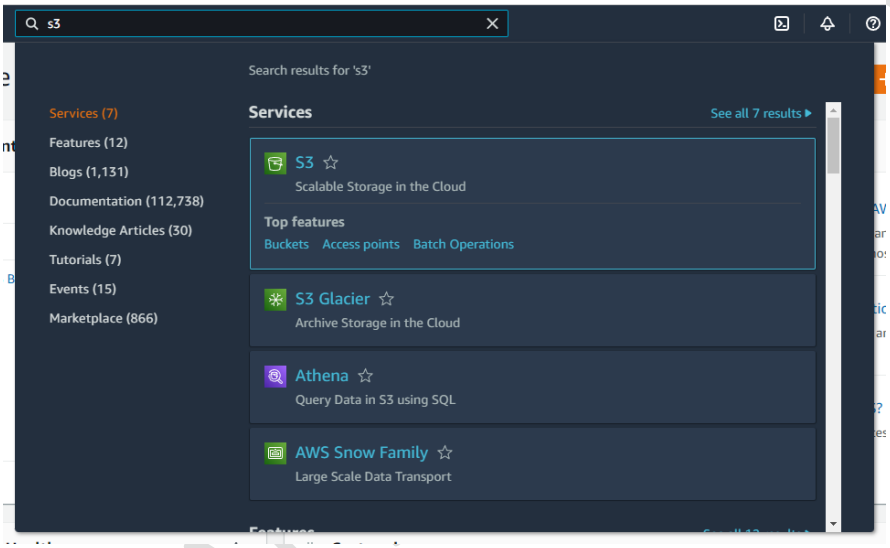
SNOWFLAKE CONTINUOUS DATA LOADING

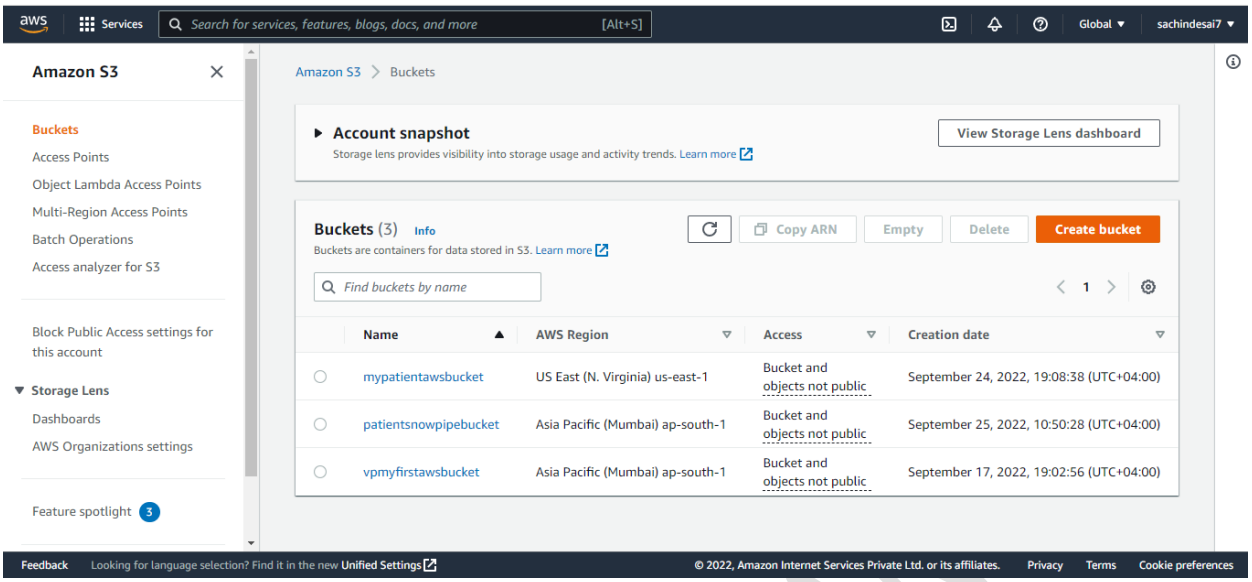
1]. Create an AWS account in aws.amazon.com

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

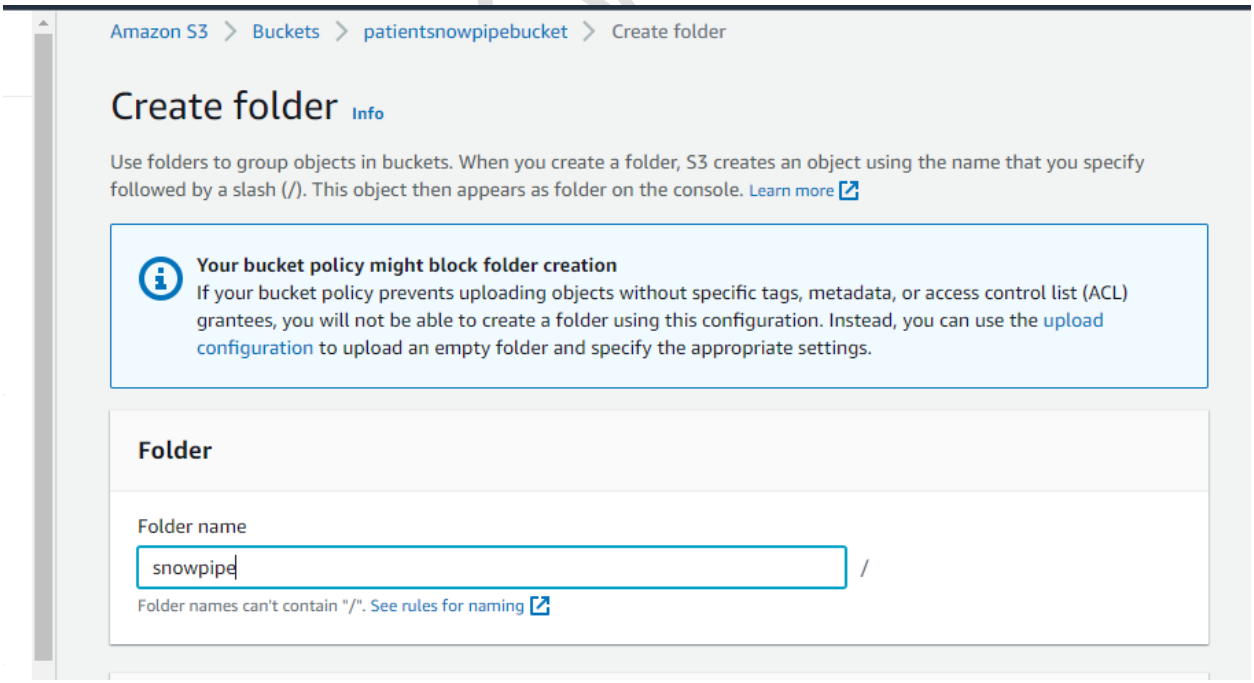
3]. Go to the Console home and search for S3 (Simple Storage Service) and click on it.

