Lab 7 – Walkthrough (EC2 / PuTTY / Jupyter Notebook / GitHub)
This document is a step-by-step walkthrough of the system configuration described in "Lab 7 - Introduction to GitHub, AWS, Jupyter, SSH, PySpark and Spark"

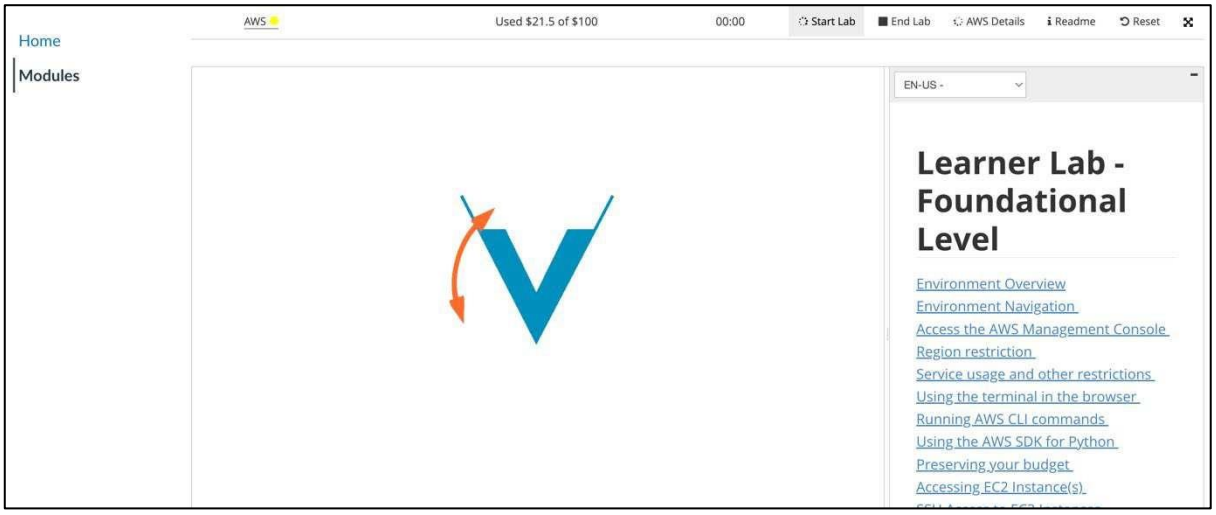
1. Canvas course acceptance

You will have received an invitation to join a Canvas course, click accept and follow the link to the course.

2. Configure and launch an EC2 instance

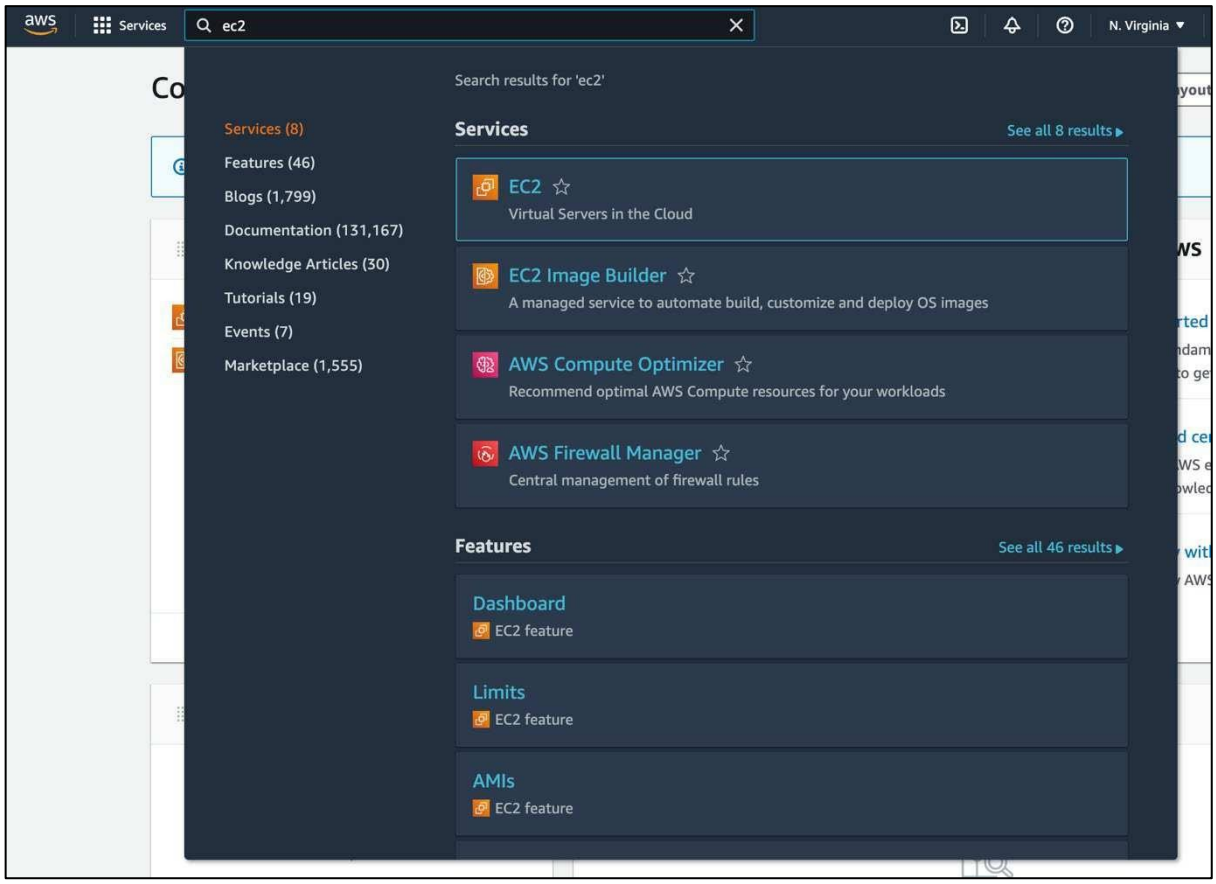
Go to Modules and click "Learner Lab – Foundational Services". Note that the URL should not be https://canvas.auckland.ac.nz/, instead it should be https://awsacademy.instructure.com/. Also note that further instructions are available in the "Student Guide.pdf", please refer to that if you get stuck.

