Workshop-002: ETL process using airflow

 Sebastian Belalcazar Mosquera

https://github.com/SEBASBELMOS/workshop-002

Overview

This project simulates a real-world Data Engineering task, focusing on building an **ETL pipeline** using **Apache Airflow**. The objective is to:

- **Extract** data from three sources: the Spotify API, a CSV file (Spotify dataset), and a database (Grammys dataset).
- **Transform** and merge the data using Apache Airflow.
- **Load** the processed data into a database and Google Drive as a CSV.
- Create a **dashboard** using Power BI to display insights from the database.

Dataset Information

1. **Spotify Dataset** (*spotify_dataset.csv*)

- Source: CSV file
- Description: Contains metadata and audio features of Spotify tracks.
- Link: Spotify Tracks Dataset on Kaggle

  Key Columns:

  - track_id: Unique track identifier.

- artists: Artist(s) name(s).
- track_name: Song title.
- popularity: Score (0-100) of track popularity.
- duration_ms: Track length in milliseconds.
- danceability: Suitability for dancing (0-1).
- energy: Intensity measure (0-1).
- loudness: Loudness in decibels.
- tempo: Beats per minute (BPM).
- track_genre: Genre of the track.

1. **Grammy Awards Dataset** (*the_grammy_awards.csv*)

- Source: Initial database load
- Description: Details Grammy nominations and wins from 1958 to 2019.
- Link: Grammy Awards on Kaggle

  Key Columns:

  - year: Year of the Grammy event.
  - category: Award category (e.g., Best Pop Solo Performance).
  - nominee: Nominated song or album.
  - artist: Associated artist(s).
  - winner: Boolean indicating win (True/False).

1. **Spotify API**

- Source: API
- Description: In order to improve the analysis, the key column will be `followers` of each artist.
- Link: Spotify for Developers

  Key Column:

  - followers: Artist Followers.

