Hue: The Next Generation SQL Assistant for Hive in CDW (Preview)

Date published: 2021-06-02
Date modified: 2021-06-10

Legal Notice

Contents

Overview

Hue is a web-based interactive query editor that enables you to interact with databases and data warehouses. Data architects, SQL developers, and data engineers use Hue to create data models, clean data to prepare it for analysis, and build and test SQL scripts for applications. Hue is integrated with Apache Hive and Apache Impala. You can access Hue from the Cloudera Data Warehouse (CDW) Virtual Warehouses.

CDW 1.1.2-b1520 offers the combined abilities of Data Analytics Studio (DAS) such as intelligent query recommendation, query optimization, and query debugging framework, and rich query editor experience of Hue, making Hue the next generation SQL assistant for Hive in CDW.

Hue offers powerful execution, debugging, and self-service capabilities to the following key Big Data personas:

- Business Analysts
- Data Engineers
- Data Scientists
- Power SQL users
- Database Administrators
- SQL Developers

Business Analysts (BA) are tasked with exploring and cleaning the data to make it more consumable by other stakeholders, such as the data scientists. With Hue, they can import data from various sources and in multiple formats, explore the data using File Browser and Table Browser, query the data using the smart query editor, and create dashboards. They can save the queries, view old queries, schedule long-running queries, and share them with other stakeholders in the organization. They can also use Cloudera Data Visualization (Viz App) to get data insights, generate dashboards, and help make business decisions.

Data Engineers design data sets in the form of tables for wider consumption and for exploring data, as well as scheduling regular workloads. They can use Hue to test various Data Engineering (DE) pipeline steps and help develop DE pipelines.

Data scientists predominantly create models and algorithms to identify trends and patterns. They then analyze and interpret the data to discover solutions and predict opportunities. Hue provides quick access to structured data sets and a seamless interface to compose queries, search databases, tables, and columns, and execute query faster by leveraging Tez and LLAP. They can run ad hoc queries and start the analysis of data as pre-work for designing various machine learning models.

Power SQL users are advanced SQL experts tasked with analyzing and fine-tuning queries to improve query throughput and performance. They often strive to meet the TPC decision support (TPC-DS) benchmark. Hue enables them to run complex queries and provides intelligent

recommendations to optimize the query performance. They can further fine-tune the query parameters by comparing two queries, viewing the explain plan, analyzing the Directed Acyclic Graph (DAG) details, and using the query configuration details. They can also create and analyze materialized views.

The **Database Administrators** (DBA) provide support to the data scientists and the power SQL users by helping them to debug long-running queries. They can download the query logs and diagnostic bundles, export and process log events to a database, or download it in JSON file format for further analysis. They monitor all the queries running in the system and terminate long-running queries to free up resources. Additionally, they may also provision databases, manage users, monitor and maintain the database and schemas for the Hue service, and provide solutions to scale up the capacity or migrate to other databases, such as PostgreSQL, Oracle, MySQL, and so on.
All Hue users can download logs and share them with their DBA or Cloudera Support for debugging and troubleshooting purposes.

SQL Developers can use Hue to create data sets to generate reports and dashboards that are often consumed by other Business Intelligence (BI) tools, such as Cloudera Data Visualization.

Hue can sparsely be used as a Search Dashboard tool, usually to prototype a custom search application for production environments.

Accessing Hue from the CDW UI

To query Hive/LLAP Virtual Warehouses using Hue, you must enable this feature in your CDP environment. Contact your Cloudera account team to have a preview feature enabled in your CDP account.

Note: If you are on a CDW version 1.1.2-b1347 or lower, then you must create a new Virtual Warehouse to use the new Hue SQL editor with DAS features. Upgrading the existing Virtual Warehouses (versions 7.2.10.0-36 or lower) may not work.

Steps

1. Log in to the CDP web interface and navigate to the Data Warehouse service.
2. In the Data Warehouse service, navigate to the **Overview** page.
3. On the **Overview** page under Virtual Warehouses, click the **options** menu in the upper right corner of a Hive Virtual Warehouse tile, and select **Open Hue**.
   The Hive query editor is displayed.
   You can view databases and query tables, and enable the S3 or ABFS File Browser to import files and create tables.

New features in Hue

Hue now offers additional query insights to optimize query performance and troubleshoot Hive queries. You can view query details and Directed Acyclic Graph (DAG) information from the Job Browser. You can compare two queries, terminate long-running or unintentionally triggered queries, view query history submitted from Hue as well as other JDBC or ODBC clients, and download debug bundles.

