Managing Clusters

Date published: 2019-12-17
Date modified:

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2023. All rights reserved.

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

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

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

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

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

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

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

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

Contents

Managing clusters 5

Retry a cluster 5

Resize a cluster 5

Stop a cluster 6

Restart a cluster 8

Stopping and restarting cluster services 9

Performing manual repair 9

Terminate cluster 12

Terminate cluster nodes 12

Force terminate cluster 13

Deleting clusters when termination fails 13

Creating a multi-AZ Data Hub on AWS 14

Placement group support 15

Vertically scaling instance types 16

Upgrading Data Hubs 17

Support matrix for Data Hub upgrades 21
Backing up before an upgrade 21
Performing a Data Hub major/minor version upgrade 25
Post-upgrade tasks 28
Adding configs manually 28

Upgrading Cloudera Operational Database clusters 29
Performing a Data Hub maintenance upgrade 30
Performing a Data Hub OS upgrade 31
Upgrading a Data Hub with the CDP CLI 34
Troubleshooting upgrade 35

Autoscaling clusters 36
Autoscaling behavior 37
Configuring autoscaling 38
Using YARN queues with autoscaling 42
Configuring multiple YARN queues for autoscaling 42
Manually recovering from load-based scaling failures 47
Autoscaling FAQ 49

Upgrading Data Lake/Data Hub database 49
Database upgrade known limitations and troubleshooting 57
Installing Postgres 11 packages manually 58

Managing certificates 59
Renewing private/host certificates on Data Lake and Data Hub clusters 60
Manually renewing public certificates for Data Lake and Data Hub clusters 62

More cluster management options 62

Managing clusters from CLI 63

Managing clusters

You can manage your Data Hub clusters from the Data Hub cluster dashboard in the 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:

Retry 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 the Data Hub allow you to manage the Data Hub. Note that EnvironmentAdmin and Owner of the environment can also manage Data Hubs.

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.

Procedure

1. Log in to the CDP web interface.
2. Navigate to the 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.

Resize 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 the Data Hub allow you to manage the Data Hub. Note that EnvironmentAdmin and Owner of the environment can also manage Data Hubs.

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 CDP web interface.
2. Navigate to the 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

Stop a cluster

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 the Data Hub allow you to manage the Data Hub. Note that EnvironmentAdmin and Owner of the environment can also manage Data Hubs.

Note: Most clusters that use ephemeral volumes cannot be stopped via 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 CDP web interface.
2. Navigate to the 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

Restart a cluster

Restart 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 the Data Hub allow you to manage the Data Hub. Note that EnvironmentAdmin and Owner of the environment can also manage Data Hubs.

Note: Most clusters that use ephemeral volumes cannot be stopped/restarted via 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 CDP web interface.
2. Navigate to the 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
Stop 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 the Data Hub allow you to manage the Data Hub. Note that EnvironmentAdmin and Owner of the environment can also manage Data Hubs.

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 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, CDP 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.<p>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 <filepath>/hadoopfs/fsN/</filepath> (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 <filepath>/hadoopfs/fs1/save_me</filepath> 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.

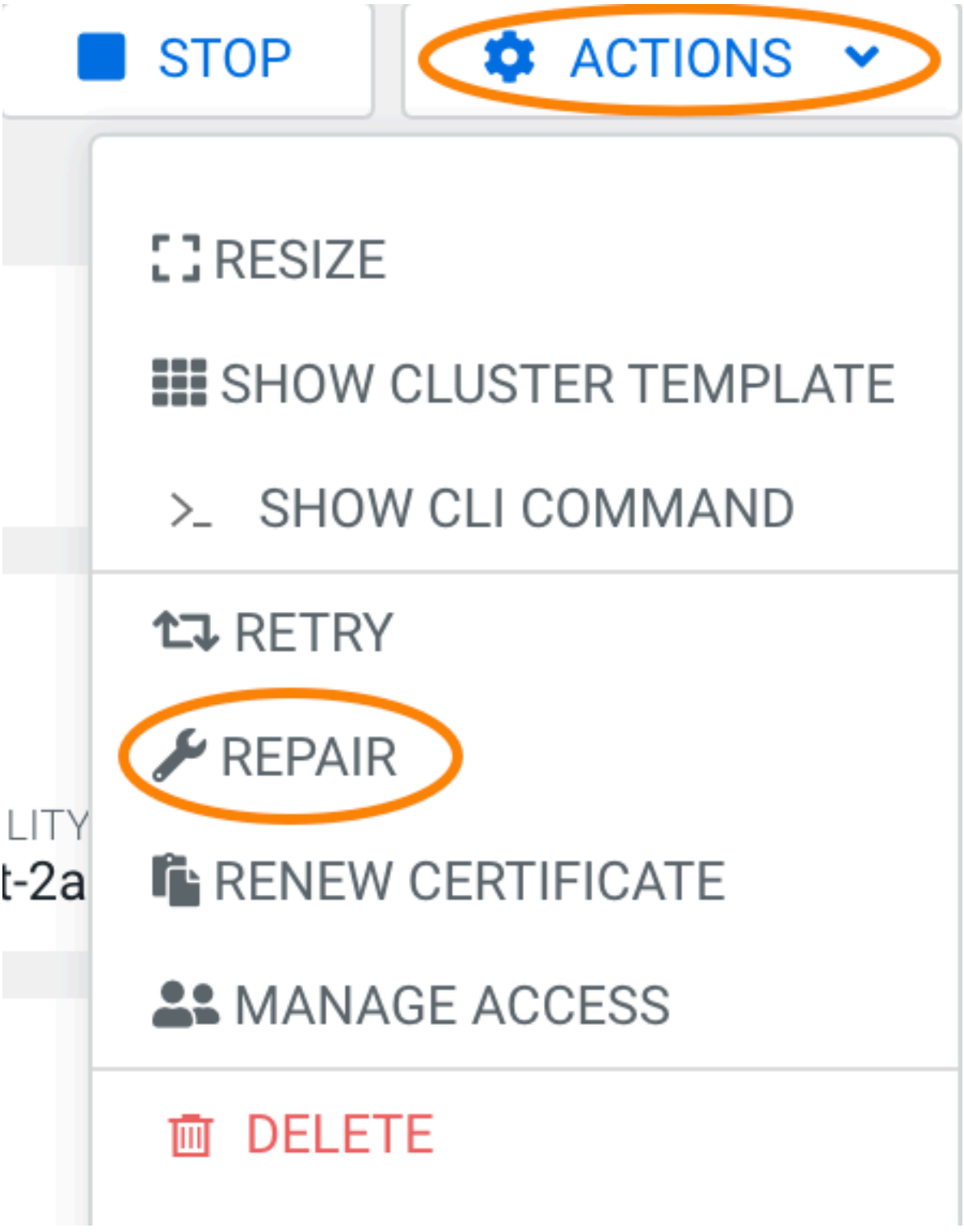
If a node is marked as deleted from the cloud provider side, you must perform manual repair before restarting the cluster. If you do not perform the manual repair, the cluster will not restart.

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

To perform manual repair from CDP web interface:

1. Log in to the CDP web interface.
2. Navigate to 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

Terminate cluster

You can terminate any cluster managed by Data Hub.

About this task

Required role: Owner at the scope of the 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 CDP web interface.
2. Navigate to the 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 terminate cluster

Terminate cluster nodes

Terminate cluster nodes

Rather than terminating an entire 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 Data Hub (which provides delete access).

Procedure

1. In the CDP 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 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 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 terminate cluster

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

Before you begin

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

Cluster deletion may fail if 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 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 CDP web interface.
2. Navigate to the 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, 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 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 Management Console > Data Hub Clusters > cluster details > Hardware section.

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

  On AWS and GCP, 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 Data Hub on AWS

By default, CDP provisions Data Hubs 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 CDP, refer to Deploying CDP in multiple AWS availability zones.

Create a multi-AZ Data Hub

You can create a multi-AZ Data Hub via CDP 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 Data Hub as usual. In the Advanced Options > Network and Availability, you select the multiple subnets across which the Data Hub 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 Data Hub, 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, 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 --instance-groups option.

Scaling a multi-AZ Data Hub

If there is an availability zone that is offline, CDP 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, CDP 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 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.

Placement group support

You can configure a Data Hub cluster on AWS for placement group support. Placement groups support placing VM instances across different physical hardware within a single availability zone. Placement groups can help ensure cluster availability in the event of a physical hardware failure within an availability zone.

In a Data Hub cluster, each host group can be associated with a placement group, of which there are three supported types: partition, spread, or cluster. Cloudera recommends using the partition type as the default for all host groups. For a partition placement group, the partition count will always be 2 and is not configurable.

Note: Only specific host groups, such as the master host group of the Data Engineering HA template, can be configured with the spread placement group. The spread placement group has a maximum instance limit of seven instances per placement group, per availability zone. Configuring multiple spread placement groups does not guarantee hardware isolation across the placement group.

Configuring placement group support requires adding custom properties to a cluster definition at the host group level. You can define the strategy as “NONE,” “PARTITION,” “SPREAD,” or “CLUSTER.”

For example:

```
{
  "environmentName": "sample-env",
  "instanceGroups": [
```

```
    {
      "nodeCount": 1,
      "name": "master",
      "type": "GATEWAY",
      "recoveryMode": "MANUAL",
      "template": {
        "aws": {
          "encryption": {
            "type": "NONE",
            "key": null
          },
          "placementGroup": {
            "strategy": "PARTITION"
          }
        },
        "instanceType": "m5.2xlarge",
        "rootVolume": {
          "size": 50
        },
        "attachedVolumes": [
          {
            "size": 100,
            "count": 1,
            "type": "standard"
          }
        ],
        "cloudPlatform": "AWS"
      },
```

Placement groups have a number of rules and limitations. Importantly, it is possible to run out of placement groups within an availability zone. Refer to the AWS documentation for detailed rules and limitations.

Vertically scaling instance types

If necessary, you can select a larger or smaller instance type for a Data Hub or Data Lake cluster after it has been deployed.

Before you begin

You must stop the Data Lake or Data Hub cluster before you vertically scale any of the instances.

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.

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.

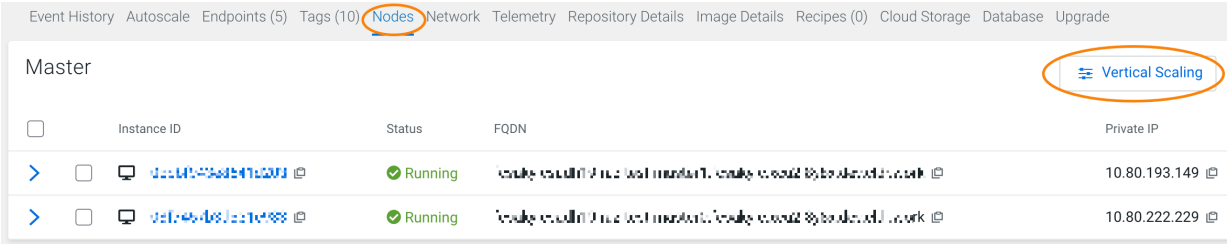
Vertical scaling is supported on AWS only.

