Cloudera Replication Manager

Replication Manager Reference

Date published: 2019-11-07
Date modified: 2024-02-28

https://docs.cloudera.com/

Legal Notice

Contents

CDP CLI for Replication Manager

You can use CDP CLI commands to create and manage HDFS and Hive replication policies in Replication Manager. You can also register the ABFS and AWS cloud credentials to use in Replication Manager. The CDP CLI commands for Replication Manager are under the "replicationmanager" CDP CLI option.

Prerequisites

To use CDP CLI commands for Replication Manager, ensure that the following are available:

1. CDP CLI client.

   For information about installing a CDP CLI client, see Installing CDP CLI client.
2. Access to CDP CLI.

   Choose one of the following methods to log into CDP CLI:

   - Interactive method. This login method grants a 12-hour access key to the CLI. For more information, see Logging into CLI/SDK.
   - Traditional method. In this method, you generate access credentials and configure the ~/.cdp/credentials file with the key pair. This login method allows you to withdraw the access permission by removing the access credentials from the ~/.cdp/credentials file. For more information, see Generating an API access key and Configuring CDP client with the API access key.

Access CLI help

CDP CLI includes help that can be accessed by using the cdp help command. To get more information about a certain CDP CLI, you can use cdp *[***module-name***] [***command-name***]* help command.

You can also find all of the CDP CLI commands in the CDP CLI Reference documentation.

Related Information

Introduction to Replication Manager

Support matrix for Replication Manager on CDP Public Cloud

Using HDFS replication policy

Using Hive replication policy

CDP CLI options for Replication Manager

You can use the CDP CLI commands to create, activate, and delete HDFS and Hive replication policies.

CDP CLI options

You can use the following CDP CLI options to perform tasks in Replication Manager:

CDP CLI option	Description
abort-policy*	Aborts all running instances of the specified replication policy. Provide the CRN of the cluster where the replication policy is created and stored.
activate-hbase-policy*	Resumes the specified suspended HBase replication policy. If the target Cloudera Manager API version is higher than 50, the command resumes all the HBase replication policies between the same source and destination cluster. However, if the target Cloudera Manager API version is lower than 45, the command resumes only the specified HBase replication policy. This command is not available for target Cloudera Manager API versions between 45 and 50.

CDP CLI option	Description
activate-policy	Resumes the specified suspended HDFS or Hive replication policy. Provide source CRN for HDFS replication policies and target CRN for Hive replication policies.
activate-snapshot-policy*	Resumes a suspended snapshot policy run.
collect-diagnostic-bundle	Triggers the diagnostic bundle collection for the specified HDFS or Hive replication policy in the target Cloudera Manager. Use the commandId in the command output to download the diagnostic bundle.
continue-hbase-setup*	Continues the setup of an HBase replication policy. **Note:** If an HBase replication policy has a classic cluster source, the HBase service on the source is not restarted automatically. You must restart the source manually and then run the continue-hbase-setup command. This command is not required if the source cluster is not a classic cluster.
create-abfs-credential	Adds a ABFS cloud credentials to use in Replication Manager. **Important:** Before you register the ABFS cloud credentials in Replication Manager, ensure that you update cloud credentials in the source cluster Cloudera Manager UI. For more information, see Registering ABFS Cloud account in Replication Manager.
create-aws-credential	Adds AWS cloud credentials to use in Replication Manager. **Important:** Before you add AWS credentials, see Registering Amazon S3 cloud account.
create-gcs-credential*	Adds GCS cloud credentials to use in Replication Manager.
create-hbase-policy*	Creates an HBase replication policy with the given name on the specified cluster.
create-policy	Creates an HDFS or Hive replication policy based on the provided parameters. Provide source CRN for HDFS replication policies and target CRN for Hive replication policies. **Important:** Only a non-machine user can run this CDP CLI command. Otherwise, an HTTP 500 error appears.
create-snapshot-policy*	Creates a HDFS or HBase snapshot policy depending on the details you provide in the snapshot definition.
delete-credential	Deletes the registered credentials from Replication Manager.
delete-hbase-policy*	Deletes the specified HBase replication policy permanently. Enter `--force` if a normal delete fails. For example, when the source cluster is unreachable.
delete-policy	Deletes the specified HDFS or Hive replication policy. Provide source CRN for HDFS replication policies and target CRN for Hive replication policies.
delete-snapshot-policy*	Deletes the specified snapshot policy.
download-diagnostic-bundle	Generates a bundleFile, a binary string in base64 encoded format, as a ZIP file. Run a script to save the response as a file to your machine. For example, `cat response.json | jq -r '.bundleFile' | base64 -D > bundle.zip`. The file is generated only if the bundleStatus in the `get-command-status` command shows *DOWNLOADABLE WITH CLI*.
get-cluster-config*	Retrieves configuration of a specific cluster.

