RETAIL SALES DATA ANALYSIS PROJECT

SNOWFLAKE CONTINUOUS DATA LOADING

1]. Create an AWS account in aws.amazon.com

2]. After successful account creation and activation, you can use the AWS service.

3]. Go to the Console home and search for S3 (Simple Storage Service) and click on it.

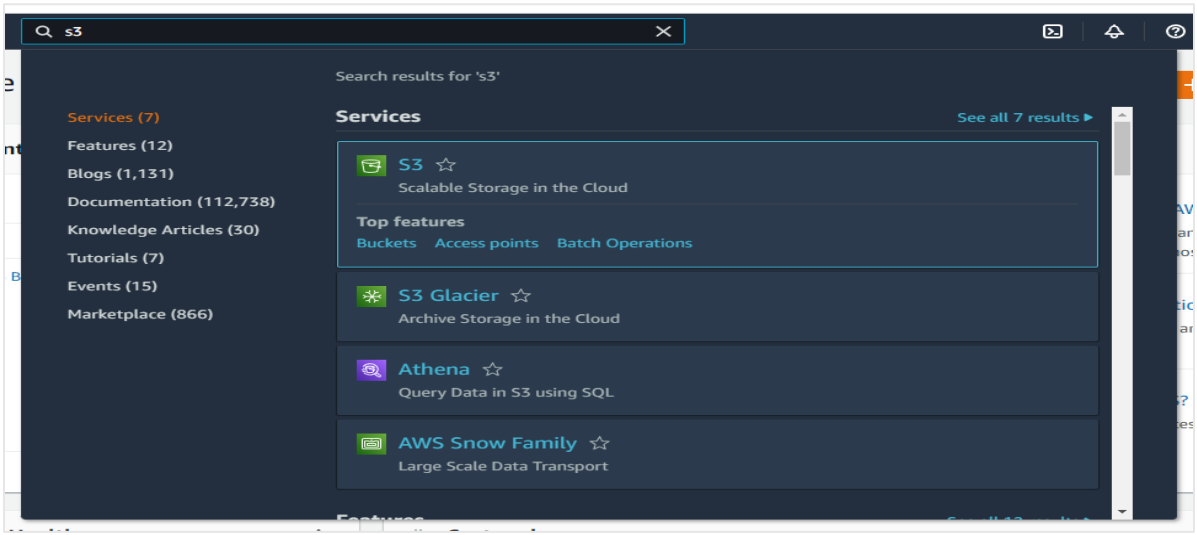

Figure 1: s3 bucket searching

4]. Create S3 bucket

Bucket name: **sk-retail-bucket**

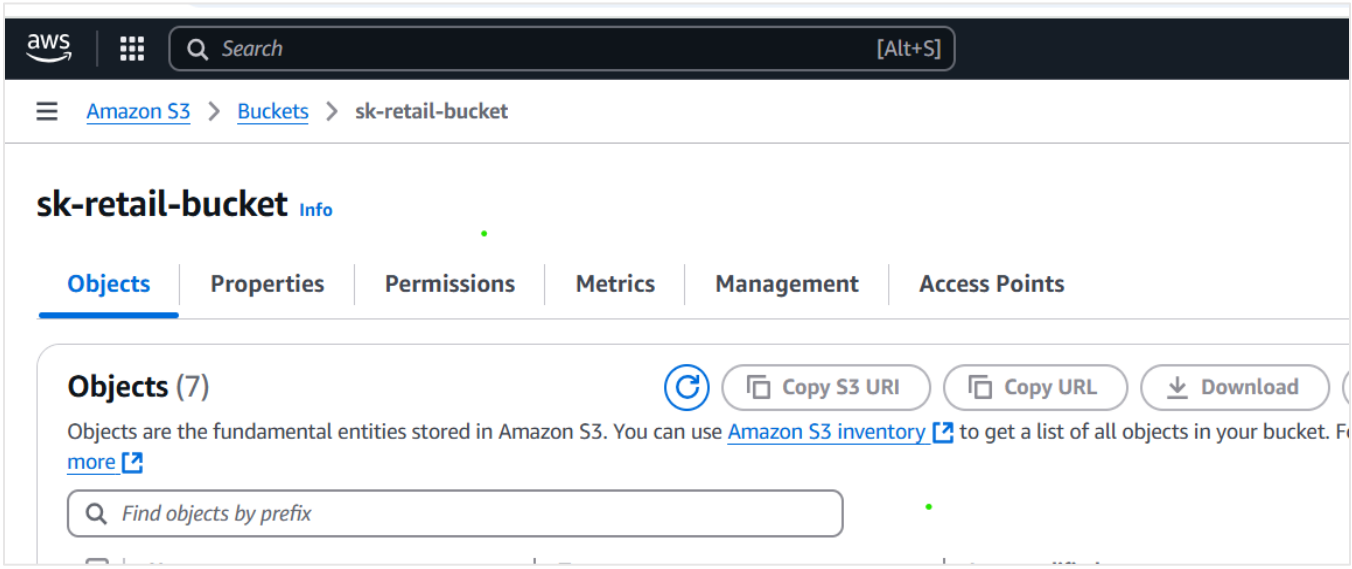

Figure 2: S3 bucket creation

5]. Create a folder inside the bucket (e.g. DEMOGRAPHIC)

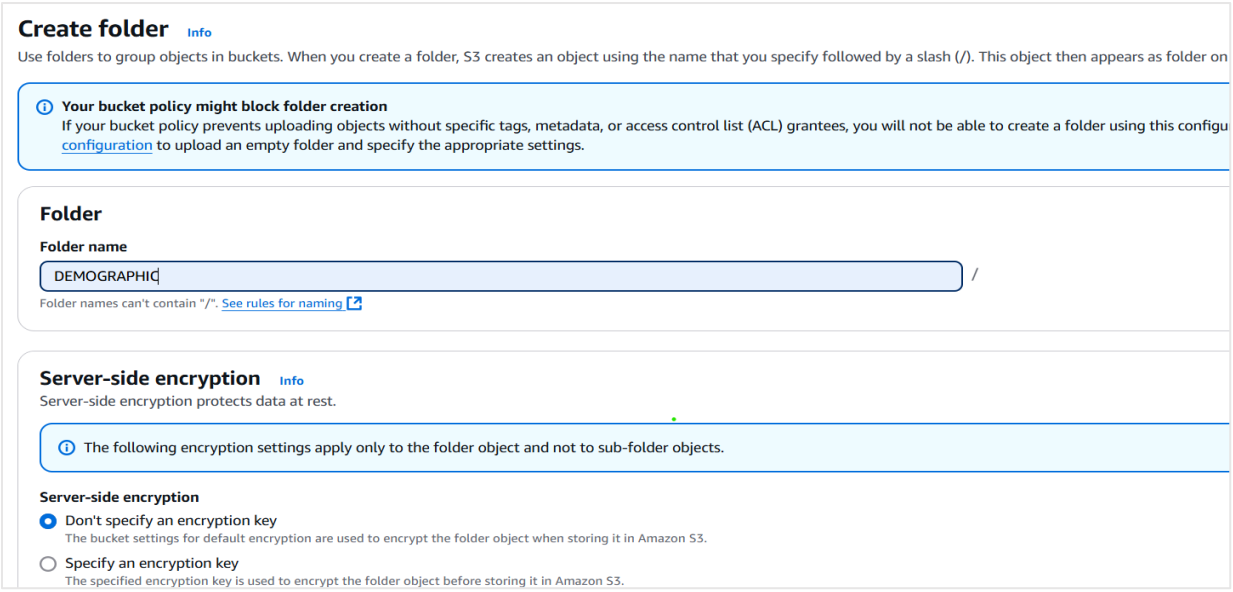

Figure 3: Folder creation inside bucket

For this project we created 7 folders in our bucket:

1) CAMPAIN_DSC
2) CAMPAIN
3) COUPON_REDEMPT
4) COUPON
5) DEMOGRAPHIC
6) PRODUCT
7) TRANSACTION

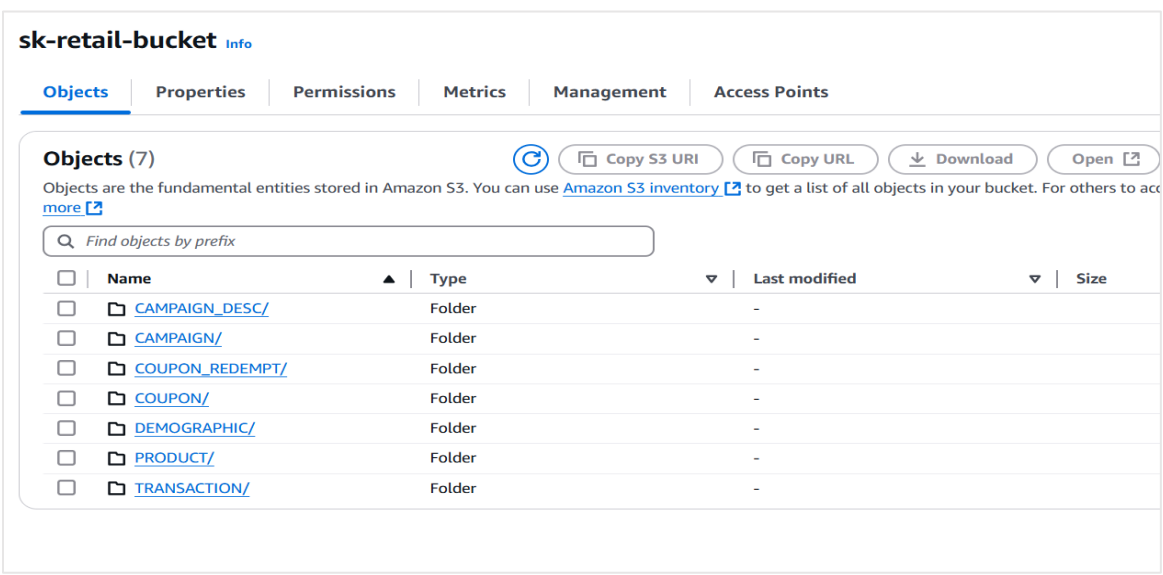

Figure 4: All folders

6]. **Once the S3 bucket and folders are created, search and select the IAM (Identity and Access Management) service from the AWS console**.

