Cloudera Runtime 7.3.2

Administering Hue

Date published: 2020-07-28
Date modified: 2026-03-31

https://docs.cloudera.com/

Legal Notice

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.

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

Cloudera Data Explorer (Hue) configuration files

Hue roles are configured with the following three configuration files: hue.ini, hue_safety_valve.ini, and hue_safety_valve_server.ini.

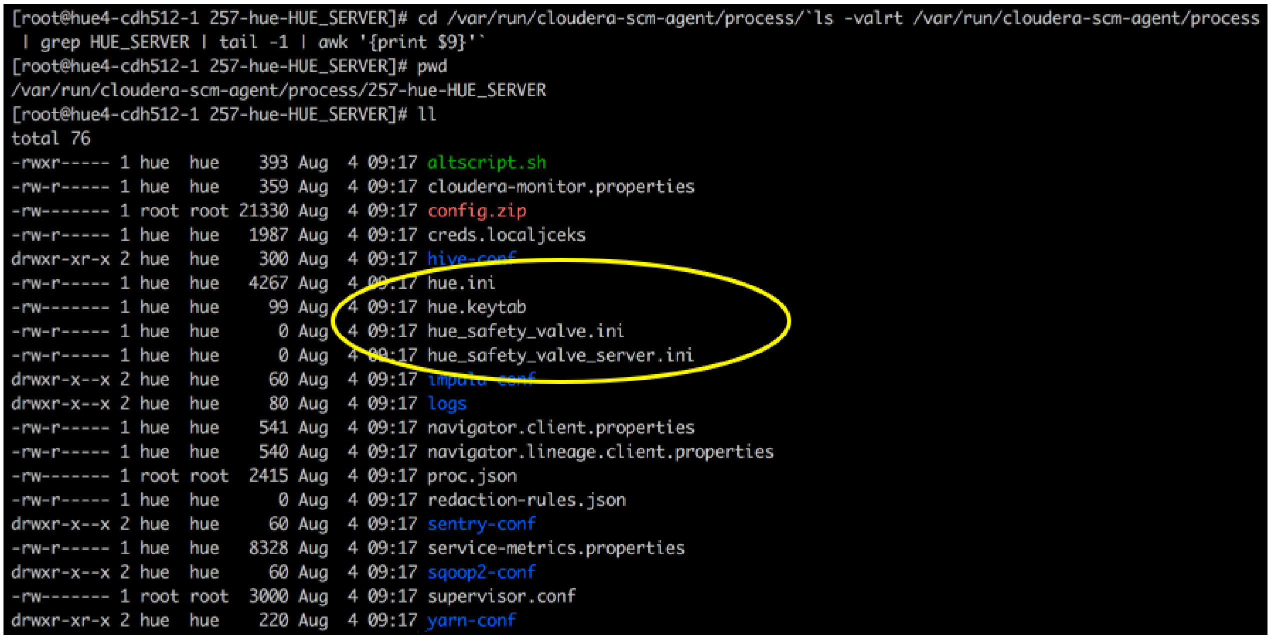
The hue.ini file is the first file that is auto-generated when you add the Hue service to your Cloudera cluster using Cloudera Manager. The hue_safety_valve.ini file is used to override bugs or configure properties that are missing from the Cloudera Manager UI. The hue_safety_valve_server.ini file is used to configure only the Hue role, and not the full Hue service (which includes the Hue Load Balancer). The hue_safety_valve_server.ini file is not used in practice.

Cloudera recommends that you do not edit the .ini files from the command line because they are stored in dynamic directories named by process ID and populated from the Cloudera Manager database. To add configurations that you cannot add directly from the Cloudera Manager UI, such as Authentication Backend for SAML, use the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field under the Hue service configurations in Cloudera Manager.

Run the following command to view the .ini files per process ID:

```
ls -ltr /var/run/cloudera-scm-agent/process/`ls -valrt /var/run/cloudera-scm
-agent/process | grep HUE_SERVER | tail -1 | awk '{print $9}'`
```

Figure 1: Terminal showing Hue configuration files



```
[root@hue4-cdh512-1 257-hue-HUE_SERVER]# cd /var/run/cloudera-scm-agent/process/`ls -valrt /var/run/cloudera-scm-agent/process
 | grep HUE_SERVER | tail -1 | awk '{print $9}'`
[root@hue4-cdh512-1 257-hue-HUE_SERVER]# pwd
/var/run/cloudera-scm-agent/process/257-hue-HUE_SERVER
[root@hue4-cdh512-1 257-hue-HUE_SERVER]# ll
total 76
-rwxr----- 1 hue  hue    393 Aug  4 09:17 altscript.sh
-rw-r----- 1 hue  hue    359 Aug  4 09:17 cloudera-monitor.properties
-rw------- 1 root root 21330 Aug  4 09:17 config.zip
-rw-r----- 1 hue  hue   1987 Aug  4 09:17 creds.localjceks
drwxr-xr-x 2 hue  hue    300 Aug  4 09:17 hive-conf
-rw-r----- 1 hue  hue   4267 Aug  4 09:17 hue.ini
-rw------- 1 hue  hue     99 Aug  4 09:17 hue.keytab
-rw-r----- 1 hue  hue      0 Aug  4 09:17 hue_safety_valve.ini
-rw-r----- 1 hue  hue      0 Aug  4 09:17 hue_safety_valve_server.ini
drwxr-x--x 2 hue  hue     60 Aug  4 09:17 impala-conf
drwxr-x--x 2 hue  hue     80 Aug  4 09:17 logs
-rw-r----- 1 hue  hue    541 Aug  4 09:17 navigator.client.properties
-rw-r----- 1 hue  hue    540 Aug  4 09:17 navigator.lineage.client.properties
-rw------- 1 root root  2415 Aug  4 09:17 proc.json
-rw-r----- 1 hue  hue      0 Aug  4 09:17 redaction-rules.json
drwxr-x--x 2 hue  hue     60 Aug  4 09:17 sentry-conf
-rw-r----- 1 hue  hue   8328 Aug  4 09:17 service-metrics.properties
drwxr-x--x 2 hue  hue     60 Aug  4 09:17 sqoop2-conf
-rw------- 1 root root  3000 Aug  4 09:17 supervisor.conf
drwxr-xr-x 2 hue  hue    220 Aug  4 09:17 yarn-conf
```

The process directory for any given role is mirrored in Cloudera Manager. Go to Hue Instances , select a role such as Hue Server, and then click the Processes tab.

Related Information

hue.ini

Cloudera Data Explorer (Hue) configurations in Cloudera Runtime

Cloudera Manager exposes properties that allow you to insert custom configuration text into XML configuration, property, and text files, or into an environment. The configuration snippet mechanism is intended for use in cases where there is a configuration setting that is not exposed as a configuration property in the Cloudera Managerweb interface. Configuration snippets generally override normal configuration. You can configure custom properties for Hue by specifying them in the Advanced Configuration Snippet (Safety valve) field in Cloudera Manager.

The following table lists the safety valve parameters supported by Hue in Cloudera Runtime:

Parameter	Description

`[[desktop]]`
`app_blacklist`
	Used to add or remove applications, such as the File Browser, Impala, Hive, Oozie, and so on from the blocked list of applications.

`[desktop]`
`   [[session]]`
`    ttl=[***NUMBER-OF-SECONDS***]`
	Used to configure the duration of a user session. The ttl property determines the length of time that the cookie with the user's session ID lives before expiring. After the ttl setting is reached, the user's session expires whether it is active or not.
	The default setting for ttl is 1,209,600 seconds, which equals two weeks.

`[jobbrowser]`
`[[query_store]]`
`server_url=[***QUERY-PROCESSOR-URL***]`
	Used to display the **Queries** tab for Hive and Impala on the **Job Browser** page. This configuration is enabled by default and is not exposed in the Hue safety valve.
	However, to hide the **Queries** tab, you can override the configuration by adding the following lines in Cloudera Manager Clusters Hue Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini :
	`[jobbrowser]`
	`[[query_store]]`
	`is_enabled=false`
	To enable the **Queries** tab, set is_enabled to true.

`[aws]`
`[[aws_accounts]]`
`[[[default]]]`
`access_key_id=[***AWS-ACCESS-KEY***]`
`secret_access_key=[***SECRET-ACCESS-KEY***]`
`region=[***AWS-REGION***]`
	Used to enable the S3 File Browser for Hue without IDBroker.

`[aws]`
`has_iam_detection=true`
`[[aws_accounts]]`
`[[[default]]]`
`region=[***AWS-REGION***]`
	Used to enable the S3 File Browser for Hue with IDBroker.

`[desktop]`
`[azure]`
`  [[azure_accounts]]`
`    [[[default]]]`
`      client_id=[***AZURE-ACCOUNT-CLIENT-ID***]`
`      client_secret=[***AZURE-ACCOUNT-CLIENT-SECRET***]`
`      tenant_id=[***AZURE-ACCOUNT-TENANT-ID***]`
 	
`    [[abfs_clusters]]`
`      [[[default]]]`
`  fs_defaultfs=abfs://[***CONTAINER-NAME***]@[***AZURE-STORAGE-ACCOUNT-NAME***]>.dfs.core.windows.net`
`              webhdfs_url=https://[***AZURE-STORAGE-ACCOUNT-NAME***].dfs.core.windows.net/`
	Used to enable the ABFS File Browser for Hue without IDBroker.

`[desktop]`
`[azure]`
	Used to enable the ABFS File Browser for Hue with IDBroker.

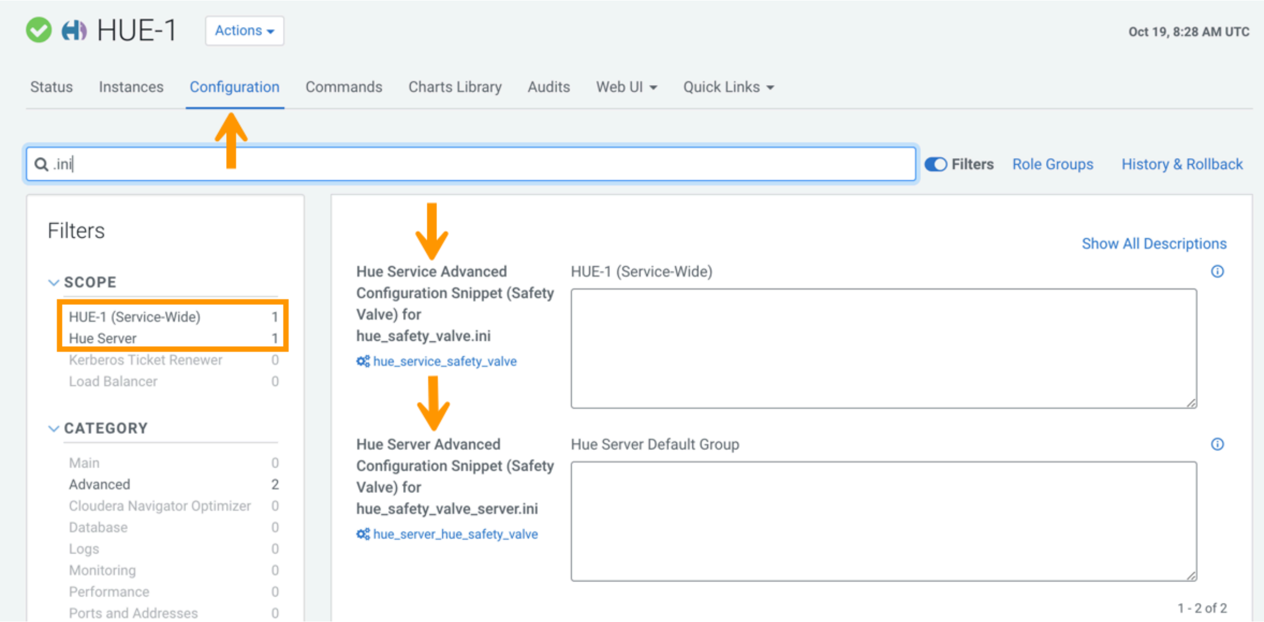
Contact Cloudera Support if you are required to use a configuration snippet that is not explicitly documented.

Introduction to Cloudera Data Explorer (Hue) Advanced Configuration Snippet (Safety valves)

To customize and configure Hue properties, such as authentication or enabling S3 or ABFS browsers, and so on that you cannot directly configure from the Cloudera Manager UI, you can use the Advanced Configuration Snippet field under the Hue cluster configurations in Cloudera Manager.

Advanced safety valves

The field for hue_safety_valve.ini is service-wide and affects all Hue roles listed on the Instances tab. The field for hue_safety_valve_server.ini only affects Hue Server role instances.

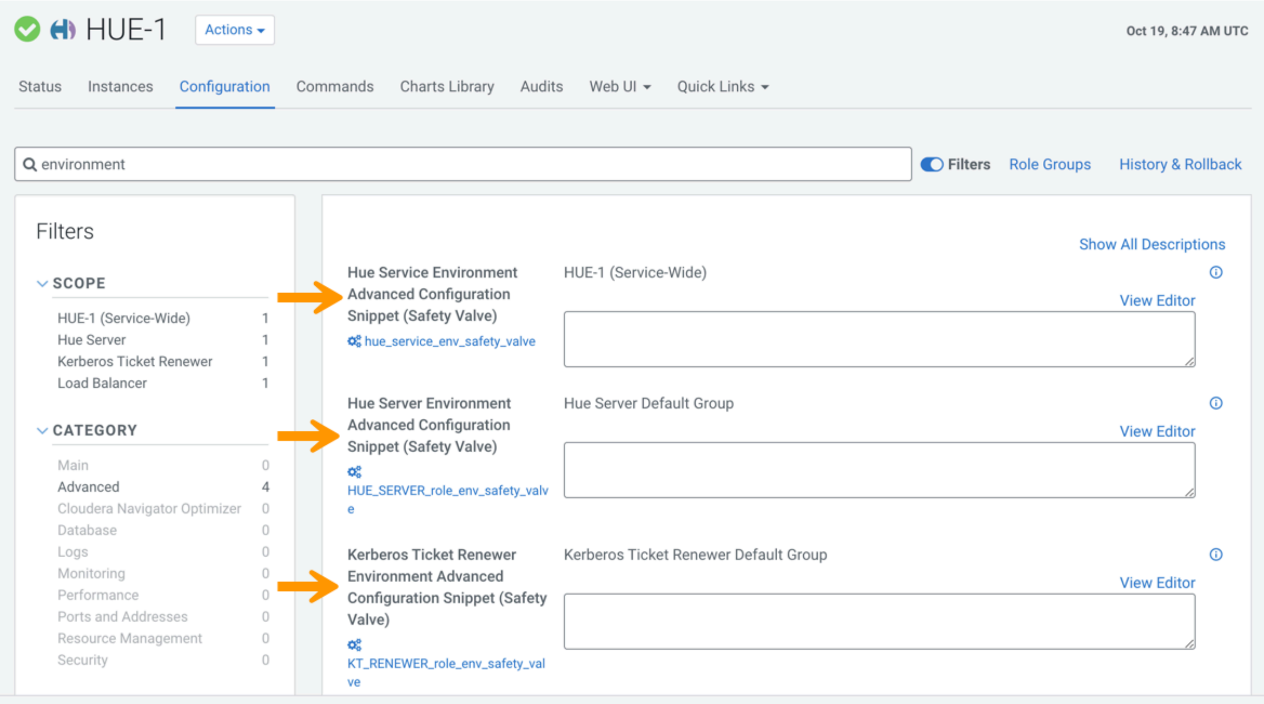


Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini
hue_service_safety_valve
HUE-1 (Service-Wide)

Hue Server Advanced Configuration Snippet (Safety Valve) for hue_safety_valve_server.ini
hue_server_hue_safety_valve
Hue Server Default Group

Environment safety valves

Environment safety-valves let you configure environment variables across the service or for specific role instances. These safety-valves lack headers.



SCOPE

HUE-1 (Service-Wide)	1
Hue Server	1
Kerberos Ticket Renewer	1
Load Balancer	1

CATEGORY

Main	0
Advanced	4
Cloudera Navigator Optimizer	0
Database	0
Logs	0
Monitoring	0
Performance	0
Ports and Addresses	0
Resource Management	0
Security	0

Hue Service Environment Advanced Configuration Snippet (Safety Valve) — hue_service_env_safety_valve — HUE-1 (Service-Wide)

Hue Server Environment Advanced Configuration Snippet (Safety Valve) — HUE_SERVER_role_env_safety_valve — Hue Server Default Group

Kerberos Ticket Renewer Environment Advanced Configuration Snippet (Safety Valve) — KT_RENEWER_role_env_safety_valve — Kerberos Ticket Renewer Default Group

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
- vLLM

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.

Secure Approach for Passing a Token in Cloudera Manager

This approach enables secure token management through Cloudera Manager. You first encode the credentials and then add the encoded value to the AI Interface API Secret property.

Procedure

1. Log on to Cloudera Manager and go to Data Explorer (Hue) Configuration .
2. In the search text box, enter AI Interface API Secret and add the token value you want to encode.
3. Add the following configuration in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [desktop]
   [[ai_interface]]
   service='azure'
   service_version='2024-02-15-preview'
   model_name='[***DEPLOYMENT-NAME***]'
   base_url="https://[***RESOURCE***].openai.azure.com/"
   ```

4. Click Save Changes.

Open approach for passing a token

In this approach, you specify the token value in the Hue Advanced Configuration Snippet field in Cloudera Manager.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.

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***]'
```

vLLM

```
[[ai_interface]]
 service='vllm'
 model_name='[***Place MODEL name here***]'
 base_url="https://[***RESOURCE***]/v1"
 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 succesuccessful completion of model deployment, log in to the Cloudera Management Console as an Administrator.
2. Go to Environments and select your environment.
3. Go to the Data Lake tab and click on the CM URL to open Cloudera Manager.
4. Go to Clusters Data Explorer (Hue) service Configuration and select add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

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

5. 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.

Note: By default, CDP_TOKENS expire after 1 hour. Until authentication via API keys becomes available, it is recommended to create a CDP_TOKEN as a system user and set a higher value for the Data Lake Knox token TTL configuration.

Procedure

1. Upon installing and setting up the Cloudera AI Inference service, log in to the Cloudera Management Console as an Administrator.
2. Go to Environments and select your environment.
3. Go to the Data Lake tab and click on the CM URL to open Cloudera Manager.
4. Go to Clusters Data Explorer (Hue) service Configuration and select add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

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

5. 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 installing and setting up the Cloudera AI Inference service, log in to the Cloudera Management Console as an Administrator.
2. Go to Environments and select your environment.
3. Go to the Data Lake tab and click on the CM URL to open Cloudera Manager.
4. Go to Clusters Data Explorer (Hue) service Configuration and select add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

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

5. 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.

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.

Procedure

1. Upon installing and setting up the Cloudera AI Inference service, log in to the Cloudera Management Console as an Administrator.
2. Go to Environments and select your environment.
3. Go to the Data Lake tab and click on the CM URL to open Cloudera Manager.
4. Go to Clusters Data Explorer (Hue) service Configuration and select add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

```
[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.
5. 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 installing and setting up the Cloudera AI Inference service, log in to the Cloudera Management Console as an Administrator.
2. Go to Environments and select your environment.
3. Go to the Data Lake tab and click on the CM URL to open Cloudera Manager.
4. Click Save Changes.

Results

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

Configure SQL AI Assistant using vLLM

vLLM is an open-source library for LLM inference and serving. This topic describes how to set up SQL AI Assistant with a model hosted using vLLM.

Before you begin

To know more about the installation and its requirements, see the vLLM documentation.

Procedure

1. Log in to the Cloudera Management Console as an Administrator.
2. Go to Environments and select your environment.
3. Go to the Data Lake tab and click on the CM URL to open Cloudera Manager.
4. Click Save Changes.

Results

You see ✨ Assistant on the Data Explorer SQL editor, and the SQL AI Assistant connects to the specified model hosted using vLLM.

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 Manager.

List of service and model-related configurations

You can configure the AI services and models you want to use by going to Cloudera Manager Clusters Data Explorer (Hue) service Configurations Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini 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)

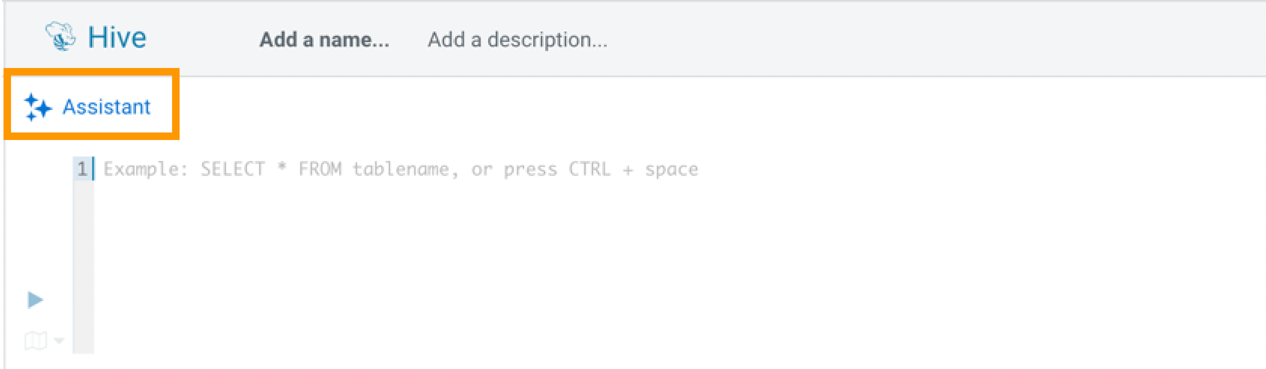
A SQL AI Assistant has been integrated into Data Explorer with the capability to leverage the power of Large Language Models (LLMs) for various SQL tasks. It helps you to create, edit, optimize, fix, and succinctly summarize queries using natural language and makes SQL development faster, easier, and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

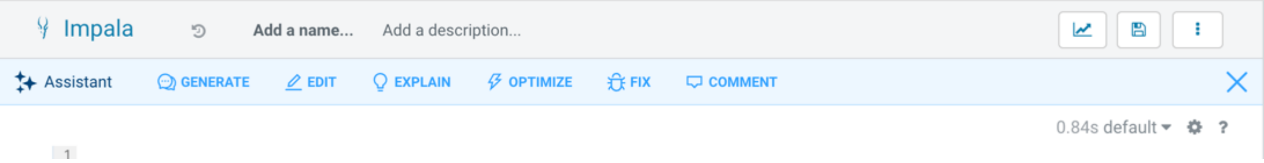
Procedure

Click ✨ Assistant on the Data Explorer (Hue) SQL editor:



Results

The following options are displayed:

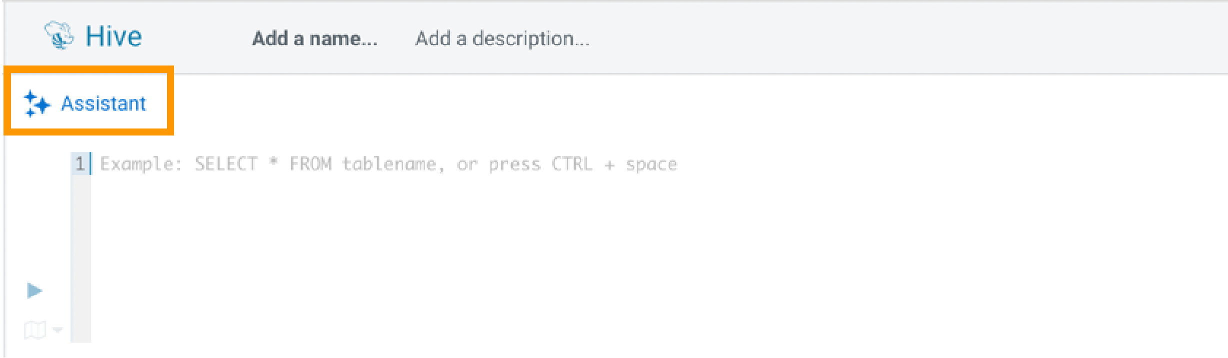


Generating SQL from natural language in Cloudera Data Explorer (Hue)

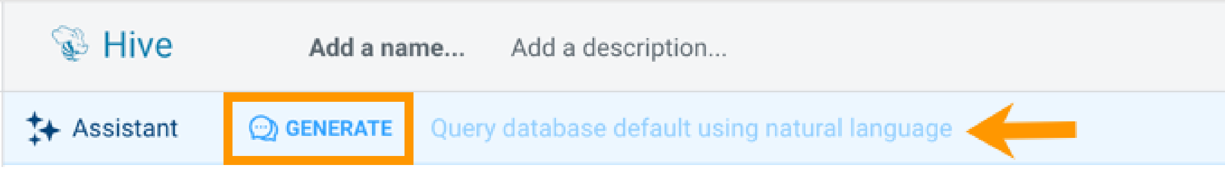
The SQL AI Assistant in Cloudera Data Warehouse helps you to generate SQL queries by entering a prompt in natural language. You can then insert the generated SQL in the Data Explorer SQL editor and run it as usual.

