Cloudera on Premises Data Services 1.5.5

Cloudera Data Services on premises Data Recovery

Date published: 2023-12-16
Date modified: 2025-11-08

https://docs.cloudera.com/

Legal Notice

Contents

Data Recovery Service overview

The Data Recovery Service (DRS) is a microservice in Cloudera Data Services on premises. It allows you to back up and restore Kubernetes namespaces and resources on both Cloudera Embedded Container Service (ECS) and OpenShift Container Platform (OCP) for a few services such as Cloudera Control Plane and Cloudera Data Warehouse (CDW).

1.5.5 SP1 and higher versions provide the following features:

- You can view the Backup and Restore Manager on the Cloudera Management Console Backup Manager Control Plane page.
- You can use the S3 locations in Ozone, in AWS buckets, or in both as an external backup location to back up and restore the Cloudera Control Plane.
- DRS automatically backs up Virtual Warehouse configurations from the database along with the namespace backup. You can restore the backup, when required.

The following sections discuss how to back up and restore Cloudera Control Plane in detail. Contact your Cloudera account team to determine whether your Cloudera service supports DRS, and if so, which components of DRS are being supported.

Cloudera recommends that you create a backup of your Kubernetes namespace before a maintenance activity, before you upgrade, and in general, as a best practice.

Role required: *PowerUser*

By default, DRS is located in the *[***CLOUDERA INSTALLATION NAMESPACE***]*-drs namespace. For example, if the Cloudera Data Services on premises installation is located in the *cdp* namespace, the *drs* namespace is automatically named *cdp-drs*. If you have multiple Cloudera Data Services on premises installations (as in OCP), DRS is named accordingly.

When you initiate the backup event in the Backup and Restore Manager for Cloudera Control Plane, DRS takes a backup of the following resources and data:

- Kubernetes resources associated with the Cloudera namespace and the embedded vault namespaces of the Cloudera Control Plane in Cloudera Data Services on premises. The resources include deployment-related information, stateful sets, secrets, and configmaps.
- Data used by the stateful pods, such as the data in the embedded database and Kubernetes persistent volume claim.

Available methods to back up and restore environment

The following methods are available to back up and restore your environment:

DRS automatic backups

Starting from Cloudera Data Services on premises 1.5.4, DRS automatic back ups for Cloudera Control Plane, CDW, and Cloudera Data Engineering (CDE) are enabled by default on ECS clusters for new installations or after cluster upgrade to version 1.5.4 or higher.

You can disable this option, if required. You can also configure the external storage in Longhorn for ECS, and then initiate the DRS automatic backup to it. For more information, see *DRS automatic backups*.

Note: This functionality is not available for OCP.

Important: Automatic backups help ensure that a backup is available. An automatic backup is a consolidated backup of all the Data Services data, so the restore option is not supported for these backups. Automatic backups use a different code path than the Data Service-specific backups and work at the platform level. Service-specific automatic backups are currently not supported.

For more information, see *DRS automatic backups*.

Service-specific CDP CLI options

You can use the CDP CLI options to back up and restore namespaces for Cloudera Control Plane and CDW.

For the list of available CDP CLI options that you can use for backup and restore purposes, see drscp and dw.

Backup and Restore Manager

You can back up and restore namespaces for Cloudera Control Plane and CDW on the Backup and Restore Manager page.

To access this page, click Cloudera Management Console Backup Manager Control Plane .

External backup locations

You can use S3 locations in Ozone, in AWS buckets, or in both as external backup locations. The Backup Manager uses Kopia to externally back up the persistent volume claims to the Ozone and AWS buckets. The Kopia repository saves the backups with the *cloudera-backups* prefix within the bucket.

Kopia is an open-source backup tool that Backup Manager leverages to back up the Control Plane resources to a Kopia repository, and restore the backup when required.

How backup and restore events work in DRS

Backup event

The backup event does not have any downtime impact, and you can backup the Cloudera Control Plane while it is running.

When you create a backup, DRS:

1. initiates the backup event or job for the chosen backup entity,

   For example, the Cloudera Control Plane in Cloudera Data Services on premises.
2. assigns an ID called backupCrn to the backup event,

   The backupCRN appears in the CRN column on the Backup and Restore Manager Backups tab. Click the CRN to view more details about the backup event on the Backup *[***NAME OF BACKUP***]* modal window.
3. creates a backup of the persistent volume claim (PVC) snapshots of the Cloudera Control Plane namespaces and the backup event's PVC.

Restore event

Note: Do not delete the *[***CLOUDERA INSTALLATION NAMESPACE***]*-drs namespace while the restore event is in progress.

When you start the restore event, DRS:

1. initiates the restore event for the chosen backup,
2. assigns an ID called restoreCrn to the restore event,

   The restoreCRN appears as CRN on the Backup and Restore Manager Restores tab. Click the CRN to view more details about the restore event.

3. deletes the existing resources and data,

   During this stage of the restore event, the ECS restore vault is sealed and the POD is down which might appear as a failure in the Cloudera Control Plane environment. After the restore event is complete, the vault and POD are auto-recovered and restored. Depending on the number of resources and data, this step might take a maximum of 10 minutes to complete.
4. restores the resources and data from the backup.

   The restore event has a downtime impact because the pods and data are recreated.

DRS automatic backups

By default, Cloudera Data Services on premises 1.5.4 and higher versions enable Data Recovery Service (DRS) automatic backups for the Cloudera Control Plane, CDE, and CDW in the compute cluster of ECS. The automatic backups are stored in the Longhorn in-cluster storage. You can also configure the external storage in ECS, and then initiate the automatic back ups to it.

The following storage options are available to store the DRS automatic backups in ECS:

In-cluster storage

By default, DRS automatic backups use Longhorn in-cluster storage. If necessary, you can configure the storage configuration settings in Longhorn by navigating to the Cloudera Manager Clusters *[***CLUSTER NAME***]* Status *[***ECS CLUSTER NAME***]* Web UI Storage UI page.

By default, Kubernetes initiates the first automatic backup within 30 minutes after the backup policy creation is complete, and then takes subsequent backups every hour.

You can change the backup retain count to take backups on an hourly, daily, or weekly basis and you can also disable the DRS automatic backup functionality (set ENABLED to false) using the `kubectl edit cj automatic-backup -n cdp-drs` command. For more information about using this command in DRS, see *Initiating DRS automatic backups*.

External storage

ECS uses Longhorn as the underlying storage provisioner. In Longhorn, you can store snapshots externally using an S3 compatible storage such as Ozone or NFS v4. After you configure the external storage, edit the automatic-backup cron job to initiate the automatic backups.

Important: Automatic backups help ensure that a backup is available. An automatic backup is a consolidated backup of all the Data Services data, and the restore option is not supported for these backups. Automatic backups use a different code path than the Data Service-specific backups and work at the platform level. Service-specific automatic backups are currently not supported.

Configuring external storage in ECS for DRS automatic backups

Before you initiate Data Recovery Service (DRS) automatic backups to the external storage in Longhorn, you must complete the prerequisites.

Procedure

1. Complete the following prerequisites:
   a) Ensure that the following requirements are met depending on the storage you choose for DRS automatic backups:
      - An S3 compatible storage, such as Ozone, must be available in the base cluster. You must have the required access key and secret to the storage, and the provisioned bucket must have a minimum of 5 TB storage space.
      - An NFS v4 storage must have a minimum of 5 TB of free space.
   b) You must have SSH access to the base cluster node.

      Note: If you are using Ozone storage, ensure that you have the SSH access to the base cluster running the S3 gateway service.

   c) You must have SSH access to the ECS master node.

      Tip: After you log into the terminal of the master node, run the following commands to access kubectl utility:

      1. `export PATH="/var/lib/rancher/rke2/bin:/opt/cloudera/parcels/ECS/docker/:$PATH"`
      2. `export KUBECONFIG=/etc/rancher/rke2/rke2.yaml`
2. Perform the following steps to change the default volume snapshot class value from *snap* (this value saves snapshots in the in-cluster storage in Longhorn) to *bak* (this value saves snapshots in the external storage in Longhorn):
   a) Run the kubectl edit vsclass longhorn kubectl command.
   b) Change the type parameter to bak as shown in the following sample snippet:

```
apiVersion: snapshot.storage.k8s.io/v1
deletionPolicy: Delete
driver: driver.longhorn.io
kind: VolumeSnapshotClass
metadata:
name: longhorn
parameters:
type: bak
```

3. Complete the following steps if you are using Ozone S3 storage in Longhorn:
   a) Run the scp root@*[***BASE_CLUSTER_HOST***]*:/var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_cacerts.pem command to obtain the TLS certificate for Ozone.

      DRS uses this certificate to communicate with the S3 gateway service using HTTPS.
   b) Create a secret that Longhorn can use for S3 access. To accomplish this task, you must have the S3 access key, S3 secret, S3 endpoint, and S3 certificate for Ozone storage. You must also enable a virtual host to use the S3 compatible endpoint (Ozone).

      The following sample snippet shows the kubectl command to create a secret:

```
kubectl create secret generic ozone-secret
--from-literal=AWS_ACCESS_KEY_ID=s3g/drs1-1.drs1.root.hwx.site@ROOT.HW
X.SITE
--from-literal=AWS_SECRET_ACCESS_KEY=9d9e46cc77bb510821f0dbc42c584a8b
7482b51dec9d3eb63c
--from-literal=AWS_ENDPOINTS=https://drs1.root.hwx.site:9879/longhorn
--from-literal=VIRTUAL_HOSTED_STYLE=true --from-file=AWS_CERT=cm-auto-g
lobal_cacerts.pem
```

```
-n longhorn-system
```

For more information, see *Longhorn documentation*.

Tip: Longhorn's URL is built using a combination of the AWS_ENDPOINTS value and the S3 virtual path as explained in Step 5.

4. Run the `kubectl edit deploy cdp-release-thunderhead-drsprovider -n cdp-drs` command, and set the TAKE_PVC_CLONE environment value to false.

   This step ensures that the backups do not create a persistent volume claim (PVC) clone for external snapshot.

   By default, Longhorn configuration is set to in-cluster storage and this storage requires a PVC copy to perform the DRS restore operation (DRS uses CSI snapshot technology). Therefore, to use the external storage, you must configure the volume snapshot class to *bak* and then configure the TAKE_PVC_CLONE environment value to false.

5. Configure the volume for NFS storage or bucket for Ozone S3 (on the Setting General page) in the Longhorn UI to save the backups.

   a) Enter the nfs://… URL in the **Backup Target** field if you are using NFS storage.

   b) Enter the required values in the following fields if you are using Ozone S3 storage:

   - s3://*[***BUCKET***]@[***DUMMY REGION***]*/ URL in the **Backup Target** field. For example, s3://drs1-1@cdp/.
   - *[***SECRET THAT YOU GENERATED IN STEP 3B***]* in the **Backup Target Credential Secret**. For example, ozone-secret.

   The s3://*[***BUCKET***]@[***DUMMY REGION***]*/ URL is a virtual S3 URL that you can create using the original Ozone S3 URL, where,

   - bucket is the hostname. Longhorn prefixes the AWS_ENDPOINTS to the bucket value. For example, the sample snippet in Step 3 shows the hostname value as drs1-1.drs1.root.hwx.site. In this instance, drs1-1 is the bucket name and the rest of the hostname drs1.root.hwx.site is the AWS_ENDPOINTS hostname.
   - dummyregion can be any value and is not used.

   Troubleshooting: To verify whether Longhorn successfully registered the Ozone S3 credential secret, click the Backup page. No errors must appear on the page.

   If any error or message appears about the secret and the certificate having newlines or space, run the `kubectl edit lhs backup-target-credential-secret -n longhorn-system` command and set the value to the secret you created in Step 3b.

What to do next

Initiate the DRS automatic backups using the `updateAutoBackupPolicy` CDP CLI command. Alternatively, you can edit the "automatic-backup" (a Kubernetes cron job) to initiate the DRS automatic backups.

Initiating DRS automatic backups

After you configure the external storage in ECS, you can initiate the Data Recovery Service (DRS) automatic backups using the "updateAutoBackupPolicy" CDP CLI command. Alternatively, you can edit the "automatic-backup" (a Kubernetes cron job) to initiate the DRS automatic backups.

Before you begin

The preferred method to initiate the DRS automatic backups is to use the `updateAutoBackupPolicy` CDP CLI command in the CDP client. For more information about DRS CDP CLI commands, see CLI reference for using DRS on Control Plane.

About this task

The following steps show an alternate method to initiate DRS automatic backups using kubectl commands.

Procedure

1. Run the `kubectl edit cj automatic-backup -n cdp-drs` command.
2. Configure the `ENABLED` environment variable to true to enable automatic backups, configure the namespaces (if they are not configured), and then configure the backup retain count to take backups on an hourly, daily, or weekly basis. You can also choose a combination of two or more periods to take backups. Save the cron job.

   The backup retain count determines the number of backup instances to generate.

   Important: If you do not want to set a backup for a particular period, ensure that the count is set to 0. This is because the retain count is set to 1 (minimum backup retain count) by default.

   DRS generates n+1 backups by default where n is the backup retain count. Therefore, the minimum number of backups at any point in time is 2 by default. For example, if you set the HOURLY_COUNT parameter to 2, three instances are generated; therefore, two backups are taken every hour. If you set the WEEKLY_COUNT parameter to 0, no instances are created and no backups are generated.

