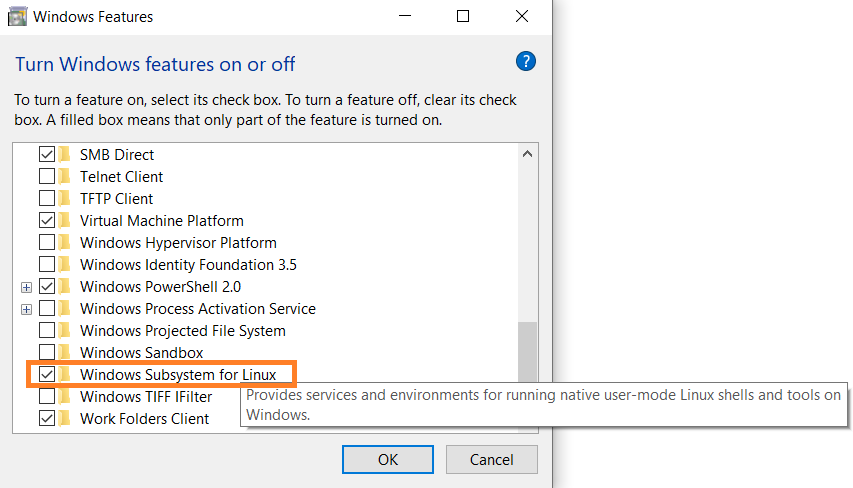
# Airflow instalation on Windows

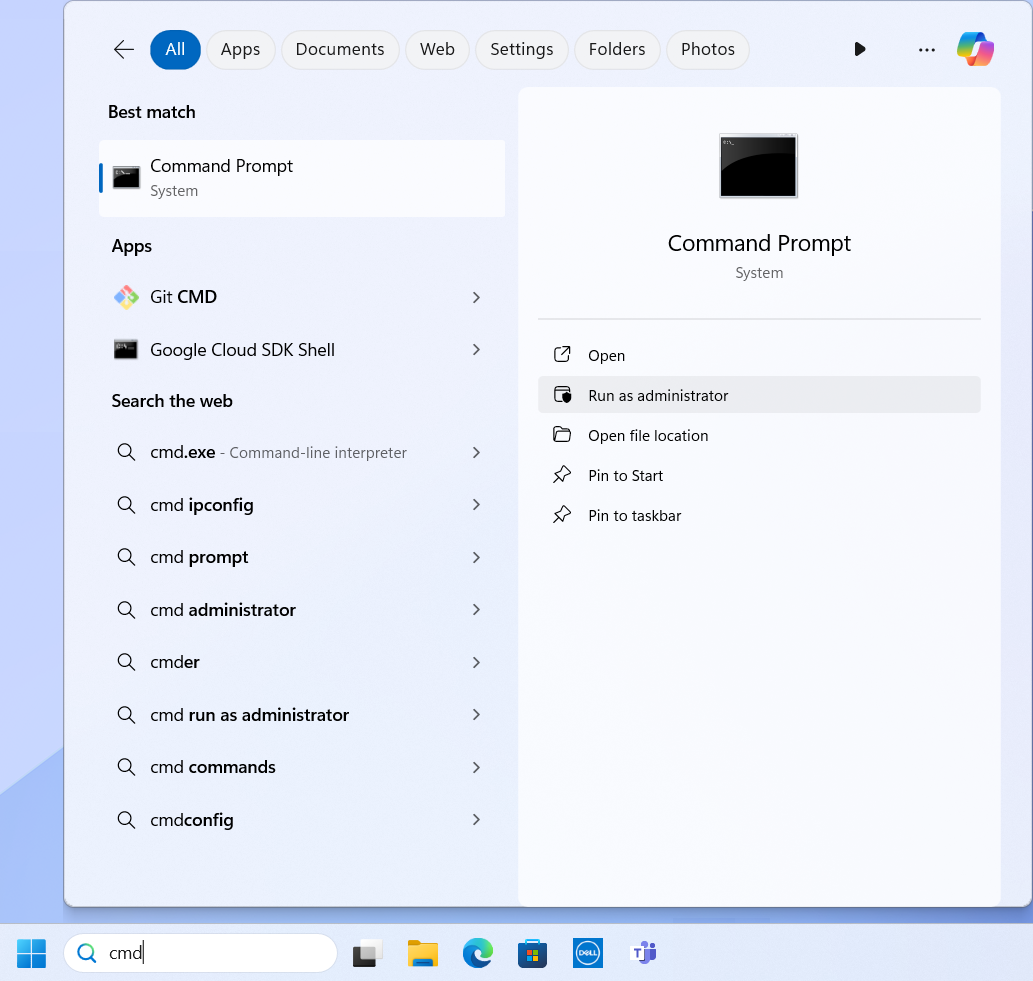
You need to have Windows 10 or higher, because you need to have Windows Subsystem for Linux (WSL). If this program is not enabled, you must perform the following steps.

1. Look in the Windows main menu "Turn Windows features on or off" and enable the Windows Linux Subsystem option.



This could cause Windows to have to restart.