Project Structure

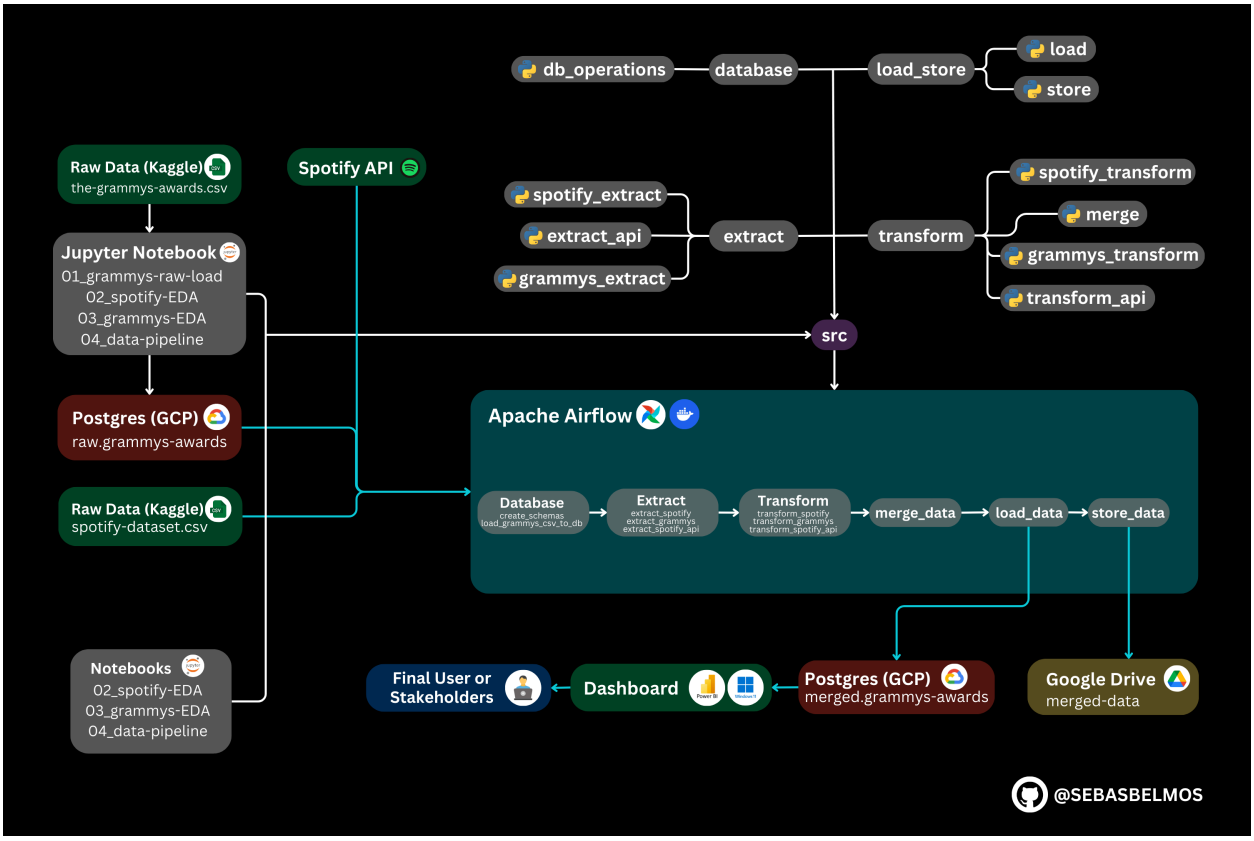
Folder/File	Description

assets/	Static resources (images, documentation, etc.)
data/	Dataset used in the project (ignored in .gitignore)
dags/	Stores Apache Airflow Dags
├── **tasks/**	Stores Apache Airflow Tasks that will be used by our dag
docs/	Documentation, Guides and workshop PDFs
drive_config/	Stores the client secret from OAuth and saved credentials to store the merge data
notebooks/	Jupyter Notebooks with analysis
├── 01_grammys-raw-load.ipynb	Loads Grammys data into PostgreSQL
├── 02_spotify-EDA.ipynb	Exploratory Data Analysis of Spotify dataset
├── 03_grammys-EDA.ipynb	Exploratory Data Analysis of Grammys dataset
├── 04_data-pipeline.ipynb	ETL pipeline execution and Google Drive credentials creation
src/	Python scripts for Airflow tasks and utilities
venv/	Virtual environment (ignored in .gitignore)
env/	Environment variables (ignored in .gitignore)
├── .env	Stores credentials and paths
pbi/	Power BI files (ignored in .gitignore)
docker-compose.yaml	Docker configuration
requirements.txt	All the libraries required to execute this project properly
README.md	This file

Tools and Libraries

- **Programming Language:** Python 3.13 → Download here
- **Workflow Orchestration**: Apache Airflow → Documentation here
- **Data Handling:** pandas → Documentation here
- **Database:** PostgreSQL → Download here
- **Database Interaction:** SQLAlchemy → Documentation here
- **Visualisation:** Power BI Desktop → Download here
- **Environment:** Jupyter Notebook → VSCode tool used
- **Storage:** Google Drive & PyDrive2 → Documentation here

Workflow



Installation and Setup

1. **Clone the Repository:**

```
git clone <https://github.com/SEBASBELMOS/workshop-002.git>
cd workshop-002
```

2. **Database Google Cloud Platform.**

> To create the databases in GCP, you can follow this guide.

- Use the `public IP` for connections, and ensure the IP `0.0.0.0/0` is added to authorised networks for testing.

3. **Google Drive Auth.**

> To create the Google Drive Authentication, you can follow this guide.

- Ensure `PyDrive2` is installed as part of the dependencies (requirements.txt).

4. **Configure PyDrive2 (*settings.yaml*).**

> To configure the settings.yaml file for authentication and authorisation, follow this guide

5. **Virtual Environment (This must be done in Ubuntu or WSL).**
   - Create virtual environment.

     ```
     python3 -m venv workshop2
     ```

   - Activate it using this command:

     ```
     source workshop2/bin/activate
     ```

   - Install all the requirements and libraries with this command:

     ```
     pip install -r requirements.txt
     ```

6. **Enviromental variables**

> Realise this in VS Code.

