Steps to Integrate AWS and Snowflake

Craeted By:
Anngrah Dhar

Created By: Anngrah Dhar
Last update by: Anngrah Dhar
Last updated on: 17/05/2023

Table of Contents

Objective

To fetch data from Amazon S3 bucket through Snwoflake

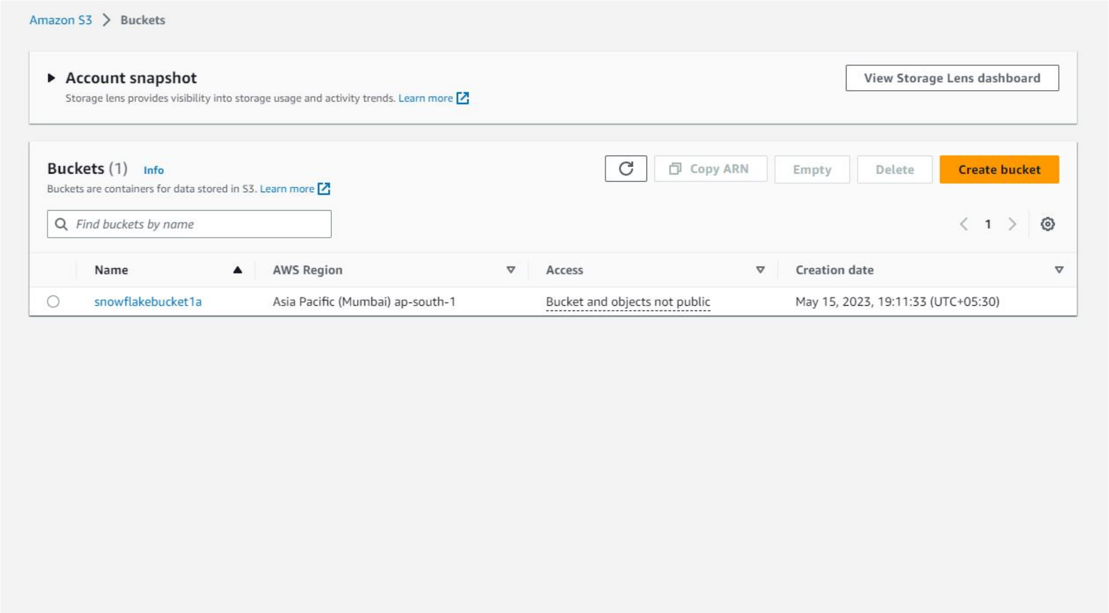
Prerequisites

AWS Service account needs to be set up.
Snowflake should also be set up.

Procedure 1: Integrating connection between Snowflake and AWS

Step 1:

Creation of an **AWS S3 bucket**.



▸ Account snapshot

Storage lens provides visibility into storage usage and activity trends. Learn more

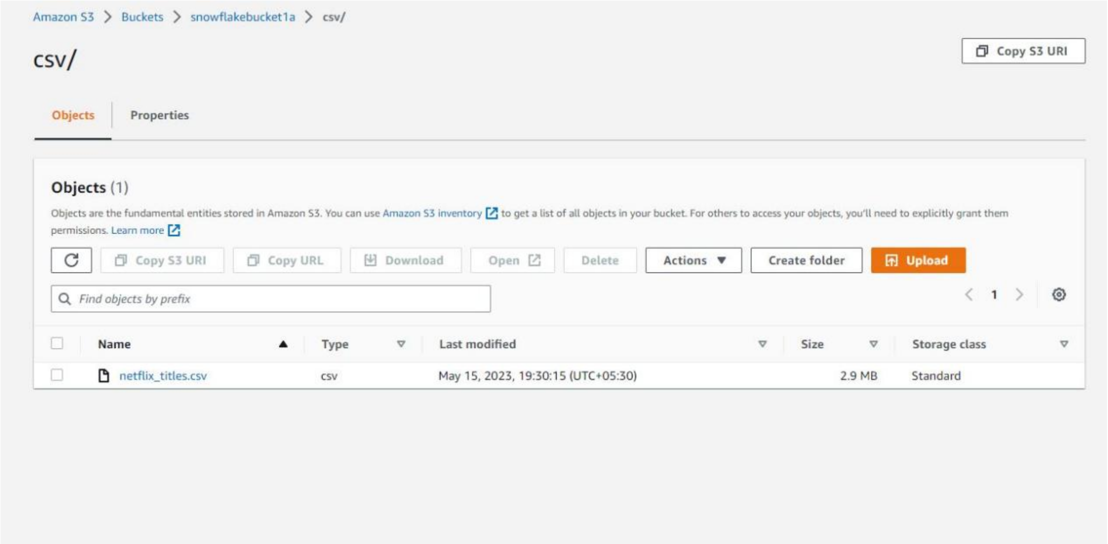
Buckets (1)

