL. The path will be automatically derived.
[Learn more at How to set up a custom repository.](#)

Other Software

Cloudera recommends the use of parcels for installation over packages, because parcels enable Cloudera Manager to easily manage the software on your cluster, automating the deployment and upgrade of service binaries. Electing not to use parcels will require you to manually upgrade packages on all hosts in your cluster when software updates are available, and will prevent you from using Cloudera Manager's rolling upgrade capabilities.

Install Method Use Packages
 Use Parcels (Recommended)
[Parcel Repositories & Network Settings](#) [Other Parcel Configurations](#)

CDH Version **Versions that are too new for this version of Cloudera Manager (7.7.3) will not be shown.**
 Cloudera Runtime 7.1.8-1.cdh7.1.8.p0.30990532

Additional Parcels SPARK3 3.3.0.3.3.7180.0-274-1.p0.31212967
 None

[Cancel](#) [← Back](#) [Continue →](#)

Step 11. Select the appropriate option for JDK.

Note: We selected the “Manually Manage JDK” option as shown below.

Add Private Cloud Base Cluster

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

Select JDK

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

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

Manually manage JDK

i Please ensure that a supported JDK is **already installed** on all hosts. You will need to manage installing the unlimited stre

Install a Cloudera-provided version of OpenJDK

By proceeding, Cloudera will install a supported version of OpenJDK version 8.

Install a system-provided version of OpenJDK

By proceeding, Cloudera will install the default version of OpenJDK version 8 provided by the Operating System.

Step 12. Provide the SSH login credentials for the hosts to install Cloudera packages. Click Continue.

Add Private Cloud Base Cluster

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

Enter Login Credentials

Root access to your hosts is required to install the Cloudera packages. This installation requires sudo/pbrun privileges to become root.

SSH Username ⓘ

Authentication Method All hosts accept same password
 All hosts accept same private key

Password

Confirm Password

SSH Port

Simultaneous Installations
(Running a large number of installations at once can)

Step 13. Cloudera Agent installation wizard displays. Click Continue after the successful Cloudera Agent installation on all hosts.

Add Private Cloud Base Cluster

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

Install Agents

Installation completed successfully.

11 of 11 host(s) completed successfully.

| Hostname | IP Address | Progress | Status | |
|-------------------------------|------------|-------------------------------------------------------------|----------------------------------------|-------------------------|
| cdipdn01.sjc-cdip.cisco.local | 10.4.1.81 | <div style="width: 100%; background-color: #28a745;"></div> | ✓ Installation completed successfully. | Details |
| cdipdn02.sjc-cdip.cisco.local | 10.4.1.82 | <div style="width: 100%; background-color: #28a745;"></div> | ✓ Installation completed successfully. | Details |
| cdipdn03.sjc-cdip.cisco.local | 10.4.1.83 | <div style="width: 100%; background-color: #28a745;"></div> | ✓ Installation completed successfully. | Details |
| cdipdn04.sjc-cdip.cisco.local | 10.4.1.84 | <div style="width: 100%; background-color: #28a745;"></div> | ✓ Installation completed successfully. | Details |
| cdipdn05.sjc-cdip.cisco.local | 10.4.1.85 | <div style="width: 100%; background-color: #28a745;"></div> | ✓ Installation completed successfully. | Details |
| cdipdn06.sjc-cdip.cisco.local | 10.4.1.86 | <div style="width: 100%; background-color: #28a745;"></div> | ✓ Installation completed successfully. | Details |
| cdipdn07.sjc-cdip.cisco.local | 10.4.1.87 | <div style="width: 100%; background-color: #28a745;"></div> | ✓ Installation completed successfully. | Details |
| cdipdn08.sjc-cdip.cisco.local | 10.4.1.88 | <div style="width: 100%; background-color: #28a745;"></div> | ✓ Installation completed successfully. | Details |
| cdipnn01.sjc-cdip.cisco.local | 10.4.1.90 | <div style="width: 100%; background-color: #28a745;"></div> | ✓ Installation completed successfully. | Details |
| cdipnn02.sjc-cdip.cisco.local | 10.4.1.89 | <div style="width: 100%; background-color: #28a745;"></div> | ✓ Installation completed successfully. | Details |
| cdipnn03.sjc-cdip.cisco.local | 10.4.1.91 | <div style="width: 100%; background-color: #28a745;"></div> | ✓ Installation completed successfully. | Details |

Rows per page: 25 ▲ 1 - 11 of 11 |< < > >|

Cancel
← Back
Continue →

Step 14. Parcels Installation wizard reports status parcels distribution and activation on all hosts part of the cluster creation. Click Continue.

Add Private Cloud Base Cluster

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

Install Parcels

The selected parcels are being downloaded and installed on all the hosts in the cluster.

▼ Cloudera Runtime 7.1.8
Downloaded: 100% Distributed: 11/11 (37.7 MIB/s) Unpacked: 11/11 Activated: 11/11

All (11)
 Running (0)
 Failed (0)
 Completed (11)

| Hostname | Throughput | Status | Errors |
|-------------------------------|------------|--------------------------------------------------------------------------|--------|
| cdipdn08.sjc-cdip.cisco.local | 47.2 MIB/s | <div style="width: 100%; background-color: #28a745;"></div> DISTRIBUTED | |
| cdipnn03.sjc-cdip.cisco.local | 4.3 MIB/s | <div style="width: 100%; background-color: #28a745;"></div> DISTRIBUTED | |
| cdipdn03.sjc-cdip.cisco.local | 4.5 MIB/s | <div style="width: 100%; background-color: #28a745;"></div> DISTRIBUTED | |
| cdipdn04.sjc-cdip.cisco.local | 4.8 MIB/s | <div style="width: 100%; background-color: #28a745;"></div> DISTRIBUTED | |
| cdipdn06.sjc-cdip.cisco.local | 4.4 MIB/s | <div style="width: 100%; background-color: #28a745;"></div> DISTRIBUTED | |
| cdipdn05.sjc-cdip.cisco.local | 5.7 MIB/s | <div style="width: 100%; background-color: #28a745;"></div> DISTRIBUTED | |
| cdipnn01.sjc-cdip.cisco.local | 3.9 MIB/s | <div style="width: 100%; background-color: #28a745;"></div> UNPACKING | |
| cdipdn07.sjc-cdip.cisco.local | 5.8 MIB/s | <div style="width: 100%; background-color: #28a745;"></div> DISTRIBUTED | |
| cdipdn02.sjc-cdip.cisco.local | 5.2 MIB/s | <div style="width: 100%; background-color: #28a745;"></div> DISTRIBUTED | |
| cdipnn02.sjc-cdip.cisco.local | 3.4 MIB/s | <div style="width: 100%; background-color: #28a745;"></div> DISTRIBUTING | |
| cdipdn01.sjc-cdip.cisco.local | 4.6 MIB/s | <div style="width: 100%; background-color: #28a745;"></div> DISTRIBUTED | |

Rows per page: 25 ▲ 1 - 11 of 11 |< < > >|

Cancel
← Back
Continue →

Step 15. Inspect Cluster by running Inspect Network Performance and Inspect Hosts for new cluster creation. Review inspector summary. Click Finish.

The screenshot shows the Cloudera Manager interface. On the left is a dark sidebar with navigation options: Clusters, Hosts, Diagnostics, Audits, Charts, Replication, Administration, Data Services (with a 'New' badge), Parcels, Running Commands, Support, and a user profile for 'admin'. The main area displays a table with the following content:

| Status | Description |
|--------|----------------------------------------------------------------------------------------------------------------------------------------|
| ✔ | Inspector ran on all 11 hosts. |
| ✔ | Individual hosts resolved their own hostnames correctly. |
| ✔ | No errors were found while looking for conflicting init scripts. |
| ✔ | No errors were found while checking /etc/hosts. |
| ✔ | All hosts resolved localhost to 127.0.0.1. |
| ✔ | All hosts checked resolved each other's hostnames correctly and in a timely manner. |
| ✔ | Host clocks are approximately in sync (within ten minutes). |
| ✔ | Host time zones are consistent across the cluster. |
| ✔ | No users or groups are missing. |
| ✔ | No conflicts detected between packages and parcels. |
| ✔ | No kernel versions that are known to be bad are running. |
| ✔ | No problems were found with /proc/sys/vm/swappiness on any of the hosts. |
| ✔ | No performance concerns with Transparent Huge Pages settings. |
| ✔ | Hue Python version dependency is satisfied. |
| ✔ | Hue Psycopg2 version for PostgreSQL is satisfied for both CDH 5 and CDH 6. |
| ✔ | A compatible version of the operating system is installed on the hosts in a Private Cloud Containerized Cluster. |
| ✔ | Ports 80 and 443 are available for use on the hosts in a Private Cloud Containerized Cluster. |
| ✔ | A minimum of 16 cores are available for hosts in a Private Cloud Containerized Cluster. |
| ✔ | Storage availability is sufficient for the hosts in a Private Cloud Containerized Cluster. |
| ✔ | The hosts with GPUs that are part of a Private Cloud Containerized Cluster have nVidia Drivers and nvidia-container-runtime installed. |
| ✔ | message.inspector.version.hostCountsForCdh5Cdh6AndCdh7 |
| ✔ | All checked hosts in each cluster are running the same version of components. |
| ✔ | All managed hosts have consistent versions of Java. |
| ✔ | All checked Cloudera Management Daemons versions are consistent with the server. |
| ✔ | All checked Cloudera Management Agents versions are consistent with the server. |

Add Private Cloud Base Cluster

The screenshot shows the 'Inspect Cluster' step in the Cloudera installation wizard. On the left, a vertical navigation pane lists steps from 'Cluster Basics' to 'Inspect Cluster', with 'Inspect Cluster' highlighted as step 8. The main content area is titled 'Inspect Cluster' and contains a message: 'You have created a new empty cluster. Cloudera recommends that you run the following inspections. For accurate measurements, Cloudera r sequentially.' Below this, two inspection items are listed: 'Inspect Network Performance' and 'Inspect Hosts'. Both are marked with a green checkmark. For 'Inspect Network Performance', the status is 'Success', last run 'a few seconds ago', and duration '7.27s'. For 'Inspect Hosts', the status is 'Success', last run 'a few seconds ago', and duration '5.02s'. Each item has a 'Show Inspector Results' link, a 'Run Again' button, and a 'More' dropdown. At the bottom, there is a checkbox labeled 'I understand the risks of not running the inspections or the detected issues, let me continue with cluster setup.'

Step 16. Select services to install. Choose from a combination or services defined or select custom services. Services required based on selection will be automatically added.

The screenshot shows the 'Add Cluster - Configuration' wizard, specifically the 'Select Services' step. On the left, a vertical navigation pane lists steps from '1 Select Services' to '9 Summary', with '1 Select Services' highlighted. The main content area is titled 'Select Services' and contains the instruction: 'Choose a combination of services to install.' There are four radio button options: 'Data Engineering', 'Data Mart', 'Operational Database', and 'Custom Services'. 'Data Engineering' is described as 'Process, develop, and serve predictive models.' with services: HDFS, YARN, YARN Queue Manager, Ranger, Atlas, Hive, Hive on Tez, Spark, Oozie, Hue, and Data Analytics Studio. 'Data Mart' is described as 'Browse, query, and explore your data in an interactive way.' with services: HDFS, Ranger, Atlas, Hive, Hive on Tez, Impala, and Hue. 'Operational Database' is described as 'Real-time insights for modern data-driven business.' with services: HDFS, Ranger, Atlas, and HBase. 'Custom Services' is selected and described as 'Choose your own services. Services required by chosen services will automatically be included.'

Note: It is important to select host(s) to deploy services based on role intended it for. For detailed information, go to: [Runtime Cluster Hosts and Role Assignments](#)

Table 6. Cloudera Data Platform Private Cloud Base host and Role assignment example

| Service Name | Host |
|------------------------------|----------------------------------------------------------------------------------|
| NameNode | cdipnn02, cdipnn03 (HA) |
| HistoryServer | cdipnn02 |
| JournalNodes | cdipnn01, cdipnn02, cdipnn03 |
| ResourceManager | cdipnn02, cdipnn03 (HA) |
| Hue Server | cdipnn01 |
| HiveMetastore Server | cdipnn01 |
| HiveServer2 | cdipnn02 |
| HBase Master | cdipnn02 |
| Oozie Server | cdipnn01 |
| ZooKeeper | cdipnn01, cdipnn02, cdipnn03 |
| DataNode | cdipdn01 - cdipdn08 |
| NodeManager | cdip01 to cdip16 |
| RegionServer | cdipdn01 - cdipdn08 |
| Impala Catalog Server Daemon | cdipnn02 |
| Impala State Store | cdipnn03 |
| Impala Daemon | cdipdn01 - cdipdn08 |
| Solr Server | cdipdn03 (can be installed on all hosts if needed if there is a search use case) |
| Spark History Server | cdipnn02 |
| Spark Executors | cdipdn01 - cdipdn08 |

Step 17. Select services and host assignment in Add cluster configuration wizard.

