CDP Public Cloud Management Console

Data Lakes

Date published: 2019-08-22
Date modified:

CLOUDERA

Legal Notice

Contents

Introduction to Data Lakes

A Data Lake is a service which provides a protective ring around the data stored in a cloud object store, including authentication, authorization, and governance support.

A Data Lake provides a way for you to centrally apply and enforce authentication, authorization, and audit policies across multiple workload clusters—even as the workload clusters are created and terminated based on demand. When you register an environment in CDP, a Data Lake is automatically deployed for that environment. The Data Lake runs in the virtual network of the environment and provides security and governance layer for the environment's workload resources (such as Data Hub clusters). All workload resources are automatically "attached" to the Data Lake: the attached cluster workloads access data and run in the security context provided by the Data Lake.

While workloads are temporary, the security policies around your data schema are long-running and shared for all workloads. As your workloads come and go, the Data Lake instance lives on, providing consistent and available security policy definitions and auditing that are available for current and future workloads. All information related to schema (Hive Metastore), security policies (Ranger), audit (Ranger), and metadata management and governance (Atlas) is stored on external locations (external databases and cloud storage). These external locations leverage the security and availability features guaranteed by the cloud provider to ensure that even if one or all virtual hosts in a Data Lake fail, the storage remains and the Data Lake hosts can be replaced and reattached to the data storage with little or no downtime and no data loss.

A Data Lake cluster uses Apache Knox to provide a protected gateway for access to Data Lake component UIs. Knox is also installed on all workload clusters, providing a protected gateway for access to cluster component UIs.
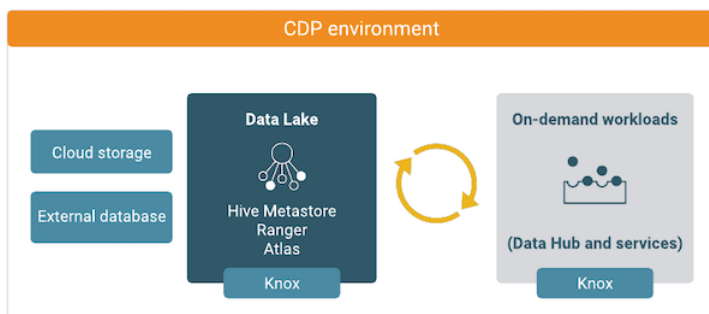
The following technologies provide capabilities for the Data Lake:

Component	Technology	Description
Schema	Apache Hive Metastore	Provides Hive schema (tables, views, and so on). If you have two or more workloads accessing the same Hive data, you need to share schema across these workloads.
Policy	Apache Ranger	Defines security policies around Hive schema. If you have two or more users accessing the same data, you need security policies to be consistently available and enforced.
Audit	Apache Ranger	Audits user access and captures data access activity for the workloads.
Governance	Apache Atlas	Provides metadata management and governance capabilities.
Gateway	Apache Knox	Supports a single workload endpoint that can be protected with SSL and enabled for authentication to access to resources.
Storage	Cloud provider storage, such as AWS S3 or Azure Storage	Isolates Data Lake storage from the compute resources. Data Lake storage is created when the Data Lake is created and is deleted when the Data Lake is terminated. Once created, the Data Lake storage lifecycle is separate from the Data Lake hosts' lifecycle: in case of a Data Lake host failure, the Data Lake storage remains and is reattached to new Data Lake host or hosts.

Related Information

Data Lake security

Apache Ranger authorization

Audit overview

Governance overview

Azure Load Balancers in Data Lakes and Data Hubs

Data Lake storage

Data Lake storage leverages the security and high-availability guarantees from the cloud provider, allowing Data Lakes to regenerate hosts as needed, without data loss and with little or no downtime for workload services.

Data Lake storage is designed to reside in external storage locations separate from the hosts running the Data Lake services. This configuration protects CDP workloads from data loss should one or all of the Data Lake nodes fail. New hosts created in the Data Lake repair process are re-attached to the persistent data storage and Data Lake services return to normal.

> **Disposition: / Status:**
> For the moment, we are avoiding talking about "backup" until we have a more complete solution.

> **Disposition: / Status:**
> Avoid references to multiple availability zones until we are taking advantage of that feature from the cloud providers.

The following table provides links to information from cloud providers about the service level guarantees they provide for each storage type.

AWS	Azure	GCP	Component Storage Content
Amazon RDS See Automated backups, snapshots, and automatic host replacement.	Azure Database for PostgreSQL See Availability guarantees.	Google Cloud SQL See Cloud SQL.	HMS SQL catalog Ranger policy data Cloudera Manager metadata
Amazon S3 See Availability and durability guarantees.	Azure Disk Storage See Resiliency and disaster recovery protections.	Google Cloud Storage See Google Cloud Storage.	Ranger audits Component logs HMS /warehouse directories **Disposition: / Status:** Atlas lineage (HBase) (For AWS Data Lakes running Cloudera Runtime 7.2.7+). Atlas lineage data for Azure remains on Azure Managed Disks (see availability and durability below).

AWS	Azure	GCP	Component Storage Content
Amazon EBS See Availability and durability levels.	Azure Managed Disks See Availability and durability levels.	Google Persistent Disk See Persistent Disk.	**Disposition: / Status:** Atlas lineage (HBase) (For all Azure Data Lakes and AWS Data Lakes running Cloudera Runtime versions prior to 7.2.7) Atlas search index (Solr) Kafka data (to support Atlas) Zookeeper metadata

Data Lake storage is created when a Data Lake is instantiated for an environment. When an environment is no longer needed and is terminated, the corresponding Data Lake is terminated and the external storage is cleaned up.

Data Lake storage persists through a Data Lake repair cycle; new hosts created in the repair are re-attached to the storage locations.

> **Note:** Any S3 bucket that you designate for Data Lake cloud storage on AWS must be in the same region as the environment.

Disposition: / Status:

HBase Data Storage on AWS S3

Atlas stores its lineage data on HBase. Beginning with Cloudera Runtime 7.2.7, HBase data is stored on AWS S3 for Data Lakes in AWS environments. On Runtime versions prior to Runtime 7.2.7, the Data Lake cluster uses HDFS on Amazon EBS for storing Atlas data.

This means that as of Runtime 7.2.7, HDFS is no longer required for Atlas storage, with the exception of the HBase WAL file. The HDFS layer between the Atlas Index Search (Solr) and the local AWS EBS file system has also been removed. Solr data (the Atlas search index) remains on EBS directly, and not on HDFS on EBS.

> **Note:** HDFS has not been completely removed from the Data Lake cluster, as the HBase WAL directory is set on HDFS.

Has the core configuration service been implemented yet? Draft text below:

"The HDFS service has been replaced with the "Core configuration service," which provides standalone cluster-wide configuration. You cannot use both the HDFS service and Core configuration service in the same cluster, but the Data Lake cluster can use the core configuration service while workload clusters use the HDFS service."

This configuration is available only for new AWS environments using Runtime 7.2.7 or higher.

As far as I know, Data Lake backup and restore still cannot be done between two different versions of Runtime, nor is Data LAke upgrade GA (yet). When appropriate, we will add this note: "If you upgrade the Data Lake to 7.2.7 from a version older than 7.2.7, the HDFS storage configuration persists as it did in the older environment."

There are multiple benefits to storing Atlas data on S3:

• Storing Atlas data on S3 (instead of HDFS on top of EBS) results in significant cost savings for both light and medium duty Data Lakes.
• The HDFS file system requires manual tuning to increase or decrease storage, but cloud storage supports automatic resizing. The Data Lake storage will not get full, which can help you to avoid scheduled downtime.
• Removing HDFS reduces the Data Lake startup time.

Not yet implemented benefit: "Removing HDFS reduces CPU requirements in the Data Lake, which in turn reduces the EBS size for EC2 instances. Smaller EC2 instances mean greater cost savings."

S3 storage for Atlas data is automatically enabled when you create a new AWS environment with Runtime 7.2.7 or higher.

You can verify whether or not Atlas data is being stored on S3 by checking HBase configurations for the Data Lake in Cloudera Manager. Search for the parameter "HDFS Root Directory." It should point to an S3 link and not simply say "/hbase."

Data Lake scale

The scale of a Data Lake affects how many workload clusters can access your data using the security and governance services configured in the Data Lake, as well as resiliency of the Data Lake.

CDP supports both light duty Data Lakes and enterprise Data Lakes for AWS, Azure, and GCP. Medium duty and enterprise Data Lakes incur additional cost over light duty Data Lakes, but are required for production scenarios that require resiliency and scale. Medium duty and enterprise Data Lakes also have the ability to service a larger number of clients concurrently. See the sections below to understand the differences between light duty, medium duty and enterprise Data Lakes.

Important:

Enterprise Data Lakes are recommended for production workloads that require resiliency and scale.

Light duty Data Lakes do not provide resiliency. They are susceptible to data loss and down-time in case of a node failure. Cloudera recommends using this form factor for development and test use cases.

At this time, the following Data Lake scales are supported in CDP:

Feature	Light Duty Data Lake	Enterprise Data Lake	Medium Duty Data Lake (Discontinued as of Runtime 7.2.18)
High availability	Not available	✔	✔
Backups	✔	✔	✔
Availability Zones	Single availability zone	Multiple availability zones (AWS and Azure only)	Single availability zone
Security	Kerberos + LDAP/AD	Kerberos + LDAP/AD	Kerberos + LDAP/AD
Scale	About 5 concurrent workload clusters	About 20 concurrent workload clusters	About 20 concurrent workload clusters
Node count	1 master node running SDX Services 1 IDBroker node running networking authentication services	2 IDBroker nodes running authentication services 2 master nodes running core services 3 core nodes running HDFS, Kafka, Solr, and HBase 2 gateway nodes 1 auxiliary node	2 IDBroker nodes running authentication services 2 master nodes running core services in HA-enabled mode, with replication for resilience and scale 3 core nodes running HDFS, Kafka, Solr, and HBase 2 gateway nodes running services with API/UI access 1 auxiliary node for services that cannot run in HA mode
Fault tolerance	Services unavailable during cluster node repair	Availability of services depends on the node being repaired. With the exception of the gateway and auxiliary nodes, the remaining groups can typically survive a single node failure without affecting workloads or UI/API access. In the event of a gateway node failure on a medium duty Data Lake, the load-balancer will seamlessly route to the other gateway node. As Cloudera Manager runs on only one gateway node (either 0 or 1), if the Cloudera Manager server gateway node fails, CM will not be available at all, but UI and API calls that bypass CM will be routed to the healthy gateway node by the load balancer. If the non-CM server gateway node goes down, CM will still be available, and the load balancer will seamlessly route to the healthy gateway node.	Availability of services depends on the node being repaired. With the exception of the gateway and auxiliary nodes, the remaining groups can typically survive a single node failure without affecting workloads or UI/API access. In the event of a gateway node failure on a medium duty Data Lake, the load-balancer will seamlessly route to the other gateway node. As Cloudera Manager runs on only one gateway node (either 0 or 1), if the Cloudera Manager server gateway node fails, CM will not be available at all, but UI and API calls that bypass CM will be routed to the healthy gateway node by the load balancer. If the non-CM server gateway node goes down, CM will still be available, and the load balancer will seamlessly route to the healthy gateway node.
Cloud-based load balancer	Not applicable, since there is only one instance of services running.	Network-based load balancer for front UI and API services.	Network-based load balancer for front UI and API services.
Additional comments		Enterprise Data Lake is available for environments using Runtime 7.2.17 and newer.	Medium duty Data Lake has been discontinued as of Runtime 7.2.18.

Light Duty Data Lakes

Light Duty Data Lake Architecture

| Compute Engine Clients | UI Access Clients | API Access Clients |

Cloud-based load balancer

IDBroker Node

IDBroker

Master Node

Knox
Cloudera Manager
services

Ranger
Atlas
Kafka
HMS
Solr
HBase
HDFS
Zookeeper

If the master node of a light duty Data Lake fails, compute engine clients such as Hive, Impala, and Spark, are partially resilient due to caching; but new queries cannot run without updated policy information, and audit information can also be affected. Because the Knox gateway also runs on the master node, clients with UI (such as the Ranger Admin UI and Atlas UI) or API access are unavailable in the event of a master node failure. In a light-duty Data Lake, the cloud-based load balancer exists for networking purposes and has no effect on the scale.

If the IDBroker node fails, compute-engine clients are affected because cloud access tokens cannot be verified. Clients with UI/API access remain available.

Enterprise Data Lakes

Enterprise Data Lakes, available for Runtime 7.2.17 and newer, are a redefined version of medium duty Data Lakes that still offer failure resilience, but utilize resources and allocate memory more efficiently than a medium duty Data Lake at the same cost. Enterprise Data Lakes are configured such that services that do not need to scale are in the master hostgroup; services that need to scale vertically are in the gateway hostgroup; and services that can scale both horizontally and vertically are in the core hostgroup.

Enterprise Data Lakes can handle more intensive workloads than medium duty Data Lakes, support Ranger tag and user sync in HA mode, and when deployed in multi-AZ mode, remain operational during an availability zone outage.

Note: Multi-AZ deployments are currently only supported on AWS and Azure.

When compared to medium duty Data Lakes, RAZ will perform better and memory allocations for HBase, Solr, and Atlas have been increased. Other memory configurations have also been optimized.

Important: If you create an enterprise Data Lake through the CDP CLI, use the most recent version of the CLI. Older CLI versions do not support EDL creation.

Attention: The horizontal scale hostgroups noted in the diagram above contain a single service. Horizontal scaling is not yet supported, but enterprise Data Lakes are horizontal-scale ready for when the feature is available. When the feature is available, you will be able to use these hostgroups to horizontally scale these services as needed.

Medium Duty Data Lakes (Discontinued as of Runtime 7.2.18)
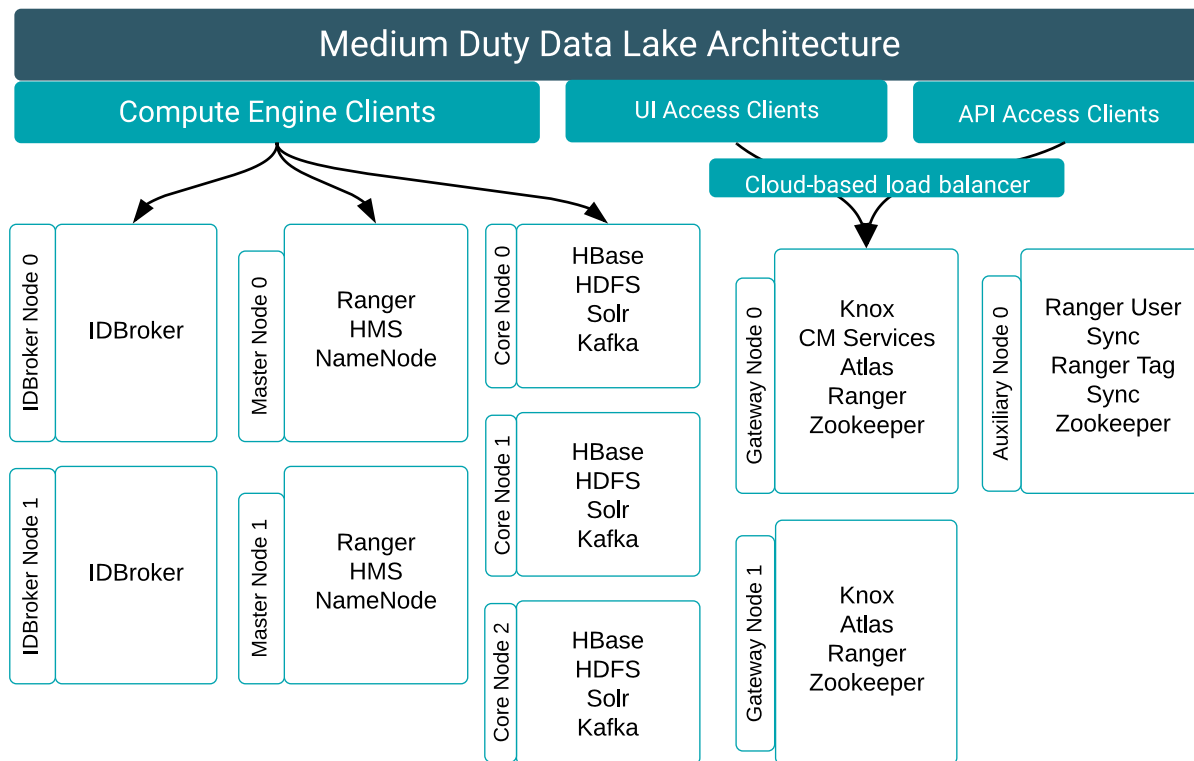
Medium duty Data Lakes provide failure resilience for compute engine clients such as Hive, Impala, and Spark; as well as failure resilience for clients with UI and API access, such as the Ranger Admin UI and the Atlas UI.

Note that while CM is shown as running on Gateway Node 0, it can be installed on either gateway node 0 or gateway node 1. You can see which node has CM installed by looking at the **Hardware** tab of the Data Lake for the gateway node marked "CM Server."

Failures in a medium duty Data Lake impact services as follows:

*   Master node failure. Compute engine clients (for example, Hive, Impala, and Spark) are resilient to the failure, due to fallback high availability with smart client failover.
*   IDBroker node failure. Both compute engine clients that use standard data connectors (Hive, Impala, Spark) and compute engine clients that use custom data connectors (for example, Hue) are resilient to the failure.
*   Gateway node failure. Load-balanced UI and API access are available without interruption.
*   Core node failure. Compute engine clients (for example, Hive, Impala, and Spark) are resilient to the failure, due to fallback high availability with smart client failover.
*   Auxiliary node failure. Ranger user and tag sync are unavailable.

Important: Medium duty Data Lakes have been discontinued as of Runtime 7.2.18. You can upgrade a medium duty Data Lake from 7.2.16 to 7.2.17, but will not be able to upgrade it further. You can create a new 7.2.17 medium duty Data Lake through the CDP CLI, but Cloudera recommends using the Enterprise Data Lake for new deployments.

If you want to scale an existing light or medium duty Data Lake to an enterprise Data Lake, you can perform Data Lake resizing.

Related Information
Azure Load Balancers in Data Lakes and Data Hubs
Deploying CDP in multiple availability zones

Data Lake repair

If a Data Lake node fails, an administrator can trigger a manual repair process to restore the failed node and reconnect it to the persistent Data Lake storage.

For each Data Lake cluster, CDP detects the following failures indicate that one or more nodes needs repair:

- The node is unresponsive, from a crash or termination
- The Cloudera Manager agent process is unresponsive

> **Note:** There may be failure modes that are not exposed at the level of the CDP Data Lake. If you encounter a service that is not responsive or is running but giving unexpected or incorrect results, no error will appear in the Data Lake details. Service-level errors should be managed through Cloudera Manager. For example, from Cloudera Manager, you can review the service's logs to determine the cause of the problem, make changes to the service configuration, and restart the service.

When CDP detects a node failure, a CDP administrator has the option to repair the failure manually. Note that during the repair process, the Data Lake services are not available to the attached workload clusters. Therefore, before triggering a Data Lake repair, consider stopping any jobs running on your workload clusters and restarting them after the Data Lake is restored. Audits and metadata will continue to be queued for collection through the restoration process.

When a node fails, you'll see a notification about node failure printed in the Event History tab for the Data Lake, the affected node is marked as unhealthy in the Hardware tab, and a button to start the repair process appears at the top of the Data Lake details. You can also select the Repair icon next to a host group on the Hardware tab to select specific nodes for repair. When your CDP administrator triggers node repair, the repair process:

1. Detaches all non-ephemeral disks from the failed nodes.
2. Removes the failed nodes.
3. Provisions new nodes of the same type, no upgrades are applied.
4. Reattaches the disks to the new volumes.
5. Reconnects services to the external database.

Related Information
Performing manual Data Lake repair
Data Lake storage
Cloudera Manager Health Tests
Cloudera Manager logs

Creating an AWS environment with a medium duty data lake using the CLI

You can use the CDP CLI to create an AWS environment with a medium duty data lake.

About this task
Required role: EnvironmentCreator

Before you begin

Before you use the CDP CLI, run the following command to verify that your environment is pointing to the correct profile:

```
cdp --profile {PROFILE}
```

Run the following command to verify that your environment name is not already taken:

```
environments describe-environment --environment-name {ENVNAME}
```

Procedure

1. Create a new environment:

```
cdp environments create-aws-environment --cli-input-json file://{ENV_FIL
E_PATH}
```

2. To set the IDBroker mappings, run the following command:

```
cdp environments set-id-broker-mappings --environment-name "$ENVNAME" --
data-access-role "$DATAACCESSROLE" --baseline-role "$BASELINEROLE" --set-
empty-mappings
```

3. Run the following command to create the data lake cluster within the environment, where INSTANCEPROFILE is the instance profile for your specific account, and BUCKET is the path of a valid S3 location to store the data. This S3 path can be either the root of a bucket or a sub-folder:

```
cdp datalake create-aws-datalake --datalake-name "NAME" --environment-name
 "ENVNAME" --cloud-provider-configuration instanceProfile="INSTANCEPROFILE
",storageBucketLocation="s3://MYBUCKET" --scale MEDIUM_DUTY_HA  --runtime
 7.2.7
```

4. Run the following command to check the status of the Data Lake:

```
cdp datalake list-datalakes --environment-name ${ENVNAME}
```

You should be able to look at the list of data lakes, locate yours by ENVNAME and check the status.

Understanding Data Lake details

To access information related to your Data Lake cluster and access cluster actions, navigate to the Management Console service > Data Lakes.

Each Data Lake cluster is represented by an entry on the Data Lakes page. To get more information about a specific Data Lake cluster, click on the tile representing your cluster.

Environment Details

This section includes information related to the CDP environment in which the Data Lake cluster is running:

Item	Description
Cloud Provider	The logo of the cloud provider where the cluster is running.
Name	The name of the environment used to create the cluster.
Credential	The name of the credential used to create the cluster.
Region	The region in which the cluster is running in the cloud provider infrastructure.
Availability Zone	The availability zone within the region in which the cluster is running.

Services

Click logos in the Services section to open the user interface for the components that are running in the Data Lake cluster.
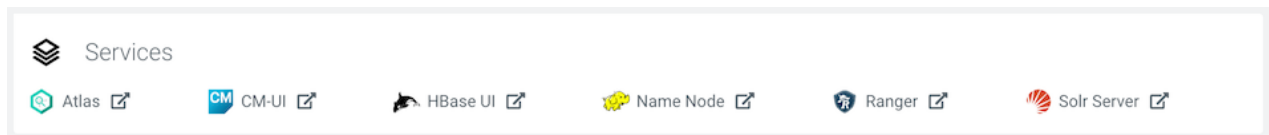
Cloudera Manager Info

The Cloudera Manager Info section provides the following information:

Item	Description
CM URL	Link to the Cloudera Manager web UI.
CM Version	The Cloudera Manager version which the cluster is currently running.
Platform Version	The Cloudera Runtime platform version which the cluster is currently running.

Event History and other tabs

The Data Lake page provides additional details organized in tabs, starting with the Event History tab:

Item	Description
Event History	Events logged for the cluster, with the most recent event at the top. The Download option allows you to download the event history to a local file. The events are formatted in JSON and compressed.

Item	Description
Hardware	Information about your cluster instances: instance names, instance IDs, instance types, their status, fully qualified domain names (FQDNs), and private and public IP addresses. Click ≫ to access information about the instance, storage, image, and packages installed on the image.
Cloud Storage	External storage locations for database and files used by Data Lake services, such as HMS database, Ranger audit database, HBase files (storage for Atlas metadata).
Tags	User-defined tags, listed in the order they were added.
Endpoints	Endpoints for various cluster services, such as the URL for the Ranger user interface for defining data access policies.
Recipes	Future home of a list of custom scripts attached to this Data Lake. Each "recipe" lists its name, type, and the host group on which it was executed.
Attached clusters	The workload clusters using the services of this Data Lake; this information repeats the list of clusters found in the other tabs of this CDP Environment.
Repository Details	Cloudera Manager and Cloudera Runtime repository information, in more detail than shown in the Cloudera Manager Info section.
Image Details	Cluster node base image details.
Network	Names of the network and subnet in which the cluster is running and the links to the related cloud provider console
Telemetry	The instance profile and cloud storage location specified during environment setup under Log Storage and Audits for service logs.

Show CLI Command

You can click the Show CLI Command button to review the CLI command used to create the Data Lake, and copy it if you want to create a similar Data Lake through the CDP CLI. Ensure that any Data Lake that you create has a unique name. For more information on the CDP CLI commands to create a Data Lake, review the CLI documentation for Data Lakes.

