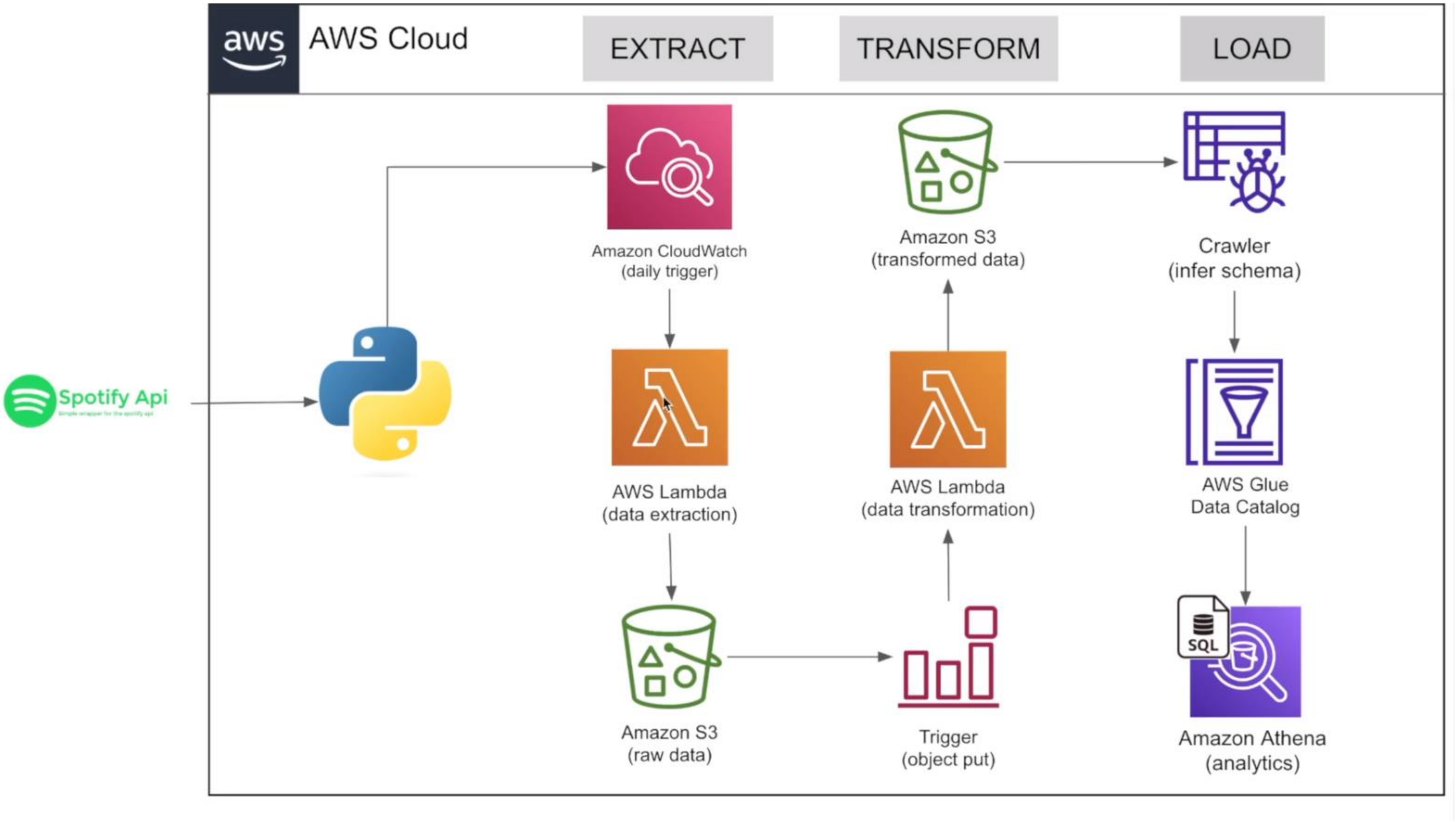
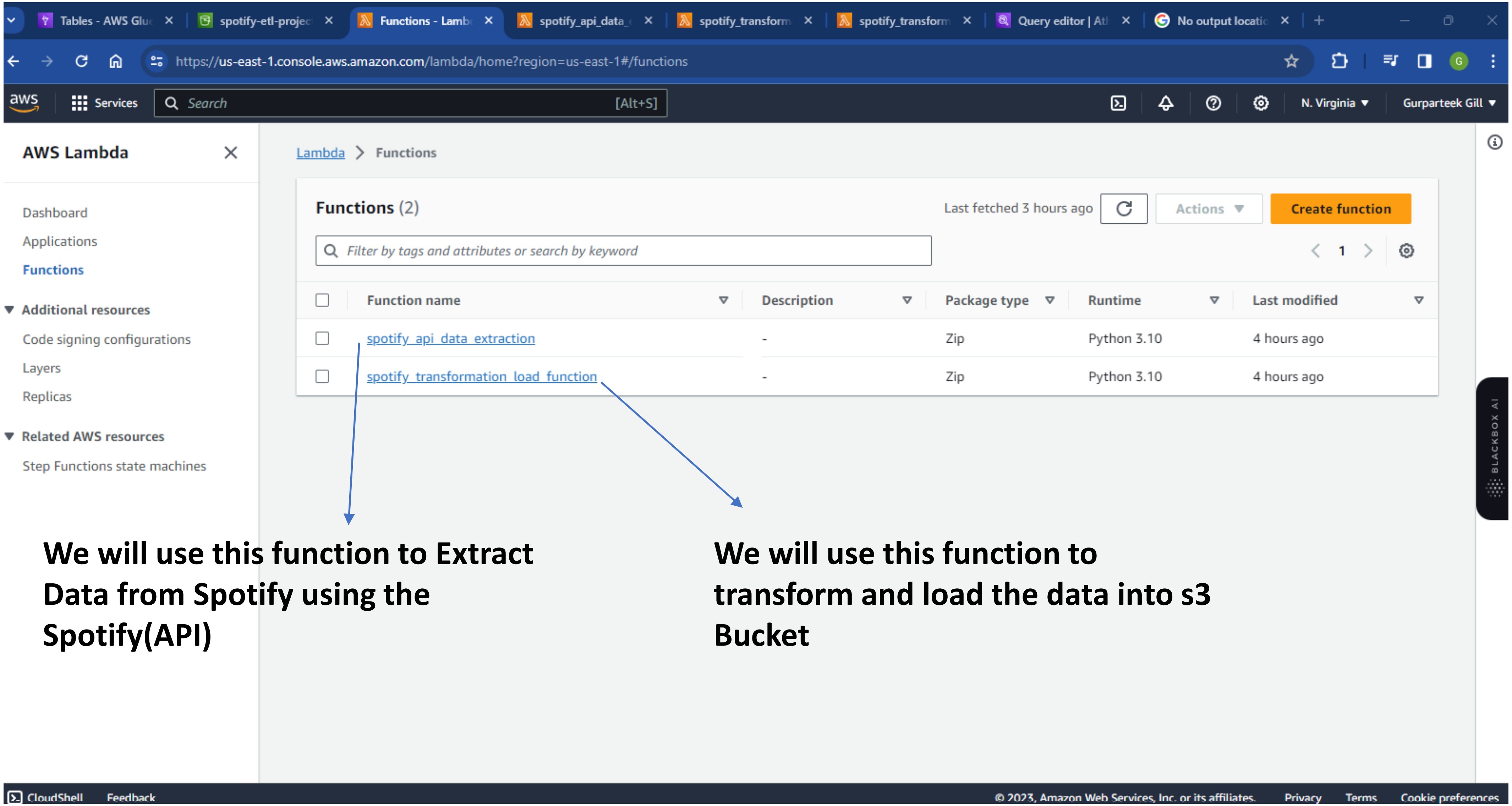
Architecture OF ETL Pipeline Created Using AWS





