Table of Contents

1.0 Introduction.................................................................................................................. 2

1.1 Data Gathering...................................................................................................................... 3

1.1.1Company Data.................................................................................................................................. 3 1.1.2 Emission Data......................................................................................................................................... 4 2.0 Data Storage and Processing ....................................................................................... 7

2.1 Data Storage and File Choice ................................................................................................ 7

2.2 Database Selection............................................................................................................... 7

2.3 Data Size and Complexity ....................................................................................................10

Model building and training

Endpoint and API testing

3.1 Airflow and Docker ................................................................................................................................13

AWS IAM

3.0 Reproducibility and Version Control........................................................................... 13

3.2 GitHub ................................................................................................................................14

4.0 Project Management................................................................................................. 15

Data lineage

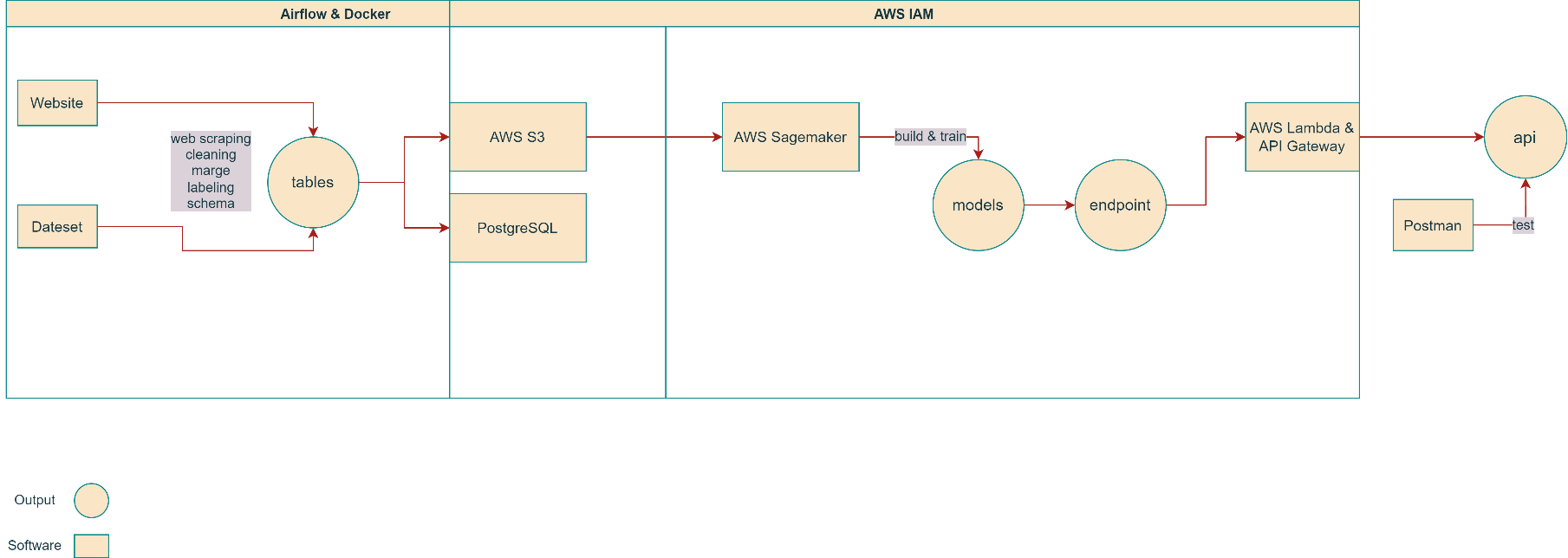
5.0 Real world application with the dataset..................................................................... 15

6.0 Summary and Limitations .......................................................................................... 16

7.0 References................................................................................................................. 17

Introduction

This project is aim to gather new datasets that contain company, geographic information as well as emission information. This project standalone contains a simple model, which provides simple prediction, and the trained endpoint can be accessed by API. My dissertation will build on this project with addition larger size company data and third-party satellite data.

The overall structure of this project is showed in the below pipeline:

The entire process can be divided into two parts. The first part is the data collecting part, which is automated by the help of airflow and docker. The data file is stored in AWS S3 database, then it will be used as model input to train a machine learning model. The final output is an api that contains the endpoint of a trained model. It will then be used to make prediction. The second part is fully AWS cloud based, and it is managed by AWS IAM.

Data Gathering

There are mainly two data sources. Company information data is captured from UK government website:

https://find-and-update.company-information.service.gov.uk/alphabetical-search

by python web scraping technique.

Unlike company data, emission data is from an existing excel file:

https://data.gov.uk/dataset/723c243d-2f1a-4d27-8b61-cdb93e5b10ff/uk-local-authority-and-regional-carbon-dioxide-emissions-national-statistics-2005-to-2019

Here is the data headers from the two sources:

|  |  |
| --- | --- |
| Data retired from company information gov uk | Data retired from emissions dataset gov uk |
| 1. Company name | 1. Local authority |
| 1. Company number | 2. Region country |
| 1. Status | 3. Year |
| 1. Link | 4. Industry electricity |
| 1. Address | 5. Industry gas |
| 1. Type | 6. Industry other fuels |
| 1. Incorporated time | 7. Large industrial installations |
| 1. Industry code | 8. Agriculture |
| 1. Post code | 9. Industry total |
| 1. Latitude | 10. Operator |
| 1. Longitude | 11. Site |
|  | 1. Post code |
|  | 1. Reference |
|  | 1. Substance name |

Company Data:

The company data is consisted of two tables: company list table and company details table.

The data retrieval process from Gov UK does not have too much restriction. However, the size of the data is extremely large. For time and resource concern, only the data for the first 100 pages are retrieved. Each page includes 40 records of company. In this case, there are 4000 records in the company list table. Based on the urls stored in the company list, company details are captured and store in the company detail table. Likewise, for time and resource concern, there are only the first 700 company details data is retrieved.

Emission Data

Emission data was entirely downloaded from the Gov UK dataset website. Within the dataset, two excel sheets were used in this project. One shows the data of Co2 emission for companies in different years. The another one shows the co2 emission in an area scale.