Services: HDFS, Ranger, Atlas, and HBase

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

| Service Type | Description |
|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <input checked="" type="checkbox"/> Atlas | Apache Atlas provides a set of metadata management and governance services that enable you to find, organize, and manage data assets. This service requires Kerberos. |
| <input type="checkbox"/> Cruise Control | Cruise Control simplifies the operation of Kafka clusters automating workload rebalancing and self-healing. |
| <input checked="" type="checkbox"/> Data Analytics Studio | Data Analytics Studio is the one stop shop for Apache Hive warehousing. Query, optimize and administrate your data with this powerful interface. |
| <input checked="" type="checkbox"/> HBase | Apache HBase is a highly scalable, highly resilient NoSQL OLTP database that enables applications to leverage big data. |
| <input checked="" type="checkbox"/> HDFS | Apache Hadoop Distributed File System (HDFS) is the primary storage system used by Hadoop applications. HDFS creates multiple replicas of data blocks and distributes them on compute hosts throughout a cluster to enable reliable, extremely rapid computations. |
| <input checked="" type="checkbox"/> Hive | Apache Hive is a SQL based data warehouse system. In CDH 6 and earlier, this service includes Hive Metastore and HiveServer2. In Cloudera Runtime 7.0 and later, this service only includes the Hive Metastore; HiveServer2 and other components of the Hive execution engines are part of the Hive on Tez service. |
| <input checked="" type="checkbox"/> Hive on Tez | Hive on Tez is a SQL query engine using Apache Tez. |
| <input checked="" type="checkbox"/> Hue | Hue is the leading SQL Workbench for optimized, interactive query design and data exploration. |
| <input checked="" type="checkbox"/> Impala | Apache Impala provides a real-time SQL query interface for data stored in HDFS and HBase. Impala requires the Hive service and shares the Hive Metastore with Hue. |
| <input type="checkbox"/> Kafka | Apache Kafka is a publish-subscribe messaging rethought as a highly scalable distributed commit log. |

Step 18. Assign roles. Click Continue

Step 19. Select database type and enter database hostname, username, and password on Setup database. Click Test Connection. After successful connection test click Continue.

Oozie Server

✔ Successful

Currently assigned to run on **cdipnn01.sjc-cdip.cisco.local**.

| | | |
|------------|-------------------------------|---------------|
| Type | Database Hostname | Database Name |
| PostgreSQL | cdipnn01.sjc-cdip.cisco.local | oozie |
| Username | Password | |
| oozie | •••••••• | |

Ranger

✔ Successful

| | | |
|---------------|-----------------------|-------------------------------|
| Type | Use JDBC URL Override | Database Hostname |
| PostgreSQL | No | cdipnn01.sjc-cdip.cisco.local |
| Database Name | Username | Password |
| ranger | rangeradmin | •••••••• |

Data Analytics Studio

✔ Successful

| | | |
|------------|-------------------------------|---------------|
| Type | Database Hostname | Database Name |
| PostgreSQL | cdipnn01.sjc-cdip.cisco.local | das |
| Username | Password | |
| das | •••••••• | |

Show Password

Test Connection

Notes:

- The value in the **Database Hostname** field must match the value you used for the hostname when creating the database.

← Back

Continue →

Step 20. Enter the required parameters.

- Assign Roles
- Setup Database
- 4 Enter Required Parameters**
- 5 Review Changes
- 6 Configure Kerberos
- 7 Command Details
- 8 Command Details
- 9 Summary

Enter Required Parameters

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <p>Knox Master Secret gateway_master_secret gateway_master_secret</p> | <p>Knox Gateway Default Group Undo</p> <input type="password" value="....."/> |
| <p>IDBroker Master Secret idbroker_master_secret idbroker_master_secret</p> | <p>Knox IDBroker Default Group Undo</p> <input type="password" value="....."/> |
| <p>Ozone Service ID ozone.service.id ozone.service.id</p> | <p>Ozone (Service-Wide) Undo</p> <input type="text" value="ozone"/> |
| <p>Ranger Admin User Initial Password (Use strong password as per updated 7.1.8+ password criteria). rangeradmin_user_password rangeradmin_user_password</p> | <p>Ranger (Service-Wide) Undo</p> <input type="password" value="....."/> |
| <p>Ranger Usersync User Initial Password (Use strong password as per updated 7.1.8+ password criteria). rangerusersync_user_password rangerusersync_user_password</p> | <p>Ranger (Service-Wide) Undo</p> <input type="password" value="....."/> |
| <p>Ranger Tagsync User Initial Password (Use strong password as per updated 7.1.8+ password criteria). rangertagsync_user_password rangertagsync_user_password</p> | <p>Ranger (Service-Wide) Undo</p> <input type="password" value="....."/> |
| <p>Ranger KMS Keyadmin User Initial Password (Use strong password as per updated 7.1.8+ password criteria). keyadmin_user_password keyadmin_user_password</p> | <p>Ranger (Service-Wide) Undo</p> <input type="password" value="....."/> |

Step 21. Review the changes and edit the configuration parameters as per your requirements.

Add Cluster - Configuration

- Select Services
- Assign Roles
- Setup Database
- Enter Required Parameters
- 5 Review Changes**
- 6 Configure Kerberos
- 7 Command Details
- 8 Command Details
- 9 Summary

Review Changes

All (303) Atlas (18) Core Configuration (1) Data Analytics Studio (44) HBase (9) HDFS (8) YARN (1)

Knox (18) Livy (4) Livy for Spark 3 (6) Cloudera Management Service (7) Oozie (3) Ozone (56) YARN (1)

Hive on Tez (5) Spark (5) Spark 3 (2) Zeppelin (7) ZooKeeper (2)

| | |
|------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <p>Ozone Service ID ozone.service.id ozone.service.id</p> | <p>sjc2-cdip > Ozone (Service-Wide) ↶</p> <input type="text" value="ozone"/> |
| <p>Ozone SCM Service ID ozone.scm.service.id ozone.scm.service.id</p> | <p>sjc2-cdip > Ozone (Service-Wide)</p> <input type="text" value="scm1"/> |
| <p>Ozone SCM Primordial Node ID ozone.scm.primordial.node.id ozone.scm.primordial.node.id</p> | <p>sjc2-cdip > Ozone (Service-Wide) Undo</p> <input type="text" value="cdipnn03.sjc-cdip.cisco.local"/> |

Step 22. Configure Kerberos and Keep Review and customize the configuration changes based on your requirements.

Add Cluster - Configuration

- ✓ Select Services
- ✓ Assign Roles
- ✓ Setup Database
- ✓ Enter Required Parameters
- ✓ Review Changes
- 6 Configure Kerberos**
- 7 Command Details
- 8 Command Details
- 9 Summary

Configure Kerberos

Enable Kerberos for this cluster

Kerberos is a network authentication protocol that provides security for your cluster.

Install Kerberos client libraries on all hosts before proceeding.

```
# RHEL / CentOS
$ yum install krb5-workstation krb5-libs

# if Red Hat IPA is used as the KDC
$ yum install freeipa-client
```

```
# SUSE
$ zypper install krb5-client

# if Red Hat IPA is used as the KDC
$ zypper install freeipa-client
```

```
# Ubuntu
$ apt-get install krb5-user

# if Red Hat IPA is used as the KDC
$ apt-get install freeipa-client
```

Configure DataNode Ports

Configure the privileged ports required by DataNodes in a secure HDFS service.

DataNode Transceiver Port ⓘ

1004

DataNode HTTP Web UI Port ⓘ

1006

Step 23. Click Continue after Cloudera Manager successfully runs enable Kerberos command.

Add Cluster - Configuration

- Select Services
- Assign Roles
- Setup Database
- Enter Required Parameters
- Review Changes
- Configure Kerberos
- 7 Command Details**
- 8 Command Details
- 9 Summary

Command Details

Enable Kerberos Command

Status ✔ Finished Context [sjc2-cdip](#) Nov 3, 2:47:32 PM 2.2m

Successfully enabled Kerberos.

Completed 7 of 7 step(s).

Show All Steps Show Only Failed Steps Show Only Running Steps

- > ✔ Stop cluster
- > ⚠ Stop Cloudera Management Services
At least one role must be started.
- > ✔ Deploy krb5.conf
- > ✔ Configure all services to use Kerberos
- > ✔ Wait for credentials to be generated
- > ✔ Deploy client configuration
- > ✔ Start Cloudera Management Services

Step 24. Installation wizard run first command to start cluster roles and services. Click Continue.

Assign Roles

Setup Database

Enter Required Parameters

Review Changes

Configure Kerberos

Command Details

8 Command Details

9 Summary

First Run Command

Status ✔ **Finished** Context [sje2-cdip](#) Nov 3, 2:50:26 PM 8.8m

Finished First Run of the following services successfully: Core Configuration, ZooKeeper, HDFS, YARN Queue Manager, CDP-INFRA-SOLR, Ranger, HBase, Kafka, Knox, Ozone, YARN, Atlas, Tez, Hive, Spark 3, Spark, Hive on Tez, Impala, Livy, Livy for Spark 3, Oozie, Data Analytics Studio, Hue, Zeppelin, Cloudera Management Service.

Completed 1 of 1 step(s).

Show All Steps Show Only Failed Steps Show Only Running Steps

| | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------|
| <ul style="list-style-type: none"> Run a set of services for the first time. Successfully completed 22 steps. | Nov 3, 2:50:26 PM | 8.8m |
| <ul style="list-style-type: none"> Execute 13 steps in sequence. Successfully completed 22 steps. <ul style="list-style-type: none"> Ensuring that the expected software releases a... Waiting for credentials to be generated Execute 7 steps in parallel Execute 4 steps in parallel Execute 23 steps in parallel Execute 10 steps in parallel Execute 3 steps in parallel Execute 2 steps in parallel Execute 4 steps in parallel Execute 3 steps in parallel Execute 2 steps in parallel Verifying successful startup of services Execute 7 steps in parallel Execute 4 steps in parallel Execute 23 steps in parallel Execute 10 steps in parallel Execute 3 steps in parallel Execute 2 steps in parallel Execute 4 steps in parallel Execute 3 steps in parallel Execute 2 steps in parallel Execute 2 steps in parallel Verifying successful startup of services | Nov 3, 2:50:31 PM | 69.01s |
| | Nov 3, 2:51:40 PM | 35.34s |
| | Nov 3, 2:52:16 PM | 85.43s |
| | Nov 3, 2:53:41 PM | 99.09s |
| | Nov 3, 2:55:20 PM | 64.68s |
| | Nov 3, 2:56:25 PM | 29.62s |
| | Nov 3, 2:56:54 PM | 38.51s |
| | Nov 3, 2:57:33 PM | 35.19s |
| | Nov 3, 2:58:08 PM | 33.86s |
| | Nov 3, 2:58:42 PM | 31.01s |
| | Nov 3, 2:59:13 PM | 143ms |

[← Back](#)
[Continue →](#)

Step 25. Click Finish on the Summary page.

CLUSTER
Manager

Search

- Clusters
- Hosts
- Diagnostics
- Audits
- Charts
- Replication
- Administration
- Data Services New

Add Cluster - Configuration

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

Summary

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

Note: You might need to adjust configuration parameters of the cluster after successful first run command execution. Apply the changes and restart the cluster.

Scale the Cluster

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

Note: For production clusters, it is recommended to set up Name Node and Resource manager in High Availability mode.

This implies that there will be at least 3 master nodes, running the Name Node, YARN Resource manager, the failover counterpart being designated to run on another node and a third node that would have similar capacity as the other two nodes.

All the three nodes will also need to run zookeeper and quorum journal node services. It is also recommended to have a minimum of 8 Data Nodes in a cluster. Please refer to the next section for details on how to enable HA.

Enable High Availability

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

Ozone Manager High Availability

Configuring High Availability (HA) for the Ozone Manager (OM) enables you to run redundant Ozone Managers in your Ozone cluster and prevents the occurrence of a single point of failure in the cluster from the perspective of namespace management. In addition, Ozone Manager HA ensures continued interactions with the client applications for read and write operations.

Ozone Manager HA involves a leader OM that handles read and write requests from the client applications, and at least two follower OMs, one of which can take over as the leader in situations such as:

- Unplanned events such as a crash involving the node that contains the leader OM.
- Planned events such as a hardware or software upgrade on the node that contains the leader OM

A High Availability (HA) configuration of the Ozone Manager (OM) involves one leader OM node and two or more follower nodes. The leader node services read and write requests from the client. The follower nodes closely keep track of the updates made by the leader so that in the event of a failure, one of the follower nodes can take over the operations of the leader

```
# ozone admin om getserviceroles -id=ozone
om1 : FOLLOWER (cdipnn01.sjc-cdip.cisco.local)
om3 : FOLLOWER (cdipnn03.sjc-cdip.cisco.local)
om2 : LEADER (cdipnn02.sjc-cdip.cisco.local)
```

Note: For more information visit, [Considerations for configuring High Availability on the Ozone Manager.](#)

Storage Container Manager High Availability

Configuring HA for the Storage Container Manager (SCM) prevents the occurrence of a single point of failure in an Ozone cluster to manage the various types of storage metadata and ensures continued interactions of the SCM with the Ozone Manager (OM) and the DataNodes.

SCM HA involves the following:

- A leader SCM that interacts with the OM for block allocations and works with the DataNodes to maintain the replication levels required by the Ozone cluster.
- At least two follower SCMs that closely keep track of the updates made by the leader so that in the event of a failure, one of the follower nodes can take over the operations from the leader.

