Cloudera Manager 7.13.2

Installation

Date published: 2020-11-30
Date modified: 2026-02-27

CLOUDERA

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2026. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Cloudera Base on premises Installation Guide 6

Version and Download Information 6
Cloudera Runtime Version Information 6
Cloudera Runtime Download Information 7
Cloudera Manager Support Matrix 8
Cloudera Base on premises Trial Download Information 11

System Requirements 12
Hardware Requirements 12
Cloudera Runtime 12
Operating System Requirements 24
Database Requirements 28
RDBMS High Availability Support 29
Java Requirements 31
Networking and Security Requirements 33
Data at Rest Encryption Requirements 38
Third-party filesystems 39

Trial Installation 39
Installing a Trial Cluster 39
Before You Begin a Trial Installation 40
Download the Trial version of Cloudera Base on premises 40
Run the Cloudera Manager Server Installer 41
Install Cloudera Runtime 45
Set Up a Cluster Using the Wizard 58
Stopping the Embedded PostgreSQL Database 60
Starting the Embedded PostgreSQL Database 61
Changing Embedded PostgreSQL Database Passwords 61
Migrating from the Cloudera Manager Embedded PostgreSQL Database Server to an External PostgreSQL Database 63
Prerequisites 63
Identify Roles that Use the Embedded Database Server 63
Migrate Databases from the Embedded Database Server to the External PostgreSQL Database Server 65

FIPS compliance and integration guide 69
FIPS compliance overview for Cloudera Base on premises 69
Using FIPS with Cloudera products 69
Key differences between FIPS 140-2 and FIPS 140-3 70
FIPS requirements and constraints 70
Support matrix for FIPS 70
FIPS deployments: Constraints and operational guardrails 72
Upgrading from JDK 8 to JDK 17 on FIPS 74
Preparing for FIPS integration 76

FIPS integration for JDK 17....76
Transition from FIPS 140-2 to FIPS 140-3....84
Support matrix for data migration....84
Migrating from FIPS 140-2 to FIPS 140-3....84

Install and configure additional required components....86

Production Installation....89
Before You Install....90
Install and Configure Databases....90
Configure Network Names....145
Setting SELinux Mode....146
Disabling the Firewall....147
Enable an NTP Service....147
Impala Requirements....148
Cloudera Runtime Cluster Hosts and Role Assignments....150
Configuring Local Package and Parcel Repositories....157
Installing Cloudera Manager....158
Installing Cloudera Runtime....159
Set Up a Cluster Using the Wizard....164
Select Services....164
Assign Roles....164
Setup database....165
Enter Required Parameters....173
Review Changes....173
Configure Kerberos....173
Command Details....174
Summary....174
(optional) Enable high availability for Cloudera Manager....174
(Recommended) Enable Auto-TLS....174
(Recommended) Enable Kerberos....174
Additional Steps for Apache Ranger....175
Enable Plugins....175
Add Solr WebUI Users....175
Update the Time-to-live configuration for Ranger Audits....175
Determine susceptibility of Ranger database upgrade failure....176
Installing Apache Knox....177
Apache Knox install role parameters....179
Setting Up Data at Rest Encryption for HDFS....181
Installing Ranger KMS cllocated with data cluster....181
Installing Ranger KMS in a federated deployment....187
Installing Cloudera Navigator Encrypt....189
Setting Up an Internal Repository....189
Installing Navigator Encrypt (RHEL-Compatible)....189
Installing Navigator Encrypt (SLES-12)....191
Installing Navigator Encrypt (SLES-15)....191
Installing Navigator Encrypt (Ubuntu)....193
Installing for Ranger KMS....194
Setting Up TLS for Navigator Encrypt Clients....194
Entropy Requirements....195
Uninstalling and Reinstalling Navigator Encrypt....196
Installing Ranger RMS....196

Installation Reference.....204
Ports.....204
Ports Used by Cloudera Manager.....204
Ports Used by Cloudera Runtime Components.....209
Ports Used by DistCp.....217
Ports Used by Third-Party Components.....218
Service Dependencies in Cloudera Manager.....219
Cloudera Manager sudo command options.....221
Introduction to Parcels.....222

After You Install.....222
Deploying Clients.....223
Initializing Solr and creating HDFS home directory.....223
Testing the Installation.....223
Checking Host Heartbeats.....224
Run a MapReduce job.....224
Running a MapReduce job using CLI.....224
Running a MapReduce job using the UI.....224
Testing with Hue.....226
Deploying Atlas service.....226
Secure Your Cluster.....228
Installing the GPL Extras Parcel.....229
Configuring HDFS properties to optimize log collection.....229
Migrating from H2 to PostgreSQL database in YARN Queue Manager.....230

Troubleshooting Installation Problems.....231

Uninstalling Cloudera Manager and Managed Software.....235
Record User Data Paths.....235
Stop all Services.....235
Deactivate and Remove Parcels.....235
Delete the Cluster.....236
Uninstall the Cloudera Manager Server.....236
Uninstall Cloudera Manager Agent and Managed Software.....237
Remove Cloudera Manager, User Data, and Databases.....237
Uninstalling a Cloudera Runtime Component From a Single Host.....238

Cloudera Base on premises Installation Guide

Use this Installation Guide to learn how to install Cloudera software, including Cloudera Manager, Cloudera Runtime, and other managed services, in a production or trial environment.

Related Information

Version and Download Information
System Requirements
Trial Installation
Production Installation
Custom Installation Solutions
Installation Reference
After You Install
Troubleshooting Installation Problems
Uninstalling Cloudera Manager and Managed Software

Version and Download Information

The following topics describe the available versions and download locations for Cloudera Manager and Cloudera Runtime.

Related Information

Cloudera Base on premises Installation Guide

Cloudera Runtime Version Information

Version numbers for current and previous releases of Cloudera Runtime 7.x.

Cloudera Runtime 7.3.2.0 is based on Apache Hadoop 3. For more information, see Cloudera Runtime Component Versions.

Release date: March 31, 2026

Previous releases:

- Cloudera Runtime 7.3.1.500 SP3 Release Date: November 5, 2025
- Cloudera Runtime 7.3.1.400 SP2 Release Date: June 27, 2025
- Cloudera Runtime 7.3.1.200 SP1 Release Date: April 24, 2025
- Cloudera Runtime 7.3.1 Release Date: December 10, 2024
- Cloudera Runtime 7.1.9 SP1 Release Date: July 19, 2024
- Cloudera Runtime 7.1.7 SP3 Release Date: May 07, 2024
- Cloudera Runtime 7.1.9 Release Date: September 08, 2023
- Cloudera Runtime 7.1.7 SP2 Release Date: January 31, 2023
- Cloudera Runtime 7.1.8 Release Date: August 30, 2022
- Cloudera Runtime 7.1.7 SP1 Release Date: March 30, 2022
- Cloudera Runtime 7.1.7 Release Date: August 5, 2021
- Cloudera Runtime 7.1.6 Release Date: March 3, 2021
- Cloudera Runtime 7.1.5 Release Date: November 30, 2020
- Cloudera Runtime 7.1.4 Release Date: October 13, 2020
- Cloudera Runtime 7.1.3 Release Date: August 10, 2020
- Cloudera Runtime 7.1.2 Release Date: July 13, 2020

- Cloudera Runtime 7.1.1 Release Date: May 22, 2020
- Cloudera Runtime 7.0.3 Release Date: November 22, 2019

Cloudera Runtime Download Information

Important: Access to Cloudera Runtime parcels for production purposes requires authentication. To access the parcels at the locations below, you must first have an active subscription agreement and obtain a license key file along with the required authentication credentials (username and password).

The license key file and authentication credentials are provided in an email sent to customer accounts from Cloudera when a new license is issued. If you have an existing license with a Cloudera Base on premises Edition entitlement, you might not have received an email. In this instance you can identify the authentication credentials from the license key file. If you do not have access to the license key, contact your account representative to receive a copy.

To identify your authentication credentials using your license key file, complete the following steps:

- From Cloudera.com, log into the Cloudera.com account associated with the Cloudera Base on premises license and subscription agreement.
- On the Cloudera Base on premises Download page, click Download Now and scroll down to the Credential Generator.
- In the Generate Credentials text box, copy and paste the text of the “PGP Signed Message” within your license key file and click Get Credentials. The credentials generator returns your username and password.



Important: Make a note of the authentication credentials. You might need them during installation to complete tasks such as configuring a remote parcel repository.



Note:

- For information on repositories for the Cloudera Runtime 7.3.1 service packs and cumulative hotfixes, see Release Matrix.
- For information on repositories for the Cloudera Runtime 7.1.9 SP1 cumulative hotfixes, see Cumulative hotfixes.
- For information on repositories for the Cloudera Runtime 7.1.9 cumulative hotfixes, see Cumulative hotfixes.
- For information on repositories for the Cloudera Runtime 7.1.7 SP3 cumulative hotfixes, see Cumulative hotfixes.

When you obtain your authentication credentials, use them to form the URL where you can access the Cloudera Runtime repository in the Cloudera Archive. Cloudera Manager can also download the Cloudera Runtime parcels directly during the installation process.

The repositories for Cloudera Runtime 7.x are listed in the following tables:

Table 1: Cloudera Runtime 7.3.2.0

Parcel Repository Location
`https://[USERNAME]:[PASSWORD]@archive.cloudera.com/p/cdh7/7.3.2.0/parcels`

Table 2: Cloudera Runtime 7.3.1.0

Parcel Repository Location
`https://[USERNAME]:[PASSWORD]@archive.cloudera.com/p/cdh7/7.3.1.0/parcels`

Table 3: Cloudera Runtime 7.1.9.1000 (Service Pack 1):

Parcel Repository Location
`https://[USERNAME]:[PASSWORD]@archive.cloudera.com/p/cdh7/7.1.9.1000/parcels`

Table 4: Cloudera Runtime 7.1.9.0:

Parcel Repository Location
`https://[USERNAME]:[PASSWORD]@archive.cloudera.com/p/cdh7/7.1.9.0/parcels`

Cloudera Manager Support Matrix

This support matrix lists the Cloudera Manager version support for Cloudera Base on premises and the Cloudera Data Services on premises.

Note: Not all combinations of , , and are supported. Ensure that the version of you are using supports the version of and you have selected. For more information, see the Support Matrix.

The versions of , , and CDH clusters that can be managed by are limited to the following:

For CDP Private Cloud Base

Table 5: Cloudera Manager support for Cloudera Base on premises

Cloudera Manager Version	Supported CDH/Cloudera Runtime versions
Cloudera Manager 7.13.3	• Cloudera Runtime 7.3.3.0 • Cloudera Runtime 7.3.2 SP2 (7.3.2.20000) • Cloudera Runtime 7.3.2 SP1 (7.3.2.10000) • Cloudera Runtime 7.3.2.0 • Cloudera Runtime 7.3.1.600 and higher • Cloudera Runtime 7.1.9 SP2, including Cumulative Hotfix 2 and subsequent cumulative hotfixes (CHFx) • Cloudera Runtime 7.1.7 SP3 CHF10
Cloudera Manager 7.13.2	• Cloudera Runtime 7.3.2 • Cloudera Runtime 7.3.1.500 SP3 • Cloudera Runtime 7.3.1.400 SP2 • Cloudera Runtime 7.3.1.200 SP1 • Cloudera Runtime 7.3.1 • Cloudera Runtime 7.1.9 SP1
Cloudera Manager 7.13.1 Latest cumulative hotfix	• Cloudera Runtime 7.3.1.500 SP3 • Cloudera Runtime 7.3.1.400 SP2 • Cloudera Runtime 7.3.1.200 SP1 • Cloudera Runtime 7.3.1 • Cloudera Runtime 7.1.9 SP1 • Cloudera Runtime 7.1.9 • Cloudera Runtime 7.1.8 • Cloudera Runtime 7.1.7 SP3 • Cloudera Runtime 7.1.7 SP2 • Cloudera Runtime 7.1.7 SP1 • Cloudera Runtime 7.1.7

Cloudera Manager Version	Supported CDH/Cloudera Runtime versions
Cloudera Manager 7.13.1	• Cloudera Runtime 7.3.1 • Cloudera Runtime 7.1.9 SP1 • Cloudera Runtime 7.1.9 • Cloudera Runtime 7.1.8 • Cloudera Runtime 7.1.7 SP3 • Cloudera Runtime 7.1.7 SP2 • Cloudera Runtime 7.1.7 SP1 • Cloudera Runtime 7.1.7 • CDH 6.3 • CDH 6.2 • CDH 6.1 • CDH 6.0
Cloudera Manager 7.11.3 Latest cumulative hotfix	• Cloudera Runtime 7.1.9 SP1 • Cloudera Runtime 7.1.7 SP3 • Cloudera Runtime 7.1.9 • Cloudera Runtime 7.1.7 SP2 • Cloudera Runtime 7.1.8 • Cloudera Runtime 7.1.7 SP1 • Cloudera Runtime 7.1.7 • Cloudera Runtime 7.1.6 • Cloudera Runtime 7.1.5 • Cloudera Runtime 7.1.4 • Cloudera Runtime 7.1.3 • Cloudera Runtime 7.1.2 • Cloudera Runtime 7.1.1 • Cloudera Runtime 7.0.3 • CDH 6.3 • CDH 6.2 • CDH 6.1 • CDH 6.0
Cloudera Manager 7.11.3  **Note:** You must install Python 3.8 (or 3.9 for RHEL 9.1) on all hosts before installing or upgrading to Cloudera Manager 7.11.3. For more information, see the Installing Python 3.	• Cloudera Runtime 7.1.9 • Cloudera Runtime 7.1.7 SP2 • Cloudera Runtime 7.1.8 • Cloudera Runtime 7.1.7 SP1 • Cloudera Runtime 7.1.7 • Cloudera Runtime 7.1.6 • Cloudera Runtime 7.1.5 • Cloudera Runtime 7.1.4 • Cloudera Runtime 7.1.3 • Cloudera Runtime 7.1.2 • Cloudera Runtime 7.1.1 • Cloudera Runtime 7.0.3 • CDH 6.3 • CDH 6.2 • CDH 6.1 • CDH 6.0
Cloudera Manager 7.7.3 should only be used when you need to use Python 3.8 for the Cloudera Manager agents. You must install Python 3.8 on all hosts before installing or upgrading to Cloudera Manager 7.7.3. Cloudera Manager 7.7.3-CHF2 supports only RHEL 8.4, RHEL 8.6, and RHEL 7.9. See the CDP Private Cloud Base Installation Guide for more information.	• Cloudera Runtime 7.1.8

Cloudera Manager Version	Supported CDH/Cloudera Runtime versions
Cloudera Manager 7.7.1  **Note:** Cloudera recommends you to use latest cumulative hotfix of Cloudera Manager 7.7.1 with Cloudera Runtime 7.1.7-SP2.	• Cloudera Runtime 7.1.7 SP2 • Cloudera Runtime 7.1.8 • Cloudera Runtime 7.1.7 SP1 • Cloudera Runtime 7.1.7 • Cloudera Runtime 7.1.6 • Cloudera Runtime 7.1.5 • Cloudera Runtime 7.1.4 • Cloudera Runtime 7.1.3 • Cloudera Runtime 7.1.2 • Cloudera Runtime 7.1.1 • Cloudera Runtime 7.0.3 • CDH 6.3 • CDH 6.2 • CDH 6.1 • CDH 6.0
7.6.7  **Important:** Do not upgrade to Cloudera Manager 7.6.7 if you are running Cloudera Data Services on premises in your deployment.	• Cloudera Runtime 7.1.7 SP2 • Cloudera Runtime 7.1.7 SP1 • Cloudera Runtime 7.1.7 • Cloudera Runtime 7.1.6 • Cloudera Runtime 7.1.5 • Cloudera Runtime 7.1.4 • Cloudera Runtime 7.1.3 • Cloudera Runtime 7.1.2 • Cloudera Runtime 7.1.1 • Cloudera Runtime 7.0.3 • CDH 6.3 • CDH 6.2 • CDH 6.1 • CDH 6.0 • CDH 5.16.2
7.6.1  **Important:** Do not upgrade to Cloudera Manager 7.6.1 if you are running Cloudera Data Services on premises in your deployment.	• Cloudera Runtime 7.1.7 SP1 • Cloudera Runtime 7.1.7 • Cloudera Runtime 7.1.6 • Cloudera Runtime 7.1.5 • Cloudera Runtime 7.1.4 • Cloudera Runtime 7.1.3 • Cloudera Runtime 7.1.2 • Cloudera Runtime 7.1.1 • Cloudera Runtime 7.0.3 • CDH 6.3 • CDH 6.2 • CDH 6.1 • CDH 6.0 • CDH 5.13 - 5.16

For CDP Private Cloud Data Services

Table 6: Cloudera Manager support for Cloudera Data Services on premises

Supported Cloudera Manager versions with Cloudera Data Services on premises	Supported Cloudera Runtime versions with Cloudera Data Services on premises
7.13.2.0	Cloudera Data Services on premises 1.5.5 SP2 supported only with: • Cloudera Runtime 7.3.2.0 • Cloudera Runtime 7.3.1.706 SP3 CHF2 • Cloudera Runtime 7.1.9 SP1 CHF14 or higher

Supported Cloudera Manager versions with Cloudera Data Services on premises	Supported Cloudera Runtime versions with Cloudera Data Services on premises
7.13.1 Cumulative hotfix 7 (7.13.1.706)	Cloudera Data Services on premises 1.5.5 SP2 supported only with: • Cloudera Runtime 7.3.1.706 SP3 CHF2 • Cloudera Runtime 7.1.9 SP1 CHF14 or higher
7.13.1 Cumulative hotfix 5 (7.13.1.501)	Cloudera Data Services on premises 1.5.5 SP1 supported only with: • Cloudera Runtime 7.3.1.500 SP3 • Cloudera Runtime 7.1.9 SP1 CHF11 or higher • Cloudera Runtime 7.1.7 SP3 CHF10 or higher
7.13.1 Cumulative hotfix 4 (7.13.1.402)	Cloudera Data Services on premises 1.5.5 CHF1 supported only with: • Cloudera Runtime 7.3.1.400 SP2 • Cloudera Runtime 7.1.9 SP1 CHF7 or higher • Cloudera Runtime 7.1.7 SP3 CHF10 or higher
7.13.1 Cumulative hotfix 3 (7.13.1.303)	Cloudera Data Services on premises 1.5.5 supported only with: • Cloudera Runtime 7.1.9 SP1 CHF7 or higher • Cloudera Runtime 7.1.7 SP3 CHF10 or higher
7.13.1 Cumulative hotfix 1 (7.13.1.100)	Cloudera Data Services on premises 1.5.4 SP2 supported only with: • Cloudera Runtime 7.1.9 SP1 CHF4 or higher • Cloudera Runtime 7.1.9 CHF7 or higher • Cloudera Runtime 7.1.7 SP3
7.11.3 cumulative hotfix 11	Cloudera Data Services on premises 1.5.4 SP1 supported only with: • Cloudera Runtime 7.1.9 SP1 CHF4 or higher • Cloudera Runtime 7.1.9 CHF7 or higher • Cloudera Runtime 7.1.7 SP3
7.11.3 cumulative hotfix 9.1 (version: 7.11.3.24)	Cloudera Private Cloud Data Services 1.5.4 CHF3 supported only with: • Cloudera Runtime 7.1.9 SP1 • Cloudera Runtime 7.1.9 CHF6 or higher • Cloudera Runtime 7.1.7 SP3
7.11.3 cumulative hotfix 7 Data Services (version: 7.11.3.14)	Cloudera Private Cloud Data Services 1.5.4 CHF1 supported only with: • Cloudera Runtime 7.1.9 SP1 • Cloudera Runtime 7.1.9 CHF5 or higher • Cloudera Runtime 7.1.7 SP3
7.11.3 cumulative hotfix 6	Cloudera Private Cloud Data Services 1.5.4 supported only with: • Cloudera Runtime 7.1.9 CHF6 • Cloudera Runtime 7.1.8 CHF22 or higher • Cloudera Runtime 7.1.7 SP3

Cloudera Base on premises Trial Download Information

You can try the Cloudera Base on premises Edition of Cloudera Data Platform for 60 days without obtaining a license key file.

To download Cloudera Base on premises without obtaining a license key file, visit the Cloudera Base on premises Trial Download page, click Try Now, and follow the download instructions. When you install Cloudera Base on premises without a license key, you are performing a trial installation that includes an embedded PostgreSQL database and is not suitable for a production environment. For more information on trial installations, see the trial installation documentation.

A 60-day trial of Cloudera Base on premises Edition can be enabled permanently with the appropriate license. To obtain a Cloudera Base on premises Edition license, fill in the Contact Us form or call 866-843-7207

Related Information

Trial Installation

System Requirements

Refer to the following topics for specific information about hardware, operating system, and database requirements to install and use the Cloudera Manager and Cloudera Runtime components. Review the requirements before installing or upgrading the Cloudera Manager and Cloudera Runtime Cluster.

Related Information

Cloudera Base on premises Installation Guide

Hardware Requirements

This topic specifies the hardware requirements for Cloudera Base on premises.

As you create the architecture of your cluster, you will need to allocate Cloudera Manager and Cloudera Runtime roles among the hosts in the cluster to maximize your use of resources. Cloudera provides some guidelines about how to assign roles to cluster hosts. See Recommended Cluster Hosts and Role Distribution. When multiple roles are assigned to hosts, add together the total resource requirements (memory, CPUs, disk) for each role on a host to determine the required hardware.



Attention: All recommendations for the number of cores refer to logical cores, not physical cores.

For more information about sizing for a particular component, see the following minimum requirements:

Cloudera Runtime

Hardware requirements for Cloudera Runtime components.

Atlas

You must verify the supported hardware components for Atlas.

Memory	CPU	Disk	Additional Dependencies
Small: 4 GB Large: 32 GB	Minimum: 4 Medium: 8 Large: 16	No special requirement because HBase is used for storage.	Solr Shards: 4 (property: atlas_solr_shards) The shards for Atlas collections within Solr is determined by this number.

HDFS heap sizing

You can provision an HDFS cluster for optimal performance based on the desired storage capacity.

Component	Memory	CPU	Disk
JournalNode	1 GB (default) Set this value using the Java Heap Size of JournalNode in Bytes HDFS configuration property.	1 core minimum	1 dedicated disk

Component	Memory	CPU	Disk
NameNode	• Minimum: 1 GB (for proof-of-concept deployments) • Add an additional 1 GB for each additional 1,000,000 blocks Snapshots and encryption can increase the required heap memory. See *Sizing NameNode Heap Memory*. Set this value using the Java Heap Size of NameNode in Bytes HDFS configuration property.	Minimum of 4 dedicated cores; more may be required for larger clusters	• Minimum of 2 dedicated disks for metadata • 1 dedicated disk for log files (This disk may be shared with the operating system.) • Maximum disks: 4
DataNode	Minimum: 4 GB Maximum: 8 GB Increase the memory for higher replica counts or a higher number of blocks per DataNode. When increasing the memory, Cloudera recommends an additional 1 GB of memory for every 1 million replicas above 4 million on the DataNodes. For example, 5 million replicas require 5 GB of memory. Set this value using the Java Heap Size of DataNode in Bytes HDFS configuration property.	Minimum: 4 cores. Add more cores for highly active clusters.	Minimum: 4 Maximum: 24 The maximum acceptable size will vary depending upon how large average block size is. The DN's scalability limits are mostly a function of the number of replicas per DN, not the overall number of bytes stored. That said, having ultra-dense DNs will affect recovery times in the event of machine or rack failure. Cloudera does not support exceeding 100 TB per data node. You could use 12 x 8 TB spindles or 24 x 4TB spindles. Cloudera does not support drives larger than 8 TB. Configure the disks in JBOD mode. Do not use RAID/LVM/ZFS.

Warning: Running Cloudera Runtime on storage platforms other than direct-attached physical disks can provide suboptimal performance. Cloudera Enterprise and the majority of the Hadoop platform are optimized to provide high performance by distributing work across a cluster that can utilize data locality and fast local I/O.

HBase

Know the hardware resource requirements for HBase server in Cloudera Base on premises.

Component	Java Heap	CPU	Disk
Master	• 100-10,000 regions: 4 GB • 10,000 or more regions with 200 or more Region Servers: 8 GB • 10,000 or more regions with 300 or more Region Servers: 12 GB Set this value using the Java Heap Size of HBase Master in Bytes HBase configuration property.	Minimum 4 dedicated cores. You can add more cores for larger clusters, when using replication, or for bulk loads.	1 disk for local logs, which can be shared with the operating system and/or other Hadoop logs

Component	Java Heap	CPU	Disk
Region Server	• Minimum: 8 GB • Medium-scale production: 16 GB • Heap larger than 16 GB requires special Garbage Collection tuning. See *Configuring the HBase BlockCache*. Set this value using the Java Heap Size of HBase RegionServer in Bytes HBase configuration property.	Minimum: 4 dedicated cores	• 4 or more spindles for each HDFS DataNode • 1 disk for local logs (this disk can be shared with the operating system and/or other Hadoop logs
Thrift Server	1 GB - 4 GB Set this value using the Java Heap Size of HBase Thrift Server in Bytes HBase configuration property.	Minimum 2 dedicated cores.	1 disk for local logs, which can be shared with the operating system and other Hadoop logs.



Note: Consider adding more HBase Thrift Servers for production environments and deployments with a large number of Thrift client to scale horizontally.

Related Information

Configuring HBase BlockCache

Hive

Understand the Java Heap memory, CPU, and storage requirements needed to run HiveServer (HS2) and Hive Metastore (HMS).

Component	Java Heap		CPU	Disk
HiveServer 2	Single Connection	4 GB	Minimum 4 dedicated cores	Minimum 1 disk This disk is required for the following: • HiveServer2 log files • stdout and stderr output files • Configuration files • Operation logs stored in the operation_logs_dir directory, which is configurable • Any temporary files that might be created by local map tasks under the /tmp directory
	2-10 connections	4-6 GB		
	11-20 connections	6-12 GB		
	21-40 connections	12-16 GB		
	41 to 80 connections	16-24 GB		
	Cloudera recommends splitting HiveServer2 into multiple instances and load balancing them once you start allocating more than 16 GB to HiveServer2. The objective is to adjust the size to reduce the impact of Java garbage collection on active processing by the service.			
	Set this value using the Java Heap Size of HiveServer2 in Bytes Hive configuration property.			

Component	Java Heap		CPU	Disk
Hive Metastore	Single Connection	4 GB	Minimum 4 dedicated cores	Minimum 1 disk This disk is required so that the Hive metastore can store the following artifacts: • Logs • Configuration files • Backend database that is used to store metadata if the database server is also hosted on the same node
	2-10 connections	4-10 GB		
	11-20 connections	10-12 GB		
	21-40 connections	12-16 GB		
	41 to 80 connections	16-24 GB		
	Set this value using the Java Heap Size of Hive Metastore Server in Bytes Hive configuration property.			
Beeline CLI	Minimum: 2 GB		N/A	N/A

Hue

Review the memory, CPU, and storage requirements needed for running a Hue server in Cloudera Base on premises.

Component	Memory	CPU	Disk
Hue Server	• Minimum: 4 GB • Maximum 10 GB • If the cluster uses the Hue load balancer, add additional memory	Minimum: 1 Core to run Django When Hue is configured for high availability, add additional cores	Minimum: 10 GB for the database, which grows proportionally according to the cluster size and workloads. When Hue is configured for high availability, add space is required for the /tmp (temporary) directory, approximately 5GB.

The term "cluster size" refers to the number of nodes in the cluster. "Workload" in Hue means the number of queries run and the number of concurrent unique users using the application in a given period of time.

A minimum of 10GB is needed for the database. The Hive MetaStore service largely uses the database. The database grows in size quickly because of the query history that it retains. To optimize performance, you must regluarly cleanup old documents and queries.

Note: Hue is limited by cgroup settings. In Cloudera Manager, all memory soft/hard limits are set to -1.

Related Information

Adding a Load Balancer for Hue

Impala

Sizing requirements for Impala can vary significantly depending on the size and types of workloads using Impala.

Component	Native Memory	JVM Heap	CPU	Disk
Impala Daemon	Set this value using the Impala Daemon Memory Limit configuration property. • Minimum: 32 GB • Recommended: 128 GB	Set this value using the Java Heap Size of Impala Daemon in Bytes configuration property for the Coordinator Impala Daemons. • Minimum: 4 GB • Recommended: 8 GB	• Minimum: 4 • Recommended: 16 or more CPU instruction set: AVX2	• Minimum: 1 disk • Recommended: 8 or more

Component	Native Memory	JVM Heap	CPU	Disk
Catalog Server	Set this value using the Java Heap Size of Catalog Server in Bytes configuration property. • Minimum: 4 GB • Recommended: 8 GB		• Minimum: 4 • Recommended: 16 or more CPU instruction set: AVX2	• Minimum and Recommended: 1 disk

For the networking topology for multi-rack cluster, Leaf-Spine is recommended for the optimal performance.

Kafka

Kafka requires a fairly small amount of resources, especially with some configuration tuning. By default, Kafka, can run on as little as 1 core and 1GB memory with storage scaled based on requirements for data retention.

CPU is rarely a bottleneck because Kafka is I/O heavy, but a moderately-sized CPU with enough threads is still important to handle concurrent connections and background tasks.

Kafka brokers tend to have a similar hardware profile to HDFS data nodes. How you build them depends on what is important for your Kafka use cases.

Use the following guidelines:

To affect performance of these features:	Adjust these parameters:
Message Retention	Disk size
Client Throughput (Producer & Consumer)	Network capacity
Producer throughput	Disk I/O
Consumer throughput	Memory

A common choice for a Kafka node is as follows:

Component	Memory/Java Heap	CPU	Disk
Broker	• RAM: 64 GB • Recommended Java heap: 4 GB Set this value using the Java Heap Size of Broker Kafka configuration property. See Other Kafka Broker Properties table.	12- 24 cores	• 1 HDD For operating system • 1 HDD for Zookeeper dataLogDir • 10- HDDs, using Raid 10, for Kafka data
KRaft Controller	• RAM: 4 GB • Recommended Java heap: 4 GB Set these values using the Additional KRaft Controller Java Options Kafka property.	1 dedicated core	A single 64 GB SSD
Cruise Control	1 GB	1 core **Note:** A moderately-sized CPU with enough threads is important to handle metric fetching from Kafka and background tasks.	Because Cruise Control stores its data in Kafka the storage requirements will depend on the retention settings of the related Kafka topics.
Kafka Connect	0.5 - 4 GB heap size depending on the Connectors in use.	4 cores **Note:** Depends on the Connectors in use.	

Component	Memory/Java Heap	CPU	Disk
MirrorMaker	1 GB heap Set this value using the Java Heap Size of MirrorMaker Kafka configuration property.	1 core per 3-4 streams	No disk space needed on MirrorMaker instance. Destination brokers should have sufficient disk space to store the topics being copied over.
Schema Registry	1 GB heap	2 cores	1 MB Serialization JAR files may be uploaded and may be of any size. The disk usage depends on the JAR files uploaded. The files may be stored locally on the same host where SchemaRegistry is running or in HDFS if available.
Streams Messaging Manager **Note:** The hardware requirements for SMM depends on the number of Kafka partitions.	8 GB heap	8 cores	5 GB
Streams Replication Manager	• 1 GB heap for SRM driver • 1 GB heap for SRM Service	The performance of the SRM driver is mostly impacted by network throughput and latency.	No resources required

Networking requirements: Gigabit Ethernet or 10 Gigabit Ethernet. Avoid clusters that span multiple data centers.

Kafka and Zookeeper: It is common to run ZooKeeper on 3 broker nodes that are dedicated for Kafka. However, for optimal performance Cloudera recommends the usage of dedicated Zookeeper hosts. This is especially true for larger, production environments.

Ranger KMS

Component	Memory	CPU	Disk
Ranger KMS **Note:** Cloudera recommends using machines with CPUs that support the AES-NI instruction set and have a similar performance to the CPUs available to the NameNodes, so as not to introduce a bottleneck to HDFS client operations.	8 GB	1 GHz 64-bit quad core	20 GB, using moderate to high-performance drives

Kudu

Understand the resource requirements for Kudu before making the resource configuration changes in Cloudera Manager.

Component	Memory	CPU	Disk
Tablet Server	• Minimum: 4 GB • Recommended: 10 GB Additional hardware may be required, depending on the workloads running in the cluster.	Kudu currently requires a CPU that supports the SSSE3 and SSE4.2 instruction sets. If you are to run Kudu inside a VM, enable SSE4.2 pass-through to pass through SSE4.2 support into the VM.	1 disk for write-ahead log (WAL). Using an SSD drive may improve performance.

Component	Memory	CPU	Disk
Master	• Minimum: 256 MB • Recommended: 1 GB	Kudu currently requires a CPU that supports the SSSE3 and SSE4.2 instruction sets. If you are to run Kudu inside a VM, enable SSE4.2 pass-through to passthrough SSE4.2 support into the VM.	1 disk

Related Information

Apache Kudu configuration

Oozie

Understand the resource requirements for Oozie before making the resource configuration changes in Cloudera Manager.

Component	Java Heap	CPU	Disk
Oozie	• Minimum: 1 GB (this is the default set by Cloudera Manager). This is sufficient for less than 10 simultaneous workflows, without forking. • If you notice excessive garbage collection, or out-of-memory errors, increase the heap size to 4 GB for medium-size production clusters or to 8 GB for large-size production clusters. • Set this value using the Java Heap Size of Oozie Server in Bytes Oozie configuration property.	No resources required	No resources required

Additional tuning:

For workloads with many coordinators that run with complex workflows (a max concurrency reached! warning appears in the log and the Oozie admin -queuedump command shows a large queue):

- Increase the value of the oozie.service.CallableQueueService.callable.concurrency property to 50.
- Increase the value of the oozie.service.CallableQueueService.threads property to 200.

Do not use a Derby database as a backend database for Oozie.

Ozone hardware recommendations

This guide helps you choose hardware for Ozone based on your data storage needs. Following these recommendations will ensure that you get optimum performance from your Ozone cluster.

Table 7: Recommendations

The values are minimums and can go higher depending on your requirements. However, the above recommendations will apply to the vast majority of deployments.

Node Type	Chasis	CPU	Node RAM	RAM for each service	OS Disk	Meta Disk (NVMe)	Data Disk	Network	Disk Controllers	GPU
Master node (OM and SCM and Recon)	1U	2 x 20c	256 GB	64 GB	2 x 480 GB SSD	2 x 4 TB	-	2x 25Gbps	-	-

Node Type	Chasis	CPU	Node RAM	RAM for each service	OS Disk	Meta Disk (NVMe)	Data Disk	Network	Disk Controllers	GPU
Datanode (Ozone, no compute)	2U	2 x 12c		31 GB**		2 x 1.5 TB	24 x 16TB		2x 12 Gbps (low)	
Datanode (Ozone, mixed compute)		2 x 24c	512 GB			2 x 3 TB				Optional
Compute node (No Storage)	1U			-		1 x 4 TB	-	1x 25Gbps	-	

**Avoid using heap sizes of 32 GB to 47 GB because the JVM cannot use Compressed oops for heap sizes > 31 GB. This reduces the effective memory available to the process. If you want to configure heaps > 31 GB, then use a heap size of at least 48 GB or higher.

Notes

The above configuration will support up to 10B keys because of the 4 TB NVMe on the master nodes.

The absolute minimum recommended configuration is 3 master nodes and 9 datanodes. This will support Erasure Coding with the RS(6,3) configuration with full High Availability. Additional datanodes can be added in increments of 1 to increase storage.

Network

The network between the datanodes and the compute nodes cannot be oversubscribed by more than 2:1. Networking is sized to support the full (real-world) bandwidth of the drives across the network. More drives require faster networks, both at the server level and the switch level.

NVMe

NVMe should be configured in RAID1 pairs to provide business continuity for Ozone metadata in case of hardware failure.

The master nodes and datanodes use NVMe to store Ozone metadata. The compute nodes use NVMe for shuffle (Spark, MapReduce, and Tez) and caching (LLAP). The mixed compute datanodes use NVMe for both Ozone metadata and shuffle (Spark, MapReduce, and Tez) plus caching (LLAP).

Cloudera recommends mounting Ozone partitions across the NVMe drive pair as RAID1 (800GB) with the remaining space used for shuffle or cache as independent JBOD partitions. RAID can be configured either in hardware or in software.

Note:

- Cloudera recommends NVMe for storing OM and SCM metadata.
- Spinning media to store OM or SCM metadata is unsupported.
- For non-production clusters like staging or development clusters, write-optimized SSDs are recommended.

Example sizing calculator

Suggested Value of Parameter	Logical Capacity 2PB	Logical Capacity 8PB	Logical Capacity 16PB	Additional information
Number of Data Nodes if using Erasure Coding rs(6,3)	9	31	64	These are calculated based on actual file storage required (See Row 1)

Suggested Value of Parameter	Logical Capacity 2PB	Logical Capacity 8PB	Logical Capacity 16PB	Additional information
Logical data size proposed (TB, EC 6,3)	2304	8192	16384	-
Raw disk capacity (TB)	3456	12288	24576	-
Number of Data Nodes if using triple replication	16	64	128	These are calculated based on actual file storage required (See Row 1)
Logical data size - conservative using 3x (TB)	2048	8192	16384	-
Raw disk capacity (TB)	6144	24576	49152	-

Related Information

Ozone Architecture

Phoenix

Know the hardware resource requirements for Phoenix Query Server (PQS) in Cloudera Base on premises.

Component	Java Heap	CPU	Disk
Phoenix Query Server	1 GB - 4 GB Set this value using the Phoenix Query Server Max Heapsize configuration property. Increase this property value if you run any of these queries, aggregates, joins, or subqueries and if the query processing requires more memory.	Minimum 2 dedicated cores.	1 disk for local logs, which can be shared with the operating system and other Hadoop logs.

Ranger

Memory	CPU	Disk	Additional Dependencies
Ranger Admin: 1 GB minimum, then adjust heap as required (8 GB-16 GB)	1 core minimum	No special requirement.	
Ranger Usersync: 1 GB minimum	1 core minimum	No special requirement.	
Ranger Tagsync: 1 GB minimum	1 core minimum	No special requirement.	

Solr

Component	Java Heap	CPU	Disk
Solr	• Small workloads, or evaluations: 16 GB • Smaller production environments: 32 GB • Larger production environments: 96 GB is sufficient for most clusters. Set this value using the Java Heap Size of Solr Server in Bytes Solr configuration property.	• Minimum: 4 • Recommended: 16 for production workloads	No requirement if Solr uses HDFS for storage. If Solr uses local file system: • Faster disks, such as SSD can provide a significant performance improvement. • Occasionally a node may need disk space equal to 2-2.5 times the size of shards on the node for storage and overhead (for segment merging and shard recovery). • For heavy ingest and query loads Solr typically performs best on dedicated nodes, partially due to available OS cache for Solr files.

Note the following considerations for determining the optimal amount of heap memory:

- Size of searchable material: The more searchable material you have, the more memory you need. All things being equal, 10 TB of searchable data requires more memory than 1 TB of searchable data.
- Content indexed in the searchable material: Indexing all fields in a collection of logs, email messages, or Wikipedia entries requires more memory than indexing only the Date Created field.
- The level of performance required: If the system must be stable and respond quickly, more memory may help. If slow responses are acceptable, you may be able to use less memory.

Related Information

Deployment Planning for Cloudera Search

Spark

Component	Java Heap	CPU	Disk
Spark History Server	Minimum: 512 MB Set this value using the Java Heap Size of History Server in Bytes Spark configuration property.	1 **Important:** Cloudera recommends that you adjust the number of CPUs and memory for the Spark History Server based on your specific cluster usage patterns.	Minimum 1 disk for log files.

Livy

Understand the resource requirements for Livy before making the resource configuration changes in Cloudera Manager.

Component	Java Heap	CPU	Disk
Livy	Minimum: 512 MB Set this value using the Maximum size for the Java process heap memory Livy configuration property.	Minimum 1 core	Minimum 1 disk

Zeppelin

Understand the resource requirements for Zeppelin before making the resource configuration changes in Cloudera Manager.

Following are the resource requirements for Zeppelin to run together with the other components. In general, Zeppelin mainly contains one `Zeppelin Server` and multiple `Zeppelin Interpreters`. The interpreters may range from 2 to 6. By default, the four supported interpreters in Cloudera are JDBC, Livy, Markdown, and Angular.

Component	Java Heap	CPU	Memory
Zeppelin	Minimum: 1 GB Set the heap size using the Zeppelin Server Advanced Configuration Snippet (Safety Valve) for the `zeppelin-env.sh` property in Cloudera Manager Zeppelin configuration page by setting the ZEPPELIN_MEM and ZEPPELIN_INTP_MEM environment variables. For example, export ZEPPELIN_MEM = -Xms1024m -Xmx 2048m export ZEPPELIN_INTP_MEM= -Xms1024m -Xmx 2048m	The Zeppelin Server and Zeppelin Interpreter processes are not restricted in the container. So, the CPU requirement for the Zeppelin process is not easy to define in Cloudera. But the physical nodes that Zeppelin runs on should at least have 8 physical cores because Zeppelin may spin up hundreds of threads when busy.	**For Zeppelin Server:** Minimum memory: 1 GB Recommended minimum memory: 2 GB Maximum Memory: 16 GB Recommended maximum memory: 8 GB **For Zeppelin Interpreter:** Minimum memory: 1 GB Recommended minimum memory: 2 GB Maximum Memory: 16 GB Recommended maximum memory: 4 GB **Note:** By default, the running Zeppelin Server and all Interpreters use at least 1 + 4 = 5 GB memory in total. Cloudera recommends 64 GB for the physical node that Zeppelin runs on.

Note: You can change the resource configuration using Cloudera Manager based on your needs.

YARN

Component	Java Heap	CPU	Other Recommendations
Job History Server	• Minimum: 1 GB • Increase memory by 1.6 GB for each 100,000 tasks kept in memory. For example: 5 jobs @ 100, 000 mappers + 20,000 reducers = 600,000 total tasks requiring 9.6 GB of heap. See the Other Recommendations column for additional tuning suggestions. Set this value using the Java Heap Size of JobHistory Server in Bytes YARN configuration property.	Minimum: 1 core	• • Set the mapreduce.jobhistory.loadedtasks.cache.size property to a total loaded task count. Using the example in the Java Heap column to the left, of 650,000 total tasks, you can set it to 700,000 to allow for some safety margin. This should also prevent the JobHistoryServer from hanging during garbage collection, since the job count limit does not have a task limit.

Component	Java Heap	CPU	Other Recommendations
NodeManager	Minimum: 1 GB. Configure additional heap memory for the following conditions: • Large number of containers • Large shxmluffle sizes in Spark or MapReduce Set this value using the Java Heap Size of NodeManager in Bytes YARN configuration property.	• Minimum: 8-16 cores • Recommended: 32-64 cores	Disks: • Minimum: 8 disks • Recommended: 12 or more disks Networking: • Minimum: Dual 1Gbps or faster • Recommended: Single/Dual 10 Gbps or faster
ResourceManager	Minimum: 1 GB Configure additional heap memory for the following conditions: • More jobs • Larger cluster size • Number of retained finished applications (configured with the yarn.resourcemanager .max-completed-applications property. • Scheduler configuration Set this value using the Java Heap Size of ResourceManager in Bytes YARN configuration property.	Minimum: 1 core	
Other Settings	• Set the ApplicationMaster Memory YARN configuration property to 512 MB • Set the Container Memory Minimum YARN configuration property to 1 GB.	N/A	N/A

Related Information

Tuning Apache Hadoop YARN

ZooKeeper

Understand the resource requirements for ZooKeeper before making the resource configuration changes in Cloudera Manager.

Component	Java Heap	CPU	Disk
ZooKeeper Server	• Minimum: 1 GB • Increase heap size when watching 10,000 - 100,000 ephemeral znodes and are using 1,000 or more clients. Set this value using the Java Heap Size of ZooKeeper Server in Bytes ZooKeeper configuration property.	Minimum: 4 cores	ZooKeeper was not designed to be a low-latency service and does not benefit from the use of SSD drives. The ZooKeeper access patterns – append-only writes and sequential reads – were designed with spinning disks in mind. Therefore Cloudera recommends using HDD drives.

Related Information

Add a ZooKeeper service

Operating System Requirements

This topic describes the operating system requirements for Cloudera Base on premises. Cloudera Runtime 7.3.2 and Cloudera Manager 7.13.2 support JDK 17 only (OpenJDK, OracleJDK, and AzulJDK). Cloudera removed support for JDK 8 and JDK 11 in this release.



Important: Cloudera Runtime 7.3.2 and Cloudera Manager 7.13.2 support the x86-64 and ARM64 architectures. This release does not support IBM PowerPC.

Cloudera Base on premises Supported Operating Systems

See the Cloudera Support Matrix for detailed information about supported operating systems.

Important information about Cloudera Runtime and Cloudera Manager Supported Operating Systems

Cloudera Runtime provides parcels for select versions of RHEL-compatible operating systems.



Important:

In order to be covered by Cloudera Support:

- All Cloudera Runtime hosts in a logical cluster must run on the same major OS release.
- Cloudera supports a temporarily mixed OS configuration during an OS upgrade project.
- Cloudera Manager must run on the same OS release as one of the clusters it manages.

Cloudera recommends running the same minor release on all cluster nodes. However, the risk caused by running different minor OS releases is considered lower than the risk of running different major OS releases.

Points to note:

- Cloudera does not support Cloudera Runtime cluster deployments in Docker containers.
- Cloudera Enterprise is supported on platforms with Security-Enhanced Linux (SELinux) enabled and in enforcing mode. Cloudera is not responsible for policy support or policy enforcement. If you experience issues with SELinux, contact your OS provider.



Important: Cloudera Manager 7.13.2 supports only SLES 15 but not SLES 12. So it is not possible to have temporarily mixed OS configurations during the upgrade.

Software Dependencies

- Python - Python dependencies for the different Cloudera components is mentioned below:
  Cloudera Manager

  Cloudera recommends you to install Python 3.11 before upgrading Cloudera Manager to 7.13.2 version to ensure smooth transition with minimal downtime.

  Cloudera recommends you to install Python 3.9 or Python 3.11 before upgrading Cloudera Manager to 7.13.1 CHF5 version to ensure smooth transition with minimal downtime.

  For information about migrating Python for various operating systems, see Python Guidelines.

Important:

The following requirements apply to your specific Cloudera Manager version. You must refer to the *Python Support Matrix* for operating system-specific details.

Applicable Version	Dependency Note
Cloudera Manager 7.13.2	Verify supported Python versions for your specific Operating System in the Python support matrix.
Cloudera Manager 7.13.1	Verify supported Python versions for your specific Operating System in the Python support matrix.

Cloudera Data Explorer (Hue)

The following are the supported Python versions for the operating systems for the other Cloudera Runtime releases:

Python Version	Cloudera Runtime 7.1.9		Cloudera Runtime 7.3.1		Cloudera Runtime 7.3.2	
	7.1.9 SP1 CHF 9 (Cloudera Manager 7.13.1.400)	7.1.9 SP2	7.3.1	7.3.1.500 SP3 (Cloudera Manager 7.13.1.500)	7.3.2 (Cloudera Manager 7.13.2)	7.3.2 SP1
Python 3.12	Not supported	Not supported	Not supported	Not supported	Not supported	RHEL 10.1, Red Hat 8, Red Hat 9, Rocky Linux 9, RHEL 8 ARM 64, SLES 15, Ubuntu 22
Python 3.11	Red Hat 8, Red Hat 9, SLES 15, Ubuntu 22	Red Hat 8, Red Hat 9, SLES 15, Ubuntu 22	Not supported	Red Hat 8, Red Hat 9, Rocky Linux 9, RHEL 8 ARM 64, SLES 15, Ubuntu 22	Red Hat 8, Red Hat 9, Rocky Linux 9, RHEL 8 ARM 64, SLES 15, Ubuntu 22	RHEL 10.1, Red Hat 8, Red Hat 9, Rocky Linux 9, RHEL 8 ARM 64, SLES 15, Ubuntu 22
Python 3.10	Not supported	Not supported	SLES 15, Ubuntu 22	Not supported	Not supported	Not supported
Python 3.9	Red Hat 8, Red Hat 9	Not supported	Red Hat 8, Red Hat 9	Red Hat 8, Red Hat 9, Rocky Linux 9, RHEL 8 ARM 64	Not supported	Not supported
Python 3.8	Red Hat 8, SLES 12, Ubuntu 20, Ubuntu 22	Not supported	Ubuntu 20	Red Hat 8, Ubuntu 20, Ubuntu 22	Not supported	Not supported

Spark

Important: Spark 2 is not supported in Cloudera Runtime 7.3.1.

Spark 2.4 supports Python 2.7 and 3.4-3.7.

Important: Spark 2 will be deprecated in Cloudera Runtime 7.1.9. Therefore, 7.1.9 is the last runtime release where Spark 2 is supported. For more information, see Deprecation Notices in Cloudera Runtime.

Spark 2.4 supports Python 2.7 and 3.4-3.7.

Spark 3.0 supports Python 2.7 and 3.4 and higher, although support for Python 2 and 3.4 to 3.5 is deprecated.

Spark 3.1 supports Python 3.6 and higher.

If the right level of Python is not picked up by default, set the PYSPARK_PYTHON and PYSPARK_DRIVER_PYTHON environment variables to point to the correct Python executable before running the `pyspark` command.

CDS (Cloudera Distribution of Spark) 3.3 supports Python 3.7 and higher.



Note: For a complete list of supported Python versions in different Spark versions, refer to the *Spark Python Supportability Matrix*.

- Perl - Cloudera Manager requires perl.
- python-psycopg2 - Cloudera Manager 7 has a dependency on the package python-psycopg2. PostgreSQL-backed Hue in Runtime 7 requires a higher version of psycopg2 than is required by the Cloudera Manager dependency. For more information, see *Installing the psycopg2 Python Package*.
- iproute package - Cloudera Base on premises has a dependency on the iproute package. Any host that runs the Cloudera Manager Agent requires the package. The required version varies depending on the operating system:

Table 8: iproute package

Operating System	iproute version
RHEL	iproute
Ubuntu	iproute2
SLES	iproute2

- **rpcbind package** - Cloudera Base on premises has a dependency on the `rpcinfo` command which is usually found in the rpcbind package. Any host that runs the Cloudera Manager Agent requires this package. The required version varies depending on the operating system.

Filesystem Requirements

The Hadoop Distributed File System (HDFS) is designed to run on top of an underlying filesystem in an operating system.

Supported Filesystems

Cloudera recommends that you use either of the following filesystems tested on the supported operating systems:

- ext3: This is the most tested underlying filesystem for HDFS.
- ext4: This scalable extension of ext3 is supported in more recent Linux releases.



Important: Cloudera does not support in-place upgrades from ext3 to ext4. Cloudera recommends that you format disks as ext4 before using them as data directories.

- XFS: This is the default filesystem in RHEL.
- S3: Amazon Simple Storage Service

Kudu Filesystem Requirements - Kudu is supported on ext4 and XFS. Kudu requires a kernel version and filesystem that supports hole punching. Hole punching is the use of the fallocate(2) system call with the FALLOC_FL_PUNCH_HOLE option set.

File Access Time

Linux filesystems keep metadata that record when each file was accessed. This means that even reads result in a write to the disk. To speed up file reads, Cloudera recommends that you disable this option, called atime, using the noatime mount option in /etc/fstab:

```
/dev/sdb1 /data1 ext4 defaults,noatime 0
```

Apply the change without rebooting:

```
mount -o remount /data1
```

Filesystem Mount Options

The filesystem mount options have a sync option that allows you to write synchronously.

Using the sync filesystem mount option reduces performance for services that write data to disks, such as HDFS, YARN, Kafka and Kudu. In Cloudera, most writes are already replicated. Therefore, synchronous writes to disk are unnecessary, expensive, and do not measurably improve stability.

NFS and NAS options are not supported for use as DataNode Data Directory mounts, even when using Hierarchical Storage features.

Cloudera supports mounting /tmp with the noexec option. Mounting /tmp as a filesystem with the noexec option is sometimes done as an enhanced security measure to prevent the execution of files stored there.

Filesystem Requirements

You can control resource allocation for Cloudera Manager and Cloudera Runtime services (nproc, nofile, etc) from /etc/security/limits.conf, and through `init` scripts on traditional SysV Init systems. However, on systems using systemd the limits either needs to be set in the service's unit file, or in /etc/systemd/system.conf, or in files present under /etc/systemd/system.conf.d/*. This is due to a known limitation with systemd as it does not use PAM login sessions (pam_limits.so) for daemon services, thereby ignoring the limits defined in /etc/security/limits.conf. Both Cloudera Manager Agent and Supervisord (responsible for starting Cloudera Runtime services) are daemonised during system initialisation.

You can perform either of the following steps to modify the resource limit:

1. For system-wide change, uncomment the process properties from /etc/systemd/system.conf, or create an override .conf under /etc/systemd/system.conf.d/. This requires a **nix* system reboot for the changes to take effect. For more information, see Limits.conf.
2. To apply custom limits on Cloudera Runtime services, add the required process properties to the `[Service]` section in /usr/lib/systemd/system/cloudera-scm-supervisord.service.

   For instance, to customise the number of child processes a process can fork. You can set the property as follows:

   ```
   LimitNPROC=<value>
   ```

   Then reload the configuration by running the following command for the limits to be applied in the subsequent service restarts:

   ```
   # systemctl daemon-reload
   ```

   Here are the list of available process properties.

nscd for Kudu

Although not a strict requirement, it's highly recommended that you use nscd to cache both DNS name resolution and static name resolution for Kudu.

Configuring system level operating system

Cloudera recommends you to set up the following configurations:

- Disabling Transparent Hugepages (THP)
- vm.swappiness Linux Kernel Parameter

For setting these configurations, see Disabling Transparent Hugepages (THP) and Setting the vm.swappiness Linux Kernel Parameter.

Related Information

Installing the psycopg2 Python package for PostgreSQL-backed Hue
Spark Python Supportability Matrix

Database Requirements

This topic describes the database requirements for Cloudera Base on premises.

See Cloudera Support Matrix for detailed information about supported databases based on the Cloudera and Cloudera Manager versions.

Important: When you restart processes, the configuration for each of the services is redeployed using information saved in the Cloudera Manager database. If this information is not available, your cluster cannot start or function correctly. You must schedule and maintain regular backups of the Cloudera Manager database to recover the cluster in the event of the loss of this database. For more information, see *Backing Up Databases*.

Cloudera Manager and Cloudera 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. You must ensure latency between Cloudera Manager server and the database is < 10 ms. You can verify the latency with a simple SQL command from your Cloudera Manager server host to the database. Start your database's command line client tool and connect to the Cloudera Manager database. Run the SQL command `SELECT 1;`

Important: Migrating from one external database server to a different type of database server after installing Cloudera Manager and Cloudera Runtime services is a complex process that requires modification of the schema and matching the data in the database tables to the new schema. Cloudera expects you to perform the migration using any off-the-shelf tool. If you require any assistance, Cloudera recommends that you must engage with Cloudera Professional Services team.

After installing a database, upgrade to the latest patch and apply appropriate updates. Available updates may be specific to the operating system on which it is installed.

Notes:

- Cloudera recommends installing the databases on different hosts than the services, located in the same data center. Separating databases from services can help isolate the potential impact from failure or resource contention in one or the other. It can also simplify management in organizations that have dedicated database administrators.
- Cloudera does not support Percona for MySQL as a backend database for Hive Metastore (HMS).
- Use the appropriate UTF8 encoding for Metastore, Oozie, Hive, and Hue.

  Oozie also supports UTF8MB4 character encoding out of box without any configuration change when the Oozie custom database is created with the encoding of UTF8MB4.

  MySQL and MariaDB must use the MySQL utf8 encoding, not utf8mb4.
- Ranger only supports the InnoDB engine for MySQL and MariaDB databases.
- YARN Queue Manager requires a Postgres database server with a dedicated database.
- For MySQL 5.7, you must install the MySQL-shared-compat or MySQL-shared package. This is required for the Cloudera Manager Agent installation.
- MySQL GTID-based replication is not supported.

- Both the Community and Enterprise versions of MySQL are supported, as well as MySQL configured by the AWS RDS service.
- Before upgrading from CDH 5 to CDH 6, check the value of the COMPATIBLE initialization parameter in the Oracle Database using the following SQL query:

  ```
  SELECT name, value FROM v$parameter WHERE name = 'compatible'
  ```

  The default value is 12.2.0. If the parameter has a different value, you can set it to the default as shown in the Oracle Database Upgrade Guide.

  Note: Before resetting the COMPATIBLE initialization parameter to its default value, make sure you consider the effects of this change can have on your system.

Related Information
Required Databases

RDBMS High Availability Support

Various Cloudera components rely on backing RDBMS services as critical infrastructure. You may require Cloudera components to support deployment in environments where RDBMS services are made highly-available. High availability (HA) solutions for RDBMS are implementation-specific, and can create constraints or behavioral changes in Cloudera components.

This section clarifies the support state and identifies known issues and limitations for HA deployments.

Upgrading Cloudera Manager and the Cloudera Manager database
When upgrading Cloudera Manager, there may be a minimum version requirement for the database server.

Ensure that the Cloudera Manager database server is upgraded to at least this minimum requirement prior to starting the new version of Cloudera Manager for the first time.

1. Stop Cloudera Management Service.
2. Upgrade the Cloudera Manager RPMs.
3. Upgrade the Cloudera Manager database server version.
4. Start the Cloudera Manager.

Upgrading Cloudera Base on premises and the Cloudera Base on premises Services databases
When upgrading Cloudera Base on premises to a new version, the new version of Cloudera Base on premises may have a minimum version requirement for the database server which holds services metadata.

If the database server needs to be upgraded, follow this proces:

1. Stop Cloudera Base on premises services which depend on the database server, or alternatively, stop the entire cluster.
2. Upgrade the database server version.
3. Upgrade Cloudera Base on premises using Cloudera Manager.
4. Start the stopped services, or the entire cluster.

   When a database server upgrade is required, it is not possible to perform a rolling upgrade of the cluster.

Difference between High Availability and Load Balancing

Understanding the difference between HA and load balancing is important for Cloudera components, which are designed to assume services are provided by a single RDBMS instance. Load balancing distributes operations across multiple RDBMS services in parallel, while HA focuses on service continuity. Load balanced deployments are often used as part of HA strategies to overcome demands of monitoring and failover management in an HA environment. While less easier to implement, load-balanced deployments require applications tailored to the behavior and limitations of the particular technology.

Support Statement: Cloudera components are not designed for and do not support load balanced deployments of any kind. Any HA strategy involving multiple active RDBMS services must ensure all connections are routed to a single RDBMS service at any given time, regardless of vendor or HA implementation/technology.

General High Availability Support

Cloudera supports various RDBMS options, each of which have multiple possible strategies to implement HA. Cloudera cannot reasonably test and certify on each strategy for each RDBMS. Cloudera expects HA solutions for RDBMS to be transparent to Cloudera software, and therefore are not supported and debugged by Cloudera. It is the responsibility of the customer to provision, configure, and manage the RDBMS HA deployment, so that Cloudera software behaves as it would when interfacing with a single, non-HA service. Cloudera will support and help customers troubleshoot issues when a cluster has HA enabled. While diagnosing database-related problems in Cloudera components, customers may be required to temporarily disable or bypass HA mechanisms for troubleshooting purposes. If an HA-related issue is found, it is the responsibility of the customer to engage with the database vendor so that a solution to that issue can be found.

Support Statement: Cloudera Support may require customers to temporarily bypass HA layers and connect directly to supported RDBMS back-ends to troubleshoot issues. Issues observed only when connected through HA layers are the responsibility of the customer DBA staff to resolve.

RDBMS Storage Sizing

The amount of RDBMS storage space used by Cloudera Base on premises varies depending on the services that are installed and the operations performed. Cloudera Base on premises requires at least 500 GB RDBMS storage for production systems.



Important: For more information about database storage and disk layout, see Disk layout.

You can better estimate the RDBMS storage space by deploying a test cluster with the approximate proportion of service roles that the full cluster can bear. Later, execute a sample set of operations, (including Data Recovery backup) for about 24 hours and observe the storage usage on the RDBMS. Next, extrapolate the usage to the full cluster size.

Sharing an RDBMS with other applications

The ability to share an RDBMS storage between Cloudera Base on premises and other applications depends on many factors. Cloudera recommends that you do not share the RDBMS used by Cloudera Base on premises with any other application.

For non-production clusters where cluster size is small, not expected to grow, and occasional glitches are tolerable, it is acceptable to share a database with other applications.

MySQL

For a production cluster, Cloudera Runtime services must not share a database server with other applications. For small clusters, this database can be shared by the Cloudera Runtime services. For large clusters (hosts > 500), each Cloudera Runtime service must have its own database server.

PostgreSQL

If you have a dedicated database team managing high-performance hardware, with the Cloudera Base on premises databases stored on their own spindles (or raid array), then it can be possible to have the DB server shared with other applications. When the cluster size is very large (hosts > 1000), there might be performance issues between shared applications. Cloudera recommends that you do not share the Cloudera Base on premises database server with other application usages.

If you do not have a dedicated database team that can analyze and tune RDBMS performance, it is recommended to follow the advice for MySQL as detailed above.

Oracle

For single-server Oracle installations, see the above description related to PostgreSQL.

If you are using a clustered system like Oracle RAC, with multiple servers, it is possible to use a shared DB service, since it is no longer a single server. The end user's DB team must monitor DB latency, scale the hardware, or tune DB parameters to ensure optimal performance.

Latency target

For end users attempting to tune a shared RDBMS, ensure that elapsed times must not exceed 40 milliseconds for the 99th percentile of SELECT statements on indexed single-row queries.

Java Requirements

Cloudera Base on premises comprises Cloudera Manager and Cloudera Runtime services. To successfully upgrade the service, you need the following Java prerequisites.

Important:

Starting with Cloudera Manager 7.13.2.0, Cloudera removes the Install a Cloudera-provided version of OpenJDK installation option from the Cloudera Manager UI and deleted the OpenJDK 8 RPM package from all Cloudera Download Repository locations. You must manually install a supported version of OpenJDK or Oracle JDK on the Cloudera Manager Server host and all cluster managed hosts before you install or upgrade Cloudera Manager. For a list of compatible Java versions, see Cloudera Support Matrix.

Supported JDK versions

See the Cloudera Support Matrix for detailed information about supported JDK versions.

As Cloudera transitions to JDK 17 as its default OpenJDK release, the upgrade flow assumes that you must upgrade the existing OpenJDK from the current release to JDK 17 before starting the upgrade process, unless specifically stated otherwise.

- The current version of OpenJDK 17 is 17.0.16 (as of September 2025).
- For OpenJDK, Oracle JDK, or Azul JDK, Cloudera requires a minimum release of 17.0.11. Use the latest releases, as Cloudera supports them and prefers them, even if documentation does not list them.

Configuring Required JDK 17 JVM Parameters (add-opens and add-exports) for Cloudera Runtime Components

When using JDK 17, Cloudera Runtime components and applications (such as YARN, MapReduce, Spark, and Hive) require explicit Java reflection and package export access (--add-opens and --add-exports flags) to run successfully.

In Cloudera Manager 7.13.1 and 7.13.2 / Cloudera Runtime 7.3.1, 7.3.2, and higher versions, these required parameters are automatically pre-configured by default. No manual configuration is required.

However, for CDP Private Cloud Base 7.1.9 (Base, SP1, and SP2), you must verify their presence. If missing, configure the relevant service in Cloudera Manager to handle them automatically by performing the following steps.

YARN (MapReduce and DistCP Applications)

Running MapReduce Application Masters (AM) and MapReduce containers with JDK 17 requires specific Java reflection access flags. These parameters are typically managed automatically in the NodeManager environment (yarn_nodemanager_admin_env).

You must verify if these parameters are present. If they are missing, you do not need to turn on container flags manually. Instead, you can configure Cloudera Manager to handle them automatically using one of the following two methods:
Method 1: Enable mapreduce_jvm_add_opens_as_default (Recommended)

This property automatically appends the required JDK 17 flags to the MapReduce Application Master (AM) and map/reduce container launch commands.

1. Log in to the Cloudera Manager Admin Console.
2. From the main dashboard or the Clusters drop-down menu, select your cluster and click on the YARN service.
3. Click on the Configuration tab at the top of the YARN service page.
4. In the Search filter box on the left, type or paste the exact configuration key: mapreduce_jvm_add_opens_as_default
5. Locate the property in the filtered results (it is typically titled Add add-opens flags to MR containers).
6. Select the checkbox to change its value from false to true.
7. Click Save Changes.
8. Select Restart from the list of Actions at the top of the page to restart the affected YARN/MapReduce service components.

Method 2: Manually Update the NodeManager Admin Environment

If you prefer not to use the automated flag, you can explicitly inject the three required parameters into the NodeManager service's environment configurations.

1. Log in to the Cloudera Manager Admin Console.
2. From the main dashboard or the Clusters drop-down menu, select your cluster and click on the YARN service.
3. Click on the Configuration tab at the top of the YARN service page.
4. In the Search filter box on the left, type or paste the configuration key: yarn_nodemanager_admin_env
5. Locate the Containers Environment Variable property.
6. Click into the value field and append the following three flags on a single line (ensure they are space-separated):

```
--add-opens=java.base/java.lang=ALL-UNNAMED --add-exports=java.base/sun.net.dns=ALL-UNNAMED --add-exports=java.base/sun.net.util=ALL-UNNAMED
```

7. Click Save Changes.
8. Select Restart from the list of Actions at the top of the page to restart the affected YARN/MapReduce service components.

Important JDK Support Matrix and Configuration Notes

Review critical Java compatibility requirements, configuration guidelines, and operational constraints for Cloudera Manager and Cloudera Runtime components before deploying or updating your JDK environment.

Important:
Kerberos Encryption Prerequisite for OpenJDK 17 Upgrade (aes256-cts)

Upgrading to OpenJDK 17 assumes that no deprecated Kerberos encryption types are in use across your cluster services. Before attempting the OpenJDK 17 upgrade, you must ensure that aes256-cts is configured as the base encryption type on both the KDC side and client side.

Recommended Garbage Collector (G1GC) and Heap Configuration for Spark on JDK 17

JDK 17 introduced changes in garbage collection, moving from ParallelGC to G1GC as the default method, which is also the recommended method for Apache Spark.

Cloudera recommends increasing -XX:G1HeapRegionSize to prevent an excess of region allocations, depending on the workload and heap size assigned to the executor.

OpenJDK Support and Testing Scope

- Cloudera tests only the OpenJDK builds provided by each supported operating system vendor and specifically listed in the Cloudera Support Matrix.
- Minor update releases above the stated minimum are supported unless explicitly excluded, but might not be individually tested. Cloudera reserves the right to exclude specific Java updates if security is compromised.

TLS version support across Cloudera Manager and Cloudera Runtime on JDK 17

With JDK 17, Cloudera Manager supports both TLS 1.2 and TLS 1.3, while Cloudera Runtime release supports TLS 1.2 only.

Operating System OpenJDK Distribution Requirement

You must ensure that the operating system chosen for deployment includes the required OpenJDK release as part of its native distribution package.

Networking and Security Requirements

This topic describes the networking and security requirements for Cloudera Base on premises.

Cloudera Runtime and Cloudera Manager Supported Transport Layer Security Versions

Note:

- Cloudera Manager supports TLS 1.2 for Java 8, Java 11, and Java 17 or higher versions. For TLS 1.0 and TLS 1.1, Cloudera Manager supports Java 8, though Cloudera recommends avoiding the use of TLS 1.0 and TLS 1.1.
- Domain name requirement: Avoid environments where domain names contain non-LDH characters (characters other than letters, digits, or hyphens) for Hadoop deployments, as they are unsupported.

The following components are supported by the indicated versions of Transport Layer Security (TLS):

Table 9: Components Supported by TLS

Component	Role	Name	Port	Version
Cloudera Manager	Cloudera Manager Server		7182	TLS 1.2
Cloudera Manager	Cloudera Manager Server		7183	TLS 1.2
Flume			9099	TLS 1.2
Flume		Avro Source/Sink		TLS 1.2

Component	Role	Name	Port	Version
Flume		Flume HTTP Source/Sink		TLS 1.2
HBase	Master	HBase Master Web UI Port	60010	TLS 1.2
HDFS	NameNode	Secure NameNode Web UI Port	50470	TLS 1.2
HDFS	Secondary NameNode	Secure Secondary NameNode Web UI Port	50495	TLS 1.2
HDFS	HttpFS	REST Port	14000	TLS 1.1, TLS 1.2
Hive	HiveServer2	HiveServer2 Port	10000	TLS 1.2
Hue	Hue Server	Hue HTTP Port	8888	TLS 1.2
Impala	Impala Daemon	Impala Daemon Beeswax Port	21000	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala Daemon	Impala Daemon HiveServer2 Port	21050	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala Daemon	Impala Daemon Backend Port	22000	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala StateStore	StateStore Service Port	24000	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala Daemon	Impala Daemon HTTP Server Port	25000	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala StateStore	StateStore HTTP Server Port	25010	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala Catalog Server	Catalog Server HTTP Server Port	25020	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala Catalog Server	Catalog Server Service Port	26000	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Oozie	Oozie Server	Oozie HTTPS Port	11443	TLS 1.1, TLS 1.2
Ranger	Ranger Admin	Admin HTTPS Port	6182	TLS 1.2
Ranger KMS	Ranger KMS Server	Ranger KMS HTTPS Port	9494	TLS 1.2
Solr	Solr Server	Solr HTTP Port	8983	TLS 1.1, TLS 1.2
Solr	Solr Server	Solr HTTPS Port	8985	TLS 1.1, TLS 1.2
Spark	History Server		18080	TLS 1.2

Component	Role	Name	Port	Version
YARN	ResourceManager	ResourceManager Web Application HTTP Port	8090	TLS 1.2
YARN	JobHistory Server	MRv1 JobHistory Web Application HTTP Port	19890	TLS 1.2

Cloudera Runtime and Cloudera Manager Networking and Security Requirements

The hosts in a Cloudera Manager deployment must satisfy the following networking and security requirements:

- Cluster Host Requirements:

  The hosts you intend to use must satisfy the following requirements:

  - You must be able to log in to the Cloudera Manager Server host using the root user account or an account that has passwordless sudo privileges.
  - The Cloudera Manager Server host must have uniform SSH access on the same port to all hosts. For more information, see *Cloudera Runtime and Cloudera Manager Networking and Security Requirements*.
  - All hosts must have access to standard package repositories for the operating system and either archive.cloudera.com or a local repository with the required installation files.
  - SELinux must be disabled or set to permissive mode before running the installer.
- Networking Protocols Support

  Cloudera requires IPv4. IPv6 is not supported and must be disabled.

  Important: Refer to your OS documentation or contact your OS vendor for instructions on disabling IPv6.

  See also *Configure Network Names*.
- Multihoming Support

  Multihoming Cloudera Runtime or Cloudera Manager is not supported outside specifically certified Cloudera partner appliances such as Oracle Big Data Appliance, Teradata Appliance for Hadoop, and Cray® Urika-XA system. Cloudera finds that current Hadoop architectures combined with modern network infrastructures and security practices remove the need for multihoming. Multihoming, however, is beneficial internally in appliance form factors to take advantage of high-bandwidth InfiniBand interconnects.

  Although some subareas of the product may work with unsupported custom multihoming configurations, there are known issues with multihoming. In addition, unknown issues may arise because multihoming is not covered by our test matrix outside the Cloudera-certified partner appliances.
- Entropy

  Data at rest encryption requires sufficient entropy to ensure randomness.

