Cloudera Data Warehouse on premises 1.5.5

Planning and setting up Cloudera Data Warehouse on premises

Date published: 2020-08-17
Date modified: 2025-11-08



https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2026. 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 Cloudera Data Warehouse.....5

Requirements.....5
Low resource requirements.....5
Standard resource requirements.....7
Security requirements for Cloudera Data Warehouse on premises.....8
Port requirements for AD.....8
Database requirements.....9
User roles and other prerequisites.....10
Pod placement policy and rack awareness in Cloudera Data Warehouse on premises.....11
Configuring cluster issuer for Certificate Manager.....11

Activate OpenShift environments.....12

Activate ECS environments.....13

Create first Virtual Warehouse.....15

Set up Data Viz.....15
About the Cloudera Data Explorer (Hue) SQL AI Assistant.....16
About setting up the Cloudera Data Explorer (Hue) SQL AI Assistant.....18
Prerequisites for configuring Cloudera Data Explorer (Hue) SQL AI Assistant.....18
(Recommended) Secure approach for passing a token.....18
Open approach for passing a token.....19
Configure SQL AI Assistant using Cloudera AI Workbench.....20
Configure SQL AI Assistant using the Cloudera AI Inference service.....20
Configure SQL AI Assistant using the Microsoft Azure OpenAI service.....21
Configure SQL AI Assistant using the Amazon Bedrock Service.....21
Configure SQL AI Assistant using the OpenAI platform.....22
Complete list of model-related configurations for setting up the Cloudera Data Explorer (Hue) SQL AI Assistant.....23
Start SQL AI Assistant.....24
Generate SQL from NQL.....25
Edit query in natural language.....25
Explain query in natural language.....26
Optimize SQL query.....27
Fixing a query in Cloudera Data Explorer (Hue).....28
Generate comment for a SQL query.....29
Multi database support for SQL query.....30
Fact support for SQL query.....31
Cloudera Data Explorer (Hue) SQL AI API overview (Preview).....32
SQL AI REST API Documentation.....33
Enabling SQL AI Assistant API via UI.....35

Cloudera Data Explorer (Hue) SQL AI Assistant FAQ..... 35

About deploying the shared Cloudera Data Explorer (Hue) service..... 37

Access control for the shared Cloudera Data Explorer (Hue) service..... 37
Creating a shared Cloudera Data Explorer (Hue) instance..... 38
Rebuilding a shared Cloudera Data Explorer (Hue) service..... 39
Upgrading a shared Cloudera Data Explorer (Hue) instance..... 39
FAQ for shared Cloudera Data Explorer (Hue) service..... 39

Planning and setting up Cloudera Data Warehouse on premises

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

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

Requirements for deploying Cloudera Data Warehouse on premises

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

Low resource mode requirements

Review the memory, storage, and hardware requirements for getting started with the Cloudera Data Warehouse 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 Cloudera Data Warehouse 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 Cloudera Data Warehouse 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.

- Cloudera Manager must be installed and running.
- Cloudera Data Services on premises must be installed and running. See Installing on OpenShift and Installing on ECS for more details.
- An environment must have been registered with Cloudera Management Console on the on premises. See Cloudera on premises Environments for more details.
- In addition to the general requirements, Cloudera Data Warehouse 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 Cloudera Base on premisesnode

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
Data Explorer (backend)	1	8 GB	--	1
Data Explorer (frontend)	0.5	8 GB	--	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
Data Explorer (backend)	1	8 GB	--	1
Data Explorer (frontend)	0.5	8GB	--	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 service in standard resource mode on Red Hat OpenShift and Cloudera Embedded Container Service.

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

- Cloudera Manager must be installed and running.
- Cloudera Data Services on premises must be installed and running. See Installing on OpenShift and Installing on ECS for more details.
- An environment must have been registered with Cloudera Management Console on the on premises. See Cloudera on premises Environments for more details.
- In addition to the general requirements, Cloudera Data Warehouse 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 Cloudera Embedded Container Service 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 Cloudera Embedded Container Service.

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 Cloudera Base on premises	10 GB/s guaranteed bandwidth to every Cloudera Base on premises 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 on premises

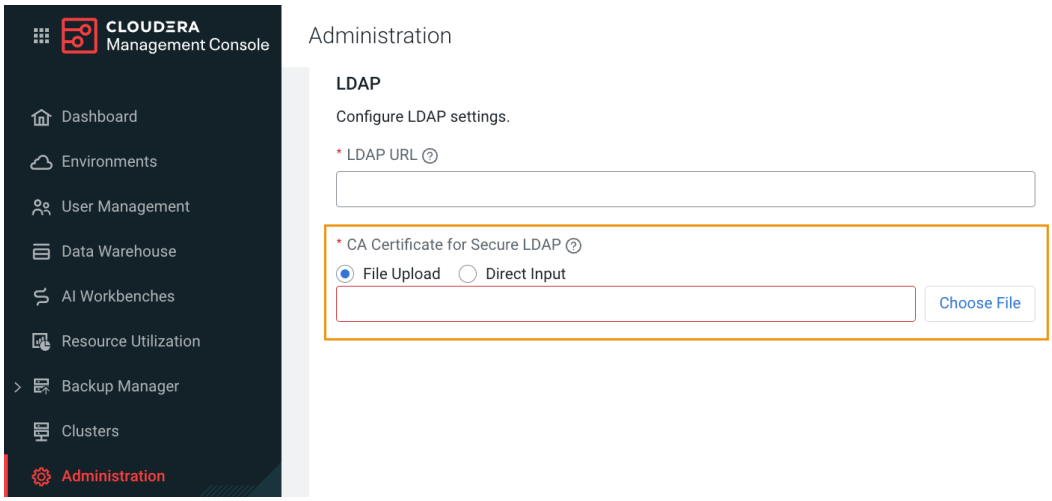
This topic describes security requirements needed to install and run Cloudera Data Warehouse on premises service on Red Hat OpenShift and Embedded Container Service (ECS) clusters.

Required OpenShift/ECS cluster permissions

The Cloudera Data Warehouse service requires the "cluster-admin" role on the OpenShift and Cloudera Embedded Container Service 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.

Cloudera on premises LDAP certificate requirement

A certificate authority (CA) certificate for secure LDAP must be uploaded to the Administration page of Cloudera Management Console to run Cloudera Data Warehouse on premises service:



Port requirements for AD in Cloudera Data Warehouse on premisesd

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

In Cloudera Data Warehouse, 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 Cloudera 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
```

Cloudera Data Warehouse performs port validation when you activate an environment in Cloudera Data Warehouse. 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 Cloudera 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 on premises

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 on premises.

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

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 Cloudera Data Warehouse 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 Cloudera 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 Cloudera 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 Cloudera 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***>
```

Cloudera resource roles and other prerequisites

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

Unsupported compression

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

Cloudera resource roles

Required role: PowerUser

The following Cloudera resource roles are associated with the Cloudera Data Warehouse service. A Cloudera 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 Cloudera 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 Cloudera Data Warehouse clusters (Virtual Warehouses) that are associated with specific environments.

Requirements for Data Explorer

Data Explorer in Cloudera Data Warehouse requires WebHDFS to be enabled on the Cloudera Base on premises 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 Cloudera Data Services on premises 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 Cloudera Data Services on premises

Pod placement policy and rack awareness in Cloudera Data Warehouse on premises

In Cloudera Data Warehouse, Kubernetes node affinity rules are based on the rack topology defined in Cloudera Manager. If sufficient resources are available, Cloudera Data Warehouse 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 Cloudera Data Warehouse 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

Configuring cluster issuer for Certificate Manager

A third-party Certificate Manager is installed by default as part of the Cloudera Embedded Container Service installation. Learn how you can configure cluster issuers with the appropriate annotations to enable the use of certificate manager in Cloudera Data Warehouse.



Important:

- Third-party certificate manager is available only for new installations performed on Cloudera Embedded Container Service.
- Cloudera currently supports only Venafi Trust Protection Platform (TPP) as the certificate issuer.

It is recommended that you configure the certification manager before creating any Database Catalogs or Virtual Warehouses in Cloudera Data Warehouse. For installing a cluster issuer, see Setting up Certification Manager using Venafi TPP. To validate if there is a valid cluster issuer, see the following rules:

- The cluster issuer must have the following annotation: issuer.cdp.cloudera.com/type=longlived
- The cluster issuer must have the label set as follows: issuer.cdp.cloudera.com/project=[***CDP_NAMESPACE***]

Alternatively, you can configure the certificate duration by setting the issuer.cdp.cloudera.com/duration annotation in the cluster issuer. The data service applies the specified duration for all certificate requests. For example, to configure a certificate duration of 6 months, set issuer.cdp.cloudera.com/duration=4380h.

