

In-Place Upgrade of Cloudera Base on premises to higher versions

Date published: 2019-11-22

Date modified: 2024-12-10



Legal Notice

© Cloudera Inc. 2025. 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

Upgrading Cloudera Base on premises to a higher version.....	8
---	----------

About using this online Upgrade Guide.....	8
---	----------

How much time should I plan for to complete my upgrade?.....	9
---	----------

Upgrading the Operating System to a new Major Version.....	10
---	-----------

Step 1: Getting Started with Operating System Upgrades.....	10
Prerequisites.....	10
Step 2: Backing Up Host Files Before Upgrading the Operating System.....	11
Backing Up.....	11
Backing up Cloudera Manager databases.....	11
Step 3: Before you upgrade the Operating System to a new Major Version.....	13
Stop Running Clusters.....	13
Stop Cloudera Manager Server.....	13
Remove Packages & Parcels.....	13
Upgrade the Operating System.....	14
Step 4: After you upgrade the Operating System to a new Major Version.....	14
Establish Access to the Software.....	14
Reinstall Cloudera Manager Server, Daemon & Agent Packages.....	16
Start Cloudera Manager Server & local Agent.....	17
Install and start Agent on remaining hosts.....	17
Start Cloudera Management Service and Clusters.....	18

Upgrading the Operating System to a new Minor Version.....	19
---	-----------

Step 1: Getting Started with Operating System Upgrades.....	19
Prerequisites.....	19
Step 2: Backing Up Host Files Before Upgrading the Operating System.....	19
Backing Up.....	19
Backing up Cloudera Manager databases.....	20
Step 3: Before you upgrade the Operating System to a new Minor Version.....	21
Decommission and Stop Running Roles.....	21
Stop Cloudera Manager Agent.....	22
Stop Cloudera Manager Server & Agent.....	22
Stop Databases.....	23
Remove Packages & Parcels.....	23
Upgrade the Operating System.....	24
Step 4: After you upgrade the Operating System to a new Minor Version.....	24
Establish Access to the Software.....	25
Reinstall Cloudera Manager Server, Daemon & Agent Packages.....	26
Start Databases.....	27
Start Server & Agent.....	27
Start Roles.....	27

Upgrading the JDK.....	28
Manually Installing Oracle JDK 1.8.....	28
OpenJDK.....	30
Manually Installing OpenJDK.....	30
Manually Migrating to OpenJDK.....	32
Using AES-256 Encryption.....	36
Configuring a Custom Java Home Location.....	37
Tuning JVM Garbage Collection.....	38
 Upgrading Python.....	 41
 Upgrading Cloudera Manager 7.....	 44
Step 1: Getting Started Upgrading Cloudera Manager 7.....	45
Collect Information.....	46
Preparing to Upgrade Cloudera Manager.....	46
Step 2: Backing Up Cloudera Manager 7.....	47
Collect Information for Backing Up Cloudera Manager.....	48
Back Up Cloudera Manager Agent.....	48
Back Up the Cloudera Management Service.....	49
Stop Cloudera Manager Server & Cloudera Management Service.....	50
Back Up the Cloudera Manager Databases.....	50
Back Up Cloudera Manager Server.....	51
(Optional) Start Cloudera Manager Server & Cloudera Management Service.....	52
Step 3: Upgrading the Cloudera Manager Server.....	53
Establish Access to the Software.....	54
Install Java (JDK).....	56
Upgrade the Cloudera Manager Server.....	58
Step 4: Upgrading the Cloudera Manager Agents.....	62
Upgrade the Cloudera Manager Agents (Cloudera Manager 7.0.3 and higher).....	63
Step 5: After You Upgrade Cloudera Manager.....	67
Perform Post-Upgrade Steps.....	68
Upgrading Cloudera Navigator Key Trustee Server 7.1.x.....	71
Upgrading Cloudera Navigator Key HSM.....	72
Upgrading Cloudera Navigator Encrypt.....	75
Troubleshooting a Cloudera Manager Upgrade.....	79
The Cloudera Manager Server fails to start after upgrade.....	79
Re-Running the Cloudera Manager Upgrade Wizard.....	79
Reverting a Failed Cloudera Manager Upgrade.....	80
Ensure Cloudera Manager Server and Agent are stopped.....	80
Restore the Cloudera Manager Database (if necessary).....	80
Establish Access to the Software.....	81
Downgrade the Cloudera Manager Packages.....	83
Validate the configuration of the Cloudera Manager Server or Agent.....	84
Start Cloudera Manager Again.....	84
 Installing dependencies for Hue.....	 85
Installing the psycopg2 Python package for PostgreSQL database.....	85
Installing MySQL client for MySQL databases.....	87
Installing MySQL client for MariaDB databases.....	89

Getting started with Zero Downtime Upgrade (ZDU).....91

Software Requirements.....	92
Configuring components before starting ZDU.....	92
Check and enable High Availability service for your components.....	97
ZDU Component Support.....	99
Service components limitations.....	100
Glossary of terminologies.....	103

Upgrading a Cloudera Runtime Cluster..... 104

Step 1: Getting Started Upgrading a Cluster.....	105
Collect Information.....	107
Preparing to Upgrade a Cluster.....	107
Step 2: Review Notes and Warnings.....	114
Step 3: Backing Up the Cluster.....	119
Back Up Databases.....	119
Back Up ZooKeeper.....	121
Back up Solr.....	121
Back Up HDFS.....	121
Back Up Key Trustee Server and Clients.....	122
Back Up HSM KMS.....	123
Back Up Navigator Encrypt.....	123
Back Up HBase.....	123
Back Up YARN Queue Manager.....	123
Back up Atlas.....	124
Back Up Sqoop 2.....	125
Back Up Hue.....	125
Back Up Impala.....	125
Prerequisites for external database.....	126
Step 4: Back Up Cloudera Manager.....	126
Collect Information for Backing Up Cloudera Manager.....	126
Back Up Cloudera Manager Agent.....	127
Back Up the Cloudera Management Service.....	127
Stop Cloudera Manager Server & Cloudera Management Service.....	128
Back Up the Cloudera Manager Databases.....	128
Back Up Cloudera Manager Server.....	129
(Optional) Start Cloudera Manager Server & Cloudera Management Service.....	130
Step 5: Access Parcels.....	131
Step 6: Enter Maintenance Mode.....	131
Step 7: Run the Upgrade Cluster Wizard.....	131
Step 8: Finalize the HDFS or Ozone Upgrade.....	133
Step 9: Complete Post-Upgrade steps for upgrades to Cloudera Base on premises.....	134
Step 10: Exit Maintenance Mode.....	143

ZDU known issues..... 143**Applying a Service Pack..... 148****How to Install Cumulative Hotfix (CHF)..... 149**

Manual upgrade to Cloudera Base on premises..... 151

Upgrade Ranger database and apply patches.....	151
Setup Ranger Admin Component.....	151
Start Ranger.....	152
Set up the Ranger Plugin service.....	152
Start Kudu.....	152
Start ZooKeeper.....	152
Upgrade HDFS Metadata.....	152
Start HDFS.....	152
Start YARN QueueManager.....	152
Import Sentry Policies to Ranger.....	152
Start HBASE.....	152
Start YARN QueueManager.....	153
Clean NodeManager Recovery Directory (YARN).....	153
Reset ACLs on YARN Zookeeper nodes.....	153
Install YARN MapReduce Framework Jars.....	153
Start YARN.....	153
Deploy Client Configuration Files.....	153
Reinitialize Solr State for Upgrade.....	153
Bootstrap Solr Configuration.....	154
Start Solr.....	154
Bootstrap Solr Collections.....	154
Create HDFS Home directory.....	154
Create Ranger Plugin Audit Directory.....	154
Start infrastructure Solr.....	154
Start HBASE.....	154
Start KAFKA.....	154
Create Ranger Kafka Plugin Audit Directory.....	154
Create HBase tables for Atlas.....	155
Start Atlas.....	155
Create Ranger Atlas Plugin Audit Directory.....	155
Start Phoenix.....	155
Install MapReduce Framework Jars.....	155
Start YARN.....	155
Deploy Client Configuration Files.....	155
Upgrade the Hive Metastore Database.....	155
Start Hive.....	156
Create Hive Warehouse Directory.....	156
Create Hive Warehouse External Directory.....	156
Create Hive Sys database.....	156
Create Ranger Plugin Audit Directory.....	156
Start Impala.....	156
Create Ranger Plugin Audit Directory.....	156
Create Spark Driver Log Dir.....	157
Start Spark.....	157
Start Livy.....	157
Upgrade Oozie Database Schema.....	157
Updating column types in Oozie Database.....	157
Upgrade Oozie SharedLib.....	158
Upload Tez tar file to HDFS.....	159
Migrate Hive tables for Cloudera Base on premises upgrade.....	159
Create Ranger Plugin Audit Directory.....	159
Start Hive on Tez.....	159
Start Hue.....	159

Start the Remaining Cluster Services.....	159
Validate the Hive Metastore Database Schema.....	159
Test the Cluster and Finalize HDFS Metadata.....	160
Clear the Upgrade State Table.....	160
Troubleshooting Upgrades.....	160
Cloudera Runtime upgrade failure.....	160
"Access denied" in install or update wizard.....	161
Possible Reasons.....	161
Possible Solutions.....	161
The Hive on Tez service and the RangerAdmin role of the Ranger service kept going down.....	161
Possible Reasons.....	161
Possible Solutions.....	161
Cluster hosts do not appear.....	161
Possible Reasons.....	162
Possible Solutions.....	162
Cannot start services after upgrade.....	162
Possible Reasons.....	162
Possible Solutions.....	162
HDFS DataNodes fail to start.....	162
Possible Reasons.....	162
Possible Solutions.....	162
Cloudera services fail to start.....	162
Possible Reasons.....	163
Possible Solutions.....	163
Host Inspector Fails.....	163
Configuring a Local Package Repository.....	163
Creating a Permanent Internal Repository.....	164
Setting Up a Web server.....	164
Downloading and publishing the package repository for Cloudera Manager.....	164
Creating a Temporary Internal Repository.....	165
Configuring Hosts to Use the Internal Repository.....	166
Configuring a Local Parcel Repository.....	166
Using an Internally Hosted Remote Parcel Repository.....	166
Setting Up a Web Server.....	166
Downloading and Publishing the Parcel Repository.....	168
Configuring Cloudera Manager to Use an Internal Remote Parcel Repository.....	168
Using a Local Parcel Repository.....	169
Applications Upgrade.....	169
Rollback and Downgrade Cloudera Base on premises.....	170
Downgrade or Rollback from Cloudera Base on premises 7.3.1.....	170
Downgrade from Cloudera Base on premises 7.3.1.....	171
Rollback from Cloudera Base on premises 7.3.1.....	177
Downgrade or Rollback from CDP Private Cloud Base 7.1.9.....	189
Downgrade from CDP Private Cloud Base 7.1.9.....	190
Rollback from CDP Private Cloud Base 7.1.9.....	197
Procedure to Rollback between SP or CHF releases of the same Cloudera Base on premises release line.....	210

Upgrading Cloudera Base on premises to a higher version

**Important:**

Before performing an upgrade of Cloudera Manager or the Cloudera Runtime, creating a backup of all the metadata databases is important. This includes the Cloudera Manager database, and the various Cloudera Runtime component databases such as Hive Metadata Server, Ranger Admin, Ranger KMS, Schema Registry, etc. The backups are necessary if there is a reason to rollback to the prior version.

An upgrade from Cloudera Base on premises to a higher version of Cloudera Base on premises has the following high-level workflow:

1. Prepare to upgrade:
 - a. Review the [Supported in-place upgrade paths](#) for your upgrade.
 - b. Review the [Requirements and Supported Versions](#) for your upgrade
 - c. Review the [Release Notes](#) for the version of Cloudera Base on premises you are upgrading to.
 - d. Review the [CDP Upgrade/Migrate Troubleshooting Articles](#). These Cloudera Knowledge Base articles describe common issues encountered by Cloudera customers during upgrades and migrations. (Cloudera login required.)
 - e. Gather information on your deployment. See [Step 1: Getting Started Upgrading Cloudera Manager 7](#) on page 45 and [Step 1: Getting Started Upgrading a Cluster](#).
 - f. If you have Cloudera Flow Management installed, verify its compatibility with your target Cloudera Base on premises version before starting the upgrade process. For more information, see the Cloudera Flow Management [Upgrade and migration paths](#) documentation.
 - g. Plan how and when to begin your upgrade.
2. If necessary, [Upgrade the JDK](#).
3. Review [Upgrading Python](#) for your upgrade.
4. If necessary, [Upgrade the Operating System](#).
5. Upgrade Cloudera Manager to version 7.1.1 or higher. After upgrading to Cloudera Manager 7.1.1 or higher, Cloudera Manager can manage upgrading your cluster to a higher version. See [Upgrading Cloudera Manager 7](#) on page 44.
6. Use Cloudera Manager to upgrade from lower version of Cloudera Runtime to a higher version of Cloudera Runtime. See [Upgrading a Cluster](#).




Note: You can retain older parcels for 30 days in case it's necessary to revert an upgrade, and then delete the older parcels.

About using this online Upgrade Guide

How to fill in forms to customize the documentation for your upgrade.

This online version of the Cloudera Upgrade Guide allows you to create a customized version of the guide on many pages that only includes the steps required for your upgrade. Use the My Environment form at the top of pages in this guide to select the Cloudera Manager, CDH or Cloudera Runtime version for your upgrade as well as the operating system version, database type, and other information about your upgrade. After making these selections, the pages in the guide will only include the required steps for your upgrade. The information you enter is retained on each page in the guide.


Figure 1: My Environment Form Example

My Environment 

Fill in the following form to create a customized set of instructions for your environment.

Current Cloudera Manager	Version	<input type="text" value="Choose..."/>
Install Method		<input type="text" value="Choose..."/>
Operating System		<input type="text" value="Choose..."/>
HDFS High Availability		<input type="text" value="Choose..."/>
Using Cloudera Navigator		<input type="text" value="Choose..."/>
Current Cluster Version		<input type="text" value="Choose..."/>
New Cluster Version		<input type="text" value="Choose..."/>

Fill out the form above before you proceed.

To share this environment with others, click the  icon next to My Environment to copy a link specific for this environment to the clipboard.

When a form similar to this appears at the top of pages in the Upgrade Guide, select your Cloudera Manager version, cluster version, and other information about your upgrade using the drop-down lists.



Note: The HDP upgrade procedures do not include a My Environment form at the top of the page.

How much time should I plan for to complete my upgrade?

An in-place upgrade can take a variable amount of time to complete. Learn about how to plan for and shorten the amount of time required for your upgrade.

The amount of time required for an in-place upgrade depends on many factors, including:

- The number of hosts in your clusters.
- The mix of services you have deployed in your clusters.
- The amount of data stored in your clusters.

Generally, an upgrade can be completed in 24-48 hours. Upgrades from HDP to Cloudera may take somewhat longer due to the Ambari to Cloudera Manager conversion process (AM2CM).

The following table provides some additional information to help you plan for your upgrade.

Table 1: Upgrade Time Planning

Component/Process	Notes
Cloudera Runtime Parcel	The Cloudera Runtime parcel must be distributed to all hosts before upgrading the hosts. Downloading the parcel directly from archive.cloudera.com over the internet may add additional time. You can download the parcels and serve them from a local web server to decrease this time. In addition, after downloading the parcels to a local repository, you can distribute them in advance of launching the upgrade wizard to save additional time.
Cloudera Manager	You must upgrade Cloudera Manager before upgrading your clusters. Cloudera Manager can continue to manage older versions of Cloudera Runtime and CDH until the upgrade.
Cluster cold start	The cluster will need to be restarted at least once during an in-place upgrade. The amount of time required for a restart depends on how many files and blocks are stored in the cluster and the number of hosts in the cluster.
Hive	The Hive strict managed migration process can take a significant amount of time. See for more information about mitigating that impact. See Understanding the Hive upgrade (CDH)
HBase checks	While Running HBase checks does not take significant time, remediating any issues can take significant time. To save time during the upgrade, you can plan to do this before running the Upgrade Wizard.
Solr export/backup	This process depends on how much data has to be imported after the upgrade.

Upgrading the Operating System to a new Major Version

This topic describes the additional steps needed to upgrade the operating system of a host managed by Cloudera Manager to a higher version for major releases

Upgrading the operating system to a different major release is called a major release upgrade. For example, upgrading from Redhat 6.8 to 7.4. This is a much more complex procedure and some operating systems do not support these upgrades without a complete OS reinstall. Therefore, this topic does not cover the procedures for upgrading specific operating systems.

In general, a major OS upgrade process requires shutting down all cluster components and Cloudera Manager. All Cloudera packages and parcels must be removed and reinstalled for the new OS.

This topic primarily describes all the additional steps such as backing up essential files and removing and reinstalling all the necessary Cloudera Enterprise packages and parcels.



Note: You must determine whether to upgrade the operating system or Cloudera Manager first:

For example, consider the following upgrade from Cloudera Manager 5.13.3 to Cloudera Manager 7.1.4 and an operating system upgrade from RHEL 7.6 to RHEL 7.8

- Cloudera Manager 5.13.3 supports only RHEL 7.6
- Cloudera Manager 7.1.4 supports RHEL 7.6 and RHEL 7.8

In this case you must upgrade Cloudera Manager first, otherwise Cloudera Manager version 5.13.3 would be deployed on an unsupported operating system (RHEL 7.6) and may fail.

When upgrading the OS to RHEL 8, you need to update the Java version to OpenJDK 8. You also need to ensure that the client has the same Java version on Kafka brokers, Zookeeper nodes, AD, and the client side.

Step 1: Getting Started with Operating System Upgrades

Prerequisites

- Ensure that the versions of Cloudera Manager and CDH or Cloudera Runtime support your new operating system.
 - See [Operating System Requirements](#) for Cloudera Base on premises.

If you are using unsupported versions, see [Upgrade Cloudera Manager](#) or [Upgrading a Cloudera Runtime Cluster](#).

- Ensure that the host has access to the Cloudera Manager server, daemon and agent packages that are supported for the new operating system, either by having access to <https://archive.cloudera.com> or a [local package repository](#).
- Ensure that the Cloudera Manager server has access to the parcels that are using supported for the new Operating System, either by having access to <https://archive.cloudera.com> or a [local parcel repository](#).
- If you have a patched package or parcel installed, make sure you have the same package or parcel for the new Operating System and it has been made available to Cloudera Manager.
- Understand that performing a major release upgrade for the operating system in-place may be quite tricky and risky.

Step 2: Backing Up Host Files Before Upgrading the Operating System

This topic describes how to backup important files on your host before upgrading the operating system.

Backing Up

1. Create a top-level backup directory.

```
export CM_BACKUP_DIR=`date +%F`-CM
echo $CM_BACKUP_DIR
mkdir -p $CM_BACKUP_DIR
```

2. Back up the Agent directory and the runtime state.

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-agent.tar --exclude=*.sock /etc/cloudera-scm-agent /etc/default/cloudera-scm-agent /var/run/cloudera-scm-agent /var/lib/cloudera-scm-agent
```

3. Back up the Server directories:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-scm-server /etc/default/cloudera-scm-server
```

4. Back up the Cloudera Manager databases. See [Backing up Cloudera Manager databases](#) on page 11



Note: Backup is recommended but not always required for a minor release upgrade.

Backing up Cloudera Manager databases

Cloudera recommends that you schedule regular backups of the databases that Cloudera Manager uses to store configuration, monitoring, and reporting data and for managed services that require a database:

Backing Up PostgreSQL Databases

To back up a PostgreSQL database, use the same procedure whether the database is embedded or external:

1. Log in to the host where the Cloudera Manager Server is installed.
2. Get the name, user, and password properties for the Cloudera Manager database from `/etc/cloudera-scm-server/db.properties`:

```
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
```

```
com.cloudera.cmf.db.password=NnYfWIjlbk
```

3. Run the following command as root using the parameters from the preceding step:

```
# pg_dump -h HOSTNAME -p 7432 -U scm > /tmp/scm_server_db_backup.$(date +%Y%m%d)
```

4. Enter the password from the com.cloudera.cmf.db.password property in step 2.
5. To back up a database created for one of the roles on the local host as the *ROLEUSER* user:

```
# pg_dump -h HOSTNAME -p 7432 -U ROLEUSER > /tmp/ROLEDB
```

6. Enter the password specified when the database was created.

Backing Up MariaDB Databases

To back up the MariaDB database, run the mysqldump command on the MariaDB host, as follows:

```
mysqldump -hHOSTNAME -uUSERNAME -pPASSWORD DATABASE > /tmp/DATABASE-BACKUP.sql
```

For example, to back up the Activity Monitor database amon created in [Creating Databases for Cloudera Software](#), on the local host as the root user, with the password amon_password:

```
mysqldump -pamon_password amon > /tmp/amon-backup.sql
```

To back up the sample Activity Monitor database amon on remote host myhost.example.com as the root user, with the password amon_password:

```
mysqldump -hmyhost.example.com -uroot -pamon_password amon > /tmp/amon-backup.sql
```

Backing Up MySQL Databases

To back up the MySQL database, run the mysqldump command on the MySQL host, as follows:

```
mysqldump -hHOSTNAME -uUSERNAME -pPASSWORD DATABASE > /tmp/DATABASE-BACKUP.sql
```

For example, to back up the Activity Monitor database amon created in [Creating Databases for Cloudera Software](#), on the local host as the root user, with the password amon_password:

```
mysqldump -pamon_password amon > /tmp/amon-backup.sql
```

To back up the sample Activity Monitor database amon on remote host myhost.example.com as the root user, with the password amon_password:

```
mysqldump -hmyhost.example.com -uroot -pamon_password amon > /tmp/amon-backup.sql
```

You can back up all database using the following command:

```
mysqldump --all-databases -pPASSWORD > /tmp/all1/all.sql
```



Backing Up Oracle Databases

For Oracle, work with your database administrator to ensure databases are properly backed up.

Step 3: Before you upgrade the Operating System to a new Major Version

This topic describes steps you must perform before upgrading the operating system on all hosts managed by Cloudera Manager.

Stop Running Clusters

1. Log in to the Cloudera Manager Admin Console.
2. On the Home Status tab, click the  options menu to the right of each Cluster and select Stop to stop all the services in the cluster.
3. Do one of the following to stop the Cloudera Management Service:
 - a. Select Clusters Cloudera Management Service .
 - b. Select Actions Stop .
 - On the Home Status tab, click the  options menu to the right of Cloudera Management Service and select Stop.

Stop Cloudera Manager Server

1. On the Cloudera Manager Server host, stop the Cloudera Manager Service by running the following command:

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

Remove Packages & Parcels

Packages and parcels for the older operating system will not be able to start on the new operating system. Perform the following steps on every host in the cluster:

1. Run the following command to remove old packages from the host:

RHEL / CentOS

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

```
sudo rm /etc/yum.repos.d/cloudera*manager.repo*
```

SLES

```
sudo zypper remove cloudera-manager-server cloudera-manager-daem  
ons cloudera-manager-agent
```

```
sudo rm /etc/zypp/repos.d/cloudera*manager.repo*
```

Ubuntu

```
sudo apt-get remove cloudera-manager-server cloudera-manager-dae  
mons cloudera-manager-agent
```

```
sudo rm /etc/apt/sources.list.d/cloudera*.list*
```

2. Remove old parcels from the host.

Empty the contents of the following directories. These are the defaults for parcel storage - if you use other directories, please change accordingly.

```
sudo rm -rf /opt/cloudera/parcels/*
```

```
sudo rm -rf /opt/cloudera/parcel-cache/*
```

Upgrade the Operating System



Important: When there are no Hadoop services or Cloudera Manager roles running, you may proceed to upgrade the operating system, make sure to leave the data partitions (for example, dfs.data.dir) unchanged.

Use the operating system upgrade procedures provided by your operating system vendor (for example: RedHat or Ubuntu) to download their software and perform the operating system upgrade.

Step 4: After you upgrade the Operating System to a new Major Version

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)

This topic describes how to upgrade the operating system on a Cloudera Manager managed host.



Important:

Before re-installing Cloudera Manager packages, ensure that you install the correct version of Python 3. See [Installing Python 3](#) for selecting and installing the appropriate Python 3.

Establish Access to the Software

Cloudera Manager needs access to a package repository that contains the updated software packages. You can choose to access the Cloudera public repositories directly, or you can [download those repositories and set up a local repository](#) to access them from within your network. If your cluster hosts do not have connectivity to the Internet, you must set up a local repository.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Log in to each cluster host.

```
ssh CLUSTER_HOST
```

3. Remove any older files in the existing repository directory:

RHEL / CentOS

```
sudo rm /etc/yum.repos.d/cloudera*manager.repo*
```

SLES

```
sudo rm /etc/zypp/repos.d/cloudera*manager.repo*
```

Ubuntu

```
sudo rm /etc/apt/sources.list.d/cloudera*.list*
```

4. Fill in the form at the top of this page.

5. Create a repository file so that the package manager can locate and download the binaries. Do one of the following, depending on whether or not you are using a local package repository:
 - Using a local package repository. (Required when cluster hosts do not have access to the internet.)
 - a. Configure a [local package repository](#) hosted on your network.
 - b. In the Package Repository URL, replace the entire URL with the URL for your local package repository. A username and password are not required to access local repositories.
 - c. Click Apply.
 - Using the Cloudera public repository
 - a. Substitute your *USERNAME* and *PASSWORD* in the Package Repository URL where indicated in the URL.
 - b. Click Apply



Tip: If you have a mixed operating system environment, adjust the Operating System filter at the top of the page for each operating system. The guide will generate the repo file for you automatically here.

6. RHEL / CentOS

Create a file named `/etc/yum.repos.d/cloudera-manager.repo` with the following content:

```
[cloudera-manager]
# Packages for Cloudera Manager
name=Cloudera Manager
baseurl=https://archive.cloudera.com/p/cm7/**Cloudera Manager
version**/redhat[version]/yum/
gpgkey=https://archive.cloudera.com/p/cm7/**Cloudera Manager
version**/redhat[version]/yum/RPM-GPG-KEY-cloudera
gpgcheck=1
```

Copy this file to the same location on all cluster hosts.

SLES

Create a file named `/etc/zypp/repos.d/cloudera-manager.repo` with the following content:

```
[cloudera-manager]
# Packages for Cloudera Manager
name=Cloudera Manager
baseurl=https://archive.cloudera.com/p/cm7/**Cloudera Manager
version**/sles[version]/yum/
gpgkey=https://archive.cloudera.com/p/cm7/**Cloudera Manager
version**/sles[version]/yum/RPM-GPG-KEY-cloudera
gpgcheck=1
```

Copy this file to the same location on all cluster hosts.

Ubuntu

Create a file named `/etc/apt/sources.list.d/cloudera_manager.list` with the following content:

```
# Packages for Cloudera Manager
deb [arch=amd64] https://
myUsername:myPassword@archive.cloudera.com/p/cm7/**Cloudera
```

```
Manager version**]/ubuntu[version]/apt focal-cm[**Cloudera
Manager version**] contrib
```

Run the following command:

```
sudo apt-get update
```

Copy this file to the same location on all cluster hosts.

The repository file, as created, refers to the most recent maintenance release of the specified minor release. If you would like to use a specific maintenance version, for example 5.15.1, replace 5.15 with 5.15.1 in the generated repository file shown above.

7. A Cloudera Manager upgrade can introduce new package dependencies. Your organization may have restrictions or require prior approval for installation of new packages. You can determine which packages may be installed or upgraded:

RHEL / CentOS

```
yum deplist cloudera-manager-agent
```

SLES

```
zypper info --requires cloudera-manager-agent
```

Ubuntu

```
apt-cache depends cloudera-manager-agent
```

Reinstall Cloudera Manager Server, Daemon & Agent Packages

On the Cloudera Manager Server host, re-install the removed Cloudera packages.

1. Install the packages.

Include the cloudera-manager-server-db-2 package in the command only if you are using the embedded PostgreSQL database.

RHEL / CentOS

```
sudo yum clean all
sudo yum install cloudera-manager-server cloudera-manager-daemons
cloudera-manager-agent cloudera-manager-server-db-2
```

SLES

```
sudo zypper clean --all
sudo zypper install cloudera-manager-server cloudera-manager-dae
mons cloudera-manager-agent cloudera-manager-server-db-2
```

Ubuntu

```
sudo apt-get clean
```

```
sudo apt-get update
```

```
sudo apt-get install cloudera-manager-server cloudera-manager-dae
mons cloudera-manager-agent cloudera-manager-server-db-2
```

2. Verify that the configuration files (that were backed up) are intact. Correct if necessary.

3. If you customized the `/etc/cloudera-scm-agent/config.ini` file, your customized file is renamed with the extension `.rpmsave` or `.dpkg-old`. Merge any customizations into the `/etc/cloudera-scm-agent/config.ini` file that is installed by the package manager.

Start Cloudera Manager Server & local Agent

On the host with Cloudera Manager Server, the appropriate services typically will start automatically on reboot. Otherwise, start the Cloudera Manager Server and the local Agent as necessary.

1. Start the Cloudera Manager Server by running the following command:

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

2. Start the Cloudera Manager Agent.

```
sudo systemctl start cloudera-scm-agent
```

If the agent starts without errors, no response displays.

3. Verify that the Cloudera Manager Agent downloaded a proper parcel for your new operating system. You can use the following command to check in Cloudera Manager logs for downloaded parcels:

```
grep "Completed download" /var/log/cloudera-scm-agent/cloudera-scm-agent.log
```

The download might take some time. Look for the operating system in the names of the downloaded parcels. The output of the command should look similar to the following:

```
# grep "Completed download" /var/log/cloudera-scm-agent/cloudera-scm-agent.log
[02/May/2023 07:26:44 +0000] 20004 WorkerThread downloader INFO Completed download of http://nightly-7x-rf-1.nightly-7x-rf.root.hwx.site:7180/cm/parcel/download/CDH-7.2.17-1.cdh7.2.17.p0.40465599-el7.parcel code=200 state=downloaded
[02/May/2023 07:31:00 +0000] 20004 WorkerThread downloader INFO Completed download of http://nightly-7x-rf-1.nightly-7x-rf.root.hwx.site:7180/cm/parcel/download/FLINK-1.12-csa1.4.0.0-cdh7.1.6.0-297-13947709-el7.parcel code=200 state=downloaded
```

Verify that the timestamps of the log lines are recent. Re-run the `grep` command again until the log lines appear. Verify the operating system name and major version in the parcel names and match the new operating system installed on the host.

In the above output example, `el7` refers to Redhat Enterprise Linux 7.



Important: If you are installing parcels from a local repository, then parcels for the new operating system must be downloaded to the local repository first. For more information, see [Step 6: Access Parcels](#) and [Configuring a Local Parcel Repository](#).

Install and start Agent on remaining hosts

1. Install and start the Cloudera Manager Agent package on all the remaining hosts of the cluster by running the following command:

RHEL / CentOS

```
sudo yum clean all
```

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

SLES

```
sudo zypper clean -all
```

```
sudo zypper install cloudera-manager-daemons cloudera-manager-agent
```

Ubuntu

```
sudo apt-get clean
```


```
sudo apt-get update
```

```
sudo apt-get install cloudera-manager-daemons cloudera-manager-agent
```

2. Verify that the configuration files (that were backed up) are intact. Correct if necessary.
3. If you customized the `/etc/cloudera-scm-agent/config.ini` file, your customized file is renamed with the extension `.rpmsave` or `.dpkg-old`. Merge any customizations into the `/etc/cloudera-scm-agent/config.ini` file that is installed by the package manager.
4. Start the Cloudera Manager Agent


```
sudo systemctl start cloudera-scm-agent
```

Start Cloudera Management Service and Clusters

1. Log in to Cloudera Manager as an Administrator.
2. Go to **Hosts All Hosts** page and verify that the status of all the hosts are in green.
3. Go to **Clusters Parcels** and verify that all needed parcels are in the **Distributed** and **Activated** state.
4. Start the Cloudera Management Service as follows:
 - a. On the **Home Status** tab, click the  options menu to the right of **Cloudera Management Service** and select **Start**.



Attention: Verify that the Cloudera Management Service is started.

5. For each cluster on the home page, click the  options menu to the right of each Cluster and select **Start** to start all the services in each of the cluster.



Attention: Verify that all clusters are up and running.

Upgrading the Operating System to a new Minor Version

This topic describes the additional steps needed to upgrade the operating system of a host managed by Cloudera Manager to a higher version for minor release.

Upgrading the operating system to a higher version but within the same major release is called a minor release upgrade. For example, upgrading from Redhat 6.8 to 6.9. This is a relatively simple procedure that involves properly shutting down all the components, performing the operating system upgrade, and then restarting everything in reverse order.

This topic primarily describes all the additional steps such as backing up essential files and removing and reinstalling all the necessary Cloudera Enterprise packages and parcels.



Note: You must determine whether to upgrade the operating system or Cloudera Manager first:

For example, consider the following upgrade from Cloudera Manager 5.13.3 to Cloudera Manager 7.1.4 and an operating system upgrade from RHEL 7.6 to RHEL 7.8

- Cloudera Manager 5.13.3 supports only RHEL 7.6
- Cloudera Manager 7.1.4 supports RHEL 7.6 and RHEL 7.8

In this case you must upgrade Cloudera Manager first, otherwise Cloudera Manager version 5.13.3 would be deployed on an unsupported operating system (RHEL 7.6) and may fail.

Step 1: Getting Started with Operating System Upgrades

Prerequisites

- Ensure that the versions of Cloudera Manager and CDH or Cloudera Runtime support your new operating system.
 - See [Operating System Requirements](#) for Cloudera Base on premises.

If you are using unsupported versions, see [Upgrade Cloudera Manager](#) or [Upgrading a Cloudera Runtime Cluster](#).

- Ensure that the host has access to the Cloudera Manager server, daemon and agent packages that are supported for the new operating system, either by having access to <https://archive.cloudera.com> or a [local package repository](#).
- Ensure that the Cloudera Manager server has access to the parcels that are using supported for the new Operating System, either by having access to <https://archive.cloudera.com> or a [local parcel repository](#).
- If you have a patched package or parcel installed, make sure you have the same package or parcel for the new Operating System and it has been made available to Cloudera Manager.
- Understand that performing a major release upgrade for the operating system in-place may be quite tricky and risky.

Step 2: Backing Up Host Files Before Upgrading the Operating System

This topic describes how to backup important files on your host before upgrading the operating system.

Backing Up

1. Create a top-level backup directory.

```
export CM_BACKUP_DIR=`date +%F`-CM"
echo $CM_BACKUP_DIR
mkdir -p $CM_BACKUP_DIR
```

2. Back up the Agent directory and the runtime state.

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-agent.tar --exclude=*.sock /etc/cloudera-scm-agent /etc/default/cloudera-scm-agent /var/run/cloudera-scm-agent /var/lib/cloudera-scm-agent
```

3. Back up the Server directories:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-scm-server /etc/default/cloudera-scm-server
```

4. Back up the Cloudera Manager databases. See [Backing up Cloudera Manager databases](#) on page 11



Note: Backup is recommended but not always required for a minor release upgrade.

Backing up Cloudera Manager databases

Cloudera recommends that you schedule regular backups of the databases that Cloudera Manager uses to store configuration, monitoring, and reporting data and for managed services that require a database:

Backing Up PostgreSQL Databases

To back up a PostgreSQL database, use the same procedure whether the database is embedded or external:

1. Log in to the host where the Cloudera Manager Server is installed.
2. Get the name, user, and password properties for the Cloudera Manager database from `/etc/cloudera-scm-server/db.properties`:

```
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=NnYfWIjlbk
```

3. Run the following command as root using the parameters from the preceding step:

```
# pg_dump -h HOSTNAME -p 7432 -U scm > /tmp/scm_server_db_backup.$(date +%Y%m%d)
```

4. Enter the password from the `com.cloudera.cmf.db.password` property in step 2.
5. To back up a database created for one of the roles on the local host as the *ROLEUSER* user:

```
# pg_dump -h HOSTNAME -p 7432 -U ROLEUSER > /tmp/ROLEDB
```

6. Enter the password specified when the database was created.

Backing Up MariaDB Databases

To back up the MariaDB database, run the `mysqldump` command on the MariaDB host, as follows:

```
mysqldump -hHOSTNAME -uUSERNAME -pPASSWORD DATABASE > /tmp/DATABASE-BACKUP.sql
```

For example, to back up the Activity Monitor database `amon` created in [Creating Databases for Cloudera Software](#), on the local host as the root user, with the password `amon_password`:

```
mysqldump -pamon_password amon > /tmp/amon-backup.sql
```

To back up the sample Activity Monitor database amon on remote host myhost.example.com as the root user, with the password amon_password:

```
mysqldump -hmyhost.example.com -uroot -pamon_password amon > /tmp/amon-backup.sql
```

Backing Up MySQL Databases

To back up the MySQL database, run the mysqldump command on the MySQL host, as follows:

```
mysqldump -hHOSTNAME -uUSERNAME -pPASSWORD DATABASE > /tmp/DATABASE-BACKUP.sql
```

For example, to back up the Activity Monitor database amon created in [Creating Databases for Cloudera Software](#), on the local host as the root user, with the password amon_password:

```
mysqldump -pamon_password amon > /tmp/amon-backup.sql
```

To back up the sample Activity Monitor database amon on remote host myhost.example.com as the root user, with the password amon_password:

```
mysqldump -hmyhost.example.com -uroot -pamon_password amon > /tmp/amon-backup.sql
```

You can back up all database using the following command:

```
mysqldump --all-databases -pPASSWORD > /tmp/all/all.sql
```

Backing Up Oracle Databases

For Oracle, work with your database administrator to ensure databases are properly backed up.

Step 3: Before you upgrade the Operating System to a new Minor Version

This topic describes steps you must perform before upgrading the operating system on a host managed by Cloudera Manager.



Warning: When applying operating system patches, the vendor may change libraries critical to Cloudera software. Therefore, you must always plan to do a cluster restart. Alternatively, you can decommission the hosts before an upgrade or patch and recommission once the operation is complete.

Decommission and Stop Running Roles

1. Log in to the Cloudera Manager Admin Console.
2. From the All Hosts page, select the host that you wish to upgrade. Cloudera recommends that you upgrade only one host at a time.
3. Select Begin Maintenance (Suppress Alerts/Decommission) from the Actions menu.
4. Select Host Decommission from the Actions menu. Any roles that do not require decommission will be skipped.

5. If the operating system upgrade procedure takes less than 30 minutes per node, you do not need to decommission the DataNode.

If the Cloudera Manager and CDH/ version are both 5.14 or greater, you can also choose the Take DataNode Offline feature.

If in doubt, decommission the roles.

- When a DataNode is decommissioned, the NameNode ensures that every block from the DataNode is still available across the cluster as specified by the replication factor. This procedure involves copying blocks off the DataNode in small batches. In cases where a DataNode has several thousand blocks, decommissioning takes several hours.
- When a DataNode is turned off without being decommissioned:
 - The NameNode marks the DataNode as dead after a default of 10m 30s (controlled by the `dfs.heartbeat.interval` and `dfs.heartbeat.recheck.interval` configuration properties).
 - The NameNode schedules the missing replicas to be placed on other DataNodes.
 - When the DataNode comes back online and reports to the NameNode, the NameNode schedules blocks to be copied to it while other nodes are decommissioned or when new files are written to HDFS.
- You can also speed up the decommissioning of a DataNode by increasing values for these properties:
 - `dfs.max-repl-streams`: The number of simultaneous streams used to copy data.
 - `dfs.balance.bandwidthPerSec`: The maximum amount of bandwidth that each DataNode can utilize for balancing, in bytes per second.
 - `dfs.namenode.replication.work.multiplier.per.iteration`: NameNode configuration requiring a restart, defaults to 2 but can be raised to 10 or higher.

This determines the total amount of block transfers to begin in parallel at a DataNode for replication, when such a command list is being sent over a DataNode heartbeat by the NameNode. The actual number is obtained by multiplying this value by the total number of live nodes in the cluster. The result number is the number of blocks to transfer immediately, per DataNode heartbeat.

6. Once that is completed, select the same host again and choose Stop Roles on Hosts.



Warning: If you have not enabled high availability for HDFS, HBase, MapReduce, YARN, Oozie, or Sentry, stopping the running single master role will cause an outage for that service. Specifically, secondary roles on other hosts will stop abruptly. Cloudera recommends that you stop these services prior to the host upgrade.



Important: When upgrading hosts that are part of a ZooKeeper quorum, ensure that the majority of the quorum is still available.

Stop Cloudera Manager Agent

1. Hard Stop the Cloudera Manager Agent.
RHEL 7, SLES 12, Ubuntu 18.04 and higher

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

RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04 or 14.04

```
sudo service cloudera-scm-agent hard_stop
```



Important: This will ask you to confirm with `hard_stop_confirmed` because this will terminate any Hadoop services on the host (if any) unconditionally.

Stop Cloudera Manager Server & Agent

1. Hard Stop the Cloudera Manager Agent.
RHEL 7, SLES 12, Ubuntu 18.04 and higher

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

- RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04 or 14.04**

```
sudo service cloudera-scm-agent hard_stop
```



Important: This will ask you to confirm with `hard_stop_confirmed` because this will terminate any Hadoop services on the host (if any) unconditionally.

2. Stop the Cloudera Management Service.
 - a. Log in to the Cloudera Manager Admin Console.
 - b. Select Clusters Cloudera Management Service .
 - c. Select Actions Stop .
3. Stop the Cloudera Manager Server.

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

Stop Databases

1. If you are using the embedded PostgreSQL database, stop the Cloudera Manager Embedded PostgreSQL database:

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

If you are not using the embedded PostgreSQL database and you attempt to stop it, you might see a message indicating that the service cannot be found.

In case of using the embedded PostgreSQL, if you see a message that the shutdown failed, then the embedded database is still running, probably because services are connected to the Hive metastore. If the database shutdown fails due to connected services, issue the following command:

```
sudo service cloudera-scm-server-db next_stop_fast  
sudo service cloudera-scm-server-db stop
```

2. If there are other database servers running on this host, they must be stopped also.

Remove Packages & Parcels

Packages for the older operating system won't be able to start on the new operating system. Remove old packages from the host.

1. RHEL / CentOS

```
sudo yum remove cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server-db-2
```

SLES

```
sudo zypper remove cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server-db-2
```

Ubuntu

```
sudo apt-get purge cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server-db-2
```

2. RHEL / CentOS

```
sudo yum remove cloudera-manager-server cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server-db-2
```

SLES

```
sudo zypper remove cloudera-manager-server cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server-db-2
```

Ubuntu

```
sudo apt-get purge cloudera-manager-server cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server-db-2
```

3. Remove old CDH parcels from the host. These are built for your old operating system.

The Cloudera Manager agent will download and activate the proper parcel for the new operating system when it is started.

Empty the contents of the following directories. These are the defaults for parcel storage - if you use other directories, please change accordingly.

```
sudo rm -rf /opt/cloudera/parcels/*
```

```
sudo rm -rf /opt/cloudera/parcel-cache/*
```

Upgrade the Operating System



Important: When there are no Hadoop services or Cloudera Manager roles running from this host, you may proceed to upgrade the operating system of this host, make sure to leave the data partitions (for example, `dfs.data.dir`) unchanged.

Use the operating system upgrade procedures provided by your operating system vendor (for example: RedHat or Ubuntu) to download their software and perform the operating system upgrade.

Step 4: After you upgrade the Operating System to a new Minor Version

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)

This topic describes how to upgrade the operating system on a Cloudera Manager managed host.

Establish Access to the Software

Cloudera Manager needs access to a package repository that contains the updated software packages. You can choose to access the Cloudera public repositories directly, or you can [download those repositories and set up a local repository](#) to access them from within your network. If your cluster hosts do not have connectivity to the Internet, you must set up a local repository.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Log in to each cluster host.

```
ssh CLUSTER_HOST
```

3. Remove any older files in the existing repository directory:

RHEL / CentOS

```
sudo rm /etc/yum.repos.d/cloudera*manager.repo*
```

SLES

```
sudo rm /etc/zypp/repos.d/cloudera*manager.repo*
```

Ubuntu

```
sudo rm /etc/apt/sources.list.d/cloudera*.list*
```

4. Fill in the form at the top of this page.
5. Create a repository file so that the package manager can locate and download the binaries. Do one of the following, depending on whether or not you are using a local package repository:
 - Using a local package repository. (Required when cluster hosts do not have access to the internet.)
 - a. Configure [a local package repository](#) hosted on your network.
 - b. In the Package Repository URL, replace the entire URL with the URL for your local package repository. A username and password are not required to access local repositories.
 - c. Click Apply.
 - Using the Cloudera public repository
 - a. Substitute your *USERNAME* and *PASSWORD* in the Package Repository URL where indicated in the URL.
 - b. Click Apply



Tip: If you have a mixed operating system environment, adjust the Operating System filter at the top of the page for each operating system. The guide will generate the repo file for you automatically here.

6. RHEL / CentOS

Create a file named `/etc/yum.repos.d/cloudera-manager.repo` with the following content:

```
[cloudera-manager]
# Packages for Cloudera Manager
name=Cloudera Manager
baseurl=https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/5.15
gpgkey=https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/RPM-
GPG-KEY-cloudera
```

```
gpgcheck=1
```

SLES

Create a file named `/etc/zypp/repos.d/cloudera-manager.repo` with the following content:

```
[cloudera-manager]
# Packages for Cloudera Manager
name=Cloudera Manager
baseurl=https://archive.cloudera.com/cm5/sles/12/x86_64/cm/5.15
gpgkey=https://archive.cloudera.com/cm5/sles/12/x86_64/cm/RPM-
GPG-KEY-cloudera
gpgcheck=1
```

Ubuntu

Create a file named `/etc/apt/sources.list.d/cloudera_manager.list` with the following content:

```
# Packages for Cloudera Manager
deb https://archive.cloudera.com/cm5/debian/jessie/amd64/cm/
jessie-cm5.15 contrib
deb-src https://archive.cloudera.com/cm5/debian/jessie/amd64/cm/
jessie-cm5.15 contrib
```

Run the following command:

```
sudo apt-get update
```

The repository file, as created, refers to the most recent maintenance release of the specified minor release. If you would like to use a specific maintenance version, for example 5.15.1, replace 5.15 with 5.15.1 in the generated repository file shown above.

7. A Cloudera Manager upgrade can introduce new package dependencies. Your organization may have restrictions or require prior approval for installation of new packages. You can determine which packages may be installed or upgraded:

RHEL / CentOS

```
yum deplist cloudera-manager-agent
```

SLES

```
zypper info --requires cloudera-manager-agent
```

Ubuntu

```
apt-cache depends cloudera-manager-agent
```

Reinstall Cloudera Manager Server, Daemon & Agent Packages

On the Cloudera Manager Server host, re-install the removed Cloudera packages.

1. Install the packages.

Include the `cloudera-manager-server-db-2` package in the command only if you are using the embedded PostgreSQL database.

RHEL / CentOS

```
sudo yum clean all
```

```
sudo yum install cloudera-manager-server cloudera-manager-daemons  
cloudera-manager-agent cloudera-manager-server-db-2
```

SLES

```
sudo zypper clean --all  
sudo zypper install cloudera-manager-server cloudera-manager-daemons  
cloudera-manager-agent cloudera-manager-server-db-2
```

Ubuntu

```
sudo apt-get clean
```

```
sudo apt-get update
```

```
sudo apt-get install cloudera-manager-server cloudera-manager-daemons  
cloudera-manager-agent cloudera-manager-server-db-2
```

2. Verify that the configuration files (that were backed up) are intact. Correct if necessary.
3. If you customized the /etc/cloudera-scm-agent/config.ini file, your customized file is renamed with the extension .rpmsave or .dpkg-old. Merge any customizations into the /etc/cloudera-scm-agent/config.ini file that is installed by the package manager.

Start Databases

1. If you are using the embedded PostgreSQL database, start the database:

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

2. If there were database servers stopped, they must be restarted.

Start Server & Agent

The appropriate services typically will start automatically on reboot. Otherwise, start the Server & Agent as necessary.

1. Start the rpcbind service if it is not automatically started.

```
sudo service rpcbind start
```

2. Start the Cloudera Manager Agent.

```
sudo systemctl start cloudera-scm-agent
```

If the agent starts without errors, no response displays.

3. Start the Cloudera Manager Server.

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

4. Verify that the Agent downloaded a proper parcel for your new operating system. You can use the following command to check in logs for downloaded parcels:

```
grep "Completed download" /var/log/cloudera-scm-agent/cloudera-scm-agent  
.log
```

(Download might take some time. Look for the operating system in the names of the downloaded parcels.)

Start Roles

1. From the All Hosts page, select the host that you have just upgraded.
2. Choose End Maintenance (Enable Alerts/Decommission) from the Actions menu and confirm.
3. Start any roles that were running on this host and were stopped.
4. Choose Host Recommission from the Actions menu and confirm.
5. Choose Start Roles on Hosts from the Actions menu and confirm.
6. Start any services that were stopped due to lack of high availability.

Upgrading the JDK

Cloudera Manager, Cloudera Runtime, and CDH require a supported version of the Java Development Kit (JDK) to be installed on all hosts. For details, see [Cloudera Java Requirements](#).

Loading Filters ...



Warning:

- If you are upgrading from a lower major version of the JDK to JDK 1.8 or from JDK 1.6 to JDK 1.7, and you are using AES-256 bit encryption, you must install new encryption policy files. (In a Cloudera Manager deployment, you automatically install the policy files; for unmanaged deployments, install them manually.) See [Using AES-256 Encryption](#) on page 36. This step is not required when using JDK 1.8.0_162 or greater. JDK 1.8.0_162 enables unlimited strength encryption by default.

You must also ensure that the Java Truststores are retained during the upgrade.

Cloudera recommends the following for keystores and truststores for Cloudera Manager clusters:

- Create a separate keystore for each host. Each keystore should have a name that helps identify it as to the type of host—server or agent, for example. The keystore contains the private key and should be password protected.
- Create a single truststore that can be used by the entire cluster. This truststore contains the root CA and intermediate CAs used to authenticate certificates presented during TLS/SSL handshake. The truststore does not need to be password protected. (See [Understanding Keystores and Truststores](#) to Truststore for more information about the truststore for TLS/SSL and Cloudera clusters.)

There are several procedures you can use to upgrade the JDK:

- [Installing Java During an Upgrade](#)

If you are upgrading to Cloudera Manager 6.0.0 or higher, you can manually install JDK 1.8 on the Cloudera Manager server host, and then, as part of the Cloudera Manager upgrade process, you can specify that Cloudera Manager upgrade the JDK on the remaining hosts.



Note: Cloudera Manager only installs Oracle JDK. You can upgrade to OpenJDK using these steps.

- [Manually Installing Oracle JDK 1.8](#) on page 28

You can manually install JDK 1.8 on all managed hosts. If you are upgrading to any version of Cloudera Manager 5.x, you must use this procedure. Continue with the steps in the next section.

- [Manually Migrating to OpenJDK](#) on page 32

Manually Installing Oracle JDK 1.8



Important: Manual upgrade of Oracle JDK 1.8 requires down time to stop and restart your cluster.



Important: Cloudera Manager Server with JDK 8 does not support G1GC.

You can manually install Oracle JDK 1.8 on all managed hosts. If you are upgrading to any version of Cloudera Manager 5.x, you must use the following procedure:

1. Download the .tar.gz file for one of the 64-bit versions of Oracle JDK 1.8 from [Java SE 8 Downloads](#). (This link is correct at the time of writing, but can change.)
2. Perform the following steps on all hosts that you are upgrading:
 - a. Log in to the host as root using ssh.
 - b. Copy the downloaded .tar.gz file to the host.
 - c. Extract the JDK to the folder /usr/java/JDK-VERSION. For example:

```
tar xvfz /PATH/TO/jdk-8u<UPDATE_VERSION>-linux-x64.tar.gz -C /usr/java/
```

3. If you have configured TLS for Cloudera Manager (see [Encrypting Data in Transit](#)), copy the jssecacerts file from the previous JDK installation to the new JDK installation. This step is not required when using JDK 1.8.0_162 or greater. JDK 1.8.0_162 enables unlimited strength encryption by default.

For example:

```
cp PREVIOUS_JAVA_HOME/jre/lib/security/jssecacerts NEW_JAVA_HOME/jre/lib/security
```

(Substitute *PREVIOUS_JAVA_HOME* and *NEW_JAVA_HOME* with the paths to the JDK installations.)

4. Configure the location of the JDK on cluster hosts.
 - a. Open the Cloudera Manager Admin Console.
 - b. In the main navigation bar, click the Hosts tab. If you are configuring the JDK location on a specific host only, click the link for that host.
 - c. Click the Configuration tab.
 - d. Select Category Advanced .
 - e. Set the Java Home Directory property to the custom location.
 - f. Click Save Changes.
5. On the Cloudera Manager Server host only (not required for other hosts):
 - a. Open the file /etc/default/cloudera-scm-server in a text editor.
 - b. Edit the line that begins with export JAVA_HOME (if this line does not exist, add it) and change the path to the path of the new JDK (you can find the path under /usr/java).

For example: (RHEL and SLES)

```
export JAVA_HOME="/usr/java/jdk1.8.0_141-cloudera"
```

For example: (Ubuntu)

```
export JAVA_HOME="/usr/lib/jvm/java-8-oracle-cloudera"
```

- c. Save the file.
- d. Restart the Cloudera Manager Server.

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

6. Restart the Cloudera Management Service.
 - a. Log in to the Cloudera Manager Admin Console.
 - b. Select Clusters Cloudera Management Service .
 - c. Select Actions Restart .

7. Restart all clusters:

- a. On the Home Status tab, click the Options menu to the right of the cluster name and select Restart.
- b. Click Restart that appears in the next screen to confirm. If you have enabled [high availability for HDFS](#), you can choose [Rolling Restart](#) instead to minimize cluster downtime. The Command Details window shows the progress of stopping services.

When All services successfully started appears, the task is complete and you can close the Command Details window.

8. Delete the files from your previous Java installation. If you do not delete these files, Cloudera Manager and other components may continue to use the old version of the JDK.

OpenJDK

You must install a supported version of OpenJDK. If your deployment uses a version of OpenJDK lower than 1.8.0_232, see [TLS Protocol Error with OpenJDK](#).



Important: For OpenJDK 8u241 and higher versions running on Kerberized clusters, you must disable referrals:

- Cloudera Manager 7.1.1 or higher:
 1. Log in to the Cloudera Manager Admin Console.
 2. Go to Administration Settings .
 3. Select the Advanced category.
 4. Locate the JVM Arguments for Java-based services parameter and enter the following:

```
-Dsun.security.krb5.disableReferrals=true
```

5. Restart any stale services.
- Cloudera Manager 7.0.3 or lower:
 1. Edit the Java Security file on all hosts by adding or changing the following configuration :
 - 2.

```
sun.security.krb5.disableReferrals=true
```

If the configuration already exists and is set to false, change it to true.

