AWS Data Pipeline for Spotify Analytics

1. Introduction

This project demonstrates how to set up a complete AWS data pipeline for processing and analyzing Spotify data. The pipeline involves various AWS services including S3, Glue, Athena, and QuickSight. The goal of this project is to extract, transform, and load (ETL) data from a staging area to a data warehouse and then use Athena and QuickSight for querying and visualizing the data.

2. Architecture Overview

The architecture for this project is designed as a data flow from S3, through Glue, Athena, and QuickSight, as shown below:

1. **S3 Staging**: Raw data is stored in S3 in the 'staging' folder.

2. **AWS Glue**: ETL jobs in AWS Glue clean, join, and prepare the data.

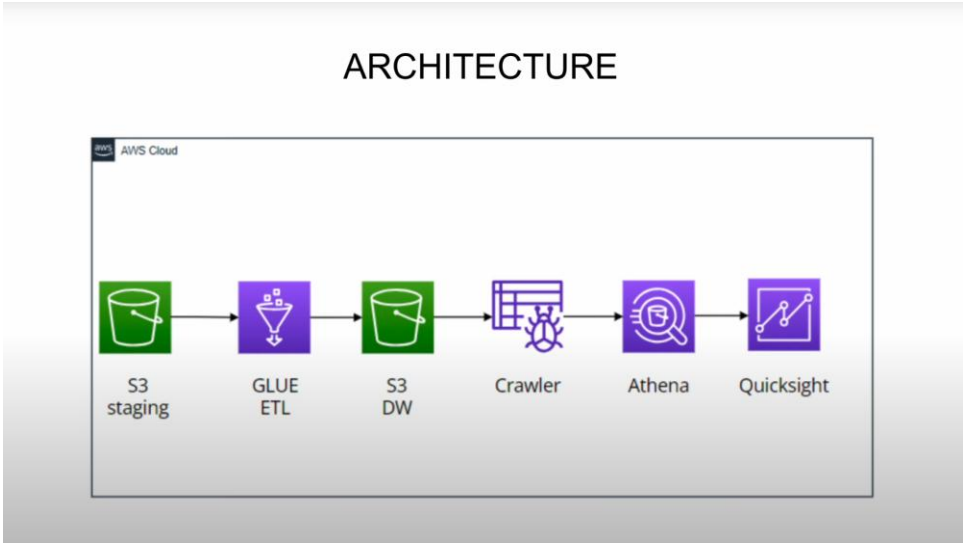
3. **Athena**: Queries the data using SQL via Athena.

4. **QuickSight**: Visualization and analytics using QuickSight.

The S3 storage and Glue ETL jobs work in conjunction to transform raw data into usable insights.

Architecture Diagram

Below is the architecture diagram illustrating the flow of data through the pipeline:

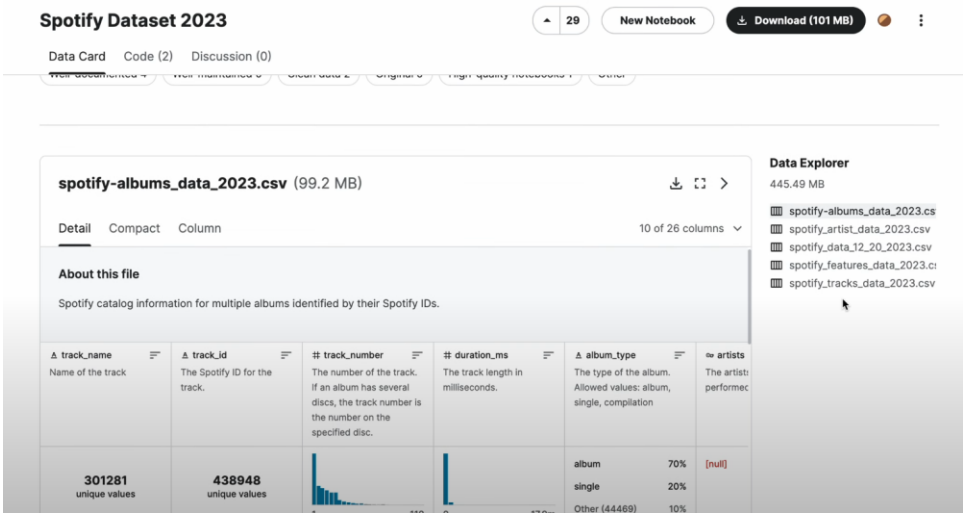


3. Dataset

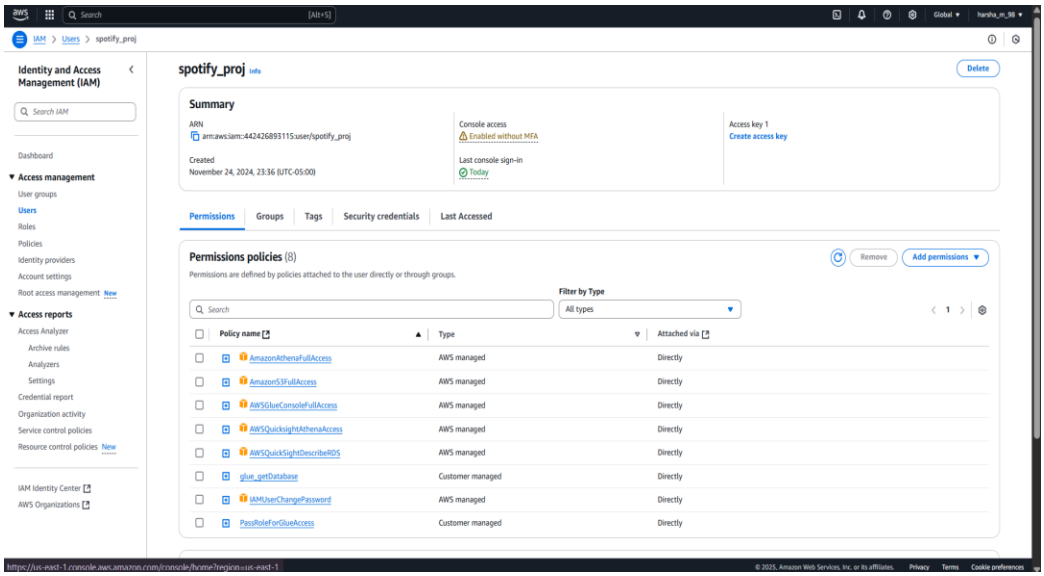
The dataset used for this project is the Spotify data, available at: https://www.kaggle.com/datasets/tonygordonjr/spotify-dataset-2023. This dataset contains information about tracks, albums, artists, and other related metrics like track duration and popularity.

Dataset Preview

Below is a preview of the dataset stored on Kaggle:



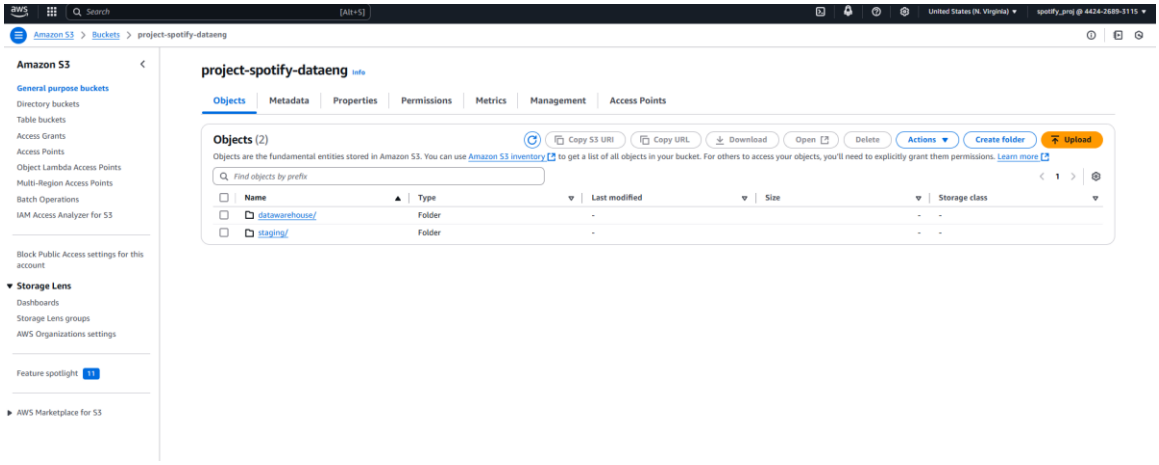
4. Creation of user and assigning roles for the user



