**Steps to Run the Project**

1. **Install Visual Studio Code**

* Download and install [Visual Studio Code](https://code.visualstudio.com/).

1. **Install Astro CLI**

* Follow the instructions here to install the Astro CLI.
* Edit system environment variables if required.

1. **Install Docker Desktop**

* Download and install Docker Desktop here.

1. **Update Docker’s .wslconfig File**

* Navigate to C:\Users\<YourUsername>\.wslconfig and update it with the following configuration:

[wsl2]

memory=6GB

processors=6

1. **Run Docker Desktop**

* Open Docker Desktop to start the Docker service.

1. **Initialize Airflow Environment Using Astro CLI**

* Open Visual Studio Code and create a new folder for the project.
* Navigate to the terminal in Visual Studio Code and run the command:

astro dev init

* This command initializes the Airflow environment in the folder created.

1. **Update the .env File**

* Add the following configurations to the .env file:

FROM quay.io/astronomer/astro-runtime:11.6.0

ENV AIRFLOW\_\_SCHEDULER\_\_SCHEDULER\_ZOMBIE\_TASK\_THRESHOLD 3600

ENV AIRFLOW\_\_CORE\_\_DEFAULT\_TASK\_EXECUTION\_TIMEOUT 3600

ENV AIRFLOW\_\_SCHEDULER\_\_LOCAL\_TASK\_JOB\_HEARTBEAT\_SEC 600

ENV AIRFLOW\_\_SCHEDULER\_\_JOB\_HEARTBEAT\_SEC 10

ENV AIRFLOW\_\_SCHEDULER\_\_SCHEDULER\_HEARTBEAT\_SEC 10

1. **Update the requirements.txt File**

* Add the following configuration to the requirements.txt file:

astro-sdk-python[amazon, snowflake] >=1.1.0

1. **Start Airflow Core Components**

* Run the command:

astro dev start

* This starts the Airflow core components (Scheduler, Executor, Worker, Webserver, Metadata Database).

1. **Access Airflow Webserver**

* Once all components have started, navigate to <http://localhost:8080>.

1. **Create S3 Bucket**

* Create two separate folders in the bucket to store Raw data and transformed data

1. **Create Snowflake Tables**

* Create snowflake Database schema and tables as shown in Snowflake\_worksheet\_for\_dsesigning\_DB\_schema.txt

1. **Add ETL Scripts to DAG Folder**

* Place scripts/AIRFLOW\_ETL.py and scripts/etlfucntions.py inside the DAG folder in your project directory.
* Update bucket and snowflake table details in scripts/AIRFLOW\_ETL.py and scripts/etlfucntions.py files.

1. **Configure Airflow Connections**

* In the Airflow Webserver, navigate to Airflow Connections and add the following connections:
  + 1. **HTTP API**
       - **Connection Id**: http-api
       - **Connection Type**: HTTP
       - **Host**: <https://api.fda.gov/drug/event.json>?
    2. **AWS Default**
       - **Connection Id**: aws-default
       - **Connection Type**: Amazon Web Services
       - **AWS Access Key ID**: <Your AWS access key>
       - **AWS Secret Access Key**: <Your AWS secret key>
    3. **Snowflake**
       - **Connection Id**: snowflake\_conn\_id
       - **Connection Type**: Snowflake
       - **Login**: <Snowflake account username>
       - **Password**: <Snowflake account password>
       - **Account**: <Your account ID>
       - **Warehouse**: <Warehouse name>
       - **Database**: <Database name>
       - **Region**: <Region name>
       - **Role**: <Type of Role>

1. **Trigger the DAG**

* After a few minutes, you should see the DAG at <http://localhost:8080>.
* Schedule the DAG as per your needs and trigger it.