Project Overview-

Welcome to my project.

I've created an End to end data pipeline project on a Spotify data set using AWS Glue and Amazon Crawler.

Then Used Amazon S3 bucket to store the input and output files accordingly and used Amazon Athena to query the tables made by the crawler.

The Goal of this project as shown in the architecture diagram, is to demonstrate the features of the aws platform to upload documents, extract and clean the documents and transform them according to the user's needs.

Under this section, I've mentioned each and every steps by which one can be able to visualize and understand the scenario, the objectives and the tools applied while making this project from root level.

Thank you.

Project Architecture-

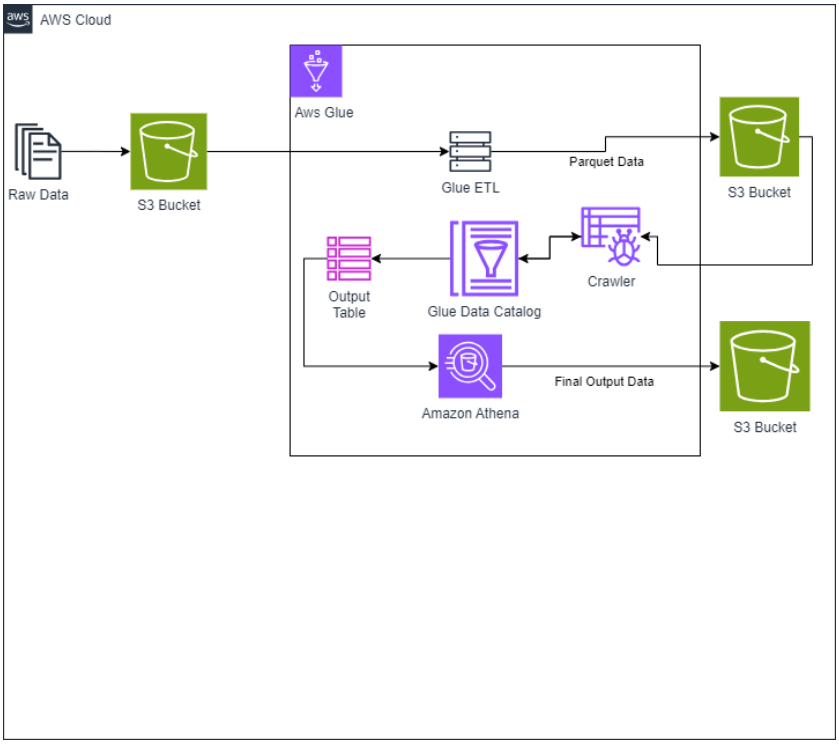

Fig1-Project Architecture

Detailed steps-

1.Make an Amazon Aws account first and validate your account to keep using it(It provides a one year of free tier service which one can use efficiently for making projects or in their day to day activities.)

After that, you can use it as “Pay-as-you-go” model by which, you will have to pay for the services that you use only.

2. **Collecting dataset-** Collecting random datasets can be very easy task as we have lot of datasets that are available in the internet for free. You can choose any datasets of your choice and start to work on them.

For this particular project, I’ve used the Spotify (2013) dataset which I’m providing within this folder.

3.Uploading the file- At first make a S3 bucket and after the bucket creation, go to the S3 bucket and upload the csv files.

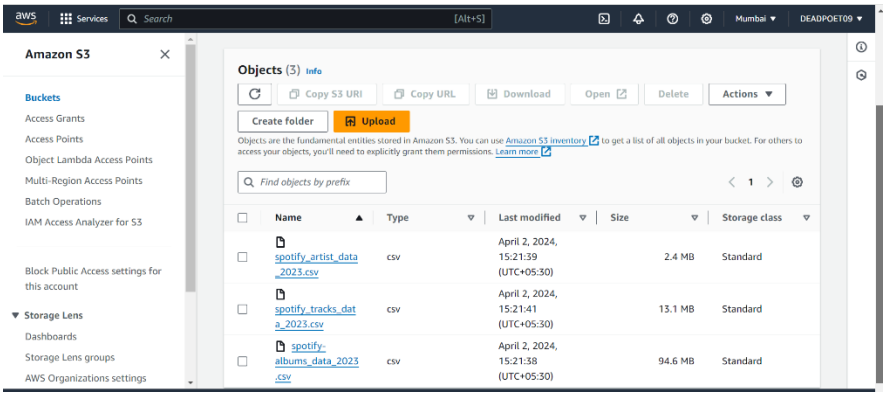

Fig 2-Input Files

4.After uploading the files, go to the Amazon Glue(It is not a free tier service but It’s cost is very minimal(For one hour of use, it will cost you under 100 inr) and start building the ETL pipeline.