Note: For more information visit, [Considerations for configuring High Availability on Storage Container Manager](#) and [Storage Container Manager operations in High Availability](#)

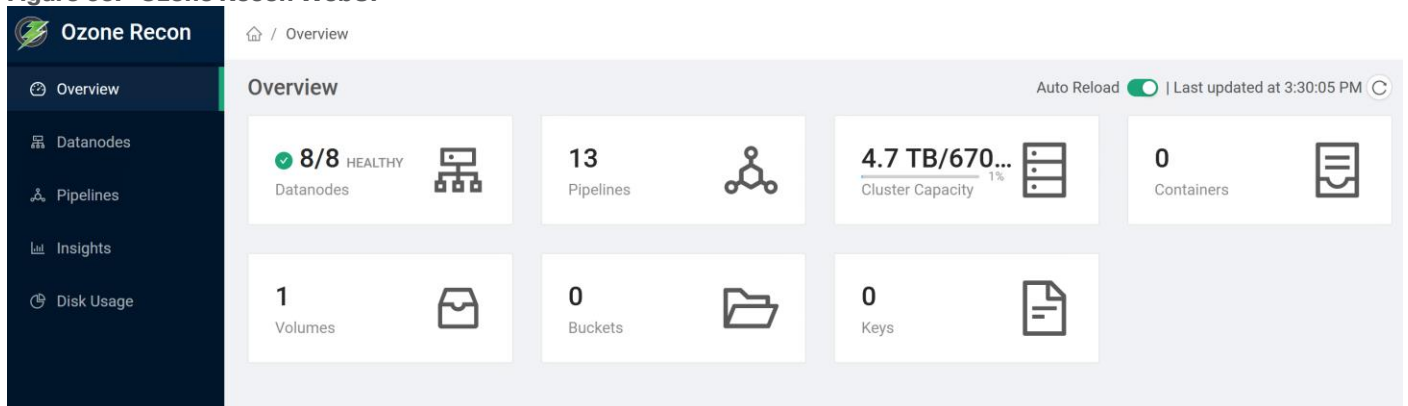
Recon Web User Interface

Recon is a centralized monitoring and management service within an Ozone cluster that provides information about the metadata maintained by different Ozone components such as the Ozone Manager (OM) and the Storage Container Manager (SCM).

Recon keeps track of the metadata as the cluster is operational and displays the relevant information through a dashboard and different views on the Recon web user interface. This information helps in understanding the overall state of the Ozone cluster.

The metadata that components such as OM and SCM maintain are quite different from one another. For example, OM maintains the mapping between keys and containers in an Ozone cluster while SCM maintains information about containers, Data Nodes, and pipelines. The Recon web user interface provides a consolidated view of all these elements.

Figure 53. Ozone Recon WebUI



For more information, go to: <https://docs.cloudera.com/cdp-private-cloud-base/7.1.8/ozone-storing-data/topics/ozone-recon-web-ui.html>

Configure Ozone to Work with Prometheus

You can configure your Ozone cluster to enable Prometheus for real time monitoring of the cluster.

To enable Prometheus to work on your Ozone cluster, you must download the required binary to a specific parcel directory and use Cloudera Manager to add the Ozone Prometheus role instance.

Download the Prometheus binary from <https://github.com/prometheus/prometheus/releases/tag/v2.16.0> and untar it to the following internal parcel directory on the host where you want Prometheus installed:

```
# /opt/cloudera/parcels/CDH/lib/hadoop-ozone/share/
```

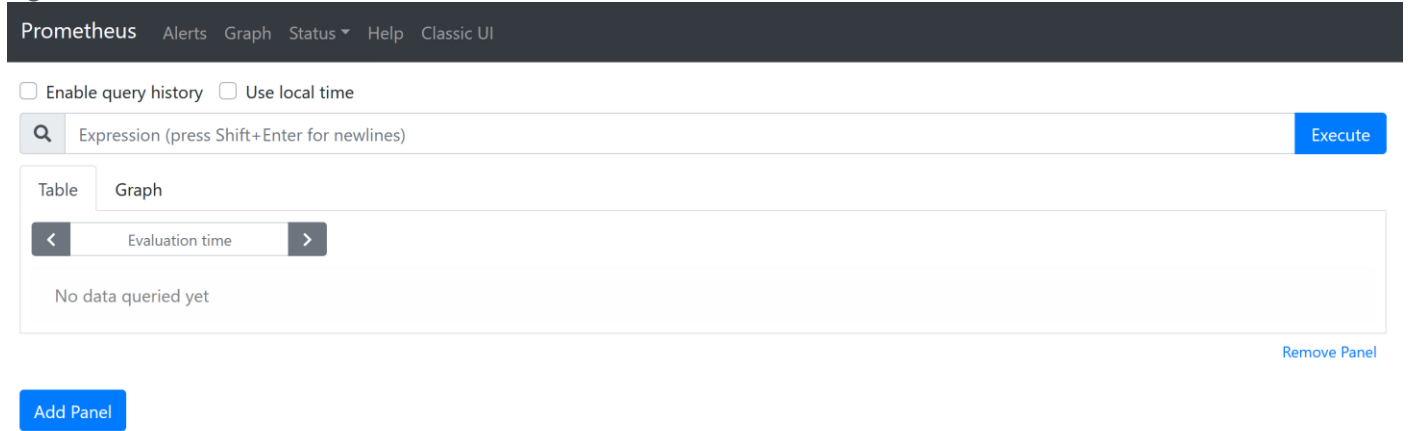
Using Cloudera Manager, add the Ozone Prometheus role instance to the Ozone service.

If you do not see Ozone Prometheus in the list of role instances to configure, it means that the role instance is not configured correctly. In this situation, the Prometheus logs (/var/log/hadoop-ozon/ozon-prometheus.log) on the Prometheus instance host show a FileNotFound error.

Start the Ozone Prometheus Role Instance

For detailed information, go to: <https://docs.cloudera.com/cdp-private-cloud-base/7.1.8/ozon-storing-data/topics/ozon-prometheus-config.html>

Figure 54. Ozone Prometheus WebUI



Change Ozone Metadata Directory

ozon.metadata.dirs allows Administrators to specify where the metadata must reside. Usually you select your fastest disk (SSD if you have them on your nodes). OzoneManager, SCM and datanode will write the metadata to this path.

In the development/test environment, users configure all the metadata directory with a single location, also known as All-In-One location for simplicity. In production environments, individual services such as Ozone Manager, Storage Container Manager and Data Node can set up dedicated NVMe for metadata for best performance.

Figure 55. Change Directory Configuration on Cloudera Manager for Ozone Metadata

Q metadata.dir

Filters

▼ SCOPE

- Ozone (Service-Wide) 0
- Gateway 0
- Ozone DataNode 1
- Ozone Manager 1
- Ozone Prometheus 0
- Ozone Recon 1
- Storage Container Manager 1
- S3 Gateway 0

▼ CATEGORY

- Main 4
- Advanced 0
- Logs 0
- Monitoring 0
- Performance 0

| | |
|--------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Datanode Metadata Directory</p> <p>ozone.metadata.dirs ozone.metadata.dirs</p> | <p>Ozone DataNode Default Group Undo</p> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;">/ozone/nvme/hadoop-ozone/datanode/ozone-metadata</div> |
| <p>Ozone Manager Metadata Directory</p> <p>ozone.metadata.dirs ozone.metadata.dirs</p> | <p>Ozone Manager Default Group ↩</p> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;">/ozone/nvme/hadoop-ozone/om/ozone-metadata</div> |
| <p>Recon Metadata Directory</p> <p>ozone.metadata.dirs ozone.metadata.dirs</p> | <p>Ozone Recon Default Group ↩</p> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;">/ozone/nvme/hadoop-ozone/recon/ozone-metadata</div> |
| <p>Storage Container Manager Metadata Directory</p> <p>ozone.metadata.dirs ozone.metadata.dirs</p> | <p>Storage Container Manager Default Group ↩</p> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;">/ozone/nvme/hadoop-ozone/scm/ozone-metadata</div> |

Configure Dedicated Ozone Service Metadata DB Directories

For production environments, we recommend each Ozone component to have its own metadata in RockDB stored at a dedicated location and preferably on NVMe for best performance.

Figure 56. Dedicated Ozone Service Metadata DB Directories

The screenshot displays the Cloudera Manager WebUI configuration page for Ozone Service Metadata DB Directories. At the top, there is a search bar containing 'db.dir'. Below the search bar, the interface is divided into a left sidebar with filters and a main content area with a table of configurations.

Filters:

- SCOPE:**
 - Ozone (Service-Wide): 0
 - Gateway: 0
 - Ozone DataNode: 0
 - Ozone Manager: 1
 - Ozone Prometheus: 1
 - Ozone Recon: 3
 - Storage Container Manager: 1
 - S3 Gateway: 0
- CATEGORY:**
 - Main: 6
 - Advanced: 0
 - Logs: 0
 - Monitoring: 0
 - Performance: 0
 - Ports and Addresses: 0
 - Resource Management: 0
 - Security: 0
 - Stacks Collection: 0
- STATUS:**
 - Error: 0
 - Warning: 0

Table of Configurations:

| Directory Name | Default Group | Path |
|--------------------------------------------------------------------------------------------------------------------|-----------------------------------------|------------------------------------------|
| Ozone Manager Data Directory ozone.om.db.dirs ozone.om.db.dirs | Ozone Manager Default Group | /ozone/nvme/hadoop-ozone/om/data |
| Prometheus Data Directory ozone.prometheus.db.dir ozone.prometheus.db.dir | Ozone Prometheus Default Group | /ozone/nvme/hadoop-ozone/prometheus/data |
| Recon Data Directory ozone.recon.db.dir ozone.recon.db.dir | Ozone Recon Default Group | /ozone/nvme/hadoop-ozone/recon/data |
| Recon StorageContainerManager Data Directory ozone.recon.scm.db.dirs ozone.recon.scm.db.dirs | Ozone Recon Default Group | /ozone/nvme/hadoop-ozone/recon/scm/data |
| Recon OzoneManager Data Directory ozone.recon.om.db.dir ozone.recon.om.db.dir | Ozone Recon Default Group | /ozone/nvme/hadoop-ozone/recon/om/data |
| Storage Container Manager Data Directory ozone.scm.db.dirs ozone.scm.db.dirs | Storage Container Manager Default Group | /ozone/nvme/hadoop-ozone/scm/data |

Procedure 10. Change the Ozone Directory configuration to NVMe

- Step 1.** Log into the Cloudera Manager WebUI and click My Clusters.
- Step 2.** From the configuration drop-down list select All “var/log” and “/var/lib” Directories.
- Step 3.** Click Save.

CDS 3.3 Powered by Apache Spark

Apache Spark is a general framework for distributed computing that offers high performance for both batch and interactive processing. It exposes APIs for Java, Python, and Scala. This document describes CDS 3.3 Powered by Apache Spark. It enables you to install and evaluate the features of Apache Spark 3 without upgrading your CDP Private Cloud Base cluster. For detailed API information, see the [Apache Spark project site](#).

CDS 3.3 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, distributed as a parcel and the Cloudera Service Descriptor file is available in Cloudera Manager for CDP 7.1.8. On CDP Private Cloud Base, a Spark 3 service can coexist with the existing Spark 2 service. The configurations of the two services do not conflict and both services use the same YARN service. The port of the Spark History Server is 18088 for Spark 2 and 18089 for Spark 3. Spark 3 installs and uses its own external shuffle service.

Note: Spark 3.3 is the first Spark version with the log4j2 dependency. Previous versions contained the log4j1 dependency. If you are using any custom logging related changes, you must rewrite the original log4j properties’ files using log4j2 syntax, that is, XML, JSON, YAML, or properties format.

Note: CDS 3.3 Powered by Apache Spark is an add-on service for CDP Private Cloud Base and is only supported with Cloudera Runtime 7.1.8 and higher. Spark 2 is included in CDP and does not require a separate parcel.

CDS 3 for GPUs

CDS 3.3 with GPU Support is an add-on service that enables you to take advantage of the RAPIDS Accelerator for Apache Spark to accelerate Apache Spark 3 performance on existing CDP Private Cloud Base clusters.

Unsupported Connectors

This release does not support the following connectors:

- SparkR
- Oozie
- Zeppelin

Limitations of Spark in CDP

Limitations of Spark (in comparison to Apache Spark 3.3) in CDP are as follows:

- spark.sql.orc.compression.codec config doesn't accept zsdt value.
- spark.sql.avro.compression.codec config doesn't accept zstandard value.
- Specifying avroSchemaUrl is not supported in datasource options.

For more details, go to: <https://docs.cloudera.com/cdp-private-cloud-base/7.1.8/cds-3/topics/spark-spark-3-overview.html>

CDS 3.3 Powered by Apache Spark Requirements for GPUs

Each cluster host must have the following software installed:

- Java – JDK 8 or JDK 11.
- Scala – Scala 2.12
- Python – Python 3.7 and higher

Note: Cloudera recommends using JDK 8, as most testing has been done with JDK 8. Remove other JDK versions from all cluster and gateway hosts to ensure proper operation.