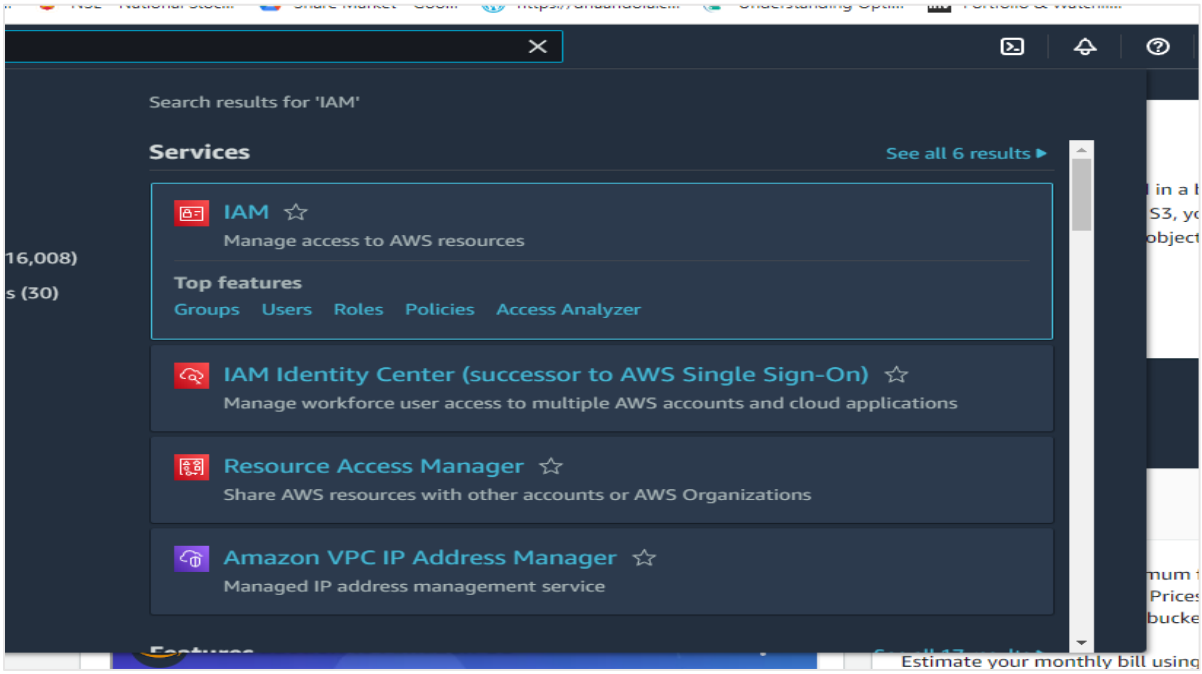

Figure 5: IAM role in AWS

7]. Click on the Policies from IAM Dashboard

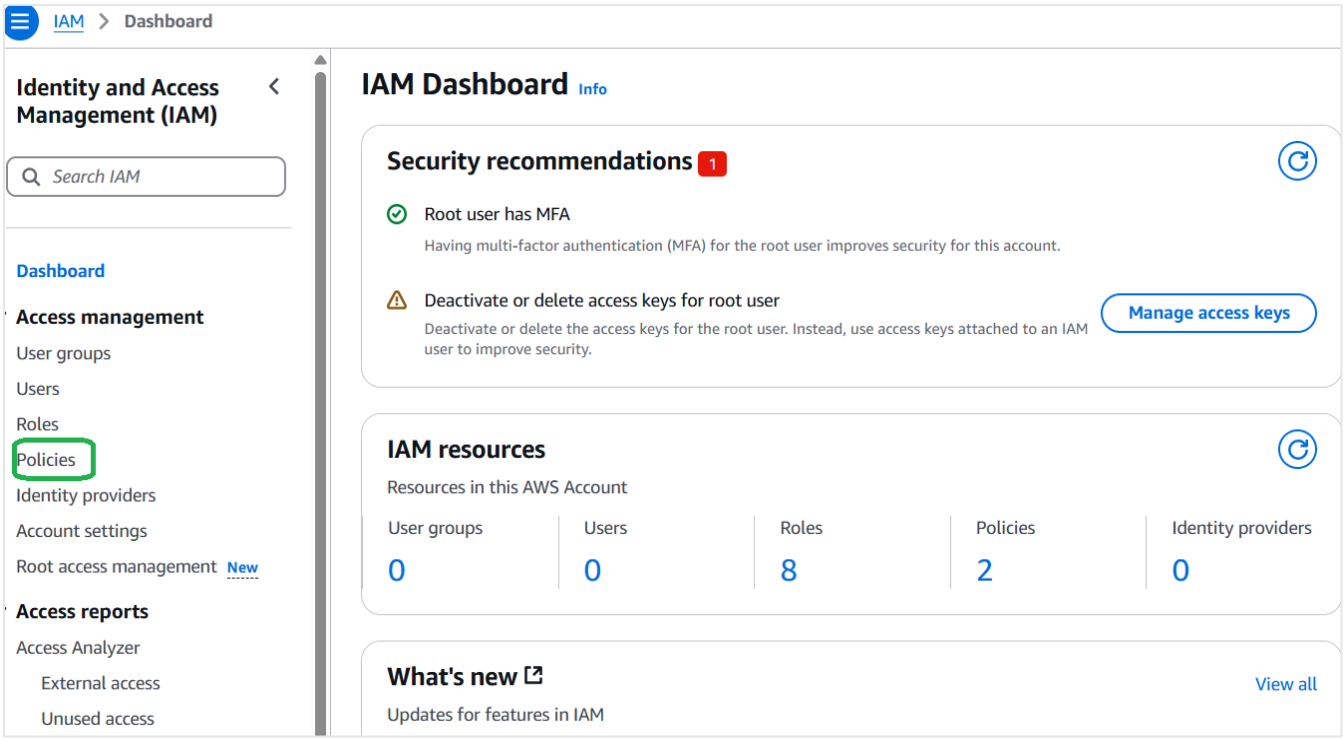

Figure 6:Search for policy

8]. Create IAM policy for the bucket by clicking on the “Create Policy” button

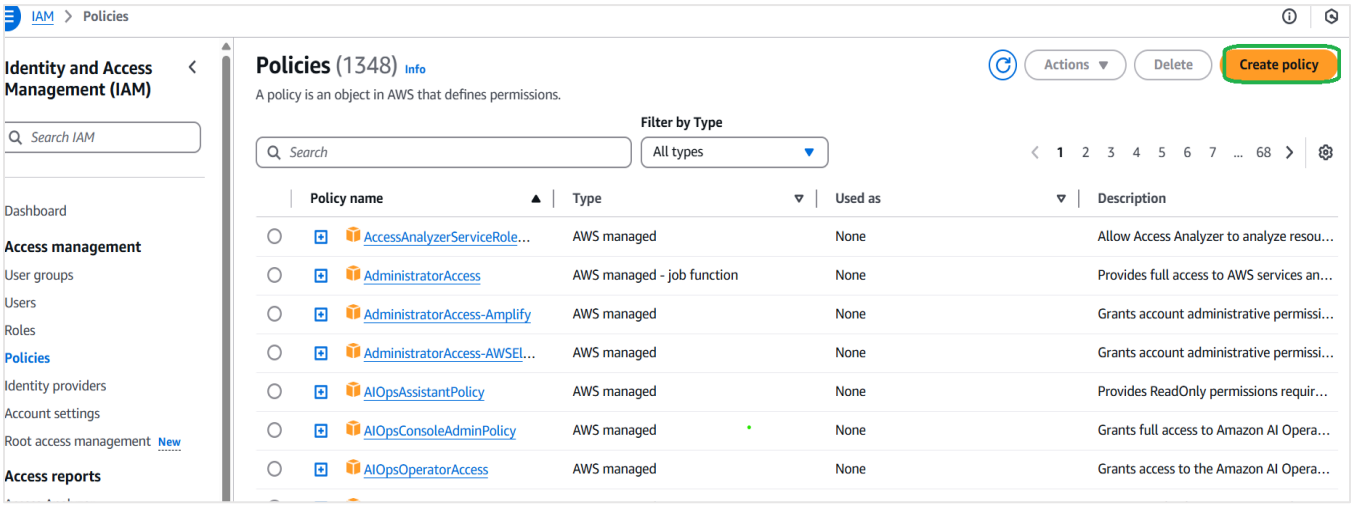

Figure 7: Create policy

9]. Click on the JSON tab and replace the existing text with the text given in the reference Document click given link for documents.

[Option 1: Configuring a Snowflake Storage Integration to Access Amazon S3 | Snowflake Documentation]

After clicking on the above link you will get following doc then just copy the code.

(It is under the step no. 8 from the document)

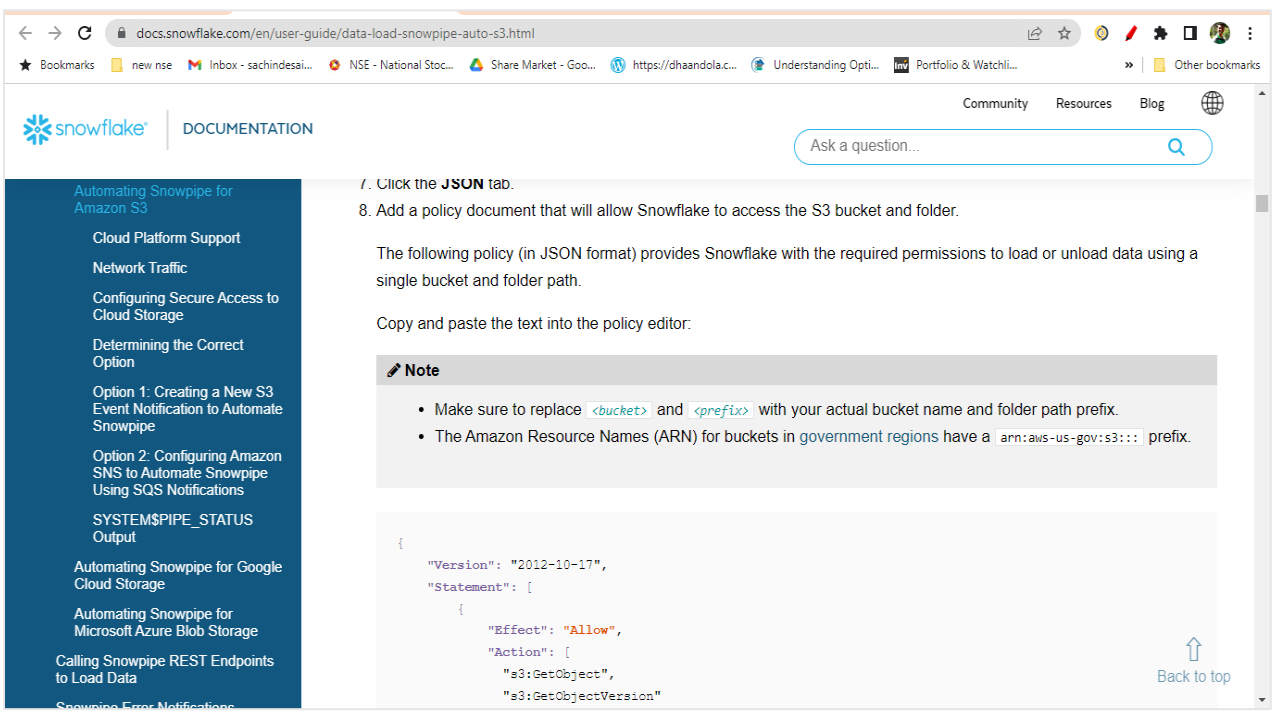

Figure 8: Policy permission script

10]. Replace the <bucket> and <prefix > with your actual bucket name and folder path.

