

Migrating from CDH to CDP Public Cloud

Date published: 2023-10-11

Date modified: 2024-09-30

CLOUdera

Legal Notice

© Cloudera Inc. 2025. All rights reserved.

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

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

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

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

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

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

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

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

Contents

Cloudera Migration Assistant Overview.....	4
Supported Cloudera on cloud migration paths.....	4
Release Notes.....	4
Cloudera Migration Assistant server deployment.....	7
Deploying Cloudera Migration Assistant locally or with Docker.....	8
Deploying Cloudera Migration Assistant with parcel.....	13
Enabling TLS/SSL for Cloudera Migration Assistant.....	15
Storing secrets in Vault.....	17
Migrating to Cloudera on cloud with Cloudera Migration Assistant.....	18
Reviewing prerequisites before migration.....	18
Registering source clusters.....	19
Scanning the source cluster.....	21
Creating collections for migration.....	26
Registering destination clusters.....	27
Migrating from source cluster to destination cluster.....	30
Migrating Spark applications.....	35
Migrating Oozie workflows.....	38
Migrating SQL queries.....	40
Migrating HBase tables.....	42

Cloudera Migration Assistant Overview

Cloudera Migration Assistant is a user interface based extensible tool to assist Hadoop (CDH) users to easily migrate data, metadata and certain workloads to the various form factors of Cloudera.

Supported Cloudera on cloud migration paths

Cloudera Migration Assistant (previously AM2CM) can be used to migrate from the legacy CDH and Cloudera Base on premises platforms to Cloudera on cloud. The supported migration paths vary based on the Cloudera Migration Assistant version.

Cloudera Migration Assistant version	Migration paths	Cloud provider	Workloads	Data
Cloudera Migration Assistant 3.5	Cloudera Private Cloud Base 7.1.7, 7.1.8, 7.1.9 # Cloudera Public Cloud 7.2.18 migration	AWS, Azure	SQL Oozie Spark	HDFS files HMS tables HBase tables
Cloudera Migration Assistant 3.4	Cloudera Private Cloud Base 7.1.7, 7.1.8, 7.1.9 # Cloudera Public Cloud 7.2.18 migration	AWS, Azure	SQL Oozie Spark	HDFS files HMS tables HBase tables
Cloudera Migration Assistant 3.3	CDH 5.1.x # Cloudera Public Cloud 7.2.x CDH 6.3.x # Cloudera Public Cloud 7.2.x Cloudera Private Cloud Base 7.1.7# Cloudera Public Cloud 7.2.x	AWS, Azure	SQL Oozie	HDFS files HMS tables HBase tables
Cloudera Migration Assistant 3.2	CDH 5.1.x # Cloudera Public Cloud 7.2.x CDH 6.3.x # Cloudera Public Cloud 7.2.x	AWS	SQL Oozie (Hive action)	HDFS files HMS tables HBase tables
Cloudera Migration Assistant 3.0	CDH 5.1.x # Cloudera Public Cloud 7.2.x CDH 6.3.x # Cloudera Public Cloud 7.2.x	AWS	SQL	HDFS files HMS tables
Cloudera Migration Assistant 2.8	CDH 6.3.x # Cloudera Public Cloud 7.2.17 (Technical Preview)	AWS	SQL	HDFS files HMS tables

Release Notes

Learn about the known issues, and fixed issues and behavioral changes in Cloudera Migration Assistant.

3.5.0

What's new

- AMCM-2422 Enhanced **Migration Dashboard** with Component Versions
- AMCM-793 Managing Secrets with **Vault**
- AMCM-2455 **Mask** supplied **passwords** in configuration and log files
- AMCM-2308 **Spark3** to Spark3 Support

- AMCM-2400 Execution Instructions for **Livy**
- AMCM-2594 Parcel compatibility check to CSD
- Various CVE fixes

Behavioral changes

- Support for upgrading from HDP to Cloudera platform is discontinued from this release.

3.4.1

What's new

- AMCM-2489 Configurable JVM heap size for CDH discovery tool

Fixed issues

- AMCM-2692 Stop All Services In The Cluster - fails with non-default password
- AMCM-2637 Spark application parameters with single - won't work
- AMCM-2570 Remove Label assignment removes HDFS locations
- AMCM-2650 SQL Workload Migration doesn't replicate its SQL file from HDFS
- AMCM-2625 Multiple HiveSQL migration ignores the HiveSQL files
- AMCM-2626 NPE when trying to get source server clusters after db upgrade
- AMCM-2624 /root/.cdp/credentials Missing for hive migration
- AMCM-2636 Use master node in execution Instruction

Known issues

- AMCM-2457 Python 3.12.x dependency collision on Cloudera Migration Assistant server node

3.4.0

What's new

- AMCM-1168 Spark Application Migration
 - Exploring Past Spark Workloads via Spark History Server
 - Spark Workload's dependency discovery based on spark-submit and application parameters
 - Limitations
 - CDH6 and Cloudera Base on premises as Source with Spark 2
 - No spark code change available
 - No automated jar and Python package dependency discovery
- AMCM-2426 Store SSH Settings as Credentials
- AMCM-2300 Upgradability - Tech Preview
- Several UX improvements and fixes

Known issues

- AMCM-2457 Python 3.12.x dependency collision on Cloudera Migration Assistant server node

3.3.1

What's new

- Parcel deployment - available as Technical Preview

The parcel deployment enables you to install Cloudera Migration Assistant as a service in Cloudera Manager.

Fixed issues

- AMCM-2273 Cloudera Migration Assistant doesn't handle long clusternames
- AMCM-2233 Create a field to manage the number of Ansible forks
- AMCM-2232 Fix user settings validation on target registration when changing clusters

- AMCM-2218 Replace Browser icon (Favico) to Cloudera Migration Assistant icon small version
- AMCM-2067 Obtain Hive HMS DB Password from User Directly
- Pre-checks on Cloudera Migration Assistant Server Start:
 - AMCM-2188 Checking if Java11 and JAVA_HOME exists
 - AMCM-2188 Checking Java before starting the server
 - AMCM-2186 Checking Python version

3.3.0

What's new

- AMCM-1764 Azure Support
- AMCM-1418 Cloudera Base on premises to Cloudera on cloud Migration (without security metadata migration)
- AMCM-1361 Oozie Migration with Map-Reduce action and generic support for all other action types
- AMCM-1765 Cloudera Migration Assistant Landing Page
- AMCM-1878 Side-bar Redesign
- AMCM-2135 One-time HBase Migration without Cloudera Replication Manager

Limitations

You need to manually migrate Ranger policies from Cloudera Base on premises to Cloudera on cloud as described in [Migrating from source cluster to destination cluster](#).

3.2.1

Fixed Issues

- AMCM-1879 Missing airgapped for localhost (#1696)
- AMCM-1908 Hive SQL Export button does not work (#1693)
- AMCM-1906 Fix Broken background color (#1691)
- AMCM-1888 Show Refresh option of scanned data tables as in UI design - rebase fix (#1686)
- AMCM-1880 ps is missing from the rhel8 docker image (#1659) (#1685)
- AMCM-1902 Make navigation panel not scrollable (#1679) (#1684)
- AMCM-1872 Filtered scan results are reset after switch to another page (#1655) (#1678)
- AMCM-1830 discovery_bundle_builder fails if no mysql-connector-java.jar is found (#1682)
- AMCM-1892 L&S Data migration Cloudera Replication Manager policy should have default value for frequencyInSec (#1681) (#1683)
- AMCM-1888 Show Refresh option of scanned data tables as in UI design (#1666) (#1670)
- AMCM-1887 Fix Mapping discrepancies (#1662) (#1671)
- AMCM-1891 Change API info (#1668) (#1675)
- AMCM-1867 Set step status before perform (#1667) (#1674)
- AMCM-1820 Toast message appears unexpectedly (#1663) (#1669)
- AMCM-1871 Fix wrong position of collection component (#1643) (#1672)

Known Issues

AMCM-1860 Airgapped download

The following files need to be downloaded and copied manually before any --airgapped install.

- <CMA_ROOT_DIR>
 - atlas-migration-exporter-0.8.0.2.6.6.0-332.tar.gz
 - jdbc-drivers/
 - mysql-connector-java-5.1.46.jar
 - ojdbc8.jar
 - postgresql-42.3.4.jar

Only add the JDBC driver your cluster uses.

Behavioral Changes

- /info endpoint API response changed to correctly return the product name (AMCM-1891)

Fixed Common Vulnerabilities and Exposures

Learn more about the Common vulnerabilities and Exposures (CVEs) that were fixed in this release.

- [CVE-2024-1597](#): org.postgresql:postgresql vulnerable to SQL Injection via line comment generation - AMCM-1910
- /info endpoint API response changed to correctly return the product name (AMCM-1891)

3.2.0

Known Issues

HBase migration from CDH needs extra parcels Installed

In order to use Cloudera Migration Assistant for you need obtain the following parcels from Cloudera Support :

- CDH5 : CLOUDERA_OPDB_REPLICATION-1.0-1.CLOUDERA_OPDB_REPLICATION5.14.4.p0.31473501-el7.parcel
- CDH6 : CLOUDERA_OPDB_REPLICATION-1.0-1.CLOUDERA_OPDB_REPLICATION6.3.3.p0.8959316-el7.parcel

These along with their hash files need to be copied to parcels directory in Cloudera Migration Assistant root directory

Cloudera Migration Assistant server deployment

You can deploy the Cloudera Migration Assistant server (cma-server) to any of your source cluster nodes, or in the case of local or docker mode, to an external node that has visibility to the cluster. It requires 1.5 GB of extra memory, and unless you are in parcel deployment mode, you can install Cloudera Migration Assistant with or without internet connection. You can choose between deploying the Cloudera Migration Assistant server locally, using Docker or with parcel.

Dependencies

The following components must be installed on the cma-server host:

- [Python 3.8.12](#) or higher version
- Docker deployment requires [docker 20+](#) or higher versions
- Local and parcel deployment requires [JDK11](#) (with JAVA_HOME set)

In case you do not have JDK11 installed on you cluster, you can download it using the following commands:

```
wget https://download.java.net/java/GA/jdk11/9/GPL/openjdk-11.0.2_linux-x64_bin.tar.gz -C /usr/java/
ln -s /usr/java/jdk-11 /usr/java/jdk-11.0.2/
```

You can skip specifying Java Home if it is located on any of the following default paths:

- /usr/lib/jvm/java-11
- /usr/java/jdk-11
- /usr/lib/jvm/jdk-11
- /usr/lib64/jvm/jdk-11
- /usr/lib/jvm/zulu-11
- /usr/lib/jvm/zulu11
- /usr/lib/jvm/java-11-zulu-openjdk
- /usr/lib/jvm/java-11-oracle

When setting up Cloudera Migration Assistant 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 Cloudera Migration Assistant under the following directory:

```
cma-[***VERSION NUMBER**]/am2cm-ansible/python_requirements/
```

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



Note: Even though the installation of Cloudera Migration Assistant can be completed without internet connection, you need to ensure that you have internet connection when downloading the JDBC drivers and Atlas artifacts. The following files must be manually downloaded and copied before installing Cloudera Migration Assistant in an air-gapped environment:

- [***CMA ROOT DIRECTORY**]
 - atlas-migration-exporter-0.8.0.2.6.6.0-332.tar.gz
 - jdbc-drivers/
 - mysql-connector-java-5.1.46.jar
 - ojdbc8.jar
 - postgresql-42.3.4.jar

Only add the JDBC driver your cluster uses.

Deploying Cloudera Migration Assistant locally or with Docker

Learn more about how to deploy Cloudera Migration Assistant locally and in docker with or without internet connection.

