Cloudera Manager 7.11.3

Replication Manager for CDP Private Cloud Base

Date published: 2020-11-30
Date modified: 2024-07-19

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2024. All rights reserved.

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

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

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

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

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

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

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

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

Contents

Replication Manager in CDP Private Cloud Base..... 6

Support matrix for Replication Manager on CDP Private Cloud Base..... 9

Port and network requirements for Replication Manager on CDP Private Cloud Base..... 13

Prepare to replicate using replication policies.....19

Cloudera license requirements for Replication Manager.....19
Configuring SSL/TLS certificate exchange between two Cloudera Manager instances.....19
Add source cluster as peer to use in replication policies..... 21
Adding a peer to use in replication policy.....22
Modifying peers to use in replication policy..... 23
Configuring peers with SAML authentication..... 23
Enabling replication between clusters with Kerberos authentication..... 23
Required ports in Kerberos authentication-enabled clusters for replication..... 23
Considerations for realm names to use for replication..... 23
Prepare Kerberos authentication-enabled clusters for replication.....24
Kerberos connectivity test..... 25
Replicating from unsecure to secure clusters..... 25
Replication of encrypted data.....26
Encrypting data in transit between clusters..... 26
Security considerations for encrypted data during replication.....27
Configuring heap size to replicate large directories using replication policies..... 28
Retaining logs for Replication Manager..... 28

Atlas replication policies (technical preview)..... 28

Preparing to create Atlas replication policies..... 29
Creating Atlas replication policies..... 30
Manage, monitor, and troubleshoot Atlas replication policies..... 32
Error appears during Atlas replication policy run..... 33

HDFS replication policies..... 34

HDFS replication policy considerations.....34
Guidelines to add or delete source data during replication job run..... 34
Improve network latency during replication job run..... 34
Performance and scalability limitations to consider for replication policies..... 34
Guidelines to use snapshot diff-based replication.....35
HDFS replication in Sentry-enabled clusters.....36
Specifying hosts to improve HDFS replication policy performance..... 37
Creating HDFS replication policy to replicate HDFS data..... 37
How to use the post copy reconciliation script for HDFS replication policies..... 44
View HDFS replication policy details..... 46

View historical details for an HDFS replication policy.... 48
Monitoring the performance of HDFS replication policies.... 50

Hive external table replication policies.... 52
Hive replication policy considerations.... 55
Specifying hosts to improve Hive replication policy performance.... 55
Understanding how DDL commands affect Hive tables during replication.... 55
Disabling replication of parameters during Hive replication.... 56
Accommodate HMS changes for Hive replication policies.... 57
Creating a Hive external table replication policy.... 57
Sentry to Ranger replication using Hive external tables.... 64
Importing Sentry privileges into Ranger policies.... 66
Replicating data to Impala clusters.... 67
Replication of Impala and Hive User Defined Functions (UDFs).... 68
Monitoring the performance of Hive/Impala replication policies.... 68

Hive ACID table replication policies.... 71
Preparing to create Hive ACID table replication policies.... 71
Configure two-way trust between clusters.... 72
Configure parameters for Hive ACID table replication policies.... 74
Configure file access control lists for Impala user.... 77
Creating Hive ACID table replication policy.... 78
Managing Hive ACID table replication policies.... 80
Troubleshooting Hive ACID table replication policies.... 81

Iceberg replication policies.... 84
How Iceberg replication policy works.... 85
How Atlas metadata replication for Iceberg tables work.... 85
Preparing to create Iceberg replication policies.... 86
Creating Iceberg replication policy.... 86
Manage and monitor Iceberg replication policies.... 89

Ozone replication policies.... 90
Preparing clusters to replicate Ozone data.... 91
Configuring properties for OBS bucket replication using Ozone replication policies.... 93
Creating Ozone replication policies.... 95
Managing Ozone replication policies.... 98

Ranger replication policies.... 101
How Ranger replication policy works.... 101
Preparing clusters for Ranger replication policy creation.... 102
Creating Ranger replication policies.... 103
Managing Ranger replication policies.... 105

Troubleshooting replication policies between on-premises clusters.... 106

Snapshots and snapshot policies.... 109
How Replication Manager uses snapshots.... 110
HDFS snapshots.... 110

Ozone snapshots and replication methods..... 112
Creating snapshot policies in Replication Manager.....112
Manage and monitor snapshot policies.....114
Snapshots History details in Replication Manager..... 114
Troubleshooting snapshot policies in Replication Manager..... 115
Restoring HDFS snapshots in Cloudera Manager..... 116
Restoring Ozone snapshots in Cloudera Manager..... 117
Managing HDFS snapshots in Cloudera Manager.....118
Browse HDFS directories.....118
Enabling and disabling snapshots for HDFS directories..... 119
Taking and deleting HDFS snapshots.....119

Using DistCp to migrate HDFS data from HDP cluster to CDP Private Cloud Base cluster..... 119
Migrating data from secure HDP cluster to unsecure CDP Private Cloud Base cluster using DistCp.....120
Enabling the hdfs user to run the YARN jobs on the HDP cluster..... 120
Configuration changes on the CDP Private Cloud Base cluster.....121
Running the DistCp job on the HDP cluster..... 121
Migrating data from secure HDP cluster to secure CDP Private Cloud Base cluster.....121
Configuration changes on HDP cluster and CDP Private Cloud Base cluster.....122
Configuring a user to run YARN jobs on both the clusters..... 123
Running DistCp job on the CDP Private Cloud Base cluster.....124

Replication Manager in CDP Private Cloud Base

CDP Private Cloud Base Replication Manager is a service in Cloudera Manager. You can create replication policies in this service to replicate data across data centers for various use cases which include disaster recovery scenarios, running hybrid workloads, migrating data to/from cloud, or a generic backup/restore scenario. You can also create HDFS, HBase, or Ozone snapshot policies to take snapshots of HDFS directories, HBase tables, or Ozone buckets respectively.

Note:

- Replication Manager requires a valid license. To understand more about Cloudera license requirements, see Managing Licenses.
- Minimum required role - Replication Administrator or Full Administrator.
- Before you create replication policies, ensure that the source cluster and target cluster are supported by Replication Manager. For information about supported clusters and supported replication scenarios by Replication Manager, see Support matrix for Replication Manager on CDP Private Cloud Base on page 9.

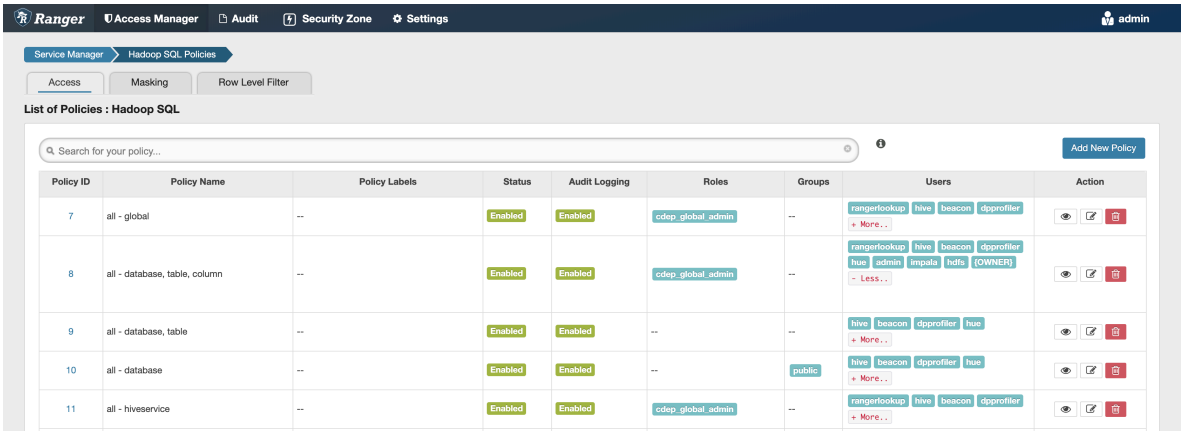
Cloudera Manager provides the following key functionalities in the Cloudera Manager Admin Console that can be leveraged by Replication Manager:

- Select datasets that are critical for your business operations.
- Monitor and track progress of your snapshots and replication jobs through a central console and easily identify issues or files that failed to be transferred.
- Issue Alert when a snapshot or replication job fails or is aborted so that the problem can be diagnosed quickly.

You can also perform a dry run of the replication policy to verify the configuration and to understand the cost of the overall operation before actually copying the entire dataset.

Important: The hdfs user must have access to all the Hive datasets, including all the operations. Otherwise, Hive import fails during the replication process. To provide access, perform the following steps:

1. Log in to Ranger Admin UI.
2. Go to the Service Manager Hadoop_SQL Policies Access section, and provide *hdfs* user permission to the all-database, table, column policy name.



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

Atlas replication policies

These replication policies replicate the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities between CDP Private Cloud Base 7.1.9 SP1 clusters using Cloudera Manager

7.11.3 CHF7 or higher. During an Atlas replication policy run, Replication Manager exports the Atlas metadata and data lineage to a staging directory in the target cluster, and then imports into the target cluster. You can enter the required staging directory during the replication policy creation process.

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

- Disaster recovery scenarios. You can back up the Atlas metadata and data lineage periodically, and restore it to the same cluster or a different cluster as required.
- High availability scenarios.
- Prevent accidental access of Ranger policies and Atlas metadata for specific Hive external tables and Iceberg tables. You can accomplish this by running both Ranger, Hive external table, and Iceberg replication policies on the required tables in the disaster-recovery cluster. The replication policies replicate the data and its associated metadata and access controls.

Note: Replicating Atlas metadata using Hive external table replication policies and Iceberg replication policies, and replicating the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities in the source cluster to the target cluster using Atlas replication policies is a technical preview feature. It is not recommended for production deployments.

Cloudera recommends that you try this feature in development or test environments. To enable this feature, contact your Cloudera account team.

HDFS replication policies

These policies replicate HDFS data and metadata from CDH (version 5.10 and higher) clusters to CDP Private Cloud Base (version 7.0.3 and higher) clusters.

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

- copying data from legacy on-premises systems to Amazon S3, Microsoft ADLS Gen2 (ABFS), and GCP, or from cloud buckets to on-premise systems.
- replicating required data to another cluster to run load-intensive workflows on it which optimizes the primary cluster performance.
- deploying a complete backup-restore solution for your enterprise.

Hive external table replication policies

These policies replicate HDFS, Hive external tables (without manual translation of Hive datasets to HDFS datasets, or vice versa), Hive metastore data, Impala metadata (catalog server metadata) associated with Impala tables registered in the Hive metastore, Impala data, and Sentry permissions to Ranger from CDH (version 5.10 and higher) clusters to CDP Private Cloud Base (version 7.0.3 and higher) clusters. In this instance, applications that depend on external table definitions stored in Hive, operate on both replica and source as the table definitions are updated.

Some use cases where you might find these replication policies useful is to:

- backup legacy data for future use or archive cold data.
- replicate or move data to cloud clusters to run analytics.
- implement a complete backup and disaster recovery solution.

Tip: You can use the Hive REPL DUMP/LOAD commands to perform a one-time data replication. However for periodic data replication between clusters, Cloudera Replication Manager is the recommended approach.

Hive ACID table replication policies

These policies replicate HDFS, Hive managed (ACID) data and metadata between CDP Private Cloud Base (version 7.1.8 and higher) clusters using Cloudera Manager version 7.7.1 or higher.

Important: To replicate managed tables (ACID) and external tables in a database successfully, you must perform the following steps in the order shown below:

1. Create Hive ACID table replication policy for the database to replicate the managed data.
2. After the replication completes, create the Hive external table replication policy to replicate the external tables in the database.

Tip: Ensure that the target database name is the same as the source database name, otherwise issues appear during or after data replication.

Some use cases where these replication policies can be used by security-conscious organizations such as financial organizations and others is to:

- replicate non-sensitive data to cloud deployments to use as a backup.
- migrate data to another cluster to run load-intensive workflows.
- use the failover functionality to make the disaster recovery cluster as your primary cluster so that the data ingestion being performed by a replication policy is uninterrupted.

Tip: You can use the Hive REPL DUMP/LOAD commands to perform a one-time data replication. However for periodic data replication between clusters, Cloudera Replication Manager is the recommended approach.

Iceberg replication policies

Iceberg replication policies replicate Iceberg tables between CDP Private Cloud Base 7.1.9 or higher clusters using Cloudera Manager 7.11.3 or higher versions.

Iceberg replication policies can:

- replicate metadata and catalog from the source cluster Hive Metastore (HMS) to target cluster HMS.

  The catalog is an HDFS file that has a list of data files and manifest files to copy from the source cluster to the target cluster. The manifest files contain the metadata for the data files.
- replicate data files in the HDFS storage system from the source cluster to the target cluster. The Iceberg replication policy can replicate only between HDFS storage systems.
- replicate all the snapshots from the source cluster by default. This allows you to run time travel queries on the target cluster.

Some use cases where you can use Iceberg replication policies are to:

- implement disaster recovery by replicating Iceberg tables between on-premises clusters.
- implement passive disaster recovery with planned failover and incremental replication at regular intervals between two similar systems. For example, between an HDFS to another HDFS system.

Ozone replication policies

You can create Ozone replication policies to replicate data in Ozone buckets between CDP Private Cloud Base 7.1.8 clusters or higher using Cloudera Manager 7.7.1 or higher.

Ozone replication policies support data replication between:

- FSO buckets in source and target clusters using ofs protocol.
- legacy buckets in source and target clusters using ofs protocol.

  Note:

  - If one or both of the source and destination buckets is a legacy bucket, then the ozone.om.enable.filesystem.paths flag (cluster-level configuration property) in the ozone-site.xml file must be enabled on the cluster(s) with the legacy bucket.
  - Ozone replication uses `ofs` by default to replicate FSO or LEGACY buckets.
- OBS buckets in source and target clusters that support S3A filesystem using the S3A scheme or replication protocol.

You can use these policies to replicate or migrate the required Ozone data to another cluster to run load-intensive workloads, back up data, or for backup-restore use cases.

Ranger replication policies

The Ranger replication policies migrate the Ranger policies and roles for HDFS, Hive, and HBase services between Kerberos-enabled CDP Private Cloud Base 7.1.9 or higher clusters using Cloudera Manager 7.11.3. It can also migrate Ranger audit logs in HDFS.

Some use cases where you can use Ranger replication policies are:

- when Ranger is used for file system-level access control for HDFS and Hive and you want to copy the Ranger policies to another cluster for backup purposes.
- when you want to move/replicate Ranger policies for Hive (SQL) or HBase data to another cluster for disaster recovery purposes.

HDFS, HBase, and Ozone snapshot policies

The HDFS, HBase, or Ozone snapshot policies take regular point-in-time snapshots of HDFS directories, HBase tables, or Ozone buckets respectively.

Snapshots act as a backup, and you can restore an HDFS directory, HBase table, or Ozone bucket to a previous version or to another location on the same HDFS, HBase, or Ozone service as necessary. Snapshots are also used by HDFS, Hive, and Ozone replication policies. The first replication policy run replicates all the data and metadata from the chosen directories. The subsequent replication policy runs leverage snapshot-diffs to replicate the changed data.

Support matrix for Replication Manager on CDP Private Cloud Base

CDP Private Cloud Base Replication Manager can replicate HDFS directories, Hive external tables, Impala data, Hive ACID tables, Iceberg tables, Ranger policies and roles for HDFS, Hive, and HBase services, and data in Ozone buckets.

See the following sections for the supported cluster and runtime versions:

- Replicate from CDH and CDP Private Cloud Base source clusters section lists the cluster and runtime versions to:
  - replicate data from CDH source clusters
  - replicate data between CDP Private Cloud Base clusters using same storage
  - replicate data between CDP Private Cloud Base clusters using different storage
- Replicate HDFS and Hive data to cloud storage
- Replicate from HDP 2 and HDP 3 source clusters

Replication policies support the following scenarios:

Kerberos

Replication Manager supports the following replication scenarios when Kerberos authentication is used on a cluster:

- Secure source to a secure destination.
- Insecure source to an insecure destination.

- Insecure source to a secure destination. The following requirements must be met for this scenario:
  - When a destination cluster has multiple source clusters, all the source clusters must either be secure or insecure. Replication Manager does not support a mix of secure and insecure source clusters.
  - The destination cluster must run Cloudera Manager 7.x or higher.
  - The source cluster must run a compatible Cloudera Manager version.
  - This replication scenario requires additional configuration. For more information, see Replicating from unsecure to secure clusters on page 25.

Transport Layer Security (TLS)

You can use TLS with Replication Manager. Additionally, Replication Manager supports replication scenarios where TLS is enabled for non-Hadoop services (Hive/Impala) and TLS is disabled Hadoop services (such as HDFS,YARN, and MapReduce).

Apache Knox

When Cloudera Manager is configured with Knox and the source and target clusters are Knox-SSO enabled, you must ensure that you use the Cloudera Manager port in the peer URL when you add the source and target clusters as peers.

Replicate from CDH and CDP Private Cloud Base source clusters

The following tables list the source and destination clusters, lowest supported versions of Cloudera Manager, and the services that are available for each supported cloud provider for CDH and CDP Private Cloud Base source clusters; ensure that the target database name is the same as the source database name, otherwise issues appear during or after data replication:

Table 1: Replicate data from CDH source clusters

Source cluster	Lowest supported source Cloudera Manager version	Lowest supported source Cloudera Runtime version	Lowest supported destination cluster version	Supported services on Replication Manager
CDH 5 CDH 6	6.3.0	5.10	CDP Private Cloud Base 7.0.3	HDFS, Sentry to Ranger*, Hive external tables
*To perform Sentry to Ranger replication using HDFS and Hive external table replication policies, you must have installed Cloudera Manager version 6.3.1 and higher on the source cluster and Cloudera Manager version 7.1.1 and higher on the target cluster.				

Table 2: Replicate data between CDP Private Cloud Base clusters using same storage

Source cluster	Lowest supported source Cloudera Manager version	Lowest supported source Cloudera Runtime version	Destination cluster	Supported services on Replication Manager
CDP Private Cloud Base	7.1.1	7.1.1	CDP Private Cloud Base	• HDFS • Hive external tables
CDP Private Cloud Base	7.7.1	7.1.8	CDP Private Cloud Base	• Hive ACID tables* • Use Cloudera Manager APIs to replicate Ozone buckets.
CDP Private Cloud Base	7.7.1 CHF4	7.1.8	CDP Private Cloud Base	Ozone buckets
CDP Private Cloud Base	7.11.3	7.1.9	CDP Private Cloud Base	• Iceberg tables • Ranger policies and roles, and Ranger audit logs in HDFS**

Source cluster	Lowest supported source Cloudera Manager version	Lowest supported source Cloudera Runtime version	Destination cluster	Supported services on Replication Manager
CDP Private Cloud Base	7.11.3 CHF7	7.1.9 SP1	CDP Private Cloud Base	Atlas replication policies***

- *You can use REPL commands or Replication Manager to replicate Hive ACID tables between CDP Private Cloud Base 7.1.8 or higher versions using Cloudera Manager versions 7.7.1 or higher.
- **You can also create Ranger replication policies on Kerberos-enabled CDP Private Cloud Base 7.1.8 or higher clusters using Cloudera Manager 7.7.1 CHF6 and higher, if the Ranger replication feature flag is enabled.
- ***Replicating Atlas metadata using Hive external table replication policies and Iceberg replication policies, and replicating the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities in the source cluster to the target cluster using Atlas replication policies is a technical preview feature. It is not recommended for production deployments. Cloudera recommends that you try this feature in development or test environments. To enable this feature, contact your Cloudera account team.

Table 3: Replicate data between CDP Private Cloud Base clusters using different storage

Source cluster	Lowest supported source Cloudera Manager version	Lowest supported source Cloudera Runtime version	Destination cluster	Supported services on Replication Manager
CDP Private Cloud Base	7.11.3 CHF1	7.1.9	CDP Private Cloud Base	Replicate the data and metadata for Hive external tables from: • source cluster using HDFS to a target cluster using Dell EMC Isilon storage. • source cluster using Dell EMC Isilon storage to a target cluster using HDFS.
CDP Private Cloud Base	7.11.3 CHF2	7.1.9	CDP Private Cloud Base	Replicate Hive ACID tables and Iceberg tables from: • source cluster using HDFS to a target cluster using Dell EMC Isilon storage. • source cluster using Dell EMC Isilon storage to a target cluster using HDFS.
CDP Private Cloud Base	7.11.3 CHF7	7.1.9 SP1	CDP Private Cloud Base	Replicate metadata-only for Ozone storage-backed Hive external tables using Hive external table replication policies. You must replicate the data using Ozone replication policies.

Important: Hive external table replication policies do not support managed to managed table replication. When you replicate from a CDH cluster to a CDP Private Cloud Base cluster, Replication Manager converts managed tables to external tables.

Therefore, to replicate managed tables (ACID) and external tables in a database successfully, you must perform the following steps in the order shown below:

1. Create Hive ACID table replication policy for the database to replicate the managed data.
2. After the replication completes, create the Hive external table replication policy to replicate the external tables in the database.

Ensure that the target cluster version is CDP Private Cloud Base 7.1.8 or higher.

Replicate HDFS and Hive data from on-premises to cloud storage

CDP Private Cloud Base Replication Manager supports the following replication scenarios:

- Replicate to and from Amazon S3 from CDH 5.14+ and Cloudera Manager version 5.13+.

  Replication Manager does not support S3 as a source or destination when S3 is configured to use SSE-KMS.
- Replicate to and from Microsoft ADLS Gen1 from CDH 5.13+ and Cloudera Manager 5.15, 5.16, 6.1+.
- Replicate to Microsoft ADLS Gen2 (ABFS) from CDH 5.13+ and Cloudera Manager 6.1+.
- Supports snapshots from CDH 5.15+ and Cloudera Manager 5.15+.
- Replicate HDFS and Hive external tables from CDP Private Cloud Base 7.11.3 CHF3 and higher clusters using Dell EMC Isilon storage to CDP Public Cloud clusters on AWS, Azure, and GCP.
- Replicate HDFS and Hive external tables from CDP Private Cloud Base 7.1.9 SP1 and higher to CDP Public Cloud clusters on GCP.

Starting in Cloudera Manager 6.1.0, Replication Manager ignores Hive tables backed by Kudu during replication. The change does not affect functionality since Replication Manager does not support tables backed by Kudu. This change was made to guard against data loss due to how the Hive Metastore, Impala, and Kudu interact.

Replicate from HDP 2 and HDP 3 source clusters

Replicating to and from HDP to Cloudera Manager 7.x is not supported by Replication Manager. However, you can replicate data using other methods. The following table lists the methods and the supported data replications to CDP Private Cloud Base clusters that are supported:

Lowest supported source version	Services that require alternate replication methods
HDP 2.6.5	HDFS. Use DistCp to replicate data.
HDP 3.1.1	HDFS. Use DistCp to replicate data.
HDP 3.1.1	• HBase. Use HBase replication to replicate HBase data. • Hive external tables. For information to replicate data, contact Cloudera Support.
HDP 3.1.5	Hive ACID tables to CDP 7.1.6 and higher clusters. Use REPL commands to replicate data. **Note:** Requires HDP 3.1.5 hotfixes.

Port and network requirements for Replication Manager on CDP Private Cloud Base

Before you create replication policies in Replication Manager, ensure that the network and security requirements for the clusters are complete. You must also ensure that the required ports are open and accessible on the source hosts and CDP Private Cloud Base hosts to allow communication between the source and destination Cloudera Manager servers and the HDFS, Hive, MapReduce, and YARN hosts. Ensure that the ports on the source and target cluster are connected.

Network and security requirements

You must ensure that the networking and security requirements for CDP Private Cloud Base are complete. For example, the cluster hosts must have a working network name resolution system, a correctly formatted /etc/hosts file, and must have properly configured the forward and reverse host resolution through DNS. For more information about the networking and security requirements, see Networking and security requirements for CDP Private Cloud Base.

Services and default port

The following table shows a list of services that Replication Manager requires, their default ports, and a brief description, and then a sample snippet is provided to illustrate the mapping of ports between the source and target clusters to use them in CDP Private Cloud Base Replication Manager:

Service	Default Port	
Cloudera Manager HTTP (Web UI)	7180 **Note:** 7183 when TLS enabled	Used for Management Control (CM.*) Open on specific source and destination IP address and not on all source IP addresses to communicate to the peer (source) Cloudera Manager. After you configure the source and destination clusters, the destination Cloudera Manager connects to source Cloudera Manager on port 7180/7183 during peering. **Note:** If TLS is enable port 7180 remain open, but redirec all reques to HTTPS on port 7183.

Service	Default Port	Used for
HDFS NameNode	8020	All Primary Nodes flow by HDFS and Hive/Impala replication to communicate from destination HDFS and MapReduce hosts to source HDFS NameNode(s).
HDFS DataNode	50010 / 9866 is used for DataNode HTTP server port. **Note:** 1004 is used for DataNode HTTPS server port.	All Secondary Nodes flow by HDFS and Hive/Impala replication to communicate from destination HDFS and MapReduce hosts to source HDFS DataNode(s).

Service	Default Port	
NameNode WebHDFS	9870 **Note:** 9871 if TLS is enabled.	Used for data flow for Apache Hadoop HttpFS service to provide HTTP access to HDFS. HttpFS has a REST HTTP API supporting all HDFS filesystem operations (both read and write). For more information, see Using HttpFS.
YARN Resource Manager	8032	Used for data flow to access the YARN ResourceManager. For more information, see YARN Configuration Properties.

Service	Default Port	
Hive Metastore	9083	All Management Nodes (CM*) Used for Hive/Impala replication to query or access Hive Metastore. For more information, see Configure metastore location and HTTP mode.
Impala Catalog Server	26000	All Management Nodes (CM*) Internal data flow during Hive/Impala replication. The catalog service uses this port to communicate with the Impala daemons.
Ranger KMS	9292 **Note:** 9494 if TLS enabled	All Primary Nodes Used for data flow during replication of encrypted data. For more information, see Migrating Keys.

Service	Default Port	Used
Kerberos KDC Server and KRB5 services	88	All Used for authentication flow by Replication Manager when Kerberos authentication is enabled on the clusters. Open the port on all the hosts on the destination cluster.
*Cloudera Manager		

For information about ports required for Ozone replication policies, see Ports used by Apache Ozone.

Sample snippet to illustrate ports mapping on source and target clusters

Some ports must be open on specific hosts of source and target clusters to facilitate and optimize the performance of Replication Manager. The following sample snippet lists the ports that are required to be open on specific hosts and how to map/connect it to other hosts to use these clusters in replication policies.

```
On the target cluster:

Target_CM* :7180 --> Source_CM :7180
Target_CM :7183 --> Source_CM :7183
Target_CM :9000 --> Source_agents :9000**
Target_CM :8020 --> Source_NameNodes :8020
Target_CM :50010 --> Source_DataNodes :50010
Target_CM :1004 --> Source_DataNodes :1004
Target_CM :50070 --> Source_NameNodes :50070***
Target_CM :8032 --> Source_ResourceManager :8032
Target_NameNodes :8020 --> Source_NameNodes :8020
Target_NameNodes :50070 --> Source_NameNodes :50070
Target_NameNodes :50010 --> DR DataNodes :50010
Target_NameNodes :1004 --> DR DataNodes :1004
Target_DataNodes :50010 --> DR DataNodes :50010
Target_DataNodes :1004 --> DR DataNodes :1004
Target_ResourceManager :8032 --> Source__ResourceManager :8032
Target_DataeNodes :8020 --> Source_NameNodes :8020
Target_CM :1006 --> Source_DataNodes :1006***
Target_NameNodes :1006 --> Source_DataNodes :1006
Target_DataNodes :1006 --> Source_DataNodes :1006
Target_CM :14000 --> Source_HttpFS :14000
```

```
On the source cluster:

Source_CM :7180 --> Target_CM :7180
Source_CM :7183 --> Target_CM :7183
Source_CM :9000 --> Target_agents :9000
Source_CM :8020 --> Target_NameNodes :8020
Source_CM :50010 --> Target_DataNodes :50010
Source_CM :1004 --> Target_DataNodes :1004
Source_CM :50070 --> Target_webHDFS :50070
Source_CM :8032 --> Target_ResourceManager :8032
Source_NameNodes :8020 --> Target_NameNodes :8020
Source_NameNodes :50070 --> Target_NameNodes :50070
Source_NameNodes :50010 --> Target_DataNodes :50010
Source_NameNodes :1004 --> Target_DataNodes :1004
Source_DataNodes :50010 --> Target_DataNodes :50010
Source_DataNodes :1004 --> Target_DataNodes :1004
Source_ResourceManager :8032 --> Target_ResourceManager :8032
Source_DataeNodes :8020 --> Target_NameNodes :8020
Source_CM :1006 --> Target_DataNodes :1006
Source_NameNodes :1006 --> Target_DataNodes :1006
Source_DataNodes :1006 --> Target_DataNodes :1006
Source_CM :14000 --> Target_HttpFS :14000


*Cloudera Manager
**Cloudera Manager agent uses port 9000
***WebHDFS NameNode uses port 50070 and WebHDFS DataNode uses port 1006
```

Prepare to replicate using replication policies

Before you use Replication Manager, you must understand some of the requirements about data replication and configure the parameters as necessary.

Cloudera license requirements for Replication Manager

You must have the necessary licenses to perform your tasks in Replication Manager.

For more information about Cloudera license requirements, see Managing Licenses.

Configuring SSL/TLS certificate exchange between two Cloudera Manager instances

You must manually set up an SSL/TLS certificate exchange between two Cloudera Manager instances that manage source and target cluster respectively. Replication Manager uses this information to set up the peers for secure data replication.

About this task

Replication Manager supports Cloudera Manager high availability functionality only after you manually configure the SSL/TLS certificate exchange.

When the source Cloudera Manager is configured for high availability and is Auto-TLS enabled, the certificate exchange is initiated from the source cluster to the target cluster where the certificate is exported from the load balancer node of the source cluster.

Important: The following sample commands use the *open-jdk-11* Java version. Use the Java version that you use in CDP clusters in these commands.

Procedure

1. Go to the truststore location in *source* Cloudera Manager, and perform the following steps:
   a) List the contents of the keystore file and password using the *[***KEYTOOL PATH***]* -list -keystore *[***TRUSTSTORE JKS FILE LOCATION ***]* -storepass *[***TRUSTSTORE PASSWORD***]* command.

      For example, /usr/lib/jvm/java-openjdk-11/bin/keytool - list -keystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks -storepass *[***TRUSTSTORE PASSWORD***]*

      Tip:
      - The keytool path can be located in various locations including the keytool itself. For example, it can be located in /usr/lib/jvm/java-openjdk-11/bin/keytool or `/usr/java/default/bin/keytool`.
      - You can locate the truststore password using the `cat /etc/hadoop/conf/ssl-client.xml` command. You can enter the SSL password for the /etc/hadoop/conf/ssl-client.xml file when prompted.
      - Alternatively, you can also run the following commands instead of the command in Step a:

```
export JAVA_HOME=[***KEYTOOL LOCATION***]

export TRUSTSTORE_JKS=[***TRUSTSTORE JKS FILE LOCATION***]

export TRUSTSTORE_PASSWORD=[***PASSWORD IN THE SSL-CLIENT.XML
 FILE***]
$JAVA_HOME/keytool -list -keystore
$TRUSTSTORE_JKS -storepass
$TRUSTSTORE_PASSWORD
```

   b) Export the certificate contents in the host to a file using the *[***KEYTOOL***]* -exportcert -keystore *[***TRUSTSTORE JKS FILE LOCATION ***]* -alias *[***CM ALIAS ON SRC CM***]* -file *./[***TXT file, for example: source-cert.txt***]* -storepass *[***TRUSTSTORE_PASSWORD***]* command.

      For example,

```
/usr/java/default/bin/keytool -exportcert -keystore /var/lib/cloudera-sc
m-agent/agent-cert/cm-auto-global_truststore.jks -alias cmrootca-0 -file
 ./source-cert.txt -storepass [***TRUSTSTORE PASSWORD***]
```

   c) Copy the text file to all the hosts of the *target* cluster Cloudera Manager securely using the scp -i *[***PEM FILE***] [***TXT file - source-cert.txt***]* root@*[***HOST IP***]*:/home/ command.
   d) Import the certificate into the keystore file on all the hosts of the *target* cluster Cloudera Manager using the *[***KEYTOOL***]* -importcert -noprompt -v -trustcacerts -keystore *[***TRUSTSTORE JKS FILE LOCATION ***]* -alias *[***CM ALIAS ON DEST CM***]* -file *./[***TXT file - source-cert.txt***]* --storepass *[***TRUSTSTORE_PASSWORD***]* command.

      For example,

```
/usr/java/default/bin/keytool -importcert -noprompt -v -trustcacerts -ke
ystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.
jks -alias cmrootca-1 -file ./source-cert.txt --storepass [***TRUSTSTORE
 PASSWORD***]
```

2. Go to the truststore location in *target* Cloudera Manager, and perform the following steps:
   a) List the contents of the keystore file and password using the *[***KEYTOOL PATH***]* -list -keystore *[***TRUSTSTORE JKS FILE LOCATION ***]* -storepass *[***TRUSTSTORE PASSWORD***]* command.
   b) Export the certificate contents in the host to a file using the *[***KEYTOOL***]* -exportcert -keystore *[***TRUSTSTORE JKS FILE LOCATION ***]* -alias *[***CM ALIAS ON DEST CM***]* -file ./*[***TXT file, for example: dest-cert.txt***]* -storepass *[***TRUSTSTORE_PASSWORD***]* command.
   c) Copy the text file to all the hosts of the *source* cluster Cloudera Manager securely using the scp -i *[***PEM FILE***] [***TXT file - dest-cert.txt***]* root@*[***HOST IP***]*:/home/ command.
   d) Import the certificate into the keystore file on all the hosts of the *source* Cloudera Manager using the *[***KEYTOOL***]* -importcert -noprompt -v -trustcacerts -keystore *[***TRUSTSTORE JKS FILE LOCATION ***]* -alias *[***CM ALIAS ON SRC CM***]* -file ./*[***TXT file - dest-cert.txt***]* --storepass *[***TRUSTSTORE PASSWORD***]* command.
3. **Note:** Perform this step only for Ozone replication policies.

   Import the S3G CA certificate from the cluster to the local JDK path using the following commands:
   a) Run the keytool -importkeystore -destkeystore *[***JDK CACERTS LOCATION***]* -srckeystore *[***CM-AUTO-GLOBAL_TRUSTSTORE.JKS LOCATION***]* -srcalias *[***CM ALIAS ON SRC CM***]* command on all the hosts of the *source* Cloudera Manager.

      For example, keytool -importkeystore -destkeystore /usr/java/default/lib/security/cacerts -srckeystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks -srcalias cmrootca-0
   b) Run the following commands on all the hosts of the target Cloudera Manager:
      1. keytool -importkeystore -destkeystore *[***JDK CACERTS LOCATION***]* -srckeystore *[***CM-AUTO-GLOBAL_TRUSTSTORE.JKS LOCATION***]* -srcalias *[***CM ALIAS ON SRC CM***]*
      2. keytool -importkeystore -destkeystore *[***JDK CACERTS LOCATION***]* -srckeystore *[***CM-AUTO-GLOBAL_TRUSTSTORE.JKS LOCATION***]* -srcalias *[***CM ALIAS ON DEST CM***]*

      Tip: Enter the security/jssecacerts path for the -destkeystore attribute if the file exists. Otherwise, enter the security/cacerts path.