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

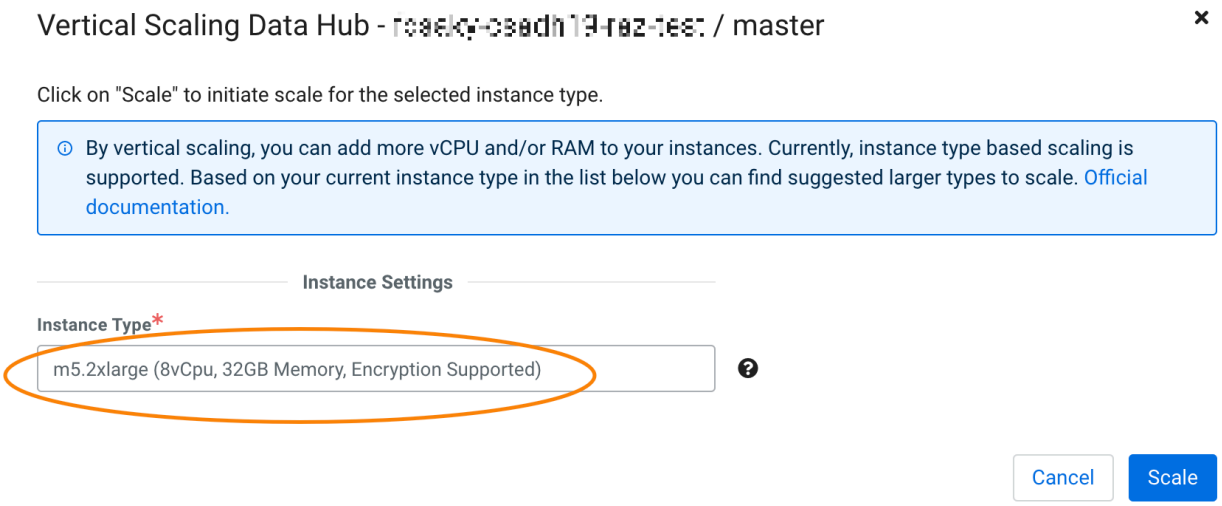
Procedure

1. In the CDP main navigation menu, click Data Hubs 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 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>"
```

   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

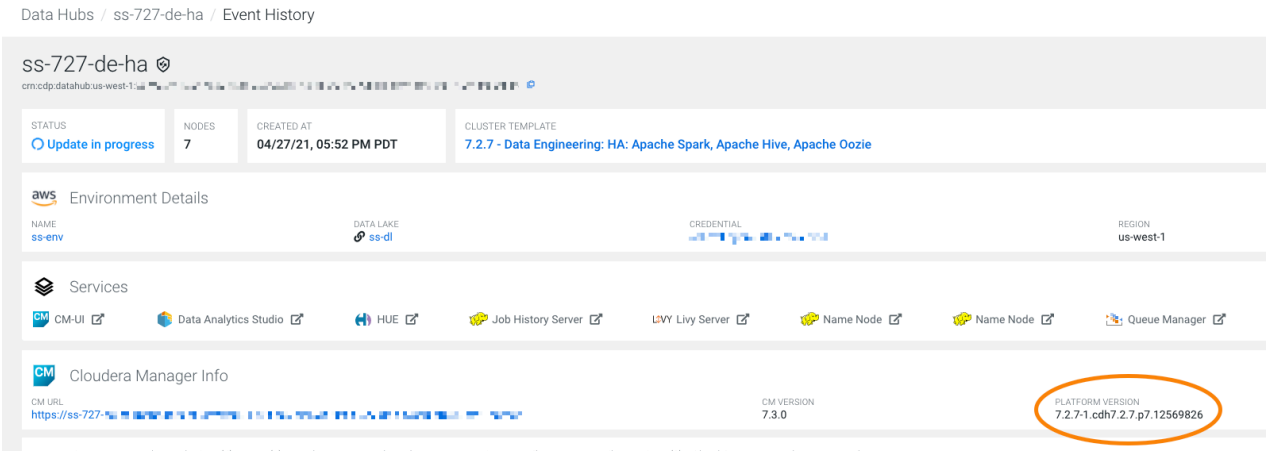
After you have vertically scaled the cluster, configure the services on the cluster to use the additional or reduced resources/memory.

Upgrading Data Hubs

You can upgrade a Data Hub cluster in one of two ways: Runtime and Cloudera Manager major/minor version upgrades and maintenance/“hotfix” upgrades.

Cloudera Runtime and Cloudera Manager Major/Minor Version Upgrades

A CDP environment consists of both Data Lake and 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 Runtime major/minor version is 7.2.7.

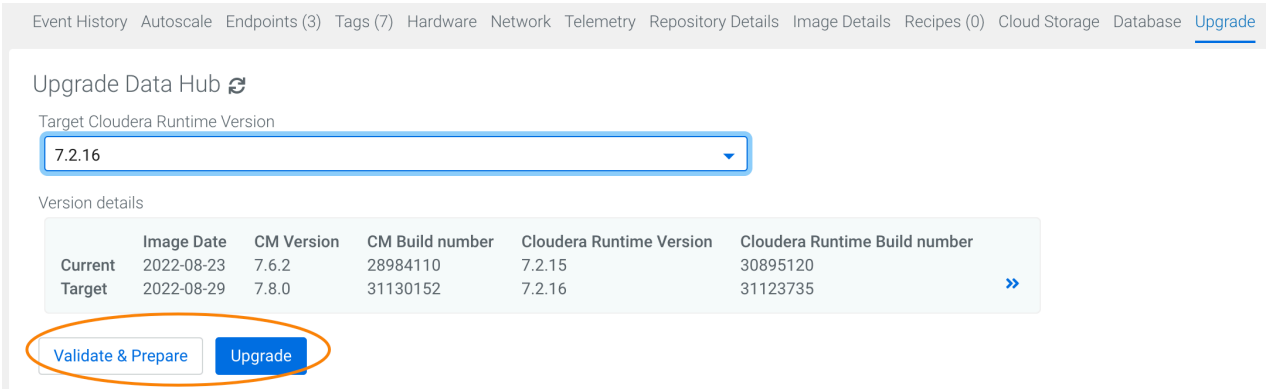
A Data Hub major/minor version upgrade initiates an upgrade of the major/minor Cloudera Manager and Runtime versions, as well as the required additional parcels (Spark3, Flink, Profiler, and Cloudera Flow Management). The root volumes, additional volumes, and the cluster state is retained. A major/minor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

A major/minor version upgrade is not an OS upgrade, as the OS/VM packages are not updated and VMs are not replaced. You can initiate an OS upgrade in a separate step. Major/minor upgrades do not update other CDP packages such as Salt.

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 CM 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.

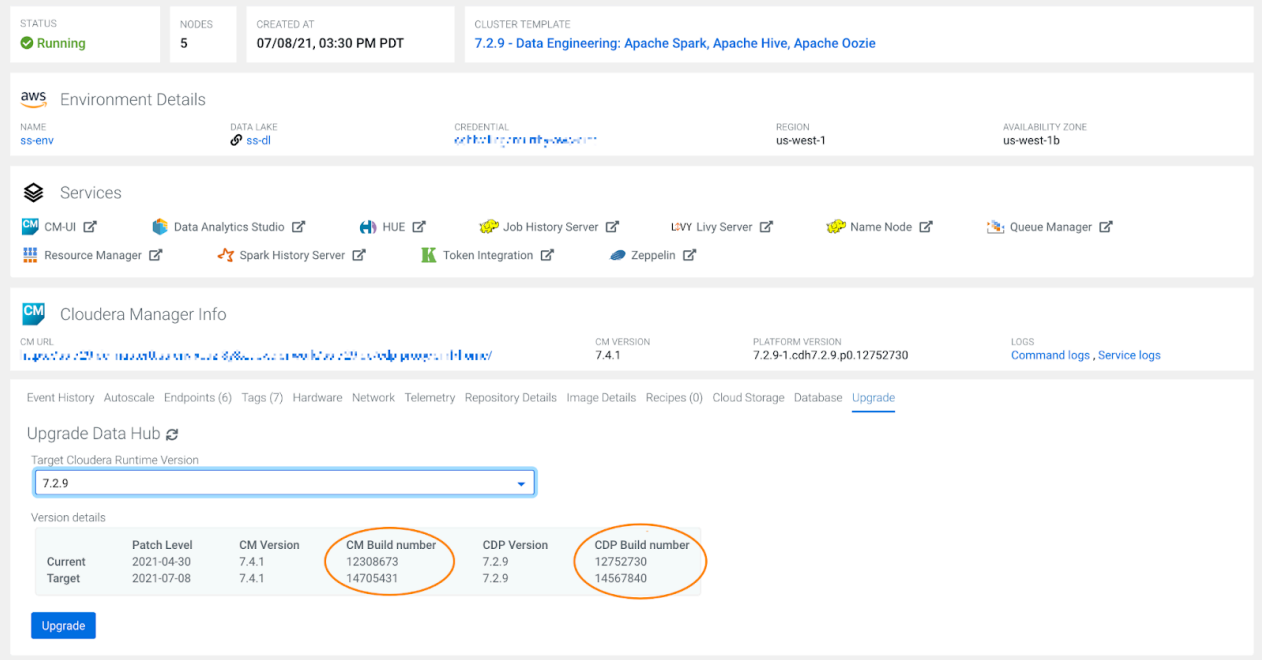
Always test any cluster upgrade in a development environment first.

Maintenance upgrades

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

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

When a maintenance 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 CM/CDP versions are the same, but the build numbers differ:



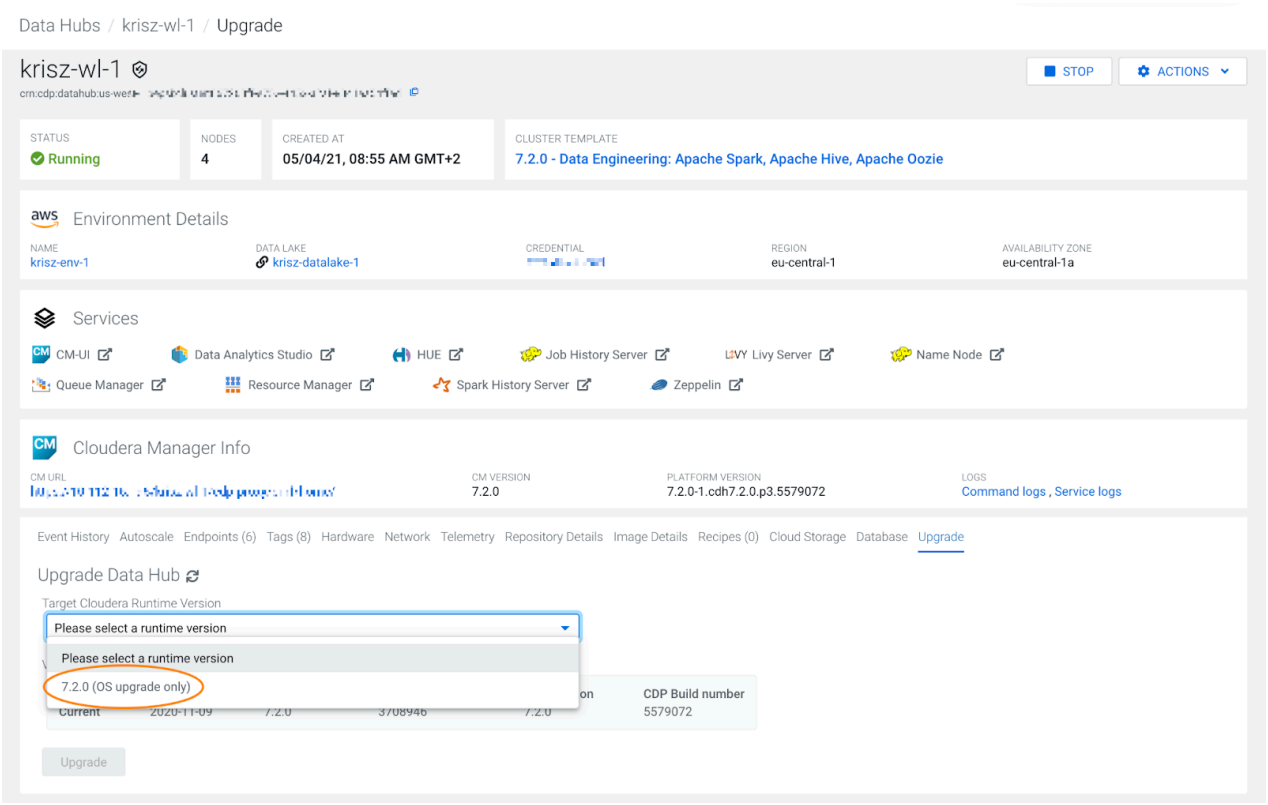
Maintenance upgrades are available from Runtime 7.2.7 onward for RAZ and non-RAZ Data Hubs.

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 (CM, Runtime, and additional parcels). You can perform an OS upgrade independent of a Data Hub upgrade.

If an OS upgrade is available, it will appear in the Upgrade Data Hub menu with "(OS Upgrade Only)" when you select a Runtime version:



Limitations

- This is not a zero-downtime upgrade and service outages will occur.
- Ranger Authorization: Data Hub maintenance upgrades with RAZ are supported only for Runtime versions 7.2.7+. Major/minor version upgrades with RAZ are supported only for 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.
- Cloudera Operational Database cannot be upgraded through the Data Hub user interface and must be upgraded through the CDP beta CLI. For more information see *Upgrading Cloudera Operational Database*.

Prerequisites

- There is required downtime of the environment during upgrades, so plan the upgrade accordingly.
- Verify using the *Support matrix for Data Hub upgrades* that all 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 Runtime/CM upgrade to ensure application compatibility with the new platform version.
- The Data Lake and all the Data Hubs in an environment must be upgraded to the same major/minor version. If you plan to upgrade your Data Hub clusters to a later major/minor version, you must first backup and then upgrade the Data Lake to that version.
- Verify that any Experiences you use are running the latest version available.

Related Information

Data Lake upgrade

Support matrix for Data Hub upgrades

Upgrading Cloudera Operational Database clusters

Support matrix for Data Hub upgrades

Data Hub upgrades are supported only for the templates and 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 above
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
Data Catalog	Not supported. Use backup and restore	
COD	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 Runtime versions 7.2.10-7.2.12 to 7.2.14+.

Maintenance Upgrades

Data Hub maintenance upgrades, both RAZ and non-RAZ, are supported only for 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 CM 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 Data Hub upgrade is an in-place upgrade and the data on attached volumes will be retained. The following section outlines data stored in the 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 CM.

Data Engineering clusters

The underlying data is in cloud storage. However, the following data is specific to each 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. 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

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
```

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 Data Platform 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 Data Platform 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

