

Cloudera Manager 7.9.5

# Replication Manager for CDP Private Cloud Base

Date published: 2020-11-30

Date modified: 2023-01-25

# CLouDERA

<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

<b>Replication Manager in CDP Private Cloud Base.....</b>	<b>5</b>
<b>Support matrix for Replication Manager on CDP Private Cloud Base.....</b>	<b>7</b>
<b>Port requirements for Replication Manager on CDP Private Cloud Base.....</b>	<b>9</b>
<b>Prepare to replicate using replication policies.....</b>	<b>14</b>
Cloudera license requirements for Replication Manager.....	15
Configuring SSL/TLS certificate exchange between two Cloudera Manager instances.....	15
Add source cluster as peer to use in replication policies.....	16
Configuring a peer to use in replication policy.....	16
Modifying peers to use in replication policy.....	17
Configuring peers with SAML authentication.....	17
Enabling replication between clusters with Kerberos authentication.....	18
Required ports in Kerberos authentication-enabled clusters for replication.....	18
Considerations for realm names to use for replication.....	18
Prepare Kerberos authentication-enabled clusters for replication.....	18
Kerberos connectivity test.....	19
Replicating from unsecure to secure clusters.....	20
Replication of encrypted data.....	21
Encrypting data in transit between clusters.....	21
Security considerations for encrypted data during replication.....	22
Configuring heap size to replicate large directories using replication policies.....	22
Retaining logs for Replication Manager.....	22
<b>HDFS replication policies.....</b>	<b>23</b>
HDFS replication policy considerations.....	23
Guidelines to add or delete source data during replication job run.....	23
Improve network latency during replication job run.....	23
Performance and scalability limitations to consider for replication policies.....	24
Guidelines to use snapshot diff-based replication.....	24
HDFS replication in Sentry-enabled clusters.....	25
Specifying hosts to improve HDFS replication policy performance.....	26
Creating HDFS replication policy to replicate HDFS data.....	26
View HDFS replication policy details.....	31
View historical details for an HDFS replication policy.....	33
Monitoring the performance of HDFS replication policies.....	35
<b>Hive external table replication policies.....</b>	<b>37</b>
Hive replication policy considerations.....	38
Specifying hosts to improve Hive replication policy performance.....	38
Understanding how DDL commands affect Hive tables during replication.....	38
Disabling replication of parameters during Hive replication.....	39
Accommodate HMS changes for Hive replication policies.....	39

Creating a Hive external table replication policy.....	39
Sentry to Ranger replication for Hive external tables.....	45
Importing Sentry privileges into Ranger policies.....	46
Replicating data to Impala clusters.....	47
Replication of Impala and Hive User Defined Functions (UDFs).....	47
Monitoring the performance of Hive/Impala replication policies.....	48
<b>Hive ACID table replication policies.....</b>	<b>50</b>
Prepare to create Hive ACID table replication policies.....	50
Configure two-way trust between clusters.....	51
Configure parameters for Hive ACID table replication policies.....	53
Configure file access control lists for Impala user.....	56
Creating Hive ACID table replication policy.....	57
Managing Hive ACID table replication policies.....	59
Troubleshooting Hive ACID table replication policies.....	60
<b>Ozone replication policies using Cloudera Manager APIs.....</b>	<b>62</b>
Preparing clusters to replicate Ozone data.....	63
Configuring properties for OBS bucket replication using Ozone replication policies.....	63
Creating Ozone replication policies using Cloudera Manager APIs.....	64
Managing Ozone replication policies.....	68
<b>Managing replication policies.....</b>	<b>69</b>
<b>Troubleshooting replication policies between on-premises clusters.....</b>	<b>69</b>
<b>Snapshots.....</b>	<b>71</b>
Using snapshots with replication.....	71
Snapshot policies in Replication Manager.....	71
Creating and managing snapshot policies.....	72
Snapshots history.....	73
Hive/Impala replication using snapshots.....	73
Orphaned snapshots.....	74
Managing HDFS snapshots in Cloudera Manager.....	74
Browse HDFS directories.....	75
Enabling and disabling HDFS snapshots.....	75
Taking and deleting HDFS snapshots.....	75
Restoring HDFS snapshots.....	76
<b>Using DistCp to migrate HDFS data from HDP cluster to CDP Private</b>	
<b>Cloud Base cluster.....</b>	<b>77</b>
Migrating data from secure HDP cluster to unsecure CDP Private Cloud Base cluster using DistCp.....	77
Enabling the hdfs user to run the YARN jobs on the HDP cluster.....	77
Configuration changes on the CDP Private Cloud Base cluster.....	78
Running the DistCp job on the HDP cluster.....	78
Migrating data from secure HDP cluster to secure CDP Private Cloud Base cluster.....	79
Configuration changes on HDP cluster and CDP Private Cloud Base cluster.....	79
Configuring a user to run YARN jobs on both the clusters.....	80
Running DistCp job on the CDP Private Cloud Base cluster.....	81

## Replication Manager in 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 or HBase snapshot policies to take snapshots of HDFS directories and HBase tables 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 7.

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 use Cloudera Manager to schedule, save, and restore snapshots of HDFS directories and HBase tables.



**Tip:** Perform a *dry run* to verify configuration and understand the cost of the overall operation before actually copying the entire dataset.



**Important:** 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.

Policy ID	Policy Name	Policy Labels	Status	Audit Logging	Roles	Groups	Users	Action
7	all - global	--	Enabled	Enabled	cdep_global_admin	--	rangerlookup, hive, beacon, dpprofiler + More...	👁️ 📄 🗑️
8	all - database, table, column	--	Enabled	Enabled	cdep_global_admin	--	rangerlookup, hive, beacon, dpprofiler, hdfs, hbase, impala, hdfs, OWNER + More... Less...	👁️ 📄 🗑️
9	all - database, table	--	Enabled	Enabled	--	--	hive, beacon, dpprofiler, hbase + More...	👁️ 📄 🗑️
10	all - database	--	Enabled	Enabled	--	public	hive, beacon, dpprofiler, hbase + More...	👁️ 📄 🗑️
11	all - hiveservice	--	Enabled	Enabled	cdep_global_admin	--	rangerlookup, hive, beacon, dpprofiler + More...	👁️ 📄 🗑️

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

### HDFS replication policies

These policies replicate HDFS data and metadata from 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 or Microsoft ADLS Gen2 (ABFS) cloud buckets 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.

### Ozone replication policies

You use Cloudera Manager APIs to 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 supports the following functionalities:

- Replicates data between FSO buckets in source and target clusters using ofs protocol. Supports incremental replication using file checksums.
- Replicates data between legacy buckets in source and target clusters using ofs protocol. Supports incremental replication using file checksums.



**Note:** If one or both of the source and destination buckets are legacy buckets, 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.

- Replicates data between 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.

### HDFS and HBase snapshot policies

These policies take regular point-in-time snapshots of HDFS directories and HBase tables respectively.

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

## Support matrix for Replication Manager on CDP Private Cloud Base

Replication Manager replicates HDFS, Hive external tables, Impala data, Hive ACID tables, and data in Ozone buckets.

### Replicate data from CDH and CDP Private Cloud Base source clusters

The following table lists the source and destination clusters, lowest supported versions of Cloudera Manager, and the services that are available for each supported cloud provider for 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

The following table lists the source and destination clusters, lowest supported versions of Cloudera Manager, and the services that are available for each supported cloud provider for CDP Private Cloud Base source clusters:

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	<ul style="list-style-type: none"> <li>• HDFS</li> <li>• Hive external tables</li> </ul>
CDP Private Cloud Base	7.7.1	7.1.8	CDP Private Cloud Base	<ul style="list-style-type: none"> <li>• Hive ACID tables*</li> <li>• Use Cloudera Manager APIs to replicate Ozone buckets.</li> </ul>

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	7.7.1 CHF4	7.1.8 CHF4	CDP Private Cloud Base	Ozone buckets
*You can use <a href="#">REPL commands</a> 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.				



