Cloudera Data Hub

Managing Clusters

Date published: 2019-12-17
Date modified: 2025-08-18

https://docs.cloudera.com/

Legal Notice

Contents

Managing clusters

You can manage your Cloudera Data Hub clusters from the Cloudera Data Hub cluster dashboard in the Cloudera Management Console > Data Hub Clusters.

You can access cluster management options by clicking on the tile representing the cluster that you want to access. The actions available for your cluster are listed in the top right corner:

Retrying a cluster

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: The DataHubAdmin or Owner roles at the scope of Cloudera Data Hub allow you to manage the Cloudera Data Hub cluster. Note that EnvironmentAdmin and Owner of the environment can also manage Cloudera Data Hub clusters.

Only failed stack or cluster creations can be retried. A retry can be initiated any number of times on a failed creation process.

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

Note: In case of high node-count AWS clusters, multiple small requests are sent to AWS instead of a single large one to avoid insufficient capacity, request limit exceeded errors when starting the cluster.

Procedure

1. Log in to the Cloudera web interface.
2. Navigate to the Cloudera Management Console > Data Hub Clusters.
3. Browse to cluster details.
4. Click Actions and select Retry.

   Note:

   Only failed stack or cluster creations can be retried, so the option is only available in these two cases.
5. Click Yes to confirm. The operation continues from the last failed step.

Resizing a cluster

The resize option allows you to add or remove cluster nodes.

Before you begin

Ensure that you are using a cluster that can be resized.

Note: You can scale up and down Flow Management clusters since 7.2.11 and you can scale up and down Streams Messaging clusters since 7.2.12. For more information, see *Scaling up or down a NiFi cluster* and *Scaling Kafka broker nodes*.

About this task

Required role: The DataHubAdmin or Owner roles at the scope of Cloudera Data Hub allow you to manage the Cloudera Data Hub cluster. Note that EnvironmentAdmin and Owner of the environment can also manage Cloudera Data Hub clusters.

Unhealthy clusters can be resized, provided that the failed node is not a gateway node. If a new node fails to come up after scaling, the upscale operation will still complete successfully, and the failed nodes will be marked as "zombie" nodes. The stack is usable without these nodes, and you can clean them up manually. The "zombie" status indicates that the node provision has failed.

Procedure

1. Log in to the Cloudera web interface.
2. Navigate to the Cloudera Management Console > Data Hub Clusters
3. Browse to cluster details.
4. Click Actions and select Resize. The cluster resize dialog is displayed.
5. Using the +/- controls, adjust the number of nodes for a chosen host group.

   Note:

   You can only modify one host group at a time. It is not possible to resize the Cloudera Manager server host group (by default, this is the master host group).

   Note:

   Downscaling the worker host group below 3 nodes is not possible on clusters that have at least 3 worker nodes by design.
6. Click Yes to confirm the scale-up/scale-down.

   While nodes are being added or removed, cluster status changes to “Update In Progress”. Once the operation has completed, cluster status changes back to “Running”. Messages similar to the following are written to even history:

```
Cluster scaled up
8/2/2019, 6:23:24 PM
Scaling up the cluster
8/2/2019, 6:14:05 PM
Stack successfully upscaled
8/2/2019, 6:14:04 PM
Mounting disks on new nodes
8/2/2019, 6:14:04 PM
Bootstrapping new nodes
8/2/2019, 6:13:48 PM
Infrastructure metadata extension finished
8/2/2019, 6:13:48 PM
Billing changed due to upscaling of cluster infrastructure
8/2/2019, 6:13:47 PM
Adding 1 new instances to the infrastructure
8/2/2019, 6:12:34 PM
```

Related Information

Scaling up or down a NiFi cluster

Scaling Kafka broker nodes

Stopping a cluster

Cloudera Data Hub supports stopping and restarting clusters. Once a cluster is in the "Running" state, it can be stopped.

About this task

Required role: The DataHubAdmin or Owner roles at the scope of Cloudera Data Hub allow you to manage the Cloudera Data Hub cluster. Note that EnvironmentAdmin and Owner of the environment can also manage Cloudera Data Hub clusters.

Note: Most clusters that use ephemeral volumes cannot be stopped via Cloudera Data Hub. If a cluster has a host group that contains only yarn-NODEMANAGER and/or any "-GATEWAY" service roles, using ephemeral volumes as temporary storage for these host groups is supported, which allows for stop-start of the cluster.

Procedure

1. Log in to the Cloudera web interface.
2. Navigate to the Cloudera Management Console > Data Hub Clusters
3. Browse to cluster details.
4. Click Stop to stop a currently running cluster.
5. Click Yes to confirm.
6. Your cluster status changes to “Stopping in progress” and then to “Stopped”.

   Messages similar to the following are written to the even history:

```
Synced instance states with the cloud provider.
8/2/2019, 6:29:39 PM
Updated metadata of instance i-053eb3a4e560d99e5 to stopped as the cloud
 provider reported it as stopped.
8/2/2019, 6:29:39 PM
Updated metadata of instance i-081925ba5152e36d2 to stopped as the cloud
 provider reported it as stopped.
8/2/2019, 6:29:38 PM
Updated metadata of instance i-00dc4cb6a7b9429ec to stopped as the cloud
provider reported it as stopped.
8/2/2019, 6:29:37 PM
Updated metadata of instance i-08649b04abc3b54a2 to stopped as the cloud p
rovider reported it as stopped.
8/2/2019, 6:29:36 PM
Billing stopped, the cluster and its infrastructure have been stopped
8/2/2019, 6:29:34 PM
Infrastructure successfully stopped
8/2/2019, 6:29:34 PM
Infrastructure is now stopping
8/2/2019, 6:28:51 PM
Cluster stopped
8/2/2019, 6:28:50 PM
Cloudera Manager services have been stopped.
8/2/2019, 6:28:49 PM
Stopping Cloudera Manager services.
8/2/2019, 6:27:28 PM
Stopping cluster
8/2/2019, 6:27:28 PM
Cluster infrastructure stop requested
```

```
8/2/2019, 6:27:26 PM
```

Once stopping the infrastructure has completed, you will see a Start option to restart your cluster. When a cluster is put in the “stopped” state, cluster VMs are given back to the cloud provider. To provision new VMs, restart the cluster.

Related Information

Restarting a cluster

Restarting a cluster

If your cluster is in the “Stopped” state, you can restart it by using the "Start" option.

About this task

Required role: The DataHubAdmin or Owner roles at the scope of Cloudera Data Hub allow you to manage the Cloudera Data Hub cluster. Note that EnvironmentAdmin and Owner of the environment can also manage Cloudera Data Hub clusters.

Note:

- Most clusters that use ephemeral volumes cannot be stopped/restarted via Cloudera Data Hub. If a cluster has a host group that contains only yarn-NODEMANAGER and/or any "-GATEWAY" service roles, using ephemeral volumes as temporary storage for these host groups is supported, which allows for stop-start of the cluster.
- In case of high node-count AWS clusters, multiple small requests are sent to AWS instead of a single large one to avoid insufficient capacity, request limit exceeded errors when starting the cluster.

Procedure

1. Log in to the Cloudera web interface.
2. Navigate to the Cloudera Management Console > Data Hub Clusters
3. Browse to cluster details.
4. Click Start. This option is only available when the cluster has been stopped.
5. Click Yes to confirm.

   Your cluster status changes to “Start in progress” and then once new VMs have been provisioned and registered, it changes to “Running”. Messages similar to the following are written to event history:

```
Synced instance states with the cloud provider.
8/2/2019, 6:33:37 PM
Updated metadata of instance ip-10-97-82-237.cloudera.site to running as
 the cloud provider reported it as running.
8/2/2019, 6:33:37 PM
Updated metadata of instance ip-10-97-83-214.cloudera.site to running as t
he cloud provider reported it as running.
8/2/2019, 6:33:36 PM
Updated metadata of instance ip-10-97-82-184.cloudera.site to running as
the cloud provider reported it as running.
8/2/2019, 6:33:36 PM
Updated metadata of instance ip-10-97-82-151.cloudera.site to running as
 the cloud provider reported it as running.
8/2/2019, 6:33:35 PM
Cluster started; Cluster manager ip:10.97.82.237
8/2/2019, 6:33:34 PM
Cloudera Manager services have been started.
8/2/2019, 6:33:33 PM
```

```
Starting Cloudera Manager services.
8/2/2019, 6:31:15 PM
Starting cluster
8/2/2019, 6:30:07 PM
Billing started, the cluster and its infrastructure have successfully been
 started
8/2/2019, 6:30:06 PM
Infrastructure successfully started
8/2/2019, 6:30:06 PM
Infrastructure is now starting
8/2/2019, 6:29:51 PM
Cluster start requested
8/2/2019, 6:29:51 PM
```

What to do next

Once your cluster status is “Running”, the cluster can be used.

Related Information

Stopping a cluster

Stopping and restarting cluster services

If you would like to stop and restart services, you should use Cloudera Manager.

For Cloudera Manager documentation, refer to the link below.

Related Information

Managing Cloudera Runtime services

Performing manual repair

Manual repair should be performed on a cluster that has nodes marked as unhealthy.

Required role: The DataHubAdmin or Owner roles at the scope of Cloudera Data Hub allow you to manage the Cloudera Data Hub cluster. Note that EnvironmentAdmin and Owner of the environment can also manage Cloudera Data Hub clusters.

Note:

The manual repair option is available for all clusters and covers all nodes, including the Cloudera Manager server node (by default this is the master node). There is an exception: for clusters created before March 26, 2021, repair of the master node is enabled only if the Cloudera Data Hub cluster uses an external database. Ephemeral disks cannot be reattached; they are deleted and new disks are provisioned.

> **Important:**
>
> Due to their inherent ephemeral nature, Cloudera cannot preserve data that resides on the root disk. A manual repair operation on an unhealthy node will replace the disk mounted as root, which means all data on the root disk is lost.
>
> To avoid the loss of important data, do not store it in the root disk. Instead, do one of the following:
>
> - Store data that needs to persist beyond the lifetime of the cluster in S3 or ADLS Gen 2, depending on the platform in use.
> - Store data that needs to survive a repair operation in /hadoopfs/fsN/ (where N is an integer), as long as it is not so large that it could crowd out components that use that location.
>
>   For example, storing 1 GB of data in /hadoopfs/fs1/save_me would be an option to ensure that the data is available in a replacement node after a manual repair operation.

When a cluster has unhealthy nodes, a warning is displayed:

- Cluster tile on the cluster dashboard shows unhealthy nodes
- Nodes are marked as "UNHEALTHY" in the Hardware section
- Cluster's event history shows "Manual recovery is needed for the following failed nodes"

There are two ways to repair the failed nodes:

- Repair the failed nodes: (1) All non-ephemeral disks are detached from the failed nodes. (2) Failed nodes are removed (3) New nodes of the same type are provisioned. (4) The disks are attached to the new volumes, preserving the data.
- Delete the failed nodes: Failed nodes are deleted with their attached volumes.

If a node is marked as deleted from the cloud provider side, you must perform manual repair before restarting the cluster. If you do not perform the manual repair, the cluster will not restart.

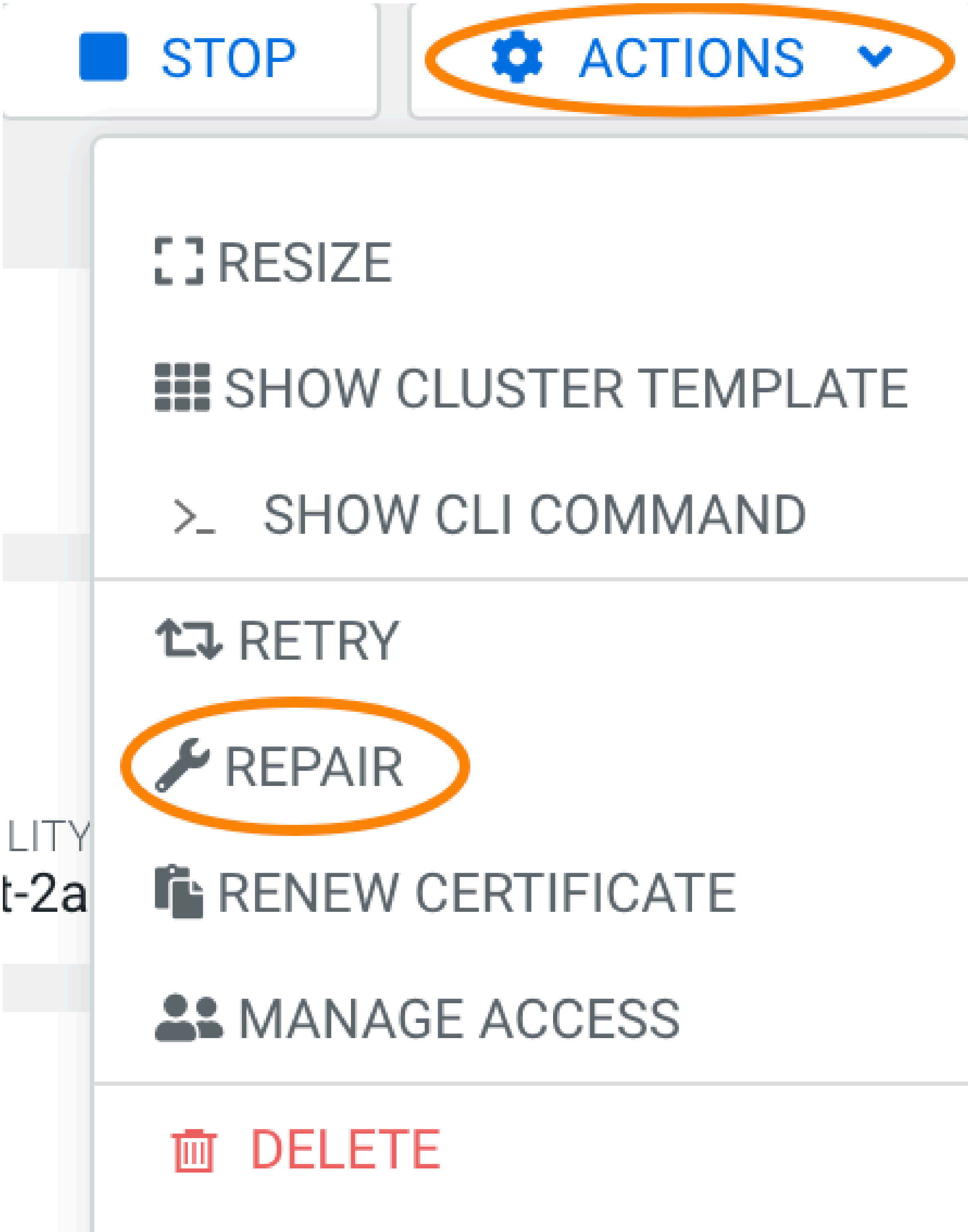
You can perform manual repair from the Cloudera web interface or CLI.

To perform manual repair from Cloudera web interface:

1. Log in to the Cloudera web interface.
2. Navigate to Cloudera Management Console > Data Hub Clusters.
3. Browse to the cluster details.

4. Choose one of the following options:

- To repair all of the failed nodes in a particular host group, select Actions > Repair and then select the host group to repair. Only one host group can be selected at a time:



- 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). Alternatively you can use the CDP CLI: cdp datahub repair-cluster help

- 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.

5. By default, unhealthy nodes are removed and then replaced. If you would like to just remove the nodes without replacing them, select Remove only.
6. Click Repair.
7. Once the recovery flow is completed, the cluster status changes to 'RUNNING'.

Related Information

Node repair

Terminating a cluster

You can terminate any cluster managed by Cloudera Data Hub.

About this task

Required role: Owner at the scope of the Cloudera Data Hub (which provides delete access).

If you want to delete individual cluster nodes instead of terminating an entire cluster, see *Terminate cluster nodes*.

Procedure

1. Log in to the Cloudera web interface.
2. Navigate to the Cloudera Management Console > Data Hub Clusters
3. Browse to cluster details.
4. Click Terminate.
5. Click Yes to confirm.

   All cluster-related resources will be deleted, unless the resources (such as networks and subnets) existed prior to cluster creation or are used by other VMs in which case they will be preserved.

Related Information

Force terminating a cluster

Terminating cluster nodes

Terminating cluster nodes

Rather than terminating an entire Cloudera Data Hub cluster, you have the option to terminate one or more nodes of the cluster.

About this task

Required role: Owner at the scope of the Cloudera Data Hub (which provides delete access).

Procedure

1. In the Cloudera UI, navigate to Data Hub Clusters and click on the cluster details tile.
2. Click the Hardware tab.
3. Select the checkbox of the node(s) that you want to delete from the desired instance group.
4. Click the Delete (garbage can) icon.

5. Click Yes to confirm.

   Node deletion may fail if the Cloudera Management Console is unable to delete one or more of the cloud resources that were part of the node's infrastructure. Use the Force terminate option to remove the node entry from the Cloudera Management Console web UI. If you use this option, you must also check your cloud provider account to see if there are any resources that must be deleted manually.

Related Information

Performing manual repair

Force terminating a cluster

The force terminate option can be used when cluster deletion fails.

Before you begin

Required role: Owner at the scope of the Cloudera Data Hub (which provides delete access).

Cluster deletion may fail if Cloudera Data Hub is unable to delete one or more of the cloud resources that were part of your cluster infrastructure. Use the Terminate > Force terminate option to remove the cluster entry from the Cloudera Data Hub web UI. If you use this option, you must also check your cloud provider account to see if there are any resources that must be deleted manually.

Procedure

1. Log in to the Cloudera web interface.
2. Navigate to the Cloudera Management Console > Data Hub Clusters
3. Browse to cluster details.
4. Click Terminate.
5. Check Force terminate.
6. Click Yes to confirm.

   When terminating a cluster with Kerberos enabled, you have an option to disable Kerberos prior to cluster termination. This option removes any cluster-related principals from the KDC.
7. This deletes the cluster tile from the UI.
8. Log in to your cloud provider account and manually delete any resources that failed to be deleted.

What to do next

If force-termination fails, you may have to delete the cluster manually.

Related Information

Deleting clusters when termination fails

Deleting clusters when termination fails

Upon cluster termination, Cloudera Data Hub only terminates the resources that it created. It does not terminate any resources (such as networks, subnets, roles, and so on) which existed prior to cluster creation.

The proper way to delete clusters is by using the Terminate option available from the cluster details in the Cloudera Data Hub service console. If the terminate process fails, try the Terminate > Force terminate option. If the force termination does not delete all cluster resources, delete the resources manually:

- To find the VMs, click on the links available in the Cloudera Management Console > Data Hub Clusters cluster details > Hardware section.

  If you terminate instances on the cloud provider without using Cloudera Data Hub, during the sync operation, Cloudera Data Hub realizes that instances have been removed and marks the instances as DELETED_ON_PROVIDER_SIDE.

  On AWS and GCP, Cloudera Data Hub automatically removes all VM dependencies such as attached disks, network interfaces, public IPs and storage accounts. On Azure, you must manually delete the disks created for that instance and release the public IP assigned to that instance.
- To find the network and subnets, see the Cluster Information pane in the cluster details.

Creating a multi-AZ Cloudera Data Hub cluster on AWS

By default, Cloudera provisions Cloudera Data Hub clusters in a single AWS availability zone (AZ), but you can optionally choose to deploy them across multiple availability zones (multi-AZ).

For general information about multi-AZ in Cloudera, refer to Deploying Cloudera in multiple AWS availability zones.

Create a multi-AZ Cloudera Data Hub cluster

You can create a multi-AZ Data Hub via Cloudera UI or CDP CLI within an environment. Note that the CLI allows you to manually specify subnets, which is not possible via the UI.

Steps

For CDP UI

Create your Cloudera Data Hub cluster as usual. In the Advanced Options > Network and Availability, you select the multiple subnets across which the Cloudera Data Hub cluster is to be provisioned. If multiple subnets are selected, the Cloudera Manager node group will only have one subnet for each AZ; All other nodes will have all the selected subnets.

For CDP CLI

When creating a multi-AZ Cloudera Data Hub cluster, the creation request should contain the --multi-az option. For example:

```
cdp datahub create-aws-cluster \
--cluster-name tb-datamart-multiaz \
--environment-name tb-multiaz-env \
--cluster-template-name "7.2.15 - COD Edge Node" \
--instance-groups nodeCount=3,instanceGroupName=coordinator,instanceGro
upType=CORE,instanceType=r5d.4xlarge,rootVolumeSize=100,attachedVolumeCo
nfiguration=\[\{volumeSize=300,volumeCount=2,volumeType=ephemeral\}\],re
coveryMode=MANUAL,volumeEncryption=\{enableEncryption=false\} nodeCount=
1,instanceGroupName=master,instanceGroupType=GATEWAY,instanceType=r5.4xl
arge,rootVolumeSize=100,attachedVolumeConfiguration=\[\{volumeSize=100,v
olumeCount=1,volumeType=standard\}\],recoveryMode=MANUAL,volumeEncryption=
\{enableEncryption=false\} nodeCount=2,instanceGroupName=executor,instance
GroupType=CORE,instanceType=r5d.4xlarge,rootVolumeSize=100,attachedVolum
eConfiguration=\[\{volumeSize=300,volumeCount=2,volumeType=ephemeral\}\]
,recoveryMode=MANUAL,volumeEncryption=\{enableEncryption=false\} \
--image id=23e20852-7865-4980-a045-539296340b55,catalogName=cdp-default \
--profile mowdev \
--multi-az
```

By default, Cloudera Data Hub instances are distributed across the subnets provided during environment registration. If you prefer to specify the subnets manually, you can add the --subnet-ids option and specify subnets where you would like to deploy the hosts, overwriting the default node distribution. For example:

```
--subnet-ids "subnet-013855b2fc32c2cd8" "subnet-02b9054ec829374fe" "subn
et-085c9ff36b38c0b35"
```

If multiple subnets are provided, the Cloudera Manager node group will only have one subnet for each AZ; All other nodes will have all the selected subnets.

The subnets passed via the --subnet-ids option will be applied to all cluster instance groups. If you would like to specify custom subnet ID lists for any given instance group, you can pass them in the subnetIds inside the --instan ce-groups option.

Scaling a multi-AZ Cloudera Data Hub cluster

If there is an availability zone that is offline, Cloudera may not detect the outage. In such a case, if you know that a certain availability zone is offline, you can scale your cluster and manually specify where the new nodes should be provisioned.

When scaling a multi-AZ cluster, Cloudera automatically distributes the new nodes in a round-robin fashion across all available availability zones, prioritizing the least used availability zones. If you prefer to manually control the distribution of nodes across subnets during Cloudera Data Hub scaling, the desired availability zones can be controlled via the related subnets during upscales with the optional --preferred-subnet-ids field.

For example:

```
cdp datahub scale-cluster --cluster-name tb-datamart-multiaz \
  --instance-group-name "coordinator" \
  --instance-group-desired-count 5 \
  --preferred-subnet-ids "subnet-013855b2fc32c2cd8"
 "subnet-02b9054ec829374fe" "subnet-085c9ff36b38c0b35"
```

If you manually specify the subnets in this manner, this overwrites the default behavior.

Creating a multi-AZ Cloudera Data Hub cluster on Azure

By default, Cloudera provisions Cloudera Data Hub clusters in a single Azure availability zone (AZ), but you can optionally choose to deploy them across multiple availability zones (multi-AZ).

For general information about multi-AZ in Cloudera, refer to Deploying Cloudera in multiple Azure availability zones.

Create a multi-AZ Cloudera Data Hub cluster

You can create multi-AZ Cloudera Data Hub clusters within any existing environment. Detailed steps are provided below.

Prerequisites

You can create a multi-AZ Cloudera Data Hub cluster in a multi-AZ environment only. If you are trying to create a multi-AZ Cloudera Data Hub cluster in an environment that uses the default AZ distribution, you need to first edit that environment and add AZs to it.

Steps

For Cloudera UI

To enable multi-AZ when creating a Cloudera Data Hub cluster on Azure, navigate to the Advanced Options > Network And Availability and in the “Azure Availability Zones” section click the toggle button next to Enable using multiple availability zones.

For CDP CLI

You can create a multi-AZ Cloudera Data Hub cluster by adding the --multi-az option to the Cloudera Data Hub cluster creation command.

In the --instance-groups parameter, you can optionally include the availabilityZones to select the specific availability zones that should be used. If this parameter is not provided, all three AZs are used. For example:

```
cdp datahub create-azure-cluster \
 --cluster-name test-cluster1 \
 --environment-name test-env \
 --cluster-template-name "7.2.17 - Data Engineering: Apache Spark, Apache
Hive, Apache Oozie" \
 --multi-az \
```

```
cdp datahub create-azure-cluster \
 --cluster-name test-cluster1 \
 --environment-name test-env \
 --cluster-template-name "7.2.17 - Data Engineering: Apache Spark, Apache
Hive, Apache Oozie" \
 --multi-az \
 --instance-groups
          nodeCount=1,instanceGroupName=compute,instanceGroupType=CORE,ins
tanceType=Standard_D5_v2,rootVolumeSize=100,attachedVolumeConfiguration=
\[\{volumeSize=100,volumeCount=0,volumeType=StandardSSD_LRS\}\],recovery
Mode=MANUAL,availabilityZones=\[1,2\]
          nodeCount=0,instanceGroupName=gateway,instanceGroupType=CORE,
instanceType=Standard_D8_v3,rootVolumeSize=100,attachedVolumeConfigurati
on=\[\{volumeSize=100,volumeCount=1,volumeType=StandardSSD_LRS\}\],recov
eryMode=MANUAL,availabilityZones=\[2,3\]
          nodeCount=1,instanceGroupName=master,instanceGroupType=GATEWA
Y,instanceType=Standard_D16_v3,rootVolumeSize=100,attachedVolumeConfigur
ation=\[\{volumeSize=100,volumeCount=1,volumeType=StandardSSD_LRS\}\],re
coveryMode=MANUAL,availabilityZones=\[1,2,3\]
          nodeCount=3,instanceGroupName=worker,instanceGroupType=CORE,inst
anceType=Standard_D5_v2,rootVolumeSize=100,attachedVolumeConfiguration=\
[\{volumeSize=100,volumeCount=1,volumeType=StandardSSD_LRS\}\],recoveryM
ode=MANUAL,availabilityZones=\[1,3\]
```

Scaling a multi-AZ Cloudera Data Hub cluster

If there is an availability zone that is offline, Cloudera may not detect the outage. In such a case, if you know that a certain availability zone is offline, you can scale your cluster and manually specify where the new nodes should be provisioned.

When scaling a multi-AZ cluster, Cloudera automatically distributes the new nodes in a round-robin fashion across all available availability zones, prioritizing the least used availability zones. If you prefer to manually control the distribution of nodes across zones during Cloudera Data Hub scaling, the desired availability zones can be controlled via the preferred zones during upscales with the optional --preferred-zones field.

For example:

```
cdp datahub scale-cluster --cluster-name tb-datamart-multiaz \
  --instance-group-name "coordinator" \
  --instance-group-desired-count 5 \
  --preferred-zones "1" "2" "3"
```

If you manually specify the zones in this manner, this overwrites the default behavior.

Vertically scaling instances and disks

If necessary, you can select a larger or smaller instance type for a Cloudera Data Hub or Data Lake cluster after it has been deployed, or add, delete, or modify attached disks.

Vertically scaling instance types

If necessary, you can select a larger or smaller instance type for a Cloudera Data Hub or Data Lake cluster after it has been deployed in AWS, Azure, and GCP.

Before you begin

You must stop the Data Lake or Cloudera Data Hub cluster before you vertically scale any of the instances. See Change the instance type in AWS documentation for more information.

About this task

Selecting a larger instance type adds more vCPU and/or RAM to your instances. Instances can be scaled both up and down, but scaling down to a smaller size requires 4 CPU and a minimum of 4 GB memory.

Note: Do not scale your instance manually on the cloud provider side as this can possibly result in errors in the future. It is possible that Cloudera will be out of sync with the actual instance scale and a repair or upgrade could fail, or the previous scale could possibly be reinstated during a repair action. Use the Cloudera Control Plane process described here instead.

Limitations:

- You cannot scale down to a target instance type that has fewer ephemeral volumes than the current instance type.
- Vertical scaling to Azure v5 instances is not supported and results in the following error:

```
Unable to resize since changing from resource disk to non-resource disk VM
 size and vice-versa is not allowed.
```

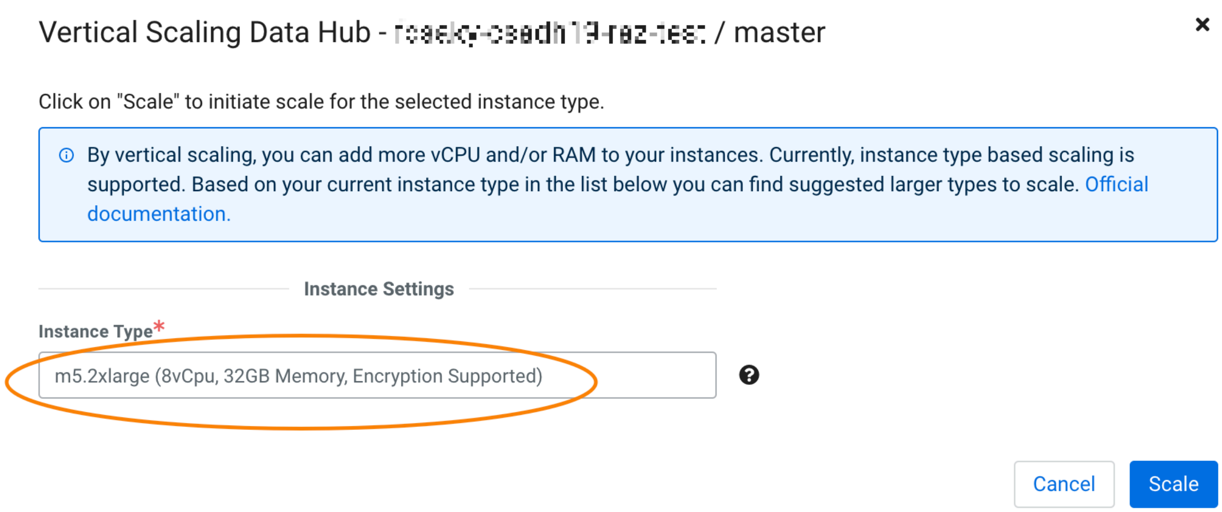
- Vertical scaling is supported on AWS, Azure, and GCP.
- If you are using an instance without ephemeral disks, you can scale up or down to a new instance with ephemeral disks; however, the reverse is not supported. You cannot start with an instance with ephemeral disks and move to an instance without ephemeral disks.