      For example,

```
keytool -importkeystore -destkeystore /usr/java/default/lib/security/cac
erts
-srckeystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_tru
ststore.jks -srcalias cmrootca-0
keytool -importkeystore -destkeystore /usr/java/default/lib/security/ca
certs
-srckeystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_tr
uststore.jks -srcalias cmrootca-1
```

      Note: If you do not complete Step 3 before you create and run an Ozone replication policy, an SSL certificate exception might appear during the file listing phase of the ozone replication policy job run.

Add source cluster as peer to use in replication policies

You must assign the source cluster as a peer to replicate the data. The Cloudera Manager Server that you are logged into is the destination for replicated data. From the Admin Console of this target Cloudera Manager instance, you designate a peer Cloudera Manager Server as a source from which to replicate data. Therefore, you designate the required source Cloudera Manager instance as a peer in the target Cloudera Manager instance.

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

Adding a peer to use in replication policy

Before you replicate data from source cluster to destination cluster, you must connect the target Cloudera Manager instance with the peer (source Cloudera Manager), and then test the connectivity.

Before you begin

Consider the following points before you add a peer:

- The required source and target clusters must be healthy and available.
- If your cluster uses SAML authentication, see *Configuring peers with SAML authentication* before configuring a peer.
- Cloudera recommends that TLS/SSL be used. A unknown exception of typejavax.ws.rs.processingexception while connecting to https://*[***SOURCE CLUSTER CLOUDERA MANAGER SERVER***]*:7183 warning appears if the URL scheme is HTTP instead of HTTPS.

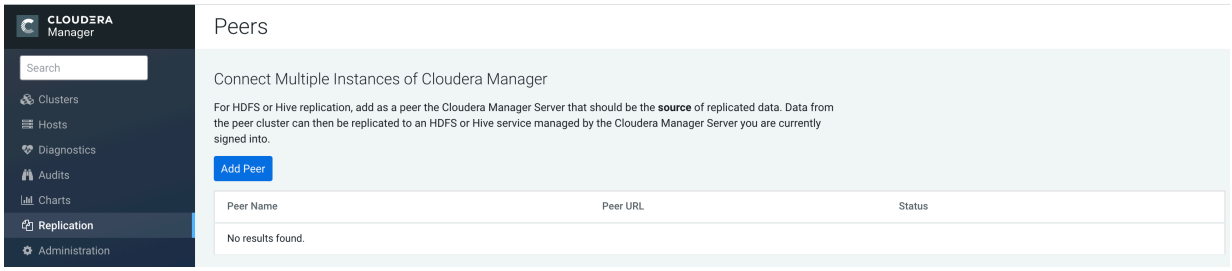
  After configuring both the peers (source and target Cloudera Manager instances) to use TLS/SSL, add the remote source cluster root CA certificate to the local Cloudera Manager truststore, and vice versa. For more information, see Configuring SSL/TLS certificate exchange between two Cloudera Manager instances
- When Cloudera Manager is configured with Knox and the source and target clusters are Knox-SSO enabled, ensure that you use the Cloudera Manager port in the peer URL when you add the source and target clusters as peers.

Procedure

1. Go to the Cloudera Manager Replication Peers page.

   If there are no existing peers, Add Peer appears along with a short message. If peers already exist, they appear in the **Peers** list.

   The following sample image shows the **Peers** page:



2. Click Add Peer.
3. Enter the following details in the Add Peer modal window:

Option	Description
Peer Name	Enter a user-friendly name for the source Cloudera Manager instance.
Peer URL	Enter the full URI for the remote source Cloudera Manager instance. This includes the URL and the port of the instance.
Peer Admin Username	Enter a username that is valid on the remote Cloudera Manager. The role assigned to the login user on the source Cloudera Manager server must be *User Administrator* or *Full Administrator*.
Peer Admin Password	Enter a password that is valid on the source remote Cloudera Manager.
Create User With Admin Role	Choose to add the peer as an admin peer. This option is mandatory to create Ranger replication policies.

4. Click Add to create the peer relationship.

Results

The peer is added to the **Peers** list. Cloudera Manager automatically tests the connection between the Cloudera Manager Server and the peer. You can also click Test Connectivity to test the connection. Test Connectivity also tests the Kerberos configuration for the clusters.

Modifying peers to use in replication policy

After you add a replication source as a peer, you can modify or delete the peers as required.

Procedure

1. Go to the Cloudera Manager Replication Peers page.
2. Select a peer, and click Actions Edit .
3. Update the peer configuration as required, and click Update Peer to save your changes.

Tip: Select a peer, and click Actions Delete to delete the peer.

Configuring peers with SAML authentication

If your cluster uses SAML Authentication, you can create a Cloudera Manager user account that has the User Administrator or Full Administrator role before you create a peer.

Procedure

1. Create a Cloudera Manager user account that has the *User Administrator* or *Full Administrator* role.

   You can also use an existing user that has one of these roles. Since you use this user to create the peer relationship, you can delete the user account after you add the peer.
2. Create or modify the peer.
3. Delete the Cloudera Manager user account that was just created.

Enabling replication between clusters with Kerberos authentication

To enable replication between clusters, additional steps are required to ensure that the source and destination clusters can communicate.

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

Important: Replication Manager works with clusters in different Kerberos realms even without a Kerberos realm trust relationship. The Cloudera Manager configuration properties Trusted Kerberos Realms and Kerberos Trusted Realms are used for Cloudera Manager and CDH configuration, and are not related to Kerberos realm trust relationships.

If you are using standalone DistCp between clusters in different Kerberos realms, you must configure a realm trust.

Required ports in Kerberos authentication-enabled clusters for replication

When using Replication Manager with Kerberos authentication-enabled clusters, ensure that the port used for Kerberos KDC Server and KRB5 services are open to all hosts on the destination cluster. By default, this is port 88.

You must also ensure that the required ports listed in the following page are open: Port and network requirements for Replication Manager on CDP Private Cloud Base on page 13.

Considerations for realm names to use for replication

You must consider the realm names if the source and destination clusters each use Kerberos for authentication before you create a replication policy.

Use one of the following configurations to prevent conflicts during replication job runs:

- If the clusters do not use the same KDC (Kerberos Key Distribution Center), Cloudera recommends that you use different realm names for each cluster. Additionally, if you are replicating across clusters in two different realms, see the steps for Prepare Kerberos authentication-enabled clusters for replication on page 24 to setup trust between those clusters.
- You can use the same realm name if the clusters use the same KDC or different KDCs that are part of a unified realm, for example where one KDC is the master and the other is a secondary KDC.

Note: If you have multiple clusters that are used to segregate production and non-production environments, this configuration could result in principals that have equal permissions in both environments. Make sure that permissions are set appropriately for each type of environment.

Important: If the source and destination clusters are in the same realm but do not use the same KDC or the KDCs are not part of a unified realm, the replication job will fail.

Prepare Kerberos authentication-enabled clusters for replication

Before you create replication policies between clusters that use Kerberos authentication, you must prepare the source and destination clusters.

Procedure

1. On the hosts in the destination cluster, ensure that the krb5.conf file (typically located at /etc/kbr5.conf) on each host has the following information:
   a) The KDC information for the source cluster's Kerberos realm. For example:

```
[realms]
 SRC.EXAMPLE.COM = {
  kdc = kdc01.src.example.com:88
  admin_server = kdc01.example.com:749
  default_domain = src.example.com
 }
 DST.EXAMPLE.COM = {
  kdc = kdc01.dst.example.com:88
  admin_server = kdc01.dst.example.com:749
  default_domain = dst.example.com
 }
```

   b) Realm mapping for the source cluster domain. You configure these mappings in the [domain_realm] section. For example:

```
[domain_realm]
 .dst.example.com = DST.EXAMPLE.COM
 dst.example.com = DST.EXAMPLE.COM
 .src.example.com = SRC.EXAMPLE.COM
 src.example.com = SRC.EXAMPLE.COM
```

   Caution: If you have a scenario where the hostname(s) are inconsistent, you must go to Cloudera Manager Host All Hosts and ensure that all those hosts are covered in a similar manner as seen in domain_realm section.

2. On the destination cluster, perform the following steps to add the realm of the source cluster to the Trusted Kerberos Realms configuration property:
   a) Go to the Cloudera Manager Clusters *CORE_SETTINGS* page.
   b) Search for the Trusted Kerberos Realms property, and enter the source cluster realm.
   c) Click Save Changes.
3. Go to the Administration Settings page.
4. Search for the Domain Name(s) field, and enter any domain or host names you want to map to the destination cluster KDC. Add as many entries as you need. The entries in this property are used to generate the domain_realm section in krb5.conf file.

5. If domain_realm is configured in the Advanced Configuration Snippet (Safety Valve) for remaining krb5.conf property, remove the entries for it.
6. Click Save Changes.

Kerberos connectivity test

As part of the Test Connectivity, Cloudera Manager tests for properly configured Kerberos authentication on the source and destination clusters that run the replication. Test Connectivity runs automatically when you add a peer for replication, or you can manually initiate Test Connectivity from the Actions menu.

Kerberos connectivity test is available when the source and destination clusters run Cloudera Manager 5.12 or later. You can disable the Kerberos connectivity test by setting feature_flag_test_kerberos_connectivity to false with the Cloudera Manager API: `api/<version>/cm/config`.

If the test detects any issues with the Kerberos configuration, Cloudera Manager provides resolution steps based on whether Cloudera Manager manages the Kerberos configuration file.

Cloudera Manager tests the following scenarios:

- Whether both the clusters are Kerberos-enabled or not.
- Replication is supported from unsecure cluster to secure cluster (starting Cloudera Manager 6.1 and later).
- Replication is not supported if the source cluster uses Kerberos and target cluster is unsecure.
- Whether both clusters are in the same Kerberos realm. Clusters in the same realm must share the same KDC or the KDCs must be in a unified realm.
- Whether clusters are in different Kerberos realms. If the clusters are in different realms, the destination cluster must be configured according to the following criteria:
  - Destination HDFS services must have the correct Trusted Kerberos Realms setting.
  - The krb5.conf file has the correct domain_realm mapping on all the hosts.
  - The krb5.conf file has the correct realms information on all the hosts.
- Whether the local and peer KDC are running on an available port. This port must be open for all hosts in the cluster. The default port is 88.

After Cloudera Manager runs the tests, Cloudera Manager makes recommendations to resolve any Kerberos configuration issues.

Kerberos recommendations

If Cloudera Manager manages the Kerberos configuration file, Cloudera Manager configures Kerberos correctly for you and then provides the set of commands that you must manually run to finish configuring the clusters.

If Cloudera Manager does not manage the Kerberos configuration file, Cloudera Manager provides the manual steps required to correct the issue.

Replicating from unsecure to secure clusters

Replication Manager can replicate data from an unsecure cluster (one that does not use Kerberos authentication) to a secure cluster (a cluster that uses Kerberos) but the reverse is not true.

About this task

Important: Replication Manager does not support replicating from a secure cluster to an unsecure cluster.

Before you replicate from an unsecure cluster to secure cluster, ensure that the following conditions are met:

- The destination cluster is managed by Cloudera Manager 6.1.0 or higher. The source cluster is managed by Cloudera Manager 5.14.0 or higher in order to be able to replicate to Cloudera Manager 6.

- Same user exists on all the hosts on both the source and destination clusters. If required, specify this user in the Run As Username field when you create a replication policy.

Note: In replication scenarios where a destination cluster has multiple source clusters, all the source clusters must either be secure or unsecure. Replication Manager does not support replication from a mixture of secure and unsecure source clusters.

Procedure

1. On a host in the source or destination cluster, add a user with the following command:

   sudo -u hdfs hdfs dfs -mkdir -p /user/*[***USERNAME***]*

   For example, the following command creates a user named milton:

   sudo -u hdfs hdfs dfs -mkdir -p /user/milton
2. Set the permissions for the user directory with the following command:

   sudo -u hdfs hdfs dfs -chown <username> /user/username

   For example, the following command makes milton the owner of the milton directory:

   sudo -u hdfs hdfs dfs -chown milton /user/milton
3. Create the supergroup group for the user you created in step 1 with the following command:

   groupadd supergroup
4. Add the user you created in step 1 to the group you created:

   usermod -G supergroup <username>

   For example, add milton to the group named supergroup:

   usermod -G supergroup milton
5. Repeat this process for all hosts in the source and destination clusters so that the user and group exists on all of them.

What to do next

After you complete this process, specify the user you created in the Run As Username field when you create a replication policy.

Replication of encrypted data

HDFS supports encryption of data at rest (including data accessed through Hive). This topic describes how replication works within and between encryption zones and how to configure replication to avoid failures due to encryption.

Encrypting data in transit between clusters

A source directory and destination directory may or may not be in an encryption zone. If the destination directory is in an encryption zone, the data on the destination directory is encrypted. If the destination directory is not in an encryption zone, the data on that directory is not encrypted, even if the source directory is in an encryption zone. Encryption zones are not supported in CDH versions 5.1 or lower.

When you configure encryption zones, you also configure Ranger Key Management Server (KMS) to manage encryption keys. To access encrypted data, the user must be authorized on the KMS for the encryption zones they need to interact with. The user you specify in the General Run As Username field during the HDFS replication policy creation process must have this authorization. The key administrator must add ACLs to the KMS for that user to prevent authorization failure. During replication, data travels from the source cluster to the destination cluster using DistCp. For clusters that use encryption zones, configure encryption of KMS key transfers between the source and destination using TLS/SSL protocol.

Note: The decryption and encryption steps happen in the same process on the hosts where the MapReduce jobs that copy the data run. Therefore, data in plain text only exists within the memory of the Mapper task. If a KMS is in use on either the source or destination clusters, and you are using encrypted zones for either the source or destination directories, configure TLS/SSL for the KMS to prevent transferring the key to the mapper task as plain text.

You might come across the following three scenarios when using encryption zones:

Scenario	Steps taken by Replication Manager to replicate data
Replicating data from an encrypted zone on the source cluster to an encrypted zone on a destination cluster.	**1.** Data is decrypted at source as it is read from the source cluster (using the key for the source encryption zone). **2.** The (decrypted) data is transferred on wire using DistCp through TLS/SSL protocol. **3.** The data is encrypted when it is written to the destination cluster (using the key for the destination encryption zone). The data transmission is encrypted only if you have configured encryption for HDFS data transfer.
Replicating from an encryption zone on the source cluster to an unencrypted zone on the destination cluster.	**1.** Data is decrypted at source as it is read from the source cluster (using the key for the source encryption zone). **2.** The (decrypted) data is transferred on wire using DistCp through TLS/SSL protocol. **3.** The data remains unencrypted.
Replicating from an unencrypted zone on the source cluster to an encrypted zone on the destination cluster.	The data is available as is after replication.
Important: Ensure that you select the Advanced Skip Checksum check property during HDFS replication policy creation for the above scenarios to avoid replication failure.	

To configure encryption of data transmission between source and destination clusters:

- Enable TLS/SSL for HDFS clients on both the source and the destination clusters. You may also need to configure trust between the SSL certificates on the source and destination.
- Enable TLS/SSL for the two peer Cloudera Manager Servers.
- Encrypt data transfer using HDFS data transfer encryption.

The following blog post provides additional information about encryption with HDFS: https://blog.cloudera.com/blog/2013/03/how-to-set-up-a-hadoop-cluster-with-network-encryption/.

Security considerations for encrypted data during replication

The user you specify in the Run As Username field during replication policy creation requires full access to both the key and the data directories being replicated. This is not a recommended best practice for KMS management. If you change permissions in the KMS to enable this requirement, you could accidentally provide access for this user to data in other encryption zones using the same key. If a user is not specified in the Run As Username field, the replication runs as the default user, hdfs.

To access encrypted data, the user must be authorized on the KMS for the encryption zones they need to interact with. The user you specify in the General Run As Username field during replication policy creation must have this authorization. The key administrator must add ACLs to the KMS for that user to prevent authorization failure.

Key transfer using the KMS protocol from source to the client uses the REST protocol, which requires that you configure TLS/SSL for the KMS. When TLS/SSL is enabled, keys are not transferred over the network as plain text.

Configuring heap size to replicate large directories using replication policies

Before you replicate the data in directories that has thousands of files and subdirectories, increase the heap size in the hadoop-env.sh file.

Procedure

1. Go to the destination Cloudera Manager *HDFS service* Configuration tab.
2. Locate the HDFS Replication Environment Advanced Configuration Snippet (Safety Valve) for hadoop-env.sh property.
3. Enter the HADOOP_CLIENT_OPTS=-Xmx*[***REQUIRED HEAP SIZE***]* key-value pair.

   For example, if you enter HADOOP_CLIENT_OPTS=-Xmx*1g*, the heap size is set to 1 GB. Adjust the heap size depending on the number of files and directories being replicated.
4. Click Save Changes.
5. Restart the HDFS service.

Retaining logs for Replication Manager

By default, Cloudera Manager retains Replication Manager logs for 90 days. You can change the number of days Cloudera Manager retains logs or disable log retention.

About this task

Important: Automatic log expiration purges custom set replication log and metadata files too. These paths are set by Log Path and Directory for Metadata arguments available in the UI as per the schedule fields. It is the user's responsibility to set valid paths (For example, specify the legal HDFS paths that are writable by current user) and maintain this information for each replication policy.

Procedure

1. Go to the Cloudera Manager *HDFS Service* Configuration tab.
2. Search for the Backup and Disaster Log Retention property.
3. Enter the number of days you want to retain the logs.

   Tip: Enter -1 to disable log retention.
4. Restart the service.

Atlas replication policies (technical preview)

You can create Atlas replication policies to replicate the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities between CDP Private Cloud Base 7.1.9 SP1 clusters using Cloudera Manager 7.11.3 CHF7 or higher. During an Atlas replication policy run, Replication Manager exports the Atlas metadata and data lineage to a staging directory in the target cluster, and then imports into the target cluster. You can enter the required staging directory during the replication policy creation process.

You can use one of the following methods to replicate Atlas metadata and data lineage for Hive external tables and Iceberg tables:

- Create Atlas replication policy to replicate the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities in the source cluster to the target cluster.

- Choose General Replicate Atlas Metadata during the Hive external table replication policy creation or edit process to replicate the metadata associated with the chosen Hive external tables.
- Choose General Replicate Atlas Metadata during the Iceberg replication policy creation or edit process to replicate the metadata associated with the chosen Iceberg tables.

Note: Replicating Atlas metadata using Hive external table replication policies and Iceberg replication policies, and replicating the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities in the source cluster to the target cluster using Atlas replication policies is a technical preview feature. It is not recommended for production deployments.

Cloudera recommends that you try this feature in development or test environments. To enable this feature, contact your Cloudera account team.

Some use cases where you can use Atlas replication policies are:

- Disaster recovery scenarios. You can back up the Atlas metadata and data lineage periodically, and restore it to the same cluster or a different cluster as required.
- High availability scenarios.
- Prevent accidental access of Ranger policies and Atlas metadata for specific Hive external tables and Iceberg tables. You can accomplish this by running both Ranger, Hive external table, and Iceberg replication policies on the required tables in the disaster-recovery cluster. The replication policies replicate the data and its associated metadata and access controls.

Preparing to create Atlas replication policies

Before you create an Atlas replication policy, you must complete the certain prerequisite tasks.

Before you begin

Note: Replicating Atlas metadata using Hive external table replication policies and Iceberg replication policies, and replicating the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities in the source cluster to the target cluster using Atlas replication policies is a technical preview feature. It is not recommended for production deployments.

Cloudera recommends that you try this feature in development or test environments. To enable this feature, contact your Cloudera account team.

Procedure

1. Enable the Atlas replication feature flag on the source and target cluster. For more information, contact your Cloudera account team.
2. Ensure that the source cluster and target cluster versions are CDP Private Cloud Base 7.1.9 SP1 or higher using Cloudera Manager version 7.11.3 CHF7 or higher versions.
3. Ensure that you have the Atlas user credentials in addition to the Replication Administrator or Full Administrator roles to create an Atlas replication policy. The `atlas` user must also have relevant read and write permissions to the staging locations.
4. Configure the following as line-separated key-value pairs for the Cloudera Manager Clusters *Atlas service* Configuration Atlas Server Advanced Configuration Snippet (Safety Valve) for conf/atlas-application.properties property on both source and target Cloudera Manager instances to optimize the Atlas replication jobs.

Key-value pair	Description
atlas.metadata.namespace=prod	Updates the Atlas Namespace to prod. Default value is cm. Add the same key-value pair for the Hive service in the target Cloudera Manager Clusters *Atlas service* Configuration Hive Service Advanced Configuration Snippet (Safety Valve) for atlas-application.properties property.

Key-value pair	Description
atlas.client.ha.retries=3	Updates the Atlas client retries for Atlas replication. Default value is 3.
atlas.client.connectTimeoutMSecs=1000000	Updates the Atlas client connection timeout for Atlas replication. Default value is 1000000.
atlas.client.readTimeoutMSecs=1000000	Updates the Atlas client read timeout for Atlas Replication. Default value is 1000000.
updateTypeDefinition=true	Updates the type definition behavior. By default, the value is true. Before you modify this property value, consult your Cloudera account team.
STAGING_RETENTION_PERIOD=5	Updates the retention count for previous Atlas replication job run's export output files. Default value is 5. Configure this parameter only for the target Cloudera Manager instance.

What to do next

You can create Atlas replication policies.

Creating Atlas replication policies

You can create Atlas replication policies to replicate the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities between CDP Private Cloud Base 7.1.9 SP1 clusters using Cloudera Manager 7.11.3 CHF7 or higher.

Before you begin

Note: Replicating Atlas metadata using Hive external table replication policies and Iceberg replication policies, and replicating the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities in the source cluster to the target cluster using Atlas replication policies is a technical preview feature. It is not recommended for production deployments.

Cloudera recommends that you try this feature in development or test environments. To enable this feature, contact your Cloudera account team.

Procedure

1. Add the source cluster as a peer to the target cluster. An Atlas replication policy requires a replication peer to locate the source data. You can use an existing peer or add a new peer.

   For information about adding a source cluster as a peer, see Adding cluster as a peer.

   Note: Peers that have Atlas service added to their clusters can be used as sources when you create an Atlas replication policy.

2. Go to the Cloudera Manager Replication Replication Policies page in the target cluster where the peer is set up.
3. Click Create Replication Policy Atlas Replication Policy .

   The **Create Atlas Replication Policy** wizard appears.
4. Configure the following options on the **General** tab:

Option	Description
Policy Name	Enter a unique name for the replication policy.
Source	Choose the source cluster that has the required peer, the required source data to replicate, and the source Atlas service.

Option	Description
Destination	Choose the target cluster. The drop-down list shows the clusters that are managed by the current Cloudera Manager.
Schedule	Choose: • Immediate to run the replication policy immediately after policy creation is complete. • Once to run the schedule one time in the future. Set the date and time. • Recurring to run the schedule periodically in the future. Set the date, time, and interval between runs. You must consider the following factors before you configure the replication frequency or recurring schedule: • The anticipated rate of change and the frequency of the schedule can predict the RTO (Recovery Time Objective) and RPO (Recovery Point Objective) during a disaster recovery process. Therefore, choose a schedule that provides an optimal RTO and RPO. • Recurring frequency impacts the compute load on the entire system. That is, frequent replication affects the overall compute capacity of the participating nodes in the replication process which in turn can impact other workloads running on these nodes.
Fetch type	Choose one of the fetch type options to use during the Atlas export operation: • FULL fetches all the directly and indirectly connected entities along with the entity in scope. • CONNECTED fetches on the directly connected entities, both parent and child entities, along with the entity in scope. **Note:** The Atlas replication policy can replicate only the ozone_key value for the Ozone tables when you choose CONNECTED Fetch Type. • INCREMENTAL fetches the optimized version of CONNECTED fetch type, that is only the delta is copied in subsequent runs.
Match type	Choose one of the following match type to use during the Atlas export operation • STARTS_WITH searches for the entity names that start with the specified criteria. • ENDS_WITH searches for the entity names that end with the specified criteria. • CONTAINS searches for an entity that has the specified criteria as a sub-string. • MATCHES searches for an entity that matches a regular expression with the specified criteria.
Entity regex filter	Enter a regex filter to filter the entities.
Atlas entity types to include	Add one or more Atlas entity types to use during the Atlas export operation.
Skip lineage	Choose to skip data lineage for the Atlas tables in the source cluster.
Staging Directory	Enter a relative path to the staging directory on the target cluster.

5. Click Create.

Results

The replication policy appears on the **Replication Policies** page. It can take up to 15 seconds for the task to appear.

If you selected Immediate in the **Schedule** field, the replication job starts replicating after you click Save Policy.

Manage, monitor, and troubleshoot Atlas replication policies

After you create a replication policy, you can run the replication job, disable or delete the job, edit the policy configuration, or view the replication job history in Replication Manager.

Note: Replicating Atlas metadata using Hive external table replication policies and Iceberg replication policies, and replicating the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities in the source cluster to the target cluster using Atlas replication policies is a technical preview feature. It is not recommended for production deployments.

Cloudera recommends that you try this feature in development or test environments. To enable this feature, contact your Cloudera account team.

Replication policy details on the Replication Policies page

On the Cloudera Manager Replication Replication Policies page, you can view the following details about the replication policy:

- Shows a row of information for each replication policy, and the following columns for each replication policy:
  - Internally generated **ID** for the replication policy. Click the column label to sort the replication policies.
  - Replication policy **Name** that you provide during replication policy creation.
  - Replication policy **Type**.
  - **Source** cluster in the replication policy.
  - **Destination cluster** in the replication policy.
  - Average **Throughput** per mapper/file for all the files written.

    Note: The throughput does not include the combined throughput of all mappers and the time taken to perform a checksum on a file after the file is written.
  - Replication job **Progress**.
  - Timestamp when the replication job **Completed**.
  - Replication policy job's **Next Run**.
- Provides the following options under the **Actions** menu:
  - **Show History** opens the **Replication History** page for the replication policy.
  - Edit Configuration enables you to change the replication policy options.
  - Dry Run simulates a run of the replication job where no files or tables are copied. After the dry run completes, select Actions Show History to view the potential error messages. The number and size of files or tables that are copied in an actual replication appears on the **Replication History** page.
  - **Run Now** initiates a replication job.
  - Collect Diagnostic Data opens the **Send Diagnostic Data** dialog box where you can:
    - Collect Diagnostic Data for the last 10 runs of the replication policy, and Download it as a ZIP file to your machine.
    - Select Send Diagnostic Data to Cloudera (optionally, add a Cloudera support ticket number and comments) and click Collect Diagnostic Data to automatically send the bundle to Cloudera Support for further assistance.
  - **Disable** an active replication policy. You can Enable it later, as necessary.
- **Delete** the replication policy permanently. Deleting a replication policy does not delete copied files or tables.

Replication History page

Click Actions Show History for a replication policy on the **Replication Policies** page to view the **Replication History** page.

On the **Replication History** page, you can view the following run details about a replication policy job:

- Shows the replication policy **Name**; replication policy **Type**; **Source** cluster name; **Destination** cluster name; and **Next Run** of the replication policy.
- Shows a row of information for each replication policy job run, and the following columns for each replication policy:

Column	Description
Start Time	Shows the timestamp when the replication job started.
Duration	Shows the time taken to complete the replication job.
Outcome	Shows the replication job status as **Running**, **Successful**, or **Failed**.
Atlas Entities Replicated	Shows the number of tables for which the Atlas metadata and lineage is being replicated.
Export Status	Shows the current status as **Running**, **Successful**, or **Failed** of the export process of Atlas metadata and data lineage from the source cluster to the staging directory on the target cluster.
Import Status	Shows the current status as **Running**, **Successful**, or **Failed** of the import process of the Atlas metadata and data lineage into the required directory on the target cluster.

- Expand a job to view the following information on the **All Recent Commands** window:
  - **Status** of the replication job.
  - **Atlas Replication** in the **Context** field opens the Clusters Atlas Replication window where more details about the replication policy job appears.
  - Replication job **Started At** timestamp.
  - **Duration** to complete the job.
  - **Download** the results to your machine.
  - Expand to **Show All Steps**, **Show Only Failed Steps**, or **Show Only Running Steps** for the commands used by Atlas replication policy.
  - **Show Command Timing** shows the timeline for the commands used by the Atlas replication policy.

Error appears during Atlas replication policy run

You can diagnose the errors that appear after you initiate an Atlas replication policy run or during the replication policy run.

Solution

Procedure

- If the error appears after you initiate the Atlas replication policy run, you can perform the following steps to diagnose the error:
  a) Go to the target Cloudera Manager Replication Replication Policies page.
  b) Click Actions Show History for the required Atlas replication policy.
  c) Expand the section in the **Start Time** column.
  d) Click **Command Details** to view the **stdout** and **stderr** tab to diagnose the error.
  e) Open the cloudera-scm-server.log file located in the /var/log/cloudera-scm-server/ location if you require more details to diagnose the issue.
- If the error appears during the Atlas replication policy run, you can perform the following steps to diagnose the error:
  a) Go to the Cloudera Manager Running Commands page.
  b) Click the `Atlas Server` link on the given node.
  c) Open the application.log file or Role logs to diagnose the error.

HDFS replication policies

HDFS replication policies enable you to copy (replicate) your HDFS data from one HDFS service to another and synchronize the data set on the destination service with the data set on the source service. The destination service must be managed by the Cloudera Manager Server where the replication is being set up, and the source service can be managed by that same server or by a peer Cloudera Manager Server. You can also replicate HDFS data within a cluster by specifying different source and destination directories.

Remote Replication Manager automatically copies HDFS metadata to the destination cluster as it copies files. HDFS metadata need only be backed up locally.

Note:

- Replication Manager requires a valid license. To understand more about Cloudera license requirements, see Managing Licenses.
- Minimum required role - Replication Administrator or Full Administrator.
- Before you create replication policies, ensure that the source cluster and target cluster are supported by Replication Manager. For information about supported clusters and supported replication scenarios by Replication Manager, see Support matrix for Replication Manager on CDP Private Cloud Base on page 9.

HDFS replication policy considerations

Before you create an HDFS replication policy, you must understand how source data is affected when you add or delete source data during replication, the network latency issues, the performance and scalability limitations, the snapshot diff-based replication guidelines, and how to bypass Sentry ACLs during replication.

Guidelines to add or delete source data during replication job run

When a replication policy is replicating data, you must ensure that you follow a few guidelines to maintain source data for successful data replication.

Follow the below guidelines for successful data replication:

- Do not modify the source directory. This is because a file added during replication is not replicated, and the replication fails if you delete a file during replication.
- All the files in the directory are closed. This is because replication fails if any source files are open.

Tip: If you cannot ensure that all source files are closed, clear the Abort on Error option in the replication policy to continue replication despite errors. After the replication job completes, identify the opened files in the log. Ensure that these files are closed before the next replication occurs.

Improve network latency during replication job run

High latency among clusters can cause replication jobs to run more slowly, but does not cause them to fail.

For best performance, latency between the source cluster NameNode and the destination cluster NameNode should be less than 80 milliseconds. You can test latency using the Linux ping command. Cloudera has successfully tested replications with latency of up to 360 milliseconds. As latency increases, replication performance degrades.

Performance and scalability limitations to consider for replication policies

Before you create an HDFS replication policy, you must consider a few performance and scalability limitations.

The performance and scalability limitations include:

- Maximum number of files for a single replication job is 100 million.
- Maximum number of files for a replication policy that runs more frequently than once in 8 hours is 10 million.

- Throughput of the replication job depends on the absolute read and write throughput of the source and destination clusters.
- Regular rebalancing of your HDFS clusters is required for efficient operation of replications.

Note: Cloudera Manager provides downloadable data that you can use to diagnose HDFS replication performance.

Guidelines to use 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 can take advantage of snapshot diff-based replication.

After every replication, the Replication Manager retains a snapshot on the source cluster. Replication Manager uses the snapshot copy on the source cluster to perform incremental backup for the next replication cycle.

Replication Manager retains snapshots on the source cluster and uses snapshot diff-based replication only if:

- Source and target clusters are managed by Cloudera Manager 5.15 and higher.
- Source cluster is managed by Cloudera Manager 5.15.0 or higher when the destination is Amazon S3 or Microsoft ADLS.

  Important: Snapshot-diff-based replication from S3/ABFS to HDFS is not supported because S3/ABFS does not support snapshots.

- Source and target CDH versions are 5.13.3 or higher, 5.14.2 or higher, and 5.15 or higher.

The following guidelines must be met to use snapshot diff-based replication efficiently in replication policies:

- Source and target clusters are managed by Cloudera Manager 5.15.0 or higher.
- Source and target clusters run CDH version 5.15.0 or higher, 5.14.2 or higher, or 5.13.3 or higher.
- HDFS snapshots are immutable.

  Tip: Search for Enable Immutable Snapshots option in the Cloudera Manager Clusters *HDFS service* Configuration tab.

- Snapshot root directory is set as low in the hierarchy as possible.
- User used to create and run the replication policy is a super user or the owner of the snapshottable root. This is because the run-as-user (specified in the replication policy) must have the required permissions to list the snapshots.
- Paths from both source and destination clusters in the replication policy must be present under a snapshottable root, or must be snapshottable.

  Tip: 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.

- All the HDFS paths for the tables in a database is snapshottable or under a snapshottable root for a Hive replication policy to replicate the database successfully.

  For example, if the database being replicated has external tables, all the external table HDFS data locations should be snapshottable. This is because if the external table locations are not snapshottable, Replication Manager does not generate a diff report. The Replication Manager needs a diff report to use the snapshot diff feature.

Important: Do not use snapshot diff for globbed paths because it is not optimized for globbed paths.

FAQs

What do I do when snapshot diff-based replication fails because an encrypted subdirectory exists in the source data?

To resolve this issue, create an exclusion regex in the replication policy to exclude the subdirectory during replication. Create another replication policy to replicate the encrypted subdirectory.

During what circumstances does the Replication Manager initiate a complete data replication?

Replication Manager initiates a complete replication for the following scenarios:

- When you do not choose Abort on Snapshot Diff Failures (when you create a replication policy in Replication Manager) and errors appear during the replication process.

  In this case, the Replication Manager continues to replicate and performs a complete replication after it encounters an error.
- When one or more of the following parameters that you set in the replication policy changes:
  - Delete Policy
  - Preserve Policy
  - Target Path
  - Exclusion Path.
- When a change in the target directories is detected.

  Replication Manager ensures that the next HDFS snapshot replication is a complete replication.

HDFS replication in Sentry-enabled clusters

When you run an HDFS replication policy on a Sentry-enabled source cluster, the replication policy copies files and tables along with their permissions. Cloudera Manager version 6.3.1 and above is required to run HDFS replication policies on a Sentry-enabled source cluster.