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	snowflakebucket1a	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	May 15, 2023, 19:11:33 (UTC+05:30)

Step 2:

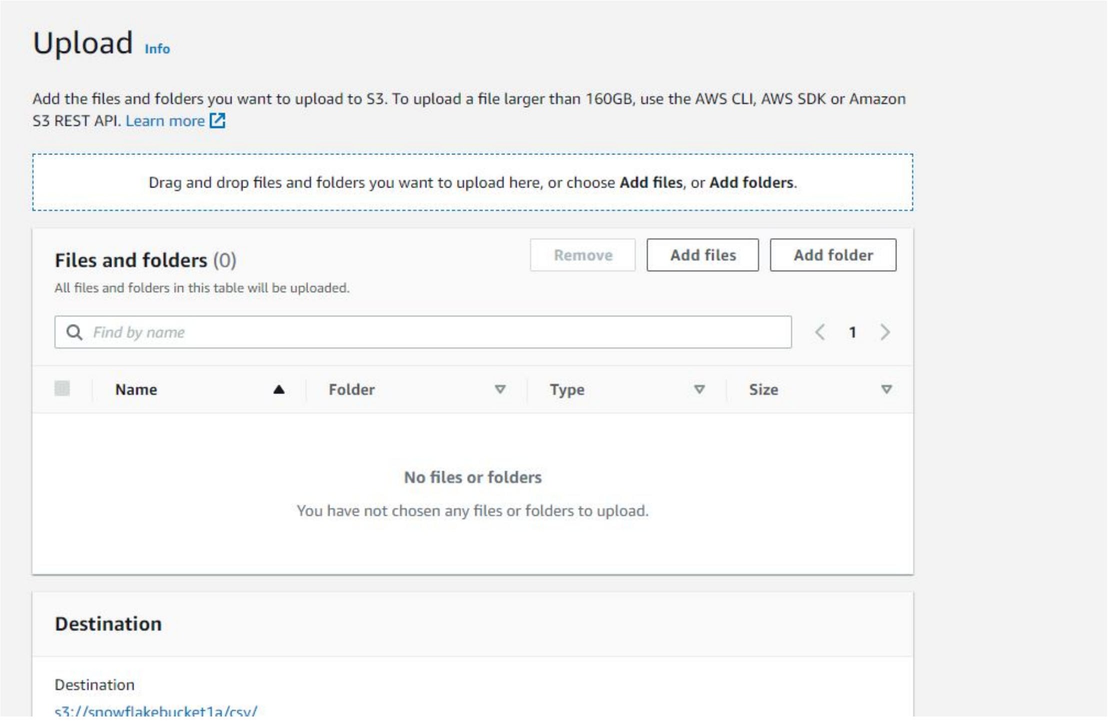
Inside the bucket create a folder and name it as **csv** and save it.
Then import the **csv file** in it.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	netflix_titles.csv	csv	May 15, 2023, 19:30:15 (UTC+05:30)	2.9 MB	Standard

Click on **upload**



Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

All files and folders in this table will be uploaded.

No files or folders

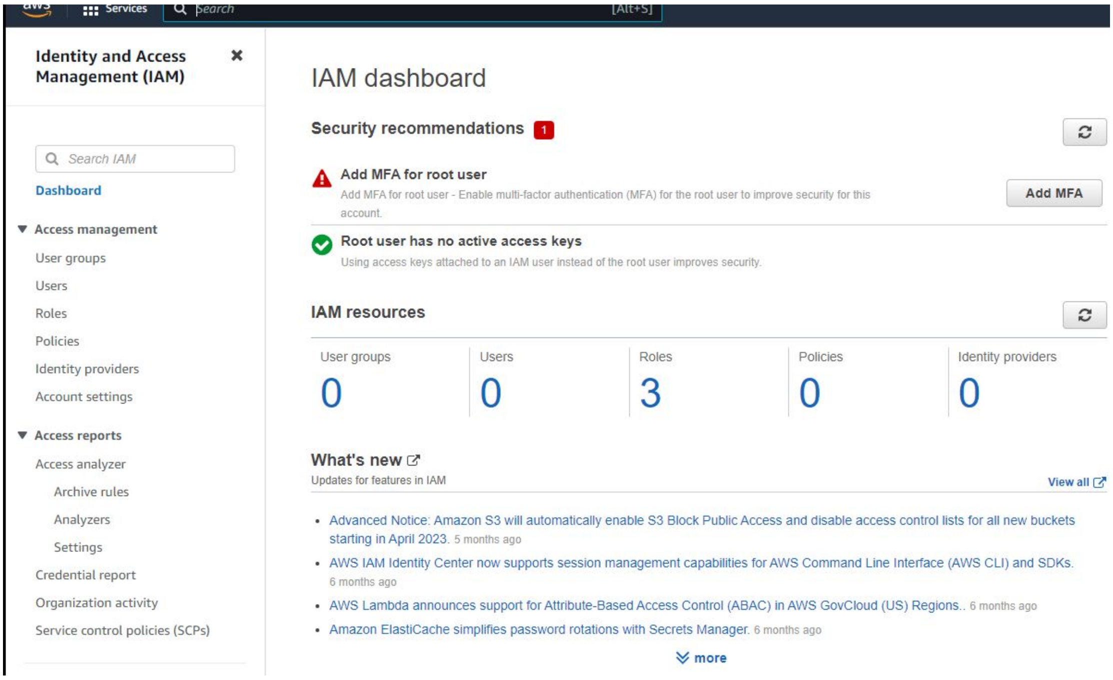
You have not chosen any files or folders to upload.