5. S3 Setup

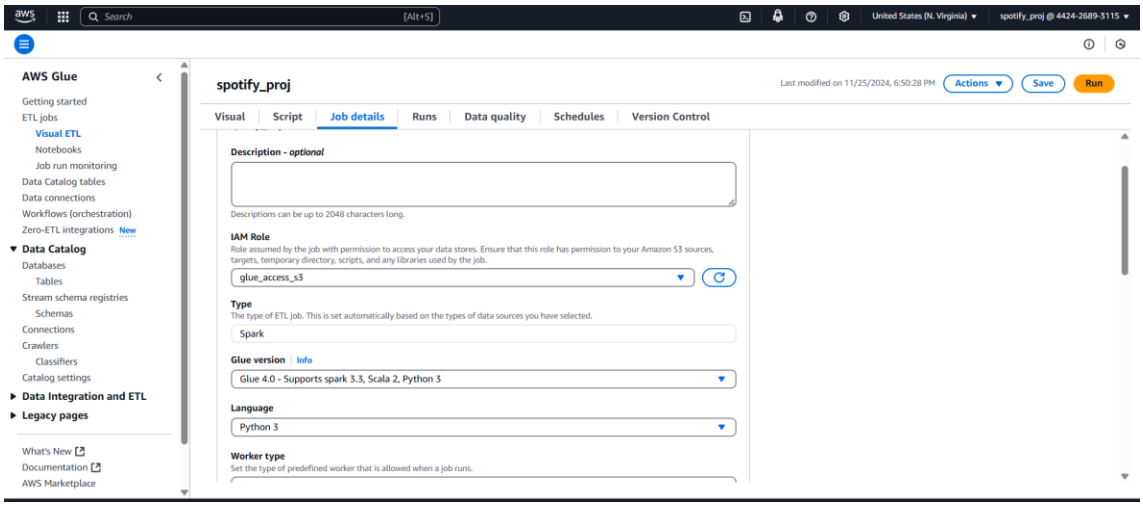
The project uses two main S3 buckets: '**project-spotify-dataeng**' for storing the raw and transformed data, and '**athena-spotify-query-output**' for storing the results of queries run in Athena.

The data is divided into two folders within the 'project-spotify-dataeng' bucket: '**staging**' and '**datawarehouse**'. The raw dataset is first uploaded into the 'staging' folder, and after processing, the cleaned and transformed data is stored in the 'datawarehouse' folder in Parquet format.



5. IAM Role Setup for Glue

For security, an IAM role named '**glue_access_s3**' was created. This role provides the necessary permissions for AWS Glue to access data in S3 and perform ETL operations. The role was attached to the Glue jobs.

