

HDP3 to CDP Private Cloud Base One Stage upgrade

HDP3 to CDP Private Cloud Base One Stage upgrade

Date published: 2022-12-02

Date modified: 2024-03-19

The Cloudera logo is displayed in a bold, orange, sans-serif font. The word "CLOUDERA" is written in all caps, with a stylized 'E' that has three horizontal bars.

<https://docs.cloudera.com/>

Legal Notice

© Cloudera Inc. 2024. 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

In-place upgrade overview.....	5
Cluster environment readiness.....	5
Disk space and mountpoint considerations.....	6
Downloading and Publishing the Package Repository.....	7
Downloading and Publishing the Parcel Repository.....	10
Hadoop Users (user:group) and Kerberos Principals.....	12
Upgrading the cluster's underlying OS.....	29
In-Place and Restore.....	29
Move and Decommission.....	30
Versions and supported services for migration.....	30
Software download matrix for HDP 3.1.5 and 2.6.5 to CDP 7.1.x.....	30
Sample data ingestion.....	31
Merge Independent Hive and Spark Catalogs.....	31
Cloudera Manager Installation and Setup.....	32
Installing Cloudera Management Service.....	33
Setting up CMA server.....	35
Registering Ambari Cloudera Manager pair for source cluster.....	38
Registering Ambari Cloudera Manager pair for target cluster.....	40
Preparing configurations.....	41
HDP to CDP Private Cloud Base Upgrade.....	43
Execution steps.....	46
Troubleshooting.....	50

Backup HDP services from CDP 7.1.x.....	51
--	-----------

Rollback HDP services from CDP 7.1.x.....	51
--	-----------

Automated rollback.....	52
Manual rollback.....	53
Restore old configuration symlinks.....	53
Kerberos.....	54
ZooKeeper.....	54
Ambari Infra Solr.....	54
Ranger.....	55
HDFS.....	58
YARN.....	60
HBase.....	60
Kafka.....	60
Atlas.....	60
Hive.....	61
Spark.....	61
Oozie.....	61
Knox.....	61
Zeppelin.....	61

In-place upgrade overview

The process of upgrading to CDP Private Cloud Base involves understanding the supported in-place upgrade paths and verifying the software and hardware considerations and requirements prior to performing the upgrade steps.

About this task

To plan your upgrade from Ambari managed HDP 3.1.5 or 2.6.5 to CDP Private Cloud Base, you must be aware of the in-place upgrade path along with the pre-upgrade, upgrade, and post-upgrade tasks.



Note:

- HDP one-stage upgrade supports the upgrade from HDP 3.1.5 to CDP 7.1.7 SP1, CDP 7.1.8, CDP 7.1.7 SP2, and 7.1.9.
- HDP one-stage upgrade supports the upgrade from HDP 2.6.5 to CDP 7.1.8 and CDP 7.1.7 SP2.
- CMA 2.8.0 and higher requires JDK11 in local mode.

One stage versus two stage upgrade:

One stage	Two stage
No upgrade to HDP 7 or intermediate bits	Upgrade from HDP 3.1.5 or 2.6.5 cluster to HDP 7.1.x (HDP Intermediate Bits)
No upgrade to Ambari 7. Direct upgrade from Ambari to Cloudera Manager	Upgrading Management of the cluster from Ambari to Cloudera Manager
Executing meta-data (schema) upgrades done as part of post transition steps that allows for an extended cluster uptime	Executing meta-data (schema) upgrades done by Ambari

The following table shows the supported versions of the AM2CM tool, Cloudera Manager, and Runtime for each upgrade path:

Upgrade path	AM2CM version	Cloudera Manager	Runtime
HDP 3.1.5 to CDP 7.1.9	AM2CM 3.2	Cloudera Manager 7.11.3	CDH-7.1.9
HDP 3.1.5 to CDP 7.1.7 SP2	AM2CM 3.2	Cloudera Manager 7.6.7	CDH-7.1.7 SP2 including Accumulo
HDP 3.1.5 to CDP 7.1.8	AM2CM 3.2	Cloudera Manager 7.7.3	CDH-7.1.8 including Accumulo
HDP 3.1.5 to CDP 7.1.7 SP1	AM2CM 3.2	Cloudera Manager 7.6.1	CDH-7.1.7 SP1
HDP 2.6.5 to CDP 7.1.7 SP2	AM2CM 3.2	Cloudera Manager 7.6.7	CDH-7.1.7 SP2 including Accumulo
HDP 2.6.5 to CDP 7.1.8	AM2CM 3.2	Cloudera Manager 7.7.3	CDH-7.1.8 including Accumulo

Cluster environment readiness

You must ensure all the nodes have the supported operating system, Java version, and base Ambari version. Verify the disk space and mount point requirements before you begin the upgrade to the recommended interim HDP bits and CDP Private Cloud Base.


Operating System on all nodes	Operating System & Upgrading the cluster's underlying OS
Requirements and Supported Versions	CDP Private Cloud Base Requirements and Supported Versions
Java version on all nodes	Java Versions
Ambari base version on all nodes	Ambari version is 2.7.5.x.
Repositories	Software download matrix
Review disk space	Disk space and mountpoint considerations



Note: You must deactivate and uninstall Spark in Ambari before distributing the parcel in Cloudera Manager.

Disk space and mountpoint considerations

Review the minimum disk space requirements before you upgrade from HDP 3.1.5 to CDP 7.1.x.

Partition	Storage	Detail
/usr/hdp	10 GB	Minimum space required for each installed HDP version.
/opt/cloudera	100 GB for CM 100 GB for all hosts	Minimum space required for each installed and retained CDP version.
/usr/hdp	35 GB	If you are upgrading from HDP 3.1.5.x to CDP 7.1.x, there is an interim step to upgrade to HDP intermediate bits. You would need a minimum 30 GB available space to make the transition.
/var/log	200 GB - 500 GB	Minimum space required for storing the logs.  Note: Ensure this is not part of the root OS partition.
/var/*	2 GB	Minimum space required for Cloudera Manager agent and for the Cloudera components.
/tmp	20GB	At least 20 GB of free space required for storing the temp data by CLDR services.
/<data-dir>	Varies	Datanode, Kafka Logs, Namenode Image and Edits, Journal Node, ZooKeeper. These should all be on separate mounts to avoid disk issues with the os partition.

Partition	Storage	Detail
		<p>Performance Impact: For the Namenode, Journal Node, and ZooKeeper data directories, these should be on dedicated disks (and mounts) for the most optimal performance of these critical services. Disk contention with other write operations will have an impact on these services.</p> <p>Performance Impact: Data-directories for Datanode and Kafka-Logs should be simple JBOD drives. RAID support has a support impact and these services are storage redundant at the service level. RAID is NOT recommended for these service data directories.</p>
/<yarn-local>	Varies by workload “yarn.nodemanager.local-dirs” Between 5-25% of host storage depending on workload types.	Applications that are heavy in MR technologies will benefit tremendously by using the “SSD” storage.

For more information on the hardware requirements for Runtime components, see [Cloudera Runtime](#).

For more information on the CM server storage requirements, see [Cloudera Manager Server](#).

For more information on the CDP Private Cloud Base requirements, see [CDP Private Cloud Base Requirements and Supported Versions](#).

Downloading and Publishing the Package Repository

Download the package repository for the product you want to install.

Cloudera Manager 7.11.3

1. Download the package repository for the product you want to install:

Cloudera Manager 7

To download the files for a Cloudera Manager release, download the repository tarball for your operating system. Then unpack the tarball, move the files to the web server directory, and modify file permissions. For example:

```
$ sudo mkdir -p /var/www/html/cloudera-repos/cm7
```

```
$ wget https://[username]:[password]@archive.cloudera.com/p/cm7/7.11.3/repo-as-tarball/cm7.11.3-redhat7.tar.gz
```

```
$ tar xvfz cm7.11.3-redhat7.tar.gz -C /var/www/html/cloudera-repos/cm7 --strip-components=1
```

```
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. If you do not see anything, your Web server may have been configured to not show indexes.

Cloudera Manager 7.7.1

1. Download the package repository for the product you want to install:

Cloudera Manager 7

To download the files for a Cloudera Manager release, download the repository tarball for your operating system. Then unpack the tarball, move the files to the web server directory, and modify file permissions. For example:

```
$ sudo mkdir -p /var/www/html/cloudera-repos/cm7
```

```
$ wget https://[username]:[password]@archive.cloudera.com/p/cm7/7.7.1/repo-as-tarball/cm7.7.1-redhat7.tar.gz
```

```
$ tar xvfz cm7.7.1-redhat7.tar.gz -C /var/www/html/cloudera-repos/cm7 --strip-components=1
```

```
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. If you do not see anything, your Web server may have been configured to not show indexes.

Cloudera Manager 7.6.7

1. Download the package repository for the product you want to install:

Cloudera Manager 7

To download the files for a Cloudera Manager release, download the repository tarball for your operating system. Then unpack the tarball, move the files to the web server directory, and modify file permissions. For example:

```
$ sudo mkdir -p /var/www/html/cloudera-repos/cm7
```

```
$ wget https://[username]:[password]@archive.cloudera.com/p/cm7/7.6.7/repo-as-tarball/cm7.6.7-redhat7.tar.gz
```

```
$ tar xvfz cm7.6.7-redhat7.tar.gz -C /var/www/html/cloudera-repos/cm7 --strip-components=1
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/cm7
```


Cloudera Manager 7.6.1

1. Download the package repository for the product you want to install:

Cloudera Manager 7

To download the files for a Cloudera Manager release, download the repository tarball for your operating system. Then unpack the tarball, move the files to the web server directory, and modify file permissions. For example:

```
$ sudo mkdir -p /var/www/html/cloudera-repos/cm7
```

```
$ wget https://[username]:[password]@archive.cloudera.com/p/cm7/7.6.1/repo-as-tarball/cm7.6.1-redhat7.tar.gz
```

```
$ tar xvfz cm7.6.1-redhat7.tar.gz -C /var/www/html/cloudera-repos/cm7 --strip-components=1
```

```
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. If you do not see anything, your Web server may have been configured to not show indexes.

Cloudera Manager 7.4.4

1. Download the package repository for the product you want to install:

Cloudera Manager 7

To download the files for a Cloudera Manager release, download the repository tarball for your operating system. Then unpack the tarball, move the files to the web server directory, and modify file permissions. For example:

```
$ sudo mkdir -p /var/www/html/cloudera-repos/cm7
```

```
$ wget https://[username]:[password]@archive.cloudera.com/p/cm7/7.4.4-24429768/repo-as-tarball/cm7.4.4-redhat7.tar.gz
```

```
$ tar xvfz cm7.4.4-redhat7.tar.gz -C /var/www/html/cloudera-repos/cm7 --strip-components=1
```

```
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. If you do not see anything, your Web server may have been configured to not show indexes.

Cloudera Manager 7.3.1

1. Download the package repository for the product you want to install:

Cloudera Manager 7

To download the files for a Cloudera Manager release, download the repository tarball for your operating system. Then unpack the tarball, move the files to the web server directory, and modify file permissions. For example:

```
$ sudo mkdir -p /var/www/html/cloudera-repos/cm7
```

```
$ wget https://[username]:[password]@archive.cloudera.com/p/cm7/7.3.1/repo-as-tarball/cm7.3.1-redhat7.tar.gz
```

```
$ tar xvfz cm7.3.1-redhat7.tar.gz -C /var/www/html/cloudera-repos/cm7 --strip-components=1
```

```
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. If you do not see anything, your Web server may have been configured to not show indexes.

Downloading and Publishing the Parcel Repository

Download the parcels that you want to install and publish the parcel directory.

Procedure

1. Download manifest.json and the parcel files for the product you want to install:

Runtime 7.1.9

To download the files for the latest Runtime 7 release, run the following commands on the Web server host only if you are upgrading from HDP 3.1.5 to CDP 7.1.9:

```
sudo mkdir -p /var/www/html/cloudera-repos
sudo wget --recursive --no-parent --no-host-directories https://[username]:[password]@archive.cloudera.com/p/cdh7/7.1.9/parcels/ -P /var/www/html/cloudera-repos
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/p/cdh7
```

Runtime 7.1.8

To download the files for the latest 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/cdh7/7.1.8/parcels/ -P /var/www/html/cloudera-repos
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/p/cdh7
```

Runtime 7.1.7.3000

Apache Impala, Apache Kudu, Apache Spark 2, and Cloudera Search are included in the Runtime parcel. To download the files for the latest 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/cdh7/7.1.7.3000/parcels/ -P /var/www/html/cloudera-repos
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/p/cdh7
```

Runtime 7.1.7.2000

Apache Impala, Apache Kudu, Apache Spark 2, and Cloudera Search are included in the Runtime parcel. To download the files for the latest 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/cdh7/7.1.7.2000/parcels/ -P /var/www/html/cloudera-repos
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/p/cdh7
```

Runtime 7.1.7.1000

Apache Impala, Apache Kudu, Apache Spark 2, and Cloudera Search are included in the Runtime parcel. To download the files for the latest 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/cdh7/7.1.7.1000/parcels/ -P /var/www/html/cloudera-repos
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/p/cdh7
```

Runtime 7.1.7.78

Apache Impala, Apache Kudu, Apache Spark 2, and Cloudera Search are included in the Runtime parcel. To download the files for the latest 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/cdh7/7.1.7.78/parcels/ -P /var/www/html/cloudera-repos
```

```
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/p/cdh7
```



Note: 7.1.7.78 is same as 7.1.7.0. However, 7.1.7.78 includes a critical vulnerability in log4j.

