

Migration guide for Cloudera Migration Assistant

Date published: 2026-06-18

Date modified: 2026-06-18



Legal Notice

© Cloudera Inc. 2026. All rights reserved.

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

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

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

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

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

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

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

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

Contents

Registering clusters.....	4
Scanning clusters.....	5
Creating collections for migration.....	9
Migrating HDFS data to Ozone.....	12
Migrating Ranger policies [Technical Preview].....	14
Migrating Airflow DAGs from to a Cloudera Workflow Orchestrator Git repository [Technical Preview].....	18

Registering clusters

Clusters Register clusters automatically by deploying and starting the CMA Agent on the source cluster, eliminating manual registration workflows within the CMA Master UI.

About this task



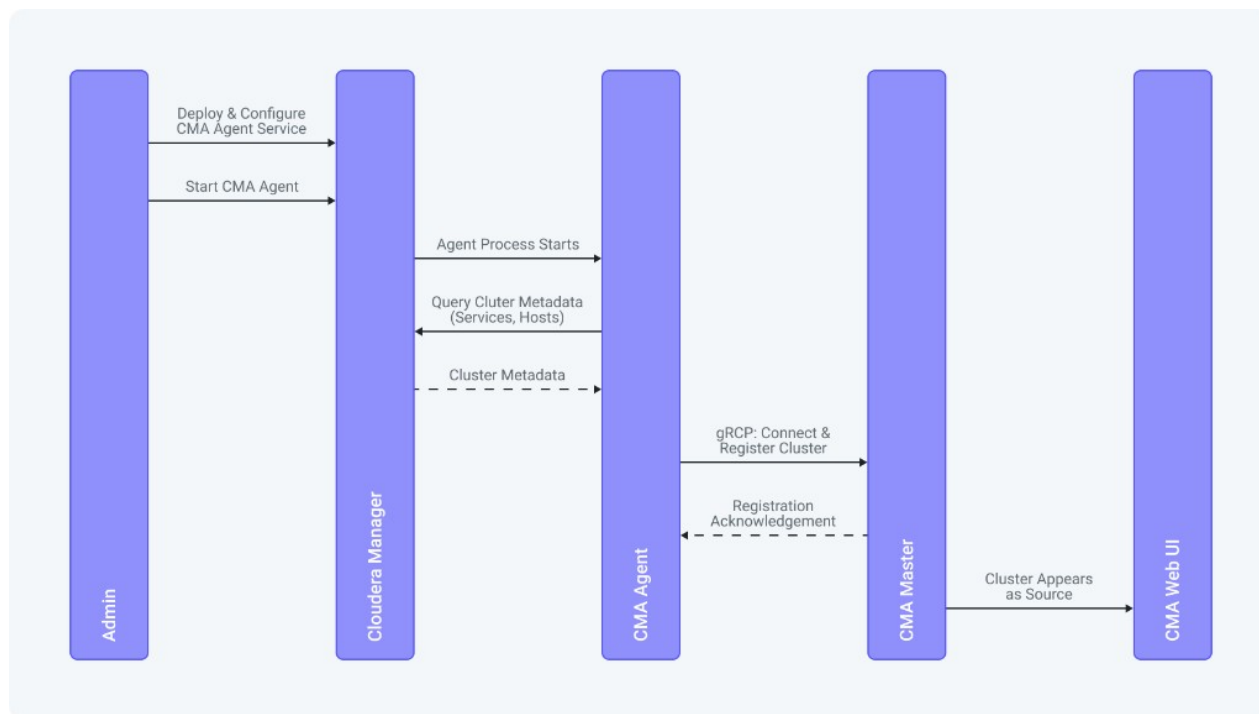
Note: Clusters register automatically when a CMA Agent is deployed and started on the cluster.

This workflow is for engineers who register clusters, scan workloads, use collections, and run migration guides.

Cloudera Migration Assistant uses an agent-based auto-discovery model. When the CMA Agent starts on a cluster, the CMA Agent performs the following actions:

- Connects to the local Cloudera Manager using the configured credentials.
- Discovers the cluster services, hosts, and metadata.
- Establishes an outbound gRPC connection to the CMA Master.
- Registers the cluster with the CMA Master, including discovered metadata.
- Surfaces the cluster automatically on the Clusters page in the CMA Master UI as a source, ready for scanning.

Figure 1: Cluster registration flow



Before you begin

- The CMA parcel must be distributed and activated on the cluster.
- CMA Master must be running and reachable from the source cluster network.
- Cloudera Manager administrator credentials must be available.

Procedure

1. In Cloudera Manager on the source cluster, select Add Service and select CMA_AGENT.

2. Configure the required properties.

Table 1: CMA Agent configuration properties

Property	Description
cma_gateway_url	URL of the CMA Gateway, for example, https://master-host:8093. Required when the CMA Agent is on a different cluster than the CMA Master. The agent uses this URL to discover the Master host, gRPC port, and Auth Server URI automatically through the discovery endpoint (GET /api/agent/config). When connecting from a different cluster, set the CMA Master service dependency to none in Cloudera Manager.
cm_username	Cloudera Manager administrator user name.
cm_password	Cloudera Manager administrator password.
cma_user_home	Home directory for the cma-agent user, for example, /var/lib/cma-agent.

3. Set cma_gateway_url when the CMA Agent is on a different cluster than the CMA Master.
4. Start the CMA Agent service.
5. Wait for the agent to connect to the CMA Master and finish the registration automatically.

Results

The source cluster is displayed on the **Clusters** page in the CMA Master UI. Click the cluster name to review details and services.

What to do next

Scan workloads on the registered source cluster. For instructions, see *Scanning clusters*.

Related Information

[Scanning clusters](#)

Scanning clusters

Scan registered clusters to discover datasets and workloads you can migrate, then organize results into collections.

About this task

You need to scan the source cluster to identify the available datasets and workloads that can be migrated.

Available scan types

The scan options depend on the cluster platform type:

CDP Base clusters:

Scan type	Description
HDFS data scan	Scans HDFS directories and files on the source cluster
Hive table scan	Scans Hive metastore for tables and databases
Ozone data scan	Scans Ozone volumes, buckets, and keys
Ranger policy scan	Scans Ranger access, masking, and row-filter policies (technical preview)

Cloudera Data Services clusters:

Scan type	Description
Cloudera Data Engineering for Airflow	Scans Cloudera Data Engineering virtual clusters for Airflow DAGs and resource files (technical preview)

