

CDW Public Cloud Backup and Restore

Date published:

Date modified:

CLOUDBERA

Legal Notice

© Cloudera Inc. 2024. All rights reserved.

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

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

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

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

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

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

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

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

Contents

Backing up and restoring CDW.....	4
Prerequisites.....	4
Cluster URLs after reactivation.....	5
Automatically backing up the environment.....	6
Backing up the environment and objects.....	6
Monitoring Hue and Data Visualization database backup.....	8
Manually backing up the environment.....	9
Backing up AWS environment activation parameters.....	9
Backing up Azure activation parameters.....	11
Backing up observability configurations.....	12
Backing up Virtual Warehouse parameters.....	12
Backing up Hue.....	13
Backing up Data Visualization applications.....	14
Decommissioning the existing environment.....	15
Deleting the Virtual Warehouse and Data Visualization deployments.....	15
Deactivating the environment.....	15
Automatically restoring the environment.....	16
Restoring the environment and objects.....	16
Testing the restoration.....	19
Monitoring Hue and Data Visualization restoration.....	19
Manually restoring the environment.....	20
Manually reactivating the environment.....	20
Modifying configurations after activation.....	22
Disabling end user access.....	23
Recreating the Virtual Warehouses.....	24
Restoring Hue.....	26
Restoring Data Visualization.....	29
Enabling end user access.....	30
Monitoring environment restoration.....	30

Backing up and restoring CDW

The backup and restore procedures for AWS and Azure replace the in-place upgrade Cloudera offered for AWS environments. To get the supported Kubernetes version, you back up your old AWS or Azure environment and start up a new environment using the restoration process.

The backup/restore feature saves your environment parameters, making it possible to recreate your environment with the same settings, URL, and connection strings you used in your previous environment.

You can back up and restore CDW using automatic and manual procedures. The following list summarizes the following high-level procedures to upgrade:

1. Back up the configurations of the Database Catalog and Virtual Warehouse(s) in the existing environment.
2. Deactivate your CDW environment.
3. [Optional] manually activate a new environment using manually backed up configurations.
4. Restore the environment, default Database Catalog, and Virtual Warehouses.
5. Monitor the Hue and Data Visualization database restore process.

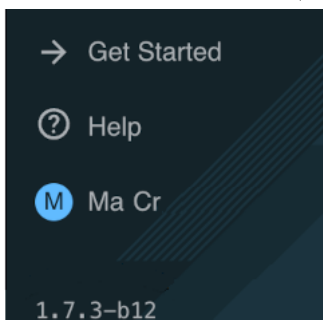
Prerequisites

Before proceeding to back up and restore CDW, you must meet a number of prerequisites.

The following prerequisites are mandatory for a successful backup and restore of CDW.

- You have not enabled the MULTI_DEFAULT_DBC entitlement.
- Your Database Catalogs are not custom (non-default) ones.
- CDP CLI 0.9.99 or later is installed and configured.
- You have Cluster Administrator privileges and can access the CDW web UI.
- You must use the same Cloudera Data Warehouse version to restore files that you used to back up those files.

For example, using a backup file from 1.6.2-b197 (released Feb 13, 2023) for restoration will not work. The Cloudera Data Warehouse (CDW) application version 1.7.3-b12, for example, appears in the UI shown below:



The CDW application version is not the same as your cluster, Database Catalog, or Virtual Warehouse versions.

The following prerequisite is necessary if you have an Azure cluster and you choose to automatically activate the environment:

- Your Azure cluster runs CDW application version 1.6.3-b319 (released May 5, 2023) or later.

You cannot automatically activate an Azure cluster that runs CDW application version 1.6.2-b197 (released Feb 13, 2023) or earlier.

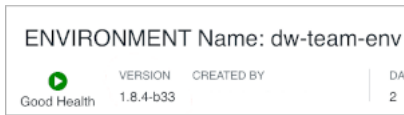
The following prerequisites are necessary if you choose to manually activate the environment.

- The AWS CLI or Azure CLI is installed and configured.
- The kubectl (or k9s equivalent) is installed and configured.

A CDW cluster is up and running with one Database Catalog and one or more Hive or Impala Virtual Warehouses.

Finding the version of your CDW environment

In Cloudera Data Warehouse, select your environment, click Edit. The Environment Details includes the version.



Importance of bringing down the cluster

Backing up and restoring CDW requires bringing down the cluster to ensure successful cluster restoration. During downtime, CDW, you must prevent end-users from accessing the cluster. If downtime is not feasible due to your operational model, you can use a workaround that [disables end user access](#) instead of bringing down the cluster.

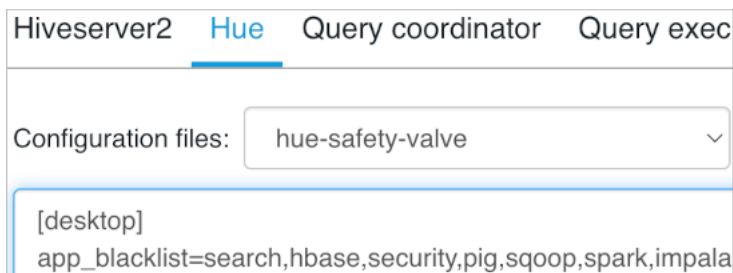
You lose any manual modification of the Kubernetes objects or configurations when you bring down the cluster. Modifications applied using the CDW UI and settings defined during creation are preserved.

Cleaning up old Hue history

Significant Hue history can accumulate in the database of long running clusters. Using the restore tool to restore such a large database can consume inordinate memory resources and result in failures. If your users work heavily with Hue, you need to clean up the old history from the database before backing up Hue as follows:

1. Navigate to one of the Virtual Warehouses and click Edit.
2. In Configurations Hue Configuration files select hue-safety-valve.
3. Change the configuration as follows:

```
[desktop]
app_blacklist=search,hbase,security,pig,sqoop,spark,impala
```



4. In the huebackend pod, start a shell session in the Hue container.

```
kubectl exec -it huebackend-0 -c hue -n <virtual warehouse id> /bin/bash
```

5. Run the cleanup tool using the following command:

```
cd /opt/hive
DESKTOP_DEBUG=True ./build/env/bin/hue desktop_document_cleanup --keep-days 60
```

The command should succeed without an error. The Hue cleanup affects only the history. No saved queries will be lost.

Cluster URLs after reactivation

It is important to understand the difference between cluster URLs before and after activation.

Current URL format

New CDW environments deploy a URL in a new format that will be preserved even after environment reactivation. The new format is:

```
<vw name>.<dw environment name>.<tenant id>.cloudera.site
```

New format example:

```
Hue-cli-update-vw-config-hive.dw-dwx-qr2j9b.xcu2-8y8x.cloudera.site
```

The new URL consists of the following components:

- Virtual Warehouse name
- CDW environment name
- Static tenant identifier

If this format is already in use, the current URLs will be preserved for both Hue and JDBC.

Old URL format

Old CDW environments use a different format. Old format example:

```
Hue-cli-update-vw-config-hive.env-qwertyu.dw.xcu2-8y8x.cloudera.site
```

The old URL consists of the following components:

- Virtual Warehouse name
- A random generated environment identifier
- A .dw separator
- Tenant ID

URL actions and recommendations

If a new environment is being activated, the old format will change to the new format. It is highly recommended to move to the new format to simplify the environment backup and restore in the future.

This recommendation is because the old format has a dynamic environment id that changes upon reactivation which changes the URL endpoints. Whereas, the new format has a static environment ID that does not change upon reactivation and maintains the URL endpoints. Additionally, Cloudera may deprecate the old format in a future release.

Should there be a need to preserve the old URL format in old CDW environments, a workaround is available and documented as an <OPTIONAL> step in the "[Reactivating the environment](#)" section below. To be able to preserve the old URL format, contact support to enable the CDW_CUSTOM_CLUSTER_ID entitlement.

Automatically backing up the environment

Backing up the environment consists of capturing environment activation parameters, observability configurations, Virtual Warehouse parameters, the Hue data, and Data Visualization applications.

Some backup procedures automate the process. Some manual backup is also required.

Backing up the environment and objects

You can export environment configurations, which you use later to automatically restore the entire CDW environment and all logical objects, such as Database Catalogs, Virtual Warehouses, and Data Visualization applications. The procedure preserves the data and configurations of the logical objects.

Before you begin

- You must temporarily deploy at least one Virtual Warehouse that runs 2023.0.14.0-15 or later to your environment as described in the steps below if you meet both of the following conditions:
 - You have not deployed Runtime version 2023.0.14.0-15 (released May 5, 2023) or later in any Virtual Warehouse in your cluster.
 - You have deployed only Runtime version 2023.0.13.0-20 (released Feb 7, 2023) or earlier in any Virtual Warehouse in your cluster.
 - Create a Virtual Warehouse that runs 2023.0.14.0-15 or later.
 - Delete the Virtual Warehouse you just created.