5.**ETL pipeline-**i) At first, create a data source as S3 bucket were you stored your data sets.

ii) From the transform section ,pull the join function to put a join between “Album” and the “Artist” datasets and name it a specific value lets say “Join-Album-Artist”.(Put the format according to the dataset.Here we are using CSV so I made that as CSV)

iii) Again pull one more join function to join the “Tracks” and “Join-Album-Artist” values and rename it with “Join_With_Tracks” ”.(Put the format according to the dataset.Here we are using CSV so I made that as CSV)

.

iv) After doing this,we can be able to find out that there are many similar columns(For example,the ".id" column is repeated twice).So we need a Transform function to delete all the unnecessary datas from the "Join_With_Tracks" dataset.

v) Finally you can bring an output bucket to put the output and mark it as a parquet format and save it(**The reason to use parquet format-Less data scanned at ingestion leading to faster scans and lower query and memory costs because you only have to read the columns that you actually need**.).

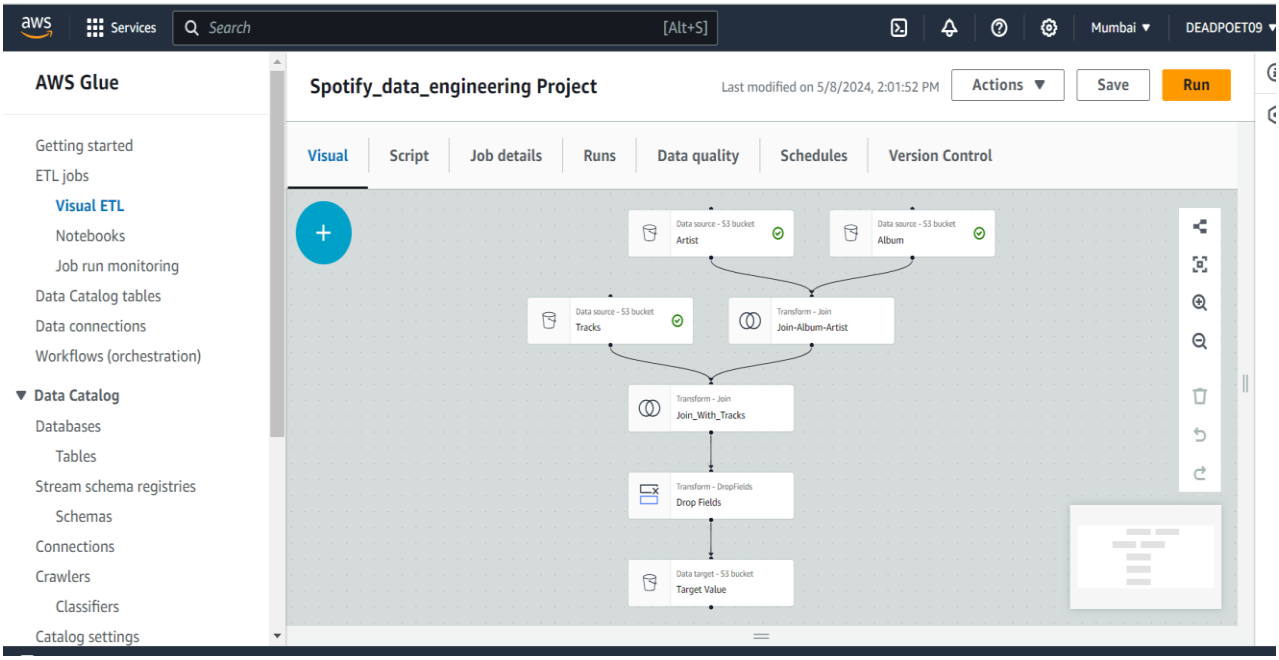

Fig3- The ETL process

vi)After that just hit Run and you are good to go(Provided with the "Spotify_data_engineering Project.py" that was auto generated after the "Run" button was exwcuted).

