

Create and execute a Shell script from Airflow

Create a Basic Shell Script

Exercise 2 - Create a basic shell script

Here we will define a shell script `extract_transform_load.sh` which will define a pipeline of tasks such as

- extract
- transform
- load

For now, let the shell script have basic echo statements for `extract` , `transform` and `load` .

```
1  #!/bin/bash
2
3  echo "Extract data"
4
5  echo "Transform data"
6
7  echo "Load data"
```

In the next section we will define a `DAG` to call and execute the shell script.

- Following creating a basic shell script and starting an instance of airflow through docker

Explore Anatomy of DAG

An Apache Airflow DAG is a python program. It consists of these logical blocks.

- Imports
- DAG Arguments
- DAG Definition
- Task Definitions
- Task Pipeline

A typical `imports` block looks like this.

```
1  # import the libraries
2
3  from datetime import timedelta
4  # The DAG object; we'll need this to instantiate a DAG
5  from airflow import DAG
6  # Operators; we need this to write tasks!
7  from airflow.operators.bash_operator import BashOperator
8  # This makes scheduling easy
9  from airflow.utils.dates import days_ago
```

A typical `DAG Arguments` block looks like this.

```
1  #defining DAG arguments
2
3  # You can override them on a per-task basis during operator initialization
4  default_args = {
5      'owner': 'Ramesh Sannareddy',
6      'start_date': days_ago(0),
7      'email': ['ramesh@somemail.com'],
8      'email_on_failure': True,
9      'email_on_retry': True,
10     'retries': 1,
11     'retry_delay': timedelta(minutes=5),
12 }
```

DAG arguments are like settings for the DAG.

The above settings mention

- the owner name,
- when this DAG should run from: `days_age(0)` means today,
- the email address where the alerts are sent to,
- whether alert must be sent on failure,
- whether alert must be sent on retry,
- the number of retries in case of failure, and
- the time delay between retries.

A typical `DAG definition` block looks like this.

```
1 # define the DAG
2 dag = DAG(
3     dag_id='sample-etl-dag',
4     default_args=default_args,
5     description='Sample ETL DAG using Bash',
6     schedule_interval=timedelta(days=1),
7 )
```

Here we are creating a variable named `dag` by instantiating the DAG class with the following parameters.

`sample-etl-dag` is the ID of the DAG. This is what you see on the web console.

We are passing the dictionary `default_args`, in which all the defaults are defined.

`description` helps us in understanding what this DAG does.

`schedule_interval` tells us how frequently this DAG runs. In this case every day. (`days=1`).

A typical `task definitions` block looks like this:

```
1 # define the tasks
2
3 # define the task named extract_transform_and_load to call the shell script
4 extract_transform_and_load = BashOperator(
5     task_id='extract_transform_and_load',
6     bash_command='/home/project/airflow/dags/extract_transform_load.sh "',
7     dag=dag,
8 )
```

A task is defined using:

- A task_id which is a string and helps in identifying the task.
- What bash command it represents. Here we are calling the shell script `extract_transform_load.sh` which we previously defined
- Which dag this task belongs to.

A typical `task pipeline` block looks like this:

```
1 # task pipeline
2 extract_transform_and_load
```

When we execute the task `extract_transform_and_load` the code in the shell script gets executed.

Exercise 4 - ETL process on a /etc/passwd file

Here we will first

- Create a new shell script called `my_first_dag.sh` to perform the ETL process.
- Create a `DAG` file `my_first_dag.py` which will run daily and defines a task `etl` to call the shell script `my_first_etl.sh`.
- Submit the `DAG`

Create a new shell script `my_first_dag.sh` by selecting **File->New File**.

- The shell script extracts the first, third and sixth fields from `/etc/passwd` file using the `cut` command and writes to a new file `extracted-data.txt`
- Next the `extracted-data.txt` is transformed to a `csv` file and loaded into a new file called `transformed-data.csv` using `tr` command.

Copy the code below in the shell script.

```
1 #!/bin/bash
2 echo "extract_transform_and_load"
3 cut -d":" -f1,3,6 /etc/passwd > /home/project/airflow/dags/extracted-data.txt
4
5 tr ":" " ," < /home/project/airflow/dags/extracted-data.txt > /home/project/ai w/
```

Click `Save` to save the shell script.

Create a new `DAG` file `my_first_dag.py` by selecting **File->New File**.