   The following sample snippet shows the environment variables required for DRS automatic backups:

```
env:
 - name: ENABLED
 value: "true"
 - name: HOURLY_COUNT
 value: "1"
 - name: DAILY_COUNT
 value: "1"
 - name: WEEKLY_COUNT
 value: "1"
```

Results

By default, Kubernetes initiates the first automatic backup within 30 minutes after the backup policy creation is complete.

Tip: You can configure the cron `schedule` environment variable (using kubectl commands or `updateAutoBackupPolicy` CDP CLI command) to control the next job run.

The cron schedule uses the *[***MINUTE FROM 0-59***] [***HOUR FROM 0-23***] [***DAY OF THE MONTH FROM 1-31***] [***MONTH FROM 1-12***] [***DAY OF THE WEEK FROM 0-7 WHERE 7 IS FOR SUNDAY***]* cron syntax.

For example, if you configure schedule: '*/40 * * * *', the backup runs after 40 minutes.

What to do next

Backup instances, depending on the chosen schedules, are generated and appear on the Cloudera Private Cloud Data Services Management Console Dashboard Backup Overview View Details Backup and Restore Manager Backups tab. The instance name is auto-generated. Click the backup instance to view more details.

Backup Manager in Cloudera Management Console

You can backup and restore the Kubernetes namespaces and resources on Cloudera Embedded Container Service (ECS) and OpenShift Container Platform (OCP) for the Control Plane on the "Backup Manager > Control Plane" page in Cloudera Data Services on premises Management Console.

On the Backups tab, you can view existing backup entities, create backups, and perform actions on a backup. On the **Restores** tab, you can view the restore entities.

Tip: Data Recovery Service (DRS) is a microservice in Cloudera Data Services on premises that backs up and restores the Kubernetes namespaces and resources of supported services. Backup Manager leverages the DRS capabilities to backup and restore namespaces in Management Console.

Backup and Restore Manager

The “Backup and Restore Manager” appears after you click Cloudera Data Services on premises Management Console Backup Manager Control Plane section. The "Backups" tab lists all the backups and the "Restores" tab lists all the restore events.

Click New Backup on the **Backup and Restore Manager** page to initiate a backup event. The page shows the backup entity name, and provides the following tabs:

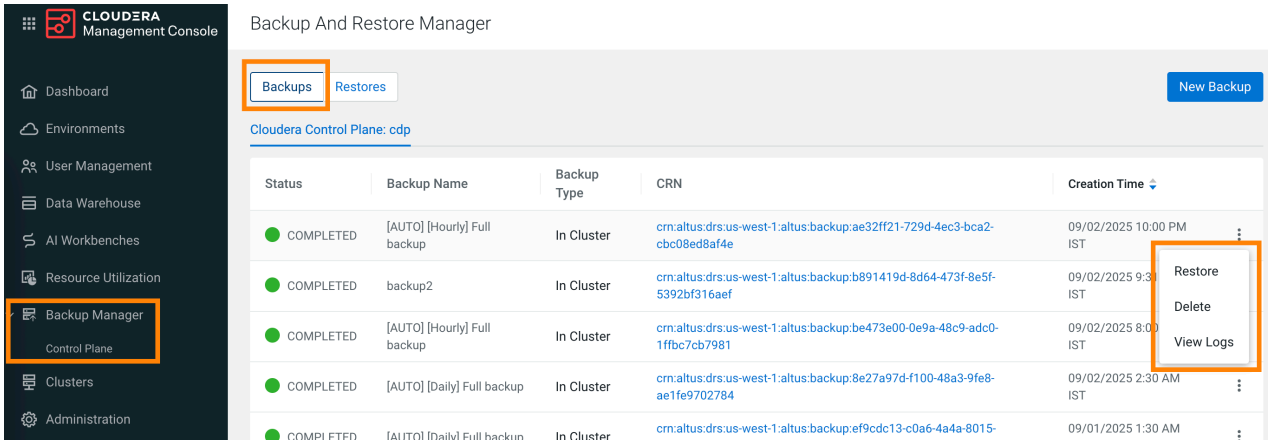
Backups tab

The "Backups" tab lists all the available backups. You can create backups, or perform actions such as restore, delete, or view logs for each back up as necessary.

The following table lists the columns that appear on the **Backups** tab:

Column name	Description
Status	Current backup event status. The event states include NOT_STARTED, IN_PROGRESS, COMPLETED, PARTIALLY_FAILED, and FAILED.
Backup Name	Unique name given to the backup event while initiating the backup event.
Backup Type	Indicates the location type as In Cluster or External for the backup.
CRN	Automatically assigned ID or backupCrn for the backup event. Customer Resource Number (CRN) is the *Cloudera*-specific identifier provided for the event/job. Click the CRN to view more details about the event on the Backup Details modal window.
Creation Time	Timestamp when the backup event was initiated.

The following image shows the **Backups** tab on the **Backup and Restore Manager** page:



You can perform the following actions on each successful backup event:

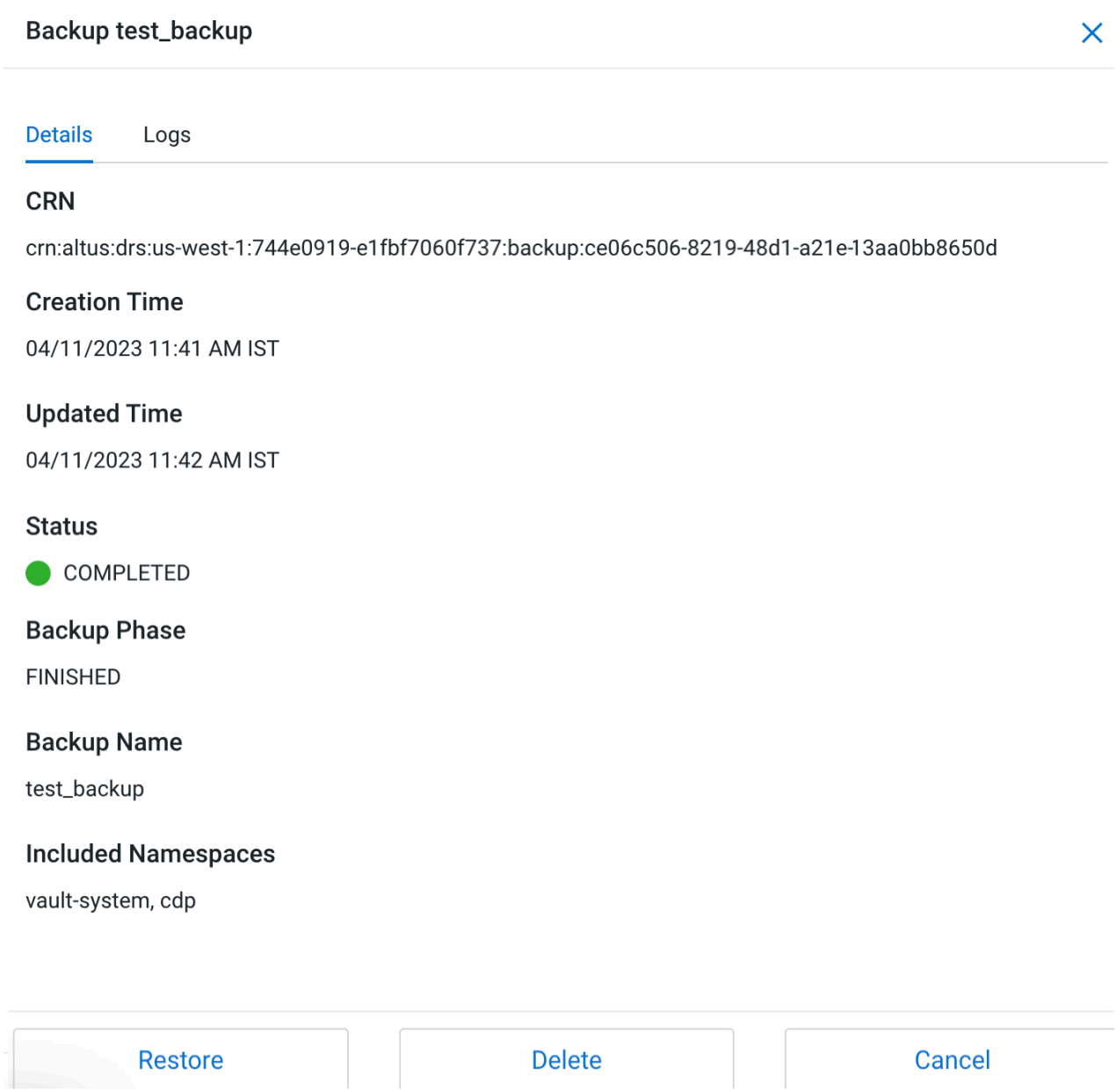
- Restore the backup.
- Delete the backup permanently.
- View Logs opens the Backup Details modal window.

Backup Details modal window

On the Backup Details modal window, you can choose to Restore the backup, Delete the backup, or click Cancel to close the window. The window also shows the following tabs:

Tab	Description
Details	Shows the Backup Name and the following details: • TheCRN of the backup event. • The Creation Time and date when the backup was initiated. • The Updated Time and date when the event was last updated. • The current Status of the backup event appears as NOT_STARTED, IN_PROGRESS, COMPLETED, PARTIALLY_FAILED, and FAILED. • The Backup Phase or current phase of the running backup event appears as PENDING, PRE_VALIDATION, SAVING_OBJECTS, CREATING_SNAPSHOTS, EXTERNALLY_BACKING_UP, and FINISHED. • The Backup Name that was provided for the backup event during the backup creation process. • The Included Namespaces in the backup event.
Logs	Provides the log details about the backup event.

The following image shows the Backup Details modal window:



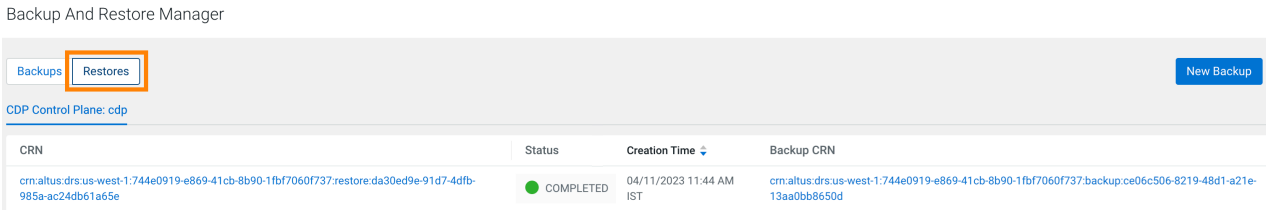
Restores tab

The "Restores" tab lists all the available restore events.

The following table lists the columns that appear on the **Restores** tab:

Column name	Description
CRN	Automatically assigned ID or restoreCrn for the restore event. When you click the CRN, the **Restore Details** modal window appears.
Status	Current restore event status as COMPLETED or FAILED.
Creation Time	Timestamp when the restore event was initiated.
Backup CRN	CRN of the backup event that is being restored. When you click the **Backup CRN**, the Backup Details modal window appears.

The following sample image shows the **Restores** tab:

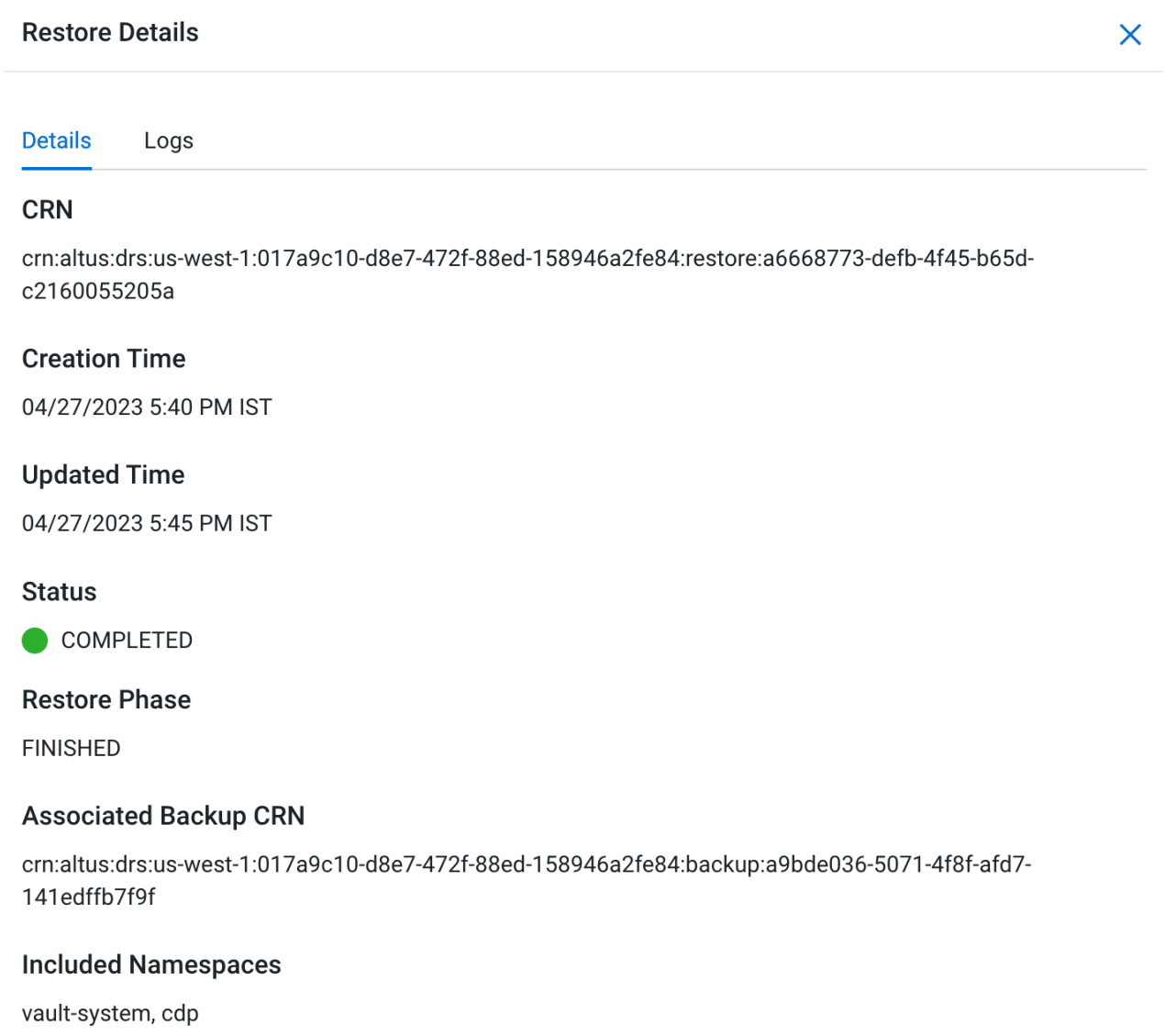


Restore Details modal window

When you click the CRN of a restore event, the following tabs appear on the **Restore Details** modal window:

Column name	Description
Details	• The **CRN** of the restore event. • The **Creation Time** and date of the restore event. • the completed or **Updated Time** and date of the restore event. • The current **Status** of the restore event. • The **Restore Phase** or current phase of the running event shows as IN PROGRESS, PENDING, FAILED, or FINISHED. • The **Associated Backup CRN** of the backup event that was restored. • The **Included Namespaces** in the restore event. • The **Warnings** or **Errors**that might appear for the restore event. When a warning appears, you can continue to use the backup or restore event. However, it is advisable to scrutinize the warning to avoid any potential issues. Errors appear if the restore event has failed.
Logs	Provides the log details about the backup event.

