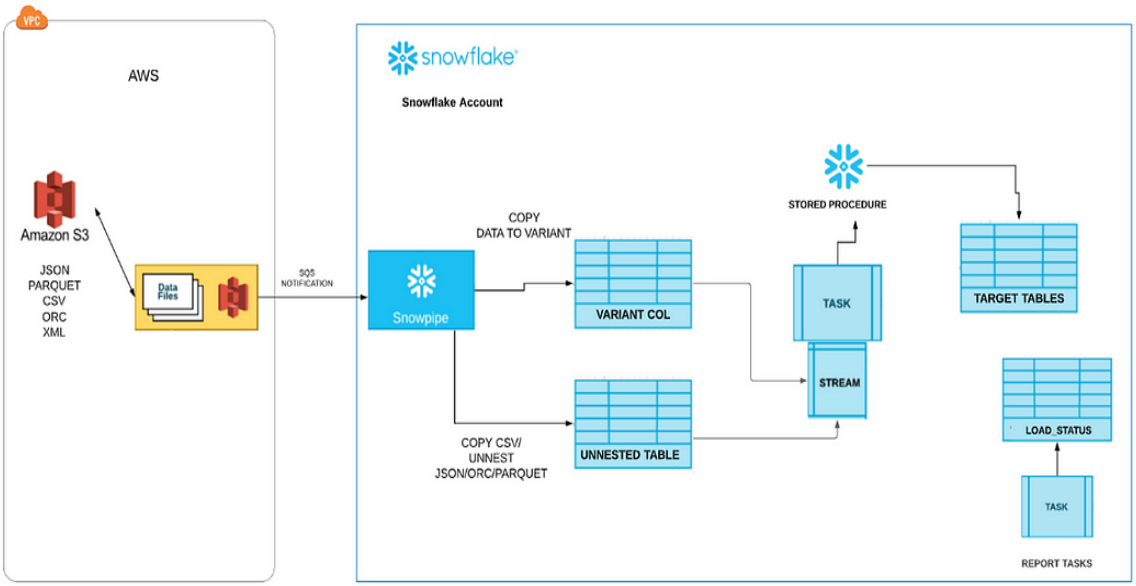
BUILD A SNOWFLAKE CONTINUOUS DATA PIPELINE USING SNOWPIPE, AWS, STREAMS, TASKS & EXTERNAL STAGES

GOALS

In this project, we to build Data PIPELINE to automate the manual steps involved in building and managing ELT logic for transforming and optimizing continuous data loads using Snowflake DATA PIPELINE.

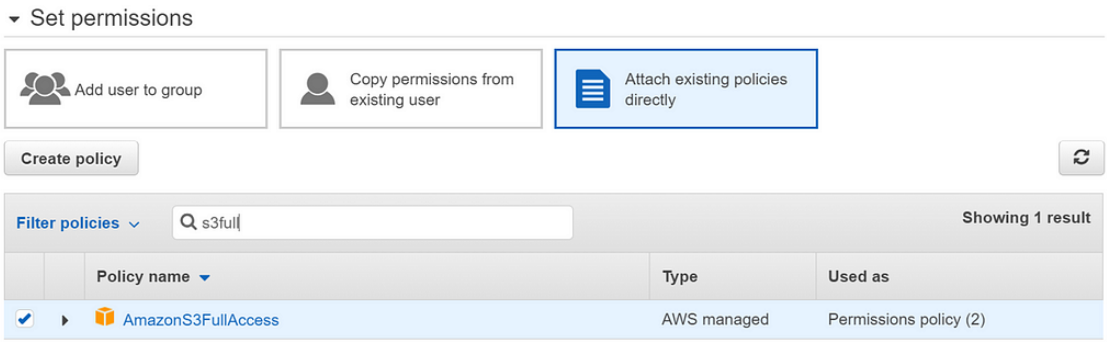
We will use the Snowflake features to enable continuous data pipelines.

