AWS Batch Data Pipeline

Objective: To create a Batch Data pipeline in AWS by using the following services

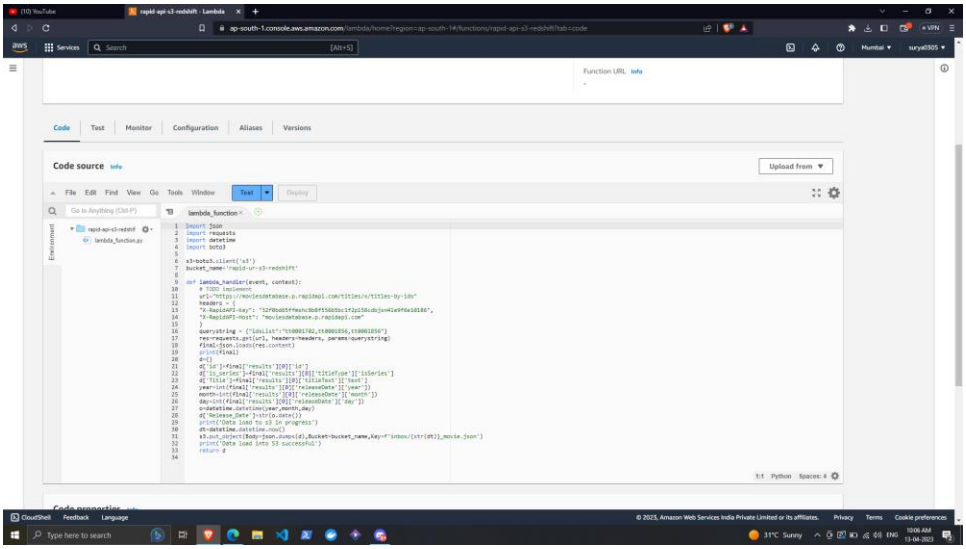
- Lambda
- S3
- Glue
- Redshift Serverless
- Event Bridge

The Data is pulled from one of the Rapid API

Creation of Lambda Function

I had used a Movie Database API from RapidAPI for the free version it can only provide 2 records.

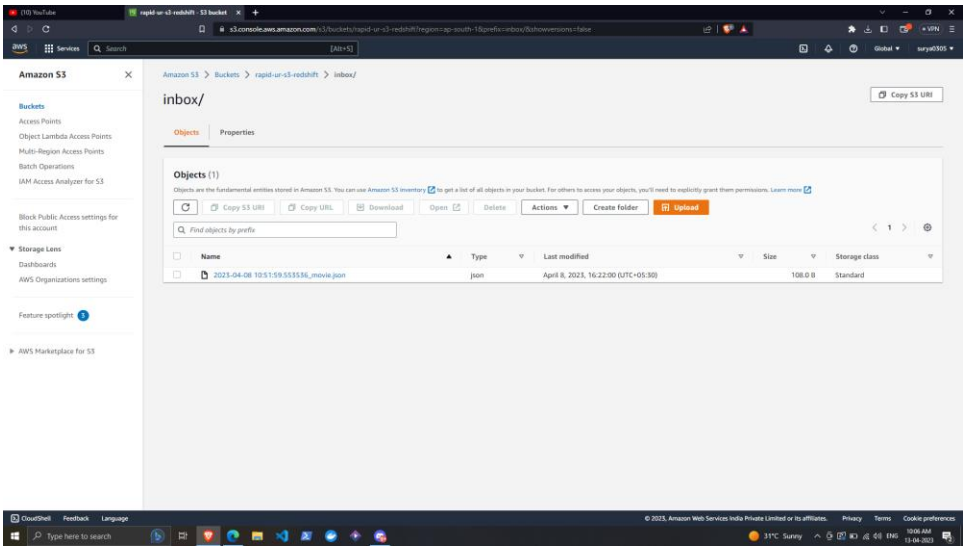
First, I had created a lambda function in which it calls the API and generates data and using pandas and JSON libraries had generated required data.



In S3 created a bucket in which the lambda output is stored in the JSON format.

Data:

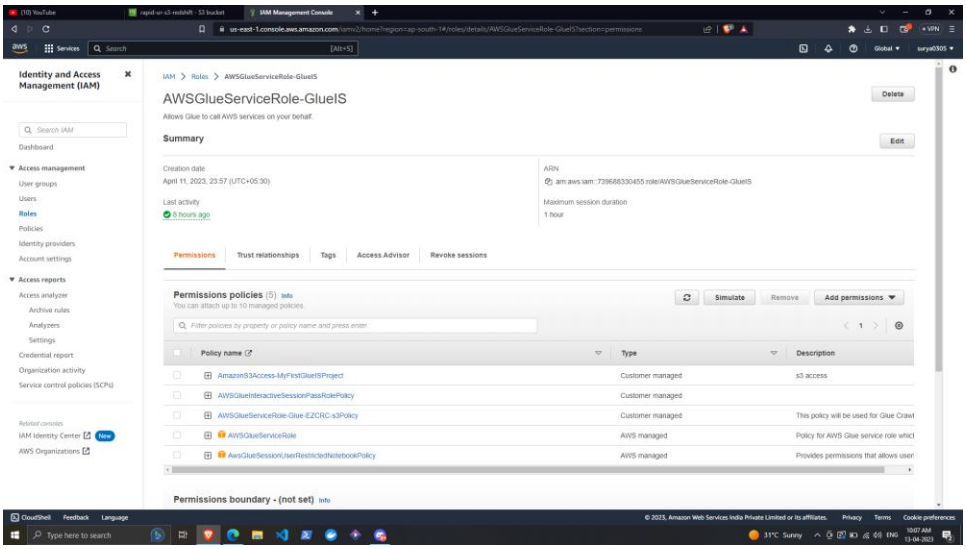
{"id": "tt0001702", "is_series": false, "Title": "The Indian Maiden's Lesson", "Release_Date": "1911-04-22"}