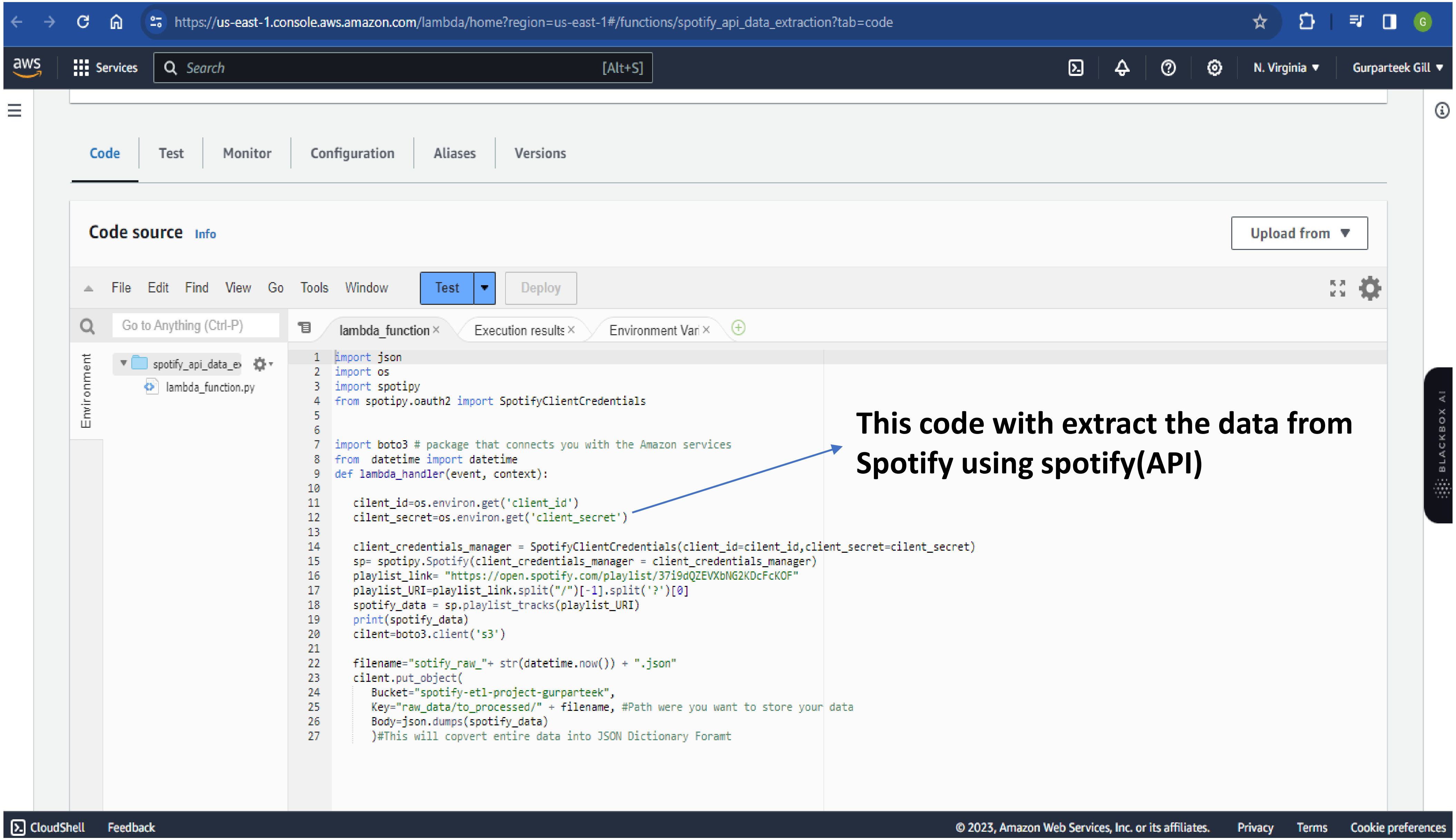
Functions (2)

□	Function name	Description	Package type	Runtime	Last modified
□	spotify_api_data_extraction	-	Zip	Python 3.10	4 hours ago
□	spotify_transformation_load_function	-	Zip	Python 3.10	4 hours ago

We will use this function to Extract Data from Spotify using the Spotify(API)

We will use this function to transform and load the data into s3 Bucket

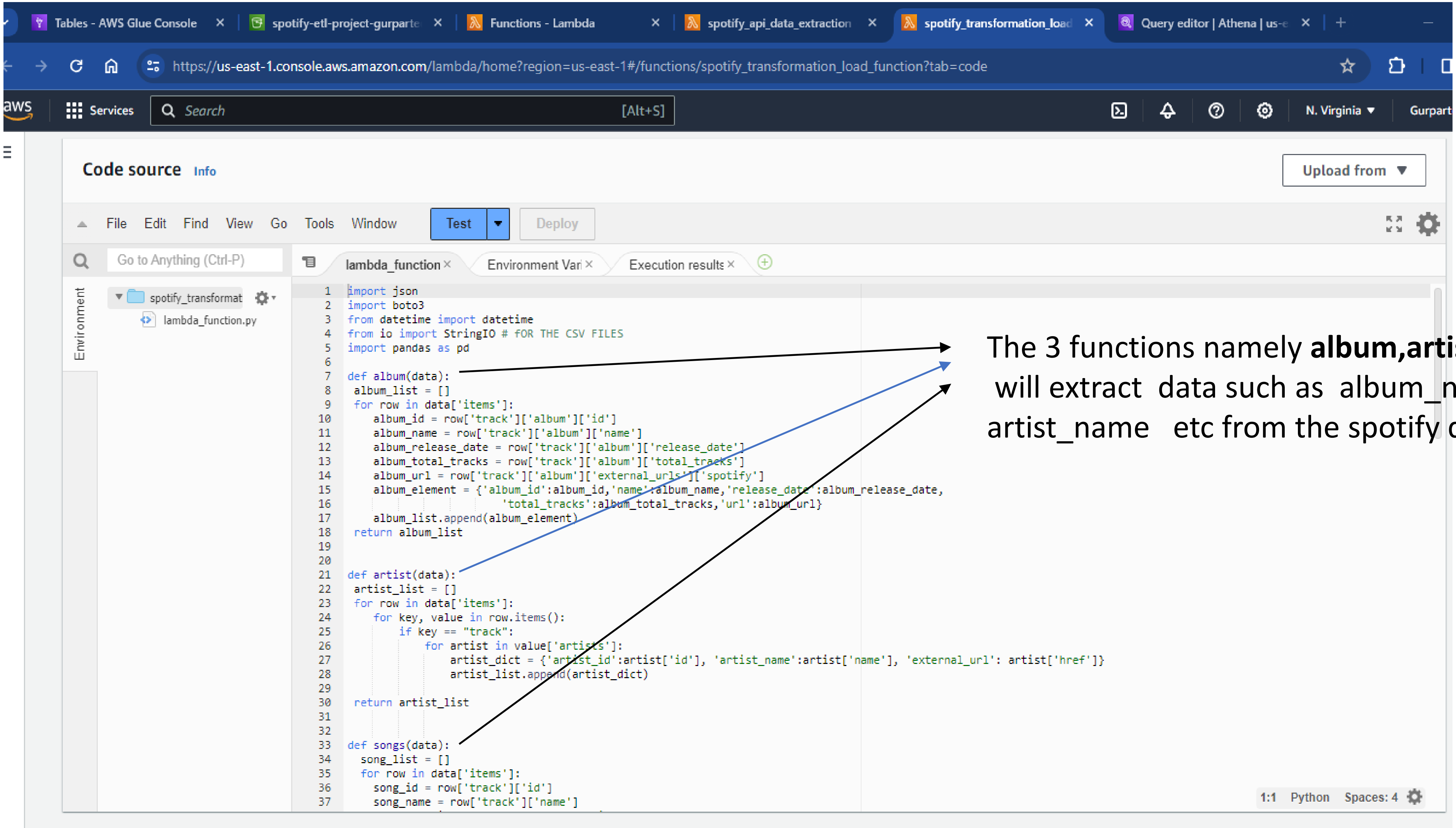
Spotify_api_data_extraction (Function)



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


import boto3 # package that connects you with the Amazon services
from  datetime import datetime
def lambda_handler(event, context):

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

    client_credentials_manager = SpotifyClientCredentials(client_id=cilent_id,client_secret=cilent_secret)
    sp= spotipy.Spotify(client_credentials_manager = client_credentials_manager)
    playlist_link= "https://open.spotify.com/playlist/37i9dQZEVXbNG2KDcFcKOF"
    playlist_URI=playlist_link.split("/")[-1].split('?')[0]
    spotify_data = sp.playlist_tracks(playlist_URI)
    print(spotify_data)
    cilent=boto3.client('s3')