**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.




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

### Replicate data 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:

**Table 1: Replicate data from HDP 2 and HDP 3 source clusters**

Lowest supported source version	Services that require alternate replication methods
HDP 2.6.5	HDFS. Use <a href="#">DistCp</a> to replicate data.
HDP 3.1.1	HDFS. Use <a href="#">DistCp</a> to replicate data.
HDP 3.1.1	<ul style="list-style-type: none"> <li>• HBase. Use <a href="#">HBase replication</a> to replicate HBase data.</li> <li>• Hive external tables. For information to replicate data, contact Cloudera Support.</li> </ul>
HDP 3.1.5	Hive ACID tables to CDP 7.1.6 and higher clusters. Use <a href="#">REPL commands</a> to replicate data.  <b>Note:</b> Requires HDP 3.1.5 hotfixes.

### Supported replication scenarios

#### Supported features

- Supports replication 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.
- Supports replication to and from Microsoft ADLS Gen1 from CDH 5.13+ and Cloudera Manager 5.15, 5.16, 6.1+.
- Supports replication 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+.

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.

### Sentry-related replication

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.

When the source cluster is Sentry-enabled and you want to run HDFS replication policies, use the `hdfs` user to run the replication policy. The replication policy copies the permissions of replicated files and tables to the target cluster. To use any other user account, make sure that you configure the user account to bypass Sentry ACLs during replication.

When you create a Hive external table replication policy, choose the appropriate options to ensure that the Sentry permissions are migrated to Ranger permissions. The Replication Manager uses the `authzmigrator` tool to move data from Sentry to Ranger during Hive external table replication.

### 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 20.

### 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.


## Port requirements for Replication Manager on CDP Private Cloud Base

Before you create replication policies in Replication Manager, ensure that the following 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, as required.

Service	Default Port
<p>Cloudera Manager HTTP (Web UI)</p>	<p>7180</p> <p> <b>Note:</b> 7183 when TLS enabled</p>

**Note:** Cloudera Management Nodes (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 enabled port 7180 remain open, but redirected all requests to HTTP on port 7183.

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

Service	Default Port	
NameNode WebHDFS	9870  <b>Note:</b> 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 <a href="#">Using HttpFS</a> .
YARN Resource Manager	8032	Used for primary nodes flow to access the YARN ResourceManager. For more information, see <a href="#">YARN Configuration Properties</a> .

Service	Default Port	
Hive Metastore	9083	<p>Used for Hive/Impala replication to query or access Hive Metastore. For more information, see <a href="#">Configure metastore location and HTTP mode</a>.</p>
Impala Catalog Server	26000	<p>Internal data flow during Hive/Impala replication. The catalog service uses this port to communicate with the Impala daemons.</p>

Service	Default Port	
Key Management Server (KMS)	16000	<p><b>Required Primary Nodes</b></p> <p>flow during replication of encrypted data. Applies to both Java KeyStore KMS and Key Trustee KMS. For more information, see <a href="#">Migrating Keys</a>.</p>
Kerberos KDC Server and KRB5 services	88	<p><b>Required</b></p> <p>for authentication flow by Replication Manager when Kerberos authentication is enabled on the clusters.</p> <p>Open the port on all the hosts on the destination cluster.</p>
*Cloudera Manager		

## 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



**Important:** 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.



**Note:** 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. List the contents of the keystore file located in the source cluster Cloudera Manager.

```
/usr/lib/jvm/java-openjdk-11/bin/keytool -list -keystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks -storepass [***password***]
```



**Tip:** The global truststore password is available in the `/etc/hadoop/conf/ssl-client.xml` file.

2. Export the certificate contents to the `cert.txt` file.

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

3. Copy the `cert.txt` file to all the hosts of the target cluster Cloudera Manager securely.

```
$ mv cert.txt othercert.txt
$ scp -i hw-qe-keypair.pem othercert.txt root@[***host_ip***]:/home/
```

4. Import the certificate into the keystore file on all the hosts of the target cluster Cloudera Manager.

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

5. Run steps 1 and 2 in the target cluster Cloudera Manager, and then copy the `cert.txt` file to all the hosts in the source cluster Cloudera Manager securely, and import the certificate into the keystore file on all the hosts of the source cluster Cloudera Manager (steps 3 and 4).

6. 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 following command on all the nodes of the source cluster:

```
keytool -importkeystore -destkeystore [***jdk_cacerts_location***] -srckeystore [***cm-auto-global_truststore.jks_location***] -srcaalias [***cm_alias***]
```

For example,

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

b) Run the following commands on all the nodes of the target cluster:

```
keytool -importkeystore -destkeystore [***jdk_cacerts_location***] -srckeystore [***cm-auto-global_truststore.jks_location***] -srcaalias [***cm_alias_0***]
```

```
keytool -importkeystore -destkeystore [***jdk_cacerts_location***] -srckeystore [***cm-auto-global_truststore.jks_location***] -srcaalias [***cm_alias_1***]
```

For example,

```
keytool -importkeystore -destkeystore /usr/java/default/lib/security/cacerts -srckeystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks -srcaalias cmrootca-0
keytool -importkeystore -destkeystore /usr/java/default/lib/security/cacerts -srckeystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks -srcaalias cmrootca-1
```



**Tip:** You can use the `-deststorepass` option if you do not know the JDK trust store password.

## 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 replications set up using that Cloudera Manager instance. From the Admin Console of this destination Cloudera Manager instance, you can designate a peer Cloudera Manager Server as a source of HDFS and Hive data for replication.

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator).

## Configuring a peer to use in replication policy

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

### About this task

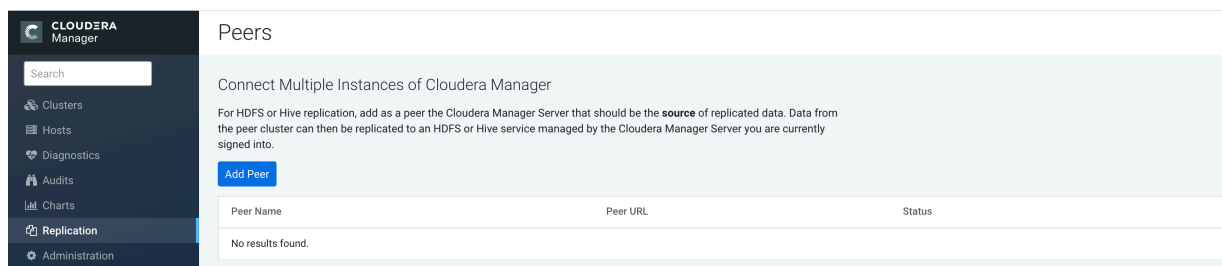
If your cluster uses SAML Authentication, see [Configuring peers with SAML authentication](#) on page 17 before configuring a peer.



## 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.



2. Click Add Peer.
3. In the Add Peer dialog box, provide a name, the peer URL (including the port) of the Cloudera Manager Server source for the data to be replicated, and the login credentials for that server.



**Important:** The role assigned to the login on the source server must be either a *User Administrator* or a *Full Administrator*.

Cloudera recommends that TLS/SSL be used. A warning is shown if the URL scheme is http instead of https. After configuring both peers to use TLS/SSL, add the remote source Cloudera Manager TLS/SSL certificate to the local Cloudera Manager truststore, and vice versa.



**Important:** 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.

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 requirements for Replication Manager on CDP Private Cloud Base](#) on page 9.

