**How to Install Apache Airflow on Windows without Docker**

**Requirements:**

- Windows Subsystem for Linux (WSL2)

- Windows 10 or higher

- Python 3.8 or higher

**Steps:**

Step 1: Install WSL2 (Windows Subsystem for Linux 2) on Windows 10

Step 2: Set Up the Virtual Environment

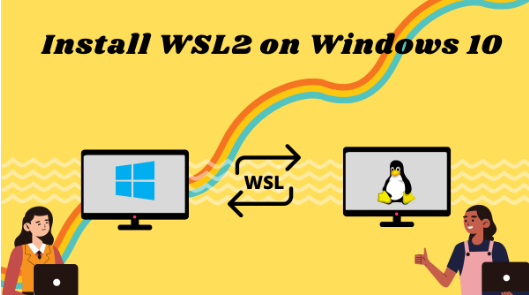
Step 3: Set Up the Airflow Directory

Step 4: Install Apache Airflow

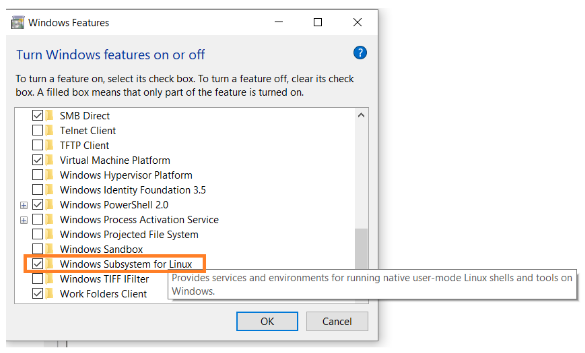
Step 5: Create an Airflow User

Step 6: Run the Airflow Scheduler & Webserver

**Step 1: Install WSL2 (Windows Subsystem for Linux 2) on Windows 10**

**

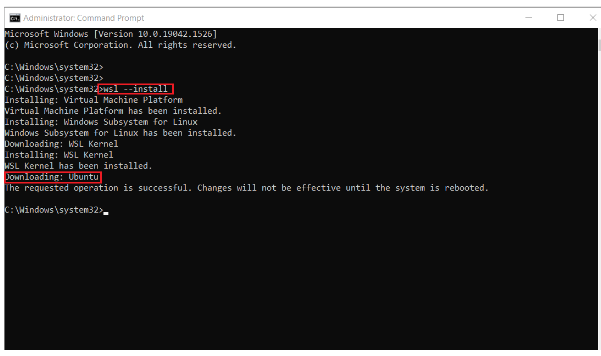
* Go to Start. Search for "Turn Windows features on or off."
* Check the option Windows Subsystem for Linux.



* Open Command Prompt as an administrator.
* Run the command below:

wsl –install

Note: By default, Ubuntu will be installed. But you can install any distro of your choice.

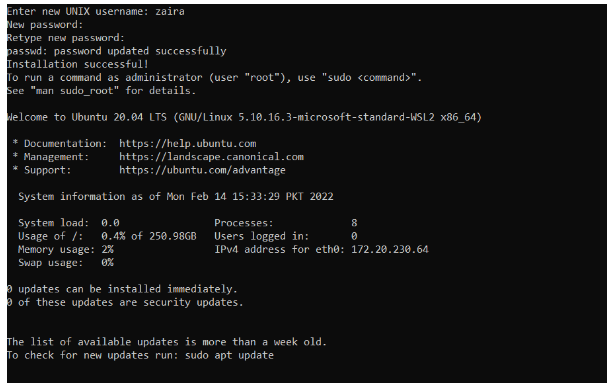


Once installation is complete, you'll need to reboot your Windows machine.

After restarting, you might see a window like this:

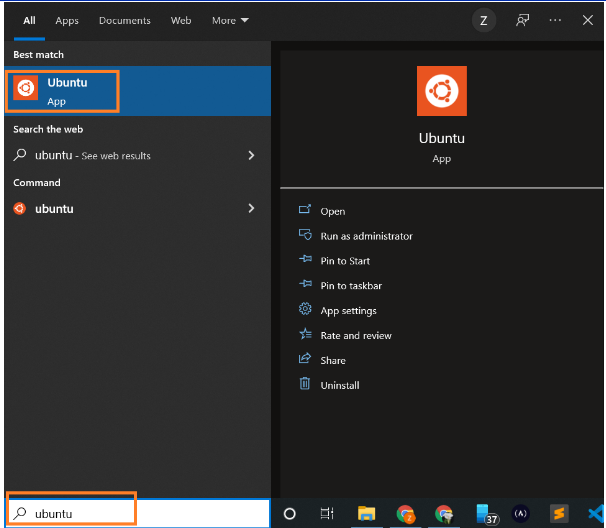


Once installation of Ubuntu is complete, you'll be prompted to enter your username and password.