For information on vertically scaling FreeIPA, see Vertically scale FreeIPA instances.

Procedure

1. In the Cloudera main navigation menu, click Data Hub or Data Lakes and select the cluster that requires a larger instance type.
2. Scroll to the bottom of the page and click the **Nodes** tab.
3. Click the Vertical Scaling icon on the top right of the host group that you want to scale.

4. Select a larger instance type from the drop-down menu of suggested instance types.



5. Click Scale. You can monitor the action from the **Event History** tab.

   Alternatively, you can use the CDP CLI to select a new instance for the Data Lake or Cloudera Data Hub cluster:

   Data Lake cluster:

```
cdp datalake start-datalake-vertical-scaling
--datalake <your-data-lake-name-or-its-crn>
--group <master>
--instance-template instanceType="<m5.4xlarge>"
```

   Cloudera Data Hub cluster:

```
cdp datahub start-cluster-vertical-scaling
--datahub <your-data-hub-name-or-its-crn>
--group <master>
--instance-template instanceType="<m5.4xlarge>"
```

What to do next

After you have vertically scaled the cluster, configure the services on the cluster to use the additional or reduced resources/memory.

Vertically scaling disks

With the growing amount of data, it might be necessary to add, delete, or modify disks attached to Data Lake or Cloudera Data Hub clusters in AWS.

There are many clusters that are deployed with standard magnetic storage. With the growing lineage data, these disks are running out of space on core nodes. These need to be moved to General Purpose SSDs (gp2/gp3 on AWS) and or resized to a bigger disk.

The disks attached to the Data Lake and Cloudera Data Hub clusters can be changed or resized in AWS without downtime.

Note: Do not scale your instance manually on the cloud provider side as this can possibly result in errors in the future. It is possible that Cloudera will be out of sync with the actual instance scale and a repair or upgrade could fail, or the previous scale could possibly be reinstated during a repair action. Use the Cloudera Control Plane process described here instead.

Limitations

When using this preview feature, be aware of the following limitations:

- This feature is only available for AWS.
- The disks can only be resized up, meaning you cannot reduce the size of an attached block storage. If there are multiple disks of different sizes, the size of all the disks attached to the instances in the group that are smaller or lesser than the requested size will be increased to the requested size.
- This feature will only resize additional block storages in an instance and not the root volume.
- Clusters and cluster services must be in running state before disk vertical scaling is performed.
- Current implementation does not support this feature through Cloudera UI; It is available only through Beta CDP CLI. To install Beta Cloudera, refer to Installing Beta CDP CLI.
- The disk modification feature on AWS can only be used once in 6 hours. This is a limitation on the AWS side.

Permissions

This feature requires the following permissions to be added to the cross-account policy described in Cross-account access IAM role.

- ec2:ModifyVolume
- ec2:DescribeVolumesModifications
- ec2:DescribeVolumeStatus

The following table explains why Cloudera needs these permissions:

Permission	Description
ec2:ModifyVolume	It is required to modify the volume attributes such as type, size and IOPS capacity. Without this, volume modifications cannot be performed by Cloudera.
ec2:DescribeVolumesModifications	It is required to verify whether the volume modifications performed by Cloudera were successful. Only upon successful modification, other steps like resizing will be done.
ec2:DescribeVolumeStatus	This is required to make sure that the volume being modified is attached to an instance and not an orphaned volume.

Modifying disks

The disk volumes attached to the instances in a host group can be modified using CLI commands.

Note: Do not scale your instance manually on the cloud provider side as this can possibly result in errors in the future. It is possible that Cloudera will be out of sync with the actual instance scale and a repair or upgrade could fail, or the previous scale could possibly be reinstated during a repair action. Use the Cloudera Control Plane process described here instead.

Modifying disks using CDP CLI

Use the following Beta CDP CLI command to modify the volumes attached to the instances in a host group. Replace the placeholders with actual values. For example *[***DATALAKE-NAME***]* should be replaced with an actual name. As part of this update, the instance-template parameter in the vertical scaling command has been made optional. But one of the instance-template or disk-options have to be provided.

Additional parameters for vertical scaling have been added to both the datalake and datahub commands. The modification request is sent as part of the --disk-options parameter.

The *[***VOLUME-TYPE***]* placeholder is for the type of volume the disks are being modified to. It is an optional field and should be added if the volume type has to be modified, in which case the type will not be modified.

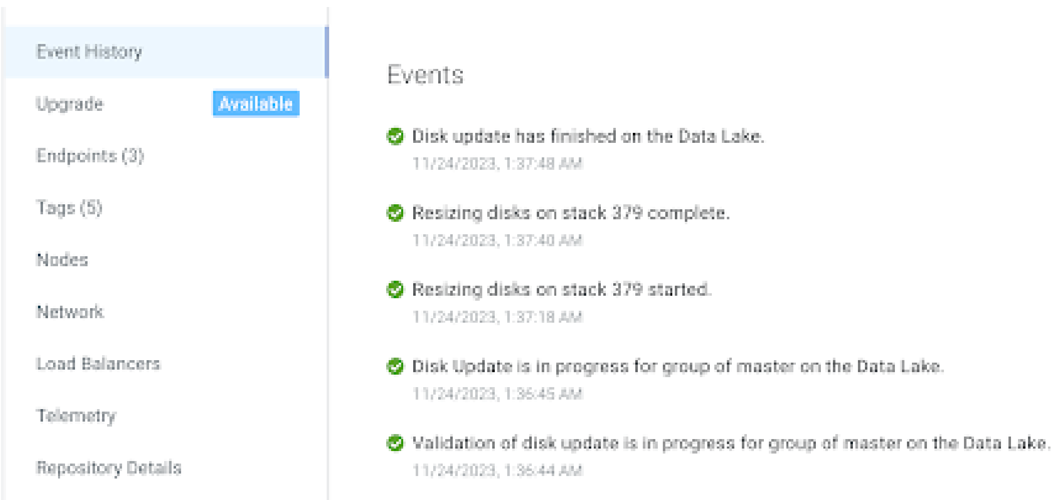
The *[***SIZE***]* placeholder is optional as well and is for the size the disks are being increased to in GB. Specify only the integer value.

```
//DATA LAKE
cdp datalake start-datalake-vertical-scaling
          --datalake [***DATALAKE-NAME***]
          --group [***INSTANCE-GROUP-NAME***]
          --disk-options modifyDisks="{volumeType=\"[***VOLUME-TYPE***]\"
,size=[***SIZE***]}"



//DATA HUB

cdp datahub start-cluster-vertical-scaling
          --datahub [***DATAHUB-NAME***]
          --group [***INSTANCE-GROUP-NAME***]
          --disk-options modifyDisks="{volumeType=\"[***VOLUME-TYPE***]\",
size=[***SIZE***]}"
```

Verifying that modification is complete

The change to the disk can be verified through Cloudera UI, AWS console, or by logging into the instances directly, after the flow is completed in the Event History.

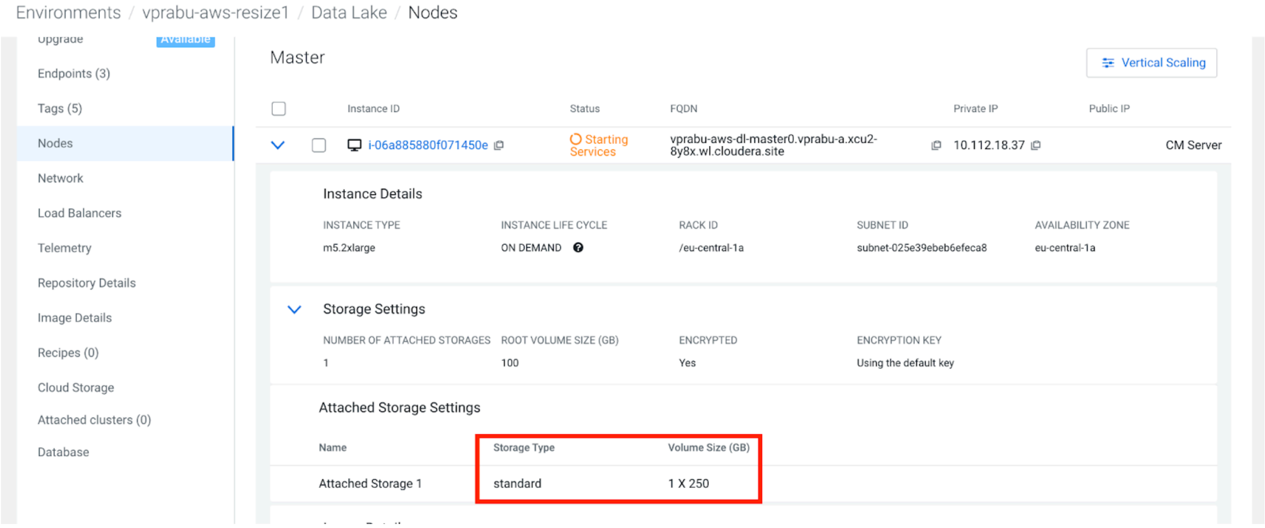
The following is a screenshot from Event History showing the completion of the disk update:



Verifying disk size/type in Cloudera UI

1. Navigate to the Cloudera Management Console Data Lake or Cloudera Management Console Data Hub
2. Click into the Data Lake or Cloudera Data Hub that was modified.
3. Click Nodes in the left hand tree.
4. Open the accordion of the instance group that was modified.
5. Open the Storage Settings accordion in any of the instances in the group.

The Storage Type and Volume Size are updated based on the request.

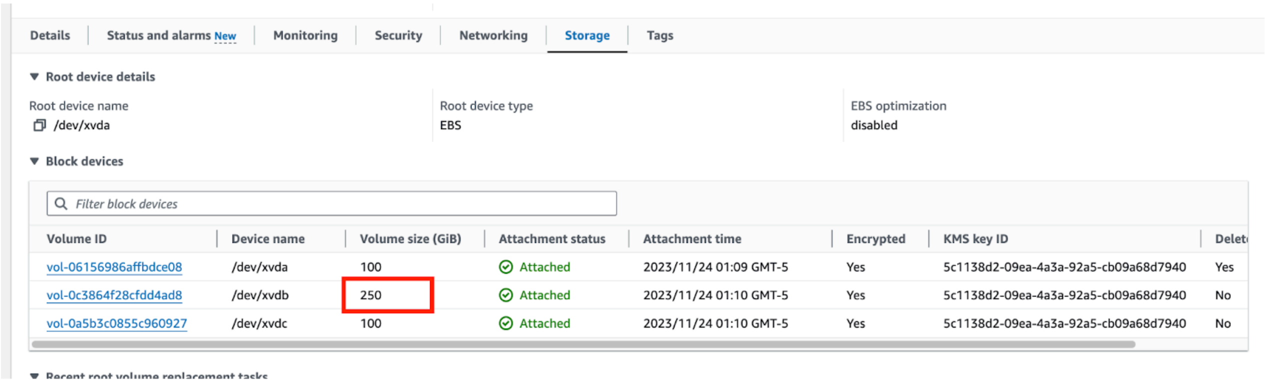


Verifying disk size/type in AWS Console

1. Navigate to the Cloudera Management Console Data Lake or Cloudera Management Console Data Hub
2. Click into the Data Lake or Cloudera Data Hub that was modified.
3. Click Nodes in the left hand tree.
4. Click on the AWS link for the instance and log into AWS Console.
5. Select the Storage tab for the instance.

The volume size is updated as per the request.

You can also click into an individual volume to make sure the type and size are modified correctly.



Adding disks

Additional volumes can be added to the instances in a host group using CLI commands.

In some cases when customers run out of memory, it is cheaper to add an additional volume instead of resizing to a bigger volume. This feature allows for adding additional block storages to an instance group with minimum interruption. Only the services running on the instance group are stopped, as the additional volume has to be mounted and configured for use on the instances.

Based on the request, 'N' number of disks of the same type and size can be added to the instance group.

Adding disks using CDP CLI

Use the following CDP CLI Beta command to add additional volumes attached to the instances in a host group. Replace the placeholders with actual values. For example, *[***DATALAKE-NAME***]* should be replaced with an actual name. As part of this update, the instance-template parameter in the vertical scaling command has been made optional. But one of the -- instance-template or -- disk-options have to be provided.

The addDisks request is sent as part of the --disk-options parameter. All parameters for the add disk input are required.

The *[***NUMBER_OF_DISKS***]* placeholder is for the number of volumes being added.

The *[***VOLUME-TYPE***]* placeholder is for the type of volume being added.

The *[***SIZE***]* placeholder is for the size the disks being added in GB. Specify only the integer value.

The cloudVolumeUsageType is the purpose for which the disk is being added; it is an enum field with either “GEN ERAL” or “DATABASE” value.

```
//DATALAKE

cdp datalake start-datalake-vertical-scaling --datalake=[***DATALAKE_NAME***
] --group=[***DATALAKE_INSTANCE_GROUP_NAME***] --disk-options addDisks="{num
berOfDisks=[***NUMBER_OF_DISKS_TO_ADD***], volumeType=\"[***TYPE_OF_COLUME_T
O_ADD***]\",size=[***SIZE_OF_VOLUME***],cloudVolumeUsageType=\"GENERAL\"|\"D
ATABASE\"}"

//DATAHUB
cdp datahub start-cluster-vertical-scaling --datahub=[***DATAHUB_NAME***] --
group=[***DATAHUB_INSTANCE_GROUP_NAME***] --disk-options addDisks="{numberO
fDisks=[***NUMBER_OF_DISKS_TO_ADD***], volumeType=\"[***TYPE_OF_COLUME_TO_AD
D***]\",size=[***SIZE_OF_VOLUME***],cloudVolumeUsageType=\"GENERAL\"|\"DATAB
ASE\"}"
```

Deleting disks

To save costs, you can delete volumes attached to instances in an instance group using CLI commands.

In cases where only compute services are being run on an instance group, any block storage attached to the instance is going to cost the customer even if the cluster is stopped. As compute services do not store any persistent data on attached volumes, having additional volumes to store temporary data is unnecessary. In such cases, customers can use this command to delete all attached volumes on instances in an instance group.

This can be done only for compute instance groups on only Data Hubs, as Data Lakes need persistent volumes. This command deletes all additional volumes for instances in an instance group.

Deleting disks using CDP CLI

Use the following CDP CLI Beta command to delete all attached additional volumes to the instances in a host group. Replace the placeholders with actual values. For example, *[***DATALAKE-NAME***]* should be replaced with an actual name. As part of this update, the instance-template parameter in the vertical scaling command has been made optional. But one of the -- instance-template or -- disk-options have to be provided.

The delete disks request is sent as part of the --disk-options parameter. All parameters for the add disk input are required.

The deleteDisks parameter accepts a boolean value, either true or false.

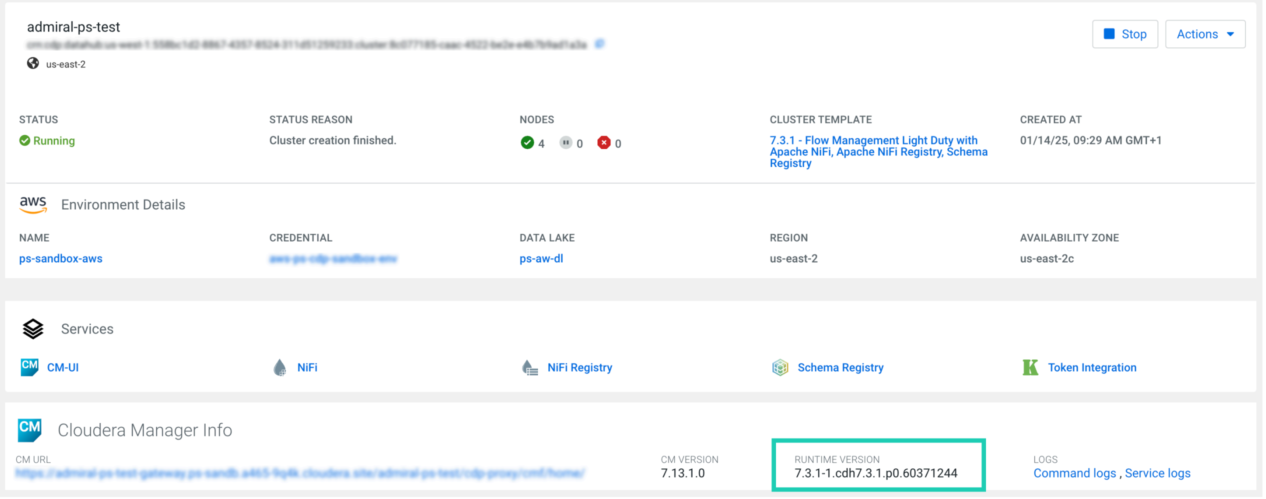
```
//DATAHUB

cdp datahub start-cluster-vertical-scaling --datahub=[***DATAHUB_NAME***] --
group=[***DATAHUB_INSTANCE_GROUP_NAME***] --disk-options deleteDisks=true|
false
```

Upgrading Cloudera Data Hub clusters

You can upgrade a Cloudera Data Hub cluster in one of three ways: Cloudera Runtime and Cloudera Manager major/minor version upgrades, service pack upgrades, and OS upgrades.

Cloudera Runtime and Cloudera Manager Major/Minor Version Upgrades

A Cloudera environment consists of both Data Lake and Cloudera Data Hub clusters, and currently these clusters should run the same major/minor versions of Cloudera Runtime and Cloudera Manager. This major/minor version is the first three digits of the Platform Version displayed along with the Cloudera Manager information:



In the image above, the Cloudera Runtime major/minor version is 7.3.1.

A Cloudera Data Hub major/minor version upgrade initiates an upgrade of the major/minor Cloudera Manager and Cloudera Runtime versions, as well as the required additional parcels (Spark3, Flink, Profiler, and Cloudera Flow Management). The root volumes, additional volumes, and the cluster state is retained. A major/minor version upgrade is an express upgrade where the services will not be available during the process.

A major/minor version upgrade also includes the OS upgrade. This means that after the Cloudera Runtime upgrade, the OS upgrade is automatically triggered, and the OS/VM packages and other Cloudera packages, such as Salt, will be updated as part of the Cloudera Data Hub upgrade.

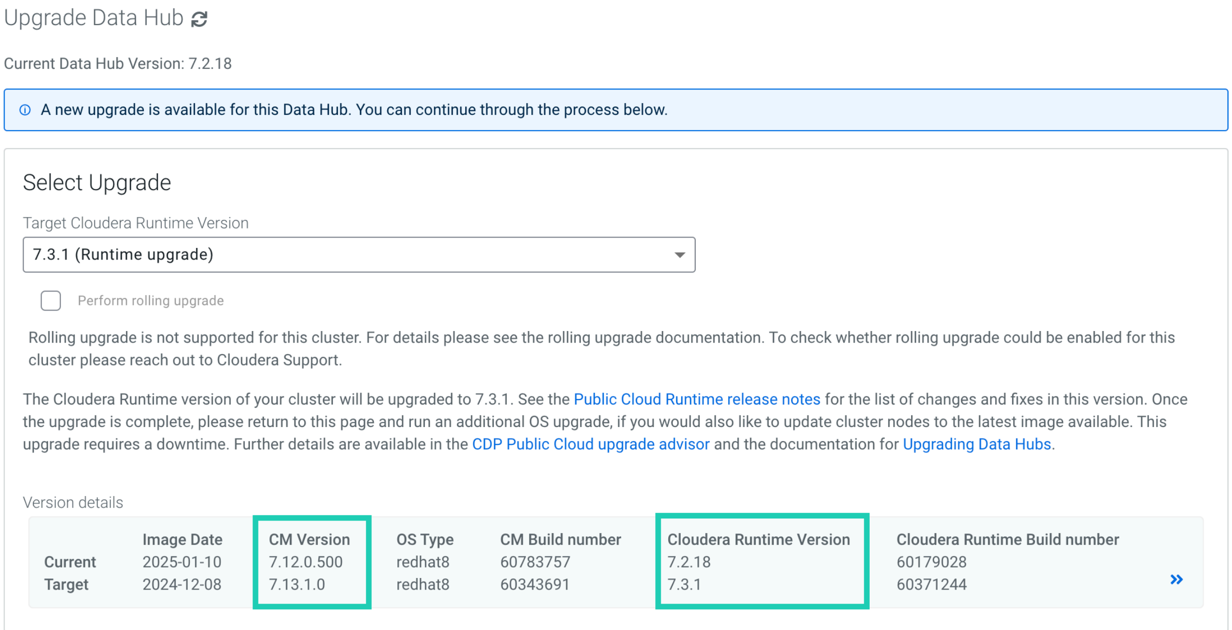
Note: In the following cases, the OS upgrade is only performed automatically for the Gateway nodes:

- When upgrading a Cloudera Operational Database cluster
- When rolling upgrade is requested for large clusters (more than 20 nodes)

A major/minor version upgrade does not redistribute the services according to the cluster template of the newer version. Nor does it add new services according to the new cluster template. Major/minor version upgrades do not add new service configurations in the new cluster template. These additional configurations have to be applied manually.

This is not a zero-downtime upgrade, and there will be service outages during the upgrade.

When a major/minor version upgrade is available, you will be able to select the target version for upgrade from the Upgrade tab at the bottom of the Data Hub details page:



For supported versions and templates, see *Support matrix for major/minor upgrades*.

Note: Upgrading a custom template is supported, if the template contains only services that are included in any supported built-in templates.

If your cluster contains a custom service, such as a third-party Cloudera Manager service, contact Cloudera support for an entitlement that will allow you to upgrade the cluster. Cloudera cannot support failed upgrades for clusters that run unsupported services.

Always test any cluster upgrade in a development environment first.

Service pack upgrades

The service pack upgrade process checks to see if a new Cloudera Runtime or Cloudera Manager service pack is available, and then upgrades the Cloudera Data Hub cluster to the newest builds. Service pack upgrades do not upgrade to a new major/minor version of Cloudera Runtime and Cloudera Manager; they only upgrade to the latest build of a service pack version.

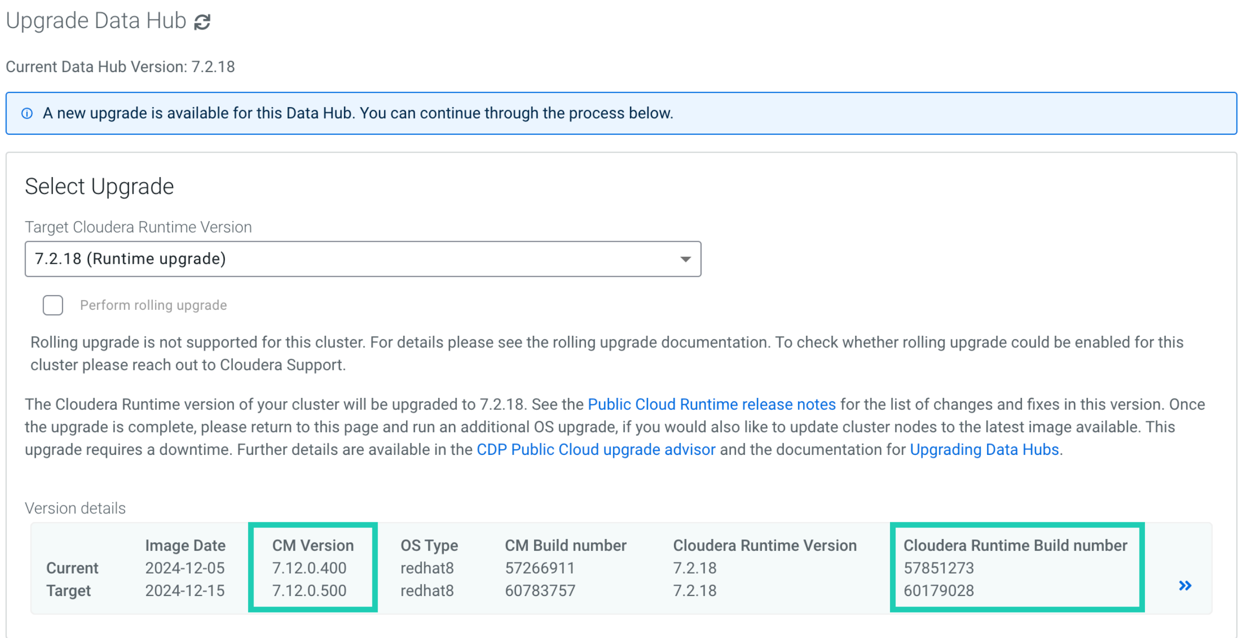
A service pack upgrade can be conducted on a single Cloudera Data Hub cluster in an environment, or multiple Cloudera Data Hub clusters. You can perform a service pack upgrade independent of a Data Lake upgrade.

A service pack upgrade also includes the OS upgrade. This means that after the Cloudera Runtime upgrade, the OS upgrade is automatically triggered, and the OS/VM packages and other Cloudera packages, such as Salt, will be updated as part of the Cloudera Data Hub upgrade.

Note: In the following cases, the OS upgrade is only performed automatically for the Gateway nodes:

- When upgrading a Cloudera Operational Database cluster
- When rolling upgrade is requested for large clusters (more than 20 nodes)

When a service pack upgrade is available, you will be able to select the target version (which is the same as the current version) for upgrade from the Upgrade tab at the bottom of the Data Hub details page. When you select the target version, note that the Cloudera Manager/Cloudera Runtime versions are the same, but the build numbers differ:



Service pack upgrades are available from Cloudera Runtime 7.2.7 onward for RAZ and non-RAZ Cloudera Data Hub clusters.

Important: The Cloudera Runtime 7.2.16.3 service pack has new dependencies that were not present in most of the previous Cloudera Runtime versions. Because of this, you may be unable to upgrade directly to this service pack version and future service packs without first upgrading to a more recent Cloudera Runtime version.

If you plan to upgrade your existing Cloudera Data Hub clusters from a previous release to 7.2.16.3 or later, you can verify whether or not you will first be required to perform an additional upgrade step:

1. Select the Cloudera Data Hub 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 the latest 7.2.16 service pack.

If your current Cloudera 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 the 7.2.16.3 service pack.

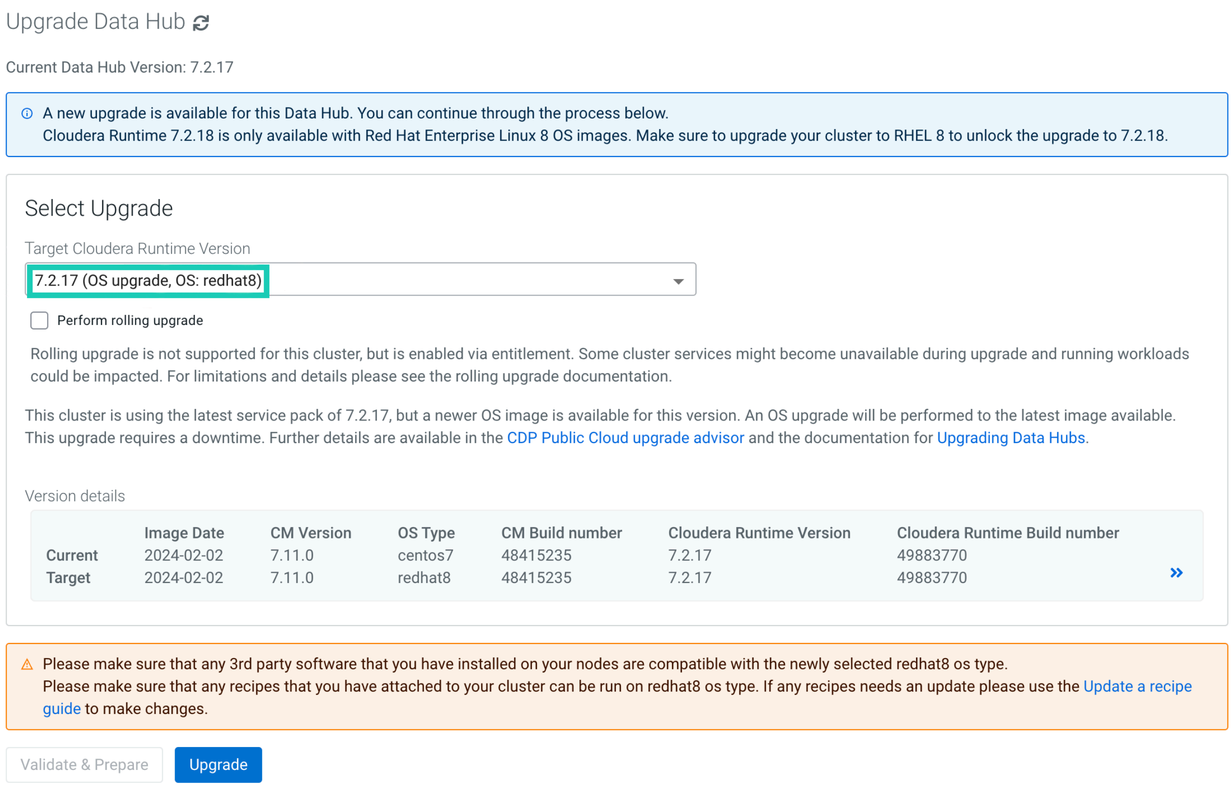
If your current Cloudera Runtime version is 7.2.12, 7.2.14, or 7.2.15, these steps may include upgrading to the most recent service pack of your current Cloudera Runtime version, as well as performing a separate OS upgrade for your current Cloudera Runtime version, before you can perform a major/minor version upgrade to 7.2.16.3.

