Integrating Snowflake with Authomize

Integration with Snowflake provides visibility about your Snowflake users, groups, roles, permissions, and access to databases and their tables. After integration, Authomize provides visibility into the database infrastructure, helping to close the gaps of who has access to what and with what permissions and increasing the security and compliance of the data layer IAM management.

Snowflake integration workflow

Integration requires the following steps:

1. Configure a Snowflake user.
2. Integrate Snowflake with Authomize.

Configure a Snowflake user

Prerequisites

Before you begin, ensure that you can log into Snowflake with ACCOUNTADMIN privileges.

1. Log in to Snowflake.
   Note the account ID for logging in. You will need it to complete the integration.
2. From the menu, click **Worksheets**.
3. From the Worksheets page, click + (at the top right corner of the page) and select **SQL Worksheet**.
   A worksheet is displayed where you can run SQL queries.
4. Set up an Authorized Role, using the following queries:

   ```
   USE ROLE ACCOUNTADMIN;

   -- Create a user with the least privilege to carry out the tasks
   CREATE OR REPLACE ROLE AUTHOMIZE_ROLE;

   CREATE OR REPLACE USER AUTHOMIZE_USER
   PASSWORD = '<your_password>';

   -- Note the default role will be used during scan
   ALTER USER AUTHOMIZE_USER SET DEFAULT_ROLE = AUTHOMIZE_ROLE;

   -- Add user to Authomize role
   GRANT ROLE AUTHOMIZE_ROLE TO USER AUTHOMIZE_USER;

   -- Activities are inside views of SNOWFLAKE database
   GRANT IMPORTED PRIVILEGES ON DATABASE SNOWFLAKE TO ROLE AUTHOMIZE_ROLE;
   ```

5. Run the queries (In line 7, replace YOUR_PASSWORD with a password of your choice.

6. Create a dedicated database and warehouse to allow Authomize to run a stored procedure, which will automatically update with new databases. Use the following queries:

   ```
   -- Database for creation of stored procedure
   CREATE OR REPLACE DATABASE AUTHOMIZE_DATABASE;

   -- Give access to database to ROLE_NAME
   GRANT USAGE, MONITOR ON DATABASE AUTHOMIZE_DATABASE TO ROLE
   AUTHOMIZE_ROLE;

   -- Create warehouse for running the stored procedure
   CREATE OR REPLACE WAREHOUSE AUTHOMIZE_WAREHOUSE WITH
       WAREHOUSE_SIZE = 'XSMALL'
       WAREHOUSE_TYPE = 'STANDARD'
       AUTO_SUSPEND = 300
       AUTO_RESUME = TRUE;
   ```

7. Create a new stored procedure that will scan all databases in Snowflake during each connector run. This procedure grants Authomize user usage permissions to all new databases. The stored procedure is created by the owner but is not modifiable, ensuring consistent actions.

