Adding Databricks Data Warehouses

A Databricks data warehouse contains the tables and views that you want to import as fact datasets and dimension datasets into AtScale cubes. It also contains aggregate-table instances that either you or the AtScale engine creates in a database that you specify.

> 💡 To configure a Databricks data warehouse, you must have a special AtScale license. Ask your sales representative about obtaining a Databricks-enabled license.

Before You Begin

For AtScale:

- ▲ Use admin role.

  Ensure that your user account is assigned to a role that allows adding and managing data warehouses. For details, see Assigning Individual Users to Roles.

For Databricks:

- ▲ Use a supported Databricks version

  Make sure you are using a version of Databricks that is supported by AtScale Enterprise Cloud. For details, see Supported Tools and Platforms.

- ▲ Configure supported features

  When using Databricks SQL or Unity Catalog, make sure they are configured as needed. For details, see Databricks SQL guide and What is Unity Catalog.

- ▲ Generate a personal access token

  Since Databricks does not support JDBC authentication with username and password, you have to generate a personal access token. Databricks recommends using access tokens belonging to service principals; for more information about obtaining them, see Manage service principals. When creating the Databricks connection (described below), use the token as the password, and `token` as the username.

> 📄 Alternatively, you can use Databricks Partner Connect. This way Databricks provisions the required resources like a service principal, a personal access token, a SQL warehouse on your behalf. For more information, see Databricks' Atscale Partner Connect documentation in AWS or Azure.

Add The Data Warehouse

1. Log in to AtScale.
2. Go to **Settings > Data Warehouses** and click **Create Data Warehouse**.
3. In the Add a Data Warehouse dialog box, select one of the following for **Type of Data Warehouse**:

   - Databricks SQL when using the Photon Runtime with a Databricks SQL warehouse.

4. Enter a unique name for the data warehouse. AtScale displays this name in the Design Center and uses it in log files.
5. Enter the **External Connection ID**.

   This input will be inherited by the **Name** value, but can be overridden by enabling the **Override generated value** slider.

1. Enter the name of the Catalog and Schema for AtScale to use when creating aggregate tables. Consider the following:

   - In case the catalog name contains non-alphabet characters, you should surround the name with backtick (`) characters, like this: `catalog-name`
   - AtScale supports Databricks SQL clusters with both Unity Catalog enabled and disabled, and thus three-part namespace support is enabled for Databricks SQL.
   - With Unity Catalog disabled, aggregate schema will reside in default catalog hive_metastore. And with Unity Catalog enabled, it's the user choice to decide in which Catalog-->Schema the aggregates can be stored.

2. Specify the Data Warehouse as a **Read-only source**.

   - Select this option if AtScale will be configured to access this database with credentials that do not have permission to create new tables, or if you need to prevent AtScale from placing any aggregate tables in this data warehouse.
   - If **Read-only source** is enabled upon editing an existing data warehouse with aggregates present, all existing aggregates stored in this data warehouse will be deactivated.

3. Enter the name of the **Aggregate Schema**.

   AtScale reads and writes aggregate data to this database. The AtScale service account user must have ALL privileges for this database. BI tool user accounts should not have the select permission for this database.

1. In the Table storage formats section, specify the storage format for new tables (if you have a preference).

   Select default to let the AtScale engine use the format configured in the `aggregates.tableConfig.preferredStorageFormat` engine setting. For more information, see Other Settings for Both System-Defined and User-Defined Aggregates.

2. If you need to use objects in a file system:

    1. Go to the **Configure Filesystem** section and choose the **Enable Filesystem** option.
    2. Choose the file system type.
    3. Enter the URI and other required details.

3. Click **Save** to complete the setup.

Add A Databricks Connection

After setting up your data warehouse, you must add a connection to the data warehouse before you can run queries. You can establish multiple connections with different configurations to the same data warehouse. If multiple connections are configured, you can assign which types of queries are permitted per connection; for details, see the Query Roles section below.

1. Expand the drop-down adjacent to the listed data warehouse and click **Create Connection** (or **Add Databricks Connection** if you have other existing connections).
2. Enter the **Name** of your Databricks connection.
3. Enter the **Host** location of the Databricks server.
4. For **Port**, enter the port number to route transmitted data. The default port number is 443.
5. Enter the **Username** and **Password** for connecting to the database.

    If you have external secret manager enabled and configured, enter the corresponding credentials.

6. In the **Extra JDBC Flags** field, enter the JDBC flags that are needed for the connection:

    - Access your Databricks cluster to retrieve your JDBC information.
    - Make sure all of the following flags are set: `transportMode` , `ssl` , `httpPath` , `AuthMech` , `UID` , `PWD` .
    - Optionally, set the `UseNativeQuery=1` flag.
    - Here is an example for setting the flags: `transportMode=http;ssl=1;httpPath=<path>;AuthMech=3;UID=token;PWD=<tokenID>;UseNativeQuery=1` .

7. Test the connection.
8. Click **Save** to complete the setup.

Query Roles

If multiple Databricks connections are configured, you can toggle which types of queries (Large Interactive, Small Interactive or System) are permitted per Databricks connection. A query role can be assigned to only one connection at a time.

Expand the drop-down for the Databricks data warehouse to choose the allowed query roles for each connection:

- **Large Interactive Queries**: Use this engine for large user queries. These are usually BI queries involving full scans of the raw fact table or ones that involve high-cardinality dimensions.

- **Small Interactive Queries**: Use this engine for small user queries. These are usually BI queries that can be optimized by AtScale aggregates or ones that involve low-cardinality dimensions.

- **System Queries**: Use this engine for AtScale system processing, such as building aggregate tables or querying system statistics. The System query role user must not be subject to resource limitation rules that would prevent it from building aggregate tables.