### 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 18 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/krb5.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 *HDFS service* Configuration tab.
  - 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 a Key Management Server (KMS) to manage encryption keys. During replication, Cloudera Manager uses TLS/SSL to encrypt the keys when they are transferred from the source cluster to the destination cluster. When an HDFS replication command that specifies an encrypted source directory runs, Cloudera Manager temporarily copies the encryption keys from the source cluster to the destination cluster, using TLS/SSL (if configured for the KMS) to encrypt the keys. Cloudera Manager then uses these keys to decrypt the encrypted files when they are received from the source cluster before writing the files to the destination cluster.



**Important:** When you create HDFS replication policy, you must select the `Advanced Skip Checksum check` property to prevent replication failure 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.

Even when the source and destination directories are both in encryption zones, the data is decrypted as it is read from the source cluster (using the key for the source encryption zone) and encrypted again when it is written to the destination cluster (using the key for the destination encryption zone). The data transmission is encrypted if you have configured encryption for HDFS data transfer.



**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.

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.

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=-Xmx1g`, 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.

## 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 7.

## 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

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 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.

### 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 7
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 `hdfsrc` 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.

- Go to the Cloudera Manager Replication Policies page, click Create Replication Policy.

- Select HDFS Replication Policy.

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

- 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 the directory (or file) you want to replicate.
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 the destination path to save the source files.
Schedule	Choose: <ul style="list-style-type: none"> <li>Immediate to run the schedule immediately.</li> <li>Once to run the schedule one time in the future. Set the date and time.</li> <li>Recurring to run the schedule periodically in the future. Set the date, time, and interval between runs.</li> </ul>



Option	Description
Run As Username	<p>Enter the user to run the replication job in the field. By default this is <code>hdfs</code>.</p> <p>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 <code>min.user.id</code> property in the YARN or MapReduce service.) Verify whether the user running the job has a home directory, <code>/user/username</code>, owned by <code>username:supergroup</code> in HDFS. This user must have permissions to read from the source directory and write to the destination directory.</p> <p>Note the following:</p> <ul style="list-style-type: none"> <li>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 <code>hdfs</code> user is banned by default from running YARN containers.</li> <li>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.</li> </ul> <p>To view the properties, go to the YARN service and search for the properties on the Configuration tab.</p>

6. Configure the following options on the Resources page:

Option	Description
Scheduler Pool	<p>(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:</p> <ul style="list-style-type: none"> <li>MapReduce – Fair scheduler: <code>mapred.fairscheduler.pool</code></li> <li>MapReduce – Capacity scheduler: <code>queue.name</code></li> <li>YARN – <code>mapreduce.job.queue.name</code></li> </ul>
Maximum Map Slots	Enter the number of map slots per mapper, as required. The default value is 20.
Maximum Bandwidth	Enter the bandwidth per mapper, as required. The default is 100 MB.
Replication Strategy	<p>Choose Static or Dynamic. Determines whether the file replication tasks must be distributed among the mappers statically or dynamically. The default is Dynamic.</p> <p>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.</p>

## 7. Configure the following options on the Advanced Options tab:

Option	Description
Add Exclusion	<p>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.</p> <p>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:</p> <pre data-bbox="889 464 1317 512">.* /user / \. snapshot / . + / bdr . *</pre> <p>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:</p> <pre data-bbox="889 659 1110 707">.* /user + / bdr . *</pre> <p>You can add more than one regular expression to exclude.</p>
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	<p>Select the following option based on your requirements:</p> <ul style="list-style-type: none"> <li>• Skip Checksum Checks - Determines whether to skip checksum checks on the copied files. If selected, checksums are not validated. Checksums are checked by default.</li> </ul> <p> <b>Important:</b> You must skip checksum checks to prevent replication failure due to non-matching checksums in the following cases:</p> <ul style="list-style-type: none"> <li>• Replications from an encrypted zone on the source cluster to an encrypted zone on a destination cluster.</li> <li>• Replications from an encryption zone on the source cluster to an unencrypted zone on the destination cluster.</li> <li>• Replications from an unencrypted zone on the source cluster to an encrypted zone on the destination cluster.</li> </ul> <p>Checksums are used for two purposes:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> </ul> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> <li>• 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.</li> </ul>
Preserve	<p>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. When Extended attributes is checked, and both the source and destination clusters support extended attributes, replication preserves them. (This option only displays when both source and destination clusters support extended attributes.)</p> <p> <b>Note:</b> 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.</p>

Option	Description
Delete Policy	<p>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:</p> <ul style="list-style-type: none"> <li>• Keep Deleted Files - Retains the destination files even when they no longer exist at the source. (This is the default.)</li> <li>• Delete to Trash - If the HDFS trash is enabled, files are moved to the trash folder.</li> <li>• 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.</li> </ul>
Alerts	Whether to generate alerts for various state changes in the replication workflow. You can alert on failure, on start, on success, or when the replication workflow is aborted.

#### 8. 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](#).

## 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	<p>Average throughput per mapper/file of all the files written.</p>  <p><b>Note:</b> 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.</p>
Progress	Current replication job status.
Completed	<p>Time stamp when the replication job completed.</p> <p>Click the Completed column label to sort the replication policies table by time.</p>
Next Run	<p>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.</p> <p>Click the Next Run column label to sort the replication policies table by the next run date.</p>
Actions	<p>Click:</p> <ul style="list-style-type: none"> <li>• Show History to open the Replication History page for a replication policy.</li> <li>• Edit Configuration to change the replication policy options as required.</li> <li>• 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.</li> <li>• Run Now to run the replication task immediately.</li> <li>• 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.</li> </ul> <p>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.</p> <ul style="list-style-type: none"> <li>• Disable   Enable to disable the replication policy or enable the disabled replication policy. No further replications are scheduled for disabled replication policies.</li> <li>• Delete to remove the replication policy permanently from Replication Manager. Deleting a replication policy does not delete copied files or tables.</li> </ul>

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:



The screenshot shows the Cloudera Manager interface for Replication Policies. The left sidebar contains navigation options: Clusters, Hosts, Diagnostics, Audits, Charts, Replication (selected), and Administration. The main content area is titled "Replication Policies" and includes a search bar, a "Last Refreshed 12:06 PM" timestamp, and a "Create Replication Policy" button. A "Filters" panel on the left shows counts for STATUS (Failed: 1, Succeeded: 2, Running: 0, Disabled: 0, Dry-run: 0), TYPE (HDFS: 2, HDFS-S3: 0, Hive: 1, Hive-S3: 0, HDFS-ADLS: 0, Hive-ADLS: 0), SOURCE (Cluster 1: 1, Cluster 1 @ test: 2), and TARGET. The main table lists three policies with their details and history.

ID	Name	Type	Source	Destination	Throughput	Progress	Completed	Next Run
5	test	HDFS	HDFS-1 Cluster 1	HDFS-1 Cluster 1			7:59 PM	None scheduled.
Message		HDFS replication command succeeded.						
From		/tmp						
To		/tmp/rece						
8	tsadf	HDFS	HDFS-1 Cluster 1 @ test	HDFS-1 Cluster 1			12:33 AM	None scheduled.
Message		HDFS replication command succeeded.						
From		/tmp						
To		/tmp/rec						
12	testadsf	Hive	HIVE-1 Cluster 1 @ test	HIVE-1 Cluster 1			1:29 AM	None scheduled.
Message		Hive Replication Import step failed.						
Objects:		Custom Databases						