   ```
   -- Create a stored procedure that'll grant usage privileges on all
   databases- Initialize to execute as owner
   CREATE OR REPLACE PROCEDURE
   AUTHOMIZE_DATABASE.public.grant_usage_on_all_dbs(role_name STRING)
       returns varchar not null
       language javascript
       execute as owner
       as
       $$
       // Function to execute a single grant statement
       function execute_Statement(query, dbName, ret) {
           ret += "\n- " + query;

           try {
               snowflake.execute( {sqlText: query} );
               ret += "\n\t-SUCCESS-";
           } catch (err) {
               ret += "\n\tQuery Failed for " + dbName;
               ret += "\n\tCode: " + err.code;
               ret += "\n\tState: " + err.state;
               ret += "\n\tMessage: " + err.message;
               ret += "\n\tStack Trace:\n" + err.stackTraceTxt;
           }

           return ret;
       }

       // we build up the return value string
       let ret = "USAGE access granted on: ";

       // Get all databases
   ```

```
    const res = snowflake.execute( {sqlText: "SHOW DATABASES;"} );

    // Iterate through each database row
    while (res.next())  {
        // Extract the database name
        const dbName = '"' + res.getColumnValue(2) + '"';

        // Add each DB processed to the return value
        ret += "\n\n\n- " + dbName;

        if (dbName === '"SNOWFLAKE"' || dbName === 
'"SNOWFLAKE_SAMPLE_DATA"') {
            ret += "\n\t-Imported privileges on snowflake db already 
added to the role-";
        } else {
            // Create grant usage queries on database and its 
schemas/tables
            const grantDbQuery = `GRANT USAGE ON DATABASE ` + dbName + 
` TO ROLE AUTHOMIZE_ROLE;`;
            const grantSchemaQuery = `GRANT USAGE ON ALL SCHEMAS IN 
DATABASE  ` + dbName + ` TO ROLE AUTHOMIZE_ROLE;`;
            const grantFutureSchemaQuery = `GRANT USAGE ON FUTURE 
SCHEMAS IN DATABASE ` + dbName + ` TO ROLE AUTHOMIZE_ROLE;`;
            const grantTableQuery = `GRANT SELECT ON ALL TABLES IN 
DATABASE  ` + dbName + ` TO ROLE AUTHOMIZE_ROLE;`;
            const grantFutureTableQuery = `GRANT USAGE ON FUTURE 
SCHEMAS IN DATABASE ` + dbName + ` TO ROLE AUTHOMIZE_ROLE;`;

            // Execute each grant query and add it to the return value
            ret = execute_Statement(grantDbQuery, dbName, ret);
            ret = execute_Statement(grantSchemaQuery, dbName, ret);
            ret = execute_Statement(grantFutureSchemaQuery, dbName, 
ret);
            ret = execute_Statement(grantTableQuery, dbName, ret);
            ret = execute_Statement(grantFutureTableQuery, dbName, 
ret);
        };

    };
```

8. Grant the Authomize role the permission to execute the stored procedure. Use the following queries:

```
-- Grant usage on procedure and usage/operate on warehouse to the 
authomize role
GRANT USAGE ON PROCEDURE 
AUTHOMIZE_DATABASE.public.grant_usage_on_all_dbs(STRING) TO ROLE 
AUTHOMIZE_ROLE;
GRANT USAGE, OPERATE ON WAREHOUSE AUTHOMIZE_WAREHOUSE TO ROLE 
AUTHOMIZE_ROLE;

-- Execute the stored procedure
USE WAREHOUSE AUTHOMIZE_WAREHOUSE;
CALL 
AUTHOMIZE_DATABASE.public.grant_usage_on_all_dbs('AUTHOMIZE_ROLE');
```