Procedure

1. Download the binaries of the Cloudera Migration Assistant tool from <https://archive.cloudera.com/cma/3.5/tars/> with one of the following commands:

For With internet connection

- ```
wget https://archive.cloudera.com/cma/3.5/tars/cma-3.5.0.0-8-bin.tar.gz
```
- ```
curl https://archive.cloudera.com/cma/3.5/tars/cma-3.5.0.0-8-bin.tar.gz
```

For Without internet connection

- ```
mkdir cma-3.5
cd cma-3.5
wget https://archive.cloudera.com/cma/3.5/tars/cma-3.5.0.0-8-bin.tar.gz
wget https://archive.cloudera.com/cma/3.5/tars/cma-extras-gpl-3.5.0.0-8-bin.tar.gz-bin.tar.gz
```
- ```
mkdir cma-3.5
cd cma-3.5
curl https://archive.cloudera.com/cma/3.5/tars/cma-3.5.0.0-8-bin.tar.gz
curl https://archive.cloudera.com/cma/3.5/tars/cma-extras-gpl-3.5.0.0-8-bin.tar.gz-bin.tar.gz
```

When the required binaries are downloaded successfully, the directory structure should look like the following example:

```
drwxr-xr-x 14 testuser testuser 4096 febr 27 13:21 cma-3.5.0.0-8/
-rw-rw-r-- 1 testuser
testuser 518140466 febr 27 13:28 cma-3.5.0.0-8-bin.tar.gz
-rw-rw-r-- 1 testuser
testuser 85089637 febr 27 13:28 cma-extras-gpl-3.5.0.0-8.tar.gz
```

The supported version of Cloudera Migration Assistant is 2.8.0 and higher.

2. Extract the downloaded file using the following command:

```
tar xzf cma-3.5.0.0-8-bin.tar.gz
```

3. Start the Cloudera Migration Assistant server locally or in a Docker container. The preferred method is the Docker mode.

Ensure that Python 3.8.12 or a higher version is installed on the host. In case you do not define the python executable when running the script, you will be prompted to enter the python executable path.

- Docker

For With internet connection

Run the `cma-docker.sh` script in the untarred top-level folder to launch the Cloudera Migration Assistant server in a Docker container. `cma-[***VERSION NUMBER**]/bin/cma-docker.sh --start`



Note: The script creates the docker image if necessary. Additionally, the script provides the following operations to manage the Cloudera Migration Assistant Docker container: start, stop, restart, or rebuild. If you want to explore other available options, run the following command: `cma-[***VERSION NUMBER**]/bin/cma-docker.sh --help`.

For Without internet connection

```
cd cma-[***VERSION NUMBER**]/
bin/cma-docker.sh --start --airgapped --python-executable=python3
```



Note: If the GPL file is not located in the same directory as the Cloudera Migration Assistant file, you can use the following command, where you define the path of the GPL file:

```
cma-[***VERSION NUMBER**]/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
```

- Locally

For With internet connection

Run the `cma-local.sh` script in the untarred top-level folder and follow its instructions to launch the Cloudera Migration Assistant server locally. `cma-[***VERSION NUMBER**]/bin/cma-local.sh --start`



Note: The script creates a Python virtual environment in the top-level folder where the dependencies are installed. Additionally, the script provides the following operations to manage the Cloudera Migration Assistant locally: start, stop, restart, or rebuild. To explore other available options, run the following command: `cma-[***VERSION NUMBER**]/bin/cma-local.sh --help`.

For Without internet connection

```
cd cma-[***VERSION NUMBER**]/
```

```
bin/cma-local.sh --start --airgapped --python-executable=python3
```



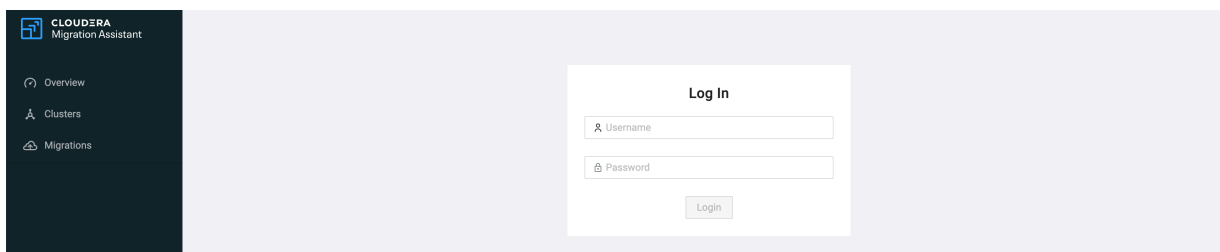
Note: If the GPL file is not located in the same directory as the Cloudera Migration Assistant file, you can use the following command, where you define the path of the GPL file:

```
cma-[***VERSION NUMBER**]/bin/cma-local.sh --start --  
airgapped  
--cma-extras-gpl-tar-location=[***ABSOLUTE PATH  
TO EXTRAS GPL***]
```

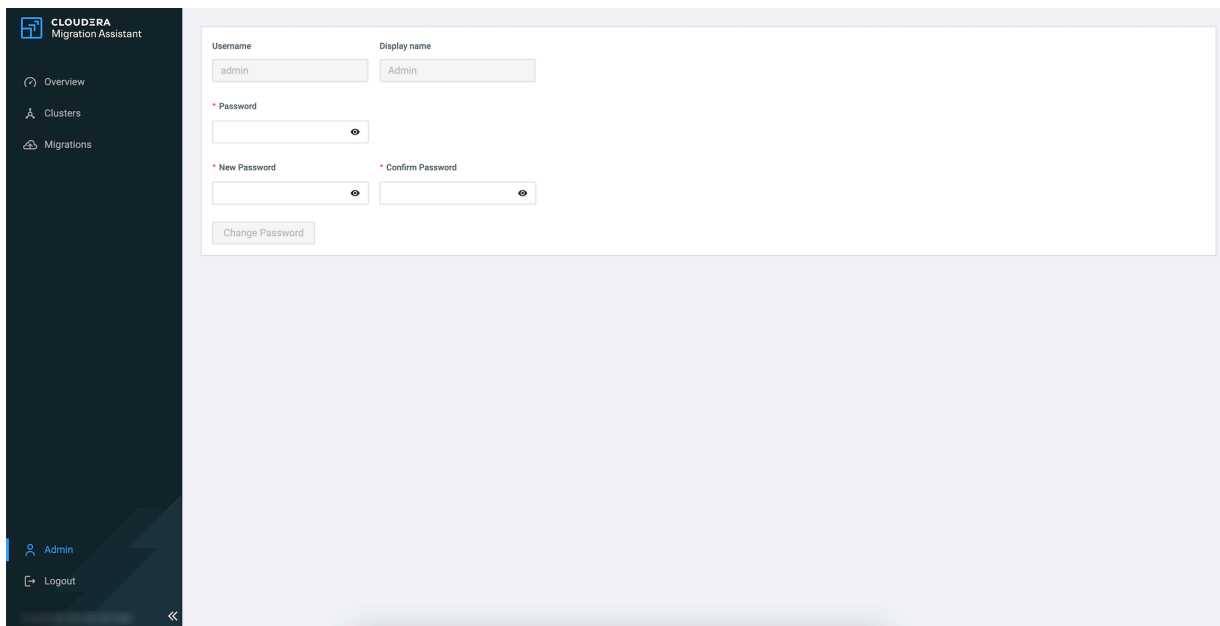
Check that the local pypi repository is installed correctly.

```
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
```

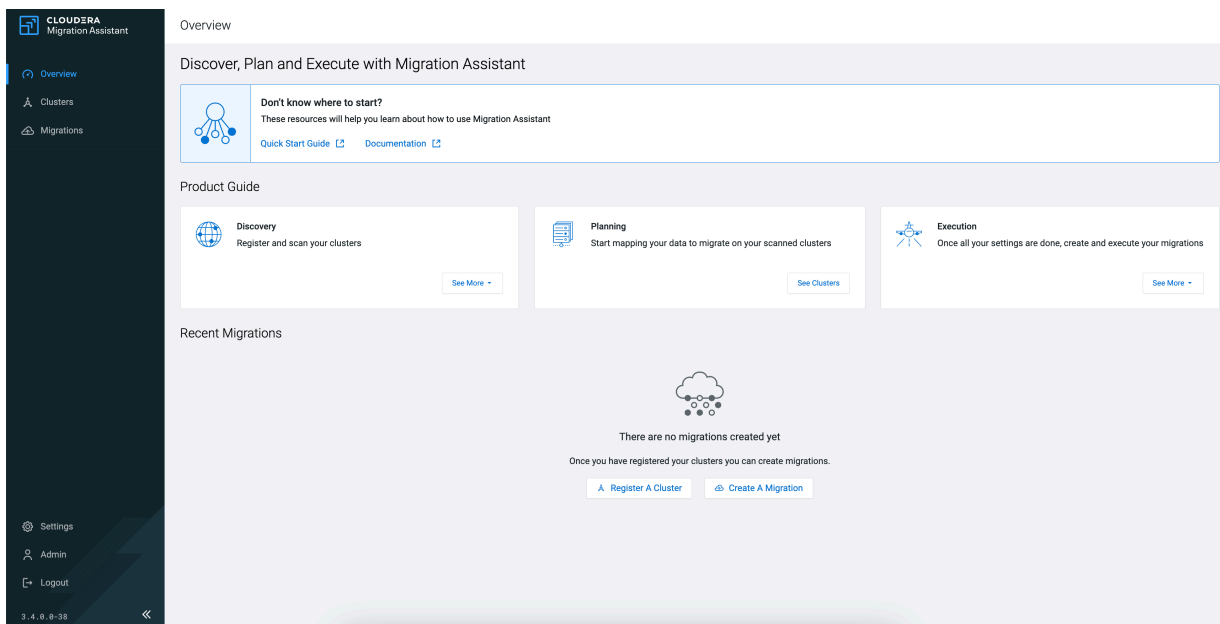
4. Access <http://localhost:8090> in a browser, such as Chrome or Firefox, to open the Cloudera Migration Assistant tool.



The default user password pair is admin-admin, but you can change your password at the user profile page.



After changing the password, you can start using the Cloudera Migration Assistant.



Deploying Cloudera Migration Assistant with parcel

Learn more about how to deploy Cloudera Migration Assistant with a parcel in Cloudera Manager.

About this task

Cloudera Migration Assistant can be an add-on service in Cloudera Manager. To deploy Cloudera Migration Assistant with a parcel, you need to upload the Cloudera Migration Assistant Custom Service Descriptor (CSD) files to the default CSD directory, and add the Cloudera Migration Assistant parcel to your cluster using Cloudera Manager.

**Note:**

The parcel deployment of Cloudera Migration Assistant is in Technical Preview and not ready for production deployment. Cloudera encourages you to explore these features in non-production environments and provide feedback on your experiences through the *Cloudera Community Forums*.

Procedure

1. Copy the Cloudera Migration Assistant CSD files to /opt/cloudera/csd/ directory on the Cloudera Manager node:

```
wget -P /opt/cloudera/csd/ https://archive.cloudera.com/cma/3.5/csd/CMA-3.5.0.0-8.jar
```

Cloudera Manager automatically detects the CSD files.

2. Change the ownership of the CSD files.

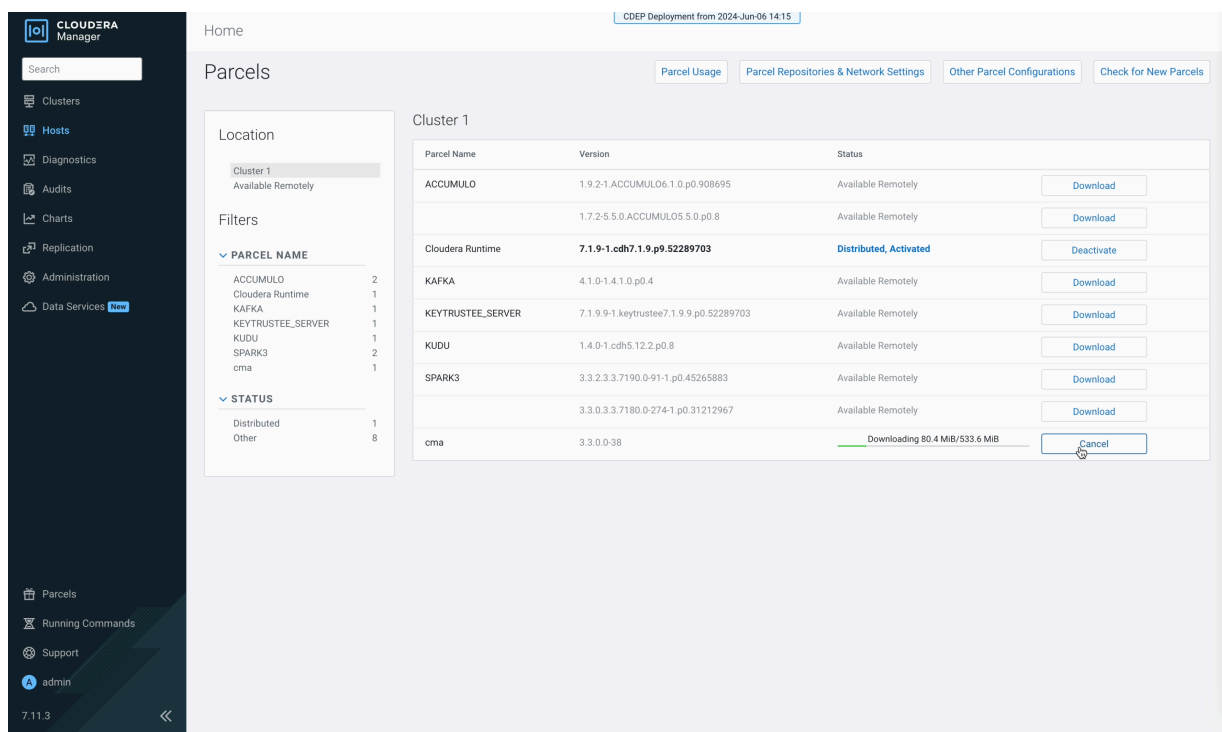
```
chown cloudera-scm:cloudera-scm /opt/cloudera/csd/CMA-3.5.0.0-8.jar
```

