Apache Airflow Installation Steps

Let’s start with the installation of the Apache Airflow. Now, if already have pip installed in your system, you can skip the first command. To install pip run the following command in the terminal.

```
sudo apt install python3-pip
```

Next airflow needs a home on your local system. By default ***~/airflow*** is the default location but you can change it as per your requirement.

```
export AIRFLOW_HOME=~/airflow
```

Now, install the apache airflow using the pip with the following command.

```
sudo pip3 install apache-airflow==1.10.12

sudo pip3 install SQLAlchemy==1.3.15
```

Airflow requires a database backend to run your workflows and to maintain them. Now, to initialize the database run the following command.

```
airflow db init
```

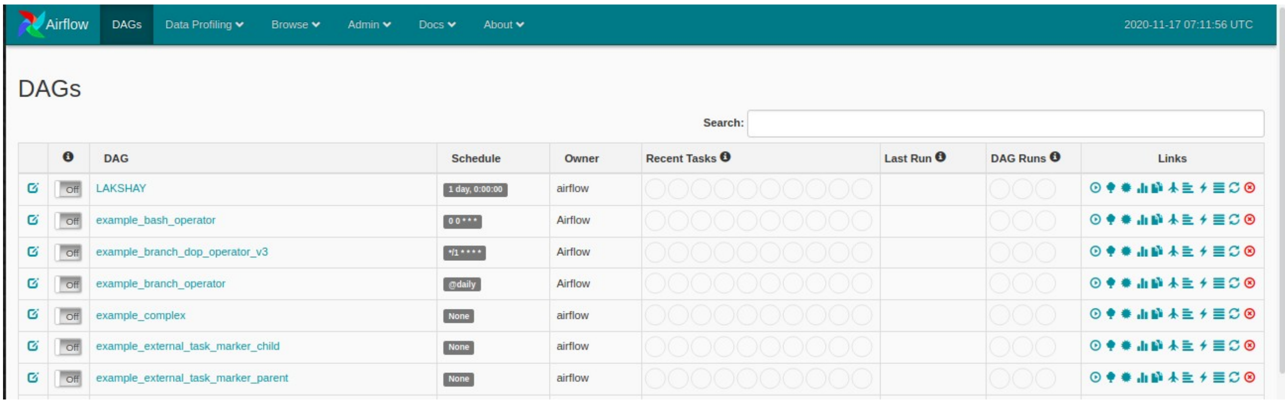
We have already discussed that airflow has an amazing user interface. To start the webserver run the following command in the terminal. The default port is 8080 and if you are using that port for something else then you can change it.

```
airflow webserver -p 8080
```

Now, start the airflow schedular using the following command in a different terminal. It will run all the time and monitor all your workflows and triggers them as you have assigned.

```
airflow scheduler
```

Now, create a folder name dags in the airflow directory where you will define your workflows or DAGs and open the web browser and go open: http://localhost:8080/admin/ and you will see something like this:



DAGs

Search:

	ℹ	DAG	Schedule	Owner	Recent Tasks ℹ	Last Run ℹ	DAG Runs ℹ	Links
	Off	LAKSHAY	1 day, 0:00:00	airflow				
	Off	example_bash_operator	0 0 * * *	Airflow				
	Off	example_branch_dop_operator_v3	*/1 * * * *	Airflow				
	Off	example_branch_operator	@daily	Airflow				
	Off	example_complex	None	airflow				
	Off	example_external_task_marker_child	None	airflow				
	Off	example_external_task_marker_parent	None	airflow				

Components of Apache Airflow

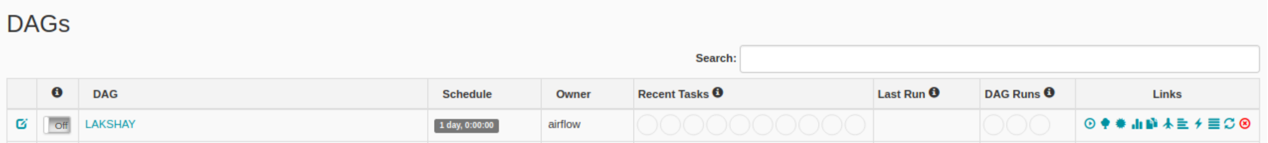
- **DAG**: It is the Directed Acyclic Graph – a collection of all the tasks that you want to run which is organized and shows the relationship between different tasks. It is defined in a python script.
- **Web Server**: It is the user interface built on the Flask. It allows us to monitor the status of the DAGs and trigger them.
- **Metadata Database**: Airflow stores the status of all the tasks in a database and do all read/write operations of a workflow from here.
- **Scheduler**: As the name suggests, this component is responsible for scheduling the execution of DAGs. It retrieves and updates the status of the task in the database.