Data Storage and Processing

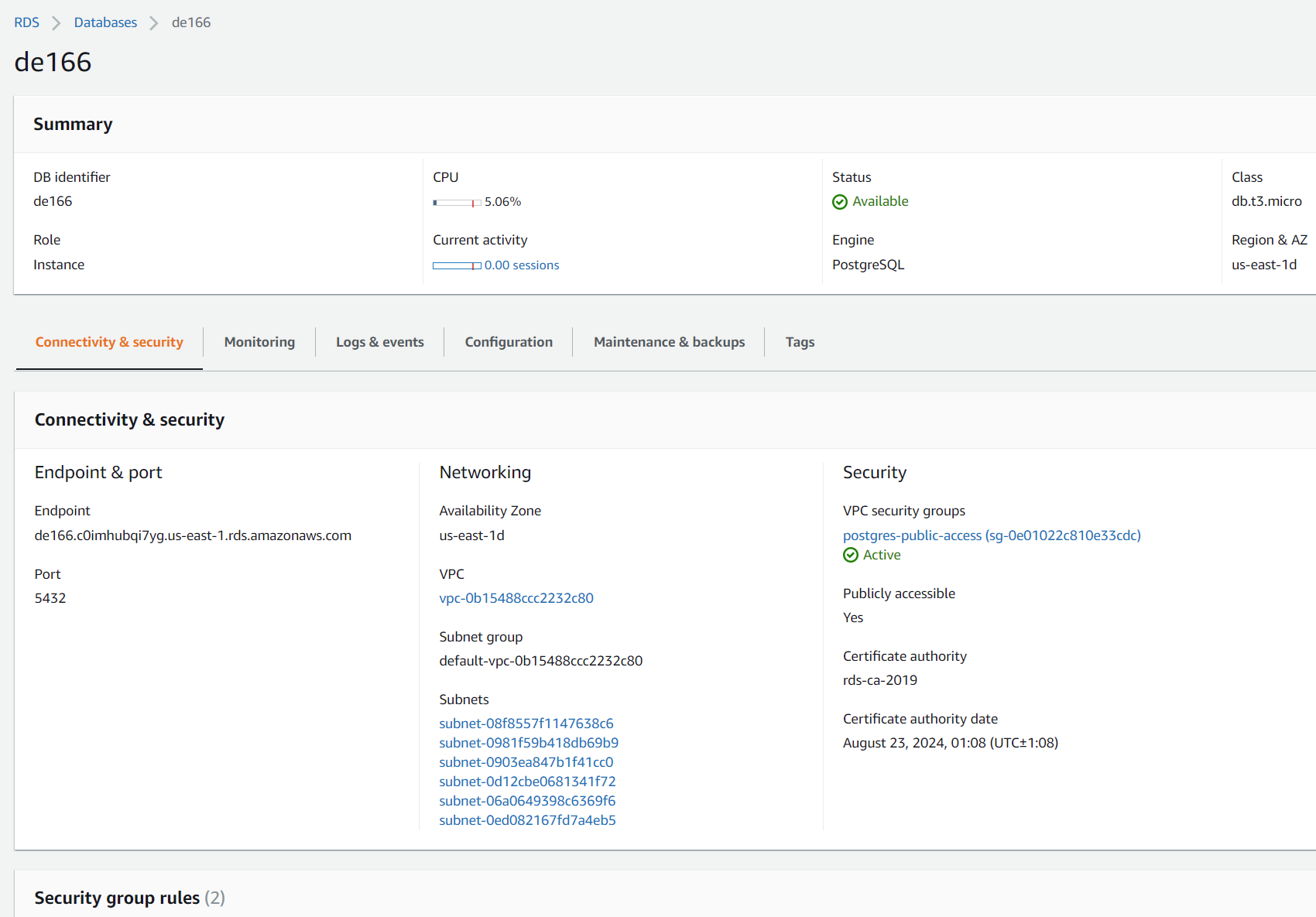
After gathering the data from different sources, these data was stored in csv and parquet formats since some S3 operation requires CSV format and Parquet has advantage on storing large size data. Parquet is efficient and performant in both storage and processing. If your dataset has many columns, and your use case typically involves working with a subset of those columns rather than entire records, Parquet is optimized for that kind of work. (Team et al., 2018)

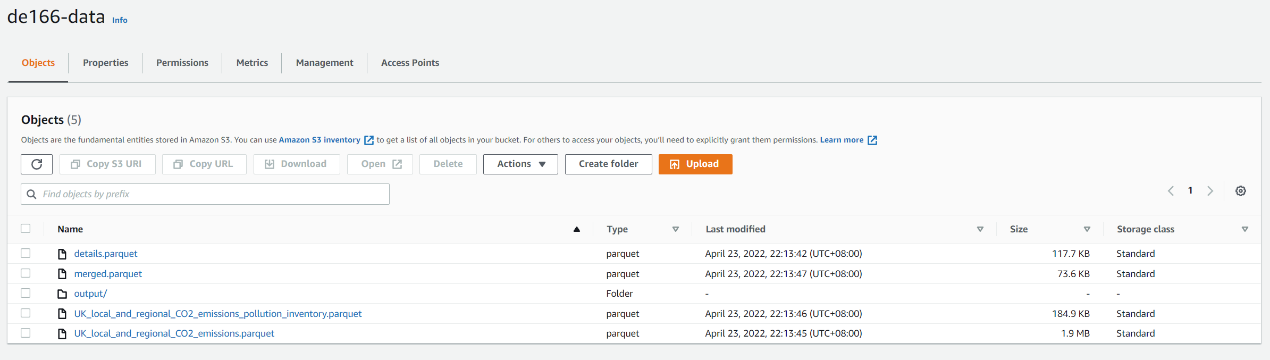
In the processing stage, data from different sources are cleaned, merge, and labeling by simple rules. These includes basic type change, string replace, format changing, fill or drop necessary data, get dummy variable for certain data and so on.

Database Selection.

There are two databases (PostrgreSQL and AWS S3) were used in this project. As a relational database, PostgreSQL is highly reliable, stable and secure. It is also easy to manage since tables have standard relations and schemas with each other. The reason for having AWS S3 it that many other AWS service require or perform better on S3’s data. Meanwhile, AWS S3 has the advantage of scalable and security. This project may require more data in the future so a scalable system is preferred.

The two figures below are the screenshots of PostrgreSQL and AWS S3.