Accessing Data Lake services

You can access your Data Lake security and governance services such as Atlas and Ranger from the Endpoints tab from Data Lake details in the Management Console.

Required role: EnvironmentAdmin, Data Steward, or Owner of the environment

To access data lake UIs and endpoints navigate to the Management Console > Data Lakes and click on the tile representing your Data Lake. This brings you to the Data Lake cluster details page:

- The URLs to data lake service UIs are listed directly on this page, in the Services section. Click on the URL for the service that you would like to access and you will be logged in automatically with your CDP credentials. All the UIs are accessible via the Knox gateway: The URLs listed connect you to a chosen service via Knox and Knox securely passes your CDP credentials.
- To access API endpoints, navigate to the Endpoints tab. If you need to access the endpoints, refer to Accessing Non-SSO Interfaces Using IPA Credentials.
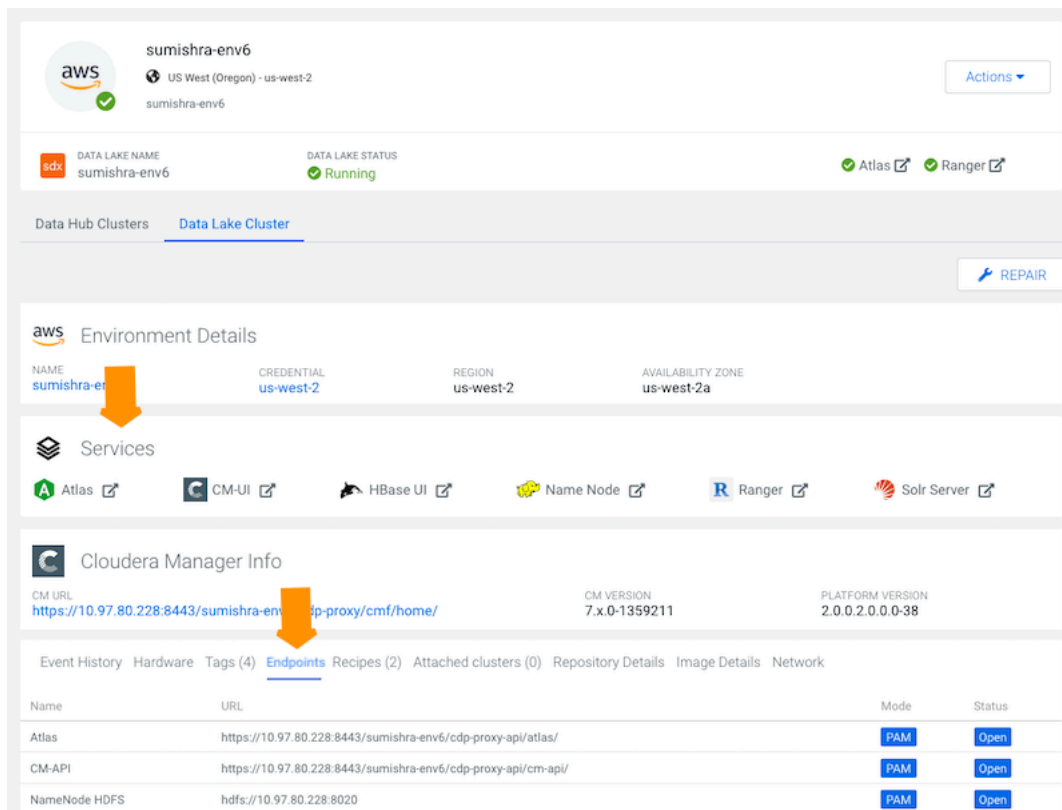
Security exception

The first time you access the UIs, your browser will attempt to confirm that the SSL Certificate is valid. Since CDP automatically generates a certificate with self-signed CA, your browser will warn you about an untrusted connection and ask you to confirm a security exception. Depending on your browser, perform the steps below to proceed:

Browser	Steps
Firefox	Click Advanced > Click Add Exception… > Click Confirm Security Exception
Safari	Click Continue
Chrome	Click Advanced > Click Proceed…

You can also view your available data lakes via CDP CLI using the following commands:

```
cdp datalake list-datalakes
cdp datalake describe-datalake
cdp datalake get-cluster-host-status
cdp datalake get-cluster-service-status
```

Related Information

Apache Ranger authorization

Audit overview

Governance overview

Accessing a Data Lake cluster via SSH

If you plan to access a Data Lake (for example for troubleshooting purposes) via a command line client, SSH into the master node.

SSH to a Data Lake node as your own CDP user

> **Disposition: / Status:**
> CB-16100 New EnvironmentPriviledgedUser role. Updated the required role. Previously this said that he required roles were: EnvironmentUser, Data Steward, or EnvironmentAdmin. Please review this whole page.

Required role: EnvironmentPriviledgedUser

A user who have the required CDP role assigned to them can SSH to Data Lake cluster nodes using their own CDP user with their own private SSH key that matches the public key previously provided in CDP.

On Mac OS, you can use the following syntax to SSH to the VM:

```
ssh -i <path-to-private-key-file> <cdpusername>@<nodeIPaddress>
```

For example:

```
ssh jsmith@90.101.0.132
```

On Windows, you can access your cluster via SSH by using an SSH client such as PuTTY. For more information, refer to How to use PuTTY on Windows.

SSH to a Data Lake node as cloudbreak user

Required role: No CDP role is required

CDP administrators can access Data Lake cluster nodes as cloudbreak user with the SSH key provided during cluster creation.

On Mac OS, you can use the following syntax to SSH to the VM:

```
ssh -i <path-to-cloudbreak-private-key-file> cloudbreak@<nodeIPaddress>
```

For example:

```
ssh -i ~/.ssh/cloudbreak-private-key cloudbreak@90.101.0.132
```

On Windows, you can access your cluster via SSH by using an SSH client such as PuTTY. For more information, refer to How to use PuTTY on Windows.

Administering a Data Lake

To manage authorization and audit policies, and metadata use Apache Ranger and Apache Atlas.

Required role: Environment Admin, DataSteward, or the Owner of the environment

Close integration of Atlas with Apache Ranger enables you to define, administer, and manage security and compliance policies consistently across all components of the Hadoop stack.

Documentation	Description
Security: Apache Ranger authorization	How to set up fine grained access control for Cloudera Runtime services.

Documentation	Description
Security: Apache Ranger auditing	How to set up access auditing and reporting for Cloudera Runtime services.
Governance: Apache Atlas	Searching with metadata, working with classifications, exploring using lineage, and more.

For links to documentation, refer to:

Related Information
Security documentation
Governance documentation

Monitoring a Data Lake

You can monitor the status of your Data Lake from the CDP web UI or CLI.

Required role: EnvironmentAdmin, Data Steward, or Owner of the environment

Monitoring Data Lake cluster via UI

To access information related to your Data Lake cluster from the CDP web UI, navigate to the Management Console service > Data Lakes. Each Data Lake cluster is represented by an entry on the Data Lakes page. To get more information about a specific Data Lake cluster, click on the tile representing your cluster. When a Data Lake cluster is healthy, its status should be Running.

To check health of specific hosts and services, navigate to Cloudera Manager.

Monitoring Data Lake cluster via CLI

You can view your available Data Lake clusters via CDP CLI using the following commands:

```
cdp datalake list-datalakes
cdp datalake describe-datalake
cdp datalake get-cluster-host-status
cdp datalake get-cluster-service-status
```

The cdp datalake list-datalakes command allows you to view a list of all available Data Lakes. For example:

```
cdp environments list-datalakes
{
    "datalakes": [
        {
            "datalakeName": "zookeeper-190920-144828-vg7",
            "crn": "crn:cdp:datalake:us-west-1:9d74eee4-1cad-45d7-b645-7cc
f9edbb73d:datalake:4529591f-53ea-4196-90fc-5d780d7063a8",
            "status": "RUNNING",
            "environmentCrn": "crn:cdp:environments:us-west-1:9d74eee4-1c
ad-45d7-b645-7ccf9edbb73d:environment:b1935d68-85d5-4f50-a023-56fa96d01c45",
            "creationDate": "2019-09-20T12:49:55.669000+00:00",
            "statusReason": "Datalake is running"
        },
        {
            "datalakeName": "zookeeper-sqqsx",
            "crn": "crn:cdp:datalake:us-west-1:9d74eee4-1cad-45d7-b645-7c
cf9edbb73d:datalake:92d66fed-c5d2-437c-a6eb-a54e40d36287",
            "status": "RUNNING",
            "environmentCrn": "crn:cdp:environments:us-west-1:9d74eee4-1cad-
45d7-b645-7ccf9edbb73d:environment:1eb291b3-dd23-4bdd-a3e8-09579afdf5a8",
            "creationDate": "2019-09-25T09:24:08.017000+00:00",
```

```
                        "statusReason": "Datalake is running"
                }
        ]
}
```

The cdp datalake describe-datalake command allows you to obtain basic information about a specific Data Lake cluster. For example:

```
cdp datalake describe-datalake --datalake-name test-data-lake
{
    "datalake": {
        "crn": "crn:cdp:datalake:us-west-1:9d74eee4-1cad-45d7-b645-7ccf9ed
bb73d:datalake:aa2e8e3e-2d6f-410b-bf3c-a3e02112bfc8",
        "datalakeName": "test-data-lake",
        "status": "RUNNING",
        "environmentCrn": "crn:cdp:environments:us-west-1:9d74eee4-1cad-45d
7-b645-7ccf9edbb73d:environment:574aa1cb-7a51-45a2-97ae-dead97072145",
        "credentialCrn": "crn:altus:environments:us-west-1:9d74eee4-1cad-
45d7-b645-7ccf9edbb73d:credential:83c861b6-5f62-4b83-a466-06de751a3964",
        "cloudPlatform": "AWS",
        "creationDate": "2019-09-20T22:09:22.422000+00:00",
        "clouderaManager": {
            "version": "7.x.0",
            "clouderaManagerRepositoryURL": "http://cloudera-build-us-west
-1.vpc.cloudera.com/s3/build/1445641/cm7/7.0.1/redhat7/yum/",
            "clouderaManagerServerURL": "https://adar-test-data-lake.adar-
tes.xcu2-8y8x.workload-dev.cloudera.com:8443/test-data-lake/cdp-proxy/cmf/ho
me/"
        },
        "productVersions": [
            {
                "name": "CDH",
                "version": "7.0.1-1.cdh7.0.1.p0.1443705"
            }
        ],
        "statusReason": "Datalake is running",
        "awsConfiguration": {
            "instanceProfile": "arn:aws:iam::069336058373:instance-profile/
idbroker-assume-role"
        }
    }
}
```

The cdp datalake get-cluster-host-status command allows you to obtain information about the health of each of your Data Lake hosts. For example:

```
cdp datalake get-cluster-host-status --cluster-name test-data-lake
{
    "hosts": [
        {
            "hostid": "5c8fb276620f0aa54bdd111e33ba5f58",
            "hostname": "idbroker1.cloudera.site",
            "healthSummary": "GOOD"
        },
        {
            "hostid": "30f27ab8472c9677985f04efc2b800c4",
            "hostname": "master0.cloudera.site",
            "healthSummary": "GOOD"
        }
    ]
}
```

The cdp datalake get-cluster-service-status command allows you to obtain information about the health of each service running on the Data Lake cluster. For example:

```
cdp datalake get-cluster-service-status --cluster-name test-data-lake
{
    "services": [
        {
            "type": "ZOOKEEPER",
            "state": "STARTED",
            "healthSummary": "GOOD",
            "healthChecks": [
                {
                    "name": "ZOOKEEPER_SERVERS_HEALTHY",
                    "summary": "GOOD"
                }
            ]
        },
        {
            "type": "HDFS",
            "state": "STARTED",
            "healthSummary": "GOOD",
            "healthChecks": [
                {
                    "name": "HDFS_DATA_NODES_HEALTHY",
                    "summary": "GOOD"
                },
                {
                    "name": "HDFS_VERIFY_EC_WITH_TOPOLOGY",
                    "summary": "DISABLED"
                }
            ]
        },
        {
            "type": "SOLR",
            "state": "STARTED",
            "healthSummary": "GOOD",
            "healthChecks": [
                {
                    "name": "SOLR_SOLR_SERVERS_HEALTHY",
                    "summary": "GOOD"
                }
            ]
        },
        {
            "type": "HIVE",
            "state": "STARTED",
            "healthSummary": "GOOD",
            "healthChecks": [
                {
                    "name": "HIVE_HIVEMETASTORES_HEALTHY",
                    "summary": "GOOD"
                }
            ]
        },
        {
            "type": "RANGER",
            "state": "STARTED",
            "healthSummary": "GOOD",
            "healthChecks": [
                {
                    "name": "RANGER_RANGER_ADMIN_HEALTHY",
                    "summary": "GOOD"
                },
```

```
                {
                    "name": "RANGER_RANGER_RANGER_TAGSYNC_HEALTH",
                    "summary": "GOOD"
                },
                {
                    "name": "RANGER_RANGER_RANGER_USERSYNC_HEALTH",
                    "summary": "GOOD"
                }
            ]
        },
        {
            "type": "HBASE",
            "state": "STARTED",
            "healthSummary": "GOOD",
            "healthChecks": [
                {
                    "name": "HBASE_REGION_SERVERS_HEALTHY",
                    "summary": "GOOD"
                }
            ]
        },
        {
            "type": "KAFKA",
            "state": "STARTED",
            "healthSummary": "GOOD",
            "healthChecks": [
                {
                    "name": "KAFKA_KAFKA_BROKER_HEALTHY",
                    "summary": "GOOD"
                }
            ]
        },
        {
            "type": "ATLAS",
            "state": "STARTED",
            "healthSummary": "GOOD",
            "healthChecks": [
                {
                    "name": "ATLAS_ATLAS_SERVER_HEALTHY",
                    "summary": "GOOD"
                }
            ]
        },
        {
            "type": "KNOX",
            "state": "STARTED",
            "healthSummary": "GOOD",
            "healthChecks": [
                {
                    "name": "KNOX_IDBROKER_HEALTHY",
                    "summary": "GOOD"
                },
                {
                    "name": "KNOX_KNOX_GATEWAY_HEALTHY",
                    "summary": "GOOD"
                }
            ]
        }
    ]
}
```

Retry a Data Lake

When stack provisioning or cluster creation fails, use the Retry option to resume the process from the last failed step.

About this task

Required role: EnvironmentAdmin or the Owner of the environment

Only failed stack or Data Lake creation can be resolved using a retry operation. You can potentially run a retry operation any number of times on a failed creation process, where each time it runs it resumes the creation process after the last successful step.

In some cases the cause of a failed stack provisioning or Data Lake creation may be eliminated by simply rerunning the process. For example, in case of a temporary network outage, a retry operation may be successful. In other cases, a manual modification is required before a retry operation can succeed. For example, if you are using a custom image but some configuration is missing causing the creation process to fail, you must log in to the provisioned node and fix the issue; after that you can run the retry operation to resume the Data Lake creation process.

Procedure

1. Log in to the CDP web interface.
2. Navigate to Management ConsoleData Lakes.
3. Browse to the Data Lake details.
4. Click Retry.

   Note:

   Only failed stack or Data Lake creation are affected by a retry operation, so the option is only available in these cases.

5. Click Yes to confirm. The operation continues from the last failed step.

Upgrading Data Lake/Data Hub database

This document describes the process to upgrade the database to the latest version supported by CDP Public Cloud services. You may use CDP UI or CDP CLI to perform this upgrade.

Disposition: / Status:
This topic is in both Data Hub and Management Console (Data Lake) doc, so if you update the topic cherry-pick the changes to both dh-cloud and mc-cloud.

About this task

Several CDP Public Cloud services, including the Data Lake cluster and the Data Hub cluster templates and Data Services, require a relational database. Most of these databases are external and are provisioned during the initial deployment of the respective service.

The databases used by the Data Lake and some of the Data Hub templates are hosted on external instances that are provisioned during the initial deployment of the respective service. For these external databases CDP Public Cloud leverages cloud-native service offerings of the three supported Cloud Service Providers (AWS RDS for PostgreSQL, Azure Database for PostgreSQL and Cloud SQL for PostgreSQL).

Databases used by other Data Hub templates are hosted on an embedded database instance, typically co-located on the Cloudera Manager host, in order to reduce the resource footprint.

Cloudera provides a database upgrade capability in CDP Public Cloud that allows moving both external and embedded databases to a higher major version.

The database upgrade is a fully automated operation. The upgrade process itself completes all of the required steps, including creating a backup, stopping and upgrading the database, restarting the database, and running post-upgrade maintenance tasks. You are not required to manually stop the Postgres instances before the upgrade.

Attention: In accordance with the PostgreSQL Versioning Policy, the cloud database services mentioned above may end support for PostgreSQL major version 11 on November 9, 2023 or shortly thereafter. Different cloud providers may have extended support for PostgreSQL 11. Despite this, Cloudera recommends upgrading to PostgreSQL 14 when the upgrade is available to you in CDP. Currently, upgrading the Data Lake or Data Hub database to PostgreSQL 14 is available for clusters on AWS and GCP, with Azure upgrades to PostgreSQL 14 available soon. Azure databases that require an upgrade will be upgraded from PostgreSQL 10 to 11.

Important: In order to avoid disruption to the deployed Data Lake and Data Hub services, caused by configuration changes to the underlying database service by the Cloud Service Providers, it is recommended that the database upgrade in CDP Public Cloud is performed before the End of Life date.

If you wish to disregard this recommendation, you may do so considering the risks involved as per the Cloud Service Provider policies.
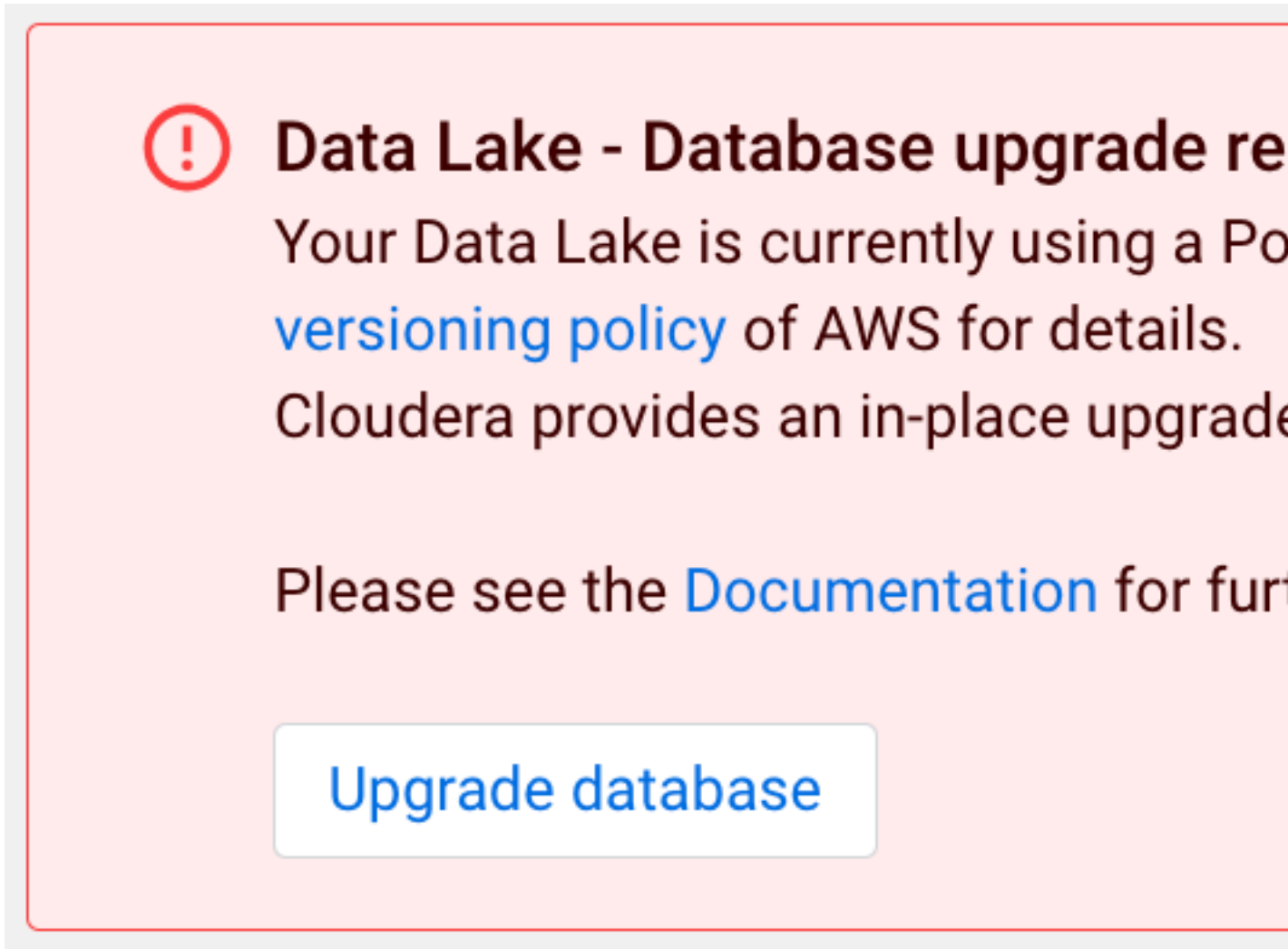
The database upgrade is a separate operation, complementary to the existing maintenance, minor/major version and OS upgrades, as described in the CDP Public Cloud Upgrade Advisor.

This is a one-time operation. Once the database of a Data Lake or Data Hub has been successfully upgraded to the newer major version, no further action is needed for the respective cluster.

Note: Cloudera recommends that the database upgrade is performed separately from other upgrade actions.

If a cluster uses a database that requires an upgrade, you will receive a notification, as shown below, on the Management Console UI.

> **Note:** Clusters running Cloudera Runtime version 7.2.6 or lower need to be upgraded to a more recent runtime version before they are eligible for a database upgrade.

Running the database upgrade operation on the Data Hub cluster will mean that all cluster services (Cloudera Manager and Rutime services) are stopped on the cluster automatically without having to stop them manually. For the Data Lake database upgrade, it is recommended that attached Data Hubs and Data services are in stopped state.

> **Note:** Cloudera strongly recommends stopping all workloads in Data Services that interact with the Data Lake.
>
> If you are concerned about stopping the workloads in your deployment, contact Cloudera support for a custom upgrade path.

For AWS and GCP environments, the Database Upgrade operation will trigger a backup and a major version upgrade for the attached external database. But for Azure environments, the mechanism is different; as in the background, it will create a new database instance with a higher major version and transfer the data from the older database instance.

> **Note:** During Postgres database upgrade for Data Lakes and Data Hubs on AWS and Azure, there is a possibility that manually changed configs of the database server will be reverted to the original configs. For more information, see Database upgrade known limitations.

Instructions

Here are the UI and CLI instructions to perform Database Upgrade on Data Lake and Data Hub:

For CDP UI

Steps

1. In CDP Management Console UI, go to Environments. Select the cluster to perform the upgrade from the list of available clusters. The clusters are eligible for this upgrade are indicated in the right most column:
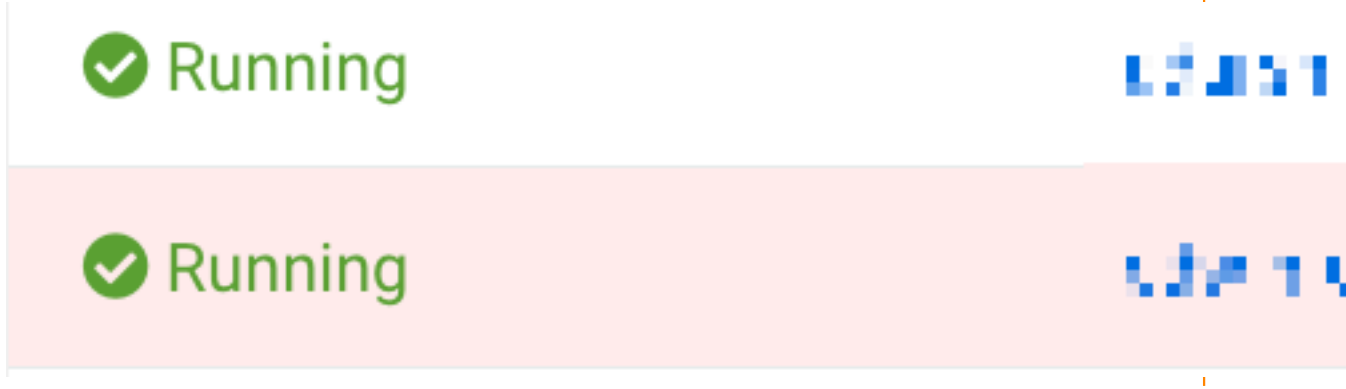
2. Once you select the cluster, you will see a message asking to update the Postgres version. Click the Upgrade database.