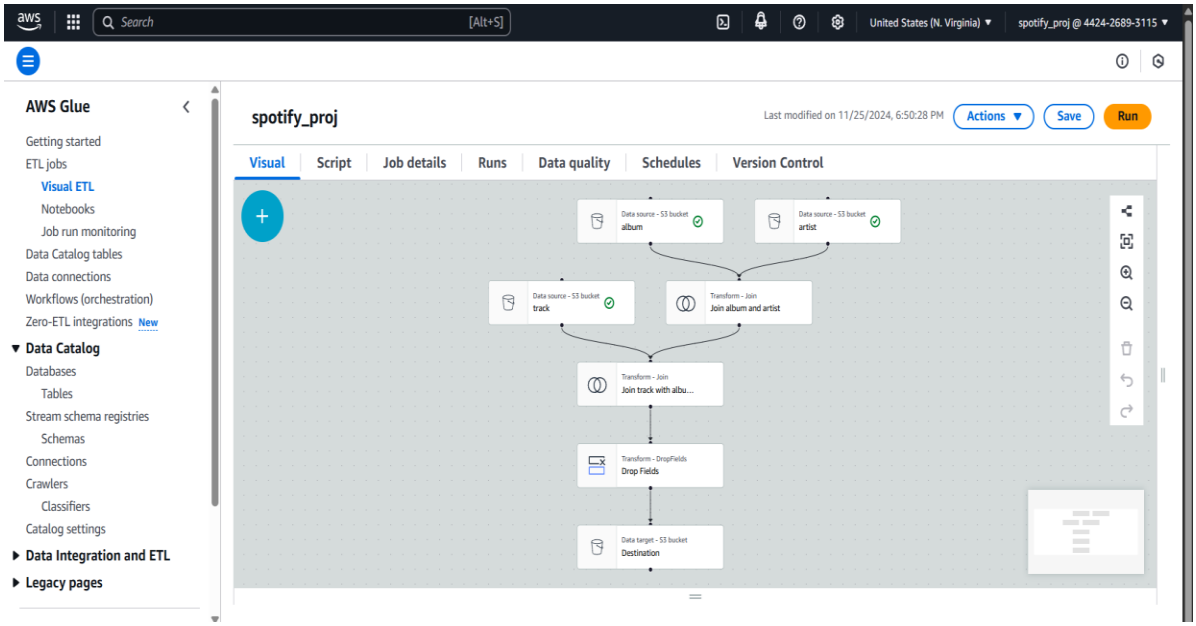


6. AWS Glue Visual ETL

AWS Glue Visual ETL was used to join data from multiple sources in the staging area and clean it. Specifically, the data from the '**track**', '**album**', and '**artist**' tables were joined based on their relationships. Afterward, unnecessary columns, like IDs, were dropped, and the transformed data was stored in the data warehouse folder.

AWS Glue Visual ETL Workflow

Below is the visual ETL workflow in AWS Glue:



7. Running the Glue Job

The job was run with the following configurations: - **IAM Role**: glue_access_s3 (allows access to the data stored in S3) - **Glue Version**: 4.0 (supports Spark 3.3, Python 3) - **Worker Type**: G.1X (4 vCPU, 16 GB RAM) - **Number of Workers**: 5

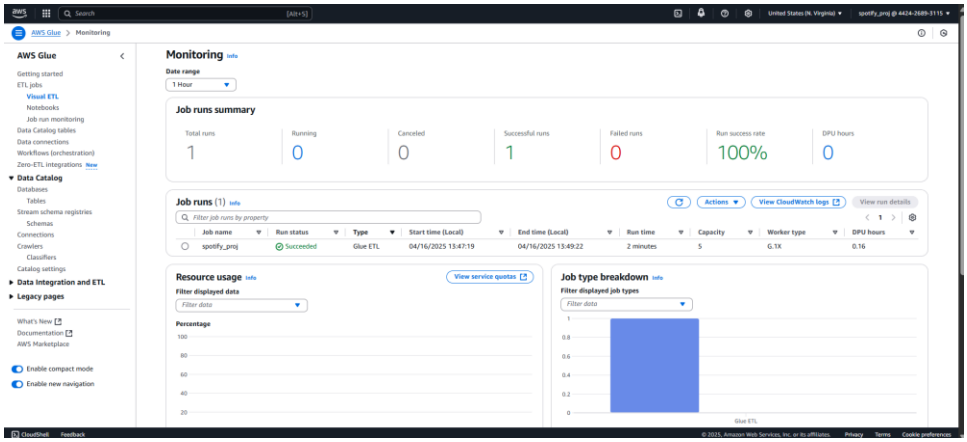
Once the job was run successfully, the output was stored in the S3 'datawarehouse' folder in Parquet format.