The steps above resolve a Hue schema incompatibility issue before backing up and restoring Hue.
- Add [bucket encryption](#) to your managed policy and attach the policy to the node instance role.
- You must use the CDP CLI version 0.9.99 or later.
- You must [clean up Hue history](#) before doing this backup if Hue is used heavily.

About this task

The procedure below backs up the environment and objects, which includes Virtual Warehouse parameters. Use the CDP CLI `dw backup-cluster` command to create the backup data.



Note: If you originally activated your cluster with the early CDW version 1.1.2 and then used in-place upgrades to upgrade the underlying kubernetes cluster, consider creating a manual backup as well as automatically backing up the environment and objects.

Procedure

Use the CDP CLI `dw backup-cluster` command to create the backup data.`

```
export CDP_PROFILE=<test / prod / etc>
export CLUSTER_ID=<the-id-of-the-cluster> # the current ID (original ID) of
the cluster

cdp \
  --profile ${CDP_PROFILE} \
  dw backup-cluster \
  --cluster-id ${CLUSTER_ID} 1>dump_${CLUSTER_ID}.json
```

Example content of the `dump_${CLUSTER_ID}.json` file:

```
{
  "clusterId": "env-lqhwqs",
  "operationId": "94197da9-fff7-4414-8b56-a30446c75119",
  "timestamp": "2023-08-16T20:22:00+00:00",
  "data": "UEsDBBQACAAIAAAAAAAAAAAAAA...
  "md5": "5f427b11f01f5540fa961aba8ea232aa"
}
```

The cluster ID is the unique CDW environment identifier. You can use the operation ID to query the backup execution details using the CLI. The data holds the object data and the configuration. The md5 is a hash for this data. In case the data and this file is lost, the cluster objects cannot be restored automatically.

What to do next

- Monitor database backup jobs

The backup will automatically start the Hue backup and Data Visualization database backup jobs [that you can monitor](#). Make sure that the database backup jobs finish before destroying the cluster. If the cluster is deleted before the jobs are finished, you cannot recover the application contents.

- Alert settings

The compaction observability alert settings are backed up. If the configuration has been modified, make a copy of the configurations, and apply them to the new cluster after restoration.

Using one of the following ways, get the value of the Alert Manager settings:

- Use the CDW UI:

Navigate to your environment tile, click Edit, and in Alert Settings, add the alert settings.

- Use kubectl:

```
kubectl get configmap -n istio-system alertmanager -o json
```

- Azure environments

Azure environments activated prior to 1.6.3-b319 (released May 5, 2023) support only manual environment backup. New activations require a managed identity for cluster creation. Old clusters do not have this setting available. Automatic recovery is not an option if your Azure was activated in 1.6.3-b319 (released May 5, 2023).

- Grafana dashboards

Any changes made to the Grafana dashboards will be lost. A new cluster will be provisioned, the data from the previous cluster won't be carried over to the new Grafana deployment.

Monitoring Hue and Data Visualization database backup

The automatic backup procedure saves the Data Visualization database contents to the configured logs or data folders based on availability.

Hue

During the manual or automatic Hue database backup operation it is critical to block any traffic to the running Hue services. If you cannot bring down the cluster, Cloudera recommends you disable end user access to the cluster endpoints. Failing to do so results in errors in addition to existing key constraints and other issues.

Automatic Hue backup

Automatic backup of Hue extracts the saved query and query history and loads them to the new cluster.

Monitoring Hue backup

The backup starts a job to load the database dump file, but does not wait for the job to complete. If you have a large database, the job can take up to an hour to complete. Ensure you allow enough time for the job to succeed.

To monitor Hue backup, log into the cluster and monitor the job status under the database catalog namespace.

```
$ kubectl get jobs -n <database catalog id>
```

The output that shows the hue-backup job looks something like this:

```
$ kubectl get jobs -n warehouse-1692037411-96hk
NAME                                COMPLETIONS  DURATION
AGE
hue-backup-ed2b8bd-1d53-4d23-a0f9-87d8ec658f74  1/1           11s
113s
hue-query-processor-db-create-job      1/1           8s
42h
```

Data Visualization

The automatic backup procedure saves the Data Visualization database contents to the configured logs or data folders based on availability.

Automatic backup

Automatic backup of Data Visualization extracts the dashboards, tables and connections. Make sure to wait for the job to finish before destroying the cluster.

Monitoring backup of Data Visualization

The backup starts a job to create the database dump file, but it does not wait for it to complete. In case your database size is large, it can take up to 20 minutes for the job to complete. Make sure to leave enough time for the job to succeed. To monitor Data Visualization backup, you can log into the cluster and see the job status under the viz namespace using the following command to extract the dashboards, tables and connections:

```
$ kubectl get jobs -n <data visualization id>
```

The output looks something like this:

```
$ kubectl get jobs -n viz-1692216942-fc2g
NAME                                COMPLETIONS   DURATION
AGE
viz-backup-d874515a-be7e-4902-ac75-269c14f9580c  1/1           3m3s
10m
viz-webapp-vizdb-create-job          1/1           57s
99m
```

Manually backing up the environment

Backing up the environment consists of capturing environment activation parameters, observability configurations, Virtual Warehouse parameters, the Hue data, and Data Visualization applications.

If you automatically backed up the environment following procedures above, do not perform procedures below for backing up the environment.

Backing up AWS environment activation parameters

You back up AWS environment activation parameters using the CDW UI , AWS CLI, and kubectl.

About this task

[AWS environment activation settings](#) that need to be available in the new environment may include IP-CIDRs for the Kubernetes cluster and load balancer, the deployment mode setting, reduced permissions mode, and overlay networks. It is required that you gather and document these settings to have the environment behave the same after the back-up/restore process as before. The activation parameter values are available in the CDW UI. Additionally, some parameters may be fetched using AWS CLI and kubectl.

The following steps gather and document the environment parameters.

Procedure

Deployment Mode (Network selection)

1. Get subnet information in one of the following ways:

- Use the CDW UI
 - Navigate to Environment Details General Details , and note the settings for **Public Subnets** or **Private Subnets**.
- Use the AWS CLI

```
aws cloudformation describe-stacks --stack-name env-q4tzxd-dwx-stack --o
utput json --query "Stacks[0].{StackName:StackName, PublicSubnetIds:Outpu
ts[?OutputKey=='PublicSubnetIds'].OutputValue, PrivateSubnetIds:Outputs[
?OutputKey=='PrivateSubnetIds'].OutputValue}"
```

IP-CIDR for kubernetes cluster and load balancer

- In the CDW UI, navigate to `Environment Details Configurations`, and note the settings of `Enable IP-CIDR` for Kubernetes cluster and `Enable IP-CIDR` for the load balancer.

Overprovision nodes

- Document the value of `Overprovision nodes`.
Stored behind the `CDW_CLUSTER_OVERPROVISIONER` entitlement.

Use Custom ECR repository

- Document the value of `Use Custom ECR repository`.
Stored behind the `CDP_CUSTOM_REPO` entitlement.

Use Overlay Network

- Using `kubectl`, get the value of `Use Overlay Network`.

```
kubectl get daemonsets -n kube-system
```



Note: If the `daemonSet aws-node` is not present, the overlay network is enabled.

Attach Managed policy ARN to Node Role

- Using the AWS CLI, fetch the `nodeInstanceRole`.

```
export nodeInstanceRole=$(aws iam list-roles --query "Roles[0].{RoleName: RoleName}" | grep "env-q4tzxd-dwx-stack-NodeInstanceRole-*" | tr -d '"' | cut -f 2 -d ':' | awk '{ $1=$1; }1')
```

- List the policies attached to this `nodeInstanceRole`.

```
aws iam list-role-policies --role-name $nodeInstanceRole
```

- Check for customized policies, which are not included in the following policies:

```
"PolicyNames": [
  "cluster-autoscaler",
  "dynamodb",
  "ebs",
  "efs",
  "kms",
  "limits-monitoring",
  "s3-list-all-buckets",
  "s3-read-only-buckets",
  "s3-read-write-own-buckets"
]
```

AMI ID

- Using one of the following ways, get the value of the AMI ID only if you need to use a custom AMI:

- Use the CDW UI:
Navigate to `Environment Details Configurations`, and note the setting of `AMI ID`.
- Use AWS CLI:

```
aws cloudformation describe-stacks --stack-name env-q4tzxd-dwx-stack --output json --query "Stacks[0].{StackName:StackName, AMI:Parameters[?ParameterKey=='EksAmi'].ParameterValue}"
```

Reduced Permissions Mode

- Using the CDW UI, determine whether or not your current policy has standard activation permission.