Environments / ▚▚▙▜ ▐▌▚▚▚

(!) **Data Lake - Database upgrad**

Your Data Lake is currently using a

versioning policy of AWS for detai

Cloudera provides an in-place upg

Please see the Documentation for

Upgrade database

aws ✓

▚▚▙▐▌▚▚▚ ◯
▚▚▐▌▚▚▚▐▌▚

🌐 US West (Oregon) - us-w

Data Lake upgrade availab

3. Click Upgrade in the confirmation box.

Environments / ⬛⬛⬛⬛⬛⬛⬛

⚠ **Data Lake - Database upgrade**

Your Data Lake is currently using a

versioning policy of AWS for detail

Cloudera provides an in-place upg

Please see the Documentation for

Upgrade database

aws ✓

⬛⬛⬛⬛⬛⬛⬛

⬛⬛⬛⬛⬛⬛⬛

🌐 US West (Oregon) - us-we

Data Lake upgrad

Up

4. Once the Data Lake database is updated, check for the Data Hubs for that Data Lake, if there is any database upgrade notification and perform the database upgrade as described above.

> **Note:** The Database upgrade needs to be performed in every Data Lake and Data Hub cluster separately, one by one.

For CDP CLI

Data Lake Database upgrade:

You can perform Data Lake database upgrade using cdp datalake   start-database-upgrade CLI command.

The --target-version parameter is optional. If you do not provide it, the database will be upgraded to either PostgreSQL 14 (AWS and GCP) or Postgres 11 (Azure). You can also use the VERSION_11 value if you specifically want to upgrade to PostgreSQL 11.

```
cdp datalake start-database-upgrade --help --form-factor public
NAME
       start-database-upgrade  -  Upgrades the database of the Data Lake
 clus-
       ter.
DESCRIPTION
       This command initiates the upgrade of the database  of  the  Data
 Lake
       cluster.
SYNOPSIS
           start-database-upgrade
          --datalake <value>
          --target-version <value>
          [--cli-input-json <value>]
          [--generate-cli-skeleton]

OPTIONS
       --datalake (string)
          The name or CRN of the Data Lake.

       --target-version (string)
          The database engine major version to upgrade to.

          Possible values:

          o VERSION_11
```

Data Hub Database upgrade:

You can perform Data Hub database upgrade using cdp datahub   start-database-upgrade CLI command.

The --target-version parameter is optional. If you do not provide it, the database will be upgraded to either PostgreSQL 14 (AWS and GCP) or Postgres 11 (Azure). You can also use the VERSION_11 value if you specifically want to upgrade to PostgreSQL 11.

```
cdp datahub start-database-upgrade --help --form-factor public
NAME
       start-datahub-upgrade  -  Upgrades the database of the Data Hub c
lus-
       ter.

DESCRIPTION
       This command initiates the upgrade of the database  of  the  Data
Hub
       cluster.
SYNOPSIS
           start-database-upgrade
```

```
            --datahub <value>
            --target-version <value>
            [--cli-input-json <value>]
            [--generate-cli-skeleton]

OPTIONS
        --datahub (string)
           The name or CRN of the Data Hub.

        --target-version (string)
           The database engine major version to upgrade to.
           Possible values:
           o VERSION_11
```

The progress of the upgrade can be tracked on the respective service's Event History page. You can verify a successful database upgrade in the Event History or in the Database tab of the cluster. Once the upgrade is complete, Cloudera recommends verifying your workloads before attempting an additional Runtime or OS upgrade.

Note: As part of the database upgrade operation, PostgreSQL 11 or 14 client binaries will be installed on the cluster hosts, replacing earlier client versions. This may impact third-party components or custom services running on the cluster hosts.

Database upgrade known limitations and troubleshooting

Below are the known limitations associated with the database upgrade of Data Lake and Data Hubs and ways to troubleshoot them.

Disposition: / Status:
This topic is in both Data Hub and Management Console (Data Lake) doc, so if you update the topic cherry-pick the changes to both dh-cloud and mc-cloud.

Known limitations and troubleshooting:

• Performing the Database Upgrade on Runtime versions 7.2.6 or below

Cloudera has verified PostgreSQL version 11 compatibility for Runtime version 7.2.7 and above. There is no known reason why older Runtimes should not be compatible with PostgreSQL version 10.

Workaround: You can request an entitlement that allows the Database Upgrade to be performed on older Runtime versions on an exceptional basis.

• Performing the Database Upgrade on Data Lakes with attached Data Hubs that cannot be stopped

Technically, the Database Upgrade can be performed on a Data Lake without stopping the attached Data Hubs. However, please be aware that during the upgrade, the Hive Metastore database will likely become temporarily unavailable and this can cause serious disruption or in the worst case can result in an inconsistent state for workloads running in Data Hubs or Data Services.

Workaround: If you acknowledge the risk and confirm that all cluster services and third party components relying on the Hive Metastore will be stopped for the time of the Database Upgrade, Cloudera can grant an entitlement that allows performing the upgrade with a running Data Hub cluster on an exceptional basis.

- PostgreSQL client binaries will upgraded to version 11 on all clusters hosts

  As part of the upgrade process we will try to install the PostgreSQL 11 libraries, pulling them from archive.cloudera.com. If the installation of these libraries does not succeed, a notification message will be sent that installation was attempted, but failed for some reason (network connectivity issues, etc).

  Workaround: Follow the process to install the libraries manually, see Installing PostergeSQL 11 packages manually.

  > **Note:** Failing to install the PostgreSQL 11 client libraries as part of the Database Upgrade process will cause the Data Lake backup and restore operations to stop working correctly.

- Upgrading embedded databases

  Data Hub clusters using an embedded database will not require the Database Upgrade operation to be performed. The embedded database, including client libraries will be automatically upgraded during an OS upgrade.

  > **Note:** This capability is currently disabled and will be activated later.

  Workaround: If you need to upgrade the embedded databases of your Data Hub clusters, contact Cloudera to enable this capability on an exceptional basis. Once this entitlement has been granted, your embedded databases can be upgraded by performing an OS upgrade.

- Exceeding the End of Life deadline

  Data Lake and Data Hub clusters that are not upgraded until November 10, 2022 will continue to run on a PostgreSQL version 10 instance of the underlying AWS, Azure or GCP database service. As this instance will be considered End-of-Life (EoL) by the respective Cloud Service provider, they may reserve the right to schedule an automated major version upgrade, resulting in a temporary downtime. In the case of extreme events the Cloud Service Provider may also stop the instance, see Versioning policy- Azure Database for PostgreSQL. In either case, your CDP Public Cloud workloads may be seriously impacted.

  Workaround: Cloudera recommends performing the Database Upgrade via the CDP UI, or CLI as soon as possible.

- Possibility of custom config reset after Database upgrade on AWS and Azure

  During Postgres database upgrade for Data Lakes and Data Hubs there is a possibility that manually changed configs of the database server, that the control plane does not know about, will be reverted to the original configs.

  Reason: On Azure the custom config can possibly reset during the database upgrade because Cloudbreak deletes and recreates the database server with the configs that the control plane knows about, so custom configs will be reverted.

  On AWS if SSL enforcement is enabled then the database server uses a custom parameter group with the SSL enforcement settings (created by control plane) and if the customer made any custom changes to this custom parameter group then those changed will be reverted, because the database upgrade requires the recreation of the custom parameter group.

Installing Postgres 14 packages manually

Steps for manual installation of PostgreSQL 14 packages.

Disposition: / Status:
This topic is in both Data Hub and Management Console (Data Lake) doc, so if you update the topic cherry-pick the changes to both dh-cloud and mc-cloud.

About this task

The last step of the Database upgrade flow is the installation of PostgreSQL 14 packages on the cluster hosts. This is relevant in the case of an operating system image that does not yet contain the PostgreSQL 14 packages.

The required repositories are being hosted in the same location that is used for Cloudera Runtime upgrades: https:// archive.cloudera.com/p/postgresql/postgresql14/redhat7/

If for some reason the package installation fails, it is required for the customers to manually install the aforementioned packages because otherwise the pg_dump utility driving the backup functionality will stop working.

- Method 1 : Installation using Cloudera hosted

  This method works only if you have proper network access and paywall credentials to the archive.cloudera.com repository as the required metadata is already pushed onto the nodes during the RDS upgrade process.

  SSH into the master node and run the following with superuser privileges:

  ```
  {code}
  source activate_salt_env
  salt '*' state.apply postgresql/pg14-install
  {code}
  ```

- Method 2: Manual installation

  Using this method you will install PostgreSQL packages using the official repo file

  1. SSH into the master node and run the following with superuser privileges (install PostgreSQL packages using the official repo file)

     ```
     yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL
     -7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
     ```

  2. Install required packages

     ```
     yum install -y postgresql14-server postgresql14 postgresql14-contrib pos
     tgresql14-docs
     ```

Installing Postgres 11 packages manually

Steps for manual installation of PostgreSQL 11 packages.

Disposition: / Status:
This topic is in both Data Hub and Management Console (Data Lake) doc, so if you update the topic cherry-pick the changes to both dh-cloud and mc-cloud.

About this task

The last step of the Database upgrade flow is the installation of PostgreSQL 11 packages on the cluster hosts. This is relevant in the case of an operating system image that does not yet contain the PostgreSQL 11 packages.

The required repositories are being hosted in the same location that is used for Cloudera Runtime upgrades: https:// archive.cloudera.com/p/postgresql/11/redhat7/

If for some reason the package installation fails, it is required for the customers to manually install the aforementioned packages because otherwise the pg_dump utility driving the backup functionality will stop working.

- Method 1 : Installation using Cloudera hosted

  This method works only if you have proper network access and paywall credentials to the archive.cloudera.com repository as the required metadata is already push onto the nodes during the RDS upgrade process.

  SSH into the master node and run the following with superuser privileges.

  ```
  source activate_salt_env
  salt '*' state.apply postgresql/pg11-install
  ```

- Method 2: Manual installation

  Using this method you will install PostgreSQL packages using the official repo file

  1. SSH into the master node and run the following with superuser privileges (install PostgreSQL packages using the official repo file)

  ```
  yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL
  -7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
  ```

  2. Install required packages

  ```
  yum install -y postgresql11-server postgresql11 postgresql11-contrib pos
  tgresql11-docs
  ```

Data Lake upgrade

When new versions or builds of Cloudera Runtime/Cloudera Manager are available for the Data Lake service, you can initiate a Data Lake upgrade. An OS upgrade may also be available. Use either the CDP CLI or the Management Console to initiate an upgrade.

There are several potential types of Data Lake upgrades:

- Runtime and Cloudera Manager version upgrades, called "major/minor version" upgrades, which are available when a new Runtime and CM version is released.
- Service pack upgrades, which do not change the major/minor Runtime or CM release, but upgrade the Data Lake to the latest CM and/or Runtime service packs available in the given version. These upgrades are made available as needed and can deliver targeted bug fixes for Cloudera Runtime, Cloudera Manager, or both.
- OS upgrades, which do not change any CM or Runtime builds, but update the underlying cloud image. If an OS upgrade is available along with a major/minor version upgrade, the major/minor upgrade will incorporate the OS upgrade.

The type of upgrade that is available for a Data Lake is explicit on the Upgrade tab of the Data Lake details page. Any available upgrades are visible in the Target Cloudera Runtime Version drop-down menu. For example:

A major/minor version upgrade, where the major/minor versions of CM and Runtime are upgraded together:

Disposition: / Status:
CB-24372 RHEL upgrade - Screenshots on this page have been updated.

Upgrade Data Lake ⟳

Current Data Lake Version: 7.2.15

> ⓘ A new upgrade is available for this Data Lake. You can continue through the process below.
> Cloudera Runtime 7.2.18 is only available with Red Hat Enterprise Linux 8 OS images. Make sure to upgrade your cluster to RHEL 8 to unlock the upgrade to 7.2.18.

Select Upgrade

Target Cloudera Runtime Version

[7.2.17 (Runtime upgrade, OS: centos7) ▾]

☐ Perform rolling upgrade

Rolling upgrade is not supported for this cluster, but is enabled via entitlement. Some cluster services might become unavailable during upgrade and running workloads could be impacted. For limitations and details please see the rolling upgrade documentation.

The Cloudera Runtime version of your cluster will be upgraded to 7.2.17 and an OS upgrade will also be performed. See the Public Cloud Runtime release notes for the list of changes and fixes in this version. This upgrade requires a downtime. Further details are available in the CDP Public Cloud upgrade advisor and the documentation for Upgrading Data Hubs.

Version details

	Image Date	CM Version	OS Type	CM Build number	Cloudera Runtime Version	Cloudera Runtime Build number
Current	2023-11-29	7.6.2	centos7	42790603	7.2.15	47111845
Target	2024-02-02	7.11.0	centos7	48415235	7.2.17	49883770

A service pack upgrade, where no version upgrade is available, but a new Cloudera Runtime and/or Cloudera Manager service pack is available for upgrade:

Upgrade Data Lake ⟳

Current Data Lake Version: 7.2.17

> ⓘ A new upgrade is available for this Data Lake. You can continue through the process below.
> Cloudera Runtime 7.2.18 is only available with Red Hat Enterprise Linux 8 OS images. Make sure to upgrade your cluster to RHEL 8 to unlock the upgrade to 7.2.18.

Select Upgrade

Target Cloudera Runtime Version

[7.2.17 (Runtime upgrade, OS: centos7) ▾]

☐ Perform rolling upgrade

Rolling upgrade is not supported for this cluster, but is enabled via entitlement. Some cluster services might become unavailable during upgrade and running workloads could be impacted. For limitations and details please see the rolling upgrade documentation.

The Cloudera Runtime version of your cluster will be upgraded to 7.2.17 and an OS upgrade will also be performed. See the Public Cloud Runtime release notes for the list of changes and fixes in this version. This upgrade requires a downtime. Further details are available in the CDP Public Cloud upgrade advisor and the documentation for Upgrading Data Hubs.

Version details

	Image Date	CM Version	OS Type	CM Build number	Cloudera Runtime Version	Cloudera Runtime Build number
Current	2023-11-29	7.11.0	centos7	47027442	7.2.17	46967063
Target	2024-02-02	7.11.0	centos7	48415235	7.2.17	49883770

An OS upgrade is designated by "(OS upgrade, OS: <target-os>)" when you select the drop-down menu:

Upgrade Data Lake ⟳

Current Data Lake Version: 7.2.15

ⓘ A new upgrade is available for this Data Lake. You can continue through the process below.
Cloudera Runtime 7.2.18 is only available with Red Hat Enterprise Linux 8 OS images. Make sure to upgrade your cluster to RHEL 8 to unlock the upgrade to 7.2.18.

Select Upgrade

Target Cloudera Runtime Version

| Please select a runtime version | ▼ |

Please select a runtime version

7.2.17 (Runtime upgrade, OS: centos7)

7.2.18 (Runtime upgrade, OS: centos7)

7.2.15 (OS upgrade, OS: centos7)

nentation. To check whether rolling upgrade could be enabled for this

	Image Date	CM Version	OS Type	CM Build number	Cloudera Runtime Version	Cloudera Runtime Build number
Current	2023-11-20	7.6.2	centos7	42790603	7.2.15	47111845

⚠️ **Attention:** There are important differences between the nature of major/minor, service pack, and OS upgrades. Read the details below carefully to understand which type of upgrade is appropriate for your situation.

Process

The Data Lake upgrade process will:

1. Check for a newer Cloudera Manager and Runtime version or build, and a new OS image.
2. Automatically create a backup of the Data Lake (for major/minor and service pack upgrades).
3. Execute the Data Lake upgrade.
4. Verify the Data Lake state.

Major/minor version upgrades

Major/minor version upgrades are available as new versions of Cloudera Runtime and Cloudera Manager are released. Version upgrades combine a Runtime and CM upgrade into one operation. For example, this upgrade might involve upgrading from Runtime 7.2.8 to 7.2.9, and Cloudera Manager from 7.4.0 to 7.4.1.

Data Lake version upgrades require you to stop all of the Data Hubs in the environment before performing the Data Lake upgrade. The upgrade process stops all of the Data Lake services, downloads and distributes the new Cloudera Runtime, restarts the services, and deletes the old Cloudera Runtime. This process includes launching entirely new instance(s) from new cloud image(s). Once the Data Lake upgrade is complete, you can then either upgrade your Data Hub clusters to the same version as the Data Lake (if your Data Hub cluster type is supported for upgrade), or delete and recreate the clusters with the new Data Lake version. If your Data Hub is on Runtime version 7.2.16 or later, it is compatible with a Data Lake on a newer Runtime version (7.2.17+). You can independently upgrade your Data Hubs at a later time if you choose to, though it is not required

⚠️ **Important:** Cloudera Runtime 7.2.17 has new dependencies that were not present in most of the previous Runtime versions. Because of this, you may be unable to upgrade directly to this version and future service packs without first upgrading to a more recent Runtime version.

If you plan to upgrade your existing Data Lakes from a previous release to 7.2.17 or later, you can verify whether or not you will first be required to perform an additional upgrade step:

1. Select the Data Lake that you want to upgrade and click on the Upgrade tab.
2. If you see a warning message about missing prerequisites, follow the given steps to perform the additional upgrade before you upgrade to 7.2.17.

If your current Runtime version is 7.2.16 or any 7.2.16 service packs, these additional steps will include first performing an OS upgrade before upgrading to 7.2.17.

If your current Runtime version is 7.2.12, 7.2.14, or 7.2.15, these steps may include upgrading to the most recent service pack of your current Runtime version, as well as performing a separate OS upgrade for your current Runtime version, before you can perform a major/minor version upgrade to 7.2.17.

Service pack upgrades

The service pack upgrade process checks to see if a new Cloudera Manager or Cloudera Runtime (CDP) build is available, and then upgrades the Data Lake to the newest builds. Service pack upgrades do not upgrade to a new version of Cloudera Manager or Cloudera Runtime; they only upgrade to the latest service pack of a particular version. For example, a service pack upgrade cannot take the Data Lake from Runtime version 7.2.11 to version 7.2.12, but instead upgrades Runtime 7.2.11 to a newer 7.2.11 build. These upgrades are made available as needed and can deliver targeted bug fixes for Cloudera Runtime, Cloudera Manager, or both. If desired, specific older service packs can also be chosen using the CDP CLI. The service pack upgrade process follows the same steps as the version upgrade process.

OS upgrades

OS upgrades may be available sporadically as new images are created. OS upgrades are typically released to address security vulnerabilities or other issues on the host system. The OS upgrade process includes launching entirely new instances with the new OS image. An OS upgrade triggers the execution of any pre-service-deployment, post-cluster-manager-start, or post-service-deployment recipes.

Rolling upgrades

Certain Data Lake upgrades can be performed in a rolling fashion, depending on the Data Lake shape, Data Lake OS, and the Runtime version you are upgrading to and from. For more information, see Data Lake rolling upgrades.

Data Lake upgrade support matrix

The following Data Lake upgrade paths are supported.

In the below table, find your current Runtime version to find out the newest Runtime version that you can upgrade to:

Table 1:

Current Runtime Version	Target Runtime Version
7.2.7	7.2.16
7.2.8	7.2.16
7.2.9	7.2.16
7.2.10	7.2.16
7.2.11	7.2.16
7.2.12	7.2.17

Current Runtime Version	Target Runtime Version
7.2.14	7.2.17
7.2.15	7.2.17
7.2.16	7.2.17
7.2.17 (RHEL + EDL) **Disposition: / Status:** CB-24452 - 7.2.18	7.2.18

⚠️ **Important:**

- **Disposition: / Status:**
  CB-24452 - 7.2.18

  In order to upgrade to 7.2.18, your Data Lake must be using RHEL 8 and Enterprise or Light Duty shape.
- If you are upgrading to 7.2.17, you may be required to perform a service pack upgrade first.

  If you are planning an update to Runtime 7.2.18, refer to Upgrading to Runtime 7.2.18.

- For service pack upgrades, both non-RAZ (Ranger Authorization) and RAZ-enabled Data Lakes are eligible for upgrade from versions 7.2.7+.
- For major/minor version upgrades, Ranger Authorization (RAZ) enabled Data Lakes are eligible for upgrade from versions 7.2.10-7.2.12 to 7.2.14+.

Before you begin

Before you begin a Data Lake upgrade, note the requirements and limitations listed below.

Requirements

- Required role to perform Data Lake upgrade: EnvironmentAdmin or Owner over the environment
- The Data Lake must be running and in a healthy state.
- You should stop any Data Hubs and any data services (such as CDW or CDE) that are running. For the Cloudera Data Warehouse Experience, you should stop any Virtual Warehouses that are running prior to beginning any upgrade or backup/restore process. Stopping Experiences is not required for service pack upgrades, but any Data Hubs or data services that are not stopped will error out during the upgrade process.
- If you use a custom image catalog and you don't see upgrades available, you may need to update your custom image catalog with new images.
- **Disposition: / Status:**
  CB-24372 RHEL upgrade

  If the upgrade involves upgrading from CentOS to RHEL, review the Prerequisites for upgrading from CentOS to RHEL.
- Expect at least two hours of downtime while the upgrade completes. Plan the upgrade during a time of low activity.
- Optionally, you can take a backup of the Data Lake. The Data Lake upgrade process will automatically take a backup before the upgrade procedure begins, but you have the option of disabling the automatic backup if you

would prefer to do this step separately. For instructions on performing a backup and restore, see *Backup and restore for the Data Lake*. If the upgrade fails for any reason, you can restore the Data Lake from the backup.

> ⚠️ **Important:** Even if you are using the automatic backup integrated with the Data Lake upgrade flow, verify that you have correctly configured the Data Lake backup process. Configuring the backup process includes granting the required permissions from the cloud provider side, and adding a RAZ policy for RAZ-enabled environments. See *Configuring and running Data Lake backups* for more information on required configurations for backup. You do not need to actually run the backup if you plan to use the auto-backup during a Data Lake upgrade.
>
> For RAZ-enabled AWS environments, it is required that you manually add the RAZ backup policy for any AWS Data Lake created with a Runtime version prior to 7.2.15. However, for new environments created with a Data Lake running Runtime versions 7.2.15+, the RAZ backup policy is automatically configured for RAZ-enabled environments. If your Data Lake has been upgraded to 7.2.15 (as opposed to a new Data Lake created with 7.2.15), you will still have to manually add the RAZ backup policy.
>
> For RAZ-enabled Azure environments, it is required that you manually add the RAZ backup policy for any Azure Data Lake created with a Runtime version prior to 7.2.14. However, for new environments created with a Data Lake running Runtime versions 7.2.14+, the RAZ backup policy is automatically configured for RAZ-enabled environments. If your Data Lake has been upgraded to 7.2.14+ (as opposed to a new Data Lake created with 7.2.14), you will still have to manually add the RAZ backup policy.

The upgrade requires 27 GB space on the CM server node and 20 GB on every other instance. If space is insufficient on your Data Lake, upgrade will not be permitted.

Limitations

Note the following limitations for the Data Lake upgrade:

- Data Lake upgrade does not include the upgrade of the FreeIPA software or the operating system on the instance(s) running FreeIPA. To upgrade FreeIPA, see *Upgrade FreeIPA*.
- Data Lake resizing (moving from a light duty to a medium duty Data Lake) during an upgrade is not supported.
- Zero-downtime upgrade of Data Lake services is not supported.
- If a Data Lake has attached Data Hubs that are not eligible for upgrade, the Data Lake itself is not eligible for upgrade. You must delete any Data Hubs that are ineligible for upgrade before proceeding with the Data Lake upgrade. See *Data Hub Upgrade* for more information about which Data Hubs are eligible for upgrade.
- Service pack upgrades for RAZ-enabled Data Lakes are available only for Runtime versions 7.2.7+.
- Major/minor version upgrades for RAZ-enabled Data Lakes are available only for Runtime versions 7.2.12+.
- **Disposition: / Status:**
  CB-24372 RHEL upgrade

  A Data Lake must be using Runtime 7.2.17 to be eligible for CentOS to RHEL upgrade. If you do not see the option to upgrade from CentOS to RHEL, ensure that your Data Lake is using Runtime 7.2.17.
- **Disposition: / Status:**
  CB-24372 MD to EDL upgrade

  Runtime 7.2.18 and newer do not support Medium Duty Data Lake shape and no upgrades are possible from 7.2.17 to 7.2.18 without doing a resize operation on the Data Lake prior to upgrading to 7.2.18.

Related Information

Backup and restore for the Data Lake

Upgrade FreeIPA

Data Hub Upgrade

Configuring and running Data Lake backups

Upgrading a Data Lake

If a new Runtime/CM version or build is available for the Data Lake, you can initiate an upgrade from the Management Console. An OS upgrade may also be available.

