Adding SQL-On-Hadoop Engines

If you are using Hadoop as a data warehouse, you must configure one or more SQL-on-Hadoop engines for running queries.

Before You Begin

- Ensure that your user ID in the Design Center is assigned the Super User role or is assigned the Manage Data Warehouses role permission.
- Ensure that you know the AtScale username for authentication with the SQL-on-Hadoop engine. If using Kerberos authentication, this would be the AtScale principal name instead.
- If you are using LDAP authentication to the Hive metastore, ensure that you know the LDAP password for the username for connections to the SQL-on-Hadoop engine.
- Ensure that you know which extra JDBC flags, if any, you need to use for connections to the Hive metastore. Because all connections use the HiveServer2 protocol, all JDBC flags are the same as those specified for that protocol.For example, if you are not using any authentication, you would specify this JDBC flag:

  ```
  ;auth=noSasl
  ```

- Ensure that you know the JDBC port to use for connections. The default ports are as follow:

  HiveServer2
  10000

  Impala
  21050

- Ensure that you know the name of the host where the master process of the SQL-on-Hadoop engine resides. If you have a clustered configuration, you can list the primary and the backup hosts separated by commas.
- If you would like error messages to include a link to the SQL-on-Hadoop engine's management console, ensure that you know the URL to the console.

About This Task

There are four query-processing roles that you can assign to SQL-on-Hadoop engines. An engine can assume more than one of these roles, though it might be better suited for a specific role.

Large Interactive Queries
These are large, expensive queries issued to AtScale from a client application. These are usually queries that cannot be optimized by AtScale aggregates or ones that involve high-cardinality dimensions. Hive or Impala are good choices for large interactive queries.

Small Interactive Queries

These are small, inexpensive queries issued to AtScale from a client application. These are usually queries that can be optimized by AtScale aggregates or ones that involve low-cardinality dimensions. Spark or Impala are good choices for small interactive queries.

System Queries

These are the queries generated by the AtScale engine to build aggregate tables or query system statistics. Hive or Impala is a good choice for system queries.

Canary Queries

These are queries that are generated by AtScale to find out whether a user issuing a query that hits an aggregate table has permission to view the data in that table. Permissions that are defined in security dimensions are not copied to aggregate tables.

For example, suppose you have a fact dataset that contains information about internet sales and a dimension for customer gender. You create a security dimension that restricts the users who can view data by gender. User A is given this permission.

Next, you define an aggregate on order_quantity by gender. User A runs a query that will use that aggregate. The AtScale engine must check whether user A has permission to view the data in that aggregate, even though user A was given access to gender data by the security dimension. The AtScale engine creates a canary query that the engine passes the SQL-on-Hadoop engine that is assigned the Canary Queries role in the data warehouse being used.

> **Note:** The data warehouse must be associated with at least one Hive or Impala SQL-on-Hadoop engine that supports canary queries.

The canary query returns no data from the aggregate. This query is only a test of the permission of the user issuing the original query.

If the result of the canary query confirms that user A has permission to view the data in the aggregate, the AtScale engine issues the user's query. If the result denies that user A has permission, the user's query is not processed.

You cannot assign the Canary Queries role unless Impersonation is enabled from the Hadoop Data Warehouse set up page.

Either feature allows the AtScale engine to impersonate a user when issuing a canary query. In the example above, the engine impersonates user A when issuing the canary query.

If the SQL-on-Hadoop engine that you added is the first in the data warehouse, the engine is assigned all of the query-processing roles.

If the SQL-on-Hadoop engine that you added is the second or a subsequent engine, you must use assign the engine one or more roles before it will be used for query processing.

Procedure

To add an SQL-on-Hadoop engine:

1. Choose Settings from the top navigation menu, select **Data Warehouses**.
2. Under **Data Warehouses**, expand the data warehouse that you want to add an SQL-on-Hadoop engine to.
3. Click **Create Connection**.
4. Enter a unique name for the connection.
5. Select the **Server Type** from the available selections.
6. Enter the **Hostname** of the hosts which have this SQL engine installed. Multiple hosts can be separated by commas. Ranges are supported such as 192.168.1.[1-10], or 192.168.1.[1, 3, 5].
7. Enter the **Port** number.
8. Specify any **Extra JDBC** flags.
9. Enter an **Engine Management Console URL**.
10. Select the **Authorization** type. If Username and Password authentication is used, select **Password**. Select **Kerberos** if a Kerberos principal is in use.
11. Test the connection.
12. Click **Save** to complete the setup.

What To Do Next

After you click Save, the Design Center returns to the overview page for the current data warehouse. Use this page to assign roles to your SQL-on-Hadoop engines.

Related reference Adding Data Warehouses