

BashOperator Demonstration

This document will guide you through the demonstration of the BashOperator in Session 2 of the Airflow module.

Prerequisites:

sample_bash.py(the code explained in the video)

What are we doing?

In this demonstration, we will be creating a DAG with one task. This will be a Bash task made using the BashOperator. The Purpose of this DAG is to check whether a user exists and if it doesn't create that user.

The final result of whether the user is created or exists should be printed/echoed

Please follow the instructions below:

1. Login to your EMR instance.
2. Activate the Python virtual environment using the following command:

source /home/hadoop/airflow/bin/activate

```
[hadoop@ip-172-31-33-82 ~]$ source /home/hadoop/airflow/bin/activate  
(airflow) [hadoop@ip-172-31-33-82 ~]$ |
```

3. Now you need to place the **sample_bash.py** file in the **/home/hadoop/airflow/dags** directory. (You can use WinSCP or create a new file and paste the code in that file)
4. To ensure that the file there are no issues/errors with the file is it considered good practice to compile the program using the following command:

python sample_bash.py

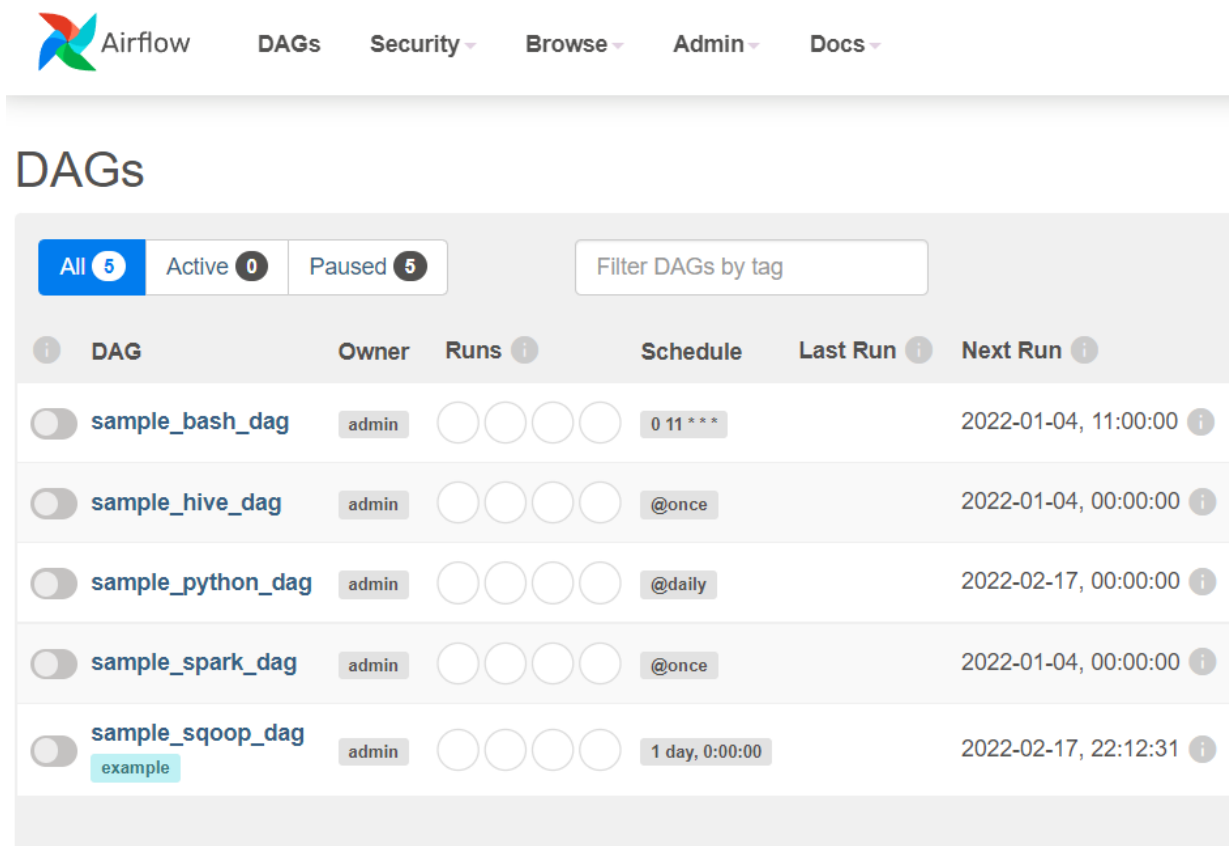
5. You can also use the following command to list the dags in your instance:

airflow dags list

6. Once you have made sure that your dag file has no issues you can go to the Airflow UI which is hosted in the URL : **your_public_dns:8082**

Note: You can find you_publin_ip in your AWS EMR dashboard (IPv4 Public DNS)

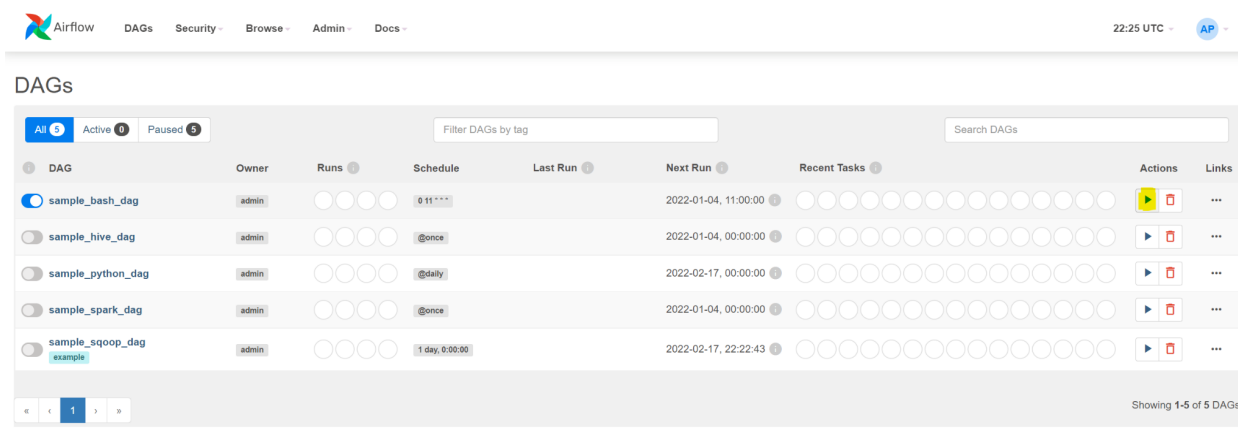
7. Switch ON the DAG(sample_bash_dag)



The screenshot shows the Airflow DAGs page. The top navigation bar includes links for Airflow, DAGs, Security, Browse, Admin, and Docs. The main heading is "DAGs". Below the heading, there are filters for "All" (5), "Active" (0), and "Paused" (5). A search bar labeled "Filter DAGs by tag" is also present. The table lists five DAGs:

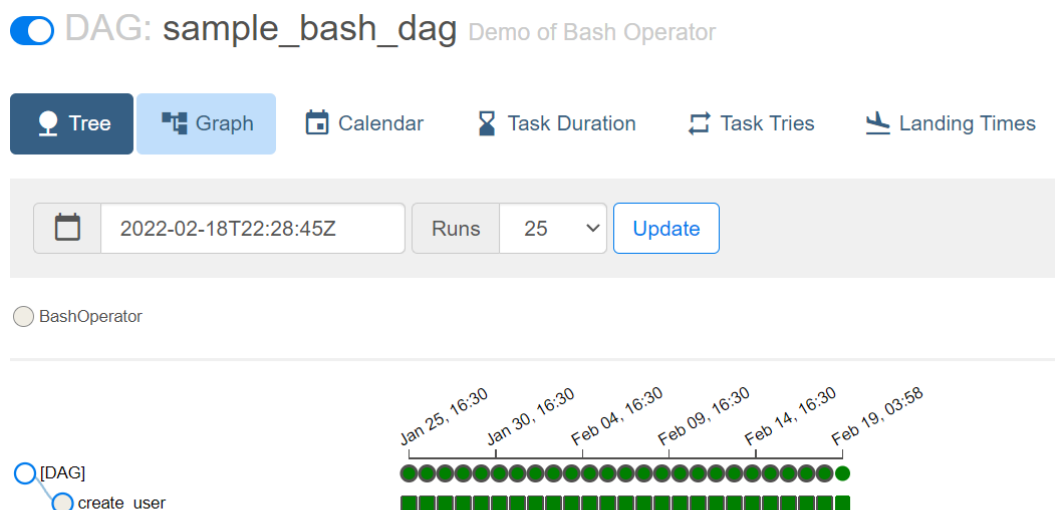
DAG	Owner	Runs	Schedule	Last Run	Next Run
<input type="checkbox"/> sample_bash_dag	admin	○ ○ ○ ○ ○	0 11 * * *		2022-01-04, 11:00:00
<input type="checkbox"/> sample_hive_dag	admin	○ ○ ○ ○ ○	@once		2022-01-04, 00:00:00
<input type="checkbox"/> sample_python_dag	admin	○ ○ ○ ○ ○	@daily		2022-02-17, 00:00:00
<input type="checkbox"/> sample_spark_dag	admin	○ ○ ○ ○ ○	@once		2022-01-04, 00:00:00
<input type="checkbox"/> sample_sqoop_dag example	admin	○ ○ ○ ○ ○	1 day, 0:00:00		2022-02-17, 22:12:31

8. After switching on the DAG, you need to start the DAG by clicking on the play button



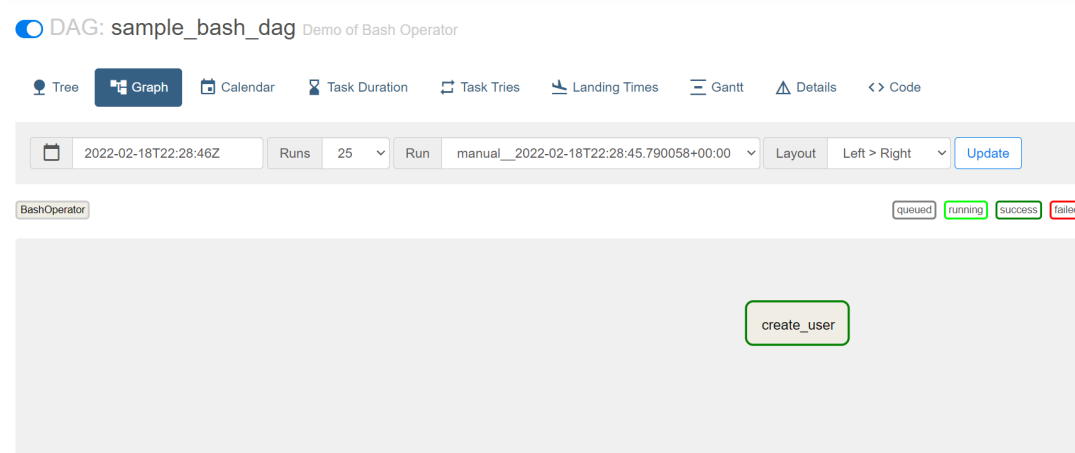
The screenshot shows the Airflow DAGs page after the "sample_bash_dag" has been switched on. The "All" filter now shows 5 DAGs, "Active" shows 1, and "Paused" shows 4. The "sample_bash_dag" is now marked as "Active" with a blue circle. The "Runs" column for "sample_bash_dag" shows a series of circles, indicating recent runs. The "Actions" column for "sample_bash_dag" shows a play button (▶) and a stop button (■). The "Recent Tasks" column for "sample_bash_dag" shows a series of circles, indicating recent tasks. The "sample_sqoop_dag" is marked as "example" with a teal circle.

9. Click on the sample_bash_dag to see the execution of your DAG



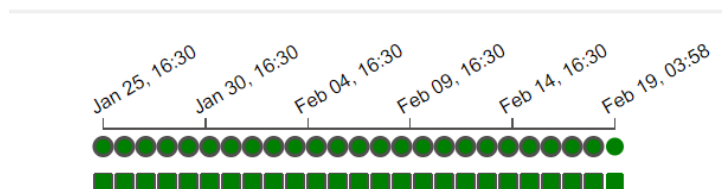
10. You can switch to different tabs to know more about your DAG

For example, you can see the graph view of our DAG below



11. You can verify the logs of the DAG run to see how the DAG has executed.

Just click on any task run (Green Squares) and click on View Logs



Task Instance: create_user
at: 2022-02-17, 11:00:00 UTC

Instance Details Rendered Log All Instances Filter Upstream

Download Log (by attempts):

1

Task Actions

Ignore All Deps Ignore Task State Ignore Task Deps Run

Past Future Upstream Downstream Recursive Failed Clear

Past Future Upstream Downstream Mark Failed

Past Future Upstream Downstream Mark Success

Close

```

session.merge(self)

[2022-02-18, 22:28:47 UTC] {warnings.py:110} WARNING - /home/hadoop/airflow/lib64/python3.7/site-packages/airflow/models/renderedtielfields.py:197: SAWarning: tuple(cls.dag_id, cls.task_id, cls.execution_date).notin_(subq2),

[2022-02-18, 22:28:47 UTC] {warnings.py:110} WARNING - /home/hadoop/airflow/lib64/python3.7/site-packages/airflow/models/renderedtielfields.py:198: SAWarning: ).delete(synchronize_session=False)

[2022-02-18, 22:28:47 UTC] {taskinstance.py:1426} INFO - Exporting the following env vars:
AIRFLOW_CTX_DAG_OWNER=admin
AIRFLOW_CTX_DAG_ID=sample_bash_dag
AIRFLOW_CTX_TASK_ID=create_user
AIRFLOW_CTX_EXECUTION_DATE=2022-02-18T22:28:45.790058+00:00
AIRFLOW_CTX_DAG_RUN_ID=manual__2022-02-18T22:28:45.790058+00:00
[2022-02-18, 22:28:47 UTC] {subprocess.py:62} INFO - Tmp dir root location:
/tmp
[2022-02-18, 22:28:47 UTC] {subprocess.py:74} INFO - Running command: ['bash', '-c', '\n id -u admin;\n if [ $? -eq 0 ]; \n then echo "User Ex
[2022-02-18, 22:28:47 UTC] {subprocess.py:85} INFO - Output:
[2022-02-18, 22:28:47 UTC] {subprocess.py:89} INFO - 979
[2022-02-18, 22:28:47 UTC] {subprocess.py:89} INFO - User Exists
[2022-02-18, 22:28:47 UTC] {subprocess.py:93} INFO - Command exited with return code 0
[2022-02-18, 22:28:47 UTC] {warnings.py:110} WARNING - /home/hadoop/airflow/lib64/python3.7/site-packages/airflow/models/xcom.py:104: SAWarning: TypeDeco
cls.key == key, cls.execution_date == execution_date, cls.task_id == task_id, cls.dag_id == dag_id

[2022-02-18, 22:28:47 UTC] {taskinstance.py:1277} INFO - Marking task as SUCCESS. dag_id=sample_bash_dag, task_id=create_user, execution_date=20220218T22
[2022-02-18, 22:28:47 UTC] {local_task_job.py:154} INFO - Task exited with return code 0
[2022-02-18, 22:28:47 UTC] {local_task_job.py:264} INFO - 0 downstream tasks scheduled from follow-on schedule check
[2022-02-18, 22:28:47 UTC] {warnings.py:110} WARNING - /home/hadoop/airflow/lib64/python3.7/site-packages/airflow/models/dag.py:1076: SADeprecationWarnin
return qry.value(DagModel.is_paused)

[2022-02-18, 22:28:47 UTC] {warnings.py:110} WARNING - /home/hadoop/airflow/lib64/python3.7/site-packages/airflow/models/dag.py:1076: SADeprecationWarnin
return qry.value(DagModel.is_paused)

```

12. You can switch off your DAG if you don't want it to run anymore.