User Interface

Now that you have installed the Airflow, let’s have a quick overview of some of the components of the user interface.

DAGS VIEW

It is the default view of the user interface. This will list down all the DAGS present in your system. It will give you a summarized view of the DAGS like how many times a particular DAG was run successfully, how many times it failed, the last execution time, and some other useful links.



DAGs

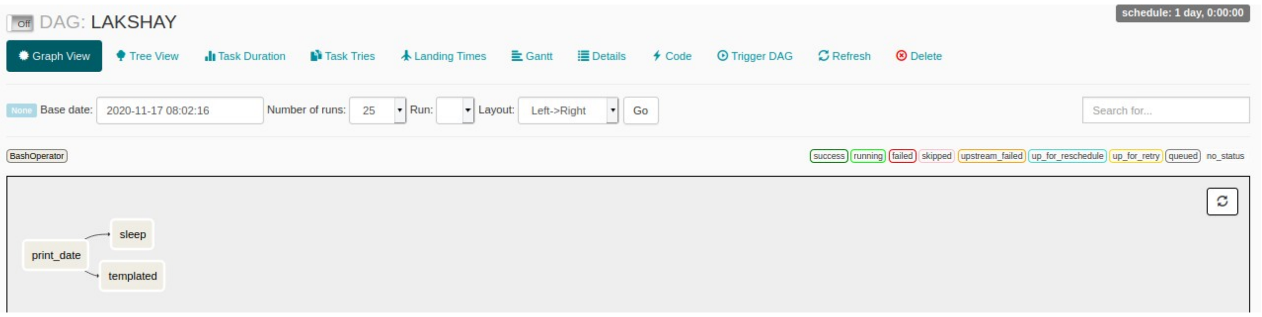
Search:

		DAG	Schedule	Owner	Recent Tasks	Last Run	DAG Runs	Links
	Off	LAKSHAY	1 day, 0:00:00	airflow				

GRAPH VIEW

In the graph view, you can visualize each and every step of your workflow with their dependencies and their current status. You can check the current status with different color codes like:

🟩	Task is successfully completed.
🟩	Task is in progress.
🟥	Task failed
🟪	Task has been skipped
🟨	Task failed once, executor is retrying
🟦	Task is in the queue.
⬜	No Status