  See entropy requirements in *Data at Rest Encryption Requirements*.
- Cluster hosts must have a working network name resolution system and correctly formatted /etc/hosts file. All cluster hosts must have properly configured forward and reverse host resolution through DNS. The /etc/hosts files must:

  - Contain consistent information about hostnames and IP addresses across all hosts
  - Not contain uppercase hostnames
  - Not contain duplicate IP addresses

  Cluster hosts must not use aliases, either in /etc/hosts or in configuring DNS. A properly formatted /etc/hosts file should be similar to the following example:

```
127.0.0.1 localhost.localdomain localhost
192.168.1.1 cluster-01.example.com cluster-01
192.168.1.2 cluster-02.example.com cluster-02
```

```
192.168.1.3 cluster-03.example.com cluster-03
```

- In most cases, the Cloudera Management Service must have SSH access to the cluster hosts when you run the installation or upgrade wizard. You must log in using a root account or an account that has password-less sudo permission. For authentication during the installation and upgrade procedures, you must either enter the password or upload a public and private key pair for the root or sudo user account. If you want to use a public and private key pair, the public key must be installed on the cluster hosts before you use Cloudera Manager.

  Cloudera Manager uses SSH only during the initial install or upgrade. Once the cluster is set up, you can disable root SSH access or change the root password. Cloudera Manager does not save SSH credentials, and all credential information is discarded when the installation is complete.
- The Cloudera Manager Agent runs as root so that it can make sure that the required directories are created and that processes and files are owned by the appropriate user (for example, the hdfs and mapred users).
- Security-Enhanced Linux (SELinux) must not block Cloudera Manager or Cloudera Runtime operations.

  Note: Cloudera Enterprise is supported on platforms with Security-Enhanced Linux (SELinux) enabled and in enforcing mode. Cloudera is not responsible for SELinux policy development, support, or enforcement. If you experience issues running Cloudera software with SELinux enabled, contact your OS provider for assistance.

  If you are using SELinux in enforcing mode, Cloudera Support can request that you disable SELinux or change the mode to permissive to rule out SELinux as a factor when investigating reported issues.
- Firewalls (such as iptables and firewalld) must be disabled or configured to allow access to ports used by Cloudera Manager, Cloudera Runtime, and related services.
- For RHEL and CentOS, the /etc/sysconfig/network file on each host must contain the correct hostname.
- Cloudera Manager and Cloudera Runtime use several user accounts and groups to complete their tasks. The set of user accounts and groups varies according to the components you choose to install. Do not delete these accounts or groups and do not modify their permissions and rights. Ensure that no existing systems prevent these accounts and groups from functioning. For example, if you have scripts that delete user accounts not in an allowlist, add these accounts to the list of permitted accounts. Cloudera Manager, Cloudera Runtime, and managed services create and use the following accounts and groups:

Table 10: Users and Groups

Component (Version)	Unix User ID	Groups	Functionality
Apache Atlas	atlas	atlas, hadoop	Apache Atlas by default has atlas as user and group. It is configurable
Apache Flink	flink	flink	The Flink Dashboard runs as this user.
Apache HBase	hbase	hbase	The Master and the RegionServer processes run as this user.
Apache HBase Indexer	hbase	hbase	The indexer servers are run as this user.
Apache HDFS	hdfs	hdfs, hadoop	The NameNode and DataNodes run as this user, and the HDFS root directory as well as the directories used for edit logs should be owned by it.

Component (Version)	Unix User ID	Groups	Functionality
Apache Hive Hive on Tez	hive	hive	The HiveServer2 process and the Hive Metastore processes run as this user.A user must be defined for Hive access to its Metastore DB (for example, MySQL or Postgres) but it can be any identifier and does not correspond to a Unix uid. This is javax.jdo.option.ConnectionUserName in hive-site.xml.
Apache Impala	impala	impala, hive	Impala services run as this user.
Apache Kafka	kafka	kafka	Kafka brokers, mirrorMaker, and Connect workers run as this user.
Apache Knox	knox	knox	Apache Knox Gateway Server runs as this user
Apache Kudu	kudu	kudu	Kudu services run as this user.
Apache Livy	livy	livy	The Livy Server process runs as this user
Apache NiFi	nifi	nifi	Runs as the nifi user
Apache NiFi Registry	nifiregistry	nifiregistry	Runs as the nifiregistry user
Apache Oozie	oozie	oozie	The Oozie service runs as this user.
Apache Ozone	hdfs	hdfs, hadoop	Ozone Manager, Storage Container Manager (SCM), Recon and Ozone Datanodes run as this user.
Apache Parquet	~	~	No special users.
Apache Phoenix	phoenix	phoenix	The Phoenix Query Server runs as this user
Apache Ranger	ranger	ranger, hadoop	Ranger Admin, Usersync and Tagsync services by default have ranger as user and ranger, hadoop as groups. It is configurable.
Apache Ranger KMS	kms	kms	Ranger KMS runs with kms user and group. It is configurable.
Apache Ranger Raz	rangerraz	ranger	Ranger Raz runs with rangerraz user and is part of the ranger group.
Apache Ranger RMS	rangerrms	ranger	Ranger RMS runs with rangerrms user and is part of the ranger group.
Apache Solr	solr	solr	The Solr processes run as this user.
Apache Spark	spark	spark	The Spark History Server process runs as this user.
Apache Sqoop	sqoop	sqoop	This user is only for the Sqoop1 Metastore, a configuration option that is not recommended.

Component (Version)	Unix User ID	Groups	Functionality
Apache YARN	yarn	yarn, hadoop	Without Kerberos, all YARN services and applications run as this user. The LinuxContainerExecutor binary is owned by this user for Kerberos.
Apache Zeppelin	zeppelin	zeppelin	The Zeppelin Server process runs as this user
Apache ZooKeeper	zookeeper	zookeeper	The ZooKeeper processes run as this user. It is not configurable.
(all versions)	cloudera-scm	cloudera-scm	Clusters managed by run Server, monitoring roles, and other Server processes as cloudera-scm. Requires keytab file named cmf.keytab because name is hard-coded in .
Cruise Control	cruisecontrol	hadoop	The Cruise Control process runs as this user.
HttpFS	httpfs	httpfs	The HttpFS service runs as this user. See “HttpFS authentication” for instructions on how to generate the merged httpfs-http.keytab file.
Hue	hue	hue	Hue services run as this user.
Hue Load Balancer	apache	apache	The Hue Load balancer has a dependency on the apache2 package that uses the apache user name. does not run processes using this user ID.
Schema Registry	schemaregistry	hadoop	The Schema Registry process runs as this user.
	streamsmsgmgr	streamsmsgmgr	The processes run as this user.
	streamsrepmgr	streamsrepmgr	The processes run as this user.

Data at Rest Encryption Requirements

This topic describes the data at rest encryption requirements for Cloudera Base on premises.

Encryption comprises several components, each with its own requirements.

Data at rest encryption protection can be applied at a number of levels within Hadoop:

- OS filesystem-level
- Network-level
- HDFS-level (protects both data at rest and in transit)

This section contains the various hardware and software requirements for all encryption products used for Data at Rest Encryption.

For more information on supported operating systems, see Cloudera Support Matrix.

For more information on the components, concepts, and architecture for encrypting data at rest, see Encrypting Data at Rest.

Third-party filesystems

This topic describes the third-party filesystems supported by Cloudera Base on premises.

Cloudera Base on premises supports the following third-party filesystems:

Trial Installation

These topics provide instructions for installing the trial version of Cloudera Base on premises in a non-production environment for demonstration and proof-of-concept use cases.

In these procedures, Cloudera Manager automates the installation of the JDK, Cloudera Management Service Server, an embedded PostgreSQL database, Cloudera Manager Agent, Cloudera Runtime, and other managed services on cluster hosts. Cloudera Manager also configures databases for the Cloudera Management Service and Hive Metastore, Ranger, and for Cloudera Management Service roles.

This installation method is recommended for trial deployments, but is not supported for production deployments because it is not designed to scale. To use this method, server and cluster hosts must satisfy the following requirements:

- All hosts must have a supported operating system installed.
- You must be able to log in to the Cloudera Management Service host using the root user account or an account that has passwordless sudo privileges.
- The Cloudera Management Service host must have uniform SSH access on the same port to all hosts. For more information, see Cloudera Runtime and Cloudera Manager Networking and Security Requirements.
- All hosts must have access to standard package repositories for the operating system and either archive.cloudera .com or a local repository with the required installation files.
- SELinux must be disabled or set to permissive mode before running the installer.

Related Information

Cloudera Base on premises Trial Download Information

Cloudera Base on premises Installation Guide

Installing a Trial Cluster

In this procedure, Cloudera Manager automates the installation of the Oracle JDK, Cloudera Manager Server, embedded PostgreSQL database, Cloudera Manager Agent, Cloudera Runtime, and managed service software on cluster hosts. Cloudera Manager also configures databases for the Cloudera Manager Server and Hive Metastore and optionally for Cloudera Management Service roles.



Important: This procedure is intended for trial and proof-of-concept deployments only. It is not supported for production deployments because it is not designed to scale.

Cluster Host Requirements:

The hosts you intend to use must satisfy the following requirements:

- You must be able to log in to the Cloudera Manager Server host using the root user account or an account that has passwordless sudo privileges.
- The Cloudera Manager Server host must have uniform SSH access on the same port to all hosts. For more information, see *Cloudera Runtime and Cloudera Manager Networking and Security Requirements*.
- All hosts must have access to standard package repositories for the operating system and either archive.cloudera .com or a local repository with the required installation files.
- SELinux must be disabled or set to permissive mode before running the installer.

Refer to the following topics for the steps required to install a trial cluster.

Before You Begin a Trial Installation

Before you begin a trial installation, you must disable SELinux if you want the Cloudera Manager installer to run. You can also optionally configure an HTTP proxy.

(Optional) Configure an HTTP Proxy

The Cloudera Manager installer accesses archive.cloudera.com by using yum on RHEL systems. zypper on SLES systems, or apt-get on Ubuntu systems. If your hosts access the Internet through an HTTP proxy, you can configure yum system-wide, to access archive.cloudera.com through a proxy.

To do so, modify the system configuration on every cluster host as follows:

OS	File	Property
RHEL-compatible	/etc/yum.conf	proxy=http://*SERVER*:*PORT*/
SLES	/root/.curlrc	--proxy=http://*SERVER*:*PORT*/
Ubuntu	/etc/apt/apt.conf	Acquire::http::Proxy "http ://*SERVER*:*PORT*";

Disable SELinux



Note: Cloudera Base on premises is supported on platforms with Security-Enhanced Linux (SELinux) enabled and in enforcing mode. Cloudera is not responsible for SELinux policy development, support, or enforcement. If you experience issues running Cloudera software with SELinux enabled, contact your OS provider for assistance.

If you are using SELinux in enforcing mode, Cloudera Support can request that you disable SELinux or change the mode to permissive to rule out SELinux as a factor when investigating reported issues.

Although Cloudera supports running Cloudera software with SELinux enabled, the Cloudera Manager installer will not proceed if SELinux is enabled. Disable SELinux or set it to permissive mode before running the installer.

After you have installed and deployed Cloudera Manager and Cloudera Runtime, you can re-enable SELinux by changing SELINUX=permissive back to SELINUX=enforcing in /etc/selinux/config (or /etc/sysconfig/selinux), and then running the following command to immediately switch to enforcing mode:

```
setenforce 1
```

If you are having trouble getting Cloudera Software working with SELinux, contact your OS vendor for support. Cloudera is not responsible for developing or supporting SELinux policies.

Download the Trial version of Cloudera Base on premises

You can download the trial version of Cloudera Base on premises from the Cloudera Download site.

About this task

You can use the trial software for 60 days without obtaining a license key file. The trial installation includes an embedded PostgreSQL database and is not suitable for a production environment.

Procedure

1. Go to the trial download page for Cloudera Base on premises.
2. Click Try Now.
3. Follow the download-instructions.

What to do next

Run the Cloudera Manager Server Installer.

Run the Cloudera Manager Server Installer

Run the Cloudera Manager installer to the cluster host to which you are installing the Cloudera Manager Server. By default, the automated installer binary (cloudera-manager-installer.bin) installs the highest version of Cloudera Manager.

Before you begin

- Download the trial software.

Procedure

1. Run the Cloudera Manager installer:
   a) Change cloudera-manager-installer.bin to have execute permissions:

      ```
      chmod u+x cloudera-manager-installer.bin
      ```

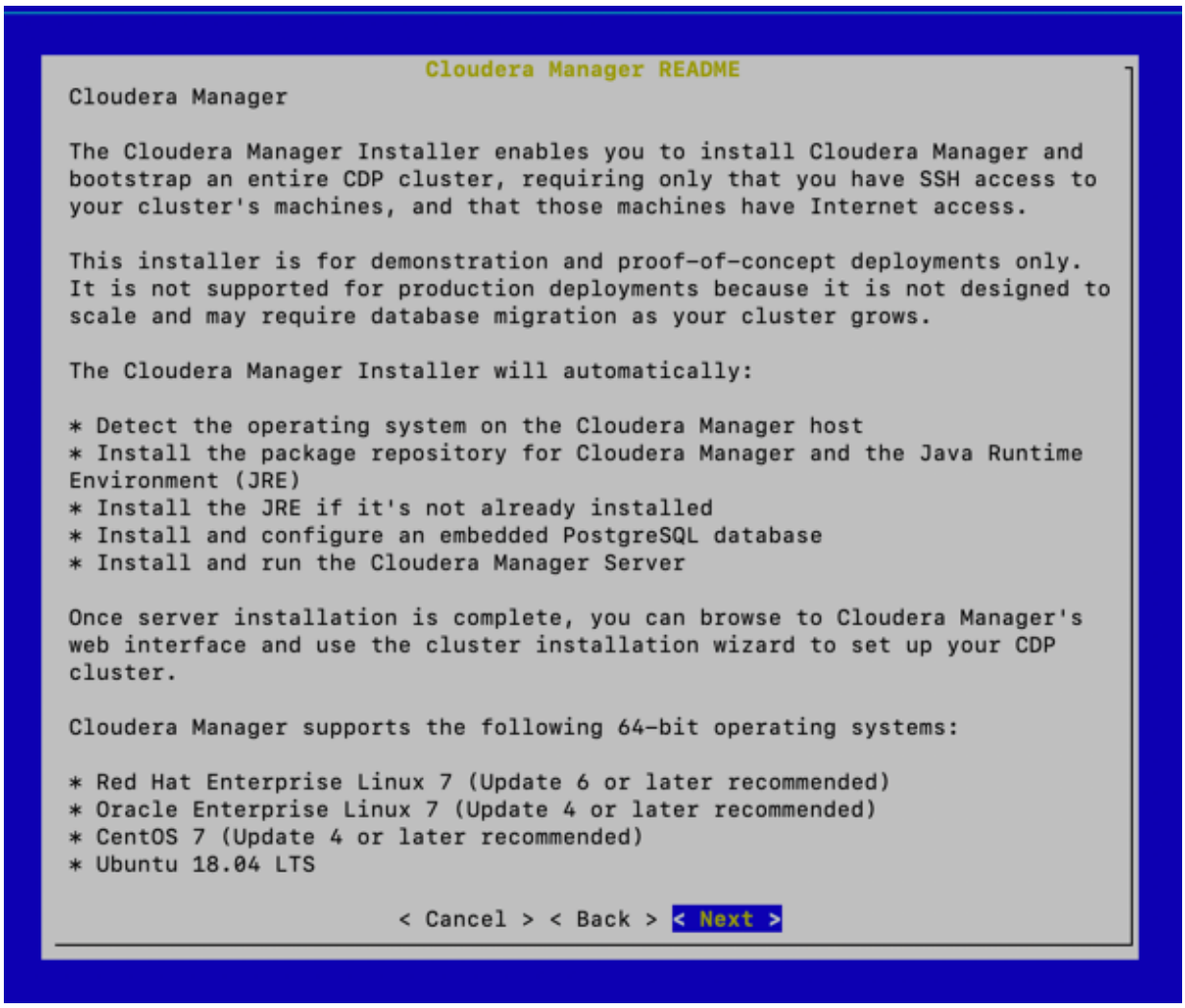
   b) Run the Cloudera Manager Server installer:

      ```
      sudo ./cloudera-manager-installer.bin
      ```

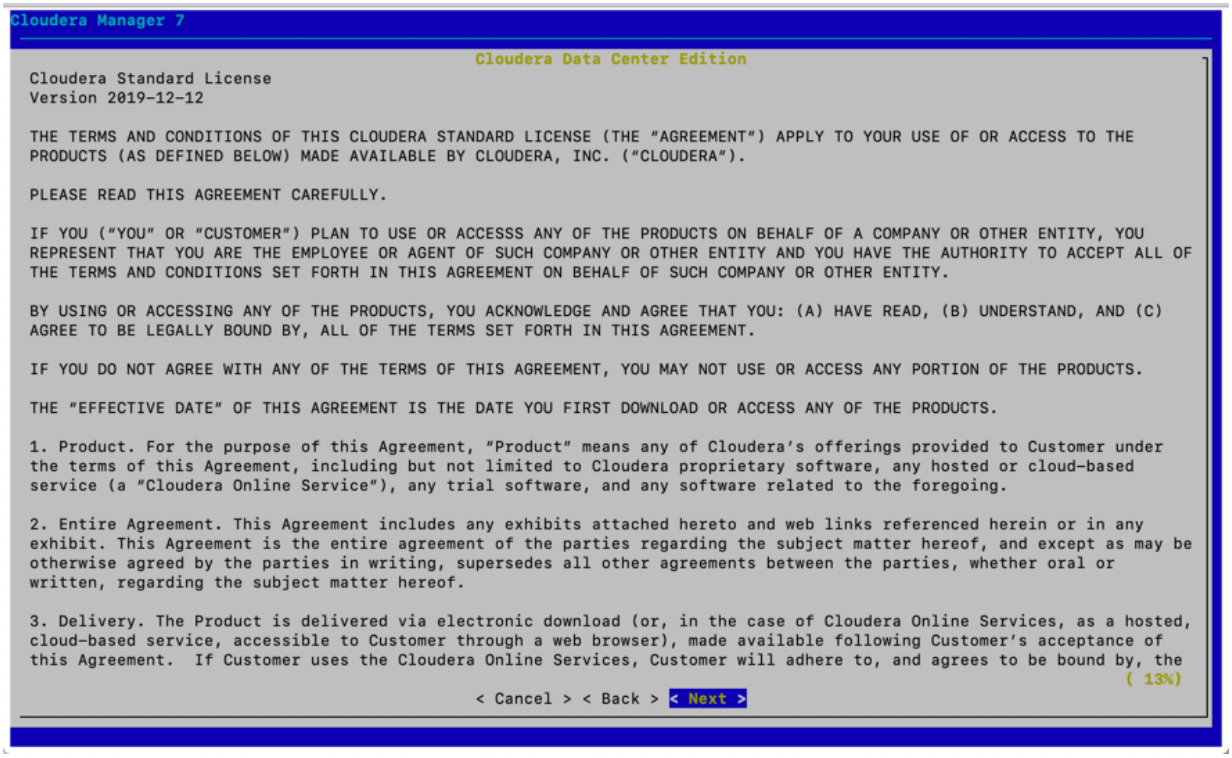
   c) For clusters without Internet access: Install Cloudera Manager packages from a local repository:

      ```
      sudo ./cloudera-manager-installer.bin --skip_repo_package=1
      ```

   The Cloudera Manager **Read Me** page appears.



2. Click Next.
   The Cloudera Standard License page appears.



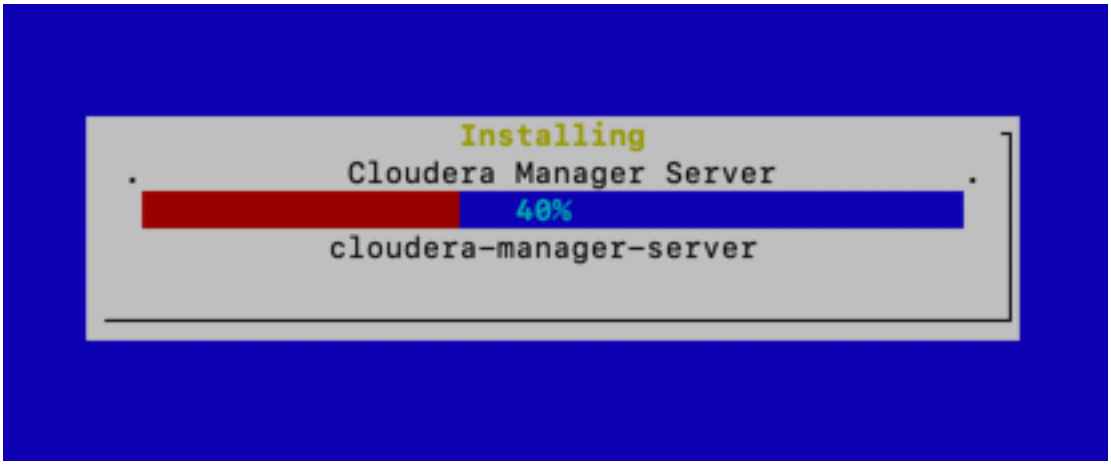
3. Click Next to accept the license agreement.

   The the installer starts and does the following:

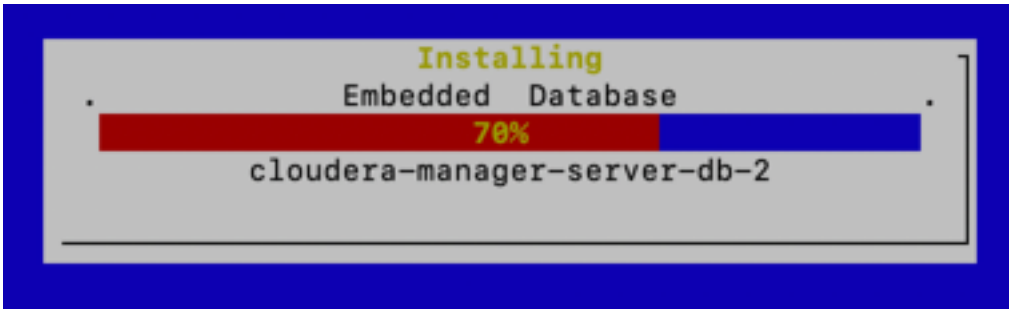
   a. Installs Oracle JDK.



   b. Installs the Cloudera Manager Server.



   c. Installs the embedded PostgreSQL packages and starts the database and Cloudera Manager Server.



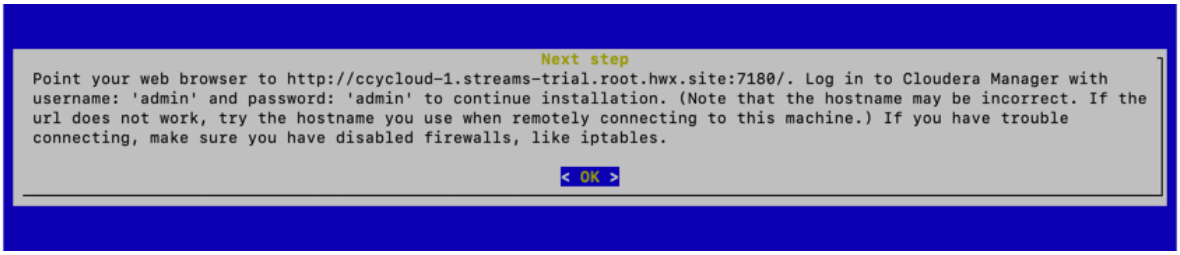
Note: If the installation is interrupted, run the following command on the Cloudera Manager Server host before you retry the installation:

```
sudo /usr/share/cmf/uninstall-cloudera-manager.sh
```

The log files for the installer are stored in /var/log/cloudera-manager-installer/.

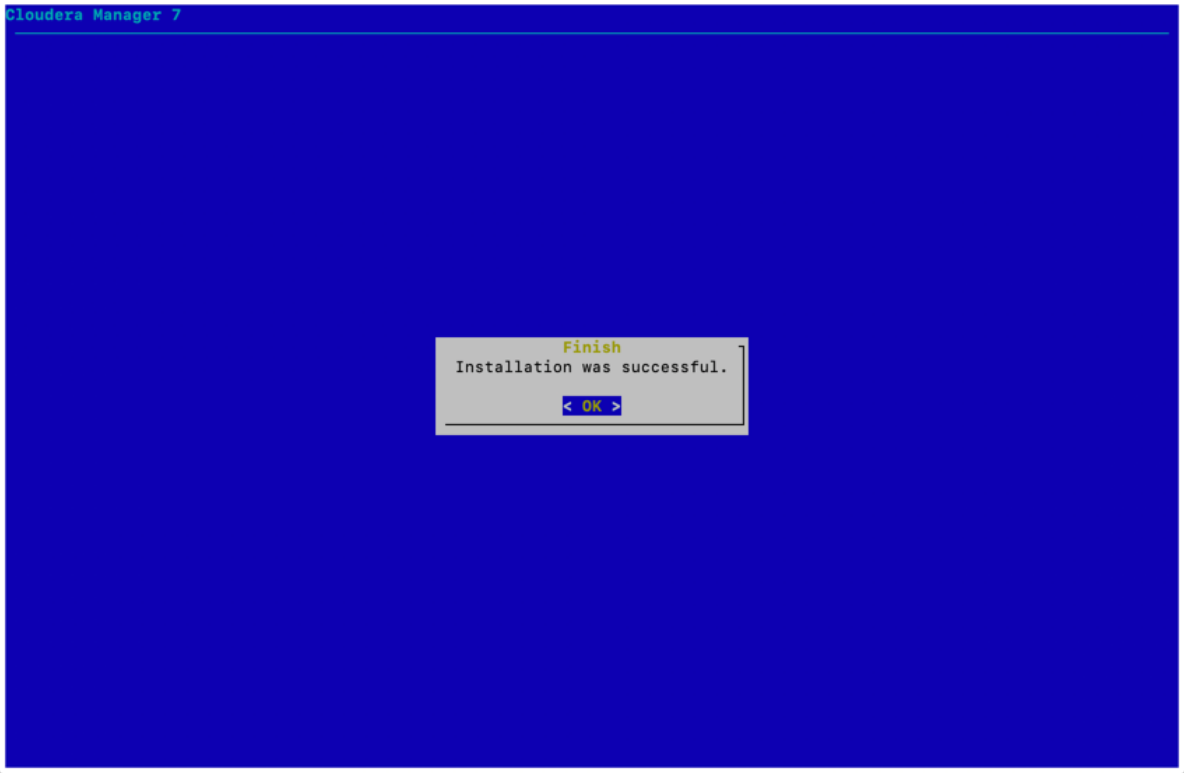
4. Exit the installer:
   a) When the installation completes, the complete URL for the Cloudera Admin Console displays, including the default port number: 7180.

Important: Make a note of this URL or take a screen capture as you will need it for the next task.



   b) Click OK.

      The success message appears.
   c) Click OK to exit the installer.



d) Wait a few minutes for the Cloudera Manager Server to start. To observe the startup process, run sudo tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log on the Cloudera Manager Server host. When you see the following log entry, the Cloudera Admin Console is ready:

```
INFO WebServerImpl:com.cloudera.server.cmf.WebServerImpl: Started Jetty
server.
```

What to do next

Install Cloudera Runtime

Install Cloudera Runtime

After you have installed Cloudera Manager, log in to Cloudera Manager to access the **Add Cluster - Installation** wizard. Here you will add hosts to form a cluster and install Cloudera Runtime and Cloudera Manager Agent software.

Before you begin

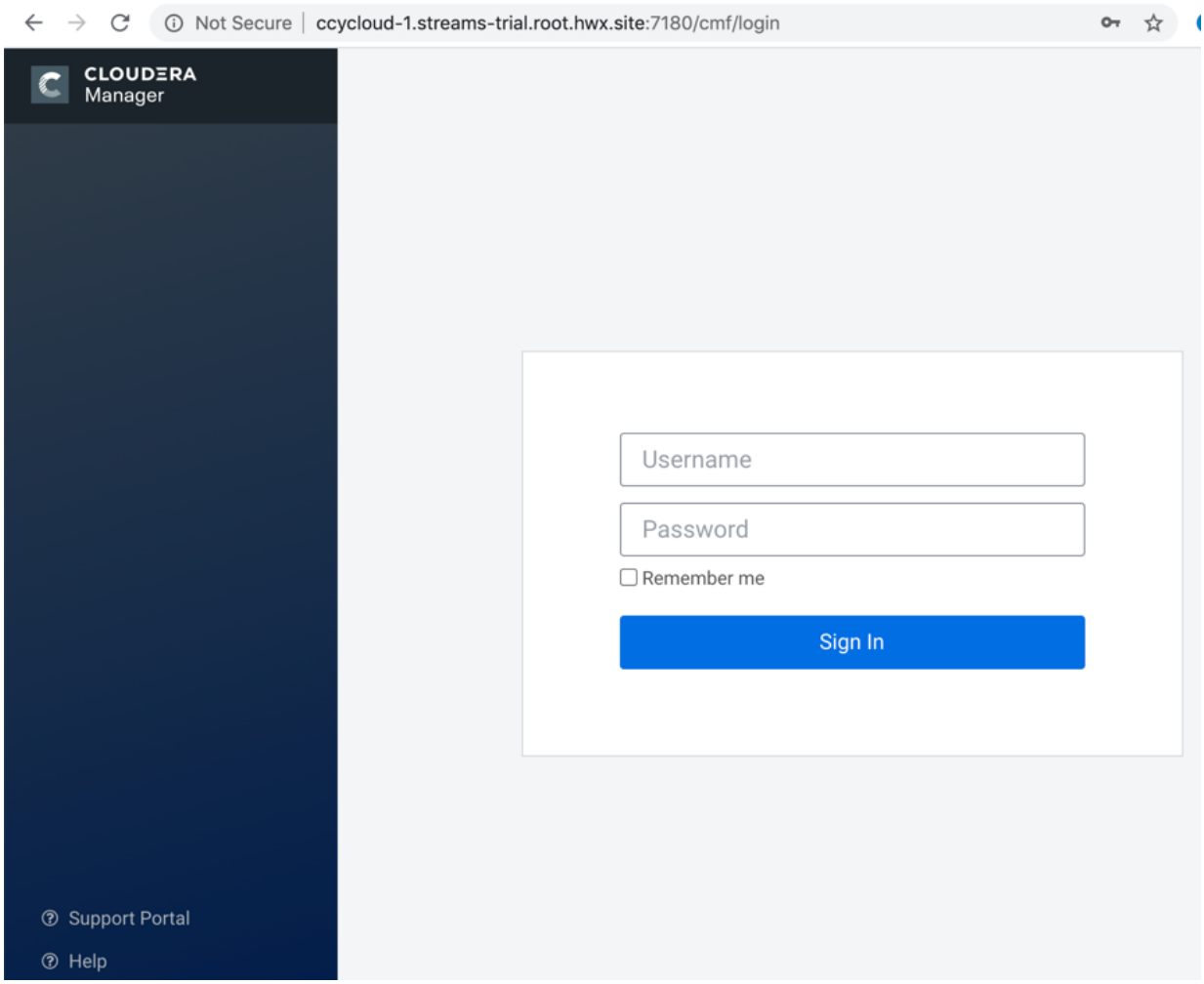
- You have installed Cloudera Manager.

Procedure

1. In a web browser, enter the URL that the Cloudera Manager Installer displayed in the previous task: http ://<*SERVER_HOST*>:7180, where <*SERVER_HOST*> is the FQDN or IP address of the host where the Cloudera Manager Server is running.

   For example: http://ccycloud-1.streams-trial.root.hwx.site:7180

   The **Cloudera Manager Sign In** page appears.



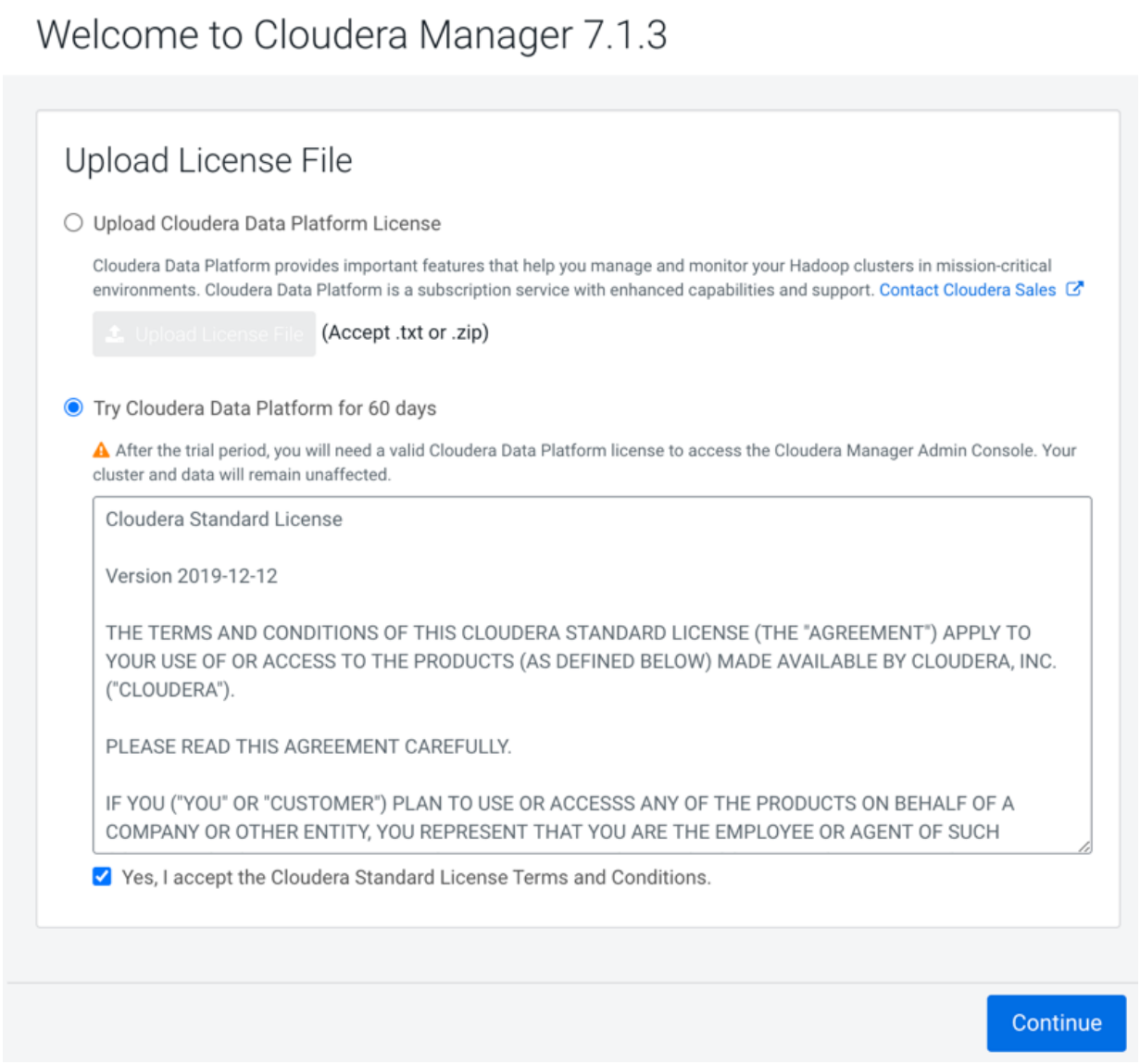
2. Sign in with the default credentials:
   - Username: admin
   - Password: admin

   Click Sign In.



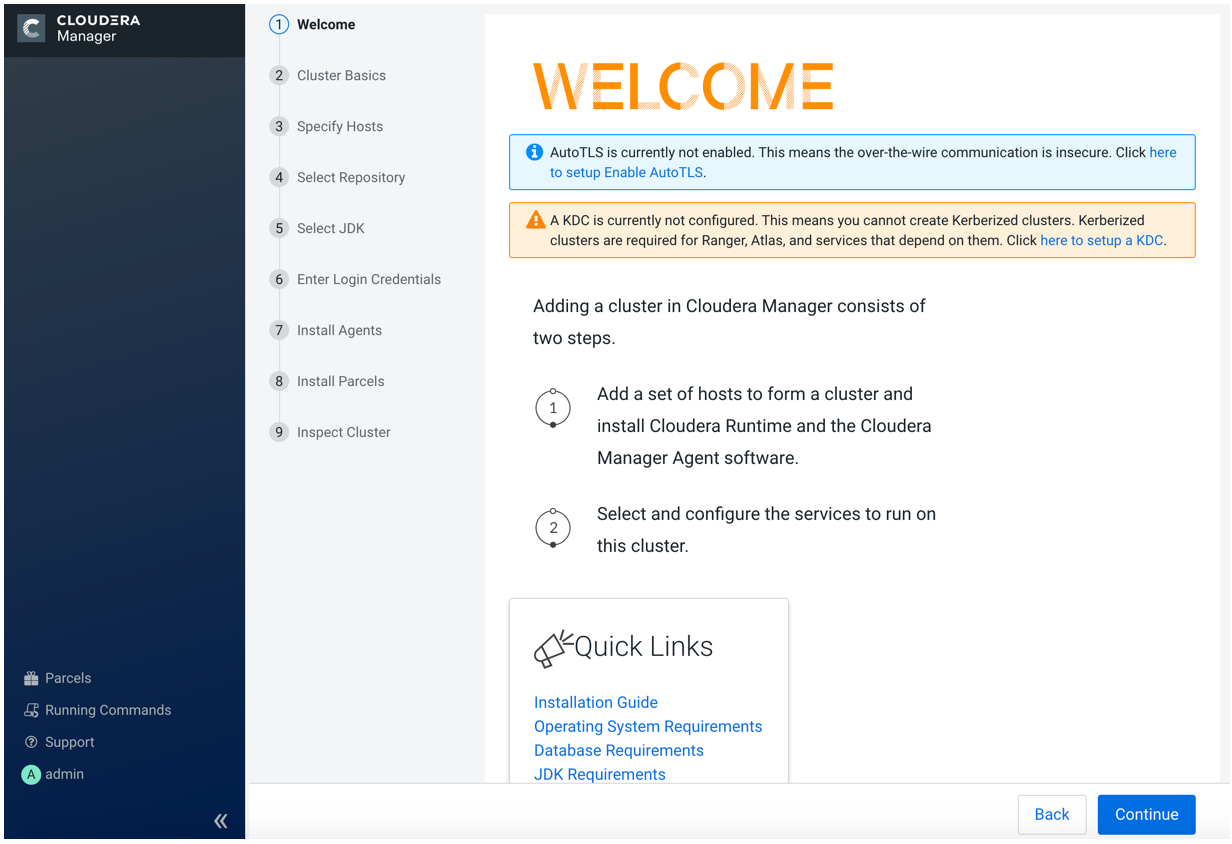
The **Welcome to Cloudera Manager** page appears.

3. Select:
   - Try Cloudera for 60 days
   - Yes, I accept the Cloudera Data Platform Standard License Terms and Conditions

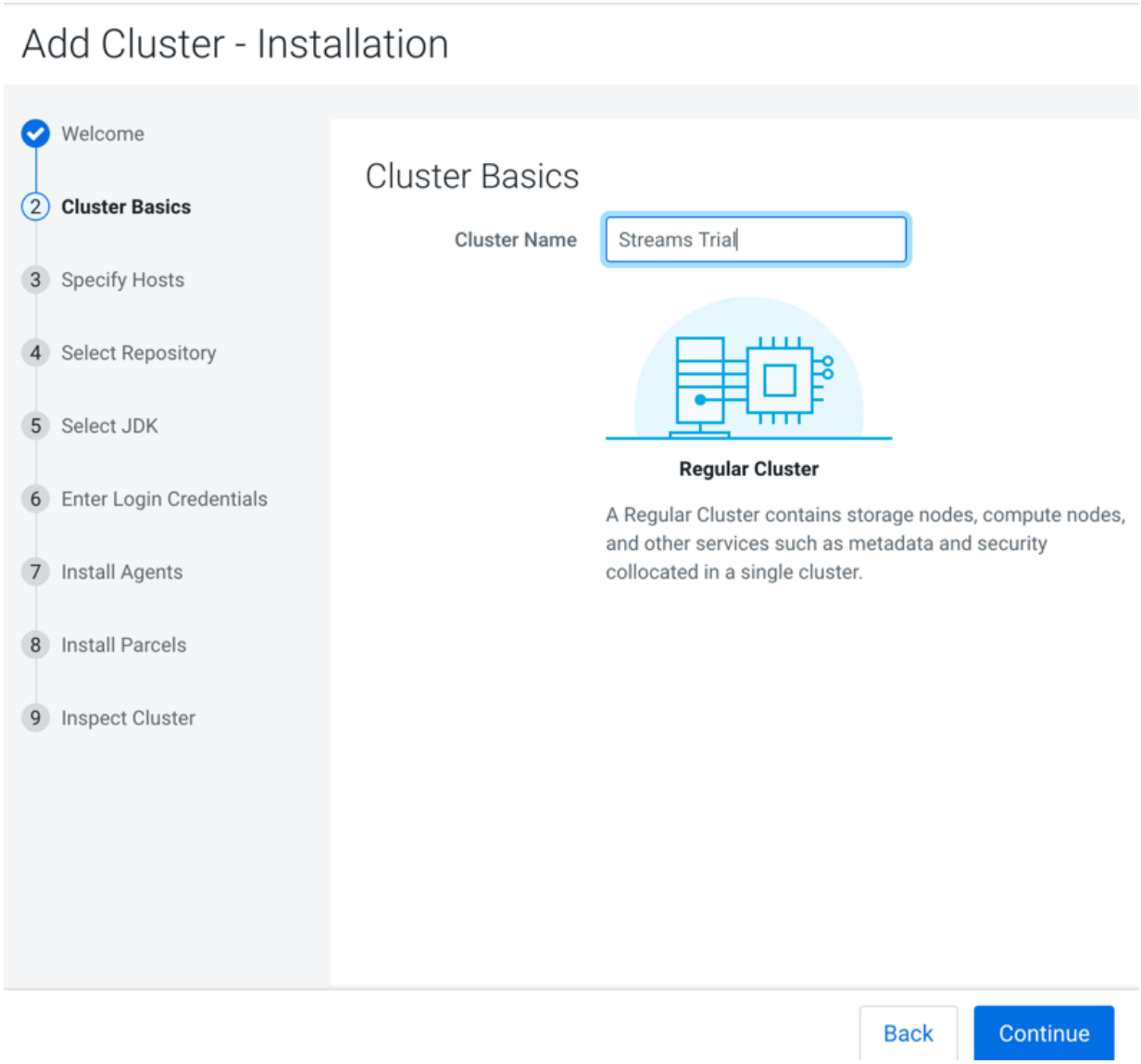


4. Click Continue.

   The **Add Cluster - Installation** page, **Welcome** section appears. The steps on the left let you know where you are in the workflow.

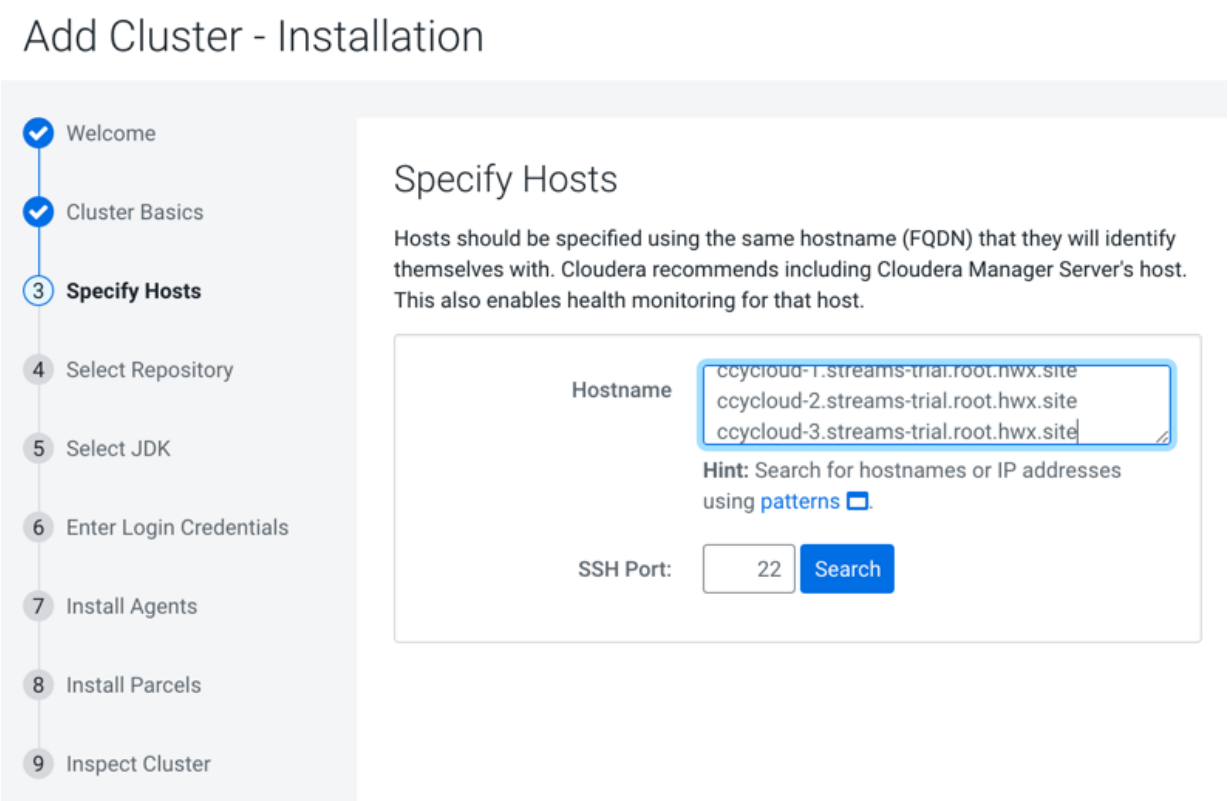


5. Click Continue.
   The Cluster Basics section appears.
6. Enter a name for the cluster and click Continue.



The **Specify Hosts** section appears.

7. Enter the cluster host names or IP addresses in the Hostnames field.

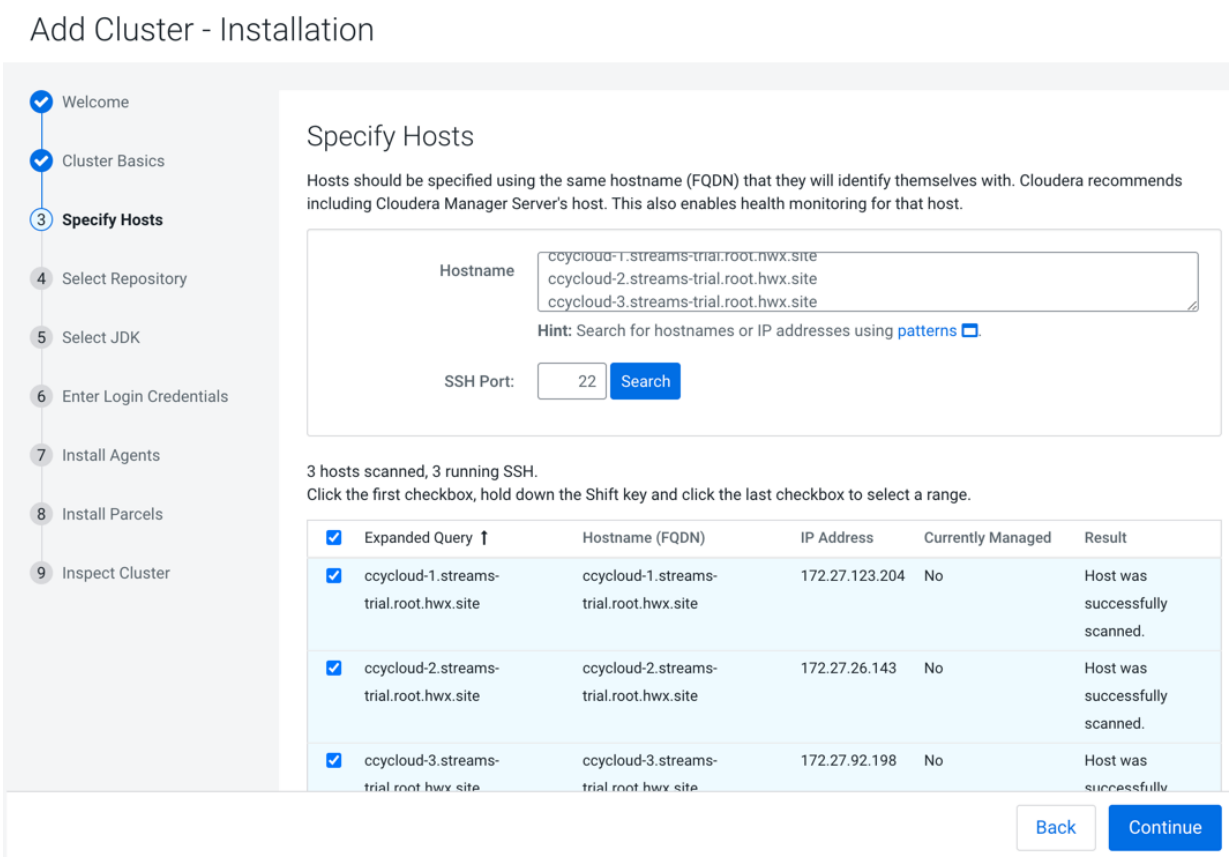


You can specify host name and IP address ranges as follows:

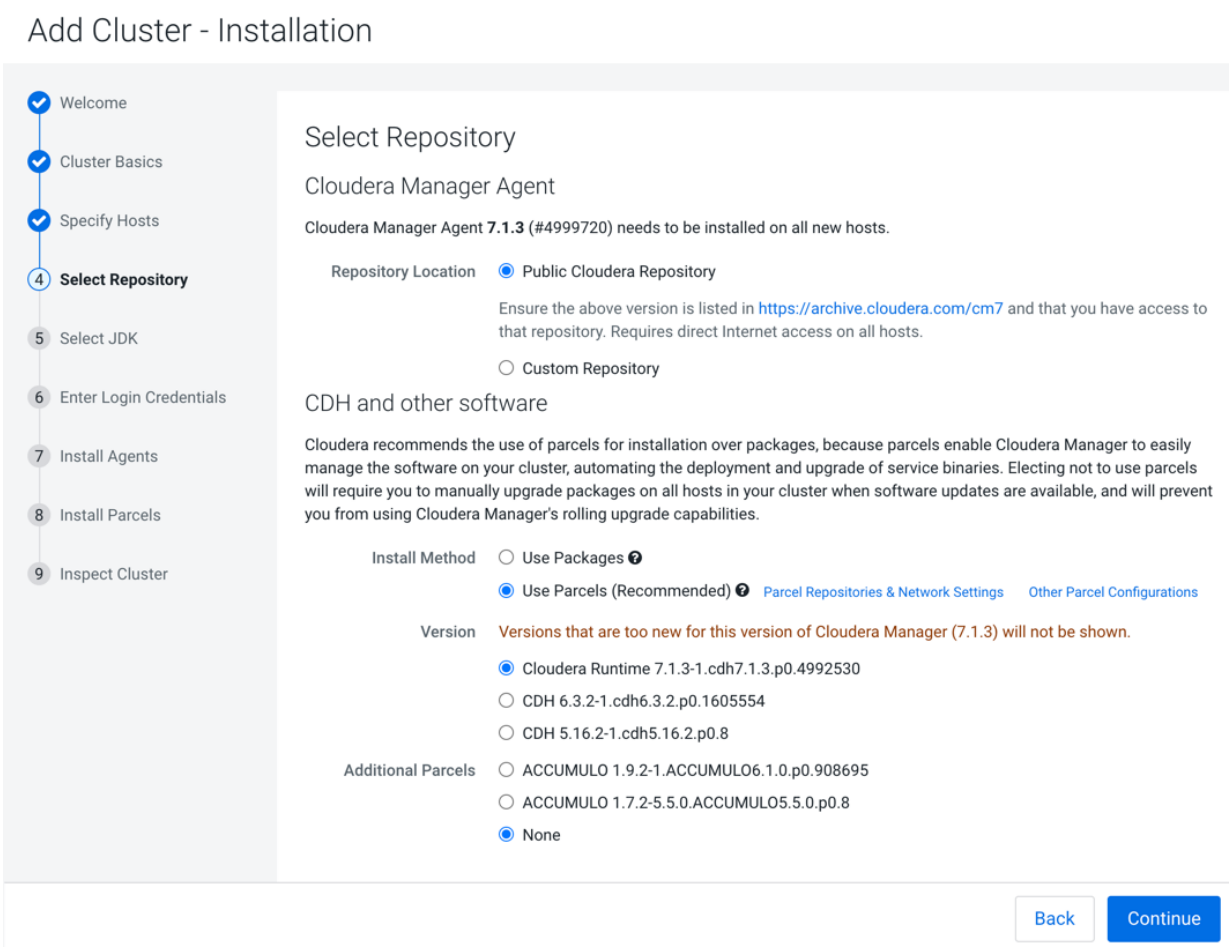
Expansion Range	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].example.com	host1.example.com, host2.example.com, host3.example.com
host[07-10].example.com	host07.example.com, host08.example.com, host09.example.com, host10.example.com

8. Click Search.

   Cloudera Manager discovers the hosts.



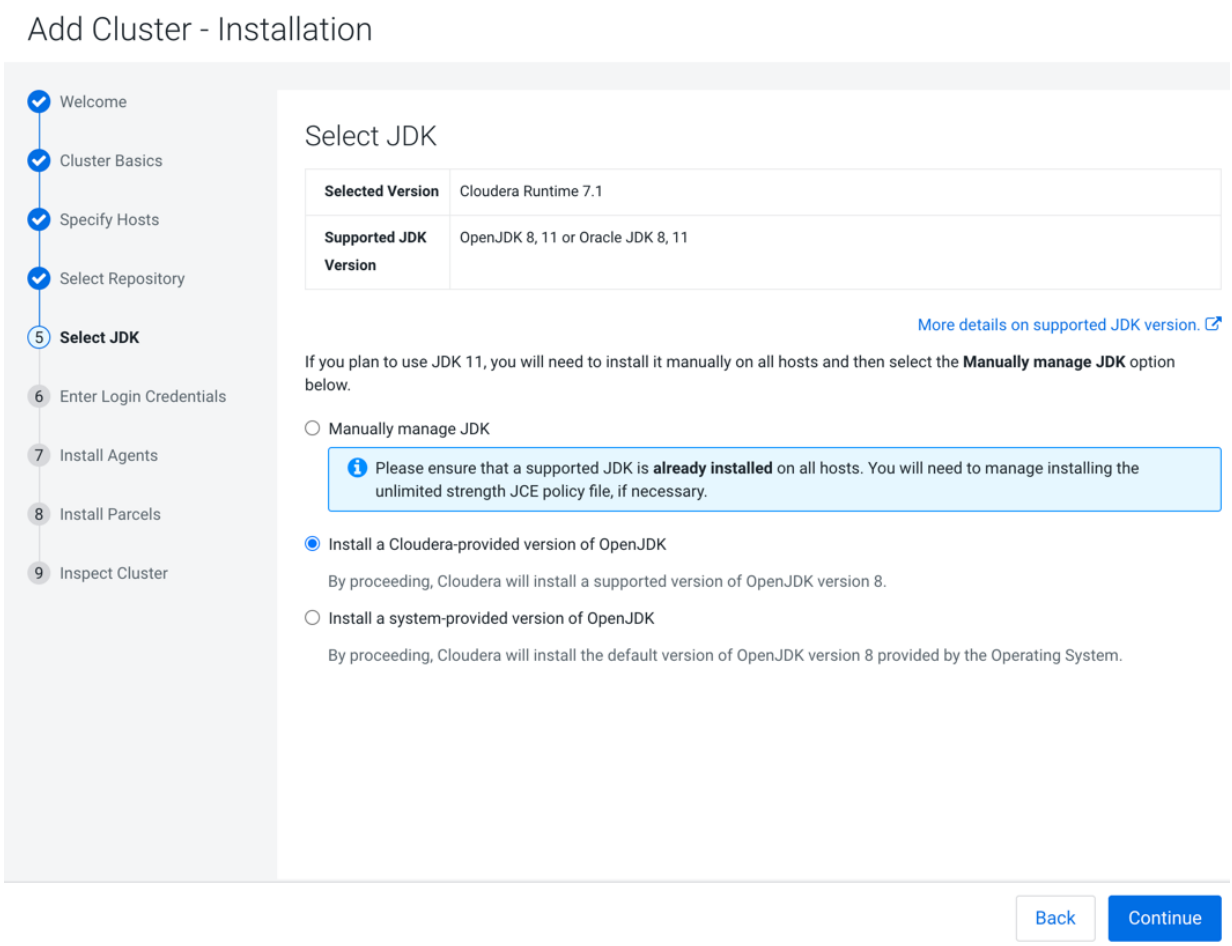
9. Verify host entries, deselect any that you do not want to install services on, and click Continue.
   The **Select Repository** section appears.
10. Select the following options:
    - Public Cloudera Repository
    - Use Parcels
    - The version of Cloudera Runtime that you want to install.
    - In the Additional Parcels section, None.



11. Click Continue.

The **Select JDK** section appears.

12. Select Install a Cloudera-provided version of OpenJDK.

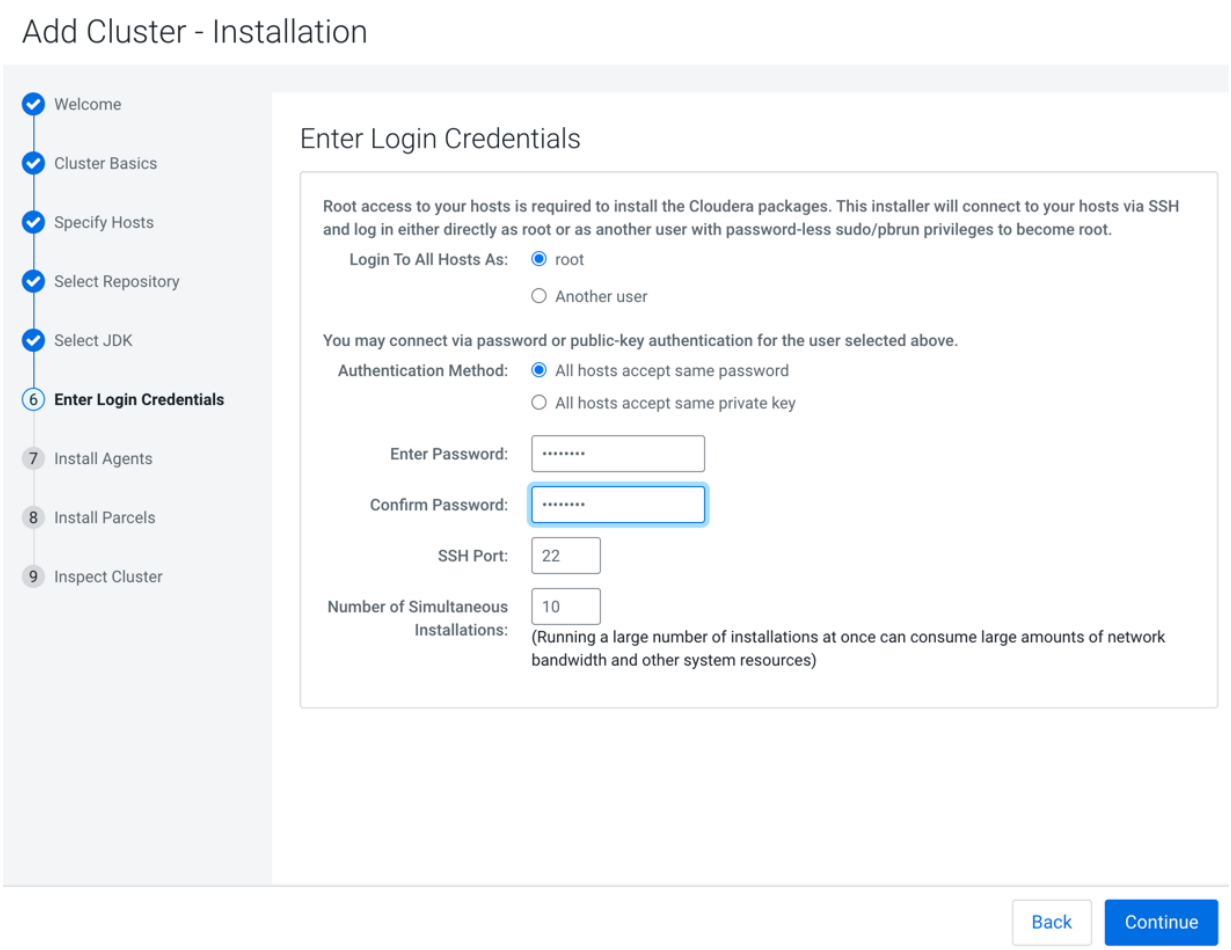


13. Click **Continue**.

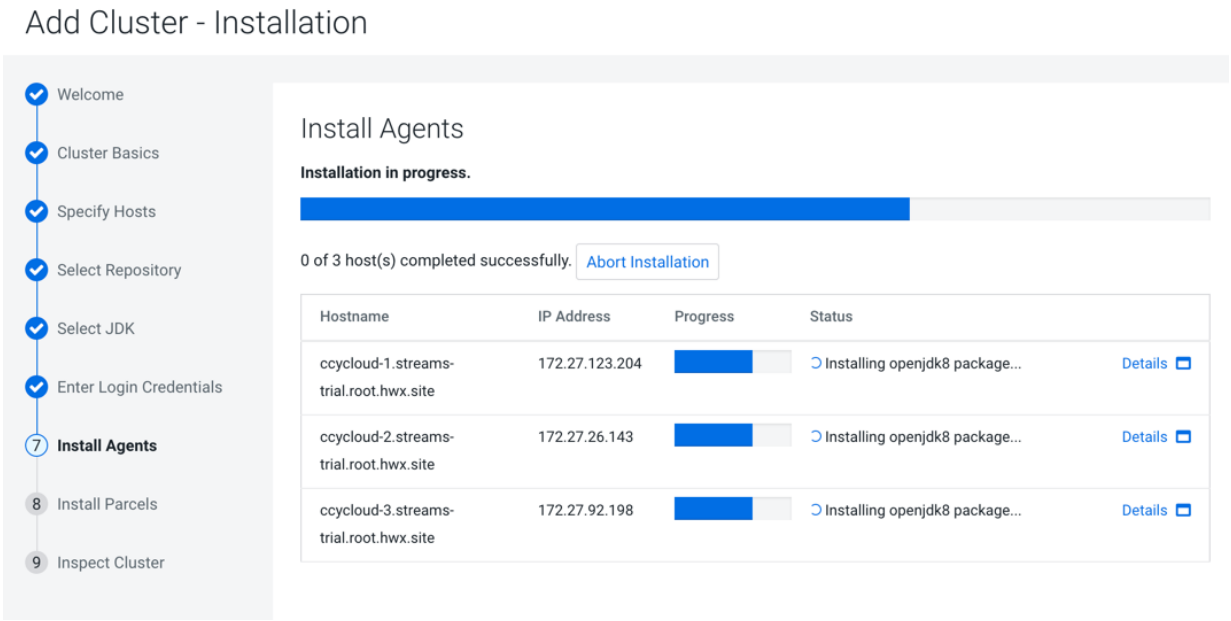
    The **Enter Login Credentials** section appears.

14. Do the following:

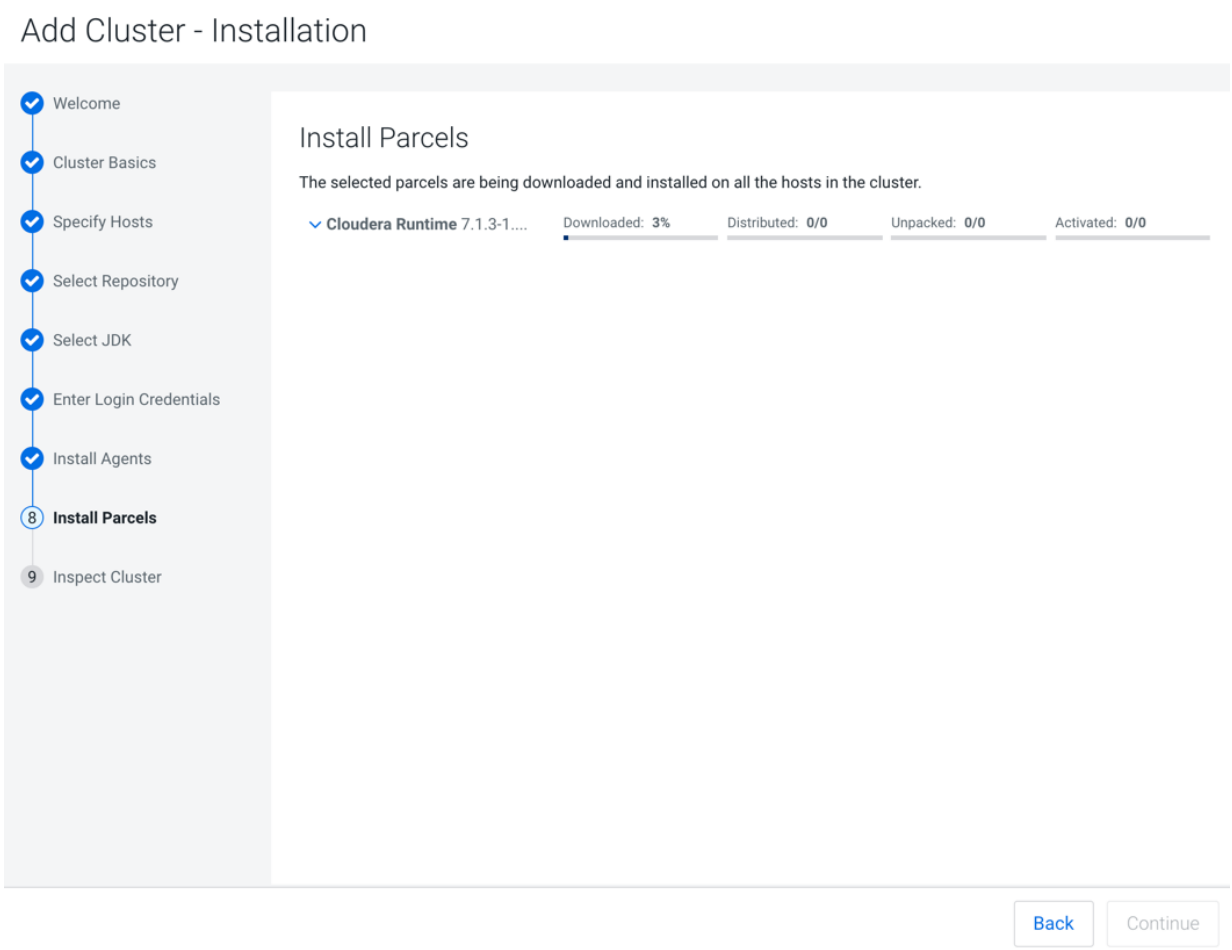
    - Select root.
    - Select All hosts accept same password.
    - Enter the password for the account that allows root access to your hosts.
    - Click Continue.



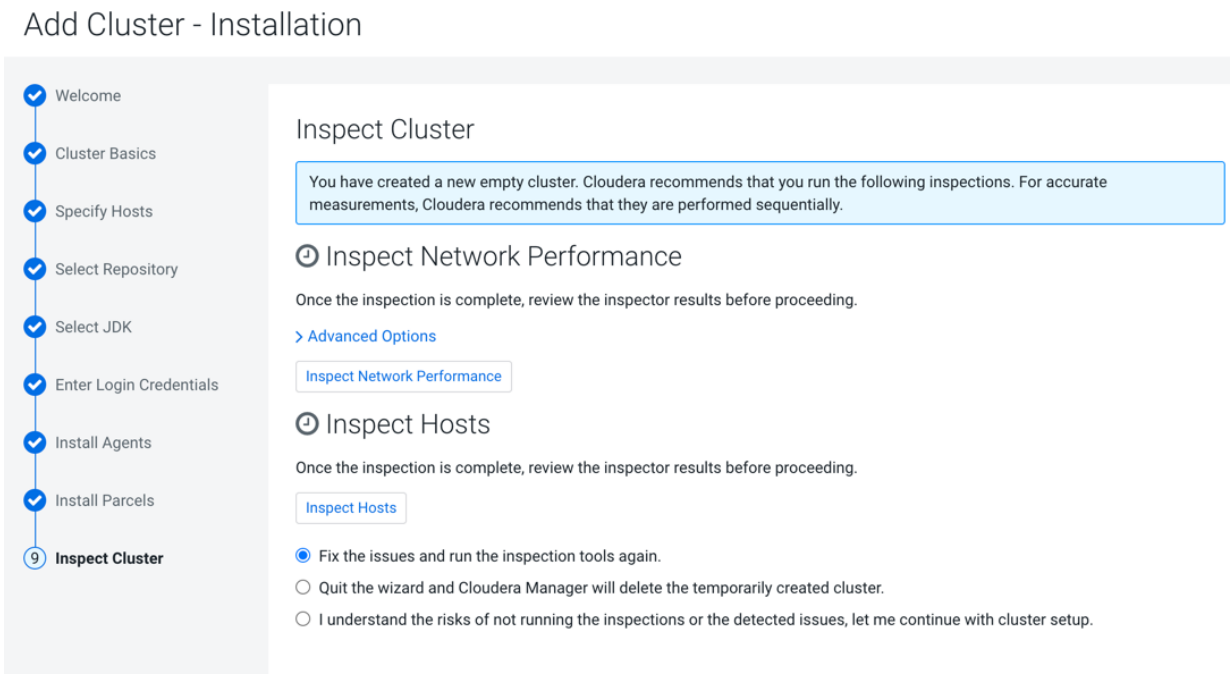
The **Install Agents** section appears showing the progress of the installation.



After the agents are installed, the **Install Parcels** section appears showing the progress of the parcel installation.



After the parcels are installed the **Inspect Cluster** section appears.

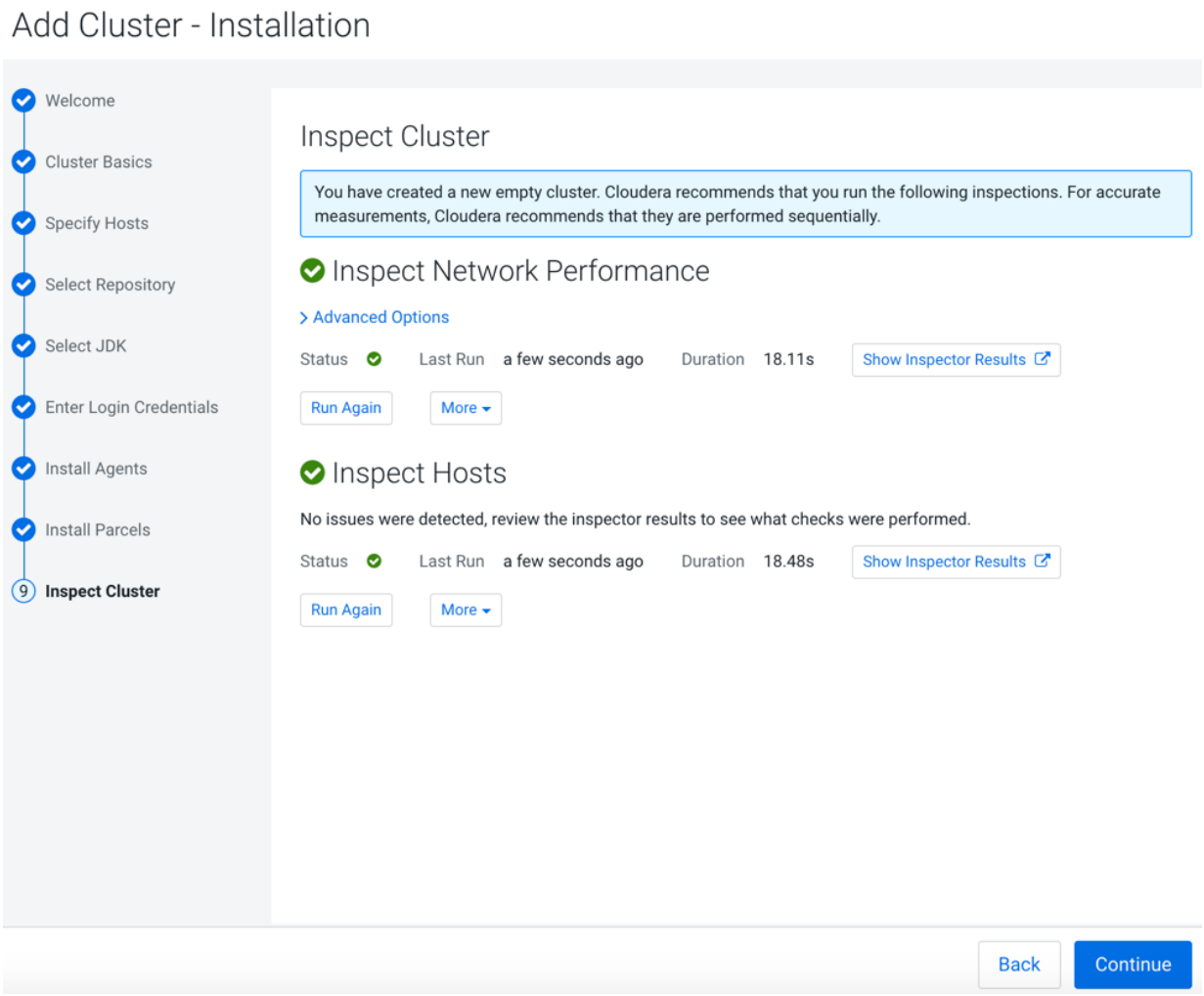


15. Do the following:

a) Select Inspect Network Performance.