In addition to the requirements for CDS 3.3; each host with a GPU must have the following software installed:

- GPU drivers v450.80.02 or higher and CUDA version 11.0 or higher
 - Download and install the [CUDA Toolkit](#) according to the operating system. The toolkit installer also provides the option to install the GPU driver.
- NVIDIA Library
 - NVIDIA RAPIDS version 22.06. For more information, see [NVIDIA Release Notes](#)
- UCX (Optional)
 - Clusters with Infiniband or RoCE networking can leverage [Unified Communication X](#) (UCX) to enable the [RAPIDS Shuffle Manager](#). For information on UCX native libraries support, see [\(Optional\) Installing UCX native libraries](#).

Note: CDS 3.3 with GPU support requires cluster hosts with NVIDIA Pascal or better GPUs, with a [compute capability](#) rating of 6.0 or higher. For more information, see [Getting Started](#) at the RAPIDS website.

Note: Cloudera and NVIDIA recommend using NVIDIA-certified systems. For more information, see [NVIDIA-Certified Systems](#) in the NVIDIA GPU Cloud documentation.

Procedure 1. Install JDK 8

Download jdk8 for Linux: <https://www.oracle.com/java/technologies/javase/javase8u211-later-archive-downloads.html#license-lightbox>

Step 1. Install JDK8.

```
# ansible all -m shell -a "dnf install -y java-1.8.0-openjdk-devel java-1.8.0-openjdk-headless"
```

Procedure 2. Install Scala 2.12

Download Scala 2.12: <https://downloads.lightbend.com/scala/2.12.17/scala-2.12.17.rpm>

Step 1. Download Scala 2.12 and copy to admin nodes(s).

```
# scp scala-2.12.17.rpm cdipnn01:/root/  
# ansible all -m copy -a "src=/root/scala-2.12.17.rpm dest=/root/."
```

Step 2. Install Scala 2.12.

```
# ansible all -m command -a "yum install -y /root/scala-2.12.17.rpm"
```

Procedure 3. Install NVIDIA CUDA Toolkit

Prerequisites

To use NVIDIA CUDA on your system, you will need the following installed:

- CUDA-capable GPU
- A supported version of Linux with a gcc compiler and toolchain
- CUDA Toolkit (available at <https://developer.nvidia.com/cuda-downloads>)

For more information on supported Linux distribution review to Table 1 in the CUDA Toolkit documentation: <https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html>

Prior to installation, verify the Linux version and kernel is supported, CUDA-capable GPU is installed, and gcc is installed on the system.

```
# ansible datanodes -m shell -a "lspci | grep -i nvidia"  
cdipdn06.sjc-cdip.cisco.local | CHANGED | rc=0 >>  
0000:98:00.0 3D controller: NVIDIA Corporation GA100 [A100 PCIe 80GB] (rev a1)  
  
# ansible datanodes -m shell -a "uname -a"  
cdipdn08.sjc-cdip.cisco.local | CHANGED | rc=0 >>  
Linux cdipdn01.sjc-cdip.cisco.local 4.18.0-372.9.1.el8.x86_64 #1 SMP Fri Apr 15 22:12:19 EDT 2022 x86_64  
x86_64 x86_64 GNU/Linux  
  
# ansible datanodes -m shell -a "uname -m && cat /etc/*release"  
cdipdn05.sjc-cdip.cisco.local | CHANGED | rc=0 >>  
x86_64  
NAME="Red Hat Enterprise Linux"  
VERSION="8.6 (Ootpa)"  
ID="rhel"  
ID_LIKE="fedora"  
VERSION_ID="8.6"  
PLATFORM_ID="platform:el8"
```



```

PRETTY_NAME="Red Hat Enterprise Linux 8.6 (Ootpa)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:redhat:enterprise_linux:8::baseos"
HOME_URL="https://www.redhat.com/"
DOCUMENTATION_URL="https://access.redhat.com/documentation/red_hat_enterprise_linux/8/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"

# ansible datanodes -m shell -a "gcc --version"
cdipdn01.sjc-cdip.cisco.local | CHANGED | rc=0 >>
gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-10)

#### Run following command to update/install GCC ####
# dnf install gcc

#### Alternatively Development Tools package will also install additional libraries as well as the g++
compiler
# dnf groupinstall "Development Tools"

#### Run following command to install kernel-header and/or kernel-devel ####
# sudo dnf install kernel-devel-$(uname -r) kernel-headers-$(uname -r)

# ansible datanodes -m shell -a "uname -r"
cdipdn04.sjc-cdip.cisco.local | CHANGED | rc=0 >>
4.18.0-372.9.1.el8.x86_64

#####Enable optional repos - On RHEL 8 Linux only, execute the following steps to enable optional
repositories#####

# subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
# subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
# subscription-manager repos --enable=codeready-builder-for-rhel-8-x86_64-rpms

```

The CUDA Toolkit can be installed using either of two different installation mechanisms: distribution-specific packages (RPM and Deb packages), or a distribution-independent package (runfile packages).

The distribution-independent package has the advantage of working across a wider set of Linux distributions but does not update the distribution's native package management system. The distribution-specific packages interface with the distribution's native package management system. It is recommended to use the distribution-specific packages, where possible.

Note: Standalone installers are not provided for architectures other than the x86_64 release. For both native as well as cross development, the toolkit must be installed using the distribution-specific installer.

Note: [Disable Nouveau Driver](#) by creating “/etc/modprobe.d/blacklist-nouveau.conf” with the following contents:

```

# vi /etc/modprobe.d/blacklist-nouveau.conf
blacklist nouveau
options nouveau modeset=0

```

Step 1. Download Nvidia CUDA Toolkit by selecting target platform from the link: https://developer.nvidia.com/cuda-11.0-update1-download-archive?target_os=Linux&target_arch=x86_64&target_distro=RHEL&target_version=8&target_type=rpmlocal

CUDA Toolkit 11.4 Downloads

Home

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

| | | | | | | | | |
|------------------|-----------------------------|-------------------------------|---------------------------------|--------------------------|----------------------|----------------------|------------------------|----------------------------|
| Operating System | Linux | Windows | | | | | | |
| Architecture | x86_64 | ppc64le | arm64-sbsa | | | | | |
| Distribution | CentOS | Debian | Fedora | OpenSUSE | RHEL | SLES | Ubuntu | WSL-Ubuntu |
| Version | 7 | 8 | | | | | | |
| Installer Type | rpm (local) | rpm (network) | runfile (local) | | | | | |

Download Installer for Linux RHEL 8 x86_64

The base installer is available for download below.

>Base Installer

Installation Instructions:

```
$ wget https://developer.download.nvidia.com/compute/cuda/11.4.0/local_installers/cuda-repo-rhel8-11-4-local-11.4.0_470.42.01-1.x86_64.rpm
$ sudo rpm -i cuda-repo-rhel8-11-4-local-11.4.0_470.42.01-1.x86_64.rpm
$ sudo dnf clean all
$ sudo dnf -y module install nvidia-driver:latest-dkms
$ sudo dnf -y install cuda
```

Step 2. Copy CUDA Toolkit to nodes with GPU(s).

```
# wget https://developer.download.nvidia.com/compute/cuda/11.4.4/local_installers/cuda-repo-rhel8-11-4-local-11.4.4_470.82.01-1.x86_64.rpm
# scp cuda-repo-rhel8-11-4-local-11.4.4_470.82.01-1.x86_64.rpm cdipnn01:/root/
# ansible datanodes -m copy -a "src=/root/cuda-repo-rhel8-11-4-local-11.4.4_470.82.01-1.x86_64.rpm dest=/root/."
```

Step 3. Install CUDA Toolkit.

```
# ansible datanodes -m command -a "rpm -ivh /root/cuda-repo-rhel8-11-4-local-11.4.4_470.82.01-1.x86_64.rpm"
# ansible datanodes -a "dnf clean all"
# ansible datanodes -m command -a "dnf -y module install nvidia-driver:latest-dkms"
# ansible datanodes -m command -a "dnf -y install cuda"
```

Step 4. The toolkit installer also provides the option to install the GPU driver. Additionally, download the latest Nvidia Driver for Linux RHEL 8 by selecting the appropriate selection from the drop-down list and click Search: <https://www.nvidia.com/Download/index.aspx?lang=en-us>

NVIDIA Driver Downloads

Select from the dropdown list below to identify the appropriate driver for your NVIDIA product.

| | | |
|-------------------|---------------------|---|
| Product Type: | Data Center / Tesla | ▼ |
| Product Series: | A-Series | ▼ |
| Product: | NVIDIA A100 | ▼ |
| Operating System: | Linux 64-bit RHEL 8 | ▼ |
| CUDA Toolkit: | 11.4 | ▼ |
| Language: | English (US) | ▼ |

Search

Step 5. Click Download.

Data Center Driver For Linux RHEL 8

Version: 470.141.10
Release Date: 2022.10.19
Operating System: Linux 64-bit RHEL 8
CUDA Toolkit: 11.4
Language: English (US)
File Size: 259.77 MB

Download

Step 6. Click Agree & download.

Download

By clicking the **"Agree & Download"** button below, you are confirming that you have read and agree to be bound by the [License For Customer Use of NVIDIA Software](#) for use of the driver. The driver will begin downloading immediately after clicking on the **"Agree & Download"** button below. NVIDIA recommends users update to the latest driver version. Please review [NVIDIA Product Security](#) for more information.

Agree & Download

Decline

Step 7. Copy and upgrade Nvidia driver on data nodes with GPU.

```
# scp nvidia-driver-local-repo-rhel8-470.141.10-1.0-1.x86_64.rpm cdipnn01:/root/  
# ansible datanodes -m copy -a "src=/root/nvidia-driver-local-repo-rhel8-470.141.10-1.0-1.x86_64.rpm  
dest=/root/."  
# ansible datanodes -m shell -a "rpm -ivh nvidia-driver-local-repo-rhel8-470.141.10-1.0-1.x86_64.rpm"  
# ansible datanodes -m shell -a "dnf clean all"
```

```
# ansible datanodes -m shell -a "dnf -y module install nvidia-driver:latest-dkms"
# ansible datanodes -m shell -a "dnf install -y cuda"
# ansible datanodes -m shell -a "reboot"
```

Step 8. Reboot nodes.

```
# ansible datanodes -m command -a "reboot"
```

Step 9. After reboot, run “nvidia-smi” to list available GPU device(s).

```
[root@cdipdn06 ~]# nvidia-smi
Mon Nov 14 14:26:27 2022
+-----+
| NVIDIA-SMI 470.141.10      Driver Version: 470.141.10   CUDA Version: 11.4   |
+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                           MIG M. |
+-----+-----+-----+
|    0   NVIDIA A100 80G...  Off   | 00000000:98:00.0 Off |             0         |
| N/A   38C    P0      64W / 300W |  0MiB / 80994MiB |      3%    Default   |
|                                           Disabled  |
+-----+-----+-----+
+-----+
| Processes: |
| GPU   GI    CI          PID    Type   Process name          GPU Memory |
|          ID    ID                          |            | Usage      |
+-----+-----+-----+
| No running processes found |
+-----+
```

Procedure 4. Post-installation Steps

Step 1. Add this path to the PATH variable:

```
# ansible datanodes -m shell -a "export PATH=/usr/local/cuda-11.4/bin${PATH:+:${PATH}}"
```

Step 2. In addition, when using the runfile installation method, change LD_LIBRARY_PATH environment variables for 64-bit operating systems:

```
# ansible datanodes -m shell -a "export LD_LIBRARY_PATH=/usr/local/cuda-11.4/lib64\${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}"
```

```
#### For 32-bit operating system run below command ####
#export LD_LIBRARY_PATH=/usr/local/cuda-11.4/lib\${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

Note: These paths change when using a custom install path with the runfile installation method. After reboot run “nvidia-smi” to list available GPU device(s). Refer to the CUDA Toolkit post-installation actions: <https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#post-installation-actions>

```
# ansible datanodes -m shell -a "cat /proc/driver/nvidia/version"
cdipdn02.sjc-cdip.cisco.local | CHANGED | rc=0 >>
NVRM version: NVIDIA UNIX x86_64 Kernel Module 470.141.10 Thu Sep 22 00:43:55 UTC 2022
GCC version: gcc version 8.5.0 20210514 (Red Hat 8.5.0-10) (GCC)

# nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2021 NVIDIA Corporation
Built on Mon_Oct_11_21:27:02_PDT_2021
Cuda compilation tools, release 11.4, V11.4.152
Build cuda_11.4.r11.4/compiler.30521435_0