The following sample image shows the Restore Details modal window:



Using an external backup location

You can use S3 locations in Ozone, in AWS buckets, or in both as external backup locations to backup and restore Kubernetes namespaces and resources on Cloudera Embedded Container Service (ECS) and OpenShift Container Platform (OCP) for the Control Plane. You can add the external location in the "Backup and Restore" tab on the "Administration" page

The Backup Manager can use an external location to save the backups after you add the external location in the Cloudera Management Console Administration Backup and Restore tab. The Backup Manager uses Kopia to externally back up the persistent volume claims to the Ozone and AWS buckets. The Kopia repository saves the backups with the *cloudera-backups* prefix within the bucket.

Tip: Kopia is an open-source backup tool that Backup Manager leverages to back up the Control Plane resources to a Kopia repository, and restore the backup when required. Kopia externally backs up the persistent volume claims to the specified bucket. The Kopia repository also saves the backups with the prefix *cloudera-backups* within the bucket.

The following details appear for each external backup location, that you add, on the **Backup and Restore** tab:

- **Creation Status** shows the Kopia repository creation status.

  If the creation status fails, the **Failed** status appears with a View Log link that you can click to view more details about the failure.
- **Location Name** shows the name that you specified during the creation process.
- **External Bucket Name** shows the external bucket name that you specified during the creation process.
- The icon opens the **Edit Location** side panel.

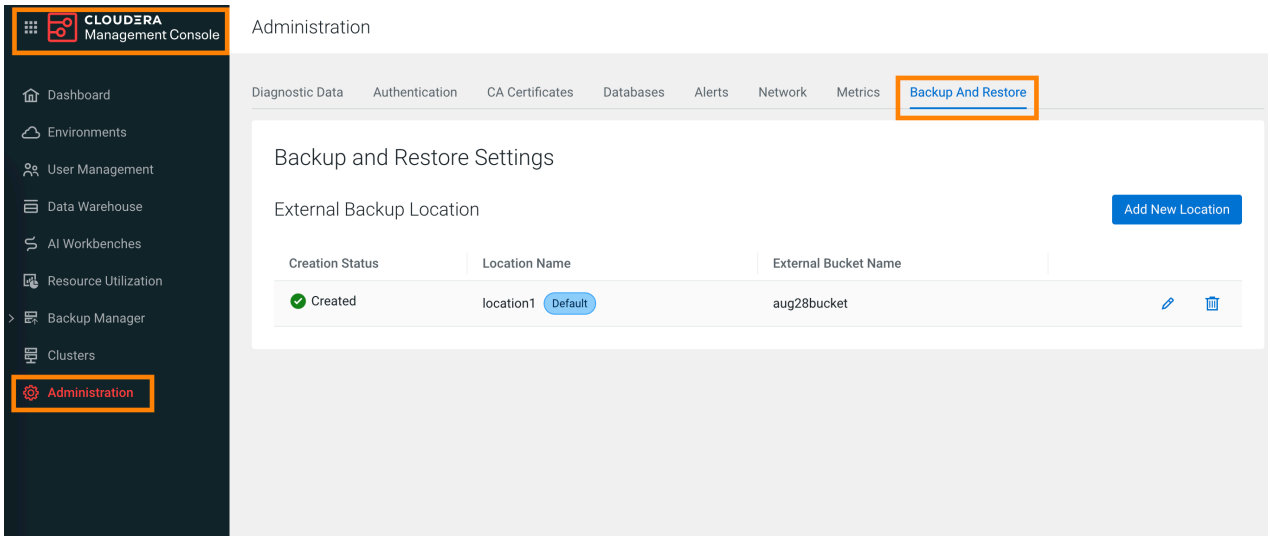
  You can only edit the Access Key and Secret Key of the external location.

  Note: Ensure that you Test Connection before you save the changed details.
- The icon deletes the external location details and clears the Kopia-related data in the bucket.

  If there are any backups in the external location and you click the delete icon, a message popup appears which notifies you to delete the backups before you delete the location.

  Note: This action does not delete any other data that is not related to backup.

The following image shows the Backup and Restore Manager page:

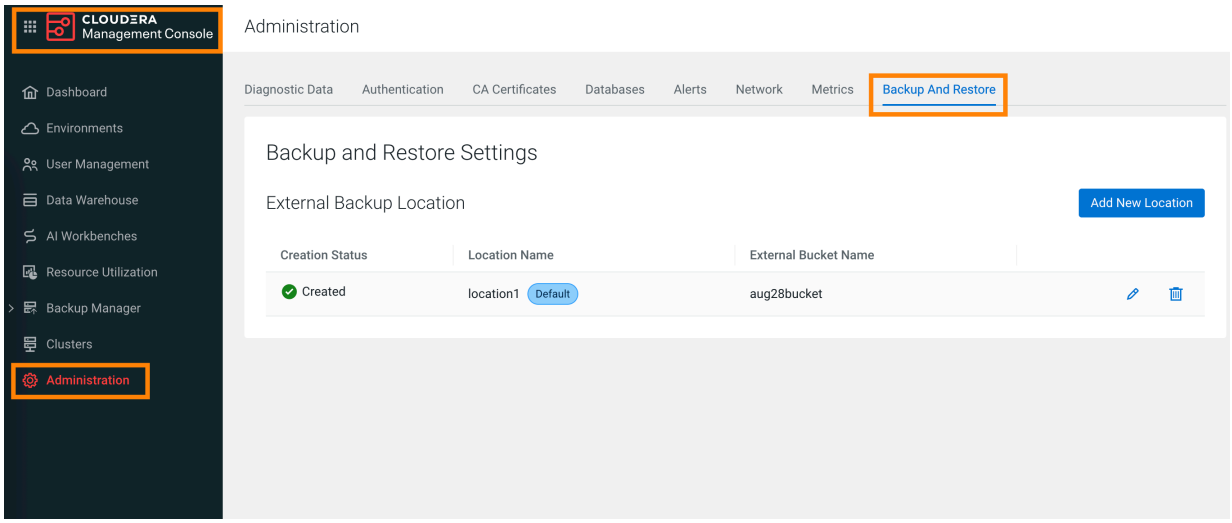


Adding external backup location

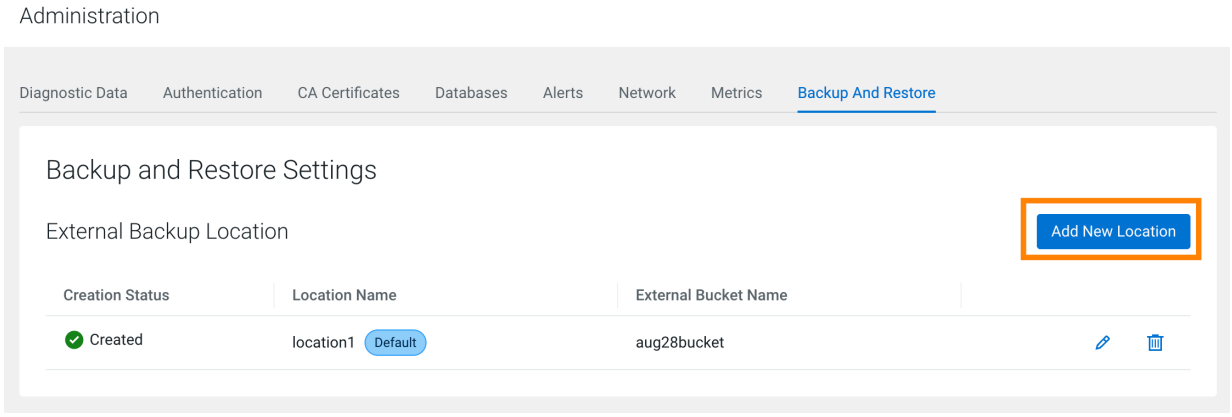
You can add one or more external backup locations to backup and restore Kubernetes namespaces and resources on Cloudera Embedded Container Service (ECS) and OpenShift Container Platform (OCP) for the Control Plane. After you add a location, you can edit only the "Access Key" and "Secret Key" and set the location as default.

Procedure

1. Go to the Cloudera Data Services on premises Cloudera Management Console Administration Backup and Restore tab.



2. Click Add New Location to add a new external location.



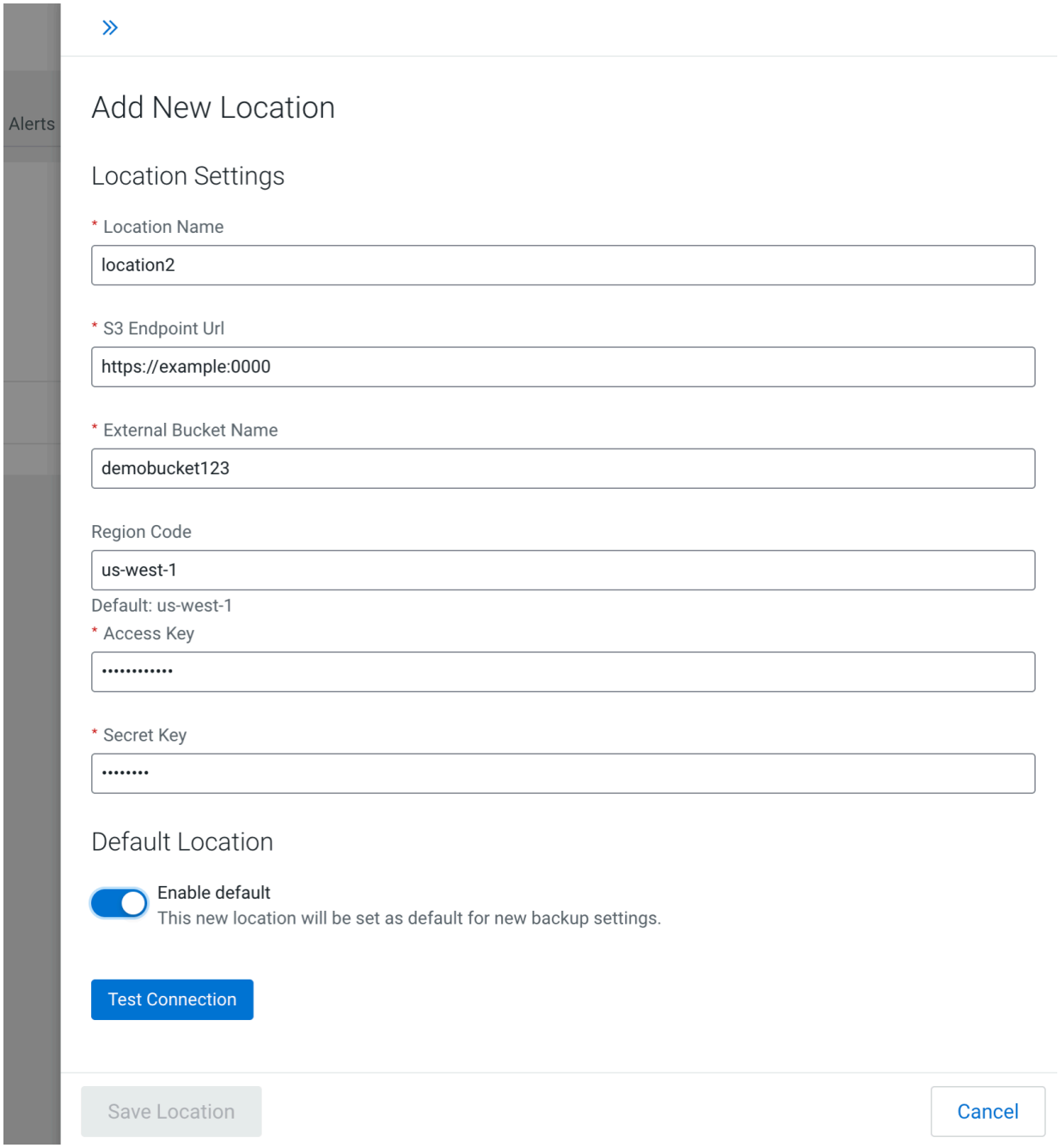
This action opens the **Add New Location** side panel.