Also set the S3: prefix to “ *”

"s3: prefix”: [
“*"

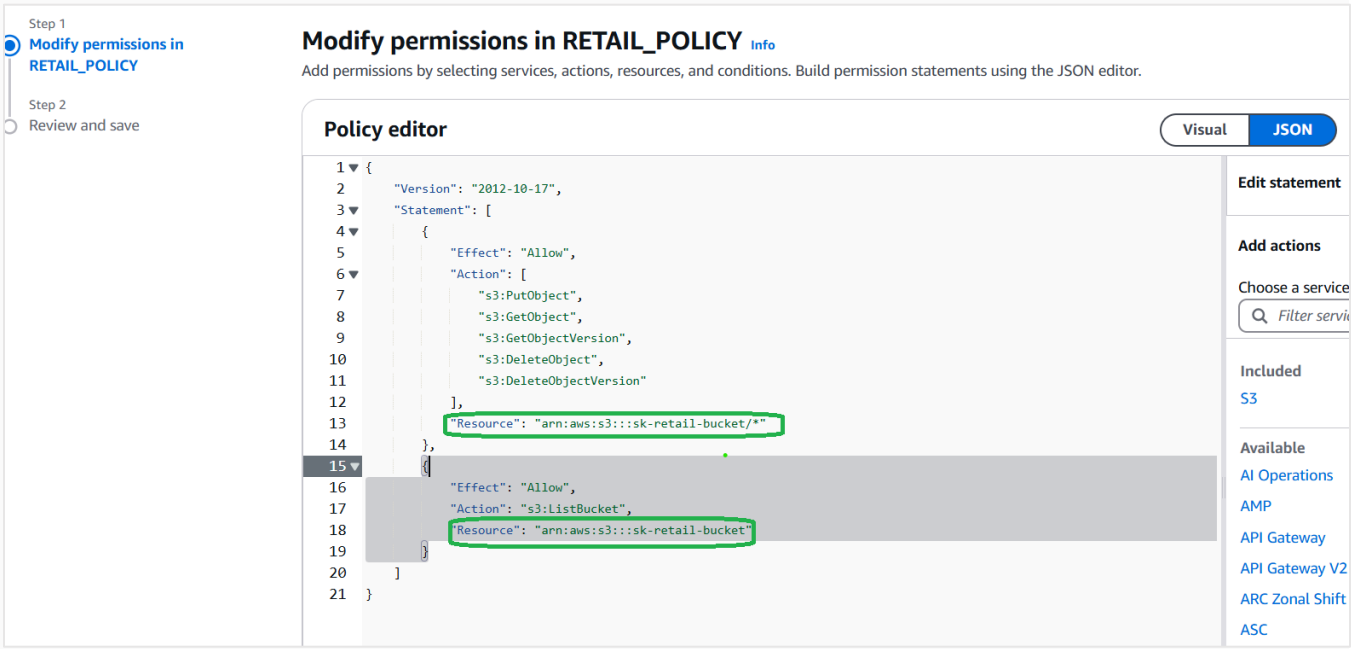

Figure 9: Policy JSON script

11]. Click Next then skip the Add Tags. Enter the policy name → Click Create Policy.

Your policy will get created. (eg: RETAIL_POLICY)