CDP CLI option	Description
get-command-status	Returns the current status of the collect-diagnostic-bundle command. You can also view the status of any Cloudera Manager command using the relevant input command ID.
get-credentials	Returns the cloud credentials that are available on the specified cluster. If you provide the credential name and credential ID, the ID is considered by the CDP CLI command.
get-hbase-time-series*	Returns the time series data for an HBase replication peer depending on the provided parameters.
get-snapshot-policy*	Retrieves the details about the specified snapshot policy.
list-all-credentials	Provides a detailed list of cloud credentials across all clusters that are available for Replication Manager.
list-cluster-service-statuses	Provides the current status of the services on all the clusters that are available for Replication Manager.
list-clusters	Provides a detailed list of all the clusters that are available for Replication Manager.
list-paired-hbase-clusters*	Lists the paired clusters to use for HBase replication policies. The first-time setup is complete for paired clusters.
list-policies	Lists the available replication policies across all the clusters. Enter the source cluster CRN to view the HDFS replication policies and target cluster CRN to view all the Hive replication policies.
list-policy-jobs*	Returns the list of jobs triggered by the replication policy. This command is a paginated operation, therefore multiple API calls might be required to retrieve the entire dataset of results. You can use one of the following methods to retrieve the results: Use --no-paginate argument to disable pagination. Use --max-items *[***total number of items to return***]* --starting-token *[***token to start paginating***]* --page-size *[***page size***]* to return a specific number of jobs at a time.
list-snapshot-policies*	Returns a list of all the snapshot policies across all available clusters.
list-snapshot-policy-jobs*	Retrieves the snapshot history of the specified snapshot policy.
repair-hbase-policy*	Runs the last failed command for a failed HBase replication policy.
rerun-policy*	Runs the specified replication policy.
restore-snapshot*	Restores the specified snapshot.
retry-failed-hbase-first-time-setup*	Runs the first time setup configuration between the clusters in the specified HBase replication policy if the first time setup failed for the replication policy. You can use the following arguments as required: • --machine-user *[***enter username and password of machine user***]* Syntax: user=*[***string***]*,password=*[***string***]*, createUser=*[***enter 'true' to create a new machine user, or 'false' to return an error if the user does not exist***]* • --target-restart-type *[***enter RESTART or ROLLING_RESTART to specify the restart type***]* - Only a non-machine user can run the replicationmanager create-policy CDP CLI command to create a replication policy. • --source-restart-type *[***enter RESTART or ROLLING_RESTART to specify the restart type***]*
retry-failed-hbase-snapshots*	Reruns only the failed initial snapshots in the replication policy if the policy failed to replicate existing data in some or all the tables.
suspend-hbase-policy*	Pauses an active HBase replication policy.

CDP CLI option	Description
suspend-policy	Stops all the replication tasks defined for the HDFS or Hive replication policy. Provide source CRN for HDFS replication policies and target CRN for Hive replication policies.
suspend-snapshot-policy*	Suspends the specified snapshot policy.
update-abfs-credential*	Updates the ABFS cloud credentials.
update-aws-credential*	Updates the AWS cloud credentials for Replication Manager.
update-gcs-credential*	Updates the GCS cloud credentials for Replication Manager.
update-hbase-policy*	Modifies an existing HBase replication policy. You can change the policy name, policy description, and also delete one or more tables from the replication policy.
update-policy*	Modifies the replication policy. To modify the replication policy, make appropriate changes in the JSON file, save the file, and use it in the update-policy command. Use the rerun-policy command to run the policy. Some elements that you cannot modify include cloud credential, source or target cluster, or source dataset. Create another replication policy to use the new values.
update-snapshot-policy*	Modifies an existing snapshot policy depending on the arguments you choose.
verify-hbase-cluster-pair*	Verifies whether the specified pair of clusters are paired, not paired, or wrongly paired. The wrongly paired status indicates that one or both of the specified clusters are paired to other clusters.
*The option is a technical preview feature and is not ready for production deployment. The components are provided 'as is' without warranty or support. Further, Cloudera assumes no liability for the use of preview components, which should be used by customers at their own risk. For more information, contact your Cloudera account team.	

CDP CLI options to create a replication policy

The following parameters are available in the replicationmanager create-policy CDP CLI option:

Parameter	Description
--cluster-crn	Enter the cluster CRN. To determine the cluster CRN, run the list-clusters command. Provide the following CRN when you create a replication policy: • Source cluster CRN for HDFS replication policy. • Target cluster CRN for Hive replication policy.
--policy-name	Enter a unique name for the replication policy.
--policy-definition	Enter the policy definition in the $(cat *[***policy definition file name***]*) format, and then enter the cluster CRN and policy name as command arguments.
--cli-input-json	Enter the policy definition JSON file path using the cat command to read the data from the file to create and run the replication policy.
--generate-cli-skeleton	Shows a policy definition template in JSON format. You can copy the output of this command to a file, add the required parameters, and save it as a JSON file. You can use the filename while creating a replication policy.

Related Information

Technical preview CDP CLI options for Replication Manager

Adding cloud credentials in Replication Manager using CDP CLI

To replicate data to a storage cloud account, you must register the cloud credentials, so that the Replication Manager can access your cloud account. The supported cloud storage accounts are Amazon S3 and Azure Blob Filesystem (ABFS). You can add, update, or delete AWS or ABFS cloud credentials to use in Replication Manager using CDP CLI.