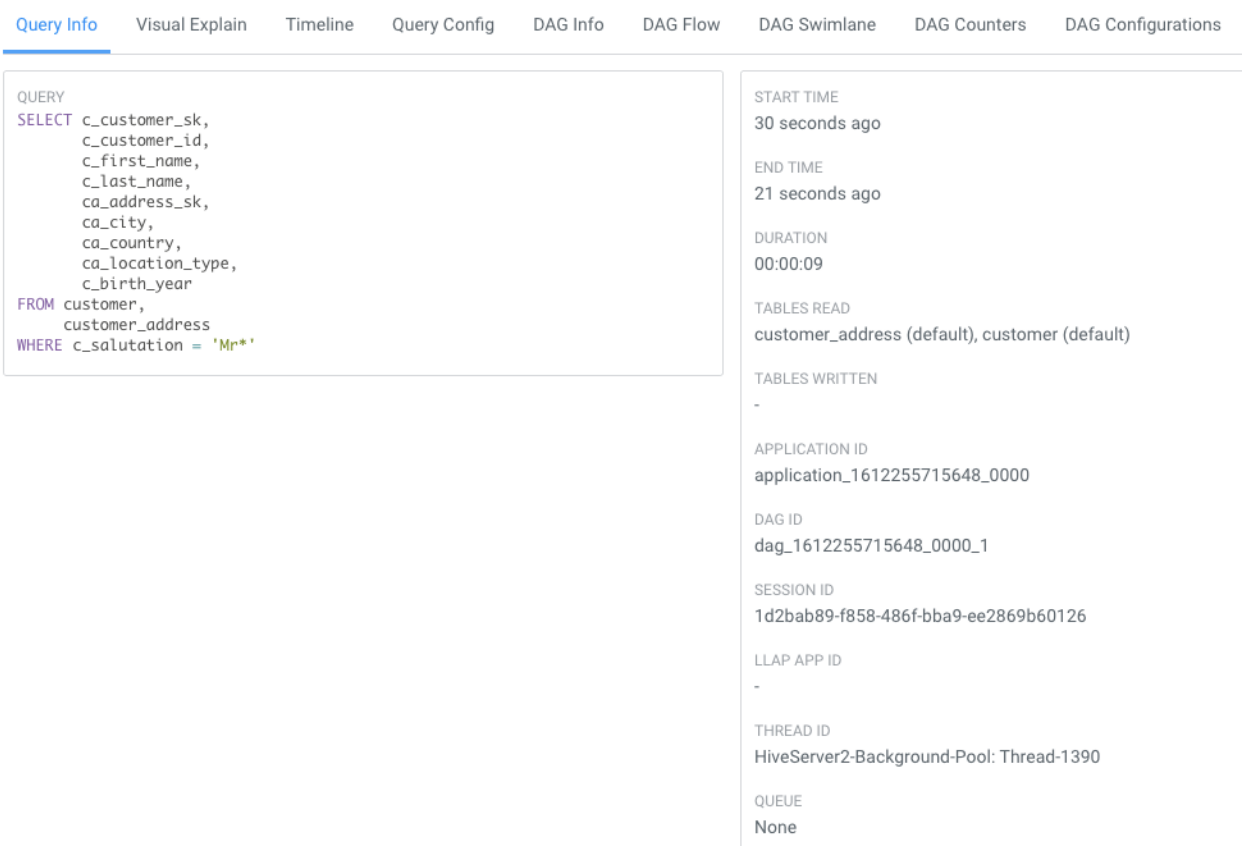
Viewing the query details

You can view the following details for Hive queries from the Hue Job Browser:

- Query information
- Visual explain
- Query timeline
- Query configuration
- DAG information
- DAG flow
- DAG swimlane
- DAG counters
- DAG configurations

Viewing the query information

The **Query Info** tab provides information such as, the Hive query ID, the user who executed the query, the start time, the end time, the total time taken to execute the query, the tables that were read and written, application ID, DAG IDs, session ID, LLAP app ID, thread ID, and the queue against which the query was run.

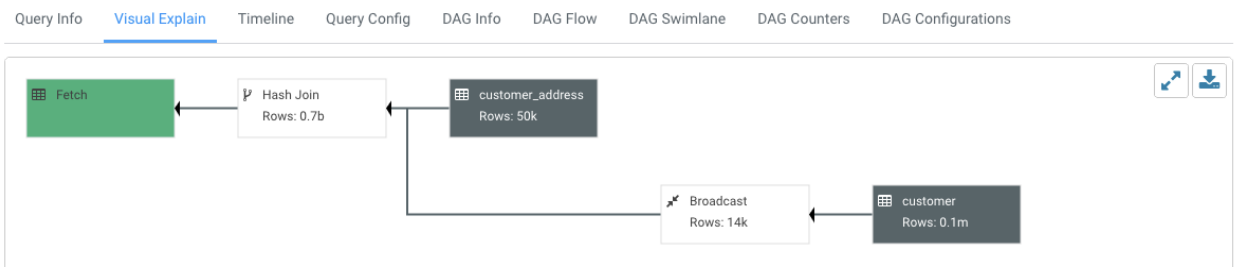


Viewing the Explain plan for a query

The Visual Explain feature provides a graphical representation of the query execution plan. The Explain plan is read from right to left. It provides details about every stage of query execution.