8. Monitoring the Glue Job

The job was monitored via the AWS Glue console. Upon successful completion, the job's success rate was shown to be 100%. The run time was 2 minutes, and 0 DPU hours were used.

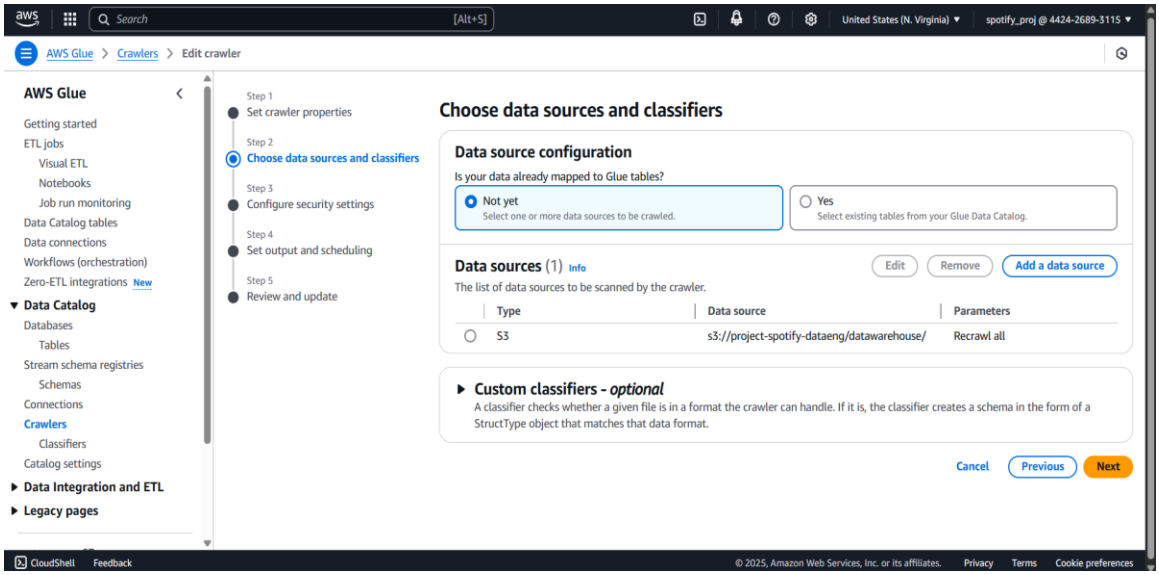
Glue Job Monitoring

Below is the monitoring dashboard showing the status of the Glue job:



9. AWS Glue Crawler

A Glue Crawler was configured to scan the data in the 'datawarehouse' folder in S3 and map it to a Glue database called '**spotify**'. The crawler was set up with an appropriate IAM role to access the data and catalog it for querying.