Runtime 7.1.6.0

Apache Impala, Apache Kudu, Apache Spark 2, and Cloudera Search are included in the Runtime parcel. To download the files for the latest 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/cdh7/7.1.6.0/parcels/ -P /var/www/html/cloudera-repos
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/p/cdh7
```

Sqoop Connectors

To download the parcels for a Sqoop Connector release, run the following commands on the Web server host. This example uses the latest available Sqoop Connectors:

```
sudo mkdir -p /var/www/html/cloudera-repos
sudo wget --recursive --no-parent --no-host-directories http://archive.cloudera.com/sqoop-connectors/parcels/latest/ -P /var/www/html/cloudera-repos
sudo chmod -R ugo+rX /var/www/html/cloudera-repos/sqoop-connectors
```

If you want to create a repository for a different Sqoop Connector release, replace latest with the Sqoop Connector version that you want. You can see a list of versions in the parcels parent directory.

2. 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.

Hadoop Users (user:group) and Kerberos Principals

During the Cloudera Manager installation process, several Linux user accounts and groups are created by default. These are listed in the table below. Integrating the cluster to use Kerberos for authentication requires creating Kerberos principals and keytabs for these user accounts.

Table 1: Users and Groups

Component (Version)	Unix User ID	Groups	Functionality
Apache Atlas	atlas	atlas, hadoop	Apache Atlas by default has atlas as user and group. It is configurable
Apache Flink	flink	flink	The Flink Dashboard runs as this user.
Apache HBase	hbase	hbase	The Master and the RegionServer processes run as this user.
Apache HBase Indexer	hbase	hbase	The indexer servers are run as this user.

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

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

Keytabs and Keytab File Permissions

Linux user accounts, such as hdfs, are mapped to the username portion of the Kerberos principal names, as follows:

```
username/host.example.com@EXAMPLE.COM
```

For example, the Kerberos principal for Apache Hive would be:

```
hive/host.example.com@EXAMPLE.COM
```

Keytabs that contain multiple principals are merged automatically from individual keytabs by Cloudera Manager. If you override a service configuration to not use the CM-provided keytab, then you must ensure that all the principals required for the given role instance on a specific host are merged together in the keytab file you deploy manually on that host.

For example, for Filename (*.keytab), the Atlas keytab filename would be atlas.keytab, HBase would be hbase.keytab, and Cloudera Manager would be cmf.keytab and scm.keytab.

Keytab File Owner:Group matters when Cloudera Manager starts a role. For example, Cloudera Manager starts the role "DataNode". Cloudera Manager launches the DataNode process as a user (here, "hdfs"). Because that process needs to access the HDFS keytab, Cloudera Manager puts the HDFS keytab in the DataNode's process directory, and the keytab is given the owner:group that is listed in the table. Thus, the DataNode process properly owns the keytab file.

The tables below lists the usernames to use for Kerberos principal names, for clusters managed by Cloudera Manager.

Apache Atlas

Role: atlas-ATLAS_SERVER

Kerberos Principals

atlas

Filename (*.keytab)

atlas

Keytab File Owner:Group

atlas:atlas

File Permission (octal)

600

Apache Flink

Role: flink

Kerberos Principals

flink

Filename (*.keytab)

flink

Keytab File Owner:Group

flink:flink

File Permission (octal)

600

Apache HBase

Role: hbase-HBASETHRIFTSERVER

Kerberos Principals

hbase, HTTP

Filename (*.keytab)

hbase, HTTP

Keytab File Owner:Group

hbase:hbase

File Permission (octal)

600

Role: hbase-REGIONSERVER

Kerberos Principals

hbase, HTTP

Filename (*.keytab)

hbase, HTTP

Keytab File Owner:Group

hbase:hbase

File Permission (octal)

600

Role: hbase-HBASERESTSERVER**Kerberos Principals**

hbase, HTTP

Filename (*.keytab)

hbase, HTTP

Keytab File Owner:Group

hbase:hbase

File Permission (octal)

600

Role: hbase-MASTER**Kerberos Principals**

hbase, HTTP

Filename (*.keytab)

hbase, HTTP

Keytab File Owner:Group

hbase:hbase

File Permission (octal)

600

Apache HBase indexer**Role: ks_indexer-HBASE_INDEXER****Kerberos Principals**

hbase, HTTP

Filename (*.keytab)

hbase

Keytab File Owner:Group

hbase:hbase

File Permission (octal)

600

Apache HDFS**Role: hdfs-NAMENODE****Kerberos Principals**

hdfs, HTTP

Filename (*.keytab)

hdfs

Keytab File Owner:Group

hdfs:hdfs

File Permission (octal)

600

Role: hdfs-DATANODE**Kerberos Principals**

hdfs, HTTP

Filename (*.keytab)

hdfs

Keytab File Owner:Group

hdfs:hdfs

File Permission (octal)

600

Role: hdfs-SECONDARYNAMENODE**Kerberos Principals**

hdfs, HTTP

Filename (*.keytab)

hdfs

Keytab File Owner:Group

hdfs:hdfs

File Permission (octal)

600

Apache Hive, Hive on Tez**Role: hive-HIVESERVER2****Kerberos Principals**

hive

Filename (*.keytab)

hive

Keytab File Owner:Group

hive:hive

File Permission (octal)

600

Role: hive-HIVEMETASTORE**Kerberos Principals**

hive

Filename (*.keytab)

hive

Keytab File Owner:Group

cloudera-scm:cloudera-scm

File Permission (octal)

600

Apache Impala**Role: impala-STATESTORE****Kerberos Principals**

impala, HTTP

Filename (*.keytab)

impala

Keytab File Owner:Group

impala:impala

File Permission (octal)

600

Role: impala-CATALOGSERVER**Kerberos Principals**

impala, HTTP

Filename (*.keytab)

impala

Keytab File Owner:Group

impala:impala

File Permission (octal)

600

Role: impala-IMPALAD**Kerberos Principals**

impala, HTTP

Filename (*.keytab)

impala

Keytab File Owner:Group

impala:impala

File Permission (octal)

600

Apache Kafka**Role: kafka-KAFKA_BROKER****Kerberos Principals**

kafka

Filename (*.keytab)

kafka

Keytab File Owner:Group

kafka:kafka

File Permission (octal)

600

Role: kafka-KAFKA_MIRROR_MAKER**Kerberos Principals**

kafka_mirror_maker

Filename (*.keytab)

kafka

Keytab File Owner:Group

kafka:kafka

File Permission (octal)

600

Role: kafka-KAFKA_CONNECT**Kerberos Principals**

kafka

Filename (*.keytab)

kafka

Keytab File Owner:Group

kafka:kafka

File Permission (octal)

600

Apache Knox

Role: knox-KNOX_GATEWAY

Kerberos Principals

knox, HTTP

Filename (*.keytab)

hbase

Keytab File Owner:Group

knox:knox

File Permission (octal)

600

Apache Kudu

Role: kudu-KUDU_MASTER

Kerberos Principals

kudu

Filename (*.keytab)

kudu

Keytab File Owner:Group

kudu:kudu

File Permission (octal)

600

Role: kudu-KUDU_TSERVER

Kerberos Principals

kudu

Filename (*.keytab)

kudu

Keytab File Owner:Group

kudu:kudu

File Permission (octal)

600

Apache Livy

Role: livy-LIVY_SERVER

Kerberos Principals

livy

Filename (*.keytab)

livy

Keytab File Owner:Group

livy:livy

File Permission (octal)

600

Apache NiFi**Role:** nifi**Kerberos Principals**

nifi, HTTP

Filename (*.keytab)

nifi

Keytab File Owner:Group

nifi:nifi

File Permission (octal)

600

Apache NiFi Registry**Role:** nifiregistry**Kerberos Principals**

nifiregistry, HTTP

Filename (*.keytab)

nifiregistry

Keytab File Owner:Group

nifiregistry:nifiregistry

File Permission (octal)

600

Apache Oozie**Role:** oozie-OOZIE_SERVER**Kerberos Principals**

oozie, HTTP

Filename (*.keytab)

oozie

Keytab File Owner:Group

oozie:oozie

File Permission (octal)

600

Apache Ozone**Role:** ozone-OZONE_MANAGER**Kerberos Principals**

om, HTTP

Filename (*.keytab)

ozone

Keytab File Owner:Group

hdfs:hdfs

File Permission (octal)

600

Role: ozone-STORAGE_CONTAINER_MANAGER**Kerberos Principals**

scm, HTTP

Filename (*.keytab)

ozone

Keytab File Owner:Group

hdfs:hdfs

File Permission (octal)

600

Role: ozone-OZONE_DATANODE**Kerberos Principals**

dn, HTTP

Filename (*.keytab)

ozone

Keytab File Owner:Group

hdfs:hdfs

File Permission (octal)

600

Role: ozone-OZONE_RECON**Kerberos Principals**

recon, HTTP

Filename (*.keytab)

ozone

Keytab File Owner:Group

hdfs:hdfs

File Permission (octal)

600

Role: ozone-S3_GATEWAY**Kerberos Principals**

HTTP

Filename (*.keytab)

ozone

Keytab File Owner:Group

hdfs:hdfs

File Permission (octal)

600

Apache Phoenix**Role: phoenix-PHOENIX_QUERY_SERVER****Kerberos Principals**

phoenix, HTTP

Filename (*.keytab)

phoenix

Keytab File Owner:Group

phoenix:phoenix

File Permission (octal)

600

Apache Ranger

Role: ranger-RANGER_ADMIN

Kerberos Principals

rangeradmin, rangerlookup, HTTP

Filename (*.keytab)

ranger

Keytab File Owner:Group

ranger:ranger

File Permission (octal)

600

Role: ranger-RANGER_USERSYNC

Kerberos Principals

rangerusersync

Filename (*.keytab)

ranger

Keytab File Owner:Group

ranger:ranger

File Permission (octal)

600

Role: ranger-RANGER_TAGSYNC

Kerberos Principals

rangertagsync

Filename (*.keytab)

ranger

Keytab File Owner:Group

ranger:ranger

File Permission (octal)

600

Apache Ranger KMS

Role: ranger-RANGER_TAGSYNC

Kerberos Principals

rangerkms, HTTP

Filename (*.keytab)

ranger_kms

Keytab File Owner:Group

kms:kms

File Permission (octal)

600

Apache Ranger Raz**Role:** ranger-RANGER_RAZ**Kerberos Principals**

rangerraz, HTTP

Filename (*.keytab)

rangerraz

Keytab File Owner:Group

ranger:rangerraz

File Permission (octal)

600

Apache Ranger RMS**Role:** ranger-RANGER_RMS**Kerberos Principals**

rangerrms

Filename (*.keytab)

rangerrms

Keytab File Owner:Group

ranger:rangerrms

File Permission (octal)

600

Apache Solr**Role:** solr-SOLR_SERVER**Kerberos Principals**

solr, HTTP

Filename (*.keytab)

solr

Keytab File Owner:Group

solr:solr

File Permission (octal)

600

Apache Spark**Role:** spark_on_yarn-SPARK_YARN_HISTORY_SERVER**Kerberos Principals**

spark

Filename (*.keytab)

spark

Keytab File Owner:Group

spark:spark

File Permission (octal)

600

Apache YARN**Role:** yarn-NODEMANAGER

Kerberos Principals

yarn, HTTP

Filename (*.keytab)

yarn

Keytab File Owner:Group

yarn:hadoop

File Permission (octal)

644

Role: yarn-RESOURCEMANAGER**Kerberos Principals**

yarn, HTTP

Filename (*.keytab)

yarn

Keytab File Owner:Group

yarn:hadoop

File Permission (octal)

600

Role: yarn-JOBHISTORY**Kerberos Principals**

mapred

Filename (*.keytab)

mapred

Keytab File Owner:Group

yarn:hadoop

File Permission (octal)

600

Apache Zeppelin**Role: zeppelin-ZEPPELIN_SERVER****Kerberos Principals**

zeppelin, HTTP

Filename (*.keytab)

zeppelin

Keytab File Owner:Group

zeppelin:zeppelin

File Permission (octal)

600

Apache ZooKeeper**Role: zookeeper-server****Kerberos Principals**

zookeeper

Filename (*.keytab)

zookeeper

Keytab File Owner:Group

zookeeper:zookeeper

File Permission (octal)

600

Cloudera Management**Role: cloudera-mgmt-REPORTSMANAGER****Kerberos Principals**

hdfs

Filename (*.keytab)

headlamp

Keytab File Owner:Group

cloudera-scm:cloudera-scm

File Permission (octal)

600

Role: cloudera-mgmt-SERVICEMONITOR**Kerberos Principals**

hue

Filename (*.keytab)

cmon

Keytab File Owner:Group

cloudera-scm:cloudera-scm

File Permission (octal)

600

Role: cloudera-mgmt-ACTIVITYMONITOR**Kerberos Principals**

hue

Filename (*.keytab)

cmon

Keytab File Owner:Group

cloudera-scm:cloudera-scm

File Permission (octal)

600

Cloudera Manager**Kerberos Principals**

cloudera-scm, HTTP

Filename (*.keytab)

cmf, scm

Keytab File Owner:Group

cloudera-scm:cloudera-scm

File Permission (octal)

600

CruiseControl**Role:** cruise_control-CRUISE_CONTROL_SERVER**Kerberos Principals**

cruisecontrol, kafka, HTTP

Filename (*.keytab)

cruise_control

Keytab File Owner:Group

cruisecontrol:hadoop

File Permission (octal)

600

HttpFS**Role:** hdfs-HTTPFS**Kerberos Principals**

httpfs, HTTP

Filename (*.keytab)

httpfs

Keytab File Owner:Group

httpfs:httpfs

File Permission (octal)

600

Hue**Role:** hue-KT_RENEWER**Kerberos Principals**

hue

Filename (*.keytab)

hue

Keytab File Owner:Group

hue:hue

File Permission (octal)

600

Schema Registry**Role:** schemaregistry-SCHEMA_REGISTRY_SERVER**Kerberos Principals**

schemaregistry, HTTP

Filename (*.keytab)

schemaregistry

Keytab File Owner:Group

schemaregistry:hadoop

File Permission (octal)

600

Streams Messaging Manager**Role:** streams_messaging_manager-STREAMS_MESSAGING_MANAGER_SERVER

Kerberos Principals

streamsmgmr, HTTP

Filename (*.keytab)

streams_messaging_manager

Keytab File Owner:Group

streamsmgmr:streamsmgmr

File Permission (octal)

600

Streams Replication Manager**Role: streams_replication_manager-STREAMS_REPLICATION_MANAGER_DRIVER****Kerberos Principals**

streamsrepmgr

Filename (*.keytab)

streams_replication_manager

Keytab File Owner:Group

streamsrepmgr:streamsrepmgr

File Permission (octal)

600

Role: streams_replication_manager-STREAMS_REPLICATION_MANAGER_SERVICE**Kerberos Principals**

streamsrepmgr

Filename (*.keytab)

streams_replication_manager

Keytab File Owner:Group

streamsrepmgr:streamsrepmgr

File Permission (octal)

600

Create Service Principals and Keytab Files for HDP

This section is optional. During the HDP 3.1.5 to HDP intermediate bits upgrade, Ambari can generate the principals and keytabs. However, before upgrading, you can manually generate the principals and keytabs. First, create the principal using mandatory naming conventions and then create the keytab file with the principal's information. Lastly, copy the keytab file to the keytab directory on the appropriate service host.

To create a service principal, use the kadmin utility. The kadmin utility is a command-line driven utility where you can run Kerberos commands to manipulate the central database. To start kadmin, run the following commands:

1. 'kadmin \$USER/admin@REALM'
2. kadmin: addprinc -randkey \$principal_name/\$service-host-FQDN@\$hadoop.realm

**Note:**

- a. You must have a principal with administrative permissions to run the above commands.
- b. The randkey is used to generate the password.
- c. The \$principal_name part of the name must match the values in the table below.

In the example mentioned in step 2, each service principal's name is appended with a fully qualified domain name of the host on which the principal is running. This is to provide a unique principal name for services that run on multiple

hosts, like DataNodes and TaskTrackers. The addition of the hostname serves to distinguish, for example, a request from DataNode A from a request from DataNode B. This is important for two reasons:

- If the Kerberos credentials for one DataNode are compromised, it does not automatically compromise all other DataNodes.
- If multiple DataNodes have the same principal and are simultaneously connecting to the NameNode, and if the Kerberos authenticator sent has the same timestamp, then the authentication is rejected as a replay request.



Note: The NameNode, Secondary NameNode, and Oozie require two principals each.

If you are configuring High Availability (HA) for a Quorum-based NameNode, you must also generate a principle (jn/\$FQDN) and keytab (jn.service.keytab) for each JournalNode. JournalNode also requires the keytab for its HTTP service. If the JournalNode is deployed on the same host as a NameNode, the same keytab file (spnego.service.keytab) can be used for both. In addition, HA requires two NameNodes. Both the active and standby NameNodes require their own principal and keytab files. The service principals of the two NameNodes can share the same name, specified with the dfs.namenode.kerberos.principal property in hdfs-site.xml, but the NameNodes still have different fully qualified domain names.

Service	Component/Role	Principal Name	Mandatory Keytab Filename
HDFS	NameNode	nn/\$FQDN	nn.service.keytab
	SecondaryNameNode	nn/\$FQDN	nn.service.keytab
	DataNode	dn/\$FQDN	dn.service.keytab
	Journal Server*	jn/\$FQDN	jn.service.keytab
	NameNode HTTP	HTTP/\$FQDN	spnego.service.keytab
	SecondaryNameNode HTTP	HTTP/\$FQDN	spnego.service.keytab
MapReduce	MR2 History Server	jhs/\$FQDN	nm.service.keytab
	MR2 History Server HTTP	HTTP/\$FQDN	spnego.service.keytab
YARN	Node Manager	nm/\$FQDN	nm.service.keytab
	Resource Manager	rm/\$FQDN	rm.service.keytab
	YARN Timeline Server	yarn-ats/\$FQDN	yarn-ats.service.keytab
	HTTP	HTTP/\$FQDN	spnego.service.keytab
Oozie	Oozie Server	oozie/\$FQDN	oozie..service.keytab
	Oozie HTTP	HTTP/\$FQDN	spnego.service.keytab
Hive	HiverServer2, HMS	hive/\$FQDN	hive.service.keytab
	Hive HTTP	HTTP/\$FQDN	spnego.service.keytab
HBase	HBase Master Server	hbase/\$FQDN	hbase.service.keytab
	HBase RegionServer	hbase/\$FQDN	hbase.service.keytab
Kafka	Kafka Broker	kafka/\$FQDN	kafka.service.keytab
Zeppelin	Zeppelin Server	zeppelin/\$FQDN	zeppelin.service.keytab
Zookeeper		zookeeper/\$FQDN	zk.service.keytab
Knox		knox/\$FQDN	knox.service.keytab
Ranger	Admin Server	rangeradmin/\$FQDN	rangeradmin.service.keytab
	Lookup Server	rangerlookup/\$FQDN	rangerlookup.service.keytab
	KMS	rangerkms/\$FQDN	rangerkms.service.keytab
	UserSync	rangerusersync/\$FQDN	rangerusersync.service.keytab
	TagSync	rangertagsync/\$FQDN	rangertagsync.service.keytab

Service	Component/Role	Principal Name	Mandatory Keytab Filename
AMS		amshbase/\$FQDN	ams-hbase.master.keytab
		amsmon/\$FQDN	ams.collector.keytab
		amszk/\$FQDN	ams-zk.service.keytab
Spark2		spark/\$FQDN	spark.service.keytab
Druid		druid/\$FQDN	druid.service.keytab
Infra-Solr		infra-solr/\$FQDN	ambari-infra-solr.service.keytab
Atlas		atlas/\$FQDN	atlas.service.keytab
Livy		livy/\$FQDN	livy.service.keytab

* Only required if you are setting up NameNode HA. For example, to create the principal for a DataNode service, run the command `kadmin: addprinc -randkey dn/$datanode-host@$hadoop.realm`

3. Extract the related keytab file and place it in the keytab directory of the respective components. The default directory is `/etc/krb5.keytab`.

- `kadmin: xst -k $keytab_file_name $principal_name/fully.qualified.domain.name`

You must use the mandatory names for the `$keytab_file_name` variable shown in the table above. For example, to create the keytab files for the NameNode, run the command `kadmin: xst -k nn.service.keytab nn/$namenode-host`
`kadmin: xst -k spnego.service.keytab HTTP/$namenode-host`

After creating the keytab files, copy the keytab files to the keytab directory of the respective service hosts.

4. On each service in your cluster, verify that the correct keytab files and principals are associated with the correct service using the `klist` command. For example, on the NameNode, run the command `klist -k -t /etc/security/nn.service.keytab`

Upgrading the cluster's underlying OS

Ensure that all your hosts in the cluster are on the operating systems supported with the HDP intermediate bits and Ambari 7.1.x.x before starting the upgrade from HDP 3.1.5.x to HDP intermediate bits.

Only RHEL, CentOS, and Ubuntu operating systems are supported with the HDP intermediate bits and Ambari 7.1.x.x. Ensure that all your hosts in the cluster are on the supported operating system before starting the upgrade from HDP 3.1.5.x to HDP intermediate bits. For more information on the supported versions of Operating systems, see [Operating system requirements](#).



Note: SLES 12 SP5 is now supported for use with the HDP intermediate bits and CDP Private Cloud Base 7.1.4 and higher.

For many, this is a process that takes time and orchestration between multiple teams within your organization. Two high-level guidelines for moving from one major operating system version to another is as follows:

In-Place and Restore:

Perform an In-place OS refresh and use Ambari Restore Host feature

Move and Decom:

Move Masters and Decom/Recom Workers

Each option has advantages and disadvantages and high-level decision criteria.

In-Place and Restore

Review the activities involved in ensuring important metadata and data are stored on a volume that is not being used by the operating system, and leveraging component high availability to maintain maximum cluster availability before starting the upgrade to HDP intermediate bits.

This option should be used in medium to large clusters (25 or more nodes), with operational teams that have environment automation experience, and have followed best practices when setting up component High Availability and HDP directory structures (such as ensuring that the HDP component data and metadata are not stored on the root volume).

This option involves going through each host in the cluster and ensuring important metadata and data are stored on a volume that is not being used by the operating system, and leverages component high availability to maintain maximum cluster availability. When visiting each host, the host is shut down, the operating system volume is refreshed with the new version of the chosen operating system, the host is configured with the same IP address and hostname, all volumes are re-mounted, and the Ambari agent is installed and configured. After the host has rejoined the cluster, the Ambari Recover Host functionality is used to reinstall, reconfigure, and start services. To ensure that no data is lost during the reinstall of the operating system, verify that your OS volumes do not contain any HDP data or metadata. Additionally, during the OS reinstall, make sure that you do not erase or reformat any non-operating-system volumes, such as HDFS data drives, as this may result in data loss.

Move and Decommission

You have the option to replace worker nodes with new operating system and move master nodes to hosts with new operating system when operating teams either do not have access to the operating system or have not followed the best practices when setting up the HDP directory structures.

This option should be used in smaller clusters (under 25 nodes), where operational teams may not have access to operating system and configuration management automation tooling or have not yet followed best practices when setting up the HDP directory structures (such as ensuring HDP component data and metadata are not stored on the root volume).

This option involves decommissioning worker nodes and replacing them with worker nodes that have the new operating system version on them. For master nodes, the move-master operation is used to move all masters off of a host, and on to a new host with the new operating system version on them. Decommissioning worker nodes can take a great deal of time, depending on the density of the nodes, and move-master operations require many cluster services to be restarted, so this is a time-consuming process that requires multiple periods downtime, but it does not require any operating system level operations to be performed.

Versions and supported services for migration

The following table describes the CMA tool versions that support services (data and workload) for HDP to CDP migration.

CMA version	Data	Workload
3.0.0	HDFS and HMS tables	SQL
3.2.0	HBase tables	<ul style="list-style-type: none"> SQL with Discovery of Dependant tables Oozie with Hive action

Software download matrix for HDP 3.1.5 and 2.6.5 to CDP 7.1.x

All the download links related to HDP, Ambari, Cloudera Manager, CDP Private Cloud Base, and so on are available here.

Product	Download location	Note
Cloudera Manager	Cloudera Manager	
Cloudera Runtime	Cloudera Runtime	Includes parcels for Cloudera Runtime 7.1.x and the Sqoop connectors.

Product	Download location	Note
AM2CM 3.2.0.0 tool - This tool consists of log4j security updates.	CMA 3.2	This latest AM2CM tool supports the upgrades from: <ol style="list-style-type: none"> 1. HDP 3.1.5 to CDP 7.1.9 2. HDP 3.1.5 to CDP 7.1.7 SP2 3. HDP 3.1.5 to CDP 7.1.8 4. HDP 3.1.5 to CDP 7.1.7 SP1 5. HDP 2.6.5 to CDP 7.1.7 SP2 6. HDP 2.6.5 to CDP 7.1.8
AM2CM 2.8.1.0 tool - This tool consists of log4j security updates.	AM2CM 2.8.1.0	This latest AM2CM tool supports the upgrades from: <ol style="list-style-type: none"> 1. HDP 3.1.5 to CDP 7.1.9 2. HDP 3.1.5 to CDP 7.1.7 SP2 3. HDP 3.1.5 to CDP 7.1.8 4. HDP 3.1.5 to CDP 7.1.7 SP1 5. HDP 2.6.5 to CDP 7.1.7 SP2 6. HDP 2.6.5 to CDP 7.1.8
AM2CM 2.6.2.0 tool - This tool consists of log4j security updates.	AM2CM 2.6.2.0	This latest AM2CM tool supports the upgrades from: <ol style="list-style-type: none"> 1. HDP 3.1.5 to CDP 7.1.9 2. HDP 2.6.5 to CDP 7.1.7 SP2 3. HDP 2.6.5 to CDP 7.1.8
AM2CM 2.6.0.0 - This tool consists of log4j security updates.	AM2CM 2.6.0.0	The upgrade path is HDP 3.1.5 to CDP 7.1.7 SP2
	AM2CM 2.6.0.0	The upgrade path is HDP 3.1.5 to CDP 7.1.8
AM2CM 2.4.3 tool - This tool consists of log4j security updates.	AM2CM 2.4.3.0	The upgrade path is HDP 3.1.5 to CDP 7.1.7 SP1

Sample data ingestion

Cloudera recommends you use a subset of your workload or any sample data for any jobs or queries and do a benchmarking. You can record the time taken for the critical jobs and compare the performance of pre and post upgrade setup.

Cloudera recommends you use a subset of your workload or any sample data for any jobs or queries and do a benchmarking if required. You can record the time taken for the critical jobs and compare the performance of pre and post upgrade setup.

Merge Independent Hive and Spark Catalogs

If you upgraded to HDP 3.1.5 from an earlier version of HDP 3.x and did not convert independent catalogs to a shared catalog, you must do this before migrating tables to CDP.

About this task

In HDP 3.0 - 3.1.4, Spark and Hive use independent catalogs for accessing tables created using SparkSQL or Hive tables. A table created from Spark resides in the Spark catalog. A table created from Hive resides in the Hive catalog. Databases fall under the catalog namespace, similar to how tables belong to a database namespace. In HPD 3.1.5, Spark and Hive share a catalog in Hive metastore (HMS) instead of using separate catalogs.

The Apache Hive schematool in HDP 3.1.5 and CDP releases supports the mergeCatalog task. This task performs the following actions:

- Detects conflicts in DB names across catalogs and in case of conflicts, lists each conflict, and exits.
- When there are no conflicts, the following changes occur:
 - Adds EXTERNAL=true to the table properties of all managed tables in the source catalog.
 - Adds external.table.purge=true in table properties of all managed tables in the source catalog.
 - Sets tableType=EXTERNAL for all managed tables in the source catalog.
 - Sets CTLG_NAME=<toCatalog> for all databases in the source catalog.

Use the following syntax to merge the databases in the catalog named spark into a catalog named hive, which is the default catalog for HiveServer (HS2).