3. Enter the following details in the **Add New Location** side panel:
   a) The external location details include:

Option	Description
Location Name	Enter a location name. The name must start and end with an alphanumeric character, and can use lowercase alphanumeric characters, dashes, and periods. The location name is translated to the custom resource in the background and is used as the custom resource name by Backup Manager.
S3 Endpoint URL	Enter the S3 endpoint URL.
External Bucket Name	Enter the bucket name.
Region Code	Enter the region code of the bucket.
Access Key	Enter the access key.
Secret Key	Enter the secret key.
Enable default	Toggle to set the external location as the default location.

   b) Click Test Connection to ensure that the details you entered are correct, and the S3 bucket is accessible.

c) Click Save Location.



Results

The location details appear on the **Backup and Restore** tab.

What to do next

You can use this location in Backup Manager to back up and restore Control Plane.

Create, restore, and manage backups of Cloudera Control Plane

The Backup and Restore Manager in the Cloudera Data Services on premises Management Console helps you to backup and restore Kubernetes namespaces and resources on Cloudera Embedded Container Service (ECS) and OpenShift Container Platform (OCP). You can view logs to troubleshoot an issue, and delete the backups if necessary. Alternatively, you can use CDP CLI to accomplish these tasks.

The following sections show how to create a backup of the Kubernetes namespaces and resources in the Cloudera Control Plane, restore a backup, delete a backup, view logs for an event, and how to use CDP CLI commands to create and restore a backup.

Creating a backup of Cloudera Control Plane

You can back up the Kubernetes namespaces and resources on Cloudera Embedded Container Service (ECS) and OpenShift Container Platform (OCP) for the Control Plane on the Cloudera Data Services on premises > Cloudera Management Console >Backup Manager > Control Plane page. You can choose to back up the resources within the cluster or to an external location.

Before you begin

Ensure that the following prerequisites are complete:

- You must have the *PowerUser* role.
- For OCP, ensure that a VolumeSnapshotClass is installed with a CSI driver that matches the CSI driver for the storage class used.
- Before you use an external location for backups, you must ensure that you have enough storage on your local system. This is because a local copy of persistent volume claim (PVC) is created temporarily which is later deleted after the backup event is complete. For example, if you are backing up 1 TB data, you require a minimum of 2 TB or more storage on your local system, where 1 TB is the original PVC and the copy requires another 1 TB.

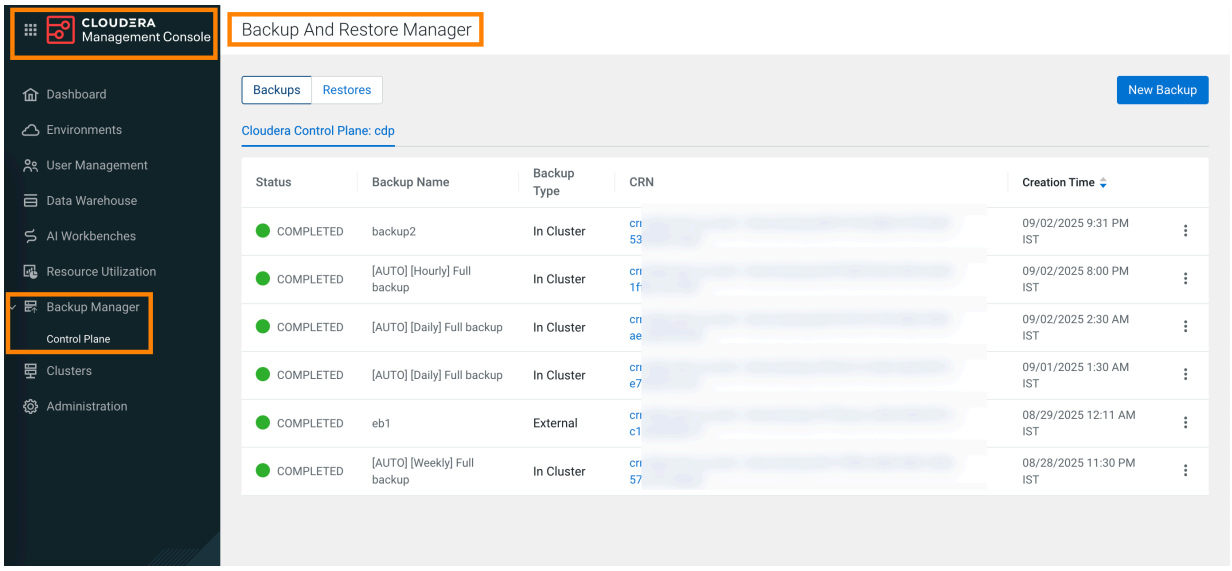
About this task

When you create a backup of the Cloudera Control Plane, Data Recovery Service (DRS) initiates the backup event or job for the chosen backup entity, assigns an ID called backupCrn to the backup event, and creates a backup of the persistent volume claim (PVC) snapshots of the Cloudera Control Plane namespaces and the backup event's PVC. CRN or Customer Resource Number is the Cloudera-specific identifier provided for an event or job.

When you choose external backup, additional external backup events or jobs for the PVCs in the Cloudera Control namespaces are initiated to backup the data externally using Kopia.

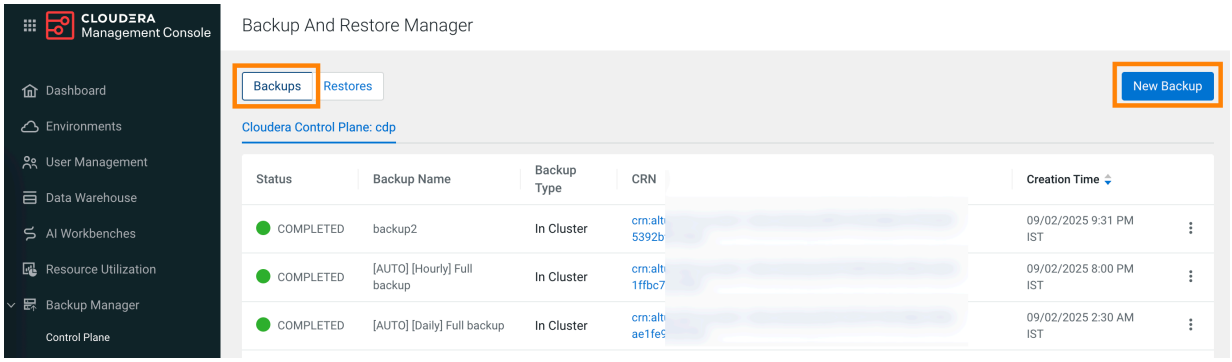
Procedure

1. Go to the Cloudera Data Services on premises Cloudera Management Console Backup Manager Control Plane page.



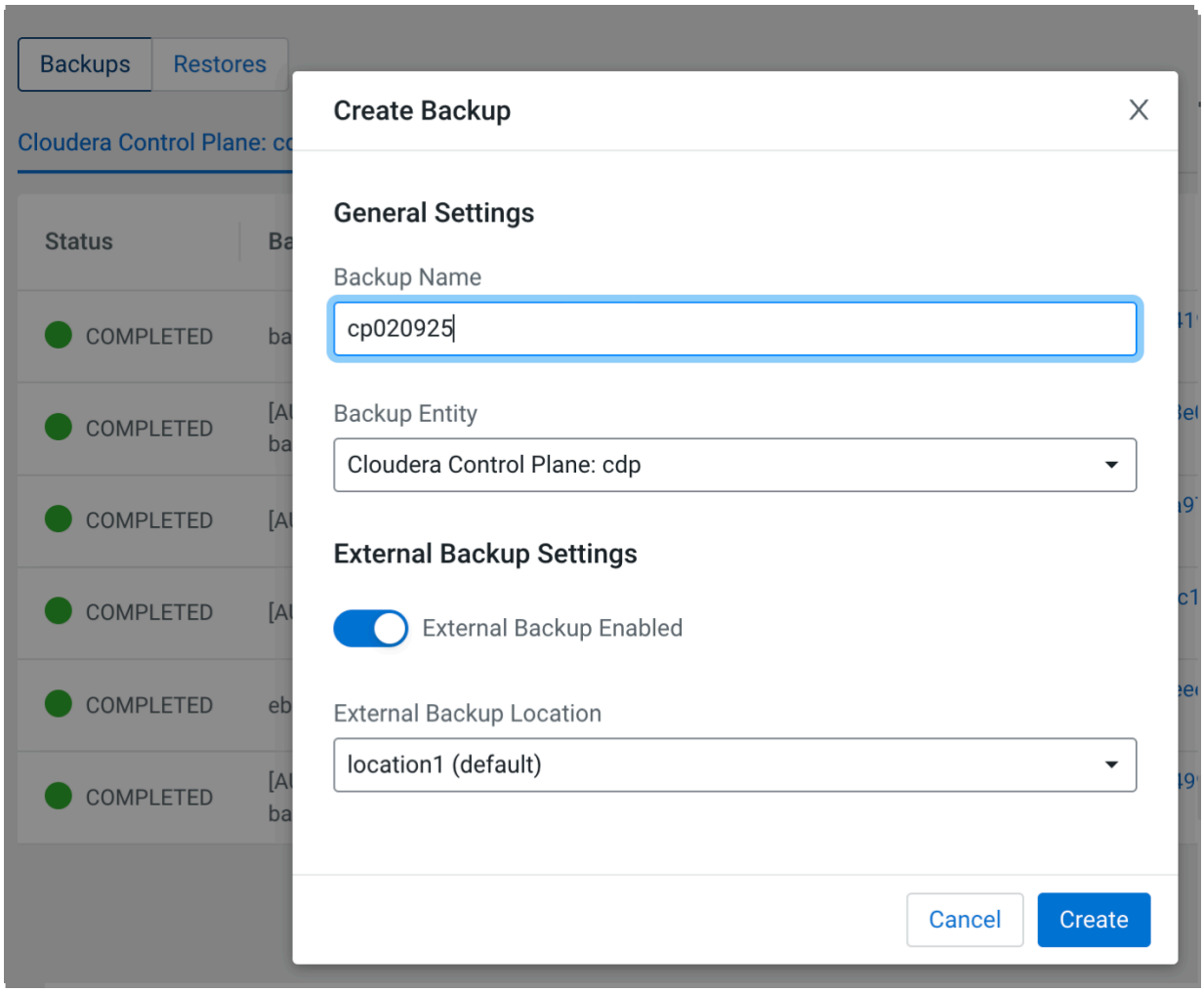
w

2. Click New Backup on the **Backups** tab.



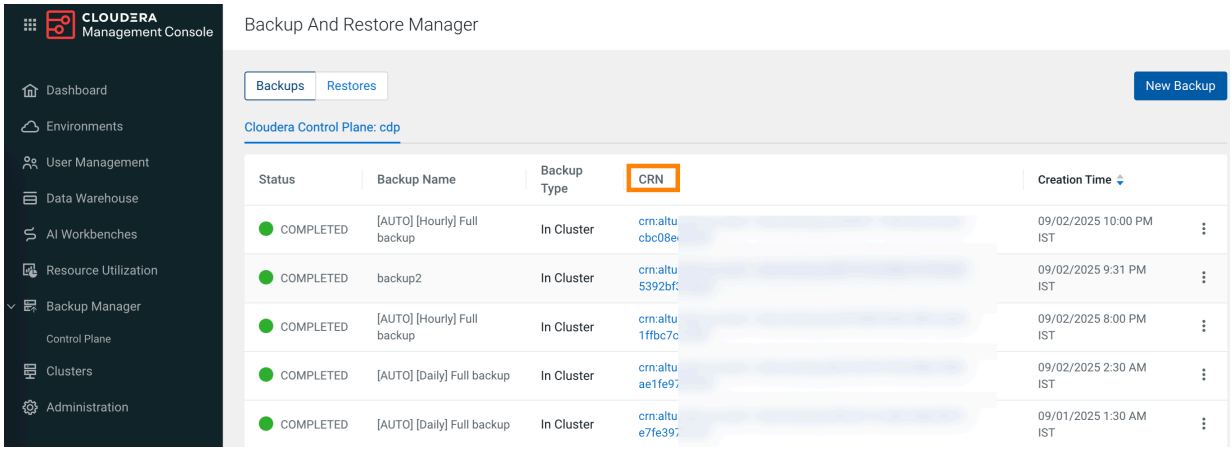
The **Create Backup** modal window appears.

3. Enter the following details on the **Create Backup** modal window:
   a) Enter a unique Backup Name.
   b) Choose the Backup Entity that you want to back up.
   c) Back up the resources to an external location by performing one of the following actions:
      1. Toggle External Backup Enabled.
      2. Choose another external location if you do not want to use the default external location.
   d) Click Create.

