# Steps in AWS

## AWS S3:

**Amazon S3** ✕

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

Feature spotlight 7

▶ AWS Marketplace for S3

Amazon S3 › Buckets › spotify-etl-bucket-kavya

# spotify-etl-bucket-kavya  Info

**Objects**   Properties   Permissions   Metrics   Management   Access Points

## Objects (3) Info

[ ↻ ]  [ Copy S3 URI ]  [ Copy URL ]  [ Download ]  [ Open ↗ ]  [ Delete ]  [ 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 ↗

🔍 Find objects by prefix                                    ‹ 1 › ⚙

| ☐ | Name ▲ | Type ▽ | Last modified ▽ | Size ▽ | Storage class ▽ |
|---|---|---|---|---|---|
| ☐ | 📄 Output/ | Folder | - | - | - |
| ☐ | 📄 raw_data/ | Folder | - | - | - |
| ☐ | 📄 transformed_data/ | Folder | - | - | - |

---

**Amazon S3** ✕

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

Feature spotlight 7

▶ AWS Marketplace for S3

Amazon S3 › Buckets › spotify-etl-bucket-kavya › raw_data/

# raw_data/                                    [ Copy S3 URI ]

**Objects**   Properties

## Objects (2) Info

[ ↻ ]  [ Copy S3 URI ]  [ Copy URL ]  [ Download ]  [ Open ↗ ]  [ Delete ]  [ 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 ↗

🔍 Find objects by prefix                                    ‹ 1 › ⚙

| ☐ | Name ▲ | Type ▽ | Last modified ▽ | Size ▽ | Storage class ▽ |
|---|---|---|---|---|---|
| ☐ | 📄 processed/ | Folder | - | - | - |
| ☐ | 📄 to_process/ | Folder | - | - | - |

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

Feature spotlight 7

▶ AWS Marketplace for S3

Amazon S3 > Buckets > spotify-etl-bucket-kavya > transformed_data/

# transformed_data/

Copy S3 URI

**Objects** | Properties

## Objects (3) Info

🔄 | Copy S3 URI | Copy URL | ⬇ Download | Open ↗ | Delete | 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 ↗

🔍 Find objects by prefix

‹ 1 › ⚙

| ☐ | Name ▲ | Type ▽ | Last modified ▽ | Size ▽ | Storage class ▽ |
|---|---|---|---|---|---|
| ☐ | 📄 album_data/ | Folder | - | - | - |
| ☐ | 📄 artists_data/ | Folder | - | - | - |
| ☐ | 📄 songs_data/ | Folder | - | - | - |

# AWS Lambda:

**AWS Lambda** ✕

Dashboard

Applications

Functions

▼ Additional resources

Code signing configurations

Event source mappings

Layers

Replicas

▼ Related AWS resources

Step Functions state machines

Lambda > Functions

## Functions (2)

Last fetched now 🔄 | Actions ▼ | Create function

🔍 Filter by tags and attributes or search by keyword

‹ 1 › ⚙

| ☐ | Function name ▽ | Description ▽ | Package type ▽ | Runtime ▽ | Last modified ▽ |
|---|---|---|---|---|---|
| ☐ | spotify_data_api_extract | - | Zip | Python 3.8 | 6 days ago |
| ☐ | spotify_transformation_load_function | - | Zip | Python 3.8 | 6 days ago |

# spotify_data_api_extract

Throttle | Copy ARN | Actions ▼

## ▼ Function overview  Info

Export to Application Composer | Download ▼

Diagram | Template

```
λ  spotify_data_api_extract
≋  Layers                        (1)
```

+ Add trigger

+ Add destination

**Description**
-

**Last modified**
6 days ago

**Function ARN**
arn:aws:lambda:us-east-1:471112946910:function:spotify_data_api_extract

**Function URL**  Info
-

---

Code | Test | Monitor | Configuration | Aliases | Versions

### Code source  Info

Upload from ▼

▲ File  Edit  Find  View  Go  Tools  Window    Test ▼  Deploy    ⛶ ⚙

🔍 Go to Anything (⌘ P)    📋 | lambda_function ✕ | Environment Vari ✕ ⊕

```
Environment
▼ 🗁 spotify_data_api_ex  ⚙▼
    ◇ lambda_function.py
```

```
1  import json
2  import os
3  import spotipy
4  from spotipy.oauth2 import SpotifyClientCredentials
5  import boto3
6  from datetime import datetime
7
8
9  def lambda_handler(event, context):
10
11     client_id = os.environ.get('client_id')
12     client_secret = os.environ.get('client_secret')
13
```

---

# spotify_transformation_load_function

Throttle | Copy ARN | Actions ▼

## ▼ Function overview  Info

Export to Application Composer | Download ▼

Diagram | Template

```
λ  spotify_transformation_load_f
   unction
≋  Layers                        (1)
```

