AWS Create Ubuntu 20 Elastic Compute Instance

In this tutorial, we will be creating an AWS Elastic Compute (EC2) instance with Ubuntu 20 as the operating system.

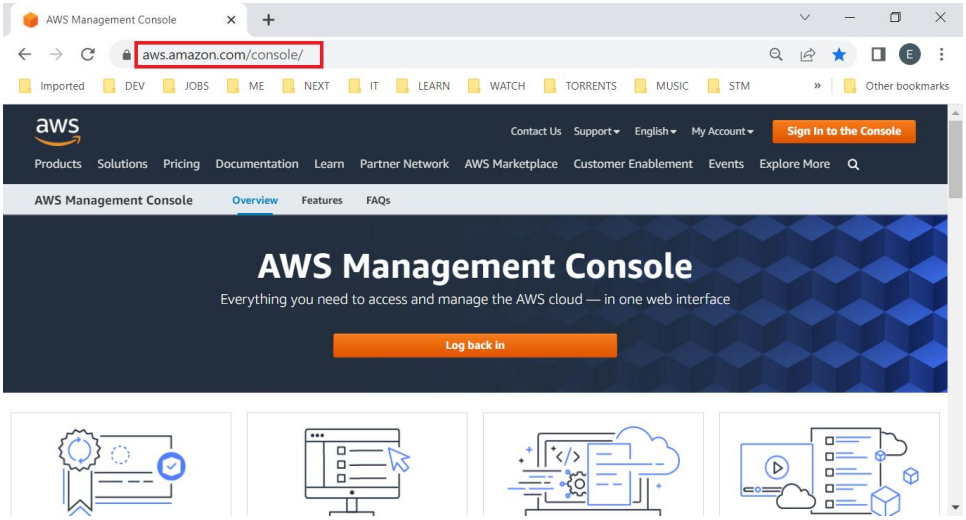
To complete this tutorial you will need an AWS Free Tier account.

If you do not have an AWS account, you can access my "AWS Create Free Tier Account" tutorial here.

Steps to complete tutorial:

- Update Virtual Private Cloud (VPC)
- Create Ubuntu 20 EC2 Instance
- Connect to Ubuntu 20

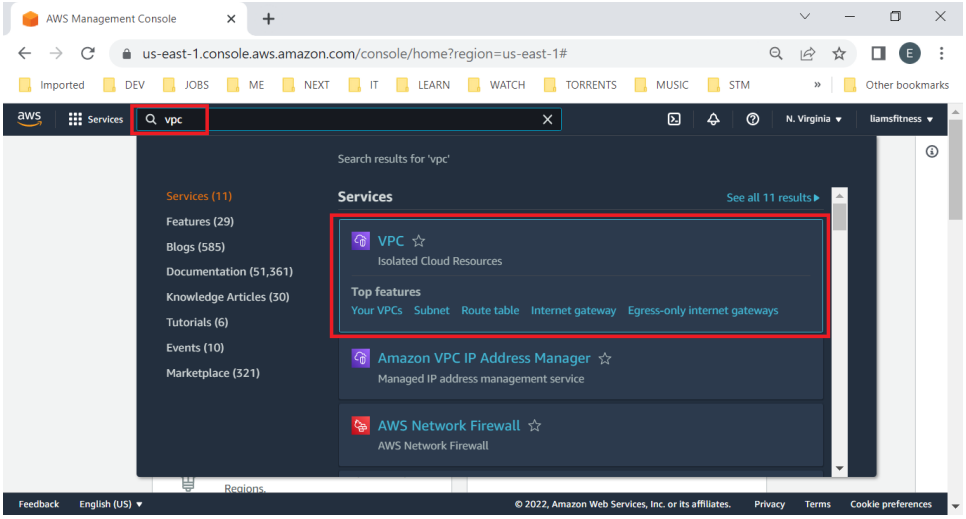
To begin, go to the following website, https://aws.amazon.com/console/ and log in to the console.



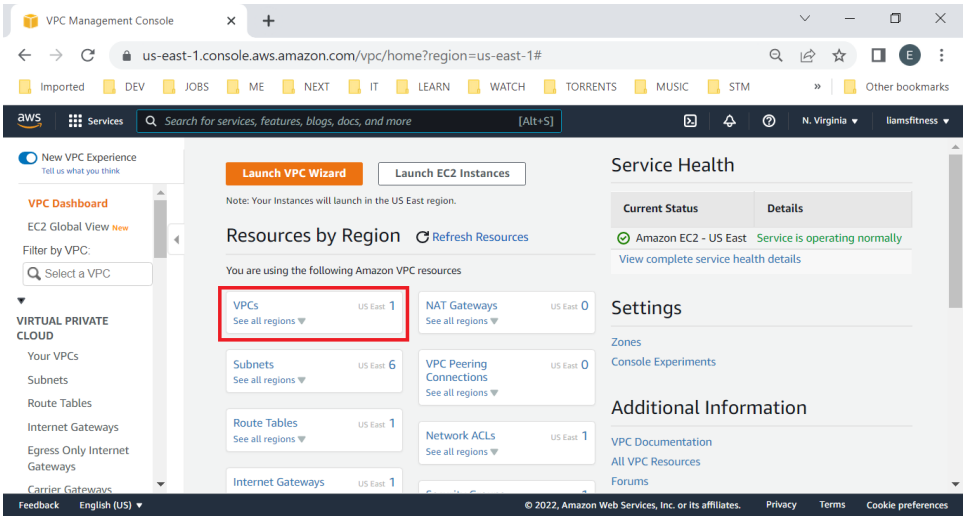
Before we create the EC2 instance, we will configure our default VPC (Virtual Private Cloud) so that it will be easier to work with moving forward. The VPC will allow us to launch resources in an isolated virtual network.

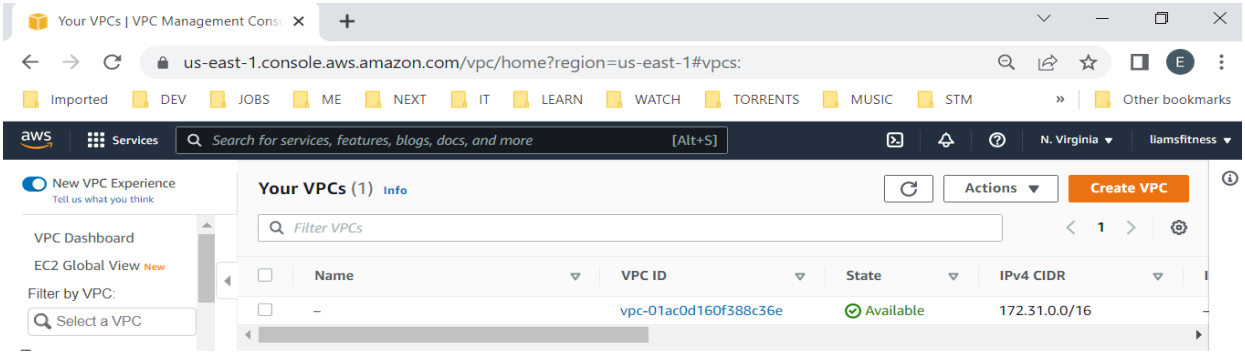
Update Virtual Private Cloud (VPC)

