Adding Google BigQuery Data Warehouses

A Google BigQuery data warehouse is an instance of Google BigQuery that contains the tables and views that you want to access as cube facts and dimensions. It also contains aggregate-table instances that either you or the AtScale engine creates in a BigQuery dataset that you specify.

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.
- Aggregate Dataset: Ensure that you know the BigQuery dataset and ProjectID to use for aggregate tables to be built in the data warehouse.

  AtScale reads and writes aggregate data to this dataset. The AtScale service account user must have ALL privileges for this dataset.

- Assign your Google BigQuery service account these roles: `BigQuery Data Editor` and `BigQuery Job User`.

  The service account used will be the one used for setting up system query role. If Aggregates are being stored in a project different from the project where Fact data resides, then this service account needs BigQuery Data Viewer and BigQuery Job User for all the projects that contain fact data

- The Google BigQuery service account used for setting up large/small query roles needs to have read access on all the projects with Fact data, plus on the project where aggregate dataset resides. At the minimum, the BigQuery Data Viewer and BigQuery Job User roles.
- One of the options when you add Google BigQuery as a data warehouse is to specify a Google Cloud Bucket. You can set up AtScale to rebuild aggregate tables whenever a file is added to the bucket, as explained in Triggering Aggregate Rebuilds. Ensure that the bucket you use is multi-regional.
- Consider that Google BigQuery Views cannot be previewed in Design Center.

Authentication

You can choose to authenticate with a JSON credential file, or with Application Default Credentials (ADC)/Environment-provided service account. For additional details about Google authentication types, see Google Cloud Authentication. In both cases you need to perform some steps in advance.

When using the JSON credential file for your Google Cloud Platform service account:

- Ensure that the file can be accessed by AtScale.
- The file for the service accounts can be created from the Google Cloud Platform console from the Menu > IAM & admin > Service accounts. The file contains seven keys and their associated values, as in this example:

```
{ "name": "projects/my-project-123/serviceAccounts/my-sa-123@my-project-123.iam.gserviceaccount.com",
"projectId": "my-project-123", "uniqueId": "113948692397867021414", "email": "my-sa-123@my-project-
123.iam.gserviceaccount.com", "displayName": "my service account", "etag": "BwUp3rVlzes=", "oauth2ClientId":
"117249000288840666939" }
```

The authentication that is based on Google Compute Node service account credentials can be used with the following setup:

- A user-managed service account is attached to the Google Compute Node that runs AtScale.
- The service account has access to project(s) or impersonates another service account that has access in Google BigQuery.
- If the service account has access to multiple Google BigQuery projects, then multiple AtScale data warehouses can be set up corresponding to the needed projects.

You can use the following commands to check which projects the compute node service account has access to:

- `gcloud config list`
- `gcloud projects list`

ADC-Based Authentication From On-Premise Node

The `GOOGLE_APPLICATION_CREDENTIALS` environment variable is used for providing the location of a credential JSON. You can use it in AtScale in the following way:

1. Update the ~/.bash_profile file of the `atscale` user, and set the environment variable `GOOGLE_APPLICATION_CREDENTIALS` to point to the location of the service account key or credential file:

```
cat ~/.bash_profile | grep GOOGLE
export GOOGLE_APPLICATION_CREDENTIALS=/home/atscale/SAcctCreds.json
```

2. Reload `bash_profile` using the `source` command, or start a new terminal.

```
source ~/.bash_profile
```

3. Restart all AtScale services; this needed to pick up the environment variable:

```
/opt/atscale/bin/atscale_stop
/opt/atscale/bin/atscale_start
```

4. Check if the environment variable is picked up:

   ```
   cat /proc/`/opt/atscale/current/pkg/jdk/bin/jcmd -l | grep engine | awk '{print $1}'`/environ | tr '\0' '\n' | grep GOOGLE
   ```

   You should get a result like this:

   ```
   GOOGLE_APPLICATION_CREDENTIALS=/home/atscale/SAcctCreds.json
   ```

Procedure

To add a new Google BigQuery data warehouse:

1. Choose Settings from the top navigation menu, select **Data Warehouses**.
2. Click on **CREATE DATA WAREHOUSE**.
3. Under **Type of Data Warehouse**, select **BigQuery**.
4. In the **Authentication** section, set one of the following sources for the Service Account:
   - Use service account key JSON file: when set, you would need to upload the JSON credential file as described below.
   - Use Application Default Credentials (ADC)
   - Use Application Default Credentials (ADC) impersonated with Service Account
5. Enter a unique name for the data warehouse.
6. (Optional) Enter the **External Connection ID** (formerly Query Name).

   The External Connection ID defaults to the data warehouse name. Override by entering a different name and enabling **Override generated value**.

7. In the Aggregate Project ID field, enter name of the ProjectID in which the Aggregate Schema resides.

   The service account used for setting up system query role will be the one that creates aggregates and needs have All privileges on the project.

8. In the Aggregate Schema field, enter the name of the BigQuery dataset to use when creating aggregate tables. (You must create a dataset; AtScale does not create a dataset if one does not exist.)
9. 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.

10. If desired, enable **Impersonation** to allow AtScale to communicate with the data platform using the BI tool end-user user credentials. This allows the data warehouse administrator to apply more detailed security policies when securing data.
11. If Impersonation is enabled, toggle the settings as needed:

    - **Always Use Canary Queries**: Whether canary queries are required in light of aggregate misses.
    - **Allow Partial Aggregate Usage**: Allows mixed aggregate and raw data queries.

    > AtScale cannot import data-loader bundles while impersonation is enabled. It is suggested that you import your desired data-loader bundles, commonly used for training, before enabling impersonation or use a separate AtScale installation for training purposes.

12. You can optionally specify a connection to a Google Cloud Bucket:

    - You can set up AtScale to rebuild aggregate tables whenever a file is added to the bucket, as explained in Triggering Aggregate Rebuilds.
    - Turn the Connect a Google Cloud Bucket option on.
    - Enter the bucket name in the corresponding field; ensure that the bucket you use is multi-regional.
    - If you have chosen Use service account key JSON file for authentication, use the button in the **SERVICE ACCOUNT KEY JSON FILE** section to add the JSON file.
    - If you have chosen Use Application Default Credentials (ADC) impersonated with Service Account for authentication, enter the service principal.

13. Click **Test Connection** to test the data warehouse connection.

Add A GBQ Connection

After setting up your data warehouse, you must add a connection to the data warehouse before you can run queries. Expand the Google BigQuery Data Warehouse, and choose Create Connection.

1. Enter a unique name for the connection.
2. Only when authenticating with JSON credential file:

    1. Upload the file associated with your Google BigQuery Connection.
    2. Optionally, enable Project Id for query execution, and enter the BigQuery project in which queries are executed. It will be passed when creating a job. If not specified, uses the project associated with the service account key JSON.

3. Only when authenticating with ADC:

   Optionally, enable Project Id for query execution, and enter the BigQuery project in which queries are executed. It will be passed when creating a job. If not specified, uses the projectID associated with the service-account attached to the compute node, or the service-account referred by `GOOGLE_APPLICATION_CREDENTIALS` variable.

4. Only when authenticating with ADC impersonated with Service Account:

    1. Add the service principal.
    2. Enable Project Id for query execution, and enter the BigQuery project in which queries are executed. It will be passed when creating a job. If not specified, uses the projectID associated with the service-account attached to the compute node, or the service-account referred by `GOOGLE_APPLICATION_CREDENTIALS` variable.

5. Test the connection, and then save it.

Multiple connections can be set up, such that aggregate creation is handled by a different service account than the one that handles user query execution.