# Apache Airflow Final Project Report — ETL Toll Data Pipeline

## 1. Introduction

In today's data-driven world, efficient data extraction, transformation, and loading (ETL) processes are crucial for analytics and decision-making. This project demonstrates the use of **Apache Airflow** — a powerful workflow orchestration tool — to automate and schedule ETL tasks.

The objective is to process toll data from various file formats (CSV, TSV, and fixed-width text) and consolidate it into a single transformed dataset that can be used for traffic pattern analysis.

By completing this project, I gained **hands-on experience in workflow automation, task dependencies, data transformation, and Airflow DAG creation**.

## 2. Skills Learned

Through this project, I developed the following skills:

- **Apache Airflow DAG Design**: Creating and managing Directed Acyclic Graphs for task automation
- **Bash Scripting in Airflow**: Using BashOperator to execute Linux commands for ETL steps
- **ETL Workflow Automation**: Automating extraction, transformation, and loading processes
- **Data Manipulation in Linux**: Handling CSV, TSV, and fixed-width text files using shell commands
- **Dependency Management**: Establishing relationships between sequential Airflow tasks
- **Error Handling and Retries**: Configuring retries and alerts for robust pipeline execution
- **Airflow UI Monitoring**: Monitoring DAG runs, logs, and task progress visually

## 3. Objectives

This project aims to develop Apache Airflow DAG that will:

- Extract data from a csv file
- Extract data from a tsv file
- Extract data from a fixed-width file
- Transform the data
- Load the transformed data into the staging area

## Task 1: Set up the lab environment

- Start Apache Airflow.
- Open a terminal and create a directory structure for the staging area as follows:
  **/home/project/airflow/dags/finalassignment/staging**
- Execute the following commands to give appropriate permission to the directories.

  ```
  sudo chmod -R 777 /home/project/airflow/dags/finalassignment
  ```

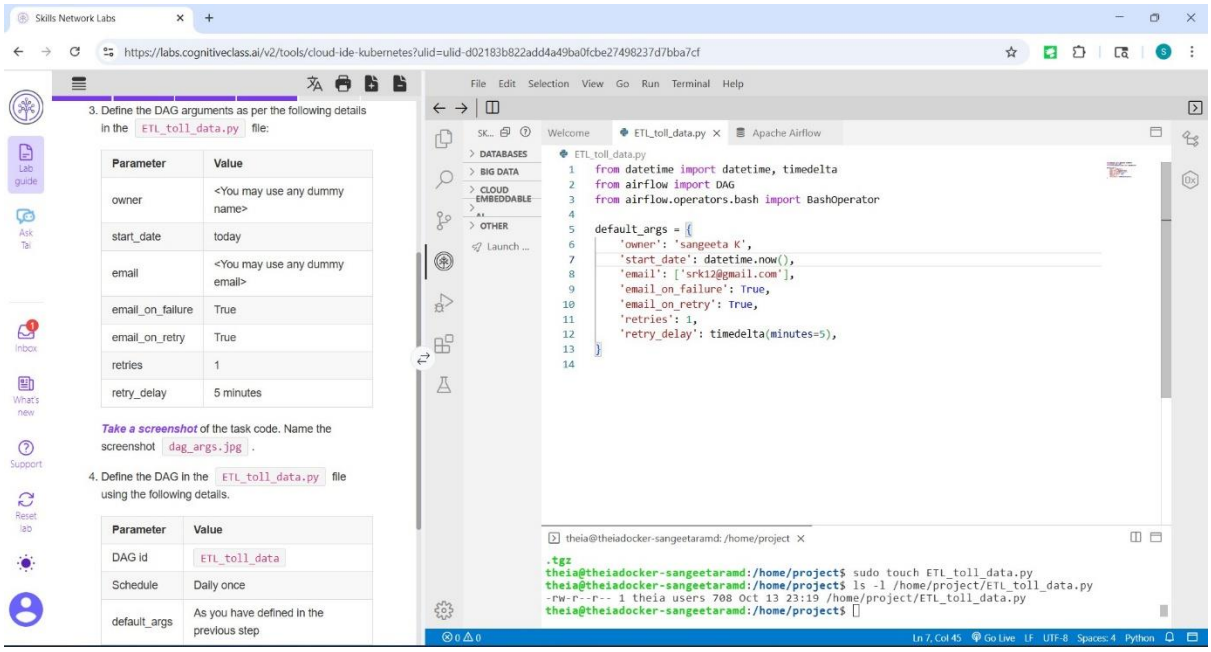
- Download the data set from the source to the following destination using the curl command.

  ```
  sudo curl https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Final%20Assignment/tolldata.tgz -o /home/project/airflow/dags/finalassignment/tolldata.tgz
  ```

