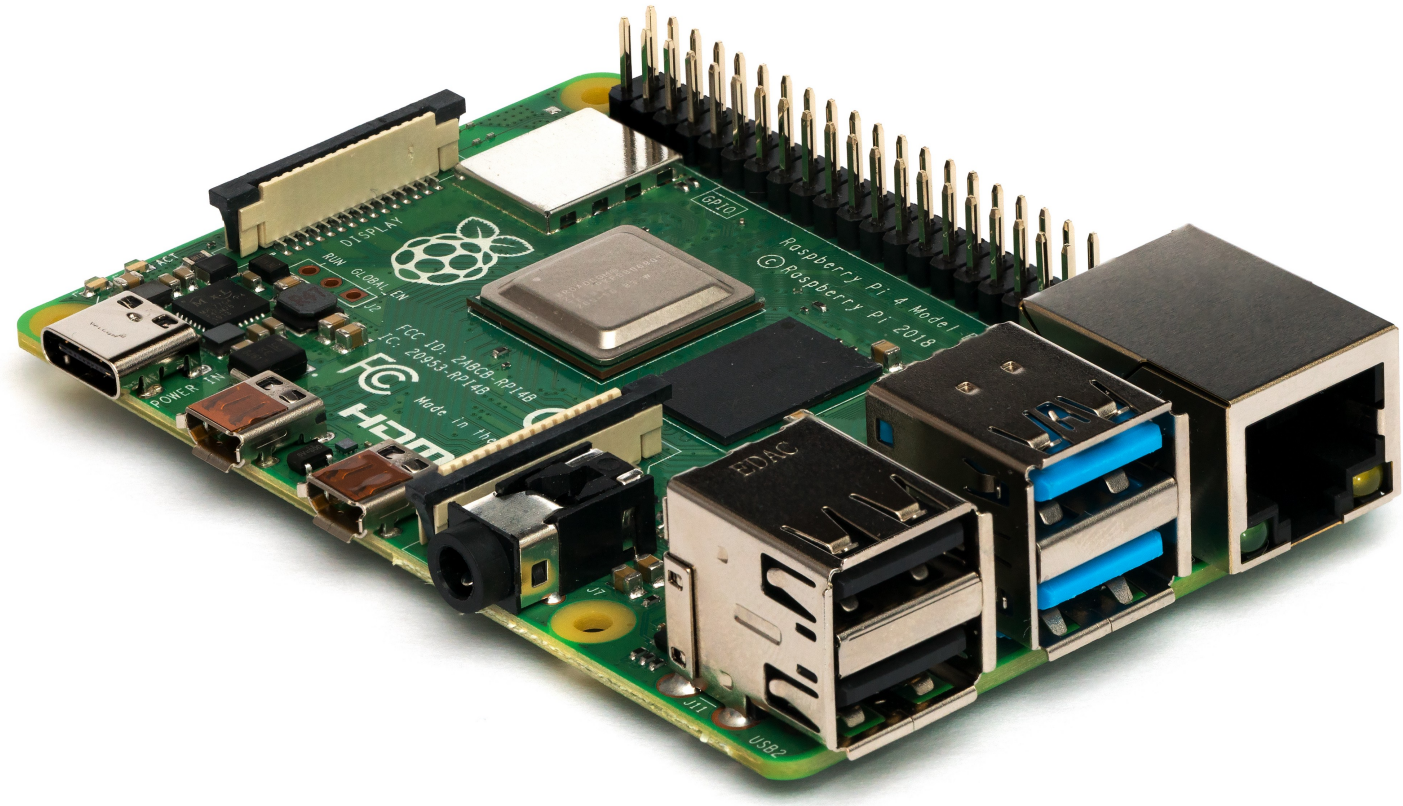
Apache Airflow Running on a Raspberry PI

Craig Godden-Payne
Jul 7 · 3 min read ★



I had an underutilized raspberry pi, running raspian headless.

I set it up a couple of months back to run pihole

To utilise my pi further, I decided to see if I could run Airflow on it. I have used airflow in the past to run python tasks and thought it could come in handy for running some

automation tasks at home.

Turns out it is very easy to set up and runs really well!

What is Airflow

> *Apache Airflow*
>
> *Airflow is a platform created by the community to programmatically author, schedule and monitor workflows.*

Home

No more command-line or XML black-magic! Use all Python features to create your workflows including date time formats...

airflow.apache.org

Setting up PreRequisites

First SSH onto the raspberry pi.