Once logged in, enter VPC in the search bar and select "VPC Isolated Cloud Resource"

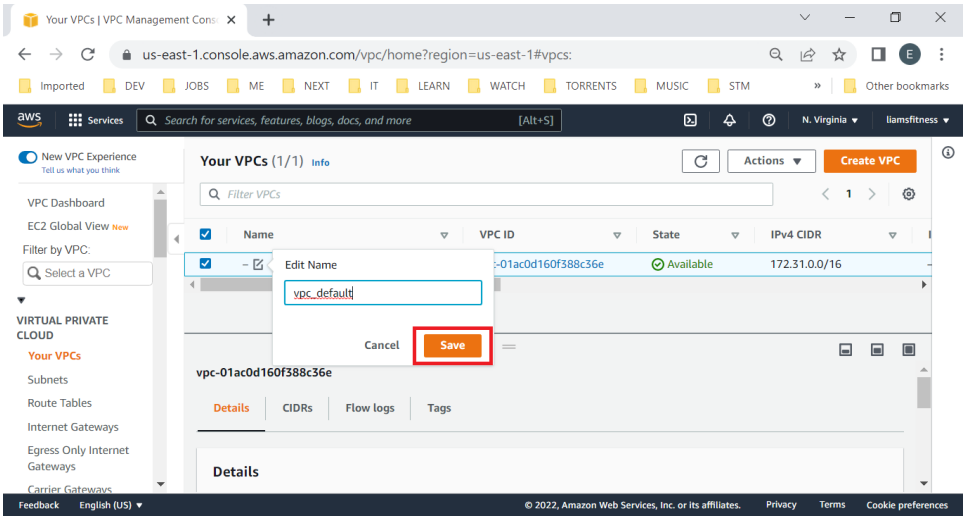


On the VPC Dashboard, select "**VPCs**"

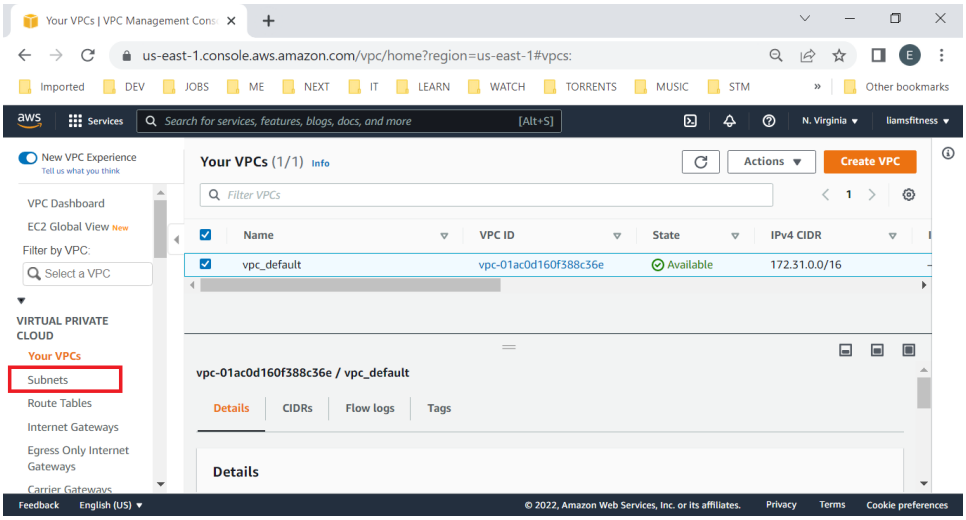




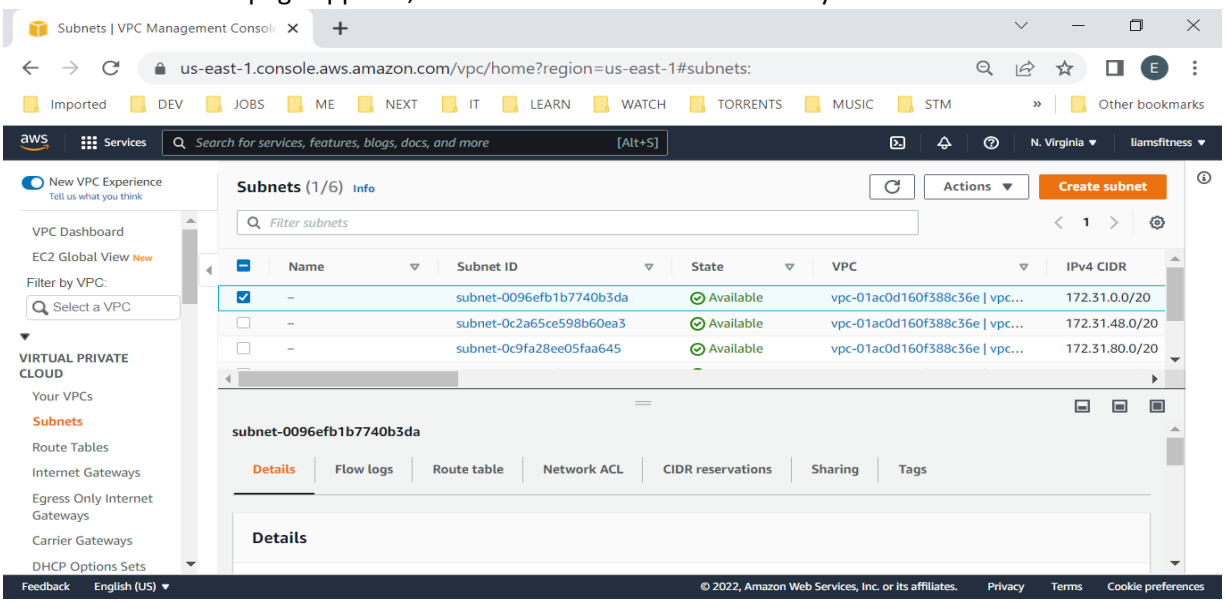
Let's set the name to "**vpc_default**" and click save.