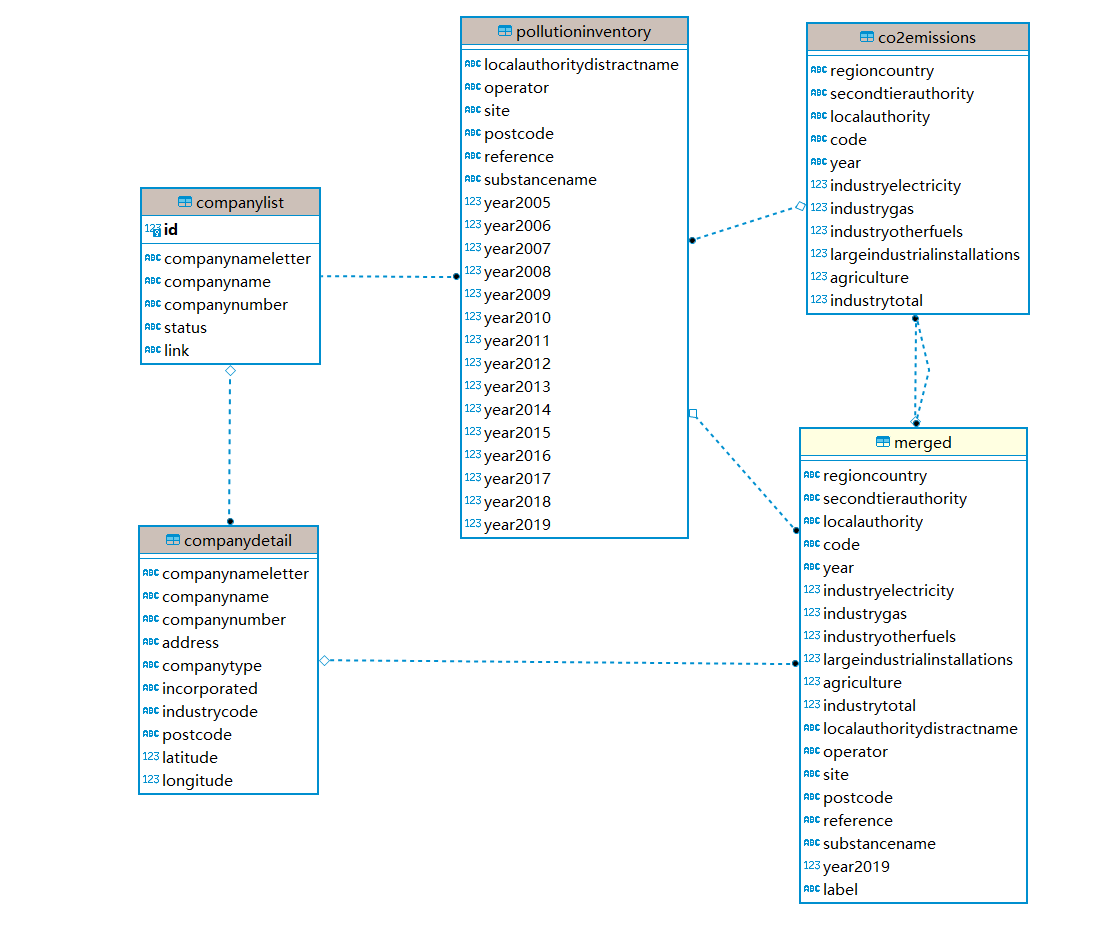


Data Size and Complexity

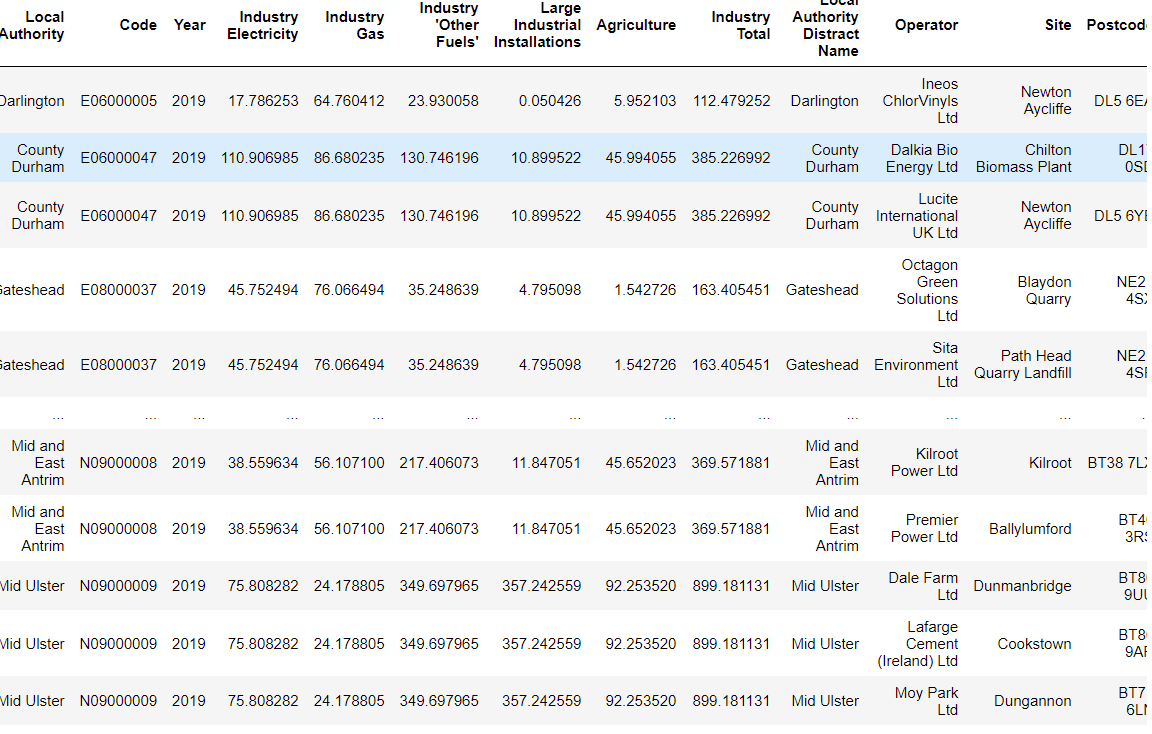
In this project, pyspark and psycopg2 are used to achieve better processing speed for large dataset. As a baseline, the maximin records number for one table is 4000, but in the future, it may largely increase.