    filename="sotify_raw_"+ str(datetime.now()) + ".json"
    cilent.put_object(
        Bucket="spotify-etl-project-gurparteek",
        Key="raw_data/to_processed/" + filename, #Path were you want to store your data
        Body=json.dumps(spotify_data)
        )#This will copvert entire data into JSON Dictionary Foramt
```

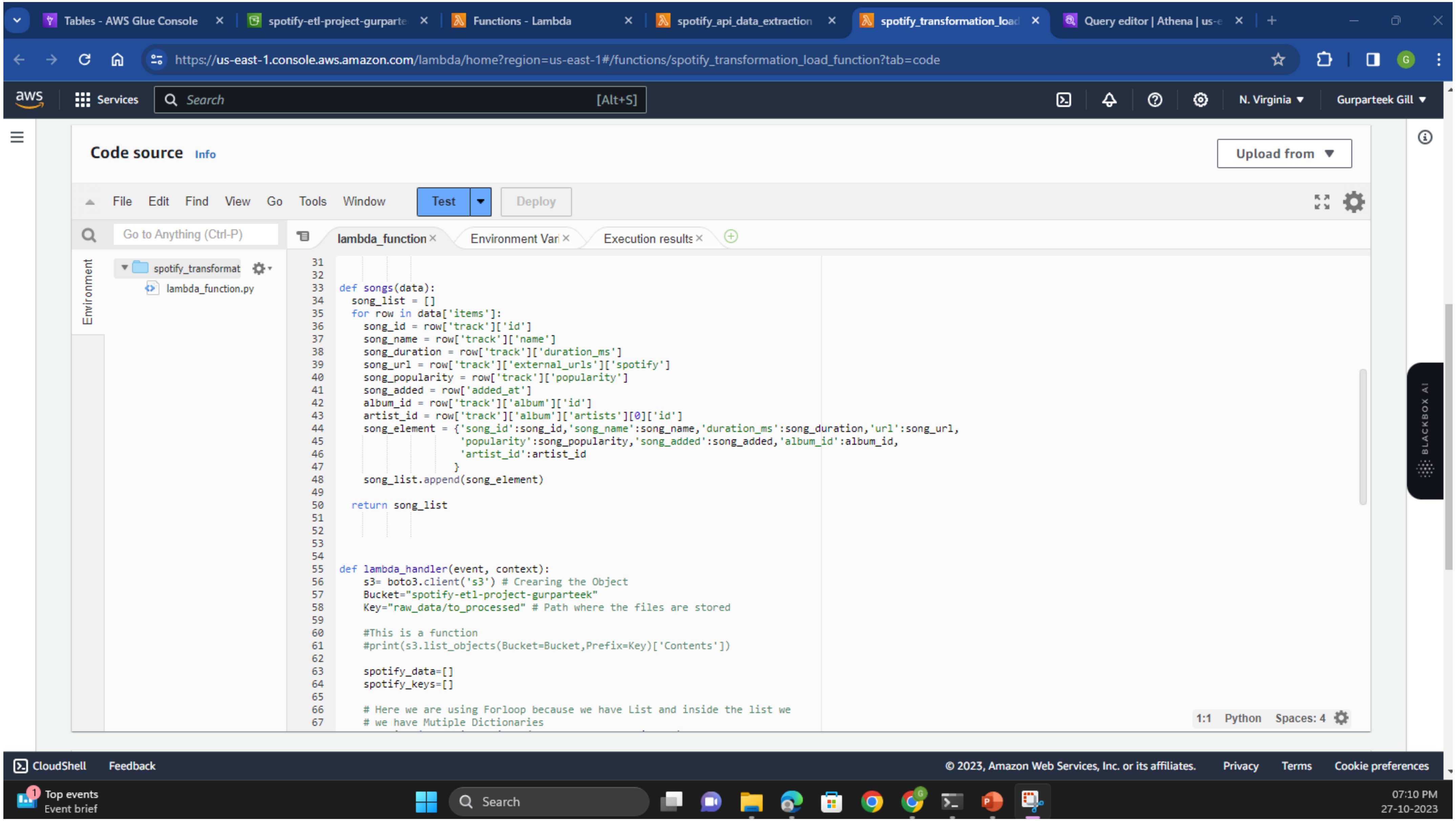
This code with extract the data from Spotify using spotify(API)

Spotify_transformation_load_function (Function)

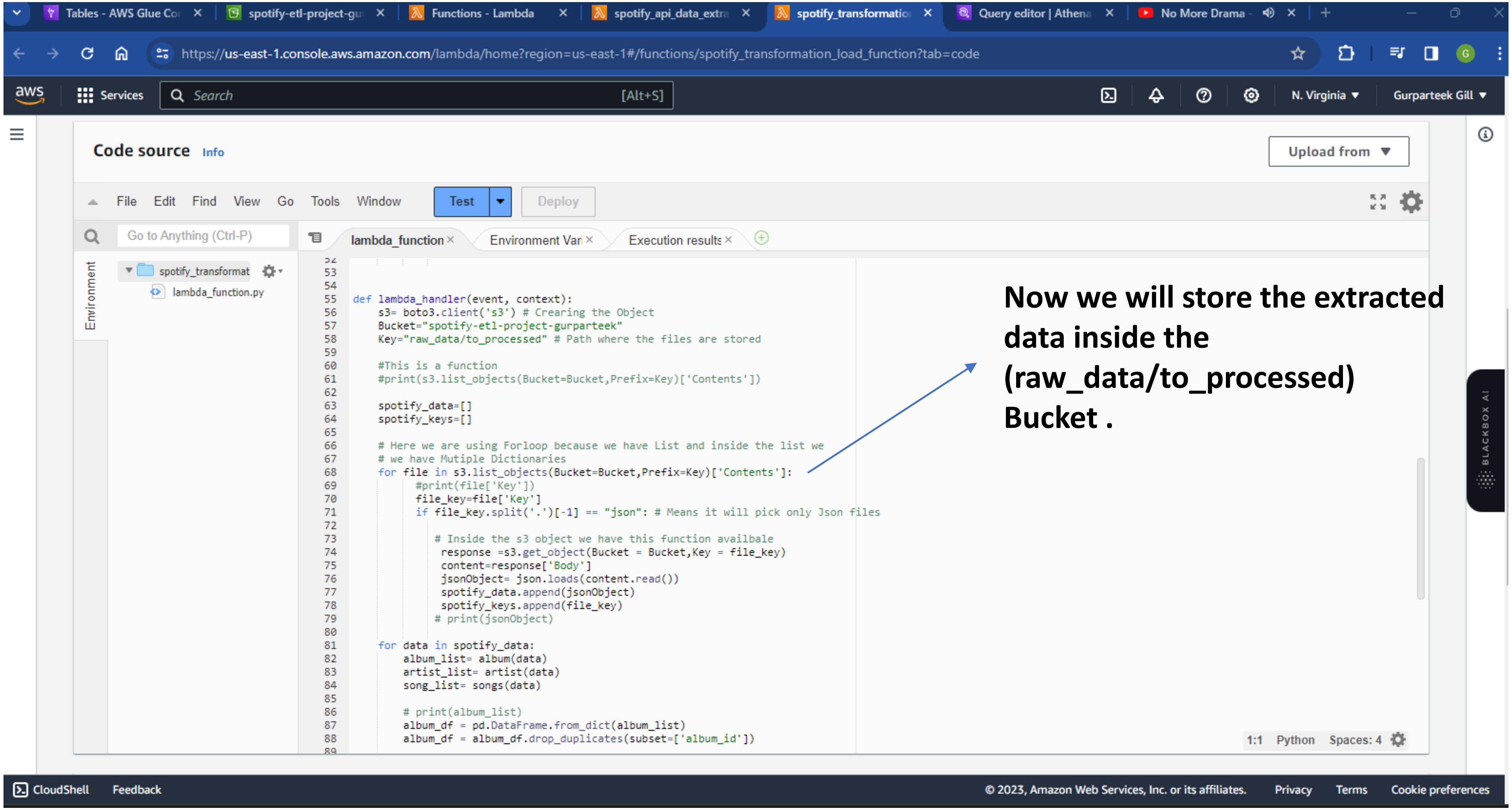

Tables - AWS Glue Console
spotify-etl-project-gurparte
Functions - Lambda
spotify_api_data_extraction
spotify_transformation_load
Query editor | Athena | us-e
https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/spotify_transformation_load_function?tab=code
aws
Services
Search
[Alt+S]
N. Virginia
Gurpart
Code source
Info
Upload from
File
Edit
Find
View
Go
Tools
Window
Test
Deploy
Go to Anything (Ctrl-P)
lambda_function
Environment Var
Execution results
Environment
spotify_transformat
lambda_function.py
1:1
Python
Spaces: 4

```python
import json
import boto3
from datetime import datetime
from io import StringIO # FOR THE CSV FILES
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}
    album_list.append(album_element)
 return album_list


def artist(data):
 artist_list = []
 for row in data['items']:
    for key, value in row.items():
        if key == "track":
            for artist in value['artists']:
                artist_dict = {'artist_id':artist['id'], 'artist_name':artist['name'], 'external_url': artist['href']}
                artist_list.append(artist_dict)

 return artist_list


def songs(data):
  song_list = []
  for row in data['items']:
    song_id = row['track']['id']
    song_name = row['track']['name']
```

The 3 functions namely **album,artist and songs** will extract data such as album_name, artist_name etc from the spotify data.

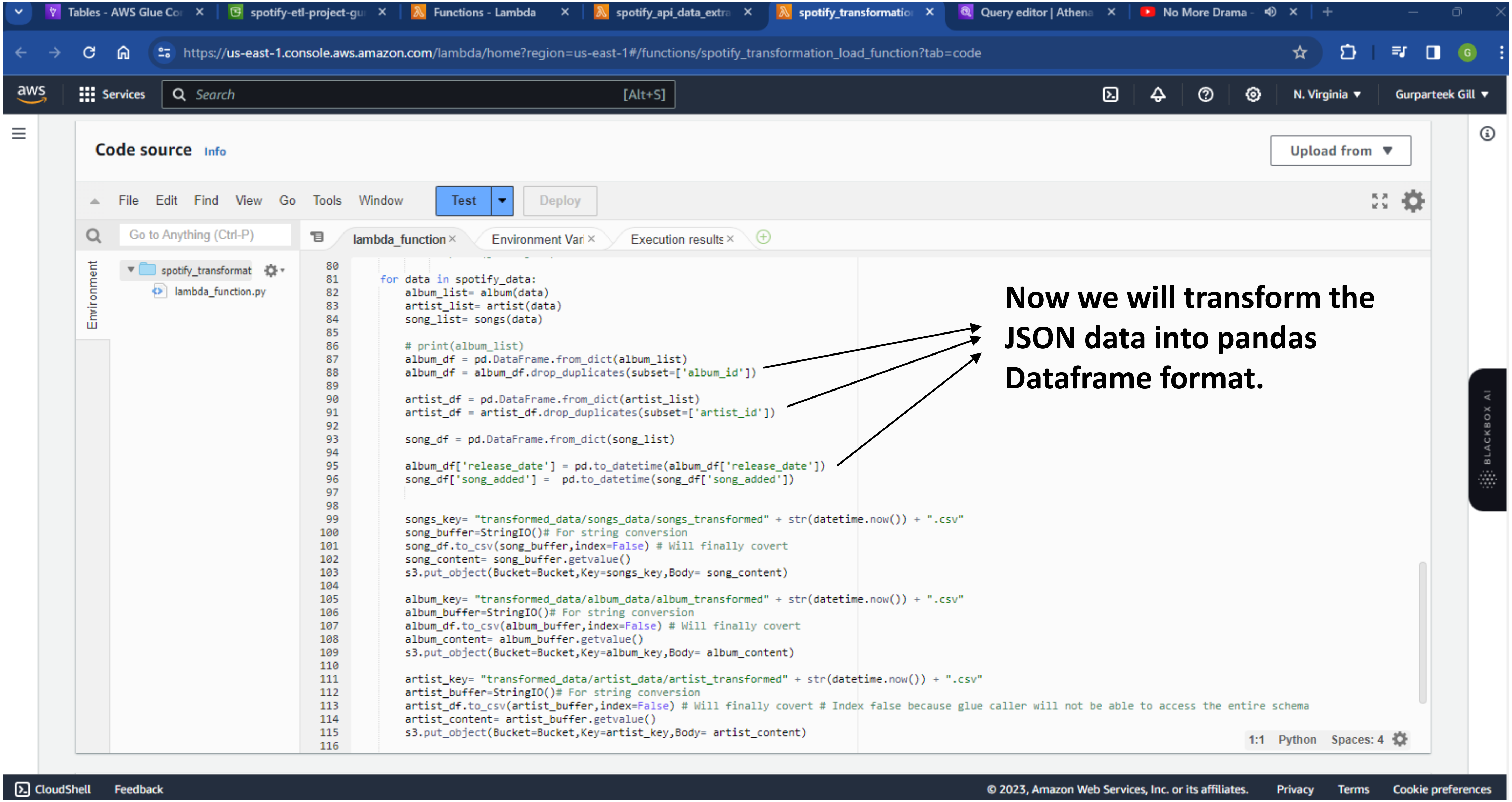