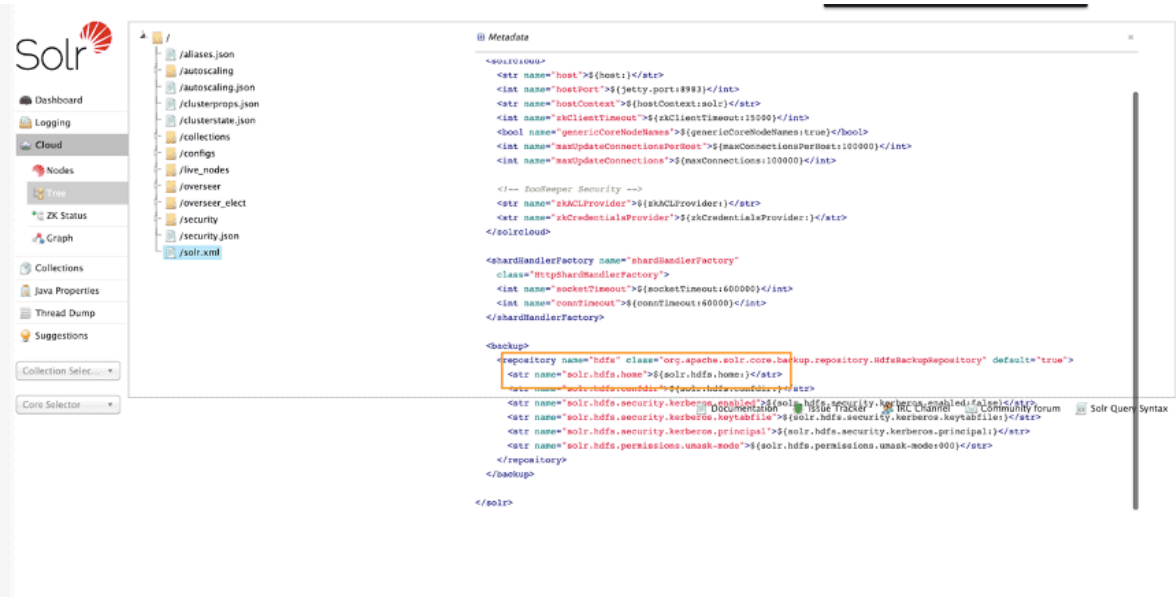
Complete the following steps to backup DDE clusters:

- Assign the datalake-admin-bucket policy to RANGER_AUDIT_ROLE.
- Create a subfolder for this Data Hub inside your S3 backup location.
- Log in to Data Hub gateway node.
- kinit with the proper user keytab if needed.
- Follow this backup guide, 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 Data Hub inside of your S3 backup location.
- Ensure snapshots are enabled on Cloudera Manager. They are enabled by default.
- Log in to Data Hub gateway node.
- 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 Data Hub major/minor version upgrade

A major/minor version upgrade initiates an upgrade of the major/minor Cloudera Manager and 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, perform a major/minor version upgrade on each Data Hub attached to the Data Lake.

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 Data Hub cluster that you are upgrading.

Required role: DatahubAdmin or Owner over the Data Hub

Procedure

1. Start the cluster.
2. Before you begin the Data Hub 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>

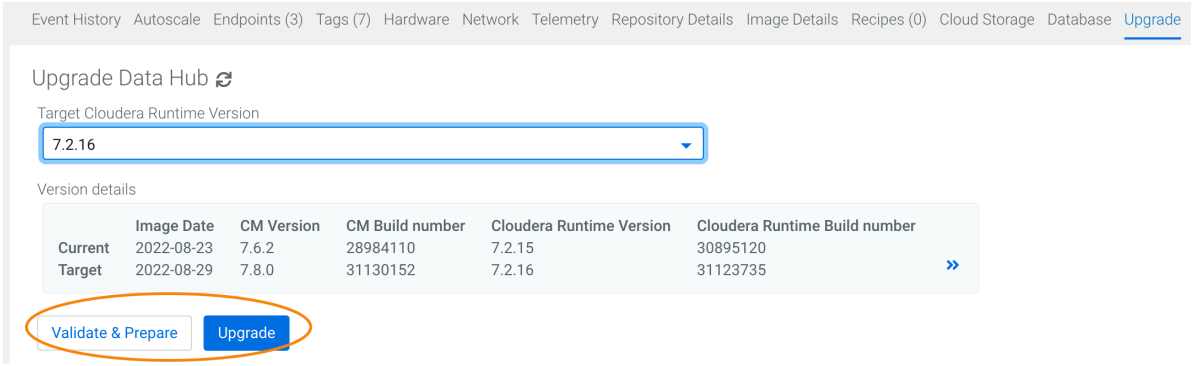
      ```
      yarn/nightly-7x-1-1.nightly-7x-1.root.hwx.site@ROOT.HWX.SITE
      ```

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

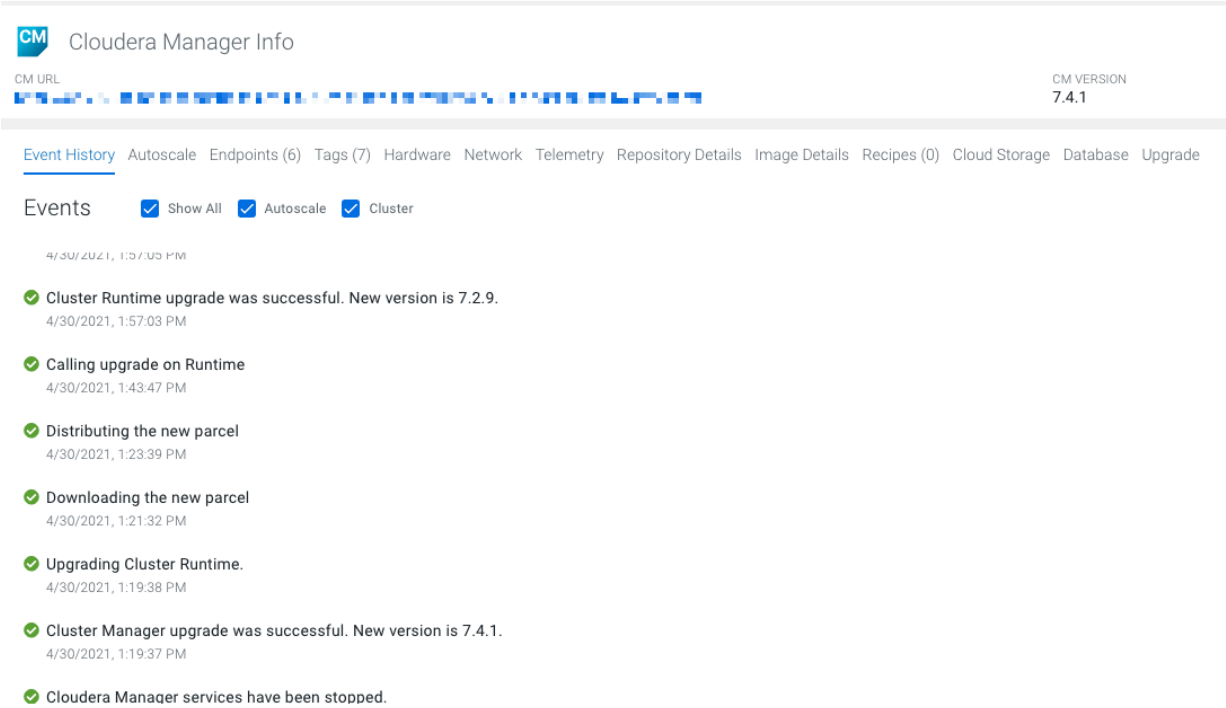
      ```
      kinit -kt yarn.keytab yarn/nightly-7x-1-1.nightly-7x-1.root.hwx.site@ROOT.HWX.SITE
      ```

   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

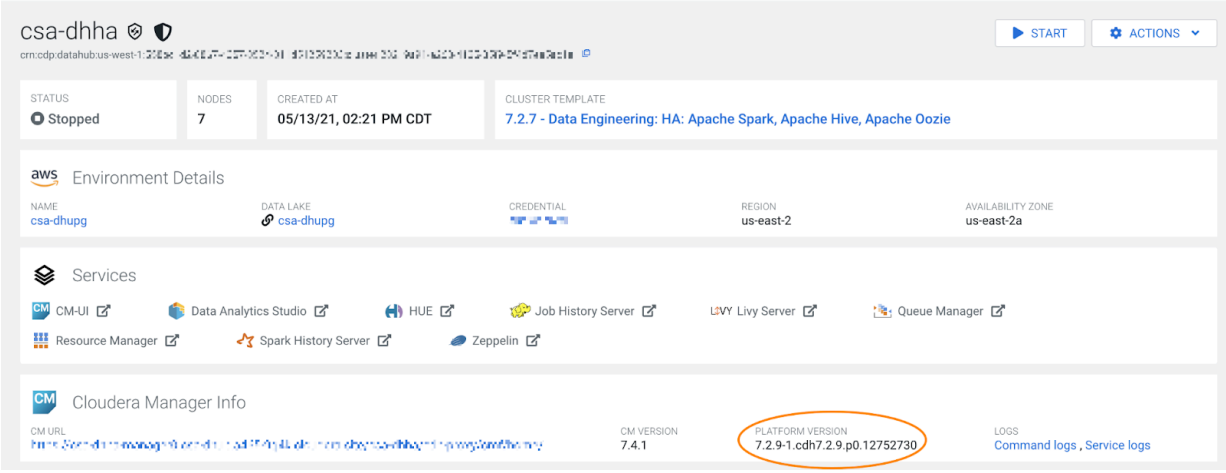
4. For Streams Messaging clusters, if you are upgrading from Runtime version 7.2.12 to 7.2.14, complete the following steps:
   a) Open the CM UI for the Streams Messaging cluster.
   b) Click on the Cruise Control service, then click on Configurations.
   c) Search for 'RackAwareGoal' in the search 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 you use autoscaling, disable autoscaling on the cluster.
6. Upgrade the cluster. To upgrade the cluster with the UI:
   a) In the left-hand menu, click Data Hubs and then select the Data Hub 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) 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.
   e) When the validate and prepare process is complete, click Upgrade.



7. Monitor the upgrade progress using the Data Hub **Event History** tab:



8. 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):



9. 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 Data Hub upgrade process, you must repeat the upgrade procedure for every Data Hub 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 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 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

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.

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 CM 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 CDP Beta CLI to run the upgrade-database command.

For more information on CDP CLI beta commands, see the *CDP CLI Beta command reference*.

Before you perform the COD upgrade: 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:

This 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

Performing a Data Hub maintenance upgrade

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

About this task

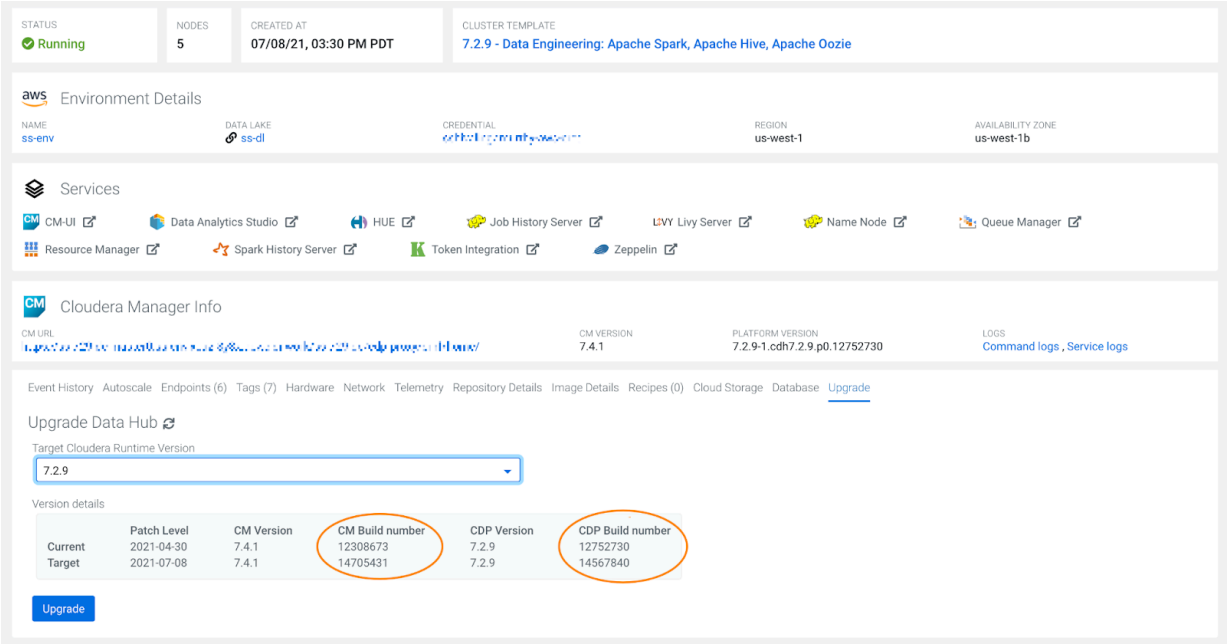
Complete the steps for each Data Hub cluster that you are upgrading.

Required role: DatahubAdmin or Owner over the Data Hub

Procedure

1. In the left-hand menu, click Data Hubs and then select the Data Hub 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 Rutime 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 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 maintenance upgrade is complete, re-install the Oozie shared libraries and YARN MapReduce Framework JARs.
   a) Navigate to the cluster's Oozie service in CM and click the Status tab.
   b) Click ActionsInstall Oozie ShareLib.
   c) Navigate to the cluster's YARN service in CM and click the Status tab.
   d) Click ActionsInstall YARN MapReduce Framework JARs.