The version I am running of Raspian was *Raspbian GNU/Linux 10 (buster)*

On my instance, python and pip was not installed, so I made sure to install them first:

```
sudo apt update
sudo apt install -y python python-pip
```

```
pip install --upgrade setuptools
```

I checked the version to make sure they were fairly up to date. At the time I was running the following versions:

```
Python 2.7.16 (default, Oct 10 2019, 22:02:15)
pip 18.1 from /usr/lib/python2.7/dist-packages/pip (python 2.7)
```

Now Python and Pip are installed.

Setup Airflow

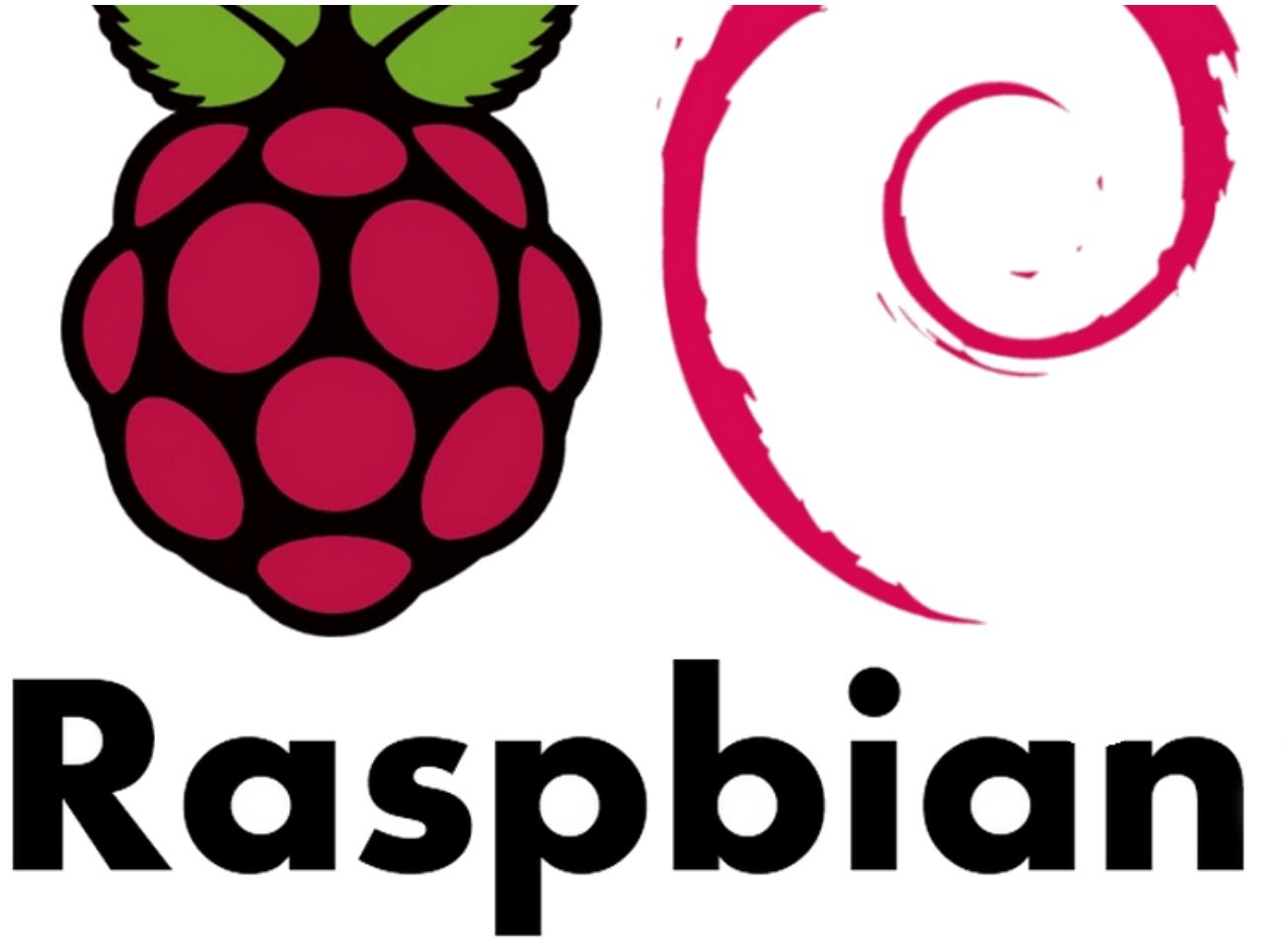
The first stage of setup, is to set the airflow home folder:

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

Then install airflow, which conveniently can be installed from pip.

```
pip install apache-airflow
```

In my case, it took a while to run.