Before you begin

To perform Sentry to Ranger replication using HDFS replication policies, you must have installed Cloudera Manager version 6.3.1 and higher on the source cluster and Cloudera Manager version 7.1.1 and higher on the target cluster. Use the hdfs user to run HDFS replication policies on a source cluster that is Sentry-enabled. To use a different user account, you must configure the user account to bypass the Sentry ACLs during the replication process.

Consider the following points before you create an HDFS replication policy:

- When Sentry is not available or when Sentry does not manage the authorization for a resource such file or directory in the source cluster, HDFS uses its internal ACLs to manage resource authorization.
- When Sentry is enabled for the source cluster and you use the hdfs user to create the HDFS replication policy, HDFS copies the ACLs configured in Sentry for the replicated files and tables to the target cluster.
- When Sentry is enabled and you use a different user name to run the HDFS replication policy, both Sentry ACLs and HDFS internal ACLs are copied which results in incorrect HDFS metadata in the target cluster. If the Sentry ACLs are not compatible with HDFS ACLs, the replication job fails. Create another user to bypass the Sentry ACLs during the replication process to avoid such compatibility issues.

To avoid compatibility issues between HDFS and Sentry ACLs for a non-hdfs user, you must complete the following steps:

Procedure

1. Create a user account that Replication Manager jobs can use to bypass the Sentry ACLs.

   For example, create a user named bdr-only-user.

2. Perform the following steps on the source cluster:
   a) In the Cloudera Manager Admin Console, go to the Clusters *HDFS service* Configuration tab.
   b) Search for NameNode Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml property.
   c) Enter the following property details:

      Name - Enter dfs.namenode.inode.attributes.provider.bypass.users.

      Value - Enter *[***USERNAME, USERNAME@REALMNAME***]*, where *[***USERNAME***]* is the user you created in step 1 and the *[***REALMNAME***]* is the Kerberos realm name.

      For example, if the username is bdr-only-user on the realm elephant, enter **bdr-only-user, bdr-only-user@ElephantRealm**
   d) Restart the NameNode.
3. Repeat step 2 on the destination cluster.
4. When you create an HDFS replication policy, specify the user you created in step 1 in the Run As Username and Run on Peer as Username fields.

   Note: The Run As Username field launches the MapReduce job to copy data. The Run on Peer as Username field runs copy listing on source, if different than Run as Username.

What to do next

Note: Ensure that you set the value of Run on Peer as Username same as Run as Username. Otherwise, Replication Manager reads ACL from the source as hdfs, which pulls the Sentry provided ACLs over to the target cluster and applies them to the files in HDFS. This can result in additional usage of NameNode heap in the target cluster.

Specifying hosts to improve HDFS replication policy performance

If your cluster has clients installed on hosts with limited resources, HDFS replication may use these hosts to run commands for the replication, which can cause performance degradation. You can limit HDFS replication to run only on selected DataNodes by specifying a "whitelist" of DataNode hosts.

Procedure

1. Go to the Cloudera Manager Clusters *HDFS service* Configuration tab.
2. Locate the HDFS Replication Environment Advanced Configuration Snippet (Safety Valve) property.
3. Add the HOST_WHITELIST property, and enter a comma-separated list of hostnames to use for HDFS replication policies.

   For example,

   ```
   HOST_WHITELIST=host-1.mycompany.com,host-2.mycompany.com
   ```
4. Click Save Changes.

Creating HDFS replication policy to replicate HDFS data

You must set up your clusters before you create an HDFS replication policy. You can also use CDP Private Cloud Base Replication Manager to replicate HDFS data from on-premises to cloud, however you cannot replicate data from one cloud instance to another using Replication Manager.

Before you begin

To replicate HDFS data from on-premises to cloud, you must have the appropriate credentials to access the cloud account. Additionally, you must create buckets in S3 and GCP or Data Lake store in ADLS. Replication Manager backs up file metadata, including extended attributes and ACLs when you replicate data to cloud storage.

Procedure

1. Verify whether your cluster conforms to one of the supported replication scenarios. For more information, see Support matrix for Replication Manager on CDP Private Cloud Base on page 9
2. If you are using different Kerberos principals for the source and destination clusters, add the destination principal as a proxy user on the source cluster. For example, if you are using the hdfssrc principal on the source cluster and the hdfsdest principal on the destination cluster, add the following properties to the HDFS service Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml property on the source cluster:

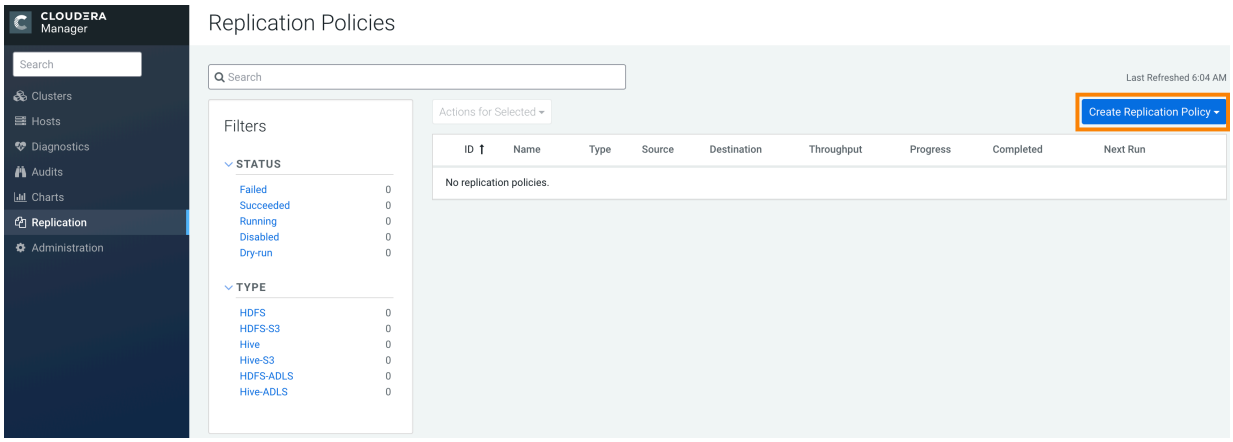
```
<property>
    <name>hadoop.proxyuser.hdfsdest.groups</name>
    <value>*</value>
</property>
<property>
    <name>hadoop.proxyuser.hdfsdest.hosts</name>
    <value>*</value>
</property>
```

   Deploy the client configuration and restart all services on the source cluster, if the source cluster is managed by a different Cloudera Manager server than the destination cluster.
3. Add the required credentials in Cloudera Manager to access the cloud storage to replicate HDFS to/from cloud storage.
   a) To add AWS credentials, see How to Configure AWS Credentials.

      Ensure that the following basic permissions are available to provide read-write access to S3 through the S3A connector:

```
s3:Get*
s3:Delete*
s3:Put*
s3:ListBucket
s3:ListBucketMultipartUploads
s3:AbortMultipartUpload
```

   b) To add ADLS credentials, perform the following steps:
      1. Click Add AD Service Principal on the Cloudera Manager Admin Console Administration External Accounts Azure Credentials page for the source cluster.
      2. Enter the Name, Client ID, Client Secret Key, and Tenant Identity for the credential in the **Add AD Service Principal** modal window.
      3. Click Add.
4. Click Create Replication Policy on the Cloudera Manager Replication Replication Policies page.



5. Select HDFS Replication Policy.

   The **Create HDFS Replication Policy** wizard appears.

6. Configure the following options on the **General** page:

Option	Description
Name	Enter a unique name for the replication policy.
Source	Select the source HDFS service. You can select HDFS services managed by a peer Cloudera Manager Server, local HDFS services (managed by the Cloudera Manager Server for the Admin Console you are logged into).
Source Path	Enter one of the following values depending on your source cluster: • Directory (or file) on the on-premises cluster. • s3a://*[***BUCKET NAME***]/[***PATH***]* path to replicate from Amazon S3. • adl://*[***ACCOUNTNAME***]*.azuredatalakestore.net/*[***PATH***]* path to replicate from ADLS Gen 1. • abfs[s]://*[***FILESYSTEM***]@[***ACCOUNT NAME***]*.dfs.core.windows.net/*[***PATH***]*/ path to replicate from ADLS Gen 2. You can also use a glob path to specify more than one path for replication.
Destination	Select the destination HDFS service from the HDFS services managed by the Cloudera Manager Server for the Admin Console you are logged into.
Destination Path	Enter one of the following values to save the source files: • Directory (or file) on the on-premises cluster. • s3a://*[***BUCKET NAME***]/[***PATH***]* path to replicate to Amazon S3. • adl://*[***ACCOUNT NAME***]*.azuredatalakestore.net/*[***PATH***]*path to replicate to ADLS Gen 1. • abfs[s]://*[***FILESYSTEM***]@[***ACCOUNT NAME***]*.dfs.core.windows.net/*[***PATH***]*/ path to replicate to ADLS Gen 2.
Schedule	Choose: • Immediate to run the schedule immediately. • Once to run the schedule one time in the future. Set the date and time. • Recurring to run the schedule periodically in the future. Set the date, time, and interval between runs. Replication Manager ensures that the same number of seconds elapse between the runs. For example, if you choose the Start Time as January 19, 2022 11.06 AM and Interval as 1 day, Replication Manager runs the replication policy for the first time at the specified time in the timezone the replication policy was created in, and then runs it exactly after 1 day that is, after 24 hours or 86400 seconds.

Option	Description
Run As Username	Enter the user to run the replication job in the field. By default this is hdfs. If you want to run the job as a different user, enter the user name. If you are using Kerberos, you must provide a user name here, and it must be one with an ID greater than 1000. (You can also configure the minimum user ID number with the min.user.id property in the YARN or MapReduce service.) Verify whether the user running the job has a home directory, /user/username, owned by username:supergroup in HDFS. This user must have permissions to read from the source directory and write to the destination directory. Note the following: • The user must not be present in the list of banned users specified with the Banned System Users property in the YARN configuration. For security purposes, the hdfs user is banned by default from running YARN containers. • The requirement for a user ID that is greater than 1000 can be overridden by adding the user to the "white list" of users that is specified with the Allowed System Users property. To view the properties, go to the YARN service and search for the properties on the Configuration tab.
Run on peer as Username	Enter the username if the peer cluster is configured with a different superuser. This is applicable in a kerberized environment.

7. Configure the following options on the Resources page:

Option	Description
Scheduler Pool	(Optional) Enter the name of a resource pool in the field. The value you enter is used by the MapReduce Service you specified when Cloudera Manager executes the MapReduce job for the replication. The job specifies the value using one of these properties: • MapReduce – Fair scheduler: mapred.fairscheduler.pool • MapReduce – Capacity scheduler: queue.name • YARN – mapreduce.job.queuename
Maximum Map Slots	Enter the number of map tasks that the DistCp MapReduce job can use for the replication policy. Default is 20.
Maximum Bandwidth	Enter the bandwidth limit for each mapper. Default is 100 MB. The total bandwidth used by the replication policy is equal to Maximum Bandwidth multiplied by Maximum Map Slots. Therefore, you must ensure that the bandwidth and map slots you choose do not impact other tasks or network resources in the target cluster. **Tip:** The Throughput field on the Cloudera Manager Replication Policies page shows the maximum bandwidth set for the replication policy during replication policy creation.
File listing threads	Choose the Override DistCp default option and configure the number of threads (a maximum of 128 threads) that the HDFS replication policy must use during the copylisting phase of replication. By default, Replication Manager uses the default value of 20 threads for the copylisting phase of replication. The default number of threads for the copylisting phase of replication (using replication policies) can be set in the core-site.xml or hdfs-site.xml file for the HDFS service. You can set a maximum of 128 threads only. **Important:** Increasing this value increases the load on the HDFS NameNode of the source cluster which in turn increases the network bandwidth used by the replication jobs.

Option	Description
Replication Strategy	Choose Static or Dynamic. Determines whether the file replication tasks must be distributed among the mappers statically or dynamically. The default is Dynamic. Static replication distributes file replication tasks among the mappers up front to achieve a uniform distribution based on the file sizes. Dynamic replication distributes file replication tasks in small sets to the mappers, and as each mapper completes its tasks, it dynamically acquires and processes the next unallocated set of tasks.

8. Configure the following options on the Advanced Options tab:

Option	Description

Add Exclusion	Click the link to exclude one or more paths from the replication. Enter a regular expression-based path in the Regular Expression-Based Path Exclusion field.
	
	When you add an exclusion, include the snapshotted relative path for the regex. For example, to exclude the /user/bdr directory, use the following regular expression, which includes the snapshots for the bdr directory:
	
	`.*/user/\.snapshot/.+/bdr.*`
	
	To exclude top-level directories from replication in a globbed source path, specify the relative path for the regex without including .snapshot in the path. For example, to exclude the bdr directory from replication, use the following regular expression:
	
	`.*/user+/bdr.*`
	
	You can add more than one regular expression to exclude.

MapReduce Service	Select the MapReduce or YARN service to use.
Log path	Enter an alternate path for the logs.
Description	Enter a description of the replication policy.

Option	Description
Error Handling	Select the following option based on your requirements: • Skip Checksum Checks - Determines whether to skip checksum checks on the copied files. If selected, checksums are not validated. Checksums are checked by default. **Important:** You must skip checksum checks to prevent replication failure due to non-matching checksums in the following cases: • Replications from an encrypted zone on the source cluster to an encrypted zone on a destination cluster. • Replications from an encryption zone on the source cluster to an unencrypted zone on the destination cluster. • Replications from an unencrypted zone on the source cluster to an encrypted zone on the destination cluster. Checksums are used for two purposes: • To skip replication of files that have already been copied. If Skip Checksum Checks is selected, the replication job skips copying a file if the file lengths and modification times are identical between the source and destination clusters. Otherwise, the job copies the file from the source to the destination. • To redundantly verify the integrity of data. However, checksums are not required to guarantee accurate transfers between clusters. HDFS data transfers are protected by checksums during transfer and storage hardware also uses checksums to ensure that data is accurately stored. These two mechanisms work together to validate the integrity of the copied data. • Skip Listing Checksum Checks - Determines whether to skip checksum check when comparing two files to determine whether they are same or not. If skipped, the file size and last modified time are used to determine if files are the same or not. Skipping the check improves performance during the mapper phase. Note that if you select the Skip Checksum Checks option, this check is also skipped. • Abort on Error - Determines whether to abort the job on an error. If selected, files copied up to that point remain on the destination, but no additional files are copied. Abort on Error is not selected by default. • Abort on Snapshot Diff Failures - If a snapshot diff fails during replication, Replication Manager uses a complete copy to replicate data. If you select this option, the Replication Manager aborts the replication when it encounters an error instead. • Restart replication using non-incremental (bootstrap) replication on replication failure - Select to run the next replication job as a bootstrap replication if the replication job fails. Replication Manager replicates all the specified directories and files in the first HDFS replication policy job. This is also called bootstrap replication or non-incremental replication. Subsequent replication jobs are snapshot-based incremental replication. **Important:** When you choose this option for HDFS replication policies that are replicating huge amounts of data (otherwise called large policies), Replication Manager takes a long time to complete the bootstrap replication. This operation overwrites existing data and might consume more network resources to replicate the large amount of data.

Option	Description
Preserve	Whether to preserve the block size, replication count, permissions (including ACLs), and extended attributes (XAttrs) as they exist on the source file system, or to use the settings as configured on the destination file system. By default source system settings are preserved. When Permission is checked, and both the source and destination clusters support ACLs, replication preserves ACLs. Otherwise, ACLs are not replicated. To preserve permissions to HDFS, you must be running as a superuser on the destination cluster. Use the Run As Username option to ensure that is the case. When Extended attributes is checked, and both the source and destination clusters support extended attributes, replication preserves them. This option appears when both the source and destination clusters support extended attributes. When you preserve the attributes on the destination cluster, the HDF replication factor is also preserved.
Delete Policy	Determines whether files that were deleted on the source should also be deleted from the destination directory. This policy also determines the handling of files in the destination location that are unrelated to the source. Options include: • Keep Deleted Files - Retains the destination files even when they no longer exist at the source. (This is the default.). • Delete to Trash - If the HDFS trash is enabled, files are moved to the trash folder. • Delete Permanently - Uses the least amount of space; use with caution. This option does not delete the files and directories in the top level directory. This is in line with rsync/Hadoop DistCp behavior.
Alerts	Choose to generate alerts for various state changes in the replication workflow. You can choose to generate an alert On Failure, On Start, On Success, or On Abort of the replication job. You can configure alerts to be delivered by email or sent as SNMP traps. If alerts are enabled for events, you can search for and view the alerts on the **Events** tab, even if you do not have email notification configured. For example, if you choose Command Result that contains the Failed filter on the Diagnostics Events page, the alerts related to the On Failure alert for all the replication policies for which you have set the alert appear. For more information, see Managing Alerts and Configuring Alert Delivery.

9. Click Save Policy.
The replication policy appears in the **Replication Policies** table. It can take up to 15 seconds for the task to appear.

If you selected Immediate in the Schedule field, the replication job starts replicating after you click Save Policy.

- If your replication job takes a long time to complete, see Improve network latency during replication job run to improve network latency.
- If files change before the replication finishes, the replication might fail. For more information, see Guidelines to add or delete source data during replication job run.
- For efficient replication, consider making the directories snapshottable. For more information, see Guidelines to use snapshot diff-based replication.
- If your cluster has clients installed on hosts with limited resources, HDFS replication may use these hosts to run commands for the replication, which might cause performance degradation. To limit HDFS replication to run only on selected DataNodes, you can specify a "whitelist" of DataNode hosts. For more information, see Specifying hosts to improve HDFS replication policy performance.

How to use the post copy reconciliation script for HDFS replication policies

CDP Private Cloud Base versions 7.7.1 CHF22 and 7.11.3 CHF8 and higher support the latest version of PCR or Post Copy Reconciliation script. You can use different methods to run the PCR script on your HDFS replication policies depending on your requirements. You can run the PCR script for HDFS replication policies between on-premises clusters if you are using the supported target cluster version. You can also set the options to record the debug information and leverage the extra logging capabilities for troubleshooting purposes.

Some use cases where you can use the PCR script are:

- When replicating large amounts of data. You might want to verify whether all the data was replicated successfully.
- After a recovery/failover scenario. You might want to check data integrity.
- When there is a change on target but no snapshot for it is available on the target. You might want to verify if the data on the source and target are in sync.

What is a PCR script?

The PCR script validates the data that is replicated using the HDFS replication policy. It checks whether the HDFS replication was successful by verifying whether the source and target locations have the same content. It accomplishes this task by performing a full file listing on the source and target after replication. It then uses the file listing to compare the following attributes:

Paths of source and target data

The PCR script compares this attribute by default.

File sizes

You can disable this comparison using the pcrEnableLengthCheck=false query parameter in the PCR API.

File last modification time

You can disable this comparison using the pcrEnableModtimeCheck=false query parameter in the PCR API.

Cyclic redundancy check (CRC) checksums

PCR checks this attribute when available. You can disable this comparison using the pcrEnableCrc Check=false query parameter in the PCR API. For example, /clusters/*[***CLUSTER NAME***]*>/ services/*[***SERVICE***]*/replications/*[***SCHEDULE ID***]*/postCopyReconciliation?pcrE nableCrcCheck=false&pcrEnableModtimeCheck=false

To compare checksums, the source must support the checksum extension for the "DistCpFileStatus" class. PCR compares checksums for HDFS replication policies only if the following conditions are met:

- Replication is between on-premises clusters
- Both the source and target clusters must support the checksum extension
- Target clusters support PCR
- Source and target files are not encrypted
- Source and target files have the same block size

To write the checksums for PCR, you can enable one of the following:

- Enable the CRC FileStatus extension only for PCR by setting the ENABLE_FILESTATUS_CR C_FOR_ADDITIONAL_DEBUG_STEPS = true key-value pair in the target Cloudera Manager Clusters *HDFS service* Configuration hdfs_replication_env_safety_valve property.

- Enable the CRC FileStatus extension globally by setting the following key-value pairs for the target Cloudera Manager Clusters *HDFS service* Configuration hdfs_replication_env_safety_valve property:
  - ENABLE_FILESTATUS_EXTENSIONS = true
  - ENABLE_FILESTATUS_CRC_EXTENSION= true

Replication Manager performs the following steps when you run the PCR script or when you include PCR command step in an HDFS replication policy:

1. Checks whether the snapshots are available on the source and target. When available, the snapshot is listed in the next command step, otherwise, the source and target directories are listed directly.
2. Performs a full file listing on the source and target. If the source supports file listing, the source file listing runs as a remote command on the source. The listing file is then transferred to the target. File listing of the source and target happens in parallel.
3. Runs the PCR to compare the two file listings after which the results are saved in the mismatch_paths.tsv file and all_paths.tsv file (if enabled). If a fail-on status is detected, the replication policy run fails.

The PCR and the replication runs for the same replication job must not overlap. If they overlap, the replication run is not impacted but the PCR results become unreliable. Therefore, do not run the PCR script when the replication run is active.

The debug output of PCR is available in the mismatch_paths.tsv file on the target HDFS, and is saved in the $logDir/debug directory. For example, hdfs://user/hdfs/.cm/distcp/2023-08-24_206/debug/mismatch_paths.tsv.

If you want to restore the earlier format of PCR, set the com.cloudera.enterprise.distcp.post-copy-reconciliation.legacy-output-format.enabled = true key value pair in the target Cloudera Manager Clusters *HDFS service* Configuration hdfs_replication_hdfs_site_safety_valve property.

Different methods to run PCR

You can use one of the following methods to run PCR on an HDFS replication policy:

Run the PCR script using API

Use the /clusters/*[***CLUSTER NAME***]*>/services/*[***SERVICE***]*/replications/*[***SCHEDULE ID***]*/postCopyReconciliation API.

When you set the API parameters, you can choose to compare one or all the supported attributes (file size, file modification time, and CRC checksums) during the PCR script run. By default, the checks for these attributes are enabled.

Include PCR as part of replication job

To include the PCR script in an HDFS replication policy as a command step, enter the SCHEDULES_WITH_ADDITIONAL_DEBUG_STEPS” = *[***ENTER COMMA-SEPARATED LIST OF NUMERICAL IDS OF THE REPLICATION POLICIES***]* key-value pair in the target Cloudera Manager Clusters *HDFS service* Configuration hdfs_replication_env_safety_valve property, and then run the replication policy. The PCR step is added automatically to subsequent replication runs. In this method, PCR runs as a command step and does not interfere with the replication process.

You can also enable the checks individually for each attribute (file size, file modification time, and CRC checksums) when you include the PCR script as part of the replication job. You can accomplish this task by setting the following variables to true in the target Cloudera Manager Clusters *HDFS service* Configuration hdfs_replication_hdfs_site_safety_valve property before you run the replication policy. Replication Manager validates only the attributes that you have set to true:

- com.cloudera.enterprise.distcp.post-copy-reconciliation.length-check.enabled
- com.cloudera.enterprise.distcp.post-copy-reconciliation.modtime-check.enabled
- com.cloudera.enterprise.distcp.post-copy-reconciliation.crc-check.enabled

Debug and extra logging for PCR

Additionally, you can perform the following steps to enable the debug steps and extra logging for PCR which might assist you to troubleshoot issues:

- To save the debug-related information, enter the following key-value pairs in the target Cloudera Manager Clusters *HDFS service* Configuration hdfs_replication_hdfs_site_safety_valve property:
  - com.cloudera.enterprise.distcp.post-copy-reconciliation.fail-on = MISSING_ON_TARGET, MISSING_ON_SOURCE, OTHER_MISMATCH, ANY_MISMATCH, or NONE

    The mismatch_paths.tsv file is updated.
  - com.cloudera.enterprise.distcp.post-copy-reconciliation.all-paths=true

    An entry is added to the all_paths.tsv file for each compared path.
- To initiate and save extra logging information, enter the EXTRA_LOG_CONFIGS_*[***NUMERICAL ID OF THE REPLICATION POLICY***] = [***VALUE***]* key-value pair in the target Cloudera Manager Clusters *HDFS service* Configuration hdfs_replication_env_safety_valve property.

  For example, if your on-premises cluster is on Microsoft Azure, the value is

```
log4j.rootLogger=INFO,console;hadoop.root.logger=INFO,console;log4j.appe
nder.console=org.apache.log4j.ConsoleAppender;log4j.appender.console.tar
get=System.err;log4j.appender.console.layout=org.apache.log4j.PatternLay
out;log4j.appender.console.layout.ConversionPattern=%d{yy/MM/dd
          HH:mm:ss} %p %c{2}:
          %m%n;log4j.logger.org.apache.hadoop.fs.azurebfs.services.AbfsIoU
tils=DEBUG,console;log4j.logger.org.apache.hadoop.fs.azurebfs.services.A
bfsClient=DEBUG,console;log4j.logger.distcp.SimpleCopyListing=DEBUG,cons
ole;log4j.logger.distcp.SnapshotDiffGenerator=DEBUG,console
```

The extra debug logs are available in the $logDir/debug file. For example, hdfs://user/hdfs/.cm/distcp/2023-08-24_206/debug.

View HDFS replication policy details

The Replications Policies page displays a row of information about each replication policy which includes recent messages about the last replication job run.

You can limit the replication jobs that are displayed by selecting filters on the left. If you do not see an expected policy, adjust or clear the filters. Use the search box to search the list of replication policies for path, database, or table names.

Note: Only one job corresponding to a replication policy can occur at a time; if another job associated with that same replication policy starts before the previous one has finished, the second one is canceled.

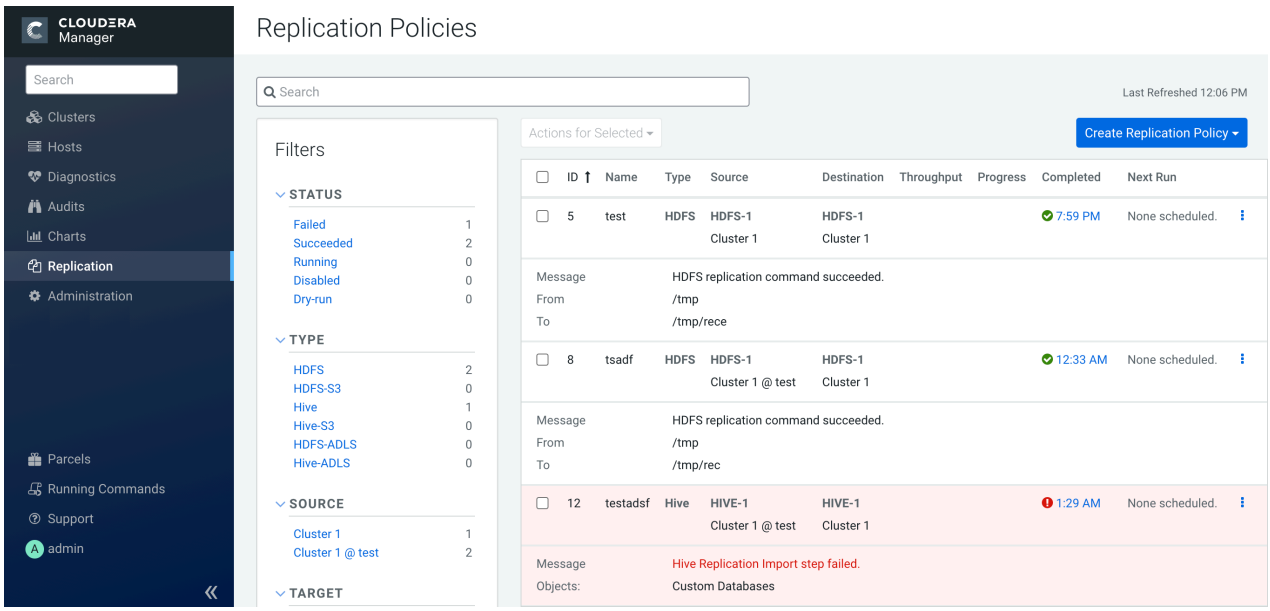
The following table describes the columns in the Replication Policies page:

Column	Description
ID	Internally generated ID number for the replication policy. Provides a convenient way to identify a policy. Click the ID column label to sort the replication policies table by ID.
Name	Unique name you specify when you created the replication policy. Click the Name column label to sort the replication policies table by name.
Type	Shows HDFS or Hive as the replication policy type.
Source	Source cluster for the replication.
Destination	Target cluster for the replication.

Column	Description
Throughput	Average throughput per mapper/file of all the files written. **Note:** The throughput does not include the combined throughput of all mappers and the time taken to perform a checksum on a file after the file is written.
Progress	Current replication job status.
Completed	Time stamp when the replication job completed. Click the Completed column label to sort the replication policies table by time.
Next Run	Date and time for the next scheduled replication which depends on the schedule parameters you specified during policy creation. Hover over the date to view additional details about the scheduled replication. Click the Next Run column label to sort the replication policies table by the next run date.
Actions	Click: • Show History to open the Replication History page for a replication policy. • Edit Configuration to change the replication policy options as required. • Dry Run to simulate a run of the replication task where no files or tables are copied. After the dry run completes, select Show History to view the potential error messages and the number and size of files or tables that would be copied in an actual replication appears on the Replication History page. • Run Now to run the replication task immediately. • Collect Diagnostic Data to open the Send Diagnostic Data screen where you can collect replication-specific diagnostic data for the last 10 runs of the replication policy. In the Send Diagnostic Data screen, select Send Diagnostic Data to Cloudera to automatically send the bundle to Cloudera Support. You can also enter a ticket number and comments when sending the bundle. After you click Collect and Send Diagnostic Data, the Replication Manager generates the bundle and opens the Replications Diagnostics Command screen. When the command finishes, click Download Result Data to download a zip file containing the bundle. • Disable | Enable to disable the replication policy or enable the disabled replication policy. No further replications are scheduled for disabled replication policies. • Delete to remove the replication policy permanently from Replication Manager. Deleting a replication policy does not delete copied files or tables.

When a replication job is in progress, the **Last Run** column shows a spinner and progress bar, and each stage of the replication task is indicated in the message beneath the job's row. Click Command Details to view the command run details. If the job is successful, the number of files copied is indicated. If there have been no changes to a file at the source since the previous job, then that file is not copied. As a result, after the initial job, only a subset of the files may actually be copied, and this is indicated in the success message. Click Actions Show History to view more information about the completed job.

The following sample image shows the **Replication Policies** page in Cloudera Manager:



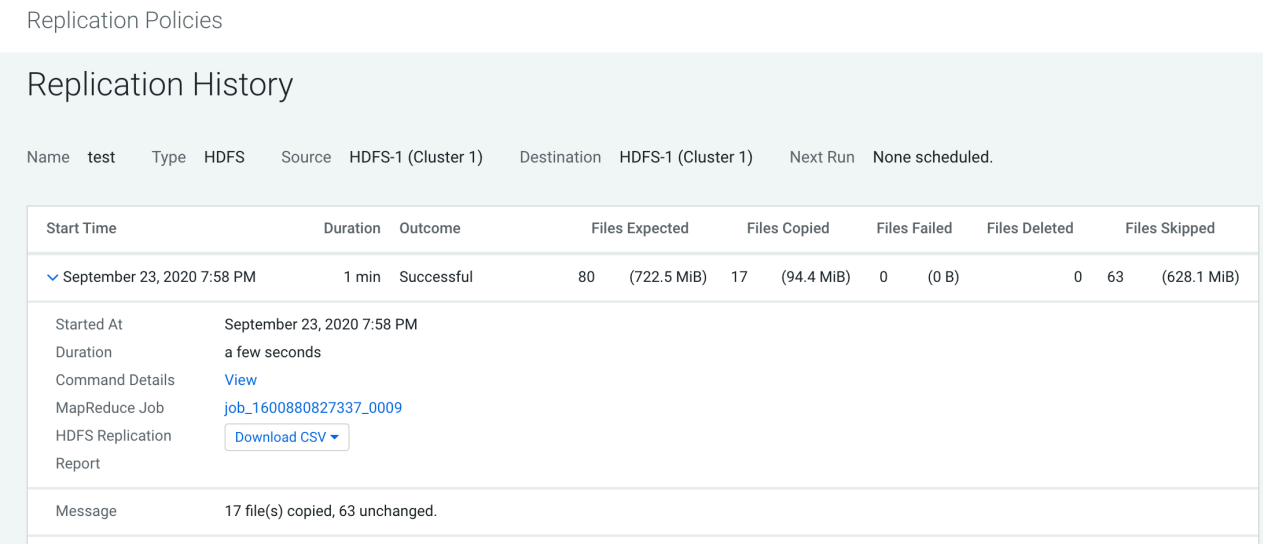
View historical details for an HDFS replication policy

You can view the historical details about the replication jobs on the Replication History page.

The following table lists the columns that appear on the Replication History page when you click Actions Show History to view the previously run replication jobs:

Column	Description
Start Time	Shows the job details. Expand the section to view the following job details: • Started At timestamp is when the replication job started. • Duration to complete the job. • Command Details appear in a new tab after you click View. The Command Details page shows the details and messages about each step during the command run. Click Context to view the service status page relevant to the command, and click Download to download the summary as a JSON file. Expand Step to choose Show All Steps, Show Only Failed Steps, or Show Only Running Steps. You can perform the following tasks in this section: • View the actual command string. • View the start time and duration for the command run. • View the host status page for the command by clicking the host link. • View the full log file for the command by selecting the stdout or stderr tab. For more information, see Viewing Running and Recent Commands. • MapReduce Job details appear after you click the job link. • Download the following HDS Replication Reports in CSV format after you click Download CSV: • Listing report contains the list of files and directories copied during the replication job. • Status report contains the full status report of the files where the replication status is shown as: • **ERROR** occurred during replication, therefore the file was not copied. • **DELETED** for deleted files. • **SKIPPED** for up-to-date files that were not replicated. • Error Status Only report contains the status report of all copied files with errors. The file lists the status, path, and message for the copied files with errors. • Deleted Status Only report contains the status report of all deleted files. The file lists the status, path, and message for the databases and tables that were deleted. • Skipped Status Only report contains the status report of all skipped files. The file lists the status, path, and message for the databases and tables that were skipped. • Performance report contains a summary report about the performance of the running replication job. The report includes the last performance sample for each mapper that is working on the replication job. • Full Performance report contains the performance report of the job. The report shows the samples taken for all the mappers during the full execution of the replication job. • (Dry Run only) Replicable Files shows the number of files that would be replicated during an actual replication. • (Dry Run only) Replicable Bytes shows the number of bytes that would be replicated during an actual replication. • View the number of Impala UDFs replicated. (Displays only for Hive/Impala replications where Replicate Impala Metadata is selected.) • If a user was specified in the Run As Username field when creating the replication job, the selected user appears. • View messages returned from the replication job.
Duration	Time taken for the replication job to complete.
Outcome	Status of the replication job as **Successful** or **Failed**.
Files Expected	Number of files expected to be copied and its file size based on the parameters of the replication policy.
Files Copied	Number of files copied and its file size for the replication job.
Files Failed	Number of files that failed to be copied and its file size for the replication job.
Files Deleted	Number of files that were deleted and its file size for the replication job
Files Skipped	Number of files skipped and its file size for the replication job. The replication process skips files that already exist in the destination and have not changed.