# gcc --version
gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-10)
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
# ldd --version
ldd (GNU libc) 2.28
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.
```

Enable CDS 3.3 with GPU Support

To activate the CDS 3.3 with GPU Support feature on suitable hardware, you need to create a Yarn role group and optionally make configuration changes to enable the NVIDIA RAPIDS Shuffle Manager.

Prerequisites

Refer to section [Setup Parcels for CDS 3.3 powered by Apache Spark](#) to install Spark3 in existing CDP PvC Base cluster or in a new cluster deployment.

Procedure 1. Enable CDS 3.3 with GPU support

Step 1. Check that all the software prerequisites are satisfied. If not, you might need to upgrade or install other software components first.

Step 2. In the Cloudera Manager Admin Console, add the CDS parcel repository to the Remote Parcel Repository URLs in Parcel Settings as described in Parcel Configuration Settings.

Note: If your Cloudera Manager Server does not have Internet access, you can use the CDS Powered by Apache Spark parcel files: put them into a new parcel repository, and then configure the Cloudera Manager Server to target this newly created repository. Download the CDS 3.3 parcel, distribute the parcel to the hosts in your cluster, and activate the parcel. For instructions, see Managing Parcels.

Procedure 2. NVIDIA Library

The RAPIDS Accelerator for Apache Spark leverages GPUs to accelerate processing via the [RAPIDS libraries](#).

The RAPIDS Accelerator for Apache Spark combines the power of the [RAPIDS cuDF](#) library and the scale of the Spark distributed computing framework. The RAPIDS Accelerator library also has a built-in accelerated shuffle based on [UCX](#) that can be configured to leverage GPU-to-GPU communication and RDMA capabilities.

Existing Apache Spark applications with no code change. Launch Spark with the RAPIDS Accelerator for Apache Spark plugin jar and enable a configuration setting:

```
spark.conf.set('spark.rapids.sql.enabled','true')
```

If you plan to convert existing Spark workload from CPU to GPU, please refer to [Spark workload qualification](#) to check if your Spark Applications are good fit for the RAPIDS Accelerator for Apache Spark.

The RAPIDS Accelerator for Apache Spark requires each worker node in the cluster to have [CUDA](#) installed.

The RAPIDS Accelerator for Apache Spark consists of two jars: a plugin jar along with the RAPIDS cuDF jar, which is either preinstalled in the Spark classpath on all nodes or submitted with each job that uses the RAPIDS Accelerator for Apache Spark. For more details, please visit [Getting Started with the RAPIDS Accelerator for Apache Spark](#)

CDS 3.3 requires NVIDIA RAPIDS version 22.06. For more details, visit [CDS for GPUs software requirement](#).

For more information, go to [NVIDIA release notes for spark-rapids version 22.06.0](#)

To leverage Nvidia RAPIDS, YARN and the Spark executors have to be able to access the Spark RAPIDS libraries. The required jars are present in [Getting Started with RAPIDS Accelerator with on premise cluster or local mode](#).

Step 1. Download NVIDIA RAPIDS version 22.06 from the link: <https://nvidia.github.io/spark-rapids/docs/archive.html#download-v22060>

```
Download - RAPIDS Accelerator for Apache Spark 22.06.0 jar
# wget https://repol.maven.org/maven2/com/nvidia/rapids-4-spark_2.12/22.06.0/rapids-4-spark_2.12-22.06.0-cuda11.jar

Download - RAPIDS Accelerator for Apache Spark 22.06.0 jars.asc
# wget https://repol.maven.org/maven2/com/nvidia/rapids-4-spark_2.12/22.06.0/rapids-4-spark_2.12-22.06.0-cuda11.jar.asc

Download - PUB KEY
# wget https://keys.openpgp.org/vks/v1/by-fingerprint/7A8A39909B9B202410C2A26F1D9E1285654392EF
```

Note: This package is built against CUDA 11.5 and all CUDA 11.x versions are supported through [CUDA forward compatibility](#). It is tested on V100, T4, A2, A10, A30 and A100 GPUs with CUDA 11.0-11.5. You will need to ensure the minimum driver (450.80.02) and CUDA toolkit are installed on each Spark node.

Note: For more information, go to [Spark 3 GPU Scheduling](#)

Step 2. Run below commands to verify signature.

```
# gpg --import 7A8A39909B9B202410C2A26F1D9E1285654392EF
gpg: directory '/root/.gnupg' created
gpg: keybox '/root/.gnupg/pubring.kbx' created
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 1D9E1285654392EF: public key "NVIDIA Spark (For the signature of spark-rapids release jars) <sw-spark@nvidia.com>" imported
gpg: Total number processed: 1
gpg:             imported: 1

# gpg --verify rapids-4-spark_2.12-22.06.0.jar.asc rapids-4-spark_2.12-22.06.0.jar
gpg: Signature made Fri 17 Jun 2022 06:51:03 AM PDT
gpg:             using RSA key 1D9E1285654392EF
gpg: Good signature from "NVIDIA Spark (For the signature of spark-rapids release jars) <sw-spark@nvidia.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:             There is no indication that the signature belongs to the owner.
Primary key fingerprint: 7A8A 3990 9B9B 2024 10C2 A26F 1D9E 1285 6543 92EF
```

Step 3. Copy rapids-4-spark jar and getGpuResources.sh script to admin node.

```
# scp rapids-4-spark_2.12-22.06.0-cuda11.jar cdipnn01:/root/
```

Step 4. The cuDF jar is now bundled with the rapids-4-spark jar and should not be specified. Combined cuDF jar and rapids-4-spark jar to a single rapids-4-spark jar. The RAPIDS Accelerator jar (rapids-4-spark jar) is the only jar that needs to be passed to Spark. Download “rapids-4-spark jar”

Step 5. Create directory “/opt/sparkRapidsPlugin/.” Copy files downloaded in step 1.

```
# ansible datanodes -m shell -a "mkdir -p /opt/sparkRapidsPlugin/"
# ansible datanodes -m copy -a "src=/root/rapids-4-spark_2.12-22.06.0-cuda11.jar
dest=/opt/sparkRapidsPlugin/."
```

Step 6. [Download](#) and install [getGpusResources.sh](#) GPU Discovery script on all nodes.

```
# ansible datanodes -m copy -a "src=/root/getGpusResources.sh dest=/opt/sparkRapidsPlugin/."
# ansible datanodes -m file -a "dest=/opt/sparkRapidsPlugin/getGpusResources.sh mode=755"
# ansible datanodes -m shell -a "/opt/sparkRapidsPlugin/getGpusResources.sh"
```

Step 7. Export location to the above jar.

```
# ansible datanodes -m shell -a "export SPARK_RAPIDS_DIR=/opt/sparkRapidsPlugin"
# ansible datanodes -m shell -a "export SPARK_RAPIDS_PLUGIN_JAR=${SPARK_RAPIDS_DIR}/rapids-4-spark_2.12-22.06.0-cuda11.jar"
```

Procedure 1. Configure GPU scheduling and isolation

Prerequisite: YARN NodeManager must be installed with the Nvidia drivers.

Step 1. In Cloudera Manager, navigate to Hosts > Hosts Configuration. Search for cgroup.

Step 2. Select the Enable Cgroup-based Resource Management checkbox. Click Save Changes.

The screenshot shows the Cloudera Manager interface for Hosts Configuration. The left sidebar contains navigation options: Clusters, Hosts (selected), Diagnostics, Audits, Charts, Replication, Administration, Data Services (New), Parcels, Running Commands, Support, and admin. The main content area is titled 'Hosts Configuration' and has a search bar containing 'cgroup'. Below the search bar, there are 'Filters' and 'History & Rollback' options. The 'Filters' section shows a table with categories and statuses:

| CATEGORY | |
|---------------------|---|
| Advanced | 0 |
| Monitoring | 0 |
| Parcels | 0 |
| Resource Management | 1 |

| STATUS | |
|-------------------|---|
| Error | 0 |
| Warning | 0 |
| Edited | 1 |
| Non-Default | 1 |
| Include Overrides | 0 |

The main configuration area shows 'Enable Cgroup-based Resource Management' with a checked checkbox and an 'Undo' button. Below it, there is a link for 'Add Host Overrides'. The bottom of the page shows a summary: '1 Edited Value Reason for change: Modified Enable Cgroup-based Resource Management' and a 'Save Changes(CTRL+S)' button.

Step 3. Navigate to YARN > Configuration. Search for cgroup.

Step 4. Find the Use CGroups for Resource Management property and Always use Linux Container Executor; Click the checkbox to enable it for the applicable clusters.

The screenshot shows the Cloudera Manager interface. On the left is a dark sidebar with the Cloudera Manager logo and a search bar. Below the search bar are navigation icons for Clusters, Hosts, Diagnostics, Audits, Charts, Replication, Administration, Data Services (marked as 'New'), Parcels, Running Commands, Support, and a user profile for 'admin'. The main area is divided into sections: a top navigation menu with items like Monitoring, Performance, Ports and Addresses, Proxy, Resource Management (8), Resource Types, Security, Stacks Collection, and YARN Services Management (0); a 'STATUS' section showing 0 Errors, 0 Warnings, 2 Edited, 2 Non-Default, and 0 Include Overrides; and a configuration table for 'Docker Containers'. The table lists properties such as 'yarn.nodemanager.runtime.linux.docker.default-ro-mounts', 'docker.allowed.ro-mounts', 'yarn.docker.allowed.ro-mounts', 'yarn.nodemanager.linux-container-executor.resources-handler.class', 'yarn.service_cgroups', 'yarn.nodemanager.container-executor.class', and 'yarn.service_lce_always'. Some properties have checkboxes for 'YARN (Service-Wide)' and 'Undo' buttons. At the bottom, a summary bar indicates '2 Edited Values' and provides a 'Reason for change' field containing 'Modified Use CGroups for Resource Management, Always Use Linux Con' and a 'Save Changes(CTRL+S)' button.

Procedure 2. Set up a Yarn role group to enable GPU usage

- Step 1.** In Cloudera Manager, navigate to cluster > YARN > Configuration.
- Step 2.** Search for gpu.
- Step 3.** Find the Enable GPU Usage property and select the Node Manager Default Group checkbox.
- Step 4.** Find the Node Manager GPU Devices Allowed property and define the GPU devices that are managed by YARN using one of the following ways.
 - Use the default value “auto,” for auto detection of all GPU devices. In this case YARN manages all GPU devices.
 - Manually define the GPU devices that are managed by YARN.
- Step 5.** Find the NodeManager GPU Detection Executable property and define the location of nvidia-smi. By default, this property has no value, and it means that YARN checks the following paths to find nvidia-smi:
 - /usr/bin
 - /bin
 - /usr/local/nvidia/bin
- Step 6.** Click Save Changes.

The screenshot shows the Cloudera Manager interface for the YARN configuration of cluster 'sjc2-cdip'. The left sidebar contains navigation options like Clusters, Hosts, Diagnostics, Audits, Charts, Replication, Administration, Data Services, Parcels, Running Commands, and Support. The main content area is titled 'Configuration' and includes a search bar with 'gpu' entered. A 'Filters' panel on the left shows a tree view of configuration scopes and categories. The main configuration area lists several GPU-related settings:

- Enable GPU Usage:** Checked, with a link to `gpu_enabled`.
- NodeManager GPU Devices Allowed:** Set to 'auto', with a link to `yarn.nodemanager.resource-plugins.gpu.allowed-gpu-devices` and `gpu_plugin_allowed_devices`.
- NodeManager GPU Detection Executable:** Set to 'NodeManager Default Group', with a link to `yarn.nodemanager.resource-plugins.gpu.path-to-discovery-executables` and `gpu_plugin_detector_path`.

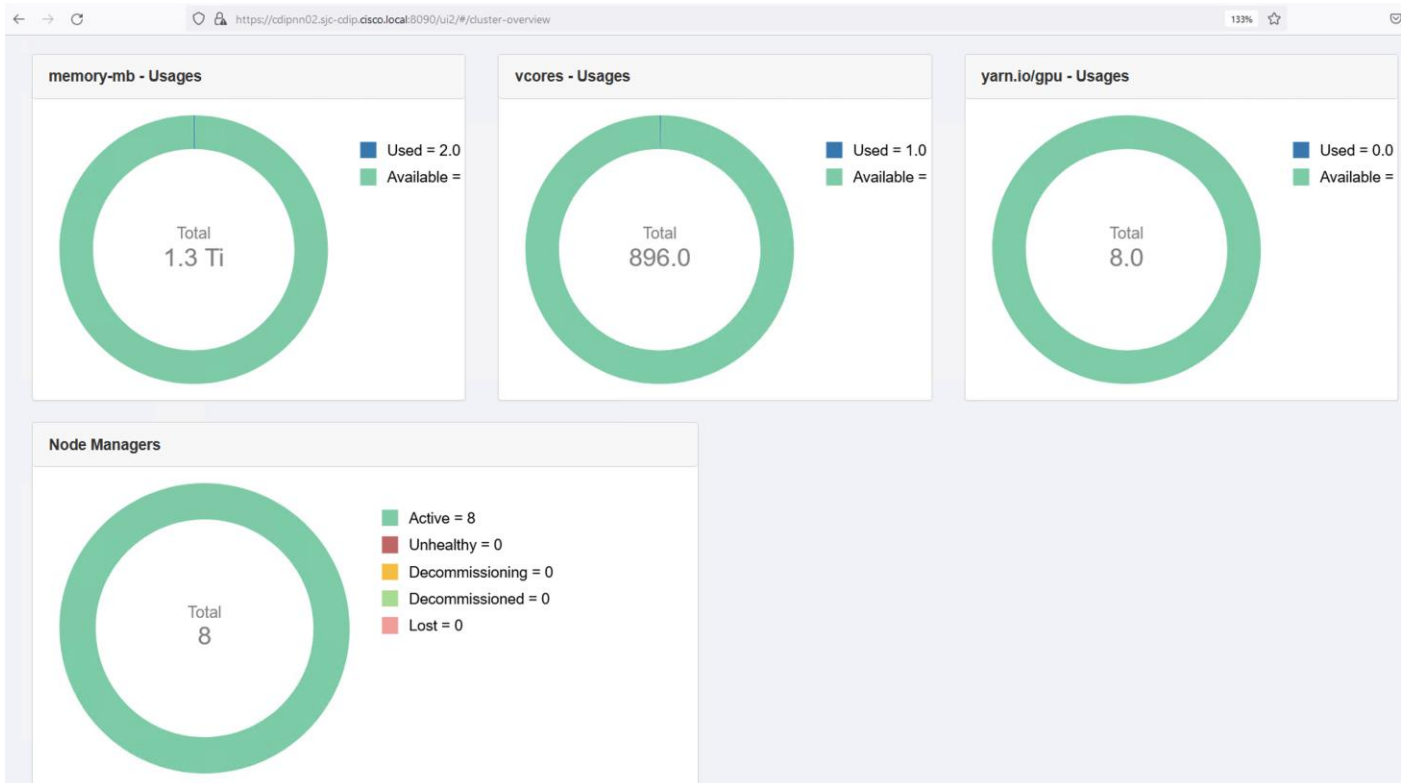
Step 7. Click the Stale Configuration: Restart needed button on the top of the page.

Step 8. Click Restart Stale Services.

Note: This step restarts all services with stale configurations.

Step 9. Select Re-deploy client configuration and click Restart.

Step 10. Validate GPU resources in Cloudera Manager, navigate to Cluster > YARN > WebUI > Resource Manager WebUI.



Procedure 3. Configure NVIDIA RAPIDS Shuffle Manager

The NVIDIA RAPIDS Shuffle Manager is a custom ShuffleManager for Apache Spark that allows fast shuffle block transfers between GPUs in the same host (over PCIe or NVLink) and over the network to remote hosts (over RoCE or Infiniband).

CDS 3.3 with GPU Support has built-in support for UCX, no separate installation is required.

The minimum UCX requirement for the RAPIDS Shuffle Manager is [UCX 1.11.2](#).

In order to enable the RAPIDS Shuffle Manager, UCX user-space libraries and its dependencies must be installed on the host and inside Docker containers (if not baremetal). A host has additional requirements, like the MLNX_OFED driver and `nv_peer_mem` kernel module.

Step 1. Fetch and install the UCX package. The UCX packages for CentOS 8 are divided into different RPMs. For example, UCX 1.13.1 available at <https://github.com/openucx/ucx/releases/download/v1.13.1/ucx-v1.13.1-centos8-mofed5-cuda11.tar.bz2>

