



Cloud Data Pipeline Mastery: End-to-End Analytics with AWS

“Building Scalable and Versatile Data Solutions for Any Dataset”

Sai Srivatsa Thangallapelly

Cloud Data Pipeline Mastery: End-to-End Analytics with AWS	1
Abstract	3
Introduction	3
Project Architecture Overview	3
Prerequisites	3
AWS Services Used	3
Data Source	3
Data Description	4
Hands-On Steps	4
Step 1: Setting Up AWS IAM User	4
Step 2: Creating S3 Buckets	7
Step 3: Setting Up AWS Glue for ETL	11
Step 4: Creating a Data Catalog with AWS Glue Crawler	20
Step 5: Querying Data with AWS Athena	24
Step 6: Visualizing Data with AWS QuickSight	27
Conclusion	30

Abstract

This project focuses on building an end-to-end data engineering pipeline using AWS services to analyze and visualize Spotify data. The project demonstrates how to ingest, process, store, and visualize large datasets, making them suitable for real-world applications in data-driven decision-making. The use of AWS Glue, S3, Athena, and QuickSight allows for an efficient, scalable, and cost-effective solution to process and analyze data, offering insights that can benefit various stakeholders in the music industry. This project was inspired and guided by the tutorials on the Date with Data.

Introduction

In this project, we will build an end-to-end data engineering pipeline using AWS cloud services. The project will focus on processing and analyzing Spotify data using various AWS tools like S3, Glue, Athena, and QuickSight.

Project Architecture Overview

1. **Staging Layer:** Raw data is stored in an S3 bucket.
2. **ETL Pipeline:** AWS Glue processes and transfers data from the staging layer to the data warehouse.
3. **Data Warehouse:** Processed data is stored in another S3 bucket.
4. **Data Catalog:** AWS Glue Crawler creates a database and tables for the data warehouse.
5. **Data Analysis:** AWS Athena queries the processed data.
6. **Data Visualization:** AWS QuickSight visualizes the data.

Prerequisites

- An AWS account
- Basic understanding of AWS services like S3, Glue, Athena, and QuickSight

AWS Services Used

- **Amazon S3:** For storing raw and processed data.
- **AWS Glue:** For building and managing ETL pipelines.
- **AWS Athena:** For querying data using SQL-like syntax.
- **AWS QuickSight:** For visualizing data.

Data Source

The data used in this project is sourced from the [Spotify Dataset 2023](#) available on Kaggle. The dataset, created by Tony Gordon Jr., includes detailed information about Spotify albums, artists, tracks, and various audio features like danceability, energy, loudness, and more. The dataset is available in CSV format and has been pre-processed for use in this project.

Data Description

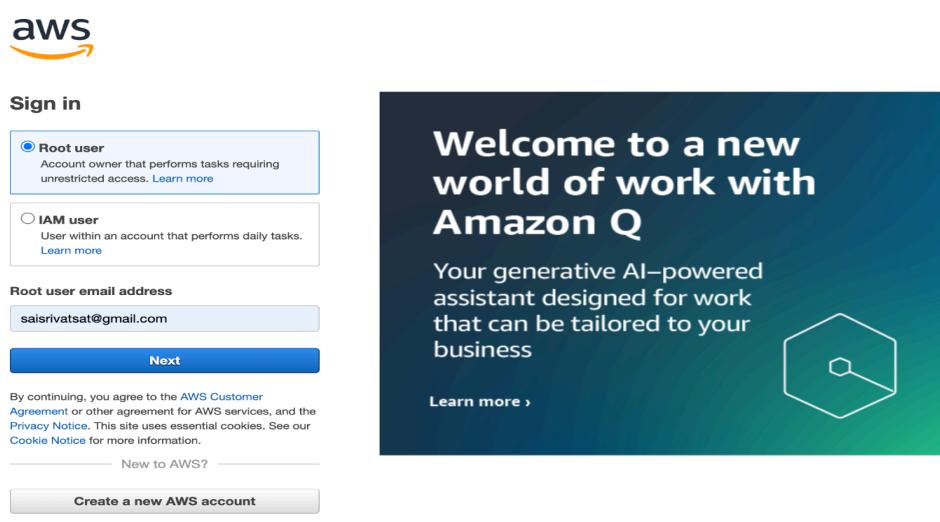
- **Albums:** Contains details of all the albums, including album ID, name, popularity, and release date.
- **Artists:** Contains information about the artists, including their names, number of followers, and genres.
- **Tracks:** Contains track-level data, including track ID, popularity, and other features like danceability and energy.
- **Spotify Features:** Contains various audio features like loudness, mode, speechiness, and valence.

Hands-On Steps

Step 1: Setting Up AWS IAM User

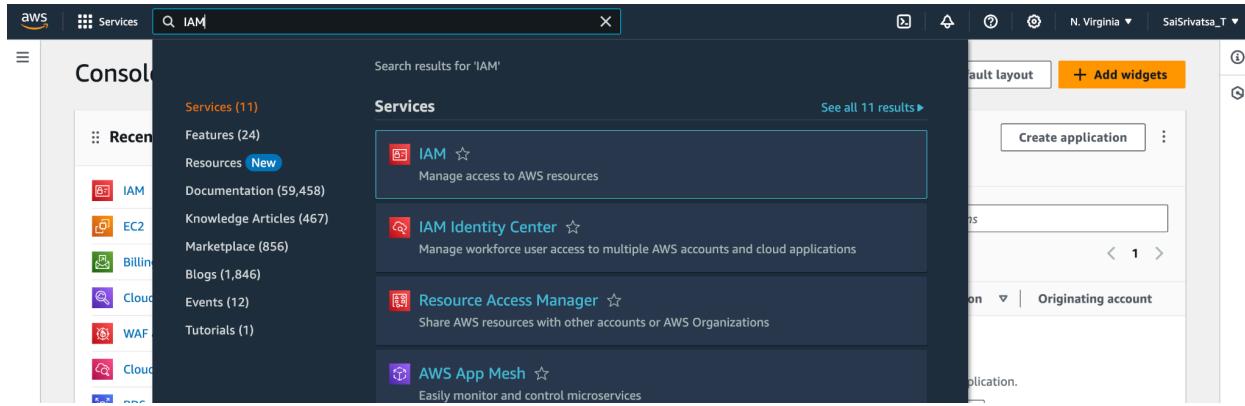
1. Log in to AWS Management Console:

- Open your web browser and go to the [AWS Management Console](#).
- Log in using your root account credentials.

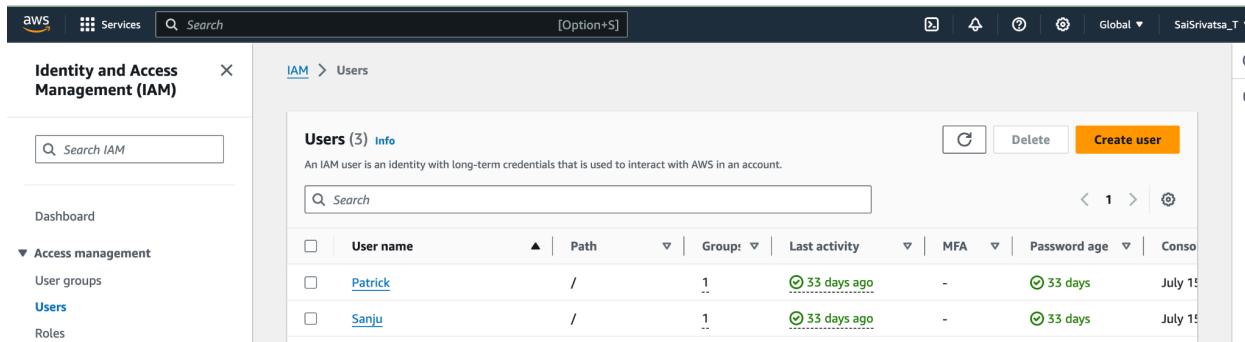


2. Create a New IAM User:

- In the AWS Management Console, search for "IAM" in the search bar and select the IAM service.



- In the IAM dashboard, click on "Users" from the left-hand menu and then click "Create user."



- Enter the Following Details**

- Username:** Enter `project_user`.
- Access Type:** Select "Provide user access to the AWS Management Console - optional"
- Console Password:** Choose "Custom password" and set a secure password.
- Password Reset:** Optionally, require the user to reset the password upon first login.

3. Assign Permissions:

- On the "Set permissions" page, select "Attach policies directly."

- Search for and select the following policies:

- AmazonS3FullAccess**
- AWSGlueConsoleFullAccess**
- AmazonAthenaFullAccess**
- AmazonQuickSightFullAccess**
- AWSQuickSightDescribeRDS**
- IAMFullAccess**(not included in the screenshot)

- Click "Next" and then "Create user."