The merged table is the input for model, and it needs data from both company data and emission data. However, due to the limited record number in company detail, the merged table is mainly consisted by data from emission data. This issue will be solve by increasing the record number in company details.

This picture below shows the schema for the database.

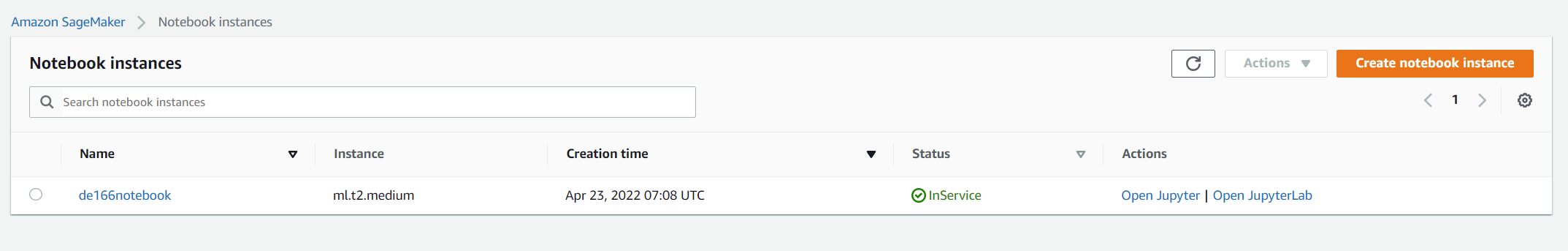


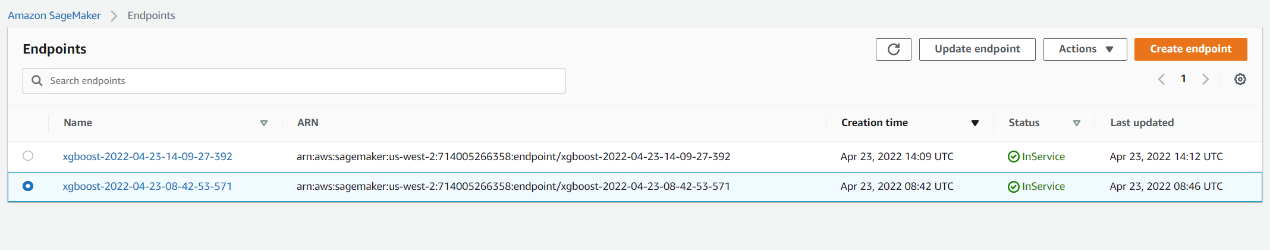
This figure presents an overview of the the merged table. However, to apply a xgboost model, this dataset needs a further cleaning. The label comlunm was moved to the first column, and necessary categorial data was turned into dummy variables.

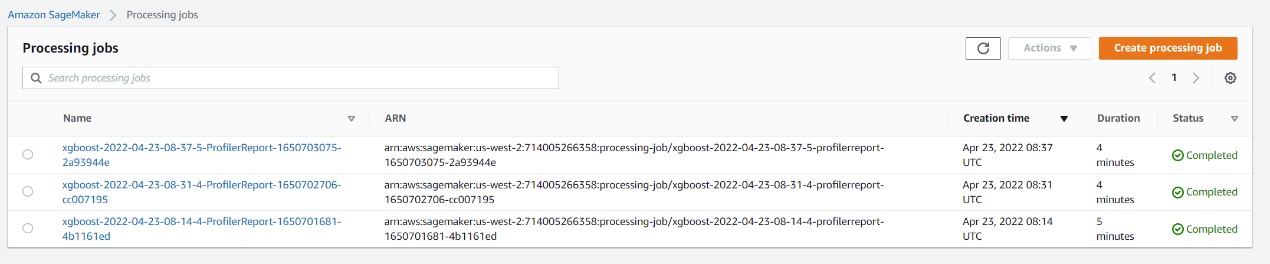


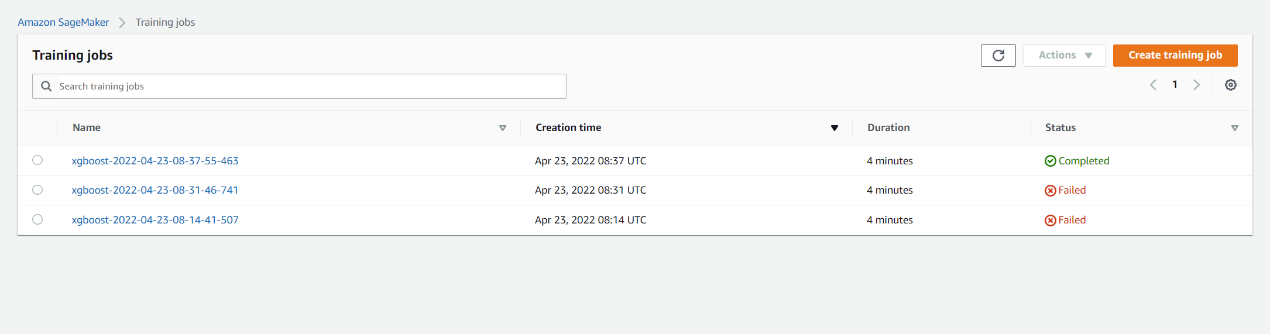
Model building and training

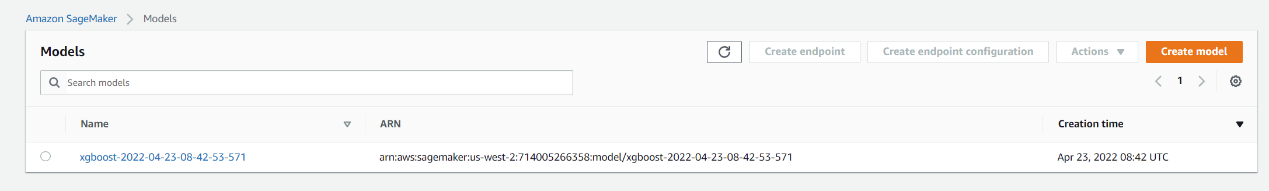
The entire model building and training process was built on AWS sagemaker. First I trigged the notebook instance, then based on the script in the notebook, it performed a processing job and training job. Meanwhile, the xgboost model is created. After the train job was done, the endpoint was created, and it was ready to predict.