If the current policy does not have the standard activation permission, when you [reactivate the environment](#) you see a prompt to use the Reduced Permissions Mode.

In Reduced Permissions Mode, you deploy the cluster manually. A Cloudformation template is generated and the you must deploy the cluster and apply some role-based access control roles to the Kubernetes cluster.

Enable CloudWatch logs

- Using the CDW UI, navigate to Environment Details Configurations Enable CloudWatch Logs to get the value of Enable CloudWatch logs..

CloudWatch logs provide better visibility into cluster operations in addition to the diagnostic bundles within CDW. Enabling this option will not impact the restore procedure, even if it was not previously configured in the old environment.

Additional External Buckets

- Using the CDW UI, gather values of the Additional External Buckets parameter. Navigate to Environment Details Configurations Bucket Name .

Backing up Azure activation parameters

You configure almost all (99%) of Azure cloud resources using environment activation parameters. These parameters are available by querying Azure resource providers in the old environment. You use these parameters, which you manually document, during the activation of the new environment.

About this task

The [Azure environment activation settings](#) you want to carry over to the new environment include the compute VM size (E16ds_v4 or E16_v3), any user-assigned, managed identity, subnets, private CDW and IP CIDRS, kubenet networking, and minimum permissions.

Procedure

- Obtain a managed identity for Azure activations.

The new, required managed identity parameter provides privileges to deploy the AKS cluster. For more information about required minimal privileges, see ["Setting up minimum permissions"](#).

- Query the environment to get the Azure environment activation settings you want to carry over to the new environment.

Query the environment to get the activation parameters.

```
az aks show -n <AKS_CLUSTER_NAME> -g <CDW_RESOURCE_GROUP> --query '{Agentpools:agentPoolProfiles[0].{Name:name, Version:orchestratorVersion, State:provisioningState, AZ: availabilityZones, SKU:vmSize, VnetSubnet:vnetSubnetId, PodSubnet: podSubnetId, CDW_Timestamp:tags.timestamp, PowerState: powerState}, Api: apiServerAccessProfile, NetworkType:networkProfile.networkPlugin, DockerCIDR: networkProfile.dockerBridgeCidr, outboundType:networkProfile.outboundType,privateFQDN:privateFqdn, Identity:identity, FQDN: fqdn, AKSVersion:kubernetesVersion, Location:location, SKU: sku, OMS: addOnProfiles.omsagent}' -o jsonc
```

The query output maps to the following activation parameters:

- Compute VM Size: Agentpoolc.SKU
- Subnet: Agentpoolc.VnetSubnet
- Private CDW: api.enablePrivateCluster
- Managed identity: Identity
- Availability Zones: agentpoolss.AZ
- AKS Monitoring: oms
- K8s CIDR: api.authorizedIpRanges
- Kubenet: NetworkType

- Docker CIDR: dockerCidr
 - AKS DNS Zone: api.privateDnsZone
 - OutboundType: outboundType
3. Get the internal load balancer settings in one of the following ways.
 - Use a query.

```
az resource list -g MC_<AKS_CLUSTER_NAME>_<REGION> --query "[?type == 'Microsoft.Network/loadBalancers'].{Name: name, Type: type}" -o jsonc
```

If the output lists an internal load balancer, the environment has been activated with the **Enable internal load balancers** option.

- Use the CDW UI described in the following steps.

Using the CDW UI to get internal load balancer settings

4. Using the CDW UI, in the Data Warehouse service, expand Environments by clicking the More....
5. In Environments, click the search icon and locate the environment that you want to view.
6. Click Edit Configurations
7. Note the IP range for the load balancer.

Backing up observability configurations

Environment configurations are used to monitor and observe an environment.

About this task

If you did not make changes to the environment for Observability or the Alert Manager, skip this step.

Procedure

Alert Manager setting

1. Using one of the following ways, get the value of the Alert Manager settings:

- Use the CDW UI:
Navigate to your environment tile, click Edit, and in Alert Settings, add the alert settings.
- Use kubectl:

```
kubectl get configmap -n istio-system alertmanager -o json
```

Observability configurations

2. Using the CDW UI, click Edit, and in Observability, copy the configurations. Observability::json > ObservabilityConfig>

Backing up Virtual Warehouse parameters

You use the CDP CLI version 0.9.88 or later.

Procedure

Use the CDP CLI `dw backup-cluster` command to create the backup data.

```
export CDP_PROFILE=<test / prod / etc>
export CLUSTER_ID=<the-id-of-the-cluster> # the current ID (original ID) of
the cluster

cdp \
  --profile ${CDP_PROFILE} \
  dw backup-cluster \
```

```
--cluster-id ${CLUSTER_ID} 1>dump_${CLUSTER_ID}.json
```

Example content of the `dump_${CLUSTER_ID}.json` file:

```
{
  "clusterId": "env-qr5cj7",
  "timestamp": "2023-02-16T10:29:16+00:00",
  "data": "UESDBBQ...AAAAAAAAAAAAABkYXRhUESFBgAAAAABAAEAMgAAAKuBAQAAAA==",
  "md5": "5a18129ad01b75f315ae518a37004804"
}
```

In this file, the “clusterId” field denotes the cluster from which the backup was taken. The “data” field contains the configuration backup data itself.

Backing up Hue

Backing up Hue is an automated process that saves the Hue database content. The process places the content in configured logs or data folders based on availability. If, for any reason, you want to manually back up the database, you can choose to do so.

Before you begin

- You must temporarily deploy at least one Virtual Warehouse that runs 2023.0.14.0-15 or later to your environment as described in the steps below if you meet both of the following conditions:
 - You have not deployed Runtime version 2023.0.14.0-15 (released May 5, 2023) or later in any Virtual Warehouse in your cluster.
 - You have deployed only Runtime version 2023.0.13.0-20 (released Feb 7, 2023) or earlier in any Virtual Warehouse in your cluster.
1. Create a Virtual Warehouse that runs 2023.0.14.0-15 or later.
 2. Delete the Virtual Warehouse you just created.

The steps above resolve a Hue schema incompatibility issue before backing up and restoring Hue.

Manual backup

If anything goes wrong with the automatic backup of Hue, or if you just prefer a manual process, you can back up Hue manually. You can choose to manually save and restore the Hue data to keep the Hue saved queries and query history for your Virtual Warehouses on the new cluster.

About this task

One Hue database is shared between all Virtual Warehouses, so you execute the following steps only once.

Procedure

1. Find Hue pods and namespaces.

```
$ kubectl get pods --all-namespaces --field-selector metadata.name=huebackend-0
```

2. Use SSH to access the Hue pod on the virtual warehouse cluster.

```
$ kubectl exec -it huebackend-0 -n <virtual warehouse ID> -c hue -- /bin/bash
```

For example:

```
$ kubectl exec -it huebackend-0 -n compute-1668714083-8ms4 -c hue -- /bin/bash
```

3. Back up Hue data to the /tmp directory on the Hue pod. In the container, your current directory should be /opt/hive.

```
$ ./build/env/bin/hue dumpdata --indent 2 -o /tmp/data.json
```

4. Copy the backup from the Hue pod to the local machine.

```
$ kubectl cp <virtual warehouse ID>/huebackend-0:/tmp/data.json -c hue /tmp/data.json
```

Backing up Data Visualization applications

You can use kubectl or k9s to back up Data Visualization (DataViz) applications in an AWS environment.

About this task

To keep the charts and dashboards that you created, you must save and restore Data Visualization data.

If you did not make changes to the environment for Data Visualization, skip these steps; otherwise, perform these steps on each of your Data Visualization applications because each Data Visualization uses a separate database.

Before you begin

- Data Visualization must use the DataLake Postgres instance for storing its metadata.

Procedure

1. Find the necessary information, such as database name, host, port, user, and password.

```
$ kubectl get secrets/pg-db-secret -o=jsonpath={.data.'\'.pgpass'} -n viz-1680129861-kbtr | base64 -D
```

```
postgres-service:5432:metastore:hive:ZufGC6Dmh03N1iW042uTosZtr4XvCtJIYPQ==
```

2. Use kubectl or k9s to access the Hue container in the targeted CDW environment (need KUBECONFIG setup), and find the namespace for Hue and for DataViz.

```
$ kubectl get pods --all-namespaces --field-selector metadata.name=huebackend-0
$ kubectl get pods --all-namespaces --field-selector metadata.name=viz-webapp-0
```

If you are running multiple DataViz instances, make a note of the DataViz namespaces and their user-provided friendly names from the CDW UI. On the new cluster, the namespace names will change, so it's important to know where to load the corresponding DataViz deployment.

