Spotify Data Pipeline End-To-End Python Data Engineering Project

Project Description:

In this project, we aim to create an ETL (Extract, Transform, Load) pipeline by leveraging the Spotify API on the AWS cloud platform. The pipeline's primary purpose is to retrieve data from the Spotify API concerning the top 50 global songs, transform this data into a desired format, and subsequently store it in an AWS data repository.

Pipeline Overview:

The project can be divided into three primary phases:

📌Extract:

- The first step involves extracting data from the Spotify API, which is executed using Python, with the help of Spotipy library.
- AWS Lambda is used to deploy the extraction function.
- Automation is achieved using AWS CloudWatch to trigger the extraction function at regular intervals.
- The raw extracted data is stored in an AWS S3 Bucket within the "to_processed" folder.

📌Transform:

- The transformation begins by setting up S3 triggers to activate a second lambda function whenever new data is deposited in the S3 Bucket.
- Data transformation is performed using another AWS Lambda function.

Spotify Data Pipeline End-To-End Python Data Engineering Project

- The transformed data is stored in the S3 Bucket, distributed across different folders categorizing it by album, artist, and song. It also involves moving data from the "to_processed" folder to the "processed" folder.

📌Load:

- For schema inference, Glue Crawler is usd whenever new data is added to the S3 Bucket.
- An AWS Glue Data Catalog is established to manage the Metadata Repository.
- The final dataset is made queryable and analyzable using Amazon Athena, enabling users to obtain desired information from the data.

Implement Complete Data Pipeline Data Engineering Project using Spotify

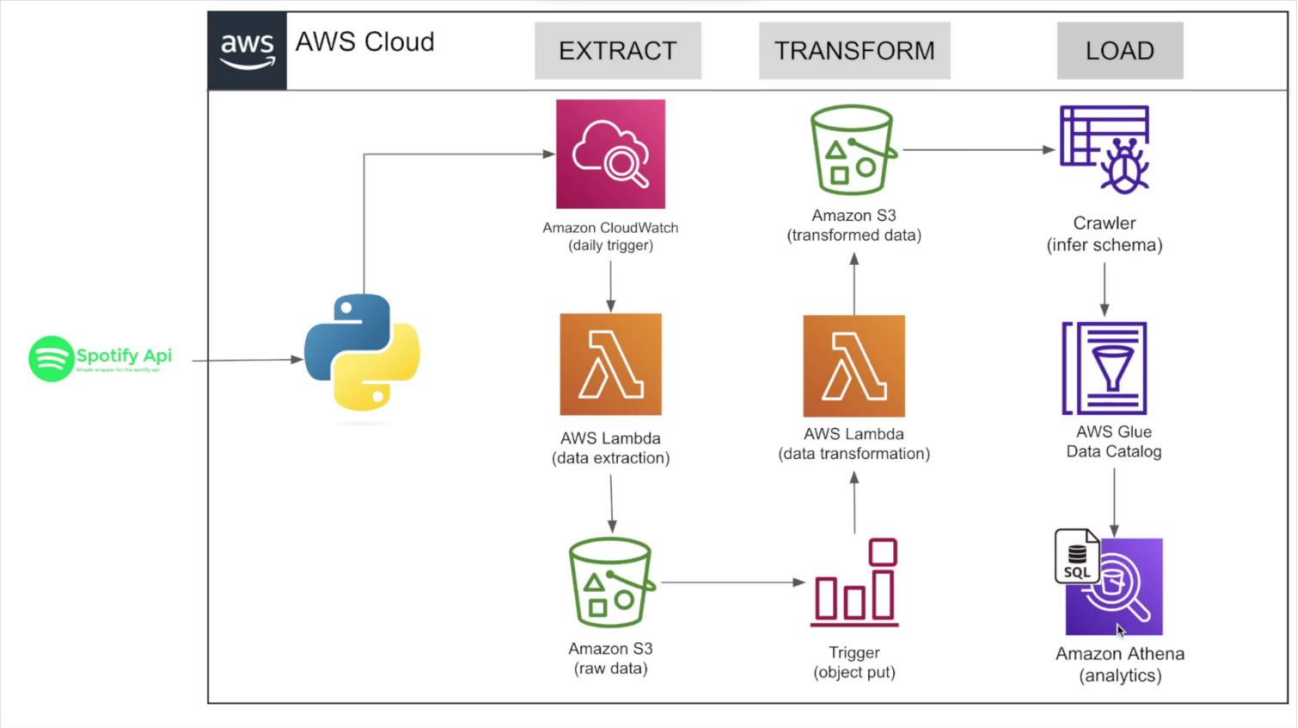
- Integrating with Spotify API and extracting Data
- Deploying code on AWS Lambda for Data Extraction
- Adding trigger to run the extraction automatically
- Writing transformation function
- Building automated trigger on transformation function
- Store files on S3 properly
- Building Analytics Tables on data files using Glue and Athena

Objective

- Designing a fully-automated ETL pipeline on AWS to extract Spotify's weekly Top Songs - Global playlist and store the Artist, Album and Song data on AWS Glue Data Catalog tables.

Spotify Data Pipeline End-To-End Python Data Engineering Project

Solution Architecture



Implementation of Services

Spotify API

One would need developer app credentials (Client ID, Client Secret) to transact with the Spotify Web API.

Spotify Data Pipeline End-To-End Python Data Engineering Project

AWS Services Used in this Project

AWS Lambda - Data Extraction

Create the Lambda Function for Data Extraction spotify_api_data_extract which runs on an appropriate Python runtime (eg. Python 3.8+). This function will be executed according to the schedule setup by the EventBridge Rule. Add the data extraction code to the created lambda_function.py. Ensure that the spotipy package is available in connected Lambda Layer for it to be accessed in the function environment.

Amazon S3 - for Raw and Transformed data

Create an S3 bucket spotify-etl-project-demo which will manage both raw data (from the Spotify API) as well as the processed structured data to be used downstream. Create two primary folders: raw_data/ and transformed_data/. raw_data/: will contain subfolders - processed/, to_processed/ which will contain and manage the raw playlist data. transformed_data/: will contain subfolders - artist_data/, album_data/, song_data/ which will contain and manage the artist, album and song structured data.

Amazon EventBridge

Create new EventBridge Rule to run at a given schedule (eg. every 1 day, week). Set the trigger's target to be the Lambda function for Data Extraction from the Spotify API.

AWS Lambda - Data Transformation

Create the Lambda Function for Data Transformation spotify_transformation_load_function which runs on an appropriate Python runtime (eg. Python 3.8+). This function will be triggered whenever a new file is dropped into

Spotify Data Pipeline End-To-End Python Data Engineering Project

the raw_data/to_processed/ folder of the spotify-etl-project-demo S3 bucket. The code will extract the artist, album and song data and save as structured data in the transformed folder, as well as manage the raw data by moving it into the processed folder. Add the data transformation code to the created lambda_function.py. Ensure that the AWSSDKPandas-Python38 AWS Lambda Layer is added to the function for the pandas package to be accessed in the function environment.

AWS Glue Crawler - infer schema & Amazon Athena - analytics

Create respective Glue Crawlers to infer schema of transformed artist, album and song data and to store their metadata in Glue Data Catalog. Set the respective S3 data folders as the data source for the crawlers; also specify the schema metadata like headers, column datatypes by creating the corresponding Glue Classifiers. This will allow the crawler to correctly parse the files in the data source. The output could be a specific database in the Glue Data Catalog. Running the created Crawlers could be based on a specific schedule or even manually (eg. a day after new data arrives). The Data Catalog database tables can be viewed and queried on Amazon Athena.