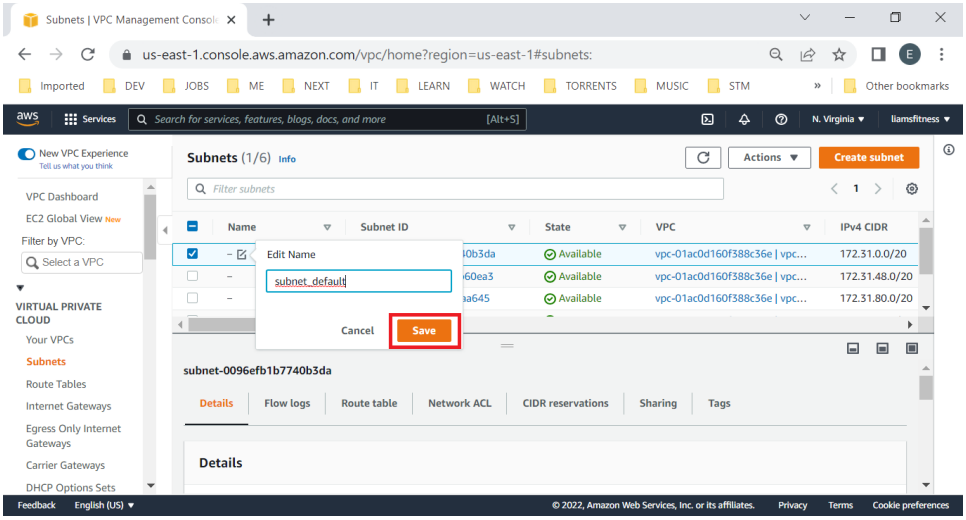
Now, on the left hand side of the screen click "**Subnets**"

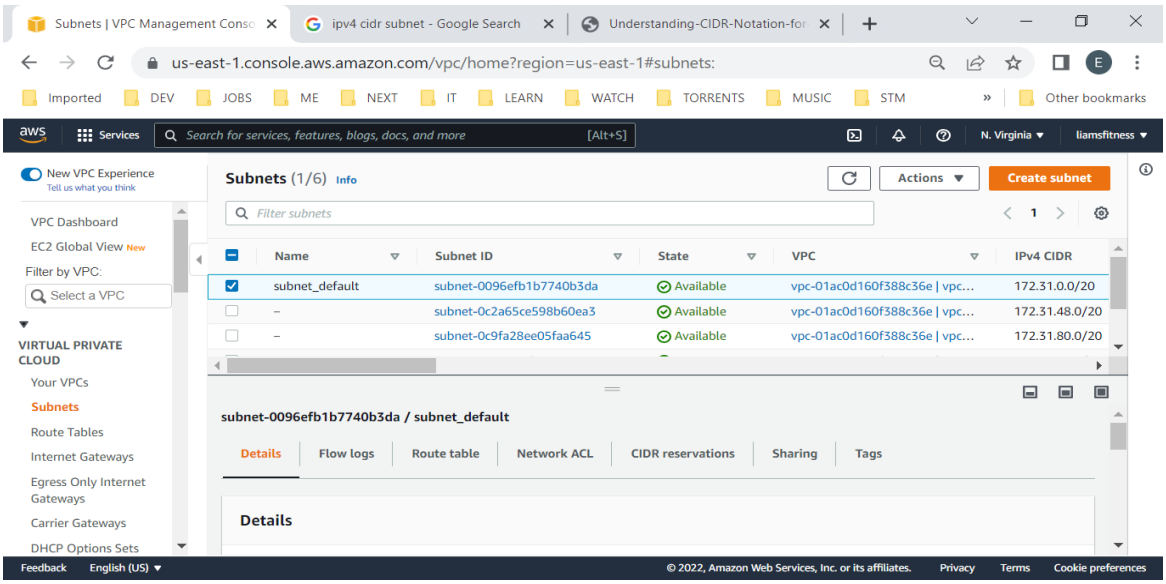


Once the subnet list page appears, notice that 6 subnets have already been created for us.



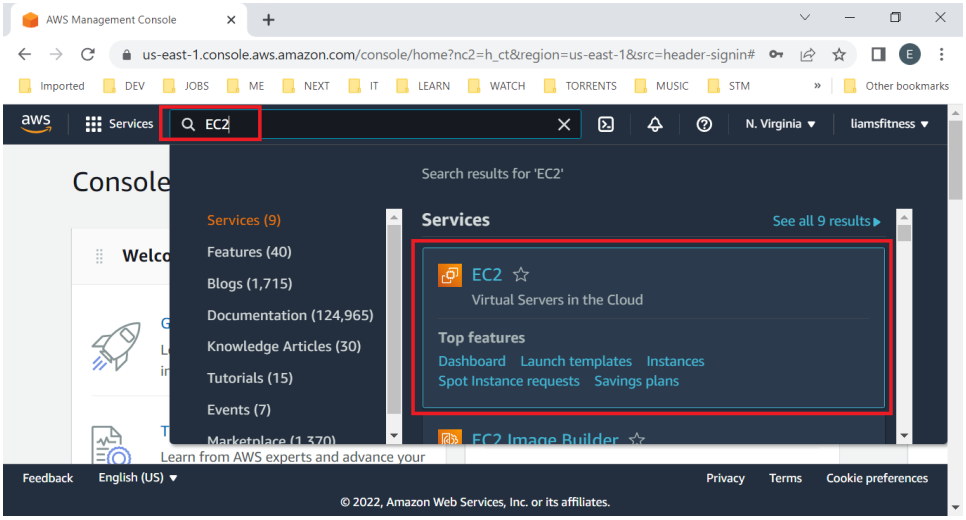
Set the name of the first subnet to "**subnet_default**" and click save. Also note that the CIDR IPv4 subnet is 172.31.0.0/20.





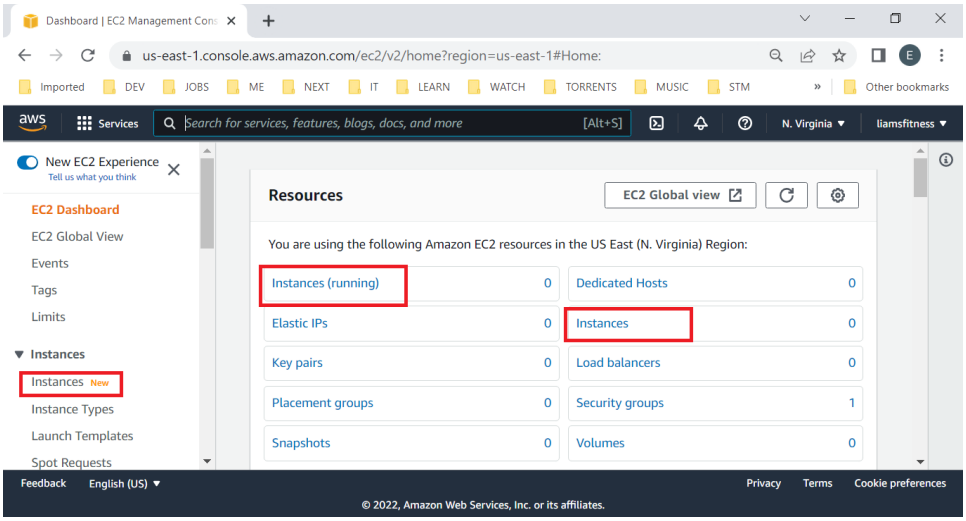
Create Ubuntu 20 EC2 Instance

Now we can proceed to create our instance, enter EC2 in the search bar and select the 1st EC2 listing.