## Task 2: Create imports, DAG argument, and definition

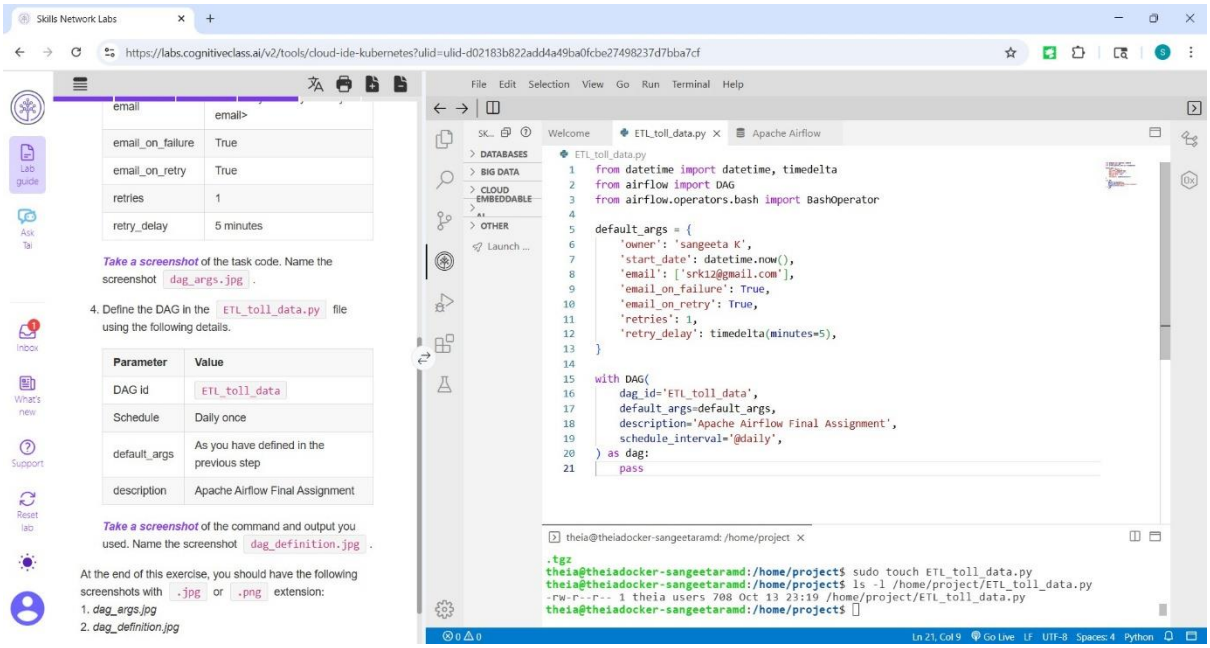
- Create a new file named ETL_toll_data.py in /home/project directory and open it in the file editor.
- Import all the necessary packages to build the DAG.
- Define the DAG arguments as per the following details in the ETL_toll_data.py file:

| Parameter | Value |
|---|---|
| owner | <You may use any dummy name> |
| start_date | today |
| email | <You may use any dummy email> |
| email_on_failure | True |
| email_on_retry | True |
| retries | 1 |
| retry_delay | 5 minutes |



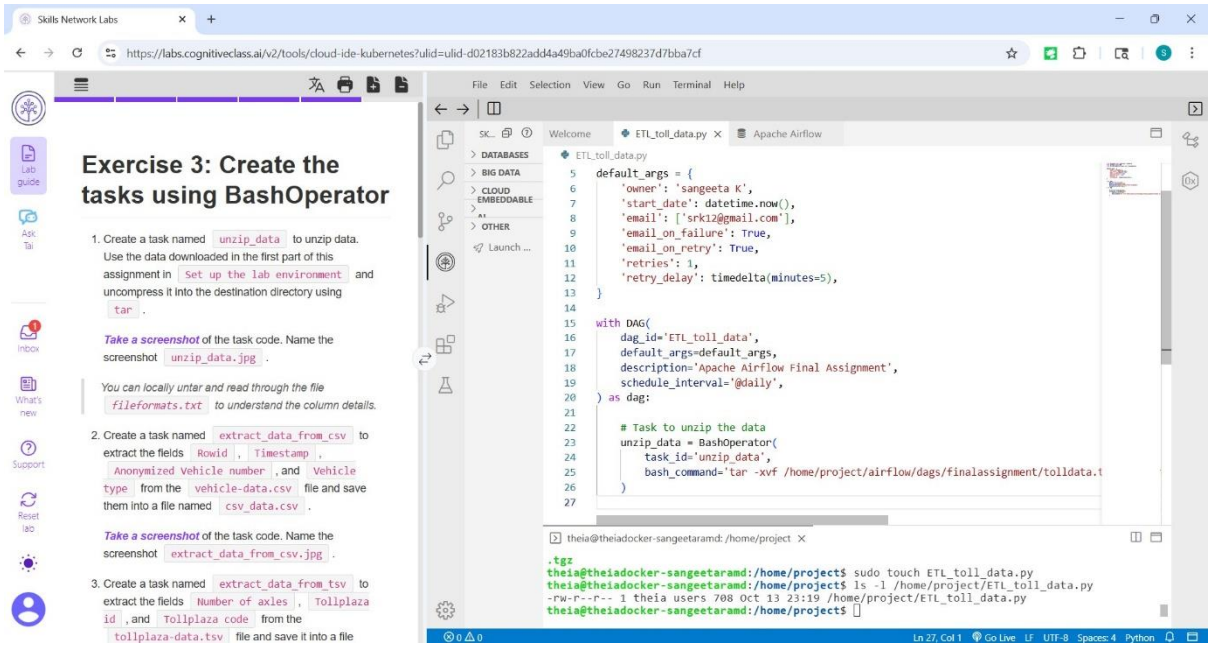
- Define the DAG in the ETL_toll_data.py file using the following details.

| **Parameter** | **Value** |
|---|---|
| DAG id | ETL_toll_data |
| Schedule | Daily once |
| default_args | As you have defined in the previous step |
| description | Apache Airflow Final Assignment |