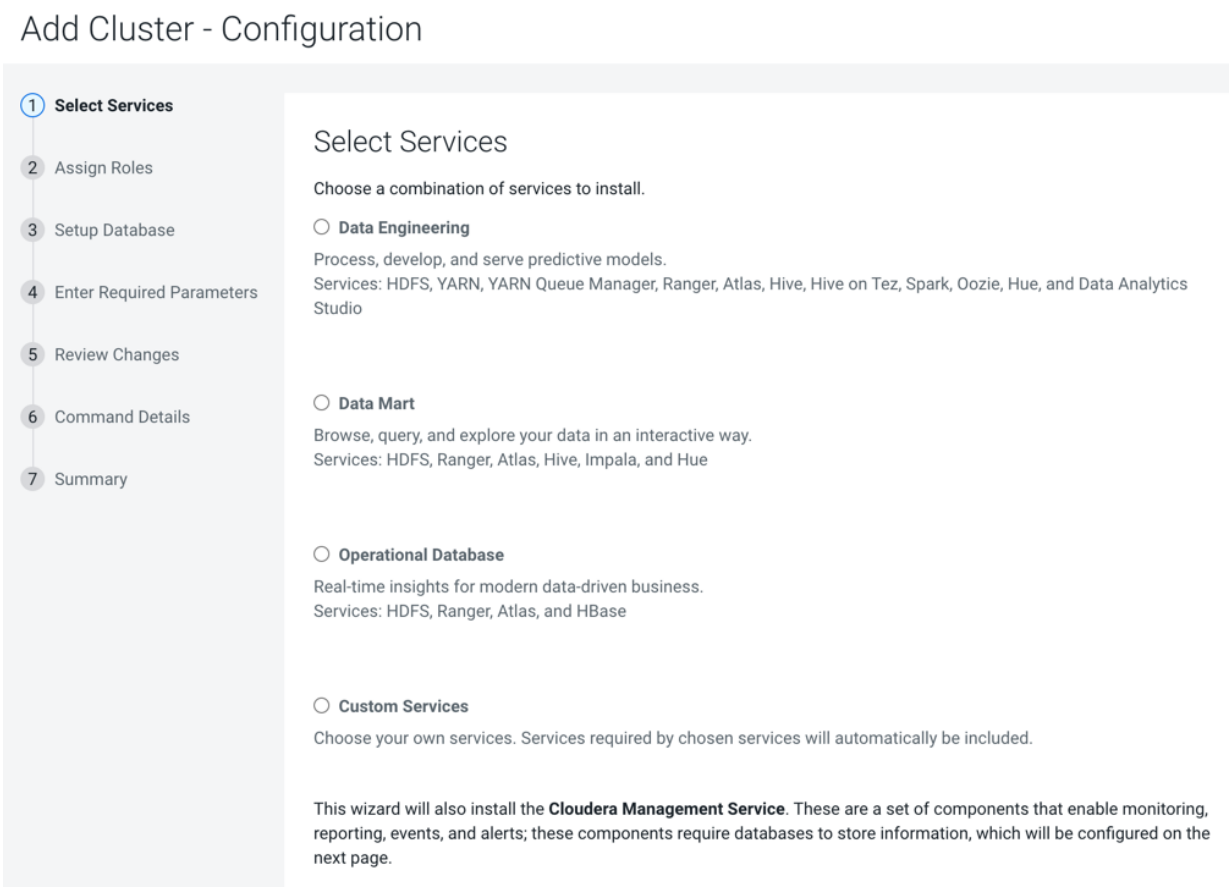
You can click Advanced Options to customize some ping parameters.

b) After the network inspector completes, click Show Inspector Results to view the results in a new tab. Address any reported issues, and click Run Again.
c) Click Inspect Hosts.
d) After the host inspector completes, click Show Inspector Results to view the results in a new tab. Address any reported issues, and click Run Again.



16. Click Continue.

The **Add Cluster - Configuration** page appears.



Results

This completes the **Add Cluster - Installation** wizard.

What to do next

Set up a cluster.

Set Up a Cluster Using the Wizard

After completing the Cluster Installation wizard, the Cluster Setup wizard automatically starts. The following sections guide you through each page of the wizard.

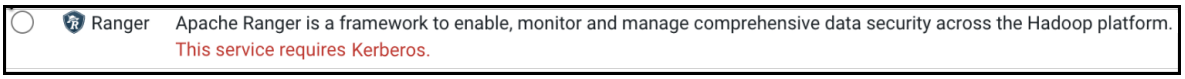
Select Services

The Select Services page allows you to select the services you want to install and configure. Make sure that you have the appropriate license key for the services you want to use.

Important:

In case you deploy a scratch cluster that is not yet kerberised, you cannot select any service that requires Kerberos as a prerequisite including Ranger, RangerKMS, Atlas, and Knox.

Each of these services carries such a note near it (see example for Ranger):



Selecting either of these services to be installed without having a Kerberos based cluster leads to a non-functioning cluster. It is advised to install these services separately only after enabling Kerberos on the cluster. For more information, see Set up a KDC.

Note that if you do not add Kerberos to the cluster at the beginning of the wizard (for example, setting Cloudera Manager as Key Distribution Center), you need to selct only Customer Services option and pick the non-kerberos services you want to deploy. Post enabling Kerberos, you need to repeat the wizard and complete the Kerberos based services.

You can choose from:

Regular (Base) Clusters

Cloudera Data Engineering

Process develop, and serve predictive models.

Services included: HDFS, YARN, YARN Queue Manager, Ranger, Atlas, Hive, Hive on Tez, Spark, Oozie, and Hue

Data Mart

Browse, query, and explore your data in an interactive way.

Services included: HDFS, Ranger, Atlas, Hive, and Hue

Cloudera Operational Database

Real-time insights for modern data-driven business.

Services included: HDFS, Ranger, Atlas, and HBase

Custom Services

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

After selecting the services you want to add, click Continue. The Assign Roles page displays.

Assign Roles

The Assign Roles page suggests role assignments for the hosts in your cluster. You can click on the hostname for a role to select a different host. You can also click the View By Host button to see all the roles assigned to a host.

To review the recommended role assignments, see *Recommended Cluster Hosts and Role Distribution*.

After assigning all of the roles for your services, click Continue. The Setup Database page displays.

Setup Database

When using the Cloudera Manager installer with the embedded database, the Setup Database page is pre-populated with the database names and passwords. Click Test Connection to validate the settings. If the connection is successful, a green checkmark and the word Successful appears next to each service. If there are any problems, the error is reported next to the service that failed to connect. Some databases will be created in a future step. For these, the words Skipped. Cloudera Manager will create this database in a later step. appear next to the green checkmark.

After verifying that each connection is successful, click Continue. The Review Changes page displays.

Enter Required Parameters

The **Enter Required Parameters** page lists required parameters for the Cloudera Manager API client, and Ranger.

If you do not have an existing user for the Cloudera Manager API client, use the default username and password "admin" for both the The Existing Cloudera Manager API Client Username and The Existing Cloudera Manager API Client Password.

The Ranger Admin user, Usersync user, Tagsync User, and KMS Keyadmin User are created during cluster deployment. In this page you must give a password for each of these users.



Note: Passwords for the Ranger Admin, Usersync, Tagsync, and KMS Keyadmin users must be a minimum of 8 characters long, with at least one alphabetic and one numeric character. The following characters are not valid: " ' \ ` ´.

The Ranger database host, name, user, and user password were configured when you created the required Ranger database. If you ran the gen_embedded_ranger_db.sh script to create the Ranger database, the output of the script contained the host and database user password. Enter those here. The default database name is "ranger" and the default database user is "rangeradmin."

Review Changes

The Review Changes page lists default and suggested settings for several configuration parameters, including data directories.



Warning: Do not place DataNode data directories on NAS devices. When resizing an NAS, block replicas can be deleted, which results in missing blocks.

Review and make any necessary changes, and then click Continue. The Command Details page displays.

Command Details

The Command Details page lists the details of the First Run command. You can expand the running commands to view the details of any step, including log files and command output. You can filter the view by selecting Show All Steps, Show Only Failed Steps, or Show Running Steps.

After the First Run command completes, click Continue to go to the Summary page.

Summary

The Summary page reports the success or failure of the setup wizard. Click Finish to complete the wizard. The installation is complete.

Cloudera recommends that you change the default password as soon as possible by clicking the logged-in username at the top right of the home screen and clicking Change Password.

Stopping the Embedded PostgreSQL Database

To stop the embedded PostgreSQL database, stop the services and servers in the order listed below.

Procedure

1. Log in to the Cloudera Manager user interface and stop the services that have a dependency on the Hive metastore (Hue, Impala, and Hive) in the following order:
   - Stop the Hue and Impala services.
   - Stop the Hive service.
2. On the Cloudera Manager **Home** page, click the 3 vertical dots next to Cloudera Management Service and select Stop to stop the Cloudera Management Service.
3. Stop the Cloudera Manager Server.

RHEL:

```
sudo systemctl stop cloudera-scm-server.service
```

4. Stop the Cloudera Manager Server database.
   RHEL:

```
sudo systemctl stop cloudera-scm-server-db.service
```

Starting the Embedded PostgreSQL Database

To start the embedded PostgreSQL database, start the servers and services in the order listed below.

Procedure

1. Start the Cloudera Manager Server database.
   RHEL:

```
sudo systemctl start cloudera-scm-server-db.service
```

2. Start the Cloudera Manager Server.
   RHEL:

```
sudo systemctl start cloudera-scm-server.service
```

3. Log into Cloudera Manager and start the Cloudera Manager Service. On the Cloudera Manager **Home** page, click the 3 vertical dots next to Cloudera Management Service and select Start.
4. In the Cloudera Manager user interface, start the services that have a dependency on the Hive metastore (Hue, Impala, and Hive) in the following order:
   - Start the Hive service.
   - Start the Hue and Impala services.

Changing Embedded PostgreSQL Database Passwords

The embedded PostgreSQL database has generated user accounts and passwords. You can change a password associated PostgreSQL database account.

About this task

You can see the generated accounts and passwords during the installation process and you should record them at that time.

To find information about the PostgreSQL database account that the Cloudera Manager Server uses, read the /etc/cloudera-scm-server/db.properties file:

```
# cat /etc/cloudera-scm-server/db.properties

Auto-generated by scm_prepare_database.sh
#
Sat Oct 1 12:19:15 PDT 201
#
com.cloudera.cmf.db.type=postgresql
com.cloudera.cmf.db.host=localhost:7432
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=TXqEESuhj5
```

To change a password associated with a PostgreSQL database account:

Procedure

1. Obtain the root password from the /var/lib/cloudera-scm-server-db/data/generated_password.txt file:

```
# cat /var/lib/cloudera-scm-server-db/data/generated_password.txt

MnPwGeWaip

The password above was generated by /usr/share/cmf/bin/initialize_embedded
_db.sh (part of the cloudera-scm-server-db package)
and is the password for the user 'cloudera-scm' for the database in the
current directory.

Generated at Fri Jun 29 16:25:43 PDT 2012.
```

2. On the host on which the Cloudera Manager Server is running, log into PostgreSQL as the root user:

```
psql -U cloudera-scm -p 7432 -h localhost -d postgres
Password for user cloudera-scm: MnPwGeWaip
psql (8.4.18)
Type "help" for help.

postgres=#
```

3. Determine the database and owner names:

```
postgres=# \l
                                     List of databases
   Name    |    Owner     | Encoding | Collation  |   Ctype    |
 Access privileges
-----------+--------------+----------+------------+------------+-------
-----------------------------
 amon      | amon         | UTF8     | en_US.UTF8 | en_US.UTF8 |
 hive      | hive         | UTF8     | en_US.UTF8 | en_US.UTF8 |
 nav       | nav          | UTF8     | en_US.UTF8 | en_US.UTF8 |
 navms     | navms        | UTF8     | en_US.UTF8 | en_US.UTF8 |
 postgres  | cloudera-scm | UTF8     | en_US.UTF8 | en_US.UTF8 |
 rman      | rman         | UTF8     | en_US.UTF8 | en_US.UTF8 |
 scm       | scm          | UTF8     | en_US.UTF8 | en_US.UTF8 |
 template0 | cloudera-scm | UTF8     | en_US.UTF8 | en_US.UTF8 | =c/"clou
dera-scm"
                                                                 : "cloudera
-scm"=CTc/"cloudera-scm"
 template1 | cloudera-scm | UTF8     | en_US.UTF8 | en_US.UTF8 | =c/"cl
oudera-scm"
                                                                 : "cloude
ra-scm"=CTc/"cloudera-scm"
(9 rows)
```

4. Set the password for an owner using the \password command. For example, to set the password for the amon owner, do the following:

```
postgres=# \password amon
Enter new password:
Enter it again:
```

5. Configure the role with the new password:
   a) In the Cloudera Admin Console, select Clusters Cloudera Management Service .
   b) Click the Configuration tab.

c) In the Scope section, select the role where you are configuring the database.
d) Select CategoryDatabase category.
e) Set the *ROLE NAME* Database Password property.
f) Enter a Reason for change, and then click Save Changes to commit the changes.

Migrating from the Cloudera Manager Embedded PostgreSQL Database Server to an External PostgreSQL Database

If you have already used the embedded PostgreSQL database and you are unable to redeploy a fresh cluster, you must migrate the embedded PostgreSQL database sever to an external PostgreSQL database.

Cloudera Manager provides an embedded PostgreSQL database server for trial and proof of concept deployments when creating a cluster. To remind users that this embedded database is not suitable for production, Cloudera Manager displays the banner text: "You are running Cloudera Manager in non-production mode, which uses an embedded PostgreSQL database. Switch to using a supported external database before moving into production."

If, however, you have already used the embedded database, and you are unable to redeploy a fresh cluster, then you must migrate to an external PostgreSQL database.



Note: This procedure does not describe how to migrate to a database server other than PostgreSQL. Moving databases from one database server to a different type of database server is a complex process that requires modification of the schema and matching the data in the database tables to the new schema. It is recommended that you engage with Cloudera Professional Services if you wish to perform a migration to an external database server other than PostgreSQL.

Prerequisites

Before migrating the Cloudera Manager embedded PostgreSQL database to an external PostgreSQL database, ensure that your setup meets the following conditions:

- The external PostgreSQL database server is running.
- The database server is configured to accept remote connections.
- The database server is configured to accept user logins using md5.
- No one has manually created any databases in the external database server for roles that will be migrated.



Note: To view a list of databases in the external database server (requires default superuser permission):

```
sudo -u postgres psql -l
```

- All health issues with your cluster have been resolved.

For details about configuring the database server, see *Configuring and Starting the PostgreSQL Server*.



Important: Only perform the steps in *Configuring and Starting the PostgreSQL Server*. Do not proceed with the creation of databases as described in the subsequent section.

For large clusters, Cloudera recommends running your database server on a dedicated host. Engage Cloudera Professional Services or a certified database administrator to correctly tune your external database server.

Identify Roles that Use the Embedded Database Server

Before you can migrate to another database server, you must first identify the databases using the embedded database server.

About this task

When the Cloudera Manager Embedded Database server is initialized, it creates the Cloudera Manager database and databases for roles in the Management Services. The Installation Wizard (which runs automatically the first time you

log in to Cloudera Manager) or Add Service action for a cluster creates additional databases for roles when run. It is in this context that you identify which roles are used in the embedded database server.

To identify which roles are using the Cloudera Manager embedded database server:

Procedure

1. Obtain and save the cloudera-scm superuser password from the embedded database server. You will need this password in subsequent steps:

   ```
   head -1  /var/lib/cloudera-scm-server-db/data/generated_password.txt
   ```

2. Make a list of all services that are using the embedded database server. Then, after determining which services are not using the embedded database server, remove those services from the list. The scm database must remain in your list. Use the following table as a guide:

   Table 11: Cloudera Manager Embedded Database Server Databases

Service	Role	Default Database Name	Default Username
Cloudera Manager Server		scm	scm
Cloudera Management Service	Activity Monitor	amon	amon
Hive	Hive Metastore Server	hive	hive
Hue	Hue Server	hue	7uu7uu7uhue
Oozie	Oozie Server	oozie_oozie_server	oozie_oozie_server
Cloudera Management Service	Reports Manager	rman	rman
Ranger		ranger	rangeradmin

3. Verify which roles are using the embedded database. Roles using the embedded database server always use port 7432 (the default port for the embedded database) on the Cloudera Manager Server host.

   a. Verify which roles are using the embedded database. Roles using the embedded database server always use port 7432 (the default port for the embedded database) on the Cloudera Manager Server host.

      For Cloudera Management Services:

      1. Select Cloudera Management Service > Configuration, and type "7432" in the Search field.
      2. Confirm that the hostname for the services being used is the same hostname used by the Cloudera Manager Server.

      Note:

      If any of the following fields contain the value "7432", then the service is using the embedded database:

      - Activity Monitor
      - Reports Manager

      For the Oozie Service:

      1. Select Oozie service > Configuration, and type "7432" in the Search field.
      2. Confirm that the hostname is the Cloudera Manager Server.

      For Hive and Hue Services:

      1. Select the specific service > Configuration, and type "database host" in the Search field.
      2. Confirm that the hostname is the Cloudera Manager Server.
      3. In the Search field, type “database port” and confirm that the port is 7432.
      4. Repeat these steps for each of the services (Hive and Hue).

4. Verify the database names in the embedded database server match the database names on your list (Step 2). Databases that exist on the database server and not used by their roles do not need to be migrated. This step is to confirm that your list is correct.

   Note: Do not add the postgres, template0, or template1 databases to your list. These are used only by the PostgreSQL server.

```
psql -h localhost -p 7432 -U cloudera-scm -l
```

```
Password for user cloudera-scm: <password>

                                                List of databases
       Name        |       Owner        | Encoding |  Collate   |   Ctype
    |         Access
--------------------+--------------------+----------+------------+-------
-----+---------------
amon                | amon               | UTF8     | en_US.UTF8 | en_US.U
TF8 |
hive                | hive               | UTF8     | en_US.UTF8 | en_US.UT
F8 |
hue                 | hue                | UTF8     | en_US.UTF8 | en_US
.UTF8 |
navms               | navms              | UTF8     | en_US.UTF8 | en_US.
UTF8 |
oozie_oozie_server | oozie_oozie_server | UTF8     | en_US.UTF8 | en_US.U
TF8 |
postgres            | cloudera-scm       | UTF8     | en_US.UTF8 | en_US.UT
F8 |
rman                | rman               | UTF8     | en_US.UTF8 | en_US
.UTF8 |
scm                 | scm                | UTF8     | en_US.UTF8 | en_US.
UTF8 |
template0           | cloudera-scm       | UTF8     | en_US.UTF8 | en_US.U
TF8 | =c/"cloudera-scm"
template1           | cloudera-scm       | UTF8     | en_US.UTF8 | en_US.
UTF8 | =c/"cloudera-scm"
(12 rows)
```

Results

You should now have a list of all roles and database names that use the embedded database server, and are ready to proceed with the migration of databases from the embedded database server to the external PostgreSQL database server.

Migrate Databases from the Embedded Database Server to the External PostgreSQL Database Server

After you identify the roles that use the embedded database, you are ready to migrate from the embedded database server to an external PostgreSQL database server.

About this task

While performing this procedure, ensure that the Cloudera Manager Agents remain running on all hosts. Unless otherwise specified, when prompted for a password use the cloudera-scm password.

Note: After completing this migration, you cannot delete the cloudera-scm postgres superuser unless you remove the access privileges for the migrated databases. Minimally, you should change the cloudera-scm pos tgres superuser password.

Procedure

1. In Cloudera Manager, stop the cluster services identified as using the embedded database server. Be sure to stop the Cloudera Management Service as well. Also be sure to stop any services with dependencies on these services. The remaining Cloudera Runtime services will continue to run without downtime.

   Note: If you do not stop the services from within Cloudera Manager before stopping Cloudera Manager Server from the command line, they will continue to run and maintain a network connection to the embedded database server. If this occurs, then the embedded database server will ignore any command line stop commands (Step 2) and require that you manually stop the process, which in turn causes the services to crash instead of stopping cleanly.
2. Navigate to Hosts > All Hosts, and make note of the number of roles assigned to hosts. Also take note whether or not they are in a commissioned state. You will need this information later to validate that your scm database was migrated correctly.
3. Stop the Cloudera Manager Server. To stop the server:

   ```
   sudo service cloudera-scm-server stop
   ```
4. Obtain and save the embedded database superuser password (you will need this password in subsequent steps) from the generated_password.txt file:

   ```
   head -1  /var/lib/cloudera-scm-server-db/data/generated_password.txt
   ```
5. Export the PostgreSQL user roles from the embedded database server to ensure the correct users, permissions, and passwords are preserved for database access. Passwords are exported as an md5sum and are not visible in plain text. To export the database user roles (you will need the cloudera-scm user password):

   ```
   pg_dumpall -h localhost -p 7432 -U cloudera-scm -v --roles-only -f "/var/
   tmp/cloudera_user_roles.sql"
   ```
6. Edit the /var/tmp/cloudera_user_roles.sql file to remove any CREATE ROLE and ALTER ROLE commands for databases not in your list. Leave the entries for the cloudera-scm user untouched, because this user role is used during the database import.

   Important: If the external PostgresSQL database is an Amazon's Relational Database Service (RDS), then remove all entries for ALTER ROLE or CREATE ROLE commands from the /var/tmp/cloudera_user_roles.sql file for the Cloudera Manager database's user such as cloudera-scm, and then add the following command for the same user:

   ```
   CREATE ROLE cloudera-scm WITH NOSUPERUSER INHERIT NOCREATEROLE NOCRE
   ATEDB LOGIN NOREPLICATION NOBYPASSRLS PASSWORD '<stripped>';
   ```
7. Export the data from each of the databases on your list you created in *Identify Roles that Use the Embedded Database Server*:

   ```
   pg_dump -F c -h localhost -p 7432 -U cloudera-scm [database_name]  > /var/
   tmp/[database_name]_db_backup-$(date +"%m-%d-%Y").dump
   ```

   The following is a sample data export command for the scm database:

   ```
   pg_dump -F c -h localhost -p 7432 -U cloudera-scm scm > /var/tmp/scm_db_
   backup-$(date +%m-%d-%Y).dump
   ```

   ```
   Password:
   ```
8. Stop and disable the embedded database server:

   ```
   service cloudera-scm-server-db stop
   chkconfig cloudera-scm-server-db off
   ```

Confirm that the embedded database server is stopped:

```
netstat -at | grep 7432
```

9. Back up the Cloudera Manager Server database configuration file:

```
cp /etc/cloudera-scm-server/db.properties /etc/cloudera-scm-server/db.pr
operties.embedded
```

10. Copy the file /var/tmp/cloudera_user_roles.sql and the database dump files from the embedded database server host to /var/tmp on the external database server host:

```
cd /var/tmp
scp cloudera_user_roles.sql *.dump <user>@<postgres-server>:/var/tmp
```

11. Import the PostgreSQL user roles into the external database server.

    The external PostgreSQL database server superuser password is required to import the user roles. If the superuser role has been changed, you will be prompted for the username and password.

    Note: Only run the command that applies to your context; do not run both commands.

    - To import users when using the default PostgreSQL superuser role:

      ```
      sudo -u postgres psql -f /var/tmp/cloudera_user_roles.sql
      ```

    - To import users when the superuser role has been changed:

      ```
      psql -h <database-hostname> -p <database-port> -U <superuser> -f /var/tm
      p/cloudera_user_roles.sql
      ```

      For example:

      ```
      psql -h pg-server.example.com -p 5432 -U postgres -f /var/tmp/cloudera_u
      ser_roles.sql
      ```

      ```
      Password for user postgres
      ```

12. Import the Cloudera Manager database on the external server. First copy the database dump files from the Cloudera Manager Server host to your external PostgreSQL database server, and then import the database data:

    Note: To successfully run the pg_restore command, there must be an existing database on the database server to complete the connection; the existing database will not be modified. If the -d <existing-databas e> option is not included, then the pg_restore command will fail.

    ```
    pg_restore -C -h <database-hostname> -p <database-port> -d <existing-dat
    abase> -U cloudera-scm -v <data-file>
    ```

    Repeat this import for each database.

    The following example is for the scm database:

    ```
    pg_restore -C -h pg-server.example.com -p 5432 -d postgres -U cloudera-scm
     -v /var/tmp/scm_server_db_backup-20180312.dump
    ```

    ```
    pg_restore: connecting to database for restore
    Password:
    ```

13. Update the Cloudera Manager Server database configuration file to use the external database server. Edit the /etc/cloudera-scm-server/db.properties file as follows:
    a) Update the com.cloudera.cmf.db.host value with the hostname and port number of the external database server.
    b) Change the com.cloudera.cmf.db.setupType value from "EMBEDDED" to "EXTERNAL".
14. Start the Cloudera Manager Server and confirm it is working:

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

    Note that if you start the Cloudera Manager GUI at this point, it may take up to five minutes after executing the start command before it becomes available.

    In Cloudera Manager Server, navigate to Hosts > All Hosts and confirm the number of roles assigned to hosts (this number should match what you found in Step 2); also confirm that they are in a commissioned state that matches what you observed in Step 2.
15. Update the role configurations to use the external database hostname and port number. Only perform this task for services where the database has been migrated.

    For Cloudera Management Service:

    a. Select Cloudera Management Service > Configuration, and type "7432" in the Search field.
    b. Change any database hostname properties from the embedded database to the external database hostname and port number.
    c. Click Save Changes.

    For the Oozie Service:

    a. Select Oozie service > Configuration, and type "7432" in the Search field.
    b. Change any database hostname properties from the embedded database to the external database hostname and port number.
    c. Click Save Changes.

    For Hive and Hue Services:

    a. Select the specific service > Configuration, and type "database host" in the Search field.
    b. Change the hostname from the embedded database name to the external database hostname.
    c. Click Save Changes.
16. Start the Cloudera Management Service and confirm that all management services are up and no health tests are failing.
17. Start all Services via the Cloudera Manager web UI. This should start all services that were stopped for the database migration. Confirm that all services are up and no health tests are failing.
18. On the embedded database server host, remove the embedded PostgreSQL database server:
    a) Make a backup of the /var/lib/cloudera-scm-server-db/data directory:

    ```
    tar czvf /var/tmp/embedded_db_data_backup-$(date +”%m-%d-%Y”).tgz /var/lib/cloudera-scm-server-db/data
    ```

    b) Remove the embedded database package:

    For RHEL/SLES:

    ```
    rpm --erase cloudera-manager-server-db-2
    ```

    For Ubuntu:

    ```
    apt-get remove cloudera-manager-server-db-2
    ```

    c) Delete the /var/lib/cloudera-scm-server-db/data directory.

FIPS compliance and integration guide

Installing and configuring Federal Information Processing Standards (FIPS) encryption across Cloudera Base on premises requires SafeLogic module setup, host prerequisite verification, JDK 17 upgrades, and version migration adherence.

FIPS compliance overview for Cloudera Base on premises

FIPS-compliant data processing in Cloudera Base on premises integrates SafeLogic validated cryptographic modules to meet federal security mandates including FIPS 140-3, FISMA, and FedRAMP.

Cloudera Base on premises provides a secure, enterprise-grade data processing and analytics platform capable of operating in a FIPS-compliant mode.

FIPS publication 140-3 (FIPS 140-3) is the latest U.S. government standard issued by the National Institute of Standards and Technology (NIST) defining requirements for cryptographic modules. To satisfy federal compliance mandates including FISMA, FedRAMP, and DISA STIGs, information systems must use cryptographic modules validated through the Cryptographic Module Validation Program (CMVP).

In Cloudera Base on premises 7.1.5 and higher versions, the platform supports deployment on FIPS-enabled operating systems integrated with FIPS-validated cryptographic modules. This integration replaces standard, non-compliant Java Cryptographic Extension (JCE) providers such as, standard Bouncy Castle, OpenSSL, NSS, and Libgcrypt libraries with validated cryptographic modules provided by SafeLogic.

Using FIPS with Cloudera products

FIPS 140-3 compliance in Cloudera Base on premises enables validated cryptographic encryption, strong authentication, and data governance through integrated SafeLogic CryptoComply modules.

Cloudera Base on premises achieves FIPS compliance by integrating directly with SafeLogic CryptoComply modules at the host level. To enable FIPS mode, you must configure the underlying host operating system in FIPS mode, install the SafeLogic CryptoComply modules, and configure Cloudera Base on premises to utilize those modules for cryptographic operations.

Cloudera Base on premises 7.1.5 and higher versions are capable of running in a FIPS-compliant mode. Specifically, FIPS 140-2 compliance applies to versions 7.1.5 and higher, while FIPS 140-3 compliance applies to versions 7.3.2 SP1 and higher. Cloudera currently supports a subset of platform components and features running on a FIPS-compliant operating system. Cloudera does not guarantee that the platform components themselves are FIPS 140-3 compliant.

To satisfy FIPS 140-3 and federal compliance standards, Cloudera Base on premises supports FIPS-approved keystores and algorithms with FIPS 140-3 validated cryptographic modules across its security architecture.

Cloudera recommends deploying the following components:

- Encryption in motion using TLS (Auto-TLS support).
- Encryption at rest with HDFS Transparent Data Encryption (TDE) and Ranger KMS as the backend keystore.
- Strong authentication with Kerberos and LDAP.
- Authorization, audit, and data governance with Apache Ranger and Apache Atlas.

Before upgrading, verify FIPS support for the target Cloudera Base on premises and Cloudera Manager versions using the Cloudera support matrix.

Caution:

- Configuring FIPS-compliant mode is supported only for fresh and clean cluster installations.
- Converting an existing, non-FIPS Cloudera cluster to a FIPS-compliant mode is not supported.
- Upgrading directly from FIPS 140-2 to 140-3 is not supported.
- Upgrades between supported Cloudera versions with FIPS already enabled are fully permissible, provided they maintain the same level of compliance such as transitioning from FIPS 140-3 to FIPS 140-3.

Enabling FIPS mode on a RHEL or CentOS-based operating system, configuring the required external databases for FIPS 140-3 compliance are outside the scope of Cloudera documentation. Consult your operating system and database vendor documentation for specific configuration procedures.

For more information about the supported platform components, features and all limitations, see Understanding the prerequisites.

Key differences between FIPS 140-2 and FIPS 140-3

In Cloudera Runtime 7.3.2 SP1and higher versions require FIPS 140-3. The transition from FIPS 140-2 to FIPS 140-3 represents a major evolution in security standards designed to address modern infrastructure vulnerabilities.

Table 12: Differences between FIPS 140-2 and FIPS 140-3

Category	FIPS 140-2	FIPS 140-3
Status and validation lifecycle	Legacy 140-2 module will expire in Sep, 2026. For example, standard SafeLogic and RHEL 8 modules expire through 2026.	Enforces active FIPS 140-3 validated cryptographic modules across all federal deployments.
Operating system integration	Only compatible with legacy operating systems such as RHEL 7 RHEL 8.	Requires modern operating system infrastructure such as RHEL 9.
Cryptographic modules (SafeLogic)	Uses SafeLogic CryptoComply modules (CCS, CCJ, and Libgcrypt) validated under legacy FIPS 140-2 security criteria.	Upgrades to the SafeLogic CCJ 4.x module, to maintain compliance as legacy certificates transition to the historical list.
Container and image standards	Relies heavily on Red Hat Universal Base Images (UBI 8) running standard open-source library wrappers.	Shifts container deployments toward Chainguard images to reduce CVE exposure while using built-in FIPS 140-3 Go or OpenSSL modules.

FIPS requirements and constraints

Review the supported deployments, infrastructure requirements, and operational constraints for establishing a FIPS-compliant Cloudera cluster.

Support matrix for FIPS

To establish a certified FIPS 140-3 configuration within Cloudera Base on premises, your environment must meet the baseline infrastructure prerequisites.

Supported Cloudera deployments

Review the specific Cloudera Manager versions, Cloudera Runtime releases and ecosystem components that support FIPS mode to secure your on-premises deployments through deep cryptographic integration with the underlying host operating system.

Cloudera versions

Cloudera achieves an authentic, compliant architecture by deep system integration, binding runtime configuration directly to the underlying host's cryptographic layer. The following Cloudera Manager and Cloudera Base on premises versions are supported in FIPS mode:

FIPS 140-2:

- Cloudera Manager 7.11.3, 7.13.1 and 7.13.2
- Cloudera Base on premises 7.1.9 (SP1, SP2), 7.3.1 (SP1, SP2, SP3) and 7.3.2

FIPS 140-3:

- Cloudera Manager 7.13.2.10000
- Cloudera Base on premises 7.3.2 SP1

Cloudera components support matrix

Important:

Contact your Cloudera Sales or Support team to obtain the Cloudera Base on premises SafeLogic FIPS modules.

The following core data engines and platform tools within the Cloudera ecosystem are fully supported for FIPS operations:

- Data management and storage – HDFS (including Transparent Data Encryption), Hadoop, Hadoop Credential Provider, HBase, Phoenix, Hive, Hive-on-Tez, Hive Meta Store, and Hive Warehouse Connector.
- Compute, processing, and streaming – Atlas, MapReduce, YARN, Queue Manager, Spark, Livy, Tez, Kafka, Schema Registry, Streams Messaging Manager, Streams Replication Manager, Solr, Impala, Iceberg, Kudu, Sqoop, Oozie, OMID, ZooKeeper and Zeppelin.
- User interface and operations – Cloudera Manager, Hue, and Cruise Control.
- Security – Ranger, Ranger KMS, TLS, Kerberos, Key Trustee Server, Knox.

Note: Navigator Encrypt is not supported in FIPS-compliant configurations.

System requirements

Review the supported operating system distributions, Java Runtime environments, and external metadata database configurations required for configuring Cloudera Manager and Cloudera Runtime deployments in FIPS mode.

Operating system requirements

- Cloudera supports Red Hat Enterprise Linux (RHEL) 9.6 minor versions for both FIPS and non-FIPS environments.
- Operating system instances must be pre-configured and hardened in FIPS mode before installation according to STIG and Center for Internet Security (CIS) Benchmark Level 2 guidelines.

For more information, see Operating system requirements.

Java Runtime environments (JRE/JDK)

FIPS-compliant deployments require OpenJDK 17 running across all cluster hosts.

For more information, see Java requirements.

External metadata databases

The embedded version of PostgreSQL is not supported for FIPS-compliant clusters. External databases must be explicitly configured to communicate using FIPS 140-3-compliant encryption.

The following database versions are supoprted:

- PostgreSQL 14, 15, 16, 17 and 18

For instructions on installing and configuring a database, see Install and configure database.

FIPS deployments: Constraints and operational guardrails

Review unsupported configurations, active known issues, and shared infrastructure responsibilities for FIPS cluster deployments. These environments integrate platform components with SafeLogic cryptographic modules to establish secure operational guardrails while delegating RHEL operating system maintenance.

Unsupported FIPS deployments

The following deployment scenarios and component features are not supported in FIPS-compliant environments:

- Changing the FIPS status of an existing cluster post-installation.
- Using the MapReduceIndexerTool (MRIT) localfs component when Secure Hash Algorithm 2 (SHA-2) compatibility is required.

Responsibilities and guardrails

FIPS operational boundaries define how Cloudera platform components integrate with SafeLogic cryptographic modules while leaving base operating system and database configuration to the customer.

While Cloudera integrates its platform components to work seamlessly with FIPS-validated cryptography, system operators must manage the following underlying platform compliance boundaries:

- Cryptographic layer integrity – Cloudera platform components are configured to leverage FIPS-validated libraries appropriately, but the underlying verification certificates are held directly by the third-party module vendors (SafeLogic and Red Hat).
- OS and database maintenance – Enabling FIPS mode on RHEL or configuring external metadata databases to communicate using FIPS-compliant TLS parameters is the sole responsibility of the customer.

Known issues in FIPS

Known issues in Cloudera Manager FIPS 140-3 mode on RHEL 8.10 cause OpenSSL symlink requirements and HBase ExportSnapshot SSL initialization failures.

OpenSSL HMAC symlink for RHEL8.x FIPS

Cloudera Manager 7.13.2 and lower versions

Cloudera Manager 7.13.1.400, 7.13.2.0 and 7.11.3 CHF15

When running RHEL 8.10 in FIPS mode, you must manually create symbolic links to the OpenSSL libraries on all cluster hosts before initiating a Cloudera Manager cluster installation or upgrade.

This issue affects and only applies to operating system versions other than RHEL 8.10.

```
cd /usr/lib64
ln -sf /usr/lib64/.libcrypto.so.1.1.1k.hmac /lib64/.libcrypto.
so.hmac
ln -sf /usr/lib64/.libssl.so.1.1.1k.hmac /lib64/.libssl.so.hmac
```

HBase ExportSnapshot to S3 fails with BCFKS KeyStore error

Cloudera Runtime 7.3.2.10000 with Cloudera Manager 7.13.2 in FIPS 140-3 mode.

In FIPS 140-3 mode, Cloudera Manager creates a BCFKS truststore, but the truststore path and password are not automatically applied to every JVM that participates in the ExportSnapshot workflow. As a result, Secure Sockets Layer (SSL) initialization for cloud storage clients (S3, GCS, ABFS, etc.) can fail, with the BCFKS KeyStore corrupted: MAC calculation failed error message.

The issue can occur in two stages:

1. The HBase client fails because the truststore-related JVM properties are missing from HBAS E_OPTS.
2. The MapReduce job fails because the YARN map and reduce task JVM options also do not include the required truststore and FIPS-related properties.

1. Configure the required JVM properties manually so that both the HBase client and the MapReduce task JVMs can load the Cloudera Manager-generated BCFKS truststore.

Table 13: Required JVM property configurations for HBase and YARN

Area	Required action	Reason
HBase Client Environment safety valve	Add the truststore JVM properties to HBASE_OPTS `HBASE_OPTS="$HBASE_OPTS -Djavax.net.ssl.trustStore=/var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks -Djavax.net.ssl.trustStoreType=bcfks -Djavax.net.ssl.trustStorePassword=BtjodWcTovMMjnvgGC3zgApuFwxyX39Pp2BOf1bnsJw"`	Allows the HBase client process to initialize SSL correctly when accessing S3
YARN MapReduce Map Task Java Opts	Add truststore and FIPS-related JVM properties `+${IP_JAVA_OPT} -Djavax.net.ssl.trustStore=/var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks -Djavax.net.ssl.trustStoreType=bcfks -Djavax.net.ssl.trustStorePassword=qQK0YwzTFHrF8pQocSJdHwahGoQNSq3TLgaWqnaxIGM -Dcom.safelogic.cryptocomply.fips.approved_only=true --add-modules=com.safelogic.cryptocomply.fips.core --add-modules=bctls --add-exports=java.base/sun.security.provider=com.safelogic.cryptocomply.fips.core`	Allows mapper task JVMs to use the same truststore and crypto configuration

Area	Required action	Reason
YARN MapReduce Reduce Task Java Opts	Add truststore and FIPS-related JVM properties `+${IP_JAVA_OPT} -Djavax.net.ssl.trustStore=/var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks -Djavax.net.ssl.trustStoreType=bcfks -Djavax.net.ssl.trustStorePassword=qQK0YwzTFHrF8pQocSJdHwahGoQNSq3TLgaWqnaxIGM -Dcom.safelogic.cryptocomply.fips.approved_only=true --add-modules=com.safelogic.cryptocomply.fips.core --add-modules=bctls --add-exports=java.base/sun.security.provider=com.safelogic.cryptocomply.fips.core`	Allows reducer task JVMs to use the same truststore and crypto configuration

2. After updating the safety valves, redeploy the client configuration and restart the affected services before rerunning the ExportSnapshot workflow.

Upgrading from JDK 8 to JDK 17 on FIPS

Upgrade from JDK 8 to JDK 17 on FIPS-compliant hosts by modifying system Java environment variables, provider policies, and Cloudera Manager configuration settings.

About this task

You must change the Java configurations on all hosts and in Cloudera Manager to upgrade from JDK 8 to JDK 17.

Procedure

1. Create the ccj.sh file in the /etc/profile.d/ directory to configure the Java environment variable.

```
echo "export JDK_JAVA_OPTIONS='--module-path=/opt/cloudera/fips/ccj-4.0.0.jar:/opt/cloudera/fips/bctls-safelogic.jar --add-exports java.base/sun.security.provider=com.safelogic.cryptocomply.fips.core --add-modules com.safelogic.cryptocomply.fips.core,bctls.safelogic'" >/etc/profile.d/ccj.sh
sudo chmod +x /etc/profile.d/ccj.sh
```

2. Edit the $JAVA_HOME/conf/security/java.security file to configure the java.security policy.
3. Set the security.useSystemPropertiesFile property value to false.

Note: Skip this step, if your instance of JDK 17 was installed through a tarball.

4. Comment out the default security.provider line and insert the following prioritized cryptographic providers:

```
#
# List of providers and their preference orders (see above):
#
security.provider.1=com.safelogic.cryptocomply.jcajce.provider.CryptoCo
mplyFipsProvider
security.provider.2=org.bouncycastle.jsse.provider.BouncyCastleJsseProv
ider fips:CCJ
security.provider.3=SUN
security.provider.4=SunRsaSign
security.provider.5=SunEC
security.provider.6=SunJSSE
security.provider.7=SunJCE
security.provider.8=SunJGSS
security.provider.9=SunSASL
security.provider.10=XMLDSig
security.provider.11=SunPCSC
security.provider.12=JdkLDAP
security.provider.13=JdkSASL
```

5. Comment out the default fips.providers line and register the SafeLogic and BouncyCastle compliance modules:

```
#
# Security providers used when FIPS mode support is active
#
fips.provider.1=com.safelogic.cryptocomply.jcajce.provider.CryptoComplyFi
psProvider
fips.provider.2=org.bouncycastle.jsse.provider.BouncyCastleJsseProvider f
ips:CCJ
fips.provider.3=SUN
fips.provider.4=SunRsaSign
fips.provider.5=SunEC
fips.provider.6=SunJSSE
fips.provider.7=SunJCE
fips.provider.8=SunJGSS
fips.provider.9=SunSASL
fips.provider.10=XMLDSig
fips.provider.11=SunPCSC
fips.provider.12=JdkLDAP
fips.provider.13=JdkSASL
```

6. Update the system parameters within your Cloudera Manager configuration file:
   a) Open the /etc/default/cloudera-scm-server file.
   b) In the sections designated for JDK17 or higher, and uncomment the following FIPS configurations, ensuring that they point to the correct JAR files and modules:

```
# Enable FIPS mode
#
# To enable FIPS mode set the -Dcom.cloudera.cmf.fipsMode to true
#
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS} -Dcom.cloudera.cmf.fipsMode=true"
#
# If JDK version is 11 or higher:
# Uncomment and provide values below to include CCJ with FIPS mode
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS}
-Dcom.cloudera.cloudera.cmf.fipsMode.jdk11plus.ccj.jar.path=/opt/cloude
ra/fips/ccj-4.0.0.jar
```

```
-Dcom.cloudera.cloudera.cmf.fipsMode.jdk11plus.ccj.moduleName=ccj_mod
ule_name"
#
# If JDK version is 11 or higher:
# Uncomment and provide values below to include BCTLS with FIPS mode
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS}
-Dcom.cloudera.cloudera.cmf.fipsMode.jdk11plus.bctls.jar.path=/opt/cl
oudera/fips/bctls-safelogic.jar
-Dcom.cloudera.cloudera.cmf.fipsMode.jdk11plus.bctls.moduleName=bctls_m
odule_name"
```

Example

If the .ccj jar file name is ccj-test-4.0.0.jar, then the module name becomes cj.test. The version numbers are ignored.

You can extract and confirm the precise runtime module descriptor name by using the following command:

```
$ sudo ${JAVA_HOME}/bin/jar --file=/opt/cloudera/fips/ccj-test-4.0.0.jar
 --describe-module
No module descriptor found. Derived automatic module.
ccj.test@4.0.0 automatic <---- module Name is ccj.test
requires java.base mandated contains com.safelogic.cryptocomply
```

What to do next

Validate that the Java installation is correct. For instructions, see Validate the CCJ and CCS installations for JDK 17.

Preparing for FIPS integration

Prepare for FIPS integration in Cloudera Base on premises by verifying host operating system compliance and obtaining required SafeLogic cryptographic modules.

FIPS integration for JDK 17

Configure FIPS integration for JDK 17 in your Cloudera environment to enable system-wide compliant cryptographic policies and secure data-at-rest encryption.

Preparing the hosts

Prepare cluster hosts for FIPS integration by configuring system entropy, enabling RHEL FIPS mode, and installing JDK 17.

Procedure

1. Run the `cat/proc/sys/kernel/random/entropy_avail` command to check the available entropy.

   Note: On RHEL 9, the entropy pool size is capped at 256. A value of 256 is normal and expected due to changes in the kernel's random number generator.

2. Install and start one of the following tools to maintain high entropy levels on each host:

   - rng-tools — Utilizes Hardware Random Number Generators (HWRNGs) or hypervisor entropy sources. For more information on entropy requirements, see Data at Rest Encryption Requirements.

     Run the following commands to install the package, configure the tool to start on system boot, and launch the rngd service immediately:

     ```
     sudo dnf install rng-tools
     sudo systemctl enable rngd
     sudo systemctl start rngd
     ```

- haveged — Generates entropy using CPU execution time jitter. This tool is located in the EPEL repository. For advanced configuration instructions, see haveged documentation.

  Note: The haveged tool is found in the EPEL repository but is generally not required for RHEL 9. The kernel includes built-in jitterentropy support.

  Run the following commands to install the package, configure the tool to start on system boot, and launch the haveged service immediately:

```
sudo dnf install haveged
sudo systemctl enable haveged
sudo systemctl start haveged
```

Note: Virtual machines might have insufficient sources of entropy, which can cause extreme slowness of many Cloudera Manager and Cloudera operations even with rngd and haveged enabled. To provide sufficient entropy for the strong cryptographic functions mandated by FIPS compliance, perform following hypervisor-level adjustments:

a. Ensure that the HWRNG is enabled by setting disableHwrng to false.

b. Configure alternative entropy sources if necessary.

3. Configure the host operating system for FIPS compliance.

   For instructions on enabling the system-wide cryptographic policy for FIPS mode, see:

   - For systems running RHEL 8, see Red Hat security hardening documentation.
   - For systems running RHEL 9, see Red Hat Security hardening documentation.

4. Run either of the following commands to confirm FIPS mode activation across all hosts:

Table 14: FIPS mode activation confirmation commands

Command	Expected results	Meaning
cat /proc/sys/crypto/fips_enabled	crypto.fips_enabled = 1	A value of 1 confirms FIPS is active
sysctl crypto.fips_enabled	crypto.fips_enabled = 1	A value of 1 confirms FIPS is active

5. Set up the repository for Cloudera Manager and the necessary package installations.

   a) Download the Cloudera Manager repository configuration file for your OS version from the following link:

```
https://[username]:[password]@archive.cloudera.com/p/cm7/7.13.2.10000/redhat9/yum/cloudera-manager.repo
```

   b) Edit /etc/yum.repos.d/cloudera-manager.repo to replace the changeme placeholder values with your user name and password.

```
[cloudera-manager]
name=Cloudera Manager 7.13.2.10000
baseurl=https://archive.cloudera.com/p/cm7/7.13.2.10000/redhat9/yum/
gpgkey=https://archive.cloudera.com/p/cm7/7.13.2.10000/redhat9/yum/
RPM-GPG-KEY-cloudera
username=changeme
password=changeme
gpgcheck=1
enabled=1
autorefresh=0
type=rpm-md
```

   c) Set up a local repository for environments with no access to https://archive.cloudera.com. For instructions, see Configuring local package and parcel repositories .

6. Deploy JDK 17 across all cluster hosts using one of the following installation methods:

   - OpenJDK17

- Oracle JDK17
- manual tarball archive

Warning: Do not use the yum install method. It installs a RedHat-modified JDK package, that restricts Cloudera installation when FIPS mode is active.

7. Download the cloudera-sasl JAR file.
   a) Download the sasl-sha256aes file from the Cloudera archive:
      - Open https://archive.cloudera.com/p/cdh7/<cdp_version>/maven-repository/com/cloudera/sasl/sasl-sha256aes/, where <cdp_version> is the four digit version number, such as, 7.1.9.20000.
      - Select the 0.1.0.<cdp_version>-<xx> folder, <xx> is an internal build iteration number. For example, for 7.1.9.20000 it is 24.
      - Download the sasl-sha256aes-0.1.0.<cdp_version>-<xx>.jar file.
   b) Copy the file to the following locations on all hosts in the cluster:

   a. /opt/cloudera/fips/ (or to your designated CCJ/BCTLS storage)
   b. /opt/cloudera/parcels/CDH/jars/
   c. /opt/cloudera/parcels/CDH/lib/hadoop/
8. Install CryptoComply for Java (CC for Java) SafeLogic on all hosts.
   a) Download the required SafeLogic CC Java module JAR, BCTLS Java module JAR and BCUTIL Java module JAR files.
   b) Create the /opt/cloudera/fips/ directory using the `sudo mkdir /opt/cloudera/fips/` command.
   c) Copy the ccj-4.0.0.jar, bctls-2.0.17.1.jar and bcutil-2.0.1.jar files to the /opt/cloudera/fips/ directory.
   d) Change the file permissions of the JAR files to root using the following command:

```
chown root: /opt/cloudera/fips/ccj-4.0.0-fips.jar
chown root: /opt/cloudera/fips/bctls-2.0.17.1.jar
chown root: /opt/cloudera/fips/bcutil-2.0.1.jar
```

   e) Set the file permissions of both JAR files to 0644 using the following command:

```
chmod 0644 /opt/cloudera/fips/ccj-4.0.0-fips.jar
chmod 0644 /opt/cloudera/fips/bctls-2.0.17.1.jar
chmod 0644 /opt/cloudera/fips/bcutil-2.0.1.jar
```

   Note: Ensure that the ccj-4.0.0-fips.jar, bctls-2.0.17.1.jar and bcutil-2.0.1.jar files are stored in the same directory.
9. Create a configuration script named ccj.sh under /etc/profile.d/ to configure the Java environment variables.

```
echo "export JDK_JAVA_OPTIONS='--module-path=/opt/cloudera/fips/ccj-4.0.0-
fips.jar:/opt/cloudera/fips/bctls-2.0.17.1.jar
--add-exports java.base/sun.security.provider=com.safelogic.cryptocomply.f
ips.core
--add-modules com.safelogic.cryptocomply.fips.core,bctls'">/etc/profile.d/
ccj.sh
```

10. Configure the java.policy file, by adding the CCJ configuration block inside the closing bracket at the end of the $JAVA_HOME/conf/security/java.policy file, in the following format:

```
//CCJ Java Permissions
permission java.lang.RuntimePermission "getProtectionDomain";
permission java.lang.RuntimePermission "accessDeclaredMembers";
permission java.util.PropertyPermission "java.runtime.name", "read";
permission java.security.SecurityPermission "putProviderProperty.CCJ";
//CCJ Key Export and Translation
permission com.safelogic.cryptocomply.crypto.CryptoServicesPermission "exp
ortKeys";
```

```
//CCJ SSL
permission com.safelogic.cryptocomply.crypto.CryptoServicesPermission "t
lsAlgorithmsEnabled";
//CCJ Setting of Default SecureRandom
permission com.safelogic.cryptocomply.crypto.CryptoServicesPermission "d
efaultRandomConfig";
//CCJ Setting CryptoServicesRegistrar Properties
permission com.safelogic.cryptocomply.crypto.CryptoServicesPermission "glo
balConfig";
//CCJ Enable JKS
permission com.safelogic.cryptocomply.jca.enable_jks "true";
};
```

11. Append the security provider configuration to the java.security file.

```
security.provider.1=com.safelogic.cryptocomply.jcajce.provider.CryptoCom
plyFipsProvider
security.provider.2=org.bouncycastle.jsse.provider.BouncyCastleJsseProvi
der fips:CCJ
security.provider.3=com.cloudera.security.sasl.ClouderaSaslProvider
security.provider.4=SUN
security.provider.5=SunRsaSign
security.provider.6=SunEC
security.provider.7=SunJSSE
security.provider.8=SunJGSS
security.provider.9=SunSASL
security.provider.10=XMLDSig
security.provider.11=SunPCSC
security.provider.12=JdkLDAP
security.provider.13=JdkSASL
security.provider.14=SunPKCS11

ssl.KeyManagerFactory.algorithm=X.509
ssl.TrustManagerFactory.algorithm=PKIX
security.useSystemPropertiesFile=false
keystore.type=bcfks
```

What to do next

Install Cloudera Manager Server

Installing Cloudera Manager server

Install Cloudera Manager server daemons and configure system environment flags to enable FIPS mode alongside SafeLogic cryptographic path variables for Java environments.

Procedure

1. Log in to the Cloudera Manager server host.
2. Install the Cloudera Manager server.

```
sudo yum install cloudera-manager-daemons cloudera-manager-agent cloudera-
manager-server
```

3. Add the following line at the end of the /etc/default/cloudera-scm-server file:

```
export
CMF_ADD_TO_CLASSPATH="${CMF_ADD_TO_CLASSPATH}:/opt/cloudera/fips/bcutil-
2.0.1.jar"
```

4. In Cloudera Manager 7.13.2.1000 and higher versions, the fipsSaslMode can be enabled by appending the following line to the /etc/default/cloudera-scm-server file:

```
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS} -Dcom.cloudera.cmf.fipsSaslMode=t
rue"
```

5. Update the Cloudera Manager settings by accessing the /etc/default/cloudera-scm-server file and uncomment the following configurations related to FIPS:

```
# Enable FIPS mode
#
# To enable FIPS mode set the -Dcom.cloudera.cmf.fipsMode to true
#
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS} -Dcom.cloudera.cmf.fipsMode=true"
#
# If JDK version is 11 or higher:
# Uncomment and provide values below to include CCJ with FIPS mode
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS}
-Dcom.cloudera.cloudera.cmf.fipsMode.jdk11plus.ccj.jar.path=/opt/cloudera/
fips/ccj-4.0.0-fips.jar
-Dcom.cloudera.cloudera.cmf.fipsMode.jdk11plus.ccj.moduleName=ccj_module_
name"
#
# If JDK version is 11 or higher:
# Uncomment and provide values below to include BCTLS with FIPS mode
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS}
-Dcom.cloudera.cloudera.cmf.fipsMode.jdk11plus.bctls.jar.path=/opt/cloude
ra/fips/bctls-2.0.17.1.jar
-Dcom.cloudera.cloudera.cmf.fipsMode.jdk11plus.bctls.moduleName=bctls_mo
dule_name"
```

The value of the *[***MODULENAME***]* variable of the ccj and bctls modules depends on how the JAR files are published. If the JAR file has the **Automatic Module Name** label inside the META-INF/MANIFEST.MF file of the JAR file, use that as the module name. Otherwise, use the name of the JAR file, separated by dots, as the default module name.

For a JAR file named ccj-test-4.0.0.jar, the assigned module name is ccj.test, as the versioning information is not included.

To identify the specific module name, run the following command:

```
sudo ${JAVA_HOME}/bin/jar --file=/opt/cloudera/fips/ccj-4.0.0-fips.jar --
describe-module
No module descriptor found. Derived automatic module.
com.safelogic.cryptocomply.fips.core automatic <---- module Name is com.s
afelogic.cryptocomply.fips.core
```

What to do next

Validate the CCJ and CCS installations

Validating the CCJ and CCS installations

Validate CryptoComply for Java (CCJ) and CryptoComply for Server (CCS) installations in FIPS mode by verifying kernel module status, active security providers, and maximum cipher key lengths.

Procedure

1. Run the following command to verify that the FIPS configuration is enabled:

```
sysctl crypto.fips_enabled
```

The expected output is

```
crypto.fips_enabled = 1
```

2. Run the following command to test that the kernel module blocks non-FIPS algorithms, such as MD5:

```
echo greeting | openssl md5
```

A failure of this command confirms that FIPS mode is active.

3. On the Cloudera Manager server host, create a Java test script named ListSecurityProviders.java:

```
cat > ListSecurityProviders.java <<-EOF
import java.security.Provider;
import java.security.Security;

public class ListSecurityProviders {
 public static void main(String[] args) {
 Provider[] providers = Security.getProviders();
 for (Provider provider : providers) {
 System.out.println("Provider: " + provider.getName());
 System.out.println("Version: " + provider.getVersionStr());
 System.out.println("Info: " + provider.getInfo());
 System.out.println();
 }
 }
}
EOF
```

This process locates the ccj JAR file at the designated path, assuming that the bctls file is co-located.

4. Run the following command with the SafeLogic module path to verify that the FIPS-compliant cryptographic security providers are properly loaded and configured in the Java runtime environment (JRE):

```
java -p /opt/cloudera/fips/ ListSecurityProviders.java
```

The terminal displays the list of security providers linked to these modules. For example,

```
Provider: CCJ
Version: <version>
Info: CryptoComply® for Java version <version>
Provider: BCJSSE
Version: <version>
Info: Bouncy Castle JSSE Provider Version <version>
```

5. Run the following command to get the maximum allowed key length:

```
read -r -d '' do_maxAESKeyLength <<EOF
java.lang.System.out.println(javax.crypto.Cipher.getMaxAllowedKeyLength("
AES/CBC/PKCS5Padding"));
EOF
answer=`${JAVA_HOME}/bin/jrunscript -Dcom.safelogic.cryptocomply.fips.a
pproved_only=true -e "$do_maxAESKeyLength"`
echo $answer
```

The expected output is:

```
2147483647
```

What to do next

Install and configure databases

Installing and configuring databases

Configure external databases using FIPS-compliant JDBC drivers and Bouncy Castle FIPS Keystore (BCFKS) formats to establish secure TLS connections across cluster components.

About this task

Note: The embedded PostgreSQL database is not supported on FIPS-enabled operating systems.

Procedure

1. Configure the database to meet FIPS requirements by following the guidelines provided by your database vendor.
2. Enable TLS/SSL client support on your database, to maintain FIPS compliance for all Java Database Connectivity (JDBC) connections. For details, refer to your databse vendor documentation.
3. Set up the JDBC Driver with TLS/SSL and Bouncy Castle FIPS Keystore (BCFKS) to comply with FIPS standards. For additional details, see Configuring SSL/TLS from the various CDH Services to their respective PostgreSQL Databases.
4. Complete all the database configurations to ensure that they are fully prepared for integration with Cloudera Runtime and Cloudera Manager components. For instaructions, see Install and Configure PostgreSQL for Cloudera.

   Note: On FIPS-enabled clusters running Cloudera Runtime 7.1.9 SP1 or higher versions or higher versions, if Cloudera Data Explorer (Hue) uses PostgreSQL as the backend database, you must manually install the psycopg2 package (version 2.9.5 or higher) from source on all Hue hosts. The standard psycopg2-binary package includes its own libssl library, that is not FIPS-compliant. For instructions, see Installing the psycopg2 Python package for PostgreSQL database on a FIPS cluster (RHEL 8).
5. Start the Cloudera Manager instance and run the following command to start the server:

```
sudo systemctl start cloudera-scm-server
```

What to do next

Configure the Spark Cluster

Configuring the Spark Cluster

Configure the Spark cluster for FIPS environments by adding Cloudera Manager advanced configuration snippet settings to export SafeLogic modules and define Java cryptographic runtime options.

Procedure

1. Set the spark-env.sh advanced configuration snippet in Cloudera Manager for your deployed Spark role:
   - For SPARK_ON_YARN>GATEWAY: Add the spark-conf/spark-env.sh_client_config_safety_valve advanced configuration snippet settings for Spark cluster mode in the Cloudera Manager instance:

     ```
     export SPARK_SUBMIT_OPTS="$SPARK_SUBMIT_OPTS
     --add-exports=java.base/sun.security.provider=bctls
     --add-exports=java.base/sun.security.provider=com.safelogic.cryptocom
     ply.fips.core
     --add-modules=com.safelogic.cryptocomply.fips.core,bctls
     --module-path=/opt/cloudera/fips"
     ```
   - For SPARK3_ON_YARN>GATEWAY: Add the spark3-conf/spark-env.sh_client_config_safety_valve advanced configuration snippet settings for Spark cluster mode in the Cloudera Manager instance:

     ```
     export SPARK_SUBMIT_OPTS="$SPARK_SUBMIT_OPTS
     --add-exports=java.base/sun.security.provider=bctls
     ```

```
--add-exports=java.base/sun.security.provider=com.safelogic.cryptocom
ply.fips.core
--add-modules=com.safelogic.cryptocomply.fips.core,bctls --module-path=/
opt/cloudera/fips"
```

2. Set the spark-defaults.conf advanced configuration snippet in Cloudera Manager to ensure Spark operates correctly in a FIPS environment:
   - For SPARK_ON_YARN>GATEWAY: Add settings to spark-conf/spark-defaults.conf_client_config_safety_valve.

```
spark-conf/spark-defaults.conf_client_config_safety_valve
spark.yarn.am.extraJavaOptions=--add-exports=java.base/sun.security.pr
ovider=bctls
--add-exports=java.base/sun.security.provider=com.safelogic.cryptocomp
ly.fips.core
--add-modules=com.safelogic.cryptocomply.fips.core,bctls --module-path=/
opt/cloudera/fips
-Dcom.safelogic.cryptocomply.fips.approved_only=true -Djava.net.preferI
Pv4Stack=true
-Djdk.tls.ephemeralDHKeySize=2048 -Djdk.tls.trustNameService=true
-Dorg.bouncycastle.jsse.client.assumeOriginalHostName=true
spark.driver.extraJavaOptions=--add-exports=java.base/sun.security.prov
ider=bctls
--add-exports=java.base/sun.security.provider=com.safelogic.cryptocomply
.fips.core
--add-modules=com.safelogic.cryptocomply.fips.core,bctls --module-path=/
opt/cloudera/fips
-Dcom.safelogic.cryptocomply.fips.approved_only=true -Djava.net.prefe
rIPv4Stack=true
-Djdk.tls.ephemeralDHKeySize=2048 -Djdk.tls.trustNameService=true
-Dorg.bouncycastle.jsse.client.assumeOriginalHostName=true
spark.executor.extraJavaOptions=--add-exports=java.base/sun.security.
provider=bctls
--add-exports=java.base/sun.security.provider=com.safelogic.cryptocomply
.fips.core
--add-modules=com.safelogic.cryptocomply.fips.core,bctls --module-path=/
opt/cloudera/fips
-Dcom.safelogic.cryptocomply.fips.approved_only=true -Djava.net.prefe
rIPv4Stack=true
-Djdk.tls.ephemeralDHKeySize=2048 -Djdk.tls.trustNameService=true
-Dorg.bouncycastle.jsse.client.assumeOriginalHostName=true
```

   - For SPARK3_ON_YARN>GATEWAY: Add settings to spark3-conf/spark-defaults.conf_client_config_safety_valve.

```
spark.yarn.am.extraJavaOptions=--add-exports=java.base/sun.security.prov
ider=bctls
--add-exports=java.base/sun.security.provider=com.safelogic.cryptocomply
.fips.core
--add-modules=com.safelogic.cryptocomply.fips.core,bctls --module-path=/
opt/cloudera/fips
-Dcom.safelogic.cryptocomply.fips.approved_only=true -Djava.net.prefe
rIPv4Stack=true
-Djdk.tls.ephemeralDHKeySize=2048 -Djdk.tls.trustNameService=true
-Dorg.bouncycastle.jsse.client.assumeOriginalHostName=true
spark.driver.defaultJavaOptions=--add-exports=java.base/sun.security.
provider=bctls
--add-exports=java.base/sun.security.provider=com.safelogic.cryptocomply
.fips.core
--add-modules=com.safelogic.cryptocomply.fips.core,bctls --module-path=/
opt/cloudera/fips
```

```
-Dcom.safelogic.cryptocomply.fips.approved_only=true -Djava.net.prefe
rIPv4Stack=true
-Djdk.tls.ephemeralDHKeySize=2048 -Djdk.tls.trustNameService=true
-Dorg.bouncycastle.jsse.client.assumeOriginalHostName=true
spark.executor.defaultJavaOptions=--add-exports=java.base/sun.securit
y.provider=bctls
--add-exports=java.base/sun.security.provider=com.safelogic.cryptocomp
ly.fips.core
--add-modules=com.safelogic.cryptocomply.fips.core,bctls
```

.

Transition from FIPS 140-2 to FIPS 140-3

Review the compliance timeline and critical upgrade limitations for transitioning from FIPS 140-2 to FIPS 140-3 in Cloudera Base on premises.

The NIST Cryptographic Module Validation Program (CMVP) has set a strict timeline for the transition from FIPS 140-2 to FIPS 140-3, with all active FIPS 140-2 certificates moving to Historical status on September 21, 2026.

To maintain federal compliance requirements, Cloudera recommends updating your security modules to FIPS 140-3. FIPS 140-3 support is available in Cloudera Base on premises 7.3.2 SP1 and higher releases. For more information about the differences between FIPS 140-2 and FIPS 140-3, see Key differences between FIPS 140-2 and FIPS 140-3.

Important:

- Configuring FIPS-compliant mode is supported only for fresh, clean cluster installations.
- Converting an existing, non-FIPS Cloudera cluster to a FIPS-compliant mode is not supported.
- Upgrading directly from FIPS 140-2 to 140-3 is not supported.

Support matrix for data migration

Data migration between FIPS compliance levels requires a fresh cluster installation for Cloudera Base on premises and Cloudera Manager workloads.

Upgrades between supported Cloudera versions with FIPS already enabled are fully supported, provided they maintain the same level of compliance.

When migrating between different compliance levels, such as from 140-2 to 140-3, you must perform a fresh installation and migrate your data from the source cluster to the new cluster, using Replication Manager.

The following table lists the supported source and target versions for migrationlists the supported source and target versions for migration, provided the source clusters are FIPS-enabled:

Table 15: Support source and target versions for FIPS data migration

Component	FIPS 140-2 (source version)	FIPS 140-3 (target version)
Cloudera Manager	7.11.3, 7.13.1 and 7.13.2	7.13.2.10000
Cloudera Base on premises	7.1.9 (SP1, SP2), 7.3.1 (SP1, SP2, SP3) and 7.3.2	7.3.2 SP1

Migrating from FIPS 140-2 to FIPS 140-3

Migrate your FIPS 140-2 workloads to a FIPS 140-3 cluster in Cloudera Base on premises by establishing mutual trust stores and Java environment options between source and target hosts.

About this task

Perform a fresh installation of 7.3.2 SP1 to provision a FIPS 140-3 cluster, and migrate your workloads from the existing FIPS 140-2 cluster.

Before you begin

Your environment must meet the following conditions:

- **Hive external table replication** – To enable Hive replication between FIPS-enabled clusters, the source cluster must run Cloudera Manager 7.13.1.800 or higher versions.
- **HBase replication** – Direct HBase replication is not supported. The migration requires exporting HBase tables to HDFS, transferring them to the target cluster by HDFS replication, and importing them back into HBase. HBase tables can be exported in multiple ways, including using snapshots. For instructions, see Configuring HBase snapshots.

Procedure

1. Locate the internal Certificate Authority (CA) certificate on the source cluster. On clusters with AutoTLS enabled, this is stored on the Cloudera Manager Server host.

```
cat /opt/cloudera/CMCA/trust-store/cm-auto-in_cluster_ca_cert.pem
```

2. Copy the extracted cm-auto-in_cluster_ca_cert.pem file from the source cluster to all hosts within the target cluster.
3. On the target cluster Cloudera Manager Server host, query the backend database to find the required truststore password. The following example assumes a local PostgreSQL deployment:

```
sudo -u postgres psql
-- Connect to the CM database
\c [cm_database]
-- Retrieve the truststore password
SELECT * FROM CONFIGS WHERE attr LIKE 'truststore_password';
```

4. Run the following command on all hosts in the target cluster to import the source cluster CA certificate into the BCFKS-formatted truststore.

```
/usr/java/default/bin/keytool -storetype BCFKS \
  -provider com.safelogic.cryptocomply.jcajce.provider.CryptoComplyFips
Provider \
  -providerpath /opt/cloudera/fips/[ccj_JAR_path] \
  -importcert -alias cmrootca-source \
  -keystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststo
re.jks \
  -storepass [truststore_password] \
  -file [/path/to/cm-auto-in_cluster_ca_cert.pem]
```

5. To complete the two-way trust, repeat steps 1 through 4, but reverse the roles. Extract the CA from the target cluster and import it into the truststore of all hosts in the source cluster. Use a unique alias, such as cmrootca-target.
6. For services such as Ozone replication on FIPS clusters, you must ensure Java clients use the correct BCFKS truststore. Create an environment script on all hosts to specify the global truststore and its credentials.

```
echo "export JDK_JAVA_OPTIONS='--module-path=/opt/cloudera/fips/ccj-[ver
sion].jar:/opt/cloudera/fips/bctls.jar --add-exports java.base/sun.secur
ity.provider=com.safelogic.cryptocomply.fips.core --add-modules com.safe
logic.cryptocomply.fips.core,bctls -Djavax.net.ssl.trustStore=/var/lib/c
loudera-scm-agent/agent-cert/cm-auto-global_truststore.jks -Djavax.net.s
sl.trustStoreType=BCFKS -Djavax.net.ssl.trustStorePassword=[truststore_p
assword]'" > /etc/profile.d/ccj.sh
sudo chmod +x /etc/profile.d/ccj.sh
```

What to do next

Warning: If certificates are rotated or recreated on either the source or target cluster, you must repeat the entire trust configuration process. Failing to do so results in a loss of mutual connectivity between clusters.

Related Information

Enabling replication between clusters with Kerberos authentication

Install and configure additional required components

Use the following steps to install additional required components for FIPS.

About this task

Note:

FIPS publications (including the 140-2 publication) do not mandate the use of Apache Knox, Kerberos, Apache Ranger, Apache Atlas, HDFS TDE, or Ranger KMS. However, it is typical that these features are employed with Cloudera deployments in government regulated environments to achieve many types of government authorizations with various government standards, such as FedRAMP or FISMA. FIPS and NIST publications serve as the basis for many of these government standards.

Before you begin

Default Truststore Behavior in FIPS Mode

When Cloudera Manager runs in FIPS-enabled mode with the default keystore type set to BCFKS in java.security, Cloudera Manager does not read from the JVM's default truststore (cacerts).