By default, certificate manager requests certificates with a 90-day expiration and automatically renews them when they are 2/3 of the way through their validity period. This means certificates are renewed after 60 days. If the certificate expiration is modified to 1 year, the certificate manager will renew the certificate after 8 months.

Note:

- It is recommended to create a unique and valid cluster issuer following the preceding rules.
- If the cluster issuer is not in a ready state, the Kubernetes cluster-level certificate is used, which may result in certificate errors in the browser.
- If the certification manager related settings are modified in any way, you must rebuild the Database Catalog, Virtual Warehouses, Data Explorer, and the Cloudera Data Visualization instance. A Refresh operation does not recreate or request the certificates.
- When the Certification Manager is enabled, the trust store file provided by the Cloudera Data Warehouse becomes invalid for connecting to Beeline.

  You can fetch a trust store file from the Cloudera Data Warehouse UI to configure TLS and connect to Beeline. For instructions on downloading the trust store file, see Downloading root certificates from Cloudera Data Warehouse web UI.

  However, when the certification manager is enabled, Cloudera Data Warehouse continues to provide the same trust store file, even though the Hive cluster uses a different certificate. The issue arises because Cloudera Data Warehouse cannot provide a trust store compatible with the configured certificate issuer.

  To resolve this issue, use your own certificate authority and set up a trust store on your system. You can do this by fetching the public key of the root CA (in .pem or .crt format), and then creating a new trust store using the following command:

```
keytool -importcert -trustcacerts -file rootca.pem -alias rootca -ke
ystore truststore.jks -storepass changeit
```

Activating OpenShift environments

This topic describes how to activate an environment to use for Cloudera Data Warehouse on premises 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 Cloudera environment. Activating an environment causes Cloudera 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 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 Cloudera Data Warehouse 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 Cloudera Data Services on premises platform from an older release

to the latest release, and you want to continue using external database for Data Explorer 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 Cloudera Data Warehouse 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 Cloudera Data Warehouse.

Procedure

1. Log in to Cloudera 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 Cloudera Data Warehouse 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. This mode is deprecated and it will be removed in the future releases.
7. Enable the Use dedicated nodes for executors option to schedule Hive and Impala executor and coordinator pods on the worker nodes tainted for Cloudera Data Warehouse.
8. Click Activate.

Related Information

Advanced Configuration in Cloudera Data Warehouse on premises

How predefined Kerberos principals are used in Cloudera Data Warehouse on premises

Activating Embedded Container Service environments

This topic describes how to activate an environment to use for Cloudera Data Warehouse on premises 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 Cloudera environment. Activating an environment causes Cloudera 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 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 Cloudera Data Services on premises platform from an older release to the latest release, and you want to continue using external database for Data Explorer 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 Cloudera Data Warehouse 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 Cloudera Control Plane when you add a on premises 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 Cloudera Data Warehouse.

Procedure

1. Log in to Cloudera 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 Cloudera Data Warehouse 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 Cloudera Data Warehouse.
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 Cloudera Data Warehouse on premises
How predefined Kerberos principals are used in Cloudera Data Warehouse on premises

Creating your first Virtual Warehouse

After you activate an environment in Cloudera Data Warehouse, 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 Trino Virtual Warehouses.

Before you begin



Important: (On OpenShift environments) To activate an environment for the Cloudera Data Warehouse 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, Cloudera Data Warehouse automatically creates the local file system. No additional steps are needed.

Procedure

1. Log in to the Cloudera 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.

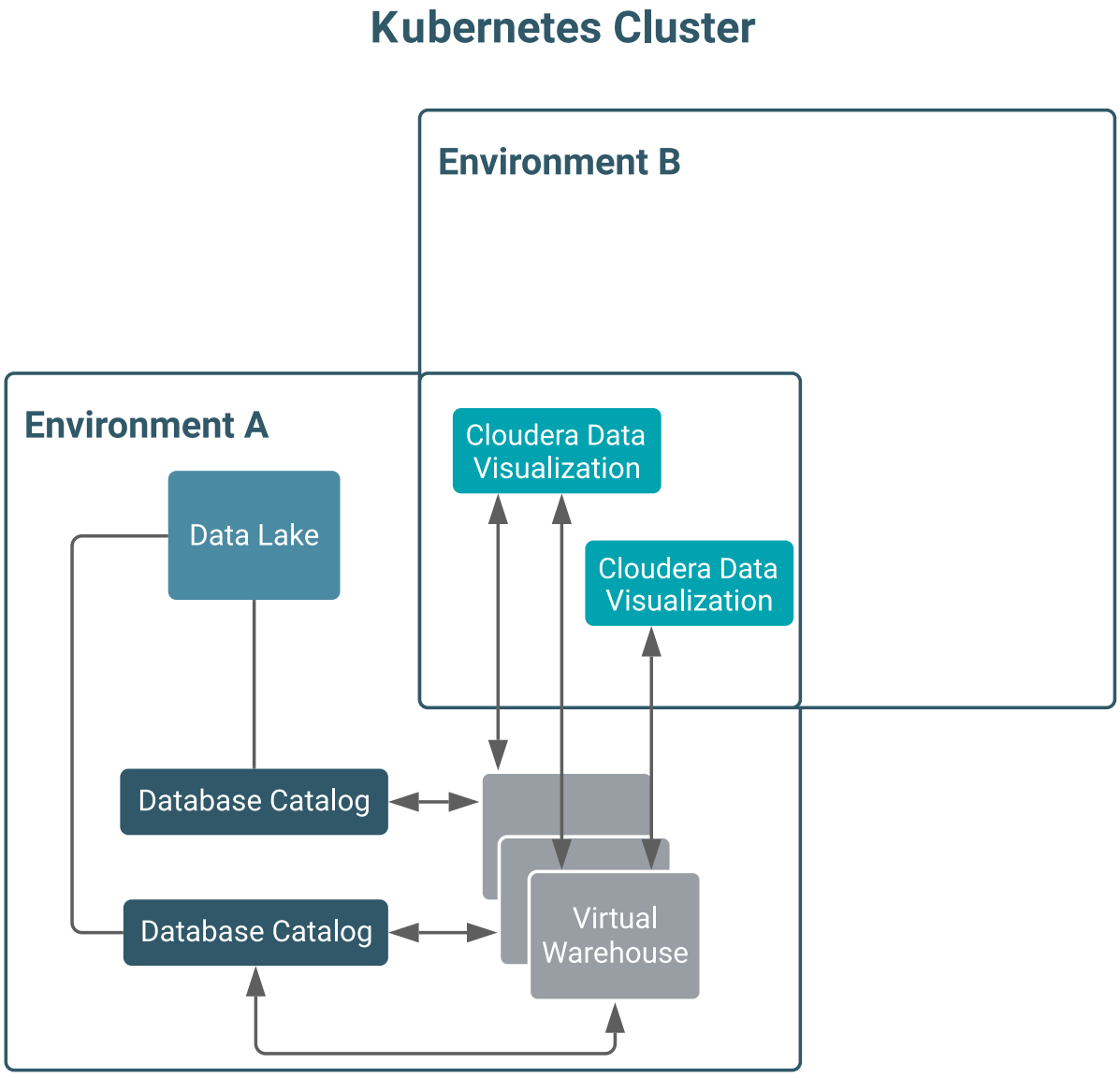
Results

You can submit workloads and run queries using Data Explorer. 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 integrates Data Visualization for building graphic representations of data, dashboards, and visual applications based on Cloudera Data Warehouse data, or other data sources you connect to. You, and authorized users, can explore data across the entire Cloudera 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 Cloudera Data Warehouse. 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.



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 the Cloudera Data Explorer (Hue) SQL AI Assistant

Learn about the AI models and services that Data Explorer uses to run the SQL AI Assistant and its limitations. Review what data is shared with the LLM models before you start using the SQL AI Assistant with Data Explorer.

A SQL AI Assistant has been integrated into Data Explorer with the capability to leverage the power of Large Language Models (LLMs) for various SQL tasks. It helps you to create, edit, optimize, fix, and succinctly summarize queries using natural language and makes SQL development faster, easier, and less error-prone. Multiple SQL dialects are supported, including Hive, Impala, and Trino.