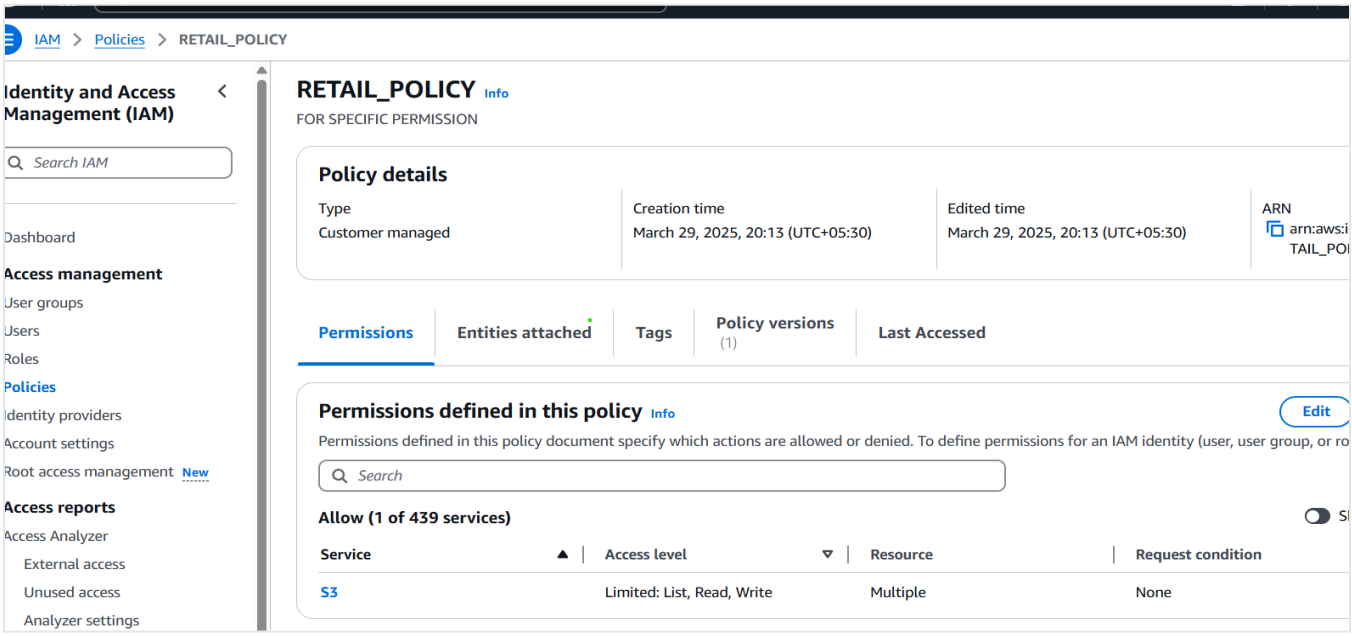

Figure 10: Created Policy

12]. Create IAM Role. Click on Create Role

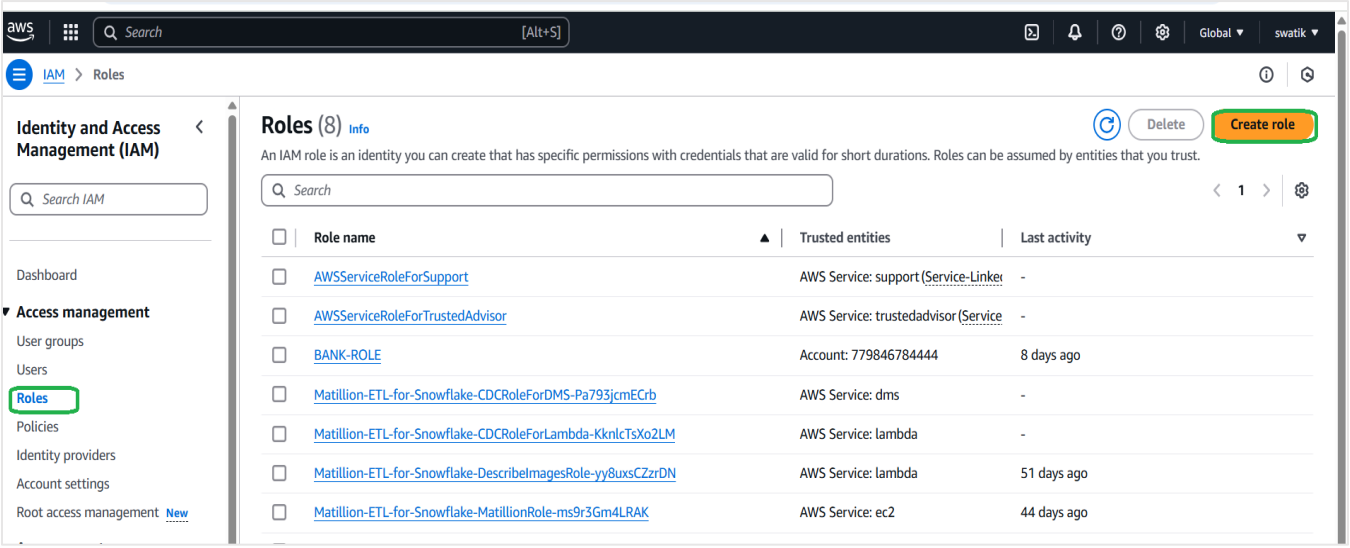

Figure 11: Creating role

13]. Select AWS Account from Trusted Entity Type.

You will get your account number selected by default when you select AWS account.

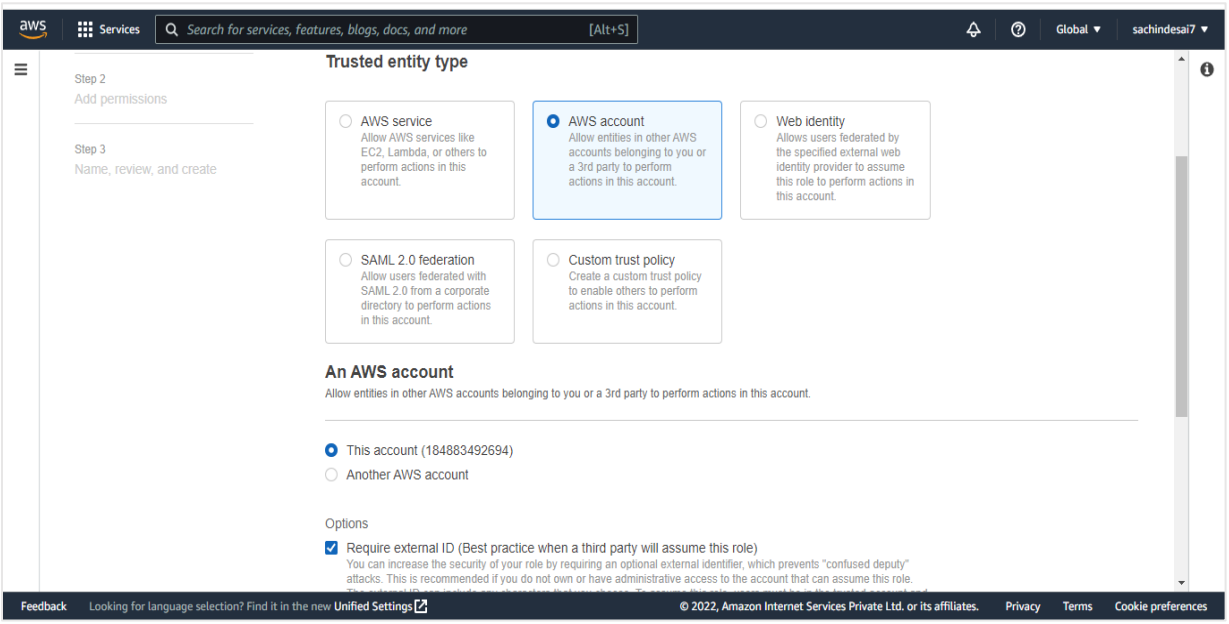

Figure 12: AWS account selection

14] Check Require external ID and enter 000 (as currently we are not having it) and click next (Optional)

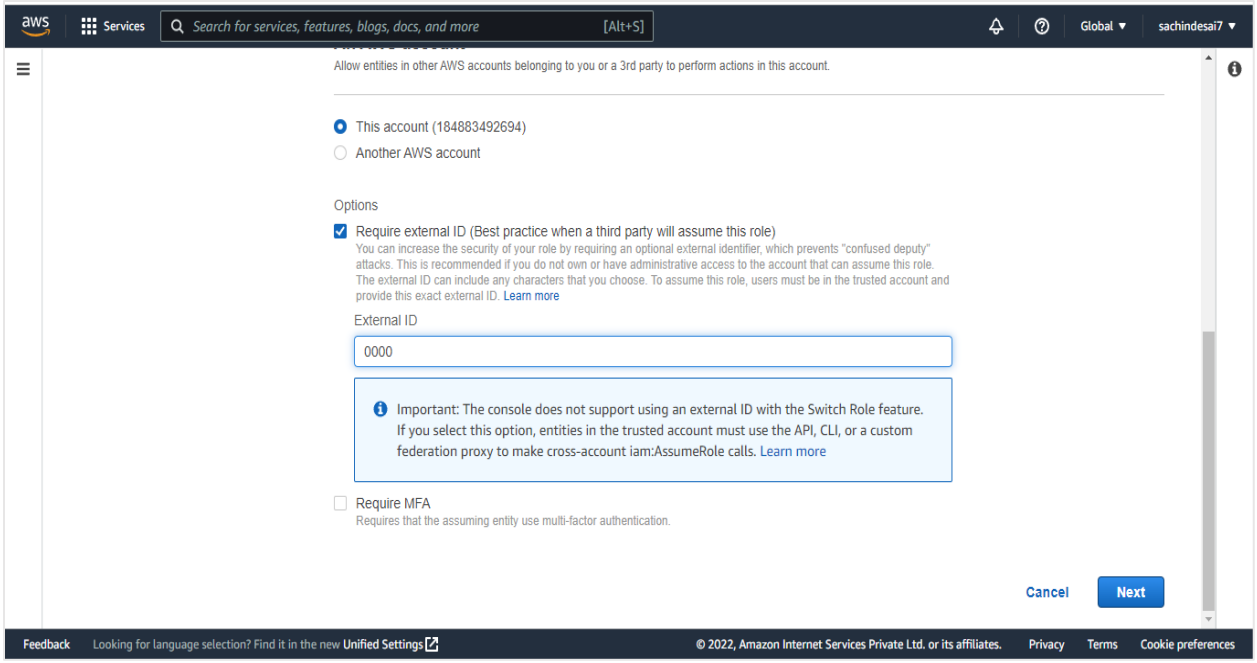

Figure 13: Provide external id

15]. On the next page, Select the IAM policy that you have created. Linked policy to the IAM role.