To establish a connection with the database, we use a module called *connection.py*. This Python script retrieves a file containing our environment variables. Here's how to create it:

1. Inside the cloned repository, create a new directory named *env/*.
2. Within that directory, create a file called *.env*.
3. In the *.env file*, define the following six environment variables (without double quotes around values):

   ```
   #PostgreSQL Variables
   PG_HOST = #host address, e.g. localhost or 127.0.0.1
   PG_PORT = #PostgreSQL port, e.g. 5432
   ```

```
PG_USER = #your PostgreSQL user
PG_PASSWORD = #your user password
PG_DATABASE = #your database name, e.g. postgres

#Google Drive Variables
#Path to the client secrets file used for Google Drive authentication.
CLIENT_SECRETS_PATH = "/path/to/your/credentials/client_secrets.json"

#Path to the settings file for the application configuration.
SETTINGS_PATH = "/path/to/your/env/settings.yaml"

#Path to the file where Google Drive saved credentials are stored.
SAVED_CREDENTIALS_PATH = "/path/to/your/credentials/saved_credentials.jso
n"

#The ID of your Google Drive folder. This can be found in the link in your folder.
FOLDER_ID = # your-drive-folder-id

SPOTIFY_CLIENT_ID=client_id

SPOTIFY_CLIENT_SECRET=client_secret
```

7. **Execute Docker Containers**

   After generating your own `docker-compose.yaml`, you must run these commands in your terminal in order to execute your containers:

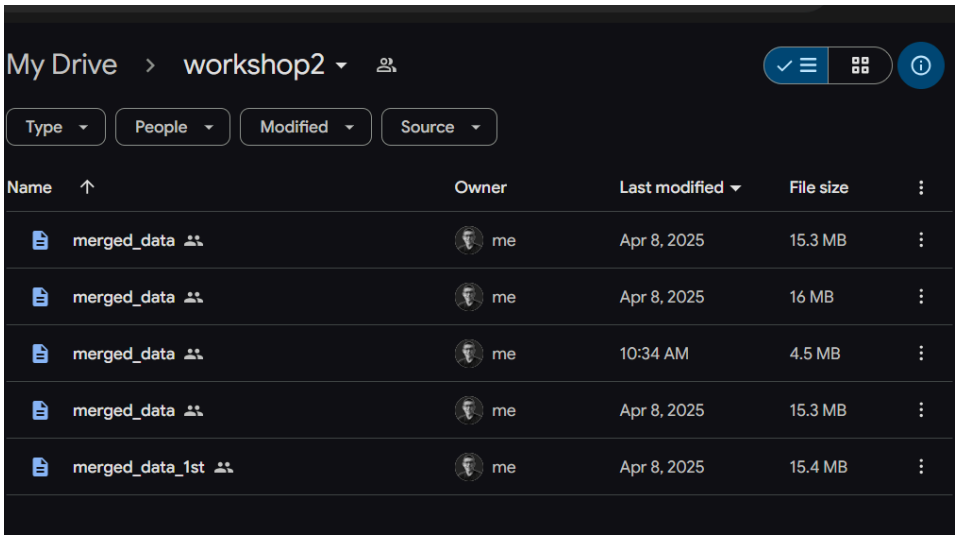
   - `docker-compose build` to create them.
   - `docker-compose up -d` to execute them.
   - `docker-compose down` to turn them off.

8. **Create the connection to the Spotify API**

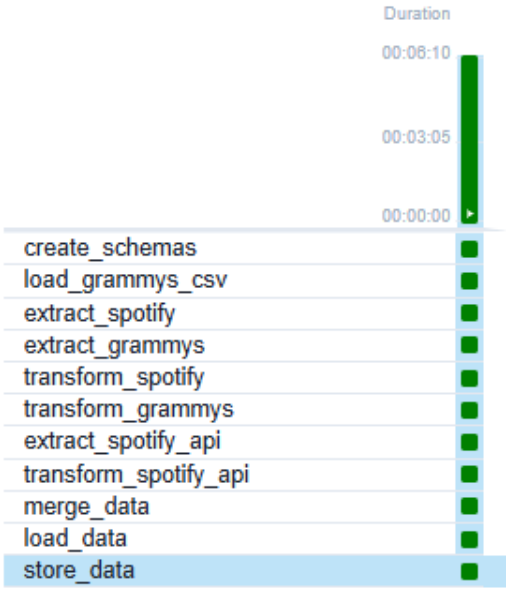
   > To generate the client ID and client secret, follow this guide.