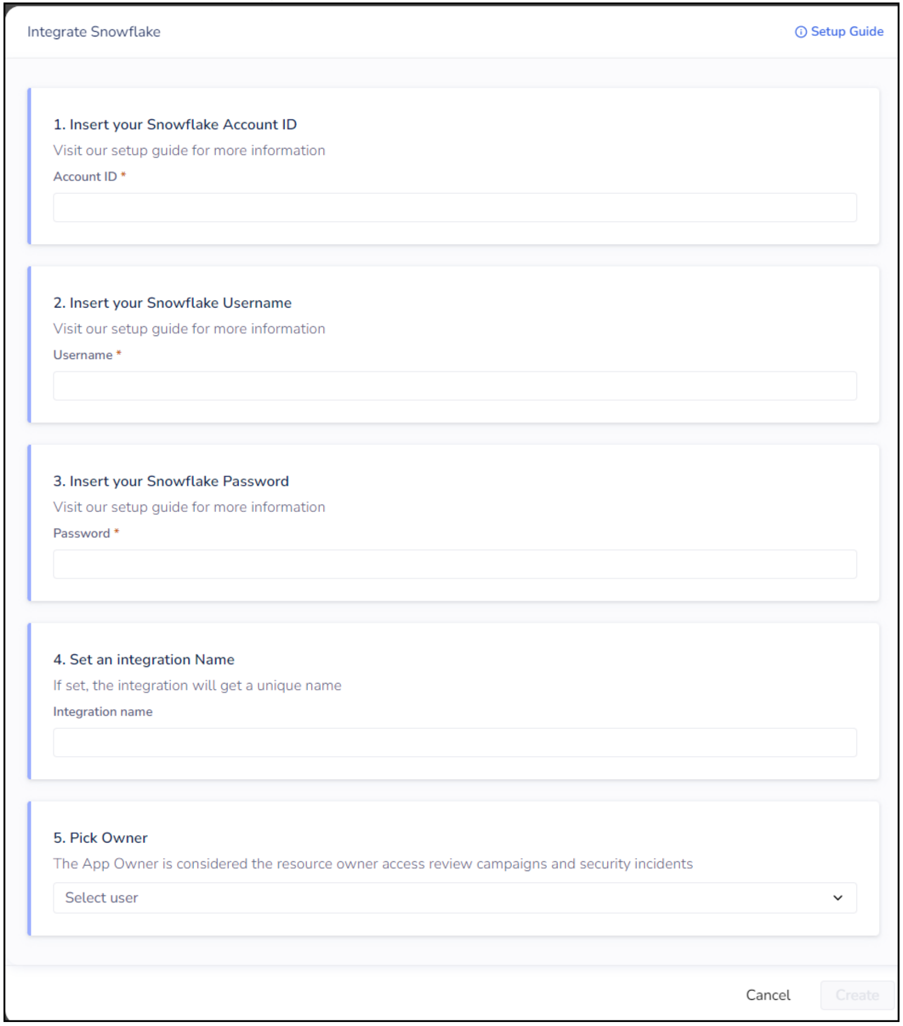
Integrate Snowflake with Authomize

You will use the Snowflake login authentication credentials to enable the integration.

Prerequisites

- You can access Authomize as a system administrator.
- You have the Snowflake account ID and the username and password created in the previous procedure.

1. From the Integrations > Data Sources page, click **Add New App**.
2. Click the **Snowflake** app.
   The Snowflake dialog is displayed:



Integrate Snowflake

1. Insert your Snowflake Account ID
Visit our setup guide for more information
Account ID *

2. Insert your Snowflake Username
Visit our setup guide for more information
Username *

3. Insert your Snowflake Password
Visit our setup guide for more information
Password *

4. Set an integration Name
If set, the integration will get a unique name
Integration name

5. Pick Owner
The App Owner is considered the resource owner access review campaigns and security incidents
Select user

3. Enter the Snowflake account ID, username, and password.
4. (Optional) You can enter a unique name for this integration.
   By default, the integration will be named Snowflake.
5. (Optional) Select a current Authomize user as the app owner.

6. Click **Create**.

The Snowflake tile is displayed as a connected app. The synchronization process begins, and its status will be shown when it is completed.

Data collected

The following tables show the data that is collected and how it is mapped in Authomize:

Users

Snowflake Users	**Authomize Accounts**
name	name
email	Email
First name	First name
last name	Last Name
Status	Status
comment	Description
Employee type	IsManaged (yes if internal)
MFA status	Is multifactor enabled (yes if enabled)
Last success login	Last login

User Roles:
Mapped to Authomize Type group, with origin type = Role
The name of the role is equal to Snowflake role's name

Database Roles:
Mapped to Authomize Type group, with origin type = Database Role
The name of the role is equal to Snowflake role's name

Assets

Snowflake Asset	How those are represented on Authomize assets				
	Name	Description	Type	Origin Type	createdAt

Database	name	comment	Database	Database	Created on (api field)
Schemas	name	comment	Other	Schema	Created on (api field)
Table	name	comment	Table	Table	Created on (api field)