3. Select one of the Hue namespaces and shell into the container.

```
$ kubectl exec -it huebackend-0 -n <virtual warehouse ID> -c hue -- /bin/bash
```

4. Get the dump using the code below by providing a DataViz namespace. All DataViz databases can be backed up from this container. If you have multiple namespaces, make a note of the friendly DataViz name pairs.

```
pg_dump -U hive -h postgres-service -W -F t <DataViz namespace>_vizdb > ./viz_pg_dump.tar
```

If you have multiple DataViz instances running, backup all databases in this step. It is important to use a naming convention that will allow you to identify which database back-up contains the contents of the corresponding DataViz database. This is needed, as the namespace names will be different on the new cluster.

- Copy the dump file to your local machine.

For example:

```
kubectl cp <virtual warehouse ID>/huebackend-0:/opt/hive/viz_pg_dump.tar
~/Downloads/logs/viz_pg_dump.tar -c hue
```

Decommissioning the existing environment

You follow procedures to first delete the Virtual Warehouses in the CDW Environment and the Data Visualization resources (visuals). Next, you deactivate the environment.


The topic "[Delete Data Visualization visuals](#)" describes the resources that are deleted when you delete a Virtual Warehouse as described in the procedure below.

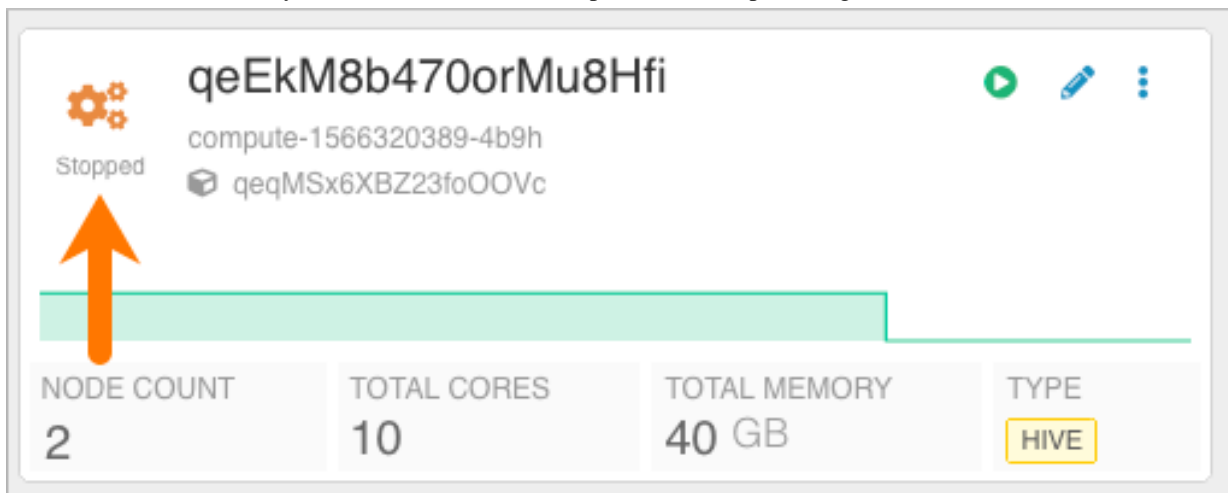
Deleting the Virtual Warehouse and Data Visualization deployments


You must delete the Virtual Warehouses in the CDW Environment and the Data Visualization resources (visuals).

About this task

Procedure

- Log in to the CDP web interface, navigate to Data Warehouse Overview , note the name of the Virtual Warehouse you want to modify, and note which Database Catalog it is configured to access.
- In the Virtual Warehouse you want to delete, click Suspend  to stop running the Virtual Warehouse.



- Click the options  of the Virtual Warehouse you want to delete, and select Delete. When you delete a Virtual Warehouse, the Data Visualization visuals are deleted as described in "[Delete Data Visualization visuals](#)".


Deactivating the environment

Assuming you have already deleted your Virtual Warehouses, you need to delete your non-default (custom) Database Catalog and Virtual Warehouses, and then deactivate the environment.

About this task

You cannot delete default Database Catalogs created during environment activation. Default catalogs are deleted when the environment is deactivated.

Procedure

1. Log in to the CDP web interface, navigate to Data Warehouse Database Catalogs , and locate your Database Catalog.
2. Click Actions, and select Delete.
3. Search and locate the environment that you want to deactivate.
4. Click Deactivate .
5. Click OK to deactivate the environment.

Automatically restoring the environment

You can automatically reactivate the entire environment using the CLI, which includes your cluster. Automatic restoration enables all settings of the environment that you backed up.

An environment-level automated restoration CLI option restores the environment, the deployed Database Catalog, Virtual Warehouse, and Data Visualization entities. If a Virtual Warehouse or a Data Visualization object is not present on the cluster, but the backup file contains it, the Virtual Warehouse or Data Visualization object will be restored to the cluster. If such an entity is already deployed, no changes or configuration updates will take place.

The CLI `dw restore-cluster` command can be used in the following ways:

- Passing the environment's Cloudera resource name (crn) will activate the cluster from the backup file and restore all the entities and database contents.
- Passing an activated environment identifier will restore all the entities and database contents to the running environment. This is useful when you need to change activation parameters, but requires manual reactivation.

Automatic restoration consists of the following operations in the order shown here:

- Activates the environment and waits for infrastructure creation
- Applies the cluster services and sets up the environment
- Creates the default Database Catalog
- Updates the Database Catalog configuration to apply customer configuration customizations
- Starts the Hue database restore job in the database catalog namespace asynchronously
- Deploys the Virtual Warehouse instances
- Deploys the Data Visualization instances
- Starts the Data Visualization restore job in the individual namespaces asynchronously

Before you begin the restoration of Hue, if you cannot bring down the cluster, use the recommended workaround to [disable end user access](#) to the cluster endpoints. The automatic restoration process does not wait for the database operations to be finished. You must monitor the status of the jobs using the operation id to make sure the process finishes. For more information, see [Monitoring Hue and Data Visualization restoration](#) and [Monitoring environment restoration](#).

Details about the restore process

The restore process is designed to be an idempotent process, it can be restarted as many times as you want. If the environment is activated and healthy, you can run the restore operation multiple times to restore the Virtual Warehouse and Data Visualization objects. For every restore operation, the Hue database restore will run. This operation will overwrite the Hue database contents. If a Virtual Warehouse or a Data Visualization object is not present on the cluster, but the backup file contains it, it will be restored to the cluster. In case such an entity is already deployed, no changes or configuration updates will take place.

Restoring the environment and objects

You learn how to use the `dw restore-cluster` command, which you can use either to pass the environment's Cloudera resource name (crn) or to pass the identifier of an activated environment.

About this task

Passing the Cloudera resource name (crn) will activate the cluster from the backup file and restore all the entities and database contents.

Passing an activated environment resource name will restore all the entities and database contents to the running environment. Passing the environment identifier is useful when you need to change activation parameters, but requires manual reactivation.

In the steps below, use `dw restore-cluster` to pass the Cloudera resource name (crn) to activate the cluster.

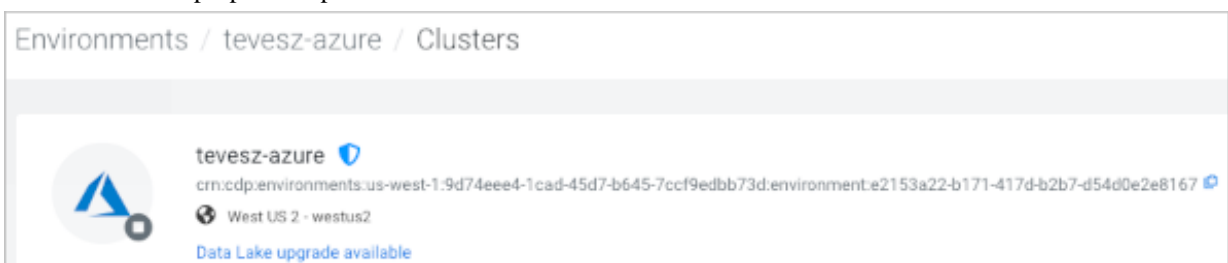
Before you begin

- Your Azure cluster must run version 1.6.3-b319 (released May 5, 2023) or later.
You cannot automatically activate an Azure cluster that runs version 1.6.2-b197 (released Feb 13, 2023) or earlier.
- You must use the same Cloudera Data Warehouse version to restore files that you used to back up those files.
Using a backup file from 1.6.2-b197 (released Feb 13, 2023) for restoration will not work.
- Check that the size of your Hue backup file is smaller than 6GB. If the backup file 6GB or larger, do not automatically restore the environment. Go to the procedure for [manually restoring the environment](#).