Related Information

Troubleshooting upgrade

Performing a Data Hub OS upgrade

Perform a 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 Data Hub

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 CDP 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 CDP 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 CDP 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:

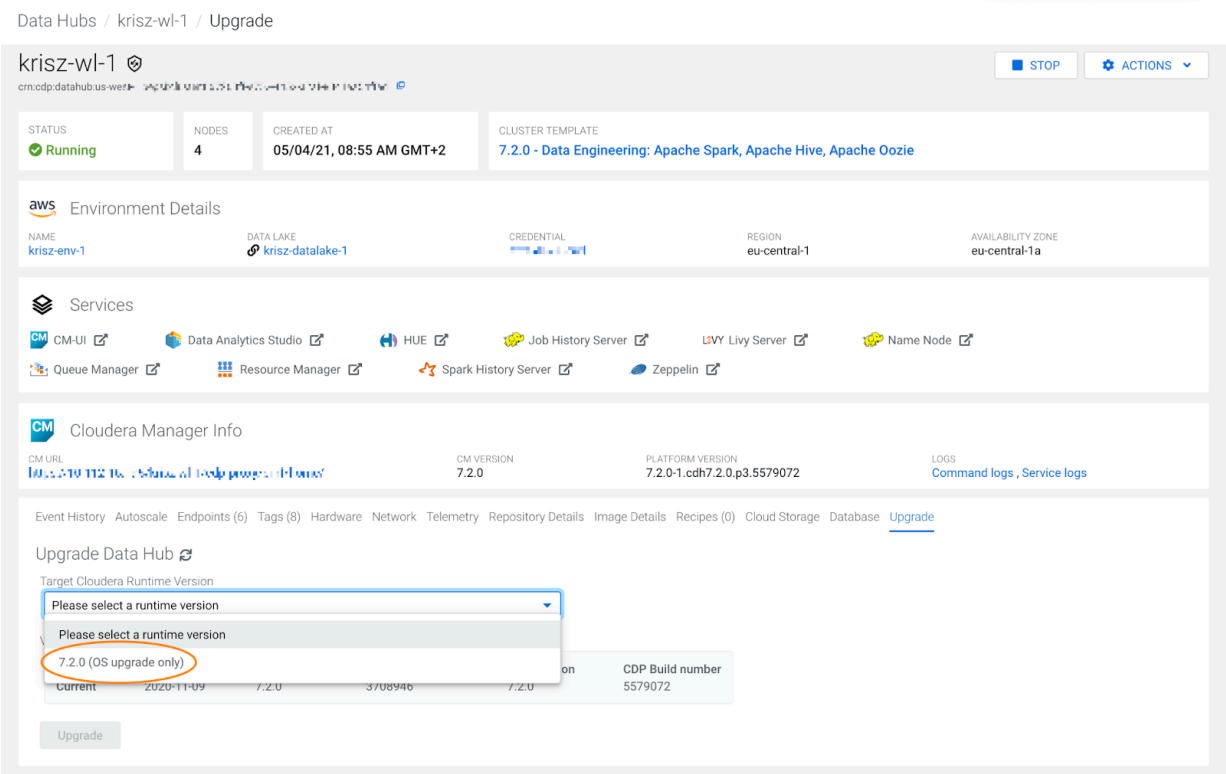
1. Open the Cloudera Manager UI.
2. Go to the NiFi service.
3. Select the Configuration tab.
4. Search for nifi.registry.working.directory.
5. Set the new value and click Save.

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 CDP, click Data Hubs and then select the Data Hub 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 Upgrade only” next to the target version.



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

What to do next

If you upgraded from CDP 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.

Upgrading a Data Hub with the CDP CLI

You can initiate a Data Hub 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.