Next, select and prepare one of the following AI services of your choice for hosting an LLM, and then configure the SQL AI Assistant in Data Explorer:

- Cloudera AI Workbench
- Cloudera AI Inference service
- Microsoft Azure OpenAI service
- Amazon Bedrock service
- OpenAI platform

Note:

To use cloud-hosted models, you must provide the requisite network connections to ensure that Cloudera Data Services on premises nodes can communicate with the AWS, Azure, and OpenAI services that are hosted on the cloud. Cloudera recommends that you configure host-level internet proxy on all Cloudera Data Services on premises nodes.

You must have access to the Hugging Face to download the required sentence transformer model, and ensure your system can connect to the internet, specifically, huggingface.co. Since these models aren't pre-bundled, they must be downloaded during setup. For more information, see Hugging Face.

The SQL AI Assistant uses a Retrieval Augmented Generation (RAG)-based architecture for augmenting results. It uses the sentence-transformer library for semantic search, and Data Explorer can be configured with any of the pre-trained models for better multi-lingual support. By default, "all-MiniLM-L6-v2" models are used.

Embedding Model	Language Support
all-MiniLM-L6-v2	English
distiluse-base-multilingual-cased-v1	Arabic, Chinese, Dutch, English, French, German, Italian, Korean, Polish, Portuguese, Russian, Spanish, and Turkish.

What data is shared with the LLM models

The following details are shared with the LLMs:

- Everything that a user inputs
- Dialect in use
- Table details such as table name, column names, column data types and related keys, partitions, and constraints that the logged-in user has access to.
- Three sample rows from the tables (as per the best practices specified in *Evaluating the Text-to-SQL Capabilities of Large Language Models*)

Limitations

Non-deterministic nature

LLMs are non-deterministic, which means you cannot guarantee the same output for the same input every time, and it can lead to different responses to similar queries.

Ambiguity

LLMs may struggle to handle ambiguous queries or contexts. SQL queries often rely on specific and unambiguous language, but LLMs can misinterpret or generate ambiguous SQL queries, leading to incorrect results.

Hallucinations

In the context of LLMs, hallucination refers to a phenomenon where these models generate text or responses that are incorrect, nonsensical, or fabricated. Occasionally you might see incorrect identifiers or literals in the response.

About setting up the Cloudera Data Explorer (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, explain, and fix queries using natural language in Data Explorer.

First, you must obtain clearance from your organization's infosec team to ensure 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, select and prepare one of the following AI services of your choice for hosting an LLM, and then configure the SQL AI Assistant in Data Explorer:

- Cloudera AI Workbench
- Cloudera AI Inference service
- Microsoft Azure OpenAI service
- Amazon Bedrock service
- OpenAI platform

Prerequisites for configuring Cloudera Data Explorer (Hue) SQL AI Assistant

To configure the SQL AI Assistant in Data Explorer, 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

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 Data Explorer 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.

Before you begin

To run the `kubectl` commands listed below, you must:

1. Have `kubectl` installed on your local machine.
2. Download and configure the kubeconfig file for the specific Cloudera Data Warehouse environment. You can obtain this from the Cloudera Management Console under the Container Service environment settings.
3. Ensure your user account or service principal has the necessary Role-Based Access Control (RBAC) permissions to edit secrets within the Virtual Warehouse namespace.

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

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 for passing a token in production deployments.

Here's a list of the open token values in the hue-safety-valve field to configure the SQL AI Assistant:

For Open token

Microsoft Azure OpenAI

```
[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***]'
```

Configure SQL AI Assistant using Cloudera AI Workbench

This topic describes how to deploy and configure the SQL AI Assistant using the Cloudera AI Workbench. With the added support for Cloudera AI Workbench, you can securely deploy and run your own models within a virtual private cloud. This self-contained integration offers enhanced control and privacy within your environment.

Before you begin

To know more about creating and deploying models using Cloudera AI Workbench, see Create and deploy the model.

Procedure

1. Upon successful completion of model deployment, log in to the Cloudera Data Warehouse service.
2. Go to the Virtual Warehouses tab, locate the Virtual Warehouse on which you want to enable this feature, and click ⋮ Details .
3. Go to Configurations Data Explorer (Hue) , select hue-safety-valve from the Configuration files drop-down menu, and add the following lines :

```
[desktop]
[[ai_interface]]
service='cml'
model='llama'
model_ref='[***Place model access key here***]'
base_url='https://[***RESOURCE***].cloudera.site/model'
```

4. Click Save Changes.

Results

You see ✨ Assistant on the Data Explorer SQL editor, where the SQL AI Assistant utilizes the model hosted in the Cloudera AI Workbench.

Configure SQL AI Assistant using the Cloudera AI Inference service

This topic describes configuring the SQL AI Assistant using the Cloudera AI Inference service.

Before you begin

To know more about installing and setting up the Cloudera AI Inference service, see Prerequisites for setting up the Cloudera AI Inference service.

Procedure

1. Upon successful completion of model deployment, log in to the Cloudera Data Warehouse service.
2. Go to the Virtual Warehouses tab, locate the Virtual Warehouse on which you want to enable this feature, and click ⋮ Details .
3. Go to Configurations Data Explorer (Hue) , select hue-safety-valve from the Configuration files drop-down menu, and add the following lines :

```
[desktop]
[[ai_interface]]
service='caii'
model_name='[***Place MODEL name here***]'
base_url="https://[***RESOURCE***]/v1"
```

4. Click Save Changes.

Results

You see ✨ Assistant on the Data Explorer SQL editor, where the SQL AI Assistant utlizes the model hosted in Cloudera AI Inference service.

Configure SQL AI Assistant using the Microsoft Azure OpenAI service

Microsoft Azure allows for dedicated deployments of OpenAI GPT models. You can use Azure's OpenAI service instead of the publicly hosted OpenAI APIs, as it enables data processing within your Azure Virtual Network (VNet) network. GPT models can also be integrated with the Data Explorer SQL AI Assistant using Azure's OpenAI service.

Before you begin

Obtain a Microsoft Azure subscription by working with your organization's IT team and registering for access to the Azure OpenAI service. For more information, see Create and deploy an Azure OpenAI Service resource.

Procedure

1. Upon successful completion of model deployment, log in to the Cloudera Data Warehouse service.
2. Go to the Virtual Warehouses tab, locate the Virtual Warehouse on which you want to enable this feature, and click ⋮ Details .
3. Go to Configurations Data Explorer (Hue) , select hue-safety-valve from the Configuration files drop-down menu, and add the following lines :

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

4. Click Save Changes.

Results

You see ✨ Assistant on the Data Explorer SQL editor, and the SQL AI Assistant will connect to the specified model on the Microsoft Azure OpenAI service.

Configure SQL AI Assistant using the Amazon Bedrock Service

This topic describes how to configure the SQL AI Assistant using the Amazon Bedrock Service.

Before you begin

You must have an AWS account with Bedrock access.

Note: Cloudera recommends that you review the official AWS documentation for Generate an Amazon Bedrock API key to ensure you follow the most recent instructions, as Cloudera prerequisites might not reflect the most recent updates from AWS.

1. Log in to the Amazon Bedrock service.
2. Obtain your access key and secret as follows:
   a. Go to the IAM console.
   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.

Procedure

1. Upon successful completion of model deployment, log in to the Cloudera Data Warehouse service.
2. Go to the Virtual Warehouses tab, locate the Virtual Warehouse on which you want to enable this feature, and click ⋮ Details .
3. Go to Configurations Data Explorer (Hue) , select hue-safety-valve from the Configuration files drop-down menu, and add the following lines :

```
[aws]
[[bedrock_account]]
access_key_id_script='echo $AWS_BEDROCK_ACCESS_KEY_ID'
secret_access_key_script='echo $AWS_BEDROCK_SECRET_ACCESS_KEY'
region='us-east-1'
[desktop]
[[ai_interface]]
service='bedrock'
model='claude'
```

AWS_BEDROCK_ACCESS_KEY_ID and AWS_BEDROCK_SECRET_ACCESS_KEY must be added as encoded values under hue-secret. For more information, see Secure approach for passing a token.

4. Click Save Changes.

Results

You see ✨ Assistant on the Data Explorer SQL editor, and the SQL AI Assistant will connect to the specified model in the Amazon Bedrock service.

Configure SQL AI Assistant using the OpenAI platform

This topic describes how to set up SQL AI Assistant and connect to a model on the OpenAI platform.

Before you begin

You must have created an account with the OpenAI platform.