Destination

s3://snowflakebucket1a/csv/

Click on **Add files** and then upload the file and scroll down and upload it.
Click on **close.**

Step 3:
In AWS, search **IAM service.**
And then select **Roles** on the left side.



Services Search [Alt+S]

Identity and Access Management (IAM)

Search IAM

Dashboard

- Access management
  - User groups
  - Users
  - Roles
  - Policies
  - Identity providers
  - Account settings
- Access reports
  - Access analyzer
    - Archive rules
    - Analyzers
    - Settings
  - Credential report
  - Organization activity
  - Service control policies (SCPs)

IAM dashboard

Security recommendations 1

Add MFA for root user
Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.
Add MFA

Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.

IAM resources

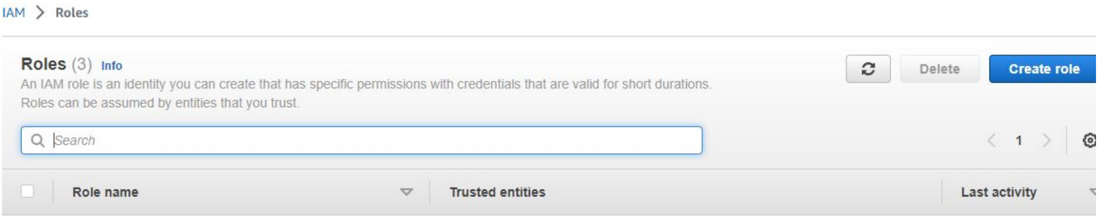
User groups	Users	Roles	Policies	Identity providers
0	0	3	0	0

What's new
Updates for features in IAM
View all

- Advanced Notice: Amazon S3 will automatically enable S3 Block Public Access and disable access control lists for all new buckets starting in April 2023. 5 months ago
- AWS IAM Identity Center now supports session management capabilities for AWS Command Line Interface (AWS CLI) and SDKs. 6 months ago
- AWS Lambda announces support for Attribute-Based Access Control (ABAC) in AWS GovCloud (US) Regions.. 6 months ago
- Amazon ElastiCache simplifies password rotations with Secrets Manager. 6 months ago

more

Click on **Create Role**



IAM > Roles

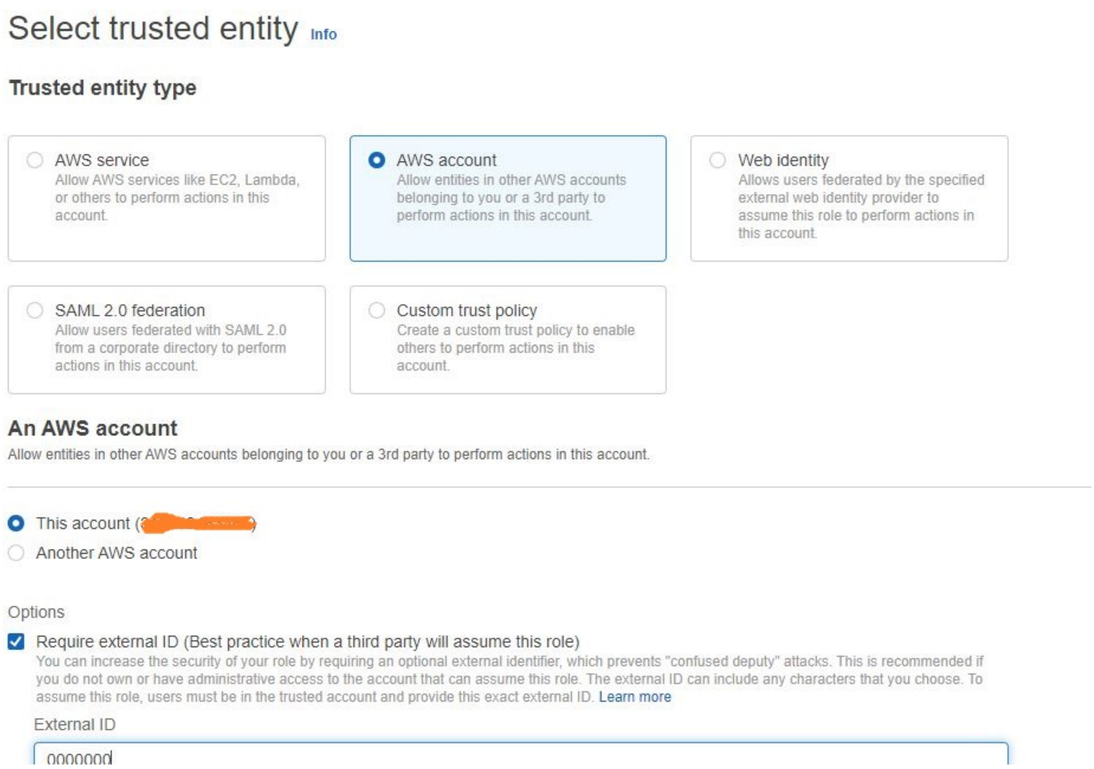
Roles (3) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Delete Create role