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	Spotify_Project_IAM_User	Console password type	Custom password
			Require password reset Yes

Permissions summary

Name	Type	Used as
AmazonAthenaFullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy
AWSGlueConsoleFullAccess	AWS managed	Permissions policy
AWSQuicksightAthenaAccess	AWS managed	Permissions policy
AWSQuickSightDescribeRDS	AWS managed	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create user

4. Log in as IAM User:

- Log out of your root account.
- Log in to the AWS Management Console using the newly created IAM user credentials.

aws Services Search [Option+S] Ohio Spotify_Project_IAMUser @ 8517-2547-6155 ▾

Console Home Info

Recently visited Info

No recently visited services

Explore one of these commonly visited AWS services.

EC2 S3 RDS Lambda

View all services

Applications (0) Info

Create application

Region: US East (Ohio)

us-east-2 (Current Region) Find applications

Name Description Region Originating account

Access denied

Go to myApplications

Welcome to AWS Info

Getting started with AWS

Learn the fundamentals and find valuable information to get the most out of AWS.

AWS Health Info

Cost and usage Info

Current month costs Access denied

Cost breakdown Access denied

Forecasted month end costs Access denied

Step 2: Creating S3 Buckets

1. Navigate to S3 Service:

- In the AWS Management Console, search for “S3” in the search bar and select the S3 service.

The screenshot shows the AWS Management Console search results for 's3'. The search bar at the top has 's3' typed into it. Below the search bar, the 'Services' section is displayed, showing a list of services. 'S3' is the first item in the list, followed by 'S3 Glacier', 'AWS Snow Family', and 'Storage Gateway'. To the right of the search results, there is a sidebar with options like 'Create application' and 'Originating account'.

2. Create a New S3 Bucket:

- Click on "Create bucket."

The screenshot shows the Amazon S3 service page. The left sidebar includes links for Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens (with sub-links for Dashboards, Storage Lens groups, and AWS Organizations settings), and a Feature spotlight. The main content area features an 'Account snapshot - updated every 24 hours' section with a link to 'All AWS Regions' and a 'View Storage Lens dashboard' button. Below this is a table titled 'General purpose buckets' showing one entry: 'myglobals3' (Name), 'US East (N. Virginia) us-east-1' (AWS Region), and 'View analyzer for us-east-1' (IAM Access Analyzer). The table also includes columns for Creation date (August 2, 2024, 18:21:28 (UTC-05:00)) and actions (Edit, Copy ARN, Empty, Delete, Create bucket).

- **Bucket Name:** Enter {GlobalUnique|NoCaps|NoUnderscore}.
- **Region:** Select the region closest to you.

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region
US East (Ohio) us-east-2

Bucket name Info

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

- Leave all other settings as default and scroll down to click "Create bucket."

3. Create Folders in the S3 Bucket:

- Click on the bucket **project-data** you just created.

Successfully created bucket "spotify-aws-prjct"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

View details X

Amazon S3 > Buckets

▶ Account snapshot - *updated every 24 hours* All AWS Regions
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

[General purpose buckets](#) [Directory buckets](#)

General purpose buckets (2) <small>Info All AWS Regions</small>			
Buckets are containers for data stored in S3.			
Name	AWS Region	IAM Access Analyzer	Creation date
myglobals3	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 2, 2024, 18:21:28 (UTC-05:00)
spotify-aws-prjct	US East (Ohio) us-east-2	View analyzer for us-east-2	August 17, 2024, 21:44:42 (UTC-05:00)

[Create bucket](#)

- Click on "Create folder."

Amazon S3 > Buckets > spotify-aws-prjct

spotify-aws-prjct Info

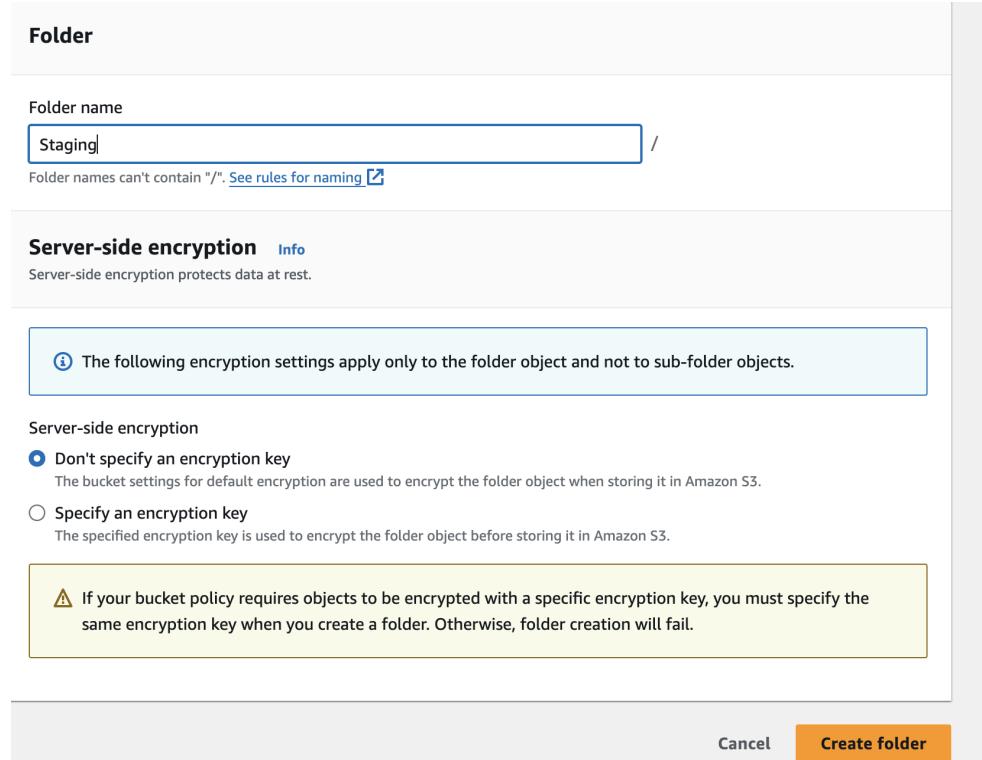
[Objects \(0\) Info](#) [Actions ▾](#) [Create folder](#) [Upload](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Name	Type	Last modified	Size	Storage class
No objects You don't have any objects in this bucket.				

[Upload](#)

- **Folder Name:** Enter **staging** and click "Create folder."



- Repeat the above step to create another folder named **data-warehouse**.

Name	Type	Last modified	Size	Storage class
datawarehouse/	Folder	-	-	-
staging/	Folder	-	-	-

4. Upload Pre-Processed CSV Files:

In real-time data in a staging layer will be coming from through Dynamo DB or our database instance, but for this project, we are not making use of Dynamo DB or database hence we are adding our data manually

- Click on the **staging** folder.
- Click "Upload" and then "Add files."
- Select the pre-processed CSV files (**albums.csv**, **artists.csv**, **tracks.csv**) from your local machine.

The screenshot shows the AWS S3 'Upload' interface. In the 'Files and folders' section, three CSV files are listed: 'spotify_artist_data_2023.csv', 'spotify_tracks_data_2023.csv', and 'spotify-albums_data_2023.csv'. Below the table is a 'Destination' section with the URL 's3://spotify-aws-prjct/staging/'. At the bottom right is a prominent orange 'Upload' button.

- Click "Upload" to upload the files to the **staging** folder.

The screenshot shows the AWS S3 'Upload: status' page. It displays a summary table with one row: 'Succeeded' with '3 files, 110.0 MB (100.00%)'. Below this is a 'Files and folders' table showing the same three CSV files from the previous step, all marked as 'Succeeded'.

Summary					
Destination	Succeeded	Failed			
s3://spotify-aws-prjct/staging/	3 files, 110.0 MB (100.00%)	0 files, 0 B (0%)			

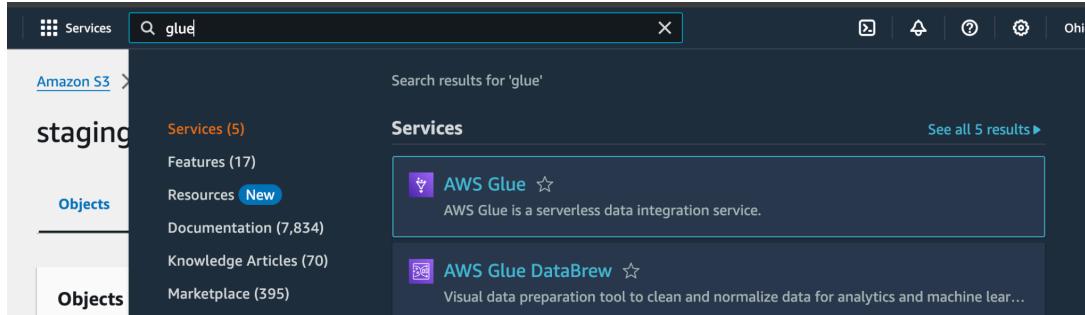
Files and folders (3 Total, 110.0 MB)						
Name	Folder	Type	Size	Status	Error	
spotify_artis...	-	text/csv	2.4 MB	Succeeded	-	
spotify_trac...	-	text/csv	13.1 MB	Succeeded	-	
spotify-albu...	-	text/csv	94.6 MB	Succeeded	-	

Step 3: Setting Up AWS Glue for ETL

In this step, we'll try to create a data pipeline that will transfer our data from the staging layer to the data warehouse. we'll be making use of AWS glue to create a data pipeline. AWS glue is a managed service provided by AWS to create a data pipeline.