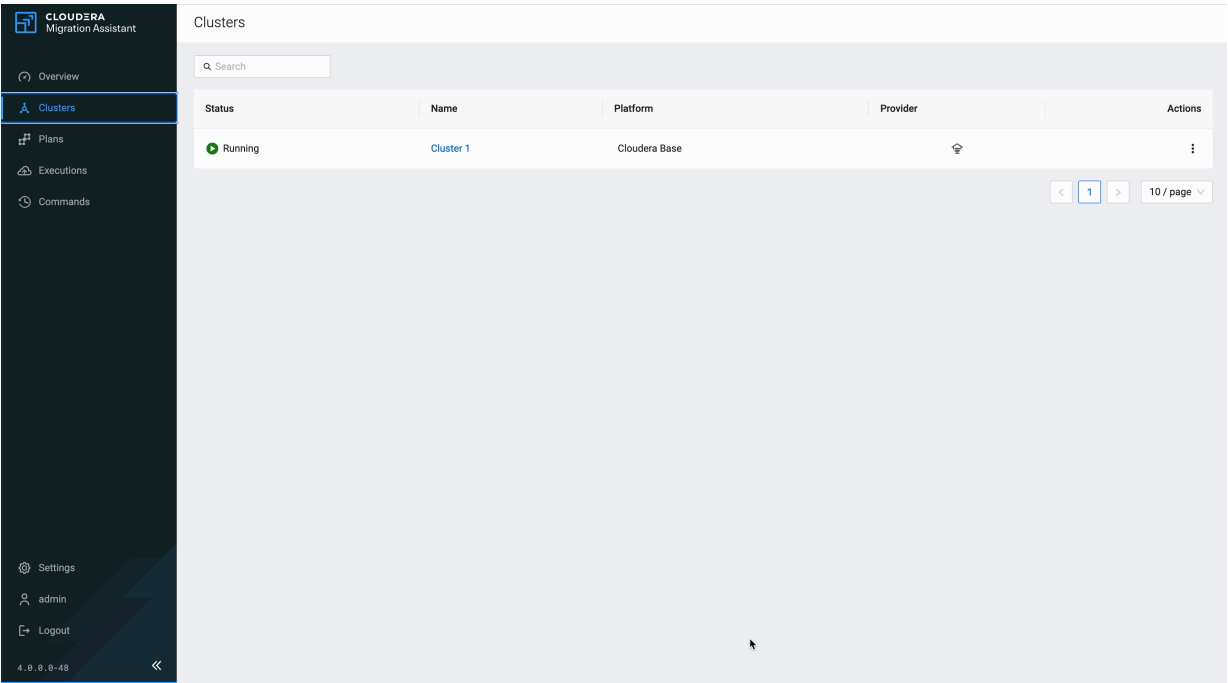
Git clusters:

Scan type	Description
Git repository scan	Scans the target Git repository file tree

Procedure

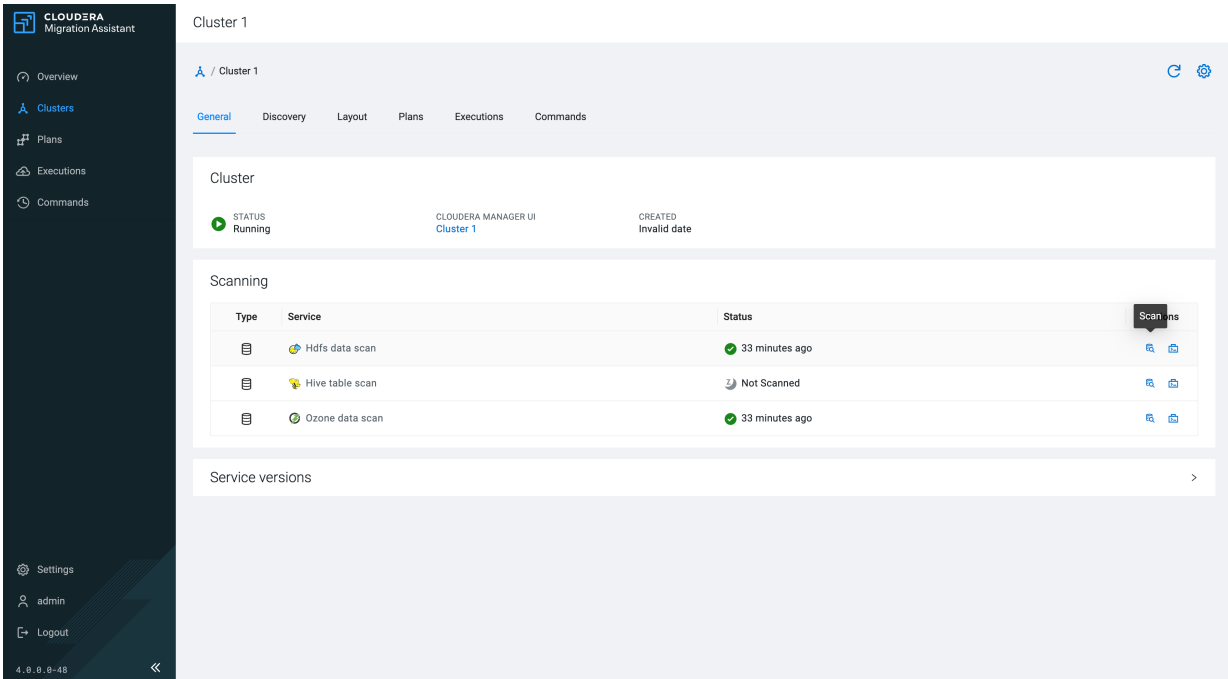
- 1. Navigate to the Clusters page and click the cluster you want to scan.

Figure 2: Cloudera Migration Assistant Clusters page



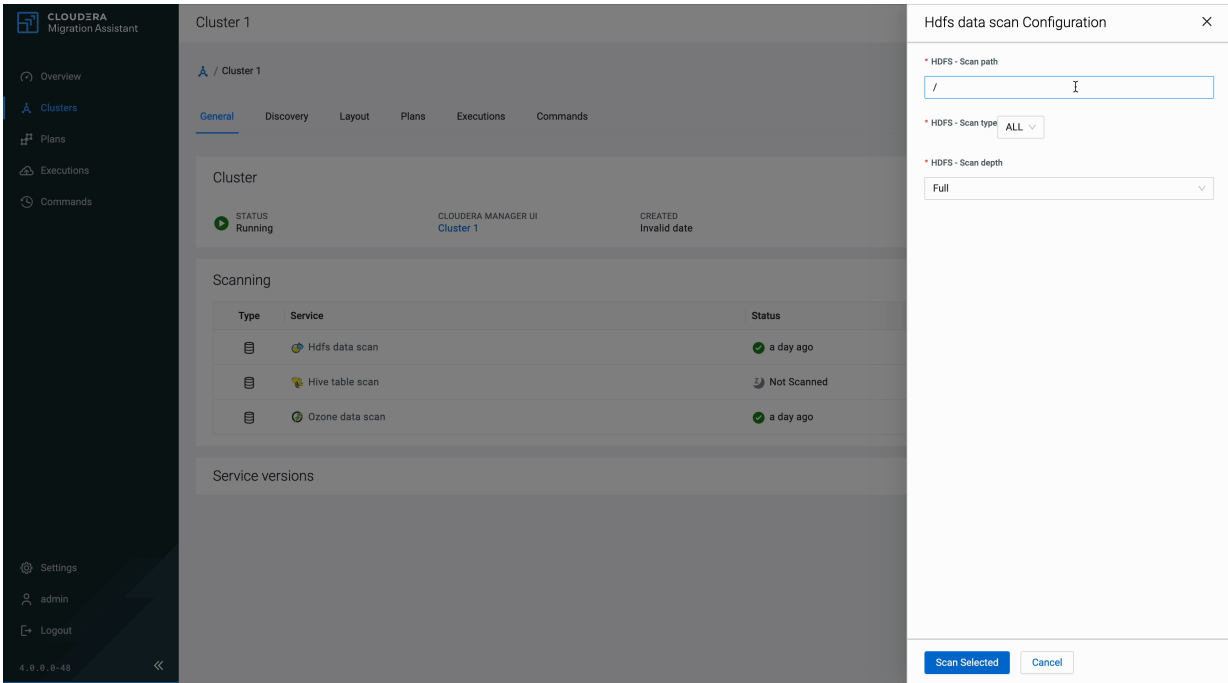
- 2. On the cluster detail page, find the Scanning section, which lists the available scan types for your cluster. Click the configuration icon on the right side of the scan type you want to trigger.

