****

With this playful nod let's groove into the world of Spotify data engineering, where the rhythm of AWS services sets the stage. In this project, we'll journey through the AWS cloud, exploring how it powers Spotify's data infrastructure. From storage to processing, AWS orchestrates the symphony behind the scenes, transforming raw data into harmonious insights. So, grab your headphones and get ready to explore the AWS-powered beats of Spotify's data engineering universe!"

**SPOTIFY END TO END DATA ENGINEERING PROJECT USING AWS CLOUD**

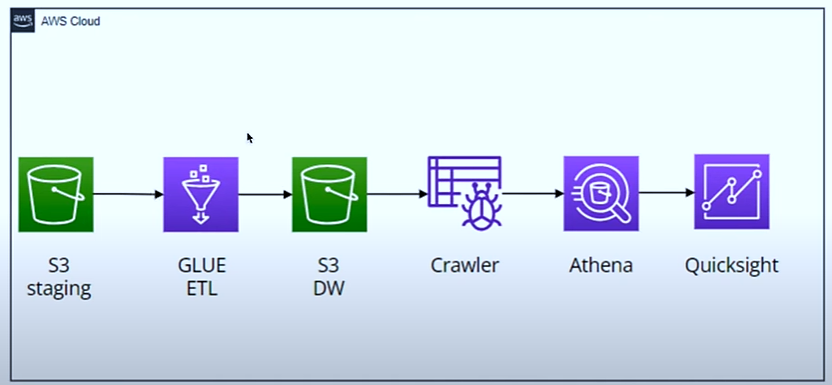
* AWS Services Used: S3, Glue, Athena, Quick Sight.
* **DATA SOURCE:**

Data used in this project is the Spotify dataset from 2023, sourced from Kaggle:

<https://www.kaggle.com/datasets/tonygordonjr/spotify-dataset-2023>

The dataset comprises five CSV files. As part of preprocessing the data, three CSV files were constructed: albums, artists, and tracks. These files can be accessed here: <https://drive.google.com/drive/folders/1PgZQDvw5GnvVQuhV7-MtxIZHnLsZA-Zs>

* **ARCHITECTURE:**

****

* The architecture outlined is generally correct for setting up a data pipeline in AWS:

1. Staging Data in S3: Data is stored in the S3 staging layer, serving as the initial repository for raw data.
2. ETL Pipeline with AWS Glue: AWS Glue is used to construct the Extract, Transform, Load (ETL) pipeline, facilitating the movement of data from S3 to the data warehouse. Glue automates the process of discovering, cataloguing and transforming data to make it suitable for analysis.
3. Database Creation with Crawler: After the data is transferred to the data warehouse, a crawler is employed to dynamically create a database schema and populate tables based on the data structure. This helps organize and make the data accessible for querying.
4. Querying Data with Athena: AWS Athena allows for ad-hoc querying of the data present in the tables created by the crawler. It enables users to run SQL queries directly against the data stored in S3, without requiring the setup or management of complex infrastructure.
5. Visualization with Quick Sight: Finally, AWS Quick Sight is utilized for data visualization, enabling users to create interactive dashboards and reports to derive actionable insights from the data. Quick Sight connects directly to the data sources, including Athena, to visualize data in real-time.