Click "Start Lab" in the top-right.

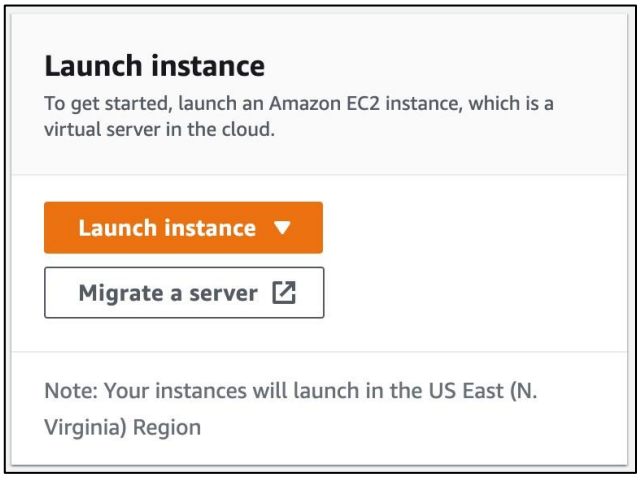


Once the lab has started, the icon next to "AWS" will turn green. Click "AWS" to go to your AWS Console.

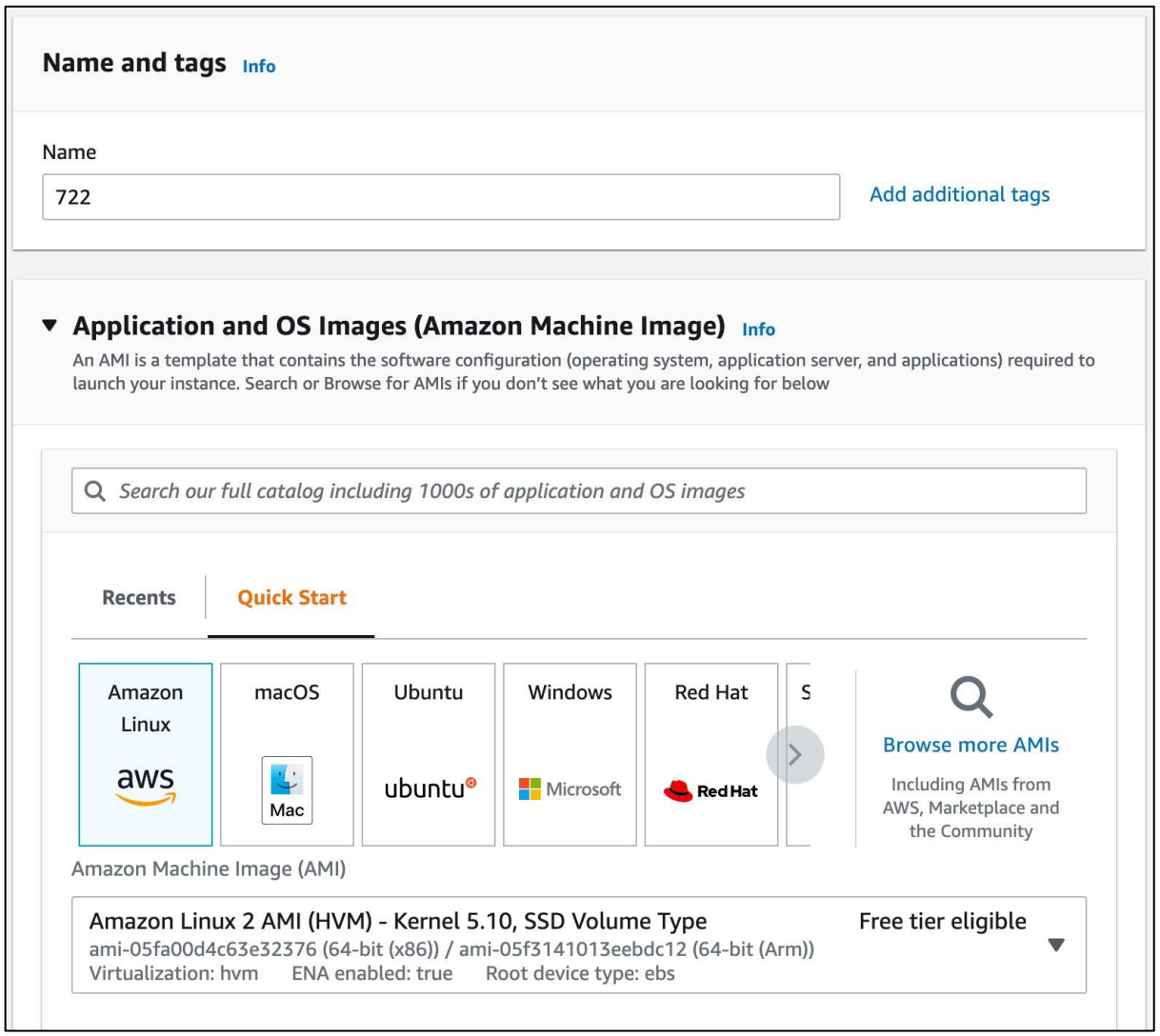