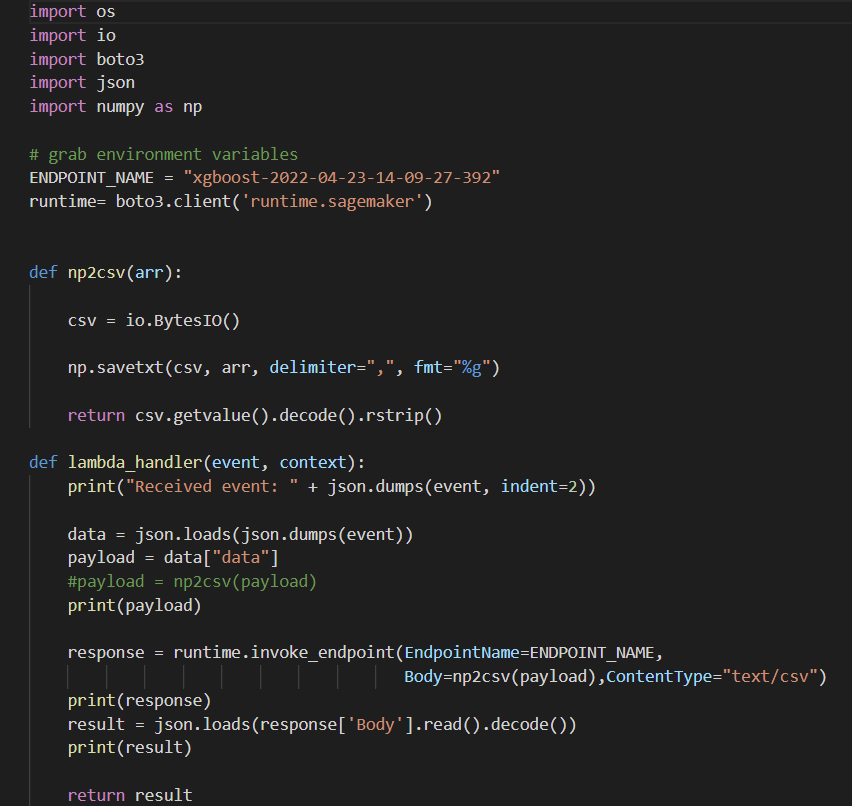


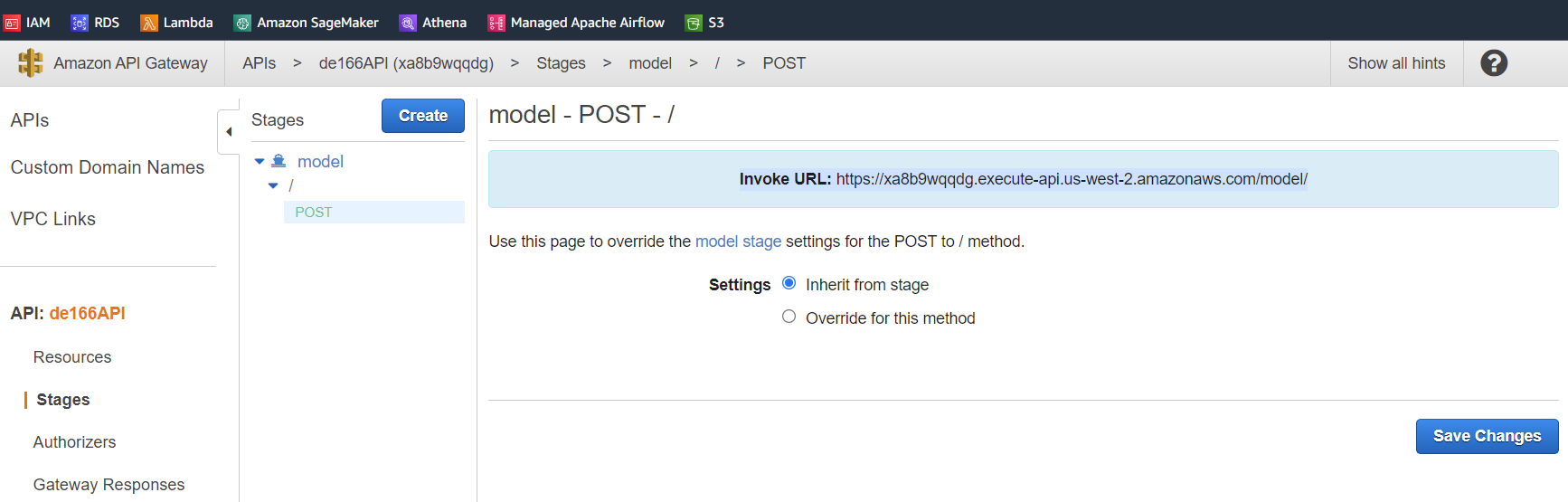
Endpoint and API testing

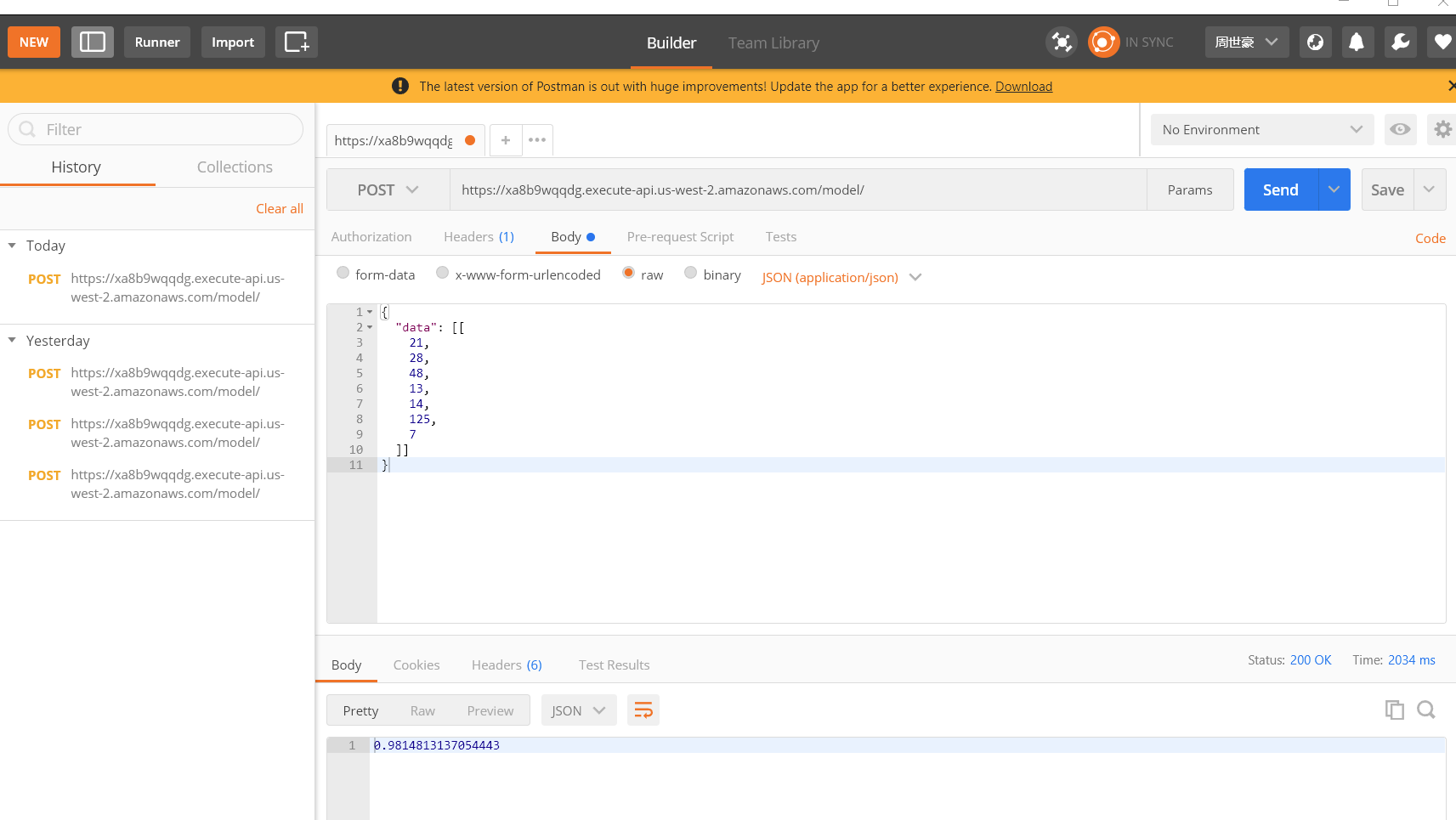
lambda function is a function that takes input data and invokes the trained model endpoint to predict. After that it returns the prediction output. However, it is hard to trigger by the lambda function alone. For this reason, the REST api was built.

The REST api was built from AWS API Gateway, and it provides a link to call the prepared lambda function.

The testing stage was contributed by Postman. By sending a json script in a post request to the our api, the lambda function was triggered. It then called the endpoint and inputted the data we wrote in json script. Finally, it returned the expected prediction outcome.





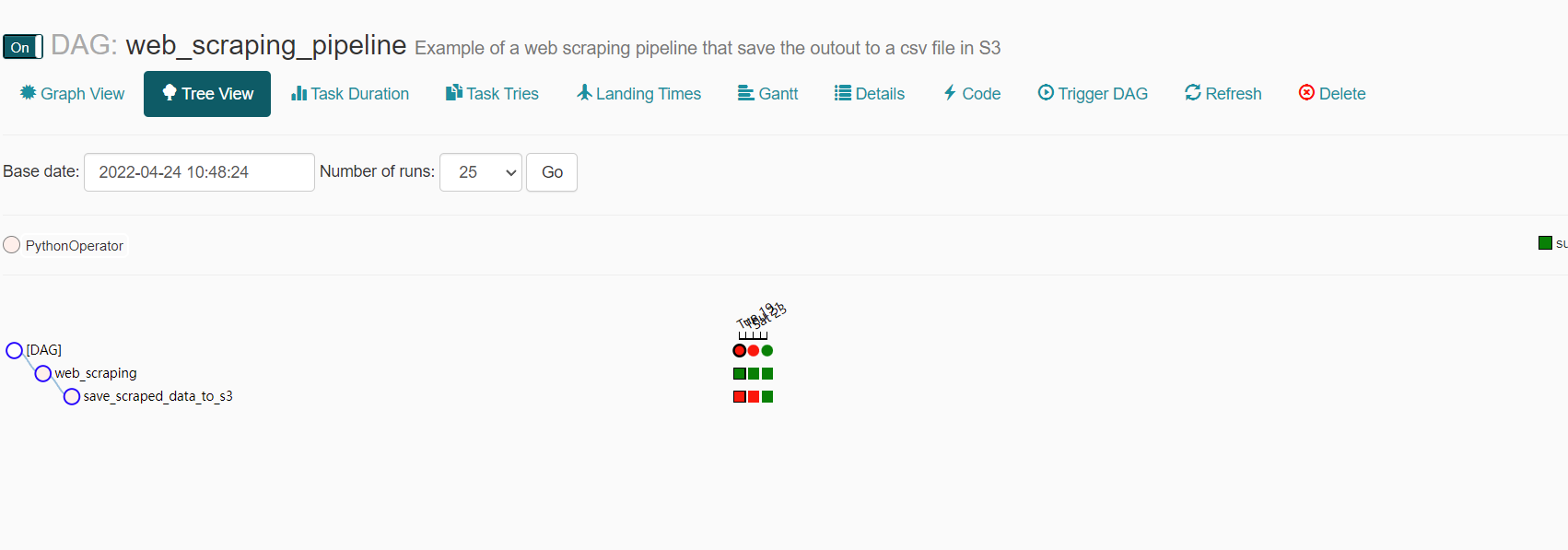


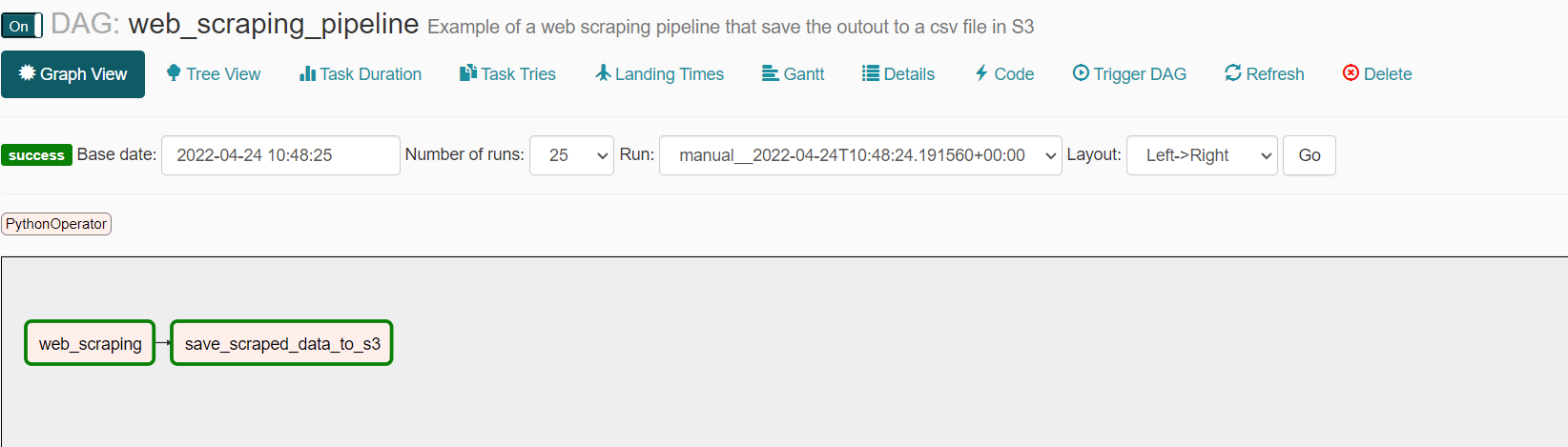
Airflow and Docker

As mentioned earlier in the introduction. The entire process can be views as two parts. The first part is from gathering data to storing in the database. In practice, this part of process is complicated and highly depend on a stable environment. It is easy to make mistakes since it has so many detail steps. In order to automatedly generate dataset by simply passing an url, a combination of Airflow and Docker is made for this project.

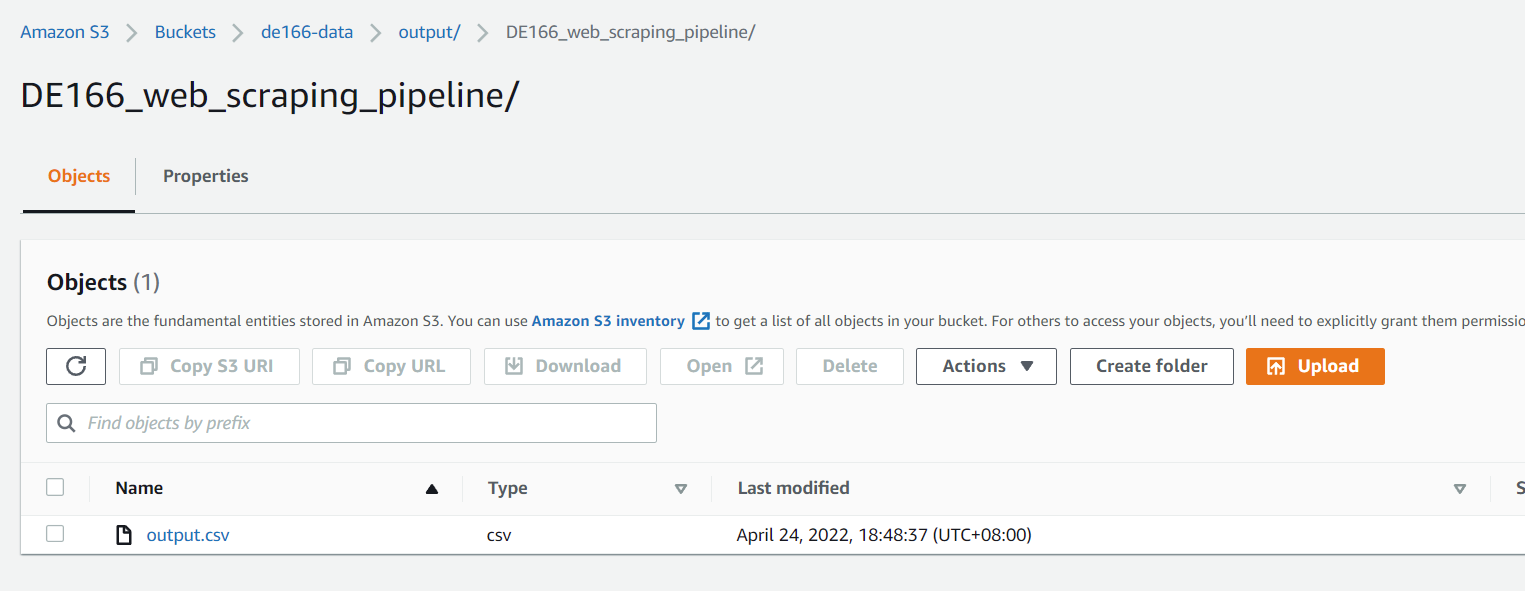
By utilizing Docker, the entire workflow can be performed in a stable setup with all the necessary package or software installed. To be more specific, Docker packs all the supporting packages and software into a container, and run the program in the container. It is similar to a virtual machine but more light weight than a virtual machine since it doesn’t contain the entire operation system and it does not require any additional physical hardware.

