Spotify End-To-End Data Pipeline Project

Business Use Case

A music enthusiast wants to create a **database of top global songs**, which will be automatically updated on a **daily or weekly** schedule by extracting data from the **Spotify API**. The aim is to track popular songs, artists, and playlists over time. To achieve this, a data pipeline is built to automate the process of **extracting**, **transforming**, and **visualizing** the music data using AWS services and Python code.

Introduction

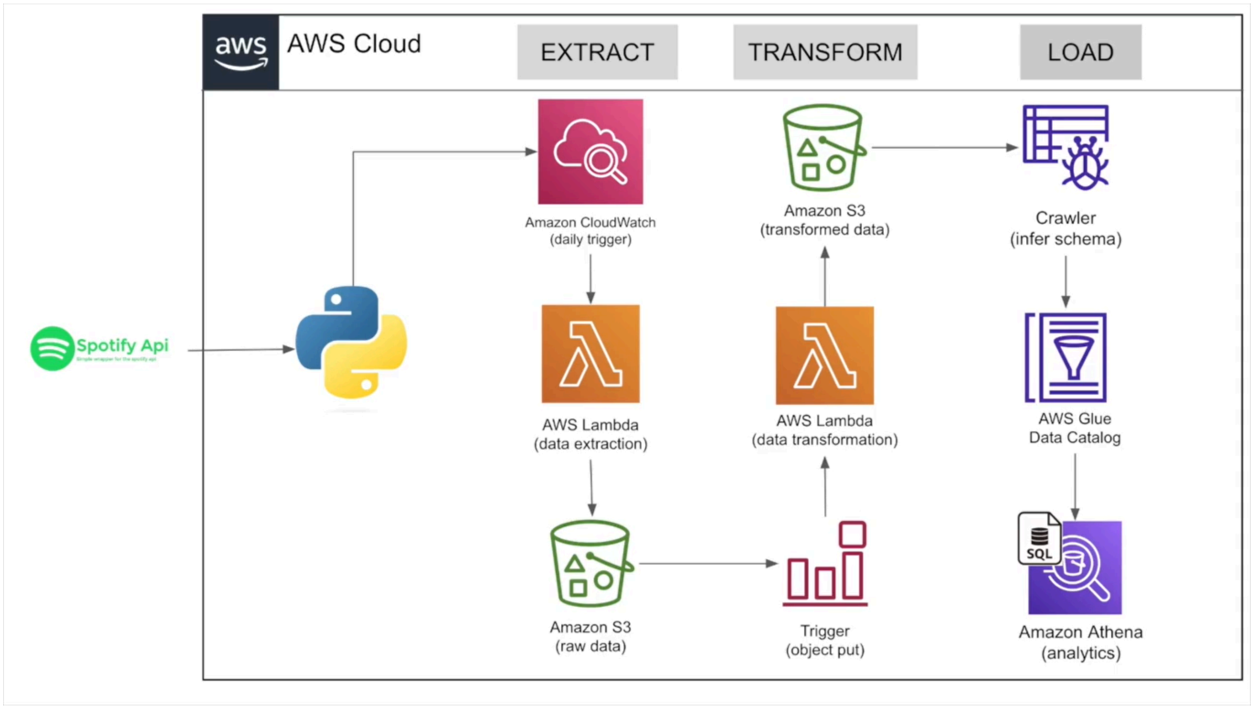
The **Spotify End-to-End Data Pipeline** project demonstrates the complete process of extracting, transforming, and loading (ETL) Spotify music data. The primary goal of this project is to create an automated pipeline that weekly collects music data, processes it, and stores it in a database for further analysis.

💡 Key AWS Services & Technologies:

- **Amazon CloudWatch**: Triggers the pipeline on a scheduled basis (daily or weekly).
- **AWS Lambda**: Runs Python code for data extraction and transformation.
- **Amazon S3**: Stores both raw and transformed data.
- **AWS Glue Crawler**: Automatically infers schema and catalogs the data for querying.

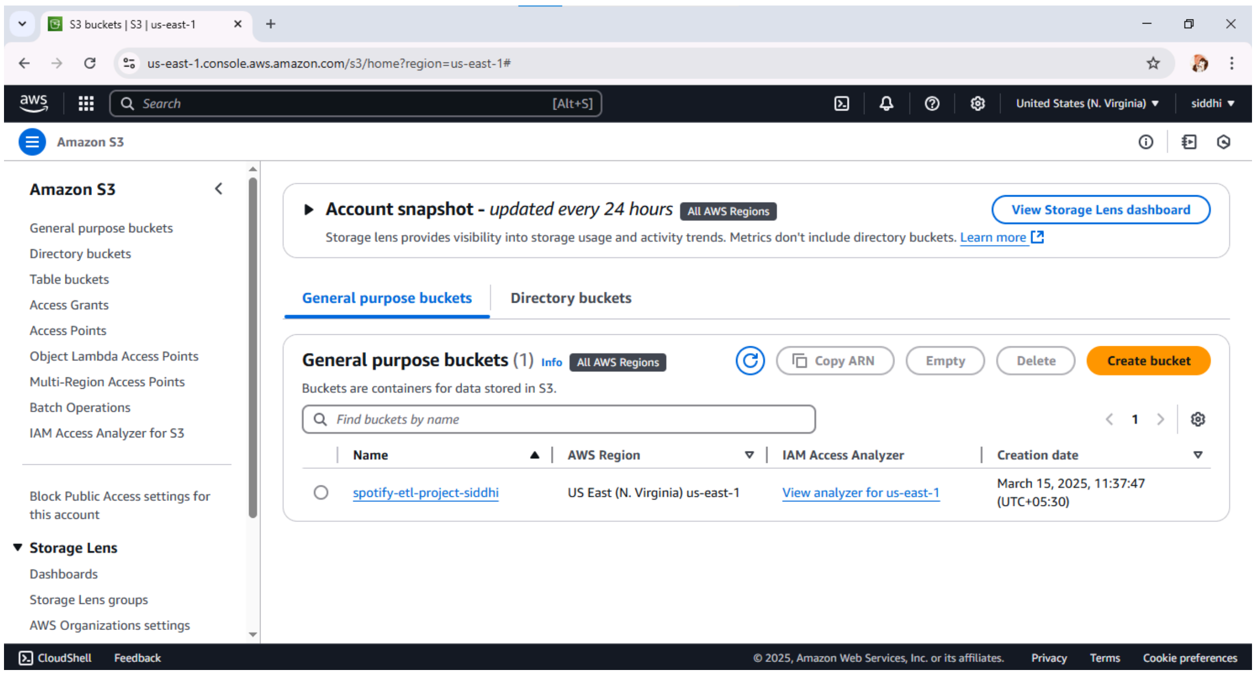
- **Amazon Athena**: Executes interactive SQL queries directly on data in Amazon S3.
- **Spotify API**: Source of music data (playlists, tracks, artists).

ARCHITECTURE DIAGRAM :


AWS Cloud
EXTRACT
TRANSFORM
LOAD
Spotify Api
Amazon CloudWatch (daily trigger)
AWS Lambda (data extraction)
Amazon S3 (raw data)
Amazon S3 (transformed data)
AWS Lambda (data transformation)
Trigger (object put)
Crawler (infer schema)
AWS Glue Data Catalog
Amazon Athena (analytics)
SQL

🛠️DEMONSTRATION OF THE PROJECT AND STEPS TO BUILD ETL PIPELINE :

1. Navigate to the **S3 service** in the AWS Console and create a bucket. I created a bucket named **spotify-etl-project-siddhi.**