## 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	<p>Shows the job details.</p> <p>Expand the section to view the following job details:</p> <ul style="list-style-type: none"> <li>Started At timestamp is when the replication job started.</li> <li>Duration to complete the job.</li> <li>Command Details appear in a new tab after you click View.</li> </ul> <p>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.</p> <p>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:</p> <ul style="list-style-type: none"> <li>View the actual command string.</li> <li>View the start time and duration for the command run.</li> <li>View the host status page for the command by clicking the host link.</li> <li>View the full log file for the command by selecting the stdout or stderr tab.</li> </ul> <p>For more information, see <a href="#">Viewing Running and Recent Commands</a>.</p> <ul style="list-style-type: none"> <li>MapReduce Job details appear after you click the job link.</li> <li>Download the following HDS Replication Reports in CSV format after you click Download CSV: <ul style="list-style-type: none"> <li>Listing report contains the list of files and directories copied during the replication job.</li> <li>Status report contains the full status report of the files where the replication status is shown as: <ul style="list-style-type: none"> <li><b>ERROR</b> occurred during replication, therefore the file was not copied.</li> <li><b>DELETED</b> for deleted files.</li> <li><b>SKIPPED</b> for up-to-date files that were not replicated.</li> </ul> </li> <li>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.</li> <li>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.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul> </li> <li>(Dry Run only) Replicable Files shows the number of files that would be replicated during an actual replication.</li> <li>(Dry Run only) Replicable Bytes shows the number of bytes that would be replicated during an actual replication.</li> <li>View the number of Impala UDFs replicated. (Displays only for Hive/Impala replications where Replicate Impala Metadata is selected.)</li> <li>If a user was specified in the Run As Username field when creating the replication job, the selected user appears.</li> <li>View messages returned from the replication job.</li> </ul>
Duration	Time taken for the replication job to complete.
Outcome	Status of the replication job as <b>Successful</b> or <b>Failed</b> .
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:

## Replication Policies

## Replication History

Name	test	Type	HDFS	Source	HDFS-1 (Cluster 1)	Destination	HDFS-1 (Cluster 1)	Next Run	None scheduled.
------	------	------	------	--------	--------------------	-------------	--------------------	----------	-----------------

Start Time	Duration	Outcome	Files Expected	Files Copied	Files Failed	Files Deleted	Files Skipped
September 23, 2020 7:58 PM	1 min	Successful	80 (722.5 MiB)	17 (94.4 MiB)	0 (0 B)	0	63 (628.1 MiB)
Started At	September 23, 2020 7:58 PM						
Duration	a few seconds						
Command Details	<a href="#">View</a>						
MapReduce Job	<a href="#">job_1600880827337_0009</a>						
HDFS Replication	<a href="#">Download CSV</a>						
Report							
Message	17 file(s) copied, 63 unchanged.						

## 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 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.

Replication Policies

Replication History

Name	test	Type	HDFS	Source	HDFS-1 (Cluster 1)	Destination	HDFS-1 (Cluster 1)	Next Run	None scheduled.
Start Time	Duration	Outcome	Files Expected	Files Copied	Files Failed	Files Deleted	Files Skipped		
September 23, 2020 7:58 PM	1 min	Successful	80 (722.5 MiB)	17 (94.4 MiB)	0 (0 B)	0	63 (628.1 MiB)		
Started At	September 23, 2020 7:58 PM								
Duration	a few seconds								
Command Details	<a href="#">View</a>								
MapReduce Job	<a href="#">job_1600880827337_0009</a>								
HDFS Replication Report	<a href="#">Download CSV</a>								
Message	ed.								
September 23, 2020 7:43	Successful	63 (628.1 MiB)	15 (93.2 MiB)	0 (0 B)	0	48 (534.9 MiB)			
September 23, 2020 7:41	Successful	48 (534.9 MiB)	13 (92 MiB)	0 (0 B)	0	35 (442.9 MiB)			
September 23, 2020 7:39	Successful	35 (442.9 MiB)	11 (90.8 MiB)	0 (0 B)	0	24 (352.2 MiB)			
September 23, 2020 7:37	Successful	24 (352.2 MiB)	9 (89.6 MiB)	0 (0 B)	0	15 (262.6 MiB)			

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.



### 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 7.

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.



**Caution:** Because of the warehouse directory changes between CDH clusters and CDP Private Cloud Base, Hive external table replication does not copy the table data from the database and tables specified in the source cluster. But the replication job gets successfully run without any disruptions. While replicating from CDH clusters to CDP Private Cloud Base, it is recommended that the HDFS Destination Path is defined. If HDFS Destination Path is not defined and 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.



**Important:** Since Hive3 has a different default table type and warehouse directory structure, 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 will be created under external Hive warehouse directory set by hive.metastore.warehouse.external.dir Hive configuration parameter. Users 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 users want to replicate the same database from Hive2 to Hive3 (that will have different paths by design), they need to use Force Overwrite option per policy to avoid any mismatch issues.



**Note:** While replicating from Sentry to Ranger, the minimum supported Cloudera Manager version is 6.3.1 and above.

Configuration notes:

- 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 tables backed by Kudu, Replication Manager filters out Kudu tables for a Hive external table replication in order 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 tables in a 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 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 Delete Policy for the replication job is set to Delete to Trash or Delete Permanently. The corresponding data files are deleted on the destination during the next replication job.

## 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.

### Before you begin

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.

Replication Manager functions consistently across HDFS and Hive:

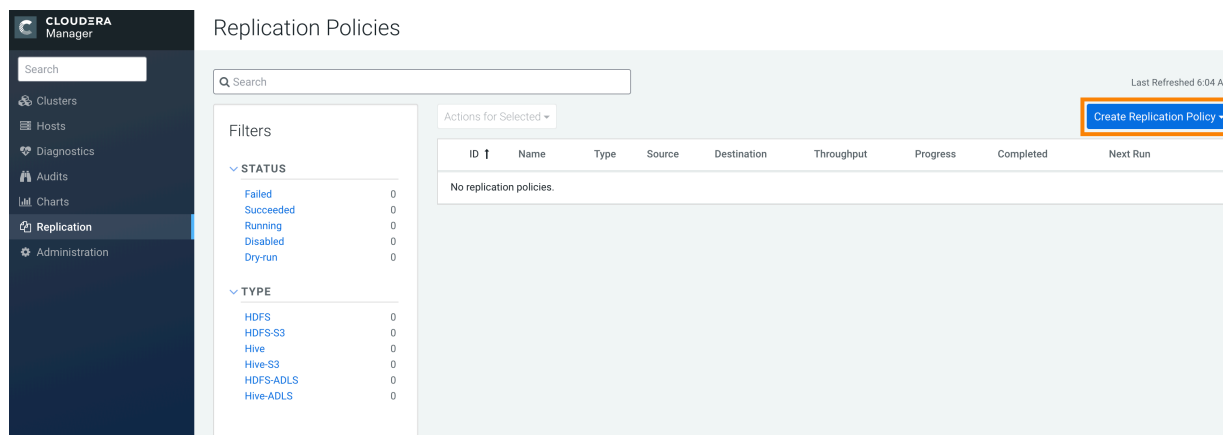
- Replication policies can be set up on files or directories in HDFS and on external tables in Hive—without manual translation of Hive datasets to HDFS datasets, or vice versa. Hive Metastore information is also replicated.
- Applications that depend on external table definitions stored in Hive, operate 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:

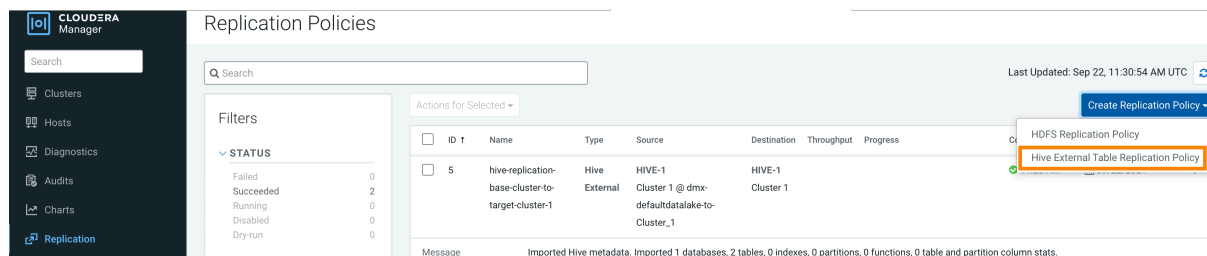
1. Log in to Ranger Admin UI.
  2. Go to the Service Manager Hadoop\_SQL Policies Access section, and provide hdfs user permission to the all-database, table, column policy name.
- On the target cluster, the hive user must have Ranger admin privileges. The same hive user performs the metadata import operation.