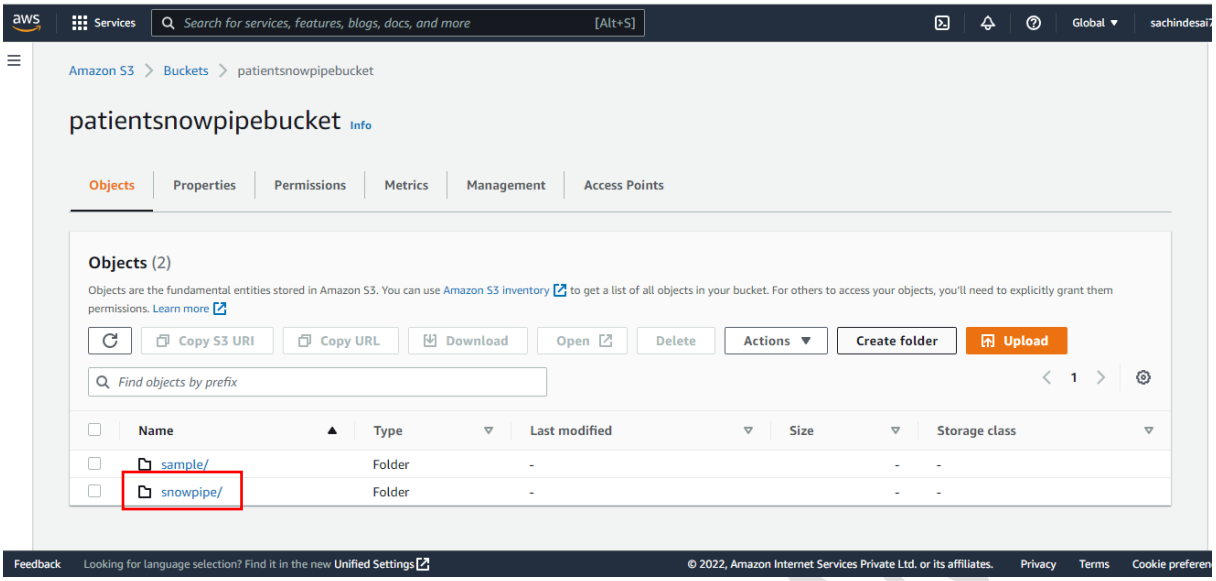


4]. Create S3 bucket

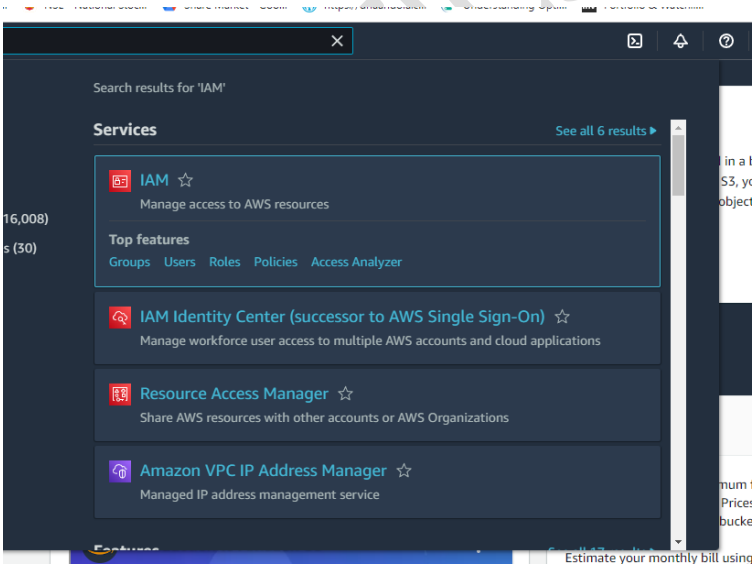


5]. Create a folder inside the bucket (e.g. snowpipe)