Procedure

1. Click ✨ Assistant on the Data Explorer (Hue) SQL editor:



2. Click GENERATE.



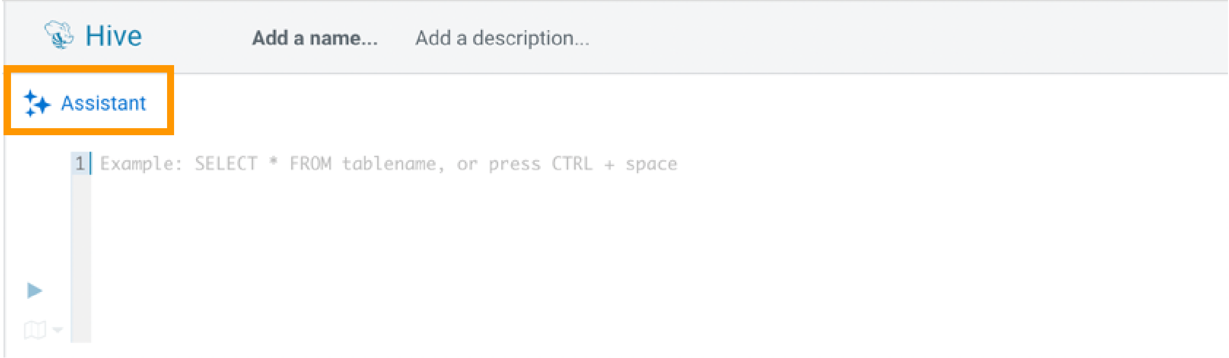
A SQL query is generated based on your input prompt. Click Insert to insert the query into the editor and run it.

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

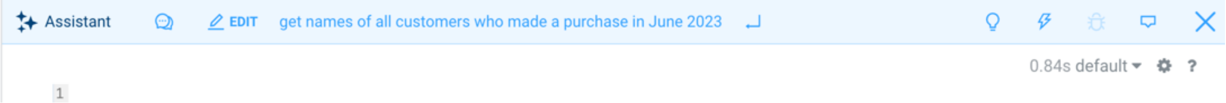
You can edit a query in natural language and generate it again to fine-tune your results using the Data Explorer SQL AI Assistant. You can then insert the SQL query into the editor and run it as usual.

Procedure

1. Click ✨ Assistant on the Data Explorer (Hue) SQL editor:



2. Click EDIT.



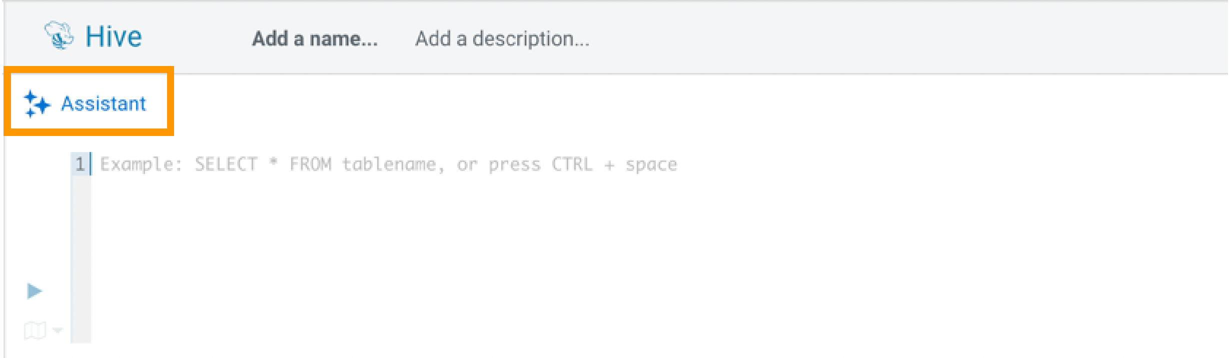
Select the query from the list and edit it as needed. Then press enter to regenerate the SQL.

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

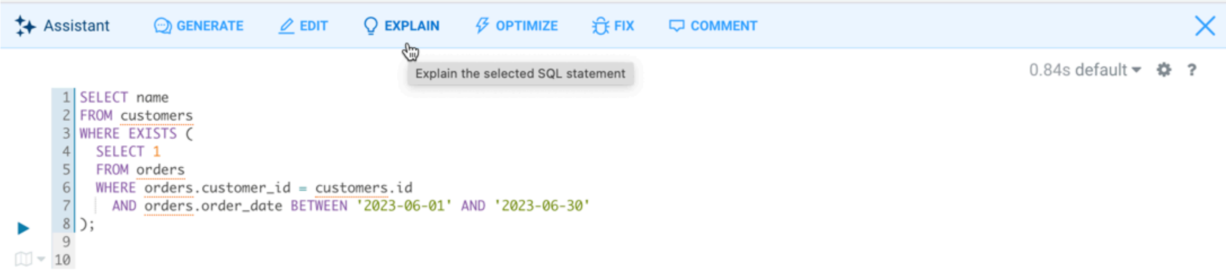
You can use the SQL AI Assistant to breakdown and understand a complex SQL query in natural language.

Procedure

1. Insert a SQL query in the Data Explorer editor that you would like to understand in natural language.
2. Click ✨ Assistant on the Data Explorer (Hue) SQL editor:

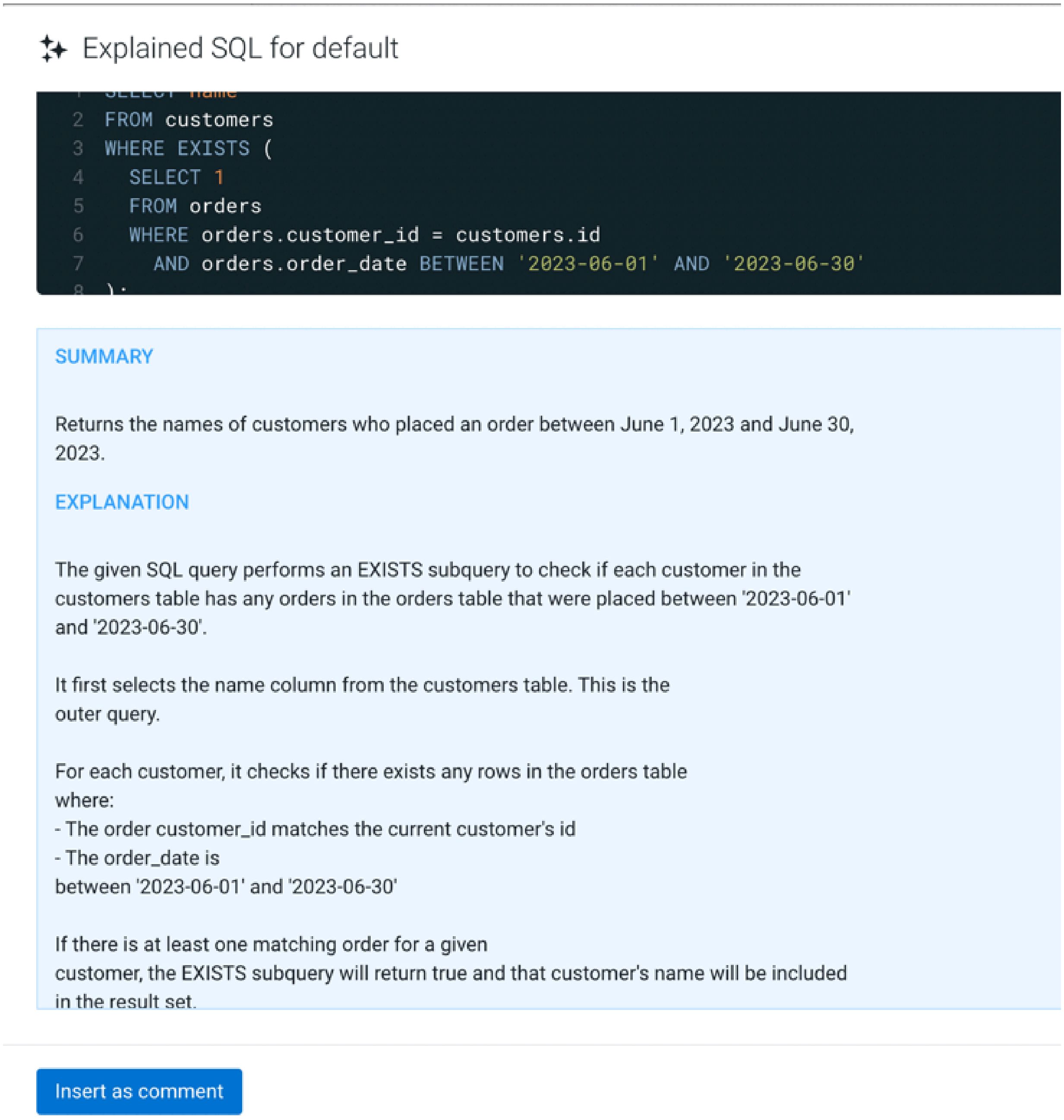


3. Click EXPLAIN.



```
SELECT name
FROM customers
WHERE EXISTS (
  SELECT 1
  FROM orders
  WHERE orders.customer_id = customers.id
    AND orders.order_date BETWEEN '2023-06-01' AND '2023-06-30'
);
```

The LLM generates the explanation of the SQL query.



Explained SQL for default

```
SELECT name
FROM customers
WHERE EXISTS (
  SELECT 1
  FROM orders
  WHERE orders.customer_id = customers.id
    AND orders.order_date BETWEEN '2023-06-01' AND '2023-06-30'
);
```

SUMMARY

Returns the names of customers who placed an order between June 1, 2023 and June 30, 2023.

EXPLANATION

The given SQL query performs an EXISTS subquery to check if each customer in the customers table has any orders in the orders table that were placed between '2023-06-01' and '2023-06-30'.

It first selects the name column from the customers table. This is the outer query.

For each customer, it checks if there exists any rows in the orders table where:
- The order customer_id matches the current customer's id
- The order_date is between '2023-06-01' and '2023-06-30'

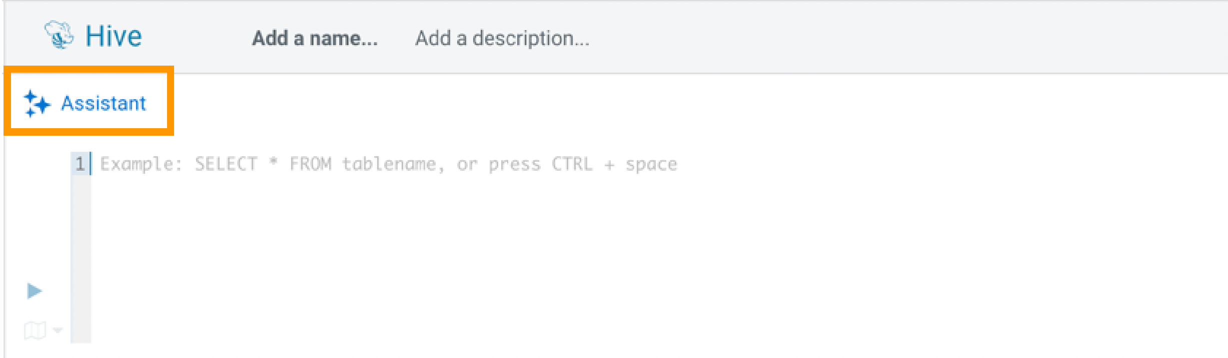
If there is at least one matching order for a given customer, the EXISTS subquery will return true and that customer's name will be included in the result set.

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.

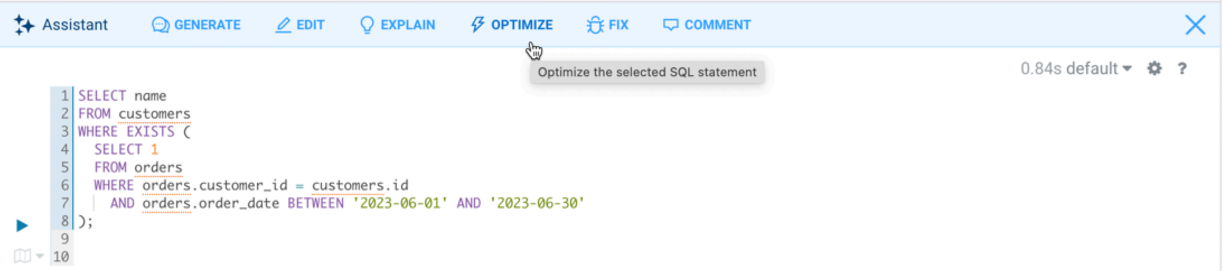
Procedure

1. Insert a SQL query in the Data Explorer editor that you would like to optimize.
2. Click ✨ Assistant on the Data Explorer (Hue) SQL editor:



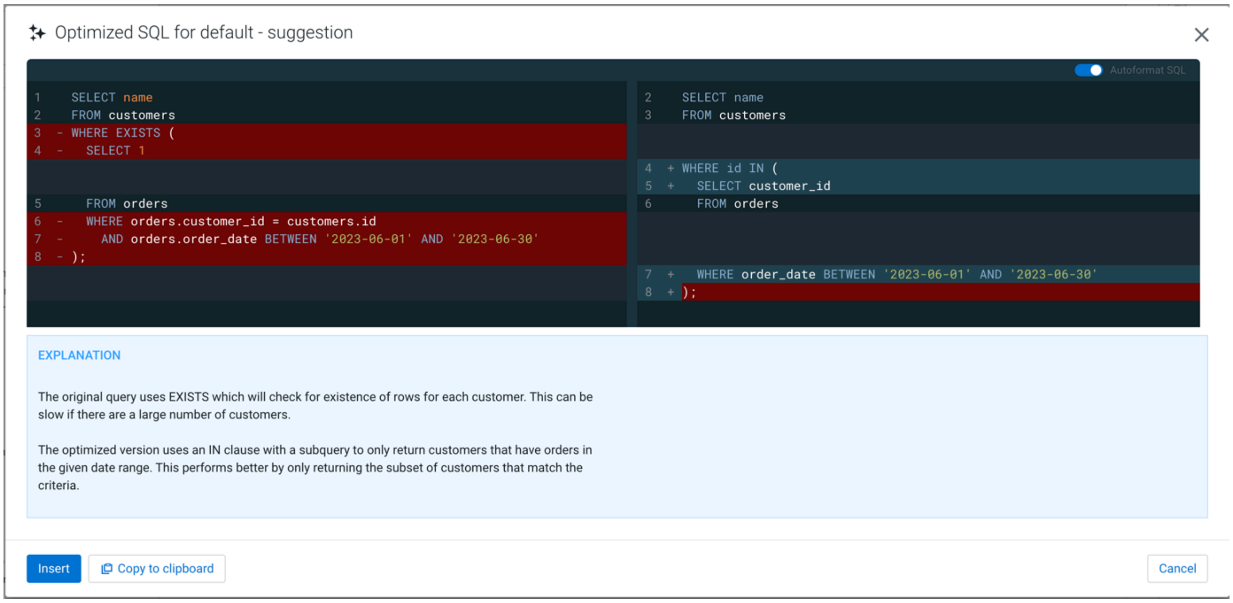
```
Example: SELECT * FROM tablename, or press CTRL + space
```

3. Click OPTIMIZE.



```
SELECT name
FROM customers
WHERE EXISTS (
  SELECT 1
  FROM orders
  WHERE orders.customer_id = customers.id
    AND orders.order_date BETWEEN '2023-06-01' AND '2023-06-30'
);
```

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.



```
   SELECT name
   FROM customers
 - WHERE EXISTS (
 -   SELECT 1
     FROM orders
 -   WHERE orders.customer_id = customers.id
 -     AND orders.order_date BETWEEN '2023-06-01' AND '2023-06-30'
 - );
```

```
   SELECT name
   FROM customers
 + WHERE id IN (
 +   SELECT customer_id
     FROM orders
 +   WHERE order_date BETWEEN '2023-06-01' AND '2023-06-30'
 + );
```

EXPLANATION

The original query uses EXISTS which will check for existence of rows for each customer. This can be slow if there are a large number of customers.

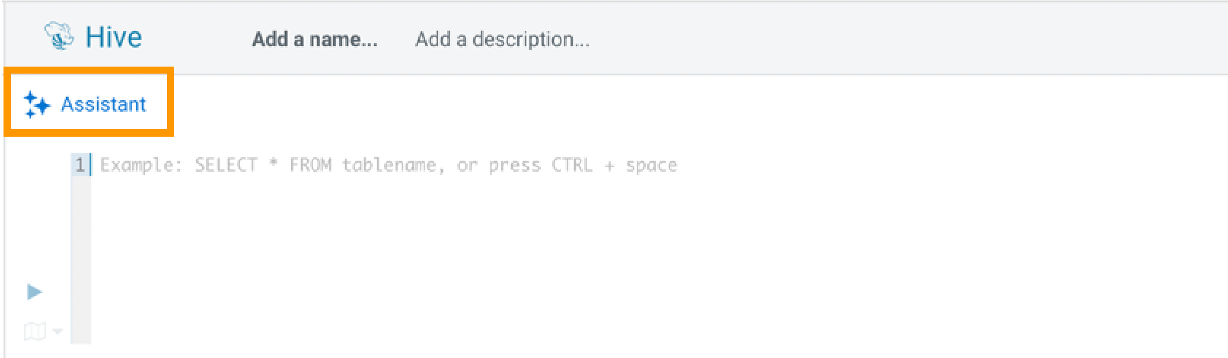
The optimized version uses an IN clause with a subquery to only return customers that have orders in the given date range. This performs better by only returning the subset of customers that match the criteria.

Fixing a query in Cloudera Data Explorer (Hue)

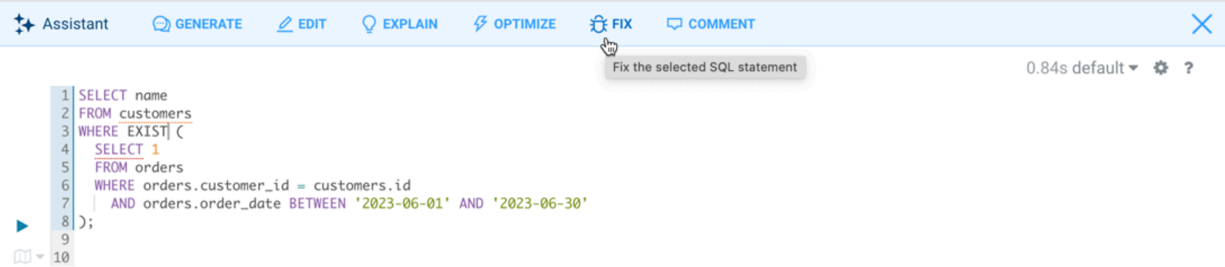
You can use the SQL AI Assistant to fix a broken SQL query. Data Explorer identifies the issues in SQL syntax and provides the corrected version.

Procedure

1. Insert a SQL query in the Data Explorer editor that you would like to fix.
2. Click ✨ Assistant on the Data Explorer (Hue) SQL editor:

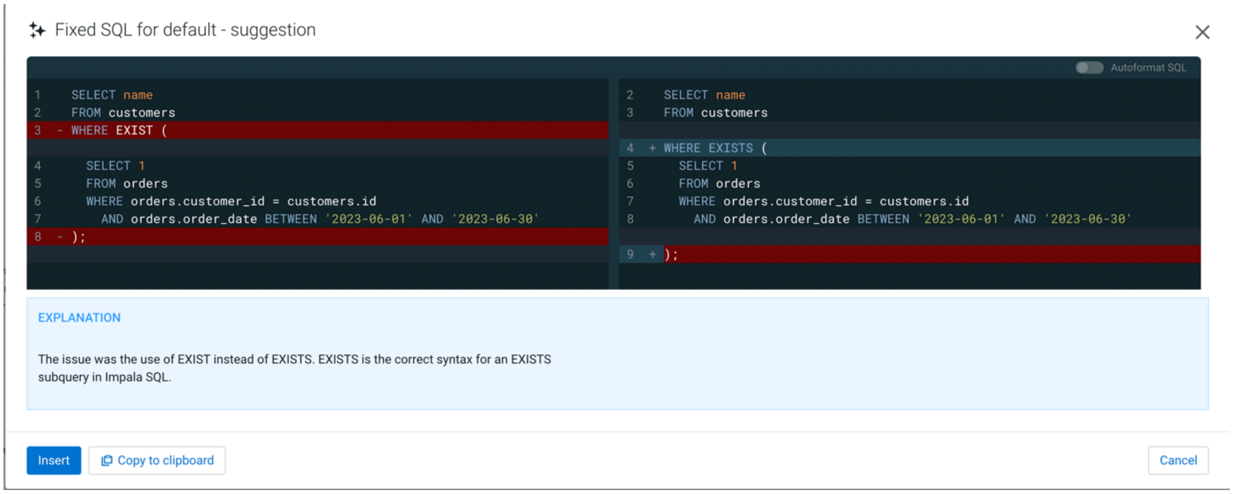


3. Click FIX.



```
SELECT name
FROM customers
WHERE EXIST (
  SELECT 1
  FROM orders
  WHERE orders.customer_id = customers.id
    AND orders.order_date BETWEEN '2023-06-01' AND '2023-06-30'
);
```

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



```
SELECT name
FROM customers
- WHERE EXIST (

  SELECT 1
  FROM orders
  WHERE orders.customer_id = customers.id
    AND orders.order_date BETWEEN '2023-06-01' AND '2023-06-30'
- );
```

```
SELECT name
FROM customers

+ WHERE EXISTS (
  SELECT 1
  FROM orders
  WHERE orders.customer_id = customers.id
    AND orders.order_date BETWEEN '2023-06-01' AND '2023-06-30'

+ );
```

EXPLANATION

The issue was the use of EXIST instead of EXISTS. EXISTS is the correct syntax for an EXISTS subquery in Impala SQL.