The following sample image shows the historical details about an HDFS replication policy which includes the replication policy name, policy type, source and target cluster details, and the next scheduled run:



Monitoring the performance of HDFS replication policies

You can monitor the progress of an HDFS replication policy using the performance data that you can download as a CSV file from the Cloudera Manager Admin console.

About this task

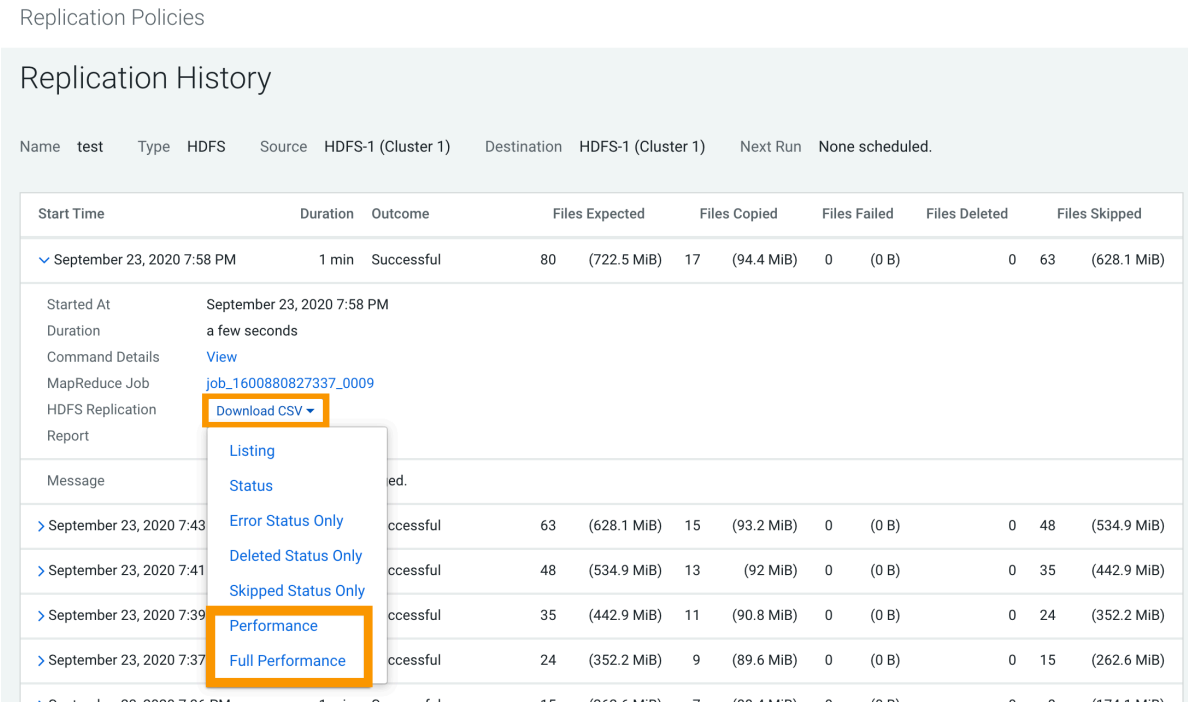
The performance report contains information about the files being replicated, the average throughput, and other details that can help diagnose performance issues during HDFS replications. You can view this performance data for running HDFS replication jobs and for completed jobs. The performance data is collected every two minutes. Therefore, no data is available during the initial execution of a replication job because not enough samples are available to estimate throughput and other reported data.

To view the performance data for a running HDFS replication policy, perform the following steps:

Procedure

1. Go to the Cloudera Manager Replication Replication Policies page.
2. Locate and select the replication policy. Click Actions Show History .

3. Click Download CSV for the HDFS Replication Report field, and choose one of the following options to download the following performance reports:
   - Performance file contains a summary report about the performance of the replication job which includes the last performance sample for each mapper working on the replication job.
   - Full Performance file contains the complete performance report about the job which includes all the samples taken for all mappers during the full run of the replication job.



4. Open the file in a spreadsheet program such as Microsoft Excel.

   The following columns appear in the CSV file:
   - Timestamp when the performance data was collected.
   - Host where the YARN or MapReduce job was running.
   - Number of Bytes Copied for the file currently being copied.
   - Time Elapsed (ms) for the copy operation of the file currently being copied.
   - Number of Files Copied.
   - Avg Throughput (KB/s) since the start of the file currently being copied in kilobytes per second.
   - File size of the Last File (bytes).
   - Time taken to copy Last File Time (ms).
   - Last file throughput (KB/s) that is being copied in kilobytes per second.
5. Download the following CSV reports to view more information about the replication job:
   - Listing report contains the list of files and directories copied during the replication job.
   - Status report contains the full status report of the files where the replication status is shown as:
     - **ERROR** occurred during replication, therefore the file was not copied.
     - **DELETED** for deleted files.
     - **SKIPPED** for up-to-date files that were not replicated.
   - Error Status Only report contains the status report of all copied files with errors. The file lists the status, path, and message for the copied files with errors.
   - Deleted Status Only report contains the status report of all deleted files. The file lists the status, path, and message for the databases and tables that were deleted.
   - Skipped Status Only report contains the status report of all skipped files. The file lists the status, path, and message for the databases and tables that were skipped.

Note the following limitations and known issues about the replication reports:

- If you click the CSV download too soon after the replication job starts, Cloudera Manager returns an empty file or a CSV file that has columns headers only and a message to try later when performance data has actually been collected.
- If you employ a proxy user with the form user@domain, performance data is not available through the links.
- If the replication job only replicates small files that can be transferred in less than a few minutes, no performance statistics are collected.
- If you specify the Dynamic Replication Strategy during replication policy creation, statistics regarding the last file transferred by a MapReduce job hide previous transfers performed by that MapReduce job.
- Only the last trace per MapReduce job is reported in the CSV file.

Hive external table replication policies

Hive external table replication policies enable you to copy (replicate) your Hive metastore and data from one cluster to another and synchronize the Hive metastore and data set on the 'destination' cluster with the source, based on a specified replication policy. You can also use CDP Private Cloud Base Replication Manager to replicate Hive/Impala data to cloud, however you cannot replicate data from one cloud instance to another using Replication Manager.

Note:

- Replication Manager requires a valid license. To understand more about Cloudera license requirements, see Managing Licenses.
- Minimum required role - Replication Administrator or Full Administrator.
- Before you create replication policies, ensure that the source cluster and target cluster are supported by Replication Manager. For information about supported clusters and supported replication scenarios by Replication Manager, see Support matrix for Replication Manager on CDP Private Cloud Base on page 9.

The destination cluster must be managed by the Cloudera Manager Server where the replication is being set up, and the *source* cluster can be managed by that same server or by a peer Cloudera Manager Server.

Limitations and guidelines to consider for Hive external table replication policies

You must consider the following limitations and guidelines before you replicate Hive external tables using Hive external table replication policies.

The following list provides the limitations and guidelines to consider for Hive external table replication policies:

Hive replication policy considerations

Before you create Hive external table replication polices, you must know how to specify the hosts to improve performance; understand how DDL commands affect Hive tables during replication; how to disable parameter replication in Cloudera Manager; and which properties to configure for Hive replication in dynamic environments. For more information, see Hive external table replication policy considerations.

Replicate to S3, ADLS, or GCP

Replication Manager replicates Hive/Impala data from on-premises to cloud; however you cannot replicate data from one cloud instance to another using Replication Manager.

To replicate Hive/Impala data from on-premises to cloud you must have the appropriate credentials to access the cloud account. Additionally, you must create buckets in S3 and GCP or Data Lake store in ADLS. Replication Manager backs up file metadata, including extended attributes and ACLs when you replicate data to cloud storage.

CDP Private Cloud Base Replication Manager supports the following replication scenarios:

- Replicate to Amazon S3 from CDH 5.14+ and Cloudera Manager version 5.13+.

  Replication Manager does not support S3 as a source or destination when S3 is configured to use SSE-KMS.
- Replicate to Microsoft ADLS Gen1 from CDH 5.13+ and Cloudera Manager 5.15, 5.16, 6.1+.
- Replicate to Microsoft ADLS Gen2 (ABFS) from CDH 5.13+ and Cloudera Manager 6.1+.
- Replicate from CDP Private Cloud Base 7.11.3 CHF3 and higher clusters using Dell EMC Isilon storage to CDP Public Cloud clusters on AWS, Azure, and GCP.
- Replicate from CDP Private Cloud Base 7.1.9 SP1 and higher to CDP Public Cloud clusters on GCP.

Replicate from CDH clusters

- Because of the warehouse directory changes between CDH clusters and CDP Private Cloud Base, Hive external table replication policies do not copy the table data from the database and tables specified in the source cluster. But the replication job completes successfully without any disruptions.

  Caution: While replicating from CDH clusters to CDP Private Cloud Base, it is recommended that the HDFS Destination Path is defined. If the HDFS Destination Path is not defined and the Replicate HDFS File is set as true, the data is replicated with the original source name. For example, the replicated table data was to reside under /warehouse/tablespace/external/hive directory but the data was replicated to /user/hive/warehouse location. Also, not defining HDFS Destination Path before the replication process can result in a large chunk of HDFS space being used for unwanted data movement.
- Hive3 has a different default table type and warehouse directory structure, therefore the following changes apply while replicating Hive data from CDH5 or CDH6 versions to CDP Private Cloud Base:
  - When you replicate from a CDH cluster to a CDP Private Cloud Base cluster, all tables become External tables during Hive external table replication. This is because the default table type is ACID in Hive3, which is the only managed table type. As of this release, Replication Manager does not support Hive2 -> Hive3 replication into ACID tables and all the tables will necessarily be replicated as External tables.

    Note: Managed tables are not supported by Replication Manager when you replicate data between CDP Private Cloud Base clusters.
  - Replicated tables are created under the external Hive warehouse directory set by hive.metastore.warehouse.external.dir Hive configuration parameter. You have to make sure that this has a different value than hive.metastore.warehouse.dir Hive configuration parameter, that is the location of Managed tables.
  - If you want to replicate the same database from Hive2 to Hive3 (that has different paths by design), you must use the Force Overwrite option per policy to avoid any mismatch issues.
- Hive external table replication policies do not support managed to managed table replication. When you replicate from a CDH cluster to a CDP Private Cloud Base cluster, Replication Manager converts managed tables to external tables. Therefore, to replicate managed tables (ACID) and external tables in a database successfully, you must perform the following steps in the order shown below:

  1. Create Hive ACID table replication policy for the database to replicate the managed data.
  2. After the replication completes, create the Hive external table replication policy to replicate the external tables in the database.

Migrate Sentry to Ranger

You require source Cloudera Manager version 6.3.1 and higher and target Cloudera Manager version 7.1.1 and higher.

Replicate Atlas metadata

Atlas metadata for the chosen Hive external tables can be replicated using Hive external table replication policies from CDP Private Cloud Base 7.1.9 SP1 or higher using Cloudera Manager 7.11.3 CHF7 or higher.

During the Hive external table replication policy creation process, when you choose the General Replicate Atlas Metadata option, Replication Manager:

1. runs a bootstrap replication for all the chosen Hive external tables and its Atlas metadata during the first replication policy run. Bootstrap replication replicates all the available Hive external tables' data and its associated Atlas metadata.
2. runs incremental replication on the Hive external tables' data and its Atlas metadata during subsequent replication runs. Here, the delta of the data and metadata gets replicated during each run.

Ensure that you have the Atlas user credentials in addition to the Replication Administrator or Full Administrator roles to replicate Atlas metadata. The `atlas` user must also have relevant read and write permissions to the staging locations.

Note: Replicating Atlas metadata using Hive external table replication policies and Iceberg replication policies, and replicating the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities in the source cluster to the target cluster using Atlas replication policies is a technical preview feature. It is not recommended for production deployments.

Cloudera recommends that you try this feature in development or test environments. To enable this feature, contact your Cloudera account team.

Metadata-only replication for Ozone storage-backed Hive external tables

Metadata-only replication for Ozone storage-backed Hive external tables is supported from CDP Private Cloud Base 7.1.9 SP1 or higher using Cloudera Manager 7.11.3 CHF7 or higher. You must replicate the data using Ozone replication policies.

Note: This is a technical preview feature. It is `not` recommended for production deployments. Cloudera recommends that you try this feature in development or test environments. To enable this feature, contact your Cloudera account team.

Replication Manager replicates Ozone-backed Hive external tables when the Destination staging path parameter contains an ofs:// path. By default (when no ofs:// path is used) Ozone backed external tables are ignored. When the ofs:// staging path is provided, only the Ozone backed tables and databases are replicated. An error appears if non-Ozone backed table or database are found in the replication scope.

The destination staging path can be specified on service, volume, or bucket level, and is used to map the replicated database and table locations according to the following table:

Destination staging path	Mapping of source database/table/partition locations. Note: Bucket relative locations are always kept unchanged.
ofs://*[***DST OM SERVICE***]*	Source volume and bucket names are used on the destination. ofs://*[***DST OM SERVICE***]/[***SRC VOLUME***]/[***SRC BUCKET***]/[***SRC PATH***]*
ofs://*[***DST OM SERVICE***]/[***DST VOLUME***]*	Source bucket names are used on the destination, specified destination volume is used, source volume is dropped. Replicated entities (tables, databases) cannot spread across multiple volumes. ofs://*[***DST OM SERVICE***]/[***DST VOLUME***]/[***SRC BUCKET***]/[***SRC PATH***]*

Destination staging path	Mapping of source database/table/partition locations. Note: Bucket relative locations are always kept unchanged.
ofs://*[***DST OM SERVICE***]/[***DST VOLUME***]/[***DST BUCKET***]*	Source volume and bucket names are dropped, specified bucket is used as the destination. Replicated entities cannot spread across multiple buckets. ofs://*[***DST OM SERVICE***]/[***DST VOLUME***]/[***DST BUCKET***]/[***SRC PATH***]*

Configuration-related guidelines

- If the hadoop.proxyuser.hive.groups configuration has been changed to restrict access to the Hive Metastore Server to certain users or groups, the hdfs group or a group containing the hdfs user must also be included in the list of groups specified for Hive/Impala replication to work. This configuration can be specified either on the Hive service as an override, or in the core-site HDFS configuration. This applies to configuration settings on both the source and destination clusters.
- If you configured on the target cluster for the directory where HDFS data is copied during Hive/Impala replication, the permissions that were copied during replication, are overwritten by the HDFS ACL synchronization and are not preserved.

Note: If your deployment includes Iceberg tables or tables backed by Kudu, Replication Manager filters out the Iceberg tables and the Kudu tables during the Hive external table replication process to prevent data loss or corruption.

Hive replication policy considerations

Before you create a Hive replication policy, you must consider when to specify the hosts to improve performance, understand how DDL commands affect Hive tables during replication, how to disable parameter replication in Cloudera Manager, and the additional properties to configure for Hive replication in dynamic environments.

Specifying hosts to improve Hive replication policy performance

When your cluster has Hive clients installed on hosts with limited resources and the Hive/Impala replication policies use these hosts to run commands for the replication, the replication job performance might degrade. To improve the replication job performance, you can specify the hosts to use during replication so that the lower-resource hosts are not used.

Procedure

1. Go to the Cloudera Manager Clusters *Hive service* Configuration tab.
2. Locate the Hive Replication Environment Advanced Configuration Snippet (Safety Valve) property.
3. Add the HOST_WHITELIST property and enter a comma-separated list of hostnames to use for Hive/Impala replication policies.

   For example, HOST_WHITELIST=host-1.mycompany.com,host-2.mycompany.com.
4. Click Save Changes.

Understanding how DDL commands affect Hive tables during replication

Before you create Hive replication policies, you must understand how DDL commands affect the Hive tables during replication.

The following scenarios explain how the tables are affected when you use the drop table and truncate table DDL commands on Hive external tables in a Hive external table replication policy:

- You drop a table in a replication policy after the policy has run at least once. The table remains on the destination cluster and does not get dropped during subsequent replication runs.
- You rename a table on the source cluster. On the target cluster, a table is created with the new name and the old table is also retained.
- You drop a table on the destination cluster and the table is still included in the replication job. The table is re-created on the destination during the next replication job.
- You drop a table partition or index on the source cluster. The next replication job drops it on the destination cluster.
- You truncate a table, and the Advanced Delete Policy field is set to Delete to Trash or Delete Permanently in the Hive external table replication policy wizard. The corresponding data files are deleted on the destination during the next replication job.

The following table shows the resulting actions in the target cluster when you run DDL commands in the source cluster on the Hive external tables used in an Hive external table replication policy:

DDL command on Hive external tables in source cluster	Resulting actions on the table/partition and the folder it resides on the source cluster	Resulting actions on the table/partition and the folder it resides on the target cluster after replication
create table	• Creates table / partition • Creates folder	• Creates table • Creates folder
drop table	• Drops table • Retains the folder	• Retains table • Retains folder
drop table with 'external.table.purge'='true'	• Drops table • Deletes folder	• Retains table • Retains folder
rename table	• Renames table • Retains folder name	• Creates a table with the new name and the original table is retained. • The folder is retained and houses newly created and original tables.
add partition	• Creates partition • Creates folder	• Creates partition • Creates folder
drop partition	• Drops partition • Retains folder	• Drops partition • Retains folder
drop partition with 'external.table.purge'='true'	• Drops partition • Deletes folder	• Drops partition • Retains folder
rename partition	• Renames partition • Retains folder	• Renames partition • Retains folder

Disabling replication of parameters during Hive replication

Parameters of databases, tables, partitions, and indexes are replicated by default during Hive/Impala replications. You can disable the replication of parameters during Hive replication in Cloudera Manager.

Procedure

1. Go to the Cloudera Manager Clusters *Hive Service* Configuration tab.
2. Enter the following parameter for the Hive Replication Environment Advanced Configuration Snippet property:

```
REPLICATE_PARAMETERS=false
```

3. Click Save Changes.
4. Restart the Hive service.

Accommodate HMS changes for Hive replication policies

To use Replication Manager for Hive replication in environments where the Hive Metastore (HMS) changes often, such as when a database or table gets created or deleted, you must configure additional properties to accommodate the changes.

Procedure

1. Go to the Cloudera Manager Clusters *HDFS Service* Configuration tab.
2. Search for the HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml property on the source cluster.
3. Add the following key-value pairs:
   - replication.hive.ignoreDatabaseNotFound and true
   - replication.hive.ignoreTableNotFound and true
4. Click Save Changes.
5. Restart the HDFS service.

Creating a Hive external table replication policy

You must set up your clusters before you create a Hive/Impala replication policy. You can also use CDP Private Cloud Base Replication Manager to replicate Hive/Impala data from on-premises to cloud, however you cannot replicate data from one cloud instance to another using Replication Manager.

Before you begin

Metadata-only replication for Ozone storage-backed Hive external tables is supported from CDP Private Cloud Base 7.1.9 SP1 or higher using Cloudera Manager 7.11.3 CHF7 or higher. You must replicate the data using Ozone replication policies.

Before you create a Hive external table replication policy, you must consider when to specify the hosts to improve performance, understand how DDL commands affect Hive tables during replication, how to disable parameter replication in Cloudera Manager, and the additional properties to configure for Hive replication in dynamic environments. For more information, see *Hive external table replication policy considerations*.

To replicate Hive/Impala data to cloud, you must have the appropriate credentials to access the cloud account. Additionally, you must create buckets in S3 and GCP or Data Lake store in ADLS. Replication Manager backs up file metadata, including extended attributes and ACLs when you replicate data to cloud storage.

Replication Manager functions consistently across HDFS and Hive:

- Replication policies can be set up on files or directories in HDFS and on external tables in Hive—without manual translation of Hive datasets to HDFS datasets, or vice versa. Hive Metastore information is also replicated.
- Applications that depend on external table definitions stored in Hive, operate on both replica and source as table definitions are updated.
- Set the Ranger policy for hdfs user on 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. To provide access, perform the following steps:
  1. Log in to Ranger Admin UI.
  2. Go to the Service Manager Hadoop_SQL Policies Access section, and provide hdfs user permission to the all-database, table, column policy name.
- On the target cluster, the hive user must have Ranger admin privileges. The same hive user performs the metadata import operation.

Tip: 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.

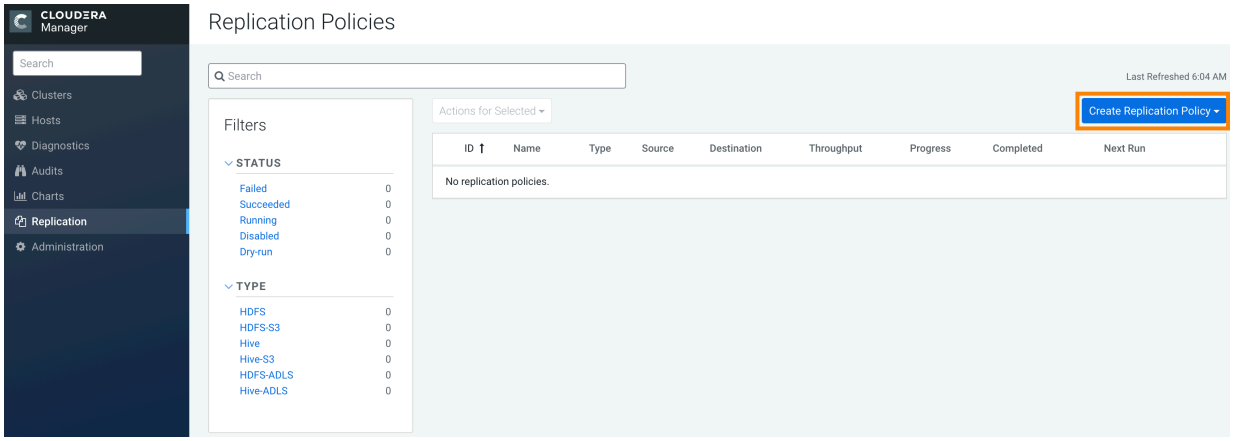
Procedure

1. If the source cluster is managed by a different Cloudera Manager server than the destination cluster, configure a peer relationship.
2. Add the required credentials in Cloudera Manager to access the cloud storage to replicate Hive/Impala data to cloud storage. You can enter the s3a://*[***BUCKET NAME***]*/*[***PATH***]* path to replicate to Amazon S3 and adl://*[***ACCOUNT NAME***]*.azuredatalakestore.net/*[***PATH***]* path to replicate to ADLS.
   a) To add AWS credentials, see How to Configure AWS Credentials.

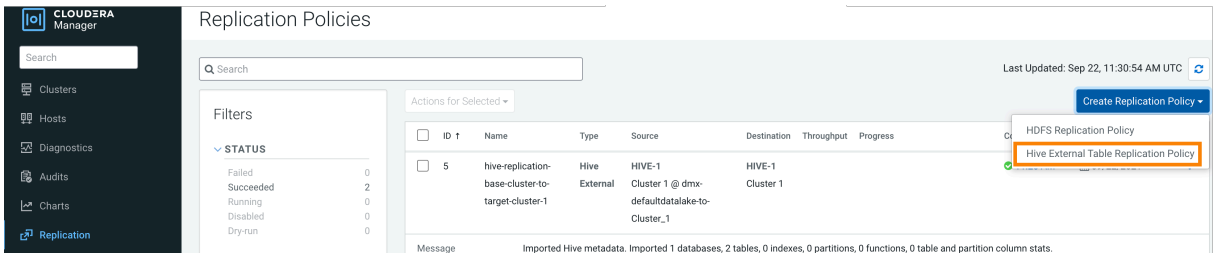
      Ensure that the following basic permissions are available to provide read-write access to S3 through the S3A connector:

      ```
      s3:Get*
      s3:Delete*
      s3:Put*
      s3:ListBucket
      s3:ListBucketMultipartUploads
      s3:AbortMultipartUpload
      ```

   b) To add ADLS credentials, perform the following steps:
      1. Click Add AD Service Principal on the Cloudera Manager Admin Console Administration External Accounts Azure Credentials page for the source cluster.
      2. Enter the Name, Client ID, Client Secret Key, and Tenant Identity for the credential in the **Add AD Service Principal** modal window.
      3. Click Add.
3. Click Create Replication Policy on the Cloudera Manager Replication Replication Policies page.



4. Select Hive External Table Replication Policy.



5. Configure the following options on the General tab:

Option	Description
Name	Enter a unique name for the replication policy.
Source	Select the cluster with the Hive service you want to replicate.
Destination	Select the destination for the replication. If there is only one Hive service managed by Cloudera Manager available as a destination, this is specified as the destination. If more than one Hive service is managed by this Cloudera Manager, select from among them.
Destination Staging Path	Enter one of the following: • A valid HDFS path without the external table base directory to store the Hive data and metadata, a root for creating table directories. Replication Manager uses this path to create the table directory on the target cluster. For example, if the Destination Staging Path is /mypath/ and the table location on the source cluster is /user/hive/warehouse/bdr.db/tab1. Enter /mypath in the field. After replication, the table location on the target cluster is /mypath/user/hive/warehouse/bdr.db/tab1. • To replicate metadata of Ozone backed external tables, add the ofs:// path. Enter the Ozone service and volume or bucket level path in one of the following formats depending on your requirements: • ofs://*[***DST OM SERVICE***]* • ofs://*[***DST OM SERVICE***]/[***DST VOLUME***]* • ofs://*[***DST OM SERVICE***]/[***DST VOLUME***]/[***DST BUCKET***]* For information about the path mapping, see Metadata-only replication for Ozone storage-backed Hive external tables.
Permissions	Select one of the following permissions: • Do not import Sentry Permissions (Default) • If Sentry permissions were exported from the CDH cluster, import both Hive object and URL permissions • If Sentry permissions were exported from the CDH cluster, import only Hive object permissions

Option	Description

Databases	Select Replicate All to replicate all the Hive databases from the source, or enter the database names and table names.
	
	To replicate only selected databases, clear the option and enter the database name(s) and tables you want to replicate.
	
	• Specify multiple databases and tables using the plus symbol to add more rows to the specification.
	• Specify multiple databases on a single line by separating their names with the pipe (|) character. For example: mydbname1|mydbname2|mydbname3.
	• Use regular expressions in the database or table fields as shown in the following examples:
	  • To specify any database or table name, enter the following regular expression:
	  `[\w].+`
	  • To specify any database or table except the one named 'myname', enter the following regular expression:
	  `(?!myname\b).+`
	  • To specify all the tables in the db1 and db2 databases, enter the following regular expression:
	  `db1|db2`
	  `[\w_]+`
	  • To specify all the tables of the db1 and db2 databases (alternate method), enter the following regular expression:
	  `db1`
	  `[\w_]+`
	  Click + icon and enter the following expression:
	  `db2`
	  `[\w_]+`

Schedule	Choose: • Immediate to run the schedule immediately. • Once to run the schedule one time in the future. Set the date and time. • Recurring to run the schedule periodically in the future. Set the date, time, and interval between runs. Replication Manager ensures that the same number of seconds elapse between the runs. For example, if you choose the Start Time as January 19, 2022 11.06 AM and Interval as 1 day, Replication Manager runs the replication policy for the first time at the specified time in the timezone the replication policy was created in, and then runs it exactly after 1 day that is, after 24 hours or 86400 seconds.
Run As Username	Enter the username to run the MapReduce job. By default, MapReduce jobs run as hdfs. To run the MapReduce job as a different user, enter the user name. If you are using Kerberos, you must provide a user name here, and it must have an ID greater than 1000. **Note:** The user running the MapReduce job should have read and execute permissions on the Hive warehouse directory on the source cluster. If you configure the replication job to preserve permissions, superuser privileges are required on the destination cluster.

Option	Description
Run on peer as Username	Enter the username if the peer cluster is configured with a different superuser. This is applicable in a kerberized environment.
Replicate Atlas Metadata	Choose to copy the Atlas metadata associated with the chosen Hive external tables. For more information, see Replicate Atlas metadata.

6. Configure the following options on the **Sentry-Ranger Migration** tab:

Option	Description
Sentry export authorization-migration-site.xml extra properties	Enter one or more additional arguments to either add a new property or to override an existing property in the authorization-migration-site.xml file. The authzmigrator tool uses these new/modified properties during the Sentry export process on the source cluster.
Ranger import authorization-migration-site.xml extra properties	Enter one or more additional arguments to either add a new property or to override an existing property in the authorization-migration-site.xml file. The authzmigrator tool uses these new/modified properties during the Ranger import process on the target cluster.

The **Sentry-Ranger Migration** tab appears after you choose the If Sentry permissions were exported from the CDH cluster, import both Hive object and URL permissions or If Sentry permissions were exported from the CDH cluster, import only Hive object permissions option in the General Permissions field.

The **Sentry-Ranger Migration** tab is available in Cloudera Manager version 7.7.1 CHF18 and higher versions and in 7.11.3 CHF5 and higher versions. For more information about the migration of Sentry policies to Ranger policies, see *Migrate Sentry to Ranger using Hive external tables replication policies*.

7. Configure the following options on the Resources tab:

Option	Description
Scheduler Pool	(Optional) Enter the name of a resource pool in the field. The value you enter is used by the MapReduce Service you specified when Cloudera Manager executes the MapReduce job for the replication. The job specifies the value using one of these properties: • MapReduce – Fair scheduler: mapred.fairscheduler.pool • MapReduce – Capacity scheduler: queue.name • YARN – mapreduce.job.queuename
Maximum Map Slots	Enter the number of map tasks that the DistCp MapReduce job can use for the replication policy. Default is 20.
Maximum Bandwidth	Enter the bandwidth limit for each mapper. Default is 100 MB. The total bandwidth used by the replication policy is equal to Maximum Bandwidth multiplied by Maximum Map Slots. Therefore, you must ensure that the bandwidth and map slots you choose do not impact other tasks or network resources in the target cluster. **Tip:** The Throughput field on the Cloudera Manager Replication Policies page shows the maximum bandwidth set for the replication policy during replication policy creation.
Replication Strategy	Choose Static or Dynamic to determine whether the file replication tasks must be distributed among the mappers statically or dynamically. The default is Dynamic Static replication distributes file replication tasks among the mappers up front to achieve a uniform distribution based on the file sizes. Dynamic replication distributes file replication tasks in small sets to the mappers, and as each mapper completes its tasks, it dynamically acquires and processes the next unallocated set of tasks.

8. Configure the following options on the Advanced tab where you can specify an export location, modify the parameters of the MapReduce job that performs the replication, and select a MapReduce service (if there is more than one in your cluster):

Option	Description
Replicate HDFS Files	Clear the option to skip replicating the associated data files.
Force Overwrite	Select the option to overwrite data in the destination metastore if incompatible changes are detected. For example, if the destination metastore was modified, and a new partition was added to a table, this option forces deletion of that partition, overwriting the table with the version found on the source. **Important:** If the Force Overwrite option is not selected, and the Hive/Impala replication process detects incompatible changes on the source cluster, Hive/Impala replication fails. This sometimes occurs with recurring replications, where the metadata associated with an existing database or table on the source cluster changes over time.
Directory for metadata file	Enter / or a valid folder path in the target cluster to save the metadata file. If the field is empty or if the specified folder does not exist, Replication Manager creates a new folder. For example, the /.cm/hive-staging/ directory containing the Hive metadata is stored in the specified target HDFS path during replication, before the metadata is imported into the metastore service. If the field is empty, the /.cm/hive-staging/ directory is generated in the /user/$*[***PROXY USER***]* location on target cluster where the proxyuser is hdfs.
Number of concurrent HMS connections	Enter the number of concurrent Hive Metastore connections. The connections are used to concurrently import and export metadata from Hive. Increase the number of threads to improve Replication Manager performance. By default, a new replication policy uses 4 connections. **a.** If you set the value to 1 or more, Replication Manager uses multi-threading with the number of connections specified. **b.** If you set the value to 0 or fewer, Replication Manager uses single threading and a single connection. Note that the source and destination clusters must run a Cloudera Manager version that supports concurrent HMS connections, Cloudera Manager 5.15.0 or higher and Cloudera Manager 6.1.0 or higher.
MapReduce Service	Select the MapReduce or YARN service to use.
Log path	Enter an alternate path for the logs.
Description	Enter a description of the replication policy.

Option	Description
Error Handling	Select: • Skip Checksum Checks to determine whether to skip checksum checks on the copied files. If selected, checksums are not validated. Checksums are checked by default. **Important:** You must skip checksum checks to prevent replication failure due to non-matching checksums in the following cases: • Replications from an encrypted zone on the source cluster to an encrypted zone on a destination cluster. • Replications from an encryption zone on the source cluster to an unencrypted zone on the destination cluster. • Replications from an unencrypted zone on the source cluster to an encrypted zone on the destination cluster. Checksums are used for two purposes: • To skip replication of files that have already been copied. If Skip Checksum Checks is selected, the replication job skips copying a file if the file lengths and modification times are identical between the source and destination clusters. Otherwise, the job copies the file from the source to the destination. • To redundantly verify the integrity of data. However, checksums are not required to guarantee accurate transfers between clusters. HDFS data transfers are protected by checksums during transfer and storage hardware also uses checksums to ensure that data is accurately stored. These two mechanisms work together to validate the integrity of the copied data. • Skip Listing Checksum Checks to determine whether to skip checksum check when comparing two files to determine whether they are same or not. If skipped, the file size and last modified time are used to determine if files are the same or not. Skipping the check improves performance during the mapper phase. Note that if you select the Skip Checksum Checks option, this check is also skipped. • Abort on Error to determine whether to abort the job on an error. If selected, files copied up to that point remain on the destination, but no additional files are copied. Abort on Error is not selected by default. • Abort on Snapshot Diff Failures if you want Replication Manager to use a complete copy to replicate data when snapshot diff fails during replication. If you select this option, the Replication Manager aborts the replication when it encounters an error instead.
Preserve	Determines whether to preserve the Block Size, Replication Count, and Permissions as they exist on the source file system, or to use the settings as configured on the destination file system. By default, settings are preserved on the source. **Note:** You must be running as a superuser to preserve permissions. Use the Run As Username option to ensure that is the case.

Option	Description
Delete Policy	Determines whether files that were deleted on the source should also be deleted from the destination directory. This policy also determines the handling of files in the destination location that are unrelated to the source. Choose: • Keep Deleted Files to retain the destination files even when they no longer exist at the source. This is the default. • Delete to Trash if the HDFS trash is enabled. • Delete Permanently to use the least amount of space; use with caution. This option does not delete the files and directories in the top level directory. This is in line with rsync/Hadoop DistCp behavior.
Alerts	Choose to generate alerts for various state changes in the replication workflow. You can choose to generate an alert On Failure, On Start, On Success, or On Abort of the replication job. You can configure alerts to be delivered by email or sent as SNMP traps. If alerts are enabled for events, you can search for and view the alerts on the **Events** tab, even if you do not have email notification configured. For example, if you choose Command Result that contains the Failed filter on the Diagnostics Events page, the alerts related to the On Failure alert for all the replication policies for which you have set the alert appear. For more information, see Managing Alerts and Configuring Alert Delivery.