```
schematool -dbType <database> -mergeCatalog spark -toCatalog hive [-verbose]
[-dryRun]
```

The default names of the catalogs are spark and hive. The dryRun option rolls back the changes.

To merge catalogs:

Procedure

1. On the operating system command line, run a Hive schematool query test, using the dryRun option to roll back changes. For example:

```
bin/schematool -mergeCatalog spark -toCatalog hive -verbose -dbType mysql
--dryRun
```

2. Check the output, and if there are no conflicts, merge catalogs. For example:

```
bin/schematool -mergeCatalog spark -toCatalog hive -verbose -dbType mysql
```

Cloudera Manager Installation and Setup

Install Cloudera Manager, install Cloudera Manager agent and daemons, add Cloudera Management service, and finally configure clusters to use Kerberos. Kerberos and TLS will be added by Cloudera Migration Assistant (CMA).

About this task

Procedure

1. Prepare to install and configure the Cloudera Manager packages. For more information, see [Configuring Repository](#). Do this if you have not done it already. Confirm that the repo is set up.
2. Install Cloudera Manager Server. For more information on installing Cloudera Manager Server, see [Installing Cloudera Manager](#).



Note: During the upgrade process, you can place Cloudera Manager and its related Cloudera Manager Services components on the same node as Ambari. Ensure that the node has sufficient capacity to temporarily run Cloudera Manager and Ambari in parallel. If you do not wish to colocate these services, you can place them on separate management nodes in the cluster.

3. Preconfigure the databases for:

- Ranger
- Cloudera Manager Server
- Cloudera Management Service roles - Reports Manager
- Data Analytics Studio (DAS) - Supported with PostgreSQL only (applicable while upgrading to 7.1.8 or lower)
- Hue
- Each Hive metastore
- Oozie
- Schema Registry
- Streams Messaging Manager

For more information, see [Setup Cloudera Manager database](#) and [Install and Configure Databases](#).

4. Install Agent on all hosts in the cluster. It is possible to add hosts to Cloudera Manager using the [Installation Wizard](#).
5. Start Cloudera Manager Server and Cloudera Manager agent on all hosts. For more information, see [Cloudera Manager Agent](#) and [Cloudera Manager Server](#).
6. Install Cloudera Manager User licence. For more information, see [Installation Wizard](#). (Upload the license file and

exit the cluster setup by clicking the Cloudera Manager icon



Caution:

- Do not set up a cluster using the Wizard (Step 7) .
- Do not proceed to Welcome (Add Cluster - Installation).

7. Add Hosts to a cluster. To add hosts to a cluster, see [Adding Hosts to a cluster](#).
8. Add Cloudera Manager management service to the cluster. To add services to the cluster, see [Select Services](#). For more information see the [Adding Cloudera Management services](#) documentation.



Note: Some of the services in Cloudera Manager management service may not come up due to port conflicts in HDP. The respective service needs to be stopped in the Ambari-manager HDP cluster to workaround the issue. For more information, see [Ports collisions](#).

9. The AM2CM tool migrates service principal names from the service user names in the HDP cluster. If the HDP cluster has default service usernames then Cloudera Manager is configured with default principal names. For example hdfs, yarn, and hive. If the HDP cluster contains user names like cstm-hdfs and cstm-hive, then Cloudera Manager is configured with the principal names with same names. For more information, see [Hadoop Users \(user:group\) and Kerberos Principals](#).

Installing Cloudera Management Service

You can automate the installation of Cloudera Management Service through the command line or using the following steps. [Installing Cloudera Management Service](#) is part of step 2 in [Cloudera Manager Installation and Setup](#).

Port Collisions

Ports from the installed Cloudera Manager roles can conflict with your source cluster service ports. You can change or stop the ports in conflict.

For example, if there are default service ports for HDP Services configured in step 8 of [Installing Cloudera Management Service](#), then you must stop the following services:

- SMARTSENSE
- Druid

- Zeppelin

Stopping the services through Ambari REST API:

```
curl -X PUT -u admin:admin_pwd -H "X-Requested-By: ambari" -k "https://your-ambari-server.com:8443/api/v1/clusters/cl1/services/SMARTSENSE" -d "{\"RequestInfo\": {\"context\": \"Stopping Services that collide with Cloudera Management Service Roles\"}, \"Body\": {\"ServiceInfo\": {\"state\": \"INSTALLED\"}}}"
```

Getting CM API version

If you want to use Cloudera Manager 7.11.3, you must use v49 in the following commands and replace host, user, name, and password.