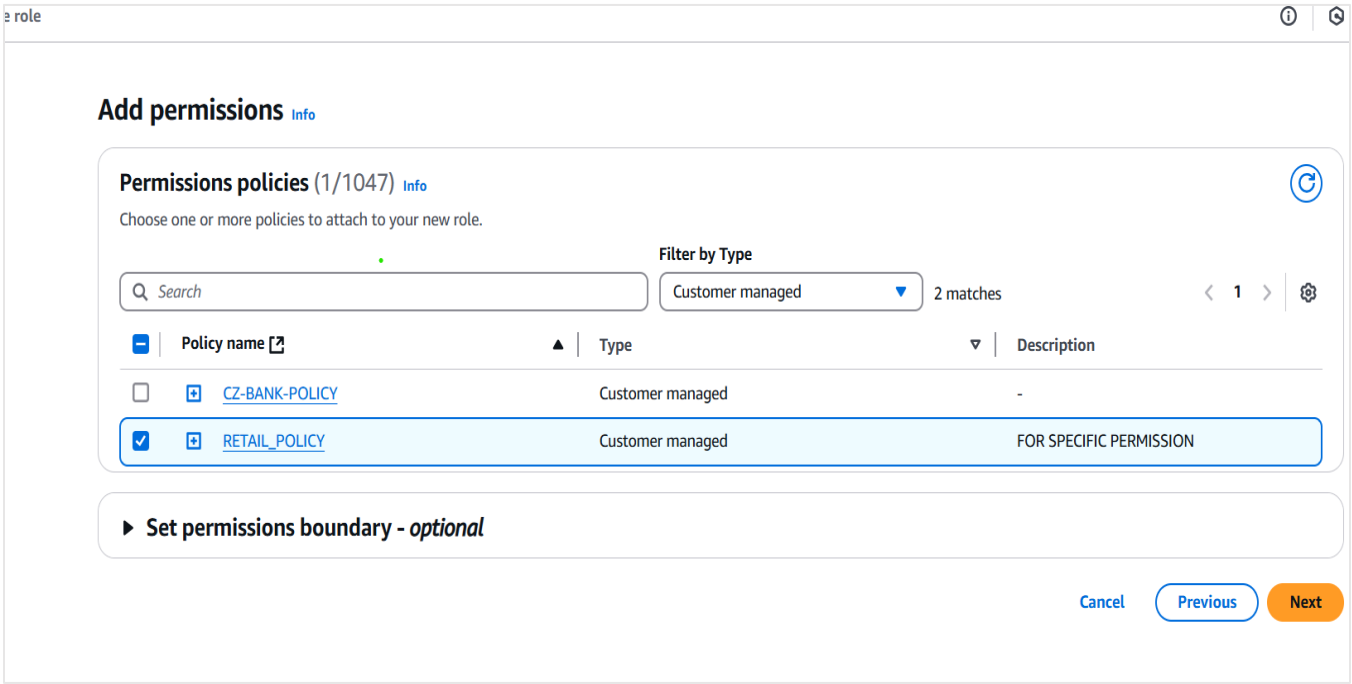

Figure 14: Linked role to policy

16]. On the next page Enter any unique name to the role you are creating. The description is optional.

Click on the Create Role (Skip the Add Tags).

Click on the role that you have created. It will show you the summary page.

You will get the following window

Note down the Role ARN, which we will need when we create the 'Storage Integration'.

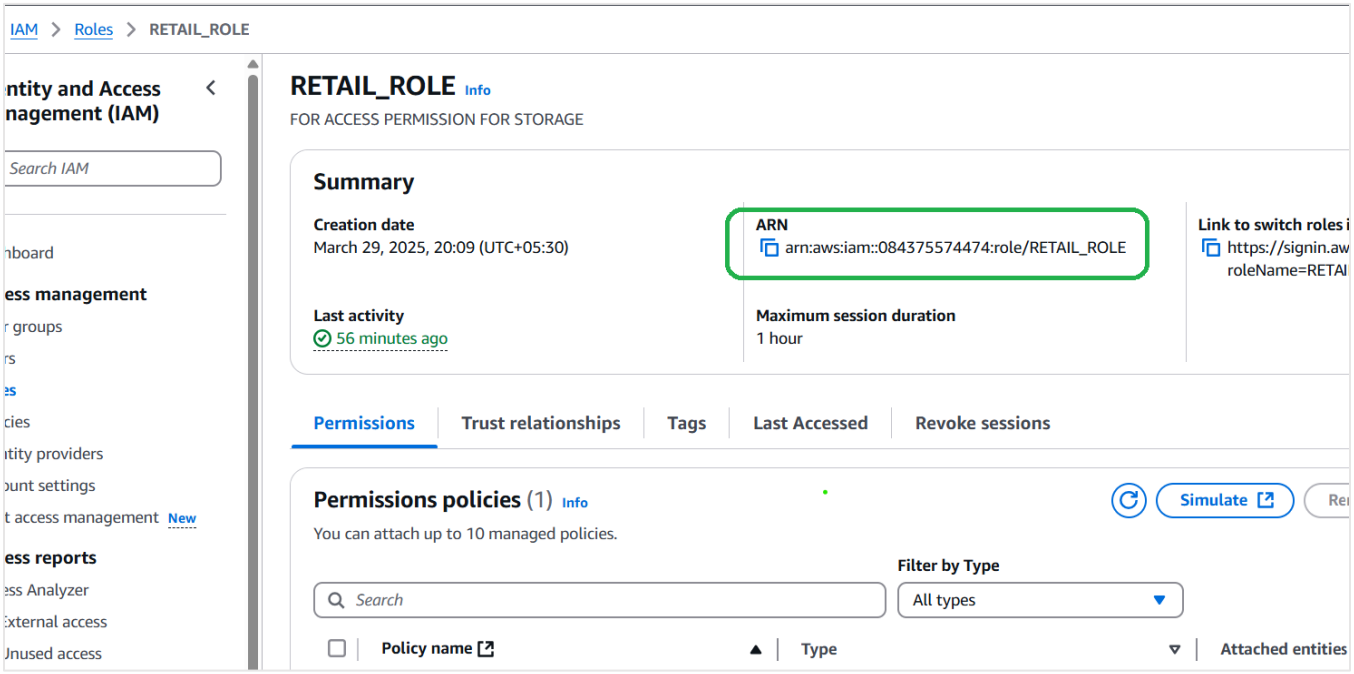

Figure 15: Retail role created

17]. Login to the Snowflake Account.

Create Cloud Storage Integration in Snowflake and map S3 user/role with it (STORAGE_AWS_ROLE_ARN).

-- CREATE STORAGE INTEGRATION

```
CREATE OR REPLACE STORAGE INTEGRATION  s3_int_retail

TYPE = EXTERNAL_STAGE

STORAGE_PROVIDER = S3

ENABLED = TRUE

STORAGE_AWS_ROLE_ARN = 'arn:aws:iam::084375574474:role/RETAIL_ROLE'

STORAGE_ALLOWED_LOCATIONS = ('s3://sk-retail-bucket/');
```

- s3_int_retail → cloud storage integration name
- STORAGE_AWS_ROLE_ARN→ is the Amazon Resource Name (ARN) of the role you created
- STORAGE_ALLOWED_LOCATIONS → is the name of a S3 bucket that stores your data files (eg: sk-retail-bucket)

18]. In Snowflake worksheet run command

-- Desc integration integration_name;

DESC STORAGE INTEGRATION s3_int_retail;

And note down the STORAGE_AWS_IAM_USER_ARN and STORAGE_AWS_EXTERNAL_ID from the result set