9. Click Save Policy.

- If your replication job takes a long time to complete, see Improve network latency during replication job run to improve network latency.
- If files change before the replication finishes, the replication might fail. For more information, see Guidelines to add or delete source data during replication job run.
- For efficient replication, consider making the Hive Warehouse Directory and the directories of any external tables snapshottable, so that the replication job creates snapshots of the directories before copying the files. For more information, see Hive/Impala replication using snapshots and Guidelines to use snapshot diff-based replication.
- If your cluster has Hive clients installed on hosts with limited resources and the Hive/Impala replication policies use these hosts to run commands for the replication, the replication job performance might degrade. To specify the hosts to use during replication so that the lower-resource hosts are not used to improve the replication job performance, see Specifying hosts to improve Hive replication policy performance.

Sentry to Ranger replication using Hive external tables

When you create or edit a Hive external table replication policy in CDP Private Cloud Base Replication Manager, you can choose to migrate the Sentry policies for Hive objects, Impala data, and URLs that are being replicated. Replication Manager converts the Sentry policies to Ranger policies for the migrated data in the target cluster.

To migrate Sentry policies to Ranger policies using Hive external table replication policies, you must have installed Cloudera Manager version 6.3.1 and higher on the source cluster and Cloudera Manager version 7.1.1 and higher on the target cluster.

Note: You must consider the following points before you initiate the Sentry to Ranger replication using Hive external tables replication policies.

- When you replicate a subset of the tables in a database, the database-level policies are converted to equivalent table-level policies for each table being replicated. For example, ALL on database is converted to ALL on table individually for each table replicated.
- There will be no reference to the original role names in Ranger. The permissions are granted directly to the groups and users with respect to the resource and not the role. This is a different format to the Sentry to Ranger migration during an in-place upgrade to CDP Private Cloud Base, which does import and use the Sentry roles.
- Regardless of whether a policy is modified or not, each policy is re-created during each replication. If you want to continue data replication and you also want to modify the target cluster's Ranger policies (and keep those modifications), you must disable the Sentry to Ranger migration on subsequent replication policy runs. To accomplish this task, edit the required Hive external table replication policy and choose the Do not import Sentry Permissions (Default) option in the General Permissions field.

Replication Manager performs the following tasks automatically during the replication job run to migrate Sentry policies in the source cluster to Ranger policies in the target cluster:

1. Exports each Sentry policy as a single JSON file using the authzmigrator tool. The JSON file contains a list of resources, such as URI, database, table, or column and the policies that apply to it.
2. Copies the exported Sentry policies to the target cluster using the DistCp tool.
3. Ingests the Sentry policies into Ranger after filtering the policies related to the replication job using the authzmigrator tool through the Ranger REST API endpoint. To filter the policies, the Replication Manager uses a filter expression that is passed to the authzmigrator tool by Cloudera Manager.

Tip: During Hive external tables' replication, the authzmigrator tool uses the properties in the authorization-migration-site.xml file to export Sentry policies from the source cluster and import Ranger policies into the target cluster.

Modify properties on Sentry-Ranger Migration tab

Starting from Cloudera Manager version 7.7.1 CHF18 and higher and 7.11.3 CHF5 and higher, you can modify the properties in the authorization-migration-site.xml file during the Hive external table replication creation or edit process on the **Sentry-Ranger Migration** tab.

The **Sentry-Ranger Migration** tab appears in the Hive external table replication policy wizard after you choose the If Sentry permissions were exported from the CDH cluster, import both Hive object and URL permissions or If Sentry permissions were exported from the CDH cluster, import only Hive object permissions option in the General Permissions field.

You can add one or more key-value arguments to either add a new property or override an existing property in the authorization-migration-site.xml file. You can add:

- key-value pairs for the properties to use during the Sentry export process on the source cluster in the Sentry export authorization-migration-site.xml extra properties field.
- key-value pairs for the properties to use during the Ranger import process on the target cluster in the Ranger import authorization-migration-site.xml extra properties field.

For example, if you want to use the URL prefix as specified in the authorization.migration.destination.location.prefix parameter in the authorization-migration-site.xml file, skip the Sentry policies with *Owner* privileges from the migration process, and also inform Replication Manager that the Sentry and Ranger policies have role-based permissions, you must perform the following steps:

1. Create or edit a Hive external table replication policy on the Cloudera Manager Replication Replication Policies page.
2. Choose the If Sentry permissions were exported from the CDH cluster, import both Hive object and URL permissions or If Sentry permissions were exported from the CDH cluster, import only Hive object permissions option in the General Permissions field.

3. Enter the following key-value pairs in the Sentry-Ranger Migration Sentry export authorization-migration-site.xml extra properties field:
   - authorization.migration.role.permissions = true

     When set to true, the parameter informs Replication Manager that the Sentry policies use *roleBasedPermissions* and that it must use the same during the Sentry export process.
   - authorization.migration.skip.owner.policy = true

     When set to true, Replication Manager skips the Sentry policies with *Owner* privileges during migration.
4. Enter the following key-value pairs in the Sentry-Ranger Migration Ranger import authorization-migration-site.xml extra properties field:
   - authorization.migration.destination.location.prefix = *[***DESTINATION LOCATION PREFIX***]*

     Enter the required destination location prefix depending on your requirements. For example, if you are migrating Sentry policies from a CDH source cluster to a target CDP Private Cloud Base cluster, the prefix must match the CDP cluster's namespace. In this instance, if the rootPath parameter is hdfs://*[***CDP NAMESERVICE***]*, then you must enter authorization.migration.destination.location.prefix=hdfs://*[***CDP NAMESERVICE***]*
   - authorization.migration.url.ignore.scheme = (*[***ENTER COMMA-SEPARATED PREFIXES TO USE DURING MIGRATION. FOR EXAMPLE, s3, file***]*)

     The authorization.migration.url.ignore.scheme property is dependent on two other properties, that is authorization.migration.translate.url.privileges and authorization.migration.destination.location.prefix in the authorization-migration-site.xml file.

     If the authorization-migration-site.xml file contains authorization.migration.translate.url.privileges = true, authorization.migration.destination.location.prefix = hdfs://ns1, and the authorization.migration.url.ignore.scheme property is not set, all the URL policies' prefixes are replaced with hdfs://ns1 after the import process is complete. However, if a file:///opt/somevalue URL is available, then the URL becomes hdfs://ns1/opt/somevalue after the import process.

     If you set the config authorization.migration.url.ignore.scheme = s3,file parameter in the **Sentry-Ranger Migration** tab, then the above URL is skipped from updating as its prefix starts with file. Therefore, the URL file:///opt/somevalue remains as is after the import process.
   - authorization.migration.role.permissions = true

     When set to true, the parameter informs Replication Manager that the Ranger policies must use *roleBasedPermissions* and to use role-based permissions during the Sentry import process.

Importing Sentry privileges into Ranger policies

How to complete the process of translating Sentry privileges into Ranger policies.

About this task

No one-to-one mapping between Sentry privileges and Ranger service policies exists. Upgrading your platform involves translating Sentry privileges to their equivalents within Ranger service policies. After upgrading Cloudera Manager and your cluster, this post-upgrade step completes the translation process.

Procedure

1. In Ranger Actions , click Import Sentry Policies.

2. Read the following points that describe how 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 OWNER privilege.
   - Sentry OWNER WITH GRANT OPTION privileges are translated to Ranger OWNER 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 use the object store location as the base location.
   - If your cluster contains the Kafka service and the Kafka sentry policy had "action": "ALL" permission, the migrated Ranger policy for "cluster" resource will be missing the "alter" permission. This is only applicable for "cluster" resource. You need to add the policy manually after the upgrade. This missing permission does not have any functional impact. Adding the "alter" permission post upgrade is needed only for completeness because the 'configure' permission allow alter operations.
   - Sentry "alter" permission on cluster and topic is translated to "configure" in Ranger.

   The following table shows how actions in Sentry translate to corresponding actions in Ranger:

Table 4: Sentry Actions to Ranger Actions

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

Replicating data to Impala clusters

Impala metadata is replicated as part of regular Hive/Impala replication operations. Impala metadata replication is performed as a part of Hive external table replication. Impala replication is only supported between two CDH clusters. The Impala and Hive services must be running on both clusters.

Replicating Impala Metadata

To enable Impala metadata replication, set the Advanced Replicate Impala Metadata field to Yes during Hive external table replication policy creation. After the replication job completes, you can view the Impala UDFs (user-defined functions) on the target cluster, just as on the source cluster. As part of replicating the UDFs, the binaries in which they are defined are also replicated.

Note: To run queries or DDL statements on tables that have been replicated to a destination cluster, you must run the Impala INVALIDATE METADATA statement on the destination cluster to prevent queries from failing.

Invalidating Impala Metadata

For Impala clusters that do not use LDAP authentication, configure Advanced Invalidate Impala Metadata on Destination during Hive external table replication policy creation so that the replication job automatically invalidates Impala metadata after replication completes. If the clusters use Sentry, the Impala user should have permissions to run INVALIDATE METADATA.

The configuration causes the Hive/Impala replication job to run the Impala INVALIDATE METADATA statement per table on the destination cluster after completing the replication. The statement purges the metadata of the replicated tables and views within the destination cluster's Impala upon completion of replication, allowing other Impala clients at the destination to query these tables successfully with accurate results. However, this operation is potentially unsafe if DDL operations are being performed on any of the replicated tables or views while the replication is running. In general, directly modifying replicated data/metadata on the destination is not recommended. Ignoring this can lead to unexpected or incorrect behavior of applications and queries using these tables or views.

Note: If the source contains UDFs, you must run the INVALIDATE METADATA statement manually and without any tables specified even if you configure the automatic invalidation.

Alternatively, you can run the INVALIDATE METADATA statement manually for replicated tables.

Replication of Impala and Hive User Defined Functions (UDFs)

By default, for clusters where the version of CDH is 5.7 or higher, Impala and Hive UDFs are persisted in the Hive Metastore and are replicated automatically as part of Hive/Impala replication.

After a replication job is complete, you can see the number of Impala and Hive UDFs that were replicated during the last run of the schedule on the Replication Policies page. You can also view the number of replicated UDFs on the Replication History page for previously-run replications.

Monitoring the performance of Hive/Impala replication policies

You can monitor the progress of a Hive/Impala replication policy using performance data that you download as a CSV file from Replication Manager.

Before you begin

This file contains information about the tables and partitions being replicated, the average throughput, and other details that can help diagnose performance issues during Hive/Impala replications. You can view this performance data for running Hive/Impala replication jobs and for completed jobs. The performance data is collected every two minutes. Therefore, no data is available during the initial execution of a replication job because not enough samples are available to estimate throughput and other reported data.

Procedure

1. To view the performance data for a running Hive/Impala replication policy, perform the following steps:
   a) Go to the Cloudera Manager Replication Replication Policies page.
   b) Locate and select the replication policy. Click Actions Show History .
   c) Click Download CSV for the HDFS Replication Report field, and choose one of the following options to download the following performance reports:
      - Performance file contains a summary report about the performance of the replication job which includes the last performance sample for each mapper working on the replication job.
      - Full Performance file contains the complete performance report about the job which includes all the samples taken for all mappers during the full run of the replication job.
   d) Open the file in a spreadsheet program such as Microsoft Excel.

      The following columns appear in the CSV file:
      - Timestamp when the performance data was collected.
      - Host where the YARN or MapReduce job was running.
      - Number of Bytes Copied for the file currently being copied.
      - Time Elapsed (ms) for the copy operation of the file currently being copied.
      - Number of Files Copied.
      - Avg Throughput (KB/s) since the start of the file currently being copied in kilobytes per second.
      - File size of the Last File (bytes).
      - Time taken to copy Last File Time (ms).
      - Last file throughput (KB/s) that is being copied in kilobytes per second.
   e) Download the following CSV reports to view more information about the replication job:
      - Listing report contains the list of files and directories copied during the replication job.
      - Status report contains the full status report of the files where the replication status is shown as:
        - **ERROR** occurred during replication, therefore the file was not copied.
        - **DELETED** for deleted files.
        - **SKIPPED** for up-to-date files that were not replicated.
      - Error Status Only report contains the status report of all copied files with errors. The file lists the status, path, and message for the copied files with errors.
      - Deleted Status Only report contains the status report of all deleted files. The file lists the status, path, and message for the databases and tables that were deleted.
      - Skipped Status Only report contains the status report of all skipped files. The file lists the status, path, and message for the databases and tables that were skipped.

2. To view the performance data for a completed Hive/Impala replication policy, perform the following steps:
   a) Go to the Cloudera Manager Replication Replication Policies page.
   b) Locate and select the replication policy. Click Actions Show History .
   c) Click Download CSV for the Hive External Table Replication Report field, and choose one of the following options to download the following performance reports in CSV format:
      - Results file contains a listing of replicated tables.
      - Performance file contains a summary report about the performance of the replication job.

      Note: The option to download the HDFS replication reports might not appear if the HDFS phase of the replication skipped all the HDFS files because they have not changed, or if the Advanced Replicate HDFS Files option is not selected during Hive/Impala replication policy creation.
   d) Open the file in a spreadsheet program such as Microsoft Excel.

      The following columns appear in the CSV file:
      - Timestamp when the performance data was collected.
      - Host where the YARN or MapReduce job was running.
      - DbName or database name.
      - TableName or table name.
      - TotalElapsedTimeSecs is the number of seconds elapsed from the start of the copy operation.
      - TotalTableCount is the total number of tables to be copied. The value of the column shows -1 for replications where Cloudera Manager cannot determine the number of tables being changed.
      - TotalPartitionCount is the total number of partitions to be copied. If the source cluster is running Cloudera Manager 5.9 or lower, this column shows -1 because older releases do not report this information.
      - DbCount is the current number of databases copied.
      - DbErrorCount is the number of failed database copy operations.
      - TableCount is the total number of tables for all databases copied so far.
      - CurrentTableCount is the total number of tables copied for the current database.
      - TableErrorCount is the total number of failed table copy operations.
      - PartitionCount is the total number of partitions copied so far for all tables.
      - CurrPartitionCount is the total number of partitions copied for the current table.
      - PartitionSkippedCount is the number of partitions skipped because they were copied in the previous run of the replication job.
      - IndexCount is the total number of index files copied for all databases.
      - CurrIndexCount is the total number of index files copied for the current database.
      - IndexSkippedCount is the number of index files skipped because they were not altered. Due to a bug in Hive, this value is always zero.
      - HiveFunctionCount is the number of Hive functions copied.
      - ImpalaObjectCount is the number of Impala objects copied.

Note the following limitations and known issues about the replication reports:

- If you click the CSV download too soon after the replication job starts, Cloudera Manager returns an empty file or a CSV file that has columns headers only and a message to try later when performance data has actually been collected.
- If you employ a proxy user with the form user@domain, performance data is not available through the links.
- If the replication job only replicates small files that can be transferred in less than a few minutes, no performance statistics are collected.
- If you specify the Dynamic Replication Strategy during replication policy creation, statistics regarding the last file transferred by a MapReduce job hide previous transfers performed by that MapReduce job.
- Only the last trace per MapReduce job is reported in the CSV file.

Hive ACID table replication policies

You can create the Hive ACID table replication policies in CDP Private Cloud Base Replication Manager to copy ACID tables between CDP Private Cloud Base clusters for backup, load balancing, and other purposes.

Hive ACID table replication policies can:

- replicate ACID tables.
- perform incremental replication based on metastore events.

Important: To replicate managed tables (ACID) and external tables in a database successfully, you must perform the following steps in the order shown below:

1. Create Hive ACID table replication policy for the database to replicate the managed data.
2. After the replication completes, create the Hive external table replication policy to replicate the external tables in the database.

Note: Do not drop a database that is under replication in source cluster or target cluster.

Hive ACID table replication policies cannot replicate data:

- between cloud-based clusters.
- in external tables.
- within the same cluster.

Preparing to create Hive ACID table replication policies

Before you create a Hive ACID table replication policy, you must prepare the clusters for replication.

Before you begin

- Replication Manager requires a valid license. To understand more about Cloudera license requirements, see Managing Licenses.
- Minimum required role - Replication Administrator or Full Administrator.
- Before you create replication policies, ensure that the source cluster and target cluster are supported by Replication Manager. For information about supported clusters and supported replication scenarios by Replication Manager, see Support matrix for Replication Manager on CDP Private Cloud Base on page 9.

To perform Hive ACID table replication using Replication Manager, Cloudera Manager Server must manage the target cluster. You can use the same server or a peer Cloudera Manager Server to manage the source cluster. Hive ACID table replication policies use Hive scheduler to schedule the frequency of replication policy job runs.

Procedure

1. Set up a two-way trust between the CDP Private Cloud Base clusters. For more information, see Configure two-way trust between clusters on page 72
2. Configure a peer relationship only if the source cluster is managed by a different Cloudera Manager server than the target cluster. For more information, see Configuring a peer relationship.
3. Configure the hive.repl.cm.enabled=true key-value pair on the source cluster for the following services to turn on the ChangeManager:

Service	Action
Hive-on-Tez For example, *Hive-on-Tez-1*	On the Configuration tab, search for Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml property and set the key-value pair

Service	Action
Hive For example, *Hive-1*	On the Configuration tab, search for Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml property and set the key-value pair.
Hive For example, *Hive-1*	On the Configuration tab, search for Enable ChangeManager for Hive replication parameter and select it.

Important: Restart the Hive and Hive-on-Tez service after you configure the key-value pair.

4. Configure Hive configuration parameters for Hive ACID tables. For more information, see Advanced Hive configuration parameters for Hive ACID table replication policies on page 75.

   Optionally, to optimize the replication policy performance, you can configure the parameters in Recommended Hive configuration parameters for Hive ACID table replication policies on page 74 and Parameters to optimize Hive ACID table replication performance on page 75 as necessary.
5. Enable the Hive ACID table replication feature flag on the source and target cluster.

   For more information, contact your Cloudera account team.
6. Complete the following steps if LDAP authorization is enabled:
   a) Go to the Cloudera Manager Clusters *HIVE-ON-TEZ SERVICE* Configuration tab.
   b) Choose Enable LDAP Authentication for HiveServer2.
   c) Enter the LDAP URL in the ldap[s]://*[***HOST***]*:*[***PORT***]* format.
   d) Enter the base LDAP distinguished name (DN) for the LDAP server in LDAP BaseDN. For example, ou=dev, dc=xyz.
   e) Restart the service.
   f) Enter the LDAP username in the config.ldap.repl.user.display_name (or hiveserver2_ldap_replication_user) property.
   g) Enter the LDAP password in the config-ldap-repl.password.display_name (or hiveserver2_ldap_replication_password) property.
   h) Save the configuration.

Configure two-way trust between clusters

A two-way trust between the source cluster and target cluster is required when both the clusters use different Kerberos KDC servers with the same realm or different realms. The staging directory is on the target cluster. It allows the source cluster to access staging on the target cluster for both the DistCp and YARN jobs after you configure the two-way trust between the clusters. The administrator must set up a one-way trust in order to use replication between two kerberized clusters. You can also set up a one-way trust when the staging directory is on the source cluster. Optionally, a two-way trust can be configured.

Clusters using different Kerberos KDC Servers with same realm

When the clusters use different Kerberos KDC servers with the same realm, you must point both the clusters to a single Kerberos KDC server and regenerate the keytabs of the migrated cluster in Cloudera Manager.

To point the clusters to a single Kerberos KDC server, perform the following steps:

1. Create a source cluster and a target cluster that belong to the same realm.

   For example, assume that the realm name is EXAMPLE.COM.

   Note: In this example, EXAMPLE.COM points to the KDC server on the source cluster. It can point to the target cluster as well.
2. Set up the /etc/krb5.conf file on all the hosts of both the source and target clusters.

3. Perform the following steps *only* on the target cluster:
   - [realms] section - In the target cluster, copy EXAMPLE.COM from the source cluster's KDC, admin_server, and default_domain settings.
   - [domain_realm] section - Enlist all the hosts of both source and target clusters.

To regenerate the keytabs of the migrated cluster, perform the following steps:

1. Log into Cloudera Manager with administrator privileges.
2. Stop all the services including the Cloudera Management Service.
3. Go to the Administration Security Kerberos credentials page.
4. Click Setup KDC for this Cloudera Manager option.
5. In the **Setup KDC for this Cloudera Manager** wizard, choose the following options:
   a. On the **Getting Started** page, select MIT KDC. Select I have completed all the above steps after you make sure that all the steps in this page are complete.
   b. Click Continue.
   c. On the Enter KDC Information page, update the KDC Server Host information as per the source cluster configuration.
   d. Click Continue.
   e. Enter the required details in all pages of the wizard to complete the setup.
6. Go to the Administration Security Kerberos Credentials page.
7. Select all the listed Principal values, and click Regenerate Selected.
8. Restart the Cloudera Management Service and the clusters.

Clusters using different Kerberos KDC Servers with different realms

When the CDP Private Cloud Base source cluster and target cluster use different Kerberos KDC servers with different realms, you must set up a two-way KDC trust between the clusters.

Hive ACID table replication policies use a common staging location on the source or target cluster. To set the staging location path, use the hive.repl.rootdir configuration parameter to configure the HDFS root directory for all replication dumps in the source cluster. The REPL DUMP command dumps data into the staging location and the REPL LOAD command reads the data from the staging location. The REPL DUMP command runs in the source cluster and the REPL LOAD command runs in the target cluster.

When the staging location is on the target cluster, the source cluster hosts access the target HDFS staging location. The target KDC trusts the connections from the source using trusted keytabs. Similarly, if the staging location is on the source cluster, the target cluster hosts access the source HDFS staging location.

To set up two-way trust between the CDP Private Cloud Base source and target cluster, perform the following steps:

1. Create clusters that belong to different Kerberos realms.

   For example, assume that you have Realm: “DRT” for the target cluster and Realm: “DRS” for the source cluster.
2. Set up the /etc/krb5.conf file on all hosts of both the source and target hosts:
   a. [realms]section - Enlist both the DRS and DRT realms, DRS from the source cluster's Kerberos KDC, admin_server, and default_domain settings.
   b. [domain_realm] - Enlist all the hosts of both source and target clusters.
   c. Add krbtgt/DRS@DRT principal on both the source and target hosts that have HDFS NameNode role.

```
$ sudo kadmin.local
kadmin.local: addprinc -pw cloudera krbtgt/DRS@DRT
WARNING: no policy specified for krbtgt/DRS@DRT; defaulting to no policy
Principal "krbtgt/DRS@DRT" created
kadmin.local: listprincs
```

3. In Cloudera Manager, perform the following steps:
   a. Enable DRT as Trusted Kerberos Realm in source cluster HDFS service's configuration.
   b. Enable DRS as Trusted Kerberos Realm (trusted_realm) in target cluster's configuration along with the source host name where HDFS NameNode role is present.
   c. Enable DRS as Trusted Kerberos Realm in target cluster HDFS service's configuration.
   d. Access the remote HDFS endpoint to verify whether the trust set up is successful. To access the remote HDFS endpoint, run the following commands:

```
kinit krbtgt/DRS@DRT
hadoop fs -ls hdfs://[***REMOTE HDFS ENDPOINT***]>:8020/
```

Configure parameters for Hive ACID table replication policies

Before you create a Hive ACID table replication policy, you must configure the required Hive parameters.

Recommended Hive configuration parameters for Hive ACID table replication policies

Before you create a Hive ACID table replication policy, you can configure the recommended parameters for optimum performance.

Procedure

1. Configure the following properties on the Cloudera Manager Clusters *Hive-on-Tez Service* Configuration page, and then restart the Hive-on-Tez service:
   a) Configure maximum concurrent policies to run at a time.

      Search for the Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml property, enter hive.scheduled.queries.max.executors parameter, and the required value.

      For example, if you set the value to 30, Replication Manager runs a maximum of 30 replication policies at a time.
   b) Configure the connection pool size. Ensure that the value is equal to or higher than the number of configured maximum concurrent policies.

      Search for the Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml property, enter datanucleus.connectionPool.maxPoolSize parameter and the required value.
2. Enable bootstrap load to run DistCp jobs in parallel from a single replication policy using the REPL LOAD command on the source cluster to set hive.exec.parallel to true, and then set the hive.exec.parallel.thread.number parameter equal to the number of cores at session level.

   For example, if the number of available cores in the source cluster is 128 and you want to run parallel replication policies, run the following commands:

```
set hive.exec.parallel.thread.number=128
REPL LOAD [***DATABASE NAME***] FROM [***DIRECTORY NAME***] WITH ('hive
.exec.parallel'='true'')
```

3. Preserve owner or user permissions, group permissions, and HDFS ACLs in source and target clusters during replication.

   You can append the DistCp command line options in any combination (u for user, g for group, p for permission, and a for ACL) to the distcp.options command to preserve the permissions during Hive ACID table replication.

The other DistCp command line options that you can use are r for replication number, b for block size, c for checksum-type, x for XAttr, and t for timestamp.

Note: You must have superuser privileges to preserve the user and group permissions, and HDFS ACLs.

You can use DistCp options only for the DistCp jobs that are initiated by Hive. To preserve the permissions and ACLs, set the DistCp command line options using the WITH clause in the REPL LOAD and REPL DUMP commands.

For example, to preserve the owner or user permissions, group permissions, and ACLs, run the REPL LOAD *[***DATABASE NAME***]* FROM *[***DIRECTORY NAME***]* WITH distcp.options.puga command.

Advanced Hive configuration parameters for Hive ACID table replication policies

You can configure the additional Hive service configuration parameters as necessary.

Procedure

1. Configure the event time to live (TTL) parameter to 7 days in the Hive-on-Tez service.
   a) Go to the Cloudera Manager Clusters *Hive-on-Tez service* Configuration page.
   b) Search for the Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml property.
   c) Enter the hive.metastore.event.db.listener.timetolive parameter and value as 7. The unit for the parameter is days.
   d) Enter the hive.repl.cm.retain parameter, and the value as 7d. 7d indicates seven days.

   Note: The hive.metastore.event.db.listener.timetolive parameter value must match the hive.repl.cm.retain parameter. Therefore, you need to configure the hive.repl.cm.retain parameter to 7 days as well. When you change one of the two properties, make sure that you update the other parameter with the same value.

   Tip: When an event's TTL expires, the event is removed from the metastore and the replication policy job shows a FAILED_ADMIN state with the error Notification events are missing in the meta store. To recover from this state, re-bootstrap the database.
2. Configure the event time to live (TTL) parameter to 7 days in the Hive service.
   a) Go to the Cloudera Manager Clusters *Hive service* Configuration page.
   b) Search for the Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml property.
   c) Enter the hive.metastore.event.db.listener.timetolive parameter and value as 7. The unit for the parameter is days.
   d) Enter the hive.repl.cm.retain parameter, and the value as 7d. 7d indicates seven days.

   Note: The hive.metastore.event.db.listener.timetolive parameter value must match the hive.repl.cm.retain parameter. Therefore, you need to configure the hive.repl.cm.retain parameter to 7 days as well. When you change one of the two properties, make sure that you update the other parameter with the same value.
3. Configure the metastore.scheduled.queries.execution.timeout parameter to 600 seconds.
4. Configure the metastore.housekeeping.threads.on parameter to true.

   Caution: Ensure that you set this parameter on only ONE instance of Hive Metastore. You can set this by navigating to the Hive 1 (Hive Metastore) / Instances Configuration page and then set the instance specific configuration.
5. Restart the services after you configure the parameters.

Parameters to optimize Hive ACID table replication performance

To optimize Hive ACID table replication performance, you can configure Hive configuration parameters.

hive.repl.retry.initial.delay

Configure the first retry delay in seconds.

The default value is 60 seconds.

hive.repl.retry.backoff.coefficient

Configure the exponential delay between retries. (Previous Delay) * (Backoff Coefficient) determines the next retry interval.

The default value is 1.2.

hive.repl.retry.jitter

Configure the random jitter to avoid all retries happening at the same time.

The default value is 30 seconds.

hive.repl.retry.max.delay.between.retries

Configure the maximum allowed retry delay in minutes after including exponential backoff.

The default value is 60 minutes.

hive.repl.retry.total.duration

Configure the total allowed retry duration in hours which is inclusive of all retries. Once this is exhausted, the policy instance is marked as failed and needs manual intervention to restart.

The default value is 24 hrs.

hive.repl.approx.max.load.tasks

Configure an approximate maximum number of tasks to run before the next set of tasks is dynamically generated. This is an approximate value because Hive stops at a slightly higher number as some events lead to a task increment that might cross the specified limit.

The default value is 10000.

hive.repl.partitions.dump.parallelism

Configure the number of threads to dump partition data information during repl dump.

The default value is 100.

hive.repl.run.data.copy.tasks.on.target

Configure the parameter to true so that replication runs the data copy tasks during the repl load operation.

The default value is true.

hive.repl.file.list.cache.size

Configure the threshold for the maximum number of data copy locations to be kept in memory. When the hive.repl.run.data.copy.tasks.on.target parameter is set to true, this parameter is not considered.

The default value is 10000.

hive.repl.load.partitions.batch.size

Configure the maximum number of partitions of a table to batch together during a replication load. All the partitions in a batch makes a single metastore call to update the metadata. The data for these partitions is copied before the metadata batch is copied.

The default value is 10000.

hive.exec.copyfile.maxnumfiles

Configure the maximum number of files that Hive uses to perform sequential HDFS copies between directories. To increase the copy speed for a large number of files, distributed copies (distcp) are used.

The default value is 1L.

hive.exec.copyfile.maxsize

Configure the maximum file size in bytes that Hive uses to perform single HDFS copies between directories. To increase the copy speed for bigger files, distributed copies (distcp) are used.

The default value is 32L * 1024 * 1024.

hive.exec.parallel.thread.number

Maximum number of Hive ACID table replication policies that can run in parallel. The maximum number of parallel policies is equal to the number of available cores in the source cluster. Set this property at session level.

Before you set this value, configure the hive.exec.parallel parameter to true by running the REPL LOAD command using the WITH clause.

Configure file access control lists for Impala user

Before you create Hive ACID table replication policies, you need to configure the file access control lists for an Impala user to access the cmroot directory based on whether the source cluster has one encryption zone, multiple encryption zones, or no encryption zone.

One encryption zone or no encryption zone

When the source cluster has only one encryption zone or no encryption zone, you can run the following commands to provide the Impala user access to cmroot directory:

- hdfs dfs -setfacl -m default:group:hive:rwx *[***CMROOT PATH IN hive.repl.cmrootdir***]*
- hdfs dfs -setfacl -m user:impala:rwx *[***CMROOT PATH IN hive.repl.cmrootdir***]*

Multiple encryption zones

When there are multiple encryption zones in the source cluster, you must manually configure the file access control lists for Impala users for each encryption zone. In each encryption zone, a cmroot directory is available in the root of the encryption zone.

Run the following commands to set the file access control list for the user and group:

- hdfs dfs -setfacl -m default:group:hive:rwx *[***ENCRYPTION ZONE PATH or VALUE OF hive.repl.cm.encryptionzone.rootdir***]*
- hdfs dfs -setfacl -m user:impala:rwx *[***ENCRYPTION ZONE PATH or VALUE OF hive.repl.cm.encryptionzone.rootdir***]*

Note: The default value of hive.repl.cm.encryptionzone.rootdir is .cmroot.

For example, if the first encryption zone is /user/hive/encr1 and the other encryption zone is /user/hive/encr2, you must provide permissions for both the encryption zones. To provide the required permissions, perform the following steps:

1. Run the following commands to provide access permissions to the cmroot directory in the first encryption zone:

```
hdfs dfs -setfacl -m default:group:hive:rwx /user/hive/encr1/.cmroot
hdfs dfs -setfacl -m user:impala:rwx /user/hive/encr1/.cmroot
```

2. Run the following commands to provide access permissions to the cmroot directory in the second encryption zone:

```
hdfs dfs -setfacl -m default:group:hive:rwx /user/hive/encr2/.cmroot
hdfs dfs -setfacl -m user:impala:rwx /user/hive/encr2/.cmroot
```

Creating Hive ACID table replication policy

You can create a Hive ACID replication policy after you set up the environment and configure the required parameters.

Before you begin

Verify whether the prerequisites are met. For more information, see Preparing to create Hive ACID table replication policies on page 71.

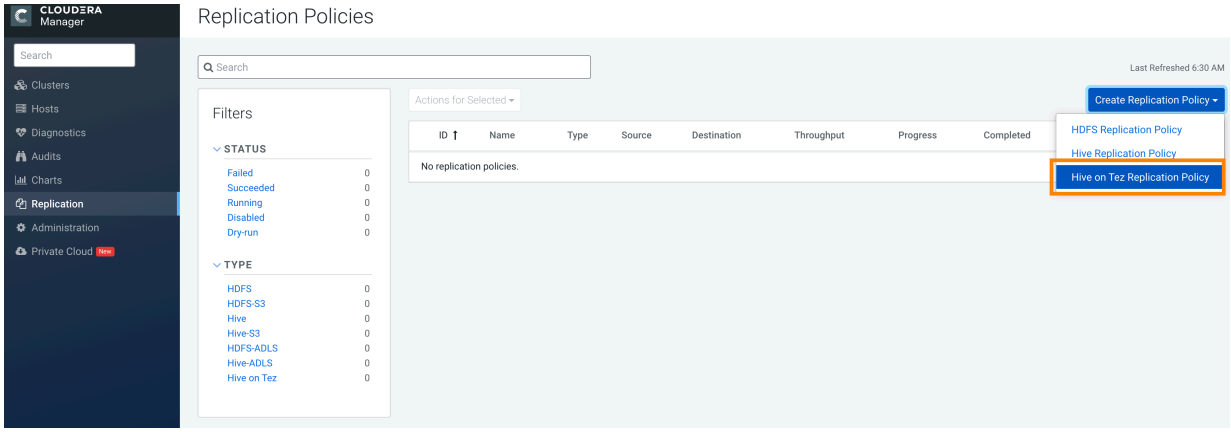
Important: To replicate managed tables (ACID) and external tables in a database successfully, you must perform the following steps in the order shown below:

1. Create Hive ACID table replication policy for the database to replicate the managed data.
2. After the replication completes, create the Hive external table replication policy to replicate the external tables in the database.

Procedure

1. Go to the Replication Replication Policies page in the Cloudera Manager for the target cluster where the peer is set up.
2. Click Create Replication Policy.
3. Select Hive ACID Table Replication Policy.

   The following sample image shows the Hive ACID Table Replication Policy option on the **Replication Policies** page in Cloudera Manager:



   The **Create Hive ACID Table Replication Policy** wizard appears.

4. In the **General** tab, configure the following options:

Option	Action to perform
Policy Name	Enter a unique name for the replication policy.
Source	Choose the source cluster.
Destination	Choose the target cluster.

Option	Action to perform
Destination Staging Path	Enter a valid path to the staging location. The path must have the required permissions for the *hive* user. For example: /user/hive/data. Hive ACID table replication policy uses a common staging location on the source or target cluster. After you enter a valid path to the staging location, the hive.repl.rootdir parameter is configured with this path. The REPL DUMP command dumps data in the staging location in the source cluster. The REPL LOAD command reads the data from the staging location. The REPL DUMP command runs in the source cluster and the REPL LOAD command runs in the target cluster. **Important:** Do not change or delete this path until the replication policy is in force and do not edit or modify the path during the replication life cycle. If you edit, change, or delete the path, replication errors occur or the replication is incomplete.
Source Database	Enter the source database name.
Schedule	Schedule the replication job as necessary. The Hive ACID table replication policy uses the Hive scheduler to schedule the replication policies. You can choose the following schedule options: • Every - Choose an interval. The schedule starts at the next exact interval based on the server time. If you want to start it immediately, issue a Run Now operation. For example, if you are in California and the destination cluster is in the Europe (Frankfurt) region, the schedule is set based on the server time, that is, Frankfurt local time. • Unix Cron Expression - Enter a valid Unix cron expression. You can generate the Unix cron expression in the following page, and then use it in the Create Hive ACID Table Replication Policy wizard: https://www.freeformatter.com/cron-expression-generator-quartz.html For example, you can use 0 */30 * ? * * to schedule the policy to run every 30 minutes, 0 0 */4 ? * * to schedule the run for every 4 hours, or 0 0 0 * * ? every day at midnight.
Run as Username	Enter hive.