```
curl http://admin:admin@your-cm-server.com:7180/api/[**version**]
```

Creating the Cloudera Management Service (CMS)

```
curl -X PUT -u admin:admin_pwd "http://your-cm-server.com:7180/api/v49/cm/service" -H "Content-Type: application/json" -d "{\"displayName\": \"Cloudera Management Service\"}"
```

Auto-assigning the roles

```
curl -X PUT -u admin:admin_pwd "http://your-cm-server.com:7180/api/v49/cm/service/autoAssignRoles" -H "accept: application/json"
```

Auto-configuring the CMS roles

```
curl -X PUT -u admin:admin_pwd "http://your-cm-server.com:7180/api/v49/cm/service/autoConfigure" -H "accept: application/json"
```

The Cloudera Manager versions 7.7.1 and higher have different default Reports Monitor database configurations described in [Install and Configure MySQL for Cloudera Software](#) documentation (refer to rman). This can make the rman user unable to connect to its database.

You must update the following parameters:

- headlamp_database_host = <your DB hostname>
- headlamp_database_name = <your reports manager DB name, rman as default>
- headlamp_database_user = <you reports manager user name, rman as default>
- headlamp_database_password = <your reports manager pwd>

For updating the parameters, you must create a json file. For example, rman_db.json and update the values of parameters.

```
{ "items": [ { "name": "headlamp_database_host", "value": "your-DB-server.com" }, { "name": "headlamp_database_name", "value": "rman_db" }, { "name": "headlamp_database_user", "value": "rman_user" }, { "name": "headlamp_database_password", "value": "rman_pwd" } ] }
```

Updating curl to the Cloudera Manager server:

```
curl -u admin:admin_pwd -X POST "http://your-cm-server.com:7180/api/v49/cm/service/roleConfigGroups/mgmt-REPORTSMANAGER-BASE/config" -d @rman_db.json
```

Review if Cloudera Manager has the payroll credentials configured. In Cloudera Manager, navigate to Parcel > Parcel Repository & Network Settings and check:

- remote_repo_override_user
- remote_repo_override_password

You can also review the paypal credentials through Rest API call by posting the payload (repo_access.json) to Cloudera Manager server: "items": [{ "name": "remote_repo_override_user", "value": "your cloudera repo user name" }, { "name": "headlamp_database_password", "value": "your cloudera repo password" }] } curl -u admin:admin_pwd -X POST "http://your-cm-server.com:7180/api/v49/cm/config" -d @repo_access.json

Starting Cloudera Management service

```
curl -u admin:admin_pwd -X POST "http://your-cm-server.com:7180/api/v49/cm/service/commands/restart" -H "accept: application/json"
```

Setting up CMA server

You must create a docker image to run the Clouder Migration Assistant server or install it locally. You can deploy it to any of your cluster nodes or an external node that has visibility to the cluster and has at least 1.5 GB of extra memory. `${cloudera.version}` here represents the CMA versions such as 3.2.0.0-21 and so on.

Before you begin

To identify the exact filename of the archive, open [https://archive.cloudera.com/cma/\\${cloudera.version}/](https://archive.cloudera.com/cma/${cloudera.version}/) and see the archive file available.

CMA without internet connection

When setting up CMA without internet connection, the installation script ensures to install the required Python dependencies without using internet connection and creates the Python Package Index (pypi) repository locally.

You can view the list of components installed with CMA under the following directory: `cma-${cloudera.version}/am2cm-ansible/python_requirements/`

The Python requirements file details the Python packages that are needed to set up the virtual environment to run CMA. No internet connection is used to download these components when setting up CMA in an air-gapped network.



Note: Even though the installation of CMA can be completed without internet connection, you need to ensure that you have internet connection when downloading the JDBC drivers and Atlas artifacts.

Downloading CMA

Download the latest CMA binaries from <https://archive.cloudera.com/cma/3.2/>. The supported version of CMA is 2.4.1.1 and higher. For more information on the CMA versions, see [Software download matrix](#)

- With internet connection:

1. `wget https://archive.cloudera.com/cma/${base.version}/tars/cma-${cloudera.version}-bin.tar.gz` or `curl https://archive.cloudera.com/cma/${base.version}/tars/cma-${cloudera.version}-bin.tar.gz --output cma-${cloudera.version}-bin.tar.gz`
2. Extract the downloaded file: `tar xzf cma-${cloudera.version}-bin.tar.gz`

- Without internet connection:

```
1. mkdir cma-${cloudera.version}
2. cd cma-${cloudera.version}
3. wget https://archive.cloudera.com/cma/${base.version}/tars/cma-
   ${cloudera.version}-bin.tar.gz

   wget https://archive.cloudera.com/cma/${base.version}/tars/cma-extras-gpl-
   ${cloudera.version}-bin.tar.gz

or

curl https://archive.cloudera.com/cma/${base.version}/tars/cma-
   ${cloudera.version}-bin.tar.gz --output cma-${cloudera.version}-bin.tar.gz

curl https://archive.cloudera.com/cma/${base.version}/tars/cma-extras-gpl-
   ${cloudera.version}-bin.tar.gz --output cma-${cloudera.version}-bin.tar.gz
```

When the required binaries for air-gapped install are successfully downloaded, the directory structure looks like the following example:

```
drwxr-xr-x 14 testuser testuser 4096 febr 27 13:21 cma-3.2.0.0-14\
-rw-rw-r-- 1 testuser testuser 518140466 febr 27 13:28 cma-3.2.0.0-14-
bin.tar.gz
-rw-rw-r-- 1 testuser testuser 85089637 febr 27 13:28 cma-extras-
gpl-3.2.0.0-14.tar.gz
```

Extract the downloaded file: `tar xzf cma-${cloudera.version}-bin.tar.gz`

Starting CMA server

After extracting the compressed file, there are two ways to start the CMA server - in a Docker container or locally.

Starting CMA server in Docker Container

Ensure that Docker 20+ is installed on the host. For more information, see the [Installing Docker Engine](#) documentation.

In case you do not define the Python executable when running the script, you will be prompted to enter the Python executable path.



Note: The script will automatically create the Docker image if necessary. Additionally, the script provides the following operations to manage the CMA Docker container: start, stop, restart, or rebuild. If you want to explore other available options, run the following command: `cma-${cloudera.version}/bin/cma-docker.sh --help`.

- With internet connection:

Run the `cma-docker.sh` script in the untarred top-level folder to launch the CMA server in a Docker container: `cma-${cloudera.version}/bin/cma-docker.sh --start`

- Without internet connection:

```
cd ${cloudera.version}/bin/cma-docker.sh --start --airgapped --python-
executable=python3
```

If the GPL file is not located in the same directory as the CMA file, you can use the following command, where you define the path of the GPL file: `cma-${cloudera.version}/bin/cma-docker.sh --start --airgapped --cma-extras-gpl-tar-location=`<ABSOLUTE PATH TO EXTRAS GPL>``

Check that the local pypi repository is installed correctly: `netstat -atnp | grep 9003` (Not all processes could be identified, non-owned process info will not be shown, you would have to be root to see it all.) `tcp 0 0 0.0.0.0:9003 0.0.0.0:* LISTEN 201503/python3`

Starting CMA server locally

The preferred way of running CMA server is using docker, however if you have constraints on Docker you can install it directly on any host. Ensure that Python 3.8 or later and Java runtime version 11 are installed on the host.



Note: The script will create a Python virtual environment in the top-level folder where the dependencies will be installed. Additionally, the script provides the following operations to manage the CMA locally: start, stop, restart, or rebuild. If you want to explore other available options, run the following command: `cma-${cloudera.version}/bin/cma-local.sh --help`.

- With internet connection:

Run the `cma-local.sh` script in the untarred top-level folder and follow its instructions to launch the CMA server locally: `cma-${cloudera.version}/bin/cma-local.sh --start`

- Without internet connection:

```
cd ${cloudera.version}/bin/cma-local.sh --start --airgapped --python-executable=python3
```

If the GPL file is not located in the same directory as the CMA file, you can use the following command, where you define the path of the GPL file: `cma-${cloudera.version}/bin/cma-local.sh --start --airgapped --cma-extras-gpl-tar-location=<ABSOLUTE PATH TO EXTRAS GPL>`

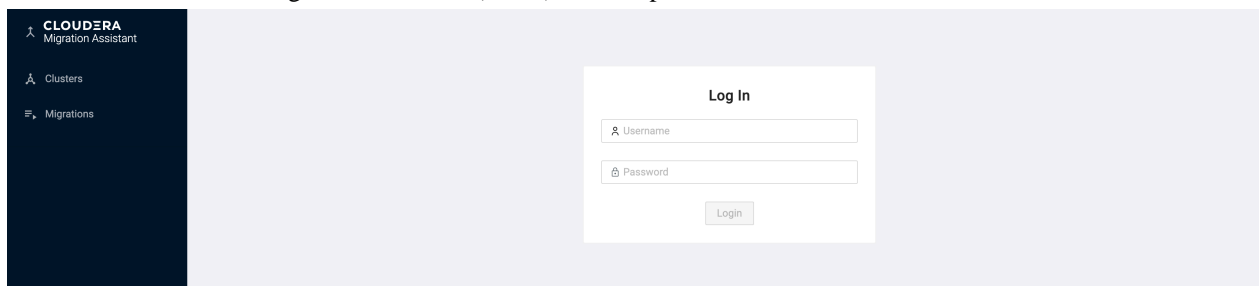
Check that the local pypi repository is installed correctly: `netstat -atnp | grep 9003` (Not all processes could be identified, non-owned process info will not be shown, you would have to be root to see it all.) `tcp 0 0 0.0.0.0:9003 0.0.0.0:* LISTEN 201503/python3`

After startup

After the CMA server starts, open