About this task

Perform one of the following steps, as necessary, to manage cloud credentials in Replication Manager using CDP CLI:

Procedure

- To add ABFS credentials, run the following command:

  replicationmanager create-abfs-credential --name *[***credential name***]* --clusters *[***cluster crns separated by space***]* --type *[***ACCESSKEY or CLIENTKEY***]* --access-key *[***ABFS access key***]* --storage-account-name *[***ABFS storage account name***]* --client-id *[***client ID of Active Directory service principal account***]* --client-secret-key *[***client Key of Active Directory service principal account***]* --tenant-id *[***tenant ID of Active Directory service principal account***]*

  Note:

  - Enter the access key and storage account name if you choose ACCESSKEY type.
  - Enter the client ID, client secret key, and tenant ID if you choose CLIENTKEY type.

  Before you add ABFS credentials, see Registering ABFS Cloud account in Replication Manager.
- To add AWS credentials, run the following command:

  replicationmanager create-aws-credential --name *[***credential name***]* --clusters *[***cluster crns separated by space***]* --type *[***IAM or ACCESSKEY***]* --access-key *[***AWS access key***]* --secret-key *[***AWS secret key***]*

  Important: Before you add AWS credentials, see Registering Amazon S3 cloud account.
- To update ABFS credentials, run the following command:

  replicationmanager update-abfs-credential --name *[***credential name***]* --type *[***enter ACCESSKEY or CLIENTKEY***]* --access-key *[***abfs access key***]* --storage-account-name *[***abfs storage account name***]* --client-id *[***abfs client id***]* --client-secret-key *[***abfs client secret key***]* --tenant-id *[***abfs tenant id]*
- To update AWS credentials, run the following command:

  replicationmanager update-aws-credential --name *[***credential name***]* --type *[***enter IAM or ACCESSKEY***]* --access-key *[***abfs access key***]* --secret-key *[***aws secret key***]*

  Note:

  - No parameters are required for IAM type.
  - Enter the access key and secret key if you choose ACCESSKEY type.
- To view all the available registered credentials in a cluster, run the following command:

  replicationmanager get-credentials --cluster-crn *[***cluster crn***]*

  Optionally, you can use the --credential-name *[***credential name***]* or --credential-id *[***credential id***]* option to view specific credentials.
- To delete a registered credential from Replication Manager, run the following command:

  replicationmanager delete-credential --name *[***credential name***]*

Creating HDFS replication policy using CDP CLI

You can use CDP CLI to create an HDFS replication policy. Only a non-machine user can run the "replicationmanager create-policy" CDP CLI command to create a replication policy.

Procedure

1. Log into Replication Manager CDP CLI setup using the cdp --profile *[***profile-name***]* replicationmanager command.
2. List the clusters to verify whether the required clusters are available using the cdp --profile *[***profile-name***]* replicationmanager list-clusters command.
3. Verify whether the required services are running on the source cluster using the cdp --profile *[***profile-name***]* replicationmanager list-cluster-service-statuses command.
4. Ensure that the cloud credentials are available using the cdp --profile *[***profile-name***]* replicationmanager list-all-credentials command.
5. Create a policy definition JSON file.

   To accomplish this task, perform the following steps:

   a) Open a policy definition JSON file, or copy the output of the cdp --profile *[***profile_name***]* replicationmanager create-policy --generate-cli-skeleton command to a JSON file to generate a policy definition JSON file.

      For example, `cdp --profile hdfs1 replicationmanager create-policy --generate-cli-skeleton > rm_hdfs1.json`

   b) Enter the required parameters.

   c) Save the file.

   Note: Remove the key-value pairs that are not required in the policy definition JSON file for the specific policy. For example, remove the hiveArguments key-value pairs when you create an HDFS policy.

6. Run the cdp --profile *[***profile_name***]* replicationmanager create-policy --cli-input-json *[****policy definition json file path using cat***]* command to create the HDFS replication policy.

   Important: You must use the cat command to read the data from the policy definition JSON file.

   For example: cdp --profile local-dev replicationmanager create-policy --cli-input-json “$(cat temp/rm_hdfs1.json)”

What to do next

Use one of the following methods to verify whether the replication policy is running as expected:

- Run the cdp --profile *[***profile_name***]* replicationmanager list-policies --cluster-crn *[***CRN of the cluster where the replication policies are stored***]* command.
- View the policy status on the **Replication Policies** page in Replication Manager.

Related Information

Creating HDFS replication policy using Replication Manager

HDFS replication policy definition JSON file

The policy definition JSON file contains all the parameters required to create an HDFS replication policy. When you edit the file to define an HDFS replication policy, remove the parameters that are not required for the replication policy.

Parameters in HDFS replication policy definition JSON file

The following table lists the parameters in the policy definition JSON file that are required for an HDFS replication policy:

Parameter	Description
clusterCrn	Enter the source cluster CRN for HDFS replication policy. Replication Manager saves the replication policy in the specified cluster CRN.
policyName	Enter a unique name for the replication policy.
name	Enter the unique name for the policy.
type	Enter FS to create an HDFS replication policy.
path	Enter the HDFS file path in the source cluster.
mapReduceService	Enter the MapReduce or YARN service for the replication policy to use.
logPath	Enter an alternate path for the logs, if required.
replicationStrategy	Enter STATIC or DYNAMIC to determine whether the file replication tasks should be distributed among the mappers statically or dynamically. The default is DYNAMIC. Static replication distributes file replication tasks among the mappers up front to achieve an uniform distribution based on the file sizes. Dynamic replication 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.
skipChecksumChecks	Enter true to skip checksum checks. The default is true. Checksums are used to perform the following tasks: • To skip replication of files that have already been copied. When set to true, the replication job skips copying a file if the file lengths and modification times are identical between the source and destination clusters. Otherwise, the job copies the file from the source to the destination. • To redundantly verify the integrity of data. However, checksums are not required to guarantee accurate transfers between clusters. HDFS data transfers are protected by checksums during transfer and storage hardware also uses checksums to ensure that data is accurately stored. These two mechanisms work together to validate the integrity of the copied data.
skipListingChecksumChecks	Enter true to skip checksum check while comparing two files to determine whether they are the same or not. Otherwise, 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:** If you set skipChecksumChecks to false, the skipListingChecksumChecks is also set to false by default.
abortOnError	Enter true to stop the policy job when an error occurs. This ensures that the files copied up to that point remain on the destination, but no additional files are copied. The default is false.
abortOnSnapshtDiffFailures	Enter true to stop the replication job if a snapshot diff fails during replication.

Parameter	Description
preserve	Enter true to preserve the block size, replication count, permissions (including ACLs), and extended attributes (XAttrs) as they exist on the source file system. • blockSize • replicationCount • permissions • extendedAtrributes Enter false to use the settings as configured on the destination file system. By default, the source system settings are preserved. In an HDFS replication policy, when the permissions parameter is set to true and both the source and destination clusters support ACLs, replication preserves ACLs. Otherwise, ACLs are not replicated. When extendedAtrributes is set to true and both the source and destination clusters support extended attributes, the replication process preserves them. If you select one or more of the Preserve options and you are replicating to S3 or ADLS, the values of all of these items are saved in metadata files on S3 or ADLS. When you replicate from S3 or ADLS to HDFS, you can set the options you want to preserve.
deletePolicy	Enter one of the following options: • KEEP_DELETED_FILES - Retains the destination files even when they no longer exist at the source. This is the default option. • DELETE_TO_TRASH - Moves files to the trash folder if the HDFS trash is enabled. (Not supported when replicating to S3 or ADLS.) • DELETE_PERMANENTLY - Uses the least amount of space; use with caution.
alerts	Configure the following parameters as required: • onFailure - Enter true to generate alerts when the replication job fails. • onStart - Enter true to generate alerts when the replication job starts. • onSuccess - Enter true to generate alerts when the replication job completes successfully. • onAbort - Enter true to generate alerts when the replication job is aborted.
exclusionFilters	Enter one or more directory paths to exclude from replication.
frequencyInSec	Auto-populated after the policy runs successfully. Shows the time duration between two replication jobs in seconds.
targetDataset	Auto-populated after the policy runs successfully. Shows the target location where the replicated files are available on the target cluster.
cloudCredentials	Enter the cloud credentials.
sourceCluster	Shows the source cluster name.
targetCluster	Shows the target cluster name in the dataCenterName$cluster name format. For example, "DC-US$My Destination 17".
startTime	Shows the start time of the replication job in the YYYY-MM-DDTHH:MM:SSZ format.
endTime	Shows the end time of the replication job in the YYYY-MM-DDTHH:MM:SSZ format.
distcpMaxMaps	Enter the maximum map slots to limit the number of map slots per mapper. The default value is 20.
distcpMapBandwidth	Enter the maximum bandwidth to limit the bandwidth per mapper. The default is 100 MB.
queueName	Enter the YARN queue name if not set to Default queue name. By default, the Default queue name is used.

Parameter	Description
tdeSameKey	Enter true if the source and destination are encrypted with the same TDE key.
description	Enter a description for the policy.
enableSnapshotBasedReplication	Enter true to enable snapshot-based replication.
cloudEncryptionAlgorithm	Enter the cloud encryption algorithm.
cloudEncryptionKey	Enter the cloud encryption key.
plugins	Enter the plugins to deploy on all the nodes in the cluster if you have multiple repositories configured in your environment.
cmPolicySubmitUser	Enter the following options: • userName - Enter the user name that you are using to run the policy. • sourceUser - Enter the source cluster username, if any.

Sample HDFS replication policy definition JSON file

The following snippet shows the contents of the HDFS replication policy definition JSON file. While editing the file, ensure that you remove the key-value pairs that are not required for the HDFS replication policy. For example, remove the hiveArguments key-value pairs when you create a HDFS replication policy.

```
{
 "name": "string",
 "type": "FS"|"HIVE",
 "sourceDataset": {
  "hdfsArguments": {
   "path": "string",
                     "mapReduceService": "string",
                     "logPath": "string",
   "replicationStrategy": "DYNAMIC"|"STATIC",
   "errorHandling": {
    "skipChecksumChecks": true|false,
    "skipListingChecksumChecks": true|false,
    "abortOnError": true|false,
    "abortOnSnapshotDiffFailures": true|false
   },
   "preserve": {
    "blockSize": true|false,
    "replicationCount": true|false
    "permissions": true|false,
    "extendedAttributes": true|false
   },
   "deletePolicy": "KEEP_DELETED_FILES"|"DELETE_TO_TRASH"|"DELETE_PERMANENTL
Y",
   "alerts": {
    "onFailure": true|false,
    "onStart": true|false,
    "onSuccess": true|false,
    "onAbort": true|false
   },
   "exclusionFilters": ["string", ...]
   },
   "hiveArguments": {
    "databasesAndTables": [
     {
      "database": "string",
      "tablesIncludeRegex": "string",
      "tablesExcludeRegex": "string",
     }
```

```
      ...
    ],
    "sentryPermissions": "INCLUDE"|"EXCLUDE",
    "skipUrlPermissions": true|false,
    "numThreads": integer
   }
  },
  "frequencyInSec": integer,
  "targetDataset": "string",
  "cloudCredentials": "string",
  "sourceCluster": "string",
  "targetCluster": "string",
  "startTime": "string",
  "endTime": "string",
  "distcpMaxMaps": integer,
  "distcpMapBandwidth": integer,
  "queueName": "string",
  "tdeSameKey": true|false,
  "description": "string",
  "enableSnapshotBasedReplication": true|false
  "cloudEncryptionAlgorithm": "string",
  "cloudEncryptionKey": "string",
  "plugins": ["string", ...],
  "hiveExternalTableBaseDirectory": "string",
  "cmPolicySubmitUser": {
   "userName": "string",
   "sourceUser": "string"
  }
 }
```