I did hit a couple of issues:

```
Pygments requires Python '>=3.5' but the running Python is 2.7.16
apispec requires Python '>=3.5' but the running Python is 2.7.16
```

So I manually installed at a version that was compatible:

```
pip install pygments=2.4.2
pip install apispec==2.0.2
pip install pandas==0.17.1
```

Since my version of airflow is going to be fairly lightweight, I'm not going to run the database on a separate instance, or setup async task handers in celery.

You can check the default config, which will be stored at

```
~/airflow/airflow.cfg
```

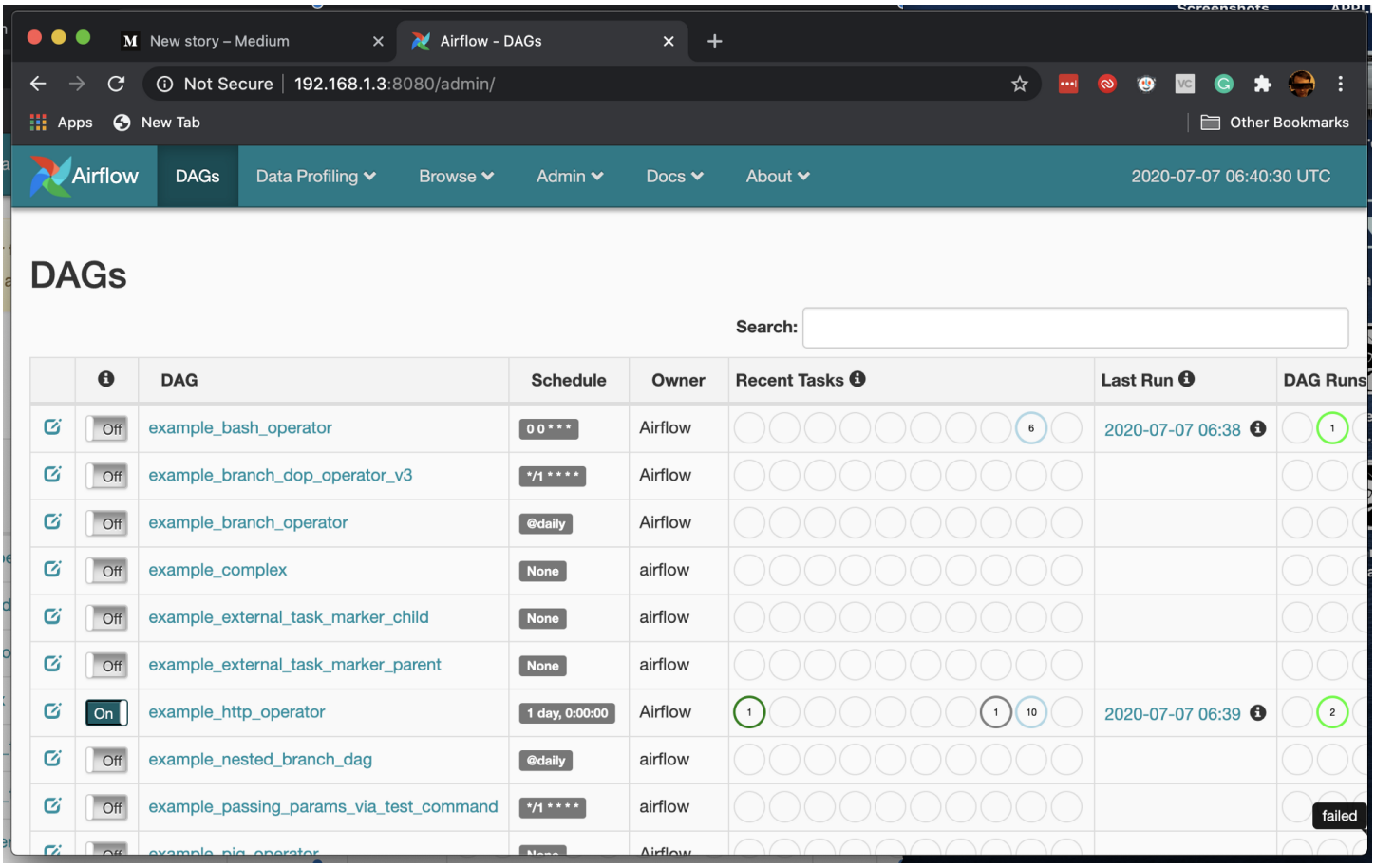
If you want to store the database in another location, change it in this config, otherwise if you want to use sqlite, just continue by running the command:

```
airflow initdb
```