Procedure

1. Upon successful completion of model deployment, log in to the Cloudera Data Warehouse service.
2. Go to the Virtual Warehouses tab, locate the Virtual Warehouse on which you want to enable this feature, and click ⋮ Details .
3. Go to Configurations Data Explorer (Hue) , select hue-safety-valve from the Configuration files drop-down menu, and add the following lines :

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

You can specify the model_name (optional) and define the model. If no model is defined, the default model (gpt-3.5-turbo-16k) will be used.

4. Click Save Changes.

Results

You see ✨ Assistant on the Data Explorer SQL editor, and the SQL AI Assistant will connect to the specified model on the OpenAI platform.

Complete list of model-related configurations for setting up the Cloudera Data Explorer (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 the Cloudera Data Warehouse web interface.

List of service and model-related configurations

In Cloudera Data Warehouse, you can configure by going to Virtual Warehouse CONFIGURATIONS Data Explorer (Hue) hue-safety-valve . The following template shows the base structure of the configurations and adding the following lines:

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

AI interface-related configurations

Here is the complete list of configurations under [[ai_interface]], which allows you to specify the service and model to be used:

AI interface config key	Description
service	API service to be used for AI tasks. AI is disabled when a service is not configured. For exam Workbench and Cloudera AI Inference service are API services.
service_version	API service version to be used for AI tasks.
trusted_service	Indicates whether the LLM is trusted or not. Turn on to disable the warning. The default value
model	The AI model you want to use for AI tasks. For example, gpt and llama.
model_name	The fully qualified name of the model to be used. For example, gpt-3.5-turbo-16k.
model_ref	The `model_ref` is a placeholder for adding the access key of the specific model you want to u
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 True.
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.
enabled_sql_tasks	A comma-separated list of SQL-related AI tasks available in the Editor.

User Input Validation for Data Explorer SQL AI

Following is the complete list of configurations under [[ai_interface]]. It helps to specify the input validation to enhance security and optimize performance.

AI interface config key	Description
user_input_max_length	Ensure the configured user input length is not exceeded. The default limit is 1000, but you can configure it to a higher value if needed.
user_input_remove_characters	Remove specific characters from user input, such as newlines (\n), tab spaces (\t), and others, to ensure clean and consistent formatting.
user_input_banned_keyphrases	Block user input if certain configured keyphrases are found.
user_input_banned_regex	Block user input if a configured regex pattern match is found.
user_input_block_html	Escape HTML tags to prevent malicious activities and ensure secure input handling. This config accepts a boolean value: True to escape HTML tags or False to allow raw HTML. The default value is set to False.

The following sample configuration sets the validations for user input:

```
[[ai_interface]]
service='azure'
model_name='[***DEPLOYMENT-NAME***]'
base_url='https://[***RESOURCE***].cloudera.site/model'
token='[***RESOURCE-KEY***]'
user_input_max_length=1000
user_input_remove_characters="&\n\r\t"
user_input_banned_keyphrases=""
user_input_banned_regex=""
user_input_block_html="False"
```

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 vector_search or v
embedding_model	The model you want to use for data-embedding. This must be compatible with SentenceTransf
top_k	Number of top-ranking items returned by semantic search.
cache_size	Size of the LRU cache used for storing embedding.

Starting SQL AI Assistant in Cloudera Data Explorer (Hue)

A SQL AI Assistant has been integrated into Data Explorer with the capability to leverage the power of Large Language Models (LLMs) for various SQL tasks. It helps you to create, edit, optimize, fix, and succinctly summarize

queries using natural language and makes SQL development faster, easier, and less error-prone. You can also generate comments and insert them into your queries to improve readability.

About this task

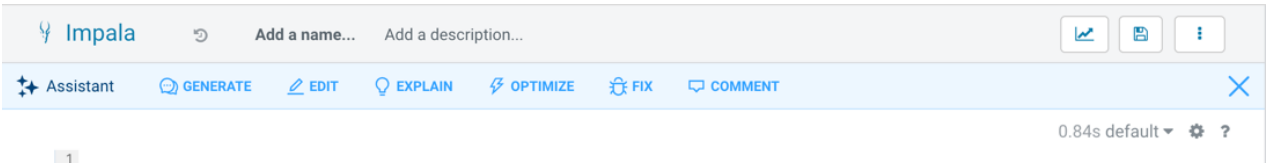
Attention: The SQL AI Assistant operates only on the database that you have selected in the Data Explorer editor, and not necessarily on the one that is displayed on the left-assist bar.

Procedure

1. Log in to the service as DWUser.
2. Open Data Explorer corresponding to your Virtual Warehouse.
3. Click Assistant on the Data Explorer SQL editor:

Results

The following options are displayed:

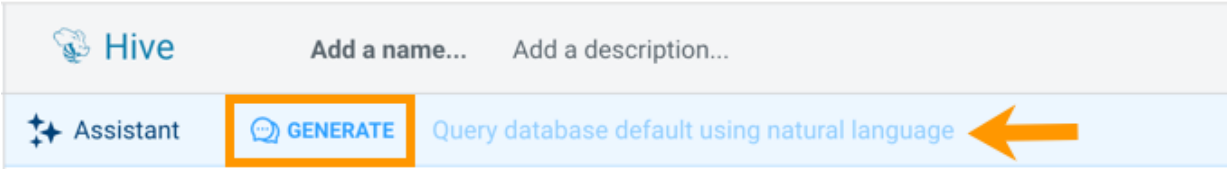


Generating SQL from natural language in Cloudera Data Explorer (Hue)

The SQL AI Assistant in Cloudera Data Warehouse helps you to generate SQL queries by entering a prompt in natural language. You can then insert the generated SQL in the Data Explorer SQL editor and run it as usual.

Procedure

1. Log in to the service as DWUser.
2. Open Data Explorer corresponding to your Virtual Warehouse.
3. Click Assistant on the Data Explorer SQL editor:
4. Click GENERATE.



A SQL query is generated based on your input prompt. Click Insert to insert the query into the editor and run it.

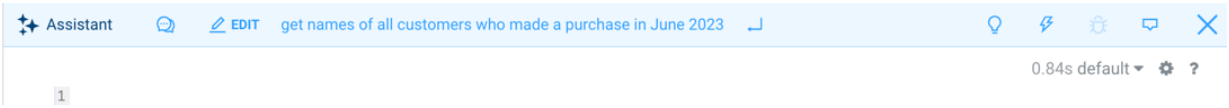
Editing the query in natural language in Cloudera Data Explorer (Hue)

You can edit a query in natural language and generate it again to fine-tune your results using the Data Explorer SQL AI Assistant. You can then insert the SQL query into the editor and run it as usual.

Procedure

1. Log in to the service as DWUser.
2. Open Data Explorer corresponding to your Virtual Warehouse.

3. Click Assistant on the Data Explorer SQL editor:
4. Click EDIT.



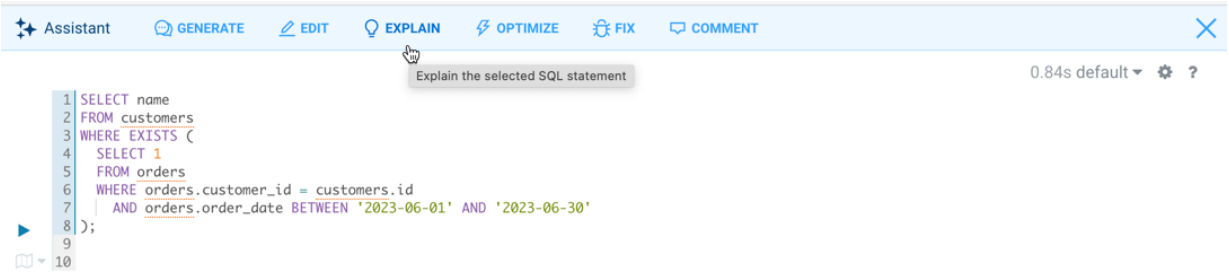
Select the query from the list and edit it as needed. Then press enter to regenerate the SQL.

Getting an explanation of a SQL query in natural language in Cloudera Data Explorer (Hue)

You can use the SQL AI Assistant to breakdown and understand a complex SQL query in natural language.

Procedure

1. Log in to the service as DWUser.
2. Open Data Explorer corresponding to your Virtual Warehouse.
3. Insert a SQL query in the Data Explorer editor that you would like to understand in natural language.
4. Click Assistant on the Data Explorer SQL editor:
5. Click EXPLAIN.



The LLM generates the explanation of the SQL query.