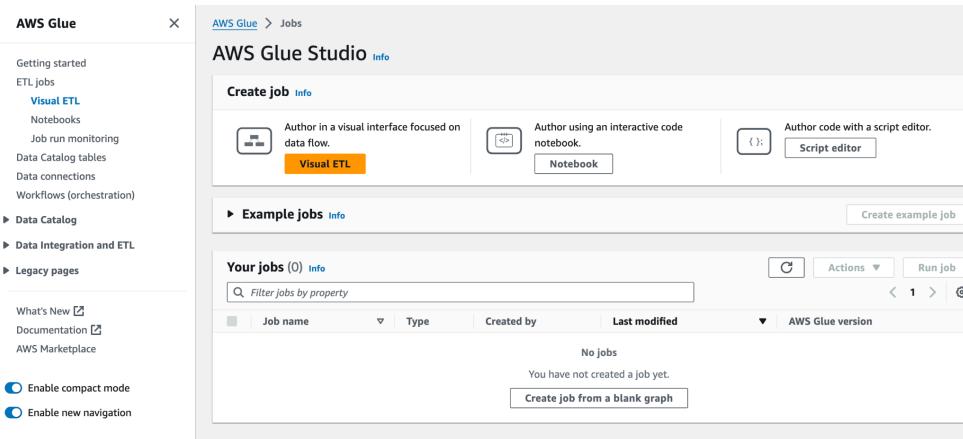
1. Navigate to AWS Glue Service:

- In the AWS Management Console, search for “Glue” in the search bar and select the AWS Glue service.



2. Create a New Glue Job:

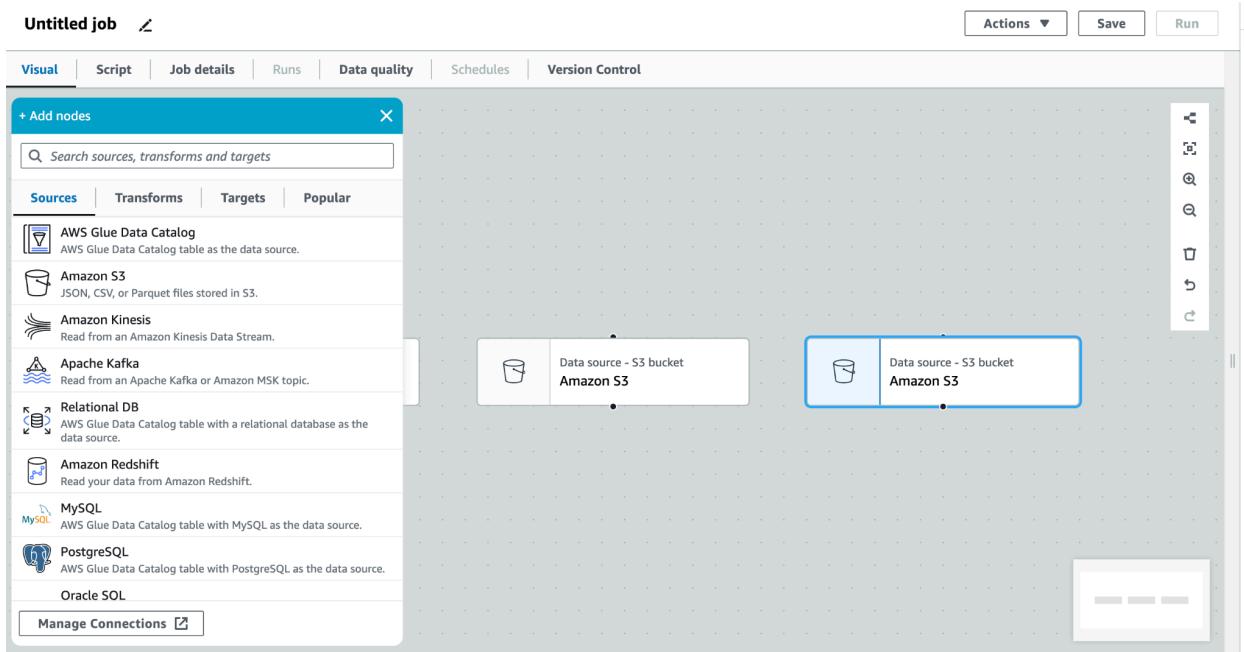
- In the AWS Glue dashboard, click on "Visual ETL" under the "ETL Jobs" section.



- Click "Visual ETL." under Create job.

3. Set Up Data Sources:

- Drag and drop source and destination S3 buckets, as we have three CSV files (`albums.csv`, `artists.csv`, `tracks.csv`)



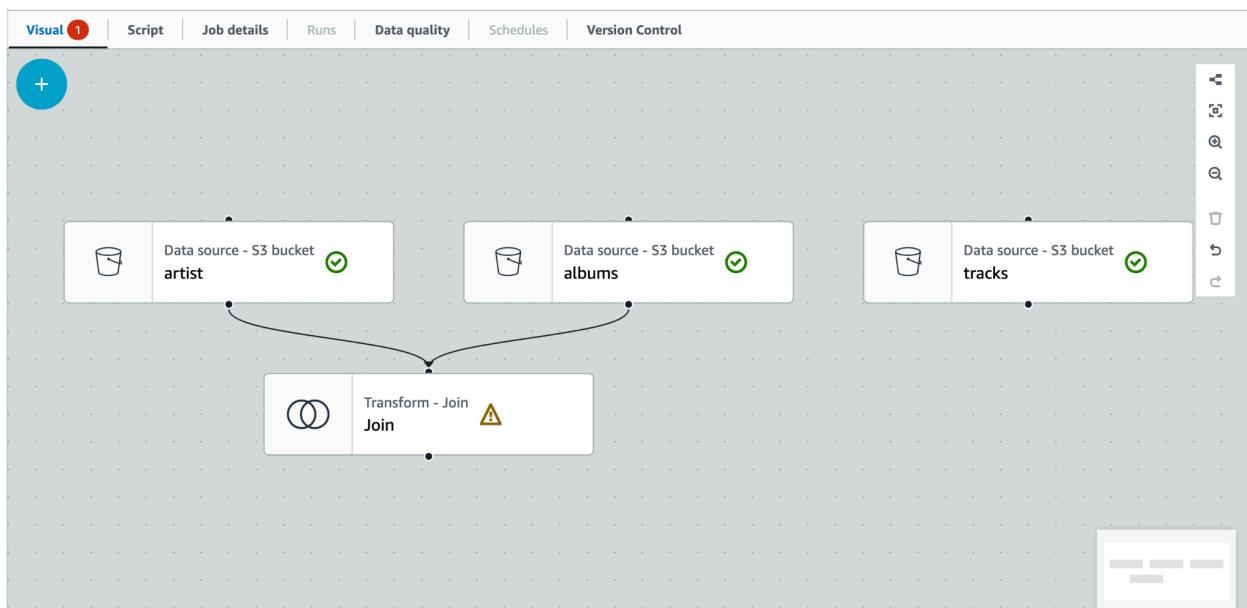
- Select the First Amazon S3 bucket and rename it with “artist” and click on browse and select the file from the **s3://spotify-aws-prjct/staging/spotify_artist_data_2023.csv**. Select the Data format as CSV.

- Repeat the same steps for the other 2 S3 Buckets.