9. **Dag Validation**

   After running the dag, you will see this file in your Google Drive Folder:

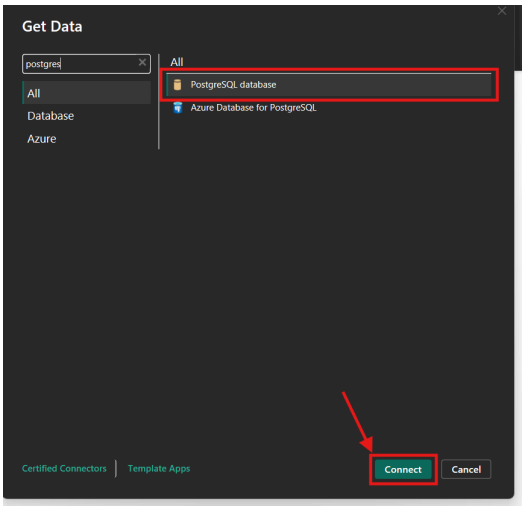


And you will see this in your Airflow Home:

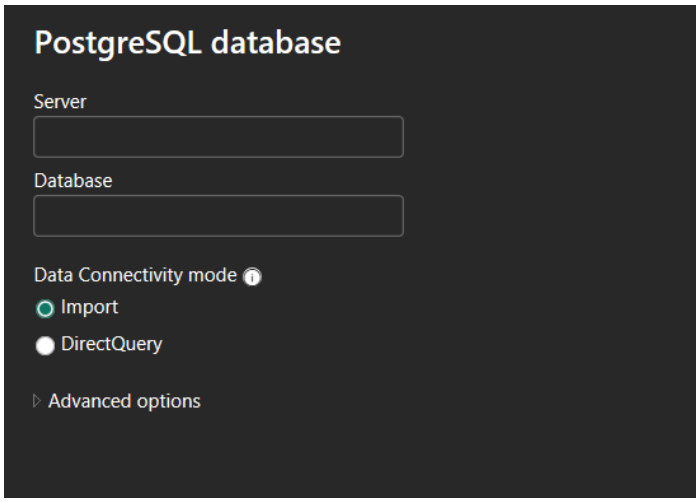


Power BI Connection (Not necessary)

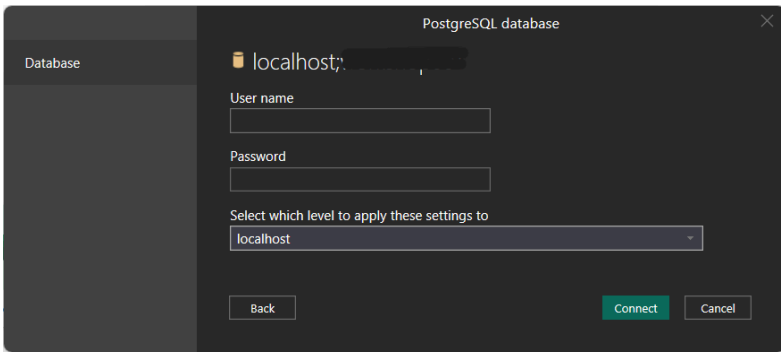
1. Open Power BI Desktop and create a new dashboard.
2. Select the *Get data* option, then choose the "*PostgreSQL Database*" option.



3. Insert the *PostgreSQL Server* and *Database Name*.



4. Fill the following fields with your Postgres credentials.



5. After establishing the connection, the tables of your db will be displayed. You need to select the ones you need, and then start creating your own dashboards.

- Open my Power BI Visualisation here

  dashboard.pdf

Author

Created by **Sebastian Belalcazar Mosquera**. Connect with me on LinkedIn for feedback, suggestions, or collaboration opportunities!