For the Glue job I had created a role **AWSGlueServiceRole-GlueIS**

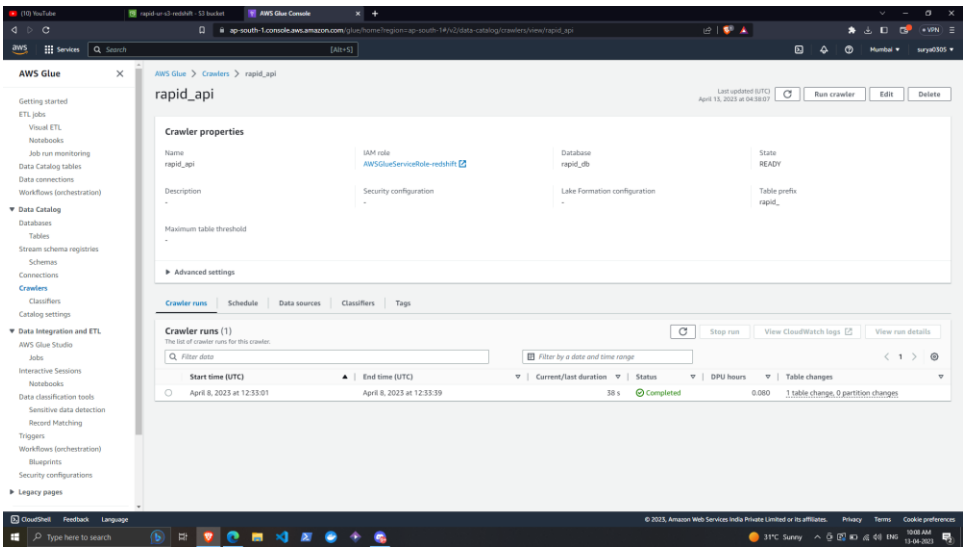
The following policies attached are

- AmazonS3Access-MyFirstGlueISProject
- AWSGlueInteractiveSessionPassRolePolicy
- AWSGlueServiceRole-Glue-EZCRC-s3Policy
- AWSGlueServiceRole
- AwsGlueSessionUserRestrictedNotebookPolicy

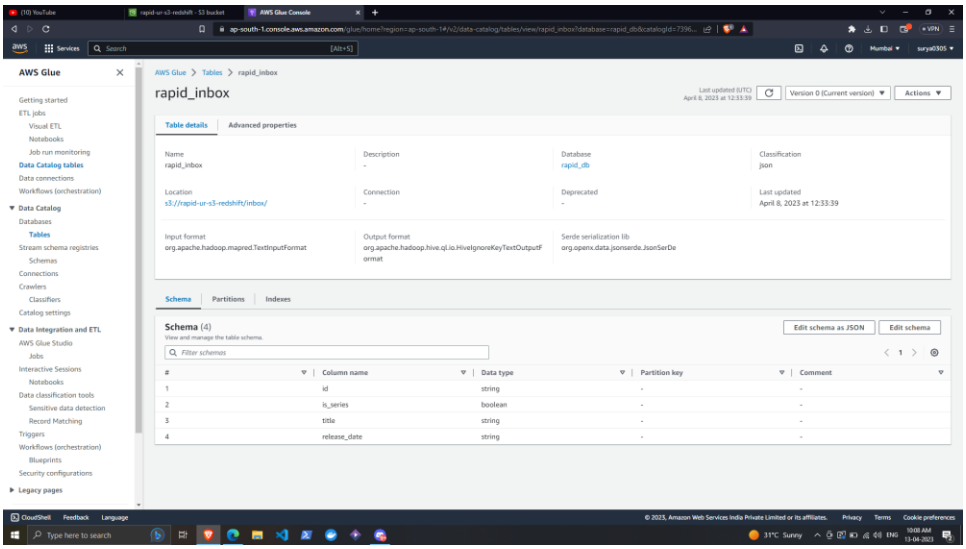


Creating an AWS Glue crawler

I had created a crawler in Glue to store the metadata info in Glue Catalog as shown below