In the AWS Console, search for EC2. Under “Services” click the first service highlighted in the blue box below. **Please note that your location in the top-right must be set to “N. Virginia” (i.e. North Virginia).**



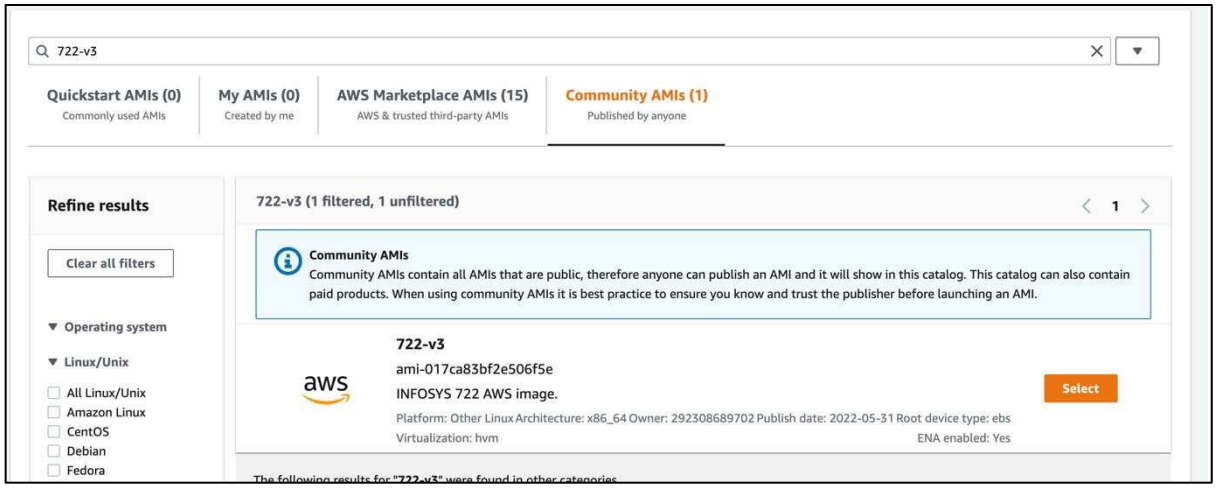
Click “Launch instance”.