```python


def songs(data):
  song_list = []
  for row in data['items']:
    song_id = row['track']['id']
    song_name = row['track']['name']
    song_duration = row['track']['duration_ms']
    song_url = row['track']['external_urls']['spotify']
    song_popularity = row['track']['popularity']
    song_added = row['added_at']
    album_id = row['track']['album']['id']
    artist_id = row['track']['album']['artists'][0]['id']
    song_element = {'song_id':song_id,'song_name':song_name,'duration_ms':song_duration,'url':song_url,
                    'popularity':song_popularity,'song_added':song_added,'album_id':album_id,
                    'artist_id':artist_id
                   }
    song_list.append(song_element)

  return song_list




def lambda_handler(event, context):
    s3= boto3.client('s3') # Crearing the Object
    Bucket="spotify-etl-project-gurparteek"
    Key="raw_data/to_processed" # Path where the files are stored

    #This is a function
    #print(s3.list_objects(Bucket=Bucket,Prefix=Key)['Contents'])

    spotify_data=[]
    spotify_keys=[]

    # Here we are using Forloop because we have List and inside the list we
    # we have Mutiple Dictionaries
```



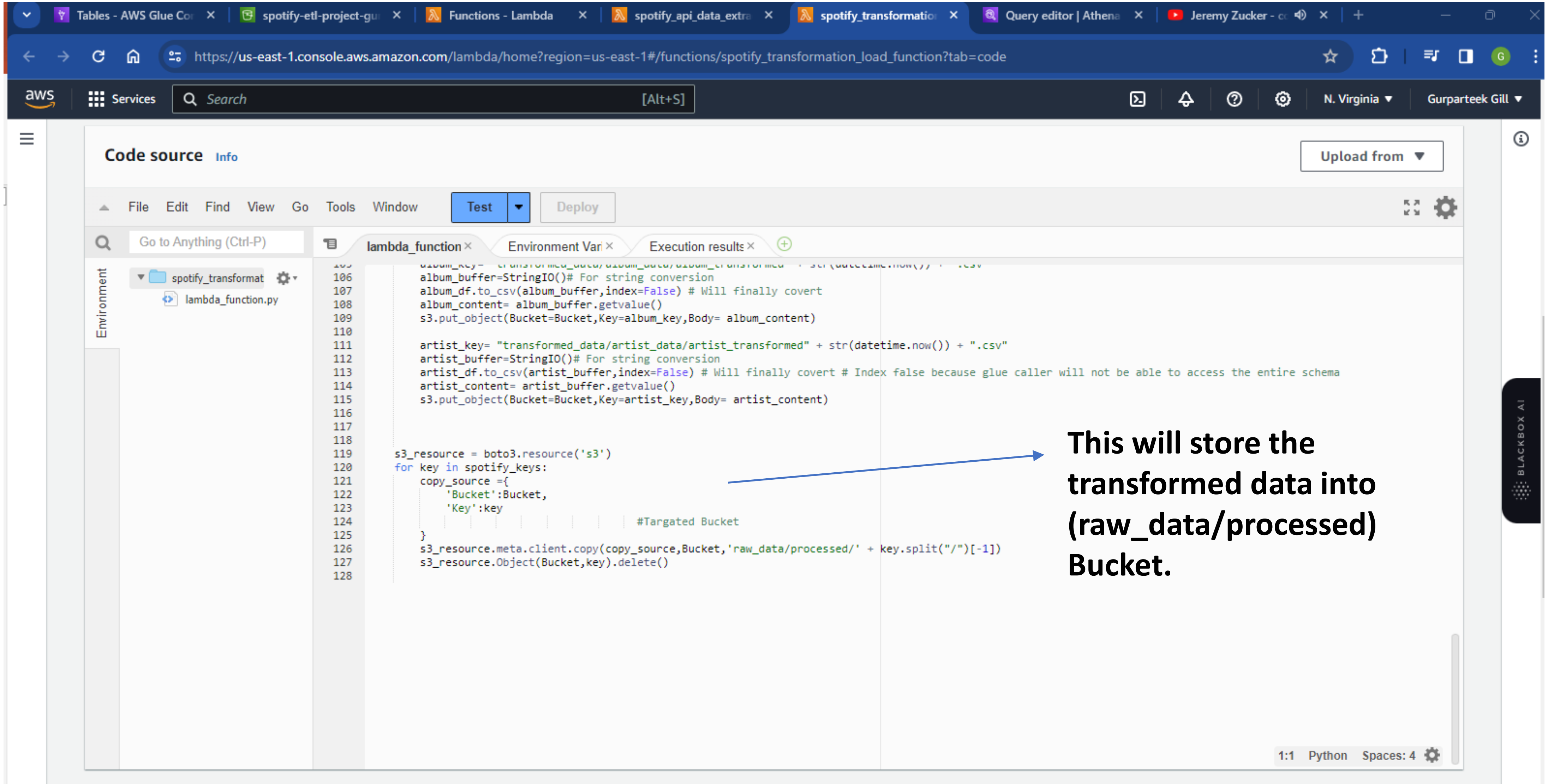
```python


def lambda_handler(event, context):
    s3= boto3.client('s3') # Crearing the Object
    Bucket="spotify-etl-project-gurparteek"
    Key="raw_data/to_processed" # Path where the files are stored

    #This is a function
    #print(s3.list_objects(Bucket=Bucket,Prefix=Key)['Contents'])

    spotify_data=[]
    spotify_keys=[]

    # Here we are using Forloop because we have List and inside the list we
    # we have Mutiple Dictionaries
    for file in s3.list_objects(Bucket=Bucket,Prefix=Key)['Contents']:
          #print(file['Key'])
          file_key=file['Key']
          if file_key.split('.')[-1] == "json": # Means it will pick only Json files

             # Inside the s3 object we have this function availbale
              response =s3.get_object(Bucket = Bucket,Key = file_key)
              content=response['Body']
              jsonObject= json.loads(content.read())
              spotify_data.append(jsonObject)
              spotify_keys.append(file_key)
             # print(jsonObject)

    for data in spotify_data:
        album_list= album(data)
        artist_list= artist(data)
        song_list= songs(data)

        # print(album_list)
        album_df = pd.DataFrame.from_dict(album_list)
        album_df = album_df.drop_duplicates(subset=['album_id'])
```

Now we will store the extracted data inside the (raw_data/to_processed) Bucket .



```python
    for data in spotify_data:
        album_list= album(data)
        artist_list= artist(data)
        song_list= songs(data)

        # print(album_list)
        album_df = pd.DataFrame.from_dict(album_list)
        album_df = album_df.drop_duplicates(subset=['album_id'])

        artist_df = pd.DataFrame.from_dict(artist_list)
        artist_df = artist_df.drop_duplicates(subset=['artist_id'])

        song_df = pd.DataFrame.from_dict(song_list)

        album_df['release_date'] = pd.to_datetime(album_df['release_date'])
        song_df['song_added'] =  pd.to_datetime(song_df['song_added'])


        songs_key= "transformed_data/songs_data/songs_transformed" + str(datetime.now()) + ".csv"
        song_buffer=StringIO()# For string conversion
        song_df.to_csv(song_buffer,index=False) # Will finally covert
        song_content= song_buffer.getvalue()
        s3.put_object(Bucket=Bucket,Key=songs_key,Body= song_content)

        album_key= "transformed_data/album_data/album_transformed" + str(datetime.now()) + ".csv"
        album_buffer=StringIO()# For string conversion
        album_df.to_csv(album_buffer,index=False) # Will finally covert
        album_content= album_buffer.getvalue()
        s3.put_object(Bucket=Bucket,Key=album_key,Body= album_content)

