

Getting Started with Amazon Web Services (AWSs)

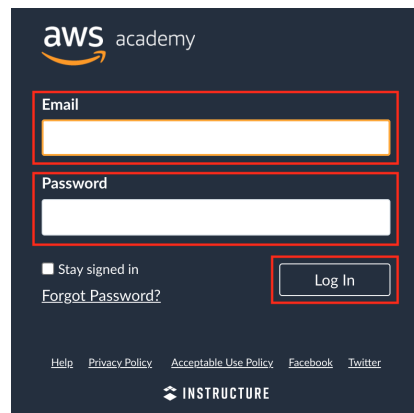
August 29, 2024

1 Introduction

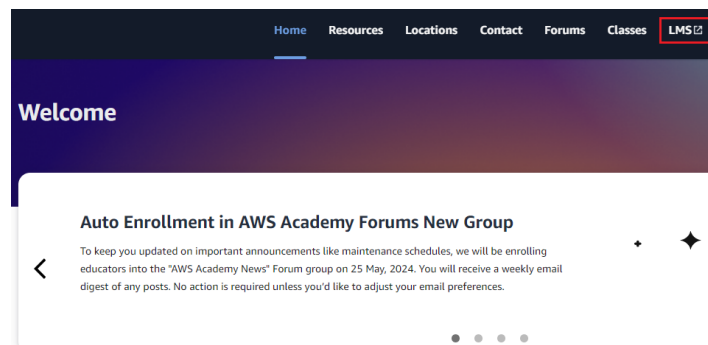
This article explains how get access to a Linux environment using AWSs through *AWS Academy Learner Lab (AWS Lab)*. You will use this platform to work on your assignments for EECS 489 during this semester.

2 Using AWS Academy Learner Lab

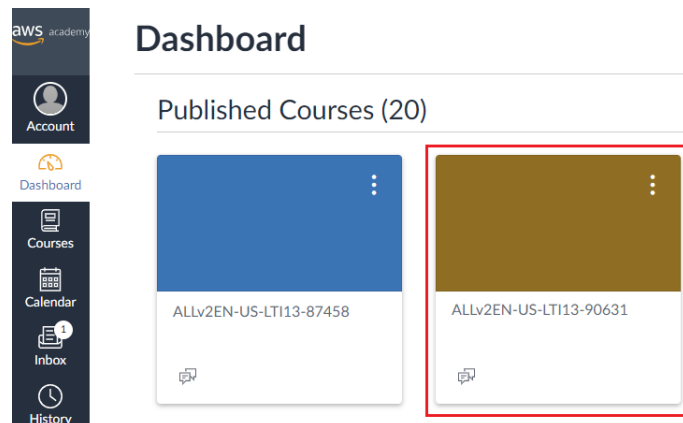
Step 1: You should have received an email invitation that explains how to set up your AWS Lab account. Once you have set it up, go to AWS Lab portal at https://www.awsacademy.com/LMS_Login. Select **Student Login**. Type in your email and password to log in.



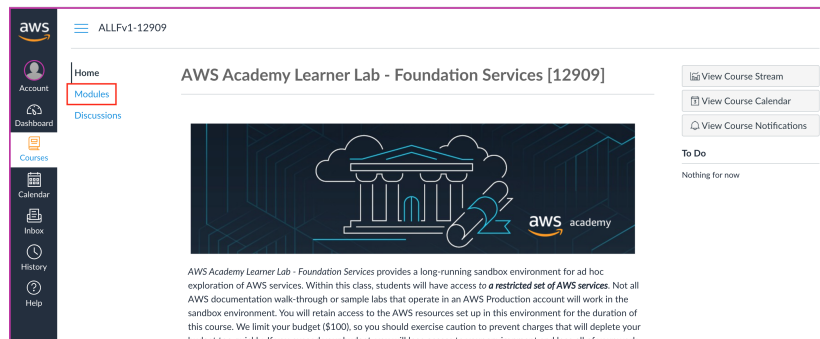
Step 2: Once you have logged in, you should see a welcome screen as given below. Click on the LMS link given at the top right corner