About this task

In most cases it is not required that you destroy/recreate any Data Hubs attached to the Data Lake cluster. For major/minor version upgrades, you must upgrade the Data Hubs themselves after you upgrade the Data Lake, with the exception of Data Hubs on Runtime version 7.2.16 and later. If your Data Hub is on Runtime version 7.2.16 or later, it is compatible with a Data Lake on a newer Runtime version (7.2.17+). You can independently upgrade your Data Hubs at a later time if you choose to, though it is not required.

Any Data Hubs or data services that are not stopped during a Data Lake upgrade will error out during the upgrade process.

Required role: EnvironmentAdmin or Owner over the environment

Procedure

1. Stop all Data Hubs attached to the environment.

2. From the Management Console, click Data Lakes<Environment Name>, scroll to the bottom of the Data Lake details page, and click the Upgrade tab.

3. Click the Target Cloudera Runtime Version drop-down menu to see any available upgrades for a given Runtime version.

   If a new build is available for the selected version, the UI displays the current and target versions and build numbers. If only an OS upgrade is available, the UI displays "(OS upgrade only)."

   When a major/minor version upgrade is available, you'll be able to select a new Runtime version:

   Disposition: / Status:
   CB-24372 RHEL upgrade - Screenshots on this page have been updated

   Upgrade Data Lake ⟳

   Current Data Lake Version: 7.2.15

   ⓘ A new upgrade is available for this Data Lake. You can continue through the process below.
   Cloudera Runtime 7.2.18 is only available with Red Hat Enterprise Linux 8 OS images. Make sure to upgrade your cluster to RHEL 8 to unlock the upgrade to 7.2.18.

   Select Upgrade

   Target Cloudera Runtime Version

   7.2.17 (Runtime upgrade, OS: centos7)   ⌄

   ☐ Perform rolling upgrade

   Rolling upgrade is not supported for this cluster, but is enabled via entitlement. Some cluster services might become unavailable during upgrade and running workloads could be impacted. For limitations and details please see the rolling upgrade documentation.

   The Cloudera Runtime version of your cluster will be upgraded to 7.2.17 and an OS upgrade will also be performed. See the Public Cloud Runtime release notes for the list of changes and fixes in this version. This upgrade requires a downtime. Further details are available in the CDP Public Cloud upgrade advisor and the documentation for Upgrading Data Hubs.

   Version details

	Image Date	CM Version	OS Type	CM Build number	Cloudera Runtime Version	Cloudera Runtime Build number
Current	2023-11-29	7.6.2	centos7	42790603	7.2.15	47111845
Target	2024-02-02	7.11.0	centos7	48415235	7.2.17	49883770

   If a rolling upgrade is available, select the Perform rolling upgrade checkbox if you would like to perform this type of upgrade. The availability of a rolling upgrade depends on the current and target Runtime versions, the Data Lake shape, and the Data Lake OS. See Data Lake rolling upgrades for more information.

4. If you want to skip the automatic backup that is taken before the upgrade, uncheck the Automatic backup box. For more information on what is backed up during a Data Lake backup, see *Data Lake backup and restore*.

5. Click Validate and Prepare to check for any configuration issues and begin the Cloudera Runtime parcel download and distribution. Using the validate and prepare option does not require downtime and makes the maintenance window for an upgrade shorter. Validate and prepare also does not make any changes to your cluster and can be run independently of the upgrade itself. Although you can begin the upgrade without first running the validate and prepare option, using it will make the process smoother and the downtime shorter.

6. Click Upgrade to initiate the upgrade.

7. Click the Event History tab to monitor the upgrade process and verify that it completes successfully.

If the upgrade fails for any reason, check the Data Lake logs through Cloudera Manager for troubleshooting information and retry the upgrade. If you cannot fix the problem manually, you may be able to recover the Data Lake cluster after a failed upgrade. For more information see *Recovering from failed upgrades*.

What to do next

For major/minor upgrades, if the upgrade is successful, you can proceed to upgrading your attached Data Hubs if required. Data Hub clusters must run the same Runtime version as the Data Lake, with the exception of Data Hubs on Runtime version 7.2.16 and later. If your Data Hub is on Runtime version 7.2.16 or later, it is compatible with a Data Lake on a newer Runtime version (7.2.17+). You can independently upgrade your Data Hubs at a later time if you choose to, though it is not required. For service pack and OS upgrades, you can restart your Data Hubs, data services, and any stopped Virtual Warehouses.

Related Information

Backup and restore for the Data Lake

Recovering from failed upgrades

Upgrading a Data Lake manually via CLI

You can initiate a Data Lake upgrade with the CDP CLI. Using the same CLI command, you can also search for and validate available images to upgrade to, and generate JSON templates for specific upgrade scenarios.

Disposition: / Status:
CB-24372 RHEL upgrade - Added the following new section for obtaining image ID

Obtain image ID

If your Data Lake upgrade includes upgrading from CentOS to RHEL 8, prior to attempting an upgrade you need to obtain an ID of a target RHEL 8 image. You can obtain it from the image catalog by finding an image with your target Runtime version which has an OS Type of RHEL8.

Image Catalogs / cloudbreak-default

cloudbreak-default   Delete

https://cloudbreak-imagecatalog.s3.amazonaws.com/v3-test-cb-image-catalog.json

Base Images   **Cloudera Runtime Images**

🔍 Search

UUID	Provider	OS Type	CM Version	CM Build Number	Stack Name	Stack Version	CDP Build Number	Tags	Created On	Published On
64e20e39-4e47-45f6-af31-619097ed9deb	Google Cloud	RHEL8	7.12.0.0	50381610	Cloudera Runtime	7.2.18	50345525	Default	2/19/2024	2/20/2024
ab7c20b4-54e8-44af-a189-3c753ecbae9e	Amazon	RHEL8	7.11.0	48415235	Cloudera Runtime	7.2.17	49883770	Default	2/2/2024	2/2/2024
d0a2f924-9216-45a0-8492-32c64d2a2adf	Azure	RHEL8	7.11.0	48415235	Cloudera Runtime	7.2.17	49883770	Default	2/2/2024	2/2/2024
7d74a1ca-c1aa-4d87-89ac-a43ebb1739d7	Amazon	RHEL8	7.9.2	38837416	Cloudera Runtime	7.2.16	38457977		3/21/2023	3/21/2023
fef3aec6-561f-46e5-bbad-4f847a45f364	Azure	RHEL8	7.13.0.0	49779725	Cloudera Runtime	7.3.0	49808143	Default	2/21/2024	2/21/2024
0d6da670-ce0c-4c71-8d0a-d370be3e76e6	Azure	RHEL8	7.12.0.0	50571627	Cloudera Runtime	7.2.18	50499817	Default	2/23/2024	2/23/2024
0cb55fb1-83b2-4fe7-afb1-d5043b401a94	Google Cloud	RHEL8	7.11.0	48415235	Cloudera Runtime	7.2.17	49883770	Default	2/2/2024	2/2/2024
c7a42cbc-f0a9-4fc2-93ff-59ae834c2d3b	Amazon	RHEL8	7.11.0	40466989	Cloudera Runtime	7.2.17	40465599		5/2/2023	5/2/2023

Once you have identified the ID, you can provide it in the upgrade CLI command by using the --image-id flag.

Upgrade steps

1. Run the cdp datalake upgrade-datalake command.

In order to use this command for upgrading from CentOS to RHEL, ensure to provide an image ID of a RHEL 8 image.

The command has the following options:

```
cdp datalake upgrade-datalake
          --datalake-name <value>
          [--image-id <value>]
          [--runtime <value>]
          [--lock-components | --no-lock-components]
          [--dry-run | --no-dry-run]
          [--show-available-images | --no-show-available-images]
          [--show-available-image-per-runtime | --no-show-available-image-
per-runtime]
          [--skip-backup | --no-skip-backup]
          [--skip-ranger-hms-metadata | --no-skip-ranger-hms-metadata]
          [--skip-atlas-metadata | --no-skip-atlas-metadata]
          [--skip-ranger-audits | --no-skip-ranger-audits]
          [--skip-backup-validation | --no-skip-backup-validation]
          [--cli-input-json <value>]
          [--generate-cli-skeleton]
```

Option	Description
--datalake-name (string)	Required. The name or CRN of the Data Lake to upgrade.
--image-id (string)	The ID of an image to upgrade to. **Disposition: / Status:** CB-24372 RHEL upgrade - Added this sentence. If upgrading from CentOS to RHEL, make sure to provide an image ID of a target RHEL image.
--runtime (string)	The Runtime version to upgrade to. In a maintenance upgrade this parameter is the current Runtime version. When you specify the Runtime version, the upgrade uses the latest image ID of the given Runtime version from the same image catalog used for Data Lake creation. If you specify an invalid Runtime version, you'll receive an error message that the version is not supported for upgrade.
--lock-components | --no-lock-components (boolean)	Use --lock components to perform an OS upgrade only.
--dry-run | --no-dry-run (boolean)	Checks the eligibility of an image to upgrade. Can be used in conjunction with any other parameter, returning the available image (with respect to image Id, Runtime or lock-components set) without performing any actions.
--show-available-images | --no-show-available-images (boolean)	Returns the list of images that are eligible to upgrade to.
--show-available-image-per-runtime | --no-show-available-image-per-runtime (boolean)	Returns the latest image that is eligible to upgrade to, for each Runtime version with at least one available upgrade candidate.
--skip-backup | --no-skip-backup	If provided, will skip the backup flow for the upgrade process.

Option	Description
--skip-ranger-hms-metadata | --no-skip-ranger-hms-metadata	Skips the backup of the databases backing HMS/Ranger services. Redundant if –skip-backup is included. If this option is not provided, the HMS/Ranger services are backed up by default.
--skip-atlas-metadata | --no-skip-atlas-metadata	Skips the backup of the Atlas metadata. Redundant if –skip-backup is included. If this option is not provided, the Atlas metadata is backed up by default.
--skip-ranger-audits | --no-skip-ranger-audits	Skips the backup of the Ranger audits. Redundant if –skip-backup is included. If this option is not provided, Ranger audits are backed up by default.
--skip-backup-validation | --no-skip-backup-validation	Skips the validation steps that run prior to the backup. Redundant if –skip-backup is included. If this option is not provided, the validations are performed by default.
--cli-input-json (string)	Performs service operation based on the JSON string provided. The JSON string follows the format provided by --generate-cli-skeleton. If other arguments are provided on the command line, the CLI values will override the JSON-provided values.
--generate-cli-skeleton (boolean)	Prints a sample input JSON to standard output. Note the specified operation is not run if this argument is specified. The sample input can be used as an argument for --cli-input-json.

When you run the cdp datalake upgrade-datalake command to initiate an upgrade, you have one of three options:

a. Specify one of either --image-id, --runtime, or --lockComponents, which makes an explicit choice of the exact image, Runtime (latest OS), or latest OS (same Runtime) for upgrade.
b. Specify both --image-id and --lockComponents, which specifies an image and ensures the image represents an OS only upgrade.
c. Specify none of the --image-id, --runtime, or --lockComponents parameters, which initiates a Runtime/CM upgrade to the latest compatible version and OS image.

Outside of upgrade, you can use the following options:

• --show-available-images/--no-show-available-images
• --show-available-images-per-runtime/--no-show-available-images-per-run   time
• --dry-run

Examples of valid inputs:

```
cdp datalake upgrade-datalake --datalake-name my-datalake --dry-run
cdp datalake upgrade-datalake --datalake-name my-datalake --image-id d1c
520b1-987d-461f-7860-918f43994c04
cdp datalake upgrade-datalake --datalake-name my-datalake --image-id d1c52
0b1-987d-461f-7860-918f43994c04 --dry-run
cdp datalake upgrade-datalake --datalake-name my-datalake --runtime 7.2.11
cdp datalake upgrade-datalake --datalake-name my-datalake --runtime 7.2
.11 --dry-run
cdp datalake upgrade-datalake --datalake-name my-datalake--lock-components
cdp datalake upgrade-datalake --datalake-name my-datalake --show-availa
ble-image-per-runtime
cdp datalake upgrade-datalake --datalake-name my-datalake- --show-availabl
e-images
```

Examples of incorrect inputs:

```
cdp datalake upgrade-datalake --datalake-name my-datalake --image-id 7.2
.11
cdp datalake upgrade-datalake --datalake-name my-datalake--runtime d1c52
0b1-987d-461f-7860-918f43994c04
```

```
cdp datalake upgrade-datalake --datalake-name my-datalake--lock-components
 --imageid imageid --runtime runtime
cdp datalake upgrade-datalake --datalake-name my-datalake --show-availabl
e-image-per-runtime --show-available-images
cdp datalake upgrade-datalake --datalake-name my-datalake --show-availab
le-image-per-runtime --dry-run
cdp datalake upgrade-datalake --datalake-name my-datalake --show-availabl
e-images --dry-run
```

Recovering from failed upgrades

If a Data Lake upgrade fails and you are unable to manually troubleshoot the problem, you may be able to use the recovery process to return the cluster to its pre-upgrade state.

About this task

If FreeIPA is available and the Data Lake cluster is in a recoverable state (meaning that there has been an uncorrected failed upgrade or failed recovery), a recovery option may be available after a failed upgrade. Recovery after a failed upgrade retains the Data Lake CRN, UMS mappings, load balancers, and RDS instance and brings up new instances with the original image and Runtime version, but new disks and new databases.

> **Note:** Data backup and restore is not currently part of the recovery process. Ensure that you have a Data Lake backup from which you can manually restore the Data Lake data after the successful recovery. The presence of the backup is not validated by the Management Console.

Disposition: / Status:
If FreeIPA is available and the Data Lake cluster is in a recoverable state (meaning that there has been an uncorrected failed upgrade or failed recovery), a recovery option will be presented in the Data Lake UI after a failed upgrade. Recovery after a failed upgrade retains the Data Lake CRN, UMS mappings, load balancers, and RDS instance and brings up new instances with the original image and Runtime version, but new disks and new databases.

Procedure

1. Use the CDP CLI to recover the Data Lake after a failed upgrade:

```
cdp datalake recover-datalake
--datalake-name <value>
[--recovery-type <value>]
```

Parameter	Description
--datalake-name	Name or CRN of the Data Lake that you want to recover after a failed upgrade.
--recovery-type	The type of the recovery. The default value is RECOVER_WITHOUT_DATA. Currently, the option RECOVER_WITH_DATA is not supported. **Disposition: / Status:** • RECOVER_WITH_DATA • RECOVER_WITHOUT_DATA When the RECOVER_WITH_DATA option is supported, a validation checks to see if a backup of

Parameter	Description
	the DL has ever been taken, and if so, will restore the existing backup automatically.

The status of the Data Lake appears as "Datalake recovery in progress. Recovery process takes a while as the nodes are being terminated and new nodes are launched with the original runtime."

Event History   Endpoints (6)   Tags (5)   Hardware   Network   Telemetry   Repository Details   Image Details   Recipes (0)   Cloud Storage   Attached clusters (0)   Database   Upgrade

Events

⬇ DOWNLOAD

- ✅ Cluster recovery has been completed
  25/01/2022, 17:32:20

- ✅ Installing CDP services
  25/01/2022, 17:08:19

- ✅ Starting Cloudera Manager
  25/01/2022, 17:02:40

- ✅ The generation of valid certificate has been failed, installation of your cluster is continuing with a generated self-signed certificate.
  25/01/2022, 17:02:22

- ✅ Bootstrapping cluster
  25/01/2022, 17:01:33

- ✅ Creating infrastructure
  25/01/2022, 16:52:55

- ✅ Setting up CDP image
  25/01/2022, 16:52:49

2. Restore the Data Lake from the pre-upgrade backup. For more information, see *Restore Data Lake content*.

3. If necessary, run the cdp datalake sync-component-versions-from-cm command from the CDP CLI.

   When an upgrade fails, the versions of Cloudera Manager, Runtime, and other components may become out-of-sync with the CDP Management Console. Similarly, if you try to fix errors by installing parcels manually, it may not be reflected in the CDP Management Console.

   To overcome the mismatch between versions reflected in the Management Console, run the cdp datalake sync-component-versions-from-cm CDP CLI command. This commands reads the CM, Runtime, and other parcel versions (if applicable) from CM and updates the versions in the CDP Management Console. Using this command forces the CDP Management Console back in sync so that it shows the actual versions installed in CM.

   ```
   cdp datalake sync-component-versions-from-cm --datalake-name <datalake n
   ame>
   ```

Related Information
Configuring and running Data Lake restore

Performing manual Data Lake repair

If a Data Lake node fails, an administrator can start a manual recovery process from the CDP web interface. Because the state of Data Lake services is stored externally, the repair operation is able to deploy the services on a new node and reattach the all workload clusters without data loss and with minimal downtime.

Required role: EnvironmentAdmin or Owner of the environment

When a Data Lake cluster has unhealthy nodes, warnings appear in the Data Lake page:

- Nodes are marked as "UNHEALTHY" in the Hardware tab for the Data Lake.
- Data Lake cluster's Event History shows "Manual recovery is needed for the following failed nodes."

You can perform manual repair from the CDP web UI or CLI.

Manual repair from web UI

To perform manual repair from CDP web UI:

1. Log in to the CDP web interface.
2. Navigate to the affected Data Lake using Management Console Data Lakes.
3. In the Data Lake details page, click choose one of the following options:

   - To repair failed nodes in a specific host group, click Repair and select the host group that should be repaired. Only one host group can be selected at a time. Then click Repair.
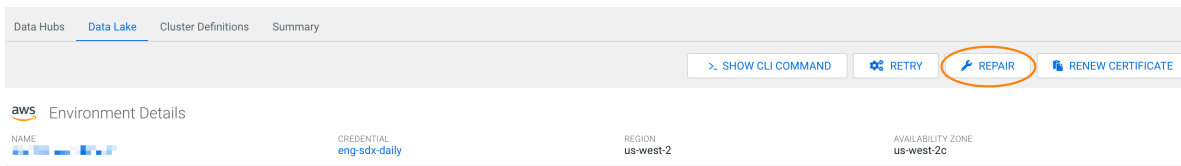
   > **Note:** If no host groups are listed as in need of repair, use Cloudera Manager to determine what might be causing the problem you are experiencing.

   - To repair a single node failure or select certain nodes within a host group to repair, select the Hardware tab and then the repair icon next to the host group that contains the failed node(s).
4. When you initiate a repair from the Hardware tab, you also have the option to delete any volumes attached to the instance. This can be useful if a volume is lost on the cloud provider side. To delete the attached volumes, select the Delete Volumes checkbox.

When the recovery flow is completed, the cluster status changes to "RUNNING".

Manual repair from CLI

To perform manual repair from the CLI, use the following commands:

- cdp datalake list-datalakes – Check the status and health of your Data Lake clusters
- cdp datalake describe-datalake – Check the status and health of a specific Data Lake cluster
- cdp datalake repair-datalake – Perform Data Lake cluster repair.

Related Information
Data Lake repair
Data Lake storage
Cloudera Manager health tests
Cloudera Manager logs

Backup and restore for the Data Lake

You can backup and restore the metadata maintained in the Data Lake services. Use this comprehensive backup to restore your Data Lake's metadata to the state it was at when the backup was taken.

The backup and restore operation creates a comprehensive backup that improves the likelihood of data in the backup to be synchronized across all the services running in the Data Lake.

Required role: EnvironmentAdmin

> **Note:** Data Lake backup and restore is supported from Cloudera Runtime 7.2.1+ on AWS and Cloudera Runtime 7.2.2+ on Azure, Cloudera Runtime 7.2.15+ on GCP, and Cloudera Runtime 7.1.0/7.2.0 on HDFS.

Use the Data Lake backup and restore:

- in preparation for a Data Lake upgrade.
- for archiving Data Lake content for later reference.

- to promote Data Lake content from test to production.

There will be a downtime when a Data Lake backup is performed, as some Data Lake services will be stopped. Additionally, access to the HMS database will be blocked by default for the duration of the backup, but you can optionally bypass this using the CLI option no-close-db-connnections when you run the backup. This option allows Data Hub workloads to continue running during the Data Lake backup process. See *Configuring and running Data Lake backups* for more information.

Important: Data lake backup/restore operations should be performed when the Data Lake is quiescent. In order to have a consistent backup Cloudera recommends that no workloads are running when the backup is performed. In order to make sure there are no updates to HMS, the backup and restore process closes all the connections to the HMS backend database. This is important for migration use cases where you would like to move metadata from one environment to another. However, you are not required to stop the Data Hub workloads before performing backup. If you want to keep Data Hubs running during a backup, you must keep the HMS database connections open to the Data Lake using the no-close-db-connnections CLI option. If the database connections are closed, Data Hub workloads will fail.

Note that a Data Lake Backup includes metadata about your cluster workloads and does not include the data itself.

At this time, you can trigger a Data Lake backup through a CDP command-line operation from any host with connectivity to the Data Lake. The system checks to make sure there isn't another backup or restore in progress.

What's backed up?

The backup operation saves a full snapshot of data from all SDX services:

- Atlas:

    - Audit events, saved in HBase tables
    - Lineage data, saved as Janus graph data in HBase tables
    - Edge, vertex, and full text indexes, saved in Solr collections
- Ranger:

    - Audit logs, saved as a Solr collection
    - Permissions and tags, saved in RDBMS tables
- HMS Metadata, saved in RDBMS tables

How do I keep the backup secure?

The backup files are stored on AWS S3, ADLS, or Google Cloud Storage with encryption enabled.

How often should backups run?

Disposition: / Status:
//REVIEWERS: Please check this section for accuracy.

You can run backups as part of these events:

- Upgrades: a backup can be performed before performing an upgrade. This backup can be used to restore the existing environment or create a new environment in case the upgrade fails in a manner that requires you to re-build the original environment.
- Moving the Data Lake metadata (Atlas lineages, Ranger policies and audit information, and HMS metadata) from one environment to another.

When backups are taken, there is downtime for principal services (see "Principal services" below) due to the requirement to shut down HMS service on any attached Data Hubs.

Is there any validation that occurs before the backup/restore operations?

Yes. Before a backup or restore operation begins, a validation process checks for the most common causes of a backup or restore failure. These validations include verifying that the required permissions are granted on the backup

location with the cloud provider (AWS and Azure) and that the required Ranger RAZ permissions are granted on the backup location. For the backup operation only, the validation process also checks if there are any stale Ranger Audit permissions that are over 90 days old.

You can use CLI commands when you run the backup/restore operations to skip the validation process, or run the validation process without proceeding to the backup/restore operations. See *Configuring and running Data Lake backups* and *Configuring and running Data Lake restore* for more information.

Data Lake Restore vs Repair

Data Lake repair replaces the compute resources and reconnects them to the persistent Data Lake storage. Data Lake restore replaces the existing Data Lake content with content from a Data Lake backup.

Principal services

The following principal services affect backup and restore operations:

- On the Data Lake:

  - Atlas
  - HMS
- HMS Services on any attached Data Hub.

When a backup is performed, the Atlas service is stopped. The HMS service will be impacted if the "--no-close-db-co nnections" option is not provided.

Dependent services

Backup and restore operations are dependent on the following services:

- HBase
- Solr
- ZooKeeper
- Databases services (e.g. Postgres)

They must be running during a backup or restore operation.

Other considerations

Do not stop or restart Data Lake services in Cloudera Manager if you are planning on running backup or restore operations. If you stop or restart Data Lake services from Cloudera Manager, such as restarting all services, or restarting the HBase or Solr services, Data Lake backup and restore operations from the CLI will be allowed to go forward, but may fail.

In order to have a consistent backup Cloudera recommends that no workloads are running when the backup is performed. In order to make sure there are no updates to HMS, the backup and restore process closes all the connections to the HMS backend database. This is important for migration use cases where you would like to move metadata from one environment to another. However, you are not required to stop the Data Hubs attached to a Data Lake before performing backup. If you want to keep Data Hubs running during a backup and restore operation, you must keep the HMS database connections open to the Data Lake using the no-close-db-connnections CLI option. If the database connections are closed, Data Hub workloads will fail.

You should stop all Data Hubs before performing a restore operation; a running Data Hub will prevent restore from functioning, and may result in inconsistent data in the backup.

What's supported?