        artist_key= "transformed_data/artist_data/artist_transformed" + str(datetime.now()) + ".csv"
        artist_buffer=StringIO()# For string conversion
        artist_df.to_csv(artist_buffer,index=False) # Will finally covert # Index false because glue caller will not be able to access the entire schema
        artist_content= artist_buffer.getvalue()
        s3.put_object(Bucket=Bucket,Key=artist_key,Body= artist_content)
```

Now we will transform the JSON data into pandas Dataframe format.

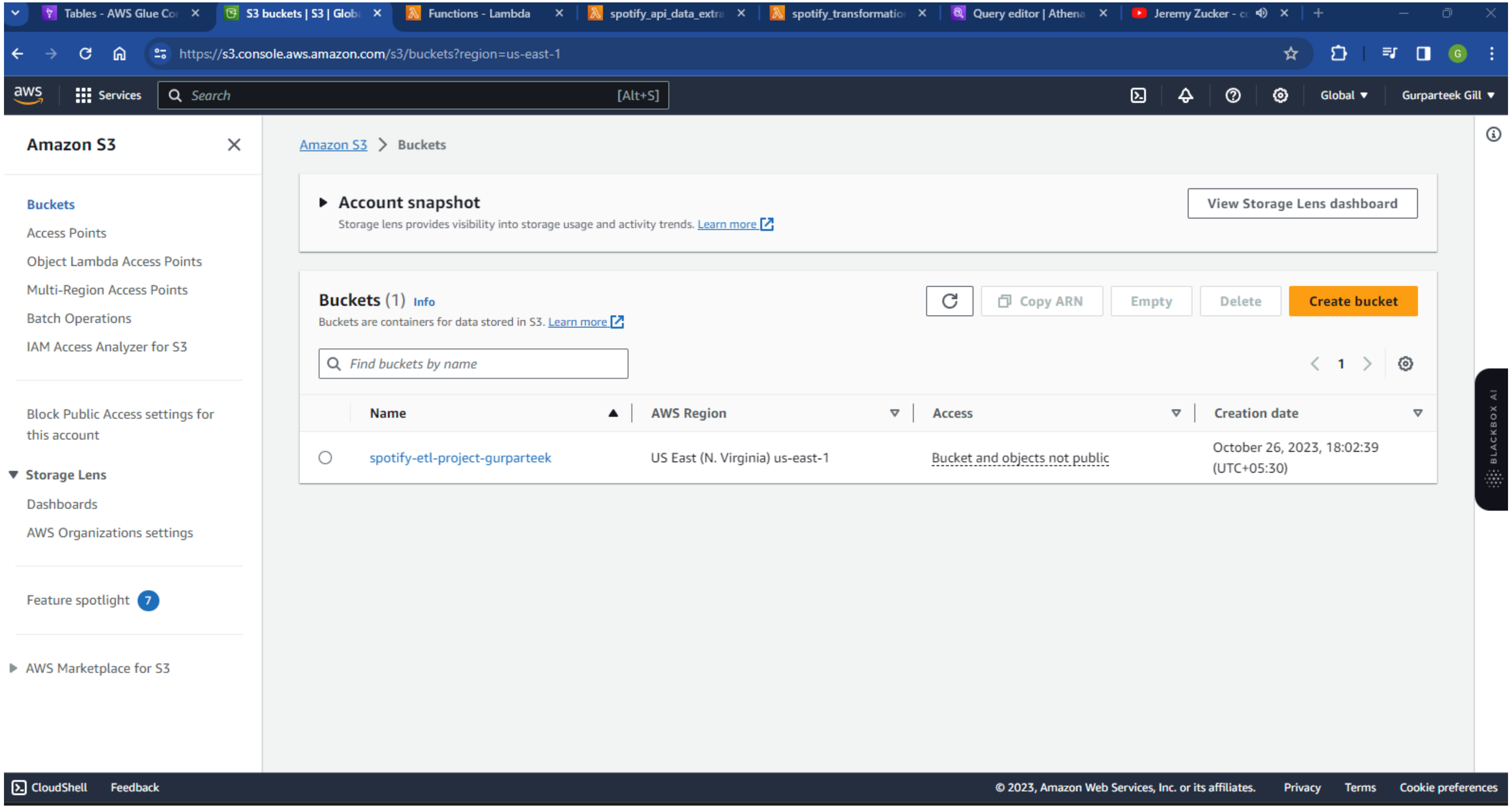


```python
        album_key= "transformed_data/album_data/album_transformed" + str(datetime.now()) + ".csv"
        album_buffer=StringIO()# For string conversion
        album_df.to_csv(album_buffer,index=False) # Will finally covert
        album_content= album_buffer.getvalue()
        s3.put_object(Bucket=Bucket,Key=album_key,Body= album_content)

        artist_key= "transformed_data/artist_data/artist_transformed" + str(datetime.now()) + ".csv"
        artist_buffer=StringIO()# For string conversion
        artist_df.to_csv(artist_buffer,index=False) # Will finally covert # Index false because glue caller will not be able to access the entire schema
        artist_content= artist_buffer.getvalue()
        s3.put_object(Bucket=Bucket,Key=artist_key,Body= artist_content)