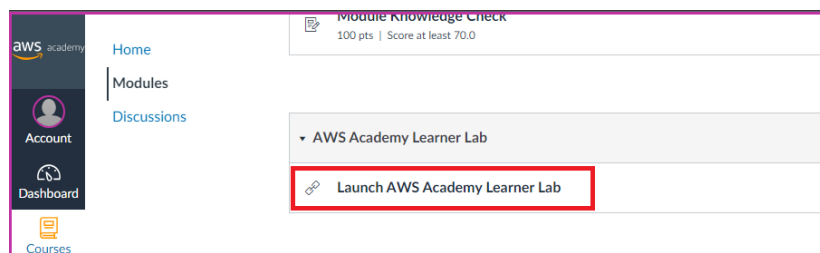
Step 3: This will take you to the AWS Lab dashboard, which looks like the following. You will see a list of courses. Select the one as highlighted below (90631).



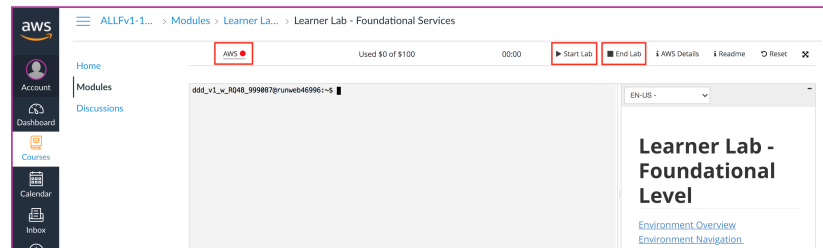
Step 4: It will take you to the course homepage. Select **Modules**.



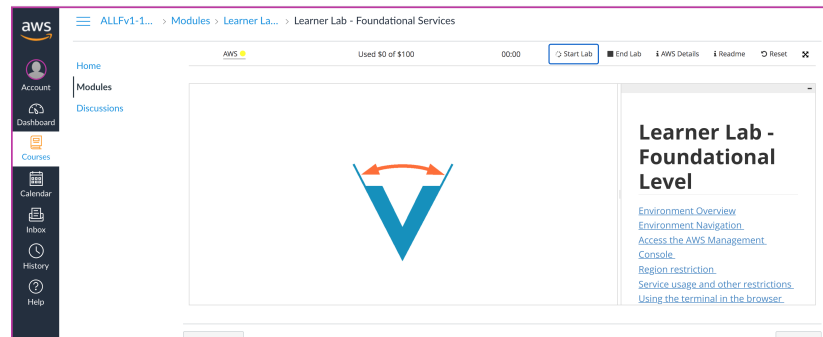
Step 5: Select **Learner Lab - Foundational Services**



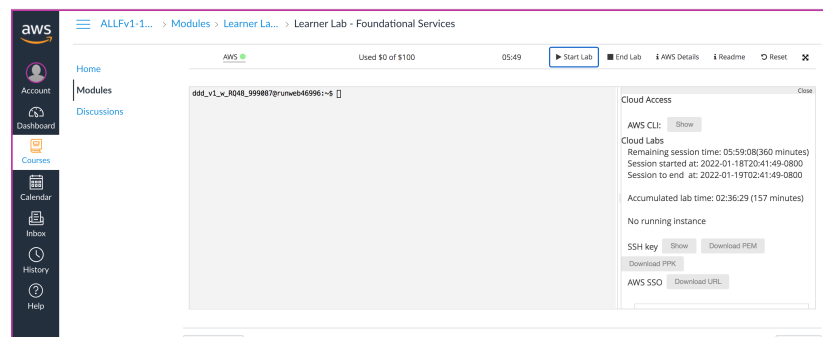
Step 6: This is your lab interface, where the important dials are highlighted using red boxes. The *Readme* includes a collection of helpful documentation on using the lab environment.