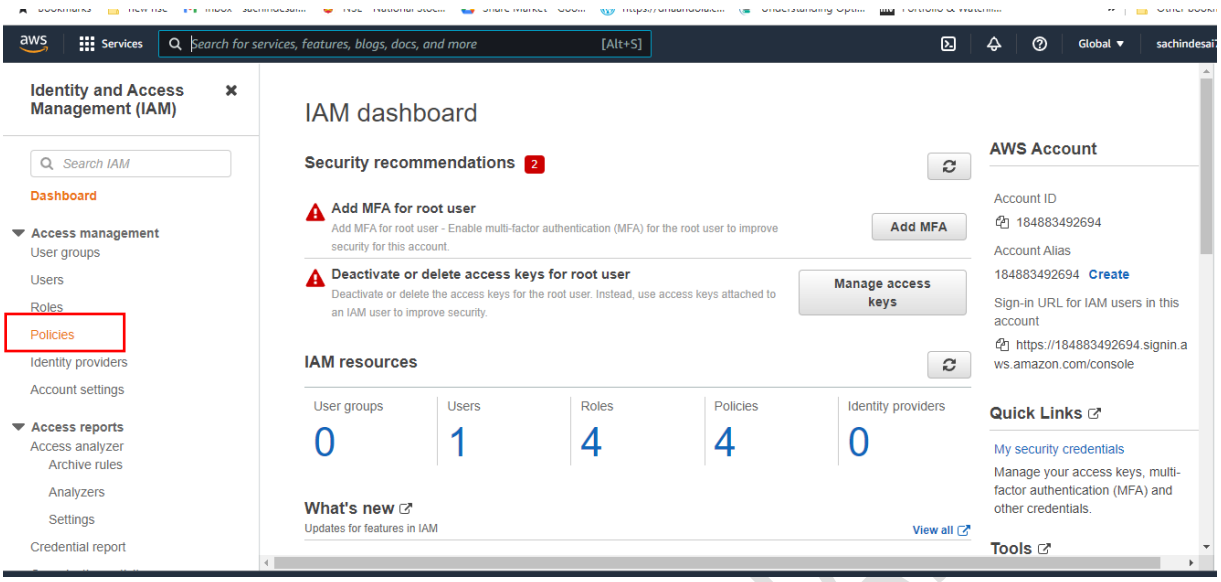




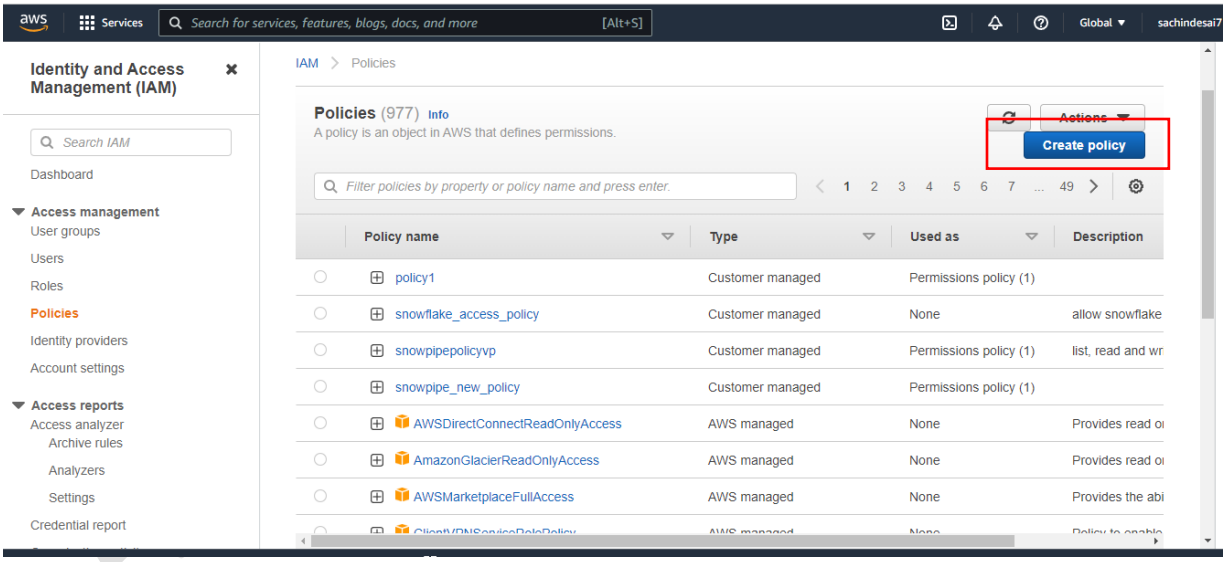
6]. Once the S3 bucket and folder are created, search and select the IAM (Identity and Access Management) service from the AWS console.



7]. Click on the Policies from IAM Dashboard



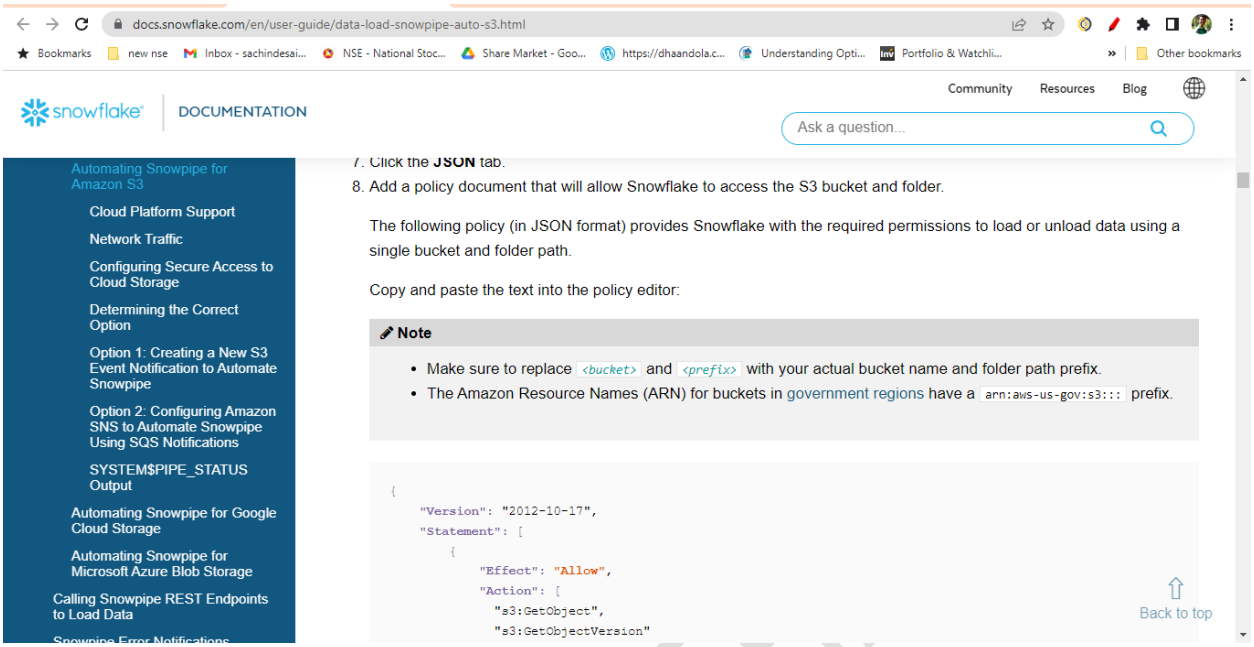
8]. Create IAM policy for the bucket by clicking on the “Create Policy” button



9]. Click on the JSON tab and replace the existing text with the text given in the reference Document (https://docs.snowflake.com/en/user-guide/data-load-snowpipe-auto-s3.html).