Search

□	Role name	Trusted entities	Last activity

Select **AWS Account** and your id would be selected and for the the time being put **External ID** as 0000.



Select trusted entity Info

Trusted entity type

○ AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

● AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

An AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

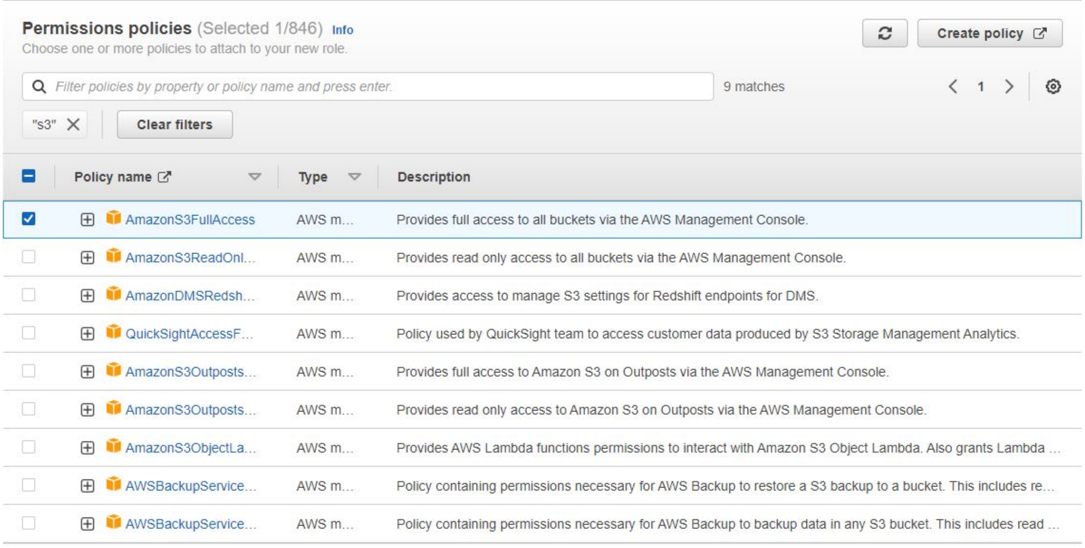
● This account ([redacted])
○ Another AWS account

Options

☑ Require external ID (Best practice when a third party will assume this role)
You can increase the security of your role by requiring an optional external identifier, which prevents "confused deputy" attacks. This is recommended if you do not own or have administrative access to the account that can assume this role. The external ID can include any characters that you choose. To assume this role, users must be in the trusted account and provide this exact external ID. Learn more

External ID
0000000

Now in the search option search **s3** and select the option **AmazonS3FullAccess** and then next.