    s3_resource = boto3.resource('s3')
    for key in spotify_keys:
        copy_source ={
            'Bucket':Bucket,
            'Key':key
                                        #Targated Bucket
        }
        s3_resource.meta.client.copy(copy_source,Bucket,'raw_data/processed/' + key.split("/")[-1])
        s3_resource.Object(Bucket,key).delete()

```

This will store the transformed data into (raw_data/processed) Bucket.

s3 Bucket

Amazon S3



▸ Account snapshot

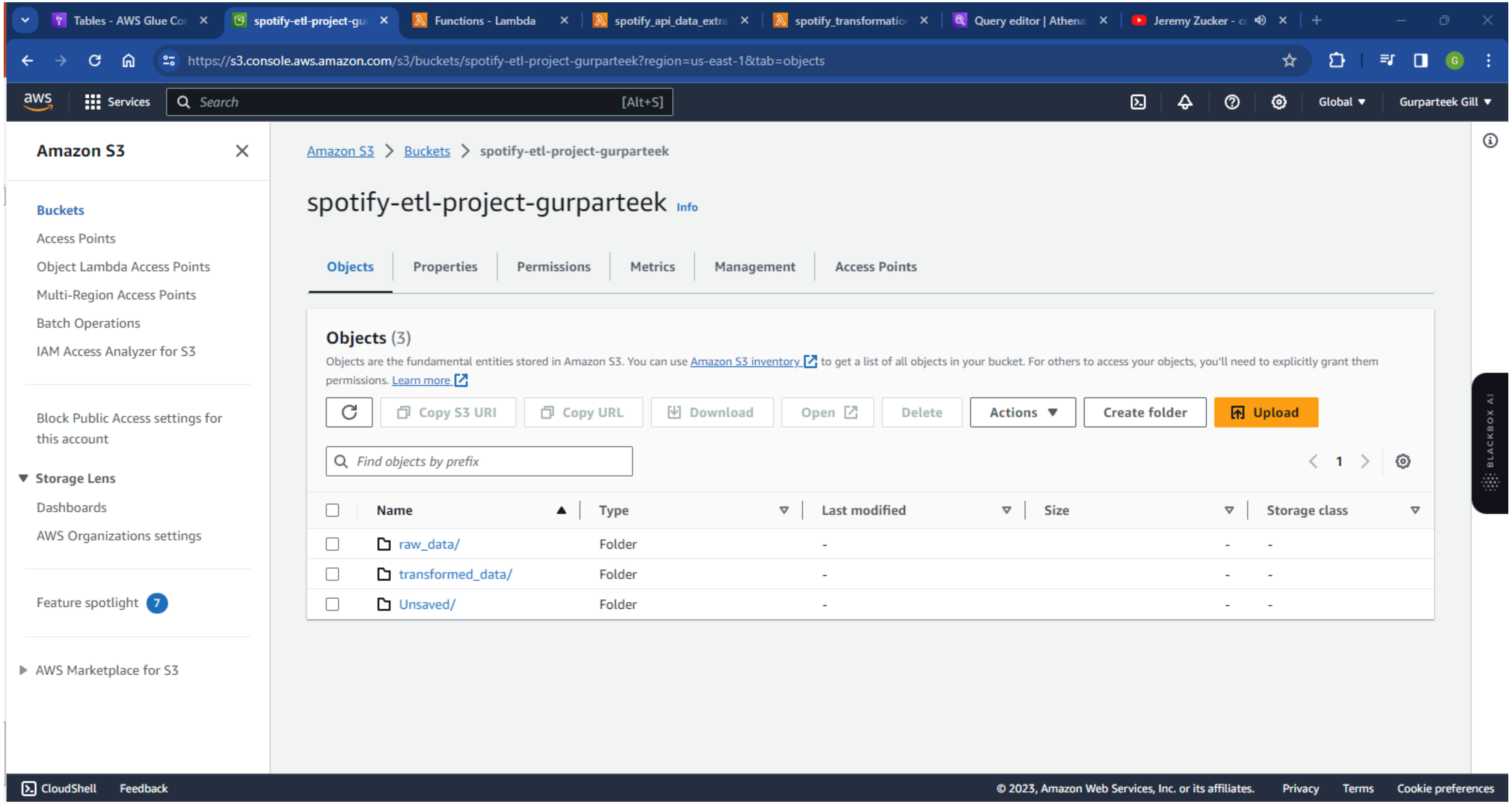
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	spotify-etl-project-gurparteek	US East (N. Virginia) us-east-1	Bucket and objects not public	October 26, 2023, 18:02:39 (UTC+05:30)

© 2023, Amazon Web Services, Inc. or its affiliates.

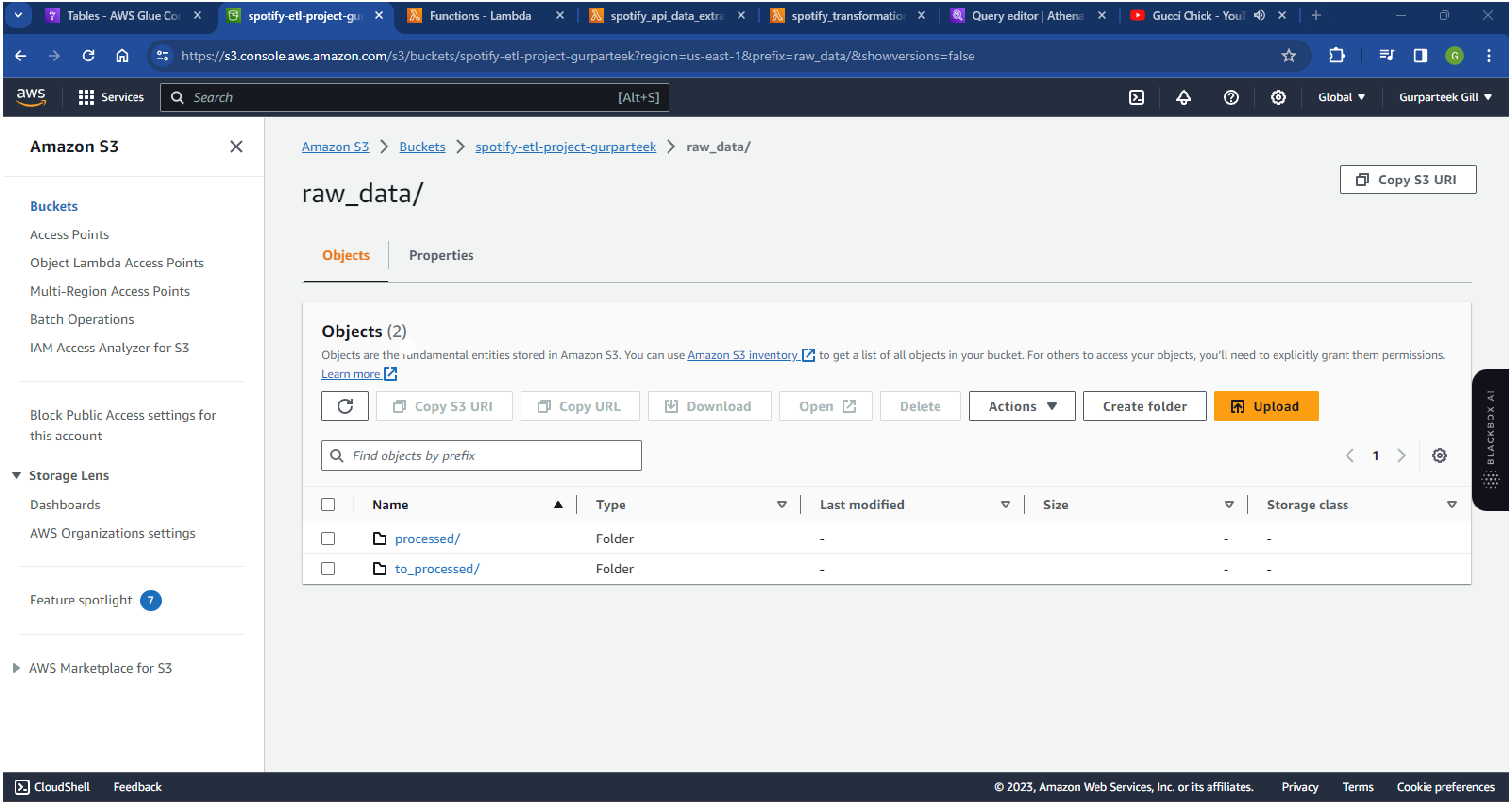


spotify-etl-project-gurparteek

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
□	raw_data/	Folder	-	-	-
□	transformed_data/	Folder	-	-	-
□	Unsaved/	Folder	-	-	-

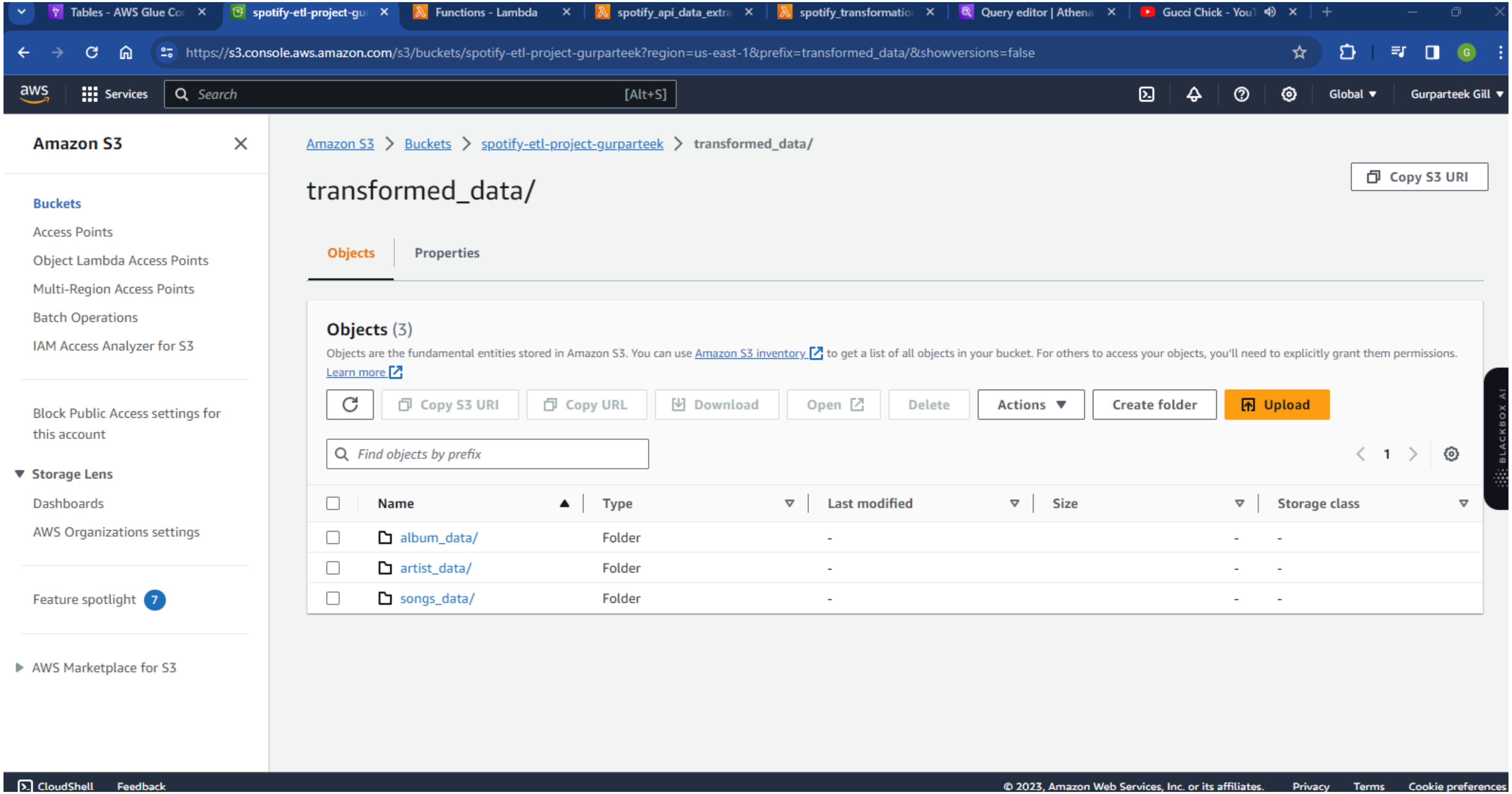


raw_data/

Objects (2)

Objects are the ₁undamental 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	-	-	-



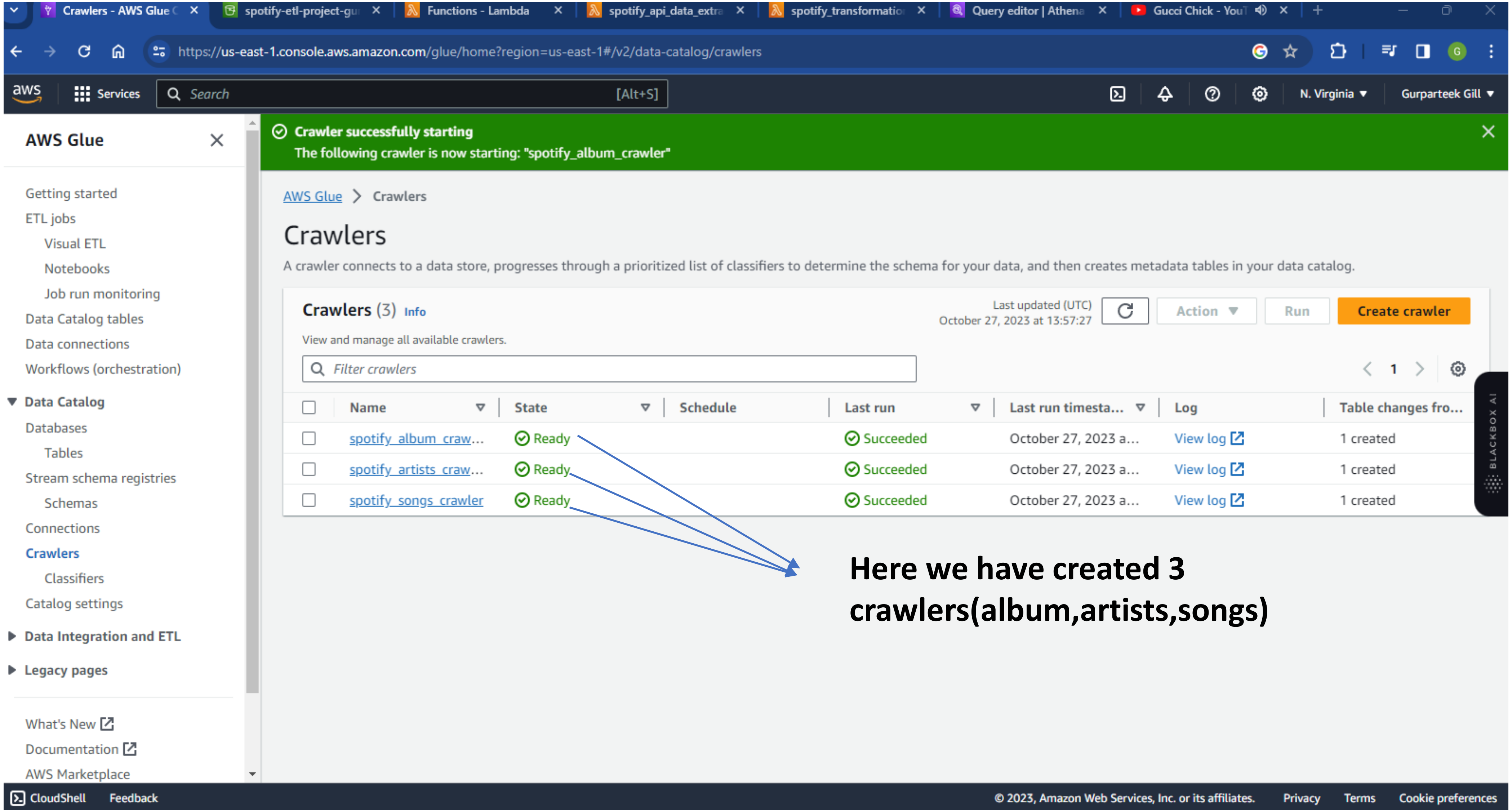
transformed_data/

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

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

Crawler



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

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)
October 27, 2023 at 13:57:27

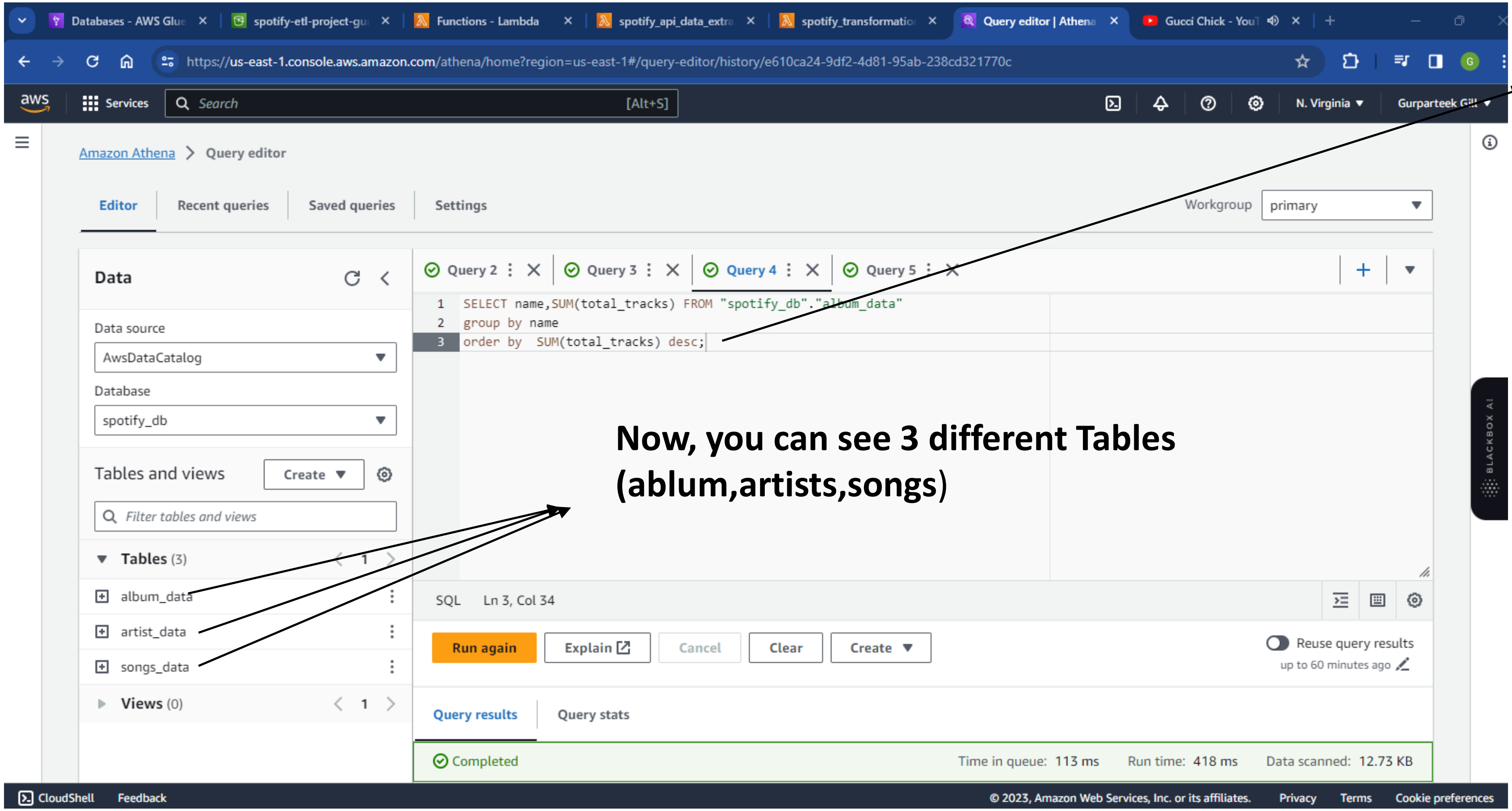
View and manage all available crawlers.

Filter crawlers

□	Name	State	Schedule	Last run	Last run timesta...	Log	Table changes fro...