1. Open a terminal (command prompt) as administrator



Run the next commands

wsl --install

You will see the options to install, so you must choose a distribution and install it, in this case you must install Ubuntu

wsl --install -d Ubuntu

If when installing a terminal opens with the following error

Error: 0x800701bc WSL 2 requires an update to its kernel component. For information please visit https://aka.ms/wsl2kernel

You must open a browser at the following link

<https://learn.microsoft.com/en-us/windows/wsl/install-manual>

In the link are the steps to fix the previous error

1. Once the Linux distribution is installed, a terminal will open with the Linux command prompt, the following commands must be executed, in this case for an Ubuntu distribution

Add an user: <user>

Add a password: <password>

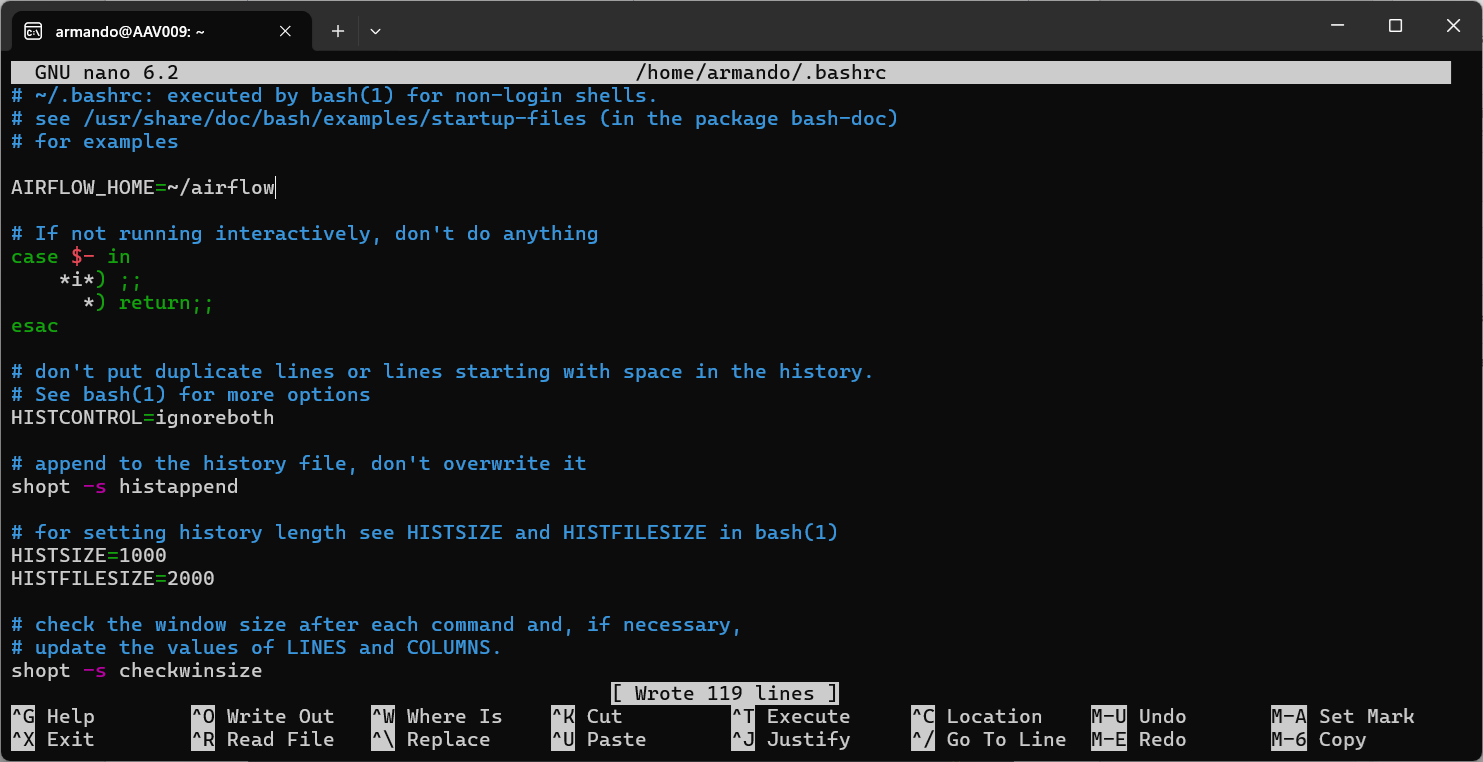
Next, the Airflow installation is performed. In the Linux window that opened during the Ubuntu installation, the following commands must be executed. In some commands it will ask if you want to continue and you must indicate yes..

|  |  |
| --- | --- |
| **Commands** | **Actions** |
| sudo apt update | previously installed packages are updated |
| sudo apt install python3.11 | python 3.11 is installed |
| sudo apt install python3-pip | the Python package manager called pip is installed |
| cd ~ | is changed to the current user's home directory |
| pip install virtualenv | the virtualenv tool is installed that allows you to create Python virtual environments |
| sudo apt install python3-virtualenv | the virtualenv tool for Python 3 is installed |
| virtualenv airflow\_env | a new Python virtual environment called airflow\_env is created with the help of virtaulenv tool |
| source airflow\_env/bin/activate | the airflow\_env virtual environment is activated |
| mkdir airflow | a directory is created in the user's home directory called airflow |
| nano ~/.bashrc | the .bashrc file opens |

For the last command, the .bashrc file will be opened and the following variable must be added