To view the Explain plan for a query:

1. Open the Hue web interface.
2. Go to the **Job Browser** by clicking **Jobs** on the left assist panel and click **Queries**.
3. Select the query for which you want to view the Explain plan and click on the **Visual Explain** tab.

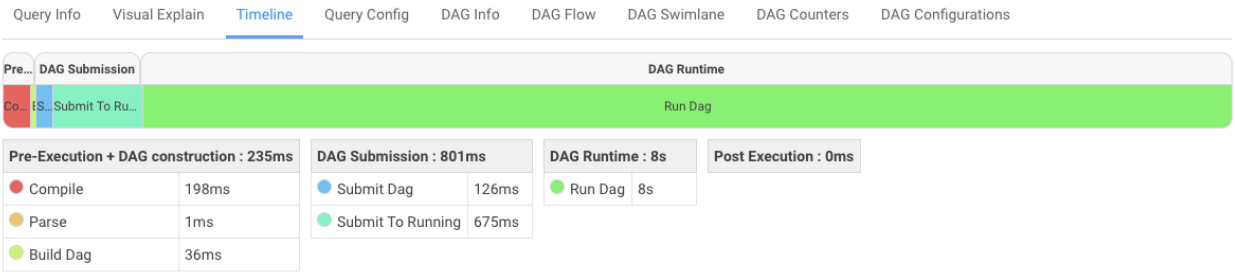


You can also download the query Explain plan in JSON format by clicking the download icon.

Viewing the query timeline

The **Timeline** tab provides a visual representation of Hive performance logs and shows the time taken by each stage of the query execution.

The the following snippet shows the pre-execution, runtime, and post-execution phases of a query:



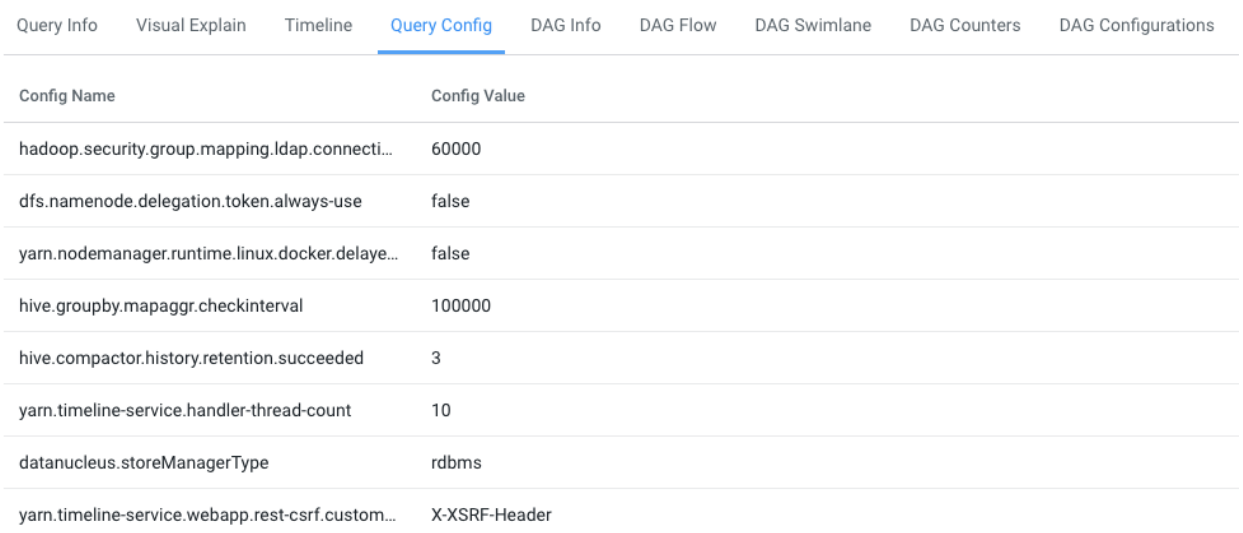
- Pre-execution and DAG construction: It is the first phase of query execution and is executed on the Hive engine. It constitutes the time taken to compile, parse, and build the DAG for the next phase of the query execution.
- DAG submission: It is the second phase in which the DAG that was generated in Hive is submitted to the Tez engine for execution.
- DAG runtime: It shows the time taken by the Tez engine to execute the DAG.
- Post-execution: It is the last phase of query execution in which the files in S3/ABFS are moved or renamed.

