Replication Manager Overview

Date published: 2019-11-15
Date modified: 2024-07-08

CLOUDERA

Legal Notice

Contents

Replication Manager in Cloudera on cloud

Replication Manager is a service in Cloudera on cloud. You can create replication policies in Replication Manager to copy and migrate data from CDH (version 5.13 and higher) clusters (HDFS, Hive, and HBase data) and Cloudera Base on premises (version 7.1.4 and higher) clusters (HDFS, Hive external tables, and HBase data) to Cloudera on cloud clusters. You can also replicate HDFS data from cloud storage to classic clusters (CDH or Cloudera Base on premises clusters), and Hive external tables to Data Hubs. The supported Cloudera on cloud services include Amazon S3 and Microsoft Azure ADLS Gen2 (ABFS). Replicating Hive managed tables using Replication Manager from HDP clusters to Cloudera on cloud is a beta feature and is not available for general use.

Replication Manager can replicate between services where at least the source service or the destination service is a local service. The term *local service* refers to a service of a cluster managed by Cloudera Manager. In the on-premises to on-premises replication scenarios, you create the replication policy in the target Cloudera Manager. It is not possible to create replication policies between two non-local services, services on separate peer clusters, or between cloud and a peer cluster's service. In these unsupported scenarios, you can create replication policies on the Cloudera Manager that has the service that you are replicating to/from.

Before you create replication policies, you must ensure that the clusters are supported by Replication Manager. For more information, see Support matrix for Replication Manager on Cloudera on cloud. Ensure that the target cluster's and source cluster's Cloudera on cloud and Cloudera Manager versions are the same, or the target cluster's version is higher than the source cluster's version.

You can access the Replication Manager service on the Cloudera on cloud web interface. To replicate data between clusters, add the source on-premises clusters as classic clusters on the Cloudera Management Console Clusters page, add/create one or more Cloudera on cloud SDX Data Lakes and/or Data Hubs, and then create the replication policies in Replication Manager. The **Replication Policies** page shows the progress and status of replication policy jobs. You can also use CDP CLI to create HDFS and Hive replication policies.

Replication Manager provides the following functionalities that you can use to accomplish your data replication goals:

HDFS replication policies

These policies replicate HDFS data and metadata from on-premises clusters (CDH, Cloudera Base on premises, and HDP) to cloud storage buckets such as S3 and ABFS, and from cloud storage to classic clusters (CDH or Cloudera Base on premises clusters). You can choose the frequency of replicating data.

Some use cases where you can use HDFS replication policies include:

• Moving legacy data (from CDH clusters) to cloud deployments (AWS or Azure on Cloudera on cloud).
• Archiving cold data.
• Replicating the required data to another cluster to run analytics on it.

Hive replication policies

These policies support table-level replication and can replicate Hive external tables from on-premises clusters (CDH and Cloudera Base on premises) to cloud storage such as S3 and ABFS and to Data Hubs. They also can:

• replicate data stored in Hive tables, Hive metadata, data in Hive metastore, and Impala metadata (catalog server metadata) associated with Impala tables registered in the Hive metastore, and

> **Note:** Hive2 managed tables are converted to external tables after replication.

• migrate Sentry permissions to Ranger.

> **Note:** To perform the Sentry policy replication, you must be running the Sentry service on CDH 5.12 or higher, or any CDH 6.x version.

Some use cases where you can use Hive replication policies include:

- Backing up data periodically.
- Performing a recovery operation when necessary.
- Creating a development and test system for engineers to run quality checks.

HBase replication policies

You can create these policies to replicate HBase data from a source classic cluster (CDH or Cloudera Base on premises cluster), COD, or Data Hub to a target Data Hub or COD cluster. You can also copy or replicate HBase data between different environments within a Virtual Private Cloud (VPC) using these policies. Any future data change in the source cluster is pushed to the target cluster automatically without user intervention.

Some use cases where you can use HBase replication policies include:

- Performing an active-active disaster recovery with conflict resolution (enabling other disaster recovery use cases which provides an efficient utilization of resources).
- Copying required data to the cloud clusters for heavy-duty analytics workloads which helps to optimize on-premises cluster performance.
- Utilizing the continuous data synchronize feature to implement a hybrid cloud that in turn helps you to use it in various other use cases.

CDP CLI for HDFS and Hive replication policies

You can also use CDP CLI commands to create HDFS and Hive replication policies. The CDP CLI commands for Replication Manager are under the replicationmanager CDP CLI option. For more information, see *CDP CLI for Replication Manager*.

CDP CLI is a unified tool to manage all the Cloudera on cloud services. This gives you the flexibility to use it across services from a single pane, view required information in a single scroll (for example, you can view all available clusters' service status in a single page), and collaborate to troubleshoot issues.

Related Information
Support matrix for Cloudera Replication Manager
HBase replication policy
CDP CLI for Cloudera Replication Manager

Replication Manager terminology

Replication Manager is a service that can be accessed through the Cloudera on cloud web interface in Cloudera Data Platform. You can create replication policies in Replication Manager. You can also use CDP CLI commands to create replication policies.

Term	Description
Replication Manager Service	The web UI that runs on the Cloudera Data Platform host.
Data center	The facility that contains the computer, server, and storage systems and associated infrastructure, such as routers and switches. Corporate data is stored, managed, and distributed from the data center. In an on-premises environment, a data center is often composed of a CDH cluster or Cloudera Base on premises cluster. However, a single data center can contain multiple on-premises clusters.
Cloud data lake or data lake	A Cloudera on cloud cluster on the cloud, using virtual machines, with data retained on cloud storage. A cloud data lake requires minimal services for metadata and governance, such as Hive metastore, Ranger, and Atlas.

Term	Description
Cloud storage	A storage retained in a cloud account, such as Amazon S3 web service or Microsoft Azure.
On-premises cluster	A CDH cluster in a data center or a Cloudera Base on premises cluster, with Apache services running, such as HDFS, Yarn, HMS, Hiveserver2, Ranger, and Atlas. Replication behavior is similar to IaaS cluster replication. The data is on local HDFS.
Replication policy	A set of rules applied to a replication relationship. The rules include which clusters serve as source and destination, the type of data to replicate, the schedule for replicating data, and so on.
Job	An instance of a replication policy that is running or is completed
Source cluster	The cluster that contains the source data that is replicated to a destination cluster. Source data could be an HDFS dataset, Hive database, or HBase tables.
Target cluster	The cluster to which the data is replicated.

Related Information

How replication policies work

HDFS replication policy

Hive replication policy

HBase replication policy

Support matrix for Cloudera Replication Manager

Fine-grained permission to access Cloudera Replication Manager

You can restrict access to specific users to view and use Cloudera Replication Manager in a Cloudera on cloud environment so that you can govern the access to critical replication functionalities.

Currently, any user in a Cloudera on cloud environment can view and use Cloudera Replication Manager to create, run, and manage replication policies. However, in some deployments, it is essential that only a few authorized users have access to Replication Manager. This requirement arises when you want to provide an added layer of control which aligns with the best practices for data management and security, and also to enhance security and control over replication management which includes monitoring the replication jobs, and troubleshooting issues efficiently.

Providing role-based access control (RBAC) to Replication Manager users

You can provide fine-grained permissions to specific users to view and use Cloudera Replication Manager

Procedure

1. Enable the RBAC entitlement. Contact your Cloudera account team to accomplish this task.
2. Identify the users that require access to Cloudera Replication Manager.
3. Go to the Cloudera Management Console User Management page. You can manage the role assignments on this page.

   ⚠️ **Important:** Administrators with the PowerUser role can add, modify, or delete users and groups in the Cloudera on cloud environment.

4. Assign the ReplicationAdmin role to one or more users, and Save the changes.

> **Note:** You must have the ReplicationAdmin or PowerUser role to use Replication Manager if the entitlement is enabled.

5. Optionally, create a group to manage the Replication Manager users and their roles. For example, replicationusers.

Access Replication Manager in Cloudera on cloud

You can access the Replication Manager service by logging into Cloudera Data Platform.

When you log into Cloudera Data Platform, the Cloudera on cloud web interface appears. Click Replication Manager to view the **Overview** page of the Replication Manager.

Replication Manager has the following pages:

- Overview
- Classic Clusters
- Cloud Credentials
- Replication Policies

The following image shows the Cloudera on cloud web interface:

You can also access the Replication Manager service by logging into the Management Console. In the Management Console, click [icon] and select Replication Manager.

Overview

When you click Replication Manager on the Cloudera on cloud web interface, the Overview page appears. The page provides a snapshot of the Replication Manager service. It provides insights into issues and updates related to various entities and resources through dashboards like Classic Clusters, Replication Policies, Notifications, and so on.