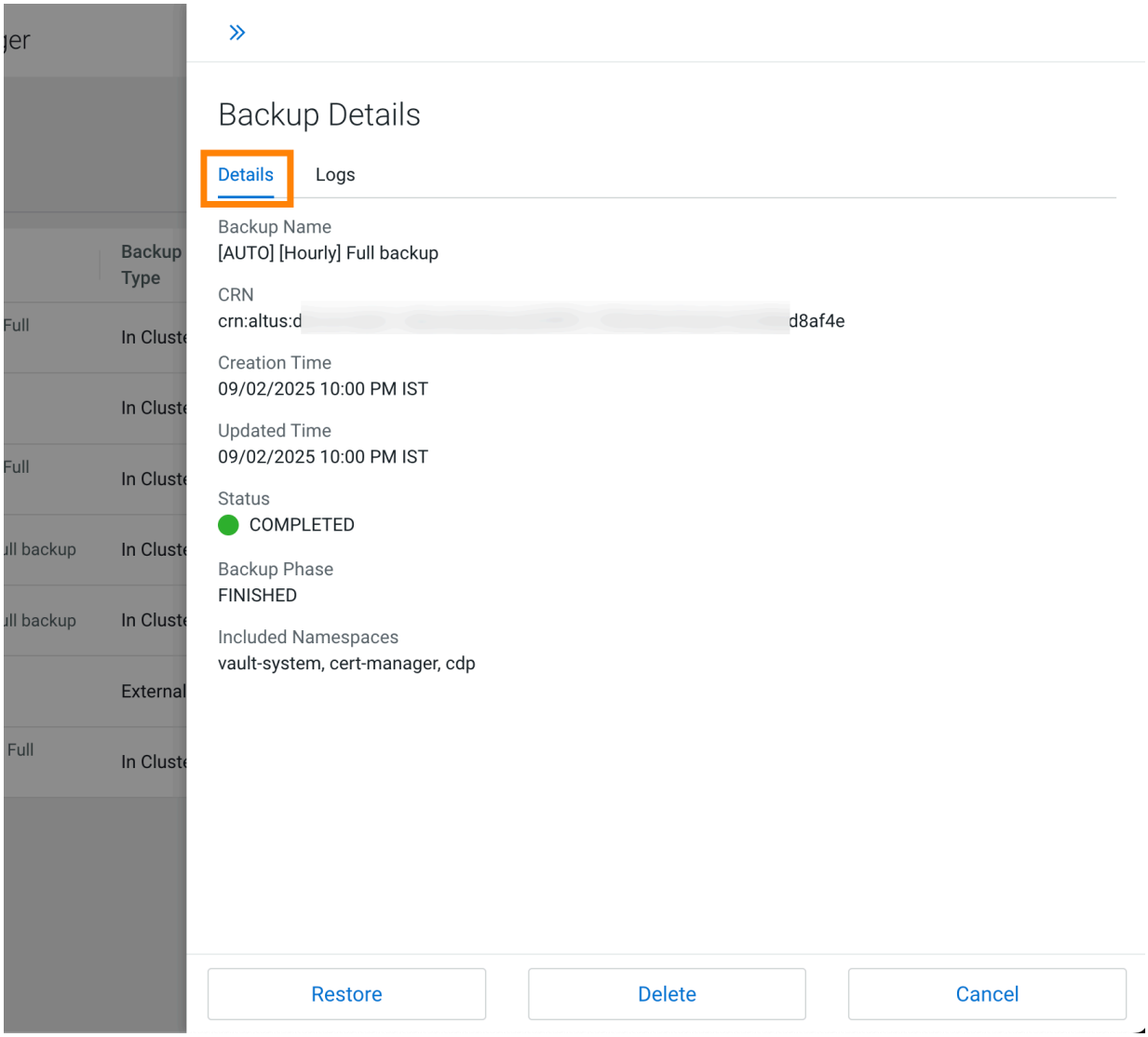


4. DRS initiates the backup event and generates a backupCRN which is an automatically assigned ID for the backup event.

   The backupCRN appears in the **CRN** column on the **Backups** tab that you can click to view the backup event details.



5. Click the CRN to view the current status of the backup event on the Backup Details modal window.



For more information about the **Backup Details** tab, see Backups tab on page 10.

Restoring a backup of Cloudera Control Plane

The Backup Manager in the Cloudera Data Services on premises Management Console helps you to restore Kubernetes namespaces and resources on Cloudera Embedded Container Service (ECS) and OpenShift Container Platform (OCP). When you start the restore event for a backup, Data Recovery Service (DRS) initiates the restore event based on the chosen backup, assigns an ID called restoreCrn to the restore event, deletes the existing resources and data, and restores the resources and data from the backup.

Before you begin

Ensure that the following prerequisites are complete:

- You must have the *PowerUser* role.
- For OCP, ensure that a VolumeSnapshotClass is installed with a CSI driver that matches the CSI driver for the storage class used.

Important: The restore event has a downtime impact because the PODs and data are recreated. During the restore event, the ECS restore vault is sealed and the POD is down which might appear as a failure in the Cloudera Control Plane environment. After the restore event is complete, the vault and POD are auto-recovered and restored. Depending on the number of resources and data, this step might take a maximum of 10 minutes to complete. If the environment does not come up, see the logs to troubleshoot. You can also contact your Cloudera account team.

Caution: Multiple restores consume disk space. Therefore, Cloudera recommends that you identify and remove the (restore) replicas, that are not required, manually on a regular basis.

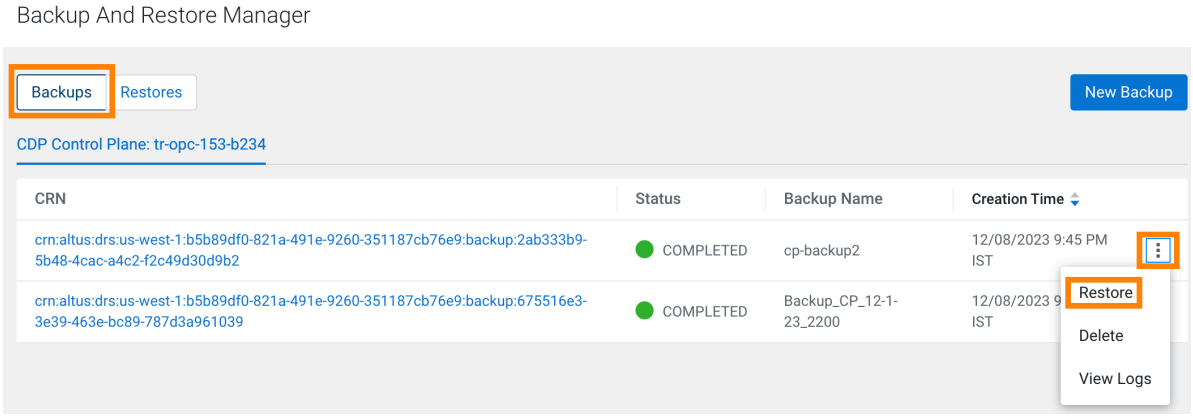
Procedure

1. Go to the Cloudera Data Services on premises Cloudera Management Console Backup Manager Control Plane page.
   The **Backup and Restore Manager** page appears.

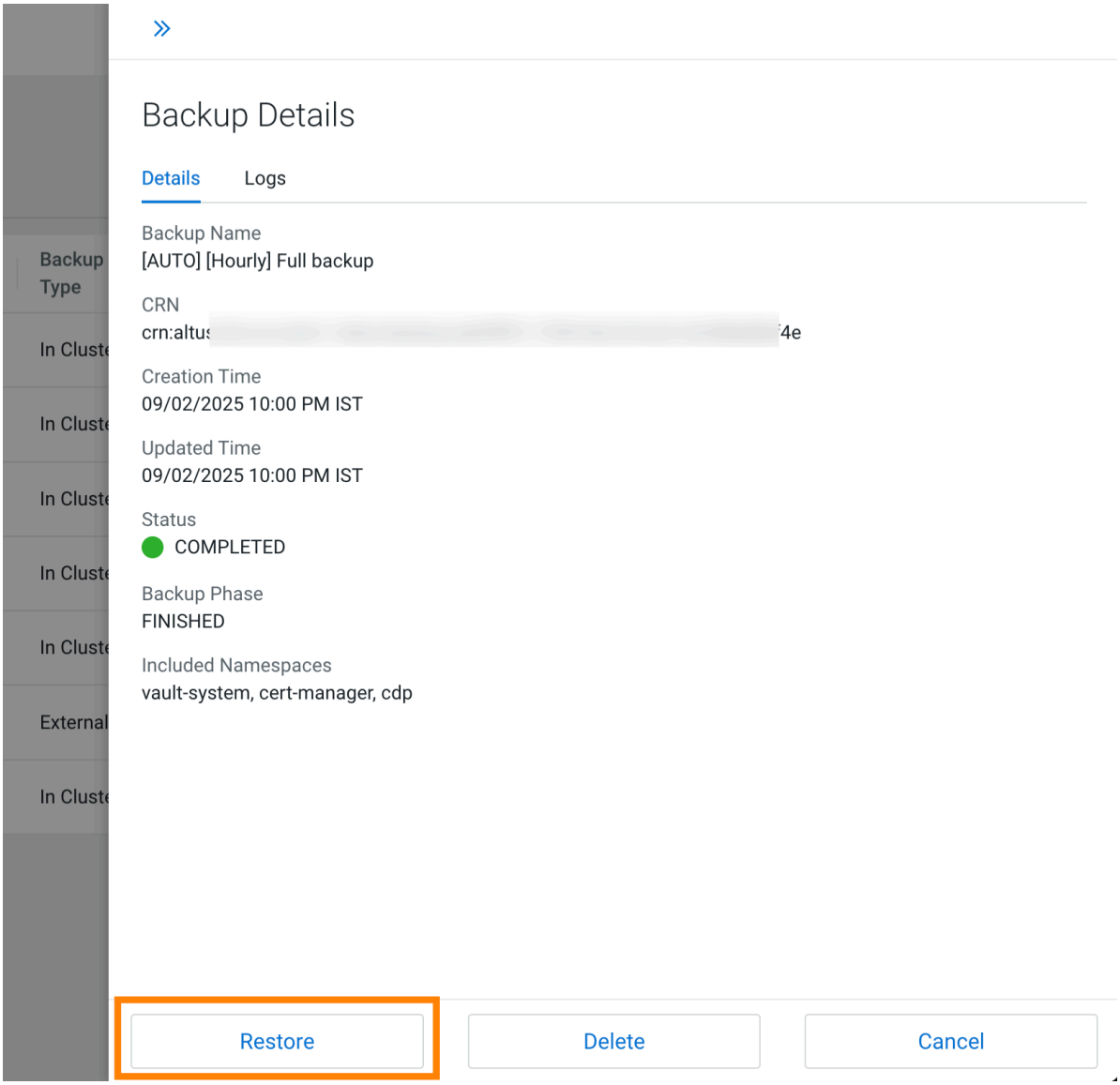
2. **Important:** Do not delete the *[***CLOUDERA INSTALLATION NAMESPACE***]*-drs namespace while the restore event is in progress. For example, if the Cloudera Data Services on premises installation is located in the cdp namespace, the drs namespace is automatically named cdp-drs.

   Choose one of the following methods to restore a backup:

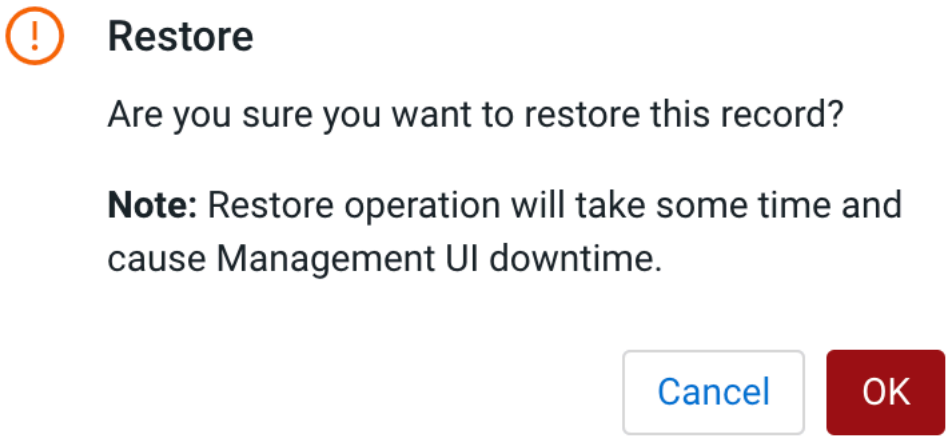
   - Click Actions Restore on the **Backups** tab.

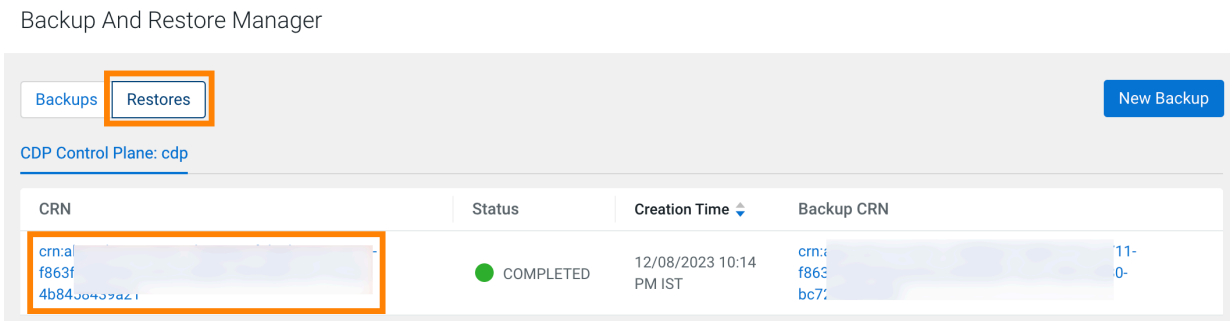
   - Click the CRN of the required backup on the **Backups** tab, click Restore on the Backup Details modal window.