Duration data about each phase are distilled into more granular metrics based on query execution logs.

Viewing the Hive configurations for a query

The **Query Config** tab provides the configuration properties and settings that are used in the Hive query. You can use this tab to verify that configuration property values align with your expectations.

Following is a snippet of the query configuration:



Viewing DAG information

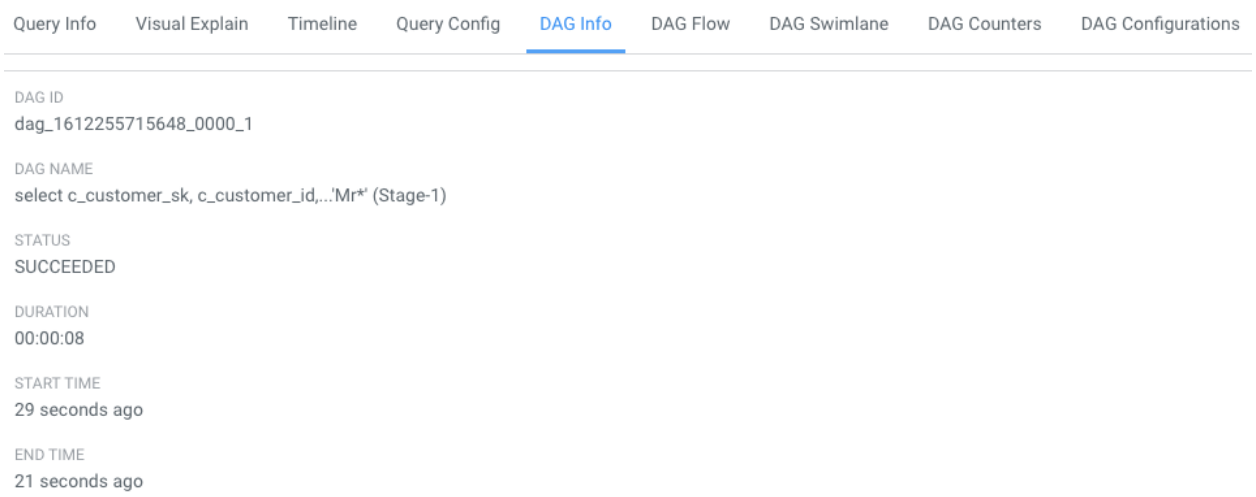
DAG is created by the Hive engine every time you query the Hive Virtual Warehouse. The Hive SQL queries are compiled and converted into a Tez execution graph also known as a DAG. DAG is a collection of vertices where each vertex executes a fragment of the query or script.

Directed connections between vertices determine the order in which they are executed. For example, the vertex to read a table must be run before a filter can be applied to the rows of that table. As another example, consider a vertex that reads a user table that is very large and distributed across multiple computers and multiple racks. Reading the table is achieved by running many tasks in parallel.

Hue provides a web interface to view detailed information about DAGs.

Note: The DAG information tabs (DAG Info, DAG Flow, DAG Swimlane, DAG Counters, DAG Configurations) are displayed only if the Tez engine is used for query execution. The Tez engine is typically utilized for complex queries.

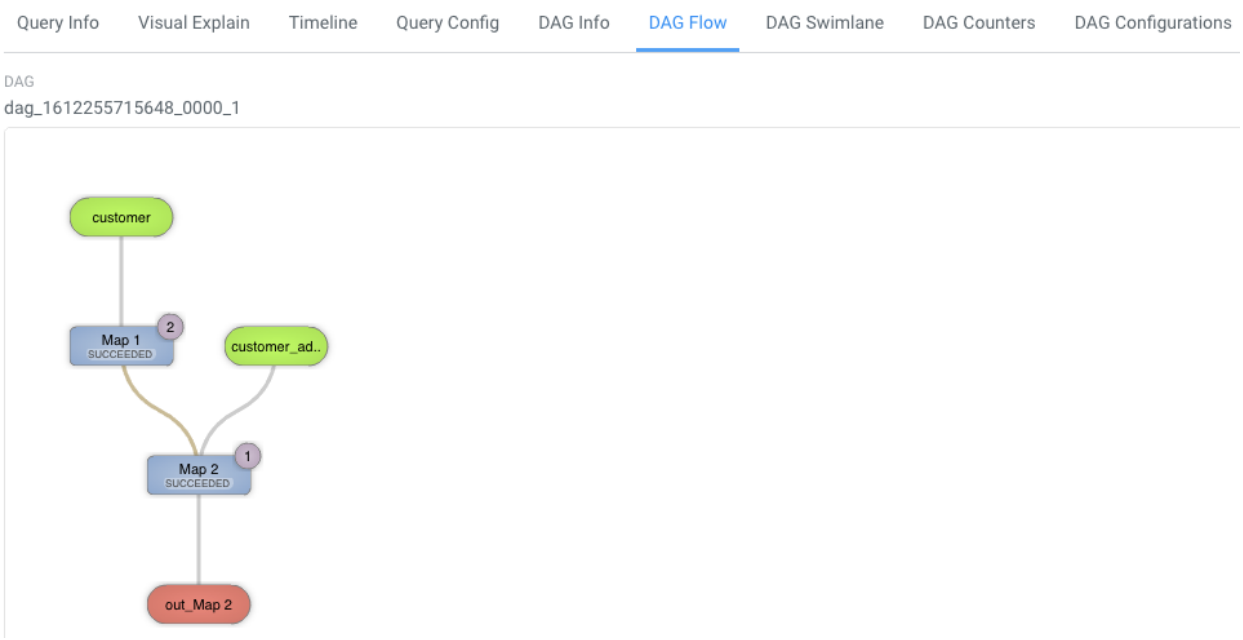
The **DAG Info** tab shows the DAG ID, DAG name, the status of the query, the time taken to execute the DAG, start time, and end time.