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

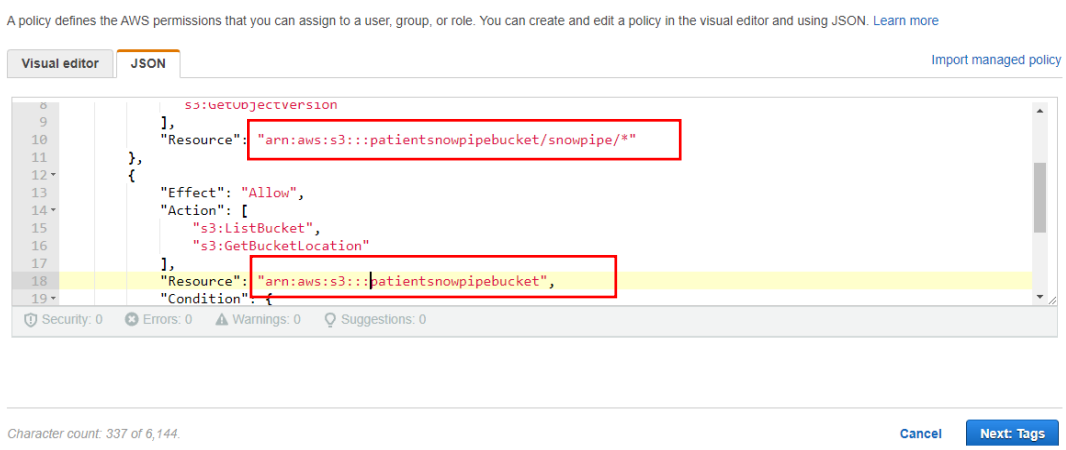
(It is under the step no. 8 from the document)



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

Also set the S3:prefix to " *"

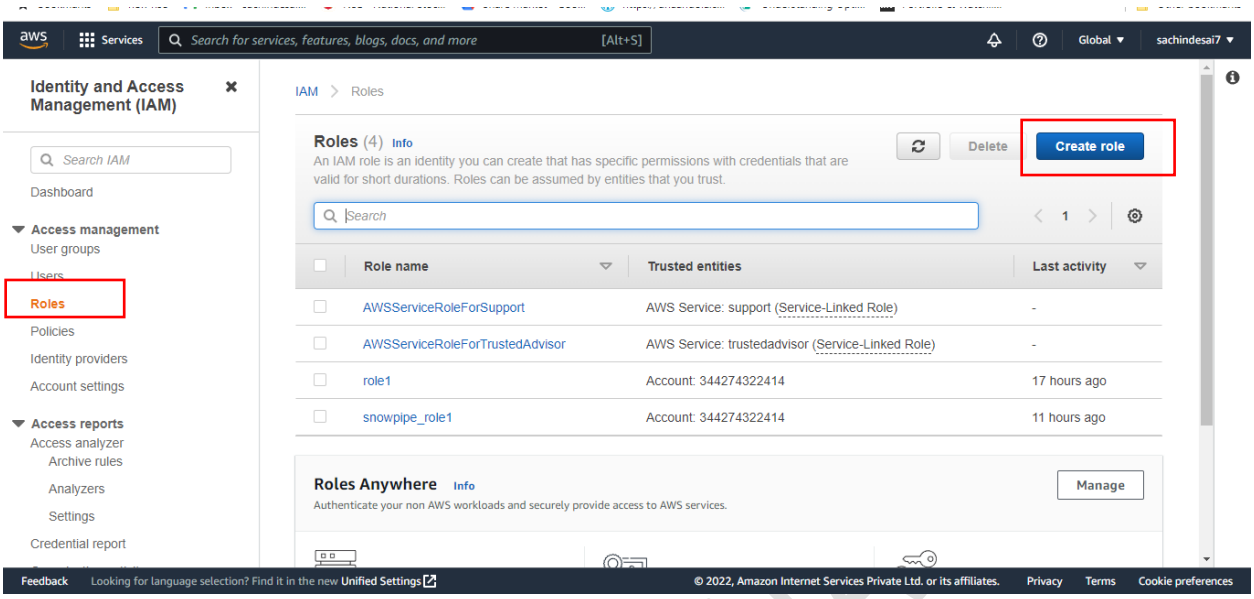
```
"s3:prefix":[
            "*"
```



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

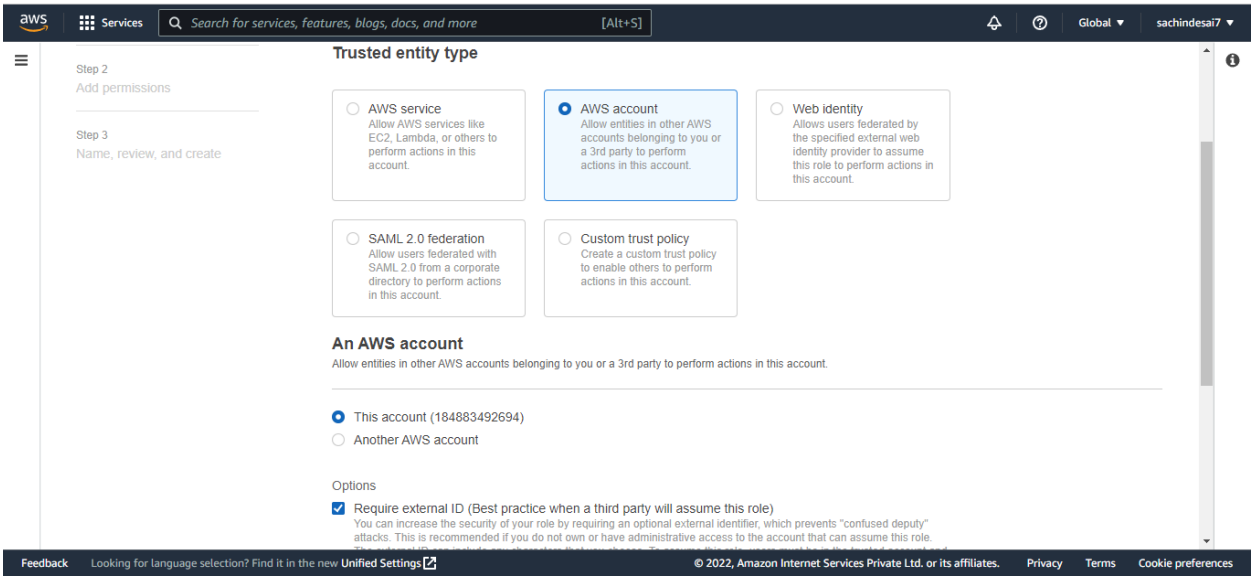
Your policy will get created.