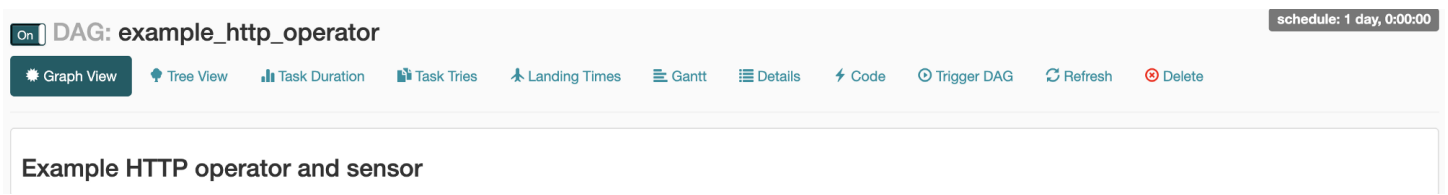
And once this completes, run the airflow webserver:

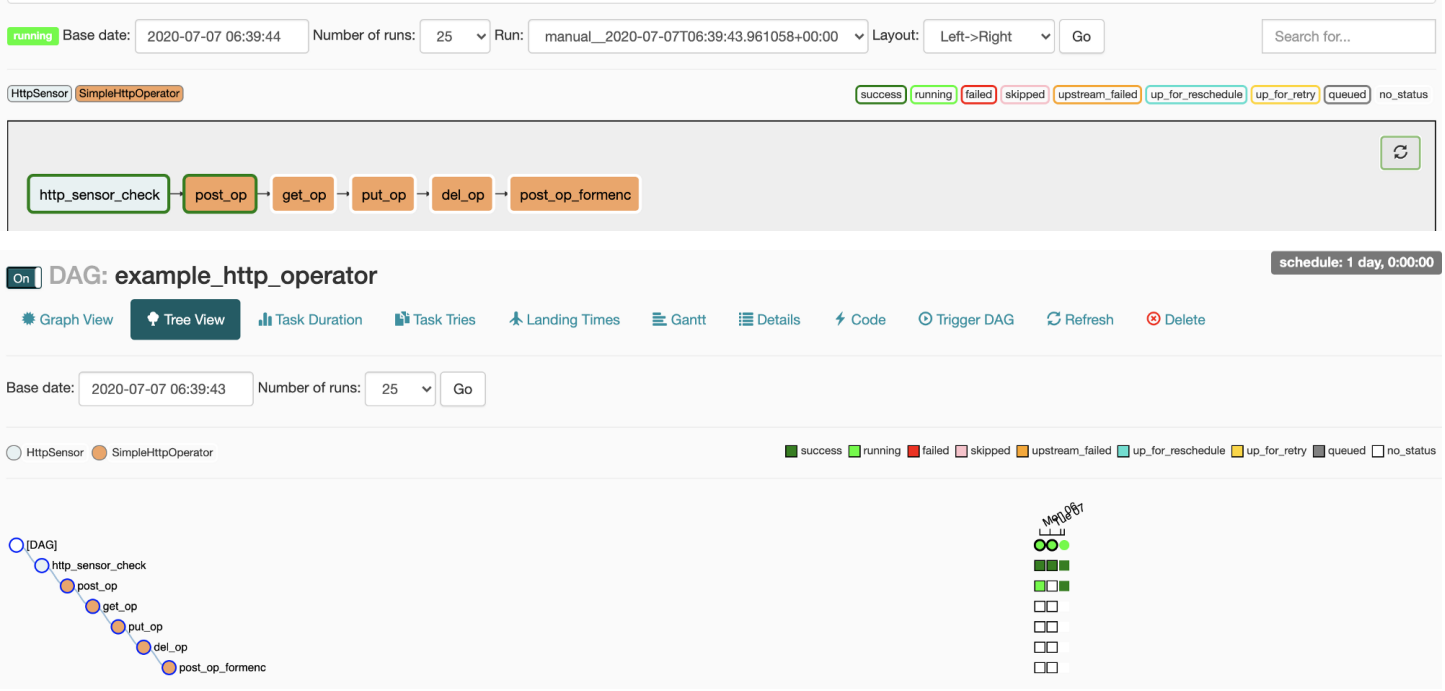
```
airflow webserver -p 8080 & airflow scheduler
```

Then navigate to using your browser



Awesome airflow is now running. You can now add your own DAGs, or test out the example DAGs that are provided





I tested Airflow, and it all works fine, the performance isn't great but it works, and its really handy having a Python task runner working on my PI!