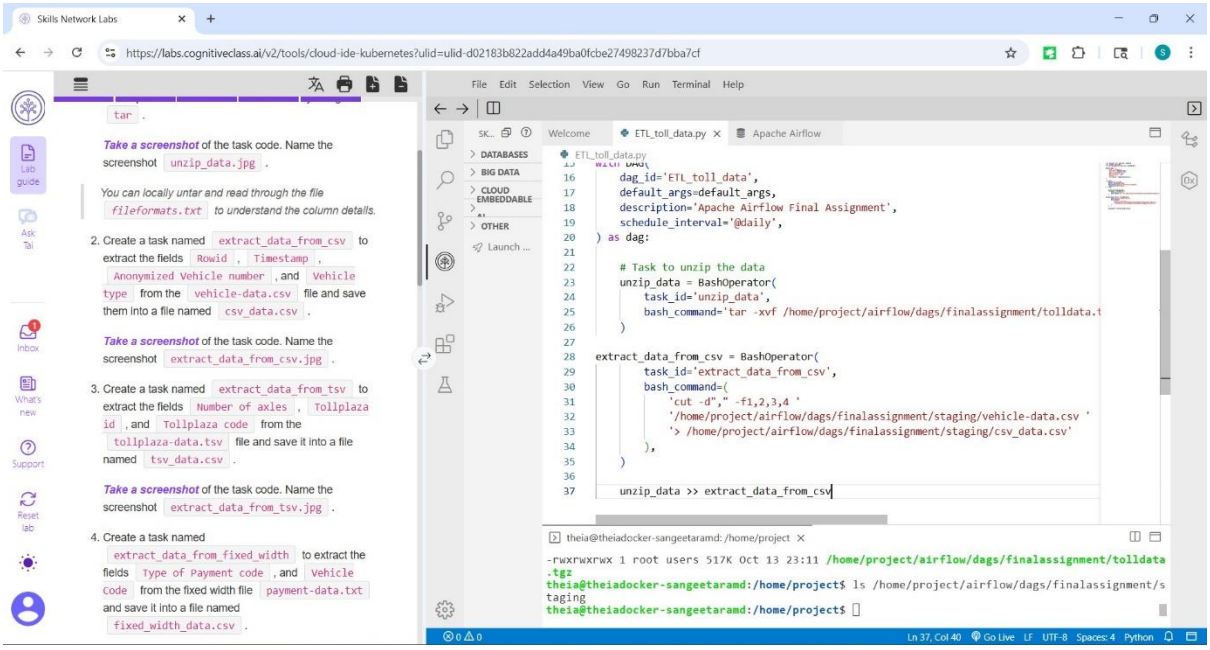


## Task 3: Create the tasks using BashOperator

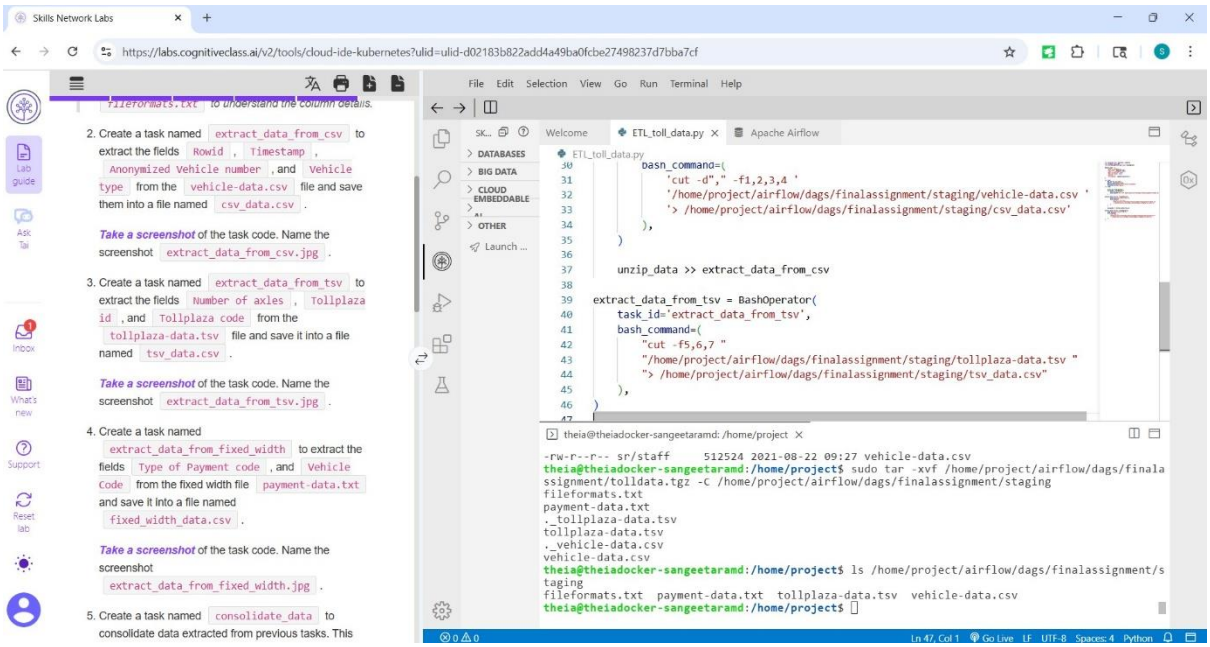
- Create a task named unzip_data to unzip data. Use the data downloaded in the first part of this assignment in Set up the lab environment and uncompress it into the destination directory using tar.



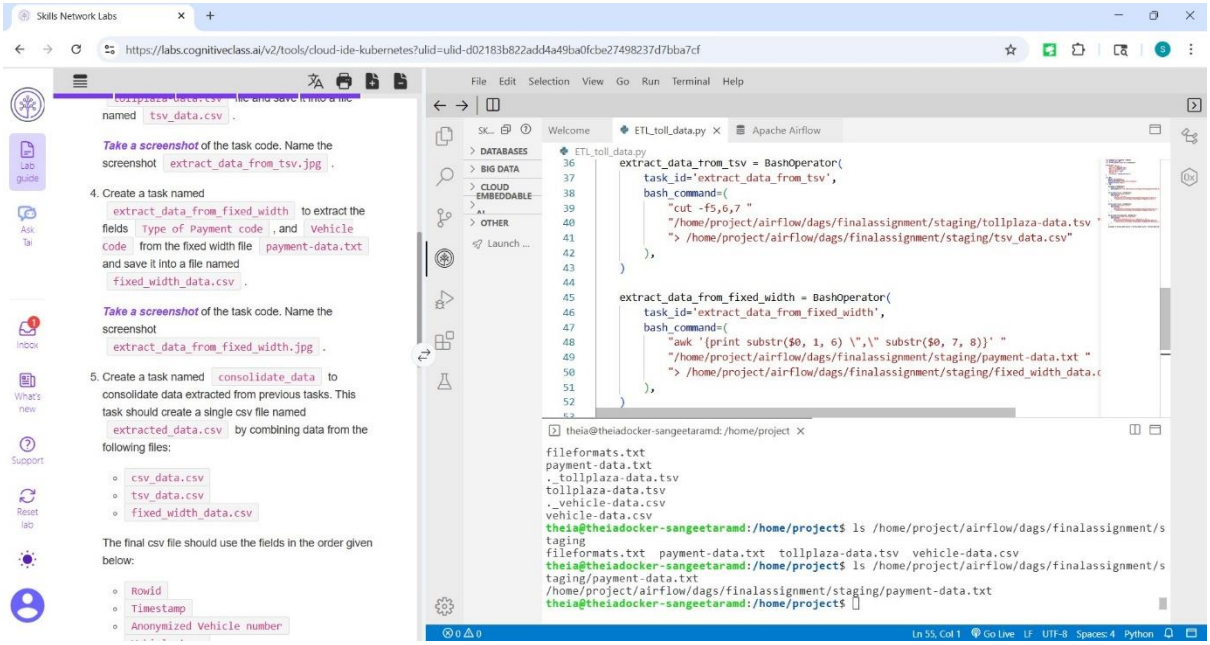
- Create a task named extract_data_from_csv to extract the fields Rowid, Timestamp, Anonymized Vehicle number, and Vehicle type from the vehicle-data.csv file and save them into a file named csv_data.csv.