Name your instance anything you like. Then, under “Application and OS Images” click “Browse more AMIs”

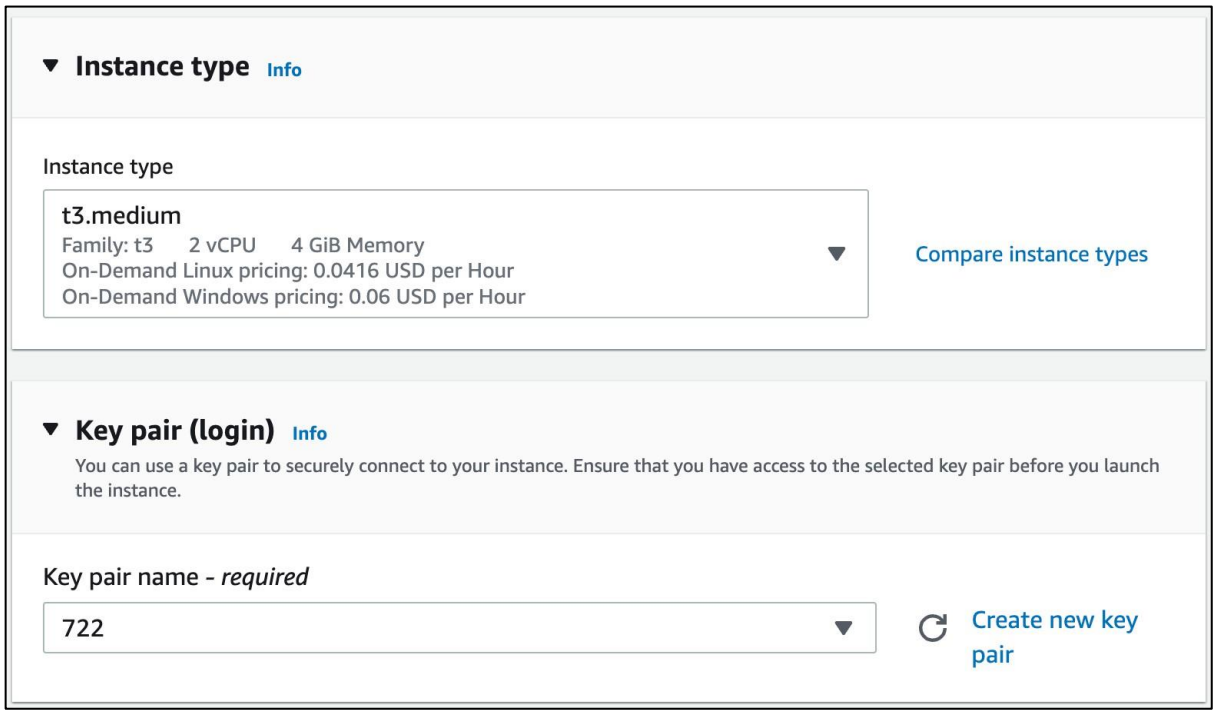


Click “Community AMIs” and search for “722-v3”. When you see the AMI shown below, click “Select” on the right.

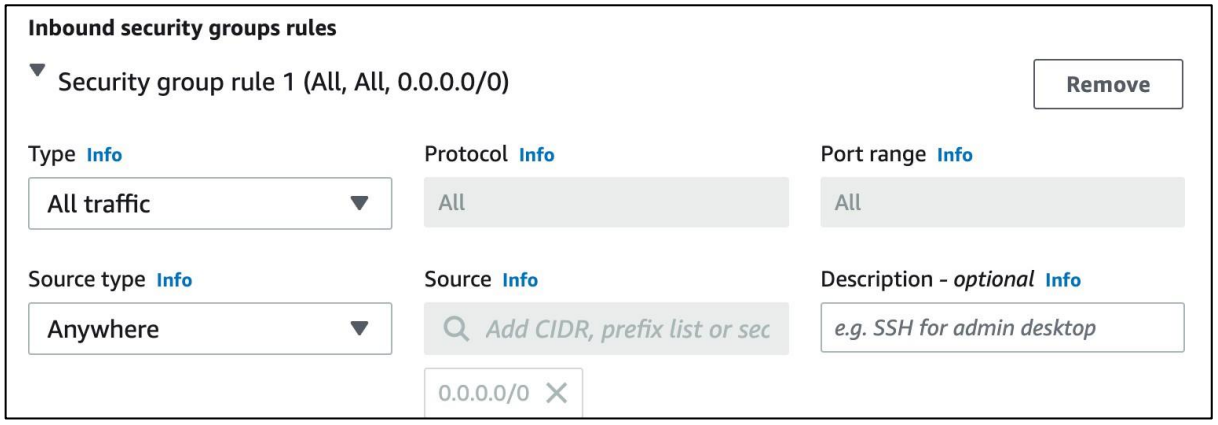


Under “Instance type” change the selection to “t3.medium”. **Note that you only have $100 credit, so do not change this to an instance that is more expensive than $0.05 USD per hour otherwise you may run out of credit before your assignment due date.**