3. Restart the cluster.

For more information, see the [KB article](#).

Manually Installing OpenJDK

Before installing or upgrading Cloudera Manager and CDH/Cloudera Runtime, perform the steps in this section to install [OpenJDK](#) on all hosts in your cluster(s).

When you install Cloudera Enterprise, Cloudera Manager includes an option to install Oracle JDK. De-select this option.



Note: If you intend to enable [Auto TLS](#), note the following:

You can specify a PEM file containing trusted CA certificates to be imported into the Auto-TLS truststore. If you want to use the certificates in the cacerts truststore that comes with OpenJDK, you must convert the truststore to PEM format first. However, OpenJDK ships with some intermediate certificates that cannot be imported into the Auto-TLS truststore. You must remove these certificates from the PEM file before importing the PEM file into the Auto-TLS truststore. This is not required when upgrading to OpenJDK from a cluster where Auto-TLS has already been enabled.



Important: Cloudera Manager Server with JDK 8 does not support G1GC.

1. Log in to each host and run the following command:

RHEL

OpenJDK 8*

```
sudo yum install java-1.8.0-openjdk-devel
```

OpenJDK 11*

```
sudo yum install java-11-openjdk-devel
```

OpenJDK 17*

```
sudo yum install java-17-openjdk-devel
```

Ubuntu

OpenJDK 8*

```
sudo apt-get install openjdk-8-jdk
```

OpenJDK 11*

```
sudo apt install openjdk-11-jdk
```

OpenJDK 17*

```
sudo apt install openjdk-17-jdk
```

SLES

OpenJDK 8*

```
sudo zypper install java-1_8_0-openjdk-devel
```

OpenJDK 11*

```
zypper install java-11-openjdk-devel
```

OpenJDK 17*

```
sudo zypper --non-interactive install java-17-openjdk-devel
```

* Azul OpenJDK, OpenJDK 8, OpenJDK 11, and OpenJDK 17 are TCK certified for Cloudera.

2. If you are using the SLES operating system, Cloudera Manager needs an additional configuration so that the JDK can be located:
 - a. Log in to the Cloudera Manager server host.
 - b. Open the following file in a text editor:

```
/etc/default/cloudera-scm-server
```

- c. Add the following line:

```
export JAVA_HOME=PATH TO THE JAVA INSTALLATION DIRECTORY
```

For example:

```
export JAVA_HOME=/usr/lib64/jvm/java-1.8.0-openjdk-1.8.0
```

- d. Save the file.
- e. Restart the Cloudera Manager Server.

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



Important: If you are upgrading to OpenJDK 11 or higher, set `allow_weak_crypto` to 'true' in `/etc/krb5.conf` to get Kerberos setup working with JDK11(or higher) and keep this set during the Cloudera Manager upgrade.



Important: If you are upgrading to OpenJDK 1.8.0_392 or higher, or Oracle JDK 1.8u351 or higher, set `allow_weak_crypto` to 'true' in `/etc/krb5.conf` to get Kerberos setup working with OpenJDK 1.8.0_392 or higher, or Oracle JDK 1.8u351 or higher and keep this set during the Cloudera Manager upgrade.

Manually Migrating to OpenJDK

If you have Oracle JDK 1.7, Oracle JDK 1.8, or OpenJDK 8* installed on the hosts managed by Cloudera Manager, use the steps in this section to transition your deployment to use [OpenJDK](#). The steps below require you to restart all clusters, which will cause downtime as the hosts restart. If your clusters have enabled [high availability for HDFS](#), you can use a [Rolling Restart](#) to restart the clusters without downtime. Note that until the rolling restart completes, some of the hosts in your cluster will still be using the Oracle JDK. If you do not want a temporarily mixed environment, you can stop the cluster before performing the steps in this section to transition the JDK.

OpenJDK 11* is supported as of Cloudera Manager and CDH 6.3. Note the following:

- You must upgrade to Cloudera Manager 6.3 or higher, **before** upgrading to OpenJDK 11*.
- The package names used when installing the OpenJDK 11* are different and are noted in the steps below.
- The path for the default truststore has changed from (OpenJDK 8*) `jre/lib/security/cacerts` to (OpenJDK 11*) `lib/security/cacerts`
- See the following blog post for general information about migrating to Java 11: [All You Need to Know For Migrating To Java 11](#).

OpenJDK 17* is supported as of Cloudera Manager 7.11.3 and Cloudera 7.1.9. Note the following:

- You must upgrade to Cloudera Manager 7.11.3 or higher, **before** upgrading to OpenJDK 17*.
- The package names used when installing the OpenJDK 17* are different and are noted in the steps below.
- The path to default truststore for OpenJDK 17* is `lib/security/cacerts`.
- See the following blog post for general information about migrating to Java 17: [Migrate to Java 17](#).

1. Find out the package name of your currently installed JDK by running the following commands. The grep commands attempt to locate the installed JDK. If the JDK package is not returned, try looking for the string **jdk**.
RHEL

Oracle JDK 8

```
yum list installed |grep oracle
```

OpenJDK 8*, OpenJDK 11*, or OpenJDK 17*

```
yum list installed |grep openjdk
```

Ubuntu**Oracle JDK 8**

```
apt list --installed | grep oracle
```

OpenJDK 8*, OpenJDK 11*, or OpenJDK 17*

```
apt list --installed | grep openjdk
```

SLES**Oracle JDK 8**

```
zypper search --installed-only |grep oracle
```

OpenJDK 8*, OpenJDK 11*, or OpenJDK 17*

```
zypper search --installed-only |grep openjdk
```

The command will return values similar to the following example::

oracle-j2sdk1.7.x86_64	1.7.0+update67-1	java-1.8.0-
openjdk-devel		

The Oracle JDK package name in the above example is: oracle-j2sdk1.7.x86_64. The OpenJDK package is java-1.8.0-openjdk-devel.

2. Log in to each host managed by Cloudera Manager (including the Cloudera Manager server host) and run the following command to install OpenJDK:

RHEL

OpenJDK 8*

```
sudo yum install java-1.8.0-openjdk-devel
```

OpenJDK 11*

```
sudo yum install java-11-openjdk-devel
```

OpenJDK 17*

```
sudo yum install java-17-openjdk-devel
```

Ubuntu

OpenJDK 8*

```
sudo apt-get install openjdk-8-jdk
```

OpenJDK 11*

```
sudo apt install openjdk-11-jdk
```

OpenJDK 17*

```
sudo apt install openjdk-17-jdk
```

SLES

OpenJDK 8*

```
sudo zypper install java-1_8_0-openjdk-devel
```

OpenJDK 11*

```
zypper install java-11-openjdk-devel
```

OpenJDK 17*

```
sudo zypper --non-interactive install java-17-openjdk-devel
```

3. (This step is required for Oracle JDK 8 or OpenJDK 8* only) On the Cloudera Manager Server host only (not required for other hosts):

- a. Open the file `/etc/default/cloudera-scm-server` in a text editor.
- b. Edit the line that begins with `export JAVA_HOME` (if this line does not exist, add it) and change the path to the path of the new JDK (the JDK is usually installed in `/usr/lib/jvm`)(or `/usr/lib64/jvm` on SLES 12), but the path may differ depending on how the JDK was installed).

For example:

RHEL 7, 8

```
export JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk"
```

Ubuntu

```
export JAVA_HOME="/usr/lib/jvm/openjdk-8-jdk"
```

SLES

```
export JAVA_HOME="/usr/lib64/jvm/java-1.8.0-openjdk"
```

- c. Save the file.
- d. Restart the Cloudera Manager Server.

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

4. Tune the JDK (OpenJDK 11* or OpenJDK 17*).

OpenJDK 11* or OpenJDK 17* uses new defaults for garbage collection and other Java options specified when launching Java processes. Due to these changes you may need to tune the garbage collection by adjusting the Java options used to run cluster services, which are configured separately for each service using the service's configuration parameters. To locate the correct parameter, log in to the Cloudera Manager Admin Console, go to the cluster and service you want to configure and search for "Java Configuration Options".

When using OpenJDK 11* or OpenJDK 17*, Cloudera Manager and most services use G1GC as the default method of garbage collection. Java 8 used "ConcurrentMarkSweep" (CMS) for garbage collection. When using G1GC, the pauses for garbage collection are shorter, so components will usually be more responsive, but they are more sensitive to JVMs with overcommitted memory usage. See [Tuning JVM Garbage Collection](#) on page 38.




Important: For OpenJDK 17*, Cloudera Manager and most services run with default GC without any custom tuning for any service.

5. Restart the Cloudera Management Service.

- a. Log in to the Cloudera Manager Admin Console.
- b. Select Clusters Cloudera Management Service .
- c. Select Actions Restart .

6. Restart all clusters:

- a. On the Home Status tab, click  to the right of the cluster name and select either Restart or Rolling Restart. Selecting [Rolling Restart](#) minimizes cluster downtime and is available only if you have enabled [Auto TLS](#).
- b. Click Restart or Rolling Restart that appears in the next screen to confirm. The Command Details window shows the progress of stopping services.

When All services successfully started appears, the task is complete and you can close the Command Details window.

7. Remove the JDK:

a. Perform the following steps on all hosts managed by Cloudera Manager:

1. Run the following command to remove the JDK, using the package names from Step 1: (If you do not delete these files, Cloudera Manager and other components may continue to use the old version of the JDK.)

RHEL

```
yum remove <JDK PACKAGE NAME>
```

Ubuntu

```
apt-get remove <JDK PACKAGE NAME>
```

SLES

```
zypper rm <JDK PACKAGE NAME>
```

2. Confirm that the package has been removed:

RHEL

```
yum list installed |grep -i java
```

Ubuntu

```
apt list --installed | grep -i java
```

SLES

```
zypper search --installed-only |grep -i java
```

Using AES-256 Encryption



Note: This step is not required when using JDK 1.8.0_162 or greater. JDK 1.8.0_162 enables unlimited strength encryption by default.

If you are using CentOS/Red Hat Enterprise Linux 5.6 or higher, or Ubuntu, which use AES-256 encryption by default for tickets, you must install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy File on all cluster and Hadoop user machines. For JCE Policy File installation instructions, see the README.txt file included in the jce_policy-x.zip file. For more information, see [Java Cryptography Extension \(JCE\) Unlimited Strength Jurisdiction Policy File](#)

Alternately, you can configure Kerberos to not use AES-256 by removing aes256-cts:normal from the supported_enctypes field of the kdc.conf or krb5.conf file. After changing the kdc.conf file, you must restart both the KDC and the kadmin server for those changes to take affect. You may also need to re-create or change the password of the relevant principals, including, potentially the Ticket Granting Ticket principal (krbtgt/REALM@REALM). If AES-256 is still used after completing steps, the aes256-cts:normal setting existed when the Kerberos database was created. To fix this, create a new Kerberos database and then restart both the KDC and the kadmin server.

To verify the type of encryption used in your cluster:

1. On the local KDC host, type this command to create a test principal:

```
kadmin -q "addprinc test"
```

2. On a cluster host, type this command to start a Kerberos session as the test principal:

```
kinit test
```

3. On a cluster host, type this command to view the encryption type in use:

```
klist -e
```

If AES is being used, output like the following is displayed after you type the klist command; note that AES-256 is included in the output:

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: test@SCM
Valid starting      Expires              Service principal
05/19/11 13:25:04  05/20/11 13:25:04  krbtgt/SCM@SCM
      Etype (skey, tkt): AES-256 CTS mode with 96-bit SHA-1 HMAC, AES-256 CT
      S mode with 96-bit SHA-1 HMAC
```

Configuring a Custom Java Home Location



Note: Cloudera strongly recommends installing Oracle JDK at `/usr/java/<JDK-VERSION>` and OpenJDK at `/usr/lib/jvm`, which allows Cloudera Manager to auto-detect and use the correct JDK version. If you install the JDK anywhere else, you must follow these instructions to configure Cloudera Manager with your chosen location. The following procedure only changes the JDK location for Cloudera Management Services and cluster processes that are launched by the Cloudera Manager agents.



Important:

The procedure described on this page does not affect the JDK used by other non-Cloudera processes, including Hadoop processes such as the `hdfs` command.

Although not recommended, the Java Development Kit (JDK), which Cloudera services require, may be installed at a custom location if necessary. These steps assume you have already installed the JDK during product installation or as part of an upgrade.

To modify the Cloudera Manager configuration to ensure the JDK can be found:

1. Log into the Cloudera Manager server host.
2. Open the following file in a text editor:

```
/etc/default/cloudera-scm-server
```

3. Add the following line:

```
export JAVA_HOME=PATH TO THE JAVA INSTALLATION DIRECTORY
```

For example:

```
export JAVA_HOME=/usr/lib64/jvm/java-1.8.0-openjdk-1.8.0
```

4. Save the file.
5. Restart the Cloudera Manager Server.

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

6. Open the Cloudera Manager Admin Console.
7. In the main navigation bar, click the Hosts tab. If you are configuring the JDK location on a specific host only, click the link for that host.
8. Click the Configuration tab.

9. Select Category Advanced.
10. Set the Java Home Directory property to the custom location.
11. Click Save Changes.
12. Restart all services.

Tuning JVM Garbage Collection

When using OpenJDK 11* or OpenJDK 17*, Cloudera Manager and most Cloudera Runtime services use G1GC as the default method of garbage collection. (Java 8 used "ConcurrentMarkSweep" (CMS) for garbage collection.) When using G1GC, the pauses for garbage collection are shorter, so components will usually be more responsive, but they are more sensitive to overcommitted memory usage. You should monitor memory usage to determine whether memory is overcommitted.

Cloudera Manager alerts you when memory is overcommitted on cluster hosts. To view these alerts and adjust the allocations:

1. Log in to the Cloudera Manager Admin Console
2. Go to Home Configuration Configuration Issues.
3. Look for entries labeled Memory Overcommit Validation Threshold and note the hostname of the affected host.
4. Go to Hosts All Hosts and click on the affected host.
5. Click the Resources tab.
6. Scroll down to the Memory section.

A list of roles instances and their memory allocations are displayed. The Description column displays the configuration property name where the memory allocation can be set.

7. To adjust the memory allocation, search for the configuration property and adjust the value to reduce the overcommitment of memory. You may need to move some roles to other hosts if there is not sufficient memory for the roles running on the host.
8. After making any changes, Cloudera Manager will indicate that the service has a stale configuration and prompt you to [restart the service](#).

You may also need to adjust the Java options used to start Java processes. You can add Java startup options using Cloudera Manager configuration properties that are available for all service roles. Cloudera has provided default arguments for some of the services where they are needed. You can add to these, or completely override all of the provided Java options. For more information on configuring G1GC, see [The OpenJDK documentation](#).

If default options are provided, the role configuration specifies a single value, `{{JAVA_GC_ARGS}}`. This value is a placeholder for the default Java Garbage Collection options provided with Cloudera Manager and Cloudera Runtime.

To modify Java options:

1. Log in to the Cloudera Manager Admin Console.
2. Go to the service where you want to modify the options. (For the Cloudera Manager Service Monitor, select the Cloudera Management Service.)
3. Select the Configuration tab.
4. Enter "Java" in the search box.
5. Locate the Java Configuration Options property named for the role you want to modify. For example, in the HDFS service, you will see parameters like Java Configuration Options for DataNode and Java Configuration Options for JournalNode.
6. To add to the Java options, enter additional options before or after the `{{JAVA_GC_ARGS}}` placeholder, separated by spaces. For example:

```
{{JAVA_GC_ARGS}} -XX:MaxPermSize=512M
```

* Azul OpenJDK, OpenJDK 8, OpenJDK 11, and OpenJDK 17 are TCK certified for Cloudera.

7. To replace the default Java options, delete the `{{JAVA_GC_ARGS}}` placeholder and replace it with one or more Java options, separated by spaces.
8. The service will now have a stale configuration and must be restarted. See [Restarting a service](#).



Important: No additional GC tuning has been applied to any service if they are running with OpenJDK 17*.



Important: Cloudera Manager Server with JDK 8 does not support G1GC.

Table 2: Default Java Options

Service and Role	Default Java 8 Options	Default Java 11 Options
<ul style="list-style-type: none"> Cloudera Manager Service Monitor 	<pre>-XX:+UseConcMarkSweepGC -XX:+UseParNewGC</pre> <p>To enable G1GC:</p> <pre>-XX:+UseG1GC -XX:-UseConcMarkSweepGC -XX:-UseParNewGC</pre>	
<ul style="list-style-type: none"> HDFS DataNode HDFS NameNode HDFS Secondary NameNode 	<pre>-XX:+UseParNewGC -XX:+UseConcMarkSweepGC - XX:CMSInitiatingOccupancyFraction=70 -XX: +CMSParallelRemarkEnabled</pre>	<pre>-XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=70 -XX:+CMSParallelRemarkEnabled</pre>
<ul style="list-style-type: none"> Hive Metastore Server HiveServer 2 WebHCat Server 	<pre>-XX:+UseParNewGC -XX:+UseConcMarkSweepGC - XX:CMSInitiatingOccupancyFraction=70 -XX: +CMSParallelRemarkEnabled</pre>	None, G1GC is enabled by default.
<ul style="list-style-type: none"> HBase REST Server HBase Thrift Server HBase Master HBase RegionServer 	<pre>-XX:+UseParNewGC -XX:+UseConcMarkSweepGC - XX:CMSInitiatingOccupancyFraction=70 -XX: +CMSParallelRemarkEnabled</pre>	None, G1GC is enabled by default.
<ul style="list-style-type: none"> HBase Region Server 	<pre>-XX:+UseParNewGC -XX:+UseConcMarkSweepGC - XX:CMSInitiatingOccupancyFraction=70 -XX: +CMSParallelRemarkEnabled -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps</pre>	<pre>-verbose:gc -Xlog:gc</pre>
<ul style="list-style-type: none"> MapReduce JobTracker MapReduce TaskTracker 	<pre>-XX:+UseParNewGC -XX:+UseConcMarkSweepGC - XX:CMSInitiatingOccupancyFraction=70 -XX: +CMSParallelRemarkEnabled</pre>	None, G1GC is enabled by default.
<ul style="list-style-type: none"> Solr Server 	<pre>-XX:+UseParNewGC -XX:+UseConcMarkSweepGC - XX:CMSInitiatingOccupancyFraction=70 -XX: +CMSParallelRemarkEnabled</pre>	None, G1GC is enabled by default.
<ul style="list-style-type: none"> YARN JobHistory Server 	<pre>-XX:+UseParNewGC</pre>	<pre>-Dlibrary.leveldbjni.path={{</pre>

Upgrading Python

Learn how to migrate Python for different combinations of Cloudera Manager and Cloudera Runtime on the supported operating systems.

Migrating Python versions when using Cloudera Manager 7.13.1 CHF4 (7.13.1.400) with Cloudera Runtime 7.1.9 SP1 CHF9

Migrating from Python 3.8 or Python 3.9 to Python 3.11 on supported operating systems (Recommended) Installing Python 3.11 before upgrading Cloudera Manager to 7.13.1 CHF4 (7.13.1.400)

This is the recommended workflow for customers upgrading Cloudera Manager to 7.13.1 CHF4 (7.13.1.400) version.

1. Install Python 3.11 from the standard repository. For information about installing Python 3.11 on supported operating systems, see [Installing Python 3.11 standard package on supported operating systems](#).



Important:

- Install Python 3.11 before upgrading Cloudera Manager to 7.13.1 CHF4 version to avoid performing a Cloudera Manager Agent restart later. If you install Python 3.11 after upgrading Cloudera Manager to 7.13.1 CHF4 version, then you must perform Cloudera Manager Agent restart after installing Python 3.11. For information about restarting Cloudera Manager Agents, see [Starting, Stopping, and Restarting Cloudera Manager Agents](#).
 - Similarly, if you perform the Python migration on a cluster already running Cloudera Runtime 7.1.9 SP1 CHF9, then you must restart all Cloudera Runtime services respectively after installing Python 3.11 so that they will pick up the Python 3.11 version. To restart all Cloudera Runtime services, restart the cluster. For information about restarting a cluster, see [Restarting a Cluster](#).
 - **Applicable for SLES 15 SP4, SLES 15 SP5, and Ubuntu 22.04:** Before updating the Cloudera Runtime to 7.1.9 SP1 CHF9 or higher versions, you must install Python 3.11 on all hosts before updating Cloudera Manager to 7.13.1.400 or higher versions.
2. Upgrade Cloudera Manager to 7.13.1 CHF4 version. For information about upgrading Cloudera Manager, see [Upgrading Cloudera Manager 7](#).
 3. Upgrade Cloudera Runtime to 7.1.9 SP1 CHF9 version. For information about upgrading Cloudera Runtime Cluster, see [Upgrading a Cloudera Runtime Cluster](#).

Migrating from Python 3.8 to Python 3.9 on RHEL 8, Oracle Linux 8, Rocky Linux 8 (Recommended) Installing Python 3.9 before upgrading Cloudera Manager to 7.13.1 CHF4 (7.13.1.400)

This is the recommended workflow for customers upgrading Cloudera Manager to 7.13.1 CHF4 (7.13.1.400) version on RHEL 8, Oracle Linux 8, Rocky Linux 8.

1. Install Python 3.9 from the standard repository. For information about installing Python 3.9 on supported operating systems, see [Installing Python 3.9 standard package on RHEL 8](#).



Important:

- Install Python 3.9 before upgrading Cloudera Manager to 7.13.1 CHF4 version to avoid performing a Cloudera Manager Agent restart later. If you install Python 3.9 after upgrading Cloudera Manager to 7.13.1 CHF4 version, then you must perform Cloudera Manager Agent restart after installing Python 3.9.
 - Similarly, if you perform the Python migration on a cluster already running Cloudera Runtime 7.1.9 SP1 CHF9, then you must restart all Cloudera Runtime services respectively after installing Python 3.9 so that they will pick up the Python 3.9 version. To restart all Cloudera Runtime services, restart the cluster. For information about restarting a cluster, see [Restarting a Cluster](#).
 - **Applicable for SLES 15 SP4, SLES 15 SP5, and Ubuntu 22.04:** Before updating the Cloudera Runtime to 7.1.9 SP1 CHF9 or higher versions, you must install Python 3.11 on all hosts before updating Cloudera Manager to 7.13.1.400 or higher versions.
2. Upgrade Cloudera Manager to 7.13.1 CHF4 version. For information about upgrading Cloudera Manager, see [Upgrading Cloudera Manager 7](#).
 3. Upgrade Cloudera Runtime to 7.1.9 SP1 CHF9 version. For information about upgrading Cloudera Runtime Cluster, see [Upgrading a Cloudera Runtime Cluster](#).

Migrating from Python 3.8 to Python 3.9 on RHEL 8.8 or RHEL 8.10



Important: The following steps are applicable only when using Cloudera Manager from version 7.13.1 to 7.13.1 CHF4 (7.13.1.400) with Cloudera Runtime 7.3.1.400 SP2 or lower versions.

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

(Recommended) Installing Python 3.9 on RHEL 8 before upgrading Cloudera Manager to 7.13.1 and Cloudera Runtime to 7.3.1

Learn how to migrate Cloudera Manager and Cloudera Runtime from Python 3.8 to Python 3.9 on RHEL 8.8 or RHEL 8.10. This is the recommended workflow for customers upgrading Cloudera Manager to 7.13.1 version and Cloudera Runtime to 7.3.1 version since the minimum recommended version of Python is now 3.9.

1. Install Python 3.9 from standard repository. For information about installing Python 3.9 on RHEL 8, see [Installing Python 3.9 standard package on RHEL 8](#).



Important:

- Install Python 3.9 before upgrading Cloudera Manager to 7.13.1 version to avoid performing a Cloudera Manager Agent restart later. If you install Python 3.9 after upgrading Cloudera Manager to 7.13.1 version, then you must perform Cloudera Manager Agent restart after installing Python 3.9.
 - Install Python 3.9 before upgrading Cloudera Runtime to 7.3.1 version as Hue requires Python 3.9 and does not start until you install Python 3.9. If you want to upgrade Cloudera Runtime to 7.3.1, then you must install Python 3.9 on all Hue servers before you upgrade Cloudera Runtime to 7.3.1 version.
2. Upgrade Cloudera Manager to 7.13.1 version. For information about upgrading Cloudera Manager, see [Upgrading Cloudera Manager 7](#).
 3. Upgrade Cloudera Runtime to 7.3.1 version. For information about upgrading Cloudera Runtime Cluster, see [Upgrading a Cloudera Runtime Cluster](#).
 4. **Optional:** Uninstall Python 3.8. To uninstall Python 3.8 run the following command:

```
yum remove python38
```

(Not Recommended) Installing Python 3.9 on RHEL 8 after upgrading Cloudera Manager to 7.13.1 and Cloudera Runtime to 7.3.1

Learn how to upgrade Cloudera Manager and Cloudera Runtime while delaying the upgrade of Python from 3.8 to 3.9 version on RHEL 8.8 or RHEL 8.10. This is not a recommended workflow due to Hue supports only Python 3.9 in Cloudera Runtime 7.3.1.



Important: Hue does not start until you install Python 3.9.

1. Upgrade Cloudera Manager to 7.13.1 version. For information about upgrading Cloudera Manager, see [Upgrading Cloudera Manager 7](#).
2. Upgrade Cloudera Runtime to 7.3.1 version. For information about upgrading Cloudera Runtime Cluster, see [Upgrading a Cloudera Runtime Cluster](#).



Warning:

Hue will fail to start. Continue the Cloudera Runtime Cluster upgrade process manually by following the manual upgrade instructions from [Manual upgrade to Cloudera Base on premises](#).

3. Install Python 3.9 from standard repository. For information about installing Python 3.9 on RHEL 8, see [Installing Python 3.9 standard package on RHEL 8](#).



Important: If you install Python 3.9 after upgrading Cloudera Manager to 7.13.1 version and Cloudera Runtime to 7.3.1 version, then you must restart the Cloudera Manager Agent and all Cloudera Runtime services respectively after installing Python 3.9.

4. Restart Cloudera Manager Agents. For information about restarting Cloudera Manager Agents, see [Starting, Stopping, and Restarting Cloudera Manager Agents](#).

After restarting Cloudera Manager Agents, Cloudera Manager Agent will pick up the highest Python version available between Python 3.8 and Python 3.9.

5. Restart all Cloudera Runtime services so that they will pick up the Python 3.9 version. To restart all Cloudera Runtime services, restart the cluster. For information about restarting a cluster, see [Restarting a Cluster](#).



Important: Hue starts now successfully.

6. **Optional:** Uninstall Python 3.8. To uninstall Python 3.8 run the following command:

```
yum remove python38
```

When Cloudera Manager is upgraded to 7.13.1 version and when Cloudera Runtime is either 7.1.8, 7.1.7 SP3 or 7.1.9 for RHEL 8, all the Cloudera Runtime process will continue to use Python 3.8, whereas Cloudera Manager Agent will pick up the highest Python version available between Python 3.8 and Python 3.9.

Cloudera Manager 7.13.1

Using Python 3 with the Cloudera Manager Agents

You must install Python 3 on all hosts before upgrading to Cloudera Manager 7.13.1. See [Installing Python 3](#).

Cloudera Manager 7.11.3

Using Python 3 with the Cloudera Manager Agents

You must install Python 3 on all hosts before upgrading to Cloudera Manager 7.11.3. See [Installing Python 3](#).

Cloudera Manager 7.7.3

Using Python 3.8 with the Cloudera Manager Agents

If you require Python 3.8 to be used on your cluster hosts, you must install Python 3.8 before you upgrade Cloudera Manager. See [Installing Python 3.8 for Cloudera Manager 7.7.3](#)

**Important:**

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-CHF4 supports only RHEL 8.4, RHEL 8.6, and SLES 15 SP4. Cloudera Manager 7.7.3 supports only RHEL 7.9, RHEL 8.4, and RHEL 8.6. For more information, see the [CDP Private Cloud Base Installation Guide](#).

Upgrading Cloudera Manager 7

Steps to upgrade Cloudera Manager.

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)



Important: To upgrade Cloudera Manager from any 5.x or 6.x version to a higher version of Cloudera Manager 5.x or 6.x, do not perform the following instructions on this page. Instead, use the instructions from the [Cloudera Enterprise Upgrade Guide](#).

**Important:**

From Cloudera Manager 7.13.1 onwards Cloudera Manager by default does not support Cloudera Runtime 6 cluster because CDH 6 jars have security vulnerabilities.

If you want to run Cloudera Manager 7.13.1 or higher versions with CDH 6 cluster then you need to install an additional cloudera-manager-daemons-cdh6 rpm/debian package on every cluster host after installation/upgrade to Cloudera Manager 7.13.1. This package is available along with the other Cloudera Manager packages on the public repository.



Important: To perform Cloudera Manager upgrade for the following upgrade paths, you must follow the upgrade instructions that are used for upgrading the Cloudera Manager from Cloudera Manager 7.7.1 to Cloudera Manager 7.11.3:

- Cloudera Manager 7.10.1 to Cloudera Manager 7.11.3 CHF4
- Cloudera Manager 7.9.5 to Cloudera Manager 7.11.3 CHF1

These topics describes how to upgrade Cloudera Manager to a higher version of Cloudera Manager 7.1 and higher, including major, minor, and maintenance releases. The upgrade procedures use operating system command-line package commands to upgrade Cloudera Manager, and then complete the upgrade using Cloudera Manager.

When you upgrade Cloudera Manager, you use RPM-based package commands to upgrade the software on the Cloudera Manager server host and then Cloudera Manager manages upgrading the Cloudera Manager Agents on the remaining managed hosts. Cloudera Manager can also automatically install some versions of the required JDK on the managed hosts.

Upgrades are not supported between all versions of Cloudera Manager, CDH, or Cloudera Runtime. See [Supported Upgrade Paths](#).

Cloudera Navigator is also upgraded when you upgrade Cloudera Manager 5.x or 6.x. Cloudera Navigator has been replaced by Apache Atlas as of Cloudera Runtime 7.0.3. If you are using Cloudera Manager 7.0.3 or higher to manage CDH clusters, those clusters can continue using Cloudera Navigator.

The Cloudera Manager upgrade process does the following:

- Upgrades the database schema to reflect the current version.
- Upgrades the Cloudera Manager Server and all supporting services.

- Upgrades the Cloudera Manager agents on all hosts.
- Redeploys client configurations to ensure that client services have the most current configuration.
- Upgrades Cloudera Navigator (for upgrades to Cloudera Manager 7.1, you can transition Cloudera Navigator to Apache Atlas).

To upgrade Cloudera Manager, you perform the following tasks:

1. Back up the Cloudera Manager server databases, working directories, and several other entities. These backups can be used to restore your Cloudera Manager deployment if there are problems during the upgrade.
2. Upgrade the Cloudera Manager server software on the Cloudera Manager host using package commands from the command line (for example, yum on RHEL systems). Cloudera Manager automates much of this process and is recommend for upgrading and managing your CDH/Cloudera Runtime clusters.
3. Upgrade the Cloudera Manager agent software on all cluster hosts. The Cloudera Manager upgrade wizard can upgrade the agent software (and, optionally, the JDK), or you can install the agent and JDK software manually. The CDH or Cloudera Runtime software is not upgraded during this process.

Upgrading Cloudera Manager does not upgrade Cloudera Runtime clusters. See [Upgrading a Cloudera Runtime Cluster](#) on page 104 for upgrade procedures.

Step 1: Getting Started Upgrading Cloudera Manager 7



Note: Not all combinations of Cloudera Manager and Cloudera Runtime are supported. Ensure that the version of Cloudera Manager you are using supports the version of Cloudera Runtime you have selected. For details, see [Cloudera Manager support for Cloudera Runtime, CDH and CDP Private Cloud Experiences](#).

Before you upgrade Cloudera Manager, you need to gather some information and review the limitations and release notes. Fill in the My Environment form below to customize your Cloudera Manager upgrade procedures. See the [Collect Information](#) section below for assistance in locating the required information.



Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported



Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher



Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.



Warning: Before upgrading CDH 5 clusters with Sentry to 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.



Important: If HDFS ACL sync is enabled (`hdfs_sentry_sync_enable=true`) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see [Installing Ranger RMS](#).



Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.



Warning: You cannot upgrade from a cluster that uses Oracle 12.



Warning: You cannot upgrade from a cluster that uses Oracle 19.

Collect Information

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Collect the following information about your environment and fill in the form above. This information will be remembered by your browser on all pages in this Upgrade Guide.

- a. The current version of the Operating System:

```
lsb_release -a
```

Database parameters:

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

```
...
com.cloudera.cmf.db.type=mysql
com.cloudera.cmf.db.host=DATABASE_HOSTNAME:DATABASE_PORT
com.cloudera.cmf.db.name=SCM
com.cloudera.cmf.db.user=SCM
com.cloudera.cmf.db.password=SOME_PASSWORD
```

- b. Log in to the Cloudera Manager Admin console and find the following:

1. The version of Cloudera Manager used in your cluster. Go to [Support About](#).
2. The version of the JDK deployed in the cluster. Go to [Support About](#).

Preparing to Upgrade Cloudera Manager

- Access to Cloudera Manager binaries for production purposes requires authentication. In order to download the software, you must first have an active subscription agreement and obtain a license key file along with the required authentication credentials (username and password). See [Cloudera Manager Download Information](#).
- You must have SSH access to the Cloudera Manager server hosts and be able to log in using the root account or an account that has password-less sudo permission for all hosts.
- Review the following when upgrading to Cloudera Manager 7.1 or higher:

[Cloudera Base on premises Requirements and Supported Versions](#)

- Before upgrading the Cloudera Manager, remove all Navigator role instances to avoid a restart loop. For more information, see [Removing Navigator role instances](#).
- You may be required to upgrade the operating system before upgrading. See [Operating System Requirements](#) to determine operating system support for the version of Cloudera Manager you are upgrading to. Depending on the support, you may need to upgrade the operating system.

If you must or choose to upgrade to a supported operating system, you must determine whether to upgrade the operating system first or Cloudera Manager first. If the current version of Cloudera Manager and the version you are upgrading to both support a newer version of the operating system but the new version of Cloudera Manager does not support the older operating system, you must upgrade to the newer operating system before upgrading Cloudera Manager. If this is not true, then you must upgrade Cloudera Manager before upgrading the operating system.

See [Upgrading the Operating System to a new Major Version](#) on page 10.

- Install a [supported version](#) of the Java Development Kit (JDK) on all hosts. If you are upgrading to Cloudera Manager and CDP Private Cloud Base 7.1.1 and higher, you can choose to install OpenJDK 1.8 instead of the Oracle JDK.

There two options for JDK installation:

- Manually install the Oracle JDK or OpenJDK on all hosts.
- Manually install the Oracle JDK 1.8 on the Cloudera Manager host, and then select the Install Oracle Java SE Development Kit checkbox when prompted while running the Cloudera Manager Upgrade wizard.

See [Upgrading the JDK](#) on page 28

- Review the Release Notes.
 - Cloudera Base on premises
 - [Cloudera Manager Release Notes](#)
 - [Cloudera Runtime Release Notes](#)
- Review the [Cloudera Security Bulletins](#).
- The embedded PostgreSQL database installed with the [Trial Installer](#) is not supported in production environments because a trial installation cannot easily be upgraded, backed up, or migrated into a production-ready configuration without manual steps requiring down time.

Consider [Migrating from the Cloudera Manager Embedded PostgreSQL Database Server to an External PostgreSQL Database](#) before upgrading Cloudera Manager.

- If your cluster uses Oracle for any databases, before upgrading CDH 5, 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 effect this change can have on your system.

Step 2: Backing Up Cloudera Manager 7

Cloudera recommends that you backup Cloudera Manager before upgrading.

This topic contains procedures to back up Cloudera Manager. Cloudera recommends that you perform these backup steps before upgrading. The backups will allow you to rollback your Cloudera Manager upgrade if needed.



Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported



Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher



Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.



Warning: Before upgrading CDH 5 clusters with Sentry to 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.



Important: If HDFS ACL sync is enabled (`hdfs_sentry_sync_enable=true`) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see [Installing Ranger RMS](#).



Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

Collect Information for Backing Up Cloudera Manager

Information you should collect before backing up Cloudera Manager.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Collect database information by running the following command:

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

For example:

```
...
com.cloudera.cmf.db.type=...
com.cloudera.cmf.db.host=DATABASE_HOSTNAME:DATABASE_PORT
com.cloudera.cmf.db.name=SCM
com.cloudera.cmf.db.user=SCM
com.cloudera.cmf.db.password=SOME_PASSWORD
```

3. Collect information (host name, port number, database name, user name and password) for the following databases.
 - Reports Manager

You can find the database information by using the Cloudera Manager Admin Console. Go to **Clusters Cloudera Management Service Configuration** and select the Database category. You may need to contact your database administrator to obtain the passwords.

4. Find the host where the Service Monitor, Host Monitor and Event Server roles are running. Go to **Clusters Cloudera Management Service Instances** and note which hosts are running these roles.

Back Up Cloudera Manager Agent



Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups, defined by the environment variable `CM_BACKUP_DIR`, which is used in all the backup commands. You may change these destination paths in the command as needed for your deployment.

The tar commands in the steps below may return the following message. It is safe to ignore this message:

```
tar: Removing leading `/' from member names
```

Backup up the following Cloudera Manager agent files on all hosts:

- Create a top level backup directory.

```
export CM_BACKUP_DIR="`date +%F`-CM"
echo $CM_BACKUP_DIR
mkdir -p $CM_BACKUP_DIR
```

- Back up the Cloudera Manager Agent directory and the runtime state.

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-agent.tar --exclude=*.sock /etc/cloudera-scm-agent /etc/default/cloudera-scm-agent /var/run/cloudera-scm-agent /var/lib/cloudera-scm-agent
```

- Back up the existing repository directory.

RHEL / CentOS

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/yum.repos.d
```

SLES

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/zypp/repos.d
```

Ubuntu

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/apt/sources.list.d
```

Back Up the Cloudera Management Service



Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups. You may change these destination paths in the command as needed for your deployment.

1. Stop the Cloudera Management Service.
 - a. Log in to the Cloudera Manager Admin Console.
 - b. Go to Cloudera Management Service .
 - c. Select Actions Stop .
2. On the host where the Service Monitor role is configured to run, backup the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-SERVICE-MONITOR /var/lib/cloudera-service-monitor-`date +%F`-CM
```

3. On the host where the Host Monitor role is configured to run, backup the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-HOST-MONITOR /var/lib/cloudera-host-monitor-`date +%F`-CM
```

4. On the host where the Event Server role is configured to run, back up the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-SCM-EVENTSERVER /var/lib/cloudera-scm-eventserver-`date +%F`-CM
```

5. Restart the Cloudera Management Service if you are not upgrading Cloudera Manager immediately.
 - a. Log in to the Cloudera Manager Admin Console.
 - b. Select Clusters Cloudera Management Service .
 - c. Select Actions Start .

Stop Cloudera Manager Server & Cloudera Management Service

1. Stop the Cloudera Management Service.
 - a. Log in to the Cloudera Manager Admin Console.
 - b. Select Clusters Cloudera Management Service .
 - c. Select Actions Stop .
2. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

3. Stop the Cloudera Manager Server.

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

Back Up the Cloudera Manager Databases

During the upgrade, Cloudera Manager modifies the schema of the Cloudera Manager database. In case of failures during the upgrade, it may be necessary to rollback to the previous version of Cloudera Manager while addressing the upgrade failures.

When performing a rollback to a previous version, the Cloudera Manager Database must be restored to the previous database schema. For this reason, you must do the Cloudera Manager database backup, so that it can be restored if a rollback is necessary.

**Tip:**

You can get the name, user, and password properties for the Cloudera Manager database from the `/etc/cloudera-scm-server/db.properties` file:

```
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=NnYfWIj1bk
```

1. Back up the Cloudera Manager server database as per the instructions provided in your respective database documentation on how to backup and restore the databases.

2. Back up All other Cloudera Manager databases - Use the database information that [you collected in a previous step](#). You may need to contact your database administrator to obtain the passwords.

These databases can include the following:

- 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 database. If this information is not available, your cluster cannot start or function correctly. You must schedule and maintain regular backups of the database to recover the cluster in the event of the loss of this database.

- Oozie Server - Contains Oozie workflow, coordinator, and bundle data. Can grow very large. (Only available when installing CDH 5 or CDH 6 clusters.)
- Sqoop Server - Contains entities such as the connector, driver, links and jobs. Relatively small. (Only available when installing CDH 5 or CDH 6 clusters.)
- Reports Manager - Tracks disk utilization and processing activities over time. Medium-sized.
- Hive Metastore Server - Contains Hive metadata. Relatively small.
- Hue Server - Contains user account information, job submissions, and Hive queries. Relatively small.
- Sentry Server - Contains authorization metadata. Relatively small.
- Navigator Audit Server - Contains auditing information. In large clusters, this database can grow large. (Only available when installing CDH 5 or CDH 6 clusters.)
- Navigator Metadata Server - Contains authorization, policies, and audit report metadata. Relatively small. (Only available when installing CDH 5 or CDH 6 clusters.)
- DAS PostgreSQL server - Contains Hive and Tez event logs and DAG information. Can grow very large.
- Ranger Admin - Contains administrative information such as Ranger users, groups, and access policies. Medium-sized.
- Streaming Components:
 - Schema Registry - Contains the schemas and their metadata, all the versions and branches. You can use either MySQL, Postgres, or Oracle.



Important: For the Schema Registry database, you must set collation to be case sensitive.

- Streams Messaging Manager Server - Contains Kafka metadata, stores metrics, and alert definitions. Relatively small.

For more information about the number of databases that should be backed up, and restored if necessary, see [Required Databases](#).

Back Up Cloudera Manager Server



Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups, defined by the environment variable CM_B ACKUP_DIR, which is used in all the backup commands. You may change these destination paths in the command as needed for your deployment.

The tar commands in the steps below may return the following message. It is safe to ignore this message:

```
tar: Removing leading `/' from member names
```

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Create a top-level backup directory.

```
export CM_BACKUP_DIR=`date +%F`-CM"
echo $CM_BACKUP_DIR
mkdir -p $CM_BACKUP_DIR
```

3. a. Back up the Cloudera Manager Server directories along with the CSP key file when Credential Storage Provider (CSP) is enabled:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-scm-server /etc/default/cloudera-scm-server ***/path/to/csp/keys***
```



Important: In the above command, you must replace the `***path/to/csp/key***` with the actual path where the CSP key has been stored from the following options:

- `/var/lib/cloudera-scm-server/csp-data`
- `/opt/cloudera/`
- `/etc/cloudera/`
- `/var/cloudera/`

- b. Back up the Cloudera Manager Server directories without CSP key file:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-scm-server /etc/default/cloudera-scm-server
```

4. Back up the existing repository directory.

RHEL / CentOS

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/yum.repos.d
```

SLES

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/zypp/repos.d
```

Ubuntu

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/apt/sources.list.d
```

(Optional) Start Cloudera Manager Server & Cloudera Management Service

Start the Cloudera Manager server and Cloudera Management Service.

If you will be immediately upgrading Cloudera Manager, skip this step and continue with [Step 3: Upgrading the Cloudera Manager Server](#) on page 53.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Start the Cloudera Manager Server.

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

3. Start the Cloudera Management Service.

- a. Log in to the Cloudera Manager Admin Console.
- b. Select Clusters Cloudera Management Service .
- c. Select Actions Start .

Step 3: Upgrading the Cloudera Manager Server

You can also use the procedures on this page to install Cloudera Manager patches. You must obtain the download URL for the patch before proceeding – you will use this information later in this procedure.



Important: Upgrades to 7.0.3 are not supported.



Note: The embedded PostgreSQL version within Key Trustee Server is upgraded from 14.2 to 14.16 in 7.1.9 SP1 CHF10. If you are using KTS, you need to upgrade your to 7.11.3 CHF17 first in order to upgrade to 7.1.9 SP1 CHF10. Upgrading from CDH6 to a 7.1.9 SP1 CHF is supported with 7.11.3 CHF11, and not supported with 7.11.3 CHF12 and later versions.



Note: Upgrades from CDH 6.x are supported only for upgrades to Cloudera Manager 7.4.4 or higher and Cloudera Runtime 7.1.7 or higher. Upgrades from CDH 6.0 are not supported. For a full list of supported upgrades, see [Upgrade Paths](#).



Warning: Upgrading to Cloudera Manager version 7.7.1 and above introduces a new service called CORE_SETTINGS-2 which leads to stale configurations on different services in the cluster.

This topic provides procedures for backing up the Cloudera Manager Server.

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)

After you complete the steps in [Step 1: Getting Started Upgrading Cloudera Manager 7](#) on page 45 and [Step 2: Backing Up Cloudera Manager 7](#) on page 47, continue with the following:



Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported



Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher



Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.



Warning: Before upgrading CDH 5 clusters with Sentry to 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.



Important: If HDFS ACL sync is enabled (hdfs_sentry_sync_enable=true) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see [Installing Ranger RMS](#).



Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.



Important: Please note the following:

- A valid Enterprise license file and a username and password are required to download and install the software. You can obtain the username and password from the [Cloudera CDH Download](#) page. See Your license file must be current and uploaded to .

To upload a license:

1. Download the license file and save it locally.
 2. In , go to the Home page.
 3. Select Administration License .
 4. Click Upload License.
 5. Browse to the license file you downloaded.
 6. Click Upload.
- If you are using Express, you cannot upgrade or CDH.
 - Several steps in the procedures have changed and now require the username and password.
 - Download URLs have changed.



Important: If you encounter problems, see the following:

- [Troubleshooting a Cloudera Manager Upgrade](#) on page 79
- [Reverting a Failed Cloudera Manager Upgrade](#) on page 80

Establish Access to the Software

Cloudera Manager needs access to a package repository that contains the updated software packages. You can choose to access the Cloudera public repositories directly, or you can [download those repositories and set up a local repository](#) to access them from within your network. If your cluster hosts do not have connectivity to the Internet, you must set up a local repository.

If you have enabled high availability for Cloudera Manager, perform the following steps on the hosts for both the active and passive instances of the Cloudera Manager server.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Remove any older files in the existing repository directory:

RHEL / CentOS

```
sudo rm /etc/yum.repos.d/cloudera*manager.repo*
```

SLES

```
sudo rm /etc/zypp/repos.d/cloudera*manager.repo*
```

Ubuntu

```
sudo rm /etc/apt/sources.list.d/cloudera*.list*
```

3. Fill in the form at the top of this page.

4. Create a repository file so that the package manager can locate and download the binaries.



Note: If you are upgrading to a Cloudera Manager patch, substitute the download URL for the patch when creating the repository file (cloudera-manager.repo or cloudera_manager.list).

Do one of the following, depending on whether or not you are using a local package repository:

- Use a local package repository. (Required when cluster hosts do not have access to the internet.) See [Configuring a Local Package repository](#).
- Use the Cloudera public repository

RHEL / CentOS

- a. Create a file named /etc/yum.repos.d/cloudera-manager.repo with the following content:

```
[cloudera-manager]
name=Cloudera Manager
baseurl=https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
VERSION>/redhat<OS MAJOR VERSION>/yum/
gpgkey=https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
VERSION>/redhat<OS MAJOR VERSION>/yum/RPM-GPG-KEY-cloudera
username=CHANGEME
password=CHANGEME
gpgcheck=1
enabled=1
autorefresh=0
type=rpm-md
```

Replace *CHANGEME* with your *USERNAME* and *PASSWORD* in the /etc/yum.repos.d/cloudera-manager.repo file.

SLES

- a. Create a file named /etc/zypp/repos.d/cloudera-manager.repo with the following content:

```
[cloudera-manager]
name=Cloudera Manager
baseurl=https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
VERSION>/sles<OS MAJOR VERSION>/yum/
gpgkey=https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
VERSION>/sles<OS MAJOR VERSION>/yum/RPM-GPG-KEY-cloudera
username=CHANGEME
password=CHANGEME
gpgcheck=1
enabled=1
autorefresh=0
type=rpm-md
```

- b. Replace *CHANGEME* with your *USERNAME* and *PASSWORD* in the /etc/zypp/repos.d/cloudera-manager.repo file.

Ubuntu

Debian is not a supported operating system for Cloudera Manager 6.x.

- a. Create a file named /etc/apt/sources.list.d/cloudera_manager.list with the following content:

```
# Cloudera Manager <CLOUDERA MANAGER VERSION>
deb [arch=amd64]
http://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/<CLOUDERA
MANAGER VERSION>/ubuntu1804/apt -cm<CLOUDERA MANAGER VERSION>
contrib
```

- b. Run the following command:

```
sudo apt-get update
```

- c. Replace *CHANGEME* with your *USERNAME* and *PASSWORD* in the `/etc/apt/sources.list.d/cloudera_manager.list` file.



Tip: If you have a mixed operating system environment, adjust the Operating System filter at the top of the page for each operating system. The guide will generate the repo file for you automatically here.

5. A Cloudera Manager upgrade can introduce new package dependencies. Your organization may have restrictions or require prior approval for installation of new packages. You can determine which packages may be installed or upgraded:

RHEL / CentOS

```
yum deplist cloudera-manager-agent
```

SLES

```
zypper info --requires cloudera-manager-agent
```

Ubuntu

```
apt-cache depends cloudera-manager-agent
```

Install Java (JDK)

Oracle JDK 1.8 is required on all cluster hosts managed by Cloudera Manager 6.0.0 or higher. If it is [supported for your version of Cloudera Manager](#), you can also install OpenJDK 8*, OpenJDK 11*, or OpenJDK 17*. See [Manually Installing OpenJDK](#). If OpenJDK 8* is already installed on your hosts, skip the steps in this section.

If you are upgrading to Cloudera Manager 6.0.0 or higher, you can manually install JDK 8 on the Cloudera Manager server host, and then, as part of the Cloudera Manager upgrade process, you can specify that Cloudera Manager upgrade the JDK on the remaining hosts.

A supported JDK is required on all hosts. During a Cloudera Manager upgrade, you can install OpenJDK 8* on the Cloudera Manager server host, and then Cloudera Manager can install the new JDK on the managed hosts. You can also choose to install Oracle JDK 8, OpenJDK 8*, or OpenJDK 11* manually, on all hosts before beginning the upgrade.



Note: Cloudera Manager no longer installs Oracle JDKs.

A supported JDK is required on all hosts. During a Cloudera Manager upgrade, you can install OpenJDK 8* on the Cloudera Manager server host, and then Cloudera Manager can install the new JDK on the managed hosts. You can also choose to install Oracle JDK 8, OpenJDK 8*, OpenJDK 11*, or OpenJDK 17* manually, on all hosts before beginning the upgrade.



Note: Cloudera Manager no longer installs Oracle JDKs.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

* Azul OpenJDK, OpenJDK 8, OpenJDK 11, and OpenJDK 17 are TCK certified for Cloudera.

2. Stop the Cloudera Manager Server.

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

3. Remove the JDK:

- a. Perform the following steps on all hosts managed by :

1. Run the following command to remove the JDK, using the package names from Step 1: (If you do not delete these files, and other components may continue to use the old version of the JDK.)

RHEL

```
yum remove <JDK PACKAGE NAME>
```

Ubuntu

```
apt-get remove <JDK PACKAGE NAME>
```

SLES

```
zypper rm <JDK PACKAGE NAME>
```

2. Confirm that the package has been removed:

RHEL

```
yum list installed |grep -i java
```

Ubuntu

```
apt list --installed | grep -i java
```

SLES

```
zypper search --installed-only |grep -i java
```

4. Install OpenJDK

RHEL

OpenJDK 8*

```
sudo yum install java-1.8.0-openjdk-devel
```

OpenJDK 11*

```
sudo yum install java-11-openjdk-devel
```

OpenJDK 17*

```
sudo yum install java-17-openjdk-devel
```

Ubuntu

OpenJDK 8*

```
sudo apt-get install openjdk-8-jdk
```

OpenJDK 11*

```
sudo apt install openjdk-11-jdk
```

OpenJDK 17*

```
sudo apt install openjdk-17-jdk
```

SLES

OpenJDK 8*

```
sudo zypper install java-1_8_0-openjdk-devel
```

OpenJDK 11*

```
zypper install java-11-openjdk-devel
```

OpenJDK 17*

```
sudo zypper --non-interactive install java-17-openjdk-devel
```

5. Start the Cloudera Manager Server.

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

Upgrade the Cloudera Manager Server

1. Log in to the Cloudera Manager server host.
2. If your cluster is running the embedded PostgreSQL database, stop all services that are using the embedded database. These can include:
 - Hive service and all services such as Impala and Hue that use the Hive metastore
 - Oozie
 - Sentry
 - Sqoop

3. Stop the Cloudera Management Service.

- a. Log in to the Cloudera Manager Admin Console.
- b. Select Clusters Cloudera Management Service .
- c. Select Actions Stop .



Important: Not stopping the Cloudera Management Service at this point might cause management roles to crash or the Cloudera Manager Server might fail to restart.

4. Ensure that you have disabled any scheduled replication or snapshot jobs and wait for any running commands from the Cloudera Manager Admin Console to complete before proceeding with the upgrade.



Important: If there are replication jobs, snapshot jobs, or other commands running when you stop Cloudera Manager Server, Cloudera Manager Server might fail to start after the upgrade.

5. If you have any Hive Replication Schedules that replicate to a cloud destination, delete these replication clusters before continuing with the upgrade. You can re-create these Replication Schedules after the Cloudera Manager upgrade is complete.
6. If your cluster is running Ubuntu version 18, stop all clusters before upgrading Cloudera Manager. (For each cluster, go to *CLUSTER NAME* Actions Stop .)
7. Stop the Cloudera Manager server host, agent, and embedded database (if installed). If you have enabled high availability for Cloudera Manager, perform the following steps on the hosts for both the active and passive instances of Cloudera Manager:

- a. Log in to the Cloudera Manager Server host:

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

- b. Stop the Cloudera Manager Server.

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

- c. If you are using the embedded PostgreSQL database, stop the Cloudera Manager Embedded PostgreSQL database:

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

If you are not using the embedded PostgreSQL database and you attempt to stop it, you might see a message indicating that the service cannot be found.

In case of using the embedded PostgreSQL, if you see a message that the shutdown failed, then the embedded database is still running, probably because services are connected to the Hive metastore. If the database shutdown fails due to connected services, issue the following command:

```
sudo service cloudera-scm-server-db next_stop_fast
sudo service cloudera-scm-server-db stop
```

- d. Stop the Cloudera Manager Agent.