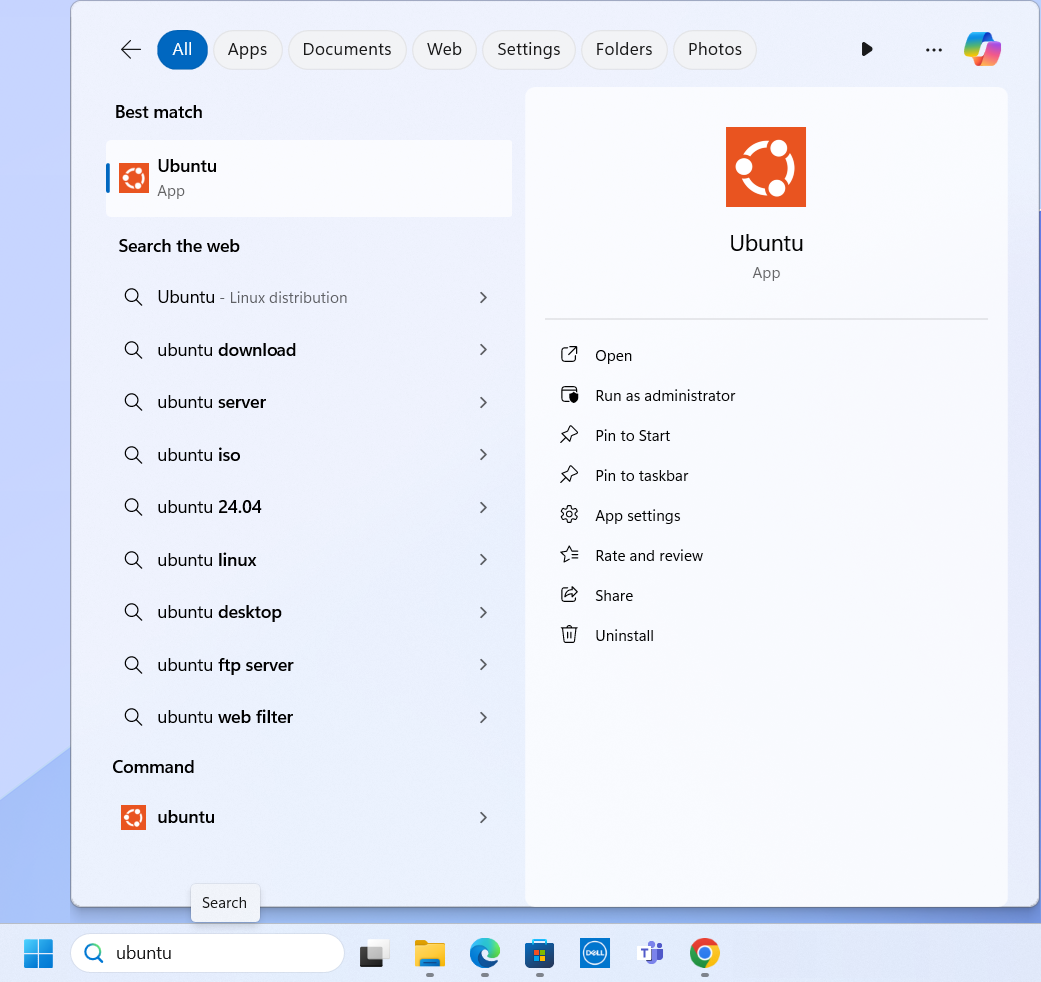
AIRFLOW\_HOME=~/airflow

As it's shown in the following



To save the changes is with ctrl + s and to exit the file is with ctrl + x

A new Linux command prompt opens to validate that the changes have been added, from the Windows search engine search for Ubuntu to open it



In this new terminal the following commands must be executed

|  |  |
| --- | --- |
| **Commands** | **Actions** |
| cd ~ | is changed to the current user's home directory |
| cd $AIRFLOW\_HOME | it will be changed to the airflow directory |
| pip install apache-airflow | Apache Airflow is installed |
| airflow version | returns the installed version of Airflow |
| airflow db init | Airflow database starts |
| airflow users create --username <user> airflow --password <password> --firstname <name> --lastname <lastname> --role Admin --email <email> | the Airflow administrator user is created |
| airflow users list | the list of Airflow users is shown |
| mkdir dags | a directory is created in the Airflow directory |

The last command is very important, because in that created directory, you will store the DAGS that you need to create and then run in the Airflow web console.

Once the above commands have been executed, all Linux terminals are closed.

## Starting Airflow

To start Apache Airflow, two new Linux command prompt terminals must be opened. In the first terminal you have to execute the following commands

|  |  |
| --- | --- |
| **Commands** | **Actions** |
| cd ~ | is changed to the current user's home directory |
| source airflow\_env/bin/activate | Python virtual environment is activated |
| cd airflow | it will be changed to the Airflow directory |
| airflow scheduler | Airflow scheduler starts |

In the second terminal you have to execute the following commands

|  |  |
| --- | --- |
| **Comandos** | **Accion que realiza** |
| cd ~ | is changed to the current user's home directory |
| source airflow\_env/bin/activate | Python virtual environment is activated |
| cd airflow | it will be changed to the Airflow directory |
| airflow webserver | Ariflow web server is started at path localhost:8080 |

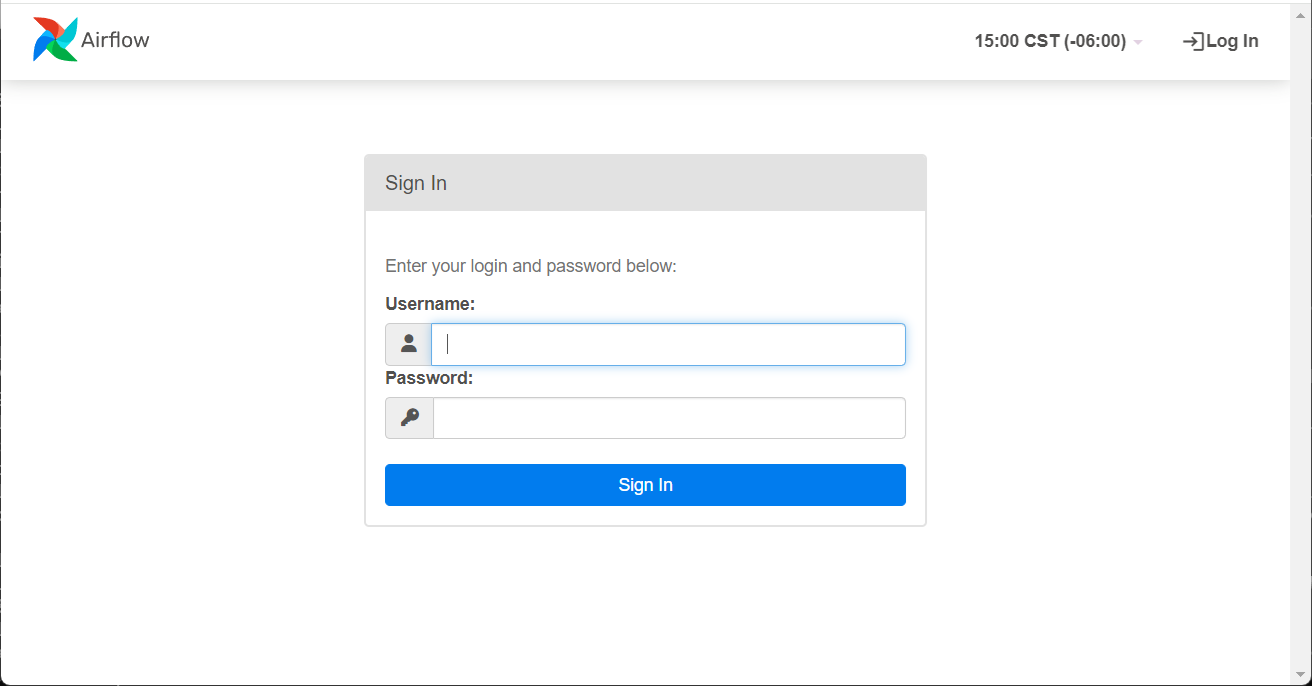
When the commands are executed in each of the windows, the processes will remain running in the foreground, so in order to return control to each of the windows it is necessary to use ctrl + c, to interrupt the commands and return control.