The following panels appear on the Overview page:

- Classic Clusters
- Policies
- Jobs
- Issues & Updates

Click Create Policy to create a replication policy.
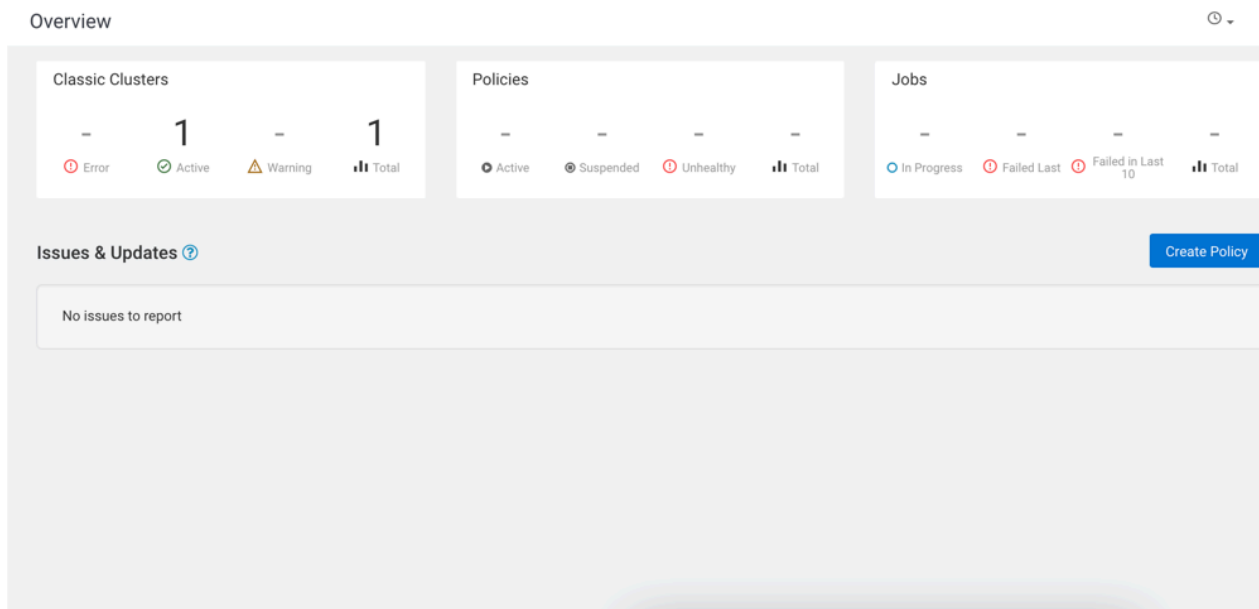
Classic Clusters

The Classic Clusters panel on the Overview page tracks the total number of clusters enabled for Replication Manager, the number of clusters that are in an error state, the number of clusters that are active, and the number of clusters for which a warning is issued.

You must register your existing on-premises Cloudera Distribution of Hadoop (CDH) and Cloudera Base on premises clusters on the Cloudera Management Console, after which you can copy or move your data to the cloud. In Cloudera on cloud, these clusters are called *classic clusters*.

The Classic Clusters panel shows the following cluster status:

- **Active** clusters that are currently available to run the replication jobs.
- Clusters having less than 10% disk capacity appears as **Warning**. Click the number to open a table to track the cluster name and the exact disk capacity remaining.
- **Total** number of clusters that are in use.
- Number of clusters that are currently not running as expected or in **Error** state.

To investigate the issues associated with clusters that have an error or warning status, use Cloudera Manager for on-premises clusters.

Policies

The Policies panel on the Overview page tracks the number of replication policies that are in use and their status.

The **Policies** panel shows the following status for the replication policies:

- **Active** replication policies in Submitted state or Running state. This item is not actionable.
- Replication policies that are **Suspended** by the administrator. This item is not actionable.
- **Unhealthy** replication policies that are associated with a cluster designated as Error on the **Classic Clusters** panel. Click the number to find out the policy names, the names of its associated source clusters and destination clusters, and the services that have stopped on the source cluster or destination cluster.

- **Total** number of running policies.

When you click **Policies** on the **Overview** page, the **Replication Policies** page appears. On this page, you can view the replication policy details.

Jobs

The Jobs panel on the Overview page tracks the total number of running and failed jobs and their status in Replication Manager.

The **Jobs** panel shows the following details:

- **In Progress** jobs or jobs in running state.
- **Failed Last** replication policies are the policies for which the last job failed to complete.
- **Failed in Last 10** replication policies are the policies for which at least one of the last ten jobs failed.
- **Total** number of available jobs.

You can click the number in the panel to apply a filter to the **Issues & Updates** table to view the required policies.
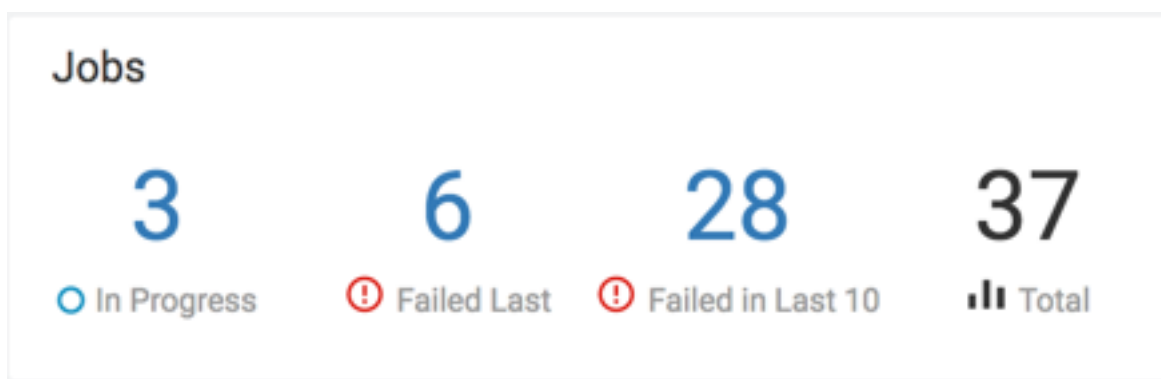
Issues & Updates

The Issues & Updates panel lists the replication policies that have running jobs with at least one job in Failed status in the most recent ten jobs. If you do not see any policy, it indicates that the last ten jobs of all the replication policies were successful.

The **Issues & Updates** panel has the following columns:

- Current **Job Status** of the job. When the job status is running, the status circle icon and a progress bar appear. For jobs that are not running, the status circle icon appears with Success, Failed, or Ignored text. Hover over the Failed status and click View Log to view the job log.
- **Source** cluster associated with the replication policy.
- **Destination** cluster associated with the policy.
- **Service** indicates whether the data being replicated is HDFS, Hive, or HBase.
- Replication **Policy** name.
- Policy History of the last ten job statuses. The column also shows the status as colored dots which you can click to view the policy details on the **Policies** page:

  - Green indicates that the job completed successfully.
  - Red indicates that the job did not complete.
  - Gray indicates that the job did not start because a previous instance of the policy is still in progress. Only one run of a job can be in progress at one time. If a job is consistently ignored, edit the replication policy to modify its frequency.

  When you click the colored dots, the page appears with the filter preset to show the information about the specified policy.

- **Transferred/Files** is the amount of data transferred, in gigabytes, and the number of objects transferred, if available. When a job is running, the column shows In Progress.

- **Runtime** or time taken to complete the most recent job.
- Most recent job that **Started**.
- Most recent job that **Ended**.

The **Actions** menu shows the following options:

- **Abort Job** aborts a running job.
- **Re-run Job** starts another instance of the policy. This option is not available for running jobs.
- **Edit Policy** settings. This option is not available for expired policies.
- **Delete Policy** removes the replication policy permanently. The delete operation cannot be undone.
- **Suspend Policy** only if the job is running.
- **Activate Policy** resumes a suspended replication policy.

	Job Status	Source		Destination	Service	Policy	Policy History	Runtime	Started	Ended	
	Failed	mycluster0	→		Hive	hive-repl		00h32m	2d ago	2d ago	⋮
	Failed	mycluster0	→		Hive	bucket_bootstrap4_h...		01h24m	a day ago	a day ago	⋮
	Skipped	mycluster0	→		Hive	bucket_bootstrap3_h...		<1m	seconds ago	seconds ago	⋮
	Skipped	mycluster0	→		Hive	bootstrap_with_drop...		<1m	a minute ago	a minute ago	⋮
	Failed_admin	mycluster0	→		Hive	partition_bootstrap1_...		00h01m	2d ago	2d ago	⋮
	Skipped	mycluster0	→	beacon-idbroker	Hive	atlas_timeout		<1m	3d ago	3d ago	⋮
	Failed	mycluster0	→	beacon-idbroker	Hive	testPKS4		00h34m	3d ago	3d ago	⋮
	Skipped	mycluster0	→		Hive	view_bootstrap_1565...		<1m	seconds ago	seconds ago	⋮
	In Progress	mycluster0	→	beacon-idbroker	Hive	hivedemo		Running	30m ago		⋮
	Skipped	mycluster0	→		Hive	basic_bootstrap1565...		<1m	2m ago	2m ago	⋮

1 - 10 of 15   ← 1 2 ··· →   10 results/page ▾

Classic Clusters page

The Classic Clusters page specifies the total number of clusters enabled for Replication Manager, the number of clusters that are in an error state, the number of clusters that are active, and the number of clusters for which a warning is issued.