- Create a task named extract_data_from_tsv to extract the fields Number of axles, Tollplaza id, and Tollplaza code from the tollplaza-data.tsv file and save it into a file named tsv_data.csv.

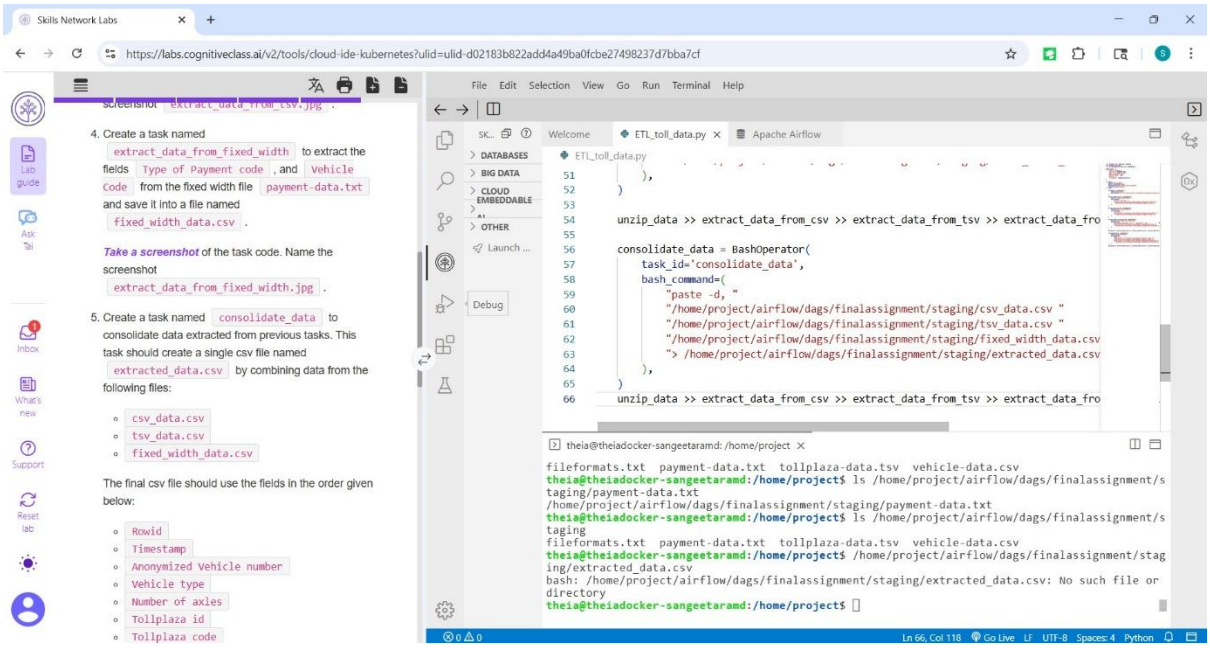


- Create a task named extract_data_from_fixed_width to extract the fields Type of Payment code, and Vehicle Code from the fixed width file payment-data.txt and save it into a file named fixed_width_data.csv.

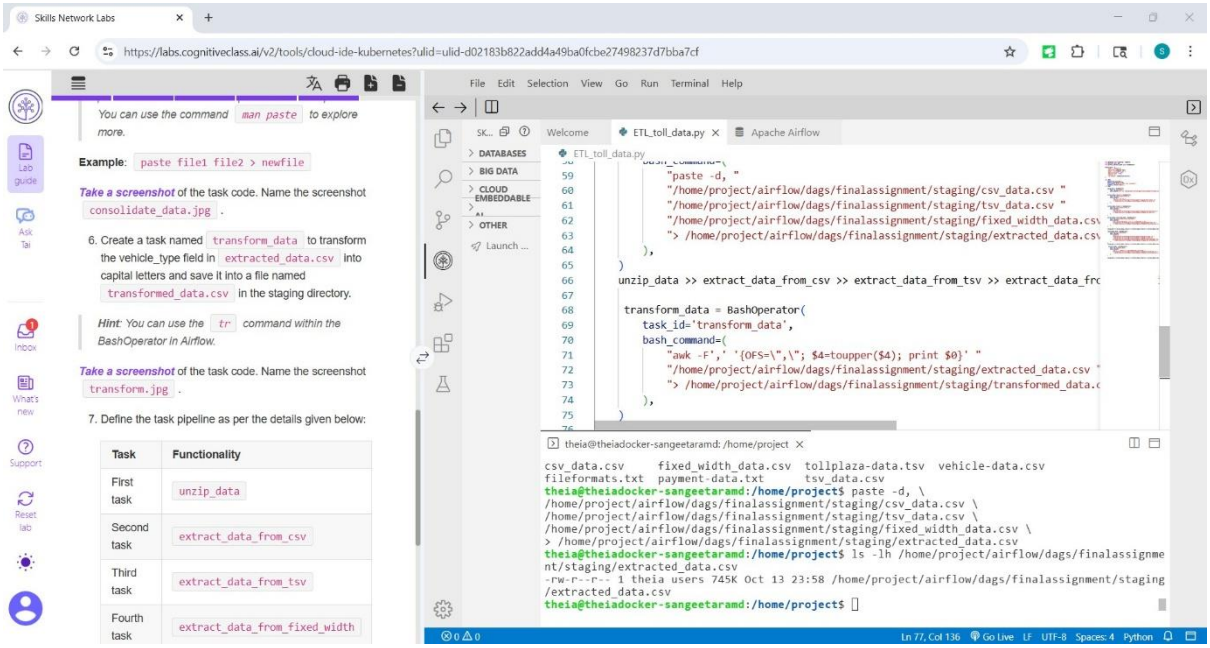



- Create a task named consolidate_data to consolidate data extracted from previous tasks. This task should create a single csv file named extracted_data.csv by combining data from the following files:
  - csv_data.csv
  - tsv_data.csv
  - fixed_width_data.csv

  The final csv file should use the fields in the order given below:
  - Rowid
  - Timestamp
  - Anonymized Vehicle number
  - Vehicle type
  - Number of axles
  - Tollplaza id
  - Tollplaza code
  - Type of Payment code
  - Vehicle Code

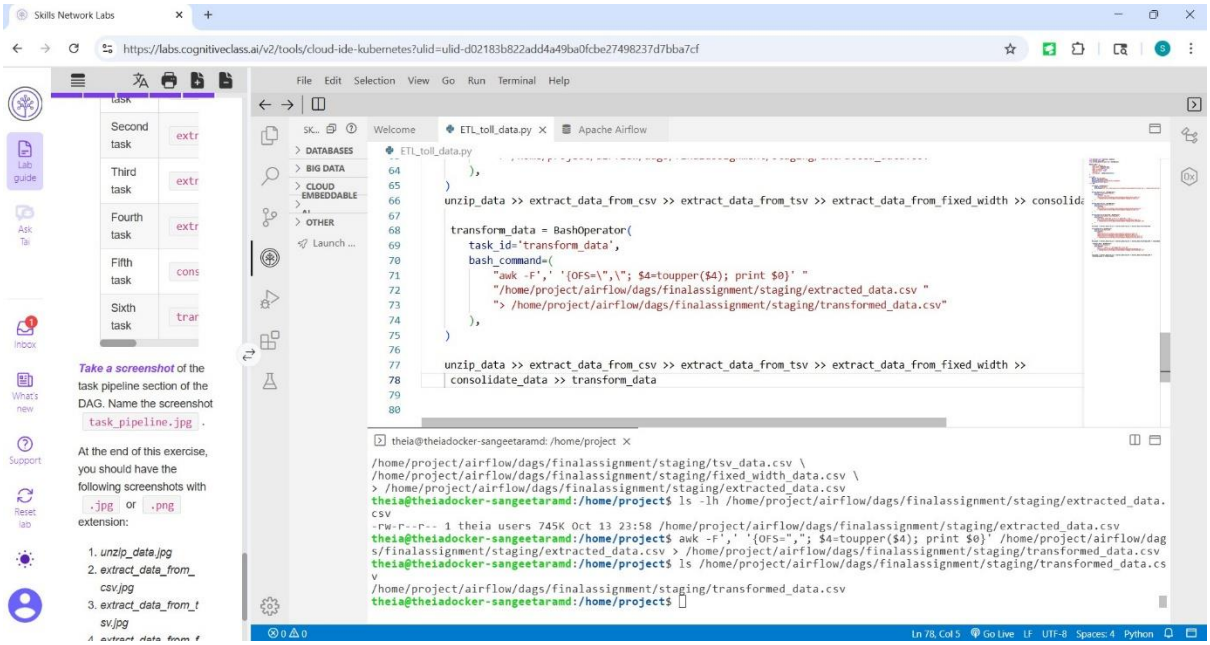

Skills Network Labs
https://labs.cognitiveclass.ai/v2/tools/cloud-ide-kubernetes?ulid=ulid-d02183b822add4a49ba0fcbe27498237d7bba7cf
screenshot extract_data_from_csv.jpg .
4. Create a task named extract_data_from_fixed_width to extract the fields Type of Payment code , and Vehicle Code from the fixed width file payment-data.txt and save it into a file named fixed_width_data.csv .
Take a screenshot of the task code. Name the screenshot extract_data_from_fixed_width.jpg .
5. Create a task named consolidate_data to consolidate data extracted from previous tasks. This task should create a single csv file named extracted_data.csv by combining data from the following files:
csv_data.csv
tsv_data.csv
fixed_width_data.csv
The final csv file should use the fields in the order given below:
Rowid
Timestamp
Anonymized Vehicle number
Vehicle type
Number of axles
Tollplaza id
Tollplaza code
File Edit Selection View Go Run Terminal Help
Welcome
ETL_toll_data.py
Apache Airflow
DATABASES
BIG DATA
CLOUD EMBEDDABLE
OTHER
Launch ...
Debug
unzip_data >> extract_data_from_csv >> extract_data_from_tsv >> extract_data_fro
consolidate_data = BashOperator(
task_id='consolidate_data',
bash_command=(
"paste -d, "
"/home/project/airflow/dags/finalassignment/staging/csv_data.csv "
"/home/project/airflow/dags/finalassignment/staging/tsv_data.csv "
"/home/project/airflow/dags/finalassignment/staging/fixed_width_data.csv
"> /home/project/airflow/dags/finalassignment/staging/extracted_data.csv
),
)
unzip_data >> extract_data_from_csv >> extract_data_from_tsv >> extract_data_fro
theia@theiadocker-sangeetaramd: /home/project
fileformats.txt payment-data.txt tollplaza-data.tsv vehicle-data.csv
theia@theiadocker-sangeetaramd:/home/project$ ls /home/project/airflow/dags/finalassignment/staging/payment-data.txt
/home/project/airflow/dags/finalassignment/staging/payment-data.txt
theia@theiadocker-sangeetaramd:/home/project$ ls /home/project/airflow/dags/finalassignment/staging
fileformats.txt payment-data.txt tollplaza-data.tsv vehicle-data.csv
theia@theiadocker-sangeetaramd:/home/project$ /home/project/airflow/dags/finalassignment/staging/extracted_data.csv
bash: /home/project/airflow/dags/finalassignment/staging/extracted_data.csv: No such file or directory
theia@theiadocker-sangeetaramd:/home/project$
Ln 66, Col 118
Go Live
LF
UTF-8
Spaces: 4
Python