After executing each of the commands, you can now see the console from the following link in a browser

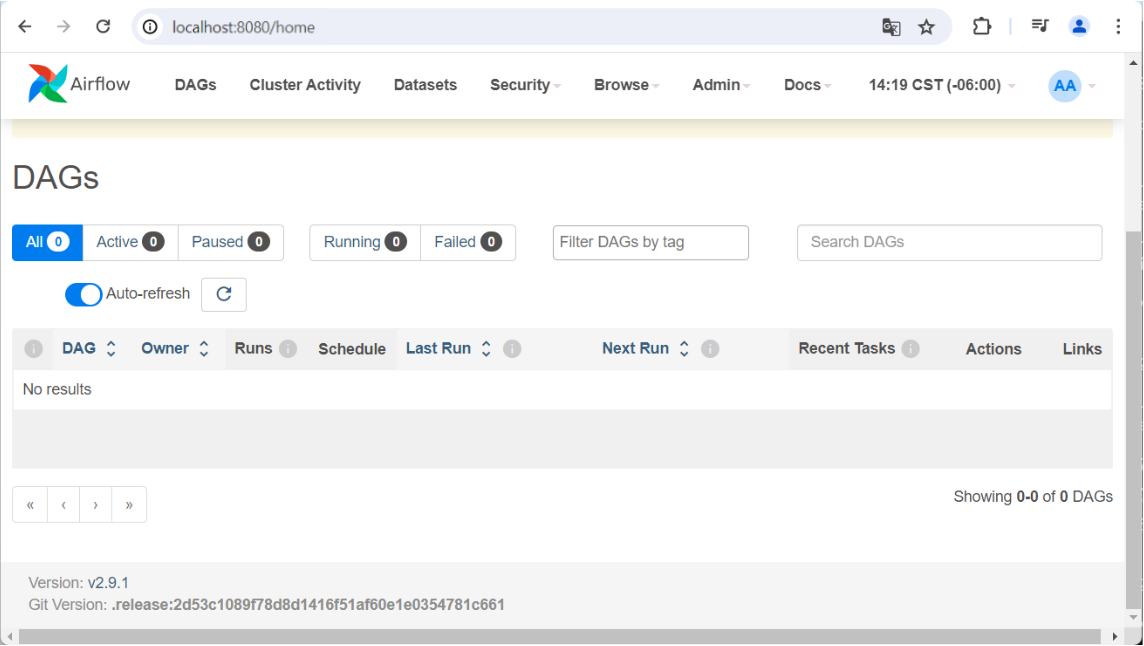
<http://localhost:8080/home>

When the Airflow console is opened for the first time, it will ask you for a username and password, which are the same ones used in the command

airflow users create --username <user> --password <password> --firstname <name> --lastname <lastname> --role Admin --email a<email>