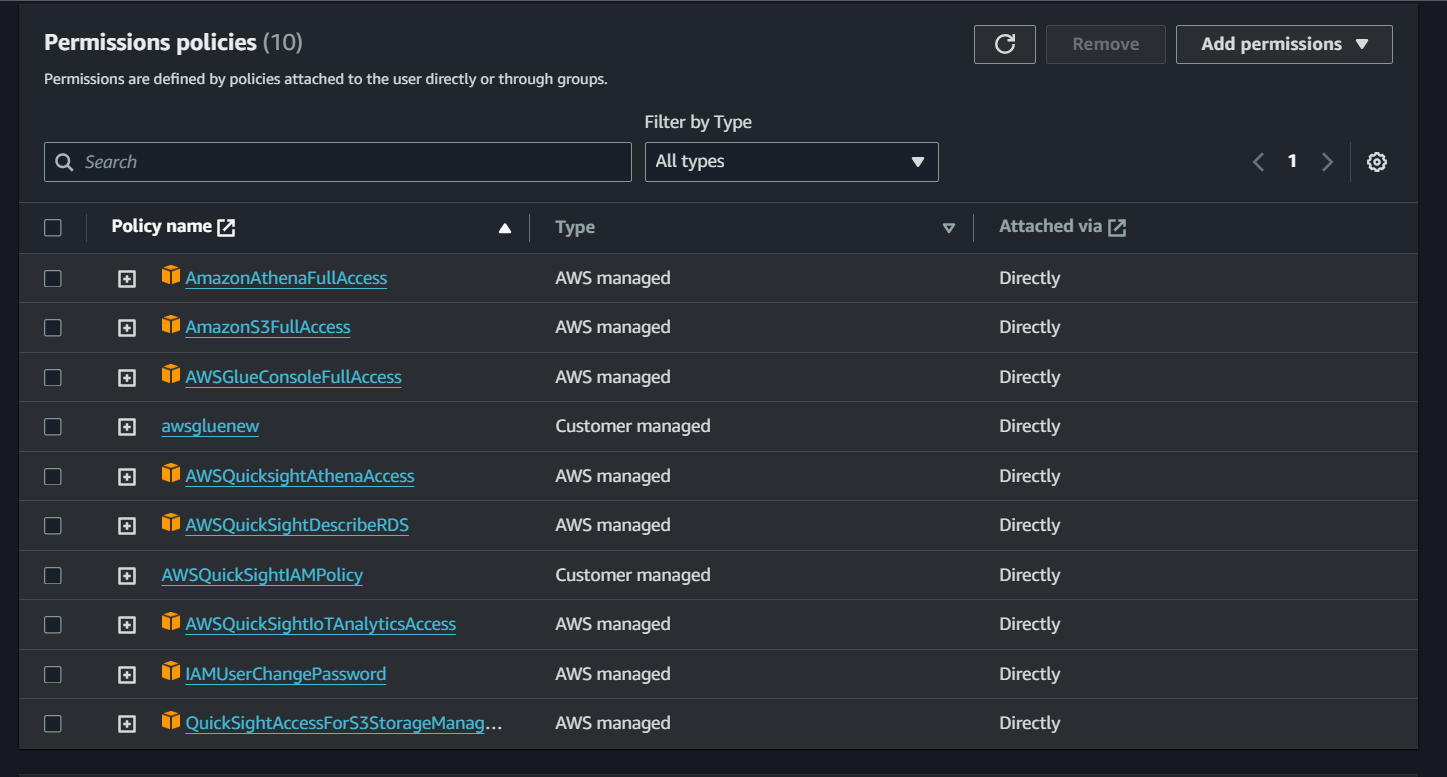
**STEPWISE:**

* **STEP 1** : Create IAM user

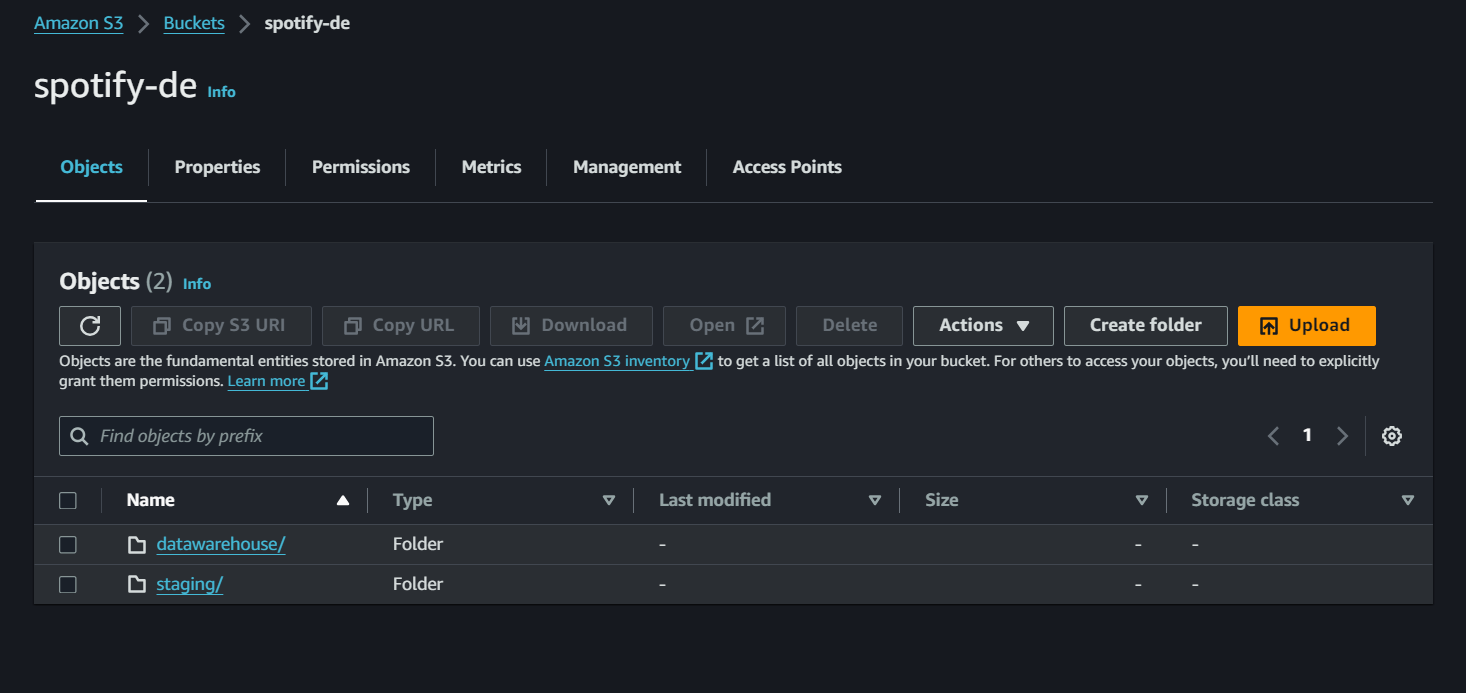
Why IAM user?

