Snowflake Web UI, SnowSQL CLI, and Azure Active Directory (AAD) Integration

This project details the integration of **Azure Active Directory (AAD)** with **Snowflake** to establish a secure, centralized, and streamlined identity and access management (IAM) framework. The solution enables users to leverage their existing corporate AAD credentials for Single Sign-On (SSO) authentication into the Snowflake Web UI and for secure connections via the SnowSQL Command Line Interface (CLI). This eliminates the need to manage separate Snowflake passwords, enhancing security through centralized policy enforcement and multi-factor authentication (MFA).

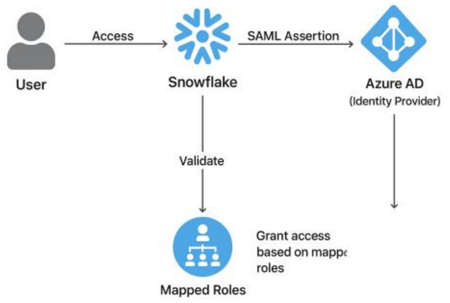
Objectives

The primary objectives of this project are:

- **Enable Single Sign-On (SSO):** Allow users to authenticate to Snowflake using their Azure AD credentials, providing a seamless and secure login experience.
- **Centralize Identity Management:** Manage all user identities and groups centrally within Azure AD, simplifying user lifecycle management (onboarding, offboarding, role changes).
- **Enhance Security Posture:** Leverage Azure AD's security features, including Conditional Access policies and Multi-Factor Authentication (MFA), for all Snowflake access.
- **Enable Federated Authentication for SnowSQL:** Configure SnowSQL, the Snowflake CLI, to authenticate via Azure AD, enabling automation and scripted workflows without storing native Snowflake credentials.
- **Establish Role-Based Access Control (RBAC):** Map Azure AD groups to Snowflake roles to enforce the principle of least privilege.

System Architecture & Design

The integration uses the Security Assertion Markup Language (SAML 2.0) protocol for SSO. The high-level authentication flow is as follows:



1. **User Access Request:** A user attempts to access the Snowflake Web UI or connects via SnowSQL.
2. **Redirection to IdP:** Snowflake redirects the user to Azure AD (the Identity Provider).
3. **Authentication:** The user authenticates with their AAD credentials (including MFA if enforced by a Conditional Access policy).
4. **SAML Assertion:** Upon successful authentication, Azure AD generates a SAML assertion and sends it back to Snowflake.

5. **Access Grant:** Snowflake validates the SAML assertion, creates a session, and grants the user access based on their assigned Snowflake roles.

Implementation Guide

1. Prerequisites

- An active **Azure AD** (P1 or P2 premium license recommended for Conditional Access).
- An active **Snowflake** account with the ACCOUNTADMIN role privilege.
- Global Administrator access in Azure AD.
- SnowSQL CLI installed on a client machine.

2. Phase 1: Azure AD (Identity Provider) Configuration

1. **Register a New Enterprise Application:**
   - In the Azure Portal, navigate to **Azure Active Directory** > **Enterprise applications**.
   - Click **New application** > **Create your own application**.
   - Enter a name (e.g., "Snowflake Production") and select **Integrate any other application you don't find in the gallery (Non-gallery)**.
2. **Configure Single Sign-On:**
   - In the newly created app, go to **Single sign-on** > **SAML**.
   - Click **Edit** on the Basic SAML Configuration section.
     - **Identifier (Entity ID):** https://<your_snowflake_account_url>.snowflakecomputing.com
     - **Reply URL (Assertion Consumer Service URL):** https://<your_snowflake_account_url>.snowflakecomputing.com/fed/login
   - Save the configuration.
3. **Download Federation Metadata XML:**
   - In the **SAML Signing Certificate** section, download the **Federation Metadata XML** file. This file is required for Snowflake configuration.

3. Phase 2: Snowflake (Service Provider) Configuration

1. **Create a Security Integration in Snowflake:**
   - Execute the following SQL command in the Snowflake Web UI (as ACCOUNTADMIN), providing the contents of the downloaded XML file.

sql

```
CREATE SECURITY INTEGRATION azure_ad_int
TYPE = SAML2
ENABLED = TRUE
SAML2_ISSUER = 'https://sts.windows.net/<your-azure-ad-tenant-id>/'
SAML2_SSO_URL = 'https://login.microsoftonline.com/<your-azure-ad-tenant-id>/saml2'
SAML2_PROVIDER = 'AZURE'
SAML2_X509_CERT = '<Paste the contents of the Certificate from the XML file>'
SAML2_SP_INITIATED_LOGIN_PAGE_LABEL = 'Azure AD'
SAML2_ENABLE_SP_INITIATED = TRUE;
```

2.

Retrieve the Snowflake Entity ID and ACS URL:

- Run the following command and note the values. You will need them if you want to double-check the Azure AD app configuration.

```
sql


DESCRIBE SECURITY INTEGRATION azure_ad_int;
```

4.4.

Phase 3: User Provisioning and Role Mapping

1. **Assign Users/Groups in Azure AD:**
   - Go back to your Enterprise Application in Azure AD.
   - Navigate to **Users and groups** and assign the relevant users or, more efficiently, **Azure AD Groups** to the application.
2. **Create Corresponding Roles in Snowflake:**
   - In Snowflake, ensure that roles exist which correspond to the business functions of your Azure AD groups (e.g., BI_ANALYST_ROLE, DATA_ENGINEER_ROLE).
3. **Create Role Mappings (in Snowflake):**
   - For each AAD user who will log in via SSO, a corresponding user must exist in Snowflake. The username should typically be the user's AAD User Principal Name (UPN).
   - Create the user and grant them the appropriate role. The LOGIN_NAME must match the AAD UPN.

```sql
CREATE USER "jane.doe@yourcompany.com"
LOGIN_NAME = 'jane.doe@yourcompany.com'
DEFAULT_ROLE = BI_ANALYST_ROLE
DEFAULT_WAREHOUSE = COMPUTE_WH
MUST_CHANGE_PASSWORD = FALSE; -- Critical for SSO users

GRANT ROLE BI_ANALYST_ROLE TO USER "jane.doe@yourcompany.com";
```

5. Phase 4: Client Connectivity (SnowSQL CLI)

To authenticate SnowSQL using Azure AD, you must use the externalbrowser authentication method.

1. **Configure the SnowSQL Config File:**

   Add or modify a connection section as follows:

   ```ini
   [connections.azure_ad_connection]
   accountname = <your_snowflake_account_identifier>
   username = <your_aad_upn> # e.g., jane.doe@yourcompany.com
   authenticator = externalbrowser
   # No password is specified
   ```

2. **Test the Connection:**
   - Execute the following command in your terminal/command prompt:

     ```bash
     snowsql -c azure_ad_connection
     ```

   - This will open your default web browser and prompt you to authenticate via Azure AD. Upon successful authentication, the SnowSQL CLI will connect.