🪣 S3

+ Add trigger

+ Add destination

**Description**
-

**Last modified**
6 days ago

**Function ARN**
arn:aws:lambda:us-east-1:471112946910:function:spotify_transformation_load_function

**Function URL**  Info
-

---

Code | Test | Monitor | Configuration | Aliases | Versions

### Code source  Info

Upload from ▼

▲ File  Edit  Find  View  Go  Tools  Window    Test ▼  Deploy    ⛶ ⚙

🔍 Go to Anything (⌘ P)    📋 | lambda_function ✕ | Execution results ✕ | Environment Var ✕ ⊕

```
Environment
▼ 🗁 spotify_transformati  ⚙▼
    ◇ lambda_function.py
```

```
1  import json
2  import boto3
3  from datetime import datetime
4  from io import StringIO
5  import pandas as pd
6
7  def album(data):
8      album_list = []
9      for row in data['items']:
10         album_id = row['track']['album']['id']
11         album_name = row['track']['album']['name']
12         album_release_date = row['track']['album']['release_date']
13         album_total_tracks = row['track']['album']['total_tracks']
14         album_url = row['track']['album']['external_urls']['spotify']
15         album_element = {'album_id':album_id,'name':album_name,'release_date':album_release_date,
```

# AWS Glue:



**AWS Glue** ✕

- Getting started
- ETL jobs
  - Visual ETL
  - Notebooks
  - Job run monitoring
- Data Catalog tables
- Data connections
- Workflows (orchestration)
- **Data Catalog**
  - Databases
    - Tables
  - Stream schema registries
    - Schemas
  - Connections
  - **Crawlers**
    - Classifiers

AWS Glue > Crawlers

## Crawlers

A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

### Crawlers (3)  Info

Last updated (UTC)
February 18, 2024 at 20:59:21

[ Action ▼ ]  [ Run ]  [ Create crawler ]

View and manage all available crawlers.

🔍 Filter crawlers

< 1 >  ⚙

| ☐ | Name ▽ | State ▽ | Schedule | Last run ▽ | Last run timestamp ▽ | Log | Table changes from la… |
|---|--------|---------|----------|------------|----------------------|-----|------------------------|
| ☐ | spotify_album_crawler | ⊘ Ready | | ⊘ Succeeded | February 12, 2024 at 2… | View log ↗ | 1 created |
| ☐ | spotify_artist_crawler | ⊘ Ready | | ⊘ Succeeded | February 12, 2024 at 2… | View log ↗ | 1 created |
| ☐ | spotify_songs_crawler | ⊘ Ready | | ⊘ Succeeded | February 12, 2024 at 2… | View log ↗ | 1 created |

---

- Getting started
- ETL jobs
  - Visual ETL
  - Notebooks
  - Job run monitoring
- **Data Catalog tables**
- Data connections
- Workflows (orchestration)
- **Data Catalog**
  - Databases
    - Tables
  - Stream schema registries
    - Schemas
  - Connections
  - Crawlers
    - Classifiers
  - Catalog settings

AWS Glue > Tables

## Tables

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

### Tables (3)

Last updated (UTC)
February 18, 2024 at 20:59:08

[ Delete ]  [ Add tables using crawler ]  [ Add table ]

View and manage all available tables.

🔍 Filter tables

< 1 >  ⚙

| ☐ | Name ▲ | Database ▽ | Location ▽ | Classification ▽ | Deprecated ▽ | View data | Data quality |
|---|--------|-----------|-----------|------------------|--------------|-----------|--------------|
| ☐ | album_data | spotify_db | s3://spotify-etl-bucket-kavy | CSV | - | Table data | View data quality |
| ☐ | artists_data | spotify_db | s3://spotify-etl-bucket-kavy | CSV | - | Table data | View data quality |
| ☐ | songs_data | spotify_db | s3://spotify-etl-bucket-kavy | CSV | - | Table data | View data quality |

# Athena:



Amazon Athena > Query editor

Editor | Recent queries | Saved queries | Settings

Workgroup [ primary ▼ ]

**Data**

Data source
[ AwsDataCatalog ▼ ]

Database
[ spotify_db ▼ ]

Tables and views  [ Create ▼ ]

🔍 Filter tables and views

▼ Tables (3)  < 1 >
- album_data
- artists_data
- songs_data

▶ Views (0)  < 1 >

Query 1 | ⊘ Query 2 | +  ▼

```
1  SELECT * FROM "spotify_db"."songs_data" limit 10;
```

SQL  Ln 1, Col 1

[ Run again ]  [ Explain ↗ ]  [ Cancel ]  [ Clear ]  [ Create ▼ ]

Reuse query results
up to 60 minutes ago ✎

**Query results** | Query stats

⊘ Completed

Time in queue: 107 ms   Run time: 423 ms   Data scanned: 8.74 KB

**Results (10)**

[ Copy ]  [ Download results ]

🔍 Search rows

< 1 >  ⚙

| # ▽ | song_id ▽ | song_name ▽ | duration_ms ▽ | url ▽ | popularity ▽ | so |
|-----|-----------|-------------|---------------|-------|--------------|-----|
| 1 | 0R6NfOiLzLj4O5VbYSJAjf | La Diabla | 172264 | https://open.spotify.com/track/0R6NfOiLzLj4O5VbYSJAjf | 99 | 20 |
| 2 | 3rUGC1vUpkDG9CZFHMur1t | greedy | 131872 | https://open.spotify.com/track/3rUGC1vUpkDG9CZFHMur1t | 100 | 20 |