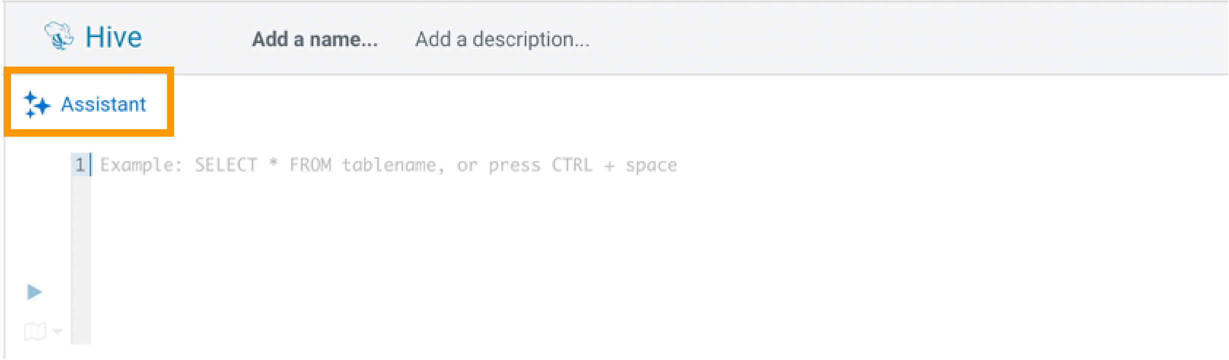
Click Insert to insert the fixed query in the Data Explorer editor and run it.

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

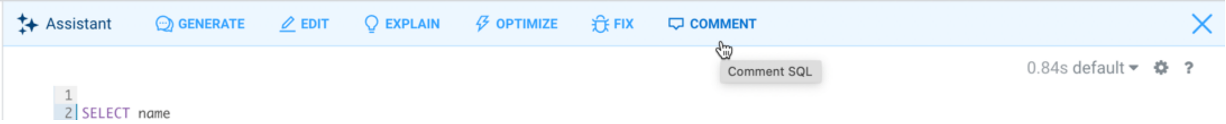
The Data Explorer SQL AI Assistant can generate a comment explaining what SQL query does. You can insert it into the query to improve readability.

Procedure

1. Insert a SQL query in the Data Explorer editor for which you want to generate a comment.
2. Click ✨ Assistant on the Data Explorer (Hue) SQL editor:

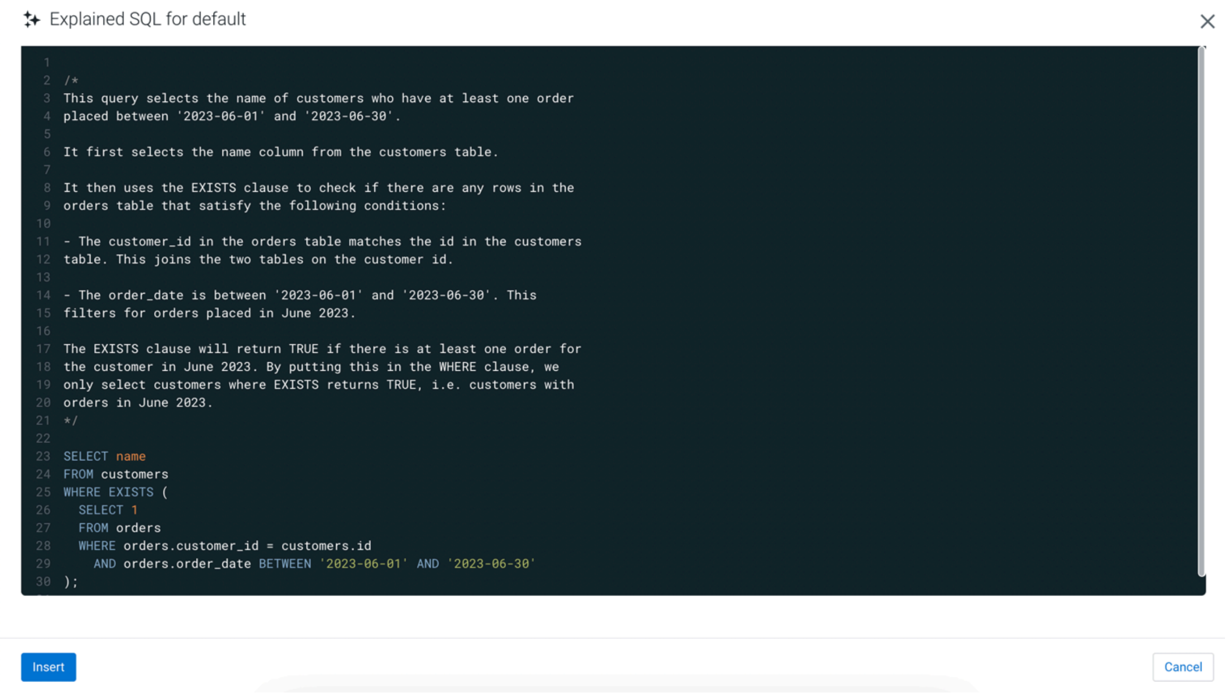


3. Click COMMENT.



```
SELECT name
```

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



```

/*
This query selects the name of customers who have at least one order
placed between '2023-06-01' and '2023-06-30'.

It first selects the name column from the customers table.

It then uses the EXISTS clause to check if there are any rows in the
orders table that satisfy the following conditions:

- The customer_id in the orders table matches the id in the customers
table. This joins the two tables on the customer id.

- The order_date is between '2023-06-01' and '2023-06-30'. This
filters for orders placed in June 2023.

The EXISTS clause will return TRUE if there is at least one order for
the customer in June 2023. By putting this in the WHERE clause, we
only select customers where EXISTS returns TRUE, i.e. customers with
orders in June 2023.
*/

SELECT name
FROM customers
WHERE EXISTS (
  SELECT 1
  FROM orders
  WHERE orders.customer_id = customers.id
    AND orders.order_date BETWEEN '2023-06-01' AND '2023-06-30'
);
```

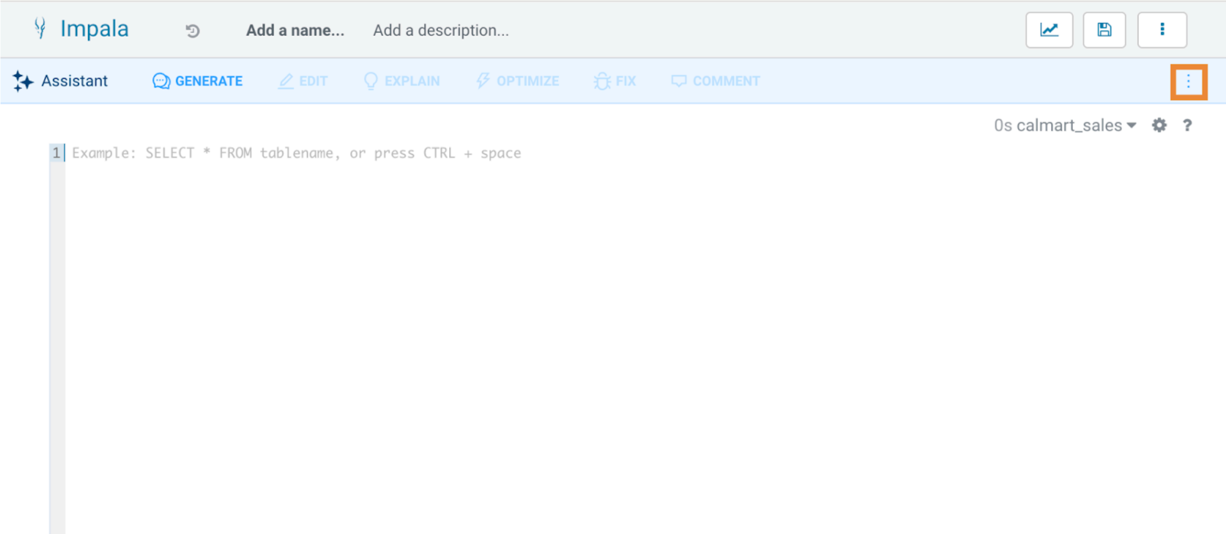
Click Insert to insert the comment into the query.

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.

   Example: SELECT * FROM tablename, or press CTRL + space

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.

Cloudera Data Explorer (Hue) SQL AI Assistant FAQ

A collection of frequently asked questions about Hue SQL AI Assistant.

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
- vLLM

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.

Cloudera Data Explorer (Hue) logs

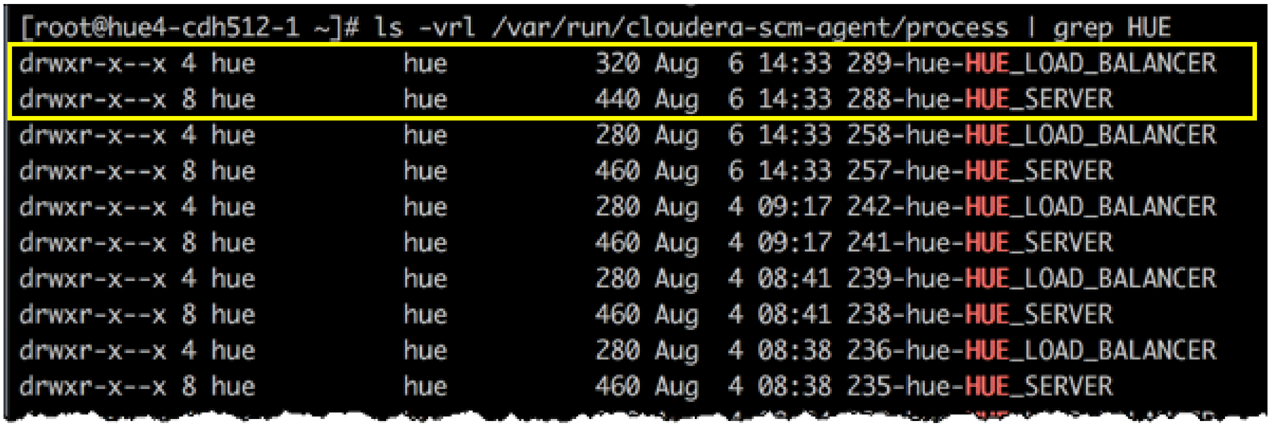
Cloudera Manager generates standard stream logs when each Hue role instance starts and stops. The Hue service, which is built on the Django framework, generates log4j logs when the service is running.

Standard stream logs

Cloudera Manager logs the start and stop of each supervised Hue process in standard stream logs (stdout.log, stderr.log).

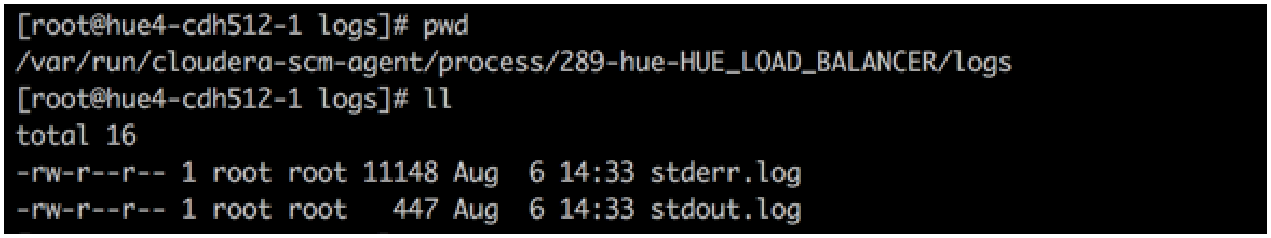
When the Hue service restarts, Cloudera Manager generates a new directory for each supervised process of a Hue role instance:

```
ls -vrl /var/run/cloudera-scm-agent/process | grep HUE
```



```
[root@hue4-cdh512-1 ~]# ls -vrl /var/run/cloudera-scm-agent/process | grep HUE
drwxr-x--x 4 hue          hue          320 Aug  6 14:33 289-hue-HUE_LOAD_BALANCER
drwxr-x--x 8 hue          hue          440 Aug  6 14:33 288-hue-HUE_SERVER
drwxr-x--x 4 hue          hue          280 Aug  6 14:33 258-hue-HUE_LOAD_BALANCER
drwxr-x--x 8 hue          hue          460 Aug  6 14:33 257-hue-HUE_SERVER
drwxr-x--x 4 hue          hue          280 Aug  4 09:17 242-hue-HUE_LOAD_BALANCER
drwxr-x--x 8 hue          hue          460 Aug  4 09:17 241-hue-HUE_SERVER
drwxr-x--x 4 hue          hue          280 Aug  4 08:41 239-hue-HUE_LOAD_BALANCER
drwxr-x--x 8 hue          hue          460 Aug  4 08:41 238-hue-HUE_SERVER
drwxr-x--x 4 hue          hue          280 Aug  4 08:38 236-hue-HUE_LOAD_BALANCER
drwxr-x--x 8 hue          hue          460 Aug  4 08:38 235-hue-HUE_SERVER
```

It writes to a nested logs directory for each running instance:



```
[root@hue4-cdh512-1 logs]# pwd
/var/run/cloudera-scm-agent/process/289-hue-HUE_LOAD_BALANCER/logs
[root@hue4-cdh512-1 logs]# ll
total 16
-rw-r--r-- 1 root root 11148 Aug  6 14:33 stderr.log
-rw-r--r-- 1 root root   447 Aug  6 14:33 stdout.log
```

Configuration errors are written here because they prevent Hue servers and load balancers from starting properly.

Tip: Testing the LDAP configuration from Cloudera Manager (Clusters Hue service Test LDAP Configuration) also writes to standard stream logs which you can search using the following command: ls -vrl /var/run/cloudera-scm-agent/process | grep ldaptest

Cloudera Data Explorer (Hue) service Django logs

When the Hue service is running, Hue generates logs in /var/log/hue using log4j. Load balancer logs are in /var/run/httpd. You can view these logs in Hue at http://hueserver:port/logs.

Table 1: Hue service logs

Log Name	Description
access.log	Filtered list of successful attempts to access Hue Web UI.

Log Name	Description
audit.log	Captures user operations, such as: • USER_LOGIN • USER_LOGOUT • EXPORT • DOWNLOAD • DELETE_USER • DELETE_GROUP • EDIT_USER • CREATE_USER • EDIT_GROUP • CREATE_GROUP • EDIT_PERMISSION • ADD_LDAP_USERS • ADD_LDAP_GROUPS • SYNC_LDAP_USERS_GROUPS
audit/hue_server_audit_wal.log	Audit log visible in Apache Atlas.
error.log	Filtered list of all nontrivial errors
kt_renewer.log	Kerberos ticket renews
metrics-hue_server/metrics.log	Populates charts in Cloudera Manager
migrate.log	Database and table migrations + First Run of Hue server
rungunicornserver.log	Hue (Gunicorn) web server information.
hue_install.log	Contains the log produced during installation

Enabling DEBUG logging for Cloudera Data Explorer (Hue) logs

DEBUG is available for the Hue server logs in /var/log/hue. By default, the Hue service writes INFO level messages and keeps a small buffer of log messages at all levels in memory. If you want detailed, debug-level information, then you can enable this feature from Hue configuration in Cloudera Manager.

About this task

In the debug mode, Hue displays a detailed traceback when an exception occurs. Debugging information may contain sensitive data.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration and select the Enable Django Debug Mode option.

   The Enable Django Debug Mode option is used to toggle between INFO and DEBUG-level logging.
3. Restart the Hue service.

Enabling httpd log rotation for Cloudera Data Explorer (Hue)

Several log files, such as access logs and error logs, are produced by the Apache HTTP server. Over time, these files can grow in size and can cause disk issues. You can prevent this problem by using Apache's rotatelogs utility and defining when to rotate logs – either based on a time interval or maximum size of the log.

About this task

You can configure the Apache server to rotate the logs when a certain filesize is reached (for example, 5 Bytes or 10 KBytes) or at a set time interval (for example, daily or weekly), or both. If you want to specify both the time and size, the size must be specified after the time. Log rotation occurs whenever time or size limits are reached.

The log files are generated as per the configurations defined in the httpd.conf file.

Before you begin

The rotatelogs utility must be present on your Operating System, and you must know its location so that you can use it. Run the following command from the Hue load balancer host system to get the path of the rotatelogs utility:

```
[root@example-domain-hue-HUE_LOAD_BALANCER]# which rotatelogs
/usr/sbin/rotatelogs
```

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration .
3. Configure the log rotation policy based on your criteria in the Load Balancer Advanced Configuration Snippet (Safety Valve) for httpd.conf field.

   To rotate logs whenever the specific file size is reached, add the following lines:

   ```
   CustomLog "|/usr/sbin/rotatelogs /var/log/hue-httpd/access_log.%Y-%m-%d-
   %H_%M_%S [***FILE-SIZE***](B|K|M|G)" common
   ErrorLog "|/usr/sbin/rotatelogs /var/log/hue-httpd/error_log.%Y-%m-%d-%H_
   %M_%S [***FILE-SIZE***](B|K|M|G)"
   ```

   For example:

   ```
   CustomLog "|/usr/sbin/rotatelogs /var/log/hue-httpd/access_log.%Y-%m-%d-
   %H_%M_%S 5B" common
   ErrorLog "|/usr/sbin/rotatelogs /var/log/hue-httpd/error_log.%Y-%m-%d-%H_
   %M_%S 5B"
   ```

   This configuration rotates the log file whenever it reaches a size of 5 bytes, and the suffix to the log file name will be created with the format access_log. YYYY-mm-dd-HH_MM_SS and error_log.YYYY-mm-dd-HH_MM_SS.

   To rotate logs daily, add the following lines:

   ```
   CustomLog "|/usr/sbin/rotatelogs -l /var/log/hue-httpd/access_log.%Y-%m-
   %d-%H_%M_%S 86400" common
   ErrorLog "|/usr/sbin/rotatelogs -l /var/log/hue-httpd/error_log.%Y-%m-%d-
   %H_%M_%S  86400"
   ```

   This creates the files /var/log/access_log.yyyy.mm.dd and /var/log/error_log.yyyy.mm.dd where yyyy is the year, mm is the month, and dd is the day of the month. The log operation switches to a new file every day at midnight, local time.

   For more information, see *rotatelogs - Piped logging program to rotate Apache logs*.
4. Click Save Changes.
5. Restart the affected services.

Related Information

rotatelogs - Piped logging program to rotate Apache logs

Cloudera Data Explorer (Hue) supported browsers

Data Explorer works with the two most recent LTS (long term support) or ESR (extended support release) browsers. Cookies and JavaScript must be enabled.

The lists the minimum tested versions of the most common browsers:

- Chrome: (Version history)
- Firefox: (Version history)
- Safari (Mac only): Version history
- Microsoft Edge: (Version history)

Data Explorer can display in other browsers and in older versions of the common browsers, but you might not have access to all features.

Enabling cache-control HTTP headers when using Cloudera Data Explorer (Hue)

You can enable Data Explorer to use HTTP headers such as Cache-Control, Pragma, and Expires to ensure that your browser always uses the fresh version of the resource. You must set the value of the custom_cache_control property to true in Hue's Advanced Configuration Snippet.