Support for AWS	Runtime 7.2.1 and above
Support for Azure	Runtimes 7.2.2 and above
Support for GCP	Runtimes 7.2.15 and above

Support for Backup and restore across Data Lakes with different shapes For example, a backup taken on a light duty Data Lake can be used to restore a medium duty Data Lake.	Runtimes 7.2.8 and above
Support for RAZ-enabled Data Lakes	Runtimes 7.2.10 and above
Support for Runtimes 7.1.0 and 7.2.0	Data Lake backup only writes to HDFS on the Data Lake. The backup should be moved to and from cloud storage using the provided procedures. Contact Cloudera Support for more information.
Cross-version support (Restoring a backup taken from a different version of Cloudera Runtime)	You can take a backup of a Data Lake that runs one version of Cloudera Runtime and restore the backup to a Data Lake that runs a different version of Runtime. The backup version must be an earlier/ lower version Runtime than the Data Lake that you are restoring to. Version limitations apply and a Ranger/HMS schema upgrade may be required. See *Cross-version support for Data Lake backup and restore* for more details.

What's not supported?

Backup and restore across different cloud providers is not supported	Example: Restoring a backup of an AWS Data Lake to an Azure Data Lake, or an Azure Data Lake to an AWS Data Lake is not supported.
Recovery for individual settings or pieces of metadata.	Example: Recovering just the Hive SQL Ranger policies from a given backup, or recovering just the HMS metadata of a particular database.
Point in Time Recovery of data synced with metadata	A restore will only recover HMS metadata and will apply that to the existing data used by the Data Hubs of the environment. This means that HMS metadata may not be in sync with the data.

Limitations in the backup and restore system

• With Cloudera Runtime 7.2.1, the initial backup on AWS may be written to an S3 bucket. All subsequent backups must be written to the same bucket as the initial backup. This limitation does not exist for later runtime versions.

Disposition: / Status:

All batch backup systems suffer from the problem that the backup operation is not instantaneous and work continues on the system during the backup. This time lag causes a gap between the current system content and the backup content. Should you have to restore a backup, the gap makes it possible for the Data Lake backup content to be out-of-sync with the current workload data. In addition, the backup operation can catch metadata operations in the middle of a transaction among the Data Lake services or among a service's components. Be aware of these two areas of potential mismatches when a backup is restored:

Workload data and Data Lake metadata out-of-sync

When Data Lake content is restored, some of the restored metadata could be out-of-sync with the current data. Some examples of this kind of mismatch are:

• Hive Metastore: If there were DDL operations performed in the HMS after the last backup was performed, those operations will be lost when the Data Lake backup is restored. In cases like these HMS metadata could be out-of-sync with actual data.
• Ranger: Permissions created/altered/deleted after the last backup was performed will be lost when the Data Lake is restored. These changes will have to be recreated.
• Ranger: Audits against resource or tag policies that were collected after the last backup was performed will be lost when the Data Lake backup is restored. There is no way to recreate these audits.

To reduce the likelihood of metadata and data being out-of-sync after backup content is restored, consider the following guidelines for timing backups:

• Reduce the interval between backup. The chances of inconsistencies is reduced when the interval between backups is shorter.

- Perform backups while workloads are in a quiescent state. An example would when services are shut down right before an upgrade.

Metadata out-of-sync

The automated backup operation performs backups for each Data Lake metadata component concurrently. Each backup operation provides consistency guarantees individually for that component; however, taken together, the set of backups does not guarantee system-wide consistency.

For example, Atlas lineage information is stored both in Solr collections and HBase tables. As HBase and Solr backups can not be done transactionally, it is possible that there could be some lineage information in one service and not in another. To minimize the impact of metadata mismatches among services, the backups of Solr and HBase are performed concurrently.

- Atlas: If Atlas metadata for lineage is being updated while a backup is in progress, there is a chance that metadata recorded in HBase tables does not have matching data in the Solr index. If the data is mismatched between the two systems, entities that exist in HBase may not appear in the Basic Search (Solr index). To resynchronize the index with the HBase data, see <link to https://github.com/apache/atlas/blob/master/tools/atlas-index-repair/README>

Related Information
Cross-version support for Data Lake backup and restore

Cross-version support for Data Lake backup and restore

You can take a backup of a Data Lake that runs one version of Cloudera Runtime and restore the backup to a Data Lake running a different version of Runtime.

⚠️ **Important:** Depending on the target backup and restore versions, a schema update may be required for Ranger and HMS. For cases where a schema upgrade is required, contact Cloudera Support.

Cross-version backup and restore is supported for the following Data Lake versions:

Table 2:

Source (Backup) Data Lake Runtime Version	Target (Restore) Data Lake Runtime Version	Schema Upgrade Required?
7.2.12	7.2.15	Yes
7.2.12	7.2.16	Yes
7.2.14	7.2.15	No
7.2.14	7.2.16	Yes
7.2.15	7.2.16	Yes
7.2.15	7.2.17	Yes
7.2.16	7.2.17	No

For these Runtime versions, cross-version backup and restore is supported for all cloud providers and all Data Lake shapes.

Configuring and running Data Lake backups

The Data Lake provides a command line interface for managing Data Lake backup and restore operations. The system checks to make sure there isn't another backup or restore in progress.

Configure the backup

Before you begin

- Create the S3, ABFS, or GCS backup location before performing the backup. For Azure, the container where the backup is stored should be in the same storage account as the Data Lake being backed up.
- Shut down principal services (see Principal services on page 49). This will help avoid mismatches between Data Lake metadata and data used by workloads and mismatches among the metadata stored in the Data Lake.
- Stop all Data Hubs attached to the Data Lake before you perform any backup or restore operations.
- Stop any Virtual Warehouses that are running.

Configuring backups for AWS:

- Apply the IAM policy for Data Lake backup to the following roles:

    - DATALAKE_ADMIN_ROLE
    - RANGER_AUDIT_ROLE

      For more informaiton on IAM roles, see *Minimal setup for cloud storage*.

      In the IAM policy for Data Lake backup, be sure to replace the <BACKUP_BUCKET> variable with the backup location used.

Note that if you plan to restore the Data Lake backup that you are taking, you must also apply a restore policy to certain roles. For more information on restore see Configuring and running Data Lake restore on page 60.

Configuring RAZ for backup

This section applies only to RAZ-enabled AWS Data Lakes. For RAZ-enabled Azure Data Lakes, see the section below.

Add a RAZ policy to allow the backups to be written to and read from the backup location.

- Open the Ranger UI.
- Go to the cm_s3 policy list.

- Add a new policy:
  - Policy name: backup_policy
  - S3 bucket: The bucket where backups will be stored
  - Path: The path(s) in the bucket where backup will be written

    Note: If more than one bucket will be used for backup, create a separate policy for each bucket.

- Add read and write permissions for the atlas, hbase, hdfs, and solr users under "Allow Conditions."

> **Note:** If RAZ is enabled on the Data Lake, then during the backup you will see denied audit logs in Ranger related to the HBase user attempting to change ownership of the backup directory. These are expected and don't include a problem with the backup.

Configuring backups for Azure:

- Verify that the following identities have the "Storage Blob Data Contributor" role on the container where the backup is stored:

  - Data Lake Admin identity
  - Ranger Audit Logger identity

Configuring RAZ for backup

This section applies only to RAZ-enabled Azure Data Lakes. For RAZ-enabled AWS Data Lakes, see the section above.

Add a RAZ policy to allow the backups to be written to and read from the backup location.

- Open the Ranger UI.
- Go to the cm_adls policy list.

- Add a new policy:

  - Policy name: backup_policy
  - Storage Account: The storage account where backups will be stored
  - Storage Account Container: The container where backups will be stored
  - Path: The path(s) in the bucket where backup will be written

    Note: If more than one storage account or container will be used for backup, create a separate policy for each account/container.

- Add read, write, list, delete, delete recursive, and move permissions for the atlas, hbase, hdfs, and solr users under "Allow Conditions."

> **Note:** If RAZ is enabled on the Data Lake, then during the backup you will see denied audit logs in Ranger related to the HBase user attempting to change ownership of the backup directory. These are expected and don't include a problem with the backup.

Configuring backups for GCP:

Verify that the Ranger Audit Service account has the following required permissions:

- resourcemanager.projects.get
- resourcemanager.projects.list
- storage.buckets.get
- storage.objects.create
- storage.objects.delete
- storage.objects.get
- storage.objects.getIamPolicy
- storage.objects.list
- storage.objects.setIamPolicy
- storage.objects.update

Note that the Ranger Audit service account permissions listed above should be granted to a custom role, not the default Storage Object Admin role.

You should also modify the scope of the Data Lake Admin and Ranger Audit service accounts to include the Backups bucket, if the bucket is different from the main data storage bucket. For more information see Minimum setup for cloud storage.

Run the backup

Procedure

1. Log into a computer that has connectivity to the Data Lake host.
2. Install the CDP CLI Client.
3. Switch to a user account that has the environment admin role.
4. Run a backup.

   Use the following command to run the Data Lake backup: $ cdp datalake   backup-datalake --datalake-name < name> --backup-location <cloud storage   location>

   Where the options are the following:

Option	Example	Description
--datalake-name	finance-dl	This is the name of the Data Lake as configured in the CDP environment. Required.
--backup-location	s3a://acme-finance-admin-bucket/backup-archiveor abfs://<container-name>@mydatalakesan.dfs.core.windows.net/backup_01/	The fully qualified name of the S3 bucket and object or ABFS location where the backup operation writes files. For S3 use the "S3a" file system protocol. Required.
--backup-name	pre-upgrade0420	An optional label that helps you identify one backup from another. The backup name can be used to identify a backup for restoring.
--close-db-connections | --no-close-db-connections		Specifies whether Ranger/HMS connections to the Data Lake should be closed or not during the backup. If you want to take the backup without workload downtime, use no-close-db-connections. Using this option means there could be changes to the Ranger/HMS data while the backup is performed. The connections are closed by default.
--skip-validation | --no-skip-validation		Using --skip-validation skips the validation that occurs before the backup process begins. This validation checks for required permissions that are often the source of backup/restore failures. See *Backup and restore for the Data Lake* for more details.
--validation-only | --no-validation-only		--validation-only runs the pre-backup and restore validation process, but does not proceed to the actual backup/restore operation. See *Backup and restore for the Data Lake* for more details.
--skip-ranger-hms-metadata | --no-skip-ranger-hms-metadata		Skips the backup of the databases backing HMS/Ranger services. If this option is not provided, the HMS/Ranger services are backed up by default.
--skip-atlas-metadata | --no-skip-atlas-metadata		Skips the backup of the Atlas metadata. If this option is not provided, the Atlas metadata is backed up by default.

Option	Example	Description
--skip-ranger-audits | --no-skip-ranger-audits		Skips the backup of the Ranger audits. If this option is not provided, Ranger audits are backed up by default.

For backups, the --skip-ranger-hms-metadata and --skip-atlas-metadata flags cannot be used at the same time.

On AWS:

```
$ cdp datalake backup-datalake --datalake-name finance-dl
     --backup-location s3a://acme-finance-admin-bucket/backup-archive
     --backup-name pre-upgrade0420
```

On Azure:

```
$ cdp datalake backup-datalake --datalake-name my-datalake
     --backup-location abfs://<container-name>@mydatalakesan.dfs.core.w
indows.net/backup_01/
```

On GCP:

```
$ cdp datalake backup-datalake --datalake-name my-datalake —backup-locat
ion gs://<bucket-name>/backup
```

The output of the command shows the current status of the operation. Note the internal state shows the status of each separate backup operation. If any of the individual backups fail, the overall status is failed and the backup cannot be restored. (Line breaks added for readability.)

```
{
     "accountId": "9d74eee4-1cad-45d7-b654-7ccf9edbb73d",
     "backupId": "415927d9-9f7d-4d42-8000-71630e5938ca",
     "internalState": "{ATLAS_ENTITY_AUDIT_EVENTS_TABLE=IN_PROGRESS, EDGE_
INDEX_COLLECTION=IN_PROGRESS, DATABASE=SUCCESSFUL, FULLTEXT_INDEX_COLLEC
TION=IN_PROGRESS, ATLAS_JANUS_TABLE=IN_PROGRESS, RANGER_AUDITS_COLLECTIO
N=IN_PROGRESS, VERTEX_INDEX_COLLECTION=IN_PROGRESS}",
     "status": "IN_PROGRESS",
     "startTIme": "2021-04-20 20:10:16.567"
     "endTIme": "2021-04-20 20:10:45.341"
     "backupLocation":"s3a://acme-finance-admin-bucket/backup-archive
/backup-archive",
     "backupName": "pre-upgrade0420"
     "failureReason": ""
}
```

What to do next
To see the status of the backup after the initial command, see "Checking the status of a Data Lake backup."
Related Information

Backup and restore for the Data Lake

Minimal setup for cloud storage (AWS)

Minimal setup for cloud storage (Azure)

Minimum setup for cloud storage (GCP)

Checking the status of a Data Lake backup

Checking the status of a Data Lake backup

After configuring and running a backup of your Data Lake, you can check the status of the backup operation.

Checking backup status

Use the following command to see the status of a Data Lake backup:

```
$ cdp datalake backup-datalake-status
      --datalake-name <name>
    [--backup-id <generated-ID>]
    [--backup-name <name>]
    [--cli-input-json <string>]
    [--generate-cli-skeleton]
```

where the options are the following:

Option	Example	Description
--datalake-name	finance-dl	This is the name of the Data Lake as configured in the CDP environment.
[--backup-id]	415927d9-9f7d-4d42-8000-71630e5938ca	The system-generated ID for the backup. If you don't know the ID or name for the backup, run the list-datalake-backups command to see the available backups. If neither an ID or a name is provided, the command shows the status of the most recent backup operation.
[--backup-name]	pre-upgrade0420	The user-provided name for the backup. If you don't know the name or ID for the backup, run the list-datalake-backups command to see the available backups. If neither an ID or a name is provided, the command shows the status of the most recent backup operation.
--cli-input-json <string>		Performs service operation based on the JSON string provided. The JSON string follows the format provided by --generate-cli-skeleton. If other arguments are provided on the command line, the CLI values will override the JSON-provided values.
--generate-cli-skeleton		Prints a sample JSON configuration file to standard output. If this argument is specified, only the template is produced: the list-backup command does not run.

For example:

```
$ cdp datalake backup-datalake-status
      --datalake-name finance-dl --backup-id
      415927d9-9f7d-4d42-8000-71630e5938ca
```

The output of the command shows the current status of the backup operation. Note the internal state shows the status of each separate backup operation. If any of the individual backups fail, the overall status is failed and the backup cannot be restored.

```
{
    "accountId": "9d74eee4-1cad-45d7-b654-7ccf9edbb73d",
    "userCrn": "crn:altus:iam:us-west-1:9d74eee4-1cad-45d7-b654-7ccf9
edbb73d:user:c44ac52c-625b-410c-a46c-8db204de4d92",
    "internalState": "{ATLAS_ENTITY_AUDIT_EVENTS_TABLE=SUCCESSFUL, EDGE_IND
EX_COLLECTION=SUCCESSFUL, DATABASE=SUCCESSFUL, FULLTEXT_INDEX_COLLECTION=SUC
CESSFUL, ATLAS_JANUS_TABLE=SUCCESSFUL, RANGER_AUDITS_COLLECTION=SUCCESSFUL,
VERTEX_INDEX_COLLECTION=SUCCESSFUL}",
    "status": "SUCCESSFUL",
```

```
    "startTIme": "2021-04-20 20:10:16.567"
    "endTIme": "2021-04-20 20:32:22.012"
    "backupLocation":"s3a://acme-finance-admin-bucket/backup-archive
/backup-archive",
    "backupName": "pre-upgrade0420"
    "failureReason": ""
}
```

Listing metadata for all backups

Use the following command to show all Data Lake backups in a given storage location:

```
$ cdp datalake list-datalake-backups
      --datalake-name <name>
    [--cli-input-json <string>]
    [--generate-cli-skeleton]
```

where the options are the following:

Option	Example	Description
--datalake-name <name>	finance-dl	This is the name of the Data Lake as configured in the CDP environment.
--cli-input-json <string>		Performs service operation based on the JSON string provided. The JSON string follows the format provided by --generate-cli-skeleton. If other arguments are provided on the command line, the CLI values will override the JSON-provided values.
--generate-cli-skeleton		Prints a sample JSON configuration file to standard output. If this argument is specified, only the template is produced: the list-backup command does not run.

For example:

```
$ cdp datalake list-datalake-backups
          --datalake-name finance-dl
```

The output of the command shows the metadata for all backups stored for this Data Lake. Note that if a backup is listed with status failed, it cannot be restored.

```
{
    "accountId": "9d74eee4-1cad-45d7-b654-7ccf9edbb73d",
    "userCrn": "crn:altus:iam:us-west-1:9d74eee4-1cad-45d7-b654-7ccf9
edbb73d:user:c44ac52c-625b-410c-a46c-8db204de4d92",
    "backupId": "415927d9-9f7d-4d42-8000-71630e5938ca",
    "internalState": "{ATLAS_ENTITY_AUDIT_EVENTS_TABLE=SUCCESSFUL, EDGE_IND
EX_COLLECTION=SUCCESSFUL, DATABASE=SUCCESSFUL, FULLTEXT_INDEX_COLLECTION=SUC
CESSFUL, ATLAS_JANUS_TABLE=SUCCESSFUL, RANGER_AUDITS_COLLECTION=SUCCESSFUL,
VERTEX_INDEX_COLLECTION=SUCCESSFUL}",
    "status": "SUCCESSFUL",
    "startTIme": "2021-04-20 20:10:16.567"
    "endTIme": "2021-04-20 20:32:22.012"
    "backupLocation":"s3a://acme-finance-admin-bucket/backup-archive
/backup-archive",
    "backupName": "pre-upgrade0420"
    "failureReason": "null"
}
{
    "accountId": "9d74eee4-1cad-45d7-b654-7ccf9edbb73d",
```

```
    "userCrn": "crn:altus:iam:us-west-1:9d74eee4-1cad-45d7-b654-7ccf9
edbb73d:user:c44ac52c-625b-410c-a46c-8db204de4d92",
    "backupId": "6543de7d-8d22-23e4-9123-54375ec123b4",
    "internalState": "{ATLAS_ENTITY_AUDIT_EVENTS_TABLE=SUCCESSFUL, EDGE_INDE
X_COLLECTION=SUCCESSFUL, DATABASE=SUCCESSFUL, FULLTEXT_INDEX_COLLECTION=SUCC
ESSFUL, ATLAS_JANUS_TABLE=SUCCESSFUL, RANGER_AUDITS_COLLECTION=SUCCESSFUL, V
ERTEX_INDEX_COLLECTION=SUCCESSFUL}",
    "status": "SUCCESSFUL",
    "startTIme": "2021-04-19 20:09:41.341"
    "endTIme": "2021-04-19 20:28:22.822"
    "backupLocation":"s3a://acme-finance-admin-bucket/backup-archive
/backup-archive",
    "backupName": ""
    "failureReason": "null"
}...
```

Troubleshooting Data Lake backup operations

Possible issues with Data Lake backups and suggested resolutions.

"failureReason": "[Gateway Timeout]"

This probably caused by a network or process timeout issue. If this doesn't resolve itself after a few minutes, check the messages at the environment and Data Lake levels to make sure there's not some larger issue happening.

"failureReason": "[HBase service HBase does not have a running Master.]"

This happens when the HBase service is not running. Check the HBase service page in Cloudera Manager to resolve any problems and restart the service.

"failureReason": "[Unable to get user data from UMS for CRN...

```
"failureReason": "[Unable to get user data from UMS for CRN crn:altus:iam:us
-west-1:7d24tin4-1ced-47d2-v375-8ccf3ndjj71d:user:7e4e753v-8n6t-4bj6-49op-g6
0894bc063y]"
```

This error appears when the user is not authorized to start a backup. The user needs to be an environment admin role. (Go to User Management in the Management Console)

"failureReason": "Failed to backup core=ranger_audits_[...]

Disposition: / Status:
CDPSDX-1995

```
(cdpclienv) [Wed Jun  3 16:32:27 CDT 2020 - smith@smith-7681-mbp15:/Users/sm
ith/git]$cdp datalake backup-datalake --datalake-name smith-dr-7  --backup-l
ocation hdfs://smith-dr-7-master0.smith.xcu2-8y8x.dev.cldr.work:8020/smithba
ck --backup-name "smith-test-7-6"
{
    "accountId": "7d24tin4-1ced-47d2-v375-8ccf3ndjj71d",
    "backupId": "32732sa2-1c95-4e33-a957-16d7fb645807",
    "internalState": "{ATLAS_ENTITY_AUDIT_EVENTS_TABLE=IN_PROGRESS, EDGE_IN
DEX_COLLECTION=IN_PROGRESS, DATABASE=SUCCESSFUL, FULLTEXT_INDEX_COLLECTION=I
N_PROGRESS, ATLAS_JANUS_TABLE=IN_PROGRESS, RANGER_AUDITS_COLLECTION=IN_PROGR
ESS, VERTEX_INDEX_COLLECITON=IN_PROGRESS}",
    "status": "IN_PROGRESS",
    "startTime": "2020-06-04 14:35:38.195",
    "endTime": "Thu Jun 04 14:35:47 GMT 2020",
```

```
    "backupLocation": "hdfs://smith-dr-7-master0.smith.xcu2-8y8x.dev.cldr.wo
rk:8020/smithback/",
    "failureReason": ""
}
(cdpclienv) [Thu Jun  4 09:35:47 CDT 2020 - smith@smith-7681-mbp15:/Users/
smith/git]$cdp datalake backup-datalake-status --datalake-name smith-dr-7
--backup-name "smith-test-7-6"
{
    "accountId": "7d24tin4-1ced-47d2-v375-8ccf3ndjj71d",
    "userCrn": "crn:altus:iam:us-west-1:7d24tin4-1ced-47d2-v375-8ccf3ndjj7
1d:user:7e4e753v-8n6t-4bj6-49op-g60894bc063y",
    "internalState": "{ATLAS_ENTITY_AUDIT_EVENTS_TABLE=SUCCESSFUL, EDGE_I
NDEX_COLLECTION=SUCCESSFUL, FULLTEXT_INDEX_COLLECTION=SUCCESSFUL, DATABASE=S
UCCESSFUL, RANGER_AUDITS_COLLECTION=FAILED, ATLAS_JANUS_TABLE=SUCCESSFUL, VE
RTEX_INDEX_COLLECITON=SUCCESSFUL}",
    "status": "FAILED",
    "startTime": "2020-06-04 14:35:38.195",
    "endTime": "2020-06-04 14:35:58.833",
    "backupLocation": "hdfs://smith-dr-7-master0.smith.xcu2-8y8x.dev.cldr.
work:8020/smithback/",
    "failureReason": "Failed to backup core=ranger_audits_shard1_replica_
n21 because org.apache.solr.core.SolrCoreInitializationException: SolrCore '
ranger_audits_shard1_replica_n21' is not available due to init failure: org.
apache.hadoop.ipc.RemoteException(org.apache.hadoop.ipc.RetriableException):
 NameNode still not started\n\tat org.apache.hadoop.hdfs.server.namenode.Nam
eNodeRpcServer.checkNNStartup(NameNodeRpcServer.java:2239)\n\tat org.apache.
hadoop.hdfs.server.namenode.NameNodeRpcServer.setSafeMode(NameNodeRpcServer.
java:1225)\n\tat org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolSer
verSideTranslatorPB.setSafeMode(ClientNamenodeProtocolServerSideTranslatorPB
.java:853)\n\tat org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtoco
lProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolPr
otos.java)\n\tat org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcI
nvoker.call(ProtobufRpcEngine.java:528)\n\tat org.apache.hadoop.ipc.RPC$Serv
er.call(RPC.java:1070)\n\tat org.apache.hadoop.ipc.Server$RpcCall.run(Server
.java:984)\n\tat org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:912)\n
\tat java.security.AccessController.doPrivileged(Native Method)\n\tat javax.
security.auth.Subject.doAs(Subject.java:422)\n\tat org.apache.hadoop.securit
y.UserGroupInformation.doAs(UserGroupInformation.java:1876)\n\tat org.apache
.hadoop.ipc.Server$Handler.run(Server.java:2882)\n"
}
```

When the Data Lake is shut down and then restarted, sometimes the Solr service starts incorrectly, causing backups and restore to fail.

To resolve, restart Solr service after the Data Lake is started.

{{Failed to check the existance of s3a://eng-sdx-datalake/smith-perf-1/backup_01/. Is it valid?}} or Could not verify the existence of s3a://eng-sdx-datalake/smith-perf-1/backup_01/ -- Is it accessible?

These error messages can be caused by either a permissions issue (the Data Lake/environment does not have access to the bucket), or you have designated a bucket that does not exist.

"Solr failure: Could not find a backup repository with name ..."

If you receive this error and you are attempting a backup on Runtime version 7.2.0 or earlier, then backup and restore operations are not supported on your current version.

Configuring and running Data Lake restore

Data Lake restore replaces the Data Lake data content: the metadata managed by each of the Data Lake services.