5. Click the Resources tab to configure the following options:

Option	Description
Scheduler Pool	(Optional) Enter the name of a resource pool. The value you enter is used by the MapReduce Service when Cloudera Manager runs the MapReduce job for the replication. You can use one of the following property: • MapReduce – Fair scheduler: *mapred.fairscheduler.pool* • MapReduce – Capacity scheduler: *queue.name* • YARN – *mapreduce.job.queuename* **Note:** The DistCp job running on the target cluster or a YARN uses the *mapreduce.job.queuename* property.
Maximum Number of Copy Mappers	Enter the maximum number of simultaneous copy mappers for DistCp. The default value is 20.

Option	Description
Maximum Bandwidth per Copy Mapper	Enter the bandwidth limit for each mapper. Default is 100 MB. The total bandwidth used by the replication policy is equal to Maximum Bandwidth multiplied by Maximum Map Slots. Therefore, you must ensure that the bandwidth and map slots you choose do not impact other tasks or network resources in the target cluster. **Tip:** The Throughput field on the Cloudera Manager Replication Policies page shows the maximum bandwidth set for the replication policy during replication policy creation.

6. Select the Advanced tab to configure the following options:

Option	Description
Policy Description	(Optional). Enter a brief description for the replication policy.
Overrides	Enter key-value pairs to override parameters for Hive ACID table replication configuration. For example, when you use HA-based clusters, you can enter the relevant key-value pairs. During the replication process, the Replication Manager overrides the key-values pairs. You can also pass the DistCp argument.

7. Click Save to run the Hive ACID table replication policy.

When a non-recoverable error appears with the FAILED_ADMIN status for a replication job, perform the following steps to fix the error:

1. Go to the error log path.
2. Search for the file named _non_recoverable.
3. Check the error stack that is printed in the _non_recoverable file.
4. Fix the error.
5. Delete the _non_recoverable file. For the next replication jobs in the policy to function normally, the _non_rec overable file must be deleted.

Managing Hive ACID table replication policies

After you create a replication policy, you can run the replication job, disable or delete the job, edit the policy configuration, or view the replication job history in Cloudera Manager.

Procedure

1. Go to the Cloudera Manager Replication Replication Policies page.

   The following replication policy details appear on the page:

Columns	Description
ID	Automatically generated replication policy ID.
Name	Name of the replication policy.
Type	Shows Hive ACID for Hive ACID replication policies.
Source	Source cluster used in the replication policy.
Destination	Target cluster used in the replication policy.
Progress	Shows a spinner when the replication policy job is running.
Completed	Timestamp when the replication job is submitted to the Hive service.
Next Run*	Shows Managed by Hive message. Hover over the message to see more information about the next scheduled run.

Columns	Description
Message	Shows the status of the replication job. The following job states of the replication job run appear depending on the replication job status: • **Waiting for Update** appears after the replication policy creation is complete and remains until the job status is confirmed by the Hive service. • **Running** appears when the replication job is in progress. • **Failed** appears after the replication policy has failed. • **Skipped** appears when the replication job is skipped. • **Success** appears after the replication job completes successfully.
*When you schedule and submit a Hive ACID replication policy, the **Next Run** field shows the None scheduled message on the **Replication Policies** page. When the next run is scheduled, the date and time do not appear. You can ignore the None scheduled message as the replication job runs on Hive as scheduled or as per the schedule clause. Note that the schedules are managed by Hive. Cloudera Manager does not run any scheduled runs.	

2. Select the required replication policy.
3. Click Actions to view the following action items:
   a) Show History opens the **Replication History** page where you can view the replication policy job history.

      On this page, you can view the replication policy name, the replication policy type, the chosen source and destination clusters for the policy, and the next scheduled run.

      The page also shows the following statistics for each replication policy job:

      - Start Time of a replication policy job.
      - Duration or time taken to complete the job.
      - Outcome of the current job status.
      - Origin of collected Hive metrics. Click SOURCE or TARGET in the field to view the metrics for the replication job.
      - Total number of Tables to be replicated to the number of tables replicated successfully.
      - Functions column is incremented whenever a function is processed during dump and load operations.
      - Events column is incremented for every event dumped during dump operation and every event loaded during load operation. The counts for dump and load operation might not match because they are distinct operations.

   b) Edit Configuration allows you edit the schedule of the replication policy.
   c) Run Now runs the replication job.
   d) Disable the selected replication job.
   e) Delete the selected replication job.

Troubleshooting Hive ACID table replication policies

The troubleshooting scenarios in this topic help you to troubleshoot the Hive ACID table replication policies in Replication Manager.

In Cloudera Manager, the history of a schedule appears as FAILED and the status shows SKIPPED. Why are the SKIPPED runs listed as FAILED runs?

This scenario appears when there is no data to load during replication load on the target cluster.

FAILED with SKIPPED status might indicate an issue with the dump schedule on the source cluster. This can also appear when the dump completes after the load starts which might result in no data to load. Note that the first run (bootstrap) of the schedule takes a longer time than the subsequent (incremental) runs. Hence, the Hive query on the target side (load) might fail because the query runs at the same time as on source before the source completes the dumping operation.

How to recover a schedule from FAILED_ADMIN state?

When a non-recoverable error appears for a replication job and the status says FAILED_ADMIN, you can perform the following steps to recover a schedule from this state:

1. Go to the error log path.
2. Search for the file _non_recoverable.
3. Search for the error stack in the _non_recoverable file.
4. Fix the error.
5. Delete the _non_recoverable.

Note: For the next replication jobs in the policy to function normally, the _non_recoverable file must be deleted.

Why are notification events missing in the metastore?

One of the possible errors that might appear with FAILED_ADMIN status is when the notification events' TTL expires. This results in notifications being deleted in the metastore. In this scenario, the workaround is to start a fresh bootstrap phase of replication.

To re-bootstrap the database in the source cluster, perform the following steps:

1. Use beeline to drop the target database.
2. Remove the dump directory on HDFS for the required policy. The path of the _non_recoverable error file path has the dump directory path.

The policy schedule resumes automatically with the bootstrap phase.

How does Hive ACID table replication handle default and custom locations for databases and tables?

The following use cases show how the default location and custom locations for databases and tables are handled during Hive ACID table replication:

- Use case 1 - Database location and managedlocation properties.
  - If the source database properties location and managedlocation are set to the default location (<dbname>.db.toLowerCase()), the target database properties location and managedlocation are also set to the default location after replication.
  - If the source database properties location and managedlocation are set to custom locations, the target database properties location and managedlocation retain the corresponding custom locations on the target cluster after replication.

  By default, the custom location is retained on the target cluster. You can disable this behaviour by configuring the hive.repl.retain.custom.db.locations.on.target policy-level configuration property to false. When you disable this property and run the Hive ACID table replication, the replicated database locations on the target cluster are set to default locations, irrespective of whether the database locations on the source are set to default or custom locations.

- Use case 2: Table location and managedlocation properties.
  - After replication, a replicated managed table inherits the parent's database managedlocation property irrespective of whether the managedlocation property of the parent's database is set to the default location or custom location on the source cluster.
  - After replication, a replicated external table derives its location from the value of the hive.repl.replica.external.table.base.dir property and the external table location on the source cluster.

    For example, if an external table ext_tab1 is located at /ext_loc/ext_tab1/ on the source cluster and the hive.repl.replica.external.table.base.dir property is configured as /ext_base1 on the target, the location for ext_tab1 on the target cluster is /ext_base1/ext_loc/ext_tab1.

    The hive.repl.replica.external.table.base.dir property is derived from the value you set for the External Table Base Directory option in the Hive ACID table replication policy.

Which replication policy in Replication Manager replicates both the managed tables (ACID) and external tables in a database?

To replicate managed tables (ACID) and external tables in the database successfully, you must first replicate the ACID tables using Hive ACID table replication policy. After the replication policy run completes, create the Hive external table replication policy to replicate the external tables in the database.

Note: You must ensure that the target database name is the same as the source database name, otherwise issues appear during or after data replication.

To accomplish this task, perform the following steps:

1. Create a Hive ACID table replication policy where you choose the required database. The replication policy replicates data and metadata of the ACID tables in the database.

   The first run of the replication policy performs a bootstrap replication. During bootstrap replication, the target database is created and all the ACID tables are replicated to the target database. The subsequent policy runs are incremental. During incremental replication, only the source database changes between the current run and previous run are replicated.
2. Ensure that the first Hive ACID table replication policy run is complete in Replication Manager.
3. Create a Hive external table replication policy for the database. After policy creation is complete, a full replication (bootstrap) of data and metadata of all the external tables from the source database to target database is initiated. After the bootstrap replication is complete, the next policy run jobs leverage the HDFS snapshots to perform incremental replication of external table data.

   Note: Subsequent replication job runs perform full metadata replication and incremental data replication.

What table types in Hive does Replication Manager support?

The following replication policies replicate the given table types in Hive:

- Hive ACID table replication policies replicate data and metadata of the following table types in Hive:
  - Managed: CRUD transactional
  - Managed: Insert-only transactional
- Hive external table replication policies replicate data and metadata of external tables.

After creating a Hive external table replication policy, the "Bootstrap REPL LOAD is not allowed on Database: sourceDB as it is not empty. One or more tables/functions exist." error appears. How to resolve this issue?

This error appears if you create the Hive external table replication policy before you create the Hive ACID table external table policy.

To resolve this issue, the administrator must run the following steps:

1. Drop the database with the replicated data and metadata on the target cluster.
2. Create a Hive ACID table replication policy.
3. After the Hive ACID table replication policy run completes, create a Hive external table replication policy.

Can an existing Hive ACID replication policy on a database be deleted and then recreated?

Yes, you can delete and recreate the Hive ACID replication policy on a database.

Before you begin: You must identify whether there is an existing Hive external table replication policy running on the database.

Perform the following steps to delete an existing Hive ACID replication policy on a database and then recreate it:

1. Go to the target Cloudera Manager Replication Replication Policies page.
2. Click Actions Disable for the Hive external table replication policy that is replicating the database.
3. Click Actions Delete for the required Hive ACID replication policy.
4. Delete the contents in the staging location.

   Tip: The staging location is the Destination Staging Path that you specified on the **General** tab during the Hive ACID replication policy creation process.

5. Reset the `repl.source.for` value in the database properties using the ALTER DATABASE *[***DATABASE NAME***]* SET DBPROPERTIES('repl.source.for'='’); command.
6. Drop the database on the target cluster using the DROP DATABASE *[***DATABASE NAME***]* command.
7. Create a Hive ACID replication policy for the database on the target Cloudera Manager Replication Replication Policies page.
8. **Important:** Perform the next step after the first run (bootstrap run) of the Hive ACID replication policy is complete.

   Click Actions Enable for the Hive external table replication policy that you disabled in Step 3.

Iceberg replication policies

Iceberg replication policies replicate Iceberg V2 tables, created using Spark (read-only with Impala), between CDP Private Cloud Base 7.1.9 or higher clusters using Cloudera Manager 7.11.3 or higher versions.

Starting from CDP Private Cloud Base 7.1.9 SP1 using Cloudera Manager 7.11.3 CHF7, you can enter the maximum number of snapshots to process for an export batch, add one or more key- value pairs to hdfs-site.xml and core-site.xml files on the source and target clusters, replicate Atlas metadata for Iceberg tables, and replicate Iceberg tables residing in custom directories using Iceberg replication policies.

Note: Replicating Atlas metadata using Hive external table replication policies and Iceberg replication policies, and replicating the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities in the source cluster to the target cluster using Atlas replication policies is a technical preview feature. It is not recommended for production deployments.

Cloudera recommends that you try this feature in development or test environments. To enable this feature, contact your Cloudera account team.

Apache Iceberg is a cloud-native, high-performance open table format for organizing petabyte-scale analytic datasets on a file system or object store. Iceberg supports ACID compliant tables which includes row-level deletes and updates and can define large analytic data tables using open format files.

Iceberg replication policies:

- replicate the metadata and catalog from the source cluster Hive Metastore (HMS) to target cluster HMS.

  The catalog contains the current metadata pointer/file and is stored in the Hive HMS. The metadata file contains the snapshots. The snapshots point to the manifest list that has the manifest files. The manifest files in turn point to the data files.
- replicate the data files in the HDFS storage system from the source cluster to the target cluster. The Iceberg replication policy can replicate only between HDFS storage systems.
- replicate all the snapshots from the source cluster. This allows you to run time travel queries on the target cluster.
- replicate at table-level.

  You must ensure that the tables are in the default warehouse location because Iceberg replication policies do not replicate tables in a custom location.

Some use cases where you can use Iceberg replication policies are to:

- replicate Iceberg tables between on-premises clusters to archive data or run analytics.
- implement passive disaster recovery with planned failover and incremental replication at regular intervals between two similar systems. For example, between an HDFS to another HDFS system.

How Iceberg replication policy works

Replication Manager performs several steps to replicate the Iceberg tables when you create or run an Iceberg replication policy.

About this task

The following list shows a few high-level steps that are completed during the replication process:

Procedure

1. Determines the tables to replicate depending on the choice you made during the Iceberg replication policy creation process.
2. Reads the table names to fetch the checkpoint for the tables from the target cluster HMS. A checkpoint is the metadata about the latest Iceberg snapshot for a table on the target cluster and is saved in an HDFS file.
3. Initiates the exportCLI command in the source cluster to generate a list of files (manifest files, data files, and delete files) to copy from the source cluster to the target cluster.
4. Copies the files from the source cluster to the target cluster using DistCp jobs which takes advantage of the transfer bandwidth of the target cluster.

   The job copies the data files directly to the target data root directory, and it copies the metadata files to a temporary staging location where it is further processed as explained in the next step.
5. Transforms the copied manifest files to point to the correct manifest files pointers and data files on the target cluster, deletes the pre-transformed manifest files, and updates the target HMS with the latest snapshot.

How Atlas metadata replication for Iceberg tables work

Atlas metadata for the chosen Iceberg tables can be replicated using Iceberg replication policies.

Note: Replicating Atlas metadata using Hive external table replication policies and Iceberg replication policies, and replicating the metadata and data lineage of all the Hive external tables, Iceberg tables, and any other Atlas supported entities in the source cluster to the target cluster using Atlas replication policies is a technical preview feature. It is not recommended for production deployments.

Cloudera recommends that you try this feature in development or test environments. To enable this feature, contact your Cloudera account team.

During the Iceberg replication policy creation process, if you:

- choose the General Replicate Atlas Metadata option, Replication Manager:
  1. runs a bootstrap replication for all the chosen Iceberg tables and its Atlas metadata during the first replication policy run. Bootstrap replication replicates all the available Iceberg data and its associated Atlas metadata.
  2. runs incremental replication on the Iceberg data and its Atlas metadata during subsequent replication runs. Here, the delta data and metadata gets replicated during each run.
- choose to replicate an Iceberg table that was created using 'create table as select (CTAS)', Replication Manager sets the Skip lineage option to false and the Fetch type option to CONNECTED during the Iceberg replication policy run.

  Tip: You can create an Iceberg table based on an existing Hive, Spark, or Impala table using the CTAS query where you can optionally include a partitioning spec for the created table.

Use case

You have an original or base table named T1. You create table T2 using CTAS from T1. Similarly, you create T3 from T2, and T4 from T3. During the Iceberg replication policy creation process, you choose T2 as source table, and then choose Replicate Atlas metadata. In this scenario, Replication Manager performs the following tasks during the replication policy run:

1. Sets Skip lineage to false, and Fetch type to CONNECTED during Atlas replication step.
2. Replicates T2 and all the Atlas entities connected to it, which includes the hdfs_path.
3. Replicates T1 and T3 Iceberg tables.

Preparing to create Iceberg replication policies

Before you create an Iceberg replication policy, you must complete the prerequisites.

Before you begin

Iceberg replication policies can replicate only Iceberg V2 tables, created using Spark (read-only with Impala).

Procedure

- Ensure that the source cluster and target cluster versions are CDP Private Cloud Base 7.1.9 or higher using Cloudera Manager version 7.11.3 or higher versions.
- Activate the Iceberg Replication parcel. The parcel might be included in your Cloudera Runtime distribution or in a separate distribution. For more information, contact your Cloudera account team.
- Add the Iceberg Replication service on both the clusters?

  To add a service, go to the Cloudera Manager Clusters *[***CLUSTER NAME***]* page and click Actions Add Service . For more information see, Adding a Service.
- Ensure that you have the Atlas user credentials in addition to the Replication Administrator or Full Administrator roles to replicate Atlas metadata. The `atlas` user must also have relevant read and write permissions to the staging locations.

Creating Iceberg replication policy

You can create Iceberg replication policies in CDP Private Cloud Base 7.1.9 clusters using Cloudera Manager 7.11.3 or higher versions to replicate Iceberg V2 tables, created using Spark (read-only with Impala).

Procedure

1. Add the source cluster as a peer to the target cluster. An Iceberg replication policy requires a replication peer to locate the source data. You can use an existing peer or add a new peer.

   For information about adding a source cluster as a peer, see Adding cluster as a peer.

   Note: Peers that have Iceberg Replication service added to their clusters can be used as sources when you create an Iceberg replication policy.

2. Go to the Cloudera Manager Replication Replication Policies page in the target cluster where the peer is set up.
3. Click Create Replication Policy Iceberg Replication Policy .

   The **Create Iceberg Replication Policy** wizard appears.
4. Configure the following options on the **General** tab:

Option	Description
Policy Name	Enter a unique name for the replication policy.
Source	Choose the source cluster that has the required peer, the required source data to replicate, and the source Iceberg Replication service.
Destination	Choose the target cluster that has the required target Iceberg Replication service. The drop-down list shows the clusters that are managed by the current Cloudera Manager.
Schedule	Choose: • Immediate to run the replication policy immediately after policy creation is complete. • Once to run the schedule one time in the future. Set the date and time. • Recurring to run the schedule periodically in the future. Set the date, time, and interval between runs. You must consider the following factors before you configure the replication frequency or recurring schedule: • The anticipated rate of change and the frequency of the schedule can predict the RTO (Recovery Time Objective) and RPO (Recovery Point Objective) during a disaster recovery process. Therefore, choose a schedule that provides an optimal RTO and RPO. • Recurring frequency impacts the compute load on the entire system. That is, frequent replication affects the overall compute capacity of the participating nodes in the replication process which in turn can impact the other workloads running on these nodes.
Inclusion Table Filters	Enter the one or more database and table names to include for replication. The table name can be a Java Regular Expression, or the complete table name that is stored in the catalog. Use “|” to separate the table names.
Exclusion Table Filters	Enter the one or more database and table names to exclude from replication. The table name can be a Java Regular Expression, or the complete table name that is stored in the catalog.
Run as Username	Enter the username to run the MapReduce job. By default, MapReduce jobs run as hdfs. To run the MapReduce job as a different user, enter the user name. If you are using Kerberos, you must provide a user name here, and it must have an ID greater than 1000.
Replicate Table Column Statistics	Choose to copy the table column statistics associated with the chosen Iceberg tables.

Option	Description
Alternate target data root	Optional. Specify an alternate root location for all the tables in the replication scope. All the Iceberg table data/metadata are rebased in this location and keeps the source directory structure intact.
Replicate Atlas Metadata	Choose to copy the Atlas metadata and data lineage associated with the chosen Iceberg tables. For more information, see How Atlas metadata replication for Iceberg tables work

5. Configure the following options on the **Resources** tab, Replication Manager uses these options to optimize the DistCp jobs during data replication:

Option	Description
Custom YARN Queue	Optional. Enter the name of the YARN queue for the cluster to which the replication job is submitted if you are using Capacity Scheduler queues to limit resource consumption. The default value for this field is *default*.
Maximum Number of Copy Mappers	Optional. Enter the number of map slots per mapper, as required. The default value is 20.
Maximum Bandwidth Per Copy Mapper	Optional. Enter the bandwidth per mapper, as required. The default value for the bandwidth is 100 MB per second for each mapper. The total bandwidth used by the replication policy is equal to Maximum Bandwidth multiplied by Maximum Map Slots. Therefore, you must ensure that the bandwidth and map slots you choose do not impact other tasks or network resources in the target cluster. Adjust this setting so that each map task is throttled to consume only the specified bandwidth. Each map task (simultaneous copy) 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. You can adjust this setting so that each map task is throttled to consume only the specified bandwidth and the net bandwidth used tends towards the specified value.

6. Configure the following options on the **Advanced** tab:

Option	Description
Use Batch Size	Choose and enter the maximum number of snapshots to process for an export batch. This limits the amount of work to be processed in a single batch and can improve throughput. By default, this option is clear so as to process all the available snapshots in an export batch.
Advanced Configuration Snippet (Safety Valve) for source hdfs-site.xml	Add one or more key-value pairs to the hdfs-site.xml file on the source cluster. New key-value pairs are added to the file. Existing key-value pairs are overwritten in the file.
Advanced Configuration Snippet (Safety Valve) for source core-site.xml	Add one or more key-value pairs to the core-site.xml file on the source cluster. New key-value pairs are added to the file. Existing key-value pairs are overwritten in the file.
Advanced Configuration Snippet (Safety Valve) for destination hdfs-site.xml	Add one or more key-value pairs to the hdfs-site.xml file on the target cluster. New key-value pairs are added to the file. Existing key-value pairs are overwritten in the file.
Advanced Configuration Snippet (Safety Valve) for destination core-site.xml	Add one or more key-value pairs to the core-site.xml file on the target cluster. New key-value pairs are added to the file. Existing key-value pairs are overwritten in the file.

7. Click Create.

> **Note:** Only one Iceberg replication policy can actively replicate from the same Iceberg table at any point in time. If a replication is started from an Iceberg table which is already being replicated, it is considered as an error and the replication policy starting the concurrent replication fails and it appears as **Failed** on the **Replication Policies** page.

Results

The replication policy appears on the **Replication Policies** page. It can take up to 15 seconds for the task to appear.

If you selected Immediate in the **Schedule** field, the replication job starts replicating after you click Save Policy.

Manage and monitor Iceberg replication policies

After you create an Iceberg replication policy, you can perform and monitor various tasks related to the replication policy. You can view the job progress and replication logs. You can edit the advanced options to optimize a job run. You can suspend a job and also activate a suspended job.

Replication policy details on the Replication Policies page

On the **Replication Policies** page, you can view the following details about the replication policy:

- Shows a row of information for each replication policy, and the following columns for each replication policy:
  - Internally generated **ID** for the replication policy. Click the column label to sort the replication policies.
  - Replication policy **Name** that you provide during replication policy creation.
  - Replication policy **Type**.
  - **Source** cluster in the replication policy.
  - **Destination cluster** in the replication policy.
  - Average **Throughput** per mapper/file for all the files written.

    > **Note:** The throughput does not include the combined throughput of all mappers and the time taken to perform a checksum on a file after the file is written.

  - Replication job **Progress**.
  - Timestamp when the replication job **Completed**.
  - Replication policy job's **Next Run**.
- Provides the following options under the **Actions** menu:
  - **Show History** opens the **Replication History** page for the replication policy.
  - **Edit Configuration** enables you to change the replication policy options as required.
  - **Run Now** initiates a replication job.
  - **Disable** an active replication policy.

    You can Enable it later, as necessary.
- **Delete** the replication policy permanently. Deleting a replication policy does not delete copied files or tables.

Replication History page

Click Actions Show History for a replication policy on the **Replication Policies** page to view the **Replication History** page.

On the **Replication History** page, you can view the following run details about a replication policy job:

- Shows the replication policy **Name**; replication policy **Type**; **Source** cluster name; **Destination** cluster name; and **Next Run** of the replication policy.

- Shows a row of information for each replication policy job run, and the following columns for each replication policy:

Column	Description
Start Time	Shows the timestamp when the replication job started.
Origin TimeStamp (UTC)	Shows the timestamp when the export step started on the source cluster.
Duration	Shows the time taken to complete the replication job.
Outcome	Shows the replication job status as **Running**, **Successful**, or **Failed**.
Number of tables processed	Shows the number of tables processed by the replication policy job.
Number of files copied	Shows the number of successfully copied files.
Number of files deleted	Shows the number of files deleted by the replication policy job.
Number of manifests transformed	Shows the number of copied manifest files (with incorrect path replacements) that were corrected to point to the correct data files on the target cluster.

- Expand a job to view the following information on the **All Recent Commands** window:
  - **Status** of the replication job.
  - **Iceberg Replication** in the **Context** field opens the Clusters Iceberg Replication window where more details about the replication policy job appears.
  - Replication job **Started At** timestamp.
  - **Duration** to complete the job.
  - **Download** the results to your machine.
  - Expand to **Show All Steps**, **Show Only Failed Steps**, or **Show Only Running Steps** for the commands used by Iceberg replication policy.
  - **Show Command Timing** shows the timeline for the commands used by the Iceberg replication policy.

Ozone replication policies

Apache Ozone is a scalable, distributed, and high performance object store optimized for big data workloads and can handle billions of objects of varying sizes. Ozone storage is co-located on HDFS. You can create Ozone replication policies in CDP Private Cloud Base Replication Manager to replicate data in Ozone buckets between CDP Private Cloud Base 7.1.8 clusters or higher using Cloudera Manager 7.7.1 or higher.

Cloudera supports the following types of Ozone storage:

- Object store buckets (OBS), which are storage buckets where all the keys are written into a flat namespace and can be accessed using S3 interface provided by Ozone.
- File System Optimization (FSO), which are Hadoop-compatible file system buckets where the rename and delete operations on the directories are atomic. These buckets can be accessed using Filesystem APIs and S3 interfaces.
- Legacy buckets, which are Ozone buckets created prior to CDP Private Cloud Base 7.1.8 and use the Ozone File System (ofs) protocol or scheme.

You can use Ozone replication policies to replicate or migrate the required Ozone data to another cluster to run load-intensive workloads, back up data, or for backup-restore use cases.

Ozone replication policies support data replication between:

- FSO buckets in source and target clusters using ofs protocol.

- legacy buckets in source and target clusters using ofs protocol.

  Note:

  - If one or both of the source and destination buckets is a legacy bucket, then the ozone.om.enable.filesystem.paths flag (cluster-level configuration property) in the ozone-site.xml file must be enabled on the cluster(s) with the legacy bucket.
  - Ozone replication uses `ofs` by default to replicate FSO or LEGACY buckets.
- OBS buckets in source and target clusters that support S3A filesystem using the S3A scheme or replication protocol.

How Ozone replication works

Ozone snapshots are enabled for all the buckets and volumes. If the incremental replication feature is enabled on the source and target clusters, to replicate Ozone data you can choose one of the following methods during the Ozone replication policy creation process:

Full file listing

By default, the Ozone replication policies use the full file listing method which takes a longer time to replicate data. In this method, the first Ozone replication policy job run is a bootstrap job; that is, all the data in the chosen buckets are replicated. During subsequent replication policy runs, Replication Manager performs the following high-level steps:

1. Lists all the files.
2. Performs a checksum and metadata check on them to identify the relevant files to copy. This step depends on the advanced options you choose during the replication creation process. During this identification process, some unchanged files are skipped if they do not meet the criteria set by the chosen advanced options.
3. Copies the identified files from the source cluster to the target cluster.

Incremental only

In this method, the first replication policy job run is a bootstrap job, and subsequent job runs are incremental jobs.

To perform the incremental job, Replication Manager leverages Ozone snapshots and the snapshot-diff capability to generate a diff report. The diff report contains the changed or new data from the source cluster. The subsequent replication policy replicates data based on the diff report.

Incremental with fallback to full file listing

In this method, the first replication policy job run is a bootstrap job, and subsequent job runs are incremental jobs. However, if the snapshot-diff fails during a replication policy job run, the next job run is a full file listing run. After the full file listing run succeeds, the subsequent runs are incremental runs. This method takes a longer time to replicate data if the replication policy job falls back to the full file listing method.

Preparing clusters to replicate Ozone data

You must prepare the clusters, create buckets in the target cluster, and configure additional configurations for OBS bucket replication before you create Ozone replication policies in CDP Private Cloud Base Replication Manager.

About this task

Before you create Ozone replication policies, ensure that the following prerequisites are complete:

Procedure

- Have you added the source cluster as a peer to the target cluster?

  For information about adding a source cluster as a peer, see Adding cluster as a peer.

- Have you created the bucket on the target cluster of the same type as the bucket on the source cluster from which the replication policy replicates data?

  Tip: Create a volume and then the bucket. For more information, see Managing storage elements using CLI..

  The following sample commands create a volume and an FSO bucket:

  ```
  ozone sh volume create o3://ozone1/vol1
  ozone sh bucket create o3://ozone1/vol1/buck1 --layout FILE_SYSTEM_OPTIM
  IZED
  ```

- Are the additional configurations required for OBS bucket replication configured when the source bucket is an OBS bucket?

  For more information, see Configuring properties for OBS bucket replication using Ozone replication policies on page 93.
- Are the source and target clusters SSL-enabled? If so, ensure that the SSL/TLS certificate exchange between two Cloudera Manager instances that manage source and target clusters respectively is configured.

  For more information, see Configuring SSL/TLS certificate exchange between two Cloudera Manager instances on page 19.
- Is Kerberos enabled on both the clusters? If so, perform the following steps:
  a) Configure a user with permissions to access HDFS and Ozone.
  b) Run the `sudo usermod -a -G om bdr` command to add the group name of the user (For example, the group name bdr) to the Ozone service configuration in target Cloudera Manager:

  Important: If Kerberos is enabled on both the clusters, you must run the kinit -kt /*[***PATH***]*/*[***TO***]*/om.keytab om command (the absolute path to the Ozone service's keytab) before you run any Ozone commands.
- Is Ranger enabled on the source cluster? If so, complete the following steps on the Ranger UI from source Cloudera Manager:
  a) Log into Ranger UI from source Cloudera Manager.
  b) Click cm_ozone on the **Service Manager** page.
  c) Add the user (that you configured in the previous step) to the all - volume, bucket, key policy name, and then set the groups for this policy as public.
- Is Ranger KMS enabled on the source and target clusters? If so, complete the following steps for the kms-site.xml file for the Ranger_KMS service on the source and target clusters:
  a) Locate and open the kms-site.xml file on the source Cloudera Manager.
  b) Add the following key-value pairs:
     - hadoop.kms.proxyuser.om.hosts=*
     - hadoop.kms.proxyuser.om.groups=*
     - hadoop.kms.proxyuser.om.users=*
  c) Save the file.
  d) Restart the Ranger_KMS service for the changes to take effect.
  e) Locate and open the kms-site.xml file on the target Cloudera Manager.
  f) Add the following key-value pairs:
     - hadoop.kms.proxyuser.om.hosts=*
     - hadoop.kms.proxyuser.om.groups=*
     - hadoop.kms.proxyuser.om.users=*
  g) Save the file.
  h) Restart the Ranger_KMS service for the changes to take effect.

Configuring properties for OBS bucket replication using Ozone replication policies

Before you replicate OBS buckets, you must configure additional properties that assist Ozone replication policies in CDP Private Cloud Base Replication Manager to replicate data in OBS buckets.

Procedure

1. Add the key-value pairs in the following table to the Ozone Client Advanced Configuration Snippet (Safety Valve) property in the ozone-site.xml file in the source cluster. Starting from CDP Private Cloud Base 7.1.9 SP1 using Cloudera Manager 7.11.3 CHF7, add the following key-value pairs to the ozone_replication_core_site_safety_valve property in the source cluster:

Property	Description
fs.s3a.endpoint	Enter the same value as in Ozone S3 gateway web UI as the source cluster. **Tip:** The source and target cluster have their own S3A endpoint URL.
hadoop.tmp.dir	Enter the temporary directory on the source cluster to buffer file uploads. Ensure that the user running the Ozone replication policy jobs has write access to the Hadoop temporary folder.
fs.s3a.secret.key	See Step 3 to get the required value.
fs.s3a.access.key	See Step 3 to get the required value.
fs.s3a.impl	Enter org.apache.hadoop.fs.s3a.S3AFileSystem.
ozone.om.snapshot.load.native.lib (Available from 7.1.9 CHF3 onwards)	Enter false. Incremental Ozone replication policy run uses snapshot-diff operation. This parameter ensures that the replication policy run is not affected if the snapshot-diff operation goes down during the replication policy run.

2. Add the key-value pairs in the following table to the Ozone Client Advanced Configuration Snippet (Safety Valve) property in the ozone-site.xml file in the target cluster. Starting from CDP Private Cloud Base 7.1.9 SP1 using Cloudera Manager 7.11.3 CHF7, add the following key-value pairs to the ozone_replication_core_site_safety_valve property in the target cluster:

Property	Description
fs.s3a.endpoint	Enter the same value as in Ozone S3 gateway web UI as the target cluster. **Tip:** The source and target cluster have their own S3A endpoint URL.
fs.s3a.secret.key	See Step 3 to get the required value.
fs.s3a.access.key	See Step 3 to get the required value.
fs.s3a.change.detection.version.required	Enter false.
fs.s3a.change.detection.mode	Enter none.
fs.s3a.path.style.access	Enter true.
fs.s3a.impl	Enter org.apache.hadoop.fs.s3a.S3AFileSystem.
hadoop.tmp.dir	Enter the temporary directory on the target cluster to buffer file uploads. Ensure that the user running the Ozone replication policy jobs has write access to the Hadoop temporary folder.

Property	Description
ozone.om.snapshot.load.native.lib (Available from 7.1.9 CHF3 onwards)	Enter false. Incremental Ozone replication policy run uses snapshot-diff operation. This parameter ensures that the replication policy run is not affected if the snapshot-diff operation goes down during the replication policy run.

3. If Kerberos is enabled on the source and target cluster, run the `ozone s3 getsecret --om-service-id=serviceId` command to get the secret and access key. Otherwise, enter any arbitrary value for the secret and access key.

   You can store the keys in a credstore such as JCEKS for non Auto-TLS clusters. After you store the keys, perform the following steps:

   a. Configure the credstore file path for the hadoop.security.credential.provider.path property in the ozone-site.xml file. For more information, see Using DistCp with Amazon S3.
   b. Add the HADOOP_CREDSTORE_PASSWORD parameter to the YARN Service Environment Advanced Configuration Snippet (Safety Valve) property for the YARN service in source Cloudera Manager.

      Note: If no password is set, enter none for the property.

4. The /s3v volumes store S3 buckets. By default, you can access the buckets in /s3v volumes using S3 interface. To access other buckets through the S3 interface, you must create a “symbolic linked” bucket. You can use the ‘symbolic linked’ bucket in Ozone replication policies.

