The code for this project is a lambda function written in Python that can be executed in the AWS Lambda service.

It starts by importing the necessary libraries like JSON, os, Spotify, boto3, datetime, and SpotipyClientCredentials from spotipy.oauth2.

A screenshot of a computer program

Description automatically generated

Fig 1: Code snippet for the data extraction from Spotify API

1. The entry point for the Lambda function, which is triggered by an event, is the lambda\_handler function.
2. Using os.environ.get, the client\_id and client\_secret credentials for the Spotify API are obtained from environment variables (Spotipy, n.d.).
3. Using the credentials collected in the previous step, a SpotifyClientCredentials instance is created.
4. The client credentials manager is used to create a Spotify client (sp).
5. Using the user\_playlists method of the Spotify client, the code retrieves the playlists belonging to the user "spotify".
6. The link to a Spotify playlist is stored in the playlist\_link variable.
7. By splitting the playlist link and extracting the last element, the playlist URI can be extracted from the playlist link.
8. The Spotify client's playlist\_tracks method retrieves the playlist's tracks by supplying the playlist URI (Spotipy, n.d.).
9. The code uses boto3.client('s3') to initialize the AWS S3 client.
10. A filename is generated using the current timestamp to ensure that the S3 object is unique.
11. The S3 client's put\_object method is used to upload the data as a JSON file to the specified S3 bucket and key (AWS, n.d.).

* The Bucket parameter specifies the name of the S3 bucket.
* The Key parameter specifies the key (path) where the file will be stored within the bucket.
* The Body parameter contains the JSON data, which is serialized using json.dumps.

This is where the Lambda function execution stops.

This script downloads a Spotify playlist, gets the tracks from it, and saves the track information as a JSON file in an S3 bucket. The code has hardcoded values for the S3 bucket name and file path.

A computer screen shot of a music album

Description automatically generated

A screen shot of a computer code

Description automatically generated

Fig 2: Code snippet for data transformation.

1. JSON, boto3 (AWS SDK for Python), datetime, StringIO, and pandas are significant libraries and modules.
2. The functions of the album, artist, and song are specified. From Spotify data, these functions extract relevant data and list it for storage.

A screenshot of a computer program

Description automatically generated

Fig 3: Code snippet for converting data from JSON to a pandas data frame.

1. The entry point for the Lambda function is defined as the lambda\_handler function. The boto3 client is used to establish a connection to an S3 bucket and retrieve a list of files located in the requested directory.
2. The function loads the contents of each JSON file in the S3 bucket, extracts the Spotify data, and adds it to the spotify\_data list.
3. The album, artist, and song functions are called to extract album, artist, and song information for each Spotify data object in spotify\_data. album\_df, artist\_df, and song\_df pandas DataFrames are then made using the retrieved data.

A computer screen shot of a program

Description automatically generated

A close-up of a computer code

Description automatically generated

Fig 4: Creating a dataframe in pandas.

1. By eliminating duplicates and changing the date and time columns to the proper data types, the DataFrames are changed.
2. The updated timestamp is then included in the file name and the modified data is then saved as CSV files to the S3 bucket.
3. The original JSON files in the S3 bucket are relocated to a different location for archiving or deletion after the transformation and storage processes are finished.

A screenshot of a computer

Description automatically generated

Fig 6: The extracted raw data in JSON format.

A screenshot of a computer

Description automatically generated

Fig 7: Transformed raw data into CSV format with time stamp.