About this task

You may need to restore a Data Lake backup when:

- You are migrating Data Lake content to a new environment.
- A Data Lake repair fails.
- You need to delete and re-create a Data Lake.
- An upgrade fails or needs to be rolled back.

The Data Lake restore removes existing data stores and recreates them from the specified backup. This includes dropping database tables, dropping HBase tables, and deleting Solr collections.

> **Note:** When using the Data Lake backup and restore system, you should avoid using backups from other sources: the backup operation minimizes inconsistencies among service metadata. If data is restored from other sources such as an independent database backup, the restore operation can't guarantee consistency across the Data Lake metadata.

The system checks to make sure there isn't another backup or restore in progress.

Before you begin

There is downtime when a Data Lake restore is performed, as some Data Lake services will be stopped. Additionally, access to the HMS/Ranger databases will be blocked for the duration of the restore. Do not attempt to run workloads when you are running Data Lake restore.

For AWS:

- Apply the IAM policy for Data Lake restore to the following roles:

    - DATALAKE_ADMIN_ROLE
    - RANGER_AUDIT_ROLE
    - LOG_ROLE

        For more informaiton on IAM roles, see *Minimal setup for cloud storage*.

        In the IAM policy for Data Lake restore, be sure to replace the <your-backup-bucket> variable with the backup location used.

For Azure:

- Verify that the following identities have the "Storage Blob Data Contributor" role on the container where the backup is stored:

    - Data Lake Admin identity
    - Ranger Audit Logger identity

For GCP:

Verify that the Logger Service account has the following required permissions:

- storage.buckets.get
- storage.objects.create
- storage.objects.get
- storage.objects.list

Verify that the Ranger Audit Service account has the following required permissions:

- resourcemanager.projects.get
- resourcemanager.projects.list
- storage.buckets.get
- storage.objects.get
- storage.objects.getIamPolicy

Note that the Ranger Audit service account permissions listed above should be granted to a custom role, not the default Storage Object Admin role.

> **Note:** It is highly recommended to use the same data bucket (AWS and GCP)/storage account/container (Azure) and IAM roles/identities in the new environment where the metadata is restored.
>
> If you want to use a different data bucket (AWS and GCP)/storage account/container (Azure) along with IAM Roles/Identities, additional steps are required:
>
> - Make sure the IAM Roles/Identities used in the new environment have read/write permissions on the data location used in the older environment, as well as the backup location.
> - If a different data bucket (AWS and GCP)/storage account/container (Azure) is used, understand that the new data will be stored in the new location provided and the existing data would still be in the older location, causing the data to be distributed in two different locations. This may not be desirable.

For Cloudera Data Warehouse:

If you are a CDW user restoring a Data Lake to a new environment, perform the following steps from the CDW UI before running a Data Lake restore:

1. Record details of your Database Catalogs and Virtual Warehouses, such as:

    - Number of Virtual Warehouses associated with each Database Catalog and their names
    - Configurations associated with each Virtual Warehouse and Database Catalog, especially the configurations which were customized

    These configurations are not backed up and restored during the Data Lake backup and restore process. When you activate CDW again in the new CDP environment, you must re-apply the CDW configurations.

2. Delete existing Virtual Warehouses and user-created Database Catalogs associated with your CDW environment. The meta-data for the Virtual Warehouses associated with the default Database Catalog are preserved. The data for your tables would be preserved, as long as the cloud storage locations like S3 buckets are intact.

3. Deactivate the CDW environment, which deletes the default Database Catalog.

After you perform the Data Lake restore, you can activate the CDW environment from the CDW UI and re-create any Virtual Warehouses.

Important considerations regarding CDW metadata and data after Data Lake restore:

For Virtual warehouses associated with Default Database Catalog:

- Metadata like databases, tables and views will be restored.
- Data associated with existing tables would be visible as long as the cloud storage locations, such as objects in S3 buckets, were not deleted/modified after steps 2 and 3 above.
- Query historys and saved queries in Hue or DAS would not be visible.
- Any customizations to Virtual Warehouse or Database Catalog configurations are not retained after restore.
- The Hive/Impala Runtime version in the Virtual Warehouse and Database Catalog would be the latest inline with the corresponding CDW version.

For Virtual warehouses associated with a non-default Database Catalog:

- No metadata or data would be restored/visible.

Internal Ranger users password reset

Disposition: / Status:
CDPSDX-2415

As part of the restore operation, the RDS database will be restored unless it is specifically excluded. Note that the passwords for internal Ranger users (admin, keyadmin, etc.) are stored in RDS. When the RDS database is restored, it will replace the passwords for all internal Ranger users with the password for that user that was saved when the backup was originally done. This does not impact users that log into Ranger via SSO.

The user accounts impacted include, but are not limited to:

- admin user

- keyadmin user
- tagsync user
- usersync user

After a restore, a user with Ranger admin access can log into the Ranger UI to update the passwords of these users if desired.

> ⚠ **Important:** Because the admin user is impacted by this behavior, at least one SSO account on the Data Lake being restored to should have Ranger administrative access, to prevent a potential loss of administrative access if for some reason the admin user's password is not known after the restore.

Procedure

1. Switch to a user account with environment admin role.
2. Restore the backup.

   Use the following command to restore a Data Lake backup:

   ```
   $ cdp datalake restore-datalake --datalake-name <name>
   ```

   where the options are the following:

Option	Example	Description
--datalake-name	finance-dl	Name of the Data Lake as configured in the CDP environment. Required.
--backup-id	415927d9-9f7d-4d42-8000-71630e5938ca	System-generated ID for the backup. If you don't know the ID, run the list-datalake-backups command to see the IDs. If neither an ID nor backup name is provided, the command restores the most recent successful backup operation. If both an ID and a backup name are provided and don't refer to the same backup, the backup specified by the ID is used.
--backup-name		User-supplied name for the backup. If a backup ID is provided, the backup name is not needed.
--include-database | --no-include-database	TRUE	DEPRECATED. The database is included in the restore by default. To skip it, use the –skip-database flag.
--skip-ranger-hms-metadata | --no-skip-ranger-hms-metadata		Skips the restore of the databases backing HMS/Ranger services. If this option is not provided, then by default the Atlas lineage will be restored if the backup used includes the Atlas lineage information.
--skip-atlas-metadata | --no-skip-atlas-metadata		Skips the restore of the Atlas metadata. If this option is not provided, then by default the Atlas metadata will be restored if the backup used includes the Atlas metadata.
--skip-ranger-audits | --no-skip-ranger-audits		Skips the restore of the Ranger audits. If this option is not provided, then by default the Ranger audits will be restored if the backup used includes the Ranger audits.
--skip-validation | --no-skip-validation		Using --skip-validation skips the validation that occurs before the backup process begins. This validation checks for required permissions that are often the source of backup/restore failures. See *Backup and restore for the Data Lake* for more details.

Option	Example	Description
--validation-only | --no-validation-only		--validation-only runs the pre-backup and restore validation process, but does not proceed to the actual backup/restore operation. *Backup and restore for the Data Lake*
--backup-location-override	s3a://acme-finance-admin-bucket/backup-archive /backup-archive	Backup location. When provided, will be used to lookup the backup. If provided, the –backup-id parameter is required.

```
$ cdp datalake restore-datalake
--datalake-name finance-dl
--backup-id 415927d9-9f7d-4d42-8000-71630e5938ca
--backup-name <value>]
--no-skip-ranger-hms-metadata
--no-skip-atlas-metadata
--no-skip-ranger-audits
--backup-location-override s3a://acme-finance-admin-bucket/backup-archive
/backup-archive
```

The output of the command shows the current status of the operation. Note the internal state shows the status of each separate restore operation. If any of the individual restore operations fail, the overall status is failed and the restoration is stopped (not transactional). If this happens, review and correct the failure and run the restore again.

```
{
    "accountId": "8g49gju4-4has-97h7-b391-7jre9edve47n",
    "restoreId": "f0gq74h7-3b13-477a-b07c-cb74v211b81c",
    "backup-id": 415927d9-9f7d-4d42-8000-71630e5938ca",
    "internalState": "{ATLAS_ENTITY_AUDIT_EVENTS_TABLE=IN_PROGRESS, EDGE
_INDEX_COLLECTION=IN_PROGRESS, FULLTEXT_INDEX_COLLECTION=IN_PROGRESS, ED
GE_INDEX_COLLECTION_DELETE=IN_PROGRESS, VERTEX_INDEX_COLLECITON_DELETE=I
N_PROGRESS, RANGER_AUDITS_COLLECTION_DELETE=IN_PROGRESS, ATLAS_JANUS_TAB
LE=IN_PROGRESS, RANGER_AUDITS_COLLECTION=IN_PROGRESS, VERTEX_INDEX_COLLE
CITON=IN_PROGRESS, FULLTEXT_INDEX_COLLECTION_DELETE=IN_PROGRESS}",
    "status": "IN_PROGRESS",
    "startTIme": "2021-04-21 10:30:01.022"
    "endTIme": ""
    "backupLocation":"s3a://acme-finance-admin-bucket/backup-archive
/backup-archive",
    "failureReason": "null"
}
```

What to do next
To see the status of the backup after the initial command, see *Checking the status of a Data Lake backup.*
Related Information

Backup and restore for the Data Lake

Minimal setup for cloud storage (AWS)

Minimal setup for cloud storage (Azure)

Showing Data Lake restore status

How to check the status of a Data Lake restore operation.

Check status of Data Lake restore

Use the following command to see the status of a Data Lake restore:

```
$ cdp datalake restore-datalake-status
      --datalake-name <name>
    [--cli-input-json <string>]
    [--generate-cli-skeleton]
```

where the options are the following:

Option	Example	Description
--datalake-name	finance-dl	This is the name of the Data Lake as configured in the CDP environment.
[--restore-id]	f0da74a9-3b22-477a-b07c-cb69b211b81c	ID for a specific restore, as reported in the output of the original restore command. If a restore ID is not specified, the command returns the status of the most recent restore operation.
--cli-input-json <string>		Performs service operation based on the JSON string provided. The JSON string follows the format provided by --generate-cli-skeleton.
		If other arguments are provided on the command line, the CLI values will override the JSON-provided values.
--generate-cli-skeleton		Prints a sample JSON configuration file to standard output. If this argument is specified, only the template is produced: the list-backup command does not run.

For example:

```
$ cdp datalake restore-datalake-status
         --datalake-name finance-dl
```

The output of the command shows the current status of the restore operation. Note the internal state shows the status of each separate backup operation. If any of the individual restore operations fail, the overall status is failed and the restoration is aborted. Note that the internal restore status lists operations for deleting the existing Solr collections in addition to the operations to restore the backed up collections.

```
{
    "accountId": "8y63idy3-2ygn-98h6-j630-7uie9renq93e",
    "restoreId": "f0da74a9-3b22-477a-b07c-cb69b211b81c",
    "backup-id": 415927d9-9f7d-4d42-8000-71630e5938ca",
    "userCrn": "crn:altus:iam:us-west-1:8y63idy3-2ygn-98h6-j630-7uie9renq93e
:user:c87db52v-639m-613g-j94w-8hy944hn4i64",
    "internalState": "{ATLAS_ENTITY_AUDIT_EVENTS_TABLE=SUCCESSFUL, EDGE_IND
EX_COLLECTION=SUCCESSFUL, FULLTEXT_INDEX_COLLECTION=SUCCESSFUL, EDGE_INDEX_C
OLLECTION_DELETE=SUCCESSFUL, VERTEX_INDEX_COLLECITON_DELETE=SUCCESSFUL, RANG
ER_AUDITS_COLLECTION_DELETE=SUCCESSFUL, ATLAS_JANUS_TABLE=SUCCESSFUL, RANGER
_AUDITS_COLLECTION=SUCCESSFUL, VERTEX_INDEX_COLLECITON=SUCCESSFUL, FULLTEXT_
INDEX_COLLECTION_DELETE=SUCCESSFUL}",
    "status": "SUCCESSFUL",
    "startTIme": "2021-04-21 10:30:01.022"
    "endTIme": "2021-04-21 11:22:22.055"
    "backupLocation":"s3a://acme-finance-admin-bucket/backup-archive
/backup-archive",
    "backupName": "pre-upgrade0420"
    "failureReason": "null"
}
```

Restoring to a RAZ Data Lake

You can restore a Data Lake backup from a non-RAZ Data Lake to a RAZ Data Lake, or from a RAZ Data Lake to a different RAZ Data Lake.

Best Practices

For best results, it's recommended to use the same Storage Location Base, Logs Location Base, and Backup Location Base between the source Data Lake and the destination RAZ Data Lake. After the restore, the Ranger policies will be replaced with the policies from the original Data Lake. This means that if the Storage Location Base, Log Location Base, and/or Backup Location Base are different between the source Data Lake and the destination Data Lake, the restored Ranger policies will reference the locations from the source Data Lake.

- If you intend to use the locations from the source Data Lake, make sure the roles associated with the destination RAZ Data Lake have sufficient permissions to access the original storage locations.
- If you intend to use the destination RAZ Data Lake locations, the Ranger policies (cm_s3 for AWS, cm_adls for Azure) will need to be updated to reference the correct storage locations after the restore.

Preparing the Data Lakes

Add a Ranger policy to allow the backups to be read from the original backup location. For the restore operation, only write permissions are not required. This must be done on both the original source Data Lake before the backup is taken, and on the RAZ destination Data Lake before the restore is done.

AWS

On the source Data Lake:

1. Open the Ranger UI.
2. Go to the cm_s3 policy list.

3. Add a new policy:

   a. Policy name: restore_to_raz
   b. S3 bucket: The bucket where the original backups were written
   c. Path: The path in the bucket where the original backups were written

Policy Type   **Access**

Policy Name *   restore_to_raz   ⓘ

Policy Label   Policy Label

S3 Bucket *   ✕ original-backup-bucket

Path *   ✕ /hreeve-dl/backups/

4. Add read permissions for the atlas, hbase, hdfs, and solr users under "Allow Conditions".

Select User	Permissions	Delegate Admin
✕ atlas ✕ hbase ✕ hdfs ✕ solr	Read ✏	☐

5. Repeat steps on the destination Data Lake.

Azure

Note: The source Data Lake and destination Data Lake must be configured to use the same Storage Account for the Log, Storage, and Backup Location Bases. They do not have to be configured to use the same Storage Container.

On the source Data Lake:

1. Open the Ranger UI.
2. Go to the cm_adls policy list.

3. Add a new policy:

    a. Policy name: restore_to_raz

    b. Storage Account: The storage account where backups will be stored.

    c. Storage Account Container: The container where backups will be stored.

    d. Path: The path(s) in the bucket where backup will be written.

        Note: If more than one storage container will be used for backup, create a separate policy for each container.

4. Add read and list permissions for the atlas, hbase, hdfs, and solr users under "Allow Conditions."

5. Repeat steps on the destination Data Lake.

Take the backup of the non-RAZ data lake

After you prepare the Data Lakes, see Configure backups for a Data Lake for instructions on running the backup.

Run the restore

After you take the Data Lake backup, see Restore Data Lake content for instructions on running the restore. Use the backup-id from the backup taken in the previous step.

After the restore

If the Storage Location Base, Logs Location Base, and Backup Locations Base are the same between the source Data Lake and the destination Data Lake, this section can be skipped.

After the restore, the Ranger policies will be replaced with the policies from the original Data Lake. This means that if the Storage Location Base, Log Location Base, and/or Backup Location Base are different between the original Data Lake and the RAZ Data Lake, the restored Ranger policies will reference the locations from the original Data Lake.

• If you intend to use the locations from the original Data Lake, make sure the roles associated with the RAZ Data Lake have sufficient permissions to access the original storage locations.
• If intending to use the RAZ Data Lake locations, the Ranger policies (cm_s3 for AWS, cm_adls for Azure) will need to be updated to reference the correct storage locations after the restore.

Troubleshooting Data Lake restore operations

Possible issues with Data Lake restore and suggested resolutions.

Principal services running during restore

The most likely errors in restoring data from backup is that a service is in a state that is incompatible with the restore. Principal services (see Principal services on page 49) must be stopped before running the restore. Dependent services (see Dependent services on page 49) must be running to allow the restore to recreate their data. The restore checks the status of the principal services; however, if one of the dependent services is stopped and cannot be accessed to perform the restore operation, the restore operation will fail.

"failureReason": "[Datalake database restore failed.]"

Disposition: / Status:
CDPSDX-2430

If the principal services are running on the datalake during a restore operation, restore will fail with the following error message:

```
{
    "accountId": "8y63idy3-2ygn-98h6-j630-7uie9renq93e",
    "restoreId": "7c5c92c7-e3d3-408c-b18f-03bcfe0c9369",
    "backupId": "003b9882-e2fa-4fcc-ae8f-528de176c668",
    "userCrn": "crn:altus:iam:us-west-1:8y63idy3-2ygn-98h6-j630-7uie9renq93e
:user:c87db52v-639m-613g-j94w-8hy944hn4i64",
    "internalState": "{ATLAS_ENTITY_AUDIT_EVENTS_TABLE=SUCCESSFUL, EDGE_IND
EX_COLLECTION=SUCCESSFUL, DATABASE=FAILED, FULLTEXT_INDEX_COLLECTION=SUCCESS
FUL, EDGE_INDEX_COLLECTION_DELETE=SUCCESSFUL, VERTEX_INDEX_COLLECITON_DELETE
=SUCCESSFUL, RANGER_AUDITS_COLLECTION_DELETE=SUCCESSFUL, RANGER_AUDITS_COLLE
CTION=SUCCESSFUL, ATLAS_JANUS_TABLE=SUCCESSFUL, VERTEX_INDEX_COLLECITON=SUCC
ESSFUL, FULLTEXT_INDEX_COLLECTION_DELETE=SUCCESSFUL}",
    "status": "FAILED",
    "startTime": "2020-08-28 18:27:54.11",
    "endTime": "2020-08-28 18:29:55.507",
    "backupLocation": "s3a://eng-sdx-daily-datalake/smith-br-1/backup_01/",
    "failureReason": "[Datalake database restore failed.]"
}
```

To correct this scenario, stop the principal services and re-run the restore-datalake operation.

Failed restore renders Data Lake inoperable

Disposition: / Status:
CDPSDX-2417

If the restore operation fails, the Data Lake will be rendered inoperable. A restore-datalake operation must be re-run and complete successfully for the Data Lake to re-gain functionality

Related Information

Backup and restore for the Data Lake

Data Lake resizing

Data Lake resizing is the process of scaling up a light duty or medium duty Data Lake to the medium duty or enterprise form factor, which have greater resiliency than light duty and can service a larger number of clients. You can trigger the resize in the CDP UI or through the CDP CLI. As part of Data Lake resizing via CDP CLI, you can also resize from single-AZ to multi-AZ.

Overview

During a typical Data Lake scaling operation, the metadata maintained in the Data Lake services is automatically backed up, a new enterprise or medium duty Data Lake is created within the environment, and the Data Lake metadata is automatically restored to the new enterprise or medium duty Data Lake.

As part of the Data Lake resizing, you can optionally resize an existing single availability zone (single-AZ) Data Lake to a multiple availability zone (multi-AZ) Data Lake. To resize your Data Lake from single to multi-AZ, add the --multi-az flag to the Data Lake resize command.

Supportability matrix

The following table illustrates your Data Lake resizing options:

	Source			Target		Supported?
Runtime version	**Scale**	**Deployment**		**Scale**	**Deployment**	
7.2.16 and prior	Light	SingleAZ		Medium	SingleAZ	Yes
7.2.16 and prior	Light	SingleAZ		Medium	MultiAZ	Yes
7.2.16 and prior	Light	SingleAZ		Enterprise	Any	No
7.2.16 and prior	Medium	Any		Enterprise	Any	No
7.2.17+	Light	SingleAZ		Medium	Any	No
7.2.17+	Light/Medium	SingleAZ		Enterprise	SingleAZ	Yes
7.2.17+	Light/Medium	SingleAZ		Enterprise	MultiAZ	Yes
7.2.17+	Medium	MultiAZ		Enterprise	SingleAZ	No
7.2.17+	Medium	MultiAZ		Enterprise	MultiAZ	Yes

Note:

Note that:

- Resizing is only supported for Cloudera Runtime versions 7.2.7 and above, because medium duty Data Lakes are not supported for earlier versions.
- Resizing from a light or medium duty Data Lake to an enterprise Data Lake is supported only for Data Lakes on Runtime versions 7.2.17 and above, because enterprise Data Lakes are supported only on Runtime versions 7.2.17+.
- On Runtime versions 7.2.17 and above, you can only resize to an enterprise Data Lake, because medium duty Data Lakes were deprecated in Runtime 7.2.17.
- If you need to resize a light duty Data Lake on Runtime 7.2.16 or earlier, it can only be resized to a medium duty Data Lake.

Note:

RAZ-enabled Data Lakes are currently eligible for automatic restore during a resizing operation only if you are resizing:

- An AWS Data Lake on Cloudera Runtime version 7.2.15+
- An Azure Data Lake on Cloudera Runtime version 7.2.16+

For older Runtime versions, the Data Lake will be automatically backed up, but must be manually restored after the resizing is complete. If RAZ is in use on a Runtime version that is ineligible for automatic restore, before you start the Data Lake backup, make sure that the appropriate Ranger policy exists with access to the backup location in the cloud. See instructions for manually restoring a RAZ-enabled Data Lake here.

Resizing a Data Lake is supported with all data services.

Before you begin, note the following:

- The resizing operation requires an outage and should be performed during a maintenance window. No metadata changes may occur during the resizing, as these changes will no longer be present once the resizing operation completes (the previously backed up metadata is being restored). Suspend any operations that may result in any SDX metadata change during the resizing operation.
- Data Hub clusters should be stopped before the resizing operation begins. For any cluster that cannot be stopped, stop all of the services on the Data Hub through the Cloudera Manager UI.
- With CDF 2.0 or lower, some flows must be re-created after a resizing operation.

Limitations

1. **Disposition: / Status:**
   CB-24993 CM customizations

   Cloudera Manager configurations are not retained when a Data Lake is resized (they are lost when a new Data Lake cluster is created as part of backup and restore operation). Therefore, prior to performing a resize you should note all the custom Cloudera Manager configurations of your Data Lake and then once the resizing operation is completed, reapply them.

2. **Disposition: / Status:**
   DOCS-20058 Document a limitation that will be removed in Cb. 2.83

   If a Data Lake has been vertically scaled, the following limitations apply:

   - If Data Lake VM instances are vertically scaled using runbooks or via vertical scaling, they will return to default types after resizing.
   - If Data Lake storage disks are vertically scaled using runbooks or via vertical scaling, they will return to default sizes after resizing.
   - If the storage and/or image type of the remote database are resized using the cloud provider console, they will fall back to defaults after resizing.

   As a workaround, you should perform vertical scaling after the resize is performed.

3. **Disposition: / Status:**
   CB-24993 Added this limitation based on feedback

   If resizing from Medium Duty to Enterprise Data Lake, you must be on Runtime 7.2.17 before attempting the resize.

4. If you would like to resize an existing single availability zone (single-AZ) Data Lake to a multiple availability zone (multi-AZ) Data Lake, the following limitations apply:

   - The AZ resizing functionality is currently available for AWS only, as CDP does not yet support multi-AZ for Azure and GCP.
   - Existing Data Hubs attached to the Data Lake are not resized to multi-AZ as part of the Data Lake resizing process.
   - The single to multi-AZ resizing is only available when resizing a Data Lake via CDP CLI. The single-AZ to multi-AZ resizing is not available via the Data Lake resizing option in the CDP web interface.

Prerequisites

Prior to resizing the Data Lake, ensure that the following are in place:

1. The Data Lake must be running to perform the resizing operation.
2. For RAZ-enabled Data Lakes, update the appropriate Ranger policy to give the backup and restore feature permission to access the backup location in the cloud. See instructions for configuring RAZ for backup here.
3. Make sure that Atlas is up to date and has processed all the lineage data in Kafka. To do this, follow the steps in Checking that Atlas is up-to-date. If Atlas is not up to date, lineage/audit information in Kafka that is not processed by Atlas will be lost.
4. If you are using CDW, you must upgrade to version 1.4.1 or higher before you can resize the Data Lake. Determine the CDW version you are on by clicking edit on the environment:

5. If you are using CDW, stop the virtual warehouses and data catalogs associated with the environment.
6. If you are using CDE, do one of the following:

   a. Upgrade to CDE 1.15, or
   b. Create new service.

      1. Take a backup of your jobs following Backing up Cloudera Data Engineering jobs.
      2. Create a new DE service and virtual cluster.
      3. Restore the jobs following the instructions in Restoring Cloudera Data Engineering jobs from backup.
7. If you are using CML:

   a. Backup CML workspaces (AWS only). If backup is not supported, then proceed to the next step.
   b. Suspend CML workspaces. If the suspend capability is not available, follow the steps in Refreshing CML governance pods after resizing the Data Lake.