```
sudo systemctl stop cloudera-scm-agent
```

8. Upgrade the Cloudera Manager software. If you have enabled high availability for Cloudera Manager, perform the following steps on the hosts for both the active and passive instances of the Cloudera Manager server.

- a. Upgrade the packages. Include the cloudera-manager-server-db-2 package in the command only if you are using the embedded PostgreSQL database.

RHEL / CentOS

```
sudo yum clean all
```

```
sudo yum upgrade cloudera-manager-server cloudera-manager-daemons
cloudera-manager-agent cloudera-manager-server-db-2
```

SLES

```
sudo zypper clean --all
sudo zypper up cloudera-manager-server cloudera-manager-daemons
cloudera-manager-agent cloudera-manager-server-db-2
```

Ubuntu

```
sudo apt-get clean
```

```
sudo apt-get update
```

```
sudo apt-get dist-upgrade
```

```
sudo apt-get install cloudera-manager-server cloudera-manager-daemons
cloudera-manager-agent cloudera-manager-server-db-2
```

You might be prompted about your configuration file version:

```
Configuration file '/etc/cloudera-scm-agent/config.ini'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ? Your options are:
Y or I : install the package maintainer's version
N or O : keep your currently-installed version
D : show the differences between the versions
Z : start a shell to examine the situation
The default action is to keep your current version.
```

You may receive a similar prompt for `/etc/cloudera-scm-server/db.properties`. Answer N to both prompts.

You may be prompted to accept the GPG key. Answer y.

```
Retrieving key from https://archive.cloudera.com/.../cm/RPM-GPG-KEY-cloudera
Importing GPG key ...
Userid      : "Yum Maintainer <webmaster@cloudera.com>"
Fingerprint: ...
From        : https://archive.cloudera.com/.../RPM-GPG-KEY-cloudera
```



Note: If you receive the following error message when running these commands: [Errno 14] HTTP Error 404 - Not Found, make sure the URL in the `cloudera-manage.list cloudera-manager.repo` file is correct and is reachable from the Cloudera Manager server host.

- b. If you customized the `/etc/cloudera-scm-agent/config.ini` file, your customized file is renamed with the extension `.rpmsave` or `.dpkg-old`. Merge any customizations into the `/etc/cloudera-scm-agent/config.ini` file that is installed by the package manager.
- c. Verify that you have the correct packages installed.

Ubuntu

```
dpkg-query -l 'cloudera-manager-*'
```

```

Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aW
ait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                               Version                               Description
++-+-----+-----+-----+-----+-----+-----+-----+
=====
ii cloudera-manager-agent 5.15.0-0.cm...~sq The Cloudera Manager
   Agent
ii cloudera-manager-daemo 5.15.0-0.cm...~sq Provides daemons for
   monitoring Hadoop and related tools.
ii cloudera-manager-serve 5.15.0-0.cm...~sq The Cloudera Manager
   Server

```

RHEL / CentOS / SLES

```
rpm -qa 'cloudera-manager-*
```

```

cloudera-manager-server-5.15.0-...
cloudera-manager-agent-5.15.0-...
cloudera-manager-daemons-5.15.0-...
cloudera-manager-server-db-2-5.15.0-...

```

- d. If you are using the embedded PostgreSQL database, start the database:

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

9. Start the Cloudera Manager agent and server. If you have enabled high availability for Cloudera Manager, start only the host for the active instance of Cloudera Manager:

- a. Start the Cloudera Manager Agent.

```
sudo systemctl start cloudera-scm-agent
```

If the agent starts without errors, no response displays.

- b. The Cloudera Manager server now requires 4GB of heap. On the Cloudera Manager server host, edit the `/etc/default/cloudera-scm-server` file and change the line that begins with `export CMF_JAVA_OPTS=`. Change the `-Xmx2G` parameter to `-Xmx4G`.
- c. Start the Cloudera Manager Server.

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

- d. If you have problems starting the server or the agent, such as database permissions problems, you can use log files to troubleshoot the problem:

Server log:

```
tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log
```

Agent log:

```
tail -f /var/log/cloudera-scm-agent/cloudera-scm-agent.log
```

or

```
tail -f /var/log/messages
```

10. Verify that all cluster hosts appear in the Cloudera Manager Admin Console.

- a. Use a Web browser to open the Cloudera Manager Admin Console using the following URL:

```
http://CLUSTER_MANAGER_SERVER_HOSTNAME:7180/cm/upgrade
```

It can take several minutes for the Cloudera Manager Server to start, and the Cloudera Manager Admin Console is unavailable until the server startup is complete and the Upgrade Cloudera Manager page displays.

- b. Verify that all cluster hosts are visible. (Go to Hosts All Hosts.)

11. If you have enabled high availability for Cloudera Manager, start the host for the passive instance of Cloudera Manager:

- a. Start the Cloudera Manager Agent.

```
sudo systemctl start cloudera-scm-agent
```

If the agent starts without errors, no response displays.

- b. The Cloudera Manager server now requires 4GB of heap. On the Cloudera Manager server host, edit the `/etc/default/cloudera-scm-server` file and change the line that begins with `export CMF_JAVA_OPTS=`. Change the `-Xmx2G` parameter to `-Xmx4G`.
- c. Start the Cloudera Manager Server.

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

- d. If you have problems starting the server or the agent, such as database permissions problems, you can use log files to troubleshoot the problem:

Server log:

```
tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log
```

Agent log:

```
tail -f /var/log/cloudera-scm-agent/cloudera-scm-agent.log
```

or

```
tail -f /var/log/messages
```

To complete the Cloudera Manager upgrade, continue with [Step 4: Upgrading the Cloudera Manager Agents](#) on page 62.

Step 4: Upgrading the Cloudera Manager Agents

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)



Important: Upgrades to 7.0.3 are not supported.



Note: The embedded PostgreSQL version within Key Trustee Server is upgraded from 14.2 to 14.16 in 7.1.9 SP1 CHF10. If you are using KTS, you need to upgrade your to 7.11.3 CHF17 first in order to upgrade to 7.1.9 SP1 CHF10. Upgrading from CDH6 to a 7.1.9 SP1 CHF is supported with 7.11.3 CHF11, and not supported with 7.11.3 CHF12 and later versions.



Important: If you encounter problems, see the following:

- [Troubleshooting a Cloudera Manager Upgrade](#) on page 79



Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported



Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher



Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.



Warning: Before upgrading CDH 5 clusters with Sentry to 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.



Important: If HDFS ACL sync is enabled (`hdfs_sentry_sync_enable=true`) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see [Installing Ranger RMS](#).



Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

Upgrade the Cloudera Manager Agents (Cloudera Manager 7.0.3 and higher)

1. Ensure that the `ptrace_scope` operating system control is set to 0:

The Cloudera Manager Agent installation process uses a re-parenting mechanism to ensure that running Cloudera Services are not impacted by the Cloudera Manager Agent upgrade. This re-parenting mechanism utilizes the Linux kernel's `ptrace` capability. If the `ptrace_scope` system control is set to a non-zero value, then the installer will not be able to re-parent running Cloudera services. The Cloudera Manager Agent RPM will refuse to install if the `ptrace_scope` control has a non-zero value. For more information, see <https://www.kernel.org/doc/Documentation/security/Yama.txt>

- Verify that the value of `ptrace_scope` is set to zero. Run the following command to check the value:

```
cat /proc/sys/kernel/yama/ptrace_scope
```

- If the value is not set to 0, set the value to 0 by running the following command on all cluster hosts:

```
echo 0 >> /proc/sys/kernel/yama/ptrace_scope
```

If you do not want to allow `ptrace_scope` to run, run the following command on all cluster hosts to force the Cloudera Manager Agent upgrade:

```
touch /tmp/CLOUDERA_SKIP_PTRACE_CHECK_ON_UPGRADES
```

- If necessary, you can set `ptrace_scope` to its original value after the agent upgrades are complete.



Note: Ubuntu 18 and Ubuntu 20 set `ptrace_scope` to a non-zero value by default.

- After upgrading and starting the Cloudera Manager server, open the Cloudera Manager Admin Console (if you have not already done so) using the following URL:

```
https://CLOUDERA_MANAGER_SERVER_HOSTNAME:7183/cm/upgrade
```

The Upgrade Cloudera Manager screen displays:

Upgrade Cloudera Manager

✓ The Cloudera Manager Server is now running **7.1.1**

- JDK Version: 1.8.0_162
- [Release Notes](#)

This wizard helps you upgrade the Cloudera Manager agents, configure and start the Cloudera Management Service as described in [Upgrade Cloudera Manager](#) documentation. By proceeding, you agree to the [Terms and Conditions](#) and [Privacy Policy](#).

✓ Upgraded Cloudera Manager Agents

Hosts	Count	Operating System	Java Home Directory	Agent Version
er0519a-1.er0519a.root.hwx.site	1	centos 7.5.1804	/usr/java/jdk1.8.0_162-cloudera	7.1.1 (#3243646)

You need to ensure that a [Supported JDK](#) is installed on all hosts. If a specific JDK is preferred, then you should configure the **Java Home Directory** property for all hosts to ensure that the correct JDK is used. [Configure Java Home Directory](#)

⚠ Cloudera Manager Agents not upgraded

Hosts	Count	Operating System	Java Home Directory	Agent Version
er0519a-[2-4].er0519a.root.hwx.site	3	centos 7.5.1804	/usr/java/jdk1.8.0_162-cloudera	5.16.2 (#1.cm5162.p0.7)

Upgrade Cloudera Manager Agent Packages

When running the Upgrade Wizard to upgrade Cloudera Manager Agent packages, the wizard can also install JDK 1.8 for you.

⚠ Host Inspector

You must run the Host Inspector on this cluster; the results are valid for two days. Once the inspection is complete, review the inspector results before upgrading.

[Run Host Inspector](#) ☐ Skip this step. I understand the risks.

⚠ Start Cloudera Management Service

[Start Cloudera Management Service](#)

- Click Upgrade Cloudera Manager Agent packages

The Upgrade Cloudera Manager Agent Packages page displays the Select Repository step.

- Select one of the following:

- Select Public Cloudera Repository if the Cloudera Manager server host has access to the internet.
- Select the Custom Repository If you are using a [local package repository](#) instead of the public repository at <https://archive.cloudera.com>, option and enter the Custom Repository URL.

- Click Continue.

- The Select JDK screen displays the available options for the JDK used in the cluster. Choose one of the following options to install a JDK:

- Manually Manage JDK – Select this option if you have already installed a supported JDK. For information on installing a JDK, see [Upgrading the JDK](#) on page 28.
- Install a Cloudera-provided version of OpenJDK – Cloudera Manager installs OpenJDK 8 on all your cluster hosts, except for the Cloudera Manager server host(s).
- Install a system-provided version of OpenJDK – Cloudera Manager installs the default version of OpenJDK provided by the host operating system.

- Click Continue.

The Enter Login Credentials page displays.

8. Specify the credentials and initiate Agent installation:


- a.** Select root for the root account, or select Another user and enter the username for an account that has password-less sudo permission.
- b.** Select an authentication method:
 - If you choose the All hosts accept same password option, enter and confirm the password.
 - If you choose the All hosts accept same private key option, provide a passphrase and path to the required key files.
- c.** Modify the default SSH port if necessary.
- d.** Specify the maximum Number of Simultaneous Installations to run at once. The default and recommended value is 10. Adjust this parameter based on your network capacity.


9. Click Continue.

The Cloudera Manager Agent packages and, if selected, the JDK are installed.

10. When the installations are complete, click Finish.

The Upgrade Cloudera Manager page displays the status of the upgrade. If you see a message listing Cloudera Manager Agents not upgraded, wait a few minutes for the agents to heartbeat and then click the Refresh button.

 Cloudera Manager Agents not upgraded
 Refresh

Hosts	Count	Operating System	Java Home Directory	Agent Version
 er0519a-[2, 4].er0519a.root.hwx.site	2	These hosts did not heartbeat in the last two minutes. They may be upgraded a few seconds ago, so click the Refresh button to check the status again. You can upgrade them from this page later if they are offline temporarily right now.		

If some hosts do not report a heartbeat, you must upgrade the Cloudera Manager Agents manually. Do the following on these hosts:

- Ensure that the hosts have access to the package repositories. See [Configure a Repository for Cloudera Manager](#).
- Run the following commands on all affected hosts to remove the cloudera-manager-agent, cloudera-manager-daemons, and cloudera-manager-server packages and install the agent and daemons. (Omit cloudera-manager-server on the Cloudera Manager server host):

Operating System	Command
RHEL	<pre>sudo yum remove cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server</pre> <pre>sudo yum clean all</pre> <pre>sudo yum install cloudera-manager-agent cloudera-manager-daemons</pre>
SLES	<pre>sudo zypper remove cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server</pre> <pre>sudo zypper refresh -s</pre> <pre>sudo zypper install cloudera-manager-agent cloudera-manager-daemons</pre>
Ubuntu or Debian	<pre>sudo apt-get purge cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server</pre> <pre>sudo apt-get update</pre> <pre>sudo apt-get install cloudera-manager-agent cloudera-manager-daemons</pre>

- Copy the agent's config.ini file from the backup taken before upgrading to /etc/cloudera-scm-agent/config.ini. (See [Back Up Cloudera Manager Agent](#) on page 48)
- Restart the agents:

```
sudo systemctl stop cloudera-scm-supervisord.service
sudo systemctl start cloudera-scm-agent
```

- After the Agents are all upgraded, Click Run Host Inspector to run the host inspector. Inspect the output and correct any warnings. If problems occur, you can make changes and then rerun the inspector.
- When you are satisfied with the inspection results, click Start the Cloudera Management Service.
- Confirm that you want to start the Cloudera Management Service by clicking Continue.

14. After the Cloudera Management Service has started, click Finish.

You will see a message indicating that the Cloudera Management Service has started.

The upgrade is now complete.

15. Click the Home Page link to return to the Home page. Review and fix any critical configuration issues. You may need to restart any clusters if they indicate stale configurations.

To return to the Upgrade Cloudera Manager page, go to HostsAll HostsReview Upgrade Status.

16. If you stopped any clusters before upgrading Cloudera Manager, start them now. (For each cluster, go to *CLUSTER NAME*ActionsStart.)
17. If you set `ptrace_scope` to 0 and want to use the original or a different value, you can reset it by running the following command on all hosts:

```
echo [NEW_VALUE] >> /proc/sys/kernel/yama/ptrace_scope
```

18. If you have the Cloudera Embedded Container Service (ECS) deployed in any clusters, do the following
 - a. Restart the ECS Cluster. Go to the ECS cluster, click the actions menu and select Restart.
 - b. Unseal the Vault. Go to the ECS service and click Actions Unseal .

Step 5: After You Upgrade Cloudera Manager

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)



Note: [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-CHF4 supports only RHEL 8.4, RHEL 8.6, and SLES 15 SP4. See the [CDP Private Cloud Base Installation Guide](#) for more information.



Warning: Upgrades from Cloudera Manager 5.12 and lower to Cloudera Manager 7.1.1 or higher are not supported



Important: Upgrading Cloudera Manager to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade Cloudera Manager to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade Cloudera Manager to version 7.6.5 or higher



Warning: For upgrades from CDH clusters with Cloudera Navigator to Cloudera Runtime 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.



Warning: Before upgrading CDH 5 clusters with Sentry to 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.




Important: If HDFS ACL sync is enabled (`hdfs_sentry_sync_enable=true`) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see [Installing Ranger RMS](#).



Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. Cloudera recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

Perform Post-Upgrade Steps

1. If you upgraded the JDK, do the following:
 - a. If the Cloudera Manager Server host is also running a Cloudera Manager Agent, restart the Cloudera Manager Server:
 - b. Restart the Cloudera Manager Server.

```
sudo systemctl restart cloudera-scm-server
```
 - c. Open the Cloudera Manager Admin Console and set the Java Home Directory property in the host configuration:
 1. Go to HomeAll HostsConfiguration.
 2. Set the value to the path to the new JDK.
 3. Click Save Changes.
 - d. Restart all services:
 1. On the HomeStatus tab, click  next to the cluster name, select Restart and confirm.
2. Start the Cloudera Management Service and adjust any configurations when prompted.
 - a. Log in to the Cloudera Manager Admin Console.
 - b. Select Clusters Cloudera Management Service .
 - c. Select Actions Start .

3. Add and configure the Stub DFS service and reset the dependent services:

- a. Add the Stub DFS service to each cluster managed by Cloudera Manager that contains the services returned by the query you ran when upgrading the Cloudera Manager server. (Step 4, on the [Upgrade the Cloudera Manager Server](#) on page 58 page.)

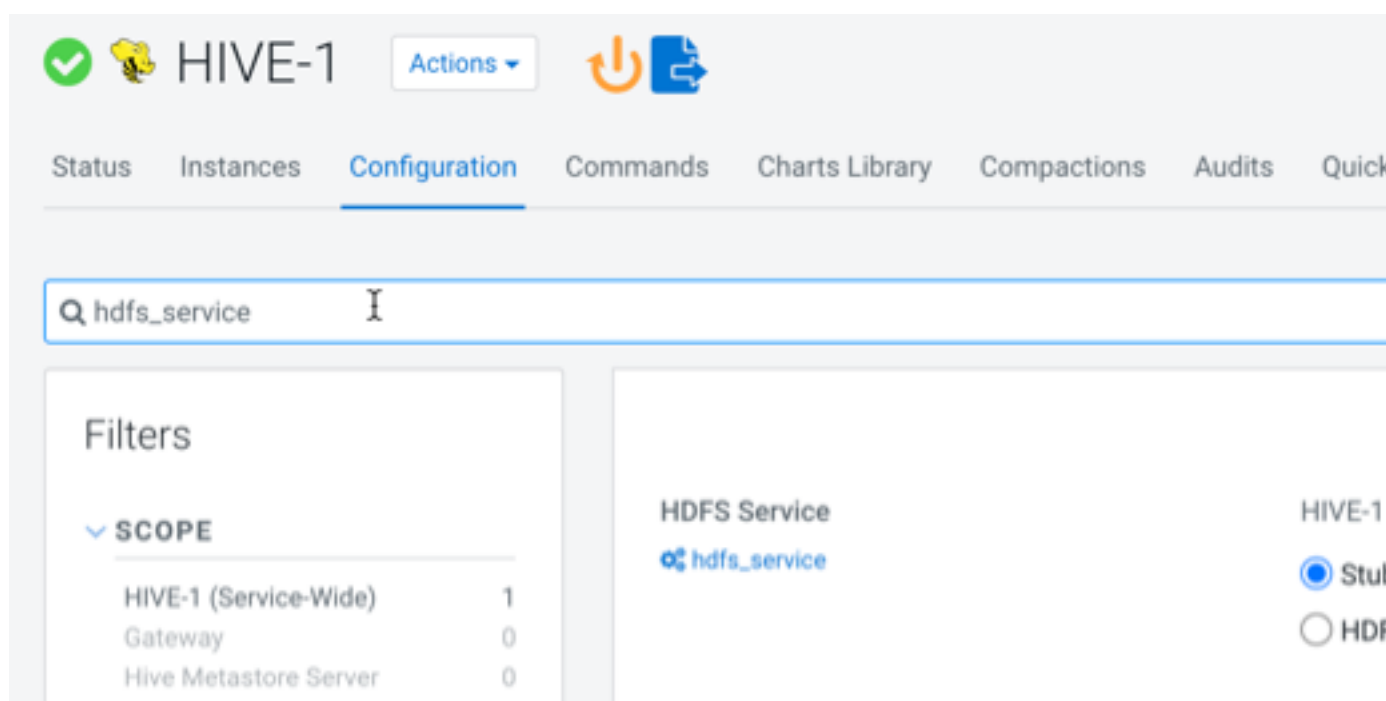
1. In the Cloudera Manager Admin Console, go to the status page the cluster.
2. Click the Actions menu and select Add Service.
3. Select the Stub DFS service.
4. Click Continue.

The Assign Roles page displays.

5. Accept the role assignments, or change them if necessary.
6. Click Continue.

Cloudera Manager adds the Stub DFS service.

- b. In the Cloudera Manager Admin Console, for each of these services, go to the Configuration tab and set the configuration property `hdfs_service` to Stub DFS. For example:



4. If your deployment uses LDAP, you may see that its health test has a Disabled status, you can configure an LDAP Bind Distinguished Name and password to enable the health test.

In the Cloudera Manager Admin Console, go to AdministrationSettingsExternal Authentication and set the following parameters:

- LDAP Bind Distinguished Name for Monitoring
- LDAP Bind Password for Monitoring

If these parameters are left blank, Cloudera Manager attempts to use the bind credentials specified for authentication.

5. If you have deployed Apache Atlas in the cluster, restart the Apache Atlas service.
6. If you have deployed Kafka in the cluster, perform a rolling restart the Kafka service.

7. If Cloudera Manager reports [stale configurations](#) after the upgrade, you might need to restart the cluster services and redeploy the client configurations. If any managed cluster includes the Hive and YARN components, this is required. If you will also be upgrading CDH, this step is not required.

Stale configurations can occur after a Cloudera Manager upgrade when a default configuration value has changed, which is often required to fix a serious problem. Configuration changes that result in Cloudera Manager reporting stale configurations are described the [release notes](#):

8. If you are using Streams Messaging Manager, you need to configure database related configuration properties.



Note: If you are also upgrading your distribution to Cloudera Runtime, you can choose to skip database configuration and complete it during the upgrade to Runtime.

- a. Select the Streams Messaging Manager service.
- b. Go to Configuration.
- c. Find and configure the following properties:
 - Streams Messaging Manager Database User Password
 - Streams Messaging Manager Database Type
 - Streams Messaging Manager Database Name
 - Streams Messaging Manager Database User
 - Streams Messaging Manager Database Host
 - Streams Messaging Manager Database Port

- d. Click Save Changes.


9. If you are using Schema Registry, you need to configure database related configuration properties.




Note: If you are also upgrading your distribution to Cloudera Runtime, you can choose to skip database configuration now and complete it during the upgrade to Runtime.

- a. Select the Schema Registry service.
- b. Go to Configuration.
- c. Find and configure the following properties:
 - Schema Registry Database User Password
 - Schema Registry Database Type
 - Schema Registry Database Name
 - Schema Registry Database User
 - Schema Registry Database Host
 - Schema Registry Database Port

- d. Click Save Changes.

10. On the HomeStatus tab, click  next to the cluster name, select Restart and confirm.


11. On the HomeStatus tab, click  next to the cluster name, select Deploy Client Configuration and confirm.


12. If you disabled any backup or snapshot jobs before the upgrade, now is a good time to re-enable them

13. If you deleted any Hive Replication schedules before the Cloudera Manager upgrade, re-create them now.

14. Check if you have set `allow_weak_crypto` to `true` in `/etc/krb5.conf` during JDK upgrade. If yes, proceed to the last step to complete the Cloudera Manager Upgrade, else proceed to the next step to set the kerberos encryption type.

15. If you have upgraded to Cloudera Manager 7.11.3, and are using JDK 11 or higher, and Cloudera Manager is managing the krb5.conf (when Cloudera Manager Administration Settings Manage krb5.conf through Cloudera Manager is enabled) then,
 - a. Stop the cluster.
 - b. Set the value of Cloudera Manager Administration Settings Kerberos Encryption Types to any 'AES'. Cloudera recommends setting it to 'aes256-cts'.
 - c. If MIT is in use, make sure the KDC version is greater than 1.18.2 to support any aes256 encryption type and:
 - Edit the kdc.conf and add aes-256 to supported_enctypes.
 - Restart KDC.
 - d. If, and only if, Active Directory (KDC) server is in use, then set Cloudera Manager Administration Settings Active Directory Set Encryption Types to 'True'.
 - e. Complete the Deploy Kerberos Client Configuration wizard OR call the Cloudera Manager API `deployClusterClientConfig`.
 - f. Regenerate the credentials. Go to Cloudera Manager Administration Security Kerberos Credentials tab, select all and click Regenerate Selected .

 **Note:** If the KDC is AD, then Regenerate Credentials can fail due to infrastructure issues. One common reason could be the replication delay between AD servers.

 **Note:** You can alternatively use the Cloudera Manager API to regenerate the credentials i.e. use `deleteCredentials` and then call `generateCredentials` API call. These API calls must be made once the cluster has been stopped.
 - g. Start the cluster.
16. Restart the Cloudera Manager server.
17. The Cloudera Manager upgrade is now complete. If Cloudera Manager is not working correctly, or the upgrade did not complete, see [Troubleshooting a Cloudera Manager Upgrade](#) on page 79.

Upgrading Cloudera Navigator Key Trustee Server 7.1.x

How to upgrade Cloudera Navigator Key Trustee Server 7.1.x.

About this task



Caution: Cloudera recommends that you do not create new encryption keys, encryption zones, or clients during an upgrade.



Note: Before upgrading Key Trustee Server, back up the Key Trustee Server. Refer [Backing up Key Trustee Server and clients](#) and [Restoring Navigator Key Trustee Server](#) for more information.

You must create an internal repository to install or upgrade the Cloudera Navigator data encryption components. For instructions on creating internal repositories, see the following topic:

- [Configuring a Local Parcel Repository](#)

From CDP Private Cloud Base 7.1.6, the KEYTRUSTEE_SERVER parcel is available in the same location in which the Cloudera Runtime parcel is placed. If you have configured the parcel repository for CDP Private Cloud Base upgrade, the KEYTRUSTEE_SERVER parcel is displayed automatically.

Upgrading Cloudera Navigator Key Trustee Server 7.1.x Using Cloudera Manager

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

Procedure

1. Add your internal parcel repository to Cloudera Manager following the instructions in [Configuring Cloudera Manager Server Parcel Settings](#).

- Download, distribute, and activate the latest Key Trustee Server parcel on the cluster containing the Key Trustee Server host, following the instructions in [Managing Parcels](#).



Important: Do not accept the prompt from Cloudera Manager to perform a rolling restart. Instead, restart the KTS services manually, beginning with the active database and then the active KTS, followed by the passive database and then the passive KTS.



Important: The KEYTRUSTEE parcel in Cloudera Manager is not the Key Trustee Server parcel; it is the Key Trustee KMS parcel. The parcel name for Key Trustee Server is KEYTRUSTEE_SERVER.

What to do next

If you have upgraded from a Key Trustee version lesser than 7.1.4 to a version greater than 7.1.4, then perform these steps on both the Active and Passive KTS nodes. This is to mitigate the SSL handshake issue caused due to absence of Subject Alternative name in KTS after upgrading to version > 7.1.4.

- Stop the KTS service from Cloudera Manager.
- Navigate to the location `/var/lib/keytrustee/.keytrustee/.ssl/`

```
cd /var/lib/keytrustee/.keytrustee/.ssl/
```

- Backup the cert files `ssl-cert-keytrustee-pk.pem` and `ssl-cert-keytrustee.pem`

```
mv ssl-cert-keytrustee-pk.pem ssl-cert-keytrustee-pk_backup.pem
mv ssl-cert-keytrustee.pem ssl-cert-keytrustee_backup.pem
```

- Re generate the cert file using the command:

```
ktadmin init
```

- Configure the keyhsm to trust the new cert file.

```
keyhsm trust /var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-keytrustee.pem
```

- For testing and validation execute the following command:

```
curl -vk https://$(hostname-f):11371/test_hsm
```



Note: The above two steps are only applicable if you are using Key HSM.

- Start the KTS service from Cloudera Manager.

Upgrading Cloudera Navigator Key HSM

Learn how to upgrade Navigator Key HSM.

Setting Up an Internal Repository

Although it is possible to upgrade Cloudera Navigator KeyHSM by using the KeyHSM RPM package directly, Cloudera recommends setting up a YUM package repository to perform the upgrade.

The steps given below assume that a repository containing the KeyHSM RPM package downloaded from the [paywall](https://wiki.centos.org/HowTos/CreateLocalRepos) has been created. For more information on creating such a repository, see <https://wiki.centos.org/HowTos/CreateLocalRepos>.

Upgrading Key HSM (Minor and Patch Version Upgrades)

If you are upgrading from Key HSM 1.x (shipped with CDH 5.x and earlier) to Key HSM 7.x, use the instructions in [Upgrading Key HSM \(Major Version Upgrades\)](#) on page 73; do not use the procedure documented in this section.



Important: If you have implemented Key Trustee Server high availability, upgrade Key HSM on each Key Trustee Server.

1. Install the KeyHSM Repository

Add the internal repository that you created.

2. Stop the Key HSM Service

Stop the Key HSM service before upgrading:

```
sudo service keyhsm shutdown
```

3. Upgrade Navigator Key HSM

Upgrade the Navigator Key HSM package using yum:

```
sudo yum update keytrustee-keyhsm
```

Cloudera Navigator Key HSM is installed to the /usr/share/keytrustee-server-keyhsm directory by default.

4. Start the Key HSM Service

Start the Key HSM service:

```
sudo service keyhsm start
```

Upgrading Key HSM (Major Version Upgrades)



Important: Only use this procedure if you are upgrading from Key HSM 1.x (shipped with CDH 5.x and earlier) to Key HSM 7.x. There is a unique configuration issue that impacts this upgrade scenario, and the steps here are different from those required for all minor Key HSM upgrades. This procedure is not applicable to minor version or patch release upgrades.

1. Install the KeyHSM Repository

Add the internal repository that you created.

2. Stop the Key HSM Service

Stop the Key HSM service before upgrading:

```
sudo service keyhsm shutdown
```

3. Upgrade Navigator Key HSM

Upgrade the Navigator Key HSM package using yum:

```
sudo yum update keytrustee-keyhsm
```

Cloudera Navigator Key HSM is installed to the /usr/share/keytrustee-server-keyhsm directory by default.

4. Rename Configuration Files that were created earlier

For Key HSM major version upgrades, previously-created configuration files do not authenticate with the HSM and Key Trustee Server, so you must recreate these files by re-executing the setup and trust commands. First, navigate to the Key HSM installation directory and rename the applications.properties, keystore, and truststore files:

```
cd /usr/share/keytrustee-server-keyhsm/  
mv application.properties application.properties.bak  
mv keystore keystore.bak  
mv truststore truststore.bak
```

5. Initialize Key HSM

Run the service keyhsm setup command in conjunction with the name of the target HSM distribution:

```
sudo service keyhsm setup [keysecure|thales|luna]
```

For more details, see [Initializing Navigator Key HSM](#).

6. Establish Trust Between Key HSM and the Key Trustee Server

The Key HSM service must explicitly trust the Key Trustee Server certificate (presented during TLS handshake). To establish this trust, run the following command:

```
sudo keyhsm trust /path/to/key_trustee_server/cert
```

For more details, see [Integrating Key HSM with Key Trustee Server](#).

7. Start the Key HSM Service

Start the Key HSM service:

```
sudo service keyhsm start
```

8. Establish Trust Between Key Trustee Server and Key HSM

Establish trust between the Key Trustee Server and the Key HSM by specifying the path to the private key and certificate:

```
sudo ktadmin keyhsm --server https://keyhsm01.example.com:9090 \
--client-certfile /etc/pki/cloudera/certs/mycert.crt \
--client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
```

For a password-protected Key Trustee Server private key, add the `--passphrase` argument to the command (enter the password when prompted):

```
sudo ktadmin keyhsm --passphrase \
--server https://keyhsm01.example.com:9090 \
--client-certfile /etc/pki/cloudera/certs/mycert.crt \
--client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
```

For additional details, see [Integrating Key HSM with Key Trustee Server](#).

9. Remove Configuration Files From Previous Installation

After completing the upgrade, remove the saved configuration files from the previous installation:

```
cd /usr/share/keytrustee-server-keyhsm/
rm application.properties.bak
rm keystore.bak
rm truststore.bak
```

Key Trustee Server SSL Certificate Regeneration

When Key HSM is upgraded to Cloudera version 7.1.4 or above, the SSL certificates of the Key Trustee Server (KTS) might need to be regenerated if the self-signed certificates that are created by the `ktadmin` command are being used.

Perform the following steps to regenerate the KTS SSL certificate:

1. Stop the KTS service from Cloudera Manager.
2. Navigate to the location `/var/lib/keytrustee/.keytrustee/.ssl/` to take a backup of the certificate files `ssl-cert-keytrustee-pk.pem` and `ssl-cert-keytrustee.pem`:

```
cd /var/lib/keytrustee/.keytrustee/.ssl/
```

3. Backup the certificate files:

```
mv ssl-cert-keytrustee-pk.pem ssl-cert-keytrustee-pk_backup.pem
mv ssl-cert-keytrustee.pem ssl-cert-keytrustee_backup.pem
```

4. Regenerate the certificate file:

```
ktadmin init
```

5. Configure the Key HSM to trust the new certificate file:

```
keyhsm trust /var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-keytrustee.pem
```

6. Restart the Key HSM service.**7. Start the KTS service from Cloudera Manager.****8. Run the following command to test and validate certificate regeneration:**

```
curl -vk https://$(hostname -f):11371/test_hsm
```

Upgrading Cloudera Navigator Encrypt

Setting Up an Internal Repository

You must create an internal repository to install or upgrade the Cloudera Navigator data encryption components. For instructions on creating internal repositories, see the following topics:

- [Configuring a Local Parcel Repository](#)
- [Configuring a Local Package Repository](#)

Upgrading Navigator Encrypt (RHEL-Compatible)

Before you begin the upgrade process, refer to Product Compatibility Matrix for Cloudera Navigator Encryption and ensure that you have the minimum requisite operating system version(s) installed.

1. Install the Cloudera Repository

Add the internal repository you created. See [Configuring Hosts to Use the Internal Repository](#) for more information.

Import the GPG key by running the following command:

```
sudo rpm --import http://repo.example.com/path/to/gpg_gazzang.asc
```

2. Stop Navigator Encrypt

Stop the Navigator Encrypt service:

```
sudo service navencrypt-mount stop
```

For RHEL 7, use systemctl instead:

```
sudo systemctl stop navencrypt-mount
```

3. Upgrade Navigator Encrypt

Upgrade the Navigator Encrypt client using yum:

```
sudo yum update navencrypt
```

4. Start Navigator Encrypt

Start the Navigator Encrypt service:

```
sudo service navencrypt-mount start
```

For RHEL 7, use systemctl instead:

```
sudo systemctl start navencrypt-mount
```

5. If you are using an RSA master key file, change the master key to use OAEP padding:

```
# navencrypt key --change --rsa-oaep
...
>> Choose NEW MASTER key type:
  1) Passphrase (single)
  2) Passphrase (dual)
  3) RSA private key
Select: 3
Type MASTER RSA key file:
Type MASTER RSA key passphrase:
```

To check the type of padding currently in use:

```
# navencrypt key --get-rsa-padding
Type your Master key
Type MASTER RSA key file:
Type MASTER RSA key passphrase:

Verifying Master Key against keytrustee (wait a moment)...
RSA_PKCS1_OAEP_PADDING
```

Upgrading Navigator Encrypt (SLES)

1. Install the Cloudera Repository

Add the internal repository that you created. See [Configuring Hosts to Use the Internal Repository](#) for more information.

Import the GPG key by running the following command:

```
sudo rpm --import http://repo.example.com/path/to/gpg_gazzang.asc
```

2. Stop Navigator Encrypt

Stop the Navigator Encrypt service:

```
sudo service navencrypt-mount stop
```

3. Upgrade the Kernel Module Package (KMP)

```
sudo zypper update cloudera-navencryptfs-kmp-<kernel_flavor>
```

Replace `kernel_flavor` with the kernel flavor for your system. Navigator Encrypt supports the default, xen, and ec2 kernel flavors.

4. Upgrade the Navigator Encrypt Client

Upgrade Navigator Encrypt:

```
sudo zypper update navencrypt
```

5. Enable Unsupported Modules

Edit `/etc/modprobe.d/unsupported-modules` and set `allow_unsupported_modules` to 1. For example:

```
#
# Every kernel module has a flag 'supported'. If this flag is not set loading
# this module will taint your kernel. You will not get much help with a
# kernel
# problem if your kernel is marked as tainted. In this case you firstly have
# to avoid loading of unsupported modules.
#
# Setting allow_unsupported_modules 1 enables loading of unsupported modules
# by modprobe, setting allow_unsupported_modules 0 disables it. This can
# be overridden using the --allow-unsupported-modules command line switch.
allow_unsupported_modules 1
```

6. Start Navigator Encrypt

Start the Navigator Encrypt service:

```
sudo service navencrypt-mount start
```

7. If using an RSA master key file, then you should change the master key to use OAEP padding:

```
# navencrypt key --change --rsa-oaep
...
>> Choose NEW MASTER key type:
  1) Passphrase (single)
  2) Passphrase (dual)
  3) RSA private key
Select: 3
Type MASTER RSA key file:
Type MASTER RSA key passphrase:
```

To check the type of padding currently in use:

```
# navencrypt key --get-rsa-padding
Type your Master key
Type MASTER RSA key file:
Type MASTER RSA key passphrase:

Verifying Master Key against keytrustee (wait a moment)...
RSA_PKCS1_OAEP_PADDING
```

Upgrading Navigator Encrypt (Debian or Ubuntu)

1. Install the Cloudera Repository

Add the internal repository that you created. See [Configuring Hosts to Use the Internal Repository](#) for more information.

- Ubuntu

```
echo "deb http://repo.example.com/path/to/ubuntu/stable $DISTRIB_CODENAME main" | sudo tee -a /etc/apt/sources.list
```

- Debian

```
echo "deb http://repo.example.com/path/to/debian/stable $DISTRIB_CODENAME main" | sudo tee -a /etc/apt/sources.list
```

Import the GPG key by running the following command:

```
wget -O - http://repo.example.com/path/to/gpg_gazzang.asc | apt-key add -
```

Update the repository index with apt-get update.

2. Stop Navigator Encrypt

Stop the Navigator Encrypt service:

```
sudo service navencrypt-mount stop
```

3. Upgrade the Navigator Encrypt Client

Upgrade Navigator Encrypt:

```
sudo apt-get install navencrypt
```

4. Start Navigator Encrypt

Start the Navigator Encrypt service:

```
sudo service navencrypt-mount start
```

5. If using an RSA master key file, then you should change the master key to use OAEP padding:

```
# navencrypt key --change --rsa-oaep
...
>> Choose NEW MASTER key type:
1) Passphrase (single)
2) Passphrase (dual)
3) RSA private key
Select: 3
Type MASTER RSA key file:
Type MASTER RSA key passphrase:
```

To check the type of padding currently in use:

```
# navencrypt key --get-rsa-padding
Type your Master key
Type MASTER RSA key file:
Type MASTER RSA key passphrase:

Verifying Master Key against keytrustee (wait a moment)...
RSA_PKCS1_OAEP_PADDING
```

Best Practices for Upgrading Navigator Encrypt Hosts

Following are some best practices for upgrading operating systems (OS) and kernels on hosts that have Navigator Encrypt installed:

- Make sure that the version you are upgrading to is supported by Navigator Encrypt. See the product compatibility matrix for [Product Compatibility Matrix for Cloudera Navigator Encryption](#) for more information.
- Always test upgrades in a development or testing environment before upgrading production hosts.
- If possible, upgrade the entire operating system instead of only upgrading the kernel.
- If you need to upgrade the kernel only, make sure that your OS version supports the kernel version to which you are upgrading.
- Always back up the `/etc/navencrypt` directory before upgrading. If you have problems accessing encrypted data after upgrading the OS or kernel, restore `/etc/navencrypt` from your backup and try again.

Troubleshooting a Cloudera Manager Upgrade

The Cloudera Manager Server fails to start after upgrade.

The Cloudera Manager Server fails to start after upgrade.

Possible Reasons

There were active commands running before upgrade. This includes commands a user might have run and also commands Cloudera Manager automatically triggers, either in response to a state change, or something configured to run on a schedule, such as Backup and Disaster Recovery replication or snapshot jobs.

Possible Solutions

- Stop any running commands from the Cloudera Manager Admin Console or wait for them to complete. See [Aborting a Pending Command](#).
- Ensure that you have disabled any scheduled replication or snapshot jobs from the Cloudera Manager Admin Console to complete before proceeding with the upgrade. See [HDFS Replication](#).

Re-Running the Cloudera Manager Upgrade Wizard

Minimum Required Role: [Full Administrator](#). This feature is not available when using Cloudera Manager to manage Data Hub clusters.

The first time you log in to the Cloudera Manager server after upgrading your Cloudera Manager software, the upgrade wizard runs. If you did not complete the wizard at that time, or if you had hosts that were unavailable at that time and still need to be upgraded, you can re-run the upgrade wizard:

1. Click the Hosts tab.
2. Click Re-run Upgrade Wizard or Review Upgrade Status. This takes you back through the installation wizard to upgrade Cloudera Manager Agents on your hosts as necessary.
3. Select the release of the Cloudera Manager Agent to install. Normally, this is the Matched Release for this Cloudera Manager Server. However, if you used a custom repository (instead of `archive.cloudera.com`) for the Cloudera Manager server, select Custom Repository and provide the required information. The custom repository allows you to use an alternative location, but that location must contain the matched Agent version.

4. Specify credentials and initiate Agent installation:

- a. Select root for the root account, or select Another user and enter the username for an account that has password-less sudo privileges.
- b. Select an authentication method:
 - If you choose password authentication, enter and confirm the password.
 - If you choose public-key authentication, provide a passphrase and path to the required key files.

You can modify the default SSH port if necessary.

- c. 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.
- d. Click Continue.

When you click Continue, the Cloudera Manager Agent is upgraded on all the currently managed hosts. You cannot search for new hosts through this process. To add hosts to your cluster, click the Add New Hosts to Cluster button.

Reverting a Failed Cloudera Manager Upgrade

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)

This topic describes how to reinstall the same version of Cloudera Manager you were using previously, so that the version of your Cloudera Manager Agents match the server. The steps below assume that the Cloudera Manager Server is already stopped (because it failed to start after the attempted upgrade).



Important:

The following instructions assume that a Cloudera Manager upgrade failed, and that the upgraded server never started, so that the remaining steps of the upgrade process were not performed. The steps below are not sufficient to revert from a running Cloudera Manager deployment.

To revert a running Cloudera Manager Server and Cloudera Manager Agents after a successful upgrade, see [\(Optional\) Cloudera Manager Rollback Steps](#), which have instructions for reverting the Cloudera Manager packages on all hosts.

Ensure Cloudera Manager Server and Agent are stopped.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Stop the Cloudera Manager Server.

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

3. Stop the Cloudera Manager Agent.

```
sudo systemctl stop cloudera-scm-agent
```

```
sudo systemctl stop cloudera-scm-supervisord
```

Restore the Cloudera Manager Database (if necessary)

If your Cloudera Manager upgrade fails, you need to determine whether the upgrade process has successfully completed updating the schema of the Cloudera Manager database. If the schema update has begun, you must restore the Cloudera Manager database using a [backup](#) taken before you began the upgrade.

1. To determine whether the schema has been updated, examine the Cloudera Manager server logs and look for a message similar to the following: Updated Schema Version to 60000. (The version number may be different for your environment.)

Run the following command to find the log entry:

```
grep 'Updated Schema Version to ' /var/log/cloudera-scm-server/cloudera-scm-server.log
```

2. If required, restore the database.

The procedure for restoring the database depends on the type of database used by Cloudera Manager.

3. If you are using the embedded PostgreSQL database, stop the Cloudera Manager Embedded PostgreSQL database:

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

If you are not using the embedded PostgreSQL database and you attempt to stop it, you might see a message indicating that the service cannot be found.

In case of using the embedded PostgreSQL, if you see a message that the shutdown failed, then the embedded database is still running, probably because services are connected to the Hive metastore. If the database shutdown fails due to connected services, issue the following command:

```
sudo service cloudera-scm-server-db next_stop_fast
sudo service cloudera-scm-server-db stop
```

Establish Access to the Software

Cloudera Manager needs access to a package repository that contains the updated software packages. You can choose to access the Cloudera public repositories directly, or you can [download those repositories and set up a local repository](#) to access them from within your network. If your cluster hosts do not have connectivity to the Internet, you must set up a local repository.

If you have enabled high availability for Cloudera Manager, perform the following steps on the hosts for both the active and passive instances of the Cloudera Manager server.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Remove any older files in the existing repository directory:

RHEL / CentOS

```
sudo rm /etc/yum.repos.d/cloudera*manager.repo*
```

SLES

```
sudo rm /etc/zypp/repos.d/cloudera*manager.repo*
```

Ubuntu

```
sudo rm /etc/apt/sources.list.d/cloudera*.list*
```

3. Create a repository file so that the package manager can locate and download the binaries.



Note: If you are upgrading to a Cloudera Manager patch, substitute the download URL for the patch when creating the repository file (cloudera-manager.repo).

Do one of the following, depending on whether or not you are using a local package repository:

- Use a local package repository (Required when cluster hosts do not have access to the internet.) See [Configuring a Local Package repository](#).
- Use the Cloudera public repository

RHEL / CentOS

- Create a file named /etc/yum.repos.d/cloudera-manager.repo with the following content:

```
[cloudera-manager]
name=Cloudera Manager
baseurl=https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
VERSION>/redhat<OS MAJOR VERSION>/yum/
gpgkey =https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
VERSION>/redhat<OS MAJOR VERSION>/yum/RPM-GPG-KEY-cloudera
username=CHANGEME
password=CHANGEME
gpgcheck=1
enabled=1
autorefresh=0
type=rpm-md
```

Replace *CHANGEME* with your *USERNAME* and *PASSWORD* in the /etc/yum.repos.d/cloudera-manager.repo file.

SLES

- Create a file named /etc/zypp/repos.d/cloudera-manager.repo with the following content:

```
[cloudera-manager]
name=Cloudera Manager
baseurl=https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
VERSION>/sles<OS MAJOR VERSION>/yum/
gpgkey =https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
VERSION>/sles<OS MAJOR VERSION>/yum/RPM-GPG-KEY-cloudera
username=CHANGEME
password=CHANGEME
gpgcheck=1
enabled=1
autorefresh=0
type=rpm-md
```

- Replace *CHANGEME* with your *USERNAME* and *PASSWORD* in the /etc/zypp/repos.d/cloudera-manager.repo file.

Ubuntu

Debian is not a supported operating system for Cloudera Manager 6.x.

- Create a file named /etc/apt/sources.list.d/cloudera_manager.list with the following content:

```
# Cloudera Manager <CLOUDERA MANAGER VERSION>
deb [arch=amd64]
http://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/<CLOUDERA
MANAGER VERSION>/ubuntu1804/apt -cm<CLOUDERA MANAGER VERSION>
contrib
```

- b. Run the following command:

```
sudo apt-get update
```

- c. Replace *CHANGE_ME* with your *USERNAME* and *PASSWORD* in the `/etc/apt/sources.list.d/cloudera_manager.list` file.



Tip: If you have a mixed operating system environment, adjust the Operating System filter at the top of the page for each operating system. The guide will generate the repo file for you automatically here.

4. A Cloudera Manager upgrade can introduce new package dependencies. Your organization may have restrictions or require prior approval for the installation of new packages. You can determine which packages may be installed or upgraded:

RHEL / CentOS

```
yum deplist cloudera-manager-agent
```

SLES

```
zypper info --requires cloudera-manager-agent
```

Ubuntu

```
apt-cache depends cloudera-manager-agent
```

Downgrade the Cloudera Manager Packages



Note: Make sure the repository file above matches the specific maintenance version before the upgrade.

1. Downgrade the packages. Note: Only add `cloudera-manager-server-db-2` if you are using the embedded PostgreSQL database.

RHEL / CentOS

```
sudo yum clean all
sudo yum repolist
```

```
sudo yum downgrade "cloudera-manager-*
```

SLES

```
sudo zypper clean --all
```

```
sudo zypper dup -r BASEURL
```

Ubuntu

There is no action that downgrades Cloudera Manager to the version currently in the repository.

2. Verify that you have the correct packages installed.

Ubuntu

```
dpkg-query -l 'cloudera-manager-*
```

```
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aW
ait/Trig-pend
```

```
| / Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
| / Name                               Version                               Description
+++-----
=====
ii cloudera-manager-agent 5.15.0-0.cm...~sq The Cloudera Manager
   Agent
ii cloudera-manager-daemo 5.15.0-0.cm...~sq Provides daemons for
   monitoring Hadoop and related tools.
ii cloudera-manager-serve 5.15.0-0.cm...~sq The Cloudera Manager
   Server
```

RHEL / CentOS / SLES

```
rpm -qa 'cloudera-manager-*
```

```
cloudera-manager-server-5.15.0-...
cloudera-manager-agent-5.15.0-...
cloudera-manager-daemons-5.15.0-...
cloudera-manager-server-db-2-5.15.0-...
```

Validate the configuration of the Cloudera Manager Server or Agent

The configuration files for the Cloudera Server are under `/etc/cloudera-scm-server`, while those of the Agent are under `/etc/cloudera-scm-agent`.

Since you have downgraded the Cloudera packages, the downgrade process will not overwrite the configuration files, and the values you set before will remain.

Start Cloudera Manager Again

1. If you are using the embedded PostgreSQL database, start the database:

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

2. Start the Cloudera Manager Server by running the following command:

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

3. Once the server is up (and you can access the UI), start the Cloudera Manager Agents on all the clusters by running the following command:

```
sudo systemctl start cloudera-scm-agent
```

If the agent starts without errors, no response displays.



Important: A restart of Cloudera Manager Agents is required on all hosts in the cluster.

4. Start the Cloudera Management Service.
 - a. Log in to the Cloudera Manager Admin Console.
 - b. Select Clusters Cloudera Management Service .
 - c. Select Actions Start .



Note: Troubleshooting: If you have problems starting the server, such as database permissions problems, you can use the server's log to troubleshoot the problem.