Managing HDFS replication policies using CDP CLI

You can use CDP CLI to perform various actions on a replication policy. You can suspend a running HDFS replication policy job and then activate it. You can also delete a replication policy.

About this task

You can perform the following actions to manage an HDFS replication policy:

Procedure

- Run the replication policy using the following command:

  cdp --profile *[***profile name***]* replicationmanager rerun-policy --cluster-crn *[***target cluster crn***]* --policy-name *[***policy name***]*
- Abort a running policy job using the following command:

  cdp --profile *[***profile name***]* replicationmanager abort-policy --cluster-crn *[***source cluster crn***]* --policy-name *[***policy name***]*
- Suspend a running policy job using the following command:

  cdp --profile *[***profile name***]* replicationmanager suspend-policy --cluster-crn *[***source cluster crn***]* --policy-name *[***policy name***]*
- Activate a suspended policy job using the following command:

  cdp --profile *[***profile name***]* replicationmanager activate-policy --cluster-crn *[***source cluster crn***]* --policy-name *[***policy name***]*

- Update the replication policy using the following command after you update the existing policy definition JSON file for the replication policy:

  cdp --profile *[***profile name***]* replicationmanager update-policy --cluster-crn *[***target cluster crn***]* --policy-name *[***policy name***]* --update-policy-definition *[***policy definition***]*

  Note:
  - Remove the key-value pairs that are not required in the policy definition JSON file for the specific policy.
  - Some key-value pairs cannot be edited, such as policy type, cloud credential, source cluster, target cluster, and source dataset in an existing replication policy. Instead you can create another replication policy with the required key-value pairs.
- Run the following commands in the given sequence to download a diagnostic bundle for the specified replication policy:

  a) cdp --profile *[***profile name***]* replicationmanager collect-diagnostic-bundle --cluster-crn *[***target cluster crn***]* --policy-name *[***policy name***]*

  The command initiates the collection operation of the diagnostic bundle for the specified replication policy on the target Cloudera Manager.

  The following sample snippet shows the command output:

```
{
    "commandId": 58747,
    "name": "Replication Diagnostics Collection",
    "active": true,
    "startTime": "2022-11-07T12:27:25.872Z"
}
```

  b) cdp --profile *[***profile name***]* replicationmanager get-command-status --cluster-crn *[***target cluster crn***]* --policy-name *[***policy name***]*

  The command returns diagnostic bundle collection status as:

  - The diagnostic bundle collection is **IN PROGRESS** on the Cloudera Manager server.
  - The diagnostic bundle collection is complete and can be **DOWNLOADABLE WITH URL** using the URL specified in the *resultDataUrl* field in the command output.
  - The diagnostic bundle collection is complete and can be **DOWNLOADABLE WITH CLI** using the download-diagnostic-bundle CDP CLI operation in Step 3.
  - The diagnostic bundle collection **FAILED** on the server.

  Tip: Optionally, you can use this command to get the current status of any Cloudera Manager command.

  The following sample snippet shows the command output when the bundleStatus is **DOWNLOADABLE WITH CLI**:

```
{
    "commandId": 58741,
    "success": true,
    "active": false,
    "name": "Replication Diagnostics Collection",
    "resultDataUrl": "http://[***cm host***]:[***cm port***]/cmf/comma
nd/58741/download",
    "resultMessage": "Replication diagnostics collection succeeded.",
    "bundleStatus": "DOWNLOADABLE WITH CLI",
    "bundleStatusMessage": "The bundle can be downloaded with the down
load-diagnostic-bundle operation."
```

```
}
```

Tip: The command ID in the output is used in Step 3 to download the bundle.

c) cdp --profile *[***profile name***]* replicationmanager download-diagnostic-bundle --cluster-crn *[***target cluster crn***]* --command-id *[***command ID***]*

Run this command only if the bundleStatus shows **DOWNLOADABLE WITH CLI** in the Step 3 command output. The command output appears as a binary string in base64 encoded format on the screen.