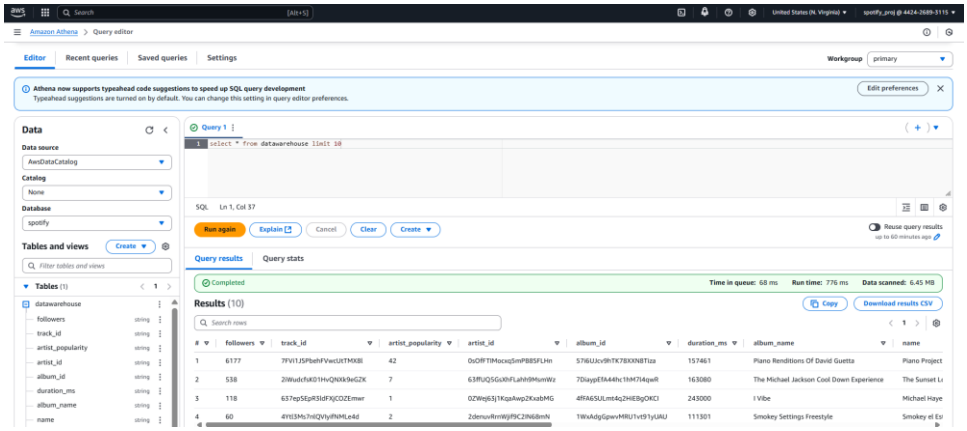


10. Querying the Data with Athena

Once the data was cataloged, AWS Athena was used to query the data. Athena was connected to the '**spotify**' database, and SQL queries were run to explore and analyze the data. The results were stored in an S3 bucket, '**athena-spotify-query-output**'.

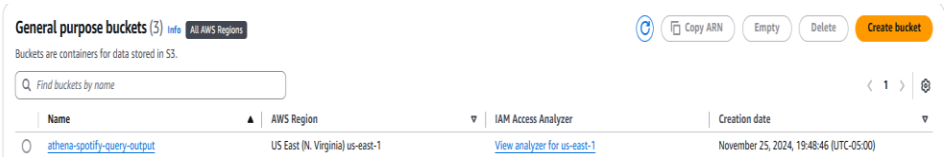
Athena Query Results

Below is an example query result in Athena:



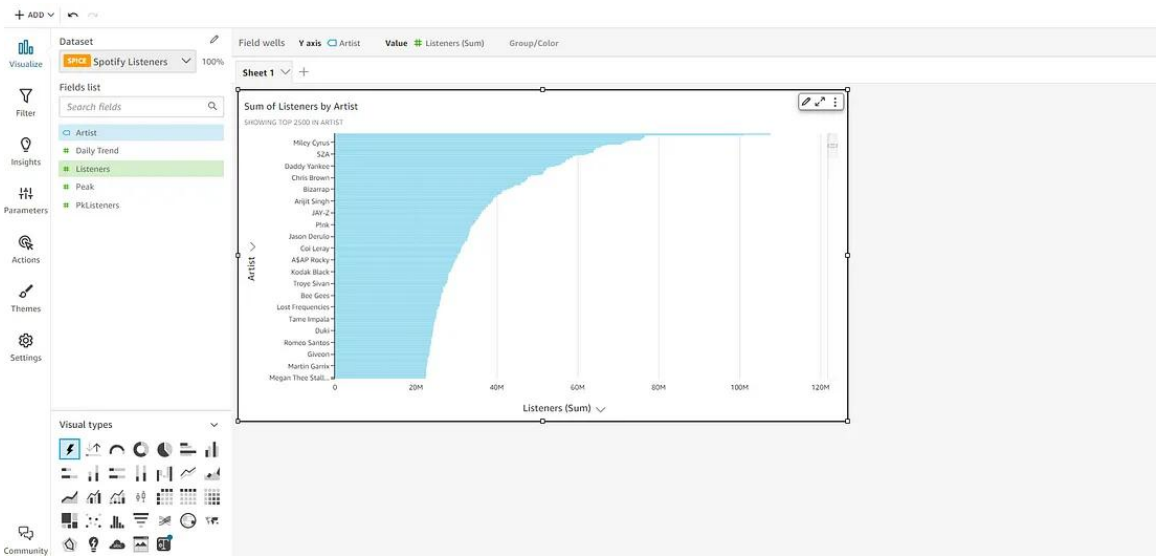
S3 Output Bucket

Below is the S3 bucket used for storing Athena query results :



11. QuickSight Integration

Finally, AWS QuickSight was used to create visualizations and reports based on the queries run in Athena. The QuickSight service was connected to the Athena data source and configured to use the results stored in the output S3 bucket.



12. Conclusion

This project demonstrates the end-to-end process of setting up an AWS data pipeline for Spotify data. By leveraging AWS services like S3, Glue, Athena, and QuickSight, we were able to automate the ETL process, catalog the data, query it, and create visual insights to analyze Spotify's dataset.