The **Classic Clusters** page shows the cluster health status, cluster name, cluster version, number of nodes in the cluster, number of replication policies in the cluster, and the location of the cluster. Use the Actions menu to create a replication policy for the cluster, launch Cloudera Manager, or sync the cluster configuration. Click Add to create a replication policy.

The Classic Clusters map panel shows the geolocation of each cluster and helps you to easily identify the status of cluster services, using the following interactive markers on the map:

- Red indicates that at least one required service has stopped on the cluster.
- Orange indicates that all the required services are running on the cluster but the remaining disk capacity on the cluster is less than 10%.
- Green indicates that all the required services are running on the cluster and the remaining disk capacity is greater than 10%.

Hover over a marker on the map to view the data center associated with the cluster, the cluster name, and the number of Replication Manager policies that are associated with that cluster.

To investigate the issues associated with clusters that have an error or warning status, launch Cloudera Manager.

Classic Clusters

Status			Versions	Nodes	Policies	Location	
● HEALTHY	dc1	/	Cluster 1	CDH-6.3.9999	—	0	Delhi, India	⋮
● HEALTHY		/	Cluster 1	CDH-6.9999.1	3	4	Buenaventura, Colombia	⋮
● HEALTHY	us-west-2	/	Cluster 1	CDH-6.9999.1	—	0	San Francisco, United States of America	⋮
● HEALTHY	dd2	/	Cluster 2	CDH-6.3.0	3	0	Manacapuru, Brazil	⋮
● HEALTHY	budapest	/	Cluster 1	CDH-6.9999.1	4	1	Budapest, Hungary	⋮
● HEALTHY		/	Cluster 1	CDH-7.0.0	3	7	Buenos Aires, Argentina	⋮
● HEALTHY	dd	/	Cluster 1		—	0	Manacapuru, Brazil	⋮
● HEALTHY	sudipto	/	mycluster0	HDP-2.6.5	5	3	Sydney, Australia	⋮

Cloud Credentials page

The Cloud Credentials page shows the registered cloud credentials for Replication Manager. To replicate data to or from a storage cloud account, you must register the cloud credentials, so that the Replication Manager can access your cloud account. The supported cloud storage accounts are Amazon S3 and Azure Blob Filesystem (ABFS). On the Cloud Credentials page, you can add cloud credentials. You can also update or delete the credentials when necessary.

When you add cloud credentials for your Amazon S3 account, you can choose one of the following authentication methods:

- Access secret key. To use this authentication type, you require an AWS Access Key and an AWS Secret key that you obtain from Amazon. Cloudera Manager stores these values securely and does not store them in world-readable locations. The credentials are masked and encrypted in the configurations passed to processes managed by Cloudera Manager, and redacted from the logs.
- IAM role. Amazon Identity and Access Management (IAM) can be used to create users, groups, and roles for use with Amazon Web Services, such as EC2 and Amazon S3. IAM role-based access provides the same level of access to all clients that use the role.

⚠️ **Important:** You can choose the IAM role authentication type only when the following conditions are met:

- The source cluster is hosted on an AWS EC2 infrastructure.
- The source cluster Cloudera Manager and all the nodes in the cluster are running on an EC2 instance.
- The source cluster Cloudera Manager has the same IAM role.

For information about configuring AWS credentials, see Introduction to role based provisioning credential in AWS.

You can perform the following tasks on the Cloud Credentials page to manage cloud credentials:

Add cloud credentials

You can add cloud credentials for your S3 or ABFS account. For information about adding cloud credentials, see Working with Cloud Credentials.

📝 **Note:** Unregistered credentials can impact the replication process. Credentials associated with a cluster node that do not have updated credentials are called unregistered credentials. For example, if a node is down when the credentials are changed on a bucket or when the node is brought up that has the old credentials.

Update cloud credentials

You can update the cloud credentials based on various factors. When the bucket configuration such as secret or access keys, bucket name or endpoint, and encryption type is changed, it can affect the Replication Manager replication policy run and might require an update to the Replication Manager cloud credentials.

Credential changes are picked up by the next run of the policy. When you change the credentials, the in-progress policy runs might fail but the succeeding runs pick up the changes.

To update a cloud credential, click Actions Update .

Delete cloud credentials

You can delete unwanted credentials from the Replication Manager. When you delete cloud credentials, the
replication policies that use the deleted cloud credentials might fail. To avoid failures, delete the Replication Manager
cloud policies associated with the deleted credentials and recreate the policies with the new credentials. You can view
a list of policies associated with specific credentials on the **Cloud Credentials** page.

To delete a cloud credential, click Actions Delete .

Replication Policies page

The "Replication Policies" page shows the number of replication policies that are active, the number of policies
that have been suspended, the number of policies that are in error state, and the total number of replication policies
available in Replication Manager. The page also provides a detailed view about the replication policies.

The Replication Policies page shows a replication policy status dashboard which shows the following panels and the
number of policies with the status:

*   **Error** shows the number of replication policies associated with a cluster designated as Error on the Classic
    Clusters map. Click the number to understand the policy names, the names of the source and destination clusters,
    and which services are stopped on the source or destination cluster.
*   Active replication policies that are in Submitted or Running state. This item is not actionable.
*   **Suspended** replication policies that have been suspended by the administrator. This item is not actionable.
*   **Total** number of running policies.

Click a replication policy to view more details about the policy. Click Actions (   ) to perform more actions on a
replication policy.

Additionally, you can perform the following tasks on the **Replication Policies** page:

*   Change the timezone, if required, using   .

*   View the list of unreachable clusters using   .

Replication policy details

You can also view the following policy details on the Replication Policies page:

*   Current policy Status .
*   Policy Type shows HDFS, Hive, or HBase.
*   Replication policy Name .
*   Source cluster name.
*   Destination cluster name.
*   Jobs that were run for the replication policy and its current status.
*   Duration or time taken to run the policy.
*   Last Success timestamp of the last successful run.
*   Next Run timestamp of the next scheduled run.

Optimize Replication Policies page performance

By default, the replication policies are loaded only partially on the **Replication Policies** page, therefore the page might display incomplete statistics about a job status. This is because the job history is necessary to decide whether a policy failed or succeeded. The replication policies with failed jobs might take a longer time to load.

You can change the page load behavior depending on your requirements using [clock icon]. Choose one of the following options to load the **Replication Policies** page faster by delaying to load the job history:

• Delay loading job history when it takes too long attempts to load the job history, but omits the load operation above a certain threshold. By default, Replication Manager uses this option.
• Never load job history minimizes the load on Cloudera Manager and maximizes Replication Manager performance.
• Always load job history ensures that the job history is always loaded for all the displayed replication policies.

Use Case

Sometimes, Replication Manager fails to reach a healthy Cloudera Manager when there is a temporary networking blip or when there is a load spike on Cloudera Manager. When a cluster becomes unreachable for Replication Manager, the cluster is placed in the list of unreachable

clusters (the list appears when you click [warning icon]). Replication Manager retries to reach the cluster again after 20 minutes. After you confirm that the Cloudera Manager is healthy and expect it to be reachable by Replication Manager, you can force reload the **Replication Policies** page using

[reload icon] to reconnect every cluster.

For more information, see Replication Policies page does not display all the replication policies.

How replication policies work

In Replication Manager, you create replication policies to establish the rules you want applied to your replication jobs. The policy rules you set can include which cluster is the source and which is the destination, what data is replicated, what day and time the replication job occurs, the frequency of job runs, and bandwidth restrictions.

The first time you run a job (an instance of a policy) with data that has not been previously replicated, Replication Manager creates a new folder or database and bootstraps the data. During a bootstrap operation, all data is replicated from the source cluster to the destination. As a result, the initial execution of a job can take a significant amount of time, depending on how much data is being replicated, network bandwidth, and so on. So you should plan the bootstrap operation accordingly.

After the bootstrap operation succeeds, an incremental data replication is automatically performed. This job synchronizes, between the source and destination clusters, any events that occurred during the bootstrap process. After the data is synchronized, the replicated data is ready for use on the destination. Data is in a consistent state only after incremental replication has captured any new changes that occurred during bootstrap.