You can use any method to parse the response. Alternatively, you can also use one of the following commands to parse the response:

- cdp --profile *[***profile name***]* replicationmanager download-diagnostic-bundle --cluster-crn *[***target cluster crn***]* --command-id *[***command ID***]* > *[***<file>.json***]*

  The diagnostic bundle is saved in the specified file in JSON format and downloaded to your machine.
- cdp --profile *[***profile name***]* replicationmanager download-diagnostic-bundle --cluster-crn *[***target cluster crn***]* --command-id *[***command ID***]* > *[***<filename>.json***]* | jq -r '.bundleFile' | base64 -D > *[***<filename>.zip***]*

  The diagnostic bundle is saved to the specified ZIP file and downloaded to your machine.
- cdp --profile *[***profile name***]* replicationmanager download-diagnostic-bundle --cluster-crn *[***target cluster crn***]* --command-id *[***command ID***]* > *[***<filename>.json***]* | jq -r '.bundleFile' | base64 -D >| bsdtar -xcf -f *[***location***]*

  The diagnostic bundle is saved as a ZIP file and extracted to the specified location on your machine automatically.

- Delete the replication policy using the following command:

  cdp --profile *[***profile name***]* replicationmanager delete-policy --cluster-crn *[***source cluster crn***]* --policy-name *[***policy name***]*

Creating Hive replication policy using CDP CLI

You can use CDP CLI to create an Hive replication policy. Only a non-machine user can run the "replicationmanager create-policy" CDP CLI command to create a replication policy.

Procedure

1. Log into Replication Manager CDP CLI setup using the cdp --profile *[***profile-name***]* replicationmanager command.
2. List the clusters to verify whether the required clusters are available using the cdp --profile *[***profile-name***]* replicationmanager list-clusters command.
3. Verify whether the required services are running on the source cluster using the cdp --profile *[***profile-name***]* replicationmanager list-cluster-service-statuses command.
4. Ensure that the cloud credentials are available using the cdp --profile *[***profile-name***]* replicationmanager list-all-credentials command.

5. Create a policy definition JSON file.

   To accomplish this task, perform the following steps:

   a) Open a policy definition JSON file, or copy the output of the cdp --profile *[***profile_name***]* replicationmanager create-policy --generate-cli-skeleton command to a JSON file to generate a policy definition JSON file.

      For example, `cdp --profile hive1 replicationmanager create-policy --generate-cli-skeleton > rm_hive1.json`

   b) Enter the required parameters.

   c) Save the file.

   Note: Remove the key-value pairs that are not required in the policy definition JSON file for the specific policy.

6. Run the cdp --profile *[***profile_name***]* replicationmanager create-policy --cli-input-json *[****policy definition json file path using cat***]* command to create the replication policy.

   Important: You must use the cat command to read the data from the policy definition JSON file.

   For example: cdp --profile local-dev replicationmanager create-policy --cli-input-json “$(cat temp/rm_hive1.json)”

What to do next

Use one of the following methods to verify whether the replication policy is running as expected:

- Run the cdp --profile *[***profile_name***]* replicationmanager list-policies --cluster-crn *[***CRN of the cluster where the replication policies are stored***]* command.
- View the policy status on the **Replication Policies** page in Replication Manager.

Related Information

Creating Hive replication policy using Replication Manager

Hive replication policy definition JSON file

The policy definition JSON file contains all the parameters required to create a Hive replication policy. When you edit the file to define a Hive replication policy, remove the parameters that are not required for the replication policy.

Parameters in Hive replication policy definition JSON file

The following table lists the parameters in the policy definition JSON file that are required for a Hive replication policy:

Parameter	Description
name	Enter the unique name for the policy.
type	Enter HIVE to create a Hive replication policy.
mapReduceService	Enter the MapReduce or YARN service for the replication policy to use.
logPath	Enter an alternate path for the logs, if required.

Parameter	Description
replicationStrategy	Enter STATIC or DYNAMIC to determine whether the file replication tasks should be distributed among the mappers statically or dynamically. The default is DYNAMIC. Static replication distributes file replication tasks among the mappers up front to achieve an uniform distribution based on the file sizes. Dynamic replication 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.
skipChecksumChecks	Enter true to skip checksum checks. The default is true. Checksums are used to perform the following tasks: • To skip replication of files that have already been copied. When set to true, the replication job skips copying a file if the file lengths and modification times are identical between the source and destination clusters. Otherwise, the job copies the file from the source to the destination. • To redundantly verify the integrity of data. However, checksums are not required to guarantee accurate transfers between clusters. HDFS data transfers are protected by checksums during transfer and storage hardware also uses checksums to ensure that data is accurately stored. These two mechanisms work together to validate the integrity of the copied data.
skipListingChecksumChecks	Enter true to skip checksum check while comparing two files to determine whether they are the same or not. Otherwise, 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:** If you set skipChecksumChecks to false, the skipListingChecksumChecks is also set to false by default.
abortOnError	Enter true to stop the policy job when an error occurs. This ensures that the files copied up to that point remain on the destination, but no additional files are copied. The default is false.
abortOnSnapshtDiffFailures	Enter true to stop the replication job if a snapshot diff fails during replication.
preserve	Enter true to preserve the block size, replication count, permissions (including ACLs), and extended attributes (XAttrs) as they exist on the source file system. • blockSize • replicationCount • permissions • extendedAtrributes Enter false to use the settings as configured on the destination file system. By default, the source system settings are preserved.
deletePolicy	Enter one of the following options: • KEEP_DELETED_FILES - Retains the destination files even when they no longer exist at the source. This is the default option. • DELETE_TO_TRASH - Moves files to the trash folder if the HDFS trash is enabled. (Not supported when replicating to S3 or ADLS.) • DELETE_PERMANENTLY - Uses the least amount of space; use with caution.