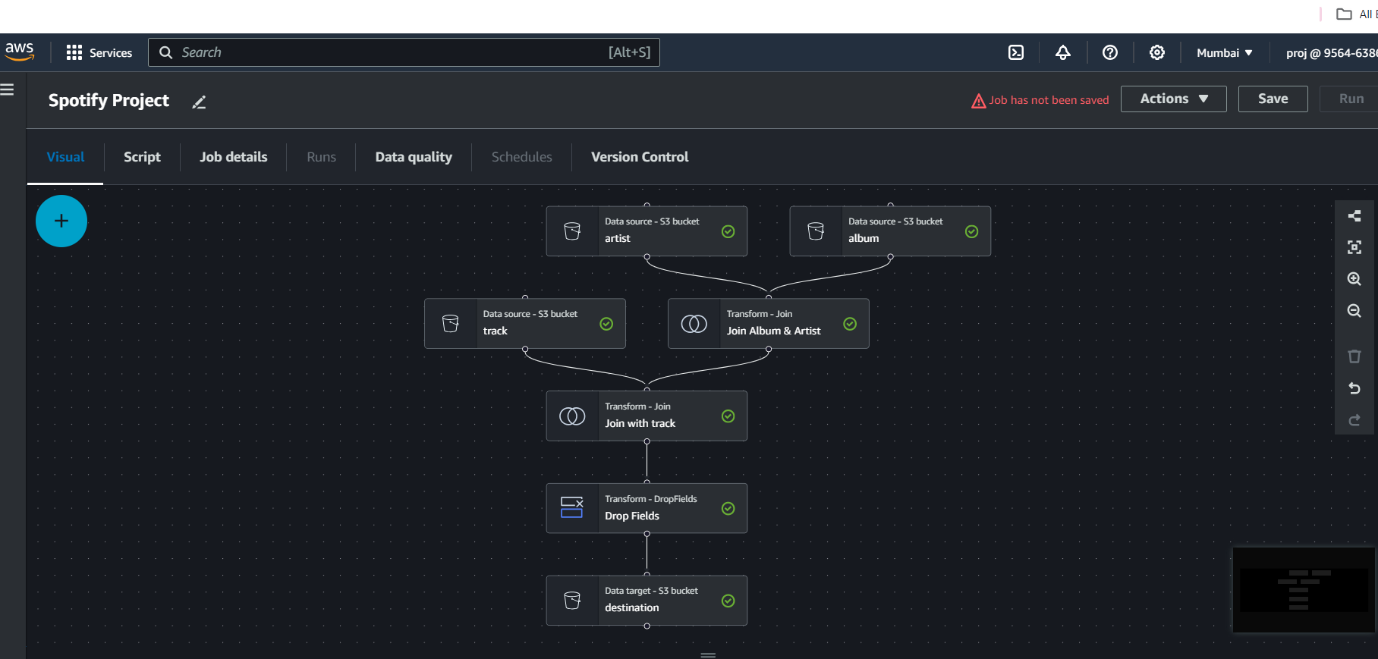
Since the root user has permissions, any misuse can lead to bigger issues, so it is recommended not to use the root user

