Final Assignment: Build an ETL Pipeline using Airflow

Exercise 1 - Prepare the lab environment

1. Start Apache Airflow.
2. Open a terminal and create a directory structure for staging area as follows: /home/project/airflow/dags/finalassignment/staging.
3. Download the dataset from the source to the destination mentioned below using wget command.
   Source : https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Final%20Assignment/tolldata.tgz
4. Download the dataset from the source to the destination mentioned below using wget command.
   cd /home/project/airflow/dags/finalassignment/staging

Exercise 2 - Create a DAG

Task 1.1 - Define DAG arguments

Task 1.2 - Define the DAG

Task 1.3 - Create a task to unzip data

- Create a task named unzip_data.
- Use the downloaded data from the url given in the first part of this assignment in exercise 1 and uncompress it into the destination directory.

Task 1.4 - Create a task to extract data from csv file

- Create a task named extract_data_from_csv.
- This task should 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.

Task 1.5 - Create a task to extract data from tsv file

- Create a task named extract_data_from_tsv.
- This task should 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.

Task 1.6 - Create a task to extract data from fixed width file

- Create a task named extract_data_from_fixed_width.
- This task should 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.

Task 1.7 - Create a task to consolidate data extracted from previous tasks

- Create a task named consolidate_data.
- 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, and Vehicle Code

- Hint: Use the bash paste command.
  paste command merges lines of files.
  Example : paste file1 file2 > newfile
- The above command merges the columns of the files file1 and file2 and sends the output to newfile.
- You can use the command man paste to explore more.

Task 1.8 - Transform and load the data

- Create a task named transform_data.
- This task should 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.

Task 1.9 - Define the task pipeline

Exercise 3 - Getting the DAG operational.

Task 1.10 - Submit the DAG

Task 1.11 - Unpause the DAG

Task 1.12 - Monitor the DAG

```
# import the libraries
from datetime import timedelta
# The DAG object; we'll need this to instantiate a DAG
from airflow import DAG
# Operators; we need this to write tasks!
from airflow.operators.bash_operator import BashOperator
# This makes scheduling easy
from airflow.utils.dates import days_ago

#defining DAG arguments

# You can override them on a per-task basis during operator initialization
default_args = {
    'owner': 'DM',
    'start_date': days_ago(0),
    'email': ['dm@email.com'],
    'email_on_failure': True,
    'email_on_retry': True,
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
}

# define the DAG
dag = DAG(
    dag_id='ETL_toll_data',
    default_args=default_args,
    description='Apache Airflow Final Assignment',
    schedule_interval= timedelta(minutes=1),
)

# define the tasks

# define task named unzip_data
unzip_data = BashOperator(
    task_id='unzip_data',
    bash_command='tar -xvzf /home/project/airflow/dags/finalassignment/staging/tolldata.tgz -
C /home/project/airflow/dags/finalassignment/staging',
    dag=dag,
)

# define task named extract_data_from_csv
extract_data_from_csv = BashOperator(
    task_id='extract_data_from_csv',
    bash_command='cut -d"," -f1-4 /home/project/airflow/dags/finalassignment/staging/vehicle-
data.csv > /home/project/airflow/dags/finalassignment/staging/csv_data.csv',
    dag=dag,
)

# define task named extract_data_from_tsv
extract_data_from_tsv = BashOperator(
```

```
    task_id='extract_data_from_tsv',
    bash_command='cut -f5-7 /home/project/airflow/dags/finalassignment/staging/tollplaza-
data.tsv| tr -d "\r" | tr "[:blank:]" "," >
/home/project/airflow/dags/finalassignment/staging/tsv_data.csv',
    dag=dag,
)

# define task named extract_data_from_fixed_width
extract_data_from_fixed_width = BashOperator(
    task_id='extract_data_from_fixed_width',
    bash_command='sudo cut -c 59-67
/home/project/airflow/dags/finalassignment/staging/payment-data.txt| tr " " "," >
/home/project/airflow/dags/finalassignment/staging/fixed_width_data.csv',
    dag=dag,
)

# define task named consolidate_data
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| tr -d "\r"| tr " "
","',
    dag=dag,
)

# define task named transform_data
transform_data = BashOperator(
    task_id='transform_data',
    bash_command='cut -d"," -f4
/home/project/airflow/dags/finalassignment/staging/extracted_data.csv| tr [a-z] [A-Z] <
/home/project/airflow/dags/finalassignment/staging/extracted_data.csv >\
     /home/project/airflow/dags/finalassignment/staging/transformed_data.csv',
    dag=dag,
)


#define task pipeline
unzip_data >> extract_data_from_csv >> extract_data_from_tsv >> extract_data_from_fixed_width
>> consolidate_data >> transform_data
```

```
theia@theiadocker-delacruzdan1:/home/project$ sudo cp ETL_toll_data.py $AIRFLOW_HOME/dags
theia@theiadocker-delacruzdan1:/home/project$ airflow dags list|grep "ETL_toll_data"
ETL_toll_data                            | ETL_toll_data.py
    | DM       | True
```

```
theia@theiadocker-delacruzdan1:/home/project$ airflow dags unpause ETL_toll_data
/home/airflow/.local/lib/python3.7/site-packages/airflow/configuration.py:528: 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.
  option = self._get_environment_variables(deprecated_key, deprecated_section, key, section)
/home/airflow/.local/lib/python3.7/site-packages/airflow/configuration.py:528: DeprecationWarning: The auth_backend option in [api] has been renamed to
auth_backends - the old setting has been used, but please update your config.
  option = self._get_environment_variables(deprecated_key, deprecated_section, key, section)
/home/airflow/.local/lib/python3.7/site-packages/airflow/configuration.py:360: FutureWarning: The auth_backends setting in [api] has had airflow.api.au
h.backend.session added in the running config, which is needed by the UI. Please update your config before Apache Airflow 3.0.
  FutureWarning,
Dag: ETL_toll_data, paused: False
```