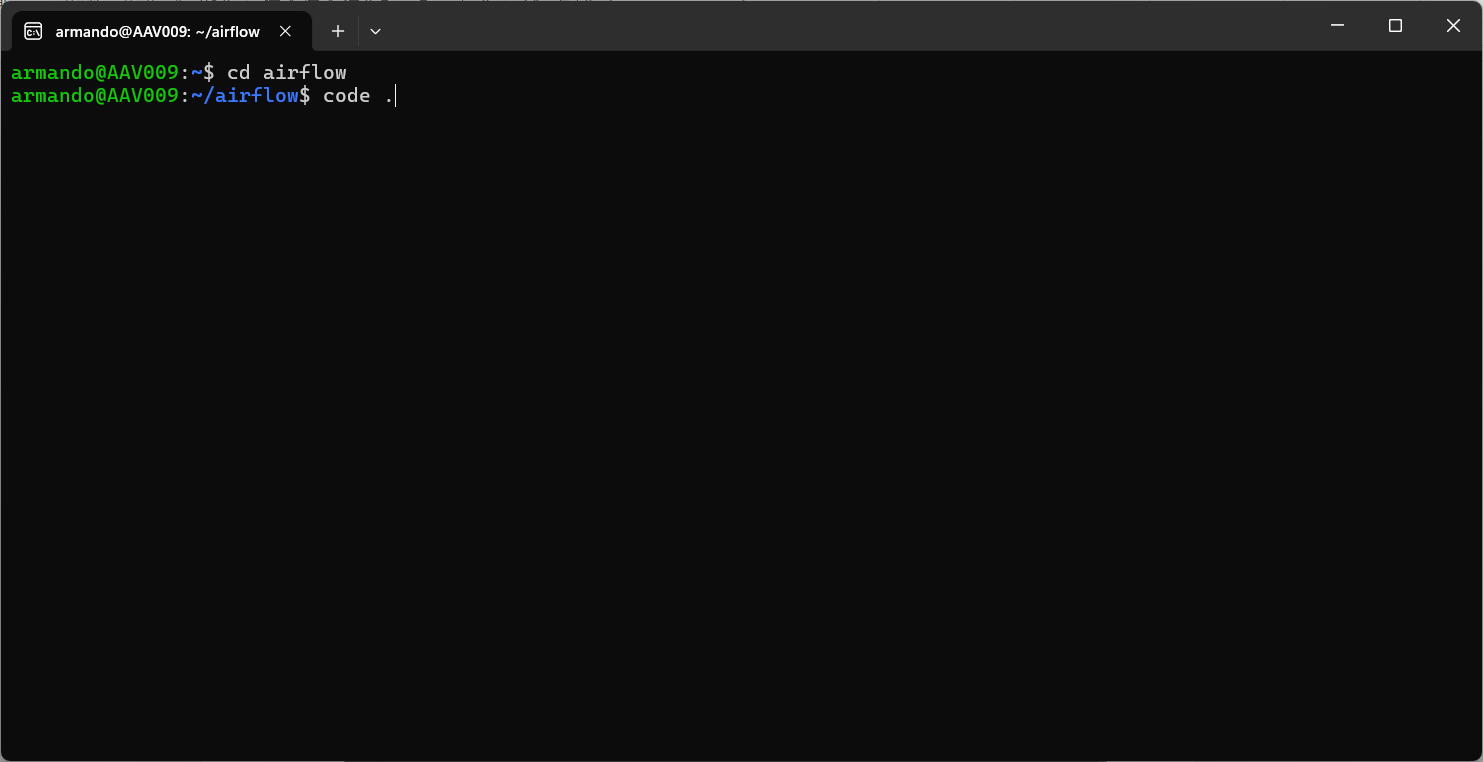
Once you give it the username and password, the Airflow console opens to see the Dags and be able to execute them.



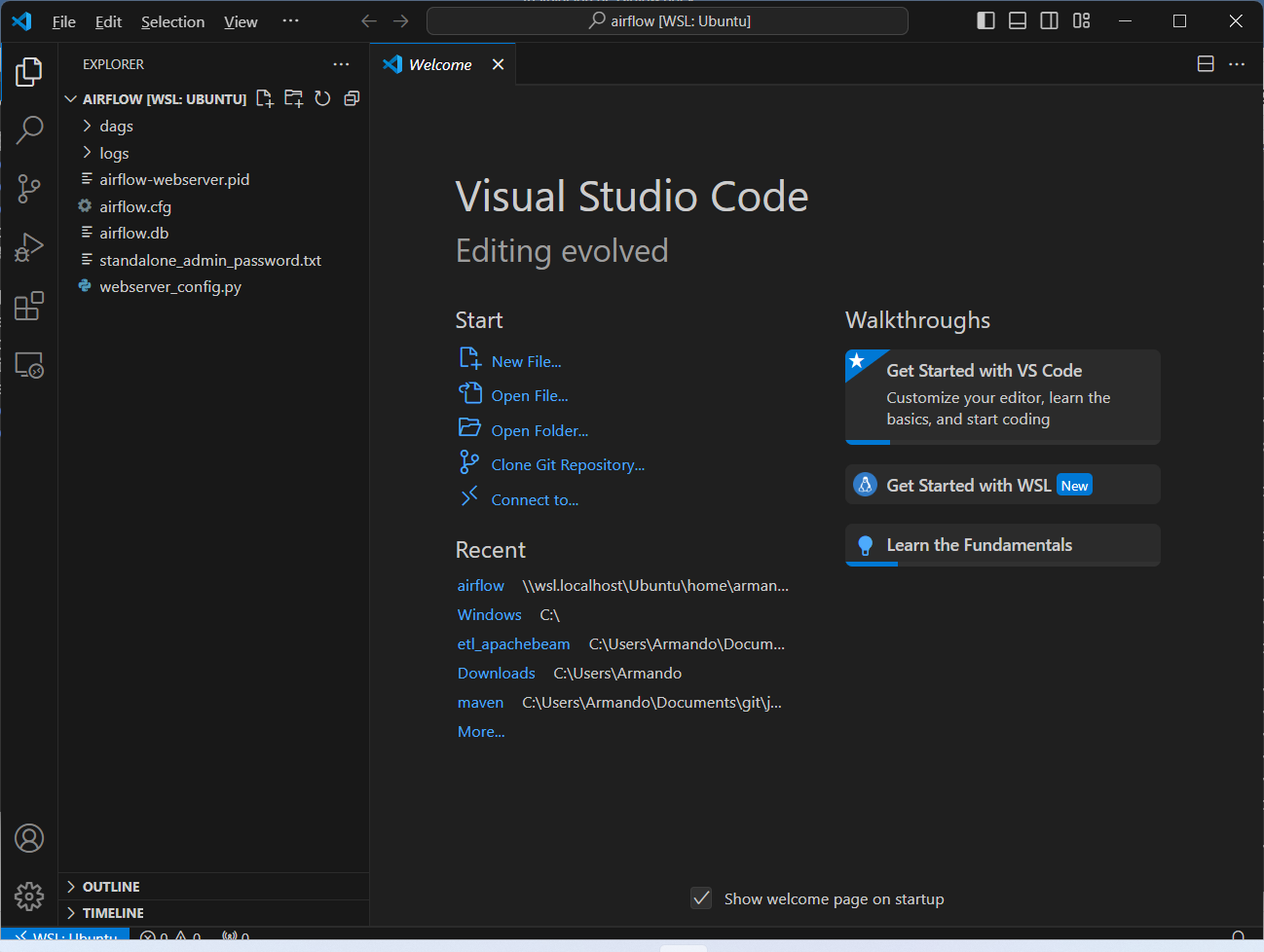
Several DAGs that come by default may appear in the console when Airflow is installed.