OS Upgrades

An OS upgrade updates the OS and VM packages to those available in the latest pre-warmed image. This is done by replacing the VM, re-attaching the attached volumes, and restarting the services. In the process, the data on root volume (for example, parcels and service logs) is lost. On larger clusters, test the OS upgrade in a development environment as the upgrade may exceed some of the cloud resource limits.

An OS upgrade does not upgrade the platform version (Cloudera Manager, Cloudera Runtime, and additional parcels). You can perform an OS upgrade independent of a Cloudera Data Hub cluster upgrade. The OS upgrade triggers the execution of any pre-service-deployment, post-cluster-manager-start, or post-service-deployment recipes.

If an OS upgrade is available, it will appear in the Upgrade Data Hub menu with “(OS Upgrade, OS: <target-OS>)” when you select a Cloudera Runtime version:



Rolling upgrades

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

Limitations

- With the exception of rolling upgrades, Cloudera Data Hub upgrades are not a zero-downtime upgrade and service outages will occur.
- Ranger Authorization: Cloudera Data Hub service pack upgrades with RAZ are supported only for Cloudera Runtime versions 7.2.7+. Major/minor version upgrades with RAZ are supported only for Cloudera Runtime versions 7.2.10-7.2.12 to versions 7.2.14+.
- During an OS upgrade, any data on the root volume (parcels, service logs, custom software) will be lost. For older clusters (created before March 26, 2021), OS upgrade is not available when the embedded Cloudera Manager database is on the root volume.
- Cloudera Operational Database cannot be upgraded through the Cloudera Data Hub user interface and must be upgraded through the Cloudera beta CLI. For more information see *Upgrading Cloudera Operational Database*.
- When upgrading the Data Lake from Cloudera Runtime 7.2.17 to 7.2.18 or 7.3.1, if Iceberg metadata is required to be captured in Atlas, then do not use mixed versions (7.2.18 or 7.3.1 Data Lake with 7.2.17 Cloudera Data Hub), as the mismatch in the Iceberg models between 7.2.17 and 7.2.18 or 7.3.1 can cause problems.
- A Cloudera Data Hub cluster must be using Cloudera Runtime 7.2.17 in order to be eligible for CentOS to RHEL upgrade. If you are not seeing the option to upgrade to RHEL, ensure that your cluster is running Cloudera Runtime 7.2.17.

Prerequisites

- There is required downtime of the environment during upgrades, so plan the upgrade accordingly.
- Verify using the *Support matrix for Cloudera Data Hub upgrades* that all Cloudera Data Hub clusters in the environment are a type and version supported for major/minor upgrades.

- Test your applications against the new platform version in a separate environment before the Cloudera Runtime/Cloudera Manager upgrade to ensure application compatibility with the new platform version.
- The Data Lake and all the Cloudera Data Hub clusters in an environment must be upgraded to the same major/minor version, with the exception of Cloudera Data Hub clusters on Cloudera Runtime version 7.2.16+, which are compatible with newer versions of the Data Lake (7.2.17+, because the Data Lake must always run a higher Cloudera Runtime version). If you plan to upgrade your Cloudera Data Hub clusters to a later major/minor version, you must first backup and then upgrade the Data Lake.
- Verify that any Experiences you use are running the latest version available.
- Before being able to upgrade to Cloudera Runtime 7.3.1, you need to upgrade Data Lakes and all Cloudera Data Hub databases to a to PostgreSQL 14.
- If the upgrade involves upgrading from CentOS to RHEL, review the Prerequisites for upgrading from CentOS to RHEL.

Related Information

Data Lake upgrade

Support matrix for Cloudera Data Hub upgrades

Upgrading Cloudera Operational Database clusters

Support matrix for Cloudera Data Hub upgrades

Cloudera Data Hub upgrades are supported only for the templates and Cloudera Runtime versions listed below.

Template	Platform versions that you can upgrade from	Platform versions that you can upgrade to
DE, DE HA, DE Spark3	7.1.0 and above	7.2.11 and above
Operational Database	7.2.0 and above	7.2.11 and 7.2.12
Streams Messaging	7.2.0 and above	7.2.11 and above
Flow Management	7.2.0 and above	7.2.11 and above
Data Discovery	7.2.0 and above	7.2.11 and above
Datamart	7.2.2 and above	7.2.11 and above
Real-Time Datamart	7.2.2 and above	7.2.11 and above
Streaming Analytics (*See warning below)	7.2.10 until 7.2.14	7.2.11 until 7.2.15
Cloudera Data Catalog	Not supported. Use backup and restore	
Cloudera Operational Database	7.2.7 and above	7.2.11 and above

Warning:

The following upgrade paths are not supported for the Streaming Analytics cluster:

- from 7.2.11 to 7.2.12
- from any version to 7.2.16

For information about how to migrate your Flink jobs to a cluster with a new version, see the Migrating Flink jobs documentation.

Ranger Authorization (RAZ) Upgrades

Major/minor version upgrades with RAZ are supported only for Cloudera Runtime versions 7.2.10-7.2.12 to 7.2.14+.

Service Pack Upgrades

Cloudera Data Hub service pack upgrades, both RAZ and non-RAZ, are supported only for Cloudera Runtime versions 7.2.7+.

Note: Upgrading a custom template is supported, if the template contains only services that are included in any supported built-in templates.

If your cluster contains a custom service, such as a third-party Cloudera Manager service, contact Cloudera support for an entitlement that will allow you to upgrade the cluster. Cloudera cannot support failed upgrades for clusters that run unsupported services.

Always test any cluster upgrade in a development environment first.

Backing up before an upgrade

The Cloudera Data Hub cluster upgrade is an in-place upgrade and the data on attached volumes will be retained. The following section outlines data stored in the Cloudera Data Hub clusters by different services. To prepare for a worst-case scenario where the cluster can't be restored, retain the data in these clusters. Also take note of any cluster configurations updated through Cloudera Manager.

Service configurations in Cloudera Manager

- Any service configurations that you have updated since the cluster creation. Typically these can be found by selecting the “Non-default values” filter on the **Configuration** menu in Cloudera Manager.

Custom drivers

Take a backup of any custom drivers for services running on Cloudera Data Hub. An OS upgrade will delete the root volume and you will not be able to recover the driver if a backup is not taken.

Data Engineering clusters

The underlying data is in cloud storage. However, the following data is specific to each Cloudera Data Hub cluster:

- Oozie - workflow and coordinator definitions
- Hue - saved queries
- Hive UDFs and Spark applications
- Oozie custom sharelibs

A script is available for backing up and restoring the Oozie and Hue databases in clusters created with the Data Engineering and Data Engineering HA templates. Execute the commands from the Cloudera Manager node. Run the following:

```
sudo su
chmod +x /srv/salt/postgresql/disaster_recovery/scripts/backup_and_restore_d
h.sh
```

Then, for backup run:

```
/srv/salt/postgresql/disaster_recovery/scripts/backup_and_resto
re_dh.sh -b <s3 or abfs path> <database names>
```

For example:

```
/srv/salt/postgresql/disaster_recovery/scripts/backup_and_resto
re_dh.sh -b s3a://my/test/path oozie hue
```

For restoring after an upgrade, run:

```
/srv/salt/postgresql/disaster_recovery/scripts/backup_and_resto
re_dh.sh -r <s3 or abfs path> <database names>
```

For example:

```
/srv/salt/postgresql/disaster_recovery/scripts/backup_and_resto
re_dh.sh -r s3a://my/test/path oozie hue
```

Streams Messaging clusters

Backing up and restoring data

Data stored on volumes:

- Zookeeper: /hadoopfs/fs1/zookeeper (only on the first attached volume)
- Kafka: /hadoopfs/fsN/kafka (for on all attached volumes, N stands for volume index)

Data stored in database:

- Schema Registry schemas

Data stored in S3/ADLS:

- Schema Registry serde jars

A script is available for backing up and restoring the Schema_registry and SMM databases in clusters created with the Streams Messaging Light Duty and Streams Messaging Heavy Duty templates. Run the following:

```
sudo su
chmod +x /srv/salt/postgresql/disaster_recovery/scripts/backup_and_restore_d
h.sh
```

Then, for backup run:

```
/srv/salt/postgresql/disaster_recovery/scripts/backup_and_resto
re_dh.sh -b <s3 or abfs path> <database names>
```

For example:

```
/srv/salt/postgresql/disaster_recovery/scripts/backup_and_resto
re_dh.sh -b s3a://my/test/path smm schema_registry
```

For restoring after an upgrade, run:

```
/srv/salt/postgresql/disaster_recovery/scripts/backup_and_resto
re_dh.sh -r <s3 or abfs path> <database names>
```

For example:

```
/srv/salt/postgresql/disaster_recovery/scripts/backup_and_resto
re_dh.sh -r s3a://my/test/path smm schema_registry
```

Backing up and restoring Kafka Connect secret password and salt

On a Streams Messaging Cloudera Data Hub cluster, you need to manually reset the Kafka Connect secret password and salt in Cloudera Manager and save them for later so that you can manually reset them again when you need to recreate the cluster from scratch. Otherwise, Kafka Connect can go into a bad health state. The only way for you to know this password and salt is to reset them and manually record them somewhere.

Set new password and salt values in Cloudera Manager for the below values and Restart Kafka Connect to ensure the new password and salt is used for encryption:

- kafka.connect.secret.global.password
- kafka.connect.secret.pbe.salt

Save the password and salt for later use because you will need them if you need to recreate the cluster.

Resetting Kafka Connect secret password when recreating a cluster from backup

After recreating a Cloudera Data Hub cluster, you need to manually reset the Kafka Connect password and salt values to the values used on the original cluster.

1. Restart Kafka Connect after the Cloudera Data Hub cluster is recreated.
2. Set the following password and salt values to the ones that you have saved earlier for the original cluster:
    - kafka.connect.secret.global.password
    - kafka.connect.secret.pbe.salt
3. Restart Kafka Connect again.

Flow Management clusters

Verify that you have committed all of your flows, then backup the following files/directories into your S3 backup location:

- nifi.properties - on any NiFi node, run: ps -ef | grep nifi.properties

  This will indicate the path to the file. For example:

  -Dnifi.properties.file.path=/var/run/cloudera-scm-agent/process/1546335400-nifi-NIFI_NODE/nifi.properties
- bootstrap.conf - on any NiFi node, run: ps -ef | grep bootstrap.conf

  This will indicate the path to the file. For example:

  -Dorg.apache.nifi.bootstrap.config.file=/var/run/cloudera-scm-agent/process/1546335400-nifi-NIFI_NODE/bootstrap.conf
- Zookeeper data for /nifi znode. From a NiFi node:

```
/opt/cloudera/parcels/CFM-<CFMversion>/TOOLKIT/bin/zk-migrator.sh -r -z
<zk node>:2181/nifi
          -f /tmp/zk-data.json
```

  If a Zookeeper node is co-located with NiFi (light duty template, it’s possible to use localhost:2181), then back up the created zk-data.json file.
- NiFi Registry data. The versioned flows are stored in an external database provisioned in your cloud provider. Perform a backup of this database following the instructions of your cloud provider and restore the backup in the new database.
- /hadoopfs/fs4/working-dir/ (for the NiFi nodes) - without the ‘work’ directory inside it. This directory contains, for example, the local state directory as well as the flow.xml.gz which represents the flow definitions.

  Note: On Cloudera Runtime versions lower than 7.2.10, this directory is located in /var/lib/nifi/working-dir.
- /hadoop/fs1/working-dir/ (for the Management node).

  Note: On Cloudera Runtime versions lower than 7.2.10, this directory is located in /var/lib/nifiregistry/working-dir.
- Any other “custom” directory, for example directories where client configs and JDBC driver are located.

Data Discovery & Exploration clusters

Complete the following steps to backup DDE clusters:

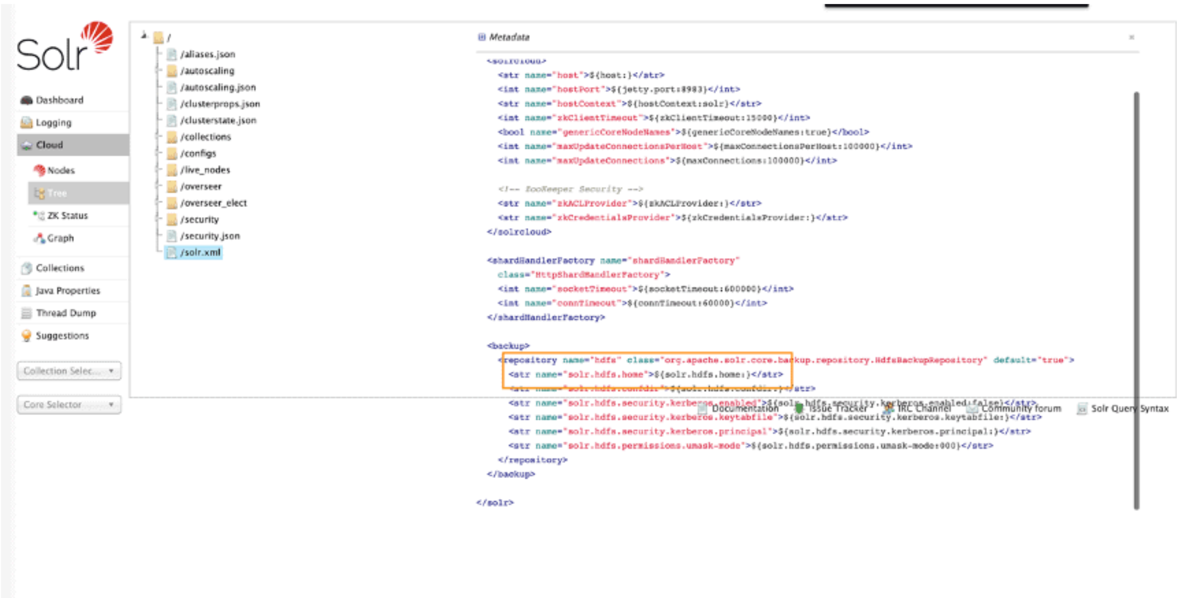
- Assign the datalake-admin-bucket policy to RANGER_AUDIT_ROLE.
- Create a subfolder for this Cloudera Data Hub cluster inside your S3 backup location.
- Log in to the gateway node of the Cloudera Data Hub cluster.
- kinit with the proper user keytab if needed.

- Follow this Backing Up and Restoring Solr Collections with these exceptions:
    - Use solrctl --get-solrxmlto get the solr.xml file.

      Edit it by changing your backup repository solr.hdfs.home parameter (see image) to your backup location (for example: s3a://datalake-bucket/backup/solr).

      User solrctl --put-solrxml to publish your edited solr.xml.

      Restart Solr.



    - You do not need to launch the prepare-backup command; instead launch create-snapshot and export-snapshot for each collection.
    - For export-snapshot, specify your destination as s3a://[your backup location]/[your data hub subfolder] (for example: s3a://datalake-bucket/backup/solr).
    - Verify on the YARN history server that all of the executions succeeded. If not, troubleshoot and relaunch any export-snapshot commands that failed.

Operational Database clusters

Complete the following steps to backup Operational Database clusters:

- Create a subfolder for this Cloudera Data Hub cluster inside of your S3 backup location.
- Ensure snapshots are enabled on Cloudera Manager. They are enabled by default.
- Log in to the gateway node of the Cloudera Data Hub cluster.
- kinit with the proper user keytab if needed.
- Launch hbase shell.
- Issue the snapshot command for each of the tables that you want to back up (for example: snapshot ‘mytable’, ‘mytable-snapshot’)
- Exit hbase shell.
- Launch ExportSnapshot command for each of the snapshots you created, for example:

```
hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot mytable-
snapshot -copy-to
        s3a://datalake-bucket/backup/hbase -mappers 10
```

- Verify on the YARN history server that all of the executions succeeded. If not, troubleshoot and relaunch any ExportSnapshot commands that failed.

Performing a Cloudera Data Hub major/minor version cluster upgrade

A major/minor version upgrade initiates an upgrade of the major/minor Cloudera Manager and Cloudera Runtime versions, as well as the required additional parcels (Spark3, Flink, Profiler, and Cloudera Flow Management). After you perform a major/minor upgrade on a Data Lake, you should in most cases perform a major/minor version upgrade on each Cloudera Data Hub cluster attached to the Data Lake. The Cloudera Data Hub clusters must run the same major/minor Cloudera Runtime version as the Data Lake--with the exception of Cloudera Data Hub clusters on Cloudera Runtime 7.2.16+, which are compatible with Data Lake versions 7.2.17+.

About this task

Note that for major/minor version upgrades, the cluster can only be upgraded to the same major/minor version as the Data Lake, so you must first upgrade the Data Lake. For instructions, see *Data Lake upgrades*.

Complete the steps for each Cloudera Data Hub cluster that you are upgrading.

Required role: DatahubAdmin or Owner over the Cloudera Data Hub cluster

Before you begin

Procedure

1. Start the cluster.
2. Before you begin the Cloudera Data Hub cluster upgrade, check if the current version of Cloudera Runtime is <= 7.1.0.0. If yes, then verify the memory settings for Cloudera Manager in the /etc/default/cloudera-scm-server file on the Cloudera Manager server host. If you find the value “-Xmx2G” in CMF_JAVA_OPTS, update it to “-Xmx4G” and restart the Cloudera Manager server after the change.
3. For clusters that contain the Hive service in Cloudera Runtime versions prior to version 7.2.2, it is required to terminate all running YARN applications before starting the upgrade. So, if the current Cloudera Runtime version is < 7.2.2:
   a) SSH to any Hive node (master or worker).
   b) Get an initial ticket-granting ticket for YARN principal. This passes the Kerberos authentication so that you can run the YARN application in shell. YARN principals are installed in /run/cloudera-scm-agent/process/xxxxx-yarn-RESOURCEMANAGER/yarn.keytab

      Enter the directory and run klist command to display the Kerberos principals in YARN keytab (sample command):

      klist -kt yarn.keytab

      Sample Kerberos principal output: <format: userid/host@domain>

      ```
      yarn/nightly-7x-1-1.nightly-7x-1.root.hwx.site@ROOT.HWX.SITE
      ```

      From the above directory, run kinit command in this format (sample command): kinit -kt yarn.keytab <kerberos-principal format: userid/host@domain>

      ```
      kinit -kt yarn.keytab yarn/nightly-7x-1-1.nightly-7x-1.root.hwx.site@ROO
      T.HWX.SITE
      ```
   c) Run the command: yarn application -list
   d) For each running YARN application, run the command: yarn application -kill <appId>
   e) Run the command (to verify that no apps are running): yarn application -list

4. For Streams Messaging clusters, if you are upgrading from Cloudera Runtime version 7.2.12 to 7.2.14, complete the following steps:
    a) Open the Cloudera Manager UI for the Streams Messaging cluster.
    b) Click on the Cruise Control service, then click on Configurations.
    c) Search for 'RackAwareGoal' in the search bar and remove the entry for 'com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareGoal' for 'Default goals,' 'Hard Goals,' 'Support Goals,' ‘Self-Healing Goals, and 'Anomaly Goals.'
    d) Save the changes and restart the Cruise Control service.
    e) Proceed with the upgrade, but note that once the upgrade is complete, add the entries back to 'Default goals', 'Hard Goals', 'Support Goals,' ‘Self-Healing Goals,’ and 'Anomaly Goals.' This time rename the value to 'com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareDistributionGoal'
5. If your cluster uses Streams Replication Manager, export or migrate aggregated metrics.

    In Cloudera Runtime 7.2.18, major changes are made to the internal Kafka Streams application of SRM. As a result, SRM by default loses all aggregated metrics that were collected before the upgrade. This means that you will not be able to query metrics with the SRM Service REST API that describe the pre-upgrade state of replications. If you want to retain the metrics, you can either export them, for archival purposes, or migrate them to the new format used by SRM. If you do not need to retain metrics, you can skip this step and continue with the upgrade.

    Exporting metrics creates a backup of the metric data, however, exported metrics cannot be imported into the SRM Service for consumption. As a result, exporting metrics is only useful for data archival purposes.

    Migrating metrics can be done in two different ways depending on whether you are doing a rolling upgrade or a non-rolling upgrade.

    - In case of a non-rolling upgrade, migration happens following the upgrade. In this case, the new version of the internal Kafka Streams application running in the upgraded cluster starts to process historical metrics as soon as it is online. However, until the metrics are processed, the SRM Service cannot serve requests regarding latest metrics and returns empty or missing responses on its REST API. The duration of this downtime depends on the number SRM Service instances and the amount of metrics in the cluster.
    - In case of a rolling upgrade, a migration process called SRM Service Migrator is initiated during the upgrade. The Migrator processes existing metrics so that they become compatible with your upgraded cluster. Depending on the size of your cluster and the amount of metrics you have, this process may take up to multiple hours to finish.

    For Export metrics

    Use the following endpoints of the SRM Service REST API to export metrics.

    If upgrading from Cloudera Runtime 7.2.12 or higher:

    - `/v2/topic-metrics/{source}/{target}/{upstreamTopic}/{metric}`
    - `/v2/cluster-metrics/{source}/{target}/{metric}`

    If upgrading from Cloudera Runtime 7.2.11 or lower:

    - `/topic-metrics/{topic}/{metric}`
    - `/cluster-metrics/{cluster}/{metric}`

    For more information regarding the SRM Service REST API, see Streams Replication Manager Service REST API or Streams Replication Manager REST API Reference.

    For Non-rolling upgrade migration

    a. In Cloudera Manager, select the SRM service.
    b. Go to Configuration.
    c. Add the following to the SRM Service Environment Advanced Configuration Snippet (Safety Valve) property:

    ```
    Key: SRM_SERVICE_SKIP_MIGRATION
    ```

```
Value: false
```

For Rolling upgrade migration

Note: All SRM Service role hosts experience increased resource utilization during a rolling upgrade if you enable migration. This is because the migration process uses the same system configurations (for example, heap size) as the SRM Service role.

a. Ensure that the target clusters of the SRM Service are available and healthy.

If a target cluster is unavailable, the upgrade will fail. As a result, if a target cluster is unavailable, or you expect a target cluster to become unavailable during the upgrade, remove it from SRM's configuration for the duration of the upgrade. Metrics in the target clusters that you remove are not migrated. Target clusters are specified in Streams Replication Manager Service Target Cluster.

b. In Cloudera Manager, select the SRM service and go to Configuration.

c. Add the following to the SRM Service Environment Advanced Configuration Snippet (Safety Valve) property:

```
Key: SRM_SERVICE_SKIP_MIGRATION
Value: false
```

d. Fine-tune the behavior of the migration process.

The SRM Service Metrics Migrator (the migration process) has a number of user configurable properties. Fine tuning the configuration can help in reducing the time it takes to migrate the metrics.

These properties do not have dedicated entries in Cloudera Manager, instead you must use SRM Service Advanced Configuration Snippet (Safety Valve) for srm-service.yaml to configure them. If you are unsure about configuration, skip configuration and continue with the next step.

Table 1: SRM Service Migrator properties and recommendations

Property	Default Value	Description	Cloudera Recommendations
streams.replication.manager.migrator.monitor.timeout.ms	3,600,000	The time in milliseconds after the Streams Replication Manager (SRM) Service Metrics Migrator times out.	Set this timeout to a value that is higher than the expected migration time. Cloudera recommends a value that is at least three times the expected migration time. The migration time depends on the amount of metrics in your deployment. The higher the number of metrics, the longer the migration process, and the higher this property must be set. For example, a deployment with 10,000 partitions (100 topics with a 100 partitions each) and a 2 hour retention period produces, at minimum, 30,000 metrics per metric emission cycle. In a case like this, migration takes around 10 minutes to finish.

Property	Default Value	Description	Cloudera Recommendations
streams.replication.manager.migrator.monitor.backoff.ms	120,000	The frequency at which the progress of the Streams Replication Manager (SRM) Service Metrics Migrator is checked.	The recommended values for this property differ depending on the version you are upgrading from. If upgrading from 7.2.12 or higher: Set this property to a value that is identical with or similar to the interval set in SRM Service Streams Commit Interval The default value of this property is identical with the default value of SRM Service Streams Commit Interval. If upgrading from 7.2.11 or lower: Set this property to 30,000 (30 seconds).
streams.replication.manager.migrator.monitor.stop.delay.ms	60,000	The amount of time in milliseconds that the Streams Replication Manager (SRM) Service Metrics Migrator processes metrics after the streams application is considered caught up	
streams.replication.manager.migrator.monitor.min.consecutive.successful.checks	3	The number of consecutive checks where the lag must be within the configured threshold to consider the Streams Replication Manager (SRM) Service Metrics Migrator successful. All target clusters of the SRM Service must be caught up for a check to be successful.	

Property	Default Value	Description	Cloudera Recommendations
streams.replication.manager.migrator.monitor.max.offset.lag	Calculated automatically if left empty.	The amount of offsets that the streams application in the Streams Replication Manager (SRM) Service Metrics Migrator is allowed to lag behind and still be considered up to date. When left empty, the migration logic will automatically calculate the maximum offset lag based on the Kafka Streams application configuration and the amount of metrics messages. Low offset lag values result in more up-to-date metrics processing following the upgrade, but also increase the time required for the upgrade to finish.	An appropriate value for this property is calculated automatically if the property is left empty. As a result, Cloudera recommends that you leave this property empty and use the automatically calculated value.

e. Click Save Change.

f. Restart the SRM service.

6. If you use autoscaling, disable autoscaling on the cluster.

7. Upgrade the cluster. To upgrade the cluster with the UI:

a) In the left-hand menu, click Data Hub Clusters and then select the Cloudera Data Hub cluster to upgrade.

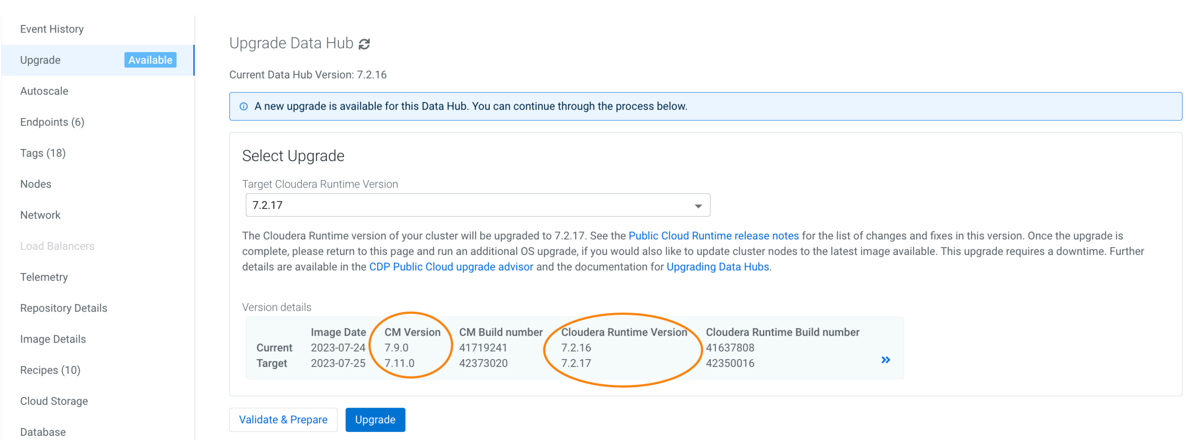
b) Scroll to the bottom of the Data Hub details page and select the Upgrade tab.

c) From the drop-down menu, select the Target Runtime Version.

d) 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 Cloudera Runtime versions, the Cloudera Data Hub template, and the Cloudera Data Hub OS. See Cloudera Data Hub rolling upgrades for more information.

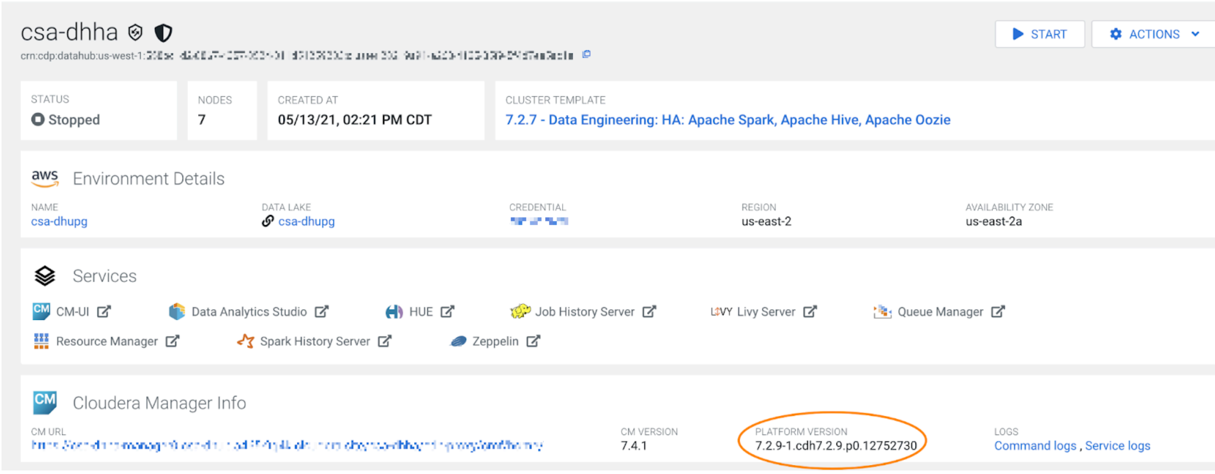
e) Click Validate and Prepare to check for any configuration issues and begin the Cloudera Runtime parcel download and distribution. Using the validate and prepare option does not require downtime and makes the maintenance window for an upgrade shorter. Validate and prepare also does not make any changes 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.

f) When the validate and prepare process is complete, click Upgrade.