□	spotify_album_craw...	✓ Ready		✓ Succeeded	October 27, 2023 a...	View log	1 created
□	spotify_artists_craw...	✓ Ready		✓ Succeeded	October 27, 2023 a...	View log	1 created
□	spotify_songs_crawler	✓ Ready		✓ Succeeded	October 27, 2023 a...	View log	1 created

Here we have created 3 crawlers(album,artists,songs)

Amazon Athena



Amazon Athena > Query editor

Data source: AwsDataCatalog

Database: spotify_db

Tables (3)

- album_data
- artist_data
- songs_data

Views (0)

Query 4

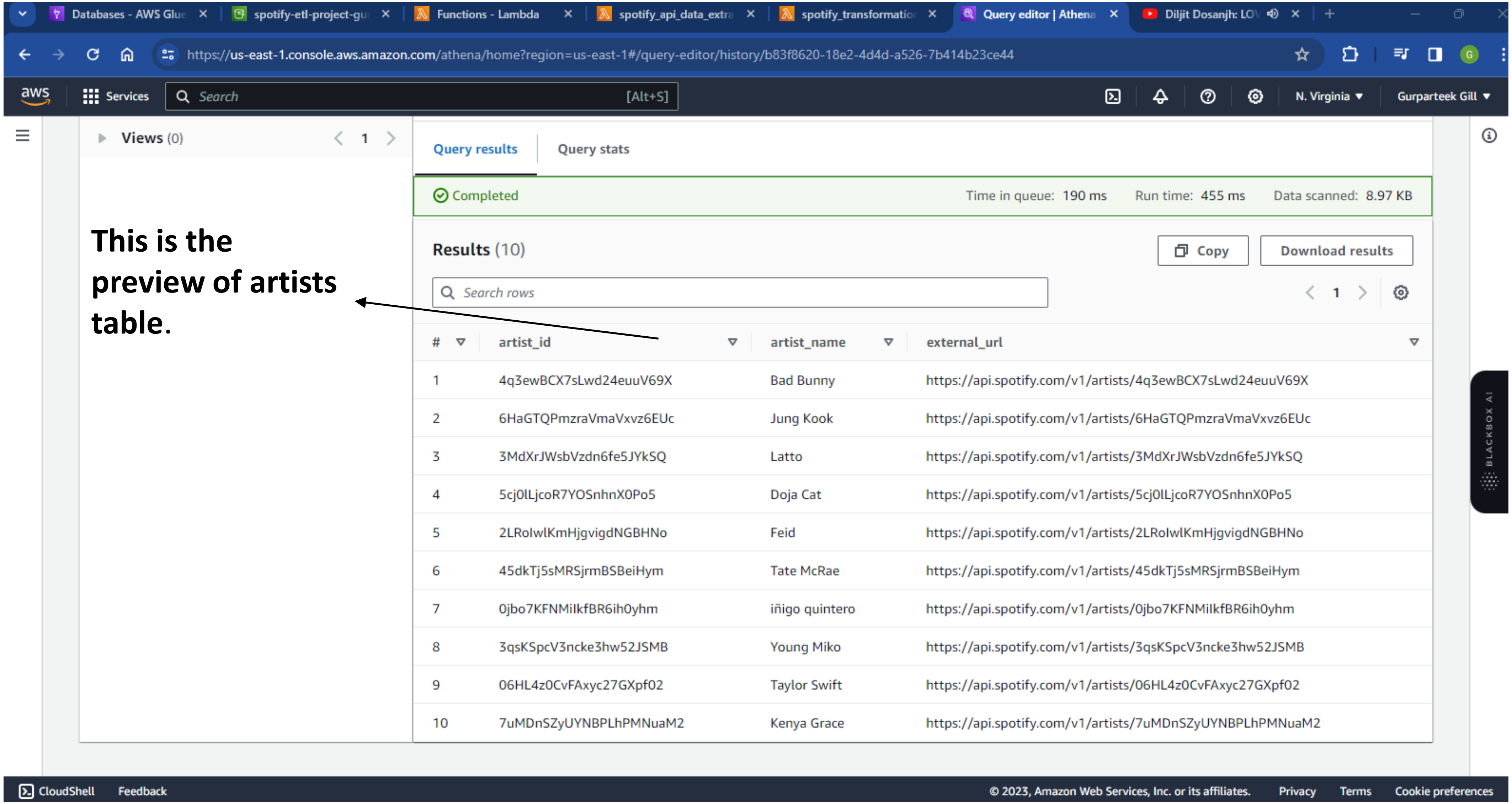
```
SELECT name,SUM(total_tracks) FROM "spotify_db"."album_data"
group by name
order by  SUM(total_tracks) desc;
```

SQL Ln 3, Col 34

Completed Time in queue: 113 ms Run time: 418 ms Data scanned: 12.73 KB

Here you can write SQL queries

Now, you can see 3 different Tables (ablum,artists,songs)



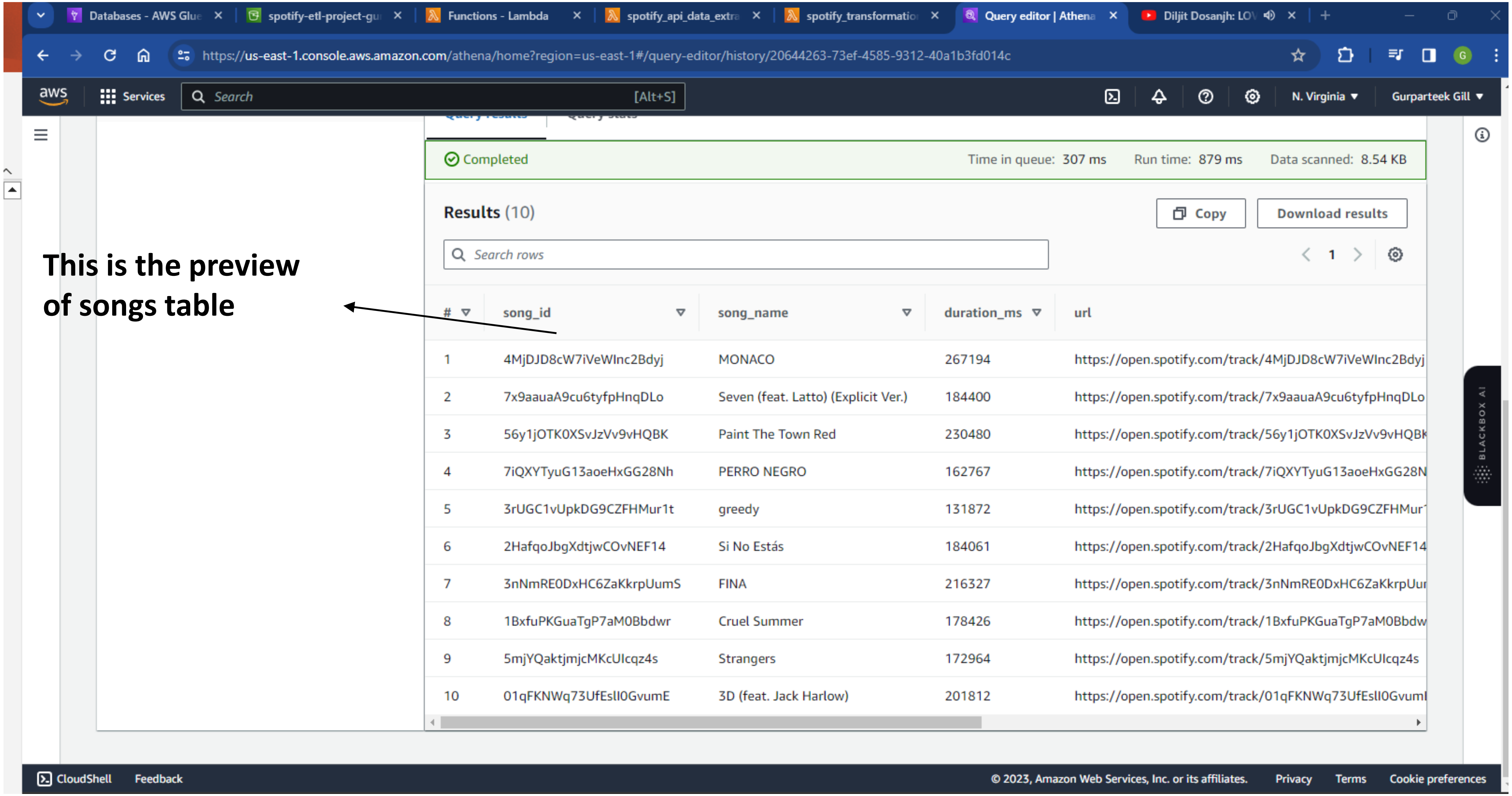
This is the preview of artists table.

Completed Time in queue: 190 ms Run time: 455 ms Data scanned: 8.97 KB

Results (10)

#	artist_id	artist_name	external_url
1	4q3ewBCX7sLwd24euuV69X	Bad Bunny	https://api.spotify.com/v1/artists/4q3ewBCX7sLwd24euuV69X
2	6HaGTQPmzraVmaVxvz6EUc	Jung Kook	https://api.spotify.com/v1/artists/6HaGTQPmzraVmaVxvz6EUc
3	3MdXrJWsbVzdn6fe5JYkSQ	Latto	https://api.spotify.com/v1/artists/3MdXrJWsbVzdn6fe5JYkSQ
4	5cj0lLjcoR7YOSnhnX0Po5	Doja Cat	https://api.spotify.com/v1/artists/5cj0lLjcoR7YOSnhnX0Po5
5	2LRoIwlKmHjgvigdNGBHNo	Feid	https://api.spotify.com/v1/artists/2LRoIwlKmHjgvigdNGBHNo
6	45dkTj5sMRSjrmBSBeiHym	Tate McRae	https://api.spotify.com/v1/artists/45dkTj5sMRSjrmBSBeiHym