Why Cloudera Manager does not use cacerts in the FIPS mode

- In standard (non-FIPS) mode, Cloudera Manager relies on the JVM's default cacerts truststore, which is in JKS format. This allows Cloudera Manager to automatically trust the default CA certificates shipped with the JDK and establish secure connections to external endpoints such as S3 or Cloudera repositories.
- In FIPS mode, only FIPS-compliant keystore formats (such as BCFKS) are permitted. Because the default cacerts is in JKS format, Cloudera Manager skips loading it to prevent startup failures and instead uses the Cloudera Manager-provided truststore exclusively. This ensures consistent and reliable operation under FIPS compliance, but means that Cloudera Manager does not automatically trust certificates present only in the JVM's default cacerts.

  As a result, connections to certain external endpoints (for example, public cloud services, S3, Cloudera download URLs) might fail unless their certificates are also present in the Cloudera Manager-provided truststore.
- If Cloudera Manager attempts to load the JKS-formatted cacerts file while running in FIPS mode, it will fail the operation.

Preserving the default JVM truststore behavior in FIPS mode

If you want Cloudera Manager to continue trusting the same set of CA certificates that the JVM ships with (cacerts), you can explicitly configure a BCFKS-converted truststore by performing the following steps:

1. Locate the existing JVM cacerts file by running the following command:

   ```
   $JAVA_HOME/lib/security/cacerts
   ```

   The default password is usually changeit.

2. Create a backup copy of the file by running the following command:

```
cp $JAVA_HOME/lib/security/cacerts /tmp/cacerts.original
```

3. Convert it to BCFKS format by running the following command:

```
$JAVA_HOME/bin/keytool -importkeystore \
  -srckeystore $JAVA_HOME/lib/security/cacerts \
  -srcstoretype JKS \
  -srcstorepass changeit \
  -destkeystore $JAVA_HOME/lib/security/cacerts.bcfks \
  -deststoretype BCFKS \
  -deststorepass changeit \
  --provider com.safelogic.cryptocomply.jcajce.provider.Cryp
toComplyFipsProvider \
  --providerpath /path/to/ccj/fips/jar
```

   Use the same value for providerpath as the one provided in /etc/default/cloudera-scm-server for -Dcom.cloudera.cloudera.cmf.fipsMode.jdk11plus.ccj.jar.path.

4. Replace the old cacerts with the converted one by running the following command:

```
mv cacerts.bcfks cacerts
```

   Now, the file name remains cacerts, but the format inside is BCFKS.

5. **Configure the Cloudera Manager Truststore**

You must ensure Cloudera Manager correctly identifies your truststore and password. While Cloudera Manager attempts to use default paths, you must manually configure the settings if you use non-default locations or passwords.

a. Identify your truststore requirements: Determine if you are using the default system truststore or a custom file (for example, a .bcfks or .jks file in a specific directory).

b. Access the configuration file: Open /etc/default/cloudera-scm-server on the Cloudera Manager Server host.

c. Define the Java options: Add or update the CMF_JAVA_OPTS variable to explicitly define the path, type, and password.

Example Configuration:

Run the following command or manually append the flags to the file:

```
export CMF_JAVA_OPTS="$CMF_JAVA_OPTS \
                      -Djavax.net.ssl.trustStore=/path/to/truststo
re.file \
                      -Djavax.net.ssl.trustStoreType=BCFKS \
                      -Djavax.net.ssl.trustStorePassword=your_pa
ssword"
```

Table 16: Usage Scenarios

Configuration Type	Action Required
Default Path & Password	Cloudera Manager typically detects the system cacerts automatically; however, explicitly defining it in CMF_JAVA_OPTS prevents resolution errors during upgrades.
Non-Default Path	You must set -Djavax.net.ssl.trustStore to the absolute path of your specific certificate file.
Non-Default Password	You must include -Djavax.net.ssl.trustStorePassword to ensure Cloudera Manager can decrypt and read the certificates.

6. Restart Cloudera Manager Server to apply the changes:

```
sudo systemctl restart cloudera-scm-server
```

Important:

- This setup preserves the same certificates as the JDK's default cacerts, but in a FIPS-compliant format.
- If necessary, you can add additional certificates (for example, cloud provider roots) to this BCFKS truststore.
- If you do not configure this, Cloudera Manager will only use the Cloudera Manager-provided truststore, and some external connections might fail.
- Be aware that setting javax.net.ssl.trustStorePassword makes the truststore password available as a JVM-wide system property. This means the password is visible to all code running inside the Cloudera Manager JVM.

  When Java code needs to open a secure connection (TLS/SSL) and does not explicitly say which truststore/keystore to use, the JVM falls back to its default SSLContext.

  So, if you set the above configurations at the JVM level, all code using the default SSLContext picks them up automatically. Unlike javax.net.ssl.trustStore and java x.net.ssl.trustStoreType, which only specify the truststore file and format, the password property also exposes a sensitive secret to the entire JVM process.

Procedure

1. Perform the Additional Steps for Apache Ranger on page 175.
2. Add Ranger to the Shadow group.

```
usermod -a -G shadow ranger
```

3. Install and Configure TLS either automatically or manually.

   If you are using Auto-TLS, see:

   - Use case 1: Use Cloudera Manager to generate an internal CA and corresponding certificates
   - Use case 2: Use an existing Certificate Authority

   If you are manually configuring TLS, see:

   - Manually Configuring TLS Encryption for Cloudera Manager

   Note: Additional manual TLS configuration steps may be required for stack components.

   Generate certificates in BCFKS format

   The standard keytool utility distributed with the JDK can generate BCFKS formatted keystores using the CCJ security provider. When the CCJ security provider is statically installed into the JDK as previously described, there is no need to pass the keytool utility the -providerpath path/to/ccj-3.0.2.1.jar or -providerclass com.safelogic.cryptocomply.jcajce.provider.ProvBCFKS arguments. It is only necessary to pass BCFKS as the storetype for the keytool operation being invoked.

   For example, keytool -importkeystore can be used to import a PKCS12 keystore into a BCFKS keystore:

```
keytool \
    -importkeystore -v \
    -srckeystore <pkcs12_keystore_file> \
    -srcstoretype PKCS12 \
    -srcstorepass <pkcs12_pass> \
    -destkeystore <bcfks_keystore_file> \
```

```
    -deststoretype BCFKS \
    -deststorepass <bcfks_keystore_pass> \
    -destkeypass <bcfks_key_pass>
```

Systems administrators and other platform implementers should consult their organization's information systems security managers for the correct procedures for generating keypairs and requesting signing of x509 certificates. The Cloudera requires the private key and signed certificate in both PEM encoded and BCFKS keystore format. The steps to accomplish this task might look similar to the following:

a) openssl genpkey
b) openssl req
c) Have the CA sign the CSR.
d) Import the private key and signed certificate into a PKCS12 keystore:

```
openssl pkcs12
```

e) Import the PKCS12 keystore into a BCFKS keystore:

```
keytool -importkeystore
```

4. Enable Kerberos authentication using the Cloudera Manager Kerberos wizard.
5. Set the kdc_timeout value in the krb5.conf file to a high enough setting to avoid client timeout errors while running queries.
   a) Open the /etc/krb5.conf file with a text editor.
   b) Under [libdefaults], set the kdc_timeout value to a minimum of 5000 (5 seconds).
6. Install Apache Knox. See Installing Apache Knox on page 177.
7. Install Ranger KMS.

   As Ranger KMS KTS and Key Trustee Server are not supported anymore, you need to migrate your encryption keys from Ranger KMS KTS and Key Trustee Server to Ranger KMS. For more information, see Key migration in UCL.

   Note: Ranger KMS uses the PBEWithMD5AndTripleDES algorithm for key encryption, which is not FIPS-compliant. Hence, for FIPS-enabled clusters, PBKDF2WithHmacSHA256 as the key derivation function (KDF) and AES/CBC/PKCS7Padding as the cipher are used for encrypting master key and zone keys. The existing keys have to be re-encrypted. The re-encryption process is as follows:

   a. Decrypt the existing keys using the older PBEWithMD5AndTripleDES algorithm and retrieve the key material.
   b. Then, encrypt the retrieved key material using the FIPS-compliant algorithm, PBKDF2WithHmacSHA256.
   c. Finally store it in the service DB.

   To decrypt the existing keys, the old PBEWithMD5AndTripleDES algorithm and the corresponding SunJCE provider are needed. This re-encryption happens when the Ranger KMS process starts.
8. Configure HDFS Transparent Data Encryption with Ranger KMS.

Production Installation

These topics provide procedures for installing Cloudera Base on premises in a production environment.

Related Information

Cloudera Base on premises Installation Guide

Before You Install

Before you begin a production installation of Cloudera Manager, Cloudera Runtime, and other managed services, review the System Requirements on page 12, in addition to the Cloudera Base on premises Release Notes.

For planning, best practices, and recommendations, review the Cloudera Base on premises Reference Architecture.

Important:

- In a typical installation process, socket's somax connection must NOT be set to a very low value (default is 128, which is very low for Hadoop systems). Cloudera recommends that you set socket's somax connection value to atleast 16000 OOTB through Cloudera Manager or host inspector.
- Security-Enhanced Linux (SELinux) allows you to set access control through policies. However, if you are unable to deploy the Cloudera Runtime cluster using your policies, you can set SELinux in permissive mode on each host of your cluster before you deploy the Cloudera Runtime parcels.
- Multiple Ozone deployments are not supported in the Cloudera cluster. Ensure that you have only one Ozone deployement Service ID in the ozone.om.service.ids configuration of the Cloudera cluster. If you have multiple Ozone deployement Service IDs in the ozone.om.service.ids configuration, then remove all of them except one.

Caution: Cloudera requires disabling the fapolicyd daemon present in RHEL 8 (and later) systems before beginning installation of Cloudera Manager application. Be informed that fapolicyd is a user space daemon that determines access rights to files based on attributes of the process and file. It can be used to either blacklist or whitelist processes or file access. Proceed with caution with enforcing the use of this daemon. Improper configuration may render the system non-functional.

Caution: Cloudera Data Services on premises 1.5.4 is not supported on Cloudera Base on premises 7.3.1. You must not install or upgrade to Cloudera Base on premises 7.3.1 if you are using Cloudera Data Services on premises 1.5.4 on your cluster as it is incompatible.

Note: The importance of security in a production environment cannot be understated. TLS and Kerberos form the baseline for secure operations of your Cloudera Runtime environment. Cloudera supports security services such as Ranger and Atlas only when they are run on clusters where Kerberos is enabled to authenticate users.

The following topics describe additional considerations you should be aware of before beginning an installation:

Install and Configure Databases

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.

Although you can deploy different types of databases in a single environment, doing so can create unexpected complications. Cloudera recommends choosing one supported database provider for all of the Cloudera databases.

Cloudera recommends installing the databases on different hosts than the services, located in the same data center. Separating databases from services can help isolate the potential impact from failure or resource contention in one or the other. It can also simplify management in organizations that have dedicated database administrators.

For information about supported databases, see Database Requirements

Required Databases

The following Cloudera Manager and Cloudera Runtime components use databases: Cloudera Manager Server, Reports Manager, Hive Metastore Server, Hue Server, Cloudera Data Visualization, Schema Registry, Oozie Server, Ranger Admin, Ranger KMS, Streams Messaging Manager Server, and YARN Queue Manager.

The type of data stored in each database and its relative size are as follows:

Cloudera Manager and Cloudera Management Service

Back up the following databases and local datastores used by Cloudera Manager and Cloudera Management Service (CMS) roles:

- Cloudera Manager Server – Contains all the information about services you have configured and their role assignments, all configuration history, commands, users, and running processes. This relatively small database (< 100 MB) is the most important to back up.

  Important: When you restart processes, the configuration for each of the services is redeployed using information saved in the Cloudera Manager database. If this information is not available, your cluster cannot start or function correctly. You must schedule and maintain regular backups of the Cloudera Manager database to recover the cluster in the event of the loss of this database.
- Reports Manager – Tracks disk utilization and processing activities over time. Medium-sized.

  Note: Ensure the JDBC connector for your database is installed on the host assigned to the Reports Manager role.
- Host Monitor and Service Monitor – These CMS roles do not use a relational database; instead, they store time-series monitoring data in local disk-based datastores that should be included in your host-level backup strategy.

Cloudera Runtime Services

Depending on the services deployed in your Cloudera Runtime 7.1.9 or higher cluster, back up the following service databases:

- Hive Metastore Server – Contains Hive table and partition metadata. Relatively small.
- Hue Server – Contains user account information, job submissions, saved workflows, and Hive/Impala queries. Relatively small.
- Cloudera Data Visualization – Stores dashboard definitions, datasets, visual configurations, and user permissions. Uses an embedded SQLite database by default, or an external relational database (PostgreSQL, MySQL, or MariaDB) in production environments.
- Schema Registry – Contains schemas, schema metadata, versions, and branches (supported on MySQL, PostgreSQL, or Oracle).

  Important: For the Schema Registry database, you must configure the database collation to be case-sensitive.
- Oozie Server – Contains Oozie workflow, coordinator, and bundle execution data. Can grow very large.

Additional Cloudera Runtime Service Databases (if deployed)

- Ranger Admin – Contains administrative information such as Ranger users, groups, and access policies. Medium-sized.
- Ranger KMS – Stores encrypted keys managed by the Ranger Key Management Service.
- Streams Messaging Manager (SMM) Server – Contains Kafka metadata, metrics, and alert definitions. Relatively small.
- YARN Queue Manager – Stores queue configurations created by YARN Queue Manager:
  - If you install Cloudera Runtime 7.1.9 CHF 2 or higher, YARN Queue Manager uses an embedded database and does not require an external relational database.
  - If you install Cloudera Runtime 7.1.9 or 7.1.9 CHF 1, YARN Queue Manager requires an external PostgreSQL database. If you upgrade from 7.1.9 or 7.1.9 CHF 1 to CHF 2 or higher, the service continues using your PostgreSQL database and does not migrate back to the embedded database.

For steps to install and configure databases for Cloudera Manager, Cloudera Runtime, and other managed services, see the setup guide for the type of database you want to use.

Related Information

Database Requirements

Install and Configure PostgreSQL for Cloudera Base on premises

To use a PostgreSQL database, follow these procedures. For information on compatible versions of the PostgreSQL database, see Database Requirements on page 28.

Note: The following instructions are for a dedicated PostgreSQL database for use in production environments, and are unrelated to the embedded PostgreSQL database provided by Cloudera for trial installations.

Installing Postgres JDBC Driver

You must install the required Postgres JDBC driver.

Download, extract, and copy the JDBC driver, renamed, to /usr/share/java/. If the target directory does not yet exist, create it.

Installing the Postgres JDBC Driver

1. To install the matching PostgreSQL driver for your OS, run the applicable command:
   RHEL

   ```
   sudo yum install postgresql-jdbc
   ```

   Ubuntu

   ```
   sudo apt-get install libpostgresql-jdbc-java
   ```

   SLES

   ```
   sudo zypper install postgresql-jdbc
   ```

2. **Deploying the Driver Dependency**

   To ensure Cloudera services successfully discover the dependency, you must link your operating system's PostgreSQL JDBC driver .jar file to postgresql-connector-java.jar (the specific filename Cloudera looks for during deployment). See the following example for a complete postgresql-jdbc.jar deployment over an RHEL 9 environment:

   The following configuration example demonstrates how to declare the PostgreSQL JDBC driver dependency, establish the connection properties within your environment, and execute a complete postgresql-jdbc.jar deployment over an RHEL 9 environment.

   Review the parameters in the following RHEL 9 deployment configuration example to ensure your application can locate the driver class and authenticate with your database instance:

   - Driver Class Path: Identifies the location of the compiled .jar file so the application runtime can load the PostgreSQL class files.
   - Connection URL: Establishes the network path, port protocol, and target database name for the connection.
   - Authentication Credentials: Defines the system user account permissions required to access the underlying schema.

   Modify the placeholder values in the following example to match your specific deployment architecture and network settings.

```
root@server1:/home/cdp # yum list postgresql-jdbc
Updating Subscription Management repositories.
Red Hat CodeReady Linux Builder for RHEL 9 x86_64 (RPMs)  130 kB/s | 4.5 kB
 00:00
Red Hat Enterprise Linux 9 for x86_64 - AppStream - Extended 135 kB/s | 4.5
 kB 00:00
Red Hat CodeReady Linux Builder for RHEL 9 x86_64 - Extended 144 KB/s | 4.5
kB 00:00
EPEL9               117 kB/s | 2.3 kB 00:00
```

```
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)   107 kB/s | 4.1 kB 0
0:00
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMS)  133 kB/s | 4.5 kB
 00:00
Red Hat Enterprise Linux 9 for x86_64 - BaseOS - Extended Upd  134 kB/s |
 4.1 kB 00:00
Available Packages
postgresql-jdoc.noarch   42.2.28-1.e19-3    rhel-9-for-x86_64-appstream-eus-
rpms
postgresql-jdbc.noarch   42.2.28-1.e19_3    rhel-9-for-x86_64-appstream-rpms
root@server1:/home/cdp # yum install postgresql-jdbc -y
Updating Subscription Management repositories.
Red Hat CodeReady Linux Builder for RHEL 9 x86_64 (RPMs)   96 kB/s | 4.5 kB
 00:00
Red Hat Enterprise Linux 9 for x86_64 - AppStream - Extended     59 kB/s |
4.5 kB 00:00
Red Hat CodeReady Linux Builder for RHEL 9 x86_64 - Extended 107 KB/s | 4.5
 kB 00:00
EPEL9                97 kB/s | 2.3 kB 00:00
Red Hat Enterprise Linux 9 for x86_64 - BaseOs (RPMs)       97 kB/s | 4.1 kB
 00:00
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMS)        102 kB/s | 4
.5 kB 00:00
Red Hat Enterprise Linux 9 for x86_64 - BaseOS - Extended Upd   96 kB/s | 4
.1 kB  00:00
Dependencies resolved.
================================================================================
=====================
Package     Arch Version    Repository        Size
================================================================================
=====================
Installing:
postgresql-jdbc   noarch 42.2.28-1.e19_3  rhel-9-for-x86_64-appstream-eus-
rpms 794 k
Installing dependencies:
 ongres-scram   noarch 1.9-betal-1.el9  rhel-9-for-x86_64-appstream-eus-rpms
 75 k
 ongres-scram-client noarch 1.9-betal-1.e19  rhel-9-for-x86_64-appstream-
eus-rpms 26  k

Transaction Summary
================================================================================
=====================
Install 3 Packages

Total download size: 895 k
Installed size: 957 k
Is this ok [y/N]: y
Downloading Packages:
(1/3): ongres-scram-1.9-betal-1.el9.noarch.rpm      55 kB/s |  75 kB 00:01
(2/3): ongres-scram-client-1.9-betal-1.el9.noarch.rpm    18 kB/s |  26 kB 00
:01
(3/3): postgresql-jdoc-42.2.28-1.e19_3.noarch.rpm    544 kB/s | 794 kB 00:01
--------------------------------------------------------------------------------
---------------------
Total               609 kB/s | 895 kB 00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing   :                 1/1
  Installing  : ongres-scram-1.9-betal-1.e19.noarch        1/3
```

```
  Installing  : ongres-scram-client-1.9-betal-1.e19.noarch       2/3
  Installing  : postgresgl-jdoc-42.2.28-1.e19 3.noarch        3/3
  Verifying   : postgresql-jdbc-42.2.28-1.el9_3.noarch        1/3
  Verifying   : ongres-scram-client-1.9-betal-1.e19.noarch       2/3
  Verifying   : ongres-scram-1.9-betal-1.el9.noarch         3/3
Installed products updated.

Installed:
  ongres-scram-1.9-betal-1.el9.noarch   ongres-scram-client-1.9-betal-1.e
l9.noarch
  postgresq1-jdbc-42.2.28-1.e19_3.noarch

Complete!
root@server1:/home/cdp # ls -l /usr/share/java
total 4396
drwxr-xr-x 2 root root  42 Nov 2 09:10 ongres-scram
-rw-r--r-- 1 root root 4498228 Dec 1  2024 oracle-connector-java.jar
lrwxrwxrwx 1 root root   30 Mar 6  2024 postgresql.jar -> postgresql-jdbc/p
ostgresql.jar
drwxr-xr-x 2 root root  28 Nov 2 09:10 postgresql-jdbc
lrwxrwxrwx 1 root root  30 Mar 6  2024 postgresql-jdbc.jar -> postgresql-jd
bc/postgresql.jar
root@server1:/home/cdp # ln -s /usr/share/java/postgresql-jdbc.jar /usr/sh
are/java/postgresql-connector-java.jar
root@server1:/home/cdp # ls -l /usr/share/java/
total 4396
drwxr-xr-x 2 root root  42 Nov 2 09:10 ongres-scram
-rw-r--r-- 1 root root 4498228 Dec 1  2024 oracle-connector-java.jar
lrwxrwxrwx 1 root root  35 Nov 2 09:20 postgresql-connector-java.jar -> /
usr/share/java/postgresql-jdbc.jar
lrwxrwxrwx 1 root root  30 Mar 6  2024 postgresql-jar -> postgresgl-jdbc/
postgresql.jar
drwxr-xr-x 2 root root  28 Nov 2 09:10 postgresql-jdbc
lrwxrwxrwx 1 root root  30 Mar 6  2024 postgresql-jdbc.jar -> postgresql-j
dbc/postgresql.jar
root@server1:/home/cdp # ls -lL /usr/share/java
total 6892
drwxr-xr-x 2 root root  42 Nov 2 09:10 ongres-scram
-rw-r--r-- 1 root root 4498228 Dec 1  2024 oracle-connector-java.jar
lrwxrwxrwx 1 root root 850044 Mar 6  2024 postgresql-connector-java.jar
lrwxrwxrwx 1 root root 850044 Mar 6  2024 postgresql-jar
drwxr-xr-x 2 root root  28 Nov 2 09:10 postgresql-jdbc
lrwxrwxrwx 1 root root 850044 Mar 6  2024 postgresql-jdbc.jar
root@server1:/home/cdp #
```

Installing PostgreSQL Server

Install the PostgreSQL packages on the PostgreSQL server.



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.
- Make sure that the data directory, which by default is /var/lib/postgresql/data/, is on a partition that has sufficient free space.
- Cloudera Manager supports the use of a custom schema name for the Cloudera Manager Server database, but not the Cloudera Runtime component databases (such as Hive and Hue). For more information, see *Scehmas* in the PostgreSQL documentation.

Install the PostgreSQL packages as follows:

RHEL

Important: To install any specific version of PostgreSQL packages, see this page.

```
sudo yum install postgresql-server
```

SLES

```
sudo zypper addrepo -t YUM http://packages.2ndquadrant.com/postgresql-z-suse
/zypper/sles-11sp3-s390x pg
sudo zypper refresh
sudo zypper in postgresql postgresql-server postgresql-contrib
```

Note: This command installs PostgreSQL 11. If you want to install a different version, you can use zypper s earch postgresql to search for an available supported version.

Ubuntu

```
sudo apt-get install postgresql
```

Installing the psycopg2 Python package for PostgreSQL database

If you are using PostgreSQL as a backend database for Data Explorer on Cloudera Base on premises 7.3.x, then you must install a version of the psycopg2 package to be at least 2.9.6 on all Data Explorer 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.

Note: For releases Cloudera Base on premises 7.3.1.100 CHF1 and above, remove the psycopg2-binary package and install psycopg2 along with its necessary dependencies.

Note: Starting with the Cloudera Runtime 7.3.2 and SP1 release, Cloudera Data Explorer (Hue) supports only Python 3.11 and Python 3.12 across all OS versions.

Before you begin, you must disable the postgresql10 section from the cloudera-manager.repo file as follows:

1. SSH in to the Cloudera Manager host as an Administrator.
2. Locate the repository file (e.g., /etc/yum.repos.d/cloudera-manager.repo on RHEL/CentOS/OEL or /etc/apt/sou rces.list.d/ on Ubuntu).
3. Set enabled=0 under the [postgresql10] section as follows:

```
[postgresql10]
name=Postgresql 10
baseurl=https://archive.cloudera.com/postgresql10/redhat8/
gpgkey=https://archive.cloudera.com/postgresql10/redhat8/RPM-GPG-KEY-PG
DG-10
enabled=0
gpgcheck=1
module_hotfixes=true
```

4. Save the file and exit.

Note: The steps to disable the postgresql10 section are applicable to all supported operating systems (CentOS, RHEL, SLES, and Ubuntu).

For CentOS RHEL OEL 7 8

The following steps apply to CentOS 8, RHEL 8:

1. SSH into the Data Explorer server host as a root user.

2. Before you install psycopg2-binary package, run the following pip command to install dependencies and to ensure smooth psycopg2 installation:

```
yum install -y postgresql postgresql-devel python3-devel gcc
```

3. Install the psycopg2-binary package as follows:
Python 3.12

```
python3.12 -m pip install --upgrade pip
python3.12 -m pip install psycopg2-binary
python3.12 -m pip show psycopg2-binary
```

Python 3.11

```
python3.11 -m pip install psycopg2-binary
python3.11 -m pip show psycopg2-binary
```

Python 3.9

```
python3.9 -m pip install psycopg2-binary
python3.9 -m pip show psycopg2-binary
```

Python 3.8

```
python3.8 -m pip install psycopg2-binary
python3.8 -m pip show psycopg2-binary
```

4. Repeat these steps on all the Data Explorer server hosts.

If you get the "Error: pg_config executable not found" error while installing the psycopg2-binary package, then run the following commands to install the postgresql, postgresql-devel, python-devel packages:

```
yum install postgresql postgresql-devel python-devel
```

For RHEL 9

The following steps apply to RHEL 9, as the minimum version of Python is 3.11:

1. SSH into the Data Explorer server host as a root user.
2. Install pip for Python as follows:

```
dnf install -y postgresql postgresql-devel python3.12-devel gcc
```

3. Add the /usr/local/bin path to the PATH environment variable:

```
export PATH=$PATH:/usr/local/bin
echo $PATH
```

4. Install the psycopg2-binary package as follows:
Python 3.12

```
python3.12 -m pip install --upgrade pip
python3.12 -m pip install psycopg2-binary
```

Python 3.11

```
python3.11 -m pip install --upgrade pip
python3.11 -m pip install psycopg2-binary
```

5. Repeat these steps on all the Data Explorer server hosts.

If you get the "Error: pg_config executable not found" error while installing the psycopg2-binary package, then run the following commands to install the postgresql, postgresql-devel, python-devel packages:

```
yum install postgresql postgresql-devel python-devel
```

For SLES

1. SSH into the Data Explorer host as a root user.
2. Install the psycopg2 package dependencies for SLES by running the following commands:

```
zypper install xmlsec1
zypper install xmlsec1-devel
zypper install xmlsec1-openssl-devel
```

3. Install the postgresql-devel package corresponding to your database version by running the following command:

```
zypper -n postgresql[***DB-VERSION***]-devel
```

4. Add the location of the installed postgresql-devel package to the PATH environment variable by running the following command:

```
export PATH=$PATH:/usr/local/bin
```

5. Install the psycopg2 package by running the following command:

```
pip3.12 install psycopg2==2.9.6 --ignore-installed
```

For Ubuntu

1. SSH into the Data Explorer host as a root user.
2. Install the psycopg2 package dependencies for Ubuntu by running the following commands:

```
apt-get install -y xmlsec1
apt-get install libxmlsec1-openssl
apt-get install libpq-dev python3-pip -y
```

3. Install the python3-dev and libpq-dev packages by running the following command:

```
apt install python3-dev libpq-dev
```

4. Add the location of the installed postgresql-devel package to the PATH environment variable by running the following command:

```
export PATH=$PATH:/usr/local/bin
```

5. Install the psycopg2 package by running the following command:

```
pip3.12 install psycopg2==2.9.6 --ignore-installed
```

Installing the psycopg2 Python package for PostgreSQL database on a FIPS cluster (RHEL 8)

If you use PostgreSQL as a backend database for Data Explorer on a FIPS cluster, you must install a version of the psycopg2 package from the source to be at least 2.9.5 on all Data Explorer hosts because the psycopg2-binary package uses its version of the libssl library file which does not support FIPS.

About this task



Note: This topic provides instructions on downloading and installing the psycopg2 package from the source. This is required for using the PostgreSQL database with Data Explorer on FIPS-enabled CDP Private Cloud Base 7.1.9 SP1 cluster and higher on RHEL 8.

Note: Starting with the Cloudera Runtime 7.3.2 and SP1 release, Cloudera Data Explorer (Hue) supports only Python 3.11 and Python 3.12 across all OS versions.

Before you begin

1. Uninstall the preinstalled psycopg2 or psycopg2-binary packages.
2. Download and install the PostgreSQL database using the following commands:

```
dnf install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL
-8-x86_64/pgdg-redhat-repo-latest.noarch.rpm
dnf install -y postgresql[***DATABASE-VERSION***]-server
yum install -y postgresql[***DATABASE-VERSION***]-devel
```

Replace *[***DATABASE-VERSION***]* with the actual database version you are want to install. For example, 16. If you do not specify the *[***DATABASE-VERSION***]*, then it defaults to 12.

Procedure

1. SSH into the Data Explorer host as a root user.
2. Download the psycopg2 package as follows:

```
wget https://files.pythonhosted.org/packages/89/d6/cd8c46417e0f7a16b4b0f
c321f4ab676a59250d08fce5b64921897fb07cc/psycopg2-2.9.5.tar.gz
```

3. Extract the tarball as follows:

```
tar -xzvf psycopg2-2.9.5.tar.gz
```

4. Locate the pg_config executable file as follows:

```
find / -name pg_config
```

5. Add the directory containing the pg_config file to the $PATH variable:

```
export PATH="/usr/pgsql-[***DATABASE-VERSION***]/bin:$PATH"
```

6. Build and install the psycopg2 package as follows:

```
/usr/bin/python3.12 setup.py build_ext --pg-config=/usr/pgsql-[***DATABA
SE-VERSION***]/bin/pg_config install
```

Configuring and Starting the PostgreSQL Server

By default, PostgreSQL only accepts connections on the loopback interface. Configure PostgreSQL to accept the connections based on hostname, IP address (including CIDR address), or MAC address. A fully qualified domain name (FQDN) is not a requirement. If you do not make these changes, the services cannot connect to and use the database on which they depend.

Before you begin

If you are making changes to an existing database, make sure to stop any services that use the database before continuing.

Procedure

1. Initialize the PostgreSQL database cluster:

OS	Command
RHEL, SLES, and Ubuntu	`/usr/bin/postgresql-setup initdb`

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

   - **RHEL**

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

   - **SLES**

     ```
     sudo su -l postgres -c "initdb --pgdata=/var/lib/pgsql/data --encoding=U
     TF-8"
     ```

   - **Ubuntu**

     ```
     echo ‘LC_ALL=“en_US.UTF-8”’ >> /etc/default/locale.conf
     ```

3. Enable SCRAM-SHA-256 authentication. Edit pg_hba.conf, which is usually found in /var/lib/pgsql/data or /etc/postgresql/<*VERSION*>/main. Add the following line:

   ```
   host all all 127.0.0.1/32 scram-sha-256
   ```

   If the default pg_hba.conf file contains the following line:

   ```
   host all all 127.0.0.1/32 ident
   ```

   then the host line specifying scram-sha-256 authentication shown above must be inserted before this ident line. Failure to do so may cause an authentication error when running the scm_prepare_database.sh script. You can modify the contents of the scram-sha-256 line shown above to support different configurations. For example, if you want to access PostgreSQL from a different host, replace 127.0.0.1 with your IP address and update postgres ql.conf, which is typically found in the same place as pg_hba.conf, to include:

   ```
   listen_addresses = '*'
   ```

   Attention: 127.0.0.1 is a loopback address. Making 127.0.0.1 the only IP for authentication means that the only host that is allowed to authenticate to the database is the one on which it is running. When you have multiple services that use the database, if they are located on other hosts, they could have problems connecting to the database and could fail. If the goal is to allow open authentication from all hosts, that should be 0.0.0.0/32, not 127.0.0.1. Otherwise, you should add lines for explicit IP addresses of hosts that need to authenticate.

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

   - Small to mid-sized clusters - Consider the following settings as starting points. If resources are limited, consider reducing the buffer sizes and checkpoint segments further. Ongoing tuning may be required based on each host's resource utilization. For example, if the Cloudera Manager Server is running on the same host as other roles, the following values may be acceptable:

     - password_encryption = scram-sha-256
     - max_connection - In general, allow each database on a host 100 maximum connections and then add 50 extra connections. You may have to increase the system resources available to PostgreSQL, as described at Connection Settings.

- shared_buffers - 256MB
- wal_buffers - 8MB
- checkpoint_segments - 16

  Note: The checkpoint_segments setting is removed in PostgreSQL 9.5 and higher, replaced by min_wal_size and max_wal_size. The PostgreSQL 9.5 release notes provides the following formula for determining the new settings:

  ```
  max_wal_size = (3 * checkpoint_segments) * 16MB
  ```

- checkpoint_completion_target - 0.9

- Large clusters - Can contain up to 1000 hosts. Consider the following settings as starting points.
  - password_encryption = scram-sha-256
  - max_connection - For large clusters, each database is typically hosted on a different host. In general, allow each database on a host 100 maximum connections and then add 50 extra connections. You may have to increase the system resources available to PostgreSQL, as described at Connection Settings.
  - shared_buffers - 1024 MB. This requires that the operating system can allocate sufficient shared memory. See PostgreSQL information on Managing Kernel Resources for more information on setting kernel resources.
  - wal_buffers - 16 MB. This value is derived from the shared_buffers value. Setting wal_buffers to be approximately 3% of shared_buffers up to a maximum of approximately 16 MB is sufficient in most cases.
  - checkpoint_segments - 128. The PostgreSQL Tuning Guide recommends values between 32 and 256 for write-intensive systems, such as this one.

    Note: The checkpoint_segments setting is removed in PostgreSQL 9.5 and higher, replaced by min_wal_size and max_wal_size. The PostgreSQL 9.5 Release Notes provides the following formula for determining the new settings:

    ```
    max_wal_size = (3 * checkpoint_segments) * 16MB
    ```

  - checkpoint_completion_target - 0.9.

  Note: On PostgreSQL12 and above, Cloudera recommends to disable jit. You can set jit=off.

5. Configure the PostgreSQL server to start at boot.

OS	Command
RHEL, SLES, and Ubuntu	`sudo systemctl enable postgresql`

6. Restart the PostgreSQL database by running the following command:

OS	Command
RHEL, SLES, and Ubuntu	`sudo systemctl restart postgresql`

Important:

On RHEL 8.4 or higher, while restarting PostgreSQL if you are unable to proceed further with the process then perform the following steps:

a. Create a directory by running the following command:

```
mkdir -p /etc/systemd/system/postgresql.service.d
```

b. Create a configuration file by running the following command:

```
cat >
/etc/systemd/system/postgresql.service.d/stop.conf <<
EOF
[Service]
ExecStopPost=+/usr/bin/systemctl daemon-reload
EOF
```

c. Reload the systemd daemon by running the following command:

```
systemctl daemon-reload
```

Creating Databases for Cloudera Software

You must create databases and service accounts for components that require databases.

About this task

The following components require databases:

- Cloudera Manager Server
- Cloudera Management Service roles:
  - Reports Manager
- Hue
- Each Hive metastore
- Oozie
- Schema Registry
- Streams Messaging Manager

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

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.

Note: The instructions for Cloudera Manager Server, Cloudera Management Service roles, Reports Manager, Hue, Hive metastores, and Oozie are documented in this topic.

Additional configuration for Ranger is documented in the following two topics. Refer to those topics for detailed instructions on the Ranger database.

To create databases for Cloudera Manager Server, Cloudera Management Service roles, Reports Manager, Hue, Hive metastores, and Oozie, complete the following steps:

Procedure

1. Connect to PostgreSQL:

```
sudo -u postgres psql
```

2. Create databases for each service you are using from the below table:

```
CREATE ROLE <USER> LOGIN PASSWORD '<PASSWORD>';
```

```
CREATE DATABASE <DATABASE> OWNER <USER> ENCODING 'UTF8';
```

You can use any value you want for <*DATABASE*>, <*USER*>, and <*PASSWORD*>. The following examples are the default names provided in the Cloudera Manager configuration settings, but you are not required to use them:

Table 17: Databases for Cloudera Software

Service	Database	User
Cloudera Manager Server	scm	scm
Reports Manager	rman	rman
Ranger RHEL/CentOS/Ubuntu	ranger	rangeradmin
Ranger KMS RHEL/CentOS	rangerkms	rangerkms
Hue	hue	hue
Hive Metastore Server	hive	hive
Oozie	oozie	oozie
Schema Registry	schemaregistry	schemaregistry
Streams Messaging Manager	smm	smm

Record the databases, usernames, and passwords chosen because you will need them later.

3. For PostgreSQL 8.4 and higher, set standard_conforming_strings=off for the Hive Metastore and Oozie databases:

```
ALTER DATABASE <DATABASE> SET standard_conforming_strings=off;
```

What to do next

- If you plan to use Apache Ranger, see the following topic for instructions on creating and configuring the Ranger database and to install the JDBC driver for the database. See Configuring a PostgreSQL Database for Ranger or Ranger KMS on page 135.
- If you plan to use Schema Registry or Streams Messaging Manager, see the following topic for instructions on configuring the database: Configuring the Database for Streaming Components on page 142
- After you install and configure PostgreSQL databases for Cloudera software, continue to Set up and Configure the Cloudera Manager Database to configure a database for Cloudera Manager.

Install and Configure MySQL for Cloudera Software

You can install a MySQL database for use with Cloudera Manager and other components that require a database.

To use a MySQL database, follow these procedures. For information on compatible versions of the MySQL database, see Database Requirements on page 28.

Before you begin

Ensure that the MySQL DB is configured with the InnoDB engine by running the following command from the MySQL shell:

```
mysql> show table status;
```

Installing the MySQL Server

Note:

- If you already have a MySQL database set up, you can skip to the section Configuring and Starting the MySQL Server on page 103 to verify that your MySQL configurations meet the requirements for Cloudera Manager.
- For MySQL 5.6 and 5.7, you must install the MySQL-shared-compat or MySQL-shared package. This is required for the Cloudera Manager Agent package installation.
- It is important that the datadir directory, which, by default, is /var/lib/mysql, is on a partition that has sufficient free space.
- Cloudera Manager installation fails if GTID-based replication is enabled in MySQL.

1. Install the MySQL database:

OS	Command

RHEL	MySQL is no longer included with RHEL. You must download the repository from the MySQL site and install it directly. You can use the following commands to install MySQL. For more information, visit the MySQL website.
	`wget <URL TO MYSQL RPM>`
	`sudo rpm -ivh <FILENAME>.rpm`
	`sudo yum update`
	`sudo yum install mysql-server`
	`sudo systemctl start mysqld`

SLES	`sudo zypper install mysql libmysqlclient_r17`
Ubuntu	`sudo apt-get install mysql-server`

Configuring and Starting the MySQL Server

Note: If you are making changes to an existing database, make sure to stop any services that use the database before continuing.

1. Stop the MySQL server if it is running.

OS	Command
RHEL Compatible, SLES, and Ubuntu	`sudo systemctl stop mysqld`

2. Move old InnoDB log files /var/lib/mysql/ib_logfile0 and /var/lib/mysql/ib_logfile1 out of /var/lib/mysql/ to a backup location.
3. Determine the location of the option file, my.cnf (/etc/my.cnf by default).
4. Update my.cnf so that it conforms to the following requirements:
   - To prevent deadlocks, set the isolation level to READ-COMMITTED.
   - Configure the InnoDB engine.

     Important: Cloudera Manager does not start if its tables are configured with the MyISAM engine. (Typically, tables revert to MyISAM if the InnoDB engine is misconfigured.)

- The default settings in the MySQL installations in most distributions use conservative buffer sizes and memory usage. Cloudera Management Service roles need high write throughput because they might insert many records in the database. Cloudera recommends that you set the innodb_flush_method property to O_DIRECT.
- Set the max_connections property according to the size of your cluster:
  - Fewer than 50 hosts - You can store more than one database (for example, both the Cloudera Manager Server and Reports Manager) on the same host. If you do this, you should:
    - Put each database on its own physical disk for best performance. You can do this by manually setting up symbolic links or running multiple database instances (each instance uses a different data directory path).
    - Allow 100 maximum connections for each database and then add 50 extra connections. For example, for two databases, set the maximum connections to 250. If you store four databases on one host (the databases for Cloudera Manager Server, Hue, Reports Manager, and Hive metastore), set the maximum connections to 450.
  - More than 50 hosts - Do not store more than one database on the same host. Use a separate host for each database/host pair. The hosts do not need to be reserved exclusively for databases, but each database should be on a separate host.
- If the cluster has more than 1000 hosts, set the max_allowed_packet property to 16M. Without this setting, the cluster may fail to start due to the following exception: com.mysql.jdbc.PacketTooBigException.
- Binary logging is not a requirement for Cloudera Manager installations. Binary logging provides benefits such as MySQL replication or point-in-time incremental recovery after database restore. Examples of this configuration follow. For more information, see The Binary Log.

Here is an option file with Cloudera recommended settings:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
transaction-isolation = READ-COMMITTED
# Disabling symbolic-links is recommended to prevent assorted security r
isks;
# to do so, uncomment this line:
symbolic-links = 0

key_buffer_size = 32M
max_allowed_packet = 16M
thread_stack = 256K
thread_cache_size = 64

# The following 3 parameters only apply to MySQL version 5.7 and lower:
query_cache_limit = 8M
query_cache_size = 64M
query_cache_type = 1

max_connections = 550
#expire_logs_days = 10
#max_binlog_size = 100M

#log_bin should be on a disk with enough free space.
#Replace '/var/lib/mysql/mysql_binary_log' with an appropriate path for
 your
#system and chown the specified folder to the mysql user.
log_bin=/var/lib/mysql/mysql_binary_log
#In later versions of MySQL, if you enable the binary log and do not set
#a server_id, MySQL will not start. The server_id must be unique within
#the replicating group.
server_id=1

binlog_format = mixed
```

```
read_buffer_size = 2M
read_rnd_buffer_size = 16M
sort_buffer_size = 8M
join_buffer_size = 8M

# InnoDB settings
innodb_file_per_table = 1
innodb_flush_log_at_trx_commit  = 2
innodb_log_buffer_size = 64M
innodb_buffer_pool_size = 4G
innodb_thread_concurrency = 8
innodb_flush_method = O_DIRECT
innodb_log_file_size = 512M

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

5. If you are using MySQL version 8, remove the following three parameters from the options file:

```
query_cache_limit = 8M
query_cache_size = 64M
query_cache_type = 1
```

6. If AppArmor is running on the host where MySQL is installed, you might need to configure AppArmor to allow MySQL to write to the binary.
7. Ensure the MySQL server starts at boot:

OS	Command
RHEL Compatible, SLES, and Ubuntu	`sudo systemctl enable mysqld`

8. Start the MySQL server:

OS	Command
RHEL Compatible, SLES, and Ubuntu	`sudo systemctl start mysqld`

9. Run /usr/bin/mysql_secure_installation to set the MySQL root password and other security-related settings. In a new installation, the root password is blank. Press the Enter key when you're prompted for the root password. For the rest of the prompts, enter the responses listed below in bold:

```
sudo /usr/bin/mysql_secure_installation
```

```
[...]
Enter current password for root (enter for none):
OK, successfully used password, moving on...
[...]
Set root password? [Y/n] Y
New password:
Re-enter new password:
Remove anonymous users? [Y/n] Y
[...]
Disallow root login remotely? [Y/n] N
[...]
Remove test database and access to it [Y/n] Y
[...]
Reload privilege tables now? [Y/n] Y
All done!
```

Installing the MySQL JDBC Driver

Install the JDBC driver on the Cloudera Manager Server host, as well as any other hosts running services that require database access.

Note: If you already have the JDBC driver installed on the hosts that need it, you can skip this section. However, MySQL 5.7 requires a 5.1 driver version 5.1.x.. You can also use version 5.1.x.x to connect to MySQL 8.x.

Important: If you are using TLS v1.2, you must use version 5.1.48.

Cloudera recommends that you consolidate all roles that require databases on a limited number of hosts, and install the driver on those hosts. Locating all such roles on the same hosts is recommended but not required. Make sure to install the JDBC driver on each host running roles that access the database.

Note: Cloudera recommends using only version 8.0 of the JDBC driver.

OS	Command

RHEL	**Important:** Using the yum install command to install the MySQL driver package before installing a JDK installs OpenJDK, and then uses the Linux alternatives command to set the system JDK to be OpenJDK. If you intend to use an Oracle JDK, make sure that it is installed before installing the MySQL driver using yum install. If you want to use OpenJDK, you can install the driver using yum.
	Alternatively, use the following procedure to manually install the driver.
	1. Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html (in .tar.gz format). As of the time of writing, you can download version 8.0.22 using wget as follows:
	`wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.22.tar.gz`
	2. Extract the JDBC driver JAR file from the downloaded file. For example:
	`tar zxvf mysql-connector-java-8.0.22.tar.gz`
	3. Copy the JDBC driver, renamed, to /usr/share/java/. If the target directory does not yet exist, create it. For example:
	`sudo mkdir -p /usr/share/java/`
	`cd mysql-connector-java-8.0.22`
	`sudo cp mysql-connector-java-8.0.22.jar /usr/share/java/mysql-connector-java.jar`

SLES	`sudo zypper install mysql-connector-java`
Ubuntu	`sudo apt-get install libmysql-java`

Installing the MySQL client

To use MySQL as a backend database for Hue, you must install the MySQL client and other required dependencies on all the Hue hosts based on your operating system.

For Cent OS

1. SSH into the Hue host as a root user.
2. Download the MySQL yum repository as follows:

```
curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el7-5.noarch.rpm
```

3. Install the package as follows:

```
rpm -ivh mysql80-community-release-el7-5.noarch.rpm
```

4. Install the required dependencies as follows:

```
yum install mysql-devel
yum install -y xmlsec1  xmlsec1-openssl
```

For MySQL version 8.0.27, add the mysql-community-client-8.0.25 client package as follows:

```
yum install mysql-community-client-8.0.25
```

5. Add the path where you installed the MySQL client and packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

6. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

For RHEL

1. SSH into the Hue host as a root user.
2. Download the MySQL yum repository as follows:

(RHEL 7)

```
curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el7-5.noa
rch.rpm
```

(RHEL 8)

```
curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el8-8.noa
rch.rpm
```

(RHEL 9)

```
curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el9-4.noa
rch.rpm
```

3. Install the package as follows:

(RHEL 7)

```
rpm -ivh mysql80-community-release-el7-5.noarch.rpm
```

(RHEL 8)

```
rpm -ivh mysql80-community-release-el8-8.noarch.rpm
```

(RHEL 9)

```
rpm -ivh mysql80-community-release-el9-4.noarch.rpm
```

4. Install the required dependencies as follows:

```
yum install mysql-devel
yum install -y xmlsec1  xmlsec1-openssl
```

5. Add the path where you installed the MySQL client and packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

6. Install the MySQL client as follows:

   (RHEL 8)

```
pip3.8 install mysqlclient
```

   (RHEL 9)

```
pip3.9 install mysqlclient
```

For SLES

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

```
zypper install libmysqlclient-devel
zypper install xmlsec1
zypper install xmlsec1-devel
zypper install xmlsec1-openssl-devel
```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

For Ubuntu

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

```
apt-get install libmysqlclient-dev
apt-get install -y xmlsec1
apt-get install libxmlsec1-openssl
```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

Creating Databases for Cloudera Software

Services that require databases

Create databases and service accounts for components that require databases:

- Cloudera Manager Server
- Cloudera Management Service roles:
  - Reports Manager
- Hue

- Each Hive metastore
- Oozie
- Schema Registry
- Streams Messaging Manager

Steps

1. Log in as the root user, or another user with privileges to create database and grant privileges:

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

```
Enter password:
```

2. Create databases for each service deployed in the cluster using the following commands. You can use any value you want for the <*DATABASE*>, <*USER*>, and <*PASSWORD*> parameters. The Databases for Cloudera Software table, below lists the default names provided in the Cloudera Manager configuration settings, but you are not required to use them.

   Configure all databases to use the utf8 character set.

   Include the character set for each database when you run the CREATE DATABASE statements described below.

```
CREATE DATABASE <DATABASE> DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8
_general_ci;
```

```
Query OK, 1 row affected (0.00 sec)
```

   Create USER by following the steps in this topic: CREATE USER Statement.

```
GRANT ALL ON <DATABASE>.* TO '<USER>'@'%' IDENTIFIED BY '<PASSWORD>';
```

```
Query OK, 0 rows affected (0.00 sec)
```

Table 18: Databases for Cloudera Software

Service	Database	User
Cloudera Manager Server	scm	scm
Reports Manager	rman	rman
Ranger RHEL/CentOS/Ubuntu	ranger	rangeradmin
Ranger KMS RHEL/CentOS	rangerkms	rangerkms
Hue	hue	hue
Hive Metastore Server	hive	hive
Oozie	oozie	oozie
Schema Registry	schemaregistry	schemaregistry
Streams Messaging Manager	smm	smm

3. Confirm that you have created all of the databases:

```
SHOW DATABASES;
```

   You can also confirm the privilege grants for a given user by running:

```
SHOW GRANTS FOR '<USER>'@'%';
```

4. 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.

Next Steps

- If you plan to use Apache Ranger, see the following topic for instructions on creating and configuring the Ranger database. See Configuring a Ranger or Ranger KMS Database: MySQL/MariaDB on page 132.
- If you plan to use Schema Registry or Streams Messaging Manager, see the following topic for instructions on configuring the database: Configuring the Database for Streaming Components on page 142.
- After you install and configure MySQL databases for Cloudera software, continue to Set up and Configure the Cloudera Manager Database to configure a database for Cloudera Manager.

Install and Configure MariaDB for Cloudera Software

You can install a MariaDB database for use with Cloudera Manager and other components that require a database.

To use a MariaDB database, follow these procedures. For information on compatible versions of the MariaDB database, see Database Requirements on page 28.

Installing MariaDB Server



Note:

- If you already have a MariaDB database set up, you can skip to the section Configuring and Starting the MariaDB Server on page 110 to verify that your MariaDB configurations meet the requirements for Cloudera Manager.
- It is important that the datadir directory (/var/lib/mysql by default), is on a partition that has sufficient free space. For more information, see Hardware Requirements on page 12.

1. Install MariaDB server:

OS	Command
RHEL compatible	`sudo yum install mariadb-server`
SLES	`sudo zypper install mariadb-server`
Ubuntu	`sudo apt-get install mariadb-server`

If these commands do not work, you might need to add a repository or use a different yum install command, particularly on RHEL compatible operating systems. For more assistance, see the following topics on the MariaDB website:

- RHEL compatible: Installing MariaDB with yum
- SLES: MariaDB Package Repository Setup and Usage
- Ubuntu: Installing MariaDB .deb Files

Configuring and Starting the MariaDB Server



Note: If you are making changes to an existing database, make sure to stop any services that use the database before continuing.

1. Stop the MariaDB server if it is running:

OS	Command
RHEL Compatible, SLES, and Ubuntu	`sudo systemctl stop mariadb`

2. If they exist, move old InnoDB log files /var/lib/mysql/ib_logfile0 and /var/lib/mysql/ib_logfile1 out of /var/lib/mysql/ to a backup location.
3. Determine the location of the option file, my.cnf (/etc/my.cnf by default).
4. Update my.cnf so that it conforms to the following requirements:

   - To prevent deadlocks, set the isolation level to READ-COMMITTED.
   - The default settings in the MariaDB installations in most distributions use conservative buffer sizes and memory usage. Cloudera Management Service roles need high write throughput because they might insert many records in the database. Cloudera recommends that you set the innodb_flush_method property to O_DI RECT.
   - Set the max_connections property according to the size of your cluster:
     - Fewer than 50 hosts - You can store more than one database (for example, both the Cloudera Manager Server and Reports Manager) on the same host. If you do this, you should:
       - Put each database on its own physical disk for best performance. You can do this by manually setting up symbolic links or running multiple database instances (each instance uses a different data directory path).
       - Allow 100 maximum connections for each database and then add 50 extra connections. For example, for two databases, set the maximum connections to 250. If you store four databases on one host (the databases for Cloudera Manager Server, Hue, Reports Manager, and Hive metastore), set the maximum connections to 450.
     - More than 50 hosts - Do not store more than one database on the same host. Use a separate host for each database/host pair. The hosts do not need to be reserved exclusively for databases, but each database should be on a separate host.
   - If the cluster has more than 1000 hosts, set the max_allowed_packet property to 16M. Without this setting, the cluster may fail to start due to the following exception: com.mysql.jdbc.PacketTooBigException.
   - Although binary logging is not a requirement for Cloudera Manager installations, it provides benefits such as MariaDB replication or point-in-time incremental recovery after a database restore. The provided example configuration enables the binary log. For more information, see The Binary Log.

   Here is an option file with Cloudera recommended settings:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
transaction-isolation = READ-COMMITTED
# Disabling symbolic-links is recommended to prevent assorted security r
isks;
# to do so, uncomment this line:
symbolic-links = 0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

key_buffer = 16M
key_buffer_size = 32M
max_allowed_packet = 32M
thread_stack = 256K
thread_cache_size = 64
query_cache_limit = 8M
query_cache_size = 64M
query_cache_type = 1

max_connections = 550
#expire_logs_days = 10
#max_binlog_size = 100M
#log_bin should be on a disk with enough free space.
```

```
#Replace '/var/lib/mysql/mysql_binary_log' with an appropriate path for yo
ur
#system and chown the specified folder to the mysql user.
log_bin=/var/lib/mysql/mysql_binary_log

#In later versions of MariaDB, if you enable the binary log and do not set
#a server_id, MariaDB will not start. The server_id must be unique within
#the replicating group.
server_id=1

binlog_format = mixed

read_buffer_size = 2M
read_rnd_buffer_size = 16M
sort_buffer_size = 8M
join_buffer_size = 8M
# InnoDB settings
innodb_file_per_table = 1
innodb_flush_log_at_trx_commit  = 2
innodb_log_buffer_size = 64M
innodb_buffer_pool_size = 4G
innodb_thread_concurrency = 8
innodb_flush_method = O_DIRECT
innodb_log_file_size = 512M
[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid
#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
```

5. If AppArmor is running on the host where MariaDB is installed, you might need to configure AppArmor to allow MariaDB to write to the binary.
6. Ensure the MariaDB server starts at boot:

OS	Command
RHEL Compatible, SLES, and Ubuntu	`sudo systemctl enable mariadb`

7. Start the MariaDB server:

OS	Command
RHEL Compatible, SLES, and Ubuntu	`sudo systemctl start mariadb`

8. Run /usr/bin/mysql_secure_installation to set the MariaDB root password and other security-related settings. In a new installation, the root password is blank. Press the Enter key when you're prompted for the root password. For the rest of the prompts, enter the responses listed below in bold:

```
sudo /usr/bin/mysql_secure_installation
```

```
[...]
Enter current password for root (enter for none):
OK, successfully used password, moving on...
[...]
Set root password? [Y/n] Y
New password:
Re-enter new password:
[...]
Remove anonymous users? [Y/n] Y
```

```
[...]
Disallow root login remotely? [Y/n] N
[...]
Remove test database and access to it [Y/n] Y
[...]
Reload privilege tables now? [Y/n] Y
[...]
All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

Installing the MySQL JDBC Driver for MariaDB

The MariaDB JDBC driver is not supported. Follow the steps in this section to install and use the MySQL JDBC driver instead.

Install the JDBC driver on the Cloudera Manager Server host, as well as any other hosts running services that require database access.

Cloudera recommends that you consolidate all roles that require databases on a limited number of hosts, and install the driver on those hosts. Locating all such roles on the same hosts is recommended but not required. Make sure to install the JDBC driver on each host running roles that access the database.



Note: Cloudera recommends using only version 8.0 of the JDBC driver.

OS	Command

RHEL	**Important:** Using the yum install command to install the MySQL driver package before installing a JDK installs OpenJDK, and then uses the Linux alternatives command to set the system JDK to be OpenJDK. If you intend to use an Oracle JDK, make sure that it is installed before installing the MySQL driver using yum install. If you want to use OpenJDK, you can install the driver using yum.
	
	Alternatively, use the following procedure to manually install the driver.
	
	1. Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html (in .tar.gz format). As of the time of writing, you can download version 8.0.22 using wget as follows:
	`wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.22.tar.gz`
	2. Extract the JDBC driver JAR file from the downloaded file. For example:
	`tar zxvf mysql-connector-java-8.0.22.tar.gz`
	3. Copy the JDBC driver, renamed, to /usr/share/java/. If the target directory does not yet exist, create it. For example:
	`sudo mkdir -p /usr/share/java/`
	`cd mysql-connector-java-8.0.22`
	`sudo cp mysql-connector-java-8.0.22-bin.jar /usr/share/java/mysql-connector-java.jar`
	
	Important:
	Resolving time zone connection errors with MySQL Connector or J 8.0.22
	
	MySQL Connector or J 8.0.22 validates server time zones against an internal list of recognized IANA time zones. If MariaDB reports its time zone as an abbreviation (such as CDT, EST, or EDT), connections from Cloudera Manager Server or other services using this driver will fail with an error similar to:
	`Caused by: com.mysql.cj.exceptions.InvalidConnectionAttributeException: The server time zone value 'CDT' is unrecognized or represents more than one time zone. You must configure either the server or JDBC driver (via the 'serverTimezone' configuration property) to use a more specific time zone value ...`
	
	While MySQL Connector or J 5.1 skipped this validation by default, version 8.0.22 enforces it. To resolve this issue while preserving the certified 8.0.22 driver, use one of the following methods:
	
	• **Method 1: Configure a JDBC URL override (Recommended - No database changes required)**
	
	Append &serverTimezone=<IANA_Time_Zone> to the JDBC connection string to specify the time zone explicitly (for example, America/Chicago or UTC).
	
	a. Cloudera Manager Server: In /etc/cloudera-scm-server/db.properties, update the connection URL property:
	`com.cloudera.cmf.orm.hibernate.connection.url=jdbc:mysql://<db-host>:3306/<db>?useSSL=false&serverTimezone=America/Chicago`
	
	b. After updating the property, restart the Cloudera Manager Server:
	`sudo systemctl restart cloudera-scm-server`
	
	c. Other Services: For other cluster services connecting to MariaDB using the 8.0.22 driver, add the serverTimezone parameter to their respective JDBC URL overrides (for example, in Cloudera Manager under Ranger configurations: Ranger Database JDBC Url Override).
	
	Method 2: Set a recognized time zone on the database server

Installing the MySQL client

To use MariaDB as a backend database for Hue, you must install the MySQL client and other required dependencies on all the Hue hosts based on your operating system.

For Cent OS

1. SSH into the Hue host as a root user.
2. Install the required dependencies as follows:

```
yum install -y xmlsec1  xmlsec1-openssl
```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

For RHEL

1. SSH into the Hue host as a root user.
2. Install the required dependencies as follows:

```
yum install -y python39-devel
yum install mysql-devel
yum install -y xmlsec1  xmlsec1-openssl
```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

(RHEL 8)
Python 3.8

```
python3.9 -m pip install mysqlclient
```

Python 3.9

```
python3.8 -m pip install mysqlclient
```

(RHEL 9)

```
pip3.9 install mysqlclient
```

For SLES

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

```
zypper install libmysqlclient-devel
zypper install xmlsec1
zypper install xmlsec1-devel
zypper install xmlsec1-openssl-devel
```

Attention: While installing the mysql-devel and libmysqlclient-devel packages on SLES15, use the "--replacefiles" zypper switch or manually enter yes on the interactive pop-up that you see when the files are being overwritten. Else, you may see an error such as: File /usr/bin/mariadb_config from install of MariaDB-devel-<version>.x86_64 conflicts with file from install of libmariadb-devel-3.1.21-150000.3.33.3.x86_64 (SLES Module Server Applications Updates).

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

For Ubuntu

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

```
apt-get install libmysqlclient-dev
apt-get install -y xmlsec1
apt-get install libxmlsec1-openssl
```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

Creating Databases for Cloudera Software

Services that require databases

Create databases and service accounts for components that require databases:

- Cloudera Manager Server
- Cloudera Management Service roles:
  - Reports Manager
- Hue
- Each Hive metastore
- Oozie
- Schema Registry
- Streams Messaging Manager

Steps

1. Log in as the root user, or another user with privileges to create database and grant privileges:

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

```
Enter password:
```

2. Create databases for each service deployed in the cluster using the following commands. You can use any value you want for the *<DATABASE>*, *<USER>*, and *<PASSWORD>* parameters. The Databases for Cloudera Software table, below lists the default names provided in the Cloudera Manager configuration settings, but you are not required to use them.

Configure all databases to use the utf8 character set.

Include the character set for each database when you run the CREATE DATABASE statements described below.

```
CREATE DATABASE <DATABASE> DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8
_general_ci;
```

```
Query OK, 1 row affected (0.00 sec)
```

Create USER by following the steps in this topic: CREATE USER Statement.

```
GRANT ALL ON <DATABASE>.* TO '<USER>'@'%' IDENTIFIED BY '<PASSWORD>';
```

```
Query OK, 0 rows affected (0.00 sec)
```

Table 19: Databases for Cloudera Software

Service	Database	User
Cloudera Manager Server	scm	scm
Reports Manager	rman	rman
Ranger RHEL/CentOS/Ubuntu	ranger	rangeradmin
Ranger KMS RHEL/CentOS	rangerkms	rangerkms
Hue	hue	hue
Hive Metastore Server	hive	hive
Oozie	oozie	oozie
Schema Registry	schemaregistry	schemaregistry
Streams Messaging Manager	smm	smm

3. Confirm that you have created all of the databases:

```
SHOW DATABASES;
```

You can also confirm the privilege grants for a given user by running:

```
SHOW GRANTS FOR '<USER>'@'%';
```

4. 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.

Next Steps

- If you plan to use Apache Ranger, see the following topic for instructions on creating and configuring the Ranger database. See Configuring a Ranger or Ranger KMS Database: MySQL/MariaDB on page 132.
- If you plan to use Schema Registry or Streams Messaging Manager, see the following topic for instructions on configuring the database: Configuring the Database for Streaming Components on page 142
- After you install and configure MariaDB databases for Cloudera software, continue to Set up and Configure the Cloudera Manager Database to configure a database for Cloudera Manager.

Configure Oracle Database for Cloudera Software

You can configure an external Oracle database for use with Cloudera Manager and other components that require a database.

To use an Oracle database, follow these procedures. For information on compatible versions of the Oracle database, see Database Requirements on page 28.

Collecting Oracle Database Information

To configure Cloudera Manager to work with an Oracle database, get the following information from your Oracle DBA:

- Hostname - The DNS name or the IP address and the port of the host where the Oracle database is installed.
- Port - The port number on which Oracle server listens.
- SID - The name of the schema that will store Cloudera Manager information.
- Username - A username for each schema that is storing information. You could have four unique usernames for the four schema.
- Password - A password corresponding to each username.
- Connection string of the DB (usually in format of USERNAME:PASSWORD@//HOSTNAME:PORT/SID)

Ensure that your customer defines the table it will create (see below) using UTF-8.

Installing the Oracle JDBC Connector

You must install the JDBC connector on the Cloudera Manager Server host and any other hosts that use a database. The JDBC connector MUST be supplied by the Oracle DBA to ensure it matches the Oracle DB server release.

Cloudera recommends that you assign all roles that require a database on the same host and install the connector on that host. Locating all such roles on the same host is recommended but not required. If you install a role, such as Reports Manager, on one host and other roles on a separate host, you would install the JDBC connector on each host running roles that access the database.

Alternatively, you can perform the following steps to distribute the Oracle JDBC connector file to all the nodes on the same directory:

1. Obtain the Oracle JDBC Driver from the Oracle DBA. For more information about compatibility versions, see Oracle Database JDBC driver and Companion Jars Downloads.
2. Cloudera recommends using ojdbc11.jar when deploying Oracle 21c across JDK 11, JDK 17, and JDK 21 environments, or when deploying Oracle 26ai (also known as Oracle 23c or 23ai) on JDK 11 or JDK 21. For Oracle 26ai or 23c deployments running on JDK 17 or JDK 21, use ojdbc17.jar.
3. Copy the Oracle JDBC JAR file as root user to /usr/share/java/oracle-connector-java.jar on the relevant nodes. The Cloudera Manager databases and the Hive Mestastore database use this shared filename and location. For example:

```
sudo mkdir -p /usr/share/java
sudo cp /tmp/ojdbc10.jar /usr/share/java/oracle-connector-java.jar
sudo chmod 644 /usr/share/java/oracle-connector-java.jar
```

Creating Databases for Cloudera Software

Provide the Oracle DBA with the information for creating the Create schema and user accounts for components that require databases (depending on the type of services that you install on the Cloudera Base on premises Cluster).

To ease the tracking, provide a prefix name for each schema you define based on the environment you build such as cdp_**tst**_db_xxx for test environment, cdp_**prd**_db_xxx for production etc. For more information about a prefix name, see the following table:

Table 20: Databases for Cloudera Private Cloud Base Cluster

Service	Database	Username
Cloudera Manager Server	cdp_xxx_db_scm	cdp_xxx_scm
Reports Manager	cdp_xxx_db_rman	cdp_xxx_rman
Hive Metastore	cdp_xxx_db_hive	cdp_xxx_hive

Service	Database	Username
Ranger	cdp_xxx_db_ranger	cdp_xxx_rangeradmin
Ranger KMS	cdp_xxx_db_rangerkms	cdp_xxx_rangerkms
Hue	cdp_xxx_db_hue	cdp_xxx_hue
Schema Registry (only if used)	cdp_xxx_db_schemaregistry	cdp_xxx_schemaregistry
Streams Messaging Manager (only if used)	cdp_xxx_db_smm	cdp_xxx_smm



Important: For information about creating the relevant schema DBs, see Creating the relevant schema DBs.

Note that Yarn Queue Manager schema in Cloudera works only with either internal DB or external PostgreSQL, so you will need to configure it separately.

You can create the Oracle database, schema and users on the host where the Cloudera Manager Server will run, or on any other hosts in the cluster. For performance reasons, you should install each database on the host on which the service runs, as determined by the roles you assign during installation or upgrade. In larger deployments or in cases where database administrators are managing the databases the services use, you can separate databases from services, but use caution.

The databases must be configured to support UTF-8 character set encoding.

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.



Note: If you are deploying Oracle RAC (for high availability), enter the database name using the following format:

```
(DESCRIPTION=(LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=5)(
TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS=(PROTOCOL=TCP)(H
OST=[***HOSTNAME***])(PORT=[***PORT***]))(CONNECT_DATA=(SERVICE_NAM
E=[***SERVICE_NAME***])))
```

Provide the Oracle DBA with a list of commands to run in order to create the above schema,users and passwords on the Oracle server. Make sure to ask the DBA to provide you back the password created for each user/schema.

Note to request the DBA to set the quota to unlimited to each user as follows::

```
ALTER USER <USER> quota unlimited on <TABLESPACE>;
```

Also, run the following command to request the DBA to set on the SYS DB of the Oracle, to ensure there are no errors in connecting to Oracle from the Cloudera DBs:

```
ALTER SYSTEM set open_cursors=1000 scope=both;
```

Creating the relevant schema DBs

The format of the user/schema table is built using “xxx” where xxx should be replaced by either dev/tst/prd/dr etc. (for different environments separation on the Oracle DB)

Also, the “password” that is marked bold should be updated with a unique password given by the Oracle DBA per the security regulation of the Oracle DB server. That password will be used for connecting to the schema and will be configured on the Cloudera Manager UI later.



Important:

Unless stated otherwise, each DB is given GRANTs based on the need. The only exception is RMAN which needs GRANT ALL.



Important:

In the following commands, you should replace the **XXX** with the string for the relevant environment, and make sure the password string is replaced with the unique password string generated by Oracle DBA during the creation of the user.

You can create the relevant schema DBs by running the following set of commands:

Cloudera Manager Server DB

```
CREATE DATABASE cdp_xxx_db_scm CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_scm'@'%' IDENTIFIED BY
 'cdp_xxx_scm_password';
GRANT SELECT_CATALOG_ROLE, CONNECT, RESOURCE ON cdp_xxx_db_scm.*
 TO cdp_xxx_scm'@'%';
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,CR
EATE SEQUENCE,CREATE TRIGGER,UNLIMITED TABLESPACE ON cdp_xxx_db_
scm.* TO cdp_xxx_scm'@'%';
ALTER USER cdp_xxx_scm DEFAULT TABLESPACE cdp_xxx_db_scm;
ALTER USER cdp_xxx_scm quota unlimited on cdp_xxx_db_scm;
```

Reports Manager DB



Important: Reports Manager (rman) needs GRANT ALL PRIVILEGES on its DB.

```
CREATE DATABASE cdp_xxx_db_rman CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_rman'@'%' IDENTIFIED BY
 'cdp_xxx_rman_password';
GRANT ALL PRIVILEGES ON cdp_xxx_db_rman.* TO 'cdp_xxx_rman'@'%';
```

Hive Metastore DB

```
CREATE DATABASE cdp_xxx_db_hive CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_hive'@'%' IDENTIFIED BY
 'cdp_xxx_hive_password';
GRANT SELECT_CATALOG_ROLE, CONNECT, RESOURCE ON cdp_xxx_db_hive.*
 TO cdp_xxx_hive'@'%';
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,C
REATE SEQUENCE,CREATE TRIGGER,UNLIMITED TABLESPACE ON cdp_xxx_db_
hive.* TO cdp_xxx_hive'@'%';
ALTER USER cdp_xxx_hive DEFAULT TABLESPACE cdp_xxx_db_hive;
ALTER USER cdp_xxx_hive quota unlimited on cdp_xxx_db_hive;
```

Ranger DB

```
CREATE DATABASE cdp_xxx_db_ranger CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_rangeradmin'@'%' IDENTIFIED BY
 'cdp_xxx_rangeradmin_password';
GRANT SELECT_CATALOG_ROLE, CONNECT, RESOURCE TO cdp_xxx_rangera
dmin'@'%';
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,CR
EATE SEQUENCE,CREATE PUBLIC SYNONYM,CREATE ANY SYNONYM,CREATE TR
IGGER,UNLIMITED TABLESPACE TO cdp_xxx_rangeradmin'@'%';
ALTER USER cdp_xxx_rangeradmin DEFAULT TABLESPACE cdp_xxx_db_rang
eradmin;
ALTER USER cdp_xxx_rangeradmin quota unlimited on cdp_xxx_db_rang
eradmin;
```



Important:

CREATE_PUBLIC_SYNONYM and CREATE_ANY_SYNONYM are optional and not mandatory.

Ranger KMS DB

```
CREATE DATABASE cdp_xxx_db_rangerkms CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_rangerkms'@'%' IDENTIFIED BY
 'cdp_xxx_rangerkms_password';
GRANT SELECT_CATALOG_ROLE, CONNECT, RESOURCE TO cdp_xxx_rangerk
ms'@'%';
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,
CREATE SEQUENCE,CREATE PUBLIC SYNONYM,CREATE ANY SYNONYM,CREATE
TRIGGER,UNLIMITED TABLESPACE TO cdp_xxx_rangerkms'@'%';
ALTER USER cdp_xxx_rangerkms DEFAULT TABLESPACE cdp_xxx_db_rang
erkms;
ALTER USER cdp_xxx_rangerkms quota unlimited on cdp_xxx_db_rang
erkms;
```

Important:

CREATE_PUBLIC_SYNONYM and CREATE_ANY_SYNONYM are optional and not mandatory.