8. Monitor the upgrade progress using the Cloudera Data Hub **Event History** tab.

9. When the upgrade is complete, verify the new version. Note that the new version is reflected in the Platform Version, and not the version displayed in the cluster template name (which will remain the same):



10. If you disabled autoscaling on the cluster, you can re-enable it after upgrade.

What to do next

If the upgrade is successful, proceed to the topic *Post-upgrade tasks*. If the upgrade fails, check the *Troubleshooting* section and re-try the upgrade.

Important: Once you have upgraded the Data Lake and started the Cloudera Data Hub upgrade process, you must repeat the upgrade procedure for every Cloudera Data Hub cluster in the environment. Do not conduct a major/minor version update asynchronously.

Related Information

Data Lake upgrade

Troubleshooting upgrade

Post-upgrade tasks

After a successful major/minor version upgrade, complete the following tasks if they apply to the cluster you upgraded.

Flow Management clusters

If doing a software-only upgrade for your Flow Management Cloudera Data Hub clusters and later repairing one of the NiFi nodes after the upgrade, you may be in a situation where the JDK used by NiFi is not the same across the nodes. This may cause issues in the NiFi UI and you may get an "Unexpected error" message.

After you upgrade a Flow Management cluster and repair a node, ensure that the same JDK is used across the NiFi nodes and if there is a JDK version mismatch, manually upgrade the JDK to match the JDK version being installed on the node that has been repaired.

Data Engineering upgrades to 7.2.11

Upgrading a cluster with the Spark service from Cloudera Runtime version 7.2.6 or 7.2.7 to version 7.2.11 may cause Spark cluster mode to fail. To workaround this issue, add the following configuration to the /etc/spark/conf/atlas-application.properties file: atlas.kafka.sasl.kerberos.service.name=kafka

Data Engineering HA/Hue HA

If you upgrade a template (such as Data Engineering HA) that contains Hue in an HA configuration, add the Hue load balancer hostnames to Cloudera Manager configurations: Cloudera Manager > Hue > Configuration > knox_proxyhosts. If you do not, you will experience errors preventing users from logging in.

Streams Messaging clusters

Clusters that contain the Streams Replication Manager service require a configuration change following a successful upgrade from Cloudera Runtime 7.2.11 or lower to 7.2.12 or higher. You must configure SRM to use its latest internal changelog data format and intra cluster hostname format. If this configuration is not completed, the SRM Service will not be able to target multiple clusters.

This post upgrade step is required because during an upgrade, SRM is configured to use a legacy version of its changelog data format and intra cluster hostname format. This is done to ensure backward compatibility so that if necessary, a rollback is possible.

Complete the following steps:

1. Verify that the SRM Service is up and running. This can be done by testing the REST API endpoints using the Swagger UI. If there are any issues, and a rollback is necessary, complete the rollback. No backward incompatible changes occurred up until this point.
2. In Cloudera Manager, select the Streams Replication Manager service.
3. Go to Configuration.
4. Find and disable the following properties:
   a. Use Legacy Internal Changelog Data Format
   b. Use Legacy Intra Cluster Host Name Format
5. Restart Streams Replication Manager.

Reset Kafka Connect secret password

After a Data Hub upgarde, you need to manually reset the Kafka Connect password and salt values to the values used before the upgrade was started. This step is required because these values are changed during the upgrade process but unless they are again set to the same values as before, Kafka Connect will go into bad health.

1. Restart Kafka Connect after the Data Hub has been recreated.
2. Set the following password and salt values to the ones that you have saved before starting the Data Hub upgrade process:
   - kafka.connect.secret.global.password
   - kafka.connect.secret.pbe.salt
3. Restart Kafka Connect again.

Configure Schema Registry to use V2 of its fingerprinting mechanism after upgrade to 7.2.18:

Following an upgrade to Cloudera Runtime 7.2.18 or later, Cloudera recommends that you configure Schema Registry to use fingerprinting V2. Fingerprinting V2 resolves an issue in V1 where schemas were not created under certain circumstances. For more information on the original issues as well as Schema Registry fingerprinting, see TSB-713. Note that even if you switch to V2, some issues might still persist, see TSB-718 for more information.

1. Access the Cloudera Manager instance managing your Cloudera Data Hub cluster.
2. Select the Schema Registry service and go to Configuration.
3. Set the Fingerprint Version property to VERSION_2.
4. Select Actions>Regenerate Fingerprints.
5. Click Regenerate Fingerprints to start the action.
6. Restart Schema Registry.

Adding configs manually

The major/minor Runtime upgrade does not install additional configs available in the newer versions of the Data Engineering cluster templates. This topic contains a list of configs added over different template versions. Use the Cloudera Manager UI to add them manually, if the config doesn't already exist.

- Clusters -> HDFS -> Configuration:

```
hdfs_verify_ec_with_topology_enabled=false
```

```
erasure_coding_default_policy=” “
```

- Clusters -> HDFS -> Configuration -> Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml:

```
fs.s3a.buffer.dir=${env.LOCAL_DIRS:-${hadoop.tmp.dir}}/s3a
HADOOP_OPTS="-Dorg.wildfly.openssl.path=/usr/lib64 ${HADOOP_OPTS}"
```

- Clusters -> Yarn -> Configuration:

```
yarn_admin_acl=yarn,hive,hdfs,mapred
```

- Clusters -> Yarn -> Configuration -> YARN Service MapReduce Advanced Configuration Snippet (Safety Valve):

```
mapreduce.fileoutputcommitter.algorithm.version=1
mapreduce.input.fileinputformat.list-status.num-threads=100
```

- Clusters -> Tez -> Configuration:

```
tez.grouping.split-waves=1.4
tez.grouping.min-size=268435456
tez.grouping.max-size=268435456
```

- Clusters -> Tez -> Configuration -> Tez Client Advanced Configuration Snippet (Safety Valve) for tez-conf/tez-site.xml:

```
tez.runtime.pipelined.sorter.lazy-allocate.memory=true
```

- Clusters -> Hive -> Configuration -> Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml:

```
fs.s3a.ssl.channel.mode=openssl
hive.txn.acid.dir.cache.duration=0
hive.server2.tez.session.lifetime=30m
hive.blobstore.supported.schemes=s3,s3a,s3n,abfs,gs
hive.orc.splits.include.fileid=false
hive.hook.proto.events.clean.freq=1h
hive.metastore.try.direct.sql.ddl=true
hive.privilege.synchronizer=false
```

- Clusters -> Hive -> Configuration:

```
hiveserver2_idle_session_timeout=14400000
```

- spark_on_yarn -> Configuration -> Spark Client Advanced Configuration Snippet (Safety Valve) for spark-conf/spark-defaults.conf:

```
spark.hadoop.mapreduce.fileoutputcommitter.algorithm.version=1
spark.hadoop.fs.s3a.ssl.channel.mode=openssl
```

- Clusters -> Hive Metastore -> Configuration -> Hive Metastore Server Advanced Configuration Snippet (Safety Valve) for hive-site.xml

```
hive.metastore.try.direct.sql.ddl=true
```

- Clusters -> <your cluster name> -> Configuration -> Suppressed Health and Configuration Issues:

```
role_config_suppression_namenode_java_heapsize_minimum_validator=true
```

Upgrading Cloudera Operational Database clusters

Cloudera Operational Database clusters cannot be upgraded through the Data Hub user interface. Instead, use the Cloudera Beta CLI to run the upgrade-database command.

For more information on CDP CLI beta commands, see the *CDP CLI Beta command reference*. For more information on rolling and non-rolling Cloudera Runtime upgrades for Cloudera Operational Database, see *Performing a Cloudera Runtime upgrade* in the Cloudera Operational Database documentation. For more information about rolling and non-rolling OS system upgrades for Cloudera Operational Database, see *Performing a Cloudera operating system upgrade* in the Cloudera Operational Database documentation.

Before you begin:

- Download and install the latest CDP Beta CLI.
- Required role: You must be logged into the Cloudera Operational Database as an ODAdmin.
- To use Cloudera Operational Database on a GCP environment, you must do it through CDP CLI with --use-hdfs flag.
- Understand Cloudera environment and user management. For more information, see the Cloudera Operational Database documentation topics *User management in Cloudera Operational Database* and *Cloudera Environments* topics.
- In the Cloudera Manager properties, increase the omid_max_heap_size property for the Omid service to at least 3GB before starting the upgrade from 7.2.9/7.2.10 to 7.2.11:

Important: Before upgrading the Cloudera Runtime version in your existing Cloudera Operational Database clusters to 7.2.16 or higher versions, you might need to perform some additional steps before upgrading. For more information, see *Data Lake upgrade validations for Python dependency* under the Cloudera Management Console what's new topic and *Upgrading Data Hubs*.

This Beta CDP CLI command upgrades an operational database in an environment to a given Runtime:

```
cdp opdb upgrade-database --environment <ENVIRONMENT-NAME> --
database <DATABASE-NAME> --runtime <RUNTIME-VERSION> [--os-upgrade-only | --
no-os-upgrade only]
```

Option	Description
--environment (string)	The name or CRN of the environment.
--database <value>	The name or CRN of the database.
--runtime <value>	The runtime version to upgrade to.
[--os-upgrade-only | --no-os-upgrade-only]	Controls whether to perform only an Operating System upgrade.

Related Information

CDP CLI Beta command reference

Rolling upgrades in Cloudera Operational Database

User management in Cloudera Operational Database

Cloudera environments

Performing a Cloudera Data Hub service pack upgrade

A service pack upgrade can be conducted on a single Cloudera Data Hub cluster in an environment, or on multiple Data Hub clusters.

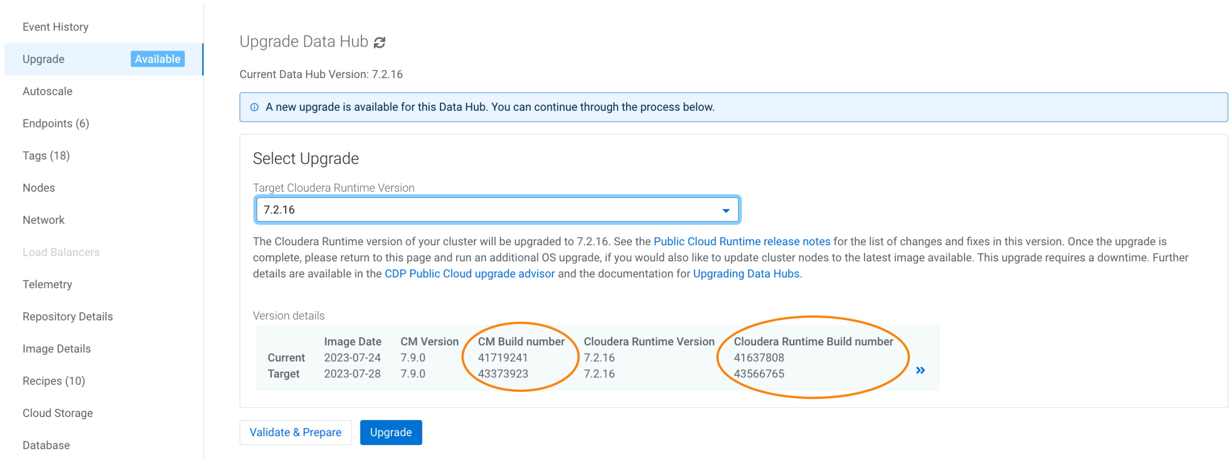
About this task

Complete the steps for each Cloudera Data Hub cluster that you are upgrading.

Required role: DatahubAdmin or Owner over the Cloudera Data Hub cluster

Procedure

1. In the left-hand menu, click Data Hub Clusters and then select the Cloudera Data Hub cluster to upgrade.
2. Scroll to the bottom of the Data Hub details page and select the Upgrade tab.
3. From the drop-down menu, select the Target Runtime Version.



4. 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.
5. Click Upgrade.
6. Monitor the upgrade progress using the Cloudera Data Hub Event History tab.
7. If the upgrade fails, check the *Troubleshooting upgrade* section and re-try the upgrade.
8. If the upgrade is successful, you can re-enable autoscaling on the cluster if desired.
9. If your cluster contains the Oozie service, after the service pack upgrade is complete, re-install the Oozie shared libraries and YARN MapReduce Framework JARs.
   a) Navigate to the cluster's Oozie service in Cloudera Manager and click the Status tab.
   b) Click ActionsInstall Oozie ShareLib.
   c) Navigate to the cluster's YARN service in Cloudera Manager and click the Status tab.
   d) Click ActionsInstall YARN MapReduce Framework JARs.

Related Information

Troubleshooting upgrade

Performing a Cloudera Data Hub OS upgrade

Perform a Cloudera Data Hub OS upgrade to update the OS and VM packages to those available in the latest pre-warmed image.

About this task

Required role: DatahubAdmin or Owner over the Cloudera Data Hub cluster

Before you begin

It your cluster uses Streams Replication Manager (SRM), it is likely that you are storing security files, like truststores, in /opt/cloudera/security/ on your cluster hosts. These files are lost during the upgrade. As a result, you must back them up before the upgrade and restore them following the upgrade.

Before you begin

Before performing any OS upgrade, make sure there is no data belonging to NiFi or NiFi Registry on the root disk of the VM (this is the case for any version before Cloudera Runtime 7.2.10).

If you have NiFi or NiFi Registry data on the root disk, run the commands below to move the data to the right location prior to performing the upgrade. Before executing these scripts, stop the NiFi and NiFi Registry services.

On the NiFi nodes:

When upgrading from Cloudera Runtime 7.2.9 or lower versions

```
hadoopDirectory=/hadoopfs/$(ls /hadoopfs/ | sort | tail -n 1)
mkdir $hadoopDirectory/working-dir
cp -R /var/lib/nifi/* $hadoopDirectory/working-dir
echo "nifi.working.directory should be set to $hadoopDirectory/working-dir"l
atest_nifi_conf_directory=$(find /run/cloudera-scm-agent/process -name nifi\
.properties | sort | tail -n 1)
latest_nifi_conf_directory=${latest_nifi_conf_directory%nifi.properties}
echo "Latest nifi conf directory used to copy files before migration: $late
st_nifi_conf_directory"mkdir -p $hadoopDirectory/working-dir/config_backup
chmod 755 $hadoopDirectory/working-dir/config_backup
cp $latest_nifi_conf_directory/nifi.properties $hadoopDirectory/working-dir/
config_backup
cp $latest_nifi_conf_directory/bootstrap.conf $hadoopDirectory/working-dir/
config_backup
chown nifi:nifi -R $hadoopDirectory/working-dir
```

After the bash script executes, update the nifi.working.directory configuration value with what the script returns.

To set nifi.working.directory, perform the following steps:

1. Open the Cloudera Manager UI.
2. Go to the NiFi service.
3. Select the Configuration tab.
4. Search for nifi.working.directory.
5. Set the new value and click Save.

Note: When upgrading from Cloudera Runtime 7.2.10 or higher, no manual steps are required.

On the management node (where NiFi Registry is):

```
hadoopDirectory=/hadoopfs/$(ls /hadoopfs/ | sort | tail -n 1)
mkdir $hadoopDirectory/working-dir
cp -R /var/lib/nifiregistry/* $hadoopDirectory/working-dir
chown nifiregistry:nifiregistry -R $hadoopDirectory/working-dir
echo "nifi.registry.working.directory should be set to
$hadoopDirectory/working-dir"
```

After the bash script executes, update the nifi.registry.working.directory configuration value with what the script returns.

To set nifi.registry.working.directory, perform the following steps:

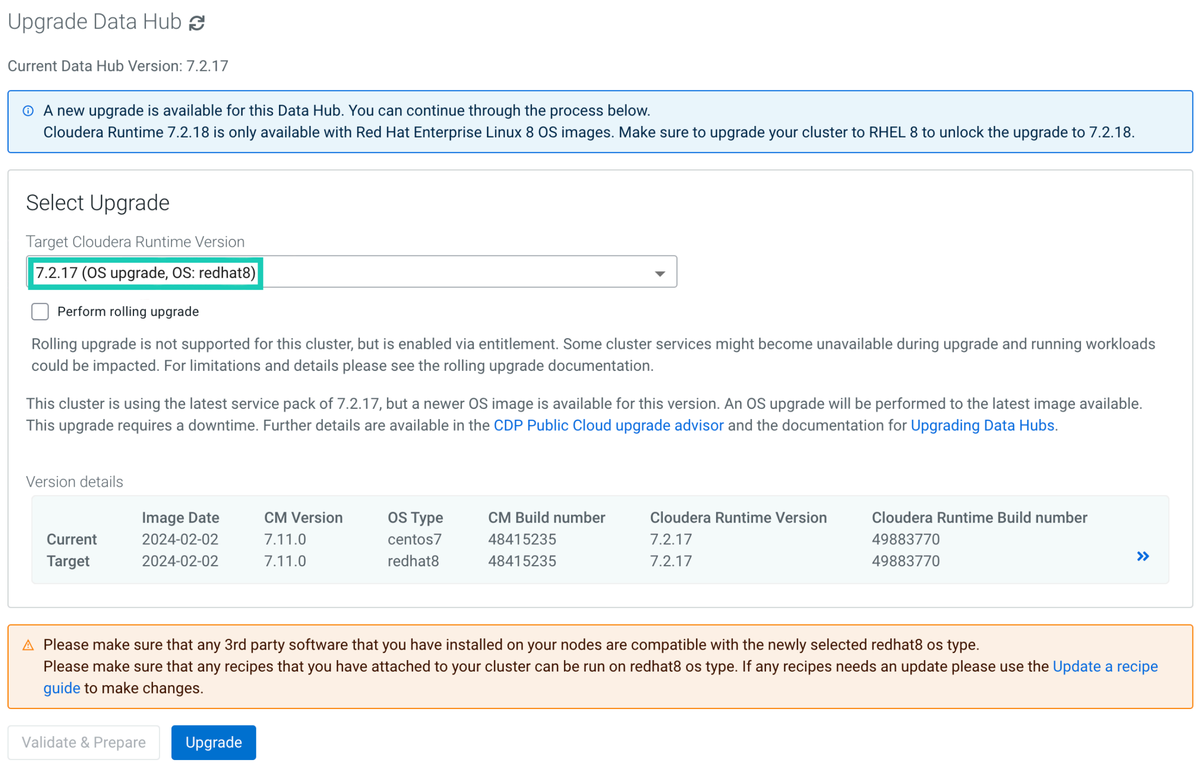
1. Open the Cloudera Manager UI.
2. Go to the NiFi service.
3. Select the Configuration tab.
4. Search for nifi.registry.working.directory.
5. Set the new value and click Save.

After the new configuration values are set on the NiFi and the management nodes, start the services.

Important: During an OS upgrade, any data on the root volume (parcels, service logs, custom software) will be lost.

Procedure

1. In the left-hand menu of Cloudera, click Data Hub Clusters and then select the Cloudera Data Hub cluster to upgrade.
2. Scroll to the bottom of the Data Hub details page and select the **Upgrade** tab.
3. From the drop-down menu, select the Target Runtime Version. An OS upgrade is indicated by the text “OS Upgr ade (OS: <TARGET-OS>)” (where <TARGET-OS> is centos7 or redhat 8) next to the target Cloudera Runtime version.



4. Click Upgrade.
5. Monitor the upgrade progress using the Cloudera Data Hub **Event History** tab.

What to do next

If you upgraded from Cloudera Runtime 7.2.9 or a lower version, perform the following tasks on the NiFi UI as post-upgrade steps when NiFi is available and running.

1. Open the NiFi UI.
2. Select Controller Settings from the Global Menu available in the top-right corner of the NiFi UI.

3. Go to the Management Controller Services tab of the **NiFi Settings** dialog.
4. Edit the Default Reporting Task SSL Context Service.

    a. Disable the running controller service by using the button.
    b. Modify the Keystore Filename value to point to the new nifi working directory. Use /hadoopfs/fs4/working-dir/cm-auto-host_keystore.jks instead of /var/lib/nifi/cm-auto-host_keystore.jks.
    c. Click APPLY.
5. Enable the Default Reporting Task SSL Context Service using the button.
6. Go to the Reporting Tasks tab.
7. Edit the Default Atlas Reporting Task by clicking the button.

    a. Modify the Atlas Configuration Directory to point to the new nifi working directory. Use /hadoopfs/fs4/working-dir/ instead of /var/lib/nifi/.
    b. Modify the Kerberos Keytab to point to the new nifi working directory. Use /hadoopfs/fs4/working-dir/nifi.keytab instead of /var/lib/nifi/nifi.keytab.
    c. Click APPLY.

What to do next

If your cluster uses SRM, restore the security files you backed up before the upgrade.

Upgrading a Cloudera Data Hub cluster with the CDP CLI

You can initiate a Cloudera Data Hub cluster upgrade (either OS, Runtime, or both) 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.

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.

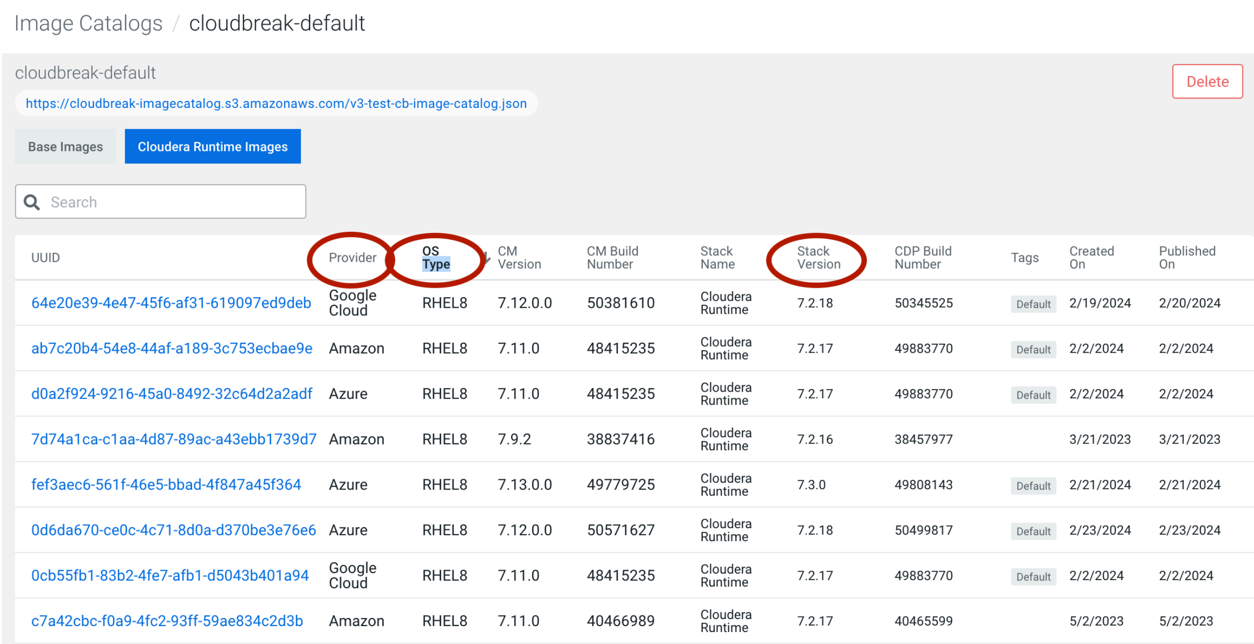


Image Catalogs / cloudbreak-default

cloudbreak-default

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

Delete

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

The cdp datahub upgrade-cluster command includes the following options:

```
cdp datahub upgrade-cluster

          --cluster-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]
          [--cli-input-json <value>]
          [--generate-cli-skeleton]
```

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

Important: The --runtime option does not upgrade the OS. Upgrading the OS is a separate process that requires specifying the --lock-components option.

Table 2:

Option	Description
--cluster-name (string)	Required. The name or CRN of the Cloudera Data Hub cluster to upgrade.
--image-id (string)	The ID of an image to upgrade to. If upgrading from CentOS to RHEL, ensure to provide a RHEL 8 image ID.
–runtime (string)	The Runtime version to upgrade to. When you specify the Cloudera Runtime version, the upgrade uses the latest image ID of the given Cloudera Runtime version from the same image catalog used for Cloudera Data Hub cluster creation.
--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-latest-available-image-per-runtime | --no-show-latest-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.

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

1. Specify one of either --image-id, --runtime, or --lockComponents, which makes an explicit choice of the exact image, Cloudera Runtime (latest OS), or latest OS (same Runtime) for upgrade.
2. Specify both --image-id and --lockComponents,which specifies an image and ensures the image represents an OS only upgrade.
3. Specify none of the --image-id, --runtime, or --lockComponents parameters, which initiates a Cloudera Runtime/Cloudera Manager 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-runtime
```

```
--dry-run
```

Cloudera Data Hub rolling upgrades

The Cloudera Data Hub rolling upgrade allows you to upgrade the cluster without stopping the Cloudera Data Hub cluster itself and incurring workload downtime.

Cloudera Data Hub rolling upgrades are only available for the following cluster types:

- Cloudera Operational Database
- Cloudera Streams Messaging

Cloudera Operational Database clusters

For a Cloudera Operational Database cluster to be eligible for rolling upgrade, the following requirements must be met:

- Currently, rolling OS upgrade is only supported on Cloudera Operational Database clusters with storage selected as HDFS or cloud without ephemeral storage.
- Rolling OS upgrade is not supported on Micro Cloudera Operational Database clusters.

Table 3:

Current Cloudera Runtime version & template	Rolling upgrade support for Cloudera Runtime?	Rolling OS upgrade support?
7.2.17.x	Yes, to 7.2.18+	Yes
7.2.18	Yes, to 7.2.18+ (including service pack upgrades to 7.2.18.100 and higher versions)	Yes
7.2.17.300+ 7.2.18+	Yes, to 7.3.1 [1]	Yes
7.3.1.0 **Note:** Rolling upgrade for Cloudera Operational Database clusters from 7.3.1.0 to 7.3.1 higher versions is in Technical Preview and only available for clusters with Ephemeral cache.	Yes, to latest 7.3.1 Service Pack (SP) or Cumulative Hotfix (CHF) version	Yes

For instructions on performing Cloudera Operational Database rolling upgrades, see Rolling upgrade in Cloudera Operational Database.

Cloudera Streams Messaging clusters

For a Streams Messaging cluster to be eligible for rolling upgrade, the following requirements must be met:

- The cluster OS must be RHEL 8. Streams Messaging clusters on CentOS are not supported.
- The cluster must be Runtime version 7.2.18 or higher version.
- The cluster template must be Streams Messaging High Availability.

[1] When upgrading to Cloudera Runtime 7.3.1 using the Cloudera Management Console, you are upgraded to the latest available Service Pack (SP) or Cumulative Hotfix (CHF) after selecting 7.3.1 as the target Cloudera Runtime version.

You can specify the exact Cloudera Runtime version, such as 7.3.1.0 or 7.3.1.10x, using CDP CLI with image ID. In this case, ensure that there is a supported upgrade path for your source Cloudera Runtime version. For more information, see the Supported Cloudera Runtime 7.3.1 upgrade paths.

Current Cloudera Runtime version & template	Rolling upgrade support for Cloudera Runtime?	Rolling OS upgrade support?
7.2.18 Streams Messaging High Availability template (Clusters **created** on 7.2.18 only)	Yes, to 7.2.18+ (including service pack upgrades to 7.2.18.100 and higher versions) and 7.3.1	Yes, for up to 20 nodes. **Important:** Rolling OS upgrades replace nodes one-by-one and can take a significant amount of time, several hours or more, depending on the number of nodes.
7.2.18+	Yes, to 7.3.1 [1]	Yes
7.3.1.0	Yes, to latest 7.3.1 Service Pack (SP) or Cumulative Hotfix (CHF) version	Yes

In some circumstances, a rolling upgrade may not be supported for a Streams Messaging cluster, but can be enabled through entitlement. The Cloudera Data Hub upgrade UI displays information about whether a rolling upgrade is available, unavailable, or may be available under entitlement. For instructions on performing a Cloudera Data Hub cluster upgrade, including rolling upgrades, see Upgrading Cloudera Data Hub clusters. For information about obtaining an entitlement for rolling upgrade, contact Cloudera Customer Support.

Cloudera Data Hub rolling upgrade limitations and issues

Cloudera Data Hub rolling upgrades have the following limitations:

Cloudera Operational Database clusters

- HBase commands may fail during the rolling upgrade with the error "ServerNotRunningYetException: Server is not running yet." HBase retries DDL operations submitted while the master is initializing until the master is fully initialized to serve the request. However, a situation might arise where the default number of retries or intervals proves to be insufficient for an operation submitted by the client to complete.

  Implementing the following configuration adjustments in your client application can support the master getting initialized up to 10 minutes:

  ```
  <property>
      <name>hbase.client.pause</name>
      <value>300</value>
   </property>
   <property>
      <name>hbase.client.retries.number</name>
      <value>20</value>
   </property>
  ```

  If you have seen a longer or shorter master initialization period, you can modify these values accordingly. These retry settings apply to all types of calls to HBase service, encompassing GET, SCAN, MUTATE, and DDLs.
- In a rolling restart, if a Cloudera Operational Database cluster has less than 10 datanodes, existing writes can fail with an error indicating a new block cannot be allocated and all nodes are excluded. This is because the client has attempted to use all the datanodes in the cluster, and failed to write to each of them as they were restarted. This will only happen on small clusters of less than 10 datanodes, as larger clusters have more spare nodes to allow the write to continue.
- When performing a maintenance upgrade or other cluster upgrade, in some occasions there can been an error when the upgrade is nearly complete, but trying to re-start services/roles. The error is similar to: "Failed to start role hue-HUE_SERVER-8cc9321b2213cc5c6846c64e1fc6b1cb of service hue in cluster cod--xoaitnb0wnl1. This role requires the following additional parcels to be activated before it can start: [cdh]."

  This is due to an agent operation that sometimes is delayed and can interfere with the role start. When this happens, resume the failed upgrade from Cloudera Manager as a 'Full Admin' user.
- During the VM replacement as part of OS upgrade, every new node gets a new IP Address, and if the old IP address is cached somewhere, HDFS requests fail with UnknownHostException. It recovers after some time (10 mins max).

- If Knox is HA and one of the Knox servers is down, then accessing the service through a Cloudera Control Plane endpoint URL (i.e., through cloud load balancer) will take approximately 30 seconds to failover the request to the available Knox instance. This also means that the services that are reached through Knox will not be available behind Knox during this period time.
- During OS upgrades, attempts to access Knox on the host being upgraded may produce occasional 403 HTTP responses. Wait and retry the failed requests.
- When upgrading Cloudera Data Hub clusters to Cloudera Runtime 7.2.18.100, you might encounter staleness in knox.jwt.client.gateway.address configuration in case its value points to the address of the Data Lake node. If staleness occurs after the upgrade, you need to run Deploy Client Configuration in Cloudera Manager.

For more limitations of Cloudera Operational Database, see Rolling upgrade in Cloudera Operational Database.

Cloudera Streams Messaging clusters

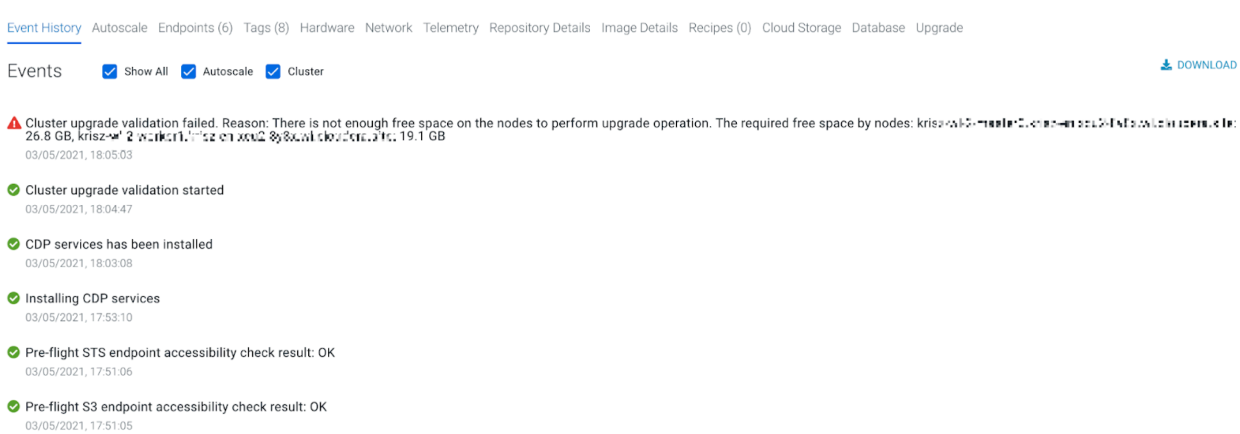
- Rolling upgrades are not supported for Cruise Control or Streams Messaging Manager (SMM). When upgrading a Streams Messaging cluster, expect that both of these services will be temporarily unavailable during the upgrade. This, however, does not impact Kafka's ability to perform a rolling upgrade.
- Rolling upgrades for Schema Registry are only supported from Cloudera Runtime 7.2.18 or higher to higher versions. When upgrading a Streams Messaging cluster from a lower version, expect that clients connecting to the Schema Registry service might experience downtime. This, however, does not impact Kafka's ability to perform a rolling upgrade.
- If Knox is HA and one of the Knox servers is down, then accessing the service through a Cloudera Control Plane endpoint URL (i.e., through cloud load balancer) will take approximately 30 seconds to failover the request to the available Knox instance. This also means that the services that are reached through Knox will not be available behind Knox during this period time.
- During OS upgrades, attempts to access Knox on the host being upgraded may produce occasional 403 HTTP responses. Wait and retry the failed requests.
- When upgrading Cloudera Data Hub clusters to Cloudera Runtime 7.2.18.100, you might encounter staleness in knox.jwt.client.gateway.address configuration in case its value points to the address of the Data Lake node. If staleness occurs after the upgrade, you need to run Deploy Client Configuration in Cloudera Manager.

Troubleshooting upgrade

Use the information in this section to troubleshoot problems with Cloudera Data Hub cluster upgrades.

Root disk space

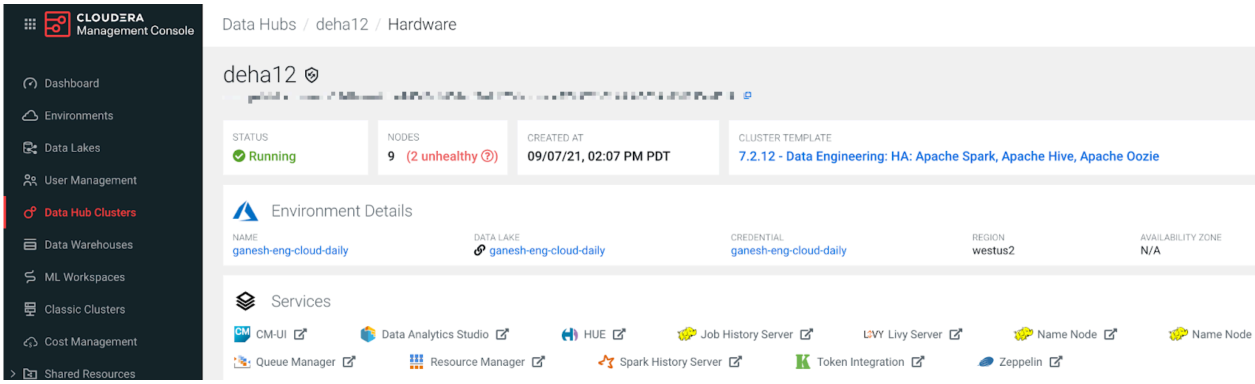
Service pack upgrades require a certain amount of root disk space. Cloudera Runtime upgrades require downloading additional parcels, and hence more storage. When you begin an upgrade, Cloudera checks the available root disk space and issues a warning if there is insufficient root disk space for the upgrade:



If your instances do not have the free space required for an upgrade (Cloudera Manager: 27 GB, other instances: 20 GB), run the scripts below to increase the root volume of the cluster nodes.

- Script to increase the root disk volume for AWS
- Script to increase the root disk volume for Azure

Oozie Shared Library mismatch



If the upgraded cluster contains the Oozie service, it may appear as being in bad health after the upgrade due to an known issue with the Oozie server shared library. On the Oozie Shared Library Check page in Cloudera Manager, you will see an error similar to: “The Oozie Server build version and the Oozie Server shared library version do not match.”

To workaround this issue, follow the steps at the end of the Performing a service pack upgrade section to re-install the Oozie shared libraries and YARN MapReduce Framework JARs.

Upgrade fails due to active Cloudera Manager commands

Upgrade may fail if there are active Cloudera Manager commands running when an upgrade is triggered. If you receive the error message “There are active commands running on Cloudera Manager, upgrade is not possible. Active commands: ApiCommand[..., name: <cm command name>,]”, then kill the active commands and retry the upgrade.

Cloudera Manager, Cloudera Runtime, or other components are out-of-sync with Cloudera

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

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

Run the command as follows: `cdp datahub sync-component-versions-from-cm --datahub-name <datahub name or CRN>`

Related Information

Performing a Cloudera Data Hub service pack upgrade

Autoscaling clusters

Autoscaling is a feature that adjusts the capacity of cluster nodes running YARN by automatically increasing or decreasing, or suspending and resuming, the nodes in a host group. You can enable autoscaling based either on a schedule that you define, or the real-time demands of your workloads.

Load-based autoscaling

Attention:

Load-based autoscaling is available for clusters with the following Cloudera Runtime versions:

- 7.2.15 and higher

Load-based auto-scaling is supported in AWS and Azure. In GCP, load-based autoscaling is available as technical preview and is under entitlement. To obtain the required entitlement, contact your Cloudera Account Representative.

Load-based autoscaling suspends and resumes (stops and starts) instances on the cloud provider to increase or decrease capacity for nodes running NodeManagers (for example, the compute host group), based upon YARN's assessment of pending demand and available capacity. Load-based autoscaling can help control costs while providing quick, on-demand cluster capacity when you need it (within a few minutes).

When you configure a load-based autoscaling policy, you choose a minimum and maximum number of nodes for the host group. The maximum number of nodes determines how many instances are provisioned, but these instances are suspended and resumed as the workload demand requires. The policy will not provision instances beyond the maximum range of nodes that you define, regardless of the demand on the cluster. You also define a cooldown period, which is the amount of time in minutes to wait before another autoscaling operation is performed.

Schedule-based autoscaling

Attention:

Schedule-based autoscaling is available for clusters with the following Cloudera Runtime versions:

- 7.2.15 and higher

Schedule-based autoscaling scales the nodes in a host group up or down based upon a schedule that you define. Schedule-based autoscaling is useful if workload demands tend to be high or low on a fairly regular, consistent basis.

When you configure a schedule-based autoscaling policy, you define a target node count, which is the number of nodes that you want to scale up or down to at a particular time. You also select whether or not to repeat the schedule, and finalize the schedule by entering a CRON expression and selecting the desired timezone. When the particular time/date that you define in the CRON expression occurs, the cluster is upscaled or downscaled to your target node count. The time taken to add nodes to a cluster varies by cloud provider and the configuration of the nodes (for example, recipes can have an impact on the time it takes to add a node).

Note: Typically, schedules (especially those configured to scale-up) should be defined in a manner to factor in the time it takes to add new nodes. For example, workloads starting at 9pm will typically set a scale-up schedule for 8:45 pm, assuming it takes around 15 minutes to add a node.

General considerations

Currently, YARN NodeManager is the only service compatible with autoscaling. Because of this, autoscaling is only available by default in the Data Engineering and Data Engineering HA cluster definitions. Autoscaling can also be configured for clusters with custom templates that include YARN NodeManager and the necessary gateway components.

For autoscaling Cloudera Operational Database clusters, refer to the Autoscaling in public cloud environments and Fast autoscaling in Cloudera Operational Database documentations.

Known Limitations

- Load-based autoscaling does not work with YARN dynamic queues.

Autoscaling behavior

Before you define an autoscaling policy, note the following autoscaling behaviors.

General behaviors

- Clusters can perform one upscale or downscale operation at a time.
- A cluster will continue to accept jobs while it is running, regardless of any in-progress upscale or downscale operations.
- Only one autoscale policy type (either load-based or schedule-based) can be configured for a single host group, in a single cluster, at a time.
- Autoscaling is available for host groups with nodes running YARN NodeManager only (and optionally client/GATEWAY components, but not any other service components).
- If there are not enough nodes available to match the requested scale operation, the operation will proceed on however many nodes are available (for example, during a request for a 10 node scale-up, if the cluster loses 1 node, the operation will proceed with scaling-up 9 nodes instead of 10).
- Autoscaling will be disabled if the cluster has any node failures on instances running YARN ResourceManager, or the ClouderaManager node.
- After scaling down, nodes will show up as UNHEALTHY in Cloudera Manager. This is expected. While scaling down, stopped nodes are put into maintenance mode, to suppress alerts.

Attention: Autoscaling is not available for Cloudera Data Hub clusters created with the default Streaming Analytics Light Duty cluster template.

Note: Make sure to create clusters with at least one node in each host group running YARN NodeManagers, for automatic resource configuration to take effect (unless explicitly configuring YARN node-level resources in the cluster template). Once a cluster has been created, the node count for the same can go to 0 (or the minimum can be set to 0 via autoscaling configuration).

Warning: Before performing the FreeIPA upgrade for Cloudera Data Hub clusters where autoscaling is enabled, you must disable autoscaling and start all compute nodes to ensure the healthy state of Cloudera Data Hub clusters. After upgrading FreeIPA, autoscaling can be enabled again. For more information, see the Autoscaling must be stopped before performing FreeIPA upgrade description in the Known issue.

Cluster management operations with load-based autoscaling enabled

Specific operations are affected as follows:

- Node repair: If the node on which Cloudera Manager is running is selected for repair, then all the stopped nodes need to be selected for repair.
- Cluster stop/start: Cluster stop is supported. During a cluster start, all nodes in the host group will be started and recommissioned.
- Node delete: Node delete (non-forced) is supported, including for STOPPED instances. Forced node delete may require you to End Maintenance (Enable Alerts/Recommission) in Cloudera Manager after performing another upscale operation.
- Retry: Do not attempt retry on an autoscale operation. Instead, see *Manually recovering from load-based scaling failures*. The nature of load-based autoscaling is to automatically try again after some time, based on the load on the cluster. There are a few scenarios where this retry will be adequate to continue cluster operations.

Only a single management operation (including scaling) can be performed on a cluster at a time. For example, if an upgrade is in progress, scaling cannot be performed until the upgrade completes.

EBS versus ephemeral storage

Caution: When you configure a load-based policy, you may receive a warning about using attached EBS volumes over a certain size. While instances are in a STOPPED state (scaled-down), EBS volumes continue to be charged. In order to reduce costs, Cloudera recommends using instances with ephemeral storage (for example, the r5d or m5d instance families) for compute nodes. Ephemeral volumes will also result in better performance of the workloads–they are typically faster, and don’t utilize network throughput when writing data.

YARN Decommission Timeout

- Schedule-based downscaling will attempt to remove nodes to reach the target count, regardless of whether the node has running tasks, or data for a running job. You can use YARN DecommissionTimeout to control the leeway available to jobs beyond the scheduled decommission time. For example, with YARN decommission timeout set to one hour, a scheduled downscale will result in the following behavior:
  - Vacant nodes (no running tasks, no tasks previously run for running jobs) will be removed immediately.
  - Other nodes will be marked as decommissioning, and will not accept new work.
  - The remaining nodes will be terminated once any jobs complete, or at the one hour (YARN decommission timeout) mark.

For load-based autoscaling, the YARN DecommissionTimeout should be set to 30 seconds. See Configuring autoscaling for more details.

Related Information

Manually recovering from load-based scaling failures

Configuring autoscaling

Configuring autoscaling

To configure autoscaling, add a load-based or schedule-based policy to a cluster and define the policy parameters.

Before you begin

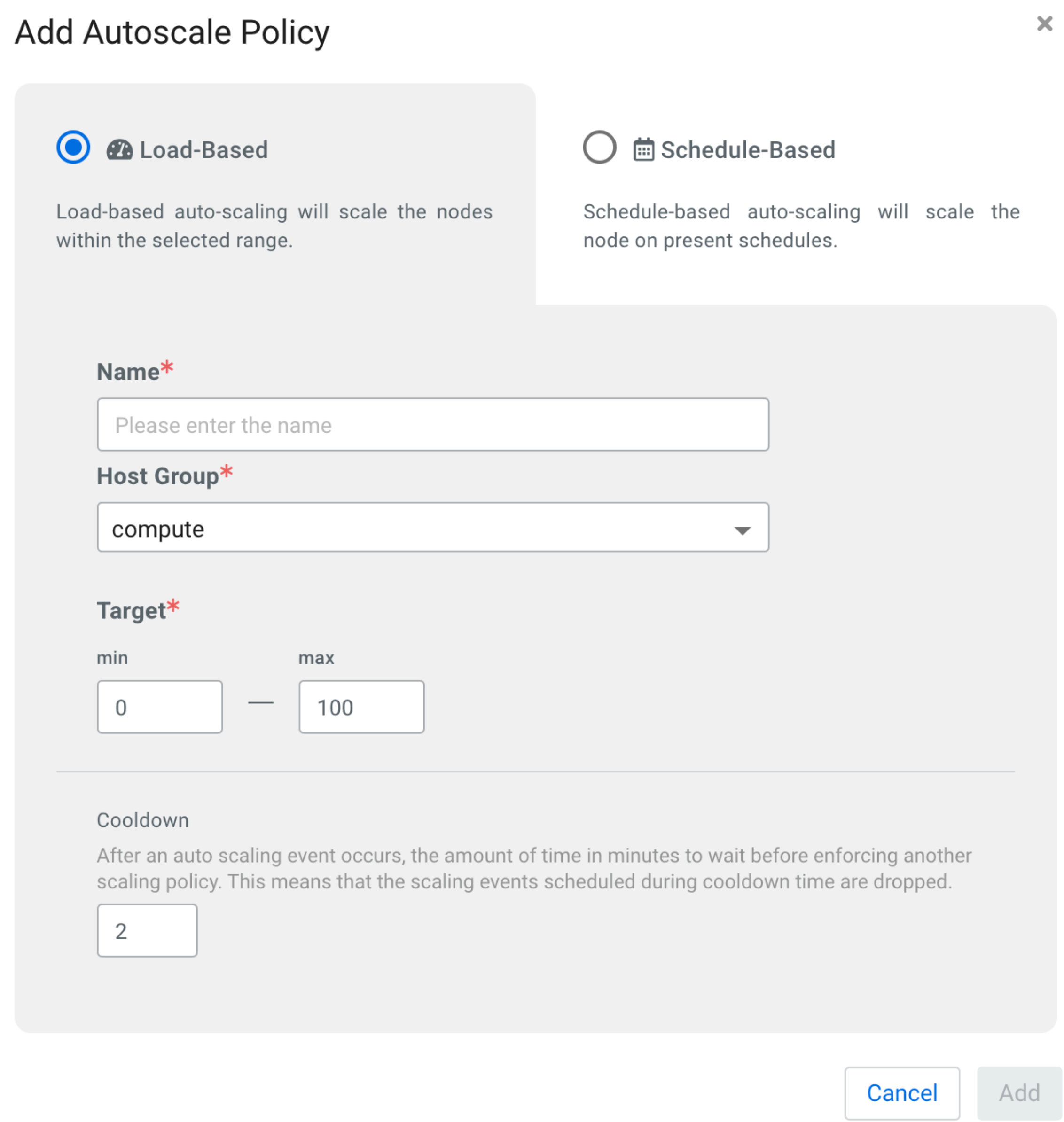
If you are configuring a load-based autoscaling policy, you must set the YARN Node Decommission Timeout property to 30 seconds. Configure the following property in Cloudera Manager: yarn_resourcemanager_nodemanager_graceful_decommission_timeout_secs

Procedure

1. From the Cloudera Management Console, click Data Hub Clusters and then select the cluster that you want to add an autoscaling policy to. You can add autoscaling policies to clusters after they have been created, not during the cluster creation process.
2. From the cluster details page, select the Autoscale tab and then click the slider button to enable autoscaling. Autoscaling is disabled by default.
3. Click Add Autoscale Policy and select Load-Based or Schedule-Based.

4. Define the policy parameters.

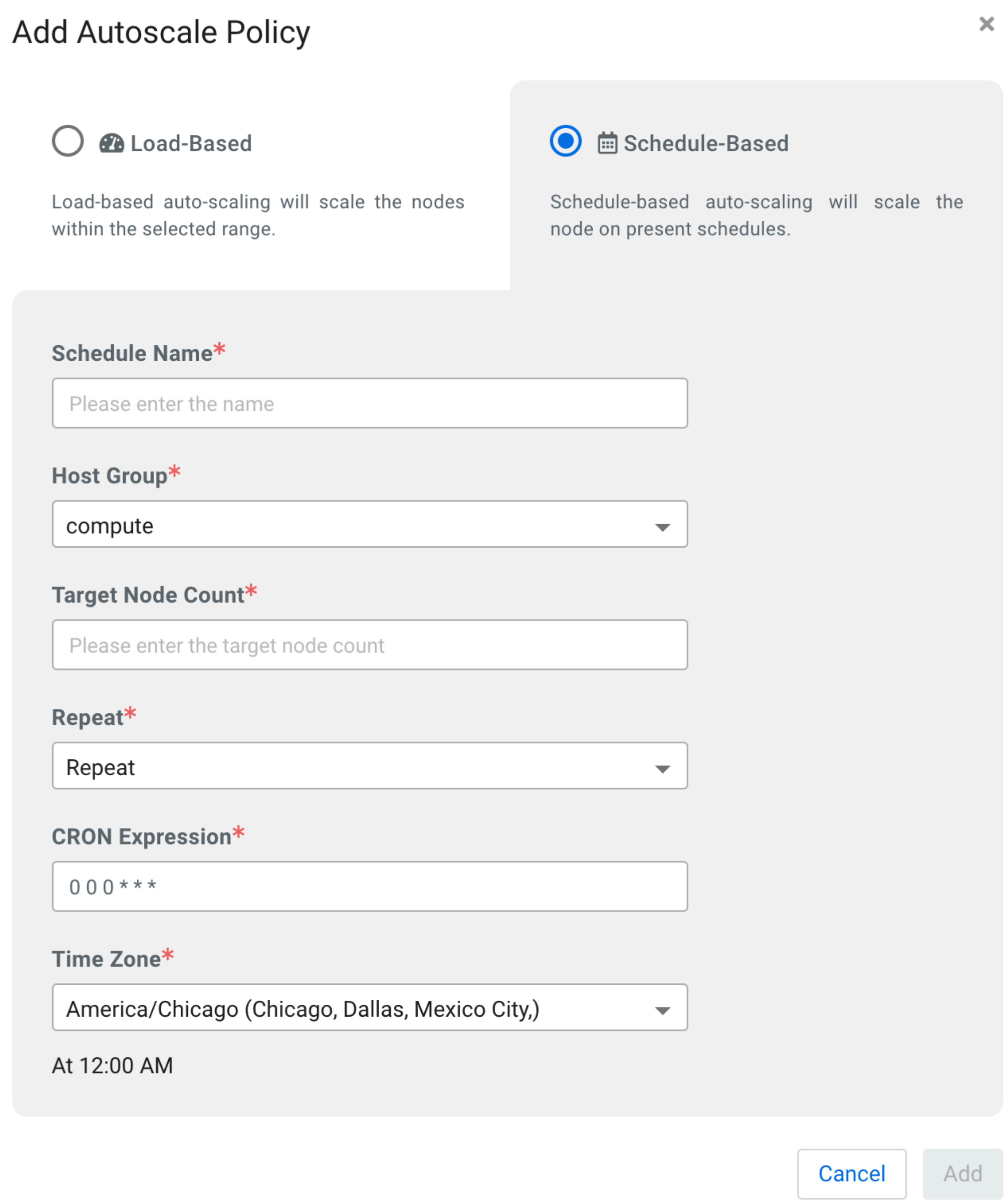
- For load-based scaling:



Parameter	Description
Name	Enter a unique name for the policy.
Hostgroup	Select the host group that you want to scale. The list of available host groups is determined by which host groups include services that can be scaled.
Target	Enter a minimum and maximum number of nodes for the policy. The maximum number of nodes determines how many instances are provisioned, but these instances are suspended and resumed as the workload demand requires. The policy will not provision instances beyond the maximum range of

Parameter	Description
	nodes that you define, regardless of the demand on the cluster.

- For schedule-based scaling:



Parameter	Description
Schedule Name	Enter a unique name for the schedule.

Parameter	Description
Hostgroup	Select the host group that you want to scale. The list of available host groups is determined by which host groups include services that can be scaled.
Target Node Count	Enter the number of nodes that you want to upscale or downscale the host group to.
Repeat	Currently, only repeating schedules are supported.
CRON Expression	Enter a CRON expression string to define the details of the schedule that you want to create.
Time Zone	Select the appropriate timezone on which to base the CRON expression.

5. Click Add. The policy appears under Auto Scaling on the **Provision Data Hub** page.

Using YARN queues with autoscaling

Beyond the regular considerations to keep in mind while configuring YARN queues, a few additional aspects need to be considered, and some additional YARN configuration parameters are required.

Using a single queue for the entire YARN cluster

Nothing specific is required when YARN is configured with a single queue only.

Using multiple queues

Workloads executing in a queue will result in a scaling operation within the bounds of the resources allocated to the specific queue.

There are manual steps that are required on the YARN side to configure multiple queues for autoscaling. The same set of steps are also required before queue configuration is changed:

- Configure Resource Manager for queue-based autoscaling. Configuring Resource Manager requires setting the yarn.resourcemanager.autoscaling.plugin-type property to the value fine in Cloudera Manager.
- User-limit-factor and other queue configuration should be tuned as usual.

Note: If all of the queues in the system are allowed to use the maximum available resources in the cluster, 'percentage' mode can be used, by setting the max-capacity for each queue to 100%. The following steps are NOT required if using percentage mode, but are required otherwise:

- Queue configuration should use \`Absolute\` values instead of percentages.
- The \`max-capacity\` for \`memory\`, and optionally \`cores\` for a cluster should correspond to the total value available to YARN when scaled up to the configured \`target maximum\`.
- The \`maximum capacity\` for a queue should be configured to be less than or equal to the \`maximum-capacity\` for the entire cluster.

Instructions for these manual steps are in the following topic, *Configuring multiple YARN queues for autoscaling*.

Configuring multiple YARN queues for autoscaling

Manual steps are required to configure multiple YARN queues for load-based autoscaling.

Before you begin

See the previous topic, *Using YARN queues with autoscaling*.

Procedure

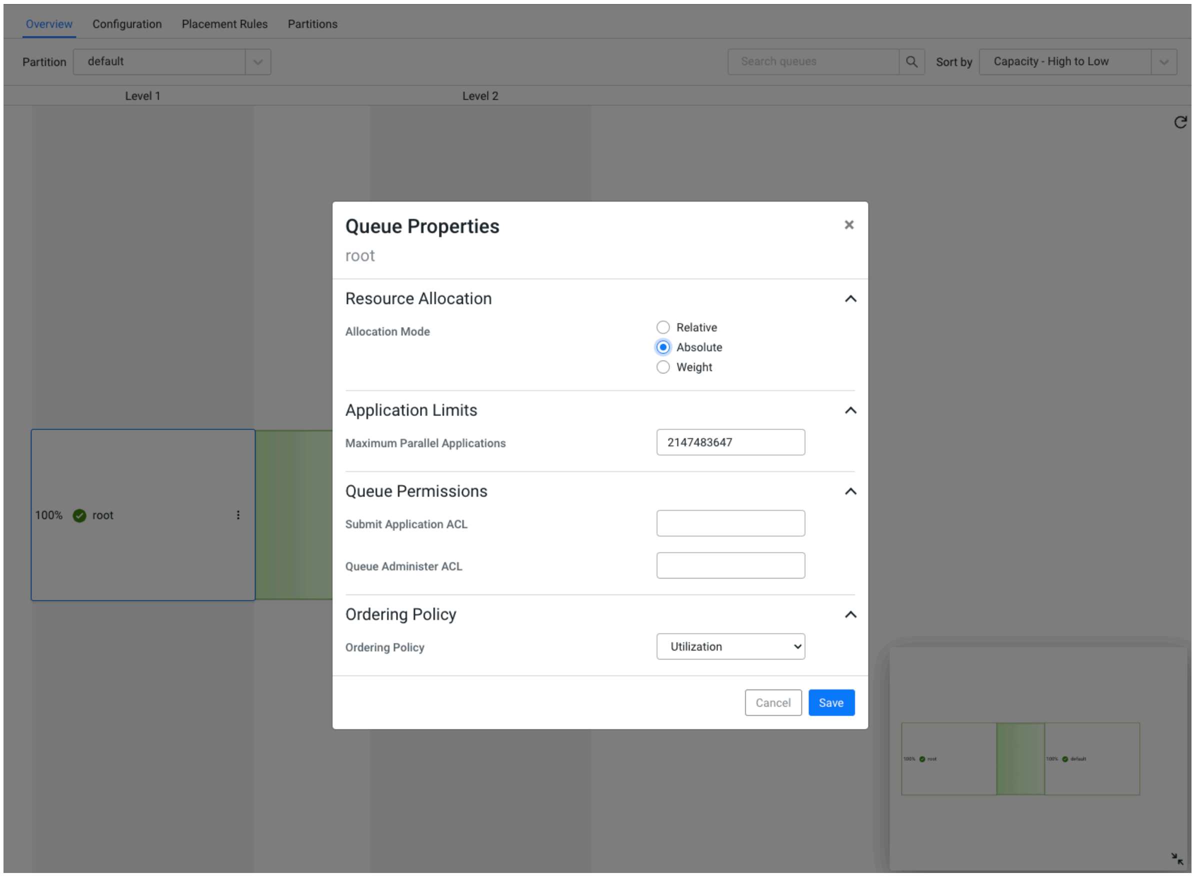
1. Set the Resource Manager plugin property to *FINE* using Cloudera Manager:
   a) In Cloudera Manager, select the YARN service.
   b) Go to the **Configuration** tab.
   c) Search for *PLUGIN-TYPE* and set the below property in the ResourceManager Advanced Configuration Snippet (Safety Valve) for yarn-site.xml field.