## Creating an Airflow DAG

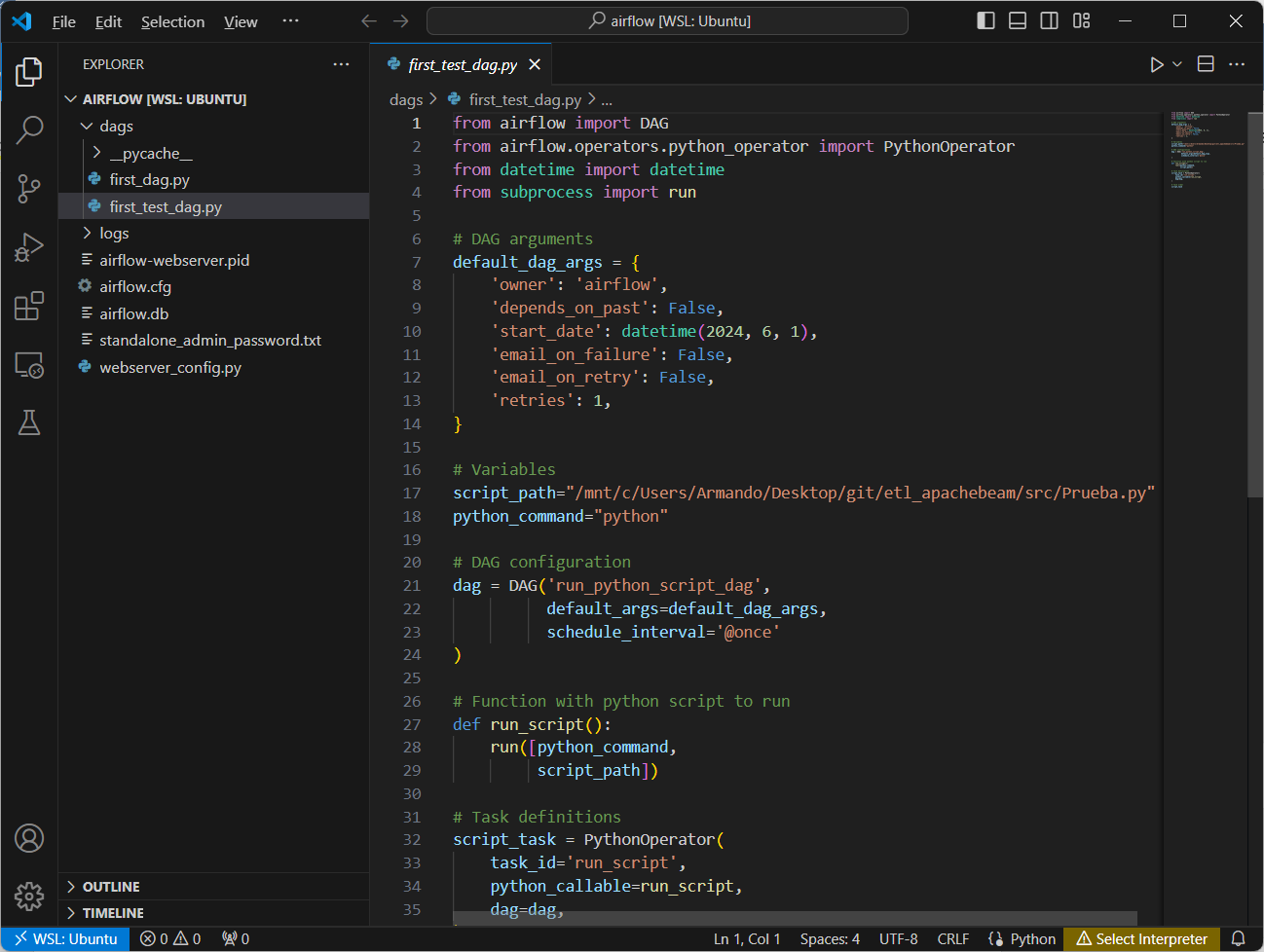
To create a new DAG, we open a new Linux terminal and execute the commands shown below, the second command works if you have previously installed Visual Studio Code software on Windows.



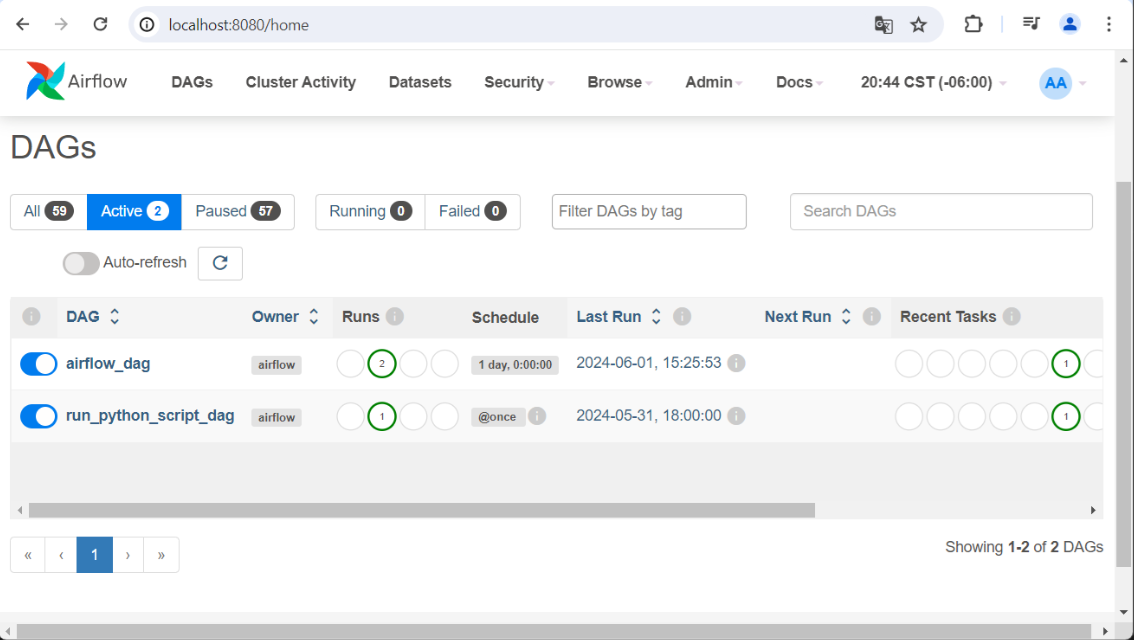
This will open a new VSCode project in Airflow's Ubuntu path.