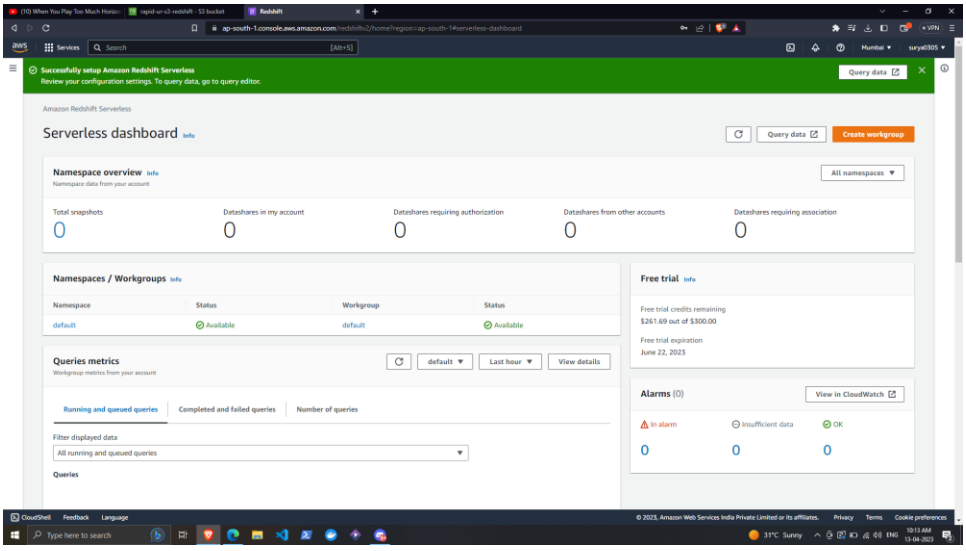


After successful crawler run the metadata is stored in **rapid_api** Database and in **rapid_inbox** table

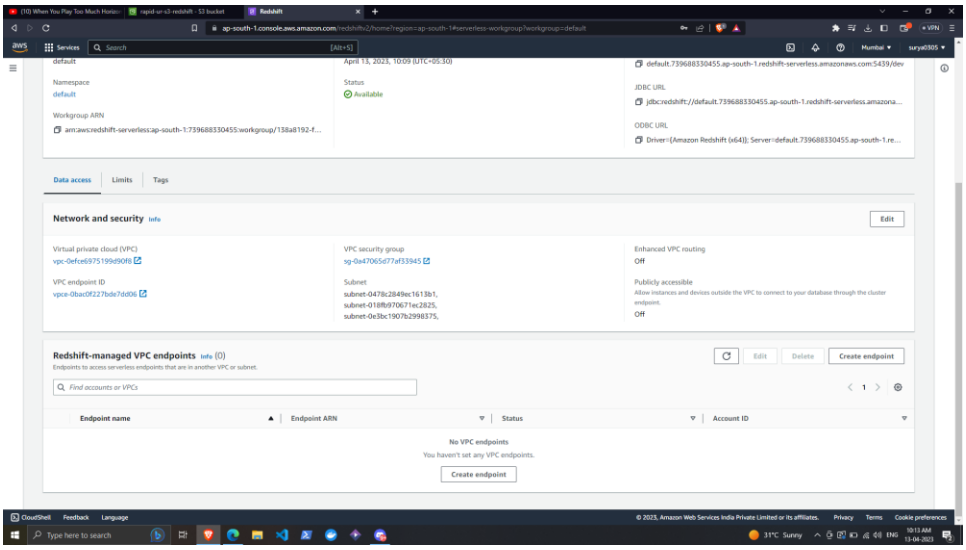


Creating a Redshift Serverless Cluster

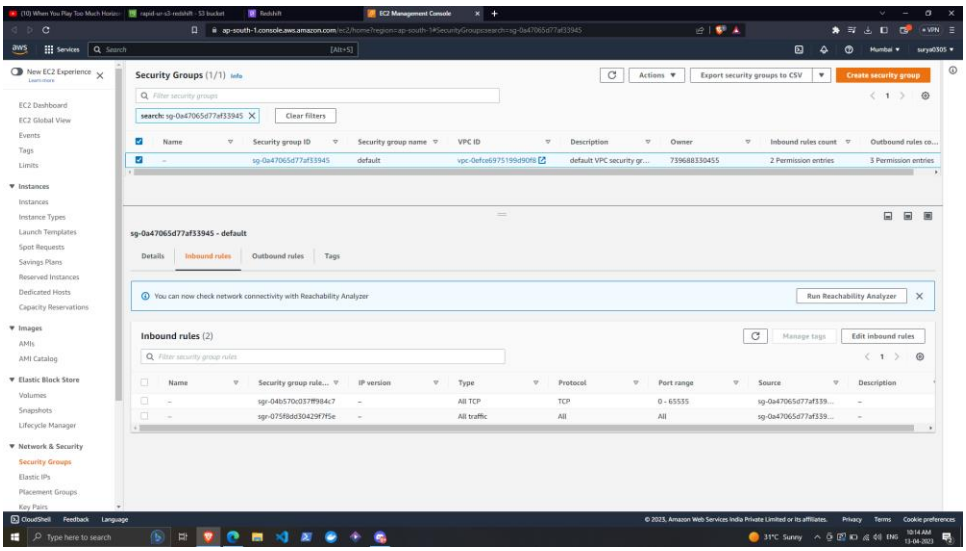
Now create a serverless redshift cluster in which mentioning username and password of our choice as they will used in GLUE-REDSHIFT connectivity.