The following table lists and describes the status of the Tez job:

Status	Description
Submitted	The DAG is submitted to Tez but is not running.
Running	The DAG is currently running.
Succeeded	The DAG was completed successfully.
Failed	The DAG failed to complete successfully.
Killed	The DAG was stopped manually.
Error	An internal error occurred when executing the DAG.

The **DAG Flow** tab displays the DAG in the form of a flowchart. It enables you to gain insight into the complexity and the progress of executing jobs, and investigate the vertices that have failures or are taking a long time to complete.

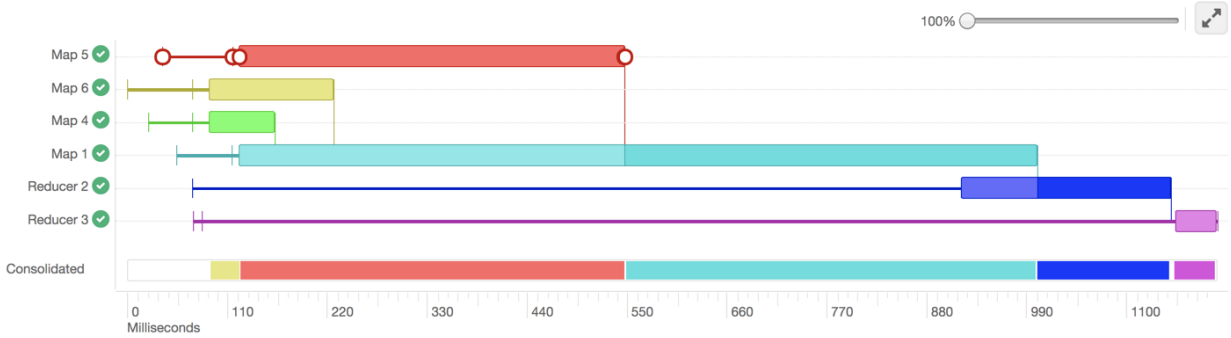


Here, the input to vertices Map 1 and Map 2 are the tables displayed in green boxes. Next, Map 2 depends on the result set generated by Map 1. Map 2 is the last vertex in the DAG flow and after it completes its execution, the query output is written to a file in a filesystem such as S3 or ABFS.

There are a few options to change the layout of the DAG flow. You can hide the input and the output nodes to view only the task vertices by clicking the **Toggle source/sink visibility** icon. You can switch between the horizontal and vertical orientation by clicking the **Toggle orientation** icon.

The **DAG Swimlane** tab displays the DAG of the vertices against time. Each mapping and reducing task is a vertex.

The following image shows the DAG swimlane:

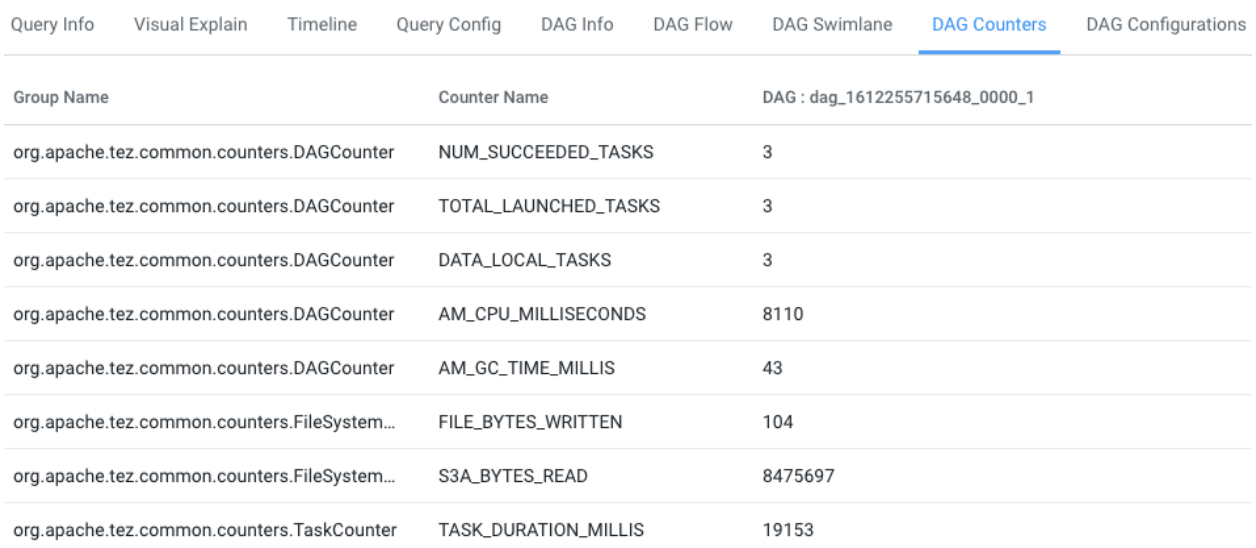


Each horizontal bar of the swimlane represents the total time taken by the vertex to complete the execution. The vertical lines indicate the time when the vertex was initialized, the time when the vertex started, the time when the first task started, the time when the last task was completed, and the time when the vertex finished its execution. When you mouse over the vertical line, the bubble displays the stage of the vertex execution and provides a timestamp. The vertical lines connecting two vertices denote the dependency of a vertex on another vertex.

In the above example, Map 1 depends on the results of Map 5. Map 1 will finish its execution only when Map 5 finishes its execution successfully. Similarly, Reducer 2 depends on Map 1 to complete its execution.

The consolidated timeline shows the percentage of time each vertex took to complete executing.

The **DAG Counters** tab provides a way to measure the progress or the number of operations that occur within a generated DAG. Counters are used to gather statistics for quality control purposes or problem diagnosis.