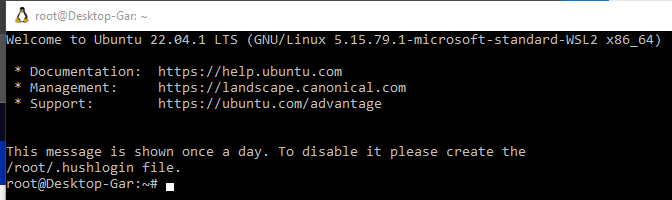


And, that's it! You are ready to use Ubuntu.

Launch Ubuntu by searching from the start menu.



And here we have our Ubuntu instance launched.

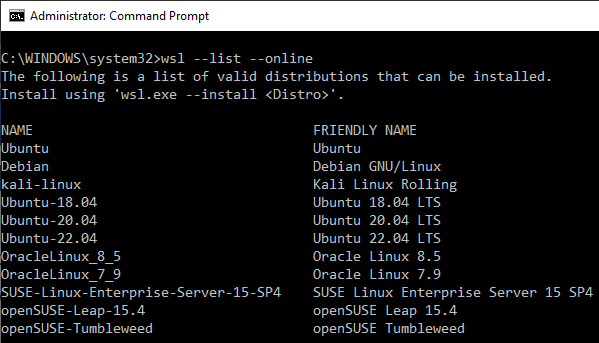


Ubuntu launched via WSL2

**How to Install a Specific Linux Distro**

If you use the default method as shown above, Ubuntu will be installed. You can find the available list of distros by running the below command on the Windows command prompt:

wsl --list --online



List the online available Linux distros

To install a specific distro, use the command below:

wsl --install -d DISTRO-NAME

For example, to install Debian, the command would be modified as follows:

wsl --install -d Debian

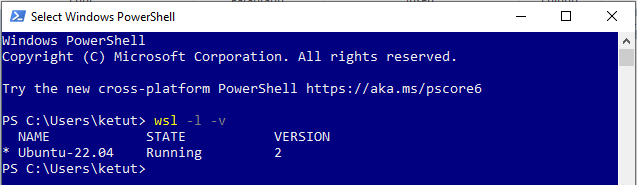
Follow the prompts and the specific distribution will be installed.

**Tip:** You can also look for updates as shown below:

wsl --update

Update WSL to the latest version

Check the status by launching Windows PowerShell.



Check status of WSL

**Step 2: 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.

Note: Make sure you are at the root of the terminal by typing:

cd ~

pip install virtualenv

Create the virtual environment like this

virtualenv airflow\_env

And then activate the environment:

source airflow\_env/bin/activate

**Step 3: 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 (you can use nano or vi):

nano ~/.bashrc

Then add the following:

AIRFLOW\_HOME=/home/ketut/airflow

**Step 4: 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

Under folder **airflow**, create a folder named **dags** to store all Airflow scripts and folder named **logs** to store all Airflow logs.

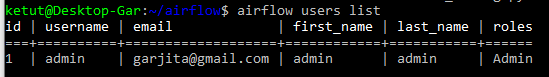
**Step 5: Create an Airflow User**

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 --assword admin --firstname admin --lastname admin --role Admin --email youremail@email.com

Check the created user:

airflow users list

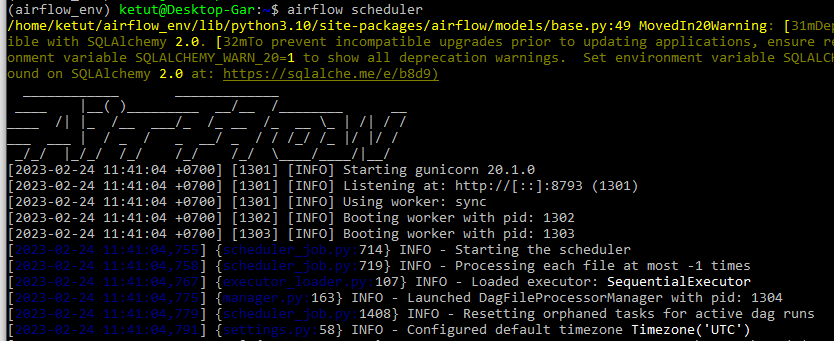


Create an Airflow user and list the created user

**Step 6: Run the Airflow Scheduler & 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



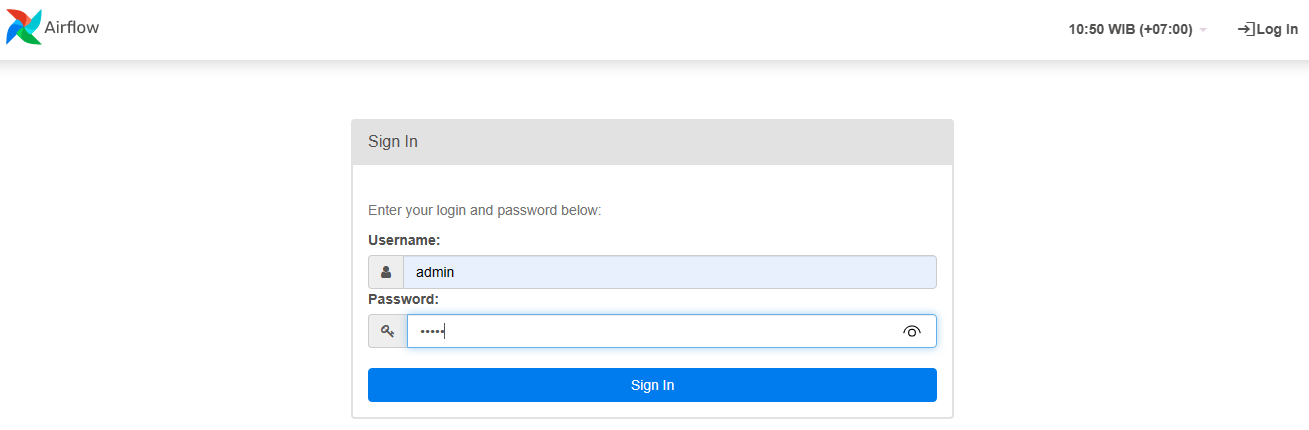
If the default port 8080 is in use, change the port by typing:

airflow webserver –port <port number>

Log in to the UI using the username created earlier with "airflow users create".

In the UI, you can view pre-created DAGs that come with Airflow by default.

[**http://localhost:8080/**](http://localhost:8080/)



**How to Create the first DAG**

A DAG is a Python script for organizing and managing tasks in a workflow.

To create a DAG, navigate into the dags folder created inside the $AIRFLOW\_HOME directory. Create a file named "hello\_world\_dag.py". Use any Editor to create script files. I prefer to use Viusial Studio (VS) Code.

Enter the code from the image below, and save it:

from airflow import DAG

from airflow.operators.python\_operator import PythonOperator

from datetime import datetime

def helloWorld():

        print("Hello World")

with DAG(dag\_id="hello\_world\_dag",

        start\_date=datetime(2023,2,7),

        schedule\_interval='\*/30 \* \* \* \*',

        catchup=False) as dag:

        task1 = PythonOperator(

        task\_id="hello\_world",

        python\_callable=helloWorld)

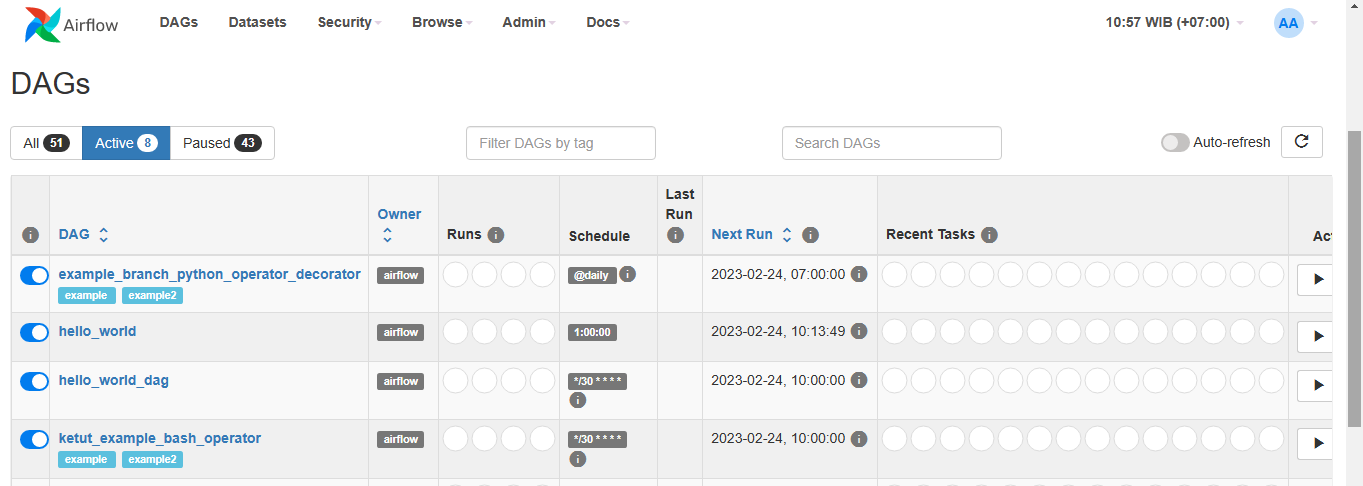
task1



Example DAG script in VS Code editor

Go to the Airflow UI and search for hello\_world\_dag. If it does not show up, try refreshing your browser.

That's it. This completes the installation of Apache Airflow on Windows.



**Wrapping Up**

This guide covered how to install Apache Airflow on a Windows machine without Docker and how to write a DAG script.

I do hope the steps outlined above helped you install airflow on your Windows machine without Docker.