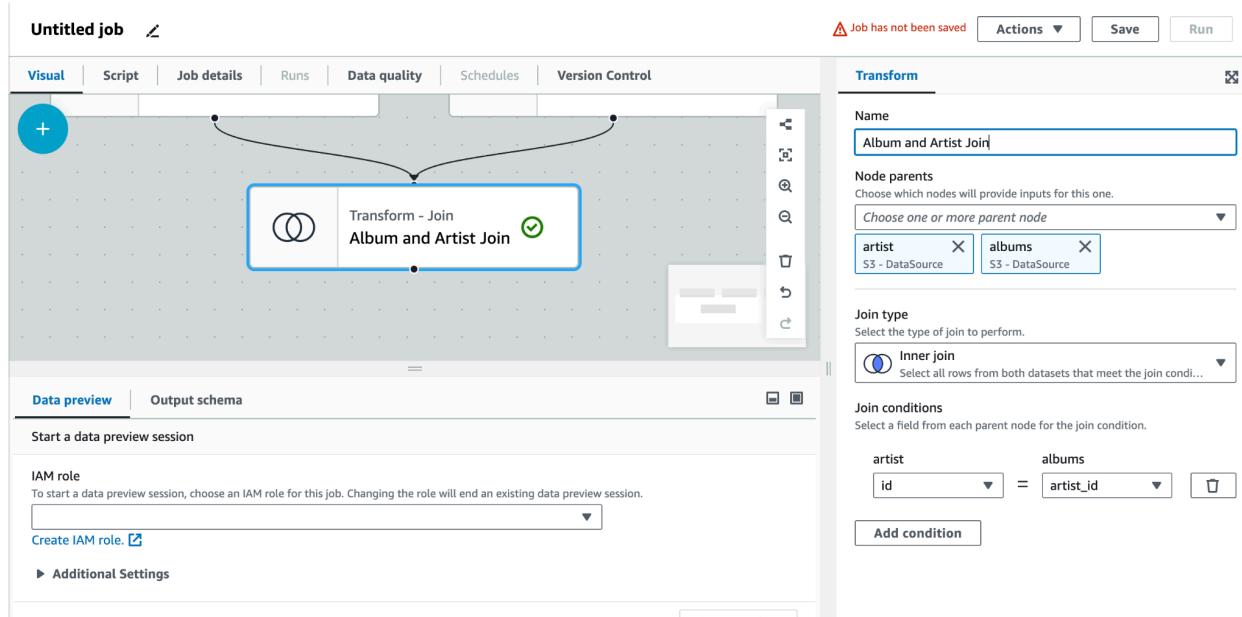
The screenshot shows the AWS Glue Data Transformation interface. At the top, there's a navigation bar with tabs: Visual, Script, Job details, Runs, Data quality, Schedules, and Version Control. The Visual tab is selected. In the main workspace, there are three data source nodes: "Data source - S3 bucket artist", "Data source - S3 bucket albums", and "Data source - S3 bucket tracks". A context menu is open over the "tracks" node. To the right, a panel titled "Data source properties - S3" is displayed, containing fields for Name (set to "tracks"), S3 source type (set to "S3 location"), S3 URL (set to "s3://spotify-aws-prjct/staging/spot"), and other settings like Data format (CSV), Delimiter (Comma), and Escape character (optional). A warning message at the top right says "Job has not been saved".

4. Configure Data Transformations:

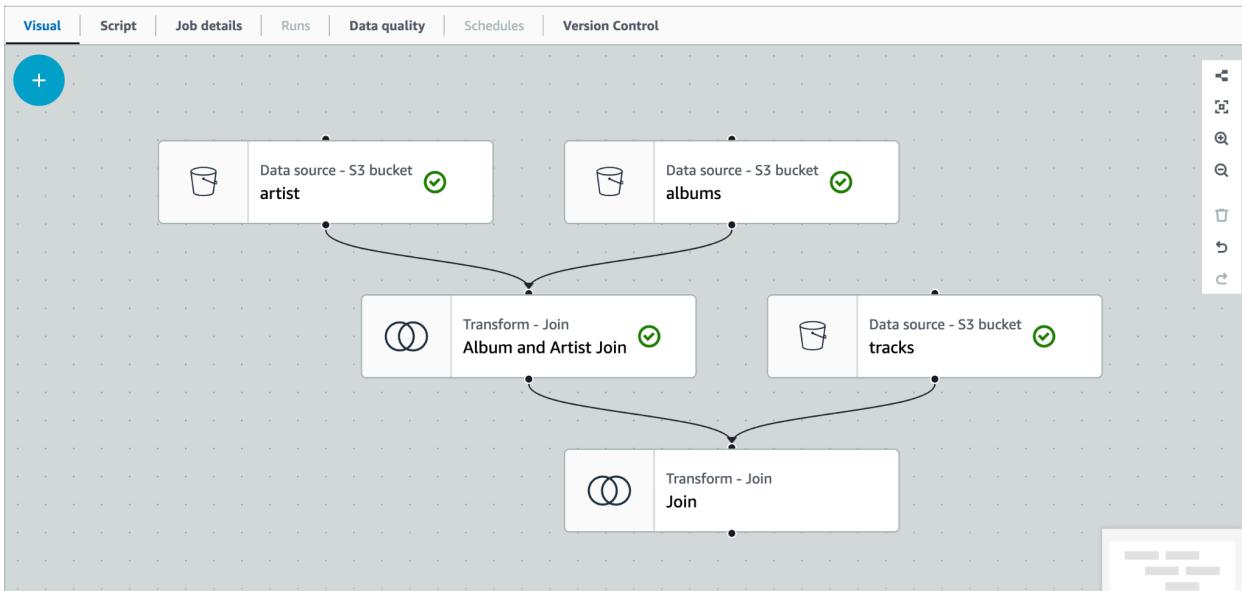
- Now to join album and artist, click on the ADD symbol, select join from **Transforms** and connect nodes as per the image below.



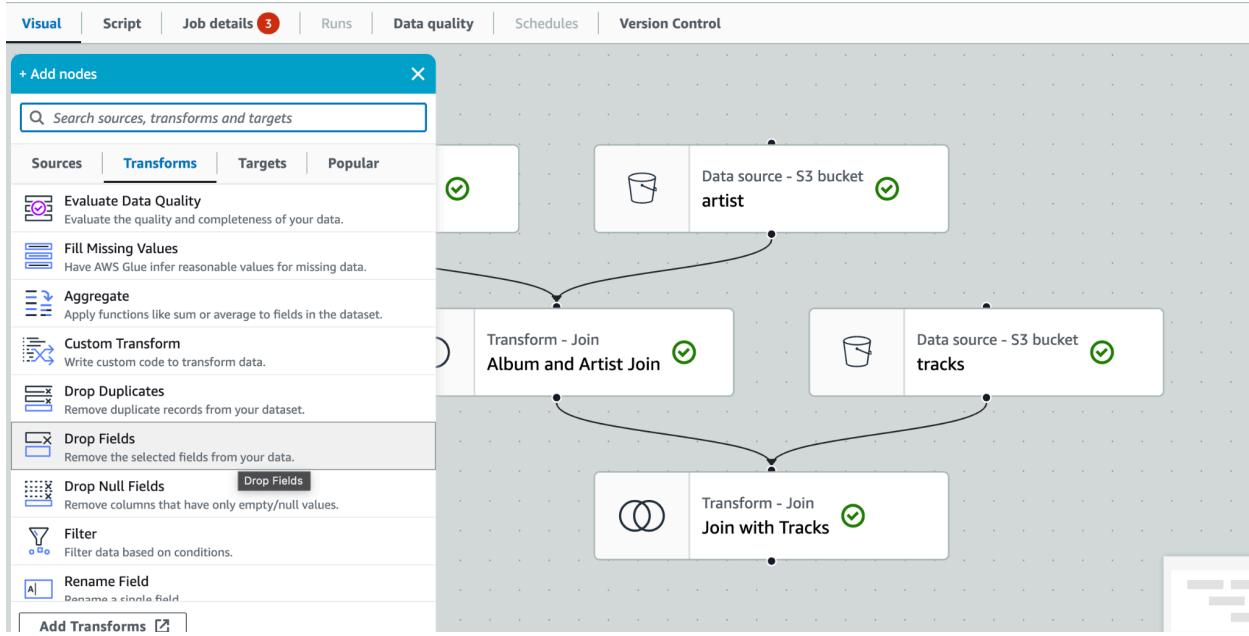
- After Joining the nodes of both the buckets to the Join Transform. Add a Condition where *artist 'id'* = *albums 'artist_id'* and rename the join “**Album and Artist Join**”.



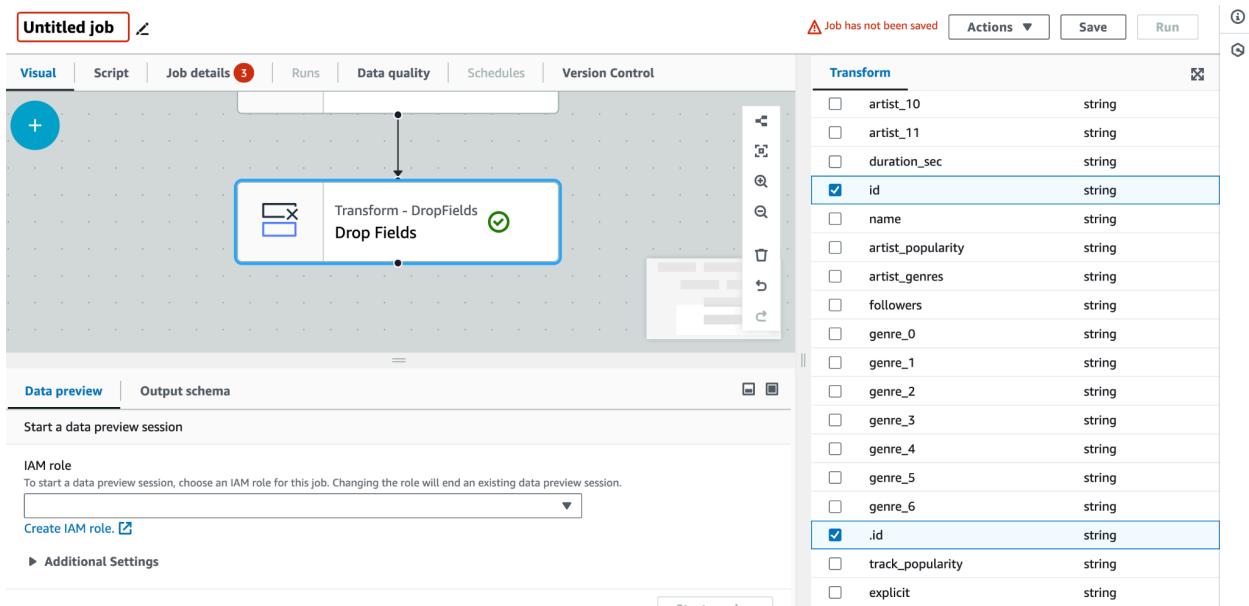
- Now add another Join Transfrom to join ‘track’ s3 bucket and ‘Album and Artist’ Join



- Now select the Join and add the condition *Album and Artist Join ‘track_id’ = tracks ‘id’*
- and rename the join as ‘Join with Tracks’.
- To drop unnecessary columns, select **Drop Fields** from Transforms node.