Figure 3: Cloudera Migration Assistant cluster scanning



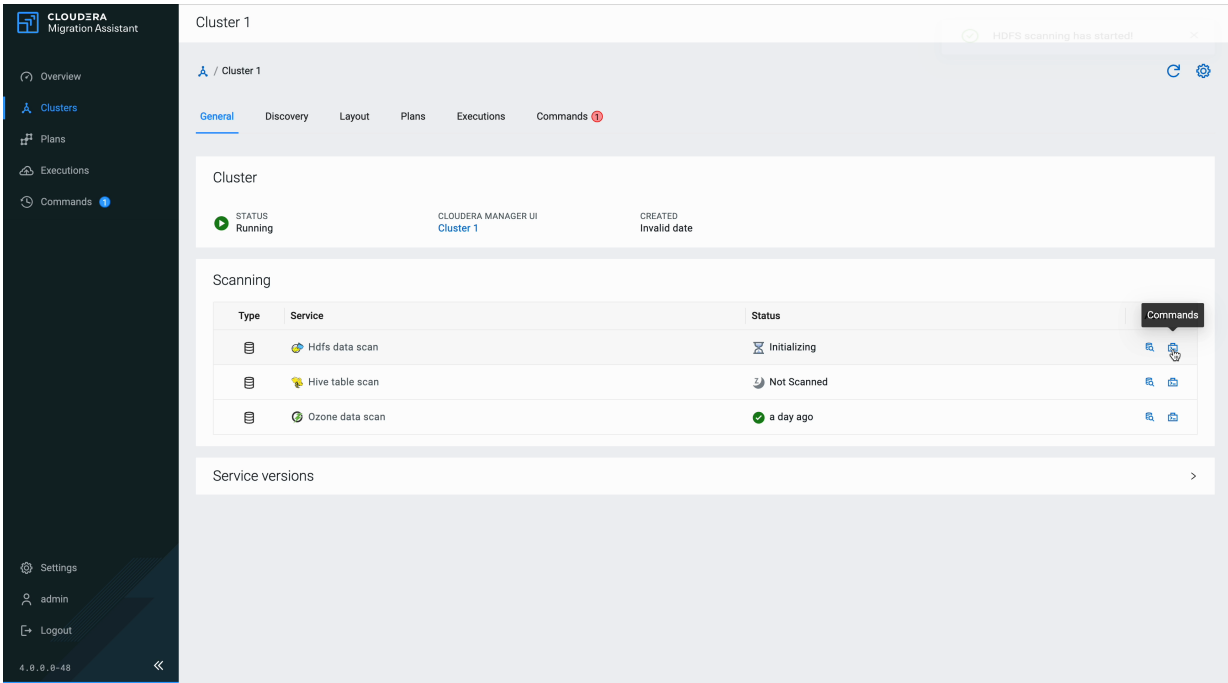
3. In the scan configuration drawer, configure the scan parameters (for example, scan path, scan type, scan depth) and click Scan Selected.

Figure 4: Cloudera Migration Assistant scan configuration



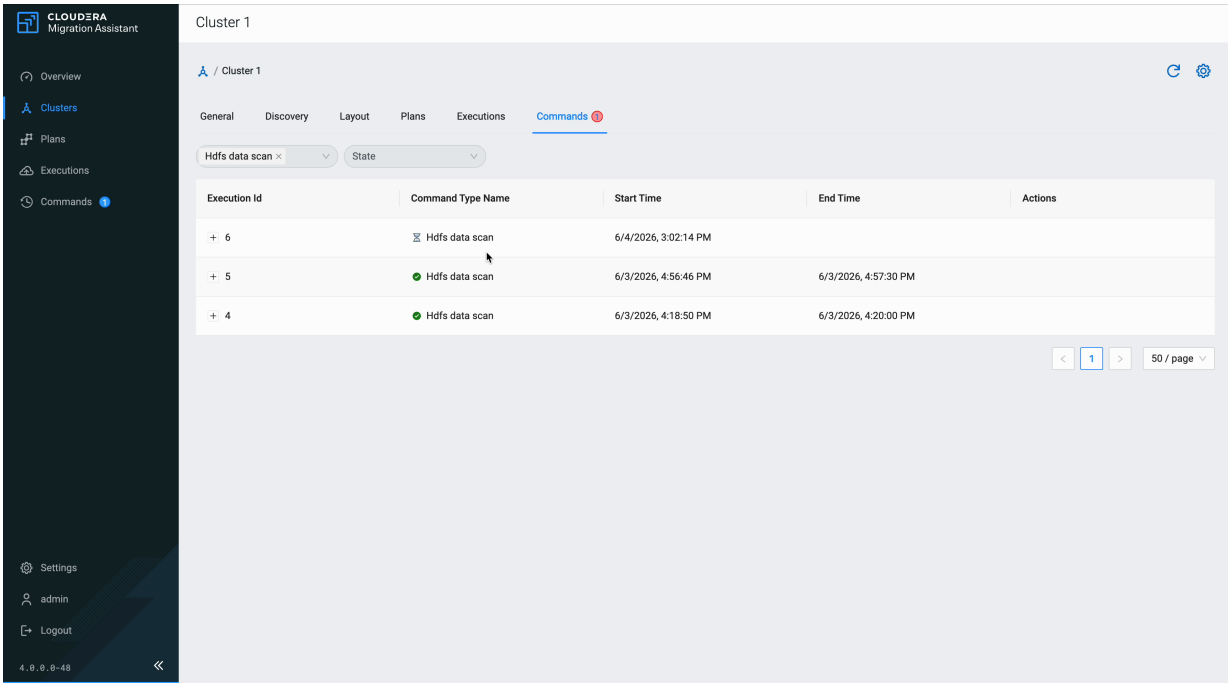
4. The scan starts and the status changes to Initializing. Track scan progress by clicking the Commands tab, which shows a badge with the number of active commands.

Figure 5: Cloudera Migration Assistant scan in progress



5. On the Commands tab, expand a command to view its Output Log, Error Log, and Configuration.

Figure 6: Cloudera Migration Assistant scan commands



Note: Scan results are also available in the data directory on the source cluster at <CMA_ROOT_DIR>/data/sources/ClusterName/Source_ID/cluster-scan/ on the CMA Master host.

Results

Selected datasets and workloads on the scanned cluster finish cataloging according to the scan types you triggered.

What to do next

Rescanning

You can rescan a cluster at any time. Rescanning replaces the previous scan results for the selected scan types. Existing collections that reference scanned items are preserved.