About this task

Note: Enabling the use of these cache-control HTTP headers can affect performance because your browser tries to fetch the latest resources instead of using cached data. To disable cache-control HTTP headers, set the value of the custom_cache_control property to false.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration and add the following line in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [desktop]
   custom_cache_control=true
   ```
3. Click Save Changes.
4. Restart the Hue service.

Adding a Cloudera Data Explorer (Hue) service with Cloudera Manager

You can use the Add a Service wizard in Cloudera Manager to add and configure a new Hue service instance.

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Go to Clusters Actions and click Add Service.

   The **Add Service to Cluster** wizard is displayed.

3. Select Hue as the Service Type and click Continue.

   The **Select Dependencies** page is displayed.
4. On the **Select Dependencies** page, select the optional dependencies that are required for your cluster and click Continue.

   The required dependency is selected by default.

   Note:

   In CDH 6 and earlier, the Hive service included the Hive Metastore and HiveServer2. In Cloudera Runtime 7.0 and later, this service only includes the Hive Metastore. HiveServer2 and other components of the Hive execution engines are part of the HIVE_ON_TEZ service.

   If you need to run Hive queries, then select an optional dependency containing the HIVE_ON_TEZ service.

   If you need to run only Impala queries, then the HIVE_ON_TEZ service is not required.
5. On the **Assign Roles** page, you can perform one of the following actions:

   - To accept the default role assignments, click Continue in the lower right corner of the page. To view the default role assignments by host, click View By Host.
   - To customize role assignments, click the field below Hue Server and Load Balancer roles. This launches a dialog box where you can select hosts where you want to add the role. Click OK after making your selection.

     The wizard evaluates host hardware configurations to determine the best hosts for each role. All worker roles are automatically assigned to the same set of hosts as the HDFS DataNode. You can reassign if necessary. Specify hostnames by IP address, rack name, or by range:

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

6. On the **Setup Database** page:

   a) Select a database vendor from the Type field and specify the Database Hostname, Database Name, Username, and Password.

   b) Click Test Connection, and when the success message appears, click Continue and Cloudera Manager starts the Hue service.
7. Review your configurations on the **Review Changes** page and click Continue.
8. On the **Command Details** page, expand the running commands to view the details of any step, including log files and command output. You can filter the view by selecting Show All Steps, Show Only Failed Steps, or Show Only Running Steps.

   After the First Run command completes, click Continue to go to the **Summary** page.
9. After the service is started, click Continue and then click Finish.

   If your cluster uses Kerberos, Cloudera Manager automatically adds a Hue Kerberos Ticket Renewer role to each host where you assigned the Hue Server role instance. See "Enabling Kerberos Authentication" for more information.

Adding a Cloudera Data Explorer (Hue) role instance with Cloudera Manager

Roles are functions that comprise a service. Role instances must be assigned to one or more hosts. You can assign roles to hosts by using Cloudera Manager.

Procedure

1. In the Cloudera Manager Home page, select the Hue service.
2. On the Hue service page, click the Instances tab.
3. On the Instances tab page, click Add Role Instances on the right side of the page.
4. On the Assign Roles page:

   a) Click Select hosts under the role names to select hosts where you want to add the role. This launches a dialog box where you can select hosts where you want to add the role.

   The wizard evaluates host hardware configurations to determine the best hosts for each role. All worker roles are automatically assigned to the same set of hosts as the HDFS DataNode. You can reassign if necessary. Specify hostnames by IP address, rack name, or by range:

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

   b) Click OK to save your hostname selection for a role.

   c) When you are finished assigning roles to hosts, click Continue in the lower right corner of the page.
5. Cloudera Manager automatically adds the roles and returns you to the Hue Instances tab.

   Note: If your cluster uses Kerberos, Cloudera Manager automatically adds a Hue Kerberos Ticket Renewer role to each host where you assigned the Hue Server role instance. Cloudera Manager throws a validation error if the new Hue Server role does not have a co-located Kerberos Ticket Renewer role. See "Enabling Kerberos for Authentication" for more information.

Setting up a Cloudera Data Explorer (Hue) service account with a custom name

In Cloudera, “hue” username is the default machine user and group name for Hue. During installation and running, Hue requires the "hue" user to exist and be part of the "hue" group. You can create a custom machine user and group name on the Hue cluster and then specify them in the Hue configurations inCloudera Manager.

Procedure

1. SSH into the Hue server host as a user with sudo privileges
2. Create a new group for your custom Hue user account by running the following command:

   ```
   sudo groupadd[***NEW-HUE-GROUP-NAME***]
   ```

   Replace *[***NEW-HUE-GROUP-NAME***]* with the custom group name you want to use.
3. Create a new user and assign it to the newly created group by running the following command:

   ```
   sudo useradd -G [***NEW-HUE-GROUP-NAME***] -m [***NEW-HUE-USER-NAME***] -
   s /usr/sbin/nologin
   ```

   The new user is stored in the /etc/passwd file, and group is stored in the /etc/group file.
4. Verify that the new user exists within the newly created group by running the following command:

   ```
   grep [***NEW-HUE-USER-NAME***] /etc/passwd
   ```

5. Change the ownership of the Hue directories as follows:

   ```
   sudo chown -R [***NEW-HUE-USER-NAME***]:[***NEW-HUE-GROUP-NAME***] /var/log/hue
   sudo chown -R [***NEW-HUE-USER-NAME***]:[***NEW-HUE-GROUP-NAME***] /var/log/hue-httpd
   sudo chown -R [***NEW-HUE-USER-NAME***]:[***NEW-HUE-GROUP-NAME***] /tmp/hue_*
   ```

6. Modify the Hue configuration in Cloudera Manager as follows:
   a) Log in to Cloudera Manager as an Administrator.
   b) Go to Clusters Hue Configuration .
   c) Specify the custom group name you created earlier*[***NEW-HUE-GROUP-NAME***]* in the Default User Group field.
   d) Specify the custom user name you created earlier*[***NEW-HUE-USER-NAME***]* in the System User field.
   e) Specify the custom group name you created earlier*[***NEW-HUE-GROUP-NAME***]* in the System Group field.
   f) Add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field.

      ```
      [desktop]
      server_user=[***NEW-HUE-USER-NAME***]
      server_group=[***NEW-HUE-GROUP-NAME***]
      default_user=[***NEW-HUE-USER-NAME***]
      ```

   g) Click Save Changes.
   h) Restart the Hue service.
7. Tail the /var/log/hue/rungunicornserver.log log file for any issues.
8. Change the permissions of the Hue home directory as follows:

   ```
   sudo cd /opt/cloudera/parcels/CDH/lib/
   sudo chown -R [***NEW-HUE-USER-NAME***]:[***NEW-HUE-GROUP-NAME***]hue
   ```

9. Change the permissions of the process directory as follows:

   ```
   sudo cd /var/run/cloudera-scm-agent/
   sudo setfacl -Rm user:[***NEW-HUE-USER-NAME***]:rwx process
   ```

10. Verify Access Control List (ACL) changes by running the following command:

    ```
    getfacl process
    ```

To validate newly configured custom user and group

Run the following Python script on your Hue hosts to validate that the new user and group names are valid and configured correctly.

This script requires you to input the user (uid) and group (gid) IDs corresponding to the custom user and group names you created and displays the effective user and group IDs that are in use. You must ensure that the user belongs to the same group while specifying the uid and gid. The script displays the “Permission denied: Operation not permitted” error if there is a mismatch between the user ID and the group ID :

```
import os
def change_user_and_group(euid, egid):
    try:
        # Change the effective group ID
        os.setegid(egid)
```

```
        print(f"Effective Group ID changed to: {egid}")
        # Change the effective user ID
        os.seteuid(euid)
        print(f"Effective User ID changed to: {euid}")
        # Verify changes
        print(f"Current Effective User ID: {os.geteuid()}")
        print(f"Current Effective Group ID: {os.getegid()}")
    except PermissionError as e:
        print(f"Permission denied: {e}")
    except Exception as e:
        print(f"An error occurred: {e}")

if __name__ == "__main__":
    # Replace with actual user and group IDs that you want to switch to
    # Change `new_euid` to the user ID (UID) of the custom user you created.
    # Change `new_egid` to the group ID (GID) of the custom group you create
d.
    # Example: new_euid = 1001  # Non-root user ID
    #           new_egid = 1001  # Non-root group ID
    new_euid = 1001  # Replace with the new user ID (UID)
    new_egid = 1001  # Replace with the new group ID (GID)
    change_user_and_group(new_euid, new_egid)
```

Options to restart the Cloudera Data Explorer (Hue) service

Restarting the Hue service is necessary after upgrading Cloudera or making configuration changes to the Hue service. You can restart Hue in regular or rolling restart mode. Rolling restart reduces the service downtime to approximately 80 to 90 seconds.

You can restart the Hue service independently or as a part of the cluster restart. When you restart the Hue service normally, Hue can become unavailable for an average of 30 minutes from the restart time (this could vary depending on the number of services in your cluster). When you restart the cluster containg the Hue service, Hue is stopped first among other services in the cluster and restarted at the end. To reduce the service downtime, Cloudera recommends that you use the “Rolling Restart” option.

Note: Cluster-level rolling restart is supported only in High Availability (HA) clusters. However, you can restart the Hue service and its roles independently in the rolling restart mode in both HA and non-HA clusters.

Restarting the Hue service independently in rolling restart mode

To restart Hue service in the rolling restart mode, use one of the following steps:

- Go to Cloudera Manager Clusters and click ⋮ corresponding to the Hue service and then click Rolling Restart, or
- Go to Cloudera Manager Clusters Hue and click Actions Rolling Restart .

Cloudera Manager shows the following options to restart the Hue service in the rolling restart mode:

Category	Restart options	Description
Restart filters	Restart roles with stale configuration only	Select this option to restart Hue if you have updated the Hue configuration.
	Restart roles with the old software version only	Select this option to restart Hue if you have upgraded the cluster.
Role types to restart	Kerberos Ticket Renewer	Selected by default. Restarts the Kerberos Ticket Renewer role.

Category	Restart options	Description
	Load Balancer	Selected by default. Restarts the Load Balancer role after the Kerberos Ticket Renewer.
	Hue Server	Selected by default. Restarts the Hue Server role after the load balancer.

When you restart Hue, Hue’s non-worker roles, such as the load balancer, Kerberos ticket renewer, and Hue server restart one after the other.

Restarting a CDP cluster containing the Hue service in the rolling restart mode

To restart an HA cluster containing the Hue service in the rolling restart mode, go to Cloudera Manager Clusters and click ⋮ corresponding to your cluster name and select Rolling Restart. You see the **Rolling Restart** modal with the list of services that support rolling restart.

Note: All Hue roles (namely the load balancer, Kerberos ticket renewer, and Hue server) are of the non-worker type. You must select one of the following options under the Roles to include section for ensuring that the Hue service is restarted in the rolling restart mode, depending on your requierments:

- Non-Workers Only, or
- All Roles

Related Information

Rolling Restart

Customizing the Cloudera Data Explorer (Hue) web interface

To customize the Hue web interface, add configuration properties in Cloudera Manager. You can customize the banner, the page logo, the cache timeout setting, and you can enable or disable anonymous usage data collection.

Adding a custom banner in Cloudera Data Explorer (Hue)

You can add a custom banner to the Hue web interface by adding your custom HTML to the Top Banner Custom HTML property in Cloudera Manager.

Procedure

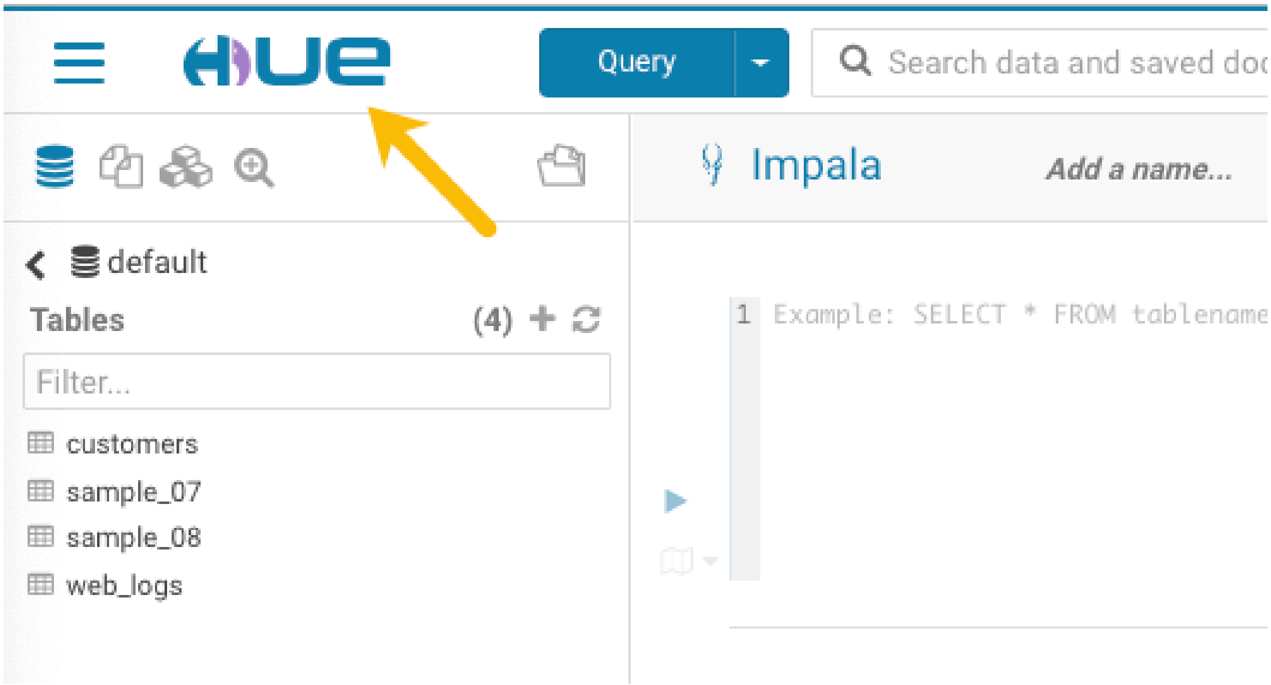
1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and add the banner test in the Top Banner Custom HTML field as follows:
   <H1>Your company's custom Hue Web UI banner</H1>
3. Click Save Changes.
4. Restart the Hue service.

Changing the page logo in Cloudera Data Explorer (Hue)

You can replace the Hue web interface logo with a custom log that is created with an SVG code. Add any type of logo you want, but your custom logo should be designed to fit into a 160 x 40 pixel space.

About this task

For example, here is the Hue logo shown in the following image:

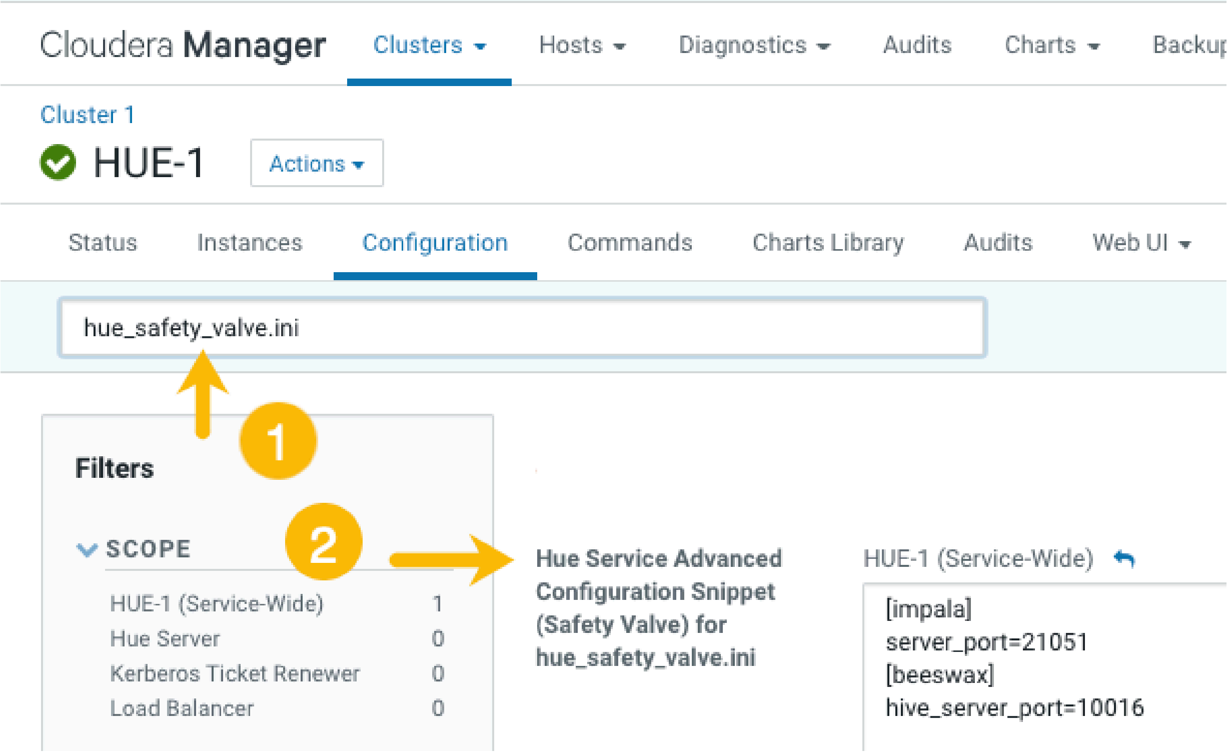


You can change this Hue logo by adding the appropriate SVG code to the logo_svg property under the [desktop] [[custom]] section in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration parameter in Cloudera Manager.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration

3. In the Search text box, type hue_safety_valve.ini to locate the configuration parameter:



hue_safety_valve.ini

Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini

HUE-1 (Service-Wide)

```
[impala]
server_port=21051
[beeswax]
hive_server_port=10016
```

4. Add the following parameters with your custom logo SVG code to the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration parameter:

```
[desktop]
[[custom]]
logo_svg='[***SVG-CODE-FOR-CUSTOM-LOGO***]'
```

For example, the following SVG code replaces the Hue logo with a red heart:

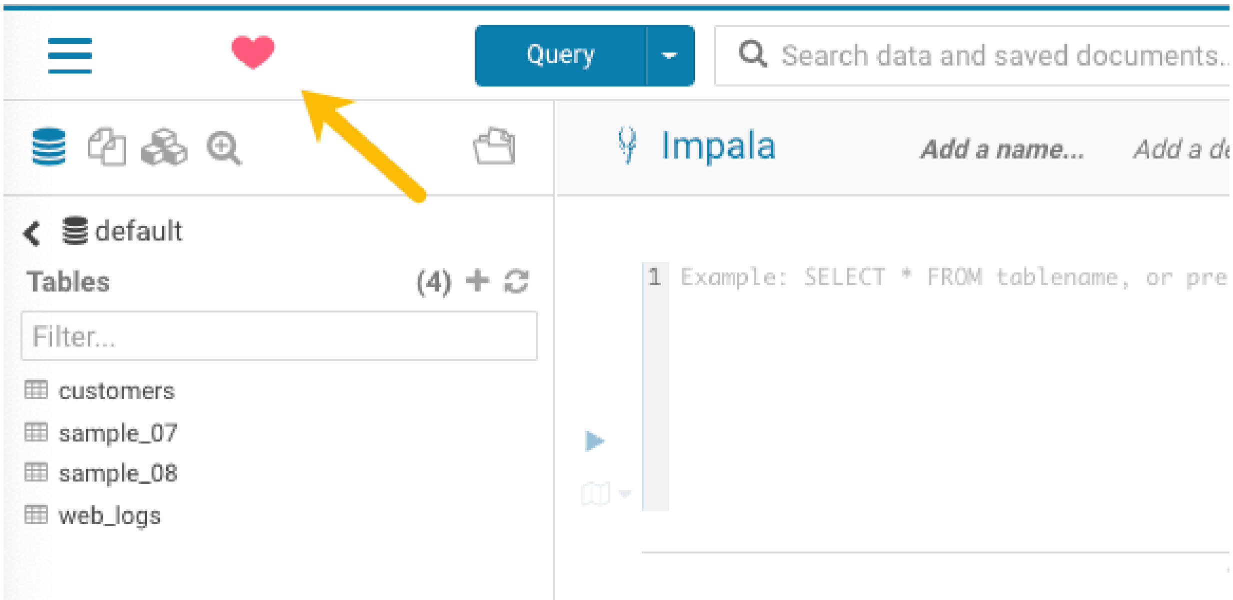
```
[desktop]
[[custom]]
logo_svg='<g><path stroke="null" id="svg_1" d="m44.41215,11.43463c-4.05
017,-10.71473
-17.19753,-5.90773 -18.41353,-0.5567c-1.672,-5.70253 -14.497,-9.95663
-18.411,0.5643c-4.35797,11.71793 16.891,22.23443 18.41163,23.95773c1.518
1,-1.36927 22.7696,-12.43803
18.4129,-23.96533z" fill="#ffffff"/> <path stroke="null" id="svg_2"
d="m98.41246,10.43463c-4.05016,-10.71473 -17.19753,-5.90773 -18.41353,-
0.5567c-1.672,-5.70253
-14.497,-9.95663 -18.411,0.5643c-4.35796,11.71793 16.891,22.23443 18.4116
4,23.95773c1.5181,-1.36927
22.76959,-12.43803 18.41289,-23.96533z" fill="#FF5A79"/> <path stroke="nu
ll" id="svg_3"
d="m154.41215,11.43463c-4.05016,-10.71473 -17.19753,-5.90773 -18.41353,-0
.5567c-1.672,-5.70253
-14.497,-9.95663 -18.411,0.5643c-4.35796,11.71793 16.891,22.23443 18.41164
,23.95773c1.5181,-1.36927 22.76959,-12.43803 18.41289,-23.96533z" fill="
#ffffff"/> </g>'
```

5. Click Save Changes.

6. Restart the Hue service.

7. In the Hue configuration page of Cloudera Manager, select Web UI Hue Load Balanced to load Hue and view your custom logo.

   If you added the sample SVG code that defines a red heart as the logo, your Hue web UI looks like this:

8.
9.
10.

Related Information

Scalable Vector Graphics

Adding a splash screen in Cloudera Data Explorer (Hue)

You can add a custom splash screen to the Hue web interface by adding your custom HTML to the Advanced Configuration Snippet in Cloudera Manager.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and add the banner test in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field as follows:

   ```
   [desktop]
   [[custom]]
   login_splash_html=[***CUSTOM-HTML***]
   ```

   ```
   [desktop]
   [[custom]]
   login_splash_html=<h1>Hue, the next-gen SQL Assistant</h1>
   ```
3. Click Save Changes.
4. Restart the Hue service.

Setting the cache timeout

Enable Hue UI caching by setting a timeout value in milliseconds. The default is 10 days or 864000000 milliseconds. Set the timeout to 0 to disable caching.

About this task

When you browse tables using the left assist panel or run queries, Hue caches this information for fetching information faster and query autocompletion. You can configure the time for which you want to Hue to cache this information by setting the value of the cacheable_ttl property under the [desktop][[custom]] section in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration property in Cloudera Manager.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration .
3. In the Search text box, type hue_safety_valve.ini to locate the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration parameter.
4. Add the following parameters with the cache timeout value to the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini configuration parameter:

   ```
   [desktop]
   [[custom]]
   cacheable_ttl=[***VALUE-IN-MILLISECONDS***]
   ```

   For example, the following configuration sets the cache timeout to the default value of 86400000 milliseconds:

   ```
   [desktop]
   [[custom]]
   cacheable_ttl=86400000
   ```
5. Click Save Changes.
6. Restart the Hue service.

Enabling or disabling anonymous usage date collection

Hue tracks anonymized pages and application versions to gather information about application usage levels. The data collected does not include hostnames or IDs. For example, the data collected has the format /2.3.0/pig or /2.5.0/beeswax/execute.

About this task

To enable or disable anonymous usage data collection:

Procedure

1. In the Cloudera Manager Admin Console, select Clusters Hue Configuration to navigate to the configuration page for Hue.
2. In the Search text box, type usage to locate the Enable Usage Data Collection check box:
   - To enable anonymous data collection, check the box, which is the default setting.
   - To disable anonymous data collection, clear the check box.
3. Enter a Reason for change..., and then click Save Changes at the bottom of the page to save the configuration change.

4. Refresh the browser page and click the restart icon ⏻ at the top of the page so the new configuration changes can be read by the server and the new data collection setting takes effect.

Configuring the number of objects displayed in Cloudera Data Explorer (Hue)

By default, a specific number of databases, tables, and columns are displayed in the Hue web interface. You can increase this number by specifying a value for the MAX_CATALOG_SQL_ENTRIES property in the Hue Advanced Configuration Snippet using Cloudera Manager.

About this task

5000 objects (databases, tables, and columns) are displayed by default on the left assist panel, auto-complete drop-down list, and on the Table Browser page in the Hue web interface. Setting the value of the MAX_CATALOG_SQL_ENTRIES property higher than the default value increases the number of objects but can downgrade performance.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini and add the following lines:

```
[beeswax]
max_catalog_sql_entries=[***NUMBER-OF-OBJECTS***]
```

3. Click Save Changes.
4. Restart the Hue service.

Using Oracle database with Cloudera Data Explorer (Hue)

You can use a customer provided external Oracle database to store Hue metadata, job and query history, and account information. In Cloudera 7.1.x, Hue supports Oracle database versions 12c and 19c.

To use an Oracle database with Hue, you need:

Oracle Instant Client library

The Oracle Instant Client connects Hue to the backend Oracle database. The Oracle Client libraries provide the necessary network connectivity and tools, such as SQL*Plus and Oracle Data Pump, to access the data.

cx_Oracle Python extension module

The cx_Oracle module loads the Oracle Client libraries. The cx_Oracle module and the Oracle Client libraries communicate with the data over Oracle Net.

Hue on releases between 7.1.8 and 7.1.9 uses Python version 3.8. or 3.9, depending on the installed operating system, and it works with the cx_Oracle 8.3.0 distributed with these releases.

For more information, see Cloudera support matrix.

Note: To use Oracle as a backend database for Hue, coordinate with you customer Oracle DBA (Database Administrator) and get from them the appropriate Oracle Instant Client library kits for download and installation. These kits are per the Oracle server DB release (e.g., Oracle 19c). Ensure they are installed on all nodes where the Hue service will be running. Example for matching kits for Oracle 19c is below:

- oracle-instantclient19.25-basic-19.25.0.0.0-1.x86_64.rpm

- oracle-instantclient19.25-devel-19.25.0.0.0-1.x86_64.rpm
- oracle-instantclient19.25-sqlplus-19.25.0.0.0-1.x86_64.rpm.

Before connecting to the Oracle database, install the cx_Oracle Python extension module on the nodes where the Hue service will be running by executing the following command:

```
sudo yum install python3-pip python3-cx-oracle
```

Use the appropriate command for operating systems other than RHEL. As before, ensure this is deployed only on the nodes running the Hue service.

After installing the Cloudera Cluster and the Hue service, configure the relevant database connection parameters in the Hue settings based on the details provided by your Oracle DBA.

Creating Cloudera Data Explorer (Hue) Schema in Oracle database

Hue requires a SQL database to store small amounts of data, including user account information as well as the history of queries and sharing permissions. Ensure that tables in the Hue schema are not compressed.

Before you begin

Due to a known bug in the Oracle database (12c and higher), you cannot alter a table if it is compressed. If you have compressed tables in the Hue schema, then you may see the “ORA-39726: unsupported add/drop column operation on compressed tables” error. Even if you uncompress an existing table, you may not be allowed to alter the columns. To resolve this, follow the below steps:

1. Create a new uncompressed table with the same structure as the compressed table.
2. Copy the data from the compressed table to the new uncompressed table.
3. Rename or delete the compressed table.
4. Rename the uncompressed table with the name of the original compressed table.

Procedure

1. Create the Hue schema, set quotas, and grant select permissions (do not grant all).

   Note: Oracle 12c users must ALTER session set to avoid creating a common user with the prefix, c##.

   ```
   vi create_hue_database.ddl
   ```

   ```
   Save in create_hue_database.ddl
   ## Change huepassword to something more secure
   CONNECT / as sysdba
   ALTER session set "_ORACLE_SCRIPT"=true;
   DROP user hue cascade;
   CREATE user hue identified by huepassword;
   ALTER user hue quota 1000m on users;
   ALTER user hue quota 100m on system;
   GRANT create sequence to hue;
   GRANT create session to hue;
   GRANT create table to hue;
   GRANT create view to hue;
   GRANT create procedure to hue;
   GRANT create trigger to hue;
   GRANT execute on sys.dbms_crypto to hue;
   ```

```
GRANT execute on sys.dbms_lob to hue;
```

```
sqlplus /nolog < create_hue_database.ddl
```

2. Verify that you can connect to Hue by running the following command:

```
sqlplus hue/[***HUE-PASSWORD***]
```

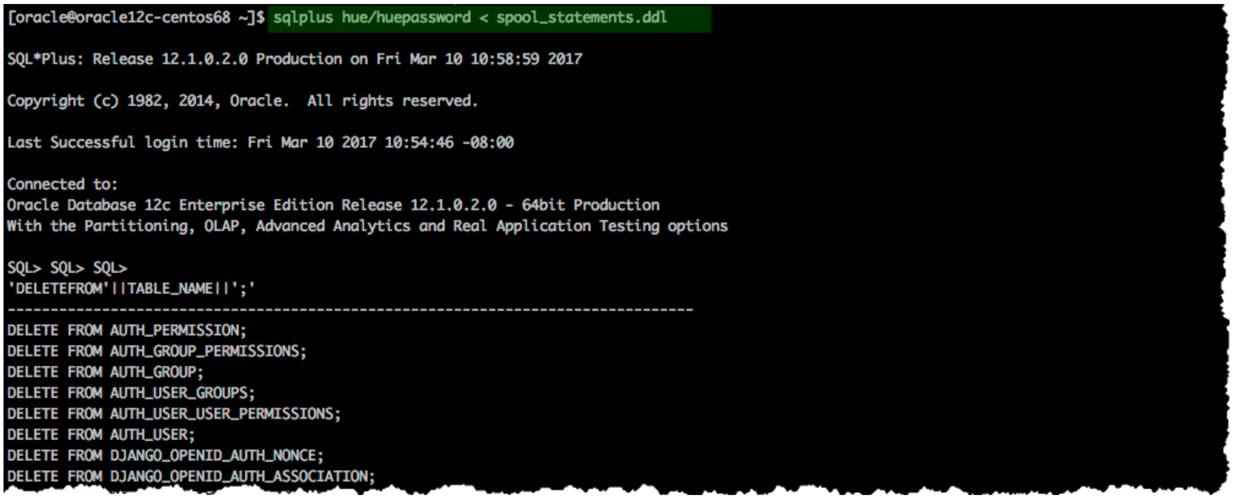
3. Clean all Hue user tables. Create a script to spool delete statements into a new file.

```
delete_from_tables.ddl
vi spool_statements.ddl
```

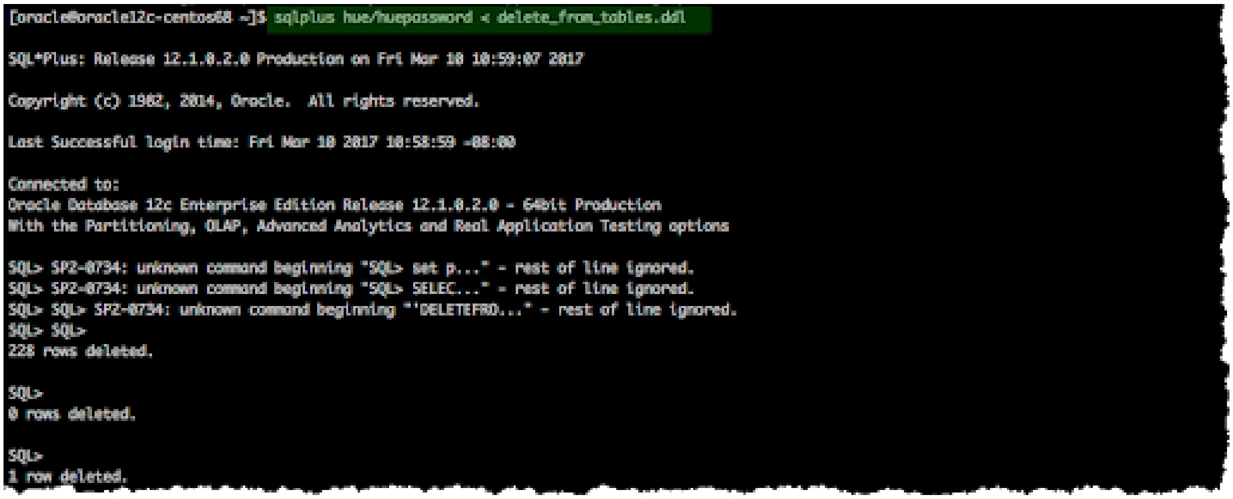
```
## Save in spool_statements.ddl (which generates delete_from_tables.ddl)
spool delete_from_tables.ddl
set pagesize 100;
SELECT 'DELETE FROM ' || table_name || ';' FROM user_tables;
commit;
spool off
quit
```

```
## Create delete_from_tables.ddl
sqlplus hue/[***HUE-PASSWORD***] < spool_statements.ddl

## Run delete_from_tables.ddl
```

```
sqlplus hue/[***HUE-PASSWORD***] < delete_from_tables.ddl
```



```
[oracle@oracle12c-centos68 ~]$ sqlplus hue/huepassword < spool_statements.ddl

SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 10 10:58:59 2017

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Last Successful login time: Fri Mar 10 2017 10:54:46 -08:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> SQL> SQL>
'DELETEFROM'||TABLE_NAME||';'
--------------------------------------------------------------------------------
DELETE FROM AUTH_PERMISSION;
DELETE FROM AUTH_GROUP_PERMISSIONS;
DELETE FROM AUTH_GROUP;
DELETE FROM AUTH_USER_GROUPS;
DELETE FROM AUTH_USER_USER_PERMISSIONS;
DELETE FROM AUTH_USER;
DELETE FROM DJANGO_OPENID_AUTH_NONCE;
DELETE FROM DJANGO_OPENID_AUTH_ASSOCIATION;
```



```
[oracle@oracle12c-centos68 ~]$ sqlplus hue/huepassword < delete_from_tables.ddl

SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 10 10:59:07 2017

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Last Successful login time: Fri Mar 10 2017 10:58:59 -08:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> SP2-0734: unknown command beginning "SQL> set p..." - rest of line ignored.
SQL> SP2-0734: unknown command beginning "SQL> SELEC..." - rest of line ignored.
SQL> SQL> SP2-0734: unknown command beginning "'DELETEFRO..." - rest of line ignored.
SQL> SQL>
228 rows deleted.

SQL>
0 rows deleted.

SQL>
1 row deleted.
```

Configuring Oracle as backend database for Cloudera Data Explorer (Hue)

You can configure Oracle as the backend database for Hue after you have downloaded, distributed, and activated the Oracle Instant Client parcel either using Oracle Service Name or SID.

Before you begin

You must have Hue and Cloudera services installed.

About this task

Use the instructions in this topic to connect Hue to a regular Oracle database. To connect Hue to a TCPS-enable Oracle database, see Set up and configure TLS 1.2 for Hue.

Procedure

1. Log into Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Category Database .

   All the database configurations for Hue are displayed.
3. Select Oracle as the Hue Database Type.
4. Specify the host on which you have installed the Oracle database in the Hue Database Hostname field.

5. Enter 1521 in the Hue Database Port field.
6. Enter user name and password in Hue Database Username and Hue Database Password fields.
7. Specify the database name in the Hue Database Name field.

   If you are using Oracle Service Name instead of SID, then specify the database name in the following format:

   ```
   [***HUE-DB-HOST***]:1521/[***SERVICE-NAME***]
   ```

   You can also configure additional parameters by entering the following in the Hue Database Name field:

   ```
   (DESCRIPTION=(LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=5)(TRANSPOR
   T_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS=(PROTOCOL=TCP)(HOST=[***HUE-
   DB-HOST***])(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=[***SERVICE-NAME***]
   )))
   ```
8. Go to Clusters Hue service Configuration Hue Service Environment Advanced Configuration Snippet (Safety Valve) and add the following parameters:

   Key: LD_LIBRARY_PATH

   Value: /opt/cloudera/parcels/ORACLE_INSTANT_CLIENT/[***INSTANT-CLIENT-WITH-VERSION***]

   Key: ORACLE_HOME

   Value: /opt/cloudera/parcels/ORACLE_INSTANT_CLIENT/[***INSTANT-CLIENT-WITH-VERSION***]

   For example: /opt/cloudera/parcels/ORACLE_INSTANT_CLIENT/instantclient_12_2

   You can obtain the value of the Instant Client version by running the following command from the Cloudera Manager server terminal /opt/cloudera/parcels/ directory:

   ```
   ls -al ORACLE_INSTANT_CLIENT/
   ```

   If you have manually set the values for LD_LIBRARY_PATH and ORACLE_HOME, then specify those.
9. Add support for a multi-threaded environment by adding the following lines in the Hue Service Environment Advanced Configuration Snippet (Safety Valve) field under the [desktop] section:

   ```
   [desktop]
   [[database]]
   options={"threaded":true}
   ```

   A multi-threaded environment is recommended for using the Oracle database.

   If you are using Oracle Service Name instead of SID, then add the following line under the [desktop] section in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [desktop]
   [[database]]
   port=0
   ```
10. Click Save Changes and restart the Hue service.

Results

If the database setup is successful and Hue starts without errors, then you should be able to log into the Hue web interface and view the database/tables.

Configuring high availability support for Oracle RAC database

For seamless connectivity and high availability, you can use Oracle Real Application Clusters (RAC) database as Hue’s backend database.

Before you begin

The correct connection string should come from your Oracle DBA and can be implemented only after the Hue service was installed on a working Cloudera Base Cluster.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration Category Database .

   The database configurations are displayed.
3. Specify the database name in the following format in the Hue Database Name field:

```
(DESCRIPTION=(LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=5)(TRANSPOR
T_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS=(PROTOCOL=TCP)(HOST=[***HOST
NAME-OR-SCAN-IP***])(PORT=[***PORT***]))(CONNECT_DATA=(SERVICE_NAME=[***
SERVICE-NAME***])))
```

4. Click Save Changes.
5. Restart the Hue service.

Using MySQL database with Cloudera Data Explorer (Hue)

Hue is packaged with a lightweight embedded PostgreSQL database for proof-of-concept deployments with one Hue server. However, you can install and use a custom MySQL database as the backend database to store Hue metadata, job and query history, and account information.

Note: MySQL 5.7 has reached its end of life (EOL) and is now covered under Oracle Lifetime Sustaining Support. Per Oracle's Lifetime Support policy, as of October 25, 2023, MySQL 5.7 is covered under Oracle Sustaining Support. You are encouraged to upgrade to MySQL 8.0.

Note: On CDP Private Cloud Base 7.1.9 SP1, Python 3.10 is not mandatory for using Hue with MySQL as its backend database.

Related Information

MySQL documentation

Downloading and installing MySQL database

Download and install MySQL database depending on your Cloudera and Operating System (OS) version.

About this task

Note: MySQL 5.7 has reached its end of life (EOL) and is now covered under Oracle Lifetime Sustaining Support. Per Oracle's Lifetime Support policy, as of October 25, 2023, MySQL 5.7 is covered under Oracle Sustaining Support. You are encouraged to upgrade to MySQL 8.0.

Procedure

1. Download the MySQL distribution from https://dev.mysql.com/downloads/ based on your OS version.

   Use the Cloudera Support Matrix for finding the compatibile versions.

2. Install the database using the following commands depending on your operating system:

   On RHEL/CentOS:

   ```
   sudo yum install mysql-server
   ```

   On SLES:

   ```
   sudo zypper install mysql
   ```

   On Ubuntu:

   ```
   sudo apt-get install mysql-server    #set root psswd when prompted
   ```

   On Debian:

   ```
   sudo apt-get install mysql-server    #set root psswd when prompted
   ```

3. Start the database server using the following commands depending on your operating system:

   On RHEL/CentOS:

   ```
   sudo service mysqld start
   ```

   On SLES:

   ```
   sudo rcmysql start
   ```

   On Ubuntu:

   ```
   sudo service mysql start
   ```

   On Debian:

   ```
   sudo service mysql start
   ```

4. Secure your installation by running the following command:

   ```
   sudo /usr/bin/mysql_secure_installation
   ```

   mysql_secure_installation is a shell script available on Unix systems, and enables you to improve the security of your MySQL installation in the following ways:

   ```
   Enter current password for root (enter for none): [If unset, press Enter
   .]
   OK, successfully used password, moving on...
   [...]
   Set root password? [Y/n] Y [Enter n if password is set.]
   New password:
   Re-enter new password:
   Remove anonymous users? [Y/n] Y
   [...]
   Disallow root login remotely? [Y/n] N
   [...]
   Remove test database and access to it [Y/n] Y
   [...]
   Reload privilege tables now? [Y/n] Y
   ```

Configuring MySQL server

The MySQL server, mysqld, has many command options and system variables that can be set at startup to configure its operation. The configurations are stored in the my.cnf file.

Procedure

1. SSH into the database server host as a root user.
2. Open the my.cnf file for editing in an editor such as vi or nano:

   On RHEL/CentOS/SLES:

   ```
   vi /etc/my.cnf
   ```

   On Debian/Ubuntu:

   ```
   vi /etc/mysql/my.cnf
   ```
3. Ensure that the bind-address property is set to 0.0.0.0 or is commented out in case of a default value.

   ```
   bind-address=0.0.0.0
   ```
4. Ensure that the default-storage-engine property is set to innodb, which is the default storage engine in MySQL versions 5.5 and higher.

   ```
   default-storage-engine=innodb
   ```
5. Ensure that the sql_mode property is set to STRICT_ALL_TABLES to avoid columns being truncated during migration.

   ```
   sql_mode=STRICT_ALL_TABLES
   ```
6. Save the file and exit.
7. Restart the database server.
8. Enable the server to automatically start on boot:

   On RHEL/CentOS:

   ```
   sudo chkconfig mysqld on
   ```

   On SLES:

   ```
   sudo chkconfig mysql on
   sudo rcmysql status
   ```

   On Ubuntu, MySQL starts automatically after installation. Run the following command to check the status:

   ```
   sudo service mysql status
   ```

   On Debian, MySQL starts automatically after installation. Run the following command to check the status:

   ```
   sudo service mysql status
   ```

Installing and configuring MySQL on RHEL 8

The steps for installing and configuring MySQL version 8.0.25 on RHEL 8 on a Cloudera cluster host are different. Make sure that you copy the corresponding my__config_8h_source.html content to the my_config.h file corresponding to your MySQL server version.

About this task

Note: MySQL 5.7 has reached its end of life (EOL) and is now covered under Oracle Lifetime Sustaining Support. Per Oracle's Lifetime Support policy, as of October 25, 2023, MySQL 5.7 is covered under Oracle Sustaining Support. You are encouraged to upgrade to MySQL 8.0.

Procedure

1. SSH into the Hue server host.
2. Install the driver for connecting to the MySQL database server:

```
yum install perl-DBD-MySQL
```

3. Configure the RHEL 8 OS repository (/etc/yum.repos.d/mysql8.repo) to point to http://repo.mysql.com/yum/mysql-8.0-community/el/8/x86_64/.
4. Install MySQL community server:

```
yum install mysql-community-server-8.0.25 mysql-community-common-8.0.25
mysql-community-devel-8.0.25
```

5. Copy the my__config_8h_source.html configuration file content to /usr/include/mysql/my_config.h.
6. Install the MySQL Python interface:

```
pip install MySQL-python --force-reinstall --ignore-installed
```

7. Grant read and execute permissions to all users and write access to the admin user:

```
chmod -R 755 /usr/lib64/python2.7/site-packages/MySQLdb /usr/lib64/pytho
n2.7/site-packages/MySQL_python* /usr/lib64/python2.7/site-packages/_mys
ql*
```

What to do next

After installing the MySQL server, create the Hue database and configure MySQL as the backend database for Hue.

Installing MySQL client for MySQL databases

To use MySQL as a backend database for Hue, you must install the MySQL client and other required dependencies on all the Hue hosts based on your operating system.

Attention: The version 2.2.0 of the MySQL client requires that you set the values of the MYSQLCLIENT_CFLAGS and MYSQLCLIENT_LDFLAGS environment variables, as follows:

```
$ export MYSQLCLIENT_CFLAGS=`pkg-config mysqlclient --cflags`
$ export MYSQLCLIENT_LDFLAGS=`pkg-config mysqlclient --libs`
```

The version 2.2.0 of the MySQL client also requires you to install the Python 3 and MySQL development headers and libraries, as follows on Debian or Ubuntu operating systems:

```
sudo apt-get install python3-dev default-libmysqlclient-dev build-essential
```

or as follows on CentOS and RHEL operating systems:

```
sudo yum install python3-devel mysql-devel
```

Alternatively, you can install and use the version 2.1.1 of the MySQL client, as follows, which does not have this requirement:

```
pip3 install mysqlclient==2.1.1
```

The Hue service user and the root user must have the necessary permissions to access libraries installed in the Python site-packages directory.

For Cent OS

1. SSH into the Hue host as a root user.
2. Download the MySQL yum repository as follows:

   ```
   curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el7-5.noarch.rpm
   ```
3. Install the package as follows:

   ```
   rpm -ivh mysql80-community-release-el7-5.noarch.rpm
   ```
4. Install the required dependencies as follows:

   ```
   yum install mysql-devel
   yum install -y xmlsec1  xmlsec1-openssl
   ```

   For MySQL version 8.0.27, add the mysql-community-client-8.0.25 client package as follows:

   ```
   yum install mysql-community-client-8.0.25
   ```
5. Add the path where you installed the MySQL client and packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```
6. Install the MySQL client as follows:

   ```
   pip3.8 install mysqlclient
   ```

For RHEL

1. SSH into the Hue host as a root user.

2. Download the MySQL yum repository as follows:

   (RHEL 7)

   ```
   curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el7-5.noarch.rpm
   ```

   (RHEL 8)

   ```
   curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el8-8.noarch.rpm
   ```

   (RHEL 9)

   ```
   curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el9-4.noarch.rpm
   ```
3. Install the package as follows:

   (RHEL 7)

   ```
   rpm -ivh mysql80-community-release-el7-5.noarch.rpm
   ```

   (RHEL 8)

   ```
   rpm -ivh mysql80-community-release-el8-8.noarch.rpm
   ```

   (RHEL 9)

   ```
   rpm -ivh mysql80-community-release-el9-4.noarch.rpm
   ```
4. Install the required dependencies as follows:

   ```
   yum install mysql-devel
   yum install -y xmlsec1  xmlsec1-openssl
   ```
5. Add the path where you installed the MySQL client and packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```
6. Install the MySQL client as follows:

   (RHEL 8)

   ```
   pip3.8 install mysqlclient
   ```

   (RHEL 9)

   ```
   pip3.9 install mysqlclient
   ```

For SLES

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

   ```
   zypper install libmysqlclient-devel
   zypper install xmlsec1
   zypper install xmlsec1-devel
   zypper install xmlsec1-openssl-devel
   ```

3. Add the path where you installed the packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```

4. Install the MySQL client as follows:

   ```
   pip3.8 install mysqlclient
   ```

For Ubuntu

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

   ```
   apt-get install libmysqlclient-dev
   apt-get install -y xmlsec1
   apt-get install libxmlsec1-openssl
   ```

3. Add the path where you installed the packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```

4. Install the MySQL client as follows:

   ```
   pip3.8 install mysqlclient
   ```

Creating the Cloudera Data Explorer (Hue) database

Hue requires a SQL database to store small amounts of data, including user account information as well as the history of queries and sharing permissions. You must create a database for Hue with UTF-8 character set and grant user privileges.

Procedure

1. SSH into the database server host as the root user.
2. Create a database for Hue with UTF-8 collation and grant user privileges:

   ```
   CREATE DATABASE [***HUE-DATABASE-NAME***] DEFAULT CHARACTER SET utf8 DEF
   AULT COLLATE utf8_general_ci;
   GRANT ALL ON [***HUE-DATABASE-NAME***].* TO '[***HUE-DATABASE-NAME***]'@'%
   ' IDENTIFIED BY '[***PASSWORD***]';
   ```

   Replace [***HUE-DATABASE-NAME***] with the actual Hue database name such as "hue".

   Ensure Hue uses UTF-8 collation and character set.