- **External Stage on s3**
- **SnowPipe**
- **Streams**
- **Tasks**
- **Stored Procedures**

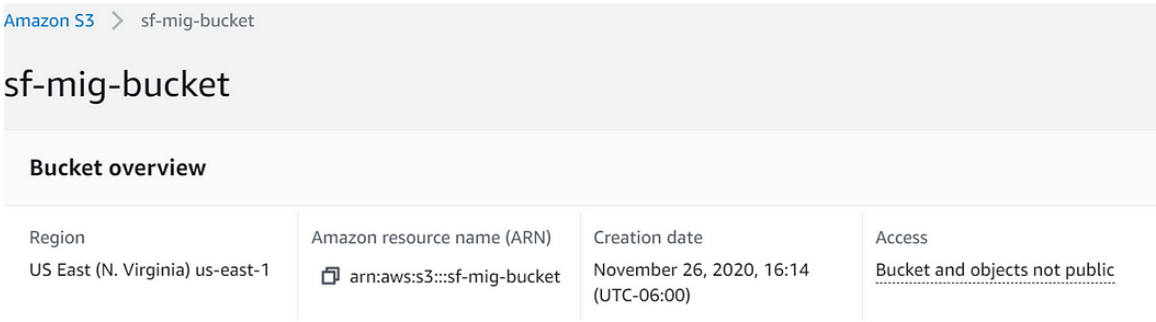


External Stage on S3:

a. **Create User in AWS with Programmatic access and copy the credentials**.



b. **Create s3 bucket**



c. **Create Stage:** Use below SQL statement in Snowflake to **create external stage on s3(AWS)**.

d. **CREATE table in Snowflake with VARIANT column**.

e. **Create a Snowpipe with Auto Ingest Enabled**

f. **Subscribe the Snowflake SQS Queue in s3:**

g. **Test Snowpipe by copying the sample JSON file and upload the file to s3 in path**

Below are few ways we can validation if Snowpipe ran successfully.

1 . Check the pipe status using below command, it shows RUNNIG and it also shows pendingFileCount.

2. Check COPY_HISTORY for the table you are loading data to. If there is any error with Data Load, you can find that error here to debug the Load issue.

3. Finally check if data is loaded to table by querying the table.

Change Data Capture using Streams, Tasks and Merge.

1.Create Streams on PERSON_NESTED table to capture the change data on PERSON_NESTED table and use TASKS to Run SQL/Stored Procedure to Unnested the data from PERSON_NESTED and create PERSON_MASTER table.

2. Create a table to Load the unnested data from PERSON_NESTED.

3. Create a TASK which run every 1 min and look for data in Stream PERSON_NESTED_STREAM, if data found in Stream then task will EXECUTE if not TASK will be SKIPPED without any doing anything.

4. **Test PIPELINE**

a) All the tables and Steam is empty, if not Truncate them.

b) Upload sample JSON data to s3 created

c) Select data from PERSON_NESTED: Snowpipe would have loaded data to PERSON_NESTED table based on s3 sqs event notification.

d) Check COPY HISTORY to know the status of COPY command and number of files copied.

e) Steams capture any data change on the source table(PERSON_NESTED). So all the new data added to PERSON_NESTED should be in PERSON_NESTED_STREAM. Stream also contains additional columns which says if its INSERT/UPDATE/DELETE and it also contain unique METADATA$ROW_ID. Check those Columns.

f) As we have created task to run every 1 min if there is data in Stream, you should be able to see the data in PERSON_MASTER table now.

g) Once stream gets consumed in any DML operation the data from stream(PERSON_NESTED_STREAM) will be erased, PERSON_NESTED_STREAM steam will be empty

now as TASK ran and loaded the data to PERSON_MASTER.

ELT IN SNOWFLAKE USING STORED PROCEDURE

a) Create stored procedure to run Multiple SQL statements to automate data Load from PERSON_MASTER to two tables PERSON_AGE(Name, Age) and PERSON_LOCATION(Name, Location). This stored procedure should be called by TASK.

b) **Stored Procedure Call** :

c) CALL PERSON_MASTER_PROCEDURE(arguments1);

Create Stored Procedure which runs below 2 SQLs.

1. **Insert data into Location table from Person Master table.**
2. **Insert data into Age table from Person Master table.**