```
ucx-devel-1.13.1-1.el8.x86_64.rpm
ucx-debuginfo-1.13.1-1.el8.x86_64.rpm
ucx-1.13.1-1.el8.x86_64.rpm
ucx-cuda-1.13.1-1.el8.x86_64.rpm
ucx-rdmacm-1.13.1-1.el8.x86_64.rpm
ucx-cma-1.13.1-1.el8.x86_64.rpm
ucx-ib-1.13.1-1.el8.x86_64.rpm
```

Step 2. Copy UCX 1.13.1 to admin node and GPU enabled datanodes.

```
# scp ucx-v1.13.1-centos8-mofed5-cuda11.tar.bz2 cdipnn01:/root/.
# ansible datanodes -m copy -a "src=/root/ucx-v1.13.1-centos8-mofed5-cuda11.tar.bz2 dest=/root/."
# ansible datanodes -m shell -a "tar -xvf ucx-v1.13.1-centos8-mofed5-cuda11.tar.bz2"
```

Step 3. The only packages required are:

```
# For a setup without RoCE or Infiniband networking
ucx-1.13.1-1.el8.x86_64.rpm
ucx-cuda-1.13.1-1.el8.x86_64.rpm
```

```
# If accelerated networking is available:
ucx-1.13.1-1.el8.x86_64.rpm
ucx-cuda-1.13.1-1.el8.x86_64.rpm
ucx-rdmacm-1.13.1-1.el8.x86_64.rpm
ucx-ib-1.13.1-1.el8.x86_64.rpm
```

Note: The CentOS RPM requires CUDA installed via RPMs to satisfy its dependencies.

Step 4. Install ucx rpm(s).

```
# ansible datanodes -m shell -a "rpm -ivh /root/ucx-1.13.1-1.el8.x86_64.rpm /root/ucx-devel-1.13.1-1.el8.x86_64.rpm /root/ucx-debuginfo-1.13.1-1.el8.x86_64.rpm /root/ucx-cuda-1.13.1-1.el8.x86_64.rpm /root/ucx-cuda-debuginfo-1.13.1-1.el8.x86_64.rpm /root/ucx-ib-1.13.1-1.el8.x86_64.rpm /root/ucx-ib-debuginfo-1.13.1-1.el8.x86_64.rpm /root/ucx-rdmacm-1.13.1-1.el8.x86_64.rpm /root/ucx-rdmacm-debuginfo-1.13.1-1.el8.x86_64.rpm /root/ucx-cma-1.13.1-1.el8.x86_64.rpm /root/ucx-cma-debuginfo-1.13.1-1.el8.x86_64.rpm"
```

Step 5. Test to check whether UCX can link against CUDA.

```
# ucx_info -d|grep cuda
# Memory domain: cuda_cpy
#   Component: cuda_cpy
#   Transport: cuda_copy
#   Device: cuda
# Memory domain: cuda_ipc
#   Component: cuda_ipc
#   Transport: cuda_ipc
#   Device: cuda
```

Step 6. execute “ucx_perftest” to test and validate.

```
# GPU ↔ GPU across the network, using GPUDirectRDMA
[root@cdipdn04 ~]# CUDA_VISIBLE_DEVICES=0 ucx_perftest -t tag_bw -s 10000000 -n 1000 -m cuda
[1668476753.916182] [cdipdn04:43820:0]          perftest.c:901 UCX WARN CPU affinity is not set (bound to
112 c
pus). Performance may be impacted.
Waiting for connection...
Accepted connection from 10.4.1.86:60116
+-----+
| API:          protocol layer                |
| Test:         tag match bandwidth          |
| Data layout:  (automatic)                  |
| Send memory:  cuda                         |
| Recv memory:  cuda                         |
| Message size: 10000000                     |
+-----+

[root@cdipdn06 ~]# CUDA_VISIBLE_DEVICES=0 ucx_perftest -t tag_bw -s 10000000 -n 1000 -m cuda cdipdn04
[1668476753.917539] [cdipdn06:42869:0]          perftest.c:901 UCX WARN CPU affinity is not set (bound to
112 c
pus). Performance may be impacted.
+-----+
|          |          |          |          |          |          |          |          |          |          |
|          |          |          |          |          |          |          |          |          |          |
| Stage   | # iterations | 50.0%ile | average | overall | average | overall | average | overall |
+-----+
[thread 0] | 317 | 3231.572 | 3162.763 | 3162.763 | 3015.32 | 3015.32 | 316 | 316
[thread 0] | 625 | 3256.638 | 3261.678 | 3211.509 | 2923.88 | 2969.55 | 307 | 311
[thread 0] | 933 | 3295.375 | 3259.530 | 3227.361 | 2925.80 | 2954.97 | 307 | 310
Final:    | 1000 | 3265.088 | 4814.628 | 3333.708 | 1980.78 | 2860.70 | 208 | 300
+-----+

# GPU ↔ GPU across the network, without GPUDirectRDMA
# UCX_IB_GPU_DIRECT_RDMA=no CUDA_VISIBLE_DEVICES=0 ucx_perftest -t tag_bw -s 10000000 -n 1000 -m cuda
[1668477016.701987] [cdipdn06:43609:0]          perftest.c:901 UCX WARN CPU affinity is not set (bound to
112 cpus). Performance may be impacted.
Waiting for connection...
Accepted connection from 10.4.1.84:60664
+-----+
| API:          protocol layer                |
| Test:         tag match bandwidth          |
| Data layout:  (automatic)                  |
| Send memory:  cuda                         |
| Recv memory:  cuda                         |
| Message size: 10000000                     |
+-----+

[root@cdipdn04 ~]# UCX_IB_GPU_DIRECT_RDMA=no CUDA_VISIBLE_DEVICES=0 ucx_perftest -t tag_bw -s 10000000 -n
1000 -m cuda cdipdn06
[1668477046.492472] [cdipdn04:44689:0]          perftest.c:901 UCX WARN CPU affinity is not set (bound to
112 cpus). Performance may be impacted.
+-----+
|          |          |          |          |          |          |          |          |          |          |
|          |          |          |          |          |          |          |          |          |          |
| Stage   | # iterations | 50.0%ile | average | overall | average | overall | average | overall |
+-----+
[1668477048.635951] [cdipdn04:44689:0]          parser.c:1911 UCX WARN unused env variable:
UCX_IB_GPU_DIRECT_RDMA (set UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[thread 0] | 319 | 3149.882 | 3150.530 | 3150.530 | 3027.03 | 3027.03 | 317 | 317
[thread 0] | 630 | 3160.279 | 3232.338 | 3190.914 | 2950.42 | 2988.72 | 309 | 313
[thread 0] | 941 | 3153.485 | 3230.157 | 3203.884 | 2952.41 | 2976.62 | 310 | 312
Final:    | 1000 | 3156.609 | 4984.848 | 3308.961 | 1913.15 | 2882.10 | 201 | 302
+-----+
```

Step 7. For Spark3, change “Enable Shuffle Service” and “Enable Dynamic Allocation” value to False.

```
# Enable Shuffle Service
spark.shuffle.service.enabled - False

# Enable Dynamic Allocation
spark.dynamicAllocation.enabled - False
```

Spark 3

Status Instances **Configuration** Commands Charts Library Audits History Server Web UI Quick Links

Q .enabled

Filters

SCOPE

| | |
|------------------------|---|
| Spark 3 (Service-Wide) | 1 |
| Gateway | 9 |
| History Server | 3 |

CATEGORY

| | |
|---------------------|----|
| Main | 11 |
| Advanced | 1 |
| Logs | 0 |
| Monitoring | 0 |
| Performance | 0 |
| Ports and Addresses | 0 |
| Resource Management | 0 |
| Security | 1 |
| Stacks Collection | 0 |

Persist Driver Logs to Dfs Spark 3 (Service-Wide)

spark.driver.log.persistToDfs.enabled
[spark_driver_log_persist_to_dfs](#)

Enable History Gateway Default Group

spark.eventLog.enabled
[spark_history_enabled](#)

Enable Shuffle Service Gateway Default Group

spark.shuffle.service.enabled
[spark_shuffle_service_enabled](#)

Enable Dynamic Allocation Gateway Default Group

spark.dynamicAllocation.enabled
[spark_dynamic_allocation_enabled](#)

For more information, go to <https://nvidia.github.io/spark-rapids/docs/additional-functionality/rapids-shuffle.html>

Procedure 4. Use GPU scheduling with distributed shell

You can run the distributed shell by specifying resources other than memory and vcores. The following is an example for distributed shell, but you can use GPU scheduling with other frameworks as well.

Use the following command to run the distributed shell and GPU without a Docker container:

```
# $SPARK_HOME/bin/spark3-shell \
  --master yarn \
  --conf spark.rapids.sql.concurrentGpuTasks=1 \
  --driver-memory 2G \
  --conf spark.executor.memory=16G \
  --conf spark.executor.cores=4 \
  --conf spark.executor.resource.gpu.amount=2 \
  --conf spark.task.cpus=1 \
  --conf spark.task.resource.gpu.amount=0.25 \
  --conf spark.rapids.memory.pinnedPool.size=2G \
  --conf spark.sql.files.maxPartitionBytes=512m \
  --conf spark.plugins=com.nvidia.spark.SQLPlugin \
  --conf spark.executor.resource.gpu.discoveryScript=./getGpusResources.sh \
  --files ${SPARK_RAPIDS_DIR}/getGpusResources.sh \
  --jars ${SPARK_RAPIDS_PLUGIN_JAR}
```

Conclusion

Cisco Data Intelligence Platform (CDIP) offers pre-validated designs both for data lake and private cloud. In these reference designs, Cisco achieved architectural innovation with partners. In addition to that, Cisco published various world record performance benchmarks with TPC (<http://www.tpc.org>) and proved linear scaling. Cisco published top performance numbers both for traditional map reduce and for Spark which is next generation of compute for crunching big data. Furthermore, CDIP offers centralized management with Cisco Intersight. Cisco Intersight innovation and addition of new features and capabilities is on the highest-gear which will bring lot of exciting innovation with the context of hybrid cloud; and all of it, is fully aligned with Cisco UCS X-series and CDIP, such as solution automation with orchestrator, observability, and monitoring.

In CDIP, Cisco UCS X-series offers excellent platform for container cloud as compute engine for modern apps in the hybrid world. In the coming years, velocity of apps modernization will be tremendous, Cisco UCS X-series is fully aligned with and there will be wave of new technologies coming over such as new compute modules, networking fabric, PCIe fabric, pooled NVMe drives, persistent memory, GPU accelerators, custom ASICs, and so on.

Cisco Data Intelligence Platform powered by Cisco UCS and Cloudera Data Platform enables enterprise-graced analytics and management platform with following key benefits:

- Future proof architecture supporting fast data ingest and management to cater to the variety of analytics workload from edge to AI.
- Ability to auto-scale or cloud burst and suspend according to workload demand.
- Consistent user experience on hybrid cloud and multi-cloud environments.
- Self-service access to integrated, multi-function analytics on centrally managed data eliminating data silos.

About the Author

Hardik Patel, Technical Marketing Engineer, Cloud and Compute Product Group, Cisco Systems, Inc.

Hardik Patel is a Technical Marketing Engineer in Cisco UCS Product Management and Datacenter Solutions Engineering. He is currently responsible for design and architect of Cisco Data Intelligence Platform based Big Data infrastructure solutions and performance. Hardik holds Master of Science degree in Computer Science with various career-oriented certification in virtualization, network, and Microsoft.

Acknowledgements

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

- Ali Bajwa, Cloudera
- Tarun Dave, Cloudera
- Matt Akins, Nvidia

Appendix

This appendix contains the following:

- [Appendix A – Bill of Materials](#)
- [Appendix B – References Used in this CVD](#)
- [Appendix C – Glossary of Terms](#)
- [Appendix D –Glossary of Acronyms](#)
- [Appendix E – Recommended for You](#)

Appendix A – Bill of Materials

Table 7. Bill of Material for Cisco UCS C240 M6SX – CDP PvC Base Cluster – Ozone Data Node

| Part Number | Description | Qty |
|-------------------|-------------------------------------------------------------|-----|
| UCS-M6-MLB | UCS M6 RACK, BLADE MLB | 1 |
| UCSC-C240-M6SX | UCS C240 M6 Rack w/o CPU, mem, drives, 2U w 24 | 8 |
| CON-OSP-UCSCXC24 | SNTC-24X7X40S UCS C240 M6 Rack | 8 |
| UCSC-M-V25-04 | Cisco UCS VIC 1467 quad port 10/25G SFP28 mLOM | 8 |
| UCSC-ADGPU-240M6 | C240M6 GPU Air Duct 2USFF/NVMe (for DW/FL only) | 8 |
| CIMC-LATEST | IMC SW (Recommended) latest release for C-Series Servers. | 8 |
| UCS-M2-960GB | 960GB SATA M.2 | 16 |
| UCS-M2-HWRAID | Cisco Boot optimized M.2 Raid controller | 8 |
| UCSX-TPM-002C | TPM 2.0, TCG, FIPS140-2, CC EAL4+ Certified, for M6 servers | 8 |
| UCSC-RAIL-M6 | Ball Bearing Rail Kit for C220 & C240 M6 rack servers | 8 |
| UCS-DIMM-BLK | UCS DIMM Blanks | 128 |
| UCSC-RIS2A-240M6 | C240 / C245 M6 Riser2A; (x8;x16;x8);StBkt; (CPU2) | 8 |
| UCSC-HSLP-M6 | Heatsink for 1U/2U LFF/SFF GPU SKU | 16 |
| UCS-SCAP-M6 | M6 SuperCap | 8 |
| UCSC-M2EXT-240M6 | C240M6 / C245M6 2U M.2 Extender board | 8 |
| CBL-RSASR3B-240M6 | C240M6 2U x2 Rear SAS/SATA cable; | 8 |

| Part Number | Description | Qty |
|-------------------|--------------------------------------------------------------|-----|
| | (Riser3B) | |
| CBL-SDSAS-240M6 | CBL C240M6X (2U24) MB CPU1(NVMe-A) to PISMO BEACH PLUS | 8 |
| UCS-P100CBL-240M5 | C240/C245 M5/M6 NVIDIA P100 /V100 /RTX /A100 /A40 /A16 Cable | 16 |
| CBL-SCAPSD-C240M6 | CBL Super Cap for PB+ C240 / C245 M6 | 8 |
| UCS-CPU-I6338 | Intel 6338 2.0GHz/205W 32C/48MB DDR4 3200MHz | 16 |
| UCS-MR-X32G2RW | 32GB RDIMM DRx4 3200 (8Gb) | 128 |
| UCSC-RAID-M6SD | Cisco M6 12G SAS RAID Controller with 4GB FBWC (28 Drives) | 8 |
| UCS-SD38T6I1X-EV | 3.8TB 2.5 inch Enterprise Value 6G SATA SSD | 192 |
| UCSC-RIS1A-240M6 | C240 M6 Riser1A; (x8;x16x, x8); StBkt; (CPU1) | 8 |
| UCSC-RIS3B-240M6 | C240 M6 Riser 3B; 2xHDD; StBkt; (CPU2) | 8 |
| UCSC-GPU-A100-80 | TESLA A100, PASSIVE, 300W, 80GB | 8 |
| NV-GRID-OPT-OUT | NVIDIA GRID SW OPT-OUT | 8 |
| UCSC-GPU-A100-80 | TESLA A100, PASSIVE, 300W, 80GB | 8 |
| NV-GRID-OPT-OUT | NVIDIA GRID SW OPT-OUT | 8 |
| UCSC-PSU1-1600W | Cisco UCS 1600W AC Power Supply for Rack Server | 16 |
| CAB-C13-C14-2M | Power Cord Jumper, C13-C14 Connectors, 2 Meter Length | 16 |
| UCS-SID-INFR-BD | Big Data and Analytics Platform (Hadoop/IoT/ITOA/AI/ML) | 8 |
| UCS-SID-WKL-BD | Big Data and Analytics (Hadoop/IoT/ITOA) | 8 |
| UCS-NVMEI4-I3200 | 3.2TB 2.5in U.2 Intel P5600 NVMe High Perf Medium Endurance | 16 |
| DC-MGT-OPTOUT | Intersight Opt Out | 1 |
| OPTOUT-OWN-EA | License not needed: Customer already owns Licenses in an EA | 1 |
| UCS-FI-64108-U | UCS Fabric Interconnect 64108 | 2 |
| CON-OSP-FI64108U | SNTC-24X7X4OS-UCS Fabric Interconnect | 2 |

| Part Number | Description | Qty |
|-------------------|--------------------------------------------------------------|-----|
| | 64108 | |
| N10-MGT018 | UCS Manager v4.2 and Intersight Managed Mode v4.2 | 2 |
| UCS-PSU-64108-AC | UCS 64108 Power Supply/100-240VAC | 4 |
| CAB-N5K6A-NA | Power Cord, 200/240V 6A North America | 4 |
| SFP-H25G-CU3M | 25GBASE-CU SFP28 Cable 3 Meter | 64 |
| QSFP-H40G-AOC3M | 40GBASE Active Optical Cable, 3m | 24 |
| UCS-ACC-64108 | UCS 64108 Chassis Accessory Kit | 2 |
| UCS-FAN-64108 | UCS 64108 Fan Module | 6 |
| RHEL-2S2V-3A= | Red Hat Enterprise Linux (1-2 CPU, 1-2 VN); 3-Yr Support Req | 1 |
| CON-ISV1-EL2S2V3A | ISV 24X7 RHEL Server 2Socket-OR-2Virtual; ANNUAL List Price | 1 |
| UCS-RHEL-TERMS | Acceptance of Terms, Standalone RHEL License for UCS Servers | 1 |

Appendix B - References Used in Guide

Cisco Infrastructure Solution for Data Analytics

<https://www.cisco.com/c/en/us/solutions/data-center-virtualization/big-data/index.html>

Design Zone for Cisco Data Intelligence Platform:

<https://www.cisco.com/c/en/us/solutions/design-zone/data-center-design-guides/data-center-big-data.html>

Cloudera Private Cloud Getting Started Guide:

<https://docs.cloudera.com/cdp-private-cloud/latest/index.html>

CDP Private Cloud Machine Learning Overview:

<https://docs.cloudera.com/machine-learning/1.3.4/index.html>

CDP Private Cloud Data Engineering Overview:

<https://docs.cloudera.com/data-engineering/1.3.4/index.html>

CDP Private Cloud Data Warehouse Overview:

<https://docs.cloudera.com/data-warehouse/1.3.4/index.html>

Appendix C - Glossary of Terms

This glossary addresses some terms used in this document, for the purposes of aiding understanding. This is not a complete list of all multicloud terminology. Some Cisco product links are supplied here also, where considered useful for the purposes of clarity, but this is by no means intended to be a complete list of all applicable Cisco products.

| | |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>aaS/XaaS (IT capability provided as a Service)</p> | <p>Some IT capability, X, provided as a service (XaaS). Some benefits are:</p> <ul style="list-style-type: none"> • The provider manages the design, implementation, deployment, upgrades, resiliency, scalability, and overall delivery of the service and the infrastructure that supports it. • There are very low barriers to entry, so that services can be quickly adopted and dropped in response to business demand, without the penalty of inefficiently utilized CapEx. • The service charge is an IT OpEx cost (pay-as-you-go), whereas the CapEx and the service infrastructure is the responsibility of the provider. • Costs are commensurate to usage and hence more easily controlled with respect to business demand and outcomes. <p>Such services are typically implemented as “microservices,” which are accessed via REST APIs. This architectural style supports composition of service components into systems. Access to and management of aaS assets is via a web GUI and/or APIs, such that Infrastructure-as-code (IaC) techniques can be used for automation, for example, Ansible and Terraform.</p> <p>The provider can be any entity capable of implementing an aaS “cloud-native” architecture. The cloud-native architecture concept is well-documented and supported by open-source software and a rich ecosystem of services such as training and consultancy. The provider can be an internal IT department or any of many third-party companies using and supporting the same open-source platforms.</p> <p>Service access control, integrated with corporate IAM, can be mapped to specific users and business activities, enabling consistent policy controls across services, wherever they are delivered from.</p> |
| <p>Ansible</p> | <p>An infrastructure automation tool, used to implement processes for instantiating and configuring IT service components, such as VMs on an IaaS platform. Supports the consistent execution of processes defined in YAML “playbooks” at scale, across multiple targets. Because the Ansible artefacts (playbooks) are text-based, they can be stored in a Source Code Management (SCM) system, such as GitHub. This allows for software development like processes to be applied to infrastructure automation, such as, Infrastructure-as-code (see IaC below).</p> <p>https://www.ansible.com</p> |
| <p>AWS (Amazon Web Services)</p> | <p>Provider of IaaS and PaaS.</p> <p>https://aws.amazon.com</p> |
| <p>Azure</p> | <p>Microsoft IaaS and PaaS.</p> <p>https://azure.microsoft.com/en-gb/</p> |
| <p>Co-located data center</p> | <p>“A colocation center (CoLo)...is a type of data center where equipment, space, and bandwidth are available for rental to retail customers. Colocation facilities provide space, power, cooling, and physical security for the server, storage, and networking equipment of other firms and also connect them to a variety of telecommunications and network service providers with a minimum of cost and complexity.”</p> <p>https://en.wikipedia.org/wiki/Colocation_centre</p> |

| | |
|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Containers (Docker) | <p>A (Docker) container is a means to create a package of code for an application and its dependencies, such that the application can run on different platforms which support the Docker environment. In the context of aaS, microservices are typically packaged within Linux containers orchestrated by Kubernetes (K8s).</p> <p>https://www.docker.com</p> <p>https://www.cisco.com/c/en/us/products/cloud-systems-management/containerplatform/index.html</p> |
| DevOps | <p>The underlying principle of DevOps is that the application development and operations teams should work closely together, ideally within the context of a toolchain that automates the stages of development, test, deployment, monitoring, and issue handling. DevOps is closely aligned with IaC, continuous integration and deployment (CI/CD), and Agile software development practices.</p> <p>https://en.wikipedia.org/wiki/DevOps</p> <p>https://en.wikipedia.org/wiki/CI/CD</p> |
| Edge compute | <p>Edge compute is the idea that it can be more efficient to process data at the edge of a network, close to the endpoints that originate that data, or to provide virtualized access services, such as at the network edge. This could be for reasons related to low latency response, reduction of the amount of unprocessed data being transported, efficiency of resource utilization, and so on. The generic label for this is Multi-access Edge Computing (MEC), or Mobile Edge Computing for mobile networks specifically.</p> <p>From an application experience perspective, it is important to be able to utilize, at the edge, the same operations model, processes, and tools used for any other compute node in the system.</p> <p>https://en.wikipedia.org/wiki/Mobile_edge_computing</p> |
| IaaS (Infrastructure as-a-Service) | <p>Infrastructure components provided aaS, located in data centers operated by a provider, typically accessed over the public Internet. IaaS provides a base platform for the deployment of workloads, typically with containers and Kubernetes (K8s).</p> |