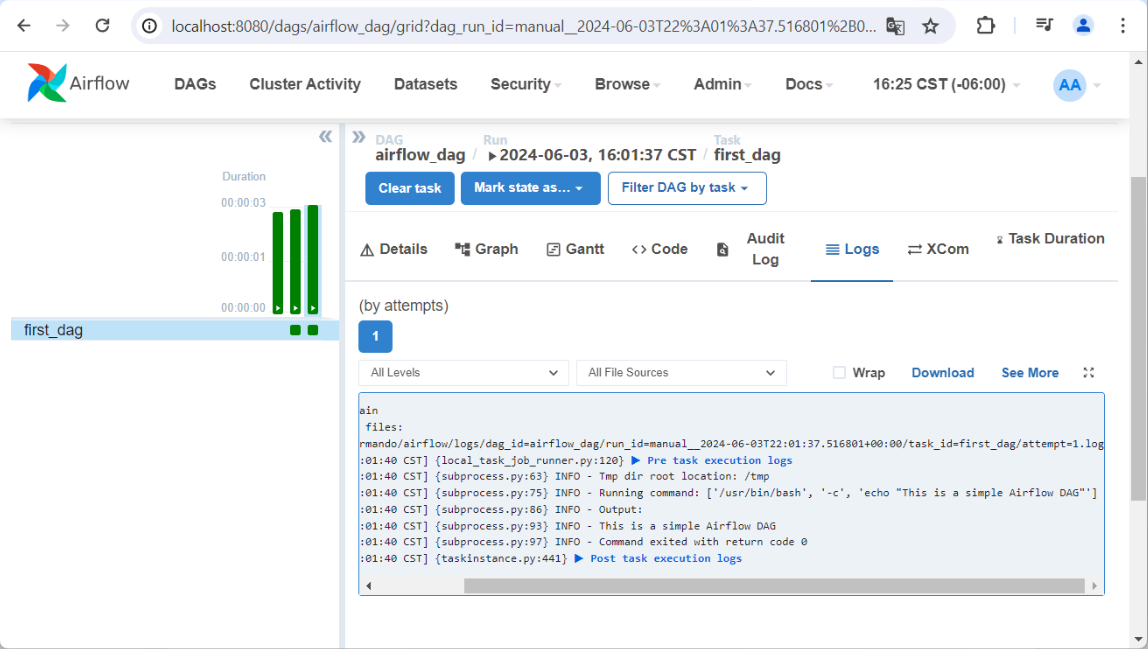
In the dags path is where the python scripts must be created, these scripts will be the DAGs that will be seen in the Airflow console, as mentioned above.



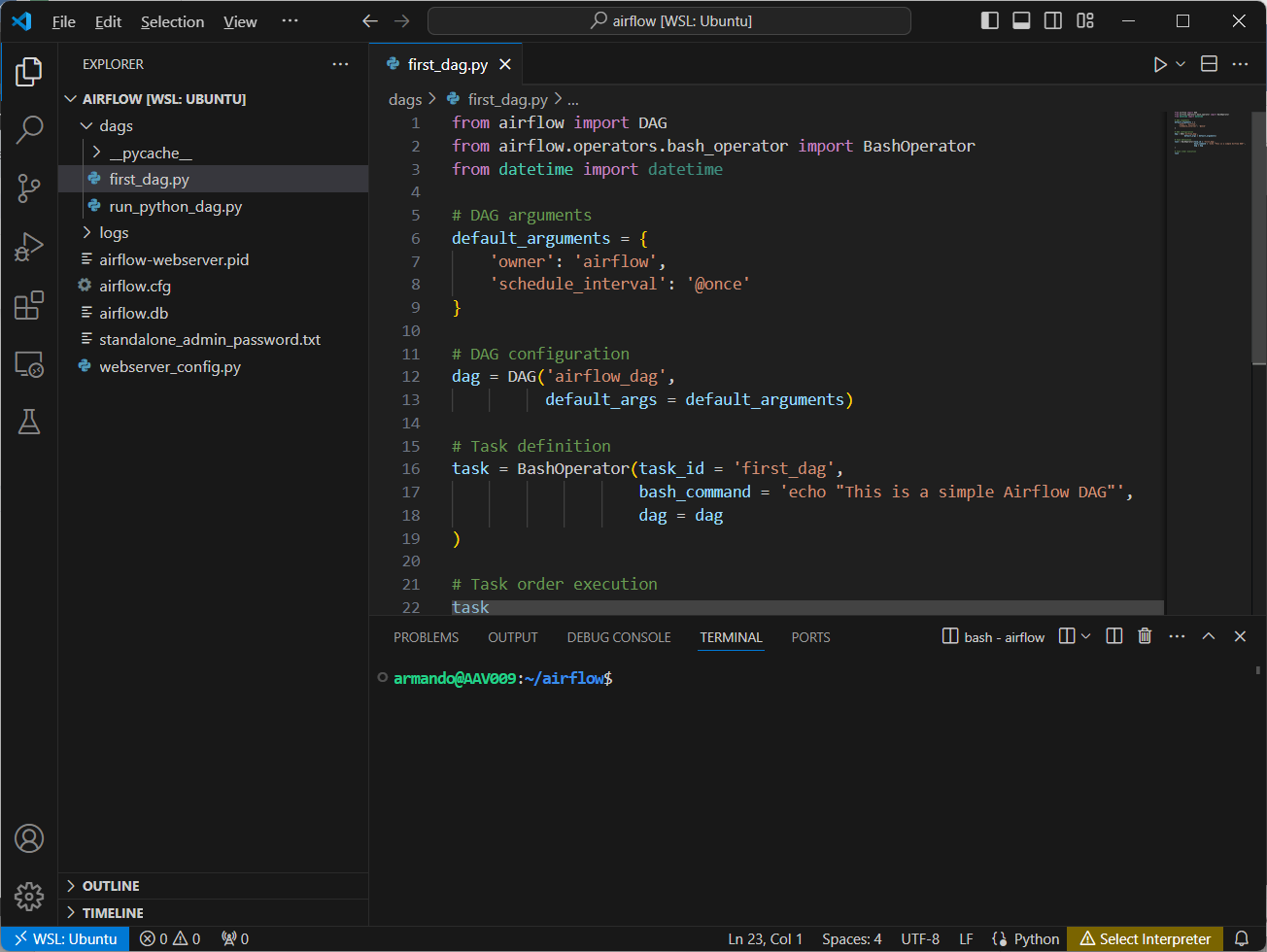
Once the scripts are saved, if they don't contain errors, they will be automatically imported into the Airflow console for later execution. If the script has an error in the console it will appear that a DAG could not be imported and will show the possible error.