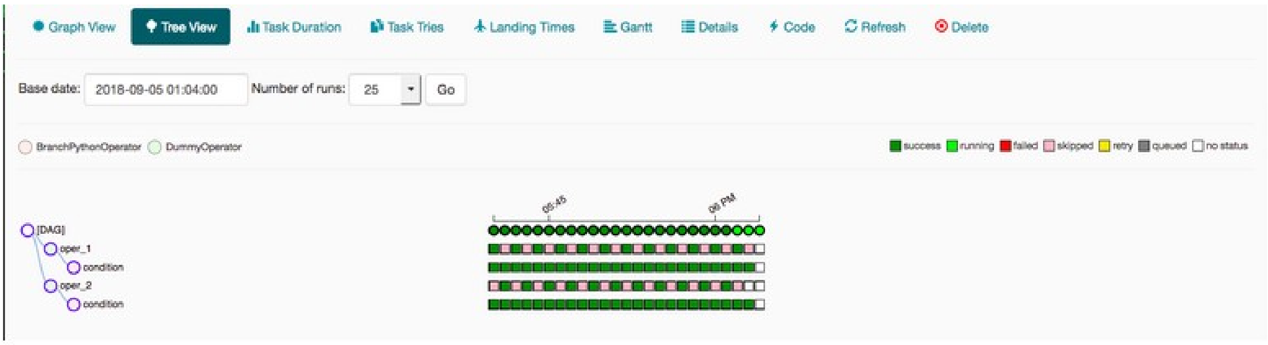
DAG: LAKSHAY

schedule: 1 day, 0:00:00

Base date: 2020-11-17 08:02:16 Number of runs: 25 Run: Layout: Left->Right

TREE VIEW

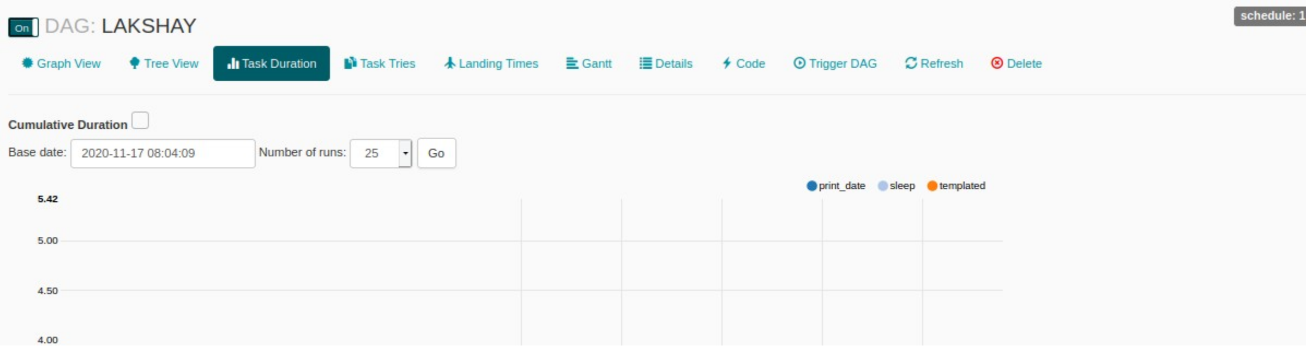
The tree view also represents the DAG. If you think your pipeline took a longer time to execute than expected then you can check which part is taking a long time to execute and then you can work on it.



Base date: 2018-09-05 01:04:00 Number of runs: 25

TASK DURATION

In this view, you can compare the duration of your tasks run at different time intervals. You can optimize your algorithms and compare your performance here.



DAG: LAKSHAY

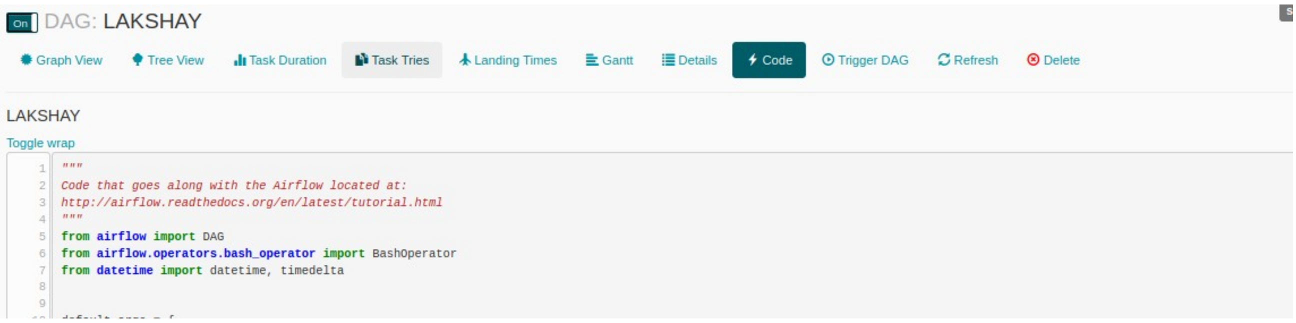
schedule: 1

Cumulative Duration □

Base date: 2020-11-17 08:04:09 Number of runs: 25

CODE

In this view, you can quickly view the code that was used to generate the DAG.



LAKSHAY

Toggle wrap

```
"""
Code that goes along with the Airflow located at:
http://airflow.readthedocs.org/en/latest/tutorial.html
"""
from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from datetime import datetime, timedelta


default_args = {
```