

End-to-End Analytics with AWS

**Building Scalable and Versatile Data
Solutions for Any Dataset**



Architecting Intelligent Data Pipelines on AWS for Scalable Analytic

Srujankumar Polepally

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 demonstrates the creation of an end-to-end data engineering pipeline using Amazon Web Services (AWS). The pipeline handles data ingestion, processing, storage, and visualization of a large dataset - specifically, Spotify data. By leveraging AWS Glue, S3, Athena, and QuickSight, the solution showcases a scalable, cost-effective, and efficient approach for transforming raw data into meaningful insights. The implementation is applicable across industries that rely on data-driven decision-making, particularly for analytics in the music industry. The project is based on tutorials from Data with Data.

Introduction:

The goal of this project is to build a comprehensive data pipeline on AWS that processes and analyzes Spotify datasets. Using a combination of AWS services such as S3, Glue, Athena, and QuickSight, we develop a workflow that transforms raw data into actionable insights.

Project Architecture Overview:

1. Staging Layer: Raw data is stored in an Amazon S3 bucket.
2. ETL Pipeline: AWS Glue extracts, transforms, and loads data into the warehouse.
3. Data Warehouse: Processed data is stored in a separate S3 bucket.
4. Data Catalog: AWS Glue Crawler catalogs the data into structured tables.
5. Data Analysis: AWS Athena queries the curated datasets.
6. Visualization: AWS QuickSight generates analytical dashboards.

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:

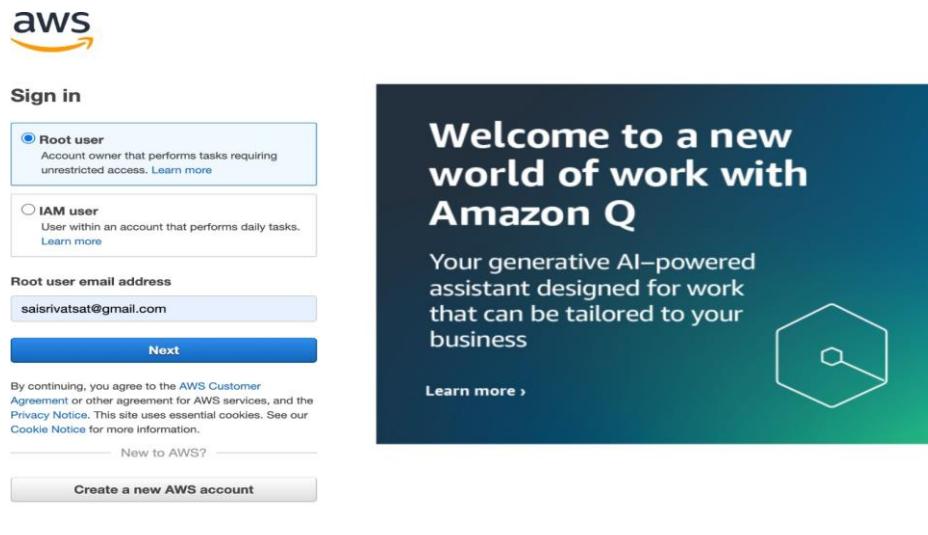
- Albums: Includes album ID, name, popularity, and release date.
- Artists: Contains artist details such as name, followers, and genres.
- Tracks: Lists track ID, popularity, and audio features like energy and danceability.
- Spotify Features: Contains metrics such as loudness, speechiness, and valence.

Hands-On Steps

Step 1: Setting Up AWS IAM User

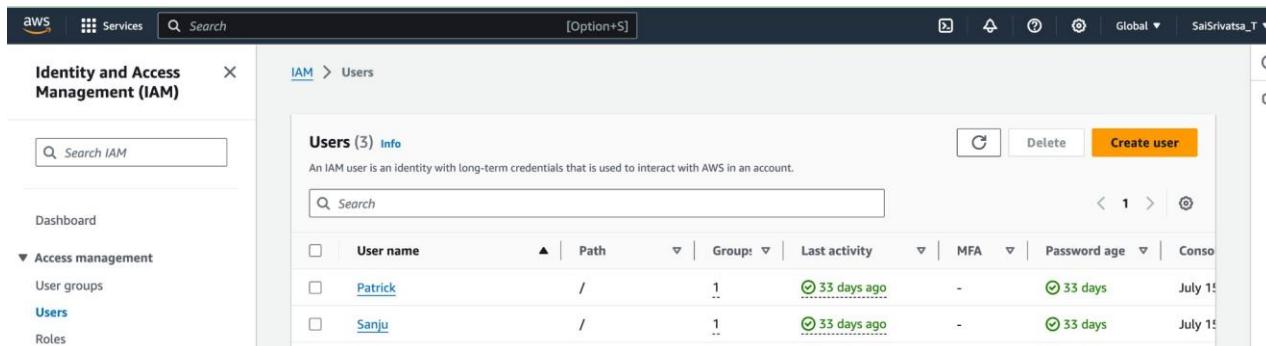
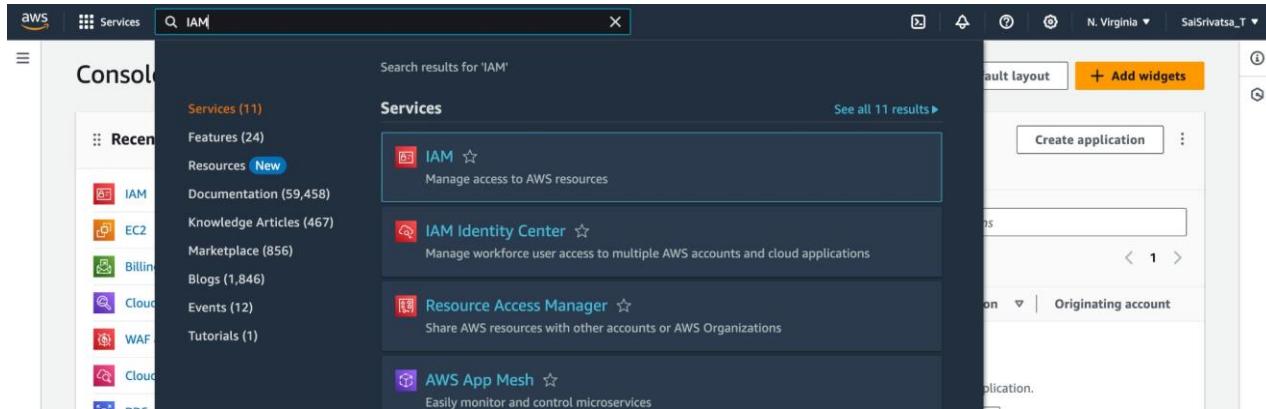
1. Log in to AWS Management Console:

- Access AWS using root credentials.



2. Create a New IAM User:

- Navigate to *IAM* → *Users* → *Create user*
- Set username: `project_user`
- Enable console access and assign a custom password.



● 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."

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.

Console Home

Recently visited

No recently visited services

Explore one of these commonly visited AWS services.

EC2 S3 RDS Lambda

View all services

Applications (0)

Region: US East (Ohio)

Create application

us-east-2 (Current Region) Find applications

Name Description Region Originating account

Access denied

Welcome to AWS

Getting started with AWS

AWS Health

Cost and usage

Current month costs Cost breakdown

Access denied Access denied

Forecasted month end costs

Access denied

Go to myApplications

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 interface. The search bar at the top contains the text "s3". Below the search bar, there is a sidebar with "Recent" and "Services (8)" sections. The main area displays search results for "s3", showing four items under the "Services" category: "S3" (Scalable Storage in the Cloud), "S3 Glacier" (Archive Storage in the Cloud), "AWS Snow Family" (Large Scale Data Transport), and "Storage Gateway" (Hybrid Storage Integration). Each service entry includes a small icon and a star rating.

2. Create a New S3 Bucket:

- Click on "Create bucket."

The screenshot shows the Amazon S3 service page. The left sidebar has sections 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 (Dashboards, Storage Lens groups, AWS Organizations settings), and a Feature spotlight. The main area has a header "Amazon S3" and a search bar. It features an "Account snapshot - updated every 24 hours" section with a "View Storage Lens dashboard" button. Below this is a table titled "General purpose buckets (1)". The table has columns for Name, AWS Region, IAM Access Analyzer, and Creation date. It lists one bucket: "myglobals3" (Name), "US East (N. Virginia) us-east-1" (AWS Region), "View analyzer for us-east-1" (IAM Access Analyzer), and "August 2, 2024, 18:21:28 (UTC-05:00)" (Creation date). At the top right of the bucket list, there are buttons for "Create bucket", "Copy ARN", "Empty", and "Delete".

- **Bucket Name:** Enter {Global|Unique|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) Info All AWS Regions

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](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

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

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.				

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

Folder

Folder name
Staging /

Folder names can't contain "/". [See rules for naming](#)

Server-side encryption [Info](#)
Server-side encryption protects data at rest.

The following encryption settings apply only to the folder object and not to sub-folder objects.

Server-side encryption

Don't specify an encryption key
The bucket settings for default encryption are used to encrypt the folder object when storing it in Amazon S3.

Specify an encryption key
The specified encryption key is used to encrypt the folder object before storing it in Amazon S3.

If your bucket policy requires objects to be encrypted with a specific encryption key, you must specify the same encryption key when you create a folder. Otherwise, folder creation will fail.

Create folder

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

Amazon S3 > Buckets > spotify-aws-prjct

spotify-aws-prjct [Info](#)

Actions	Objects	Properties	Permissions	Metrics	Management	Access Points
Actions Info	<input type="button" value="C"/> <input type="button" value="Copy S3 URI"/> <input type="button" value="Copy URL"/> <input type="button" value="Download"/> <input type="button" value="Open"/> <input type="button" value="Delete"/> <input type="button" value="Actions"/> <input type="button" value="Create folder"/> <input type="button" value="Upload"/>	Objects (2) Info				
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						
<input type="text" value="Find objects by prefix"/> <input type="button" value="<"/> <input type="button" value="1"/> <input type="button" value=">"/> <input type="button" value="Reset"/>						
	<input type="checkbox"/> Name <input type="button" value="▲"/> Type <input type="button" value="▼"/> Last modified <input type="button" value="▼"/> Size <input type="button" value="▼"/> Storage class <input type="button" value="▼"/>					
	<input type="checkbox"/> <input type="checkbox"/> datawarehouse/ <input type="checkbox"/> Folder <input type="checkbox"/> <input type="checkbox"/> -					
	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> staging/ <input type="checkbox"/> Folder <input type="checkbox"/> <input type="checkbox"/> -					

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. At the top, it says 'Upload info'. Below that, there's a large text area with placeholder text: 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Underneath, a table lists the selected files:

Name	Folder	Type
spotify_artist_data_2023.csv	-	text/csv
spotify_tracks_data_2023.csv	-	text/csv
spotify_albums_data_2023.csv	-	text/csv

Below the table, the 'Destination info' section shows the destination as 's3://spotify-aws-prj/staging/'. There are sections for 'Permissions' and 'Properties'. 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' screen. At the top, a green banner says 'Upload succeeded'. Below it, a summary table shows:

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

Below the summary is a table titled 'Files and folders' showing the details of the uploaded files:

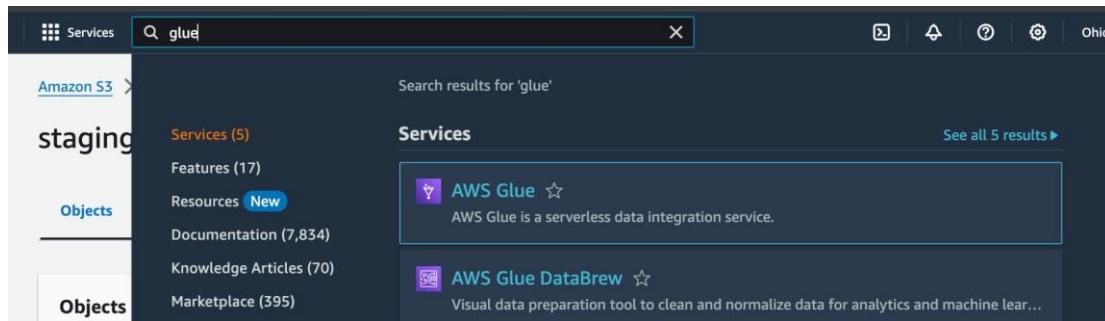
Name	Folder	Type	Size	Status	Error
spotify_artis...	-	text/csv	2.4 MB	Succeeded	-
spotify_trac...	-	text/csv	13.1 MB	Succeeded	-
spotify-albu...	-	text/csv	94.6 MB	Succeeded	-

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.

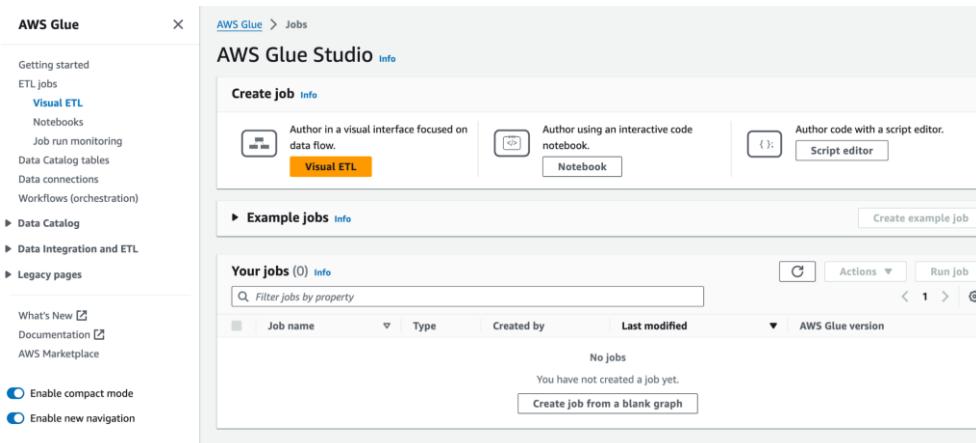
1. Navigate to AWS Glue Service:

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



2. Create a New Glue Job:

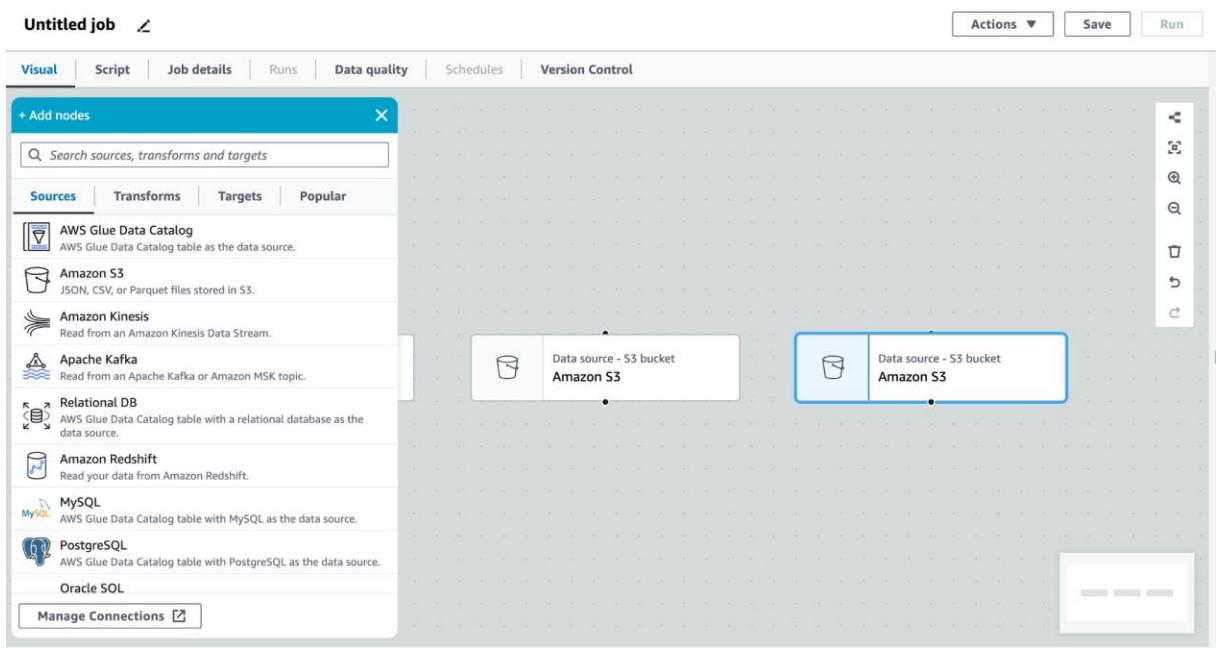
- In the AWS Glue dashboard, click on "Visual ETL" under the "ETL Jobs" section.



- 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`)



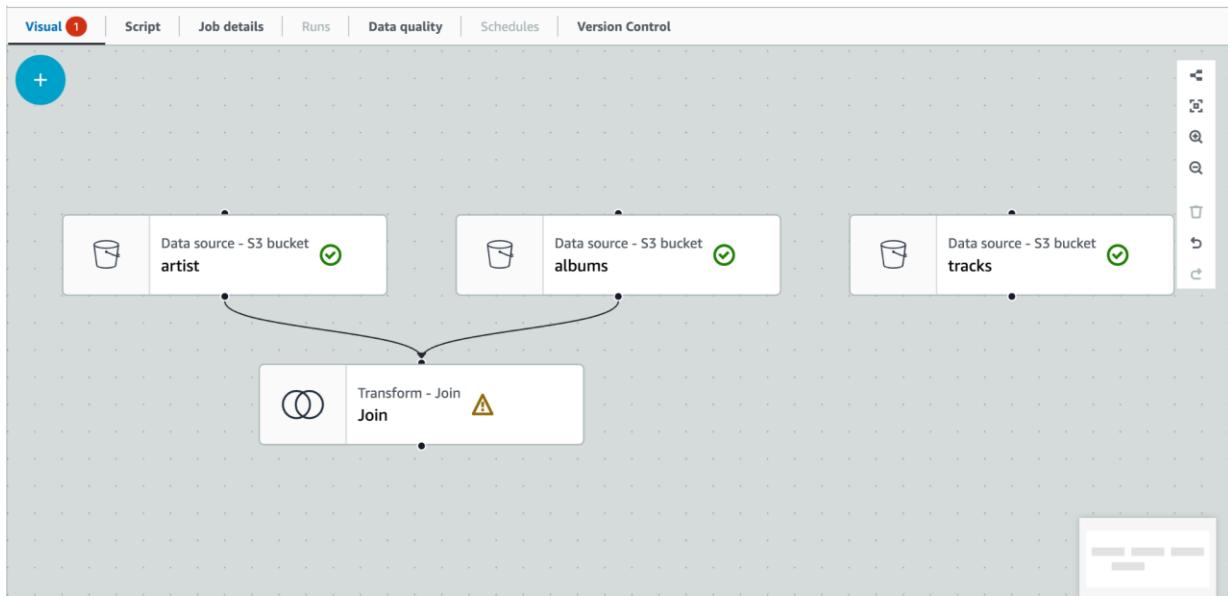
- Select the First Amazon S3 bucket and rename it with “artist” and click on browse and select the file from the **s3://spotify-aws-prjct/staging/spotify_artist_data_2023.csv**. Select the Data format as CSV.

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

The screenshot shows the AWS Glue Data Transformation interface for an "Untitled job". On the left, there's a visual editor with three data source nodes: "Data source - S3 bucket artist", "Data source - S3 bucket albums", and "Data source - S3 bucket tracks". On the right, a "Data source properties - S3" panel is open for the "tracks" bucket, showing settings like "Name: tracks", "S3 source type: S3 location", "S3 URL: s3://spotify-aws-prjct/staging/spot", and "Data format: CSV". Below the visual editor, there are tabs for "Data preview" and "Output schema", and sections for "Start a data preview session" and "IAM role".

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**”.

Untitled job

Actions ▾ **Save** **Run**

Visual | Script | Job details | Runs | **Data quality** | Schedules | Version Control

Transform

Name: Album and Artist Join

Node parents: Choose one or more parent node. artist S3 - DataSource, albums S3 - DataSource

Join type: Inner join. Select all rows from both datasets that meet the join condition.

Join conditions: Select a field from each parent node for the join condition. artist id = albums artist_id

Add condition

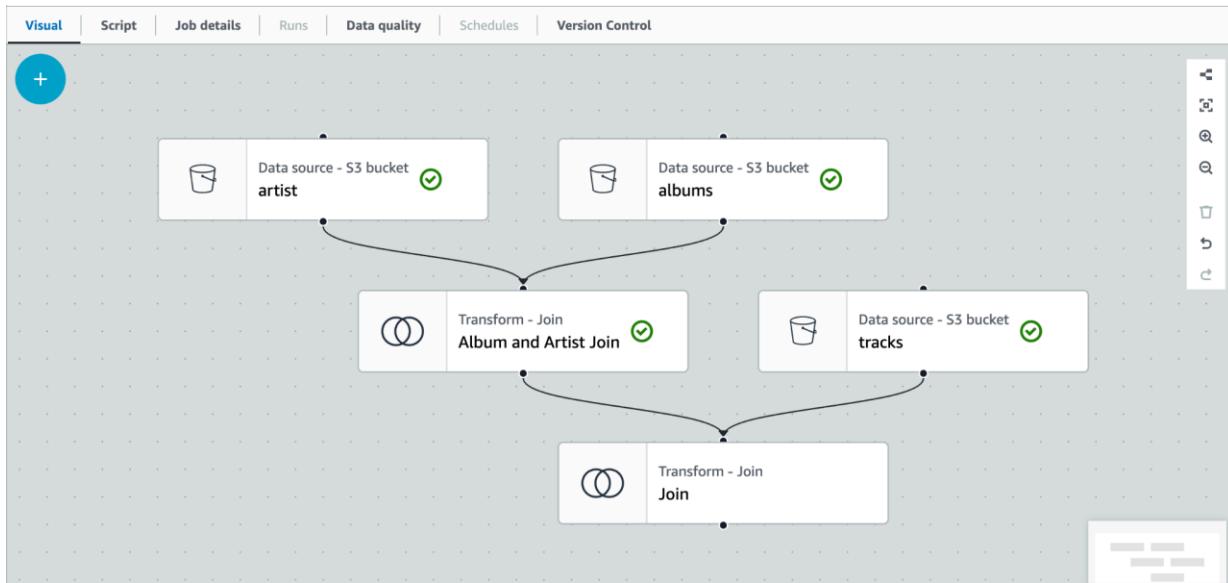
Data preview | **Output schema**

Start a data preview session

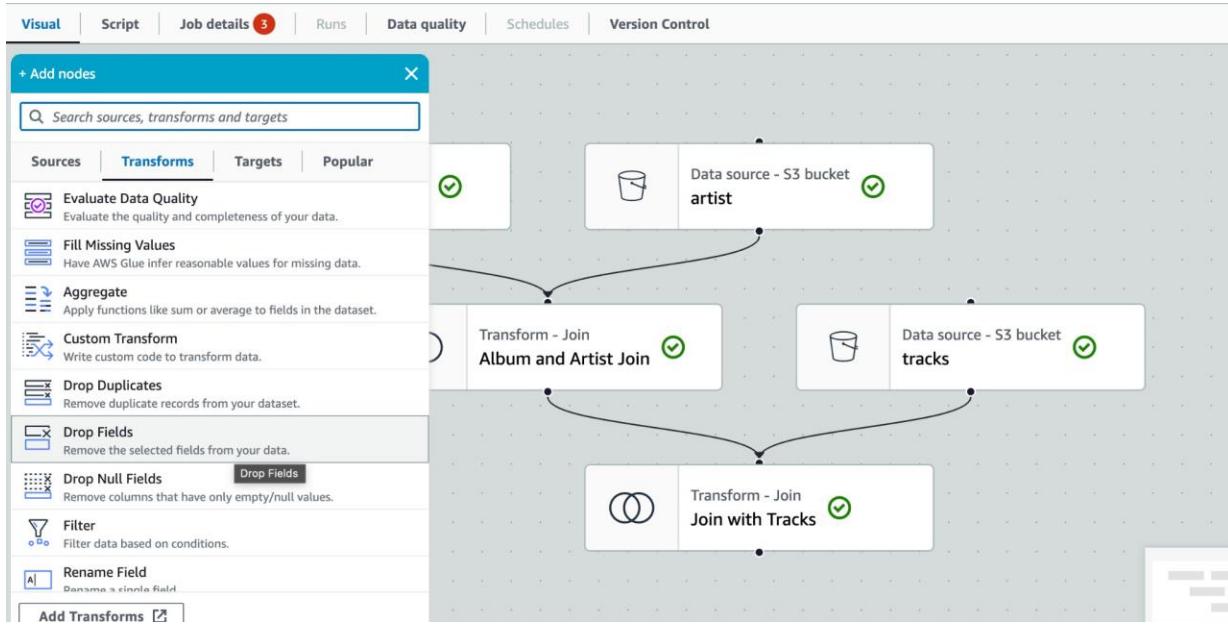
IAM role: To start a data preview session, choose an IAM role for this job. Changing the role will end an existing data preview session. Create IAM role.

Additional Settings

- 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.

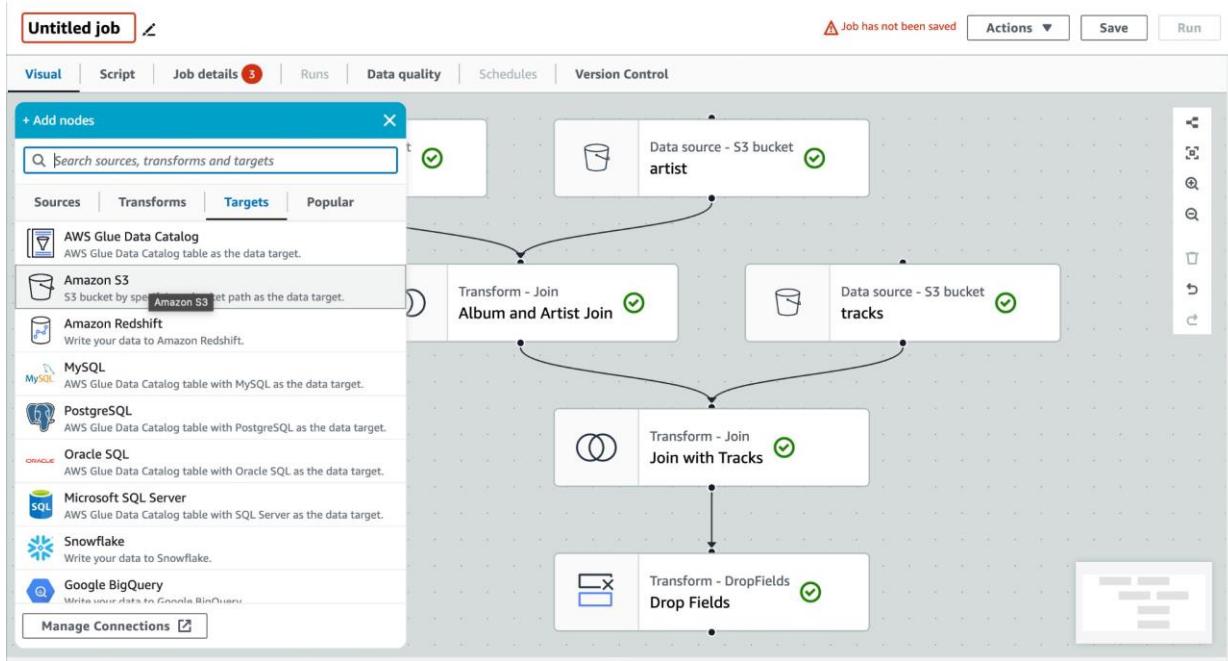


- We remove duplicate columns and Identical columns we dont need. Here id is inner join to artist id so select id.

Field	Type
artist_10	string
artist_11	string
duration_sec	string
id	string
name	string
artist_popularity	string
artist_genres	string
followers	string
genre_0	string
genre_1	string
genre_2	string
genre_3	string
genre_4	string
genre_5	string
genre_6	string
.id	string
track_popularity	string
explicit	string

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

Step 2
Add permissions

Step 3
Name, review, and create

Service or use case

Glue

Choose a use case for the specified service.
Use case
 Glue
Allows Glue to call AWS services on your behalf.