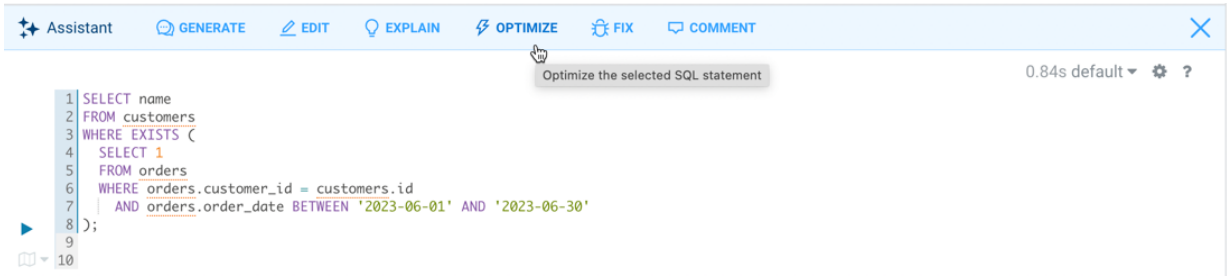
Optimizing a query in Cloudera Data Explorer (Hue)

You can use the SQL AI Assistant to optimize a SQL query. Data Explorer identifies the issues in the source query, optimizes it, and provides the optimized version of the SQL query. Data Explorer also summarizes the issues and how it optimized the query in natural language.

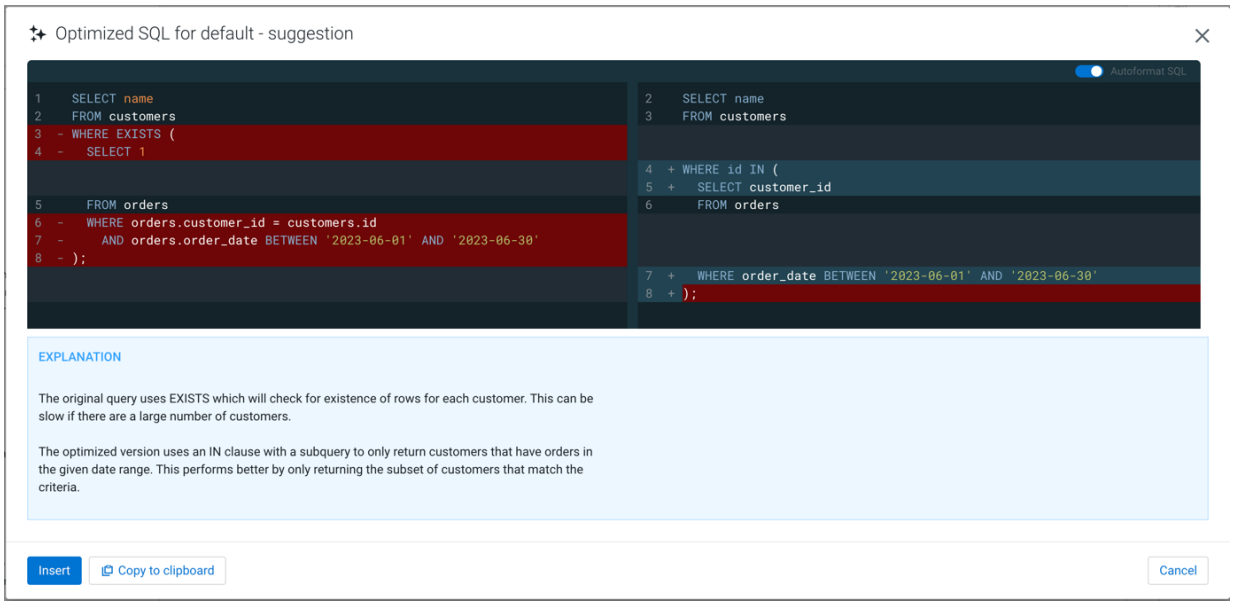
Procedure

1. Log in to the service as DWUser.
2. Open Data Explorer corresponding to your Virtual Warehouse.
3. Insert a SQL query in the Data Explorer editor that you would like to optimize.

4. Click Assistant on the Data Explorer SQL editor:
5. Click OPTIMIZE.



Data Explorer displays the original and the optimized SQL query side-by-side. It also provides an explanation of the issues in the original query and how it was optimized.

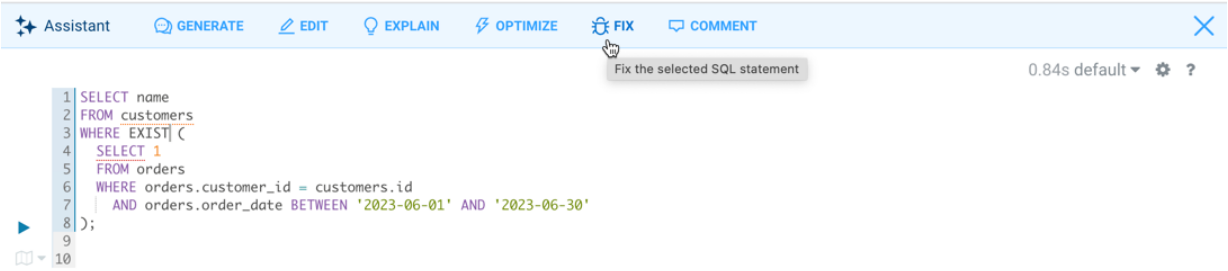


Fixing a query in Cloudera Data Explorer (Hue)

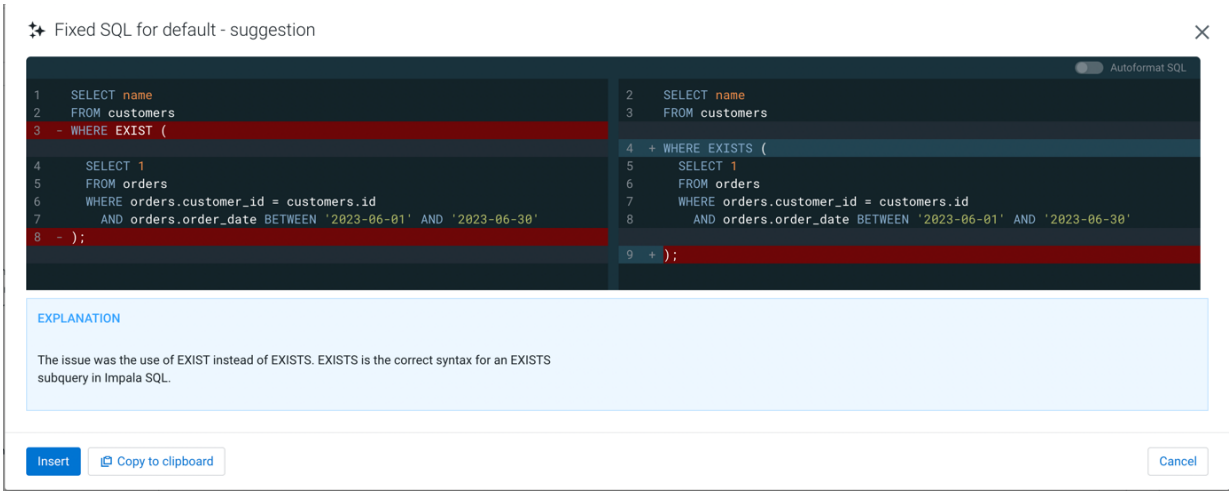
You can use the SQL AI Assistant to fix a broken SQL query. Data Explorer identifies the issues in SQL syntax and provides the corrected version.

Procedure

1. Log in to the service as DWUser.
2. Open Data Explorer corresponding to your Virtual Warehouse.
3. Insert a SQL query in the Data Explorer editor that you would like to fix.
4. Click Assistant on the Data Explorer SQL editor:
5. Click FIX.



Data Explorer displays the original and the fixed SQL query in a side-by-side comparison.



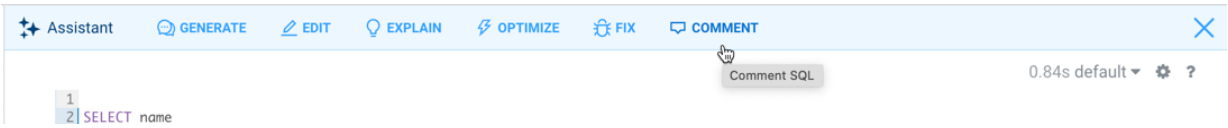
Click Insert to insert the fixed query in the Data Explorer editor and run it.