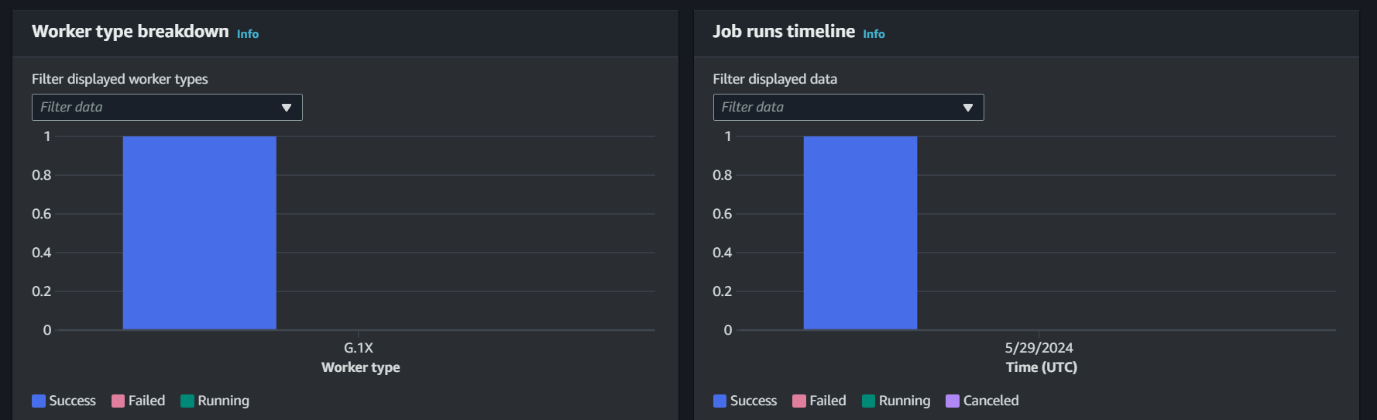
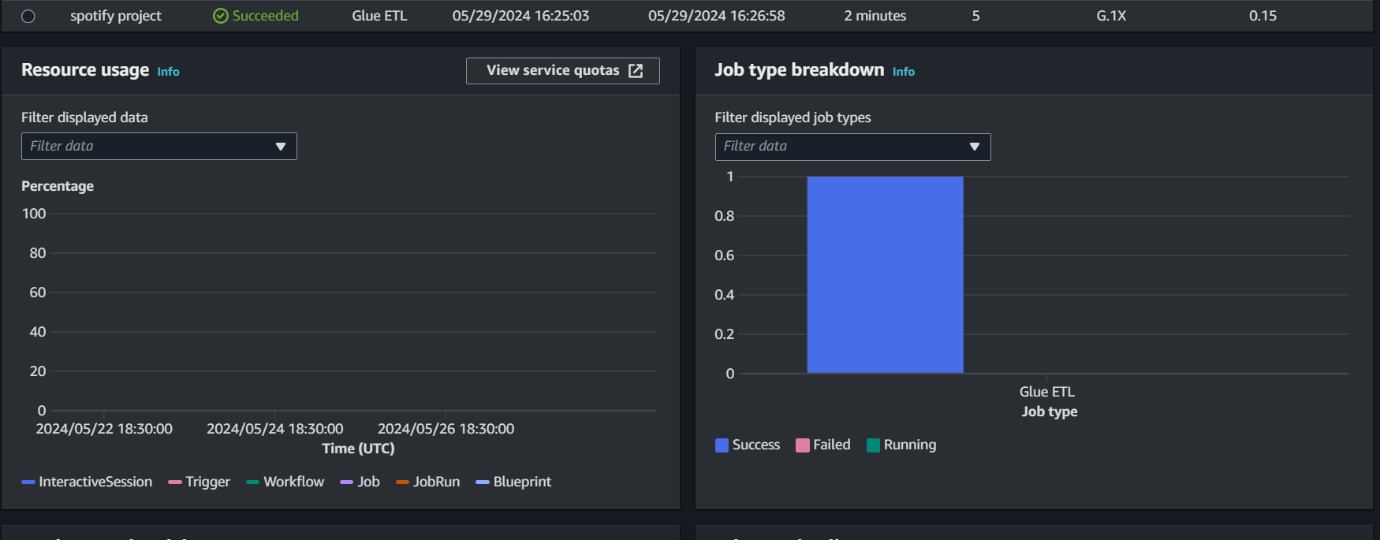
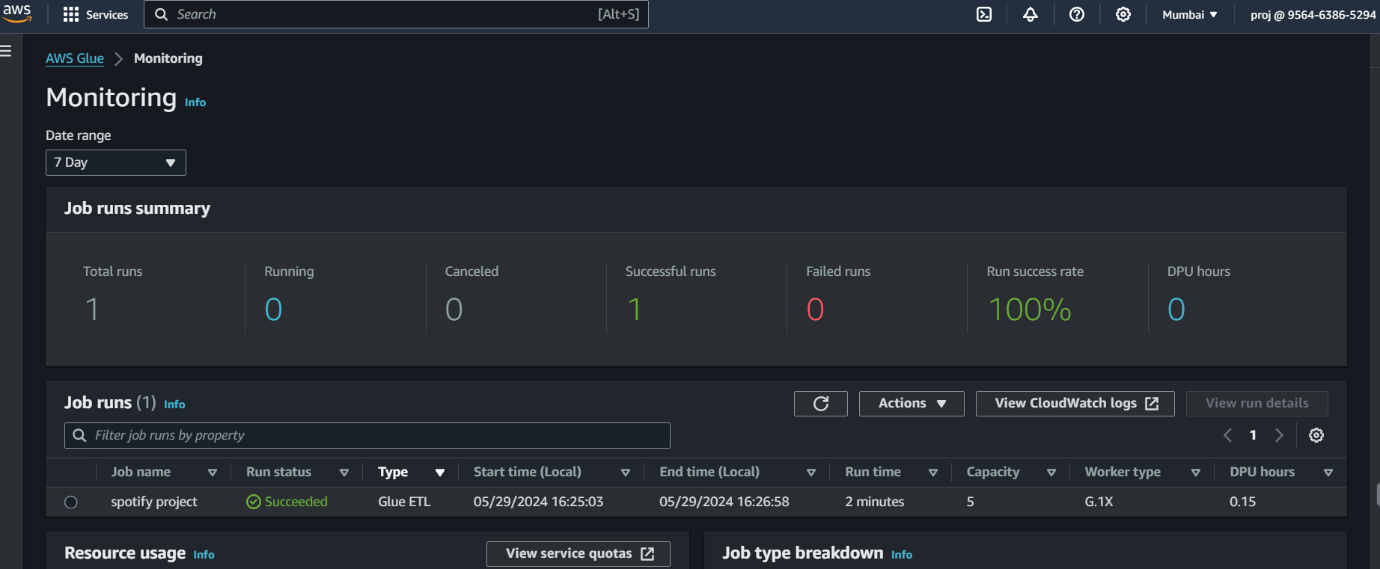
* Login to AWS root user
* Create an IAM user
  + Set username and password
* Set permissions
  + Select *attach policies directly*
  + Provide access to aws services (S3, glue, athena and quicksight )