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
AmazonDMSRedshiftS3Role	AWS managed	Provides access to manage S3 settings ...
AmazonS3FullAccess	AWS managed	Provides full access to all buckets via t...
AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	Provides AWS Lambda functions permis...
AmazonS3OutpostsFullAccess	AWS managed	Provides full access to Amazon S3 on ...
AmazonS3OutpostsReadOnlyAccess	AWS managed	Provides read only access to Amazon S...
AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all bucket...
AWSBackupServiceRolePolicyForS3Backup	AWS managed	Policy containing permissions necessar...
AWSBackupServiceRolePolicyForS3Restore	AWS managed	Policy containing permissions necessar...
QuickSightAccessForS3StorageManagementA...	AWS managed	Policy used by QuickSight team to acc...

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.
glue_s3_access

Description
Add a short explanation for this role.
Allows Glue to call AWS services on your behalf.

Step 1: Select trusted entities

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

AWS Glue > Monitoring

Monitoring

Date range: 7 Day

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

Filter displayed data: Filter data

Percentage: 100

Job type breakdown

Filter displayed job types: Filter data

Percentage: 1 (blue bar)

- Check S3 bucket if the files are Parsed

Amazon S3

Buckets

Objects (16)

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
run-				

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

Name	State	Schedule	Last run	Last run timestamp	Log	Table chang...
No resources						

Create crawler

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

Description - *optional*
Enter a description
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).

[Clear selection](#) [Add new connection](#)

Location of S3 data
 In this account
 In a different account

S3 path
 Browse for or enter an existing S3 path.
 s3://bucket/prefix/object [View](#) [Browse S3](#)
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
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)

Filter by Type

Policy name	Type	Description
AWSGlueServiceRole	AWS managed	Policy for AWS Glue service role which ...

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.

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.

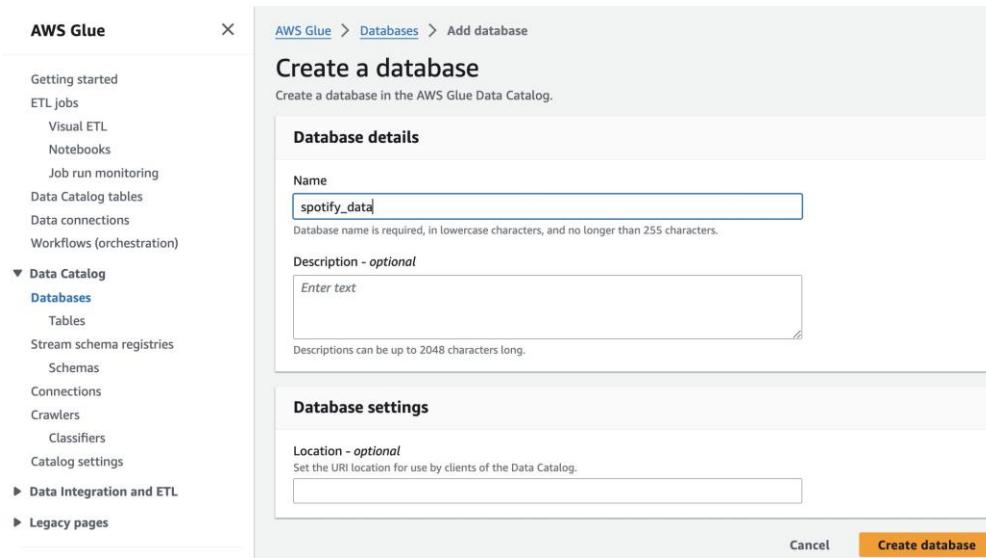
Security configuration - optional

Enable at-rest encryption with a security configuration.

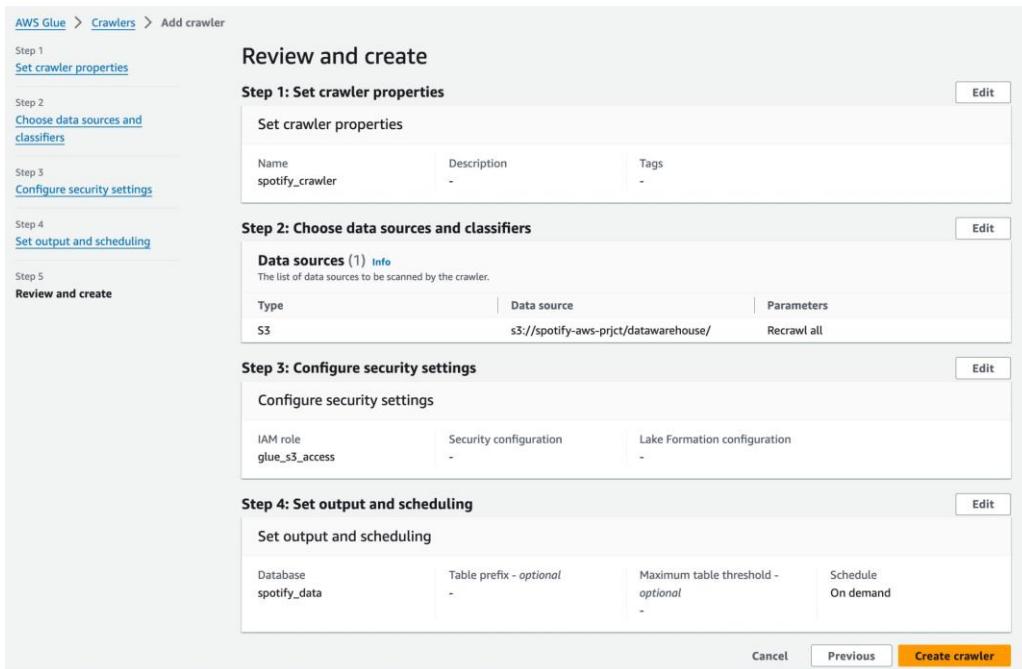
Cancel Previous Next

2. Create a New Database:

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



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



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	IAM role glue_s3_access	Database	State
spotify_crawler	glue_s3_access	spotify_data	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 Glue

Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring

Data Catalog tables

Data connections
Workflows (orchestration)

Tables

Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings

Data Integration and ETL

Legacy pages

What's New | Documentation | AWS Marketplace

Enable compact mode | Enable new navigation

datawarehouse

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

Table overview | Data quality New

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 (39)

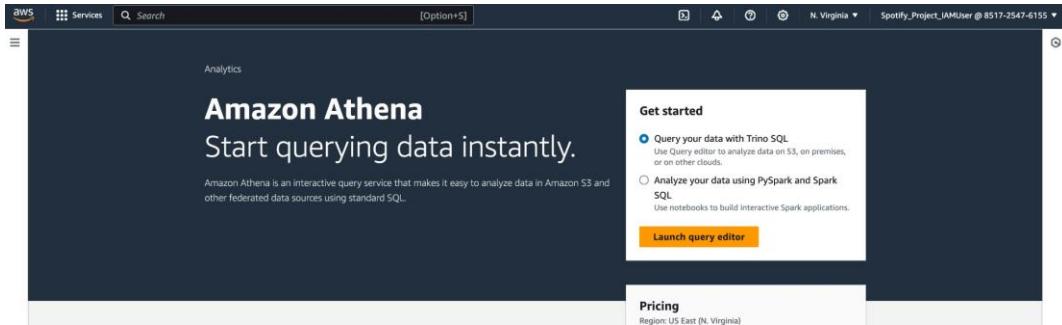
#	Column name	Data type	Partition key	Comment
1	followers	string	-	-
2	artist_10	string	-	-
3	artist_1	string	-	-
4	track_id	string	-	-

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.

2. Write SQL Queries:

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

Example Query:

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

The screenshot shows the Amazon Athena Query Editor interface. At the top, there are tabs for 'Editor', 'Recent queries', 'Saved queries', and 'Settings'. A 'Workgroup' dropdown is set to 'primary'. A message box at the top states: '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.' Below this, the 'Data' sidebar shows a 'Data source' set to 'AwsDataCatalog' and a 'Database' set to 'spotify_data'. Under 'Tables and views', there is a 'Tables (1)' section with a single entry. The main area is titled 'Query 1' and contains the SQL query: 'SELECT * FROM datawarehouse LIMIT 10;'. The number '2' is highlighted in the code.

- 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.

The screenshot shows the 'datawarehouse' database in the 'Views (0)' section. The 'SQL' tab is active, showing the query 'SELECT * FROM datawarehouse LIMIT 10;'. Below the query are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. To the right, a 'Reuse query results' button is shown, indicating results are available up to 60 minutes ago. The 'Query results' tab is selected, showing a table titled 'Completed' with 10 rows. The columns are: #, follower_s, artist_10, artist_1, track_id, artist_popularity, artist_4, and artist_5. The results table includes a 'Search rows' bar and buttons for 'Copy' and 'Download results'. The results are as follows:

#	follower_s	artist_10	artist_1	track_id	artist_popularity	artist_4	artist_5
1	1631			6fPm3bwVJBILlx0auslfI	27		2B6
2	449			267wiqyGmJDQh8Qnyr93vg	13		2w
3	1967	Vincent Schirrmacher		1fLn3H8jTCS56wQxiFMx4	26	Franz Lehár Orchester	4ra
4	65	Laucco		2zyPUUCgrupPOKPPVL5kFn	0		3Q
5	4744156	B.G.		0TyJN9PfBK7leclMbJWWH	72		4O
6	1			5FW8P4AWjQZnhv9rGVvt6M	0		4rB
7	19470			3NzDJ0qFwsFNlgdQ62qbAJ	28		0FC
8	3165	Richard Hickox		0HAbOzevFX18qQec9DMTFM	28		1g5
9	25816			5KmvShNISQFtVWZat7OkAD	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`).

Amazon S3 > Buckets > spotify-proj-athena-output > Unsaved/ > 2024/ > 08/ > 19/

19/

Copy S3 URI

Objects Properties

Objects (4) Info

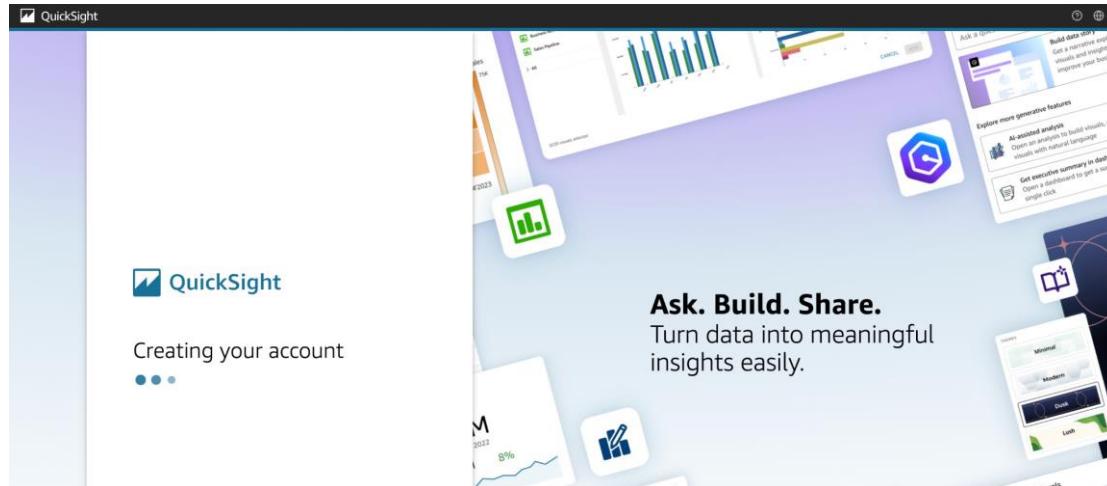
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Name	Type	Last modified	Size	Storage class
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).