Troubleshooting scan failures

Symptom	Likely cause	Resolution
Scan stays in IN_PROGRESS indefinitely	Agent lost connection during scan	Check agent connectivity, cancel and retry the scan. See <i>Scan failures</i> .
Kerberos authentication error	Agent keytab expired or missing	Verify kerberos.auth.enable and keytab configuration in the CMA Agent service
Scan completes but shows no results	The scanned service is not deployed on the cluster	Verify the service exists in Cloudera Manager on the source cluster
Hive scan fails with connection error	Hive Metastore is not running or not accessible	Verify HMS is running; check agent logs for connection details
HDFS/Ozone scan times out	Scan path is very large (millions of files)	Configure scan depth or narrow the scan path in scan settings
Ranger scan fails	Ranger Admin not accessible or Kerberos SPNEGO auth fails	Verify Ranger is running and the scan user has admin privileges

Sort the scanned data into collections to organize what will be migrated. Then proceed to create a migration plan. For instructions, see *Creating collections for migration*.

Related Information

[Creating collections for migration](#)

Creating collections for migration

After scanning, group datasets and workloads into collections so you control what migrates in each plan execution.

About this task

After scanning the source cluster, use collections to organize the datasets and workloads that need to be migrated. Collections group related items together, allowing you to run phased migrations by selecting which collections to include in a migration plan.

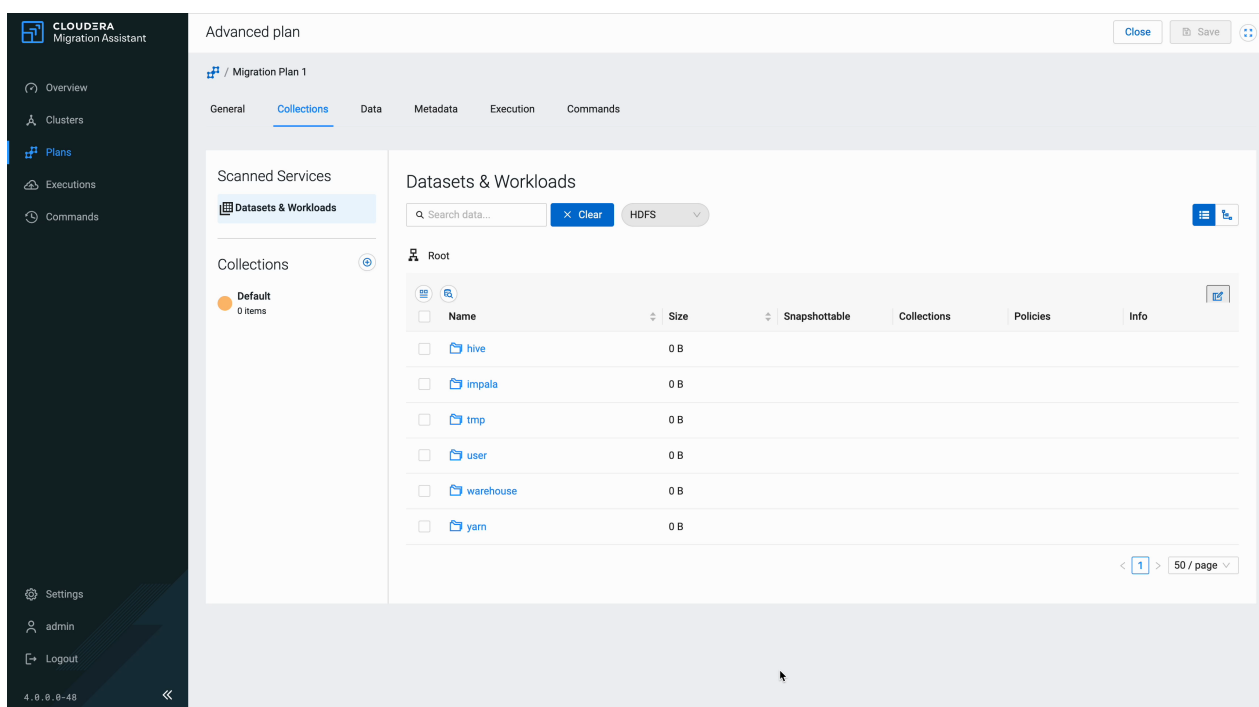
Accessing collections

Collections are managed within a migration plan. Navigate to Plans, open your plan, and click the Collections tab.

The Collections tab shows:

- Scanned Services panel (left) — lists available data sources (Datasets & Workloads)
- Data browser (center) — displays scanned items in a navigable tree, filtered by service type
- Collections sidebar (left, below Scanned Services) — lists all collections with item counts

Figure 7: Cloudera Migration Assistant collections browser



Filtering data

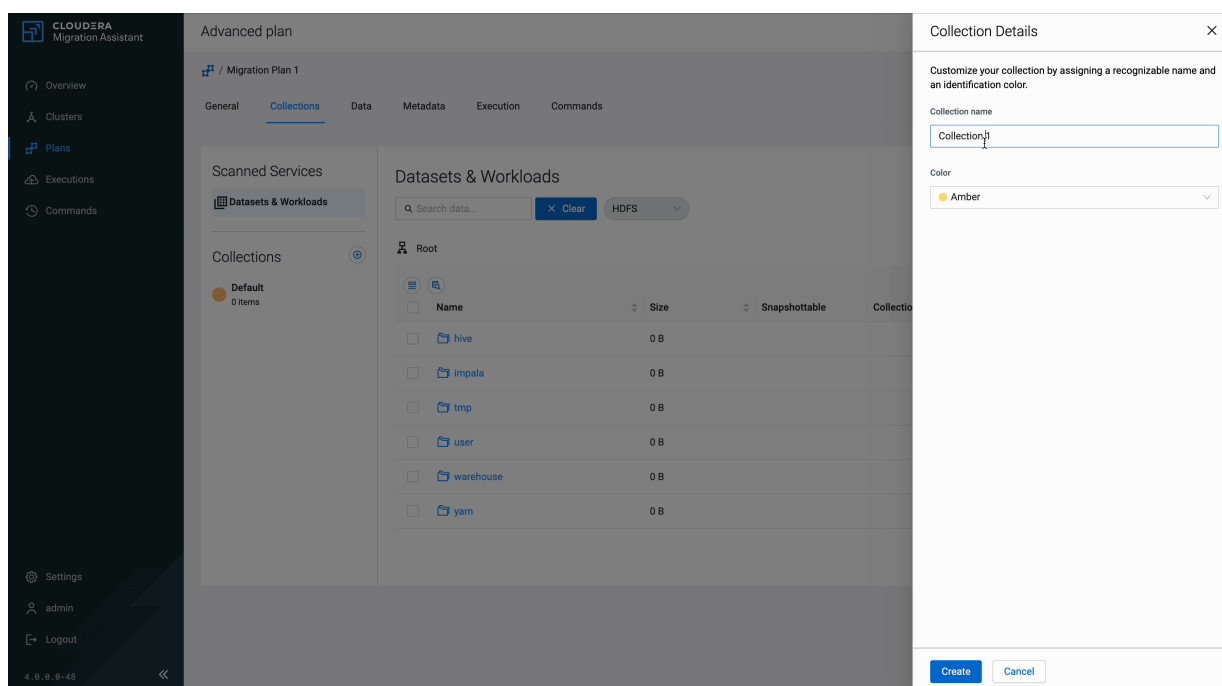
Use the service type dropdown to filter the data browser by scan type:

- HDFS — HDFS directories and files
- Hive — Hive tables and databases
- Ozone — Ozone volumes, buckets, and keys

Procedure

1. Click the + icon next to Collections in the sidebar.
2. In the **Collection Details** drawer, enter a Collection name and select a Color for identification.