   ```
   # To create (use utf8_general_ci or utf8mb4_general_ci):
   CREATE DATABASE [***HUE-DATABASE-NAME***] COLLATE = 'utf8_general_ci';
   # To view default_character_set_name and default_collation_name
   SELECT * FROM INFORMATION_SCHEMA.SCHEMATA;
   # To alter if not created with UTF8 collation
   ALTER DATABASE [***HUE-DATABASE-NAME***] COLLATE = 'utf8_general_ci';
   ```

3. Verify the connection to the Hue database by running the following command:

   ```
   mysql -u [***HUE-DATABASE-NAME***] -p
   Enter password: [***PASSWORD***]
   ```

Configuring MySQL as the backend database for Cloudera Data Explorer (Hue)

After installing and configuring the MySQL database, you can configure Hue to use it as its backend database.

Procedure

1. Log into Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and filter by Database category.
3. Select the Hue Database Type as MySQL.
4. Set the following database parameters:

   Hue Database Hostname: FQDN of host running MySQL server

   Hue Database Port: 3306

   Hue Database Username: [***USERNAME***]

   Hue Database Password: [***PASSWORD***]

   Hue Database Name: [***HUE-DATABASE-NAME***] or SID
5. Click Save Changes.
6. Restart the Hue service.

Results

You can click Hue Web UI to log on to Hue with a custom MySQL database.

Configuring MySQL server enforced with TLS 1.2 or higher to connect to Cloudera Data Explorer (Hue)

If your MySQL server is configured to use TLS version 1.2 or higher for an encrypted connection with Hue, then you may not be able to start the Hue server and the Kerberos ticket renewer. One solution to fix this problem is to upgrade the MySQL client to version 8.

About this task

One of the errors you may see while connecting to Hue if your MySQL server is configured with TLS 1.2 or higher is: “The health test result for HUE_SERVER_SCM_HEALTH has become bad: This role's process failed to start.” The error might be because the MySQL-Python module for Hue that is shipped with Cloudera uses an older libmysqlclient library which does not support TLS 1.2 and higher.

Important: You must install MySQL database version 8.x to use TLS 1.2-enforced MySQL server with Hue because it uses a newer libmysqlclient library, which supports TLS 1.2 and higher.

Procedure

1. SSH into the Hue server host as a root user.
2. Change directory to /etc/cloudera-scm-agent/ and take a backup of the config.ini file.
3. Run the following command to get a list of all the installed MySQL packages:

   ```
   rpm -qa | grep -i mysql*
   ```

   Save the list of all the installed MySQL packages for future reference.

4. Remove all MySQL Community RPM packages and the Cloudera Manager agent:

   ```
   yum erase mysql-community*|cloudera-manager-agent
   ```

   Uninstalling the Cloudera Manager agent also uninstalls the MySQL-python library. Save the list of all the MySQL packages that are uninstalled for future reference.
5. Reinstall the Cloudera Manager agent as follows:

   ```
   yum -y install cloudera-manager-agent
   ```

   This installs all the required packages:

   ```
   Package Arch Version Repository Size
   ===============================================================
   Installing:
   cloudera-manager-agent x86_64 7.2.4-7594142.el7 cloudera-manager 12 M
   Installing for dependencies:
   MySQL-python x86_64 1.2.5-1.el7 base 90 k
   mysql-community-client-plugins x86_64 8.0.24-1.el7 mysql80-community 189 k
   mysql-community-common x86_64 8.0.24-1.el7 mysql80-community 614 k
   mysql-community-libs x86_64 8.0.24-1.el7 mysql80-community 4.0 M
   mysql-community-libs-compat x86_64 8.0.24-1.el7 mysql80-community 1.2 M

   Transaction Summary
   ===============================================================
   ```
6. Install the mysql-devel (the development header files and libraries for MySQL database client applications):

   ```
   yum install mysql-devel
   ```

   ```
   ===============================================================
   Package Arch Version Repository Size
   ===============================================================
   Installing:
   mysql-community-devel x86_64 8.0.24-1.el7 mysql80-community 6.5 M

   Transaction Summary
   ===============================================================
   ```
7. Install the Python developer package:

   ```
   yum install python-devel.x86_64 0:2.7.5-90.el7
   ```

   ```
   ===============================================================
   Package Arch Version Repository Size
   ===============================================================
   Removing:
   python-devel x86_64 2.7.5-90.el7 @updates 1.0 M

   Transaction Summary
   ===============================================================
   ```
8. Remove the existing _mysql.so file:

   ```
   rm -rf /opt/cloudera/parcels/CDH-[***7.x-VERSION***]/lib/hue/build/env/l
   ib/python2.7/site-packages/_mysql.so
   ```

9. Change directory to /usr/include/mysql and check whether the my_config.h file is present.

   If the my_config.h file is present, then move to the next step.

   If the my_config.h file is not present, then:

   a) Go to https://dev.mysql.com/doc/dev/mysql-server/latest/my__config_8h_source.html and copy the contents of the my_config.h file.
   b) Create a new my_config.h file using the following command:

      ```
      vi /usr/include/mysql/my_config.h
      ```

   c) Paste the content that you copied earlier in the /usr/include/mysql/my_config.h file and save it.
10. Install the MySQL Python library as follows:

    ```
    /opt/cloudera/parcels/CDH/lib/hue/build/env/bin/pip install MySQL-python
     --force-reinstall --ignore-installed
    ```

11. Copy the _mysql.so (MySQL client library) to Cloudera local parcel directory:

    a) Locate the _mysql.so file:

       locate _mysql.so

       Output: /usr/lib64/python2.7/site-packages/_mysql.so
    b) Copy the _mysql.so file to Cloudera's local parcel directory:

       ```
       cp /usr/lib64/python2.7/site-packages/_mysql.so /opt/cloudera/parcels/CD
       H-[***7.x-VERSION***]/lib/hue/build/env/lib/python2.7/site-packages/MySQ
       L_python-1.2.5-py2.7-linux-x86_64.egg/_mysql.so
       ```

12. Log in to Cloudera Manager and restart the Hue service.

Using MariaDB database with Cloudera Data Explorer (Hue)

Hue is packaged with a lightweight embedded PostgreSQL database for proof-of-concept deployments with one Hue server. However, you can install and use a custom MariaDB database as the backend database to store Hue metadata, job and query history, and account information.

MariaDB is a fork of the MySQL relational database. Refer to the MariaDB documentation for information on how to install the database.

Related Information

MariaDB product website

MariaDB documentation

Downloading and installing MariaDB database

Download and install MariaDB database on a host in your Cloudera cluster depending on your Cloudera and OS version.

Procedure

1. Download the MariaDB distribution from https://mariadb.org/download/ based on your OS version.

   Use the Cloudera Support Matrix for finding the compatibile versions.

2. Install the database using the following commands depending on your operating system:

   On RHEL/CentOS:

   ```
   sudo yum install mariadb-server
   ```

   On SLES:

   ```
   sudo zypper install mariadb
   ```

   On Ubuntu:

   ```
   sudo apt-get install mariadb-server    #set root psswd when prompted
   ```

   On Debian:

   ```
   sudo apt-get install mariadb-server    #set root psswd when prompted
   ```

3. Start the database server using the following commands depending on your operating system:

   On RHEL/CentOS:

   ```
   sudo systemctl start mariadb
   ```

   On SLES:

   ```
   sudo systemctl start mariadb
   ```

   On Ubuntu, MariaDB starts automatically after installation. Run the following command to check the status:

   ```
   sudo systemctl status mariadb
   ```

   On Debian, MariaDB starts automatically after installation. Run the following command to check the status:

   ```
   sudo systemctl status mariadb
   ```

4. Secure your installation by running the following command:

   ```
   sudo /usr/bin/mysql_secure_installation
   ```

   mysql_secure_installation is a shell script available on Unix systems, and enables you to improve the security of your MariaDB installation in the following ways:

   ```
   Enter current password for root (enter for none): [If unset, press Enter
   .]
   OK, successfully used password, moving on...
   [...]
   Set root password? [Y/n] Y [Enter n if password is set.]
   New password:
   Re-enter new password:
   Remove anonymous users? [Y/n] Y
   [...]
   Disallow root login remotely? [Y/n] N
   [...]
   Remove test database and access to it [Y/n] Y
   [...]
   Reload privilege tables now? [Y/n] Y
   ```

Configuring MariaDB server

You can custom configure the MariaDB database by configuring the MariaDB server with the option files. The default MariaDB option file is called my.cnf on Unix-like operating systems.

Procedure

1. SSH into the database server host as a root user.
2. Open the my.cnf file for editing in an editor such as vi or nano:

   On RHEL/CentOS/SLES:

   ```
   vi /etc/my.cnf
   ```

   On Debian/Ubuntu:

   ```
   vi /etc/mysql/my.cnf
   ```
3. Ensure that the bind-address property is set to 0.0.0.0 or is commented out in case of a default value.

   ```
   bind-address=0.0.0.0
   ```
4. Ensure that the default-storage-engine property is set to innodb.

   ```
   [mysqld]
   default-storage-engine=innodb
   ```
5. Ensure that the sql_mode property is set to STRICT_ALL_TABLES to avoid columns being truncated during migration.

   ```
   sql_mode=STRICT_ALL_TABLES
   ```
6. Save the file and exit.
7. Restart the database server.
8. Enable the database server to automatically start on system boot:

   On RHEL/CentOS:

   ```
   sudo systemctl enable mariadb
   ```

   On SLES:

   ```
   sudo systemctl enable mariadb
   ```

   On Ubuntu, MariaDB starts automatically after installation. Run the following command to check the status:

   ```
   sudo systemctl status mariadb
   ```

   On Debian, MariaDB starts automatically after installation. Run the following command to check the status:

   ```
   sudo systemctl status mariadb
   ```

Installing and configuring MariaDB on RHEL 8

The steps for installing MariaDB versions 10.3 and 10.4 on RHEL 8 on a Cloudera cluster host are different.

Procedure

1. SSH into the Hue server host.
2. Remove the MariaDB GSSAPI server.

   ```
   yum remove mariadb-gssapi-server
   ```

3. Download the MariaDB repository:

   ```
   curl -LsS https://downloads.mariadb.com/MariaDB/mariadb_repo_setup | bash
    -s -- --mariadb-server-version="mariadb-[***VERSION***]"
   ```

   Replace *[***VERSION***]* with the actual MariaDB version.
4. Install the MariaDB client and server:

   ```
   yum install MariaDB-server MariaDB-client MariaDB-common MariaDB-devel m
   ariadb-libs
   ```

5. Run the following stream editor command to filter and transform the text:

   ```
   sed '/st_mysql_options options;/a unsigned int reconnect;' /usr/include/
   mysql/mysql.h -i.bkp
   ```

6. Install the MySQL Python interface:

   ```
   pip install MySQL-python --force-reinstall --ignore-installed
   ```

7. Grant read and execute permissions to all users and write access to the admin user:

   ```
   chmod -R 755 /usr/lib64/python2.7/site-packages/MySQLdb /usr/lib64/pytho
   n2.7/site-packages/MySQL_python*; chmod 755 /usr/lib64/python2.7/site-pa
   ckages/_mysql*
   ```

8. After installing the Cloudera parcels, copy the site packages:

   ```
   copy /usr/lib64/python2.7/site-packages/_mysql.so /opt/cloudera/parcels/
   CDH/lib/hue/build/env/lib/python2.7/site-packages/MySQL_python-1.2.5-py2
   .7-linux-x86_64.egg/
   ```

What to do next

After installing the MariaDB server, create the Hue database and configure MariaDB as the backend database for Hue.

Installing MySQL client for MariaDB databases

To use MariaDB as a backend database for Hue, you must install the MySQL client and other required dependencies on all the Hue hosts based on your operating system.

For Cent OS

1. SSH into the Hue host as a root user.
2. Install the required dependencies as follows:

   ```
   yum install -y xmlsec1  xmlsec1-openssl
   ```

3. Add the path where you installed the packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```

4. Install the MySQL client as follows:

   ```
   pip3.8 install mysqlclient
   ```

For RHEL

1. SSH into the Hue host as a root user.
2. Install the required dependencies as follows:

   ```
   yum install mysql-devel
   yum install -y xmlsec1  xmlsec1-openssl
   ```
3. Add the path where you installed the packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```
4. Install the MySQL client as follows:

   (RHEL 8)

   ```
   pip3.8 install mysqlclient
   ```

   (RHEL 9)

   ```
   pip3.9 install mysqlclient
   ```

For SLES

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

   ```
   zypper install libmysqlclient-devel
   zypper install xmlsec1
   zypper install xmlsec1-devel
   zypper install xmlsec1-openssl-devel
   ```

   Attention: While installing the mysql-devel and libmysqlclient-devel packages on SLES15, use the “--replacefiles” zypper switch or manually enter yes on the interactive pop-up that you see when the files are being overwritten. Else, you may see an error such as: File /usr/bin/mariadb_config from install of MariaDB-devel-<version>.x86_64 conflicts with file from install of libmariadb-devel-3.1.21-150000.3.33.3.x86_64 (SLES Module Server Applications Updates).
3. Add the path where you installed the packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```
4. Install the MySQL client as follows:

   ```
   pip3.8 install mysqlclient
   ```

For Ubuntu

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

   ```
   apt-get install libmysqlclient-dev
   apt-get install -y xmlsec1
   apt-get install libxmlsec1-openssl
   ```

3. Add the path where you installed the packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```

4. Install the MySQL client as follows:

   ```
   pip3.8 install mysqlclient
   ```

Creating the Cloudera Data Explorer (Hue) database

Hue requires a SQL database to store small amounts of data, including user account information as well as the history of queries and sharing permissions. You must create a database for Hue with UTF-8 character set and grant user privileges.

Procedure

1. SSH into the database server host as the root user.
2. Create a database for Hue with UTF-8 collation and grant user privileges:

   ```
   CREATE DATABASE [***HUE-DATABASE-NAME***] DEFAULT CHARACTER SET utf8 DEF
   AULT COLLATE utf8_general_ci;
   GRANT ALL ON [***HUE-DATABASE-NAME***].* TO '[***HUE-DATABASE-NAME***]'@'%
   ' IDENTIFIED BY '[***PASSWORD***]';
   ```

   Replace [***HUE-DATABASE-NAME***] with the actual Hue database name such as "hue".

   Ensure Hue uses UTF-8 collation and character set.

   ```
   # To create (use utf8_general_ci or utf8mb4_general_ci):
   CREATE DATABASE [***HUE-DATABASE-NAME***] COLLATE = 'utf8_general_ci';
   # To view default_character_set_name and default_collation_name
   SELECT * FROM INFORMATION_SCHEMA.SCHEMATA;
   # To alter if not created with UTF8 collation
   ALTER DATABASE [***HUE-DATABASE-NAME***] COLLATE = 'utf8_general_ci';
   ```

3. Verify the connection to the Hue database by running the following command:

   ```
   mysql -u [***HUE-DATABASE-NAME***] -p
   Enter password: [***PASSWORD***]
   ```

Configuring MariaDB as the backend database for Cloudera Data Explorer (Hue)

After installing and configuring the MariaDB database,you can configure Hue to use it as its backend database.

Procedure

1. Log into Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and filter by Database category.
3. Select the Hue Database Type as MySQL.

4. Set the following database parameters:

   Hue Database Hostname: FQDN of host running MariaDB server

   Hue Database Port: 3306

   Hue Database Username: [***USERNAME***]

   Hue Database Password: [***PASSWORD***]

   Hue Database Name: [***HUE-DATABASE-NAME***] or SID
5. Click Save Changes.
6. Restart the Hue service.

Results

You can click Hue Web UI to log on to Hue with a custom MariaDB database.

Using PostgreSQL database with Cloudera Data Explorer (Hue)

Hue is packaged with a lightweight embedded PostgreSQL database for proof-of-concept deployments with one Hue server. In production environments, you can install and use a custom PostgreSQL database as the backend database to store Hue metadata, job and query history, and account information.

Download and install PostgreSQL

RHEL and CentOS 7 include PostgreSQL version 9.2. However, on these OS versions, Hue is compatible only with PostgreSQL versions 10, 11.x, 12, 13 and 14. Therefore, you must download and install the PostgreSQL distribution separately.

About this task

Note: If you are on CDP Private Cloud Base 7.1.9 SP1, install PostgreSQL version 16 on SLES 15 SP4, SLES 15 SP5, and Ubuntu 22.

For supported PostgreSQL versions and compatibility information, see https://supportmatrix.cloudera.com/.

Before you begin

You must have root privileges or sudo access.

Procedure

1. Download the PostgreSQL distribution from https://www.postgresql.org/download/ based on your OS version.
2. SSH into the host on which you want to install the PostgreSQL database:
3. Install the PostgreSQL server using the following command:

   ```
   sudo yum install postgresql-server
   ```

   (For PostgreSQL 16 on Ubuntu 22 - (CDP Private Cloud Base 7.1.9 SP1 and higher)):

   ```
   sudo apt -y install postgresql-16
   ```

   (For PostgreSQL 16 on RHEL 9 - (CDP Private Cloud Base 7.1.9 SP1 and higher)):

   ```
   sudo dnf install -y postgresql16-server
   ```

Configure the PostgreSQL server

By default, PostgreSQL only accepts connections on the loopback interface. You must reconfigure PostgreSQL to accept connections from the fully qualified domain names (FQDN) of the hosts hosting the services for which you are configuring databases. If you do not make these changes, the services cannot connect and use the database on which they depend.

Procedure

1. Make sure that LC_ALL is set to en_US.UTF-8 and initialize the database as follows:

```
echo 'LC_ALL="en_US.UTF-8"' >> /etc/locale.conf
sudo su -l postgres -c "postgresql-setup initdb"
```

2. Enable MD5 authentication. Edit the pg_hba.conf file typically located at /var/lib/pgsql/data or /etc/postgresql/[***VERSION***]>/main. Add the following line:

```
host all all 127.0.0.1/32 scram-sha-256
```

3. If the default pg_hba.conf file contains the "host all all 127.0.0.1/32 ident" line, then the host line specifying MD5 authentication shown above must be inserted before this line. Failure to do so may cause an authentication error when running the scm_prepare_database.sh script. You can modify the contents of the MD5 line shown above to support different configurations. For example, if you want to access PostgreSQL from a different host, replace 127.0.0.1 with your IP address and update the postgresql.conf file, which is typically found in the same place as pg_hba.conf, to include the following line:

```
listen_addresses = '*'
```

4. Configure settings to ensure your system performs as expected. Update these settings in the /var/lib/pgsql/data/postgresql.conf or /var/lib/postgresql/data/postgresql.conf file based on cluster size and resources as follows:

   Small to mid-sized clusters: Consider the following settings as starting points. If resources are limited, consider reducing the buffer sizes and checkpoint segments further. Ongoing tuning may be required based on each host's resource utilization. For example, if the Cloudera Manager server is running on the same host as other roles, the following values may be acceptable:

- password_encryption = scram-sha-256
- max_connection: In general, allow each database on a host a maximum of 100 connections and then add 50 extra connections. You may have to increase the system resources available to PostgreSQL.
- shared_buffers: 256 MB
- wal_buffers: 8 MB
- checkpoint_segments: 16. Replace configuration parameter checkpoint_segments with min_wal_size and max_wal_size.

  If you previously adjusted checkpoint_segments, the following formula will give you an approximately equivalent setting:

  ```
  max_wal_size = (3 * checkpoint_segments) * 16MB
  ```

  Note:

  The checkpoint_segments setting is removed in PostgreSQL 9.5 and higher. This setting is replaced by min_wal_size and max_wal_size.

  The default setting for max_wal_size is much higher than the default setting for checkpoint_segments. So, adjusting it might no longer be necessary.

  Note:

  If the PostgreSQL cluster is configured to use scram-sha-256 authentication (PostgreSQL 10 or higher), you must ensure that database user passwords use SCRAM encryption. When you create a new user or reset an existing password, set the password encryption method and update the password.

  Run the following commands:

  ```
  SET password_encryption = 'scram-sha-256';
  ALTER USER "[***user***]" WITH PASSWORD '[***password***]';
  ```

  The setting applies only to new or updated passwords. It does not retroactively convert existing passwords.
- checkpoint_completion_target: 0.9

Large clusters: Can contain up to 1000 hosts. Consider the following settings as starting points:

- password_encryption = scram-sha-256
- max_connection: for large clusters, allow each database on a host a maximum of 100 connections and then add 50 extra connections. You may have to increase the system resources that is available to PostgreSQL.
- shared_buffers: 1024 MB. This requires that the operating system can allocate sufficient shared memory.
- wal_buffers: 16 MB. This value is derived from the shared_buffers value. Setting wal_buffers to be approximately 3% of shared_buffers, up to a maximum of approximately 16 MB is sufficient in most cases.
- checkpoint_segments: 128. The PostgreSQL Tuning Guide recommends values between 32 and 256 for write-intensive systems, such as this one. Replace configuration parameter checkpoint_segments with min_wal_size and max_wal_size.

  If you previously adjusted checkpoint_segments, the following formula will give you an approximately equivalent setting:

  ```
  max_wal_size = (3 * checkpoint_segments) * 16MB
  ```

Note:

The checkpoint_segments setting is removed in PostgreSQL 9.5 and higher. This setting is replaced by min_wal_size and max_wal_size.

The default setting for max_wal_size is much higher than the default setting for checkpoint_segments. So, adjusting it might no longer be necessary.

- checkpoint_completion_target: 0.9

5. Configure the PostgreSQL server to start when the system boots.

OS	Command

RHEL 7-compatible, SLES, and Ubuntu	`sudo systemctl start postgresql`
	`sudo systemctl enable postgresql`

6. Restart the PostgreSQL database.

OS	Command
RHEL 7-compatible, SLES, and Ubuntu	`sudo systemctl restart postgresql`

Configure PostgreSQL as the backend database for Cloudera Data Explorer (Hue)

After installing and configuring the PostgreSQL database, you can configure Hue to use it as its backend database.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service and stop the Hue service by clicking ACTIONS Stop .
3. Go to Clusters Hue service Configuration Category Database .

   All the database configurations for Hue are displayed.
4. Select PostgreSQL as the Hue Database Type.
5. Specify the fully qualified domain name (FQDN) of the host on which you have installed the PostgreSQL database in the Hue Database Hostname field.
6. Enter 5432 in the Hue Database Port field.
7. Enter user name and password in Hue Database Username and Hue Database Password fields.
8. Specify the database name in the Hue Database Name field.
9. Click Save Changes and restart the Hue service.

Results

If the database setup is successful and Hue starts without errors, then you should be able to log into the Hue web interface and view the database and tables.

Disabling the share option in Cloudera Data Explorer (Hue)

Hue allows you to share documents, queries, and workflows with other users, either for viewing only or viewing and modifying. Sharing is enabled by default in the Hue UI. For added privacy and control, you can disable sharing by setting the enable_sharing property to false in the Advanced Configuration Snippet.