Subsequent replication jobs from the same source location to the same target on the destination are incremental, so only the changed data is copied.

When a bootstrap operation is interrupted, such as due to a network failure or an unrecoverable error, the Replication Manager does not retry the job instead it runs the job at the next scheduled interval, if available. Therefore, if the bootstrap operation is interrupted, you must manually correct the issue and then run the policy.

When scheduling how often you want a replication job to run, you should consider the recovery point objective (RPO) of the data being replicated; that is, what is the acceptable lag time between the active site and the replicated data on the destination.

Related Information

HDFS replication policy

Hive replication policy

HBase replication policy

Replication policy considerations

You should take into consideration certain guidelines when creating or modifying a replication policy. It is important for you to understand the security restrictions and encryption policies within Replication Manager.

The guidelines you need to consider before you create or modify a replication policy includes:

Data security

To use an S3 or ABFS cluster for your policy, register your credentials on the Cloud Credentials page.

A user with access to the Replication Manager user interface has the ability to browse, within the Replication Manager UI, the folder structure of any clusters enabled for Replication Manager.

Therefore, users can view folders, files, and databases in the Replication Manager user interface, that they might not have access to in HDFS. Users cannot view from the Replication Manager UI the content of files on the source or destination clusters. Nor do these administrators have the ability to modify or delete folders or files that are viewable from the Replication Manager UI.

Policy properties and settings

Consider the recovery point objective (RPO) of the data being replicated when you schedule a replication policy. The RPO is the acceptable lag time between the active site and the replicated data on the destination. Ensure that the frequency is set so that a job finishes before the next job starts.

Jobs based on the same policy cannot overlap. If a job is not completed before another job starts, the second job does not execute and is given the status Skipped. If a job is consistently skipped, you might need to modify the frequency of the job.

Specify bandwidth per map, in MBps. Each map is restricted to consume only the specified bandwidth. This is not always exact. The map throttles back its bandwidth consumption during a copy in such a way that the net bandwidth used tends towards the specified value.

Cluster requirements

- Pair the clusters before you include them in a replication policy.
- With a single cluster, you can replicate data on-premises to cloud.
- With a single cluster, you cannot replicate data on-premises to on-premises.
- If the clusters are Replication Manager-enabled, it appears in the Source Cluster or Destination or Data Lake Cluster fields in the Create Policy wizard. You must ensure that the clusters you select are healthy before you start a policy instance (job).

Hive restrictions

- When creating a schedule for a Hive replication policy, you should set the frequency so that changes are replicated often enough to avoid overly large copies.
- ACID tables, managed tables, storage handler-based tables such as Apache HBase, and column statistics are not replicated. Hive2 managed tables are converted to external tables after replication.

How temporary AWS credentials for replication policies works

Some deployments require temporary AWS session credentials to provide just-in-time, minimum required access to replicate data using replication policies. You can achieve this task using IDBroker. You can use temporary AWS credentials, through the IDBroker service, to replicate HDFS data, Hive external tables, and HBase data from Kerberized Cloudera Base on premises 7.1.9 SP1 clusters or higher using Cloudera Manager 7.11.3 CHF7 or higher versions to S3 buckets using Cloudera Replication Manager.

You can also use the temporary AWS credentials to replicate the HDFS data from S3 buckets to Kerberized Cloudera Base on premises 7.1.9 SP1 clusters or higher using Cloudera Manager 7.11.3 CHF7 or higher versions.

IDBroker is a REST API built as an extension of Apache Knox's authentication services. It allows an authenticated and authorized user to exchange a set of credentials or a token for short-lived cloud vendor access tokens.

To acquire the temporary AWS credentials, you create an IDBroker topology and then map the Kerberos users (or groups) to an AWS IAM Role. During the replication policy run, Replication Manager invokes IDBroker, and the IDBroker then uses the mapping between the on-premises Kerberized user and the IAM Role to request an AWS session token for that role.

Use case

> An organization uses the same on-premises cluster across all their departments, and each department has its own AWS account so that it can replicate its required data from the on-premises cluster to its own AWS account when necessary. Each department depending on their requirements might either want to leverage the cloud storage capabilities to store data, or use the cloud processing capabilities to run workloads, analyze the data, or any other purposes.

Authentication methods to use AWS credentials in replication policies

You can choose long-term AWS cloud credentials or temporary AWS session credentials when you want to replicate HDFS data, Hive external tables, and HBase data from Cloudera Base on premises clusters to S3 buckets on Cloudera on cloud.

Long-term cloud credentials

You can use long-term credentials to replicate data to the cloud using replication policies. To use long-term cloud credentials in a replication policy, you must:

• have an AWS account, and access key and secret key for it.
• register an external account in Cloudera Manager using AWS access key and AWS secret key.

  You can add an external account on the Cloudera Manager Administration External Accounts page. The external account serves as an authentication method during data replication, using replication policies, from Cloudera Base on premises clusters to cloud.
• add the cloud credential in Cloudera on cloud Replication Manager.

The following use cases illustrate scenarios where you can use long-term AWS credentials:

• Environments where you have multiple users and multi-tenancy – In this instance, you can add an **Add Access Key Credentials** external account in Cloudera Manager for Cloudera Base on premises cluster, add the cloud credentials in the Cloudera Replication Manager, and then create a replication policy.
• Single user cluster, or where all the users of the cluster have the same privileges to the data in Amazon S3 – In this instance, you can add **IAM role-based authentication** in Cloudera Manager for Cloudera Base on premises cluster, add the cloud credentials in the Cloudera Replication Manager, and then create a replication policy.

Temporary AWS session credentials

You can use temporary AWS session credentials to provide just-in-time, minimum required access to replicate data using replication policies. Before you use temporary AWS session credentials in a replication policy, you must:

1. have an AWS account with an IAM role that has the required permissions to access the target S3 bucket and has the necessary trust relationships set up.
2. install and configure IDBroker on the Cloudera Base on premises cluster.
3. add the cloud credential in Cloudera Replication Manager.

   Alternatively, you can add an external account for the IDBroker topology in Cloudera Manager.

HDFS replication policy

You can use the HDFS replication policies in Cloudera Replication Manager to replicate HDFS data. The HDFS replication policies can replicate HDFS data and metadata from classic clusters (CDH, Cloudera Base on premises, and HDP) to Cloudera on cloud storage buckets such as S3 and ABFS, and from cloud storage to classic clusters (CDH or Cloudera Base on premises clusters). To use an on-premises cluster (CDH or Cloudera Base on premises cluster) in the replication policy, you must register it as a classic cluster in the Cloudera Management Console. To use the cloud storage for data replication, you must register the cloud credentials in Replication Manager so that the Replication Manager service can access the cloud storage. You must also verify cluster access and configure minimum ports for replication before you create HDFS replication policies.

⚠️ **Important:** Before you create replication policies, see *Support matrix for Replication Manager on Cloudera on cloud* to verify whether your clusters are supported by Replication Manager.

You can also use CDP CLI commands to create HDFS replication policies. The CDP CLI commands for Replication Manager are under the replicationmanager CDP CLI option. For more information, see *CDP CLI for Replication Manager*.

Related Information
Support matrix for Cloudera Replication Manager
Using HDFS replication policies
CDP CLI for Cloudera Replication Manager

HDFS snapshots

You can schedule taking HDFS snapshots for replication in the Replication Manager. HDFS snapshots are read-only point-in-time copies of the filesystem. You can enable snapshots on the entire filesystem, or on a subtree of the filesystem. In Replication Manager, you take snapshots at a dataset level. Understanding how snapshots work and some of the benefits and costs involved can help you to decide whether or not to enable snapshots.

To improve the performance and consistency of HDFS replications, enable the HDFS replication source directories for snapshots, and for Hive replications, enable the Hive warehouse directory for snapshots. For more information, see HDFS snapshots.

Enabling snapshots on a folder requires HDFS admin permissions because it impacts the NameNode. When you enable snapshots, all the subdirectories are automatically enabled for snapshots as well. So when you create a snapshot copy of a directory, all content in that directory including the subdirectories is included as part of the copy. If a directory contains snapshots but the directory is no longer snapshot-enabled, you must delete the snapshots before you enable the snapshot capability on the directory.

Take snapshots on the highest-level parent directory that is snapshot-enabled. Snapshot operations are not allowed on a directory if one of its parent directories is already snapshot-enabled (snapshottable) or if descendants already contain snapshots.