12]. Create IAM Role. Click on Create Role

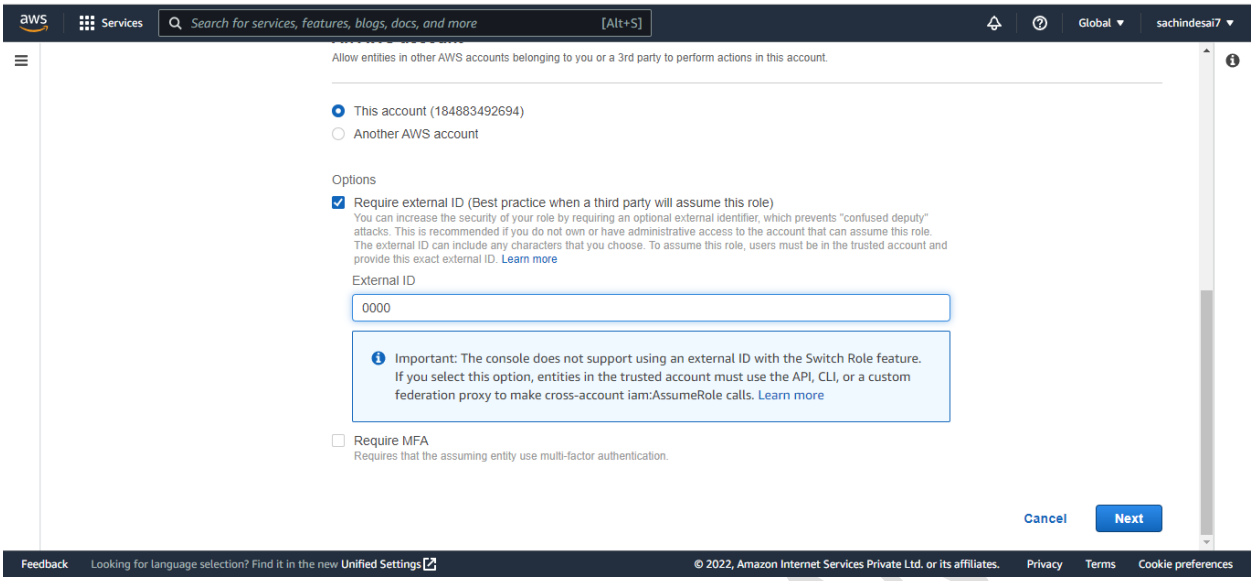


13]. Select AWS Account from Trusted Entity Type.

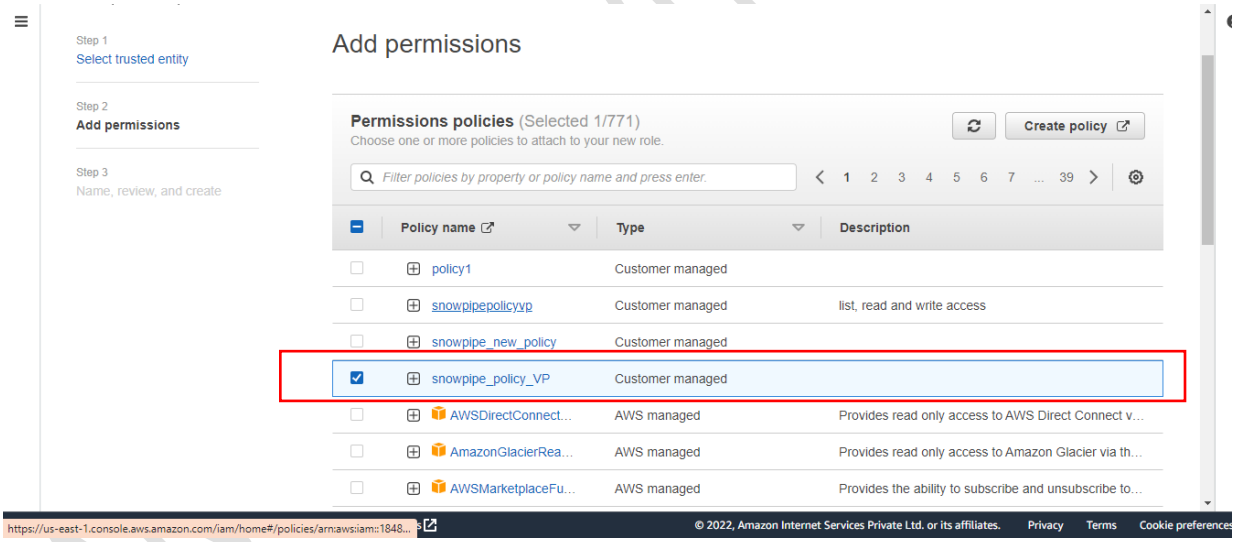
You will get your account number selected by default when you select AWS account.