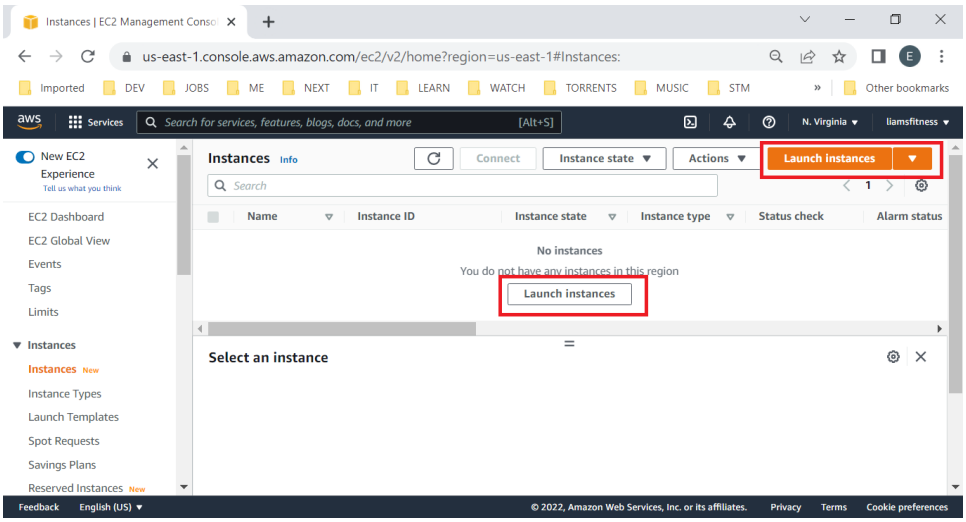


You will be brought to the EC2 Dashboard. It contains links to the resources being used in the selected AWS region. In my case it's US East (N. Virginia).

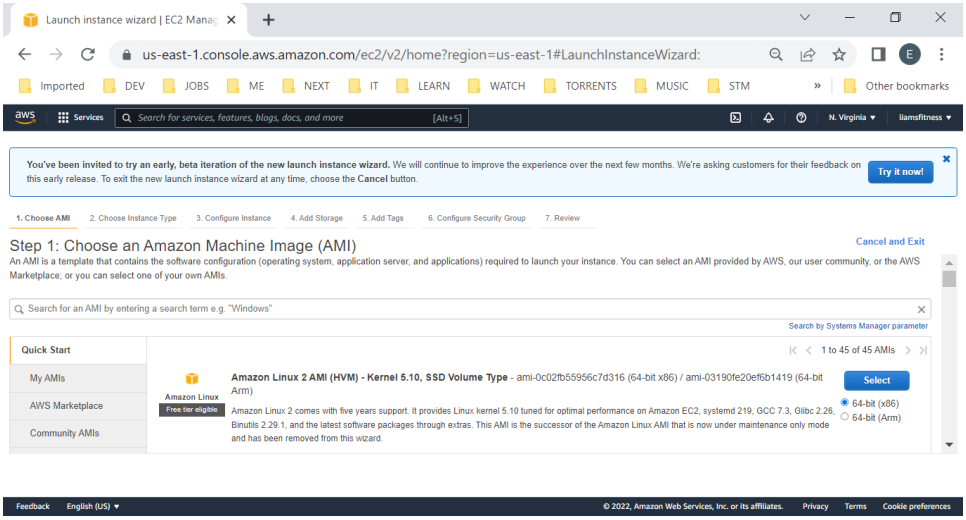
From the EC2 dashboard, click "**Instances**" (all links will work, your choice).



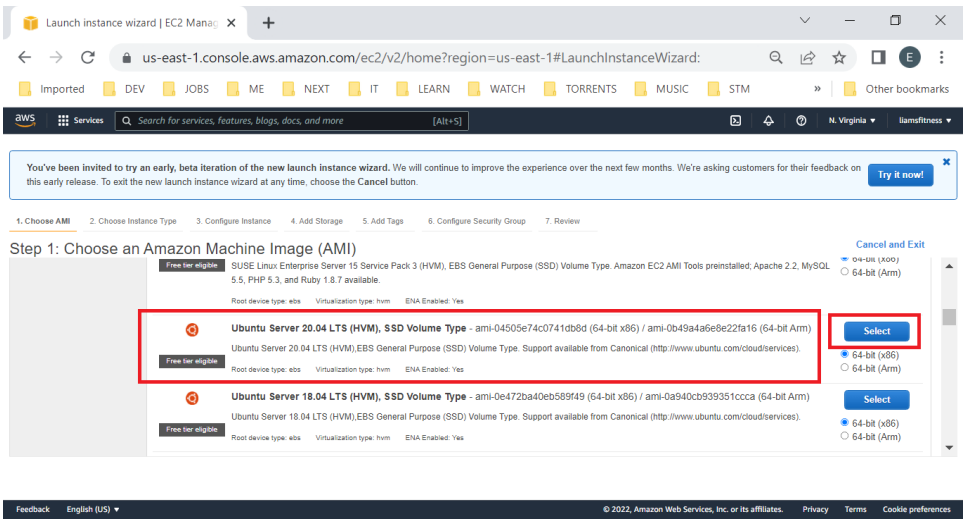
Next, click "**Launch Instances**" (either link will work, your choice).



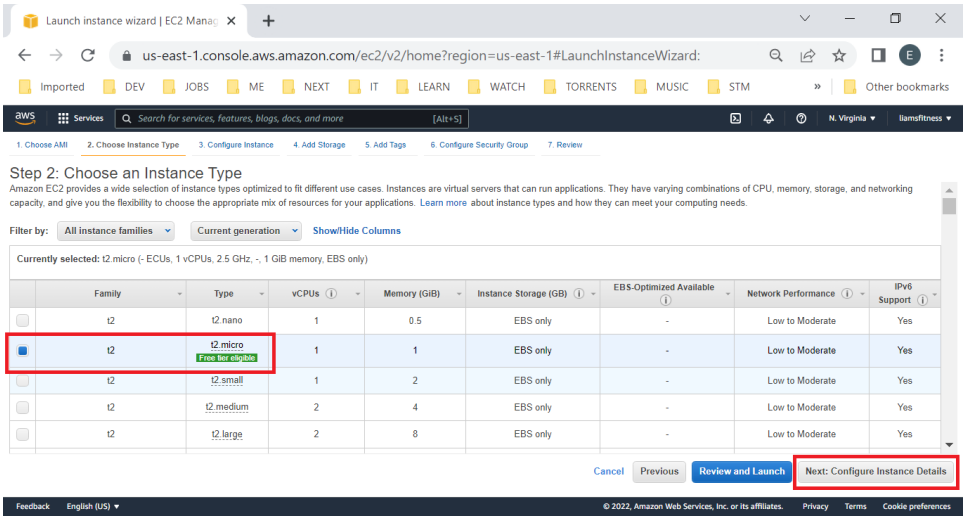
The first step will be to select the image we want to use.