3. Restart Cloudera Manager and CMS services for the changes to take effect.

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

4. Log into Cloudera Manager.
5. Select **Hosts** **Parcels** in the left navigation bar.

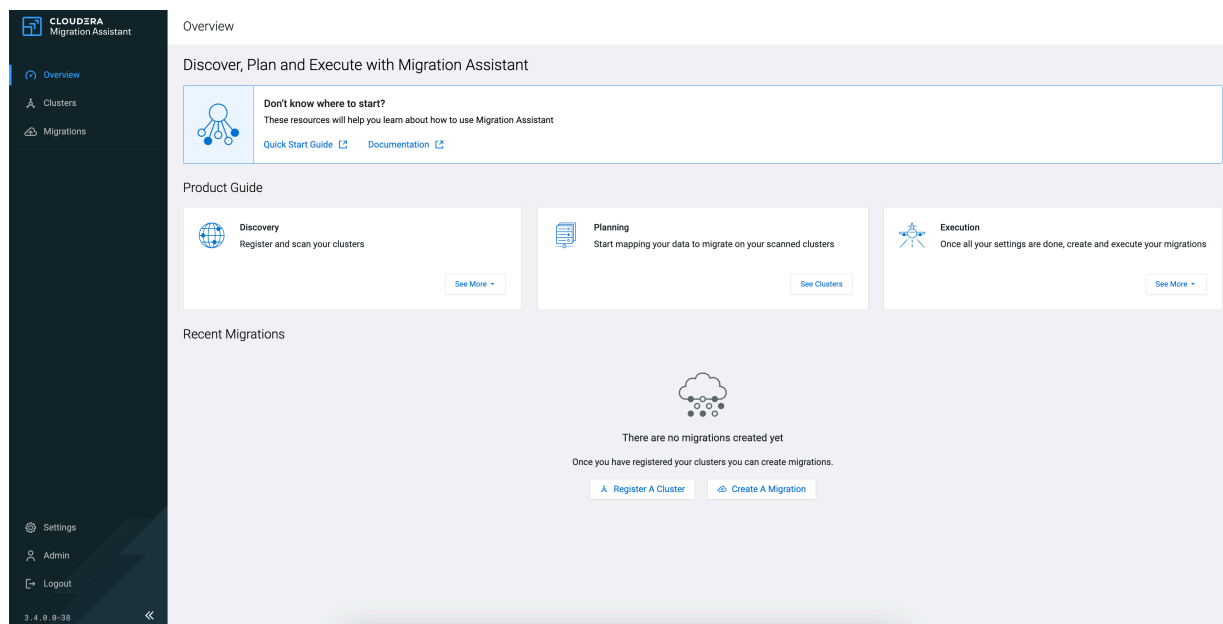
6. Search for Cloudera Migration Assistant, and click Download to download the parcel to the local repository.



7. After the download is completed, click Distribute to distribute the parcel to all clusters.
8. After the parcel is distributed, click Activate to activate the parcel.
9. Click OK when confirmation is required.
10. Click Clusters on the left navigation pane.
11. Select the drop-down menu to the right of your cluster.
12. Select Add Service.
13. From the list, select CMA as the type of service, then click Continue.
The **Add CMA Service** wizard opens.
14. Assign the Cloudera Migration Assistant server role to the hosts where you Python 3.8.12 and JDK11 installed, and click Continue.
15. Click Continue after reviewing the configurations of the Cloudera Migration Assistant service.
The first run of the Cloudera Migration Assistant service starts. When the command finishes, the Cloudera Migration Assistant service is added to the cluster.

16. Access the Cloudera Migration Assistant User Interface (UI).

After going back to the homepage of your cluster, open the Cloudera Migration Assistant service page, and click on the CMA Server UI tab that opens the Cloudera Migration Assistant UI.



Related Information

[Add-on Services](#)

[Registering source clusters](#)

Enabling TLS/SSL for Cloudera Migration Assistant

Learn more about how to enable and configure TLS/SSL for Cloudera Migration Assistant when deploying the Cloudera Migration Assistant server locally or with Docker.

Before you begin

- Ensure that you have a PKCS12 keystore created with the cma key alias. JKS keystore is also supported.

Procedure

1. Click Settings on the homepage of Cloudera Migration Assistant.

The screenshot shows the 'TLS Settings' page in the Cloudera Migration Assistant. The 'Enable TLS/SSL' checkbox is currently unchecked. Below it are several configuration fields: Certificate, Private Key, Key Password, Keystore File Location (with a default value of /cma-99.99.99-99-986/cma.ks), Keystore File Password, Keystore Type (set to pkcs12), Key Alias (set to cma), Trust Store (set to classpath:cma_truststore.p12), Trust Store Password, and Trust Store Type (set to pkcs12). At the bottom right, there are 'Save' and 'Restart' buttons.

2. Enable TLS/SSL using the checkbox on the **TLS Settings** page.

This screenshot shows the same 'TLS Settings' page, but now the 'Enable TLS/SSL' checkbox is checked. The 'Certificate' field is populated with 'classpath:cma_cert.pem' and the 'Private Key' field is populated with 'classpath:cma_key.pem'. The other fields remain the same as in the previous screenshot. The 'Save' and 'Restart' buttons are still present at the bottom right.

3. Provide the TLS/SSL Certificate in PEM format.
4. Provide the Private Key in PEM format.
5. Provide the Private Key Password if the Private Key is encrypted.
6. Provide the Keystore File Location.
7. Provide the Keystore File Password.
8. Select the Keystore Type from the drop-down menu.
9. Provide a Key Alias.
10. Provide the Trust Store.
11. Provide the Trust Store Password.
12. Provide the Trust Store Type.

13. Click Save.
14. Click Restart after saving the TLS/SSL configurations.

Results

After restarting, the Cloudera Migration Assistant server listens on the default port 8090 (HTTP) and 8093 (HTTPS), and all HTTP requests are redirected to the HTTPS port.

Storing secrets in Vault

You can store your secrets in a local Vault instance managed by Cloudera Migration Assistant. If the Vault is configured, no credentials are stored in the database, and the files created during the migration process contain only Vault paths. Additionally, credentials stored in the Vault are also masked on the UI in this case.



Note: Storing secrets in Vault is in technical preview.

Local or Docker Deployment

Starting CMA with Vault for the first time

Starting Cloudera Migration Assistant with the `--vault` option will start and configure a local Vault server, and configure the Cloudera Migration Assistant Server to connect to it. Once the connection is set up, the Cloudera Migration Assistant will store the credentials on the Vault Server.

1. When Cloudera Migration Assistant Server is not running, start Cloudera Migration Assistant with the following command:

```
cma-[***VERSION NUMBER**]/bin/cma start --vault
```

2. When Cloudera Migration Assistant Server is already running, use the following command to restart Cloudera Migration Assistant with Vault:

```
cma-[***VERSION NUMBER**]/bin/cma restart --vault
```

To restart the Vault server, you can use the following command:

```
cma-[***VERSION NUMBER**]/bin/cma vault restart
```

After the initial configuration is complete, Cloudera Migration Assistant Server expects the Vault Server to be running. If you want to stop the Vault, it is recommended that you stop CMA as well using the following commands:

```
cma-[***VERSION NUMBER**]/bin/cma vault stop  
cma-[***VERSION NUMBER**]/bin/cma stop
```



Note:

In the docker deployment, the Vault runs in the Cloudera Migration Assistant container so it cannot be started, stopped or restarted without the Cloudera Migration Assistant.

Parcel Deployment

When you deploy Cloudera Migration Assistant in parcel mode, the Vault Server role is installed on the Cloudera Migration Assistant Server node, and configured by default.

**Note:**

After the initial configuration is complete, Cloudera Migration Assistant Server and Vault Server roles can be started, stopped or restarted separately in Cloudera Manager.

Migrating to Cloudera on cloud with Cloudera Migration Assistant

The following steps will guide you through how to migrate your data, metadata and workload from a CDH or Cloudera Base on premises cluster to a Cloudera cluster on cloud.

Reviewing prerequisites before migration

Before migrating from CDH 5, CDH 6 or Cloudera Base on premises to Cloudera on cloud, review the list of prerequisites that are required for the migration process.

- Ensure that the Cloudera Migration Assistant server is deployed as described in *Setting up Cloudera Migration Assistant server*.
- The CDH 5 source cluster minimum version requirement is CDH 5.16.1 and CDH 5.16.2 in case of HBase migration.
- CDH 6 source cluster minimum version requirement is CDH 6.3.3.
- The Cloudera Base on premises source cluster minimum version requirement is 7.1.7.
- For HBase migration, you need either of the following parcels procured from Cloudera Professional Services:
 - CLOUDERA_OPDB_REPLICATION-1.0-1.CLOUDERA_OPDB_REPLICATION5.14.4.p0.31473501-el7.parcel
 - CLOUDERA_OPDB_REPLICATION-1.0-1.CLOUDERA_OPDB_REPLICATION6.3.3.p0.8959316-el7.parcel
- For data and metadata migration, you need a Data Lake cluster already created in a Cloudera environment on cloud. To create a Data Lake cluster, you can follow the process described in *Registering an AWS environment* and *Registering an Azure environment* based on your cloud provider.
- For a Hive workload migration, you need a Cloudera Data Engineering Data Hub already created in a Cloudera environment on cloud. To create a Cloudera Data Engineering Data Hub cluster, you can follow the process described in *Creating a cluster on AWS* and *Creating a cluster on Azure* based on your cloud provider.
- You must use the Cluster Connectivity Manager to manually register the source CDH cluster as a classic cluster in the Cloudera Control Plane, following the process described in *Adding a CDH cluster (CCMv2)*.
- Information to gather before you begin the migration:
 - For the source CDH cluster: The Cloudera Manager URL, Admin username and password, SSH user, port, and private key of source nodes
 - For the destination Cloudera cluster/environment: Cloudera Control Plane URL, Admin username and password, SSH user, port, and private key
 - In S3: S3 bucket access key and S3 bucket secret key, S3 credential name. Potentially, you might also need the S3 bucket base path for HDFS files, S3 bucket path for Hive external tables (these paths should auto-fill from the selected destination cluster, but can be changed if needed)
- The Cloudera Manager node of the source CDH cluster must have Python 3.8.12 or higher installed.
- Redaction needs to be off in Cloudera Manager. To disable redaction in Cloudera Manager, you can follow the process described in *Disabling Redaction of sensitive information*.

Related Information

[Setting up CMA server](#)

[Registering an AWS environment](#)

[Creating a cluster on AWS](#)

[Cluster Connectivity Manager](#)

[Adding a CDH cluster \(CCMv2\)](#)

[Disabling Redaction of sensitive information when using the Cloudera Manager API](#)

Registering source clusters

To migrate from CDH to Cloudera on cloud, you need to register the CDH or Cloudera Base on premises cluster as a source from which the data, metadata and workload will be migrated.