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

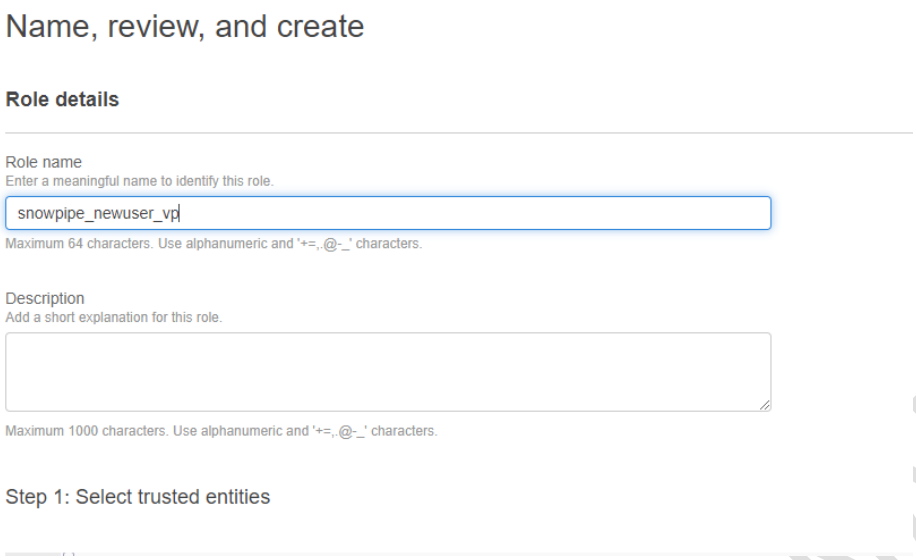


15]. On the next page, Select the IAM policy that you have created

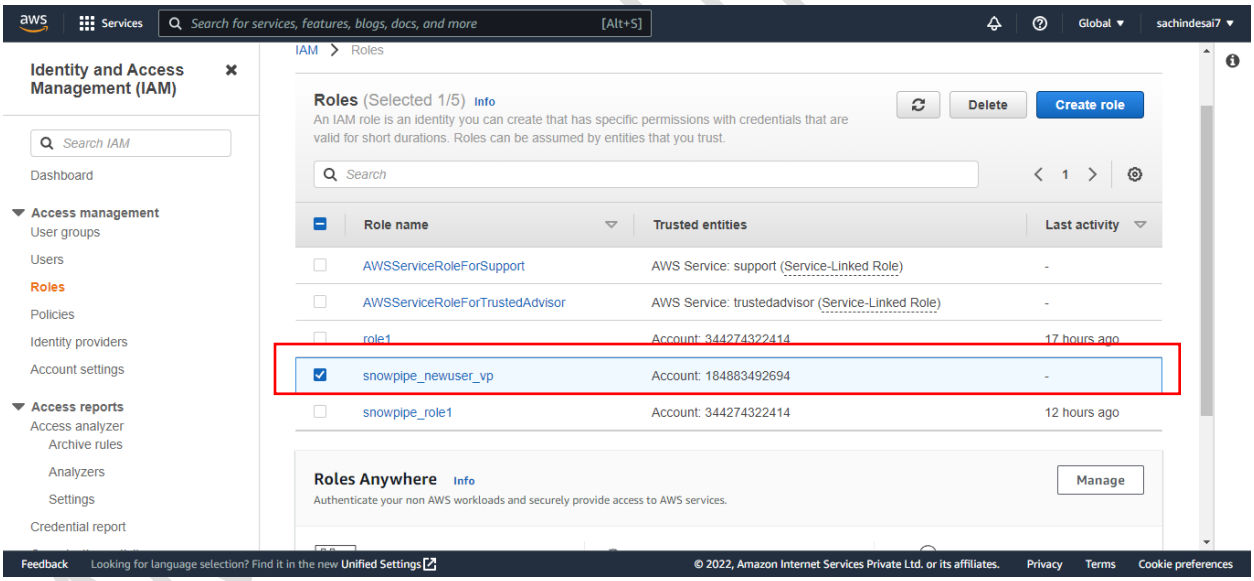


16]. On the next page Enter any unique name to the role you are creating. The description is optional.

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

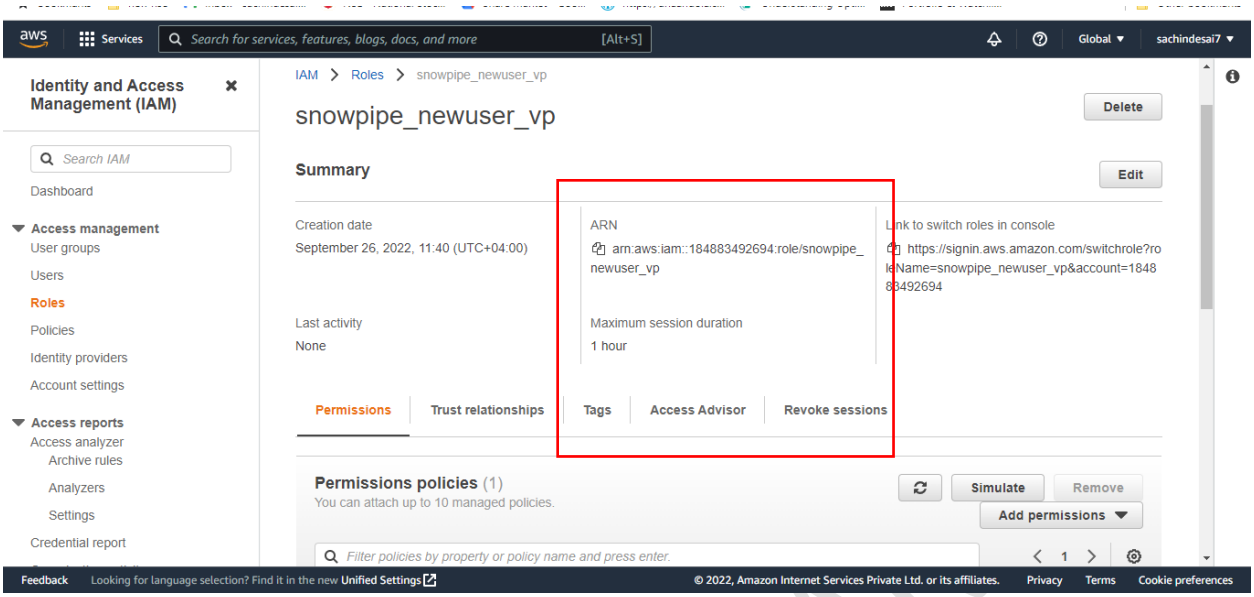


17]. Click on the role that you have created. It will show you the summary page.



You will get the following window

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



18]. Login to the Snowflake Account.