For example, in the following directory tree image, if directory-1 is snapshot-enabled but you want to replicate subdirectory-2, you cannot select only subdirectory-2 for replication. You must select directory-1 for your replication policy.
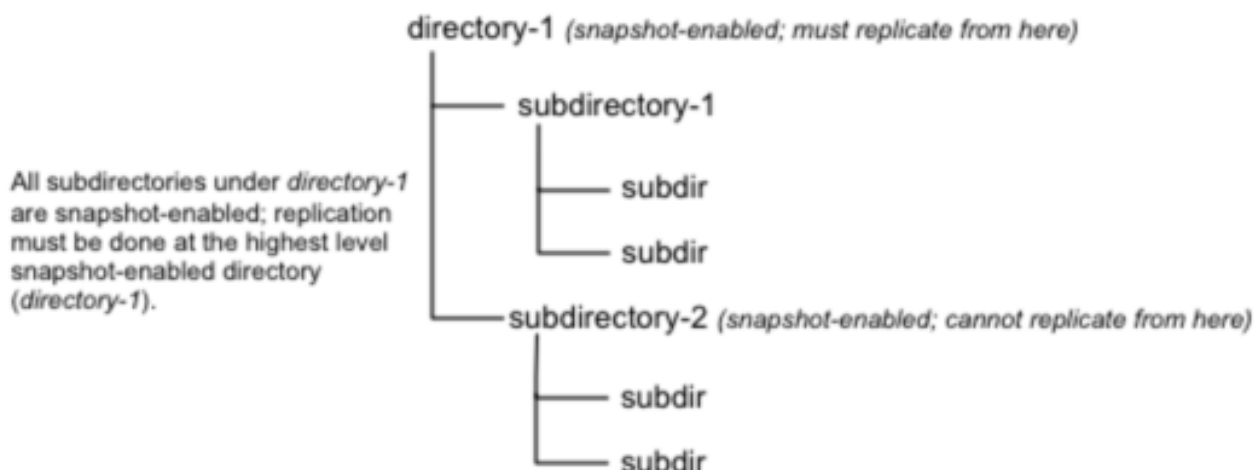
There is no limit to the number of snapshot-enabled directories you can have. A snapshot-enabled directory can accommodate 65,536 simultaneous snapshots. Blocks in datanodes are not copied during snapshot replication. The snapshot files record the block list and the file size. There is no data copying.

When snapshots are initially created, a directory named .snapshot is created on the source and destination clusters under the directory being copied. All snapshots are retained within the .snapshot directories. By default, the last snapshot of a file or directory is retained. Snapshots older than this are automatically deleted. You can configure the number of snapshots to retain when you create or edit an HDFS Snapshot Policy in the target Cloudera Manager using Cloudera Base on premises Replication Manager 7.1.1 or higher or in Backup and Disaster Recovery 5.10 or higher depending on your on-premises target cluster.

Requirements and benefits of HDFS snapshots

You might want to consider the benefits and memory cost of using snapshots. Verify the requirements before you enable snapshots.

Requirements

You must have HDFS superuser privilege to enable or disable snapshot operations. Replication using snapshots requires that the target filesystem data being replicated is identical to the source data for a given snapshot. There must be no modification to the data on the target. Otherwise, the integrity of the snapshot cannot be guaranteed on the target and replication can fail in various ways.

Benefits

Snapshot-based replication helps you to avoid unnecessary copying of renamed files and directories. If a large directory is renamed on the source side, a regular DistCp update operation sees the renamed directory as a new one and copies the entire directory.

Generating copy lists during incremental synchronization is more efficient with snapshots than using a regular DistCp update, which can take a long time to scan the whole directory and detect identical files. And because snapshots are read-only point-in-time copies between the source and destination, modification of source files during replication is not an issue, as it can be using other replication methods.

A snapshot cannot be modified. This protects the data against accidental or intentional modification, which is helpful in governance.

Memory cost

There is a memory cost to enable and maintain snapshots. Tracking the modifications that are made relative to a snapshot increases the memory footprint on the NameNode and can therefore stress NameNode memory. Because of the additional memory requirements, snapshot replication is recommended for situations where you expect to do a lot of directory renaming, if the directory tree is very large, or if you expect changes to be made to source files while replication jobs run.

Enabling and taking snapshots in Cloudera Manager

Before you take snapshots (in Cloudera Manager) or create snapshot policies (in Replication Manager) for HDFS directories, you must enable snapshots for the directories in Cloudera Manager.

Procedure

1. To enable snapshots for HDFS directories, navigate to the directory on the Cloudera Manager Clusters HDFS service File Browser tab, and click Enable Snapshots.

   Note: If you enable snapshots for a directory, you cannot enable snapshots for its parent directory. Snapshots can be taken only on directories that have snapshots enabled.

   Tip: To disable snapshots for a directory that has snapshots enabled, click Disable Snapshots. Ensure that the snapshots of the directory are deleted before you disable snapshots for the directory.

2. To take a snapshot of a directory or table, perform the following steps:
   a) Navigate to the directory or folder.
   b) Click Take Snapshot in the drop-down menu next to the full file path.
   c) Specify a unique name for the snapshot.

      The snapshot name with the timestamp of its creation appears in the snapshot list.
3. Click Actions Delete to delete a snapshot.

   Note: After you delete a snapshot, you can restore it, if required.

Hive replication policy

You can create a Hive replication policy in Cloudera Replication Manager after you configure the required Ranger policy in Ranger, register the on-premises cluster (CDH or Cloudera Base on premises) as a classic cluster in Management Console, register cloud account credentials in the Replication Manager service, verify cluster access, and configure minimum ports for replication. The replication load happens on the source on-premises cluster. You can replicate data on-premises to the cloud with a single cluster if the Metastore is running on the cloud.

These policies support table-level replication and can replicate Hive external tables from on-premises clusters (CDH and Cloudera Base on premises) to cloud storage such as S3 and ABFS and to Data Hubs. They also can:

• replicate data stored in Hive tables, Hive metadata, data in Hive metastore, and Impala metadata (catalog server metadata) associated with Impala tables registered in the Hive metastore, and

   Note: Hive2 managed tables are converted to external tables after replication.

• migrate Sentry permissions to Ranger.

   Note: To perform the Sentry policy replication, you must be running the Sentry service on CDH 5.12 or higher, or any CDH 6.x version.

Hive metadata replication involves multiple entities. Replication Manager supports replication of external tables in Hive. Hive supports replication of external tables to the target cluster and it retains all the properties of external

tables. The data files permission and ownership are preserved so that the relevant external processes can continue to write in it even after failover.

> ⚠️ **Important:** Before you create Hive replication policies, you must ensure that the required Ranger policy is set in Ranger and see Support matrix for Replication Manager on Cloudera on cloud to verify whether your clusters are supported by Replication Manager.

You can also use CDP CLI commands to create Hive replication policies. The CDP CLI commands for Replication Manager are under the replicationmanager CDP CLI option. For more information, see *CDP CLI for Replication Manager*.

The Apache Ranger access policy model consists of the following components:

- Specification of the resources that you can apply to a replication policy which includes the HDFS files and directories; Hive databases, tables, and columns; and HBase tables, column-families, and columns.
- Specification of access conditions for specific users and groups.

You must set the Ranger policy for the hdfs user on the target cluster to perform all operations on all databases and tables. The same user role is used to import Hive Metastore. The hdfs user should have access to all Hive datasets, including all operations. Otherwise, Hive import fails during the replication process.

On the target cluster, the hive user must have Ranger admin privileges. The same hive user performs the metadata import operation.

For more information about Hive replication policies to replicate data from CDH clusters to Cloudera on cloud, see Migrate Hive data from CDH to Cloudera on cloud blog.

Related Information
Support matrix for Cloudera Replication Manager
Using Hive replication policy
CDP CLI for Cloudera Replication Manager

Hive replication

Replication Manager allows you to replicate Hive databases from a source cluster to a target location on a destination cluster. The first time you run a job with data that has not been previously replicated, the Replication Manager creates a new folder or database and bootstraps the data. To replicate Hive metadata, Replication Manager performs a full replication. To replicate the data stored in Hive tables, Replication Manager uses snapshot diff-based replication to perform incremental replication.

During bootstrap operation, all of the data from the source location is copied to the destination. This bootstrapping of data can take hours to days, depending on factors such as the amount of data being copied and available network bandwidth.

After the bootstrap operation succeeds, an incremental replication is automatically performed for data replication using snapshot diff-based replication. The job synchronizes, between the source and destination clusters, any events that occurred during the bootstrap process. After the data is synchronized, the replicated data is ready for use on the destination. Data is in a consistent state only after incremental replication has captured any new changes that occurred during bootstrap.

Subsequent replication jobs from the same source location to the same target on the destination are incremental, so only the changed data is copied.