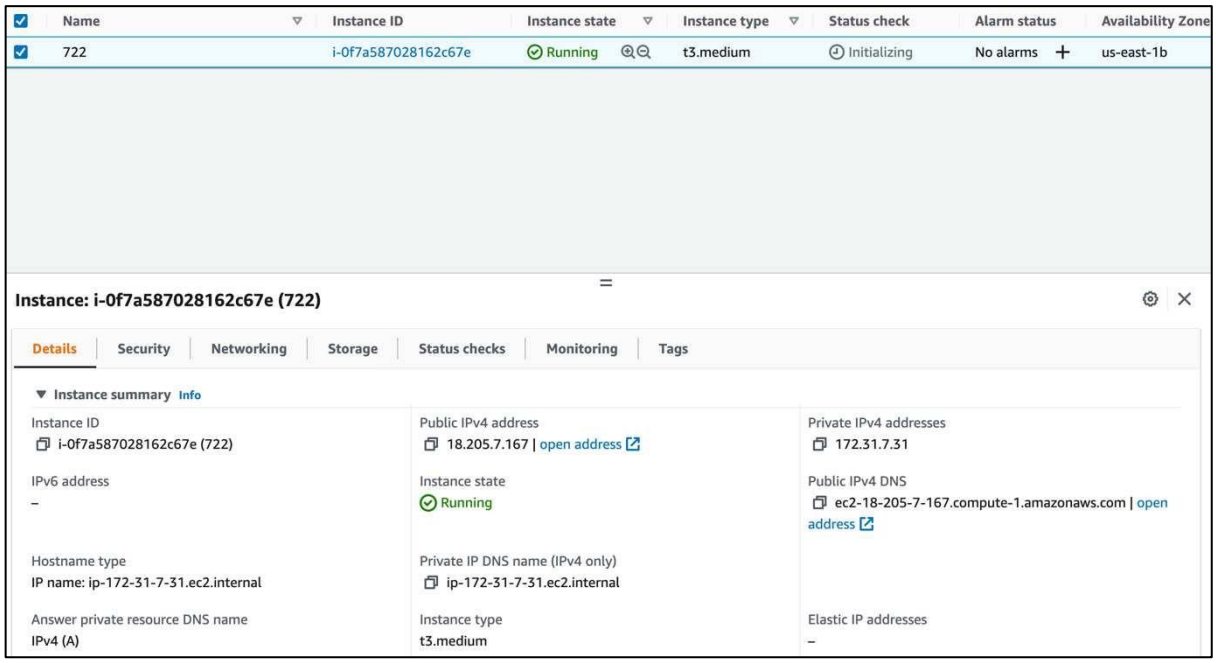
Under “Key pair” click “Create new key pair” if required and keep the downloaded file somewhere secure as you will need this file to access the server. Use the default settings.



Under “Network settings” and “Inbound security group rules”, ensure “Type” is set to “All traffic” and “Source type” is set to “Anywhere”.



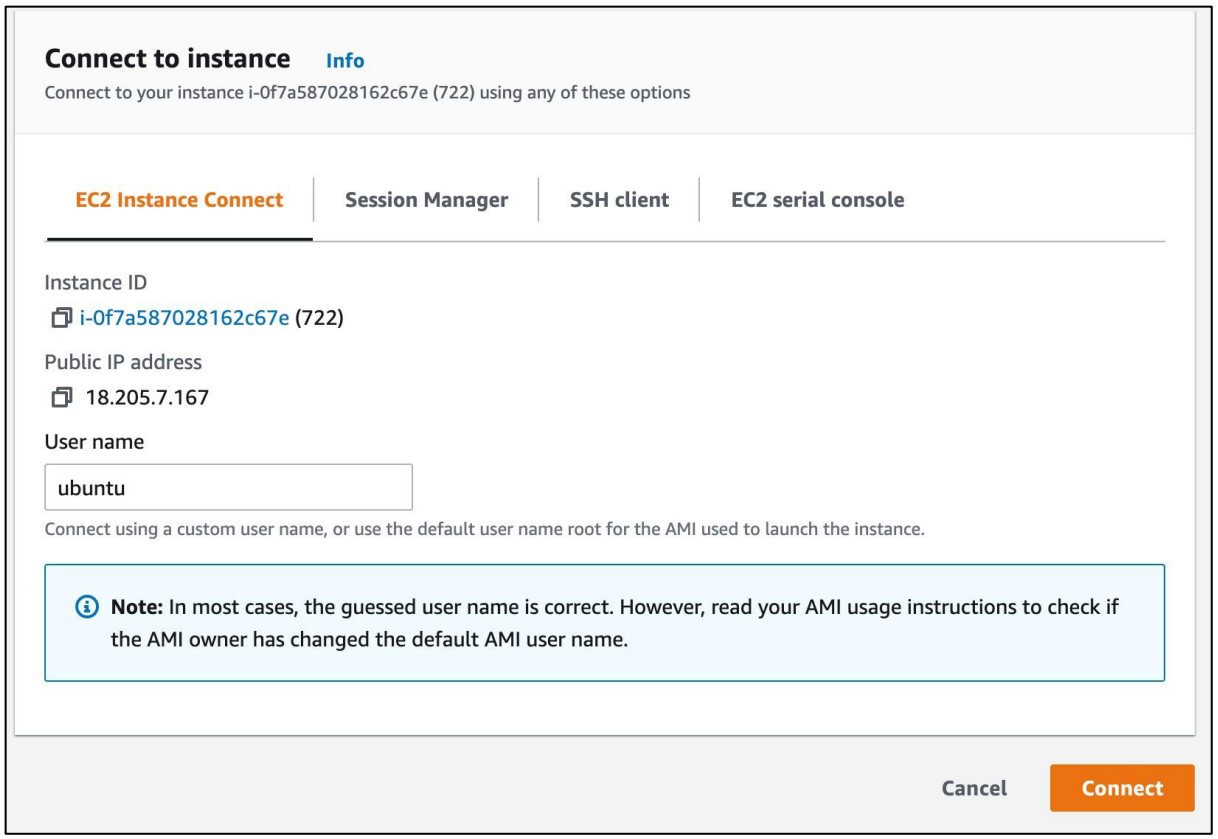
Lastly, click “Launch instance” and “View instances”. It will take a few minutes, but eventually you will see that your instance is running as shown below. Note the **IPv4 Public IP** address for later (in this case, it is 18.205.7.167).