   Configure the required OBS buckets as S3-compatible buckets using the following commands before you use it in Ozone replication policies:

   a. ozone sh volume create /s3v
   b. ozone sh volume create /*[***VOLUME NAME***]*
   c. ozone sh bucket create /*[***VOLUME NAME***]*/*[***BUCKET NAME***]*
   d. ozone sh bucket link /*[***VOLUME NAME***]*/*[***BUCKET NAME***]* /s3v/*[***SYMBOLIC LINKED BUCKET NAME***]*

5. Import the S3G CA certificate from the cluster to the local JDK path using the following commands:

   a) Run the keytool -importkeystore -destkeystore *[***JDK CACERTS_LOCATION***]* -srckeystore *[***CM-AUTO-GLOBAL_TRUSTSTORE.JKS LOCATION***]* -srcalias *[***CM_ALIAS_ON_SRC_CM***]* command on all the hosts of the *source* Cloudera Manager.

      For example, keytool -importkeystore -destkeystore /usr/java/default/lib/security/cacerts -srckeystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks -srcalias cmrootca-0

   b) Run the following commands on all the hosts of the target Cloudera Manager:

      1. keytool -importkeystore -destkeystore *[***JDK CACERTS LOCATION***]* -srckeystore *[***CM-AUTO-GLOBAL_TRUSTSTORE.JKS LOCATION***]* -srcalias *[***CM ALIAS ON SRC CM***]*
      2. keytool -importkeystore -destkeystore *[***JDK_CACERTS_LOCATION***]* -srckeystore *[***CM-AUTO-GLOBAL_TRUSTSTORE.JKS LOCATION***]* -srcalias *[***CM_ALIAS_ON_DEST_CM***]*

      For example,

```
keytool -importkeystore -destkeystore /usr/java/default/lib/security/cac
erts
-srckeystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_tru
ststore.jks -srcalias cmrootca-0
keytool -importkeystore -destkeystore /usr/java/default/lib/security/ca
certs
-srckeystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_tr
uststore.jks -srcalias cmrootca-1
```

Creating Ozone replication policies

You can create Ozone replication policies in CDP Private Cloud Base Replication Manager on the target cluster.

Before you begin

Consider the following points before you create Ozone replication policies:

- Data is replicated at bucket-level. Therefore, use *[***VOLUME***]/[***BUCKET***]* format to point to the required buckets during replication policy creation.
- Ozone replication policies perform incremental replication using file checksums and is supported by all the bucket types except OBS buckets.

Procedure

1. Go to the Cloudera Manager Replication Policies page on the target cluster.
2. Click Create Replication Policy Ozone Replication Policy .
3. On the **General** page, enter or choose the required values:

Option	Description
Name	Enter a unique name for the replication policy.
Path types	Choose one of the following path types depending on the Ozone storage: • FSO (FileSystemOptimized) to FSO - Enter the volume and bucket names in the source cluster. • OBS (ObjectStore) to OBS - Enter the bucket name in the source cluster. **Important:** Complete the steps in Configuring properties for OBS bucket replication using Ozone replication policies on page 93 before you use this option. • Full Path - Enter the path to the bucket in the ofs://*[***OZONE SERVICE ID***]/[***VOLUME NAME***]/[***BUCKET NAME***]* or s3a://*[***BUCKET NAME***]* format to replicate data between FSO or OBS buckets respectively. A bucket subpath can also be specified.
Source	Select the source cluster. **Note:** The clusters that you add as a peer on the target Cloudera Manager Peers page appear in the source cluster list.
Source Volume	Enter the source volume name.
Source Bucket	Enter the source bucket name.
Destination	Choose the target cluster.
Destination Volume	Enter the target volume name.
Destination Bucket	Enter the target bucket name.
Schedule	Choose: • Immediate to run the schedule immediately. • Once to run the schedule one time in the future. Set the date and time. • Recurring to run the schedule periodically in the future. Set the date, time, and interval between runs.

Option	Description
Listing type	Choose one of the following replication methods to replicate Ozone data: • Full file listing. • Incremental only • Incremental with fallback to full file listing To understand how each method works, see Ozone replication policies. This option appears only if the incremental replication feature is enabled on the source and target clusters.
Run As Username	Enter the username to run the MapReduce job. By default, MapReduce jobs run as *hdfs*. To run the MapReduce job as a different user, enter the user name. If you are using Kerberos, you must provide a user name here, and it must have an ID greater than 1000.
Run on Peer as Username	Enter the username if the peer cluster is configured with a different superuser. This is applicable in a kerberized environment.

4. Configure the following options on the Resources page:

Option	Description
Scheduler Pool	(Optional) Enter the name of a resource pool in the field. The value you enter is used by the MapReduce Service you specified when Cloudera Manager runs the MapReduce job for the replication. The job specifies the value using one of these properties: • MapReduce – Fair scheduler: mapred.fairscheduler.pool • MapReduce – Capacity scheduler: queue.name • YARN – mapreduce.job.queuename
Maximum Number of Copy Mappers	Enter the number of map slots per mapper, as required. The default value is 20.
Maximum Bandwidth Per Copy Mappers	Enter the bandwidth per mapper, as required. The default value for the bandwidth is 100MB per second for each mapper. The total bandwidth used by the replication policy is equal to Maximum Bandwidth multiplied by Maximum Map Slots. Therefore, you must ensure that the bandwidth and map slots you choose do not impact other tasks or network resources in the target cluster. Adjust this setting so that each map task is throttled to consume only the specified bandwidth. Each map task ((simultaneous copy) 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. You can adjust this setting so that each map task is throttled to consume only the specified bandwidth so that the net bandwidth used tends towards the specified value. **Tip:** The Throughput field on the Cloudera Manager Replication Policies page shows the maximum bandwidth set for the replication policy during replication policy creation.
Replication Strategy	Choose one of the following replication strategies: • Static distributes file replication tasks among the mappers up front to achieve an uniform distribution based on the file sizes. • Dynamic distributes the file replication tasks in small sets to the mappers, and as each mapper completes its tasks, it dynamically acquires and processes the next set of unallocated tasks. The default replication strategy is Dynamic.

5. Configure the following options on the Advanced Options tab:

Option	Description
Path exclusion	Click Add Exclusion to enter one or more regular expressions separated by comma. Replication Manager does not copy the subdirectories or files from the source that matches one of the specified regular expressions to the target cluster.
MapReduce Service	Select the MapReduce or YARN service to use.
Log path	Enter an alternate path for the logs, if required.
Description	Optionally, enter a description.
Error Handling	Select the following options as necessary: • Skip Checksum Checks - Determines whether to skip checksum checks on the copied files. If selected, checksums are not validated. Checksums are checked by default. **Note:** You must skip checksum checks to prevent replication failure due to non-matching checksums in the following cases: • Replications from an encrypted zone on the source cluster to an encrypted zone on a destination cluster. • Replications from an encryption zone on the source cluster to an unencrypted zone on the destination cluster. • Replications from an unencrypted zone on the source cluster to an encrypted zone on the destination cluster. Checksums are used for two purposes: • To skip replication of files that have already been copied. If Skip Checksum Checks is selected, the replication job skips copying a file if the file lengths and modification times are identical between the source and destination clusters. Otherwise, the job copies the file from the source to the destination. • To redundantly verify the integrity of data. However, checksums are not required to guarantee accurate transfers between clusters. HDFS data transfers are protected by checksums during transfer and storage hardware also uses checksums to ensure that data is accurately stored. These two mechanisms work together to validate the integrity of the copied data. • Skip Listing Checksum Checks - Whether to skip checksum check when comparing two files to determine whether they are same or not. If skipped, the file size and last modified time are used to determine if files are the same or not. Skipping the check improves performance during the mapper phase. Note that if you select the Skip Checksum Checks option, this check is also skipped. • Abort on Error - Whether to abort the job on an error. If selected, files copied up to that point remain on the destination, but no additional files are copied. Abort on Error is not selected by default.

Option	Description
Delete Policy	Choose the required options to determine whether the files that were deleted on the source should also be deleted from the destination directory. This policy also determines the handling of files in the destination location that are unrelated to the source. Options include: • Keep Deleted Files - Retains the destination files even when they no longer exist at the source. This is the default option. • Delete to Trash - If the HDFS trash is enabled, files are moved to the trash folder. This is not supported when replicating to S3 or ADLS. • Delete Permanently - Uses the least amount of space; use with caution.
Alerts	Choose to generate alerts for various state changes in the replication workflow. You can choose to generate an alert On Failure, On Start, On Success, or On Abort of the replication job. You can configure alerts to be delivered by email or sent as SNMP traps. If alerts are enabled for events, you can search for and view the alerts on the **Events** tab, even if you do not have email notification configured. For example, if you choose Command Result that contains the Failed filter on the Diagnostics Events page, the alerts related to the On Failure alert for all the replication policies for which you have set the alert appear. For more information, see Managing Alerts and Configuring Alert Delivery.

6. Click Create.

Results

The replication policy appears on the **Replication Policies** page. It can take up to 15 seconds for the task to appear.

If you selected Immediate in the **Schedule** field, the replication job starts replicating after you click Save Policy.

Managing Ozone replication policies

After you create an Ozone replication policy, you can perform and monitor various tasks related to the replication policy. You can view the job progress and replication logs. You can edit the advanced options to optimize a job run. You can suspend a job and also activate a suspended job.

Procedure

1. On the **Replication Policies** page, you can view the following details about the replication policy:
   a) Shows a row of information for each replication policy, and the following columns for each replication policy:
      - Internally generated **ID** for the replication policy. Click the column label to sort the replication policies.
      - Replication policy **Name** that you provide during replication policy creation.
      - Replication policy **Type**.
      - **Source** cluster in the replication policy.
      - **Destination cluster** in the replication policy.
      - Average **Throughput** per mapper/file for all the files written.

        Note: The throughput does not include the combined throughput of all mappers and the time taken to perform a checksum on a file after the file is written.
      - Replication job **Progress**.
      - Timestamp when the replication job **Completed**.
      - Replication policy job's **Next Run**.

   b) Provides the following options under the Actions menu:
      - Show History opens the **Replication History** page for the replication policy.
      - Edit Configuration allows you to change the replication policy options as required.
      - Dry Run simulates a run of the replication job where no files or tables are copied.

        After the dry run completes, select Actions Show History to view the potential error messages and the number and size of files or tables that would be copied in an actual replication appears on the **Replication History** page.
      - Run Now initiates a replication job.
      - Collect Diagnostic Data opens the **Send Diagnostic Data** dialog box where you can:
        - Collect Diagnostic Data for the last 10 runs of the replication policy, and Download it as a ZIP file to your machine.
        - Select Send Diagnostic Data to Cloudera (optionally, add a Cloudera support ticket number and comments) and click Collect Diagnostic Data to automatically send the bundle to Cloudera Support for further assistance.
      - Disable an active replication policy.

        You can Enable it later, as necessary.
      - Delete the replication policy permanently. Deleting a replication policy does not delete copied files or tables.
2. On the **Replication History** page, you can view the following details about a replication policy job:

   Tip: Click Actions Show History for a replication policy on the **Replication Policies** page to view the **Replication History** page.

   a) Shows the replication policy **Name**; replication policy **Type**; **Source** cluster name; **Destination** cluster name; and **Next Run** of the replication policy.
   b) Shows a row of information for each replication policy job run, and the following columns for each replication policy:

Column	Description
Duration	Time taken for the replication job to complete.
Outcome	Status of the replication job as **In progress**, **Successful**, or **Failed**.
Files Expected	Number of files expected to be copied and its file size based on the parameters of the replication policy.
Files Copied	Number of files copied and its file size for the replication job.

Column	Description
Files Failed	Number of files that failed to be copied and its file size for the replication job.
Files Deleted	Number of files that were deleted and its file size for the replication job
Files Skipped	Number of files skipped and its file size for the replication job. The replication process skips files that already exist in the destination and have not changed.

c) Expand a job to view the following information:

- Replication job **Started At** timestamp.
- **Duration** to complete the job.
- **Command Details** appear in a new tab after you click View.

  The **All Recent Commands** page shows the job **Status**; **Context** (click to view the service status page); **Started At** timestamp; **Duration** to complete the job run; and **Download** the job run command summary as a JSON file to your local machine.

  The page can also Show All Steps; Show Only Failed Steps; or Show only Running Steps of the replication policy job run commands with stdout and stderr output. Click Full Log file to view the logs in a new browser tab.

  For more information, seeViewing Running and Recent Commands.
- Click MapReduce Job ID to view more details about the job on the YARN service page.
- **Download CSV** files of the following **Ozone Replication Reports** to track the replication jobs and to troubleshoot issues:

Report	Description
Listing	Lists all the files and directories copied during the replication job.
Status	Shows the following status for each file as: • an **Error** occurred and the file was not copied. • a **Deleted** file. • an up-to-date file for which the replication was **Skipped**.
Error Status Only	Status report of all the copied files with errors. Each file shows the status, path, and message for the copied files with errors.
Skipped Status Only	Status report of all skipped files. Each file lists the status, path, and message for the databases and tables that were skipped.
Deleted Status Only	Status report of all deleted files. Each file lists the status, path, and message for the databases and tables that were deleted.
Performance	Summary report about the performance of the running replication job which includes the last performance sample for each mapper that is working on the replication job.
Full Performance	Performance report of the job which includes the samples taken for all mappers during the replication job.

- **Run As Username** is the username specified during replication policy creation to run the replication job.
- **Run on Peer as Username** is the username specified during replication policy creation.
- Message shows the total number of files copied to target cluster and the number of files that remain unchanged on the source cluster.

Ranger replication policies

You can create Ranger replication policies in CDP Private Cloud Base Replication Manager. The Ranger replication policies migrate the Ranger policies and roles for HDFS, Hive, and HBase services between Kerberos-enabled CDP Private Cloud Base 7.1.9 or higher clusters using Cloudera Manager 7.11.3. It can also migrate Ranger audit logs in HDFS.

Note: You can also create Ranger replication policies on Kerberos-enabled CDP Private Cloud Base 7.1.8 or higher clusters using Cloudera Manager 7.7.1 CHF6 and higher, if the Ranger replication feature flag is enabled.

Apache Ranger manages access control through a user interface that ensures consistent policy administration across Cloudera Data Platform (CDP) components. Security administrators can define security policies at the database, table, column, and file levels, and can administer permissions for specific LDAP-based groups or individual users.

The Ranger replication policy can replicate the following:

Ranger policies and roles

The Ranger policies that can be replicated include Ranger tag-based policies and Ranger resource-based policies. The replication policy always performs a complete export and import of Ranger policies.

Ranger audit logs in HDFS

Ranger audit logs can be replicated using superuser credentials. You must ensure that the Ranger audit log directory on the source cluster is snapshot-enabled. Replication Manager uses DistCp jobs to replicate Ranger HDFS audit log directories. Therefore, the first Ranger replication policy run to replicate the Ranger audit log directory is a bootstrap job and the subsequent runs are incremental.

Tip: You can go to the required Ranger audit log directory in Cloudera Manager and then enable snapshots for the directory.

You can choose to replicate only the Ranger policies and roles, or only the Ranger audit logs in HDFS during the Ranger replication policy creation process. The Ranger replication policy replicates from only one Ranger source service on the source cluster to only one Ranger destination service on the target cluster.

Some use cases where you can use Ranger replication policies are:

- when Ranger is used for file system-level access control for HDFS and Hive and you want to copy the Ranger policies to another cluster for backup purposes.
- when you want to move/replicate Ranger policies for Hive (SQL) or HBase data to another cluster for disaster recovery purposes.

How Ranger replication policy works

A Ranger replication policy can replicate Ranger policies and roles and Ranger audit logs in HDFS. The Ranger replication policy must complete several tasks to replicate the Ranger policies, roles, and Ranger audit logs successfully.

The high-level tasks that a Ranger replication policy job run performs in the background include the following steps:

1. On the source cluster, the Ranger policies and roles for the specified services are exported to a file, and the file is transferred to the target cluster.

   You can choose the services on the **Services** tab during Ranger replication policy creation.

2. Optionally, on the target cluster, the names of the Ranger service; the usernames; the file paths, database names, table names, and the URLs of the resources in the source cluster are transformed or mapped to the names in the target cluster in the file.

   You can choose the required User Mapping and Resources Mapping to transform or map on the **Advanced** tab during Ranger replication policy creation.
3. On the target cluster, the file is imported and ingested into the Ranger service.

   You can choose one of the following methods to ingest the file into Ranger service during Ranger replication policy creation:

   - Merge method (default). When you choose this method, Replication Manager merges the Ranger policies.

     For example, assume a Ranger policy in the destination Ranger service has *user1* and the same Ranger policy on the source cluster has *user2*. In this method, both *user1* and *user2* are added in the destination Ranger policy after replication.
   - Override method. When you choose this method, Replication Manager overwrites the existing Ranger policies.

     For example, assume a Ranger policy in the destination Ranger service has *user1* and the same Ranger policy on the source cluster has *user2*. In this method, *user1* is removed and *user2* is added in the destination Ranger policy after replication.

   You can choose the ingestion method on the **Advanced** tab during Ranger replication policy creation.

Preparing clusters for Ranger replication policy creation

You must prepare the clusters before you create a Ranger replication policy in CDP Private Cloud Base Replication Manager.

About this task

Ensure that the following prerequisites are complete before you create a Ranger replication policy:

Procedure

1. Are the source and target clusters Kerberos-enabled?

   You can configure SSL/TLS certificate exchange manually on source Cloudera Manager and target Cloudera Manager. For more information, see Configuring SSL/TLS certificate exchange on Cloudera Manager instances.

   Note: You can enable Auto-TLS on the clusters, if required.
2. Have you added the source cluster as a peer to the target cluster? For more information, see Adding cluster as a peer.

   Note: Ensure that you choose the Create User With Admin Role option when you add the peer.
3. Do you want to replicate the Ranger audit logs for HDFS? If so, complete the following steps:
   a) Set the `Ranger Plugin HDFS Audit Enabled (ranger_plugin_hdfs_audit_enabled)` property to true in the Cloudera Manager Ranger service Configuration tab on the source cluster and target cluster.
   b) Enable HDFS snapshots for the Ranger audit log directory in the source cluster. The destination directory to which you replicate the Ranger policies need not be snapshottable.

      By default, the Ranger audit log directory is /ranger/audit in HDFS. During Ranger replication policy creation, you can edit the log directory path to replicate a subset of logs by appending *hdfs*, *hbase*, or *atlas* at the end of

the default path. For example, if you append *hdfs* at the end of the default path, Replication Manager replicates only the HDFS Ranger audit logs.

c) Do you have the user credentials in the *supergroup* group on the HDFS NameNode host of the target cluster? Replication Manager requires superuser credentials to replicate Ranger audit log directory.

d) Do you have the user credentials in the *supergroup* group on the HDFS NameNode host of the source cluster?

Creating Ranger replication policies

You can create Ranger replication policies in CDP Private Cloud Base Replication Manager. The Ranger replication policies copy or migrate Ranger policies for HDFS, Hive, and HBase between CDP Private Cloud Base 7.1.9 or higher clusters using Cloudera Manager 7.11.3.

Procedure

1. Go to the Cloudera Manager Replication page on the target cluster.
2. Click Create Replication Policy Ranger Replication Policy .
3. Configure the following options on the **General** tab:

Option	Description
Name	Enter a unique name for the replication policy.
Source	Choose the source cluster. Ensure that you add the source peer as an admin peer on the Cloudera Manager Replication Peers page; otherwise, the replication job fails.
Destination	Choose the target cluster.
Schedule	Choose: • Immediate to run the schedule after policy creation. • Once to run the schedule one time after policy creation. Set the date and time. • Recurring to run the schedule periodically. Set the date, time, and interval between runs.
Replicate Ranger data	Select to replicate the Ranger policies and roles for the resources you choose in the **Services** tab.
Replicate Ranger audit logs in HDFS	Select to replicate the Ranger audit logs in HDFS.
Source side HDFS audit log directory*	Shows the source Ranger HDFS audit log path by default. For example, hdfs://*[***SOURCE URL***]*:8020/ranger/audit/ You can edit the log directory path to replicate only a subset of logs by appending *hdfs*, *hbase*, or *atlas* at the end of the default path. For example, if you append *hdfs* at the end of the default path, Replication Manager replicates only the HDFS Ranger audit logs.
Destination directory*	Shows the destination Ranger HDFS audit path where the source HDFS audit logs are replicated to, by default. The default path is the /ranger/audit/replication/*[***SOURCE PEER NAME BASE64***]*/*[***SOURCE CLUSTER NAME BASE64***]*/*[***SOURCE RANGER SERVICE NAME BASE64***]*/ subdirectory. The replication folder has three Base64 encoded directories to avoid illegal HDFS characters.
Maximum Number of Copy Mappers	(Optional) Enter the maximum number of simultaneous copy mappers for DistCp to replicate Ranger audit logs in HDFS. The default value is 20.

Option	Description
Maximum Bandwidth Per Copy Mapper	(Optional) Enter the bandwidth limit for each mapper to replicate Ranger audit logs in HDFS. Default is 100 MB. The total bandwidth used by the replication policy is equal to Maximum Bandwidth multiplied by Maximum Map Slots. Therefore, you must ensure that the bandwidth and map slots you choose do not impact other tasks or network resources in the target cluster. **Tip:** The Throughput field on the Cloudera Manager Replication Policies page shows the maximum bandwidth set for the replication policy during replication policy creation.
File listing threads	(Optional) Choose the Override DistCp default option and configure the number of threads (a maximum of 128 threads) that the replication policy must use during the copylisting phase of replication. By default, Replication Manager uses the default value of 20 threads for the copylisting phase of replication. The default number of threads for the copylisting phase of replication (using replication policies) can be set in the core-site.xml or hdfs-site.xml file for the HDFS service. You can set a maximum of 128 threads only. **Important:** Increasing this value increases the load on the HDFS NameNode of the source cluster which in turn increases the network bandwidth used by the replication jobs.
MapReduce Service	Select the MapReduce or YARN service to use.
Scheduler Pool	(Optional) Enter the name of a resource pool in the field. The value you enter is used by the MapReduce Service you specified when Cloudera Manager executes the MapReduce job for the replication. The job specifies the value using one of these properties: • MapReduce – Fair scheduler: mapred.fairscheduler.pool • MapReduce – Capacity scheduler: queue.name • YARN – mapreduce.job.queuename
Run as Username	Enter the username to run the replication job. Ensure that the user is in the *supergroup* group on the HDFS NameNode host of the target cluster.
Run on Peer as Username	Enter the user if the peer cluster is configured with a different superuser. Ensure that the user is in the *supergroup* group on the HDFS NameNode host of the source cluster.
*the values for the field are derived from the `ranger_plugin_hdfs_audit_url` API.	

4. Configure the following options on the **Services** tab:

Option	Description
Source Service Names	Choose one or more service names for which you want to copy or migrate the Ranger policies. You can choose HDFS, HBase, and Hive services, and also choose the tag services. Replication Manager pairs or maps the source and destination Ranger services according to their service types.
Destination Service Names	Choose the service name on the target cluster. If there are more than one Ranger service of the same type on the target cluster, choose the required service from the drop-down list.

5. Configure the following options on the **Advanced** tab:

Option	Description
Users Mapping	Enter the usernames for the services only if the usernames defined in Ranger differ in the source and target clusters.

Option	Description
Resources Mapping	Enter the resource paths for the services only if the resource path defined in Ranger differs in the source and target clusters. **Note:** If you enter the resource paths, ensure that you choose the Override policy import strategy.
Policy Import strategy	Choose one of the following methods for file ingestion: • Merge - Replication Manager merges the Ranger policies. By default, Replication Manager uses this method. For example, assume a Ranger policy in the destination Ranger service contains *user1* and the same Ranger policy on the source cluster has *user2*. In this method, both user1 and user2 are added in the destination Ranger policy after replication. • Override - Replication Manager overwrites the existing Ranger policies. For example, assume a Ranger policy in the destination Ranger service contains *user1* and the same Ranger policy on the source cluster has *user2*. In this method, user1 is removed and user2 is added in the destination Ranger policy after replication.
Description	Optionally, you can enter a brief description.
Alerts	Choose to generate alerts for various state changes in the replication workflow. You can alert On Failure, On Start, On Success, or On Abort of the replication job. You can configure alerts to be delivered by email or sent as SNMP traps. If alerts are enabled for events, you can search for and view the alerts on the **Events** tab, even if you do not have email notification configured. For example, if you choose Command Result that contains the Failed filter on the Diagnostics Events page, the alerts related to the On Failure alert for all the replication policies for which you have set the alert appear. For more information, see Managing Alerts and Configuring Alert Delivery.

6. Click Create.

Results

The replication policy appears on the **Replication Policies** page. It can take up to 15 seconds for the task to appear.

If you selected Immediate in the **Schedule** field, the replication job starts replicating after you click Save Policy.

Managing Ranger replication policies

After you create a Ranger replication policy in CDP Private Cloud Base Replication Manager, you can perform and monitor various tasks related to the replication policy. You can view the job progress and replication logs. You can edit the advanced options to optimize a job run. You can suspend a job and also activate a suspended job.

Replication policy details

The **Replication Policies** page shows a row of information for each replication policy and the following columns for each replication policy:

- Internally generated **ID** for the replication policy. Click the column label to sort the replication policies.
- Replication policy **Name** that you provide during replication policy creation.
- Replication policy **Type**.
- **Source** cluster in the replication policy.
- **Destination cluster** in the replication policy.
- Replication job **Progress**.
- Timestamp when the last replication job **Completed.**

- Timestamp of **Next Run** of replication policy job.

Actions menu

The Actions menu provides the following options:

Action	Description
Show History	Click to open the **Replication History** page for a replication policy. The **Replication History** page shows the replication policy **Name**, replication policy **Type**, **Source** cluster name, **Destination** cluster name, and the timestamp of **Next Run** of the replication policy job. The **Replication History** page shows the following summary about replicated Ranger policies in a successful Ranger replication policy: • **Total policies** in the replication policy. • **Created** or the number of Ranger policies that were successfully replicated. • Number of Ranger policies that were not replicated and **Failed.** • Number of Ranger policies **Skipped** during replication. • Number of Ranger policies there were **Skipped due to timeout** during replication.
Edit Configuration	Click to change the replication policy options as required.
Run Now	Click to initiate the replication policy job.
Collect Diagnostic Data	Click to open the **Send Diagnostic Data** modal window, where you can collect replication-specific diagnostic data for the last 10 runs of the replication policy. In the **Send Diagnostic Data** modal window, you can perform the following steps: **1.** Select Send Diagnostic Data to Cloudera to automatically send the bundle to Cloudera Support. You can also enter a ticket number and comments when sending the bundle. **2.** Click Collect and Send Diagnostic Data for Replication Manager to generate the bundle and open the **Replication Diagnostics Command** screen. **3.** Click Download Result Data to download the ZIP file containing the bundle to your machine.
Disable | Enable	Click to disable or enable the replication policy respectively. No further replications are scheduled for disabled replication policies.
Delete	Click to remove the replication policy permanently from Replication Manager.

Troubleshooting replication policies between on-premises clusters

The troubleshooting scenarios in this topic help you to troubleshoot the replication policies that you create between on-premises clusters in Replication Manager.

How can replication policy performance be optimized when there are a large number of files to replicate?

You can configure the heap size to 16 GB using the extra Java runtime option. To accomplish this task, perform the following steps:

1. Go to the source Cloudera Manager *HDFS service* Configuration tab.
2. Locate the HDFS Replication Environment Advanced Configuration Snippet (Safety Valve) property.

3. Enter the HADOOP_OPTS="-Xmx16G" key-value pair, and save the changes.
4. Restart the HDFS service.
5. Go to the target Cloudera Manager *HDFS service* Configuration tab.
6. Locate the HDFS Replication Environment Advanced Configuration Snippet (Safety Valve) property.
7. Enter the HADOOP_OPTS="-Xmx16G" key-value pair, and save the changes.
8. Restart the HDFS service.

How can file replication tasks be equitably distributed to all mappers?

The Replication Strategy option, that you can configure during the policy creation process, takes care of file replication task distribution. By default, this option is set to Dynamic; that is Replication Manager distributes the file replication tasks in small sets to the mappers, and as each mapper completes its tasks, it dynamically acquires and processes the next unallocated set of tasks.

However, you can configure it to Static. The file replication tasks among the mappers are set upfront to achieve a uniform distribution based on the file sizes.

How to determine the number of mappers and the bandwidth per mapper required for a replication policy?

Mappers in addition to copying files also perform several tasks which include creating directories, preserving permissions and other metadata, calculating checksums, and identifying files to skip for replication. The mappers might also get throttled by the network. The following example describes a typical scenario and ways to resolve issues that might arise.

Example: A replication policy incrementally copies ~100K new/modified files and skips ~10M files every few hours. You can optimize the policy performance for on-premises to on-premises clusters by:

- Configuring the mappers based on the requirements using the Maximum Map Slots option during the policy creation process. By default, this option is set to 20.
- Choose Skip Checksum Checks during the policy creation process since the number of files that are skipped is high. This ensures that checksum checks are skipped on copied files.
- Check the **Throughput** column for the replication policy on the **Replication Policies** page for average throughput per mapper/file of all the files written. You can use more mappers with less bandwidth per mapper, if required.

  You can configure Maximum Bandwidth to limit the bandwidth per mapper during the policy creation process. By default, this is set to 100 MB.

Why should you consider creating multiple replication policies instead of one replication policy?

You must consider creating multiple replication policies instead of one replication policy to replicate all the directories and files in a cluster because:

- the performance improves if you run multiple replication policies at once in parallel.
- reliability can be ensured even if a replication policy fails.
- you have the flexibility to run the replication policies with less resources and at different intervals.

How many replication policies can be run in parallel?

You can run several replication policies in parallel depending on the following factors:

- Number of available mappers
- Available network bandwidth
- Load on source and target NameNodes
- Read bandwidth on source DataNodes and write bandwidth of target DataNodes

It is recommended that you go for the lower side of these limits so that the other applications are also able to access these resources successfully. You can decide the number of concurrent replications depending on the available number of mappers and network bandwidth. For example, if you have a 10 GBps network, you might want to run

five replication policies with 20 mappers each in parallel rather than one replication policy with 100 mappers and 100 MBps bandwidth per mapper.

You might want to monitor the write speed on the target cluster if the total bandwidth is more than 100 GBps and you are utilizing all the available bandwidth for the replication policy jobs. This is because the target DataNodes require 3x (or the configured replication factor) write bandwidth for write operations.

Why use the YARN resource pool for replication policy jobs?

Replication Manager uses MapReduce or YARN framework for its replication jobs and the jobs use 20 mappers and a maximum of 100 MB/s network bandwidth utilization by default. You can change this based on the size of the clusters and total data or resources that you want to assign to the replication policy jobs.

It is recommended that you use a YARN resource pool to configure the percentage of resources you want to assign to the replication jobs. This ensures that the replication policy jobs do not consume more than the assigned percentage of resources. You can also configure isolation of resources by specifying which users can use certain resources.

To configure a new YARN resource pool, go to the Cloudera Manager Clusters YARN service Resource Pools *(Tab)* Configuration Create Resource Pool tab.

To use the configured resource pool in a replication policy, go to the Cloudera Manager Replication Policies Actions Edit Configuration Resources *(Tab)* Scheduler Pool field, and enter the YARN resource pool name.

What happens to the replication policies when an active Cloudera Manager instance fails over to the passive Cloudera Manager instance?

During the time duration when Cloudera Manager fails over a passive instance, the previously active Cloudera Manager instance is not up and the local temporary folder on the previously active Cloudera Manager host) used by replication policies becomes inaccessible for the currently active Cloudera Manager instance. Therefore, the replication policies that have a Cloudera Manager peer associated to it (Hive external replication policies and HDFS replication policies between on-premises to on-premises clusters) fail if they are initiated during that time duration. Subsequent runs of the same policy in the absence of a failover event eventually succeed.

To avoid these issues, you can implement the following solutions based on the scenarios:

- Controlled or planned Cloudera Manager failover - In this scenario, you can stop or pause existing replication policy job run. You might want to postpone creating any replication policies during the failover time duration.
- Unplanned failover - In this scenario, you can use one of the following methods:
  - Re-run the failed replication policies.
  - Wait for the next planned replication policy run.
  - Restore the replicated content to a previous snapshot and re-run the replication policy.

When the HDFS incremental replication fails for an HDFS replication policy, the next policy run starts a full bootstrap replication. How can this issue be mitigated?

When an HDFS replication policy (incremental replication) fails, the last successfully replicated snapshot gets deleted. Therefore, the next policy run starts a full bootstrap replication. For large datasets, the bootstrap replication takes a long time to complete.

To mitigate this issue, set the deleteLatestSourceSnapshotOnJobFailure flag to false using REST APIs for the replication policy. After you set the flag to false, the last replicated snapshot is not deleted even when the replication fails. Therefore, the next policy run is an incremental run.

How to resolve replication policies that fail with the “Custom keytab configuration is required for this service” error?

This error appears for replication policies that use Kerberos-enabled clusters on Isilon storage.

To mitigate this issue, perform the following steps:

1. Create a custom Kerberos keytab and Kerberos principal that the replication jobs can use to authenticate to storage and other CDP services.
2. Go to the target Cloudera Manager Administration Settings page.
3. Search for the following properties, and enter the required values:
   - Custom Kerberos Keytab Location – Enter the location of the custom Kerberos keytab.
   - Custom Kerberos Principal Name – Enter the principal name to use for replication between secure clusters.

   For more information about the parameters, see Cloudera Manager Server Properties for replication.

   Important: To replicate data using replication policies that use Kerberos-enabled clusters on Isilon storage, you must:

   - ensure that the source and target clusters have the same set of users and groups. When you set the ownership of files (or when maintaining ownership), if a user or group does not exist, the `chown` command fails on Isilon. For more information, see Performance and Scalability Limitations.
   - enter the Custom Kerberos Principal Name value in the **Run As Username** field during the replication policy creation process.

   Cloudera recommends that you do not select the Replicate Impala Metadata option for Hive/Impala replication policies. To use this feature, create a custom principal in the hdfs/hostname@realm or impala/hostname@realm format.
4. Add the hadoop.security.token.service.use_ip = false key-value pair to the HDFS Service Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml and Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml properties.

Tip:

If the replication MapReduce job fails and the following error appears, set the Isilon cluster-wide time-to-live setting to a higher value on the target cluster:

```
java.io.IOException: Failed on local exception: java.io.IOException:
  org.apache.hadoop.security.AccessControlException:
  Client cannot authenticate via:[TOKEN, KERBEROS];
  Host Details : local host is: "foo.mycompany.com/172.1.2.3";
  destination host is: "myisilon-1.mycompany.com":8020;
```

A higher value might cause workloads to be less distributed which might affect the load balancing in the Isilon cluster. To mitigate this issue, you can use a value of 60 as a good starting point. For example, the `isi networks modify pool subnet4:nn4 --ttl=60` command configures the Isilon cluster-wide time-to-live setting to 60.

To view the settings for a subnet, you can run the `isi networks list pools --subnet subnet3 -v` command.