```
http://localhost:8090
```

in a browser. Cloudera Migration Assistant (CMA) Server opens.



This confirms that Cloudera Migration Assistant (CMA) Server is successfully installed.

The default username is admin and password is admin. However, you can change your password on the user profile page.

The screenshot displays the Cloudera Migration Assistant (CMA) registration wizard. The interface consists of a dark sidebar on the left and a main content area on the right. The sidebar contains the following elements: the Cloudera logo and 'Migration Assistant' text, a 'Clusters' link with a cluster icon, a 'Migrations' link with a migration icon, and at the bottom, an 'Admin' link with a user icon and a 'Logout' link with a door icon. The main content area contains a registration form with the following fields: 'Username' (containing 'admin'), 'Display name' (containing 'Admin'), '* Password' (with a toggle icon), '* New Password' (with a toggle icon), and '* Confirm Password' (with a toggle icon). A 'Change Password' button is located below the password fields.

You also need to make sure you that CMA host has a line of sight to the cluster nodes, by configuring `/etc/hosts` or `/etc/resolv.conf` files.

Registering Ambari Cloudera Manager pair for source cluster

As a prerequisite, you must have an instance of Cloudera manager available and running on a cluster.

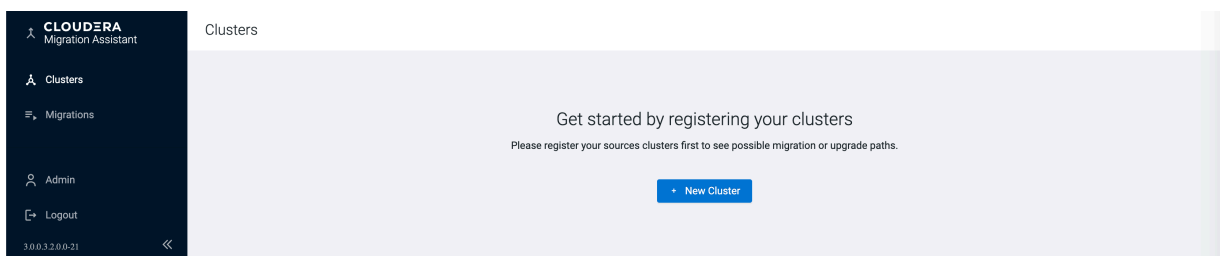
About this task

One transition represents a single upgrade path from a base cluster to a target cluster. A single Cloudera Migration Assistant (CMA) server is designed to handle multiple upgrades of Ambari-Cloudera Manager Pairs.

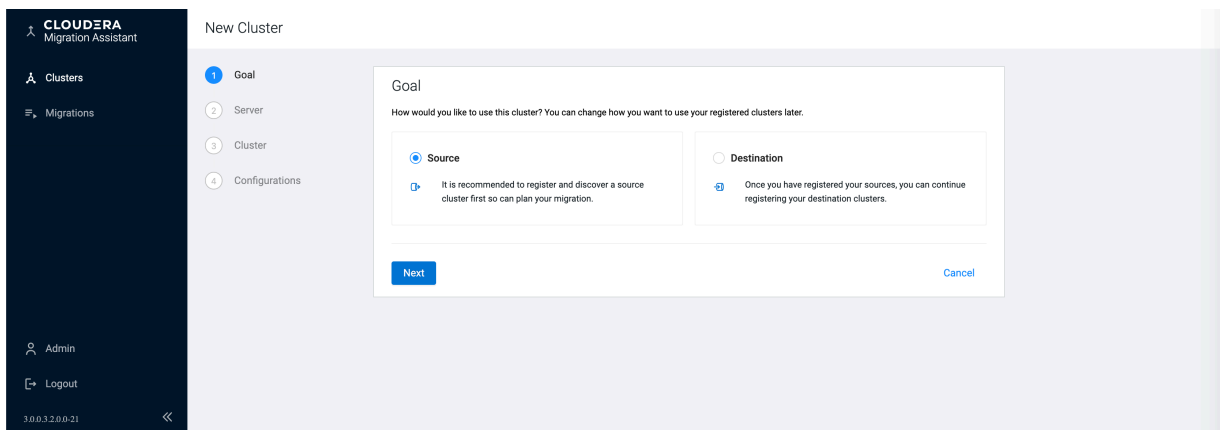
The process starts with a registration wizard, gathering all the cluster data into the data directory, and then continues with the execution phase where the collected data is used by Ansible scripts carrying out the required upgrade steps.

Procedure

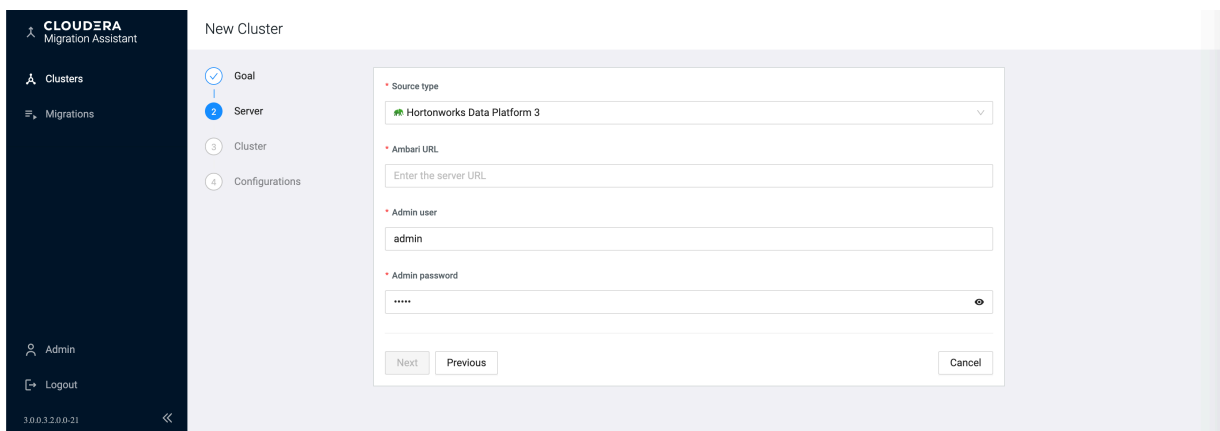
1. On the left navigation pane, click New Cluster.



2. Select Source. Click Next.



3. Select the cluster type > enter the Ambari URL > admin user and admin password. The admin user can be replaced with any other user who has the right to export blueprint, stop, and start services. Click Next.



4. Select the cluster you want to upgrade.

5. Select the Configuration based on the authentication method for your cluster.

- Select Use existing if you want to use the SSH configuration and keys from the existing CMA server to access the hosts.
- Select New if you want to use a newly provided SSH key to configure the Ansible automatically.
 - a. Provide the SSH User.
 - b. Provide the SSH Port.
 - c. Copy the Private SSH Key to the SSH Key box or upload a key file containing the key by clicking Choose File.
 - d. Click Create

After the registration is successful, the HDP cluster registered under Clusters.

Platform	Name	Server	Type	Actions
Hortonworks Data Platform 3	cd1	https://ctr-e19-1701672056237-108764-01-000002.comops.site:8443	Source	

Registering Ambari Cloudera Manager pair for target cluster

This section helps you to register the Ambari Cloudera Manager pair for the target cluster which is the CDP Private Cloud Base.

Before you begin

You must have Cloudera Manager with Cloudera Management Service installed and running. For more information, see the [Installing Cloudera Manager](#) and [Cloudera Manager Service](#) documentation.

Procedure

1. On the left navigation pane, click **New Cluster Destination Next**.

CLUSTER Migration Assistant

New Cluster

1 Goal

2 Server

3 Cluster

4 Configurations

Admin

Logout

3.0.0.3.2.1.0-14

Goal

How would you like to use this cluster? You can change how you want to use your registered clusters later.

☐ Source

☒ Destination

It is recommended to register and discover a source cluster first so you can plan your migration.

Once you have registered your sources, you can continue registering your destination clusters.

Next Cancel

2. Provide a name to the new CDP Private Cloud Base cluster that will be created by CMA in Cloudera Manager during the upgrade (migration) execution.

CLUSTER Migration Assistant

New Cluster

1 Goal

2 Server

3 Cluster

4 Configurations

Admin

Logout

3.0.0.3.2.1.0-14

Server

* Target Type

Cloudera Data Platform - Private Cloud Base

* Cloudera Manager URL

http://ctr-e19-1701672056237-108764-01-000002.comops.site:7180

* Admin User

admin

* Admin Password

Next Previous Cancel

3. The access credentials for the target cluster is the same as the source cluster.

CLUSTER Migration Assistant

New Cluster

1 Goal

2 Server

3 Cluster

4 Configurations

Admin

Logout

3.0.0.3.2.1.0-14

Cluster

* Cluster Name

MyNewCDP_PvC_Base

* Source

https://ctr-e19-1701672056237-108764-01-000002.comops.site:8443 - c1

Next Previous Cancel

4. In the Configurations page, settings from the source cluster are copied to the target cluster.

CLUSTER Migration Assistant

Clusters

Search

+ New

Platform	Name	Server	Type	Actions
Hortonworks Data Platform 3	c1	https://ctr-e19-1701672056237-108764-01-000002.comops.site:8443	Source	
Cloudera Data Platform - Private Cloud Base	MyNewCDP_PvC_Base	http://ctr-e19-1701672056237-108764-01-000002.comops.site:7180	Destination	

< 1 > 10 / page

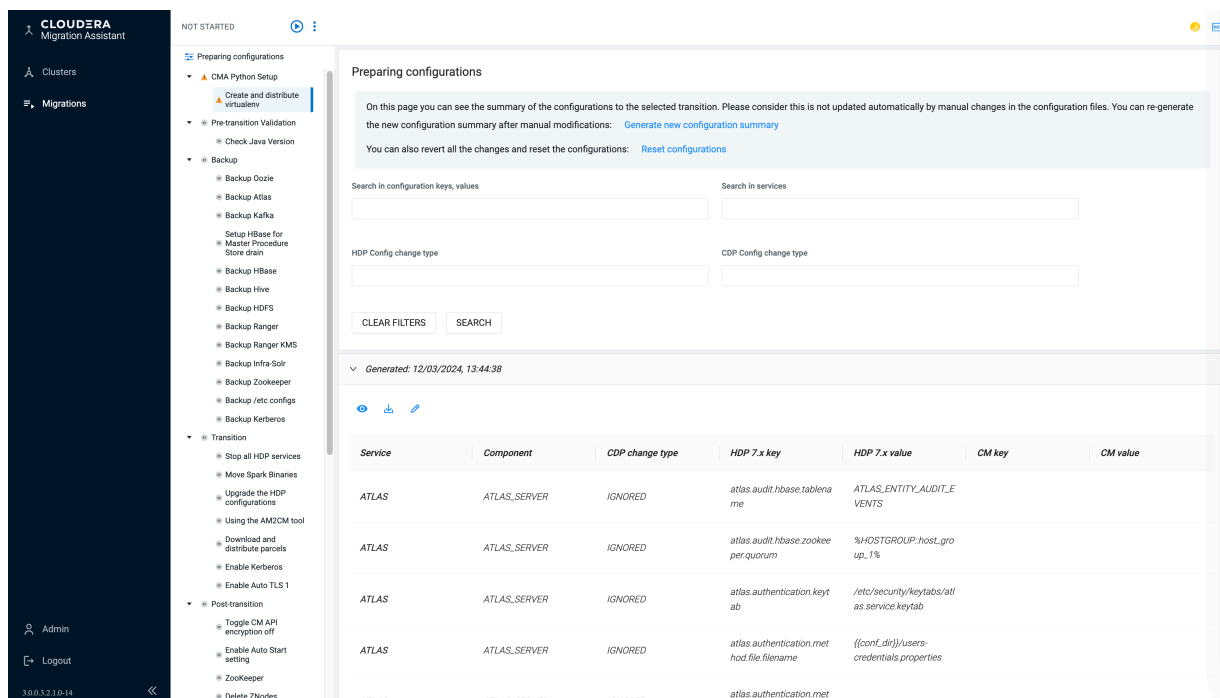
After the registration is successful, the CDP cluster registered under Clusters.

Preparing configurations

This section helps you in understanding how the Ambari parameters are mapped to the new cluster on Cloudera Manager, analyze the differences, and update the mappings accordingly.

Procedure

- In the Preparing Configurations page, you can review and compare the configuration key-value pairs in the current cluster versus the new CDP Private Cloud Base cluster.



Preparing configurations

On this page you can see the summary of the configurations to the selected transition. Please consider this is not updated automatically by manual changes in the configuration files. You can re-generate the new configuration summary after manual modifications: [Generate new configuration summary](#)

You can also revert all the changes and reset the configurations: [Reset configurations](#)

Search in configuration keys, values: Search in services:

HDP Config change type: CDP Config change type:

[CLEAR FILTERS](#) [SEARCH](#)

Generated: 12/03/2024, 13:44:38

Service	Component	CDP change type	HDP 7.x key	HDP 7.x value	CM key	CM value
ATLAS	ATLAS_SERVER	IGNORED	atlas.audit.hbase.tablename	ATLAS_ENTITY_AUDIT_EVENTS		
ATLAS	ATLAS_SERVER	IGNORED	atlas.audit.hbase.zookeeper.quorum	%HOSTGROUP-host_group.1%		
ATLAS	ATLAS_SERVER	IGNORED	atlas.authentication.keytab	/etc/security/keytabs/atlas.service.keytab		
ATLAS	ATLAS_SERVER	IGNORED	atlas.authentication.metastore.hod.file.filename	{{conf_dir}}/users-credentials.properties		
ATLAS	ATLAS_SERVER	IGNORED	atlas.authentication.metastore	true		

- You can edit the mapping rules and generate a new configuration summary with changes. You can try multiple settings before mapping the HDP parameters to CDP. You can export the configurations to a CSV file, share with others, and reset the mapping rules to default and start again.
- The Export configuration files option helps you to download a zip file of the migration ruleset. You can also import an old ruleset with the Import configuration files function.

- If the default mapping behavior is not accepted, you can edit the mapping using the icon. CDP Change Type describes how the HDP configuration parameters are mapped to the CDP Private Cloud Base configuration parameters. For example,
 - ADDITIONAL-PARAM:** Newly introduced parameter in the CDP Private Cloud Base cluster. This parameter does not have an HDP equivalent parameter.
 - IGNORED:** This parameter is not mapped to CDP Private Cloud Base cluster.
 - IGNORED-DEFAULT:** CMA considers the default value on HDP cluster and sets the default CDP value on the CDP cluster. For example, the default port value is 7085 for ranger.service.shutdown.port. Accordingly, the default port value is updated on the CDP cluster.
 - MAPPED:** HDP key/value pair are mapped to a different CDP key/value pair
 - NO-SAFETY-VALVE-IGNORED:** Configurations without defined mapping or safety valves are ignored.
 - SAFETY-VALVE(safety_valve_type):** This HDP configuration is migrated to the CDP safety valve configuration. For more information, see [Cloudera Manager Custom Configuration](#).

The screenshot displays the Cloudera Migration Assistant interface. On the left, a sidebar lists migration steps under the 'Migrations' tab. The main panel, titled 'Preparing configurations', provides a summary of the configurations to be migrated. It includes search bars for configuration keys and HBASE, and fields for HDP and CDP configuration change types. Below this, a table lists the generated configurations for HBASE services.

Service	Component	CDP change type	HDP 7.x key	HDP 7.x value	CM key	CM value
HBASE	MASTER	MAPPED	hbase.master.info.port	30010	hbase_master_info_port	30010
HBASE	MASTER	MAPPED	hbase.master.port	20000	hbase_master_port	20000
HBASE	REGIONSERVER	MAPPED	hbase.regionserver.info.port	20030	hbase_regionserver_info_port	20030
HBASE	REGIONSERVER	MAPPED	hbase.regionserver.port	16020	hbase_regionserver_port	16020

- The radio buttons help you to add or remove a configuration from the IGNORED list. If you change the configuration, the row color changes to highlight the changed configurations. You can also add new configurations to the mapping.
- You can add a new additional parameter or a parameter to a safety valve by selecting the CDP change type.
- After completing the review and editing all the parameters, click Finalize Configurations Changes to generate a new configuration summary with the new ruleset. This process can be performed multiple times and the am2cm tool will use the last version of the ruleset.

HDP to CDP Private Cloud Base Upgrade

This section helps you to from HDP to CDP Private Cloud Base.

Procedure

1. Add the registered source cluster details. Click Next

The screenshot shows the 'Source' configuration step in the Cloudera Migration Assistant. The left sidebar contains navigation links: Cloudera Migration Assistant, Clusters, Migrations, Admin, and Logout. The main panel has a progress bar with steps 1 to 5. Step 1, 'Source', is active. It contains a 'Source type' dropdown menu set to 'Hortonworks Data Platform 3' and a 'Source' text field with the URL 'https://ctr-e19-1701672056237-108764-01-000002.comops.site:8443 - cl1'. A 'New Source' link is to the right of the text field. A 'Next' button is at the bottom.

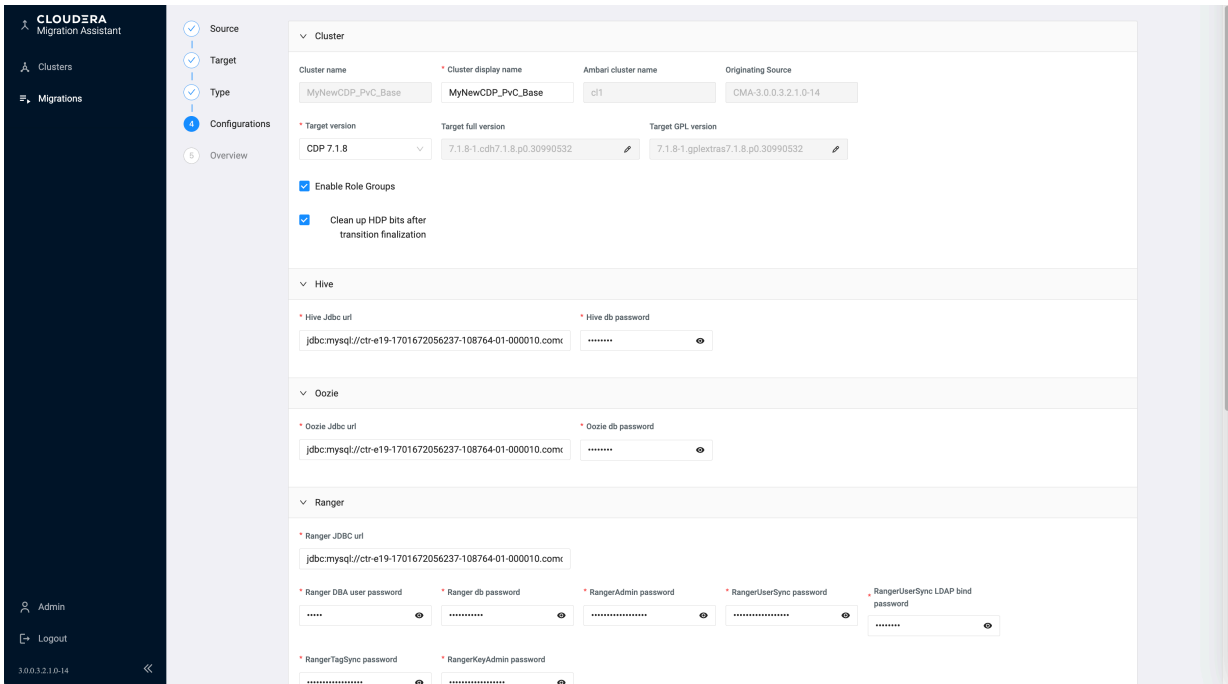
2. Add the registered target cluster details. Click Next

The screenshot shows the 'Target' configuration step. The 'Target type' dropdown is set to 'Cloudera Data Platform - Private Cloud Base'. The 'Target' text field contains 'MyNewCDP_PvC_Base', with a 'New Target' link to its right. 'Next' and 'Back' buttons are at the bottom.

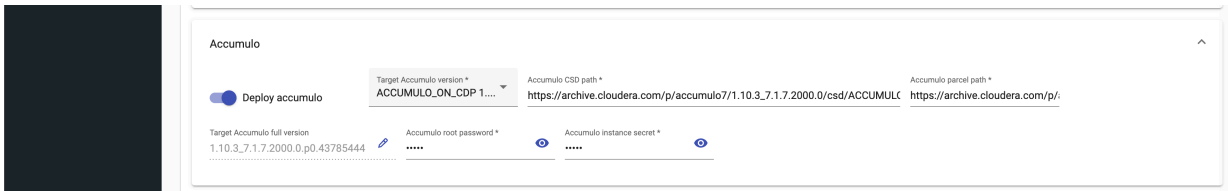
3. The target cluster type is CDP Private Cloudera Base. Click Next

The screenshot shows the 'Type' step. It features a diagram titled 'Upgrade to CDP PvC Base - 1 stage' showing an arrow from a server icon to a cloud icon labeled 'CDP'. Below the diagram, text describes an in-place upgrade of the base HDP 2/3 cluster. 'Next' and 'Back' buttons are at the bottom.

4. Configure the CDP Private Cloud Base Runtime parcel and credentials.



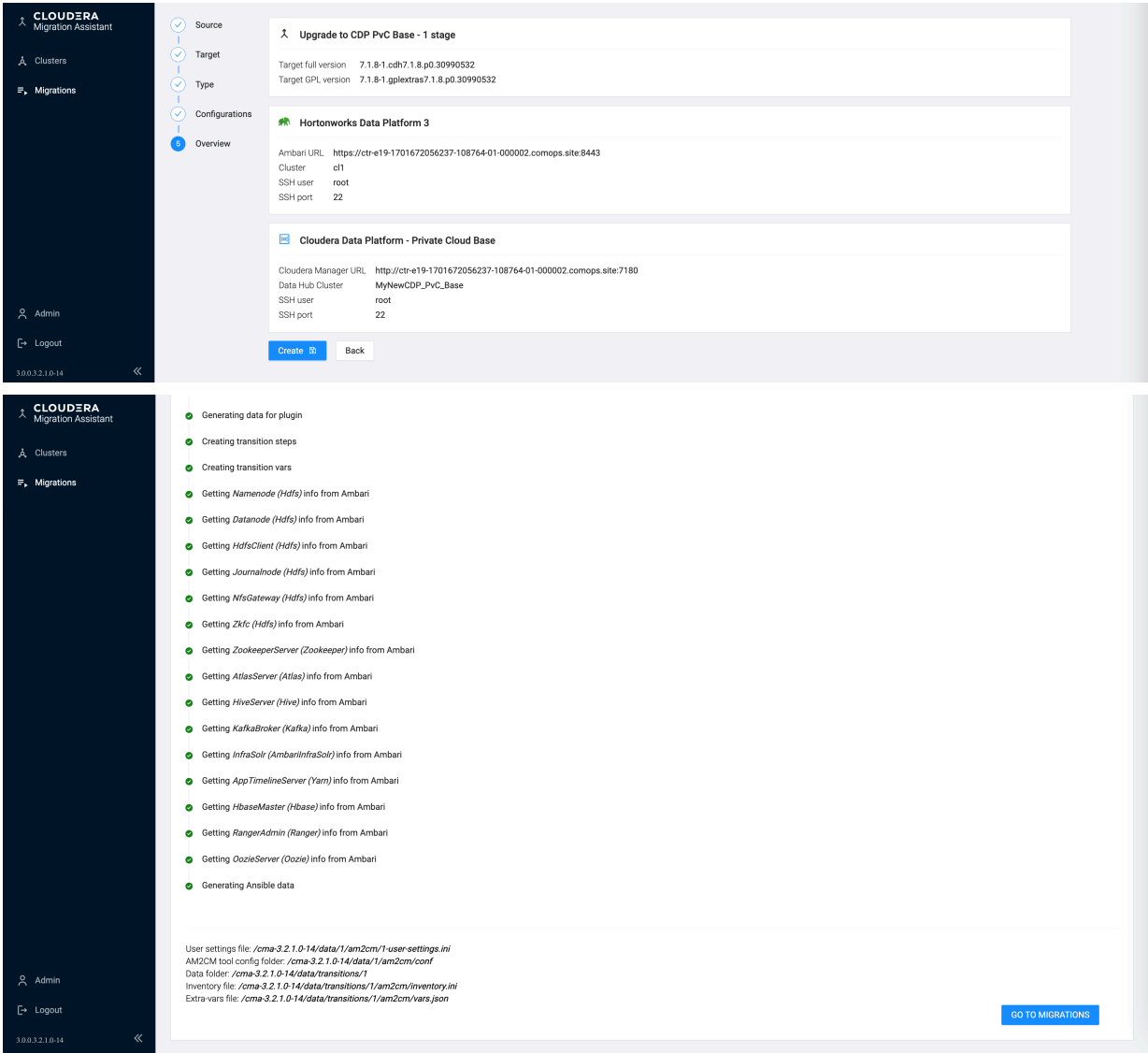
5. You can select the CDP Private Cloud Base Runtime parcel from the major released versions or you can edit the full target version specifying exact hotfix version numbers. Ensure that the Cloudera Manager version you manually installed must match the CDP Runtime version you selected.
- As CMA can not access the service specific passwords on the HDP cluster, you must provide them. You can collect it in advance.
 - If the source cluster has the Accumulo service installed, you can migrate the data to Accumulo on the target cluster. Since Accumulo is not part of the CDP Runtime parcel, you must install ACCUMULO_ON_CDP parcel and toggle Deploy Accumulo switch as shown in the below image.



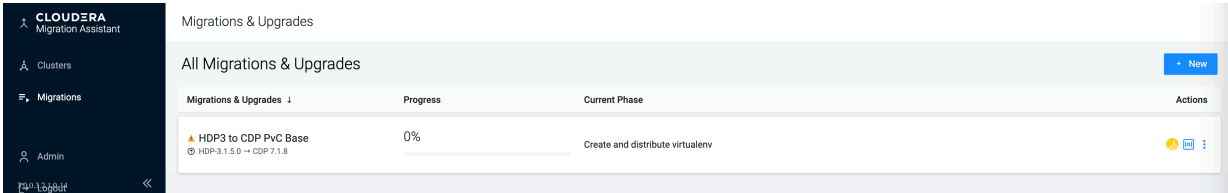
Note: See [Supported Platforms](#) to check the platform pairs where the Accumulo upgrade is supported.

Click Next

6. Click Create to complete the registration.



7. On the Migration tab, you can review the progress.



Note: The transition data is stored in <\$AM2CM_ROOT>/data/transtions/ directory. The following files are important for future references:

- *-var.json: Parameters collected during the registration process. This external VARS file is passed on to the ansible scripts.
- *-inventory.ini: The hosts and their roles mapped to an ansible inventory file.


Execution steps


You must prepare all the services for transition.

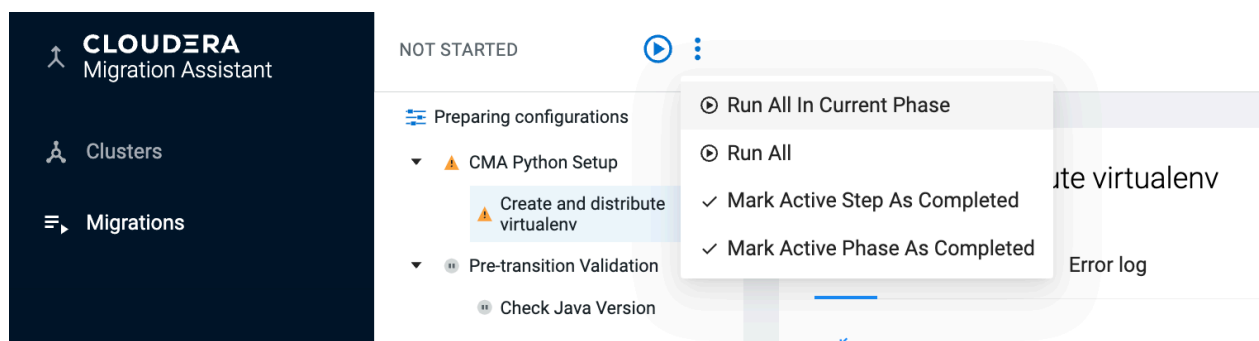
Group transition steps

On the left navigation pane, the migration steps are grouped by phases awaiting execution. These phases are differentiated by their effect on the production cluster.

Ansible scripts are run on the control node. However, most tasks are run on the cluster nodes also. As a consequence, CMA requires Python and some packages (see <CMA_ROOT_DIR>/am2cm-ansible/requirements.txt) on all nodes to work.

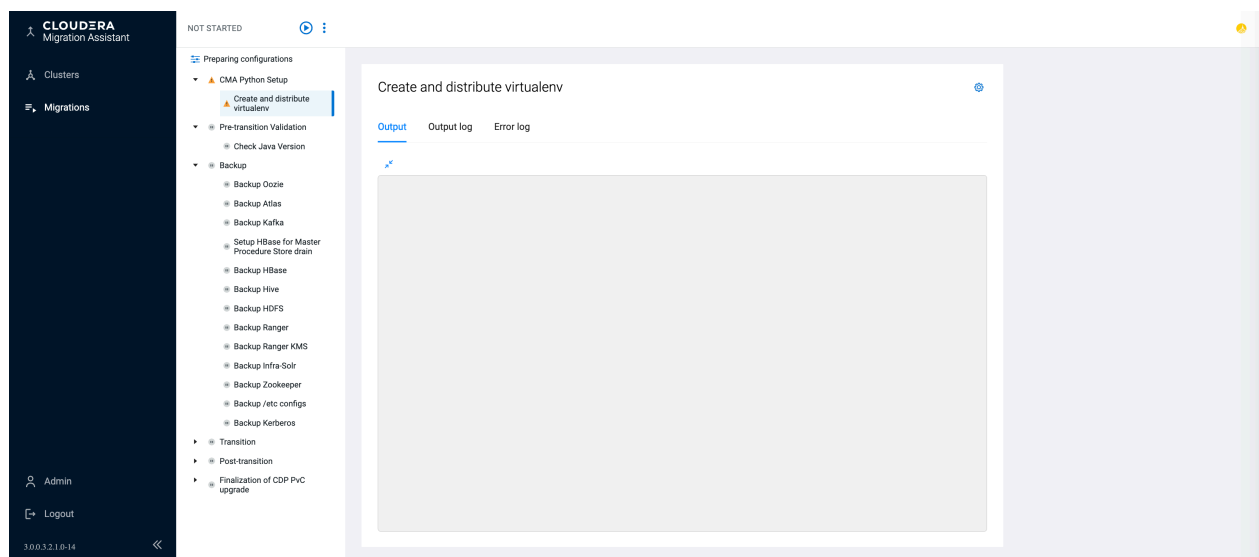
- To start the upgrade process, click .

- To run multiple tasks in phases, click . You can click Run All in Current Phase or Run All. However, Cloudera recommends you to use Run All on dev clusters only.