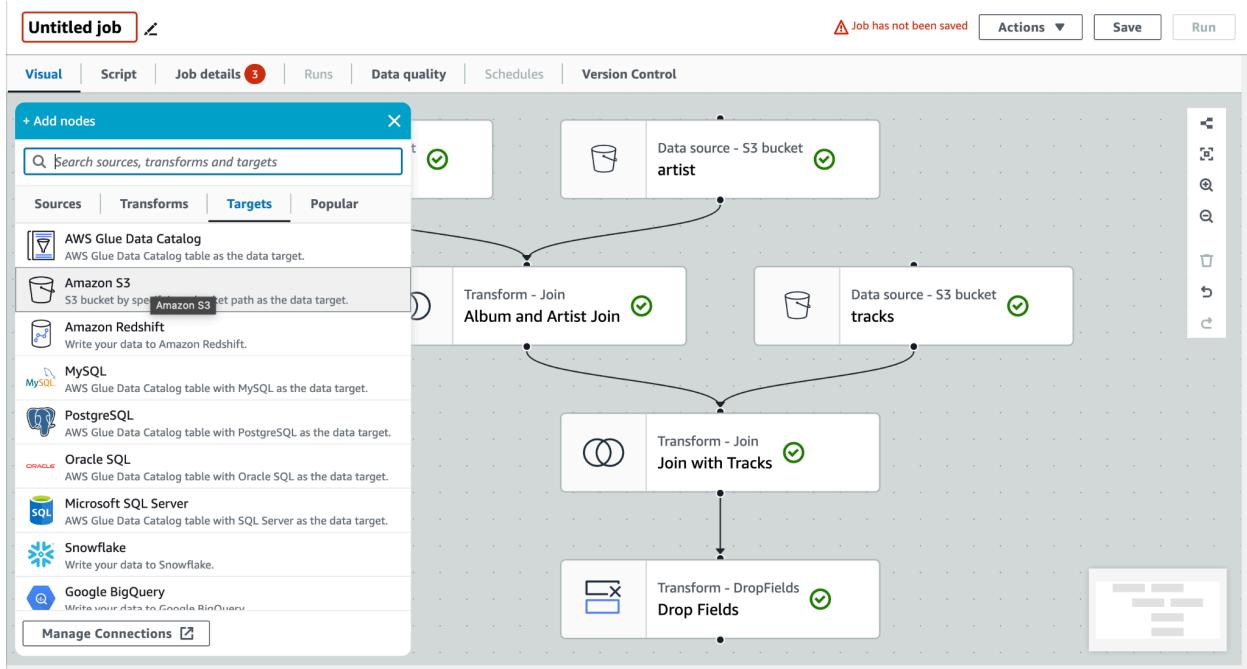


- We remove duplicate columns and Identical columns we dont need. Here id is inner join to artist id so select id.



5. Set Up Data Target:

- Next step is to add the destination. Select Amazon S3 bucket in the Targets Section and add



- Rename the Destination node and add the Target location “s3://spotify-aws-prjct/datawarehouse/” and make sure the compression type is Snappy.
- Add the **Job Name**: Enter a name like **Spotify Project**.
- As there is No IAM role. Login to root user and create an IAM role using below steps in the screenshots.

Step 1
Select trusted entity

Trusted entity type

- AWS service
- AWS account
- Web identity
- SAML 2.0 federation
- Custom trust policy

Use case

Choose a use case for the specified service.
Use case

- Glue

Cancel **Next**

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

Permissions policies (1/946) Info

Choose one or more policies to attach to your new role.

Policy name	Type	Description
<input type="checkbox"/> AmazonDMSRedshiftS3Role	AWS managed	Provides access to manage S3 settings ...
<input checked="" type="checkbox"/> AmazonS3FullAccess	AWS managed	Provides full access to all buckets via t...
<input type="checkbox"/> AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	Provides AWS Lambda functions permi...
<input type="checkbox"/> AmazonS3OutpostsFullAccess	AWS managed	Provides full access to Amazon S3 on ...
<input type="checkbox"/> AmazonS3OutpostsReadOnlyAccess	AWS managed	Provides read only access to Amazon S...
<input type="checkbox"/> AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all bucket...
<input type="checkbox"/> AWSBackupServiceRolePolicyForS3Backup	AWS managed	Policy containing permissions necessar...
<input type="checkbox"/> AWSBackupServiceRolePolicyForS3Restore	AWS managed	Policy containing permissions necessar...
<input type="checkbox"/> QuickSightAccessForS3StorageManagementA...	AWS managed	Policy used by QuickSight team to acc...

Filter by Type All types 9 matches

▶ Set permissions boundary - optional

Cancel Previous Next

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+,-,_-' characters.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=., @-/[\{\}]\\$\%^\^0-`~`

Step 1: Select trusted entities Edit

Trust policy

```

1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Principal": {
7-         "Service": "glue.amazonaws.com"
8-       },
9-       "Action": "sts:AssumeRole"
10-    }
11-  ]
}

```

- **IAM Role:** Select the IAM role that has the required permissions.
- **Type:** Choose "Spark."
- **Glue Version:** Choose the latest Glue version available.
- **Python Version:** Select Python 3.
- **Script Path:** Leave the default script path or specify a path in your S3 bucket.
- **Click Save**

Spotify Project

Job details

Basic properties

Name: Spotify Project

Description - optional:

IAM Role: glue_s3_access

Type: Spark

Glue version: Glue 4.0 - Supports spark 3.3, Scala 2, Python 3

Language: Python 3

6. Run the Glue Job:

- Review your job settings.
- Click "Run job" to start the ETL process, transforming and moving data from the staging layer to the data warehouse.

Spotify Project

Runs

Successfully started job

Successfully started job Spotify Project. Navigate to [Run details](#) for more details.

Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacity (DPU)	Worker type	Glue version
Running	0	08/17/2024 23:22:43	-	1 s	5 DPU	G.1X	4.0

Monitoring

Job runs summary

Total runs	Running	Canceled	Successful runs	Failed runs	Run success rate	DPU hours
1	0	0	1	0	100%	0

Job runs (1) Info

Job name	Run status	Type	Start time (Local)	End time (Local)	Run time	Capacity	Worker type	DPU hours
Spotify Project	Succeeded	Glue ETL	08/17/2024 23:22:43	08/17/2024 23:24:56	2 minutes	5	G.1X	0.17

Resource usage

Job type breakdown

- Check S3 bucket if the files are Parsed

Amazon S3 > Buckets > spotify-aws-prjct > datawarehouse/

Objects (16) Info

Name	Type	Last modified	Size	Storage class
run-1723955064689-part-block-0-r-00000-snappy.parquet	parquet	August 17, 2024, 23:24:39 (UTC-05:00)	3.8 MB	Standard
run-1723955064689-part-block-0-r-00001-snappy.parquet	parquet	August 17, 2024, 23:24:39 (UTC-05:00)	3.9 MB	Standard

Step 4: Creating a Data Catalog with AWS Glue Crawler

1. Create a New Crawler:

- In the AWS Glue dashboard, click on "Crawlers" under the "Data catalog" section.
- Click "Create crawler."

AWS Glue > Crawlers

Crawlers (0) Info

Name	State	Schedule	Last run	Last run timestamp	Log	Table changes...
spotify_crawler	Not running					

- **Crawler Name:** Enter `spotify_crawler`.

AWS Glue > Crawlers > Add crawler

Step 1
Set crawler properties

Step 2
Choose data sources and classifiers

Step 3
Configure security settings

Step 4
Set output and scheduling

Step 5
Review and create

Set crawler properties

Crawler details Info

Name: Name can be up to 255 characters long. Some character set including control characters are prohibited.

Description - optional:
 Descriptions can be up to 2048 characters long.

Tags - optional
Use tags to organize and identify your resources.

Cancel **Next**

- **Data Store:** Select S3 and provide the path to the data-warehouse folder.

