Deploying Apache Airflow on an EC2 Instance

JUNE 2024

ISSUED BY

Medha Prodduturi

Introduction

This guide will walk you through the process of connecting AWS EC2 Instance with Visual Studio Code followed by setting up Apache Airflow on an AWS EC2 Instance, allowing you to effectively orchestrate complex computation workflows and data processing pipelines.

In this article:

Connecting VSCode with EC2

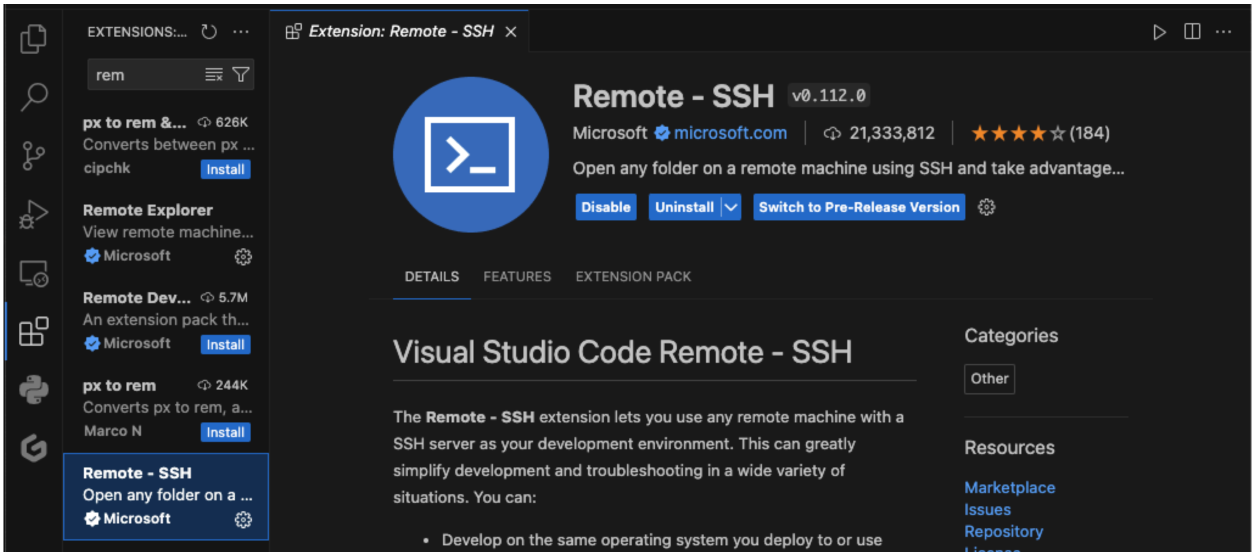
Prerequisites

1. An active AWS account.
2. Visual Studio Code installed on your local machine.
3. An EC2 instance launched and running within your AWS account.

4. A downloaded .pem key-pair file following EC2 launch
5. Basic understanding of Apache Airflow, AWS EC2 Instance, and Visual Studio Code.

Step 1: Download Remote-SSH

Navigate to Extensions (left-side) in VSCode and download the extension “Remote-SSH’ by Microsoft (shown below).



Open any folder on a remote machine using SSH and take advantage...

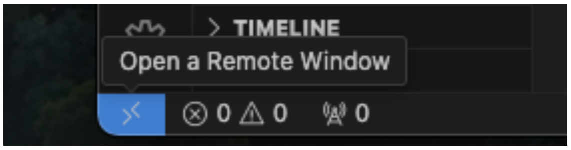
Visual Studio Code Remote - SSH

The **Remote - SSH** extension lets you use any remote machine with a SSH server as your development environment. This can greatly simplify development and troubleshooting in a wide variety of situations. You can:

- Develop on the same operating system you deploy to or use

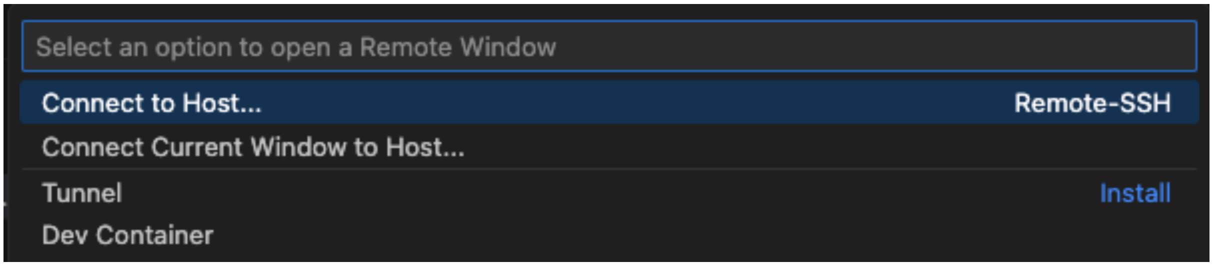
Step 2: Open a Remote Window

Click the blue >< symbols located on the bottom left corner of VSCode to open a remote window.