Note: In some cases, some of the steps must be performed manually. In such a case, skip the steps by marking them completed and proceed.

The CMA Python setup creates temporary Python 3.9.8 virtual environments on all nodes, unless there is already a higher version installed. The cluster nodes can be on an air gap environment. However, the control node needs access to the Internet.



You can run the Pre-transition Validation steps anytime and does not disturb the cluster. Performing the steps in this phase ensures that your cluster is ready for upgrade.

The screenshot shows the Cloudera Migration Assistant interface. On the left, a sidebar lists various backup steps under the 'Backup' section, including Backup Oozie, Backup Atlas, Backup Kafka, Setup HBase for Master Procedure Store drain, Backup HBase, Backup Hive, Backup HDFS, Backup Ranger, Backup Ranger KMS, Backup Infra-Solr, Backup Zookeeper, Backup etc configs, and Backup Kerberos. The main panel displays the 'Create and distribute virtualenv' task with its output log, showing successful execution of commands to create and distribute virtualenv across multiple hosts.



Note: You must ensure that you have checked the cluster environment readiness. For more information, see the [Cluster environment readiness](#) documentation.

To take the backup, the cluster must be in the maintenance mode even if it does not change payload data. If you have additional (non-HDP) services and application, then you need additional backup implementation in-place. You need to add a custom backup steps extending the `cma-server/resources/transitions/am2cm/sections/hdpX-backup.yml`

The screenshot shows the Cloudera Migration Assistant interface. On the left, a sidebar lists various backup steps under the 'Backup' section, including Backup Atlas, Backup Kafka, Setup HBase for Master Procedure Store drain, Backup HBase, Backup Hive, Backup HDFS, Backup Ranger, Backup Ranger KMS, Backup Infra-Solr, Backup Zookeeper, Backup etc configs, and Backup Kerberos. The main panel displays the 'Reference Documents' section with a link to 'Documentation'.

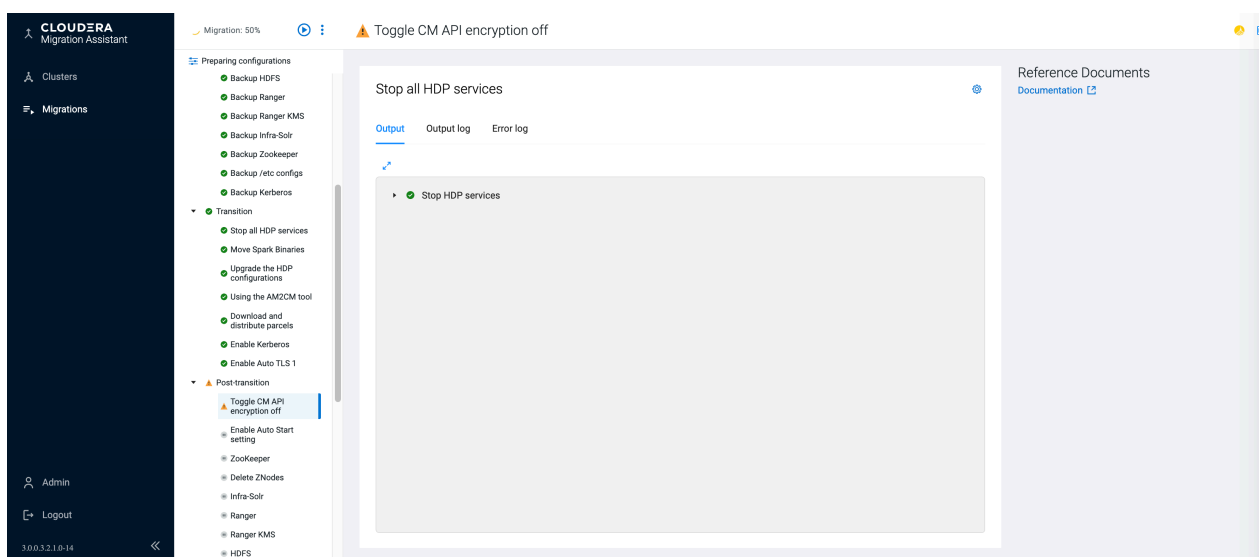
The Transition and Post-transition steps require a freshly installed empty Cloudera Manager with Cloudera Management Service that is compatible with the intended CDP runtime version. For information, see the [Cloudera Manager Installation and Setup](#) and [Installing Cloudera Management Service](#) documentation.

During the transition, there is a change in the binaries and update configurations. However, in post-transition steps, the payload data structure is changed for compatibility with the new service versions.

If you have a private parcel repository location available, you can change it by clicking `Download and distribute parcels` Settings Parcel .



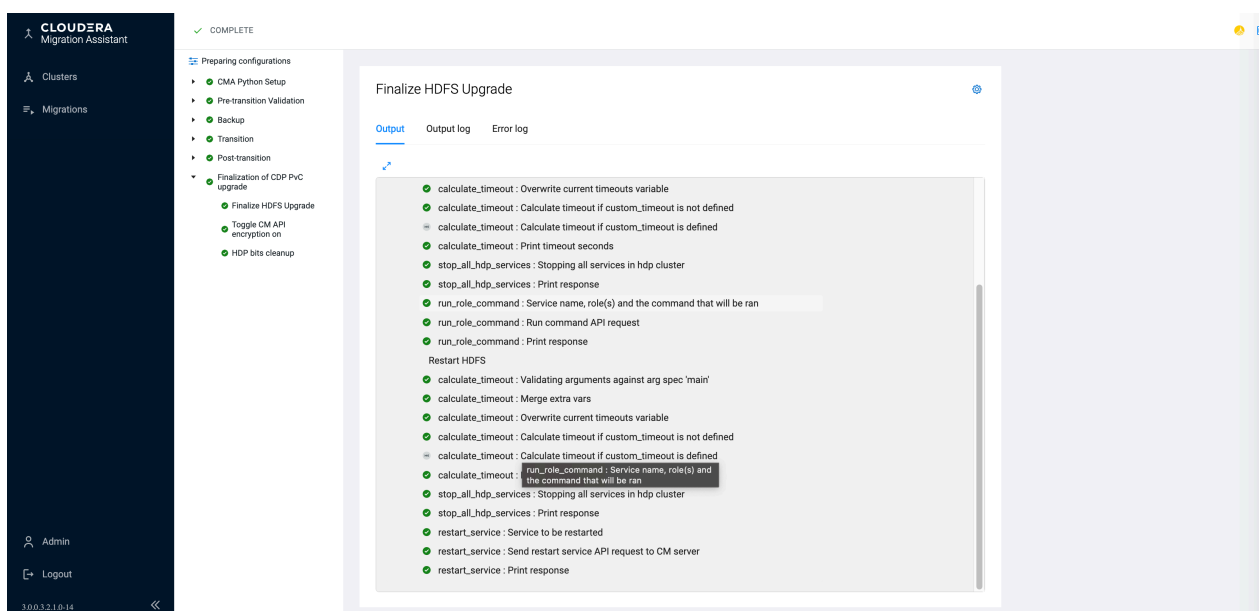
Note: From CDP 7.1.9 on, Queue Manager can only have Postgres as an internal DB (instead of H2). By default CMA installs a new Postgres DB instance, however you can specify an already existing DB here in Settings tab of the Queue Manager Step.





The CDP Private Cloud Base cluster is now ready for the Finalization steps.



Warning: If you want to rollback to the previous cluster, you must not proceed with the Finalization steps. After finalizing the cluster, you cannot rollback the cluster.



Individual transition steps

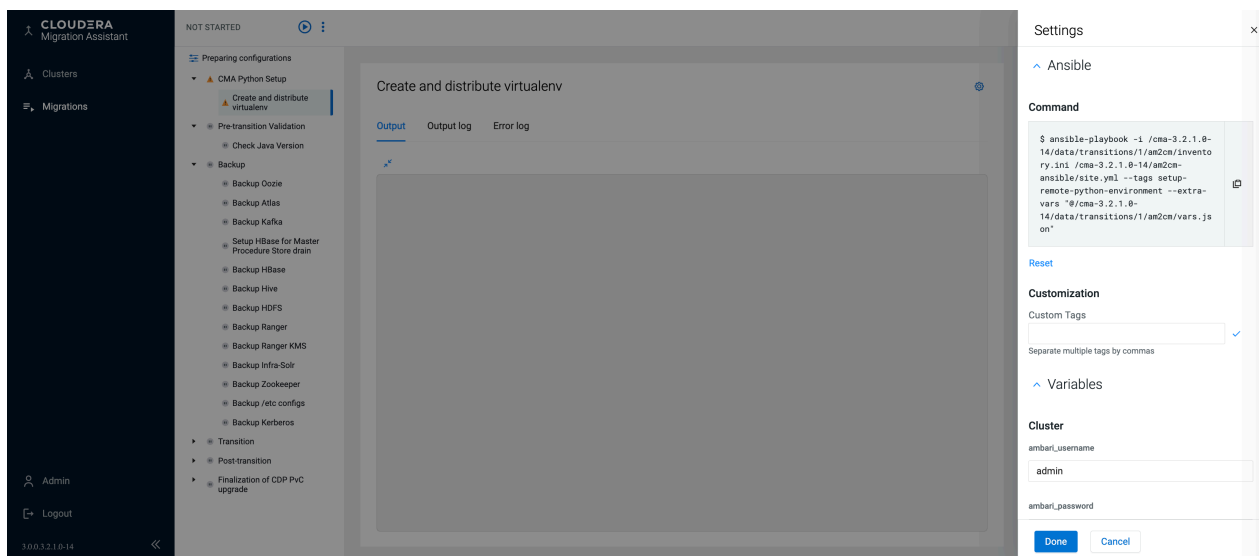
You can manually transition by clicking  and using the documentation. After the manual transition is complete, you must notify the CMA server by clicking  to indicate that the manual transition is complete.

However, the preferred approach is to use the Ansible script. The UI calls these scripts one by one based on their tags. You can change some of the input parameters for these scripts. Also, the scripts are part of CMA allowing you to modify the scripts. For information, see `<CMA_ROOT_DIR>/am2cm_ansi`. To modify the transition flow, you need to edit the .yaml file that is available. For information, see folder `<CMA_ROOT_DIR>/cma-server/config/transitions/am2cm`.

For every service there is a tag displayed. You can use the tag to identify the ansible task being executed. The actual ansible command is called in the control node where the CMA server (cma-server) is located and is shown below the

tag. After you click the icon, output and error log tabs appear below the ansible command, allowing you to check the output logs associated with the ansible upgrade playbook that is executed.

These scripts are manually executable on the am2cm-ansible directly on the control node (container) and also encouraged to be modified when needed.



Troubleshooting

This section provides you a list of possible causes and solutions to debug and resolve issues that you might face while upgrading HDP to CDP Private Cloud Base cluster.

Accessing CMA root folder in Docker

If you used Docker to upgrade, create a bash session inside the container and go to the am2cm folder by running the following command: `docker container exec -it <container_id> /bin/bash`

Examining CMA Server logs

Access the CMA root folder in Docker. The AM2CM (CMA) server logs are in the \$AM2CM_ROOT directory with the name am2cm-server.log.

Examining Transition Data

Access the CMA root folder in Docker. Use the transition id of your transition to enter \$AM2CM_ROOT/data/<transition id>

Adding or removing transition steps

Access the CMA root folder in Docker. Run the following command: `cd cms-server/config/transitions`

You will see transition-definition.yml

Manually editing Ansible input parameters

Access the CMA root folder in Docker. Use the data/<transition_id> directory within \$AM2CM_ROOT.

Editing transition parameters

<transition id>-inventory.ini: Inventory file with the hostnames and roles

<transition id>-vars.json & group_vars/: Ansible input that is extracted during the registration process

<transition id>-user-settings.ini: Input used by the am2cm tool. For more information, see [Transitioning HDP 3.1.5 cluster to CDP Private Cloud Base 7.1.x cluster using the AM2CM tool](#) and [Transitioning HDP 2.6.5 cluster to CDP Private Cloud Base 7.1.x cluster using the AM2CM tool](#).

conf/: Other configurations used by the am2cm tool

logs/: Logs from the ansible and tools used during the transition

Rewriting the Ansible scripts

You can change and add new Ansible scripts. All scripts are located in the folder am2cm-ansible. To change the ansible scripts, grep recursively for the tags that you want to change.

Backup HDP services from CDP 7.1.x

You must take a backup of HDP services before rolling back from CDP to HDP.

Automated Backup

The CMA Server takes the backup of everything except Ambari as it is not affected by the upgrade process. To back up Ambari, see the [Back Up Ambari for HDP 3.1.5](#) and [Back Up Ambari for HDP 2.6.5](#) documentation.

The backup handles HDP services in HDP. If you have MPacks or other application level services you must add those to the transition or perform separately.

Manual Backup

In case you skipped the Automated backup steps in CMA, then you can manually perform the following steps:

- Backing up the [HDP 2](#) or [HDP 3](#) cluster
- Backup /etc config symlinks for all HDP services on all hosts by running the following command: `cp -d /etc/<service>/conf /etc/<service>/conf.hdp.bak`
- Backup /etc/krb5.conf by running the following command: `cp /etc/krb5.conf /etc/krb5.conf.bak`

Rollback HDP services from CDP 7.1.x

To roll back, you must first backup using the [automated backup procedure](#) or [manually](#) starting the transition process. You can roll back an upgrade from HDP 3.1.5.x or 2.6.5 to CDP Private Cloud Base 7.1.x. The rollback restores your HDP cluster to the state it was in before the upgrade, including Kerberos and TLS/SSL configurations.

In 7.1.x, x represents greater than or equal to 6. For example, 7.1.6, 7.1.7, 7.1.8, and so on. Also, 7.1.7 SP1 and 7.1.7 SP2 rollback is supported.

Before you start rolling back the CDP Private Cloud Base 7 to HDP 3 or HDP 2, review the following information.

Caveats

- Any data created after the backup is lost.
- Follow all of the steps in the order presented in this topic. Cloudera recommends that you read through the backup and rollback steps before starting the backup process. You may want to create a detailed plan to help you anticipate potential problems.
- You can roll back to HDP after upgrading to CDP Private Cloud Base 7 only if the HDFS upgrade has not been finalized. The rollback restores your HDP cluster to the state it was in before the upgrade, including Kerberos and TLS/SSL configurations.

- These rollback steps depend on complete backups taken before upgrading to CDP. For steps where you need to restore the contents of a directory, clear the contents of the directory before copying the backed-up files to the directory. If you fail to do this, artifacts from the original upgrade can cause problems if you attempt the upgrade again after the rollback.

Review Limitations

The rollback procedure has the following limitations.

- HDFS – If you have finalized the HDFS upgrade, you cannot roll back your cluster.
- Configuration changes, including the addition of new services or roles after the upgrade are not retained after rolling back Ambari. 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.

Automated rollback

This section helps you to rollback automatically using the below procedure. Cloudera recommends you to use this procedure on the test clusters before performing the procedure on the production clusters.

Procedure

1. Go to the `am2cm-ansible` folder:
 - a) If you have used Docker to upgrade, create a bash session inside the container and go to the `am2cm-ansible` folder: `docker container exec -it <container_id> /bin/bashcd am2cm-ansible/`
 - b) If you have used local installation to upgrade, activate the python virtual environment and go to the `am2cm-ansible` folder: `source <path/to/am2cm-2.x.y.0-bb>/venv/bin/activate cd <path/to/am2cm-2.x.y.0-bb>/am2cm-ansible/`

2. Execute the rollback playbooks: Running the rollback playbooks require you to specify an inventory and extra-vars file. You can review the previous upgrade steps to identify the files specific to your environment.
 - a) Rollback Cloudera Manager to Ambari and restore /etc configuration directories `ansible-playbook -i <path/to/the/inventory.ini> playbooks/rollback/site.yml --extra-vars "@<path/to/the/vars.json>" --tags ambari-rollback,etc-config-rollback`
 - b) Rollback Kerberos `ansible-playbook -i <path/to/the/inventory.ini> playbooks/rollback/site.yml --extra-vars "@<path/to/the/vars.json>" --tags kerberos-rollback`
 - c) Rollback HDP Services:

Run the following command for all HDP services installed in the cluster in `ansible-playbook -i <path/to/the/inventory.ini> playbooks/rollback/site.yml --extra-vars "@<path/to/the/vars.json>" --tags <service-rollback-tag>`

The `<service-rollback-tag>` defaults to `<service>-rollback`, where `<service>` is replaced by the name of the HDP service to be rolled back.

For Ambari Infra Solr, the `<service-rollback-tag>` tag must be `infra-solr-rollback`.

You can skip some of the services that your cluster may not require. However, some services are dependent on each other. Cloudera recommends you to rollback in the order listed below.

- zookeeper-rollback
- infra-solr-rollback
- ranger-rollback
- ranger-kms-rollback
- hdfs-rollback
- yarn-rollback
- Kafka-rollback
- hbase-rollback
- atlas-rollback
- hive-rollback
- oozie-rollback

Manual rollback

This procedure helps you to rollback your cluster manually. While automated rollback is faster, manual rollback gives you full control on the process.

Manual rollback of HDP 2 or HDP 3 cluster

For manual rollback of CDP to HDP 2, see the [Procedure to Rollback from CDP 7.1.7 SP1 to CDP 7.1.7](#) documentation.

For manual rollback of CDP to HDP 3, see the [Procedure to Rollback from CDP 7 to HDP 3](#) documentation.

Restore old configuration symlinks

Restore old /etc configuration symlinks.

About this task

In the "Backup /etc configs" step, the original `etc/<service>/conf` symlinks were copied to the `/etc/<service>/conf.hdp.bak` symlink. You can restore them on each host using the following commands.

Procedure

1. `rm -f /etc/<service>/conf`

2. `cp -d /etc/<service>/conf.hdp.bak /etc/<service>/conf`

If the Backup /etc configs step was skipped during the upgrade process, you must manually reset the symbolic links to point to the correct configuration folder.

Kerberos

You must restore `krb5.conf`, regenerate keytabs, and check Ranger and Oozie HA mode.

Procedure

1. Restore `krb5.conf`. In the backup kerberos step, the original `/etc/krb5.conf` file was copied to the `/<path/to/am2cm-kerberos-backup>/krb5.conf.bak`. You can restore them on each host using the following command. * `<path/to/am2cm-kerberos-backup>` defaults to `/usr/am2cm/hdp-backup/kerberos` `cp <path/to/am2cm-kerberos-backup>/krb5.conf.bak /etc/krb5.conf` If the backup kerberos step was skipped during the upgrade process, you must manually reset the `krb5.conf` file.
2. Regenerate keytabs in Ambari UI.
3. If Ranger is in HA mode, you have to manually regenerate the `ranger.ha.keytab`
4. If Oozie is in HA mode, you have to manually regenerate the `oozie.ha.keytab`.

ZooKeeper

Use the following step to restore the backed-up data for ZooKeeper. For zookeeper, the default folder is `/hadoop/zookeeper`.

Procedure

1. Execute the following command to clean up and restore:

```
rm -rf /hadoop/zookeeper
tar xf up.tar.gz -C /
```

2. Start Zookeeper in Ambari UI.

Ambari Infra Solr

Use the following steps to restore backed-up ambari infrasolr data.

Procedure

1. Start Ambari-Infra service in Ambari UI.
2. Restore backed-up infra-solr data on the node where the infra-solr is installed.

For unsecured cluster

```
curl -v
"http://${INFRA_SOLR_URL}/solr/vertex_index/replication?command=restore&location=/path/to/backup/directory&name=vertex_index_backup"

curl -v
"http://${INFRA_SOLR_URL}/solr/edge_index/replication?command=restore&location=/path/to/backup/directory&name=edge_index_backup"

curl -v
"http://${INFRA_SOLR_URL}/solr/fulltext_index/replication?command=restore&location=/path/to/backup/directory&name=fulltext_index_backup"
curl -v
"http://${INFRA_SOLR_URL}/solr/ranger_audits/replication?command=restore&location=/path/to/backup/directory&name=ranger_audits_backup"
curl -v
```

```
"http://${INFRA_SOLR_URL}/solr/hadoop_logs/replication?command=restore&location=/path/to/backup/directory&name=hadoop_logs_backup"
curl -v
"http://${INFRA_SOLR_URL}/solr/audit_logs/replication?command=restore&location=/path/to/backup/directory&name=audit_logs_backup"

curl -v
"http://${INFRA_SOLR_URL}/solr/history/replication?command=restore&location=/path/to/backup/directory&name=history_backup"
```

For secured cluster

If the cluster is Kerberized, then you must kinit as the service principal.