Figure 8: Cloudera Migration Assistant create collection

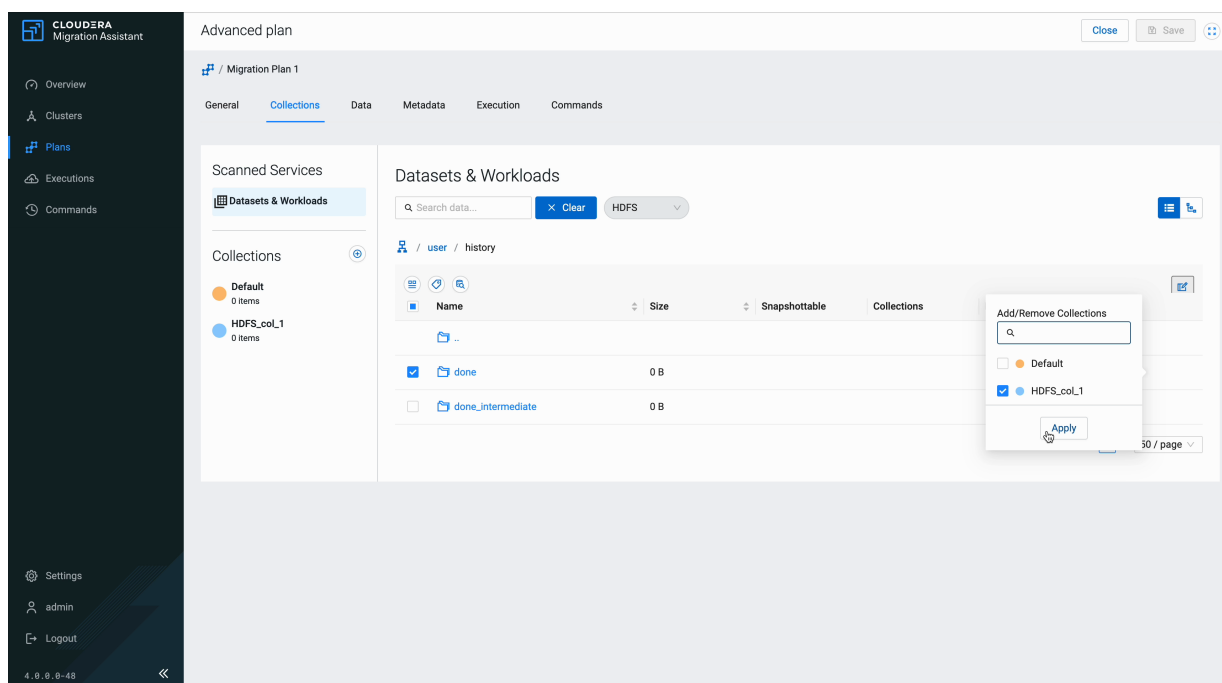


3. Click Create.

The new collection appears in the sidebar with 0 items.

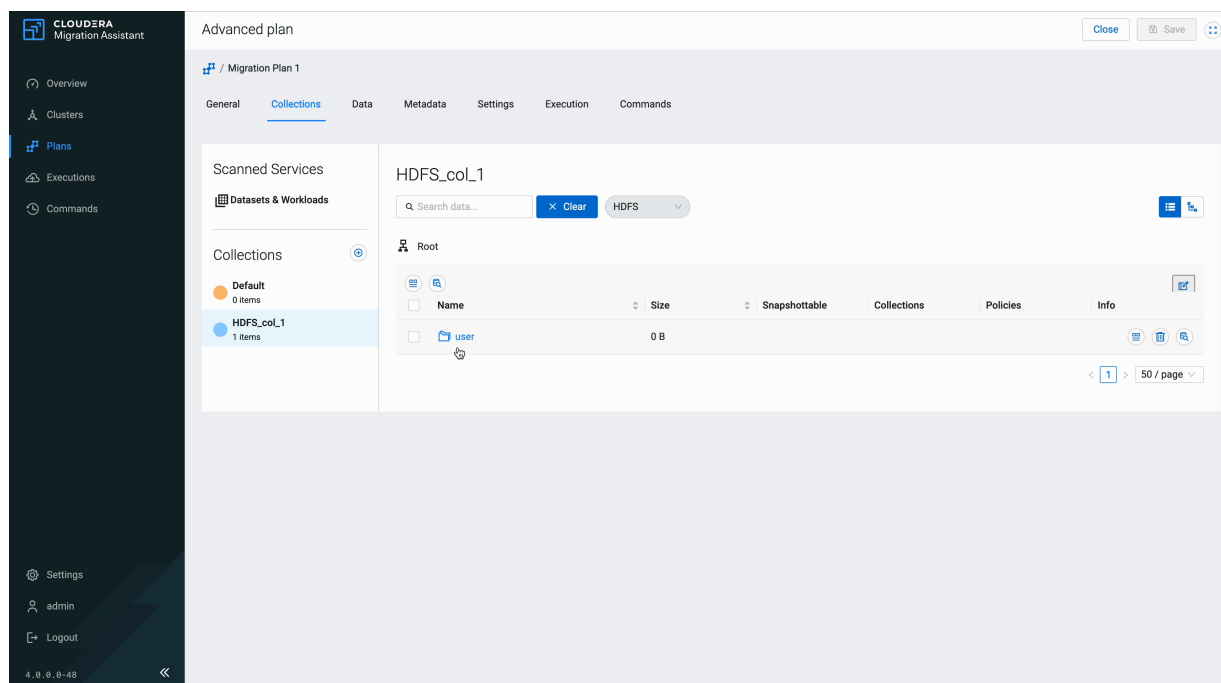
4. Browse the data tree to find the items you want to migrate. Click folder names to navigate deeper into the hierarchy.
5. Select items using the checkboxes next to each item.
6. Click the collection icon that appears on the selected item row.
7. In the **Add/Remove Collections** popup, check the collections you want to assign the item to.
8. Click Apply.

Figure 9: Cloudera Migration Assistant assign to collection



- Click a collection name in the sidebar to view its contents. The data browser switches to show only the items assigned to that collection.

Figure 10: Cloudera Migration Assistant view collection



Results

Collections group scanned datasets for phased migration planning.

What to do next

You can edit or remove collections from the sidebar. Removing a collection does not affect the source data — only the grouping is deleted.

Create a migration plan to begin migrating the collected datasets. See the migration type guides for specific workflows:

- *Migrating HDFS data to Ozone*
- *Migrating Airflow DAGs from Cloudera Data Engineering to a Cloudera Workflow Orchestrator Git repository*
- *Migrating Ranger policies*

Related Information

[Registering clusters](#)

Migrating HDFS data to Ozone

Copy curated HDFS directories from a source cluster to Apache Ozone on a target cluster without altering source data, using Cloudera Replication Manager-backed executions.

About this task

Planning completes with Cloudera Replication Manager replication policies copying every selected directory into deployed Ozone layout targets. The source HDFS data remains intact.

Before you begin

- Both source (HDFS) and target (Ozone) clusters must be displayed on the **Clusters** page. For instructions, see *Registering clusters*.
- The target cluster must host Apache Ozone with the service running.
- You must model and deploy Ozone volumes, buckets, and folders in the **Storage Layout Editor**. Without deployed layout destinations, executions have nowhere to persist data. For instructions, see *Managing Ozone storage*.