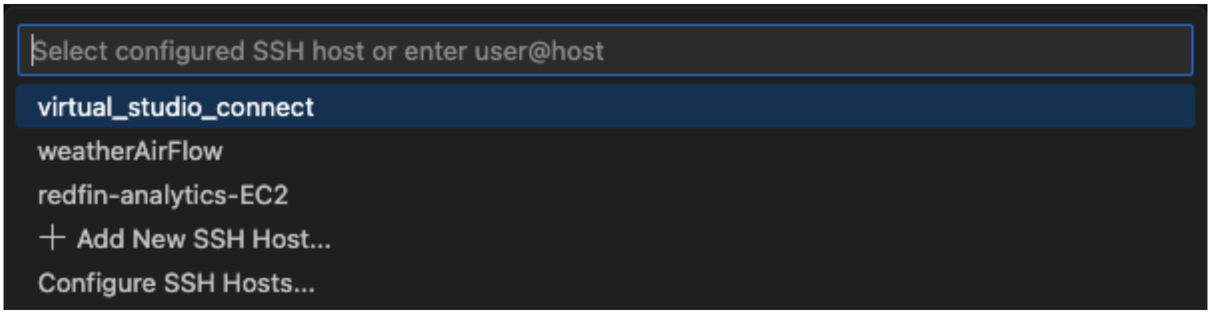


Step 3: Connect & Configure Host

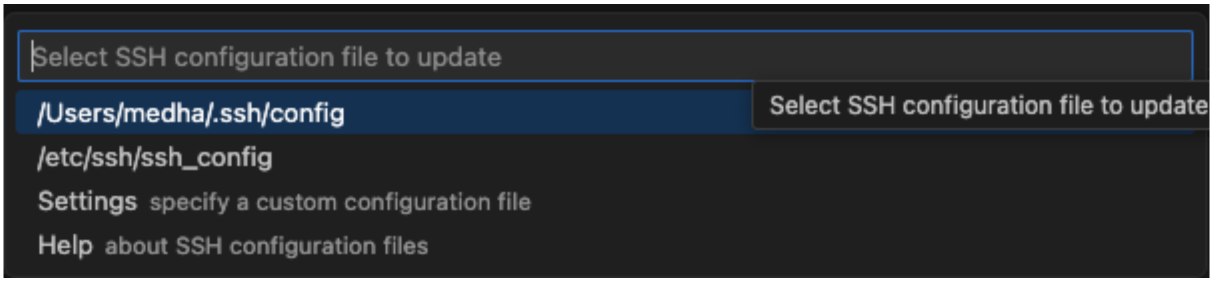
- Click on “Connect to Host”


Select an option to open a Remote Window
Connect to Host... Remote-SSH
Connect Current Window to Host...
Tunnel Install
Dev Container

- Click on “Configure SSH Hosts”



- Select the config folder


Select SSH configuration file to update
/Users/medha/.ssh/config
Select SSH configuration file to update
/etc/ssh/ssh_config
Settings specify a custom configuration file
Help about SSH configuration files

Step 4: Input EC2 Instance details

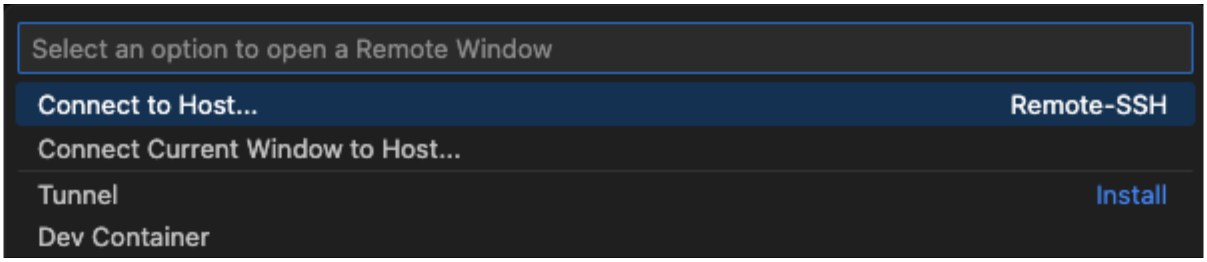
In the config file, enter the following code according to your EC2 Instance & save it.

```
#Replace this with your EC2 Instance name
Host weatherAirFlow
   # Replace this hostname with your EC2 Public IP address
   HostName 00.000.000.000
   User ubuntu
   #Replace this with your pem file location
   IdentityFile /Users/medha/Downloads/redfin-keypair.pem
```