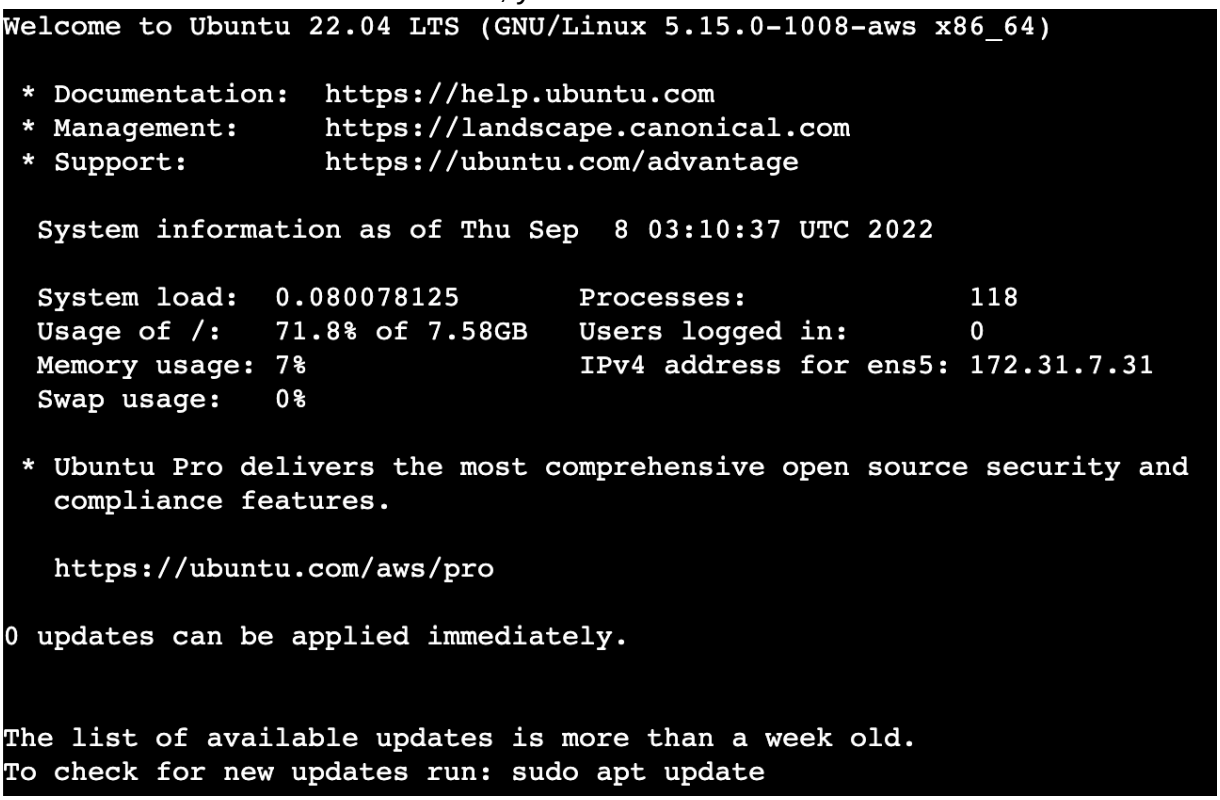
3. Connecting to your instance

Right-click your instance and click “Connect”.

