

Airflow Installation

Apache Airflow is a tool that helps you manage and schedule data pipelines.

Requirements:

You need Python 3.8 or higher, Windows 10 or higher, and the Windows Subsystem for Linux (WSL2)

WSL2 should be installed on your operating system.

Step 1: Set Up the Virtual Environment

To work with Airflow on Windows, you need to set up a virtual environment. To do this, you'll need to install the `virtualenv` package.

`pip install virtualenv`

Create the virtual environment like this:

`virtualenv airflow_env`

Step 2: Set Up the Airflow Directory

Create a folder named `airflow`.

Now that you have created this folder, you have to set it as an environment variable. Open a `.bashrc` script from the terminal with the command:

`nano ~/.bashrc`

Then write the following:

`AIRFLOW_HOME=/c/Users/[YourUsername]/airflow`

Step 3: Install Apache Airflow

With the virtual environment still active and the current directory pointing to the created Airflow folder, install Apache Airflow:

```
pip install apache-airflow
```

Initialize the database:

```
airflow db init
```

Step 4: Create an Airflow User

When airflow is newly installed, you'll need to create a user. This user will be used to login into the Airflow UI and perform some admin functions.

```
airflow users create --username admin --password admin --firstname admin --lastname admin --role Admin --email youremail@email.com
```

Check the created user:

```
airflow users list
```

Step 5: Run the Webserver

Run the scheduler with this command:

```
airflow scheduler
```

Launch another terminal, activate the airflow virtual environment, cd to \$AIRFLOW_HOME, and run the webserver:

```
airflow webserver
```

or **airflow webserver --port <port number>**

or

1. The first step is to download Docker Desktop from the official website. For this article, I installed 4.4.4 version.
2. After installing Docker Desktop, we need to download a docker-compose.yaml file that you can also find here.
3. Now that we have both files, we need to create our airflow directory. Go to the following path: C:/Users/<your_user>/. Inside of that directory, create a folder called docker and inside of docker create another folder called airflow.
4. Now that we have our airflow folder, we must do the following: a)Create three folders called dags, plugins and logs respectively; b) Move our YAML to that directory.
5. Now we are ready to start our instance of Airflow in docker. We need to open a PowerShell window and go to the directory above.
6. Then, we need to run the following commands:

docker-compose up airflow-init

docker-compose up

Note: After second command, a window will keep running some code. We can close this window without any problem.

7. After that, we can go to our Docker Desktop app and we will see that a container named airflow was created inside section Containers/Apps. This container will have 7 sub-containers inside.
8. With this, we have our Apache Airflow instance completely ready to start developing our DAGS.
9. Go to localhost:8080, login with user “airflow” and password “airflow” and start coding.

Or

Create an Ubuntu EC2 instance:

- 1) Launch a new Ubuntu EC2 instance (at the very least a t3.medium instance).
- 2) Create a new Security Group with the following permissions for Inbound Rules: SSH on port 22, HTTP on port 80, TCP on port 8080 all from your IP.
3. Attach the Security Group to your EC2 instance.

II. Install and Configure Airflow

SSH into the instance using a key file OR use EC2 instance connect (at the time of writing EC2 instance connect was buggy for Ubuntu instances).

Run the following commands

```
sudo apt install python3-pip  
  
$ sudo apt-get install software-properties-common  
  
$ sudo apt-add-repository universe  
  
$ sudo apt-get update  
  
$ sudo apt-get install python-setuptools  
  
$ sudo apt install python3-pip  
  
$ sudo apt-get install libmysqlclient-dev  
  
$ sudo apt-get install libssl-dev  
  
$ sudo apt-get install libkrb5-dev
```

3. A good practice before running the next two commands is to create a python virtual environment and activate it:

```
$ pip3 install apache-airflow[s3,aws,postgres]
```

```
$ pip3 install typing_extensions
```

4. Create a directory called ‘airflow’ in /home/ubuntu/.

```
$ cd ~$ mkdir airflow
```

4. Add AIRFLOW_HOME to your environment variables in ‘/etc/environment’ (this like `~/.bashrc` or `~/.bash_profile` is where Ubuntu stores its environment variables for its bash terminal)

For ex.

```
sudo vim /etc/environment
```

and then add

```
AIRFLOW_HOME = '/home/ubuntu/airflow'
```

5. Run the following to initialize the Airflow metadata DB and a user login:

```
$ airflow db init
```

```
$ airflow users create — username admin — password <password> — firstname  
Anonymous — lastname Admin — role Admin — email admin@example.org
```