If a bootstrap operation is interrupted, such as due to a network failure or an unrecoverable error, the Replication Manager automatically retries the job. If a retry succeeds, the replication job continues from the point at which it was interrupted. If the automatic retries are not successful, you must manually correct the issue before running the policy again. When you activate the policy again, the replication job resumes from the point at which it was suspended.

Functions such as User Defined Functions (UDF) in Hive can be replicated. To enable this, you can create UDFs using a syntax. For example, the following sample code shows an UDF creation syntax:

```
CREATE FUNCTION [db_name.]function_name AS class_name  USING JAR|FILE|ARCHIV
E 'file_uri' [, JAR|FILE|ARCHIVE 'file_uri'] ;
```

Snapshot diff-based replication

By default, Replication Manager uses snapshot differences ("diff") to improve performance by comparing HDFS snapshots and only replicating the files that are changed in the source directory.

While Hive metadata requires a full replication, the data stored in Hive tables takes advantage of snapshot diff-based replication. To replicate a database using a Hive replication policy, ensure that all the HDFS paths for the tables in that database are either snapshottable or under a snapshottable root. For example, if the database that is being replicated has external tables, all the external table HDFS data locations should be snapshottable too. Failing to do so can cause the Replication Manager to fail to generate a diff report. Without a diff report, Replication Manager will not use snapshot diff.

An HDFS directory is referred to as snapshottable if an administrator - having superuser privilege or having owner access to the directory - has enabled snapshots for the directory in Cloudera Manager.

You must ensure that the following guidelines are met for efficient incremental replication:

• HDFS snapshots are immutable.

> **Tip:** In the source Cloudera Manager, go to Clusters *HDFS SERVICE* Configuration section, and search for Enable Immutable Snapshots.

• Snapshot root directory is set to as low in the hierarchy as possible.
• Replication Manager user is a super user or the owner of the snapshottable root. This is because the user specified in the Run-as-username field in the replication policy must have the permission to list the snapshots.
• Paths from both source and destination clusters in the replication policy are under a snapshottable root or are snapshottable for the replication policy to run using snapshot diff.

Replication Manager performs a complete replication when one or more of the following change: Delete Policy, Preserve Policy, Target Path, or Exclusion Path.

> **Note:** Ensure the source data does not contain an encrypted subdirectory. This is because snapshot diff-based replication might fail if an encrypted subdirectory exists in the source data.

Hive tables

Managed tables are Hive owned tables where the entire lifecycle of the tables' data are managed and controlled by Hive. External tables are tables where Hive has loose coupling with the data. Replication Manager replicates external tables successfully to a target cluster, and the Hive2 managed tables are converted to external tables after replication.

Hive supports replication of external tables with data to target cluster and it retains all the properties of external tables. The data files' permission and ownership are preserved so that the relevant external processes can continue to write in it even after failover.

The writes on external tables are performed using the Hive SQL commands and the data files can also be accessed and managed by processes outside of Hive. If an external table or partition is dropped, only the metadata associated with the table or partition is deleted but the underlying data files stay intact. A typical example for an external table is to run analytical queries on HBase or Druid owned data using Hive, where the data files are written by HBase or Druid and Hive reads them for analytics.

> **Important:** Hive Materialized Views replication is not supported. However, Replication Manager does not skip it during replication and the replicated data might not work as expected in the target cluster.

When you create a schedule for a Hive replication policy, set the frequency so that changes are replicated often enough to avoid overly large copies.

You might come across the following use cases during Hive replication:

Replication Manager upgrade use case

In a normal scenario, if you have external tables that are replicated as managed tables, after the upgrade process, you must drop those tables from the target cluster and set the base directory. In the next instance, they get replicated as external tables.

Conflicts in external tables' data location for multiple source clusters replication to the same target cluster

To handle the conflicts in external tables' data location for multiple source clusters replication to the same target cluster, the Replication Manager assigns an unique base directory for each source cluster under which the external tables' data from the corresponding source cluster is copied.

For example, if the external table location in a source cluster is /ext/hbase_data, then the location in the target cluster after replication is <base_dir>/ext/hbase_data. You can use the DESCRIBE TABLE command to track the new location of external tables.

Replication conflicts between HDFS and Hive external table location

When you run the Hive replication policy on an external table, the data is stored on the target directory at a specific location. Next, when you run the HDFS replication policy which tries to copy data at the same external table location, Replication Manager ensures that the Hive data is not overridden by HDFS.

For example, when you run a Hive replication policy on an external table, the policy creates a target directory /tmp/db1/ext1. When you run an HDFS replication policy, the policy should not override the data by replicating on the /tmp/db1/ext1 directory.

Conflicts during external tables replication process

Conflicts appear when two Hive replication policies on DB1 and DB2 (either from the same source cluster or different source clusters) have external tables that point to the same data location (for example, /abc) and are replicated to the same target cluster. To avoid such conflicts, you must set different paths for the external table base directory configuration, for both the policies.

For example, set /db1 for DB1 and /db2 for DB2. This ensures that the target external table data location is different for both databases. For example, /db1/abcd and /db2/abcd.

> **Note:** Replication conflicts are not supported for on-premises to cloud scenario.

Hive cloud replication

Replication Manager supports replication of the Hive database from a cluster with underlying HDFS to another cluster with cloud storage. It uses push-based replication, with the replication job running on the cluster with HDFS.

Hive stores its metadata in Hive Metastore, but the underlying data is stored in HDFS or cloud storage. In a Hadoop cluster with Hive service, the Hive warehouse directory can be configured with either HDFS or cloud storage.

You can perform the following tasks with Hive replication:

• Rename the dataset in the policy that is replicated.
• Create a pull-based policy on the source cluster to move data from the target back into the source cluster Hive database.

Hive replication from an HDFS-based cluster to a cloud storage-based cluster requires the following components:

• Source cluster - The cluster with a Hive warehouse directory on local HDFS. This can be an on-premises cluster or an IaaS cluster with data on local HDFS. The required services are HDFS, YARN, Hive, Ranger, and Knox.

- Destination cluster - The cluster with data on cloud storage. The cluster minimally requires Hive Metastore, Ranger, and Knox.

Replication Manager does not manage Ranger policies, and Personally Identifiable Information (PII) or any other secure data that gets replicated from on-premises to Amazon S3. You must manage these items outside of Replication Manager.

Table-level replication

To enable table-level replication, you must specify the list of tables to be replicated in a given replication policy. Table-level replication enables you to replicate just the critical tables. It also helps you to speed-up the replication process and also reduces network bandwidth utilization.

You can define table-level replication using regular expressions. You can include or exclude tables in a database for Hive replication during the Hive replication policy creation process.

The following examples illustrate how you can include or exclude Hive tables in the Hive replication policy:

- To include only table1, table2, and table3 in a database for replication, enter the database name in the Database field, and then enter table1|table2|table3 in the Tables field.
- To exclude table5, table7, and table9 and include the rest of the tables in the database, enter the database name in the Database field, and then enter (?!table5|table7|table9).+ in the Tables field.

Limitations using table-level replication

- If a table is dynamically added for replication due to changes in regular expression or added to the include list, the tables' data may not be point-in-time consistent with other tables which are already replicated incrementally. However, this inconsistency is seen for a very small duration until the completion of the next incremental replication after tables are added in the bootstrapped manner.
- Hive does not support overlapping replication policies such as db., db.[t1], and *. to the same target database but works as expected if the target databases are different.

Migrate Sentry authorization policies into Ranger

During Hive replication, Replication Manager migrates Sentry authorization policies into Ranger as part of the replication policy.

The Sentry service serves authorization metadata from the database-backed storage but does not handle actual privilege validation. The Hive and Impala services are clients of this service and it enforces Sentry privileges when the services are configured to use Sentry. Replication Manager allows administrators to migrate the existing Sentry permissions from the source CDH cluster to the Ranger policies in Cloudera on cloud.

When you create a replication policy, you can choose to migrate the Sentry policies for the resources that you want to migrate. During replication policy job run, the resources and its Sentry policies are migrated to the destination cluster. To migrate the Sentry policies for the resources, select the Include Sentry Permissions with Metadata option in the Additional Settings page of the Create Replication Policy wizard.

Note:

To perform the Sentry policy replication, you must be running the Sentry service on CDH 5.12 or higher, or any CDH 6.x version.

The Sentry Permissions section of the Create Replication Policy wizard has the following options:

- Include Sentry Permissions with Metadata migrates the Sentry permissions during the replication job run.
- Exclude Sentry Permissions with Metadata ensures that the Sentry permissions are not migrated during the replication job.
- Choose Skip URI Privileges if you do not want to include URI privileges when you migrate Sentry permissions. During migration, the URI privileges are translated to point to an equivalent location in S3. If the resources have a different location in Amazon S3, do not migrate the URI privileges because the URI privileges might not be valid.