The screenshot shows the QuickSight web interface. On the left, there is a sidebar with navigation links: Favorites, Recent, My folders, Shared folders, Dashboards, Data stories, Analyses (selected), Datasets (selected), Topics, and Community. The main area is titled 'Analyses' and displays four sample analyses: 'Sales Pipeline analysis', 'Web and Social Media Analy...', 'People Overview analysis', and 'Business Review analysis'. Each analysis card includes a 'SAMPLE' button and three-dot options menu.

2. Connect QuickSight to Athena:

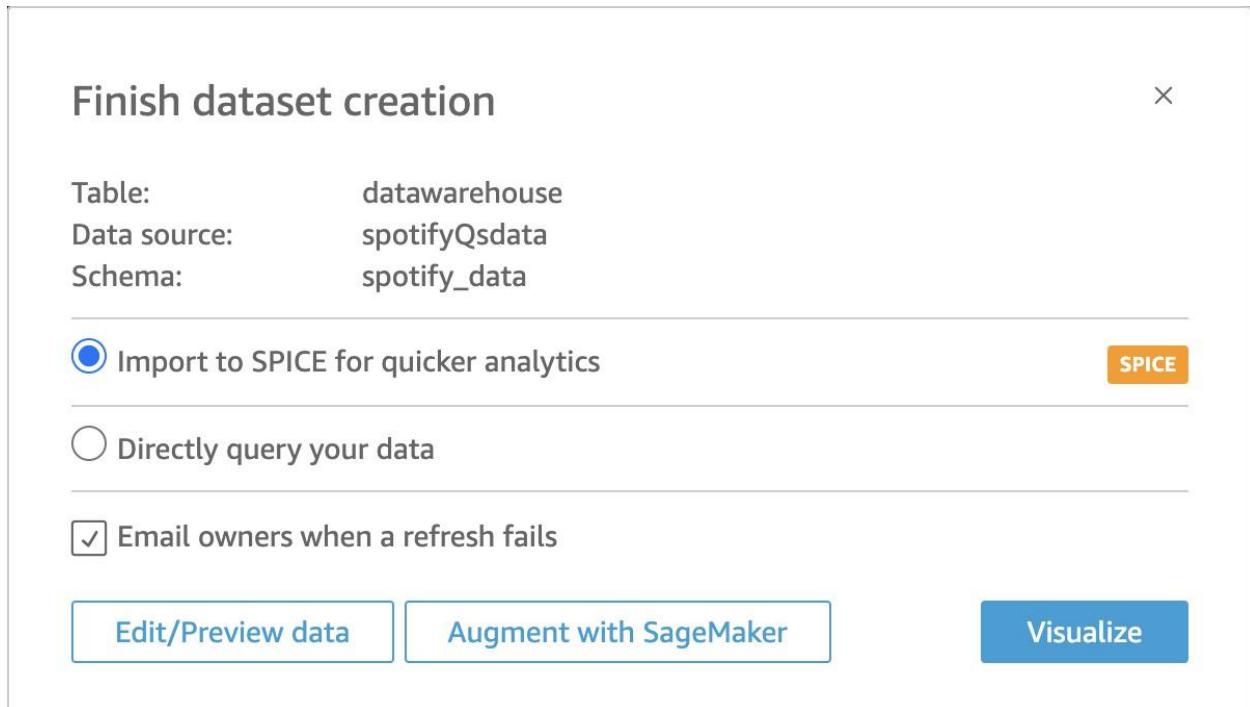
- Once signed in, go to "Datasets" and click "New dataset."

The screenshot shows the QuickSight web interface with the 'Datasets' section selected in the sidebar. The main area displays a list of existing datasets: 'Business Review', 'Sales Pipeline', 'People Overview', and 'Web and Social Media Analytics'. Each dataset has a 'SPICE' icon, an owner column (Me), and a 'Last Modified' column (a minute ago). A 'New dataset' button is located at the top right of the dataset list.

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

The screenshot shows the 'Create a Dataset' dialog in QuickSight. On the left, there's a sidebar with 'Datasets' selected. The main area has a 'Create a Dataset' section with 'FROM NEW DATA SOURCES' and a 'Create' button. Below it are three data source options: 'Upload a file (.csv, .tsv, .clif, .elf, .xlsx, .json)', 'S3', and 'Redshift'. A central modal window titled 'Choose your table' is open, showing the 'spotifyQsdata' catalog. Under 'Catalog: contain sets of databases.', 'AwsDataCatalog' is selected. Under 'Database: contain sets of tables.', 'spotify_data' is selected. Under 'Tables: contain the data you can visualize.', 'datawarehouse' is selected. At the bottom of the modal are 'Edit/Preview data', 'Use custom SQL', and a 'Select' button.

- 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.

QuickSight | datawarehouse analysis

File Edit Data Insert Sheets Objects Search

ACTUAL SIZE PUBLISH NEW LOOK

Data

Dataset: SPICE datawarehouse

Search fields

+ CALCULATED FIELD

album_id
album_name
album_popularity
album_type
artist_0
artist_1
artist_10
artist_11
artist_12
artist_3
artist_4
artist_5
artist_6
artist_7
artist_8
artist_9
artist_genres
artist_id
artist_popularity
artists
duration_ms

Visuals

+ ADD CHANGE VISUAL TYPE

Sheet 1 +

AutoGraph
Add 1 or more fields to build a visual.

ADD DATA

Add a dimension or measure

- 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.