Procedure

- Get your environment resource name from the Cloudera portal by selecting the environment that is not activated, and clicking Manage.

The environment properties open.



Under the environment resource name the Cloudera resource name (crn) appears.

```
crn:cdp:environments:us-west-1:98765432-abcd-45d7-b645-7ccf9edbb73d:environment:00000000-7bf2-4aeb-af71-f2bf2c038588
```

- Create a CLI skeleton file to serve the base file for the restore command.
For example, replace your environment resource name placeholder `<your cluster name>` with the environment resource name of the newly activated cluster (for example `env-npk886` shown step 3 of [Reactivating the environment](#)).

```
export CLUSTER_NAME="<your cluster name>"
cdp \
  dw restore-cluster \
  --generate-cli-skeleton 1>restore_${CLUSTER_NAME}_cli_input.json
```

- Open `restore_<CLUSTER_NAME>_cli_input.json` for editing, and fill in the `clusterId` and the `data` fields.
For example:

```
{
  "clusterId": "crn:cdp:environments:us-west-1:98765432-abcd-45d7-b645-7ccf9edbb73d:environment:00000000-7bf2-4aeb-af71-f2bf2c038588",
  "data": "UESDBBQ...AAAAAAAAAAAAABkYXRhUESFBgAAAAABAAEAMgAAAKuBAQAAAA==" ,
}
```

4. Use the `dw restore-cluster` command, provide the same `CLUSTER_NAME` as you used in step 3 and use the `CDP_PROFILE` from your [CLI configuration](#).

```
export CLUSTER_NAME="<your cluster name>"
export CDP_PROFILE="<your CDP CLI profile>"

cdp \
  --profile ${CDP_PROFILE} \
  dw restore-cluster \
    --cli-input-json file://restore_${CLUSTER_NAME}_cli_input.json
```

Example output:

```
{
  "clusterId": "crn:cdp:environments:us-west-1:98765432-abcd-45d7-b645-7ccf9edbb73d:environment:00000000-7bf2-4aeb-af71-f2bf2c038588",
  "operationId": "62408134-3d8c-46e8-a914-0f427fc3b1b1",
  "action": "Create",
  "message": "the cluster will be created",
  "dbcRestorePlans": [
    {
      "ref": "test-aws-dl-default",
      "id": "warehouse-1692719478-xrm4",
      "action": "Create",
      "message": "the SDX-type DB Catalog will be created based on the data referenced in the backup as test-aws-dl-default"
    }
  ],
  "hueRestorePlans": [
    {
      "ref": "test-aws-dl-default",
      "id": "warehouse-1692719478-xrm4",
      "action": "Create",
      "message": "Hue restore is started for warehouse-1692719478-xrm4 DB Catalog, referenced in the backup data as test-aws-dl-default. Restore will overwrite Hue database with the backup if it isn't empty."
    }
  ],
  "hiveRestorePlans": [
    {
      "ref": "test-hive",
      "action": "Create",
      "message": "the test-hive Hive Virtual Warehouse will be created and attached to the warehouse-1692719478-xrm4 DB Catalog"
    }
  ],
  "impalaRestorePlans": [
    {
      "ref": "test-impala",
      "action": "Create",
      "message": "the test-impala Impala Virtual Warehouse will be created and attached to the warehouse-1692719478-xrm4 DB Catalog"
    }
  ],
  "vizRestorePlans": [
    {
      "ref": "test-viz",
      "action": "Create",
      "message": "the test-viz Data Visualization will be created"
    }
  ]
}
```

```
}

```

After several minutes the environment will be activated, the Virtual Warehouses will be created in the new cluster and attached to the Database Catalog. The Virtual Warehouse and Data Visualization ids will be changed.

The Data Visualization database will be recovered. However, because this is a new deployment, the recovered connections will be broken.

5. Monitor the environment restoration as described in [Monitoring environment restoration](#).
6. Adjust the Data Visualization connection settings to point to the new Virtual Warehouse(s).

Testing the restoration

The environment and entity automated restore process can deploy objects to any given environment. You can validate all entities and their settings to gain confidence that the restore operation will succeed for the production environment.

If the data lake storage path for the restored environment is different from than the data lake storage path for the backed up environment, the database restore jobs will fail. The jobs will not be able to reach the database backup file paths; this is expected. The restoration will report the failure, but all entity deployment occurs normally. Customers with older environments might want to consider either testing the restore process first or making a [manual backup of the cluster](#) and its properties.

To test the restore process, define a test data lake environment `crn` in the restore file and follow the details from the [“Automatically restoring the environment”](#) page.

Monitoring Hue and Data Visualization restoration

The restore process is designed to be an idempotent process, it can be restarted as many times as you want. In case the environment is activated and healthy, the restore operation can be run multiple times to restore the Virtual Warehouse and Data Visualization objects.

Hue automatic restoration

For every restore operation, the Hue database restore will run. This operation will overwrite the Hue database contents. Automatic restoration of Hue loads the saved query and query history to the new cluster. In case the restore operation is called multiple times, the Hue database restore job will be run. This job will overwrite the current query history and saved queries with the contents of the backup.

Monitoring Hue restoration

The restoration starts a job to load the database dump file, but does not wait for the job to complete. If you have a large database, the job can take up to an hour to complete. Ensure you allow enough time for the job to succeed. If the job does not succeed, [troubleshoot Hue restoration](#).

To monitor Hue restoration, log into the cluster and monitor the job status under the database catalog namespace.

```
$ kubectl get jobs -n <database catalog id>
```

The output that shows the hue-restore job looks something like this:

```
$ kubectl get jobs -n warehouse-1692037411-96hk
  NAME                                COMPLETIONS  DURATIO
N  AGE
  hue-restore-edeb2b8bd-1d53-4d23-a0f9-87d8ec658f74  1/1           11s
  hue-query-processor-db-create-job                1/1           8s
  42h
```

Data Visualization automatic restoration

If a Data Visualization object is not present on the cluster, but the backup file contains it, it will be restored to the cluster. In case such an entity is already deployed, no changes or configuration updates will take place.

Automatic restoration of Data Visualization loads the dashboards, tables, and connections to the new applications. Make sure to wait for the job to finish before destroying the cluster.

To monitor restoration of Data Visualization, you can log into the cluster and see the job status under the viz namespace using the following command.

```
$ kubectl get jobs -n <data visualization id>
```

The output will be similar to this, the viz-restore job shows the status.

```
$ kubectl get jobs -n viz-1692216942-fc2g
NAME                                COMPLETIONS  DURAT
ION  AGE
viz-restore-d874515a-be7e-4902-ac75-269c14f9580c  1/1           3m3s
10m
viz-webapp-vizdb-create-job           1/1           57s
99m
```

The job logs contain the upload path where the backup file has been downloaded from.

Automatic restoration of Data Visualization

Automatic backup and restore for Data Visualization extracts the dashboards, tables and connections. Make sure to wait for the job to finish before destroying the cluster. In the event of a restoration failure, try [manually restoring Data Visualization](#).

Monitoring Data Visualization restoration

To monitor the restoration of Data Visualization, you can log into the cluster and see the job status under the viz namespace using the following command.

```
$ kubectl get jobs -n <data visualization id>
```

The output looks something like this:

```
$ kubectl get jobs -n viz-1692216942-fc2g
NAME                                COMPLETIONS  DURATION
AGE
viz-restore-d874515a-be7e-4902-ac75-269c14f9580c  1/1           3m3s
10m
viz-webapp-vizdb-create-job           1/1           57s
99m
```

Manually restoring the environment

To manually reactivate the environment, you modify configurations, recreate Virtual Warehouses, and restore Hue and Data Visualization. You can automatically reactivate the environment using a number of procedures.

Manual restoration consists of the following operations in the order shown below:

Manually reactivating the environment

You learn how to reactivate the AWS or Azure environment.

About this task

Follow this procedure to [reactivate the AWS environment](#) or [reactivate the Azure environment](#), and then ensure the reactivated environment is configured the same as the deactivated one. You must add the activated parameters that were backed-up and documented in the previous steps to the new reactivated environment.

Using the CDP CLI

You parameterize the CLI create-cluster command to activate the cluster, as described in the [CLI documentation](#).

1. Activate the cluster by passing the options you retrieved backing up AWS or activating Azure.

To see all options run the following command.

```
cdp dw create-cluster -help
```

Examples for shorthand and JSON syntax are available.