The following image shows the Sentry Permissions section in the Create Replication Policy wizard:

The following steps are completed during the migration of Sentry policies into Ranger:

- The Export operation runs in the source cluster. During this operation, the Sentry permissions are fetched and exported to a JSON file. This file might be in a local file system or HDFS or S3, based on the configuration.
- The Translate and Ingest operations take place on the target cluster. In the translate operation, Sentry permissions are translated into a format that can be read by Ranger. The permissions are then imported into Ranger. When the permissions are imported, they are tagged with the source cluster name and the time that the ingest took place. After the import, the file containing the permissions is deleted.

> **Note:** During Hive replication from an on-premises CDH cluster to a cloud cluster, the Replication Manager migrates Sentry authorization policies into Ranger as part of the replication policy. However, no import operation is initiated if the end service in the cloud cluster (AWS) is Sentry.

A Ranger policy is created for each resource, such as a database, table, or column. The policy name is derived from the resource name. For example, if the resource is Database:dinosaurs, table= theropods, then the derived policy name is database=dinosarus->table=theropods.

The priority for migrated policies is set to normal in Ranger. The normal priority allows you to create another policy for the same resource that overrides the policy that is imported from Sentry.

Sentry to Ranger permissions

There are no one-to-one mapping between Sentry privileges and Ranger service policies, therefore the Sentry privileges are translated to their equivalents within the Ranger service policies.

The following list illustrates how the Sentry privileges appear in Ranger after the migration:

- Sentry permissions that are granted to roles are granted to groups in Ranger.

- Sentry permissions that are granted to a parent object are granted to the child object as well. The migration process preserves the permissions that are applied to child objects. For example, a permission that is applied at the database level also applies to the tables within that database.
- Sentry OWNER privileges are translated to the Ranger ALL privilege.
- Sentry OWNER WITH GRANT OPTION privileges are translated to Ranger ALL with Delegated Admin checked.
- Sentry does not differentiate between tables and views. When view permissions are migrated, they are treated as table names.
- Sentry privileges on URIs uses the object store location as the base location.

The table below shows how actions in Sentry are applied to the corresponding action in Ranger:

Table 1: Sentry Actions to Ranger Actions

Sentry Action	Ranger Action
SELECT	SELECT
INSERT	UPDATE
CREATE	CREATE
REFRESH	REFRESH
ALL	ALL
SELECT with Grant	INSERT
INSERT with Grant	INSERT
CREATE with Grant	CREATE
ALL with Grant	ALL with Delegated Admin Checked

HBase replication policy

You can replicate HBase and Phoenix tables using HBase replication policies in Cloudera Replication Manager. An HBase replication policy replicates the data at table-level granularity. After you create an HBase replication policy, you can delete one or more tables from the policy.

The replication policy replicates the data in the specified tables and continues to replicate the generated data (that is, future changes in data) unless you suspend the policy or delete the tables. You can replicate only existing HBase data, generated HBase data, or both depending on your requirements. You also can choose to replicate all the HBase tables or only the required tables in a database.

Before you create an HBase replication policy, you must:

- verify whether your clusters are supported by Replication Manager.
- understand how first-time setup configuration works.
- understand how cluster pairing works.
- understand the available methods to replicate HBase data.

You can also replicate HBase data simultaneously between multiple clusters.

Supported clusters for HBase replication policies

Before you create HBase replication policies, you must verify whether your clusters are supported by Replication Manager.

The following image shows the lowest supported versions that you can use for HBase replication policies. Consult the Support matrix for Cloudera Replication Manager for the complete list of supporting clusters and scenarios:

Figure 1: Lowest supported versions for HBase replication policies
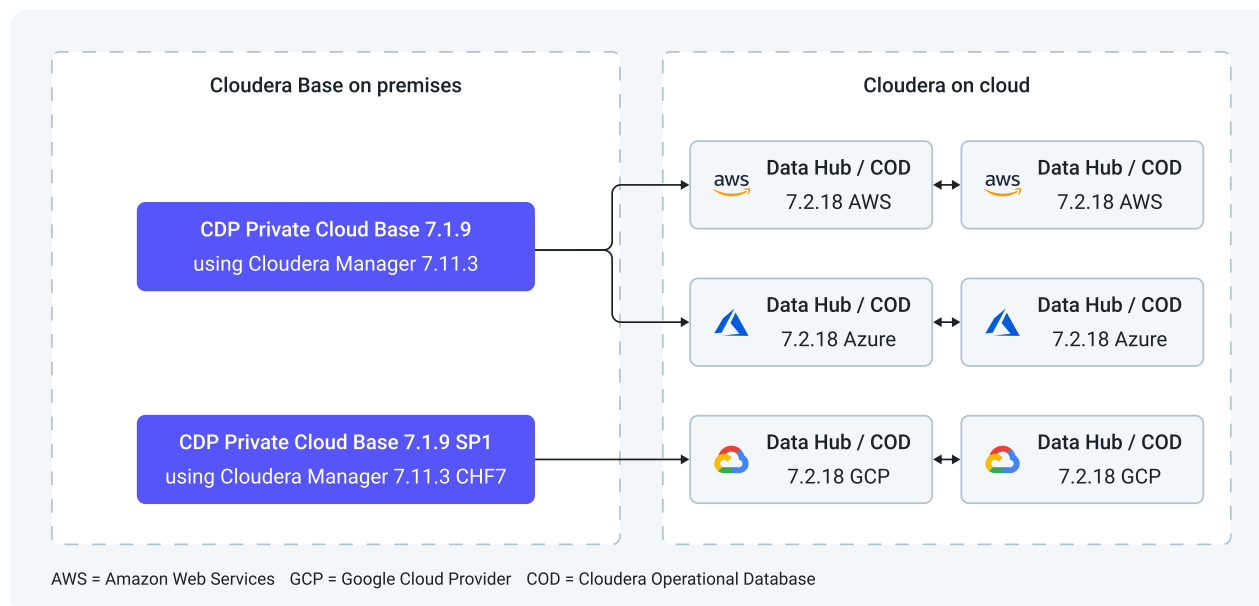
How HBase replication policies work

After you create the first HBase replication policy between a source cluster and target cluster, the Replication Manager service starts several background processes. After the processes complete, the service initiates the HBase data replication. One of the main background process is the first-time setup configuration.

*   Understanding first-time setup configuration process
*   What is a cluster pair

Steps in first-time setup configuration process

When you create the first HBase replication policy to replicate HBase data from a source cluster to a target cluster, the Replication Manager completes the first-time setup configuration steps and then replicates the data. The first-time setup configuration between a cluster pair is a one-time process, therefore subsequent HBase replication policies for the same cluster pair (ClusterA and ClusterB) do not require a first-time setup.

The first-time setup configuration completes the following steps:

1. Creates peers between the source (ClusterA) and target (ClusterB), that is, creates a cluster pair between ClusterA and ClusterB.
2. Initiates the required configuration changes. One of the steps ensures that both the clusters use the same credentials.jceks file.
3. Restarts the HBase service on both the clusters.

> ⚠️ **Important:** If you are replicating HBase data from on-premises cluster to Cloudera Operational Database (COD) or Data Hub, you must manually restart HBase service on the source cluster.

When HBase replication policies are created, modified, or deleted simultaneously, Replication Manager processes each operation independent of each other.

> 📝 **Note:** When the first-time setup between two clusters is in progress, you can create HBase replication policies between them. However, you cannot use one of these clusters with another cluster to create an HBase replication policy (in multi-cluster replication scenario) until the first-time setup is complete.

What is a cluster pair

The first-time setup configuration creates peers between the source (ClusterA) and target (ClusterB), that is, creates a cluster pair between ClusterA and ClusterB. If a cluster pair (ClusterA and ClusterB) has one or more active/ suspended HBase replication policies between them, you cannot pair either of the clusters with another cluster.

You can use ClusterA or ClusterB with another cluster in an HBase replication policy only if the following conditions are true:

- ClusterA and ClusterB do not have existing HBase replication policies.
- All existing active/suspended HBase replication policies are deleted.

A warning appears on the **Select Destination** page during HBase replication policy creation if you choose ClusterA or ClusterB as source or target (when no HBase replication policies exist between them) with another cluster or if the other cluster in the cluster pair is unreachable. To continue HBase replication policy creation, you must acknowledge the forced re-pairing of the clusters.

Related Information
Support matrix for Cloudera Replication Manager
Using HBase replication policy

Methods to replicate HBase data

You can replicate only existing HBase data, generated HBase data, or both depending on your requirements. You also can choose to replicate all the HBase tables or only the required tables in a database.