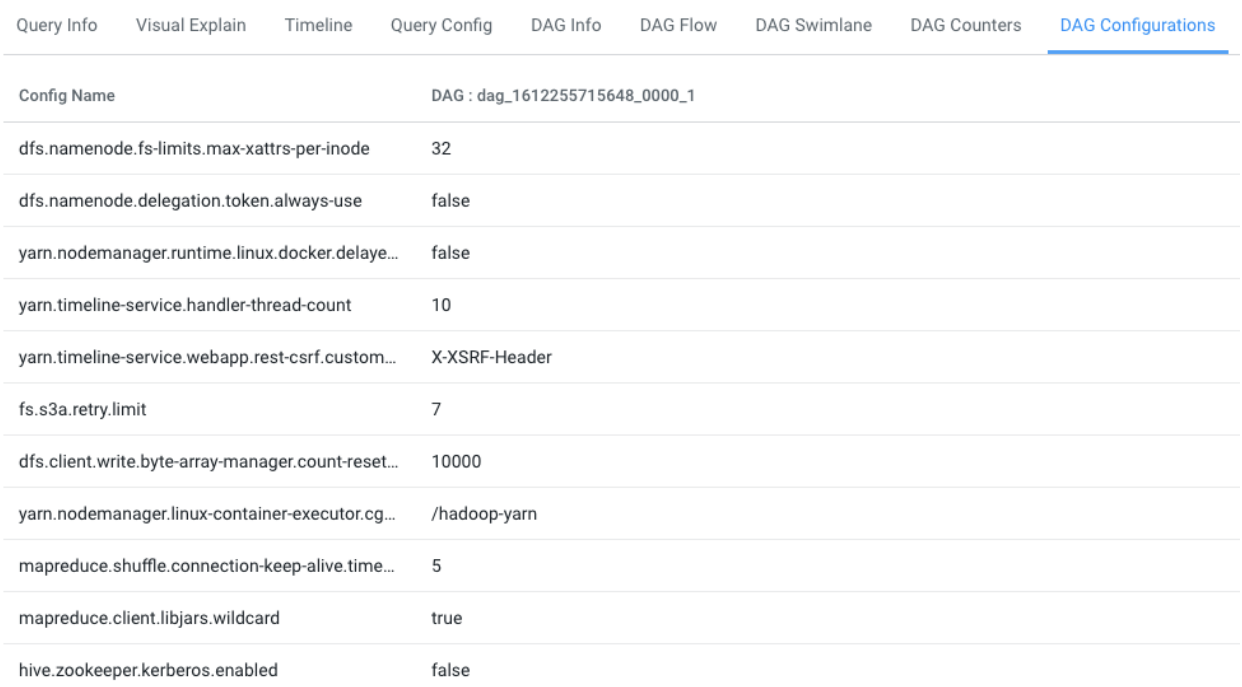


Group Name	Counter Name	DAG : dag_1612255715648_0000_1
org.apache.tez.common.counters.DAGCounter	NUM_SUCCEEDED_TASKS	3
org.apache.tez.common.counters.DAGCounter	TOTAL_LAUNCHED_TASKS	3
org.apache.tez.common.counters.DAGCounter	DATA_LOCAL_TASKS	3
org.apache.tez.common.counters.DAGCounter	AM_CPU_MILLISECONDS	8110
org.apache.tez.common.counters.DAGCounter	AM_GC_TIME_MILLIS	43
org.apache.tez.common.counters.FileSystem...	FILE_BYTES_WRITTEN	104
org.apache.tez.common.counters.FileSystem...	S3A_BYTES_READ	8475697
org.apache.tez.common.counters.TaskCounter	TASK_DURATION_MILLIS	19153

The DAG counters provide details, such as:

- Number of bytes read and written
- Number of tasks that initiated and ran successfully
- Amount of CPU and memory consumed

The **DAG Configurations** tab provides the Tez configuration details for the query that has a DAG associated with it.



Comparing queries

You can compare two queries to know how each query is performing in terms of speed and cost-effectiveness. Hue compares various aspects of the two queries, based on which you can identify what changed between the executions of those two queries, and you can debug performance-related issues between different runs of the same query.

The query comparison report provides you a detailed side-by-side comparison of your queries, including recommendations for each query, metadata about the queries, visual explain for each query, query configuration, the time taken at each stage of the query execution, and DAG information and DAG counters.

To compare two queries:

1. Open the Hue web interface.
2. Go to the **Job Browser** by clicking **Jobs** on the left assist panel and click **Queries**.
3. Select the queries that you want to compare and click **Compare**.
   The query comparison report is displayed.

Terminating Hive queries

If a Hive query is running for longer than expected, or you have accidentally triggered it, then you can stop the query to free up the resources. Hue also allows you to stop multiple queries at once.

You can configure Ranger policies to grant permissions to the users or groups to stop queries.

Only admin users or Hue superusers can stop running queries.

To stop Hive queries:

1. Open the Hue web interface.
2. Go to the **Job Browser** by clicking **Jobs** on the left assist panel and click **Queries**.
3. Select the queries that you want to stop and click **Kill**.

Viewing query history

Only Query Store Administrators can view historical queries of all users to monitor resource utilization and control costs from the Hue **Job Browser**. Non-admin users can view only their queries.

The **Job Browser** displays all the queries that were run on the Hive Virtual Warehouse from various query interfaces, such as Beeline, Hive Warehouse Connector (HWC), Tableau, and so on.

The Hue Query Store does not clean up automatically. The queries are retained in the backend database until you clean up the records. You can view queries up to five years old from the Hue web interface.

To view historical queries:

1. Open the Hue web interface.
2. Go to the **Job Browser** by clicking **Jobs** on the left assist panel and click **Queries**. The Hive queries that were run by all users for the past seven days are displayed. You can select the time period for which you want to view the historical data.