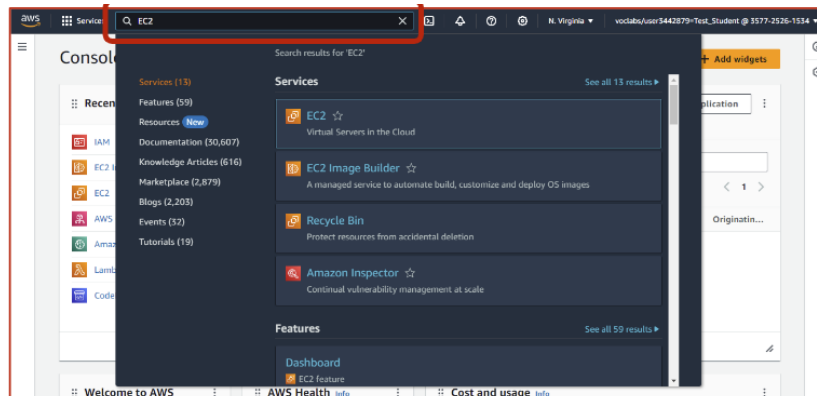
Step 7: Select **Start Lab** to start the lab. This may take several minutes the first time..



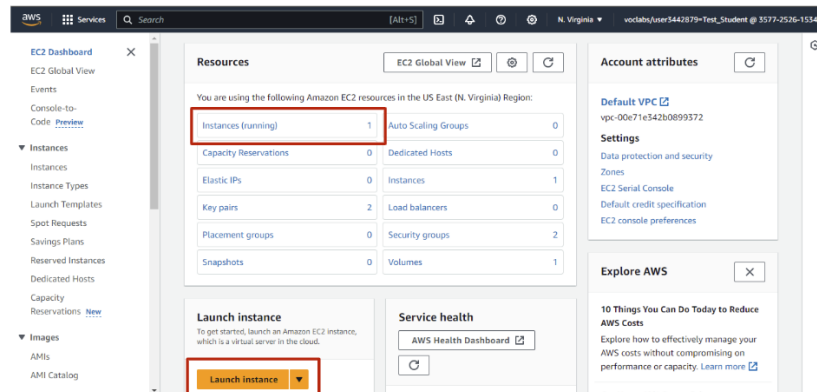
Step 8: Click on **AWS** after the circle next to it has turned **green**.



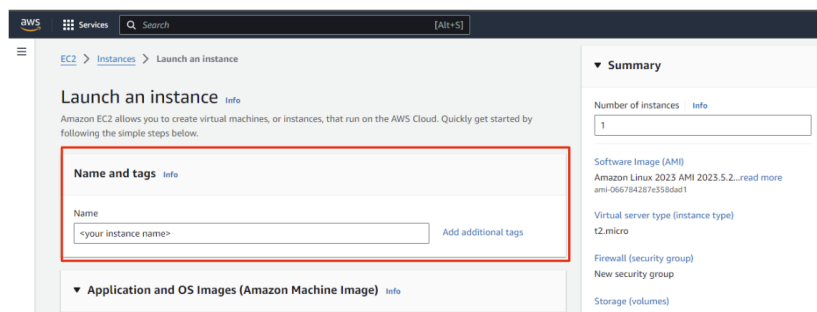
Step 9: AWS Console Home opens on a new browser tab. After you are in the AWS console, search for “EC2” in the search bar, and then click on it.