Parameter	Description
alert	Configure the following parameters as required: • onFailure - Enter true to generate alerts when the replication job fails. • onStart - Enter true to generate alerts when the replication job starts. • onSuccess - Enter true to generate alerts when the replication job completes successfully. • onAbort - Enter true to generate alerts when the replication job is aborted.
exclusionFilters	Enter one or more directory paths to exclude from replication.
databasesAndTables	• database - Enter one or more database names to include in replication. • tablesIncludeRegex - Enter one or more regular expression-based paths to tables to include in replication. For example, if you enter table1|table2|table3, Replication Manager includes the specified tables for replication. If you enter DB:db_name Table: (?!table1|table2|table3).+, Replication Manager includes all the tables in the 'db_name' database and excludes 'table1', 'table2', and 'table3' from replication. tablesExcludeRegex is a legacy option. You can enter one or more regular expression-based paths of tables to exclude in replication.
sentryPermissions	Enter INCLUDE to import both Hive object and URL permissions.
skipUrlPermissions	Enter true to import only the Hive object permissions.
numThreads	Enter the number of threads to use during replication.
frequencyInSec	Auto-populated after the policy runs successfully. Shows the time duration between two replication jobs in seconds.
targetDataset	Auto-populated after the policy runs successfully. Shows the target location where the replicated files are available on the target cluster.
cloudCredential	Enter the cloud credentials.
sourceCluster	Shows the source cluster name.
targetCluster	Shows the target cluster name in the dataCenterName$cluster name format. For example, "DC-US$My Destination 17".
startTime	Shows the start time of the replication job in the YYYY-MM-DDTHH:MM:SSZ format.
endTime	Shows the end time of the replication job in the YYYY-MM-DDTHH:MM:SSZ format.
distcpMaxMaps	Enter the maximum map slots to limit the number of map slots per mapper. The default value is 20.
distcpMapBandwidth	Enter the maximum bandwidth to limit the bandwidth per mapper. The default is 100 MB.
queueName	Enter the YARN queue name if not set to Default queue name. By default, the Default queue name is used.
tdeSameKey	Enter true if the source and destination are encrypted with the same TDE key.
description	Enter a description for the policy.
enableSnapshotBasedReplication	Enter true to enable snapshot-based replication.
cloudEncryptionAlgorithm	Enter the cloud encryption algorithm.
cloudEncryptionKey	Enter the cloud encryption key.
plugins	Enter the plugins to deploy on all the nodes in the cluster if you have multiple repositories configured in your environment.

Parameter	Description
hiveExternalTableBaseDirectory	Enter the Hive external table base directory path.
cmPolicySubmitUser	Enter the following options: • userName - Enter the user name that you are using to run the policy. • sourceUser - Enter the source cluster username, if any.

Sample Hive replication policy definition JSON file

The following snippet shows the contents of the Hive replication policy definition JSON file. While editing the file, ensure that you remove the key-value pairs that are not required for the Hive replication policy.

```
{
 "name": "string",
 "type": "HIVE",
 "sourceDataset": {
  "hdfsArguments": {
   "path": "string",
   "mapReduceService": "string",
                      "logPath": "string",
                      "replicationStrategy": "DYNAMIC"|"STATIC",
   "errorHandling": {
    "skipChecksumChecks": true|false,
    "skipListingChecksumChecks": true|false,
    "abortOnError": true|false,
    "abortOnSnapshotDiffFailures": true|false
   },
   "preserve": {
    "blockSize": true|false,
    "replicationCount": true|false
    "permissions": true|false,
    "extendedAttributes": true|false
   },
   "deletePolicy": "KEEP_DELETED_FILES"|"DELETE_TO_TRASH"|"DELETE_PERMANE
NTLY",
   "alert": {
    "onFailure": true|false,
    "onStart": true|false,
    "onSuccess": true|false,
    "onAbort": true|false
   },
   "exclusionFilters": ["string", ...]
   },
   "hiveArguments": {
    "databasesAndTables": [
     {
      "database": "string",
      "tablesIncludeRegex": "string",
      "tablesExcludeRegex": "string",
     }
     ...
    ],
    "sentryPermissions": "INCLUDE"|"EXCLUDE",
    "skipUrlPermissions": true|false,
    "numThreads": integer
   }
  },
  "frequencyInSec": integer,
  "targetDataset": "string",
  "cloudCredential": "string",
  "sourceCluster": "string",
```

```
  "targetCluster": "string",
  "startTime": "string",
  "endTime": "string",
  "distcpMaxMaps": integer,
  "distcpMapBandwidth": integer,
  "queueName": "string",
  "tdeSameKey": true|false,
  "description": "string",
  "enableSnapshotBasedReplication": true|false
  "cloudEncryptionAlgorithm": "string",
  "cloudEncryptionKey": "string",
  "plugins": ["string", ...],
  "hiveExternalTableBaseDirectory": "string",
  "cmPolicySubmitUser": {
   "userName": "string",
   "sourceUser": "string"
  }
 }
```

Managing Hive replication policies using CDP CLI

You can use CDP CLI to perform various actions on a replication policy. You can suspend a running Hive replication policy job and then activate it. You can also delete a replication policy.