Hue DB

```
CREATE DATABASE cdp_xxx_db_hue CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_hue'@'%' IDENTIFIED BY
 'cdp_xxx_hue_password';
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,CR
EATE SEQUENCE,CREATE TRIGGER,UNLIMITED TABLESPACE TO cdp_xxx_hue
'@'%';
ALTER USER cdp_xxx_hue DEFAULT TABLESPACE cdp_xxx_db_hue;
ALTER USER cdp_xxx_hue quota unlimited on cdp_xxx_db_hue;
```

STREAMS MESSAGING MANAGER (SMM) / SCHEMA REGISTRY

```
CREATE TABLESPACE cdp_xxx_db_schemaregistry ONLINE;
CREATE TABLESPACE cdp_xxx_db_smm ONLINE;
CREATE USER 'cdp_xxx_schemaregistry'@'%' IDENTIFIED BY
 'cdp_xxx_schemaregistry_password';
CREATE USER 'cdp_xxx_smm'@'%' IDENTIFIED BY
 'cdp_xxx_smm_password';

ALTER USER cdp_xxx_schemaregistry DEFAULT TABLESPACE cdp_xxx_db_
schemaregistry;
ALTER USER cdp_xxx_smm DEFAULT TABLESPACE cdp_xxx_db_smm;

GRANT CONNECT, CREATE PROCEDURE,CREATE TABLE,CREATE SEQUENCE,C
REATE INDEX,ALTER PROCEDURE,ALTER TABLE,ALTER SEQUENCE,ALTER IND
EX,DROP PROCEDURE,DROP TABLE,DROP SEQUENCE,DROP INDEX,UNLIMITED
TABLESPACE TO cdp_xxx_schemaregistry'@'%';

GRANT CONNECT, CREATE PROCEDURE,CREATE TABLE,CREATE SEQUENCE,CREA
TE INDEX,ALTER PROCEDURE,ALTER TABLE,ALTER SEQUENCE,ALTER INDEX,
DROP PROCEDURE,DROP TABLE,DROP SEQUENCE,DROP INDEX,UNLIMITED TAB
LESPACE TO cdp_xxx_smm'@'%';
```

Important: The new users for smm/schema registry need to have permissions to create or drop tables and other database structures because when the Schema Registry or SMM service starts, it automatically creates the database structure. If the database structure already exists, then the service alters it as necessary.

```
FLUSH PRIVILEGES;
```

For further information about Oracle privileges, see Authorization: Privileges, Roles, Profiles, and Resource Limitations.

Next Steps

- If you plan to use **Apache Ranger**, see the following topic for instructions on creating and configuring the Ranger database and to install the JDBC driver for the database. See Configuring a Ranger or Ranger KMS Database: Oracle on page 134.
- If you plan to use **Schema Registry** or **Streams Messaging Manager**, see the following topic for instructions on configuring the database: Configuring the Database for Streaming Components on page 142
- After you install and configure Oracle databases for Cloudera software, continue to Set up and Configure the Cloudera Manager Database to configure a database for Cloudera Manager.

Enabling TLS 1.2 on Database Server

Perform the following steps for managing the configuration of the Database Server and setting up TLS 1.2 connections to ensure the secure and proper functioning of the database environment.

Enabling TLS 1.2 for MySQL Database Server

TLS 1.2 encrypts the connection between the MySQL server and the Cloudera Manager server. You must enable TLS 1.2 for the MySQL database before setting up Cloudera Manager and add the MySQL root Certificate Authorities (CA) to the Cloudera Manager truststore.

Procedure

1. SSH into the MySQL database host.
2. Start the MySQL server:

```
service mysqld start
```

3. Establish an encrypted connection with the client:

```
mysql -p --ssl-mode=required
```

4. Verify whether TLS 1.2 is enabled on MySQL by running the following command:

```
mysql> show global variables like '%ssl%';
```

If TLS 1.2 is enabled, you see the value of have_ssl equal to YES, as follows. Otherwise, you see the value of have_ssl equal to DISABLED:

```
+---------------+----------+
| Variable_name | Value    |
+---------------+----------+
| have_openssl  | YES      |
| have_ssl      | YES      |
| ...           | ...      |
```

If TLS 1.2 is enabled, then you can skip the following steps and go to Importing the MySQL root certificate.

5. Create a certificate authority by running the following commands:

```
mkdir /etc/my.cnf.d/ssl/
cd /etc/my.cnf.d/ssl/
openssl genrsa 2048 > ca-key.pem
```

6. Create a certificate for the server using the CA certificate generated earlier by running the following command:

```
openssl req -new -x509 -nodes -days 365000 -key ca-key.pem -out ca-cert.
pem
openssl req -newkey rsa:2048 -days 365 -nodes -keyout server-key.pem -out
 server-req.pem
```

```
openssl rsa -in server-key.pem -out server-key.pem
```

7. Create a certificate for the clients using the same CA certificate by running the following command:

```
openssl x509 -req -in server-req.pem -days 365 -CA ca-cert.pem -CAkey ca-
key.pem -set_serial 01 -out server-cert.pem
```

8. Add the following lines in the /etc/my.cnf.d/server.cnf file under the [mysqld] section:

```
ssl-ca=/etc/my.cnf.d/ssl/ca-cert.pem
ssl-cert=/etc/my.cnf.d/ssl/server-cert.pem
ssl-key=/etc/my.cnf.d/ssl/server-key.pem
bind-address=*
```

You can view the content of the server.cnf file by running the following command:

```
vim /etc/my.cnf.d/server.cnf
```

9. Restart the MySQL server:

```
service mysqld restart
```

10. Check the TLS 1.2 status by running the following commands:

```
mysql -p --ssl-mode=required
> SHOW VARIABLES LIKE '%ssl%';
> status
```

Sample output:

```
> SHOW VARIABLES LIKE '%ssl%';
+---------------------------------------+-----------------+
| Variable_name                         | Value           |
+---------------------------------------+-----------------+
| admin_ssl_ca                          |                 |
| admin_ssl_capath                      |                 |
| admin_ssl_cert                        |                 |
| admin_ssl_cipher                      |                 |
| admin_ssl_crl                         |                 |
| admin_ssl_crlpath                     |                 |
| admin_ssl_key                         |                 |
| have_openssl                          | YES             |
| have_ssl                              | YES             |
| mysqlx_ssl_ca                         |                 |
| mysqlx_ssl_capath                     |                 |
| mysqlx_ssl_cert                       |                 |
| mysqlx_ssl_cipher                     |                 |
| mysqlx_ssl_crl                        |                 |
| mysqlx_ssl_crlpath                    |                 |
| mysqlx_ssl_key                        |                 |
| performance_schema_show_processlist | OFF             |
| ssl_ca                                | ca.pem          |
| ssl_capath                            |                 |
| ssl_cert                              | server-cert.pem |
| ssl_cipher                            |                 |
| ssl_crl                               |                 |
| ssl_crlpath                           |                 |
| ssl_fips_mode                         | OFF             |
| ssl_key                               | server-key.pem  |
+---------------------------------------+-----------------+

> status
```

```
SSL:   Cipher in use is ECDHE-RSA-AES128-GCM-SHA256
```

Enabling TCPS for Oracle Database Server

You must enable TCPS for the Oracle database before setting up Cloudera Manager. Enabling TCPS establishes a secure channel between the client (Cloudera Manager) and the server (Oracle Database Server).

Procedure

1. SSH into the Oracle database server host.
2. Change to the "oracle" user as follows:

```
sudo -su oracle
```

3. Append the location of ORACLE_HOME to the PATH environment variable by running the following commands:

```
export ORACLE_HOME=/opt/oracle/product/19c/dbhome_1
export PATH=${PATH}:${ORACLE_HOME}/bin
```

4. Create an auto-login wallet by running the following command:

```
orapki wallet create -wallet /opt/oracle/product/19c/dbhome_1/wallet -au
to_login
```

   An auto-login wallet uses SSL's single sign-on functionality. The users do not need to specify password each time they open the wallet.

5. Add a self-signed certificate to this wallet by running the following command:

```
orapki wallet add -wallet /opt/oracle/product/19c/dbhome_1/wallet -dn "C
N=server" -keysize 4096 -self_signed -validity 365
```

6. Export the certificate from the Oracle wallet by running the following command:

```
orapki wallet export -wallet /opt/oracle/product/19c/dbhome_1/wallet -dn
 "CN=server" -cert server_ca.cert
```

   This exports a certificate with the subject's distinguished name (-dn) (CN=server) from a wallet to the file that is specified by -cert (server_ca.cert).

7. Add the following lines to the /opt/oracle/product/19c/dbhome_1/network/admin/listener.ora configuration file:

```
SSL_CLIENT_AUTHENTICATION = FALSE
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /opt/oracle/product/19c/dbhome_1/wallet)
    )
  )
Register a new address in LISTENER:
(ADDRESS = (PROTOCOL = TCPS)(HOST = [***HOST***])(PORT = 2484))
```

8. Add the following lines to the /opt/oracle/product/19c/dbhome_1/network/admin/sqlnet.ora profile configuration file:

```
SSL_CLIENT_AUTHENTICATION = FALSE
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /opt/oracle/product/19c/dbhome_1/wallet)
```

```
    )
  )
```

9. Add the following lines to the /opt/oracle/product/19c/dbhome_1/network/admin/tnsnames.ora configuration file:

```
ORCLPDB1_SSL =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCPS)(HOST = [***HOST***])(PORT = 2484))
      (CONNECT_DATA =
        (SERVER = DEDICATED)
        (SERVICE_NAME = ORCLPDB1)
      )
      (SECURITY =
        (MY_WALLET_DIRECTORY = /opt/oracle/product/19c/dbhome_1/wallet)
      )
    )
```

10. Restart the listener by running the following commands:

```
lsnrctl stop
lsnrctl start
```

11. Check the TCPS status by running the following command

```
sqlplus cm/cmverystr0ngP4ss@ORCLPDB1_SSL
SELECT sys_context('USERENV', 'NETWORK_PROTOCOL') as network_protocol F
ROM dual;
```

Sample output:

```
NETWORK_PROTOCOL
------------------------------------------------------------------------
---------
tcps
```

Enabling TLS 1.2 for MariaDB Database Server

TLS 1.2 encrypts the connection between the MariaDB server and the Cloudera Manager server. You must enable TLS 1.2 for the MariaDB database before setting up Cloudera Manager and add the MariaDB root Certificate Authorities (CA) to the Cloudera Manager truststore.

Procedure

1. SSH into the MariaDB database host.
2. Start the MariaDB server:

```
service mysqld start
```

3. Establish an encrypted connection with the client:

```
mysql -p --ssl=true
```

4. Verify whether TLS 1.2 is enabled on MariaDB by running the following command:

```
mysql> show global variables like '%ssl%';
```

If TLS 1.2 is enabled, you see the value of have_ssl equal to YES, as follows. Otherwise, you see the value of have_ssl equal to DISABLED:

```
+---------------+----------+
| Variable_name | Value    |
+---------------+----------+
```

```
| have_openssl  | YES      |
| have_ssl      | YES      |
| ...           | ...      |
```

If TLS 1.2 is enabled, then you can skip the following steps and go to Importing the MariaDB root certificate.

5. Create a certificate authority by running the following commands:

```
mkdir /etc/my.cnf.d/ssl/
cd /etc/my.cnf.d/ssl/
openssl genrsa 2048 > ca-key.pem
```

6. Create a certificate for the server using the CA certificate generated earlier by running the following command:

```
openssl req -new -x509 -nodes -days 365000 -key ca-key.pem -out ca-cert.
pem
openssl req -newkey rsa:2048 -days 365 -nodes -keyout server-key.pem -out
 server-req.pem
openssl rsa -in server-key.pem -out server-key.pem
```

7. Create a certificate for the clients using the same CA certificate by running the following command:

```
openssl x509 -req -in server-req.pem -days 365 -CA ca-cert.pem -CAkey ca-
key.pem -set_serial 01 -out server-cert.pem
```

8. Add the following lines in the /etc/my.cnf.d/server.cnf file under the [mysqld] section:

```
ssl-ca=/etc/my.cnf.d/ssl/ca-cert.pem
ssl-cert=/etc/my.cnf.d/ssl/server-cert.pem
ssl-key=/etc/my.cnf.d/ssl/server-key.pem
bind-address=*
```

You can view the content of the server.cnf file by running the following command:

```
vim /etc/my.cnf.d/server.cnf
```

9. Run the following commands to change the permission and ownership of the server-key.pem file:

```
sudo chown mysql:mysql /etc/my.cnf.d/ssl/server-key.pem
sudo chmod 600 /etc/my.cnf.d/ssl/server-key.pem
```

10. Restart the MariaDB server:

```
service mysqld restart
```

11. Check the TLS 1.2 status by running the following commands:

```
mysql -p --ssl=true
> SHOW VARIABLES LIKE '%ssl%';
> status
```

Sample output:

```
> SHOW VARIABLES LIKE '%ssl%';
+---------------------+----------------------------------+
| Variable_name       | Value                            |
+---------------------+----------------------------------+
| have_openssl        | YES                              |
| have_ssl            | YES                              |
| ssl_ca              | /etc/my.cnf.d/ssl/ca-cert.pem    |
| ssl_capath          |                                  |
| ssl_cert            | /etc/my.cnf.d/ssl/server-cert.pem |
```

```
| ssl_cipher          |                                    |
| ssl_crl             |                                    |
| ssl_crlpath         |                                    |
| ssl_key             | /etc/my.cnf.d/ssl/server-key.pem   |
| version_ssl_library | OpenSSL 1.0.2k-fips  26 Jan 2017   |
+---------------------+------------------------------------+

> status
SSL:   Cipher in use is DHE-RSA-AES256-GCM-SHA384
```

Enabling TLS 1.2 for PostgreSQL Database Server

TLS 1.2 encrypts the connection between the PostgreSQL server and the Cloudera Manager server. You must enable TLS 1.2 for the PostgreSQL database before setting up Cloudera Manager.

Before you begin



Important:

Depending on your PostgreSQL installation, the PostgreSQL version number and the exact paths might vary in your environment. Ensure to adjust the commands according to the PostgreSQL version number. For more information about supported PostgreSQL versions, see Cloudera Support Matrix

Procedure

1. SSH into the PostgreSQL database host.
2. Start the PostgreSQL server by running the following command:

   ```
   systemctl start postgresql-14
   ```

3. Verify whether TLS 1.2 is enabled on PostgreSQL by running the following command:

   ```
   SHOW ssl;
   ```

   If TLS 1.2 is enabled, you see the value of ssl equal to on, as follows:

   ```
   ssl
   -----
    on
   (1 row)
   ```

   If TLS 1.2 is enabled, then you can skip the following steps and go to Importing the PostgreSQL root certificate.
4. Create a certificate authority by running the following commands:

   ```
   cd /var/lib/pgsql/14/data
   openssl genrsa -des3 -out server.key 1024
   openssl rsa -in server.key -out server.key
   chmod 400 server.key
   ```

5. Create a certificate for the server using the CA certificate generated earlier by running the following command:

   ```
   openssl req -new -key server.key -days 3650 -out server.crt -x509 -subj '/
   CN=hostname'
   ```

6. Change the ownership and permissions of the files by running the following commands:

   ```
   chown postgres server.crt server.key
   chmod 400 server.key server.crt
   ```

7. Go to /var/lib/pgsql/14/data and open the postgresql.conf file. To enable and enforce TLS 1.2, add or update the following parameters in the postgresql.conf file:

```
ssl = on
ssl_cert_file = '/var/lib/pgsql/14/data/server.crt'
ssl_key_file = '/var/lib/pgsql/14/data/server.key'
# Add the following lines to enforce TLS 1.2:
ssl_min_protocol_version = 'TLSv1.2'
ssl_max_protocol_version = 'TLSv1.2'
```



Important:

Explicitly setting both the minimum and maximum versions to TLSv1.2 ensures compliance with the TLS 1.2 requirement and prevents the use of older, insecure protocols (TLS 1.0 or TLS 1.1) or newer protocols (TLS 1.3) that might not be desired in a strict TLS 1.2-only environment.

8. Update the pg_hba.conf file to enforce SSL connections.

   To ensure PostgreSQL requires SSL for incoming connections from remote services (such as Hive), you must add a hostssl entry to the Host Based Authentication configuration.

   a) Open the pg_hba.conf file (typically located in the same data directory as postgresql.conf).

   b) Add a line to allow/enforce SSL connections for your network. For example, to allow all remote hosts to connect through SSL using SCRAM-SHA-256:

```
cat <<EOF >> /var/lib/pgsql/14/data/pg_hba.conf
# TYPE  DATABASE        USER            ADDRESS                 METHOD
hostssl all             all             0.0.0.0/0               scram-sh
a-256
EOF
```



Important: Adjust the address (0.0.0.0/0) to match your specific network security requirements if you wish to restrict access to specific subnets.

   c) Save the pg_hba.conf file.

9. Restart the PostgreSQL server by running the following command:

```
systemctl restart postgresql-14.service
# OR
sudo -u postgres pg_ctl reload
```

10. Check the TLS 1.2 status by running the following commands:

```
SELECT name, setting
FROM pg_settings
WHERE name LIKE '%ssl%';
```

Sample output:

```
+---------------------------------------+----------------------------+
| name                                  | setting                    |
+---------------------------------------+----------------------------+
| ssl                                   | on                         |
| ssl_ca_file                           | server.crt                 |
| ssl_cert_file                         | server.crt                 |
| ssl_ciphers                           | HIGH:MEDIUM:+3DES:!aNULL   |
| ssl_crl_dir                           |                            |
| ssl_crl_file                          |                            |
| ssl_dh_params_file                    |                            |
| ssl_ecdh_curve                        | prime256v1                 |
| ssl_key_file                          | server.key                 |
| ssl_library                           | OpenSSL                    |
```

```
| ssl_max_protocol_version                | |                          |
| ssl_min_protocol_version                | TLSv1.2                    |
| ssl_passphrase_command                  | |                          |
| ssl_passphrase_command_supports_reload | off                        |
| ssl_prefer_server_ciphers               | on                         |

+-----------------------------------------+----------------------------+
(15 rows)
```

Important: To change the default setting for ssl_ciphers from HIGH:MEDIUM:+3DES:!aNULL to HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!kRSA:!PSK:AESGCM. Add the following line to the postgresql.conf file:

```
ssl_ciphers= 'HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!kRSA:!PSK:A
ESGCM'
```

Restart the PostgreSQL server to check the new ssl_ciphers is reflected in the sample output.

Enabling Kerberos (MIT and AD) authentication for MariaDB Database Server

Perform the following steps for enabling Kerberos authentication on MariaDB Database Server and to connect Cloudera Manager Server to Kerberos enabled MariaDB. These steps are applicable for TLS 1.2 and non-TLS 1.2 clusters.

Procedure

1. SSH into the MariaDB database host.
2. Run the following command to install the auth_gssapi.so plugin:

```
sudo yum install MariaDB-gssapi-server
```

Ensure the plugin is present in the following directory /usr/lib64/mysql/plugin.

3. On the MariaDB server, create Service Principal and Keytab by running the following commands:

   a. **For MIT Kerberos**

   Run the following command to create Service Principal:

```
kadmin -p root/admin -q "addprinc -randkey mariadb/${HOST}"
```

   Note: For example, kadmin -p root/admin -q "addprinc -randkey mariadb/mariaa-1.mariaa.root.hwx.site".

   For AD Kerberos

   Run the following command on AD server to create Service Principal:

```
dsadd user CN=mariadb,CN=Users,DC=qe-infra-ad,DC=cloudera,DC=com
 -pwd Test123 -samid mariadb -upn mariadb@QE-INFRA-AD.CLOUDERA.C
OM
```

   b. **For MIT Kerberos**

   Run the following command to create Keytab:

```
kadmin -p root/admin -q "ktadd -k /path/to/mariadb.keytab mariad
b/${HOST}"
```

   For AD Kerberos

Run the following command on AD server to create Keytab:

```
ktpass.exe /princ mariadb@QE-INFRA-AD.CLOUDERA.COM /mapuser mari
adb@QE-INFRA-AD.CLOUDERA.COM /pass Test123 /out mariadb.keytab /
crypto all /ptype KRB5_NT_PRINCIPAL /mapop set
```

Copy the Keytab file to the database host.

c. For AD Kerberos only

Run the following command to map service principal name to the principal:

```
setspn -s mariadb/QE-INFRA-AD.CLOUDERA.COM mariadb
```

d. Add the Service Principal and Keytab to the /etc/my.cnf configuration file:

```
gssapi_keytab_path=/path/to/mariadb.keytab
gssapi_principal_name=service_principal_name/host.domain.com@REALM
```

4. You must install the plugin on the Mariadb (database) side. You can do this either by following Step 4. a. or Step 4. b.:

a. Run the following query on Mariadb database:

```
INSTALL SONAME 'auth_gssapi';
```

b. Add the following plugin to the /etc/my.cnf configuration file by running the following command:

```
plugin_load_add = auth_gssapi
```

Restart the MariaDB server by running the following command:

```
sudo systemctl restart mariadb
```

After performing Step 4. a. or Step 4. b., the plugin is visible in the plugin table. To verify the plugin, run the following query:

```
SHOW PLUGINS;
```

5. Create user principals and do kinit on the Cloudera Manager Server by running the following commands:

For MIT Kerberos

a. Create user principal by running the following command:

```
kadmin -p root/admin -q "addprinc cm_kerb_user"
```

b. Run the following command to create Keytab for the user in MIT:

```
kadmin -p root/admin -q "ktadd -k /mdbktb/cm_kerb_user.keytab
 cm_kerb_user"
```

c. Do kinit by running the following command:

```
kinit cm_kerb_user
```

For AD Kerberos

a. Run the following command to create user in Active Directory:

```
dsadd user CN=cm_kerb_user,CN=Users,DC=qe-infra-ad,DC=cloude
ra,DC=com -pwd Test123 -samid cm_kerb_user -upn cm_kerb_user
@QE-INFRA-AD.CLOUDERA.COM
```

b. Run the following command to create Keytab for the user in Active Directory:

```
ktpass.exe /princ cm_kerb_user@QE-INFRA-AD.CLOUDERA.COM /map
user cm_kerb_user@QE-INFRA-AD.CLOUDERA.COM /pass Test123 /ou
t cm_kerb_user.keytab /crypto all /ptype KRB5_NT_PRINCIPAL /
mapop set
```

c. Run the following command to map service principal name to the principal in Active Directory:

```
setspn -s cm_kerb_user/QE-INFRA-AD.CLOUDERA.COM cm_kerb_user
```

d. Copy the Keytab file to the Cloudera Manager Server host.

e. Do kinit by running the following command in the Cloudera Manager Server host:

```
sudo /usr/bin/kinit -kt /cdep/keytabs/cm_kerb_user.keytab -l
 1d -r 8d cm_kerb_user
```

6. On MariaDB Database, you must create the user for using the Kerberos authentication by running the following commands:

```
CREATE USER 'cm_kerb_user'@'%' IDENTIFIED WITH gssapi;
```

```
GRANT ALL PRIVILEGES ON cm.* TO 'cm_kerb_user'@'%' IDENTIFIED VIA gssapi;
```

7. Run the following command to log in the user using Kerberos and verify whether the Kerberos configuration and user are working correctly:

```
 mysql --plugin-dir=/usr/lib64/mysql/plugin --user=cm_kerb_user --host=h
ostname
```

8. Connect Cloudera Manager Server to Kerberos enabled MariaDB by performing the following steps:

a. By default mysql-connector-java.jar is available. You must have MariaDB Connector/J JDBC driver for building Java applications on top of MariaDB such as mariadb-java-client-3.1.4.jar.

Add mariadb-java-client-3.1.4.jar file to /usr/share/java location.



Important: Before you proceed to Step 8. b., you must install Cloudera Manager packages by performing the steps from Step 3: Install Cloudera Manager Server.

b. Add the following line to the /etc/default/cloudera-scm-server file.

```
export CMF_JDBC_DRIVER_JAR="${CMF_JDBC_DRIVER_JAR}:/usr/share/java/maria
db-java-client-3.3.3.jar"
```

c. Create a /etc/jaas.conf file with the following content. Also, for the keyTab section, add the correct directory location for the cm_kerb_user.keytab keytab:

```
Krb5ConnectorContext {
            com.sun.security.auth.module.Krb5LoginModule required
            useKeyTab=true
            keyTab="/mdbktb/cm_kerb_user.keytab"
            principal="cm_kerb_user@ROOT.COMOPS.SITE"
            doNotPrompt=true;
        };
```

d. For TLS cluster

Update the JDBC URL in /etc/cloudera-scm-server/db.properties as follows:

Important: Do the following modifications to the original JDBC URL:

- Modify from user=cm to user=cm_kerb_user
- Remove useSSL=true and add sslMode=PREFERRED

The updated URL should look like this:

```
com.cloudera.cmf.orm.hibernate.connection.url=jdbc:mysql://local
host:3306/cm?user=cm_kerb_user&sslMode=PREFERRED&trustCertificat
eKeyStoreUrl=file:///cdep/mariadbssl/db_keystore.jks&trustCertif
icateKeyStoreType=jks&trustCertificateKeyStorePassword=verystron
gpassword&enabledTLSProtocols=TLSv1.2
```

For non-TLS cluster

Update the JDBC URL in /etc/cloudera-scm-server/db.properties as follows:

Important: Do the following modifications to the original JDBC URL:

- Modify from user=cm to user=cm_kerb_user

The updated URL should look like this:

```
com.cloudera.cmf.orm.hibernate.connection.url=jdbc:mariadb://<<h
ost>>:3306/cm?user=cm_kerb_user
```

e. Add below properties to /etc/default/cloudera-scm-server:

```
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS}
-Djava.security.krb5.kdc=krbmariadb-1.krbmariadb.root.hwx.site
-Djava.security.krb5.realm=ROOT.HWX.SITE -Djava.security.auth.login.co
nfig=/etc/jaas.conf"
```

f. Restart the Cloudera Manager Server by running the following command:

```
sudo systemctl restart cloudera-scm-server
```

Configuring a database for Ranger or Ranger KMS

Additional steps to configure databases for Ranger or Ranger KMS.

After you have installed a database, use these steps to configure the database for Ranger or Ranger KMS. Ranger and Ranger KMS should use separate databases.

Configuring a Ranger or Ranger KMS Database: MySQL/MariaDB

Prior to upgrading your cluster to Cloudera Base on premises you must configure the MySQL or MariaDB database instance for Ranger by creating a Ranger database and user. Before you begin the transition, review the support policies of database and admin policy support for transactions.

Before you begin

A supported version of MySQL or MariaDB must be running and available to be used by Ranger. See Database Requirements.

Important:

- Ranger and Ranger KMS should use separate databases.
- Ranger only supports the InnoDB engine for MySQL and MariaDB databases.

When using MySQL or MariaDB, the storage engine used for the Ranger admin policy store tables must support transactions. InnoDB supports transactions. A storage engine that does not support transactions is not suitable as a policy store.

Procedure

1. Log in to the host where you want to set up the MySQL database for Ranger.
2. Make sure you have the MYSQL connector for MYSQL version 5.7 or higher in the /usr/share/java/ directory with name mysql-connector-java.jar.

   Important: If you are using TLS v1.2, you must use version 5.1.48

3. Edit the following file: /etc/my.cnf and add the following line:

   ```
   log_bin_trust_function_creators = 1
   ```

   Warning: If you do not add this configuration, the upgrade will fail and reverting your deployment to a stable state will be difficult.

4. Restart the database:

   ```
   systemctl restart mysqld
   ```

   or:

   ```
   systemctl restart mariadb
   ```

5. Log in to mysql:

   ```
   mysql -u root
   ```

6. Run the following commands to create the Ranger database and user.

   Substitute the following in the command:

   - (optional) Replace rangeradmin with a username of your choice. Note this username, you will need to enter it later when running the Upgrade Cluster command.
   - (optional) Replace cloudera with a password of your choice. Note this password, you will need to enter it later when running the Upgrade Cluster command.
   - *<RANGER ADMIN ROLE HOSTNAME>* – the name of the host where the Ranger Admin role will run. Note this host, you will need to enter it later when running the Upgrade Cluster command.

   ```
   CREATE DATABASE ranger;
   CREATE USER 'rangeradmin'@'%' IDENTIFIED BY 'cloudera';
   CREATE USER 'rangeradmin'@'localhost' IDENTIFIED BY 'cloudera';
   CREATE USER 'rangeradmin'@'<RANGER ADMIN ROLE HOSTNAME>' IDENTIFIED BY
   'cloudera';
   GRANT ALL PRIVILEGES ON ranger.* TO 'rangeradmin'@'%';
   GRANT ALL PRIVILEGES ON ranger.* TO 'rangeradmin'@'localhost';
   GRANT ALL PRIVILEGES ON ranger.* TO 'rangeradmin'@'<RANGER ADMIN ROLE
    HOSTNAME>';
   FLUSH PRIVILEGES;
   ```

7. Use the exit; command to exit MySQL.
8. Test connecting to the database using the following command:

   ```
   mysql -u RANGERADMIN -pCLOUDERA
   ```

9. After testing the connection, use the exit; command to exit MySQL.
10. Continue with the cluster installation or upgrade to complete the transition.

Configuring a Ranger or Ranger KMS Database: Oracle

Prior to upgrading your cluster to Cloudera Base on premises you must configure the Oracle database instance for Ranger by creating a Ranger database and user. Before you begin the transition, review the support policies of database and admin policy support for transactions.

Before you begin

A supported version of Oracle must be running and available to be used by Ranger.

Procedure

1. Log in to the host where the Oracle database is running and launch Oracle sqlplus:

```
sqlplus sys/root as sysdba
```

2. Provide the Ranger database and user instructions to the DBA to execute. Consult the username/schema name carefully in case of using multiple environments (test/dev etc.) on the same Oracle DB server. For information about creating Ranger database and user, see Creating the relevant schema DBs.

What to do next

Continue installing or upgrading your cluster.

Configuring a Ranger or Ranger KMS Database: Oracle using /ServiceName format

Prior to upgrading your cluster to Cloudera Base on premises you must configure the Oracle database instance for Ranger by creating a Ranger database and user. Before you begin the transition, review the support policies of database and admin policy support for transactions.

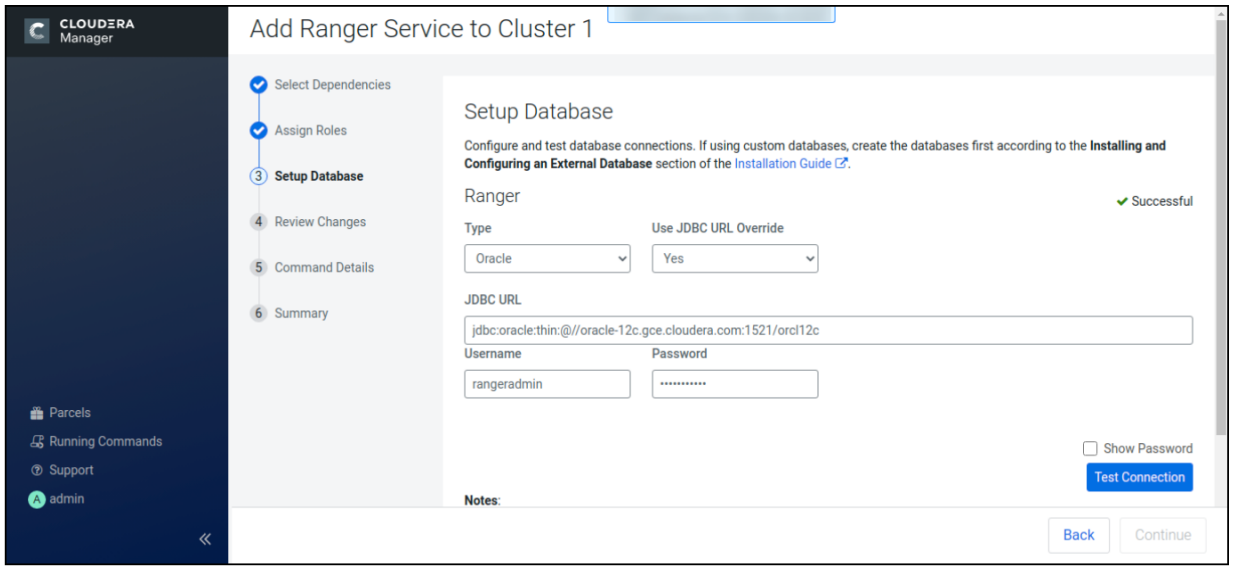
Before you begin

A supported version of Oracle must be running and available to be used by Ranger. See Database Requirements.

Important: Ranger and Ranger KMS should use separate databases.

Procedure

1. While installing Ranger service from Cloudera Manager using the installation wizard, in Setup Database, set the connection properties, as shown in the following example:



2. In Database Type, select Oracle.
3. In Use JDBC URL Override, select Yes.

4. In JDBC URL, type:

```
jdbc:oracle:thin:@//host:port/ServiceName
```

   This sets the JDBC URL to have the ServiceName connection url format.
5. In Username, type the user name required for connecting to the Oracle database Service Name you defined in the JDBC URL.
6. In Password, type the user name required for connecting to the Oracle database Service Name you defined in the JDBC URL.

   Note: Use similar steps to configure Ranger KMS service with Oracle database type using /ServiceName format. For Ranger KMS, use rangerkms rather than rangeradmin.
7. Click Test Connection.
8. After connection succeeds, click Continue.

What to do next

Continue installing or upgrading your cluster.

Configuring a PostgreSQL Database for Ranger or Ranger KMS

Complete the following steps to configure a PostgreSQL database instance for Ranger or Ranger KMS.

Configuring a PostgreSQL Database for Ranger or Ranger KMS on RHEL/Centos

Before you begin

Important: Ranger and Ranger KMS should use separate databases.

Note: For supported RHEL/Centos versions, see Cloudera Support Matrix.

Procedure

1. Run the following command to install PostgreSQL server:

```
sudo yum install postgresql-server
```

2. Initialize the Postgres database and start PostgreSQL:

```
sudo postgresql-setup initdb
sudo systemctl start postgresql
```

3. Optional: Configure PostgreSQL to start on boot:

```
sudo systemctl enable postgresql
```

4. Update the postgresql.conf file, which is usually found in /var/lib/pgsql/data or /var/lib/postgresql/data:
   - Uncomment and change #listen_addresses = 'localhost' to listen_addresses = '*'
   - Uncomment the #port = line and specify the port number (the default is 5432)
   - Optional: Uncomment and change #standard_conforming_strings= to standard_conforming_strings = off
5. Update the pg_hba.conf file, which is usually found in /var/lib/pgsql/data or /etc/postgresql/<version>/main:
   - Add the following line to allow connection to the Ranger database from any host:

```
host    ranger          rangeradmin     0.0.0.0/0               md5
```

     Note: For Ranger KMS, use rangerkms rather than rangeradmin.

Note: Ranger and Ranger KMS services are compatible with PostgreSQL server configured to use SCRAM-SHA-256 authentication. The instructions above remain applicable even when pg_hba.conf is configured to use scram-sha-256 instead of md5.

To enable SCRAM-SHA-256 authentication, perform the following steps:

a. Set the following property in postgresql.conf:

```
password_encryption = scram-sha-256
```

b. Add an authentication rule to pg_hba.conf specifying scram-sha-256 as the authentication method:

```
host ranger rangeradmin 0.0.0.0/0 scram-sha-256
```

In Step 8, ensure to install the PostgreSQL JDBC driver version 42.2.0 or later, which includes support for SCRAM-SHA-256 authentication.

6. Restart PostgreSQL:

```
sudo systemctl restart postgresql
```

7. The PostgreSQL database administrator should be used to create the Ranger databases. The following series of commands could be used to create the rangeradmin user and grant it adequate privileges. Be sure to replace 'password' with a strong password.

```
echo "CREATE DATABASE ranger;" | sudo -u postgres psql -U postgres
echo "CREATE USER rangeradmin WITH PASSWORD 'password';" | sudo -u postgr
es psql -U postgres
echo "GRANT ALL PRIVILEGES ON DATABASE ranger TO rangeradmin;" | sudo -u
 postgres psql -U postgres
```

Note: For Ranger KMS, use rangerkms rather than rangeradmin.

8. Install the PostgreSQL JDBC driver. If you would like to use the PostgreSQL JDBC driver version shipped with the OS repositories, run the following command:

```
yum install postgresql-jdbc*
```

You can also download the JDBC driver from the official PostgreSQL JDBC Driver website – https://jdbc.postgresql.org/download/.

9. Rename the Postgres JDBC driver .jar file to postgresql-connector-java.jarand copy it to the /usr/share/java directory. The following copy command can be used if the Postgres JDBC driver .jar file is installed from the OS repositories:

```
cp /usr/share/java/postgresql-jdbc.jar /usr/share/java/postgresql-connec
tor-java.jar
```

10. Confirm that the .jar file is in the Java share directory:

```
ls /usr/share/java/postgresql-connector-java.jar
```

11. Change the access mode of the .jar file to 644:

```
chmod 644 /usr/share/java/postgresql-connector-java.jar
```

What to do next

Ensure that the Ranger Solr and Ranger HDFS plugins are enabled. See Additional Steps for Apache Ranger on page 175 for details.

Configuring a PostgreSQL Database for Ranger on Ubuntu

Procedure

1. Run the following command to install PostgreSQL server:

```
apt-get install postgresql-server
```

2. Initialize the Postgres database and start PostgreSQL:

```
sudo postgresql-setup initdb
sudo systemctl start postgresql
```

3. Optional: Configure PostgreSQL to start on boot:

```
sudo systemctl enable postgresql
```

4. Edit the /var/lib/pgsql/data/postgresql.conf file:
   - Uncomment and change #listen_addresses = 'localhost' to listen_addresses = '*' *
   - Uncomment the #port = line and specify the port number (the default is 5432)
   - Optional: Uncomment and change #standard_conforming_strings= to standard_conforming_strings = off
5. Update the /var/lib/pgsql/data/pg_hba.conf file to allow connection to the Ranger database from any host:
   - Add the following line:

```
host    ranger          rangeradmin     0.0.0.0/0               md5
```



Note: Ranger and Ranger KMS services are compatible with PostgreSQL server configured to use SCRAM-SHA-256 authentication. The instructions above remain applicable even when pg_hba.conf is configured to use scram-sha-256 instead of md5.

To enable SCRAM-SHA-256 authentication, perform the following steps:

a. Set the following property in postgresql.conf:

```
password_encryption = scram-sha-256
```

b. Add an authentication rule to pg_hba.conf specifying scram-sha-256 as the authentication method:

```
host ranger rangeradmin 0.0.0.0/0 scram-sha-256
```

In Step 8, ensure to install the PostgreSQL JDBC driver version 42.2.0 or later, which includes support for SCRAM-SHA-256 authentication.

6. Restart PostgreSQL:

```
sudo systemctl restart postgresql
```

7. The PostgreSQL database administrator should be used to create the Ranger databases. The following series of commands could be used to create the rangeradmin user and grant it adequate privileges. Be sure to replace 'password' with a strong password.

```
echo "CREATE DATABASE ranger;" | sudo -u postgres psql -U postgres
echo "CREATE USER rangeradmin WITH PASSWORD 'password';" | sudo -u postgr
es psql -U postgres
echo "GRANT ALL PRIVILEGES ON DATABASE ranger TO rangeradmin;" | sudo -u
 postgres psql -U postgres
```

8. Install the PostgreSQL connector:

```
apt-get install postgresql-jdbc
```

9. Copy the connector .jar file to the Java share directory:

```
cp /usr/share/java/postgresql-jdbc.jar /usr/share/java/postgresql-connec
tor-java.jar
```

10. Confirm that the .jar file is in the Java share directory:

```
ls /usr/share/java/postgresql-connector-java.jar
```

11. Change the access mode of the .jar file to 644:

```
chmod 644 /usr/share/java/postgresql-connector-java.jar
```

What to do next

Ensure that the Ranger Solr and Ranger HDFS plugins are enabled. See the following topic, *Additional Steps for Ranger*, for details.

Configure Ranger with SSL/TLS enabled PostgreSQL Database

Steps to configure Ranger service with SSL/TLS enabled PostgreSQL database.

Before you begin

Make sure that:

- The database and database user for Ranger service are created in the required postgreSQL.
- A database server certificate is issued by a trusted certificate authority.
- The server host name matches the host name in the database server certificate.



Note: From Cloudera on premises 7.1.5 and higher, Ranger service requires postgres jdbc driver version >= 42.2.5. The Ranger code also constructs the JDBC connection string to have sslmode=verify-full, if Ranger Database SSL configurations are set in case of postgresql database type.

- Copy the database server certificate to /var/lib/ranger/ path , or use any custom path.

About this task

While installing Ranger service from Cloudera Manager using the installation wizard, stop at Setup Database to set the connection properties. The following steps apply to both FIPS-enabled and non FIPS-enabled clusters.

Procedure

1. In Setup Databases Type , select PostgreSQL.
2. In Use JDBC URL Override, select Yes.
3. In JDBC URL, type the following connection URL format:

```
jdbc:postgresql://<DB-HOST>:<DB-PORT>/<RANGER-DB>?sslmode=verify-
full&sslrootcert=
<path-to-database-server-certificate>
```

4. Click Test Connection.
5. Click Continue.
6. In Review Config, update the following configurations:

   ranger.db.ssl.enabled

   true

   ranger.db.ssl.verifyServerCertificate

true

ranger.db.ssl.auth.type

1-way

ranger.db.ssl.certificateFile

<path-to-db-server-certificate>

Enable HA for a Ranger Postgres database

Ranger supports high availability enabled databases when deployed with Postgres.

About this task

To support high availability (HA) of a Postgres Ranger database, integrate HAProxy (a third-party load balancer product) into your environment. HAProxy will handle the load balancing. Replication Manager (repmgr) handles failover and recovery of the database.

Note: Currently, recovery of a previous master (as secondary) is not supported.

Before you begin

Ranger jdbc URL must point to a load balancer URL.

Example JDBC URL to connect to load balancer:

SSL:

```
jdbc:postgresql://<lb_hostname>:<lb_port>/ranger1?sslmode=verify-ca&sslrootc
ert=/cdep/pgssl/server.crt
```

Non-SSL:

```
jdbc:postgresql://<lb_hostname>:<lb_port>/ranger1
```

where:

Lb_hostname = Load balancer hostname

For example: mvnssl-sync-1.mvnssl-sync.root.hwx.site

Lb_port = Load balancer port

For example: 6432

The following steps describe how to setup HAProxy as a load balancer and `repmgr` as replication manager for postgres database:

Procedure

1. Install the HAProxy load balancer.

   Note: Install the load balancer only on the first node.

   a) On CentOS, use yum repository to install HAProxy.

   ```
   yum install haproxy
   ```

   The latest version of HAProxy available installs in yum repository.

   b) Run the following command to ensure that HAProxy runs every time the server reboots.

   ```
   chkconfig haproxy on
   ```

c) Start the service, using the following command.

```
systemctl start haproxy && systemctl status haproxy
```

2. Install the replication manager.



Note: Perform this step on all the nodes of databases.

You can use `repmgr` package for performing replication of databases. `repmgr` is available along with postgres repository so we can directly run the install command to install this package.

```
sudo yum install repmgr_12
```



Note: this command installs postgres 12. Use a repmgr version that supports the postgres version installed.

3. Configure the HA proxy:

a) Edit the HAProxy Configuration file.

```
vi /etc/haproxy/haproxy.cfg
```

b) Add the primary database server hostname and port number.

Example configuration file for haproxy.cfg:

```
frontend                        postgres-front
    mode tcp
    option http-server-close
    timeout client          3h
    timeout http-keep-alive 10s
    bind                        *:6432
    option httpchk
    default_backend             postgres-back
    option tcplog

backend postgres-back
    mode tcp
    option http-server-close
    retries                 5
    timeout connect         3s
    timeout server          3h
    timeout http-keep-alive 10s
    option httpchk
    option tcp-check
    option tcplog
    tcp-check connect
    balance                     source
server postgresprimary <hostname_of_primary_db>:5432 check
```

4. Configure the replication manager:

Postgres service must be up & running.

a) Update the following settings in postgresql.conf on the primary server.

```
max_wal_senders = 10

max_replication_slots = 10

wal_level = 'hot_standby' or 'replica' or 'logical'

hot_standby = on
```

```
archive_mode = on

archive_command = '/bin/true'

shared_preload_libraries = 'repmgr'
```



Note: In case of synchronous replication, synchronous_standby_names config param needs to be set in postgresql.conf

b) Create a dedicated user for repmgr.

```
create user repmgr;
create database repmgr with owner repmgr;
```

c) Ensure the repmgr user has appropriate permissions in pg_hba.conf and can connect in replication mode.

```
   local   replication   repmgr                        trust
    host    replication   repmgr      127.0.0.1/32            trust
    host    replication   repmgr      192.168.1.0/24          trust

    local   repmgr        repmgr                              trust
    host    repmgr        repmgr      127.0.0.1/32            trust
    host    repmgr        repmgr      192.168.1.0/24          trust
```

d) Create a repmgr.conf on the master server with the following entries:

```
cluster='failovertest'

node_id=1

node_name=node1

conninfo='host=node1 user=repmgr dbname=repmgr connect_timeout=2'

data_directory='/var/lib/pgsql/12/data/'
failover=automatic
promote_command='/usr/pgsql-12/bin/repmgr standby promote -f /var/lib/p
gsql/repmgr.conf --log-to-file'
follow_command='/usr/pgsql-12/bin/repmgr standby follow -f /var/lib/pgs
ql/repmgr.conf --log-to-file --upstream-node-id=%n'
```



Note: Users can also provide a script to automate the updating of HAProxy configurations in case of a failover. Provide this script in the above promote_command config.

e) Register the primary server with repmgr:

```
-bash-4.2$ /usr/pgsql-12/bin/repmgr -f /var/lib/pgsql/repmgr.conf primar
y register
```

f) Create the repmgr.conf file on standby server with following settings:



Note: In the commands for the host IP info, specify the IP of the standby server.



Note: Users can also provide a script to automate the updating of HAProxy configurations in case of a failover. Provide this script in the promote_command config.

```
-bash-4.2$ cat repmgr.conf

node_id=2

node_name=node2
```

```
conninfo='host=172.16.140.137 user=repmgr dbname=repmgr connect_timeout=
2'

data_directory='/var/lib/pgsql/12/data'
failover=automatic
promote_command='/usr/pgsql-12/bin/repmgr standby promote -f /var/lib/p
gsql/repmgr.conf --log-to-file'
follow_command='/usr/pgsql-12/bin/repmgr standby follow -f /var/lib/pgs
ql/repmgr.conf --log-to-file --upstream-node-id=%n'
```

g) Start cloning the database, using the following command:

```
-bash-4.2$ /usr/pgsql-12/bin/repmgr -h 172.16.140.135 -U repmgr -d repmg
r -f /var/lib/pgsql/repmgr.conf standby clone
```

h) Register the standby server with repmgr :

```
-bash-4.2$ /usr/pgsql-12/bin/repmgr -f /var/lib/pgsql/repmgr.conf standb
y register
```

i) Enable the automatic failover, by starting the repmgrd daemon process on master and slave:

```
-bash-4.2$ /usr/pgsql-12/bin/repmgrd -f /var/lib/pgsql/repmgr.conf
```

Related Information

HA Proxy Basics: Load Balance your servers

Prerequisites for setting up a basic replication cluster with repmgr

Configuring the Database for Streaming Components

Additional steps to configure the databases for Schema Registry and Streams Messaging Manager (SMM).

Configure PostgreSQL for Streaming Components

If you are installing Schema Registry or Streams Messaging Manager (SMM), you must configure the database to store metadata.

About this task

After you install PostgreSQL, configure the database to store:

- Schema Registry data such as the schemas and their metadata, all the versions and branches.
- SMM data such as Kafka metadata, stores metrics, and alert definitions.



Important: For the Schema Registry database, you must set collation to be case sensitive.

.

Procedure

1. Log in to Postgres:

```
sudo su postgres
psql
```

2. For the Schema Registry metadata store, create a database called registry with the password registry:

```
create database registry;
CREATE USER registry WITH PASSWORD 'registry';
GRANT ALL PRIVILEGES ON DATABASE "registry" to registry;
```

3. For the SMM metadata store, create a database called streamsmsgmgr with the password streamsmsgmgr:

```
create database streamsmsgmgr;
CREATE USER streamsmsgmgr WITH PASSWORD 'streamsmsgmgr';
GRANT ALL PRIVILEGES ON DATABASE "streamsmsgmgr" to streamsmsgmgr;
```

If you cannot grant all privileges, grant the following privileges that SMM and Schema Registry require at a minimum:

- CREATE/ALTER/DROP TABLE
- CREATE/ALTER/DROP INDEX
- CREATE/ALTER/DROP SEQUENCE
- CREATE/ALTER/DROP PROCEDURE

For example:

```
grant create session to streamsmsgmgr;
grant create table to streamsmsgmgr;
grant create sequence to streamsmsgmgr;
```

Configuring MySQL for Streaming Components

If you intend to use MySQL to store the metadata for Streams Messaging Manager or Schema Registry, you must configure the MySQL database.

About this task

Configure the database to store:

- In Schema Registry, the schemas and their metadata, all the versions and branches.
- In SMM, the Kafka metadata, stores metrics, and alert definitions.

Important: For the Schema Registry database, you must set collation to be case sensitive.

.

Procedure

1. Log in to the host.
   a) Run the following command for Schema Registry:

```
ssh [MY_SCHEMA_REGISTRY_HOST]
```

   b) Run the following command for Streams Messaging Manager:

```
ssh [MY_STREAMS_MESSAGING_MANAGER_HOST]
```

2. Launch the MySQL monitor:

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

3. Create the database for the Schema Registry and the SMM metastore:

```
create database registry;
create database streamsmsgmgr;
```

4. Create Schema Registry and SMM user accounts, replacing the final IDENTIFIED BY string with your password:

```
CREATE USER 'registry'@'%' IDENTIFIED BY 'R12$%34qw';
CREATE USER 'streamsmsgmgr'@'%' IDENTIFIED BY 'R12$%34qw';
```

5. Assign privileges to the user account:

```
GRANT ALL PRIVILEGES ON registry.* TO 'registry'@'%' WITH GRANT OPTION ;
GRANT ALL PRIVILEGES ON streamsmsgmgr.* TO 'streamsmsgmgr'@'%' WITH GRANT
OPTION ;
```

   If you cannot grant all privileges, grant the following privileges that SMM and Schema Registry require at a minimum:

   - CREATE/ALTER/DROP/REFERENCES TABLE
   - CREATE/ALTER/DROP INDEX
   - CREATE/ALTER/DROP SEQUENCE
   - CREATE/ALTER/DROP PROCEDURE

   For example:

```
grant create session to streamsmsgmgr;
grant create table to streamsmsgmgr;
grant create sequence to streamsmsgmgr;
```

6. Commit the operation:

```
commit;
```

Configuring Oracle for Streaming Components

If you intend to use Oracle to store the metadata for Streams Messaging Manager or Schema Registry, you must configure the Oracle database.

About this task

After you install Oracle, configure the database to store:

- Schema Registry data such as the schemas and their metadata, all the versions and branches.
- SMM data such as Kafka metadata, stores metrics, and alert definitions.

For setting the relevant tablespaces for the SMM and Schema Registry, see Creating the relevant schema DBs.

Results

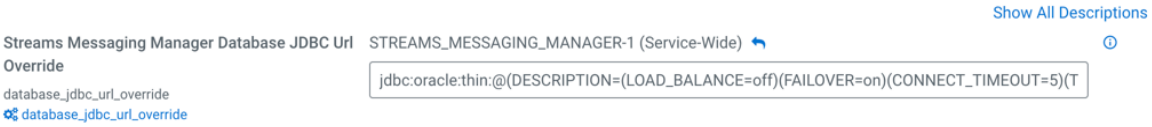
The new user needs to have permissions to create or drop tables and other database structures because when the Schema Registry or SMM service starts, it automatically creates the database structure. If the database structure already exists, then the service alters it as necessary.

Note: When using Oracle RAC, you need to enter the JDBC connection URL into the database_jdbc_url_override field in Cloudera Manager for both Schema Registry and SMM, as shown in the following images:

For Schema Registry,



For SMM,



For more information on URL for Oracle RAC, see https://docs.oracle.com/cd/E57185_01/EPMIS/apbs01s01.html.

Configure Network Names

You must configure each host in the cluster to ensure that all members can communicate with each other.

About this task

Important: Cloudera Runtime requires IPv4. IPv6 is not supported.

Tip: When bonding, use the bond0 IP address as it represents all aggregated links.

Procedure

1. Set the hostname to a unique name (not localhost).

```
sudo hostnamectl set-hostname foo-1.example.com
```

2. Edit /etc/hosts with the IP address and fully qualified domain name (FQDN) of each host in the cluster. You can add the unqualified name as well.

```
1.1.1.1  foo-1.example.com  foo-1
2.2.2.2  foo-2.example.com  foo-2
3.3.3.3  foo-3.example.com  foo-3
4.4.4.4  foo-4.example.com  foo-4
```

Important:

- The canonical name of each host in /etc/hosts must be the FQDN (for example myhost-1.example .com), not the unqualified hostname (for example myhost-1). The canonical name is the first entry after the IP address.
- Do not use aliases, either in /etc/hosts or in configuring DNS.
- Unqualified hostnames (short names) must be unique in a Cloudera Manager instance. For example, you cannot have both *HOST01.EXAMPLE.COM* and *HOST01.STANDBY.EXAMPLE.COM* managed by the same Cloudera Manager Server.

3. Edit /etc/sysconfig/network with the FQDN of this host only:

```
HOSTNAME=foo-1.example.com
```

4. Verify that each host consistently identifies to the network:
   a) Run uname -a and check that the hostname matches the output of the hostname command.
   b) Run /sbin/ifconfig and note the value of inet addr in the eth0 (or bond0) entry, for example:

```
eth0      Link encap:Ethernet  HWaddr 00:0C:29:A4:E8:97
          inet addr:172.29.82.176  Bcast:172.29.87.255  Mask:255.255.2
48.0
...
```

   c) Run host -v -t A $(hostname) and verify that the output matches the hostname command. The IP address should be the same as reported by ifconfig for eth0 (or bond0):

```
Trying "foo-1.example.com"
...
;; ANSWER SECTION:
foo-1.example.com. 60 IN A 172.29.82.176
```



Important: If the host command is not installed on your system, then install it by running the following command:

- RHEL:

```
yum install bind-utils
```

- Ubuntu:

```
apt install bind9-host
```

- SLES:

```
zypper in bind-utils
```

Setting SELinux Mode

Security-Enhanced Linux (SELinux) allows you to set access control through policies. If you are having trouble deploying Cloudera Runtime or CDH with your policies, set SELinux in permissive mode on each host before you deploy Cloudera Runtime or CDH on your cluster.

About this task



Attention:

Cloudera does not support SELinux in the enforcing mode.

Cloudera is not liable for any SELinux policy development, support, or enforcement measures. If you experience issues running Cloudera software with SELinux enabled, contact your OS provider for assistance.

If you are using SELinux in enforcing mode, Cloudera Support can request that you disable SELinux or change the mode to permissive to rule out SELinux as a factor when investigating reported issues.

Procedure

1. Check the SELinux state:

```
getenforce
```

2. If the output is either Permissive or Disabled, you can skip this task and continue to Disabling the Firewall to disable the firewall on each host in your cluster. If the output is enforcing, continue to the next step.
3. Open the /etc/selinux/config file (in some systems, the /etc/sysconfig/selinux file).
4. Change the line SELINUX=enforcing to SELINUX=permissive.

5. Save and close the file.
6. Restart your system or run the following command to disable SELinux immediately:

```
setenforce 0
```

After you have installed and deployed Cloudera Runtime or CDH, you can re-enable SELinux by changing SELINUX=permissive back to SELINUX=enforcing in /etc/selinux/config (or /etc/sysconfig/selinux), and then running the following command to immediately switch to enforcing mode:

```
setenforce 1
```

If you are having trouble getting Cloudera Software working with SELinux, contact your OS vendor for support. Cloudera is not responsible for developing or supporting SELinux policies.

Disabling the Firewall

To disable the firewall on each host in your cluster, perform the following steps on each host.

Procedure

1. If the iptables command is not installed on your system, then install it by running the following command:
   - RHEL:

     ```
     sudo yum install iptables
     ```
   - SLES:

     ```
     sudo zypper install iptables
     ```
   - Ubuntu:

     ```
     sudo apt-get install iptables
     ```
2. For iptables, save the existing rule set:

   ```
   sudo iptables-save > ~/firewall.rules
   ```
3. Disable the firewall:
   - RHEL compatible:

     ```
     sudo systemctl disable firewalld
     sudo systemctl stop firewalld
     ```
   - SLES:

     ```
     sudo chkconfig SuSEfirewall2_setup off
     sudo chkconfig SuSEfirewall2_init off
     sudo rcSuSEfirewall2 stop
     ```
   - Ubuntu:

     ```
     sudo service ufw stop
     ```

Enable an NTP Service

Cloudera Runtime requires that you configure a Network Time Protocol (NTP) service on each machine in your cluster. Most operating systems include the ntpd service for time synchronization.

About this task

Some operating systems use chronyd by default instead of ntpd. If chronyd is running (on any OS), Cloudera Manager uses it to determine whether the host clock is synchronized. Otherwise, Cloudera Manager uses ntpd.

Note: If you are using ntpd to synchronize your host clocks, but chronyd is also running, Cloudera Manager relies on chronyd to verify time synchronization, even if it is not synchronizing properly. This can result in Cloudera Manager reporting clock offset errors, even though the time is correct.

To fix this, either configure and use chronyd or disable it and remove it from the hosts.

Important:

You should verify whether ntpd or chronyd is already installed and running. If ntpd or chronyd is already installed and running, then you can skip enabling NTP service.

Refer your OS vendor's documentation to install and configure either ntpd or chronyd.

Impala Requirements

To perform as expected, Impala depends on the availability of the software, hardware, and configurations described in the following sections.

Product Compatibility Matrix

The ultimate source of truth about compatibility between various versions of Cloudera Runtime, Cloudera Manager, and various Cloudera Runtime components is the Product Compatibility Matrix.

Supported Operating Systems

The relevant supported operating systems and versions for Impala are the same as for the corresponding Cloudera Runtime platforms. For details, see the *Operating System Requirements* topic.

Hive Metastore and Related Configuration

Impala can interoperate with data stored in Hive, and uses the same infrastructure as Hive for tracking metadata about schema objects such as tables and columns. The following components are prerequisites for Impala:

To install the metastore:

1. Install a MySQL or PostgreSQL database. Start the database if it is not started after installation.
2. Download the MySQL Connector or the PostgreSQL connector and place it in the /usr/share/java/ directory.
3. Use the appropriate command line tool for your database to create the metastore database.
4. Use the appropriate command line tool for your database to grant privileges for the metastore database to the hive user.
5. Modify hive-site.xml to include information matching your particular database: its URL, username, and password. You will copy the hive-site.xml file to the Impala Configuration Directory later in the Impala installation process.

Java Dependencies

Although Impala is primarily written in C++, it does use Java to communicate with various Hadoop components:

- The officially supported JVMs for Impala are the OpenJDK JVM and Oracle JVM. Other JVMs might cause issues, typically resulting in a failure at `impalad` startup. In particular, the JamVM used by default on certain levels of Ubuntu systems can cause `impalad` to fail to start.
- Internally, the `impalad` daemon relies on the JAVA_HOME environment variable to locate the system Java libraries. Make sure the `impalad` service is not run from an environment with an incorrect setting for this variable.
- All Java dependencies are packaged in the impala-dependencies.jar file, which is located at /usr/lib/impala/lib/. These map to everything that is built under fe/target/dependency.

Networking Configuration Requirements

As part of ensuring best performance, Impala attempts to complete tasks on local data, as opposed to using network connections to work with remote data. To support this goal, Impala matches the hostname provided to each Impala daemon with the IP address of each DataNode by resolving the hostname flag to an IP address. For Impala to work with local data, use a single IP interface for the DataNode and the Impala daemon on each machine. Ensure that the Impala daemon's hostname flag resolves to the IP address of the DataNode. For single-homed machines, this is usually automatic, but for multi-homed machines, ensure that the Impala daemon's hostname resolves to the correct interface. Impala tries to detect the correct hostname at start-up, and prints the derived hostname at the start of the log in a message of the form:

```
Using hostname: impala-daemon-1.example.com
```

In the majority of cases, this automatic detection works correctly. If you need to explicitly set the hostname, do so by setting the --hostname flag.



Note: RedHat Users with server configurations that utilise networks of 10 Gb/s or more might need to adjust the vm.min_free_kbytes parameter to a significantly increased value to ensure ample space for storing packet backlogs before the operating system processes them. This helps prevent packet drops and avoids connection timeouts.

These timeouts might manifest as EndDataStream errors or generic Impala Connection timeouts.

Additional details regarding modifications to the vm.min_free_kbytes parameter and the reasons for its necessity can be found here.

Glibc Version Requirement

Impala daemons should be deployed on nodes using the same Glibc version since different Glibc version supports different Unicode standard version. Not using the same Glibc version might result in inconsistent UTF-8 behavior when UTF8_MODE is set to true.

Hardware Requirements

The memory allocation should be consistent across Impala executor nodes. A single Impala executor with a lower memory limit than the rest can easily become a bottleneck and lead to suboptimal performance.

This guideline does not apply to coordinator-only nodes.

Hardware Requirements for Optimal Join Performance

During join operations, portions of data from each joined table are loaded into memory. Data sets can be very large, so ensure your hardware has sufficient memory to accommodate the joins you anticipate completing.

While requirements vary according to data set size, the following is generally recommended:

- CPU

  Impala version 2.2 and higher uses the SSSE3 instruction set, which is included in newer processors.

  Note: This required level of processor is the same as in Impala version 1.x. The Impala 2.0 and 2.1 releases had a stricter requirement for the SSE4.1 instruction set, which has now been relaxed.

- Memory

  128 GB or more recommended, ideally 256 GB or more. If the intermediate results during query processing on a particular node exceed the amount of memory available to Impala on that node, the query writes temporary work data to disk, which can lead to long query times. Note that because the work is parallelized, and intermediate results for aggregate queries are typically smaller than the original data, Impala can query and join tables that are much larger than the memory available on an individual node.
- JVM Heap Size for Catalog Server

4 GB or more recommended, ideally 8 GB or more, to accommodate the maximum numbers of tables, partitions, and data files you are planning to use with Impala.

- Storage

  DataNodes with 12 or more disks each. I/O speeds are often the limiting factor for disk performance with Impala. Ensure that you have sufficient disk space to store the data Impala will be querying.

OS Memory Map Area Requirement for Impala

Default settings for max_map_count can be insufficient for Impala installations that run with many concurrent queries. We recommend increasing max_map_count to avoid potential failures due to exhausting memory mapping limits under heavy load.

max_map_count is the OS virtual memory parameter and defines the maximum number of memory map areas that a process can use.

To increase the max_map_countparameter in your OS and to make the above setting durable, refer to your OS documentation.

- For example, if you are using RHEL 9, add the following line to /etc/sysctl.conf:

```
/etc/sysctl.conf:vm.max_map_count=8000000
```

- Reload the config as root: sysctl -p.
- Check the new value: cat /proc/sys/vm/max_map_count.
- Restart.

User Account Requirements

For user account requirements, see the topic User Account Requirements in the Impala documentation.

Cloudera Runtime Cluster Hosts and Role Assignments

Cluster hosts can be broadly described as master hosts, utility hosts, gateway hosts, or worker hosts.

- Master hosts run Hadoop master processes such as the HDFS NameNode and YARN Resource Manager.
- Utility hosts run other cluster processes that are not master processes such as Cloudera Manager and one or more Hive Metastores.
- Gateway hosts are client access points for launching jobs in the cluster. The number of gateway hosts required varies depending on the type and size of the workloads.
- Worker hosts primarily run DataNodes and other distributed processes such as Impala.



Important: Cloudera recommends that you always enable high availability when Cloudera Runtime is used in a production environment.

The following tables describe the recommended role allocations for different cluster sizes. Note that these configurations take into account services dependencies that might not be obvious. For example, running Atlas or Ranger requires also running HBase, Kafka, Solr, and ZooKeeper. For details see Service Dependences in Cloudera Manager.



Attention: When High Availability (HA) is enabled and the total number of nodes is under 10, you must carefully plan the composition of the worker nodes. That is the utility nodes and master nodes. If you decide that your development cluster is to be HA enabled, you must add the HA configuration for at least 3-10 hosts for seamless performance.

3 - 10 Worker Hosts without High Availability

Master Hosts	Utility Hosts Gateway Hosts	Worker Hosts
Master Host 1: • NameNode • YARN ResourceManager • JobHistory Server • ZooKeeper • Kudu master • Spark History Server • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager	One host for all Utility and Gateway roles: • Secondary NameNode • Cloudera Manager • Cloudera Management Service • Cruise Control • Hive Metastore • HiveServer2 • Phoenix Query Server • Impala Catalog Server • Impala StateStore • Hue • Oozie • Gateway configuration • HBase backup master • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (Cloudera-INFRA-SOLR instance to support Atlas) • Streams Messaging Manager • Streams Replication Manager Service • ZooKeeper • Knox: One KnoxGateway service on utility or gateway hosts. • S3 Gateway	3 - 10 Worker Hosts: • DataNode • NodeManager • Impalad • Kudu tablet server • Kafka Broker • Kafka Connect • HBase RegionServer • Solr server (For Cloudera Search) • Streams Replication Manager Driver • ZooKeeper (Recommend 3 servers total) • Ozone DataNode

3 - 20 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper • JobHistory Server • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 2: • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 3: • Kudu master (Kudu requires an odd number of masters for HA.) • Spark History Server • JournalNode (requires dedicated disk) • ZooKeeper • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager	Utility Host 1: • Cloudera Manager • Cloudera Management Service • Cruise Control • Hive Metastore • Impala Catalog Server • Impala StateStore • Oozie • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (Cloudera-INFRA-SOLR instance to support Atlas) • Streams Messaging Manager • Streams Replication Manager Service • Knox: One KnoxGateway service for HA. Instead of utility, you can also select gateway hosts. Utility Host 2: • Hive Metastore • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (Cloudera-INFRA-SOLR instance to support Atlas) • Knox: One KnoxGateway service for HA. Instead of utility, you can also select gateway hosts.	One or more Gateway Hosts: • Hue • HiveServer2 • Phoenix Query Server • Gateway configuration • S3 Gateway	3 - 20 Worker Hosts: • DataNode • NodeManager • Impalad • Kudu tablet server • Kafka Broker (Recommend 3 brokers minimum) • Kafka Connect • HBase RegionServer • Solr server (For Cloudera Search, recommend 3 servers minimum) • Streams Replication Manager Driver • Ozone DataNode

20 - 80 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 2: • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 3: • ZooKeeper • JournalNode • JobHistory Server • Spark History Server • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager	Utility Host 1: • Cloudera Manager • Cruise Control • Hive Metastore • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (Cloudera-INFRA-SOLR instance to support Atlas) • Streams Messaging Manager • Streams Replication Manager Service Utility Host 2: • Cloudera Management Service • Hive Metastore • Impala Catalog Server • Impala StateStore • Oozie • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (Cloudera-INFRA-SOLR instance to support Atlas)	One or more Gateway Hosts: • Hue • HiveServer2 • Phoenix Query Server • Gateway configuration • S3 Gateway Two or more Gateway Hosts: • Knox: Two KnoxGateway services, one on each of the first two Gateway Hosts for HA. • S3 Gateway	20 - 80 Worker Hosts: • DataNode • NodeManager • Impalad • Kudu tablet server • Kafka Broker (Recommend 3 brokers minimum) • Kafka Connect • HBase RegionServer • Solr server (For Cloudera Search, recommend 3 servers minimum) • Streams Replication Manager Driver • Ozone DataNode

80 - 200 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 2: • NameNode • JournalNode • FailoverController • YARN ResourceManager • ZooKeeper • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 3: • ZooKeeper • JournalNode • JobHistory Server • Spark History Server • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager	Utility Host 1: • Cloudera Manager • Cruise Control • Streams Messaging Manager • Streams Replication Manager Service Utility Host 2: • Hive Metastore • Impala Catalog Server • Impala StateStore • Oozie Utility Host 3: • Host Monitor Utility Host 4: • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server Utility Host 5: • Hive Metastore • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server Utility Host 6: • Reports Manager Utility Host 7: • Service Monitor	One or more Gateway Hosts: • Hue • HiveServer2 • Phoenix Query Server • Gateway configuration • S3 Gateway Two or more Gateway Hosts: • Knox: Two KnoxGateway services, one on each of the first two Gateway Hosts for HA. • S3 Gateway	80 - 200 Worker Hosts: • DataNode • NodeManager • Impalad • Kudu tablet server (Recommend 100 tablet servers maximum) • Kafka Broker (Recommend 3 brokers minimum) • Kafka Connect • HBase RegionServer • Solr server (For Cloudera Search, recommend 3 servers minimum) • Streams Replication Manager Driver • Ozone DataNode

200 - 500 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: • NameNode • JournalNode • FailoverController • ZooKeeper • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 2: • NameNode • JournalNode • FailoverController • ZooKeeper • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 3: • YARN ResourceManager • ZooKeeper • JournalNode • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 4: • YARN ResourceManager • ZooKeeper • JournalNode • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 5: • JobHistory Server • Spark History Server • ZooKeeper • JournalNode • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Cloudera recommends no more than three masters for Kudu and HBase.	Utility Host 1: • Cloudera Manager • Cruise Control • Streams Messaging Manager • Streams Replication Manager Service Utility Host 2: • Hive Metastore • Impala Catalog Server • Impala StateStore • Oozie Utility Host 3: • Host Monitor Utility Host 4: • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (Cloudera-INFRA-SOLR instance to support Atlas) Utility Host 5: • Hive Metastore • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (Cloudera-INFRA-SOLR instance to support Atlas) Utility Host6: • Reports Manager Utility Host 7: • Service Monitor	One or more Gateway Hosts: • Hue • HiveServer2 • Phoenix Query Server • Gateway configuration • S3 Gateway Two or more Gateway Hosts: • Knox: Two KnoxGateway services, one on each of the first two Gateway Hosts for HA. • S3 Gateway	200 - 500 Worker Hosts: • DataNode • NodeManager • Impalad • Kudu tablet server (Recommend 100 tablet servers maximum) • Kafka Broker (Recommend 3 brokers minimum) • Kafka Connect • HBase RegionServer • Solr server (For Cloudera Search, recommend 3 servers minimum) • Streams Replication Manager Driver • Ozone DataNode

500 -1000 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: • NameNode • JournalNode • FailoverController • ZooKeeper • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 2: • NameNode • JournalNode • FailoverController • ZooKeeper • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 3: • YARN ResourceManager • ZooKeeper • JournalNode • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 4: • YARN ResourceManager • ZooKeeper • JournalNode • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 5: • JobHistory Server • Spark History Server • ZooKeeper • JournalNode • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Cloudera recommends no more than three masters for Kudu and HBase.	Utility Host 1: • Cloudera Manager • Cruise Control • Streams Messaging Manager • Streams Replication Manager Service Utility Host 2: • Hive Metastore • Impala Catalog Server • Impala StateStore • Oozie Utility Host 3: • Host Monitor Utility Host 4: • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (Cloudera-INFRA-SOLR instance to support Atlas) Utility Host 5: • Hive Metastore • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (Cloudera-INFRA-SOLR instance to support Atlas) Utility Host 6: • Reports Manager Utility Host 7: • Service Monitor	One or more Gateway Hosts: • Hue • HiveServer2 • Phoenix Query Server • Gateway configuration • S3 Gateway Two or more Gateway Hosts: • Knox: Two KnoxGateway services, one on each of the first two Gateway Hosts for HA. • S3 Gateway	500 - 1000 Worker Hosts: • DataNode • NodeManager • Impalad • Kudu tablet server (Recommend 100 tablet servers maximum) • Kafka Broker (Recommend 3 brokers minimum) • Kafka Connect • HBase RegionServer • Solr server (For Cloudera Search, recommend 3 servers minimum) • Streams Replication Manager Driver • Ozone DataNode

Related Information

Service Dependencies in Cloudera Manager
Configuring HMS for high availability
Configuring High Availability for Ranger RMS
Configuring High Availability for Ranger KMS

Configuring Ranger Usersync and Tagsync High Availability

Configuring Local Package and Parcel Repositories

Cloudera hosts two types of software repositories that you can use to install products such as Cloudera Manager or Cloudera Runtime—parcel repositories and package repositories. These repositories are effective solutions in most cases, but custom installation solutions are sometimes required.