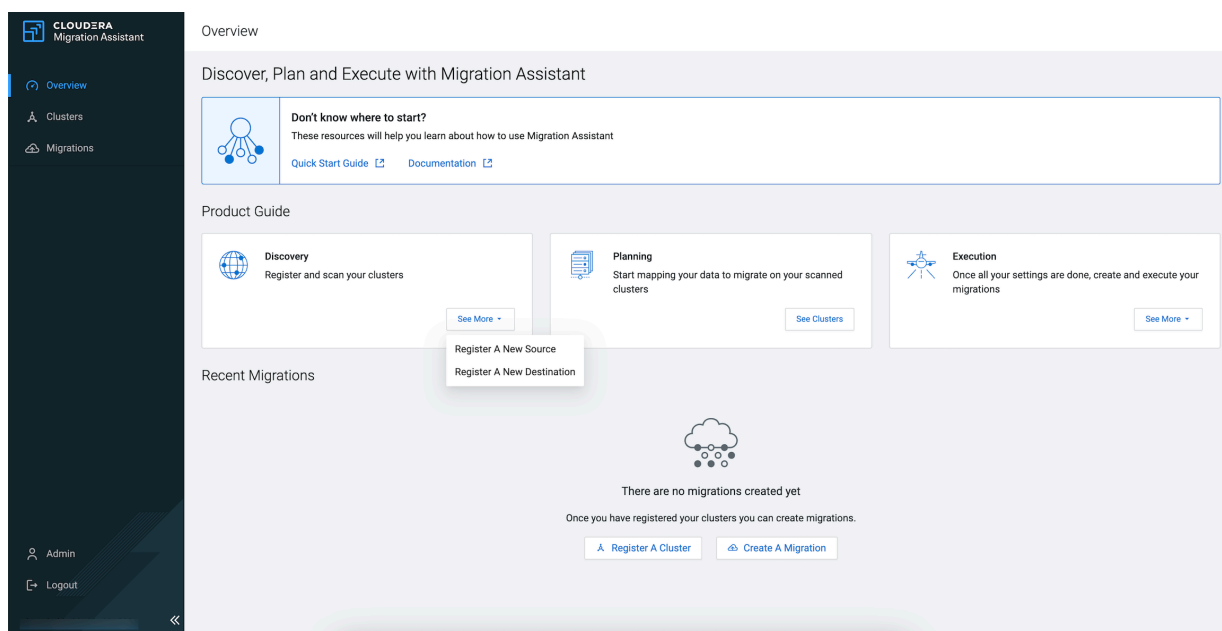
Before you begin

Make a note of the following information about the CDH cluster to complete the source registration:

- Cloudera Manager URL
- Admin username and password for Cloudera Manager
- SSH user, port, and private key

Procedure

1. Click [See More Register A New Source](#) on the homepage of Cloudera Migration Assistant to register a cluster that will be used as a source for the migration.



Alternatively, you can open the New Cluster wizard by selecting **Clusters** on the left navigation pane, and clicking **New Source**.

2. Select Cloudera Distributed Hadoop 5, Cloudera Distributed Hadoop 6 or Cloudera Base on premises as **Source Type**.
3. Provide the URL of Cloudera Manager that belongs to the CDH 5, CDH 6 or Cloudera Base on premises cluster. The URL needs to be in the following format:

```
https://[***CLOUDERA MANAGER HOSTNAME***]:[***CLOUDERA MANAGER PORT***]/
```

4. Provide the Admin user and Admin password of Cloudera Manager.
5. Click Next.
6. Choose the cluster based on the Cluster Name that you want to use for the migration. The drop-down list contains all of the clusters that are managed by Cloudera Manager on the provided host.

7. Click Next.
8. Select the Configuration Preference based on which authentication method you prefer.
 - Choose Use existing if you want to use the SSH configuration and keys of the user running Cloudera Migration Assistant server to access the hosts.
 - Choose New if you want to use a newly provided SSH key to configure Ansible automatically.
 - a. Provide the SSH User and SSH Port.
 - b. Copy the SSH Key to the SSH Key box or upload a .pem file containing the key.
 - c. If you want to use Password, provide the password and make sure the sshpass utility is installed.



Note: The user specified during SSH configuration must have **passwordless sudo rights**.

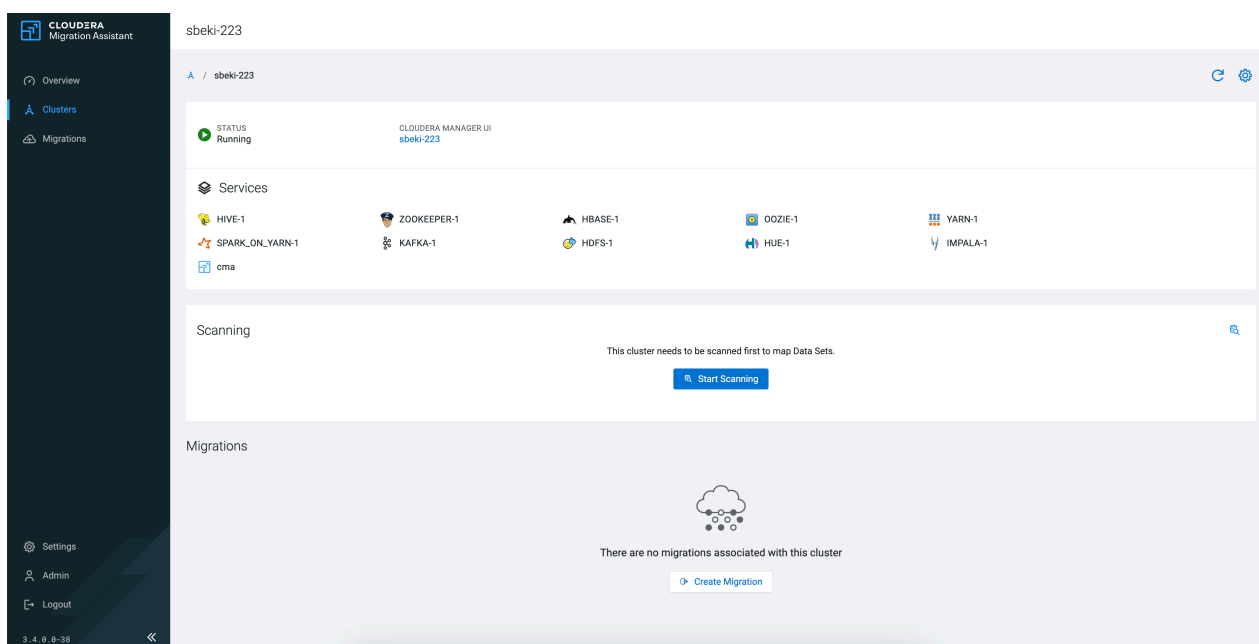
9. Click Create.

Results

The registered CDH or Cloudera Base on premises cluster is listed on the **Clusters** page.

Status	Name	Platform	Provider	Type	Actions
Running	sbeki-ls222	CDH		Source	
Running	sbeki-223	CDH		Source	

You can review the details and services of the cluster by clicking on the Name of the cluster.



What to do next

Scan the data and workloads on the registered source cluster and add labels to specify the objects, which should be included in the migration.

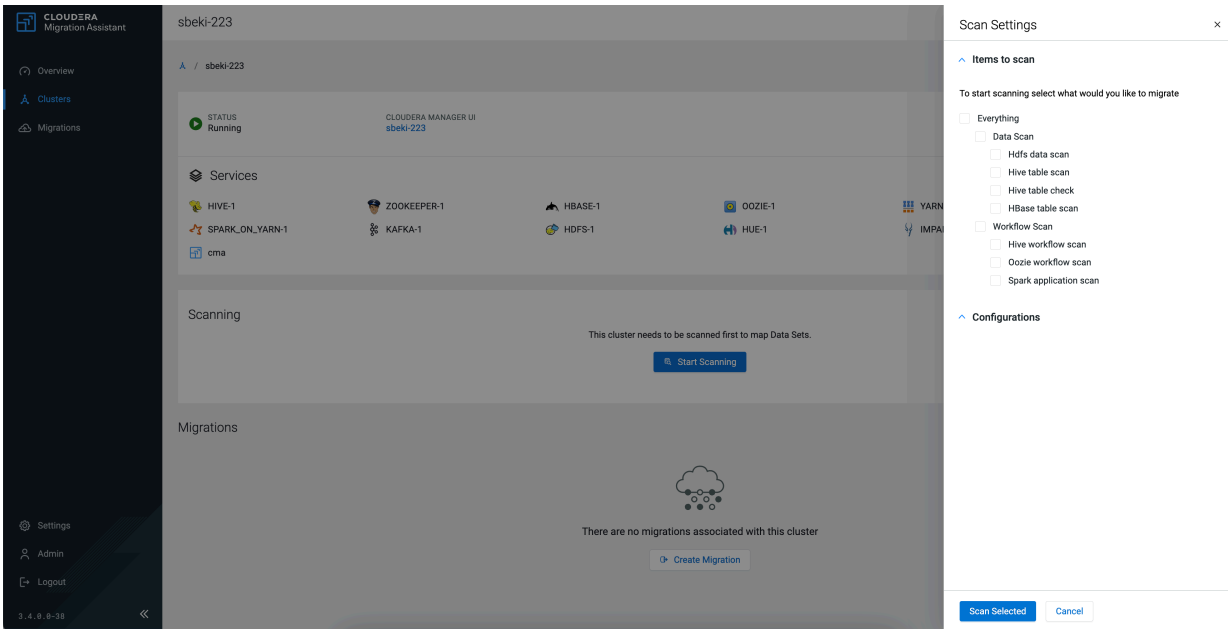
Scanning the source cluster

You need to scan the CDH or Cloudera Base on premises source cluster to identify the available datasets and workloads that can be migrated. Scanning also enables you to review and resolve syntax errors that can occur after the migration.

Procedure

1. Click on the CDH or Cloudera Base on premises cluster you want to use for the migration on the **Clusters** page.

2. Click Start Scanning to open the **Scan Settings** where you can select the data and workloads for scanning.



3. Select Everything or choose from the different scanning options.

The following items are available for scanning:

HDFS data scan

The HDFS data scan uses `_hdfs_report_` module from the *CDH Discovery Tool* to scan HDFS on the source cluster.

Hive table scan

The Hive table scan uses `_hive_metastore_` module from the *CDH Discovery Tool* to scan Hive on the source cluster.

Hive table check

Scanning Hive tables on the source cluster. `_Hive Table Check_` embeds sre and u3 sub-programs of the *Hive SRE Tooling*. The result will be visible at the SRE column of the Hive datasets.

HBase table scan

Scanning HBase tables on the source cluster.

Hive workflow scan

Scanning Hive SQL queries on the source cluster. You can pre-scan Hive2 SQL queries against Hive3 with the Hive Workflow scan option. When selecting this Hive Workflow option, you need to provide the location of your queries as shown in the following example:

- HDFS paths
 - With default namespace: `hdfs:///dir/`, `hdfs:///dir/file`
 - With specified namespace: `hdfs://namespace1/dir`, `hdfs://namespace1/dir/file`
 - With namenode address: `hdfs://nameNodeHost:port/dir`, `hdfs://nameNodeHost:port/dir/file`
- Native file paths
 - `your/local/dir`
 - `nodeFQDN:/your/local/dir/sqlFile`

Oozie workflow scan

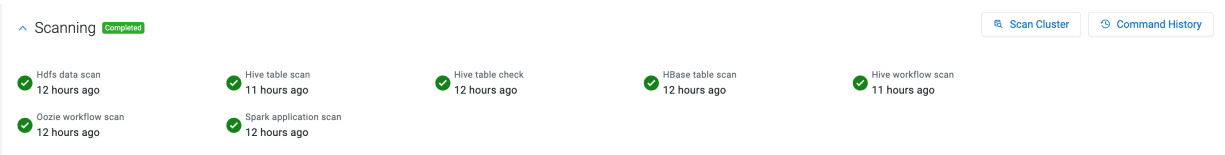
Scanning Oozie workloads on the source cluster. If you selected Oozie workflow scan, you need to provide the Number of latest days to scan.


Spark application scan


- Scanning Spark applications on the source cluster. If you selected Spark application scan, you need to provide the Number of latest days to scan.
- Spark History Server needs to be configured to be accessible by the spark user, to do that you can add spark to the list in `spark.history.ui.admin.acls` at the `SPARK_ON_YARN` service in Cloudera Manager.
- Spark jobs also need to be visible by the the user spark.

4. Click Scan selected.








You will be redirected to the scanning progress where you can monitor if the selected items are successfully scanned or encountered an error.



a)  Click **Scan Cluster** to open the **Scan Settings** again to add more items to the scan or trigger a rescan of the already scanned items.

b)  Click **Command History** to open the **Source command history** to have more insight about the scanning progress, stop an in progress scan and review the log.