```
vim /var/log/cloudera-scm-server/cloudera-scm-server.log
```

Installing dependencies for Hue

You must install the psycopg2 Python package for PostgreSQL-backed Hue and MySQL clients for MariaDB and MySQL databases depending on your operating systems.



Note: This task is applicable only if you are upgrading to Cloudera Runtime 7.1.8 and higher. If you are upgrading to Cloudera Runtime 7.1.7 or lower, then you can skip this task.

If you are using Oracle as a backend database for Hue, then review [Using Oracle database with Hue](#) to ensure that Hue connects to your database.

Installing the psycopg2 Python package for PostgreSQL database

If you are using PostgreSQL as a backend database for Hue on Cloudera Base on premises 7.x.x, then you must install a version of the psycopg2 package to be at least 2.9.5 on all Hue hosts. The psycopg2 package is automatically installed as a dependency of Cloudera Manager Agent, but the version installed is often lower than 2.9.3.

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. Change to the directory where you had downloaded the cloudera-manager.repo file. On RHEL, the file is present under the /etc/yum.repos.d directory.
3. Open the file for editing and update the value of the enabled property to 0 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 7, RHEL 7, OEL 7, CentOS 8, RHEL 8, and OEL 8:

1. SSH into the Hue server host as a root user.
2. Install the psycopg2-binary package as follows:

```
pip3.8 install psycopg2-binary
```

3. Repeat these steps on all the Hue 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 3.9:

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

```
yum install python3-pip -y
```

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:

```
pip3 install psycopg2-binary
```

5. Repeat these steps on all the Hue 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

The following steps apply to SLES 12 (upgrades to Cloudera Runtime 7.1.8) and SLES 15 SP4 (upgrades to Cloudera Runtime 7.1.9 or higher):

1. SSH into the Hue host as a root user.
2. Install the psycopg2 package dependencies 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:

```
pip3.8 install psycopg2==2.9.3 --ignore-installed
```

For Ubuntu

The following steps apply to Ubuntu 18 (upgrades to Cloudera Runtime 7.1.8) and Ubuntu 20 (upgrades to Cloudera Runtime 7.1.9 or higher):

1. SSH into the Hue host as a root user.
2. Install the psycopg2 package dependencies 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:

```
pip3.8 install psycopg2==2.9.3 --ignore-installed
```

Installing MySQL client for MySQL databases

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.



Attention: The version 2.2.0 of the MySQL client requires that you set the values of the MYSQLCLIENT_CFLAGS and MYSQLCLIENT_LDFLAGS environment variables, as follows:

```
$ export MYSQLCLIENT_CFLAGS=`pkg-config mysqlclient --cflags`
$ export MYSQLCLIENT_LDFLAGS=`pkg-config mysqlclient --libs`
```

The version 2.2.0 of the MySQL client also requires you to install the Python 3 and MySQL development headers and libraries, as follows on Debian or Ubuntu operating systems:

```
sudo apt-get install python3-dev default-libmysqlclient-dev build-essential
```

or as follows on CentOS and RHEL operating systems:

```
sudo yum install python3-devel mysql-devel
```

Alternatively, you can install and use the version 2.1.1 of the MySQL client, as follows, which does not have this requirement:

```
pip3 install mysqlclient=2.1.1
```

For Cent OS

1. SSH into the Hue host as a root user.
2. Download the MySQL yum repository as follows:

```
curl -sSO 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 -sLO https://dev.mysql.com/get/mysql80-community-release-el7-5.noarch.rpm
```

(RHEL 8)

```
curl -sLO https://dev.mysql.com/get/mysql80-community-release-el8-8.noarch.rpm
```

(RHEL 9)

```
curl -sLO https://dev.mysql.com/get/mysql80-community-release-el9-4.noarch.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
```

Installing MySQL client for MariaDB databases

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 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)

```
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
```

Getting started with Zero Downtime Upgrade (ZDU)

Understand what ZDU process is and when you can use this process to upgrade your existing Cloudera Runtime cluster to the latest version. Also, review the high-level steps involved in completing a Zero Downtime upgrade.

Before you upgrade your Cloudera Runtime cluster using the Zero Downtime Upgrade (ZDU) process, it is a best practice to review the high-level ZDU steps, the software requirements, and the limitations and considerations related to service components.

Before you plan the upgrade, you must be aware of the tasks you must complete to ensure the cluster readiness.

What Zero Downtime Upgrade (ZDU) is

Zero Downtime Upgrades (ZDUs) are an evolution to the legacy method of performing a rolling upgrade on components in a cluster. Zero Downtime Upgrades aim to optimize the upgrade process to allow end-users to continue using major services while upgrading Cloudera Runtime cluster. In this initial release, some services will perform a quick restart, greatly reducing the amount of downtime end-users have historically experienced during a cluster upgrade.

After completing the preparatory steps, you use the Cloudera Manager upgrade wizard to complete the upgrade. Cloudera Manager restarts all services after the upgrade. Services that support rolling upgrade are restarted in a rolling fashion, and do not require you to take the services offline during the upgrade. All other services are restarted normally.



Note:

- Cloudera recommends all upgrades to happen in a maintenance window by throttling and scaling down workloads during that time as a best practice.



Caution: DDLs must not be executed during the maintenance window and the upgrade will error out if this is done.

When to use ZDU process

You can use the Zero Downtime Upgrade (ZDU) process to upgrade your existing cluster running Cloudera Runtime version 7.1.7 SP2 or 7.1.8 with the latest CHF or Service Pack, to the 7.1.9 or 7.1.9 SP1 version and Cloudera Runtime version 7.1.7 SP3 or [CDP 7.1.8 cumulative hotfix 17](#) or 7.1.9 SP1 or 7.1.9 to the 7.3.1 version, using Cloudera Manager, with less or no service downtime, process interruption, or loss of data. For more information on the supported upgrade paths, see the [Supported in-place upgrade paths](#) documentation. You should use this guide if you want to upgrade your Cloudera Runtime cluster to any one of the following using the ZDU process:

- A higher version of Cloudera Runtime
- A maintenance release of Cloudera Runtime



Warning: Avoid performing a Zero Downtime Upgrade (ZDU) for Cloudera deployments running Impala when upgrading to Cloudera Runtime 7.3.1.500. This is due to Thrift protocol incompatibilities that can cause queries to fail during upgrade.

Activities to complete before starting ZDU

1. Understand and review the following:
 - [Software requirements and component preparation tasks](#)
 - [Limitations and consideration related to components](#)
 - [ZDU terminologies](#)
2. Perform the following prerequisite tasks:
 - a. [Back up the existing Cloudera Manager version 7.x.x.](#)
 - b. [Upgrade the Cloudera Manager version to 7.11.3.](#)
 - c. [Back up the cluster.](#)

Zero Downtime Upgrade (ZDU) process

1. [Review the software requirements before performing the upgrade.](#)
2. [Review notes and warnings to be aware of the pre upgrade tasks required from your end before performing the upgrade.](#)
3. [Back up the cluster managed by Cloudera Manager and back up all your components before performing the Zero Downtime Upgrade.](#)
4. [Back up Cloudera Manager again.](#)
5. [Access the Parcels required to install Cloudera Runtime.](#)
6. [Upgrade the parcels.](#)

Software Requirements

Review the software requirements to understand the minimum supported versions for upgrading Cloudera Manager and Cloudera Runtime cluster.

- Source cluster must be running Cloudera Runtime 7.1.7 or Cloudera Runtime 7.1.8 with the latest CHF or Service Pack. However, Cloudera recommends your cluster must be on a version higher than Cloudera Runtime 7.1.7 SP2 or Cloudera Runtime 7.1.8 with the latest CHF.
- Source cluster must be running Cloudera Runtime 7.1.7 SP3 or [CDP 7.1.8 cumulative hotfix 17](#) or 7.1.9 SP1 or 7.1.9 to upgrade to Cloudera Base on premises 7.3.1.
- Source cluster must be running Cloudera Manager version 7.6.1 and higher.
- All active services must have the required prerequisites (for example, high-availability) setup before starting the process.
- For more information on the supported upgrade paths, see the [Supported in-place upgrade paths](#) documentation.

Configuring components before starting ZDU

You must review and perform the prerequisite tasks related to service components before performing the ZDU upgrade. These tasks help you to prepare your components for the ZDU upgrade process.

Atlas - update Solr replication factor

In Solr, before upgrading, when you set the replication factor to 2, it implies 1 leader replica and 1 follower replica are available in each of the nodes. Later, when performing rolling upgrade/restart, you still have a follower replica in a node which can be used to serve Atlas without any downtime from the client (Atlas). Restart Atlas for the changes to take effect.

If you already have:

- Cloudera Runtime 7.1.9 installation and plan to upgrade with Zero Downtime Upgrade process and with replication factor set to 1, perform the following steps; however, performing these steps can lead to data loss:
 - Stop the Atlas service.
 - Delete the Solr collections manually. For more information, see [Managing collections in Search](#).
 - Set the replication factor to the number of available Solr brokers / instances.
 - Restart Atlas.
 - Check that the Solr collections are created with replication factor as set in Atlas.
- If you are performing a fresh Cloudera Runtime 7.1.9 installation, then set the replication factor to number of available Solr broker / instance and continue with Atlas installation.

Atlas - update Kafka replication factor to the number of available Kafka brokers

You must perform certain prerequisite tasks based on whether Kafka topics are available and Atlas is installed on your cluster.

When Kafka topics are available, use the following procedure.

1. Stop the Atlas service.
2. Delete the topics (ATLAS_HOOK, ATLAS_ENTITIES, and ATLAS_SPARK_HOOK) manually. For more information, see [Monitoring Kafka topics](#) and [Deleting a Kafka topic](#).
3. Set the replication factor to the number of available Kafka brokers in Kafka.
4. Restart Kafka.
5. Restart Atlas.
6. Check that the topics are created with a replication factor as set in Kafka.

When Kafka topics are not available, use the following procedure:

1. Set the replication factor to the number of available Kafka brokers in Kafka.
2. Restart Kafka.
3. Click Action Create Kafka topics for Atlas in Atlas service.
4. Check that the topics are created with the replication factor as set in Kafka.

When Atlas is not installed on your cluster, use the following procedure:

1. Set replication factor to the number of available Kafka brokers in Kafka, Restart Kafka.
2. Install the Atlas service.
3. Check that the topics are created with the replication factor as set in Kafka.
4. In your Kafka environment, go to Actions Create Kafka topics for Atlas .

Data Analytics Studio (Deprecated)

Data Analytics Studio (DAS) has been deprecated in 7.1.9 and is no longer available with Cloudera. Hue now replaces DAS. Before you upgrade, you must remove the DAS service from your cluster.

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters DAS service and click Actions Stop .
3. Click Actions Delete .

Cloudera recommends that you delete the DAS database to free up resources.

HDFS

You must ensure that the High Availability mode is enabled and more than one HDFS NameNode is running.

HMS

You must enable High Availability for Hive Metastore (HMS) and ensure that more than one instance of HMS is running. For details, see [Configuring HMS for high availability](#).

Hive on Tez

You must enable High Availability for HiveServer (HS2) and ensure that more than one instance of HS2 is running. For details, see [Configuring HiveServer high availability using ZooKeeper](#).



Important: During a HS2 restart, any running queries are aborted and will have to be manually resubmitted. Therefore, it is recommended that you do not submit new queries until all the HS2 instances are restarted.

You can use the graceful shutdown feature of HS2, which ensures that HS2 waits for a specified time period before shutting down, thereby allowing queries that are already running to complete before HS2 stops. However, this feature is available from Cloudera Runtime 7.1.9 onwards and can only be used for Cloudera Runtime upgrades from 7.1.9 to higher versions.

Hue

- You must enable High Availability for Hue to leverage cluster-level rolling restart. See [Configuring high availability for Hue](#).
- You must run `yum install gcc openssl-devel bzip2-devel libffi-devel zlib-devel` along with `python 3.8` command before starting the upgrade.
- You must install dependencies by running the following command along with `Python 3.8` command before starting the upgrade:

```
yum install gcc openssl-devel bzip2-devel libffi-devel zlib-devel
```

- You must install `psycopg2-binary` before starting the upgrade if you are using Postgres. See [Installing the psycopg2 Python package for PostgreSQL database](#).

Impala

You must enable High Availability (HA) for Impala Catalog server and StateStore server by configuring more than one Impala Catalog server and one StateStore server instance on different nodes. For details, see [Configuring Impala for High Availability](#)

Consideration for ZDU implementation:

- Ensure that High Availability is enabled for both the Catalog and StateStore. There might be a brief window of downtime during Catalog or StateStore failover. For details see: [Failure detection for Catalog and StateStore](#)
- Ensure at least two coordinators behind a load balancer. The batch rolling upgrade count for workers should be less than the number of coordinators.
- ZDU works effectively when the Catalog or StateStore protocol versions remain unchanged.

Kafka

The Kafka service in Cloudera consists of multiple service role types. Each role type represents core features in Kafka. The following collects the ZDU prerequisites for each role type. You must ensure that these prerequisites are met to ensure that the service can be upgraded with zero downtime.

Although the Kafka service supports ZDU if the following prerequisites are met, Kafka service monitoring provided by SMM is briefly interrupted during the upgrade because SMM does not support ZDU.

Kafka Broker

- Ensure that your Kafka Broker roles are running in HA mode. That is, you have more than a single Kafka Broker role deployed in the cluster. Cloudera recommends a minimum of 3.
- Ensure that `replication.factor` is set to at least 3 and `min.insync.replicas` is set to at least 2 for all mission critical topics. Additionally, note the following about these configuration values:
 - The value set for `min.insync.replicas` must always be at least 1 less than `replication.factor`.
 - Topics with `replication.factor=2` or `min.insync.replicas=1` are not considered highly available.
 - Topics with `replication.factor=1` become unavailable when the broker hosting the topic goes temporarily offline during the upgrade.

- Ensure that the upgrade Batch Size is set to 1.

Batch Size is the number of roles that are restarted simultaneously during the upgrade. Batch Size is configured during the Choose Upgrade Procedure step of the **Upgrade Wizard**. If Batch Size is set to a value greater than 1, Kafka Broker roles hosting the replicas of the same topic might go down simultaneously. This results in the topic becoming temporarily unavailable.

- Ensure that the Cluster Health Guarantee During Rolling Restart Kafka property is set to a value that fits your requirements. For more information, see [Rolling restart checks](#).
- If your cluster uses Cruise Control and self-healing is enabled, ensure that the time set in broker.failure.self.healing.threshold.ms is higher than the expected downtime of the brokers. This property controls the time a broker can be offline before Cruise Control starts the self-healing process. If the property is set to a low value, you risk triggering self-healing during an upgrade. The property is set to 1,800,000 ms (30 minutes) by default, which should be viable for most deployments. If you configured this property and set a custom threshold value, ensure that the value you configured is sufficiently high. If you are unsure, you can either revert to the default value, or temporarily disable Cruise Control for the duration of the upgrade. You can set the property in Cloudera Manager with Cruise Control Server Advanced Configuration Snippet (Safety Valve) for cruisecontrol.properties. Restart the Cruise Control service if you make changes to the property.

Kafka Connect

- Ensure that your Kafka Connect roles are running in HA mode. That is, you have more than a single Kafka Connect role deployed in the cluster. Cloudera recommends a minimum of 3.
- If you have external clients that connect to the Kafka Connect REST API, ensure that you are using a load balancer to redirect REST API calls. If you do not have a load balancer, expect service downtime as Kafka Connect provides no automatic failover between its REST endpoints.

MirrorMaker

Ensure that your MirrorMaker roles are running in HA mode. That is, you have more than a single MirrorMaker role deployed in the cluster. Cloudera recommends a minimum of 3.

Key Trustee Server

1. Back up Key Trustee Server manually on both Active and Passive hosts
2. Back up the Key Trustee Server configuration directory on both Active and Passive hosts
3. Move the .backup files (keytrustee-db.zip and keytrustee-conf.zip) to a secure location on both Active and Passive hosts.

Kudu

Ensure that the High Availability mode is enabled and at least three Kudu master servers and three tablet servers are running and all the tables have a replication factor of three or more.

Livy

- Ensure that the HA mode is enabled and more than one Livy Server is running.
- Ensure that the Livy service supports the rolling restart functionality in Cloudera Manager.

Omid

- Ensure that the HA mode is enabled and more than one OMID TSO server is running.
- Ensure that the OMID service supports the rolling restart functionality.
- Ensure that the Enable HA for Omid TSO server option is selected under **Configurations** in Cloudera Manager.

Phoenix

- Ensure that the HA mode is enabled and more than one PQS server is running.
- Ensure that the Phoenix service supports the rolling restart functionality.

Ranger KMS

- Before Ranger KMS KTS upgrade, perform the Key Trustee Server prerequisite tasks, and then back up Ranger KMS KTS by going to Cloudera Manager Ranger KMS KTS Actions Create Backup .
- Before upgrading Ranger KMS DB, back up Ranger and Ranger KMS DB.

Schema Registry

When upgrading from Cloudera Runtime 7.1.8 to 7.1.9 or higher, if you only have external clients connecting to the server, you can ensure zero downtime if you upgrade your external clients before upgrading the cluster. The actions you must take differ depending on the type client you are using:

- If you are using a generic HTTP client, ensure that retry and failover is implemented in the client code.
- If you are using the Cloudera distributed Schema Registry Java client, ensure that you upgrade your applications that use the client before you upgrade the cluster. You can access the latest version of the client in the Cloudera Manager Maven repository at <https://repository.cloudera.com/artifactory/cloudera-repos/>. Alternatively, you can choose to implement retry and failover yourself. The following code snippet is an example of the configuration that is required.

```
Map<String, Object> retryParams = new HashMap<>();
    retryParams.put("sleepTimeMs", SLEEP_TIME);
    retryParams.put("maxAttempts", MAX_ATTEMPTS);
    retryParams.put("timeoutMs", OVERALL_TIMEOUT);

    Map<String, Object> retryConfig = new HashMap<>();
    retryConfig.put("className", "com.hortonworks.registries.schemaregistry.retry.policy.ExponentialBackoffPolicy");
    retryConfig.put("config", retryParams);

    Map<String, ?> clientConfig = new HashMap<>();
    clientConfig.put("schema.registry.url", "url1,url2")
        clientConfig.put("schema.registry.client.retry.policy", retryConfig);
    clientConfig.put(...); // other client configuration
    SchemaRegistryClient client = new SchemaRegistryClient(clientConfig);
```

Solr

When upgrading from Cloudera Runtime 7.1.9 or higher, you can ensure zero downtime by meeting the following prerequisites for your Solr services before initiating an upgrade with rolling restart:

- The Solr service must have more than one server to perform a rolling restart. Otherwise the service will not be available during the restart.
- You are highly recommended to change the workload Solr collections to read-only before performing a rolling restart. Revert this setting after the upgrade is finished.
- Ensure you have replication factor set to greater than one for all workload Solr collections. If this criteria is not met, your queries may fail at the time of upgrade and your application may have to re-send the queries once the upgrade is successful.

SRM

The Streams Replication Manager (SRM) service in Cloudera consists of multiple service role types. Each role type represents core features in SRM. You must ensure that these prerequisites are met to ensure that the service can be upgraded with zero downtime.

Although the SRM service supports ZDU if the following prerequisites are met, replication monitoring provided by SMM is briefly interrupted during the upgrade because SMM does not support ZDU.

SRM Driver

Ensure that your SRM Driver roles are running in HA mode. That is, you have more than a single SRM Driver role deployed in the cluster. Cloudera recommends a minimum of 3.

SRM Service

- Ensure that your SRM Service roles are running in HA mode. That is, you have more than a single SRM Service role deployed in the cluster. Cloudera recommends a minimum of 3.
- If you have external clients that connect to the SRM Service REST API, ensure that you are using a load balancer to redirect REST API calls. If you do not have a load balancer, expect some service downtime as the SRM Service provides no automatic failover between its REST endpoints.
- If you are upgrading from Cloudera Runtime 7.1.8 or lower to Cloudera Runtime 7.1.9 or higher, you must enable the migration of metrics before you upgrade. Although SRM does not experience service downtime if metrics are not migrated, aggregated metrics collected before the upgrade are lost. As a result, an SRM upgrade without metric migration does not guarantee complete zero downtime. For more information, see [Step 1: Getting Started Upgrading a Cluster](#).

YARN

ResourceManager Work Preserving Recovery

YARN work preserving recovery must be enabled. Ensure that the Services YARN Configs Advanced property `yarn.resourcemanager.work-preserving-recovery.enabled` property is set to true.

ResourceManager HA

ResourceManager HA should be enabled to prevent a disruption in service during the upgrade.

This prerequisite is OPTIONAL

Upgrade from Cloudera Runtime 7.1.8 or 7.1.9 or 7.1.9 SP1 to Cloudera Base on premises 7.3.1

For security reasons, Cloudera Base on premises 7.3.1. has the `mapreduce.shuffle.ssl.enabled` option turned on. If this option is not set to true before the upgrade, already running MR2 applications fail during the upgrade in the shuffle phase with the `java.io.IOException: Exceeded MAX_FAILED_UNIQUE_FETCHES; bailing-out exception`.

When no MR2 jobs are running, configure the `mapreduce.shuffle.ssl.enabled` option to true before the upgrade. Restart the YARN service and deploy the client configurations.

Upgrade from Cloudera Runtime 7.1.8 to Cloudera Runtime 7.1.9 or 7.1.9 SP1

Upgrading from Cloudera Runtime 7.1.8 to Cloudera Runtime 7.1.9 or 7.1.9 SP1 displays the error `TestDFSIO intermittently failing with the RuntimeException: native snappy library not available`.

You must retry the job.

Check and enable High Availability service for your components

You must check and enable the High Availability support for your service component before you proceed with the upgrade process. High Availability is required to perform the ZDU process.

Service Component	HA Mode
Atlas	HA Mode Supported: Active-Passive
HBase	For HBase, if you have more than one master node and you use HDFS to store the data, then HBase is HA enabled. HA Mode Supported: Active-Passive
HDFS	HA Mode Supported: Active-Passive
Hive	HA Mode Supported: Active-Active
Hive on Tez	Add a HiveServer role and configuring HMS for high availability
Hue	Configuring high availability for Hue

Kafka	Kafka service is HA enabled by design. Multiple brokers have to be deployed to different hosts. Kafka Connect: Similar to the core Kafka, multiple Connect Workers have to be deployed.
Kafka Cruise Control	HA is not supported, Kafka Cruise Control is a single instance.
Key HSM	HA is not supported
Knox	HA Mode Supported: Active-Active
Kudu	Kudu ZDU requires a multi-master setup (at least 3 masters). If you have only 1 master, then follow the documentation.
Livy	HA Mode Supported: Active-Passive
Oozie	HA Mode Supported: Active-Active
Ozone	HA Mode Supported: Active-Passive
Phoenix	HA Mode Supported: Active-Active
Ranger	HA Mode Supported: Active-Active
Ranger KMS	HA Mode Supported: Active-Active
Schema Registry (SR)	Multiple SR servers can run on different hosts behind Knox, connecting to the same database.
Solr	HA Mode Supported: Active-Passive
Spark	HA Mode Supported: Active-Active
Sqoop	HA Mode Supported: Active-Passive
Stream Replication Manager	Enable high availability for Streams Replication Manager
YARN	HA Mode Supported: Active-Passive
ZooKeeper	HA Mode Supported: Active-Passive For ZooKeeper to be HA enabled, Cloudera recommends you to use ZooKeeper with 3 or 5 nodes. Note: One node is insufficient for ZooKeeper to be HA enabled.

Related Information

[About Atlas High Availability](#)

[Cloudera Manager HA](#)

[Enable HBase high availability using Cloudera Manager](#)

[HDFS HA Configuration](#)

[Adding a HiveServer role](#)

[Configuring HMS for high availability](#)

[Configuring high availability for Hue](#)

[Generate and configure a signing keystore for Knox in HA](#)

[Migrate to a multiple Kudu master configuration](#)

[Configuring Ranger High Availability](#)

[Configuring Ranger KMS High Availability](#)

[Improving performance in Schema Registry](#)

[Cloudera Search Architecture](#)

[Enable high availability for Streams Replication Manager](#)

[Configuring YARN ResourceManager high availability](#)

ZDU Component Support

The following components are supported when upgrading to Cloudera Runtime 7.1.9 and higher.

For the definition of ZDU terminologies, see [Glossary of terminologies](#).

Cloudera Base on premises 7.3.1

This table explains the services a component supports from the source cluster version (7.1.7 SP3, 7.1.9 SP1, 7.1.9, and 7.1.8) when upgrading to Cloudera Base on premises 7.3.1.

Component	7.1.7 SP3	7.1.8	7.1.9/7.1.9 SP1
Atlas *	Quick restart	ZDU	ZDU
HBase	ZDU	ZDU	ZDU
HDFS	ZDU	ZDU	ZDU
Hive *	ZDU	ZDU	ZDU
Hue *	Quick restart	Quick restart	Quick restart
Impala *	Quick restart	Quick restart	ZDU **
Kafka	ZDU	ZDU	ZDU
Knox *	ZDU	ZDU	ZDU
Kudu	Quick restart	Quick restart	ZDU
Livy3	Quick restart	Quick restart	ZDU
Omid	Quick restart	Quick restart	ZDU
Oozie	Quick restart	Quick restart	Quick restart
Ozone	Quick restart	Quick restart	Quick restart
Phoenix	Quick restart	Quick restart	ZDU
Ranger	Quick restart	Quick restart	Quick restart
Ranger KMS	Quick restart	Quick restart	Quick restart
Schema Registry	Quick restart	ZDU	ZDU
SMM	Quick restart	Quick restart	Quick restart
Solr Infra	Quick restart	Quick restart	ZDU
Solr Workload	Quick restart	Quick restart	ZDU
Spark3	Quick restart	Quick restart	ZDU
SRM	ZDU	ZDU	ZDU
YARN	ZDU	ZDU	ZDU
Zookeeper	ZDU	ZDU	ZDU

*

You must review the [Service component limitations](#) documentation for components that support ZDU with limitations.

**

Impala does not support ZDU upgrades from 7.1.9. Impala ZDU support starts from 7.1.9 SP1. However, ZDU for Impala are not supported when upgrading to version 7.3.1.500. This is due to Thrift protocol incompatibilities that can cause queries to fail during upgrade.

Service components limitations

You should be aware of certain limitations and considerations related to service components such as Atlas, Kudu, and YARN, to ensure that your cluster can successfully complete the Zero Downtime Upgrade (ZDU) process.



Warning:

- If there is any failure during the upgrade process, you must contact Cloudera customer support.
- Rollback: For services that do not support ZDU, you may require data to be rolled back to the pre-upgrade state, in the event of a failed upgrade. This can be accomplished using the rollback steps. However, Cloudera recommends that you contact Cloudera customer support before executing the rollback instructions.
- Downgrade: For services that support ZDU, it is possible to retain customer data in the event of a failed upgrade. This can be accomplished using the downgrade steps. This process ensures that all data written is retained till the point of failure. However, Cloudera recommends that you contact Cloudera customer support before applying the downgrade instructions.
- Rolling back a partial upgrade of a ZDU service can result in data loss. If there is a failure during the ZDU process, you must contact Cloudera customer support.

Atlas

Certain Atlas limitations related to the ZDU process are as follows:

- While Atlas goes through the process of rolling upgrade, some downtime might be expected because Atlas does not support the Active-Active model. Failover takes time because Active-Passive is the currently supported model. While the Passive instance becomes Active, there is some downtime where Atlas is not reachable and the messages from clients are queued up in Kafka.
- Solr does not support Rolling Upgrade due to which Atlas REST requests fail during the Solr upgrade.

Hive

Hive Metastore (HMS) and HiveServer (HS2) services support rolling restarts and rolling upgrades if High Availability (HA) is enabled for the services. If you are upgrading from Cloudera Runtime 7.1.7 SP2 or CDP 7.1.8 to Cloudera Runtime 7.1.9, you cannot achieve a ZDU for the HS2 service because running queries are aborted during HS2 service restarts. The aborted queries must be manually submitted again because HS2 cannot automatically resubmit these queries.

HS2 has a graceful shutdown feature to overcome this ZDU limitation. Graceful shutdown ensures that HS2 waits for a specified time period (300 seconds by default) before shutting down, thereby allowing queries that are already running, to complete before HS2 stops. This feature aids in achieving a ZDU; however, since the feature is available from Cloudera Runtime 7.1.9 onwards, it can only be used for Cloudera Runtime upgrades from 7.1.9 to higher versions.

You must also be aware of certain limitations related to the Hive upgrade process:

Hive sticky sessions

When a client connection is established to Hive through Zookeeper or Knox, a fixed connection is established with one of the HS2 instances. This session is disconnected when HS2 restarts during a Rolling Upgrade. Knox then redirects the fresh connections to another HS2 host instance that is available. As a result, the upcoming queries in the earlier session are aborted and queries are not redirected to the other HS2 instance.

The queries can be submitted after a fresh connection from Beeline is established with the second HS2 instance.

However, during a Rolling Upgrade, you must know that the other HS2 instance is also bound to restart and you will encounter the same issue. Therefore, it is recommended that you do not submit new queries until all the HS2 instances are upgraded.



Note: If you are upgrading from Cloudera Runtime 7.1.9 to higher versions, you can leverage the HS2 graceful shutdown feature (available in Cloudera Runtime 7.1.9), which ensures that HS2 waits for a specified time period before shutting down, thereby allowing queries that are already running to complete before HS2 stops.

Hive ODBC connection lost during a Knox ZDU

During an upgrade of the Knox service, when the Knox service shuts down, the ODBC connection that is established with a Hive host breaks. The issue occurs because the ODBC driver does not accept the cookies that are passed by Knox.

If HS2 establishes a client connection using an ODBC driver and through the Knox load balancer, the Hive ODBC connection is lost when the Knox service stops during a Rolling Upgrade. The issue occurs because the ODBC driver does not accept the cookies that are passed by Knox.

Establish a new client connection.

Hue

Hue supports rolling restart and zero-downtime upgrades in High Availability (HA) clusters, where the Hue service downtime is significantly less when compared to non-ZDU upgrades. The total Hue service downtime depends on various factors such as the number of Hue service role instances, network latency, hardware and connectors used, and more.



Note: All Hue roles (namely the load balancer, Kerberos ticket renewer, and Hue server) are of the non-worker type. You must select one of the following options under the "Roles to include" section to ensure that the Hue service is restarted in the rolling restart mode, depending on your requirements:

- Non-Workers Only, or
- All Roles



Attention: If you are using the ZDU approach to upgrade to 7.1.9 SP1, then you must start Hue only after upgrading Cloudera Manager to 7.11.3 CHF 7, Python to version 3.10, and Cloudera Runtime to 7.1.9 SP1. Hue does not start after upgrading only Cloudera Manager and Python but not Cloudera Runtime.

Impala



Note: When upgrading to Cloudera Runtime 7.3.1.500, zero downtime upgrades (ZDU) for Impala are not supported for Cloudera deployments running Impala. This is due to Thrift protocol incompatibilities that can cause queries to fail during upgrade.

Impala supports rolling restarts and zero-downtime upgrades (ZDU) in High Availability clusters, significantly reducing Impala service downtime compared to non-ZDU upgrades..

During the restart process, some operations can degrade depending on the services - coordinator / catalog / executors, currently restarted. During catalog restart, metadata operations such as creating new tables, adding new partitions, invalidating metadata, and so on do not work until the restart process is complete. Starting the catalog server can delay some new queries due to the need to load metadata into the cache again. During ZDU, there might be a brief window of downtime during Catalog or StateStore failover. For details see [Failure detection for Catalog and StateStore](#)

During coordinator restarts, running queries can be killed and new queries can only be submitted to alternative coordinators. Thus Impala enables service failure tolerance but does not guarantee it. To minimize potential failures during ZDU, ensure that the coordinators are managed by a load balancer. While the statestore restarts, you may notice some degradation of admission control and cluster membership updates. If you have a read-only workload that you want to continue running across a rolling restart then you should tune the `shutdown_grace_period_s` and `graceful_shutdown_deadline` parameters. The `shutdown_grace_period_s` time specifies the minimum time the daemon will wait before exiting. The default `shutdown_grace_period_s` value of 120 seconds means that rolling restart may take a long time if you have many daemons. Changing the `shutdown_grace_period_s` parameter will not take effect until after the next restart. The `graceful_shutdown_deadline` parameter specifies the maximum time that Cloudera Manager will wait for a daemon to exit, after the queries have finished. The `graceful_shutdown_deadline` parameter should be large enough to allow queries to drain, and should also be larger than `shutdown_grace_period_s`. During

the graceful shutdown period, any new requests would receive a `SERVER_SHUTTING_DOWN` error. If the load balancer is unable to handle this error during the ZDU, the request would fail, and the load balancer may not be able to detect that the coordinator is unavailable.

You can use the option of `retry_failed_queries` to automatically resubmit failed select queries due to executor daemons restarting during an upgrade. This option only supports select queries and will not support insert / drop / truncate / create queries.



Note: If you are restarting after an upgrade then cross-version incompatibilities may prevent Impala daemons from communicating between different versions during a rolling restart. Impala will function normally once all the components have restarted.

Knox

During ZDU, some client requests might fail with HTTP 503 responses. Knox starts listening on port 8443 before having completed the activation of all topologies, during which time the endpoints associated with the inactive topologies are temporarily unavailable. Any component proxied by Knox is affected (for example, Schema Registry).

As a workaround, clients should respond to HTTP 503 responses with retries.

Kudu

Cloudera recommends that you do not enable the following Kudu features until the upgrade is finalized:

- Auto-increment
- Data encryption

This is because the downgrade process is not supported if any of these features have been enabled.

YARN Queue Manager

Although Queue Manager does not support rolling upgrade or rolling restart, the amount of time that Queue Manager is down during a rolling upgrade is minimal. When YARN is not available during an upgrade, the Queue Manager UI functions in a read-only mode. This means that the Queue Manager UI displays the last-known state of the system, and you can not change queue settings, placement rules, scheduling rules, and so on. Additionally, you can not make changes through the Queue Manager APIs; therefore, you can not change the YARN Capacity Scheduler configuration.

Ozone

- When you upgrade from the lower version of Cloudera Runtime to Cloudera Runtime 7.1.9, the quota related information will be repaired during the cluster upgrade. The upgrade activity will take time based on number of keys present in the system. This is a one time activity to correct the quota and usages information for space and namespace usages.
- If you have the Ozone HttpFS role added to the Ozone service on your 7.1.8 cluster, you must stop and delete the Ozone HttpFS role from the Ozone service before upgrading the Cloudera Runtime cluster to 7.1.9. After you upgrade the Cloudera Runtime cluster to 7.1.9, you can add the HttpFS role back to the Ozone service.

OMID

OMID constantly checks the service status with ZooKeeper because OMID depends on ZooKeeper service while in HA mode. This increases the network traffic and might impact the service performance.

Schema Registry

The Schema Registry service supports rolling restarts and rolling upgrades if it is running in HA mode. This means that the individual service role instances (the Schema Registry server roles) are stopped and started in a rolling fashion during an upgrade. This allows the service to experience no downtime during an upgrade.

However, to guarantee zero downtime, all clients that connect to Schema Registry must be configured with proper retry and failover mechanisms. Otherwise, you might experience service outages on the client's side. For example, clients might try to connect to a server instance that is temporarily down during the upgrade, if retry and failover is not configured, the client will timeout and fail, resulting in service disruptions.

The Cloudera distributed Schema Registry Java client only supports retry and failover in Cloudera Runtime versions 7.1.9 or higher. As a result of this, Schema Registry's guarantees zero downtime during upgrades differ depending on your upgrade path as well as your deployment and use case.



Note: If Schema Registry is accessed through Knox, the Knox service must also run in HA mode for zero downtime during upgrades.

Upgrade from Cloudera Runtime 7.1.7 or lower

Schema Registry does not support ZDU for upgrading from Cloudera Runtime 7.1.7 or lower to a higher version.

Upgrade from Cloudera Runtime 7.1.8 to 7.1.9 or higher

If you have services or components (for example, Kafka and Kafka Connect) that run within the Cloudera Runtime cluster and connect to Schema Registry, zero downtime cannot be ensured for this upgrade path. This is because cluster internal services use the Cloudera distributed Schema Registry Java client and the client does not properly support retry and failover in versions 7.1.8 or lower. Additionally, the client version for cluster internal components cannot be upgraded separately before upgrading the cluster.

Upgrading from Cloudera Runtime 7.1.9 to a higher version

Schema Registry provides full guarantees for zero downtime as long as the service is in HA mode and all external applications implement proper retry and failover behavior.

Solr

Upgrade from Cloudera Runtime 7.1.8 or lower

Solr does not support ZDU for upgrading from Cloudera Runtime 7.1.8 or lower to a higher version.

Upgrade from Cloudera Runtime 7.1.9 or higher

Solr supports rolling restart when upgrading from Cloudera Runtime 7.1.9 or higher, provided it is running in HA mode. This means that the individual service nodes are stopped and started in a rolling fashion during an upgrade.

Spark History Server

Spark and Spark 3 applications can continue to run during rolling upgrades, but the old parcels (CDH/CDS) must not be removed while such an application runs. The old parcels can be removed once all the Spark applications have restarted.

Starting from Cloudera Runtime 7.1.9, SparkHistoryServer supports HA. Do not enable HA until after the upgrade is finalized, to avoid issues during downgrade.

Livy

Livy does not require data backups. Starting from Cloudera Runtime 7.1.9, Livy supports HA. Do not enable HA until after the upgrade is finalized, to avoid issues during the downgrade.

Glossary of terminologies

Review and understand the ZDU terminologies before performing the cluster upgrade.

- **Service interruption:** Any event that makes part of a component's services unavailable, including but not limited to client APIs, web UIs, or admin interfaces.

- Rolling Upgrade - Process of upgrading all the components without any service interruptions. Zero Downtime upgrade and upgrade by rolling restart are synonyms of Rolling Upgrade.
- Rolling Restart - Ability to restart a service without any service interruption to a component or dependent components.
- High Availability (HA) - The ability to avoid a service interruption when individual components fail within a service.
- Quick Restart - Component restart with a service interruption but with minimal downtime.
- Worker Roles - Roles that are related to data storage and processing. For example, HDFS has Datanode and Impala has Impala Daemon, and so on.
- Non-Worker - Roles that are related to master, admin, coordinator, or non-data-related activities. For example, HDFS-Namenode, Journalnode Impala- StateStore, Catalog, Hue's non-worker roles, such as the load balancer, Kerberos ticket renewer, and Hue server, and so on.
- Downgrade - Downgrade restores the software to the prior release version but data remains intact. Service interruptions may or may not be required for the downgrade process, depending on the component.
- Rollback - Rollback the state of the data of the upgraded components to the pre-upgrade state. A Rollback may require a service interruption and is not an automated operation.

Upgrading a Cloudera Runtime Cluster

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)

This topic describes how to upgrade a CDH or Cloudera Runtime cluster in any of the following scenarios:

- From Cloudera Runtime to a higher version of Cloudera Runtime.
- Upgrade Cloudera Runtime to maintenance releases.



Attention: To upgrade to Cloudera Manager or CDH 5.x or 6.x, do not use the instructions on this page. See the [Cloudera Enterprise Upgrade Guide](#).

When you upgrade a cluster, you use Cloudera Manager to upgrade the cluster software across an entire cluster using Cloudera Parcels. Package-based installations are not supported for Cloudera Runtime and Cloudera Base on premises upgrades. You must transition your CDH clusters to use Parcels before upgrading to Cloudera Base on premises. See [Migrating from Packages to Parcels](#).

Cluster upgrades update the Hadoop software and other components. You can use Cloudera Manager to upgrade a cluster for major, minor, and maintenance upgrades. The procedures vary depending on the version of Cloudera Manager you are using, the version of the cluster you are upgrading, and the version of Cloudera Runtime you are upgrading to.

Ranger plugins use a local copy of policies (client-side caching), so even if Ranger Admin server is not available during the schema update part of the process, upgrades do not impact Ranger authorization. In addition, Ranger Admin is deployed in active/active mode, so restarts do not impact the API interfaces. Ranger Admin HA will work with or without a load balancer. During Ranger Admin service upgrade, policy creates/updates are not available. This does not impact Ranger authorization since Ranger plugins continue to use the local cache.

After completing preparatory steps, you use the Cloudera Manager upgrade wizard to complete the upgrade. Cloudera Manager will restart all services after the upgrade. Services that support rolling restart will be restarted in a rolling fashion that does not require you to take the services offline during the upgrade. All other services will be restarted normally. The following services support rolling restart:

- Atlas
- HBase
- HDFS
- Hive-on-Tez
- Hive Metastore

- Ranger
- Hue
- Kafka
- Key Trustee Server
- Knox
- Kudu – see [Orchestrating a rolling restart with no downtime](#).
- MapReduce
- Oozie
- Ranger KMS
- Schema Registry
- YARN
- ZooKeeper



Warning: If there is any failure during the upgrade process, Cloudera recommends you to contact Cloudera customer support.

Step 1: Getting Started Upgrading a Cluster

Tasks you should perform before starting the upgrade.



Note: Not all combinations of Current Cluster Version to New Cluster Version are supported. Before you upgrade your Cloudera Base on premises cluster from one version to another, you must review table 2 to know the supported upgrades paths. For more information, see [Supported in-place upgrade paths](#).



Note: Not all combinations of Cloudera Manager and Cloudera Runtime are supported. Ensure that the version of Cloudera Manager you are using supports the version of Cloudera Runtime you have selected. For details, see [Cloudera Manager support for Cloudera Runtime, CDH and Cloudera Data Services on premises](#).



Note: CDP Private Cloud Data Services version 1.3.4 requires Cloudera Manager 7.5.5 and Cloudera Runtime version 7.1.6 or 7.1.7.



Important: You can upgrade to Cloudera Runtime 7.1.7 Service Pack 1 (7.1.7.1000) from all of Cloudera Runtime 7.1.x, CDH 5, and CDH 6 versions. An upgrade to Cloudera Manager 7.6.1 is required for Service Pack 1 (7.1.7.1000).



Note: If you are upgrading to Cloudera Manager 7.5.1 or higher in order to install CDP Private Cloud Data Services version 1.3.1, you must use Cloudera Runtime version 7.1.6 or 7.1.7.



Important: Upgrades from CDH 6.1, 6.2, and 6.3 are only supported for upgrades to CDP Private Cloud Base 7.1.7 or higher. Upgrades from CDH 6.0 are not supported.



Important: To upgrade to Cloudera Manager or CDH 5.x or 6.x, do not use the instructions on this page. See the [Cloudera Enterprise Upgrade Guide](#).



Warning: Upgrades to Cloudera Runtime 7.0.3 are not supported.



Note: Upgrades from CDH 6.x are supported only for upgrades to Cloudera Manager 7.4.4 or higher and Cloudera Runtime 7.1.7 or higher. Upgrades from CDH 6.0 are not supported.



Warning: Upgrades from CDH 5.12 and lower to Cloudera Base on premises are not supported. You must upgrade the cluster to CDH versions 5.13 - 5.16 before upgrading to Cloudera Base on premises.

The version of CDH or that you can upgrade to [depends on the version of](#) that is managing the cluster. You may need to [upgrade](#) before upgrading your clusters. Upgrades are not supported when using 7.0.3.

Before you upgrade a cluster,

- You need to gather information, review the limitations and release notes and run some checks on the cluster. See the [Collect Information](#) section below. Fill in the My Environment form below to customize your upgrade procedures.
- You must clean all the HBase Master procedure stores. For more details, see [Clean the HBase Master procedure store](#).

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)



Important: If you have any add-on services installed using a CSD (Custom Service Descriptor), you must use Cloudera Manager 7.1.1 or higher to install the CDH 6 version of the CSD before upgrading the cluster to Cloudera Runtime 7.1.1 or higher. During the upgrade, Cloudera Manager will prompt you to also install the Cloudera Runtime 7 version of the CSD. Cloudera Manager version 7.1.4 or higher will prompt you to install any required intermediate versions of the CSD.

To successfully complete the upgrade you must have the CDH 5, CDH 6, and Cloudera Runtime 7.x.x versions of the CSD installed. After the upgrade, you can delete the CDH 5 and CDH 6 versions of the add-on service.

This affects the following Cloudera services: CDSW, Nifi, and Nifi Registry as well as any CSDs created by third parties. See [Add-on Services](#).



Note: Isilon is not supported for CDP Private Cloud Base version 7.1.5 and lower.



Note: If your cluster uses Compute clusters in a Virtual Private Cluster (VPC) architecture, you must remove the compute cluster before upgrading the Base cluster. You can recreate the Compute cluster after the upgrade



Note:

After upgrading from CDH to CDP Private Cloud Base, the NodeManager recovery feature is enabled by default. This means that the `yarn.nodemanager.recovery.enabled` property is set to true. Cloudera recommends that you keep the NodeManager recovery feature enabled. If you set this property to false in your CDP Private Cloud Base cluster and then upgrade to a later CDP Private Cloud Base version, the feature will remain disabled.



Important:

In Cloudera Runtime 7.1.6 and higher, the way Streams Messaging Manager (SMM) integrates with Streams Replication Manager (SRM) has changed. SMM can only connect to and monitor an SRM service that is running in the same cluster as SMM. Monitoring an SRM service that is running in a cluster that is external to SMM is no longer supported.

Connectivity between the two services is disabled by default after a successful upgrade. If you want to continue using SMM to monitor SRM, you must reconnect the two services following the upgrade.



Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported



Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher



Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.



Warning: Before upgrading CDH 5 clusters with Sentry to 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.



Important: If HDFS ACL sync is enabled (`hdfs_sentry_sync_enable=true`) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see [Installing Ranger RMS](#).



Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

Collect Information

Collect the following information about your environment and fill in the form above. This information will be remembered by your browser on all pages in this Upgrade Guide.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Run the following command to find the current version of the Operating System:

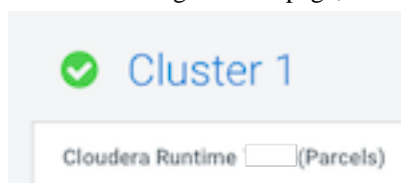
```
lsb_release -a
```

3. Log in to the Cloudera Manager Admin console and find the following:

- a. The version of Cloudera Manager used in your cluster. Go to [Support About](#).
- b. The version of the JDK deployed in the cluster. Go to [Support About](#).
- c. Whether High Availability is enabled for HDFS. Go to [Cloudera Manager Clusters HDFS](#)

If you see a standby namenode instead of a secondary namenode listed under [Cloudera Manager HDFS Instances](#), then High Availability is enabled.

- d. The current cluster version (parcels). The cluster version number along with (parcels) are displayed on the Cloudera Manager Home page, to the right of the cluster name.



Preparing to Upgrade a Cluster

1. You must have SSH access to the Cloudera Manager server hosts and be able to log in using the root account or an account that has password-less sudo permission to all the hosts.

2. Review the Requirements and Supported Versions for the new versions you are upgrading to. See: [Cloudera Base on premises 7.x.x Requirements and Supported Versions](#) If your hosts require an operating system upgrade, you must perform the upgrade before upgrading the cluster. See [Upgrading the Operating System to a new Major Version](#) on page 10.
3. Ensure that a supported version of Java is installed on all hosts in the cluster. See the links above. For installation instructions and recommendations, see [Upgrading the JDK](#) on page 28.
4. Review the following documents:

Cloudera Runtime 7.x.x

- Review the following when upgrading to Cloudera Runtime 7.x.x or higher:

[Cloudera Base on premises 7.x.x Requirements and Supported Versions](#)

5. If your deployment has defined a Compute cluster and an associated Data Context, you will need to delete the Compute cluster and Data context before upgrading the base cluster and then recreate the Compute cluster and Data context after the upgrade.
See [Starting, Stopping, Refreshing, and Restarting a Cluster](#) and [Virtual Private Clusters and Cloudera SDX](#).
6. Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
7. If the cluster uses Impala, check your SQL against the newest reserved words listed in [incompatible changes](#). If upgrading across multiple versions, or in case of any problems, check against the full list of [Impala reserved words](#).
8. If the cluster uses Hive, validate the Hive Metastore Schema:
 - a. In the Cloudera Manager Admin Console, Go to the Hive service.
 - b. Select ActionsValidate Hive Metastore Schema.
 - c. Fix any reported errors.
 - d. Select ActionsValidate Hive Metastore Schema again to ensure that the schema is now valid.
9. Run the Security Inspector and fix any reported errors.

Go to Administration Security Security Inspector .

10. Log in to any cluster node as the hdfs user, run the following commands, and correct any reported errors:

```
hdfs fsck / -includeSnapshots -showprogress
```



Note: The fsck command might take 10 minutes or more to complete, depending on the number of files in your cluster.

```
hdfs dfsadmin -report
```

See [HDFS Commands Guide](#) in the Apache Hadoop documentation.

11. Log in to any DataNode as the hbase user, run the following command, and correct any reported errors:

```
hbase hbck
```

12. If the cluster uses Kudu, log in to any cluster host and run the ksck command as the kudu user (sudo -u kudu). If the cluster is Kerberized, first kinit as kudu then run the command:

```
kudu cluster ksck <master_addresses>
```

For the full syntax of this command, see [Checking Cluster Health with ksck](#).

13. If your cluster uses Impala and Llama, this role has been deprecated as of CDH 5.9 and you must remove the role from the Impala service before starting the upgrade. If you do not remove this role, the upgrade wizard will halt the upgrade.

To determine if Impala uses Llama:

- a. Go to the Impala service.
- b. Select the Instances tab.
- c. Examine the list of roles in the Role Type column. If Llama appears, the Impala service is using Llama.

To remove the Llama role:

- a. Go to the Impala service and select **Actions Disable YARN and Impala Integrated Resource Management**. The Disable YARN and Impala Integrated Resource Management wizard displays.
- b. Click Continue.

The Disable YARN and Impala Integrated Resource Management Command page displays the progress of the commands to disable the role.

- c. When the commands have completed, click Finish.

14. If your cluster uses the Ozone technical preview, you must stop and delete this service before upgrading the cluster.

15. If your cluster uses Kafka, you must explicitly set the Kafka protocol version to match what's being used currently among the brokers and clients. Update kafka.properties on all brokers as follows.

- a. Log in to the Cloudera Manager Admin Console.
- b. Choose the Kafka service.
- c. Click Configuration.
- d. Use the Search field to find the Kafka Broker Advanced Configuration Snippet (Safety Valve) for kafka.properties configuration property.
- e. Add the following properties to the snippet.

- `inter.broker.protocol.version = [***CURRENT KAFKA VERSION***]`
- `log.message.format.version = [***CURRENT KAFKA VERSION***]`

Replace `[***CURRENT KAFKA VERSION***]` with the version of Apache Kafka currently being used. See [Cloudera Runtime component versions](#) for the Apache Kafka versions shipped with Cloudera Runtime. Make sure that you verify the component version information against the documentation version applicable to your upgrade scenario.

Additionally, make sure you enter full Apache Kafka version numbers with three values, such as 0.10.0. Otherwise, you will see an error message similar to the following:

```
2018-06-14 14:25:47,818 FATAL kafka.Kafka$:
java.lang.IllegalArgumentException: Version `0.10` is not a valid version
    at kafka.api.ApiVersion$$anonfun$apply$1.apply(ApiVersion.scala
a:72)
    at kafka.api.ApiVersion$$anonfun$apply$1.apply(ApiVersion.scala
a:72)
    at scala.collection.MapLike$class.getOrElse(MapLike.scala:128)
```

16. If your cluster uses Streams Replication Manager and you configured the Log Format property, you must take note of your configuration. The value set for Log Format is cleared during the upgrade and must be manually reconfigured following the upgrade.
17. If your cluster uses Streams Replication Manager and you are affected by OPSAPS-62546, ensure that you apply the workaround detailed in the Known issue. If the workaround is not applied, you might run into issues after the upgrade. For more information, see the [SRM Known Issues](#).

18. If your cluster uses Streams Replication Manager, export or migrate aggregated metrics.

In Cloudera Runtime 7.1.9, major changes are made to the internal Kafka Streams application of SRM. As a result, SRM by default loses all aggregated metrics that were collected before the upgrade. This means that you will not be able to query metrics with the SRM Service REST API that describe the pre-upgrade state of replications. If you want to retain the metrics, you can either export them, for archival purposes, or migrate them to the new format used by SRM. If you do not need to retain metrics, you can skip this step and continue with the upgrade.

Exporting metrics creates a backup of the metric data, however, exported metrics cannot be imported into the SRM Service for consumption. As a result, exporting metrics is only useful for data archival purposes.

Migrating metrics can be done in two different ways depending on whether you are doing a rolling upgrade or a non-rolling upgrade.

- In case of a non-rolling upgrade, migration happens following the upgrade. In this case, the new version of the internal Kafka Streams application running in the upgraded cluster starts to process historical metrics as soon as it is online. However, until the metrics are processed, the SRM Service cannot serve requests regarding latest metrics and returns empty or missing responses on its REST API. The duration of this downtime depends on the number SRM Service instances and the amount of metrics in the cluster.
- In case of a rolling upgrade, a migration process called SRM Service Migrator is initiated during the upgrade. The Migrator processes existing metrics so that they become compatible with your upgraded cluster. Depending on the size of your cluster and the amount of metrics you have, this process may take up to multiple hours to finish.

For Export metrics

Use the following endpoints of the SRM Service REST API to export metrics.

If upgrading from Cloudera Runtime 7.1.8:

- `/v2/topic-metrics/{source}/{target}/{upstreamTopic}/{metric}`
- `/v2/cluster-metrics/{source}/{target}/{metric}`

If upgrading from Cloudera Runtime 7.1.7 or lower:

- `/topic-metrics/{topic}/{metric}`
- `/cluster-metrics/{cluster}/{metric}`

For more information regarding the SRM Service REST API, see [Streams Replication Manager Service REST API](#) or [Streams Replication Manager REST API Reference](#).

For Non-rolling upgrade migration

- In Cloudera Manager, select the SRM service.
- Go to Configuration.
- Add the following to the SRM Service Environment Advanced Configuration Snippet (Safety Valve) property:

```
Key: SRM_SERVICE_SKIP_MIGRATION
Value: false
```

For Rolling upgrade migration



Note: All SRM Service role hosts experience increased resource utilization during a rolling upgrade if you enable migration. This is because the migration process uses the same system configurations (for example, heap size) as the SRM Service role.

- Ensure that the target clusters of the SRM Service are available and healthy.

If a target cluster is unavailable, the upgrade will fail. As a result, if a target cluster is unavailable, or you expect a target cluster to become unavailable during the upgrade, remove it from SRM's configuration for the duration of the upgrade. Metrics in the target clusters that you

remove are not migrated. Target clusters are specified in Streams Replication Manager Service Target Cluster.

- b. In Cloudera Manager, select the SRM service and go to Configuration.
- c. Add the following to the SRM Service Environment Advanced Configuration Snippet (Safety Valve) property:

```
Key: SRM_SERVICE_SKIP_MIGRATION
Value: false
```

- d. Fine-tune the behavior of the migration process.

The SRM Service Metrics Migrator (the migration process) has a number of user configurable properties. Fine tuning the configuration can help in reducing the time it takes to migrate the metrics.

These properties do not have dedicated entries in Cloudera Manager, instead you must use SRM Service Advanced Configuration Snippet (Safety Valve) for srm-service.yaml to configure them. If you are unsure about configuration, skip configuration and continue with the next step.

Table 3: SRM Service Migrator properties and recommendations

Property	Default Value	Description	Cloudera recommendations
streams.replication.manager.migrator.monitor.timeout.ms	3,600,000	The time in milliseconds after the Streams Replication Manager (SRM) Service Metrics Migrator times out.	<p>Set this timeout to a value that is higher than the expected migration time. Cloudera recommends a value that is at least three times the expected migration time.</p> <p>The migration time depends on the amount of metrics in your deployment. The higher the number of metrics, the longer the migration process, and the higher this property must be set.</p> <p>For example, a deployment with 10,000 partitions (100 topics with a 100 partitions each) and a 2 hour retention period produces, at minimum, 30,000 metrics per metric emission cycle. In a case like this, migration takes around 10 minutes to finish.</p>

Property	Default Value	Description	Cloudera recommendations
streams.replication.manager.migrator.monitor.backoff.ms	120,000	The frequency at which the progress of the Streams Replication Manager (SRM) Service Metrics Migrator is checked.	<p>The recommended values for this property differ depending on the version you are upgrading from.</p> <p>If upgrading from 7.1.8:</p> <p>Set this property to a value that is identical with or similar to the interval set in SRM Service Streams Commit Interval. The default value of this property is identical with the default value of SRM Service Streams Commit Interval.</p> <p>If upgrading from 7.1.7 or lower:</p> <p>Set this property to 30,000 (30 seconds).</p>
streams.replication.manager.migrator.monitor.stopp.delay.ms	60,000	The amount of time in milliseconds that the Streams Replication Manager (SRM) Service Metrics Migrator processes metrics after the streams application is considered caught up	
streams.replication.manager.migrator.monitor.minimum.consecutive.successful.checks	3	The number of consecutive checks where the lag must be within the configured threshold to consider the Streams Replication Manager (SRM) Service Metrics Migrator successful. All target clusters of the SRM Service must be caught up for a check to be successful.	

Property	Default Value	Description	Cloudera recommendations
streams.replication.manager.migrator.monitor.max.offset.lag	Calculated automatically if left empty.	The amount of offsets that the streams application in the Streams Replication Manager (SRM) Service Metrics Migrator is allowed to lag behind and still be considered up to date. When left empty, the migration logic will automatically calculate the maximum offset lag based on the Kafka Streams application configuration and the amount of metrics messages. Low offset lag values result in more up-to-date metrics processing following the upgrade, but also increase the time required for the upgrade to finish.	An appropriate value for this property is calculated automatically if the property is left empty. As a result, Cloudera recommends that you leave this property empty and use the automatically calculated value.

- e. Click Save Change.
- f. Restart the SRM service.

19. If your cluster uses Cruise Control and you have customized the goals in Cruise Control, ensure that you have created a copy from the values of the following goals before upgrading your cluster:

- Default goals
- Supported goals
- Hard goals
- Self-healing goals
- Anomaly detection goals

For more information, see the [Cruise Control Known Issues](#).

20. The following services are no longer supported as of CDP Private Cloud Base:

- Accumulo
- Sqoop 2
- MapReduce 1
- Record Service

You must stop and delete these services before upgrading a cluster.

21. Open the Cloudera Manager Admin console and collect the following information about your environment:

- a. The version of Cloudera Manager. Go to [Support About](#).
- b. The version of the JDK deployed. Go to [Support About](#).
- c. The version of CDH or Cloudera Runtime and whether the cluster was installed using parcels or packages. It is displayed next to the cluster name on the Home page.
- d. The services enabled in your cluster.

Go to [Clusters](#) *CLUSTER NAME*.

22. Back up Cloudera Manager before beginning the upgrade. Before upgrading a cluster, you should backup Cloudera Manager again, see [Step 4: Back Up Cloudera Manager](#).

Step 2: Review Notes and Warnings

Notes and warnings to consider before upgrading to Cloudera Base on premises.

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)

Note the following before upgrading your clusters:



Warning: If there is any failure during the upgrade process, you must contact Cloudera customer support.



Warning: If there is any failure during the Cloudera Runtime upgrade process, you must not delete or modify the records in the `upgrade_state` table. If you still want to delete or modify the `upgrade_state` table, you must contact the Cloudera development team for modification or deletion approval. If you proceed without approval from the Cloudera development team, it can cause additional issues while resolving the runtime upgrade failure.



Note: Cloudera recommends all upgrades to happen in a maintenance window by throttling and scaling down workloads during that time as a best practice.



Important:

Spark 3 is the default Spark version in Cloudera Runtime. Spark 2 (along with Livy 2) has been removed and no longer available in 7.3.1

Spark 3 contains a large number of changes from Spark 2. Refer to [Upgrading Spark](#) for more information on upgrading Spark clusters to 7.3.1, and [Migrating Spark Applications](#) for more information on migrating your existing Spark applications between versions 2 and 3.

**Important:**

- The embedded PostgreSQL database is NOT supported in production environments.
- The following services are no longer supported as of Enterprise 6.0.0:
 - Sqoop 2
 - MapReduce 1
 - Spark 1.6
 - Record Service
- Running Apache Accumulo on top of CDP Private Cloud Base 7.1.x cluster is not currently supported. If you try to upgrade to CDP Private Cloud Base 7.1.x, you will be asked to remove the Accumulo service from your cluster.
- Upgrading Apache HBase from CDH to Cloudera Runtime 7.1.1 gives you a warning in Cloudera Manager that the Dynamic Jars Directory feature property `hbase.dynamic.jars.dir` is deprecated. You can ignore this warning when using Apache HBase with HDFS storage on Cloudera Base on premises. The `hbase.dynamic.jars.dir` property is incompatible with Apache HBase on cloud deployments using cloud storage.
- The minor version of Cloudera Manager you use to perform the upgrade must be equal to or greater than the CDH or Cloudera Runtime minor version. Cloudera recommends that you upgrade to the latest maintenance version of Cloudera Manager before upgrading your cluster. See [Supported Upgrade Paths](#). To upgrade Cloudera Manager, see [Upgrading Cloudera Manager 7](#) on page 44.

For example:

- Supported:
 - Cloudera Manager 7.1 or higher and Cloudera Runtime 7.0
 - Cloudera Manager 7.1 and CDH 5.
 - Cloudera Manager 6.0.0 and CDH 5.14.0
 - Cloudera Manager 5.14.0 and CDH 5.13.0
 - Cloudera Manager 5.13.1 and CDH 5.13.3
- Not Supported:
 - Cloudera Manager 5.14.0 and CDH 6.0.0
 - Cloudera Manager 5.12 and CDH 5.13
 - Cloudera Manager 6.0.0 and CDH 5.6

**Note:**

After upgrading from CDH to CDP Private Cloud Base, the NodeManager recovery feature is enabled by default. This means that the `yarn.nodemanager.recovery.enabled` property is set to true. Cloudera recommends that you keep the NodeManager recovery feature enabled. If you set this property to false in your CDP Private Cloud Base cluster and then upgrade to a later CDP Private Cloud Base version, the feature will remain disabled.



Note: Upgrades to Cloudera Runtime 7.0.3 are not supported.

**Note:**

When upgrading CDH using Rolling Restart (Minor Upgrade only):

- Automatic failover does not affect the rolling restart operation.
- After the upgrade has completed, do not remove the old parcels if there are MapReduce or Spark jobs currently running. These jobs still use the old parcels and must be restarted in order to use the newly upgraded parcel.
- Ensure that Oozie jobs are idempotent.
- Do not use Oozie Shell Actions to run Hadoop-related commands.
- Rolling upgrade of Spark Streaming jobs is not supported. Restart the streaming job once the upgrade is complete, so that the newly deployed version starts being used.
- Runtime libraries must be packaged as part of the Spark application.
- You must use the distributed cache to propagate the job configuration files from the client gateway hosts.
- Do not build "uber" or "fat" JAR files that contain third-party dependencies or CDH/Cloudera Runtime classes as these can conflict with the classes that Yarn, Oozie, and other services automatically add to the CLASSPATH.
- Build your Spark applications without bundling CDH/Cloudera Runtime JARs.

Cruise Control might fail during an upgrade

Warning: Cruise Control might fail during the restart process when upgrading to CDP Private Cloud Base 7.1.4. For more information, see the [Cruise Control Release Notes](#).

Cruise Control does not function properly during the upgrade

Warning: Cruise Control does not work properly during the upgrade process of the Kafka service. This also applies to rolling upgrades.

Data replication with Streams Replication Manager might stop during a rolling upgrade

Important: In Cloudera Runtime 7.1.8, Streams Replication Manager (SRM) configuration properties and environment variables changed in a non-backward-compatible manner. As a result, during a rolling upgrade, SRM Driver roles may not be able to continue replicating data and parts of the replication work might stop until all SRM Drivers are restarted during the upgrade. The downtime differs depending on the number of SRM Drivers and the time it takes for rolling restarts to finish.

Prometheus instances used by Streams Messaging Manager require reconfiguration following an upgrade**Important:**

In Cloudera Runtime 7.1.9, a number of backward-incompatible changes are introduced for Kafka Connect. As a result, Prometheus instances set up as the metrics store for Streams Messaging Manager (SMM) will be unable to scrape Kafka Connect metrics.

Following an upgrade you must reconfigure your Prometheus instance, otherwise, you will not be able to continue monitoring Kafka Connect with SMM. Reconfiguration steps are provided in [Step 9: Complete Post-Upgrade steps for upgrades to CDP Private Cloud Base](#).

The integration between Streams Messaging Manager and Streams Replication Manager is changed



Important:

In Cloudera Runtime 7.1.6 and higher, the way Streams Messaging Manager (SMM) integrates with Streams Replication Manager (SRM) has changed. SMM can only connect to and monitor an SRM service that is running in the same cluster as SMM. Monitoring an SRM service that is running in a cluster that is external to SMM is no longer supported.

Connectivity between the two services is disabled by default after a successful upgrade. If you want to continue using SMM to monitor SRM, you must reconnect the two services following the upgrade.



Important: In Cloudera Runtime 7.1.8 and higher data at rest encryption for Kudu and range-specific hash schemas are supported features. However, if these features are used, a rollback is no longer possible. That is because a rollback would require to disable the encryption and delete all partitions with custom hash schemas.



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.

HDFS



Warning:

- Cloudera recommends you to not create any new encryption zone unless HDFS metadata is finalized.

Replication Manager

Before you use Replication Manager with CDP Private Cloud Base 7.1.9 SP1 CHF11 and Cloudera Manager 7.13.1.400, you must be aware that certain Cloudera Replication Manager fixes are not included in this combination. While general usage is supported, specific features or use cases might be impacted if they rely on the missing fixes.

To ensure full feature compatibility and to access the latest fixes, Cloudera recommends upgrading to Cloudera Manager 7.13.1.500 or higher versions when using CDP Private Cloud Base 7.1.9 SP1 CHF11.

Ozone



Important:

- Using Cloudera Manager 7.7.1, if you are upgrading from [CDP Private Cloud Base 7.1.8 with Ozone parcels](#) to CDP Private Cloud Base 7.1.9 using Cloudera Manager 7.11.3, then you must
 1. If you have [Ozone parcel 1.0](#) on the CDP Private Cloud Base 7.1.8 cluster, you must deactivate and undistribute Ozone parcel 1.0. If you have [Ozone parcel 2.x](#) on the CDP Private Cloud Base 7.1.8 cluster, you must only deactivate Ozone parcel 2.x.
 2. Upgrade Cloudera Manager from 7.7.1 to 7.11.3. Do not restart the Cloudera Base on premises cluster.
 3. Upgrade from CDP Private Cloud Base 7.1.8 to CDP Private Cloud Base 7.1.9.
- When you upgrade from the lower version of CDP Private Cloud Base to CDP Private Cloud Base 7.1.9, the quota related information will be repaired during the cluster upgrade. The upgrade activity will take time based on number of keys present in the system. This is a one time activity to correct the quota and usages information for space and namespace usages.
- If you have the Ozone HttpFS role added to the Ozone service on your CDP Private Cloud Base 7.1.8 cluster, you must stop and delete the Ozone HttpFS role from the Ozone service before upgrading the Cloudera Runtime cluster to CDP Private Cloud Base 7.1.9. After you upgrade the Cloudera Runtime cluster to CDP Private Cloud Base 7.1.9, you can add the HttpFS role back to the Ozone service.
- Non-HA Ozone upgrades are not supported using Cloudera Manager.
- Multiple Ozone deployments are not supported in the Cloudera cluster. Ensure that you have only one Ozone deployment Service ID in the ozone.om.service.ids configuration of the Cloudera cluster. If you have multiple Ozone deployment Service IDs in the ozone.om.service.ids configuration, then remove all of them except one.

Oozie ShareLib update and authorization

Updating the Oozie ShareLib is considered as an admin operation. If you have configured authorization in Oozie, then only admin users are able to trigger a ShareLib update.



Important: During a runtime upgrade, Cloudera Manager triggers a ShareLib update automatically. Please ensure that the admin users are configured correctly or this operation fails, resulting in an upgrade failure.

Hive metastore

If you have created any materialized views or MSSQL indexed views on Hive backend schemas, such as SYS or INFORMATION_SCHEMA tables, your upgrade process can fail when the upgrade SQL statements are trying to drop these tables.

You must drop the materialized views before performing an upgrade and then recreate the views after the upgrade process is complete.



Important: Cloudera recommends that you do not create any materialized views or indexed views on the SYS and INFORMATION_SCHEMA tables. However, if you have already created the views, then perform the following steps to identify such views and drop them before upgrading the cluster.

1. Ensure that you have backed up the Hive metastore (HMS) backend database before dropping materialized views.
2. Start a Hive Beeline session and run the following query to identify materialized views that are created on top of the SYS and INFORMATION_SCHEMA tables:

```
SELECT DISTINCT d.DB_LOCATION_URI, d.NAME, t.TBL_NAME, t.TBL_TYPE, t.OWNER, t.VIEW_EXPANDED_TEXT
FROM sys.TBLS t
      INNER JOIN sys.DBS d ON t.DB_ID = d.DB_ID
      INNER JOIN sys.MV_CREATION_METADATA mv ON mv.TBL_NAME = t.TBL_NAME
      INNER JOIN sys.MV_TABLES_USED tu ON mv.MV_CREATION_METADATA_ID = tu.MV_CREATION_METADATA_ID
WHERE tu.TBL_ID IN (SELECT distinct t.TBL_ID
                    FROM sys.MV_CREATION_METADATA mv
                        INNER JOIN sys.MV_TABLES_USED tu ON mv.MV_CREATION_METADATA_ID = tu.MV_CREATION_METADATA_ID
                        INNER JOIN sys.TBLS t ON tu.TBL_ID = t.TBL_ID
                        INNER JOIN sys.DBS d ON t.DB_ID = d.DB_ID
                        WHERE lower(d.NAME) IN ('sys', 'information_schema'))
AND upper(t.TBL_TYPE) = 'MATERIALIZED_VIEW';
```

3. If the query returns any materialized views, drop each view using the DROP statement.

```
DROP MATERIALIZED VIEW [db_name.]materialized_view_name;
```

4. Upgrade the cluster and recreate the views after the upgrade process is complete.

Known issue during CDP Private Cloud Base 7.1.7 SP2 to CDP Private Cloud Base 7.1.9 upgrade

OPSAPS-68279: When upgrading CDP Private Cloud Base 7.1.7 SP2 to CDP Private Cloud Base 7.1.9, sometimes the command step DeployClientConfig fails due to the following error:

Error Message: Client configuration generation requires the following additional parcels to be activated: [cdh]

This can be because of the failure of the activation of the Cloudera Base on premises 7.1.9 parcels. To verify:

1. Navigate to the parcels page.
2. See if the following error is displayed: Error when distributing to <hostname>: Script file/opt/cloudera/parcels/flood/CDH-7.1.9-1.cdh7.1.9.p0.43968053-el7.parcel/CDH-7.1.9.0.43968053-el7.parcel does not exist.

3. Using the error above, identify the host and ssh into the host by running the command `ssh <hostname>`.
4. Navigate to the agent directory by running the command `cd /var/log/cloudera-scm-agent`.
5. Find the following pattern in agent log file(s) Exception: `Untar failed with return code: 2, with tar output: stdout: [b"], stderr: [b"\ngzip: stdin: invalid compressed data--format violated\n\tar: Unexpected EOF in archive\n\tar: Unexpected EOF in archive\n\tar: Error is not recoverable: exiting now\n"]`.
1. If the above exception appears, you must restart the agent on that host by running the command `systemctl restart cloudera-scm-agent`.
2. After the agent restarts, click resume to continue with the upgrade.

Known issue during CDP Private Cloud Base 7.1.7 to CDP Private Cloud Base 7.1.9 upgrade

Yarn service fails to start after upgrading CDP from 7.1.7 to 7.1.9

After the upgrade from CDP 7.1.7 to CDP 7.1.9 Yarn Service fails to start due missing permissions for the /yarn HDFS directory for the Yarn user.

Add the following post-upgrade steps:

1. Navigate to Cloudera Manager
2. Select Yarn Actions menu
3. Select Create YARN Node Label Root Directory

Phoenix

CDPD-67700/CDPQE-30593: While upgrading CDH 6 to CDP Private Cloud Base 7.1.7 SP3 using a Phoenix parcel, Phoenix drops the Tephra support and causes a classpath issue

This issue occurs because an old Phoenix parcel is used.

To resolve this issue, deactivate the old Phoenix parcel.

Step 3: Backing Up the Cluster

Steps to back up your cluster before the upgrade.

This topic describes how to back up a cluster managed by Cloudera Manager prior to upgrading the cluster. These procedures do not back up the data stored in the cluster. Cloudera recommends that you maintain regular backups of your data using the Backup and Disaster Recovery features of Cloudera Manager.

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)

The following components do not require backups:

- MapReduce
- YARN
- Spark
- Impala

Complete the following backup steps before upgrading your cluster:

Back Up Databases



Warning: Backing up databases requires that you stop some services, which might make them unavailable during backup.

Gather the following information:

- Type of database (PostgreSQL, Embedded PostgreSQL, MySQL, MariaDB, or Oracle)
- Hostnames of the databases
- Database names
- Port number used by the databases
- Credentials for the databases

Open the Cloudera Manager Admin Console to find the database information for any of the following services you have deployed in your cluster:

- Sqoop, Oozie, and Hue – Go to *CLUSTER NAME* Configuration Database Settings .



Note: The Sqoop Metastore uses a HyperSQL (HSQLDB) database. See the [HyperSQL](#) documentation for backup procedures.



Note: Sqoop 2 is not supported in CDP Private Cloud Base.


- Hive Metastore – Go to the Hive service, select Configuration, and select the Hive Metastore Database category.
- Sentry – Go to the Sentry service, select Configuration, and select the Sentry Server Database category.
- Ranger – Go to the Ranger service, select Configuration, and search on "database."
- Queue Manager – Go to the Queue Manager service, select the Configuration tab. In the List of Filters on the left side, click the Category drop-down and select Database.
- Schema Registry and Streams Messaging Manager – Select the service, go to Configuration, and select the Database category.

To back up the databases

Perform the following steps for each database you back up:

1. If not already stopped, stop the service.

a.

On the Home Status tab, click  to the right of the service name and select Stop.

b.

Click Stop in the next screen to confirm. When you see a Finished status, the service has stopped.

2. Back up the database. Substitute the database name, hostname, port, user name, and backup directory path and run the following command:

MySQL

```
mysqldump --databases
                DATABASE_NAME
                --host=DATABASE_HOSTNAME
                --port=DATABASE_PORT -u
                DATABASE_USERNAME -p >
                BACKUP_DIRECTORY_PATH/DATABASE_NAME-backup-
`date
                +%F`-CDH.sql
```

PostgreSQL/Embedded


```
pg_dump -h DATABASE_HOSTNAME -U DATABASE_USERNAME
-W -p DATABASE_PORT DATABASE_NAME
> BACKUP_DIRECTORY_PATH/DATABASE_NAME-backup-`date +%F`-CDH.sql
```

Oracle

Work with your database administrator to ensure databases are properly backed up.

For additional information about backing up databases, see these vendor-specific links:

- MariaDB 10.2: <https://mariadb.com/kb/en/backup-and-restore-overview/>
- MySQL 5.7: <https://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html>
- PostgreSQL10: <https://www.postgresql.org/docs/10/static/backup.html>

- Oracle 12c: <https://docs.oracle.com/en/database/oracle/oracle-database/12.2/bradv/index.html>
3. Start the service.
 - a. On the HomeStatus tab, click  to the right of the service name and select Start.
 - b. Click Start in the next screen to confirm. When you see a Finished status, the service has started.

Back Up ZooKeeper

1. On all ZooKeeper hosts, back up the ZooKeeper data directory specified with the Data Directory property and ZooKeeper transaction log directory specified with the Transaction Log Directory property in the ZooKeeper configuration. The default location for both these directories is `/var/lib/zookeeper`.

For example:

```
cp -rp /var/lib/zookeeper/ /var/lib/zookeeper-backup-`date +%F`CM-CDH
```

2. To identify the ZooKeeper hosts, open the Cloudera Manager Admin console and go to the ZooKeeper service and click the Instances tab.

Record the permissions of the files and directories; you will need these to roll back ZooKeeper.

Back up Solr

You are recommended to take a backup of Solr data and indexes before an upgrade, so that data can be restored during a rollback.

When backing up the Infra Solr service, make sure that you backup all of the following collections:

- vertex_index (Atlas)
- edge_index (Atlas)
- fulltext_index (Atlas)
- ranger_audits (Ranger)

Store them in a location that is accessible to the solr service user during a rollback. Cloudera recommends you target your backups to a shared file system, even if the Solr service uses local file system.

Before you start: Solr gateway has to be installed on all edge/client nodes.

- Create a backup of Solr collections.

For information on backing up Solr collections, see [Backing up a collection from HDFS](#) (default location for Workload Solr) or [Backing up a collection from local file system](#) (default location for Infra Solr), depending on the type of storage used.

Back Up HDFS

Follow this procedure to back up an HDFS deployment.



Note: To locate the hostnames required to backup HDFS (for JournalNodes, DataNodes, and NameNodes), open the Cloudera Manager Admin Console, go to the HDFS service, and click the Instances tab.

1. If high availability is enabled for HDFS, run the following command on all hosts running the JournalNode role:

```
cp -rp /dfs/jn /dfs/jn-CM-CDH
```

2. On all NameNode hosts, back up the NameNode runtime directory. Run the following commands:

```
mkdir -p /etc/hadoop/conf.rollback.namenode
```

```
cd /var/run/cloudera-scm-agent/process/ && cd `ls -tl | grep -e "-NAMENODE\n$" | head -1`
```

```
cp -rp * /etc/hadoop/conf.rollback.namenode/
```

```
rm -f /etc/hadoop/conf.rollback.namenode/log4j.properties
```

```
cp -rp /etc/hadoop/conf.cloudera.HDFS_SERVICE_NAME/log4j.properties /etc/hadoop/conf.rollback.namenode/
```

These commands create a temporary rollback directory. If a rollback is required later, the rollback procedure requires you to modify files in this directory.

3. Back up the runtime directory for all DataNodes. Run the following commands on all DataNodes:

```
mkdir -p /etc/hadoop/conf.rollback.datanode/
```

```
cd /var/run/cloudera-scm-agent/process/ && cd `ls -tl | grep -e "-DATANODE\n$" | head -1`
```

```
cp -rp * /etc/hadoop/conf.rollback.datanode/
```

```
rm -f /etc/hadoop/conf.rollback.datanode/log4j.properties
```

```
cp -rp /etc/hadoop/conf.cloudera.HDFS_SERVICE_NAME/log4j.properties /etc/hadoop/conf.rollback.datanode/
```

4. If high availability is not enabled for HDFS, backup the runtime directory of the Secondary NameNode. Run the following commands on all Secondary NameNode hosts:

```
mkdir -p /etc/hadoop/conf.rollback.secondarynamenode/
```

```
cd /var/run/cloudera-scm-agent/process/ && cd `ls -tl | grep -e "-SECONDARYNAMENODE\n$" | head -1`
```

```
cp -rp * /etc/hadoop/conf.rollback.secondarynamenode/
```

```
rm -f /etc/hadoop/conf.rollback.secondarynamenode/log4j.properties
```

```
cp -rp /etc/hadoop/conf.cloudera.HDFS_SERVICE_NAME/log4j.properties /etc/hadoop/conf.rollback.secondarynamenode/
```



Note: For more information on backing up HDFS, see [Checkpoint HDFS](#) documentation.

Back Up Key Trustee Server and Clients

For the detailed procedure, see [Backing Up Key Trustee Server and Clients](#).

Back Up HSM KMS

When running the HSM KMS in high availability mode, if either of the two nodes fails, a role instance can be assigned to another node and federated into the service by the single remaining active node. In other words, you can bring a node that is part of the cluster, but that is not running HSM KMS role instances, into the service by making it an HSM KMS role instance—more specifically, an HSM KMS proxy role instance and an HSM KMS metastore role instance. So each node acts as an online ("hot" backup) backup of the other. In many cases, this will be sufficient. However, if a manual ("cold" backup) backup of the files necessary to restore the service from scratch is desirable, you can create that as well.

To create a backup, copy the `/var/lib/hsmkp` and `/var/lib/hsmkp-meta` directories on one or more of the nodes running HSM KMS role instances.

To restore from a backup: bring up a completely new instance of the HSM KMS service, and copy the `/var/lib/hsmkp` and `/var/lib/hsmkp-meta` directories from the backup onto the file system of the restored nodes before starting HSM KMS for the first time.

Back Up Navigator Encrypt

It is recommended that you back up Navigator Encrypt configuration directory after installation, and again after any configuration updates.

1. To manually back up the Navigator Encrypt configuration directory (`/etc/navencrypt`):

```
$ zip -r --encrypt nav-encrypt-conf.zip /etc/navencrypt
```

The `--encrypt` option prompts you to create a password used to encrypt the zip file. This password is also required to decrypt the file. Ensure that you protect the password by storing it in a secure location.

2. Move the backup file (`nav-encrypt-conf.zip`) to a secure location.



Warning: Failure to back up the configuration directory makes your backed-up encrypted data unrecoverable in the event of data loss.

Back Up HBase

Because the rollback procedure also rolls back HDFS, the data in HBase is also rolled back. In addition, HBase metadata stored in ZooKeeper is recovered as part of the ZooKeeper rollback procedure.

If your cluster is configured to use HBase replication, Cloudera recommends that you document all replication peers. If necessary (for example, because the HBase znode has been deleted), you can roll back HBase as part of the HDFS rollback without the ZooKeeper metadata. This metadata can be reconstructed in a fresh ZooKeeper installation, with the exception of the replication peers, which you must add back. For information on enabling HBase replication, listing peers, and adding a peer, see [HBase Replication](#) in the CDH 5 documentation.

Back Up YARN Queue Manager

Learn how to back up Yarn Queue Manager for CDP Private Cloud Base versions 7.1.8 and below. These steps are necessary if you want to upgrade to CDP Private Cloud Base 7.1.9 from version 7.1.8 and below as there is no ability to roll back changes if a CDP Private Cloud Base 7.1.9 upgrade is unsuccessful.

1. In Cloudera Manager, navigate to `Clusters Hosts`. Backup the configuration service database.
2. Locate the host that has the Yarn Queue Manager Store running.
3. Find the location of the config-service database file by navigating to `Cluster QueueManager Service Configurations` tab `Scope`, and click the Yarn Queue Manager Store.
4. Locate the Location for config-service DB field. If the field is empty, then use the default location: Database Location -> `/var/lib/hadoop-yarn/`
5. Open a SSH terminal and enter the following command: `ssh [***your_username***]@[***queue_manager_host_ip_address***]`
6. Navigate to the directory where the configuration database file is stored: `cd {Database Location}`

7. Find two database files with these names: `-config-service.mv.db`
`-config-service.trace.db`
 Notice that `config-service.trace.db` is in the same location.
8. Secure copy the `config-service.mv.db` and `config-service.trace.db` files to the machines where the backups are to be stored. For example: `scp -i ~/.ssh/{ssh_key} config-service.mv.db ro ot@{hostName}:{Your_Backup_Folder}/config-service.mv.db`
9. Use `sha1sum` to verify that the files in the current host and the location of where the backup is stored have the same hash.

Back up Atlas

When you plan to back up your Atlas data, it is a two-step process where you must first back up Solr and HBase data before proceeding further.

Back up Solr

Follow the instructions to back up your data in Solr. You must run these commands on single solr server.

1. `curl -ivk "https://host1.example.com:8993/solr/admin/collections?action=BACKUP&name=vertex_index_bkp&collection=vertex_index&location=/tmp/"`
2. `curl -ivk "https://host1.example.com:8993/solr/admin/collections?action=BACKUP&name=edge_index_bkp&collection=edge_index&location=/tmp/"`
3. `curl -ivk "https://host1.example.com:8993/solr/admin/collections?action=BACKUP&name=fulltext_index_bkp&collection=fulltext_index&location=/tmp/"`



Note:

1. `/tmp/` this can be replaced with the backup directory path which is to be used to store Solr backup. This path needs to be accessible for the solr service user.
2. If the cluster is kerberized, then run `kinit` against Solr keytab first and add `--negotiate -u :` after the `-ivk` flag in the above curl commands. Example: `curl -ivk --negotiate -u : https://host1.example.com:8993...`
3. If you have multiple Solr services, ensure you create the Solr backup directories on all the services. Else the backup fails indicating that there should be a shared storage used for backup.
4. In some cases if Shards are present on different nodes, backup might fail with following message: `org.apache.solr.client.solrj.impl.HttpSolrClient$RemoteSolrException: Error from server at http://host1.example.com:8993/solr: Failed to backup core=vertex_index_shard because org.apache.solr.common.SolrException: Directory to contain snapshots doesn't exist: file:///tmp/vertex_index. Note that Backup/Restore of a SolrCloud collection requires a shared file system mounted at the same path on all nodes!"`
5. Solr recommends having a backup using HDFS repository in such a scenario. For more information, see [Backup and Restore Solr Repositories](#).

Back up HBase

Follow the instructions to back up your data in HBase:

If the cluster is kerberized, then run `kinit` against HBase keytab

1. Create HBase table snapshot:

a. hbase shell

```
hbase> snapshot 'atlas_janus', 'atlas_janus_snapshot_<insert-date-here>'

hbase> snapshot 'ATLAS_ENTITY_AUDIT_EVENTS',
'atlas_entity_audit_events_snap_<insert-date-here>'

# exit
```



Note: You can use the Table Browser in Cloudera Manager to take a snapshot from the atlas_janus table.

2. Export Snapshot from server terminal:

- a. `hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot 'atlas_janus_snapshot_<insert-date-here>' -copy-to /tmp/hbasebackup/`
- b. `hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot 'atlas_entity_audit_events_snap_<insert-date-here>' -copy-to /tmp/hbasebackup/`

The contents of '/tmp/hbasebackup/' contain the table backup.

In case of below error:

```
ERROR snapshot.ExportSnapshot: Snapshot export failed
org.apache.hadoop.security.AccessControlException: Permission denied:
user=hbase, access=WRITE, inode="/user":hdfs:supergroup:drwxr-xr-x at
org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.check(FSPermissionChecker.
java:553)
```

To resolve the above error, provide the necessary permission to “hbase” user in “all - path” policy in the cm_hdfs service in Ranger. Also ensure, “hbase” user has permission in the “hbase-archive” policy as well.

Back Up Sqoop 2

If you are not using the default embedded Derby database for Sqoop 2, back up the database you have configured for Sqoop 2. Otherwise, back up the repository subdirectory of the Sqoop 2 metastore directory. This location is specified with the Sqoop 2 Server Metastore Directory property. The default location is: /var/lib/sqoop2. For this default location, Derby database files are located in /var/lib/sqoop2/repository.



Note: Sqoop 2 is not supported in CDP Private Cloud Base.

Back Up Hue

Back up the app registry file on all hosts running the Hue Server role if you have installed CDP Private Cloud Base using RPM packages.

The app registry file (app.reg) is present in the /usr/lib/hue directory if you have installed Hue using the RPM package. It is a JSON file which contains the details of all apps that are used within Hue. If you have installed Hue using the parcels, then the app.reg file may not be present on your system, and you do not need to back it up.

Run the following command to back up the app.reg file for installations using RPM packages:

```
cp -rp /usr/lib/hue/app.reg /usr/lib/hue_backup/app.reg-CM-CDH
```

Back Up Impala

Back up the Impala profile logs and logs of catalogd, statestore, and coordinators that can help investigate performance regressions. Perform the following tasks to back up the Impala cluster:

- Back up query profile logs of Impala. The logs are present in the `/profiles` directory under the coordinator log directory and the filename is similar to `impala_profile_log_1.1-1715070416212`.

To investigate performance regressions that are introduced after an upgrade, you will require the query profiles from the previous cluster (older version). You can use tools like `impala-profile-tool` to decode the query profiles into text profiles and then search for the runs before the upgrade for a comparison.

- Back up logs of `catalogd`, `statestore`, and coordinators that can also be helpful in troubleshooting. By default, the Impala logs are stored at `/var/log/catalogd/`, `/var/log/statestore/`, and `/var/log/impalad/`.

The log folders can be configured in Cloudera Manager by navigating to **Clusters IMPALA** and searching for the "Catalog Server Log Directory", "StateStore Log Directory", and "Impala Daemon Log Directory" properties.

Prerequisites for external database

This topic describes the external database prerequisites for Cloudera Base on premises. Before you upgrade your cluster, you must review the prerequisites for the external database and then perform the cluster upgrade.

Oozie database prerequisites

This topic describes the prerequisites for configuring an external database for Oozie in Cloudera Base on premises.

To configure an external database for Oozie in Cloudera Base on premises, see [Oozie database configurations](#).

Ranger database prerequisites

This topic describes the prerequisites for configuring an external database for Ranger in Cloudera Base on premises.

Ranger recommends that you must take backup of your backend database before upgrade.

Step 4: Back Up Cloudera Manager

After upgrading Cloudera Manager and before upgrading a cluster, you should backup Cloudera Manager again.

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)

Collect Information for Backing Up Cloudera Manager

Information you should collect before backing up Cloudera Manager.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Collect database information by running the following command:

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

For example:

```
...
com.cloudera.cmf.db.type=...
com.cloudera.cmf.db.host=DATABASE_HOSTNAME:DATABASE_PORT
com.cloudera.cmf.db.name=SCM
com.cloudera.cmf.db.user=SCM
com.cloudera.cmf.db.password=SOME_PASSWORD
```

3. Collect information (host name, port number, database name, user name and password) for the following databases.

- Reports Manager

You can find the database information by using the Cloudera Manager Admin Console. Go to **Clusters Cloudera Management Service Configuration** and select the Database category. You may need to contact your database administrator to obtain the passwords.

4. Find the host where the Service Monitor, Host Monitor and Event Server roles are running. Go to **Clusters Cloudera Management Service Instances** and note which hosts are running these roles.

Back Up Cloudera Manager Agent



Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups, defined by the environment variable `CM_BACKUP_DIR`, which is used in all the backup commands. You may change these destination paths in the command as needed for your deployment.

The tar commands in the steps below may return the following message. It is safe to ignore this message:

```
tar: Removing leading `/' from member names
```

Backup up the following Cloudera Manager agent files on all hosts:

- Create a top level backup directory.

```
export CM_BACKUP_DIR=`date +%F`-CM"
echo $CM_BACKUP_DIR
mkdir -p $CM_BACKUP_DIR
```

- Back up the Cloudera Manager Agent directory and the runtime state.

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-agent.tar --exclude=*.sock /etc/cloudera-scm-agent /etc/default/cloudera-scm-agent /var/run/cloudera-scm-agent /var/lib/cloudera-scm-agent
```

- Back up the existing repository directory.

RHEL / CentOS

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/yum.repos.d
```

SLES

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/zypp/repos.d
```

Ubuntu

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/apt/sources.list.d
```

Back Up the Cloudera Management Service



Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups. You may change these destination paths in the command as needed for your deployment.

1. Stop the Cloudera Management Service.
 - a. Log in to the Cloudera Manager Admin Console.
 - b. Go to Cloudera Management Service .
 - c. Select Actions Stop .
2. On the host where the Service Monitor role is configured to run, backup the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-SERVICE-MONITOR /var/lib/cloudera-service-m
onitor-`date +%F`-CM
```

3. On the host where the Host Monitor role is configured to run, backup the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-HOST-MONITOR /var/lib/cloudera-host-monitor-
`date +%F`-CM
```

4. On the host where the Event Server role is configured to run, back up the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-SCM-EVENTSERVER /var/lib/cloudera-scm-event
server-`date +%F`-CM
```

5. Restart the Cloudera Management Service if you are not upgrading Cloudera Manager immediately.
 - a. Log in to the Cloudera Manager Admin Console.
 - b. Select Clusters Cloudera Management Service .
 - c. Select Actions Start .

Stop Cloudera Manager Server & Cloudera Management Service

1. Stop the Cloudera Management Service.
 - a. Log in to the Cloudera Manager Admin Console.
 - b. Select Clusters Cloudera Management Service .
 - c. Select Actions Stop .
2. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

3. Stop the Cloudera Manager Server.

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

Back Up the Cloudera Manager Databases

During the upgrade, Cloudera Manager modifies the schema of the Cloudera Manager database. In case of failures during the upgrade, it may be necessary to rollback to the previous version of Cloudera Manager while addressing the upgrade failures.

When performing a rollback to a previous version, the Cloudera Manager Database must be restored to the previous database schema. For this reason, you must do the Cloudera Manager database backup, so that it can be restored if a rollback is necessary.



Tip:

You can get the name, user, and password properties for the Cloudera Manager database from the `/etc/cloudera-scm-server/db.properties` file:

```
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=NnYfWIj1bk
```


1. Back up the Cloudera Manager server database as per the instructions provided in your respective database documentation on how to backup and restore the databases.
2. Back up All other Cloudera Manager databases - Use the database information that [you collected in a previous step](#). You may need to contact your database administrator to obtain the passwords.

These databases can include the following:

- 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 database. If this information is not available, your cluster cannot start or function correctly. You must schedule and maintain regular backups of the database to recover the cluster in the event of the loss of this database.

- Oozie Server - Contains Oozie workflow, coordinator, and bundle data. Can grow very large. (Only available when installing CDH 5 or CDH 6 clusters.)
- Sqoop Server - Contains entities such as the connector, driver, links and jobs. Relatively small. (Only available when installing CDH 5 or CDH 6 clusters.)
- Reports Manager - Tracks disk utilization and processing activities over time. Medium-sized.
- Hive Metastore Server - Contains Hive metadata. Relatively small.
- Hue Server - Contains user account information, job submissions, and Hive queries. Relatively small.
- Sentry Server - Contains authorization metadata. Relatively small.
- Navigator Audit Server - Contains auditing information. In large clusters, this database can grow large. (Only available when installing CDH 5 or CDH 6 clusters.)
- Navigator Metadata Server - Contains authorization, policies, and audit report metadata. Relatively small. (Only available when installing CDH 5 or CDH 6 clusters.)
- DAS PostgreSQL server - Contains Hive and Tez event logs and DAG information. Can grow very large.
- Ranger Admin - Contains administrative information such as Ranger users, groups, and access policies. Medium-sized.
- Streaming Components:
 - Schema Registry - Contains the schemas and their metadata, all the versions and branches. You can use either MySQL, Postgres, or Oracle.



Important: For the Schema Registry database, you must set collation to be case sensitive.

- Streams Messaging Manager Server - Contains Kafka metadata, stores metrics, and alert definitions. Relatively small.

For more information about the number of databases that should be backed up, and restored if necessary, see [Required Databases](#).

Back Up Cloudera Manager Server



Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups, defined by the environment variable CM_B ACKUP_DIR, which is used in all the backup commands. You may change these destination paths in the command as needed for your deployment.

The tar commands in the steps below may return the following message. It is safe to ignore this message:

```
tar: Removing leading `/' from member names
```

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Create a top-level backup directory.

```
export CM_BACKUP_DIR=`date +%F`-CM
echo $CM_BACKUP_DIR
mkdir -p $CM_BACKUP_DIR
```

3. a. Back up the Cloudera Manager Server directories along with the CSP key file when Credential Storage Provider (CSP) is enabled:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-scm-server /etc/default/cloudera-scm-server ***/path/to/csp/keys***
```



Important: In the above command, you must replace the `***path/to/csp/key***` with the actual path where the CSP key has been stored from the following options:

- /var/lib/cloudera-scm-server/csp-data
- /opt/cloudera/
- /etc/cloudera/
- /var/cloudera/

- b. Back up the Cloudera Manager Server directories without CSP key file:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-scm-server /etc/default/cloudera-scm-server
```

4. Back up the existing repository directory.

RHEL / CentOS

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/yum.repos.d
```

SLES

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/zypp/repos.d
```

Ubuntu

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/apt/sources.list.d
```

(Optional) Start Cloudera Manager Server & Cloudera Management Service

Start the Cloudera Manager server and Cloudera Management Service.

If you will be immediately upgrading Cloudera Manager, skip this step and continue with [Step 3: Upgrading the Cloudera Manager Server](#) on page 53.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Start the Cloudera Manager Server.

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

3. Start the Cloudera Management Service.
 - a. Log in to the Cloudera Manager Admin Console.
 - b. Select Clusters Cloudera Management Service .
 - c. Select Actions Start .

Step 5: Access Parcels

Steps to access the Parcels required to install Cloudera Runtime.

Parcels contain the software used in your CDP Private Cloud Base clusters. If Cloudera Manager has access to the public Internet, Cloudera Manager automatically provides access to the latest version of the Cloudera Runtime 7.x.x Parcels directly from the Cloudera download site.

If Cloudera Manager does not have access to the internet, you must download the Parcels and set up a local Parcel repository. See [Configuring a Local Parcel Repository](#) on page 166. Enter the URL of your repository using the steps below.

If you want to upgrade to a different version of Cloudera Runtime 7.x.x, select the cluster version at the top of this page, and perform the following steps to add the Parcel URL:



Note: For a full list of Parcel URLs for older versions, see [Cloudera Runtime Download Information](#).

To add a new Parcel URL:

1. Log in to the Cloudera Manager Admin Console.
2. Click Parcels from the left menu.
3. Click Parcel Repositories & Network Settings.
4. In the Remote Parcel Repository URLs section, click the "+" icon and add the URL for your Parcel repository.
5. Click Save & Verify Configuration. A message with the status of the verification appears above the Remote Parcel Repository URLs section. If the URL is not valid, check the URL and enter the correct URL.
6. After the URL is verified, click Close.
7. Locate the row in the table that contains the new Cloudera Runtime parcel and click the Download button. If the parcel does not appear on the Parcels page, ensure that the Parcel URL you entered is correct.
8. After the parcel is downloaded, click the Distribute button.

Wait for the parcel to be distributed. Cloudera Manager displays the status of the Cloudera Runtime parcel distribution.

9. Click the Cloudera Manager logo to return to the home page.

Step 6: Enter Maintenance Mode

You can enable Maintenance Mode to avoid unnecessary alerts during the upgrade.

To avoid unnecessary alerts during the upgrade process, enter maintenance mode on your cluster before you start the upgrade. Entering maintenance mode stops email alerts and SNMP traps from being sent, but does not stop checks and configuration validations. Be sure to exit maintenance mode when you have finished the upgrade to re-enable Cloudera Manager alerts. [More Information](#).

On the Home > Status tab, click the actions menu next to the cluster name and select Enter Maintenance Mode.

Step 7: Run the Upgrade Cluster Wizard

The Upgrade Wizard manages the upgrade of your Cloudera Runtime software. The Upgrade Wizard is not used for upgrades to Service Packs or Hotfixes.



Important: You have selected an upgrade to Cloudera Runtime Service Pack 1 (7.1.7.1000). The upgrade process for this does not use the Upgrade Wizard if and only if the current Cluster is its own GA release. Skip the steps on this page and continue with the steps in following document: [Upgrading to a Service Pack](#).



Note: Not all combinations of Cloudera Manager and Cloudera Runtime are supported. Ensure that the version of Cloudera Manager you are using supports the version of Cloudera Runtime you have selected. See [Cloudera Manager support for Cloudera Runtime, CDH, and Cloudera Data Services on premises](#)

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator)

1. Log in to the Cloudera Manager Admin Console.
2. Ensure that you have completed the steps to add the Parcel URL in Cloudera Manager. See [Step 5: Access Parcels](#).
3. Ensure that all services in the cluster that are being upgraded are running and in good health.
4. Click the Actions menu and select Upgrade Cluster.

The Getting Started screen of the Upgrade Wizard displays.

5. Click the Upgrade to Version: drop-down and select the version of Cloudera Runtime for your upgrade.

The wizard now runs several checks to make sure that your cluster is ready for upgrade. You must resolve any reported issues before continuing.

6. The Install Services section displays any additional services that you need to install to upgrade your cluster.

If you are upgrading a cluster that has the Hive service, you will be prompted to add the Tez, Zookeeper, Hive on Tez, and YARN QueueManager services.



Warning: When you add Hive-on-Tez service, the Assign Roles page displays. You must ensure that the number of HiveServer2 roles present in the Hive service before the upgrade are included when the Assign Roles page displays. (You can verify this by opening the Cloudera Manager Admin Console Home page in a new browser tab, and going to the Instances tab in the Hive service.) If the number of HiveServer2 roles is not the same, the cluster upgrade will fail and the cluster will be unusable. If your upgrade fails, please contact Cloudera Support.

You must also select the same hosts for the HiveServer2 roles that were used before the upgrade. If you choose other hosts you must regenerate the keytabs for those hosts. See [Managing Kerberos credentials using Cloudera Manager](#).

7. The Inspector Checks section displays several inspectors you must run before continuing. If these inspectors report errors, you must resolve those before continuing.
 - Click the Show Inspector Results button to see details of the inspection.
 - Click the Run Again button to verify that you have resolved the issue.
 - If you are confident that the errors are not critical, select Skip this step. I understand the risks..

The Inspector Checks section includes the following inspectors:

- Host Inspector
- Service Inspector



Note: If the Hive service is present in the cluster, the Inspector Checks include a Validate Hive Metastore schema step. This check may return a "green" result but the validation may actually contain failures tagged with the WARN label. You should look through the inspector results for Hive and correct any failures before continuing with the upgrade.

Run these inspectors and correct any reported errors before continuing.

8. The Database Backup section asks you to verify that you have completed the necessary backups. Select Yes, I have performed these steps.
9. Click Continue. (The Continue button remains greyed out until all upgrades steps are complete and all warnings have been acknowledged.)

10. On the Choose Upgrade Procedure page, select the Rolling Restart option. The Rolling Restart option is available only when your cluster has HDFS HA enabled. Click Continue.



Note: Selecting the Full Cluster Restart option allows you to upgrade your cluster through the [regular upgrade process](#) which will have a longer cluster downtime. Also, this option is available if your cluster is not enabled with HDFS HA.

You must provide the values for the following fields. The default values are already entered.

- a. Batch size
- b. Sleep between batches
- c. Failed threshold



Note: In case you encounter issues with upgrading your cluster, you must review the [ZDU known issues](#) section. If the issue is still not resolved, you must reach out to Cloudera Support.

11. Click Continue again to shut down the cluster and begin the upgrade.

The Upgrade Cluster Command screen opens and displays the progress of the upgrade.

12. When the Upgrade steps are complete, click Continue.

The Summary page opens and displays any additional steps you need to complete the upgrade.

13. Click Continue.

Step 8: Finalize the HDFS or Ozone Upgrade

Steps to finalize the HDFS or Ozone upgrade. This step is required only when Apache Hadoop version changes.



Note: Before upgrading the dependent services such as HBase, you must verify and ensure that the HDFS safemode is off.

To determine if you can finalize the upgrade, run important workloads and ensure that they are successful. After you have finalized the upgrade, you cannot roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time.

Make sure you have enough free disk space, keeping in mind that the following behavior continues until the upgrade is finalized:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

If you have Enabled high availability for HDFS, and you have performed a rolling upgrade:

1. Go to the HDFS service.
2. Select ActionsFinalize Rolling Upgrade and click Finalize Rolling Upgrade to confirm.

If you have not performed a rolling upgrade:

1. Go to the HDFS service.
2. Click the Instances tab.
3. Click the link for the NameNode instance. If you have enabled high availability for HDFS, click the link labeled NameNode (Active).

The NameNode instance page displays.

4. Select Actions Finalize Metadata Upgrade and click Finalize Metadata Upgrade to confirm.

Finalize Ozone upgrade

To complete the upgrade of Ozone services, you must finalize the upgrade. For more information, see [Upgrading Ozone parcels](#).

Step 9: Complete Post-Upgrade steps for upgrades to Cloudera Base on premises

Steps to perform after upgrading a cluster.



Tip: If service specific instructions are not displayed on this page, then there are no post-upgrade steps required for your upgrade.

Loading Filters ... 7.13.1 7.11.3 7.7.3 7.7.1 7.6.7 7.6.1 7.4.4 7.3.1 7.1.9.1000 7.1.9 7.1.8 7.1.7.3000 7.1.7.2000 7.1.7.1000 7.1.7 7.1.6 7.3.1 7.1.9.1000 7.1.9 7.1.8 7.1.7.3000 7.1.7.2000 7.1.7.1000 7.1.7

Some components might require additional steps after you complete the upgrade to Cloudera Base on premises:



Important: During the course of the upgrade to Cloudera Base on premises if you experience any issue midway through the process, the Install YARN MapReduce Framework JARs, Tez MR Framework JARs, and Oozie Sharelib processes may not complete successfully. This action ensures that the latest MR Framework JARs exist in HDFS. For example: `/user/yarn/mapreduce/mr-framework/`, and that they are used instead of any old ones. This situation can create an issue when upgrading to a version containing the fix for log4j vulnerabilities, such as 7.1.7 Service Pack releases, because the old vulnerable log4j files are contained within the MR Framework JARs tar file.

As a post-upgrade step, you must check that location in HDFS to ensure that it is the latest version, and run the following if it does not:

```
hdfs dfs -ls /user/yarn/mapreduce/mr-framework/
```

If only the old version's tar file exists there, such as:

```
/user/yarn/mapreduce/mr-framework/3.1.1.7.1.7.78-12-mr-framework.tar.gz
```

Then perform the following: Cloudera Manager > Yarn > Actions > Install YARN MapReduce Framework JARs. You can clear out the old tar file after this is done for keeping it clean.

Then perform the following to install the YARN service dependency jars: Cloudera Manager > Actions > Install Yarn Service Dependencies. These steps will create YARN Services Framework Root Directory and uploads the Service Dependencies.

Tez MR Framework JARs and Oozie Sharelib entities must be examined in the similar manner. Check that location in HDFS to ensure that it is the latest version, and run the following if it does not:

1. Cloudera Manager > Yarn > Actions > Install YARN MapReduce Framework JARs
2. Cloudera Manager > Tez > Actions > Upload Tez tar file to HDFS
3. Cloudera Manager > Oozie > Actions > Install Oozie ShareLib

Clear any old files out of these directories:

1.

```
hdfs dfs -ls -R /user/yarn/mapreduce/mr-framework/
```
2.

```
hdfs dfs -ls -R /user/tez/
```
3.

```
hdfs dfs -ls /user/oozie/share/lib/
```

Kafka – clear protocol and log format versions



Important: Neither rollbacks nor downgrades are possible after you complete the following steps.

1. Remove the following properties from the Kafka Broker Advanced Configuration Snippet (Safety Valve) for kafka.properties configuration property.
 - inter.broker.protocol.version
 - log.message.format.version
2. Save your changes.
3. Perform a rolling restart:
 - a. Select the Kafka service.
 - b. Click Actions Rolling Restart .
 - c. In the pop-up dialog box, select the options you want and click Rolling Restart.
 - d. Click Close once the command has finished.

Kafka – run the Create Kafka Log Directory Free Capacity Check command

In Cloudera Runtime 7.1.9, a new Cloudera Manager trigger called Broker Log Directory Free Capacity Check is introduced. This trigger fires if the capacity of any log directory falls below 10%. The trigger is automatically created on new clusters, but must be created on upgraded clusters by running the Create Kafka Log Directory Free Capacity Check command. Complete the following to create the trigger:

1. In Cloudera Manager, select the Kafka service.
2. Click Actions Create Kafka Log Directory Free Capacity Check .

Once created, the trigger is available on the Kafka service **Status** page. Note that the **Status** page also contains a new chart which is introduced in 7.1.9. The chart is called **Log Directory Free Capacity**. It shows the capacity of each Kafka Broker log directory.

Kafka – enable SPNEGO

If Kerberos authentication is enabled for Kafka and you use Ranger for authorization, you need to enable SPNEGO. Enabling SPNEGO is required for the setup of the Ranger Kafka Connect plugin. If SPNEGO is not enabled, you will encounter authorization issues related to Kafka Connect.

1. In Cloudera Manager, go to Kafka Configuration .
2. Select the Enable SPNEGO Authentication For Kafka Connect property.

Kafka producer – disable idempotence or update Ranger policies

Kafka producer applications upgraded to Kafka 3.0.1, 3.1.1, or all versions after 3.1.1 have idempotence enabled by default. Idempotent producers must have Idempotent Write permission set on the cluster resource in Ranger. As a result, If you upgraded your producers and use Ranger to provide authorization for the Kafka service, you must do either of the following after upgrading your producers. If configuration is not done, the producers fail to initialize due to an authorization failure.

- Explicitly disable idempotence for your producers. This can be done by setting `enable.idempotence` to `false` for your producer applications. For more information, see [Producer Configs](#) in the Apache Kafka documentation.
- Update your policies in Ranger and ensure that producers have Idempotent Write permission on the cluster resource. For more information, see [Configuring resource-based policies](#).

Kafka Connect – resolve `ClassNotFoundException`

Upgrading a Kafka Connect cluster can fail with the following `ClassNotFoundException`:

```
Error: ClassNotFoundException exception occurred: com.cloudera.dim.kafka.config.provider.CmAgentConfigProvider
exception occurred: com.cloudera.dim.kafka.config.provider.CmAgentConfigProvider
```

This is due to the Cloudera Manager agent not being able to properly create the symlinks and alternatives when performing an upgrade. The issue also occurs when a custom path is set for `plugin.path` in Cloudera Manager, which does not contain the expected plugin JARs. Complete the following steps to resolve the issue:

1. Manually install missing symlinks or alternatives relevant to kafka-connect by checking `cloudera-scm-agent.log`.
2. Ensure that the kafka-connect libraries are present in `/var/lib/kafka` or change the `plugin.path` to a value where these libraries are present.

If required, manually copy the JARs to the location configured in `plugin.path`.

3. If the issue persists, set the `plugin.path` to:

```
/opt/cloudera/parcels/CDH-[***VERSION**]/lib/kafka_connect_ext/libs
```

This is the location where kafka-connect libraries are set when installing parcels.

Kafka Connect – configure Kerberos ATL rules

Kafka Connect now uses the cluster-wide Kerberos auth-to-local (ATL) rules by default. A new configuration property called Kafka Connect SPNEGO Auth To Local Rules is introduced. This property is used to manually specify the ATL rules. During an upgrade, the property is set to `DEFAULT` to ensure backward compatibility. If you want to use the cluster-wide Kerberos ATL rules for custom Kerberos principals, you need to clear the existing value from the Kafka Connect SPNEGO Auth To Local Rules property in Cloudera Manager Kafka Configuration .

Schema Registry - create the Ranger Schema Registry Plugin Audit Directory

The Ranger Schema Registry Plugin Audit Directory is not created automatically when upgrading a cluster. This causes the Schema Registry service to encounter an issue. As a result, following an upgrade, you must manually initiate the command in Cloudera Manager that creates the audit directory.

1. In Cloudera Manager, select the Schema Registry service.
2. Click Actions Create Ranger Schema Registry Plugin Audit Directory .

Schema Registry – configure Kerberos Name Rules

The default value for Schema Registry Kerberos Name Rules is changed. Schema Registry now automatically applies the cluster-wide auth-to-local (ATL) rules by default. However, the previous value of the Schema Registry Kerberos Name Rules property is preserved during an upgrade. You must manually clear the property following an upgrade

to transition to the new default value in **Cloudera Manager Schema Registry Configuration** when using Custom Kerberos Principal.

Schema Registry – manually update Atlas policies in Ranger to include the schemaregistry user

Following an upgrade, Schema Registry might not have the required permissions in Ranger to access Atlas. As a result, you must update the policies manually. Otherwise, Schema Registry's integration with Atlas might not function in secure clusters where Ranger authorization is enabled.

1. Access the Ranger Console (Ranger Admin web UI).
2. Click the `cm_atlas` resource-based service.
3. Add the schemaregistry user to the all - * policies.
4. Click **Manage Service Edit Service**.
5. Add the schemaregistry user to the `default.policy.users` property.

Schema Registry – configure Schema Registry to use V2 of its fingerprinting mechanism

Following an upgrade, Cloudera recommends that you configure Schema Registry to use fingerprinting V2. Fingerprinting V2 resolves an issue in V1 where schemas were not created under certain circumstances. For more information on the original issues as well as Schema Registry fingerprinting, see [TSB-713](#). Note that even if you switch to V2, some issues might still persist, see [TSB-718](#) for more information.

1. In Cloudera Manager, select the Schema Registry service.
2. Go to Configuration.
3. Set the Fingerprint Version property to `VERSION_2`.
4. Select **Actions Regenerate Fingerprints**.
5. Click **Regenerate Fingerprints** to start the action.
6. Restart Schema Registry.

Streams Messaging Manager – update the replication factor of internal topics

In Cloudera Runtime 7.1.9, the default replication factor of the `__smm*` internal SMM topics is increased to 3. The replication factor of these topics is not updated when you upgrade a cluster. As a result, Cloudera recommends that you increase the replication factor of these topics to 3 with the `kafka-reassign-partitions` tool following the upgrade. For more information, see [kafka-reassign-partitions](#).

Streams Messaging Manager – reconnect SMM with SRM

Following a successful upgrade, if you want to use SMM to monitor SRM replications, you must reconnect the two services. This is done by enabling the `STREAMS_REPLICATION_MANAGER` Service SMM property which is disabled by default.



Important: SMM can only connect to and monitor an SRM service that is running in the same cluster as SMM. Monitoring an SRM service that is running in a cluster that is external to SMM is no longer supported.

1. In Cloudera Manager, select the SMM service.
2. Go to Configuration.
3. Find and enable the `STREAMS_REPLICATION_MANAGER` Service property.
4. Click **Save Changes**.
5. Restart the service.

Streams Messaging Manager – configure Kerberos Name Rules

The default value for Kerberos Name Rules is changed. SMM now automatically applies the cluster-wide auth-to-local (ATL) rules by default. However, the previous value of the Kerberos Name Rules property is preserved during an upgrade. You must manually clear the property following an upgrade to transition to the new default value in **Cloudera Manager SMM Configuration** when using Custom Kerberos Principal.

Streams Messaging Manager – update Prometheus configuration

The default port that the Kafka Connect metrics reporter uses as well as the property that configures the port are changed. If you are using Prometheus as the metrics store for SMM, you must update your Prometheus configuration to include the new port.

Complete the following:

1. In Cloudera Manager, select the Kafka service.
2. Go to Configuration.
3. Take note of the values of Secure Jetty Metrics Port and Jetty Metrics Port properties.

Which property value you take note of depends on whether Kafka Connect is secured using TLS/SSL. Kafka Connect is secured with TLS/SSL if Enable TLS/SSL for Kafka Connect is selected. Take note of Jetty Metrics Port if Kafka Connect is not secured. Take note of Secure Jetty Metrics Port if Kafka Connect is secured.

4. If Enable Basic Authentication for Metrics Reporter is selected take note of the following properties:
 - Jetty Metrics User Name
 - Jetty Metrics Password
5. Update the kafka_connect job section in your prometheus.yml configuration file with the new port number. If Basic Authentication is enabled, add the username and password as well. For example:



Tip: This example uses 28087 as the port, which is the default value for Secure Jetty Metrics Port.