Important:

Ozone Layout Provisioning (**Storage Layout Editor**) is a Technical Preview feature in Cloudera Migration Assistant 4.0. CMA does not validate input or pre-check feasibility of settings on the target cluster. Verify your layout definitions manually before deploying.

- CMA Agents must operate on source and target clusters.
- Cloudera Replication Manager must be reachable. Cloudera Migration Assistant submits policies through Cloudera Replication Manager APIs.
- Secure clusters authenticate with Kerberos, so you must confirm that the migration user principal works for filesystem access.

Whenever Discovery trees look stale before mapping, rerun the steps in *Scanning clusters*. Collections exported from scans help bound execution scopes. For more information, see *Creating collections for migration* if you segregate phased cutovers.

Procedure

Step 1 — Scan the source cluster (HDFS)

1. Open Clusters, then select the source cluster.
2. Click Start Scanning.
3. Select Hdfs data scan.
4. Constrain the scan path, for example, /tmp or /user, to shorten runs on massive namespaces.
5. Click Scan selected and watch the progress drawer until completion.

After success, Discovery lists HDFS trees with sizing and ownership snapshots.

Step 2 — Scan the target cluster (Ozone)

6. Confirm Apache Ozone is healthy in Cloudera Manager before scanning.
7. From **Clusters**, open the target cluster.
8. Click Start Scanning.
9. Select Ozone data scan.
10. Click Scan selected and wait until the status turns green.

Ozone hierarchies populate inside Discovery afterward.

Step 3 — Create a migration plan

11. Go to Plans.
12. Click Create Plan.
13. Set Name, select the source HDFS cluster, and the target Ozone cluster.
14. Complete the wizard (Create) and land on the plan detail surface.

Step 4 — Map HDFS directories to Ozone paths on the Data tab

The Data tab displays source HDFS on the left and target storage on the right.

15. Go to Plan Data.
16. Switch the target browser mode from HDFS to Ozone using the service picker.
17. Go to your target volume, bucket, and folder staging area.
18. On the source side, go to the parent directory that contains data you want to migrate.

19. Hover the directory and click Copy. Cloudera Migration Assistant persists a structural mapping.

20. Switch to the **Review** view any time to view a list of your current structure mappings.

21. When Ozone misses a folder UI-side, create it before copying structures.

- a) In the target browser, go to the parent bucket or folder.
- b) Click Create Folder.
- c) Provide the folder name and confirm creation.

Cloudera Migration Assistant updates its metadata immediately and executions push the filesystem change during processing.

22. Repeat the mapping workflow for additional directories whenever you broaden scope.

You can map each directory to separate buckets or combine them into shared buckets. The layout must meet the Cloudera Replication Manager requirements for your tenancy model.

Step 5 — Create and run an execution

23. Open Execution.

24. Select Create Execution.

25. Complete the steps in the configuration wizard (General, Scope, Review), and then click Finish.

26. Open the execution dashboard and trigger Run All.

During execution the following stages occur:

- **Policy creation** – Cloudera Migration Assistant registers Cloudera Replication Manager policies mapping HDFS prefixes to destination Ozone keys.
- **Data replication** – Cloudera Replication Manager invokes DistCp-style transfers governed by Cloudera Replication Manager settings.
- **Verification** – CMA Master tracks step health until completion (COMPLETED or FAILED statuses per step).

Step 6 — Verify the migration

27. Confirm that the execution steps display COMPLETED.

28. Inspect Ozone hierarchies inside Discovery.

29. Validate migrated objects using the Ozone CLI.

```
ozone fs -ls ofs://<service-id>/<volume>/<bucket>/<path>
```

Results



Tip: For common regressions during executions, see *Troubleshooting HDFS to Ozone migration*.

Related Information

[Registering clusters](#)

[Scanning clusters](#)

[Creating collections for migration](#)

[Managing Ozone storage](#)

[Troubleshooting HDFS to Ozone migration](#)

Migrating Ranger policies [Technical Preview]

Scan Apache Ranger policies, display them in Discovery, and group them into collections to evaluate HDFS and Hive access controls before planning your cluster migration.

About this task



Note: This feature is in Technical Preview and is not ready for production deployments. Cloudera recommends trying this feature in test or development environments and encourages you to provide feedback on your experiences.

Ranger policy scanning exports every access, masking, and row-filter policy from the source cluster Ranger Admin. After the scan, the Discovery tab groups policies by Ranger service, such as HDFS, Hive, HBase, or Knox, and decorates HDFS and Hive browsers with indicators that show authorization coverage.

Label policies with collections to stage phased migrations.



Note: Enable the `cma.ranger-policy-migration-enabled` feature flag first. Ask your Cloudera Migration Assistant administrator to turn it on through Settings.

Before you begin

- You must install and configure Cloudera Migration Assistant. For instructions, see *Deployment*.
- The cluster must be displayed on the **Clusters** page. For instructions, see *Registering clusters*.
- You must run Cloudera Base with Ranger deployed on the source cluster.
- You must operate a Kerberized source cluster. Technical preview scanning supports Kerberos SPNEGO only.
- You must ensure that the Ranger feature flag remains enabled throughout the workflow.



Important: The scanning credential must read policies in Ranger Admin. Only policies visible to that user are retrieved.

Procedure

Step 1 — Scan Ranger policies.

1. On the **Clusters** page, select the source cluster.
2. Click Start Scanning to display **Scan Settings**.
3. Select Ranger policy scan.
4. Select Scan selected and wait for Ranger services to finish synchronizing.



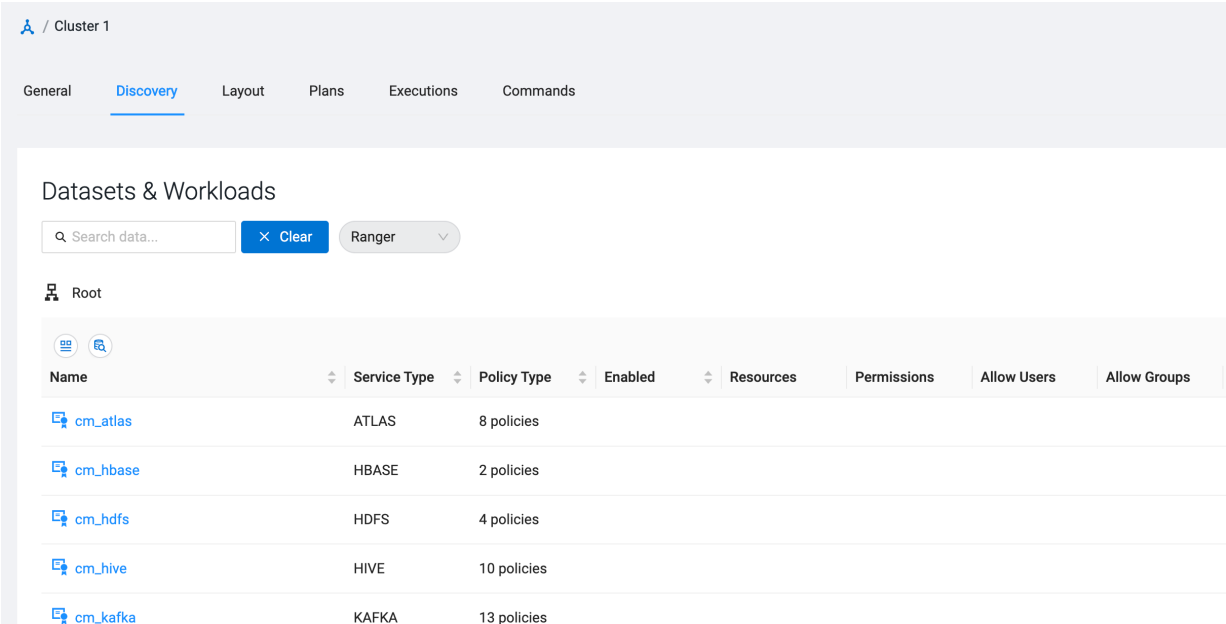
Note: The scan enumerates Ranger services, streams policies back to CMA Master, and captures Ranger users, groups, and roles for reference.