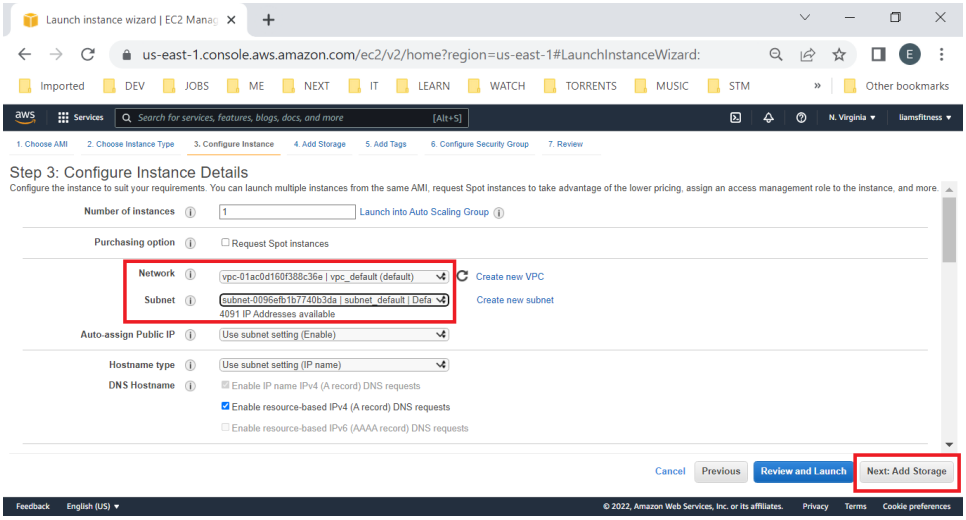
Scroll down the page until you locate "**Ubuntu 20 LTS**" and click the "**Select**" button next to the listing.



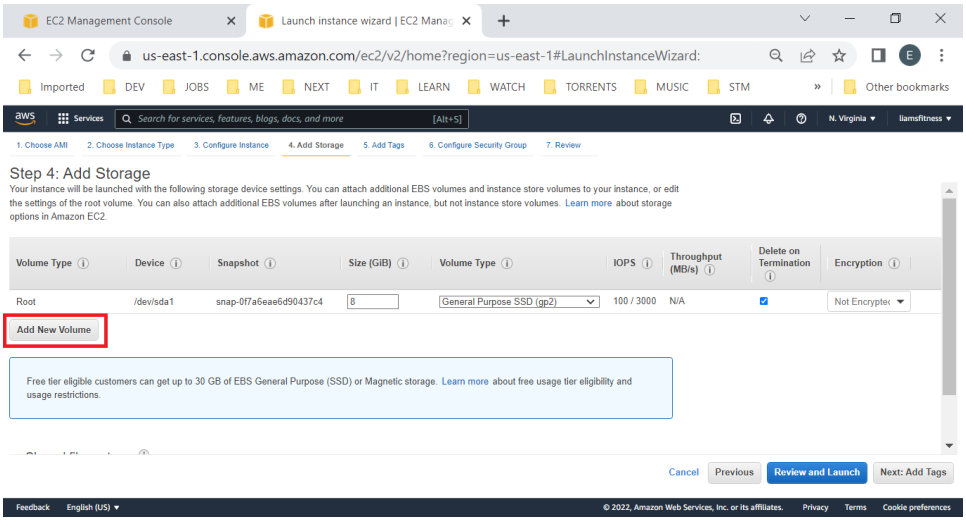
In the second step, we choose the instance type. Since we want a free instance, select "**t2.micro**" and click "**Next: Configure Instance Details**".



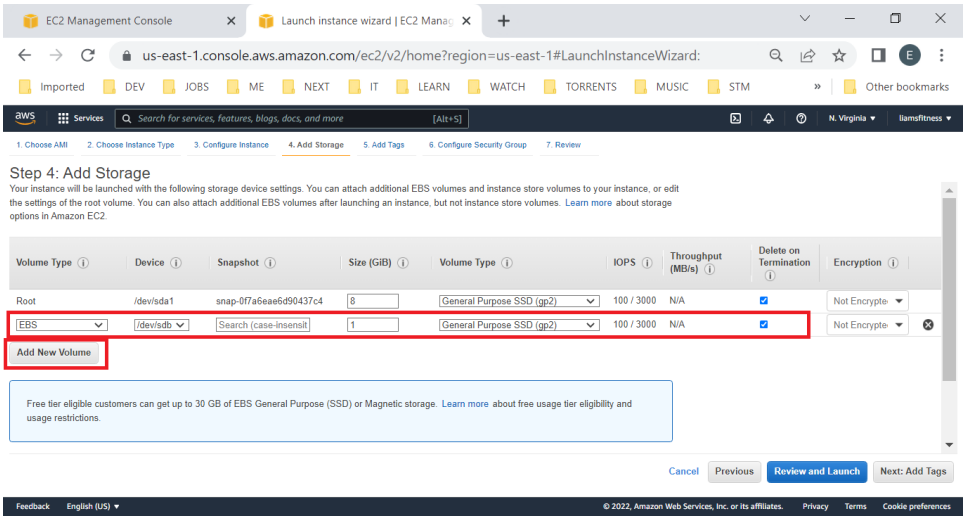
The third step allows us to select the VPC (**vpc_default**) and subnet (**subnet_default**) that we named earlier. After making those changes, click "**Next: Add Storage**".