For example, using the Cloudera-hosted software repositories requires client access over the Internet. Typical installations use the latest available software. In some scenarios, these behaviors might not be desirable, such as:

- You need to install older product versions. For example, in a Cloudera Runtime cluster, all hosts must run the same Cloudera Runtime version. After completing an initial installation, you may want to add hosts. This could be to increase the size of your cluster to handle larger tasks or to replace older hardware.
- The hosts on which you want to install Cloudera products are not connected to the Internet, so they cannot reach the Cloudera repository (for a parcel installation, only the Cloudera Manager Server needs Internet access, but for a package installation, all cluster hosts require access to the Cloudera repository). Most organizations partition parts of their network from outside access. Isolating network segments improves security, but can add complexity to the installation process.

In both of these cases, using an internal repository allows you to meet the needs of your organization, whether that means installing specific versions of Cloudera software or installing Cloudera software on hosts without Internet access.

Understanding Package Management

Before you configure a custom package management solution in your environment, understand the concepts of package management tools and package repositories.

Package Management Tools

Packages (rpm or deb files) help ensure that installations complete successfully by satisfying package dependencies. When you install a particular package, all other required packages are installed at the same time. For example, hadoop-0.20-hive depends on hadoop-0.20.

Package management tools, such as yum (RHEL), zypper (SLES), and apt-get (Ubuntu) are tools that can find and install required packages. For example, on a RHEL compatible system, you might run the command yum install hadoop-0.20-hive. The yum utility informs you that the Hive package requires hadoop-0.20 and offers to install it for you. zypper and apt-get provide similar functionality.

Package Repositories

Package management tools rely on package repositories to install software and resolve any dependency requirements. For information on creating an internal repository, see *Configuring a Local Package Repository*.

Repository Configuration Files

Information about package repositories is stored in configuration files, the location of which varies according to the package management tool.

- RHEL-compatible (yum): /etc/yum.repos.d
- SLES (zypper): /etc/zypp/zypper.conf
- Ubuntu (apt-get): /etc/apt/apt.conf (Additional repositories are specified using .list files in the /etc/apt/sources.list.d/ directory.)

For example, on a typical RedHat system, you might find:

```
ls -l /etc/yum.repos.d/
total 36
-rw-r--r--. 1 root root 1664 Dec  9  2015 RedHat-Base.repo
-rw-r--r--. 1 root root 1309 Dec  9  2015 RedHat-CR.repo
-rw-r--r--. 1 root root  649 Dec  9  2015 RedHat-Debuginfo.repo
-rw-r--r--. 1 root root  290 Dec  9  2015 RedHat-fasttrack.repo
```

```
-rw-r--r--. 1 root root  630 Dec  9  2015 RedHat-Media.repo
-rw-r--r--. 1 root root 1331 Dec  9  2015 RedHat-Sources.repo
-rw-r--r--. 1 root root 1952 Dec  9  2015 RedHat-Vault.repo
-rw-r--r--. 1 root root  951 Jun 24  2017 epel.repo
-rw-r--r--. 1 root root 1050 Jun 24  2017 epel-testing.repo
```

The .repo files contain pointers to one or more repositories. In the following excerpt from RedHat-Base.repo, there are two repositories defined: one named Base and one named Updates. The mirrorlist parameter points to a website that has a list of places where this repository can be downloaded.

```
[base]
name=RedHat-$releasever - Base
mirrorlist=http://mirrorlist.redhat.com/?release=$releasever&arch=$basearc
h&repo=os&infra=$infra
#baseurl=http://mirror.redhat.com/redhat/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-RedHat-8

#released updates
[updates]
name=RedHat-$releasever - Updates
mirrorlist=http://mirrorlist.redhat.com/?release=$releasever&arch=$basearch&
repo=updates&infra=$infra
#baseurl=http://mirror.redhat.com/redhat/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-RedHat-8
```

Listing Repositories

You can list the enabled repositories by running one of the following commands:

- RHEL compatible: yum repolist
- SLES: zypper repos
- Ubuntu: apt-get does not include a command to display sources, but you can determine sources by reviewing the contents of /etc/apt/sources.list and any files contained in /etc/apt/sources.list.d/.

The following shows an example of the output of yum repolist on a RedHat sytstem:

```
repo id               repo name                                          st
atus
base/8/x86_64         RedHat-8 - Base
9,591
epel/x86_64           Extra Packages for Enterprise Linux 8 - x86_64
12,382
extras/8/x86_64       RedHat-8 - Extras
  392
updates/8/x86_64      RedHat-8 - Updates                                  1
,962
repolist: 24,327
```

Installing Cloudera Manager

This procedure is recommended for installing Cloudera Manager which is a requirement for Cloudera Runtime in production environments.

Before you begin the installation, make sure you have reviewed the requirements and other considerations described in Before You Install.

If you want to install Cloudera Manager with high availability, there are several additional steps, including installing the Cloudera Manager software on an additional host. See Configuring Cloudera Manager for High Availability.

Important: Do not start the additional instance of Cloudera Manager until all of the installation steps including setting up a cluster are completed.

The general steps in the installation procedure are as follows:

- Step 1: Configure a Repository for Cloudera Manager
- Step 2: Deploy Cloudera Manager Server and Cloudera Manager Agents Server
- Step 3: Set up and configure the Cloudera Manager database
- Step 4: Start the Cloudera Manager Service

If you prefer to enable high availability for Cloudera Manager, then perform the step as described in (optional) Enable high availability for Cloudera Manager on page 174. Otherwise, for installing Cloudera Runtime, see Installing Cloudera Runtime.

Important: When you install the Cloudera Manager Agent package, it creates a Linux user account called cloudera-scm on each host. This user account is added to the Linux group called wheel. The cloudera-scm user account does not need to be part of the wheel group, and you can remove it manually. Note that during an upgrade, the cloudera-scm user account is added back to the wheel group.

Related Information

Use case 1: Use Cloudera Manager to generate internal CA and corresponding certificates

Use case 2: Enabling Auto-TLS with an intermediate CA signed by an existing Root CA

Use case 3: Enabling Auto-TLS with Existing Certificates

Installing Cloudera Runtime

Proceed through the installation wizard to accept licenses, install and configure Cloudera Runtime, and more.

Upload License File

On the Upload License File page, you can select either the trial version of or upload a license file:

1. Select one of the following options:
   - Upload License
   - Try for 60 days. The trial does not require a license file, but the trial expires after 60 days.

   If you have a license file for , then perform the following steps to upload the license file:

   a. Select Upload License.
   b. Click Upload License File.
   c. Browse to the location of the license file, select the file, and click Open.
   d. Click Upload.
   e. Click Continue.

   If you select the Try for 60 days option which is Edition Trial, you can upload a license file at a later time. Read the license agreement and select Yes, I accept the Standard License Terms and Conditions if you accept the terms and conditions of the license agreement. Then click Continue.
2. Click Continue to proceed with the installation.

   The Welcome page displays.

Add Cluster

The Add Cluster wizard provides a brief overview of the installation and configuration procedure, as well as some links to relevant documentation.

Click Continue to proceed with the installation.

Setup Auto-TLS

The Setup Auto-TLS page provides instructions for initializing the certificate manager for auto-TLS as part of setting up a new cluster.

This setup assumes that Cloudera Manager is the Certificate Authority (CA) to create and manage all the certificates for the cluster (either self-signed certificates or when Cloudera Manager acts as part of the customer intermediate Certificate Authority (CA) chain).

In case you do not approve adding Cloudera Manager to your intermediate CA chain, it means you are responsible to generate and maintain all the cluster certificates (that is the Use Case 3). In such a case, setting up AutoTLS should not be used.

Click Continue to proceed.

Setup a KDC

A Key Distribution Center (KDC) is required in order to create kerberized clusters. Kerberized clusters are required for Ranger, Atlas, and other services that depend on them.

The Setup KDC for this Cloudera Manager wizard walks you through the steps to configure Cloudera Manager for Kerberos Authentication. On completing the steps, the wizard displays a message indicating that KDC is setup and you can now create kerberized clusters.



Note: Setting KDC (Key Distribution Center) as part of Cloudera Manager means allowing Cloudera Manager to create and manage the Kerberos Principal creations for the cluster internally, without involving your external Kerberos server.

In case you want to manage the principals and want to create, remove, or update them on your Kerberos server, then Cloudera Manager can not be set as KDC, and the KDC wizard should not be used (hence the installation will continue without adding Kerberos at this point).

Please note that in such a case, you should not select any service which requires Kerberos (such as Ranger, RangerKMS, Atlas, Knox), and only once AutoTLS or Kerberos is set (post the Cloudera Manager Wizard deployment, separately), you can add these services to the cluster. If you deploy these services without Kerberos, the Ranger policies might not be created correctly and you need to remove them and regenerate them again.

Cluster Basics

The Cluster Basics page allows you to specify the Cluster Name

For new installations, a Regular Cluster (also called a base cluster) is the only option. You can add a compute cluster after you finish installing the cluster.

For more information on regular and compute clusters, and data contexts, see Virtual Private Clusters and SDX.

Enter a cluster name and click Continue.

Specify Hosts

Choose which hosts will run and other managed services.



Note: If you have enabled Auto-TLS, you must include the server host when you specify hosts.

1. To enable to automatically discover hosts on which to install and managed services, enter the cluster hostnames or IP addresses in the Hostnames field. You can specify hostname and IP address ranges as follows:

Expansion Range	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].example.com	host1.example.com, host2.example.com, host3.example.com

Expansion Range	Matching Hosts
host[07-10].example.com	host07.example.com, host08.example.com, host09.example.com, host10.example.com

Important: Unqualified hostnames (short names) must be unique in a instance. For example, you cannot have both *HOST01.EXAMPLE.COM* and *HOST01.STANDBY.EXAMPLE.COM* managed by the same Server.

You can specify multiple addresses and address ranges by separating them with commas, semicolons, tabs, or blank spaces, or by placing them on separate lines. Use this technique to make more specific searches instead of searching overly wide ranges. Only scans that reach hosts running SSH will be selected for inclusion in your cluster by default. You can enter an address range that spans over unused addresses and then clear the nonexistent hosts later in the procedure, but wider ranges require more time to scan.

2. Click Search. If there are a large number of hosts on your cluster, wait a few moments to allow them to be discovered and shown in the wizard. If the search is taking too long, you can stop the scan by clicking Abort Scan. You can modify the search pattern and repeat the search as many times as you need until you see all of the expected hosts.

   Note: scans hosts by checking for network connectivity. If there are some hosts where you want to install services that are not shown in the list, make sure you have network connectivity between the Server host and those hosts, and that firewalls and SE Linux are not blocking access.
3. Verify that the number of hosts shown matches the number of hosts where you want to install services. Clear host entries that do not exist or where you do not want to install services.
4. Click Continue.

   The Select Repository screen displays.

Select Repository

Important: You cannot install software using both parcels and packages in the same cluster.

The Select Repository page allows you to specify repositories for Agent and Other software.

In the Agent section:

1. Select either Repository or Custom Repository for the Agent software.
2. If you select Custom Repository, do not include the operating system-specific paths in the URL. For instructions on setting up a custom repository, see *Configuring a Local Package Repository*.

In the Other software section:

1. Select the repository type to use for the installation. In the Install Method section select one of the following:
   - Use Parcels (Recommended)

     A parcel is a binary distribution format containing the program files, along with additional metadata used by . Parcels are required for rolling upgrades. For more information, see *Parcels*.
   - Use Packages

     A package is a standard binary distribution format that contains compiled code and meta-information such as a package description, version, and dependencies. Packages are installed using your operating system package manager.

     Note: Packages are not supported for 7.0 and higher.
2. Select the version of or CDH to install. If you do not see the version you want to install:
   - Parcels – Click the Parcel Repository & Network Settings link to add the repository URL for your version. If you are using a local Parcel repository, enter its URL as the repository URL.

     Repository URLs for CDH 6 parcels are documented in CDH 6 Download Information

Repository URLs for the 7 parcels are documented in Download Information

Important: If you are using a 60-day trial license, use the following Parcel Repository URL (authentication not required):

```
https://archive.cloudera.com/cdh7/7.7.1/parcels/
```

Under Other Software CDH version , if you do not view the parcel that you are intending to install, click Parcel Respositories and Network Settings and manually add the parcel repository location from the Download Information page. For example, if you want to install the 7.1.7 Service Packs, you can pick up the parcel repository location information from Download Information page.

After adding the repository, click Save Changes and wait a few seconds for the URL to be validated. If your host uses an HTTP proxy, click the Proxy Settings button to configure your proxy.

Note that if you have a Enterprise license and are using 6.3.3 or higher to install a CDH version 6.3.3 or higher, or a version 7.0 or higher using parcels, you do not need to add a username and password or "@" to the parcel repository URL. will authenticate to the archive using the information in your license key file. Use a link to the repository in the following format:

```
https://archive.cloudera.com/p/cdh6/6.X.X/parcels/
```

If you are using a version of older than 6.3.3 to install CDH 6.3.3 or higher parcels, you must include the username/password and "@" in the repository URL during installation or when you configure a CDH 6.3.3 or higher parcel repository. After you add the repository, click Save Changes and wait a few seconds for the version to appear. If your host uses an HTTP proxy, click the Proxy Settings button to configure your proxy.

Note: only displays CDH versions it can support. If an available CDH version is too new for your version, it is not displayed. If the parcels do not appear on the Parcels page, ensure that the Parcel URL you entered is correct.

- Packages – If you selected Use Packages, and the version you want to install is not listed, you can select Custom Repository to specify a repository that contains the desired version. Repository URLs for CDH 6 version are documented in CDH 6 Download Information,

  If you are using a local package repository, enter its URL as the repository URL.

Note: only displays CDH or versions it can support. If an available version is too new for your version, it is not displayed.

3. If you selected Use Parcels, specify any Additional Parcels you want to install.
4. Click Continue.

Select JDK

Note: is no longer bundled with Oracle JDK software. provides a supported version of OpenJDK.

If you installed your own JDK version, such as Oracle JDK 8, in *Step 2: Install Java Development Kit*, select Manually manage JDK.

To allow to automatically install the OpenJDK on cluster hosts, select Install a -provided version of OpenJDK.

To install the default OpenJDK that is provided by your operating system, select Install a system-provided version of OpenJDK.

After selecting the applicable boxes, click Continue.

Enter Login Credentials

1. Enter the root name or username for the root account that has password-less sudo privileges. (In the /etc/sudoers file, the entry for this should like this:

   ```
   %<USERNAME> ALL=(ALL) NOPASSWD: ALL
   ```

2. Select an authentication method:

   - If you select the All hosts accept same password option for password authentication, enter and confirm the password.
   - If you select the All hosts accept same private key option for public-key authentication, provide a passphrase and path to the required private key files.

     Generate keys in PEM format by running the following command:

     ```
     ssh-keygen -m pem -t rsa -f ~/.ssh/id_rsa_pem
     scp ~/.ssh/id_rsa_pem.pub HOST:.ssh/
     ssh HOST 'cat .ssh/id_rsa_pem.pub >> ~/.ssh/authorized_keys'
     ```

     Note: In the above command HOST is the hostname of a host in the cluster. You must run the second and third command lines on every host in the cluster.

   You can modify the default SSH port if necessary.
3. Specify the maximum number of host installations to run at once. The default and recommended value is 10. You can adjust this based on your network capacity.
4. Click Continue.

The Install Agents page displays.

Install Agents

The Install Agents page displays the progress of the installation. You can click on the Details link to view the installation log of any host. Optionally, you can click the Abort Installation button to cancel the installation and then view the installation logs to troubleshoot the problem.

If the installation fails on any hosts, you can click the Retry Failed Hosts to retry all failed hosts.

After installing the Agent on all hosts, click Continue.

The Install Parcels page displays.

Inspect Cluster

The Inspect Cluster page provides a tool for inspecting network performance as well as the Host Inspector to search for common configuration problems. recommends that you run the inspectors sequentially:

1. Run the Inspect Network Performance tool. You can click Advanced Options to customize some ping parameters.
2. After the network inspector completes, click Show Inspector Results to view the results in a new tab.
3. Address any reported issues, and click Run Again (if applicable).
4. Click Inspect Hosts to run the Host Inspector utility.
5. After the host inspector completes, click Show Inspector Results to view the results in a new tab.
6. Address any reported issues, and click Run Again (if applicable).



Important:

If you encounter any issues and need any guidance during this process, contact support for further assistance.

If the reported issues cannot be resolved in a timely manner, and you want to abandon the cluster creation wizard to address them, then click Cancel.

After addressing any identified problems, select I understand the risks of not running the inspections or the detected issues, let me continue with cluster setup, and then click Finish.

This completes the Add Cluster wizard operation and launches the Add Cluster - Configuration wizard.

Continue to *Step 8: Set Up a Cluster Using the Wizard*.

Set Up a Cluster Using the Wizard

After you complete the Add Cluster - Installation wizard, the Add Cluster - Configuration wizard automatically starts. The following sections guide you through each page of the wizard.

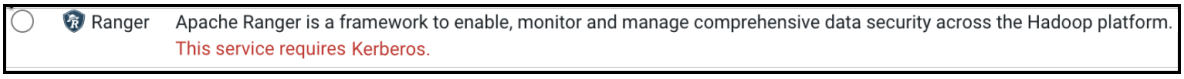
Select Services

The Select Services page allows you to select the services you want to install and configure.

Important:

In case you deploy a scratch cluster that is not yet kerberised, you cannot select any service that requires Kerberos as a prerequisite including Ranger, RangerKMS, Atlas, and Knox.

Each of these services carries such a note near it (see example for Ranger):



Selecting either of these services to be installed without having a Kerberos based cluster leads to a non-functioning cluster. It is advised to install these services separately only after enabling Kerberos on the cluster. For more information, see Set up a KDC.

Note that if you do not add Kerberos to the cluster at the beginning of the wizard (for example, setting Cloudera Manager as Key Distribution Center), you need to selct only Customer Services option and pick the non-kerberos services you want to deploy. Post enabling Kerberos, you need to repeat the wizard and complete the Kerberos based services.

After selecting the services you want to add, click Continue. The **Assign Roles** page displays.

Select one of the following options:

Regular (Base) Clusters

Cloudera Data Engineering

Process develop, and serve predictive models.

Services included: HDFS, YARN, YARN Queue Manager, Ranger, Atlas, Hive, Hive on Tez, Spark, Oozie, and Hue

Data Mart

Browse, query, and explore your data in an interactive way.

Services included: HDFS, Ranger, Atlas, Hive, and Hue

Cloudera Operational Database

Real-time insights for modern data-driven business.

Services included: HDFS, Ranger, Atlas, and HBase

Custom Services

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

Assign Roles

The Assign Roles page suggests role assignments for the hosts in your cluster.

You can click on the hostname for a role to select a different host. You can also click the View By Host button to see all the roles assigned to a host.

After assigning all of the roles for your services, click Continue. The Setup Database page displays.

Setup database

Setting up a database involves providing information such as the database hosts, names, usernames, and passwords of the database you have installed and configured, for each of the services you selected.

Important:

Currently the Cloudera Manager wizard does not support connection testing to an external Oracle as Metastore DB for the Hue/Yarn Queue Manager services.

In case the Cloudera Manager wizard is being used to deploy a fresh cluster, do not install Hue or Yarn Queue Manager, and once the Cloudera Manager wizard finishes, follow the guidelines of each service on how to manually set it to work with Oracle.

For services that support the database, you can add finer-grained customizations using a JDBC URL override.

Select the database type and enter the database name, username, and password for each service.

For MariaDB, select MySQL.

For services that support it, to specify a JDBC URL override, select Yes in the Use JDBC URL Override drop-down menu. You must also specify the database type, username, and password.

If you are using a TLS 1.2-enabled MySQL, PostreSQL, or MariaDB databases, or TCPS-enabled Oracle database for your Cloudera Runtime service, then see the service-specific instructions under Database setup details for cluster services for TLS/TCPS-enabled databases on page 165.

Tip: MySQL 8 uses caching_sha2_password as the default authentication method. Follow the instructions in Database setup details for Hive Metastore for TLS 1.2/TCPS-enabled databases for proper configuration.

Click Test Connection to validate the settings. If the connection is successful, a green checkmark and the word Successful appears next to each service. If there are any problems, the error is reported next to the service that failed to connect.

After verifying that each connection is successful, click Continue. The Review Changes page displays.

Database setup details for cluster services for TLS/TCPS-enabled databases

Cloudera Manager supports TLS connection to backend databases for Hadoop services such as Hue, Ranger, Oozie etc. You must modify the configuration of the services in Cloudera Manager to set a TLS connection with their respective databases.

Database setup details for Ranger KMS for TLS/TCPS-enabled databases

Updating the Ranger KMS database JDBC URL override and additional configuration to connect to the secure databases.

Before you begin

- Ensure that TLS has already been enabled on the Ranger database.
- Ensure Use JDBC Override URL has been set to Yes in the Setup Database page.

Procedure

1. Select/Enter the following configuration values depending on the database type.
   - MySQL

Label	Configuration	Value
Ranger KMS Database Type	ranger_kms_database_type	MySQL

Label	Configuration	Value
Ranger KMS Database User	ranger_kms_database_user	<username>
Ranger KMS Database User Password	ranger_kms_database_password	<password>
Ranger KMS Database JDBC Url Override	ranger_kms_database_jdbc_url	jdbc:mysql:///*[***DB-HOST***]*:*[***DB-PORT***]*/*[***RANGER-KMS-DB-NAME***]*?sslMode=VERIFY_CA&trustCertificateKeyStoreU version>

- Oracle

Label	Configuration	Value
Ranger KMS Database Type	ranger_kms_database_type	Oracle
Ranger KMS Database User	ranger_kms_database_user	<username>
Ranger KMS Database User Password	ranger_kms_database_password	<password>
Ranger KMS Database JDBC Url Override	ranger_kms_database_jdbc_url	jdbc:oracle:thin:@tcps:///*[***DB-HOST***]*:*[***DB-PORT***]*:*[***SERVICE_NAME***]*?javax.net.ssl.trustStore=*[***PATH_TO_TRUSTST*

- PostgreSQL

Label	Configuration	Value
Ranger KMS Database Type	ranger_kms_database_type	PostgreSQL
Ranger KMS Database User	ranger_kms_database_user	<username>
Ranger KMS Database User Password	ranger_kms_database_password	<password>
Ranger KMS Database JDBC Url Override	ranger_kms_database_jdbc_url	jdbc:postgresql:///*[***DB-HOST***]*:*[***DB-PORT***]*/*[***RANGER-KMS-DB***]*?sslmode=verify-full&sslrootcert=*[***PATH-TO-DATABASE-SERVER-CERTIFICATE***]*&enabledTLSProtocols=<TLS version>

2. Click Test Connection.
3. Once the test connection succeeds, click Continue.

Database setup details for Ranger for TLS/TCPS-enabled databases

Updating the Ranger database JDBC URL override and additional configuration to connect to the secure databases.

Before you begin

- Ensure that TLS has already been enabled on the Ranger database.
- Ensure Use JDBC Override URL has been set to Yes in the Setup Database page.

Procedure

1. Select / Enter the following configuration values depending on the database type.
   - MySQL

Label	Configuration	Value
Ranger Database Type	ranger_database_type	MySQL
Ranger Database User	ranger_database_user	<username>
Ranger Database User Password	ranger_database_password	<password>
Ranger Database JDBC Url Override	ranger_database_jdbc_url	jdbc:mysql:///*[***DB-HOST***]*:*[***DB-PORT***]*/*[***RANGER-DB-NAME***]*?sslMode=VERIFY_CA&trustCertificateKeyStoreV version>

   - Oracle

Label	Configuration	Value
Ranger Database Type	ranger_database_type	Oracle
Ranger Database User	ranger_database_user	<username>
Ranger Database User Password	ranger_database_password	<password>
Ranger Database JDBC Url Override	ranger_database_jdbc_url	dbc:oracle:thin:@tcps://*[***DB-HOST***]*:*[***DB-PORT***]*:*[***SERVICE_NAME***]*?javax.net.ssl.trustStore=*[***PATH_TO_TRUSTST*

   - PostgreSQL

Label	Configuration	Value
Ranger Database Type	ranger_database_type	PostgreSQL
Ranger Database User	ranger_database_user	<username>
Ranger Database User Password	ranger_database_password	<password>
Ranger Database JDBC Url Override	ranger_database_jdbc_url	jdbc:postgresql:///*[***DB-HOST***]*:*[***DB-PORT***]*/*[***RANGER-DB***]*?sslmode=verify-full&sslrootcert=*[***PATH-TO-DATABASE-SERVER-CERTIFICATE***]*&enabledTLSProtocols=<TLS version>

2. Click Test Connection.
3. Once the test connection succeeds, click Continue.

Database setup details for Oozie for TLS/TCPS-enabled databases

You can configure Oozie's database connection properties after you add Oozie as a service.

Procedure

1. Add Oozie as a service.

2. In the configuration wizard, configure the database type, host, port, and database name.
3. Before starting Oozie, fine-tune your Oozie's database connection, including adding your database client trustStore or certificate location to the connection string.

   For more details, see *Fine-tuning Oozie's database connection*.

Database setup details for Hive Metastore for TLS/TCPS-enabled databases

Learn how you can configure the Hive (Hive Metastore) service to securely connect to its database using TLS 1.2 or a higher version when installing a new cluster.

About this task

When installing a new cluster, Hive Metastore's database connection is set up during the **Setup Database** step of the **Add Cluster - Configuration** wizard. Complete the following when you reach this step in the wizard to configure TLS 1.2 or a higher version.



Note: Cloudera supports only TLS 1.2 and 1.3 for connections to TLS-enabled or TCPS-enabled databases.

Before you begin

- Ensure that TLS 1.2 or a higher version is enabled on the Hive Metastore database.
- Ensure that the Database SSL trusted certificates are exported and added to the Java truststore file.

Procedure

1. In the **Setup Database** page for Hive, click the Type drop-down list and select the appropriate database type.
2. Click the Use JDBC URL Override drop-down list and select Yes.
3. In the JDBC URL field, specify the appropriate JDBC URL connection string.

   The JDBC URL must contain all necessary properties required for Hive Metastore to establish a secure connection with its database. Use the following templates to construct the JDBC URL:
   MySQL

```
jdbc:mysql://[***DB-HOST***]:[***DB-PORT***]/[***DB-NAME***]?ssl
Mode=VERIFY_CA&trustCertificateKeyStoreUrl=file://[***TRUSTSTORE
-PATH***]&trustCertificateKeyStoreType=jks&trustCertificateKeySt
orePassword=[***TRUSTSTORE-PASSWORD***]&enabledTLSProtocols=TLSv
1.2
```

   PostgreSQL

```
jdbc:postgresql://[***DB-HOST***]:[***DB-PORT***]/[***DB-NAME***
]?sslMode=VERIFY_CA&trustCertificateKeyStoreUrl=file://[***TRUST
STORE-PATH***]&trustCertificateKeyStoreType=jks&trustCertificate
KeyStorePassword=[***TRUSTSTORE-PASSWORD***]&enabledTLSProtocols
=TLSv1.2
```

   MariaDB

```
jdbc:mysql://[***DB-HOST***]:[***DB-PORT***]/[***DB-NAME***]?ssl
Mode=VERIFY_CA&trustCertificateKeyStoreUrl=file://[***TRUSTSTORE
-PATH***]&trustCertificateKeyStoreType=jks&trustCertificateKeySt
orePassword=[***TRUSTSTORE-PASSWORD***]&enabledTLSProtocols=TLSv
1.2
```

   Oracle

```
jdbc:oracle:thin:@tcps://[***DB-HOST***]:[***DB-PORT***]/[***DB-
NAME***]?javax.net.ssl.trustStore=[***TRUSTSTORE-PATH***]&javax.
```

```
net.ssl.trustStorePassword=[***TRUSTSTORE-PASSWORD***]&oracle.ne
t.ssl_server_dn_match=false
```

Where,

- [***DB-HOST***], [***DB-PORT***], and [***DB-NAME***] represent the Host, Port, and Database name used for the Hive Metastore service.
- [***TRUSTSTORE-PATH***] represents the path to the Java truststore file.
- [***TRUSTSTORE-PASSWORD***] represents the password used to access the Java truststore file.

4. Click Test Connection to validate the settings.

   If the connection fails, review your configuration, fix any errors, and test the connection again.
5. Click Continue to proceed with the installation.

Results

The Hive Metastore service establishes a secure connection with its database.

Database setup details for Cloudera Data Explorer (Hue) for TLS/TCPS-enabled databases

Data Explorer automatically uses +TLSv1.2 +TLSv1.3 when you create a Cloudera cluster with the Auto-TLS option. You can also enable TLS in Data Explorer configurations after you add the Data Explorer service to your cluster using Cloudera Manager.

About this task

If TLS 1.2 is enabled on the database servers, and the databases are restricted or enforced to use these protocols, then Data Explorer automatically uses the compatible ciphers to communicate with the database securely. You do not have to configure any setting in Hue's Advanced Configuration Snippet or any other configurations. This is applicable when using MySQL, MariaDB, or PostgreSQL databases as a backend database for Data Explorer.

The following TLS configurations are supported:

- TLSv1.2: Enables TLS 1.2 exclusively.
- TLSv1.3: Enables TLS 1.3 exclusively.
- TLSv1.2 and TLSv1.3 (Default): Enables both TLS 1.2 and TLS 1.3 simultaneously.

To restrict the MySQL and MariaDB databases to use TLS 1.2, set the value of the require_secure_transport to true in the my.cnf file.

To enable TLS versions on the Data Explorer instance, go to Cloudera Manager Clusters Data Explorer (Hue) Configurations and select the Enable TLS/SSL for Hue option.

If TCPS is enabled on the Oracle database, then you can specify the connection string containing the “TCPS” protocol in the Database SID field.

Before you begin

- You must have enabled TLS 1.3, or TCPS on the Data Explorer (Hue) database.
- You must have created database users.
- You must have installed the MySQL client (for MySQL or MariaDB databases).
- You must have installed the psycopg2 Python package (for PostgreSQL database).

Procedure

1. Select the appropriate database type from the Type drop-down menu.
2. Enter the Fully Qualified Domain Name (FQDN) of the host on which you have installed the database in the Database Hostname field.

   If the database is not running on its default port, then specify the port number in the following format: *[***HUE-DB-HOST***]:[***DB-PORT***]*.

   Where,

*[***HUE-DB-HOST***]* is the FQDN of the database host

*[***DB-PORT***]* is the database port

3. Specify the database name in the Database Name field.

   For Oracle databases, specify the SID in the Database SID field.

   If you are using Oracle Service Name instead of SID, then specify the database name in the following format:

   ```
   [***HUE-DB-HOST***]:1521/[***SERVICE-NAME***]
   ```

   Where,

   *[***HUE-DB-HOST***]* is the FQDN of the database host

   *[***SERVICE-NAME***]* is the Oracle service name

   You can also specify the following connection string in the Database Name field:

   ```
   (DESCRIPTION=(LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=5)(TRANSPOR
   T_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS=(PROTOCOL=TCPS)(HOST=[***HUE-
   DB-HOST***])(PORT=[***HUE-DB-PORT***]))(CONNECT_DATA=(SERVICE_NAM
   E=[***SERVICE-NAME***])(SECURITY =
    (MY_WALLET_DIRECTORY = /[***PATH-TO-WALLET-FILE***])))
   ```

   Where,

   *[***HUE-DB-HOST***]* is the FQDN of the database host

   *[***HUE-DB-PORT***]* is the port for the Data Explorer (Hue) database

   *[***SERVICE-NAME***]* is the Oracle service name

   *[***PATH-TO-WALLET-FILE***]* is the location at which you have copied the wallet file (cwallet.sso) on the Data Explorer (Hue) host

4. Enter the database username and password you set up for Data Explorer (Hue) database in the Username and Password fields.
5. Click Test Connection.

   If the connection test fails, review your configuration, fix any errors, and rerun the connection test.
6. Click Continue to continue with cluster installation.

Database setup details for Streams Messaging Manager for TLS/TCPS-enabled databases

Learn how you can configure Streams Messaging Manager to securely connect to its database when installing a new cluster.

About this task

When installing a new cluster, the Streams Messaging Manager database connection is set up during the Setup Database step of the Add Cluster - Configuration wizard. Complete the following when you reach this step in the wizard to configure TLS.

Before you begin

- Ensure that TLS has already been enabled on the Streams Messaging Manager database.
- Ensure that a truststore file containing the database certificate is available on the Streams Messaging Manager hosts. Additionally, ensure that you know the location of the file and that the user running Streams Messaging Manager has access to the file. The default user for Streams Messaging Manager is streamsmsgmgr.

Procedure

1. Select the appropriate database type from the Type drop-down list.
2. Select yes from the Use JDBC URL Override drop-down list.
3. Enter the database username and password you set up for Streams Messaging Manager in the Username and Password fields.

   The username and password you must enter are set up in a previous step of the installation. For more information, see *Configuring the Database for Streaming Components*.
4. Enter an appropriate JDBC URL in the JDBC URL field.

   The JDBC URL must contain all necessary properties needed for Streams Messaging Manager to establish a secure connection with its database. Use the following templates to construct the JDBC URL.
   MySQL

   ```
   jdbc:mysql://[***DB HOST***]:[***DB PORT***]/[***DB NAME***]?
   useSSL=true&trustCertificateKeyStoreUrl=file://[***TRUSTSTORE
    PATH***]&trustCertificateKeyStoreType=jks&trustCertificateKeySto
   rePassword=[***TRUSTSTORE PASSWORD***]&enabledTLSProtocols=<TLS-
   version>
   ```

   PostgreSQL

   ```
   jdbc:postgresql://[***DB HOST***]:[***DB PORT***]/[***DB
    NAME***]?useSSL=true&trustCertificateKeyStoreUrl=fil
   e://[***TRUSTSTORE PATH***]&trustCertificateKeyStoreType=jks&tr
   ustCertificateKeyStorePassword=[***TRUSTSTORE PASSWORD***]&ena
   bledTLSProtocols=<TLS-version>
   ```

   Oracle

   ```
   jdbc:oracle:thin:@tcps://[***DB HOST***]:[***DB PORT***]/[***DB
    NAME***]?javax.net.ssl.trustStore=[***TRUSTSTORE PATH***]&jav
   ax.net.ssl.trustStorePassword=[***TRUSTSTORE PASSWORD***]&oracle.
   net.ssl_server_dn_match=false
   ```

   - Replace *[***DB HOST***]*, *[***DB PORT***]*, and *[***DB NAME***]* with the host, port, and name of the database.
   - Replace *[***TRUSTSTORE PATH***]* with the full path to a truststore that contains the database certificate. The truststore must be available on the host that Streams Messaging Manager is deployed on. Additionally, the user that the Streams Messaging Manager service runs as, default is streamsmsgmgr, must have access to the file.
   - Replace *[***TRUSTSTORE PASSWORD***]* with the password used to access the truststore you specify in *[***TRUSTSTORE PATH***]*.
5. Click Test Connection.

   If the connection test fails, review your configuration, fix any errors, and rerun the connection test.
6. Click Continue to continue with cluster installation.

Results

The Streams Messaging Manager service establishes a secure connection with its database.

Database setup details for Schema Registry for TLS/TCPS-enabled databases

Learn how you can configure Schema Registry to securely connect to its database when installing a new cluster.

About this task

When installing a new cluster, Schema Registry's database connection is set up during the Setup Database step of the Add Cluster - Configuration wizard. Complete the following when you reach this step in the wizard to configure TLS.

Before you begin

- Ensure that TLS has already been enabled on the Schema Registry database.
- Ensure that a truststore file containing the database certificate is available on the Schema Registry hosts. Additionally, ensure that you know the location of the file and that the user running Schema Registry has access to the file. The default user for Schema Registry is schemaregistry.

Procedure

1. Select the appropriate database type from the Type drop-down list.
2. Select yes from the Use JDBC URL Override drop-down list.
3. Enter the database username and password you set up for Schema Registry in the Username and Password fields.

   The username and password you must enter are set up in a previous step of the installation. For more information, see *Configuring the Database for Streaming Components*.
4. Enter an appropriate JDBC URL in the JDBC URL field.

   The JDBC URL must contain all necessary properties needed for Schema Registry to establish a secure connection with its database. Use the following templates to construct the JDBC URL.
   MySQL

   ```
   jdbc:mysql://[***DB HOST***]:[***DB PORT***]/[***DB NAME***]?
   useSSL=true&trustCertificateKeyStoreUrl=file://[***TRUSTSTORE
    PATH***]&trustCertificateKeyStoreType=jks&trustCertificateKeySto
   rePassword=[***TRUSTSTORE PASSWORD***]&enabledTLSProtocols=<TLS-
   version>
   ```

   PostgreSQL

   ```
   jdbc:postgresql://[***DB HOST***]:[***DB PORT***]/[***DB
    NAME***]?useSSL=true&trustCertificateKeyStoreUrl=fil
   e://[***TRUSTSTORE PATH***]&trustCertificateKeyStoreType=jks&tr
   ustCertificateKeyStorePassword=[***TRUSTSTORE PASSWORD***]&ena
   bledTLSProtocols=<TLS-version>
   ```

   Oracle

   ```
   jdbc:oracle:thin:@tcps://[***DB HOST***]:[***DB PORT***]/[***DB
    NAME***]?javax.net.ssl.trustStore=[***TRUSTSTORE PATH***]&jav
   ax.net.ssl.trustStorePassword=[***TRUSTSTORE PASSWORD***]&oracle.
   net.ssl_server_dn_match=false
   ```

   - Replace *[***DB HOST***]*, *[***DB PORT***]*, and *[***DB NAME***]* with the host, port, and name of the database.
   - Replace *[***TRUSTSTORE PATH***]* with the full path to a truststore that contains the database certificate. The truststore must be available on the host that Schema Registry is deployed on. Additionally, the user that the Schema Registry service runs as, default is schemaregistry, must have access to the file.
   - Replace *[***TRUSTSTORE PASSWORD***]* with the password used to access the truststore you specify in *[***TRUSTSTORE PATH***]*.
5. Click Test Connection.

   If the connection test fails, review your configuration, fix any errors, and rerun the connection test.
6. Click Continue to continue with cluster installation.

Results

The Schema Registry service establishes a secure connection with its database.

Enter Required Parameters

The **Enter Required Parameters** page lists required parameters for the Cloudera Manager API client, Hive, and Ranger.

Cloudera Manager API Client

If you do not have an existing user for the Cloudera Manager API client, use the default username and password "admin" for both the The Existing Cloudera Manager API Client Username and The Existing Cloudera Manager API Client Password.

Hive

If your database supports TLS connections, then configure the following parameters:

- Enable TLS/SSL to the Hive Metastore Database parameter,
- Set the Hive Metastore Client SSL/TLS Trust Store File parameter to a JKS truststore file that contains a CA certificate trusting the database's certificate.
- Set the Hive Metastore Client SSL/TLS Trust Store Password parameter to that truststore's password.

Review Changes

The Review Changes page lists default and suggested settings for several configuration parameters, including data directories.



Warning: Do not place DataNode data directories on NAS devices. When resizing an NAS, block replicas can be deleted, which results in missing blocks.

Review and make any necessary changes, and then click Continue. The Command Details page displays.

Configure Kerberos

Kerberos is a network authentication protocol that provides security for your cluster. The Configure Kerberos page allows you to enable Kerberos for your cluster.



Important:
Limitation in Kerberos library

Occasionally, when using Kerberos authentication in certain environments, the CDP Runtime cluster refuses to start.

To fix the problem, you must add this setting UDP_preference_limit=1 to every cluster host’s krb5.conf file which is typically located in /etc/krb5.conf.

After selecting Enable kerberos for this cluster, install kerberos client libraries, according to your OS type, on all hosts before proceeding.

- RHEL/ CentOS

```
$ yum install krb5-workstation krb5-libs
```

If Redhat IPA is used as the KDC,

```
yum install freeipa-client
```

- SUSE

```
zypper install krb5-client
```

If Redhat IPA is used as the KDC,

```
zypper install freeipa-client
```

- Ubuntu

```
apt-get install krb5-user
```

If Redhat IPA is used as the KDC,

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

Configure Datanode ports

Configure the privileged ports required by the datanodes in a secure HDFS service by selecting values for DataNode Transceiver Port and DataNode HTTP Web UI port.

Command Details

The Command Details page lists the details of the First Run command.

You can expand the running commands to view the details of any step, including log files and command output. You can filter the view by selecting Show All Steps, Show Only Failed Steps, or Show Only Running Steps.

If cluster deployment fails, ensure to click Resume in the wizard after you fix any issues. If you do not click Resume, the cluster may not be in a functional state.

After the First Run command completes, click Continue to go to the Summary page.

Summary

The Summary page reports the success or failure of the setup wizard.

Click Finish to complete the wizard. The installation is complete.

Cloudera recommends that you change the default password as soon as possible by clicking the logged-in username at the top right of the home screen and clicking Change Password.

(optional) Enable high availability for Cloudera Manager

After you have set up a cluster, you can enable high availability for Cloudera Manager. See Configuring Cloudera Manager for High Availability.

(Recommended) Enable Auto-TLS

Auto-TLS greatly simplifies the process of enabling and managing TLS encryption on your cluster.

For information on using Auto-TLS to simplify the process of configuring TLS encryption for Cloudera Manager, see *Configuring TLS Encryption for Cloudera Manager Using Auto-TLS*.

Related Information

Configuring TLS Encryption for Cloudera Manager Using Auto-TLS

(Recommended) Enable Kerberos

Kerberos is an authentication protocol that relies on cryptographic mechanisms to handle interactions between a requesting client and server, greatly reducing the risk of impersonation.

For information on enabling Kerberos, see *Enabling Kerberos Authentication for Cloudera.*



Note: Authorization through Apache Ranger is just one element of a secure production cluster: Cloudera supports Ranger only when it runs on a cluster where Kerberos is enabled to authenticate users.

Related Information

Enabling Kerberos Authentication for Cloudera.

Additional Steps for Apache Ranger

After installing Cloudera Manager and adding a cluster, there are additional steps required to complete the installation of Apache Ranger.

Related Information

Configure a resource-based policy: Solr

Enabling Solr clients to authenticate with a secure Solr

Update Ranger audit configration parameters

Enable Plugins

About this task

The Ranger plugins for HDFS and Solr may not be enabled by default. Ranger plugins enable Cloudera Manager stack components – such as HDFS and Solr – to connect to Ranger and access its authorization and audit services. Verify that the HDFS and Solr plugins are enabled after you install and start the Ranger service.

Procedure

1. To enable the HDFS plugin:
   a) Log in to Cloudera Manager.
   b) Go to the HDFS Service status page.
   c) Click the Configuration tab.
   d) Search for the Enable Ranger Authorization configuration property.
   e) If the Enable Ranger Authorization property is not selected, select it and save the changes.
   f) Go to the Ranger Service status page and click ActionsSetup Ranger Plugin Service.
   g) Restart the HDFS service.
2. To enable the Ranger Solr plugin:
   a) Log in to Cloudera Manager.
   b) Go to the Solr Service status page.
   c) Click the Configuration tab.
   d) Search for the Enable Ranger Authorization configuration property.
   e) If the Enable Ranger Authorization property is not selected, select it and save the changes.

      Note: Do not select the Ranger Service dependency parameter. This is used for enabling a Solr service instance that is not used by the Ranger service.

   f) Restart the Solr service.

Add Solr WebUI Users

Procedure

Add the username of any users to the Ranger Solr policy who should have access to the Solr Web UI in the Ranger Policy for Solr. The user should have full access privileges.

Update the Time-to-live configuration for Ranger Audits

How to change the default time settings that control how long Ranger keeps audit data collected by solr.

Procedure

1. From Cloudera Manager choose Ranger Configuration .
2. Search for the ranger.audit.solr.config.ttl property, and set the the number of days to keep audit data.
3. Search for the ranger.audit.solr.config.delete.trigger property, and set the number and units (days, minutes, hours, or seconds) to keep data for expired documents
4. Refresh the configuration, using one of the following two options:
   a) Click Refresh Configuration, as prompted or, if Refresh Configuration does not appear,
   b) In Actions, click Update Solr config-set for Ranger, then confirm.

Determine susceptibility of Ranger database upgrade failure

This check is essential for you if you are still on releases earlier than 7.3.2.0 (7.3.1.x, 7.2.x, or 7.1.x) and are using MySQL 8.0 or higher.

Procedure

1. Verify x_trx_log table existence.

   Execute the following SQL query in your Ranger database:

   ```
   SHOW CREATE TABLE x_trx_log;
   ```

Condition	Action
Table x_trx_log does not exist	The upgrade will not fail due to this issue. No further action is required.
Table x_trx_log exists	Proceed to step 2 to check the table's default character set (look for DEFAULT CHARSET in the output).

2. Check table’s default character set.

   Inspect the previous command output for x_trx_log’s DEFAULT CHARSET or the trx_id column’s collation.

Condition	Action
DEFAULT CHARSET is not utf8mb4	The upgrade is unlikely to fail due to this specific issue. No further action is required.
DEFAULT CHARSET is utf8mb4	If the default charset is set to utf8mb4 then, your environment is susceptible. Proceed to Step 3.

3. Verify index existence (x_trx_log_IDX_trx_id).

   If the x_trx_log table exists and the default charset is set to utf8mb4, check if the problematic index already exists:

   ```
   SELECT * FROM INFORMATION_SCHEMA.STATISTICS WHERE table_schema=DATABASE()
    AND table_name='x_trx_log' AND index_name='x_trx_log_IDX_trx_id';
   ```

Condition	Action
Index x_trx_log_IDX_trx_id exists	The upgrade will not fail due to this issue. No further action is required.
Index x_trx_log_IDX_trx_id does not exist	Proceed to the remediation step.

Remediation steps

Create the required index manually by executing the following SQL statement in the Ranger DB:

```
CREATE INDEX x_trx_log_IDX_trx_id ON x_trx_log (trx_id(190));
```

This prefix index ensures the key length stays within the 3072-byte limit for utf8mb4 environments.

Installing Apache Knox

This document provides instructions on how to install Apache Knox using the Cloudera Base on premises installation process.

About this task

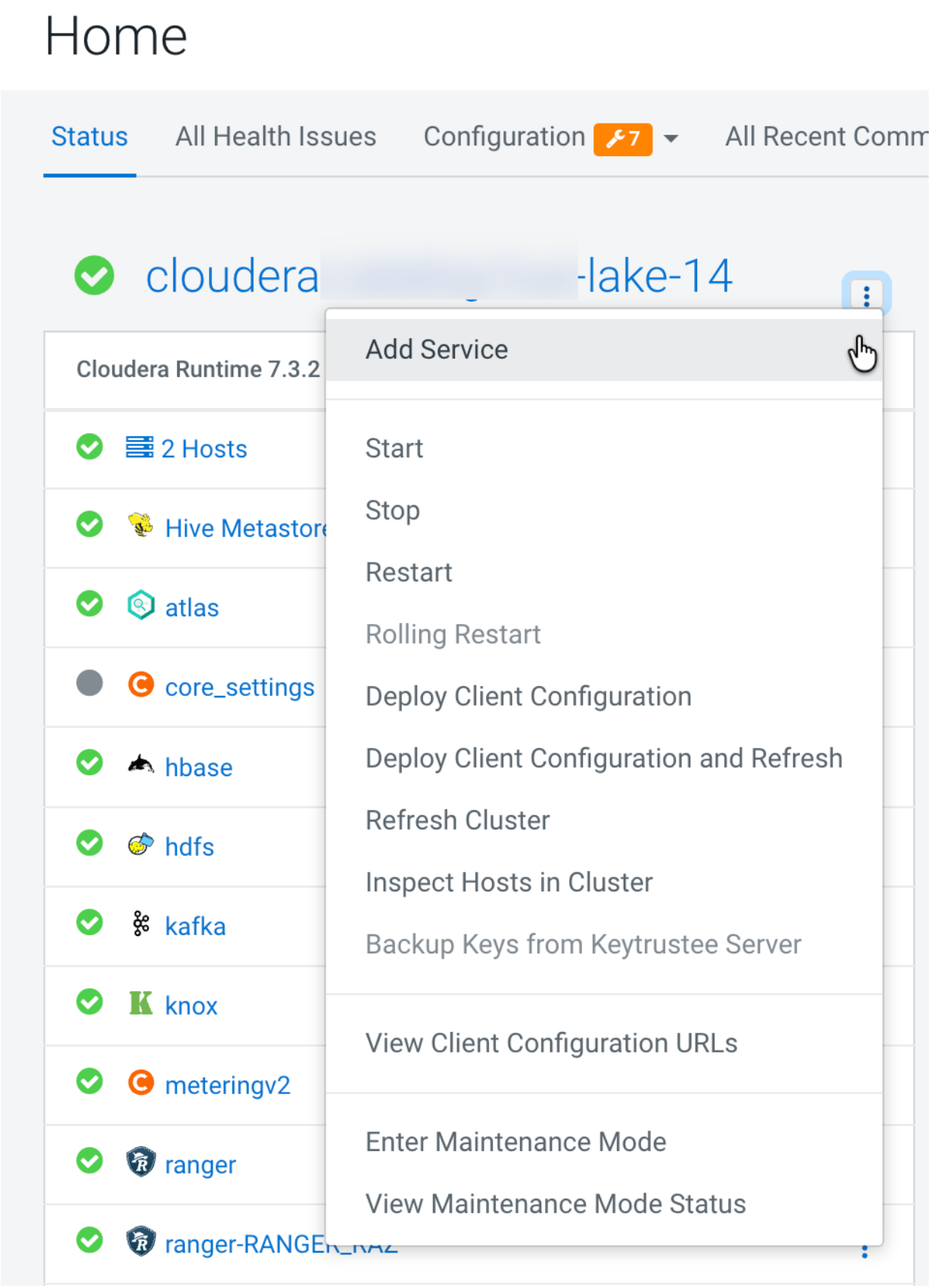
Apache Knox is an application gateway for interacting with the REST APIs and UIs. The Knox Gateway provides a single access point for all REST and HTTP interactions in your Cloudera cluster.

Before you begin

When installing Knox, you must have Kerberos enabled on your cluster.

Procedure

1. From your Cloudera Manager homepage, go to Status tab $Cluster Name ... Add Service



2. From the list of services, select Knox and click Continue.

3. On the **Select Dependencies** page, choose the dependencies you want Knox to set up:

HDFS, Ranger, Solr, Zookeeper	For users that require Apache Ranger for authorization. HDFS with Ranger. HDFS depends on Zookeeper, and Ranger depends on Solr.
HDFS, Zookeeper	HDFS depends on Zookeeper.
No optional dependencies	For users that do not wish to have Knox integrate with HDFS or Ranger.

4. On the **Assign Roles** page, select role assignments for your dependencies and click Continue:

Knox service roles	Description	Required?
Knox Gateway	If Knox is installed, at least one instance of this role should be installed. This role represents the Knox Gateway which provides a single access point for all REST and HTTP interactions with Apache Hadoop clusters.	Required
KnoxIDBroker*	It is strongly recommended that this role is installed on its own dedicated host. As its name suggests this role will allow you to take advantage of Knox's Identity Broker capabilities, an identity federation solution that exchanges cluster authentication for temporary cloud credentials.*	Optional*
Gateway	This role comes with the CSD framework. The gateway structure is used to describe the client configuration of the service on each host where the gateway role is installed.	Optional

* Note: KnoxIDBroker appears in the Assign Roles page, but it is not currently supported in Cloudera Base on premises.

5. On the **Review Changes** page, most of the default values are acceptable, but you must Enable Kerberos Authentication and supply the Knox Master Secret. There are additional parameters you can specify or change, listed in "Knox Install Role Parameters".
   a) Click Enable Kerberos Authentication

      Kerberos is required where Knox is enabled.
   b) Supply the Knox Master Secret, e.g. knoxsecret.
   c) Click Continue.
6. The **Command Details** page shows the status of your operation. After completion, your system admin can view logs for your installation under stdout.

Related Information

Apache Knox install role parameters

Apache Knox install role parameters

Reference information on all the parameters available for Knox service roles.

Service-level parameters

Table 21: Required service-level parameters

Name	In Wizard	Type	Default Value
kerberos.auth.enabled*	Yes	Boolean	false
ranger_knox_plugin_hdfs_audit_directory	No	Text	${ranger_base_audit_url}/knox
autorestart_on_stop	No	Boolean	false

Name	In Wizard	Type	Default Value
knox_pam_realm_service	No	Text	login

Knox Gateway role parameters

Table 22: Required parameters for Knox Gateway role

Name	In Wizard	Type	Default Value
gateway_master_secret	Yes	Password	-
gateway_conf_dir	Yes	Path	/var/lib/knox/gateway/conf
gateway_data_dir	Yes	Path	/var/lib/knox/gateway/data
gateway_security_dir	Yes	Path	/var/lib/knox/gateway/data/security
gateway_port	No	Port	8443
gateway_health_port	No	Port	8443[1]
gateway_path	No	Text	gateway
gateway_heap_size	No	Memory	1 GB (min = 256 MB; soft min = 512 MB)
gateway_ranger_knox_plugin_conf_path	No	Path	/var/lib/knox/ranger-knox-plugin
gateway_ranger_knox_plugin_policy_cache_directory	No	Path	/var/lib/ranger/knox/gateway/policy-cache
gateway_ranger_knox_plugin_hdfs_audit_spool_directory	No	Path	/var/log/knox/gateway/audit/hdfs/spool
gateway_ranger_knox_plugin_solr_audit_spool_directory	No	Path	/var/log/knox/gateway/audit/solr/spool

Table 23: Optional parameters for Knox Gateway role

Name	Type	Default Value
gateway_default_topology_name	Text	cdp-proxy
gateway_auto_discovery_enabled	Boolean	true
gateway_cluster_configuration_monitor_interval	Time	60 seconds (minimum = 30 seconds)
gateway_auto_discovery_advanced_configuration_monitor_interval	Time	10 seconds (minimum = 5 seconds)
gateway_cloudera_manager_descriptors_monitor_interval	Time	10 seconds (minimum = 5 seconds)
gateway_auto_discovery_cdp_proxy_enabled_*	Boolean	true
gateway_auto_discovery_cdp_proxy_api_enabled_*	Boolean	true
gateway_descriptor_cdp_proxy	Text Array	Contains the required properties of cdp-proxy topology
gateway_descriptor_cdp_proxy_api	Text Array	Contains the required properties of cdp-proxy-api topology
gateway_sso_authentication_provider	Text Array	Contains the required properties of the authentication provider used by the UIs using the Knox SSO capabilities (such as Home Page UI). Defaults to PAM authentication.

[1] Set it to the value of the gateway_port if it is not the default value.

Name	Type	Default Value
gateway_api_authentication_provider	Text Array	Contains the required properties of the authentication provider used by pre-defined topologies such as admin, metadata or cdp-proxy-api. Defaults to PAM authentication.
gateway_save_alias_command_input	Text	-

Related Information

Installing Apache Knox

Setting Up Data at Rest Encryption for HDFS

This section describes how to enable end-to-end data encryption to and from HDFS using Ranger KMS. Ranger KMS has two deployment models, depending on your organization's requirements for the separation of duties. Cloudera recommends configuring High Availability (HA) to ensure data is always highly available. To learn more about Ranger KMS, see *Ranger KMS overview*.



Important: Before setting up HDFS Data at Rest encryption, Cloudera highly recommends reading the Encrypting Data at Rest content, which provides more information about HDFS encryption, the supported architecture, planning, encryption requirements, and more.

If you require a third-party HSM for key storage, Cloudera also recommends reading the Integrating Components for Encrypting Data at Rest content.

Links are provided in the Related Information section below.

Depending on your encryption key root trustee requirements, you can enable HDFS encryption as follows:

- Ranger Key Management Service Collocated with Data Cluster, which sources the encryption zone keys from a backing database.
- Ranger Key Management Service Federated Deployment, which has a dedicated security cluster, separated from the data cluster.

Related Information

Ranger KMS overview
Encrypting Data at Rest
Data at Rest Encryption Reference Architecture
Data at Rest Encryption Requirements
Resource Planning for Data at Rest Encryption
Data Encryption Components and Solutions
Working with an HSM for Ranger KMS

Installing Ranger KMS cllocated with data cluster

The tasks and steps for installing the Ranger Key Management System (KMS) that uses a database as the backing key store.

About this task

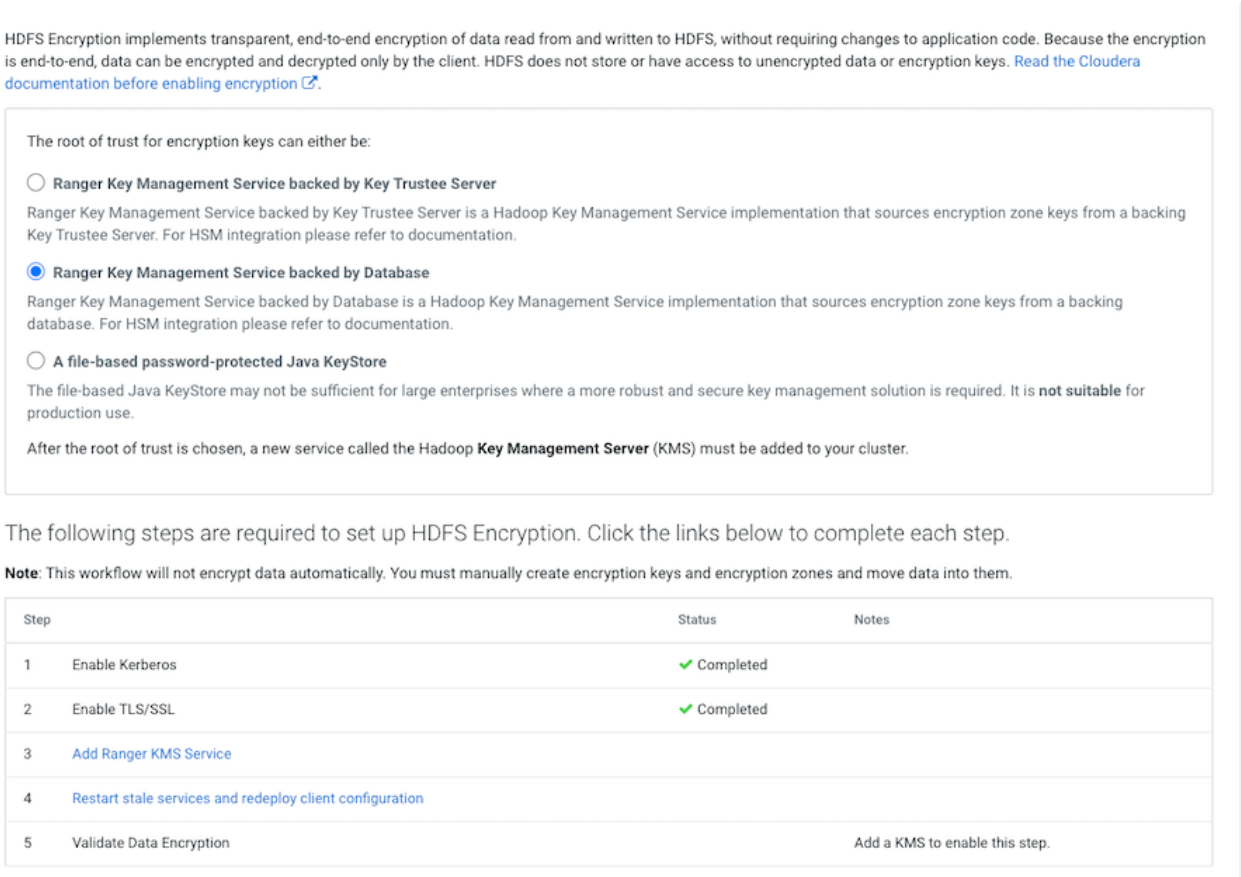
This task uses the Set up HDFS Data At Rest Encryption wizard to install a Ranger KMS that uses a database as the backing key store.

The following image shows the Set up HDFS Data At Rest Encryption page. When you select your encryption keys root of trust option, a list of tasks that you must do to enable encryption to-and-from HDFS is displayed.

You complete each task independently from the other tasks. Where, the task’s Status column indicates whether the step has been completed and the Notes column provides additional context for the task. If your Cloudera Manager

user account does not have sufficient privileges to complete a task, the Notes column indicates the privileges that are required.

When selected, each task contains links to wizards or documentation that help you complete the task. If a task is unavailable, due to insufficient privileges or an incomplete prerequisite step, no links are present and the Notes column displays the reason.



Note: It is assumed that you have already created a database on a server that does not contain the Ranger service.

For more information on how to create a database for Ranger KMS, see the Related Information links below.

The Wizard steps are as follows and must be completed in the order listed:

1. Enable Kerberos

   Note: The instructions assume that you have enabled Kerberos. If this is not the case, click the link associated with the uncompleted task and follow the Wizard's instructions.

2. Enable TLS/SSL

   Note: The instructions assume that you have enabled TLS. If this is not the case, click the link associated with the uncompleted task and follow the Wizard's instructions.

3. Add a Ranger KMS Service
4. Restart the stale services and redeploy the client configuration
5. Validate the Data Encryption

The following lists the post installation tasks for Installing the Ranger KMS backed by a Database and HA:

- Update the Ranger KMS backed by a Database service's URL
- Create a Ranger Audit Directory

Before you begin

Verify the following:

- The cluster in which Cloudera Manager and the Ranger service is installed, is up and running.
- A Ranger KMS database has been created as the underlying keyStore mechanism. This database must be separate from the Ranger database.
- Communication through secure connections is enabled with the Transport Layer Security (TLS) protocol and your network authentication is enabled with the Kerberos protocol.
- You have securely recorded the following backing key store database access credentials, as you will be required to supply them during the installation steps:
  - The Database name.
  - The Database hostname.
  - The user name and password that has full administrative privileges to the backing key store database.

Procedure

1. In a supported web browser on the cluster in which the Ranger service is installed, log in to Cloudera Manager as a user with full administrative privileges.
2. From the Cloudera Manager navigation side-bar, select Administration Security .
3. On the Security Status page, click Set up HDFS Data At Rest Encryption.
4. In the Set up HDFS Data At Rest Encryption page, select the Ranger Key Management Service backed by Database option.

   A list of tasks are displayed at the bottom of the page. To successfully set up HDFS Data at Rest encryption, these tasks must be completed.

   Important: Kerberos and TLS must be enabled. If the steps associated with these tasks do not display Completed in the Status column, before continuing, click the link associated with the uncompleted task and follow the Wizard's instructions.
5. To set up HDFS Encryption, follow the instructions as described below for each of the Set up HDFS Data At Rest Encryption Wizard's steps.

Related Information

Configuring a database for Ranger or Ranger KMS

TLS/SSL and Its Use of Certificates

Enabling Kerberos Authentication for Cloudera

Installing the Ranger KMS Service

The Set up HDFS Data At Rest Encryption wizard's installation step that installs the Ranger Key Management System (KMS) with High Availability (HA) service on your cluster.

About this task

Describes the steps that install the Ranger Key Management System (KMS) service on your cluster and associates it with your backing key store database.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Add Ranger KMS Service.

   The Add Ranger KMS Service to Cluster Wizard opens.
2. In the Assign Roles page, verify that the hostname is the required server on which to install the Ranger KMS service by clicking inside the listed server field. By default, this field is populated by the Wizard.

   The Hosts Selected page opens.

3. In the Hosts Selected page, scroll down and from the Hostname column, locate the hostname that was selected by the Wizard. Notice in the Added Roles column the Ranger KMS Server (RK) role icon. This role is added during the installation.
4. Do one of the following:
   - If the pre-selected host is correct, confirm the Wizard's choice by clicking OK.
   - If the pre-selected host is incorrect, deselect the check box of the Wizard's choice, select the hostname check box of the required server, and then click OK.

     Note: If you require more than one KMS service, select the hostname check box for each server on which to install a Ranger KMS service.
5. Back in the Assign Roles page, click Continue.

   The Setup Database page opens.
6. In the Setup Database page, do the following:
   a. In the Database Hostname field, enter the hostname of the backing key store database.
   b. In the Database Name field, enter the name of the backing key store database.
   c. In the Username field, enter the user name that has full administrative privileges to the backing key store database.
   d. In the Password field, enter the password of the user name that has full administrative privileges to the backing key store database.
   e. (Optional) Verify that the credentials you entered are correct by clicking Test Connection.
   f. Click Continue.

   The Review Changes page opens.
7. In the required Ranger KMS Master Key Password field, enter the password that will be used to encrypt the master key.

   Tip: You can confirm the password's value by clicking the ranger_kms_master_key_password link.
8. Review the rest of the settings before clicking Continue.
9. In the Command Details page, monitor the installation of the Ranger KMS server. When the Status displays Finished the Ranger KMS is installed and tested.

   Tip: If the Ranger KMS start task fails during the First Run Command process, click the Context Ranger KMS link, which opens the Ranger KMS service page. From the Actions list, select Start.
10. Click Continue.

    The Summary page opens.
11. Click Finish, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.
12. Verify that the Ranger KMS service appears in the Cloudera Manager Clusters components list and that the service has been started.

    If the Ranger KMS service was not started by the installation wizard, do the following:
    a. Go to Cloudera Manager's Home page by clicking the Cloudera Manager icon.
    b. In the Cloudera Manager Clusters components list, locate and click Ranger KMS.
    c. From the Actions menu, click Start.

What to do next

Adding Ranger KMS to a cluster triggers additional property updates for other services. Cloudera Manager may flag these with stale configuration warnings. Restart the stale services and redeploy the client configuration.

Restarting the Stale Services and Redeploying the Client Configuration

The Set up HDFS Data At Rest Encryption wizard's step for restarting stale services and redeploying the client configuration.

About this task

Describes the steps that restart stale services after installing the Data-at-Rest HDFS Ranger KMS service option on your cluster.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Restart stale services and redeploy client configuration..

   The Stale Configurations page opens.
2. Click Restart Stale Services.

   The restart Stale Services page opens.
3. Verify that the Re-deploy client configuration check box is selected and click Restart Now.
4. In the Command Details page, monitor the restart process. When the Status displays Finished, click Continue, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.

What to do next

Validate that the Data-at-Rest HDFS Ranger KMS service option can successfully encrypt your data to-and-from HDFS.

Validating Data Encryption to-and-from HDFS

The Set up HDFS Data At Rest Encryption wizard's step for validating the data encryption to-and-from HDFS.

About this task

Describes the steps which verify that the Data-at-Rest HDFS Ranger KMS service option can successfully encrypt your data to-and-from HDFS.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Validate Data Encryption.

   The Validate Data Encryption page opens, which displays a list of commands and instructions for creating an encryption zone and adding data.
2. In a terminal, log in to one of the hosts in your cluster and run each of the following commands:
   a) Create a key and directory by entering the following:

   ```
   kinit KEY_ADMIN_USER
   hadoop key create mykey1
   hdfs dfs -mkdir /tmp/zonel
   ```

   Where, *KEY_ADMIN_USER* is the key administrator whose role can perform the following actions:

   - Configure HDFS encryption and manage encryption keys
   - Start, stop, and restart Ranger KMS
   - Configure Ranger KMS Policies
   - View configuration and monitoring information in Cloudera Manager
   - View service and monitoring information
   - View events and logs
   - View Replication jobs and snapshot policies
   - View YARN applications and Impala queries

   b) Create a zone and link to the key, by entering the following:

   ```
   kinit hdfs hdfs
   hdfs crypto -createZone -keyName mykey1 -path /tmp/zone1
   ```

c) Create a file, put it in your zone, and verify that the file can be decrypted, by entering the following:

```
kinit KEY ADMIN_USER
echo "Hello World" > /tmp/helloWorld.txt
hdfs dfs -put /tmp/helloWorld.txt /tmp/zone1
hdfs dfs -cat /tmp/zonel/helloWorld.txt
rm /tmp/helloWorld.txt
```

d) Verify that the stored file is encrypted, by entering the following:

```
kinit hdfs
hdfs dfs -cat /.reserved/raw/tmp/zonel/helloWorld.txt
hdfs dfs -rm -R /tmp/zone1
hdfs crypto -listZones
```

3. When completed, click Close, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.

Post-tasks for the Data-at-Rest HDFS Ranger KMS service

The post-tasks that you must perform after you have set up the Data-at-Rest HDFS Ranger KMS service option.

About this task

Describes the post-task steps.

Depending on which Data-at-Rest HDFS Ranger KMS service option was set up, two or more of the following post-tasks must be completed:

- Update the Data-at-Rest HDFS Ranger KMS service's URL
- Create a Ranger Audit Directory
- (Ranger KMS service only) Update the Authentication Properties and KMS Hadoop cache settings

Procedure

1. Update the Data-at-Rest HDFS Ranger KMS service's URL by doing the following:
   a. In the Cloudera Manager Clusters components list, locate and click the Ranger service.
   b. Log in to the Ranger Web UI as the Ranger KMS user, whose default user name credential is keyadmin and default password is admin123.
   c. In the cm_kms service, click the Edit icon and update the KMS URL field value as follows:
      1. In the KMS URL field, enter the URL value using the following syntax:

         kms://http@*KMS_HOST1*;*KMS_HOST2*:*KMS_PORT*/kms

         Where,

         - *KMS_HOST* is the host where either the Ranger KMS or the Ranger KMS backed by a database is installed.
         - *KMS_PORT* is the port number. By default, this is 9292. For example,

           kms://http@*KMS_HOST1*;*KMS_HOST2*:9292/kms

           Important: If SSL is enabled, use https and port number 9494. For example:

           kms://https@*KMS_HOST1*;*KMS_HOST2*:9494/kms
      2. To confirm your URL setting, click Test Connection.
      3. Click Save.
2. Create a Ranger Audit Directory by doing the following:
   a. Depending on which Data-at-Rest HDFS Ranger KMS service you set up, in the Cloudera Manager Clusters components list, locate and click the Ranger KMS service.
   b. From the Actions menu, click Create Ranger Plugin Audit Directory.

c. When the Create Ranger Plugin Audit Directory message appears, confirm its creation by clicking Create Ranger Plugin Audit Directory.
d. Monitor the creation process. When the Status displays Finished, click Close.

Installing Ranger KMS in a federated deployment

Install Ranger Key Management System (KMS) in a federated deployment to isolate encryption key lifecycles within a dedicated security cluster separated from the data cluster.