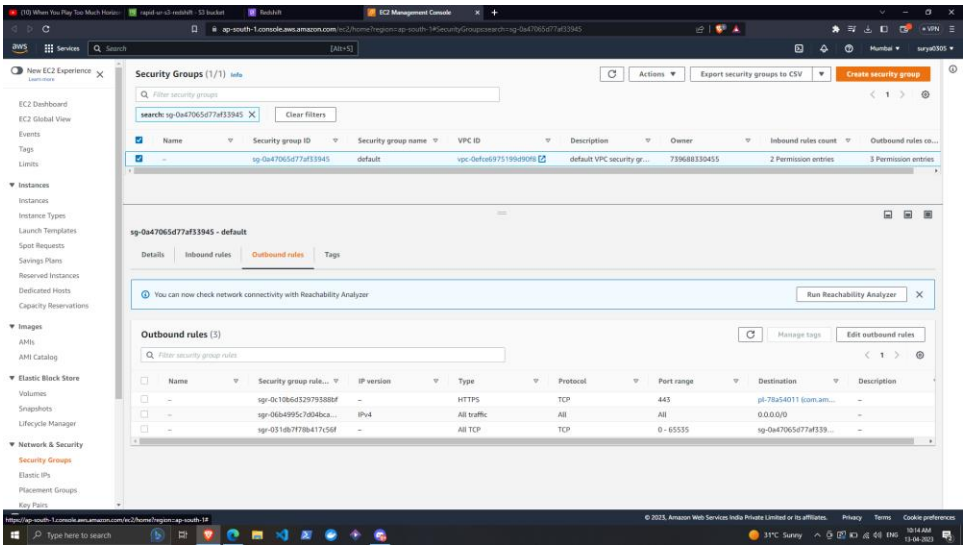
Open the workgroup of the serverless cluster where the security group is mentioned by clicking it, we will be redirected to EC2 instance window.



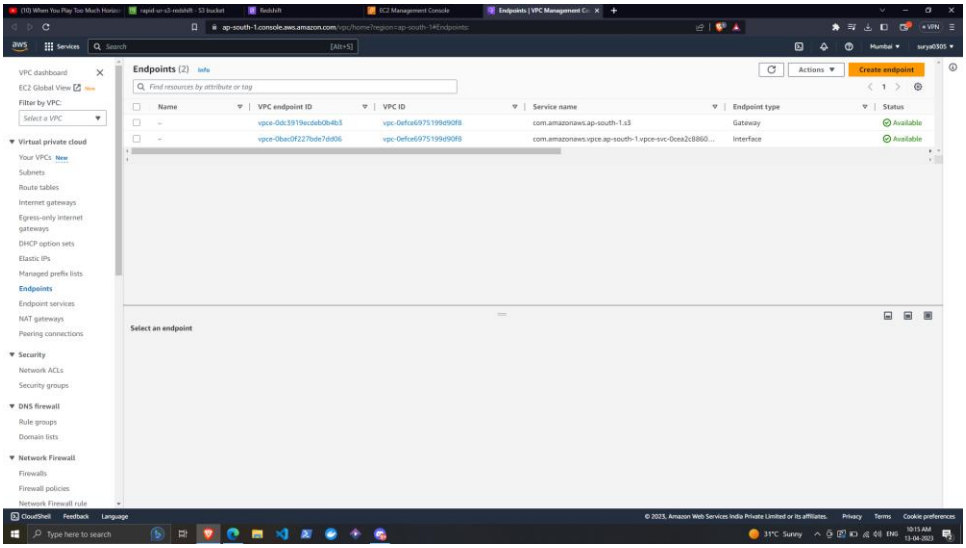
Now set up the inbound rule to allow all traffic from TCP and this rule should be a self-referring rule as shown below.



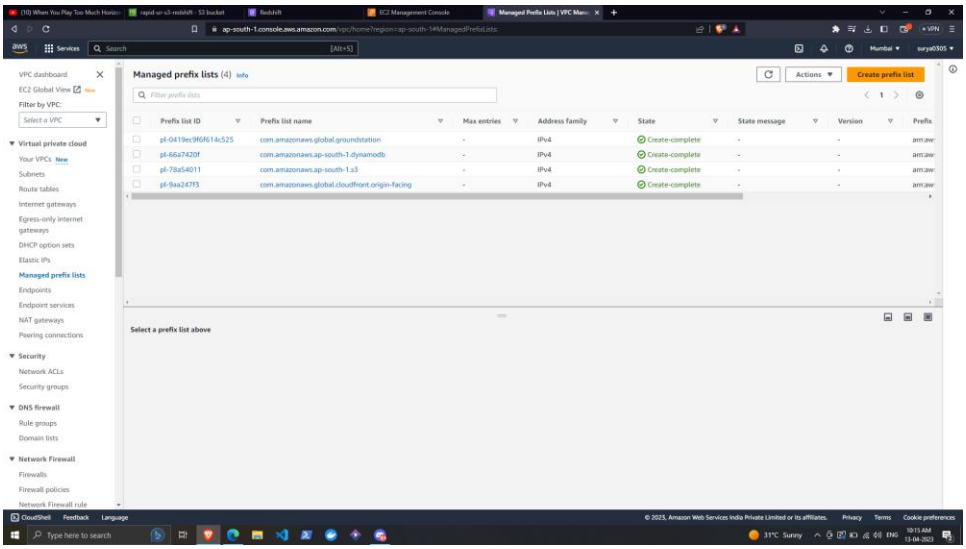
Again, setup the outbound rule will all TCP and HTTPS and both rules are self-referring rules as shown below.