The cdp datahub upgrade-cluster command has 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]
```

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

Table 1:

Option	Description
--cluster-name (string)	Required. The name or CRN of the Data Hub to upgrade.
--image-id (string)	The ID of an image to upgrade to.
–runtime (string)	The Runtime version to upgrade to. When you specify the Runtime version, the upgrade uses the latest image ID of the given Runtime version from the same image catalog used for Data Hub 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, 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 Runtime/CM upgrade to the latest compatible version and OS image.

Outside of upgrade, you can use the following options:

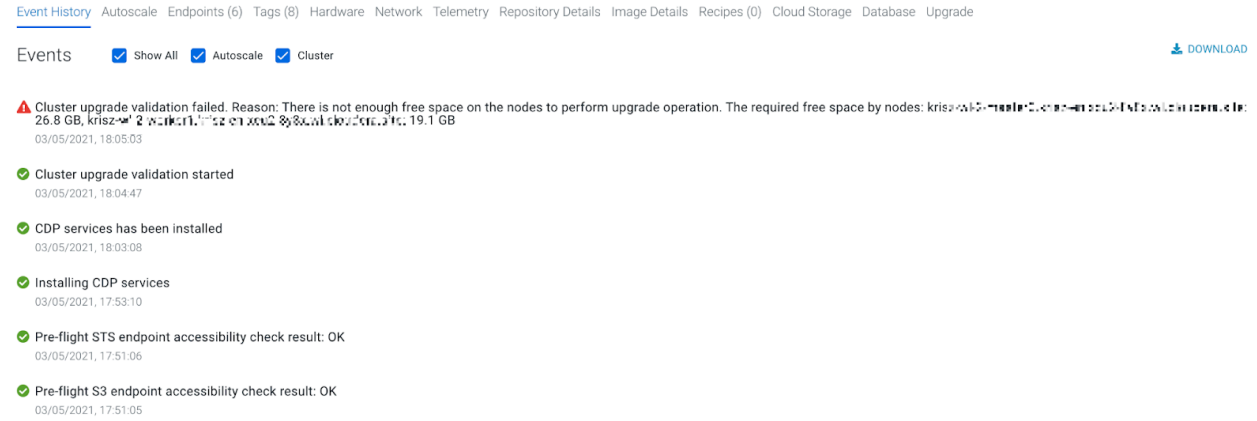
```
--show-available-images/--no-show-available-images
--show-available-images-per-runtime/--no-show-available-images-per-runtime
 --dry-run
```

Troubleshooting upgrade

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

Root disk space

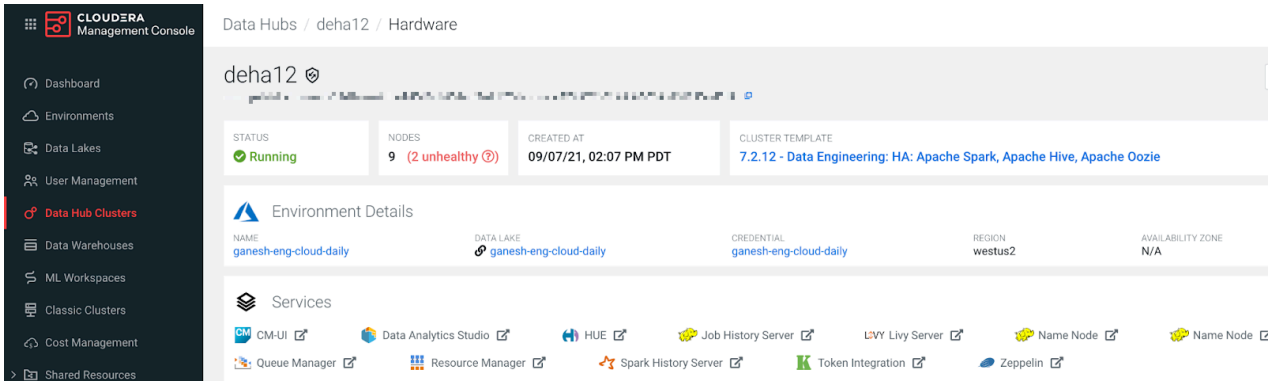
Maintenance upgrades require a certain amount of root disk space. Runtime upgrades require downloading additional parcels, and hence more storage. When you begin an upgrade, CDP checks the available root disk space and issues a warning if there is insufficient root disk space for the upgrade:



If your instances do not have the free space required for an upgrade (CM: 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 maintenance upgrade section to re-install the Oozie shared libraries and YARN MapReduce Framework JARs.

Upgrade fails due to active CM commands

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

CM, Runtime, or other components are out-of-sync with CDP

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

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

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

Related Information

Performing a Data Hub maintenance 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 only for clusters provisioned in AWS, with the following Runtime versions:

- 7.2.15 and higher

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 provisioned in AWS with the following 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.

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 only for host groups with nodes running YARN NodeManager (and optionally client/GATEWAY 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.

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).

Cluster management operations with load-based autoscaling enabled

You may receive a warning when you try to perform certain administrative operations on a cluster with a load-based autoscaling policy enabled. When a cluster is running with stopped instances (as a result of a downscale), operations like Data Hub upgrades or certificate rotation are not allowed. To perform these operations, make sure that the cluster is in a state where there are 0 stopped instances. You can do this by starting all instances that are in a stopped state (temporarily disable autoscaling, then use the beta CLI command cdp datahub start-instances --cluster CLUSTER --host-group HOST_GROUP [--instance-count INSTANCE_COUNT]). Alternatively you can delete all instances in a stopped state. Specific operations are affected as follows:

- Node repair: If there are STOPPED nodes in a cluster, and some other nodes need to be repaired - the operation can be performed by including all of the stopped nodes in the set of nodes to be repaired.
- 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.

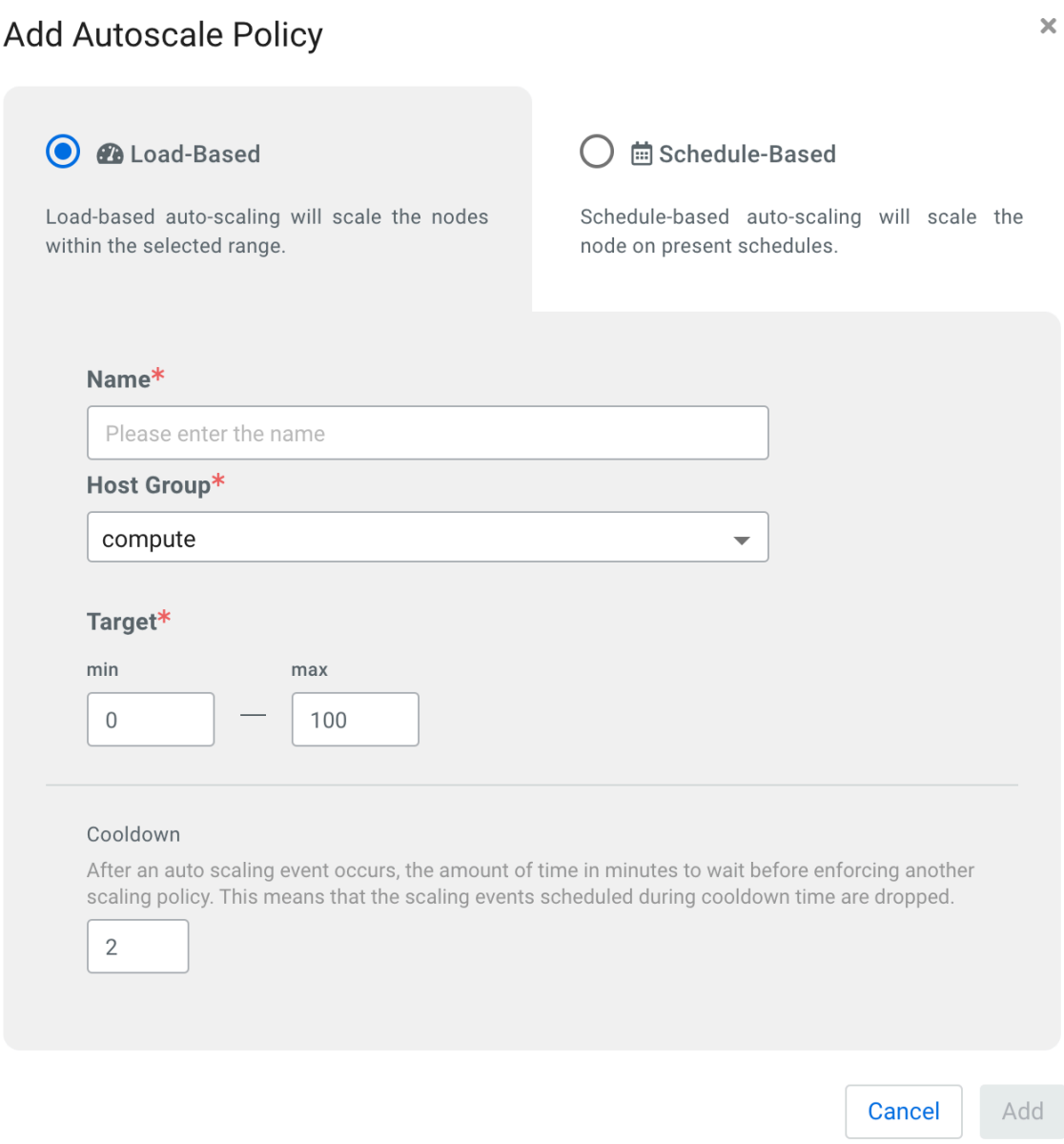
Before you begin

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 CDP 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.

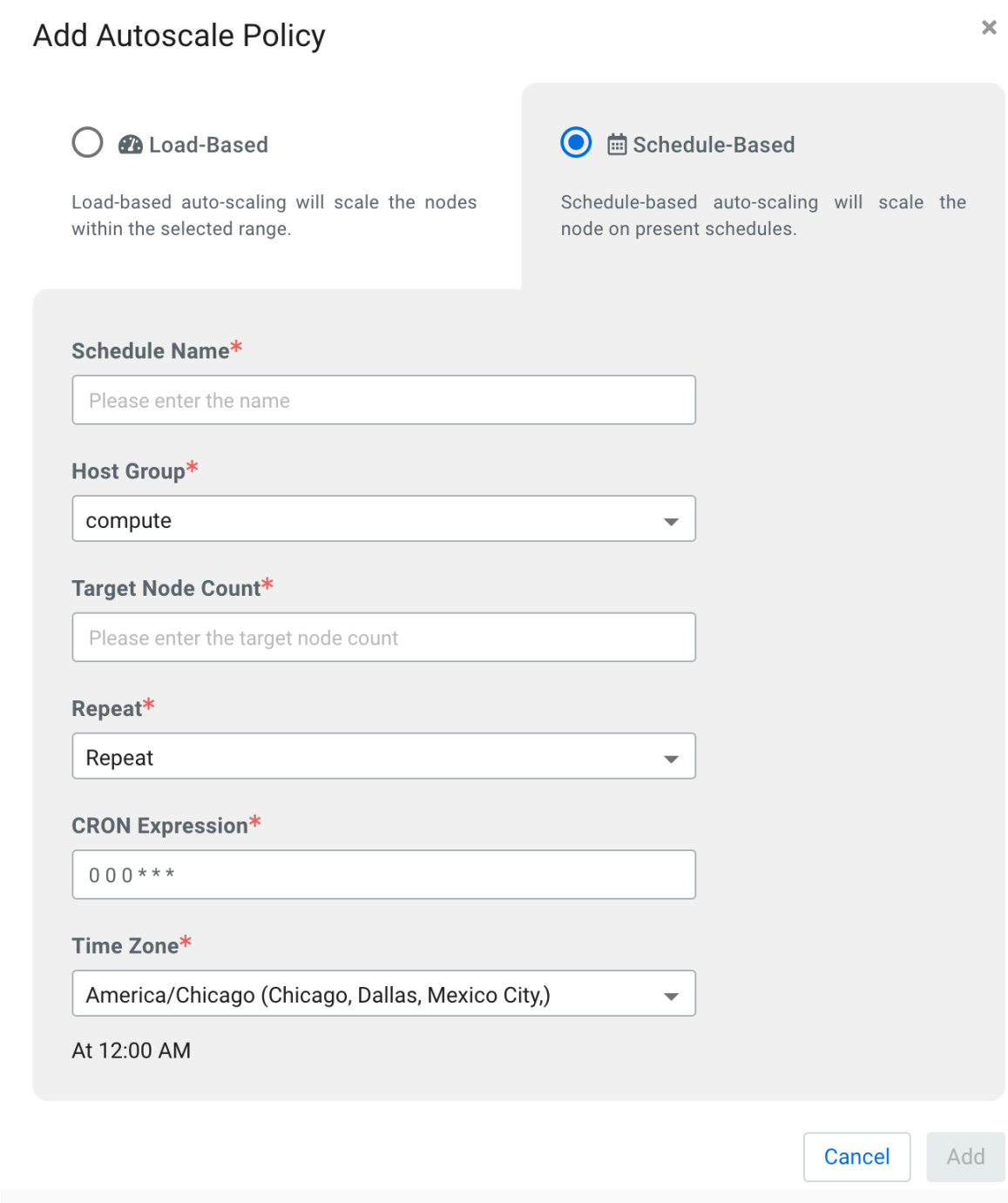
4. Define the policy parameters.
   - For load-based scaling:



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

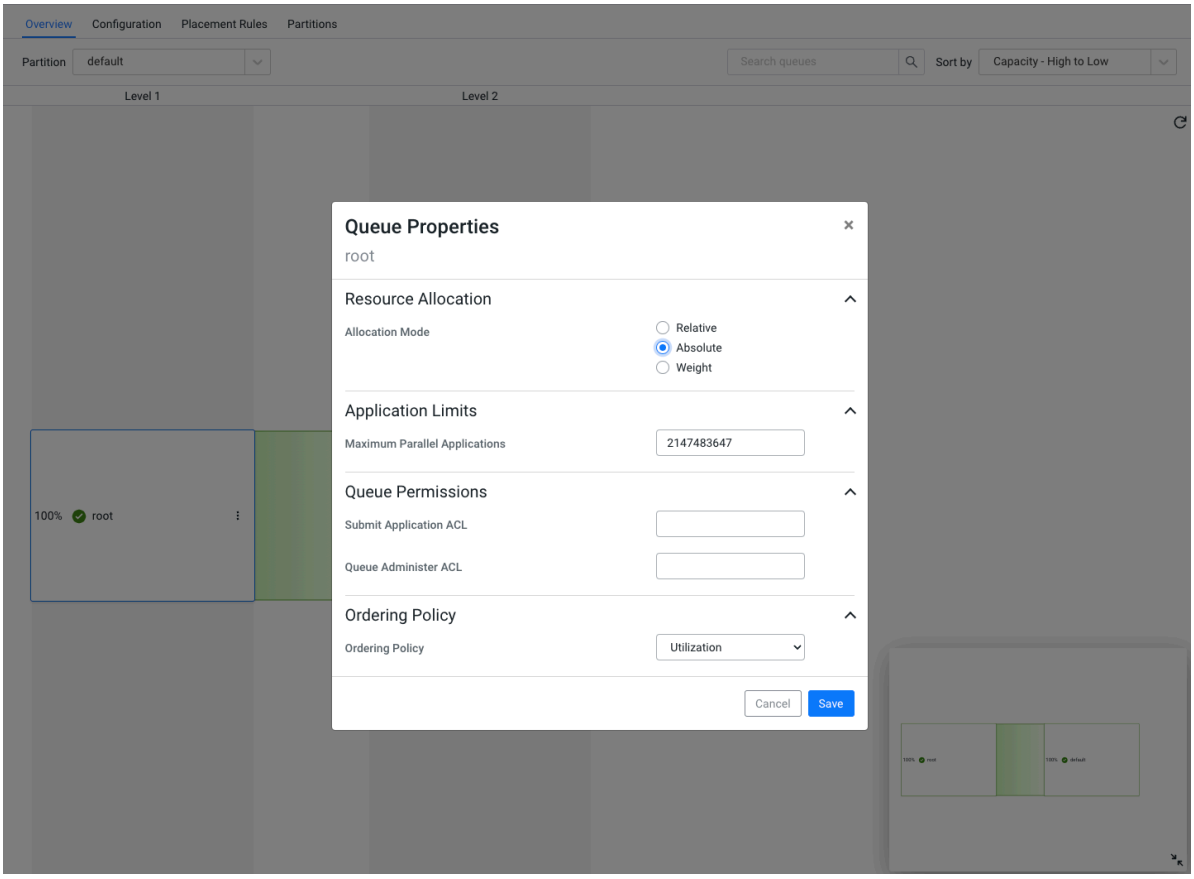
1. Set the Resource Manager plugin property to *fine* using Cloudera Manager:
   a) In Cloudera Manager, select the YARN service.
   b) Go to the **Configuration** tab.
   c) Search for *plugin-type* and set the below property in the ResourceManager Advanced Configuration Snippet (Safety Valve) for yarn-site.xml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

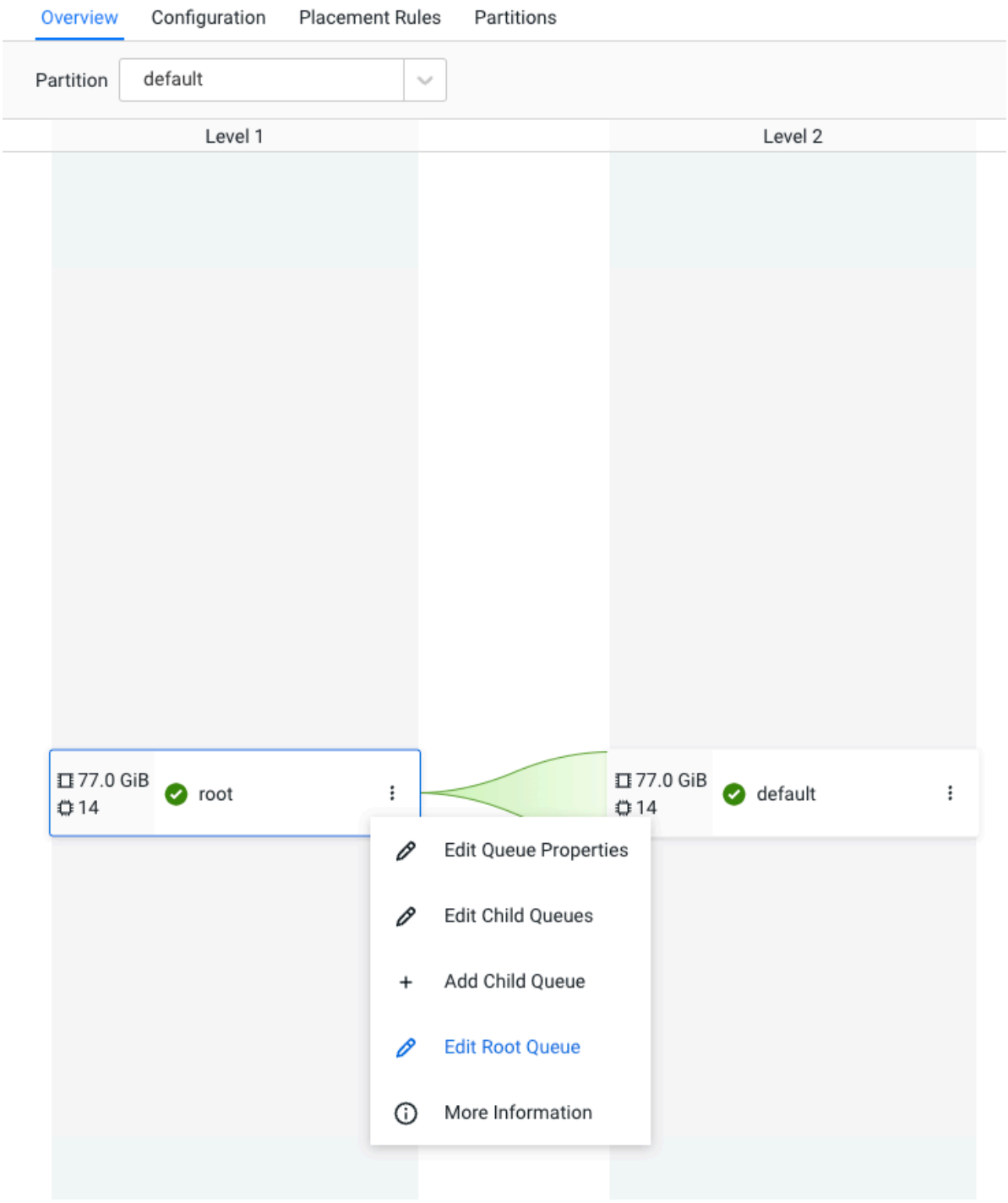
   Important: If all of the queues in the system are allowed to use the maximum available resources in the cluster, 'percentage' mode can be used, by setting the max-capacity for each queue to 100%. If you are using percentage mode in this manner, you can skip the remainder of this task.

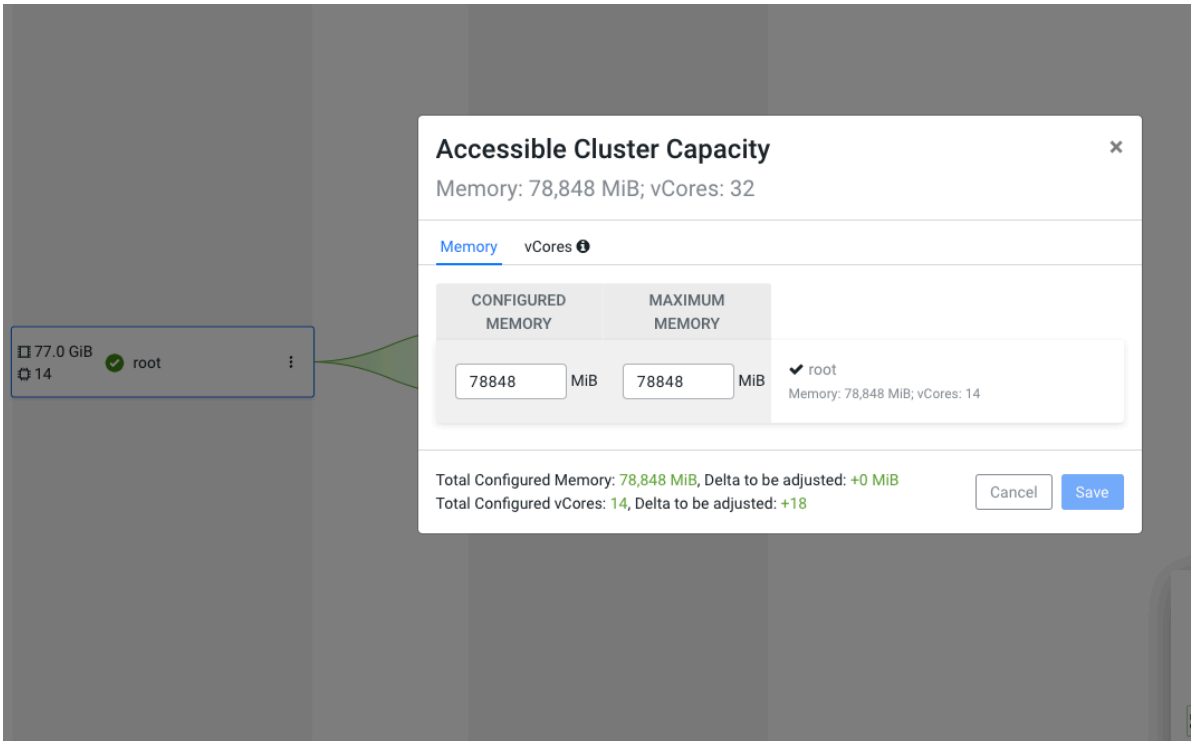
2. Ensure that the queue configuration uses absolute values instead of percentages:
   a) In the CDP Management Console, navigate to the specific Data Hub and scroll to the **Services** section. Click Queue Manager.
   b) Click on the three vertical dots on the root and select Edit Queue Properties option.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



3. Disable autoscaling on the cluster.
4. Scale-up the autoscaling hostGroup manually to the new ‘Target max’ number of nodes.

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.

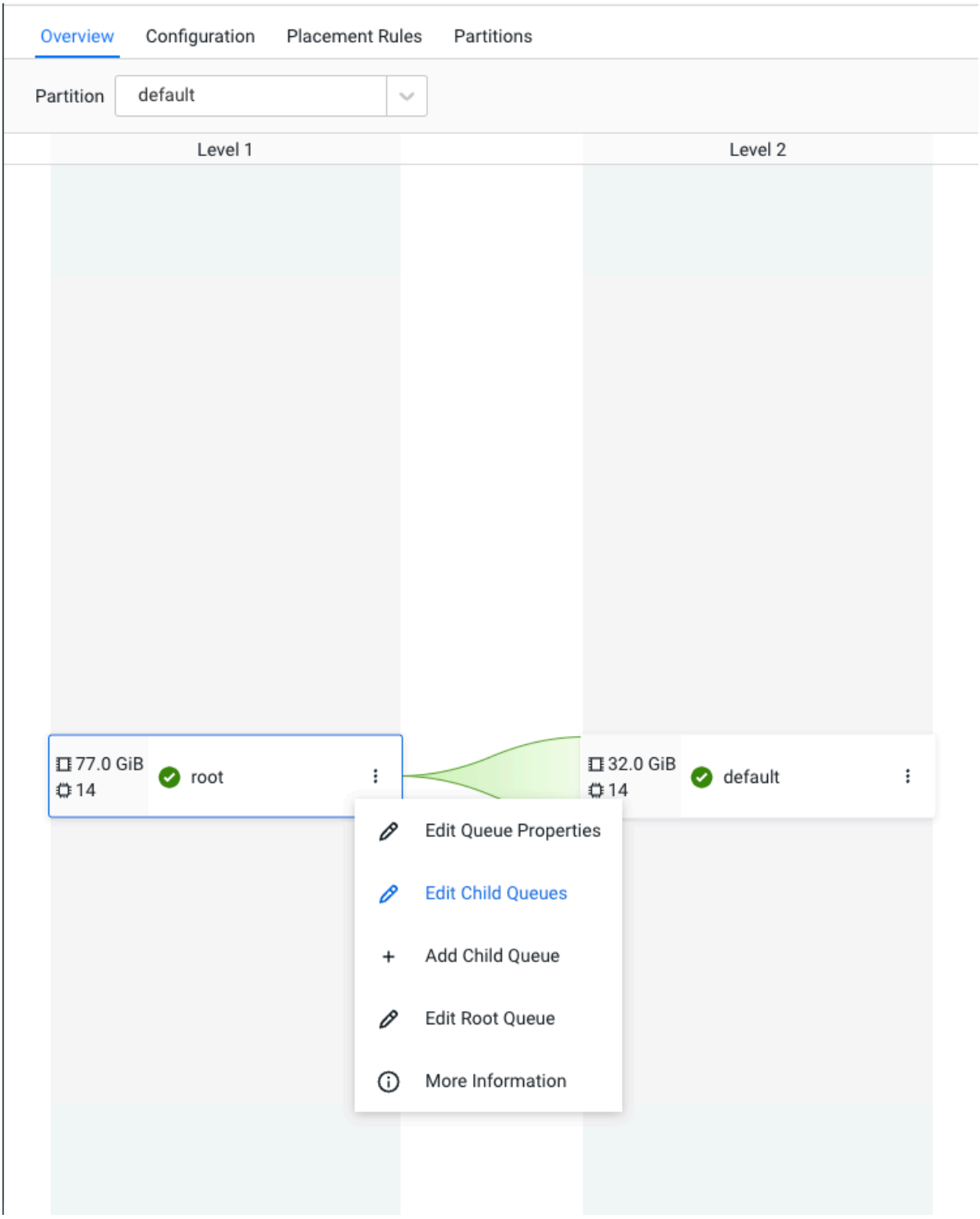


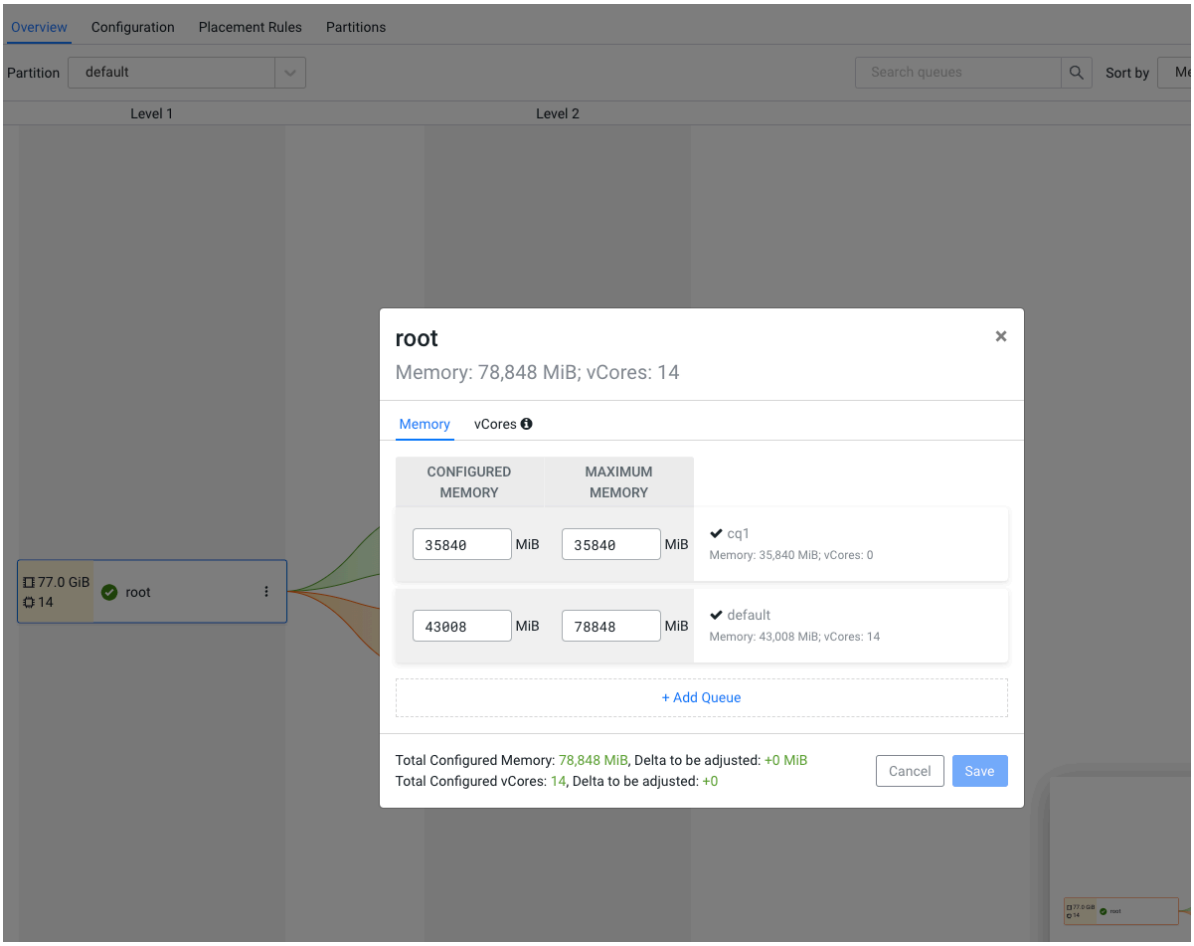


b) Edit the Child Queues as needed, and configure the resource allocation. In the example below, a new queue called `cq1` is added, which has a configured memory allocation, and maximum memory allocation of 35GB. The `default` queue has a configured memory allocation of 42GB, and a maximum memory allocation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

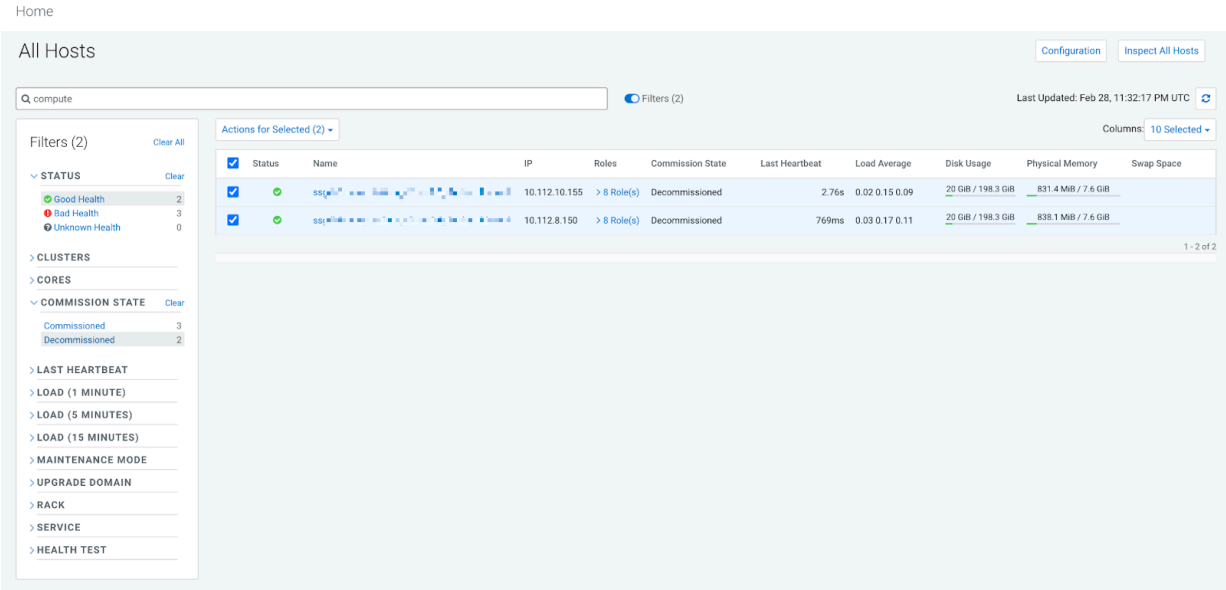
Manually recovering from load-based scaling failures

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 AWS. This can be replicated manually by changing the ‘Instance State’ on the AWS 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 SelectedEnd 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 SelectedBegin 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. This can be replicated manually by changing the ‘Instance State’ on the AWS 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 AWS 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 ClouderaManager 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 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 (ActionsResize). Re-enable autoscaling.

Autoscaling FAQ

Review frequently asked questions about 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 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 CDP 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/Data Hub database

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

About this task

