CONTENT PAGE

[1.1 Project Overview 2](#_Toc168295562)

[1.2 Scope of Project 2](#_Toc168295563)

[1.2 Project Challenges 2](#_Toc168295564)

[1.3 Solutions to Challenges 2](#_Toc168295565)

[1.4 System Architecture 3](#_Toc168295566)

[1.5 Setup and Manual Running 4](#_Toc168295567)

[1.6 Technologies Stack 5](#_Toc168295568)

[1.8 Future Updates 5](#_Toc168295569)

[1.9 Glossary 5](#_Toc168295570)

**ECONOMIC SHIFT PROJECT**

**TECHNICAL DOCUMENTATION**

### Project Overview

[Econdb](https://www.econdb.com/) is a leading data service for economic indicators and the shipping industry. This project seeks to create a pipeline for obtaining economic data from [econdb.com](https://www.econdb.com/) and make it available for dash boarding. Project aims at creating a resilient and fault proof data pipeline for obtaining the economic data.

### 1.2 Scope of Project

The project seek to automate the fetching and preparation of economic data for dash boarding and any other purpose.

### Project Challenges

The main challenge that was face during the project implementation was getting the data. This is because there were limitations placed on the API in regards to the number of API calls that can be made using the given token and expiration within a given number of days. To gain full access to the API, one could pay a 3 month subscription of $20.

Another challenge was underutilization of resources. Only a single request is made at a time.

### Solutions to Challenges

In completing this project, there were 3 ways of going about the first problem. These are:

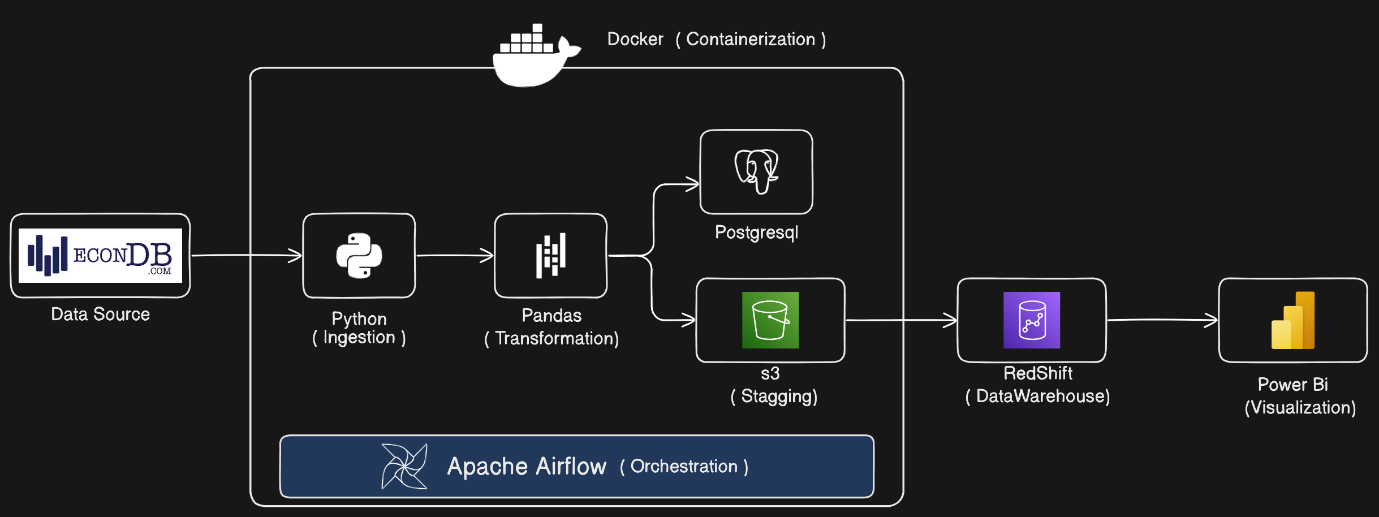
1. Paying for the 3 month subscription removing limits on the number of API (Application Programming Interface) calls.
2. Scraping the web page for the required data.
3. Requesting JSON (JavaScript Object Notation) file using for rendering data on the page.

In solving the problem, I resorted to 3 since it gave me full access to the data and semi-structure format (JSON) of the data without any additional fee. For commercial use please purchase the subscription.

To solve the second problem, multiprocessing is to be used so that multiple requests could be made which will help boost the performance. However, to avoid been detected as a bot and possibly been blocked, the project was completed without the multiprocessing aspect making it very simple.

### System Architecture

The architectural system design is as follows:



#### Data Source

The data for the pipeline is requested from the econdb.com server using python.

#### Data Transformation

The data is transformed into a JSON format while discarding unnecessary parts. This transformation is down using Pandas, a python library for data manipulation and analysis.

#### Data Loading

The cleaned data is loaded to a PostgreSQL database and an Amazon s3 bucket for staging. The data from the s3 bucket is then copied to Redshift data warehouse.

#### Data Visualization

Data from Redshift is visualized using Power BI, a business analytics tool. Power BI helps in creating interactive reports and dashboards.

#### Orchestration

The entire ETL (Extract Transform and Load) process is orchestrated using Apache Airflow. Airflow is an open-source tool used to programmatically author, schedule, and monitor workflows. It ensures that each step in the pipeline is executed in the correct order and handles dependencies between tasks.

#### Containerization

The entire pipeline is containerized using Docker. Containerization ensures that the application runs in a consistent environment across different stages of development, testing, and production. Docker containers manages all the dependencies, making the deployment process more efficient and reliable.

### Setup and Manual Running

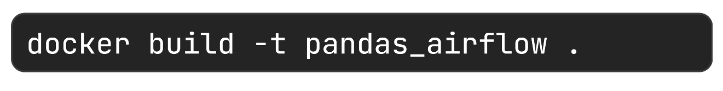
1. Clone the repo to the project by running the



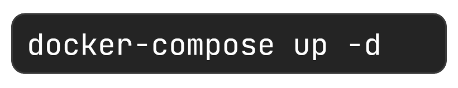
1. Navigate to the airflow folder/directory in your terminal by running



1. In the airflow directory, build the dockerfile by running

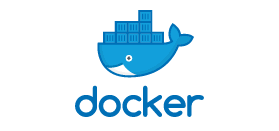


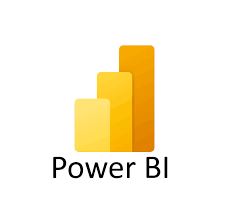
1. Update .env file with your details
2. Start the docker container by running



1. Navigate to your airflow ui at [*http://localhost:8080/*](http://localhost:8080/) in your browser
2. Run the new DAG
3. Check log to view logs of the running DAG
4. Check your s3 bucket to confirm that the files have been uploaded successfully

### Technologies Stack





### 1.8 Future Updates

Some future updates can include:

1. Updating data fetching process to allow multiprocessing to optimize the download especially when downloading bulk data for all countries.
2. Other economic features can be added (i.e. employment index, population growth, etc.)
3. Add IAC(Infrastructure As Code) technology like terraform to manage deployment
4. Alerting system to show daily updates of commodities and other futures through sms, telegram bot, etc.
5. Creating a machine learning model for predicting economic features based on trend.

### 1.9 Glossary

Abbreviation Meaning

IAC Infrastructure as Code

DAG Direct Acyclic Graph

ETL Extract Transform Load

JSON JavaScript Object Notation

API Application Programming Interface

S3 Simple Storage Service