

## Managing Virtual Warehouses

Date published: 2020-08-17

Date modified: 2022-09-16



# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

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

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

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

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

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

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

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

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

# Contents

<b>Adding a new Database Catalog.....</b>	<b>5</b>
<b>Adding a new Virtual Warehouse.....</b>	<b>5</b>
<b>Configuring Impala coordinator shutdown.....</b>	<b>6</b>
<b>Modifying the size of a Virtual Warehouse.....</b>	<b>7</b>
<b>SSL-enabled endpoints for Virtual Warehouse clients in Cloudera Data Warehouse Private Cloud.....</b>	<b>8</b>
<b>Auto-scaling Virtual Warehouses.....</b>	<b>9</b>
Tuning Hive Virtual Warehouses on private clouds.....	10
Tuning Impala Virtual Warehouses.....	12
Auto-scale threshold settings.....	13
<b>Compaction in Cloudera Data Warehouse.....</b>	<b>14</b>
How compaction works.....	15
Compactor processes.....	15
How compaction interacts with CDP Base.....	16
Cloudera Data Warehouse Private Cloud Compaction Architecture.....	16
Considerations for using compaction on Cloudera Data Warehouse Private Cloud.....	17
Change compactor configuration for Hive Virtual Warehouses on Cloudera Data Warehouse Private Cloud.....	17
<b>Debugging with Impala Web UIs.....</b>	<b>18</b>
<b>Using Ozone storage with Cloudera Data Warehouse Private Cloud.....</b>	<b>19</b>
Creating a database on Ozone for Cloudera Data Warehouse Private Cloud Virtual Warehouses.....	19
Configuring Hive/Impala logging on Ozone for Cloudera Data Warehouse Private Cloud.....	21
Specify or create an Ozone bucket for Cloudera Data Warehouse Private Cloud logs.....	21
Update Cloudera Data Warehouse Private Cloud log configuration to point to Ozone.....	22
Monitor Cloudera Data Warehouse Private Cloud logs on Ozone storage.....	23
Analyze Cloudera Data Warehouse Private Cloud logs stored on Ozone.....	24
<b>Locating Cloudera Data Warehouse Private Cloud logs.....</b>	<b>24</b>
<b>Generating and downloading diagnostic bundles.....</b>	<b>25</b>

<b>Configuring Impala Virtual Warehouses to create Impala tables in Kudu in Cloudera Data Warehouse Private Cloud.....</b>	<b>26</b>
<b>Configuring Impala Virtual Warehouses to encrypt spilled data in Cloudera Data Warehouse Private Cloud.....</b>	<b>28</b>
<b>Customizing Impala pod configuration.....</b>	<b>29</b>

## Adding a new Database Catalog

In addition to the default Database Catalog, created automatically, you can add additional Database Catalogs if you want a standalone data warehouse without any data related to the default Database Catalog.

### About this task

When you activate an environment from the Data Warehouse, a default Database Catalog is created and named after your environment. This HMS instance associated with the default Database Catalog is the same HMS as the one used by your CDP environment. You can add additional Database Catalogs if you want standalone data warehouses based on a new HMS instance. When you create a new Database Catalog, you specify which environment to use. If you make a change to the default database catalog, the change is reflected in the environment where the default Database Catalog resides. However, if you make any change to the non-default database catalogs, the change is not reflected in that environment.

You can optionally load demo data in Hue when you create a new Database Catalog.

Required role: DWAdmin

### Before you begin

You must enable the entitlement `CDW_VERSIONED_DEPLOY` to select the image version in step 6.

### Procedure

1. Log in to the CDP web interface and navigate to the Data Warehouse service.
2. In the Data Warehouse service, click Database Catalogs.
3. Click Add New.

You can also add a new Database Catalog by clicking the plus sign on the Overview page of the Data Warehouse service.

4. In Name, specify a Database Catalog name.
5. In Environments, select the name of your environment.

If you do not see the environment you want in the drop-down list, you might need to activate the environment.

6. Turn on Load Demo Data if you want to use sample airline data in Hue.
7. Click Create to create the new Database Catalog.

## Adding a new Virtual Warehouse

This topic describes how to create a Virtual Warehouse in Cloudera Data Warehouse (CDW) Private Cloud service.

### About this task

In CDW service, a Virtual Warehouse is an instance of compute resources that is equivalent to a cluster. A Virtual Warehouse provides access to the data in tables and views in the data lake that correlates to a specific Database Catalog. Virtual Warehouses can only lookup the Database Catalog that they have been configured to access.

Required role: DWAdmin

### Before you begin

Before you create a new Virtual Warehouse, determine what is the number of concurrent queries or users your Virtual Warehouse must serve during peak periods. This information helps you determine what size of Virtual Warehouse you need. Choose the size based on the number of nodes you typically use for clusters in an on-premises deployment.

Also consider the complexity of your queries and the size of the data sets that they access. Larger sized warehouses with more nodes can cache more data, which enhances performance.

Virtual Warehouse sizes you can choose from:

Virtual Warehouse Size	Number of Nodes
XSMALL	2
SMALL	10
MEDIUM	20
LARGE	40

### Procedure

1. Log in to the CDP web interface and navigate to the Data Warehouse service.
2. In the Data Warehouse service, click Virtual Warehouses in the left navigation panel.
3. On the Virtual Warehouses page, click Add New.
4. In the New Virtual Warehouse dialog box, specify a Virtual Warehouse name, the Type (Hive or Impala), which Database Catalog it queries, and the size.
5. After you choose a size, you can configure auto-scaling settings.
6. Click Create to create the new Virtual Warehouse.

## Configuring Impala coordinator shutdown

To optimize resource utilization, you need to know how to configure Impala coordinators to automatically shutdown during idle periods. You need to know how to prevent unnecessary restarts. Monitoring programs that periodically connect to Impala can cause unnecessary restarts.

### About this task

When you create a Virtual Warehouse, you can configure Impala coordinators to automatically shutdown during idle periods. The coordinator start up can last several minutes, so clients connected to the Virtual Warehouse can time out.

### Before you begin

Update impyla, jdbc, impala shell clients if used to connect to Impala.

### Procedure

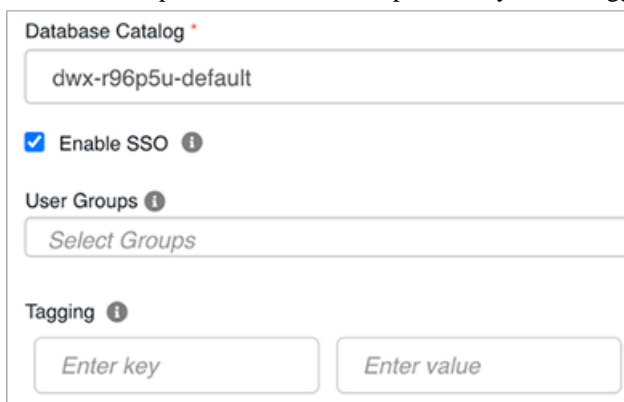
1. Log in to the Data Warehouse service.
2. Click **+** under Virtual Warehouses to add a new Virtual Warehouse.
3. Select IMPALA as the SQL engine type.



The screenshot shows a 'New Virtual Warehouse' dialog box. It has a 'Name' field with the text 'impala' and a 'Type' field with two buttons: 'HIVE' and 'IMPALA'. The 'IMPALA' button is highlighted, indicating it is the selected option.

4. Select a database catalog, or accept the default.

- Set User Groups that can access endpoints, keys, and Tagging for the Virtual Warehouse.

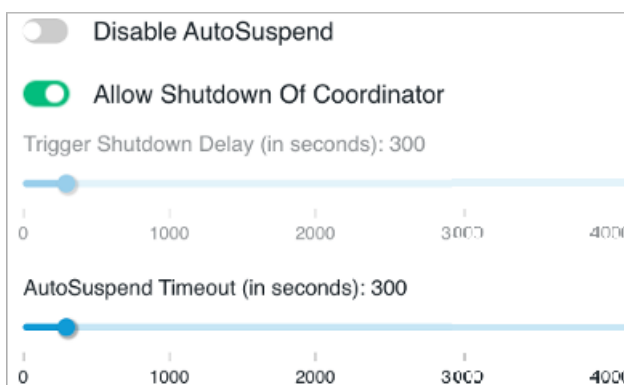


- Select a size for the Virtual Warehouse.

- Turn off Disable AutoSuspend.

The Impala coordinator does not automatically shutdown unless the Impala executors are suspended.

- Turn on Allow Shutdown of Coordinator.



After Impala executors have been suspended, the Impala coordinator waits for the time period specified by the Trigger Shutdown Delay before shutting down.

For example, if AutoSuspend Timeout = 300 seconds and Trigger Shutdown Delay=150 seconds, after 300 seconds of inactivity Impala executors suspend, and then 150 seconds later, the Impala coordinator shuts down.

## Modifying the size of a Virtual Warehouse

This topic describes how to change the size of a Virtual Warehouse in Cloudera Data Warehouse (CDW) service.

### About this task

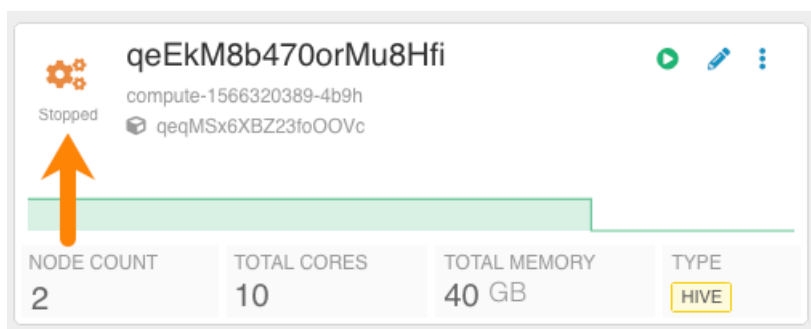
After you create a Virtual Warehouse, you can tune the auto-scaling thresholds if your workload demand increases or decreases, but you cannot adjust the size of the warehouse. The size determines the number of executors of your Virtual Warehouse, which determines the maximum number of concurrent queries it can serve. To change the size of your Virtual Warehouse, you must delete it, and then recreate it in the new size you require.

Required role: DWAdmin


### Before you begin

Only delete Virtual Warehouses that are stopped. To determine whether a Virtual Warehouse is stopped or running, check the Stopped or Running icon on its tile in the Overview page or in the list on the Virtual Warehouses page:

**Figure 1: Virtual Warehouse Stopped/Running icon on the Overview page**



### Procedure

1. Log in to the CDP web interface and navigate to the Data Warehouse service.
  2. On the Overview page of the Data Warehouse service, make note of the name of the Virtual Warehouse you want to modify and which Database Catalog it is configured to access.
  3. In the tile for the Virtual Warehouse that you want to modify, click the options menu and select Delete.
  4. Still on the Overview page of the Data Warehouse service, in the Virtual Warehouse column, click the plus sign in the upper right corner of the tile to create a new Virtual Warehouse.
  5. In the New Virtual Warehouse dialog box:
    - a) Type the name of the Virtual Warehouse that you just deleted.
-  **Important:** The fully qualified domain name of your Virtual Warehouse, which includes the Virtual Warehouse name plus the environment name must not exceed 64 characters; otherwise, Hue cannot load.
- b) Select the Database Catalog that the deleted warehouse was configured to access.
  - c) Select the new size.
  - d) Configure the auto-scaling thresholds.
6. Click Create.

## SSL-enabled endpoints for Virtual Warehouse clients in Cloudera Data Warehouse Private Cloud

In Cloudera Data Warehouse (CDW) Private Cloud 1.1, all client endpoints have been SSL-enabled. This requires that you configure the SSL certificates for client endpoints.

In CDW Private Cloud 1.1 and higher, client endpoints for web applications and Virtual Warehouse client URLs are SSL-enabled. The following endpoints use the OpenShift/Embedded Container Service cluster default certificate:

- Hue
- Data Analytics Studio (DAS) webapp
- Impala coordinator
- HiveServer2

### Domain name changes

To use the OpenShift/Embedded Container Service cluster wildcard certificate, the DNS names have been changed. The environment ID sub domain from the domain name has been removed. This creates a flat DNA structure so the cluster wildcard certificate can be applied to the endpoints.

### Generating a truststore for a self-signed certificate

You can query the service certificate and convert it to a JKS truststore using the following steps:



1. Retrieve the certificate:

```
$ openssl s_client -showcerts -connect hs2-my-cwh1.apps.cdw.mycloud.myfirm.com:443 -servername
hs2-my-cwh1.apps.cdw.mycloud.myfirm.com </dev/null|openssl x509 -outform
PEM > <mycertfile>.pem
```

2. Convert the PEM file to a truststore. You will be prompted for a password.

```
$ keytool -import -alias hs2-my-cw1.apps.cdw.mycloud.myfirm.com -file
<mycertfile>.pem -keystore <mycert>.jks
```

## Opening SSL-enabled connections with Database Catalog clients

The CDW Virtual Warehouse clients like beeline and impala-shell can open SSL-enabled connections as described in this section.

### Beeline

A beeline connection can be created using a JDBC connection string. Specifying the username and password with the '-n' and the '-p' options returns an error. The beeline CLI prompts for credentials:

```
$ beeline
beeline> !connect
jdbc:hive2://hs2-my-cwh1.apps.cdw.mycloud.myfirm.com:443/default;transportMode=http;httpPath=cliservice;
    ssl=true;retries=3;sslTrustStore=<JKS-path>;trustStorePassword
=<***password***>
Enter username for jdbc:hive2://hs2-my-cwh1.apps.cdw.mycloud.myfirm.com:443/
default:<my-user-name>
Enter password for jdbc:hive2://hs2-my-cwh1.apps.cdw.mycloud.myfirm.com:443/
default:<*****>
```



**Important:** The value for <JKS-path> is generated in the above section "Generating a truststore for a self-signed certificate."

### impala-shell

The impala-shell CLI opens a TLS/SSL-enabled connection when you use the '--ssl' option. If '--ca\_cert' is not set, impala-shell enables TLS/SSL, but does not validate the server certificate. Set the '--ca\_cert' CLI option to the local path name that points to the third-party CA certificate, or to a copy of the server certificate in the case you have a self-signed server certificate:

```
$ impala-shell --protocol='hs2-http' -i "coordinator-my-iwh2.apps.cdw.myclou
d.myfirm.com:443" --ssl
```

## OpenShift routes

OpenShift routes are used to expose the user-facing services in the CDW Private Cloud deployment. Route objects can perform edge TLS termination using the cluster-deployed certificate for the endpoints. If the cluster certificate must be rotated, the routes can pick up the new certificate automatically. It is not necessary to re-deploy or to manually configure the service in order to pick up the changes.

# Auto-scaling Virtual Warehouses

This topic provides an overview of auto-scaling in Cloudera Data Warehouse (CDW) Private Cloud.

Virtual Warehouses can use Hive or Impala as the underlying execution engine. Typically, Hive is used to support complex reports and enterprise dashboards. Impala is used to support interactive, ad-hoc analysis. When you create a Virtual Warehouse, you set auto-scaling to make sure you have adequate resources to meet increases in demand. Auto-scaling settings also insure that your Virtual Warehouse relinquishes resources when demand decreases to save costs.

### Auto-scaling: where scaling and concurrency meet

*Scaling* is the total capacity of the system and how elastic it is. System capacity requirements are based on the size of the largest query you need to run on a warehouse. *Concurrency* is the number of queries that can run at the same time in the same Virtual Warehouse.

In traditional deployments, scaling and concurrency must be planned for before you deploy your warehouse. In the cloud, the ability to acquire better scaling and concurrency elastically in response to workload demand enables the system to operate more efficiently than the maximum limits you plan for. If you run your Virtual Warehouse configured to accommodate your peak workload as a constant default configuration, you might have inefficient resource utilization when system demand falls below that level.

### Caching and auto-scaling

In CDW Private Cloud service frequently accessed data is cached on HDFS. This caching ensures that the data can be quickly retrieved for subsequent queries, boosting data warehouse performance.

### Fault-tolerance and auto-scaling

Virtual Warehouses can tolerate single-node failures of any of its workers and can continue running active queries. Auto-scaling separates nodes from each other. This node separation provides better protection when a rogue query is submitted to the warehouse. In this scenario, node failures are limited to the auto-scaled nodes, which limits the impact of the rogue query to a small part of the Virtual Warehouse. The choice to have more auto-scaling groups indirectly allows for the system to tolerate such scenarios, so it is always recommended to use auto-scaling, even if the workload is a predictable one.

## Tuning Hive Virtual Warehouses on private clouds

This topic describes how to tune Hive Virtual Warehouses in Cloudera Data Warehouse (CDW) Private Cloud.

### About this task

When you tune Hive Virtual Warehouses, you set the auto-suspend timeout, the minimum and maximum number of nodes for your virtual cluster, when your cluster should scale up, and when it should scale down.

### Procedure

1. Log in to the CDP web interface and navigate to the Data Warehouse service.
2. In the Data Warehouse service, click Overview in the left navigation pane.



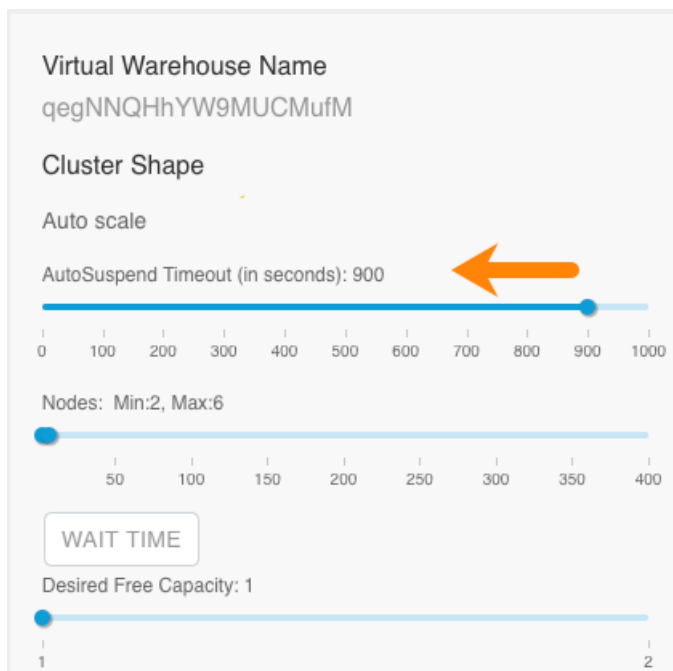
**Note:** You can also tune your data warehouse on the Virtual Warehouse page using the same steps.

3. In the Overview page under Virtual Warehouses, click



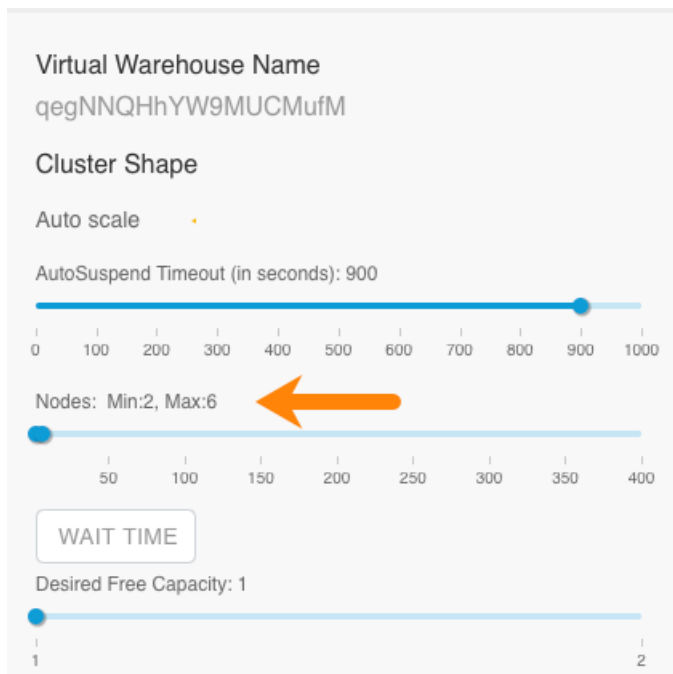
against the required virtual warehouse and select Edit.

4. Click the **SIZING AND SCALING** tab to view the properties that you can adjust to tune auto-scaling for your data warehouse:
  - a) Set the AutoSuspend Timeout property under Auto scale, which determines how many seconds the warehouse cluster is idle before it suspends itself:



This setting helps to ensure performance is not impacted by having idle resources.

- b) Set the minimum and maximum number of nodes that the cluster can contain:



Use the minimum number of nodes setting to ensure that your workloads always have resources and use the maximum number of nodes setting to contain having too many idle resources. Decide the minimum and maximum number of nodes based on your workloads similarly to how you determine node counts for your on-premises clusters. Consider the number of concurrent queries, the complexity of queries, and the volume

of queries in your workloads to determine the appropriate number of nodes to set on each Virtual Warehouse instance.

- c) Choose when your cluster auto-scales up based on the WAIT TIME setting, which sets how long queries wait in the queue to run before the cluster auto-scales up. For example, if WaitTime Seconds is set to 10, then when executing queries are waiting in the queue for 10 seconds, the cluster auto-scales up to meet query demand.



**Note:** Scaling might react to non-scalable factors to spin up clusters. For example, query wait times might increase because of inefficient queries and not because of query volume.

- d) Select Query Isolation if you have scan-heavy, data-intensive queries in your workloads.

Query Isolation enables your Virtual Warehouse to determine, based on the value you set for the `hive.query.isolation.scan.size.threshold` configuration parameter, whether to spawn dedicated nodes to run scan-heavy, data-intensive queries.

You can set this threshold parameter in the Virtual Warehouse details page for the warehouse:

1. In the Data Warehouse service UI, click Virtual Warehouses in the left navigation pane.
2. From the list of warehouses, click the Virtual Warehouse you want to configure this parameter for.
3. In the Virtual Warehouse details page, click CONFIGURATIONS Hiveserver2 .
4. Select hive-site from the Configuration files drop-down list and type isolation in the search text box to locate the parameter.
5. In the VALUE text box for the `hive.query.isolation.scan.size.threshold` parameter, enter the amount of data for your threshold in storage units. For example, 400GB.
6. Click APPLY to save your settings.

After you enable Query Isolation, two more configuration options appear:

- Max Concurrent Isolated Queries: Sets the maximum number of isolated queries that can run concurrently in their own dedicated executor nodes. Select this number based on the scan size of the data for your average scan-heavy, data-intensive query.
- Max Nodes Per Isolated Query: Sets how many executor nodes can be spawned for each isolated query.

- e) Click APPLY.

## Tuning Impala Virtual Warehouses

This topic describes how to tune Impala Virtual Warehouses in Cloudera Data Warehouse (CDW) Private Cloud.

### About this task

When you tune Impala Virtual Warehouses, you can disable the auto-suspend feature, set the minimum and maximum executor nodes allocated for the warehouse and you can set the scale up and scale down delay which determines auto-scaling behavior.

Required role: DWAdmin

### Procedure

1. Log in to the CDP web interface and navigate to the Data Warehouse service.
2. In the Data Warehouse service, navigate to the Overview page.




**Note:** You can also tune your warehouse on the Virtual Warehouse page using the same steps.

3. On the Overview page under Virtual Warehouses, click the edit icon for an Impala Virtual Warehouse in the upper right corner of the tile.

4. The next page provides properties that you can adjust to tune auto-suspend and auto-scaling for your Virtual Warehouse:
  - a) Set the auto-suspend behavior:
    - If you do not want your Virtual Warehouse to auto-suspend, click **Disable AutoSuspend**. If you enable this feature, your Virtual Warehouse does not suspend itself, even if it is idle and no workloads are being processed.
    - If you do not want to disable auto-suspend, set the **AutoSuspend Timeout**. This sets the time, in seconds, that it takes for the Virtual Warehouse to automatically suspend itself. A Virtual Warehouse auto-suspends itself when the auto-scaler has scaled back to the last executor group and those executors are idle.
  - b) Adjust the minimum or maximum number of executor nodes as needed:
 

Setting and adjusting the minimum and maximum number of executor nodes per Virtual Warehouse is very similar to setting the number of nodes for on-premises clusters. Keep in mind the number of concurrent queries, the complexity of queries, and the volume of queries in your workloads to determine the appropriate number of executor nodes to set on each Virtual Warehouse instance.
  - c) Set the **Scale Up Delay** and the **Scale Down Delay** to fine-tune when the auto-scaler starts scaling up and the number of executor groups to meet workload demand.
    - **Scale Up Delay**: Sets the length of time in seconds that the system waits before adding more executors when it detects queries waiting in the queue to execute. The time to auto-scale up is affected by how the underlying Kubernetes system is configured.
    - **Scale Down Delay**: Sets the length of time in seconds that the system waits before it removes executors when it detects idle executor groups. As with the **Scale Up Delay** setting, the time to auto-scale down is affected by how the underlying Kubernetes system is configured.
5. (Optional) If you need to tune your Impala Virtual Warehouse to run more queries per executor group, select **Use Legacy Multithreading Mode**.
 



**Note:** By default Impala Virtual Warehouses can run 3 large queries per executor group. Executors can handle more queries that are simpler and that do not utilize concurrency on the executor. When you enable legacy multithreading, the Virtual Warehouse can run 12 queries per executor group. For most read-only queries the default setting of 3 queries per executor group is sufficient.
6. Click **Apply** in the upper right of the page to save your changes.

## Auto-scale threshold settings

This topic provides information about the auto-scaling threshold settings for Hive and Impala Virtual Warehouses in Cloudera Data Warehouse (CDW) Private Cloud.

When you create new Virtual Warehouse instances, you can set auto-scaling thresholds. These thresholds set limits on automatic cluster scaling to meet workload demands. Setting these limits prevents warehouses from consuming too many resources when workload demands increase or decrease. Another important benefit of enabling auto-scaling for your Virtual Warehouse is that it further enforces node isolation, increasing warehouse fault tolerance. You can adjust the following auto-scaling thresholds:

### Hive-LLAP Virtual Warehouse auto-scaling threshold settings

The following settings are available to configure auto-scaling for Hive-LLAP Virtual Warehouses:

Hive-LLAP Auto-scaling Threshold	Description
AutoSuspend Timeout	Sets the maximum time the warehouse idles before shutting down.

Hive-LLAP Auto-scaling Threshold	Description
Nodes: Min: <n>, Max: <n>	<p>Sets the minimum and maximum number of nodes (executors) for the warehouse cluster. The maximum number of executors is limited by your cloud account limits.</p> <p>Choose the minimum and maximum number of executors based on two factors:</p> <ul style="list-style-type: none"> <li>Average number of queries that must be run concurrently for your workloads. The more queries that must be run concurrently, the larger number of executors are needed.</li> <li>The size of the data your workloads access. Larger numbers of executors can cache more data, which enhances performance.</li> </ul>
WAIT TIME	Sets how long queries wait in the queue to execute. For example, if WaitTime Seconds is set to 10, then when executing queries are waiting in the queue for 10 seconds, the cluster auto-scales up to meet query demand.
Query Isolation	Enables the Virtual Warehouse to determine, based on the value you set for the <code>hive.query.isolation.scan.size.threshold</code> configuration parameter, whether to spawn dedicated executor nodes to execute scan-heavy, data-intensive queries in isolation.
Max Concurrent Isolated Queries	Available if Query Isolation is enabled. Sets the maximum number of isolated queries that can execute concurrently in their own dedicated executor nodes. Select this number based on the scan size of the data for your average scan-heavy, data-intensive query.
Max Nodes Per Isolated Query	Available if Query Isolation is enabled. Sets how many executor nodes can be spawned for each isolated query.

### Impala Virtual Warehouse auto-scaling threshold settings

The following settings are available to configure auto-scaling for Impala Virtual Warehouses:

Impala Auto-scaling Setting	Description
Disable AutoSuspend	When you enable this control, your Virtual Warehouse does not suspend itself when the auto-scaler has scaled back to the last executor group, and those executors are idle. Instead, the Virtual Warehouse continues to consume cloud resources. You can override this behavior by disabling the Disable AutoSuspend control.
AutoSuspend Timeout	Sets the maximum time the warehouse idles before shutting down. This setting only applies when the Disable AutoSuspend control is not enabled.
Nodes: Min: <n> Max: <n>	<p>Sets the minimum and maximum number of nodes (executors) for the warehouse cluster. The maximum number of executors is limited by your cloud account limits.</p> <p>Choose the minimum and maximum number of executors based on two factors:</p> <ul style="list-style-type: none"> <li>Average number of queries that must be run concurrently for your workloads. The more queries that must be run concurrently, the larger number of executors is needed.</li> <li>The size of the data your workloads access. Larger numbers of executors can cache more data, which enhances performance.</li> </ul>
Scale Up Delay	Sets the length of time in seconds that the system waits before adding more executors when it detects queries waiting in the queue to execute. The time to auto-scale is affected by how the underlying Kubernetes system is configured.
Scale Down Delay	Sets the length of time in seconds that the system waits before it removes executors when it detects idle executor groups. As with the Scale Up Delay setting, the time to auto-scale down is affected by how the underlying Kubernetes system is configured.

## Compaction in Cloudera Data Warehouse

You understand the importance of compaction and the consequences of neglecting to perform compaction. Compaction keeps your Data Warehouse healthy.

Over time tables belonging to a workload become fragmented due to operations performed on them by your workload users. These small, obsolete files might lead to performance degradation and query latency problems. Compaction

plays a major role in improving response time to workload queries by reducing the number of underlying files for a table and eliminating the obsolete ones. Compaction runs periodically in the background to maintain the optimal state.

Running periodic compaction is a best practice for the performance for ACID transactions. ACID inserts and deletes generate the problematic files that you might need to monitor and manage. In Cloudera Data Warehouse (CDW), compaction is always performed by a Hive Virtual Warehouse.

## How compaction works

When data changes are made on Cloudera Data Warehouse (CDW) with inserts, updates, and deletes, delta files are created. The more changes that are made, the more delta files are created. When a large number of delta files are created, query performance degrades. Compaction removes these delta files to enhance query performance.


There are two types of compaction:

- Minor compaction: compacts multiple delta files into a single delta file.
- Major compaction: compacts one or more delta files and the base file for the bucket and creates a single new base file per bucket.

The goal of compaction is to "self heal" tables in order to restore the baseline query performance. All compactions are done in the background and do not prevent concurrent reads and writes of the data. After compacting, the system waits for all readers of the old files to finish and then removes the old files.

## Compactor processes

These background processes run inside the metastore and HiveServer2 in Cloudera Data Warehouse (CDW) Private Cloud. They support the data modifications made as a result of ACID transactions.

Compactor process	Description
Initiator	<p>This process runs in the metastore, which equates to the Database Catalog construct in the CDW UI, and discovers which tables and partitions are due for compaction. By default, it runs every 5 minutes.</p> <p> <b>Important:</b> For the default Database Catalog, which is the Database Catalog created automatically when an environment is activated, all compaction takes place on CDP Base.</p> <p>To change this interval:</p> <ol style="list-style-type: none"> <li>1. Identify the Database Catalog for the Virtual Warehouse on which you want to change the compaction interval by selecting the Virtual Warehouse tile. The associated Database Catalog is highlighted.</li> <li>2. In the Database Catalog, click the edit icon in the tile to launch the Database Catalog details page.</li> <li>3. On the Database Catalog details page, make sure the CONFIGURATIONS tab is selected, and then select the Metastore subtab.</li> <li>4. On the Metastore subtab, select hive-site from the drop-down list on the left, and search for the hive .compactor.check.interval KEY.</li> <li>5. Add your preferred check interval in the associated VALUE field in seconds.</li> <li>6. Click APPLY in the upper right corner of the page to apply your changes. The services are automatically updated with the new configuration.</li> </ol>
Worker	<p>This process runs in HiveServer2, which equates to the Hive Virtual Warehouse construct in the CDW UI. The worker process performs the actual compacting work. In CDW, compaction runs an INSERT statement created from the output of a SELECT statement, thereby re-writing the data to new base or delta files.</p>
Cleaner	<p>This process runs in the metastore and deletes delta files after compaction and after it determines the files are no longer needed. By default, the cleaner runs every 5 seconds (5,000 milliseconds). The check occurs on the visibility ID/transaction ID, which is a global transaction identifier.</p>

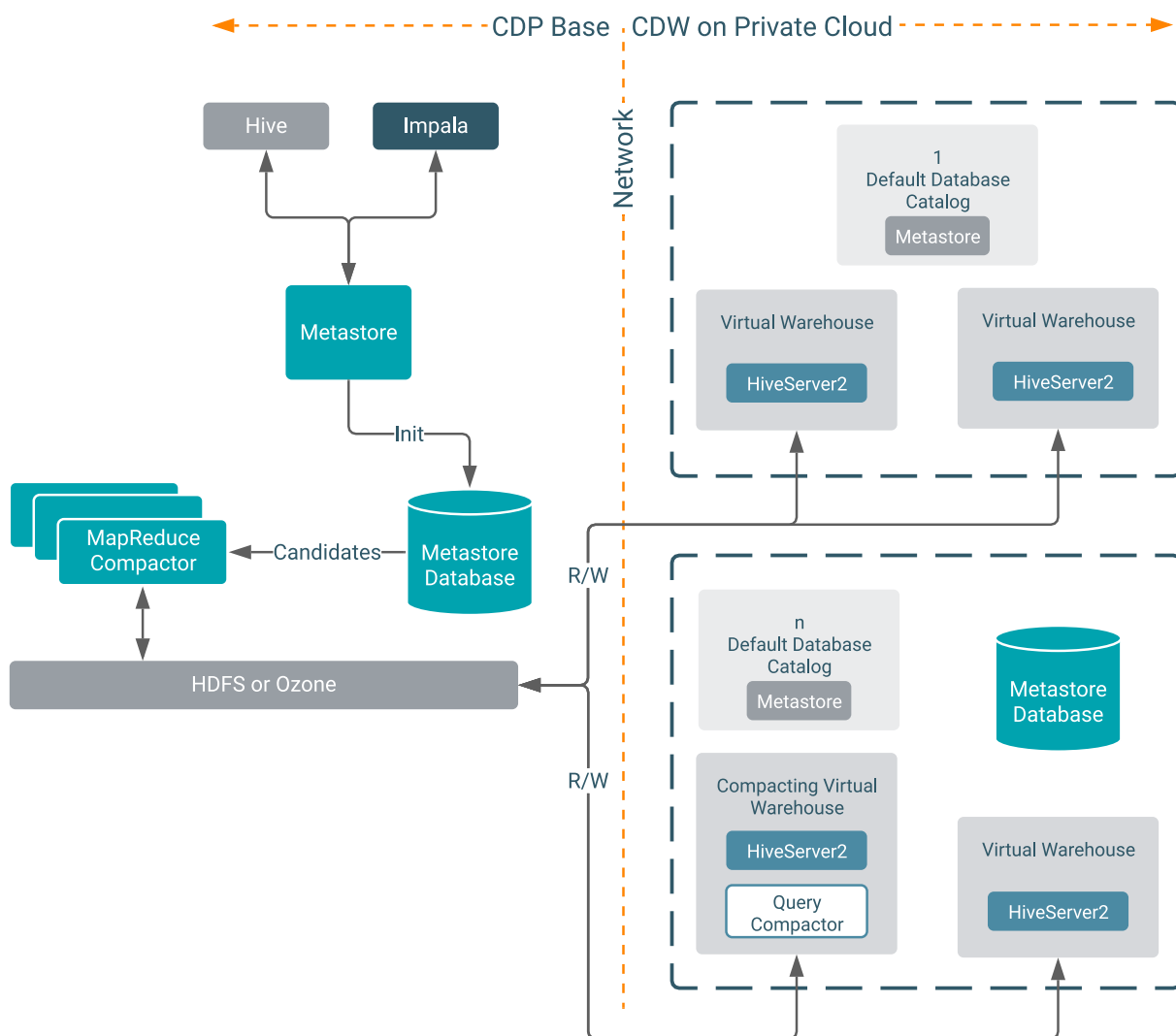
## How compaction interacts with CDP Base

In CDP Base, the initiator and cleaner processes also run in the metastore as they do in Cloudera Data Warehouse (CDW) Private Cloud. However, the worker process runs in HiveServer2 as a MapReduce task so its progress can be viewed in YARN.

In CDW, the initiator and cleaner processes run in the Database Catalog, which is the CDW UI construct that equates to the metastore. The default Database Catalog, which is created by the system when you activate an environment in CDW, maintains a connection with CDP Base and all compaction jobs for the default Database Catalog run on CDP Base. However, subsequent Database Catalogs that are created do not maintain a connection to CDP Base and compaction runs entirely in CDW. Also in CDW, the worker process that performs the compaction work runs in HiveServer2, which equates to a Hive Virtual Warehouse. However, compaction performed by the worker process in Hive Virtual Warehouses consists of queries instead of MapReduce tasks.

## Cloudera Data Warehouse Private Cloud Compaction Architecture

This diagram illustrates how the components that perform compaction interact on Cloudera Data Warehouse (CDW) Private Cloud. In CDW Private Cloud, all compaction tasks for the default Database Catalog are performed on CDP Base.





## Considerations for using compaction on Cloudera Data Warehouse Private Cloud

The first Hive Virtual Warehouse you create in Cloudera Data Warehouse (CDW) Private Cloud for a Database Catalog (not including the default Database Catalog) is automatically set as the compactor and performs all compaction work for subsequent Virtual Warehouses (Hive or Impala) created under that Database Catalog.

Consequently, you must take into account the query workload for compaction when you create the first Hive Virtual Warehouse. You must make sure that the warehouse has adequate resources to handle the compaction workload in addition to any other workloads you might run in that warehouse.



### Important:

- In the case of the default Database Catalog, all compaction takes place on CDP Base so you do not need to consider compaction queries for the Virtual Warehouses that use the default Database Catalog.
- Impala Virtual Warehouses cannot be designated as the compactor Virtual Warehouse for a Database Catalog. Compaction tasks can only be assigned to a Hive Virtual Warehouse.

## Change compactor configuration for Hive Virtual Warehouses on Cloudera Data Warehouse Private Cloud

To enhance performance, the compactor is a set of background processes that compact delta files, which are created as a by-product of data modifications. When it runs, it incurs additional load on the Hive Virtual Warehouse assigned as the compactor in Cloudera Data Warehouse (CDW) Private Cloud. You can change which Hive warehouse performs compaction to load-balance this workload as necessary.

### About this task

In CDW Private Cloud, data compaction is performed on HiveServer2, which equates to the Hive Virtual Warehouse construct in the UI. This means that compaction is essentially query execution. Compaction runs an INSERT statement created from the output of a SELECT statement and runs in the Hive Virtual Warehouse assigned as the compactor, thereby re-writing the data. The Hive Virtual Warehouse, configured as the compactor, delivers the query capacity to perform this. Therefore, when you size the Hive Virtual Warehouse that performs compaction, you must take into consideration the extra workload to run the compaction queries. That extra workload needs to be considered in addition to your other query workloads on the Hive Virtual Warehouse that is configured as the compactor.



**Important:** All compaction tasks for the warehouses that use the default Database Catalog, which is the Database Catalog automatically created for you when you activate an environment for CDW, are performed on CDP Base and do not affect the performance of Virtual Warehouses that use the default Database Catalog. For all other Database Catalogs that you create, you must consider the compaction query workload for the Hive Virtual Warehouse that performs compaction tasks.

### Before you begin

One of the Hive Virtual Warehouses must be configured as the compactor for the associated Database Catalog (excluding the default Database Catalog whose compaction is performed on CDP Base). This Hive Virtual Warehouse compactor runs all of the compaction queries for all Virtual Warehouses that use one particular Database Catalog, including Impala Virtual Warehouses. However, Impala Virtual Warehouses cannot be configured as the compactor Virtual Warehouse for a Database Catalog. Compaction tasks must be assigned to a Hive Virtual Warehouse. The first Hive Virtual Warehouse you create against a Database Catalog is automatically set as the compactor. If you decide you do not want that particular warehouse to take on the compaction workload, you can set another Hive Virtual Warehouse to perform the compaction workload by following these steps:

### Procedure

1. Log in to the CDP web interface and navigate to the Data Warehouse service.

- On the Overview page, select the Hive Virtual Warehouse that you want to set as the compactor, and click the

options menu



- In the options menu, select Set Compactor.

### Related Information

[CDW requirements for OpenShift](#)

## Debugging Impala Virtual Warehouses using Web UIs

You can use the Catalog Web UI, Coordinator Web UI, and the StateStore Web UI to debug Impala Virtual Warehouses in Cloudera Data Warehouse (CDW).

### About this task

The Impala daemons (impalad, statelord, and catalogd) debug Web UIs, which can be used in CDP Runtime by using Cloudera Manager, is also available in the CDW service. In CDW service, the following Web UIs are provided:

- **Impala Catalog Web UI** This UI provides the same type of information as the Catalog Server Web UI in Cloudera Manager. It includes information about the objects managed by the Impala Virtual Warehouse. For more information about this debug Web UI, see [Debug Web UI for Catalog Server](#) in the Impala Runtime documentation set.
- **Impala Coordinator Web UI** This UI provides the same type of information as the Impala Daemon Web UI in Cloudera Manager. It includes information about configuration settings, running and completed queries, and associated performance and resource usage for queries. For information about this debug Web UI, see [Debug Web UI for Impala Daemon](#) in the Impala Runtime documentation set.
- **Impala StateStore Web UI** This UI provides the same type of information as the StateStore Web UI in Cloudera Manager. It includes information about memory usage, configuration settings, and ongoing health checks that are performed by the Impala statelord daemon. For information about this debug Web UI, see [Debug Web UI for StateStore](#) in the Impala Runtime documentation set.

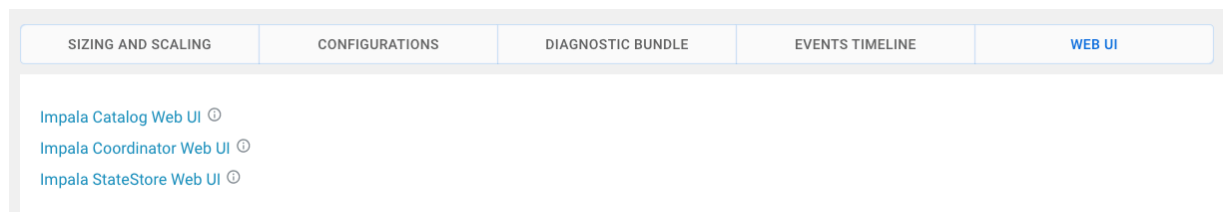
Required role: EnvironmentAdmin

### Before you begin

Make sure that you note your CDP workload user name and have set a password for it in the User Management module of Management Console. You need to use your workload user name and its associated password to log into the debug Web UIs.

### Procedure

- In the CDW UI on the Overview page, locate the Impala Virtual Warehouse for which you want to view the debug UIs, and select Edit from the options menu on the tile. This launches the details page for this Virtual Warehouse.
- In the **Virtual Warehouse** details page, select the WEB UI tab on the right. The list of debug Web UI links are displayed as shown in the following image:



- Click a Web UI link corresponding to an Impala daemon that you want to debug. You are prompted to enter your workload user name and password.

**Results**

After you are authenticated, you can view the debug Web UI and use the information to help you troubleshoot issues with your Impala Virtual Warehouse.

## Using Ozone storage with Cloudera Data Warehouse Private Cloud

The topics in this section describe how to use Apache Ozone storage with Cloudera Data Warehouse Private Cloud.

### Creating a database on Ozone for Cloudera Data Warehouse Private Cloud Virtual Warehouses

Learn how to create a database on Ozone storage that can be used by Cloudera Data Warehouse (CDW) Private Cloud Hive or Impala Virtual Warehouses.

**About this task**

By default, the Hive metastore for Database Catalogs on CDW Private Cloud points to HDFS, but you can configure the Database Catalog to point to Ozone storage instead by using the following steps. These steps change the default Hive metastore location to Ozone.


**Before you begin**

Before you re-configure the Database Catalog settings, make sure there are no running Virtual Warehouses associated with it. Either the Database Catalog has no associated Virtual Warehouses or you have suspended all the Virtual Warehouses associated with it.

## Procedure

### 1. Use the following steps to change the Database Catalog setting:

- From the Management Console Private Cloud home page left menu, navigate to Data Warehouse Overview.
- 

In the Database Catalog tile, click the edit icon .

- In the Database Catalogs detail page, click the Metastore tab, and select hive-site from the drop-down list on the left side of the tabbed page.
- Search for the following configuration properties and update them to Ozone file system paths, which start with o3fs:
  - hive.metastore.warehouse.dir
  - hive.metastore.warehouse.external.dir



**Note:** For the Hive Table creation, the warehouse directory must be set at bucket level or directory level under the hive.metastore.warehouse.dir or hive.metastore.warehouse.external.dir parameters. For more information, see [Changing the Hive warehouse location](#).

Here is an example of these properties set for the test-env-datalake-default Database Catalog:

The screenshot shows the Cloudera Data Warehouse Management Console interface. On the left is a sidebar with navigation links: Overview, Database Catalogs, and Virtual Warehouses. The main panel displays the 'Database Catalogs' section for the catalog 'test-env-datalake-default'. It shows the environment name as 'TEST-ENV', environment ID as 'ENV-L...', and database catalog ID as 'WAREHOUSE-15'. The version is '7.2.2.0-80' and the CPU count is '4'. The 'Metastore' tab is active. A dropdown menu is open, showing 'hive-site' selected. Below this, a table lists configuration properties. The first property is 'hive.metastore.warehouse.dir' with the value 'o3fs://bucket1.volume1.ozone1/test-env-datalake-default/managed'. The second property is 'hive.metastore.warehouse.external.dir' with the value 'o3fs://bucket1.volume1.ozone1/test-env-datalake-default/external'. Orange boxes highlight the dropdown menu and the property names. Orange arrows point to the values in the table.

The example values in the screenshot show the Hive warehouse locations in Ozone (set at a directory level) where Hive stores the tables. bucket1.volume1.ozone1 represents the Ozone volume, 'est-env-datalake-default' represents the Ozone bucket, and managed and external are directories where Hive stores the managed and external tables.

- Click Apply in the upper right corner of the page to save your settings. The Database Catalog begins updating.
- ### 2. After the Database Catalog has finished updating, perform one of the following actions to get started working with a Virtual Warehouse:
- Restart any associated Virtual Warehouses that you suspended before updating the Database Catalog properties by clicking the re-start icon in the upper right corner of the Virtual Warehouse tile on the Overview page.
  - [Create a new Hive or Impala Virtual Warehouse](#) associated with the updated Database Catalog.
- ### 3. Use Hue to create a database with your Virtual Warehouse. For details, see [Querying data](#).

## Results

After configuring the Database Catalog's Hive metastore to point to Ozone, you can create databases on Ozone with either an Impala or a Hive Virtual Warehouse.

## Configuring Hive/Impala logging on Ozone for Cloudera Data Warehouse Private Cloud

This section describes how to configure Cloudera Data Warehouse (CDW) on Private Cloud to store Hive and Impala logs on Ozone storage.

You can configure CDW to store Hive and Impala logs on CDP Private Cloud storage components, such as Ozone. Ozone is a good choice to store these logs because:

- Ozone efficiently handles files regardless of their size.
- In addition to Ozone's built-in CLI interface, Ozone also supports the HDFS CLI and CLIs that are compatible with AWS clients.
- CDP Private Cloud uses [fluentd](#) to push application logs to the storage layer. Ozone is a supported logging "back-end" component and has a fluentd-compatible endpoint for collecting the logs.

### Specify or create an Ozone bucket for Cloudera Data Warehouse Private Cloud logs

This topic describes how to specify an Ozone bucket to store Cloudera Data Warehouse (CDW) Private Cloud Hive and Impala logs.

#### About this task

You can either re-use the Ozone bucket that is automatically configured for storing Cloudera Machine Learning (CML) Private Cloud logs or create a new bucket to store CDW logs separately. The Ozone bucket used to store CML logs usually has a `cdplogs-` prefix.

#### Procedure

Use one of the following two methods depending on whether you want to use the existing CML log bucket or create a new one for CDW:

- To select an existing Ozone bucket, use the `ozone sh bucket list` command from the Ozone shell on your Private Cloud Base cluster. The following example shows how you can list buckets by the `cdplogs-` prefix:

```
ozone sh bucket list o3://ozone1/s3v --prefix=cdplogs
{
  "metadata" : { },
  "volumeName" : "s3v",
  "name" : "cdplogs-av-dwx-env-96c47aa9",
  "storageType" : "DISK",
  "versioning" : false,
  "creationTime" : "2020-08-01T18:29:08.686z",
  "modificationTime" : "2020-08-03T18:29:08.686z",
  "encryptionKeyName" : null,
  "sourceVolume" : null,
  "sourceBucket" : null
}
```

- To create a new bucket on Ozone, use the `ozone sh bucket create` command from the Ozone shell on your Private Cloud Base cluster. The following example shows how to create a new Ozone bucket named `cdw-logs-bucket`:

```
ozone sh bucket create o3://ozone1/s3v/cdw-logs-bucket
```



**Important:** Cloudera recommends that you use the `hive` user because this user automatically has create/read/write permissions on buckets that you create.

## Update Cloudera Data Warehouse Private Cloud log configuration to point to Ozone

This topic describes how to configure Cloudera Data Warehouse (CDW) Private Cloud to store logs on Ozone.

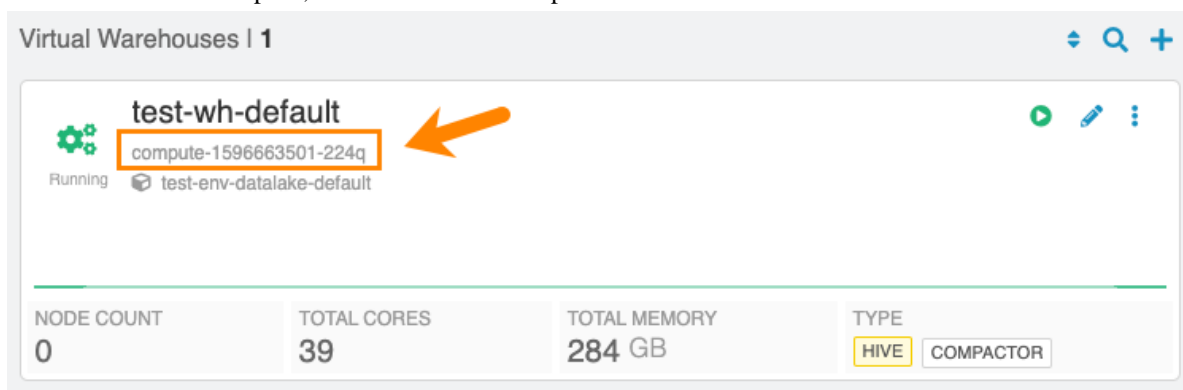
### About this task

To configure CDW Private Cloud and the underlying OpenShift cluster to store Hive and Impala logs on Ozone, you must gather some information and prepare a block of code that you will insert into the Virtual Warehouse ConfigMap on the OpenShift pod. These preliminary steps are described in the following section.

### Before you begin

Get the following information and prepare the block of code for the Virtual Warehouse ConfigMap before you start the steps of updating the configuration:

- Get the CDW namespace for your Virtual Warehouse:
  1. From the Management Console home page left menu, click Data Warehouse in the left menu. You are taken to the Overview page of CDW Private Cloud service.
  2. Locate the Virtual Warehouse you want to configure log storage for in the right-most column of the page, and locate the CDW namespace, which starts with compute- as shown below:



- Prepare the code block that must be pasted into the OpenShift ConfigMap:

Here is an example:

```
<match **>
  @type s3
  @log-level debug
  aws_key_id <access-id>
  aws_sec-key <sec-key>
  s3_bucket <bucket-name>
  s3_endpoint <ozone-s3-gateway-endpoint>
  ssl_verify_peer false
  s3_object_key_format
    "<warehouse_prefix>/warehouse/tablespace/external/hive/sys.db/logs
    /dt=%Y-%m-%d/${path_tag}/${time_slice}_${unique_file_key}.log.${file_ext
    ension}"
  time_slice_format %Y-%m-%d-%H-%M
  store_as gzip
  auto_create_bucket false
  check_apikey_on_start false
  force_path_style true
  check_bucket false
  check_object false
  <buffer path_tag, unique_file_key, time, warehouse>
  @type file
  path /tmp/fluentd-buffers/${unique_file_key}-s3.buffer
  timekey 900 # minute precision for time_slice_format to have minu
te in file name
```

```

timekey_use_utc true
chunk_limit_size 265m
flush_mode interval
flush_interval "900s"
flush_thread_count 8
flush_at_shutdown true
</buffer>
<format>
  @type single_value
  message_key log
  add_newline true
</format>
</match>

```

In the above code block example:

- `<bucket-name>` indicates the name of the Ozone bucket used for storing the CDW Private Cloud logs.
- `<ozone-s3-gateway-endpoint>` indicates the endpoint of the Ozone S3 Gateway. Get this value from the Ozone S3 Gateway Web UI page of Cloudera Manager.
- `<access_id>` and `<sec_key>` are the AWS access credentials for the Ozone S3 Gateway. Get these values by using the `kinit -kt` and the `ozone s3 getsecre` commands on the Private Cloud Base OpenShift cluster.

## Procedure

1. Using OpenShift commands, view the OpenShift project for the pod where the CDW Private Cloud instance is running by specifying the CDW namespace for the Virtual Warehouse that you noted in the [Before you begin](#) section above.

For example, if the CDW namespace is `compute-1596663501-224q`, you can view the OpenShift project with the following command:

```
oc project compute-1596663501-224q
```

2. Open the ConfigMap for the Virtual Warehouse that is associated with the CDW namespace. For example:

```
oc edit configmap warehouse-fluentd-config
```

This command opens the ConfigMap in a separate editor that is similar to `vi`.

3. Replace the match section of the ConfigMap with the code block you prepared in the [Before you begin](#) section above, and then save your changes
4. Verify that the new configuration is correctly updated by running the following command:

```
oc get namespace -o yaml | grep fluentd-status
```

If the configuration is successfully updated, the value of the `fluentd-status` returns an empty string as shown in the following example:

```

com.cloudera/fluentd-status: " "
com.cloudera/fluentd-status: " "
com.cloudera/fluentd-status: " "
com.cloudera/fluentd-status: " "

```

## Monitor Cloudera Data Warehouse Private Cloud logs on Ozone storage

This topic describes how to monitor Cloudera Data Warehouse (CDW) Private Cloud logs that are stored on Ozone.

### About this task

You can use either the Ozone S3 Gateway Web UI in Cloudera Manager or run commands in a terminal window to monitor CDW logs.



**Note:** Because fluentd buffers the logs and then pushes them to the configured endpoint, Ozone might take up to 15 minutes to display the CDW logs.

### Procedure

Use one of the following methods to monitor CDW logs in Ozone:

- Ozone S3 Gateway Web UI in Cloudera Manager:

Navigate to the following URL:

`https://<s3-gateway-endpoint>/<bucket-name>?browser=true`

Where:

- `<s3-gateway-endpoint>` indicates the endpoint of the Ozone S3 Gateway, which you can get from the Ozone S3 Gateway Web UI
- `<bucket-name>` indicates the Ozone bucket where you are storing the CDW logs.
- Run the following command from the Ozone shell: `ozone sh key list o3://<ozone.service.id>/s3v/<bucket-name>/ --prefix=<warehouse-prefix>`

Where:

- `<ozone.service.id>` indicates the identifier used for your implementation of Ozone.
- `<bucket-name>` indicates the name of the Ozone bucket where the CDW logs are stored.
- `<warehouse-prefix>` indicates the Virtual Warehouse identifier.

## Analyze Cloudera Data Warehouse Private Cloud logs stored on Ozone

This topic describes how to use Hue or Data Analytics Studio (DAS) to analyze Cloudera Data Warehouse (CDW) Private Cloud logs that are stored on Ozone.

### About this task

You can use Hue to analyze Impala logs or DAS to analyze Hive logs.



**Note:**

You must use the Hue or DAS instance that corresponds to the Virtual Warehouse whose logs are saved on Ozone. To ensure that you use the correct instance, access Hue or DAS by using the drop-down menu in the upper right corner of the Virtual Warehouse tile.

### Procedure

1. Using Hue or DAS, create an external table that points to the log data on Ozone:

```
CREATE EXTERNAL TABLE <table-name> LIKE sys.logs LOCATION 'o3fs://<bucket-name>.s3v.<ozone.service.id>/<warehouse-prefix>/warehouse/tablespace/external/hive/sys.db/logs';
```

2. Run the MSCK REPAIR TABLE command on the table you created in Step 1:

```
MSCK REPAIR TABLE <table-name>;
```

### Results

After completing the above steps, you can use SQL queries to analyze the log data.

## Locating Cloudera Data Warehouse Private Cloud logs

Learn how you can access logs for Cloudera Data Warehouse (CDW) Private Cloud.



### About this task

CDW Private Cloud logs are written to a partition on the Hive sys.logs table. These partitions are retained for 7 days by default.

### Procedure

1. Log in to the OpenShift or Experiences Compute Service (ECS) cluster and determine the location of the sys.logs table by running the following query:

```
DESCRIBE FORMATTED sys.logs;
```

This SQL statement returns information about the location of the table which contains the logs.

2. Use the location obtained in Step 1 to locate the CDW Private Cloud logs on the OpenShift or ECS clusters.

## Generating and downloading diagnostic bundles

Cloudera Data Warehouse (CDW) collects diagnostic data on workload logs, such as Impala Coordinator, Statefulset, CatalogD logs and stores it in the tmp directory on HDFS. You can download the logs using the Hue File Browser from the base cluster.

### About this task

During the lifetime of a cluster, logs are continuously written to the following directory on HDFS: [\*\*WAREHOUSE-DIR\*\*]/warehouse/tablespace/external/hive/sys.db/. When you click Collect Diagnostic Bundle from the CDW web interface, CDW collects the logs for the specified time interval and for the services that you select. These logs are compressed in a ZIP file format and stored in the tmp directory.

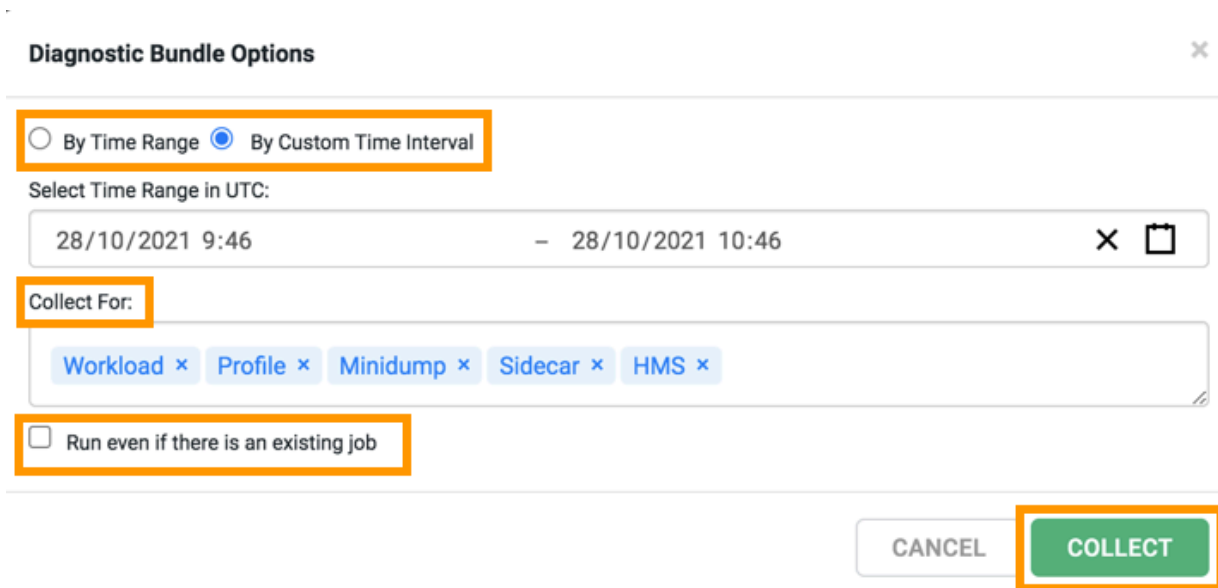


**Attention:** In 1.3.2 release of CDW Private Cloud, you can generate and download diagnostic bundles only for Impala.

### Procedure

1. Log in to the CDW service as a DWAdmin.
2. Click the options drop-down menu on the Virtual Warehouse for which you want to collect the logs and click Collect Diagnostic Bundle.

- On the **Diagnostic Bundle Options** dialog box, select the time interval and the type of logs you want to collect and click COLLECT.