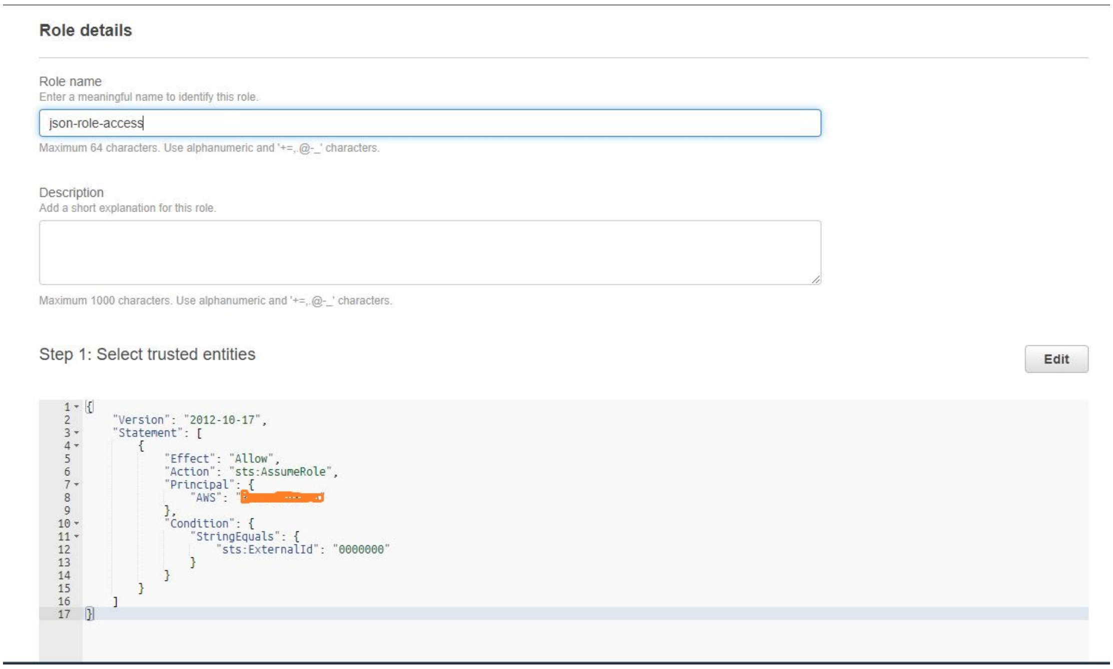
Permissions policies (Selected 1/846) Info
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter. 9 matches

"s3" ✕ Clear filters

☐	Policy name	Type	Description
☑	AmazonS3FullAccess	AWS m...	Provides full access to all buckets via the AWS Management Console.
☐	AmazonS3ReadOnl...	AWS m...	Provides read only access to all buckets via the AWS Management Console.
☐	AmazonDMSRedsh...	AWS m...	Provides access to manage S3 settings for Redshift endpoints for DMS.
☐	QuickSightAccessF...	AWS m...	Policy used by QuickSight team to access customer data produced by S3 Storage Management Analytics.
☐	AmazonS3Outposts...	AWS m...	Provides full access to Amazon S3 on Outposts via the AWS Management Console.
☐	AmazonS3Outposts...	AWS m...	Provides read only access to Amazon S3 on Outposts via the AWS Management Console.
☐	AmazonS3ObjectLa...	AWS m...	Provides AWS Lambda functions permissions to interact with Amazon S3 Object Lambda. Also grants Lambda ...
☐	AWSBackupService...	AWS m...	Policy containing permissions necessary for AWS Backup to restore a S3 backup to a bucket. This includes re...
☐	AWSBackupService...	AWS m...	Policy containing permissions necessary for AWS Backup to backup data in any S3 bucket. This includes read ...

Now give the role a **name.** Below will the script where AWS(**your account number)** and **External Id** can be found. And then click on **create role**.



Role details

Role name
Enter a meaningful name to identify this role.
json-role-access
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {
                "AWS": "[redacted]
            },
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "0000000"
                }
            }
        }
    ]
}
```

You will now find that role is created.

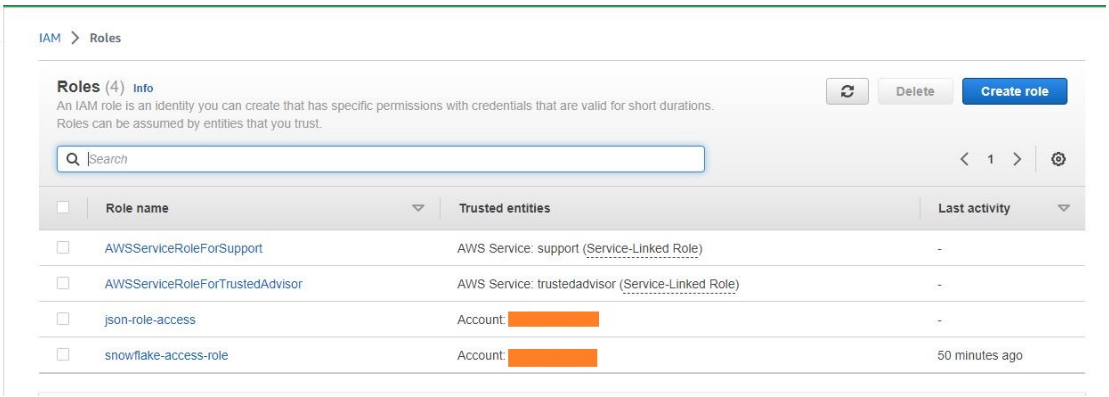
Step 4:
Now navigate to Snowflake
Open a new worksheet and type the following command:

a. We will first create the storage integration object which we will use to get permission from the aws to access the aws s3 bucket