Change the username to “ubuntu” and click “Connect” at the bottom-right as depicted below.

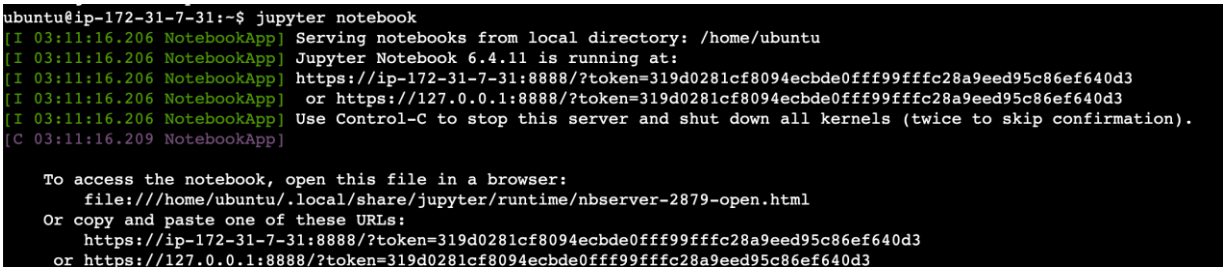


Once the connection is established, you will see the screen below:



Click into this terminal and type “jupyter notebook”. You will see a unique URL at the bottom of the screen. Note that your URL will not be exactly the same as the URL in this screenshot. **Do not close this window/tab otherwise you will no longer be able to access Jupyter Notebook.**

However, please note the service we are using will pause AWS instances after four hours. You can reset the timer by clicking “Start Lab” on Canvas > Modules > Learner Lab – Foundational Services if you would like to use AWS for an extended period of time. To regain access after it has automatically shutdown, follow the same instructions, by clicking on “Start Lab” and waiting for the AWS icon to turn green. All of your materials will be saved on persistent storage but it is important that you backup your files using Git and GitHub, as details in Section 4.



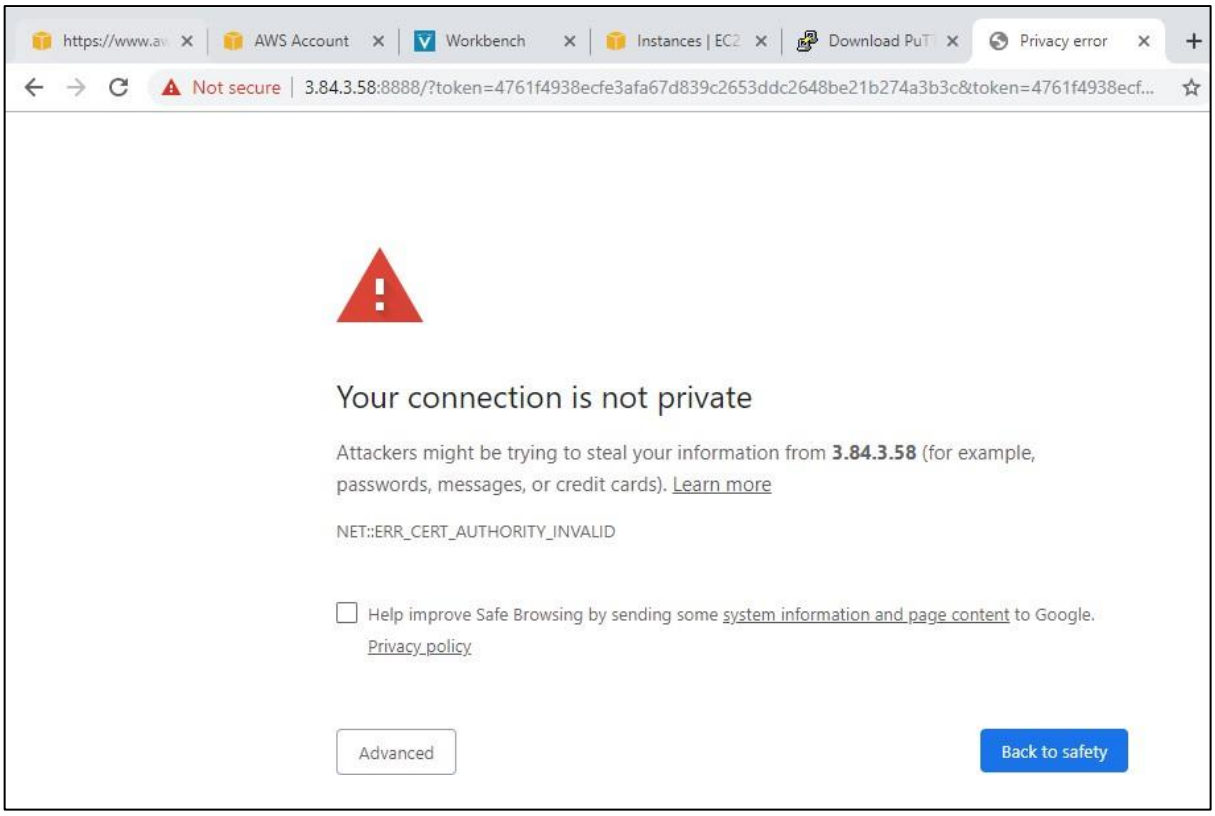
Jupyter Notebook is now running on an AWS instance in the cloud. To access, first copy and paste the Jupyter Notebook URL. Be careful not to copy both URLs by accident. Replace the highlighted part of the copied URL below with your public IP address (noted down previously).

```
https://ip-172-31-7-31:8888/?token=319d0281cf8094ecbde0fff99fffc28a9eed95c86ef640d3
```