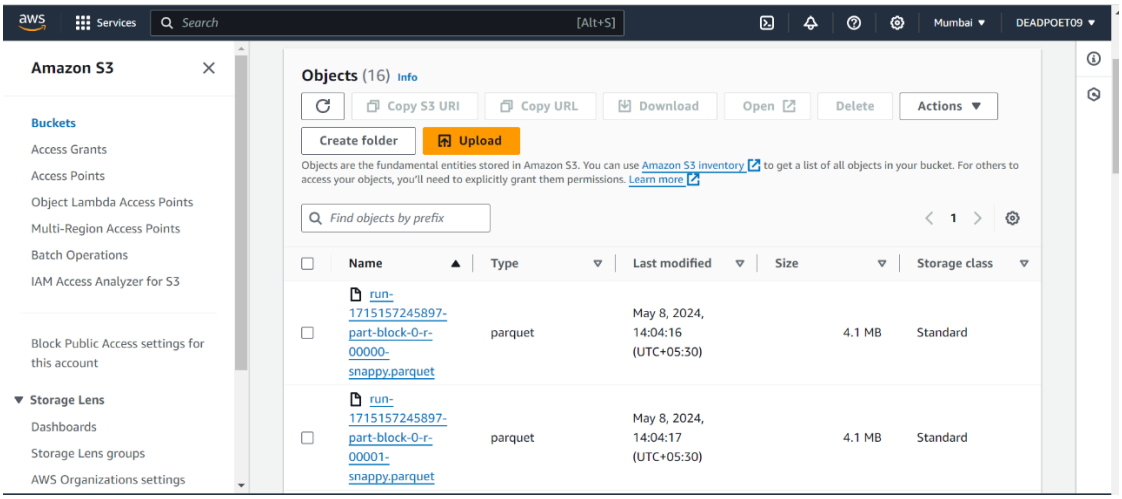

Fig4-Datas after the ETL process (Parquet format)

6.Configuring Database- Make a database of any name on which we can be able to store the Tables.(Use proper IAM rules to For Glue to access the Database).

7.Create a Glue crawler to process the parquet datas to create a data-catalog which can be further used for meaningful information.