## Procedure

1. If the source cluster is managed by a different Cloudera Manager server than the destination cluster, configure a peer relationship.
2. Go to the Cloudera Manager Replication Replication Policies page, click Create Replication Policy.




3. Select Hive External Table Replication Policy.



4. 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.
Use HDFS Destination	Select this option based on the type of destination cluster you plan to use.
Import Sentry permissions	Select one of the following permissions: <ul style="list-style-type: none"> <li>• Do not import Sentry Permissions (Default)</li> <li>• If Sentry permissions were exported from the CDH cluster, import both Hive object and URL permissions</li> <li>• If Sentry permissions were exported from the CDH cluster, import only Hive object permissions</li> </ul>





Option	Description
Replicate All	<p>Select to replicate all the Hive databases from the source.</p> <p>To replicate only selected databases, clear the option and enter the database name(s) and tables you want to replicate.</p> <ul style="list-style-type: none"> <li>Specify multiple databases and tables using the plus symbol to add more rows to the specification.</li> <li>Specify multiple databases on a single line by separating their names with the pipe ( ) character. For example: mydbname1 mydbname2 mydbname3.</li> <li>Use regular expressions in the database or table fields as shown in the following examples: <ul style="list-style-type: none"> <li>To specify any database or table name, enter the following regular expression: <pre>[ \w ] . +</pre> </li> <li>To specify any database or table except the one named 'myname', enter the following regular expression: <pre>( ? !myname\b ) . +</pre> </li> <li>To specify all the tables in the db1 and db2 databases, enter the following regular expression: <pre>db1   db2 [ \w_ ] +</pre> </li> <li>To specify all the tables of the db1 and db2 databases (alternate method), enter the following regular expression: <pre>db1 [ \w_ ] +</pre> <p>Click + icon and enter the following expression:</p> <pre>db2 [ \w_ ] +</pre> </li> </ul> </li> </ul>
Run As Username	<p>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.</p> <p> <b>Note:</b> 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.</p>
Run on peer as Username	<p>Enter the username if the peer cluster is configured with a different superuser. This is applicable in a kerberized environment.</p>

#### 5. Configure the following options on the Resources tab:



Option	Description
Scheduler Pool	<p>(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:</p> <ul style="list-style-type: none"> <li>MapReduce – Fair scheduler: mapred.fairscheduler.pool</li> <li>MapReduce – Capacity scheduler: queue.name</li> <li>YARN – mapreduce.job.queueName</li> </ul>

Option	Description
Maximum Map Slots	Enter the number of map slots per mapper, as required. The default value is 20.
Maximum Bandwidth	Enter the bandwidth per mapper, as required. The default is 100 MB.
Replication Strategy	<p>Choose Static or Dynamic to determine whether the file replication tasks must be distributed among the mappers statically or dynamically. The default is Dynamic</p> <p>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.</p>

6. 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.
Replicate Impala Metadata	<p>If both the source and destination clusters use CDH 5.7.0 or later up to and including 5.11.x, select No to avoid redundant replication of Impala metadata. This option appears if both source and destination clusters support this functionality.</p> <p>Select:</p> <ul style="list-style-type: none"> <li>• Yes to replicate the Impala metadata.</li> <li>• No to not replicate the Impala metadata.</li> <li>• Auto for Cloudera Manager to determine whether or not to replicate the Impala metadata based on the CDH version.</li> </ul> <p>To replicate Impala UDFs when the version of CDH managed by Cloudera Manager is 5.7 or lower, see <a href="#">Replicate Impala and Hive User Defined Functions (UDFs)</a>.</p>
Force Overwrite	<p>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.</p> <p> <b>Important:</b> 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.</p>
Export Path	<p>Specify a path to override the default HDFS location for the export file.</p> <p>By default, Hive metadata is exported to a default HDFS location (/user/\$[***user.name***/.cm/hive) and then imported from this HDFS file to the destination Hive metastore. In this example, user .name is the process user of the HDFS service on the destination cluster.</p> <p> <b>Note:</b> In a Kerberized cluster, the HDFS principal on the source cluster must have read, write, and execute access to the Export Path directory on the destination cluster.</p>

Option	Description
Number of concurrent HMS connections	<p>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.</p> <ul style="list-style-type: none"> <li><b>a.</b> If you set the value to 1 or more, Replication Manager uses multi-threading with the number of connections specified.</li> <li><b>b.</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.</li> </ul>
HDFS Destination Path	<p>Enter a path to override the default path.</p> <p>By default, Hive HDFS data files (for example, /user/hive/warehouse/db1/t1) are replicated to a location relative to "/" (in this example, to /user/hive/warehouse/db1/t1).</p> <p>For example, if you enter /ReplicatedData, the data files are replicated to /ReplicatedData/user/hive/warehouse/db1/t1.</p>
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	<p>Select:</p> <ul style="list-style-type: none"> <li>• 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.</li> </ul> <p> <b>Important:</b> You must skip checksum checks to prevent replication failure due to non-matching checksums in the following cases:</p> <ul style="list-style-type: none"> <li>• Replications from an encrypted zone on the source cluster to an encrypted zone on a destination cluster.</li> <li>• Replications from an encryption zone on the source cluster to an unencrypted zone on the destination cluster.</li> <li>• Replications from an unencrypted zone on the source cluster to an encrypted zone on the destination cluster.</li> </ul> <p>Checksums are used for two purposes:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> </ul> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> <li>• 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.</li> </ul>
Preserve	<p>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.</p> <p> <b>Note:</b> You must be running as a superuser to preserve permissions. Use the Run As Username option to ensure that is the case.</p>

Option	Description
Delete Policy	<p>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.</p> <p>Choose:</p> <ul style="list-style-type: none"> <li>• Keep Deleted Files to retain the destination files even when they no longer exist at the source. This is the default.</li> <li>• Delete to Trash iff the HDFS trash is enabled.</li> <li>• 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.</li> </ul>
Alerts	<p>Determines whether to generate alerts for various state changes in the replication workflow. You can alert on failure, on start, on success, or when the replication workflow is aborted.</p>

#### 7. 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 for Hive external tables

When you create or edit a Hive external table replication policy, 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. Cloudera Manager version 6.3.1 and higher is required to replicate Sentry policies to Ranger.

In a Hive external table replication policy, if 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, Replication Manager performs the following tasks automatically during the replication job run:

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 endpoint. To filter the policies, the Replication Manager uses a filter expression that is passed to the authzmigrator tool by Cloudera Manager.



**Note:** If you are replicating a subset of the tables in a database, database-level policies get converted to equivalent table-level policies for each table being replicated. (For example, ALL on database -> ALL on table individually for each table replicated).



**Caution:** There will be no reference to the original role names in Ranger. The permissions are granted directly to 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.



**Attention:** Regardless of whether a policy was modified or not, each policy will be re-created on each replication. If you wish to continue scheduling data replication but you also want to modify the target cluster's Ranger policies (and keep those modifications), you should disable the Sentry to Ranger migration on subsequent runs by editing the replication policy and choose the Do not import Sentry Permissions (Default) option.

## 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 2: 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

Sentry Action	Ranger Action
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 Replication Manager to copy ACID tables between CDP Private Cloud Base clusters for backup, load balancing, and other purposes.

**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 7.

Hive ACID table replication policies can:

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

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.



**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.

## Prepare 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 7.

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 51
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 to turn on the ChangeManager for the following services:

Service	Action
Hive-on-Tez	On the Configuration tab, search for Hive Service Advanced Configuration Snippet (Safety Valve) for <code>hive-site.xml</code> property and set the key-value pair
Hive-1	On the Configuration tab, search for Hive Service Advanced Configuration Snippet (Safety Valve) for <code>hive-site.xml</code> property and set the key-value pair.
Hive-1	On the Configuration tab, search for Enable ChangeManager for Hive replication parameter and select it.