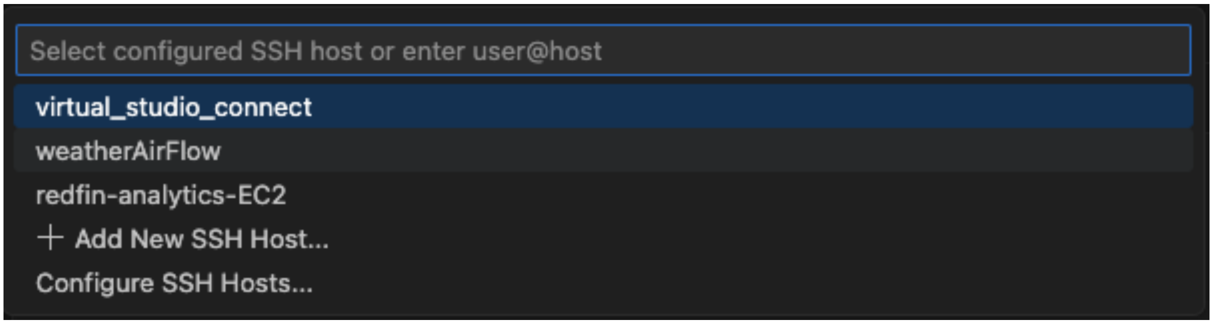
Step 5: Connect to EC2 Instance

Finally, to connect your EC2 Instance, click the blue >< symbols again.

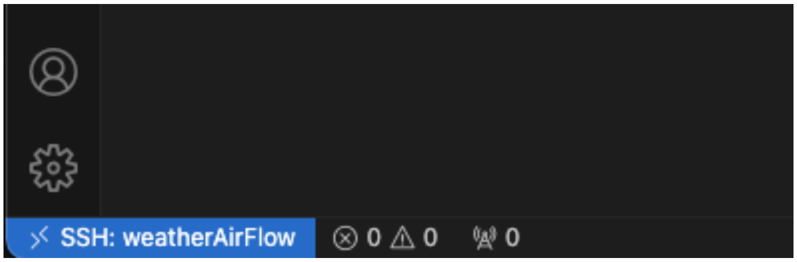
- Click on “Connect to Host” again.


Select an option to open a Remote Window
Connect to Host...
Remote-SSH
Connect Current Window to Host...
Tunnel
Install
Dev Container

- Choose the designated Host mentioned in your config file (Step 4). In this case, it is weatherAirFlow.



- Successful connection should look like this:



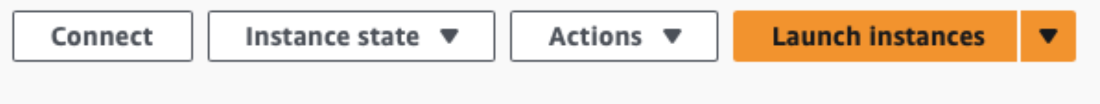
Setting Up Apache Airflow

Prerequisites

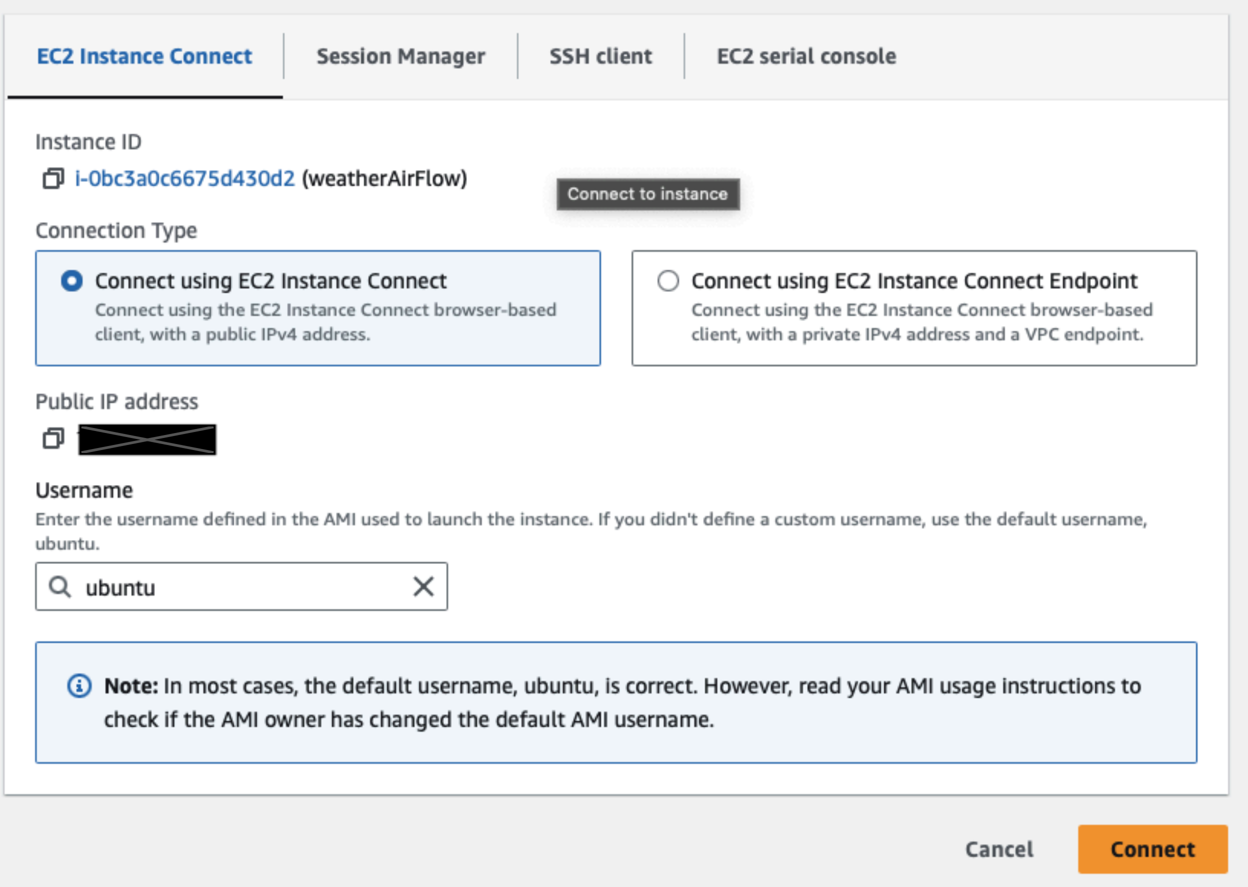
1. An active AWS account.
2. Visual Studio Code installed on your local machine.
3. An EC2 Instance launched and running within your AWS account.

Step 1: Connect your EC2 Instance

- Above the list of your instances, you will see the below navigation bar. Click on “Connect”



- Clicking on “Connect” will direct you to the below page. Click on “Connect” again.



Instance ID

i-0bc3a0c6675d430d2 (weatherAirFlow)

Connection Type

● Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

○ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

[redacted]

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Step 2: Update the package list

Run this command to update existing packages: `sudo apt-get update`

Step 3: Install Python3 package manager

Run this command to install python3 package manager: `sudo apt install python3-pip`

Step 4: Install Apache Airflow

Run this command to install airflow: `sudo pip install apache-airflow`

Step 5: Install Apache Airflow

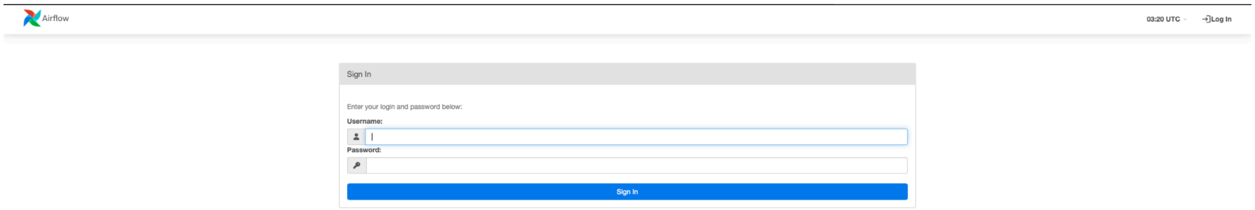
Install Amazon provider for Apache Airflow: `sudo pip install apache-airflow-providers-amazon`

Step 6: Install Apache in a standalone mode

Run this command: `airflow standalone`

Step 7: Navigate to the 8080 port

In a new tab, enter “[Your EC2 Public IP Address]:8080.” This should launch a page like below:



Sign In

Enter your login and password below:

Username:

Password:

You will find your login credentials at the end of the terminal. Enter those into the page above, and you’re in!

Conclusion

By following this guide, you have laid a solid foundation for using Apache Airflow to orchestrate your workflows on AWS. Continue exploring and experimenting to fully leverage the capabilities of Airflow in your data engineering and automation projects.