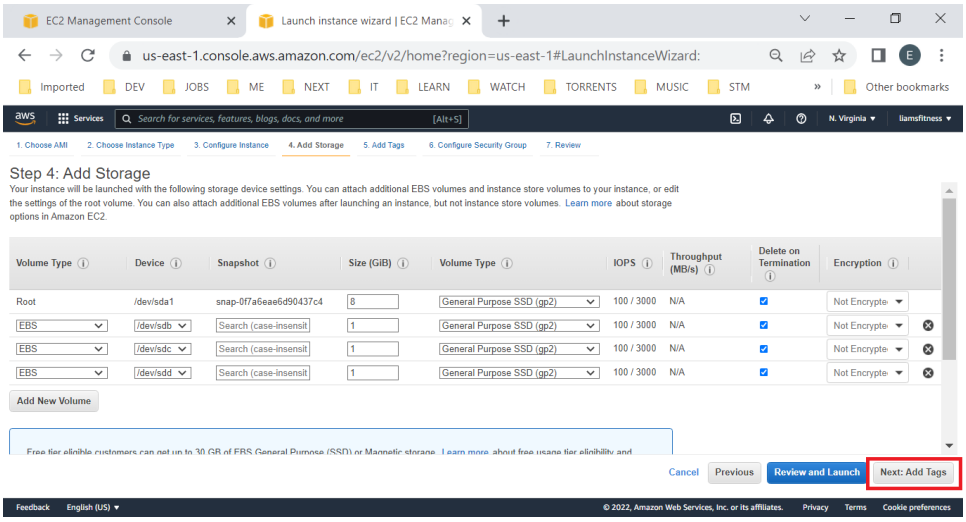
The fourth step allows us to set the storage size and add volumes if we wish. The root volume of 8GB is enough for Ubuntu 20. In a future tutorial, I will be demonstrating disk partitioning, as well as, LVM management, so I will also add 3 additional volumes.



After clicking the "**Add New Volume**" button, I set the size to 1GB and checked "**Delete on Termination**".

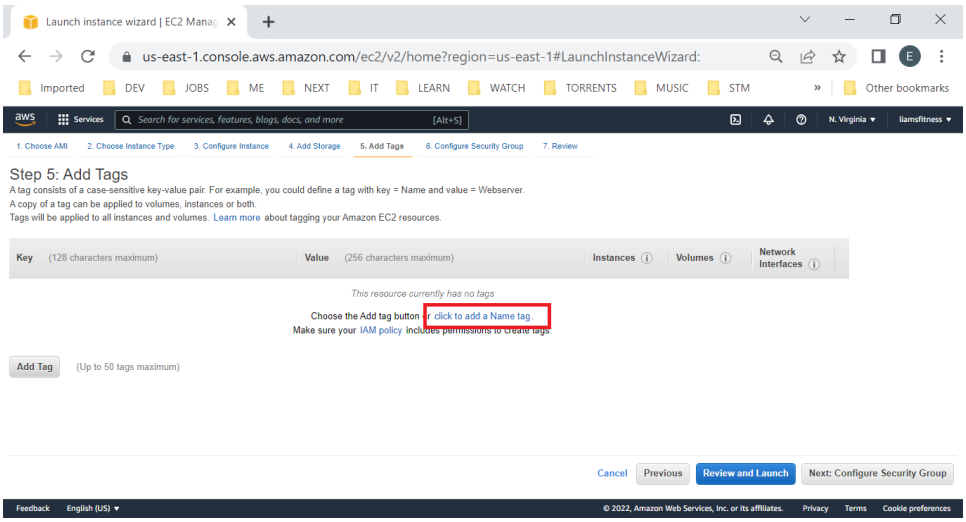


I repeated these steps 3 times total. After setting the storage, click "**Next: Add Tags**".

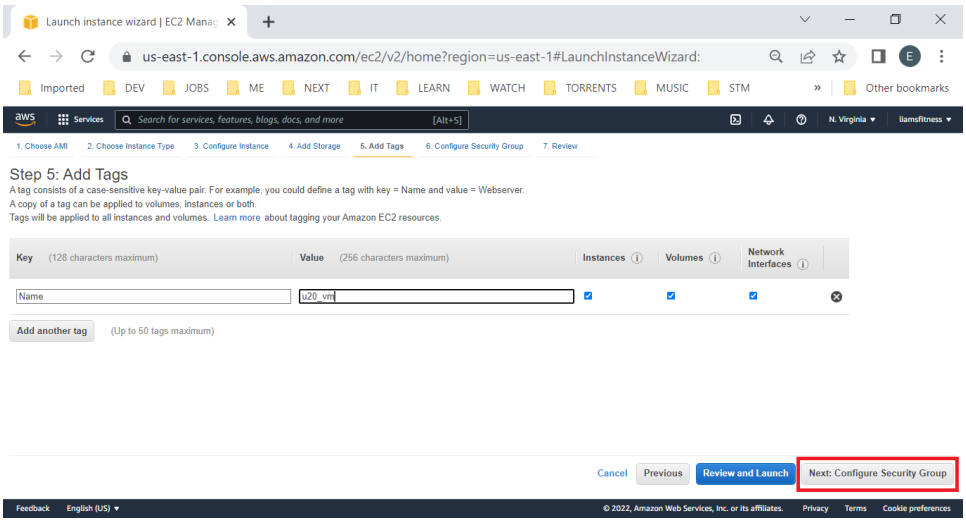


***Note*: *There is a 30GB max volume size for all your instances combined throughout the month. If you go over it, you will pay the cost.*

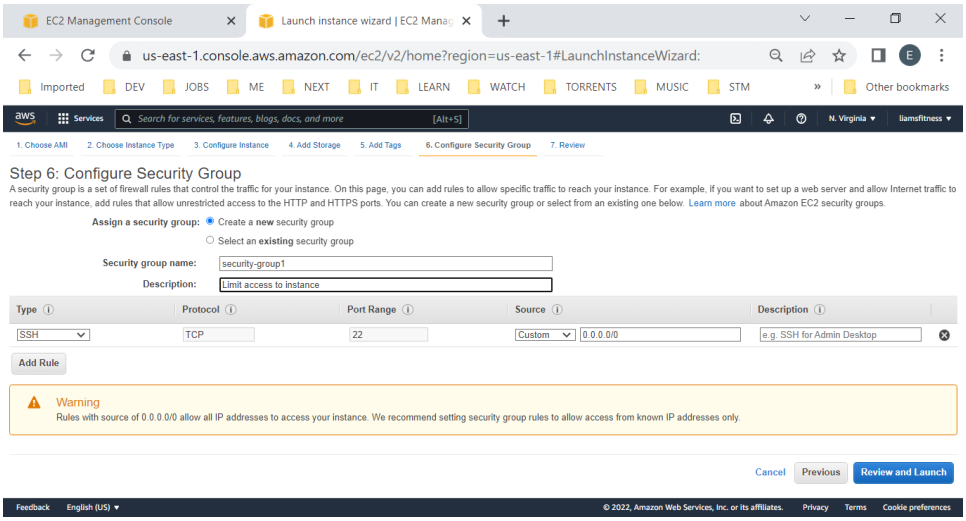
The fifth step allows us to tag our instance. Tagging helps categorize our resources. We will keep it simple by adding a name tag. To do this click the "**click to add a Name tag**" link and set the name to whatever you desire.