**Important:** Restart the Hive1 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 54.  
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 53 and [Parameters to optimize Hive ACID table replication performance](#) on page 54 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.

## 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. Navigate to the Cloudera Manager Clusters *Hive on Tez Service* Configuration page, and configure the following properties 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.
  - 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 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.encryptedzone.rootdir***]`
- `hdfs dfs -setfacl -m user:impala:rwx [***ENCRYPTION ZONE PATH or VALUE OF hive.repl.cm.encryptedzone.rootdir***]`



**Note:** The default value of `hive.repl.cm.encryptedzone.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 [Prepare to create Hive ACID table replication policies](#) on page 50.



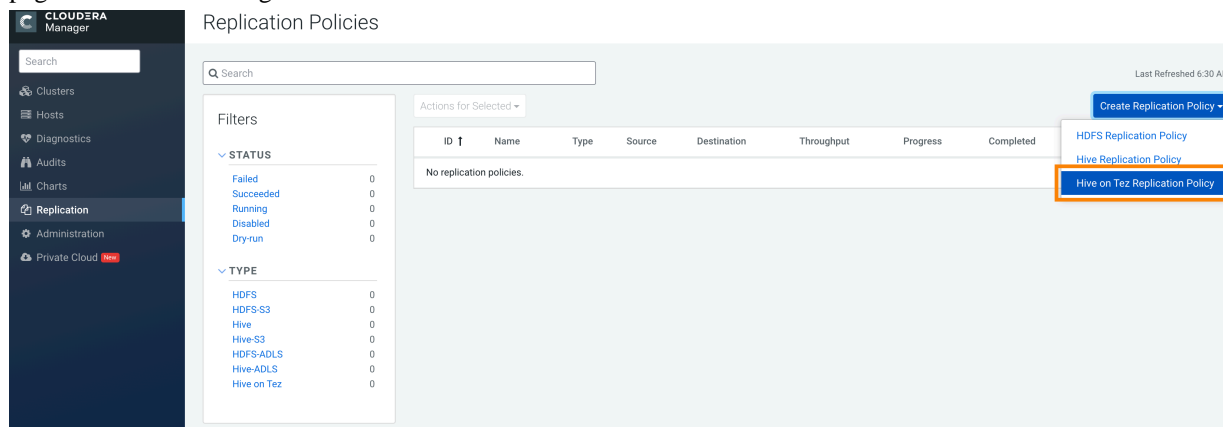
**Important:** To replicate managed tables (ACID) and external tables in a database successfully, you must perform the following steps in the order shown below:

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	<p>Enter a valid path to the staging location. The path must have the required permissions for the <i>hive</i> user. For example: <code>/user/hive/data</code>.</p> <p>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 <code>hive.repl.rootdir</code> 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.</p> <p> <b>Important:</b> 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.</p>
Source Database	Enter the source database name.
Schedule	<p>Schedule the replication job as necessary. The Hive ACID table replication policy uses the Hive scheduler to schedule the replication policies.</p> <p>You can choose the following schedule options:</p> <ul style="list-style-type: none"> <li>• 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.</li> </ul> <p>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.</p> <ul style="list-style-type: none"> <li>• Unix Cron Expression - Enter a valid Unix cron expression.</li> </ul> <p>You can generate the Unix cron expression in the following page, and then use it in the Create Hive ACID Table Replication Policy wizard:</p> <p><a href="https://www.freeformatter.com/cron-expression-generator-quartz.html">https://www.freeformatter.com/cron-expression-generator-quartz.html</a></p> <p>For example, you can use <code>0 */30 * ? * *</code> to schedule the policy to run every 30 minutes, <code>0 0 */4 ? * *</code> to schedule the run for every 4 hours, or <code>0 0 0 * * ?</code> every day at midnight.</p>
Run as Username	Enter <code>hive</code> .

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

Option	Description
Scheduler Pool	<p>(Optional) Enter the name of a resource pool.</p> <p>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:</p> <ul style="list-style-type: none"> <li>• MapReduce – Fair scheduler: <code>mapred.fairscheduler.pool</code></li> <li>• MapReduce – Capacity scheduler: <code>queue.name</code></li> <li>• YARN – <code>mapreduce.job.queue.name</code></li> </ul> <p> <b>Note:</b> The DistCp job running on the target cluster or a YARN uses the <code>mapreduce.job.queue.name</code> property.</p>
Maximum Number of Copy Mappers	Enter the maximum number of simultaneous copy mappers for DistCp. The default value is 20.
Maximum Bandwidth per Copy Mapper	Enter the maximum bandwidth for each mapper in megabytes per second. The default is 100 MB.

- 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.

- 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:

- Go to the error log path.
- Search for the file named `_non_recoverable`.
- Check the error stack that is printed in the `_non_recoverable` file.
- Fix the error.
- Delete the `_non_recoverable` file. For the next replication jobs in the policy to function normally, the `_non_recoverable` 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

- Go to the [Replication Replication Policies](#) page.
- Select the required replication policy.
- Click Actions. The following options appear:
  - Show History to view the replication job history on the **Replication History** page.
  - Edit Configuration to edit the schedule of the replication policy.
  - Run Now to run the replication job.
  - Disable the selected replication job
  - Delete the selected replication job.
- The **Replication History** page shows 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.

## Results

When you schedule and submit a 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 run.

## 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.

### What do I do 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.

Important:

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 do I 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.

## Ozone replication policies using Cloudera Manager APIs

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 use Cloudera Manager APIs to 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.

Cloudera supports the following types of Ozone storage:

- Object store buckets (OBS) 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) 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 are Ozone buckets created prior to CDP Private Cloud Base 7.1.8 and uses the Ozone File System (ofs) protocol or scheme.

Ozone replication policies use Cloudera Manager APIs to replicate data. 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.

Ozone replication policies supports the following functionalities:

- Replicates data between FSO buckets in source and target clusters using ofs protocol. Supports incremental replication using file checksums.
- Replicates data between legacy buckets in source and target clusters using ofs protocol. Supports incremental replication using file checksums.



**Note:** If one or both of the source and destination buckets are legacy buckets, 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.

- Replicates data between OBS buckets in source and target clusters that support S3A filesystem using the S3A scheme or replication protocol.

## 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.

### About this task

Before you create Ozone replication policies using Cloudera Manager APIs, you must 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 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_OPTIMIZED
```

- 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 63.
- Is Kerberos enabled on both the clusters? If so, perform the following steps:
  - a) Configure the SSL/TLS certificate exchange manually on source Cloudera Manager and target Cloudera Manager. For more information, see [Configuring SSL/TLS certificate exchange between two Cloudera Manager instances](#) on page 15.
  - b) Configure a user with permissions to access HDFS and Ozone.
  - c) Run the following command to add the group name of the user (For example, the group name *bdr*) to the Ozone service configuration in target cluster Cloudera Manager:
 