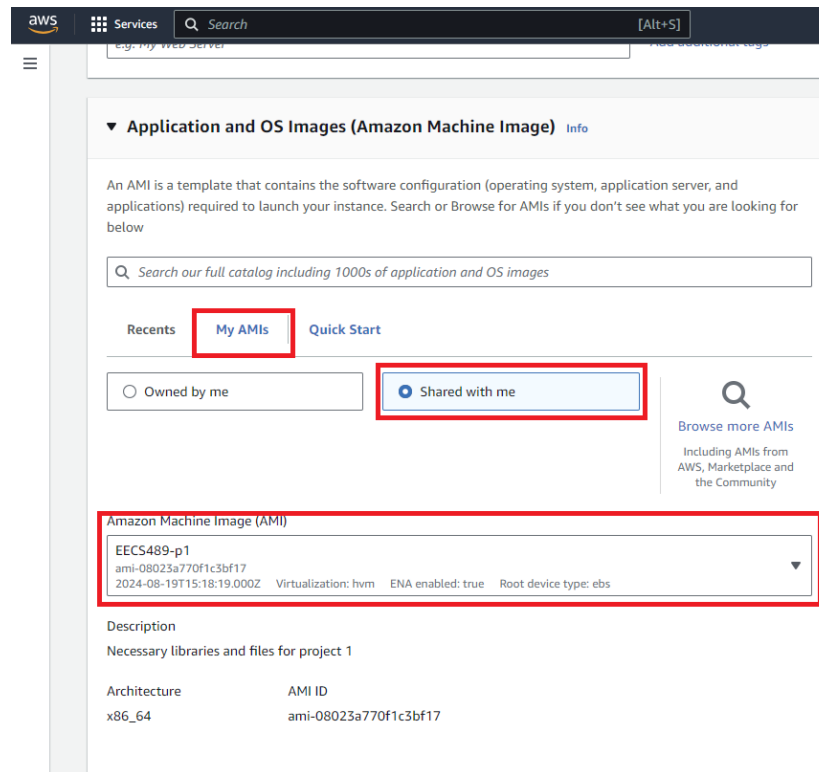
Step 10: this will take you to the following screen. To create a new virtual machine, select **Launch an instance**



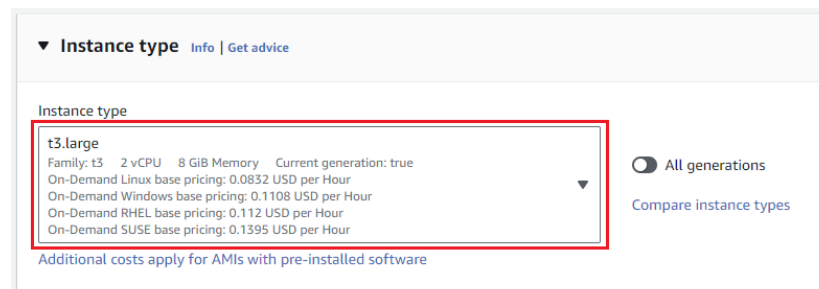
Step 11: Give a name to the new instance you are about to create.



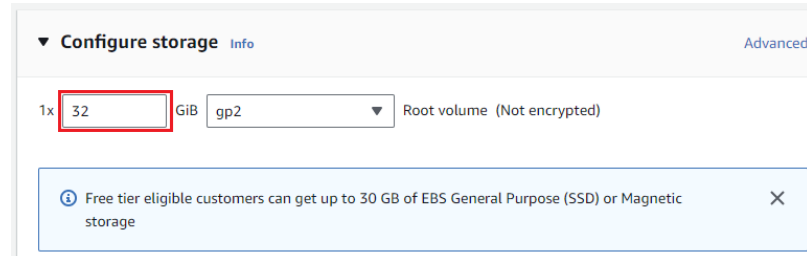
Step 12: To follow the virtual machine creation workflow, first choose an Amazon Machine Image (AMI), which is a blueprint of a preconfigured machine that is ready to run, when instantiated. Click on “My AMIs“, check “Shared with me“ option and then pick “EECS489-p1“ from the AMI drop down menu.



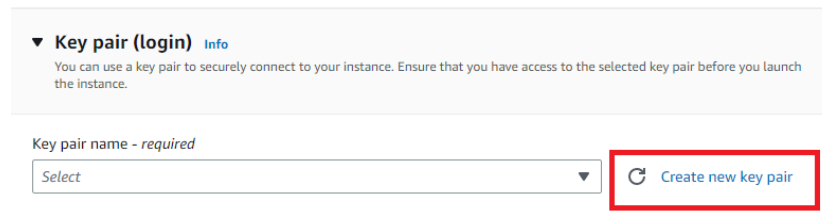
Step 13: Choose the instance type. Select *t3.large*.



