



Time to Code Your First DAG

A real ETL step by step



Your First Project

- The DAG:
 - The purpose of this DAG is to show you a typical ETL process which can be automated using Apache Airflow and Twitter. For simplicity, I'm not going to fetch data from the twitter API but if you want it, let me know.
 - It is composed of 4 tasks:
 - Fetching tweets
 - Cleaning tweets
 - Uploading tweets into HDFS
 - Loading data into HIVE
- For those who have never coded before, no worry, I will explain everything and the files which are not directly related to Apache Airflow will be provided.



The Project

- `cd ~`
- (from section 2, you should have the `airflow_files` directory)
- `source .sandbox/bin/activate`
 - If you don't see the prompt changing with `(.sandbox)`, type:
 - `rm -rf .sandbox`
 - `python3.6 -m venv .sandbox`
 - `source .sandbox/bin/activate`



The Project

- After having untar the archive you should have the following arborescence:

Don't worry if you don't have the .pyc files

```
(.sandbox) [airflow@localhost first_dag]$ tree
.
├── cleaning_tweet.py
├── data.csv
├── fetching_tweet.py
├── __pycache__
│   ├── cleaning_tweet.cpython-36.pyc
│   └── fetching_tweet.cpython-36.pyc
└── twitter.py

1 directory, 6 files
```



Let's Start Coding Our DAG

- `cd ~`
- `export PYTHONPATH=/home/airflow/airflow_files`
 - Don't forget to execute this command in each session (terminal) you use (where you run the web server, the scheduler etc).
- `vim airflow/dags/twitter.py`



The Code

You should take look at the code of `twitter.py` contained into the `first_dag` folder.

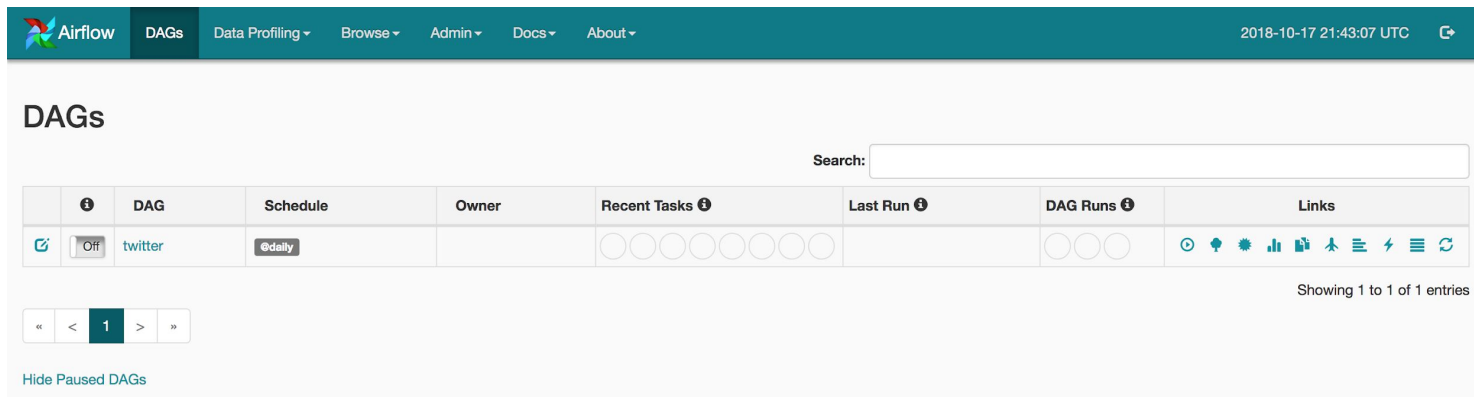


Running Airflow

- Once you DAG is created and placed into ~/airflow/dags don't forget to run the web server as well as the scheduler with the following commands:
- In a new terminal: `airflow webserver`
- In a new terminal: `airflow scheduler`

Airflow UI

- Once your Airflow UI is running you should see your dag as follow:



The screenshot displays the Airflow web interface. At the top is a teal navigation bar with the Airflow logo, a 'DAGs' tab, and several menu items: 'Data Profiling', 'Browse', 'Admin', 'Docs', and 'About'. The date and time '2018-10-17 21:43:07 UTC' are shown on the right. Below the navigation bar, the main content area is titled 'DAGs'. It features a search bar and a table of DAGs. The table has columns for 'DAG', 'Schedule', 'Owner', 'Recent Tasks', 'Last Run', 'DAG Runs', and 'Links'. A single DAG named 'twitter' is listed with a schedule of '@daily' and is currently in a 'Paused' state, indicated by a grey 'Off' button. The 'Recent Tasks' column shows eight empty circles. The 'DAG Runs' column shows three empty circles. The 'Links' column contains various icons for actions like refresh, pause, unpause, etc. At the bottom of the table, it says 'Showing 1 to 1 of 1 entries'. Below the table is a pagination control showing '1' of 1 entries. A link 'Hide Paused DAGs' is visible at the bottom left.

	DAG	Schedule	Owner	Recent Tasks	Last Run	DAG Runs	Links
	twitter	@daily					

Showing 1 to 1 of 1 entries

« < 1 > »

[Hide Paused DAGs](#)

Airflow UI

- If you click on your dag and go to “Graph View” you should see this:

The screenshot shows the Airflow web interface. The top navigation bar includes links for Airflow, DAGs, Data Profiling, Browse, Admin, Docs, and About, along with the date and time '2018-10-17 21:46:39 UTC'. A red banner at the top indicates 'No tasks found'. Below this, the DAG 'twitter' is selected, with a 'schedule: @daily' button. The 'Graph View' tab is active, showing various view options: Graph View, Tree View, Task Duration, Task Tries, Landing Times, Gantt, Details, Code, and Refresh. A search bar is present. The main area displays filters for 'Base date: 2018-10-17 00:31:15', 'Number of runs: 25', 'Run: [dropdown]', 'Layout: Left->Right', and a 'Go' button. A legend at the bottom right shows task statuses: success (green), running (blue), failed (red), skipped (yellow), retry (orange), queued (grey), and no status (white). A refresh button is located in the bottom right corner of the main area.



Final Note

- We basically have initialised our first DAG which is about getting tweets, clearing them and loading them into our HDFS.
- But, we still need to create our Tasks to inform Apache Airflow of how it should run our scripts and load data into HDFS.
- Let's do this in the next lesson...