When you create an HBase replication policy, you can choose one or more of the following methods to replicate HBase data depending on your requirements:

- Replicate only the generated data from chosen tables.
- Replicate existing data and generated data from chosen tables.
- Replicate existing tables and future tables in the database.
- Replicate existing data and generated data from chosen tables and future tables

Replicate only the generated data from chosen tables

In this method, you choose one or more tables during the replication policy creation process. Replication Manager replicates only the data that is generated after policy creation.

Replicate existing data and generated data from chosen tables

In this method, you choose one or more tables, and also choose the Select Source Perform Initial Snapshot option during the HBase replication policy creation process. Replication Manager replicates the existing data and the data that is generated after policy creation.

For example, you have two tables named 'Orders' and 'Customers' in the source cluster and you want to copy the data from these tables from March 1, 2021 onwards. To accomplish this task, you create an HBase replication policy without choosing the Perform Initial Snapshot option in the **Create Replication Policy** wizard on March 1, 2021. The data that you create, update, or delete in the source cluster after you created the HBase replication policy is automatically replicated to the target cluster.

Replicate existing tables and future tables in the database

In this method, you choose the Select Source Replicate Database option during the HBase replication policy creation process. Replication Manager replicates the generated data from the existing tables, and it replicates data from the future tables that are created after the HBase replication policy creation process is complete.

To replicate data from the future tables successfully, you must create similar empty tables on the target cluster. You can perform this action when you create or add a table to the database on the source cluster.

You can choose the Replicate Database option only if the following conditions are true:

- Target Cloudera Manager version is 7.11.0 or higher.
- Source cluster's CDH version is 6.x or higher.

  CDH 5.16.2 and higher versions also support the Replicate Database option after you upgrade the source cluster Cloudera Manager.

- No existing HBase replication policies exist between the source and target clusters.

  > **Tip:** If you want to replicate the new tables that are created after the replication policy creation is complete, you must configure the replication scope to "1" for those tables on the source cluster.
  >
  > To configure the replication scope for a table on the master cluster, run the alter *[***TABLENAME***]*, {NAME => *[***COLUMN FAMILY***]*, REPLICATION_SCOPE => 1} command for each column family that must be replicated. *REPLICATION_SCOPE* is a column-family level attribute, where the value '0' means replication is disabled, and '1' means replication is enabled.

After you select the Select Source Replicate Database option in the HBase replication policy wizard, you can choose one of the following options to determine the tables in the database to replicate:

- Replicate all user tables - Replicates all the HBase tables in the database after the replication scope of the tables are set to 1.
- Replicate only tables where replication is already enabled - Replicates only those tables for which the replication scope is already set to 1.

  This option is supported only if the target cluster Cloudera version is 7.2.17.300 using Cloudera Manager 7.11.0-h3 or higher versions or Cloudera version 7.2.16.500 using Cloudera Manager 7.9.0-h7 or higher versions, or Cloudera version 7.12.0.0.

Replicate existing data and generated data from chosen tables and future tables

In this method, you choose the Perform Initial Snapshot and Replicate Database options on the **Select Source** page during the HBase replication policy creation process. You can also choose to replicate all the tables in the database or only those tables for which the replication scope is already set to 1. Replication Manager replicates the existing and generated data from the existing tables in addition to the data in future tables.

Replicate HBase data simultaneously between multiple clusters

Starting from Cloudera on cloud 7.2.16.500, 7.2.17.200, and 7.2.18, you can create multiple HBase replication policies between multiple clusters to replicate HBase data. You must consider the limitations before you create a multi-cluster replication scenario. You can use the multi-cluster replication scenario for various use cases.

- How multi-cluster HBase replication works
- Limitations
- Sample use cases

How multi-cluster HBase replication works

The first-time setup configuration consists of several steps of which one step is to ensure that the source and target cluster use the same credentials.jceks file. Therefore, if multiple supported clusters share the same credentials.jceks file, you can replicate HBase data between them seamlessly using HBase replication policies.

The following image shows a sample multi-cluster HBase data replication scenario and a few possible directions of replication:
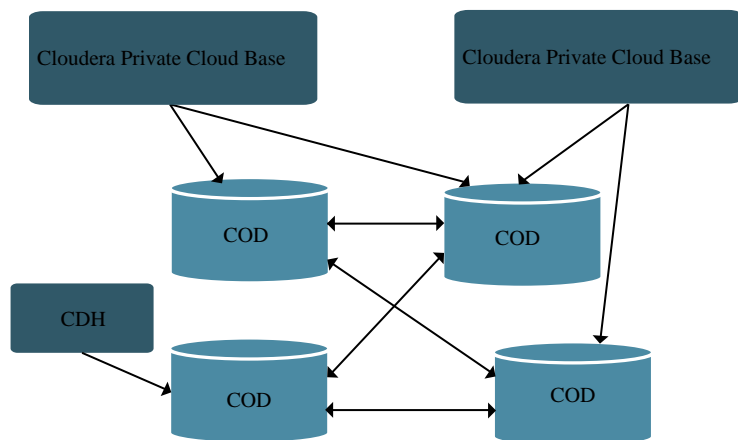
It is recommended that you do not replace the credentials.jceks file manually to create a multi-cluster HBase replication scenario. This is because when you create the first HBase replication policy between a pair of clusters, Replication Manager triggers the first-time setup process during which the credentials.jceks file in both the clusters get synchronized as required for HBase data replication.

Limitations

Consider the following limitations before you replicate the HBase data between multiple clusters using HBase replication policies:

- An HBase replication policy in a multi-cluster HBase replication setup fails when you use clusters that are part of another independent replication setup. This is because the clusters use a different credentials.jceks file. To use these clusters, you must break the cluster pairing and then create the required HBase replication policies.

    Tip: To break the cluster pairing, use the `POST /dmx/api/clusters/<target cluster crn>/hbase/resetFirstTimeSetup` API with the `{"sourceCluster": "<source cluster data center>$<source cluster name>"}` payload.

    Monitor the growing multi-cluster replication network so that it does not get disconnected. This ensures that the credentials.jceks file is the same on all clusters, the replication setup is always consistent, and no existing replication scenarios have to be reset.

- The Replication Manager UI does not allow the HBase replication policy creation to proceed if you choose a cluster (as source or target) that is in another first-time setup process. In this instance, you can wait for a few minutes to allow the first-time setup to complete and then create the HBase replication policy.

    When you create the first HBase replication policy between two clusters, the first-time setup configuration is initiated. After the configuration completes, the HBase data replication is initiated.

    Important: In a multi-cluster replication scenario, when you use a source on-premises cluster, you must manually restart the HBase service on the cluster after the first-time setup configuration completes. However, the next time you choose the same source cluster with another target cluster, the manual restart is not required. The same is true for the automatic HBase service restart in Data Hubs where the restart is performed only when the first replication is created with a particular cluster.

- The following conditions must be met to use the IDBroker credentials to create multiple HBase replication policies between multiple clusters when the target COD clusters are in separate AWS accounts or when a single AWS Role does not have access to all the required S3 buckets for all HBase target clusters:

    - Use Cloudera on cloud 7.2.18.200 or higher versions.
    - Choose the Perform Initial Snapshot option, and then specify the custom username in the Export snapshot user field in the **Select Source** page during the HBase replication policy creation process.

Use cases

Some use cases where you can use the multi-cluster HBase replication scenarios are illustrated below:

- Multiple source clusters and a single target cluster. You might have a disaster-recovery use case where you want to use a single COD to back up all the HBase data. The following image illustrates this scenario:
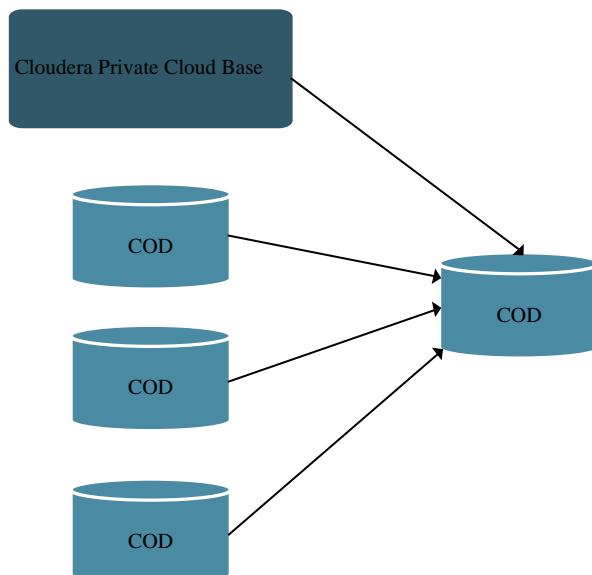
- Single source cluster and multiple target clusters. You might have a use case where all the HBase data is located in a cluster and you want to replicate only specific HBase tables to different environments to fulfill specific requirements. For example, QE environments and/or experimentation use case. The following image illustrates this scenario: