

Planning and setting up CDW on Private Cloud

Date published: 2020-08-17

Date modified: 2024-10-18



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

| | |
|---|-----------|
| Plan and setup CDW..... | 4 |
| Requirements..... | 4 |
| Low resource requirements..... | 4 |
| Standard resource requirements..... | 6 |
| Security requirements for Cloudera Data Warehouse Private Cloud..... | 7 |
| Port requirements for AD..... | 8 |
| Database requirements..... | 8 |
| User roles and other prerequisites..... | 9 |
| Pod placement policy and rack awareness in CDW Private Cloud..... | 10 |
| Activate OpenShift environments..... | 10 |
| Activate ECS environments..... | 12 |
| Create first Virtual Warehouse..... | 13 |
| Set up Data Viz..... | 14 |
| Set up SQL AI Assistant..... | 15 |
| Preparing the Microsoft Azure OpenAI service..... | 15 |
| Preparing the Amazon Bedrock Service..... | 15 |
| Preparing the OpenAI platform..... | 16 |
| Prerequisites for configuring Hue SQL AI Assistant..... | 17 |
| (Recommended) Secure approach for passing a token to configure SQL AI Assistant in Cloudera Data Warehouse..... | 17 |
| Open approach for passing a token to configure Hue SQL AI Assistant..... | 18 |
| Configure SQL AI Assistant in CDW..... | 18 |
| Service and model-related configurations for setting up the Hue SQL AI Assistant..... | 19 |

Planning and setting up CDW Private Cloud

As a Cloudera Data Warehouse (CDW) Administrator on CDP Private Cloud, learn what the CDW hardware requirements are, how to deploy CDW, and understand the various interfaces and clients that you can use to access CDW.

- Review the hardware, security, and database requirements for deploying CDW.
- Create the required CDP resource roles such as DWAdmin and DWUser.
- Activate your environment in CDW.
- Create your first Virtual Warehouse.

Requirements for deploying Cloudera Data Warehouse on Private Cloud

Review the hardware requirements for deploying Cloudera Data Warehouse (CDW) in low and standard resource modes, security and database requirements, user roles required to access and administer CDW. Also learn about the pod placement policy and how CDW applies rack awareness rules.

Low resource mode requirements

Review the memory, storage, and hardware requirements for getting started with the Cloudera Data Warehouse (CDW) service in low resource mode on Red Hat OpenShift and Embedded Container Service (ECS). This mode reduces the minimum amount of hardware needed.

To get started with the CDW service on Red Hat OpenShift or ECS low resource mode, make sure you have fulfilled the following requirements:



Important: Lowering the minimum hardware requirement reduces the up-front investment to deploy CDW on OpenShift or ECS pods, but it does impact performance. Cloudera recommends that you use the Low Resource Mode option for proof of concept (POC) purposes only. This feature is not recommended for production deployment.

Complex queries and multiple queries on HS2 may fail due to limited memory configurations for HMS and HS2 in the low resource mode.

- CDP Cloudera Manager must be installed and running.
- CDP Private Cloud must be installed and running. See [Installing on OpenShift](#) and [Installing on ECS](#) for more details.
- An environment must have been registered with Management Console on the private cloud. See [CDP Private Cloud Environments](#) for more details.
- In addition to the general requirements, CDW also has the following minimum memory, storage, and hardware requirements for each worker node using the standard resource mode:

| Component | Low resource mode deployment |
|-------------------|--|
| Nodes | 4 |
| CPU | 4 |
| Memory | 48 GB |
| Storage | 3 x 100 GB (SATA) or 2 x 200 GB (SATA) |
| Network Bandwidth | 1 GB/s guaranteed bandwidth to every CDP Private Cloud Base node |



Important: When you add memory and storage for low resource mode, it is very important that you add it in the increments stated in the above table:

- increments of 48 GB of memory
- increments of at least 100 GB or 200 GB of SATA storage

If you add memory or storage that is not in the above increments, the memory and storage that exceeds these increments is not used for executor pods. Instead, the extra memory and storage can be used by other pods that require fewer resources.

Virtual Warehouse low resource mode resource requirements

The following requirements are in addition to the low resource mode requirements listed in the previous section.

Table 1: Impala Virtual Warehouse low resource mode requirements

| Component | vCPU | Memory | Local Storage | Number of pods in XSMALL Virtual Warehouse |
|------------------------------------|---------|-----------|--------------------|--|
| Coordinator (2) | 2 x 0.4 | 2 x 24 GB | 2 x 100 GB | 2 |
| Executor (2) | 2 x 3 | 2 x 24 GB | 2 x 100 GB | 2 |
| Statestore | 0.1 | 512 MB | -- | 1 |
| Catalogd | 0.4 | 16 GB | -- | 1 |
| Auto-scaler | 0.1 | 1 GB | -- | 1 |
| Hue (backend) | 0.5 | 8 GB | -- | 1 |
| Hue (frontend) | -- | -- | -- | 1 |
| Total for XSMALL Virtual Warehouse | 8 (7.9) | 121.5 GB | 400 GB - 3 volumes | -- |

Impala Admission Control Configuration

- Maximum concurrent queries per executor: 4
- Maximum query memory limit: 8 GB

Table 2: Hive Virtual Warehouse low resource mode requirements

| Component | vCPU | Memory | Local Storage | Number of pods in XSMALL Virtual Warehouse |
|--------------------------------------|------------------|--|--------------------|--|
| Coordinator (2) | 2 x 1 | 2 x 4 GB | 2 x 100 GB | 2 |
| Executor (2) | 2 x 4 | 2 x 48 GB (16 GB heap; 32 GB off-heap) | 2 x 100 GB | 2 |
| HiveServer2 | 1 | 16 GB | -- | 1 |
| Hue (backend) | 0.5 | 8 GB | -- | 1 |
| Hue (frontend) | -- | -- | -- | 1 |
| Standalone compute operator | 0.1 | 100 MB (.1 GB) | -- | -- |
| Standalone query executor (separate) | Same as executor | Same as executor | Same as executor | -- |
| Total for XSMALL Virtual Warehouse | 21 (20.6) | 237 GB (236.1) | 400 GB - 4 volumes | -- |

Database Catalog low resource mode requirements

The HiveMetaStore (HMS) requires 2 CPUs and 8 GB of memory. Because HMS pods are in High Availability mode, they need a total of 4 CPUs and 16 GB of memory.

Data Visualization low resource requirements

Table 3: Data Visualization low resource mode requirements

| vCPU | Memory | Local Storage | Number of pods in XSMALL Virtual Warehouse |
|------|--------|---------------|--|
| 0.5 | 8 GB | -- | 1 |

Standard resource mode requirements

Review the memory, storage, and hardware requirements for getting started with the Cloudera Data Warehouse (CDW) service in standard resource mode on Red Hat OpenShift and Embedded Container Service.

To get started with the CDW service on standard resource mode, make sure you have fulfilled the following requirements:

- CDP Cloudera Manager must be installed and running.
- CDP Private Cloud must be installed and running. See [Installing on OpenShift](#) and [Installing on ECS](#) for more details.
- An environment must have been registered with Management Console on the private cloud. See [CDP Private Cloud Environments](#) for more details.
- In addition to the general requirements, CDW also has the following minimum memory, storage, and hardware requirements for each worker node using the standard resource mode:

Depending on the number of executors you want to run on each physical node, the per-node requirements change proportionally. For example, if you are running 3 executor pods per physical node, you require 384 GB of memory and approximately 1.8 TB of locally attached SSD/NVMe storage.

The following table lists the minimum and recommended compute (processor), memory, storage, and network bandwidth required for each OpenShift or ECS worker node using the Standard Resource Mode for production use case. Note that the actual node still needs some extra resources to run the operating system, Kubernetes engine, and Cloudera Manager agent on ECS.

| Component | Minimum | Recommended |
|--|---|---|
| Node Count | 4 | 10 |
| CPU per worker | 16 cores [or 8 cores or 16 threads that have Simultaneous Multithreading (SMT) enabled] | 32+ cores (can also be achieved by enabling SMT) |
| Memory per worker | 128 GB per node | 384 GB* per node |
| FAST (Fully Automated Storage Tiering) Cache - Locally attached SCSI device(s) on every worker. Preferred: NVMe and SSD. OCP uses Local Storage Operator. ECS uses Local Path Provisioner. | 1.2 TB* SATA, SSD per host | 1.2 TB* NVMe/SSD per host |
| Network Bandwidth | 1 GB/s guaranteed bandwidth to every CDP Private Cloud Base node | 10 GB/s guaranteed bandwidth to every CDP Private Cloud Base node |

* Depending on the number of executors you want to run on each physical node, the per-node requirements change proportionally. For example, if you are running 3 executor pods per physical node, you require 384 GB of memory and approximately 1.8TB (600GB per executor) of locally attached SSD/NVMe storage for FAST Cache.



Important: When you add memory and storage, it is very important that you add it in the increments as follows:

- Increments of 128 GB of memory
- Increments of 600 GB of locally attached SSD/NVMe storage

If you add memory or storage that is not in the above increments, the memory and storage that exceeds these increments is not used for executor pods. Instead, the extra memory and storage can be used by other pods that require fewer resources.

For example, if you add 200 GB of memory, only 128 GB is used by the executor pods. If you add 2 TB of locally attached storage, only 1.8 TB is used by the executor pods.

Related Information

[Hyper-Threading](#)

Security requirements for Cloudera Data Warehouse Private Cloud

This topic describes security requirements needed to install and run Cloudera Data Warehouse (CDW) Private Cloud service on Red Hat OpenShift and Embedded Container Service (ECS) clusters.

Required OpenShift/ECS cluster permissions

The CDW service requires the "cluster-admin" role on the OpenShift and ECS cluster in order to install correctly. The "cluster-admin" role enables namespace creation and the use of the OpenShift Local Storage Operator for local storage.

CDP Private Cloud LDAP certificate requirement

A certificate authority (CA) certificate for secure LDAP must be uploaded to the Administration page of Management Console to run CDW Private Cloud service:

CLUSTER Administration

Diagnostic Data Authentication

Local Administrator Account

Change Password

External Authentication

* LDAP URL ⓘ

CA Certificate for Secure LDAP ⓘ

☒ File Upload ☐ Direct Input

Choose File

☒ Use Bind DN and Password ☐ Use Anonymous Bind

Port requirements for AD in Cloudera Data Warehouse Private Cloud

Review the ports that you must use for Active Directory (AD) in Cloudera Data Warehouse (CDW) Private Cloud. Cloudera recommends that you use AD Global Catalog ports 3268 and 3269 if you are using LDAP referrals.

In CDW, neither Hive nor Impala can use the standard LDAP referrals. Therefore, you cannot use the standard LDAP ports “389” and “636” for TLS/SSL with AD. Instead, you must use Active Directory Global Catalog ports “3268” and “3269” for TLS/SSL.



Note: If you specify the standard LDAP ports for AD in the LDAP URL in Management Console, then you may see the following error messages in the Hive and Impala logs when you try to access Hive or Impala Virtual Warehouses using remote clients such as Beeline, Impala-shell, and so on:

- (Hive Virtual Warehouse):

```
"javax.naming.PartialResultException: Unprocessed Continuation Reference"
```

- (Impala Virtual Warehouse):

```
Following of referrals not supported
```

or

```
LDAP search failed with base DN=<REDACTED> and filter=<REDACTED> :  
Operations error
```

CDW performs port validation when you activate an environment in CDW. The validation process only indicates a problem if you have configured AD, but you have not included a port in the LDAP URL in the Management Console. In this scenario, the Database Catalog does not reach the Ready state, and you see the following error:

```
Active Directory servers should be used through the Global Catalog ports:  
3268/3269
```

If you specify any port number in the LDAP URL, then no error message is displayed.

Base cluster database requirements for Cloudera Data Warehouse Private Cloud

You must be aware of the requirements for the database that is used for the Hive Metastore on the base cluster (Cloudera Manager side) for Cloudera Data Warehouse (CDW) Private Cloud.

CDW supports MariaDB, MySQL, PostgreSQL, and Oracle databases for the Hive Metastore (HMS) on the base CDP cluster (Cloudera Manager side). On a default Database Catalog, Hue and HMS use an embedded PostgreSQL database that is defined when you install CDP Private Cloud.



Note: Cloudera recommends that you use an embedded database for the HMS and the Control Plane service. You can use the Data Recovery Service for backing up and restoring Kubernetes namespaces behind CDW entities (Database Catalogs and Virtual Warehouses).

If you are using PostgreSQL, MySQL, MariaDB, or Oracle database for the Hive Metastore on the base cluster, then it must meet the following requirements:

- SSL-enabled.
- Uses the same keystore containing an embedded certificate as Ranger and Atlas.

To use the same keystore with an embedded certificate for Ranger and Atlas:

- If you are using Auto-TLS:

In the Management Console **Administration** page, go to the **CA Certificates** tab and select External Database from the CA Certificate Type drop-down menu. Upload the CA certificates either by uploading a file or by direct input.

- If you are not using Auto-TLS:

Ensure that the public certificate of the certificate authority (CA) that signed the Hive metastore database's certificate is present in Cloudera Manager's JKS truststore. If the certificate is self-signed, import that certificate into Cloudera Manager's JKS truststore: In the Management Console Administration page, find the path to Cloudera Manager's JKS truststore by navigating to Administration Settings Security Cloudera Manager TLS/SSL Client Trust Store File . Import the CA's certificate into that JKS file.

To add the certificate name to an existing or a new JKS file, use the following keytool command, which uses the same example certificate name:

```
keytool -import -alias postgres -file /path/to/postgres.pem -storetype JKS -keystore /path/to/cm.jks
```

Where /path/to/cm.jks is the JKS file that is configured by Cloudera Manager.

This ensures that the file specified for Cloudera Manager TLS/SSL Client Trust Store File is passed to Management Console and workloads.



Note: If you have a JRE11 keystore you must convert it to a JRE8 keystore using the following keytool command:

```
keytool -importkeystore -srckeystore
      <path-to-my-pfx-file.pfx> -srcstoretype pkcs12 -srcstore
pass
      <***password***> -destkeystore
      <path-to-client-certificate.jks> -deststoretype JKS
      -deststorepass <***password***>
```

CDP resource roles and other prerequisites

To get started in Cloudera Data Warehouse (CDW), your data must conform to supported compression codecs, and you must obtain CDP resource roles to grant users access to a private cloud environment. Users can then get started on CDW tasks, such as activating the environment from CDW.

Unsupported compression

CDW does not support LZO compression due to licensing of the LZO library. You cannot query tables having LZO compression in Virtual Warehouses, which use CDW Impala or Hive LLAP engines.

CDP resource roles

Required role: PowerUser

The following CDP resource roles are associated with the CDW service. A CDP PowerUser must assign these roles to users who require access to the Database Catalogs and Virtual Warehouses that are associated with specific environments. After granting these roles to users and groups, they then have access to the Data Catalogs and Virtual Warehouses that are associated with the environment.

- **DWAdmin:** This role enables users or groups to grant a CDP user or group the ability to activate, terminate, launch, stop, or update services in Database Catalogs and Virtual Warehouses.
- **DWUser:** This role enables users or groups to view and use CDW clusters (Virtual Warehouses) that are associated with specific environments.

Requirements for Hue

Hue in CDW requires WebHDFS to be enabled on the CDP Private Cloud Base cluster. Worker nodes for both, Embedded Container Service (ECS) and OpenShift Container Platform (OCP), must have access to the WebHDFS (HTTPFS) port 14000.

Recommended HAProxy timeout for HA deployments

If you have enabled High Availability (HA) for CDP Private Cloud Data Services on ECS or OCP, then set the HAProxy timeout values to 10 minutes or more, depending on how long your queries run. Setting a higher timeout value is needed to support long-running queries and prevent timeouts.

Related Information

[Understanding roles in CDP Private Cloud Data Services](#)

Pod placement policy and rack awareness in CDW Private Cloud

In Cloudera Data Warehouse (CDW), Kubernetes node affinity rules are based on the rack topology defined in Cloudera Manager. If sufficient resources are available, CDW prefers the same racks for scheduling HiveServer2 (HS2), executor, and coordinator pods in Impala and Hive Virtual Warehouses.

If two executors are present on different nodes in different racks and are connected using a network interconnect, then this can cause significant performance overhead. It is a best practice to schedule executors into the same rack.

On Embedded Container Service (ECS), the Kubernetes nodes are tagged with the label `rack=[***RACK-ID***]`. You can specify a 63-character long rack ID. Only alphanumeric characters are supported. On OpenShift Container Platform (OCP), you can tag the nodes yourself as needed.

Every time CDW needs to schedule new executor pods, the executor pods are scheduled next to the existing Hive or Impala executors, coordinators, and HS2. In other words, HS2 and Impala coordinators have an affinity to the first executor group within a particular rack.

In the case of race conditions where multiple executors get scheduled at once but in different racks, the rack with the highest number of executors attracts the rest of the executors. However, it is still possible to schedule executor groups across multiple racks in case of race conditions or if the first rack is full and no executors can fit into it.

Related Information

[Specifying racks for ECS clusters](#)

Activating OpenShift environments

This topic describes how to activate an environment to use for Cloudera Data Warehouse (CDW) Private Cloud on Red Hat OpenShift Container Platform (OCP).

About this task

Before you can create a Database Catalog to use with a Virtual Warehouse, you must activate a CDP environment. Activating an environment causes CDP to connect to the Kubernetes cluster, which provides the computing resources for the Database Catalog. In addition, activating an environment enables the Cloudera Data Warehouse (CDW) service to use the existing data lake that was set up for the environment, including all data, metadata, and security.

Before you begin

- Determine which environment that uses a particular data lake is the environment you want to activate for use with a Database Catalog and Virtual Warehouse.
- For local caching, ensure that an administrator uses the Local Storage Operator to create a local file system on an SSD/NVMe for each OpenShift worker node and then mounts it to a known location on the worker node. Make

sure that this local caching location allows temporary data to be stored in a way that supports performance. You need to specify the Storage Class Name from the Local Storage Operator when you activate the environment for the CDW service in Step 4 below. For more information about creating a local file system on OpenShift worker nodes using the Local Storage Operator, see [Persistent storage using local volumes](#) in the OpenShift documentation.

- (Optional) Go to **Advanced Configuration Advanced Settings** and enable the **Use deterministic namespace names** option to use deterministic namespaces for Kerberos principals and keytabs. You cannot enable this option after activating an environment.
- (Optional) Go to **Advanced Configuration Advanced Settings** and enable the **Create databases for Virtual Warehouses** option if you are upgrading the CDP Private Cloud Data Services platform from an older release to the latest release, and you want to continue using external database for Hue and HMS. You cannot enable this option after activating an environment.
- (Optional) Go to **Advanced Configuration Advanced Settings** and turn off cluster validation by selecting the **Skip cluster validation during environment activation** option. By selecting this option, you can proceed with the environment activation even after seeing false positive errors in the CDW logs. Cluster validation includes port validation, and the Kerberos keytab configuration validation, and Root CA certificate validation for Impala Virtual Warehouses.



Note: If you have more than one OCP clusters managed using different instances of Cloudera Manager, but using the same AD server and using the same environment name, then go to the **Advanced Configuration Advanced Settings** page and ensure that the **Use deterministic namespace names** option is disabled before activating the environment in CDW.

Procedure

1. Log in to Data Warehouse service as DWAdmin.
2. Click on the Environments tab.
3. Locate the environment you want to activate and click **Activate**.

The **Activate Environment** dialog box is displayed.

4. Specify the Storage Class Name from Local Storage Operator:

This is the Storage Class Name you specified when you created the local file system for caching as described in the [Before you begin](#) section. It is the location where temporary data is stored.



Important: Be sure to specify the correct Storage Class Name when activating an environment. If an incorrect Storage Class Name is specified, the environment might activate successfully, but Virtual Warehouses that use the environment do not start.

Optionally, you can specify the Security Context Constraint Name.

5. **(To use mTLS)** Browse and upload the database client certificate and database client private key files in PEM format.

The client certificate and private key files must be in PEM format.

6. Enable low resource mode to deploy CDW on minimum hardware.



Note: Cloudera recommends that you use the Low Resource Mode option for proof of concept (POC) purposes only. This feature is not recommended for production deployment.

Complex queries and multiple queries on HS2 may fail due to limited memory configurations for HMS and HS2 in the low resource mode.

7. Enable the **Use dedicated nodes for executors** option to schedule Hive and Impala executor and coordinator pods on the worker nodes tainted for CDW.
8. If you are using an external database on your base cluster and want to use a default Database Catalog, then you must specify custom database name for Hue in the **Pre-created database names for default database catalog** field.
9. Select the quota-managed resource pool from the Resource Pool drop-down menu.

The Resource Pool drop-down menu is displayed only if you have enabled the quota management feature from Advanced Configurations.

10. Click Activate.

Related Information

[Advanced Configuration in CDW Private Cloud](#)

[How predefined Kerberos principals are used in CDW Private Cloud](#)

[Enabling quota management in CDW Private Cloud](#)

Activating Embedded Container Service environments

This topic describes how to activate an environment to use for Cloudera Data Warehouse (CDW) Private Cloud on Embedded Container Service (ECS).

About this task

Before you can create a Database Catalog to use with a Virtual Warehouse, you must activate a CDP environment. Activating an environment causes CDP to connect to the Kubernetes cluster, which provides the computing resources for the Database Catalog. In addition, activating an environment enables the Cloudera Data Warehouse (CDW) service to use the existing data lake that was set up for the environment, including all data, metadata, and security.

Before you begin

- Determine which environment that uses a particular data lake is the environment you want to activate for use with a Database Catalog and Virtual Warehouse.
- In ECS environments, the Storage Class Name is automatically obtained from Cloudera Manager.
- (Optional) Go to [Advanced Configuration Advanced Settings](#) and enable the [Use deterministic namespace names](#) option to use deterministic namespaces for Kerberos principals and keytabs. You cannot enable this option after activating an environment.
- (Optional) Go to [Advanced Configuration Advanced Settings](#) and enable the [Create databases for Virtual Warehouses](#) option if you are upgrading the CDP Private Cloud Data Services platform from an older release to the latest release, and you want to continue using external database for Hue and HMS. You cannot enable this option after activating an environment.
- (Optional) Go to [Advanced Configuration Advanced Settings](#) and turn off cluster validation by selecting the [Skip cluster validation during environment activation](#) option. By selecting this option, you can proceed with the environment activation even after seeing false positive errors in the CDW logs. Cluster validation includes port validation, and the Kerberos keytab configuration validation, and Root CA certificate validation for Impala Virtual Warehouses.



Note: A “default” environment is created by the Control Plane when you add a Private Cloud cluster. If you have more than one ECS clusters managed using different instances of Cloudera manager, but using the same Active Directory (AD) server and using the same “default” environment, then go to the [Advanced Configuration Advanced Settings](#) page and ensure that the [Use deterministic namespace names](#) option is disabled before activating the environment in CDW.

Procedure

1. Log in to Data Warehouse service as DWAdmin.
2. Click on the Environments tab.
3. Locate the environment you want to activate and click Activate.
The **Activate Environment** dialog box is displayed.

4. Enable low resource mode to deploy CDW on minimum hardware.



Note: Cloudera recommends that you use the Low Resource Mode option for proof of concept (POC) purposes only. This feature is not recommended for production deployment.

Complex queries and multiple queries on HS2 may fail due to limited memory configurations for HMS and HS2 in the low resource mode.

5. **(To use mTLS)** Browse and upload the database client certificate and database client private key files in PEM format.

The client certificate and private key files must be in PEM format.

6. Enable the Use dedicated nodes for executors option to schedule Hive and Impala executor and coordinator pods on the worker nodes tainted for CDW.

7. Select the quota-managed resource pool from the Resource Pool drop-down menu.

The Resource Pool drop-down menu is displayed only if you have enabled the quota management feature from Advanced Configurations.

8. Click Activate.

Related Information

[Advanced Configuration in CDW Private Cloud](#)

[How predefined Kerberos principals are used in CDW Private Cloud](#)

[Enabling quota management in CDW Private Cloud](#)

Creating your first Virtual Warehouse

After you activate an environment in Cloudera Data Warehouse (CDW), a default Database Catalog is automatically created. After the Database Catalog is in the running state, you can create Virtual Warehouses.

About this task

You can create Hive, Impala, or Impala Virtual Warehouses with Unified Analytics mode enabled.

Before you begin



Important: (On OpenShift environments) To activate an environment for the CDW service, someone with adequate permissions must use the Red Hat OpenShift Local Storage Operator to create a local file system on an SSD/NVMe for each OpenShift worker node and then mount it to a known location on the worker node. This creates space for local caching. The process is documented in [Activating OpenShift environments](#).

On ECS clusters, CDW automatically creates the local file system. No additional steps are needed.

Procedure

1. Log in to the data Warehouse service as DWAdmin.
2. Go to the **Virtual Warehouses** tab and click New Virtual Warehouse.
The **Create Virtual Warehouse** modal screen is displayed.
3. Specify a name for your Virtual Warehouse, select the type, specify a size and click Create Virtual Warehouse.
To create a first test Virtual Warehouse, you can proceed with the default values. For fine-tuning, sizing, configuring, creating, and upgrading Virtual Warehouses, see [Managing Virtual Warehouses](#).

Results

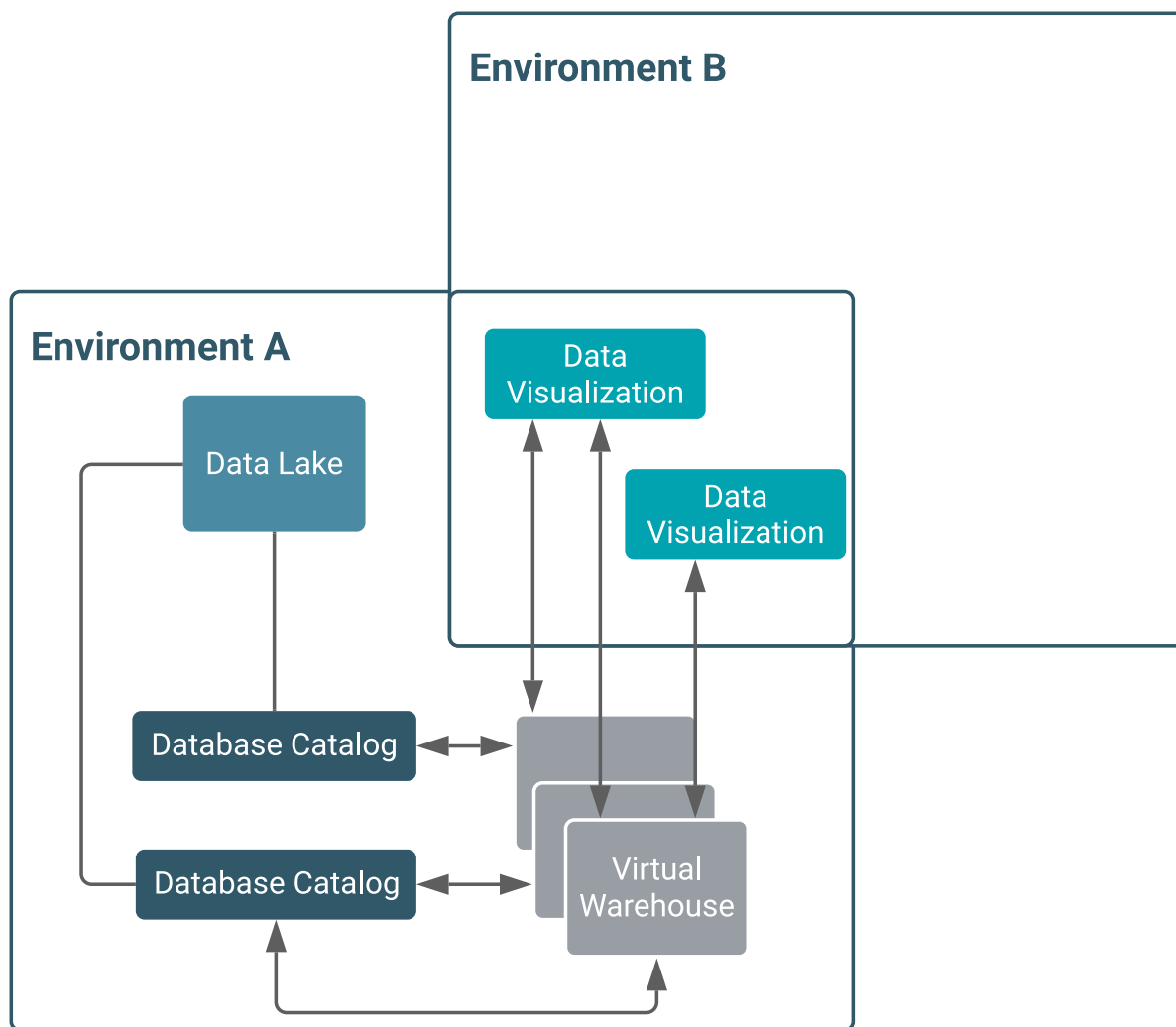
You can submit workloads and run queries using Hue. You can also use SQL clients such as beeline, impala-shell, and so on to submit workloads after you connect them to your Virtual Warehouses.

Data Visualization in Cloudera Data Warehouse

Cloudera Data Warehouse (CDW) integrates Data Visualization for building graphic representations of data, dashboards, and visual applications based on CDW data, or other data sources you connect to. You, and authorized users, can explore data across the entire CDP data lifecycle using graphics, such as pie charts and histograms. You arrange visuals on a dashboard for collaborative analysis.

You connect Data Visualization to a Virtual Warehouse as described in [Starting Data Visualization integrated in CDW](#). Similar to using a BI client, you can configure and connect to Virtual Warehouses from different clusters. You configure the connection in a familiar way, providing an IP address or host name. Data Visualization is not tied to a particular Virtual Warehouse (VW). You can access data for your visualization from multiple Data Catalogs using multiple Hive or Impala Virtual Warehouses and multiple environments.

Kubernetes Cluster



Having multiple Data Visualization instances attached to an environment, you can create dashboards for different groups. For example, Marketing and Sales can have their own private dashboards. When you delete a Virtual Warehouse, your visuals remain intact.

About setting up the Hue SQL AI Assistant

Administrators are required to set up and enable the SQL AI Assistant before Analysts can use it to generate, edit, optimize, and fix queries using natural language in Hue.

First, you must obtain clearance from your organization's infosec team to make sure it is safe to use the SQL AI Assistant because some of the table metadata and data, as mentioned in the previous section, is shared with the LLM.

Next, prepare one of the following AI platforms of your choice and then configure the SQL AI Assistant in Hue:

- Microsoft Azure OpenAI service,
- Amazon Bedrock service, or
- OpenAI platform

Preparing the Microsoft Azure OpenAI service

Microsoft Azure allows dedicated deployments of OpenAI GPT models. Using Azure's OpenAI service is much more secure than the publicly hosted OpenAI APIs because the data can be processed in your Virtual Private Cloud (VPC) network. Due to security considerations, Cloudera recommends that you use GPT models in Hue SQL AI Assistant with Azure's OpenAI service.

Procedure

1. Obtain a Microsoft Azure subscription by working with your organization's IT team.
Subscriptions vary based on your team and purpose.
2. Register to access the Azure OpenAI service.
Azure OpenAI requires registration and is currently only available to approved enterprise customers and partners. Customers who wish to use Azure OpenAI are required to submit a [registration form](#).
3. Create an Azure OpenAI resource in the Azure portal.
4. Go to Overview from the left navigation and obtain the resource URL and resource keys from the Develop tab under the resource details page.
You can use any one of the two available keys.
5. Go to Azure OpenAI Studio at <https://oai.azure.com/portal> and create your deployment under Management Deployments .
6. Select gpt-35-turbo-16k or higher.

What to do next

Enable the SQL AI Assistant in the Cloudera Data Warehouse service.

Related Information

[Configuring the SQL AI Assistant in Cloudera Data Warehouse](#)

Preparing the Amazon Bedrock Service

Amazon Bedrock is a fully managed service that makes foundation models from leading AI startups and Amazon available through an API.

Before you begin

You must have an AWS account with Bedrock access.

Procedure

1. Log in to the Amazon Bedrock service.
2. Obtain your access key and secret as follows:
 - a) Go to the IAM console: <https://console.aws.amazon.com/iam>.
 - b) Click on Users from the left menu and select the user you want to access.
 - c) Click on Security credentials.
 - d) Go to the Access keys section and note the access keys.

3. Establish Anthropic Claude access.

Claude from Anthropic is one of the best models available in Bedrock for SQL-related tasks. By default, Claude is not available on Bedrock. You need to place a special request for Claude.

After gaining access, you can try Claude in the text playground under the Amazon Bedrock service. If you are in the us-east-1 region, this must take you to <https://us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/text-playground>.

What to do next

Enable the SQL AI Assistant in the Cloudera Data Warehouse service.

Related Information

[Configuring the SQL AI Assistant in Cloudera Data Warehouse](#)

Preparing the OpenAI platform

Learn how to set up SQL AI Assistant on the OpenAI platform.

Before you begin

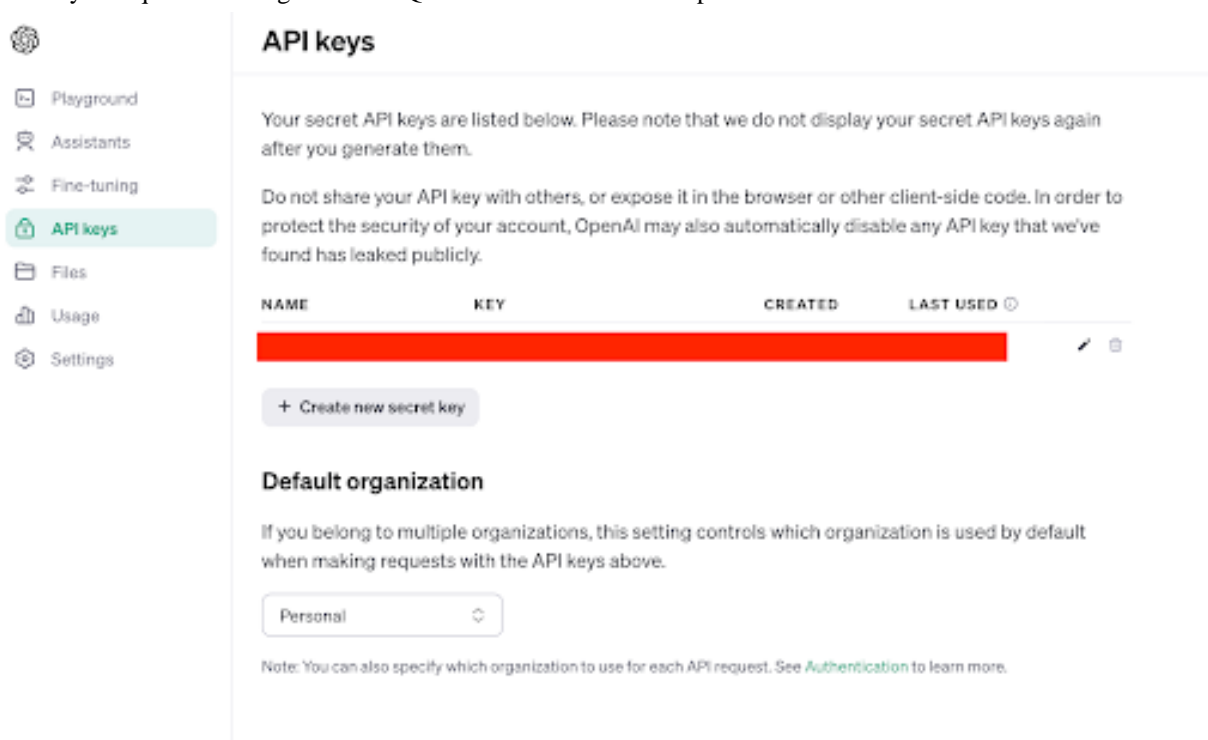
You must have created an account with the OpenAI platform.

Procedure

1. Log in to the OpenAI portal.

- Obtain the API key by navigating to the API keys menu from the left navigation pane.

The key is required to integrate Hue SQL AI Assistant with the OpenAI service.



What to do next

Enable the SQL AI Assistant in the Cloudera Data Warehouse service.

Related Information

[Configuring the SQL AI Assistant in Cloudera Data Warehouse](#)

Prerequisites for configuring Hue SQL AI Assistant

To configure the SQL AI Assistant in Hue, you must pass the token required for connecting to the LLM service. Learn about the open and secure approaches to pass the tokens, and use the one that fits your organization policy.

(Recommended) Secure approach for passing a token to configure SQL AI Assistant in Cloudera Data Warehouse

In this approach, you use Kubernetes' method of distributing secrets. You first encode the credentials and then add the encoded bit as a data item in the HUE_AI_INTERFACE_TOKEN property. The token becomes available in the Hue pod as an environment variable.

About this task



Note: Secrets are lost when you rebuild the Virtual Warehouse. You need to redo this step to continue using encoded credentials.



Note: This method is supported only on the OpenAI platform and Microsoft Azure OpenAI service. It is unsupported on Amazon Bedrock.

Procedure

1. Use a base64 encoding tool to convert your token to a base-64 representation by running the following command:

```
echo -n ' [***MY-TOKEN*** ] ' | base64
```

Replace `[***MY-TOKEN***]` with the token value you want to encode.