Checking that Atlas is up-to-date

Follow the steps below to ensure that Atlas is up-to-date and has processed all the lineage data in Kafka.

Procedure

1. SSH into the master node of your light duty Data Lake.

2. Switch to the super user for the node by running sudo   su.

3. Copy over the following script into a file called check_atlas_updated.sh:

```
#!/usr/bin/env bash
# Determine Atlas keytab path.
ATLAS_KT=$(find / -wholename "*atlas-ATLAS_SERVER/atlas.keytab" 2>/dev/n
ull | head -n 1)

# Setup required configuration files if needed.
if [[ ! -f jaas.conf ]]; then
 ATLAS_PRINCIPAL=$(klist -kt "${ATLAS_KT}" | grep -o -m 1 "atlas\/\S*")
 printf "KafkaClient {
 \tcom.sun.security.auth.module.Krb5LoginModule required
 \tuseKeyTab=true
 \tkeyTab=\"%s\"
 \tprincipal=\"%s\";\n};\n" "${ATLAS_KT}" "${ATLAS_PRINCIPAL}" > jaas.conf
fi

if [[ ! -f client.config ]]; then
 printf "security.protocol=SASL_SSL\nsasl.kerberos.service.name=kafka\n" >
 client.config
fi
# Determine the Kafka bootstrap server to use.
KAFKA_SERVER=$(grep --line-buffered -oP "atlas.kafka.bootstrap.servers=\
K.*" \
 /etc/atlas/conf/atlas-application.properties | awk -F',' '{print $1}')

# Export Kafka-specific environment variables.
export KAFKA_HEAP_OPTS="-Xms512m -Xmx1g"
export KAFKA_OPTS="-Djava.security.auth.login.config=${PWD}/jaas.conf"

# Kinit into Atlas keytab as Atlas user.
kinit -kt "$ATLAS_KT" "atlas/$(hostname -f)" 2>/dev/null

# Obtain Atlas lineage information.
LINEAGE_INFO=$(/opt/cloudera/parcels/CDH/lib/kafka/bin/kafka-consumer-gro
ups.sh \
 --bootstrap-server "${KAFKA_SERVER}" --describe --group atlas \
 --command-config="${PWD}/client.config" 2>/dev/null \
 | awk '{print $2, $6}')

if [[ -z "$LINEAGE_INFO" ]]; then
 echo "*ERROR*: Unable to get lineage info for Atlas. Please look at the
 created configuration files to make sure they look correct."
 exit 1
fi

# Parse lineage information and determine if Atlas is out of date.
LINEAGE_LAG_VALS=($LINEAGE_INFO)
NUM_LAG_VALS=${#LINEAGE_LAG_VALS[@]}
OUT_OF_DATE_TOPICS=""
for (( i = 2; i < ${NUM_LAG_VALS}; i += 2 )); do
 if [[ ${LINEAGE_LAG_VALS[${i} + 1]} != '-' && ${LINEAGE_LAG_VALS[${i} +
1]} != '0' ]]; then
     OUT_OF_DATE_TOPICS="${OUT_OF_DATE_TOPICS}${LINEAGE_LAG_VALS[$i]}, "
 fi
done

if [[ -z "$OUT_OF_DATE_TOPICS" ]]; then
 echo "Atlas is up to date! Feel free to continue with the migration."
```

```
else
 echo "The following Atlas topics are not up to date: ${OUT_OF_DATE_TOPICS
%??}!"
 echo "Please wait until Atlas is entirely up to date before continuing
with the migration."
fi
```

4. Allow the new script to be run by running chmod +x   check_atlas_updated.sh

5. Run the script with ./check_atlas_updated.sh. The script will tell you if Atlas is up to date or not. If it isn't, wait a while and check again. You should only begin the resizing process if the script tells you that Atlas is up to date.

Resizing the Data Lake through the CDP UI

You can resize a Data Lake from light or medium duty to medium duty or enterprise through the CDP UI.

About this task

Required role: EnvironmentAdmin or Owner of the environment

Before you begin

Disposition: / Status:
CB-24993 CM customizations

Cloudera Manager configurations are not retained when a Data Lake is resized (they are lost when a new Data Lake cluster is created as part of backup and restore operation). Therefore, prior to performing a resize you should note all the custom Cloudera Manager configurations of your Data Lake and then once the resizing operation is completed, reapply them.

Procedure

1. Stop all of the attached Data Hub clusters that can be stopped, to make sure that there are no changes to HMS metadata during the resizing operation. For any cluster that cannot be stopped, stop all of the services on the Data Hub through the Cloudera Manager UI.

2. Verify that the DATALAKE_ADMIN_ROLE, RANGER_AUDIT_ROLE, and LOG_ROLE have read/write permissions to the backup location. See the Data Lake backup and restore documentation for more information on these permissions. LOG_ROLE is specific to Data Lake restore.

3. In the CDP UI, click Data Lakes and select the Data Lake that you want to resize.

4. Click Resize.

You will be asked to confirm that you want to resize the Data Lake, after which the resizing process will begin. The resizing operation is finished when the Data Hub clusters have been automatically refreshed, which happens

---

74

after the original Data Lake has been deleted. Check the Event History to verify that the Data Hubs have been refreshed.

5. RAZ-enabled Data Lakes are currently eligible for automatic restore during a resizing operation only if you are resizing:

- An AWS Data Lake on Cloudera Runtime version 7.2.15+
- An Azure Data Lake on Cloudera Runtime version 7.2.16+

For older Runtime versions, the Data Lake will be automatically backed up, but you must manually restore the Data Lake after the resizing is complete. If RAZ is in use on a Runtime version that is ineligible for automatic restore, before you start the Data Lake backup, make sure that the restore_to_raz policy Ranger policy exists with access to the backup location in the cloud. See instructions for manually restoring a RAZ-enabled Data Lake here.

Resizing the Data Lake through the CDP CLI

You can resize a Data Lake from light or medium duty to medium duty or enterprise through the CDP CLI. As part of Data Lake resizing via CDP CLI, you can also resize from single-AZ to multi-AZ.

About this task

Required role: EnvironmentAdmin or Owner of the environment

Before you begin

Disposition: / Status:
CB-24993 CM customizations

Cloudera Manager configurations are not retained when a Data Lake is resized (they are lost when a new Data Lake cluster is created as part of backup and restore operation). Therefore, prior to performing a resize you should note all the custom Cloudera Manager configurations of your Data Lake and then once the resizing operation is completed, reapply them.

Procedure

1. Stop all of the attached Data Hub clusters that can be stopped, to make sure that there are no changes to HMS metadata during the resizing operation. For any cluster that cannot be stopped, stop all of the services on the Data Hub through the Cloudera Manager UI.

2. Verify that the DATALAKE_ADMIN_ROLE, RANGER_AUDIT_ROLE, and LOG_ROLE have read/write permissions to the backup location. See the Data Lake backup and restore documentation for more information on these permissions. LOG_ROLE is specific to Data Lake restore.

3. To trigger resizing from the CDP CLI, run the cdp datalake resize-datalake command. For example:

```
cdp datalake resize-datalake --datalake-name <mydatalake> --target-size
MEDIUM_DUTY_HA
```

Option	Description
--datalake-name	Name or CRN of the Data Lake that you want to upscale.
--target-size	MEDIUM_DUTY_HA or ENTERPRISE

Use the cdp datalake resize-datalake command with the --multi-az flag to resize your Data Lake from single-AZ medium duty to enterprise multi-AZ:

```
cdp datalake resize-datalake \
   --datalake-name <VALUE> \
   --target-size <VALUE> \
```

```
--multi-az
```

If the source Data Lake is multi-AZ, the --multi-az flag is ignored.

4. Monitor the Event History. The resizing operation is finished when the Data Hub clusters have been automatically refreshed, which happens after the original light duty Data Lake has been deleted. Check the Event History to verify that the Data Hubs have been refreshed.

5. RAZ-enabled Data Lakes are currently eligible for automatic restore during a resizing operation only if you are resizing:

   • An AWS Data Lake on Cloudera Runtime version 7.2.15+
   • An Azure Data Lake on Cloudera Runtime version 7.2.16+

   For older Runtime versions, the Data Lake will be automatically backed up, but you must manually restore the Data Lake after the resizing is complete. If RAZ is in use on a Runtime version that is ineligible for automatic restore, before you start the Data Lake backup, make sure that the restore_to_raz policy Ranger policy exists with access to the backup location in the cloud. See instructions for manually restoring a RAZ-enabled Data Lake here.

Resizing post-requisites

Complete the following tasks after you resize a Data Lake.

Procedure

1. If RAZ is not being used, resync the IDBroker mappings to the Data Lake.

2.
Disposition: / Status:
CB-24993 CM customizations

   Reapply your custom Cloudera Manager configurations.

3. Start the Data Catalogs and Virtual Warehouses. For each virtual warehouse, Cloudera recommends that you start, stop, and start again. This will completely refresh the Data Lake details for the virtual warehouse.

4. Start the Data Hub cluster services if you stopped them before the resizing operation. Data Hubs that were stopped before the resizing operation should continue to work when the resizing completes, by communicating with the new Data Lake automatically when they are re-started.

5. Resize up the FreeIPA cluster. See Resize FreeIPA.

6. With CDF 2.1 or higher, the steps below are sufficient. For older CDF versions, you must re-create the impacted flows.
   a) An alert suggesting restart of the flow is triggered in the Data Flow service.
   b) Restart the Flows.
   c) After restart, the flows should start working with the resized Data Lake. Not all CDP Flows will be impacted by a resized Data Lake. Only those Flows that have a dependency on the Data Lake will be alerted.

7. Resume the CML workspaces.

Recovering after a failed resizing operation

Recover from a failed resizing operation using the recovery command in the CDP CLI.

You can recover from a failed resizing operation by returning a Data Lake to its original state before the resize operation was started. Be sure that this is what you want to do before proceeding.

Data Lake recovery simply reattaches and starts the original light duty Data Lake in the environment. All of the instances, disks, and databases are unchanged from their original state.

If recovery cannot be started, or fails for any reason, reach out to the Cloudera support team, who can manually recover your Data Lake.

Trigger the recovery command through the CDP CLI:

```
cdp datalake recover-datalake --datalake-name <mydatalake>
```

Refreshing CML governance pods

If backing up and suspending CML workspaces is not possible, refresh CML governance pods after resizing a Data Lake.

Procedure

1. Generate kubeconfig for CML workspace for remote access (instructions here).

2. Configure CML kubeconfig to access the workspace (see Kubernetes documentation).

3. Run the following commands to restart goverance pods in CML:

```
kubectl scale deployments governance --replicas 0 -n mlx
#wait for 30 sec
kubectl scale deployments governance --replicas 1 -n mlx
```

Managing certificates

There are two types of certificates within CDP that you must manage: public and private, also called host certificates.

Disposition: / Status:
This topic exists in both the Data Hub and Data Lake library. If you update this topic, make sure changes are cherry-picked to the other branch as well, assuming the changes apply to both. Currently these topics are not profiled in any way.

- Public certificates are Let's Encrypt-issued certificates for Data Hub and Data Lake clusters. These certificates are available on port 443 (HTTPS) of the cluster and are responsible for enabling TLS in front of Knox and other available services on that port. They are valid for 90 days, and in most circumstances CDP will renew these certificates automatically before they expire.

  Note the following limitations in regards to automatic renewal of public certificates:

  - Data Hub or Data Lake clusters created on or after March 7, 2022 are eligible for automatic renewal of public certificates. Clusters created before March 7, 2022, must be renewed manually once following the instructions in *Manually renewing public certificates for Data Lake and Data Hub clusters*. After the public certificate for a cluster has been manually renewed once from the CDP UI or CLI, it is eligible for automatic certificate renewal in the future.
  - If an automatic renewal fails, the renewal service will retry the renewal for three consecutive days or three attempts. Any cluster that cannot be renewed by these retry attempts must be renewed manually through the CDP UI or CLI.
  - The automatic renewal is tried three times: on the 69th, 72nd and 78th day after the certificate creation date. For example, if a certificate is getting expired on September 24th, 2022, the renewal will be tried in the following sequence:

    - First renewal: September 3rd, 2022 2:00 A.M.
    - Second renewal: September 6th, 2022 2:00 A.M.
    - Third renewal: September 12th, 2022 2:00 A.M.

   In case the renewal is successful on the first attempt, the renewal will not be tried again.
  - Renewal of the certificates happens at 2 A.M. of the Control Plane time. If the Control Plane is in the United States region, the renewal starts at 2 A.M. Pacific Daylight Time (PDT). If the Control Plane is in the

European region, the renewal starts at 2 A.M. Central European Summer Time (CEST). If the Control Plane is in the East-Asian and Pacific region, the renewal starts at 2 A.M. Australian Eastern Standard Time (AEST).

- The auto renewal service does not know the status of the cluster. If the cluster is down or performing another operation, the automatic renewal may fail and you should initiate the renewal from the UI or CLI manually. Certificate renewal will not happen if the Data Hub and Data Lake clusters or the Public Cloud environment has a Stopped state.
- If the cluster is down during the renewal attempts and comes back up after the renewal retries are exhausted, automatic renewal will not happen for that cluster. The certificate has to be renewed manually from the UI or CLI.
- If a public certificate expires, you'll receive a warning that your connection is not secure when you attempt to access a Data Lake or Data Hub cluster through the CDP UI.

See *Manually renewing public certificates for Data Lake and Data Hub clusters* for instructions on renewing the public certificates manually.

- Private certificates, or host certificates, are certificates created during cluster provisioning for every host with Auto-TLS. Private/host certificates have a default expiration date of one year. As private certificates get closer to expiration, the CDP UI displays a warning that the certificate is about to expire.
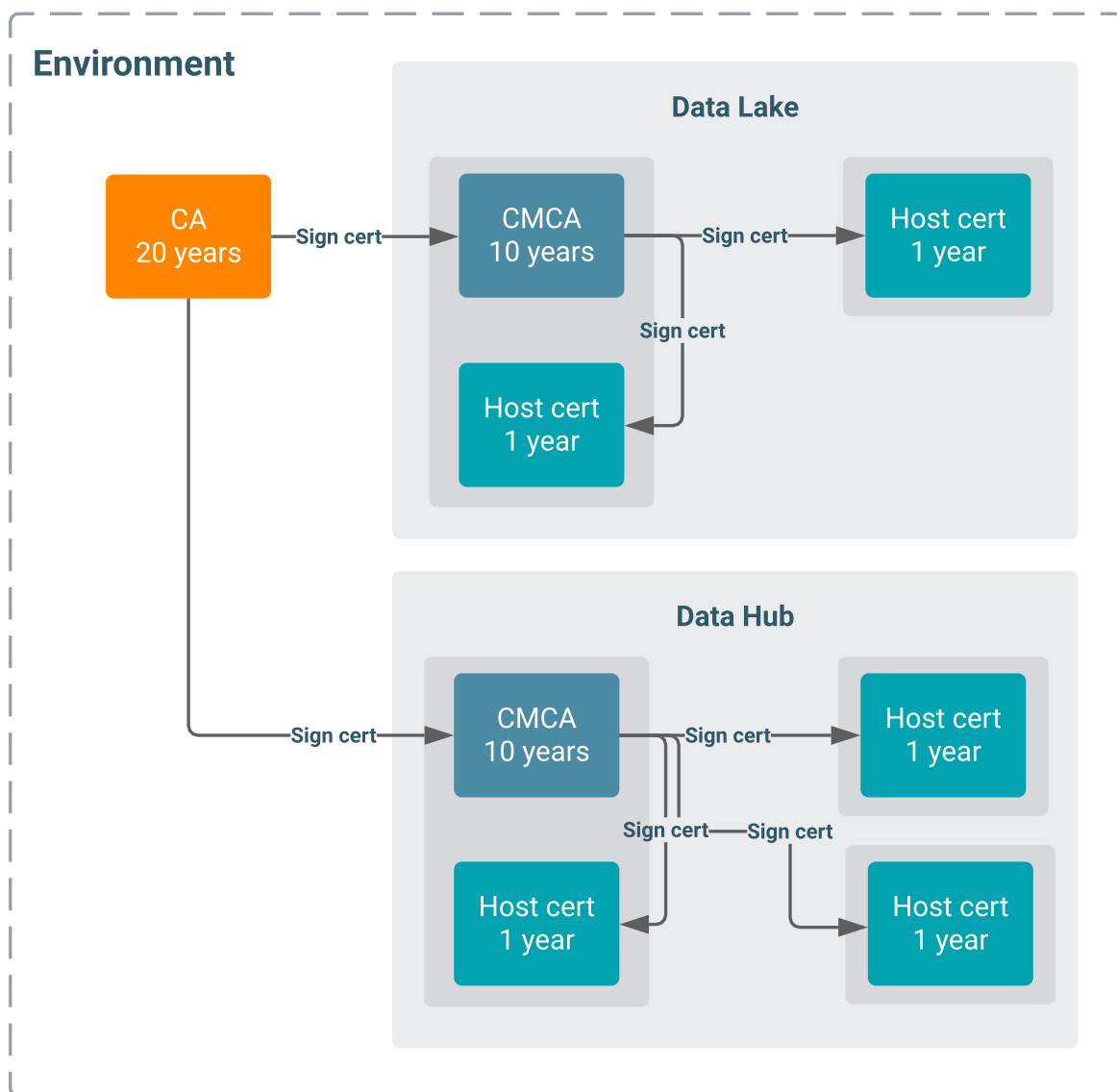
Though the CDP UI displays a warning about the expiration of private/host certificates, you are still responsible for renewing them through the UI or CDP CLI. After the certificates expire, the cluster is not functional, so you must renew them before expiration.

Renewing private/host certificates on Data Lake and Data Hub clusters

Private (host) certificates have a default expiration date of one year; to keep the Data Lake and Data Hub clusters running, you must renew the host certificates before they expire.

About this task

Required role (Data Lakes): EnvironmentAdmin or Owner of the environment

Required role (Data Hub): DatahubAdmin, Owner of the Data Hub, EnvironmentAdmin, or Owner of the environment

Disposition: / Status:
This topic exists in both the Data Hub and Data Lake library. If you update this topic, make sure changes are cherry-picked to the other branch as well, assuming the changes apply to both. Currently these topics are not profiled in any way.

During cluster provisioning, Cloudera Manager creates an intermediate certificate (CMCA) signed by FreeIPA CA. The CMCA is used to create certificates for every host with Auto-TLS.

There are two ways to renew a private/host certificate. To renew the private/host certificates at any time, use the following CLI commands:

Data Lake certificate renewal:

```
cdp datalake rotate-private-certificates --datalake <Data Lake name or CRN>
```

Data Hub certificate renewal:

```
cdp datahub rotate-private-certificates --datahub <Data Hub name or CRN>
```

Disposition: / Status:

There are two ways to renew a private/host certificate. To renew the certificate at any time, navigate to the Data Hub or Data Lake **Details** page and click the Renew Certificate button:
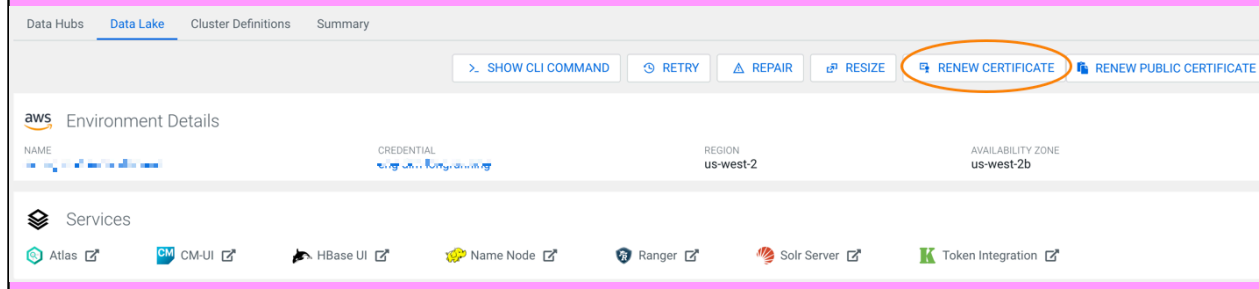
Alternatively, you can wait until the host certificate is close to expiration. During periodic cluster state synchronization, CDP uses the Cloudera Manager API to check that the HOST_AGENT_CERTIFICATE_EXPIRY apiHealthCheck alert is in a GOOD state. If the apiHealthCheck is not in a GOOD state, CDP displays a warning in the UI.

These UI warnings will display on the associated Environments, Data Lakes, or Data Hubs list and details pages. For example:
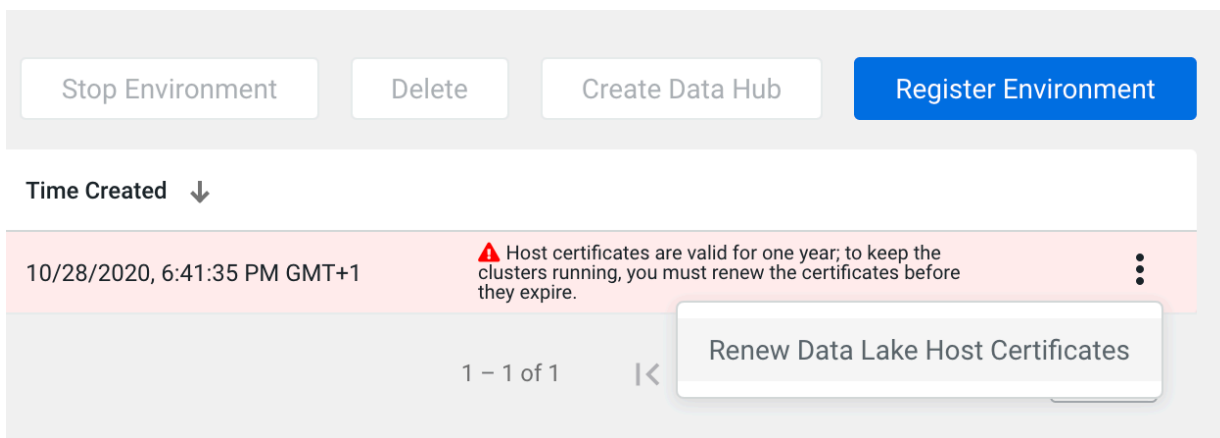
Data Lakes

To renew the host certificate once you receive an expiration warning, follow the steps below.

Note: Renewing host certificates causes some downtime because all services are restarted. The Cloudera Manager UI will be inaccessible for some time because the Cloudera Manager server is also restarted.

Procedure

1. On the Environments, Data Lakes, or Data Hubs list pages, click the three vertical dots next to the expiration message.
2. Click Renew Host Certificates or Renew Data Lake Host Certificates.

3. Click Yes when you are asked if you want to renew the certificates.

Manually renewing public certificates for Data Lake and Data Hub clusters

Public certificates are responsible for enabling TLS in front of Knox and other available services on port 443 of Data Lake and Data Hub clusters. Public certificates expire every 90 days and are often automatically renewed by CDP. If automatic renewal fails, you can renew these certificates manually.

Disposition: / Status:
This topic exists in both the Data Hub and Data Lake library. If you update this topic, make sure changes are cherry-picked to the other branch as well, assuming the changes apply to both. Currently these topics are not profiled in any way.