AWS Glue > Crawlers > Add crawler

Step 1
Set crawler properties

Step 2
Choose data sources and classifiers

Step 3
Configure security settings

Step 4
Set output and scheduling

Step 5
Review and create

Add data source

Data source
Choose the source of data to be crawled.

S3

Network connection - optional
Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any other S3 targets will also use the same connection (or none, if left blank).

Location of S3 data
 In this account
 In a different account

S3 path
Browse for or enter an existing S3 path.

 All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Subsequent crawler runs
This field is a global field that affects all S3 data sources.

Crawl all sub-folders
Crawl all folders again with every subsequent crawl.

Crawl new sub-folders only
Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.

Crawl based on events
Rely on Amazon S3 events to control what folders to crawl.

Sample only a subset of files

Exclude files matching pattern

Cancel **Add an S3 data source**

AWS Glue > Crawlers > Add crawler

Step 1
Set crawler properties

Step 2
Choose data sources and classifiers

Step 3
Configure security settings

Step 4
Set output and scheduling

Step 5
Review and create

Choose data sources and classifiers

Data source configuration

Is your data already mapped to Glue tables?

Not yet
Select one or more data sources to be crawled.

Yes
Select existing tables from your Glue Data Catalog.

Data sources (1) Info
The list of data sources to be scanned by the crawler.

Type	Data source	Parameters
<input type="radio"/> S3	s3://spotify-aws-prjct/datawarehouse/	Recrawl all

Custom classifiers - optional
A classifier checks whether a given file is in a format the crawler can handle. If it is, the classifier creates a schema in the form of a StructType object that matches that data format.

Cancel Previous Next

- **IAM Role:** Select the IAM role we created earlier and before this please do add another policy “AWSGlueServiceRole” to this role.

IAM > Roles > glue_s3_access > Add permissions

Attach policy to glue_s3_access

▶ Current permissions policies (1)

Other permissions policies (1/945)

Policy name	Type	Description
<input checked="" type="checkbox"/> AWSGlueServiceRole	AWS managed	Policy for AWS Glue service role which ...

Filter by Type: All types | 1 match | < 1 > | C | Description

Cancel Add permissions

AWS Glue > Crawlers > Add crawler

Step 1
Set crawler properties

Step 2
Choose data sources and classifiers

Step 3
Configure security settings

Step 4
Set output and scheduling

Step 5
Review and create

Configure security settings

IAM role Info

Existing IAM role

Choose an IAM role
glue_s3_access

Allows Glue to call AWS services on your behalf.

View C

Lake Formation configuration - optional
Allow the crawler to use Lake Formation credentials for crawling the data source. [Learn more.](#)

Use Lake Formation credentials for crawling S3 data source
Checking this box will allow the crawler to use Lake Formation credentials for crawling the data source. If the data source is registered in another account, you must provide the registered account ID. Otherwise, the crawler will crawl only those data sources associated to the account. Only applicable to S3, Glue Catalog, Iceberg, and Hudi data sources.

Security configuration - optional
Enable at-rest encryption with a security configuration.

Cancel Previous Next

2. Create a New Database:

- Open the duplicate tab. Go to the AWS Glue > Data catalog >> Databases, and create a new database.

The screenshot shows the 'Create a database' wizard in AWS Glue. On the left, a sidebar lists various Glue services: Getting started, ETL jobs, Visual ETL, Notebooks, Job run monitoring, Data Catalog tables, Data connections, Workflows (orchestration), Data Catalog (selected), Databases (selected), Tables, Stream schema registries, Schemas, Connections, Crawlers, Classifiers, Catalog settings, Data Integration and ETL, and Legacy pages. The main area is titled 'Create a database' with the sub-instruction 'Create a database in the AWS Glue Data Catalog.' It has two sections: 'Database details' and 'Database settings'. In 'Database details', the 'Name' field is filled with 'spotify_data'. A note says 'Database name is required, in lowercase characters, and no longer than 255 characters.' Below it is a 'Description - optional' field with placeholder 'Enter text' and a note 'Descriptions can be up to 2048 characters long.' In 'Database settings', there is a 'Location - optional' field with placeholder 'Set the URI location for use by clients of the Data Catalog.' At the bottom right are 'Cancel' and 'Create database' buttons.

- **Database Name:** Enter `spotify_data`.
- Back to the Crawlers, select the created database
- Click Next and Create the Crawler.

The screenshot shows the 'Review and create' wizard for creating a crawler. The left sidebar lists steps: Step 1 (Set crawler properties, currently selected), Step 2 (Choose data sources and classifiers), Step 3 (Configure security settings), Step 4 (Set output and scheduling), and Step 5 (Review and create). The main area is divided into five sections corresponding to these steps. Step 1: 'Step 1: Set crawler properties' shows a table with one row: Name (spotify_crawler), Description (-), and Tags (-). Step 2: 'Step 2: Choose data sources and classifiers' shows a table for 'Data sources (1) info' with one row: Type (S3), Data source (s3://spotify-aws-prjct/datawarehouse/), and Parameters (Recrawl all). Step 3: 'Step 3: Configure security settings' shows a table with three rows: IAM role (glue_s3_access), Security configuration (-), and Lake Formation configuration (-). Step 4: 'Step 4: Set output and scheduling' shows a table with four rows: Database (spotify_data), Table prefix - optional (-), Maximum table threshold - optional (-), and Schedule (On demand). At the bottom right are 'Cancel', 'Previous', and 'Create crawler' buttons.

3. Run the Crawler:

- After setting up the crawler, click "Run crawler."

Crawler successfully starting
The following crawler is now starting: "spotify_crawler"

AWS Glue > Crawlers > spotify_crawler

spotify_crawler

Last updated (UTC) August 19, 2024 at 18:22:42 | [Run crawler](#) | [Edit](#) | [Delete](#)

Crawler properties			
Name spotify_crawler	IAM role glue_s3_access	Database spotify_data	State READY
Description -	Security configuration -	Lake Formation configuration -	Table prefix -
Maximum table threshold -			
Advanced settings			

[Crawler runs](#) | [Schedule](#) | [Data sources](#) | [Classifiers](#) | [Tags](#)

Crawler runs (1 -)
The list of crawler runs for this crawler.

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
August 19, 2024 at 18:22:56	-	05 s	Running	-	-

- The crawler will scan the data in the [data-warehouse](#) folder and create corresponding tables in the [spotify_data](#) database.

AWS Services Search [Option+S]

AWS Glue > Tables > datawarehouse

datawarehouse

Last updated (UTC) August 19, 2024 at 18:55:35 | [Version 0 \(Current version\)](#) | [Actions](#)

Table details		
Name datawarehouse	Classification Parquet	Deprecated -
Database spotify_data	Location s3://spotify-aws-prjct/datawarehouse/	Column statistics No statistics
Description -	Connection -	
Last updated August 19, 2024 at 18:55:03		
Advanced properties		

[Schema](#) | [Partitions](#) | [Indexes](#) | [Column statistics - new](#)

Schema (39)
View and manage the table schema.

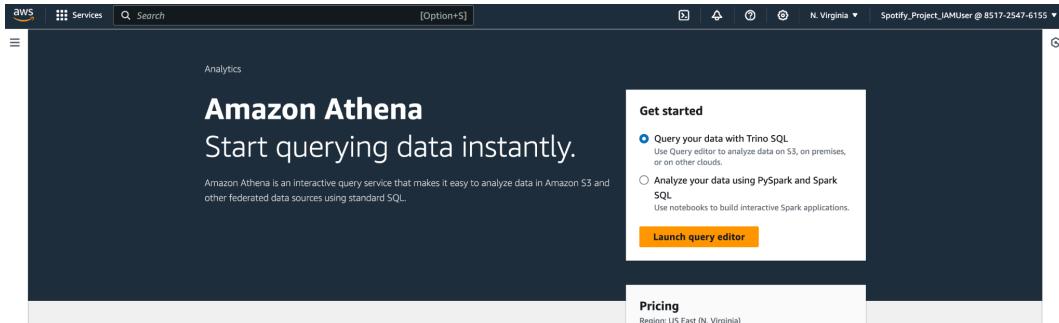
#	Column name	Data type	Partition key	Comment
1	followers	string	-	-
2	artist_10	string	-	-
3	artist_1	string	-	-
4	track_id	string	-	-

Step 5: Querying Data with AWS Athena

1. Set Up Query Result Storage:

- Navigate to AWS Athena from the AWS Management Console.

- Click on ‘Launch query editor’



- Before running any queries, you must specify a location for query results.
- Create a new S3 bucket named **spotify-proj-athena-output**.

Name	AWS Region	IAM Access Analyzer	Creation date
aws-glue-assets-851725476155-us-east-2	US East (Ohio) us-east-2	View analyzer for us-east-2	August 17, 2024, 23:21:02 (UTC-05:00)
myglobals3	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 2, 2024, 18:21:28 (UTC-05:00)
spotify-aws-prjct	US East (Ohio) us-east-2	View analyzer for us-east-2	August 17, 2024, 21:44:42 (UTC-05:00)
spotify-proj-athena-output	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 19, 2024, 14:03:36 (UTC-05:00)

- In the Athena settings, set the query result location to this new bucket.