For example, an Azure CLI activation option looks something like this:

```
cdp dw create-cluster --environment-crn <crn:cdp:environments:us-west-1:
abc:environment:123> \
--use-overlay-network --no-use-private-load-balancer \
--azure-options \
userAssignedManagedIdentity="<full managed identity identifier>",subnetI
d="<full subnet identifier>",enableSpotInstances=false,logAnalyticsWorks
paceId="<full log analytics workspace identifier>" \
--profile <customer profile>
```

An AWS CLI activation option looks something like this:

```
cdp dw create-cluster --environment-crn <crn:cdp:environments:us-west-1:
abc:environment:123> \
--use-overlay-network --use-private-load-balancer \
--aws-options \
lbSubnetIds=<list of subnet identifiers>,workerSubnetIds=<list of subnet
identifiers>,enableSpotInstances=false --profile <customer profile>
```

2. (Optional) If you need to preserve the old environment URL, specify the custom subdomain in the `dw create-cluster` command.

```
--custom-subdomain (string)
```

For example:


```
--custom-subdomain env-qwertyu.dw
```

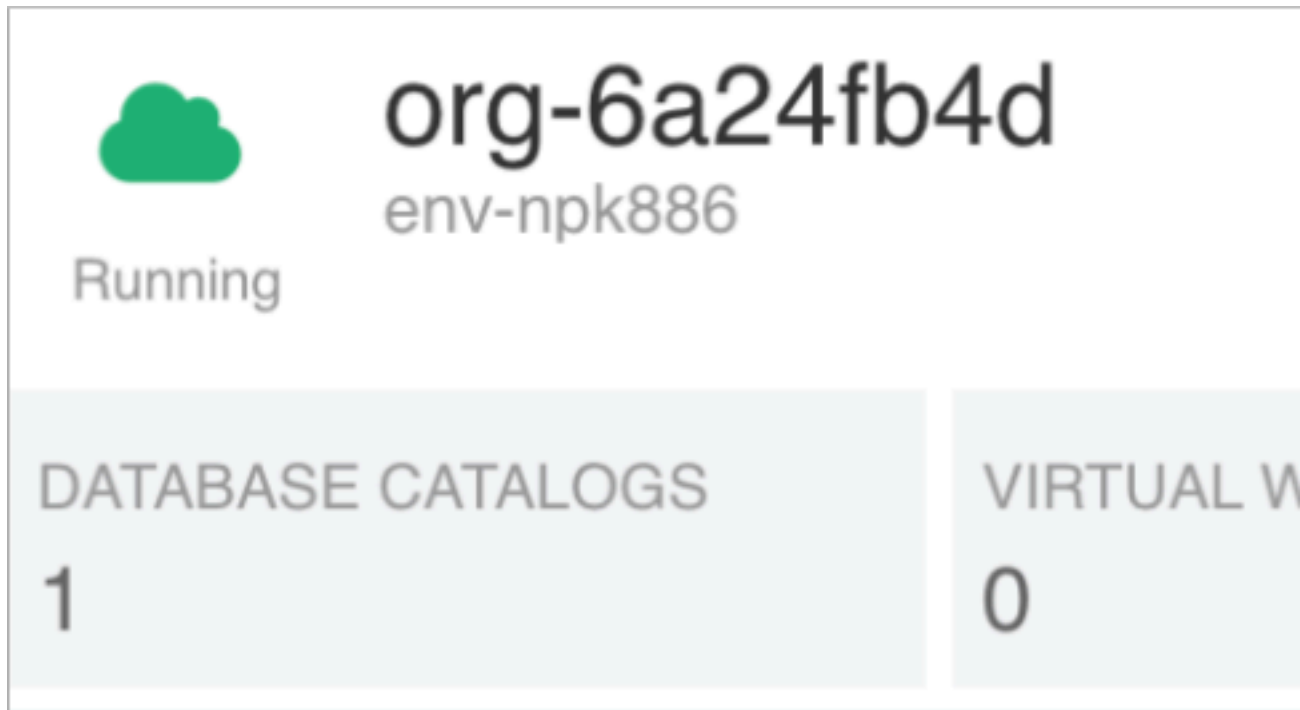
For more information about cluster URLs after reactivation, see [“Cluster URLs after reactivation”](#). For more information about the CLI, see [CDP CLI documentation](#).

Using the CDW UI

Procedure

1. In the CDW service, expand the Environments column by clicking More....
2. In Environments, search for and locate the environment that you want to activate.

- Click the start icon  to activate the environment.
The default SDX Database Catalog (represented by the tile on the right) is



created.

the CDW cluster is restored, the IDs of the environment (represented by the tile on the left) and Database Catalog change.

- In the Activation Settings, configure the environment using the information you gathered when you backed up activation parameters.
- (Optional) If you need to preserve the old environment URL, specify the custom subdomain.
In **Custom Environment Subdomain**, if the environment is env-qwertyu.dw, for example, specify the custom subdomain in the following format:

```
<old environment identifier>.dw
```

CDW UI example:

Custom Environment Subdomain

env-qwertyu.dw

For more information about cluster URLs after reactivation, see [“Cluster URLs after reactivation”](#).

- Apply changes.

Modifying configurations after activation

From the CDW UI, you configure alert and observability settings you had in the old environment.

Procedure

- In your environment tile, click Edit, and in **Alert Settings**, and add the alert settings.
- In your environment tile, click Edit, and in **Observability**, and add the observability settings.



Disabling end user access

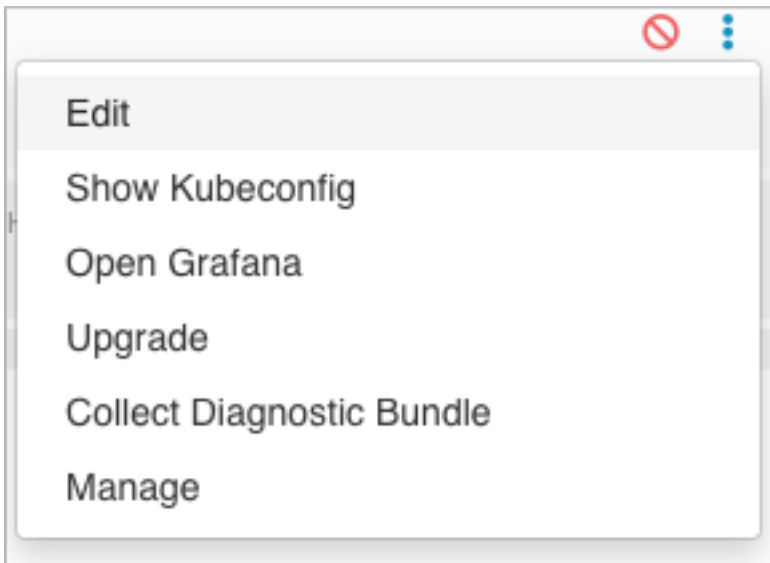
If your business cannot tolerate downtime, you can prevent end user access to your clusters by disabling end user access. Disabling end user access is recommended only when bringing down your clusters is not feasible.

About this task

This procedure is recommended for AWS or Azure clusters that you cannot bring down for some reason.

Procedure

1. In the Data Warehouse service, click Overview and expand the Environments column.
2. Click  and locate an environment having a Database Catalog you activated for CDW.
3. Click the environment options  and select Edit.



4. Click **Configurations** and make a note of the values for setting **Enable-IP_CIDR** load balancer.
5. Obtain your IP address.
For example, run `ifconfig` on Windows or `ipconfig` on Linux.

- Set **Enable IP-CIDR** for the load balancer to your IP address.
For example, replace x.x.x.x/32 with your IP address 201.123.45.100/32.

The screenshot shows the 'CONFIGURATIONS' tab in the CDW console. It features a 'Description' field with the placeholder text 'Please enter the description'. Below this is the 'Enable IP-CIDR for Kubernetes cluster:' section with a text input field containing '0.0.0.0/0'. The 'Enable IP-CIDR for the load balancer:' section has a text input field containing 'x.x.x.x/32'. At the bottom, there is a checkbox labeled 'Enable CloudWatch logs' with an information icon to its right.

- Apply changes.
No user except you can access the Virtual Warehouses. Wait until the cluster updates itself and goes back to Running state to continue the process.

What to do next

You must [enable end user access](#) later.

Recreating the Virtual Warehouses

You recreate the Virtual Warehouses in a few steps.

About this task

To recreate the Virtual Warehouses, you use the `dw restore-cluster` command. This command also restores Data Visualization apps, but additional steps are required as described in section [“Restoring Data Visualization”](#).