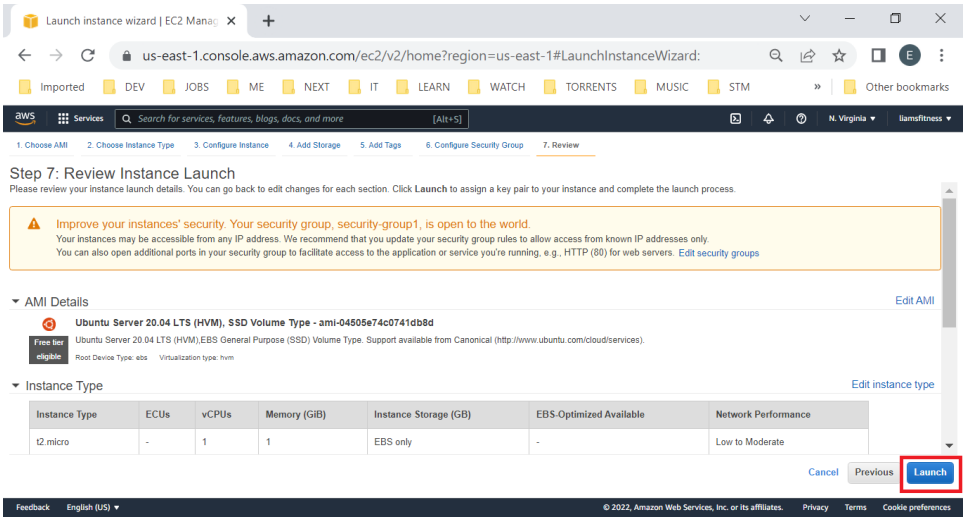
Once finished, click "**Next: Configure Security Group**".



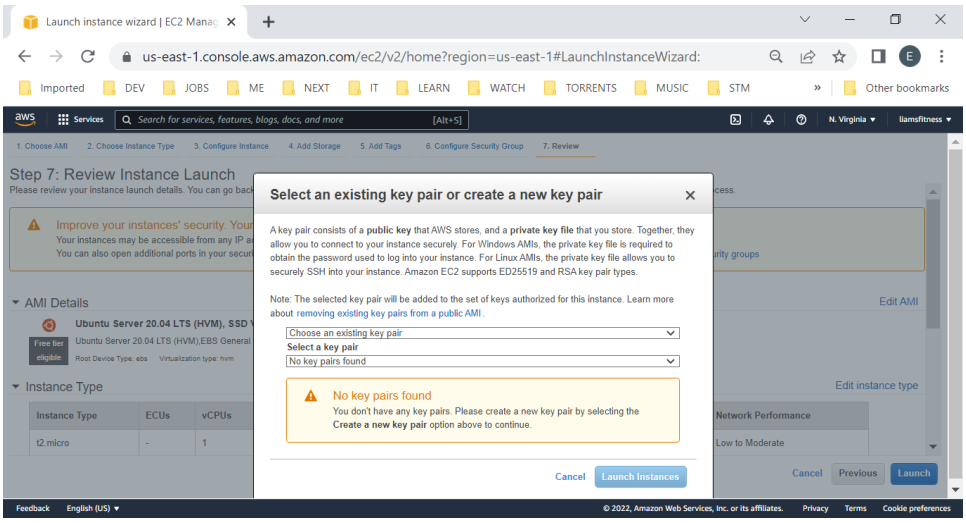
In the sixth step, before launching, we will create a security group named "**security-group1**". Note that only port 22 is open and that we are only allowing SSH connections to our instance. I also provided a small description, "**Limit access to instance**". After you've set the security group name, click "**Review and Launch**".



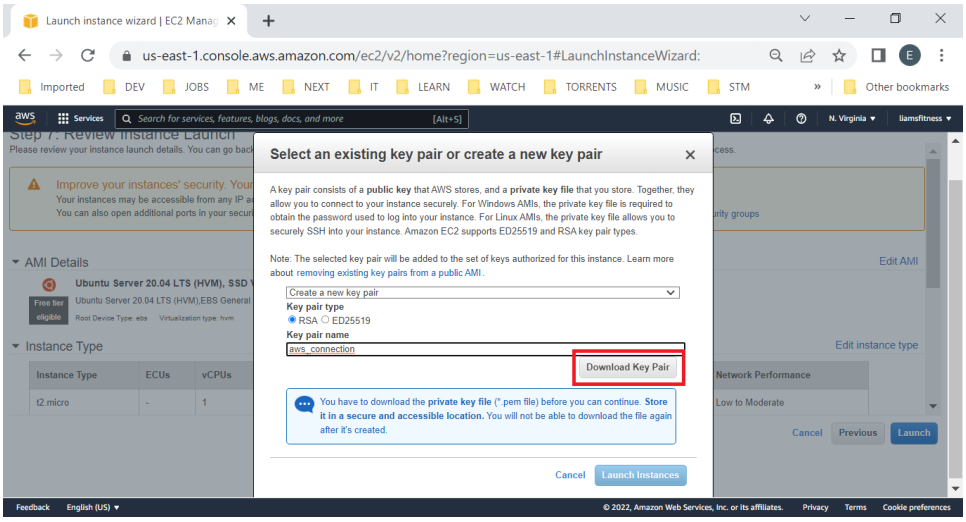
This final step allows you to review your selections. After you've finished reviewing, click "**Launch**".



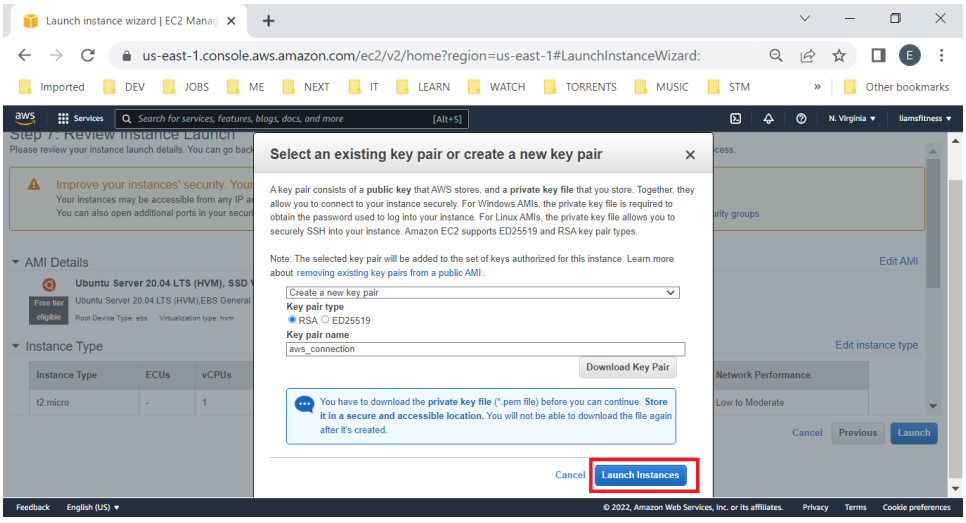
Prior to launch, we must create a key pair that can be used with an SSH connection to access our newly created instance.