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

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

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

Note: In accordance with the PostgreSQL Versioning Policy,the aforementioned cloud database services will be ending support for PostgreSQL major version 10 on November 10, 2022. AWS and GCP offer a 6 months extended support period beyond this date, whereas Azure has a strict policy.

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

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

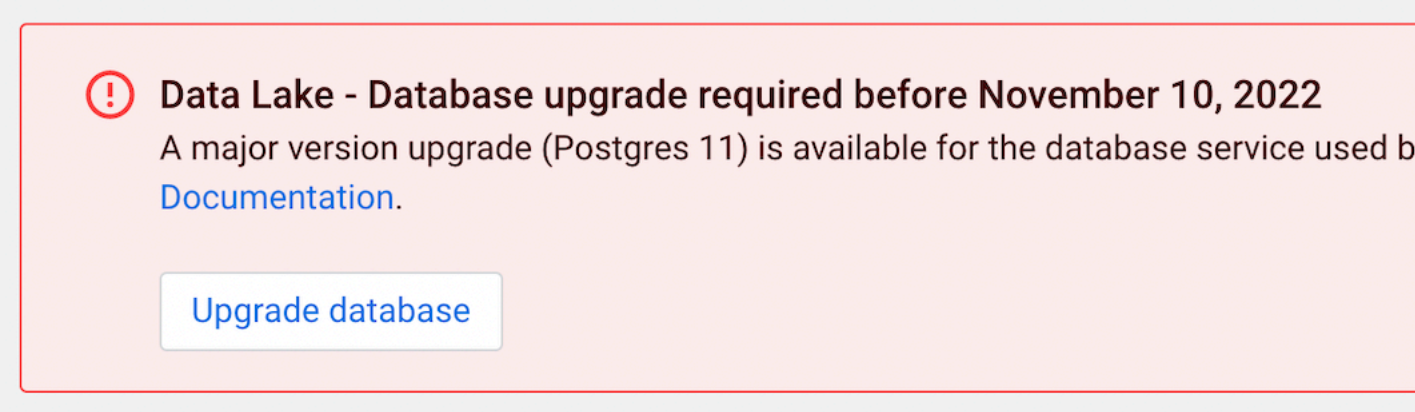
If you wish to disregard this recommendation, you may do so considering the risks involved as per the Cloud service Provider policies, see Versioning policy- Azure Database for PostgreSQL.

The Database Upgrade is a new operation, complementary to the existing maintenance, minor/major version and OS upgrades, as described in the CDP Public Cloud Upgrade Advisor.

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

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

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



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

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

Note: Cloudera strongly recommends stopping all workloads in Data Services that interact with the Data Lake.

If you are concerned about stopping the workloads in your deployment, contact Cloudera support for a custom upgrade path.

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

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

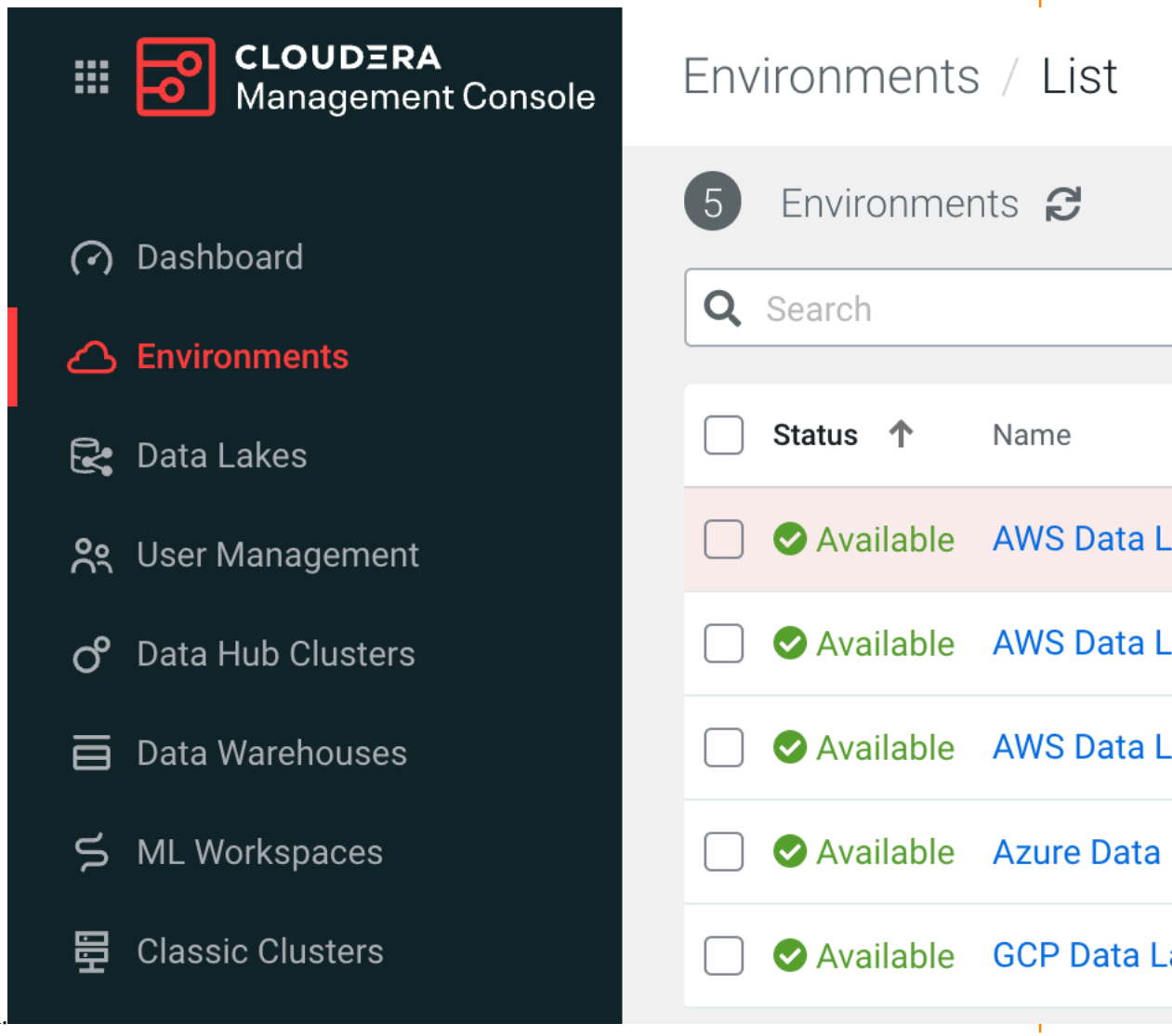
Instructions

Here are the UI and CLI instructions to perform database upgrade on Data Lake and Data Hub:

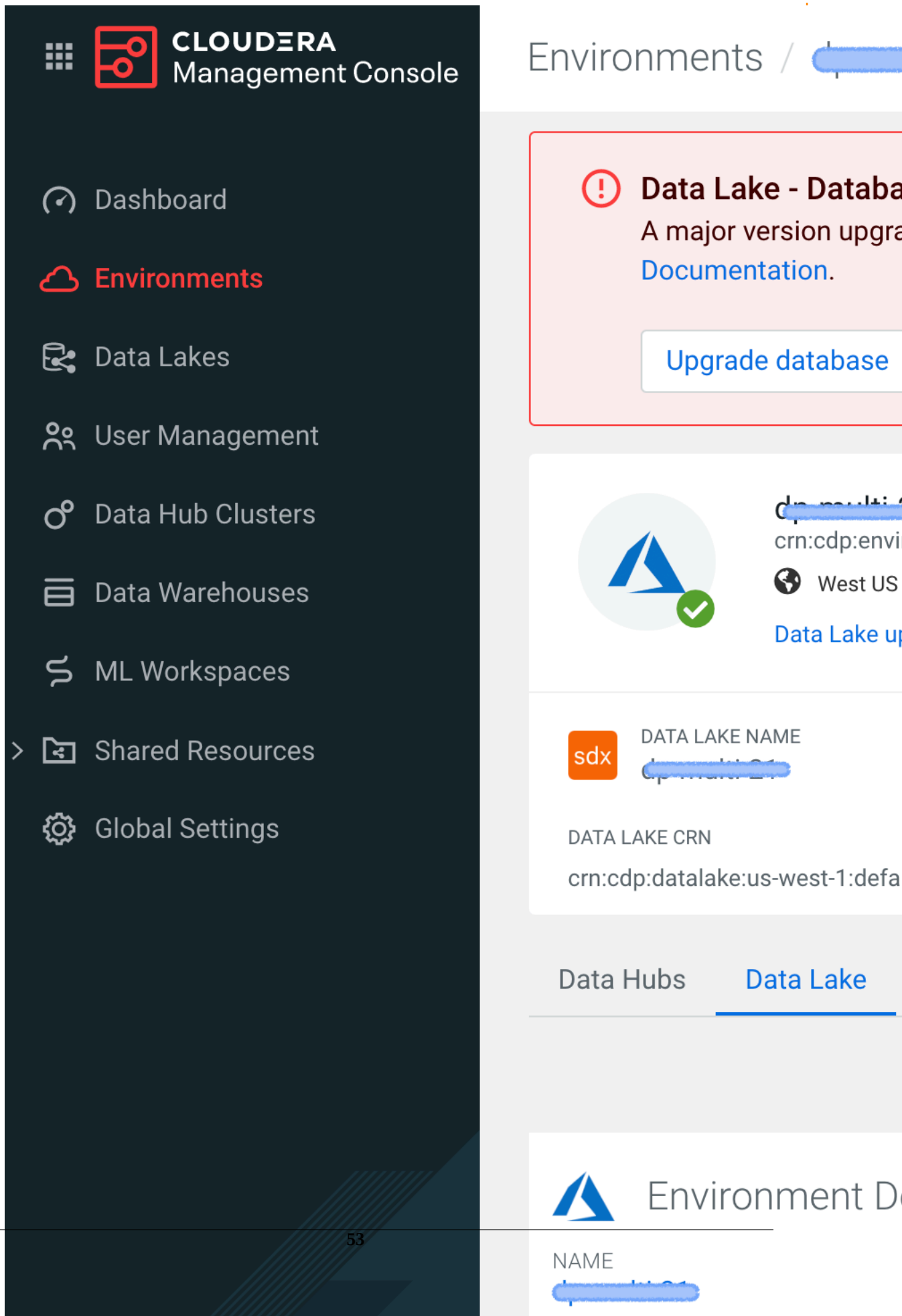
For CDP UI

Steps

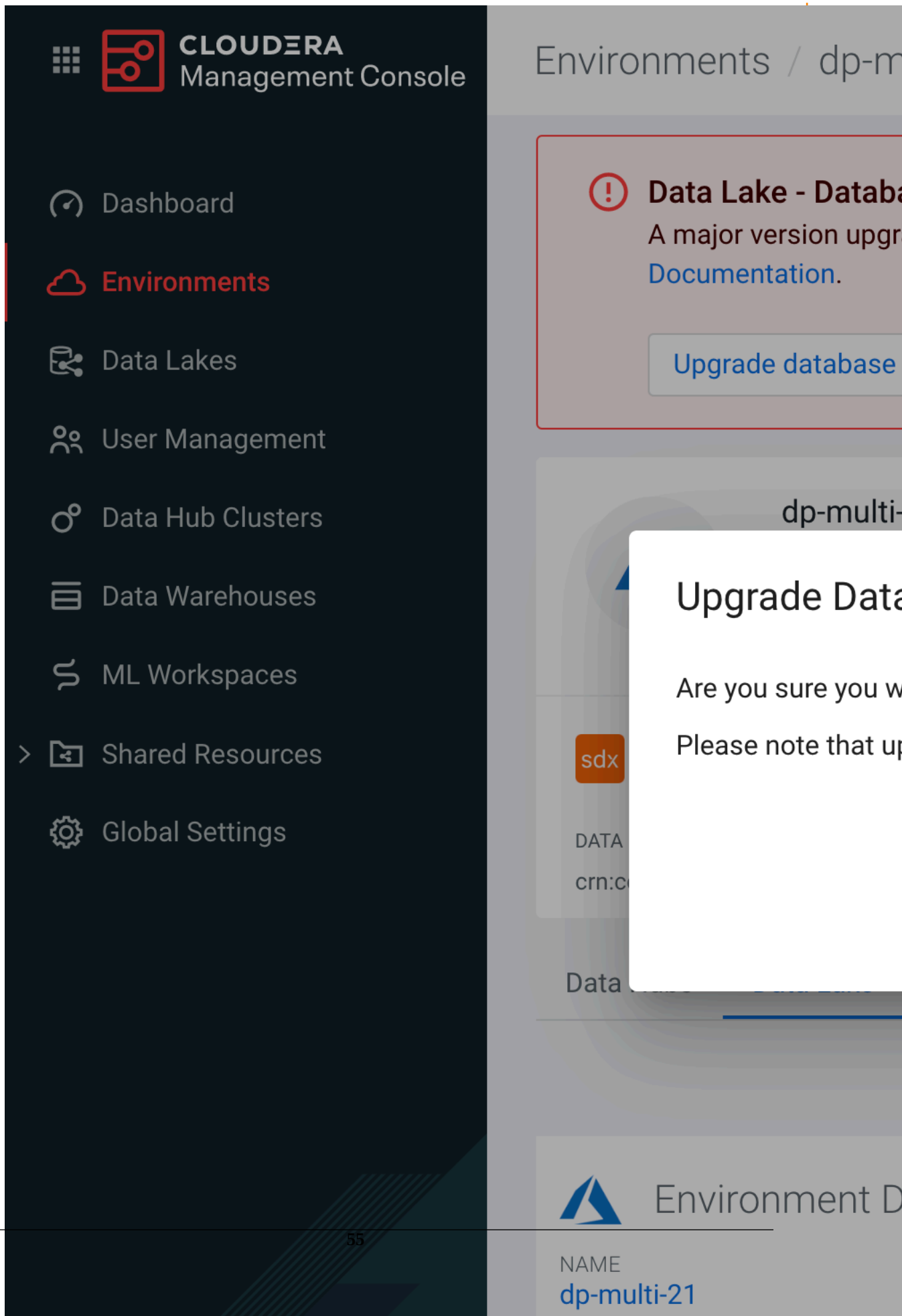
1. In CDP Management Console UI, go to Environments. Select the cluster to perform the Upgrade from the list of available clusters. The clusters are eligible for this upgrade are indicated in the right most column.



2. Once you select the cluster, you will see a message asking to update the Postgres version. Click the Upgrade database.


CLOUDERA
Management Console
Dashboard
Environments
Data Lakes
User Management
Data Hub Clusters
Data Warehouses
ML Workspaces
Shared Resources
Global Settings
Environments /
Data Lake - Databa
A major version upgra
Documentation.
Upgrade database
crn:cdp:envi
West US
Data Lake up
sdx
DATA LAKE NAME
DATA LAKE CRN
crn:cdp:datalake:us-west-1:defa
Data Hubs
Data Lake
Environment D
NAME

3. Click Upgrade in the confirmation box.


CLOUDERA
Management Console
Dashboard
Environments
Data Lakes
User Management
Data Hub Clusters
Data Warehouses
ML Workspaces
Shared Resources
Global Settings
Environments / dp-m
Data Lake - Databa
A major version upgr
Documentation.
Upgrade database
dp-multi-
Upgrade Data
Are you sure you w
Please note that up
sdx
DATA
crn:c
Data
Environment D
NAME
dp-multi-21

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

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

For CDP CLI

Data Lake Database upgrade:

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

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

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

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

          Possible values:

          o VERSION_11
```