Required role (Data Lakes): EnvironmentAdmin or Owner of the environment

Required role (Data Hub): DatahubAdmin, Owner of the Data Hub, EnvironmentAdmin, or Owner of the environment

To renew a public certificate, click the Renew Public Certificate button on the details page of a chosen Data Lake:
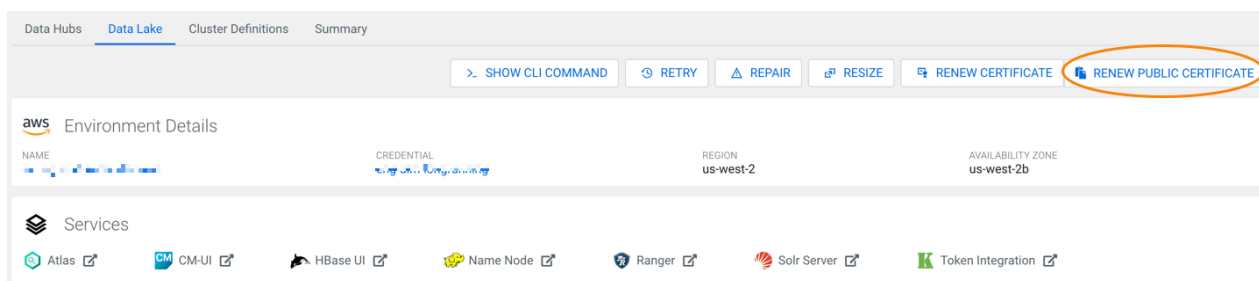
Or from the Actions menu of a Data Hub cluster details page:
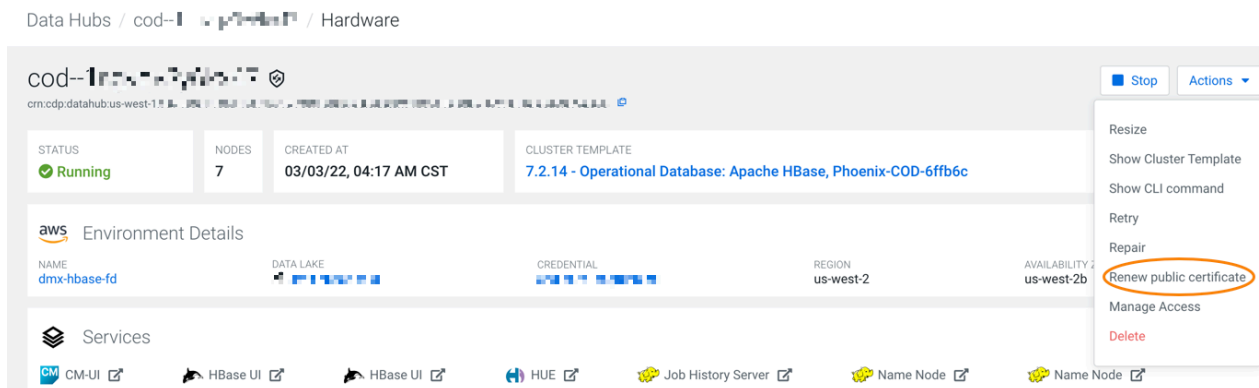

Triggering the certificate renewal may cause a minor cluster downtime of a few seconds. The entire renewal process takes a few minutes.

If you prefer to renew the certificates using the CLI, use the following commands:

Data Lake public certificate renewal:

```
cdp datalake renew-public-certificate --datalake <Data Lake name or CRN>
```

Data Hub public certificate renewal:

```
cdp datahub renew-public-certificate --datahub <Data Hub name or CRN>
```

Recipes

A recipe is a script that runs on all nodes of a selected host group at a specific time. You can use recipes to create and run scripts that perform specific tasks on your Data Hub, Data Lake, or FreeIPA cluster nodes.

Disposition: / Status:
This topic is shared between the Data Hub library and the Management Console library (Data Lake and FreeIPA doc). If you update this topic, please cherry-pick the changes to both the Data Hub and Management Console branches.

You can use recipes for tasks such as installing additional software or performing advanced cluster configuration. For example, you can use a recipe to put a JAR file on the Hadoop classpath.

Recipes can be uploaded and managed via the CDP web interface or CLI and then selected, when needed, for a specific cluster and for a specific host group. If selected, they will be executed on a specific host group at a specified time.

Recipes are stored on the Cloudera Manager server for the lifetime of the master node, and are executed at specific times of your choosing:

- pre-service-deployment (formerly pre-cluster-manager-start): During a Data Hub, Data Lake, or environment deployment, the script will be executed on every node before the CM server starts, and after node repair or OS upgrade of a cluster.
- post-cluster-manager-start: During a Data Hub or Data Lake deployment, the script will be executed on every node after the CM server starts, but before cluster installation. post-cluster-manager start recipes are also executed after node repair or OS upgrade of a cluster. This option is not available for FreeIPA recipes.
- post-service-deployment (formerly post-cluster-install): The script will be executed on every node after cluster installation on the CM server is finished, and after node repair or OS upgrade of a cluster.
- pre-termination: The script will be executed on every node before cluster termination.

> **Note:** On the master node, recipes are triggered when the CM server starts; on other nodes, recipes are triggered when the CM agent starts.

Writing recipes

Refer to these guidelines when creating your recipes.

Disposition: / Status:
This topic is shared between the Data Hub library and the Management Console library (Data Lake doc). If you update this topic, please cherry-pick the changes to both the Data Hub and Management Console branches.

When using recipes, consider the following guidelines:

- Running bash and python scripts as recipes is supported. We recommend using scripts with Shebang character sequence, for example:

```
#!/bin/sh
#!/bin/bash
#!/usr/bin/sh
#!/usr/bin/bash
#!/usr/bin/env sh
#!/usr/bin/env bash
#!/bin/sh -x
#!/usr/bin/python
#!/usr/bin/env python
```

- The scripts are executed as root. The recipe output is written to /var/log/recipes on each node on which it was executed.
- Supported parameters can be specified as variables by using mustache kind of templating with "{{{ }}}" syntax. Once specified in a recipe, these variables are dynamically replaced when the recipe is executed, generating the actual values that you provided as part of cluster creation process. For the list of parameters, refer to Recipe and cluster template parameters. For an example, see Example: Recipe using parameters.

> **Note:** Using variable parameters is not supported for FreeIPA recipes.

For example, if your cluster includes an external LDAP and your recipe includes {{{ldap.connectionURL}}}, as demonstrated in the following example

```
#!/bin/bash -e

main() {
 ping {{{ ldap.connectionURL }}}
}
```

```
[[ "$0" == "$BASH_SOURCE" ]] && main "$@"
```

then, when this recipe runs, the {{{ldap.connectionURL}}} is replaced with the actual connection URL specified as part of cluster creation process, as demonstrated in the following example:

```
#!/bin/bash -e

main() {
 ping 192.168.59.103
}
[[ "$0" == "$BASH_SOURCE" ]] && main "$@"
```

- Recipe logs can be found at /var/log/recipes/${RECIPE_TYPE}/${RECIPE_NAME}.log
- The scripts are executed on all nodes of the host groups that you select (such as "master", "worker", "compute").
- In order to be executed, your script must be in a network location which is accessible from the Management Console and the virtual network in which your cluster is located.
- Make sure to follow Linux best practices when creating your scripts. For example, don't forget to script "Yes" auto-answers where needed.
- Do not run yum update –y as it may update other components on the instances (such as salt) – which can create unintended or unstable behavior.

Example Python script

```
#!/usr/bin/python
print("An example of a python script")
import sys
print(sys.version_info)
```

Example bash script for yum proxy settings

```
#!/bin/bash
cat >> /etc/yum.conf
<<ENDOF
proxy=http://10.0.0.133:3128
ENDOF
```

Example recipe including variables

Original recipe:

```
#!/bin/bash -e

function setupAtlasServer() {
    curl -iv -u {{{ general.userName }}}:{{{ general.password }}} -H "X-Re
quested-By: ambari" -X POST -d '{"RequestInfo":{"command":"RESTART","context
":"Restart all components required ATLAS","operation_level":{"level":"SERVIC
E","cluster_name":"{{{ general.clusterName }}}","service_name":"ATLAS"}},"Re
quests/resource_filters":[{"hosts_predicate":"HostRoles/stale_configs=false&
HostRoles/cluster_name={{{ general.clusterName }}}"}]}' http://$(hostname -f
):8080/api/v1/clusters/{{{ general.clusterName }}}/requests
}

main() {
    setupAtlasServer
}

[[ "$0" == "$BASH_SOURCE" ]] && main "$@"
```

Generated recipe (to illustrate how the variables from the original recipe were replaced during cluster creation):

```
#!/bin/bash -e

function setupAtlasServer() {
    curl -iv -u admin:admin123 -H "X-Requested-By: ambari" -X POST -d '{"R
equestInfo":{"command":"RESTART","context":"Restart all components required
ATLAS","operation_level":{"level":"SERVICE","cluster_name":"super-cluster","
service_name":"ATLAS"}},"Requests/resource_filters":[{"hosts_predicate":"Hos
tRoles/stale_configs=false&HostRoles/cluster_name=super-cluster"}]}' http://
$(hostname -f):8080/api/v1/clusters/super-cluster/requests
}

main() {
    setupAtlasServer
}

[[ "$0" == "$BASH_SOURCE" ]] && main "$@"
```

Recipe and cluster template parameters

The following supported parameters can be specified as variables/dynamic parameters in recipes or cluster templates by using mustache formatting with "{{{ }}}" syntax.

Disposition: / Status:
This topic is shared between the Data Hub library and the Management Console library (Data Lake doc). If you update this topic, please cherry-pick the changes to both the Data Hub and Management Console branches.

Note: Using variable parameters is not supported for FreeIPA recipes.

The parameter keys listed below follow the following general conventions:

- { } indicates that the parameter key has multiple supported values, which are provided in this documentation. For example {fileSystemType} can be one of the following: s3, adls_gen_2, or wasb.
- [index] indicates that the parameter includes an index value for example sharedService.datalakeComponents.[index] can be "sharedService.datalakeComponents.[0]", "sharedService.datalakeComponents.[1]", and so on. There is no easy way to find out what the index will be, but you may still be able to use these parameters (for example by creating a condition to filter them).

For information on how to set these parameters dynamically in a cluster template, refer to Setting custom properties.

Custom properties

Any custom property specified in the cluster template can be used as a recipe parameter. Refer to Custom properties documentation.

General

The general parameter group includes parameters related to general cluster configuration.

Description	Example key	Example value
Name of stack	general.stackName	teststack
UUID of cluster	general.uuid	9aab7fdb-8940-454b-bc0a-62f04bce6519
Cloudera Manager user name	general.cmUserName	
Cloudera Manager password	general.cmPassword	
Cloudera Manager IP	general.clusterManagerIp	127.0.0.1

Description	Example key	Example value
Number of nodes	general.nodeCount	5
FQDN of primary gateway instance	general.primaryGatewayInstanceDiscoveryFQDN	p-10-0-88-28.example.com
Number indicating the Kafka replication factor (3 or 1)	general.kafkaReplicationFactor	1

Blueprint

The blueprint parameter group includes parameters related to cluster template configuration.

Parameter key	Description	Example key	Example value
blueprint.blueprintText	Blueprint text in JSON format	blueprint.blueprintText	
blueprint.version	Version of blueprint	blueprint.version	7.2.8

Cloud storage

The fileSystemConfigs parameter group includes parameters related to cloud storage configuration.

When forming the parameter keys, the {fileSystemType} should be replaced with an actual cloud storage type such as "s3", "adls_gen_2", or "wasb".

Parameter key	Description	Example key	Example value
File system common configurations			
fileSystemConfigs. {fileSystemType}.storageContainer	Name of container in Azure storage account (Cloudbreak + stackId)	fileSystemConfigs.s3.storageContainer	cloudbreak123
fileSystemConfigs. {fileSystemType}.type	Type of filesystem	fileSystemConfigs.s3.type	S3
fileSystemConfigs. {fileSystemType}.locations. [index].configFile	Configuration file used to configure the filesystem	fileSystemConfigs.s3.locations. [0].configFile	hbase-site
fileSystemConfigs. {fileSystemType}.locations. [index].property	Property key of filesystem path in defined config	fileSystemConfigs.s3.locations. [0].property	hbase.rootdir
fileSystemConfigs. {fileSystemType}.locations. [index].value	Value of filesystem path in defined config	fileSystemConfigs.s3.locations. [0].value	s3a://testranger/testrecipe2/apps/ hbase/data
Amazon S3 configurations			
fileSystemConfigs.s3.storageContainer	Generated name (cloudbreak + stack id number)	fileSystemConfigs.s3.storageContainer	cloudbreak7941
fileSystemConfigs.s3.locations. [index].configFile	Hadoop component configuration file	fileSystemConfigs.s3.locations. [0].configFile	zeppelin-site
fileSystemConfigs.s3.locations. [index].property	Component property name	fileSystemConfigs.s3.locations. [0].property	zeppelin.notebook.dir
fileSystemConfigs.s3.locations. [index].value	Component property value	fileSystemConfigs.s3.locations. [0].value	s3a://storagename/clustername/ zeppelin/notebook
ADLS Gen2 configurations			
fileSystemConfigs.adls_gen_2.accountName	Name of the corresponding Azure storage account	fileSystemConfigs.adls_gen_2.accountName	teststorageaccount
fileSystemConfigs.adls_gen_2.storageContainerName	Name of container in Azure storage account	fileSystemConfigs.adls_gen_2.storageContainerName	testContainerName

External database

The rds parameter group includes parameters related to external database configuration.

When forming the parameter keys, the {rdsType} should be replaced with the actual database type such as "cloudera_manager", "beacon", "druid", "hive", "oozie", "ranger", "superset", or some other user-defined type.

Parameter key	Description	Example key	Example value
rds.{rdsType}.connectionURL	JDBC connection URL	rds.hive.connectionURL	Value is specified in the following format: jdbc:postgresql://host:port/database
rds.{rdsType}.connectionDriver	JDBC driver used for connection	rds.hive.connectionDriver	org.postgresql.Driver
rds.{rdsType}.connectionUserName	Username used for the JDBC connection	rds.hive.connectionUserName	testuser
rds.{rdsType}.connectionPassword	Password used for the JDBC connection	rds.hive.connectionPassword	TestPssword123
rds.{rdsType}.databaseName	Target database of the JDBC connection	rds.hive.databaseName	myhivedb
rds.{rdsType}.host	Host of the JDBC connection	rds.hive.host	mydbhost
rds.{rdsType}.hostWithPortWithJdbc	Host of JDBC connection with port and JDBC prefix	rds.hive.hostWithPortWithJdbc	Value is specified in the following format: jdbc:postgresql://host:port
rds.{rdsType}.subprotocol	Sub-protocol from the JDBC URL	rds.hive.subprotocol	postgresql
rds.{rdsType}.connectionString	URL of JDBC the connection. In case of Ranger, this does not contain the port	rds.hive.connectionString	Value is specified in the following format: jdbc:postgresql://host:port/database
rds.{rdsType}.databaseVendor	Database vendor	rds.hive.databaseVendor	POSTGRES
rds.{rdsType}.withoutJDBCPrefix	URL of the JDBC connection without JDBC prefix	rds.hive.withoutJDBCPrefix	Value is specified in the following format: host:port/database

Gateway

The gateway parameter group includes parameters related to Knox gateway configuration.

Parameter key	Description	Example key	Example value
gateway.ssoProvider	Path to the SSO provider	gateway.ssoProvider	/test/sso/api/v1/websso
gateway.signKey	Base64 encoded signing key	gateway.signKey	
gateway.signPub	Signing certificate (x509 format)	gateway.signPub	
gateway.signCert	Public SSH key used for signing (standard public key format)	gateway.signCert	

Shared services

The sharedService parameter group includes parameters related to Data Lake configuration.

Parameter key	Description	Example key	Example value
sharedService.rangerAdminPassword	Admin password of the Ranger component	sharedService.rangerAdminPassword	Admin1234!
sharedService.datalakeCluster	Flag indicating that the cluster is a data lake cluster	sharedService.datalakeCluster	true
sharedService.datalakeClusterManagerIp	Cloudera Manager IP of data lake cluster	sharedService.datalakeClusterManagerIp	127.0.0.1
sharedService.datalakeClusterManagerFqdn	Cloudera Manager FQDN of data lake cluster (or the IP if FQDN cannot be found)	sharedService.datalakeClusterManagerFqdn	ip-10-0-88-28.example.com

Example: Recipe with parameters

If you pass the supported parameters in a recipe, their values are dynamically fetched and replaced.

Note: Using variable parameters is not supported for FreeIPA recipes.

Example recipe template (the {{{general.clusterName}}} is included as a template):

```
#!/bin/bash -e

function setupDefaultClusterFolder() {
  mkdir -p /var/log/{{{general.clusterName}}}
}

main() {
    setupDefaultClusterFolder
}

[[ "$0" == "$BASH_SOURCE" ]] && main "$@"
```

Example recipe after {{{general.clusterName}}} is set to my-super-cluster based on the actual cluster name:

```
#!/bin/bash -e

function setupDefaultClusterFolder() {
  mkdir -p /var/log/my-super-cluster
}

main() {
    setupDefaultClusterFolder
}
[[ "$0" == "$BASH_SOURCE" ]] && main "$@"
```

Register a recipe

In order to use your recipe for clusters, you must first register it with the Management Console.

About this task

Required role: EnvironmentCreator can create a shared resource and then assign users to it.

SharedResourceUser or Owner of the shared resource can use the resource.

Before you begin
If you are using CDP with a proxy, note that the CDP proxy settings do not apply to cluster recipes. If you planning to use the recipes, then you can set the proxy settings manually. You can find the proxy settings in the /etc/cdp/proxy.env file.

Procedure

1. Place your script in a network location accessible from Management Console and from the virtual network in which your clusters are located.
2. Log in to the CDP web interface.
3. Navigate to Shared ResourcesRecipes and click Register Recipe.
4. Provide the following:

Parameter	Value
Name	Enter a name for your recipe.
Description	(Optional) Enter a description for your recipe.
Execution Type	Select one of the following options: • pre-service-deployment (formerly pre-cluster-manager-start): During a Data Hub, Data Lake, or environment deployment, the script will be executed on every node (in the host group where you assigned the recipe) before the CM server starts. • post-cluster-manager-start: During a Data Hub or Data Lake deployment, the script will be executed on every node (in the host group where you assigned the recipe) after the CM server starts, but before cluster installation. This option is not available for FreeIPA recipes. • post-service-deployment (formerly post-cluster-install):: The script will be executed on every node (in the host group where you assigned the recipe) after cluster installation on the CM server is finished. • pre-termination: The script will be executed on every node (in the host group where you assigned the recipe) before cluster termination.
Script	Select one of: • File: Point to a file on your machine that contains the recipe. • Text: Paste the script.

5. Click Register.

What to do next

• When you create a Data Hub cluster, you can select a previously added recipe on the advanced Cluster Extensions page of the create cluster wizard.
• When you create an environment, you can select a previously added recipe on the Data Access and Data Lake Scaling page of the environment creation wizard, under Advanced Options > Cluster Extensions > Recipes.
• When you create an environment, you can select a previously added FreeIPA recipe on the **Region, Networking, and Security** page of the environment creation wizard, under Advanced OptionsCluster ExtensionsRecipes.
• You can also attach recipes to Data Hub or Data Lake clusters when you create an environment/Data Lake or Data Hub through the CDP CLI.

Update a recipe

You can attach or detach recipes to/from existing Data Lake clusters in an available state. Using this capability, you can update a recipe by removing it from the cluster, replacing the old recipe with a modified recipe of the same type, and attaching the new modified recipe to the cluster.

About this task

Attaching or detaching a recipe will not execute the recipe. The next execution of the recipe will take place based on the type of the recipe. After an upscale, a newly attached recipe runs only on the new hosts.

Required role (one of the following):

- PowerUser on CDP tenant
- Owner of the environment
- EnvironmentAdmin

Procedure

1. Create a new recipe (with updated/modified content) of the same type as the old recipe that you want to replace.

2. Click Data Lakes<Data Lake Name> and scroll to the Data Lake details pane at the bottom of the page. Click the Recipes tab and find the recipe that you want to remove in the list of recipes for the Data Lake.

3. Click Remove Recipe next to the name of the recipe that you want to remove, then click Yes in the confirmation window.

4. Once you have removed the old recipe, click on the Add Recipe button for the Data Lake and select the same host group that you previously used for the old recipe. Then select the name of the new recipe that contains the modified content and click Add.
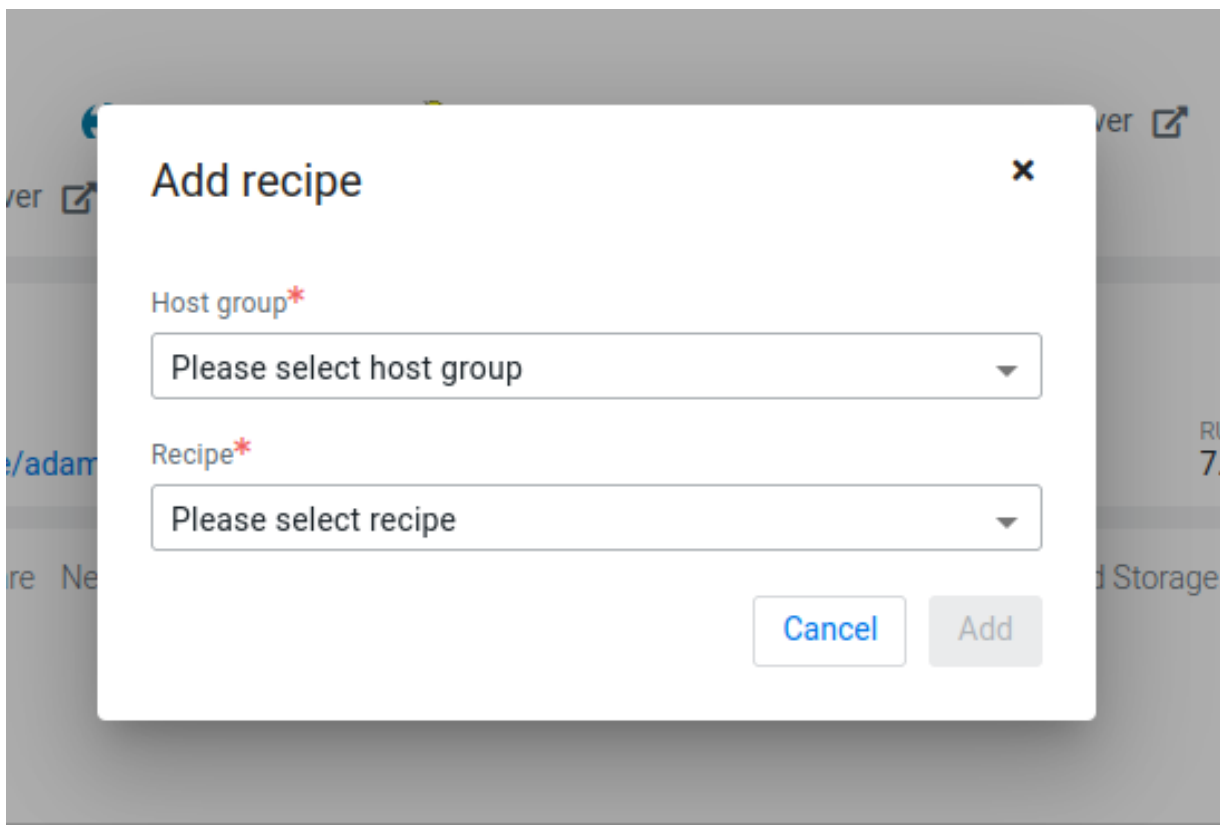
5. Alternatively, you can use the CDP CLI to attach or detach recipes from a host group:

```
cdp datalake replace-recipes --datalake <CRN or name> --instance-group-r
ecipes instanceGroupName=<instance group name>,recipeNames=<recipe names>
```

Note that the same command is used to both detach and attach a recipe. When you attach a recipe, use the recipeNames parameter to specify the recipe or recipes that you want to attach; when you detach a recipe, give the instanceGroupName but do not provide a recipeName. For example:

To detach all recipes from an instance group:

```
cdp datalake replace-recipes --datalake myDL --instance-group-recipes
```

```
                    instanceGroupName=worker,recipeNames=
```

To attach a new recipe:

```
cdp datalake replace-recipes --datalake myDL --instance-group-recipes
            instanceGroupName=worker,recipeNames=myrecipe
```

For instance groups with multiple recipes, give the recipeNames that you would like to keep. Any recipes not specified will be detached.

Results
You should see the new recipe appear for the same host group. After this change, the next recipe execution will execute the new script.

Managing recipes from CLI

You can manage recipes from CLI using cdp datahub commands.

Required role: EnvironmentCreator can create a shared resource and then assign users to it.

SharedResourceUser or Owner of the shared resource can use the resource. The Owner of the shared resource can delete it.

> **Disposition: / Status:**
> This topic is shared between the Data Hub library and the Management Console library (Data Lake doc). If you update this topic, please cherry-pick the changes to both the Data Hub and Management Console branches.

Note: Currently, recipes use cdp datahub commands regardless of whether the recipe is intended to run on Data Hub, Data Lake, or FreeIPA cluster nodes.

- Register a new recipe: cdp datahub create-recipe --recipe-name <value> --recipe-content <value> --type <value>

  Supported types:

  - PRE_SERVICE_DEPLOYMENT (formerly PRE_CLOUDERA_MANAGER_START)
  - POST_CLOUDERA_MANAGER_START (this option is not available for FreeIPA recipes)
  - POST_SERVICE_DEPLOYMENT (formerly POST_CLUSTER_INSTALL)
  - PRE_TERMINATION
- List all available recipes: cdp datahub list-recipes
- Describe a specific recipe: cdp datahub describe-recipe --recipe-name <value>
- Delete one or more existing recipes: cdp datahub delete-recipes --recipe-name <value>