Adding the Query Store Administrator users

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

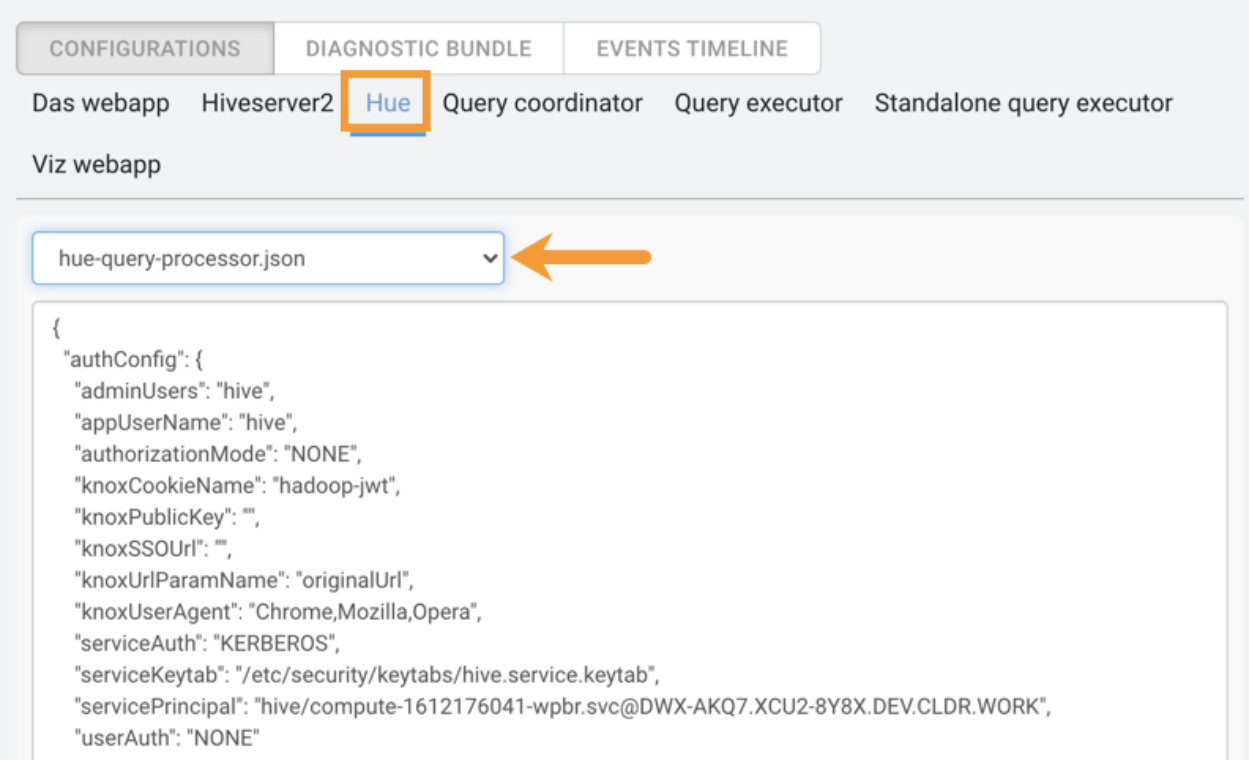
Before you begin

Make sure that the Virtual Warehouse to which you want to add the Hue Query Store Administrators users is in the running state. If it is in a stopped state, then start it, and wait until it is in the running state.

To add Query Store Administrator users:

1. Log in to the CDP web interface as DWAdmin and navigate to the Data Warehouse service.
2. In the Data Warehouse service UI **Overview** page, select the Hive Virtual Warehouse to which you want to add Hue Query Store Administrators users, and click its edit icon.

3. On the Virtual Warehouses detail page, click the **Hue** tab and from the configuration file drop-down list, select **hue-query-processor.json**:

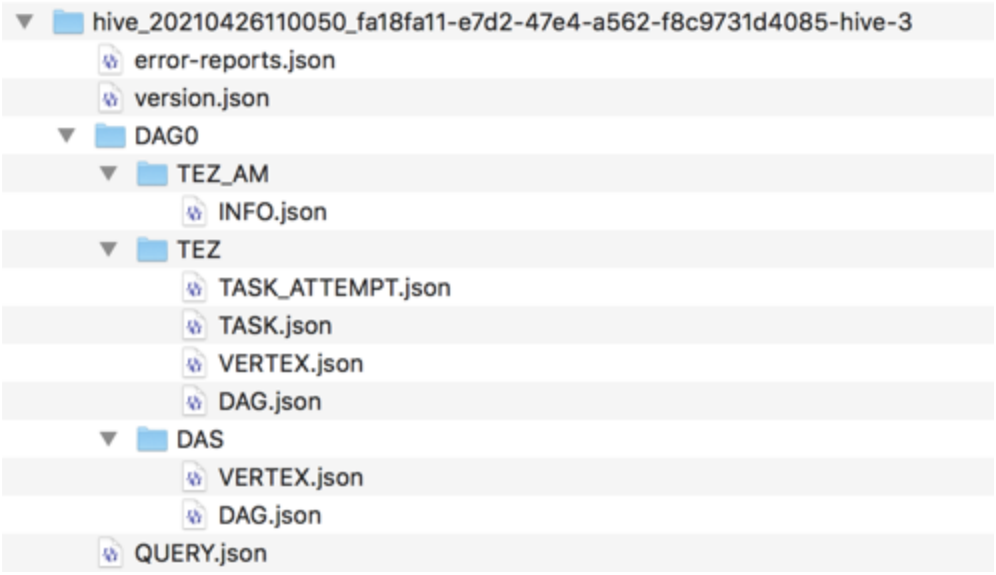


4. In the “authConfig” section, add the list of users to the “adminUsers” key. For example: `"adminUsers": "hive, [***USER-1***], [***USER-2***]"`
5. After making your changes, click **Apply** in the upper right corner of the page. The Hue service will be unavailable for approximately 5 minutes to make the update.

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.

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:



To download the debug bundle:

1. Open the Hue web interface.
2. Go to the **Job Browser** by clicking **Jobs** on the left assist panel and click **Queries**.
   The Hive queries that were run are displayed.
3. Select a query for which you want to download the debug bundle.
4. 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***]`