* IAM user is created
* Sign in as IAM user by clicking on *Console sign -in URL*
* **STEP 2**: Create S3 bucket
* S3 staging and S3 data warehouse
* Creating two folders (staging and warehouse) inside bucket
* Put our data in staging (upload files in staging folder)



* **STEP 3**: Create AWS glue
* ETL (pipeline to transform and transfer data from staging to warehouse)
* creating visual ETL ---- visual ETL helps us to generate pyspark code visually using aws services)
* Our source is S3, consisting of three files (drag all of them).
* Join the artist and album files, with the condition artist\_id(album) and id(artist).
* Join the track and the previous join, with the condition track\_id(track).
* Drop unnecessary columns (drop fields).
* Put the result in our destination (target), which is currently S3, our data warehouse.
* The pipeline is now ready.
* Our script (pyspark) is automatically prepared
* **STEP 4** : Run and monitor ETL



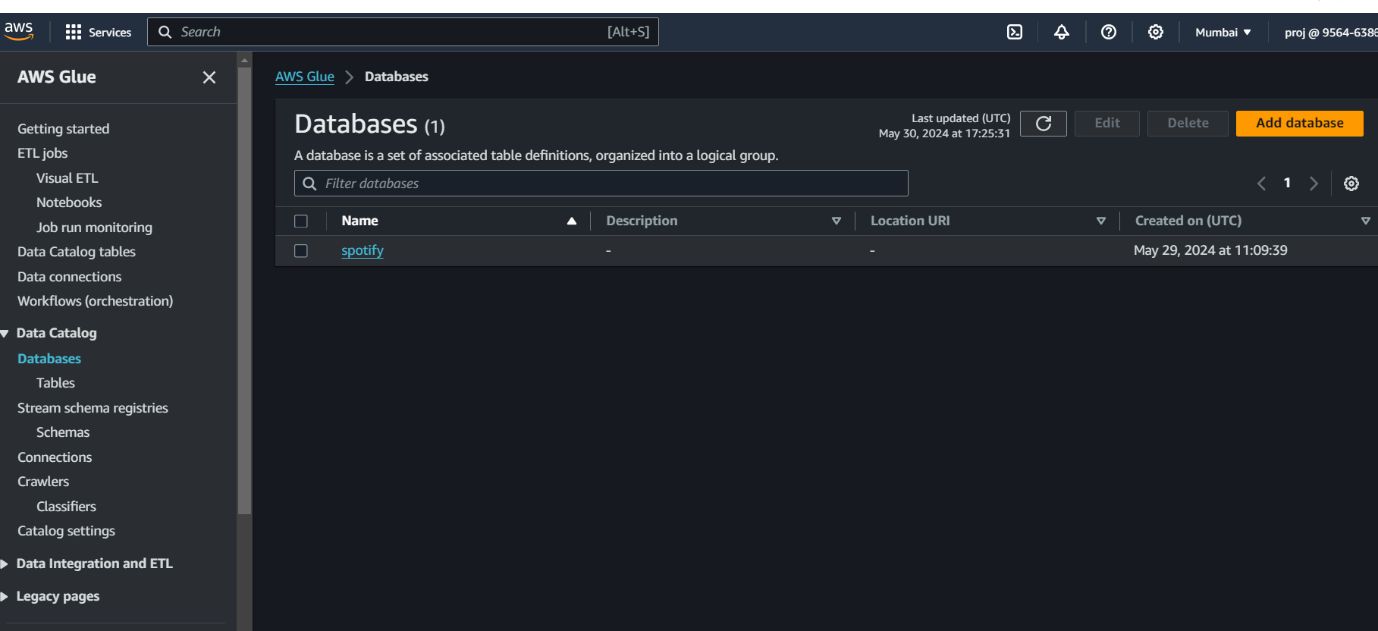
* **STEP 5:** Glue Crawler

The crawler generates metadata that enables services like Glue and Athena to perceive S3 information as a database with tables. Essentially, it facilitates the creation of a Glue catalog. Consequently, you can visualize the data within the S3 bucket as a structured database with tables. Therefore, any data residing in our S3 bucket can now be queried using standard SQL queries, provided the crawler has been executed over the database.