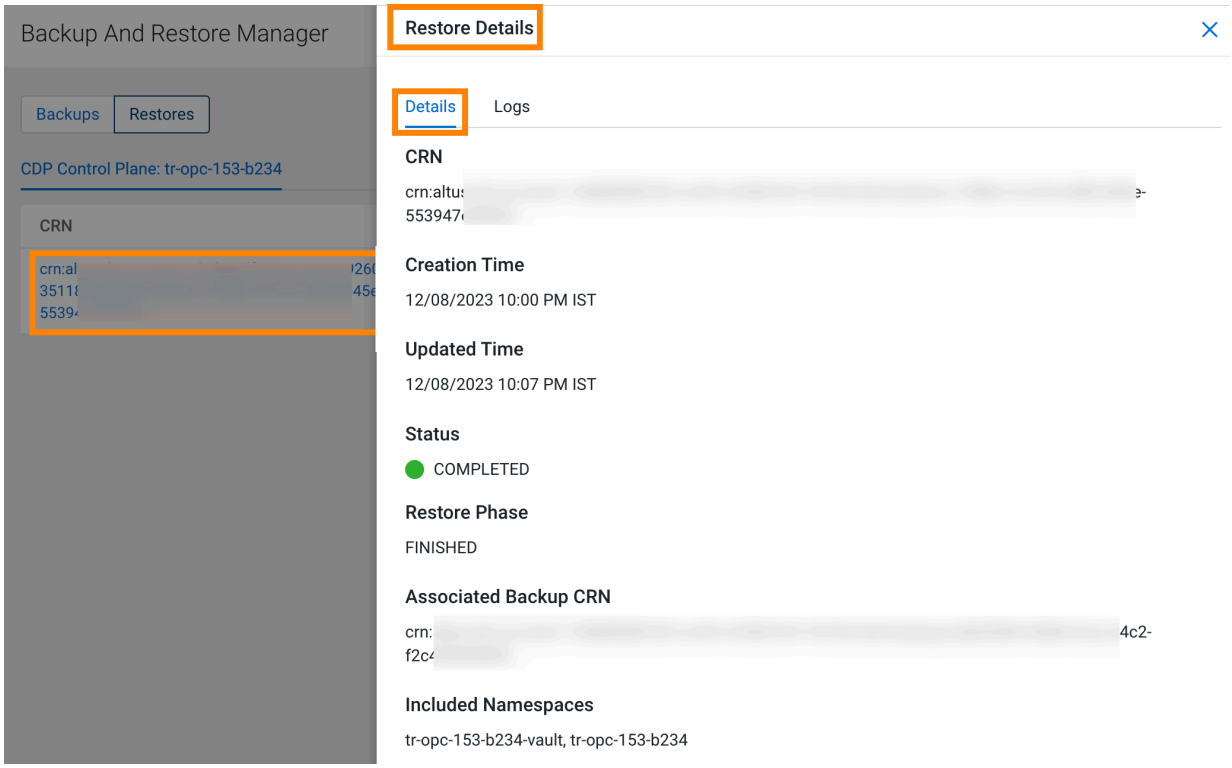
3. Click OK in the **Restore** modal window to acknowledge that you want to restore the backup.



4. Go to the **Restores** tab to view the CRN for the restore event and other details about the restore event.



5. Click the CRN to view the current restore event status on the **Restore Details** modal window.



For more information about the **Restore Details** tab, see Restores tab on page 12.

Note: After a restore event is complete, ensure that you manually delete the Persistent Volume (PV), as auto-deletion is not supported.

Deleting a backup of Cloudera Control Plane

The Backup Manager in the Cloudera Data Services on premises Management Console helps you to backup and restore Kubernetes namespaces and resources on Embedded Container Service (ECS) and OpenShift Container Platform Platform (OCP). You can delete the backups, if necessary.

Before you begin

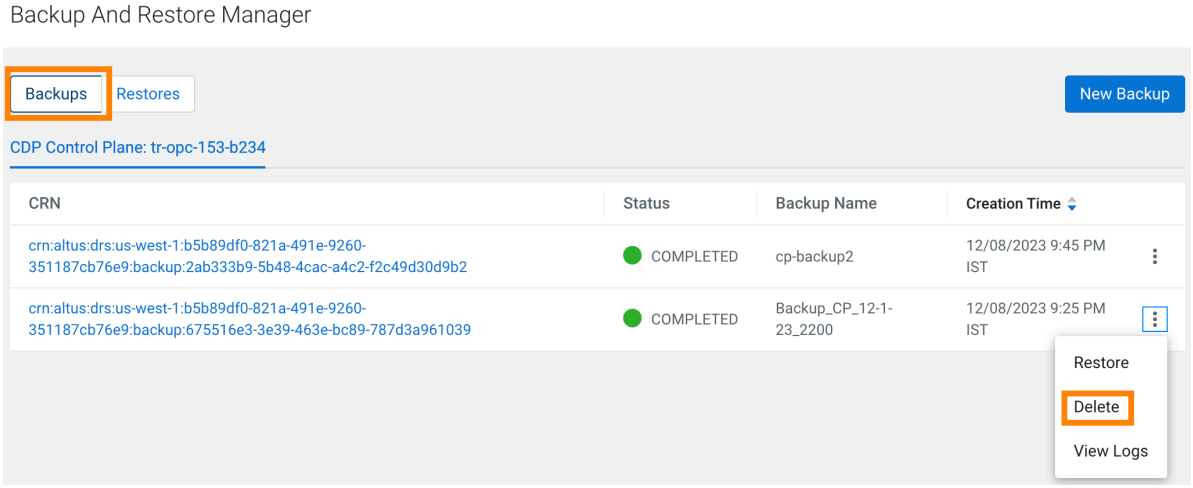
You must have the *PowerUser* role.

Procedure

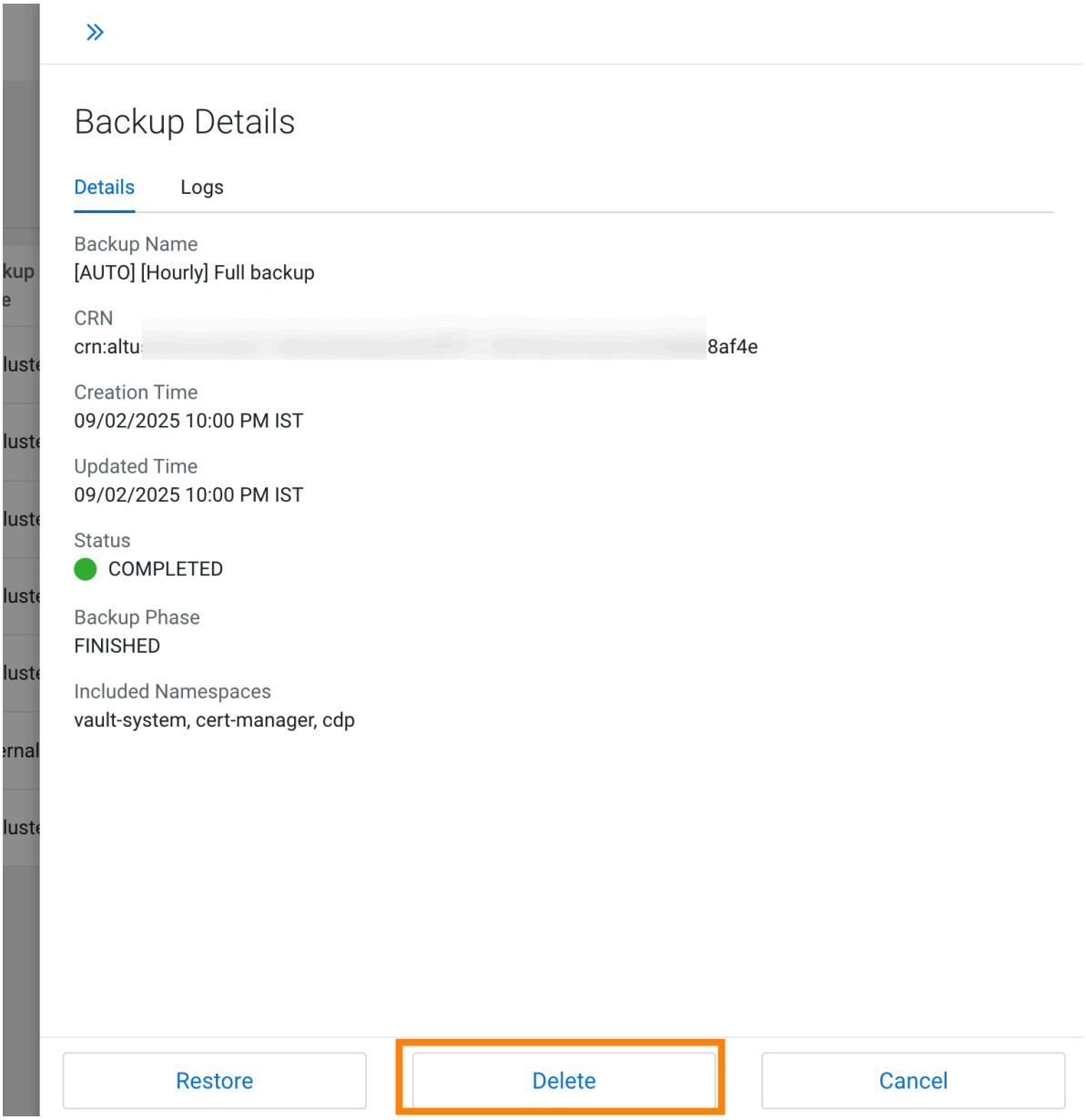
1. Go to the Cloudera Data Services on premises Cloudera Management Console Backup Manager Control Plane page.
   The **Backup and Restore Manager** page appears.

2. Choose one of the following methods to delete a backup permanently:

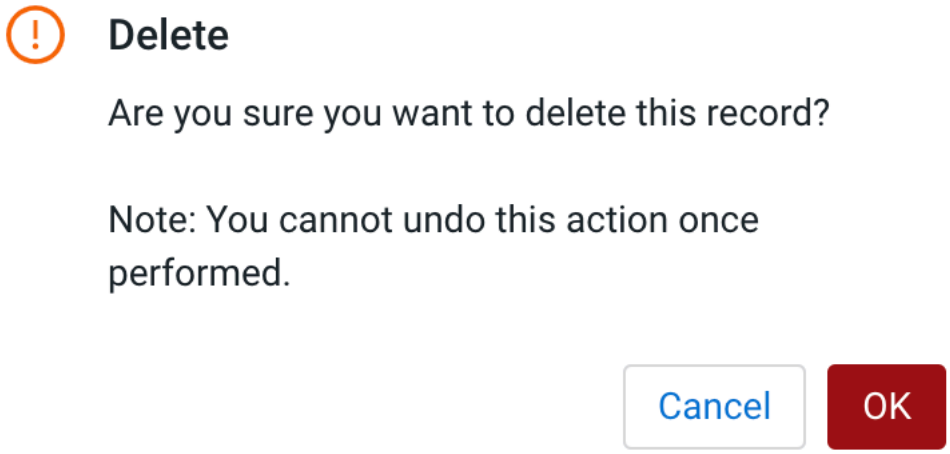
   - Click Actions Delete on the Backups tab for the required backup.

   - Click the CRN of the required backup on the **Backups** tab, and then click Delete on the Backup Details modal window.


Backup Details
Details
Logs
Backup Name
[AUTO] [Hourly] Full backup
CRN
crn:altu
8af4e
Creation Time
09/02/2025 10:00 PM IST
Updated Time
09/02/2025 10:00 PM IST
Status
COMPLETED
Backup Phase
FINISHED
Included Namespaces
vault-system, cert-manager, cdp
Restore
Delete
Cancel

3. Click OK on the Delete modal window to acknowledge that you want to delete the backup.



Viewing logs of a backup of Cloudera Control Plane

The Backup Manager in the Cloudera Data Services on premises Management Console helps you to backup and restore Kubernetes namespaces and resources on Embedded Container Service (ECS) and OpenShift Container Platform (OCP). You can view the logs of backup events and restore events to troubleshoot issues.

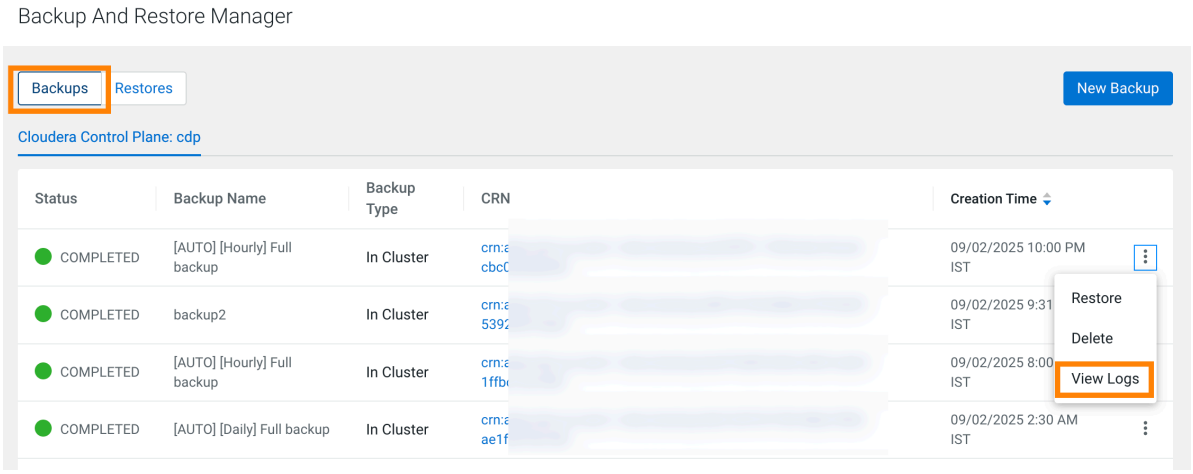
Before you begin

You must have the *PowerUser* role.