- Create a task named transform_data to transform the vehicle_type field in extracted_data.csv into capital letters and save it into a file named transformed_data.csv in the staging directory.



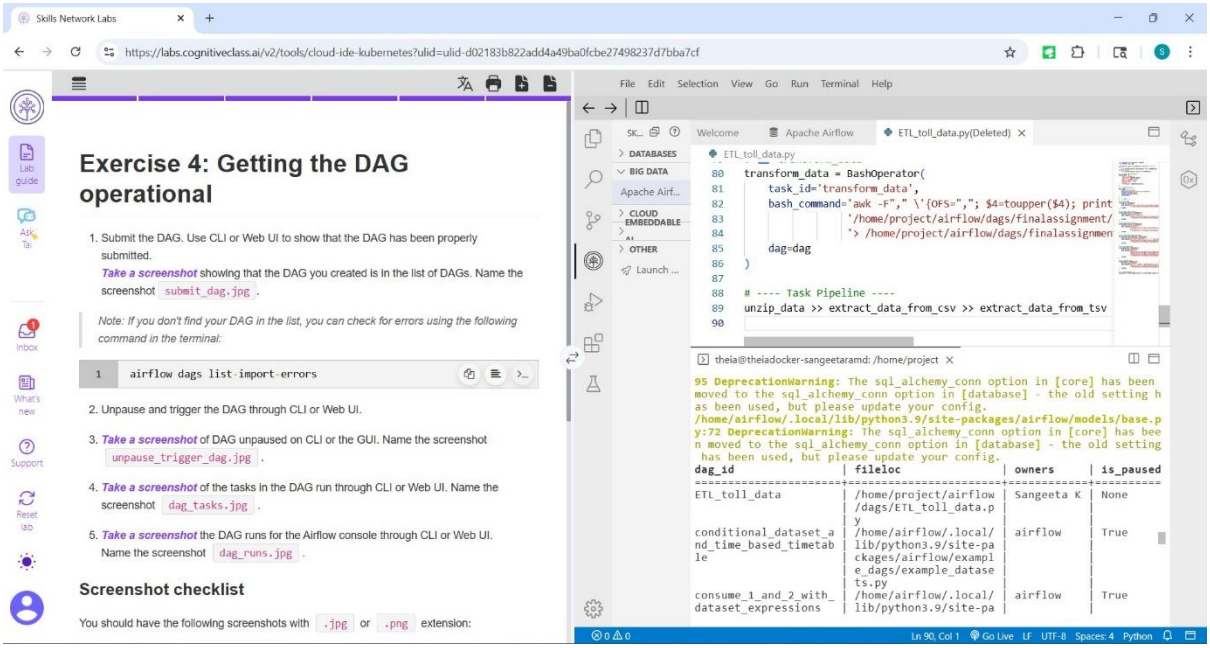

- Define the task pipeline as per the details given below:

| **Task** | **Functionality** |
|---|---|
| First task | unzip_data |
| Second task | extract_data_from_csv |
| Third task | extract_data_from_tsv |
| Fourth task | extract_data_from_fixed_width |
| Fifth task | consolidate_data |
| Sixth task | transform_data |