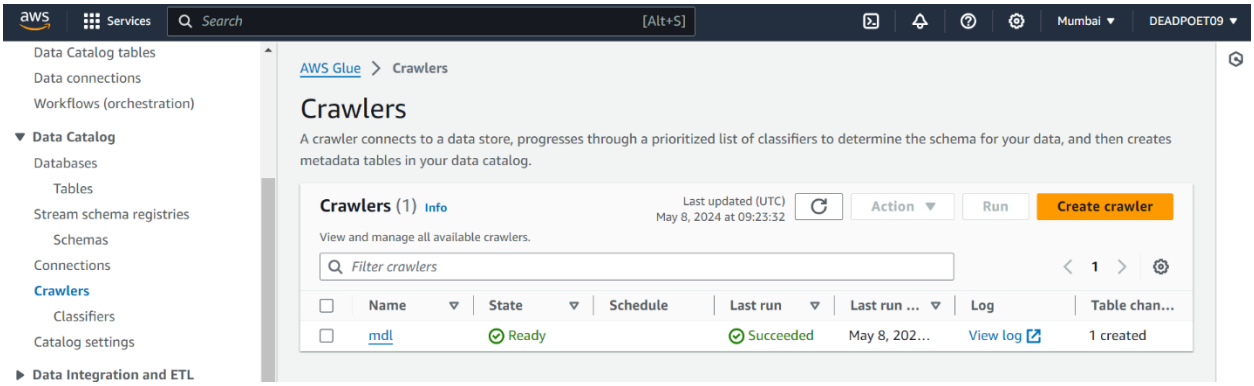

Fig5-Crawler

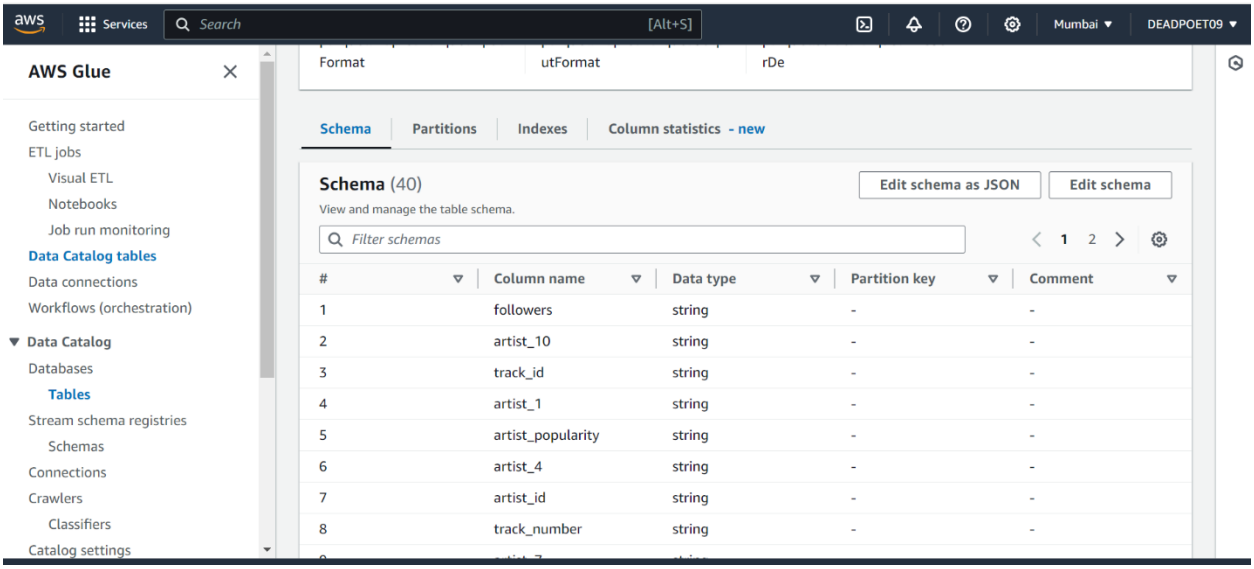

Fig6-Table after crawler is executed

8.Finaly use The Amazon Athena to query the table accordingly to your need.