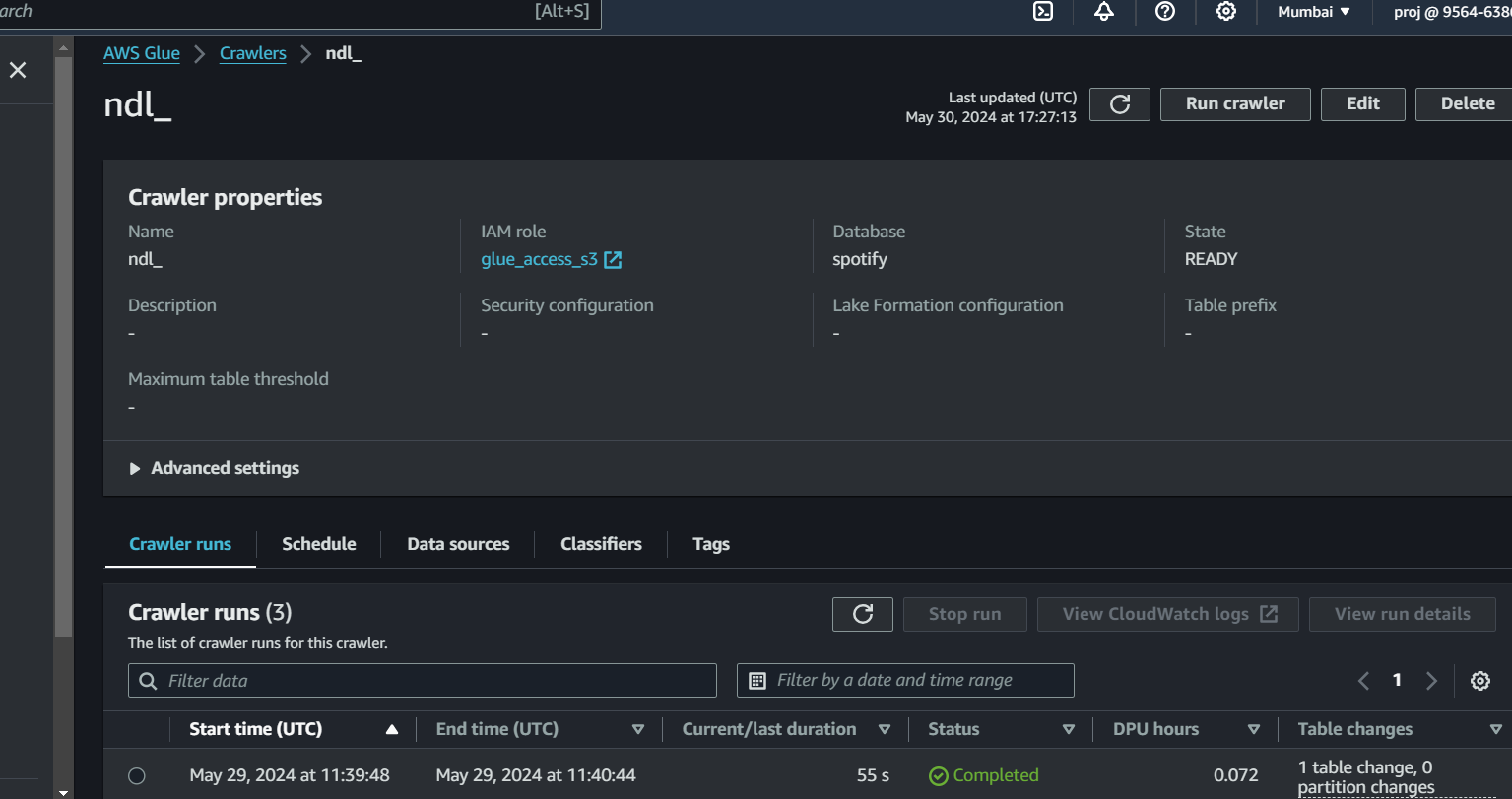
-Go to AWS glue

- Run the crawler

-Create catalog and database



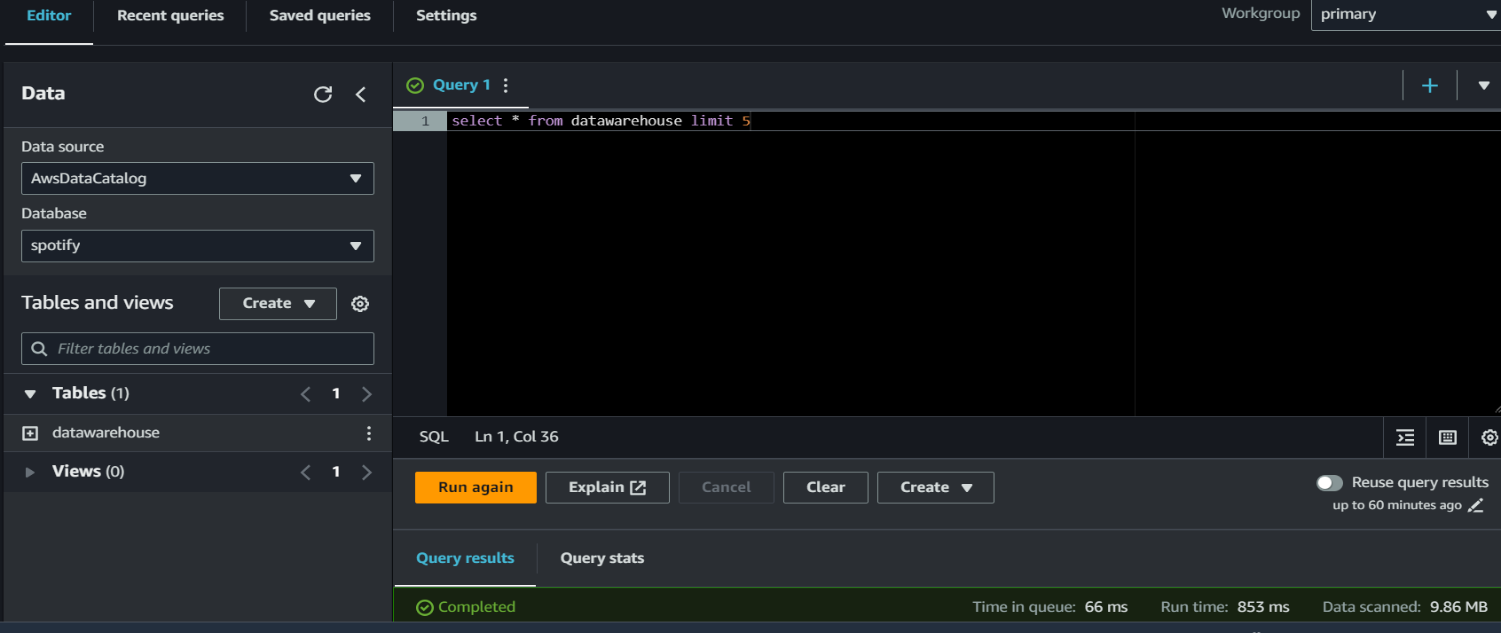
* Creating crawler
* Add data source
* run crawler to populate tables



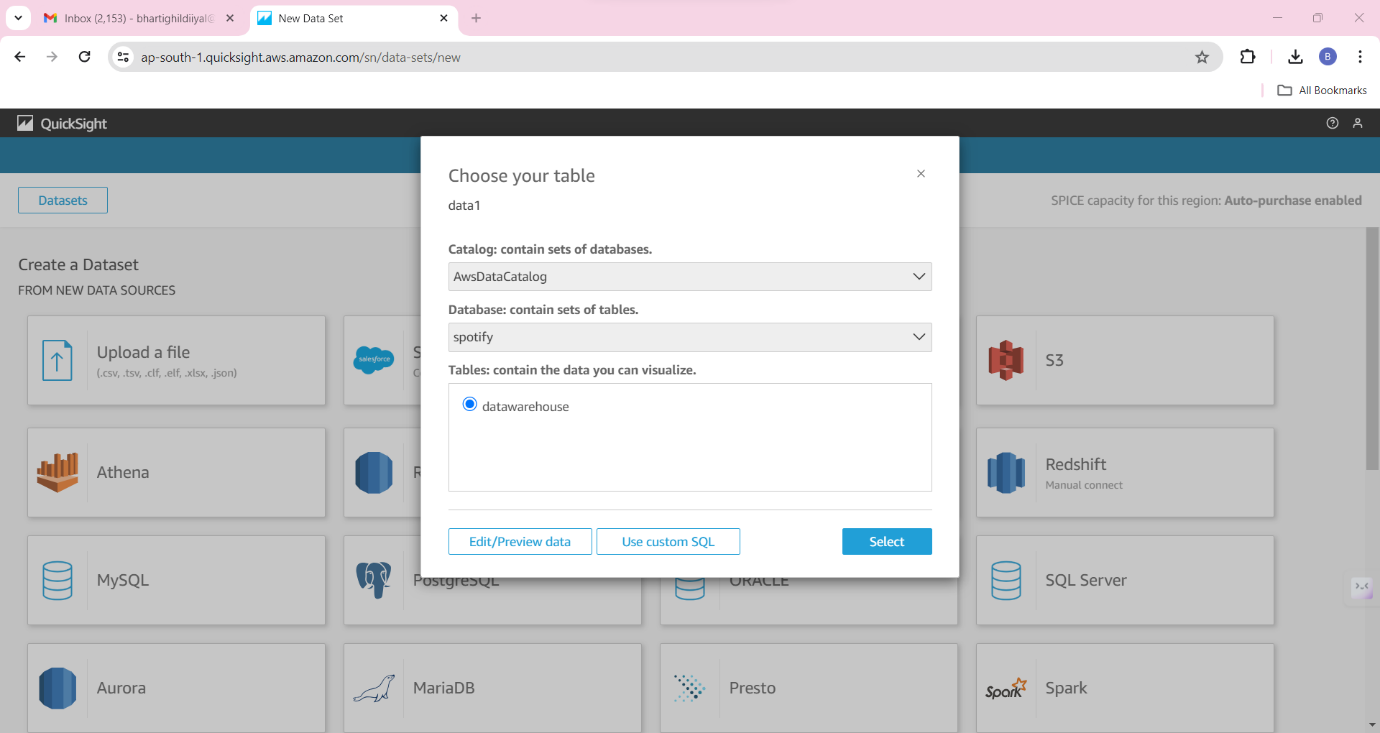
* **STEP 6** : AWS Athena (serverless analytical service)
* Go to athena
* Setup query editor
* Query using athena
* Athena is interactive query service that makes it easier to analyse directly from S3 bucket using simple SQL query.