Procedure

- Create a CLI skeleton file to serve the base file for the restore command.
For example, replace `<your new cluster id>` placeholder with the ID of the newly activated cluster (for example `env-npk886` shown step 4 of [Reactivating the environment](#)).

```
export NEW_CLUSTER_ID="<your new cluster id>"
cdp \
  --profile ${CDP_PROFILE} \
  dw restore-cluster \
```



```
--generate-cli-skeleton \ file://restore_${NEW_CLUSTER_ID}_cli_input.json
```

2. Open `restore_<new-cluster-id>_cli_input.json` for editing, and fill in the `clusterId` and the `data` fields.

To accomplish this, copy contents of the `dump_${CLUSTER_ID}.json` file from the [backup step](#) to the `data` field in the `restore_<new-cluster-id>_cli_input.json` file.

For example:

```
{
  "clusterId": "env-npk886",
  "data": "UESDBBQ...AAAAAAAAAABkYXRhUESFBgAAAAABAAEAMgAAAKuBAQAAAA==",
}
```

3. Use the `dw restore-cluster` command with the CLI input JSON file created in the previous step.

```
cdp \
--profile ${CDP_PROFILE} \
dw restore-cluster \
--cli-input-json file://restore_${NEW_CLUSTER_ID}_cli_input.json
```

Example output:

```
{
  "clusterId": "env-npk886",
  "operationId": "e279bc25-6eb2-45d5-b2a5-ebdaf8d9c809",
  "dbcRestorePlans": [],
  "hiveRestorePlans": [
    {
      "ref": "org-master-prd-hive",
      "action": "Create",
      "message": "the org-master-prd-hive Hive Virtual Warehouse will be crated and attached to the warehouse-1676473680-986m DB Catalog"
    }
  ],
  "impalaRestorePlans": [
    {
      "ref": "impala-master-prd-impala",
      "action": "Create",
      "message": "the impala-master-prd-impala Impala Virtual Warehouse will be crated and attached to the warehouse-1676473680-986m DB Catalog"
    }
  ],
  "vizRestorePlans": [
    {
      "ref": "viz-analyst",
      "action": "Create",
      "message": "the viz-analyst Data Visualization will be created"
    },
    {
      "ref": "viz-scientist",
      "id": "viz-1678182673-jzxs",
      "action": "Skip",
      "message": "the Data Visualization viz-scientist exist with id viz-1678182673-jzxs, no change will be applied"
    }
  ]
}
```

After several minutes the Virtual Warehouses will be created in the given cluster and attached to the Database Catalog. The Virtual Warehouse and Data Visualization ids have changed.

4. Monitor the environment restoration as described in [Monitoring environment restoration](#).

What to do next

You must use the new Virtual Warehouse and Data Visualization ids to restore the HUE or DataViz databases as described in the next section.

Restoring Hue

The backup procedure automatically saved the Hue database content and placed the content into the configured logs or data folders based on availability. Using the saved content, the restore process loads the data for the new Hue deployments.

Manually restoring Hue from a smaller than 6GB backup

You can manually restore the Hue database instance that you backed up. You can use this procedure for manually restoring Hue when your Hue backup is smaller than 6GB.

About this task

In the manual backup of Hue, you followed steps to dump the entire Hue database. In the following procedure, you move the Hue backup file from the dump to the new CDW environment.

Before you begin

- Check that the size of your Hue backup is smaller than 6GB.
If your Hue backup is 6GB or larger, go to “Manually restoring Hue 6GB or larger backup”.
- Do not open the Hue web interface prior to completing the steps below.
- During the manual or automatic Hue database restore operation it is critical to block any traffic to the running Hue services. If you cannot bring down the cluster, use the recommended workaround to [disable end user access](#) to the cluster endpoints. Failing to do so results in errors in addition to existing key constraints and other issues.

Procedure

1. (Optional) Edit the Hue statefulset Hue backup files that are larger than 3Gb and require a significant amount of memory to restore the database contents.

Sufficient memory is not available for the Hue pod by default. To successfully load the data, you must increase the Hue pod memory limit.

2. (Optional) Check file sizes and configure the memory accordingly.

```
kubectl edit statefulset huebackend -n <new Virtual Warehouse ID>
```

3. Set the hue container memory limit to 24Gb to provide leeway for the load command.

```
name: hue
resources:
  limits:
    memory: 24G
  requests:
    cpu: "1"
    memory: 8192M
```

4. Copy the Hue backup data to the hue pod on the new Virtual Warehouse cluster.

For example:

```
$ kubectl cp /tmp/data.json compute-1677087760-cj7q/huebackend-0:/tmp/data.json -c hue
```

5. Connect to Hue pod on new Hive/Impala Virtual Warehouse cluster.

```
$ kubectl exec -it huebackend-0 -n <new Virtual Warehouse ID> -c hue - /bin/bash
```

6. Load data to Hue onto the new Virtual Warehouse cluster.

```
$ ./build/env/bin/hue loaddata --exclude auth.permission --exclude contenttypes --ignorenonexistent /tmp/data.json
```

7. Go to the Hue UI and verify the saved queries and query history are showing up in the new CDW environment. Also, verify that Hue is working.
8. If saved queries and history are showing up, and Hue is working, proceed to the next step; otherwise, go to [Troubleshooting Hue restoration](#).
9. After the data load is finished, revert the container memory limit back to the original 8192M setting.

Manually restoring Hue from a 6GB or larger backup

You can manually restore the Hue database instance that you backed up. You can use this procedure for manually restoring Hue when your Hue backup is 6GB or larger.

About this task

In the manual backup of Hue, you followed steps to dump the entire Hue database. In the following procedure, you move the Hue backup file from the dump to the new CDW environment.

Before you begin

- Check that the size of your Hue backup is 6GB or larger.
If your Hue backup is smaller than 6GB, go to “Manually restoring Hue from a smaller than 6GB backup”.
- Do not open the Hue web interface prior to completing the steps below.
- During the manual or automatic Hue database restore operation it is critical to block any traffic to the running Hue services. If you cannot bring down the cluster, use the recommended workaround to [disable end user access](#) to the cluster endpoints. Failing to do so results in errors in addition to existing key constraints and other issues.

Procedure

1. Connect to Hue pod on new Hive/Impala Virtual Warehouse cluster.

```
$ kubectl exec -it huebackend-0 -n <new Virtual Warehouse ID> -c hue - /bin/bash
```

2. Clean the Hue database by running the flush command from the hue pod

```
./build/env/bin/hue flush
```

3. Split the json into smaller chunks.

```
HUE_BACKUP_ORIG_FILE=data.json # Change the the correct path
HUE_BACKUP_CHUNKS_DIR=hue_backup_parts # Change if needed

mkdir -p ${HUE_BACKUP_CHUNKS_DIR}
rm -rf ${HUE_BACKUP_CHUNKS_DIR}/part* | true
```

```
jq -cn --stream 'fromstream(1|truncate_stream(inputs))'
${HUE_BACKUP_ORIG_FILE} | split -l 5000 -a 4 -d -
${HUE_BACKUP_CHUNKS_DIR}/part
find ${HUE_BACKUP_CHUNKS_DIR}/part* -maxdepth 1 -type f ! -name "*.*" -exec
sh -c 'jq --slurp "." "${0}" | gzip > "${0}.json.gz' {} \;

ls -alh ${HUE_BACKUP_CHUNKS_DIR}
tar cvzf ${HUE_BACKUP_ORIG_FILE}.tar.gz
${HUE_BACKUP_CHUNKS_DIR}/part*.json.gz

echo "Generated the chunked backup file"

ls -alh ${HUE_BACKUP_ORIG_FILE}.tar.gz # This is our final output file
```

4. Import the chunked JSON:

Move the tarball of backup chunks to the cluster pod. Extract the tarball to a directory, for example /tmp/hue_backup_parts.

5. Run hue loaddata command on the pod.

```
/opt/hive/build/env/bin/hue loaddata --verbosity 3 --exclude auth.permission
--exclude contenttypes --ignorenonexistent $(find /tmp/hue_backup_parts
-type f -name '*.json.gz') # UPDATE THE PATH TO THE EXTRACTION DIRECTORY
FROM THE PREVIOUS STEP
```

Troubleshooting Hue restoration

An inaccessible Hue UI and a duplicate key error are issues you might encounter after attempting to manually restore Hue.

Inaccessible Hue UI

If the Hue UI is not accessible, try the following workarounds.