Query result location and encryption

Location of query result - optional
Enter an S3 prefix in the current region where the query result will be saved as an object.

[View](#) [Browse S3](#)

You can create and manage lifecycle rules for this bucket
Use Amazon S3 lifecycle rules to store your query results and metadata cost effectively or to delete them after a period of time.
[Learn more](#)

Expected bucket owner - optional
Specify the AWS account ID that you expect to be the owner of your query results output location bucket.

Assign bucket owner full control over query results
Enabling this option grants the owner of the S3 query results bucket full control over the query results. This means that if your query result location is owned by another account, you grant full control over your query results to the other account.

Encrypt query results

[Cancel](#) [Save](#)

2. Write SQL Queries:

- In the Athena query editor, write SQL queries to analyze the data.

Example Query:

```
SELECT * FROM datawarehouse LIMIT 10;
```



Athena now supports typeahead code suggestions to speed up SQL query development
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.

Data

Data source: AwsDataCatalog
Database: spotify_data

Tables and views

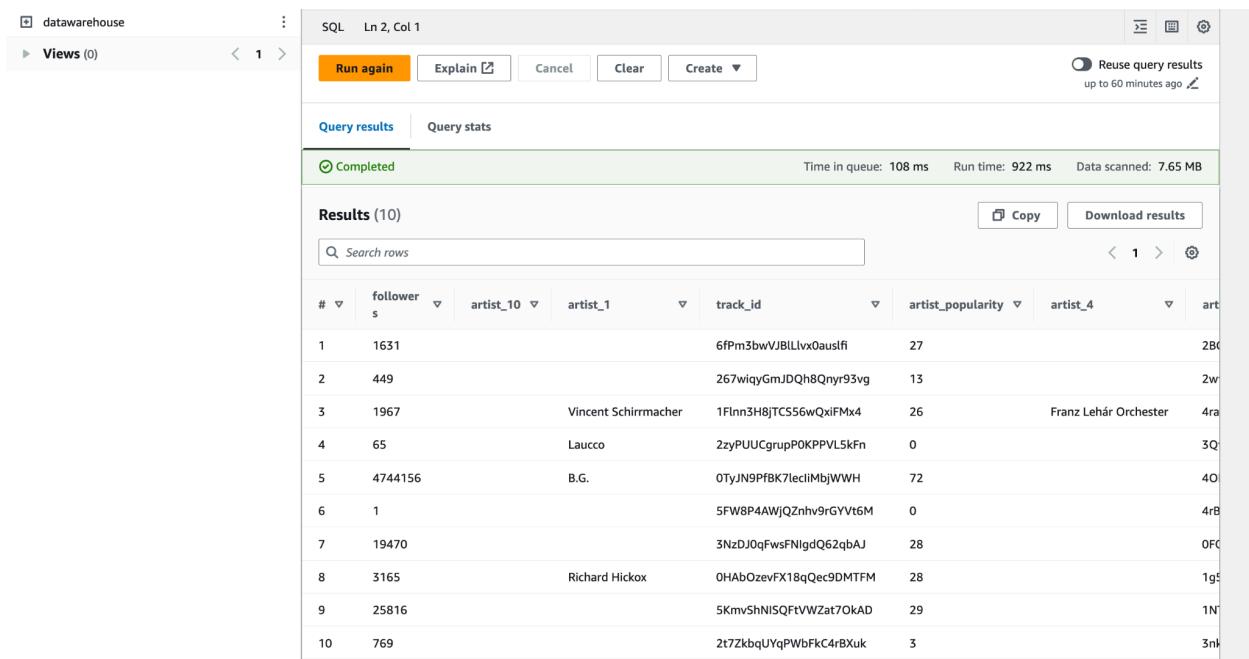
Tables (1)

```
Query 1 :  
1 SELECT * FROM datawarehouse LIMIT 10;  
2
```

- This query will fetch the first 10 records from the `data_warehouse` table.

3. Run the Query:

- Click "Run query" to execute the SQL statement.



SQL Ln 2, Col 1

Run again Explain Cancel Clear Create

Reuse query results up to 60 minutes ago

Views (0) < 1 >

Completed Time in queue: 108 ms Run time: 922 ms Data scanned: 7.65 MB

Results (10)

#	follower_s	artist_10	artist_1	track_id	artist_popularity	artist_4	artist_5
1	1631			6fPm3bwVJBLlxvOauslfI	27		2B0
2	449			267wiqyGmJDQh8Qnyrr93vg	13		2w
3	1967	Vincent Schirrmacher	1fInn3H8jTC556wQxiFMx4	26	Franz Lehár Orchester	4ra	
4	65	Laucco	2zyPUUCgrupP0KPPVLSkFn	0		3Q	
5	4744156	B.G.	0TyJN9PfBK7lecllMbjWWH	72		40	
6	1		5FW8P4AWjQ2hrhv9rGVVt6M	0		4rB	
7	19470		3NzDJ0qFwsFNlgdQ62qbAJ	28		0FC	
8	3165	Richard Hickox	0HAb0zevFX18qQec9DMTFM	28		1g5	
9	25816		5KmvShNISQFtVVZat7OkAD	29		1N	
10	769		2t7ZkbqUYqPWbFkC4rBXuk	3		3nk	

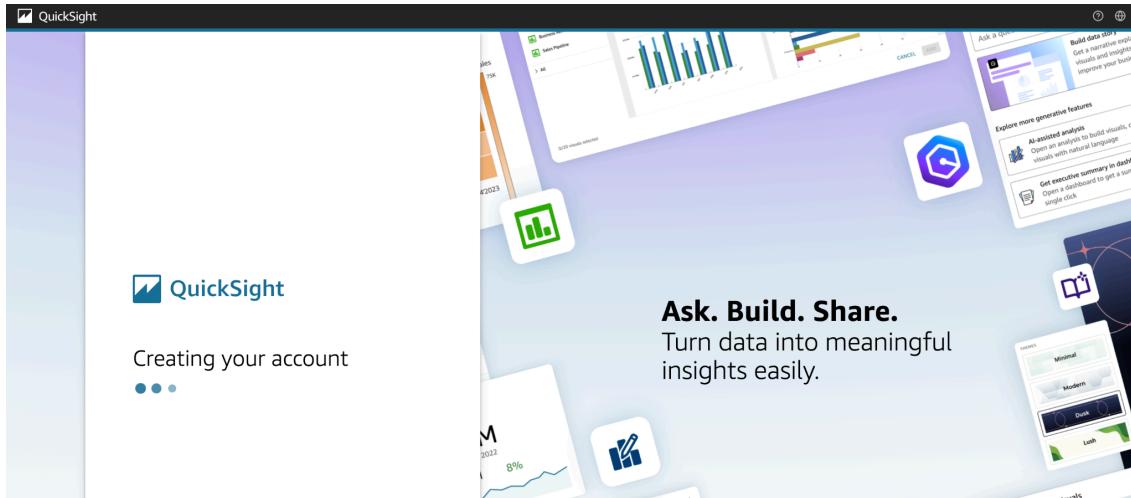
- The results will be displayed in the query editor, and the output will be saved in the specified S3 bucket (**athena-output**).

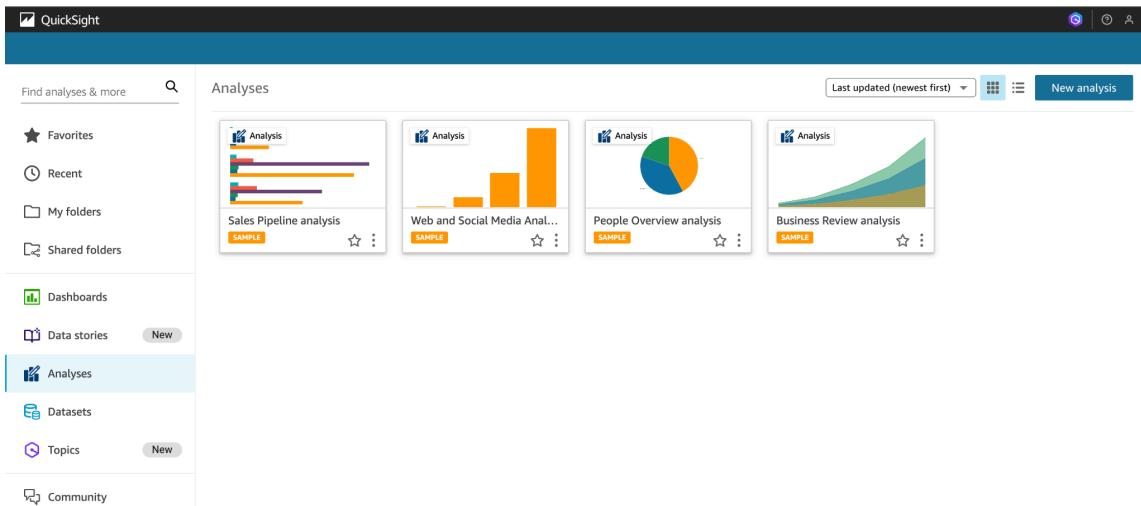
Name	Type	Last modified	Size	Storage class
04291a9f-b8c6-4849-86b9-f63cd4e37bdd.csv	csv	August 19, 2024, 14:09:14 (UTC-05:00)	421.0 B	Standard
04291a9f-b8c6-4849-86b9-f63cd4e37bdd.csv.metadata	metadata	August 19, 2024, 14:09:14 (UTC-05:00)	119.0 B	Standard
682cb8b0-4f47-4c36-899d-57ea5c55eb76.csv	csv	August 19, 2024, 14:07:46 (UTC-05:00)	4.6 KB	Standard
682cb8b0-4f47-4c36-899d-57ea5c55eb76.csv.metadata	metadata	August 19, 2024, 14:07:47 (UTC-05:00)	2.0 KB	Standard