About this task

The sharing option is always available to the admin users. To disable the share option:

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini and set the value of the enable_sharing property to false in the [desktop] section:

   ```
   [desktop]
   enable_sharing=false
   ```
3. Click Save Changes.
4. Restart the Hue service.

Enabling Cloudera Data Explorer (Hue) applications with Cloudera Manager

Most Hue applications are configured by default, based on the services you have installed.Cloudera Manager selects the service instance that Hue depends on. If you have more than one service, you may want to verify or change the service dependency for Hue. If you add a service such as Oozie after you have set up Hue, you must set the dependency because it is not done automatically.

About this task

To add a dependency in Hue:

Procedure

1. In the Cloudera Manager Admin Console, select Clusters Hue Configuration to navigate to the configuration page for Hue.
2. Filter by ScopeHue (Service-Wide) and CategoryMain .
3. Select the <*SERVICE_NAME*> Service property that you want to set a dependency for. Select none to remove a dependency.
4. Enter a Reason for change..., and then click Save Changes at the bottom of the page to save the configuration change.
5. Refresh the browser page and click the restart icon ⏻ at the top of the page so the new configuration changes can be read by the server and the new setting takes effect.

Running shell commands

You can run shell commands to administer Hue programmatically. For example, to reset the superuser password or to assign an LDAP user superuser permissions.

About this task

To run Hue shell commands:

Procedure

1. Set HUE_CONF_DIR to the latest Hue process directory:

   ```
   export HUE_CONF_DIR="/var/run/cloudera-scm-agent/process/`ls -alrt /var/
   run/cloudera-scm-agent/process | grep HUE_SERVER | tail -1 | awk '{print
    $9}'`"
   echo $HUE_CONF_DIR
   ```

2. Set environment variables used to run the Hue webserver:

   - CentOS/RHEL:

     ```
     for line in `strings /proc/$(lsof -i :8888|grep -m1 python|awk '{ print
      $2 }')/environ|egrep -v "^HOME=|^TERM=|^PWD=|^CLDR_JAVA_OPTS="`;do expo
     rt $line;done
     ```

   - Ubuntu:

     ```
     for line in `strings /proc/$(lsof -i :8888|grep -m1 hue|awk '{ print $2
      }')/environ|egrep -v "^HOME=|^TERM=|^PWD=|^CLDR_JAVA_OPTS="`;do export
     $line;done
     ```

   Note: Environment variables separated by spaces or underscores (_) are not parsed correctely, and must be excluded by using a caret symbol (^) to prevent errors.

3. Run shell subcommands

   When true, HUE_IGNORE_PASSWORD_SCRIPT_ERRORS runs the Hue shell even if hue.ini contains passwords generated by Cloudera Manager (such as bind_password and ssl_password).

   Note: Do not export HUE_IGNORE_PASSWORD_SCRIPT_ERRORS or HUE_DATABASE_PASSWORD to ensure that they are not stored and only apply to this command.

   For CDH parcel deployments:

   - Run the interactive Hue Python shell (Ctrl+D to quit)

     ```
     HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 /opt/cloudera/parcels/CDH/lib/hue/
     build/env/bin/hue shell
     ```

     Or with the database password:

     ```
     HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 HUE_DATABASE_PASSWORD=<your db passw
     ord> /opt/cloudera/parcels/CDH/lib/hue/build/env/bin/hue shell
     ```

   - Change a user password

     ```
     HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 /opt/cloudera/parcels/CDH/lib/hue/bu
     ild/env/bin/hue changepassword admin
     ```

   - Promote Hue user to superuser

     ```
     HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 /opt/cloudera/parcels/CDH/lib/hue/bu
     ild/env/bin/hue shell
     ```

```
from django.contrib.auth.models import User
a = User.objects.get(username='gwen')
a.is_superuser = True
a.save()
```

- Count all of the documents of a certain user:

  ```
  from django.contrib.auth.models import User
  from desktop.models import Document2

  user=User.objects.get(username='demo')
  Document2.objects.documents(user=user).count()

  Out[X]: 1167
  ```

- List available subcommands

  ```
  HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 /opt/cloudera/parcels/CDH/lib/hue/bu
  ild/env/bin/hue
  ```

For CDH package deployments:

-
  ```
  HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1 /usr/lib/hue/build/env/bin/hue shell
  ```

Downloading and exporting data from Cloudera Data Explorer (Hue)

Hue enables you to download or export data from Hue to HDFS or to an external storage location from Hue Editor, Hue Dashboard, and the Hue File browser. You can limit the number of rows or bytes that are downloaded or disable the export feature altogether so that you do not run out of storage space.

For a service-wide change, go to Cloudera Manager Clusters Hue service Configuration and specify the configurations in the Hue Service Advanced Configuration Snippet (Safety valve) for hue_safety_valve.ini field.

By default, Hue users can download the query results from the Hue Editor, the Hue Dashboard, and the File browser.

Limiting the number of rows to download

Specify the following in the Hue Service Advanced Configuration Snippet (Safety valve) for hue_safety_valve.ini to limit the number of rows that can be downloaded from a query before it is truncated:

```
[beeswax]
download_row_limit=X
```

X represents the number of rows that you can download.

By default, there is no download limit, and you can configure this by setting the value to “-1”:

```
[beeswax]
download_row_limit=-1
```

Limiting the number of bytes to download

Specify the following in the Hue Service Advanced Configuration Snippet (Safety valve) for hue_safety_valve.ini to limit the number of bytes that can be downloaded from a query before it is truncated:

```
[beeswax]
download_bytes_limit=X
```

X represents the number of bytes that you can download.

By default, there is no download limit, and you can configure this by setting the value to “-1”:

```
[beeswax]
download_bytes_limit=-1
```

Disabling the data download feature

Specify the following in the Hue Service Advanced Configuration Snippet (Safety valve) for hue_safety_valve.ini field to disable your users from downloading query results:

```
[desktop]
enable_download=false
```

Specify the following in the Hue Service Advanced Configuration Snippet (Safety valve) for hue_safety_valve.ini field to hide the Download button from the Hue File browser:

```
[filebrowser]
show_download_button=false
```

Backing up the Cloudera Data Explorer (Hue) database

Cloudera Manager provides you (the Hue service administrator) an option to easily back up your database without the need to engage your database administrators or perform this operation manually.

About this task

You may have the need to back up the Hue database before upgrading or migrating to a new database, or for any other purposes. You can do so from Cloudera Manager as follows:

Procedure

1. Log in to Cloudera Manager as Administrator.
2. Go to Clusters Hue service .
3. Stop the Hue service by clicking Actions Stop .
4. After the Hue service has successfully stopped, click Actions Dump Databse .
5. On the **Dump Database** popup, click Dump Databse to confirm the operation.

   The database dump is written to the /tmp/hue_database_dump.json file on the Hue server host.
6. Start the Hue service by clicking Actions Start .

Backing up and restoring Cloudera Data Explorer (Hue) documents in Cloudera Runtime

You can back up and restore Hue data, such as a Hive or Impala query you saved in a JSON document, in the Hue database to prevent losing the documents after rebuilding the Database Catalog.

About this task

In Hue, you can view the JSON documents you created or imported in My documents.

Before you begin

Avoid accessing or using the Hue web interface until the restore process is fully complete, as concurrent access may disrupt the operation and potentially lead to failure.

During the Hue database restore operation, it is critical to block any traffic to the running Hue services. If you cannot bring down the cluster, use the recommended workaround to disable end user access to the cluster endpoints. Failing to do so results in errors in addition to existing key constraints and other issues.

Procedure

1. SSH into an active instance.

   ```
   ssh user@<hue-host>
   ```

2. Navigate to the Hue environment directory.

   ```
   cd /opt/hive/build/env/bin
   ```

3. Backup Hue documents you want to preserve from the command line.

   ```
   ./build/env/bin/hue dumpdata -o /tmp/hue_data_backup.json
   ```

4. Verify backup file content.

   ```
   head /tmp/hue_data_backup.json
   ```

5. Copy the backup file to a secure location.

   ```

   # On a local machine
   cp /tmp/hue_data_backup.json /backup/location/

   # On a remote server
   scp /tmp/hue_data_backup.json user@backup-server:/backup/location/
   ```

6. Restore the Hue documents.
   a) Copy the backup JSON file back to the Hue host (if required).

      ```
      scp user@backup-server:/backup/location/hue_data_backup.json /tmp/
      ```

   b) Navigate to the Hue installation directory.

      ```
      cd /opt/cloudera/parcels/CDH/lib/hue
      ```

   c) Run the restore command.

      ```
      ./build/env/bin/hue loaddata --ignorenonexistent /tmp/hue_data_backup.js
      on
      ```

7. Restart the Hue service.

   ```
   systemctl restart hue
   ```

Results

The saved queries, workflows, and dashboards are restored.

What to do next

- Perform regular backups (for example, weekly) to prevent accidental data loss.
- Always restore to the same or a compatible Hue version.
- Verify database connectivity and schema validity before restoration.
- Restrict end-user access during the restore process to avoid conflicts.

Enabling a multi-threaded environment for Cloudera Data Explorer (Hue)

A multi-threaded environment can help reduce the time it takes to migrate data from one database to other. By default, operations such as migrating data run on a single thread. For example, if you are switching from MySQL as the backend database for Hue to Oracle, then enabling a multi-threaded environment significantly reduces the data transfer time.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini .
3. Locate the [[database]] section under [desktop] and set threaded to true:

   ```
   [desktop]
   [[database]]
   options={"threaded":true}
   ```

4. Click Save Changes.
5. Restart the Hue service.

Moving the Cloudera Data Explorer (Hue) service to a different host

After adding new hosts to your Cloudera cluster if you need to move the Hue service from one host to another, then either add a new Hue service and manually configure it on the new host or add new role instances for Hue server, Hue Load Balancer, and Kerberos Ticket Renewer on new hosts.

Adding and configuring a new Cloudera Data Explorer (Hue) service on a new host

To migrate the Hue service from one host to another within the same cluster, you can add a new Hue service on a new host using the Add Service wizard and manually configure the Hue service as needed.

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service and stop the Hue service by clicking Actions Stop .
3. Go to Clusters [***CLUSTER-NAME***] and click Add Service from the menu.
4. On the **Add Service** wizard, select Hue.
5. On the **Select Dependencies** page, select the Optional Dependencies.

   Important: In CDH 6 and earlier, the Hive service included the Hive Metastore and HiveServer2. In Cloudera Runtime 7.0 and later, this service only includes the Hive Metastore. HiveServer2 and other components of the Hive execution engines are part of the HIVE_ON_TEZ service.

   If you need to run Hive queries, then select an optional dependency containing the HIVE_ON_TEZ service. If you need to run only Impala queries, then the HIVE_ON_TEZ service is not required.
6. On the **Assign Roles** page, click the field below Hue Server and Load Balancer roles. This launches a dialog box where you can select hosts where you want to add the role.

   Click OK after making your selection.
7. On the **Setup Database** page, specify the details of your existing Hue database.
   a) Select your database vendor from the Type field and specify the Database Hostname, Database Name, Username, and Password of your existing database.
   b) Click Test Connection, and when the success message appears, click Continue.

   The wizard configures and starts Hue and its dependent services.
8. After the new Hue service is in the running state, configure the Hue Advanced Configuration Snippet as it is in the existing Hue service on the older host.
9. Click Save Changes and restart the Hue service.
10. Verify that the new Hue service functions as expected.
11. Delete the Hue service from the old host by clicking Actions Delete .

Adding new role instances for Cloudera Data Explorer (Hue) server, Cloudera Data Explorer (Hue) Load Balancer, and Kerberos Ticket Renewer on new hosts

To migrate the Hue service from one host to another within the same cluster, you can add new hosts to your Cloudera cluster and then add the Hue Server, Hue Load Balancer, and Kerberos Ticket Renewer to the new host while inheriting the configurations from the existing roles.

Procedure

1. Sign in to Cloudera Manager as an Administrator.
2. Add new hosts for Hue server, Hue Load Balancer, and Kerberos Ticket Renewer to your cluster by going to Hosts Add Hosts .
3. Go to Clusters Hue service Instances and click Add Role Instances.
4. On the **Assign Roles** page, specify the role assignments for Kerberos Ticket Renewer, Load Balancer, and Hue Server by clicking Select hosts.
5. Select the new hosts that you have added to your CDP cluster for each of these roles.
6. Click Continue, review your changes, and exit the wizard.
7. Go back to the Instances tab and start the new role instances by selecting the newly added roles and click Actions for Selected Start .

   The newly configured roles inherit settings and configurations from the existing roles, including the configuration in the Advanced Configuration Snippet.
8. Verify that the new roles have the same configuration as the existing roles and the Hue service functions as expected.

   ⚠ **Important:** If you have applied host-specific configuration, then you must manually apply the same configuration to your new host.
9. Delete the Hue roles from the old host by selecting them and clicking Actions for Selected Delete .
10. Restart the Hue service.

Adding Query Processor service to a cluster

The Query Processor service indexes Hive and Tez events and provides APIs to access them. It is required if you want to view the Queries tab (query history and query details) on the Hue Job Browser. You must install the Query Processor service on your Cloudera Base on premises clusters manually.

About this task

⚠ **Attention:** This task is applicable only if you are upgrading to CDP Private Cloud Base 7.1.8 and higher. If you are upgrading to CDP Private Cloud Base 7.1.7 SP2, 7.1.7 SP1, 7.1.7, or lower, then you can skip this task.

Note: The Query Processor service does not support MariaDB.

You can either install the Query Processor service as a dependency while adding the Hue service or after adding the Hue service.

Before you begin

This task assumes that you already have a database installed on a host in your cluster. On Cloudera, Hue Query Processor supports PostgreSQL, MySQL, and Oracle databases. The supported PostgreSQL database version for Hue Query Processor is 9.6 and higher.

Next, you need to create a database for the Query Processor service with the required roles. To create the Query Processor database:

1. SSH into your database host as a root user.

2. Start the database terminal by running the following commands:

   (PostgreSQL)

   ```
   sudo -u postgres psql
   ```

   (MySQL)

   ```
   systemctl start mysqld
   ```

   (Oracle)

   ```
   sqlplus / as sysdba
   ```

3. Run the following statement while specifying the username, password, and a database name for the Query Processor:

   (PostgreSQL)

   ```
   CREATE ROLE [***QP-USER***] WITH LOGIN PASSWORD '[***QP-PASSWORD***]';
   ALTER ROLE [***QP-USER***] CREATEDB;
   CREATE DATABASE [***QP-DATABASE***];
   GRANT ALL PRIVILEGES ON DATABASE [***QP-DATABASE***] TO [***QP-USER***];
   ```

   (MySQL)

   ```
   CREATE USER [***QP-USER***] IDENTIFIED BY '[***QP-PASSWORD***]';
   CREATE DATABASE [***QP-DATABASE***];
   USE [***QP-DATABASE***];
   GRANT ALL PRIVILEGES ON [***QP-DATABASE***].* TO [***QP-USER***];
   ```

   (Oracle)

   ```
   CREATE USER [***QP-USER***] IDENTIFIED BY [***QP-PASSWORD***];
   GRANT ALL PRIVILEGES TO [***QP-USER***];
   ```

   Note: When you create a user in the Oracle database, the associated schema is created automatically. This schema becomes the default schema for that user.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters ⋮ Add Service .
3. Select Query Processor on the **Add Service to Cluster** page and click Continue.

   The required dependencies are automatically selected.
4. Select the host on which you want to install the Query Processor by clicking View By Host. Then click Continue.

5. Select the database type, and specify the database hostname, database name, and username and password to access the database on the **Setup Database** page and click Test Connection.

   ⚠ **Attention:**

   - (For MySQL and Oracle databases) Ensure that you specify the same name in the Database Name and Database Schemas fields.
   - Oracle databases: Schema and object names are case-insensitive by default. Oracle automatically converts unquoted names to uppercase. If names are created using double quotes, they become case-sensitive and must be referenced exactly as created, including the correct case and quotes.
   - PostgreSQL databases: User, database, and object names are case-insensitive by default. PostgreSQL automatically converts unquoted names to lowercase. If names are created using double quotes, they become case-sensitive and must be referenced exactly as created, including the correct case and quotes.

   After the connection is verified, click Continue.
6. On the **Review Changes** page, accept the default settings and click Continue.

   The **Command Details** page is displayed, which shows the status of the installation and configuration.
7. Click Continue if the operation is successful.
8. Click Finish on the **Summary** page.

Results

You are redirected to the Cloudera Manager home page. You can now see the Query Processor service listed within your cluster.

What to do next

(Optional) To view the **Queries** tab on the **Job Browser** page in Hue, you must enable the Query Processor Service option in Hue configurations using Cloudera Manager.

Removing Query Processor service from cluster

You can remove the Query Processor service using Cloudera Manager when you no longer need it.

Before you begin

Because Hue has a dependency on the Query Processor service, deselect the Query Processor Service option in the Hue configurations before proceeding to remove service dependencies.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Query Processor service ⋮ Delete .

   If you have not removed the service dependencies before deleting the Query Processor, then a **Remove Service Dependencies** dialog box is displayed.

   a) Click Configure Service Dependency.
   b) Deselect the Query Processor Service option on the **Configuration** page and click Save Changes.
   c) Go to Clusters Query Processor service ⋮ Delete .

Enabling the Query Processor service in Cloudera Data Explorer (Hue)

The Queries tab on Hue's Job Browser page allows you to view only Hive query history and details. You need to enable the Query Processor service from the Hue configurations to enable the Queries tab. Hue does not display Impala query history and query details. Currently, they are not supported and will result in an error if accessed.

About this task

The Query Processor service is automatically enabled in Hue if you select it as a dependency on the **Select Dependencies** page while adding the Hue service to your cluster. But if you did not select it as a dependency while adding the Hue service, you can add the Query Processor service to your cluster (if not installed already) and then enable the Query Processor Service option in Hue configurations.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration and select the Query Processor Service option.
3. Click Save Changes.
4. Restart the Hue service.

Results

You can now view Hive query history and query details under the **Queries** tab on Hue’s **Job Browser** page.

Adding Query Processor Administrator users and groups on Cloudera Base on premises clusters

Query Processor Administrators have special privileges that enable them to view and monitor queries from all users, including the ones that are submitted from query interfaces, such as Beeline, Hive Warehouse Connector (HWC), Tableau, and so on.

About this task

This task is applicable only if you are running Hive queries.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Query Processor Configuration and enter the list of users in the Extra Auth Options field as follows:

   "adminUsers": "hive, [***USER-1***], [***USER-2***]"
3. You can also add a list of admin groups as follows:

   "adminGroups": "admin-group, [***GROUP-1***], [***GROUP-2***]"

   Note: The adminGroups setting must reference local Unix groups available on the cluster hosts. LDAP groups are not supported directly for the adminGroups configuration.
4. To configure both administrator users and groups together, use the following example:

```
"adminUsers": "user1,user2,user3",
"adminGroups": "group1,group2"
```

5. Click Save Changes.
6. Restart the Query Processor service.

Ways to clean up old queries from the Query Processor tables

Learn how to schedule a query clean-up and how to use the API to manually clean up queries from the following Query Processor tables: vertex_info, dag_details, dag_info, query_details, hive_query, tez_app_info.

Scheduling query clean-up

By default, the Hue Query Processor automatically clears queries from the Query Processor tables that are 30 days old at 2 AM every day as per the timezone set on your server. You can configure the following parameters in Cloudera Manager Query Processor service Configurations Query Processor Extra Configurations to set up a schedule based on your requirements:

- hue.query-processor.event-pipeline.cleanup.cron.expression
- hue.query-processor.event-pipeline.cleanup-interval-secs

For example:

```
"hue.query-processor.event-pipeline.cleanup.cron.expression" : "0 0 2 * * ?"
,
"hue.query-processor.event-pipeline.cleanup-interval-secs" : "2592000"
```

Manually cleaning up queries using an API

The ability to clean up queries manually in addition to the scheduled clean-up routines is useful when you have a high load of queries in a particular week that are hogging resources that you must free up. The API also runs a VACUUM command on the Query Processor table to reclaim storage that is occupied by dead tuples.

You can send an API request using tools such as cURL or Postman.

API format: `[***X-Do-As:[COMPONENT/PROCESS USER ID]***][***QUERY-PROCESSOR-ADDRESS***]/api/admin/cleanup/[***EPOCH-TIME***]`

Where,

- [***QUERY-PROCESSOR-ADDRESS***] is the query processor host address
- [***EPOCH-TIME***] is the Unix epoch time in seconds

Queries that were run before the specified epoch time are purged.

For example:

```
curl -k --negotiate -u : -H "X-Do-As: hive" "https://machine1.company.com:30
700/api/admin/cleanup/1670006742"
```

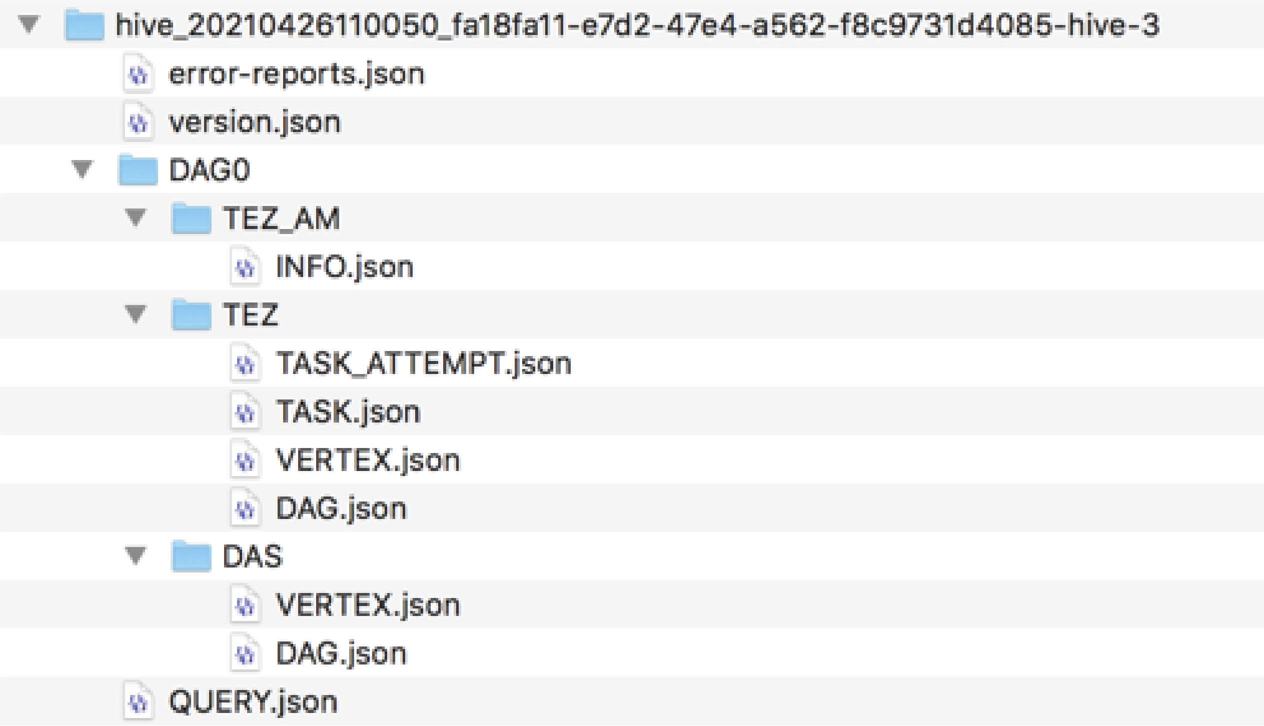
Downloading debug bundles

The debug bundle is a ZIP file that contains the query details in JSON format and an error-reports.json file, which is created only if an error occurs while the query is run.