S3 buckets | S3 | us-east-1

us-east-1.console.aws.amazon.com/s3/home?region=us-east-1#

Amazon S3

Account snapshot - *updated every 24 hours* All AWS Regions

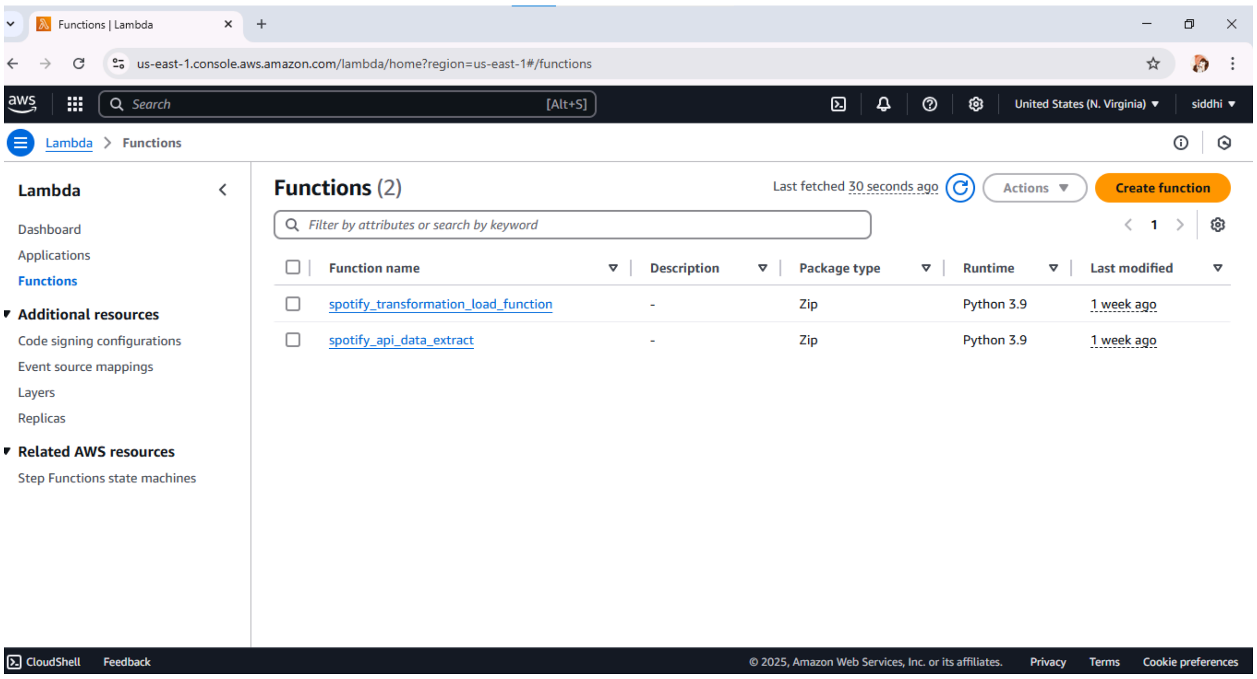
Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
spotify-etl-project-siddhi	US East (N. Virginia) us-east-1	View analyzer for us-east-1	March 15, 2025, 11:37:47 (UTC+05:30)

2. Then, I created two Lambda functions named **spotify_api_data_extract** and **spotify_transformation_load_function**.


Functions | Lambda

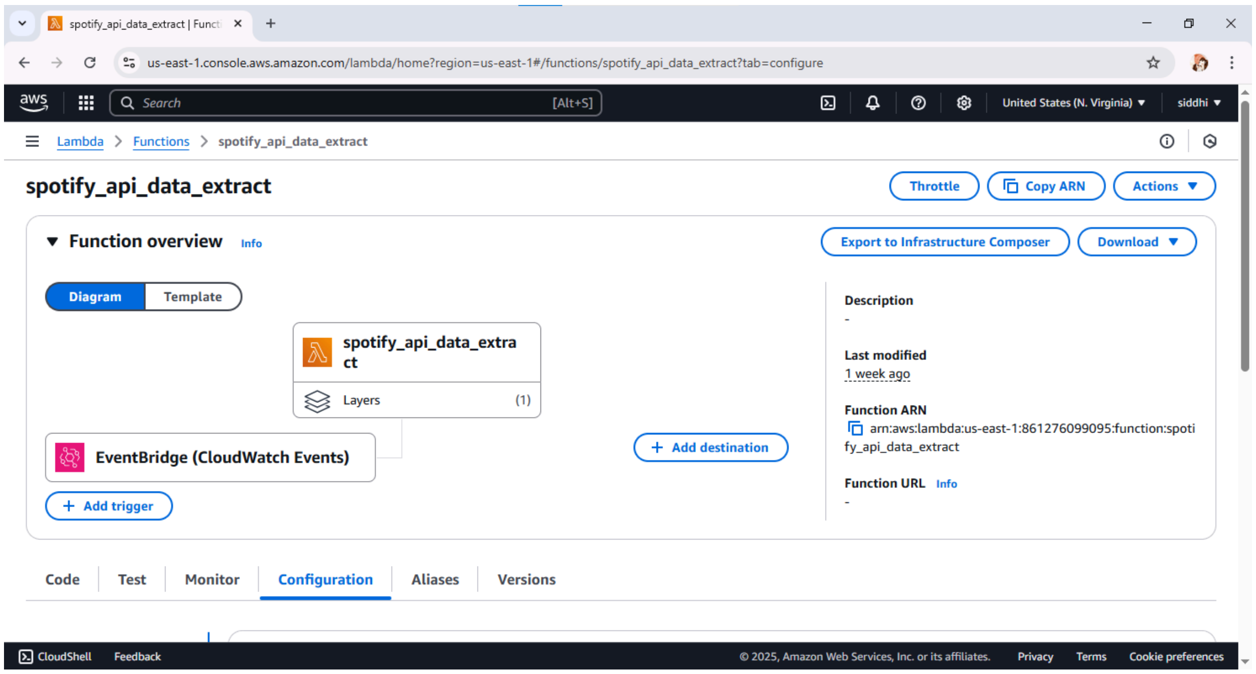
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions

Functions (2)

Last fetched 30 seconds ago

Function name	Description	Package type	Runtime	Last modified
spotify_transformation_load_function	-	Zip	Python 3.9	1 week ago
spotify_api_data_extract	-	Zip	Python 3.9	1 week ago

3. I applied a **CloudWatch alarm** to the **spotify_api_data_extract** function to **trigger it weekly** and collect **updated data**.



us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/spotify_api_data_extract?tab=configure

Lambda > Functions > spotify_api_data_extract

spotify_api_data_extract

Function overview

spotify_api_data_extract
Layers (1)

EventBridge (CloudWatch Events)

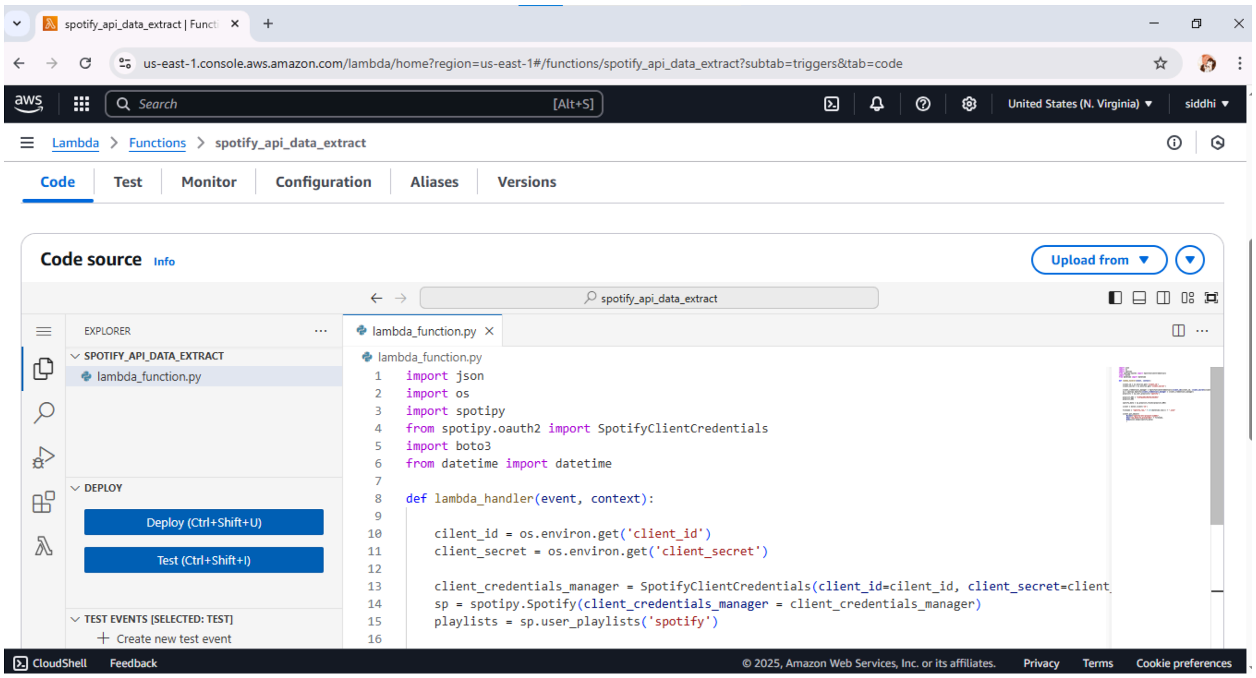
Description
-

Last modified
1 week ago

Function ARN
arn:aws:lambda:us-east-1:861276099095:function:spotify_api_data_extract

Function URL
-

Here is the function code :



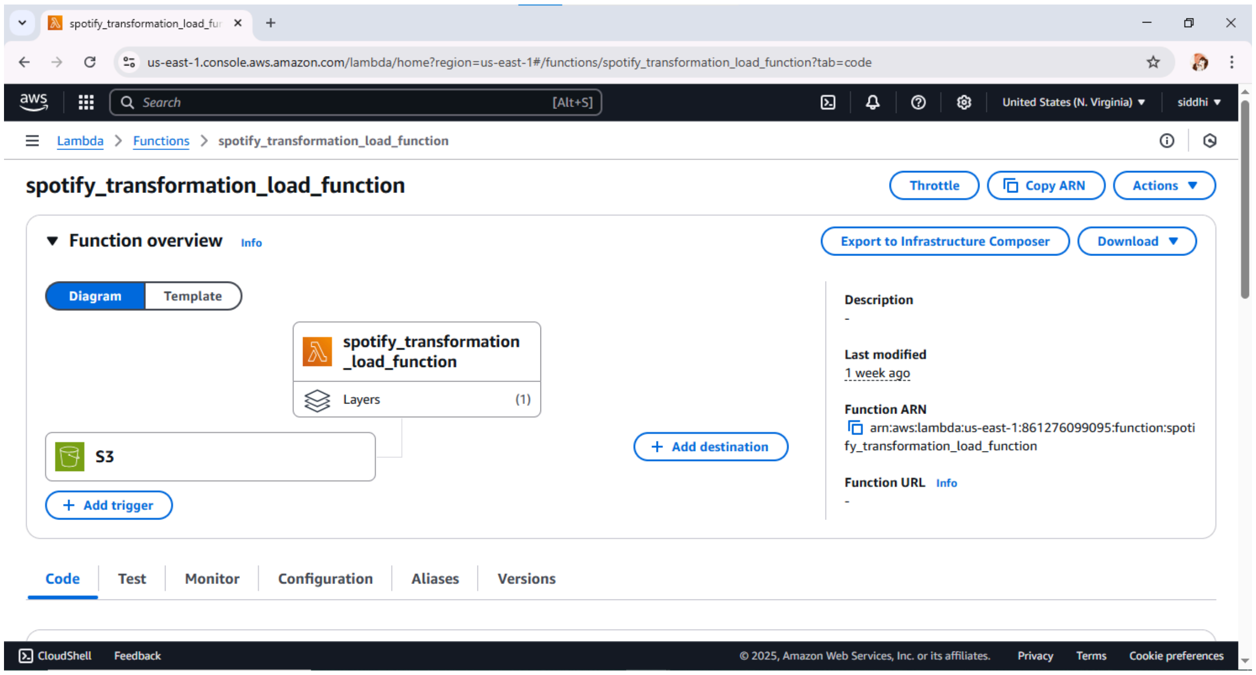
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/spotify_api_data_extract?subtab=triggers&tab=code

Lambda > Functions > spotify_api_data_extract

lambda_function.py

```python
import json
import os
import spotipy
from spotipy.oauth2 import SpotifyClientCredentials
import boto3
from datetime import datetime

def lambda_handler(event, context):

    cilent_id = os.environ.get('client_id')
    client_secret = os.environ.get('client_secret')

    client_credentials_manager = SpotifyClientCredentials(client_id=cilent_id, client_secret=client_
    sp = spotipy.Spotify(client_credentials_manager = client_credentials_manager)
    playlists = sp.user_playlists('spotify')

```

4. I added an **S3 trigger** to the **spotify_transformation_load_function**, so it **automatically triggers** whenever the **spotify_api_data_extract** function runs.



us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/spotify_transformation_load_function?tab=code

Lambda > Functions > spotify_transformation_load_function

spotify_transformation_load_function

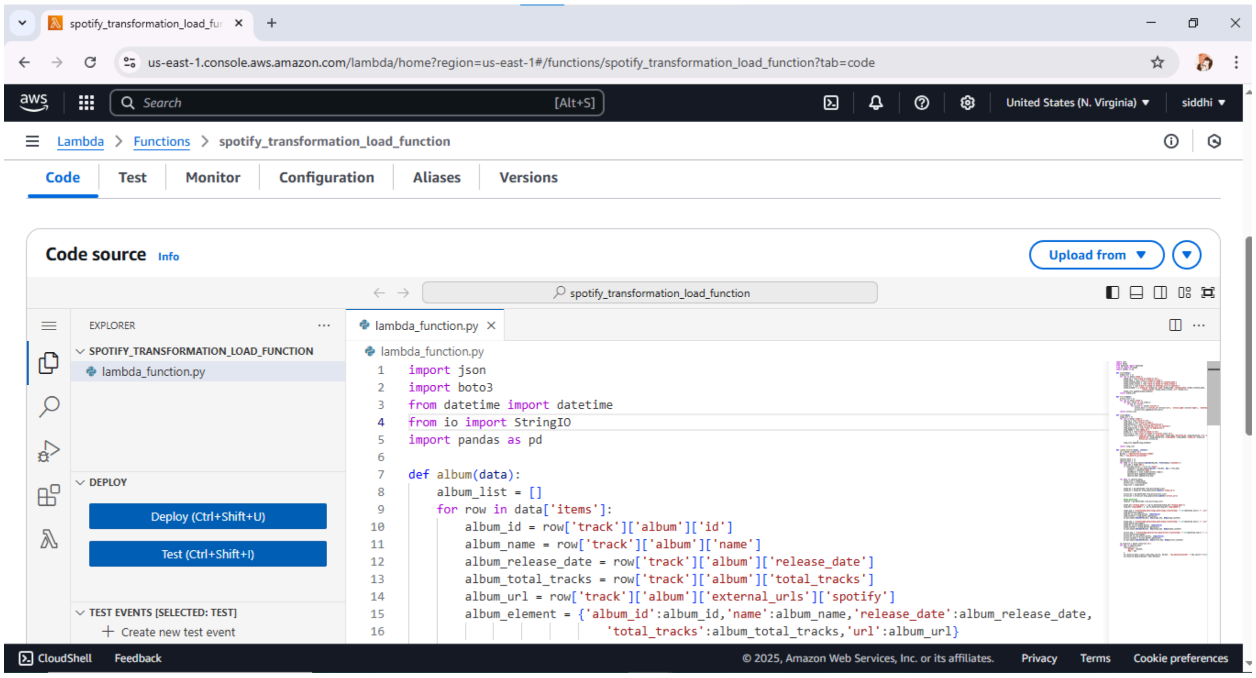
Description
-

Last modified
1 week ago

Function ARN
arn:aws:lambda:us-east-1:861276099095:function:spotify_transformation_load_function

Function URL
-

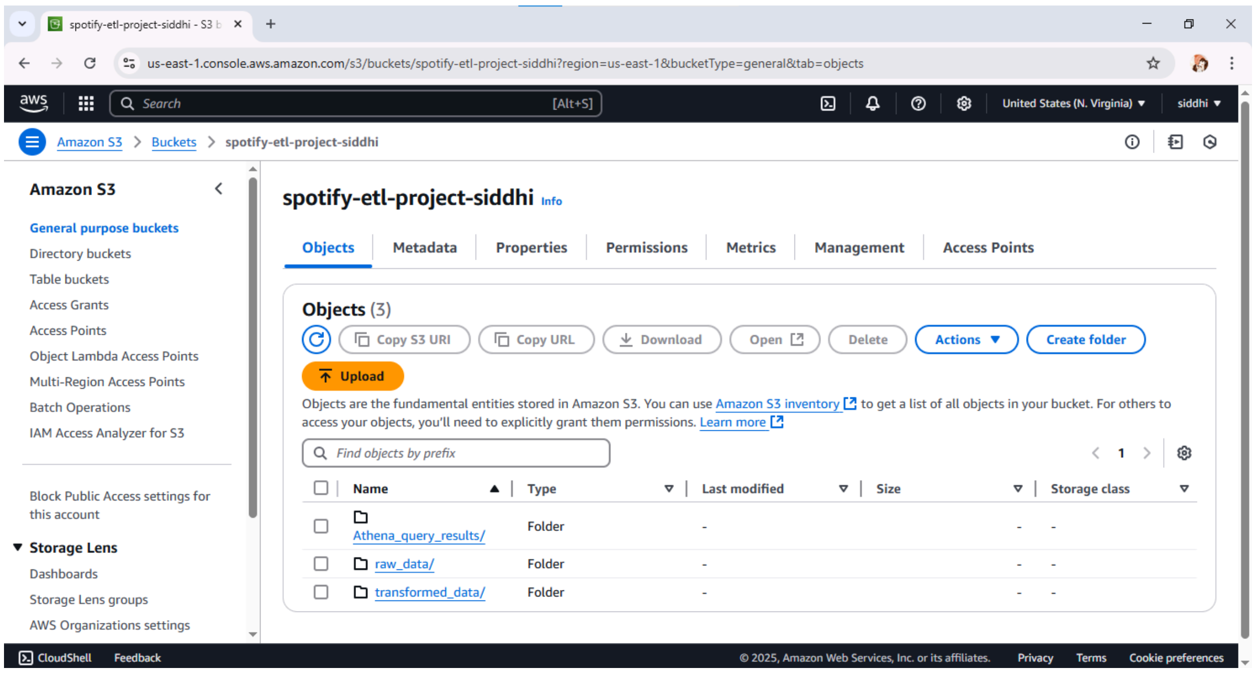
Here is the code :



```python
import json
import boto3
from datetime import datetime
from io import StringIO
import pandas as pd

def album(data):
    album_list = []
    for row in data['items']:
        album_id = row['track']['album']['id']
        album_name = row['track']['album']['name']
        album_release_date = row['track']['album']['release_date']
        album_total_tracks = row['track']['album']['total_tracks']
        album_url = row['track']['album']['external_urls']['spotify']
        album_element = {'album_id':album_id,'name':album_name,'release_date':album_release_date,
                        'total_tracks':album_total_tracks,'url':album_url}
```

5. I created three folders in my bucket:

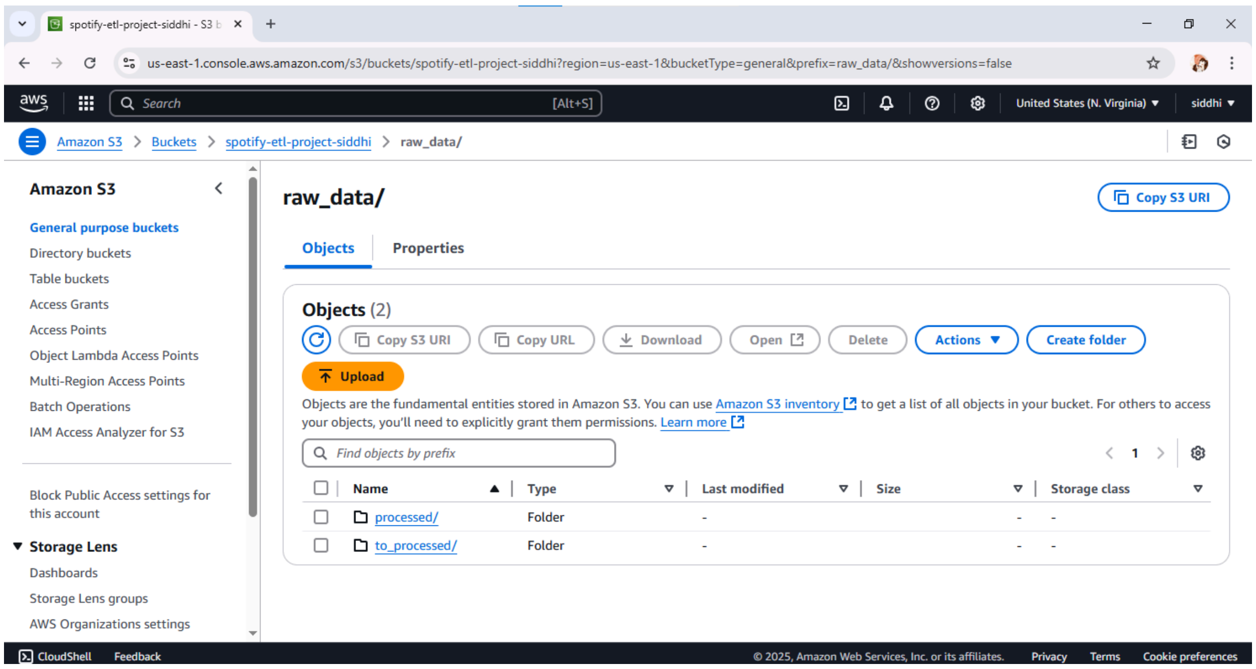
1. **raw_data** → Contains two subfolders:
   - **to_processed:** Stores data fetched from the API.
   - **processed:** Holds the processed data in **JSON format**.
2. The data in the **processed** folder is transformed by the **Lambda function** and stored separately in the **transformed_data** folder as:
   - **album_data.csv**
   - **artist_data.csv**
   - **songs_data.csv**
3. **athena_query_results** → Stores the **Athena query results**.



spotify-etl-project-siddhi - S3 b
us-east-1.console.aws.amazon.com/s3/buckets/spotify-etl-project-siddhi?region=us-east-1&bucketType=general&tab=objects
Search [Alt+S]
United States (N. Virginia)
siddhi
Amazon S3 > Buckets > spotify-etl-project-siddhi
Amazon S3
General purpose buckets
Directory buckets
Table 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
spotify-etl-project-siddhi Info
Objects
Metadata
Properties
Permissions
Metrics
Management
Access Points
Objects (3)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

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
Athena_query_results/	Folder	-	-	-
raw_data/	Folder	-	-	-
transformed_data/	Folder	-	-	-

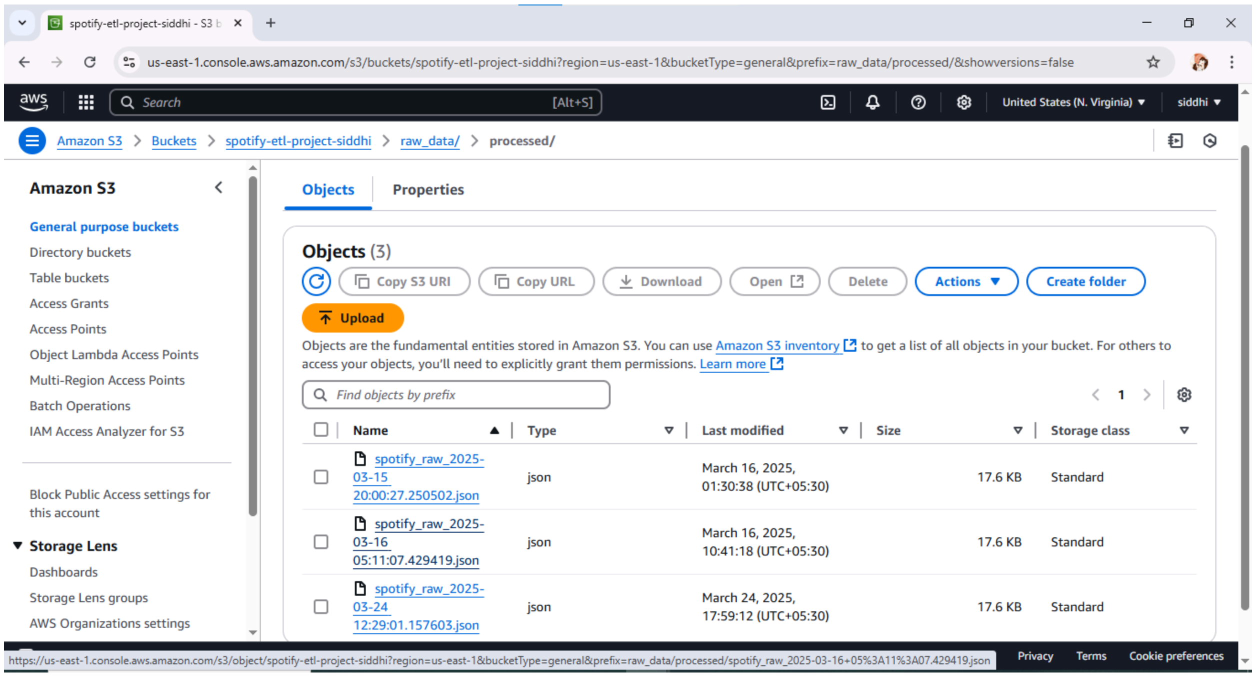


raw_data/

Objects (2)

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
processed/	Folder	-	-	-
to_processed/	Folder	-	-	-

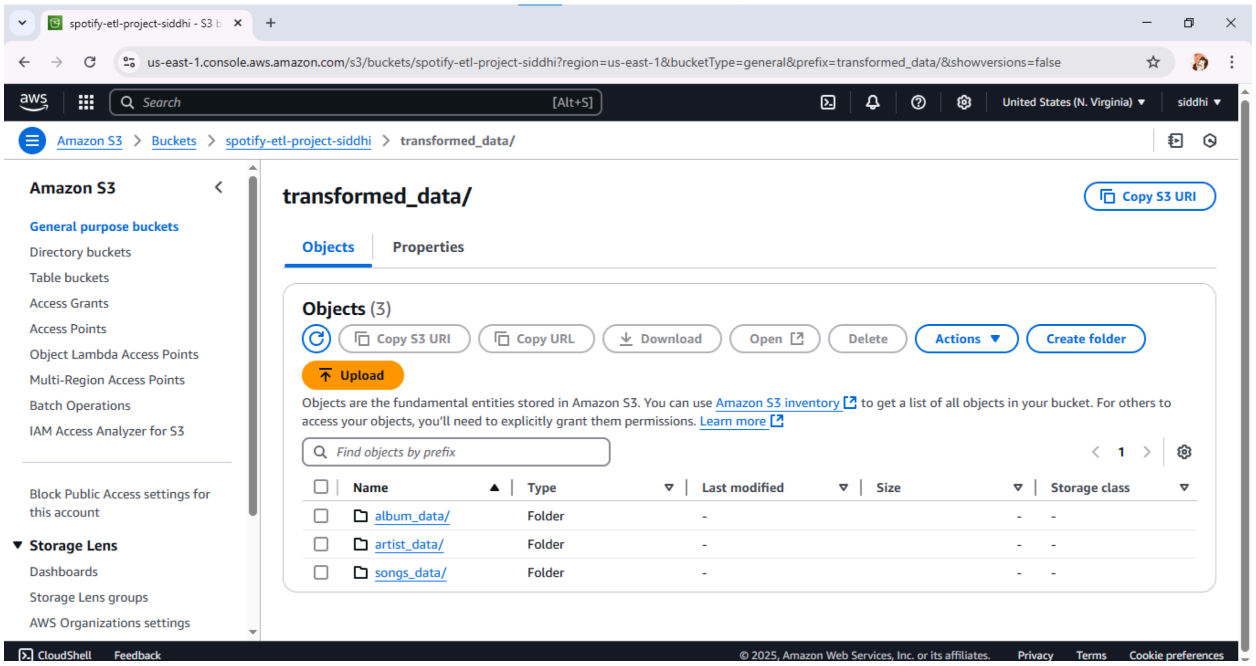


processed/

Objects (3)

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
spotify_raw_2025-03-15 20:00:27.250502.json	json	March 16, 2025, 01:30:38 (UTC+05:30)	17.6 KB	Standard
spotify_raw_2025-03-16 05:11:07.429419.json	json	March 16, 2025, 10:41:18 (UTC+05:30)	17.6 KB	Standard
spotify_raw_2025-03-24 12:29:01.157603.json	json	March 24, 2025, 17:59:12 (UTC+05:30)	17.6 KB	Standard



us-east-1.console.aws.amazon.com/s3/buckets/spotify-etl-project-siddhi?region=us-east-1&bucketType=general&prefix=transformed_data/&showversions=false

Amazon S3 > Buckets > spotify-etl-project-siddhi > transformed_data/

transformed_data/

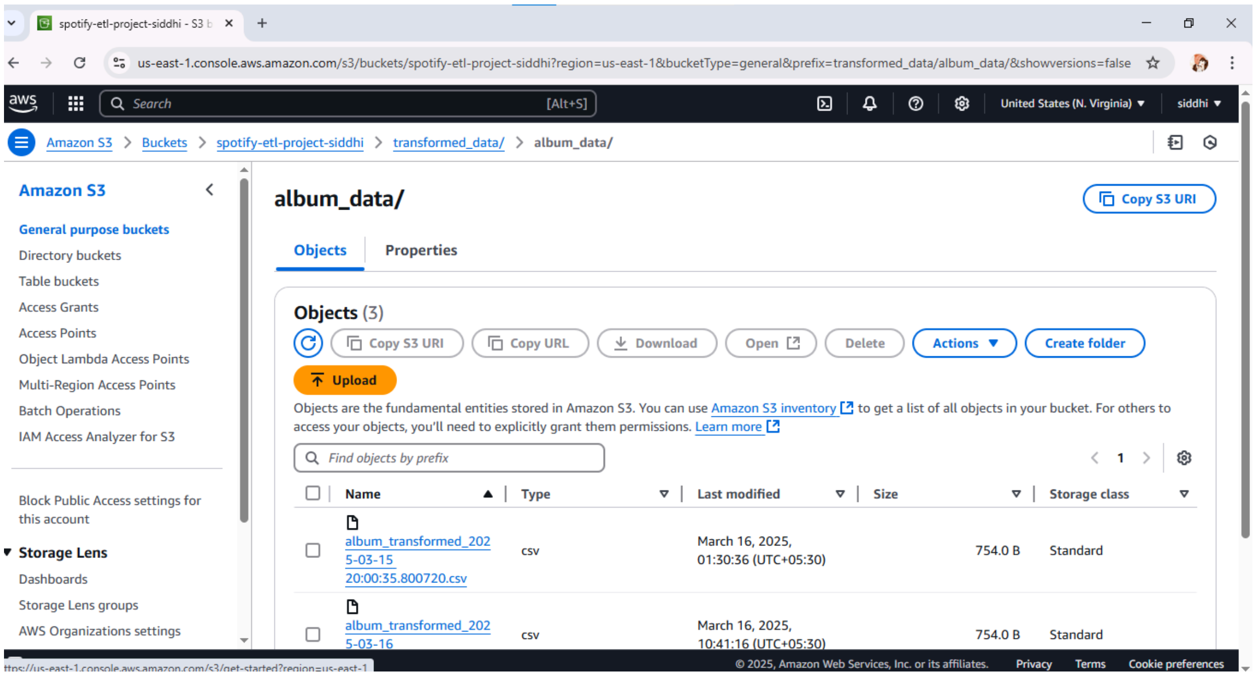
Objects (3)

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

Find objects by prefix

Name	Type	Last modified	Size	Storage class
album_data/	Folder	-	-	-
artist_data/	Folder	-	-	-
songs_data/	Folder	-	-	-

© 2025, Amazon Web Services, Inc. or its affiliates.



us-east-1.console.aws.amazon.com/s3/buckets/spotify-etl-project-siddhi?region=us-east-1&bucketType=general&prefix=transformed_data/album_data/&showversions=false

Amazon S3 > Buckets > spotify-etl-project-siddhi > transformed_data/ > album_data/

album_data/

Objects (3)

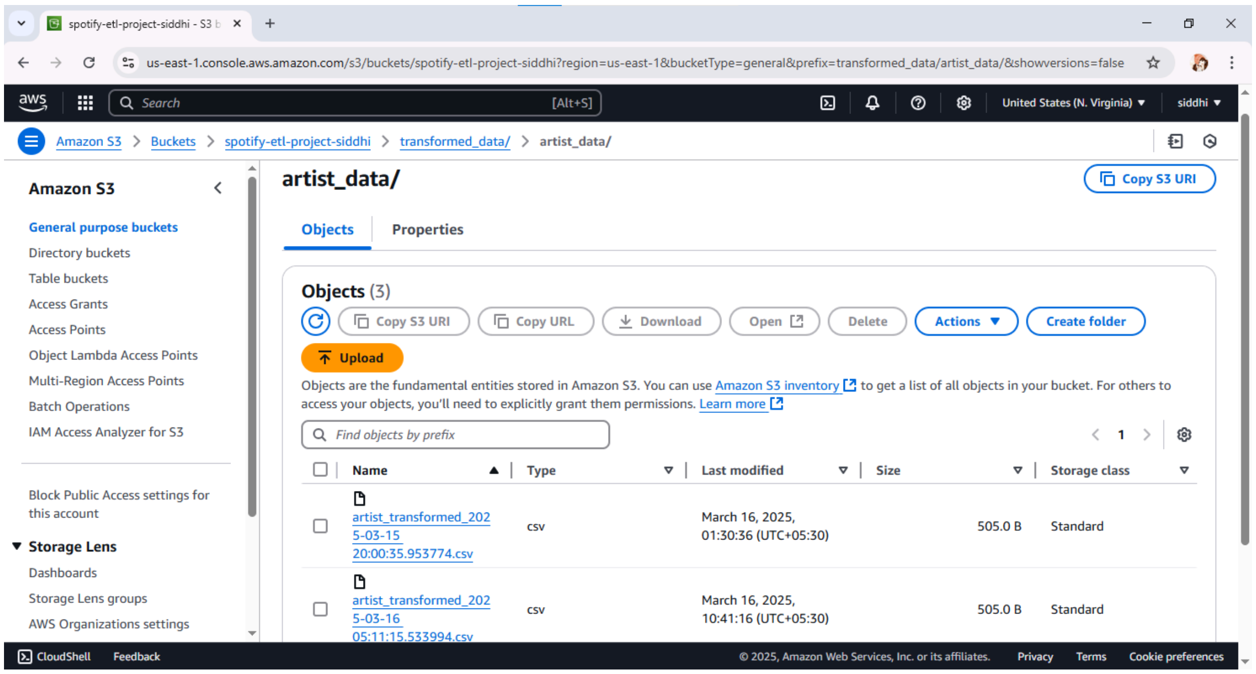
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Find objects by prefix

Name	Type	Last modified	Size	Storage class
album_transformed_2025-03-15 20:00:35.800720.csv	csv	March 16, 2025, 01:30:36 (UTC+05:30)	754.0 B	Standard
album_transformed_2025-03-16	csv	March 16, 2025, 10:41:16 (UTC+05:30)	754.0 B	Standard

© 2025, Amazon Web Services, Inc. or its affiliates.

ttps://us-east-1.console.aws.amazon.com/s3/get-started?region=us-east-1



us-east-1.console.aws.amazon.com/s3/buckets/spotify-etl-project-siddhi?region=us-east-1&bucketType=general&prefix=transformed_data/artist_data/&showversions=false

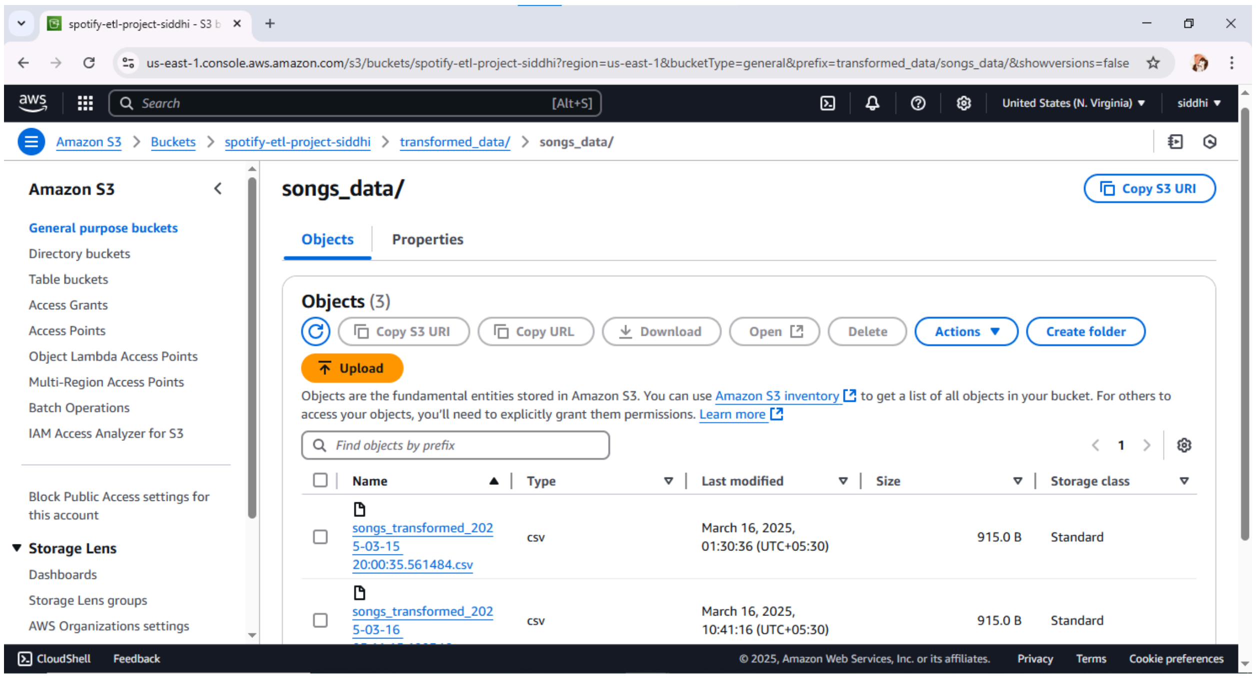
Amazon S3 > Buckets > spotify-etl-project-siddhi > transformed_data/ > artist_data/

artist_data/

Objects (3)

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
artist_transformed_2025-03-15 20:00:35.953774.csv	csv	March 16, 2025, 01:30:36 (UTC+05:30)	505.0 B	Standard
artist_transformed_2025-03-16 05:11:15.533994.csv	csv	March 16, 2025, 10:41:16 (UTC+05:30)	505.0 B	Standard



us-east-1.console.aws.amazon.com/s3/buckets/spotify-etl-project-siddhi?region=us-east-1&bucketType=general&prefix=transformed_data/songs_data/&showversions=false

Amazon S3 > Buckets > spotify-etl-project-siddhi > transformed_data/ > songs_data/

songs_data/

Objects (3)

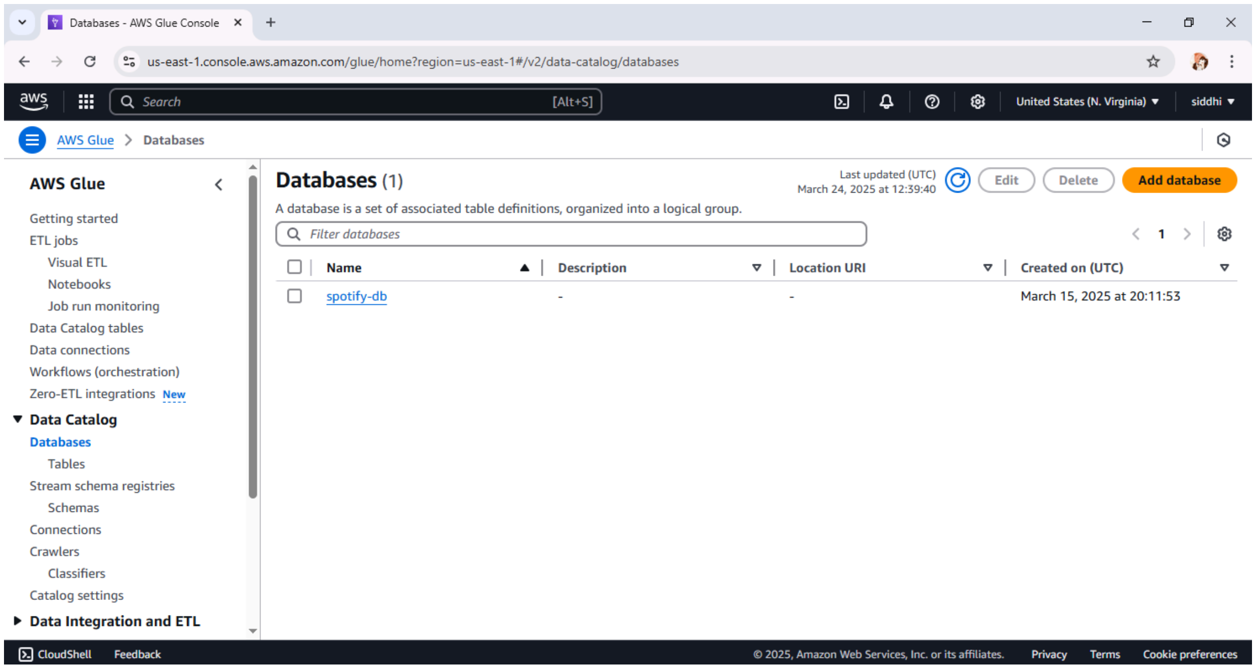
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
songs_transformed_2025-03-15 20:00:35.561484.csv	csv	March 16, 2025, 01:30:36 (UTC+05:30)	915.0 B	Standard
songs_transformed_2025-03-16	csv	March 16, 2025, 10:41:16 (UTC+05:30)	915.0 B	Standard

6. I created an **AWS Glue database** named **spotify-db**, which helps in:

- **Cataloging and organizing data** stored in S3.
- **Defining table schemas** for the transformed data.

- **Querying the data with Athena** by creating metadata tables.
- **Automating ETL workflows** for data transformation and analysis.

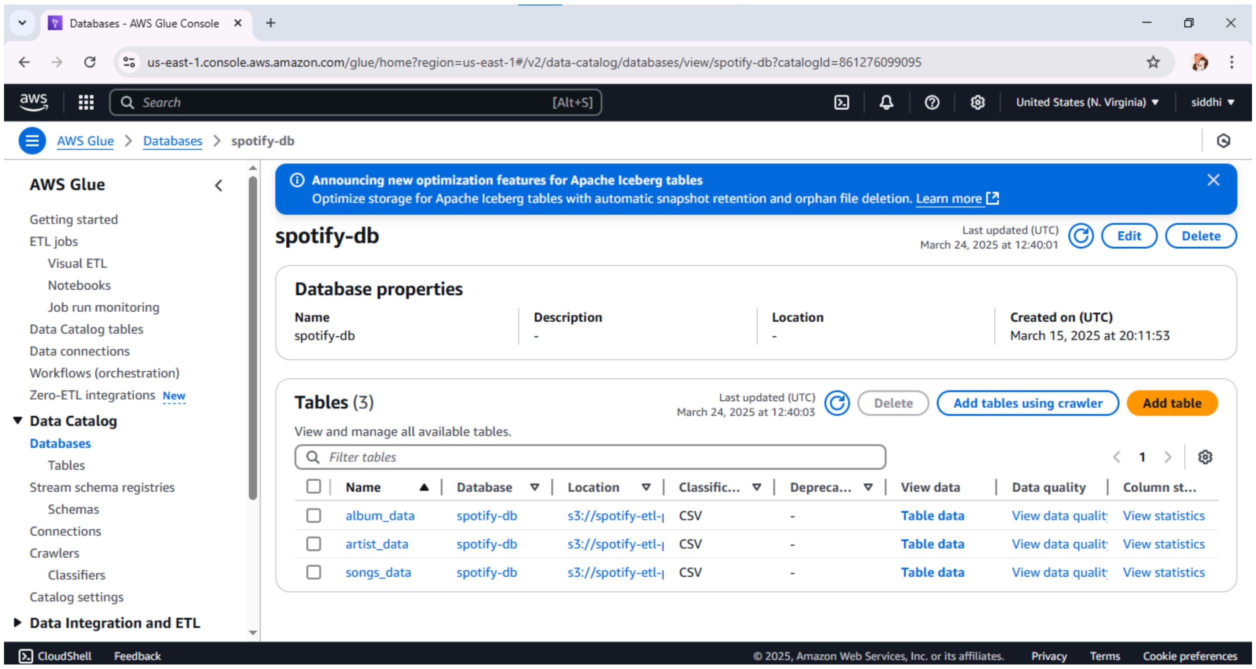


Databases (1)

Last updated (UTC) March 24, 2025 at 12:39:40

A database is a set of associated table definitions, organized into a logical group.

Name	Description	Location URI	Created on (UTC)
spotify-db	-	-	March 15, 2025 at 20:11:53



Announcing new optimization features for Apache Iceberg tables
Optimize storage for Apache Iceberg tables with automatic snapshot retention and orphan file deletion. Learn more

spotify-db

Last updated (UTC) March 24, 2025 at 12:40:01

Database properties

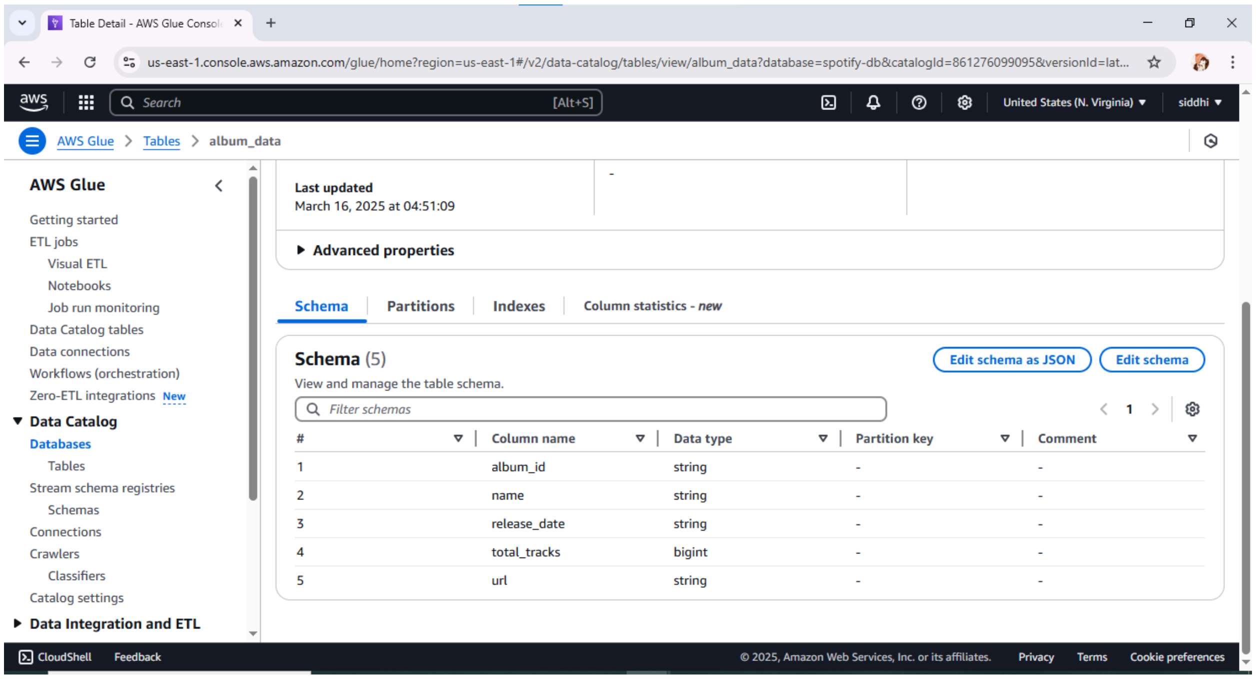
Name	Description	Location	Created on (UTC)
spotify-db	-	-	March 15, 2025 at 20:11:53

Tables (3)

Last updated (UTC) March 24, 2025 at 12:40:03

View and manage all available tables.

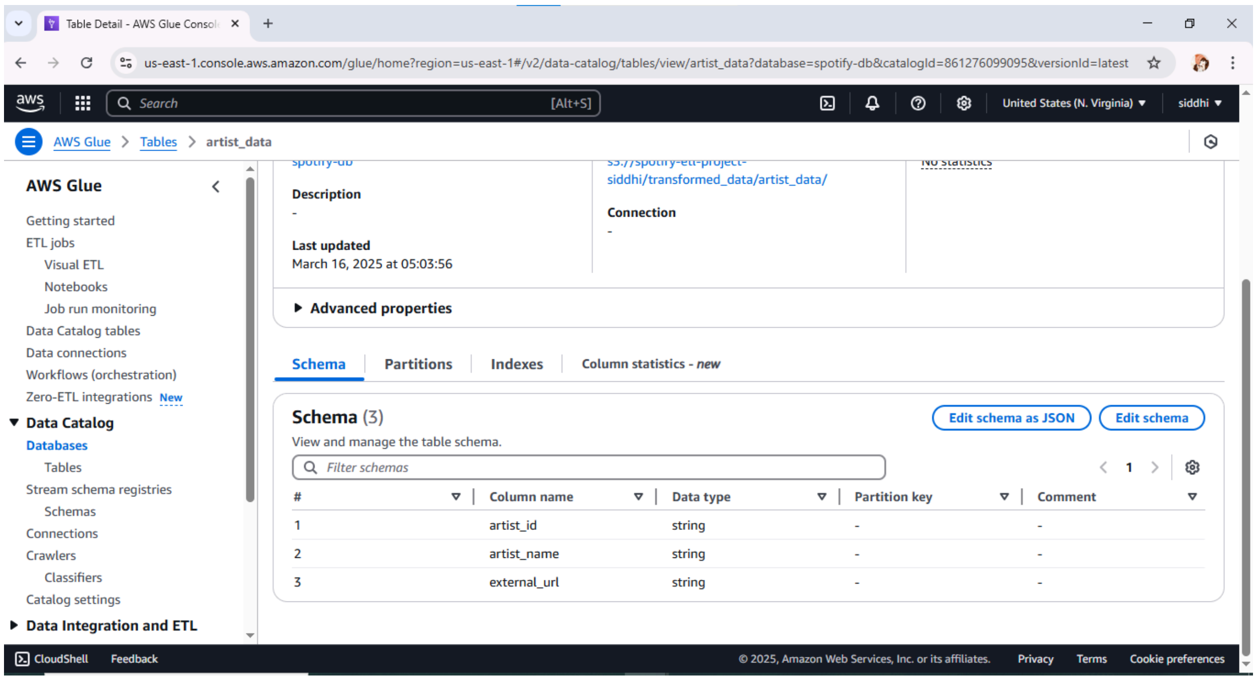
Name	Database	Location	Classific...	Depreca...	View data	Data quality	Column st...
album_data	spotify-db	s3://spotify-etl-|	CSV	-	Table data	View data qualit	View statistics
artist_data	spotify-db	s3://spotify-etl-|	CSV	-	Table data	View data qualit	View statistics
songs_data	spotify-db	s3://spotify-etl-|	CSV	-	Table data	View data qualit	View statistics



Last updated
March 16, 2025 at 04:51:09

Schema (5)
View and manage the table schema.

#	Column name	Data type	Partition key	Comment
1	album_id	string	-	-
2	name	string	-	-
3	release_date	string	-	-
4	total_tracks	bigint	-	-
5	url	string	-	-



s3://spotify-etl-project-siddhi/transformed_data/artist_data/

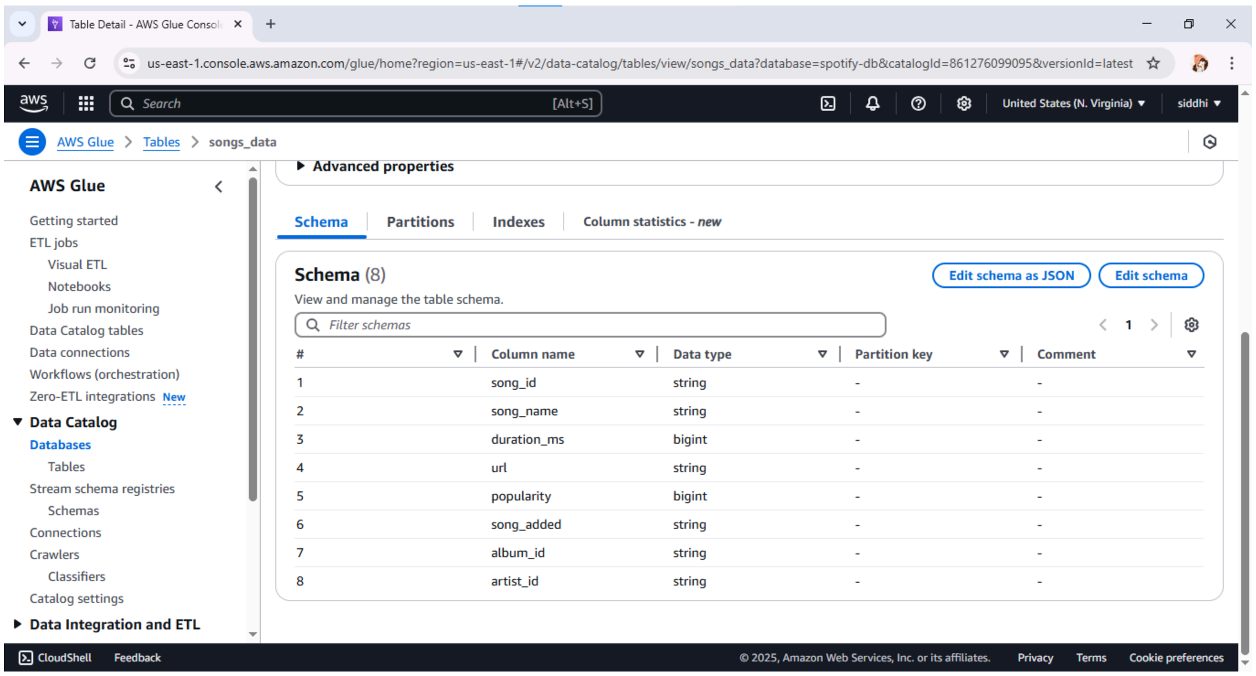
Description
-

Connection
-

Last updated
March 16, 2025 at 05:03:56

Schema (3)
View and manage the table schema.

#	Column name	Data type	Partition key	Comment
1	artist_id	string	-	-
2	artist_name	string	-	-
3	external_url	string	-	-

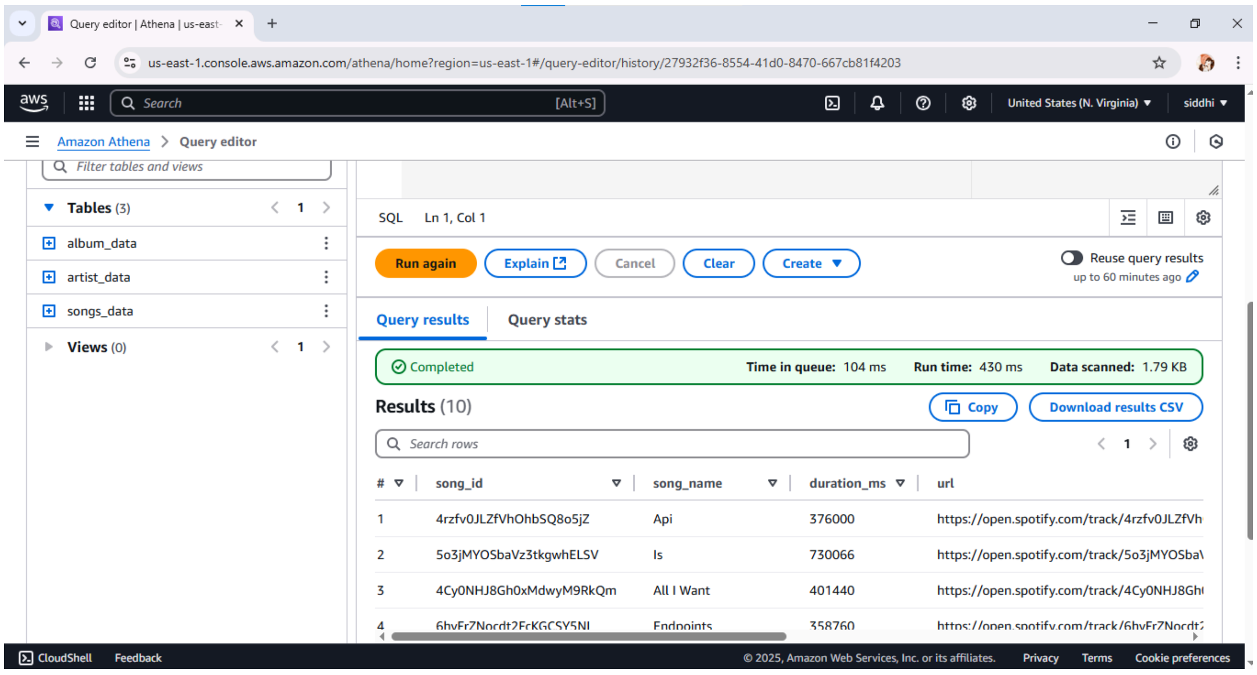


Schema (8)

View and manage the table schema.

#	Column name	Data type	Partition key	Comment
1	song_id	string	-	-
2	song_name	string	-	-
3	duration_ms	bigint	-	-
4	url	string	-	-
5	popularity	bigint	-	-
6	song_added	string	-	-
7	album_id	string	-	-
8	artist_id	string	-	-

I am using
Amazon Athena, connected to **AWS Glue**, to run queries on the **spotify-db** database for data analysis and retrieval.



Completed Time in queue: 104 ms Run time: 430 ms Data scanned: 1.79 KB

Results (10)

#	song_id	song_name	duration_ms	url
1	4rzfv0JLZfVhOhbSQ8o5jZ	Api	376000	https://open.spotify.com/track/4rzfv0JLZfVh
2	5o3jMYOSbaVz3tkgwhELSV	Is	730066	https://open.spotify.com/track/5o3jMYOSba\
3	4Cy0NHJ8Gh0xMdwyM9RkQm	All I Want	401440	https://open.spotify.com/track/4Cy0NHJ8Gh(
4	6hvFrZNocdt2FcKGCSY5NI	Endpoints	358760	https://open.spotify.com/track/6hvFrZNocdt2

Conclusion

The **Spotify ETL project** successfully automates the extraction, transformation, and loading of Spotify data using **AWS services**. The data is stored in **S3**, processed with **Lambda functions**, cataloged with **Glue**, and queried using **Athena**. This pipeline ensures efficient data processing and enables seamless analysis for gaining insights from Spotify's music data.