Data Hub Database upgrade:

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

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

DESCRIPTION
       This command initiates the upgrade of the database  of  the  Data
Hub
       cluster.
SYNOPSIS
            start-database-upgrade
          --datahub <value>
          --target-version <value>
          [--cli-input-json <value>]
          [--generate-cli-skeleton]

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

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

The progress of the upgrade can be tracked on the respective service's Event History page. Once the upgrade is complete, Cloudera recommends verifying your workloads before attempting an additional Runtime or OS upgrade.

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

Database upgrade known limitations and troubleshooting

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

Known limitations and troubleshooting:

- Performing the Database Upgrade on Runtime versions 7.2.6 or below

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

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

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

  Workaround: If you acknowledge the risk and confirm that all cluster services and third party components relying on the Hive Metastore will be stopped for the time of the Database Upgrade, Cloudera can grant an entitlement that allows performing the upgrade with a running Data Hub cluster on an exceptional basis.
- PostgreSQL client binaries will upgraded to version 11 on all clusters hosts

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

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

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

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

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

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

- Exceeding the End of Life deadline

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

  Workaround: Cloudera recommends performing the Database Upgrade via the CDP UI, or CLI as soon as possible.
- Possibility of custom config reset after Database upgrade on AWS and Azure

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

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

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

Installing Postgres 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
     ```

Managing certificates

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

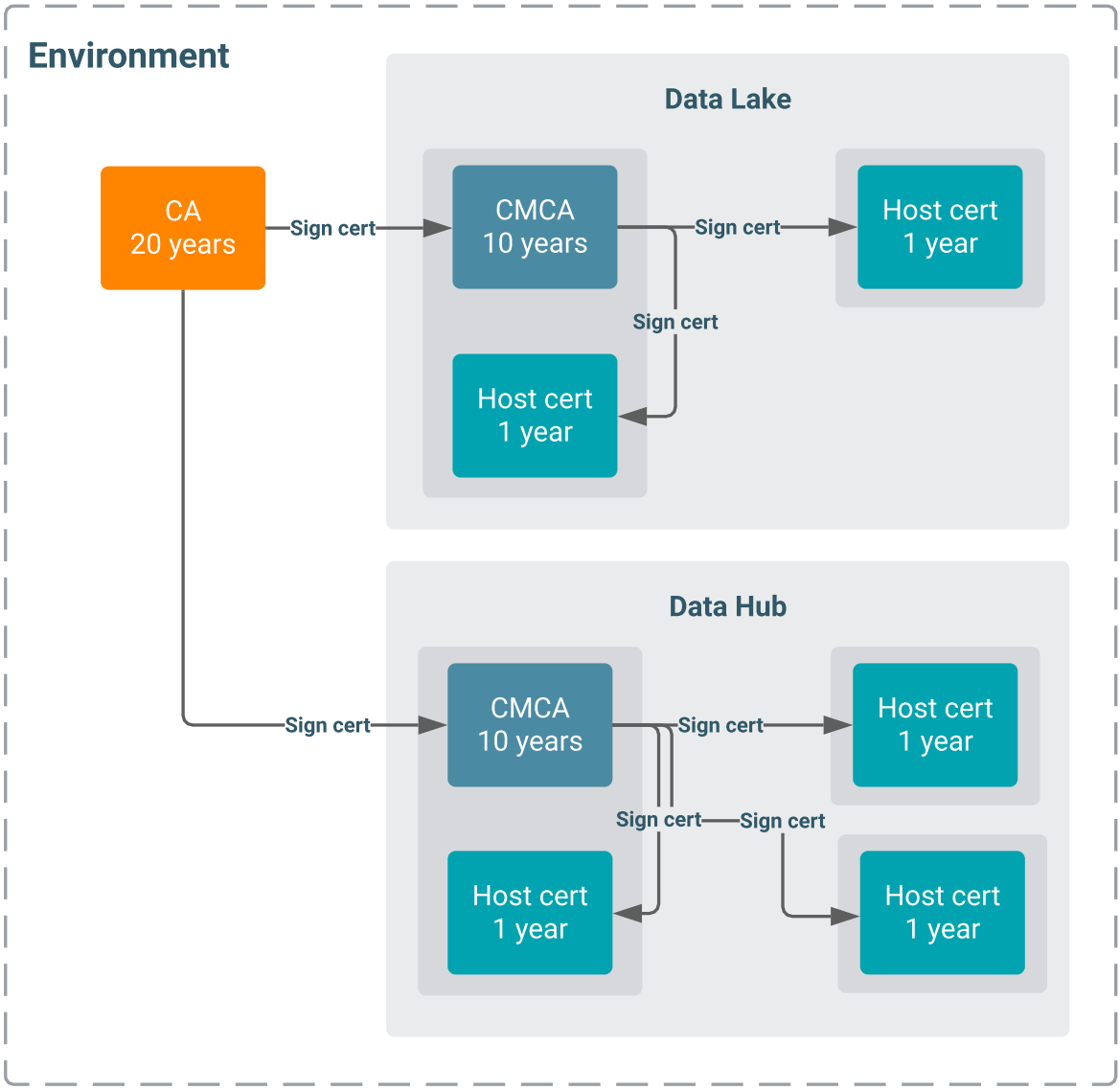
- Public certificates are Let's Encrypt-issued certificates for Data Hub and Data Lake clusters. These certificates are available on port 443 (HTTPS) of the cluster and are responsible for enabling TLS in front of Knox and other available services on that port. They are valid for 90 days, and in most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

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

  - Data Hub or Data Lake clusters created on or after March 7, 2022 are eligible for automatic renewal of public certificates. Clusters created before March 7, 2022, must be renewed manually once following the instructions in *Manually renewing public certificates for Data Lake and Data Hub clusters*. After the public certificate for a cluster has been manually renewed once from the CDP UI or CLI, it is eligible for automatic certificate renewal in the future.
  - If an automatic renewal fails, the renewal service will retry the renewal for three consecutive days or three attempts. Any cluster that cannot be renewed by these retry attempts must be renewed manually through the CDP UI or CLI.
  - The 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.
  - 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 renewed manually from the UI or CLI.
  - If a public certificate expires, you'll receive a warning that your connection is not secure when you attempt to access a Data Lake or Data Hub cluster through the CDP UI.

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

- Private certificates, or host certificates, are certificates created during cluster provisioning for every host with Auto-TLS. Private/host certificates have a default expiration date of one year. As private certificates get closer to expiration, the CDP UI displays a warning that the certificate is about to expire.



Though the CDP UI displays a warning about the expiration of private/host certificates, you are still responsible for renewing them through the UI or CDP CLI. After the certificates expire, the cluster is not functional, so you must renew them before expiration.

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

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

About this task

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

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

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

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

Data Lake certificate renewal:

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

Data Hub certificate renewal:

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

Alternatively, you can wait until the host certificate is close to expiration. During periodic cluster state synchronization, CDP uses the Cloudera Manager API to check that the HOST_AGENT_CERTIFICATE_EXPIRY apiHealthCheck alert is in a GOOD state. If the apiHealthCheck is not in a GOOD state, CDP displays a warning in the UI.

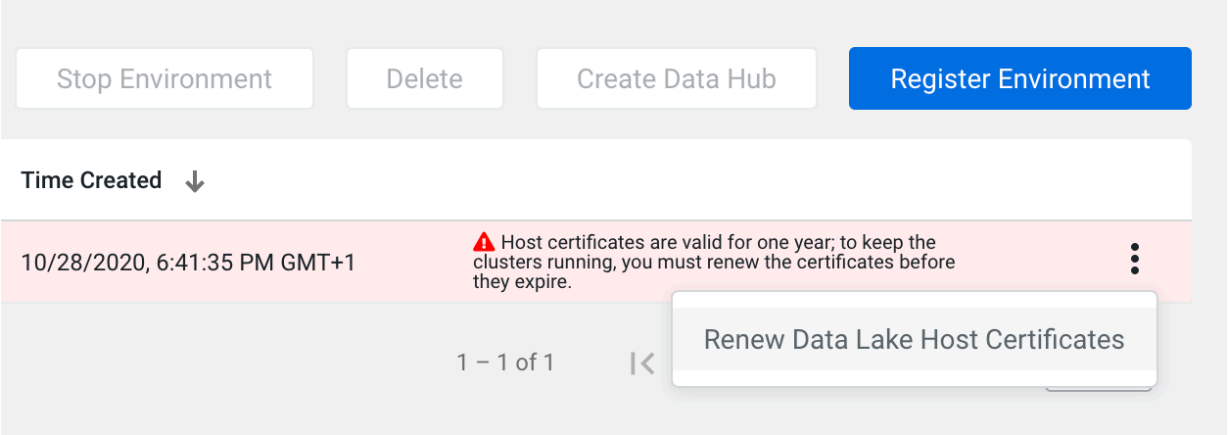
These UI warnings will display on the associated Environments, Data Lakes, or Data Hubs list and details pages. For example:

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

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

Procedure

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



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

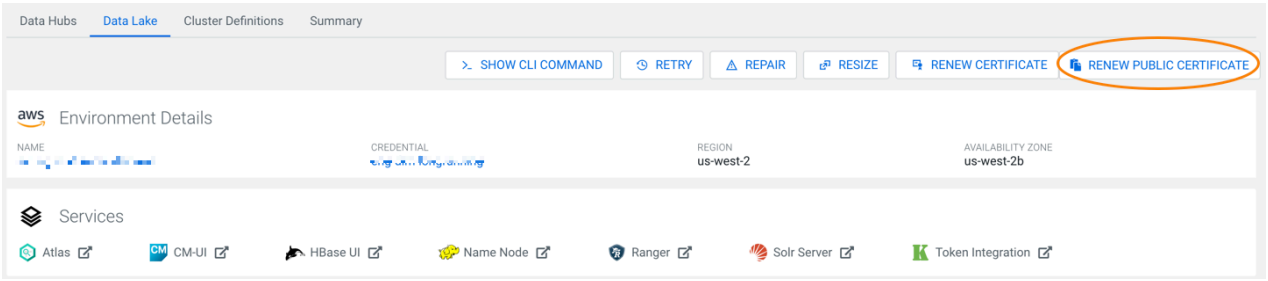
Manually renewing public certificates for Data Lake and Data Hub clusters

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

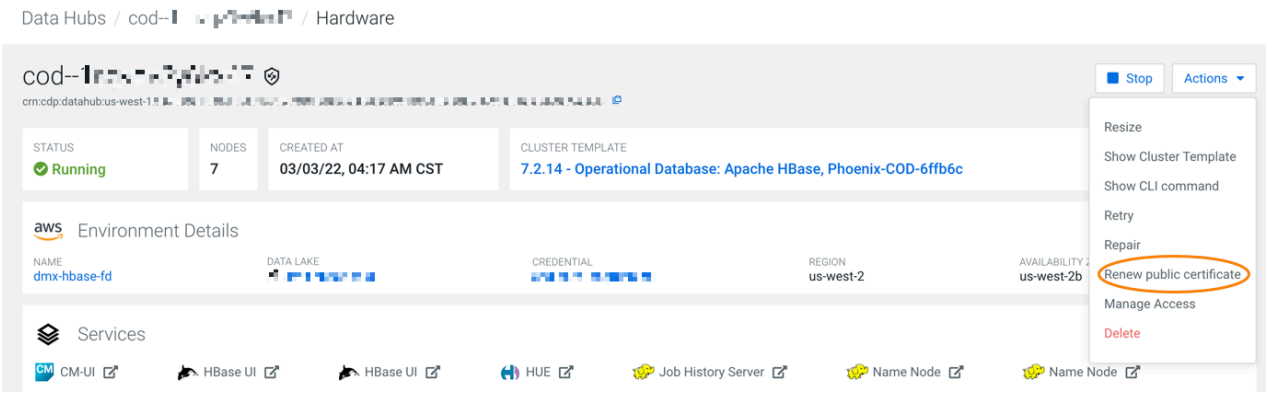
Required role (Data Lakes): EnvironmentAdmin or Owner of the environment

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

To renew a public certificate, click the Renew Public Certificate button on the details page of a chosen Data Lake:



Or from the Actions menu of a Data Hub cluster details page:



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

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

Data Lake public certificate renewal:

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

Data Hub public certificate renewal:

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

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 Data Hub Clusters

Managing clusters from CLI

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

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

Monitoring clusters

- 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>

Managing clusters

- 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>