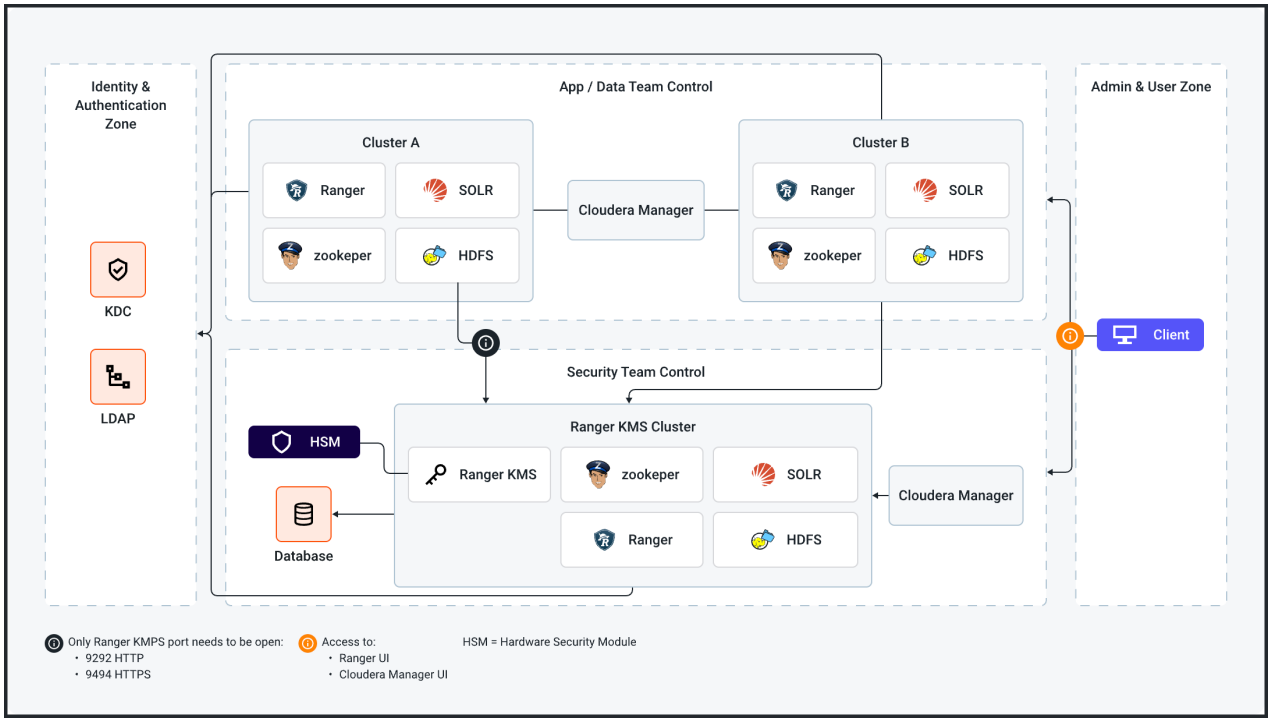
About this task

Application and data processing occur in a data cluster. The data cluster stores and processes actual datasets but does not directly manage encryption keys. Instead, a separately managed security cluster with the Ranger KMS service installed handles the key lifecycle operations, for example, generation, rotation, and storage. This separation of tasks enhances security by isolating the administration of the data cluster from the security cluster. Additionally, this architecture provides the following benefits:

- Improved scaling, which enables you to migrate to a shared cluster of horizontally scaled Ranger KMS instances.
- Isolation of service handling keys, which ensures that the keys are accessible only by the security team.

Ranger KMS can be deployed in a federated cluster for key management. The following diagram shows the architecture of a federated deployment:

Figure 1: Federated deployment architecture



At least the following services are required for data clusters:

- Ranger
- Solr
- ZooKeeper
- HDFS

At least the following services are required for the federated Ranger KMS cluster:

- Ranger KMS
- Ranger

- Solr
- ZooKeeper
- HDFS – HDFS is required in the security cluster to capture audit activities related to Ranger KMS.

Before you begin

- Verify that the following requirements are met:
  - The cluster in which Cloudera Manager and the Ranger service are installed must be running.
  - Kerberos must be enabled in your cluster.

    Note: If you did not enable Kerberos, click the link associated with the uncompleted task and follow the instructions in the wizard.
  - TLS/SSL must be enabled in your cluster.

    Note: If you did not enable TLS, click the link associated with the uncompleted task and follow the instructions in the wizard.
  - A Ranger KMS database must have been created as the underlying keyStore mechanism. This database must be separate from the Ranger database.
  - You must have securely recorded the following backing key store database access credentials, as you will need them during the installation steps:
    - The database name.
    - The database hostname.
    - The user name and password that have full administrative privileges to the backing key store database.
- Data clusters and the security cluster must be in the same Kerberos realm.

  Note: If they are in different realms, you need to set up a one-way cross-realm trust, however, the provided steps only apply to the single-realm scenarios.
- Data clusters and the security cluster must use the same Ranger Usersync configuration for syncing users. Ensure that the AD/LDAP-related configuration properties have the same values set in both the data and security clusters. For more information, refer to *Ranger Usersync*.
- Data clusters and the security cluster must have the same users at the Unix level.
- You must configure HA for Ranger KMS. For more information, refer to *Configure High Availability for Ranger KMS with DB*.

Procedure

1. Add a Ranger KMS service.

   For instructions, refer to *Installing the Ranger KMS service*.
2. Restart the stale services and redeploy the client configuration.

   For instructions, refer to *Restarting the stale services and redeploying the client configuration*.
3. Configure data clusters.

   Data clusters must have access to Ranger KMS running on the security cluster. To do so, perform the following steps:

   a) Go to Cloudera Manager Core Settings Configuration Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml .

   b) Add the following property:
      - Name: hadoop.security.key.provider.path
      - Value: kms://http@kms_host1;kms_host2:kms_port/kms

   c) Go to Cloudera Manager HDFS Configuration HDFS Service Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml .

   d) Add the following property:

- Name: hadoop.security.key.provider.path
- Value: kms://http@kms_host1;kms_host2:kms_port/kms

Note: The kms_port is 9494. If SSL is not enabled in your cluster, the kms_port is 9292.

4. Validate the data encryption.

   For instructions, refer to *Validating data encryption to-and-from HDFS*.

Related Information

Ranger Usersync
Configure High Availability for Ranger KMS with DB
Installing the Ranger KMS service
Restarting the stale services and redeploying the client configuration
Validating data encryption to-and-from HDFS

Installing Cloudera Navigator Encrypt

Learn about installing Navigator Encrypt, setting up TLS certificates on a Navigator Encrypt client, entropy requirements , and uninstalling and reinstalling NavEncrypt.

Before you begin

See Data at Rest Encryption Requirements for more information about encryption and Navigator Encrypt requirements.

About this task

Important: Before installing Cloudera Navigator Encrypt, see Encrypting Data at Rest and the Product Compatibility Matrix for Cloudera Navigator Encryption for important considerations.

Setting Up an Internal Repository

You must create an internal repository to install or upgrade Navigator Encrypt. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see Configuring a Local Package Repository.

Installing Navigator Encrypt (RHEL-Compatible)

Learn how to install RHEL compatible Navigator Encrypt. The steps below show an example of how to install NavEncrypt on a cluster running Red Hat Linux.

About this task

Note: For details about supported Linux Operating Systems, refer to the Product Compatibility Matrix for Cloudera Navigator Encryption.

Procedure

1. Install the EPEL Repository.

   Dependent packages are available through the Extra Packages for Enterprise Linux (EPEL) repository. To install the EPEL repository, install the epel-release package. The EPEL repository for each release of RHEL is different ,so confirm the host is set up correctly.

2. Install the NavEncrypt Repository.

   a) mkdir -p /root/navencrypt-repo

b) Fetch the NavEncrypt repository from the Cloudera download site, for example: wget https://archive.cloudera.com/p/navencrypt7/7.1.9.1000/rhel8/navigator-encrypt-7.1.9.1000-el8.tar.gz
c) tar -zxvf navigator-encrypt-7.1.9.1000-el8.tar.gz --directory /root/navencrypt-repo

3. Create and edit file /etc/yum.repos.d/navencrypt-repo:

   [navencrypt-repo] name=Cloudera NavEncrypt enabled=1 autorefresh=0 gpgcheck=1 baseurl=file:///root/navencrypt-repo gpgkey=file:///root/navencrypt-repo/nepub.asc

4. Install Kernel Libraries.

   For Navigator Encrypt to run as a kernel module, you must download and install the kernel development headers. Each kernel module is compiled specifically for the underlying kernel version. Running as a kernel module allows Navigator Encrypt to provide high performance and complete transparency to user-space applications.

   To determine your current kernel version, run uname -r.

   To install the development headers for your current kernel version, run:

   ```
   sudo yum install kernel-headers-$(uname -r) kernel-devel-$(uname -r)
   ```

5. (RHEL or CentOS Only) Manually Install dkms.

   With some versions of RHEL and CentOS, because of a broken dependency, you must manually install the dkms package. To do this ,you must locate a repo that has a version of dkms that is compatible with the version of RHEL the host is running.

   ```
   sudo yum install https://download-ib01.fedoraproject.org/pub/epel/7/aarch64/Packages/d/dkms-2.7.1-1.el7.noarch.rpm
   ```

6. yum install libkeytrustee
7. yum install navencrypt-kernel-module

   Note: navencrypt-kernel-module depends on package dkms. Depending on the versions of the packages yum might update the linux kernel. Check for this and if it has occurred, reboot the host.

8. Install Navigator Encrypt.

   Install the Navigator Encrypt client using the yum package manager:

   ```
   sudo yum install navencrypt
   ```

   If you attempt to install navencrypt-kernel-module with incorrect or missing kernel headers, you see a message like the following:

   ```
   Building navencryptfs 3.8.0 DKMS kernel module...

   ###################### BUILDING ERROR ######################


   Creating symlink /var/lib/dkms/navencryptfs/3.8.0/source ->
                    /usr/src/navencryptfs-3.8.0
   DKMS: add completed.
   Error! echo
   Your kernel headers for kernel 3.10.0-229.4.2.el7.x86_64 cannot be found a
   t
   /lib/modules/3.10.0-229.4.2.el7.x86_64/build or /lib/modules/3.10.0-229.4.
   2.el7.x86_64/source.

   ###################### BUILDING ERROR ######################

   Failed installation of navencryptfs 3.8.0 DKMS kernel module !
   ```

   To recover, see Navigator Encrypt Kernel Module Setup.

9. Confirm NavEncrypt is installed.

   yum list installed | egrep "naven|keytrust"

Installing Navigator Encrypt (SLES-12)

Learn how to install SLES 12 compatible Navigator Encrypt . The steps below show an example of installing SLES 12 compatible NavEncrypt, assuming the user is root.

Procedure

1. Install the NavEncrypt Repository.
   a) mkdir -p /root/navencrypt-repo
   b) Fetch the NavEncrypt repository from the Cloudera download site, for example: wget https://archive.cloudera.com/p/navencrypt7/7.1.9.1000/sles15/navigator-encrypt-7.1.9.1000_sles15.4-0.tar.gz
   c) tar -zxvf navigator-encrypt-7.1.9.1000_sles15.4-0.tar.gz --directory /root/navencrypt-repo
2. Create and edit file /etc/zypp/repos.d/navencrypt.repo:

   [navencrypt-repo] name=Cloudera NavEncrypt enabled=1 autorefresh=0 gpgcheck=1 baseurl=file:///root/navencrypt-repo gpgkey=file:///root/navencrypt-repo/nepub.asc
3. Confirm zypper can access repo.

   ```
   zypper search -r navencrypt-repo
   ```
4. zypper install libkeytrustee
5. Install the Kernel Module Package and Navigator Encrypt Client.

   Install the kernel module package (KMP) and Navigator Encrypt client with zypper:

   ```
   sudo zypper install cloudera-navencryptfs-kmp-default
   sudo zypper install navencrypt
   ```
6. Confirm NavEncrypt is installed

   zypper search -i | egrep "naven|keytrust"
7. systemctl daemon-reload

Installing Navigator Encrypt (SLES-15)

Learn how to install SLES 15 compatible Navigator Encrypt. The following steps show an example of installing SLES 15 compatible Navigator Encrypt, assuming the user is root.

Procedure

1. Install the Navigator Encrypt repository.
   a) Run the following command:

      ```
      mkdir -p /root/navencrypt-repo
      ```
   b) Fetch the Navigator Encrypt repository from the Cloudera download site.

      For example,

      ```
      wget https://archive.cloudera.com/p/navencrypt7/7.1.9.1000/sles15/navigator-encrypt-7.1.9.1000-49-sles15.tar.gz
      ```
   c) Execute the following command:

      ```
      tar -zxvf navigator-encrypt-7.1.9.1000-49-sles15.tar.gz --directory /root/navencrypt-repo
      ```

2. Create and edit the /etc/zypp/repos.d/navencrypt.repo file:

```
[navencrypt-repo] name=Cloudera NavEncrypt enabled=1 autorefresh=0 gpgch
eck=1 baseurl=file:///root/navencrypt-repo gpgkey=file:///root/navencrypt-
repo/nepub.asc
```

3. Confirm that Zypper can access the repository.

```
zypper se -v  navencrypt libkeytrustee
```

```
       S  | Name                                | Type       | Version
                | Arch   | Repository
       ---+-------------------------------------+------------+----------
------------------+--------+--------------------
          | cloudera-navencryptfs              | srcpackage | 7.1.9-0.0
                | noarch | Cloudera NavEncrypt
           name: cloudera-navencryptfs
       i+ | cloudera-navencryptfs-kmp-default | package    | 7.1.9_k5.
14.21_150400.22-0.0 | x86_64 | Cloudera NavEncrypt
           name: cloudera-navencryptfs-kmp-default
        v  | cloudera-navencryptfs-kmp-default | package    | 7.1.9_k5.1
4.21_150500.53-0.0 | x86_64 | Cloudera NavEncrypt
           name: cloudera-navencryptfs-kmp-default
       i+ | libkeytrustee                      | package    | 7.1.9.1000_s
les15.4-49       | x86_64 | Cloudera NavEncrypt
           name: libkeytrustee
       v  | libkeytrustee                      | package    | 7.1.9.1000
_sles15.5-49       | x86_64 | Cloudera NavEncrypt
           name: libkeytrustee
       i+ | navencrypt                         | package    | 7.1.9.1000_s
les15.4-49       | x86_64 | Cloudera NavEncrypt
           name: navencrypt
       v  | navencrypt                         | package    | 7.1.9.100
0_sles15.5-49       | x86_64 | Cloudera NavEncrypt
           name: navencrypt
```

You can see that the repository contains packages for SLES15-SP4 and SLES15-SP5.

4. If you are running SLES15-SP4, install:

- zypper in libkeytrustee-7.1.9.1000_sles15.4-49
- zypper in cloudera-navencryptfs-kmp-default-7.1.9_k5.14.21_150400.22-0.0
- zypper in navencrypt-7.1.9.1000_sles15.4-49

5. Confirm that Navigator Encrypt SLES15.4 is installed:

```
zypper se -v -i navencrypt libkeytrustee
```

```
S  | Name                               | Type    | Version
     | Arch   | Repository
---+------------------------------------+---------+-----------------------
------+--------+--------------------
i+ | cloudera-navencryptfs-kmp-default | package | 7.1.9_k5.14.21_15040
0.22-0.0 | x86_64 | Cloudera NavEncrypt
   name: cloudera-navencryptfs-kmp-default
i+ | libkeytrustee                      | package | 7.1.9.1000_sles15.4-49
     | x86_64 | Cloudera NavEncrypt
   name: libkeytrustee
i+ | navencrypt                         | package | 7.1.9.1000_sles15.4-49
     | x86_64 | Cloudera NavEncrypt
   name: navencrypt
```

6. If you are running SLES15-SP5, install:
   - zypper in libkeytrustee-7.1.9.1000_sles15.5-49
   - zypper in cloudera-navencryptfs-kmp-default-7.1.9_k5.14.21_150500.53-0.0
   - zypper in navencrypt-7.1.9.1000_sles15.5-49
7. Confirm that Navigator Encrypt SLES15.5 is installed:

```
zypper se -v -i navencrypt libkeytrustee
```

```
S  | Name                                | Type    | Version
     | Arch   | Repository
---+-------------------------------------+---------+-----------------------
------+--------+--------------------
i+ | cloudera-navencryptfs-kmp-default | package | 7.1.9_k5.14.21_15050
0.53-0.0 | x86_64 | Cloudera NavEncrypt
    name: cloudera-navencryptfs-kmp-default
i+ | libkeytrustee                       | package | 7.1.9.1000_sles15.5-49
     | x86_64 | Cloudera NavEncrypt
    name: libkeytrustee
i+ | navencrypt                          | package | 7.1.9.1000_sles15.5-49
     | x86_64 | Cloudera NavEncrypt
    name: navencrypt
```

8. Reload the systemd files:

```
systemctl daemon-reload
```

Installing Navigator Encrypt (Ubuntu)

Learn how to install Ubuntu compatible Navigator Encrypt . The steps below show an example of installing Ubuntu compatible NavEncrypt, assuming the user is root.

Procedure

1. Install the NavEncrypt Repository.
   a) mkdir -p /root/navencrypt-repo
   b) Fetch the NavEncrypt repository from the Cloudera download site, for example: wget https://archive.cloudera.com/p/navencrypt7/7.1.9.1000/ubuntu/navigator-encrypt-7.1.9.1000-61-ubuntu20.tar.gz
   c) tar -zxvf navigator-encrypt-7.1.9.1000-61-ubuntu20.tar.gz --directory /root/navencrypt-repo
   d) apt-key add /root/navencrypt-repo/nepub.asc
2. Install Kernel Headers.

   Determine your kernel version by running uname -r, and install the appropriate headers:

```
sudo apt-get install linux-headers-$(uname -r)
```

3. apt-get install libkeytrustee4
4. apt-get install navencrypt-kernel-module
5. Install the Navigator Encrypt Client.

   Install Navigator Encrypt:

```
sudo apt-get install navencrypt
```

6. Confirm if NavEncrypt is installed.

```
apt-cache search . | egrep "naven|keytrust"
```

Installing for Ranger KMS

If you are using Ranger KMS as your Key Management server you need to do the following:

Procedure

1. Generate a valid Kerberos ticket.

   There is a utility called navencrypt-gen-keytab that works with Cloudera Manager to create a valid Kerberos ticket.
2. Start the navencrypt-krb5 service after the keytab file is generated.

Setting Up TLS for Navigator Encrypt Clients

Transport Layer Security (TLS) certificates are used to secure communication with Navigator Encrypt. Cloudera recommends using certificates signed by a trusted Certificate Authority (CA).

About this task

If the TLS certificate is signed by an unrecognized CA, such as an internal CA, then you must add the root certificate to the host certificate truststore of each Navigator Encrypt client. Be aware that Navigator Encrypt uses the operating system's truststore, which is distinct from the JDK truststore used by Cloudera Manager.

To set up TLS certificates on a Navigator Encrypt client:

Procedure

1. If not already installed, install the CA-certificates:

   ```
   yum install ca-certificates
   ```
2. Enable the dynamic CA configuration feature:

   ```
   update-ca-trust enable
   ```
3. Copy the root certificate into the host certificate truststore:

   ```
   cp /path/to/root.pem /etc/pki/ca-trust/source/anchors/
   ```
4. Update the host certificate truststore:

   ```
   update-ca-trust
   ```

Example

Example:

```
[root@navencrypt-1 ~]# systemctl stop navencrypt-mount
Stopping navencrypt directories
 * Umounting /dev/nvtest/test1 ...                          [  OK  ]
 * Umounting /dev/nvtest/test2 ...                          [  OK  ]
 * Unloading module ...                                     [  OK  ]

[root@navencrypt-1 ~]# update-ca-trust enable
[root@navencrypt-1 ~]# cp dd-1.lab.usa.company.com.pem /etc/pki/ca-trust/s
ource/anchors/
[root@navencrypt-1 ~]# update-ca-trust

[root@navencrypt-1 ~]# systemctl stop navencrypt-mount
Starting navencrypt directories
 * Mounting '/dev/nvtest/test1'                             [  OK  ]
 * Mounting '/dev/nvtest/test2'
```

Entropy Requirements

Many cryptographic operations, such as those used with TLS or HDFS encryption, require a sufficient level of system entropy to ensure randomness; likewise, Navigator Encrypt needs a source of random numbers to ensure good performance.

About this task

Hence, you need to ensure that the hosts running Navigator Encrypt have sufficient entropy to perform cryptographic operations.

You can check the available entropy on a Linux system by running the following command:

```
cat /proc/sys/kernel/random/entropy_avail
```

The output displays the entropy currently available. Check the entropy several times to determine the state of the entropy pool on the system. On hosts running a Linux kernel version less than 5.10.119, if the entropy is consistently low (500 or less), you must increase it by installing rng-tools version 4 or higher, and starting the rngd service. On hosts running a Linux kernel version of 5.10.119 or higher the entropy version will be stable at 256, unless there are special entropy requirements in place, no further action is required.

Install rng_tools Using Package Manager

Learn how to install rng_tools using Package Manager.

About this task

If version 4 or higher of the rng-tools package is available from the local package manager (yum), then install it directly from the package manager. If the appropriate version of rng-tools is unavailable, see Building rng-tools From Source on page 195.

For RHEL, run the following commands:

```
sudo yum install rng-tools
cp /usr/lib/systemd/system/rngd.service /etc/systemd/system/
systemctl daemon-reload
systemctl start rngd
systemctl enable rngd
```

Building rng-tools From Source

If you are unable to install rng-tools using package manager, you can build from source.

About this task



Note: If your package manager only offers an older version (3.x or earlier), then you must build from source.

To install and start rngd and build from source:

1. Download the source code:

   ```
   sudo wget http://downloads.sourceforge.net/project/gkernel/rng-tools/4/r
   ng-tools-4.tar.gz
   ```

2. Extract the source code:

   ```
   tar xvfz rng-tools-4.tar.gz
   ```

3. Enter the rng-tools-4 directory:

   ```
   cd rng-tools-4
   ```

4. Run ./configure
5. Run make
6. Run make install

After you have installed rng-tools, start the rngd daemon by running the following command as root:

```
sudo rngd --no-tpm=1 -o /dev/random
```

For improved performance, Cloudera recommends configuring Navigator Encrypt to read directly from /dev/random instead of /dev/urandom.

To configure Navigator Encrypt to use /dev/random as an entropy source, add --use-random to the navencrypt-prepare command when you are setting up Navigator Encrypt.

Uninstalling and Reinstalling Navigator Encrypt

Learn how to uninstall and reinstall Navigator Encrypt.

About this task

Uninstalling Navigator Encrypt

For RHEL-compatible OSes:

```
sudo yum remove navencrypt
sudo yum remove navencrypt-kernel-module
```

These commands remove the software itself. On RHEL-compatible OSes, the /etc/navencrypt directory is not removed as part of the uninstallation. Remove it manually if required.

Reinstalling Navigator Encrypt

After uninstalling Navigator Encrypt, repeat the preceding installation instructions for your distribution.

When Navigator Encrypt is uninstalled, the configuration files and directories located in /etc/navencrypt are not removed. Consequently, you do not need to use the navencrypt register command during reinstallation. If you no longer require the previous installation configuration information in the directory /etc/navencrypt, you can remove its contents.

Installing Ranger RMS

Ranger Resource Mapping Server (RMS) enables automatic translation of access policies from Hive to HDFS.

About this task



Note: Ranger RMS is an optional service and cannot be installed through the Add Service express wizard during the initial installation of services. To install Ranger RMS, you must first have a Cloudera Base on premises cluster with Ranger, HDFS, and Hive services already installed.

Legacy CDH users used Hive policies in Apache Sentry that automatically linked Hive permissions with HDFS ACLs. This was especially convenient for external table data used by Spark or Hive.

Previously, Ranger only supported managing Hive and HDFS policies separately. Ranger RMS (Resource Mapping Server) allows you to authorize access to HDFS directories and files using policies defined for Hive tables. RMS is the service that enables Hive-HDFS ACL Sync.



Important: Cloudera does not support the MySQL server with Global Transaction Identifier (GTID) support enabled as a Ranger RMS database.

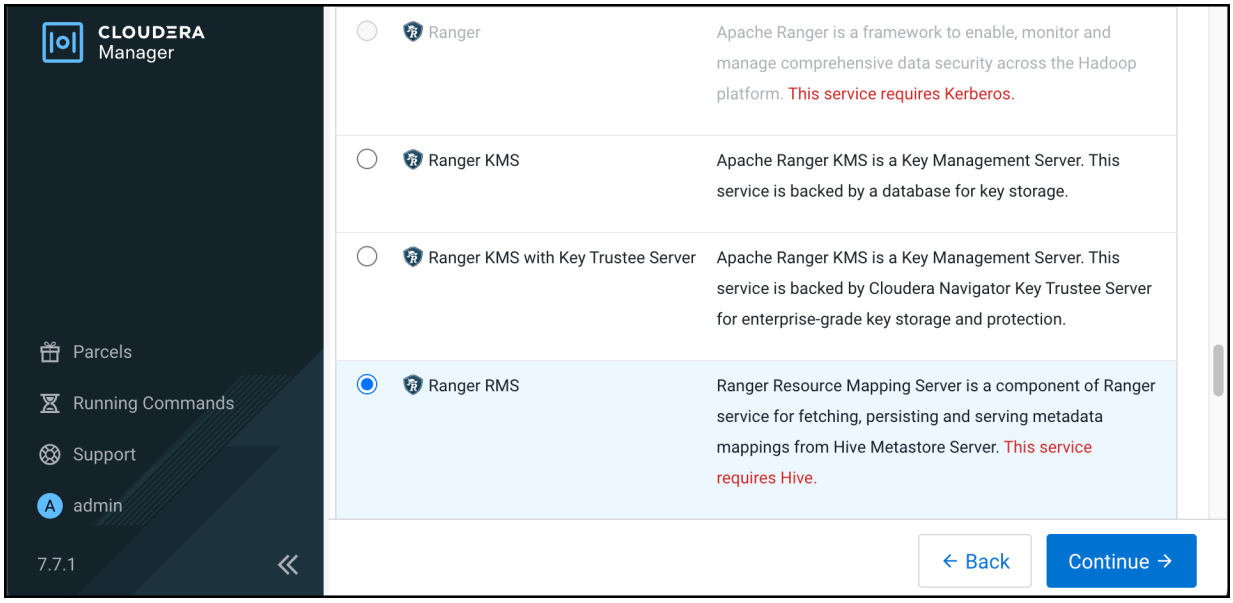
Before you begin

You must have installed:

- Cloudera Base on premises 7.1.4 or higher version cluster with Apache Ranger, Hive, and HDFS.
- Ranger RMS on the host where Hive_Gateway is available.

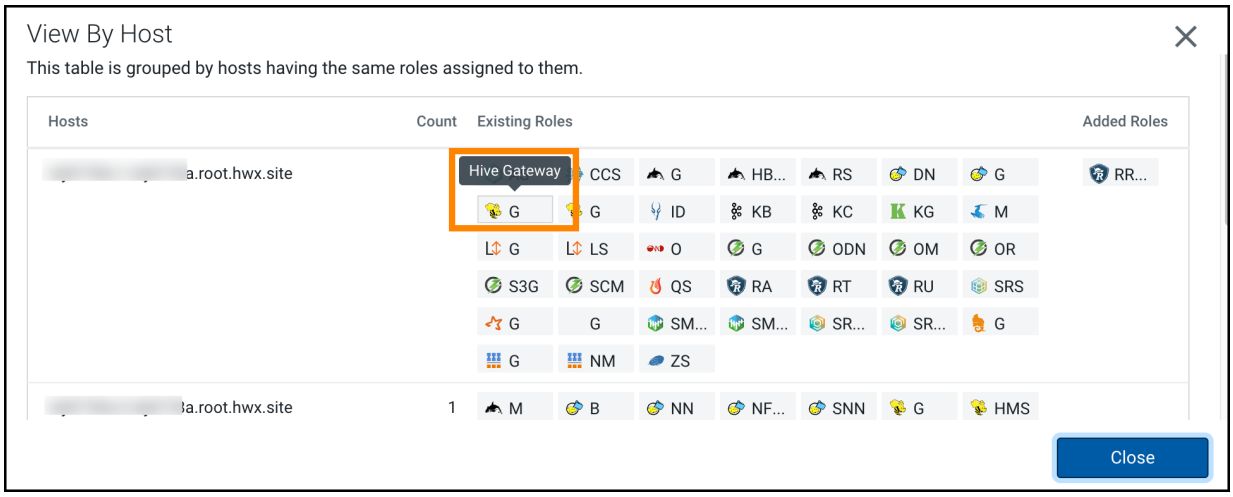
Procedure

1. On the cluster home page, click , then click Add Service.
2. Select Ranger RMS, then click Continue.

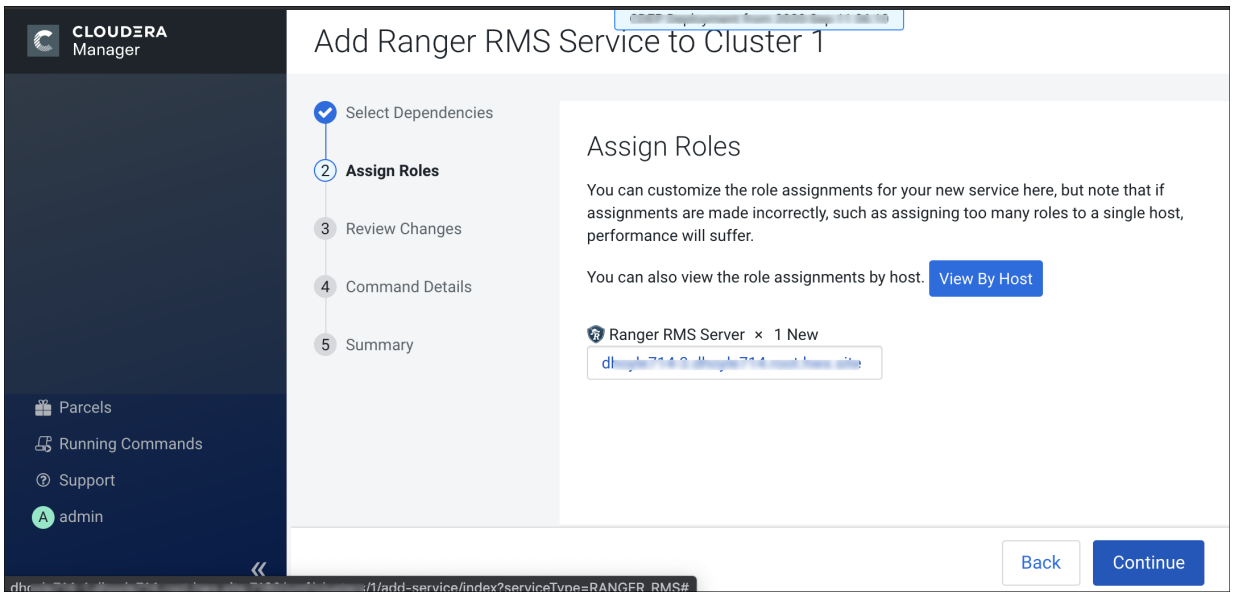


3. On Assign Roles, click View by Host.
4. On View by Host, verfiy that the host on which you install Ranger RMS has the required Hive Gateway role assigned, then click Close.

Figure 2: Verifying Hive Gateway role on a host



5. On Assign Roles, click Continue.



6. On **Review Changes**,

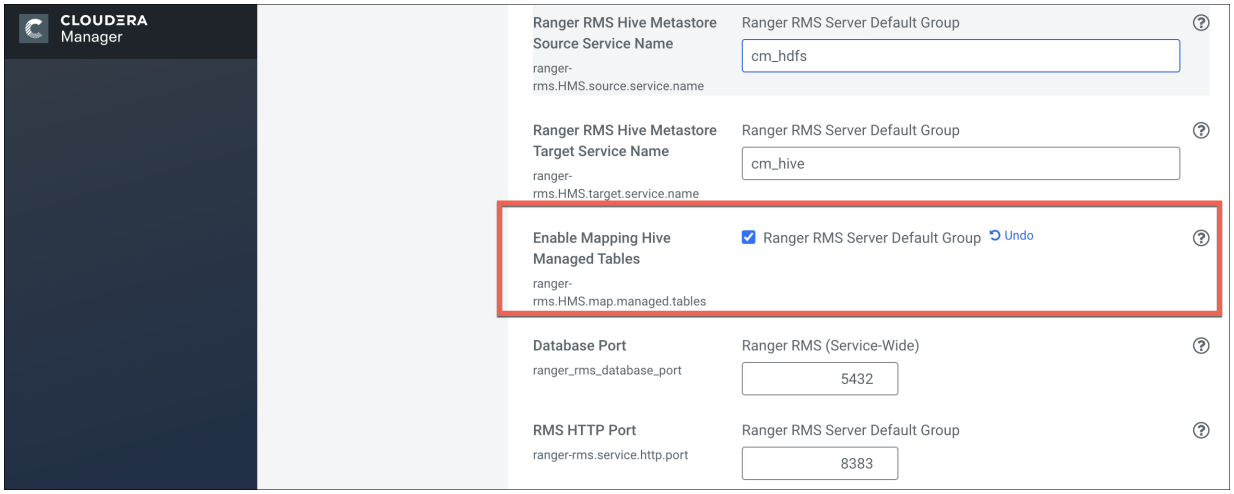
Note:

Ranger RMS uses the same database settings that are used in Ranger.

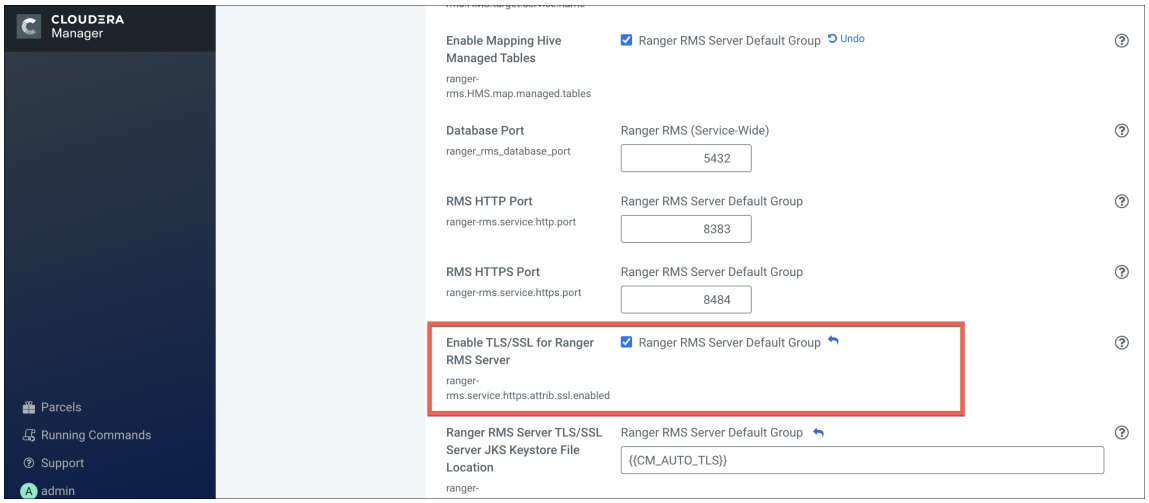
Ranger RMS should be installed in the same database instance as Ranger.

If Ranger service is using SSL-enabled database, make sure to have the database certificate / keystore / truststore file used by Ranger to connect to the SSL-enabled database present on the node where Ranger RMS is installed.

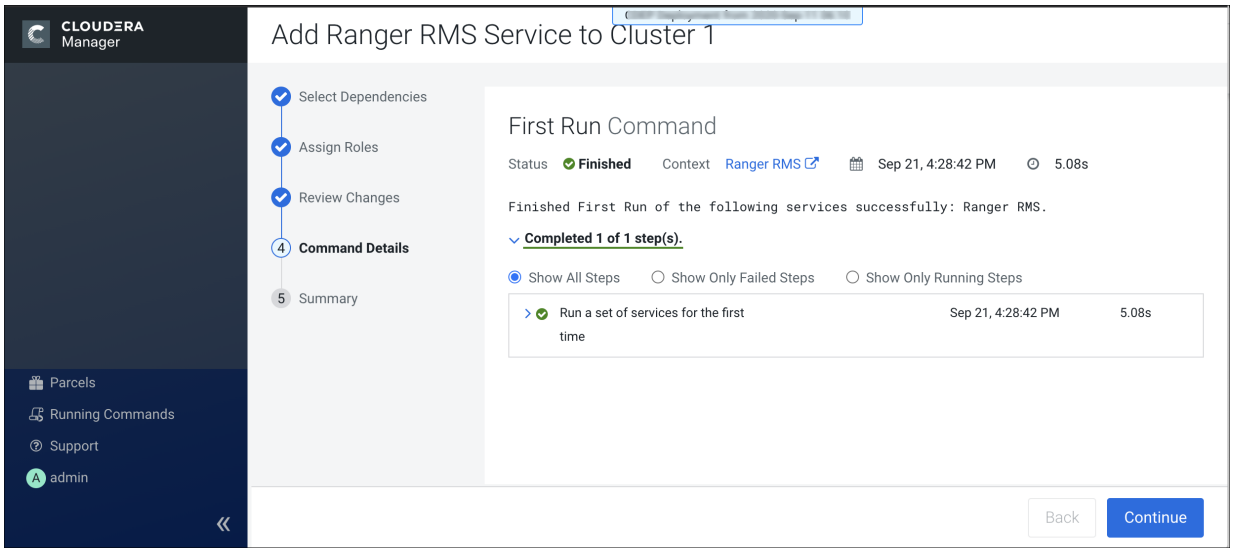
To track managed tables, select the Enable Mapping Hive Managed Tables option.



Note: If you are adding Ranger RMS in a cluster with SSL enabled, the Enable TLS/SSL for Ranger RMS Server option should be selected by default.

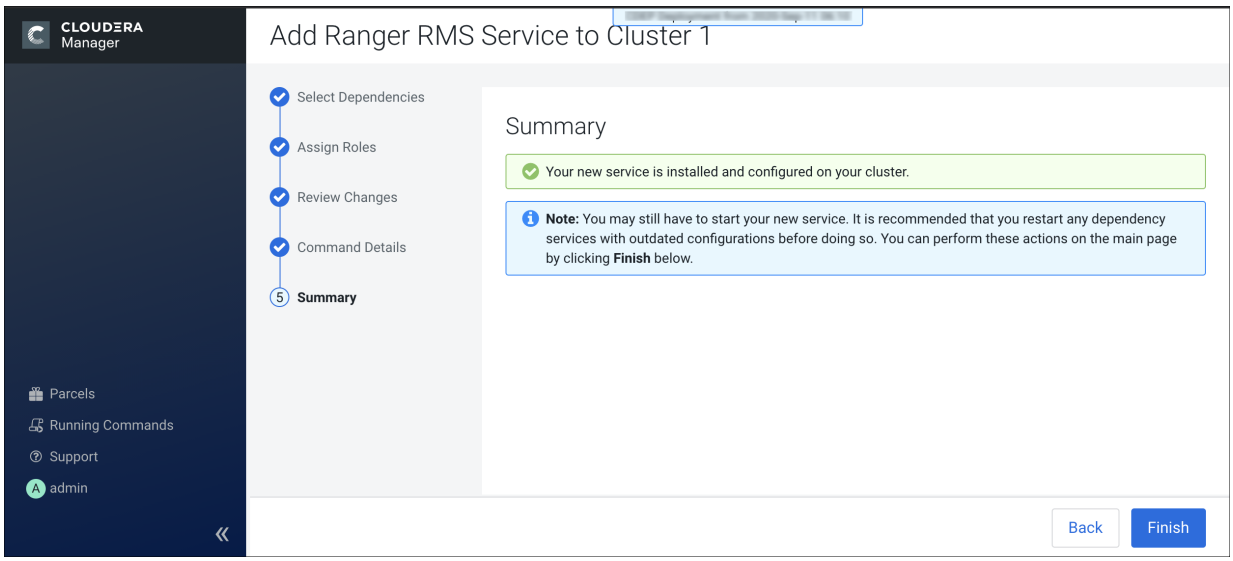


7. On the **Command Details** page, select run options, then click Continue.



8. On the **Summary** page, click Finish.

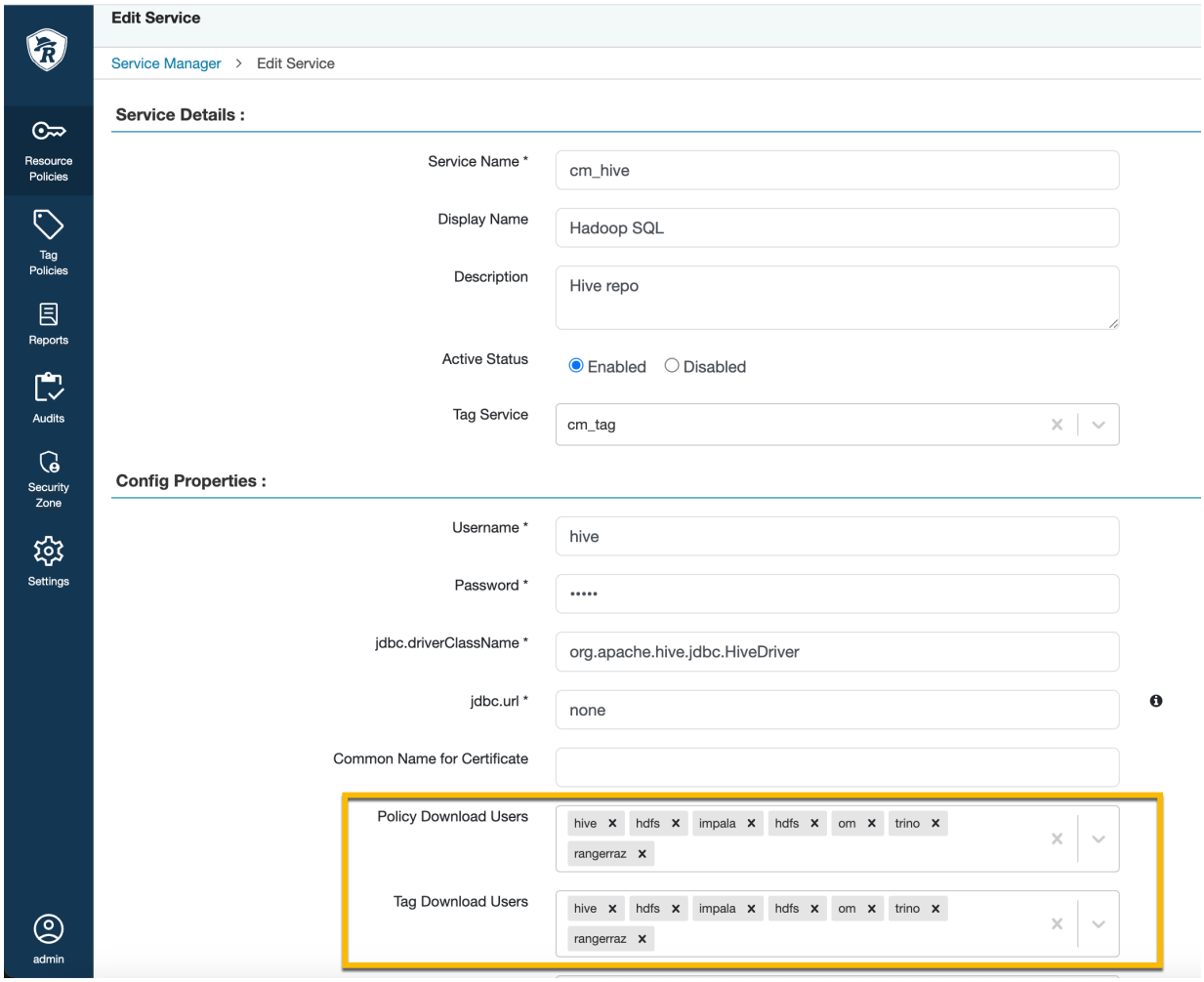
Important: Do not start the Ranger RMS service before completing the following additional configuration steps. Ranger RMS restarts when you restart stale services after completing configuration changes.



9. In Cloudera Manager Hive Service Configuration verify that the Hive Metastore Access Control and Ranger RMS Proxy User Hosts property, hadoop.proxyuser.rangerrms.hosts is set to *.

   Note: rangerrms user is given superuser privilege only for the HiveMetaStore service, so rangerrms can access metadata information without an explicit Ranger policy allowing it necessary permissions. However, Hive operations such as drop database must be authorized in the hive-server2 by Ranger policies. You must create an appropriate Ranger policy which grants the user executing this command the required permission to do so.

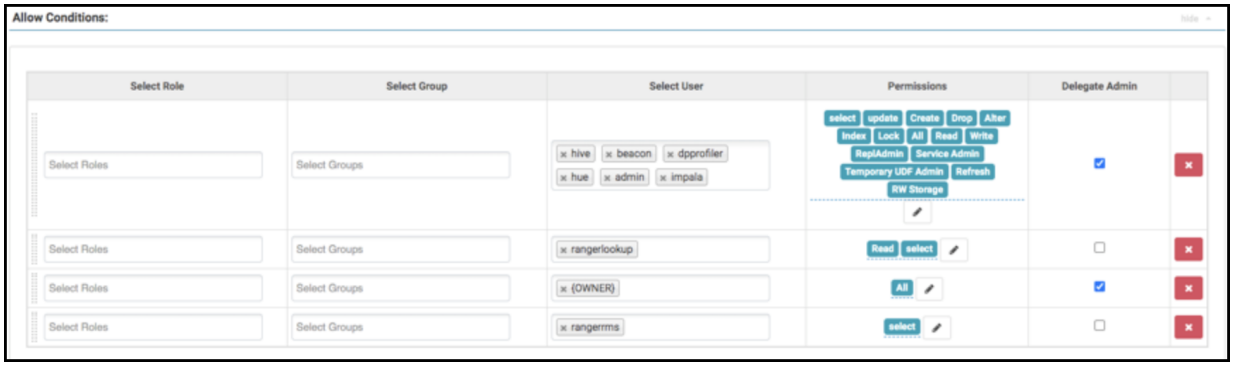
10. Log in to the Ranger Admin web UI. On the **Service Manager** page, click Edit for the Hadoop SQL service, then verify that hdfs has been added to the Policy Download Users and Tag Download Users configurations.



11. Configure Ranger policies with rangerrms user access before starting RMS and running the first sync from the Hive Metastore (HMS).

For example, you must give the rangerrms ID select access to Hive tables. This is configured under the policy "all - database, table".

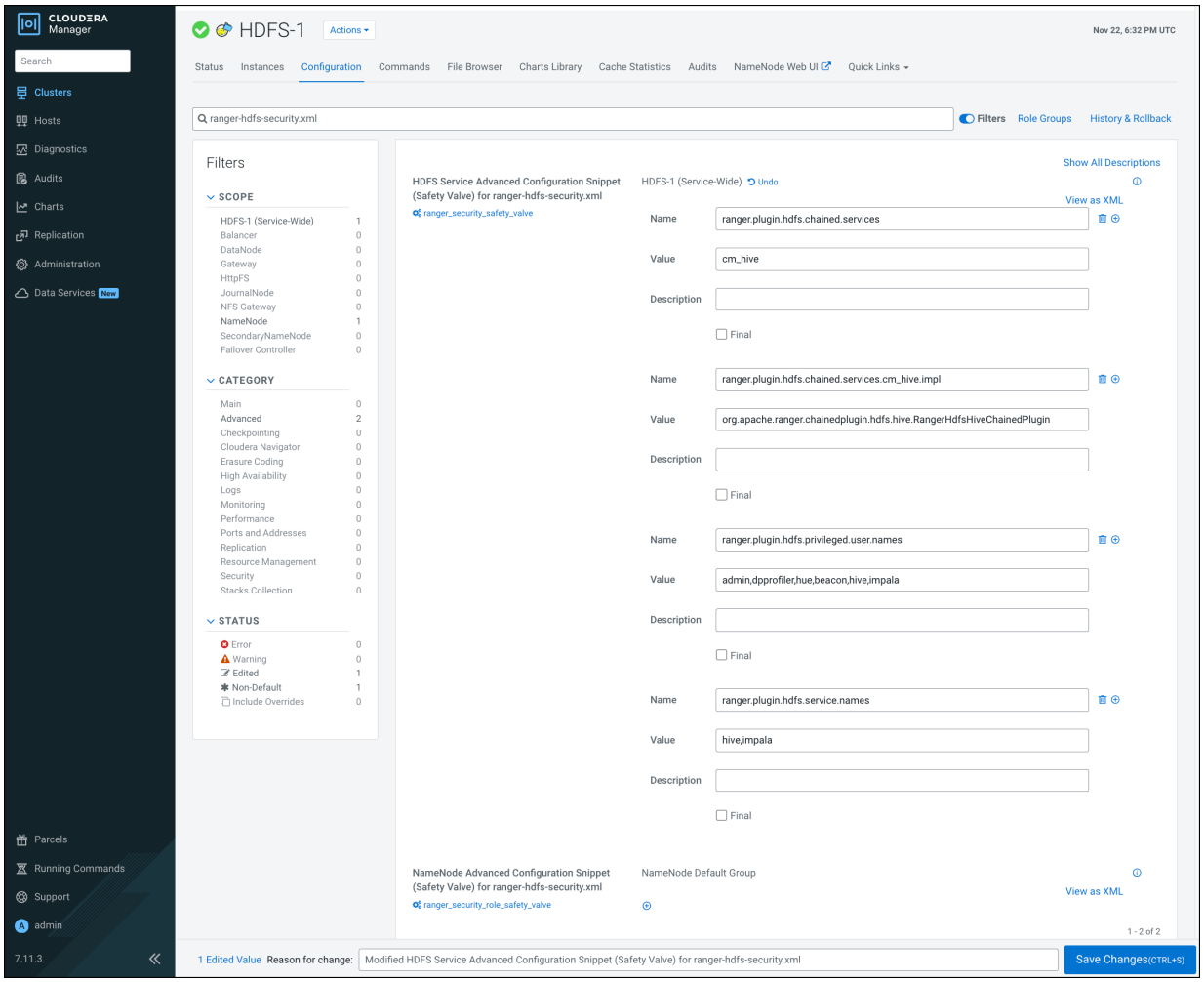
Figure 3: Granting RMS user Select access to Hive tables



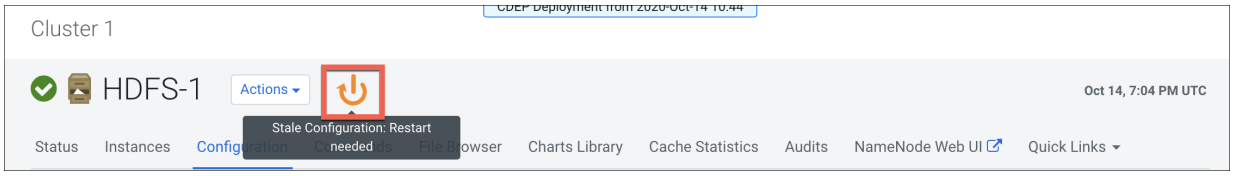
12. In Cloudera Manager, select HDFS Configuration , then search for Advanced Configuration Snippet (Safety Valve) for ranger-hdfs-security.xml. Use the Add (+) icons to add the following properties, then click Save Changes.

Name	Value
ranger.plugin.hdfs.chained.services	cm_hive
ranger.plugin.hdfs.chained.services.cm_hive.impl	org.apache.ranger.chainedplugin.hdfs.hive.RangerHdfsHiveChainedPlugin
ranger.plugin.hdfs.privileged.user.names	admin,dpprofiler,hue,beacon,hive,impala
ranger.plugin.hdfs.service.names	hive,impala

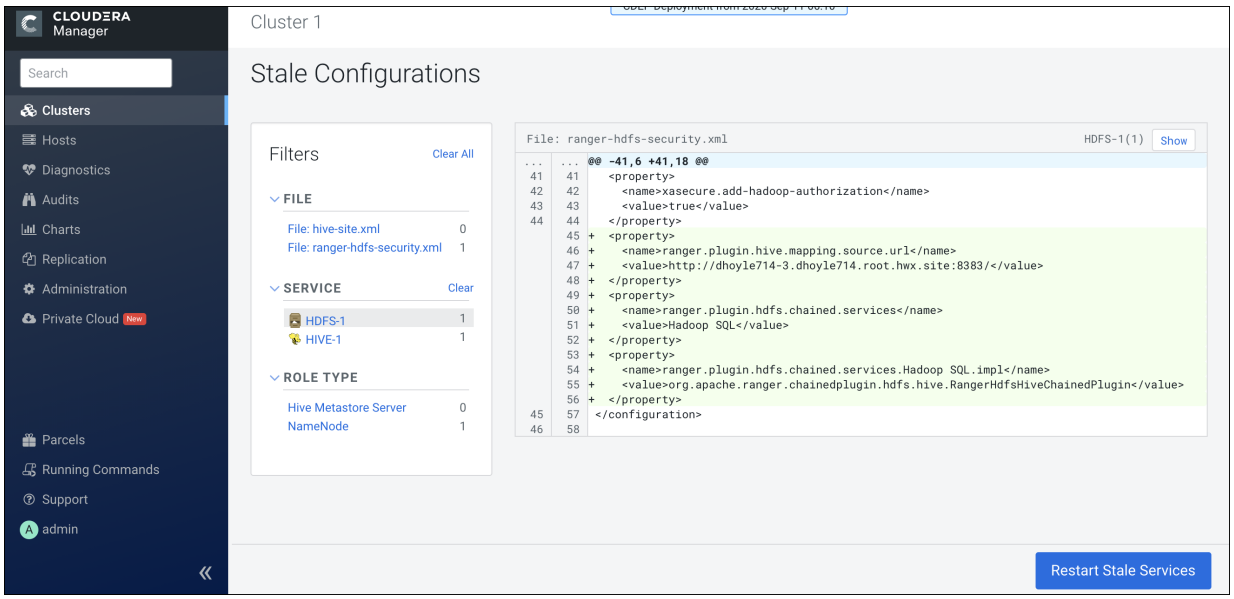
Note: The comma-separated lists that you define for hdfs privileged user names and service names are users that, based on default Hive policies, have all access permissions for all Hive resources. Therefore, for these users, checking Hive policies when they access storage locations which map to Hive resources is unnecessary, and may cause access violations if masking/row-filtering policies are configured for public group.



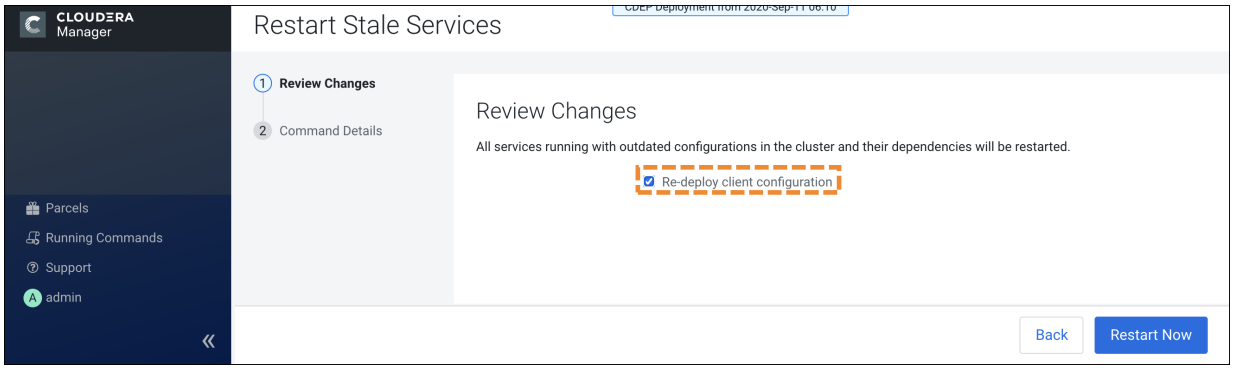
13. Click HDFS Restart.



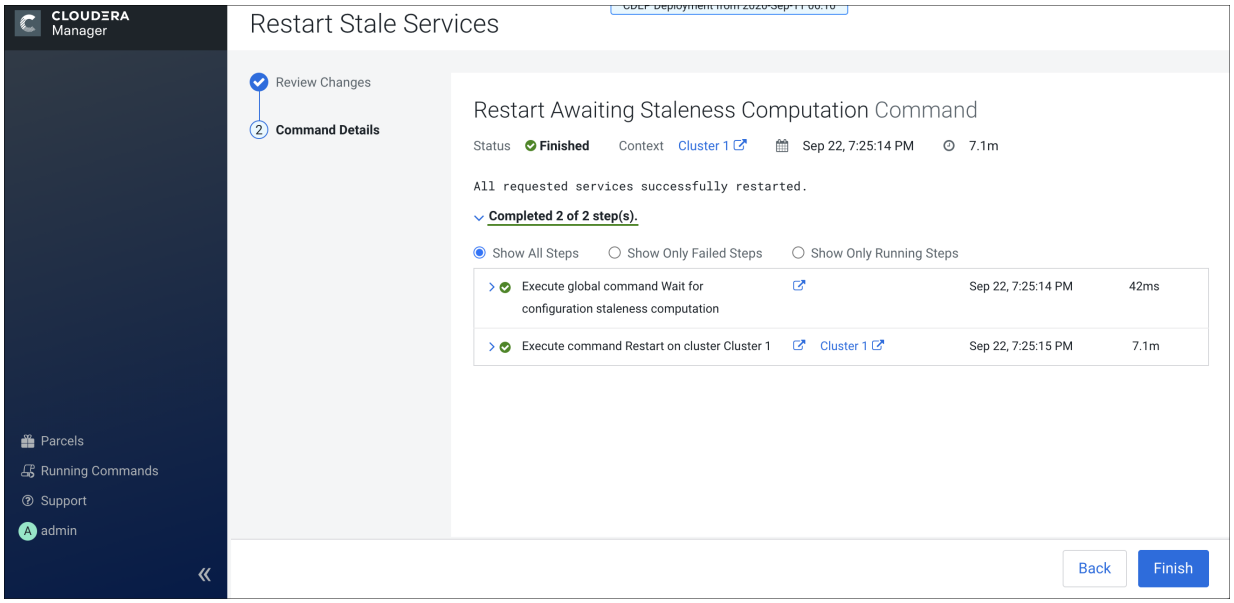
14. On the **Stale Configurations** page, click Restart Stale Services.



15. On the **Restart Stale Services** page, select the Re-deploy client configuration option, then click Restart Now.



16. Click Finish after the services restart.



Related Information

Configuring and Using Hive-HDFS ACL Sync

Installation Reference

Reference information related to Cloudera Base on premises installation.

Related Information
Cloudera Base on premises Installation Guide

Ports

Cloudera Manager, Cloudera Runtime components, managed services, and third-party components use the ports listed in the tables that follow.

Before you deploy Cloudera Manager, Cloudera Runtime, managed services, and third-party components, make sure these ports are open on each system. If you are using a firewall, such as iptables or firewalld, and cannot open all the listed ports, you must disable the firewall completely to ensure full functionality.

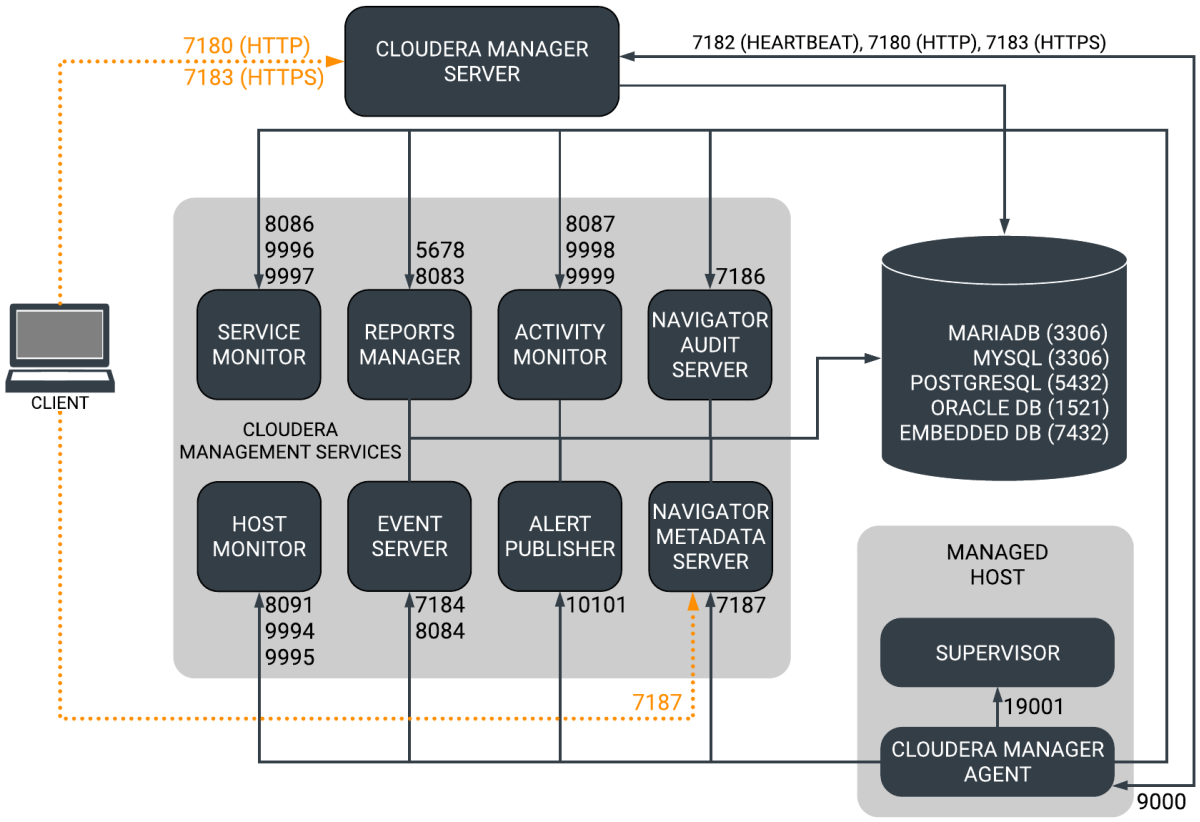
In the tables in the subsections that follow, the Access Requirement column for each port is usually either "Internal" or "External." In this context, "Internal" means that the port is used only for communication among the components (for example the JournalNode ports in an HA configuration); "External" means that the port can be used for either internal or external communication (for example, ports used by NodeManager and the JobHistory Server Web UIs).

Unless otherwise specified, the ports access requirement is unidirectional, meaning that inbound connections to the specified ports must be allowed. In most modern stateful firewalls, it is not necessary to create a separate rule for return traffic on a permitted session.

Ports Used by Cloudera Manager

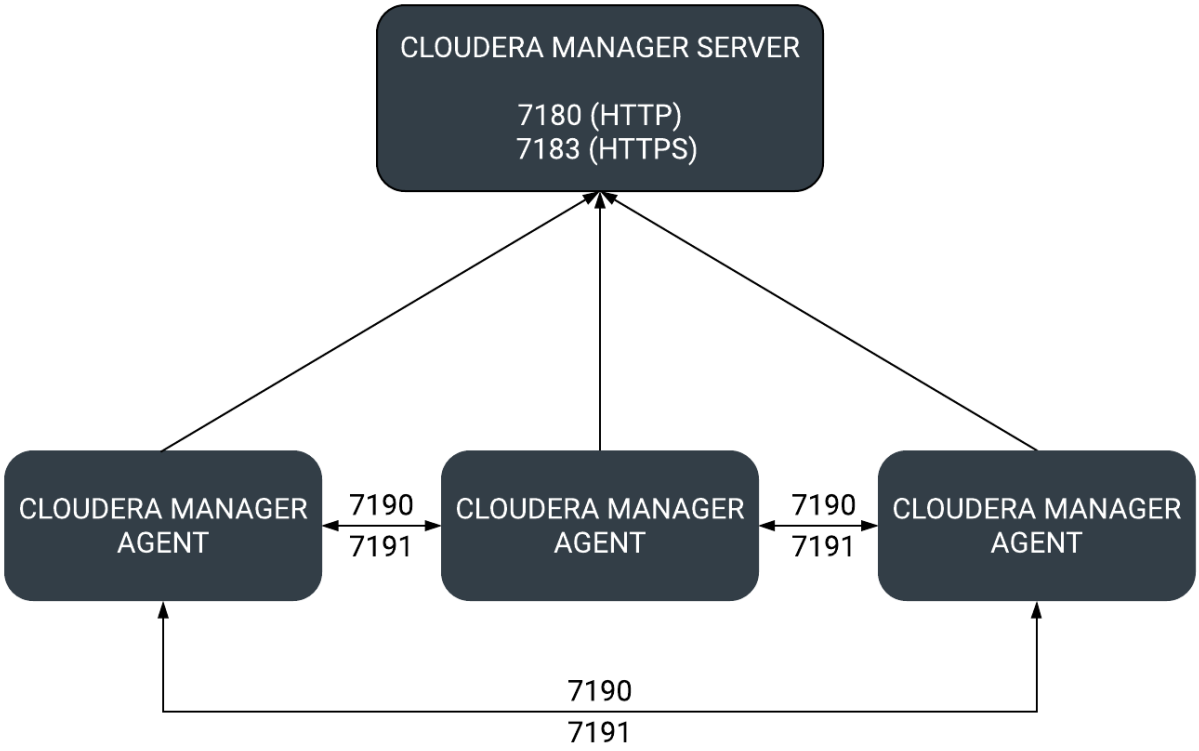
The diagrams and tables below provide an overview of some of the ports used by Cloudera Manager and Cloudera Management Service roles.

Figure 4: Ports Used by Cloudera Manager



When peer-to-peer distribution is enabled for parcels, the Cloudera Manager Agent can obtain the parcel from the Cloudera Manager Server or from other agents, as follows:

Figure 5: Ports Used in Peer-to-Peer Parcel Distribution



For further details, see the following tables. All ports listed are TCP.

In the following tables, Internal means that the port is used only for communication among the components; External means that the port can be used for either internal or external communication.

Table 24: External Ports

Component	Service	Port	Configuration	Description
Cloudera Manager Server	HTTP (Web UI)	7180	AdministrationSettingsCategoryPorts and AddressesHTTP Port for Admin Console	HTTP port used by the web console.
	HTTPS (Web UI)	7183	AdministrationSettingsCategoryPorts and AddressesHTTPS Port for Admin Console	Port used by the web console if HTTPS is enabled. If enabled, port 7180 remains open, but redirects all requests to HTTPS on port 7183.
Cloudera Manager Agent	HTTP (Debug)	9000	/etc/cloudera-scm-agent/config.ini	
Backup and Disaster Recovery	HTTP (Web UI)	7180	AdministrationSettingsCategoryPorts and AddressesHTTP Port for Admin Console	Used for communication to peer (source) Cloudera Manager.
	HTTPS (Web UI)	7183	AdministrationSettingsCategoryPorts and AddressesHTTPS Port for Admin Console	Used for communication to peer (source) Cloudera Manager when HTTPS is enabled.

Component	Service	Port	Configuration	Description
	HDFS NameNode	8020	HDFS serviceConfigurationCategoryPorts and AddressesNameNode Port	HDFS and Hive/ Impala replication: communication from destination HDFS and MapReduce hosts to source HDFS NameNode(s). Hive/ Impala Replication: communication from source Hive hosts to destination HDFS NameNode(s).
	HDFS DataNode	9866	HDFS serviceConfigurationCategoryPorts and AddressesDataNode Transceiver Port	HDFS and Hive/ Impala replication: communication from destination HDFS and MapReduce hosts to source HDFS DataNode(s). Hive/ Impala Replication: communication from source Hive hosts to destination HDFS DataNode(s).
Telemetry Publisher	HTTP	10110	Clusters Cloudera Management Service Category Ports and Addresses Telemetry Publisher Server Port	The port where the Telemetry Publisher Server listens for requests
Telemetry Publisher	HTTP (Debug)	10111	Clusters Cloudera Management Service Category Ports and Addresses Telemetry Publisher Web UI Port	The port where Telemetry Publisher starts a debug web server. Set to -1 to disable debug server.

Table 25: Internal Ports

Component	Service	Port	Configuration	Description
Cloudera Manager Server	Avro (RPC)	7182	Administration Settings Category Ports and Addresses Agent Port to connect to Server	Used for Agent to Server heartbeats
	Embedded PostgreSQL database	7432		The optional embedded PostgreSQL database used for storing configuration information for Cloudera Manager Server.
	Peer-to-peer parcel distribution	7190, 7191	Hosts All Hosts Configuration P2P Parcel Distribution Port	Used to distribute parcels to cluster hosts during installation and upgrade operations.
Cloudera Manager Agent	HTTP (Debug)	The value set for the listening_port parameter in the /etc/cloudera-scm-agent/config.ini file, plus 1.	Not directly configurable. For example, the default external port is 9000. Therefore the default internal port is 9001.	
	supervisord	19001	For example, the default internal port is 19001. You can configure the custom port under /etc/cloudera-scm-agent/ config.ini by updating supervisord_port.	This port is used to start the supervisord process.

Component	Service	Port	Configuration	Description
Event Server	Custom protocol	7184	Cloudera Management Service Configuration Category Ports and Addresses Event Publish Port	Port on which the Event Server listens for the publication of events.
	Custom protocol	7185	Cloudera Management Service Configuration Category Ports and Addresses Event Query Port	Port on which the Event Server listens for queries for events.
	HTTP (Debug)	8084	Cloudera Management Service Configuration Category Ports and Addresses Event Server Web UI Port	Port for the Event Server's Debug page. Set to -1 to disable debug server.
Alert Publisher	Custom protocol	10101	Cloudera Management Service Configuration Category Ports and Addresses Alerts: Listen Port	Port where the Alert Publisher listens for internal API requests.
Service Monitor	HTTP (Debug)	8086	Cloudera Management Service Configuration Category Ports and Addresses Service Monitor Web UI Port	Port for Service Monitor's Debug page. Set to -1 to disable the debug server.
	HTTPS (Debug)		Cloudera Management Service Configuration Category Ports and Addresses Service Monitor Web UI HTTPS Port	Port for Service Monitor's HTTPS Debug page.
	Custom protocol	9997	Cloudera Management Service Configuration Category Ports and Addresses Service Monitor Listen Port	Port where Service Monitor is listening for agent messages.
	Internal query API (Avro)	9996	Cloudera Management Service Configuration Category Ports and Addresses Service Monitor Nozzle Port	Port where Service Monitor's query API is exposed.
Activity Monitor	HTTP (Debug)	8087	Cloudera Management Service Configuration Category Ports and Addresses Activity Monitor Web UI Port	Port for Activity Monitor's Debug page. Set to -1 to disable the debug server.
	HTTPS (Debug)		Cloudera Management Service Configuration Category Ports and Addresses Activity Monitor Web UI HTTPS Port	Port for Activity Monitor's HTTPS Debug page.
	Custom protocol	9999	Cloudera Management Service Configuration Category Ports and Addresses Activity Monitor Listen Port	Port where Activity Monitor is listening for agent messages.
	Internal query API (Avro)	9998	Cloudera Management Service Configuration Category Ports and Addresses Activity Monitor Nozzle Port	Port where Activity Monitor's query API is exposed.
Host Monitor	HTTP (Debug)	8091	Cloudera Management Service Configuration Category Ports and Addresses Host Monitor Web UI Port	Port for Host Monitor's Debug page. Set to -1 to disable the debug server.
	HTTPS (Debug)	9091	Cloudera Management Service Configuration Category Ports and Addresses Host Monitor Web UI HTTPS Port	Port for Host Monitor's HTTPS Debug page.
	Custom protocol	9995	Cloudera Management Service Configuration Category Ports and Addresses Host Monitor Listen Port	Port where Host Monitor is listening for agent messages.
	Internal query API (Avro)	9994	Cloudera Management Service Configuration Category Ports and Addresses Host Monitor Nozzle Port	Port where Host Monitor's query API is exposed.
Reports Manager	Queries (Thrift)	5678	Cloudera Management Service Configuration Category Ports and Addresses Reports Manager Server Port	The port where Reports Manager listens for requests.

Component	Service	Port	Configuration	Description
	HTTP (Debug)	8083	Cloudera Management Service Configuration Category Ports and Addresses Reports Manager Web UI Port	The port where Reports Manager starts a debug web server. Set to -1 to disable debug server.

Ports Used by Cloudera Runtime Components

Cloudera Runtime components use a number of ports for associated services.

All ports listed are TCP.

In the following tables, Internal means that the port is used only for communication among the components; External means that the port can be used for either internal or external communication.