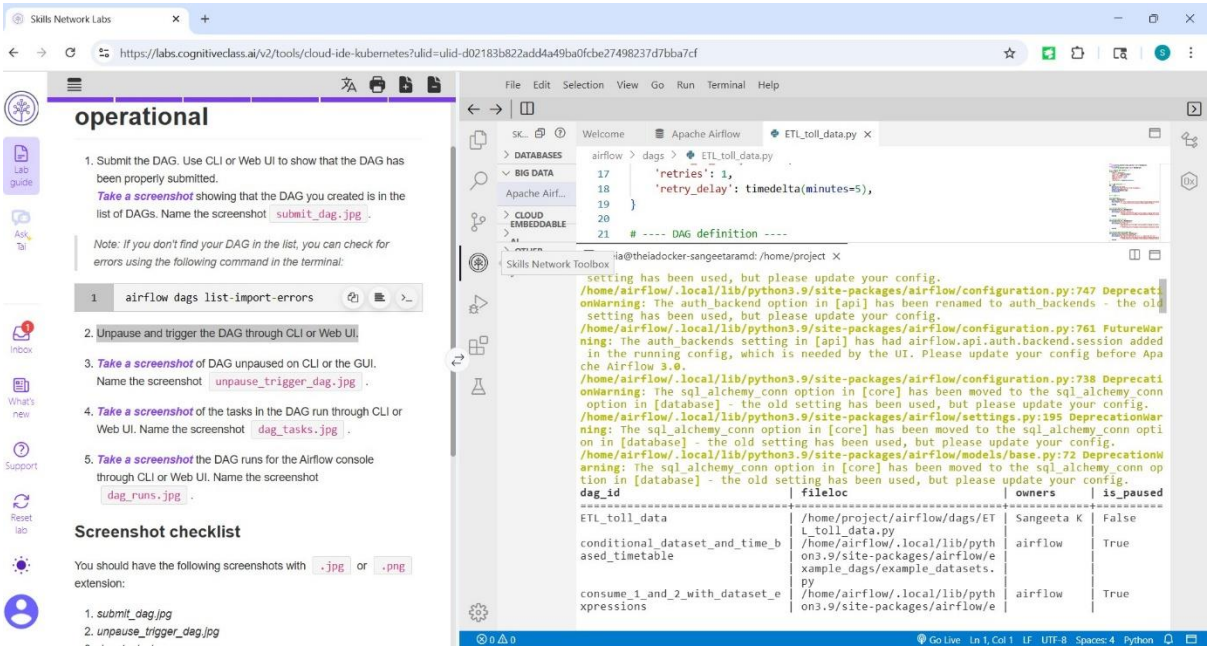

Skills Network Labs
https://labs.cognitiveclass.ai/v2/tools/cloud-ide-kubernetes?ulid=ulid-d02183b822add4a49ba0fcbe27498237d7bba7cf
File Edit Selection View Go Run Terminal Help
Second task
Third task
Fourth task
Fifth task
Sixth task
Take a screenshot of the task pipeline section of the DAG. Name the screenshot task_pipeline.jpg .
At the end of this exercise, you should have the following screenshots with .jpg or .png extension:
1. unzip_data.jpg
2. extract_data_from_csv.jpg
3. extract_data_from_tsv.jpg
Welcome
ETL_toll_data.py
Apache Airflow
DATABASES
BIG DATA
CLOUD EMBEDDABLE
OTHER
Launch ...
unzip_data >> extract_data_from_csv >> extract_data_from_tsv >> extract_data_from_fixed_width >> consolida
transform_data = BashOperator(
task_id='transform_data',
bash_command=(
"awk -F',' '{OFS=\",\"; $4=toupper($4); print $0}' "
"/home/project/airflow/dags/finalassignment/staging/extracted_data.csv "
"> /home/project/airflow/dags/finalassignment/staging/transformed_data.csv"
),
)
unzip_data >> extract_data_from_csv >> extract_data_from_tsv >> extract_data_from_fixed_width >>
consolidate_data >> transform_data
theia@theiadocker-sangeetaramd: /home/project
/home/project/airflow/dags/finalassignment/staging/tsv_data.csv \
/home/project/airflow/dags/finalassignment/staging/fixed_width_data.csv \
> /home/project/airflow/dags/finalassignment/staging/extracted_data.csv
theia@theiadocker-sangeetaramd:/home/project$ ls -lh /home/project/airflow/dags/finalassignment/staging/extracted_data.csv
-rw-r--r-- 1 theia users 745K Oct 13 23:58 /home/project/airflow/dags/finalassignment/staging/extracted_data.csv
theia@theiadocker-sangeetaramd:/home/project$ awk -F',' '{OFS=",";$4=toupper($4); print $0}' /home/project/airflow/dags/finalassignment/staging/extracted_data.csv > /home/project/airflow/dags/finalassignment/staging/transformed_data.csv
theia@theiadocker-sangeetaramd:/home/project$ ls /home/project/airflow/dags/finalassignment/staging/transformed_data.csv
/home/project/airflow/dags/finalassignment/staging/transformed_data.csv
theia@theiadocker-sangeetaramd:/home/project$
Ln 78, Col 5
Go Live
LF
UTF-8
Spaces: 4
Python