Snapshots and snapshot policies

You can create HDFS, HBase, and Ozone snapshots using Replication manager in CDP Private Cloud Base for data replication. Learn what data is backed up during replication and the methods available for replication.

What HDFS, HBase, and Ozone snapshots are

HDFS, HBase, and Ozone snapshots are point-in-time backups of HBase tables, HDFS directories, and Ozone buckets respectively. You can create HDFS, HBase, or Ozone snapshots in Cloudera Manager or using the command line as required. You can also create them at regular intervals using snapshot policies in CDP Private Cloud Base Replication Manager. HDFS and Hive replication policies leverage HDFS snapshots and Ozone replication policies leverage Ozone snapshots to implement incremental data replication. You can improve the reliability of replication policies by using snapshots.

HBase snapshots for tables and Ozone snapshots for buckets are enabled by default. However, you must enable HDFS snapshots for the required HDFS directories and subdirectories in Cloudera Manager.

Replication methods used by Replication Manager

The first HDFS, Hive, or Ozone replication policy job is a bootstrap job, that is the replication policy replicates all the data in the specified HDFS directories, Hive/Impala tables, or Ozone buckets respectively. Subsequent replication jobs use one of the following methods to replicate data:

Incremental replication method

In this method, Replication Manager uses the diff report to replicate data. The snapshot diff feature uses snapshots to generate the diff report to determine the changed or new data in the chosen directories or buckets in the source cluster. This method optimizes the replication jobs by using less time and resources during replication.

Non-incremental method

Replication Manager uses this method if the snapshot diff fails. In this method, Replication Manager performs the following high-level steps:

1. Lists all the files.
2. Performs a checksum and metadata check on them to identify the relevant files to copy. This step depends on the advanced options you choose during the replication creation process. During this identification process, some unchanged files are skipped if they do not meet the criteria set by the chosen advanced options.
3. Copies the identified files from the source cluster to the target cluster.

You can create snapshot policies in CDP Private Cloud Base Replication Manager that define the HDFS directories, HBase tables, or Ozone buckets to be snapshotted, the intervals to take snapshots, and the number of snapshots to retain for each snapshot interval. For example, you can create a snapshot policy that takes daily and weekly snapshots, and also specify that only seven daily snapshots and five weekly snapshots must be maintained.

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

How Replication Manager uses snapshots

You can create snapshot policies in CDP Private Cloud Base Replication Manager to take HDFS and Ozone snapshots at regular intervals. HDFS and Hive replication policies leverage HDFS snapshots and Ozone replication policies leverage Ozone snapshots to implement incremental data replication.

You can also create HBase snapshot policies to create HBase snapshots at regular intervals in Replication Manager. There are several use cases that leverage HBase snapshots. For more information, see HBase snapshot use cases.

HBase snapshots are enabled for all HBase tables by default. HBase snapshots are point-in-time backup of tables, without making data copies, and with minimal impact on RegionServers. HBase snapshots are supported for clusters running CDH 4.2 or higher. You can also create an HBase snapshot using Cloudera Operational Database (COD) CLI.

HDFS snapshots

Understand what HDFS snapshots are and how it helps Replication Manager during replication.

HDFS snapshots are point-in-time backup of directories without actually cloning of data. HDFS snapshots improve data replication performance and prevent errors caused by changes to a source directory. These snapshots appear on the filesystem as read-only directories that can be accessed just like other ordinary directories.

A directory is called *snapshottable* after it has been enabled for snapshots, or if a parent directory is enabled for snapshots. Subdirectories of a snapshottable directory are included in the snapshot.

Note: Cloudera recommends that you enable snapshots only the required HDFS directories. This is because when you enable snapshots for unwanted directories or which represents the entire HDFS system, snapshots of unwanted files and directories such as tmp and trash directories are also taken. These large snapshots consume memory and network resources, and might increase the server load on Namenode. The replication jobs also become inefficient because the job replicates the unwanted files and metadata. Additionally, the unwanted files do not get deleted until their snapshots are deleted.

For more information, see the HDFS Snapshots Best Practices blog.

Some replications, especially those that require a long time to finish can fail because source files are modified during the replication process. You can prevent such failures by using snapshot policies in Replication Manager. This use of snapshots is automatic with CDH versions 5.0 and higher. To take advantage of this, you must enable the relevant directories for snapshots (also called making the directory *snapshottable*).

When the replication job runs, it checks to see whether the specified source directory is snapshottable. Before replicating any files, the replication job creates point-in-time snapshots of these directories and uses them as the source for file copies. This ensures that the replicated data is consistent with the source data as of the start of the replication job. The latest snapshot for the subsequent runs is retained after the replication process is completed.

For more information, see Using HDFS snapshots.

Hive/Impala replication using snapshots

Before you create Hive external table replication policies, ensure that you enable snapshots for the databases and directories that contain the required external tables. Before you replicate Impala tables, ensure that the storage locations for the tables and associated databases are also snapshottable.

For example, if the database resides in a custom location, such as /apps/folder1/folder2/[sales.db, marketing.db, hr.db, etc.], you can enable the snapshots at the following database or directory levels depending on your requirement:

- /apps/folder1/folder2/sales.db
- /apps/folder1/folder2/marketing.db
- /apps/folder1/folder2/hr.db

Note: If you enable snapshots at the /apps, /apps/folder1, or /apps/folder1/folder2 level, large snapshots are created which might create performance and snapshot-related issues.

You can also isolate the database-level snapshots from each other so that the Hive external table replication policy replicates only the specified database.

The following table shows sample custom locations that contain the external tables and the recommended directory level to enable snapshots to isolate the database-level snapshots:

Sample custom location of external tables	Recommended directory level to enable snapshots
/data/folder1/folder2/sales/[table1, table2, table3 ... tablen]	/data/folder1/folder2/sales
/data/folder1/folder2/marketing/[table1, table2, table3 ... tablen]	/data/folder1/folder2/marketing
/data/folder1/folder2/hr/[table1, table2, table3 ... tablen]	/data/folder1/folder2/hr

Orphaned snapshots

When you edit or delete a snapshot policy, the snapshots for the files, directories, or tables that were removed from the snapshot policy are retained. These are known as *orphaned* snapshots. These snapshots are not deleted automatically because they are no longer associated with a snapshot policy.

You can identify and delete these orphaned snapshots manually through Cloudera Manager, or by creating a command-line script that uses the HDFS or HBase snapshot commands.

To avoid orphaned snapshots, you can choose one of the following methods depending on your requirements.

- Delete the snapshots before you edit or delete the associated snapshot policy.

  Cloudera Manager assigns the prefix cm-auto which is followed by a globally unique identifier (GUID) for every HDFS snapshot policy. You can view the snapshot prefix in the policy summary in the policy list, and in the delete modal window.

  Note: Before you delete a snapshot policy, ensure that you record the snapshot names in the snapshot policy and the cm-auto-guid of the snapshot policy. This is because you cannot determine the snapshot names in the snapshot policy and the cm-auto-guid of the snapshot policy after you delete the snapshot policy, and the snapshot names also do not contain any recognizable references to its snapshot policy.
- Identify the orphaned snapshots for a deleted snapshot policy using its cm-auto-guid, and delete the snapshots.

Ozone snapshots and replication methods

Understand what Ozone snapshots are and what you can replicate with Ozone snapshots. Also, learn about the replication methods you can choose for Ozone replication policies to replicate data.

What Ozone snapshots are

Ozone snapshots are point-in-time backups of buckets and volumes within it, without actually cloning the data. You can leverage snapshots and snapshot-diffs to implement incremental replication in Ozone replication policies.

Ozone data replication methods

Ozone snapshots are enabled by default for all the buckets and volumes. If the incremental replication feature is also enabled on the source and target clusters, you can choose one of the following methods to replicate Ozone data during the Ozone replication policy creation process:

Full file listing

By default, the Ozone replication policies use the full file listing method which takes a longer time to replicate data. In this method, the first Ozone replication policy job run is a bootstrap job; that is, all the data in the chosen buckets are replicated. During subsequent replication policy runs, Replication Manager performs the following high-level steps:

1. Lists all the files.
2. Performs a checksum and metadata check on them to identify the relevant files to copy. This step depends on the advanced options you choose during the replication creation process. During this identification process, some unchanged files are skipped if they do not meet the criteria set by the chosen advanced options.
3. Copies the identified files from the source cluster to the target cluster.

Incremental only

In this method, the first replication policy job run is a bootstrap job, and subsequent job runs are incremental jobs.

To perform the incremental job, Replication Manager leverages Ozone snapshots and the snapshot-diff capability to generate a diff report. The diff report contains the changed or new data from the source cluster. The subsequent replication policy replicates data based on the diff report.

Incremental with fallback to full file listing

In this method, the first replication policy job run is a bootstrap job, and subsequent job runs are incremental jobs. However, if the snapshot-diff fails during a replication policy job run, the next job run is a full file listing run. After the full file listing run succeeds, the subsequent runs are incremental runs. This method takes a longer time to replicate data if the replication policy job falls back to the full file listing method.

Creating snapshot policies in Replication Manager

You can create HDFS, HBase, and Ozone snapshot policies in CDP Private Cloud Base Replication Manager.

Before you begin

HBase and Ozone snapshots are enabled by default on the tables and buckets respectively. Ensure that you have enabled snapshots for the required HDFS directories in Cloudera Manager before you create HDFS snapshot policies.

Procedure

1. Go to the Cloudera Manager Replication Snapshot Policies page.
2. In the Create Snapshot Policy modal window, enter the following generic options that are common for HBase, HDFS, and Ozone snapshot policies:

Option	Description
Service	Choose HDFS, HBase, or Ozone depending on your requirements.
Name	Enter a name for the snapshot policy. Ensure that the snapshot policy name neither contains the characters % . ; / \ nor any character that is not ASCII printable, which includes the ASCII characters less than 32 and the ASCII characters that are greater than or equal to 127.
Description	Optional. Enter a brief description about the snapshot policy.
Schedule	Choose one or all the following frequency options and then specify the other granularity details to take snapshots depending on your requirements: • Hourly • Daily • Weekly • Monthly • Yearly You can schedule snapshots hourly, daily, weekly, monthly, or yearly, or any combination of those. Depending on the frequency you select, you can specify the time of day to take the snapshot, the day of the week, day of the month, or month of the year, and the number of snapshots to keep at each interval. Each time unit in the schedule information is shared with the time units of larger granularity. That is, the minute value is shared by all the selected schedules, hour by all the schedules for which hour is applicable, and so on. For example, if you specify that hourly snapshots are taken at the half hour, and daily snapshots taken at the hour 20, the daily snapshot will occur at 20:30. When a snapshot policy includes a limit on the number of snapshots to keep, Cloudera Manager checks the total number of stored snapshots each time a new snapshot is added, and automatically deletes the oldest existing snapshot.
Alerts	Choose one of the following snapshot policy state changes to generate alerts during snapshot creation: • On failure • On start • On success • On abort

3. When you choose the HDFS Service, you must specify the HDFS directories to include in the snapshot.

Important: Do not take snapshots of the root directory. Select the paths of the directories to include in the snapshot. The drop-down list enables you to select only the directories that are enabled for snapshotting. If no directories are enabled for snapshotting, the Click + to add a path and − to remove a path. warning appears.

4. When you choose the HBase Service, you must list the tables to include in your snapshot.

   You can use a Java regular expression to specify a set of tables. For example, finance.* matches all tables with names starting with finance. You can also create a snapshot for all tables in a given namespace, using the {*[***NAMESPACE***]*}:.* syntax.
5. When you choose the Ozone Service, you must specify the list of buckets and volumes to include in your snapshot.
6. Click Save Policy.

Results

The snapshot policy appears on the Snapshot Policies page.

Manage and monitor snapshot policies

After you create an HDFS, HBase, or Ozone snapshot policy, you can manage the snapshot policies on the “Snapshot Policies” page. You can view more details about the snapshot policy on the "Snapshot History" page.

The **Snapshot Policies** page in CDP Private Cloud Base Replication Manager shows the list of snapshot policies, filters to view the snapshot policies, and the Create Snapshot Policy option to create HDFS, HBase, and Ozone snapshot policies.

Snapshot policy details

The following snapshot policy details appear on the Cloudera Manager Replication Snapshot Policies page:

Option	Description
Policy Name	Shows the snapshot policy you provided during snapshot policy creation.
Cluster	Shows the cluster where the snapshots reside.
Service	Shows the service as HDFS, HBase, or Ozone for the snapshot policy.
Directories/Tables/Objects	Shows the directories, tables, or buckets and volumes you chose for the snapshot policy.
Last Run	Shows the timestamp for the last snapshot policy run.
Snapshot Schedule	Shows the frequency you chose for the snapshot policy.

You can use the CLUSTER, SERVICE, and SCHEDULE filters to view the required snapshot policies on the **Snapshot Policies** page.

Actions menu

You can perform the following actions on a snapshot policy on the Cloudera Manager Replication Snapshot Policies page:

- Click Actions Show History for a snapshot policy to view the existing snapshots generated by the snapshot policy.
- Click Actions Edit Configuration to modify the schedule of the snapshot policy. You can also modify the tables, directories, or buckets and volumes in the snapshot policy.
- Click Actions Disable to disable the snapshot policy.
- Click Actions Delete to delete the snapshot policy.

Snapshots History details in Replication Manager

You might need to view details about all the snapshot job runs that were run or attempted for a snapshot policy to manage job runs or to perform troubleshooting activities. The "Snapshots History" page in CDP Private Cloud Base Replication Manager provides these details.

You can view the snapshot **Policy Name**, **Type** of the snapshot policy, **Cluster** where the snapshot policy resides, **Service**, and the automatically assigned **Snapshot Prefix** for all the snapshot job runs in the snapshot policy.

The **Snapshots History** page shows a table of snapshot jobs and the following columns for the snapshot jobs:

Table 5: Snapshots History

Column	Description
Start Time	Time when the snapshot job started. Click View to open the **Managed scheduled snapshots Command** page, which displays details and messages about each step in the command run.
Outcome	Status of snapshot policy as succeeded or failed.
Paths | Tables | Buckets Processed	Shows the number of **Paths Processed** for the HDFS snapshot policy job; the number of **Tables Processed** for the HBase snapshot policy job; the number of **Buckets Processed** for the Ozone snapshot policy job.
Paths | Tables | Buckets Unprocessed	Shows the number of **Paths Unprocessed** for the HDFS snapshot policy job; the number of **Tables Unprocessed** for the HBase snapshot policy job; the number of **Buckets Unprocessed** for the Ozone snapshot policy job.
Snapshots Created	Number of snapshots created.
Snapshots Deleted	Number of snapshots deleted.
Errors During Creation	Displays a list of errors that occurred when creating the snapshot. Each error shows the related path and the error message.
Errors During Deletion	Displays a list of errors that occurred when deleting the snapshot. Each error shows the related path and the error message.

When you expand a snapshot job, you can view more details about the job. Click View to see the **Command Details** about the job on the **All Recent Commands** page.

You can view the following snapshots in the specified cluster in Cloudera Manager:

- HDFS snapshots on the Cloudera Manager Clusters *[***HDFS SERVICE***]* File Browser tab.
- HBase snapshots on the Cloudera Manager Clusters *[***HBASE SERVICE***]* Table Browser tab.
- Ozone snapshots on the Cloudera Manager Clusters *[***OZONE SERVICE***]* Bucket Browser tab.

Troubleshooting snapshot policies in Replication Manager

You must be aware of the issues related to the snapshot policies that you create between on-premises clusters in Replication Manager and how to resolve those issues.

Error when snapshot policy is edited or deleted

Errors might appear when you edit or delete a snapshot policy that contains % . ; / \ or any character that is not ASCII printable which includes the ASCII characters less than 32 and the ASCII characters that are greater than or equal to 127.

To resolve this issue, use the update command to replace the unsupported character in the policy name with an underscore, in the SNAPSHOT_POLICIES table.

To update the snapshot policy name in the SNAPSHOT_POLICIES table, perform the following steps:

1. Take a backup of the Cloudera Manager database.
2. Run the update SNAPSHOT_POLICIES set NAME = replace(NAME,CHAR(*[***ENTER CHARACTER NUMBER***]*),'_'); command to replace the unsupported character in the snapshot policy name with an underscore.

Error when target directory is out-of-sync

The

```
target has been modified since snapshot [***SNAPSHOT NAME**]
```

error might appear when Cloudera Manager HA (planned or unplanned failover) is initiated.

This error appears when the target directory is out-of-sync with the source directory during a planned or unplanned failover. The directories might go out-of-sync if the files or folders in the target directory were modified or the source directory changed and a snapshot was created to make the synchronization point. The modifications on the target cluster include creating or dropping databases, tables, and partitions in Hive.

To mitigate this issue, you can run the replication policy which bootstraps the target. However, this action might modify the target cluster and might result in data loss. In addition, the time and resources consumed to complete the bootstrap replication job is high.

However, as a safety measure if you created a snapshot (for example, sync_v1) for the target directory before you created the HDFS or Hive replication policy, then you can run the following steps to initiate the incremental replication from the source directory to target directory, which optimizes the replication job by using less time and resources:

1. Take a snapshot of the target cluster by using the hdfs dfs -createSnapshot *[***TARGET DIRECTORY***] [***NEW SNAPSHOT NAME***]* command. Ensure that you use an alternate snapshot naming pattern to distinguish this snapshot from the rest of the snapshots.

   For example, hdfs dfs -createSnapshot tar_directory mod_v1
2. Revert the target directory to the snapshot you created initially using the hadoop distcp -rdiff *[***INITIAL SNAPSHOT***] [***NEW SNAPSHOT NAME***]* -update *[***SOURCE DIRECTORY***] [***TARGET DIRECTORY***]* command.

   For example, hadoop distcp -rdiff mod_v1 sync_v1 -update src_directory tar_directory
3. Run the replication policy to sync the source and target directories.

Restoring HDFS snapshots in Cloudera Manager

Before you restore an HDFS directory from an HDFS snapshot, ensure that there is adequate disk space.

1. Go to the Cloudera Manager *HDFS service* File Browser tab.
2. Go to the directory you want to restore.
3. Click the drop-down menu next to the full file path (to the right of the file browser listings) and select one of the following:

   - Restore Directory From Snapshot
   - Restore Directory From Snapshot As...

   The Restore Snapshot dialog box appears.
4. Select Restore Directory From Snapshot As... if you want to restore the snapshot to a different directory. Enter the directory path to which the snapshot has to be restored. Ensure that there is enough space on HDFS to restore the files from the snapshot.

   Note: If you enter an existing directory path in the Restore Directory From Snapshot As... field, the directory is overwritten.
5. Select one of the following:

   - Use HDFS 'copy' command - This option runs the restore job slowly and does not require credentials in a secure cluster. It copies the contents of the snapshot as a subdirectory or as files within the target directory.
   - Use DistCp / MapReduce - This option runs the restore job faster and requires credentials (Run As) in secure clusters. It merges the target directory with the contents of the source snapshot. When you select this option,

the following additional fields, which are similar to those available when configuring a replication policy appear under More Options:

- When restoring HDFS data, if a MapReduce or YARN service is present in the cluster, the DistributedCopy (DistCp) job is used to restore directories, increasing the speed of restoration. You can choose MapReduce or YARN as the MapReduce service. For files, if a MapReduce or YARN service is not present, a normal copy is performed.
- Skip Checksum Checks - Determines whether to skip checksum checks (the default is to perform them). If checked, checksum validation is not performed.

  You must select the this property to prevent failure when restoring snapshots in the following cases:

  - Restoring a snapshot within a single encryption zone.
  - Restoring a snapshot from one encryption zone to a different encryption zone.
  - Restoring a snapshot from an unencrypted zone to an encrypted zone.

Restoring Ozone snapshots in Cloudera Manager

You can restore an Ozone snapshot to a previous version or restore the snapshot to another bucket and volume in Cloudera Manager. You can also delete a snapshot.

About this task

Ozone snapshot restore performs the following actions while restoring the Ozone snapshots:

- Overwrites the keys in the destination bucket if they are also available in the snapshot.
- Retains the keys in the destination bucket if they are not in the snapshot.
- Creates the keys that are not in the destination bucket but are available in the snapshot.

Procedure

1. Go to the Cloudera Manager Clusters *[***OZONE SERVICE***]* Bucket Browser tab.
2. Enter the Volume and Bucket, and click Go to bucket.

   On the **Bucket Browser** tab, you can:

   - View and browse the list of all the available volumes in the Ozone service.

     Click a volume in the **Name** column, or enter the name of a specific **Volume** and **Bucket** to see the list of files and directories in it. Optionally, you can Filter the results.
   - Access the **Snapshots** section where you can:

     - View the list of snapshots available for the bucket.
     - Create Snapshot instantly.
     - Restore Bucket from Snapshot.
     - Restore Bucket From Snapshot As… to a different bucket, or restore it to another volume and bucket.
     - Click Actions Delete for a snapshot to delete it permanently.

3. You can perform the following tasks depending on your requirements:
   a) To restore a bucket to its previous version, perform the following steps:
      1. Click Restore Bucket From Snapshot.
      2. Select the Snapshot to which you want to restore the volume and bucket in the **Restore Snapshot** modal window.
      3. Click Restore.

         The **Restore Ozone snapshot** modal window appears where you can view the commands which restore the volume and bucket to the specified snapshot.
   b) To restore a bucket using a snapshot to another location, perform the following steps:
      1. Click Restore Bucket From Snapshot As….
      2. Enter Destination volume and Destination bucket.
      3. Choose the Snapshot that you want to use to update the selected volume and bucket.

         Note: If the volume and bucket exists, this action overwrites similar existing data on the selected volume and bucket. If the specified volume and bucket do not exist, this action creates the volume and bucket and updates it with the data depending on the snapshot you choose.

Managing HDFS snapshots in Cloudera Manager

You can manage HDFS snapshots using Cloudera Manager or the command line.

For HDFS services, use the File Browser tab to view the HDFS directories associated with a service on your cluster. You can view the currently saved snapshots for your files. You can also delete or restore snapshots.

On the HDFS File Browser tab, you can:

- designate HDFS directories to be "snapshottable" so snapshots can be created for those directories.
- initiate immediate (unscheduled) snapshots of an HDFS directory.
- view the list of saved snapshots currently being maintained. These can include one-off immediate snapshots, as well as scheduled policy-based snapshots.
- delete a saved snapshot.
- restore an HDFS directory or file from a saved snapshot.
- restore an HDFS directory or file from a saved snapshot to a new directory or file (Restore As).

Before using snapshots, note the following limitations:

- Snapshots that include encrypted directories cannot be restored outside of the zone within which they were created.
- The Cloudera Manager Admin Console cannot perform snapshot operations (such as create, restore, and delete) for HDFS paths with encryption-at-rest enabled. This limitation only affects the Cloudera Manager Admin Console and does not affect CDH command-line tools or actions not performed by the Admin Console, such as Replication Manager which uses command-line tools. For more information about snapshot operations, see Apache HDFS snapshots documentation.

Browse HDFS directories

You can browse through the HDFS directories to select the right cluster.

To browse the HDFS directories to view snapshot activity, go to the Cloudera Manager *HDFS service* File Browser tab.

As you browse the directory structure of your HDFS, basic information (owner, group, and so on) about the directory you have selected appears.

Enabling and disabling snapshots for HDFS directories

For snapshots to be created, HDFS directories must be enabled for snapshots. You cannot specify a directory as part of a snapshot policy unless it has been enabled for snapshots.

Before you begin

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

Procedure

1. Go to the Cloudera Manager *HDFS service* File Browser tab.
2. Go to the directory you want to enable for snapshots.
3. Click the drop-down menu next to the full file path and select Enable Snapshots.

   Note: Once you enable snapshots for a directory, you cannot enable snapshots on any of its subdirectories. Snapshots can be taken only on directories that have snapshots enabled.

4. Click Disable Snapshots to disable snapshots for a directory that has snapshots enabled.

   Important: If snapshots of the directory exist, they must be deleted before snapshots can be disabled.

Taking and deleting HDFS snapshots

To take HDFS snapshots for a directory, you must first enable snapshots for the HDFS directory.

About this task

Note: You can also schedule snapshots to occur regularly by creating a snapshot policy in Replication Manager.

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

Procedure

1. Go to the Cloudera Manager *HDFS service* File Browser tab.
2. To take a snapshot of a directory, perform the following steps:
   a) Go to the directory with the snapshot you want take snapshots.
   b) Click the drop-down menu next to the full path name, and select Take Snapshot.
   c) Enter a name for the snapshot and then click OK in the Take Snapshot dialog box.
   The snapshot is added to the snapshot list.
3. To delete a snapshot for a directory, perform the following steps:
   a) Go to the directory with the snapshot you want to delete.
   b) In the list of snapshots, locate the snapshot you want to delete and click .
   c) Select Delete.

Using DistCp to migrate HDFS data from HDP cluster to CDP Private Cloud Base cluster

You can migrate data stored in HDFS from a secure HDP cluster to a secure or unsecure CDP Private Cloud Base cluster using the Hadoop DistCp tool.

Ensure that you have one of the following user accounts before you run Hadoop DistCp jobs:

- HDFS superuser - For information about creating a HDFS superuser, see Create the HDFS superuser.
- User named hdfs - By default, the hdfs user is not allowed to run YARN jobs. You must enable the hdfs user to run YARN jobs on both the clusters.

For more information about using DistCp, see Ports Used by DistCp, Distcp between Secure Clusters in Different Kerberos Realms, and Using DistCp to Copy Files.

Migrating data from secure HDP cluster to unsecure CDP Private Cloud Base cluster using DistCp

Before you run DistCp to migrate data from a secure HDP cluster to an unsecure CDP Private Cloud Base cluster, you must allow the hdfs user to run the YARN jobs on the HDP cluster in the absence of HDFS superuser account. You must also ensure that the realm name is skipped during replication and only the specified user has access to the HDP cluster.

About this task

Perform the following steps to migrate HDFS data from a secure HDP cluster to an unsecure CDP Private Cloud Base cluster:

Enabling the hdfs user to run the YARN jobs on the HDP cluster

You must make configuration changes to enable the hdfs user to run YARN jobs on the HDP cluster.

About this task

In the HDP cluster, perform the following steps on the Ambari host:

Procedure

1. Open the following file:
   /var/lib/ambari-server/resources/common-services/YARN/2.1.0.2.0/package/templates/container-executor.cfg.j2
2. Remove the hdfs entry from banned-users list and save the file.
   Sample file contents:

```
yarn.nodemanager.local-dirs={{nm_local_dirs}}
yarn.nodemanager.log-dirs={{nm_log_dirs}}
yarn.nodemanager.linux-container-executor.group={{yarn_executor_containe
r_group}}
banned.users=yarn,hdfs,mapred,bin
min.user.id={{min_user_id}}
```

3. On the YARN configuration page, verify whether the container-executor configuration template contains hdfs in the banned.users list.
4. If hdfs is listed in the banner.users list, remove it from the template and save the template.
5. Restart the following services:
   - Stale services, if any.
   - Ambari server
   - Ambari agent on each host of the cluster.
6. In the yarn.admin.acl file, add hdfs.
7. In the etc/hadoop/capacity-scheduler.xml fileSearch file, append hdfs to the yarn.scheduler.capacity.root.acl_submit_applications property.
8. Restart the YARN service.
9. Run the kinit command with the hdfs user’s keytab file to authenticate the hdfs user to the Key Distribution Center (KDC).

What to do next

Make the necessary configuration changes on the CDP Private Cloud Base cluster.

Configuration changes on the CDP Private Cloud Base cluster

During replication, the realm name must be skipped and only the specified user must have access to the HDP cluster.

Procedure

1. On the CDP Private Cloud Base cluster, the administrator must update the hadoop.security.auth_to_local configuration property based on the HDFS Kerberos principal name.

   For example, if the HDFS Kerberos principal name is hdfs@EXAMPLE.COM on the HDP cluster, then the administrator must update the hadoop.security.auth_to_local configuration property to the following value:

   RULE:[1:$1@$0](.*@EXAMPLE.COM)s/@.*//
2. Restart the stale services.

What to do next

Run the DistCp job on the HDP cluster.

Running the DistCp job on the HDP cluster

After you enable the hdfs user to run YARN jobs on the HDP cluster and make the required configuration changes on the CDP Private Cloud Base cluster, you can run the DistCp job to migrate the HDFS data from the secure HDP cluster to the unsecure CDP Private Cloud Base cluster.

Procedure

1. Make sure that you restart the cluster services before you run the DistCp job in the HDP cluster.
2. Run the following hadoop distcp command:

```
hadoop distcp -D ipc.client.fallback-to-simple-auth-allowed=true
 [***SOURCE CLUSTER***]
          [***DESTINATION CLUSTER***]
```

   For example,

```
 hadoop distcp -D ipc.client.fallback-to-simple-auth-allowed=true
        hdfs://172.27.28.200:8020/tmp/test/hosts1
    hdfs://172.27.110.198:8020/tmp/hosts1
```

   Note: A Hadoop Distcp job requires simple authentication, therefore you must run the hadoop distcp command with the ipc.client.fallback-to-simple-auth-allowed option set to true.

Migrating data from secure HDP cluster to secure CDP Private Cloud Base cluster

You can use the DistCp tool to migrate HDFS data from a secure HDP cluster to a secure CDP Private Cloud Base cluster. To migrate data, you must configure the HDP and CDP Private Cloud Base clusters on the same Active Directory (AD) KDC, set up a one-way or two-way trust between them, and then run a DistCp command to copy data.

About this task

Perform the following steps to migrate HDFS data from a secure HDP cluster to an secure CDP Private Cloud Base cluster:

Configuration changes on HDP cluster and CDP Private Cloud Base cluster

You must make some configuration changes on the HDP cluster and CDP Private Cloud Base cluster before you migrate the data from the HDP cluster to a CDP Private Cloud Base cluster.

Procedure

1. On the HDP cluster, open the core-site.xml file, enter the following properties, and save the file:

```
<property>
    <name>hadoop.security.auth_to_local</name>
    <value><RM mapping rules for HDP></value>
    <value><RM mapping rules for CDH></value>
    <description>Maps kerberos principals to local user names</description
>
</property>
```

2. On the HDP cluster, open the hdfs-site.xml file, enter the following property, and save the file:

```
<property>
  <name>dfs.namenode.kerberos.principal.pattern</name>
  <value>*</value>
</property>
```

3. Perform the above steps on the CDP Private Cloud Base cluster.
4. Create a common Kerberos principal name on both the clusters.
5. Assign the created Kerberos principal name to all the applicable NameNodes in the source and destination clusters.
6. To ensure that the same ResourceManager mapping rules are used in both the clusters, update the ResourceManager mapping rules as shown below on both the clusters:

```
<property>
    <name>hadoop.security.auth_to_local</name>
    <value>
      <HDP mapping rules>
      <CDH mapping rules>
      DEFAULT
    </value>
</property>
```

7. Configure a one-way or two-way trust between the clusters.

   To set a two-way trust between the HDP cluster and CDP Private Cloud Base cluster, perform the following steps:

   a) Create clusters that belong to different Kerberos realms.

      For example, assume that you have Realm: “DRT” for the target cluster and Realm: “DRS” for the source cluster.

   b) Set up /etc/krb5.conf on all the hosts for both the source and target hosts:

      1. [realms] section - Add both the DRS and DRT realms, DRS from the source cluster's Kerberos KDC, admin_server, and default_domain settings.
      2. [domain_realm] section - Add all the hosts of both source and target clusters.
      3. Add krbtgt/DRS@DRT principal on both the source and target hosts that have HDFS NameNode role. To accomplish this task, perform the following steps:

```
$ sudo kadmin.local
kadmin.local: addprinc -pw cloudera krbtgt/DRS@DRT
WARNING: no policy specified for krbtgt/DRS@DRT; defaulting to no
 policy
Principal "krbtgt/DRS@DRT" created
```

```
kadmin.local: listprincs
```

c) In Cloudera Manager and Ambari, perform the following steps:

1. Enable DRT as Trusted Kerberos Realm in source cluster HDFS service's configuration.
2. Enable DRS as Trusted Kerberos Realm (trusted_realm) in target cluster's configuration along with the source host name where HDFS NameNode role is present.
3. Enable DRS as Trusted Kerberos Realm in target cluster HDFS service's configuration.
4. Access the remote HDFS endpoint to verify whether the trust setup is successful. To access the remote HDFS endpoint, run the following commands:

```
kinit krbtgt/DRS@DRT
hadoop fs -ls hdfs://[***REMOTE HDFS ENDPOINT***]:8020/
```

What to do next

Configure the user to run YARN jobs on both the clusters.

Configuring a user to run YARN jobs on both the clusters

To run Hadoop DistCp jobs to migrate the data from HDP to CDP Private Cloud Base cluster, you must use HDFS superuser or hdfs user.

About this task

Ensure that you have one of the following user accounts before you run Hadoop DistCp jobs:

- HDFS superuser - For information about creating a HDFS superuser, see Create the HDFS superuser.
- User named hdfs - By default, the hdfs user is not allowed to run YARN jobs. You must enable the hdfs user to run YARN jobs on both the clusters.

Procedure

1. Perform the following steps on the HDP cluster:
   a) Open the following file:

      /var/lib/ambari-server/resources/common-services/YARN/2.1.0.2.0/package/templates/container-executor.cfg.j2

   b) Remove the hdfs entry from banned-users list and save the file.

      Sample file contents:

```
yarn.nodemanager.local-dirs={{nm_local_dirs}}
yarn.nodemanager.log-dirs={{nm_log_dirs}}
yarn.nodemanager.linux-container-executor.group={{yarn_executor_containe
r_group}}
banned.users=yarn,hdfs,mapred,bin
min.user.id={{min_user_id}}
```

   c) On the YARN configuration page, verify whether the container-executor configuration template contains hdfs in the banned.users list.
   d) If hdfs is listed in the banner.users list, remove it from the template and save the template.
   e) Restart the following services:
      - Stale services, if any.
      - Ambari server
      - Ambari agent on each host of the cluster.
   f) In the yarn.admin.acl file, add hdfs.
   g) In the etc/hadoop/capacity-scheduler.xml fileSearch file, append hdfs to the yarn.scheduler.capacity.root.acl_submit_applications property.

h) Restart the YARN service.
i) Run the kinit command with the hdfs user's keytab file to authenticate the hdfs user to the Key Distribution Center (KDC).

2. On the CDP Private Cloud Base cluster, perform the following steps:

a) Select the YARN service.
b) Click the Configuration tab.
c) Make sure that hdfs user is not listed in the banned.users list.
d) Make sure that the min.user.id property is set to 0.
e) Restart the YARN service.

What to do next

Run the DistCp job on the CDP Private Cloud Base cluster.

Running DistCp job on the CDP Private Cloud Base cluster

After you make the required configuration changes in the HDP cluster and CDP Private Cloud Base cluster and configure a user to run the YARN jobs on both the clusters, you can run the Hadoop DistCp job.

Procedure

1. Restart the cluster services on both the clusters.

2. Run the following Hadoop DistCp command:

sudo -u *[***ENTER superuser OR hdfs***]* hadoop distcp *[***SOURCE CLUSTER***] [***DESTINATION CLUSTER***]*

For example,

sudo -u <superuser> hadoop distcp hdfs://nn1:8020/source hdfs://nn2:8020/destination