```
Name:  yarn.resourcemanager.autoscaling.plugin-type
Value: fine
```

   d) Save the changes.
   e) Start or restart the YARN ResourceManager service for the changes to apply.

   Important: If all of the queues in the system are allowed to use the maximum available resources in the cluster, 'percentage' mode can be used, by setting the max-capacity for each queue to 100%. If you are using percentage mode in this manner, you can skip the remainder of this task.

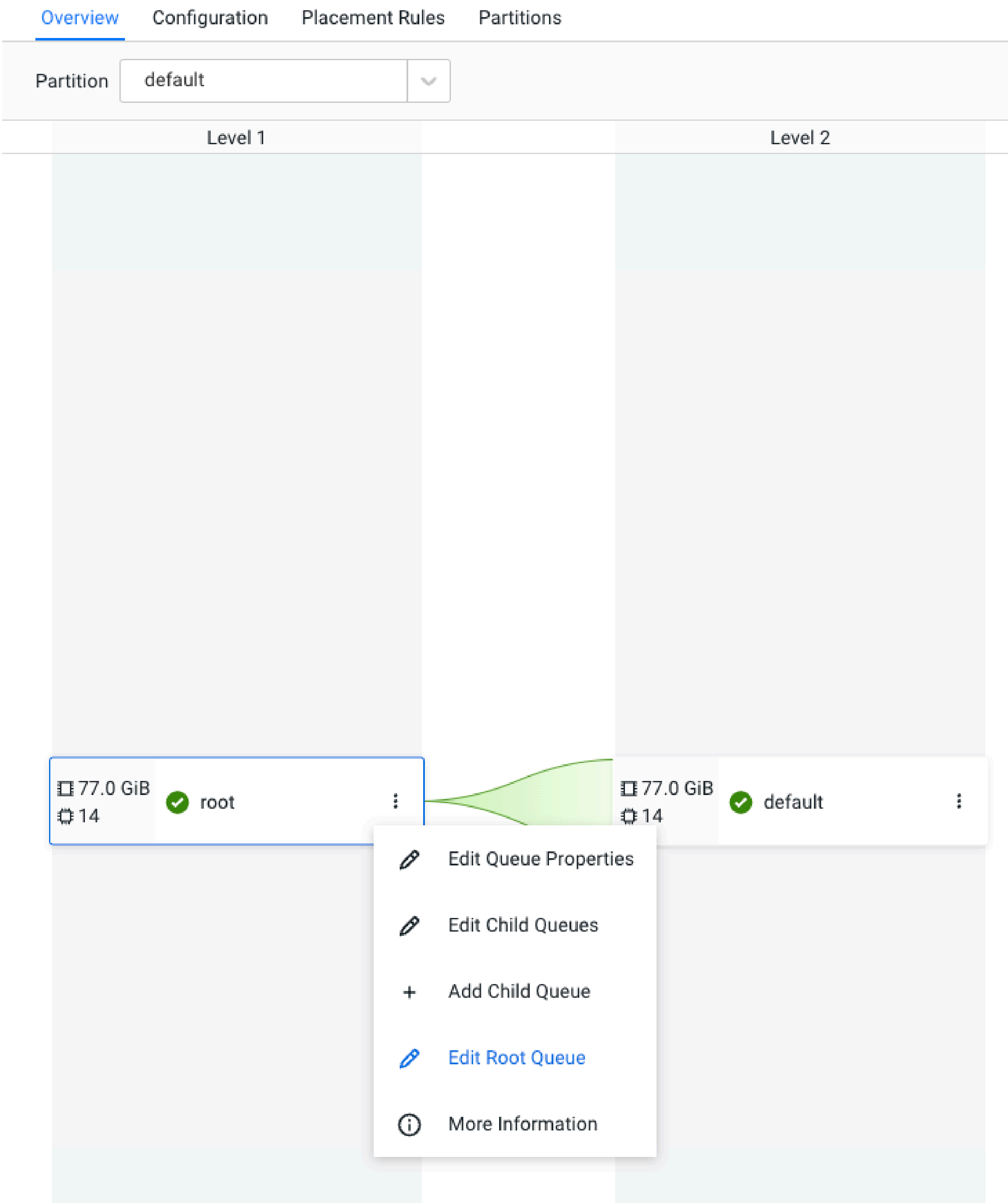
2. Ensure that the queue configuration uses absolute values instead of percentages:
   a) In the Cloudera Management Console, navigate to the specific Cloudera Data Hub cluster, and scroll to the **Services** section. Click Queue Manager.
   b) Click on the three vertical dots on the root and select Edit Queue Properties option.
   c) In the **Queue Properties** dialog box, select Absolute under Resource Allocation mode and click Save.

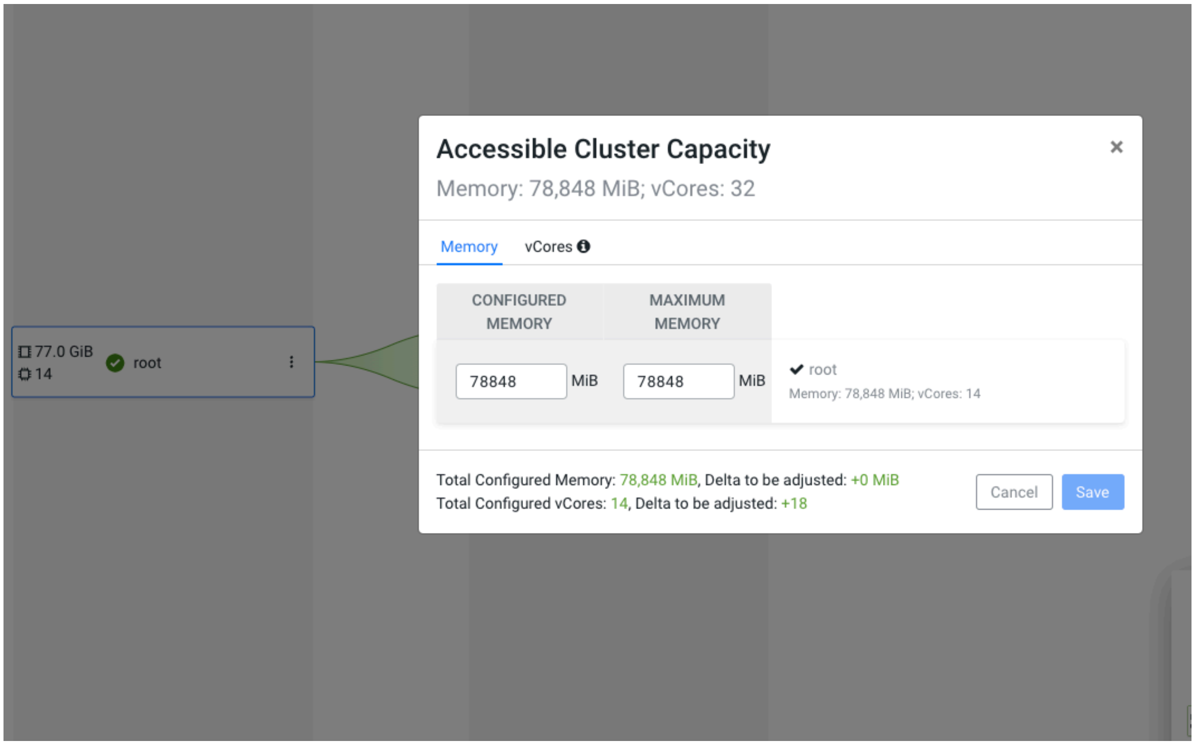


3. Disable autoscaling on the cluster. You can skip this step if you are on Cloudera Runtime version 7.2.17+.
4. Scale-up the autoscaling hostGroup manually to the new ‘Target max’ number of nodes. You can skip this step if you are on Cloudera Runtime version 7.2.17+.

5. Configure the queues:

a) In the Queue Manager UI, click on the three vertical dots on the root, select Edit Root Queue and set the 'Configured Memory' and 'Maximum Memory' to the sum total memory of the maximum number of nodes that was set in the target field of Autoscale Policy parameters. Optionally, do the same for the CPUs.



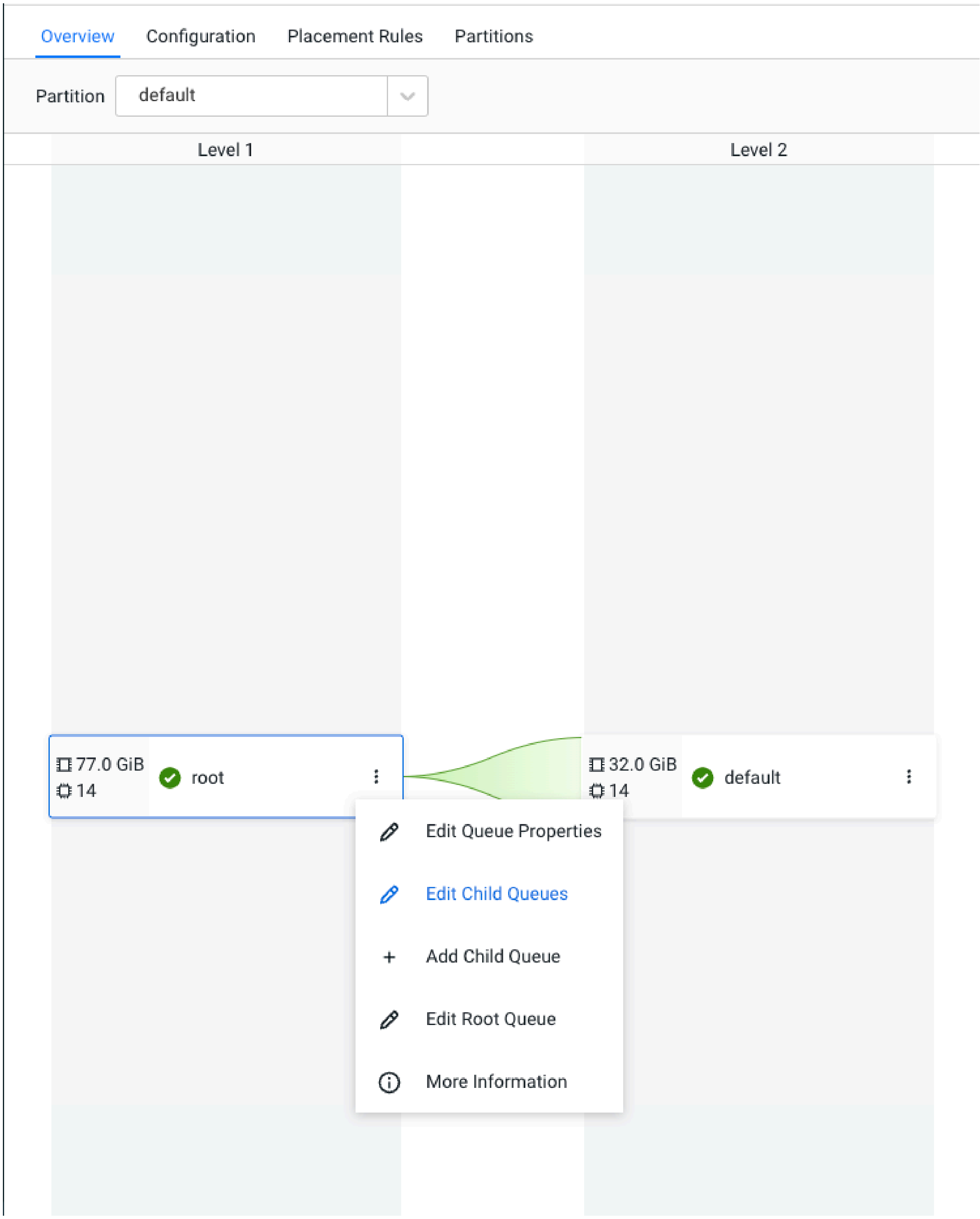


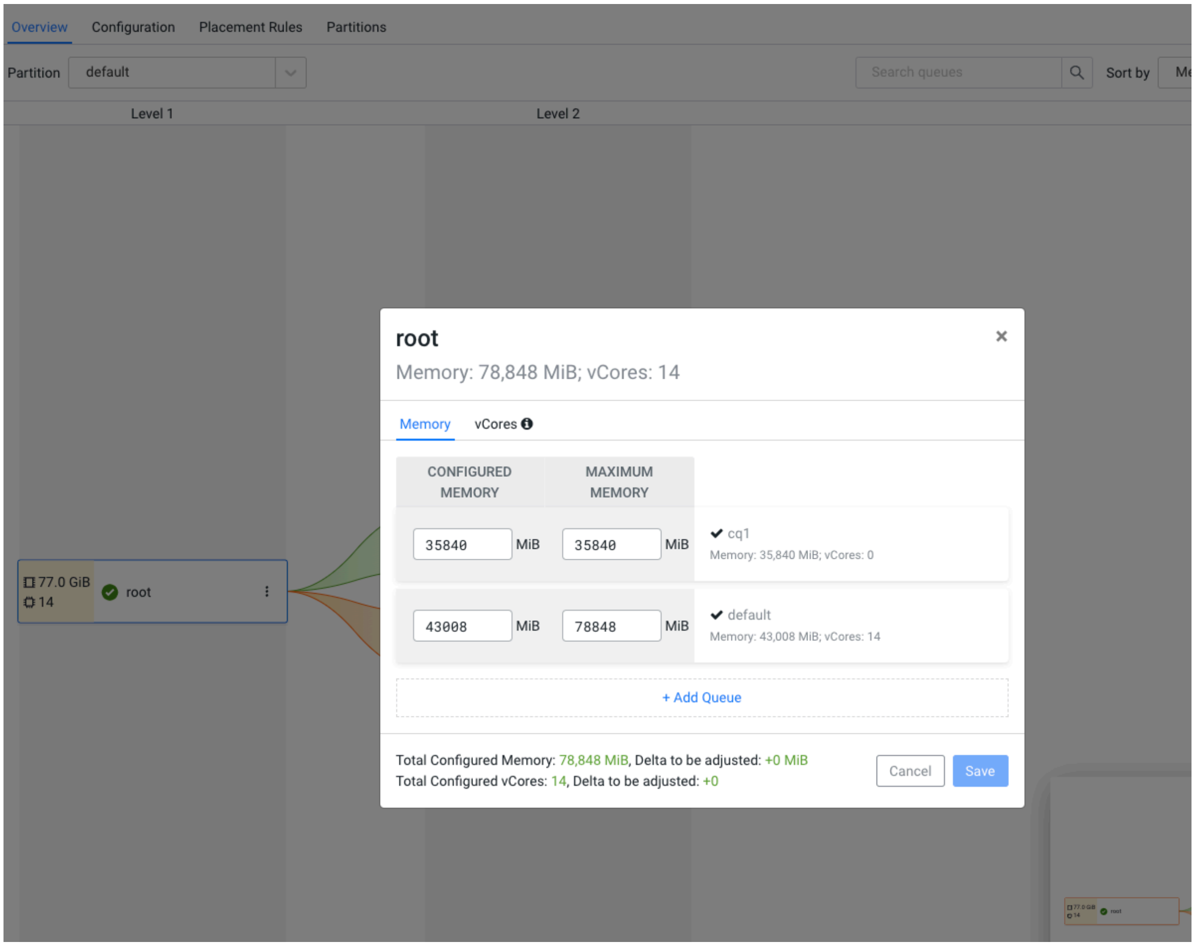
Note: With Cloudera Runtime 7.2.17+, you may exceed the Maximum Memory allowed for scenarios where the nodes will be added to the cluster manually or via autoscaling. If you exceed the Maximum Memory, select Check here to accept this change.

b) Edit the Child Queues as needed, and configure the resource allocation. In the example below, a new queue called `cq1` is added, which has a configured memory allocation, and maximum memory allocation of 35GB. The `default` queue has a configured memory allocation of 42GB, and a maximum memory allocation of 77GB.

- Jobs submitted to cq1 queue can cause the cluster to autoscale so that 35GB is available to the cq1 queue (within the cluster max of 77GB).

- Jobs submitted to the default queue can cause the cluster to autoscale so that 77GB is available to the default queue (which is the cluster max in this case).





6. Re-enable autoscaling on the cluster.

Note: In case a queue has only one type of user, for example Hive user, increase the value of the user limits within a queue to a larger number like 100.

Note: When the 'Allocation Mode' is set to 'Absolute' - if re-configuring the min or max node count for AutoScaling, make sure to update the various queue configuration values via the Queue Manager UI appropriately.

Managing autoscaling

The CDP CLI has two options for reviewing autoscaling activity.

cdp datahub list-scaling-activities

Returns all of the scaling activities (or only failed scaling activities) in the given duration, or activities that have a start time in the given time interval between start-time and end-time for a particular cluster. YARN recommendation and YARN recommendation time for all the activities are also included in the output.

```
list-scaling-activities
--cluster <value>
[--duration <value>]
[--start-time <value>]
[--end-time <value>]
[--only-failed-scaling-activities | --no-only-failed-scaling-acti
vities]
[--max-items <value>]
```

```
[--starting-token <value>]
[--page-size <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton]
```

Table 4:

Parameter	Description
--cluster	Required. The cluster Name or CRN of the cluster for which you want to list all of the scaling activities.
--duration	Value of the duration in minutes from the current point to current point minus the duration. The default value is 60 minutes.
--start-time	Time in epoch milliseconds after which you want the activities.
--end-time	Time in epoch milliseconds until which you want the activities.
--only-failed-scaling-activities | --no-only-failed-scaling-activities	Determines if only the failed scaling activities for that particular cluster are returned.
--max-items	Indicates the maximum number of scaling activities that can be present in a page. Also indicates next-token in the output, which can be used as starting-token to get the next set of scaling-activities.
--starting-token	Number of the page from which you want all the pages. Default value is 0.
--page-size	Number of entries you want to get in each API call for listing scaling activities. The default value is 100.

Sample output:

```
{
"operationId":"61ed5711-6a80-45b7-850c-b428b838145c",
"startTime":"2025-01-24T10:25:00.115000+00:00",
"yarnRecommendationTime":"2025-01-24T10:25:00.115000+00:00",
"yarnRecommendation":"Successfully polled YARN to fetch cluster
 metrics. YarnRecommendedScaleUpCount: '0', YarnRecommendedDecom
mission: [], DeterminedScaleUpCount (based on step-size and maxA
llowedUpscale): '0', DeterminedDecommissionHosts (based on step-
size and maxAllowedDownscale): []. ",
"scalingActivityReason":" Cannot trigger any scaling operation
 as there was no recommendation for scaling from yarn",
"activityStatus":"METRICS_COLLECTION_SUCCESS"
}
```

cdp datahub describe-scaling-activity

Returns the scaling activity corresponding to the cluster and operationId provided by the user.

```
describe-scaling-activity
--cluster <value>
--operation-id <value>
[--cli-input-json <value>]
[--generate-cli-skeleton]
```

Table 5:

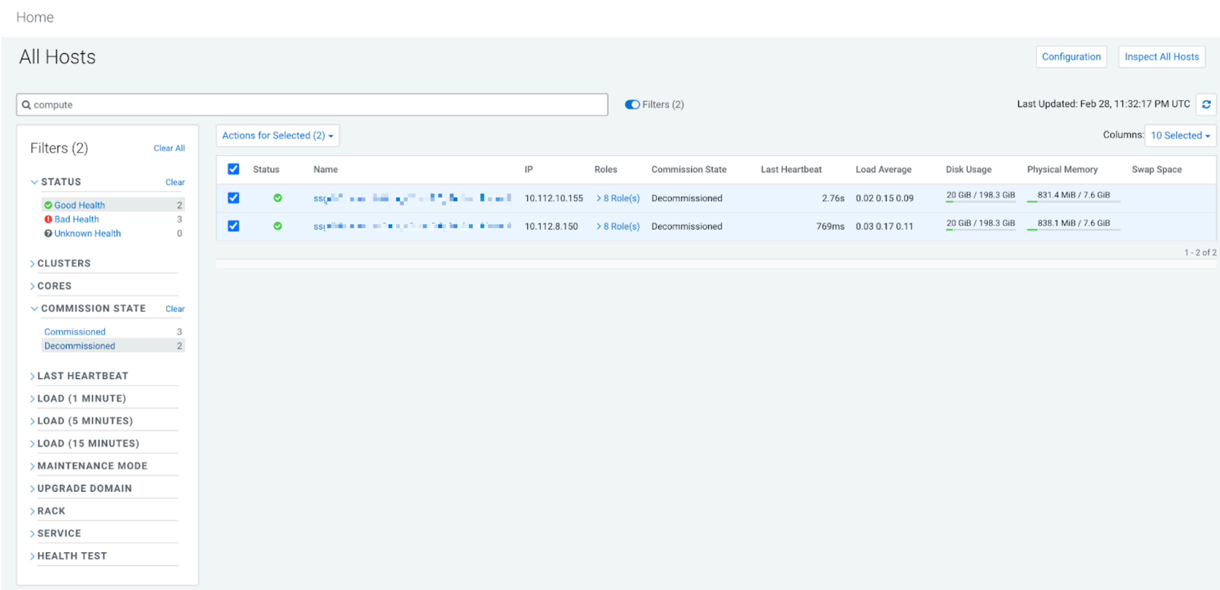
Parameter	Description
--cluster	Required. The cluster name or CRN of the cluster for which you want to fetch the particular scaling activity.
--operation-id	Required. The operation ID of the scaling activity that you want to fetch.

Manually recovering from load-based scaling failures

The mechanics for scaling are straightforward. There are four operations performed, and instances are not deleted during a load-based scale-down. Scale-up and scale-down operations are also meant to fail relatively fast, so that manual repair operations, if necessary, can be performed faster. Additionally, these operations can also be performed if there is another Management Operation (other than scaling) already in progress, which would otherwise block scaling.

The operations performed during a load-based scale-up and scale-down are outlined below.

- Scaleup: Start Instances on Cloud Provider: Specific instances for scale-up are started via the cloud provider. This can be replicated manually by changing the 'Instance State' on the cloud provider console.
- Scaleup: Commission Services via Cloudera Manager: The next step is to commission the services on these nodes. To replicate this manually, find these hosts using the filter: Status: Good Health, CommissionState: Decommissioned. Select them, then click Actions for Selected End Maintenance (Enable Alerts/Recommission) .



- Scale-down: Decommission Services via Cloudera Manager: This is the first step of a downscale operation. This can be replicated manually by selecting the relevant hosts in ClouderaManager as above, then click Actions for Selected Begin Maintenance (Suppress Alerts/Decommission) .
- Scale-down: Stop Instances on Cloud Provider: This is the second step of a scale-down operation, and specific instances are stopped via AWS or Azure. This can be replicated manually by changing the 'Instance State' on the cloud provider console.

Given the simplicity of the operations, and the fact that all the operations can be easily replicated by a cluster administrator, most failures can be handled quickly by following one or more of the procedures above. Some specific scenarios are outlined below.

Scale-up failures, where capacity is required

1. On the Autoscale tab for the cluster, use the toggle to disable autoscaling on the cluster.
2. Find stopped instances on the cloud provider console for the specific host group, and start them.

3. Wait for a few minutes for the instances to show up marked as "healthy" in Cloudera Manager (use the filter: Status: Good Health, Commission State: Decommissioned). Select instances from the appropriate host group and End Maintenance.

Cloudera Management Console has the following message: "Cloudera Manager reported health … … hostname1: [This host is in maintenance mode.], hostname2: [This host is in maintenance mode.]"

The cluster moves to 'Node Failure' state, and autoscaling is disabled. This typically indicates that some nodes are in the 'STARTED' state on the cloud-provider, but they are in 'Maintenance Mode' in Cloudera Manager. To remediate this, find the affected nodes in Cloudera Manager, and perform the steps in Scaleup: Commission Services via Cloudera Manager above. Wait for a few minutes for the Cloudera Management Console to sync state. After this, the cluster should move to the running state, and autoscale is re-enabled. Alternately, you can stop the specific nodes on the cloud provider.

Not enough nodes available - i.e. Autoscale configured with maxNodes, but the instance group has fewer than maxNodes available

Autoscaling will commission as many nodes as are available, and log an error in the Cloudera Management Console indicating that not enough nodes are available. Once all nodes are started, it will continue to try, causing a loop in which 0 nodes are added in each upscale attempt. To remedy this, when the cluster has all nodes in the RUNNING state, disable autoscaling and perform a scale-up of the host group to reach the 'maxNodes' configured for autoscaling (Actions Resize). Re-enable autoscaling.

Autoscaling FAQ

Review frequently asked questions about Cloudera Data Hub autoscaling.

A node is not running any tasks, why is it not being scaled down?

There are multiple possibilities:

- The cluster may already be at min-node-count.
- Check if running on a Cloudera Runtime version prior to 7.2.15, with max-capacity of queues not set to 100%.
- Check user-limit-factor. Is this preventing YARN from executing additional tasks for a user?
- Nodes that are not running containers, but have run containers for a running job, will not be considered for load-based downscaling. This is because the data generated on these nodes could be required while the job is running.

Will load-based autoscaling remove nodes that have running tasks?

Not unless the maximum node count limit is reduced. This forces the removal of certain nodes to reach the new limit.

I've encountered the following error: "Autoscaling Trigger Failed. Autoscaling Collection metrics via user <CRN> failed." How should I proceed?

For Cloudera accounts with a large number of users, autoscaling might encounter failures when capturing cluster metrics to scale up/down. If you encounter this issue please contact Cloudera Support.

Upgrading Data Lake or Cloudera Data Hub database

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

About this task

Note: If you are upgrading an Azure Single Server database to Azure Flexible Server, you can read a specific process description in Upgrading Azure Single Server to Flexible Server.

Several Cloudera services, including the Data Lake cluster and the Cloudera 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 Cloudera Data Hub templates are hosted on external instances that are provisioned during the initial deployment of the respective service. For these external databases, Cloudera 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 Cloudera 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 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, it is recommended to upgrade to PostgreSQL 14 when the upgrade is available to you in Cloudera.

Important: In order to avoid disruption to the deployed Data Lake and Cloudera 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 Cloudera on cloud is performed before the End of Life date.

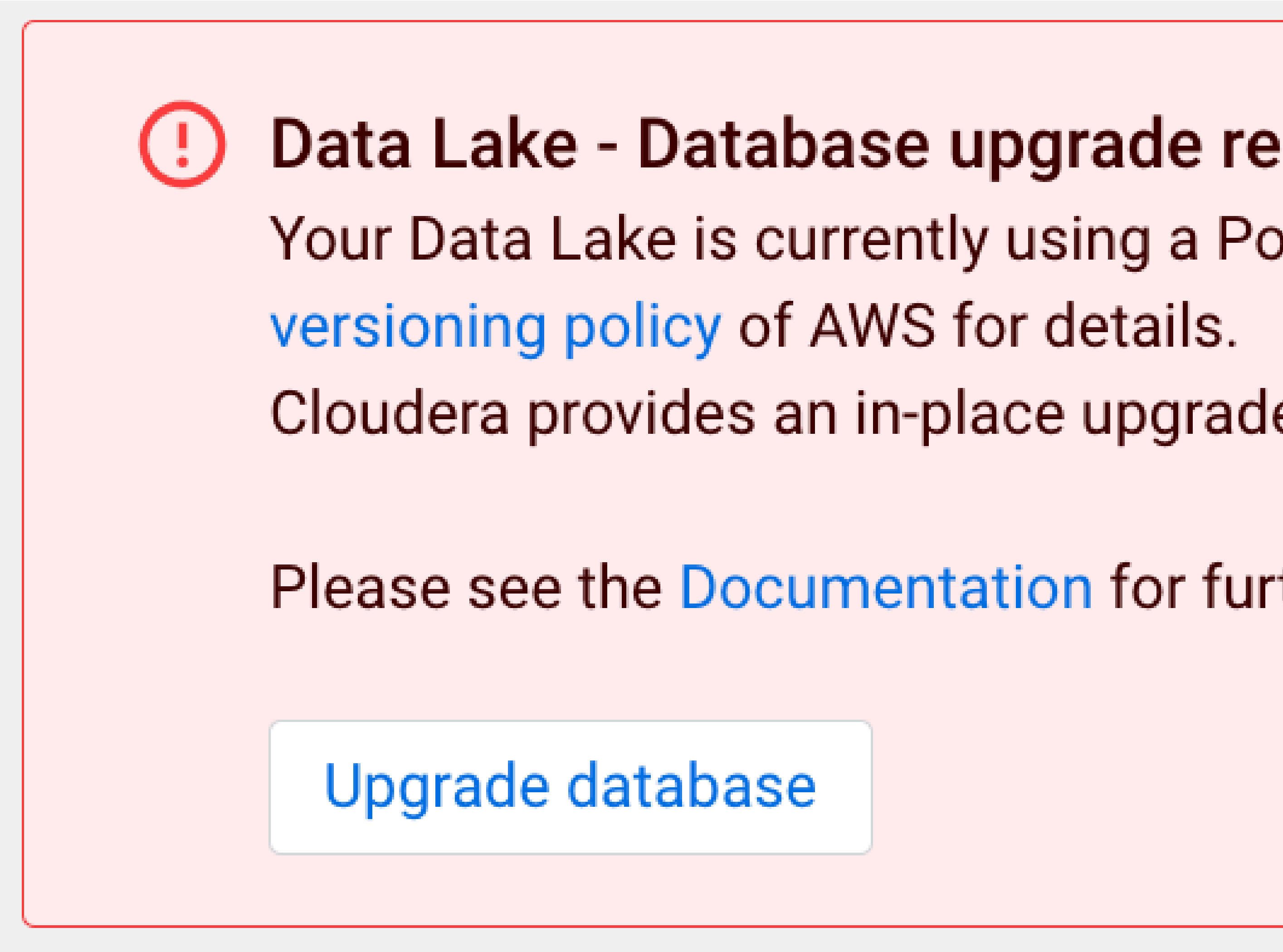
If you wish to disregard this recommendation, you may do so considering the risks involved as per the Cloud Service Provider policies.

The database upgrade is a separate operation, complementary to the existing maintenance, minor/major version and OS upgrades, as described in the Cloudera on cloud Upgrade Advisor.

This is a one-time operation. Once the database of a Data Lake or Cloudera 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 Cloudera Management Console UI.



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

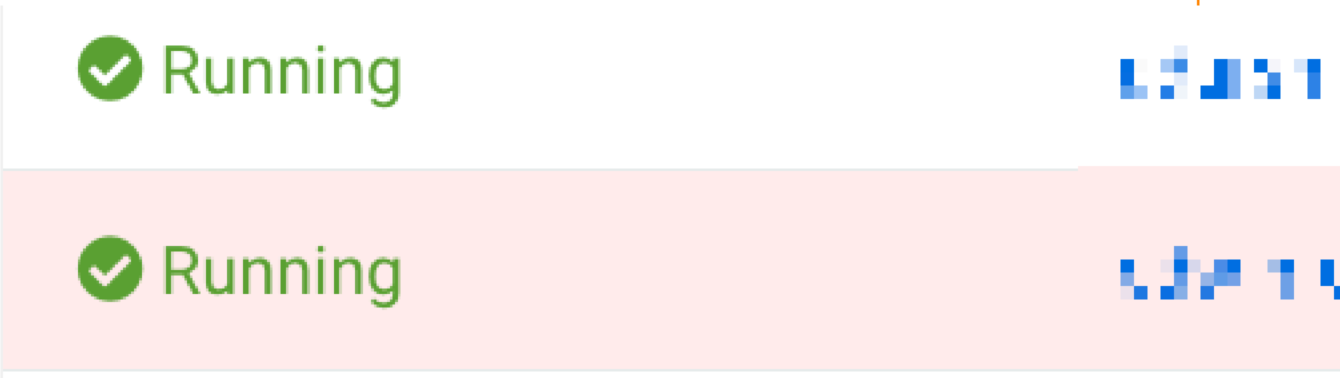
Instructions

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

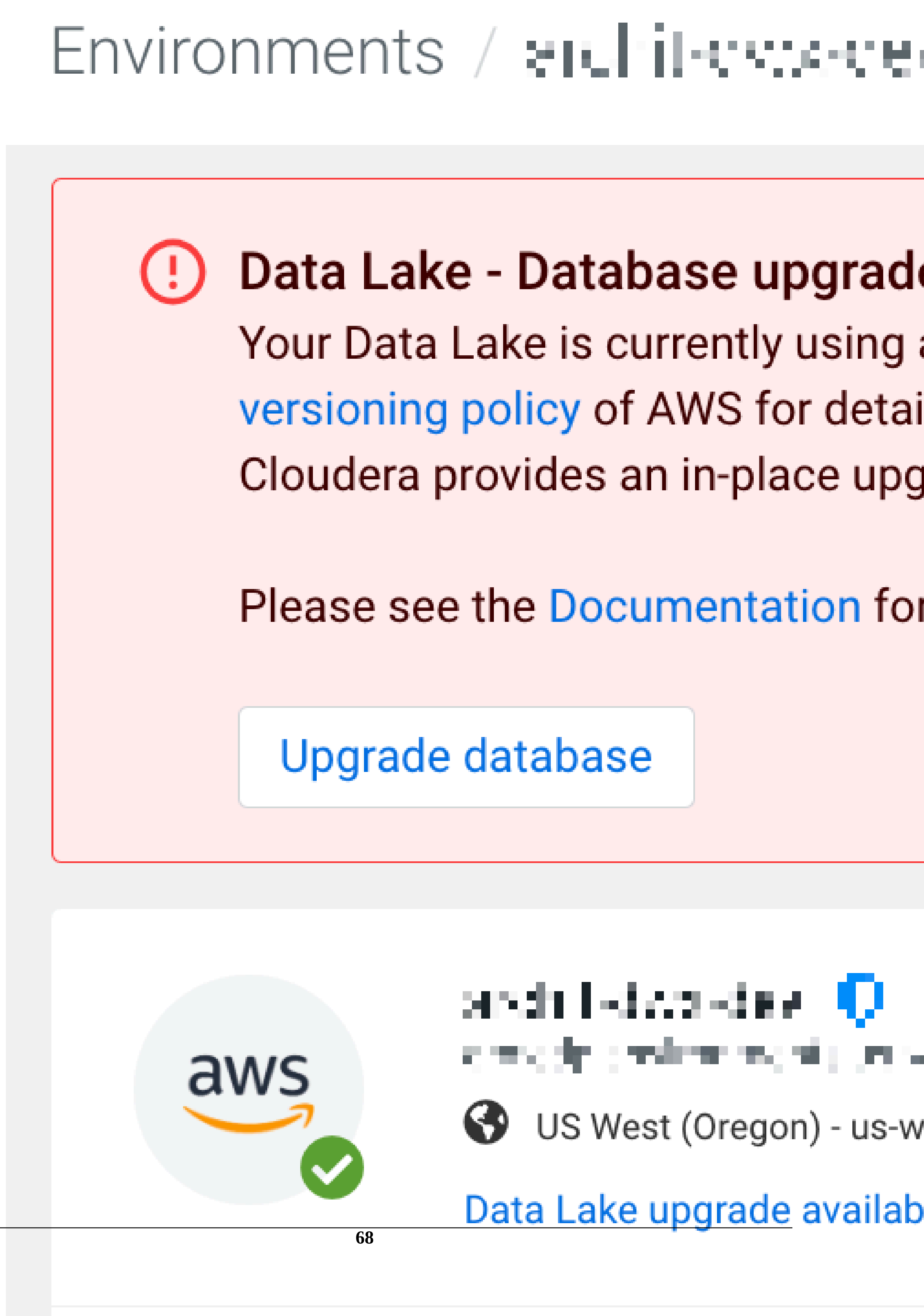
For Cloudera UI

Steps

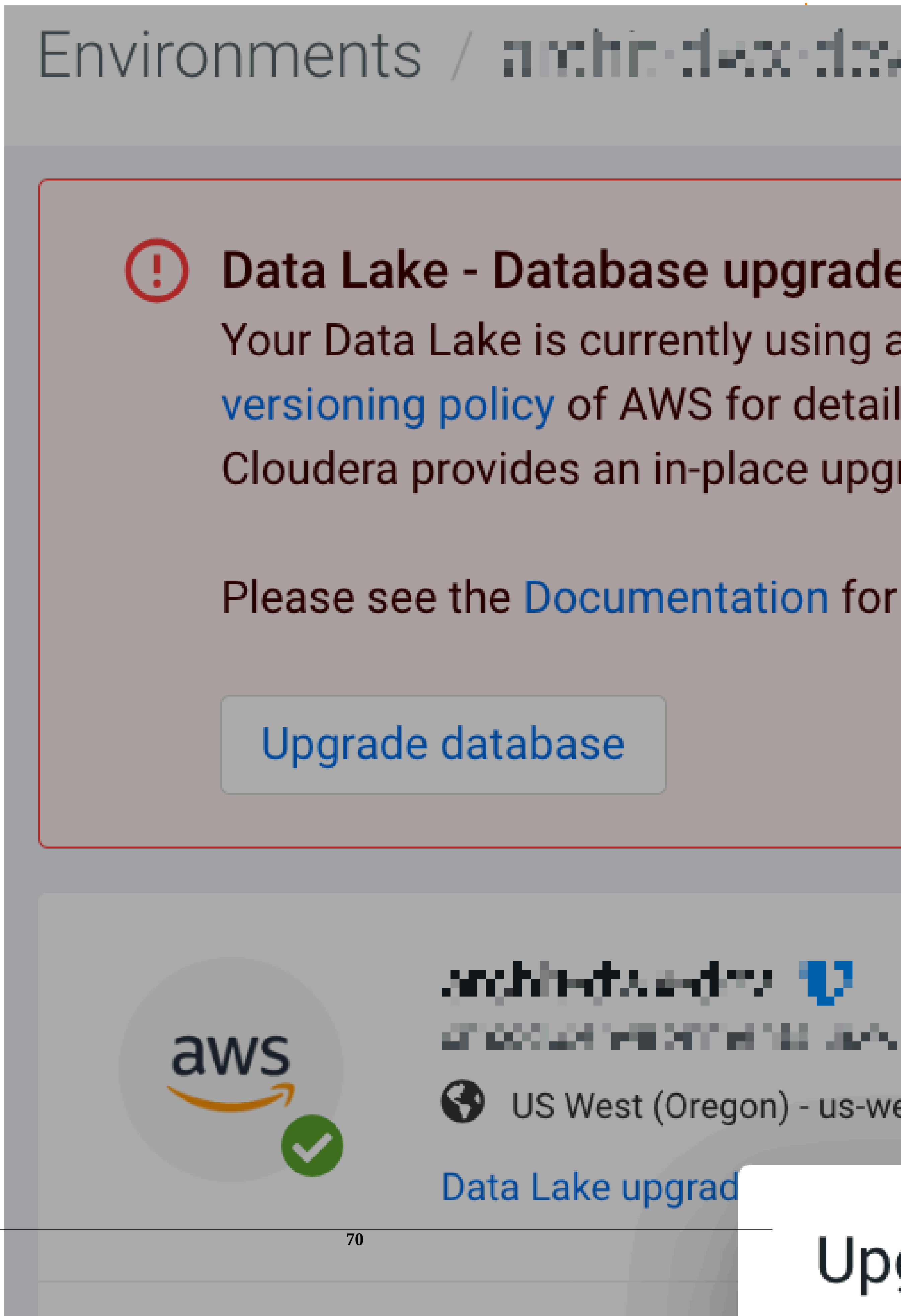
1. In Cloudera Management Console, go to Environments. Select the cluster to perform the upgrade from the list of available clusters. The clusters are eligible for this upgrade are indicated in the right most column:



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
versioning policy of AWS for detai
Cloudera provides an in-place upg
Please see the Documentation fo
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 upgr
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 Cloudera Data Hub clusters 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 Cloudera 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 PostgreSQL 14.

