SNOWFLAKE ASSIGNMENT 2

TASK 2 : Set Up Snowpipe with Azure Blob Storage and monitor using COPY_HISTORY.

1. SAS Token + REFRESH PIPE (Manual / Semi-Automated)

Goal

- Use a SAS token to allow Snowflake to read files from Azure Blob Storage.
- No Azure role assignment needed.
- Semi-auto: new files are loaded when you refresh the pipe manually.

Step 1 — Generate SAS Token in Azure

1. Go to your Azure Portal → Storage Account → sfhexastorage → Containers → snowflakecontainer.
2. Click Shared access signature.
3. Set permissions:
   - Read (r)
   - List (l)
   - Optional: Write (w) only if Snowflake writes (usually not needed).
4. Set expiry date: e.g., 1 year.
5. Copy the SAS token (without the leading ?).

Example SAS token (shortened):
sv=2025-01-01&ss=b&srt=co&sp=rl&se=2025-12-31T23:59Z&st=2025-01-01T00:00Z&spr=https&sig=XXXX

Step 2 — Create File Format in Snowflake

```
CREATE OR REPLACE FILE FORMAT csv_ff
  TYPE = 'CSV'
  FIELD_DELIMITER = ','
  SKIP_HEADER = 1
  FIELD_OPTIONALLY_ENCLOSED_BY = '"';
```

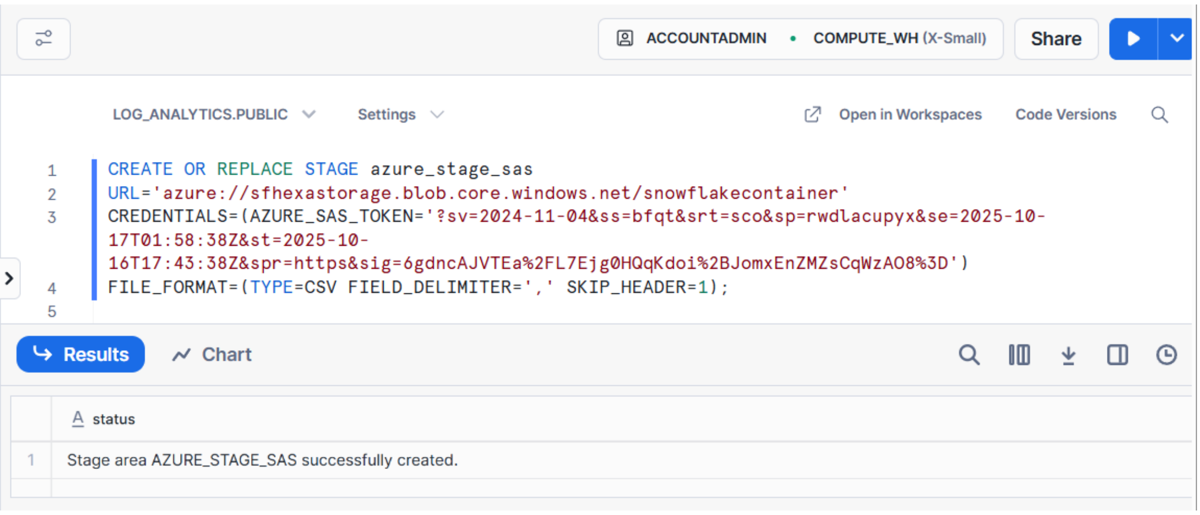
Step 3 — Create Stage using SAS Token

```
CREATE OR REPLACE STAGE azure_stage_sas
URL='azure://sfhexastorage.blob.core.windows.net/snowflakecontainer'
```

```
CREDENTIALS=(AZURE_SAS_TOKEN='sv=2025-01-01&ss=b&srt=co&sp=rl&se=2025-12-31T23:59Z&st=2025-01-01T00:00Z&spr=https&sig=XXXX')

FILE_FORMAT = csv_ff;
```

Replace the SAS token with the one you generated.



```
CREATE OR REPLACE STAGE azure_stage_sas
URL='azure://sfhexastorage.blob.core.windows.net/snowflakecontainer'
CREDENTIALS=(AZURE_SAS_TOKEN='?sv=2024-11-04&ss=bfqt&srt=sco&sp=rwdlacupyx&se=2025-10-17T01:58:38Z&st=2025-10-16T17:43:38Z&spr=https&sig=6gdncAJVTEa%2FL7Ejg0HQqKdoi%2BJomxEnZMZsCqWzAO8%3D')
FILE_FORMAT=(TYPE=CSV FIELD_DELIMITER=',' SKIP_HEADER=1);
```