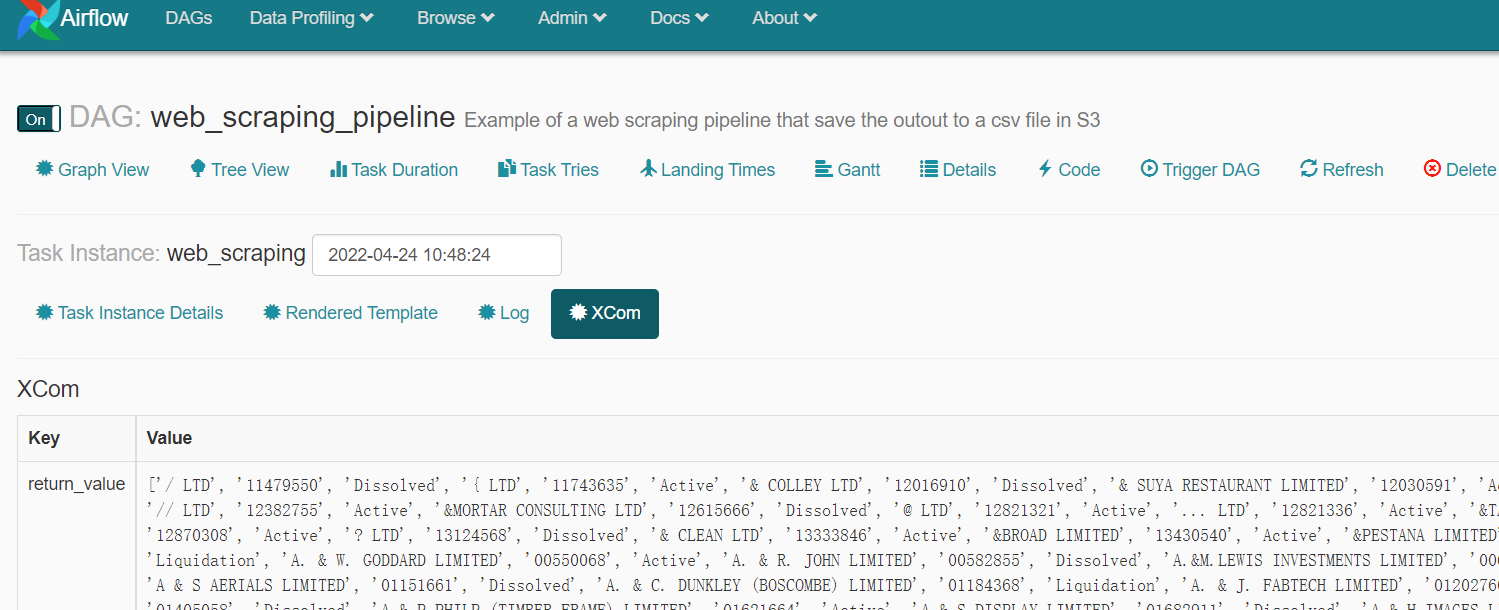
The Airflow is also built upon docker environment. After writing the DAG file and setting up all the needed variables, scheduled job can be performed daily weekly or in any other arrangement.





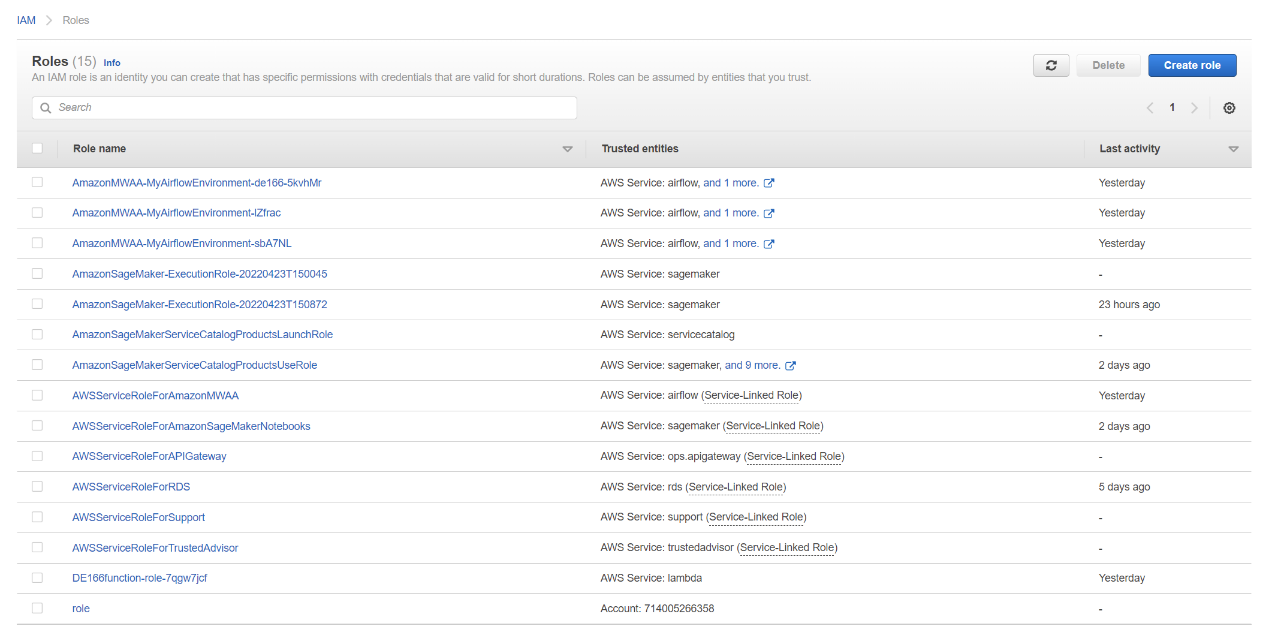






AWS IAM

The second part of the project, which is from getting data from S3 to generating the api from endpoint, was completely run on AWS cloud server. This part was managed by AWS IAM system, which is a security system that grant different level of access to users or roles.





<https://www.gov.uk/government/statistics/uk-local-authority-and-regional-carbon-dioxide-emissions-national-statistics-2005-to-2019>

<https://data.gov.uk/dataset/723c243d-2f1a-4d27-8b61-cdb93e5b10ff/uk-local-authority-and-regional-carbon-dioxide-emissions-national-statistics-2005-to-2019>

<https://find-and-update.company-information.service.gov.uk/>

<https://practicaldatascience.co.uk/data-science/how-to-geocode-and-map-addresses-in-geopy>

https://geopy.readthedocs.io/en/stable/#installation

Team, DataFlair, et al. “List the Advantage of Parquet File in Apache Spark.” DataFlair, 20 Sept. 2018, https://data-flair.training/forums/topic/list-the-advantage-of-parquet-file-in-apache-spark/.