```
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_14
```

Cloudera Data Hub Database upgrade:

You can perform Cloudera 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 PostgreSQL 14.

```
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_14
```

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 Cloudera Runtime or OS upgrade.

Note: As part of the database upgrade operation, PostgreSQL 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 Cloudera Data Hub clusters and ways to troubleshoot them.

Note: For troubleshooting information related to upgrading to Azure Flexible Server, see Troubleshooting Flexible Server. When you are upgrading an Azure Single Server database to PostgreSQL version 14, the Azure Single Server is automatically updated to Azure Flexible Server.

Known limitations and troubleshooting:

- Performing the Database Upgrade on Cloudera Runtime versions 7.2.6 or below

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

  Workaround: You can request an entitlement that allows the Database Upgrade to be performed on older Cloudera Runtime versions on an exceptional basis.
- Performing the Database Upgrade on Data Lakes with attached Cloudera Data Hub clusters that cannot be stopped

  Technically, the Database Upgrade can be performed on a Data Lake without stopping the attached Cloudera Data Hub clusters. 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 Cloudera Data Hub clusters 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 Cloudera Data Hub cluster on an exceptional basis.
- PostgreSQL client binaries will be 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

  Cloudera 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 Cloudera 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 Cloudera 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 Cloudera on cloud workloads may be seriously impacted.

  Workaround: Cloudera recommends performing the Database Upgrade via the Cloudera 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 Cloudera Data Hub clusters 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.

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.

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
     ```

Restarting instances

If you have instances that have been stopped on the cloud provider side or ones that are recommended to be restarted due to high CPU and memory usage, you can restart them without the need to start the instance on the cloud provider side or having to perform a repair operation on the stopped instance.

If the restart operation is performed on an instance in running state, then the restart operation stops the instance first, and then starts it. If the restart operation is performed on an instance that is already in stopped state, only the start operation is performed.

You can perform the restart operation even if the instances are in an unhealthy state or even if the cluster is in an unhealthy state.

Required roles

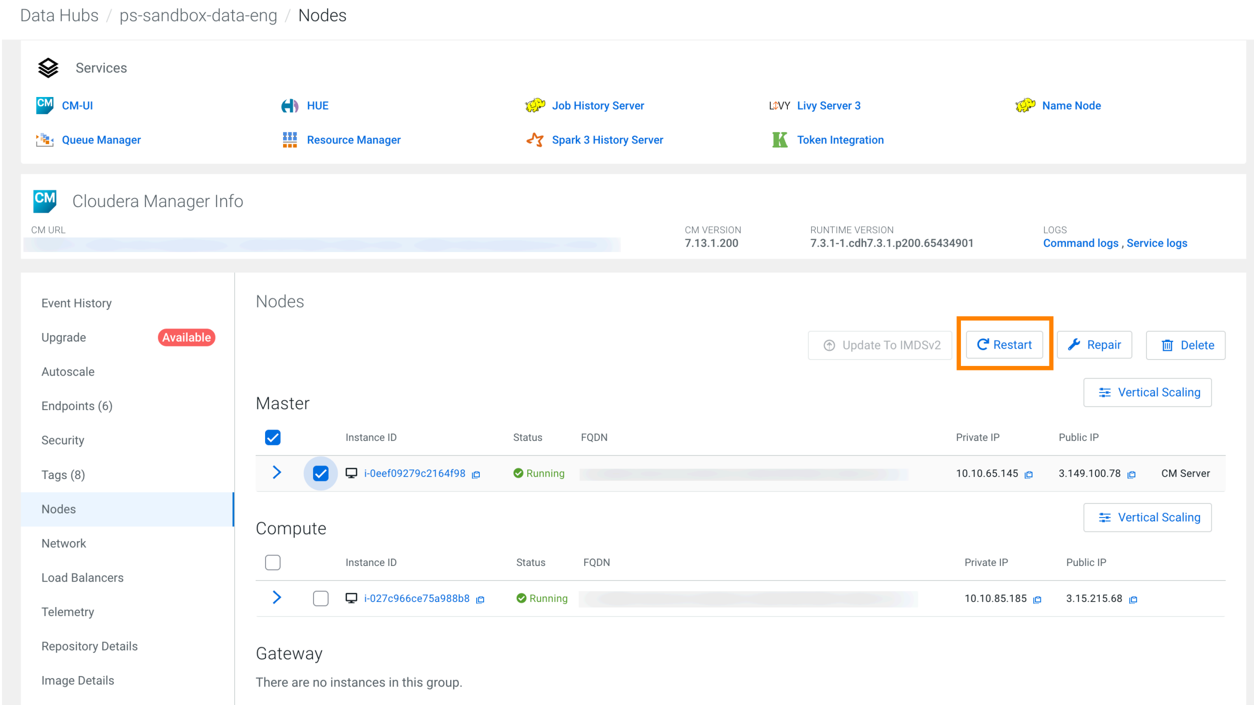
- Owner
- DataHubAdmin

Limitations

- These commands restart the instances directly on the cloud provider side but the instance can still be in decommissioned state in Cloudera Manager. Therefore, you might need to manually recommission the instances in Cloudera Manager or wait for autoscaling to happen.
- On the Cloudera Manager side, some services on the instance might become unhealthy. In the recovery flow of autoscaling they will be picked up for recovery but it is possible that they cannot be recovered. This is only true for the hostgroups where autoscaling is supported and an autoscaling policy is defined.

Cloudera Management Console steps

1. Navigate to your environment in Cloudera Management Console.
2. Select the Data Hub tab on the **Environment details** page.
3. Select the Cloudera Data Hub cluster that you want to restart.
4. Select Nodes.
5. Select the instance you want to restart.
6. Click Restart.



CLI commands

Use the following commands to restart instances:

```
cdp datahub restart-cluster-instances \
--cluster DATAHUB_NAME_NAME_OR_CRN \
--instances [INSTANCES ...]
```

Example:

```
cdp datahub restart-cluster-instances \
--cluster deazuremowdev \
--instances deazuremowdev147180w4-cefc0a29 deazuremowdev147180w3-a47b4ace
deazuremowdev147180w2-be8e75d7
```

Rotating database certificates

AWS requires the rotation of the SSL/TLS certificates used for secure communication between Cloudera on cloud Data Lakes and certain Data Hubs and the external AWS RDS database instances that they rely on. Cloudera on cloud provides multiple options to perform the required RDS certificate rotation.

Cloudera on cloud Data Lakes and certain Cloudera Data Hub clusters rely on Amazon Web Services (AWS) Relational Database Service (RDS) database instances that are provisioned by Cloudera during the creation of the respective resources. Similarly to other compute resources, these database instances are created in the cloud accounts of customers. The deployed Cloudera resources use a secure connection to communicate with the database using the SSL/TLS certificates issued by AWS.

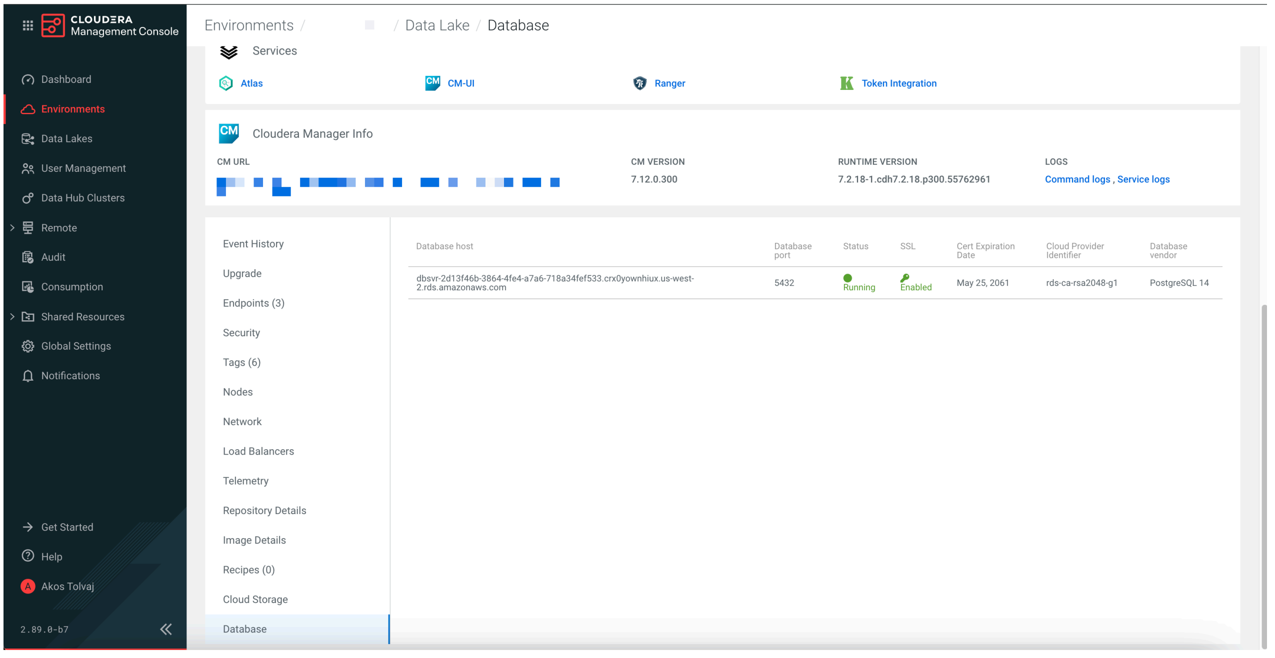
SSL/TLS certificates used by Amazon RDS instances created before December 2022 are potentially using certificates that will expire in 2024. Amazon RDS instances with an expiring certificate require that a certification rotation is performed. This will replace the soon expiring certificate with a newer one. If the RDS certification rotation is not performed by 22nd August, 2024, Amazon will perform the rotation for each RDS instance and as a result, the affected Cloudera clusters will not be able to connect to the external database instances.

The actions you need to take depend on whether you have SSL enforcement enabled or disabled on the Data Lake or Cloudera Data Hub cluster as shown in the **Database** tab of the Data Lake or Data Hub details page. This setting controls whether the database server mandates the use of an encrypted connection. It is recommended to check SSL enforcement setting on the Data Lake of your environment and separately for each Cloudera Data Hub cluster.

It is strongly recommended that, when SSL enforcement is enabled, you do not perform the RDS certificate rotation directly using AWS tools, as this can lead to a service degradation or an outage of Cloudera services using the respective database. Instead, follow the instructions below so that the new certificates can be properly installed by Cloudera before changing them on the database instances. Once you have performed the required steps in Cloudera, there are no additional actions needed in the AWS console. During the database certificate rotation, Cloudera will automatically make the changes required and rotate the certificate of the attached AWS RDS database.

To determine whether you have SSL enforcement enabled on your Data Lake, perform the following steps:

1. Sign in to the Cloudera web interface.
2. Navigate to Cloudera Management Console Environments .
3. Select the environment by clicking the environment name.
4. Click the Data Lake tab.
5. Scroll to the bottom of the page and select Database.



Similarly, perform the following steps on each Cloudera Data Hub cluster to determine whether you have SSL enforcement enabled:

1. Sign in to the Cloudera web interface.
2. Navigate to Cloudera Management Console Environments .
3. Select the environment by clicking the environment name.
4. Click the Data Hub tab.
5. Select a Cloudera Data Hub cluster.
6. Scroll to the bottom of the Data Hub details page and select Database.

If the value for **SSL** is **Enabled**, it means SSL enforcement is enabled on your Data Lake or Cloudera Data Hub cluster, and you must perform certificate rotation as described in *Rotating database certificates when SSL enforcement is enabled*. If the value is **Disabled**, no action is required on that particular Data Lake or Cloudera Data Hub cluster. For further information, see *Rotating database certificates when SSL enforcement is disabled*.

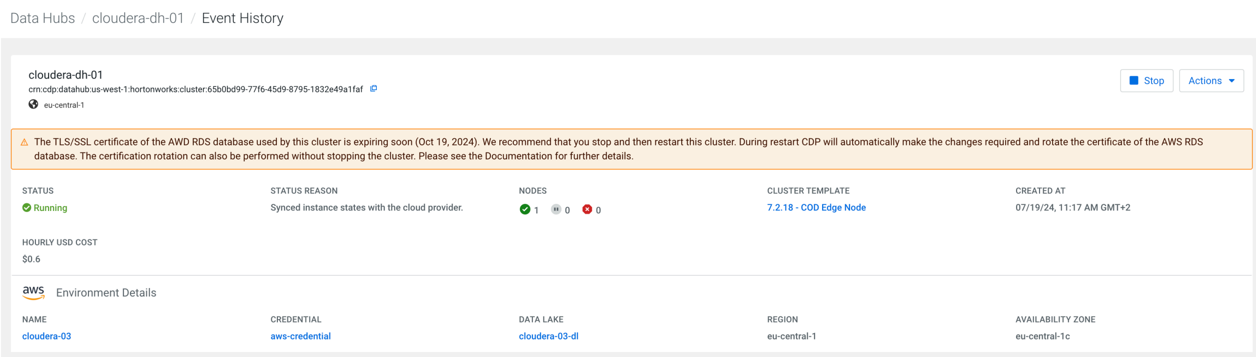
Rotating database certificates when SSL enforcement is enabled

When SSL enforcement is enabled on your database server, do not perform the RDS certificate rotation directly using AWS tools, as this can lead to a service degradation or an outage of the respective resource. Use the options provided by Cloudera on cloud to perform the required RDS certificate rotation instead.

Required role (Data Lakes): EnvironmentAdmin

Required role (Data Hub): DataHubAdmin or EnvironmentAdmin

If the following warning message is displayed on top of the Cloudera Management Console window, you must perform the TLS/SSL certificate rotation:



If you do not see the warning message, your clusters are not affected or the rotation has already been performed when you recently stopped and started your cluster or the environment.

Cloudera provides two options for rotating the expiring database certificates, with and without stopping your cluster:

Rotating the database certificate by stopping and starting your cluster

This is the default and most simple way to let Cloudera perform the certificate rotation and all changes required.

Instructions

1. Stop all Cloudera Data Hub clusters where the warning about expiring database certificates is shown.
2. Stop and start your Data Lakes where the warning message is displayed.

   Cloudera will automatically perform all Data Lake and AWS RDS changes required.
3. Start all Data Hubs that you have previously stopped.

   Cloudera will automatically perform all Cloudera Data Hub and AWS RDS changes required where the warning message is displayed.

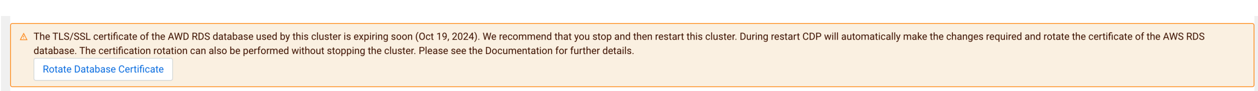
You can perform the stop and start either via UI or CLI, see Stop a cluster.

Note: It is recommended that you also suspend workloads running on Data Services attached to the Data Lake during this operation.

Rotating the database certificate without stopping the cluster

Stopping the Data Lake or Cloudera Data Hub clusters is not feasible in some scenarios, as workloads need to be stopped and data stored on ephemeral disks or ephemeral cache is lost during cluster restart.

In certain cases, Cloudera allows the database certificate rotation to be performed without stopping your cluster. If your cluster supports this, you will see a Rotate Database Certificate button in the warning message.



The Rotate Database Certificate button is only available if a rolling restart is supported on your cluster. For the list of supported versions, see the Data Lake rolling upgrades and the Cloudera Data Hub rolling upgrades documentations.

Important:

The certificate rotation can also be performed on Cloudera Operational Database databases by opening the cluster backing the Cloudera Operational Database database in the Data Hub screen and clicking the Rotate Database Certificate button there. However, this is only supported for Cloudera Operational Database clusters whose storage is selected as HDFS or cloud without ephemeral storage, and not supported on using Cloud Storage with Caching storage type.

Under some circumstances, a rolling certificate rotation may not be supported for a Data Lake or Cloudera Data Hub cluster, but can be enabled through entitlement. For information about obtaining this entitlement, contact Cloudera Customer Support.

Instructions

To perform a rolling restart, click the Rotate Database Certificate button on top of the screen, inside the warning message.

Note: Your cluster must be in a healthy status, otherwise the certificate rotation is not possible.

Important: You must rotate the certificates on each Data Hub cluster first, and then proceed with rotating the certificates on the Data Lake. Otherwise, the Data Lake rotate database certificate operation will fail with a warning message.

You can also use the following CLI commands to perform the certificate rotation with minimal downtime and without stopping the cluster. The same roles are required as for the Rotate Database Certificate action through the UI.

1. Use the following command to rotate the certificate for each Data Hub cluster.

```
cdp datahub rotate-db-certificate --datahubName <VALUE CAN BE cluster CRN
 or name>
```

2. Use the following command to rotate the certificate for the Data Lake.