1. Kill the Hue frontend pod, providing the Hue frontend pod ID, for example huefrontend-5bdc7bc7b8-8lpgj.

```
$ kubectl get pods -n <new Virtual Warehouse ID> # the pod name
$ kubectl delete pod <Hue frontend pod ID> -n <new Virtual Warehouse ID>
```

2. Increase virtual warehouse memory if the database size is large.

```
## edit hue backend container to max 16GB (note: there are two containers:
## busybox and hue)

$ kubectl edit sts huebackend -n <new Virtual Warehouse ID>
```

Duplicate key error

If you see the following duplicate key error, perform the steps below:

```
django.db.utils.IntegrityError: Problem installing fixture '/tmp/data.json':
Could not load desktop.Document(pk=2): duplicate key value violates unique
constraint "desktop_document_content_type_id_object_id_xyzxyz_uniq" DETAIL:
Key (content_type_id, object_id)=(8, 6) already exists.
```

This error likely occurs if you open Hue before completing the restoration and do not follow the prerequisite above not to open the Hue web interface.

1. Connect to Hue pod on new Hive/Impala Virtual Warehouse cluster.

```
$ kubectl exec -it huebackend-0 -n <new Virtual Warehouse ID> -c hue - /bin/bash
```

2. Clean the Hue database by running this command from the hue pod:

```
./build/env/bin/hue flush
```

3. Go to step 6 in [Restoring Hue](#) to attempt to load the data again.

Restoring Data Visualization

You can restore the Data Visualization instance that you backed up.

Before you begin

Do not open the Data Visualization web interface prior to applying the steps below.

Procedure

1. Find the necessary information like database name, host, port, user, and password. On the new cluster, this information will be different from the old cluster.

```
$ kubectl get secrets/pg-db-secret -o=jsonpath={.data.'\pgpass'} -n <viz-id> | base64 -D
```

```
postgres-service:<port>:<database name>:<user>:<password>
```

For example:

```
$ kubectl get secrets/pg-db-secret -o=jsonpath={.data.'\pgpass'} -n viz-1680129861-kbtr | base64 -D
```

```
postgres-service:5432:metastore:hive:ZufGC6Dmh03N1iW042uTosZtr4XvCtJIYPQ==
```

Using these properties you will be able to connect to the database and load the backup.

2. Use kubectl or k9s to access Hue container in the targeted CDW environment (need KUBECONFIG setup), and find the namespace for Hue and for Data Visualization.

```
$ kubectl get pods --all-namespaces --field-selector metadata.name=huebackend-0
```

```
$ kubectl get pods --all-namespaces --field-selector metadata.name=viz-webapp-0
```

If you have multiple Data Visualization instances running, match the new Data Visualization namespace names to the user-friendly names on the CDW UI.

3. Select one of the Hue namespaces and copy the dump file to your container.

```
$ kubectl cp ~/Downloads/logs/viz_pg_dump.tar <Hue container>/huebackend-0:/opt/hive/viz_pg_dump.tar -c hue
```

For example:

```
$ kubectl cp ~/Downloads/logs/viz_pg_dump.tar impala-1679934278-6pgc/huebackend-0:/opt/hive/viz_pg_dump.tar -c hue
```

4. Shell into the container.

```
$ kubectl exec -it huebackend-0 -n <Hue container> -c hue -- /bin/bash
```

5. Load the dump back to the database, taking care if you have multiple DataViz instances to load the contents back to the right database.

```
pg_restore -d <DataViz database ID> -h postgres-service ./viz_pg_dump.tar
-c -U hive
```

For example:

```
pg_restore -d viz-1680137534-87s4_vizdb -h postgres-service ./viz_pg_dum
p.tar -c -U hive
```

The `pg_restore` command might output the following errors as it runs:

```
....
pg_restore: [archiver (db)] Error from TOC entry 258; 1259 302230 TABLE
apps_apikey hive
pg_restore: [archiver (db)] could not execute query: ERROR: table "apps_a
pikey" does not exist
Command was: DROP TABLE public.apps_apikey;

WARNING: errors ignored on restore: 10
```

It is safe to continue restoring Data Visualization; just ignore the errors.



Enabling end user access

If you disabled end user access before recreating your Virtual Warehouse, you must enable access afterward.

About this task

Perform this task only if you disabled end user access.

Procedure

1. In the Data Warehouse service, click Overview and expand the Environments column.
2. Click  and locate an environment having a Database Catalog you activated for CDW.
3. Click the environment options  and select Edit.
4. Click **Configurations** and set the value of **Enable-IP_CIDR** load balancer to the setting you noted when disabling end user access.
5. Apply changes.

Monitoring environment restoration

Restoring the CDW environment is an unvaried process; no configuration update or settings change are applied to existing Virtual Warehouses or Data Visualization applications.

After restoration, a Virtual Warehouse has exactly the same settings as the original Virtual Warehouse according to the following principles.

- Configurations are not copied as is; rather a new configuration is created. All the changes made to the Virtual Warehouse configuration throughout its lifecycle are applied on top of the new configuration.
- Values that were undefined or not present in earlier versions are set to the default.
- Configurations are mapped 1-to-1 during restoration.

Track the progress of the cluster restoration using the operationID from the cluster-restore response. Due to the unvaried nature of the restore process, if any Virtual Warehouse is missing because you deleted it accidentally, or it failed to come up, you can choose to try again.

Steps

1. Check that there are no errors after running the cluster-restore command.

Example: A cluster-restore command response free of errors looks like this:

```
{
  "clusterId": "env-npk886",
  "operationId": "acbe40f4-560b-485c-833c-451a64bb76c4",
  # truncated output
}
```

2. Use the CDP CLI `dw list-events` commands in conjunction of the operation-id (obtained in the first step) to see the restoration progress.

```
cdp \
  --profile ${CDP_PROFILE} \
  dw list-events \
  --operation-id acbe40f4-560b-485c-833c-451a64bb76c4
```

The command returns the most recent events. You can use `--asc` switch to flip the ordering and see the first event. You can limit the output with the `--limit` switch.

You might experience a slight delay before the event appears in the audit app.

The output looks something like this:

```
{
  "events": [
    {
      "operationId": "acbe40f4-560b-485c-833c-451a64bb76c4",
      "event": "RestoreCluster",
      "message": "{\"type\":\"info\",\"message\":\"restore cluster operation for env-m6mcfid has finished\",\"error\":null}",
      "timestamp": "2023-08-28T12:12:53+00:00"
    },
    {
      "operationId": "acbe40f4-560b-485c-833c-451a64bb76c4",
      "serviceId": "compute-1693224718-abcd",
      "event": "Completed",
      "message": "Started hive Virtual Warehouse",
      "timestamp": "2023-08-28T12:12:50+00:00"
    }
  ]
}
```

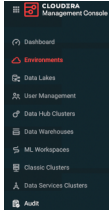
Steps with Auditing Events


1. Check that there are no errors after running the cluster-restore command.

Example: A cluster-restore command response free of errors looks like this:

```
{
  "clusterId": "env-npk886",
  "operationId": "acbe40f4-560b-485c-833c-451a64bb76c4",
  # truncated output
}
```

2. Navigate to the CDP Management Console Audit .



3. Click Expand  to see information about progress of, or possibly about errors in, the request to restore the cluster.

61323260-ace8-426c-9f19-0824247e14ad		acbe40
API Version	v3	
Mutating		
Request Parameters	{"message":"deploying impala-1","type":"info"}	
Response Parameters		
Source IP Address	127.0.0.1	
Useragent	Swagger-Codegen/1.0.0/java	

4. Alternatively, to see information about progress of, or possibly about errors, in the request to restore the cluster use the CDP CLI.

Using list-events in the audit command section can return the restore operation events. Passing the operationId to the requestId filter parameter returns the output shown below. For example:

```
cdp \
  --profile ${CDP_PROFILE} \
  audit list-events \
  --from-timestamp $(TZ=UTC date -v -ld '+%FT%T') \
  --to-timestamp $(TZ=UTC date -v -0M '+%FT%T') \
  --request-id acbe40f4-560b-485c-833c-451a64bb76c4
```

Example output (truncated):

```
{
  "auditEvents": [
    {
      "version": "1.1.0",
      "id": "238e6d6e-196f-4c76-b75b-c97b3dd3d4d5",
      "eventSource": "dw",
      "eventName": "RestoreCluster",
      "timestamp": 1678446458555,
      "actorIdentity": {
        "actorCrn": "crn:..."
      },
    },
  ],
  "accountId": "9d74eee4-...",
}
```



```
"requestId": "acbe40f4-560b-485c-833c-451a64bb76c4",
"apiRequestEvent": {
  "requestParameters": "{\"message\": \"restore cluster operation for env-
jhnwzk has finished\", \"type\": \"info\"}\",
  "mutating": true
}
},
```