

Snowflake Integration for Snowpipe Ingestion with Azure

Assumption:

- 1) User has a Snowflake account hosted on Azure.
- 2) User should have ACCOUNTADMIN role in snowflake.
- 3) Snowflake data base structure is ready

```
CREATE OR REPLACE WAREHOUSE SF_ATL_MEETUP_WH;  
CREATE OR REPLACE DATABASE SF_ATL_MEETUP_DB;  
CREATE OR REPLACE SCHEMA SF_ATL_MEETUP_SCHEMA;
```

```
USE WAREHOUSE SF_ATL_MEETUP_WH;  
USE SCHEMA SF_ATL_MEETUP_DB.SF_ATL_MEETUP_SCHEMA;
```

```
CREATE OR REPLACE FILE FORMAT SNFL_SP_JSON_FORMAT  
TYPE = 'json';
```

Step 1: Creating the Integration

Create an integration using the CREATE NOTIFICATION INTEGRATION command. An integration is a Snowflake object that references the Azure storage queue you created.

```
CREATE NOTIFICATION INTEGRATION <integration_name>  
ENABLED = true  
TYPE = QUEUE  
NOTIFICATION_PROVIDER = AZURE_STORAGE_QUEUE  
AZURE_STORAGE_QUEUE_PRIMARY_URI = '<queue_URL>  
AZURE_TENANT_ID = '<directory_ID>;
```

Where:

integration_name is the name of the new integration.

queue_URL and **directory_ID** are the queue URL and tenant ID you recorded in Retrieving the Storage Queue URL and Tenant ID. [Last step in Azure configuration]

Step 2: Granting Snowflake Access to the Storage Queue

- 1) Execute the DESCRIBE INTEGRATION command to retrieve the consent URL:

```
DESC NOTIFICATION INTEGRATION <integration_name>;
```

Where:

integration_name is the name of the integration you created in Creating the Integration.

2) Note the URL in the AZURE_CONSENT_URL column, which has the following format:
`https://login.microsoftonline.com/<tenant_id>/oauth2/authorize?client_id=<snowflake_application_id>`

3) Navigate to the URL in a web browser. The page displays a Microsoft permissions request page.

4) Click the Accept button to register Snowflake in Active Directory.

5) Log into the Microsoft Azure portal.

6) Navigate to Azure Active Directory » Enterprise applications. Verify the Snowflake application is listed.

7) Navigate to Queues » storage_queue_name, where storage_queue_name is the name of the storage queue you created in Creating a Storage Queue.

8) Search for the Snowflake application.

9) Grant the Snowflake app the following permissions:

Role: Storage Queue Data Contributor (Preview)

Assign access to: Azure AD user, group, or service principle

Select: snowflake_application_id

The Snowflake application should now be listed under Storage Queue Data Contributor (on the same dialog).

Step 3: **Create a Stage**

```
CREATE OR REPLACE STAGE mystage
```

```
URL = 'azure://myaccount.blob.core.windows.net/mycontainer'
```

```
CREDENTIALS = (AZURE_SAS_TOKEN='*****');
```

Where:

URL:

azure://**myaccount**.blob.core.windows.net/**mycontainer**

Myaccount: Name of the storage account in Azure

Mycontainer: Name of the container inside the storage account in Azure.

AZURE_SAS_TOKEN:

The screenshot shows the Azure portal interface for configuring a Shared Access Signature (SAS) for a storage account named 'sfDemoBlob'. The left sidebar contains navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Data transfer, Events, Storage Explorer (preview), Settings, Access keys, Geo-replication, CORS, Configuration, Encryption, Shared access signature (selected), Firewalls and virtual networks, Private endpoint connection..., Advanced security, and Properties. The main content area is titled 'SAS.' and includes a 'Learn more' link. The configuration options are as follows:

- Allowed services:** ☒ Blob
- Allowed resource types:** ☒ Service ☒ Container ☒ Object
- Allowed permissions:** ☒ Read ☒ Write ☒ Delete ☒ List ☒ Add ☒ Create ☐ Update ☐ Process
- Start and expiry date/time:**
 - Start:** 11/06/2019 6:04:08 PM
 - End:** 11/07/2019 2:04:08 AM (UTC-05:00) --- Current Time Zone ---
- Allowed IP addresses:** (for example, 168.1.5.65 or 168.1.5.65-168.1.5.70)
- Allowed protocols:** ☒ HTTPS only ☐ HTTPS and HTTP
- Signing key:** key1

A blue button at the bottom right says 'Generate SAS and connection string'.

Step 4: Create a Pipe with Auto-Ingest Enabled

Create a pipe using the CREATE PIPE command.

The pipe defines the COPY INTO <table> statement used by Snowpipe to load data from the ingestion queue into the target table.

```
CREATE OR REPLACE PIPE <pipe_name>
```

```
    auto_ingest = true
```

```
    integration = '<Integration_name>'
```

```
as
```

```
    COPY INTO
```

```
        <database_table>
```

```
FROM (
```

```
    SELECT
```

```
        $1 AS RAW_DATA,
```

```
        CURRENT_TIMESTAMP() AS INGESTION_TIME
```

```
FROM mystage);
```

Where:

Integration_name is the name of the integration you created in Creating the Integration.