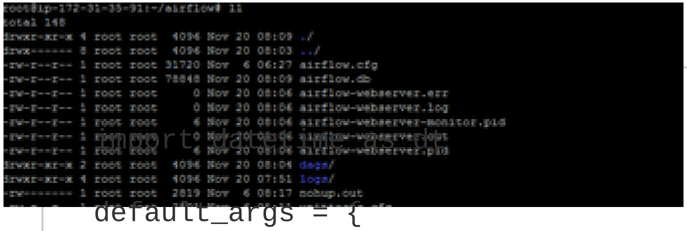
Airflow HelloWorld

Create your `dags_folder`, that is the directory where your DAG definition files will be stored in `AIRFLOW_HOME/dag.` Inside that directory create a file named **airflow_hello.py.**



```
root@ip-172-31-35-91:~/airflow# ll
total 148
drwxr-xr-x 4 root root  4096 Nov 20 08:09 ./
drwx------ 8 root root  4096 Nov 20 08:03 ../
-rw-r--r-- 1 root root 31720 Nov  6 06:27 airflow.cfg
-rw-r--r-- 1 root root 78848 Nov 20 08:09 airflow.db
-rw-r--r-- 1 root root     0 Nov 20 08:04 airflow-webserver.err
-rw-r--r-- 1 root root     0 Nov 20 08:04 airflow-webserver.log
-rw-r--r-- 1 root root     4 Nov 20 08:04 airflow-webserver-monitor.pid
-rw-r--r-- 1 root root     0 Nov 20 08:04 airflow-webserver.out
-rw-r--r-- 1 root root     4 Nov 20 08:04 airflow-webserver.pid
drwxr-xr-x 2 root root  4096 Nov 20 08:04 dags/
drwxr-xr-x 4 root root  4096 Nov 20 07:51 logs/
-rw------- 1 root root  2819 Nov  6 08:17 nohup.out
```

Configure common settings

```
import datetime as dt

default_args = {
    'owner': 'airflow',
    'start_date': dt.datetime(2019, 11, 20),
    'retries': 1,
    'retry_delay': dt.timedelta(minutes=5),
}
```

Create the DAG

Now create a DAG object that will contain tasks. Name it `airflow_hello_01`and pass it `default_args` .

```
with DAG('airflow_hello_01',
         default_args=default_args,
         schedule_interval='0 * * * *',
         ) as dag:
```

With `schedule_interval='0 0 * * *'` we've specified a run at every hour 0;Alternatively, you can use strings like `'@daily'` and `'@hourly'`.

Airflow will generate DAG runs from the `start_date` with the specified `schedule_interval`. Once a DAG is active, Airflow continuously checks in the database if all the DAG runs have successfully ran since the `start_date`. Any missing DAG runs are automatically scheduled.

Create the tasks

Tasks are represented by operators that either perform an action, transfer data, or sense if something has been done.We'll create a workflow consisting of three tasks: we'll print 'hello', wait for 10 seconds and finally print 'world'. The first two are done with the `BashOperator` and the latter with the `PythonOperator`.

```
from airflow.operators.bash_operator import BashOperator
    from airflow.operators.python_operator import PythonOperator

    def print_world():
        print('world')

    print_hello = BashOperator(task_id='print_hello',
                               bash_command='echo "hello"')
    sleep = BashOperator(task_id='sleep',
                         bash_command='sleep 5')
    print_world = PythonOperator(task_id='print_world',
                                 python_callable=print_world)
```

Sample Hello World code:

Final DAG final look like.

```
import datetime as dt

from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from airflow.operators.python_operator import PythonOperator


def print_world():
    print('world')


default_args = {
    'owner': 'airflow',
    'start_date': dt.datetime(2019, 11, 20),
    'retries': 1,
    'retry_delay': dt.timedelta(minutes=5),
}


with DAG('airflow_hello_01',
         default_args=default_args,
         schedule_interval='0 * * * *',
         ) as dag:

    print_hello = BashOperator(task_id='print_hello',
                               bash_command='echo "hello_world"')
    sleep = BashOperator(task_id='sleep',
                         bash_command='sleep 5')
    print_world = PythonOperator(task_id='print_world',
                                 python_callable=print_world)


print_hello
```

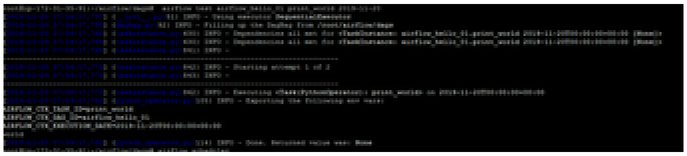
Test the DAG

First check that DAG file contains valid Python code by executing the file with Python:

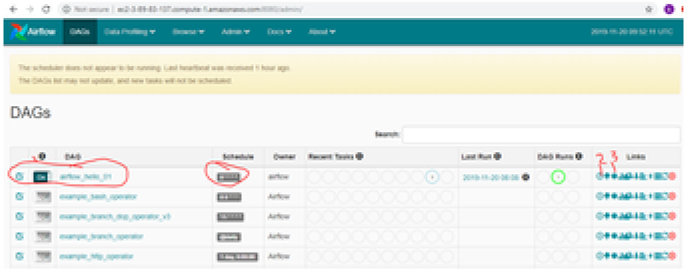
```
python airflow_hello_01
```

You can manually test a single task for a given `execution_date` with `airflow test`:

```
airflow test airflow_hello_01 print_world 2019-11-20
```



```
root@ip-172-31-35-91:~/airflow/dags# airflow test airflow_hello_01 print_world 2019-11-20
[illegible] {__init__.py:51} INFO - Using executor SequentialExecutor
[illegible] {dagbag.py:92} INFO - Filling up the DagBag from /root/airflow/dags
[illegible] {taskinstance.py:[illegible]} INFO - Dependencies all met for <TaskInstance: airflow_hello_01.print_world 2019-11-20T00:00:00+00:00 [None]>
[illegible] {taskinstance.py:[illegible]} INFO - Dependencies all met for <TaskInstance: airflow_hello_01.print_world 2019-11-20T00:00:00+00:00 [None]>
[illegible] {taskinstance.py:[illegible]} INFO -
--------------------------------------------------------------------------------
[illegible] {taskinstance.py:[illegible]} INFO - Starting attempt 1 of 1
[illegible] {taskinstance.py:[illegible]} INFO -
--------------------------------------------------------------------------------
[illegible] {taskinstance.py:[illegible]} INFO - Executing <Task(PythonOperator): print_world> on 2019-11-20T00:00:00+00:00
[illegible] {python_operator.py:[illegible]} INFO - Exporting the following env vars:
AIRFLOW_CTX_TASK_ID=print_world
AIRFLOW_CTX_DAG_ID=airflow_hello_01
AIRFLOW_CTX_EXECUTION_DATE=2019-11-20T00:00:00+00:00
world
[illegible] {python_operator.py:[illegible]} INFO - Done. Returned value was: None
```



Not secure | ec2-3-89-83-107.compute-1.amazonaws.com:8080/admin/

2019-11-20 09:52:11 UTC

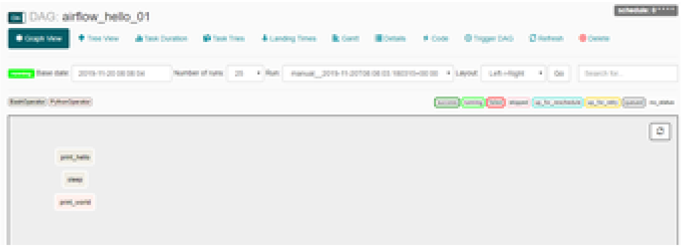
The scheduler does not appear to be running. Last heartbeat was received 1 hour ago.
The DAGs list may not update, and new tasks will not be scheduled.

DAGs

Search:

		DAG	Schedule	Owner	Recent Tasks	Last Run	DAG Runs	Links
	On	airflow_hello_01	[illegible]	airflow		2019-11-20 08:08		
		example_bash_operator	[illegible]	Airflow				
		example_branch_dop_operator_v3	[illegible]	Airflow				
		example_branch_operator	[illegible]	Airflow				
		example_http_operator	[illegible]	Airflow				

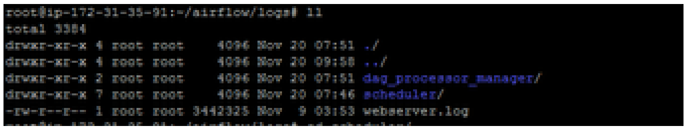
1
2
3



DAG: airflow_hello_01

schedule: [illegible]

running Base date: 2019-11-20 08:08:54 Number of runs: 25 Run: manual__2019-11-20T08:08:03.180210+00:00 Layout: Left->Right Go



```
root@ip-172-31-35-91:~/airflow/logs# ll
total 3384
drwxr-xr-x 4 root root    4096 Nov 20 07:51 ./
drwxr-xr-x 4 root root    4096 Nov 20 09:58 ../
drwxr-xr-x 2 root root    4096 Nov 20 07:51 dag_processor_manager/
drwxr-xr-x 7 root root    4096 Nov 20 07:46 scheduler/
-rw-r--r-- 1 root root 3442325 Nov  9 03:53 webserver.log
```

Go to Airflow WEBUI browser

In order to start a DAG run , First turn Workflow **ON** and then click **Trigger DAG** and finally Click **Graph View**

To check Logs `AIRFLOW_HOME/das/scheduler`