Table 26: External Ports

Component	Service	Port	Configuration	Comment
Apache Atlas	Non-SSL	31000	atlas.server.http.port	
	SSL	31443	atlas.server.https.port	This port is used only when Atlas is in SSL mode.
Apache Hadoop HDFS	DataNode	9866	dfs.datanode.address	DataNode server address and port for data transfer.
		9864	dfs.datanode.http.address	DataNode HTTP server port.
		9865	dfs.datanode.https.address	DataNode HTTPS server port.
		9867	dfs.datanode.ipc.address	DataNode IPC server port.
	NameNode	8020	fs.default.name or fs.defaultFS	fs.default.name is deprecated (but still works)
		9870	dfs.http.address or dfs.namenode.http-address	dfs.http.address is deprecated (but still works)
		9871	dfs.https.address or dfs.namenode.https-address	dfs.https.address is deprecated (but still works)
	NFS gateway	2049		nfs port (nfs3.server.port)
		4242		mountd port (nfs3.mountd.port)
		111		portmapper or rpcbind port.
		50079	nfs.http.port	The NFS gateway daemon uses this port to serve metrics. The port is configurable on versions 5.10 and higher.
		50579	nfs.https.port	The NFS gateway daemon uses this port to serve metrics. The port is configurable on versions 5.10 and higher.
	HttpFS	14000		HttpFS server port
		14001		HttpFS admin port
	Secondary NameNode	9868	dfs.namenode.secondary.http-address	The HTTP server address and port for the Secondary NameNode's web UI.
		9869	dfs.namenode.secondary.https-address	The secure HTTPS server address and port for encrypted web UI access.
	JournalNode	8480	dfs.journalnode.http-address	The HTTP server address and port for the JournalNode web UI.

Component	Service	Port	Configuration	Comment
		8481	dfs.journalnode.https-address	The secure HTTPS server address and port for encrypted web UI access.
Apache Hadoop YARN (MRv2)	ResourceManager	8032	yarn.resourcemanager.address	ResourceManager listens for client RPC traffic (default 8032). Clients use it to submit, kill, and monitor YARN applications via the ApplicationClientProtocol. It is the main entry point for users and tools (like yarn jar or the YARN client API) to talk to the RM, not for NodeManagers or ApplicationMasters.
		8033	yarn.resourcemanager.admin.address	ResourceManager listens for administrative RPC traffic (default 8033). Cluster admins use it for RM admin commands such as refreshing queues, nodes, or user/group mappings, and for HA operations like checking or changing active/standby state via yarn rmadmin. It is meant for privileged cluster management, not for normal job submission or container scheduling.
		8088	yarn.resourcemanager.webapp.address	ResourceManager serves its web UI over HTTP (default 8088). Users and admins use it in a browser to view cluster status, applications, queues, nodes, and logs without going through the RPC client port. It is for monitoring and troubleshooting, not for submitting jobs or for NodeManager/ ApplicationMaster internal RPC communication.
		8090	yarn.resourcemanager.webapp.https.address	ResourceManager serves its web UI over HTTPS (default 8090). It provides the same monitoring pages as the HTTP webapp, but with encrypted browser traffic when TLS is enabled. You use it when yarn.http.policy is set to HTTPS_ONLY or when you want secure access to the RM UI instead of plain HTTP on port 8088.
	NodeManager	8042	yarn.nodemanager.webapp.address	NodeManager serves its web UI over HTTP (default 8042). Users and admins use it in a browser to view node status, running containers, and container logs on that specific host. It is for monitoring and troubleshooting, not for RM-to-NM RPC or container lifecycle management.
		8044	yarn.nodemanager.webapp.https.address	NodeManager serves its web UI over HTTPS (default 8044). It provides the same monitoring pages as the HTTP webapp, but with encrypted browser traffic when TLS is enabled. You use it when yarn.http.policy is set to HTTPS_ONLY or when you want secure access to the NM UI instead of plain HTTP on port 8042.
	JobHistory Server	19888	mapreduce.jobhistory.webapp.address	MapReduce Job History Server serves its web UI over HTTP (default 19888). Users and admins use it in a browser to view finished MapReduce jobs, task details, counters, and links to container logs.

Component	Service	Port	Configuration	Comment
		19890	mapreduce.jobhistory.webapp.https.address	MapReduce Job History Server serves its web UI over HTTPS (default 19890). It provides the same job history pages as the HTTP webapp, but with encrypted browser traffic when TLS is enabled.
Apache HBase	Master	16000	hbase.master.port	IPC
		16010	hbase.master.info.port	HTTP
	RegionServer	16020	hbase.regionserver.port	IPC
		16030	hbase.regionserver.info.port	HTTP
	REST	20550	hbase.rest.port	The default REST port in HBase is 8080. Because this is a commonly used port, Cloudera Manager sets the default to 20550 instead.
	REST UI	8085	hbase.rest.info.port	
	HBase Thrift Server	9090	hbase.regionserver.thrift.port	
	HBase Thrift Server Web UI	9095		
Apache Hive	Metastore	9083		
	HiveServer2	10000	hive.server2.thrift.port	The Beeline command interpreter requires that you specify this port on the command line. If you use Oracle database, you must manually reserve this port.
	HiveServer2 Web User Interface (UI)	10002	hive.server2.webui.port in hive-site.xml	
Hue	Server	8888		
	Load Balancer	8889		
Apache Impala	Impala Daemon	21000		Used to transmit commands and receive results by impala-shell and version 1.2 of the Cloudera ODBC driver.
		21050		Used to transmit commands and receive results by applications, such as Business Intelligence tools, using JDBC, the Beeswax query editor in Hue, and version 2.0 or higher of the Cloudera ODBC driver.
		25000		Impala web interface for administrators to monitor and troubleshoot.
		28000		Used to transmit commands and receive results by client applications over HTTP through the HiveServer2 protocol.
	StateStore Daemon	25010		StateStore web interface for administrators to monitor and troubleshoot.
	Catalog Daemon	25020		Catalog service web interface for administrators to monitor and troubleshoot.
Apache Kafka	Kafka Broker	9092	port	The primary communication port used by producers and consumers; also used for inter-broker communication.

Component	Service	Port	Configuration	Comment
		9093	ssl_port	A secured communication port used by producers and consumers; also used for inter-broker communication.
		24042	kafka.http.metrics.port	The port that the HTTP metric reporter listens on.
		9393	jmx_port	Port for JMX.
	KRaft Controller	9192	port	KRaft server port.
		9193	ssl_port	KRaft server secure port.
		24043	kafka.http.metrics.port	The port that the HTTP metric reporter listens on.
		9395	jmx_port	Port for JMX.
	Kafka Connect	28083	rest.port	Kafka Connect Rest Port.
		28085	secure.rest.port	Kafka Connect Secure Rest Port
		28084	metrics.jetty.server.port	Jetty Metrics Port
		28087	metrics.jetty.server.secureport	Secure Jetty Metrics Port
Apache Knox	Knox Gateway	8443	gateway.port	The HTTPS port for the Gateway
	Knox Gateway (HTTPS)	8444	idbroker_gateway_port	
Apache Kudu	Master	7051		Kudu Master RPC port.
		8051		Kudu Master HTTP server port.
	TabletServer	7050		Kudu TabletServer RPC port.
		8050		Kudu TabletServer HTTP server port.
Apache Oozie	Oozie Server	11443	oozie_https_port	HTTPS
Apache Ozone	Ozone Manager	9862	ozone.om.rpc-port	RPC endpoint for clients and applications.
		9874	ozone.om.http-port	HTTP port for the Ozone Manager web UI.
		9875	ozone.om.https-port	HTTPS port for the Ozone Manager web UI.
	Storage Container Manager	9876	ozone.scm.http-port	HTTP port for the SCM UI.
		9877	ozone.scm.https-port	HTTPS port for the SCM web UI.
		9860	ozone.scm.client.port	Port used by the Ozone Manager and other clients to communicate with the SCM for container operations.
	DataNode	9882	hdds.datanode.http-address	HTTP port for the DataNode web UI.
		9883	hdds.datanode.https-address	HTTPS port for the DataNode web UI.
		9858	dfs.container.ratis.ipc	RAFT server endpoint that is used by clients and other DataNodes to replicate RAFT transactions and write data.
		9859	dfs.container.ipc	Endpoint that is used by clients and other DataNodes to read block data.
	S3 Gateway	9878	ozone.s3g.http-port	HTTP port for the S3 API REST endpoint and web UI.
		9879	ozone.s3g.https-port	HTTPS port for the S3 API REST endpoint and web UI.

Component	Service	Port	Configuration	Comment
	Recon Service	9891	ozone.recon.rpc-port	Port used by DataNodes to communicate with the Recon Server.
		9888	ozone.recon.http-port	HTTP port for the Recon service web UI and REST API.
		9889	ozone.recon.https-port	HTTPS port for the Recon service web UI and REST API.
	HttpFS	9778	httpfs.http.port	The TCP port on which the Ozone HttpFS gateway listens to handle incoming WebHDFS REST API requests.
Apache Ranger	Non-SSL	6080	ranger.service.http.port	
	SSL	6182	ranger.service.https.port	This port is used only when Ranger is in SSL mode.
	Admin Unix Auth Service Port	5151	ranger.unixauth.service.port	
	Usersync HTTP Port	8280	ranger.usersync.service.http.port	HTTP port for Ranger Usersync
	Usersync HTTPS Port	8283	ranger.usersync.service.https.port	HTTPS port for Ranger Usersync
	Tagsync HTTP Port	8180	ranger.tagsync.service.http.port	HTTP port for Ranger Tagsync
	Tagsync HTTPS Port Port	8183	ranger.tagsync.service.https.port	HTTPS port for Ranger Tagsync
Ranger KMS	Ranger KMS nodes	9292	ranger.service.http.port	HTTP port for Ranger KMS.
	Ranger KMS nodes	9494	ranger.service.https.port	HTTPS port for Ranger KMS. Only used when SSL is enabled for Ranger KMS.
Ranger RMS	Ranger RMS nodes	8383	ranger-rms.service.http.port	HTTP port for Ranger RMS.
	Ranger RMS nodes	8484	ranger-rms.service.https.port	HTTPS port for Ranger RMS. Only used when SSL is enabled for Ranger RMS.
Apache Solr	Solr Server	8983		HTTP port for all Solr-specific actions, update/query.
	Solr Server	8985		HTTPS port for all Solr-specific actions, update/query.
	Lily HBase Indexer	11060		
Apache Spark	Shuffle service	7447	spark.shuffle.service.port	Port on which the Spark external shuffle service runs.
	History Server	18089	spark.history.ui.port	HTTP port for the Spark History Server WebUI.
	History Server with TLS	18489	spark.ssl.historyServer.port	HTTPS port for Spark History Server WebUI. Only used when SSL is enabled for Spark History Server.
Apache Sqoop	Metastore	16000	sqoop.metastore.server.port	
Apache ZooKeeper	Server (with Cloudera Runtime or Cloudera Manager)	2181	clientPort	Client port.
	Server (with Cloudera Runtime or Cloudera Manager) (SSL)	2182	secureClientPort	Secure client port.

Component	Service	Port	Configuration	Comment
	Admin Server port	5181		Port number for the REST Admin Server embedded in ZooKeeper. (disabled by default).
	Prometheus Metrics Provider HTTP port	7000		Port number for Zookeeper's Prometheus MetricsProvider.
Cruise Control	Cruise Control Server	8899	webserver.http.port	This is the main port that enables access to the Cruise Control Server
Livy	Livy Server Web UI	28998	livy.server.port	
	Livy Thrift Server	20091	livy.server.thrift.port	
Omid	TSO Server	54758		
Schema Registry	Schema Registry Server	7788	schema.registry.port	REST endpoint for Schema Registry.
		7790	schema.registry.ssl.port	When SSL is enabled, REST endpoint for Schema Registry.
Streams Messaging Manager	Streams Messaging Manager Rest Admin Server	8585	streams.messaging.manager.port	Streams Messaging Manager Port
		8587	streams.messaging.manager.ssl.port	Streams Messaging Manager Port (SSL)
	Streams Messaging Manager UI Server	9991	streams.messaging.manager.ui.port	The port on which server accepts connections. This port is used for both secured and unsecured connections.
Streams Replication Manager	SRM Service	6670	streams.replication.manager.service.port	SRM Service port.
		6671	streams.replication.manager.service.ssl.port	SRM Service port when SSL is enabled.
	SRM Driver	38083	rest.port	The port that the REST API listens on for connection requests. This port is used by SRM Driver if TLS/SSL is disabled.
		38085	secure.rest.port	The secure port that the REST API listens on for connection requests. This port is used by SRM Driver if TLS/SSL is enabled.
		26001	metrics.jetty.server.port	The port that the HTTP metrics reporter listens on.
		26002	metrics.jetty.server.secureport	The port that the HTTPS metrics reporter listens on.
	SRM Service	6672	streams.replication.manager.service.metric.port	SRM Service Metrics port.
		6673	streams.replication.manager.service.metric.ssl.port	SRM Service Metrics https port.

Table 27: Internal Ports

Component	Service	Port	Configuration	Comment
Apache Hadoop HDFS	JournalNode	8485	dfs.journalnode.rpc-address	The RPC server address and port that Active and Standby NameNodes use to communicate with the JournalNode to read and write edit logs in High Availability (HA) configurations.
	NameNode	8022	dfs.namenode.servicerpc-address	Optional port used by HDFS daemons to avoid sharing the RPC port used by clients (8020). Cloudera recommends using port 8022.

Component	Service	Port	Configuration	Comment
	Failover Controller	8019	dfs.ha.zkfc.port	Used for NameNode HA. The IPC/RPC port used by the ZooKeeper Failover Controller (ZKFC) daemon to manage health checks and trigger automatic failovers between NameNodes in High Availability setups.
Apache Hadoop YARN (MRv2)	ResourceManager	8030	yarn.resourcemanager.scheduler.address	ResourceManager listens for ApplicationMaster RPC traffic (default 8030). ApplicationMasters use it to register, send heartbeats, and request or release containers from the scheduler. It is the main internal link between running applications and the RM's scheduling logic for allocating cluster resources.
		8031	yarn.resourcemanager.resource-tracker.address	ResourceManager listens for NodeManager RPC traffic (default 8031). NodeManagers use it to register with the RM, send heartbeats, report container status, and receive container start/stop commands. It is the main internal link between the RM and NMs for tracking cluster nodes and the containers running on them.
	NodeManager	8040	yarn.nodemanager.localizer.address	NodeManager runs its localizer service (default 8040). The localizer downloads job resources (jars, files, splits) from HDFS or other storage onto the node's local disk before containers start. It is internal NM infrastructure for staging application data, not for users, the RM web UI, or normal container RPC on yarn.nodemanager.address.
		8041	yarn.nodemanager.address	NodeManager listens for ContainerManager RPC traffic. The ResourceManager, ApplicationMasters, and containers use it to start/stop containers, report status, and manage lifecycle on that node.
	JobHistory Server	10020	mapreduce.jobhistory.address	MapReduce Job History Server listens for RPC/API traffic (default 10020). MapReduce ApplicationMasters and clients use it to publish and query job history events and metadata for completed or running MR jobs.
		10033	mapreduce.jobhistory.admin.address	MapReduce Job History Server listens for administrative RPC traffic (default 10033). Cluster admins use it for privileged JHS operations such as refreshing ACLs and other admin protocol commands.
	Shuffle HTTP	13562	mapreduce.shuffle.port	NodeManager's MapReduce shuffle aux service (ShuffleHandler) listens (default 13562). After map tasks finish, reduce tasks connect to this port on the node that ran the map to fetch intermediate map output data over HTTP.

Component	Service	Port	Configuration	Comment
	Queue Manager	8082	queuemanager_webapp_port	Port 8082 is the default HTTP port for the Queue Manager backend in this project. It serves both the REST API and the Queue Manager web UI.
	Config Store/Service	8080	config_service_application_connector_port	Port 8080 is the main Config Service REST API for namespaces, config sets, configs, and related operations.
	Queue Manager Config-Service	8081	config_service_admin_connector_port	Admin interface for ops endpoints like health checks and metrics, not the business API.
Apache HBase	HQuorumPeer	2181	hbase.zookeeper.property.clientPort	HBase-managed ZooKeeper mode
		2888	hbase.zookeeper.peerport	HBase-managed ZooKeeper mode
		3888	hbase.zookeeper.leaderport	HBase-managed ZooKeeper mode
Apache Impala	Impala Daemon	23000		Internal use only. Impala daemons listen on this port for updates from the statestore daemon.
		27000		Internal use only. Impala daemons use this port for KRPC based communication with each other.
	StateStore Daemon	24000		Internal use only. The statestore daemon listens on this port for registration/unregistration requests.
	Catalog Daemon	23020		Internal use only. The catalog daemon listens on this port for updates from the statestore daemon.
		26000		Internal use only. The catalog service uses this port to communicate with the Impala daemons.
Apache Kafka	Kafka Broker	9092	port	The primary communication port used by producers and consumers; also used for inter-broker communication.
		9093	ssl_port	A secured communication port used by producers and consumers; also used for inter-broker communication.
		9393	jmx_port	Internal use only. Used for administration via JMX.
		9394	kafka.http.metrics.port	Internal use only. This is the port via which the HTTP metric reporter listens. It is used to retrieve metrics through HTTP instead of JMX.
	Kafka MirrorMaker	24042	jmx_port	Internal use only. Used to administer the producer and consumer of the MirrorMaker.
Apache Ozone	Ozone Manager	9872	ozone.om.ratis-port	RPC endpoint for Ozone Manager HA instances to form a RAFT consensus ring.
	Storage Container Manager	9861	ozone.scm.datanode.port	Port used by the DataNodes to communicate with the Storage Container Manager (SCM).
		9863	ozone.scm.block.client.port	Port used by the Ozone Manager to communicate with the SCM for block related operations.

Component	Service	Port	Configuration	Comment
		9894	ozone.scm.ratis.port	Port used by the SCM to communicate with other SCMs using Ratis.
		9895	ozone.scm.grpc.port	Port used by the SCM to communicate with other SCMs about the database checkpoint downloads.
Apache Phoenix	Phoenix Query Server Port	8765	phoenix.queryserver.http.port	
Apache Solr	Solr Server	8993		Infra-Solr HTTP port
	Solr Server	8995		Infra-Solr HTTPS port
Apache ZooKeeper	Server (with Cloudera Runtime only)	2888	X in server.N =host:X:Y	Quorum port
	Server (with Cloudera Runtime only)	3888	X in server.N =host:X:Y	Leader election port
	Server (with Cloudera Runtime and Cloudera Manager)	3181	X in server.N =host:X:Y	Quorum port
	Server (with Cloudera Runtime and Cloudera Manager)	4181	X in server.N =host:X:Y	Leader election port
	ZooKeeper JMX port	9010		ZooKeeper will also use another randomly selected port for RMI. To allow Cloudera Manager to monitor ZooKeeper, you must do one of the following: • Open up all ports when the connection originates from the Cloudera Manager server. • Do the following: 1. Open a non-ephemeral port (such as 9011) in the firewall. 2. Install Oracle Java 7u4 JDK or higher. 3. Add the port configuration to the advanced configuration snippet, for example: -Dcom.sun.management.jmxremote.rmi.port=9011 4. Restart ZooKeeper.

Ports Used by DistCp

DistCp uses various ports for HDFS and HttpFS services.

All ports listed are TCP.

In the following table, the Access Requirement column for each port is usually either "Internal" or "External." In this context, "Internal" means that the port is used only for communication among the components; "External" means that the port can be used for either internal or external communication.

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
Hadoop HDFS	NameNode		8020	External	`fs.default.name` or `fs.defaultFS`	`fs.default.name` is deprecated (but still works)
	DataNode	Secure	1004	External	`dfs.datanode.address`	
	DataNode		50010	External	`dfs.datanode.address`	
WebHDFS	NameNode		50070	External	`dfs.http.address` or `dfs.namenode.http-address`	`dfs.http.address` is deprecated (but still works)
	DataNode	Secure	1006	External	`dfs.datanode.http.address`	
HttpFS	web		14000			

Ports Used by Third-Party Components

Third-party components such as PostgreSQL and LDAP use a number of ports for associated services.

In the following table, the Access Requirement column for each port is usually either "Internal" or "External." In this context, "Internal" means that the port is used only for communication among the components; "External" means that the port can be used for either internal or external communication.

Component	Service	Qualifier	Port	Protocol	Access Requirement	Configuration	Comment
Ganglia	ganglia-gmond		8649	UDP/TCP	Internal		

Component	Service	Qualifier	Port	Protocol	Access Requirement	Configuration	Comment
	ganglia-web		80	TCP	External	Via Apache `httpd`	
Kerberos	KRB5 KDC Server	Secure	88	UDP/TCP	External	`kdc_ports` and `kdc_tcp_ports` in either the `[kdcdefaults]` or `[realms]` sections of `kdc.conf`	By default only UDP
	KRB5 Admin Server	Secure	749	TCP	External	`kadmind_port` in the `[realms]` section of `kdc.conf`	
	kpasswd		464	UDP/TCP	External		
SSH	ssh		22	TCP	External		
PostgreSQL			5432	TCP	Internal		
MariaDB			3306	TCP	Internal		
MySQL			3306	TCP	Internal		
LDAP	LDAP Server		389	TCP	External		
	LDAP Server over TLS/SSL	TLS/SSL	636	TCP	External		
	Global Catalog		3268	TCP	External		
	Global Catalog over TLS/SSL	TLS/SSL	3269	TCP	External		

Service Dependencies in Cloudera Manager

The following tables list service dependencies that exist between various services in a Cloudera Manager deployment.

When configuring Cloudera Runtime for production environments, be sure that Kerberos is enabled for user authentication. Cloudera supports security services such as Ranger and Atlas when they run on clusters where Kerberos is enabled to authenticate users.

Service dependencies for Spark 2 on YARN and Cloudera Data Science Workbench are listed separately.

Table 28: Service Dependencies

Service	Dependencies	Optional Dependencies
ADLS Connector		
Atlas	• HDFS • HBase • Kafka (Kafka broker role only) • Solr	Ranger
Cruise Control	• Kafka • Zookeeper	
Data Context Connector		
HBase	• HDFS • ZooKeeper	• Atlas • Ranger
HDFS		• ADLS Connector or S3 Connector • Ranger KMS • Ranger • ZooKeeper
Hive	HDFS	• Atlas • HBase • Kudu • Ranger • Spark on YARN • YARN • ZooKeeper
Hive-on-Tez	• HDFS • Hive • Tez	• Atlas • HBase • Ranger • YARN • ZooKeeper
Hue	• HDFS • Hive	• Atlas • HBase • Hive-on-Tez • Impala • Oozie • Solr • ZooKeeper
Impala	• HDFS • Hive	• Atlas • HBase • Kudu • Ranger • YARN • ZooKeeper
Kafka	ZooKeeper	• HDFS • Ranger

Service	Dependencies	Optional Dependencies
Key-Value Store Indexer	• HBase • Solr	Ranger
Kudu		Ranger
Livy	• Spark-on-YARN • YARN	Hive
Oozie	YARN	• Hive • Spark on YARN • ZooKeeper
Ozone		• HDFS • Ranger
Ranger	• HDFS • Solr	
S3 Connector		
Schema Registry		• HDFS • Ranger
Solr	• HDFS • ZooKeeper	Ranger
Spark on YARN	YARN	• Atlas • HBase
Streams Messaging Manager	Kafka	• Ranger • Schema Registry • Zookeeper
Streams Replication Manager		Kafka
Tez	YARN	
YARN	• HDFS • ZooKeeper	Ranger
Zeppelin	• HDFS • Spark-on-YARN • YARN	• Livy
ZooKeeper		

Related Information

Cloudera Runtime Cluster Hosts and Role Assignments

Cloudera Manager sudo command options

To install, configure, start and stop the Cloudera Manager (CM), manage files, and so on, you can use the Cloudera Manager sudo commands.

Following is the list of sudo commands run by Cloudera Manager.

Note: In the list, RH = RHEL / CentOS / Oracle , and later, and SLES 11 and later,Ubuntu = All Ubuntu versions, and SLES = All SLES versions. For those command supported in all the Operating System (OS) versions, an OS flavor is not specified.

- sudo yum (RHEL) - Install or remove software.
- sudo apt-get (Ubuntu) - Install or remove software.

- sudo apt-key (Ubuntu) - Update Repository key.
- sudo sed - Edit one or more text files (stream editor).
- sudo systemctl (RHEL, Ubuntu) - Start, stop, or configure software.
- sudo service (RH) - Start or stop software.
- sudo /sbin/chkconfig sudo chkconfig (RH) - Configure software.
- sudo /usr/sbin/update-rc.d (Ubuntu) - Configure software.
- sudo id - Used for user identification.
- sudo rm - Remove files.
- sudo mv - Move or rename files.
- sudo chown - Modify file ownership.
- sudo install - Install software.
- sudo service (RHEL) - Start, stop, or restart the Cloudera Manager Server and Cloudera Manager Agents on the cluster hosts.
- sudo systemctl (RHEL, Ubuntu) - Start, stop, or restart the Cloudera Manager Server and Cloudera Manager Agents on the cluster hosts.
- sudo cp - Used for file copy.
- sudo /opt/cloudera/cm-agent/bin/cm - Used for certificate management and troubleshooting.
- sudo mkdir - Used for directory creation.
- sudo /opt/cloudera/parcels/keycloak/cloudera_keycloak.sh - Configure and startup Keycloak.
- sudo keytrustee - Used for Keystrustee backup.
- sudo ln - Manage file links.
- sudo chmod - Manage file permissions.
- sudo wget - Used to host local repositories for Cloudera Manager and CDH.
- sudo -u postgres psql postgres - Connect to PSQL as postgres user.
- sudo -E tar - Archive Cloudera Manager agent data directories prior to updates or changes.
- sudo zypper clean --all (SLES) - Clean up the repository cache for SLES package manager (zypper).
- sudo rpm (RHEL) - Install or remove the Cloudera Manager RPM packages.

Introduction to Parcels

Parcels are a packaging format that facilitate upgrading software from within Cloudera Manager.

You can download, distribute, and activate a new software version all from within Cloudera Manager. Cloudera Manager downloads a parcel to a local directory. Once the parcel is downloaded to the Cloudera Manager Server host, an Internet connection is no longer needed to deploy the parcel. For detailed information about parcels, see Overview of Parcels.

If your Cloudera Manager Server does not have Internet access, you can obtain the required parcel files and put them into a parcel repository. For more information, see Configuring a Local Parcel Repository.

After You Install

The following topics describe post-installation actions, such as deploying client configuration and some simple tests to validate the installation and confirm that everything is working as expected.

Managing Replication Peers for advanced TLS requirements on Cloudera on cloud clusters

Upgrading Cloudera Manager to version 7.13.2 SP1 on Cloudera on cloud environments (Data Lake, Cloudera Data Hub, or Cloudera Operational Database) preserves backward compatibility by default and does not enforce TLS certificate validation on existing Replication Peers.

However, enabling the CMF_FF_ENCRYPT_ALL_PORTS feature flag overrides this backward compatibility and enforces strict TLS certificate verification. To maintain Replication Peers and Replication Schedules active after enabling universal port encryption, follow the procedures in Configuring Replication Peers for advanced TLS controls on Cloudera on cloud clusters.

Related Information

Cloudera Base on premises Installation Guide

Deploying Clients

Client configuration files are generated automatically by Cloudera Manager based on the services you install.

Cloudera Manager deploys these configurations automatically at the end of the installation workflow. You can also download the client configuration files to deploy them manually.

If you modify the configuration of your cluster, you might need to redeploy the client configuration files. If a service's status is "Client configuration redeployment required," you need to redeploy those files.

Initializing Solr and creating HDFS home directory

After installing the Solr service and setting up Kerberos authentication, you need to initialize the service and create the HDFS home directory. Without this, the Solr service fails to start.

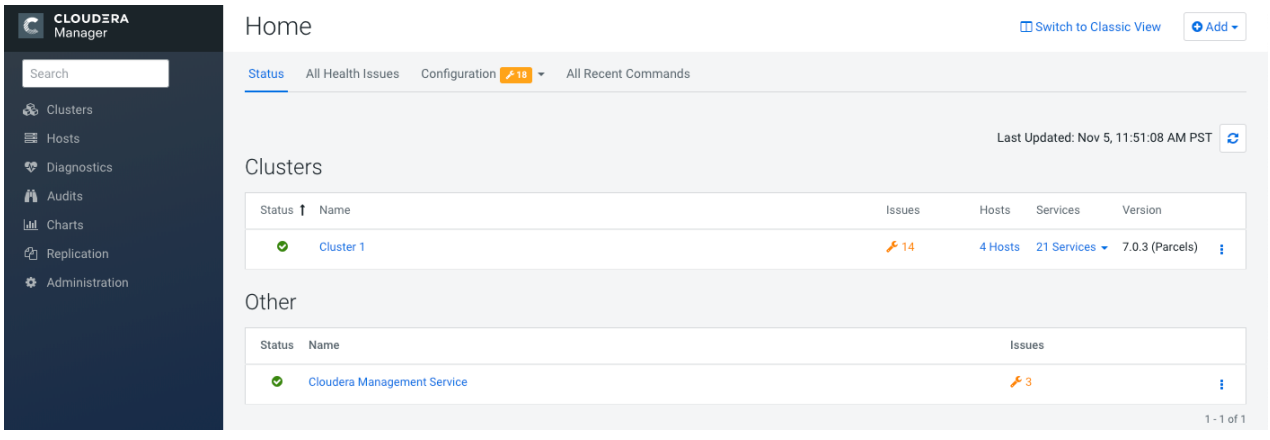
Procedure

1. In Cloudera Manager select the cluster where you want to initialize the Solr service.
2. Select the Solr service you want to initialize.
3. To initialize the Solr service select Actions Initialize Solr .
4. To create the HDFS Home directory, select Actions Create HDFS Home Dir .
5. Repeat these steps on all Solr services (Infra or workload) you need to initialize in the cluster.

Testing the Installation

Begin testing the installation from the **Home** page, where you can start by checking the health of the services.

To begin testing, start the Cloudera Manager Admin Console. Once you've logged in, the **Home** page should look something like this:



On the left side of the screen is a list of services currently running with their status information. All the services should be running with Good Health. You can click each service to view more detailed information about each

service. You can also test your installation by either checking each Host's heartbeats, running a MapReduce job, or interacting with the cluster with an existing Hue application.

Checking Host Heartbeats

One way to check whether all the Agents are running is to look at the time since their last heartbeat. You can do this by clicking the Hosts tab where you can see a list of all the hosts along with the value of their Last Heartbeat.

By default, every Agent must heartbeat successfully every 15 seconds. A recent value for the Last Heartbeat means that the Server and Agents are communicating successfully.

Run a MapReduce job

Run a PiEstimator job to manually verify that the Cloudera Base on premises installation was successful.

Running a MapReduce job using CLI

Run the Hadoop PiEstimator example job on your cluster to verify that your Cloudera Runtime installation completed successfully.

About this task

Note: If you have a secure cluster, use the kinit command line tool to authenticate to Kerberos.

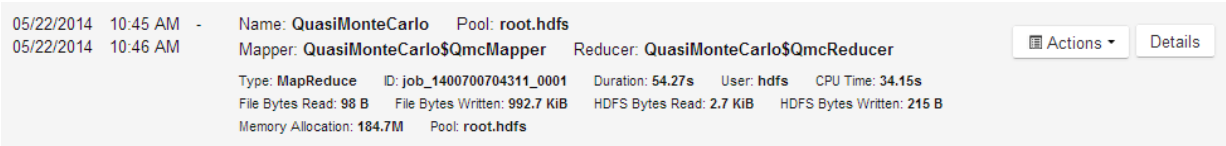
Procedure

1. Log into a host in the cluster.
2. Run the Hadoop PiEstimator example using the following command:

```
yarn jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi 10 100
```

3. In Cloudera Manager, navigate to Cluster *CLUSTERNAME* yarn Applications .
4. Check the results of the job.

   You will see an entry like the following:



Running a MapReduce job using the UI

Run the MapReduce Pi test in the Cloudera Manager UI to verify that the YARN service processes workloads without requiring a command-line access.

About this task

The Run Pi Test action provides a diagnostic tool in Cloudera Manager under the YARN service. It launches the standard Hadoop MapReduce Pi job using fixed parameters (10 maps and 100,000 samples) to conduct a lightweight functional validation of a YARN job.

Depending on your cluster environment, authentication behaves as follows:

- Cloudera Data Explorer (Hue) user available: Cloudera Manager automatically uses the Cloudera Data Explorer (Hue) user credentials to run the job.

- No Cloudera Data Explorer (Hue) user available: The UI prompts you to enter a username and password.
- Kerberos-enabled cluster: The provided credentials are used for Kerberos authentication, automatically generating a keytab from the password.
- Non-Kerberos cluster: The provided credentials are run within a local Unix user context.

Before you begin

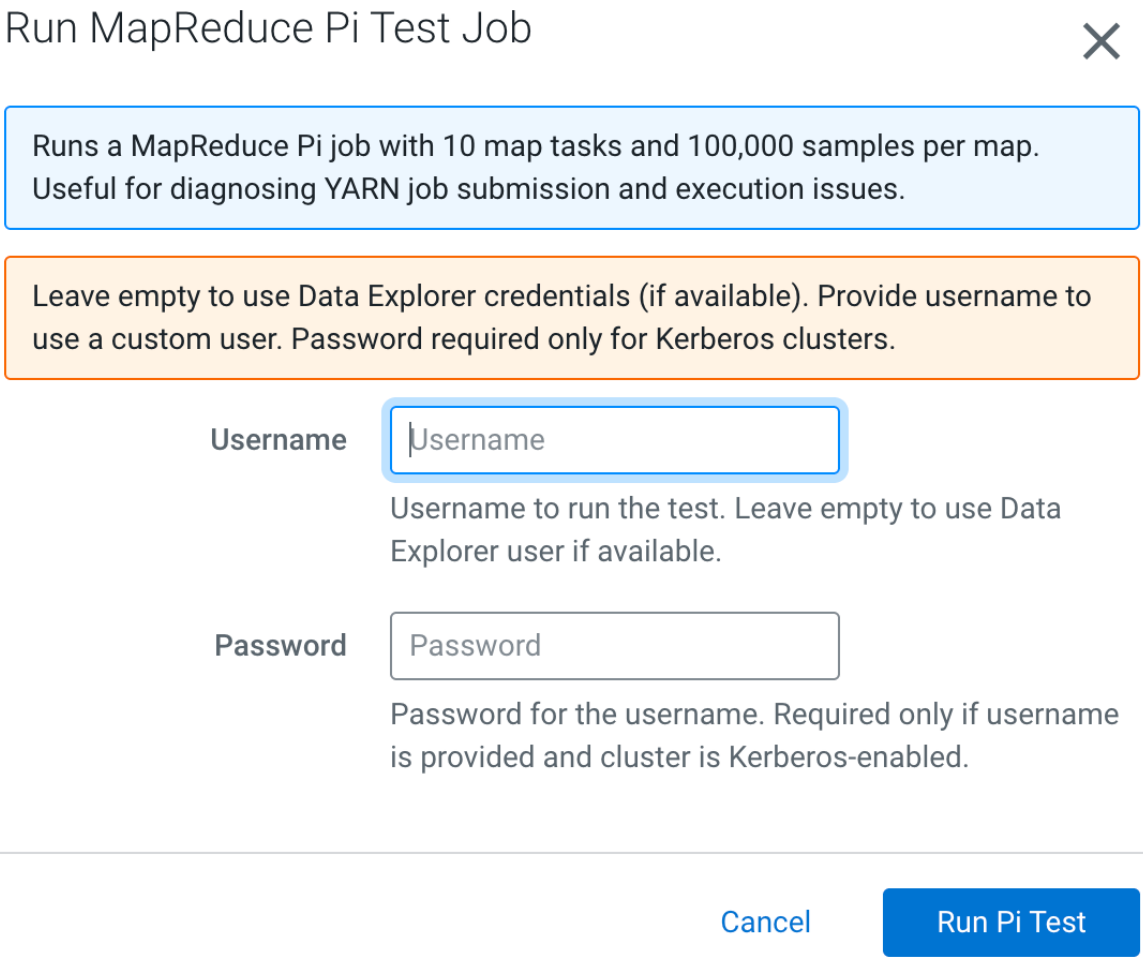
Ensure that the YARN service is running and healthy in Cloudera Manager. If your cluster does not have an active Cloudera Data Explorer (Hue) user, ensure you have valid user credentials available before proceeding.

Procedure

1. Log in to Cloudera Manager.
2. Navigate to the **YARN** service page.
3. Click the Actions drop-down menu.
4. Select Run MapReduce Pi Test Job.

   The **Run MapReduce Pi Test Job** dialog box is displayed.

   Figure 6: Run Mapreduce Pi Test Job dialog box



5. Enter your username and password.

Note: If a Cloudera Data Explorer (Hue) server role exists in the cluster, the job always runs as the Cloudera Data Explorer (Hue) user. Any username typed in the dialog is ignored.

6. Click Run Pi Test.

   The MapReduce job is now submitted. Cloudera Manager initiates the Pi test job with 10 maps and 100,000 samples.

What to do next

Once submitted, monitor the progress and status of the job through the standard YARN interfaces, such as the ResourceManager UI or the YARN Applications view.

Testing with Hue

You can test the cluster by running Hue.

About this task

Hue is a graphical user interface that allows you to interact with your clusters by running applications that let you browse HDFS and cloud object storage such as S3 and ABFS, manage a Hive metastore, and run Hive, Impala, and Search queries, and Oozie workflows.

Procedure

1. From Cloudera Manager, go to Clusters Hue service .
2. Click Web UI link and select the Hue web URL, which opens Hue in a new window.

   By default, Authentication Backend is set to AllowFirstUserDjangoBackend. This makes the first user who logs into Hue the Superuser and allows you to set the username and password, and create other users.

   You can change the Authentication Backend as per your requirements from Hue configurations in Cloudera Manager.
3. You can run a query or browse the database that you have set up for Hue.

   For more information, see the Hue documentation.

Deploying Atlas service

Post installation, you must plan to employ either the LDAP or Active Directory (AD) authentication mechanism to deploy the Atlas service in your production environment.

About this task

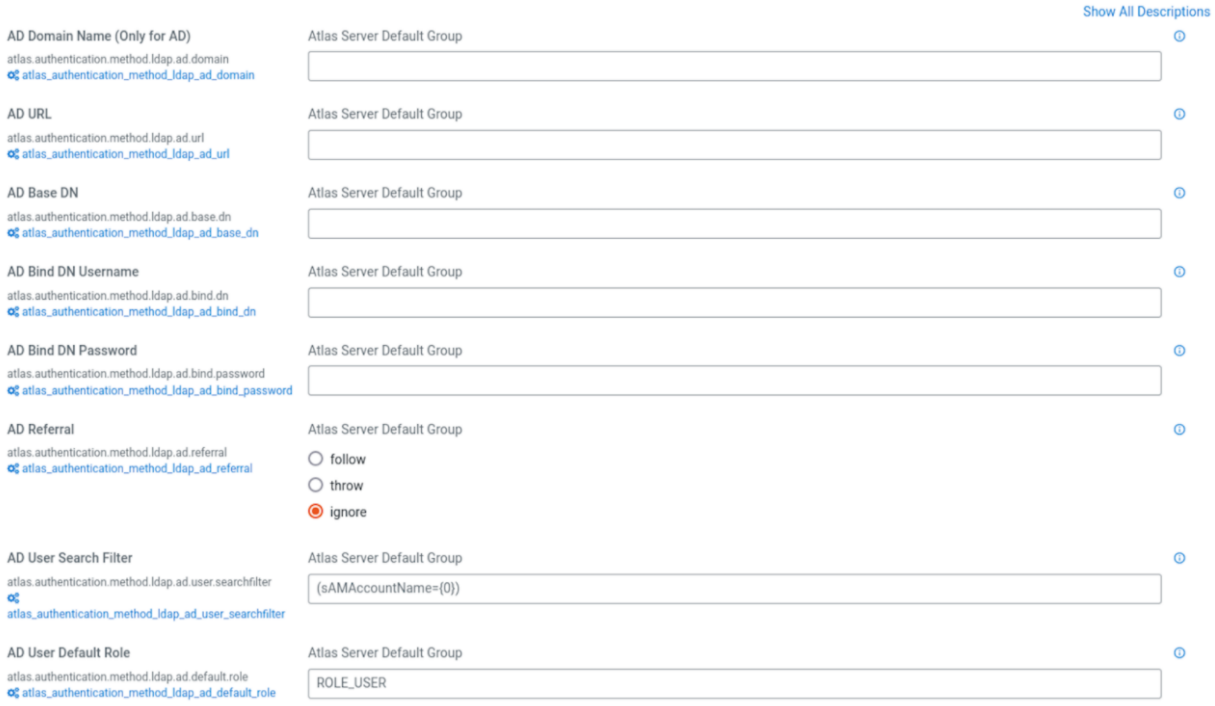
You can add LDAP or AD authentication configurations post-installation of Atlas.

Important: When you install Atlas using the Add Service method in the Cloudera Manager instance, you must make sure to clear the Enable File Authentication option.

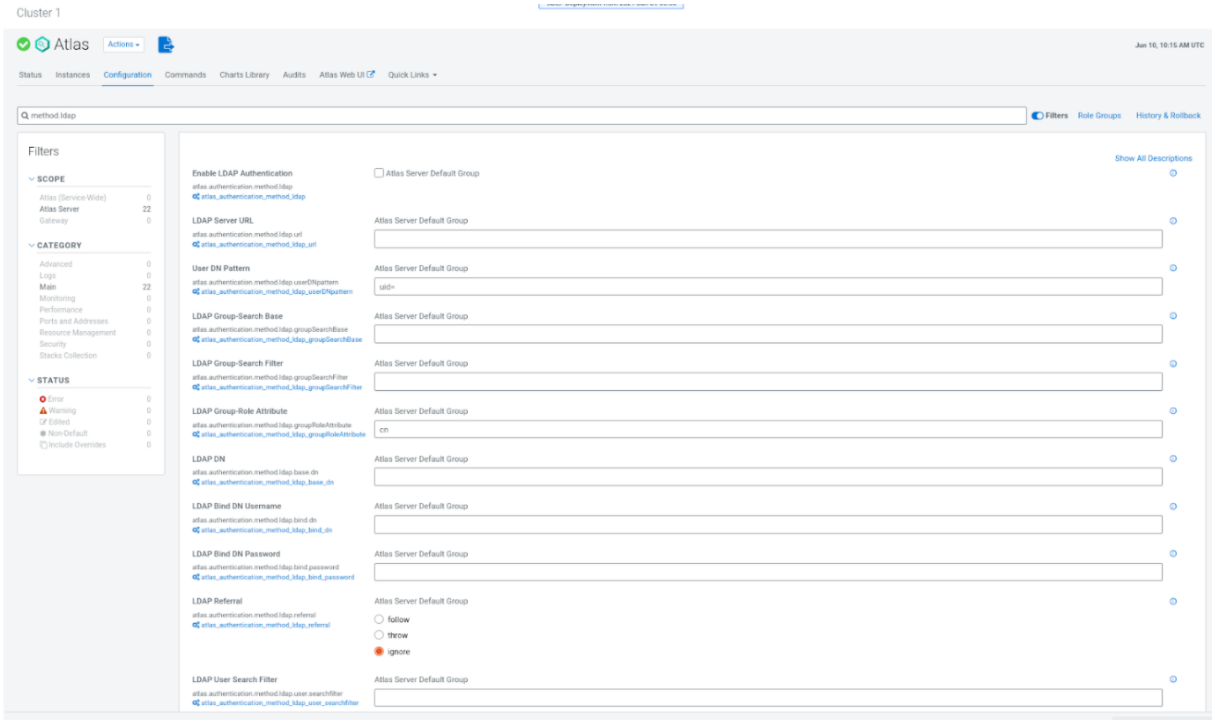
Procedure

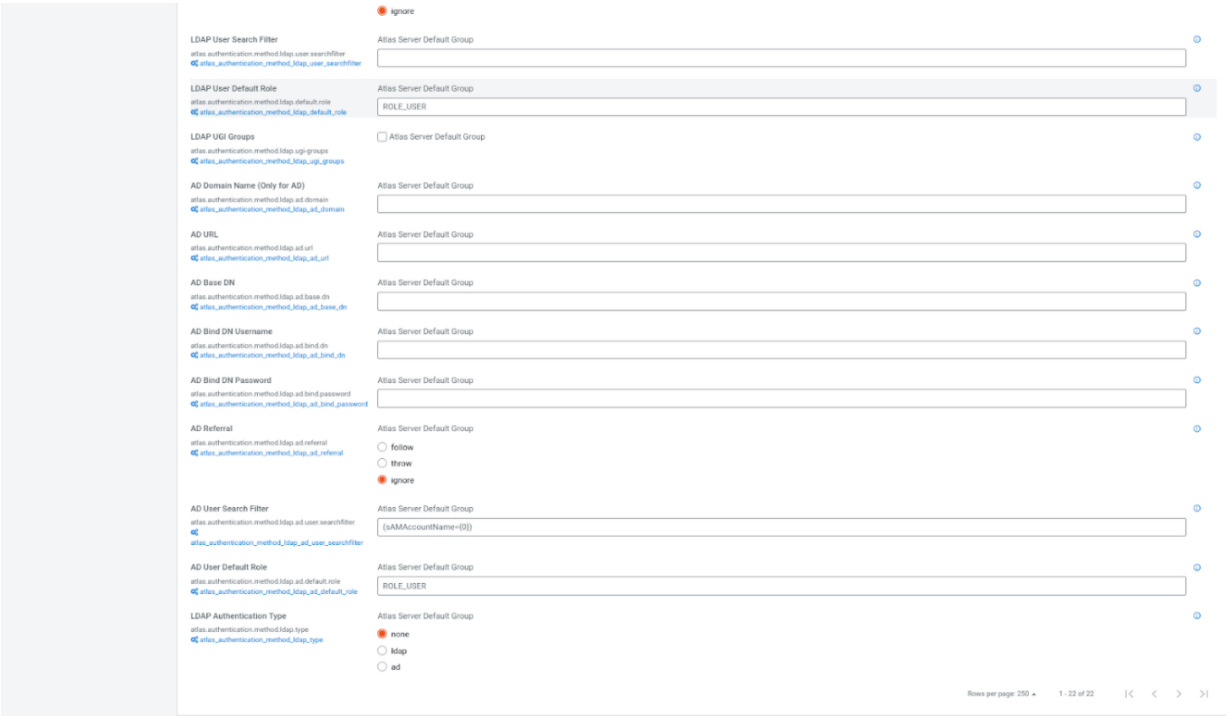
1. In your Cloudera Manager instance > Select Clusters > Configuration tab > On the search bar, use the key: atlas.authentication.method.
   The list of LDAP and AD configurations are displayed.
2. In the search, you are allowed to select the type of Atlas installation for LDAP Authentication type:
   - none
   - ldap
   - ad

3. Selecting “ad”, prompts you to use appropriate active directory values to complete the Atlas authentication type.

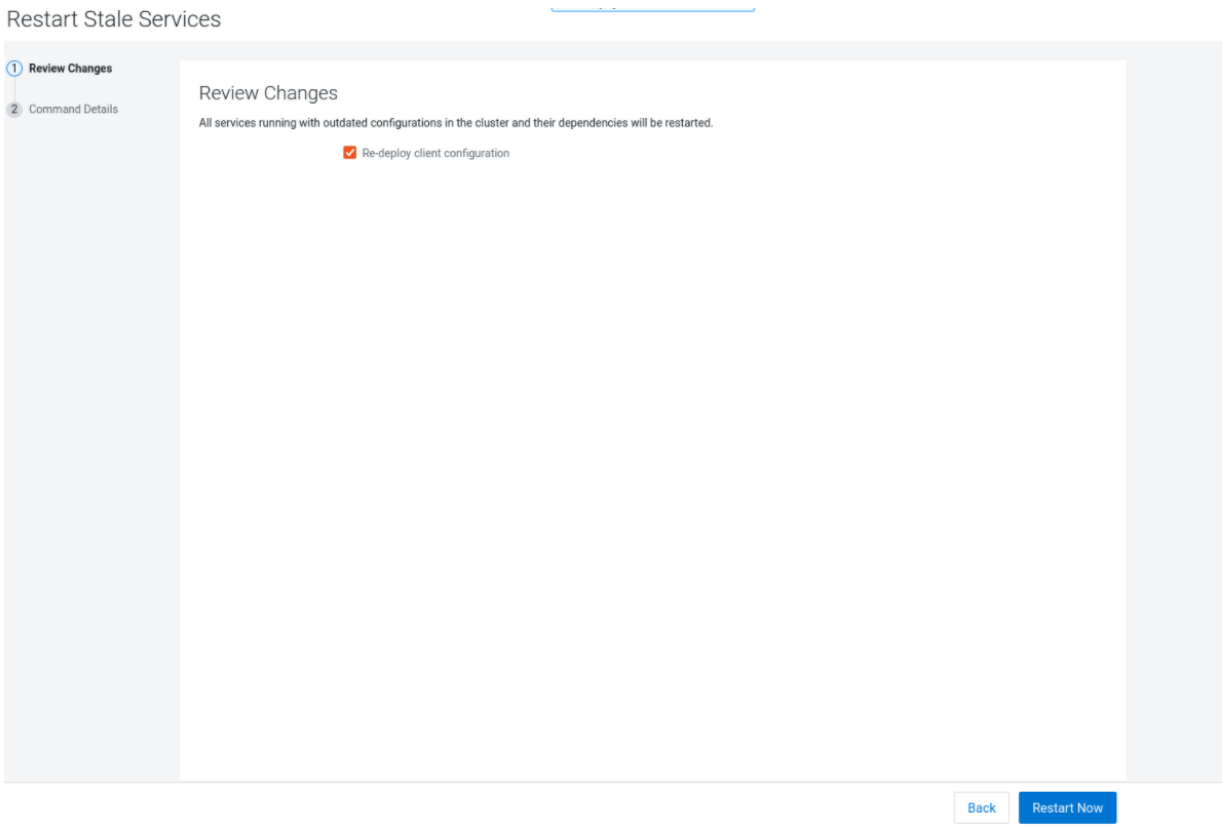


4. Selecting “ldap”, prompts you to use appropriate LDAP values to complete the Atlas authentication type.





5. You must restart the stale services.



Secure Your Cluster

After completing your Cloudera Enterprise installation and making sure that everything is working properly, secure your cluster by enabling authentication, authorization, auditing, and encryption.

For comprehensive instructions on securing your cluster, see the Security documentation.

Related Information

Security Overview

Installing the GPL Extras Parcel

GPL Extras contains functionality for compressing data using the LZO compression algorithm. To install the GPL Extras parcel:

Procedure

1. Add the appropriate repository to the Cloudera Manager list of parcel repositories. Specify the repository in Cloudera Manager as follows:

   ```
   https://USERNAME:PASSWORD@archive.cloudera.com/p/gplextras7/7.1.9.0/parcels/
   ```

   You can also download the parcel into a local parcel repository.
2. Download, distribute, and activate the parcel.
3. The LZO parcels require that the underlying operating system has the native LZO packages installed. If they are not installed on all cluster hosts, you can install them as follows:

   RHEL compatible:

   ```
   sudo yum install lzo
   ```

   Debian or Ubuntu:

   ```
   sudo apt-get install liblzo2-2
   ```

   SLES:

   ```
   sudo zypper install liblzo2-2
   ```
4. To configure LZO compression, see Configuring Services to Use LZO Compression.

Configuring HDFS properties to optimize log collection

Cloudera uses “out_webhdfs” Fluentd output plugin to write records into HDFS, in the form of log files, which are then used by different Data Services to generate diagnostic bundles. Over time, these log files can grow in size. To optimize the size of logs that are captured and stored on HDFS, you must update certain HDFS configurations in the hdfs-site.xml file using Cloudera Manager.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters HDFS service Configuration .
3. Select the Enable WebHDFS (dfs_webhdfs_enabled) option.
4. Add the following lines in the HDFS Service Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml field by clicking View as XML to enable append operations:

   ```
   <property>
     <name>dfs.support.append</name>
     <value>true</value>
   </property>
   ```

```
<property>
  <name>dfs.support.broken.append</name>
  <value>true</value>
</property>
```

5. Click Save Changes.
6. Restart the HDFS service.
7. Restart your Cloudera cluster.

Related Information

Fluentd documentation

Migrating from H2 to PostgreSQL database in YARN Queue Manager

If you prefer to migrate from an H2 database to a PostgreSQL database in YARN Queue Manager post-installation or upgrade, all your existing data in the H2 database will be transferred to PostgreSQL. The YARN Queue Manager will then establish connections and operate using the PostgreSQL database.

About this task

This migration supports only one-way transfer from H2 to PostgreSQL. Reverting to H2 after migration is not supported, and data migration from PostgreSQL back to H2 is not possible.

To continue using the embedded database, upgrade to Cloudera Base on premises 7.1.9 CHF 2 and disregard these instructions. However, if you prefer to use PostgreSQL as the external database, use Cloudera Base on premises 7.1.9 and follow these post-installation steps to migrate your database. See Known issues in 7.1.9 CHF 2 linked below for more information.

You must supply the details of a PostgreSQL database to migrate your old data to Cloudera Base on premises 7.1.9 or Cloudera Base on premises 7.1.9 CHF 1 as the only external database that Queue Manager supports in these versions is PostgreSQL. When updating to versions after 7.1.9 CHF 2, YARN Queue Manager will continue to use PostgreSQL if it was initially configured.

Important: To avoid any issues during the upgrade to version Cloudera Base on premises 7.1.9 SP1, ensure that PostgreSQL connection details are removed from the YARN database configuration if you prefer to continue using the H2 database.

Important: For users who only use the Queue Manager API directly, you need to use the UI to setup the database migration as this migration is triggered through the UI. You need to use Cloudera Manager and Queue Manager's UIs to setup the database configuration, and to load the UI to allow the data migration to complete. You must also refrain from direct API calls during an upgrade to Cloudera Base on premises 7.1.9 and refrain from direct API calls until the data migration is completed. Ensure that your data migration has succeeded before you make any API calls.

Before you begin

You must first create a database for Queue Manager and then configure the database to assign roles and permissions for the user (dbuser). This step is optional if you already have a database and are migrating from H2 to PostgreSQL.

The roles users need to assign are the following:

SUPERUSER CREATEDB CREATEROLE INHERIT LOGIN.

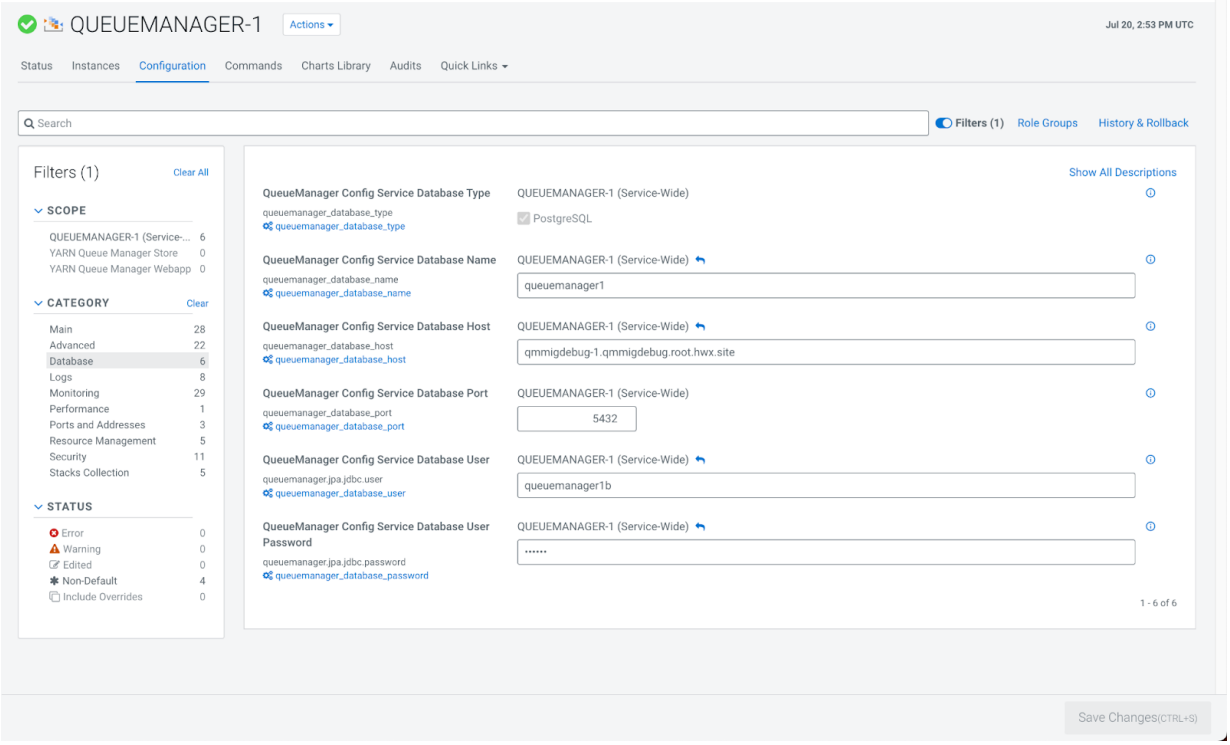
You can use the following sample query or similar query to assign roles. You must record these values for use in step 4.

```
CREATE ROLE qmadmin PASSWORD 'password' SUPERUSER CREATEDB CREATEROLE INHERI
T LOGIN;
CREATE DATABASE "configstore";
```

```
ALTER DATABASE "configstore" OWNER TO qmadmin;
```

Procedure

1. In your Cloudera Manager instance, navigate to Clusters Yarn Queue Manager .
2. Go to the Configuration tab.
3. In the left navigation menu under Category, click Database.
4. Enter the following fields that are required for the PostgreSQL database.
   - QueueManager Config Service Database Name
   - QueueManager Config Service Database Host
   - QueueManager Config Service Database Port
   - QueueManager Config Service Database User
   - QueueManager Config Service Database User Password



5. Click Save Changes.
6. In Queue Manager, navigate to Actions and click Restart

   The data migration to the PostgreSQL database may take a few minutes after Queue Manager is restarted.

Related Information

Known Issues in 7.1.9 CHF 2

Troubleshooting Installation Problems

This topic describes common installation issues and suggested solutions.

TLS Protocol Error with OpenJDK

If you are using an older version of OpenJDK 1.8 and have enabled SSL/TLS for the Cloudera Manager Admin Console, you may encounter a TLS protocol error when connecting to the Cloudera Manager Admin Console, stating

that there are no ciphers in common. This is because older versions of OpenJDK may not implement certain TLS ciphers, causing an inability to log into the Cloudera Manager Admin Console when TLS is enabled.

Workaround:

You can workaround this issue by doing one of the following:

- Upgrade OpenJDK to a supported version of OpenJDK that is higher than version 1.8.0_181.
- If it is not possible to upgrade OpenJDK, enable less secure TLS ciphers in Cloudera Manager. You can do this by opening the /etc/default/cloudera-scm-server in a text editor and adding the following line:

  ```
  export CMF_OVERRIDE_TLS_CIPHERS=<CIPHER_LIST>
  ```

  Where <*CIPHER_LIST*> is a list of TLS cipher suites separated by colons. For example:

  ```
  export CMF_OVERRIDE_TLS_CIPHERS="TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
  :TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256:TLS_ECDHE_ECDSA_WITH_AES_256_GCM_
  SHA384:TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384:TLS_DHE_RSA_WITH_AES_128_GC
  M_SHA256:TLS_DHE_RSA_WITH_AES_256_GCM_SHA384:TLS_ECDHE_ECDSA_WITH_AES_12
  8_CBC_SHA256:TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256:TLS_ECDHE_ECDSA_WITH_
  AES_128_CBC_SHA:TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384:TLS_ECDHE_RSA_WITH
  _AES_128_CBC_SHA:TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384:TLS_ECDHE_ECDSA
  _WITH_AES_256_CBC_SHA:TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA:TLS_DHE_RSA_WIT
  H_AES_128_CBC_SHA256:TLS_DHE_RSA_WITH_AES_128_CBC_SHA:TLS_DHE_RSA_WITH_A
  ES_256_CBC_SHA256:TLS_DHE_RSA_WITH_AES_256_CBC_SHA:TLS_ECDHE_ECDSA_WITH_
  3DES_EDE_CBC_SHA:TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA:TLS_EDH_RSA_WITH_3D
  ES_EDE_CBC_SHA:TLS_RSA_WITH_AES_128_GCM_SHA256:TLS_RSA_WITH_AES_256_GCM_
  SHA384:TLS_RSA_WITH_AES_128_CBC_SHA256:TLS_RSA_WITH_AES_256_CBC_SHA256:T
  LS_RSA_WITH_AES_128_CBC_SHA:TLS_RSA_WITH_AES_256_CBC_SHA:TLS_RSA_WITH_3D
  ES_EDE_CBC_SHA"
  ```

Cloudera Bug: OPSAPS-49578

Failed to start server reported by cloudera-manager-installer.bin

"Failed to start server" reported by cloudera-manager-installer.bin. /var/log/cloudera-scm-server/cloudera-scm-server .logcontains a message beginning Caused by: java.lang.ClassNotFoundException: com.mysql.jdbc.Driver...

Possible reason:

You might have SELinux enabled.

Possible solution:

Disable SELinux by running sudo setenforce 0 on the Cloudera Manager Server host. To disable it permanently, edit / etc/selinux/config.

Installation interrupted and installer does not restart

Possible reason:

You need to do some manual cleanup.

Possible solution:

See *Uninstalling Cloudera Manager and Managed Software*.

Cloudera Manager Server fails to start with MySQL

Cloudera Manager Server fails to start and the Server is configured to use a MySQL database to store information about service configuration.

Possible reason:

Tables might be configured with the ISAM engine. The Server does not start if its tables are configured with the MyISAM engine, and an error such as the following appears in the log file:

```
Tables ... have unsupported engine type ... .  InnoDB is required.
```

Possible solution:

Make sure that the InnoDB engine is configured, not the MyISAM engine. To check what engine your tables are using, run the following command from the MySQL shell: mysql> show table status;

For more information, see Install and Configure MySQL for Cloudera Software on page 102.

Agents fail to connect to Server

Agents fail to connect to Server. You get an Error 113 ('No route to host') in /var/log/cloudera-scm-agent/cloudera-scm-agent.log.

Possible reason:

You might have SELinux or iptables enabled.

Possible solution:

Check /var/log/cloudera-scm-server/cloudera-scm-server.log on the Server host and /var/log/cloudera-scm-agent/cloudera-scm-agent.log on the Agent hosts. Disable SELinux and iptables.

Cluster hosts do not appear

Some cluster hosts do not appear when you click Find Hosts in install or update wizard.

Possible reason:

You might have network connectivity problems.

Possible solution:

- Make sure all cluster hosts have SSH port 22 open.
- Check other common causes of loss of connectivity such as firewalls and interference from SELinux.

"Access denied" in install or update wizard

"Access denied" in install or update wizard during database configuration for Reports Manager.

Possible reason:

Hostname mapping or permissions are not set up correctly.

Possible solution:

- For hostname configuration, see *Configure Network Names*.
- For permissions, make sure the values you enter into the wizard match those you used when you configured the databases. The value you enter into the wizard as the database hostname must match the value you entered for the hostname (if any) when you configured the database.

  For example, if you had entered the following when you created the database

  ```
  grant all on activity_monitor.* TO 'amon_user'@'myhost1.myco.com' IDENTI
  FIED BY 'amon_password';
  ```

the value you enter here for the database hostname must be myhost1.myco.com. If you did not specify a host, or used a wildcard to allow access from any host, you can enter either the fully qualified domain name (FQDN), or localhost. For example, if you entered

```
grant all on activity_monitor.* TO 'amon_user'@'%' IDENTIFIED BY 'amon_p
assword';
```

the value you enter for the database hostname can be either the FQDN or localhost.

Databases fail to start.

Reports Manager or Service Monitor databases fail to start.

Possible reason:

MySQL binlog format problem.

Possible solution:

Set binlog_format=mixed in /etc/my.cnf. For more information, see this MySQL bug report. See also Install and Configure Databases on page 90.

Cloudera services fail to start

Possible reason:

Java might not be installed or might be installed at a custom location.

Possible solution:

See *Configuring a Custom Java Home Location* for more information on resolving this issue.

Create Hive Metastore Database Tables command fails

The Create Hive Metastore Database Tables command fails due to a problem with an escape string.

Possible reason:

PostgreSQL versions 9 and higher require special configuration for Hive because of a backward-incompatible change in the default value of the standard_conforming_strings property. Versions up to PostgreSQL 9.0 defaulted to off, but starting with version 9.0 the default is on.

Possible solution:

As the administrator user, use the following command to turn standard_conforming_strings off:

```
ALTER DATABASE <hive_db_name> SET standard_conforming_strings = off;
```

Oracle invalid identifier

If you are using an Oracle database and the Cloudera Navigator Analytics Audit Activity tab displays "No data available" and there is an Oracle error about "invalid identifier" with the query containing the reference to dbms_crypto in the log.

Possible reason:

You have not granted execute permission to sys.dbms_crypto.

Possible solution:

Run GRANT EXECUTE ON sys.dbms_crypto TO *NAV*;, where *NAV* is the user of the Navigator Audit Server database.

Related Information

Cloudera Base on premises Installation Guide

Uninstalling Cloudera Manager and Managed Software

Complete the following tasks to uninstall the Cloudera Manager Server, Agents, managed software, and databases.

Related Information

Cloudera Base on premises Installation Guide

Record User Data Paths

Record the location of the user data paths by checking the configuration in each service.

The user data paths listed in the topic *Remove User Data*, /var/lib/flume-ng /var/lib/hadoop* /var/lib/hue /var/lib/navigator /var/lib/oozie /var/lib/solr /var/lib/sqoop* /var/lib/zookeeper *DATA_DRIVE_PATH*/dfs *DATA_DRIVE_PATH*/mapred *DATA_DRIVE_PATH*/yarn, are the default settings. However, at some point they might have been reconfigured in Cloudera Manager. If you want to remove all user data from the cluster and have changed the paths, either when you installed Cloudera Runtime and managed services or at some later time, note the location of the paths by checking the configuration in each service.

Stop all Services

Stop all services for each cluster managed by Cloudera Manager.

Procedure

1. On the HomeStatus tab, click three dots to the right of the cluster name and select Stop.
2. Click Stop in the confirmation screen. The Command Details window shows the progress of stopping services. When All services successfully stopped appears, the task is complete and you can close the Command Details window.
3. On the Home Status tab, click the three dots to the right of the Cloudera Management Service entry and select Stop. The Command Details window shows the progress of stopping services.

Results

When All services successfully stopped appears, the task is complete and you can close the Command Details window.

Deactivate and Remove Parcels

If you installed using packages, skip this step and go to *Uninstall the Cloudera Manager Server*; you will remove packages in *Uninstall Cloudera Manager Agent and Managed Software*. If you installed using parcels remove them as follows:

Procedure

1. Click the parcel indicator  in the left-hand navigation bar.
2. In the Location selector on the left, select All Clusters.

3. For each activated parcel, select ActionsDeactivate. When this action has completed, the parcel button changes to Activate.
4. For each activated parcel, select ActionsRemove from Hosts. When this action has completed, the parcel button changes to Distribute.
5. For each activated parcel, select ActionsDelete. This removes the parcel from the local parcel repository.

What to do next

There might be multiple parcels that have been downloaded and distributed, but that are not active. If this is the case, you should also remove those parcels from any hosts onto which they have been distributed, and delete the parcels from the local repository.

Delete the Cluster

On the Home page, Click the drop-down list next to the cluster you want to delete and select Delete.

Uninstall the Cloudera Manager Server

The commands for uninstalling the Cloudera Manager Server depend on the method you used to install it. Refer to steps below that correspond to the method you used to install the Cloudera Manager Server.

Procedure

1. If you used the cloudera-manager-installer.bin file (the trial installer): Run the following command on the Cloudera Manager Server host:

   ```
   sudo /opt/cloudera/installer/uninstall-cloudera-manager.sh
   ```

2. If you did not use the cloudera-manager-installer.bin file: If you installed the Cloudera Manager Server using a different installation method such as Puppet, run the following commands on the Cloudera Manager Server host:

   a) Stop the Cloudera Manager Server and its database:

   ```
   sudo service cloudera-scm-server stop
   sudo service cloudera-scm-server-db stop
   ```

   b) Uninstall the Cloudera Manager Server and its database. This process described also removes the embedded PostgreSQL database software, if you installed that option. If you did not use the embedded PostgreSQL database, omit the cloudera-manager-server-db steps.

   RHEL

   ```
   sudo yum remove cloudera-manager-server
   sudo yum remove cloudera-manager-server-db-2
   ```

   SLES

   ```
   sudo zypper -n rm --force-resolution cloudera-manager-server
   sudo zypper -n rm --force-resolution cloudera-manager-server-db-2
   ```

   Ubuntu

   ```
   sudo apt-get remove cloudera-manager-server
   sudo apt-get remove cloudera-manager-server-db-2
   ```

Uninstall Cloudera Manager Agent and Managed Software

To uninstall Cloudera Manager Agent and managed software, stop the Cloudera Manager Agent on all hosts, remove the parcel installation, and run the clean command.

About this task

Do the following on all Agent hosts:

Procedure

1. Stop the Cloudera Manager Agent.

```
sudo systemctl stop cloudera-scm-supervisord.service
sudo systemctl stop cloudera-scm-agent
```

2. To uninstall managed software, run the following commands:

   RHEL

   ```
   sudo yum remove 'cloudera-manager-*'
   ```

   SLES

   ```
   sudo zypper remove 'cloudera-manager-*'
   ```

   Ubuntu

   ```
   sudo apt-get purge 'cloudera-manager-*'
   ```

3. Run the clean command:

   RHEL

   ```
   sudo yum clean all
   ```

   SLES

   ```
   sudo zypper clean
   ```

   Ubuntu

   ```
   sudo apt-get clean
   ```

Remove Cloudera Manager, User Data, and Databases

Permanently remove Cloudera Manager data, the Cloudera Manager lock file, and user data. Then stop and remove the databases.

Procedure

1. On all Agent hosts, stop any running Cloudera Manager and managed processes:

```
for u in cloudera-scm flume hadoop hdfs hbase hive httpfs hue impala llama
 mapred oozie solr spark sqoop sqoop2 yarn zookeeper; do sudo kill $(ps -u
 $u -o pid=); done
```

Note: This step should not be necessary if you stopped all the services and the Cloudera Manager Agent correctly.

2. If you are uninstalling on RHEL, run the following commands on all Agent hosts to permanently remove Cloudera Manager data. If you want to be able to access any of this data in the future, you must back it up before removing it. If you used an embedded PostgreSQL database, that data is stored in /var/lib/cloudera-scm-server-db.

```
sudo umount cm_processes
sudo rm -Rf /usr/share/cmf /var/lib/cloudera* /var/cache/yum/cloudera* /
var/log/cloudera* /var/run/cloudera*
```

3. On all Agent hosts, run this command to remove the Cloudera Manager lock file:

```
sudo rm /tmp/.scm_prepare_node.lock
```

4. This step permanently removes all user data. To preserve the data, copy it to another cluster using the distcp command before starting the uninstall process.
   a) On all Agent hosts, run the following commands:

```
sudo rm -Rf /var/lib/flume-ng /var/lib/hadoop* /var/lib/hue /var/
lib/navigator /var/lib/oozie /var/lib/solr /var/lib/sqoop* /var/lib/
zookeeper
```

   b) Run the following command on each data drive on all Agent hosts (adjust the paths for the data drives on each host):

```
sudo rm -Rf DATA_DRIVE_PATH/dfs DATA_DRIVE_PATH/mapred DATA_DRIVE_PATH/
yarn
```

5. Stop and remove the databases. If you chose to store Cloudera Manager or user data in an external database, see the database vendor documentation for details on how to remove the databases.

Uninstalling a Cloudera Runtime Component From a Single Host

The following procedure removes Cloudera Runtime software components from a single host that is managed by Cloudera Manager.

Procedure

1. In the Cloudera Manager Admin Console, select Hosts All Hosts .
   A list of hosts in the cluster displays.
2. Select the host where you want to uninstall Cloudera Runtime software.
3. Click the Actions for Selected button and select Remove From Cluster.
   Cloudera Manager removes the roles and host from the cluster.
4. Optionally, manually delete the krb5.conf file used by Cloudera Manager.