Analyze your data using PySpark and Spark SQL.

* Launch the notebook editor.
* Navigate to the query editor.
* Write the query, but to execute it, we need to configure some settings, such as creating an S3 bucket to store all query results.
* Create an S3 bucket to store the output of all Athena queries.

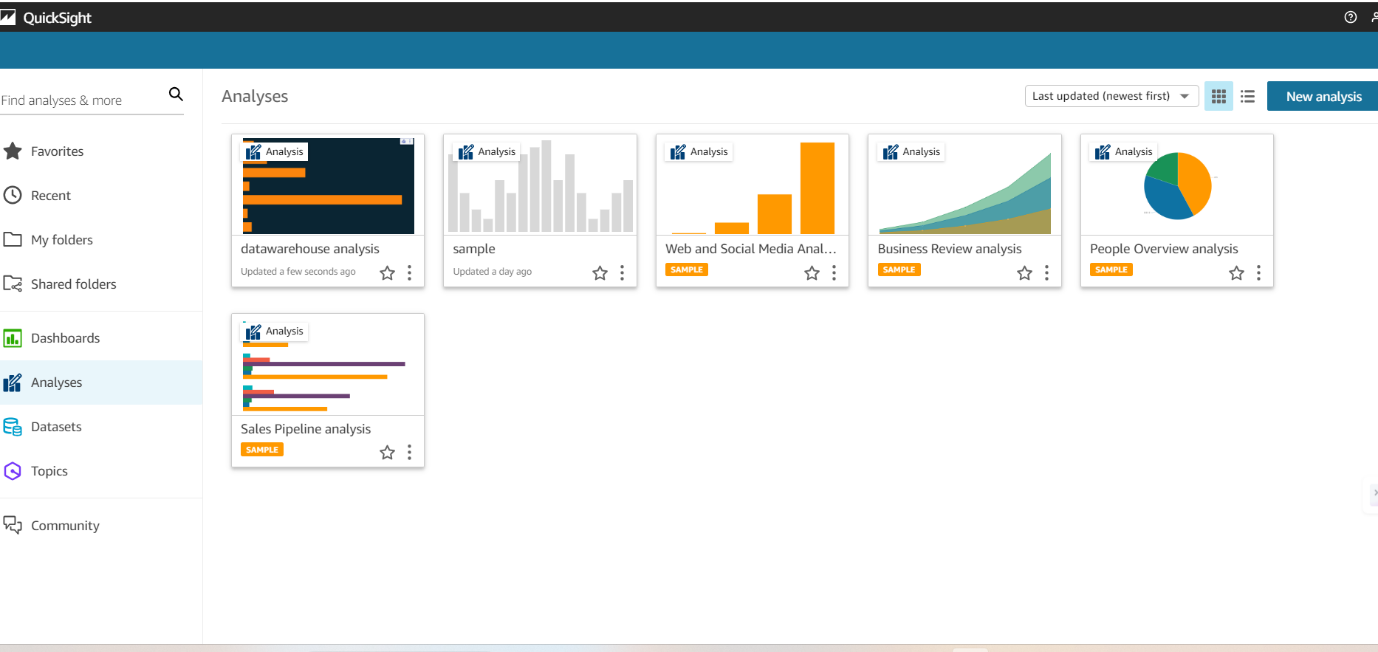


* **STEP 7** : QUICKSIGHT
  + Setup quicksight
  + Connect to DB



* Visualize your data

How business analyst people make use of quicksight to create visualise of data from S3 .



NOTE : And as we wrap up our AWS-powered Spotify adventure, remember to switch off QuickSight to keep your AWS bill in tune. Let's ensure our data exploration ends on a budget-friendly note!