About this task

Note: This feature is available only for Hive queries.

If Tez is used to run a query, then the debug bundle also contains DAG and Tez JSON files, as shown in the following image:



```
hive_20210426110050_fa18fa11-e7d2-47e4-a562-f8c9731d4085-hive-3
    error-reports.json
    version.json
    DAG0
        TEZ_AM
            INFO.json
        TEZ
            TASK_ATTEMPT.json
            TASK.json
            VERTEX.json
            DAG.json
        DAS
            VERTEX.json
            DAG.json
    QUERY.json
```

Procedure

1. Log in to the Hue web interface.
2. Click Jobs from the left assist panel.

   The **Job Browser** page is displayed.
3. Click Queries.

   The Hive queries that were run are displayed.
4. Select a query for which you want to download the debug bundle.
5. Click Download and save the ZIP file on your computer.

   The filename is in the following format:

   ```
   hive_[***HIVE-QUERY-ID***]_[***USER-ID***]_[***UNIQUE-INDEX***]
   ```

Enabling DSL search for Cloudera Data Explorer (Hue)

Domain-Specific Language (DSL) calls used for polling Atlas use the LIKE operator and hence they are inefficient, resource-intensive, and expensive. By default, Hue uses a basic search to look up data in Atlas. If needed, you can enable DSL for polling Atlas entities by setting the search_cluster property in the Advanced Configuration Snippet.

Before you begin

Obtain the Atlas metadata namespace from the Atlas application properties file. Search the file for the search_cluster property and note its value.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration .
3. Set the value of the search_cluster property in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [metadata]
   [[catalog]]
   api_url=https://[***HOSTNAME***]:[***PORT***]
   interface=atlas
   search_cluster=cm ([***ATLAS.METADATA.NAMESPACE***])
   kerberos_enabled=true
   enable_basic_search=false
   ```

   Replace [***ATLAS.METADATA.NAMESPACE***] with the value of the search_cluster property you obtained from the Atlas application properties file.
4. Click Save Changes.
5. Restart the Hue service.

Configuring Cloudera Data Explorer (Hue) to handle HS2 failover

Hue can handle HiveServer2 (HS2) failovers using ZooKeeper without the need to set up a load balancer.

About this task

Note: Zookeeper configuration only supports failover situations. It cannot be used to balance loads between the active HS2 servers.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration and add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [beeswax]
   hive_discovery_hs2=true
   hive_discovery_hiveserver2_znode=/hiveserver2
   ```

   Note: hiveserver2 is the default value of the hive_discovery_hiveserver2_znode property. If this value is different from the Hive-On-Tez configuration, then you must obtain the information from the hive.server2 .zookeeper.namespace property from Hive-On-Tez configuration.
3. Click Save Changes.
4. Restart the Hue service.

Enabling Spark 3 engine in Cloudera Data Explorer (Hue)

Hue leverages Apache Livy 3 to support Spark SQL queries in Hue on the Apache Spark 3 engine. To enable the Spark 3 engine, install the Spark 3 binaries on your cluster, add “Livy for Spark 3” and “Spark 3” services, and configure settings for these services using Cloudera Manager.

Before you begin

Note: Livy v0.6 supports Python versions upto Python 3.7. If you install Python 3.8 or higher, then you may see the following error: TypeError: required field "type_ignores" missing from Module.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Add the following services by going to Clusters Actions and clicking Add Service:
   - Livy for Spark 3
   - Spark 3

   Important:

   - While adding the Livy for Spark 3 service to your cluster, you must select Hive as optional dependency on the **Select Dependencies** page.
   - While adding the Spark 3 service to your cluster, you must select an optional dependency containing Kafka on the **Select Dependencies** page and select the Spark Authentication option on the **Review Changes** page.
3. Go to Clusters HDFS Configuration and add the following lines in the Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml field:

```
<property>
<name>hadoop.proxyuser.hue.groups</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.hue.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.spark.groups</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.spark.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.livy.groups</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.livy.hosts</name>
<value>*</value>
</property>
```

4. Click Save Changes.

5. Go to Clusters Livy for Spark 3 service Configuration and add the following configurations:
   a) Add the hue user in the Admin Users (livy.superusers) field.
   b) Go to the HMS Service field and select Hive.
   c) Click Save Changes.
6. Go to Clusters SPARK_ON_YARN Configuration Admin Users , add hue to the list of admin users (spark.history.ui.admin.acls) and click Save Changes.
7. Go to Clusters SPARK Configuration Admin Users , add hue to the list of admin users (spark.history.ui.admin.acls) and click Save Changes.
8. Go to Clusters SPARK 3 Configuration Admin Users , add hue to the list of admin users (spark.history.ui.admin.acls) and click Save Changes.
9. Go to Clusters Hue Configuration and enter the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field and click Save Changes:

```
[desktop]
app_blacklist=zookeeper, pig #custom list of blocked apps
[spark]
#This is not a thrift server port
#If this TLS/SSL is enabled then check to see whether the livy url is on h
ttps or http and modify the url accordingly.
livy_server_url=http(s)://[***LIVY-FOR-SPARK3-SERVER-HOST***]:[***LIVY-
FOR-SPARK3-SERVER-PORT***]
ssl_cert_ca_verify=false
security_enabled=true
[notebook]
[[interpreters]]
[[[sparksql]]]
name=Spark SQL
interface=livy
```

Attention:

- Ensure that the Spark application is not on the blocked list.
- Set ssl_cert_ca_verify=false if an SSL certificate is not present in Hue's truststore. ssl_cert_ca_verify=true if your environment is a secured environment.
- Set security_enabled=true for Kerberized clusters.

10. Restart the affected services.

Results

You can now select the “Spark SQL” dialect on the Hue editor and run Spark queries from Hue.

Note: Starting a Livy session can take 30-45 seconds. If you cancel a running Spark SQL query and rerun it, the Livy session expires, and a new session is created.

Enabling the Phoenix SQL editor in Cloudera Data Explorer (Hue)

To query Apache HBase tables from Hue using Phoenix SQL, configure the Phoenix interpreter settings in the Hue Advanced Configuration Snippet.

Before you begin

Add {USER} to the HBASE (cm_hbase) Ranger policy that governs access to all tables, columns, column-family, and so on. Grant {USER} the required permissions such as read, write, create, execute, admin, and so on.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration and add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

   ```
   [notebook]
   [[interpreters]]
   [[[phoenix]]]
   name=phoenix
   interface=sqlalchemy
   options='{"url": "phoenix://[***PHOENIX-URL***]:[***PHOENIX-PORT***]"}'
   ```

   Attention: If you are using security, then specify the following in the options property:

   ```
   options='{"url": "phoenix://[***PHOENIX-URL***]:[***PHOENIX-PORT***]
   ", "tls": true, "connect_args": "{\"authentication\": \"SPNEGO\", \"
   verify\": false }", "has_impersonation": true}'
   ```
3. Click Save Changes.
4. Restart the Hue service.

Using Cloudera Data Explorer (Hue) scripts

Hue scripts help you to share workflows, set up default editors, change ownership of documents, and test the backend processes using cURL commands. The Hue scripts are included with Cloudera.

Sharing workflows with users and groups

Use the following command to share workflows with users and groups:

```
./build/env/bin/hue share_all_workflows --owner [***OWNER-NAME***] --sharegr
oups [***GROUP1/USER1***],[***GROUP2/USER2***] --permissions read,write
```

To share users, use the --shareusers option instead of the --sharegroups option.

Changing the document owner

Use the following command to change the owner of the Hue documents:

```
./build/env/bin/hue change_owner_of_docs --olduser [***OLD-USER-NAME***] --n
ewuser [***NEW-USER-NAME***]
```

Testing the backend using cURL

Use the following command to get the syntax for the backend cURL commands or to test the endpoints:

```
./build/env/bin/hue get_backend_curl --showcurl --testname [***SERVICE-NAME-
WHERE-TEST-NEEDS-TO-BE-PERFORMED***] --service [***COMMA-SEPARATED-SERVICES-
TO-TEST]
```

Replace [***SERVICE-NAME***] with the name of the service where you want to perform the test. Replace [***COMMA-SEPARATED-SERVICES-TO-TEST***] with options or services such as all, httpfs, solr, oozie, rm, jhs, sparkhs, and so on.

Setting the default editor in Hue

Use the following command to set the default editor in Hue:

```
./build/env/bin/hue set_default_editor --hive/impala --username [***USERNAME
***]
```

The default value for the username is all.

Configurations for submitting a Hive query to a dedicated queue

When you run a Hive query from Hue or Beeline, HiveServer2 submits the query in YARN in the queue mapped from an end user instead of a hive user. If you run a query while specifying doAs=false, then the queries are submitted and run as a hive user, using impersonation. This is the default behaviour. You can change this behaviour by configuring the queue using "yarn_queue_manager" and "application tag" parameters in YARN.

If you see go to the **Job Browser** page in Hue and see that the Hive queries are appearing under the default queue, and not the actual user who ran the query, then this means that the queries are not being run in their dedicated queue in YARN. To submit a Hive query in its dedicated queue, you must enable application-tag based scheduling by configuring queue mapping and per queue properties.

Related Information

Configuring queue mapping to use the user name from the application tag using Cloudera Manager

Configure Per Queue Properties

Enabling browsing Ozone from Cloudera Data Explorer (Hue) on Cloudera Base on premises

Hue can read and write files on the Ozone filesystem, similar to S3 or ADLS. To access Ozone from Hue, you must add the additional configurations in the hue_safety_valve section, under Hue Service Advanced Configuration S nippet (Safety Valve) for hue_safety_valve.ini in the Cloudera Manager.

About this task

You can perform this task on any Hue instances of any environment in which you want to enable the Ozone File Browser. If multiple Hue instances exist within the same cluster, completing this task on one instance per environment is sufficient.

Before you begin

- Ensure that the Ozone HttpFS Gateway role is running in a healthy state.
- Go to Cloudera Manager Ozone Configuration and add the following entries in the HttpFS Gateway Advanced Configuration Snippet (Safety Valve) for ozone-conf/httpfs-site.xml field:

Field name	Value
Name	httpfs.proxyuser.[***PRINCIPAL-NAME***].hosts
Value	*
Name	httpfs.proxyuser.[***PRINCIPAL-NAME***].groups

Field name	Value
Value	*

Replace the [***PRINCIPAL-NAME***] with the actual Kerberos principal name. The **hive** principal is the default principal required for communication between Ozone and Hue. If this principal name is changed during installation (e.g., to a custom principal for the Hive service), use that modified principal name here instead.

- Obtain the following values from the Ozone service. You need it to construct the fs_defaultfs and webhdfs_url URLs:
  - HttpFS Gateway host name (Gateway node)
  - Ozone HttpFS Gateway TTP Web UI Port (ozone.httpfs.http-port)

    The default port is 9778. Ensure that the port used by the Ozone HttpFS Gateway (or any other configured port for that HttpFS instance) is accessible from the Hue node to the Gateway node where the Ozone HttpFS Gateway is installed.
  - Ozone Service ID (ozone.service.id).

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration and add the following lines in the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field:

```
[desktop]
rest_conn_timeout=300
enable_chunked_file_uploader=false
[[task_server_v2]]
enabled=false
[[ozone]]
[[[default]]]
fs_defaultfs=ofs://[***SERVICE-ID***] \\The ozone.service.id value
webhdfs_url=https://[***OZONE-HTTPFS-HOST***]:[***OZONE-HTTPFS-PORT***]/w
ebhdfs/v1
ssl_cert_ca_verify=true
security_enabled=true
[[database]]
port=0
options={"threaded":true}
[beeswax]
hive_discovery_hs2=true
hive_discovery_hiveserver2_znode=/hiveserver2
[hadoop]
upload_chunk_size=2147483648
```

Configurations	Descriptions
`rest_conn_timeout=300`	This optional configuration extends the default data push timeout in Ozone from 120 seconds to 300 seconds. This can enhance performance when transferring larger data chunks to Ozone, especially over slower network connections.

`enable_chunked_file_uploader=false`
`[[task_server_v2]]`
	This configuration is necessary because Ozone does not support chunked uploads or the `/append` HttpFS API. Therefore, files must be uploaded as a single, complete chunk into Ozone

Configurations	Descriptions
`enabled=false`	

`[[ozone]]`
`[[[default]]]`
`fs_defaultfs=ofs://[***SERVICE-ID***] \\The ozone.service.id value`
`webhdfs_url=https://[***OZONE-HTTPFS-HOST***]:[***OZONE-HTTPFS-PORT***]/webhdfs/v1`
`ssl_cert_ca_verify=true`
`security_enabled=true`
	This configuration enables you to browse objects within Ozone DB buckets from Hue.

`[hadoop]`
`upload_chunk_size=2147483648`
	This configuration increases the default file upload chunk size from 64 MB to 2 GB, which is the maximum supported by the Django code used by Hue when uploading files.

`[[database]]`
`port=0`
`options={"threaded":true}`
	This configuration is required only if the metastore is set on an external Oracle database.

`[beeswax]`
`hive_discovery_hs2=true`
`hive_discovery_hiveserver2_znode=/hiveserver2`
	This configuration enables ZooKeeper to handle HiveServer2 failovers.

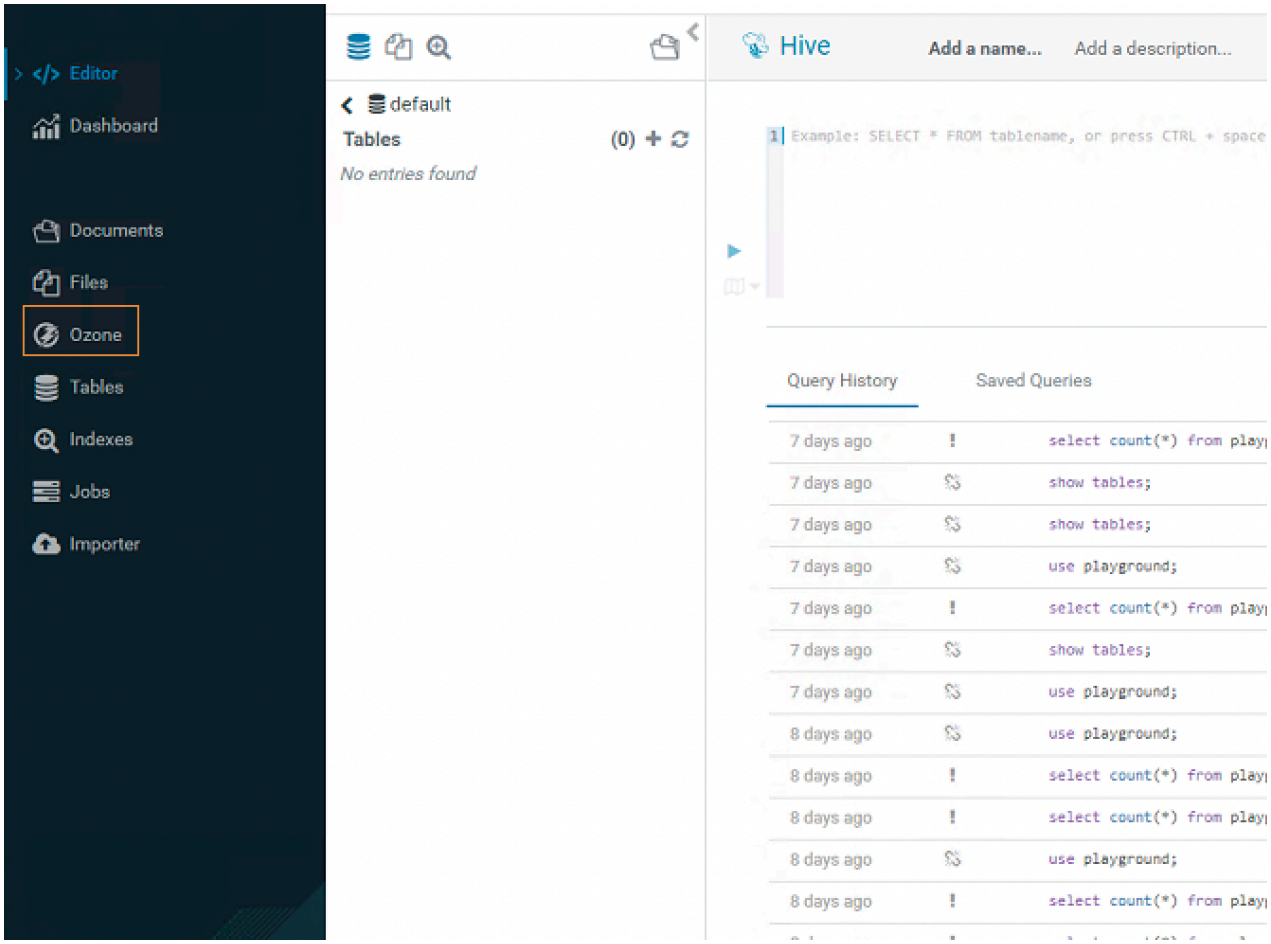
Note:

- If the cluster is not secured, set the webhdfs_url property to http.
- If Ozone is not configured in HA mode, then specify the URL for the Ozone manager in the fs_defaultfs property.

3. Click Save Changes.
4. Log in to Hue as an Administrator on any one instance within the environment.
5. Click your username in the lower-left corner of the interface, and select Administer Users.
6. Navigate to the Groups tab, select the default group, and ensure filebrowser.ofs_access:Access to OFS from filebrowser and filepicker permission is selected.
7. Click Update group to save the changes.
8. Restart the Hue service.

Results

After configuring the Hue safety valve and restarting the Virtual Warehouse, you may see that the Ozone file browser may take 10 - 15 minutes to display on the Hue web interface, as shown in the following image.



```
1 Example: SELECT * FROM tablename, or press CTRL + space
```

7 days ago	!	select count(*) from play
7 days ago		show tables;
7 days ago		show tables;
7 days ago		use playground;
7 days ago	!	select count(*) from play
7 days ago		show tables;
7 days ago		use playground;
8 days ago		use playground;
8 days ago	!	select count(*) from play
8 days ago	!	select count(*) from play
8 days ago		use playground;
8 days ago	!	select count(*) from play

Managing file extensions for Cloudera Data Explorer (Hue) uploads

You can allow and restrict file extensions across all configured file systems in Hue to control file uploads. For instance, you can allow .csv files while blocking .exe files.

Procedure

1. Log in to Cloudera Manager Clusters Hue service Configuration .
2. In the search, enter hue_safety_valve.ini to locate the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini field.
3. Set the configuration in the Hue Service Advanced Configuration Snippet (Safety Valve) for the hue_safety_valve.ini field as follows:

```
[filebrowser]
restrict_file_extensions = .sh, .exe #Specify file extensions that are not
 allowed, separated by commas.
```

```
allow_file_extensions = .tsv, .csv  #Specify file extensions that are al
lowed, separated by commas.
```

Note:

- By default, no file extensions are restricted during file uploads.
- Files are allowed only if they match the allowed file extension type and are not listed in the restricted file extension types. Otherwise, an error message is displayed.

4. Click Save Changes.
5. Restart the Hue service.

Enabling multiple editors in Cloudera Data Explorer (Hue)

Data Explorer now no longer displays all available interpreters by default. If you prefer to display all available interpreters in the UI, you can explicitly enable them using a configuration flag.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Cloudera Manager Clusters Hue Configuration .
3. Search for the Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini.
4. Add the following flag to enable displaying all interpreters in the Data Explorer navigation pane:

```

[notebook]
enable_all_interpreters=true
```

5. (Optionally) Explicitly configure interpreters by adding entries under the [[interpreters]] section:

```
[notebook]
[[interpreters]]
[[[<interpreter_name>]]]
name=[***display_name***]
interface=[***interface_type***]
```

6. Click Save Changes.
7. Restart the Data Explorer service.

Configuring timezone for Cloudera Data Explorer (Hue)

You can configure the timezone for Hue using Cloudera Manager. The default value is America/Los_Angeles.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue Configuration Time Zone and set a desired value.

   The default value is “America/Los_Angeles”. You can refer to https://timezonedb.com/time-zones for information on the timezones.
3. Click Save Changes.
4. Restart the Hue service.

How to configure a specific Python version for Hue

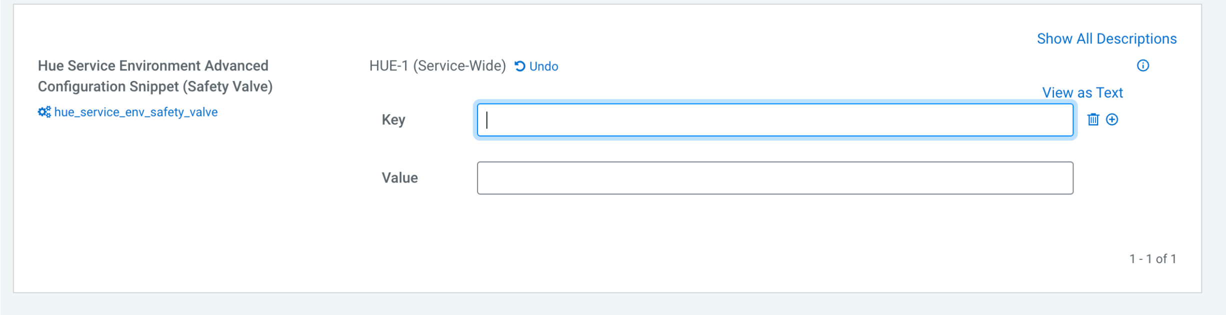
Hue provides the flexibility to use a specific version of Python, as long as that version is installed on the relevant cluster nodes. By default, Hue automatically selects the highest available Python version on the system.

Procedure

1. Log in to Clouder Manager as an Administrator.
2. Go to Clusters Hue Configuration add the following key and value in the Hue Service Environment Advanced Configuration Snippet (Safety Valve) field:

   Key: HUE_PYTHON_VERSION

   Value: *PYTHON3.X (FOR EXAMPLE, PYTHON3.11 OR PYTHON3.9)*

   
   Show All Descriptions
Hue Service Environment Advanced Configuration Snippet (Safety Valve)
hue_service_env_safety_valve
HUE-1 (Service-Wide)
Undo
View as Text
Key
Value
1 - 1 of 1

3. Click Save Changes.
4. Restart the Hue service.