Procedure

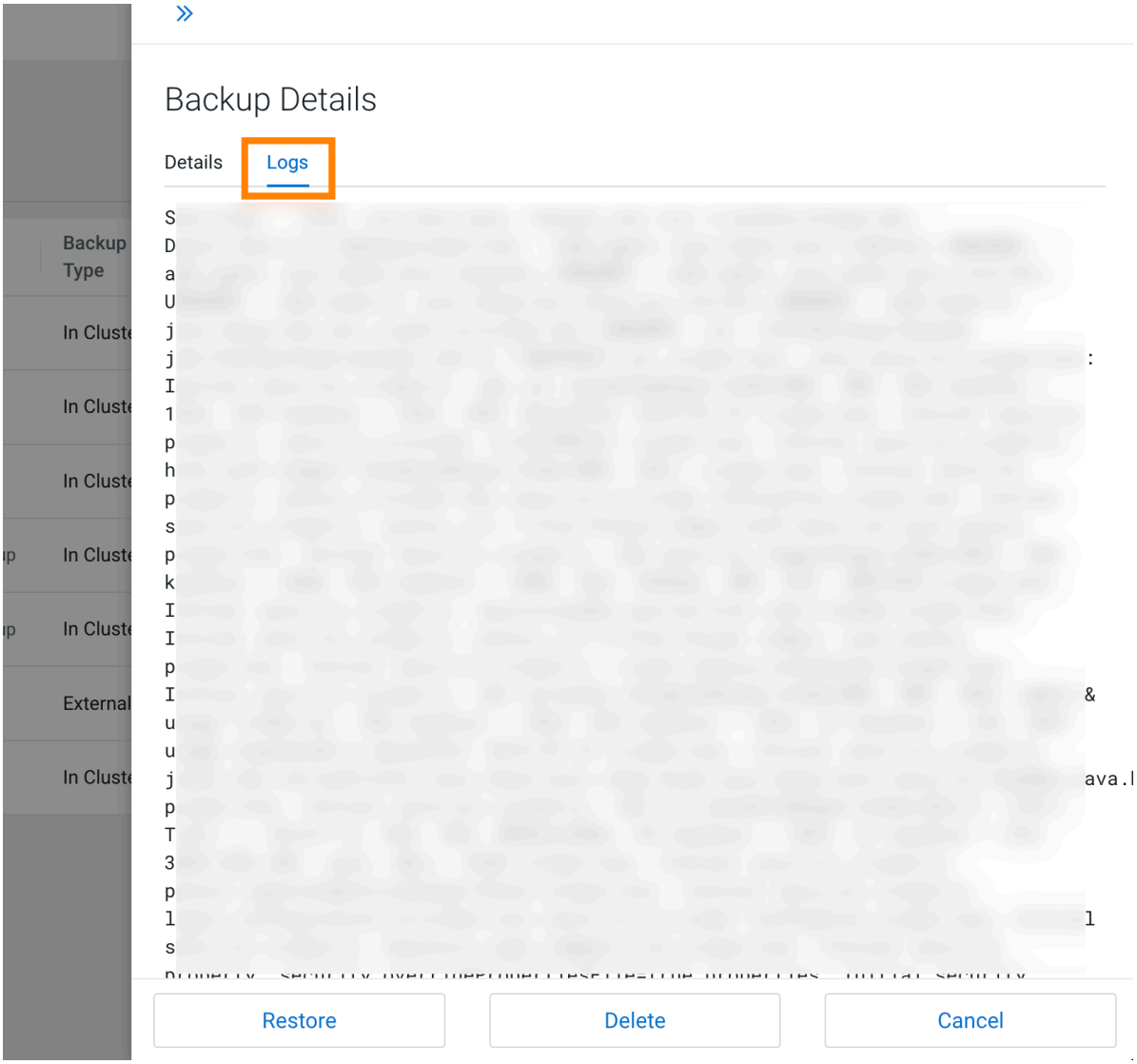
1. Go to the Cloudera Data Services on premises Cloudera Management Console Backup Manager Control Plane page.
   The **Backup and Restore Manager** page appears.

2. To view the backup event logs for a specific backup, perform the following steps:
   a) Click Actions Logs for the required backup on the **Backups** tab.



   You can also click the CRN for the required backup on the **Backups** tab, or click the Backup CRN on the **Restores** tab to view the backup event logs.

   The **Backup Details** modal window appears.
   b) Click the Logs tab on the **Backup Details** modal window.



3. To view the restore event logs for a specific restore event, perform the following steps:
   a) Click the CRN for the required restore event on the **Restores** tab.
   b) Click the Logs tab on the **Restore Details** modal window.

Using CDP CLI to back up Cloudera Control Plane and restoring it

You can use CDP CLI commands to back up and restore Cloudera Control Plane.

Before you begin

Ensure that the following prerequisites are complete:

- You must have the *PowerUser* role.
- For OCP, ensure that a VolumeSnapshotClass is installed with a CSI driver that matches the CSI driver for the storage class used.

To set up a Cloudera client to run the CDP CLI commands, see Cloudera Private Cloud CLI.

About this task

The following sample CDP CLI options show how to create a backup, restore or delete it, and monitor the progress of the events:

Procedure

1. Login to the CDP CLI setup.
2. Create a backup of Cloudera Control Plane using the create-backup CDP CLI option.

   The following sample snippet creates a backup named *Backup 2*.

   ```
   cdp.sh --form-factor private --endpoint-url https://cle-cp1.app.srd-os-0
   1.kcloud.cloudera.com drscp create-backup --backup-name "Backup 2"
   ```

3. Track the progress of the current status of the specified backupCrn (backup event) using the describe-backup CDP CLI option.

   The following sample snippet output shows the current status of the *crn:altus:drs:us-west-1:18be-4c75-8c7f-f32e697dba4a:backup:91193c4f-45f0-949c-13e232f14c9e* backupCrn.

   ```
   cdp.sh --no-verify-tls --endpoint-url https://cle-cdp.apps.drs3121-1.vpc
   .cloudera.com --no-verify-tls --form-factor private drscp describe-backup
    --backup-crn crn:altus:drs:us-west-1:18be-4c75-8c7f-f32e697dba4a:backup
   :91193c4f-45f0-949c-13e232f14c9e
   ```

4. List all the backups using the list-backup CDP CLI option.

   The following sample snippet output lists all the available backups.

   ```
   cdp.sh --no-verify-tls --endpoint-url https://cle-cdp.apps.lh-lp1-1.vpc.
   cloudera.com --no-verify-tls --form-factor private drscp list-backup
   ```

5. Restore a specific backup, using its CRN, with the restore-backup CDP CLI option.

   The following sample snippet restores the backup of *crn:altus:drs:us-west-1:88d84e3c-4c3e-9903-6c388a689690:backup:aebe-96d7-b79d10b64183* CRN.

   ```
   cdp.sh --form-factor private --no-verify-tls --endpoint-url https://cle-
   ocp1123.apps.srd-os-01.kcloud.cloudera.com drscp restore-backup --backup-
   crn crn:altus:drs:us-west-1:88d84e3c-4c3e-9903-6c388a689690:backup:aebe-96
   d7-b79d10b64183
   ```

6. Track the current status of the specified restoreCrn (restore event) using the describe-restore CDP CLI option.

   The following sample snippet output shows the current status of the *crn:altus:drs:us-west-1:a70c917a-4be8-927c-d36f3f7db2de:restore:c3b34532-4391-b62d-3f471fae5a40* restoreCrn:

   ```
   cdp.sh --form-factor private --no-verify-tls --endpoint-url https://cle-
   cp1.apps.srd-os-01.kcloud.cloudera.com drscp describe-restore --restore-c
   rn crn:altus:drs:us-west-1:a70c917a-4be8-927c-d36f3f7db2de:restore:c3b34
   532-4391-b62d-3f471fae5a40
   ```

What to do next

Explore all the available CDP CLI options to backup and restore Cloudera Control Plane and CDW in CDP CLI options for Cloudera Control Plane namespaces and CDP CLI options for Cloudera Data Warehouse (CDW) respectively.

CLI reference for using DRS on Cloudera Control Plane

You can use the Data Recovery Service (DRS) CDP CLI commands to backup and restore resources and data in the Cloudera Control Plane of Cloudera Data Services on premises.

The following table provides the CDP CLI options to backup and restore the Control Plane:

CDP CLI options	Description
create-backup	Creates a backup and archives it as a ZIP file on the same cluster. The item-name is optional for Cloudera Control Plane data recovery service. You can provide a unique backup name, so that you can identify the backup easily during restore. When you run this command, the service initiates the backup process and returns the assigned ID or `backupCrn` for the backup.
delete-backup	Deletes the specified backup (`backupCrn`) permanently. **Important:** Alerts are not generated when you run this command.
describe-backup	Shows the progress of the current status of the specified `backupCrn` (backup event).
describe-restore	Shows the progress of the current status of the specified `restoreCrn` (restore event).
get-logs	Returns logs about the specified backup, restore, or delete job and automatically creates a diagnostic bundle. You can download the bundle to your machine to analyze an issue or share it with Cloudera Support for further troubleshooting.
list-backup-entities	Lists the entities that you can backup, which includes the Control Plane namespace and its corresponding vault namespace (if embedded).
list-backups	Lists the successful backup jobs of `backupCrn`. You can filter the backup jobs using the NOT_STARTED, IN_PROGRESS, COMPLETED, PARTIALLY_FAILED, and FAILED job states.
list-restores	Lists the past restore events.
restore-backup	Restores the backup of the specified `backupCrn`. During the restore event, the existing Kubernetes resources and data are deleted and then recreated using the information in the backup. When you run the command, the service initiates the restore event and returns a `restoreCrn` value.

For more information about the DRS CDP CLI options for Cloudera Control Plane, see CDP CLI drscp.

Related Information

CDP CLI drscp

Using DRS with CDW

You can back up and restore Kubernetes namespaces behind Cloudera Data Warehouse (CDW) entities (for example, Database Catalogs, Virtual Warehouses) on demand using the Data Recovery Service (DRS). CDW leverages DRS and provides CDP CLI endpoints which you can use to create and restore backups for CDW namespaces to back up CDW metadata and configurations such as Kubernetes objects, persistent volumes, autoscaling configuration, and so on.

In 1.5.5 SP1 and higher versions, DRS automatically backs up Virtual Warehouse configurations from the database along with the namespace backup. You can restore the backup, when required.

The following limitations apply for CDW data service if you are on Embedded Container Service (ECS) or using an embedded database on Red Hat OpenShift Container Platform:

- The embedded database that CDW uses is part of the Cloudera Control Plane. You cannot back up only CDW-related entities from the embedded database using the `dw create-backup` command. You must take a backup of the Cloudera Control Plane service.
- You must restore the entire Cloudera Control Plane configurations to restore configurations stored in the CDW database. This recreates the Cloudera Control Plane namespace.

List of data recovery sub-commands for CDW

The following table lists the commands and CLI endpoints for backing up and restoring Kubernetes namespaces behind CDW entities:

DRS sub-commands for CDW	Description
create-backup	Creates an on-demand backup for the Data Warehouse including Kubernetes objects, persistent volumes, and so on. Backup requests are processed asynchronously and instantaneously.
delete-backup	Deletes an existing Data Warehouse backup. The call returns immediately. It returns a delete CRN, which is the deletion process identifier.
describe-backup	Returns the description of an existing Data Warehouse backup.
restore-backup	Restores the state of the Data Warehouse from an existing backup. It returns a restore CRN, which is the identifier of the restoration process.
describe-restore	Returns the description of the Data Warehouse restore operation.
list-backup-entities	Lists potential backup entities associated with the Data Warehouse.
list-backups	Lists backups associated with the Data Warehouse.
list-restores	Lists restores associated with the Data Warehouse.
get-logs	Returns the job logs corresponding to the specified CRN.

Related Information

CDP CLI reference of DRS commands for CDW

Troubleshooting DRS

The troubleshooting scenarios in this topic help you to troubleshoot issues that might appear for DRS in the Cloudera Control Plane. The “Backup and Restore Manager” in Cloudera Data Services on premises Management Console leverages the Data Recovery Service (DRS) capabilities to backup and restore Kubernetes namespaces and resources.

Cloudera Control Plane UI or the Backup and Restore Manager becomes inaccessible after a failed restore event?

Problem

Cloudera Control Plane UI does not come up or the Backup and Restore Manager (or drscp options) becomes inaccessible after a failed restore event.

Cause

Sometimes, some configurations take more time to restore. For example, in a shared cluster (OCP) that is heavily loaded, the restore event might surpass the set timeout limit. In this scenario, you can either wait or rerun the restore event again.

Tip: Run the restore event for such scenarios during non-peak hours.

Solution

You can perform one of the following steps after a failed restore event:

- Wait for a minimum of 15 minutes. This might resolve the issue automatically if the issue was caused due to timeout. You can verify this in the logs.
- Run restore again. This might resolve the issue if it was temporary such as, restore event during cluster maintenance.