Source command history

A / sbeki-ls222 / Scan History					
Execution Id	Command Type Name	Start Time	End Time	Log	Actions
10	 Hive workflow scan	30/09/2024, 09:33:00	30/09/2024, 09:33:33	+	
9	 Hive table scan	30/09/2024, 09:20:01	30/09/2024, 09:20:37	+	
8	 Hive workflow scan	30/09/2024, 09:20:00	30/09/2024, 09:20:34	+	
7	 Spark application scan	30/09/2024, 08:02:18	30/09/2024, 08:02:53	+	
6	 Oozie workflow scan	30/09/2024, 08:02:17	30/09/2024, 08:02:55	+	
5	 Hive workflow scan	30/09/2024, 08:02:17	30/09/2024, 08:04:10	+	
4	 HBase table scan	30/09/2024, 08:02:16	30/09/2024, 08:03:15	+	





Note: The scan results are also available at the data directory of the source cluster, which is located in the `<CMA_ROOT_DIR>/data/sources/ClusterName/Source_ID/cluster-scan/` folder on the Cloudera Migration Assistant server node.

5. Click Start Mapping to review the data, workflows and applications on the source cluster and map their configuration to the destination cluster.

For example, when reviewing **Hive SQL**, you can check and edit any SQL query related errors before migrating the workflows to Cloudera on cloud. The migration will be successful regardless of fixing the statement errors.

However, you will not be able to execute the SQL queries on the new cluster due to the compatibility issues

between Hive2 and Hive3. You can review the list of errors using  , and open the editor using  .

✓ COMPLETE

⌕ ↻

← Edit Hive SQL

💾 Save

	Statement	Error
1	SET hive.default.fileformat.managed= None	Invalid value. expects one of [none, textfile, sequencefile, rcfile, orc, parquet]
2	SET hive.execution.engine=mr	hive.execution.engine must be set to tez
3	SET hive.limit.query.max.table.partition= -1	SET hive.limit.query.max.table.partition= -1 does not exist
4	SET hive.metastore.hbase.aggr.stats.cache.entries = 10	SET hive.metastore.hbase.aggr.stats.cache.entries = 10 does not exist
5	SET hive.metastore.hbase.aggr.stats.hbase.ttl= 600	SET hive.metastore.hbase.aggr.stats.hbase.ttl= 600 does not exist
6	SET hive.metastore.hbase.aggr.stats.invalidator.frequency = 600	SET hive.metastore.hbase.aggr.stats.invalidator.frequency = 600 does not exist
7	SET hive.metastore.hbase.aggr.stats.memory.ttl = 600	SET hive.metastore.hbase.aggr.stats.memory.ttl = 600 does not exist
8	SET hive.metastore.hbase.aggregate.stats.cache.size = 3	SET hive.metastore.hbase.aggregate.stats.cache.size = 3 does not exist

```
1 SET mapred.map.tasks = 20;
2 SET hive.exec.mode.local.auto=true;
3 SET hive.merge.mapfiles=true;
4 SET hive.map.aggr=true;
5 SET hive.optimize.index.filter=false;
6 SET hive.limit.query.max.table.partition= -1;
7 SET hive.warehouse.subdir.inherit.perms = false
8 SET hive.stats.fetch.partition.stats=true;
9 SET hive.metastore.hbase.cache.ttl=600s;
10 SET hive.metastore.hbase.catalog.cache.size = 3;
11 SET hive.metastore.hbase.aggregate.stats.cache.size = 3;
12 SET hive.metastore.hbase.aggregate.stats.max.partitions = 3;
13 SET hive.metastore.hbase.aggregate.stats.false.positive.probability = 3;
14 SET hive.metastore.hbase.aggregate.stats.max.variance = 3;
15 SET hive.metastore.hbase.cache.max.writer.wait = 600;
16 SET hive.metastore.hbase.cache.max.reader.wait = 600;
17 SET hive.metastore.hbase.cache.max.full = 600;
18 SET hive.metastore.hbase.cache.clean.until = 40;
19 SET hive.metastore.hbase.connection.class = some.connection.class;
20 SET hive.metastore.hbase.aggr.stats.cache.entries = 10;
21 SET hive.metastore.hbase.aggr.stats.memory.ttl = 600;
22 SET hive.metastore.hbase.aggr.stats.invalidator.frequency = 600;
23 SET hive.metastore.hbase.aggr.stats.hbase.ttl= 600;
24 set hive.driver.parallel.compilation = false;
25 set datanucleus.connectionPool.maxPoolSize=30;
26 set datanucleus.connectionPoolingType = BONECP;
27 SET hive.auto.convert.join.noconditionaltask.size= 20971520;
28 SET hive.auto.convert.sortmerge.join=false;
29 SET hive.auto.convert.sortmerge.join.to.mapjoin = false ;
30 SET hive.cbo.enable=false;
31 SET hive.cbo.show.warnings=false;
32 SET hive.compactor.worker.threads=0;
33 SET hive.compute.query.using.stats= false;
34 SET hive.default.fileformat.managed= None;
35 SET hive.exec.dynamic.partition.mode=strict;
36 SET hive.exec.max.dynamic.partitions = 1000;
37 SET hive.exec.max.dynamic.partitions.pernode = 100;
38 SET hive.exec.reducers.max=1099;
39 SET hive.execution.engine=mr;
40 SET hive.fetch.task.conversion=minimal;
41 SET hive.fetch.task.conversion.threshold= 250MB;
42 SET hive.hashtable.key.count.adjustment=1;
43 SET hive.limit.optimize.enable=FALSE;
```

After fixing the statement errors in the SQL editor window, Save the changes. The edited queries are replicated and saved in the S3 bucket of the destination cluster. The original files are not overwritten.


After the scanning is completed, you can add the tables and workflows from the selected services to collections. **Collections** serve as an organizational method to sort out the data and workflows resulted from the scan for migration.

Hive TablesHDFS LocationHive SQLHBase TablesOozie Job Definitions

Q Search data...

Collections

✕ Clear

<input type="checkbox"/>	DB name	Table name	Type	SRE	Labels	Policy name	Policy State	
<div>No Data</div>								

Collections

Default

0 items

Results

The datasets and workflow on the CDH or Cloudera Base on premises source cluster is scanned for Hive, HDFS, HBase, Oozie and Spark.

What to do next

Sort the scanned data and workflows into collections to have more control over what is migrated from the source cluster to the target cluster.

Related Information

CDH Discovery Tool

Hive SRE Tooling

Creating collections for migration

After scanning the source cluster, you can use collections to sort the datasets that need to be migrated to the destination cluster. Collections can also be useful to organize different types of data, workflows and applications before migration.

Procedure

1. Click Start Mapping or Collections.
2. Click Hive Tables, HDFS Location, Hive SQL, HBase Tables, Oozie Job Definitions or Spark Applications based on which items you would like to add to a collections.

The different windows show the results of the scanning. For example, the Hive Tables display all the existing tables in Hive on the source cluster.

The screenshot shows the 'Mapping' window for Hive Tables. The table has columns: DB name, Table name, Type, Issues, Collections, and Policies. The data rows are:

DB name	Table name	Type	Issues	Collections	Policies
default	customers	MANAGED_TABLE	2 3	Default	
default	sample_07	MANAGED_TABLE	2 3	Default	
default	sample_08	MANAGED_TABLE	2 3	Default	
default	test	EXTERNAL_TABLE	1 3	1	

The 'Collections' sidebar on the right shows 'Default' with 0 items.

3. Select the items that you want to add to a collection, and click Add to collection.

The selected items are added to the **Default** collection, and the **Default** label is assigned to the selected items.

The screenshot shows the 'Mapping' window for Hive Tables. The table has columns: DB name, Table name, Type, Issues, Collections, and Policies. The data rows are:

DB name	Table name	Type	Issues	Collections	Policies
default	customers	MANAGED_TABLE	2 3	Default	
default	sample_07	MANAGED_TABLE	2 3	Default	
default	sample_08	MANAGED_TABLE	2 3	Default	
default	test	EXTERNAL_TABLE	1 3	1	
default	web_logs	MANAGED_TABLE	2 3	1	

The 'Collections' sidebar on the right shows 'Default' with 3 items. A notification banner at the top says 'The label 'Default' successfully add to 3 Hive table'.

You have the option to create more collections beside the **Default** collection.

4. Click **+** next to **Collections**, and customize the collection by providing a Label name and selecting a Color. Click Create.

The screenshot shows the 'Mapping' tab in the Cloudera Migration Assistant. On the left, a sidebar lists cluster services: HIVE, SQL, HDFS, HBASE, OOZIE, and SPARK. The main area displays a table of mapping results with columns: DB name, Table name, Type, Issues, Collections, and Policies. The 'Collections' dropdown is open, showing a search bar and a 'Clear' button. The 'Collection Details' dialog is open on the right, allowing customization of the collection name (set to 'Test collection') and color (set to 'Red'). The 'Create' button is highlighted.

Select more items from the scanning results, and add it to the newly created collection.

The screenshot shows the 'Mapping' tab in the Cloudera Migration Assistant. On the left, a sidebar lists cluster services: HIVE, SQL, HDFS, HBASE, OOZIE, and SPARK. The main area displays a table of mapping results with columns: DB name, Table name, Type, Issues, Collections, and Policies. The 'Collections' dropdown is open, showing a search bar and a 'Clear' button. The 'Collection Details' dialog is open on the right, allowing customization of the collection name (set to 'Test collection') and color (set to 'Red'). The 'Create' button is highlighted.

You can manage the created collections by clicking on to open the collections menu. You can review the labeled results in a collections using View Items, and modify or delete the created collection using Edit and Remove. Removing a collection does not affect the items on the source cluster, only the labelling is deleted.

Results

The datasets are labeled for migration.

What to do next

Register the destination cluster to which the datasets and workflows are going to be migrated.

Registering destination clusters

As you are migrating from CDH or Cloudera Base on premises to Cloudera on cloud, you need to register the Cloudera cluster on cloud as a destination to which the data, metadata, and workload will be migrated.

Before you begin

Note down the following information about the Cloudera cluster on cloud to complete the registration:

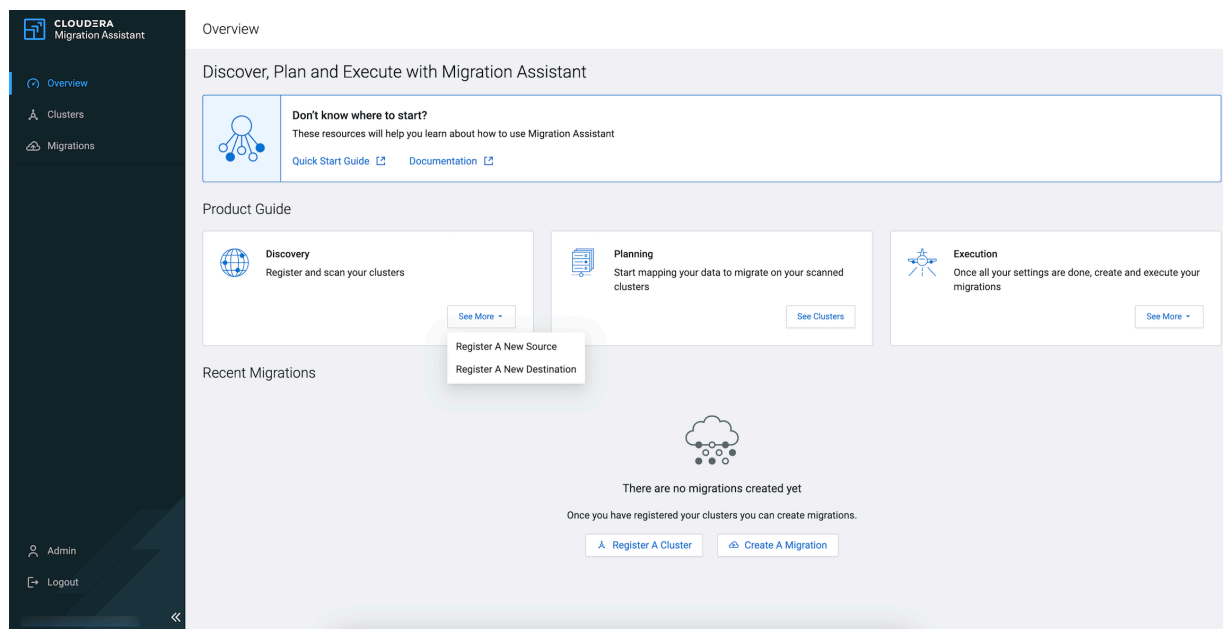
- Access key and private key

For more information about how to generate access and private key, see the [Generating an API access key](#) documentation.