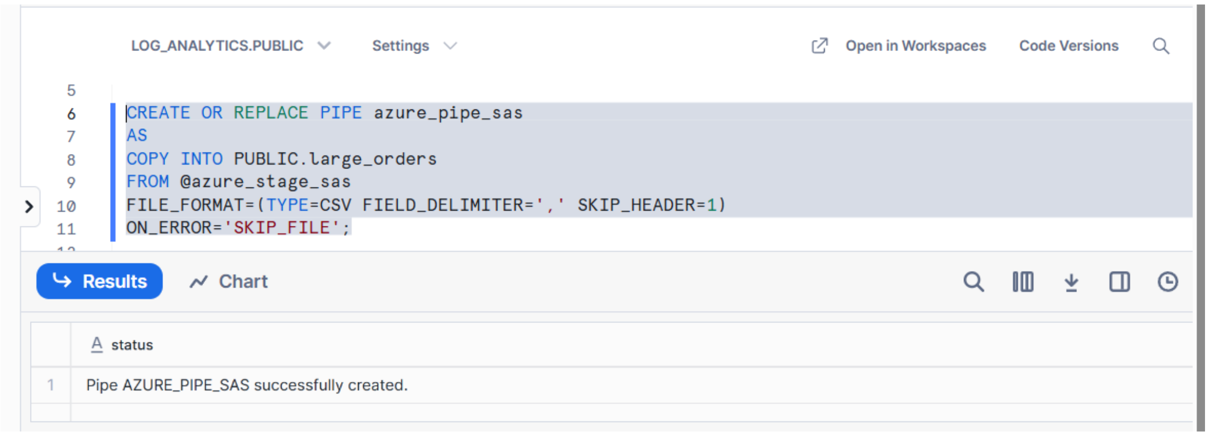
	status
1	Stage area AZURE_STAGE_SAS successfully created.

Step 4 — Create Target Table

```
CREATE OR REPLACE TABLE CUSTOMER_DATA(
  CustomerID STRING,
  Name STRING,
  Email STRING,
  Phone STRING,
  Loaded_At TIMESTAMP_LTZ DEFAULT CURRENT_TIMESTAMP()
);
```

Step 5 — Create Pipe

```
CREATE OR REPLACE PIPE azure_pipe_sas
AS
COPY INTO CUSTOMER_DATA
FROM @azure_stage_sas
FILE_FORMAT = (FORMAT_NAME = 'csv_ff')
ON_ERROR = 'CONTINUE';
```



```
CREATE OR REPLACE PIPE azure_pipe_sas
AS
COPY INTO PUBLIC.large_orders
FROM @azure_stage_sas
FILE_FORMAT=(TYPE=CSV FIELD_DELIMITER=',' SKIP_HEADER=1)
ON_ERROR='SKIP_FILE';
```

	status
1	Pipe AZURE_PIPE_SAS successfully created.

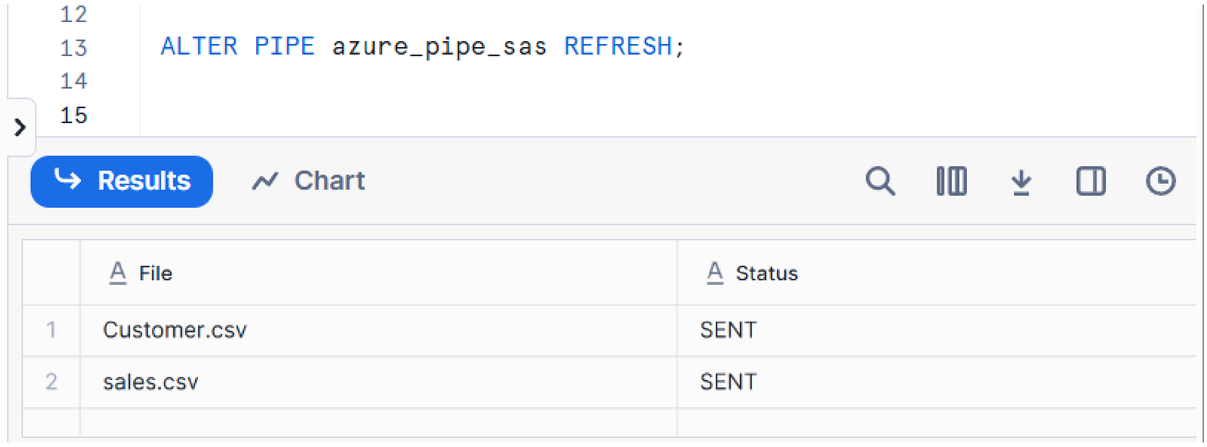
Notes:

- AUTO_INGEST = TRUE cannot be used here because it requires Azure role assignment.
- Instead, we'll manually trigger ingestion using ALTER PIPE ... REFRESH.

Step 6 — Load Files (Semi-Automatic)

Whenever new files arrive in the container:

```
ALTER PIPE azure_pipe_sas REFRESH;
```



```
ALTER PIPE azure_pipe_sas REFRESH;
```

	File	Status
1	Customer.csv	SENT
2	sales.csv	SENT

- Snowflake will check the stage for new files uploaded within the last 7 days.
- It copies them into CUSTOMER_DATA.

Step 7 — Monitor Loads

1. Check recent loads:

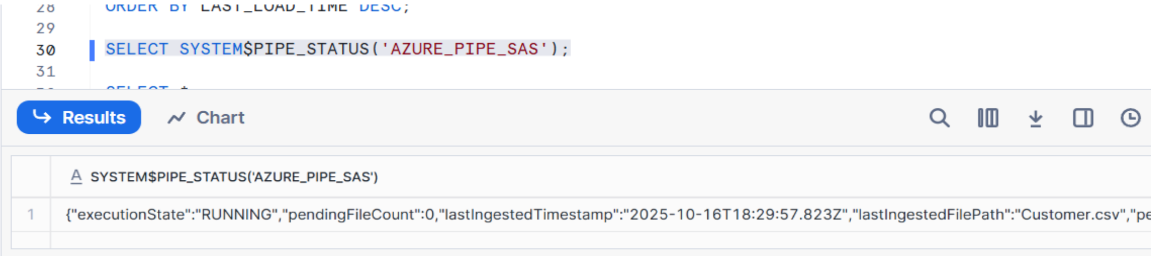
```
SELECT *
FROM TABLE(COPY_HISTORY(
  TABLE_NAME=>'CUSTOMER_DATA',
  START_TIME=>DATEADD(day, -1, CURRENT_TIMESTAMP())
))
ORDER BY LAST_LOAD_TIME DESC;
```

2. Check account-wide history (last 365 days):

```
SELECT *
FROM SNOWFLAKE.ACCOUNT_USAGE.COPY_HISTORY
WHERE TABLE_NAME = 'CUSTOMER_DATA'
ORDER BY LAST_LOAD_TIME DESC;
```

3. Optional: check pipe status:

```
SELECT SYSTEM$PIPE_STATUS('AZURE_PIPE_SAS');
```



```
ORDER BY LAST_LOAD_TIME DESC;

SELECT SYSTEM$PIPE_STATUS('AZURE_PIPE_SAS');

SELECT *
```

	SYSTEM$PIPE_STATUS('AZURE_PIPE_SAS')
1	{"executionState":"RUNNING","pendingFileCount":0,"lastIngestedTimestamp":"2025-10-16T18:29:57.823Z","lastIngestedFilePath":"Customer.csv","pe

- Returns JSON with:
  - executionState
  - numFilesQueued
  - lastReceivedMessageTimestamp

Step 8 — Automate Semi-Auto Pipeline (Optional)

Since AUTO_INGEST is not possible:

- Schedule ALTER PIPE ... REFRESH every few minutes using:
  - Snowflake Task (preferred)
  - Python / Databricks script calling Snowflake REST API

Example Snowflake Task:

```
CREATE OR REPLACE TASK refresh_customer_pipe
  WAREHOUSE = COMPUTE_WH
  SCHEDULE = 'USING CRON 0/5 * * * * UTC'
AS
ALTER PIPE azure_pipe_sas REFRESH;
```

- This runs every 5 minutes → automatically loads new files.