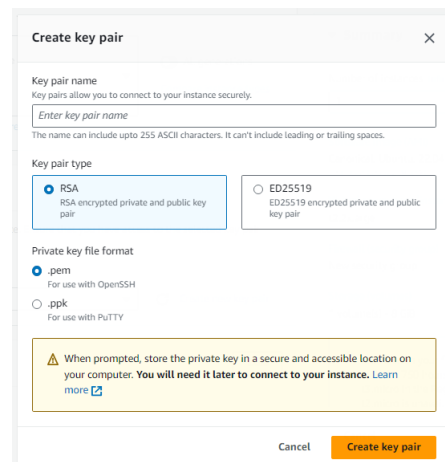
Step 14: Scroll down to **Configure storage** and type 32 in the window highlighted.



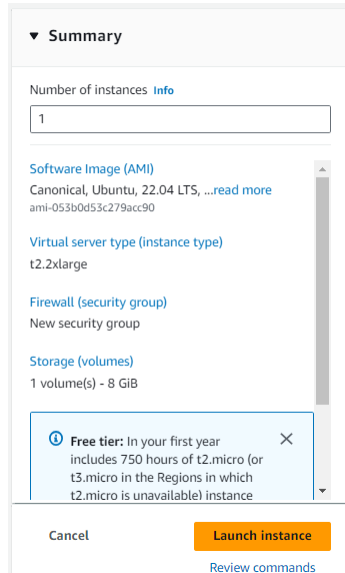
Step 15: From *Key pair (login)* section, select **Create a key pair** if you do not have an existing key pair.



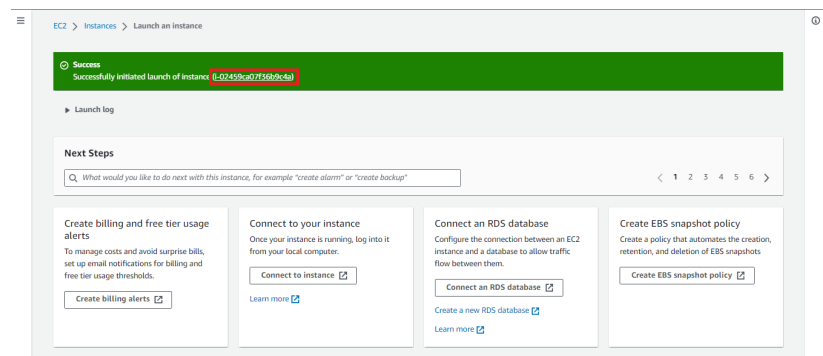
Step 16: From the pop-up, give your key pair a name and select **Create key pair**. This will download the public key ([Key pair name].pem) that is required to securely connect to your virtual machine, from your local computer. **Important: Do not lose or damage this public key. Without it, you cannot connect to your virtual machine, and you WILL NOT be able to download this key file again.**



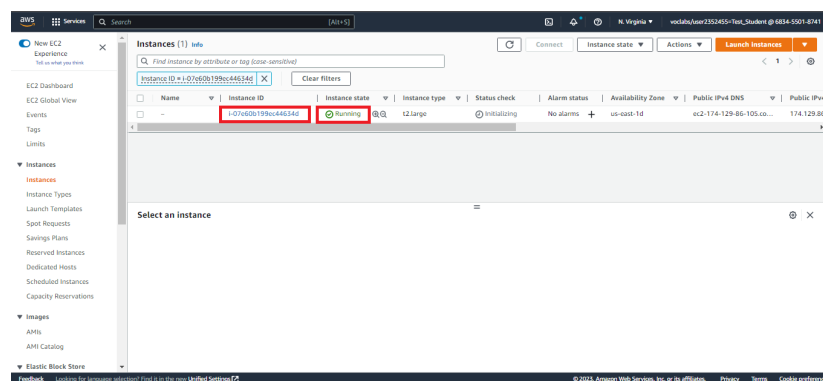
Step 17: From *Summary* section, select **Launch Instance** to start your virtual machine.