```
create or replace storage integration s3_int
  TYPE = EXTERNAL_STAGE
  STORAGE_PROVIDER = S3
  ENABLED = TRUE
  STORAGE_AWS_ROLE_ARN = ''
  STORAGE_ALLOWED_LOCATIONS = ('s3://snowflakebucket1a/csv/')
   COMMENT = 'This an optional comment'
```

You can find **STORAGE_AWS_ROLE_ARN** value from Role which you created by copying the value of **ARN.**
You can find it here:

IAM -> Roles -> <Role Created> and ARN will on the screen

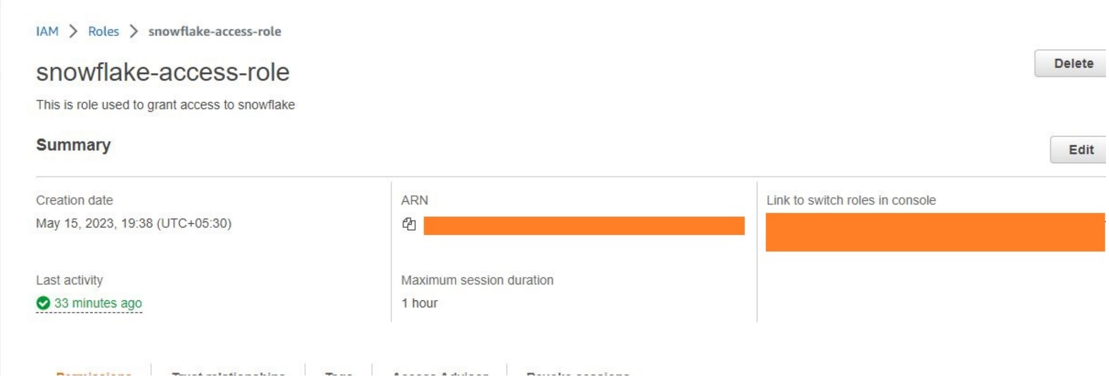


IAM > Roles

Roles (4) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
json-role-access	Account: [redacted]	-
snowflake-access-role	Account: [redacted]	50 minutes ago



IAM > Roles > snowflake-access-role

snowflake-access-role

This is role used to grant access to snowflake

Summary

Creation date
May 15, 2023, 19:38 (UTC+05:30)

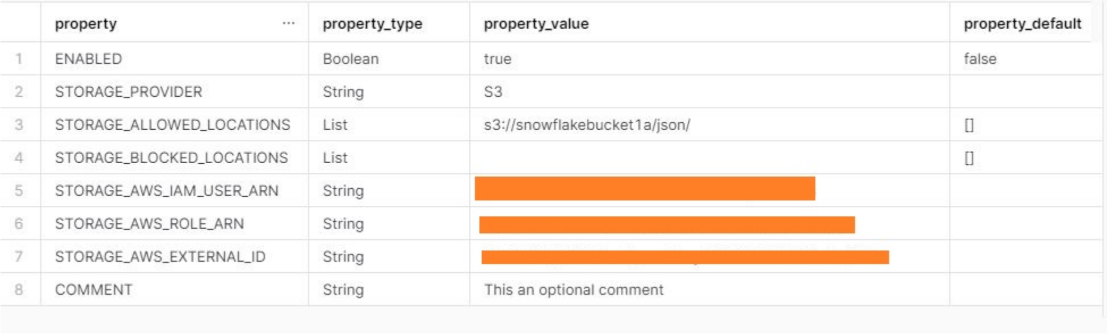
ARN
[redacted]

Link to switch roles in console
[redacted]

Last activity
33 minutes ago

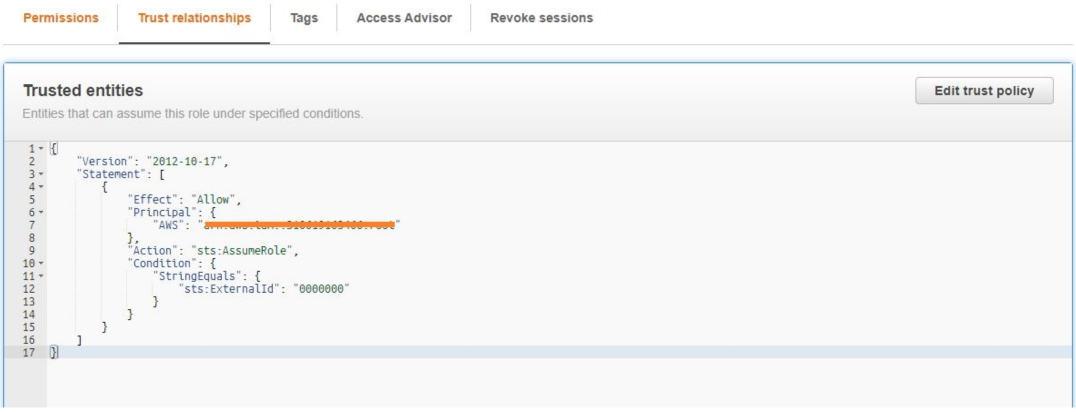
