```
default_args = {
    'owner': 'Ramesh Sannareddy',
    'start_date': days_ago(0),
    'email': ['ramesh@somemail.com'],
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
}

# defining the DAG

# define the DAG
dag = DAG(
    'my-first-dag',
    default_args=default_args,
    description='My first DAG',
    schedule_interval=timedelta(days=1),
)

# define the tasks

# define the first task

extract = BashOperator(
    task_id='extract',
    bash_command='cut -d":" -f1,3,6 /etc/passwd > extracted-data.txt',
    dag=dag,
)


# define the second task
transform_and_load = BashOperator(
    task_id='transform',
    bash_command='tr ":" "," < extracted-data.txt > transformed-data.csv',
    dag=dag,
)


# task pipeline
extract >> transform_and_load
```

Copy the code above and save it into a file named `my_first_dag.py`

Exercise 5 - Submit a DAG

Submitting a DAG is as simple as copying the DAG python file into `dags` folder in the `AIRFLOW_HOME` directory.

Open a terminal and run the command below to submit the DAG that was created in the previous exercise.

```
cp my_first_dag.py $AIRFLOW_HOME/dags
```

Verify that our DAG actually got submitted.

Run the command below to list out all the existing DAGs.

```
airflow dags list
```

Verify that `my-first-dag` is a part of the output.

```
airflow dags list|grep "my-first-dag"
```

You should see your DAG name in the output.

Run the command below to list out all the tasks in `my-first-dag`.

```
airflow tasks list my-first-dag
```

You should see 2 tasks in the output.

Practice exercises

1. 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:

1. download
2. extract
3. transform
4. load

***Task 10**: Submit the DAG.*
***Task 11**. Verify if the DAG is submitted*

▼ Click here for Hint

Follow the example Python code given in the lab and make necessary changes to create the new DAG.

▶ Click here for Solution

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2021-07-05	0.1	Ramesh Sannareddy	Created initial version of the lab

Copyright (c) 2021 IBM Corporation. All rights reserved.

Previous