Generating a comment for a query in Cloudera Data Explorer (Hue)

The Data Explorer SQL AI Assistant can generate a comment explaining what SQL query does. You can insert it into the query to improve readability.

Procedure

1. Log in to the service as DWUser.
2. Open Data Explorer corresponding to your Virtual Warehouse.
3. Insert a SQL query in the Data Explorer editor for which you want to generate a comment.
4. Click ✨ Assistant on the Data Explorer SQL editor:
5. Click COMMENT.



The SQL AI Assistant generates a detailed comment for the input SQL query.



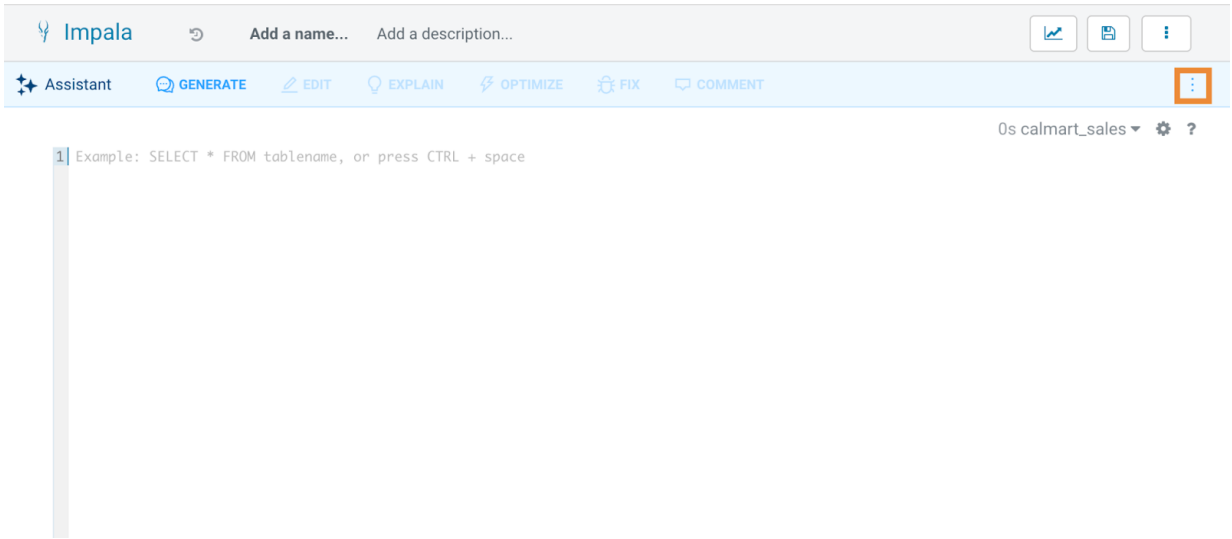
Click Insert to insert the comment into the query.

Multi database support for SQL query

The Data Explorer SQL AI Assistant now supports multi-database querying, allowing you to retrieve data from multiple databases simultaneously. This enhancement simplifies managing large datasets across different systems and enables seamless cross-database queries.

Procedure

1. Click ✨ Assistant on the Data Explorer SQL editor, then open the AI Assistant Settings.



2. Select the databases you want to include in your queries. You can choose multiple databases from the list provided and click OK.
3. Enter your prompt in natural language to generate SQL queries.

Querying across multiple databases

Database Name	Tables
CalMart_Sales_DB	Customers, Sales, Sales_Items
CalMart_Products_DB	Products, Suppliers, Product_Supplier

Use Case: Identifying Top-Selling Products and Their Suppliers.

Objective: Retrieve top-selling products along with their suppliers by combining data from two databases.

```
SELECT
si.product_id,
p.product_name,
SUM(si.quantity) AS total_quantity_sold,
s.supplier_name
FROM
CalMart_Sales_DB.Sales_Items si
JOIN
CalMart_Products_DB.Products p
ON
si.product_id = p.product_id
JOIN
CalMart_Products_DB.Product_Supplier ps
ON
p.product_id = ps.product_id
JOIN
CalMart_Products_DB.Suppliers s
ON
ps.supplier_id = s.supplier_id
GROUP BY
si.product_id, p.product_name, s.supplier_name
ORDER BY
total_quantity_sold DESC;
```

This query retrieves a comprehensive list of top-selling products along with their suppliers by combining data from multiple databases.

Fact support for SQL query

The Data Explorer SQL AI Assistant now allows you to use Facts to guide the generation of precise SQL queries. It enables the AI Assistant to process the defined custom facts in the AI Assistant settings to enrich prompt construction and improve the accuracy of complex, cross-database SQL workflows.

Key advantages

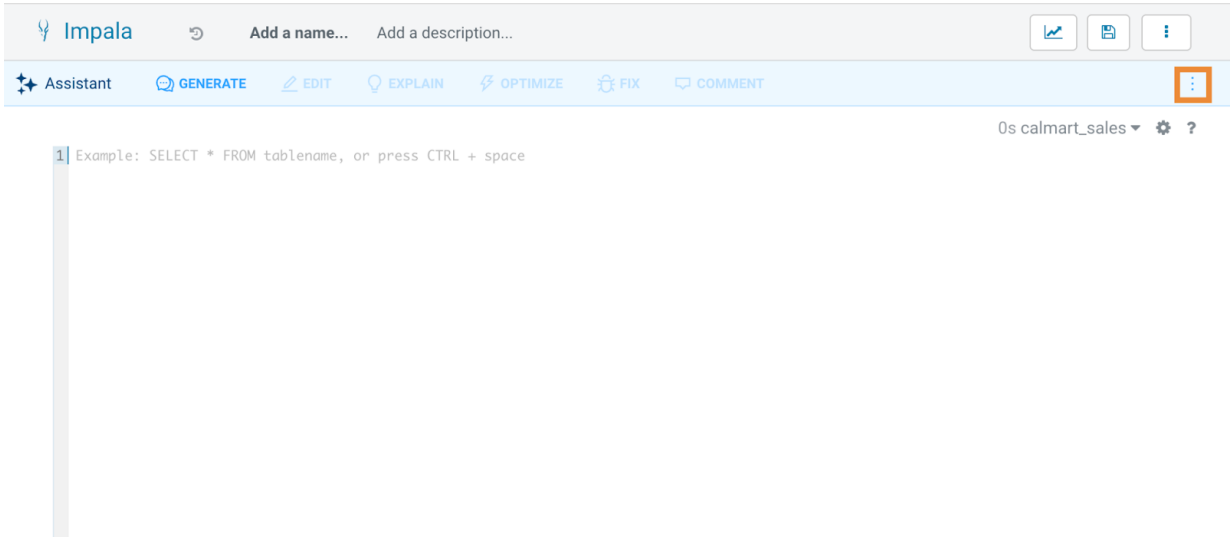
- Provides the Assistant with specific organizational logic to reduce errors in generated SQL code.
- Enables seamless querying across different systems by defining how disparate datasets relate to each other.
- Automates the inclusion of complex context, reducing the need for repetitive manual input.

Limitations

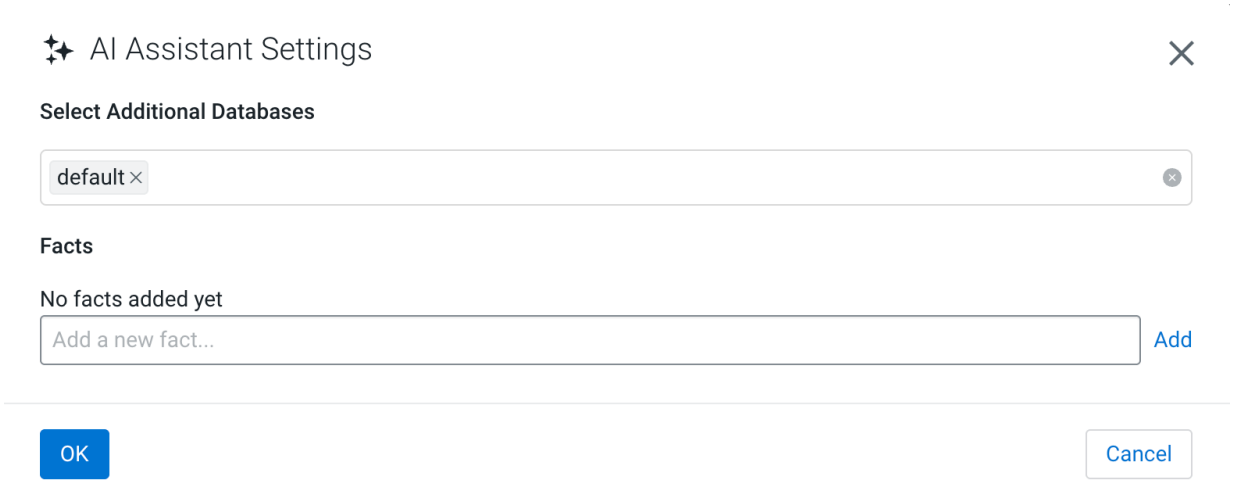
- The system does not currently enforce a character limit for facts. Including excessively long instructions can result in model hallucinations.
- You can define a list of facts that the SQL AI Assistant includes as global context when generating your queries.