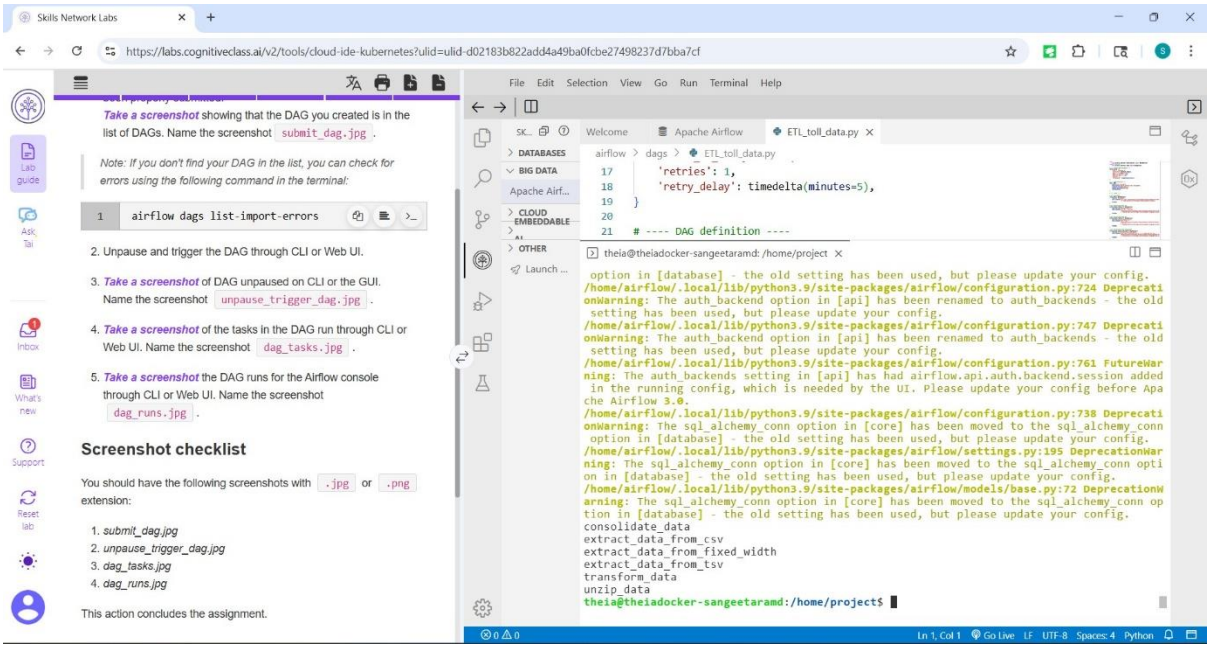
## Task 4: Getting the DAG operational

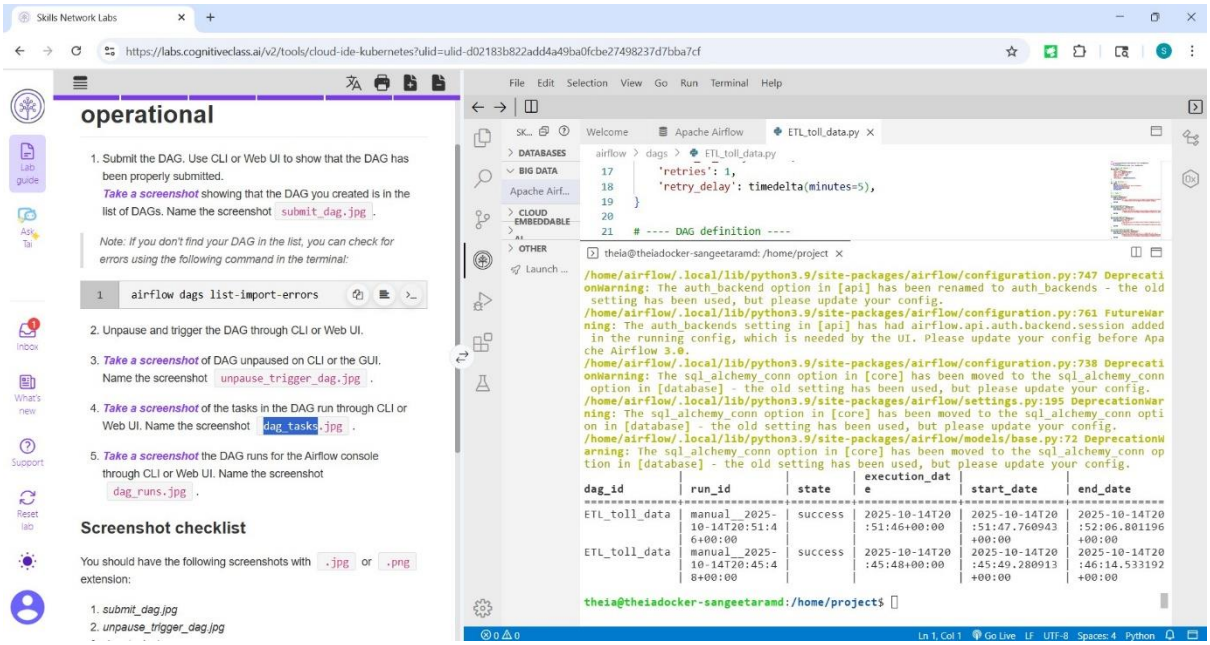
- Submit the DAG. Use CLI or Web UI to show that the DAG has been properly submitted.



- Unpause and trigger the DAG through CLI or Web UI.




Skills Network Labs
https://labs.cognitiveclass.ai/v2/tools/cloud-ide-kubernetes?ulid=ulid-d02183b822add4a49ba0fcbe27498237d7bba7cf
operational
1. Submit the DAG. Use CLI or Web UI to show that the DAG has been properly submitted.
Take a screenshot showing that the DAG you created is in the list of DAGs. Name the screenshot submit_dag.jpg .
Note: If you don't find your DAG in the list, you can check for errors using the following command in the terminal:
airflow dags list-import-errors
2. Unpause and trigger the DAG through CLI or Web UI.
3. Take a screenshot of DAG unpaused on CLI or the GUI. Name the screenshot unpause_trigger_dag.jpg .
4. Take a screenshot of the tasks in the DAG run through CLI or Web UI. Name the screenshot dag_tasks.jpg .
5. Take a screenshot the DAG runs for the Airflow console through CLI or Web UI. Name the screenshot dag_runs.jpg .
Screenshot checklist
You should have the following screenshots with .jpg or .png extension:
1. submit_dag.jpg
2. unpause_trigger_dag.jpg
File Edit Selection View Go Run Terminal Help
Welcome
Apache Airflow
ETL_toll_data.py
'retries': 1,
'retry_delay': timedelta(minutes=5),
}
# ---- DAG definition ----
theia@theiadocker-sangeetaramd: /home/project
/home/airflow/.local/lib/python3.9/site-packages/airflow/configuration.py:747 DeprecationWarning: The auth_backend option in [api] has been renamed to auth_backends - the old setting has been used, but please update your config.
/home/airflow/.local/lib/python3.9/site-packages/airflow/configuration.py:761 FutureWarning: The auth_backends setting in [api] has had airflow.api.auth.backend.session added in the running config, which is needed by the UI. Please update your config before Apache Airflow 3.0.
/home/airflow/.local/lib/python3.9/site-packages/airflow/configuration.py:738 DeprecationWarning: The sql_alchemy_conn option in [core] has been moved to the sql_alchemy_conn option in [database] - the old setting has been used, but please update your config.
/home/airflow/.local/lib/python3.9/site-packages/airflow/settings.py:195 DeprecationWarning: The sql_alchemy_conn option in [core] has been moved to the sql_alchemy_conn option in [database] - the old setting has been used, but please update your config.
/home/airflow/.local/lib/python3.9/site-packages/airflow/models/base.py:72 DeprecationWarning: The sql_alchemy_conn option in [core] has been moved to the sql_alchemy_conn option in [database] - the old setting has been used, but please update your config.
dag_id | run_id | state | execution_date | start_date | end_date
ETL_toll_data | manual__2025-10-14T20:51:46+00:00 | success | 2025-10-14T20:51:47.760943+00:00 | 2025-10-14T20:52:06.801196+00:00
ETL_toll_data | manual__2025-10-14T20:45:48+00:00 | success | 2025-10-14T20:45:49.280913+00:00 | 2025-10-14T20:46:14.533192+00:00
theia@theiadocker-sangeetaramd:/home/project$