TECHNICAL REPORT

Gans Data Engineering Project: Pipeline in the Cloud

By
Goodness Udochukwu, Okoh

Table of Content

Introduction

Given the desire for sustainable eco-friendly based mobility, simplicity and accessibility to mobility, Gans has decided to provide a solution and access the market via an e-scooter sharing system, where users of the service rent the e-scooters.

The choice of what cities in Europe to provide this service remains an open and important decision. Also, decisions about how accessible the e-scooters are to users due to the distribution of the e-scooters by usage are a thing of concern.

This project seeks to answer some of these questions by providing data which supports the choice of city and the operational time redistribution for accessibility of the service.

The project wishes to provide foundations to answer these question using the basic techniques of extraction, transformation and loading/storage as depicted in the diagram below.

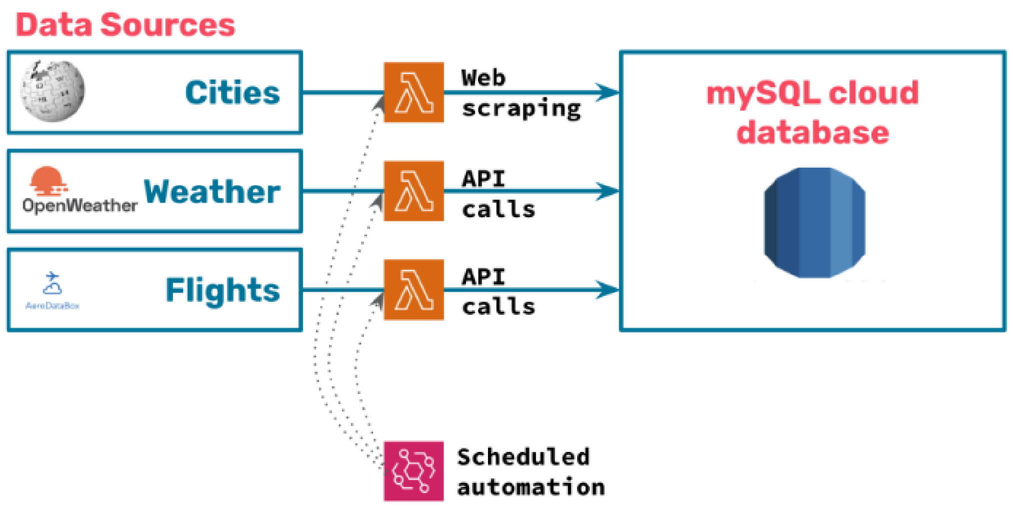

Figure 1. Diagram showing the intended methodology.

Goals and Objectives of the Project

The goal and objective of the project are;

- Obtain demographic data about cities in Europe to help the company decide into what cities to expand.
- Obtain weather and flight information about these cities to judge human traffic frequency, and manage asymmetries which develop due to these activities.
- Automate the collection of these data.

Methodology

The method utilized to achieve these objectives and goals are;

- Data collection; involves scrapping the web for data which might be necessary, and extracting these data (html format) via python library BeautifulSoup. Data is also be collected using APIs via the requests Library in python.
- Data Storage; the data obtained from the web scrapping and APIs are stored in a local database on MySQL called gans_data_analytics, and as csv files.
- Data collection automation; this involves using the Amazon Web Services relational database service (RDS), and the Amazon Lambda to setup instances and implement scripts.

Web Scrapping for Data

For Cities of Choice

To obtain necessary information for cities and countries, the choice of cities and countries is made based on the following criteria:

- Obtain the cities with the busiest airports in Europe.
- Obtain the cities with the most tourist visit.
- Obtain the cities with the highest populations.

These cities are gleaned from some websites using basic web scrapping technique and libraries i.e. BeautifulSoup, JSON, python Requests etc., with extensive data cleaning to obtain necessary information, and screening/exclusion of certain cities based on chosen criteria. The websites and the steps to webscrape are as specified in the included pdf version of the Juypter notebook called Gans Data Engineering: Pipeline in the Cloud. The obtained data are joined together to give the whole city of interest table.

Demographic data such as population of city, elevation of city, and longitude and latitude of these cities are obtained from API calls specified in the notebook.

These information for cities were put together into a table called "complete_cities_table".

For Country Demographic Data

To obtain the country demographic (physical, geographic and economic) data for the cities of interest/choice, the following sources of data were utilized;

- Gross domestic product (GDP) data was obtained from downloadable excel files provided by the World Bank. The data was extensively cleaned to isolate the required data.
- The ISO-code of countries are also obtained.
- An API service provided on rapidapi.com provides the geographical demographic data. This involved creating functions which takes input from country code and return the country demographic data.
- These information for cities were put together into a table called "complete_country_table".

The data compiled for cities and the countries are stored as csv-files, and also stored in the MySQL relational database (gans_data_analytics) on a local machine with the names; "complete_country_table", and "complete_cities_table". The connection between python and MySQL is made using SQLAlchemy. In the local database, the relationship between tables are defined, as well as the datatypes of all columns in the tables.

API Calls to Obtain Weather, Airport and Flight Arrival Information

The weather, airport and flight arrival information for the chosen cities are obtained by creating several functions to make API calls to certain database. The names of these functions are "get_cities_weather", "get_airport_info", "get_airport_weather", "airport_arrival". The input to these functions are the name of the cities as a list, and the results from one of the created functions. The weather is obtained from a 5 days – 3 hours step forecast API. Other APIs used and their implementation are as specified in the included notebook.

Implementation on the Cloud Based Service

The goal of this stage is the automation of the data extraction and storage. The cloud based service chosen is the Amazon Web Services.

This stage involved two steps;

- A relational database service (RDS) instance called "my-project3-database" which serves as the connection container to one's local machine database and other databases was created. This instance is setup to connect to the local database.
- The python scripts i.e. the functions mentioned in the foregoing, to automate the data collection were housed and implemented in AWS Lambda function called "gans_data" after proper modification of the code to fit the required syntax for the Lambda handler function. The "AWSDataWrangler-Python39", "Klayers-p39-SQLAlchemy" and "Klayers-p39-requests" layers were added to the function setup to facilitate implement-ability of the python imported libraries. A trigger called "Events Bridge (CloudWatch Events)" is setup to initiate the Lambda function at 11:50 pm everyday.

Conclusion

The methodologies employed in this project were able to provide the preliminary data that will serve to help Gans arrive at its decision on what countries, and cities to run her operation, and at what times to initiate a redistribution of e-scooters.