7	0jbo7KFNMiIkfBR6ih0yhm	iñigo quintero	https://api.spotify.com/v1/artists/0jbo7KFNMiIkfBR6ih0yhm
8	3qsKSpcV3ncke3hw52JSMB	Young Miko	https://api.spotify.com/v1/artists/3qsKSpcV3ncke3hw52JSMB
9	06HL4z0CvFAxyc27GXpf02	Taylor Swift	https://api.spotify.com/v1/artists/06HL4z0CvFAxyc27GXpf02
10	7uMDnSZyUYNBPLhPMNuaM2	Kenya Grace	https://api.spotify.com/v1/artists/7uMDnSZyUYNBPLhPMNuaM2

This is the preview of songs table



Completed Time in queue: 307 ms Run time: 879 ms Data scanned: 8.54 KB

Results (10)

Copy Download results

Search rows

#	song_id	song_name	duration_ms	url
1	4MjDJD8cW7iVeWlnc2Bdyj	MONACO	267194	https://open.spotify.com/track/4MjDJD8cW7iVeWlnc2Bdyj
2	7x9aauaA9cu6tyfpHnqDLo	Seven (feat. Latto) (Explicit Ver.)	184400	https://open.spotify.com/track/7x9aauaA9cu6tyfpHnqDLo
3	56y1jOTK0XSvJzVv9vHQBK	Paint The Town Red	230480	https://open.spotify.com/track/56y1jOTK0XSvJzVv9vHQBK
4	7iQXYTyuG13aoeHxGG28Nh	PERRO NEGRO	162767	https://open.spotify.com/track/7iQXYTyuG13aoeHxGG28N
5	3rUGC1vUpkDG9CZFHMur1t	greedy	131872	https://open.spotify.com/track/3rUGC1vUpkDG9CZFHMur
6	2HafqoJbgXdtjwCOvNEF14	Si No Estás	184061	https://open.spotify.com/track/2HafqoJbgXdtjwCOvNEF14
7	3nNmRE0DxHC6ZaKkrpUumS	FINA	216327	https://open.spotify.com/track/3nNmRE0DxHC6ZaKkrpUu
8	1BxfuPKGuaTgP7aM0Bbdwr	Cruel Summer	178426	https://open.spotify.com/track/1BxfuPKGuaTgP7aM0Bbdw
9	5mjYQaktjmjcMKcUIcqz4s	Strangers	172964	https://open.spotify.com/track/5mjYQaktjmjcMKcUIcqz4s
10	01qFKNWq73UfEslI0GvumE	3D (feat. Jack Harlow)	201812	https://open.spotify.com/track/01qFKNWq73UfEslI0Gvum