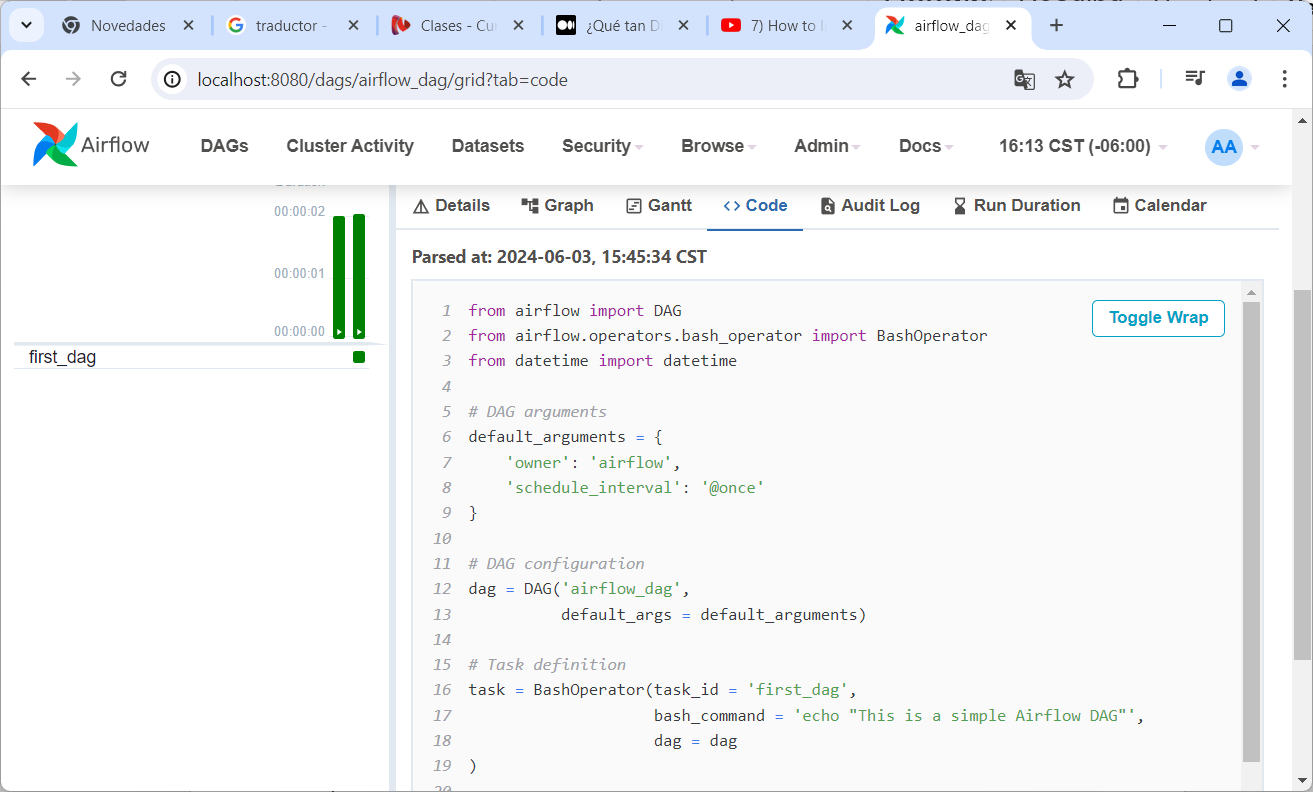
Once a DAG has been executed, the execution log can be reviewed and, if it failed, the error and its possible fix can be seen.



When you create a DAG and save it, you can view the Python code that will run the DAG from the Airflow console.



Below you can see how the same script that is saved in the dags folder is the same as the one in the Airflow console.



Although the code can be viewed from the Airflow console, it can’t be modified there. These modifications must be made from Visual Studio Code.