**Diagnostic Bundle Options**

☐ By Time Range
 ☒ By Custom Time Interval

Select Time Range in UTC:

28/10/2021 9:46 – 28/10/2021 10:46

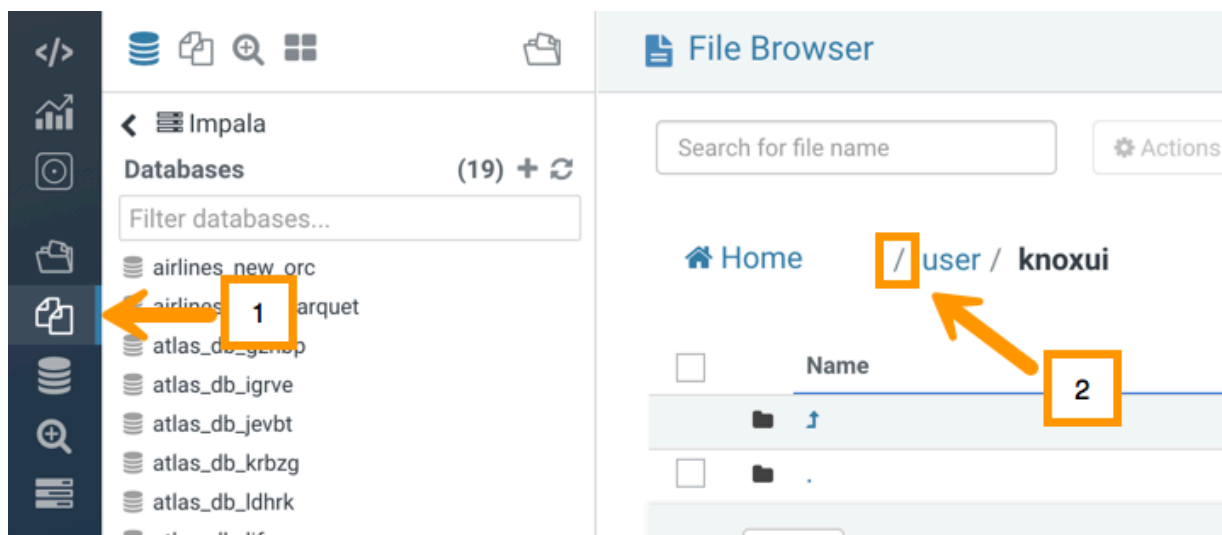
Collect For:

Workload × Profile × Minidump × Sidecar × HMS ×

☐ Run even if there is an existing job

CANCEL COLLECT

- To view the status of the job and to obtain the HDFS location where the logs are stored, select Edit from the Virtual Warehouse options menu and go to the DIAGNOSTIC BUNDLE tab. The logs are collected and bundled under the /tmp/[\*\*\*VIRTUAL-WAREHOUSE-ID-TIMESTAMP\*\*\*].zip directory.
- To access and download the logs, open the Hue service from the base cluster.
- Go to the Hue File Browser and click the forward slash (/) before the user directory as shown in the following image:



The tmp directory is displayed. You can access and download the logs to your computer by clicking Download.

## Configuring Impala Virtual Warehouses to create Impala tables in Kudu in Cloudera Data Warehouse Private Cloud

Cloudera Data Warehouse allows you to create Impala tables in Kudu. You can configure an Impala Virtual Warehouse to connect to Kudu and create Impala tables in Kudu using Hue. Or, you can create tables on the fly by

specifying the Kudu master host in the TBLPROPERTIES statement while running the query from the Hue query editor.

### About this task



**Attention:** This feature is in technical preview and not recommended for use in production environments.

### Before you begin

Obtain the hostname of the Kudu master home by going to Cloudera Manager Clusters Kudu service Instances from the CDP Management Console.

### Creating Impala tables in Kudu on the fly



To create Impala tables in Kudu without updating a Virtual Warehouse's Impala coordinator configuration, you must specify the Kudu master host in the TBLPROPERTIES statement as follows while running the query from Hue:

```
TBLPROPERTIES ('kudu.master_addresses'='[***host.example.com***]')
```

### Configuring the Virtual Warehouse to create Impala tables in Kudu

By reconfiguring an existing Impala Virtual Warehouse as follows, any tables you create will be created in Kudu.

#### Procedure

1. Log in to the Cloudera Data Warehouse service as a DWAdmin.
2. Go to an Impala Virtual Warehouse and click  Edit CONFIGURATIONS Impala coordinator and select flagfile from the drop-down list.
3. Click  and enter the following key and value:

Key	Value
kudu_master_hosts	[***HOSTNAME-OF-KUDU-MASTER***]

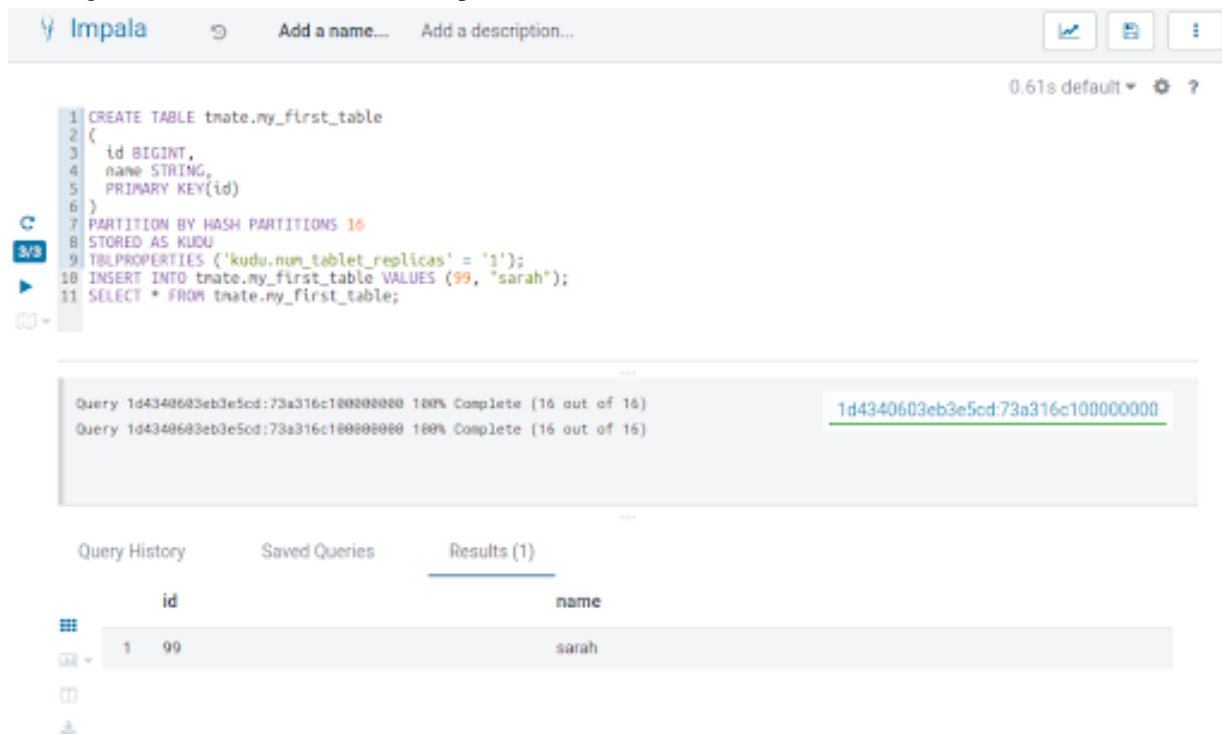
4. Click APPLY.
5. Restart the Virtual Warehouse.
6. Open Hue from the same Virtual Warehouse.
7. Enter the following lines in the query editor and click the run button:

```
# Create a new table
# Use the kudu.num_tablet_replicas if the Kudu cluster is too small
CREATE TABLE my_first_table
(
  id BIGINT,
  name STRING,
  PRIMARY KEY(id)
)
PARTITION BY HASH PARTITIONS 16
STORED AS KUDU
TBLPROPERTIES ('kudu.num_tablet_replicas' = '1');
# Insert into Kudu table
INSERT INTO my_first_table VALUES (99, "sarah");

# Verify if the data was inserted
```

```
SELECT * FROM my_first_table;
```

The above commands create an Impala table in Kudu and insert a sample record. The following is a screenshot showing the SQL commands and their output in Hue:



The screenshot shows the Hue interface with the following SQL commands and their output:

```
1 CREATE TABLE tnate.my_first_table
2 (
3   id BIGINT,
4   name STRING,
5   PRIMARY KEY(id)
6 )
7 PARTITION BY HASH PARTITIONS 16
8 STORED AS KUDU
9 TBLPROPERTIES ('kudu.num_tablet_replicas' = '1');
10 INSERT INTO tnate.my_first_table VALUES (99, "sarah");
11 SELECT * FROM tnate.my_first_table;
```

The output shows the table structure and the inserted record:

```
Query 1d4340603eb3e5cd:73a316c100000000 100% Complete (16 out of 16)
Query 1d4340603eb3e5cd:73a316c100000000 100% Complete (16 out of 16)
```

id	name
99	sarah

## Configuring Impala Virtual Warehouses to encrypt spilled data in Cloudera Data Warehouse Private Cloud

If you have encrypted HDFS on the base CDP cluster, then Cloudera recommends that you configure an Impala Virtual Warehouse to write temporary data to disk during query processing in an encrypted format using the AES-256-CFB encryption for complete security.


### About this task

In CDP Private Cloud, the temporary data is spilled to the local storage, the location of which is hard coded by the system.



**Important:** Impala does not selectively encrypt data based on whether the source data is already encrypted in HDFS. This results in at most 15 percent performance degradation when data is spilled.

### Procedure

1. Log in to the Cloudera Data Warehouse service as an administrator.
2. Go to Impala Virtual Warehouse  Edit CONFIGURATIONS Impala coordinator and select flagfile from the Configuration files drop-down list.
3. Set the value of the `disk_spill_encryption` property to true.
4. Click APPLY.
5. Go to the Impala executor tab and select flagfile from the Configuration files drop-down list.
6. Set the value of the `disk_spill_encryption` property to true.

7. Click APPLY.
8. Restart the Impala Virtual Warehouse.

## Creating custom pod configurations for Impala Virtual Warehouses

You can configure the resources used by Impala Virtual Warehouses in Cloudera Data Warehouse (CDW) Private Cloud environments to optimize Impala performance or to control resource usage in the environment.


### About this task

When you create a Virtual Warehouse, CDW allocates standard resources to the Warehouses that are suitable for most workloads. You can control the size of the Virtual Warehouse at the time of creation by choosing the number of nodes to be used. By using custom pod configurations, you can also change the resources used by the critical Impala components, such as the coordinators, executors, and catalog daemons to pack a particular number of pods into a Kubernetes node or to create extra-large daemons to handle specific workloads.