Now to create a HTTPS outbound rule which would be connecting to S3, create an endpoint to point towards S3 Gateway



The Managed prefix for the endpoint will be used by the HTTPS outbound rule

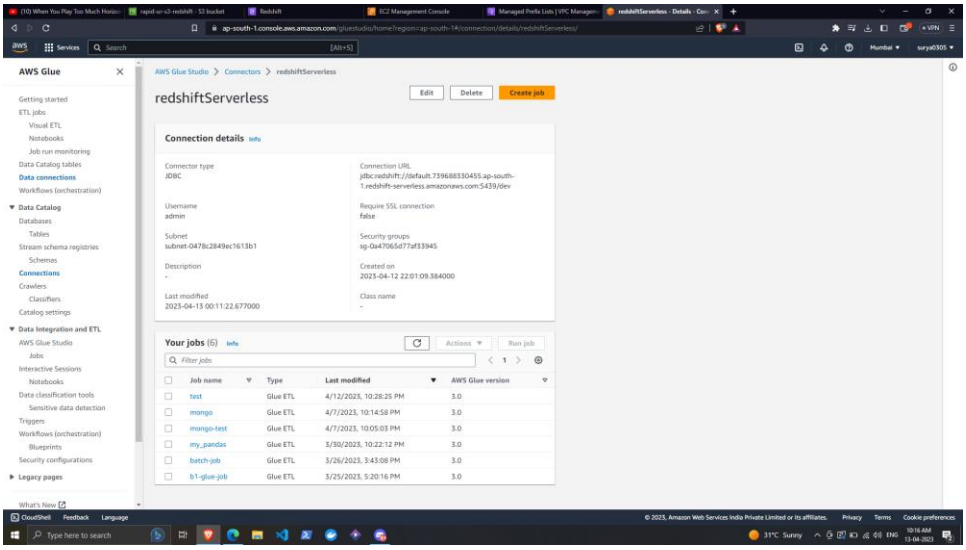

Managed prefix lists (4) Info
Actions
Create prefix list
Filter prefix lists
Prefix list ID
Prefix list name
Max entries
Address family
State
State message
Version
pl-0419ec9f6f614c525
com.amazonaws.global.groundstation
IPv4
Create-complete
pl-66a7420f
com.amazonaws.ap-south-1.dynamodb
IPv4
Create-complete
pl-78a54011
com.amazonaws.ap-south-1.s3
IPv4
Create-complete
pl-9aa247f3
com.amazonaws.global.cloudfront.origin-facing
IPv4
Create-complete
Select a prefix list above

Creation of Glue Job

First, before creating a job we need to have a connection to our redshift cluster.

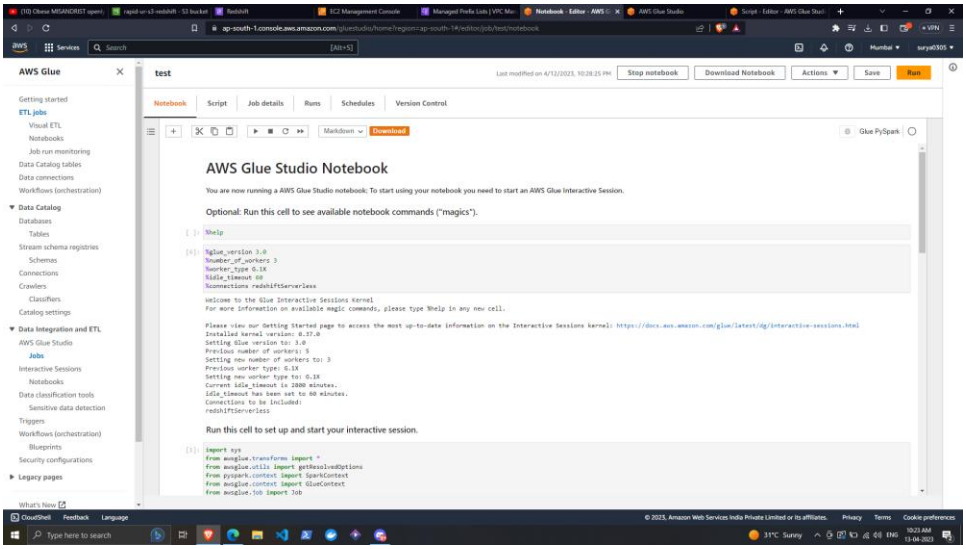
In AWS Glue open connection and use JDBC option put the credentials of username and password of the cluster.

After giving the details final details would look like as shown below.