This DAG has one task `etl` that calls the shell script `my_first_dag.sh`

```
1  # import the libraries
2
3  from datetime import timedelta
4  # The DAG object; we'll need this to instantiate a DAG
5  from airflow import DAG
6  # Operators; we need this to write tasks!
7  from airflow.operators.bash_operator import BashOperator
8  # This makes scheduling easy
9  from airflow.utils.dates import days_ago
10
11 #defining DAG arguments
12
13 # You can override them on a per-task basis during operator initialization
14 default_args = {
15     'owner': 'Ramesh Sannareddy',
16     'start_date': days_ago(0),
17     'email': ['ramesh@somemail.com'],
18     'email_on_failure': False,
19     'email_on_retry': False,
20     'retries': 1,
21     'retry_delay': timedelta(minutes=5),
22 }
23
24 # defining the DAG
25
```

```
25
26 # define the DAG
27 dag = DAG(
28     'my-first-dag',
29     default_args=default_args,
30     description='My first DAG',
31     schedule_interval=timedelta(days=1),
32 )
33
34 # define the task **extract_transform_and_load** to call shell script
35
36 #calling the shell script
37 extract_transform_load = BashOperator(
38     task_id="extract_transform_load",
39     bash_command="/home/project/airflow/dags/my_first_dag.sh ",
40     dag=dag,
41 )
42
43 # task pipeline
44 extract_transform_load
```



Submit a DAG

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.

Note: While submitting the dag that was created in the previous exercise, use **sudo** in the terminal before the command used to submit the dag.

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



Next, run the command below one by one to submit shell script in the dags folder and to change the permission for reading shell script.

```
1 cp my_first_dag.sh $AIRFLOW_HOME/dags
2 cd airflow/dags
3 chmod 777 my_first_dag.sh
```



Verify that our DAG actually got submitted.

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

```
1 airflow dags list
```



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

```
1 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`.

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



You should see 1 task in the output.

Shell Output for Above Steps

```

theia@theiadocker-craigtrupp8:/home/project$ cp my_first_dag.py
$AIRFLOW_HOME/dags
theia@theiadocker-craigtrupp8:/home/project$ cp my_first_dag_1.py
cp: missing destination file operand after 'my_first_dag_1.py'
Try 'cp --help' for more information.
theia@theiadocker-craigtrupp8:/home/project$ cp my_first_dag_1.py
$AIRFLOW_HOME/dags
theia@theiadocker-craigtrupp8:/home/project$ cp my_first_dag.sh
$AIRFLOW_HOME/dags
theia@theiadocker-craigtrupp8:/home/project$ cd airflow/dags
theia@theiadocker-craigtrupp8:/home/project/airflow/dags$ chmod 777
my_first_dag.sh
theia@theiadocker-craigtrupp8:/home/project/airflow/dags$ airflow dags list
/home/airflow/.local/lib/python3.7/site-packages/airflow/configuration.py:5
28: 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:5
28: 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:3
60: 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.
    FutureWarning,
Error: Failed to load all files. For details, run `airflow dags
list-import-errors`
dag_id | filepath
| owner | paused
=====+=====
example_bash_operator |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True
|
xample_dags/example_bash_operator.py | |
example_branch_datetime_operator |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |

```



```

True
|
xample_dags/example_branch_datetime_operator.py | |
example_branch_datetime_operator_2 | |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True
|
xample_dags/example_branch_datetime_operator.py | |
example_branch_dop_operator_v3 | |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True
|
xample_dags/example_branch_python_dop_operator_3.py | |
example_branch_labels | |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True
|
xample_dags/example_branch_labels.py | |
example_branch_operator | |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True
|
xample_dags/example_branch_operator.py | |
example_branch_python_operator_decorator | |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True
|
xample_dags/example_branch_operator_decorator.py | |
example_complex | |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True
| xample_dags/example_complex.py
| |
example_dag_decorator | |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True
|
xample_dags/example_dag_decorator.py | |
example_external_task_marker_child | |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True
|
xample_dags/example_external_task_marker_dag.py | |

```

example_external_task_marker_parent			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_external_task_marker_dag.py			
example_kubernetes_executor			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_kubernetes_executor.py			
example_local_kubernetes_executor			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_local_kubernetes_executor.py			
example_nested_branch_dag			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_nested_branch_dag.py			
example_passing_params_via_test_command			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_passing_params_via_test_command.py			
example_python_operator			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_python_operator.py			
example_short_circuit_operator			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_short_circuit_operator.py			
example_skip_dag			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
		example_dags/example_skip_dag.py	
example_sla_dag			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			

		sample_dags/example_sla_dag.py	
example_subdag_operator			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_subdag_operator.py			
example_subdag_operator.section-1			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_subdag_operator.py			
example_subdag_operator.section-2			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_subdag_operator.py			
example_task_group			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_task_group.py			
example_task_group_decorator			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_task_group_decorator.py			
example_time_delta_sensor_async			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_time_delta_sensor_async.py			
example_trigger_controller_dag			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_trigger_controller_dag.py			
example_trigger_target_dag			
/home/airflow/.local/lib/python3.7/site-packages/airflow/e		airflow	
True			
xample_dags/example_trigger_target_dag.py			
example_weekday_branch_operator			