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

CREATE OR REPLACE STORAGE INTEGRATION snowpipe_integration

TYPE = external_stage

STORAGE_PROVIDER = s3

STORAGE_AWS_ROLE_ARN = 'arn:aws:iam::184883492694:role/snowpipe_newuser_vp'

ENABLED = true

STORAGE_ALLOWED_LOCATIONS = ('*');

19]. In Snowflake worksheet run command

Desc integration integration_name;

e.g. desc integration snowpipe_integration;

And Note down the STORAGE_AWS_IAM_USER_ARN and STORAGE_AWS_EXTERNAL_ID from the result set

5	STORAGE_AWS_IAM_USER_ARN	String	arn:aws:iam::344274322414:user/eyn10000-s
7	STORAGE_AWS_EXTERNAL_ID	String	BR03385_SFCRole=2_4ZIeqwTLkI5mYMphp6kTX3D9FKQ=

20]. Now go to the AWS Console

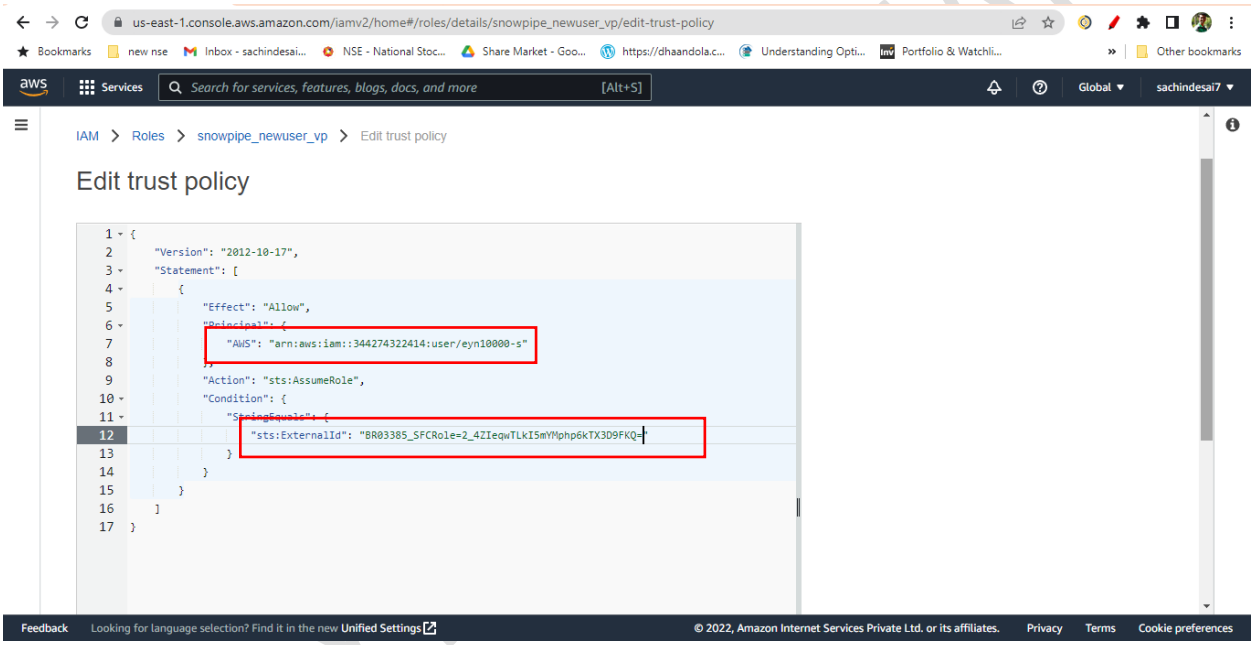
IAM ⍰ Role

Select the role you created

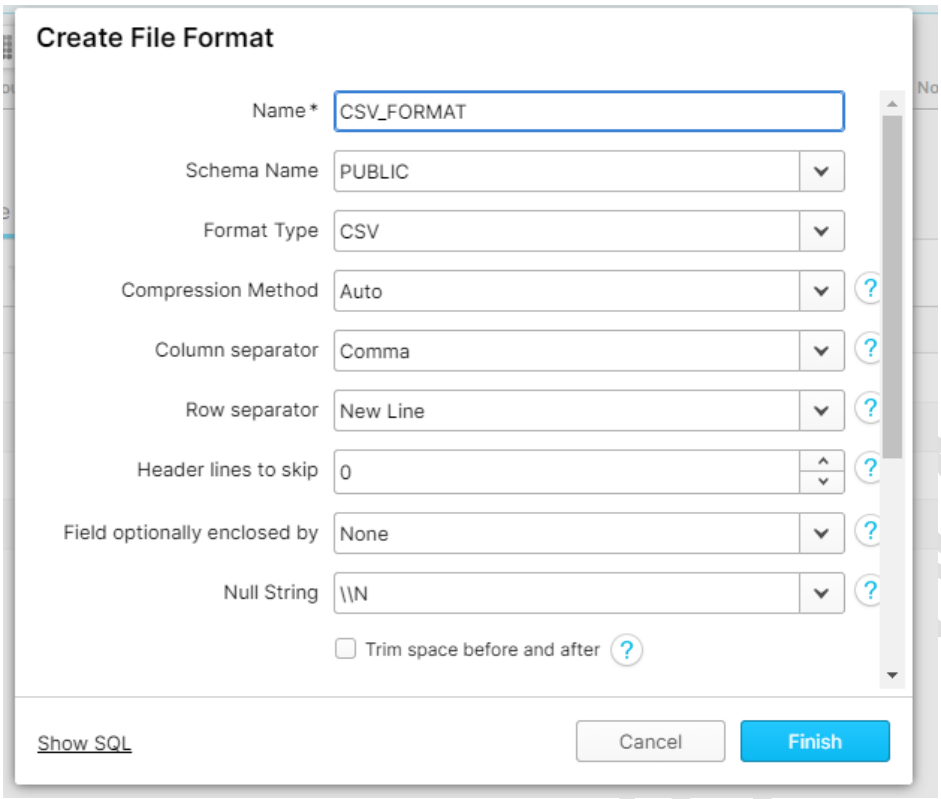
Click Trust Relationships -> Edit trust relationship

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

Click Update Policy



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



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

CREATE OR REPLACE STAGE patient_snowpipe_stage

STORAGE_INTEGRATION = snowpipe_integration

URL = 's3://patientsnowpipebucket/snowpipe' -- (Name of your bucket and folder)

FILE_FORMAT = (format_name = ' CSV_FORMAT');

23]. Now Create auto-ingest pipe.