Step 2 — Browse Ranger policies in Discovery.

Finished scans displays policies inside Discovery.

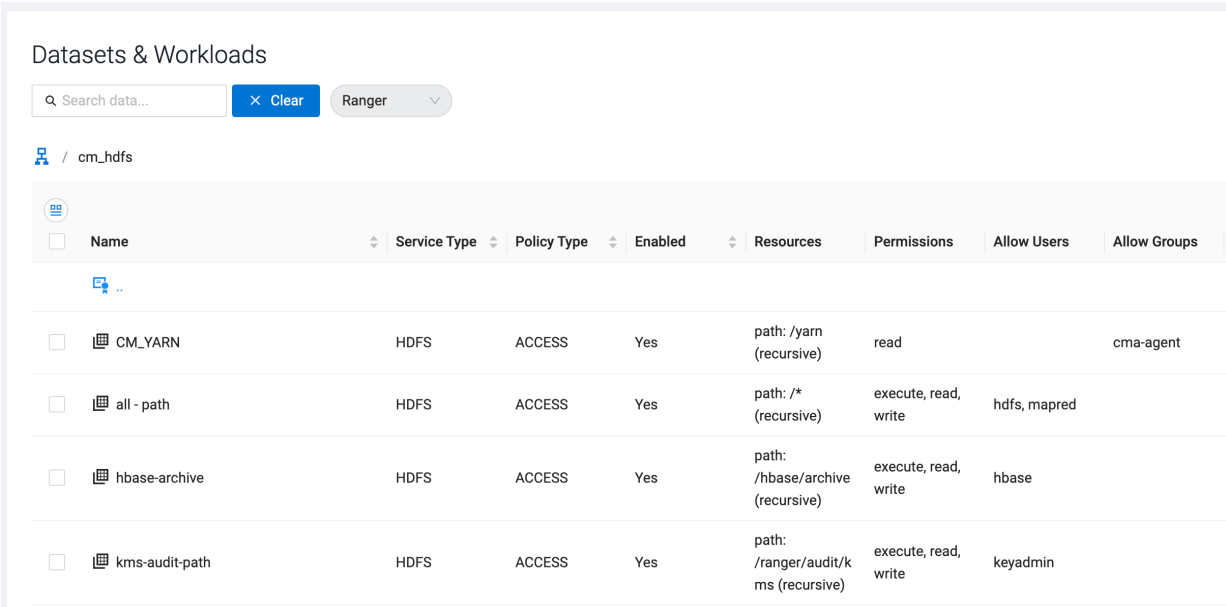
5. Go to the cluster's Discovery tab.
6. Select Ranger Policies in the navigator.
 - **Root** — Lists discovered Ranger services.
 - **Service** — Displays service metadata and policy totals.
 - **Policy** — Renders individual rows in the policy table.

Figure 11: Ranger Discovery tree



7. Expand tree nodes for your Ranger service to view their associated policies.

Figure 12: Policies for one Ranger service



8. Review the default columns in the policy grid to evaluate your policy configurations.

Table 2: Default policy grid columns

Column	Description
Name	The Ranger-defined policy title.
Service Type	The Ranger service type, for example, HDFS, Hive, or HBase.
Policy Type	The operational category of the policy, which is access, masking, or row filtering.
Enabled	The activation state of the policy that is Yes or No.

Column	Description
Resources	The matched paths or tables, including recursive and exclusion directives.
Permissions	The granted privileges.
Allow Users	The users who are granted access by the policy rules.
Allow Groups	The user groups who are granted access by the policy rules.
Collections	The applied collection badges.

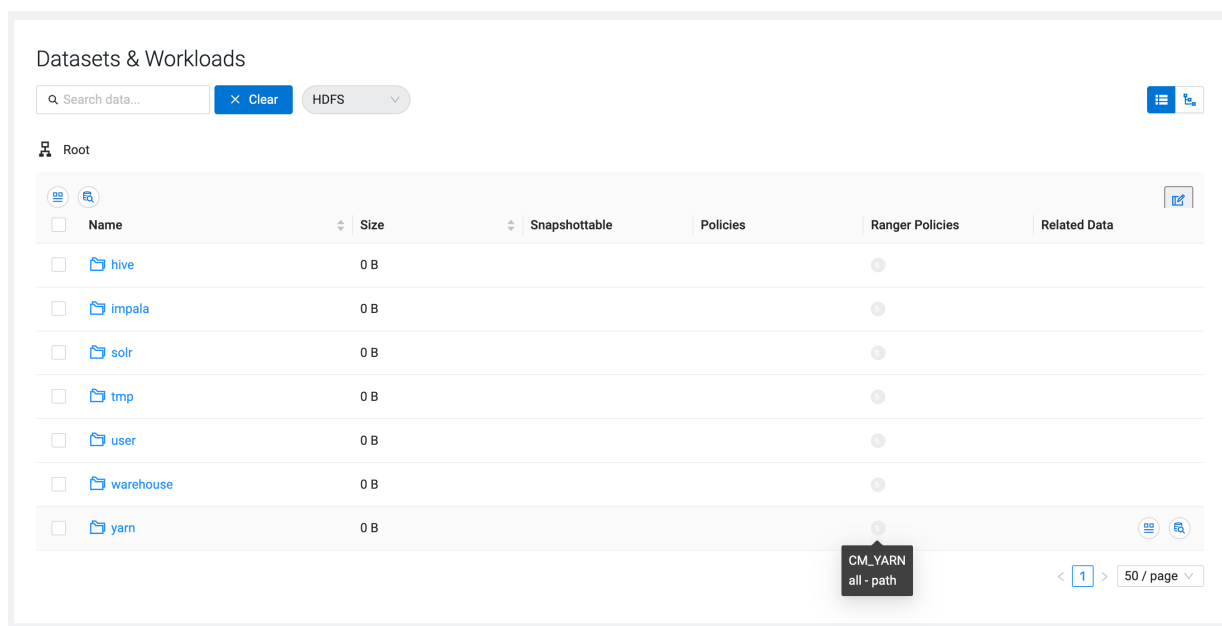
To view additional Ranger metadata, such as policy ID, deny rules, delegate admin, Ranger labels, add the corresponding columns to the grid using the column picker.

Step 3 — Review Ranger policy indicators.

Completed scans populate indicators on filesystem and relational assets.