- SSH user, port, and private key

Procedure

1. Click **See More Register A New Destination** on the homepage of Cloudera Migration Assistant to register a cluster that will be used as a source for the migration.



Alternatively, you can open the New Cluster wizard by selecting **Clusters** on the left navigation pane, and clicking **New Destination**.

2. Select Cloudera on cloud as **Target Type**.
3. Select the Control Plane URL where the cluster is located.
4. Provide the Access Key and Private Key of your Cloudera user account.
5. Click Next.
6. Choose the cluster based on the Cluster Name that you want to use for the migration.
The drop-down list contains all of the existing clusters in Cloudera on cloud that you have access to.
7. Click Next.
8. Select the Configuration Preference based on which authentication method you prefer.
 - Choose Use existing if you want to use the SSH configuration and keys of the user running Cloudera Migration Assistant server to access the hosts.
 - Choose New if you want to use a newly provided SSH key to configure Ansible automatically.
 - a. Provide the SSH User and SSH Port.
 - b. Copy the SSH Key to the SSH Key box or upload a .pem file containing the key.
 - c. If you want to use Password, provide the password and make sure the sshpass utility is installed.



Note: The user specified during SSH configuration must have **passwordless sudo rights**.

9. Provide the following information based on the cloud provider of your cluster:

- S3: S3 Bucket Access Key and S3 Bucket Secret Key
- ABFS: Client Id, Client Secret Key and Tenant Id

10. Click Create.

Results

The registered Cloudera cluster on cloud is listed on the **Clusters** page.

Status	Name	Platform	Provider	Type	Actions
Running	sbeki-1s222	CDH	AWS	Source	
Running	sbeki-223	CDH	AWS	Source	
Running	cma-demo-de-6	CDP PC	AWS	Destination	
Running	cma-cod-sbeki-2	CDP PC	AWS	Destination	

You can review the details and services of the cluster by clicking on the Name of the cluster.

Cluster: cma-demo-de-6

Status: Running

Cloud Manager UI: cma-demo-de-6

Data Hub: cma-demo-de-6

Environment: cma-tgn77

Created: 09/20/2024, 01:21 PM

Services:

- hue, query_processor, zeppelin, KNOX, HBASE
- yarn, sqoop, hdfs, spark_on_yarn, hms
- queuemanager, KAFKA, SOLR, lly, RANGER
- tez, ATLAS, hive_on_tez, oozie, zookeeper

Migrations:

There are no migrations associated with this cluster

Create Migration

What to do next

Start the migration from CDH or Cloudera Base on premises to Cloudera on cloud.

Migrating from source cluster to destination cluster

After registering the source and destination cluster, and labeling the scanned datasets, workloads and applications on the source cluster, you can start the migration process.

About this task

Because migrating data to S3 can take a long time, you can perform multiple migrations between a source and destination cluster to move the data in stages. You can also choose to migrate only part of your data as opposed to all of it. A single Cloudera Migration Assistant server is designed to handle multiple migrations.

Before you begin

Procedure

1. Click Migrations on the left navigation pane.
2. Click Start Your First Migration.
3. Select Cloudera Distributed Hadoop 5, Cloudera Distributed Hadoop 6 or Cloudera Base on premises as **Source Type**.

The registered source cluster is selected by default. You can select any other cluster using the drop-down menu. In case you have not registered a source cluster at this point, click New Source and complete the steps in [Registering the source cluster](#).

4. Click Next.

Cloudera on cloud and the registered destination cluster are selected by default. You can select any other cluster using the drop-down menu. In case you have not registered a source cluster at this point, click New Target and complete the steps in [Registering the destination cluster](#).

5. Click Next.
6. Click Next to confirm the migration path.
7. Select one or more labels for migration migrate to the destination cluster.

You can select if the migration should Run Now or be completed in a Scheduled Run. Run Now means that all of the datasets and workloads that were selected with the labels are going to be migrated as soon as the process starts. When choosing the Scheduled Run, you can select the start date of the migration, and set a frequency in which the migration process should proceed.

8. Configure the Migration by providing the required information.

The Cloudera workload user and password will be used to access Cloudera Manager of the Cloudera Data Hub cluster in Cloudera on cloud. You also must set the S3 Bucket Base Path for HDFS or Cloud Storage Path when migrating HDFS data. Some migration require SSH login credentials with sudo-less password.

New Cluster

Configuration Preferences

☐ Use existing
Select this if you want to use your own SSH configuration to access host

☒ New
Requires SSH key or password to automatically configure Ansible

SSH User
cma

SSH Port
22

To use password, the 'sshpass' utility must be installed

☐ Use SSH Key ☒ Use password

SSH Password

S3

S3 Bucket Access Key
AKIAYQAL4K7VFNVCPI

S3 Bucket Secret Key

S3 Credential Name Prefix
cma-jsz6xz

Create Previous Cancel

The remaining service specific settings on the **Configurations** page are automatically filled out, but can be changed based on your requirements.

9. Click Next.

10. Review the information on the Overview page and ensure that the information is correct.

At this point, you can go back and change any configuration if the information is not correct.

Migrate to CDP Public Cloud

Source

Source URL
Cluster: sbeki-ls222
SSH user
SSH port: 22

Cloudera Data Platform - Public Cloud

Control Plane URL
Data Hub Cluster: cma-demo-de-6
SSH user
SSH port: 22

Create Back

- 11.** Click Create to save the migration plan.. You can follow the progress of creating the migration plan.
- 12.** Click Go to Migrations, and select the created CDH to Cloudera on cloud or Cloudera Base on premises to Cloudera on cloud migration.

13. Click Execute Migration to start the migration on the **Migration Dashboard**.

Overview

Clusters

Migrations

CDH to CDP PC

Migration ID

Summary

PENDING CONFIGURATIONS 0%

TYPE Migration

SOURCE CDH 6.3.5

DESTINATION CDP PC 7.2.17

Services

HIVE 6.3.5 → 7.2.17

HDFS 6.3.5 → 7.2.17

HBASE 6.3.5 → 7.2.17

OOZIE 6.3.5 → 7.2.17

SPARK 6.3.5 → 7.2.17

Collections

View Datasets →

Name	Hive tables	HDFS locations	HBase tables	Oozie Job Definitions	Spark Execution
Collection's name	1	8	1	1	1
Collection's name	1	1	1	1	1

Drake Cooper

Before you can start the migration, you can update the Cluster, S3, HDFS and YARN specific settings on the **Configuration** tab.

Overview

Clusters

Migrations

CDH to CDP PC

Migration ID

Highlights

PENDING CONFIGURATIONS 0%

SERVICE Replication Manager

SOURCE CDH 6.3.5

DESTINATION CDP PC 7.2.17

Configuration Mapping Execution Steps

Cluster

S3

HDFS

YARN

Cluster

*Cluster name cma-demo-de-2

*Target version CDP PC 7.2.17

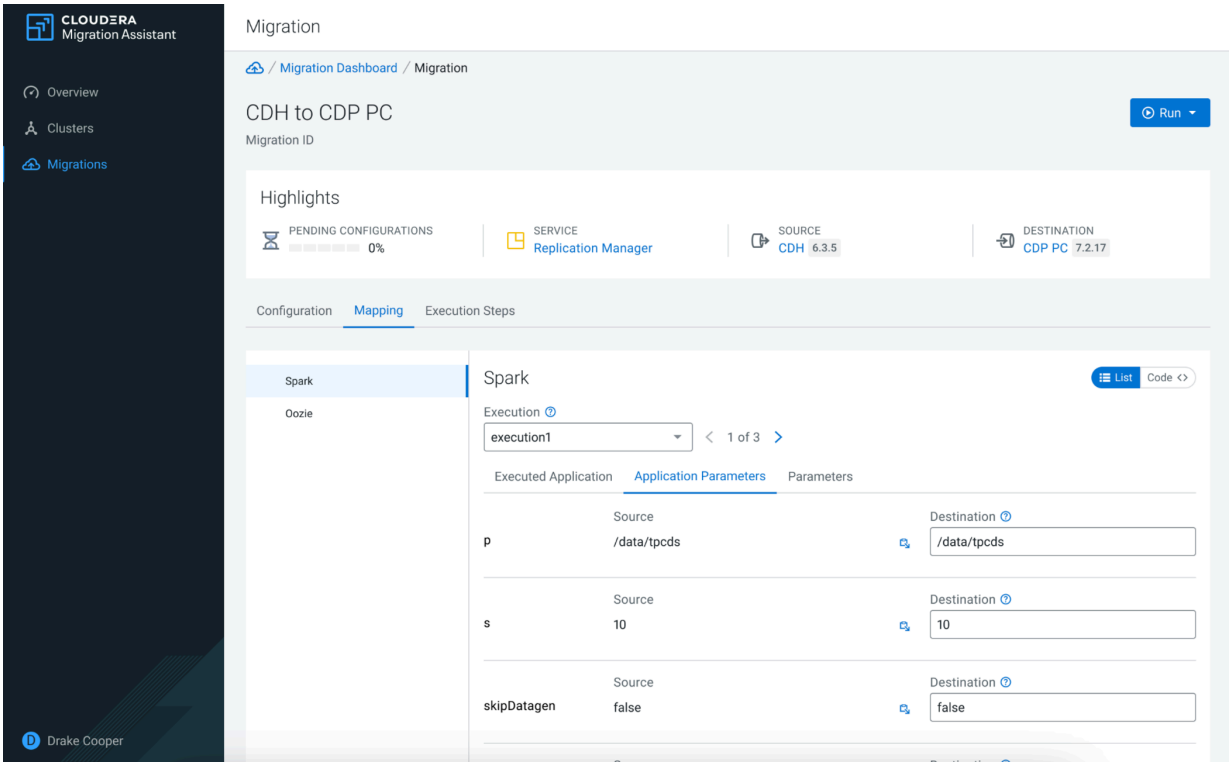
*Default label CMA-3.4-SNAPSHOT

*Classic Cluster DataCenter

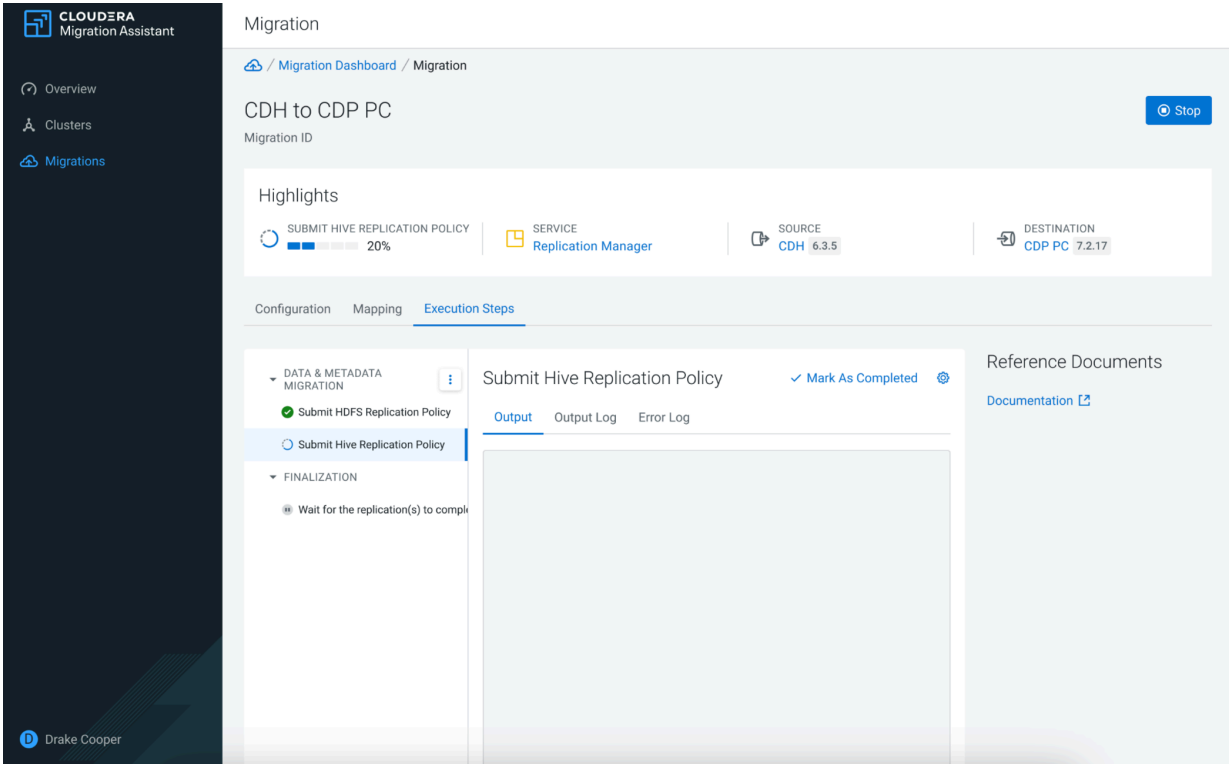
Save

Drake Cooper

Under Mapping, you can ensure that the datasets and applications from the source cluster are in a compatible format on the target cluster.