About this task

You can perform the following actions to manage a Hive replication policy:

Procedure

- Run the replication policy using the following command:

  cdp --profile *[***profile name***]* replicationmanager rerun-policy --cluster-crn *[***target cluster crn***]* --policy-name *[***policy name***]*
- Abort a running policy job using the following command:

  cdp --profile *[***profile name***]* replicationmanager abort-policy --cluster-crn *[***target cluster crn***]* --policy-name *[***policy name***]*
- Suspend a running policy job using the following command:

  cdp --profile *[***profile name***]* replicationmanager suspend-policy --cluster-crn *[***source cluster crn***]* --policy-name *[***policy name***]*
- Activate a suspended policy job using the following command:

  cdp --profile *[***profile name***]* replicationmanager activate-policy --cluster-crn *[***target cluster crn***]* --policy-name *[***policy name***]*
- Update the replication policy using the following command after you update the existing policy definition JSON file for the replication policy:

  cdp --profile *[***profile name***]* replicationmanager update-policy --cluster-crn *[***target cluster crn***]* --policy-name *[***policy name***]* --update-policy-definition *[***policy definition***]*

  Note:

  - Remove the key-value pairs that are not required in the policy definition JSON file for the specific policy.
  - Some key-value pairs cannot be edited, such as policy type, cloud credential, source cluster, target cluster, and source dataset in an existing replication policy. Instead you can create another replication policy with the required key-value pairs.

- Run the following commands in the given sequence to download a diagnostic bundle for the specified replication policy:
  a) cdp --profile *[***profile name***]* replicationmanager collect-diagnostic-bundle --cluster-crn *[***target cluster crn***]* --policy-name *[***policy name***]*

     The command initiates the collection operation of the diagnostic bundle for the specified replication policy on the target Cloudera Manager.

     The following sample snippet shows the command output:

```
{
    "commandId": 58747,
    "name": "Replication Diagnostics Collection",
    "active": true,
    "startTime": "2022-11-07T12:27:25.872Z"
}
```

  b) cdp --profile *[***profile name***]* replicationmanager get-command-status --cluster-crn *[***target cluster crn***]* --policy-name *[***policy name***]*

     The command returns diagnostic bundle collection status as:

     - The diagnostic bundle collection is **IN PROGRESS** on the Cloudera Manager server.
     - The diagnostic bundle collection is complete and can be **DOWNLOADABLE WITH URL** using the URL specified in the *resultDataUrl* field in the command output.
     - The diagnostic bundle collection is complete and can be **DOWNLOADABLE WITH CLI** using the download-diagnostic-bundle CDP CLI operation in Step 3.
     - The diagnostic bundle collection **FAILED** on the server.

     Tip: Optionally, you can use this command to get the current status of any Cloudera Manager command.

     The following sample snippet shows the command output when the bundleStatus is **DOWNLOADABLE WITH CLI**:

```
{
    "commandId": 58741,
    "success": true,
    "active": false,
    "name": "Replication Diagnostics Collection",
    "resultDataUrl": "http://[***cm host***]:[***cm port***]/cmf/comma
nd/58741/download",
    "resultMessage": "Replication diagnostics collection succeeded.",
    "bundleStatus": "DOWNLOADABLE WITH CLI",
    "bundleStatusMessage": "The bundle can be downloaded with the down
load-diagnostic-bundle operation."
}
```

     Tip: The command ID in the output is used in Step 3 to download the bundle.

c) cdp --profile *[***profile name***]* replicationmanager download-diagnostic-bundle --cluster-crn *[***target cluster crn***]* --command-id *[***command ID***]*

Run this command only if the bundleStatus shows **DOWNLOADABLE WITH CLI** in the Step 3 command output. The command output appears as a binary string in base64 encoded format on the screen.

You can use any method to parse the response. Alternatively, you can also use one of the following commands to parse the response:

- cdp --profile *[***profile name***]* replicationmanager download-diagnostic-bundle --cluster-crn *[***target cluster crn***]* --command-id *[***command ID***] > [***<file>.json***]*

  The diagnostic bundle is saved in the specified file in JSON format and downloaded to your machine.
- cdp --profile *[***profile name***]* replicationmanager download-diagnostic-bundle --cluster-crn *[***target cluster crn***]* --command-id *[***command ID***] > [***<filename>.json***]* | jq -r '.bundleFile' | base64 -D > *[***<filename>.zip***]*

  The diagnostic bundle is saved to the specified ZIP file and downloaded to your machine.
- cdp --profile *[***profile name***]* replicationmanager download-diagnostic-bundle --cluster-crn *[***target cluster crn***]* --command-id *[***command ID***] > [***<filename>.json***]* | jq -r '.bundleFile' | base64 -D >| bsdtar -xcf -f *[***location***]*

  The diagnostic bundle is saved as a ZIP file and extracted to the specified location on your machine automatically.

- Delete the replication policy using the following command:

  cdp --profile *[***profile name***]* replicationmanager delete-policy --cluster-crn *[***target cluster crn***]* --policy-name *[***policy name***]*