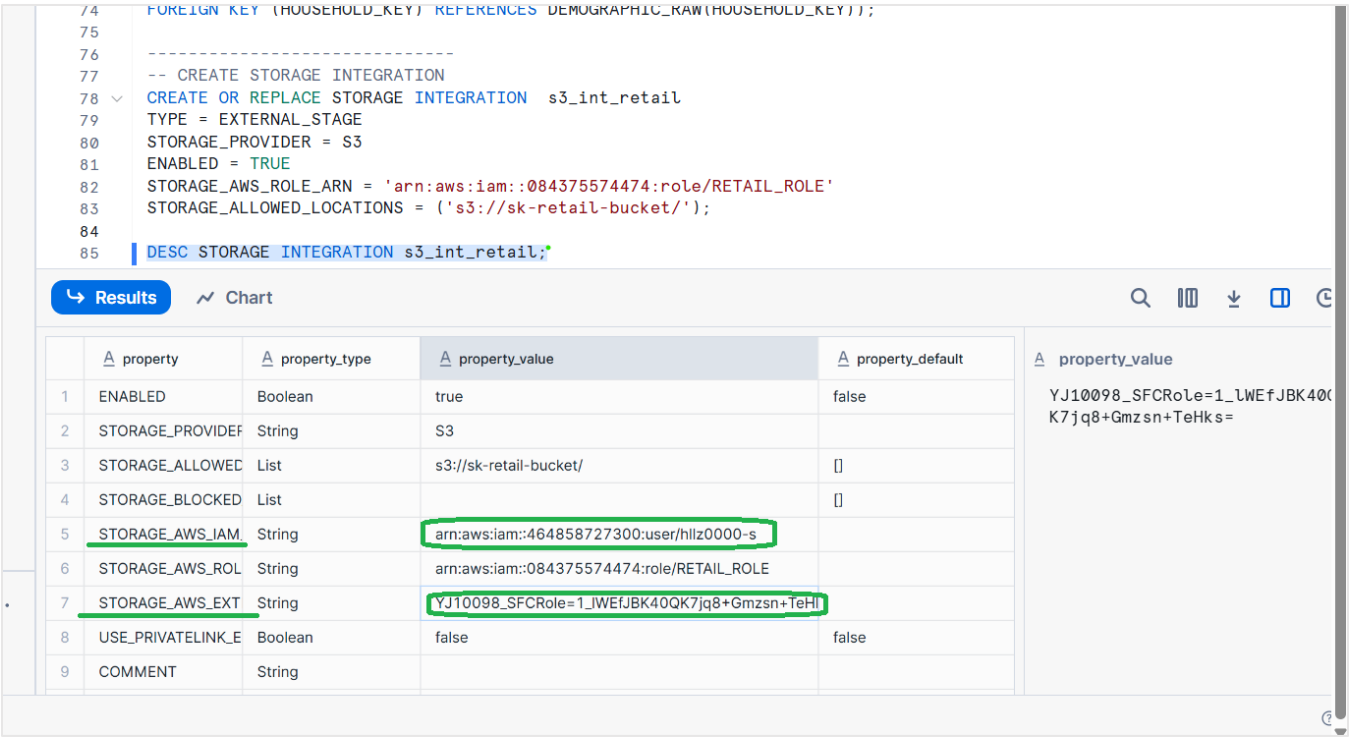

Figure 16: AWS external id

19]. Grant the IAM User Permissions to Access Bucket Objects¶

1. Log into the AWS Management Console.
2. From the home dashboard, choose Identity & Access Management (IAM)
3. Select the role you created
4. Click Trust Relationships -> Edit trust relationship
5. Replace the value of "AWS": with the AWS_IAM_USER_ARN String you got using DESC INTEGRATION command and, value of "sts:ExternalId": with AWS_EXTERNAL_ID String
6. Click Update Policy

Figure 17: edit trust policy

20]. Create Snowflake file format. This file format will be used at the time of Stage creation.

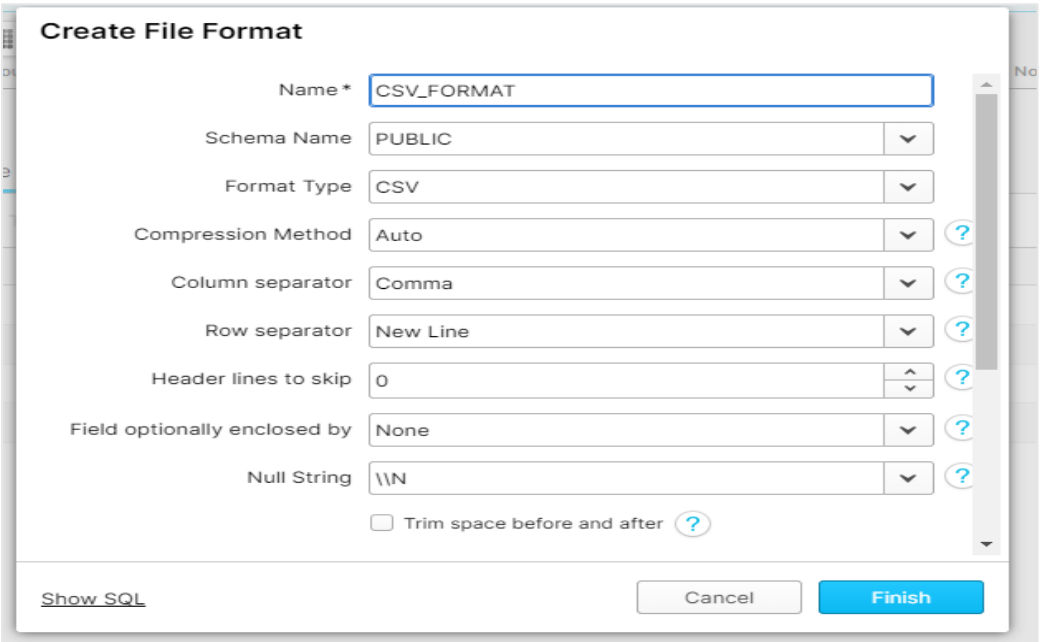

Figure 18: File format creation

Or create csv format via SQL code in snowflake snowsight

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

21]. Create a stage in snowflake pointing to your S3 bucket:

```
CREATE OR REPLACE STAGE RETAIL_STAGE
URL =  's3://sk-retail-bucket/'     --  (Name of your bucket)
FILE_FORMAT = RETAILCSV
STORAGE_INTEGRATION = s3_int_retail;
--- we can see previously created stage
SHOW STAGES;
```

22]. Create a SNOWPIPE with Auto-Ingest Enabled

```
CREATE OR REPLACE PIPE SNOWPIPE_DEMOGRAPHIC_RAW
AUTO_INGEST = TRUE
AS COPY INTO RETAIL.RETAIL_SCHEMA.DEMOGRAPHIC_RAW --table NAME DEMOGRAPHIC_RAW that you created in snowflake)
FROM @RETAIL_STAGE/DEMOGRAPHIC          ---------s3 bucket subfolder name
FILE_FORMAT = RETAILCSV;
```

23]. After creating snowpipe, get 'Notification Channel' value

Run command

```
Show pipes;
```

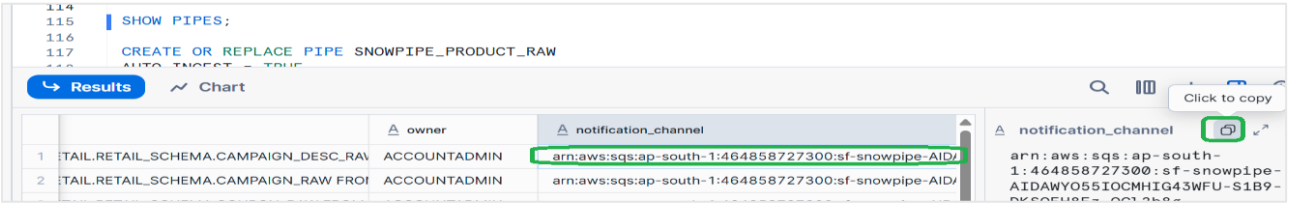

Figure 19: Copied event notification ARM

Or Go to Database Pipes

24]. Create an event on S3 bucket.