Step 6: Visualizing Data with AWS QuickSight

1. Sign Up for AWS QuickSight:

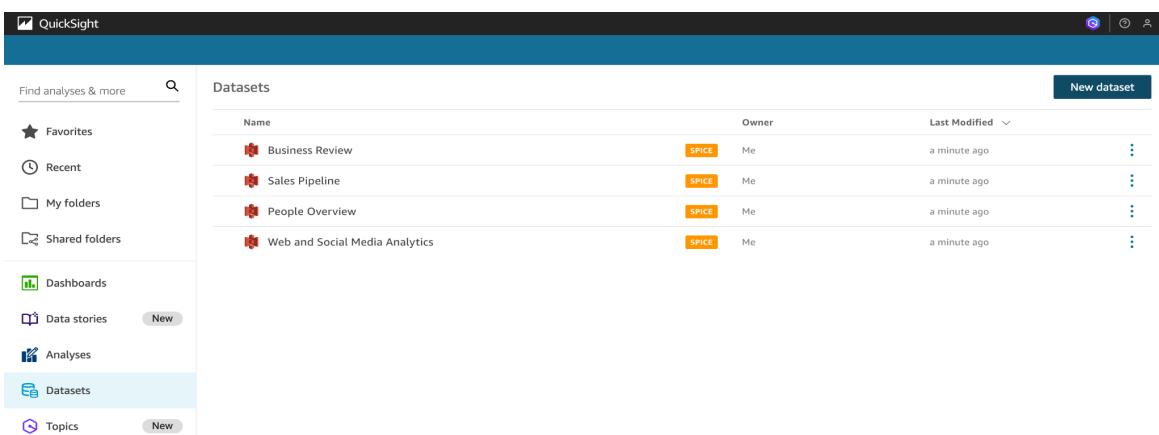
- Login to your root account and In the AWS Management Console, search for “QuickSight” and select the service.
- If you haven’t signed up for QuickSight, you’ll need to do so. Select the Enterprise Edition (free for 30 days).



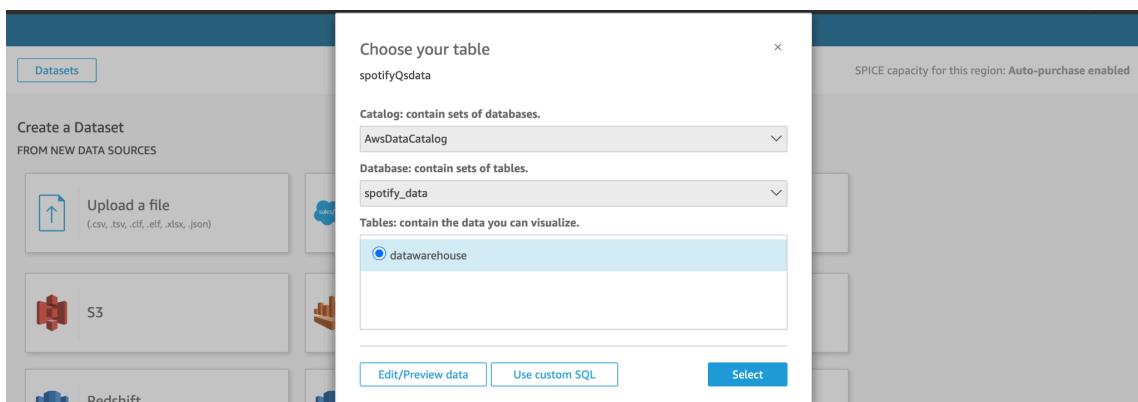


2. Connect QuickSight to Athena:

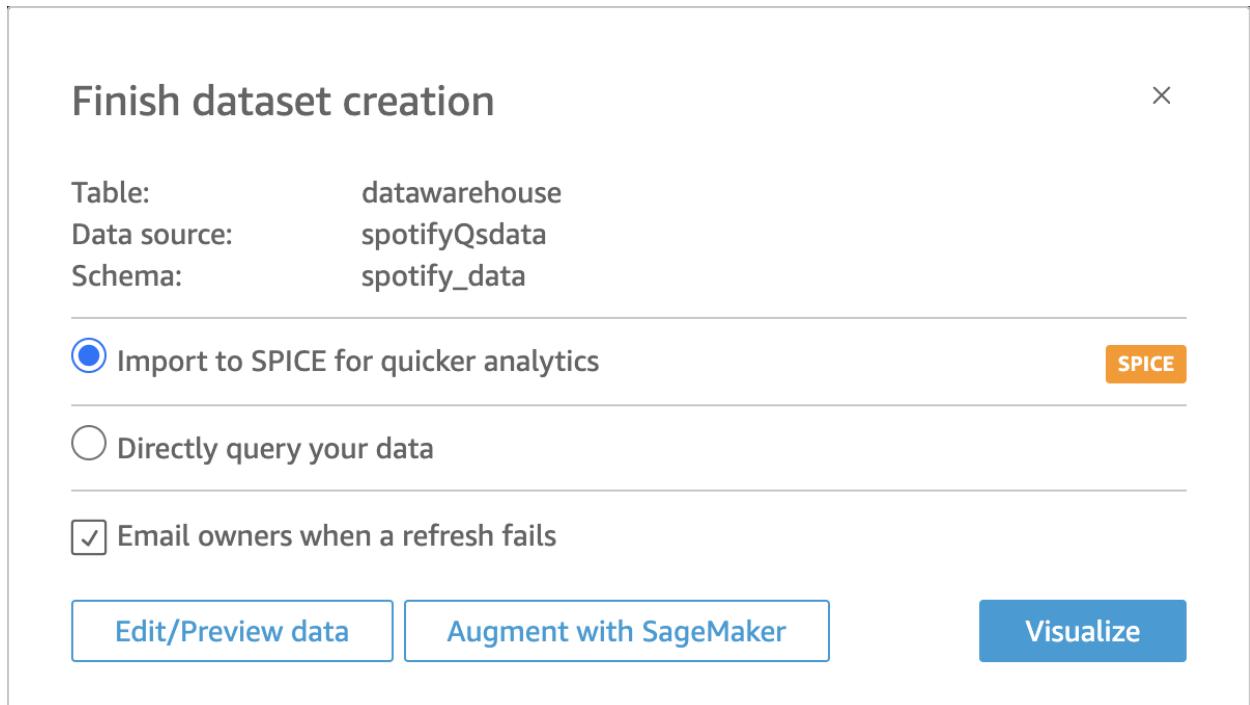
- Once signed in, go to "Datasets" and click "New dataset."



- Select "Athena" as the data source and name it with SpotifyQsData
- Choose the `spotify_data` database and the `data_warehouse` table.



- Click on Visualize.



3. Create Visualizations:

- After importing the data, you can create various types of visualizations (e.g., bar charts, line charts, pie charts) using the fields from the `data_warehouse` table.

QSight | datawarehouse analysis

File Edit Data Insert Sheets Objects Search

Dataset: SPICE datawarehouse

Visuals

Sheet 1

AutoGraph
Add 1 or more fields to build a visual.

- Example: Create a bar chart to visualize the popularity of tracks by artist.
4. **Publish and Share Dashboards:**
- Once your visualizations are ready, you can publish the dashboard and share it with stakeholders.
 - Click "Publish dashboard" and follow the prompts to share it via email or a link.
-

Conclusion

This document serves as a runbook for the Spotify Data Engineering Project. Follow each step to set up and run your end-to-end data pipeline.

Real-World Applications

The insights gained from this project can be applied in various real-world scenarios:

1. **Music Recommendations:** By analyzing the popularity of tracks and artists, platforms can improve their recommendation engines.
2. **Market Analysis:** Record labels and music producers can use the data to understand trends and make informed decisions on which genres or artists to promote.
3. **User Engagement:** Streaming services can analyze user preferences and behavior to enhance user engagement through personalized playlists and features.
4. **Business Intelligence:** The visualizations and insights derived can help business analysts make data-driven decisions to optimize marketing strategies and improve user retention.