```
sudo usermod -a -G om bdr command
```
  - d) If Ranger is enabled on the source cluster, you must perform the following steps:
    - a. Login to the Ranger UI from source Cloudera Manager.
    - b. Click `cm_ozone` on the **Service Manager** page.
    - c. Add the user (that you configured in step 2) and *hive* to the all - volume, bucket, key policy name, and then set the groups for this policy as public.




**Note:** If Kerberos is enabled on both the clusters, you must run the `kinit -kt [/]**path**]/[/]**to**]/om.keytab om` command before you run any Ozone commands. Enter the absolute path to the Ozone service's keytab in this command.

## 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 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:

Property	Description
fs.s3a.endpoint	Enter the same value as in Ozone S3 gateway web UI as the source cluster.  <b>Tip:</b> The source and target cluster have their own S3A endpoint URL.
hadoop.tmp.dir	Enter the temporary directory on the target cluster to buffer file uploads.
fs.s3a.secret.key	See Step 3 to get the required value.
fs.s3a.access.key	See Step 3 to get the required value.

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:

Property	Description
fs.s3a.endpoint	Enter the same value as in Ozone S3 gateway web UI as the target cluster.  <b>Tip:</b> 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.

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` property to the YARN Service Environment Advanced Configuration Snippet (Safety Valve) for the YARN service in source Cloudera Manager.



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

## Creating Ozone replication policies using Cloudera Manager APIs

You can create Ozone replication policies using Cloudera Manager APIs 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 Support API Explorer page on the target cluster.
2. Go to the ReplicationsResource API resource.
3. Enter the following parameters to create an Ozone replication policy in the POST `/clusters/{clusterName}/services/{serviceName}/replications` HTTP method:
  - a. Enter the cluster name in the `clusterName` field. For example, Cluster 1.
  - b. Enter the service name in the `serviceName` field. For example, OZONE-1.
  - c. Enter one or more of the API parameters in the *API parameters for Ozone replication policy* table in the body field.



**Tip:** Optionally, you can also run the auto-generated curl command on CLI.

The following table lists the available API parameters that you can use to create an Ozone replication policy:

**Table 3: API parameters for Ozone replication policy**

Parameter name	Data type	Description
active	boolean	Read-only field. Shows true when the replication policy is running. Otherwise, this shows false.
alertOnAbort	boolean	Set to true to generate an alert when a replication job stops abruptly. Default is false.
alertOnFail	boolean	Set to true to generate an alert when a replication job fails. Default is false.
alertOnStart	boolean	Set to true to generate an alert when a replication job starts. Default is false.
alertOnSuccess	boolean	Set to true to generate an alert when a replication job completes successfully. Default is false.
description	string	Enter a description for the replication policy.
displayName	string	Enter a unique name for the replication policy.
endTime	string	The timestamp after which the replication job is not triggered.
abortOnError	boolean	Set to true to stop the replication job when an error appears. The files copied up to that point remain on the destination, but no additional files are copied. Default is false.
bandwidthPerMap	number	The maximum bandwidth (in MB) per mapper in the MapReduce replication job. The default value is 100 MB per second for each mapper.
destinationPath	string	Enter the path on the target cluster to which the replication policy copies the data to.
exclusionFilters	array of string	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.

Parameter name	Data type	Description
logPath	string	Enter an alternate path for the logs, if required.
mapreduceServiceName	string	Enter a MapReduce or YARN service to use for the replication policy.
numMaps	number	Enter the maximum number of map tasks (simultaneous copies) per replication job. The default value is 20.
removeMissingFiles	boolean	Set to true to delete destination files that are missing in source. Default is false.
replicationStrategy	Data model	<p>Choose one of the following replication strategies to determine whether the file replication tasks should be distributed among the mappers statically or dynamically.</p> <ul style="list-style-type: none"> <li>Static distributes file replication tasks among the mappers up front to achieve an uniform distribution based on the file sizes.</li> <li>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 unallocated set of tasks.</li> </ul> <p>The default replication strategy is Dynamic.</p>
schedulerPoolName	string	<p>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:</p> <ul style="list-style-type: none"> <li>MapReduce – Fair scheduler: <code>mapred.fairscheduler.pool</code></li> <li>MapReduce – Capacity scheduler: <code>queue.name</code></li> <li>YARN – <code>mapreduce.job.queue.name</code></li> </ul>
skipChecksumChecks	boolean	Set to true to skip checksum checks on the copied files. Checksums are checked by default.
skipListingChecksumChecks	boolean	Set to true to skip checksum check when comparing two files to determine whether they are the same or not. If skipped, the file size and last modified time are used to determine if files are the same. Skipping the check improves performance during the mapper phase. Note that if you select the Skip Checksum Checks option, this check is also skipped.
skipTrash	boolean	Permanently deletes destination files that are missing in source. Default is null.
sourcePath	string	Enter the path to the bucket on the source cluster to replicate data from.
sourceService	data model	<p>Enter the following:</p> <ul style="list-style-type: none"> <li><code>clusterName</code> - Enter the cluster name.</li> <li><code>peerName</code> - Enter the peer name that you entered when you added the cluster as a peer.</li> <li><code>serviceName</code> - Enter the Ozone service name in Cloudera Manager.</li> </ul>

Parameter name	Data type	Description
sourceUser	string	Enter the user name to run the replication policy. If you want to specify a user name for unsecure clusters, enter null.  If you are using a kerberized cluster, enter the required user name. The replication policy uses this user name to replicate the data in the kerberized cluster.
userName	string	Enter the user name to run the replication policy. If you want to specify a username for unsecure clusters, enter null.  If you are using a kerberized cluster, enter the required user name. The replication policy uses this user name to replicate the data in the kerberized cluster.
id	number	Auto-generated replication policy ID
interval	number	Enter the duration between consecutive replication policy runs. Default is 0.
intervalUnit	data model	Enter one of the following frequencies to run the replication policy:  MINUTE; HOUR; DAY; WEEK; MONTH; YEAR
nextRun	string	Read-only. The timestamp for the next scheduled replication policy run.
paused	boolean	Set to true to pause a replication job from replicating after the policy creation is complete. Default is false.
startTime	string	The timestamp to initiate the replication job run.

## Results

The response body shows the created Ozone replication policy.

## Example

The following sample snippet shows the API parameters required to create an Ozone replication policy:

```
"items": [{
  "active": false,
  "alertOnAbort": false,
  "alertOnFail": false,
  "alertOnStart": false,
  "alertOnSuccess": false,
  "description": null,
  "displayName": "Remote 1",
  "endTime": null,
  "ozoneReplicationArguments": {
    "abortOnError": false,
    "bandwidthPerMap": 100,
    "destinationPath": "voll/repl1",
    "exclusionFilters": [],
    "logPath": null,
    "mapreduceServiceName": "YARN-1",
    "numMaps": 20,
    "removeMissingFiles": false,
    "replicationStrategy": "DYNAMIC",
    "schedulerPoolName": null,
```