| IaC (Infrastructure as-Code) | <p>Given the ability to automate aaS via APIs, the implementation of the automation is typically via Python code, Ansible playbooks, and similar. These automation artefacts are programming code that define how the services are consumed. As such, they can be subject to the same code management and software development regimes as any other body of code. This means that infrastructure automation can be subject to all of the quality and consistency benefits, CI/CD, traceability, automated testing, compliance checking, and so on, that could be applied to any coding project.</p> <p>https://en.wikipedia.org/wiki/Infrastructure_as_code</p> |
| IAM (Identity and Access Management) | <p>IAM is the means to control access to IT resources so that only those explicitly authorized to access given resources can do so. IAM is an essential foundation to a secure multicloud environment.</p> <p>https://en.wikipedia.org/wiki/Identity_management</p> |
| IBM (Cloud) | <p>IBM IaaS and PaaS.</p> <p>https://www.ibm.com/cloud</p> |
| Intersight | <p>Cisco Intersight™ is a Software-as-a-Service (SaaS) infrastructure lifecycle management platform that delivers simplified configuration, deployment, maintenance, and support.</p> <p>https://www.cisco.com/c/en/us/products/servers-unified-computing/intersight/index.html</p> |

| | |
|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GCP (Google Cloud Platform) | Google IaaS and PaaS. https://cloud.google.com/gcp |
| Kubernetes (K8s) | Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. https://kubernetes.io |
| Microservices | A microservices architecture is characterized by processes implementing fine-grained services, typically exposed via REST APIs and which can be composed into systems. The processes are often container-based, and the instantiation of the services often managed with Kubernetes. Microservices managed in this way are intrinsically well suited for deployment into IaaS environments, and as such, are the basis of a cloud native architecture. https://en.wikipedia.org/wiki/Microservices |
| PaaS (Platform-as-a-Service) | PaaS is a layer of value-add services, typically for application development, deployment, monitoring, and general lifecycle management. The use of IaC with IaaS and PaaS is very closely associated with DevOps practices. |
| Private on-premises data center | A data center infrastructure housed within an environment owned by a given enterprise is distinguished from other forms of data center, with the implication that the private data center is more secure, given that access is restricted to those authorized by the enterprise. Thus, circumstances can arise where very sensitive IT assets are only deployed in a private data center, in contrast to using public IaaS. For many intents and purposes, the underlying technology can be identical, allowing for hybrid deployments where some IT assets are privately deployed but also accessible to other assets in public IaaS. IAM, VPNs, firewalls, and similar are key technologies needed to underpin the security of such an arrangement. |
| REST API | Representational State Transfer (REST) APIs is a generic term for APIs accessed over HTTP(S), typically transporting data encoded in JSON or XML. REST APIs have the advantage that they support distributed systems, communicating over HTTP, which is a well-understood protocol from a security management perspective. REST APIs are another element of a cloud-native applications architecture, alongside microservices. https://en.wikipedia.org/wiki/Representational_state_transfer |
| SaaS (Software-as-a-Service) | End-user applications provided “aaS” over the public Internet, with the underlying software systems and infrastructure owned and managed by the provider. |
| SAML (Security Assertion Markup Language) | Used in the context of Single-Sign-On (SSO) for exchanging authentication and authorization data between an identity provider, typically an IAM system, and a service provider (some form of SaaS). The SAML protocol exchanges XML documents that contain security assertions used by the aaS for access control decisions. https://en.wikipedia.org/wiki/Security_Assertion_Markup_Language |
| Terraform | An open-source IaC software tool for cloud services, based on declarative configuration files. https://www.terraform.io |

Appendix D -Glossary of Acronyms

AAA—Authentication, Authorization, and Accounting

ACP—Access-Control Policy

ACI—Cisco Application Centric Infrastructure

ACK—Acknowledge or Acknowledgement
ACL—Access-Control List
AD—Microsoft Active Directory
AFI—Address Family Identifier
AMP—Cisco Advanced Malware Protection
AP—Access Point
API—Application Programming Interface
APIC— Cisco Application Policy Infrastructure Controller (ACI)
ASA—Cisco Adaptative Security Appliance
ASM—Any-Source Multicast (PIM)
ASR—Aggregation Services Router
Auto-RP—Cisco Automatic Rendezvous Point protocol (multicast)
AVC—Application Visibility and Control
BFD—Bidirectional Forwarding Detection
BGP—Border Gateway Protocol
BMS—Building Management System
BSR—Bootstrap Router (multicast)
BYOD—Bring Your Own Device
CAPWAP—Control and Provisioning of Wireless Access Points Protocol
CDIP - Cisco Data Intelligence Platform
CDP - Cloudera Data Platform
CDP PvC - Cloudera Data Platform Private Cloud
CDP PvC DS - Cloudera Data Platform Private Cloud Data Services
CDW - Cloudera Data Warehouse
CML - Cloudera Machine Learning
CDE - Cloudera Data Engineering
CEF—Cisco Express Forwarding
CMD—Cisco Meta Data
CPU—Central Processing Unit
CSR—Cloud Services Routers
CTA—Cognitive Threat Analytics
CUWN—Cisco Unified Wireless Network

CVD—Cisco Validated Design

CYOD—Choose Your Own Device

DC—Data Center

DHCP—Dynamic Host Configuration Protocol

DM—Dense-Mode (multicast)

DMVPN—Dynamic Multipoint Virtual Private Network

DMZ—Demilitarized Zone (firewall/networking construct)

DNA—Cisco Digital Network Architecture

DNS—Domain Name System

DORA—Discover, Offer, Request, ACK (DHCP Process)

DWDM—Dense Wavelength Division Multiplexing

ECMP—Equal Cost Multi Path

EID—Endpoint Identifier

EIGRP—Enhanced Interior Gateway Routing Protocol

EMI—Electromagnetic Interference

ETR—Egress Tunnel Router (LISP)

EVPN—Ethernet Virtual Private Network (BGP EVPN with VXLAN data plane)

FHR—First-Hop Router (multicast)

FHRP—First-Hop Redundancy Protocol

FMC—Cisco Firepower Management Center

FTD—Cisco Firepower Threat Defense

GBAC—Group-Based Access Control

GbE—Gigabit Ethernet

Gbit/s—Gigabits Per Second (interface/port speed reference)

GRE—Generic Routing Encapsulation

GRT—Global Routing Table

HA—High-Availability

HQ—Headquarters

HSRP—Cisco Hot-Standby Routing Protocol

HTDB—Host-tracking Database (SD-Access control plane node construct)

IBNS—Identity-Based Networking Services (IBNS 2.0 is the current version)

ICMP—Internet Control Message Protocol

IDF—Intermediate Distribution Frame; essentially a wiring closet.

IEEE—Institute of Electrical and Electronics Engineers

IETF—Internet Engineering Task Force

IGP—Interior Gateway Protocol

IID—Instance-ID (LISP)

IOE—Internet of Everything

IoT—Internet of Things

IP—Internet Protocol

IPAM—IP Address Management

IPS—Intrusion Prevention System

IPSec—Internet Protocol Security

ISE—Cisco Identity Services Engine

ISR—Integrated Services Router

IS-IS—Intermediate System to Intermediate System routing protocol

ITR—Ingress Tunnel Router (LISP)

LACP—Link Aggregation Control Protocol

LAG—Link Aggregation Group

LAN—Local Area Network

L2 VNI—Layer 2 Virtual Network Identifier; as used in SD-Access Fabric, a VLAN.

L3 VNI—Layer 3 Virtual Network Identifier; as used in SD-Access Fabric, a VRF.

LHR—Last-Hop Router (multicast)

LISP—Location Identifier Separation Protocol

MAC—Media Access Control Address (OSI Layer 2 Address)

MAN—Metro Area Network

MEC—Multichassis EtherChannel, sometimes referenced as *MCEC*

MDF—Main Distribution Frame; essentially the central wiring point of the network.

MnT—Monitoring and Troubleshooting Node (Cisco ISE persona)

MOH—Music on Hold

MPLS—Multiprotocol Label Switching

MR—Map-resolver (LISP)

MS—Map-server (LISP)

MSDP—Multicast Source Discovery Protocol (multicast)

MTU—Maximum Transmission Unit

NAC—Network Access Control

NAD—Network Access Device

NAT—Network Address Translation

NBAR—Cisco Network-Based Application Recognition (NBAR2 is the current version).

NFV—Network Functions Virtualization

NSF—Non-Stop Forwarding

OSI—Open Systems Interconnection model

OSPF—Open Shortest Path First routing protocol

OT—Operational Technology

PAgP—Port Aggregation Protocol

PAN—Primary Administration Node (Cisco ISE persona)

PCI DSS—Payment Card Industry Data Security Standard

PD—Powered Devices (PoE)

PETR—Proxy-Egress Tunnel Router (LISP)

PIM—Protocol-Independent Multicast

PITR—Proxy-Ingress Tunnel Router (LISP)

PnP—Plug-n-Play

PoE—Power over Ethernet (Generic term, may also refer to IEEE 802.3af, 15.4W at PSE)

PoE+—Power over Ethernet Plus (IEEE 802.3at, 30W at PSE)

PSE—Power Sourcing Equipment (PoE)

PSN—Policy Service Node (Cisco ISE persona)

pxGrid—Platform Exchange Grid (Cisco ISE persona and publisher/subscriber service)

PxTR—Proxy-Tunnel Router (LISP - device operating as both a PETR and PITR)

QoS—Quality of Service

RADIUS—Remote Authentication Dial-In User Service

REST—Representational State Transfer

RFC—Request for Comments Document (IETF)

RIB—Routing Information Base

RHEL - Red Hat Enterprise Linux

RHOCP - Red Hat OpenShift Container Platform

RLOC—Routing Locator (LISP)

RP–Rendezvous Point (multicast)
RP–Redundancy Port (WLC)
RP–Route Processer
RPF–Reverse Path Forwarding
RR–Route Reflector (BGP)
RTT–Round-Trip Time
SA–Source Active (multicast)
SAFI–Subsequent Address Family Identifiers (BGP)
SD–Software-Defined
SDA–Cisco Software Defined-Access
SDN–Software-Defined Networking
SFP–Small Form-Factor Pluggable (1 GbE transceiver)
SFP+– Small Form-Factor Pluggable (10 GbE transceiver)
SGACL–Security-Group ACL
SGT–Scalable Group Tag, sometimes reference as Security Group Tag
SM–Spare-mode (multicast)
SNMP–Simple Network Management Protocol
SSID–Service Set Identifier (wireless)
SSM–Source-Specific Multicast (PIM)
SSO–Stateful Switchover
STP–Spanning-tree protocol
SVI–Switched Virtual Interface
SVL–Cisco StackWise Virtual
SWIM–Software Image Management
SXP–Scalable Group Tag Exchange Protocol
Syslog–System Logging Protocol
TACACS+–Terminal Access Controller Access-Control System Plus
TCP–Transmission Control Protocol (OSI Layer 4)
UCS– Cisco Unified Computing System
UDP–User Datagram Protocol (OSI Layer 4)
UPoE–Cisco Universal Power Over Ethernet (60W at PSE)
UPoE+– Cisco Universal Power Over Ethernet Plus (90W at PSE)

URL—Uniform Resource Locator

VLAN—Virtual Local Area Network

VM—Virtual Machine

VN—Virtual Network, analogous to a VRF in SD-Access

VNI—Virtual Network Identifier (VXLAN)

vPC—virtual Port Channel (Cisco Nexus)

VPLS—Virtual Private LAN Service

VPN—Virtual Private Network

VPNv4—BGP address family that consists of a Route-Distinguisher (RD) prepended to an IPv4 prefix

VPWS—Virtual Private Wire Service

VRF—Virtual Routing and Forwarding

VSL—Virtual Switch Link (Cisco VSS component)

VSS—Cisco Virtual Switching System

VXLAN—Virtual Extensible LAN

WAN—Wide-Area Network

WLAN—Wireless Local Area Network (generally synonymous with IEEE 802.11-based networks)

WoL—Wake-on-LAN

xTR—Tunnel Router (LISP - device operating as both an ETR and ITR)

Appendix E - Recommended for You

To find out more about Cisco UCS Big Data solutions, go to: <https://www.cisco.com/go/bigdata>

To find out more about Cisco UCS Big Data validated designs, go to:
https://www.cisco.com/go/bigdata_design

To find out more about Cisco Data Intelligence Platform, go to:
<https://www.cisco.com/c/dam/en/us/products/servers-unified-computing/ucs-c-series-rack-servers/solution-overview-c22-742432.pdf>

To find out more about Cisco UCS AI/ML solutions, go to: <http://www.cisco.com/go/ai-compute>

To find out more about Cisco ACI solutions, go to: <http://www.cisco.com/go/aci>

To find out more about Cisco validated solutions based on Software Defined Storage, go to:
<https://www.cisco.com/c/en/us/solutions/data-center-virtualization/software-defined-storage-solutions/index.html>

Cloudera Data Platform Private Cloud latest release note, go to: <https://docs.cloudera.com/cdp-private-cloud-upgrade/latest/release-guide/topics/cdpdc-release-notes-links.html>

Cloudera Data Platform Private Cloud Base Requirements and Supported Versions, go to:
<https://docs.cloudera.com/cdp-private-cloud-upgrade/latest/release-guide/topics/cdpdc-requirements-supported-versions.html>

Cloudera Data Platform Private Cloud Data Services installation on Red Hat OpenShift Container Platform requirements and supported versions, go to: <https://docs.cloudera.com/cdp-private-cloud-data-services/1.3.4/installation/topics/cdppvc-installation-overview.html>

Cloudera Data Platform Private Cloud Data Services installation on Embedded Container Service requirements and supported versions, go to: <https://docs.cloudera.com/cdp-private-cloud-data-services/1.3.4/installation-ecs/topics/cdppvc-installation-ecs-overview.html>

Feedback

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

CVD Program

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

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

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified 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. (LDW_P3)

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)

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

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

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