Procedure

1. Click Assistant on the Data Explorer SQL editor, then open the AI Assistant Settings.



2. Define your instruction in the Facts section and click Add.



3. Click Ok to apply the changes.

Cloudera Data Explorer (Hue) SQL AI API overview (Preview)

Learn about the SQL AI Assistant, which leverages Large Language Models to assist with various SQL tasks. The SQL AI Assistant is now available through a RESTful API, enabling programmatic access and integration with external tools.

Note: This feature is in Technical Preview and is not ready for production deployments. Cloudera recommends trying this feature in test or development environments and encourages you to provide feedback on your experiences.

Access method

The SQL AI Assistant is accessible through a RESTful API. The API provides endpoints for clients and external tools to access SQL AI features.

Limitations

- This release only supports the generate task feature.
- Large schemas can result in slower response times.
- The workflow is actively being optimized to improve accuracy and performance.

SQL AI REST API Documentation

Learn more comprehensive details about the AI Assistant API.

Base URL

https://<hue-server-domain>/api/v1/ai/assistant

API Endpoint

/api/v1/ai/assistant

Method: POST

Request Headers

The following table lists the headers required for making requests

Header	Value	Description
Authorization	Bearer <your_cdp_access_token>	CDP access token for authentication
Content-Type	application/json	Request payload format

Request Body Parameters

The following table describes the fields in the request body:

Parameter	Type	Required	Description	Example
task	string	Yes	SQL AI functionality to perform. Generate is the only supported feature.	"generate"
dialect	string	Yes	SQL dialect for generation. Supported: hive, impala, trino.	"impala"
object_selectors	array	No	Optional array of strings identifying database and tables to constrain task scope. Note: If you do not define specific objects, the system automatically selects the most relevant databases and tables.	["sales_db.orders", "hr_db.employees"]
input	string	Yes	Natural language or Natural Query Language (NQL) describing the input.	"Retrieve count of active users grouped by country"

Parameter	Type	Required	Description	Example
cache_mode	string	No	Cache policy for schema loading. Values: incremental (reuse loaded data), refresh (reload always).	"incremental"
db_top_k	integer	No	Number of top databases to consider for SQL generation. Default: 3.	3
table_top_k	integer	No	Number of top tables per db to consider. Default: 10.	10

Example Request

```
POST /api/v1/ai/assistant HTTP/1.1
Host: hue.company.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json
{
  "task": "generate",
  "dialect": "impala",
  "object_selectors": ["sales_db.orders", "hr_db.employees"],
  "input": "Get the count of active users grouped by country",
  "cache_mode": "incremental",
  "db_top_k": 2,
  "table_top_k": 5
}
```

Response Body

The following table describes the fields in the response body:

Field	Type	Description	Example
sql	string	The generated or edited SQL query	"SELECT country, COUNT(*) FROM sales_db.orders WHERE status = 'active' GROUP BY country;"
assumptions	array	List of assumptions the AI made about input, schema, or intent	["Assuming 'active' status means status='active'."]
tables_used	array	List of database tables that the SQL references	["sales_db.orders"]

Example Response

```
{
  "response": "Generated SQL to get counts by country of active users.",
  "sql": "SELECT country, COUNT(*) as active_user_count FROM sales_db.orders
 WHERE status = 'active' GROUP BY country;",
  "assumptions": [
    "Assuming 'active' status means status='active'."
  ],
  "tables_used": [
    "sales_db.orders"
  ]
}
```

Error Handling

The following table lists common HTTP status codes and descriptions to help with error handling:

HTTP Status Code	Reason	Description
400 Bad Request	Invalid or missing parameter	Returned when a required field is missing or invalid values are provided
401 Unauthorized	Missing or invalid token	Authentication failure; token missing, expired or invalid
403 Forbidden	Permission denied	User lacks permission to access endpoint or resource
500 Internal Server Error	Server error	Unexpected error occurred during processing

Enabling SQL AI Assistant API via UI

You can enable the SQL AI Assistant API capabilities via Data Explorer UI by adding the required configuration to the Hue safety valve.

About this task

Follow these steps to enable the SQL AI Assistant API capabilities in Data Explorer UI.

Procedure

1. After successful model deployment, log in to Cloudera Data Warehouse service.
2. Navigate to the Virtual Warehouses tab and select the desired Virtual Warehouse to enable.
3. Click Details Configurations Data Explorer (Hue) .
4. From the Configuration files dropdown, select hue-safety-valve.
5. Add the following lines:

```
[[llm_provider]]
name='azure'
model_name='gpt-4o-hue'
api_version='2024-12-01-preview'
base_url='https://hue-ai-xxxx.openai.azure.com'
token='xxxxxxxx'
assistant_api_enabled=true
```

6. Click Apply Changes.
7. Restart the service.

Cloudera Data Explorer (Hue) SQL AI Assistant FAQ

A collection of frequently asked questions about Data Explorer SQL AI Assistant.

- General Questions on page 35
- Using SQL AI Assistant on page 36
- AI Models and Security on page 36
- Configuration and Setup on page 36

General Questions

What is the SQL AI Assistant in Data Explorer?

The SQL AI Assistant in Data Explorer is an AI-powered tool integrated into the SQL editor that helps users generate, edit, optimize, fix, and summarize SQL queries using natural language. It leverages large language models (LLMs) to assist data analysts in making SQL development faster, easier, and less error-prone.

Which SQL dialects does the SQL AI Assistant support?

Multiple SQL dialects are supported, including Hive, Impala, and Trino.

Using SQL AI Assistant

How do I launch the SQL AI Assistant?

Click the ✨ Assistant to expand the SQL AI toolbar, which provides buttons for generating, editing, explaining, optimizing, and fixing SQL statements.

For more information, see About setting up the Cloudera Data Explorer (Hue) SQL AI Assistant.

What happens when I click 'Generate' in the SQL AI Assistant?

Clicking "Generate" allows you to enter a natural language query, which the assistant converts into an SQL query. The generated SQL is presented along with assumptions made by the LLM.

For more information, see Generating SQL from natural language in Cloudera Data Explorer (Hue).

Can I create a query that joins multiple databases when using the Data Explorer SQL AI Assistant?

Yes, the Data Explorer SQL AI Assistant supports multi-database queries. You can select multiple databases in the AI Assistant Settings pop-up, allowing you to create queries that join tables across different databases.

For more information, see Multi database support for SQL query.

How does the 'Edit' function work?

The "Edit" button allows users to modify an active SQL statement. If an NQL comment precedes the statement, it can be reused by pressing Tab. Users can also enter new instructions for modifications.

For more information, see Editing the query in natural language in Cloudera Data Explorer (Hue).

How does the 'Explain' function work?

The "Explain" button provides a natural language summary and explanation of the selected SQL query, which can be inserted as a comment in the editor.

For more information, see Getting an explanation of a SQL query in natural language in Cloudera Data Explorer (Hue).

AI Models and Security

Which AI models does the SQL AI Assistant support?

The Data Explorer SQL AI Assistant supports Cloudera AI Workbench and Cloudera AI Inference service, along with several third-party services. Using the Cloudera integrations enhances the Data Explorer SQL AI Assistant by enabling the use of private models hosted within Cloudera-managed infrastructure. This ensures enhanced security and privacy while leveraging GenAI for the Data Explorer SQL-related tasks. For more information, see Supported services.

How does the SQL AI Assistant handle data privacy?

The SQL AI Assistant shares only the data that the logged-in user is authorized to access. It uses a Retrieval Augmented Generation (RAG)-based architecture to limit the number of tables sent per request. However, there is currently no way to explicitly exclude certain tables from being shared.

Configuration and Setup

What AI services are supported for integration?