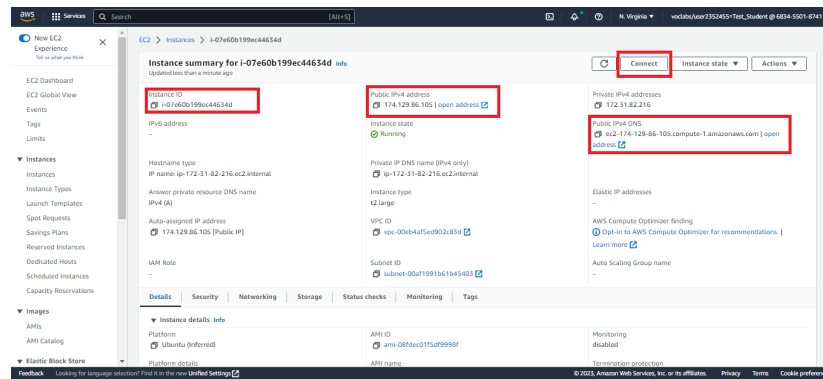
Step 18: Select the instance identifier when you get to **Launch Status** page.



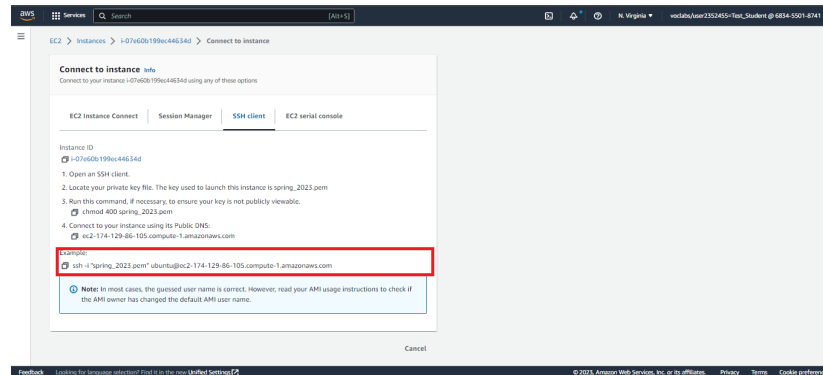
Step 19: The instance status is **Running**. If not, it may take a little while to get to that state. Select the instance identifier from the **Instances** view.



Step 20: Here, you can see some vital information about your new EC2 machine instance such as, its id, public IPv4 address, public IPv4 DNS, etc. Select **Connect** to see how to connect to your virtual machine from your local computer.



Step 21: Under **SSH client** you can find the steps that you have to follow to connect to your virtual machine. You can copy and paste the highlighted `ssh` command into the terminal to initiate the connection.



Step 22: When you see the following prompt on the terminal the *first time* you run the `ssh` command, type in **yes** and hit **return**. You should make sure that you give the full path to the `.pem` file you have saved.

```
Rakin@LAPTOP-I3UM4DUG MINGW64 ~/Desktop
$ ssh -i "/spring_2023.pem" ubuntu@ec2-174-129-86-105.compute-1.amazonaws.com
The authenticity of host 'ec2-174-129-86-105.compute-1.amazonaws.com (174.129.86.105)' can't be established.
ECDSA key fingerprint is SHA256:5A0haUuCPmHgE0LE+Zs7tYrXwt11b+XxyLLF/Vw+9Lc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```


Step 23: Voila! Now you are on your remote virtual machine and can (responsibly) use it!

```
* Support:      https://ubuntu.com/advantage

System information as of Sun Jan 15 16:23:42 UTC 2023

System load:  0.0          Processes:           104
Usage of /:   16.4% of 7.57GB Users logged in:       0
Memory usage: 2%          IP address for eth0: 172.31.82.216
Swap usage:   0%

0 updates can be applied immediately.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-82-216:~$
```

Step 24: It is very important to keep in mind that you must end your lab on AWS Lab when you do not use it. You can use the **End Lab** button on the lab console to end a lab. When a lab ends, AWS Lab shuts down all the AWS services (e.g., virtual machines) associated with the lab. However, when you start the lab again, AWS Lab brings up the AWS services as well. With regards to virtual machines, every time AWS Lab boots them up, they might get new network vitals, even though their PKI key pairs remain unchanged. You can follow [Step 20:](#) to [Step 23:](#) above, to connect to your virtual machine(s) every time you reboot them as a result of restarting an AWS Lab.