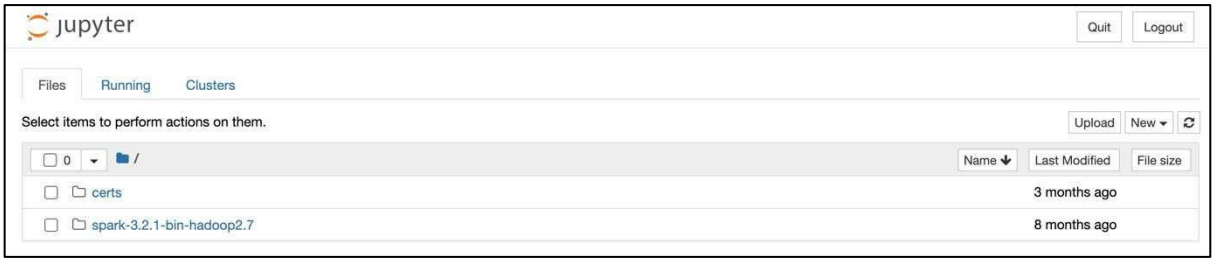
It should look similar to this (note that your public IP address will be different).

```
https://18.205.7.167:8888/?token=319d0281cf8094ecbde0fff99fffc28a9eed95c86ef640d3
```

Also note that if you are using Chrome, the following screen will pop up. To get past this screen, you can override this security measure by typing ‘thisisunsafe’. Do not type this into the URL bar, click on the screen and begin typing. Other browsers should work fine.



If you do this correctly, the page should refresh and you will be able to see the Jupyter Notebook window below. You are ready to get started with your labs!



4. Working with GitHub

Git is used for version control while GitHub is a hosting service for Git repositories. It's useful when you're working with a team, but it's still a great tool when you're working by yourself. The lab machines as well as the EC2 instance you're working on have Git installed.

1. As an initial task, let's introduce you to some basic Git commands including git clone, git add, git commit, and git push. After creating an account on GitHub, follow along with this video created by the 722 team.
    a. **Git Clone:** Git clone allows you to clone an already existing repository. Because Git encourages open-source, you have the option to clone code that already exists.
    b. **Git Add, Commit and Push:** You need to type in all three of these commands to push your data from your local repository to GitHub. If you'd like some information on the differences between the three, click here.
2. If you need more information, try going through this tutorial.
3. Note that Git is the simplest way to have a backup of your code/outputs that you'll be developing on your AWS EC2 instance. **It is your duty to keep backups of your code and outputs.**

Now that you are comfortable with the basics of Git and GitHub, you can use the clone command to get access to the lab materials.

1. First, we need to request access to Shohil's GitHub repository as it is a private repository. To request access, fill out the spreadsheet here. Alternatively, email your Course Coordinator, Ajay.
2. Once you've been given access (usually within 48 hours), login to your EC2 instance and type in the commands below.
    a. Type "mkdir 722" to create a new folder named 722.
    b. Then, "cd 722" to move into that folder.
    c. Then, type "git clone https://github.com/shohil-kishore/aws-instance" to clone Shohil's instance.
    d. Note that GitHub will ask you to input your username and password at this stage. Input your username as is, but for the password, you need to generate a personal access token. Detailed instructions can be found here.
    e. If there's an error, please try again. If it still doesn't work, please screenshot the error and send it to your Course Coordinator, Mina.

Note that it is best practice to back up your files using GitHub regularly, especially as all AWS instances shutdown after four hours automatically.