Maximum session duration
1 hour

b. We will then list the description of storage object created
DESC integration s3_int;



	property	property_type	property_value	property_default
1	ENABLED	Boolean	true	false
2	STORAGE_PROVIDER	String	S3	
3	STORAGE_ALLOWED_LOCATIONS	List	s3://snowflakebucket1a/json/	[]
4	STORAGE_BLOCKED_LOCATIONS	List		[]
5	STORAGE_AWS_IAM_USER_ARN	String	[redacted]	
6	STORAGE_AWS_ROLE_ARN	String	[redacted]	
7	STORAGE_AWS_EXTERNAL_ID	String	[redacted]	
8	COMMENT	String	This an optional comment	

From the given output, copy the value of **STORAGE_AWS_IAM_USER_ARN** and **STORAGE_AWS_EXTERNAL_ID** from the output from snowflake to the trust policy which we created earlier while setting the role.
Click on **edit trust policy** and save it.



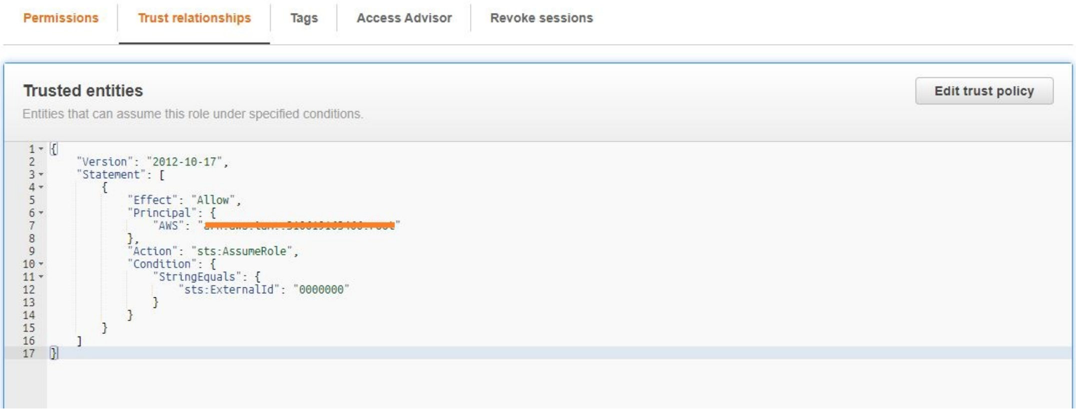
Trusted entities

Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "[redacted]"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "0000000"
                }
            }
        }
    ]
}
```