Supported services include:

- Cloudera AI Workbench
- Cloudera AI Inference service
- Microsoft Azure OpenAI
- OpenAI API
- Amazon Bedrock

Is it necessary to train the Data Explorer SQL AI Assistant on the database schema before using it?

Training is not necessary for the Data Explorer SQL AI Assistant. Once connected, the assistant can begin querying immediately. However, in cases where table names are similar or column names are repeated across multiple tables, it is recommended to ensure that the database metadata is well-maintained. Specifically:

- Table and column comments should be clear and descriptive as they are used by the assistant for context and disambiguation.
- Consider using distinct table and column naming conventions to reduce confusion.
- Ensure that schema and table relationships are properly defined, as this helps the assistant understand context.

Does the Data Explorer SQL AI Assistant only work based on the prompt, or does it also have information about the present databases? For example, can it correct the name of a table or database if I write it incorrectly within the prompt?

Yes, in most cases, the Data Explorer SQL AI Assistant can catch issues with your prompt, such as misspelled table names. This is because the user prompt is packaged alongside information about the SQL dialect in use, the database metadata, and sample rows from relevant tables. However, if another table exists with the misspelled name, the assistant might be misdirected.

About deploying the shared Cloudera Data Explorer (Hue) service

Cloudera Data Warehouse allows you to deploy a shared Data Explorer service at an environment level. Learn about the advantages and limitations of deploying a shared Data Explorer service and some FAQs that can help you understand more about the feature.

Advantages of deploying a shared Data Explorer service

By deploying a shared Data Explorer service, you can manage costs by keeping only those Virtual Warehouses running that your users need at that time. Data analysts only need to know or bookmark one Data Explorer instance URL and can run queries on any Virtual Warehouses available to them.

Each shared Data Explorer service instance has its own database where queries and query history are saved. Moreover, the shared Hue service remains active as long as the environment is active.

Limitations

When you use the Importer to create tables from files in Data Explorer, by default, Data Explorer creates a Hive table if Hive is available and uses the first Hive Virtual Warehouse that was created. You cannot select a Virtual Warehouse using which you want to create a table by importing a file. To create an Impala table using the Importer, you must first select the editor type as Impala and then click + on the Table Browser.

Access control for the shared Cloudera Data Explorer (Hue) service

You can specify user groups you created in the Cloudera Management Console, similar to how you specified them while creating the Virtual Warehouses.

When you specify user groups while creating the shared Data Explorer instance or Virtual Warehouses, the subset of users who have access to the Data Explorer instance as well as the Virtual Warehouse can submit queries through that Virtual Warehouse instance.

If you do not specify user groups while creating a shared Data Explorer instance, then all users within your organization can access the Data Explorer UI. If you do not specify user groups for a Virtual Warehouse, all users within your organization can submit queries through that Virtual Warehouse.

As a best practice, specify user groups while creating Data Explorer and Virtual Warehouse instances so that specific users have access to specific compute resources.

Key differences in database management approach for Virtual Warehouse-level Data Explorer and shared Data Explorer service

All Data Explorer instances linked to a Database Catalog through Virtual Warehouses within a Cloudera Data Warehouse environment share a single database. The Data Explorer database is not deleted unless you deactivate the environment. If you delete a Virtual Warehouse and create a new one, the Data Explorer instance linked to that Virtual Warehouse continues to display old query history and saved queries.

Each shared Data Explorer service instance has its own Data Explorer database. Cloudera Data Warehouse does not delete the Data Explorer database when you delete the shared Data Explorer service instance. The Data Explorer database exists in the backend until a database administrator manually deletes it. Each Data Explorer database is named after the shared Data Explorer service name. In case you have deleted a shared Data Explorer service instance, you can reuse the Data Explorer database by specifying the name of the Data Explorer instance you deleted. This brings back the query history and saved queries.

Cloudera Data Warehouse provides you with a one-time option to copy the Data Explorer database content from the Data Explorer database linked to a Database Catalog to the shared Data Explorer service database while creating a new shared Data Explorer service instance. The data between the two databases is not synchronized after the initial copy event.

Creating a shared Cloudera Data Explorer (Hue) instance

To deploy Data Explorer at the environment level, you can create any number of Data Explorer instances. Each Data Explorer instance has its own database. The Data Explorer instances deployed at the environment level do not share query history or saved queries.

Procedure

1. Log in to the Cloudera Data Warehouse service as a DWAdmin
2. Click the Shared Data Explorer Service option from the left navigation pane of the Cloudera Data Warehouse UI.
3. Create a shared Data Explorer instance by clicking ADD NEW on the Shared Data Explorer Service page. The Create Shared Data Explorer Service modal is displayed.
4. Specify a name for your Data Explorer instance and select an environment from the drop-down menu.
   a) Select a size for the shared Data Explorer instance from the Size drop-down menu. This indicates the number of Data Explorer backend pods you want to create.
   b) Select one of the following options from the Select the Data Explorer database initialization strategy drop-down menu to initialize a database for this Data Explorer instance.
      - Reuse if Data Explorer data is present: Select this option to use an existing Data Explorer database for your new shared Data Explorer instance.
      - Copy Virtual Warehouse database: Select this option to copy the contents of the Data Explorer database within a Database Catalog into the Shared Data Explorer Service database. This helps you to copy the shared queries and query history.

        Note: This is a one-time copy operation. After the data is copied from the Database Catalog to the shared Data Explorer database, the data between the two databases is not synchronized.

- Flush the current database content: Select this option to create a new database for the shared Data Explorer instance. Select this option to recreate a Shared Data Explorer Service instance with the same name but a fresh database. The existing database linked to the Data Explorer instance is dropped. You can optionally specify user groups and tags in the same way you specify while creating a Virtual Warehouse for fine-grained access control.

5. Click Create.

   A new shared Data Explorer service instance is created.

What to do next

Click Editor to open Data Explorer. You can select the Virtual Warehouse you want to use from the Virtual Warehouse drop-down menu on the Data Explorer web interface.

Rebuilding a shared Cloudera Data Explorer (Hue) service

The rebuild operation deletes and recreates the Data Explorer pods while preserving Data Explorer image version and configurations. By rebuilding the Data Explorer service, you fix pods that are in a bad state, thereby improving performance.

Procedure

1. Log in to the Cloudera Data Warehouse service as a DWAdmin.
2. Click the Shared Data Explorer Service option from the left navigation pane of the Cloudera Data Warehouse UI.
3. Locate the Shared Data Explorer Service instance you want to rebuild and click the More icon Rebuild .
4. Review the message on the Review Shared Data Explorer Service modal and click Review Shared Data Explorer Service.

   The “Rebuild in progress” message is displayed.

Upgrading a shared Cloudera Data Explorer (Hue) instance

If you are on an older version of the Shared Data Explorer Service, you can upgrade the Data Explorer image version by upgrading Data Explorer from the Shared Data Explorer Service page in Cloudera Data Warehouse.

Procedure

1. Log in to the Cloudera Data Warehouse service as a DWAdmin.
2. Click the Shared Data Explorer Service option from the left navigation pane of the Cloudera Data Warehouse UI.
3. Locate the Shared Data Explorer Service instance you want to upgrade and click Upgrade. The Upgrade Shared Data Explorer Service modal is displayed.
4. Click Upgrade.

   The Shared Data Explorer Service is upgraded to the latest available image version.

FAQ for shared Cloudera Data Explorer (Hue) service

A collection of frequently asked questions about deploying a Shared Data Explorer service.

Can I still use Data Explorer, which is deployed at the Virtual Warehouse level?

Yes, you can continue to access and use Data Explorer from a particular Virtual Warehouse even after deploying the Shared Data Explorer Service at the environment level.

Can I create more than one shared Data Explorer service instance?

Yes, you can create any number of Shared Data Explorer Service instances. However, Cloudera recommends deploying a single Data Explorer instance unless isolating saved queries is a requirement. When you create multiple shared Data Explorer instances, each instance has its own database. The Shared Data Explorer Service instances do not share query history or saved queries.

Can I view queries submitted from other BI tools?

Data Explorer superusers and administrators can view all queries submitted from all Virtual Warehouses linked to a Database Catalog. Other logged-in users can view only their queries on the Impala Queries and Hive Queries tabs.

Where can I specify advanced Data Explorer configurations (safety valve) for the shared Data Explorer instance?

On the Shared Data Explorer Service page, click the more options icon > Edit corresponding to the Data Explorer instance that you want to configure, go to the CONFIGURATIONS tab, and select hue-safety-valve from the Configuration files drop-down menu.