CREATE OR REPLACE PIPE patient_snowpipe

AUTO_INGEST = TRUE

AS COPY INTO tab_patient -- (table name that you created in snowflake)

FROM @patient_snowpipe_stage -- (name of the stage)

FILE_FORMAT = (FORMAT_NAME = 'CSV_FORMAT');

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

Run command

Show pipes;

name	database_name	schema_name	definition	owner	notification_channel
DEMO1_SNOWPIPE	VP_DEMODA...	PUBLIC	COPY INTO ...	ACCOUNTA...	arn:aws:sqs:ap-south-1:344274322414:sf-snowpipe-AIDAVAKCZIPXGQXWUHIMU-M-ASvzXErhxxGpKYm5xGMA
PATIENT_SNOWPIPE	VP_DEMODA...	PUBLIC	copy into ta...	ACCOUNTA...	arn:aws:sqs:ap-south-1:344274322414:sf-snowpipe-AIDAVAKCZIPXGQXWUHIMU-M-ASvzXErhxxGpKYm5xGMA

Or Go to Database ⍰ Pipes

Here also you will get the notification channel value.

Databases > VP_DEMODATABASE

Tables Views Schemas Stages File Formats Sequences Pipes

Create Drop Transfer Ownership

Search Pipes

Pipe Name	Schema	↓ Creation Time	Owner	Notification Channel	Comment
PATIENT_SNOWPIPE	PUBLIC	9/25/2022, 11:20:31...	ACCOUNTADMIN	arn:aws:sqs:ap-south-1:344274322414:sf-snow...	
DEMO1_SNOWPIPE	PUBLIC	9/25/2022, 5:34:16 ...	ACCOUNTADMIN	arn:aws:sqs:ap-south-1:344274322414:sf-snow...	

25]. This is the final step. 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.

Amazon S3 > Buckets > patientsnowpipebucket > Create event notification

Create event notification Info

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

General configuration

Event name

demo1_notification

Event name can contain up to 255 characters.

Prefix - *optional*

Limit the notifications to objects with key starting with specified characters.

images/

Suffix - *optional*

Limit the notifications to objects with key ending with specified characters.

.jpg

Check All Object create Events

Event types

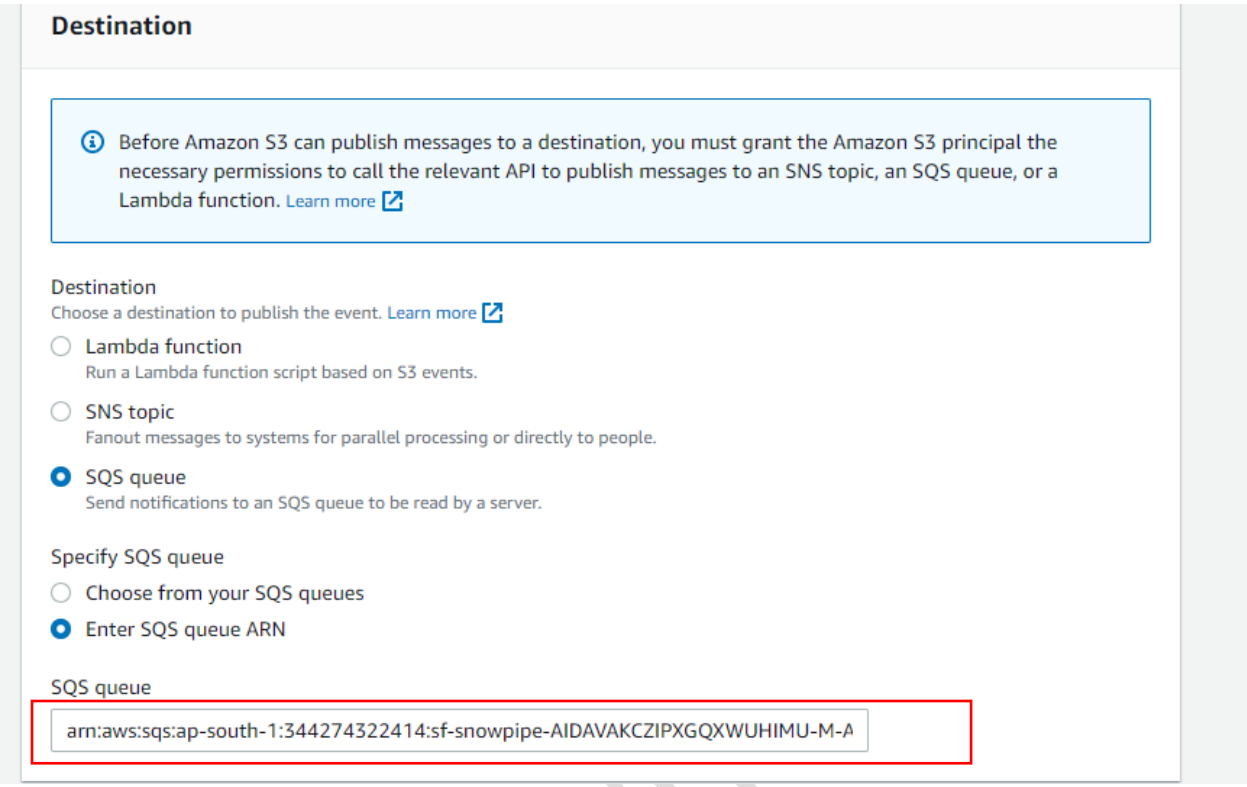
Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation

- [x] All object create events
  s3:ObjectCreated:*
- [ ] Put
  s3:ObjectCreated:Put
- [ ] Post
  s3:ObjectCreated:Post

Scroll down to Destination

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



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

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

```
select SYSTEM$PIPE_STATUS('patient_snowpipe');
```

```
select * from table(information_schema.copy_history(table_name=>'tab_patient', start_time=>
dateadd(hours, -1, current_timestamp())));
```