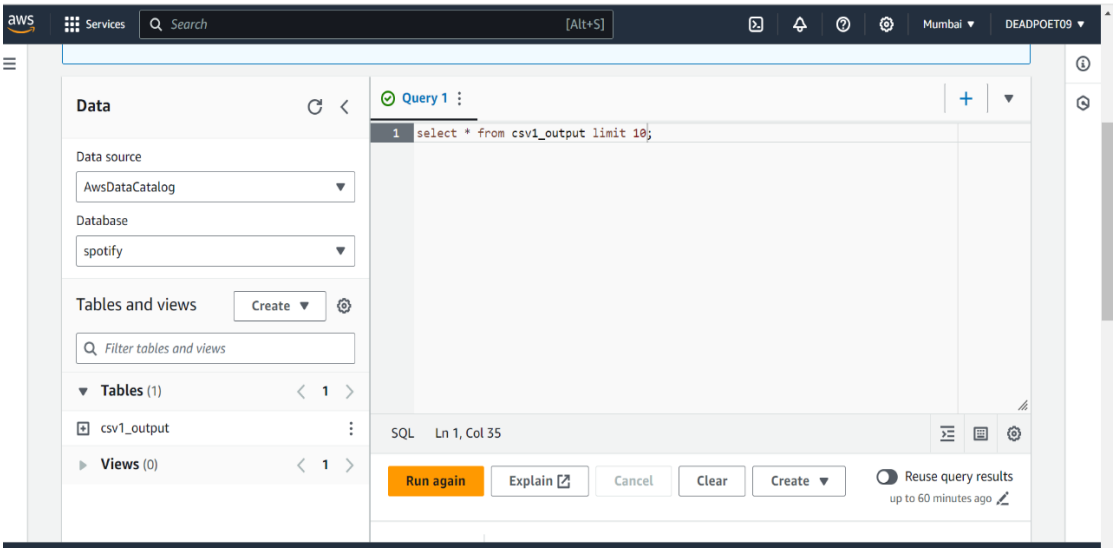

Fig7-Run query on Amazon Athena

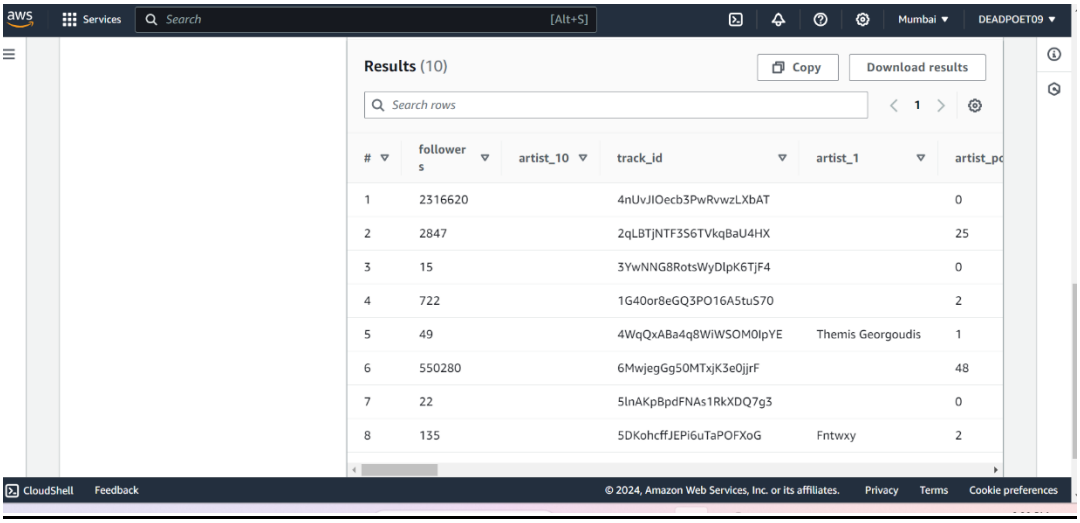

Fig8-Athena Output

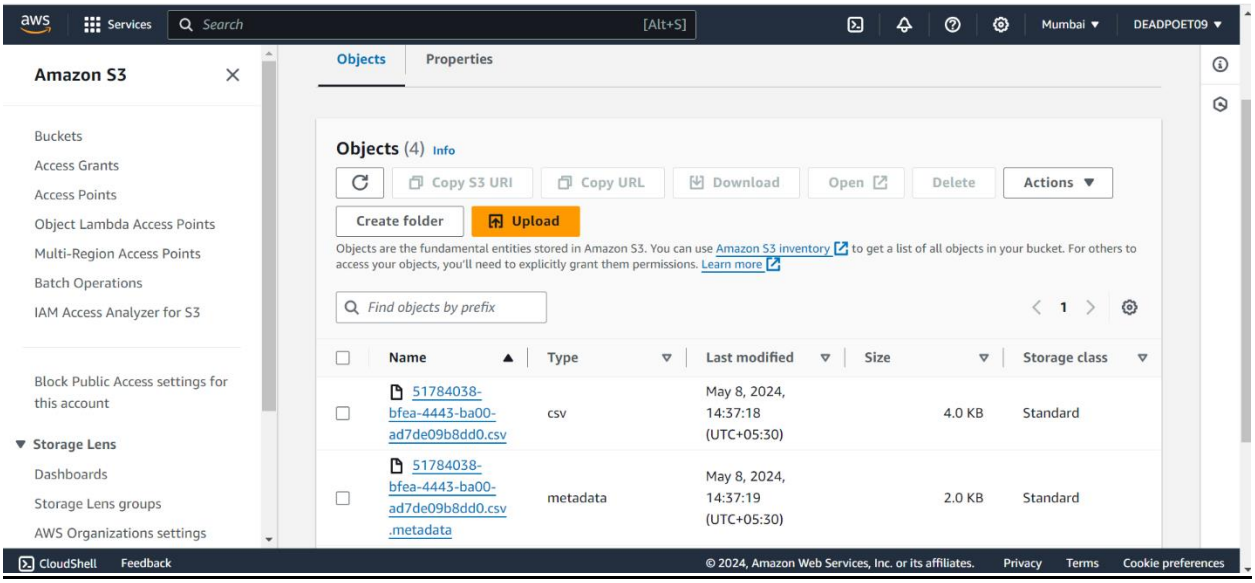

Fig9- Final Output in S3 Bucket