**Important:** This is a preview feature and not recommended for use in production environments. Cloudera recommends that you explore this feature in development or test environments.

### Procedure

1. Log in to the Data Warehouse service as a DWAdmin.
2. Go to your environment and click  Edit .  
The **Environment Details** page is displayed.
3. Click the EDIT POD CONFIGURATIONS tab.  
A pod configuration is a named resource that is configured at the environment level.

4. Select one of the following two pod configuration options from the Select Pod Configuration section:
- The Cdw Defaults option is selected by default. CDW uses default values for the pods if a specific pod configuration is not used.
  - Select the 1 x Node option for allocation of most node resources found in the environment, to the Impala executors and coordinators.

The following image shows the two default pod configurations:

Environment Details

**Environment Name:** [REDACTED]-env-1 (ID: env-k87knp)

STATUS	VERSION	CREATED BY	DATABASE CATALOGS	VIRTUAL WAREHOUSES
<span style="color: green;">✓</span> Running	1.4.0-b11	<span style="background-color: #ccc; padding: 2px;">[REDACTED]</span>	1	1

[GENERAL DETAILS](#)
[CONFIGURATIONS](#)
[EDIT POD CONFIGURATIONS](#)

**env-k87knp: 2 pod configurations (2 readonly, 0 mutable)**

Select Pod Configuration		
<input type="checkbox"/> Name	Description	Default
<input type="checkbox"/> 1 x Node	Machine generated estimates for using all resources of a Node	
<input checked="" type="checkbox"/> Cdw Defaults	The Default CDW Configuration	yes

5. Click COPY CONFIG to create and edit a new configuration with the option that you selected earlier as the basis.
- Specify the name for your configuration in the Cloned Config Name field.
  - Enter a description for the new configuration in the Description field.

**Pod Configuration: Cdw Defaults (readonly)**

Description: The Default CDW Configuration

☒ Set as default configuration

[DELETE CONFIG](#)

[COPY CONFIG](#)

Cloned Config Name: Name of new config Description for new config: Description for new confi [CREATE NEW CONFIG](#)

- Click CREATE NEW CONFIG and then click APPLY under the **EDIT POD CONFIGURATION** tab. A new pod configuration is created, which you can now customize.

6. Specify the values for the following parameters under the Coordinator section:

- Memory
- CPU
- Xmx (maximum memory allocation pool for a Java Virtual Machine)
- Xms (initial memory allocation pool for a Java Virtual Machine)
- AC Slots (admission\_control\_slots flag)
- Cache (size of the data cache)
- Scratch (limit of Impala scratch space)

### Coordinator

Memory	<input type="text" value="112640"/>
Cpus	<input type="text" value="14"/>
Xmx	<input type="text" value="25G"/>
Xms	<input type="text" value="2G"/>
AC Slots	<input type="text" value="128"/>
Cache	<input type="text" value="/opt/impala/cache:300M"/>
Scratch	<input type="text" value="{{ .Values.scratchDirs }}"/>

7. Specify the values for the following parameters under the Executor section:

- Memory
- CPU
- Xmx (maximum memory allocation pool for a Java Virtual Machine)
- Xms (initial memory allocation pool for a Java Virtual Machine)
- AC Slots (admission\_control\_slots flag)
- Cache (size of the data cache)
- Scratch (limit of Impala scratch space)

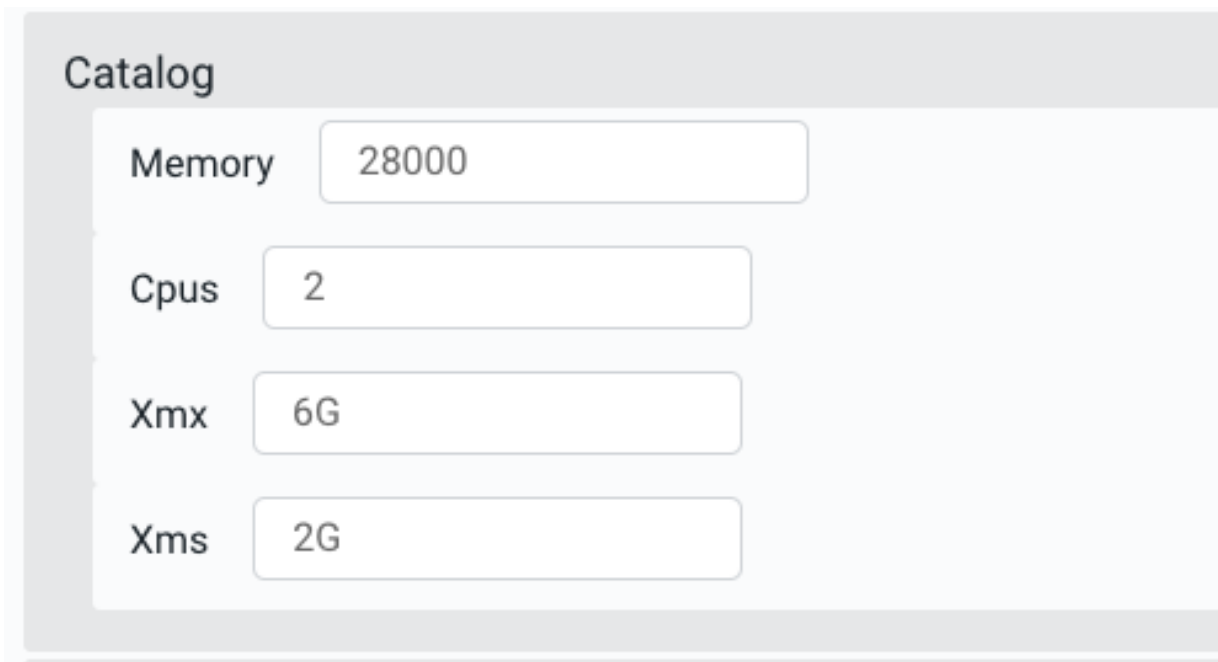
### Executor

Memory	<input type="text" value="116736"/>
Cpus	<input type="text" value="14"/>
Xmx	<input type="text" value="4G"/>
Xms	<input type="text" value="2G"/>
AC Slots	<input type="text" value="36"/>
Cache	<input type="text" value="/opt/impala/cache:3000"/>
Scratch	<input type="text" value="{{ .Values.scratchDirs }}"/>



8. Specify the values for the following parameters under the Catalog section:

- Memory
- CPU
- Xmx (maximum memory allocation pool for a Java Virtual Machine)
- Xms (initial memory allocation pool for a Java Virtual Machine)

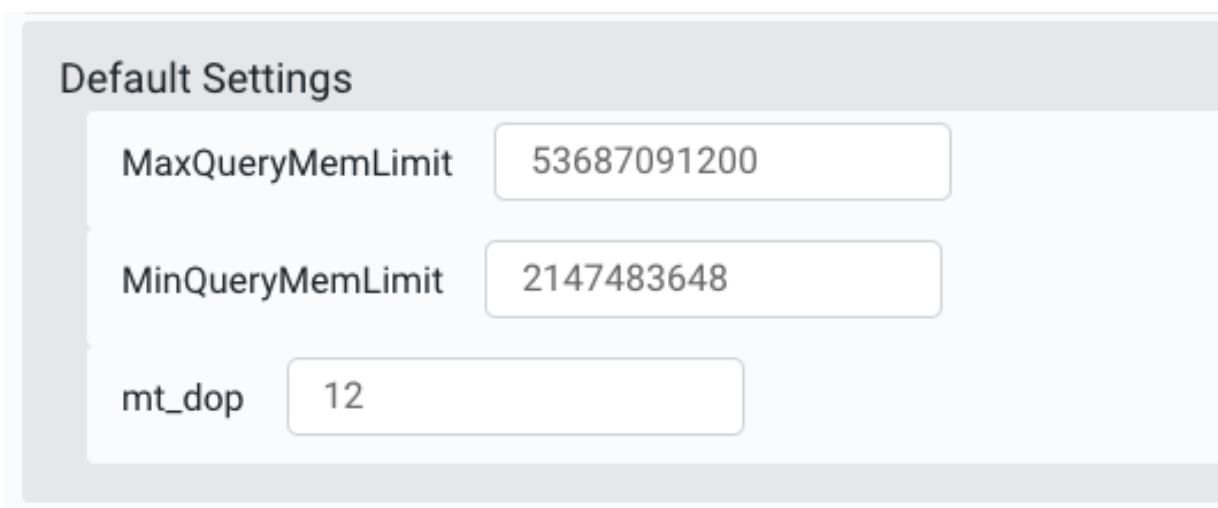


**Catalog**

Memory	28000
Cpus	2
Xmx	6G
Xms	2G

9. Specify the values for the following parameters under the Default Settings section:

- MaxQueryMemLimit
- MinQueryMemLimit
- mt\_dop



**Default Settings**

MaxQueryMemLimit	53687091200
MinQueryMemLimit	2147483648
mt_dop	12

10. Click APPLY under the **EDIT POD CONFIGURATION** tab to save the custom settings.

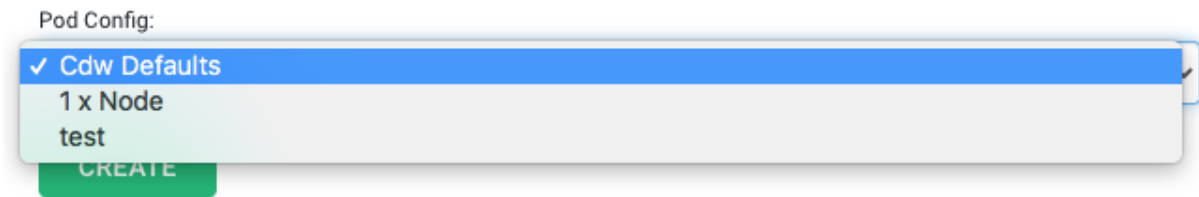
The “Configuration update initiated” message is displayed.

11. Click the Set as default configuration toggle button to make this a default pod configuration.

This makes a pod configuration the default configuration at the environment level.

**12.** Click APPLY at the top of the Environment Details page.

The new pod configuration becomes available in the Pod Config drop-down menu as shown in the following image. You can select this Impala pod configuration while creating a new Impala Virtual Warehouse:

**Results**

While adding a new Impala Virtual Warehouse, you can select the Pod Configuration to be used for resource allocation. The default value is "Cdw Defaults", but you can select other configurations available in your environment that you created using these steps.