If the Cloudera Control Plane is not restored successfully even after you follow the steps, contact Cloudera Support for further assistance.

Timeout error appears in Backup and Restore Manager

Problem

A timeout error appears in the Backup and Restore Manager or in CDP CLI (drscp) setup during a restore event.

Solution

When the restore event crosses the time set in the POD_CREATION_TIMEOUT environment property of the cdp-release-thunderhead-drsprovider deployment in the *[***CLOUDERA INSTALLATION NAMESPACE***]*-drs namespace, a timeout error appears. By default, the property is set to 900 seconds. In this scenario, you must manually verify whether the pods are up or not.

Timeout error during backup of OCP clusters

Problem

“The execution of the sync command has timed out" error appears during a backup event for OCP clusters.

Cause

This scenario is observed when the cluster is heavily used and the backup event is initiated during peak hours.

Solution

You can restart the nodes, this causes the disk to unmount and forces the operating system to write any data in its cache to the disk. After the restart is complete, initiate another backup. If any warnings appear, scrutinize to verify whether there are any dire warnings, otherwise the generated backup is safe to use. The only drawback in this scenario is the downtime impact, that is the time taken to back up the OCP clusters is longer than usual. Therefore, it is recommended that you back up the clusters during non-peak hours.

If the sync errors continue to appear, contact your IT department to check whether there is an issue with the storage infrastructure which might be preventing the sync command from completing on time.

Stale configurations in Cloudera Manager after a restore event

Cause

This scenario appears when you take a backup of the Cloudera Data Services on premises Cloudera Control Plane, upgrade Data Services, and then perform a restore. During the upgrade process, new parcels are activated and configurations in Cloudera Manager might have changed.

Solution

It is recommended that you restart Cloudera Manager after the upgrade process is complete and then initiate the restore event.

Existing namespaces are not deleted automatically after the restore event

Problem

The existing Liftie and monitoring namespaces in the environment (env2) are not deleted automatically when you perform the following steps:

1. You take a backup of an environment (env1).
2. After the backup event is complete, you create another environment (env2).
3. You restore the previously taken backup after which the Control Plane has only env1.

   The existing Liftie and monitoring namespaces in the env2 environment are not deleted after the Control Plane backup is restored.

Solution

Ensure that you manually delete the Liftie and monitoring namespaces of the env2 environment after the env1 backup is restored.

Backup event fails during volume snapshot creation process

Problem

The backup event fails during the volume snapshot creation process due to an error similar to Failed to check and update snapshot content: failed to take snapshot of the volume pvc-9d66b458-e10d-4d9c-a.

Cause

This issue might appear when multiple parallel jobs are running to take the volume snapshots of the same volume or might be because of latency issues.

Solution

Retry the backup event after a few minutes.

Restore event for an environment backup fails with an exception

Problem

When you delete an environment after the backup event, the restore operation for the environment fails and the Not able to fetch details from Cluster:... exception appears.

Cause

During the environment creation process, the environment service creates an internal Cloudera Manager user with *Full Administrator* role. The username is stored in the Cloudera Control Plane database, and the password is stored in the vault. When you delete an environment, the internal Cloudera Manager user gets deleted. The exception appears only if the password is no longer valid or might be missing. One of the reasons why the password might go missing is that while fixing a vault corruption, the vault might have been rebuilt without fixing the Cloudera Manager credentials.

Solution

Procedure

1.  **Note:** The following commands use the cdp-embedded-db-0 environment:

   Get the internal Cloudera Manager username using the following commands to determine whether the credential is valid.

   a. Login into the environment using the kubectl exec -it cdp-embedded-db-0 -n *[***CLOUDERA CONTROL PLANE NAMESPACE***]* psql command.
   b. Connect to the environment database using the \c db-env; command.
   c. Run the following SQL query in the cdp-embedded-db-0 pod:

      SELECT e.environment_crn, c.value FROM environments e JOIN configs c ON e.environment_crn = c.environment_crn WHERE e.environment_name = '*[***YOUR ENV NAME***]*' AND c. attr = 'cmUser';

      Sample output:

```
  environment_crn                  | value
-----------------------------------------------------
crn:altus:environments:us-west-1:60eed1-46de-992-90b5-0ff943dae1c8:
environment:test-saml2-env-1/48e9fcf-9620-4c8f-bc7d-caa76b1834f5
 | __cloudera_internal_user__
test-saml2-env-1-798414fe-faa6-43e1-ac9c-75c4d33ec294
```

      The __cloudera_internal_user__ test-saml2-env-1-798414fe-faa6-43e1-ac9c-75c4d33ec294 is the internal Cloudera Manager username in the sample output.

2. Get the internal Cloudera Manager password using the following commands:

   a) Run the following commands to get the root token for the embedded vault:

      1. If you are using OCP:

         $ kubectl get secret vault-unseal-key -n [***VAULT-NAMESPACE***] -o jsonpath="{. data.init\.json}" | base64 -d {"keys":["[***VALUE***]"],"keys_base64":["[***value***]="],"reco very_keys":null,"recovery_keys_base64":null,"root_token":"[***VALUE***]"} command returns the vault root token.
      2. If you are using ECS:

         a. • [root@cm_server_db_host ~]# psql -U cm cm

- select * from CONFIGS where attr like '%vault_root%';

Sample output:

```
config_id | role_id | attr | value | service_id | host_id | conf
ig_container_id | optimistic_lock_version | role_config_group_id |
 context | external_account_id | key_id
------------+---------+------------+----------------------------
+---
1546337327 | | vault_root | hvs.SvIrIhhffYEmVPEWN3TSEzks | 15463371
54 |   | | 0 | | NONE ||
```

The hvs.SvIrIhhffYEmVPEWN3TSEzks value in the above sample output is the vault token.

b) kubectl exec -it vault-0 -n *[***VAULT_NAMESPACE***]* /bin/sh

c) export VAULT_TOKEN=*[***VAULT ROOT TOKEN***]*

d) ~ $ vault secrets list -detailed -tls-skip-verify

Sample output:

```
Path          Plugin       Accessor              Default TTL    Max TTL
   Force No Cache    Replication    Seal Wrap    External Entropy Access
    Options           Description
         UUID                                    Version    Running Vers
ion          Running SHA256    Deprecation Status
----          ------       --------              -----------    -------
    --------------    -----------    ---------    --------------------
--    -------           -----------
           ----                                    -------    ----------
-----          --------------    ------------------
cubbyhole/    cubbyhole    cubbyhole_35ff7854    n/a            n/a
   false             local          false        false
    map[]             per-token private secret storage
         f2fa15ec-49-cea2-88f6-e6807c30fba3    n/a        v1.13.1+builti
n.vault    n/a               n/a
identity/     identity     identity_b7aa2294     system         system
    false             replicated     false        false
     map[]             identity store
          17990faa-e0-727a-92a3-aaaa1ff43393    n/a        v1.13.1+built
in.vault    n/a               n/a
kv/           kv           kv_2ba3b77c           system         system
     false             replicated     false        false
           map[version:2]    key/value secret storage
                    98b14495-b6-6958-04bc-1ca7c55d4590    n/a
 v0.14.2+builtin          n/a               supported
secret/       kv           kv_218f4379           system         system
    false             replicated     false        false
     map[version:2]    key/value secret storage
          06371963-e6-56c1-7ab3-d6c438720dbf    n/a        v0.14.2+built
in          n/a               supported
sys/          system       system_46e657a4       n/a            n/a
   false             replicated     true         false
    map[]             system endpoints used for control, policy and debu
gging    8ca5d96f-a45e-155a-cfc1-25a56b6a0de5    n/a        v1.13.1+buil
tin.vault    n/a               n/a
```

In this command output, kv/ is the secret path.

e) ~ $ vault kv list -tls-skip-verify kv

Sample output:

```
Keys
----
```

```
[***CLOUDERA CONTROL PLANE NAMESPACE***]
```

f) ~ $ vault kv list -tls-skip-verify kv/*[***CLOUDERA CONTROL PLANE NAMESPACE***]*

Sample output:

```
Keys
----
data/
liftie/
test
```

g) ~ $ vault kv list -tls-skip-verify kv/<[***CLOUDERA CONTROL PLANE NAMESPACE***]/data

Sample output:

```
Keys
----
[***ENV NAME1***]
[***ENV NAME2***]
```

Identify the environment for which the exception appeared.

h) ~ $ vault kv list -tls-skip-verify kv/*[***CLOUDERA CONTROL PLANE NAMESPACE***]/[***ENTER THE ENV NAME WITH THE EXCEPTION***]*

Sample output:

```
Keys
----
[***RANDOM UUID***]
```

i) ~ $ vault kv list -tls-skip-verify kv/*[***CLOUDERA CONTROL PLANE NAMESPACE***]/[***ENTER THE ENV NAME WITH THE EXCEPTION***]/[***RANDOM UUID***]*

Sample output:

```
Keys
----
cmPassword
dockerConfigJson
kubeconfig
```

j) ~ $ vault kv get -tls-skip-verify kv/*[***CLOUDERA CONTROL PLANE NAMESPACE***]/[***ENTER THE ENV NAME WITH THE EXCEPTION***]/[***RANDOM UUID***]*/ cmPassword

Sample output:

```
================ Secret Path ======================
kv/[***CLOUDERA CONTROL PLANE NAMESPACE***]/[***ENV NAME***]/[***RANDOM
 UUID***]/cmPassword

======= Metadata =======
Key                Value
---                -----
created_time       2023-11-15T04:32:36.477837897Z
custom_metadata    <nil>
deletion_time      n/a
destroyed          false
version            1

==== Data ====
Key      Value
---      -----
```

```
value    ae4cff8a-fcee-48e9-b381-4a16e883694a88c8d2
```

The value is the cmPassword (Cloudera Manager password).

3. Log into Cloudera Manager using the username (cloudera_internal_user) and password (cmPassword) that you obtained in the previous steps.
4. Run the following commands as shown to regenerate the internal Cloudera Manager credentials in bash:
   a) [root@user ~]# uuidgen command creates the first universally unique identifier (UUID) which you use in the Cloudera Manager username.

      Sample output:

      ```
      dc7c7dd7-5a58-497a-a1d1-46cd
      ```

   b) [root@user ~]# uuidgen command creates another universally unique identifier (UUID) which is the Cloudera Manager password.

      Sample output:

      ```
      9a863dc4-be61-430f-ac87-a4eba0
      ```

5. Assemble the new Cloudera Manager username using the information from the previous commands in the "__cloudera_internal_user__" + *[***ENTER THE ENV NAME WITH THE EXCEPTION***]* + "-" + *[***FIRST_UUID***]* format.

   For example, __cloudera_internal_user__cldrienv1-dc7c7dd7-5a58-497a-a1d1-46cd. In this assembled Cloudera Manager username, the prefix __cloudera_internal_user__ is followed by a string that contains the name of the environment with the exception cldrienv1 and the generated UUID dc7c7dd7-5a58-497a-a1d1-46cd separated by "-".

   The new Cloudera Manager password is the second UUID. For example, 9a863dc4-be61-430f-ac87-a4eba0
6. Go to the Cloudera Manager Support API Explorer UsersResource POST /users REST API, and perform the following steps:
   a) Click Try it out, and substitute the Cloudera Manager username and password in the following JSON string:

      ```
      {
        "items": [
          {
            "name": "[***NEW_CM_INTERNAL_USER***]",
            "password": "[***NEW_CM_INTERNAL_USER_PASSWORD***]",
            "authRoles": [
              {
                "displayName": "Full Administrator",
                "name": "ROLE_ADMIN"
              }
            ]
          }
        ]
      }
      ```

   b) Copy the JSON string into the REQUEST BODY, and click Execute.

      You get 200 response code.
7. Verify whether you can use the username and password to log into Cloudera Manager.

8. Replace the stale Cloudera Manager user with the new username with the following commands:
   a) kubectl exec -it cdp-embedded-db-0 -n *[***CLOUDERA CONTROL PLANE NAMESPACE***]* psql
   b) \c db-env;
   c) Run the following SQL queries in the cdp-embedded-db-0 pod:
      1. SELECT e.environment_crn, c.value FROM environments e JOIN configs c ON e.environment_crn = c.environment_crn WHERE e.environment_name = '*[***YOUR ENV NAME***]*' AND c.attr = 'cmUser';
      2. UPDATE configs SET value='*[***NEW CLOUDERA MANAGER INTERNAL USER***]*' WHERE environment_crn='*[***ENVIRONMENT CRN OF ENV WITH THE EXCEPTION***]*' AND attr='cmUser'; command replaces the old Cloudera Manager username.
9. Replace the stale Cloudera Manager password with the new password:
   a) Run the steps in Step 2 to find the Cloudera Manager user password credential path in the vault which should be in kv/*[***CLOUDERA CONTROL PLANE NAMESPACE***]*/*[***ENV-NAME***]*/*[***RANDOM UUID***]*/cmPassword format.
   b) Run $ vault kv patch -tls-skip-verify kv/*[***CLOUDERA CONTROL PLANE NAMESPACE***]*/*[***ENV NAME WITH THE EXCEPTION***]*/*[***RANDOM UUID***]*/cmPassword value=*[***NEW_CM_INTERNAL_USER_PASSWORD***]*
   c) Verify whether the cmPassword is changed using the $ vault kv get -tls-skip-verify kv/*[***CLOUDERA CONTROL PLANE NAMESPACE***]*/*[***ENV NAME WITH THE EXCEPTION***]*/*[***RANDOM UUID***]*/cmPassword command.