2. Open a terminal session and run the following command to add the encoded secret:

```
kubectl edit secret hue-secret -n [***VIRTUAL-WAREHOUSE-NAMESPACE***]
```

Replace `[***VIRTUAL-WAREHOUSE-NAMESPACE***]` with the actual Virtual Warehouse ID (same as the namespace) in which you want to add the secret.

3. Add the encoded value returned for your token in the `HUE_AI_INTERFACE_TOKEN` property as follows:

```
...
apiVersion: v1
data:
  HADOOP_CREDSTORE_PASSWORD: [***ENCODED-HADOOP-CREDSTORE-PASSWORD***]
  HUE_AI_INTERFACE_TOKEN: [***ENCODED-TOKEN-VALUE***]
kind: Secret
```

Replace `[***ENCODED-TOKEN-VALUE***]` with the actual encoded value returned for your token.

Open approach for passing a token to configure Hue SQL AI Assistant

In this approach, you specify the token value in the `hue-safety-valve` field in Cloudera Data Warehouse. The credentials are saved in a configuration file in the plain text format.



Note:

- Cloudera recommends that you use the open approach to pass tokens in test deployments, for proof of concept use cases. Use the secure approach in production deployments.
- Amazon Bedrock does not support encoded tokens. Therefore, to use the Hue SQL AI Assistant on Amazon Bedrock, you must use the open approach to configure the SQL AI Assistant on Amazon Bedrock.

Configuring the SQL AI Assistant in Cloudera Data Warehouse

Learn how to configure the SQL AI Assistant in Cloudera Data Warehouse to use it in Hue.

Procedure

1. Log in to the Data Warehouse service as DWAdmin.
2. Go to the Virtual Warehouses tab, locate the Virtual Warehouse on which you want to enable this feature, and click Edit .
3. Go to Configurations Hue , select `hue-safety-valve` from the Configuration files drop-down menu, and add the following lines depending on the approach you are using for passing the token and your platform:

For Secure token

Azure

```
[desktop]
[[ai_interface]]
  service='azure'
  model_name=' [***DEPLOYMENT-NAME*** ] '
```

```
base_url="https://[***RESOURCE***].openai.azure.com/"
```

OpenAI

```
[desktop]
[[ai_interface]]
service='openai'
```

For Open token

Azure

```
[desktop]
[[ai_interface]]
service='azure'
model_name='[***DEPLOYMENT-NAME***]'
base_url="https://[***RESOURCE***].openai.azure.com/"
token="[***RESOURCE-KEY***]"
```

AWS

```
[aws]
[[bedrock_account]]
access_key_id='[***ACCESS-KEY***]'
secret_access_key='[***SECRET-KEY***]'
region='us-east-1'
[desktop]
[[ai_interface]]
service='bedrock'
model='claude'
```

OpenAI

```
[desktop]
[[ai_interface]]
service='openai'
token='[***API-KEY***]'
```

4. Click Apply Changes.

Results

You see ✨ Assistant on the Hue SQL editor.

Related Information

[Preparing the Microsoft Azure OpenAI service](#)

[Preparing the Amazon Bedrock Service](#)

[Preparing the OpenAI platform](#)

Service and model-related configurations for setting up the Hue SQL AI Assistant

Review the list of service, model, and semantic search-related configurations used for custom configuring the AI services and models you want to use with the SQL AI Assistant and how to specify them in the Hue Advanced Configuration Snippet in Cloudera Data Warehouse (CDW) web interface.

List of service and model-related configurations

You can configure the AI services and models you want to use by going to **CONFIGURATIONS Hue hue-safety-valve** and adding the following lines:

```
[desktop]
[[ai_interface]]
  [***CONFIG-KEY1***]= ' [***VALUE***] '
  [***CONFIG-KEY2***]= ' [***VALUE***] '
[[semantic_search]]
  [***CONFIG-KEY1***]= ' [***VALUE***] '
  [***CONFIG-KEY2***]= ' [***VALUE***] '
```

Specify the service and model-related configurations under the `[[ai_interface]]` section as listed in the following table:

| AI interface config key | Description |
|-----------------------------|--|
| service | API service to be used for AI tasks. AI is disabled when a service is not configured. For example, <code>ai_assistant</code> . |
| trusted_service | Indicates whether the LLM is trusted or not. Turn on to disable the warning. The default value is <code>False</code> . |
| model | The AI model you want to use for AI tasks. For example, <code>gptand llama</code> . |
| model_name | The fully qualified name of the model to be used. For example, <code>gpt-3.5-turbo-16k</code> . |
| base_url | Service API base URL. |
| add_table_data | When enabled, sample rows from the table are added to the prompt. The default value is <code>True</code> . |
| table_data_cache_size | Size of the LRU cache used for storing table sample data. |
| auto_fetch_table_meta_limit | Number of tables to load from a database, initially. |
| token | Service API secret token. |
| token_script | Provides a secure way to get the service API secret token. |

List of semantic search-related configurations

Specify the semantic search-related configurations used for RAG under the `[[semantic_search]]` section, as listed in the following table:

| Semantic search config key | Description |
|----------------------------|---|
| relevancy | The technology you want to use for semantic search. Acceptable values are <code>vector_search</code> or <code>keyword_search</code> . |
| embedding_model | The model you want to use for data-embedding. This must be compatible with SentenceTransformer. |
| cache_size | Size of the LRU cache used for storing embedding. |