```
cdp datalake rotate-db-certificate --datalakeName <VALUE CAN BE cluster
CRN or name>
```

Note: If you are performing the certificate rotation using the Rotate Database Certificate button on the UI that was made available to you through entitlement or if you are using the CLI commands, it is possible that a rolling restart will not happen because it is not supported on your cluster. In such cases, a regular restart action will be evoked. In this case, minimal downtime can be expected, however, your cluster will not be stopped.

Overriding default root certificate

Cloudera automatically chooses the target root certificate for the RDS instance. Currently, the following certificate authority (CA) certificates are available for the RDS database:

- rds-ca-rsa2048-g1, which is valid for 40 years
- rds-ca-rsa4096-g1, which is valid for 100 years
- rds-ca-ecc384-g1, which is valid for 100 years

By default, rds-ca-rsa2048-g1 is the root certificate in all supported regions that is picked as a target certificate during the certificate rotation. You have the option to override the default certificate, and set one of the certificates available in your region as a target root certificate that will be picked during the certificate rotation. Overriding the default root certificate is not tied to any particular RDS instance, but applicable to the whole AWS region in the target AWS account.

The list of available root certificates in your region can be retrieved with the following AWS CLI command:

```
aws --region [*** REGION ***] rds describe-certificates
```

The command returns the output JSON containing the properties of the RDS root certificates that are currently available in the given region. The properties also detail if an override is already set for the root certificate.

If there are more than one entries, you can use the following command to set one of the root certificates as a target, and override what is configured as default:

```
aws --region [*** REGION ***] rds modify-certificates --certificate-ident
ifier [*** ROOT CERTIFICATE ***]
```

The following command is an example to override the default rds-ca-rsa2048-g1 certificaate in the us-west-1 to rds-ca-ecc384-g1:

```
aws --region us-west-1 rds modify-certificates --certificate-identifier rds-
ca-ecc384-g1
```

Note: You can also set up the override for the default root certificate. This ensures that the root certificate will remain the same in case the default certificate is changed by AWS. You can use the following command to remove the override:

```
aws --region [*** REGION ***] rds modify-certificates --remove-custom
er-override
```

The command returns the root certificate that will be used by default.

Related Information

Describing certificates | AWS CLI

Modifying certificates | AWS CLI

Rotating database certificates when SSL enforcement is disabled

If the Data Lake for your environment or your Cloudera Data Hub cluster is using an RDS where SSL enforcement is disabled, no action is required on your side. You can simply let the root certificate expire and be replaced by AWS upon expiry.

It is recommended to check SSL enforcement setting on the Data Lake of your environment and separately for each Cloudera Data Hub cluster.

A Data Lake or a Cloudera Data Hub using an RDS that is shown as SSL Disabled are essentially immune to the validity of the RDS root certificate. This is because DB connections made from Cloudera cluster services do not explicitly validate the certificate chain received from the RDS instance in such cases. The AWS RDS instance may, therefore, be left as is safely, letting its root certificate expire and be replaced automatically by AWS upon the expiry date.

Alternatively, you can instead opt to change the RDS root certificate manually using standard AWS tools like the AWS RDS Console or the AWS CLI, as described in Updating your CA certificate by modifying your DB instance or cluster in the AWS documentation.

When manually rotating the root certificate, you have the option to choose from the following available RDS certificates based on your region:

- rds-ca-rsa2048-g1, which is valid for 40 years
- rds-ca-rsa4096-g1, which is valid for 100 years
- rds-ca-ecc384-g1, which is valid for 100 years

Cloudera does not provide automation for the rotation of the RDS root certificate for databases where SSL enforcement is disabled.

Note: The Cloudera Management Console does not reflect the change of the RDS root certificate after any manual rotation.

Rotating Cloudera Data Hub secrets

To strengthen the security of your deployments, you can rotate sensitive secrets, such as database passwords or admin credentials for the Cloudera Data Hub cluster. These secrets are managed and created by either Cloudera or users.

Secret rotation can be performed using the Cloudera Management Console or CLI commands. By rotating secrets, you reduce the risk of unauthorized access and enhance the overall security of your environment. A single secret rotation typically takes no longer than five minutes, minimizing downtime and disruption.

The following table summarizes the list of secrets that can be rotated for Cloudera Data Hub:

Secret name	Secret description
Cloudera Admin password (CM_ADMIN_PASSWORD)	Used by the Cloudera Control Plane to manage Cloudera Manager, issue commands and poll info.
External Database Root Password (EXTERNAL_DATABASE_ROOT_PASSWORD)	Initializing and managing External Databases created by the Cloudera Control Plane. Creates initial users and databases through this credential.
Cloudera Manager Database Password (CM_DB_PASSWORD)	Credentials Cloudera Manager uses to connect to External Database.
Cloudbreak user root SSH public key (USER_KEYPAIR)	Public SSH key specified during the environment creation. Before rotating the SSH public key, you need to change keys on the Environment summary page, then rotate the secret for Cloudera Data Hub.
Cloudera Manager Services Database Password (CM_SERVICE_DB_PASSWORD)	Used by LOCAL services managed by Cloudera Manager to connect to the External Database.This could include multiple services, such as Hue, Atlas, Ranger and so on.
LDAP Bind Password (LDAP_BIND_PASSWORD)	Password used to connect and fetch user and group context from LDAP located on the FreeIPA nodes.
SSSD password (SSSD_IPA_PASSWORD)	Used to manage the SSSD service, which provides unified management of authentication methods on the cluster such as ssh, keytab generation and so on.
Databus access key (DBUS_UMS_ACCESS_KEY)	Machine user service credential, used for communicating with Cloudera Control Plane through the DBUS interface by services such as the metering agent, diagnostic bundle collection and telemetry publisher.
Salt boot secrets (SALT_BOOT_SECRETS)	Used for bootstrapping new Virtual Machine to the cluster during cluster creation, upscale operation, OS upgrade and repair.
Salt sign key pair (SALT_SIGN_KEY_PAIR)	Used to sign and verify files or data distributed to Salt minions. Ensures integrity and authenticity of data managed by the Salt system.
Salt master key pair (SALT_MASTER_KEY_PAIR)	Used to establish secure communication between the Salt master and minions. The public key is shared with the minions to verify the identity of the master.
Salt password (SALT_PASSWORD)	Salt user's password used to communicate with the Salt cluster.

Secret name	Secret description
Nginx server side private key (NGINX_CLUSTER_SSL_CERT_PRIVATE_KEY)	Private key of server side NGINX SSL certificate used for communication with internal services like salt-bootstrap.
Compute monitoring credentials (COMPUTE_MONITORING_CREDENTIALS)	Credentials used for compute monitoring components (prometheus, request-signer, etc.).
Cloudera Manager Intermediate Certificate Authority (CM_INTERMEDIATE_CA_CERT)	Used by Cloudera Manager and Cloudera Manager agents to serve HTTPS apis on the cluster nodes. Root CA is located and signed by FreeIPA.
Embedded DB SSL certificate (EMBEDDED_DB_SSL_CERT)	Private key of embedded postgres database SSL certificate. **Note:** Before rotating the certificate, ensure that the Cloudera Data Hub cluster in in a healthy state to avoid any failure for the secret rotation. For more information, see the Lifecycle of embedded database certificates section.

The secrets vary based on the deployment, you can use the following CLI command to list all of the available secrets for rotation:

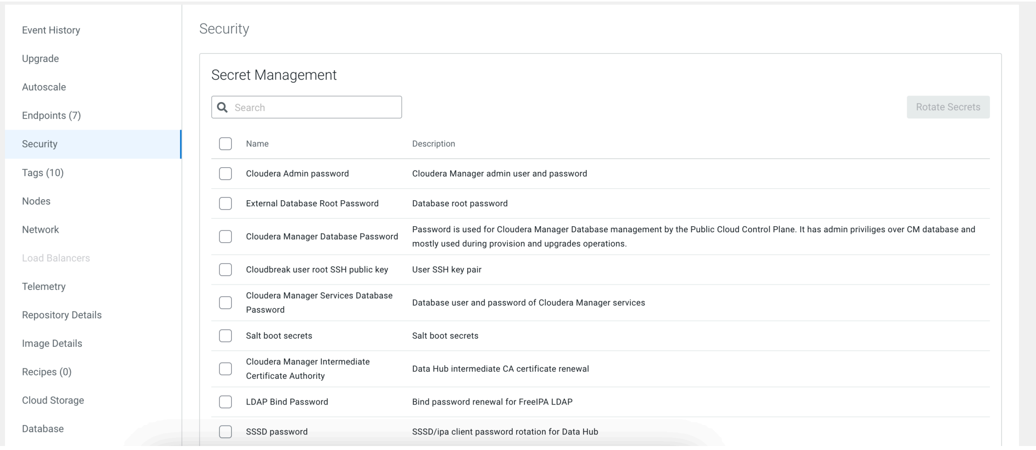
```
cdp datahub list-datahub-secret-types --datahub {CRN}
```

You can use the following steps in Cloudera Management Console or CLI commands to rotate the Cloudera Data Hub secrets:

For Cloudera Management Console

1. Navigate to your environment in Cloudera Management Console.
2. Click Data Hub on the environment details page.
3. Select the Cloudera Data Hub cluster, where you want to rotate the secrets
4. Click Security.

   Under Secret Management, the list of secrets that can be rotated will be displayed:

5. Select the secrets that you want to rotate.
6. Click Rotate Secrets.

For CLI

Use the following command to rotate the specific secret types:

```
cdp datahub rotate-secrets --datahub {DATAHUBCRN} --secret-ty
pes {SECRETENUM1,SECRETENUM2}
```

Lifecycle of embedded database certificates

The certificates for embedded databases are valid for two years. These certificates are auto-generated when a Cloudera Data Hub cluster is created. Currently, there is no option for auto-renewal and there are no alerts shown in Cloudera Data Hub when the certificates are close to expiration.

In case a certificate expires, Cloudera Manager becomes unreachable and the state of the Cloudera Data Hub cluster becomes unhealthy. When the cluster becomes unhealthy due to the expired SSL certificate, the secret rotation described in Rotating Cloudera Data Hub secrets fails.

Warning: If an embedded database certificate expires, Cloudera Manager becomes inaccessible and the cluster may become unrecoverable without manual intervention. Avoid running Cloudera Manager repair during certificate expiry scenarios, and contact Cloudera support to manually recover the cluster and rotate the certificate.

The tool for manual recovery by Cloudera support is fully available only after the first successful secret rotation. For newly created clusters, where secrets have never been rotated, Cloudera support may need to rebuild or reconstruct the rotation assets during manual recovery.

How to check the expiration date for the certificates:

In order to avoid the expiration of certificates for embedded databases, you can use the following command to review the date of expiry for the SSL certificates:

```
echo | openssl s_client -starttls postgres -showcerts -connect "$(hostname -
f)":5432 | openssl x509 -noout -dates
```

You must rotate the embedded database certificate before the expiration date as described in Rotating Cloudera Data Hub secrets.

Managing public and private certificates

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

- Public certificates are Let's Encrypt-issued certificates for Cloudera 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 Cloudera will renew these certificates automatically before they expire.

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

  - Cloudera 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 Cloudera Data Hub clusters*. After the public certificate for a cluster has been manually renewed once from the Cloudera 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 Cloudera 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 Cloudera Control Plane time. If the Cloudera Control Plane is in the United States region, the renewal starts at 2 A.M. Pacific Daylight Time (PDT). If the Cloudera Control Plane is in the European region, the renewal starts at 2 A.M. Central European Summer Time (CEST). If the Cloudera 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 Cloudera Data Hub and Data Lake clusters or the 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 Cloudera Data Hub cluster through the Cloudera UI.

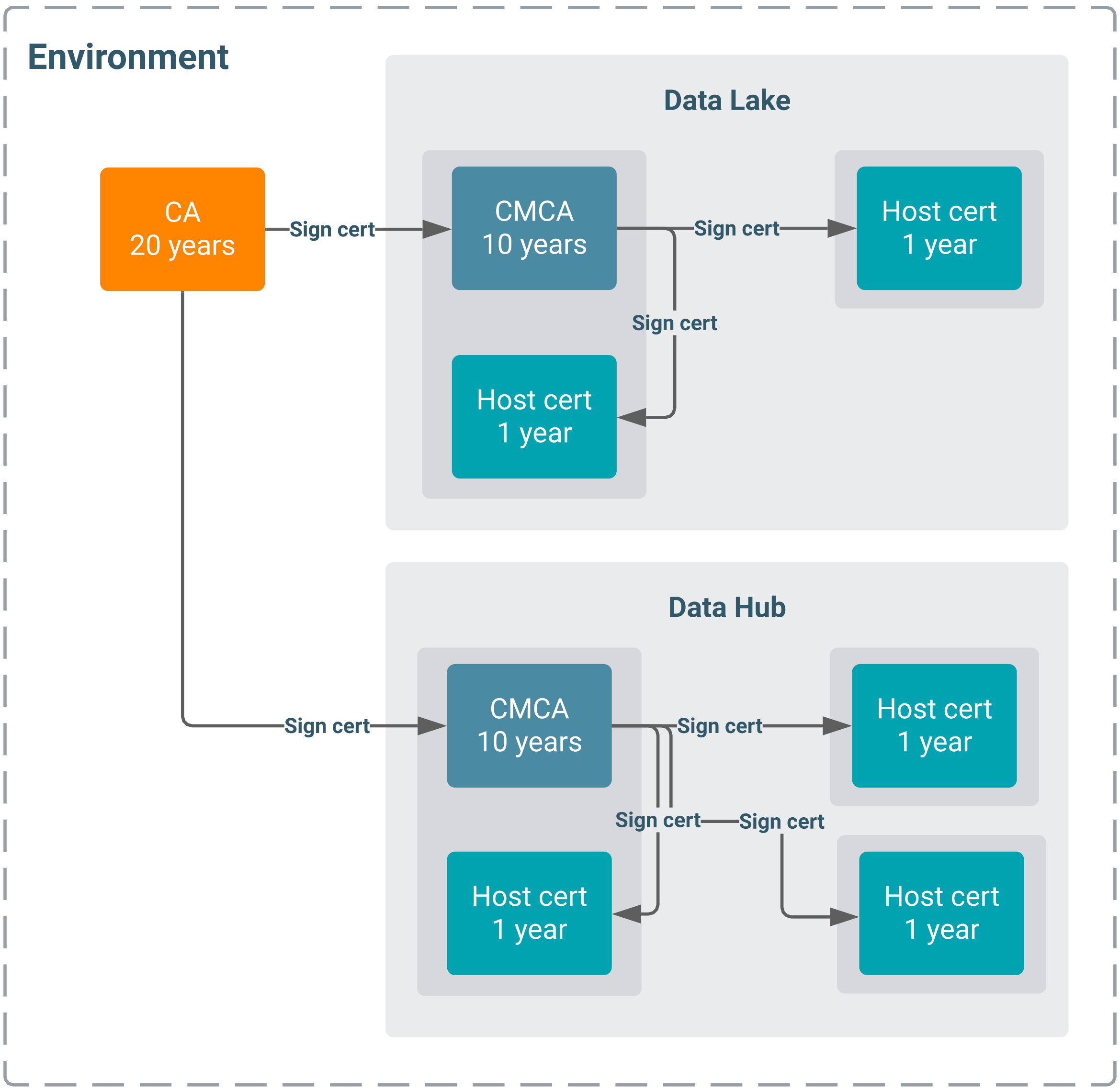
See *Manually renewing public certificates for Data Lake and Cloudera Data Hub clusters* for instructions on renewing the public certificates manually.

- Private certificates, or host certificates, are certificates created during cluster provisioning for every host with Auto-TLS.

  The root Certificate Authority (CA) is a self-signed certificate that is generated by FreeIPA during environment creation. The root CA certificate is used to generate the Cloudera Manager CA certificate for all Data Lake and Cloudera Data Hub clusters and for the Cloudera Runtime services during cluster provisioning.

  Note: Self-signed certificates are also used for NGINX port 9443 and salt-bootstrap port 7071. These certificates are necessary for the Cloudera Control Plane to manage the clusters, and they are not used by users or Cloudera Runtime services. These certificates are also not required for authentication or authorization.

  Private/host certificates have a default expiration date of one year. As private certificates get closer to expiration, the Cloudera UI displays a warning that the certificate is about to expire.



Though the Cloudera 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 Cloudera Data Hub clusters

Private (host) certificates have a default expiration date of one year; to keep the Data Lake and Cloudera 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 (Cloudera Data Hub): DatahubAdmin, Owner of the Data Hub, EnvironmentAdmin, or Owner of the environment

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:

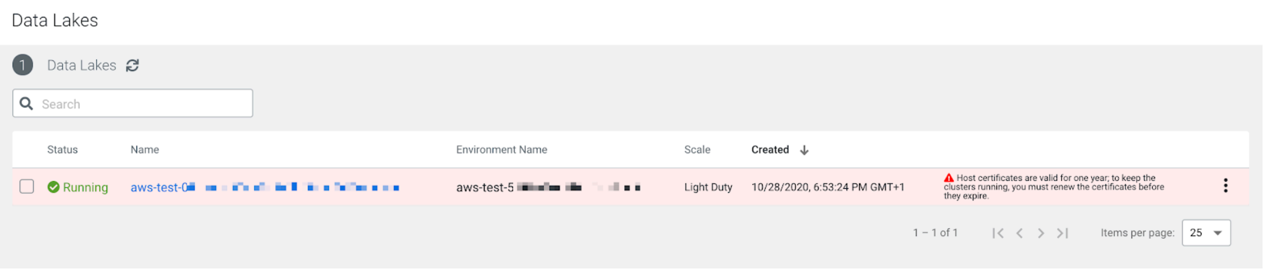
```
cdp datalake rotate-private-certificates --datalake <Data Lake name or CRN>
```

Cloudera Data Hub certificate renewal:

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

Alternatively, you can wait until the host certificate is close to expiration. During periodic cluster state synchronization, Cloudera 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, Cloudera displays a warning in the UI.

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



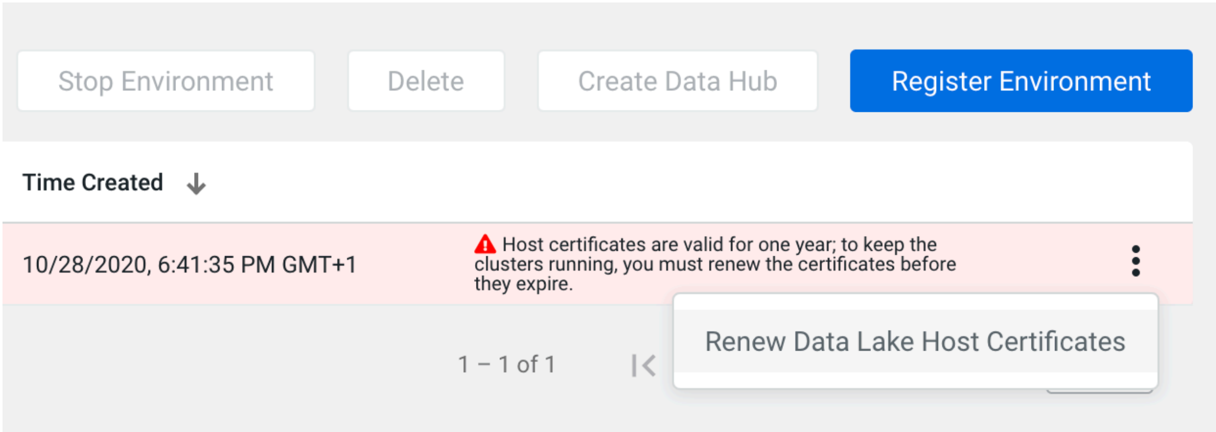
To renew the host certificate once you receive an expiration warning, follow the steps below.

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 Cloudera Data Hub clusters 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.

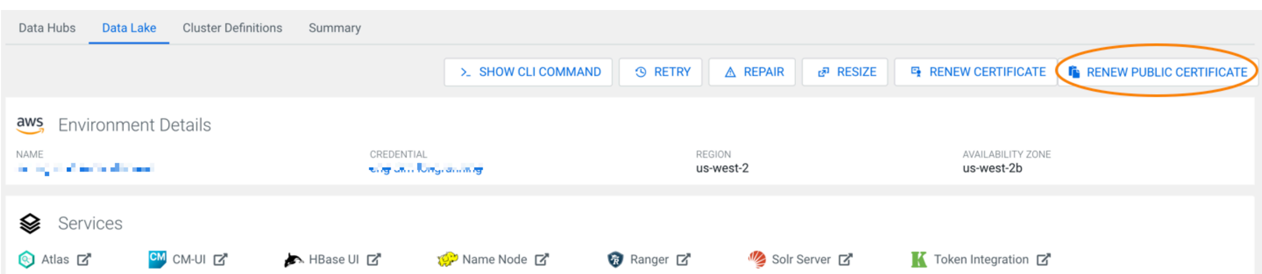
Manually renewing public certificates for Data Lake and Cloudera Data Hub clusters

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

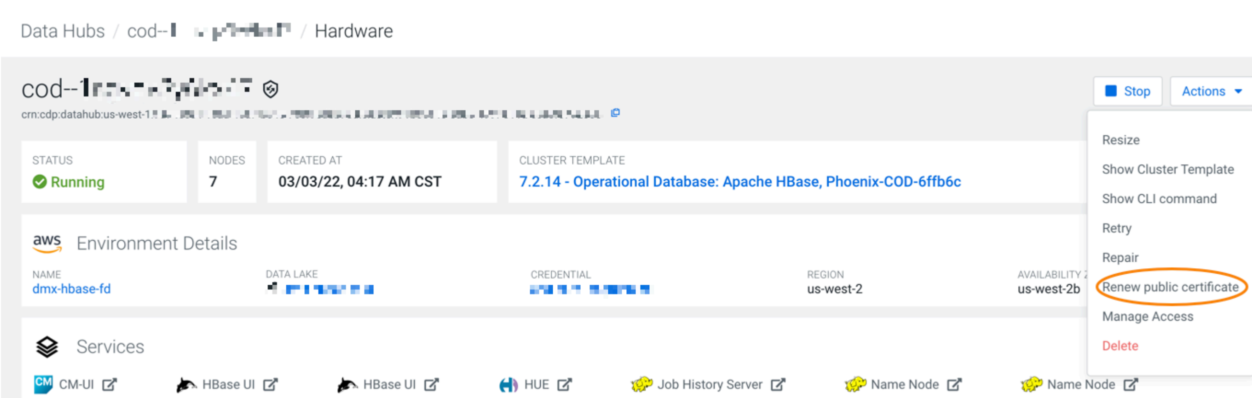
Required role (Data Lakes): EnvironmentAdmin or Owner of the environment

Required role (Cloudera 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 Cloudera 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>
```

Cloudera Data Hub public certificate renewal:

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

Monitoring clusters

You can monitor the status of your Cloudera Data Hub cluster from the Cloudera web UI or the CDP CLI.

Required role: The DataHubAdmin or Owner roles at the scope of Cloudera Data Hub allow you to manage the Cloudera Data Hub cluster. Note that EnvironmentAdmin and Owner of the environment can also manage Cloudera Data Hub clusters.

Monitoring Cloudera Data Hub clusters trough the UI

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

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

Monitoring Cloudera Data Hub clusters through the CLI

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

- List all available clusters: cdp datahub list-clusters
- Describe a specific cluster: cdp datahub describe-cluster --cluster-name <value>
- Get status of cluster hosts: cdp datahub get-cluster-host-status --cluster-name <value>
- Get status of cluster services: cdp datahub get-cluster-service-status --cluster-name <value>
-
- Get status of specified operation: cdp datahub get-operation --cluster-name <value> [--operation-id <value>]

  To use the get-operation command to get the status of a specified event, you need to specify the operation id of the operation. Every operation that starts a process running in the background, like creating, starting, stopping, or repairing a cluster, returns an operationId field in the response.

  Example:

```
cdp datahub create-aws-cluster [args]

{

    "operationId": "23e5bdb6-b3a1-4d25-95aa-262365ecc6b9",
    "cluster": {
        ...
    }
```

```
}
```

The value of this operationId can be used as the value for the --operation-id option for the get-operation command.

Example:

```
cdp datahub get-operation --cluster-name crn:cdp:datahub:us-west-1:9d74e
ee4-1cad-45d7-b645-7ccf9edbb73d:cluster:6eeb2804-c938-4564-9da1-a005250d
450b --operation-id 23e5bdb6-b3a1-4d25-95aa-262365ecc6b9
```

Output format:

```
{
    "operationId": "identifier of the operation",
    "operationName": "Short name of the operation",
    "operationStatus": "UNKNOWN | RUNNING | FAILED | FINISHED | CANCELLE
D",
    "started": "Start time of the operation"
    "ended": "End time of the operation if it is completed"
}
```

Output example:

```
{
    "operationId": "23e5bdb6-b3a1-4d25-95aa-262365ecc6b9",
    "operationName": "Provision",
    "operationStatus": "RUNNING",
    "started": "2025-01-15T10:31:37+00:00"
}
```

Unsuccessful operation statuses are stored for 2 weeks, while successful status operations are stored for 1 day.

The operation id is optional, and if it is omitted, the status of the last operation is returned.

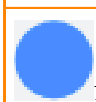
-
- Get status of latest operation: cdp datahub get-operation --cluster-name <value>

  Example:

```
cdp datahub get-operation --cluster-name crn:cdp:datahub:us-west-1:9d74e
ee4-1cad-45d7-b645-7ccf9edbb73d:cluster:6eeb2804-c938-4564-9da1-a005250d
450b

{
    "operationId": "23e5bdb6-b3a1-4d25-95aa-262365ecc6b9",
    "operationName": "Provision",
    "operationStatus": "RUNNING",
    "started": "2025-01-15T10:31:37+00:00"
}
```

Cloudera Data Hub status options

See a list of possible Cloudera Data Hub status options for the UI and CLI with explanations.

UI status	CLI status	Description
Provisioning		
 Requested	REQUESTED	A request to create or initiate the resource has been made and is in progress.

UI status	CLI status	Description
Create in progress...	CREATE_IN_PROGRESS	The resource creation process is currently underway.
Create failed	CREATE_FAILED	The resource creation process failed.
Update		
Update requested	UPDATE_REQUESTED	A request to update the resource has been initiated.
Update in progress...	UPDATE_IN_PROGRESS	An update to the resource is currently being applied.
Update failed	UPDATE_FAILED	The update process for the resource failed.
Disk update		
Disk update validation in progress...	DATAHUB_DISK_UPDATE_VALIDATION_IN_PROGRESS	Validation for a disk update on the Cloudera Data Hub is in progress.
Disk update validation failed	DATAHUB_DISK_UPDATE_VALIDATION_FAILED	The disk update validation for the Cloudera Data Hub failed.
Disk update failed	DATAHUB_DISK_UPDATE_FAILED	The disk update process for the Cloudera Data Hub failed.
Disk update resize failed	DATAHUB_DISK_UPDATE_RESIZE_FAILED	The disk resize process for the Cloudera Data Hub failed.
Recovery		
Recovery requested	RECOVERY_REQUESTED	A request to recover the resource has been initiated.
Recovery in progress...	RECOVERY_IN_PROGRESS	The resource is undergoing a recovery process.
Recovery failed	RECOVERY_FAILED	The recovery process for the resource failed.
Enable security		
Enable security failed	ENABLE_SECURITY_FAILED	The process to enable security features for the resource failed.
Delete		

UI status	CLI status	Description
Delete preparation in progress...	PRE_DELETE_IN_PROGRESS	Preparatory steps for deleting the resource are in progress.
Deleting...	DELETE_IN_PROGRESS	The resource is currently being deleted.
Deletion failed	DELETE_FAILED	The deletion process for the resource failed.
Deleted on provider	DELETED_ON_PROVIDER_SIDE	The resource was deleted directly on the cloud provider's side.
Deleted	DELETE_COMPLETED	The resource deletion process completed successfully.
Stop		
Stop in progress...	STOP_IN_PROGRESS	The resource is in the process of stopping.
Stop failed	STOP_FAILED	The resource failed to stop.
Stopped	STOPPED	The resource is fully stopped and not operational.
Start		
Start in progress...	START_IN_PROGRESS	The resource is in the process of starting.
Start failed	START_FAILED	The resource failed to start.
External database operations		
Database creation in progress...	EXTERNAL_DATABASE_CREATION_IN_PROGRESS	The external database for the resource is being created.
Database creation failed	EXTERNAL_DATABASE_CREATION_FAILED	The external database creation for the resource failed.
Database delete in progress...	EXTERNAL_DATABASE_DELETION_IN_PROGRESS	The external database for the resource is being deleted.

UI status	CLI status	Description
Database delete failed	EXTERNAL_DATABASE_DELETION_FAILED	The external database deletion for the resource failed.
Database delete finished	EXTERNAL_DATABASE_DELETION_FINISHED	The external database deletion for the resource completed.
Database start in progress...	EXTERNAL_DATABASE_START_IN_PROGRESS	The external database for the resource is starting up.
Database start failed	EXTERNAL_DATABASE_START_FAILED	The external database for the resource failed to start.
Database start finished	EXTERNAL_DATABASE_START_FINISHED	The external database for the resource has started successfully.
Database stop in progress...	EXTERNAL_DATABASE_STOP_IN_PROGRESS	The external database for the resource is being stopped.
Database stop failed	EXTERNAL_DATABASE_STOP_FAILED	The external database for the resource failed to stop.
Database upgrade in progress...	EXTERNAL_DATABASE_UPGRADE_IN_PROGRESS	The external database for the resource is being upgraded.
Database upgrade failed	EXTERNAL_DATABASE_UPGRADE_FAILED	The external database upgrade for the resource failed.
Database upgrade finished	EXTERNAL_DATABASE_UPGRADE_FINISHED	The external database upgrade for the resource completed successfully.
Load balancer update		
Load balancer update in progress...	LOAD_BALANCER_UPDATE_IN_PROGRESS	The load balancer for the resource is being updated.
Load balancer update failed	LOAD_BALANCER_UPDATE_FAILED	The load balancer update for the resource failed.
Load balancer update finished	LOAD_BALANCER_UPDATE_FINISHED	The load balancer update for the resource completed successfully.
Cluster Connectivity Manager upgrade		
CCM upgrade in progress...	UPGRADE_CCM_IN_PROGRESS	The Cluster Connectivity Manager for the resource is being upgraded.

UI status	CLI status	Description
CCM upgrade failed	UPGRADE_CCM_FAILED	The Cluster Connectivity Manager upgrade for the resource failed.
CCM upgrade finished	UPGRADE_CCM_FINISHED	The Cluster Connectivity Manager upgrade for the resource completed successfully.
Determining Data Lake size		
Determining data lake size...	DETERMINE_DATALAKE_DATA_SIZES_IN_PROGRESS	The process to determine data sizes for the associated Data Lake is underway.
Other statuses		
Running	AVAILABLE	The resource is fully operational and ready for use.
Wait for sync	WAIT_FOR_SYNC	The resource is waiting for synchronization with another system or component.
Maintenance mode	MAINTENANCE_MODE_ENABLED	The resource is in maintenance mode and unavailable for normal use.
Ambiguous	AMBIGUOUS	The resource's status is unclear or inconsistent.
Unreachable	UNREACHABLE	The resource cannot be contacted or accessed.
Node failure	NODE_FAILURE	One or more nodes in the resource have failed.
Stale	STALE	We have no information from the cluster and it has been unreachable for more than 30 days. The Cloudera Data Hub's status is outdated or no longer reflects its current state.

More cluster management options

You can access more cluster management options by logging in to Cloudera Manager.

For Cloudera Manager documentation, refer to the link below.

Related Information

Managing Cloudera Data Hub Clusters

Managing clusters from CLI

You can manage Cloudera Data Hub clusters from the CLI using the cdp datahub commands.

Required role: The DataHubAdmin or Owner roles at the scope of Cloudera Data Hub allow you to manage the Cloudera Data Hub cluster. Note that EnvironmentAdmin and Owner of the environment can also manage Cloudera Data Hub clusters.

Managing clusters

You can view and monitor your available Cloudera Data Hub clusters via the CDP CLI using the following commands:

- When stack provisioning or cluster creation fails, retry the cluster to resume the process from the last failed step: cdp datahub retry-cluster --cluster-name <value>
- Resize cluster: cdp datahub scale-cluster --cluster-name <value> --instance-group-name <value> --instance-group-desired-count <value>
- Restart a cluster that is in stopped state: cdp datahub start-cluster --cluster-name <value>
- Stop cluster that is in running state: cdp datahub stop-cluster --cluster-name <value>
- Sync your cluster with cloud provider and Cloudera Manager: cdp datahub sync-cluster --cluster-name <value>
- Repair a cluster when cluster has failed nodes: cdp datahub repair-cluster --cluster-name <value> --instance-group-names <value> --instances <value>

Deleting clusters

- Delete one or more existing clusters: cdp datahub delete-clusters --cluster-name <value>