Go to your S3 bucket that you have created. Click on Properties tab and scroll down to

Event Notification -> Click Create Event Notification

Enter any name for the Notification.

Check All Object create Events

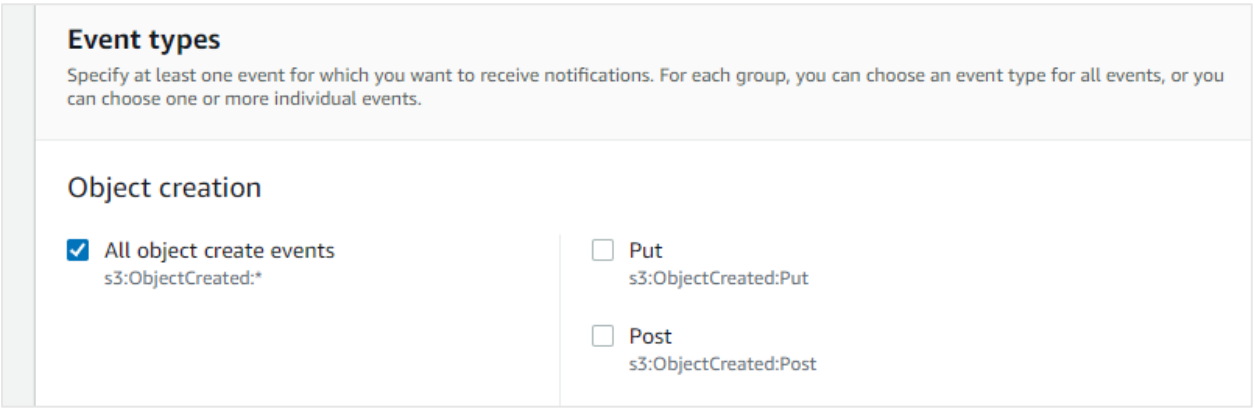

Figure 20: Create Event notification

Scroll down to Destination

Select SQS Queue ➔Select Enter SQS Queue ARN ➔ And paste that 'Notification Channel' under SQS Queue

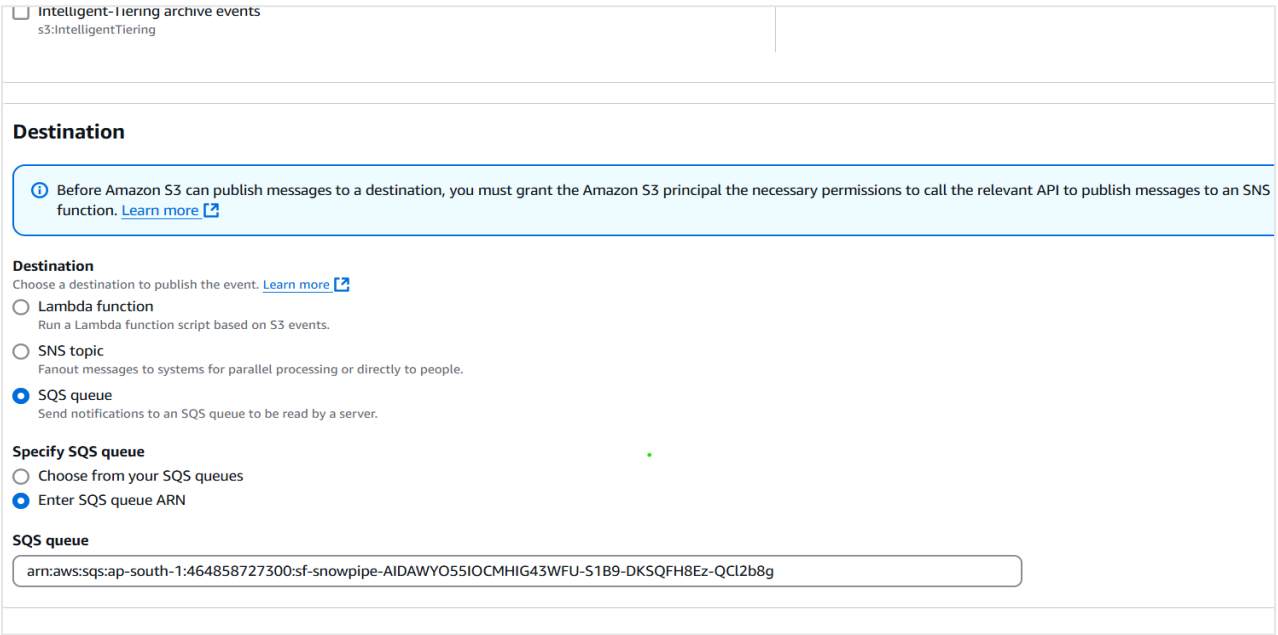

Figure 21: Paste notification ARN in SQS

Now you are ready to load the file to s3 bucket.

25]. Following are some snowpipe command which will help you to check snowpipe status

SELECT SYSTEM$PIPE_STATUS(' SNOWPIPE_DEMOGRAPHIC_RAW ');

-- REFRESH DATA IN IN BUCKET--

ALTER PIPE SNOWPIPE_DEMOGRAPHIC_RAW REFRESH;

```
ALTER PIPE SNOWPIPE_DEMOGRAPHIC_RAW REFRESH;
ALTER PIPE SNOWPIPE_CAMPAIGN_DESC_RAW REFRESH;
ALTER PIPE SNOWPIPE_CAMPAIGN_RAW REFRESH;
ALTER PIPE SNOWPIPE_PRODUCT_RAW REFRESH;
ALTER PIPE SNOWPIPE_COUPON_REDEMPT_RAW REFRESH;
ALTER PIPE SNOWPIPE_COUPON_RAW REFRESH;
ALTER PIPE SNOWPIPE_TRANSACTION_RAW REFRESH;
```

Results | Chart

	File	Status
1	/hh demographic.csv	SENT

Figure 22: refresh PIPE

Show Total records.

```
------TOTAL RECORDS IN TABLES--

SELECT COUNT(*) FROM DEMOGRAPHIC_RAW;
SELECT COUNT(*) FROM CAMPAIGN_DESC_RAW;
SELECT COUNT(*) FROM CAMPAIGN_RAW;
SELECT COUNT(*) FROM COUPON_RAW;
SELECT COUNT(*) FROM COUPON_REDEMPT_RAW;
SELECT COUNT(*) FROM PRODUCT_RAW;
SELECT COUNT(*) FROM TRANSACTION_RAW;
```

Results | Chart

	COUNT(*)
1	2500

Figure 23: Total records

In this we successfully load all data in the Snowflake.