After finishing the configuration and mapping, click Run to start either the whole migration or specific phases of the migration. You can track the process of each migration step using the **Execution Steps** tab.



The **Data & Metadata Migration** executes the data migration of the labeled datasets with Replication Manager.

You can also view the migration process of the data and workloads based on the selected services. For example, the **Hive SQL Migration** replicates the Hive SQL queries that were fixed to be Hive complied during the Hive Workload migration steps.

The **Finalization** waits until all the Cloudera Replication Manager policies complete their jobs. If the label is created as a frequently scheduled migration, the Cloudera Replication Manager waits only for the first jobs.

When migrating from Cloudera Base on premises to Cloudera on cloud, you need to manually export and import the Ranger policies from the source cluster to the destination cluster using the following curl commands:

- Exporting policies

- To export all policies:

```
curl -X GET --header "text/json" -H "Content-Type: text/json" -o file.json -u [***USERNAME***]:[***PASSWORD***] http://[***HOSTNAME***]:[***RANGER PORT***]/service/plugins/policies/exportJson"
```

- To export for specific HDFS resource:

```
curl -X GET --header "text/json" -H "Content-Type: text/json" -o file.json -u [***USERNAME***]:[***PASSWORD***] http://[***HOSTNAME***]:[***RANGER PORT***]/service/plugins/policies/exportJson?resource%3Apath=[***PATH NAME***]"
```

- To export for policies for specific resource such as Hive database and Hive column:

```
curl -X GET --header "text/json" -H "Content-Type: text/json" -o file.json -u [***USERNAME***]:[***PASSWORD***] http://[***HOSTNAME***]:[***RANGER PORT***]/service/plugins/policies/exportJson??resource%3Adatabase=[***DATABASE NAME***]&resource%3Acolumn=[***COLUMN NAME***]"
```

- Importing policies

- To Import policies from JSON file without servicesMap:

```
curl -i -X POST -H "Content-Type: multipart/form-data" -F 'file=@/path/file.json' -u [***USERNAME***]:[***PASSWORD***] http://[***HOSTNAME***]:[***RANGER PORT***]/service/plugins/policies/importPoliciesFromFile?isOverride=true
```

- To Import policies from JSON file with servicesMap:

```
curl -i -X POST -H "Content-Type: multipart/form-data" -F 'file=@/path/file.json' -F 'servicesMapJson=@/path/servicesMapping.json' -u [***USERNAME***]:[***PASSWORD***] http://[***HOSTNAME***]:[***RANGER PORT***]/service/plugins/policies/importPoliciesFromFile?isOverride=true
```

Results

The datasets and workloads selected are migrated from CDH or Cloudera Base on premises to Cloudera on cloud.

Migrating Spark applications

During the Spark workflow migration, the job JAR file, job properties and other Spark job related data are migrated from a CDH or Cloudera Base on premises cluster to a Cloudera Data Hub cluster.

About this task

Before the migration, the source cluster is scanned to collect the Spark application JAR files. During the migration process, the Spark applications are not affected on the source cluster and can remain in running state. The source code of the Spark applications also remain the same, only the JAR files are copied from the Source cluster to the Destination cluster. To migrate the files that have dependency to the Spark application, you need to ensure to select them during the migration creation. When the migration is finished, the job definitions are stored in the S3 bucket and the application properties are stored in the local filesystem.



Note: The following limitation apply to the Spark migration:

- No refactoring of the Spark application's source code for newer Spark versions
- JAR and Python package dependencies are not migrated automatically
- No R is supported
- Dependency discovery relies on spark-submit parameters **hdfs locations** only

Before you begin

- Ensure that Cloudera Migration Assistant is set up correctly using the steps in [Setting up Cloudera Migration Assistant server](#).
- Ensure that you have met the requirements detailed in [Reviewing prerequisites before migration](#).
- Ensure that you have a CDH 5, CDH 6 or Cloudera Base on premises cluster registered as a source from which you want to migrate your Spark applications. If you do not have a source cluster yet, complete the steps in [Registering source clusters](#).
- Ensure that you have a Cloudera Data Hub cluster registered as a destination cluster to which you want to migrate your Spark applications. If you do not have a destination cluster yet, complete the steps in [Registering destination clusters](#).

Procedure

1. Click on the CDH or Cloudera Base on premises cluster you want to use for the migration on the **Clusters** page.
2. Click Start Scanning to open the **Scan Settings**.
3. Select Spark application scan.



Important: In case you want migrate the Spark application dependencies, such as HDFS files, you need to make sure to select the dependent services during the Source cluster scanning, and add the dependent files and configurations to the same Collection as the Spark applications.

- a) Provide the Number of latest days to scan to define the period from which the Spark applications are collected.
- b) Click Scan selected.

You will be redirected to the scanning progress, where you can monitor if the scanning process was successful or encountered any error.

4. Click on Start Mapping to view the collected job definitions when the scan is finished.
5. Add Spark workloads to Collections.

Collections serve as an organization method to sort and bundle the job definitions into groups for the migration. You can create more collections beside the Default collection based on your requirements.

After you are finished with sorting the Spark workloads and their dependencies to collections, you can start the migration process by creating the migration plan.

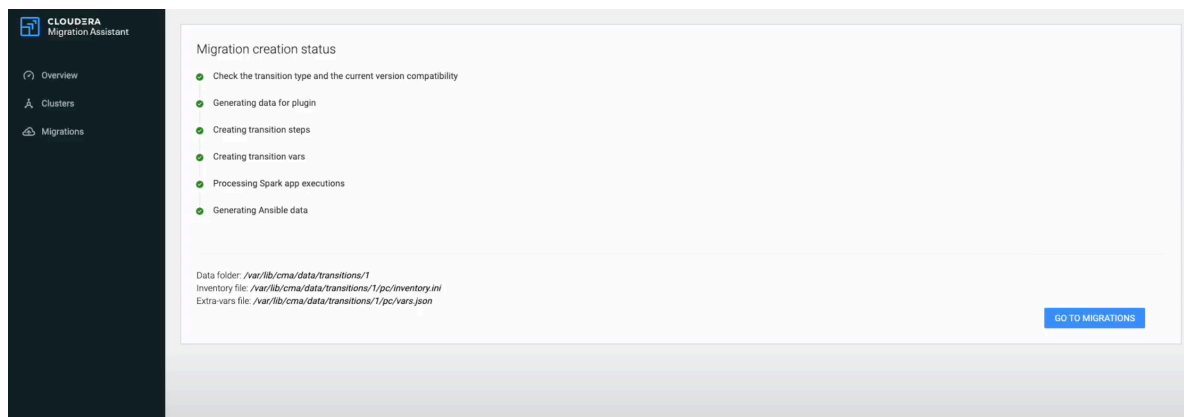
6. Click Create Migration or select Migrations Start Your First Migration .
 - a) Select the source cluster, and click Next.
 - b) Select the destination cluster, and click Next.
 - c) Select the type of migration, and click Next.
 - d) Select the collections that you want to migrate, and click Next.

You can select if the migration should Run Now or be completed in a Scheduled Run. Run Now means that the Oozie job definitions in the selected collections are going to be migrated as soon as the process starts. When

choosing the Scheduled Run, you can select the start date of the migration, and set a frequency in which the migration process should proceed.

- e) Provide the Cloudera workload user and password will be used to access Cloudera Manager of the Cloudera Data Hub cluster in Cloudera on cloud.
- f) Ensure that the path of the Folder for the Spark scripts on the target Data Lake is correct.
- g) Click Next.

An overview of the migration plan is displayed. At this point, you can go back and change any configuration if the information is not correct. If the information is correct, click Create.




7. Click Execute Migrations when the migration plan is successfully created.

You can use the **Mapping** tab to update the Spark workloads, and you can view the steps that will be completed during the migration on the **Execution Step** tab.

Under **Configuration**, you can view how the configuration are being mapped from the Source cluster to the Destination cluster. The configurable parameters on the destination cluster are filled out automatically based on how the Destination cluster was configured for the migration, but these parameters can be changed based on your requirements before executing the migration.

8. Click Run to start migration.

You also have the option to click  and select Run All. In this case the migration steps are executed manually. Choosing Run All in Current Phase enables you to manually start the next phase of the migration.

During the Spark migration, the HDFS replication policies are created if there are any HDFS file dependencies for the Spark application. In the next step, the Spark shell scripts are created on the selected cloud storage. During the finalization, there is a manual step to review if the Spark application migration was successful to the Destination cluster. You can choose how to ensure that the Spark applications are on the Destination cluster, but you can use the CLI commands provided on the screen.

What to do next

When all of the steps are successfully completed, the migration of Spark applications from CDH or Cloudera Base on premises to Cloudera on cloud is finished. You can restart the Spark applications on the destination Cloudera Data Hub cluster using Command Line Interface (CLI) or via Livy as instructed in the last step.

Configuration Mapping **Execution Steps**

- Spark App migration
 - Copy Spark dependent files to the cloud storage
 - Copy Spark shell scripts to the cloud storage
- Finalization
 - Spark Execution Instruction**

Spark Execution Instruction

✓ Mark As Completed

Manual step

If the Spark migration is completed, the executable spark submit scripts are on the target datalake:
s3a://aws-eng-cma-qe-bucket/cc-cma-z3xgvv/usr/cma/spark

In the scope of these steps workload user and password will be used: [Documentation](#)

- For execute a script enter to any cluster host under workload user and password:
ssh csso_szabolcs.beki@cma-jvo2n4-master0.cma-z3xg.svbr-nqvp.int.cldr.work
- Check the executable scripts:
hdfs dfs -ls s3a://aws-eng-cma-qe-bucket/cc-cma-z3xgvv/usr/cma/spark/run_spark*.sh
- Use the following command to execute a script:
hdfs dfs -cat s3a://aws-eng-cma-qe-bucket/cc-cma-z3xgvv/usr/cma/spark/<script name> | exec sh

You may need to add extra parameters or to do kinit for authenticating to the hdfs and make the commands work.

It is possible to call the same executions remotely via Livy API endpoint. curl cli command needs to used the following way.

- Application WordCount (application_1736165156769_0021)
curl -L -k -s -u csso_szabolcs.beki -p https://cma-jvo2n4-master0.cma-z3xg.svbr-nqvp.int.cldr.work/cma-jvo2n4/cdp-proxy-api "name": "WordCount (application_1736165156769_0021)", "file": "s3a://aws-eng-cma-qe-bucket/cc-cma-z3xgvv/usr/cma/spark/home/csso_szabolcs.beki/git/szabi-spark-examples/target", "args": ["s3a://aws-eng-cma-qe-bucket/cc-cma-z3xgvv/user/csso_szabolcs.beki/spark/szabi-spark-examples/text/large.txt", "conf": {}, "className": "com.cloudera.migration.cma.verification.WordCount"]
- Application Spark Pi Simple - Python (application_1736165156769_0020)
curl -L -k -s -u csso_szabolcs.beki -p https://cma-jvo2n4-master0.cma-z3xg.svbr-nqvp.int.cldr.work/cma-jvo2n4/cdp-proxy-api "name": "Spark Pi Simple - Python (application_1736165156769_0020)", "file": "s3a://aws-eng-cma-qe-bucket/cc-cma-z3xgvv/usr/cma/spark/home/csso_szabolcs.beki/git/szabi-spark-examples/src/ma", "args": ["100", "22"], "conf": {}

Migrating Oozie workflows

During the Oozie workflow migration, the job definitions, job properties and other Oozie job related data are migrated from a CDH or Cloudera Base on premises cluster to a Cloudera Data Hub cluster.


About this task

Before the migration, the source cluster is scanned to collect the workflows, coordinators, bundles and discover the relations between them. You also have the option to parse the Hive SQL files to obtain the related databases and tables names. During the migration process, the Oozie jobs are not affected on the source cluster and can remain in running state. When the migration is finished, the job definitions are stored in the S3 bucket and the job properties are stored in the local filesystem.

