Airflow-ETL-Project

An ETL pipeline built with Apache Airflow, Docker and PostgreSQL to automate COVID-19 data ingestion, transformation, and loading into a database.

Overview

This project demonstrates an automated ETL pipeline using Apache Airflow. It gets real-time COVID-19 data from a public API, transforms it, and loads it into a PostgreSQL database running in a Docker container.

Tech Tools

- Workflow Orchestration: Apache Airflow
- Containerisation: Docker & Docker Compose
- Database: PostgreSQL
- Language: Python
- Source API: disease.sh - COVID-19 API
- Version Control: Git & GitHub

Pipeline Architecture

COVID 19 API -> Airflow DAG (Extract, Transform and Load) -> PostgreSQL Database

How to Setup

1. Firstly, clone the repository

git clone https://github.com/Guney10/Airflow-ETL-Project.git

cd Airflow-ETL-Project

2. Start Docker containers

docker compose up -d

3. Airflow UI Access

Open the browser and go to http://localhost:8080 or you can click on 8080:8080 on Docker Desktop inside your container

Login details

- Username : admin
- Password : admin

How this all works

The Airflow DAG performs:

1. **Extract** : Which fetches data from the COVID-19 API and stores it as a CSV file
2. **Transform** : Filters columns, handles missing values accordingly, calculates fatality rate
3. **Load** : Insters the cleaned data into the PostgresSQL database DAG also has a @daily schedule which can be triggered manually and run daily.

Outputs

covid_data.csv - The raw data which is extracted

covid_data_transformed.csv - Cleaned and transformed data

PostgresSQL table - covid_stats

The sample data can be queried using SQL inside the PostgresSQL container

SQL Queries

This is all inside the container so once you run this in the terminal:

docker compose exec postgres psql -U airflow -d airflow

Then you can run example queries such as :

SELECT country, cases FROM covid_stats ORDER BY cases DESC LIMIT 5 - This will list the top 5 countries by cases

SELECT AVG(fatality_rate) FROM covid_stats - Calculates the average fatality rate

Project Management

Project tasks were tracked using Trello.

Tasks include:

- Setting up Docker + Airflow
- Creating DAG tasks (extract, transform, load)
- Connecting to PostgreSQL
- Writing README and documentation
- Recording a demo

Contribution

If you would like to contribute to this repo:

- Fork the repository
- Create a new branch by using this code (git checkout -b example/your-example)
- Commit any changes by (git commit -m "Example")

- Push branch (git push origin branch)
- Then open a Pull Request