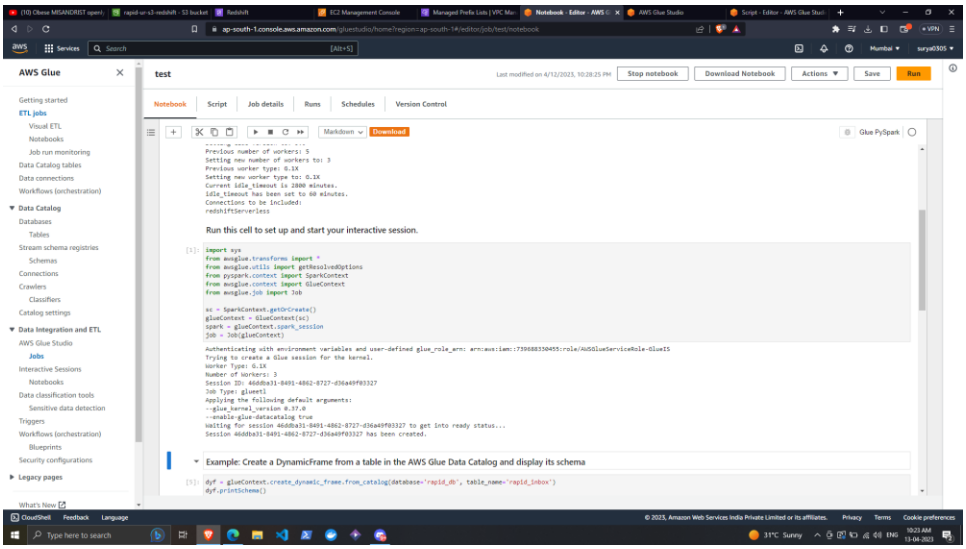
Now under the Job section, create a job using Notebook as shown below.

Use the following IAM Role **AWSGlueServiceRole-GlueIS**

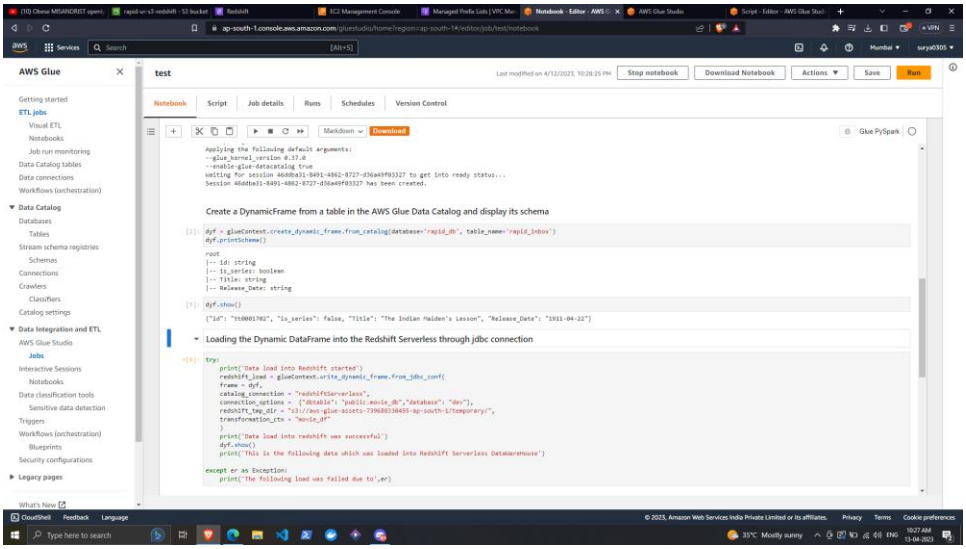


Execute the following blocks as shown below

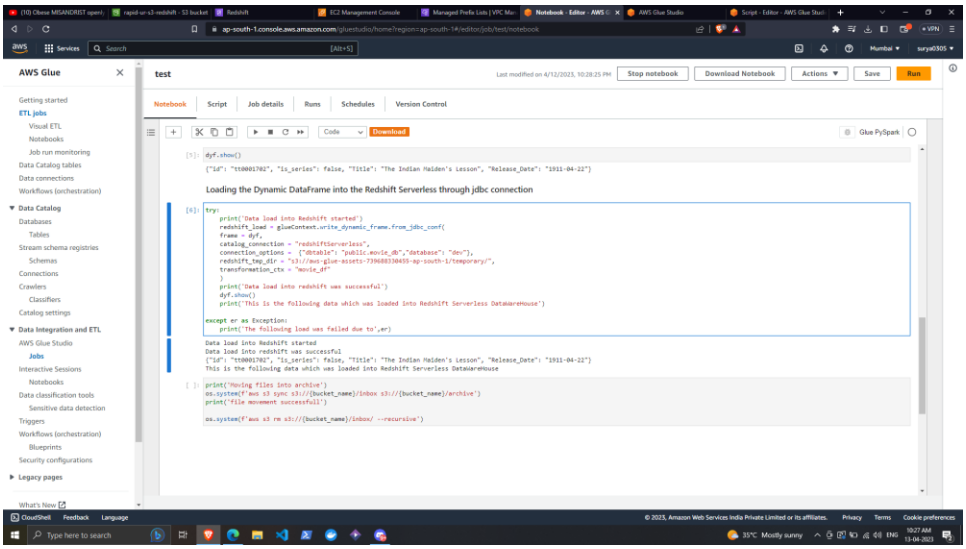
First, I am checking whether the redshift cluster is connecting or not.