Before you begin

- Ensure that Cloudera Migration Assistant is set up correctly using the steps in [Setting up Cloudera Migration Assistant server](#).
- Ensure that you have met the requirements detailed in [Reviewing prerequisites before migration](#).
- Ensure that you have a CDH 5, CDH 6 or Cloudera Base on premises cluster registered as a source from which you want to migrate your Oozie workflows. If you do not have a source cluster yet, complete the steps in [Registering source clusters](#).
- Ensure that you have a Cloudera Data Hub cluster registered as a destination cluster to which you want to migrate your Oozie workflows. If you do not have a destination cluster yet, complete the steps in [Registering destination clusters](#).

Procedure

1. Click on the CDH or Cloudera Base on premises cluster you want to use for the migration on the **Clusters** page.
2. Click Start Scanning to open the **Scan Settings**.
3. Select Oozie workflow scan.
 - a) Provide the Number of latest days to scan to define the period from which the Oozie jobs are collected.
 - b) Click Scan selected.
You will be redirected to the scanning progress, where you can monitor if the scanning process was successful or encountered any error.
4. Click on Oozie Job Definitions to view the collected job definitions when the scan is finished.
You have the option to analyze the Hive scripts when you migrate Oozie jobs that depend on Hive SQL files. In this case, Cloudera Migration Assistant scans and identifies the SQL file location stored either in HDFS or other custom directories, and adds the SQL files to the migration plan.
 - a. Enable Run Hive3Parser.
 - b. Select the Oozie jobs to analyze.
 - c. Click .


After the scan is completed, the Hive scripts related to the selected Oozie jobs are listed under Hive SQL tab.
5. Add the Oozie job definitions to Collections.
Collections serve as an organization method to sort and bundle the job definitions into groups for the migration. You can create more collections beside the Default collection based on your requirements. The Hive scripts that belong to the Oozie job definitions are automatically added to the same collection.
After you are finished with sorting the job definitions to collections, you can start the migration process by creating the migration plan.
6. Click Create Migration or select Migrations Start Your First Migration .
 - a) Select the source cluster, and click Next.
 - b) Select the destination cluster, and click Next.
 - c) Select the type of migration, and click Next.
 - d) Select the collections that you want to migrate, and click Next.
You can select if the migration should Run Now or be completed in a Scheduled Run. Run Now means that the Oozie job definitions in the selected collections are going to be migrated as soon as the process starts. When choosing the Scheduled Run, you can select the start date of the migration, and set a frequency in which the migration process should proceed.
 - e) Provide the Cloudera workload user and password will be used to access Cloudera Manager of the Cloudera Data Hub cluster in Cloudera on cloud.
 - f) Enable Oozie service configuring to prepare Oozie service on destination cluster for running jobs to include a service preparation step during the migration process.
You can set the paths used by Oozie services. These paths are used when configuring the Oozie service for migration.
 - g) Click Next.
An overview of the migration plan is displayed. At this point, you can go back and change any configuration if the information is not correct. If the information is correct, click Create.
7. Click Execute Migrations when the migration plan is successfully created.
8. Review and configure the Oozie job definitions under **Mapping** before starting the migration process.
 - a) Select a job definition to list the corresponding **Job properties** and **Workflow**.
The original and proposed values are filled out based on the source and destination cluster information.
 - b) Modify the values of the job definition based on the warnings highlighted in the **Workflow** diff view. You can save the job definition changes using the Save button.
Cloudera Migration Assistant typically looks for configuration values that are related to service endpoints, Kerberos principals, and so on. These configuration values are used to update the file locations and other

configurations accordingly. While the automatic changes work without any reservation, ensure to review the propositions and update the configurations based on the destination cluster requirements. The following properties and values should be reviewed before the migration:



- HDFS file paths changed to S3 or ABFS
 - Hostnames
 - Service settings
 - Paths to user-related directories
- c) Click Save property changes to update the configurations.

You have the option to save the changes for only the edited jobs or apply the changes to all of the jobs.

9. Click Run to start migration.

You also have the option to click  and select Run All. In this case the migration steps are executed manually. Choosing Run All in Current Phase enables you to manually start the next phase of the migration.

During the Hive SQL migration, the Hive scripts are copied to the Hive S3 bucket on the destination cluster.

When the Hive SQL Migration is finished, click  to start preparing the Oozie service on the destination cluster for running the jobs that are stored in S3. When the service preparation is finished, click  to start uploading the job definitions and configurations to the local file system and S3 bucket.

What to do next

When all of the steps are successfully completed, the migration of Oozie job definitions from CDH or Cloudera Base on premises to Cloudera on cloud is finished. You can restart the Oozie jobs on the destination Cloudera Data Hub cluster using Command Line Interface (CLI) or Hue.

Migrating SQL queries

During the Hive migration beside the SQL query, the query related tables and data are also migrated from a CDH or Cloudera Base on premises cluster to a Cloudera Data Hub cluster.

About this task

Before the migration, the source cluster is scanned to collect the SQL queries, tables and data from Hive or Impala. This migration can be used in cases when there is a heavy SQL query load and you want to unload the less time sensitive queries to another cluster. Using the scheduling feature of the underlying Cloudera Replication Manager, you can keep the queries in sync between the source and destination cluster. During the migration process, the SQL queries are not affected on the source cluster and can remain in running state.

Before you begin

- Ensure that Cloudera Migration Assistant is set up correctly using the steps in [Setting up Cloudera Migration Assistant server](#).
- Ensure that you have met the requirements detailed in [Reviewing prerequisites before migration](#).
- Ensure that you have a CDH 5, CDH 6 or Cloudera Base on premises cluster registered as a source from which you want to migrate your Hive queries. If you do not have a source cluster yet, complete the steps in [Registering source clusters](#).
- Ensure that you have a Cloudera Data Engineering Data Hub cluster registered as a destination cluster to which you want to migrate your Hive queries. If you do not have a destination cluster yet, complete the steps in [Registering destination clusters](#).

Procedure

1. Click on the CDH or Cloudera Base on premises cluster you want to use for the migration on the **Clusters** page.
2. Click Start Scanning to open the **Scan Settings**.

3. Select Hive table scan, Hive table check and Hive workflow scan.

a) Provide the Hive query parser input.

You can pre-scan Hive2 SQL queries against Hive3 with the Hive Workflow scan option. When selecting this Hive Workflow option, you need to provide the location of your queries as shown in the following example:

- HDFS paths
 - With default namespace: hdfs:///dir/, hdfs:///dir/file
 - With specified namespace: hdfs://namespace1/dir, hdfs://namespace1/dir/file
 - With namenode address: hdfs://nameNodeHost:port:/dir, hdfs://nameNodeHost:port:/dir/file
- Native file paths
 - your/local/dir
 - nodeFQDN:/your/local/dir/sqlFile

b) Click Scan selected.

You will be redirected to the scanning progress, where you can monitor if the scanning process was successful or encountered any error.

4. Click on Hive SQL to view the collected queries when the scan is finished.

You can also find the tables that are related to the queries under Hive tables.

5. Add the Hive queries to Collections.

Collections serve as an organization method to sort and bundle the queries into groups for the migration. You can create more collections beside the Default collection based on your requirements. The Hive tables that belong to the Hive queries are automatically added to the same collection.

After you are finished with sorting the queries to collections, you can start the migration process by creating the migration plan.

6. Click Create Migration or select Migrations Start Your First Migration .

- a) Select the source cluster, and click Next.
- b) Select the destination cluster, and click Next.
- c) Select the type of migration, and click Next.
- d) Select the collections that you want to migrate, and click Next.

You can select if the migration should Run Now or be completed in a Scheduled Run. Run Now means that the Hive queries in the selected collections are going to be migrated as soon as the process starts. When choosing the Scheduled Run, you can select the start date of the migration, and set a frequency in which the migration process should proceed. In case your goal is to keep the queries in sync between the source and destination cluster, select the Scheduled Run with a frequent time period for migration.


- e) Review the default configurations that are filled out automatically.
- f) Click Next.



An overview of the migration plan is displayed. At this point, you can go back and change any configuration if the information is not correct. If the information is correct, click Create.

7. Click Execute Migrations when the migration plan is successfully created.

You can view the steps that will be completed during the migration on the **Execution Step** tab.

8. Click Run to start migration.

You also have the option to click  and select Run All. In this case the migration steps are executed manually. Choosing Run All in Current Phase enables you to manually start the next phase of the migration.

During the Hive SQL migration, a replication policy is created using the Cloudera Replication Manager. When the policy is created, click  to start uploading the SQL migration. At this step, the Hive scripts from the source cluster are copied to the Hive S3 bucket on the destination. When the Hive SQL Migration is finished, click  to finalize the replication policies.

What to do next

When all of the steps are successfully completed, the migration of Hive queries from CDH or Cloudera Base on premises to Cloudera on cloud is finished. You can restart the queries on the destination Cloudera Data Engineering Data Hub cluster using Command Line Interface (CLI) or Hue.

Migrating HBase tables

During the HBase migration, the HBase tables with their related metadata are migrated from a CDH or Cloudera Base on premises cluster to an Cloudera Operational Database Data Hub cluster.

About this task

Before the migration, the source cluster is scanned to collect the HBase tables. During the migration process, the tables are not affected on the source cluster and can remain part of running jobs.

Before you begin

- Ensure that Cloudera Migration Assistant is set up correctly using the steps in [Setting up Cloudera Migration Assistant server](#).
- Ensure that you have met the requirements detailed in [Reviewing prerequisites before migration](#).
- Ensure that you have a CDH 5, CDH 6 or Cloudera Base on premises cluster registered as a source from which you want to migrate your HBase tables. If you do not have a source cluster yet, complete the steps in [Registering source clusters](#).
- Ensure that you have an Cloudera Operational Database Data Hub cluster registered as a destination cluster to which you want to migrate your HBase tables. If you do not have a destination cluster yet, complete the steps in [Registering destination clusters](#).
- Ensure that you have one of the following parcels (together with its corresponding .sha files) procured from Cloudera Support based on which CDH version you use:
 - CLOUDERA_OPDB_REPLICATION-1.0-1.CLOUDERA_OPDB_REPLICATION5.14.4.p0.31473501-el7.parcel
 - CLOUDERA_OPDB_REPLICATION-1.0-1.CLOUDERA_OPDB_REPLICATION6.3.3.p0.8959316-el7.parcel
 - Copy the OpDB Replication Manager parcels to the following directories based on the CDH version:
 - [***CMA ROOT FOLDER***/]parcels/hbase/cdh5/
 - [***CMA ROOT FOLDER***/]parcels/hbase/cdh6/

Procedure

1. Click on the CDH or Cloudera Base on premises cluster you want to use for the migration on the **Clusters** page.
2. Click Start Scanning to open the **Scan Settings**.
3. Select HBase table scan.
4. Click Scan selected.
You will be redirected to the scanning progress, where you can monitor if the scanning process was successful or encountered any error.
5. Click on HBase table to view the collected tables when the scan is finished.
6. Add the needed HBase tables to Collections.

Collections serve as an organization method to sort and bundle the tables into groups for the migration. You can create more collections beside the Default collection based on your requirements.

After you are finished with sorting the tables to collections, you can start the migration process by creating the migration plan.

7. Click Create Migration or select Migrations Start Your First Migration .


- a) Select the source cluster, and click Next.
- b) Select the destination cluster, and click Next.
- c) Select the type of migration, and click Next.
- d) Select the collections that you want to migrate, and click Next.
- e) Provide the Cloudera workload user and password will be used to access Cloudera Manager of the Cloudera Data Hub cluster in Cloudera on cloud.
- f) Click Next.

An overview of the migration plan is displayed. At this point, you can go back and change any configuration if the information is not correct. If the information is correct, click Create.


8. Click Execute Migrations when the migration plan is successfully created.

You can view the steps that will be completed during the migration on the **Execution Step** tab.

9. Click Run to start migration.

You also have the option to click  and select Run All. In this case the migration steps are executed manually. Choosing Run All in Current Phase enables you to manually start the next phase of the migration.

During the HBase table migration, the source cluster is prepared with the HBase Replication Manager parcel.

When the replication policy is submitted, click  to finalize the replication policies.

What to do next

When all of the steps are successfully completed, the migration of HBase tables from CDH or Cloudera Base on premises to Cloudera on cloud is finished. You can start using the HBase tables on the destination Cloudera Operational Database Data Hub cluster with the available services.