```
- job_name: 'kafka_connect'
  metrics_path: '/api/prometheus-metrics'

  basic_auth:
    username: 'email@username.me'
    password: 'password'

  static_configs:
    - targets: ['my-cluster-1.com:28087', 'my-cluster-2.com:28087', 'my-cluster-3.com:28087']
```

6. Reload your Prometheus configuration.

For more information, see [Configuration](#) in the Prometheus documentation.

Streams Replication Manager – disable legacy protocol and data format

During the upgrade SRM uses the legacy versions of its intra cluster protocol and internal changelog data format to ensure backward compatibility. After a successful upgrade, you must configure SRM to use the new protocol and data format. Otherwise, some features will not function.

1. In Cloudera Manager select the Streams Replication Manager service.
2. Go to configuration and clear the following properties:
 - Use Legacy Intra Cluster Host Name Format
 - Use Legacy Internal Changelog Data Format
3. Restart the SRM service.

Streams Replication Manager – reconfigure the Log Format property

The value set in the Log Format property is cleared during the upgrade. If you customized the log format using this property, its value must be manually reconfigured following the upgrade.

1. In Cloudera Manager select the SRM service.
2. Go to Configuration and configure the Log Format property.
3. Restart the SRM service.

Streams Replication Manager – export consumer group offsets

In Cloudera Runtime 7.1.8 and higher, the separator configured with `replication.policy.separator` in Streams Replication Manager applies to the names of the offsets-syncs and checkpoints internal topics if the default replication (`DefaultReplicationPolicy`) policy is in use.

If you have been using a custom separator, when SRM is restarted after the upgrade, a new set of internal topics are automatically created using the separator that is configured. These topics, however, will be empty and are only be repopulated with data after replication restarts. If you want to export translated consumer group offsets after the upgrade but before replication starts, you must export the translated consumer group offsets from the old topics as the data is not yet available in the new topics. This is done by specifically instructing the `srm-control` tool to use the default separator, which is the dot (`.`). For example:

```
srm-control offsets --config [***CONFIG FILE***] --source [SOURCE_CLUSTER] -
-target [TARGET_CLUSTER] --group [GROUP1] --export > out.csv
```

Where the `[***CONFIG FILE***]` specified in `--config` contains the following configuration entry

```
replication.policy.separator=.
```

Streams Replication Manager – delete unused internal topics

In Cloudera Runtime 7.1.9 and higher, SRM uses a new version of its internal Kafka Streams application. The topics and data of the old application (the one used before the upgrade) are not deleted automatically during the upgrade. Run the following commands to reset the state of the now unused Kafka Streams applications and to delete unused internal topics. Run the following commands on the hosts of the SRM Service role.

```
kafka-streams-application-reset \
  --bootstrap-server [***SRM SERVICE HOST***] \
  --config-file [***PROPERTIES FILE***] \
  --application-id srm-service
```

```
kafka-streams-application-reset \
  --bootstrap-server [***SRM SERVICE HOST***] \
  --config-file [***PROPERTIES FILE***] \
  --application-id srm-service-status
```

Replace `[***PROPERTIES FILE***]` with the location of a configuration file that contains all necessary security properties that are required to establish a connection with the Kafka service. This option is only required if your Kafka service is secured.

Streams Replication Manager – enable internal topics based remote topic discovery

The Remote Topics Discovery With Internal Topic property is not selected during the upgrade by default. Ensure that you select the property following an upgrade and restart the SRM service. This property enables SRM to track remote (replicated) topics using an internal Kafka topic instead of relying on the naming convention of the replication policy that is in use. This enables the SRM service to provide better monitoring insights on replications.

Streams Replication Manager – transition IdentityReplicationPolicy configuration

Cloudera Runtime 7.1.9 introduces a dedicated configuration property in Cloudera Manager to enable prefixless replication with the `IdentityReplicationPolicy`. If you have been using the `IdentityReplicationPolicy` in a previous version, Cloudera recommends that you transition your configuration to use the new configuration property. If you transition your configuration, you will be able to monitor your replications with the SRM Service, which was not possible in previous versions if the `IdentityReplicationPolicy` was in use.

1. In Cloudera Manager, select the SRM service.

2. Go to Configuration.
3. Clear the following entry from Streams Replication Manager's Replication Config.

```
replication.policy.class=org.apache.kafka.connect.mirror.IdentityReplicationPolicy
```

4. Select Enable Prefixless Replication.
5. Ensure that Remote Topics Discovery With Internal Topic is selected.

If this property is not selected when the `IdentityReplicationPolicy` is in use, replication monitoring provided by the SRM Service does not function. This property is enabled by default when Enable Prefixless Replication is selected.

6. Restart the SRM service.

Cruise Control – reconfigure goals to customized values

Provide the customized values for the Cruise Control goal sets, if you needed to copy them before upgrading your cluster. Furthermore, you must rename any mentioning of `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareGoal` to `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareDistributionGoal` as Cruise Control will not be able to start otherwise. For more information, see the [Cruise Control Known Issues](#).

Cruise Control – configure Kerberos ATL rules

Cruise Control now uses the cluster-wide Kerberos auth-to-local (ATL) rules by default. A new configuration property called SPNEGO Principal to Local Rules is introduced. This property is used to manually specify the ATL rules. During an upgrade, the property is set to DEFAULT to ensure backward compatibility. If you want to use the cluster-wide Kerberos ATL rules for custom Kerberos principals, you need to clear the existing value from the SPNEGO Principal to Local Rules property in Cloudera Manager Cruise Control Configuration .

Migrating a YARN Queue Manager database to an external database

To continue using the embedded database, upgrade to CDP Private Cloud Base 7.1.9 CHF2 and disregard these instructions. However, if you prefer to use PostgreSQL as the external database, use any CDP Private Cloud Base 7.1.9 and follow these post-installation steps to migrate your database.

You must supply the details of a PostgreSQL database to migrate your old data to CDP Private Cloud Base 7.1.9, as Queue Manager only uses PostgreSQL in CDP Private Cloud Base 7.1.9 or higher versions.



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 the Cloudera Manager and Queue Manager's UIs to setup the database configuration, and to load the UI to allow the data migration to complete properly. You must also refrain from direct API calls during an upgrade to 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). The roles users need to assign are the following: SUPERUSER CREATEDB CREATEROLE INHERIT LOGIN. You can use the sample query or a similar query below:

```
CREATE ROLE qmadmin PASSWORD 'password' SUPERUSER CREATEDB CREATEROLE INHERIT LOGIN;
CREATE DATABASE "configstore";
ALTER DATABASE "configstore" OWNER TO qmadmin;
```

1. In your Cloudera Manager instance, navigate to Clusters Yarn Queue Manager .
2. Click 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 that is to be used. You will have these values from the query that was run in the prerequisite:

- 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

QUEUEMANAGER-1 Actions

Status Instances Configuration Commands Charts Library Audits Quick Links

Search Filters (1) Role Groups History & Rollback

Filters (1) Clear All

SCOPE

- QUEUEMANAGER-1 (Service-... 6
- YARN Queue Manager Store 0
- YARN Queue Manager Webapp 0

CATEGORY

- Main 28
- Advanced 22
- Database 6
- Logs 8
- Monitoring 29
- Performance 1
- Ports and Addresses 3
- Resource Management 5
- Security 11
- Stacks Collection 5

STATUS

- Error 0
- Warning 0
- Edited 0
- Non-Default 4
- Include Overrides 0

QueueManager Config Service Database Type QUEUEMANAGER-1 (Service-Wide)

queuemanager_database_type ☒ PostgreSQL

QueueManager Config Service Database Name QUEUEMANAGER-1 (Service-Wide)

queuemanager_database_name queuemanager1

QueueManager Config Service Database Host QUEUEMANAGER-1 (Service-Wide)

queuemanager_database_host qmmigdebug-1.qmmigdebug.root.hwx.site

QueueManager Config Service Database Port QUEUEMANAGER-1 (Service-Wide)

queuemanager_database_port 5432

QueueManager Config Service Database User QUEUEMANAGER-1 (Service-Wide)

queuemanager_jpa_jdbc_user queuemanager1b

QueueManager Config Service Database User Password QUEUEMANAGER-1 (Service-Wide)

queuemanager_jpa_jdbc_password

1 - 6 of 6

Save Changes (CTRL+S)

5. Click Save Changes.

6. In Queue Manager, navigate to Actions and click Restart. The data migration to the PostgreSQLdata database may take a few minutes after Queue Manager is restarted.

Ozone - Steps to enable HDFS dependency on running Ozone

1. Log in to Cloudera Manager
2. Navigate to Clusters
3. Select the Ozone service
4. Navigate to the Configurations tab
5. Search for the HDFS Service configuration
6. Select the checkbox

Step result: This will enable the HDFS dependency on running Ozone. For more information, see the [HDFS dependency on running Ozone](#) documentation.

Considerations when upgrading Ranger RMS

Prior to CDP Private Cloud Base 7.1.7, Ranger RMS synchronizes only table metadata. Database metadata mappings were not downloaded from HMS. After migrating from CDP Private Cloud Base 7.1.7 or lower versions to CDP Private Cloud Base 7.1.8 or later versions, only pre-existing table mappings will be available. Any newly created tables or database mappings will be synchronized in RMS. Any pre-existing database mappings will not be present. To ensure the pre-existing databases are mapped correctly, you must perform a full-sync after completing the upgrade.

Oozie - Updating column types in Oozie Database

If the Oracle database is used for Oozie, then you must update the APP_PATH column type to store values with more than 255 characters. This ensures Oozie does not get stuck in PREP state when your application path exceeds the 255 character limit. If the APP_PATH column type is not updated, then Oozie fails to run the jobs with the following database error message Data too long for column 'app_path'. This scenario is also applicable to coordinator and bundle jobs. For database types other than Oracle, this update is not mandatory for using Oozie. It works without the update if the APP_PATH value does not exceed 255 characters. Also, Oozie's internal database schema validation fails with an unexpected APP_PATH column type. However, this validation does not have any effect. It just logs it's result.

You must update the APP_PATH column type in the WF_JOBS, BUNDLE_JOBS, and COORD_JOBS tables in the Oozie database for the following conditions:

- When you use the **Oracle** database for Oozie service.
- When you are upgrading Cloudera Runtime from earlier versions to 7.1.9 SP1 CHF1 version or later.

If you do not execute the following statements on the Oozie Oracle database, then the Oozie service fails to run the jobs with a database persistence error. This update is not mandatory, but highly recommended for other database types, such as MySQL, MariaDB, and PostgreSQL, due to the internal database schema validation.

Examples:

- On the Oozie Oracle database - The following example uses the Oracle sqlplus command-line tool:

```
sqlplus <OOZIE_DB_USERNAME>@localhost/<SERVICE_NAME>

SQL> ALTER TABLE <TABLE_NAME> ADD (APP_PATH_TMP CLOB);

Table altered.

SQL> UPDATE <TABLE_NAME> SET APP_PATH_TMP = APP_PATH;

X rows updated.

SQL> ALTER TABLE <TABLE_NAME> DROP COLUMN APP_PATH;

Table altered.

SQL> ALTER TABLE <TABLE_NAME> RENAME COLUMN APP_PATH_TMP TO APP_PATH;

Table altered.
```

- On the Oozie MySQL database - The following example uses the MySQL mysql command-line tool:

```
$ mysql -u root -p
Enter password:

mysql> use <OOZIE_DATABASE_NAME>;
Database changed

mysql> ALTER TABLE <TABLE_NAME> MODIFY COLUMN app_path text;
Query OK, X rows affected (0.03 sec)
Records: X Duplicates: 0 Warnings: 0
mysql> exit
Bye
```

- On the Oozie MariaDB database - The following example uses the MariaDB mysql command-line tool:

```
$ mysql -u root -p
Enter password:

MariaDB [(none)]> use <OOZIE_DATABASE_NAME>;
Database changed
```

```

MariaDB [OOZIE_DATABASE_NAME]> ALTER TABLE <TABLE_NAME> MODIFY COLUMN app_path text;
Query OK, X rows affected (2.11 sec)
Records: X Duplicates: 0 Warnings: 0

MariaDB [OOZIE_DATABASE_NAME]> exit
Bye

```

- On the Oozie PostgreSQL database - The following example uses the PostgreSQL psql command-line tool:

```

$ psql -U postgres
Password for user postgres: *****

postgres=# \c <OOZIE_DATABASE_NAME>;
You are now connected to database "<OOZIE_DATABASE_NAME>" as user "postgres".
OOZIE_DATABASE_NAME=# ALTER TABLE <TABLE_NAME> ALTER COLUMN app_path type
text;
ALTER TABLE

OOZIE_DATABASE_NAME=# \q

```

Sqoop

To know more, see [Unable to read Sqoop metastore created by an older HSQLDB version](#).


Navigator Encrypt

When upgrading from a release with a version less than or equal to 7.1.9 to a release with a version greater than 7.1.9, you must run a `sudo navencrypt acl update` command after Navigator Encrypt is upgraded and running, but before the applications using Navigator Encrypt are started. For information on how to update ACL fingerprints, see [Updating ACL Fingerprints](#).

Step 10: Exit Maintenance Mode

If you enabled Maintenance Mode before the upgrade, you must exit Maintenance Mode to complete the upgrade.

If you entered maintenance mode during this upgrade, exit maintenance mode.

On the HomeStatus tab, click  next to the cluster name and select Exit Maintenance Mode.

ZDU known issues

If the ZDU process fails, the upgrade is not aborted but paused. You must review the issues to fix the cause of failure and resume with the upgrade.

Cloudera Manager

When upgrading from CDP Private Cloud Base 7.1.7 SP2 to CDP Private Cloud Base 7.1.9, without agent TLS encryption, the distribute parcel step does not work with Python 3.8.

If you are using Cloudera Manager running with TLS on port 7183, you must switch the agent communication to use TLS by setting `use_tls=1` in `agent config.ini` and set `Use TLS Encryption for Agents` to `true` in Cloudera Manager.

OPSAPS-67953: Downgrading Cloudera Runtime 7.1.9 version to 7.1.8 CHF 10 fails with errors when a previous unfinished upgrade command is found.

Error Message: A previous unfinished upgrade command was found. To continue upgrading: perform a 'Retry' on the original command; to return to an earlier consistent state: restore a Cloudera Manager backup.

Perform the following steps to rollback to the previous version:

1. Delete records from the UPGRADE_STATE table.
2. Restart Cloudera Manager.
3. Follow the rollback / downgrade steps.

OPSAPS-68279: When upgrading CDP Private Cloud Base 7.1.7 SP2 to CDP Private Cloud Base 7.1.9, the command step DeployClientConfig may fail due to the following error:

Error Message: Client configuration generation requires the following additional parcels to be activated: [cdh]

Verify that the error is because of the 7.1.9 parcels activation failure.

1. Navigate to the parcels page.
2. See if the following error is displayed: Error when distributing to <hostname>: Script file/opt/cloudera/parcels/flood/CDH-7.1.9-1.cdh7.1.9.p0.43968053-el7.parcel/CDH-7.1.1.cdh7.1.9.0.43968053-el7.parcel does not exist.
3. Identify the host from the error message and ssh into the host by running the ssh <hostname> command.
4. Navigate to the agent directory by running the cd /var/log/cloudera-scm-agent command.
5. Find the following pattern in agent log file(s) Exception: Untar failed with return code: 2, with tar output: stdout: [b"], stderr: [b\"ngzip: stdin: invalid compressed data--format violated\n\tar: Unexpected EOF in archive\n\tar: Unexpected EOF in archive\n\tar: Error is not recoverable: exiting now\n'].
1. If the above exception appears, you must restart the agent on that host by running the command `systemctl restart cloudera-scm-agent`. After restarting the agent, the parcel distribution must be successful.
2. After the parcel distribution is successful, click resume to continue with the upgrade.

OPSAPS-67929: While upgrading from CDP Private Cloud Base 7.1.7 SP2 to CDP Private Cloud Base 7.1.9 version and if there is an upgrade failure in the middle of the process, the Resume option is not available.

You must reach out to Cloudera Support.

HBase

CDPD-58755: In a small cluster (for example, five RegionServer or DataNodes or lesser), HBase tests fail intermittently because there are not enough data nodes to failover.

Error Message: INFO:cm_server.py:2943:Final list of failed state checks: Health check: HBAS E_REGION_SERVERS_HEALTHY failed on HBASE-1 with status BAD, HBASE-1-REGIONSERVER-f3a7628ea499295abb44816cddf04854 (3c605548-280d-4e59-8425-e623ee070a54) has undesired health: BAD from failed check(s): REGION_SERVER_MASTER_CONNECTIVITY: BAD, REGION_SERVER_SCM_HEALTH: BAD

Add the following configurations using Cloudera Manager to alleviate this issue.

1. Go to Cloudera Manager HDFS Configuration HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml.
2. Add the following configurations with the suggested values.

```
dfs.client.block.write.replace-datanode-on-failure.policy=ALWAYS
dfs.client.block.write.replace-datanode-on-failure.best-effort=true
```



```
dfs.client.block.write.replace-datanode-on-failure.enable=true
```

CDPD-606480: During HBase rolling upgrade, HBase commands fail.

Error Message: ServerNotRunningYetException: Server is not running yet

HBase retries DDL operations submitted while the master is initializing until the master is fully initialized to serve the request. However, a situation might arise where the default number of retries or intervals proves to be insufficient for an operation submitted by the client to complete. Such a scenario might occur where the Master crashes due to an issue and the standby master does initialization after the takeover or the master is restarted for any reason, including upgrades taking a longer time.

The initialization time of the master can differ depending on the size of the cluster; however, implementing the following configuration adjustments in your client application can support the master getting initialized up to 10 minutes.

Add the following configuration to the client application.

```
<property>
  <name>hbase.client.pause</name>
  <value>300</value>
</property>
<property>
  <name>hbase.client.retries.number</name>
  <value>20</value>
</property>
```

However, if you have seen a longer or shorter master initialization period, you can modify these values accordingly. These retry settings apply to all types of calls to HBase service, encompassing GET, SCAN, MUTATE, and DDLs.

Ozone

OPSAPS-67340: Service Monitor is in a bad health state after restarting the Cloudera Manager server, reporting problems with descriptor and metric schema age, when Kerberos and Cloudera Manager SPNEGO authentication are enabled.

Error message: Health check: MGMT_SERVICE_MONITOR_HEALTH failed on mgmt with status BAD, mgmt-SERVICEMONITOR-02d43fad954612b08a0ea1a5df8f42aa (a4efb7ff-4a4d-4291-9dfe-88a394bf9c23) has undesired health: BAD from failed check(s): SERVICE_MONITOR_METRIC_SCHEMA_FETCH: BAD, SERVICE_MONITOR_SCM_DESCRIPTOR_FETCH: BAD

Restart Service Monitor each time the Cloudera Manager server is restarted:

1. Log in to Cloudera Manager.
2. Stop Service Monitor.
3. Restart Cloudera Manager server.
4. Start Service Monitor.



Note: If the health status is already bad, restart Service Monitor.

CDPQE-25023: Upgrading from CDP Private Cloud Base 7.1.8 or 7.1.7 SP2 to CDP Private Cloud Base 7.1.9, HBase post-upgrade validation failed for HBASE_REGION_SERVERS_HEALTHY.

Error message: INFO:cm_server.py:2943:Final list of failed state checks: Health check: HBASE_REGION_SERVERS_HEALTHY failed on HBASE-1 with status BAD, HBASE-1-REGIONSERVER-f9673e2d17e546ddec44e7724865d4f5 (043f7f4a-c46d-42af-8d82-061cba1c3f7c) has undesired health: BAD from failed check(s): REGION_SERVER_MASTER_CONNECTIVITY: BAD, REGION_SERVER_SCM_HEALTH: BAD, Health check: IMPALA_IMPALAD

S_HEALTHY failed on IMPALA-1 with status BAD, IMPALA-1-IMPALAD-f9673e2d17e546ddec44e7724865d4f5 (043f7f4a-c46d-42af-8d82-061cba1c3f7c) has undesired health: BAD from failed check(s): IMPALAD_QUERY_MONITORING_STATUS: BAD, Health check: SCHEMAREGISTRY_SCHEMA_REGISTRY_SERVER_HEALTHY failed on SCHEMAREGISTRY-1 with status BAD, SCHEMAREGISTRY_SCHEMA_REGISTRY_SERVER-2 (e0e64fc1-21ee-457f-ba12-0b94f9b6b6ba) has undesired health: BAD from failed check(s): SCHEMAREGISTRY_SCHEMA_REGISTRY_SERVER_SCM_HEALTH: BAD, SCHEMAREGISTRY_SCHEMA_REGISTRY_SERVER-1 (043f7f4a-c46d-42af-8d82-061cba1c3f7c) has undesired health: BAD from failed check(s): SCHEMAREGISTRY_SCHEMA_REGISTRY_SERVER_SCM_HEALTH: BAD

1. Log in to Cloudera Manager.
2. Navigate to Clusters.
3. Click HDFS.
4. Go to the Configurations tab.
5. In the Search tab, search for the HDFS Client Advanced Configuration Snippet (Safety Valve) parameter for hdfs-site.xml.
6. Add the following:

```
dfs.client.block.write.replace-datanode-on-failure.policy=ALWAYS
dfs.client.block.write.replace-datanode-on-failure.best-effort=true
dfs.client.block.write.replace-datanode-on-failure.enable=true
```

CDPD-56498: The Ozone service is in an unhealthy state and the basic canary test fails after Cloudera Manager upgrades from 7.7.1 to 7.11.3 or lower.

Error message: The health test result for OZONE_BASIC_HEALTH_CHECK has become bad: PERMISSION_DENIED USER <***username***> doesn't have READ permission to access volume Volume:s3v

You must add the following Ranger policies manually after you upgrade Cloudera Manager.

```
1. "name": "S3_VOLUME_POLICY_FOR_OZONE_CANARY", "resource.volume": "s3v", "users": "hue", "accessTypes": "create,write,read,list",
2. "name": "S3_BUCKET_POLICY_FOR_OZONE_CANARY", "resource.volume": "s3v", "resource.bucket": "cloudera-health-monitoring-ozone-basic-canary-bucket", "users": "hue", "accessTypes": "create,write,read,list",
3. "name": "S3_KEY_POLICY_FOR_OZONE_CANARY", "resource.volume": "s3v", "resource.bucket": "cloudera-health-monitoring-ozone-basic-canary-bucket", "resource.key": "cloudera-health-monitoring-ozone-basic-canary-key", "users": "hue", "accessTypes": "create,write,read,list,delete"
```

The Ozone canary is enabled in OZONE CSD supporting CDP Private Cloud Base 7.1.7, 7.1.8, and 7.1.9. So, after upgrading from the older Cloudera Manager version to Cloudera Manager 7.11.3. You must add the above policies to the Cloudera Base on premises cluster on the upgraded Cloudera Manager. You must add the above policies for both the Hue system user and the Hue principal name if you have a custom principal name or system user for Hue. This works seamlessly with or without an ATL rule. The default value for users is "users": "hue". The SMON principal must be the same as the Hue principal.

Ranger

CDPD-58860: As part of OPSAPS-67480 in CDP Private Cloud Base 7.1.9, default ranger policy is added from cdp-proxy-token topology, so that after a new installation of CDP Private Cloud Base 7.1.9, the Knox-Ranger policy includes cdp-proxy-token. However, upgrades do not add cdp-proxy-token to cm_knox policies automatically.

Error message: None

Manually add cdp-proxy-token to the Knox policy, using **Ranger Admin Web UI**.

1. Log in to Cloudera Manager Ranger Admin Web UI, as a Ranger administrator.
2. On Ranger Admin Web UI Service Manager Resource Knox, click cm_knox.
3. In Knox Policies, open the CDP Proxy UI, API and Token policy.
4. In Knox Topology*, add cdp-proxy-token.
5. Click Save.
6. Restart Ranger.

YARN/MapReduce

CDPD-59179: Rolling upgrades from CDP Private Cloud Base 7.1.7 SP2 to CDP Private Cloud Base 7.1.9 upgrade, the Map Reduce Sleep job's reduce task could fail after rolling upgrade.

After upgrading from CDP Private Cloud Base 7.1.7 SP2 to 7.1.9, the map task is completed as "mapreduce.Job: map 100% reduce 0%" but when the reduce is executed, an error occurs only if Resource Manager has a delay in starting.

None.

COMPX-18506: QueueManager Fails to restart after parcel downgrade

Downgrading from 7.3.1 to CDP Private Cloud Base 7.1.8 fails for Queue Manager citing incorrect username/password for Config Store.

1. SSH into the Queue Manager instance
2. Navigate to the path where config store H2 DB is (default path: /var/lib/hadoop-yarn)
3. Rename the following two DB files:
 - config-service.mv.db
 - config-service.trace.db
4. After this restart Queue Manager or continue downgrade



Note: This will result in losing user version change history.

CDPQE-36036: Change QueueManager backup and restore process during rollback above CDP Private Cloud Base 7.1.9

Downgrading from Cloudera Base on premises 7.3.1 to CDP Private Cloud Base 7.1.9 SP1 fails for Queue Manager.

Perform Yarn QueueManager db restore instead of QueueManager config restore because, from CDP Private Cloud Base 7.1.9.0 Queue Manager uses PSQL DB instead of H2 so the backup and restore process is updated accordingly.

Data Definition Language (DDL) - Impala, Hive (using HiveQL), Spark (using SparkSQL), HBase, Phoenix, Flink, and Kafka

CDPD-60862: Rolling restart fails during ZDU when DDL operations are in progress

During a Zero Downtime Upgrade (ZDU), the rolling restart of services that support Data Definition Language (DDL) statements might fail if DDL operations are in progress during the upgrade. As a result, ensure that you do not run DDL statements during ZDU.

The following services support DDL statements:

- Impala
- Hive – using HiveQL
- Spark – using SparkSQL
- HBase

- Phoenix
- Kafka

Data Manipulation Language (DML) statements are not impacted and can be used during ZDU. Following the successful upgrade, you can resume running DDL statements.

None. recommends modifying applications to not use DDL statements for the duration of the upgrade. If the upgrade is already in progress, and you have experienced a service failure, you can remove the DDLs in-flight and resume the upgrade from the point of failure.

Related Information

[Procedure to Downgrade or Rollback from CDP 7.1.9](#)

Applying a Service Pack

Applying a Service Pack is completed without running the Upgrade Wizard.



Important: To perform rollback from a just installed SP (Service Pack) release to the previously installed SP release, provided both releases belong to the same Cloudera Base on premises release line, see [Rollback between SP or CHF releases](#).



Important: Any prior upgrades must be finalized before applying a service pack. See [Step 8: Finalize the HDFS or Ozone Upgrade](#) on page 133.



Important: For the upgrades to CDP Private Cloud Base 7.1.7 Service Pack 2 (SP2), note the following requirements for upgrades to CDP Private Cloud Base 7.1.7 SP2:

- Upgrades CDP Private Cloud Base 7.1.7 Service Pack 2 are only supported from CDP Private Cloud Base 7.1.7.
- You must first upgrade to Cloudera Manager 7.6.7, which is the supported version for Service Pack 2.
- You cannot use Cloudera Manager 7.6.7 to manage deployments where Cloudera Manager is managing a CDP Private Cloud Data Services cluster.

To apply a service pack:

1. Log in to the Cloudera Manager Admin Console.
2. Ensure that you have completed the steps to add the Parcel URL for the service pack and have downloaded and distributed the parcel in Cloudera Manager. See [Step 5: Access Parcels](#) on page 131.
3. Click Parcels from the left menu.
4. Click Parcel Repositories & Network Settings.
5. Locate the row in the table that contains the new Cloudera Runtime parcel and click the Activate button.

Cloudera Manager displays the progress of the activation.

6. When the parcel is activated and complete for all hosts in the cluster, click the Actions menu next to the cluster name and select Post Cloudera Runtime Upgrade .
7. Restarting the cluster:
 - With downtime on services: Click the Actions menu and select Restart.
 - or
 - Rolling Restart the cluster to avoid downtime on services: Click the Actions menu and select Rolling Restart.

The cluster now has the new service pack applied.



Important: If a Ranger database patch is included in a minor version upgrade, such as CDP Private Cloud Base 7.1.7 SP2 CHF4+ or CDP Private Cloud Base 7.1.8 CHF5 or later versions, you must apply patches to the Ranger and Ranger RMS database manually. Perform the following steps:

1. Stop the Ranger and Ranger RMS services.
2. Go to Cloudera Manager Ranger Actions.
3. In Actions, click Upgrade Ranger Database and apply patches, then click Confirm.
4. Restart Ranger and Ranger RMS services.

As a result, DB patches will be applied to the Ranger database. When RMS server is started, it will perform a full-sync as RMS tables will be truncated during this procedure.

Mapreduce and Tez framework archives



Note: Currently, the Mapreduce and Tez framework archives are not automatically reinstalled during a parcel upgrade to the same minor version. For example, upgrading to a Service Pack or CHF version of the same runtime version.

As a post-upgrade step, you must check that location in HDFS to ensure that it is the latest version, and run the following if it does not:

```
hdfs dfs -ls /user/yarn/mapreduce/mr-framework/
```

If only the old version's tar file exists there, such as:

```
/user/yarn/mapreduce/mr-framework/3.1.1.7.1.7.78-12-mr-framework.tar.gz
```

Then perform the following: Cloudera Manager > Yarn > Actions > Install YARN MapReduce Framework JARs. You can clear out the old tar file after this is done for keeping it clean.

Tez MR Framework JARs and Oozie Sharelib entities must be examined in the similar manner. Check that location in HDFS to ensure that it is the latest version, and run the following if it does not:

1. Cloudera Manager > Yarn > Actions > Install YARN MapReduce Framework JARs
2. Cloudera Manager > Tez > Actions > Upload Tez tar file to HDFS
3. Cloudera Manager > Oozie > Actions > Install Oozie ShareLib

Clear any old files out of these directories:

1.

```
hdfs dfs -ls -R /user/yarn/mapreduce/mr-framework/
```
2.

```
hdfs dfs -ls -R /user/tez/
```
3.

```
hdfs dfs -ls /user/oozie/share/lib/
```

How to Install Cumulative Hotfix (CHF)

You can upgrade CDP Private Cloud Base 7.1.7, CDP Private Cloud Base 7.1.7 SP1, CDP Private Cloud Base 7.1.7.SP2, CDP Private Cloud Base 7.1.7.SP3, CDP Private Cloud Base 7.1.8, CDP Private Cloud Base 7.1.9 SP1, Cloudera Base on premises 7.3.1 CHF1 (7.3.1.100) and so on with the CHFs available. Cumulative Hotfix is a collection of hotfixes. You can use the same procedure to install hotfixes as well.

Before you begin



Important: To perform rollback from a just installed CHF (Cumulative Hotfix) release to the previously installed CHF release, provided both releases belong to the same Cloudera Base on premises release line, see [Rollback between SP or CHF releases](#).

To upgrade the Cloudera Base on premises cluster with the CHF or a hotfix:

1. Log in to the Cloudera Manager Admin Console.
2. Click Parcels from the left menu.
3. Click Parcel Repositories & Network Settings.



Note: If the CHF URL is available in the parcel repository, then you can skip to step 8. If not, then proceed with step 4.

4. In the Remote Parcel Repository URLs section, click the "+" icon and add the CHF URL for your Parcel repository.
5. Click Save & Verify Configuration. A message with the status of the verification appears above the Remote Parcel Repository URLs section. If the URL is not valid, check the URL and enter the correct URL.
6. After the URL is verified, click Close.
7. Locate the row in the table that contains the new Cloudera Runtime parcel and click the Download button.
8. After the download of the new Cloudera Runtime CHF parcel is complete, click the Distribute button.

Wait for the parcel to be distributed and unpacked before continuing. Cloudera Manager displays the status of the Cloudera Runtime CHF parcel distribution. Click on the status display to view detailed status for each host.

9. Click Activate. Cloudera Runtime CHF parcels are activated on the cluster.
10. When the parcel is activated, click the Cloudera Manager logo to return to the home page.

The cluster is now updated with the CHF you applied.

11. When the parcel is activated, click the Actions menu next to the cluster name and select Post Cloudera Runtime Upgrade.
12. Restart the cluster: Click the Actions menu and select Restart.



Important: If a Ranger database patch is included in a minor version upgrade, such as CDP Private Cloud Base 7.1.7 SP2-CHF4+ or CDP Private Cloud Base 7.1.8 CHF5+ , you must apply patches to the Ranger and Ranger RMS database manually. Perform the following steps:

- a. Stop the Ranger and Ranger RMS services.
- b. Go to Cloudera Manager Ranger Actions .
- c. In Actions, click Upgrade Ranger Database and apply patches, then click Confirm.
- d. Restart Ranger and Ranger RMS services.

As a result, DB patches will be applied to the Ranger database. When RMS server is started, it will perform a full-sync as RMS tables will be truncated during this procedure.

Mapreduce and Tez framework archives



Note: Currently, the Mapreduce and Tez framework archives are not automatically reinstalled during a parcel upgrade to the same minor version. For example, upgrading to a Service Pack or CHF version of the same runtime version.

As a post-upgrade step, you must check that location in HDFS to ensure that it is the latest version, and run the following if it does not:

```
hdfs dfs -ls /user/yarn/mapreduce/mr-framework/
```

If only the old version's tar file exists there, such as:

```
/user/yarn/mapreduce/mr-framework/3.1.1.7.1.7.78-12-mr-framework.tar.gz
```

Then perform the following: Cloudera Manager > Yarn > Actions > Install YARN MapReduce Framework JARs. You can clear out the old tar file after this is done for keeping it clean.

Tez MR Framework JARs and Oozie Sharelib entities must be examined in the similar manner. Check that location in HDFS to ensure that it is the latest version, and run the following if it does not:

1. Cloudera Manager > Yarn > Actions > Install YARN MapReduce Framework JARs
2. Cloudera Manager > Tez > Actions > Upload Tez tar file to HDFS
3. Cloudera Manager > Oozie > Actions > Install Oozie ShareLib

Clear any old files out of these directories:

1.

```
hdfs dfs -ls -R /user/yarn/mapreduce/mr-framework/
```
2.

```
hdfs dfs -ls -R /user/tez/
```
3.

```
hdfs dfs -ls /user/oozie/share/lib/
```

Manual upgrade to Cloudera Base on premises

Manual steps to follow for upgrading a Cloudera Runtime cluster to a higher version of Cloudera Runtime if the Upgrade Wizard fails.



Warning: The upgrade wizard does not allow a manual upgrade option. There is no route or an option from Cloudera Manager to skip restart of custom services. In this scenario, the users must proceed with manual upgrade steps.



Important:

If Cloudera Base on premises Upgrade Wizard fails with any error, search for the Knowledge Base article from the [MyCloudera](#) website to find the failure resolution for that particular error. Perform the steps that are detailed in the Knowledge Base article to resolve the error and continue with the manual upgrade process. If you do not find the Knowledge Base article for any specific failure, contact [Cloudera Support](#) to open a support case.



Important:

Perform the steps in this section only if the upgrade wizard reports a failure, or if you selected Manual Upgrade from the Upgrade Wizard (the Manual Upgrade option is only available for minor or maintenance upgrades). Manual upgrades allow you to selectively stop and restart services to prevent or mitigate downtime for services or clusters where rolling restarts are not available.

All the following steps assume the starting Cloudera Runtime version is at least 7.0.3, because those are the lowest versions that Cloudera Manager 7.1 supports. The following steps are required when you are upgrading Cloudera Runtime cluster from Cloudera Runtime 7.0.3 to Cloudera Runtime 7.1.1 or higher versions.

The following steps should be executed roughly in the order that they are listed, and should only be executed if the service is configured.

Upgrade Ranger database and apply patches

1. Go to the RANGER service.
2. Select ActionsUpgrade Ranger Database and apply patches and click Upgrade Ranger Database and apply patches to confirm.

Setup Ranger Admin Component

1. Go to the Ranger service.
2. Select ActionsSetup Ranger Admin Component and click Setup Ranger Admin Component to confirm.

Start Ranger

1. Go to the Ranger service.
2. Select ActionsStart.

Set up the Ranger Plugin service

1. Go to the Ranger service.
2. Select ActionsSetup Ranger Plugin Service and click Setup Ranger Plugin Service to confirm.

Start Kudu

1. Go to the KUDU service.
2. Select ActionsStart.

Start ZooKeeper

1. Go to the ZooKeeper service.
2. Select ActionsStart.

Upgrade HDFS Metadata

1. Go to the HDFS service.
2. Select ActionsUpgrade HDFS Metadata and click Upgrade HDFS Metadata to confirm.

Start HDFS

1. Go to the HDFS service.
2. Select ActionsStart.

Start YARN QueueManager

1. Go to the QueueManager service.
2. Select ActionsStart.

Import Sentry Policies to Ranger

1. Go to the HDFS service.
2. Select ActionsImport Sentry Policies into Ranger and click Import Sentry Policies into Ranger to confirm.

Start HBASE

1. Go to the HBASE service.

2. Select ActionsStart.

Start YARN QueueManager

1. Go to the QueueManager service.
2. Select ActionsStart.

Clean NodeManager Recovery Directory (YARN)

1. Go to the YARN service.
2. Select ActionsClean NodeManager Recovery Directory and click Clean NodeManager Recovery Directory to confirm.

Reset ACLs on YARN Zookeeper nodes

Any other upgrade if Enable ResourceManager Recovery is enabled for a Resource Manager group (for example, ResourceManager Default Group) and ZooKeeper is a dependency of YARN. Note that when YARN is running in High Availability mode, ResourceManager recovery is always enabled.

1. Go to the YARN service.
2. Select Actions Reset ACLs on YARN Zookeeper nodes
3. Click Reset ACLs on YARN Zookeeper nodes to confirm.

Install YARN MapReduce Framework Jars

1. Go to the YARN service.
2. Select ActionsInstall YARN MapReduce Framework JARs and click Install YARN MapReduce Framework JARs to confirm.

Start YARN

1. Go to the YARN service.
2. Select ActionsStart.

Deploy Client Configuration Files

1. On the Home page, click to the right of the cluster name and select Deploy Client Configuration.
2. Click the Deploy Client Configuration button in the confirmation pop-up that appears.

Reinitialize Solr State for Upgrade



Warning: Performing this step deletes the entire Solr collection.

1. Go to the SOLR service.
2. Select ActionsReinitialize Solr State for Upgrade and click Reinitialize Solr State for Upgrade to confirm.

Bootstrap Solr Configuration

1. Go to the SOLR service.
2. Select **Actions Bootstrap Solr Configuration** and click **Bootstrap Solr Configuration** to confirm.

Start Solr

1. Go to the SOLR service.
2. Select **ActionsStart**.

Bootstrap Solr Collections

1. Go to the SOLR service.
2. Select **Actions Bootstrap Solr Collections** and click **Bootstrap Solr Collections** to confirm.

Create HDFS Home directory

1. Go to the infrastructure SOLR service.
2. Select **ActionsCreate HDFS Home Dir** and click **Create HDFS Home Dir** to confirm.

Create Ranger Plugin Audit Directory

1. Go to the Solr service.
2. Select **ActionsCreate Ranger Plugin Audit Directory** and click **Create Ranger Plugin Audit Directory** to confirm.

Start infrastructure Solr

1. Go to the infrastructure SOLR service.
2. Select **ActionsStart**.

Start HBASE

1. Go to the HBASE service.
2. Select **ActionsStart**.

Start KAFKA

1. Go to the KAFKA service.
2. Select **ActionsStart**.

Create Ranger Kafka Plugin Audit Directory

1. Go to the KAFKA service.

2. Select ActionsCreate Ranger Kafka Plugin Audit Directory and click Create Ranger Kafka Plugin Audit Directory to confirm.

Create HBase tables for Atlas

1. Go to the ATLAS service.
2. Select ActionsCreate HBase tables for Atlas and click Create HBase tables for Atlas to confirm.

Start Atlas

1. Go to the ATLAS service.
2. Select ActionsStart.

Create Ranger Atlas Plugin Audit Directory

1. Go to the ATLAS service.
2. Select ActionsCreate Ranger Atlas Plugin Audit Directory and click Create Ranger Atlas Plugin Audit Directory to confirm.

Start Phoenix

1. Go to the PHOENIX service.
2. Select ActionsStart.

Install MapReduce Framework Jars

1. Go to the YARN service.
2. Select ActionsInstall YARN MapReduce Framework JARs and click Install YARN MapReduce Framework JARs to confirm.

Start YARN

1. Go to the YARN service.
2. Select ActionsStart.

Deploy Client Configuration Files

1. On the Home page, click to the right of the cluster name and select Deploy Client Configuration.
2. Click the Deploy Client Configuration button in the confirmation pop-up that appears.

Upgrade the Hive Metastore Database



Warning: Your upgrade will fail if you do not complete this step.

1. Go to the Hive service.
2. If the Hive service is running, stop it:
 - a. Select ActionsStop and click Stop to confirm.
3. Select ActionsUpgrade Hive Metastore Database Schema and click Upgrade Hive Metastore Database Schema to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.
5. Select ActionsValidate Hive Metastore Schema and click Validate Hive Metastore Schema to check that the schema is now valid.

Start Hive

1. Go to the Hive service.
2. Select ActionsStart.

Create Hive Warehouse Directory

1. Go to the HIVE service.
2. Select Actions Create Hive Warehouse Directory and click Create Hive Warehouse Directory to confirm.

Create Hive Warehouse External Directory

1. Go to the HIVE service.
2. Select Actions Create Hive Warehouse External Directory and click Create Hive Warehouse External Directory to confirm.

Create Hive Sys database

1. Go to the HIVE service.
2. Select ActionsCreate Hive Sys database and click Create Hive Sys database to confirm.

Create Ranger Plugin Audit Directory

1. Go to the HIVE service.
2. Select Actions Create Ranger Plugin Audit Directory and click Create Ranger Plugin Audit Directory to confirm.

Start Impala

1. Go to the Impala service.
2. Select ActionsStart.

Create Ranger Plugin Audit Directory

1. Go to the Impala service.
2. Select Actions Create Ranger Plugin Audit Directory and click Create Ranger Plugin Audit Directory to confirm.

Create Spark Driver Log Dir

1. Go to the SPARK_ON_YARN service.
2. Select Actions>Create Spark Driver Log Dir and click Create Spark Driver Log Dir to confirm.

Start Spark

1. Go to the SPARK_ON_YARN service.
2. Select ActionsStart.

Start Livy

1. Go to the LIVY service.
2. Select ActionsStart.

Upgrade Oozie Database Schema

1. Go to the OOZIE service.
2. If the OOZIE service is running, stop it:
Select Actions > Stop and click Stop to confirm.
3. Select Actions Upgrade Oozie Database Schema and click Upgrade Oozie Database Schema to confirm.

Updating column types in Oozie Database

If the Oracle database is used for Oozie, then you must update the APP_PATH column type to store values with more than 255 characters. This ensures Oozie does not get stuck in PREP state when your application path exceeds the 255 character limit. If the APP_PATH column type is not updated, then Oozie fails to run the jobs with the following database error message Data too long for column 'app_path'. This scenario is also applicable to coordinator and bundle jobs. For database types other than Oracle, this update is not mandatory for using Oozie. It works without the update if the APP_PATH value does not exceed 255 characters. Also, Oozie's internal database schema validation fails with an unexpected APP_PATH column type. However, this validation does not have any effect. It just logs it's result.

You must update the APP_PATH column type in the WF_JOBS, BUNDLE_JOBS, and COORD_JOBS tables in the Oozie database for the following conditions:

- When you use the **Oracle** database for Oozie service.
- When you are upgrading Cloudera Runtime from earlier versions to Cloudera Base on premises 7.1.9 SP1 CHF1 version or later.

If you do not execute the following statements on the Oozie Oracle database, then the Oozie service fails to run the jobs with a database persistence error. This update is not mandatory, but highly recommended for other database types, such as MySQL, MariaDB, and PostgreSQL, due to the internal database schema validation.

Examples:

- On the Oozie Oracle database - The following example uses the Oracle sqlplus command-line tool:

```
sqlplus <OOZIE_DB_USERNAME>@localhost /<SERVICE_NAME>
SQL> ALTER TABLE <TABLE_NAME> ADD (APP_PATH_TMP CLOB);
```

```

Table altered.

SQL> UPDATE <TABLE_NAME> SET APP_PATH_TMP = APP_PATH;

X rows updated.

SQL> ALTER TABLE <TABLE_NAME> DROP COLUMN APP_PATH;

Table altered.

SQL> ALTER TABLE <TABLE_NAME> RENAME COLUMN APP_PATH_TMP TO APP_PATH;

Table altered.

```

- On the Oozie MySQL database - The following example uses the MySQL mysql command-line tool:

```

$ mysql -u root -p
Enter password:

mysql> use <OOZIE_DATABASE_NAME>;
Database changed

mysql> ALTER TABLE <TABLE_NAME> MODIFY COLUMN app_path text;
Query OK, X rows affected (0.03 sec)
Records: X Duplicates: 0 Warnings: 0
mysql> exit
Bye

```

- On the Oozie MariaDB database - The following example uses the MariaDB mysql command-line tool:

```

$ mysql -u root -p
Enter password:

MariaDB [(none)]> use <OOZIE_DATABASE_NAME>;
Database changed

MariaDB [OOZIE_DATABASE_NAME]> ALTER TABLE <TABLE_NAME> MODIFY COLUMN a
pp_path text;
Query OK, X rows affected (2.11 sec)
Records: X Duplicates: 0 Warnings: 0

MariaDB [OOZIE_DATABASE_NAME]> exit
Bye

```

- On the Oozie PostgreSQL database - The following example uses the PostgreSQL psql command-line tool:

```

$ psql -U postgres
Password for user postgres: *****

postgres=# \c <OOZIE_DATABASE_NAME>;
You are now connected to database "<OOZIE_DATABASE_NAME>" as user "postgre
s".
OOZIE_DATABASE_NAME=# ALTER TABLE <TABLE_NAME> ALTER COLUMN app_path type
text;
ALTER TABLE

OOZIE_DATABASE_NAME=# \q

```

Upgrade Oozie SharedLib

1. Go to the Oozie service.
2. If the OOZIE service is stopped, start it:
Select ActionsStart and click Start to confirm.
3. Select ActionsInstall Oozie SharedLib and click Install Oozie SharedLib to confirm.

Upload Tez tar file to HDFS

1. Go to the TEZ service.
2. Select Actions > Upload Tez tar file to HDFS and click Upload Tez tar file to HDFS to confirm.

Migrate Hive tables for Cloudera Base on premises upgrade

1. Go to the HIVE_ON_TEZ service.
2. Select ActionsMigrate Hive tables for CDP upgrade and click Migrate Hive tables for CDP upgrade to confirm.

Create Ranger Plugin Audit Directory

1. Go to the Hive-on-Tez service.
2. Select Actions Create Ranger Plugin Audit Directory and click Create Ranger Plugin Audit Directory to confirm.

Start Hive on Tez

1. Go to the Hive-on-Tez service.
2. Select ActionsStart.

Start Hue

1. Go to the HUE service.
2. Select ActionsStart.

Start the Remaining Cluster Services

1. Use rolling restart or full restart.
2. Ensure that all services are started or restarted. You can use Cloudera Manager to start the cluster, or you can restart the services individually. The Cloudera Manager Home page indicates which services have stale configurations and require restarting.
3. To start or restart the cluster:
 - a. On the Home Status page, click the down arrow to the right of the cluster name and select Start or Restart.
 - b. Click Start that appears in the next screen to confirm. The Command Details window shows the progress of starting services.
 - c. When All services successfully started appears, the task is complete and you can close the Command Details window.

Validate the Hive Metastore Database Schema



Warning: Your upgrade will fail if you do not complete this step.

1. Select ActionsValidate Hive Metastore Schema and click Validate Hive Metastore Schema to confirm.
2. If you have multiple instances of Hive, perform the validation on each metastore database.
3. Select ActionsValidate Hive Metastore Schema and click Validate Hive Metastore Schema to check that the schema is now valid.

Test the Cluster and Finalize HDFS Metadata

To determine if you can finalize the upgrade, run important workloads and ensure that they are successful. After you have finalized the upgrade, you cannot roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time.

When you are ready to finalize the upgrade, do the following:

- 1. Go to the HDFS service.
- 2. Click the Instances tab.
- 3. Click the link for the NameNode instance. If you have enabled high availability for HDFS, click the link labeled NameNode (Active).

The NameNode instance page displays.

4. Select Actions Finalize Metadata Upgrade and click Finalize Metadata Upgrade to confirm.

Clear the Upgrade State Table

After completing all of the previous steps, do the following to complete the upgrade:

1. Log in to the Cloudera Manager server host.
2. Stop the Cloudera Manager Server.

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

3. Log in to the command-line environment for the Cloudera Manager database. (mysql, sqlplus, or postgres psql).
4. Run the following command:

```
DELETE FROM UPGRADE_STATE;
```

5. Start the Cloudera Manager Server.

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

Troubleshooting Upgrades

Cloudera Runtime upgrade failure



Warning: If there is any failure during the Cloudera Runtime upgrade process, you must not delete or modify the records in the upgrade_state table. If you still want to delete or modify the upgrade_state table, you must contact the Cloudera development team for modification or deletion approval. If you proceed without approval from the Cloudera development team, it can cause additional issues while resolving the runtime upgrade failure.

"Access denied" in install or update wizard

"Access denied" in install or update wizard during database configuration for Activity Monitor or Reports Manager.

Possible Reasons

Hostname mapping or permissions are not set up correctly.

Possible Solutions

- 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' IDENTIFIED 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_password';
```


the value you enter for the database hostname can be either the FQDN or localhost.

The Hive on Tez service and the RangerAdmin role of the Ranger service kept going down

Possible Reasons

Both issues were caused by their respective Java heap size configuration being very low.

Possible Solutions

- For Hive on Tez: Look up the hiveserver2_java_heapsize config key under Hive on Tez > Configuration . Later, change its value from 256 MiB to 2 GiB.
-  **Note:** The heap size can be verified on hive_on_tez configuration page, search for "heap".
- For Ranger: Look up the ranger_admin_max_heap_size config key under Ranger > Configuration. Later, change its value from 1 GiB to 4 GiB.

Both the services must be restarted.

Cluster hosts do not appear

Some cluster hosts do not appear when you click Find Hosts in install or update wizard.

Possible Reasons

You might have network connectivity problems.

Possible Solutions

- 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.

Cannot start services after upgrade

You have upgraded the Cloudera Manager Server, but now cannot start services.

Possible Reasons

You might have mismatched versions of the Cloudera Manager Server and Agents.

Possible Solutions

Make sure you have upgraded the Cloudera Manager Agents on all hosts. (The previous version of the Agents will heartbeat with the new version of the Server, but you cannot start HDFS and MapReduce with this combination.)

HDFS DataNodes fail to start

After upgrading, HDFS DataNodes fail to start with exception:

```
Exception in secureMainjava.lang.RuntimeException: Cannot start datanode because the configured max locked memory size (dfs.datanode.max.locked.memory) of 4294967296 bytes is more than the datanode's available RLIMIT_MEMLOCK ulimit of 65536 bytes.
```

Possible Reasons

HDFS caching, which is enabled by default in CDH 5 and higher, requires new memlock functionality from Cloudera Manager Agents.

Possible Solutions

Do the following:

1. Stop all CDH and managed services.
2. On all hosts with Cloudera Manager Agents, hard-restart the Agents. Before performing this step, ensure you understand the semantics of the `hard_restart` command by reading [Cloudera Manager Agents](#).
RHEL 7, SLES 12, Ubuntu 18.04 and higher

```
sudo systemctl stop cloudera-scm-supervisord.service
sudo systemctl restart cloudera-scm-agent
```

3. Start all services.

Cloudera services fail to start

Cloudera services fail to start.

Possible Reasons

Java might not be installed or might be installed at a custom location.

Possible Solutions

See [Configuring a Custom Java Home Location](#) on page 37 for more information on resolving this issue.

Host Inspector Fails

If you see the following message in the Host Inspector:

There are mismatched versions across the system, which will cause failures. See below for details on which hosts are running what versions of components.

When looking at the results, some hosts report Supervisor vX.X.X, while others report X.X.X-cmY.Y.Y (where X and Y are version numbers). During the upgrade, an old file on the hosts may cause the Host Inspector to indicate mismatched Supervisor versions.

This issue occurs because these hosts have a file on them at `/var/run/cloudera-scm-agent/supervisor/__STARTING_CM_VERSION__` that contains a string for the older version of Cloudera Manager.

To resolve this issue:

1. Remove or rename the `/var/run/cloudera-scm-agent/supervisor/__STARTING_CM_VERSION__` file
2. Perform a hard restart of the agents:

```
sudo systemctl stop cloudera-scm-supervisord.service
sudo systemctl start cloudera-scm-agent
```

3. Run the Host inspector again. It should pass without the warning.

Configuring a Local Package Repository

You can create a package repository for Cloudera Manager either by hosting an internal web repository or by manually copying the repository files to the Cloudera Manager Server host for distribution to Cloudera Manager Agent hosts.



Note: This internal web repository can also be used when your Cloudera Manager server or cluster does not have access to internet. You must download the installable separately from archive.cloudera.com and place the installable in the internal repository.



Important: Select a supported operating system for the versions of or CDH that you are downloading. See [CDH and Supported Operating Systems](#).



Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported



Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher



Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.



Warning: Before upgrading CDH 5 clusters with Sentry to 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.



Important: If HDFS ACL sync is enabled (`hdfs_sentry_sync_enable=true`) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see [Installing Ranger RMS](#).



Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

Creating a Permanent Internal Repository

The following sections describe how to create a permanent internal repository using Apache HTTP Server:

Setting Up a Web server

To host an internal repository, you must install or use an existing Web server on an internal host that is reachable by the Cloudera Manager host, and then download the repository files to the Web server host. The examples in this section use Apache HTTP Server as the Web server. If you already have a Web server in your organization, you can skip to [Downloading and publishing the package repository for Cloudera Manager](#) on page 164.

1. Install Apache HTTP Server:

RHEL / CentOS

```
sudo yum install httpd
```

SLES

```
sudo zypper install httpd
```

Ubuntu

```
sudo apt-get install httpd
```

2. Start Apache HTTP Server:

RHEL 8

```
sudo systemctl start httpd
```

SLES 12, Ubuntu 16 or later

```
sudo systemctl start apache2
```

Downloading and publishing the package repository for Cloudera Manager

1. Download the package repository for the product you want to install:

Cloudera Manager 7

Do the following steps to download the files for a Cloudera Manager release:

- a. Run the following command to create a local repository directory to hold the Cloudera package repository:

```
sudo mkdir -p /var/www/html/cloudera-repos/cm7
```

- b. Run the following command to download the repository tarball for your operating system:

```
wget https://[USERNAME]:[PASSWORD]@archive.cloudera.com/p/cm7/7.0.3/repo-as-tarball/cm7.0.3-redhat7.tar.gz
```

- c. Run the following command to unpack the tarball into the local repository directory:

```
tar xvfz cm7.0.3-redhat7.tar.gz -C /var/www/html/cloudera-repos/cm7 --strip-components=1
```

- d. Run the following command to modify the file permission that allows you to download the files under the local repository directory:

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/cm7
```

2. Visit the Repository URL `http://<WEB_SERVER>/cloudera-repos/` in your browser and verify the files you downloaded are present.



Important: If you do not see the list of downloaded files in your web browser, then you might have been configured not to display indexes. Verify your web browser settings.



Important:

If you downloaded the rpm files individually instead of using `repo-as-tarball`, ensure that the `allkeys*.asc` files (`allkeys.asc` and the `allkeysha256.asc` for Cloudera Manager 7.11.2 and later versions) are present at the top level of the package repository. The `allkeys*.asc` files are included in the `repo-as-tarball` file. Ensure to include `allkeys*.asc` files, if you are manually copying the package files between hosts.

The `allkeys*.asc` files are used to validate the signatures of the package files during host installation. If `allkeys*.asc` files are not available in the repository, then you cannot add a host in the Cloudera Manager.

Creating a Temporary Internal Repository

You can quickly create a temporary remote repository to deploy packages on a one-time basis. Cloudera recommends using the same host that runs Cloudera Manager, or a gateway host. This example uses [Python SimpleHTTPServer](#) as the Web server to host the `/var/www/html` directory, but you can use a different directory.

1. Download the repository you need following the instructions in [Downloading and publishing the package repository for Cloudera Manager](#) on page 164.
2. Determine a port that your system is not listening on. This example uses port 8900.
3. Start a Python SimpleHTTPServer in the `/var/www/html` directory:

```
cd /var/www/html
python -m SimpleHTTPServer 8900
```

```
Serving HTTP on 0.0.0.0 port 8900 ...
```

4. Visit the Repository URL `http://<WEB_SERVER>:8900/cloudera-repos/` in your browser and verify the files you downloaded are present.

Configuring Hosts to Use the Internal Repository

After establishing the repository, modify the client configuration to use it:

OS	Procedure
RHEL compatible	<p>Create /etc/yum.repos.d/cloudera-repo.repo files on cluster hosts with the following content, where <code><WEB_SERVER></code> is the hostname of the Web server:</p> <pre>[cloudera-repo] name=cloudera-repo baseurl=http://<WEB_SERVER>/cm/5 enabled=1 gpgcheck=0</pre>
SLES	<p>Use the zypper utility to update client system repository information by issuing the following command:</p> <pre>zypper addrepo http://<WEB_SERVER>/cm <ALIAS></pre>
Ubuntu	<p>Create /etc/apt/sources.list.d/cloudera-repo.list files on all cluster hosts with the following content, where <code><WEB_SERVER></code> is the hostname of the Web server:</p> <pre>deb http://<WEB_SERVER>/cm <CODENAME> <COMPONENTS></pre> <p>You can find the <code><CODENAME></code> and <code><COMPONENTS></code> variables in the <code>./conf/distributions</code> file in the repository.</p> <p>After creating the <code>.list</code> file, run the following command:</p> <pre>sudo apt-get update</pre>

Configuring a Local Parcel Repository

You can create a parcel repository for Cloudera Runtime either by hosting an internal Web repository or by manually copying the repository files to the Cloudera Manager Server host for distribution to Cloudera Manager Agent hosts.

Using an Internally Hosted Remote Parcel Repository

The following sections describe how to use an internal Web server to host a parcel repository:

Setting Up a Web Server

To host an internal repository, you must install or use an existing Web server on an internal host that is reachable by the Cloudera Manager host, and then download the repository files to the Web server host. The examples on this page use Apache HTTP Server as the Web server. If you already have a Web server in your organization, you can skip to [Downloading and Publishing the Parcel Repository](#) on page 168.

1. Install Apache HTTP Server:

RHEL / CentOS


```
sudo yum install httpd
```

SLES

```
sudo zypper install httpd
```

Ubuntu

```
sudo apt-get install httpd
```

2.  **Warning:** Skipping this step could result in an error message Hash verification failed when trying to download the parcel from a local repository, especially in Cloudera Manager 6 and higher.

Edit the Apache HTTP Server configuration file (/etc/httpd/conf/httpd.conf by default) to add or edit the following line in the <IfModule mime_module> section:

```
AddType application/x-gzip .gz .tgz .parcel
```

If the <IfModule mime_module> section does not exist, you can add it in its entirety as follows:



Note: This example configuration was modified from the default configuration provided after installing Apache HTTP Server on RHEL.

```
<IfModule mime_module>
#
# TypesConfig points to the file containing the list of mappings from
# filename extension to MIME-type.
#
TypesConfig /etc/mime.types
#
# AddType allows you to add to or override the MIME configuration
# file specified in TypesConfig for specific file types.
#
#AddType application/x-gzip .tgz
#
# AddEncoding allows you to have certain browsers uncompress
# information on the fly. Note: Not all browsers support this.
#
#AddEncoding x-compress .Z
#AddEncoding x-gzip .gz .tgz
#
# If the AddEncoding directives above are commented-out, then you
# probably should define those extensions to indicate media types:
#
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz .parcel

#
# AddHandler allows you to map certain file extensions to "handlers":
# actions unrelated to filetype. These can be either built into the se
rver
# or added with the Action directive (see below)
#
# To use CGI scripts outside of ScriptAliased directories:
# (You will also need to add "ExecCGI" to the "Options" directive.)
#
#AddHandler cgi-script .cgi
```

```
# For type maps (negotiated resources):
#AddHandler type-map var

#
# Filters allow you to process content before it is sent to the client
.
#
# To parse .shtml files for server-side includes (SSI):
# (You will also need to add "Includes" to the "Options" directive.)
#
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
</IfModule>
```

3. Start Apache HTTP Server:

RHEL 8

```
sudo systemctl start httpd
```

SLES 12, Ubuntu 16 or later

```
sudo systemctl start apache2
```

Downloading and Publishing the Parcel Repository

1. Look up the Cloudera Runtime *VERSION* number for your deployment on the [Cloudera Runtime Download Information](#) page. You will need this version number in the next step.
2. Download manifest.json and the parcel files for the product you want to install:

To download the files for the latest Cloudera Runtime 7 release, run the following commands on the Web server host:

```
sudo mkdir -p /var/www/html/cloudera-repos
sudo wget --recursive --no-parent --no-host-directories https://
[username]:[password]@archive.cloudera.com/p/cdh7/CLOUDERA_RUNTIME
VERSION/parcels/ -P /var/www/html/cloudera-repos

sudo chmod -R ugo+rX /var/www/html/cloudera-repos/cdh7
```

3. Visit the Repository URL `http://<WEB_SERVER>/cloudera-repos/` in your browser and verify the files you downloaded are present. If you do not see anything, your Web server may have been configured to not show indexes.

Configuring Cloudera Manager to Use an Internal Remote Parcel Repository

1. Use one of the following methods to open the parcel settings page:
 - Navigation bar
 - a. Click the parcel icon in the top navigation bar or click Hosts and click the Parcels tab.
 - b. Click the Configuration button.
 - Menu
 - a. Select AdministrationSettings.
 - b. Select CategoryParcels.
2. In the Remote Parcel Repository URLs list, click the addition symbol to open an additional row.
3. Enter the path to the parcel. For example: `http://<WEB_SERVER>/cloudera-parcels/cdh7/7.2.16.0.0/`
4. Enter a Reason for change, and then click Save Changes to commit the changes.

Using a Local Parcel Repository

To use a local parcel repository, complete the following steps:

1. Open the Cloudera Manager Admin Console and navigate to the Parcels page.
2. Select Configuration and verify that you have a Local Parcel Repository path set. By default, the directory is `/opt/cloudera/parcel-repo`.
3. Remove any Remote Parcel Repository URLs you are not using, including ones that point to Cloudera archives.
4. Add the parcel you want to use to the local parcel repository directory that you specified. For instructions on downloading parcels, see [Downloading and Publishing the Parcel Repository](#) on page 168 above.
5. In the command line, navigate to the local parcel repository directory.
6. Create a SHA1 hash for the parcel you added and save it to a file named `PARCEL_NAME.parcel.sha`.

For example, the following command generates a SHA1 hash for the parcel `CDH-6.1.0-1.cdh6.1.0.p0.770702-el7.parcel`:

```
shasum CDH-6.1.0-1.cdh6.1.0.p0.770702-el7.parcel | awk '{ print $1 }'  
> CDH-6.1.0-1.cdh6.1.0.p0.770702-el7.parcel.sha
```

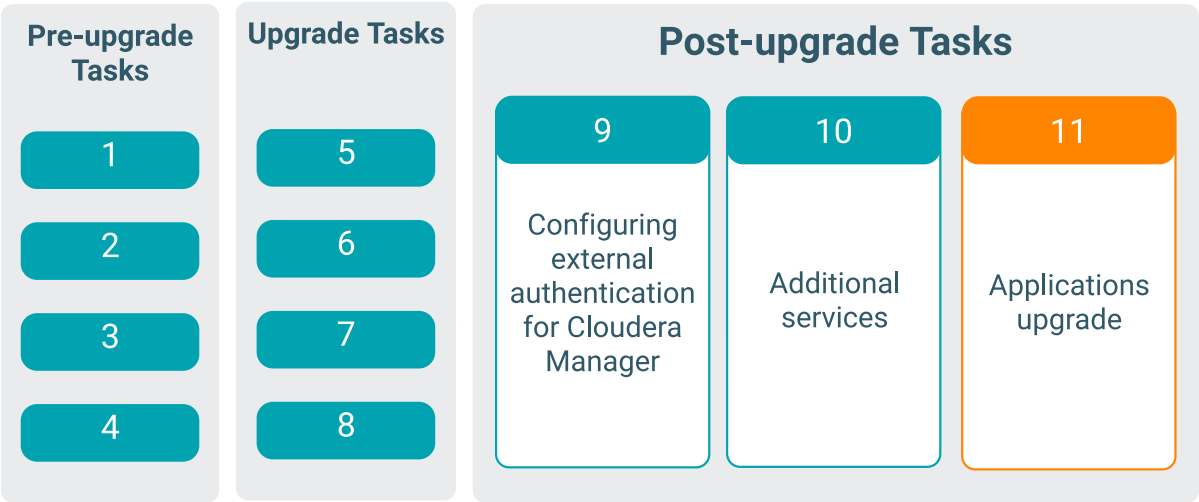
7. Change the ownership of the parcel and hash files to `cloudera-scm`:

```
sudo chown -R cloudera-scm:cloudera-scm /opt/cloudera/parcel-repo/*
```

8. In the Cloudera Manager Admin Console, navigate to the Parcels page.
9. Click Check for New Parcels and verify that the new parcel appears.
10. Download, distribute, and activate the parcel.

Applications Upgrade

After you upgrade, you must test all the services that run on your platform.



Ideally, you should have an indicative subset of jobs from your workloads. These are the tasks that you should have identified and run before the upgrade allowing you to compare pre-upgrade versus post-upgrade test results. These tests should also include any parts of the application that required code changes due to the changes in the platform. For example, to cater for changes in Hive managed versus external tables. The tests should also include a performance test. This can help to highlight missed or wrong configuration settings or point to other issues with the upgrade. Depending on your environment, perform these steps.

1. Update application code with changes required by the upgraded platform
2. Update the dependencies in the pom.xml file for custom jar files used in your applications to use the new dependencies for CDP Private Cloud Base.
3. Restart applications
4. Test the applications and verify they are functioning and performing as they were prior to upgrade



Note: After successfully upgrading from HDP to CDP Private Cloud Base, you can remove the HDP bits from the CDP Private Cloud Base cluster using the yum commands. Otherwise, when you run any security tool or security scanner for CVE detection, the HDP bits on the CDP Private Cloud Base cluster are detected as CVEs.

Rollback and Downgrade Cloudera Base on premises

Depending up the Cloudera Base on premises version, you can perform the rollback or downgrade steps.



Warning: If there is any failure during the upgrade process, Cloudera recommends you to contact Cloudera customer support.

Downgrade or Rollback from Cloudera Base on premises 7.3.1

You can rollback an upgrade from Cloudera Base on premises 7.3.1 to CDP Private Cloud Base 7.1.9 SP1, 7.1.9, [CDP 7.1.8 cumulative hotfix 17](#), or 7.1.7 SP3. You can downgrade an upgrade from Cloudera Base on premises 7.3.1 to CDP Private Cloud Base 7.1.9 SP1, 7.1.9, [CDP 7.1.8 cumulative hotfix 17](#), or 7.1.7 SP3. Before you downgrade or rollback, inspect each supported component to ensure that your intended process works seamlessly.

This section helps you understand and perform the downgrade or rollback process. In case you cannot upgrade your cluster due to an incompatible change, catastrophic failure, or accidental upgrade, you can use the below manual procedure to rollback or downgrade your cluster. In most cases, resolving the upgrade issue and continuing the upgrade is the preferred path. Performing a downgrade or rollback is an advanced operation. Not all components support downgrade and rollback. Contact Cloudera Support for guidance on performing a downgrade or rollback.

Downgrade - Downgrade restores the software to the prior release version but data remains intact. Service Interruptions may or may not be required for downgrade depending on the component.

Rollback - Rollback the state of the data of the upgraded components to the pre-upgrade state. A Rollback may require a service Interruption and is not an automated operation.



Attention: When you are rolling back from Cloudera Base on premises 7.3.1 version to CDP Private Cloud Base 7.1.9 or lower versions, follow these instructions:

- If you are using RHEL 7, there is no requirement to reinstall the Python 2.7 version.
- If you are using RHEL 8, SLES 15 SP4, and Ubuntu 20 operating systems, and in case you uninstalled the Python 2.7 version during your upgrade process, you must reinstall Python 2.7 first and then proceed towards the rollback process.

Review Limitations

The rollback procedure has the following limitations:

- HDFS – If you have finalized the HDFS upgrade, you cannot roll back your cluster.
- Compute clusters – Rollback for Compute clusters is not currently supported.
- Configuration changes, including the addition of new services or roles after the upgrade, are not retained after rolling back Cloudera Manager.



Note: Cloudera recommends that you not make configuration changes or add new services and roles until you have finalized the HDFS upgrade and no longer require the option to roll back your upgrade.

- HBase – If your cluster is configured to use HBase replication, data written to HBase after the upgrade might not be replicated to peers when you start your rollback. This topic does not describe how to determine which, if any, peers have the replicated data and how to roll back that data. For more information about HBase replication, see [HBase Replication](#).
- Kafka – Once the Kafka log format and protocol version configurations (the `inter.broker.protocol.version` and `log.message.format.version` properties) are set to the new version (or left blank, which means to use the latest version), Kafka rollback is not possible.

Downgrade from Cloudera Base on premises 7.3.1

Perform the below procedure to downgrade your cluster from Cloudera Base on premises 7.3.1 to CDP Private Cloud Base 7.1.9 SP1, 7.1.9, [CDP 7.1.8 cumulative hotfix 17](#), or 7.1.7 SP3.

Downgrade

Downgrade restores the software to the prior release version but data remains intact. Service interruptions are still expected, but the use of Rolling Restart minimizes it. After HDFS and/or Ozone is finalized, it is not possible to Downgrade or Rollback.

The following components do not support downgrade. If these components are in use, rollback may be necessary. Work with Cloudera Support to devise a plan based on the deployed components.

- Ranger
- Ranger RMS
- Ranger KMS
- KTS
- YARN Queue Manager
- Schema Registry
- Solr
- Atlas

Pre-downgrade steps

Ozone

This procedure is applicable only if you are downgrading from Cloudera Base on premises 7.3.1 to CDP Private Cloud Base 7.1.8.

1. Stop the Ozone Recon Web UI. Within Cloudera Manager UI, navigate to the Ozone service Ozone Recon Actions Stop this Ozone Recon .
2. Navigate to Configuration within the Ozone service and collect the value of `ozone.recon.db.dir` (default value is `/var/lib/hadoop-ozone/recon/data`).
3. SSH to the Ozone Recon Web UI host and move the `ozone.recon.db.dir` parent directory to a backup location: `mv /var/lib/hadoop-ozone/recon /var/lib/hadoop-ozone/recon-backup-CDP`.

HBase

Stop the HBase Master(s). Execute `kinit` as the `hbase` user if kerberos is enabled.

1. Stop Omid within Cloudera Manager UI
2. Navigate to the HBase service > Instances within Cloudera Manager UI and note the hostname of the HBase Master instance(s). Login to the host(s) and execute the following: `hbase master stop --shutDownCluster`
3. Stop the remaining HBase components. Navigate to the HBase service within Cloudera Manager UI Actions Stop

The following must be performed when downgrading to CDP Private Cloud Base 7.1.7 SP2 from Cloudera Base on premises 7.3.1. You will need to `kinit` as the `hbase` user if kerberos is enabled.

1. Contact support for the appropriate `hbck2 jar`

2. Execute a dry run of the `shortenTableinfo` command and validate the appropriate files have been identified `hbase --config /etc/hbase/conf hbck -j hbase-hbck2-X.Y.Z.jar shortenTableinfo`
3. Run the `shortenTableinfo -fix` command to fix the file format `hbase --config /etc/hbase/conf hbck -j hbase-hbck2-X.Y.Z.jar shortenTableinfo -fix`

Cruise Control

The following steps are applicable only if you downgrade from Cloudera Base on premises 7.3.1 to CDP Private Cloud Base 7.1.7 SP2 and not from Cloudera Base on premises 7.3.1 to CDP Private Cloud Base 7.1.8. You can skip this section if the Cruise Control Goal configurations were set to the default values before performing the upgrade. However, if the Cruise Control Goal configuration values were changed before performing the upgrade, then you must proceed with this section.

In Cruise Control, you must rename `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareDistributionGoal` to `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareGoal` in Cloudera Manager > Clusters > Cruise Control > Configurations tab in every occurrences as described below during downgrade process.

In Cruise Control, from CDP Private Cloud Base 7.1.8 and higher, `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareGoal` is renamed to `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareDistributionGoal`.

Perform the below steps:

1. Check the following goal sets if `RackAwareDistributionGoal` is present (Cloudera Manager > Clusters > Cruise Control > Configurations tab):
 - a. `default.goals`
 - b. `goals`
 - c. `self.healing.goals`
 - d. `hard.goals`
 - e. `anomaly.detection.goals`
2. Create a note for yourself about where `RackAwareDistributionGoal` were present
3. Remove `RackAwareDistributionGoal` from all of the goal lists
4. Perform the Cloudera Runtime downgrade process

Downgrading the Cloudera Runtime parcel

1. Navigate to Parcels within Cloudera Manager.
2. Locate the **CDP Private Cloud Base 7.1.7 SP3/7.1.8/7.1.9 SP1** parcel and click Activate.
3. Follow the wizard and address any issues from the various inspectors.



Note: Warnings are expected for HDFS and Ozone due to the unfinalized rolling upgrade, validate no other warnings exist.

4. When the parcel is activated, click the Actions menu next to the cluster name and select Post Cloudera Runtime Upgrade.

The upgrade activates the parcel and restarts services.

Restart Services with Stale Configuration

1. Navigate to Clusters and find the cluster being downgraded.
2. Inspect the list of services for the stale configuration indicator (yellow power button), if found, click on the indicator.
3. Follow the wizard to restart services with stale configuration.

Restore CDH Databases

Restore the following databases from the CDH backups. Follow the order below and restore only a single service at a time. Stop the service prior to restoring to the database. Start the service after restoring the database.

- Ranger
- Ranger KMS
- Stream Messaging Manager
- Schema Registry
- Hue (only if you are downgrading to CDP Private Cloud Base 7.1.7 SP2)

The steps for backing up and restoring databases differ depending on the database vendor and version you select for your cluster and are beyond the scope of this document.



Important: Restore the databases to their exact state when you took the backup. Do not merge in any changes that may have occurred during the subsequent upgrade.

See the following vendor resources for more information:

- MariaDB 10.2, 10.3, 10.4 and 10.5: <https://mariadb.com/kb/en/backup-and-restore-overview/>
- MySQL 5.7: <https://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html>
- MySQL 8: <https://dev.mysql.com/doc/refman/8.0/en/backup-and-recovery.html>
- PostgreSQL 10: <https://www.postgresql.org/docs/10/backup.html>
- PostgreSQL 11: <https://www.postgresql.org/docs/11/backup.html>
- PostgreSQL 12: <https://www.postgresql.org/docs/12/backup.html>
- PostgreSQL 13: <https://www.postgresql.org/docs/13/backup.html>
- PostgreSQL 14: <https://www.postgresql.org/docs/14/backup.html>
- Oracle 19c: <https://docs.oracle.com/en/database/oracle/oracle-database/19/index.html>

Rollback Cloudera Navigator Encryption Components

If you are rolling back any encryption components (Key Trustee KMS, HSM KMS, Key HSM, or Navigator Encrypt), first refer to:

- [HSM KMS High Availability Backup and Recovery](#)
- [Manually Backing Up Navigator Encrypt](#)

Start the Key Management Server

Restart the Key Management Server. Open the Cloudera Manager Admin Console, go to the KMS service page, and select Actions Start .

Rollback Key HSM

To roll back Key HSM:

1. Install the version of Navigator Key HSM to which you wish to roll back

Install the Navigator Key HSM package using yum:

```
sudo yum downgrade keytrustee-keyhsm
```

Cloudera Navigator Key HSM is installed to the /usr/share/keytrustee-server-keyhsm directory by default.

2. Rename Previously-Created Configuration Files

For Key HSM major version rollbacks, previously-created configuration files do not authenticate with the HSM and Key Trustee Server, so you must recreate these files by re-executing the setup and trust commands. First,

navigate to the Key HSM installation directory and rename the applications.properties, keystore, and truststore files:

```
cd /usr/share/keytrustee-server-keyhsm/
mv application.properties application.properties.bak
mv keystore keystore.bak
mv truststore truststore.bak
```

3. Initialize Key HSM

Run the service keyhsm setup command in conjunction with the name of the target HSM distribution:

```
sudo service keyhsm setup [keysecure|thales|luna]
```

For more details, see [Initializing Navigator Key HSM](#).

4. Establish Trust Between Key HSM and the Key Trustee Server

The Key HSM service must explicitly trust the Key Trustee Server certificate (presented during TLS handshake). To establish this trust, run the following command:

```
sudo keyhsm trust /path/to/key_trustee_server/cert
```

For more details, see [Establish Trust from Key HSM to Key Trustee Server](#).

5. Start the Key HSM Service

Start the Key HSM service:

```
sudo service keyhsm start
```

6. Establish Trust Between Key Trustee Server and Key HSM

Establish trust between the Key Trustee Server and the Key HSM by specifying the path to the private key and certificate:

```
sudo ktadmin keyhsm --server https://keyhsm01.example.com:9090 \
--client-certfile /etc/pki/cloudera/certs/mycert.crt \
--client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
```

For a password-protected Key Trustee Server private key, add the --passphrase argument to the command (enter the password when prompted):

```
sudo ktadmin keyhsm --passphrase \
--server https://keyhsm01.example.com:9090 \
--client-certfile /etc/pki/cloudera/certs/mycert.crt \
--client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
```

For additional details, see [Integrate Key HSM and Key Trustee Server](#).

7. Remove Configuration Files From Previous Installation

After completing the rollback, remove the saved configuration files from the previous installation:

```
cd /usr/share/keytrustee-server-keyhsm/
rm application.properties.bak
rm keystore.bak
rm truststore.bak
```

Rollback Navigator Encrypt

To roll back Cloudera Navigator Encrypt:

1. If you have configured and are using an RSA master key file with OAEP padding, then you must revert this setting to its original value:

```
navencrypt key --change
```

2. Stop the Navigator Encrypt mount service:

```
sudo /etc/init.d/navencrypt-mount stop
```

3. Confirm that the mount-stop command completed:

```
sudo /etc/init.d/navencrypt-mount status
```

4. If rolling back to a release lower than NavEncrypt 6.2:

- a. Print the existing ACL rules and save that output to a file:

```
sudo navencrypt acl --print+ vim acls.txt
```

- b. Delete all existing ACLs, for example, if there are a total of 7 ACL rules run:

```
sudo navencrypt acl --del --line=1,2,3,4,5,6,7
```

5. To fully downgrade Navigator Encrypt, manually downgrade all of the associated Navigator Encrypt packages (in the order listed):

- a. navencrypt
- b. (Only required for operating systems other than SLES) navencrypt-kernel-module
- c. (Only required for the SLES operating system) cloudera-navencryptfs-kmp-*<KERNEL_FLAVOR>*

Note: Replace *kernel_flavor* with the kernel flavor for your system. Navigator Encrypt supports the default, xen, and ec2 kernel flavors.

- d. libkeytrustee

6. If rolling back to a release less than NavEncrypt 6.2

- a. Reapply the ACL rules:

```
sudo navencrypt acl --add --file=acls.txt
```

7. Recompute process signatures:

```
sudo navencrypt acl --update
```

8. Restart the Navigator Encrypt mount service

```
sudo /etc/init.d/navencrypt-mount start
```

Rollback Solr

Due to on-disk format changes, Solr collections need to be restored from backup. Rollback of HDFS and ZooKeeper are NOT necessary when restoring individual collections.

For more information, see [Restoring a Collection](#) documentation.

Rollback Atlas

Rollback Atlas Solr Collections

Atlas has several collections in Solr that must be restored from the pre-upgrade backup - *vertex_index*, *edge_index*, and *fulltext_index*. These collections may already have been restored using the [Rollback Solr](#) documentation. If the collections are not yet restored, you must restore collections now using the [Rollback Solr](#) documentation.

Rollback Atlas HBase Tables

1. From a client host, start the hbase shell `hbase shell`
2. Within the hbase shell, list the snapshots, that must contain the pre-upgrade snapshots
`list_snapshots`
3. Within the hbase shell, disable the atlas_janus table, restore the snapshot, and enable the table
`disable 'atlas_janus' restore_snapshot '<name of atlas_janus snapshot from list_snapshots>' disable 'atlas_janus'`
4. Within the hbase shell, disable the ATLAS_ENTITY_AUDIT_EVENTS table, restore the snapshot, and enable the table
`disable 'ATLAS_ENTITY_AUDIT_EVENTS' restore_snapshot '<name of ATLAS_ENTITY_AUDIT_EVENTS snapshot from list_snapshots>' disable 'ATLAS_ENTITY_AUDIT_EVENTS'`
5. Restart Atlas.

Rollback YARN Queue Manager

You can rollback YARN Queue Manager using the pre-upgrade backup of `config-service.mv.db` and `config-service.trace.db`.

1. Navigate to the YARN Queue Manager service in Cloudera Manager and record the configuration value for `config_service_db_loc` (or `queuemanager_user_home_dir` if blank) and the host where the YARN Queue Manager Store is running.
2. Stop the YARN Queue Manager service.
3. SSH to the YARN Queue Manager Store host and copy the pre-upgrade `config-service.mv.db` and `config-service.trace.db` to the `config_service_db_loc` obtained in the previous step.
4. Start the YARN Queue Manager service.

Rollback Spark

You can rollback Spark in case of any failure during upgrading to Cloudera Base on premises 7.3.1.



Note: The rollback steps are the same for all CDP Private Cloud Base 7.1.x versions. The steps below describe rolling back to CDP Private Cloud Base 7.1.9.

1. Activate the CDP Private Cloud Base 7.1.9 parcel on your cluster in **Parcels Activate**.
2. Close the **Restart** window with **Close**.
3. The Spark 3 configuration issues have to be fixed after the parcel rollback:
 - a. Delete the `hdfs://` prefix in `spark.eventLog.dir`
 - b. Delete the `hdfs://` prefix in `spark.driver.log.dir`
4. Activate the SPARK3 parcel in **ClustersStatus** and in the **Actions** menu select **Restart**.
5. Restart the services in **ParcelsActivate**.
6. Deploy the Client Configuration. (See below.)
7. Perform a post-rollback check by running the pi job on a Spark 3 gateway host:

```
spark3-submit --master yarn --deploy-mode client --class org.apache.spark.examples.SparkPi /opt/cloudera/parcels/SPARK3/lib/spark3/examples/jars/spark-examples_2.12.jar 100
```

Deploy the Client Configuration

1. On the Cloudera Manager Home page, click the **Actions** menu and select **Deploy Client Configuration**.
2. Click **Deploy Client Configuration**.

Post downgrade steps

Streams Replication Manager (SRM)

Reset the state of the internal Kafka Streams application. Run the following command on the hosts of the SRM Service role.

```
kafka-streams-application-reset \
  --bootstrap-servers [***SRM SERVICE HOST***] \
  --config-file [***PROPERTIES FILE***] \
  --application-id srm-service_v2
```

Replace `[***PROPERTIES FILE***]` with the location of a configuration file that contains all necessary security properties that are required to establish a connection with the Kafka service. This option is only required if your Kafka service is secured.

Cruise Control

1. Insert the removed goal back to the relevant goal sets, but with the renamed goal name RackAwareGoal (Not RackAwareDistributionGoal)
2. Restart Cruise Control

Finalize the HDFS Upgrade

This step should be performed only after all validation is completed. For more information, see [Finalize the HDFS upgrade](#) documentation.

Knox

Following the downgrade to 7.1.7 SP2, Knox may be found in an unhealthy state. In this case, perform a Rolling Restart of the Knox service using the Cloudera Manager UI.

Rollback from Cloudera Base on premises 7.3.1

Perform the below procedure to rollback your cluster from Cloudera Base on premises 7.3.1 to CDP Private Cloud Base 7.1.9 SP1, 7.1.9, [CDP 7.1.8 cumulative hotfix 17](#), or 7.1.7 SP3.

Rollback

Rollback restores the software to the prior release and also restores the data and metadata to the pre-upgrade state. Service interruptions are expected as the cluster must be halted. After HDFS and/or Ozone is finalized, it is not possible to Rollback.

The following components do not support Rollback. If these components are in use, manual restore or recovery may be necessary. Work with Cloudera Support to devise a plan based on the deployed components.

- Ozone

Pre rollback steps

Ozone

This procedure is applicable only if you are downgrading from Cloudera Base on premises 7.3.1 to CDP Private Cloud Base 7.1.8.

1. Stop the Ozone Recon Web UI. Within Cloudera Manager UI, navigate to the Ozone service Ozone Recon Actions Stop this Ozone Recon .
2. Navigate to Configuration within the Ozone service and collect the value of ozone.recon.db.dir (default value is /var/lib/hadoop-ozone/recon/data).
3. SSH to the Ozone Recon Web UI host and move the ozone.recon.db.dir parent directory to a backup location: `mv /var/lib/hadoop-ozone/recon /var/lib/hadoop-ozone/recon-backup-CDP.`

HBase

Stop the HBase Master(s). Execute knit as the hbase user if Kerberos is enabled.

1. Stop Omid within Cloudera Manager UI

2. Navigate to the HBase service > Instances within Cloudera Manager UI and note the hostname of the HBase Master instance(s). Login to the host(s) and execute the following: `hbase master stop --shutDownCluster`
3. Stop the remaining HBase components. Navigate to the HBase service within Cloudera Manager UI Actions Stop

The following must be performed when downgrading to CDP Private Cloud Base 7.1.7 SP2 from Cloudera Base on premises 7.3.1. You will need to kinit as the hbase user if kerberos is enabled.

1. Contact support for the appropriate hbck2 jar
2. Execute a dry run of the shortenTableinfo command and validate the appropriate files have been identified `hbase --config /etc/hbase/conf hbck -j hbase-hbck2-X.Y.Z.jar shortenTableinfo`
3. Run the shortenTableinfo -fix command to fix the file format `hbase --config /etc/hbase/conf hbck -j hbase-hbck2-X.Y.Z.jar shortenTableinfo -fix`

Cruise Control

The following steps are applicable only if you downgrade from Cloudera Base on premises 7.3.1 to CDP Private Cloud Base 7.1.7 SP2 and not from Cloudera Base on premises 7.3.1 to CDP Private Cloud Base 7.1.8. You can skip this section if the Cruise Control Goal configurations were set to the default values before performing the upgrade. However, if the Cruise Control Goal configuration values were changed before performing the upgrade, then you must proceed with this section.

In Cruise Control, you must rename `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareDistributionGoal` to `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareGoal` in Cloudera Manager > Clusters > Cruise Control > Configurations tab in every occurrences as described below during downgrade process.

In Cruise Control, from CDP Private Cloud Base 7.1.8 and higher, `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareGoal` is renamed to `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareDistributionGoal`.

Perform the below steps:

1. Check the following goal sets if RackAwareDistributionGoal is present (Cloudera Manager > Clusters > Cruise Control > Configurations tab):
 - a. default.goals
 - b. goals
 - c. self.healing.goals
 - d. hard.goals
 - e. anomaly.detection.goals
2. Create a note for yourself about where RackAwareDistributionGoal were present
3. Remove RackAwareDistributionGoal from all of the goal lists
4. Perform the Cloudera Runtime downgrade process

Stop the Cluster

1. On the Home> Status tab, click the Actions menu and select Stop.
2. Click Stop on the confirmation screen. The Command Details window shows the progress of the stopping process.

When the All services successfully stopped message appears, the task is complete and you can close the Command Details window.

Rolling back the Cloudera Runtime parcel

1. Navigate to Parcels within Cloudera Manager.

2. Locate the **CDP Private Cloud Base 7.1.7 SP3/7.1.8/7.1.9 SP1** parcel and click Activate.
3. Follow the wizard and address any issues from the various inspectors.



Note: Warnings are expected for HDFS and Ozone due to the unfinalized rolling upgrade, validate no other warnings exist.

The upgrade activates the parcel and restarts services.

Restore CDH Databases

Restore the following databases from the CDH backups. Follow the order below and restore only a single service at a time. Stop the service prior to restoring to the database. Start the service after restoring the database.

- Ranger
- Ranger KMS
- Stream Messaging Manager
- Schema Registry
- Hue (only if you are rolling back to CDP Private Cloud Base 7.1.7 SP2)

The steps for backing up and restoring databases differ depending on the database vendor and version you select for your cluster and are beyond the scope of this document.



Important: Restore the databases to their exact state when you took the backup. Do not merge in any changes that may have occurred during the subsequent upgrade.

See the following vendor resources for more information:

- MariaDB 10.2, 10.3, 10.4 and 10.5: <https://mariadb.com/kb/en/backup-and-restore-overview/>
- MySQL 5.7: <https://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html>
- MySQL 8: <https://dev.mysql.com/doc/refman/8.0/en/backup-and-recovery.html>
- PostgreSQL 10: <https://www.postgresql.org/docs/10/backup.html>
- PostgreSQL 11: <https://www.postgresql.org/docs/11/backup.html>
- PostgreSQL 12: <https://www.postgresql.org/docs/12/backup.html>
- PostgreSQL 13: <https://www.postgresql.org/docs/13/backup.html>
- PostgreSQL 14: <https://www.postgresql.org/docs/14/backup.html>
- Oracle 19c: <https://docs.oracle.com/en/database/oracle/oracle-database/19/index.html>

Roll Back Cloudera Navigator Encryption Components

If you are rolling back any encryption components (Key Trustee KMS, HSM KMS, Key HSM, or Navigator Encrypt), first refer to:

- [HSM KMS High Availability Backup and Recovery](#)
- [Manually Backing Up Navigator Encrypt](#)

Start the Key Management Server

Restart the Key Management Server. Open the Cloudera Manager Admin Console, go to the KMS service page, and select Actions Start .

Roll Back Key HSM

To roll back Key HSM:

1. Install the version of Navigator Key HSM to which you wish to roll back

Install the Navigator Key HSM package using yum:

```
sudo yum downgrade keytrustee-keyhsm
```

Cloudera Navigator Key HSM is installed to the /usr/share/keytrustee-server-keyhsm directory by default.

2. Rename Previously-Created Configuration Files

For Key HSM major version rollbacks, previously-created configuration files do not authenticate with the HSM and Key Trustee Server, so you must recreate these files by re-executing the setup and trust commands. First, navigate to the Key HSM installation directory and rename the applications.properties, keystore, and truststore files:

```
cd /usr/share/keytrustee-server-keyhsm/  
mv application.properties application.properties.bak  
mv keystore keystore.bak  
mv truststore truststore.bak
```

3. Initialize Key HSM

Run the service keyhsm setup command in conjunction with the name of the target HSM distribution:

```
sudo service keyhsm setup [keysecure|thales|luna]
```

For more details, see [Initializing Navigator Key HSM](#).

4. Establish Trust Between Key HSM and the Key Trustee Server

The Key HSM service must explicitly trust the Key Trustee Server certificate (presented during TLS handshake). To establish this trust, run the following command:

```
sudo keyhsm trust /path/to/key_trustee_server/cert
```

For more details, see [Establish Trust from Key HSM to Key Trustee Server](#).

5. Start the Key HSM Service

Start the Key HSM service:

```
sudo service keyhsm start
```

6. Establish Trust Between Key Trustee Server and Key HSM

Establish trust between the Key Trustee Server and the Key HSM by specifying the path to the private key and certificate:

```
sudo ktadmin keyhsm --server https://keyhsm01.example.com:9090 \  
--client-certfile /etc/pki/cloudera/certs/mycert.crt \  
--client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
```

For a password-protected Key Trustee Server private key, add the --passphrase argument to the command (enter the password when prompted):

```
sudo ktadmin keyhsm --passphrase \  
--server https://keyhsm01.example.com:9090 \  
--client-certfile /etc/pki/cloudera/certs/mycert.crt \  
--client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
```

For additional details, see [Integrate Key HSM and Key Trustee Server](#).

7. Remove Configuration Files From Previous Installation

After completing the rollback, remove the saved configuration files from the previous installation:

```
cd /usr/share/keytrustee-server-keyhsm/
rm application.properties.bak
rm keystore.bak
rm truststore.bak
```

Roll Back Navigator Encrypt

To roll back Cloudera Navigator Encrypt:

1. If you have configured and are using an RSA master key file with OAEP padding, then you must revert this setting to its original value:

```
navencrypt key --change
```

2. Stop the Navigator Encrypt mount service:

```
sudo /etc/init.d/navencrypt-mount stop
```

3. Confirm that the mount-stop command completed:

```
sudo /etc/init.d/navencrypt-mount status
```

4. If rolling back to a release lower than NavEncrypt 6.2:

- a. Print the existing ACL rules and save that output to a file:

```
sudo navencrypt acl --print+ vim acls.txt
```

- b. Delete all existing ACLs, for example, if there are a total of 7 ACL rules run:

```
sudo navencrypt acl --del --line=1,2,3,4,5,6,7
```

5. To fully downgrade Navigator Encrypt, manually downgrade all of the associated Navigator Encrypt packages (in the order listed):

- a. navencrypt
- b. (Only required for operating systems other than SLES) navencrypt-kernel-module
- c. (Only required for the SLES operating system) cloudera-navencryptfs-kmp-*<KERNEL_FLAVOR>*

Note: Replace kernel_flavor with the kernel flavor for your system. Navigator Encrypt supports the default, xen, and ec2 kernel flavors.

6. If rolling back to a release less than NavEncrypt 6.2

- a. Reapply the ACL rules:

```
sudo navencrypt acl --add --file=acls.txt
```

7. Recompute process signatures:

```
sudo navencrypt acl --update
```

8. Restart the Navigator Encrypt mount service

```
sudo /etc/init.d/navencrypt-mount start
```

Rollback ZooKeeper

Rollback of the data and service is not expected to be necessary unless dependent services are being rolled back. If it is determined that a ZooKeeper rollback is necessary, the steps are:

1. Stop ZooKeeper
2. Restore the data backup. For example: `cp -rp /var/lib/zookeeper-backup-pre-upgrade-CM-CDH /var/lib/zookeeper/`
3. Start ZooKeeper

Rollback HDFS

You cannot roll back HDFS while high availability is enabled. The rollback procedure in this topic creates a temporary configuration without high availability. Regardless of whether high availability is enabled, follow the steps in this section.

1. Roll back all of the NameNodes. Use the [NameNode backup directory](#) you created before upgrading to Cloudera Base on premises. (/etc/hadoop/conf.rollback.namenode) to perform the following steps on all NameNode hosts:
 - a. Start the JournalNodes using Cloudera Manager:
 1. Go to the HDFS service.
 2. Select the Instances tab.
 3. Select all JournalNode roles from the list.
 4. Click Actions for Selected Start .
 - b. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file on all NameNode hosts (located in the temporary rollback directory) and update the keystore passwords with the actual cleartext passwords. The passwords will have values that look like this:

```
<property>
    <name>ssl.server.keystore.password</name>
    <value>*****</value>
</property>
<property>
    <name>ssl.server.keystore.keypassword</name>
    <value>*****</value>
</property>
```

- c. (TLS only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file and remove the `hadoop.security.credential.provider.path` property.
- d. (TLS only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file and change the value of `ssl.server.keystore.location` to `/etc/hadoop/conf.rollback.namenode/cm-auto-host_keystore.jks`
- e. Edit the /etc/hadoop/conf.rollback.namenode/core-site.xml and change the value of the `net.topology.script.file.name` property to `/etc/hadoop/conf.rollback.namenode`. For example:

```
# Original property
<property>
<name>net.topology.script.file.name</name>
<value>/var/run/cloudera-scm-agent/process/63-hdfs-NAMENODE/topology.py</value>
</property>
```

```
# New property
<property>
<name>net.topology.script.file.name</name>
<value>/etc/hadoop/conf.rollback.namenode/topology.py</value>
</property>
```

- f. Edit the `/etc/hadoop/conf.rollback.namenode/topology.py` file and change the value of `DATA_FILE_NAME` to `/etc/hadoop/conf.rollback.namenode`. For example:

```
DATA_FILE_NAME = '/etc/hadoop/conf.rollback.namenode/topology.map'
```

- g. (Kerberos enabled clusters only) Run the following command:

```
sudo -u hdfs kinit hdfs/<NameNode Host name> -l 7d -kt /etc/hadoop/conf.rollback.namenode/hdfs.keytab
```

- h. Start active Namenode with the following command:

```
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.namenode namenode -rollback
```

- i. Log in to the other Namenode and start SBNN with the following command:

```
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.namenode namenode -bootstrapStandby
```

- j. Press Yes when prompted. This also exits the process after it is done.

- k. Select Ctrl + C for active Namenode to exit the process.

2. Rollback the DataNodes.

Use the [DataNode rollback directory](#) you created before upgrading to Cloudera Base on premises (`/etc/hadoop/conf.rollback.datanode`) to perform the following steps on all DataNode hosts:

- a. (Clusters with TLS enabled only) Edit the `/etc/hadoop/conf.rollback.datanode/ssl-server.xml` file on all DataNode hosts (Located in the temporary rollback directory.) and update the keystore passwords (`ssl.server.keystore.password` and `ssl.server.keystore.keypassword`) with the actual passwords.

The passwords will have values that look like this:

```
<property>
  <name>ssl.server.keystore.password</name>
  <value>*****</value>
</property>
<property>
  <name>ssl.server.keystore.keypassword</name>
  <value>*****</value>
</property>
```

- b. (TLS only) Edit the `/etc/hadoop/conf.rollback.datanode/ssl-server.xml` file and remove the `hadoop.security.credential.provider.path` property and change the value of property.

```
ssl.server.keystore.location to /etc/hadoop/conf.rollback.datanode/cm-auto-host_keystore.jks
```

- c. Edit the `/etc/hadoop/conf.rollback.datanode/hdfs-site.xml` file and remove the `dfs.datanode.max.locked.memory` property.
- d. If you kerberos enabled cluster then make sure change the value of `hdfs.keytab` to the absolute path of `conf.rollback.datanode` folder in `core-site.xml` and `hdfs-site.xml`
- e. Run one of the following commands:

- If the DataNode is running with privileged ports (usually 1004 and 1006):

```
cd /etc/hadoop/conf.rollback.datanode
export HADOOP_SECURE_DN_USER=hdfs
export JSVC_HOME=/opt/cloudera/parcels/<PARCEL_FILENAME>/lib/bigtop-utils
hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
```

- If the DataNode is not running on privileged ports:

```
sudo hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
```

When the rolling back of the DataNodes is complete, terminate the console session by typing Control-C. Look for output from the command similar to the following that indicates when the DataNode rollback is complete:

```
Rollback of /dataroot/ycloud/dfs/dn/current/BP-<Block Group number> is complete
```

You may see the following error after issuing these commands:

```
ERROR datanode.DataNode: Exception in secureMain java.io.IOException:
    The path component: '/var/run/hdfs-sockets' in '/var/run/hdfs-sockets/dn' has permissions 0755 uid 39998 and gid 1006.
    It is not protected because it is owned by a user who is not root and not the effective user: '0'.
```

The error message will also include the following command to run:

```
chown root /var/run/hdfs-sockets
```

After running this command, the DataNode will restart successfully. Rerun the DataNode rollback command:

```
sudo hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
```

- f. If High Availability for HDFS is enabled, restart the HDFS service. In the Cloudera Manager Admin Console, go to the HDFS service and select Actions Restart .
- g. If high availability is not enabled for HDFS, use the Cloudera Manager Admin Console to restart all NameNodes and DataNodes.
 1. Go to the HDFS service.
 2. Select the Instances tab
 3. Select all DataNode and NameNode roles from the list.
 4. Click Actions for Selected Restart .
3. If high availability is not enabled for HDFS, roll back the Secondary NameNode.
 - a. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.secondarynamenode/ssl-server.xml file on all Secondary NameNode hosts (Located in the temporary rollback directory.) and update the keystore passwords with the actual cleartext passwords. The passwords will have values that look like this:

```
<property>
  <name>ssl.server.keystore.password</name>
  <value>*****</value>
</property>
<property>
  <name>ssl.server.keystore.keypassword</name>
  <value>*****</value>
</property>
```

- b. (TLS only) Edit the /etc/hadoop/conf.rollback.secondarynamenode/ssl-server.xml file and remove the hadoop.security.credential.provider.path property and change the value of property ssl.server.keystore.location to

```
/etc/hadoop/conf.rollback.secondarynamenode/cm-auto-host_keystore.jks
```

- c. Log in to the Secondary NameNode host and run the following commands:

```
rm -rf /dfs/snn/*
```



```
cd /etc/hadoop/conf.rollback.secondarynamenode/
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.secondary
namenode secondarynamenode -format
```

When the rolling back of the Secondary NameNode is complete, terminate the console session by typing Control-C. Look for output from the command similar to the following that indicates when the Secondary NameNode rollback is complete:

```
2020-12-21 17:09:36,239 INFO namenode.SecondaryNameNode: Web server init
done
```

4. Restart the HDFS service. Open the Cloudera Manager Admin Console, go to the HDFS service page, and select Actions Restart .

The Restart Command page displays the progress of the restart. Wait for the page to display the Successfully restarted service message before continuing.

For more information on HDFS, see [HDFS troubleshooting](#) documentation.

Start HBase

You might encounter other errors when starting HBase (for example, replication-related problems, region assignment-related issues, and meta region assignment problems). In this case, you must delete the znode in ZooKeeper and then start HBase again. (This deletes the replication peer information and you need to re-configure your replication schedules)

1. In Cloudera Manager, look up the value of the zookeeper.znode.parent property. The default value is /hbase.
2. Connect to the ZooKeeper ensemble by running the following command from any HBase gateway host.

```
zookeeper-client -server zookeeper_ensemble
```

To find the value to use for zookeeper_ensemble, open the /etc/hbase/conf.cloudera.<HBase service name>/hbase-site.xml file on any HBase gateway host. Use the value of the hbase.zookeeper.quorum property.



Note: If you have deployed a secure cluster, you must connect to ZooKeeper using a client jaas.conf file. You can find such a file in an HBase process directory (/var/run/cloudera-scm-agent/process/).

3. Specify the jaas.conf using the JVM flags by running the following commands in the ZooKeeper client.

```
CLIENT_JVMFLAGS=
"-Djava.security.auth.login.config=/var/run/cloudera-scm
agent/process/HBase_process_directory/jaas.conf"
zookeeper-client -server <zookeeper_ensemble>
```

The ZooKeeper command-line interface opens.

4. Enter the following command.

```
rmr /hbase
```

If you have deployed a secure cluster, enter the following command:

```
deleteall /hbase
```

If you see the message Node not empty: /hbase/tokenauth, you must re-run the same command and restart the HBase service.

5. Restart the HBase service.

After HBase is healthy, ensure that you restore the states of the Balancer and Normalizer (enable them if they were enabled before the rollback). Also re-enable the Merge and Split operations you disabled before the rollback to avoid the Master Procedure incompatibility problem.

Run the following commands in HBase Shell:

```
balance_switch true
normalizer_switch true
splitormerge_switch 'SPLIT', true
splitormerge_switch 'MERGE', true
```

Rollback Solr

1. Start the HDFS, ZooKeeper and Ranger services.
2. Restore the Solr-specific znodes in ZooKeeper using the snapshot you took before the upgrade.



Note: This step is only necessary if Solr is rolled back independent of other components. If ZooKeeper is rolled back as well, skip this step.

3. Start the Solr service.



Note: If the state of one or more Solr core is down and the Solr log contains a similar error message:

```
"org.apache.lucene.store.LockObtainFailedException: Lock obtain timed
out:
    org.apache.solr.store.hdfs.HdfsLockFactory"
```

it is necessary to clean up the HDFS locks in the index directories.

On all affected Solr nodes perform the following steps:

- a. Stop the Solr node using Cloudera Manager.
- b. Remove the HdfsDirectory@[***HEX ID***]-write.lock file from the index directory.

```
hdfs dfs -rm "/solr/[***COLLECTION NAME***]/[***CORE***]/data/
[***INDEX DIRECTORY NAME***]/HdfsDirectory@[***HEX ID***]
lockFactory=org.apache.solr.store.hdfs.HdfsLockFact
ory@[***HEX ID***]-write.lock"
```

For example:

```
hdfs dfs -rm "/solr/testCollection/core_node1/data/index/HdfsDir
ectory@5d07feac
lockFactory=org.apache.solr.store.hdfs.HdfsLockFacto
ry@7df08aad-write.lock"
```

- c. Start the Solr node using Cloudera Manager.
4. Restore the Solr collections using the backups you created before starting the upgrade. For more information, see [Restoring a Solr collection](#).



Note: Manually restoring collections on HDFS is only necessary if you roll back Solr independent of other components. If HDFS is rolled back as well, this is not necessary. You always need to manually restore collections that were stored on the local file system.

5. Restart Lily HBase Indexer (ks_indexer).

Rollback Atlas

Rollback Atlas Solr Collections

Atlas has several collections in Solr that must be restored from the pre-upgrade backup - vertex_index, edge_index, and fulltext_index. These collections may already have been restored

using the [Rollback Solr](#) documentation. If the collections are not yet restored, you must restore collections now using the [Rollback Solr](#) documentation.

Rollback Atlas HBase Tables

1. From a client host, start the HBase shell `hbase shell`
2. Within the HBase shell, list the snapshots, that must contain the pre-upgrade snapshots
`list_snapshots`
3. Within the HBase shell, disable the `atlas_janus` table, restore the snapshot, and enable the table

```
disable 'atlas_janus'
```

```
restore_snapshot '<name of atlas_janus snapshot from
list_snapshots>'
```

```
enable 'atlas_janus'
```

4. Within the HBase shell, disable the `ATLAS_ENTITY_AUDIT_EVENTS` table, restore the snapshot, and enable the table

```
disable 'ATLAS_ENTITY_AUDIT_EVENTS'
```

```
restore_snapshot '<name of ATLAS_ENTITY_AUDIT_EVENTS snapshot
from list_snapshots>'
```

```
enable 'ATLAS_ENTITY_AUDIT_EVENTS'
```

5. Restart Atlas.

Rollback Tez

After rolling back Tez, the `tez.lib.uris` property in `tez-site.xml` still points to libraries from a newer version. This causes issues with query processing.

To address this issue, you need to run "Upload Tez tar file to HDFS" from the **Tez** service's **Actions** menu. This updates the configuration and ensures `tez.lib.uris` points to the correct version.

1. Go to the **TEZ** service.
2. Select **Actions Upload Tez tar file to HDFS** and click **Upload Tez tar file to HDFS** to confirm.
3. Restart the **Hive on Tez** service to apply the updated configuration

Rollback Kudu

Rollback depends on which backup method was used. There are two forms of backup/restore in Kudu. Spark job to create or restore a full/incremental backup or backup the entire Kudu node and restore it later. Restoring the Kudu data differs between the Spark job and full node backup approaches.

See the [Kudu backup and restore](#) docs for data recovery steps. The OS level restore is not recommended if there is a backup.



Note: It is recommended to have the tables empty/deleted before restoring them as additional writes to the table after the upgrade would not be deleted after the restore.

Rollback YARN Queue Manager

You can rollback YARN Queue Manager using the pre-upgrade backup of `config-service.mv.db` and `config-service.trace.db`.

1. Navigate to the YARN Queue Manager service in Cloudera Manager and record the configuration value for `config_service_db_loc` (or `queuemanager_user_home_dir` if blank) and the host where the YARN Queue Manager Store is running.
2. Stop the YARN Queue Manager service.

3. SSH to the YARN Queue Manager Store host and copy the pre-upgrade config-service.mv.db and config-service.trace.db to the config_service_db_loc obtained in the previous step.
4. Start the YARN Queue Manager service.

Rollback Kafka

You can rollback Kafka as long as the following criteria are met.

- The log format Kafka properties `inter.broker.protocol.version` and `log.message.format.version` are set before the upgrade and are not removed or cleared after the upgrade.
- There is a backup with the pre-upgrade state of the cluster from where you can restore the data, such as a remote cluster replicated with SRM. For guidance on restoring data from an SRM-replicated cluster, contact Cloudera support.



Note: Neither rollbacks nor downgrades are possible after clearing log format Kafka properties.

Rollback Spark

You can rollback Spark in case of any failure during upgrading to Cloudera Base on premises 7.3.1.



Note: The rollback steps are the same for all CDP Private Cloud Base 7.1.x versions. The steps below describe rolling back to CDP Private Cloud Base 7.1.9.

1. Activate the CDP Private Cloud Base 7.1.9 parcel on your cluster in `Parcels Activate`.
2. Close the **Restart** window with `Close`.
3. The Spark 3 configuration issues have to be fixed after the parcel rollback:
 - a. Delete the `hdfs://` prefix in `spark.eventLog.dir`
 - b. Delete the `hdfs://` prefix in `spark.driver.log.dfsDir`
4. Activate the SPARK3 parcel in `ClustersStatus` and in the `Actions` menu select `Restart`.
5. Restart the services in `ParcelsActivate`.
6. Deploy the Client Configuration. (See below.)
7. Perform a post-rollback check by running the pi job on a Spark 3 gateway host:

```
spark3-submit --master yarn --deploy-mode client --class org.apache.spark.examples.SparkPi /opt/cloudera/parcels/SPARK3/lib/spark3/examples/jars/spark-examples_2.12.jar 100
```

Deploy the Client Configuration

1. On the Cloudera Manager Home page, click the `Actions` menu and select `Deploy Client Configuration`.
2. Click `Deploy Client Configuration`.

Restart the Cluster

You must restart the cluster using the following steps.

1. On the Cloudera Manager Home page, click the `Actions` menu and select `Restart`.
2. Click `Restart` which appears on the next screen to confirm. If you have enabled [high availability for HDFS](#), you can choose [Rolling Restart](#) instead to minimize cluster downtime. The `Command Details` window shows the progress of stopping services.

When `All services successfully started` appears, the task is complete and you can close the `Command Details` window.

Post rollback steps

Streams Replication Manager (SRM)

Reset the state of the internal Kafka Streams application. Run the following command on the hosts of the SRM Service role.