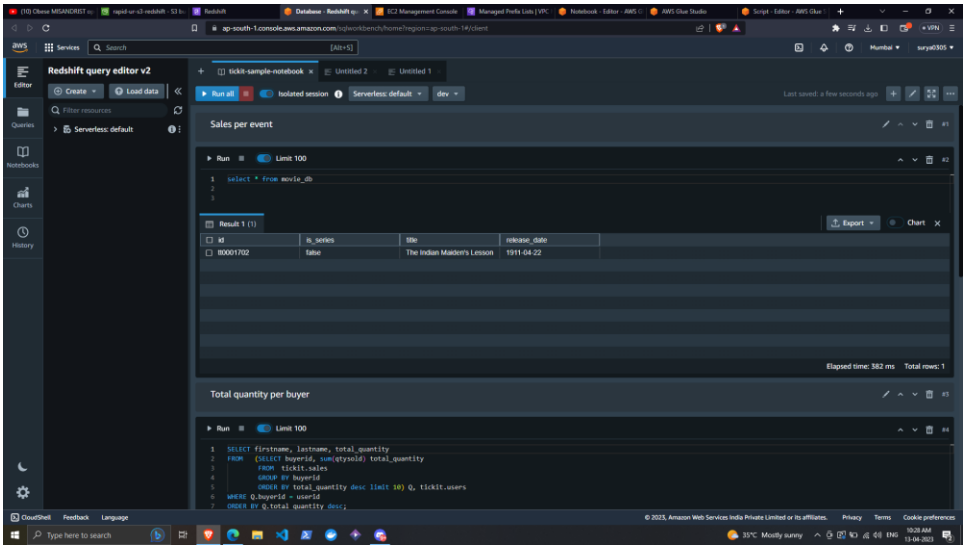
Once the connection is established, then I am reading data from catalog table



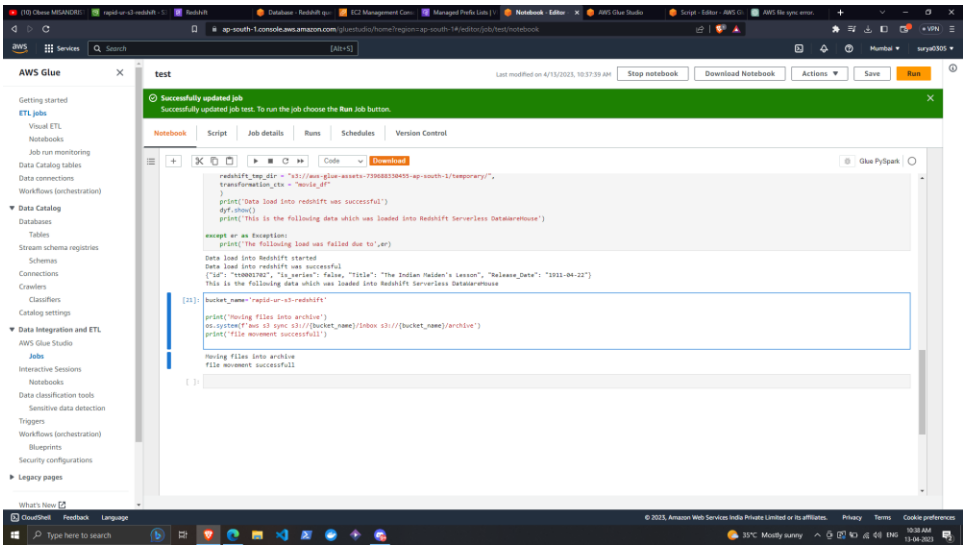
After successfully reading from the table and verifying the schema, now I am connecting to the cluster and loading the data into it. By this method we can directly load it without creating the table.

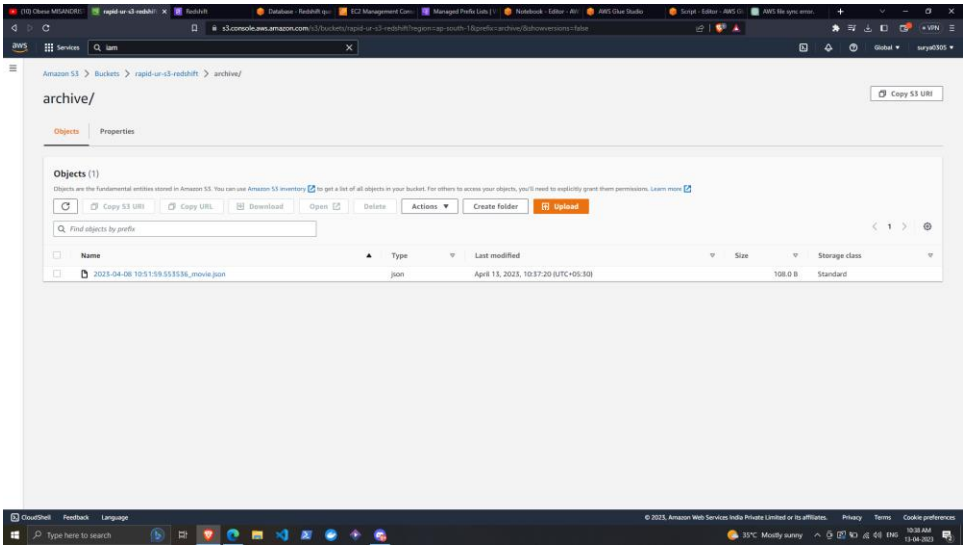


After the load is completed, the data is visible in Redshift.