```
curl -v --negotiate -u:
"http://${INFRA_SOLR_URL}/solr/vertex_index/replication?command=restore&location=/path/to/backup/directory&name=vertex_index_backup"

curl -v --negotiate -u:
"http://${INFRA_SOLR_URL}/solr/edge_index/replication?command=restore&location=/path/to/backup/directory&name=edge_index_backup"

curl -v --negotiate -u:
"http://${INFRA_SOLR_URL}/solr/fulltext_index/replication?command=restore&location=/path/to/backup/directory&name=fulltext_index_backup"
curl -v --negotiate -u:
"http://${INFRA_SOLR_URL}/solr/ranger_audits/replication?command=restore&location=/path/to/backup/directory&name=ranger_audits_backup"

curl -v --negotiate -u:
"http://${INFRA_SOLR_URL}/solr/hadoop_logs/replication?command=restore&location=/path/to/backup/directory&name=hadoop_logs_backup"

curl -v --negotiate -u:
"http://${INFRA_SOLR_URL}/solr/audit_logs/replication?command=restore&location=/path/to/backup/directory&name=audit_logs_backup"

curl -v --negotiate -u:
"http://${INFRA_SOLR_URL}/solr/history/replication?command=restore&location=/path/to/backup/directory&name=history_backup"
```

Ranger

Rollback procedure of Ranger includes restoring Ranger admin database and Ranger KMS database.

If the cluster is kerberized and Ranger is in HA mode, then you must manually regenerate the ranger.ha.keytab if you have not already done so.

Restore Ranger Admin Database

You must restore the Ranger Admin database for MySQL and PostgreSQL.

You must stop Ranger Admin and KMS service if they are running. Restore Ranger admin databases.

MySQL

Perform the following steps to restore a database.

1. Delete the existing Database.
2. Create an empty new database on the database host.
3. Restore the database using below msyql command.

```
mysql -u root
drop database ranger;
create database ranger;
```

```
GRANT ALL PRIVILEGES ON ranger.* TO 'rangeradmin'@'localhost';
$ mysql -u [username] -p existing_empty_db_name < dump_fileName.sql
```

Example

```
mysql -u rangeradmin -p rangeradmin < /root/backups/ranger/db/admin/ranger.s
ql
```

Press the Enter key. Type the database password when the password prompts.

POSTGRES

Perform the following steps to restore a database.

1. Delete an existing Database.
2. Create an empty new database in its place.
3. Run the below command on postgres db host.

```
dropdb -U owner_username dbname; [Enter db owner password at the prompt]
```

Example

```
dropdb -U rangeradmin ranger;
```

```
su - postgres
psql
create database ranger;
ALTER DATABASE ranger OWNER TO rangeradmin;
\q
exit
psql -U rangeradmin ranger < /root/backups/ranger/db/admin/ranger.sql
```

Press the Enter key. Type the database password when the password prompts.

Oracle

Set the path to Oracle home if required :

```
export ORACLE_HOME=/opt/oracle/product/12.2.0
export PATH=${PATH}:${ORACLE_HOME}/bin
export ORACLE_SID=orcl12c
```

Restore ranger admin database.

```
rm -rf del_admin_tbl_cmd.sql
sqlplus -s rangeradmin/rangeradmin << EOF
      spool on
      spool del_admin_tbl_cmd.sql
      SELECT 'DROP TABLE "' || TABLE_NAME || '" CASCADE CONSTRAINTS;'
FROM user_tables
      union ALL
      select 'drop ' || object_type || ' ' || object_name || ';' from us
er_objects
      where object_type in ('VIEW','PACKAGE','SEQUENCE', 'PROCEDURE',
'FUNCTION')
      union ALL
      SELECT 'drop '
      || object_type
      || ' '
      || object_name
      || ' force;'
FROM user_objects
```



```
WHERE object_type IN ('TYPE');
spool off
@del_admin_tbl_cmd.sql
exit;

EOF
```

Type the database password when the password prompts and then run the following command:

```
imp rangeradmin/rangeradmin file=backups/ranger/db/admin/orcl12c.sql log
=backups/ranger/db/admin/restore.log
```

Restore Ranger KMS Database

Restoring Ranger KMS involves steps for restoring MSQl, POSTGRES, and Oracle databases.

MySQL

To restore the database, perform the following:

- Delete the existing database.
- Create an empty new database on the Database host.
- Restore the database using below msyql command.

```
mysql -u root
drop database rangerkms;
create database rangerkms;
GRANT ALL PRIVILEGES ON rangerkms.* TO 'rangerkms'@'localhost';
$ mysql -u [username] -p existing_empty_db_name < dump_fileName.sql
```

Example

```
mysql -u rangerkms -p rangerkms < /root/backups/ranger/db/kms/rangerkms.sql
```

Press the Enter key. Type the database password when the password prompts.

POSTGRES

To restore data, perform the following:

- Delete the existing database.
- Create an empty new database in its place.
- Run the below command on postgres database host.

```
dropdb -U owner_username dbname; [Enter db owner password at the prompt]
```

Example

```
dropdb -U rangerkms rangerkms;
su - postgres
psql
create database rangerkms;
ALTER DATABASE rangerkms OWNER TO rangerkms;
\q
exit
psql -U rangerkms rangerkms < /root/backups/ranger/db/kms/rangerkms.sql
[Enter db owner password at the prompt as rangeradmin]
```

For Oracle

```
rm -rf del_kms_tbl_cmd.sql
sqlplus -s rangerkms/rangerkms << EOF
spool on
```

```

        spool del_kms_tbl_cmd.sql
        SELECT 'DROP TABLE "' || TABLE_NAME || '" CASCADE CONSTRAINTS;' FROM user_tables
        union ALL
        select 'drop ' || object_type || ' ' || object_name || ';' from user_objects
        where object_type in ('VIEW','PACKAGE','SEQUENCE','PROCEDURE','FUNCTION')
        union ALL
        SELECT 'drop '
        || object_type
        || ' '
        || object_name
        || ' force;'
        FROM user_objects
        WHERE object_type IN ('TYPE');
    spool off
    @del_kms_tbl_cmd.sql
    exit;
EOF

```

Press Enter and then run the following command.

```

imp rangerkms/rangerkms file=backups/ranger/db/kms/orcl12c.sql
log=backups/ranger/db/kms/restore.log

```



Note: If you have performed Ambari-Infra rollback already, then there are no additional rollback steps required to restore the Solr Collections.

HDFS

Before starting the rollback procedure, make sure that all the HDFS service roles are stopped.

About this task



Note: Before the HDFS rollback, Zookeeper, Ranger, Ambari-Metrics, and Ambari-Infra has to be rolled back and started in Ambari UI.

Procedure

1. Roll back all the JournalNodes. (Only required for clusters where high availability is enabled for HDFS). Use the JournalNode backup that you have created when you backed up HDFS before upgrading to the CDP Private Cloud Base.
 - a) Log in to each JournalNode host and do the following:
 1. remove the `$(dfs.journalnode.edits.dir)/current` directory
 2. restore the backup of `$(dfs.journalnode.edits.dir)/current` into `$(dfs.journalnode.edits.dir)/current` into `$(dfs.journalnode.edits.dir)/current`
2. Note down the target of the `/etc/hadoop/conf` symbolic link and remove it
3. Move the backup of `/etc/hadoop/conf` back to its original place, and perform these steps on all the cluster nodes where HDFS roles are installed, so on all NameNodes, JournalNodes and DataNodes.

4. Roll back all of the NameNodes.



Note: If high availability is not enabled on your cluster, then leave the Secondary NameNode as it is for now.

Use the backup of the Hadoop configuration directory you created during the backup phase.

Perform the following steps on all NameNode hosts:

- a) Start FailoverControllers and JournalNodes
 - b) If you use Kerberos authentication, authenticate with kinit with the NameNode's principal, otherwise change to the hdfs service user (usually `sudo -u hdfs`)
 - c) Run the following command: `hdfs namenode -rollback`
 - d) Restart HDFS FailoverControllers and JournalNodes in Ambari, then start the NameNodes note that one of the NameNodes should start, and one of them will remain in the starting state. When one of the NameNodes are marked as started proceed to DataNode rollback.
5. Roll back all of the DataNodes. Use the backup of the Hadoop configuration directory you created during the backup phase. Perform the following steps on all the DataNode hosts:
- a) If you use Kerberos authentication, authenticate with kinit with the NameNode's principal, otherwise change to the hdfs service user (usually `sudo -u hdfs`)
 - b) Run the following commands:

- `export HADOOP_SECURE_DN_USER=<hdfs service user>`
- `hdfs datanode -rollback`
- Look for output from the command similar to the following that indicates when the DataNode rollback is complete. wait until all storage directories are rolled back:

```
INFO common.Storage: Layout version rolled back to -57 for storage /
storage/dir_x
INFO common.Storage (DataStorage.java:doRollback(952)) - Rollback
of /storage/dir_x is complete
```



Note: If you do not see the output, check for the privileged-root-datanode-`{hostname}.err` file in the DataNode's log directory. If you see these log messages in the output, or in the privileged-root-datanode-`{hostname}.err` file for all of your DataNode data folders, then stop the process by typing `ctrl+c` as the DataNode rollback is ready.

6. If your cluster is not configured for NameNode High Availability, roll back the Secondary NameNode. Perform the following steps on the Secondary NameNode host:
 - a) Move the Secondary NameNode data directory to a backup location. (`$(dfs.namenode.name.dir)`)
 - b) If you use Kerberos authentication, authenticate with kinit with the NameNode's principal, otherwise change to the hdfs service user (usually `sudo -u hdfs`)
 - c) Run the following command: `hdfs secondarynamenode -format`

After rolling back the Secondary NameNode, 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:

```
INFO namenode.SecondaryNameNode: Web server init done
```

7. Restore the original symlink with the noted target as `/etc/hadoop/conf` on all the nodes where it has changed.
8. Restart the HDFS service. Open Ambari, and go to the HDFS service page, in the Service actions dropdown select Start.
9. Monitor the service, and if everything comes up fine, check the HDFS file system availability, you can run an `hdfs fsck /` or generate the file system listing with `hdfs dfs -ls -R /` and compare it with the one that you did as part of the backup procedure to see if everything got rolled back properly. In case of any issues, please contact Cloudera Support before you proceed.

YARN

Before starting the rollback procedure, make sure that HDFS and Zookeeper are rolled back.

1. Log in to the TIMELINE SERVICE V2.0 READER host.
2. Setup Kerberos Credentials in case of secured cluster. Locate the yarn-ats-hbase's keytab and use kinit to cache the kerberos ticket.
 - `kinit -kt path/to/yarn-ats.hbase-master.keytab yarn-ats-hbase/hostname@domain.`
 - `export JVMFLAGS="-Djava.security.auth.login.config=/etc/hadoop/conf/embedded-yarn-ats-hbase/yarn_hbase_master_jaas.conf"`
3. Delete the atsv2-hbase-secure znode in the Zookeeper `zookeeper-client -server ${zk_server_url} rmr /atsv2-hbase-secure.`

HBase

If you have performed Zookeeper and HDFS rollback already, there are no additional rollback steps required for HBase.

1. Start HBase in Ambari UI.
2. If the HBase master does not start, ZooKeeper data must be cleaned by following these steps:
 - a. Log in to the HBase Master host.
 - b. Setup Kerberos Credentials in case of secured cluster.
 1. `kinit -kt path/to/yarn-ats.hbase-master.keytab yarn-ats-hbase/hostname@domain`
 2. `export JVMFLAGS= "-Djava.security.auth.login.config=/usr/hdp/current/hbase-master/conf/hbase_master_jaas.conf"`
 - c. Delete the hbase-secure znode in the Zookeeper. `zookeeper-client -server ${zk_server_url} rmr /hbase-secure`

Kafka

To roll back Kafka, perform the following steps.

Procedure

1. Kafka service depends on Zookeeper. Make sure Zookeeper data is restored.
2. After rollback, start the Kafka service, and check, if the producers and consumers can connect to the cluster.
3. Remove the inter broker protocol and log format version settings from the Kafka settings:
 - a) Log in to Ambari.
 - b) Choose the Kafka service.
 - c) Select the Configuration page.
 - d) Find the Custom kafka-broker section.
 - e) Remove following properties:
 - `inter.broker.protocol.version=current_Kafka_version`
 - `log.message.format.version=current_Kafka_version`
 - f) Restart the Kafka service.
 - g) Start Kafka in Ambari UI

Atlas

Perform the following steps to restore HBase tables and ATLAS_ENTITY_AUDIT_EVENTS table.

Procedure

1. Stop Atlas from Ambari

2. Setup Kerberos Credentials in case of secured cluster . Locate the atlas user's keytab and use kinit to cache the kerberos ticket for atlas user .

Example

```
kinit -kt path/to/atlas.service.keytab atlas/hostname@domain
```

3. Restore the HBase tables, atlas_janus and ATLAS_ENTITY_AUDIT_EVENTS.
4. Restore the Solr Collections.

Hive

Before starting the rollback procedure, make sure that HDFS and Zookeeper have already rolled back.

You must delete your existing Hive Metastore database to roll back the Hive services.

To restore data on the node where the Hive Metastore database is located, perform the following steps.

1. Delete an existing database. Create an empty database in its place. `$ mysql -u <hive_user> drop database <hive_db>; create database <hive_db>;`
2. Restore Hive Metastore database `$ mysql -u <hive_user> <hive_db> < </path/to/dump_file>`. If you have performed HDFS rollback already, there are no additional rollback steps required for Hive.
3. Start Hive service in Ambari UI.

Spark

Know more about the Spark roll back.

- The Spark application history lives in HDFS, and with the rollback of HDFS, the history at the time of the backup is restored. Any Spark applications run after the backup and rollback will not be visible.
- After the rollback, versions of Spark applications built against HDP 2 should be used instead of versions that are rebuilt against CDP.
- Start Spark in Ambari UI.

Oozie

To roll back the Oozie service, you must restore the Oozie database. If the cluster is kerberized and Oozie is in HA mode then you must manually regenerate the oozie.ha.keytab if you have not already done so.

- Start Oozie in Ambari UI.

It is needed to re-generate the oozie.ha.keytab file using the spnego.service.keytab files from the Oozie server hosts and from the Oozie load balancer host and then distribute the generated oozie.ha.keytab onto the respective Oozie hosts. In order to achieve this, follow the steps listed in this [website](#).

- Restart Oozie.

Knox

With the backup and rollback of Ambari, Knox is also backed up and rolled back by default.

Start Knox in Ambari UI.

Zeppelin

With the backup and rollback of HDFS, Zeppelin is also backed up and rolled back by default.

Start Zeppelin in Ambari UI.



Important: Deprecation notice for Zeppelin: Zeppelin is deprecated in Cloudera Runtime 7.1.9 and 7.2.18. For more information, see the deprecation notices in the corresponding Cloudera Runtime release notes.

Related Information

[Deprecation notice for Zeppelin](#)