```
kafka-streams-application-reset \
  --bootstrap-servers [***SRM SERVICE HOST***] \
  --config-file [***PROPERTIES FILE***] \
  --application-id srm-service_v2
```

Replace [***PROPERTIES FILE***] with the location of a configuration file that contains all necessary security properties that are required to establish a connection with the Kafka service. This option is only required if your Kafka service is secured.

Cruise Control

1. Insert the removed goal back to the relevant goal sets, but with the renamed goal name RackAwareGoal (Not RackAwareDistributionGoal)
2. Restart Cruise Control

Oozie

Execute the Install Oozie ShareLib action through Cloudera Manager:

1. Go to the Oozie service.
2. Select Actions Install Oozie ShareLib .

Finalize the HDFS Upgrade

This step should be performed only after all validation is completed. For more information, see [Finalize the HDFS upgrade](#) documentation.

Runtime upgrade



Note: You must ensure that HDFS is not in Safe mode.

When the parcel is activated, click the Actions menu next to the cluster name and select Post Cloudera Runtime Upgrade.

Downgrade or Rollback from CDP Private Cloud Base 7.1.9

You can rollback an upgrade from CDP Private Cloud Base 7.1.9 to CDP Private Cloud Base 7.1.8 or 7.1.7 SP3 or 7.1.7 SP2 or 7.1.7 SP1. You can downgrade an upgrade from CDP Private Cloud Base 7.1.9 to CDP Private Cloud Base 7.1.8 or 7.1.7 SP3 or 7.1.7 SP2. Before you downgrade or rollback, make sure that you inspect each supported component to ensure that your intended process works in a seamless manner.

This section helps you understand and perform the downgrade or rollback process. In case you cannot upgrade your cluster due to an incompatible change, catastrophic failure, or accidental upgrade, you can use the below manual procedure to rollback or downgrade your cluster. In most cases, resolving the upgrade issue and continuing the upgrade is the preferred path. Performing a downgrade or rollback is an advanced operation. Not all components support downgrade and rollback. Contact Cloudera Support for guidance on performing a downgrade or rollback.

Downgrade - Downgrade restores the software to the prior release version but data remains intact. Service Interruptions may or may not be required for downgrade depending on the component.

Rollback - Rollback the state of the data of the upgraded components to the pre-upgrade state. A Rollback may require a service Interruption and is not an automated operation.



Attention: When you are rolling back from Cloudera Runtime 7.1.9 version to Cloudera Runtime 7.1.8 or lower versions, follow these instructions:

- If you are using RHEL 7, there is no requirement to reinstall Python 2.7 version.
- If you are using RHEL 8, SLES 15 SP4, and Ubuntu 20 operating systems, and in case you uninstalled Python 2.7 version during your upgrade process, you must reinstall Python 2.7 first and then proceed towards rollback process.

Review Limitations

The rollback procedure has the following limitations:

- HDFS – If you have finalized the HDFS upgrade, you cannot roll back your cluster.
- Compute clusters – Rollback for Compute clusters is not currently supported.
- Configuration changes, including the addition of new services or roles after the upgrade, are not retained after rolling back Cloudera Manager.



Note: Cloudera recommends that you not make configuration changes or add new services and roles until you have finalized the HDFS upgrade and no longer require the option to roll back your upgrade.

- HBase – If your cluster is configured to use HBase replication, data written to HBase after the upgrade might not be replicated to peers when you start your rollback. This topic does not describe how to determine which, if any, peers have the replicated data and how to roll back that data. For more information about HBase replication, see [HBase Replication](#).
- Kafka – Once the Kafka log format and protocol version configurations (the `inter.broker.protocol.version` and `log.message.format.version` properties) are set to the new version (or left blank, which means to use the latest version), Kafka rollback is not possible.

Downgrade from CDP Private Cloud Base 7.1.9

Perform the below procedure to downgrade your cluster from CDP Private Cloud Base 7.1.9 to CDP Private Cloud Base 7.1.8 or 7.1.7 SP3 or 7.1.7 SP2.

Downgrade

Downgrade restores the software to the prior release version but data remains intact. Service interruptions are still expected, but the use of Rolling Restart minimizes it. After HDFS and/or Ozone is finalized, it is not possible to Downgrade or Rollback.

The following components do not support downgrade. If these components are in use, rollback may be necessary. Work with Cloudera Support to devise a plan based on the deployed components.

- Ranger
- Ranger RMS
- Ranger KMS
- KTS
- YARN Queue Manager
- Schema Registry
- Solr
- Atlas

Pre-downgrade steps

Ozone

This procedure is applicable only if you are downgrading from CDP Private Cloud Base 7.1.9 to CDP Private Cloud Base 7.1.8.

1. Stop the Ozone Recon Web UI. Within Cloudera Manager UI, navigate to the Ozone service Ozone Recon Actions Stop this Ozone Recon .

2. Navigate to Configuration within the Ozone service and collect the value of `ozone.recon.db.dir` (default value is `/var/lib/hadoop-ozone/recon/data`).
3. SSH to the Ozone Recon Web UI host and move the `ozone.recon.db.dir` parent directory to a backup location: `mv /var/lib/hadoop-ozone/recon /var/lib/hadoop-ozone/recon-backup-CDP`.

HBase

Stop the HBase Master(s). Execute `kinit` as the `hbase` user if `kerberos` is enabled.

1. Stop Omid within Cloudera Manager UI
2. Navigate to the HBase service > Instances within Cloudera Manager UI and note the hostname of the HBase Master instance(s). Login to the host(s) and execute the following: `hbase master stop --shutDownCluster`
3. Stop the remaining HBase components. Navigate to the HBase service within Cloudera Manager UI Actions Stop

The following must be performed when downgrading to CDP Private Cloud Base 7.1.7 SP2 from CDP Private Cloud Base 7.1.9. You will need to `kinit` as the `hbase` user if `kerberos` is enabled.

1. Contact support for the appropriate `hbck2` jar
2. Execute a dry run of the `shortenTableinfo` command and validate the appropriate files have been identified `hbase --config /etc/hbase/conf hbck -j hbase-hbck2-X.Y.Z.jar shortenTableinfo`
3. Run the `shortenTableinfo -fix` command to fix the file format `hbase --config /etc/hbase/conf hbck -j hbase-hbck2-X.Y.Z.jar shortenTableinfo -fix`

Cruise Control

The following steps are applicable only if you downgrade from CDP Private Cloud Base 7.1.9 to CDP Private Cloud Base 7.1.7 SP2 and not from CDP Private Cloud Base 7.1.9 to CDP Private Cloud Base 7.1.8. You can skip this section if the Cruise Control Goal configurations were set to the default values before performing the upgrade. However, if the Cruise Control Goal configuration values were changed before performing the upgrade, then you must proceed with this section.

In Cruise Control, you must rename `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareDistributionGoal` to `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareGoal` in Cloudera Manager > Clusters > Cruise Control > Configurations tab in every occurrences as described below during downgrade process.

In Cruise Control, from CDP Private Cloud Base 7.1.8 and higher, `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareGoal` is renamed to `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareDistributionGoal`.

Perform the below steps:

1. Check the following goal sets if `RackAwareDistributionGoal` is present (Cloudera Manager > Clusters > Cruise Control > Configurations tab):
 - a. `default.goals`
 - b. `goals`
 - c. `self.healing.goals`
 - d. `hard.goals`
 - e. `anomaly.detection.goals`
2. Create a note for yourself about where `RackAwareDistributionGoal` were present
3. Remove `RackAwareDistributionGoal` from all of the goal lists
4. Perform the Cloudera Runtime downgrade process

Downgrading the Cloudera Runtime parcel

1. Navigate to Parcels within Cloudera Manager.
2. Locate the **CDP Private Cloud Base 7.1.7 SP2/7.1.8** parcel and click Upgrade.
3. Follow the wizard and address any issues from the various inspectors.



Note: Warnings are expected for HDFS and Ozone due to the unfinalized rolling upgrade, validate no other warnings exist.

The upgrade activates the parcel and restarts services.

Restart Services with Stale Configuration

1. Navigate to Clusters and find the cluster being downgraded.
2. Inspect the list of services for the stale configuration indicator (yellow power button), if found, click on the indicator.
3. Follow the wizard to restart services with stale configuration.

Restore CDH Databases

Restore the following databases from the CDH backups. Follow the order below and restore only a single service at a time. Stop the service prior to restoring to the database. Start the service after restoring the database.

- Ranger
- Ranger KMS
- Stream Messaging Manager
- Schema Registry
- Hue (only if you are downgrading to 7.1.7 SP2)

The steps for backing up and restoring databases differ depending on the database vendor and version you select for your cluster and are beyond the scope of this document.



Important: Restore the databases to their exact state as of when you took the backup. Do not merge in any changes that may have occurred during the subsequent upgrade.

See the following vendor resources for more information:

- MariaDB 10.2, 10.3, 10.4 and 10.5: <https://mariadb.com/kb/en/backup-and-restore-overview/>
- MySQL 5.7: <https://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html>
- MySQL 8: <https://dev.mysql.com/doc/refman/8.0/en/backup-and-recovery.html>
- PostgreSQL 10: <https://www.postgresql.org/docs/10/backup.html>
- PostgreSQL 11: <https://www.postgresql.org/docs/11/backup.html>
- PostgreSQL 12: <https://www.postgresql.org/docs/12/backup.html>
- PostgreSQL 13: <https://www.postgresql.org/docs/13/backup.html>
- PostgreSQL 14: <https://www.postgresql.org/docs/14/backup.html>
- Oracle 19c: <https://docs.oracle.com/en/database/oracle/oracle-database/19/index.html>

Rollback Cloudera Navigator Encryption Components

If you are rolling back any encryption components (Key Trustee Server, Key Trustee KMS, HSM KMS, Key HSM, or Navigator Encrypt), first refer to:

- [Backing up and Restoring Key Trustee Server and Clients](#)
- [HSM KMS High Availability Backup and Recovery](#)
- [Manually Backing Up Navigator Encrypt](#)

Rollback Key Trustee Server



Note: If rolling back multiple encryption product components, it is recommended that you begin with the Key Trustee Server.

To roll back Key Trustee Server, replace the currently used parcel (for example, the parcel for version CDP Private Cloud Base 7.1.9) with the parcel for the version to which you wish to roll back (for example, version CDP Private Cloud Base 7.1.8). See [Parcels](#) for detailed instructions on using parcels.

The Keytrustee Server CDP Private Cloud Base 7.1.9 upgrades the bundled Postgres engine from version 12.1 to 14.2. The upgrade happens automatically, however, downgrading to CDP Private Cloud Base 7.1.8, requires manual steps to roll back the database engine to version 12.1. Because the previously upgraded database is left unchanged, the database server will fail to start. Follow these steps to recreate the Postgres 12.1 compatible database:

1. Open the Cloudera Manager Admin Console and go to the Key Trustee Server service. If you see that Key Trustee Server has stale configurations, click the yellow or blue button and follow the prompts.
2. Make sure that the Keytrustee Server database roles are stopped. Then rename the folder containing Keytrustee Postgres database data (both on master and slave hosts):

```
mv /var/lib/keytrustee/db /var/lib/keytrustee/db-14_2
```

3. Open the Cloudera Manager Admin Console and go to the Key Trustee Server service.
4. Select the Instances tab.
5. Select the Active Database role type.
6. Click Actions for Selected Set Up the Key Trustee Server Database .
7. Click Set Up the Key Trustee Server Database to confirm.

Cloudera Manager sets up the Key Trustee Server database.

8. Start the PostgreSQL server:

```
sudo ktadmin db --start --pg-rootdir /var/lib/keytrustee/db --background
```

9. On the master KTS node: running as user keytrustee, restore the keytrustee database on active hosts by running the following commands:

```
sudo -su keytrustee
    export PATH=$PATH:/opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/12.1/bin/
    export LD_LIBRARY_PATH=/opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/12.1/lib/
    unzip -p keytrustee-db.zip | psql -p 11381 -d keytrustee
```

If the zip file is encrypted, you are prompted for the password to decrypt the file.

10. Restore the Key Trustee Server configuration directory, on Active hosts:

```
su - keytrustee
    cd /var/lib/keytrustee
    unzip keytrustee-conf.zip
```

If the zip file is encrypted, you are prompted for the password to decrypt the file.

11. Stop the PostgreSQL server : Change the login user to root and run the command:

```
sudo ktadmin db --stop --pg-rootdir /var/lib/keytrustee/db
```

12. Remove the backup files (keytrustee-db.zip and keytrustee-conf.zip) from the Key Trustee Server's host.

```
su - keytrustee
    cd /var/lib/keytrustee
    rm keytrustee-conf.zip
    rm keytrustee-db.zip
```

13. Start the Active Database role in Cloudera Manager by clicking **Actions for Selected Start**.
14. Click **Start** to confirm.
15. Select the Active Database.
16. Click **Actions for Selected Setup Enable Synchronous Replication in HA mode**.
17. Start the Passive Database instance: select the Passive Database, click **Actions for Selected Start**.
18. In the Cloudera Manager Admin Console, start the active KTS instance.
19. In the Cloudera Manager Admin Console, start the passive KTS instance.

Start the Key Management Server

Restart the Key Management Server. Open the Cloudera Manager Admin Console, go to the KMS service page, and select **Actions Start**.

Rollback Key HSM

To roll back Key HSM:

1. Install the version of Navigator Key HSM to which you wish to roll back

Install the Navigator Key HSM package using yum:

```
sudo yum downgrade keytrustee-keyhsm
```

Cloudera Navigator Key HSM is installed to the `/usr/share/keytrustee-server-keyhsm` directory by default.

2. Rename Previously-Created Configuration Files

For Key HSM major version rollbacks, previously-created configuration files do not authenticate with the HSM and Key Trustee Server, so you must recreate these files by re-executing the setup and trust commands. First, navigate to the Key HSM installation directory and rename the `applications.properties`, `keystore`, and `truststore` files:

```
cd /usr/share/keytrustee-server-keyhsm/  
mv application.properties application.properties.bak  
mv keystore keystore.bak  
mv truststore truststore.bak
```

3. Initialize Key HSM

Run the service `keyhsm setup` command in conjunction with the name of the target HSM distribution:

```
sudo service keyhsm setup [keysecure|thales|luna]
```

For more details, see [Initializing Navigator Key HSM](#).

4. Establish Trust Between Key HSM and the Key Trustee Server

The Key HSM service must explicitly trust the Key Trustee Server certificate (presented during TLS handshake). To establish this trust, run the following command:

```
sudo keyhsm trust /path/to/key_trustee_server/cert
```

For more details, see [Establish Trust from Key HSM to Key Trustee Server](#).

5. Start the Key HSM Service

Start the Key HSM service:

```
sudo service keyhsm start
```

6. Establish Trust Between Key Trustee Server and Key HSM

Establish trust between the Key Trustee Server and the Key HSM by specifying the path to the private key and certificate:

```
sudo ktadmin keyhsm --server https://keyhsm01.example.com:9090 \
  --client-certfile /etc/pki/cloudera/certs/mycert.crt \
  --client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
```

For a password-protected Key Trustee Server private key, add the `--passphrase` argument to the command (enter the password when prompted):

```
sudo ktadmin keyhsm --passphrase \
  --server https://keyhsm01.example.com:9090 \
  --client-certfile /etc/pki/cloudera/certs/mycert.crt \
  --client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
```

For additional details, see [Integrate Key HSM and Key Trustee Server](#).

7. Remove Configuration Files From Previous Installation

After completing the rollback, remove the saved configuration files from the previous installation:

```
cd /usr/share/keytrustee-server-keyhsm/
rm application.properties.bak
rm keystore.bak
rm truststore.bak
```

Rollback Navigator Encrypt

To roll back Cloudera Navigator Encrypt:

1. If you have configured and are using an RSA master key file with OAEP padding, then you must revert this setting to its original value:

```
navencrypt key --change
```

2. Stop the Navigator Encrypt mount service:

```
sudo /etc/init.d/navencrypt-mount stop
```

3. Confirm that the mount-stop command completed:

```
sudo /etc/init.d/navencrypt-mount status
```

4. If rolling back to a release lower than NavEncrypt 6.2:

- a. Print the existing ACL rules and save that output to a file:

```
sudo navencrypt acl --print+ vim acls.txt
```

- b. Delete all existing ACLs, for example, if there are a total of 7 ACL rules run:

```
sudo navencrypt acl --del --line=1,2,3,4,5,6,7
```

5. To fully downgrade Navigator Encrypt, manually downgrade all of the associated Navigator Encrypt packages (in the order listed):
 - a. navencrypt
 - b. (Only required for operating systems other than SLES) navencrypt-kernel-module
 - c. (Only required for the SLES operating system) cloudera-navencryptfs-kmp-*<KERNEL_FLAVOR>*

Note: Replace *kernel_flavor* with the kernel flavor for your system. Navigator Encrypt supports the default, xen, and ec2 kernel flavors.

6. If rolling back to a release less than NavEncrypt 6.2
 - a. Reapply the ACL rules:

```
sudo navencrypt acl --add --file=acls.txt
```

7. Recompute process signatures:

```
sudo navencrypt acl --update
```

8. Restart the Navigator Encrypt mount service

```
sudo /etc/init.d/navencrypt-mount start
```

Rollback Solr

Due to on-disk format changes, Solr collections need to be restored from backup. Rollback of HDFS and ZooKeeper are NOT necessary when restoring individual collections.

For more information, see [Restoring a Collection](#) documentation.

Rollback Atlas

Rollback Atlas Solr Collections

Atlas has several collections in Solr that must be restored from the pre-upgrade backup - *vertex_index*, *edge_index*, and *fulltext_index*. These collections may already have been restored using the [Rollback Solr](#) documentation. If the collections are not yet restored, you must restore collections now using the [Rollback Solr](#) documentation.

Rollback Atlas HBase Tables

1. From a client host, start the hbase shell `hbase shell`
2. Within the hbase shell, list the snapshots, that must contain the pre-upgrade snapshots
`list_snapshots`
3. Within the hbase shell, disable the *atlas_janus* table, restore the snapshot, and enable the table
`disable 'atlas_janus' restore_snapshot '<name of atlas_janus snapshot from list_snapshots>' disable 'atlas_janus'`
4. Within the hbase shell, disable the *ATLAS_ENTITY_AUDIT_EVENTS* table, restore the snapshot, and enable the table
`disable 'ATLAS_ENTITY_AUDIT_EVENTS' restore_snapshot '<name of ATLAS_ENTITY_AUDIT_EVENTS snapshot from list_snapshots>' disable 'ATLAS_ENTITY_AUDIT_EVENTS'`
5. Restart Atlas.

Rollback YARN Queue Manager

You can rollback YARN Queue Manager using the pre-upgrade backup of *config-service.mv.db* and *config-service.trace.db*.

1. Navigate to the YARN Queue Manager service in Cloudera Manager and record the configuration value for *config_service_db_loc* (or *queuemanager_user_home_dir* if blank) and the host where the YARN Queue Manager Store is running.

2. Stop the YARN Queue Manager service.
3. SSH to the YARN Queue Manager Store host and copy the pre-upgrade config-service.mv.db and config-service.trace.db to the config_service_db_loc obtained in the previous step.
4. Start the YARN Queue Manager service.

Deploy the Client Configuration

1. On the Cloudera Manager Home page, click the Actions menu and select Deploy Client Configuration.
2. Click Deploy Client Configuration.

Post downgrade steps

Streams Replication Manager (SRM)

Reset the state of the internal Kafka Streams application. Run the following command on the hosts of the SRM Service role.

```
kafka-streams-application-reset \
  --bootstrap-servers [***SRM SERVICE HOST***] \
  --config-file [***PROPERTIES FILE***] \
  --application-id srm-service_v2
```

Replace `[***PROPERTIES FILE***]` with the location of a configuration file that contains all necessary security properties that are required to establish a connection with the Kafka service. This option is only required if your Kafka service is secured.

Cruise Control

1. Insert the removed goal back to the relevant goal sets, but with the renamed goal name RackAwareGoal (Not RackAwareDistributionGoal)
2. Restart Cruise Control

Finalize the HDFS Upgrade

This step should be performed only after all validation is completed. For more information, see [Finalize the HDFS upgrade](#) documentation.

Knox

Following the downgrade to CDP Private Cloud Base 7.1.7 SP2, Knox may be found in an unhealthy state. In this case, perform a Rolling Restart of the Knox service using the Cloudera Manager UI.

Rollback from CDP Private Cloud Base 7.1.9

Perform the below procedure to rollback your cluster from CDP Private Cloud Base 7.1.9 to CDP Private Cloud Base 7.1.8 or 7.1.7 SP3 or 7.1.7 SP2 or 7.1.7 SP1.

Rollback

Rollback restores the software to the prior release and also restores the data and metadata to the pre-upgrade state. Service interruptions are expected as the cluster must be halted. After HDFS and/or Ozone is finalized, it is not possible to Rollback.

The following components do not support Rollback. If these components are in use, manual restore or recovery may be necessary. Work with Cloudera Support to devise a plan based on the deployed components.

- Ozone

Pre rollback steps

Ozone

This procedure is applicable only if you are downgrading from CDP Private Cloud Base 7.1.9 to CDP Private Cloud Base 7.1.8.

1. Stop the Ozone Recon Web UI. Within Cloudera Manager UI, navigate to the Ozone service Ozone Recon Actions Stop this Ozone Recon .
2. Navigate to Configuration within the Ozone service and collect the value of ozone.recon.db.dir (default value is /var/lib/hadoop-ozone/recon/data).
3. SSH to the Ozone Recon Web UI host and move the ozone.recon.db.dir parent directory to a backup location: `mv /var/lib/hadoop-ozone/recon /var/lib/hadoop-ozone/recon-backup-CDP.`

HBase

Stop the HBase Master(s). Execute kinit as the hbase user if kerberos is enabled.

1. Stop Omid within Cloudera Manager UI
2. Navigate to the HBase service > Instances within Cloudera Manager UI and note the hostname of the HBase Master instance(s). Login to the host(s) and execute the following: `hbase master stop --shutDownCluster`
3. Stop the remaining HBase components. Navigate to the HBase service within Cloudera Manager UI Actions Stop

The following must be performed when downgrading to CDP Private Cloud Base 7.1.7 SP2 from CDP Private Cloud Base 7.1.9. You will need to kinit as the hbase user if kerberos is enabled.

1. Contact support for the appropriate hbck2 jar
2. Execute a dry run of the shortenTableinfo command and validate the appropriate files have been identified `hbase --config /etc/hbase/conf hbck -j hbase-hbck2-X.Y.Z.jar shortenTableinfo`
3. Run the shortenTableinfo -fix command to fix the file format `hbase --config /etc/hbase/conf hbck -j hbase-hbck2-X.Y.Z.jar shortenTableinfo -fix`

Cruise Control

The following steps are applicable only if you downgrade from CDP Private Cloud Base 7.1.9 to CDP Private Cloud Base 7.1.7 SP2 and not from CDP Private Cloud Base 7.1.9 to CDP Private Cloud Base 7.1.8. You can skip this section if the Cruise Control Goal configurations were set to the default values before performing the upgrade. However, if the Cruise Control Goal configuration values were changed before performing the upgrade, then you must proceed with this section.

In Cruise Control, you must rename `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareDistributionGoal` to `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareGoal` in Cloudera Manager > Clusters > Cruise Control > Configurations tab in every occurrences as described below during downgrade process.

In Cruise Control, from CDP Private Cloud Base 7.1.8 and higher, `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareGoal` is renamed to `com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareDistributionGoal`.

Perform the below steps:

1. Check the following goal sets if RackAwareDistributionGoal is present (Cloudera Manager > Clusters > Cruise Control > Configurations tab):
 - a. default.goals
 - b. goals
 - c. self.healing.goals
 - d. hard.goals
 - e. anomaly.detection.goals
2. Create a note for yourself about where RackAwareDistributionGoal were present
3. Remove RackAwareDistributionGoal from all of the goal lists

4. Perform the Cloudera Runtime downgrade process

Stop the Cluster

1. On the Home> Status tab, click the Actions menu and select Stop.
2. Click Stop in the confirmation screen. The Command Details window shows the progress of the stopping process.

When the All services successfully stopped message appears, the task is complete and you can close the Command Details window.

Rolling back the Cloudera Runtime parcel

1. Navigate to Parcels within Cloudera Manager.
2. Locate the **CDP Private Cloud Base 7.1.7 SP2/7.1.8** parcel and click Upgrade.
3. Follow the wizard and address any issues from the various inspectors.



Note: Warnings are expected for HDFS and Ozone due to the unfinalized rolling upgrade, validate no other warnings exist.

The upgrade activates the parcel and restarts services.

Restore CDH Databases

Restore the following databases from the CDH backups. Follow the order below and restore only a single service at a time. Stop the service prior to restoring to the database. Start the service after restoring the database.

- Ranger
- Ranger KMS
- Stream Messaging Manager
- Schema Registry
- Hue (only if you are rolling back to CDP Private Cloud Base 7.1.7 SP2)

The steps for backing up and restoring databases differ depending on the database vendor and version you select for your cluster and are beyond the scope of this document.



Important: Restore the databases to their exact state as of when you took the backup. Do not merge in any changes that may have occurred during the subsequent upgrade.

See the following vendor resources for more information:

- MariaDB 10.2, 10.3, 10.4 and 10.5: <https://mariadb.com/kb/en/backup-and-restore-overview/>
- MySQL 5.7: <https://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html>
- MySQL 8: <https://dev.mysql.com/doc/refman/8.0/en/backup-and-recovery.html>
- PostgreSQL 10: <https://www.postgresql.org/docs/10/backup.html>
- PostgreSQL 11: <https://www.postgresql.org/docs/11/backup.html>
- PostgreSQL 12: <https://www.postgresql.org/docs/12/backup.html>
- PostgreSQL 13: <https://www.postgresql.org/docs/13/backup.html>
- PostgreSQL 14: <https://www.postgresql.org/docs/14/backup.html>
- Oracle 19c: <https://docs.oracle.com/en/database/oracle/oracle-database/19/index.html>

Roll Back Cloudera Navigator Encryption Components

If you are rolling back any encryption components (Key Trustee Server, Key Trustee KMS, HSM KMS, Key HSM, or Navigator Encryption), first refer to:

- [Backing up and Restoring Key Trustee Server and Clients](#)
- [HSM KMS High Availability Backup and Recovery](#)
- [Manually Backing Up Navigator Encrypt](#)

Roll Back Key Trustee Server



Note: If rolling back multiple encryption product components, it is recommended that you begin with the Key Trustee Server.

To roll back Key Trustee Server, replace the currently used parcel (for example, the parcel for version CDP Private Cloud Base 7.1.9) with the parcel for the version to which you wish to roll back (for example, version CDP Private Cloud Base 7.1.8). See [Parcels](#) for detailed instructions on using parcels.

The Keytrustee Server CDP Private Cloud Base 7.1.9 upgrades the bundled Postgres engine from version 12.1 to 14.2. The upgrade happens automatically, however, downgrading to CDP Private Cloud Base 7.1.8, requires manual steps to roll back the database engine to version 12.1. Because the previously upgraded database is left unchanged, the database server will fail to start. Follow these steps to recreate the Postgres 12.1 compatible database:

1. Open the Cloudera Manager Admin Console and go to the Key Trustee Server service. If you see that Key Trustee Server has stale configurations, click the yellow or blue button and follow the prompts.
2. Make sure that the Keytrustee Server database roles are stopped. Then rename the folder containing Keytrustee Postgres database data (both on master and slave hosts):

```
mv /var/lib/keytrustee/db /var/lib/keytrustee/db-14_2
```

3. Open the Cloudera Manager Admin Console and go to the Key Trustee Server service.
4. Select the Instances tab.
5. Select the Active Database role type.
6. Click Actions for Selected Set Up the Key Trustee Server Database .
7. Click Set Up the Key Trustee Server Database to confirm.

Cloudera Manager sets up the Key Trustee Server database.

8. Start the PostgreSQL server:

```
sudo ktadmin db --start --pg-rootdir /var/lib/keytrustee/db --background
```

9. On the master KTS node: running as user keytrustee, restore the keytrustee database on active hosts by running the following commands:

```
sudo -su keytrustee
export PATH=$PATH:/opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/12.1/bin/
export LD_LIBRARY_PATH=/opt/cloudera/parcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/12.1/lib/
unzip -p keytrustee-db.zip | psql -p 11381 -d keytrustee
```

If the zip file is encrypted, you are prompted for the password to decrypt the file.

10. Restore the Key Trustee Server configuration directory, on Active hosts:

```
su - keytrustee
cd /var/lib/keytrustee
unzip keytrustee-conf.zip
```

If the zip file is encrypted, you are prompted for the password to decrypt the file.

11. Stop the PostgreSQL server : Change the login user to root and run the command:

```
sudo ktadmin db --stop --pg-rootdir /var/lib/keytrustee/db
```

12. Remove the backup files (keytrustee-db.zip and keytrustee-conf.zip) from the Key Trustee Server's host.

```
su - keytrustee
cd /var/lib/keytrustee
rm keytrustee-conf.zip
rm keytrustee-db.zip
```


13. Start the Active Database role in Cloudera Manager by clicking **Actions for Selected Start**.
14. Click **Start** to confirm.
15. Select the Active Database.
16. Click **Actions for Selected Setup Enable Synchronous Replication in HA mode**.
17. Start the Passive Database instance: select the Passive Database, click **Actions for Selected Start**.
18. In the Cloudera Manager Admin Console, start the active KTS instance.
19. In the Cloudera Manager Admin Console, start the passive KTS instance.

Start the Key Management Server

Restart the Key Management Server. Open the Cloudera Manager Admin Console, go to the KMS service page, and select **Actions Start**.

Roll Back Key HSM

To roll back Key HSM:

1. Install the version of Navigator Key HSM to which you wish to roll back

Install the Navigator Key HSM package using yum:

```
sudo yum downgrade keytrustee-keyhsm
```

Cloudera Navigator Key HSM is installed to the `/usr/share/keytrustee-server-keyhsm` directory by default.

2. Rename Previously-Created Configuration Files

For Key HSM major version rollbacks, previously-created configuration files do not authenticate with the HSM and Key Trustee Server, so you must recreate these files by re-executing the setup and trust commands. First, navigate to the Key HSM installation directory and rename the `applications.properties`, `keystore`, and `truststore` files:

```
cd /usr/share/keytrustee-server-keyhsm/  
mv application.properties application.properties.bak  
mv keystore keystore.bak  
mv truststore truststore.bak
```

3. Initialize Key HSM

Run the service `keyhsm setup` command in conjunction with the name of the target HSM distribution:

```
sudo service keyhsm setup [keysecure|thales|luna]
```

For more details, see [Initializing Navigator Key HSM](#).

4. Establish Trust Between Key HSM and the Key Trustee Server

The Key HSM service must explicitly trust the Key Trustee Server certificate (presented during TLS handshake). To establish this trust, run the following command:

```
sudo keyhsm trust /path/to/key_trustee_server/cert
```

For more details, see [Establish Trust from Key HSM to Key Trustee Server](#).

5. Start the Key HSM Service

Start the Key HSM service:

```
sudo service keyhsm start
```

6. Establish Trust Between Key Trustee Server and Key HSM

Establish trust between the Key Trustee Server and the Key HSM by specifying the path to the private key and certificate:

```
sudo ktadmin keyhsm --server https://keyhsm01.example.com:9090 \
--client-certfile /etc/pki/cloudera/certs/mycert.crt \
--client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
```

For a password-protected Key Trustee Server private key, add the `--passphrase` argument to the command (enter the password when prompted):

```
sudo ktadmin keyhsm --passphrase \
--server https://keyhsm01.example.com:9090 \
--client-certfile /etc/pki/cloudera/certs/mycert.crt \
--client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
```

For additional details, see [Integrate Key HSM and Key Trustee Server](#).

7. Remove Configuration Files From Previous Installation

After completing the rollback, remove the saved configuration files from the previous installation:

```
cd /usr/share/keytrustee-server-keyhsm/
rm application.properties.bak
rm keystore.bak
rm truststore.bak
```

Roll Back Navigator Encrypt

To roll back Cloudera Navigator Encrypt:

1. If you have configured and are using an RSA master key file with OAEP padding, then you must revert this setting to its original value:

```
navencrypt key --change
```

2. Stop the Navigator Encrypt mount service:

```
sudo /etc/init.d/navencrypt-mount stop
```

3. Confirm that the mount-stop command completed:

```
sudo /etc/init.d/navencrypt-mount status
```

4. If rolling back to a release lower than NavEncrypt 6.2:

- a. Print the existing ACL rules and save that output to a file:

```
sudo navencrypt acl --print+ vim acls.txt
```

- b. Delete all existing ACLs, for example, if there are a total of 7 ACL rules run:

```
sudo navencrypt acl --del --line=1,2,3,4,5,6,7
```

5. To fully downgrade Navigator Encrypt, manually downgrade all of the associated Navigator Encrypt packages (in the order listed):
 - a. navencrypt
 - b. (Only required for operating systems other than SLES) navencrypt-kernel-module
 - c. (Only required for the SLES operating system) cloudera-navencryptfs-kmp-*<KERNEL_FLAVOR>*

Note: Replace *kernel_flavor* with the kernel flavor for your system. Navigator Encrypt supports the default, xen, and ec2 kernel flavors.

6. If rolling back to a release less than NavEncrypt 6.2
 - a. Reapply the ACL rules:

```
sudo navencrypt acl --add --file=acls.txt
```

7. Recompute process signatures:

```
sudo navencrypt acl --update
```

8. Restart the Navigator Encrypt mount service

```
sudo /etc/init.d/navencrypt-mount start
```

Rollback ZooKeeper

Rollback of the data and service is not expected to be necessary unless dependent services are being rolled back. If it is determined that a ZooKeeper rollback is necessary, the steps are:

1. Stop ZooKeeper
2. Restore the data backup. For example: `cp -rp /var/lib/zookeeper-backup-pre-upgrade-CM-CDH /var/lib/zookeeper/`
3. Start ZooKeeper

Rollback HDFS

You cannot roll back HDFS while high availability is enabled. The rollback procedure in this topic creates a temporary configuration without high availability. Regardless of whether high availability is enabled, follow the steps in this section.

1. Roll back all of the NameNodes. Use the [NameNode backup directory](#) you created before upgrading to Cloudera Base on premises. (/etc/hadoop/conf.rollback.namenode) to perform the following steps on all NameNode hosts:
 - a. Start the JournalNodes using Cloudera Manager:
 1. Go to the HDFS service.
 2. Select the Instances tab.
 3. Select all JournalNode roles from the list.
 4. Click Actions for Selected Start .
 - b. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file on all NameNode hosts (located in the temporary rollback directory) and update the keystore passwords with the actual cleartext passwords. The passwords will have values that look like this:

```
<property>
    <name>ssl.server.keystore.password</name>
    <value>*****</value>
</property>
<property>
    <name>ssl.server.keystore.keypassword</name>
    <value>*****</value>
</property>
```

- c. (TLS only) Edit the `/etc/hadoop/conf.rollback.namenode/ssl-server.xml` file and remove the `hadoop.security.credential.provider.path` property.
- d. (TLS only) Edit the `/etc/hadoop/conf.rollback.namenode/ssl-server.xml` file and change the value of `ssl.server.keystore.location` to `/etc/hadoop/conf.rollback.namenode/cm-auto-host_keystore.jks`
- e. Edit the `/etc/hadoop/conf.rollback.namenode/core-site.xml` and change the value of the `net.topology.script.file.name` property to `/etc/hadoop/conf.rollback.namenode`. For example:

```
# Original property
<property>
<name>net.topology.script.file.name</name>
<value>/var/run/cloudera-scm-agent/process/63-hdfs-NAMENODE/topology.p
y</value>
</property>
```

```
# New property
<property>
<name>net.topology.script.file.name</name>
<value>/etc/hadoop/conf.rollback.namenode/topology.py</value>
</property>
```

- f. Edit the `/etc/hadoop/conf.rollback.namenode/topology.py` file and change the value of `DATA_FILE_NAME` to `/etc/hadoop/conf.rollback.namenode`. For example:

```
DATA_FILE_NAME = '/etc/hadoop/conf.rollback.namenode/topology.map'
```

- g. (Kerberos enabled clusters only) Run the following command:

```
sudo -u hdfs kinit hdfs/<NameNode Host name> -l 7d -kt /etc/hadoop/conf.
rollback.namenode/hdfs.keytab
```

- h. Start active Namenode with the following command:

```
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.namenode namenode -
rollback
```



Important: If the error “`javax.security.auth.login.LoginException: Unable to obtain password from user.`” is displayed on a kerberos enabled cluster, make sure to change the value of `hdfs.keytab` to the absolute path of `conf.rollback.namenode` directory in `hdfs-site.xml` on an active or Standby Name node depending on where the error is reported. For example, `/etc/hadoop/conf.rollback.namenode/hdfs.keytab`.

- i. Log in to the other Namenode and start SBNN with the following command:

```
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.namenode namenode -
bootstrapStandby
```

- j. Press Yes when prompted. This also exits the process after it is done.
 - k. Select `Ctrl + C` for active Namenode to exit the process.
 - l. Restart Failovercontroller, Namenodes, and Journal nodes.
2. Rollback the DataNodes.

Use the [DataNode rollback directory](#) you created before upgrading to CDP Private Cloud Base (`/etc/hadoop/conf.rollback.datanode`) to perform the following steps on all DataNode hosts:

- a. (Clusters with TLS enabled only) Edit the `/etc/hadoop/conf.rollback.datanode/ssl-server.xml` file on all DataNode hosts (Located in the temporary rollback directory.) and update the keystore passwords (`ssl.server.keystore.password` and `ssl.server.keystore.keypassword`) with the actual passwords.

The passwords will have values that look like this:

```
<property>
```

```
<name>ssl.server.keystore.password</name>
<value>*****</value>
</property>
<property>
<name>ssl.server.keystore.keypassword</name>
<value>*****</value>
</property>
```

- b.** (TLS only) Edit the `/etc/hadoop/conf.rollback.datanode/ssl-server.xml` file and remove the `hadoop.security.credential.provider.path` property and change the value of property.

```
ssl.server.keystore.location to /etc/hadoop/conf.rollback.datanode/cm-auto-host_keystore.jks
```

- c.** Edit the `/etc/hadoop/conf.rollback.datanode/hdfs-site.xml` file and remove the `dfs.datanode.max.locked.memory` property.
- d.** If you kerberos enabled cluster then make sure change the value of `hdfs.keytab` to the absolute path of `conf.rollback.datanode` folder in `core-site.xml` and `hdfs-site.xml`
- e.** Run one of the following commands:

- If the DataNode is running with privileged ports (usually 1004 and 1006):

```
cd /etc/hadoop/conf.rollback.datanode
export HADOOP_SECURE_DN_USER=hdfs
export JSVC_HOME=/opt/cloudera/parcels/<PARCEL_FILENAME>/lib/bigtop-utils
hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
```

- If the DataNode is not running on privileged ports:

```
sudo hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
```

When the rolling back of the DataNodes is complete, terminate the console session by typing Control-C. Look for output from the command similar to the following that indicates when the DataNode rollback is complete:

```
Rollback of /dataroot/ycloud/dfs/dn/current/BP-<Block Group number> is complete
```

You may see the following error after issuing these commands:

```
ERROR datanode.DataNode: Exception in secureMain java.io.IOException:
    The path component: '/var/run/hdfs-sockets' in '/var/run/hdfs-sockets/dn' has permissions 0755 uid 39998 and gid 1006.
    It is not protected because it is owned by a user who is not root and not the effective user: '0'.
```

The error message will also include the following command to run:

```
chown root /var/run/hdfs-sockets
```

After running this command, the DataNode will restart successfully. Rerun the DataNode rollback command:

```
sudo hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
```

- f.** If High Availability for HDFS is enabled, restart the HDFS service. In the Cloudera Manager Admin Console, go to the HDFS service and select **Actions Restart**.

- g. If high availability is not enabled for HDFS, use the Cloudera Manager Admin Console to restart all NameNodes and DataNodes.
 1. Go to the HDFS service.
 2. Select the Instances tab
 3. Select all DataNode and NameNode roles from the list.
 4. Click Actions for Selected Restart .
- 3. If high availability is not enabled for HDFS, roll back the Secondary NameNode.
 - a. (Clusters with TLS enabled only) Edit the `/etc/hadoop/conf.rollback.secondarynamenode/ssl-server.xml` file on all Secondary NameNode hosts (Located in the temporary rollback directory.) and update the keystore passwords with the actual cleartext passwords. The passwords will have values that look like this:

```
<property>
  <name>ssl.server.keystore.password</name>
  <value>*****</value>
</property>
<property>
  <name>ssl.server.keystore.keypassword</name>
  <value>*****</value>
</property>
```

- b. (TLS only) Edit the `/etc/hadoop/conf.rollback.secondarynamenode/ssl-server.xml` file and remove the `hadoop.security.credential.provider.path` property and change the value of property `ssl.server.keystore.location` to

```
/etc/hadoop/conf.rollback.secondarynamenode/cm-auto-host_keystore.jks
```

- c. Log in to the Secondary NameNode host and run the following commands:

```
rm -rf /dfs/snn/*
cd /etc/hadoop/conf.rollback.secondarynamenode/
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.secondarynamenode secondarynamenode -format
```

When the rolling back of the Secondary NameNode is complete, terminate the console session by typing Control-C. Look for output from the command similar to the following that indicates when the Secondary NameNode rollback is complete:

```
2020-12-21 17:09:36,239 INFO namenode.SecondaryNameNode: Web server init done
```

4. Restart the HDFS service. Open the Cloudera Manager Admin Console, go to the HDFS service page, and select Actions Restart .

The Restart Command page displays the progress of the restart. Wait for the page to display the Successfully restarted service message before continuing.

For more information on HDFS, see [HDFS troubleshooting](#) documentation.

Start HBase

You might encounter other errors when starting HBase (for example, replication-related problems, region assignment related issues, and meta region assignment problems). In this case, you must delete the znode in ZooKeeper and then start HBase again. (This deletes the replication peer information and you need to re-configure your replication schedules)

1. In Cloudera Manager, look up the value of the `zookeeper.znode.parent` property. The default value is `/hbase`.

2. Connect to the ZooKeeper ensemble by running the following command from any HBase gateway host.

```
zookeeper-client -server zookeeper_ensemble
```

To find the value to use for `zookeeper_ensemble`, open the `/etc/hbase/conf.cloudera.<HBase service name>/hbase-site.xml` file on any HBase gateway host. Use the value of the `hbase.zookeeper.quorum` property.



Note: If you have deployed a secure cluster, you must connect to ZooKeeper using a client `jaas.conf` file. You can find such a file in an HBase process directory (`/var/run/cloudera-scm-agent/process/`).

3. Specify the `jaas.conf` using the JVM flags by running the following commands in the ZooKeeper client.

```
CLIENT_JVMFLAGS=
"-Djava.security.auth.login.config=/var/run/cloudera-scm
agent/process/HBase_process_directory/jaas.conf"
zookeeper-client -server <zookeeper_ensemble>
```

The ZooKeeper command-line interface opens.

4. Enter the following command.

```
rmr /hbase
```

If you have deployed a secure cluster, enter the following command:

```
deleteall /hbase
```

If you see the message `Node not empty: /hbase/tokenauth`, you must re-run the same command and restart the HBase service.

5. Restart the HBase service.

After HBase is healthy, ensure that you restore the states of the Balancer and Normalizer (enable them if they were enabled before the rollback). Also re-enable the Merge and Split operations you disabled before the rollback to avoid the Master Procedure incompatibility problem.

Run the following commands in HBase Shell:

```
balance_switch true
normalizer_switch true
splitormerge_switch 'SPLIT', true
splitormerge_switch 'MERGE', true
```

Rollback Solr

1. Start the HDFS, ZooKeeper and Ranger services.
2. Restore the Solr-specific `znodes` in ZooKeeper using the snapshot you took before the upgrade.



Note: This step is only necessary if Solr is rolled back independent of other components. If ZooKeeper is rolled back as well, skip this step.

3. Start the Solr service.



Note: If the state of one or more Solr core is down and the Solr log contains a similar error message:

```
"org.apache.lucene.store.LockObtainFailedException: Lock obtain timed
out:
    org.apache.solr.store.hdfs.HdfsLockFactory"
```

it is necessary to clean up the HDFS locks in the index directories.

On all affected Solr nodes perform the following steps:

- a. Stop the Solr node using Cloudera Manager.
- b. Remove the HdfsDirectory@[***HEX ID***]-write.lock file from the index directory.

```
hdfs dfs -rm "/solr/[***COLLECTION NAME***]/[***CORE***]/dat
a/[***INDEX DIRECTORY NAME***]/HdfsDirectory@[***HEX ID***]
lockFactory=org.apache.solr.store.hdfs.HdfsLockFact
ory@[***HEX ID***]-write.lock"
```

For example:

```
hdfs dfs -rm "/solr/testCollection/core_node1/data/index/HdfsDir
ectory@5d07feac
lockFactory=org.apache.solr.store.hdfs.HdfsLockFacto
ry@7df08aad-write.lock"
```

- c. Start the Solr node using Cloudera Manager.
4. Restore the Solr collections using the backups you created before starting the upgrade. For more information, see [Restoring a Solr collection](#).



Note: Manually restoring collections on HDFS is only necessary if you roll back Solr independent of other components. If HDFS is rolled back as well, this is not necessary. You always need to manually restore collections that were stored on local file system.

5. Restart Lily HBase Indexer (ks_indexer).

Rollback Atlas

Rollback Atlas Solr Collections

Atlas has several collections in Solr that must be restored from the pre-upgrade backup - vertex_index, edge_index, and fulltext_index. These collections may already have been restored using the [Rollback Solr](#) documentation. If the collections are not yet restored, you must restore collections now using the [Rollback Solr](#) documentation.

Rollback Atlas HBase Tables

1. From a client host, start the HBase shell `hbase shell`
2. Within the HBase shell, list the snapshots, that must contain the pre-upgrade snapshots `list_snapshots`
3. Within the HBase shell, disable the atlas_janus table, restore the snapshot, and enable the table

```
disable 'atlas_janus'

restore_snapshot '<name of atlas_janus snapshot from
list_snapshots>'

enable 'atlas_janus'
```


4. Within the HBase shell, disable the ATLAS_ENTITY_AUDIT_EVENTS table, restore the snapshot, and enable the table

```
disable 'ATLAS_ENTITY_AUDIT_EVENTS'

restore_snapshot '<name of ATLAS_ENTITY_AUDIT_EVENTS snapshot
from list_snapshots>'

enable 'ATLAS_ENTITY_AUDIT_EVENTS'
```

5. Restart Atlas.

Rollback Kudu

Rollback depends on which backup method was used. There are two forms of backup/restore in Kudu. Spark job to create or restore a full/incremental backup or backup the entire Kudu node and restore it later. Restoring the Kudu data differs between the Spark job and full node backup approaches.

See the [Kudu backup and restore](#) docs for data recovery steps. The OS level restore is not recommended if there is a backup.



Note: It is recommended to have the tables empty/deleted before restoring them as additional writes to the table after upgrade would not be deleted after the restore.

Rollback YARN Queue Manager

You can rollback YARN Queue Manager using the pre-upgrade backup of config-service.mv.db and config-service.trace.db.

1. Navigate to the YARN Queue Manager service in Cloudera Manager and record the configuration value for config_service_db_loc (or queuemanager_user_home_dir if blank) and the host where the YARN Queue Manager Store is running.
2. Stop the YARN Queue Manager service.
3. SSH to the YARN Queue Manager Store host and copy the pre-upgrade config-service.mv.db and config-service.trace.db to the config_service_db_loc obtained in the previous step.
4. Start the YARN Queue Manager service.

Rollback Kafka

You can rollback Kafka as long as the following criteria are met.

- The log format Kafka properties inter.broker.protocol.version and log.message.format.version are set before the upgrade and are not removed or cleared after the upgrade.
- There is a backup with the pre-upgrade state of the cluster from where you can restore the data, such as a remote cluster replicated with SRM. For guidance on restoring data from an SRM-replicated cluster, contact Cloudera support.



Note: Neither rollbacks nor downgrades are possible after clearing log format Kafka properties.

Deploy the Client Configuration

1. On the Cloudera Manager Home page, click the Actions menu and select Deploy Client Configuration.
2. Click Deploy Client Configuration.

Restart the Cluster

You must restart the cluster using the following steps.

1. On the Cloudera Manager Home page, click the Actions menu and select Restart.

- Click Restart that appears in the next screen to confirm. If you have enabled [high availability for HDFS](#), you can choose [Rolling Restart](#) instead to minimize cluster downtime. The Command Details window shows the progress of stopping services.

When All services successfully started appears, the task is complete and you can close the Command Details window.

Post rollback steps

Streams Replication Manager (SRM)

Reset the state of the internal Kafka Streams application. Run the following command on the hosts of the SRM Service role.

```
kafka-streams-application-reset \
  --bootstrap-servers [***SRM SERVICE HOST***] \
  --config-file [***PROPERTIES FILE***] \
  --application-id srm-service_v2
```

Replace `[***PROPERTIES FILE***]` with the location of a configuration file that contains all necessary security properties that are required to establish a connection with the Kafka service. This option is only required if your Kafka service is secured.

Cruise Control

- Insert the removed goal back to the relevant goal sets, but with the renamed goal name RackAwareGoal (Not RackAwareDistributionGoal)
- Restart Cruise Control

Oozie

Execute the Install Oozie ShareLib action through Cloudera Manager:

- Go to the Oozie service.
- Select Actions Install Oozie ShareLib .

Finalize the HDFS Upgrade

This step should be performed only after all validation is completed. For more information, see [Finalize the HDFS upgrade](#) documentation.

Procedure to Rollback between SP or CHF releases of the same Cloudera Base on premises release line

You can perform a rollback from a just installed CHF (Cumulative Hotfix) or SP (Service Pack) release to the previously installed CHF or SP release, provided both releases belong to the same Cloudera Base on premises release line.

Important Limitations

This rollback procedure has the following limitations:

- You cannot use this document to rollback between different Cloudera Base on premises release lines. For example, you cannot rollback from 7.3.1 to 7.1.9 or 7.1.7.
- You cannot use this document to rollback between CHF or SP releases of different Cloudera Base on premises release lines. For example, you cannot rollback from 7.3.1 SP1 CHF1 to 7.1.9 SP1 CHF1.
- You can rollback from 7.1.9 SP1 CHF7 to 7.1.9 SP1 CHF5.
- You can rollback from 7.1.9 SP1 to 7.1.9 (vanilla release).

For further limitations regarding the applicability of the rollback procedure, see [Review Limitations](#).

Terminology Clarification

- The term “*previous CHF or SP release*” refers to the CHF or SP release you installed before the current upgrade attempt started. In other words, this release is the rollback target.
- This document allows you to rollback only to the *previous CHF or SP release*, not an *arbitrary CHF or SP release* of the same release line.

To rollback to the previous CHF or SP release, execute the following steps:

Procedure

1. Log in to the Cloudera Manager Admin Console.
2. Click Parcels from the left menu.
3. Click Parcel Repositories & Network Settings.



Note: If the *previous CHF or SP release*'s URL is available in the parcel repository, then you can skip to step 8. If not, then proceed with step 4.

4. In the Remote Parcel Repository URLs section, click the "+" icon and add the *previous CHF or SP release*'s URL for your Parcel repository.
5. Click Save & Verify Configuration. A message with the status of the verification appears above the Remote Parcel Repository URLs section. If the URL is not valid, check the URL and enter the correct URL.
6. After the URL is verified, click Close.
7. Locate the row in the table that contains the new Cloudera Runtime parcel and click the Download button.
8. After the download of the new Cloudera Runtime parcel is complete, click the Distribute button.
Wait for the parcel to be distributed and unpacked before continuing. Cloudera Manager displays the status of the Cloudera Runtime parcel distribution. Click on the status display to view detailed status for each host.
9. Click Activate. Cloudera Runtime parcels are activated on the cluster.
10. When the parcel is activated, click the Cloudera Manager logo to return to the home page.
The cluster is now rolled back to the *previous CHF or SP release*.
11. When the parcel is activated, click the Actions menu next to the cluster name and select Post Cloudera Runtime Upgrade.
12. Restart the cluster: Click the Actions menu and select Restart.