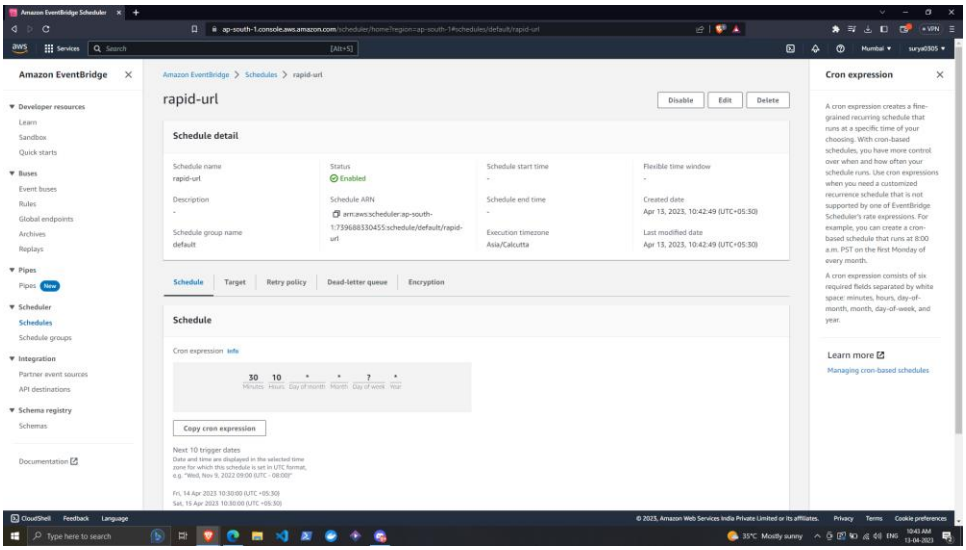
After the DB load is completed the JSON files in /inbox folder is moved to /archive folder and the files in /inbox folder is removed.

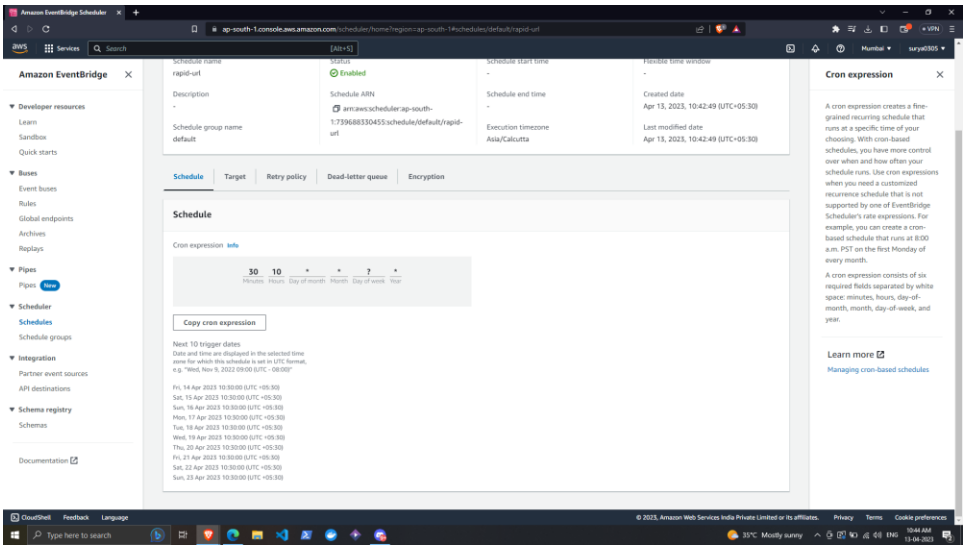

AWS Glue
test
Stop notebook
Download Notebook
Actions
Save
Run
Successfully updated job
Successfully updated job test. To run the job choose the Run Job button.
Notebook
Script
Job details
Runs
Schedules
Version Control
Download
Glue PySpark
Data load into Redshift started
Data load into redshift was successful
Having files into archive
file movement successfull


Amazon S3
Buckets
rapid-ur-s3-redshift
archive/
archive/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
Name
Type
Last modified
Size
Storage class
2023-04-08 10:51:59.553536_movie.json
json
April 13, 2023, 10:37:20 (UTC+05:30)
108.0 B
Standard

Creating a schedule for Lambda Function

Using Event-Bridge I had scheduled the lambda function to trigger daily around 10:30 AM IST





Similarly, created a rule to generate to trigger the Glue-Job