Practice exercises

Problem:

Write a DAG named ETL_Server_Access_Log_Processing.

Task 1: Create the imports block.

Task 2: Create the DAG Arguments block. You can use the default settings

Task 3: Create the DAG definition block. The DAG should run daily.

Task 4: Create the download task.

download task must download the server access log file which is available at the URL: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Apache%20Airflow/Build%20a%20DAG%20using%20Airflow/web-server-access-log.txt

Task 5: Create the extract task.

The server access log file contains these fields.

a. timestamp - TIMESTAMP

b. latitude - float

c. longitude - float

d. visitorid - char(37)

e. accessed_from_mobile - boolean

f. browser_code - int

The extract task must extract the fields timestamp and visitorid.

Task 6: Create the transform task.

The transform task must capitalize the visitorid.

Task 7: Create the load task.

The load task must compress the extracted and transformed data.

Task 8: Create the task pipeline block.

The pipeline block should schedule the task in the order listed below:

- download
- extract
- transform
- load

Task 10: Submit the DAG.

Task 11. Verify if the DAG is submitted

```
# 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': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
}

# defining the DAG

# define the DAG
dag = DAG(
    'ETL_Server_Access_Log_Processing',
    default_args=default_args,
    description='ETL_Server_Access_Log_Processing',
    schedule_interval=timedelta(days=1),
)

# define the tasks

# define the task 'download'

download = BashOperator(
    task_id='download',
    bash_command='wget "https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBM-DB0250EN-
SkillsNetwork/labs/Apache%20Airflow/Build%20a%20DAG%20using%20Airflow/web-server-access-
log.txt"',
    dag=dag,
)

# define the task 'extract'

extract = BashOperator(
    task_id='extract',
```

```
    bash_command='cut -f1,4 -d"#" web-server-access-log.txt >
/home/project/airflow/dags/extracted.txt',
    dag=dag,
)

# define the task 'transform'

transform = BashOperator(
    task_id='transform',
    bash_command='tr "[a-z]" "[A-Z]" < /home/project/airflow/dags/extracted.txt >
/home/project/airflow/dags/capitalized.txt',
    dag=dag,
)

# define the task 'load'

load = BashOperator(
    task_id='load',
    bash_command='zip log.zip capitalized.txt' ,
    dag=dag,
)

# task pipeline

download >> extract >> transform >> load
```

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