You can find the trust policy in

IAM -> Roles -> <role created> -> Scroll down and click on **Trust Relationships** tab.
And edit the trust policy.



Trusted entities

Entities that can assume this role under specified conditions.

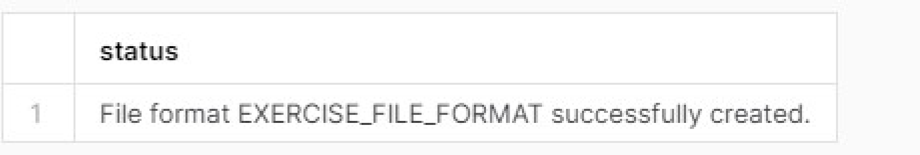
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "[redacted]"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "0000000"
                }
            }
        }
    ]
}
```

Procedure 2: It is where we work with the imported data from AWS S3 bucket

Step 1: Now navigate back to Snowflake into the same worksheet where we were working and run the following commands

a. Now we create a file format to set the type of format, headings, delimiter, etc

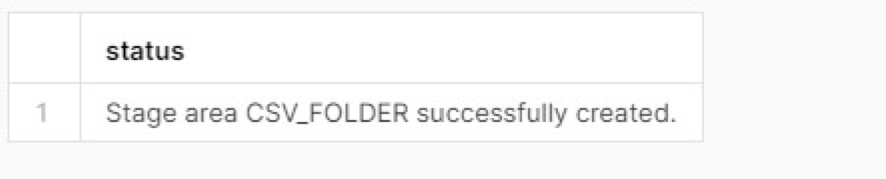
```
CREATE OR REPLACE FILE FORMAT
EXERCISE_DB.FILE_FORMAT.exercise_file_format
type=csv field_delimiter=',' skip_header=1;
```



	status
1	File format EXERCISE_FILE_FORMAT successfully created.

b. Now we create a stage object which we will use to access the csv files.

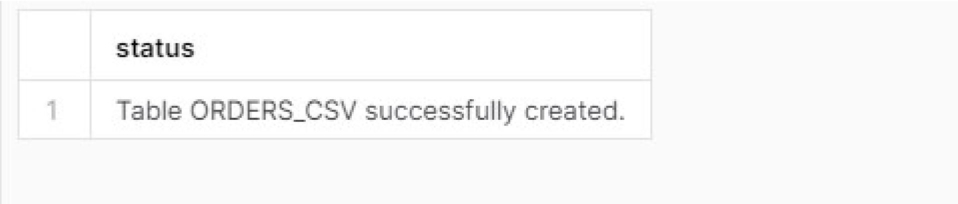
```
CREATE OR REPLACE stage MANAGE_DB.external_stages.csv_folder
   URL = 's3://snowflakebucket1a/csv/'
  STORAGE_INTEGRATION = s3_int
```



	status
1	Stage area CSV_FOLDER successfully created.

c. We will now create a table in which we will add the csv file data from bucket. The number of columns and data type of the table and csv needs to be the same.

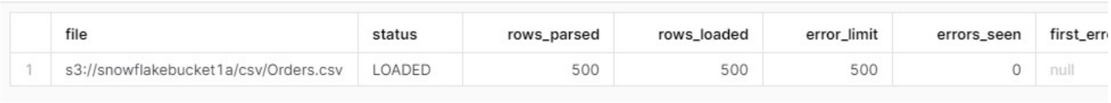
```
CREATE OR REPLACE TABLE EXERCISE_DB.PUBLIC.Orders_csv(
   Order_ID string,
   Order_Date       string,
   Customer_Name string,
   State string,
   City string
);
```



	status
1	Table ORDERS_CSV successfully created.

d. Now we will use the **COPY** command to insert rows from csv to the table which was created in the previous step

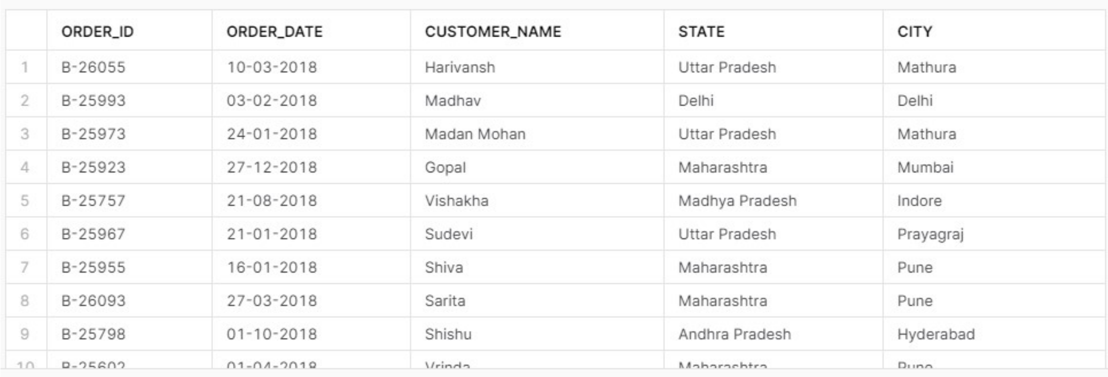
```
COPY INTO EXERCISE_DB.PUBLIC.Orders_csv
FROM @MANAGE_DB.external_stages.csv_folder
FILE_FORMAT =
(FORMAT_NAME=EXERCISE_DB.FILE_FORMAT.exercise_file_format)
FILES = ('Orders.csv')
ON_ERROR='CONTINUE'
TRUNCATECOLUMNS = true
SIZE_LIMIT=25000;
```



	file	status	rows_parsed	rows_loaded	error_limit	errors_seen	first_err
1	s3://snowflakebucket1a/csv/Orders.csv	LOADED	500	500	500	0	null

e. Now use the select query to check the data

```
SELECT * FROM EXERCISE_DB.PUBLIC.Orders_csv;
```



	ORDER_ID	ORDER_DATE	CUSTOMER_NAME	STATE	CITY
1	B-26055	10-03-2018	Harivansh	Uttar Pradesh	Mathura
2	B-25993	03-02-2018	Madhav	Delhi	Delhi
3	B-25973	24-01-2018	Madan Mohan	Uttar Pradesh	Mathura
4	B-25923	27-12-2018	Gopal	Maharashtra	Mumbai
5	B-25757	21-08-2018	Vishakha	Madhya Pradesh	Indore
6	B-25967	21-01-2018	Sudevi	Uttar Pradesh	Prayagraj
7	B-25955	16-01-2018	Shiva	Maharashtra	Pune
8	B-26093	27-03-2018	Sarita	Maharashtra	Pune
9	B-25798	01-10-2018	Shishu	Andhra Pradesh	Hyderabad
10	B-25602	01-04-2018	Vrinda	Maharashtra	Pune