```

"skipChecksumChecks": false,
"skipListingChecksumChecks": false,
"skipTrash": false,
"sourcePath": "voll/repl1",
"sourceService": {
  "clusterName": "Cluster 1",
  "peerName": "Remote Source",
  "serviceName": "OZONE-1"
},
"sourceUser": "testuser",
"userName": "testuser"
},
"id": null,
"interval": 0,
"intervalUnit": "MINUTE",
"nextRun": null,
"paused": false,
"startTime": null
}]
}

```

## Managing Ozone replication policies

After you create Ozone replication policies in Cloudera Manager using Cloudera Manager APIs, you can run the replication job, delete the replication policy, and view all existing Ozone replication policies using Cloudera Manager APIs.

### Procedure

1. Go to the [Cloudera Manager Support API Explorer](#) page on the target cluster.
2. Go to the [ReplicationsResource](#) API resource.
3. You can perform the following tasks on an Ozone replication policy:
  - a) To run an Ozone replication policy, enter the following API parameters in the POST `/clusters/{clusterName}/services/{serviceName}/replications/{scheduleId}/run` HTTP method:
    - `clusterName` - Enter the target cluster name.
    - `scheduled` - Enter the ID of an existing replication policy.
    - `serviceName` - Enter the Ozone service name in the target Cloudera Manager.
    - `dryRun` - Optionally, choose true for a dry run.
  - b) To delete an Ozone replication policy, enter the following API parameter in the DELETE `/clusters/{clusterName}/services/{serviceName}/replications/{scheduleId}` HTTP method:
    - `clusterName` - Enter the target cluster name.
    - `scheduled` - Enter the ID of an existing replication policy.
    - `serviceName` - Enter the Ozone service name in the target Cloudera Manager.
  - c) To view information about all the existing replication policies, enter the following API parameters in the GET `/clusters/{clusterName}/services/{serviceName}/replications` HTTP method:
    - `clusterName` - Enter the target cluster name.
    - `serviceName` - Enter the Ozone service name in the target Cloudera Manager.
    - `view` - Optionally, you can choose one of the following options to get a summary view or detailed view about the replication policies. Default is the summary view.  
EXPORT; EXPORT\_REDACTED; FULL; FULL\_WITH\_HEALTH\_CHECK\_EXPLANATION; SUMMARY.

## Managing replication policies

When you create a new replication policy, it is automatically enabled. If you disable a replication policy, it can be re-enabled at a later time. You can enable, disable, or delete one or more replication policies at a time.

### Procedure

1. Go to the [Cloudera Manager Replication Policies](#) page.
2. Select [Actions Disable](#) to disable an active replication policy.
3. Select [Actions Enable](#) to enable a disabled replication policy.
4. Select [Actions Delete](#) to delete a replication policy.

## 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 [Cloudera Manager HDFS service Configuration](#) tab on the source cluster.
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. Perform steps 1 through 4 on the target cluster Cloudera Manager.

### How can file replication tasks be equitably distributed to all mappers?

The `Replication Strategy` option that you can configure during policy creation 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. By default, this option is set to 20.

- Choose Skip Checksum Checks during policy creation 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. Configure Maximum Bandwidth to limit the bandwidth per mapper. 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 this 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 solutions based on the following 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 API 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.

## Snapshots

You can create HBase and HDFS snapshots using Cloudera Manager or by using the command-line.

- HBase snapshots allow you to create point-in-time backups of tables without making data copies, and with minimal impact on RegionServers. HBase snapshots are supported for clusters running CDH 4.2 or higher.
- HDFS snapshots allow you to create point-in-time backups of directories or the entire filesystem without actually cloning the data. They can 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.

## Using snapshots with replication

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.

A directory is snapshottable because it has been enabled for snapshots, or because a parent directory is enabled for snapshots. Subdirectories of a snapshottable directory are included in the snapshot.

## Snapshot policies in Replication Manager

You can create snapshot policies in Replication Manager that define the directories or tables to be snapshotted, the intervals at which snapshots should be taken, and the number of snapshots that should be kept for each snapshot interval.

For example, you can create a snapshot policy that takes daily and weekly snapshots, and specify that seven daily snapshots and five weekly snapshots should be maintained.

Minimum Required Role: [Replication Administrator](#) (also provided by Full Administrator)

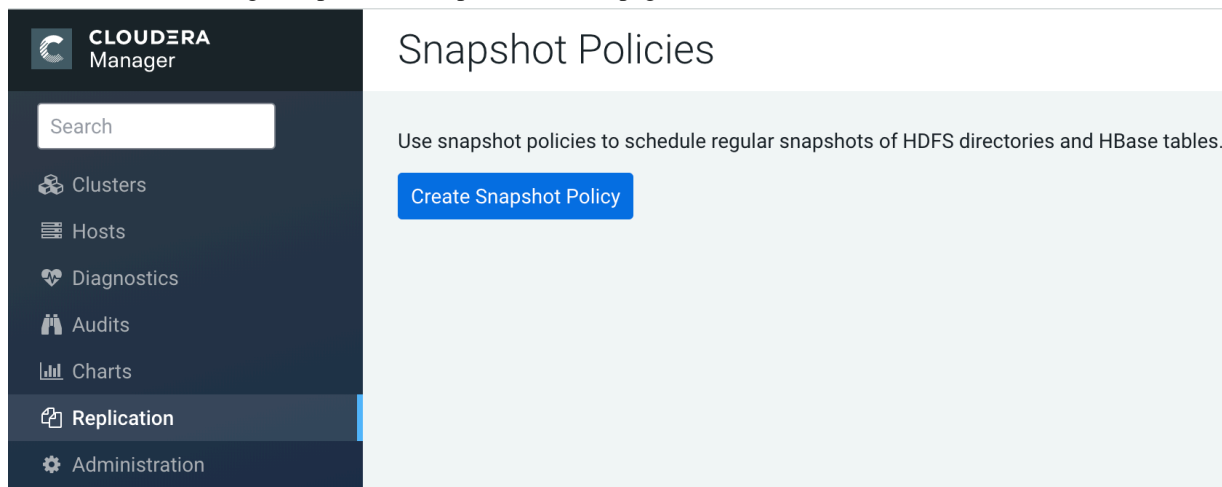
You can improve the reliability of replication policies by using snapshots.

## Creating and managing snapshot policies

You must enable an HDFS directory for snapshots in Cloudera Manager and then create snapshot policies for that directory in Replication Manager.

### Create a snapshot policy

1. Go to Cloudera Manager Replication Snapshot Policies page.



Existing snapshot policies appear in this page.

2. Click Create Snapshot Policy.
3. Select HDFS or HBase service, and the cluster for which you want to create a snapshot policy.
4. Provide a name for the snapshot policy.



**Note:** 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.

5. Optionally, provide a description.
6. Specify the directories, namespaces or tables to include in the snapshot.



**Important:** Do not take snapshots of the root directory.

- For an HDFS service, select the paths of the directories to include in the snapshot. The drop-down list allows you to select only directories that are enabled for snapshotting. If no directories are enabled for snapshotting, a warning appears.

Click **+** to add a path and **=** to remove a path.

- For an HBase service, 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.
7. Specify the snapshot Schedule. 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.

To select an interval, check its box. Fields display where you can edit the time and number of snapshots to keep. For example:



8. Specify whether Alerts should be generated for various state changes in the snapshot workflow. You can alert on failure, on start, on success, or when the snapshot workflow is aborted.
9. Click Save Policy.

The new snapshot policy appears on the Snapshot Policies page.

### Manage a snapshot policy

You can edit or delete a snapshot policy.

1. Go to Cloudera Manager Replication Snapshot Policies page.
2. Click Actions Edit to edit the snapshot policy.
3. Click Actions Delete to delete the snapshot policy.

Errors might appear when you edit or delete a snapshot policy that contains the characters % ; / \ 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 policy name in the `SNAPSHOT_POLICIES` table, perform the following steps:

1. Take a backup of the Cloudera Manager database.
2. Run the following command to replace the unsupported character in the policy name with an underscore:

```
update SNAPSHOT_POLICIES set NAME = replace(NAME,CHAR([***Enter character number***]),'_');
```

## Snapshots history

The Snapshots History page shows information about the snapshot jobs that have been run or attempted.

The **Snapshots History** page shows a table of snapshot jobs and the following columns:

**Table 4: 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 Processed	HDFS snapshots: the number of Paths Processed for the snapshot. HBase snapshots: the number of Tables Processed for the snapshot.
Paths   Tables Unprocessed	HDFS snapshots: the number of Paths Unprocessed for the snapshot. HBase snapshots: the number of Tables Unprocessed for the snapshot.
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.

## 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
<code>/data/folder1/folder2/sales/[table1, table2, table3 ... tablen]</code>	<code>/data/folder1/folder2/sales</code>
<code>/data/folder1/folder2/marketing/[table1, table2, table3 ... tablen]</code>	<code>/data/folder1/folder2/marketing</code>
<code>/data/folder1/folder2/hr/[table1, table2, table3 ... tablen]</code>	<code>/data/folder1/folder2/hr</code>

## 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.

## 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 HDFS snapshots

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.

## Restoring HDFS snapshots

Before you restore from a 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 be 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.

## 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_container_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 banned.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 [***superuser or hdfs***] hadoop distcp [***Source cluster***] [***Destination cluster***]
```

For example,

```
sudo -u <superuser> hadoop distcp hdfs://nn1:8020/source hdfs://nn2:8020/destination
```