9. Go to HDFS Locations or Hive Tables within Discovery.
10. Inspect the Ranger Policies column for the Ranger badge.
11. Open the badge to reveal matching policy titles.

Figure 13: Ranger policy indicator on HDFS



12. Understand matching logic when reconciling badges with policies.
 - **Exact match:** The policy resource matches the HDFS path or Hive table name verbatim.
 - **Recursive match:** The policy automatically includes all descendant files and subdirectories.
 - **Wildcard match:** The policy evaluates Ranger patterns, such as *, or ? against the resource path.

Step 4 — Organize policies into collections.

13. In Ranger Policies, select the policies to group.
14. Click Add to collection for the Default label.
15. Click +, enter a unique label and color, then click Create.
16. Apply collection filters whenever you narrow remediation workstreams.

Results

You now have searchable Ranger inventories, enrichment on HDFS and Hive assets, and collection labels framing migration waves.

What to do next

Groom collections, reconcile dependencies, then fold policies into downstream migration executions. For more information, see *Scanning clusters*, *Creating collections for migration*, and *Registering clusters*.

Related Information

[Deployment](#)

[Registering clusters](#)

[Scanning clusters](#)

[Creating collections for migration](#)

Migrating Airflow DAGs from to a Cloudera Workflow Orchestrator Git repository [Technical Preview]

Export curated Cloudera Data Engineering Airflow DAGs and bundled resources into a Cloudera Workflow Orchestrator-managed Git repository with automated Pull Request or Merge Request generation without modifying the remaining Cloudera Data Engineering clusters.

About this task



Note: This feature is in Technical Preview and is not ready for production deployments. Cloudera recommends trying this feature in test or development environments and encourages you to provide feedback on your experiences.

This workflow automatically performs the following actions:

- Opens GitHub/GitHub Enterprise Pull Requests, GitLab Merge Requests, or plain Git branch pushes reflecting your selections.
- Places DAG code under `/dags/<dagId>/` paths.
- Stages supporting artifacts under `/include/<service>/<vc>/<pool>/` prefixes mirrored from Cloudera platform namespaces.

Cloudera Data Engineering remains read-only for migrated assets. The target Git repository governs all future merges.

Before you begin

Administrators must ensure that the following conditions are met before migration:

- Cloudera Data Engineering deployments must be displayed under the **Cloudera Data Services** badge and Git repositories as Git on the **Clusters** page.
- The Cloudera Data Engineering-connected cluster must expose valid Cloudera Control Plane credential pairs.
- Git clusters must advertise repository HTTPS URLs, personal access tokens, and base-branch names.

For operational setup instructions, see *Setting up to Cloudera Workflow Orchestrator Git migration*.

Procedure

Step 1 — Scan the cluster

1. Identify the cluster whose name suffix reads `-cde` and open it.
2. Click Start Scanning.
3. Select Scan Cloudera Data Engineering for Airflow.

- Click Scan selected and wait until the progress indicator displays success.

Reading scan results

Discovery displays CDE Service # Virtual Cluster # Airflow DAGs. Expand each branch to inspect individual items.

Table 3: Discovery page columns

Column	Description
DAG Name	Cloudera Data Engineering job display name.
DAG ID	Unique ID (dag_id) inside the Python script.
Cron	Configured schedule expressions.
Active	Indicator of the DAG schedule status.
Operators	Inferred operators, such as BashOperator.
Triggers	Outbound trigger graph.
Triggered By	Inbound trigger graph.
Resource Pools	Cloudera Data Engineering resource pools allocated for mounted artifacts.

You can use the following interface controls to change how the data is displayed:

- Data – Switches to a file-centric tree, which is ideal before creating mappings.
- Metadata — Switches to the logical view of DAG summaries.
- List/tree toggle – Switches between hierarchical tree and flattened browsing list.



Note: Blank virtual-cluster rows often mean Knox authentication stalled during scan. Check for SKIP PED hints inside the command logs.

Step 2 — Scan the Git Repository

- Open clusters ending in -git.
- Click Start Scanning.
- Select Git Repository Scan, then click Scan selected.

The repository tree displays Name and Size metadata. Empty repositories are acceptable because executions bootstrap directories.

Step 3 — Create a Migration Plan

- On the Plans tab, click Create Plan.
- Select a descriptive name aligned with Git branch naming rules (cma/<planName> prefixes created branches).
- Select the -cde source and -git target clusters.
- Select Cloudera Data Engineering DAGs to Git Repository as transition type, then finalize creation.

Step 4 — Label DAGs on the Collections tab

Collections act as a filter that controls which DAGs are visible on downstream configuration tabs.

- In the Plan tab, go to Collections.
- Expand the Cloudera Data Engineering hierarchies.
- Select DAG rows to migrate.
- Click Add to collection, or click the + icon to create a new collection.



Tip: Revisit Collections whenever scope shifts. Your updates propagate instantly to the downstream mapping tabs.

Step 5 — Map DAG Metadata

- On the Metadata tab, review split panes listing Cloudera Data Engineering (left) and Git (right) DAG stubs.

17. Map individually (Map) or bulk-select unfinished rows (Copy to Git).
18. Defaults land at /dags/<dagId>/<dagId>.py plus /include/<service>/<vc>/<pool>/... artifact bundles.

Unmap removes targets while preserving shared artifacts when safe.



Warning: Cloudera Migration Assistant rejects conflicting dag_id collisions across virtual clusters until you resolve overlaps in Step 7.

Step 6 — Verify File Paths on the Data tab

19. Go to Data to compare mirrored trees.
20. Verify that DAG payloads and supporting files have the expected prefixes.

If you notice any incorrect file paths, go to the Metadata tab to unmap or remap those files.

Step 7 — Review DAG Source (Workloads tab)

21. Open Workloads whenever duplicate-ID badges block executions.
22. Select DAG entries to load Monaco editors with read-only scanned copy displayed in the left pane, editable version in the right pane.
23. Edit the right pane to rename dag_id, revise connections, or prune unsupported operators and save afterward.
24. Ensure that warnings clear across every DAG affected by overlaps.



Important: Renaming dag_id rewires filesystem targets automatically (/dags/<newDagId>/...).

Step 8 — Execute Git Publishing

25. In Execution, click Create Execution.
26. Monitor the sequential pipeline as it automatically processes the following stages:
 - a. **Upload Cloudera Data Engineering resources** – The CMA Agents retrieve scanned artifacts locally (skipped when unnecessary).
 - b. **Prepare Cloudera Data Engineering resources** – The CMA Master merges edited DAG payloads with staging artifacts.
 - c. **Create Git change** – The Git-role agents clone staging work, publish branch cma/<planName>, commit files, then open Pull Requests, or Merge Requests or fallback pushes depending on your Git configuration.

Step 9 — Inspect Provider Output

27. Expand the completed **Create Git Change** rows to view direct links to your repository.
 28. Click the outbound links to open the generated Pull Requests or Merge Requests in your Git provider interface.
- Merged repositories carry DAG artifacts on tracked base branches and Cloudera Data Engineering workspaces continue governing runtime workloads.

What to do next

For agent connectivity issues, duplication errors, or Git credential regressions, see *Troubleshooting Airflow DAG migration to Git*.

Related Information

[Setting Up to Git Migration \[Technical Preview\]](#)

[Troubleshooting Airflow DAG migration to Git \[Technical Preview\]](#)