```

/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True

|
xample_dags/example_branch_day_of_week_operator.py | |
example_xcom |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True

| xample_dags/example_xcom.py
| |
example_xcom_args |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True

| xample_dags/example_xcomargs.py
| |
example_xcom_args_with_operators |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True

| xample_dags/example_xcomargs.py
| |
latest_only |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True

|
xample_dags/example_latest_only.py | |
latest_only_with_trigger |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True

|
xample_dags/example_latest_only_with_trigger.py | |
my-first-dag | my_first_dag_1.py
| Darth Vader | True
tutorial |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True

| xample_dags/tutorial.py
| |
tutorial_etl_dag |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True

| xample_dags/tutorial_etl_dag.py
| |
tutorial_taskflow_api_etl |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |

```

True

```
sample_dags/tutorial_taskflow_api_etl.py |
tutorial_taskflow_api_etl_virtualenv |
/home/airflow/.local/lib/python3.7/site-packages/airflow/e | airflow |
True
sample_dags/tutorial_taskflow_api_etl_virtualenv.py |
```

```
theia@theiadocker-craigtrupp8:/home/project/airflow/dags$ airflow dags list
| grep 'my-first-dag-1'
theia@theiadocker-craigtrupp8:/home/project/airflow/dags$ airflow dags list
| grep 'my-first-dag'
my-first-dag | my_first_dag_1.py
| Darth Vader | True
```

```
theia@theiadocker-craigtrupp8:/home/project/airflow/dags$ airflow tasks
lits my-first-dag
/home/airflow/.local/lib/python3.7/site-packages/airflow/configuration.py:5
28: 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:5
28: 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:3
60: 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.
    FutureWarning,
usage: airflow tasks [-h] COMMAND ...
```

Manage tasks

positional arguments:

COMMAND

clear	Clear a set of task instance, as if they never ran
failed-deps	Returns the unmet dependencies for a task instance

list	List the tasks within a DAG
render	Render a task instance's template(s)
run	Run a single task instance
state	Get the status of a task instance
states-for-dag-run	Get the status of all task instances in a dag run
test	Test a task instance

optional arguments:

-h, --help show this help message and exit

airflow tasks command error: argument COMMAND: invalid choice: 'lits' (choose from 'clear', 'failed-deps', 'list', 'render', 'run', 'state', 'states-for-dag-run', 'test'), see help above.

```
theia@theiadocker-craigtrupp8:/home/project/airflow/dags$ airflow tasks list
```

```
/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.auth.backend.session added in the running config, which is needed by the UI. Please update your config before Apache Airflow 3.0.
```

```
FutureWarning,
```

```
usage: airflow tasks list [-h] [-S SUBDIR] [-t] [-v] dag_id
```

List the tasks within a DAG

positional arguments:

dag_id The id of the dag

optional arguments:

-h, --help show this help message and exit

-S SUBDIR, --subdir SUBDIR

File location or directory from which to look for the dag. Defaults to '[AIRFLOW_HOME]/dags' where [AIRFLOW_HOME] is the value you set for 'AIRFLOW_HOME' config you set in 'airflow.cfg'

-t, --tree Tree view
-v, --verbose Make logging output more verbose

airflow tasks list command error: the following arguments are required: dag_id, see help above.

theia@theiadocker-craigtrupp8:/home/project/airflow/dags\$ airflow tasks list my-first-dag

/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.auth.backend.session added in the running config, which is needed by the UI. Please update your config before Apache Airflow 3.0.

FutureWarning,
extract_transform_load

theia@theiadocker-craigtrupp8:/home/project/airflow/dags\$