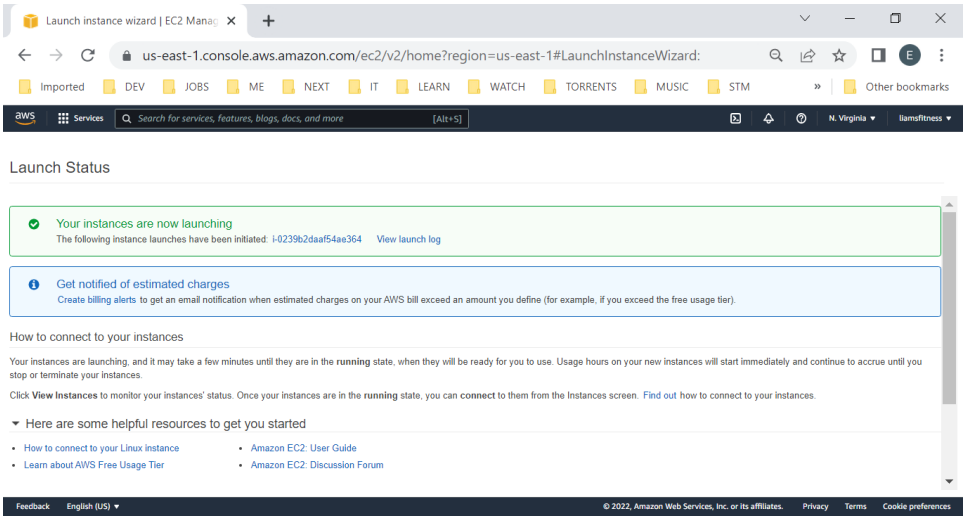
Ensure "**Create a new key pair**" is selected. Also, ensure that "**Key pair type**" is set to "**RSA**". Then, give it a name (**aws_connection**). Finally, click "**Download Key Pair**" and save it somewhere on your PC.



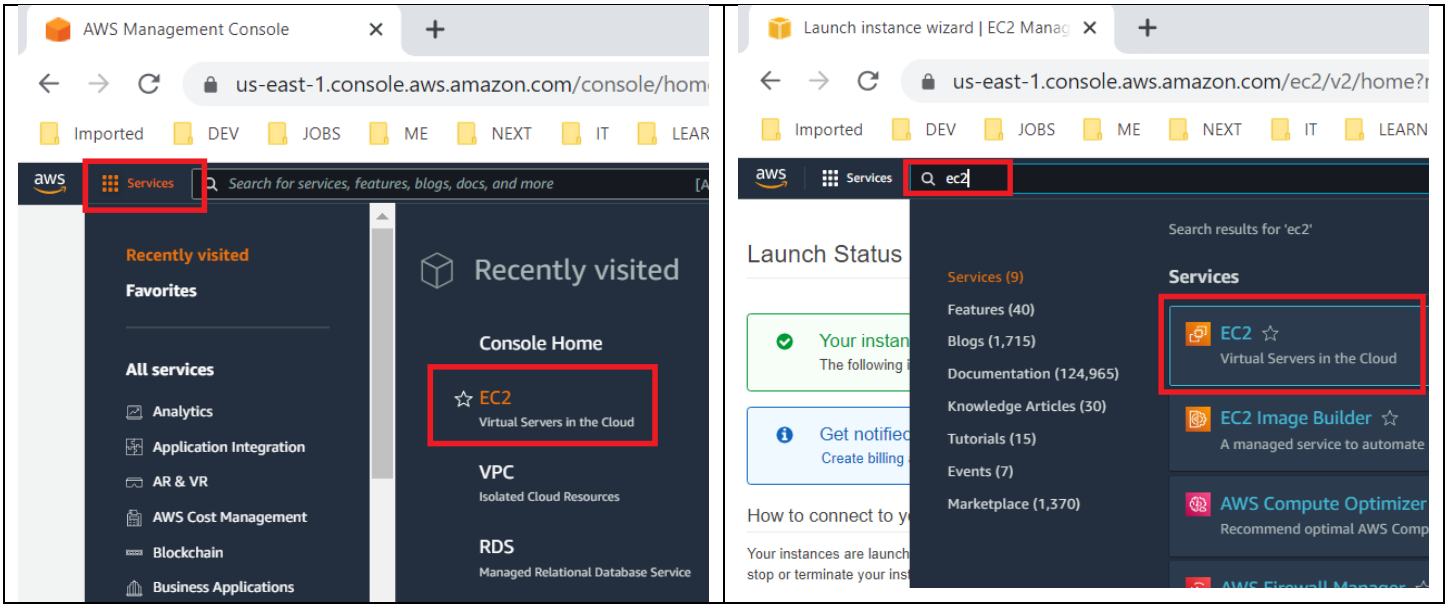
After you download the key pair (which will be named "**aws_connection.pem**"), you will be able to click "**Launch Instances**" to bring up our Ubuntu 20 instance.



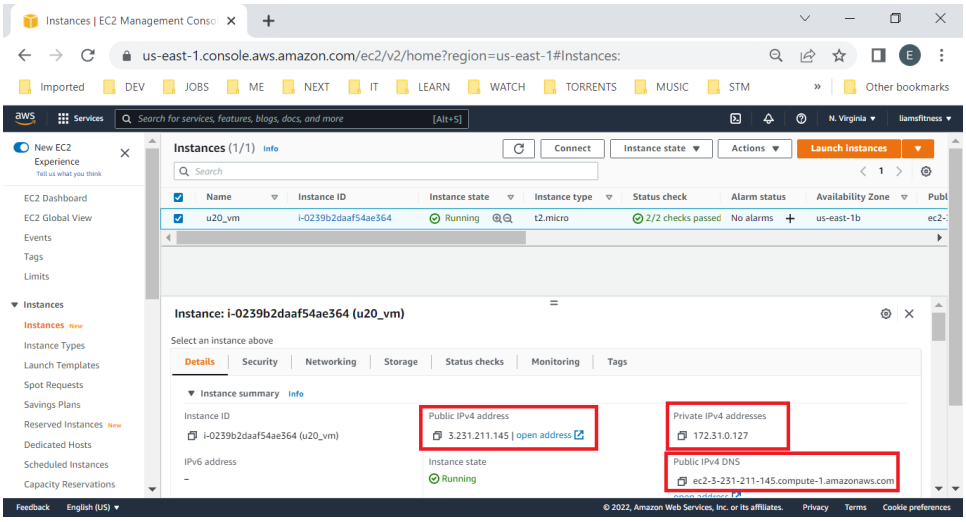
As you can see, it can take up to a minute, or two, for the instance to be brought up.



Now, you can click on "**Services**" at the top left of the screen and then select "**EC2**" under "**Recently visited**". Or, you enter "**EC2**" in the search bar to access your running instance (both methods work).

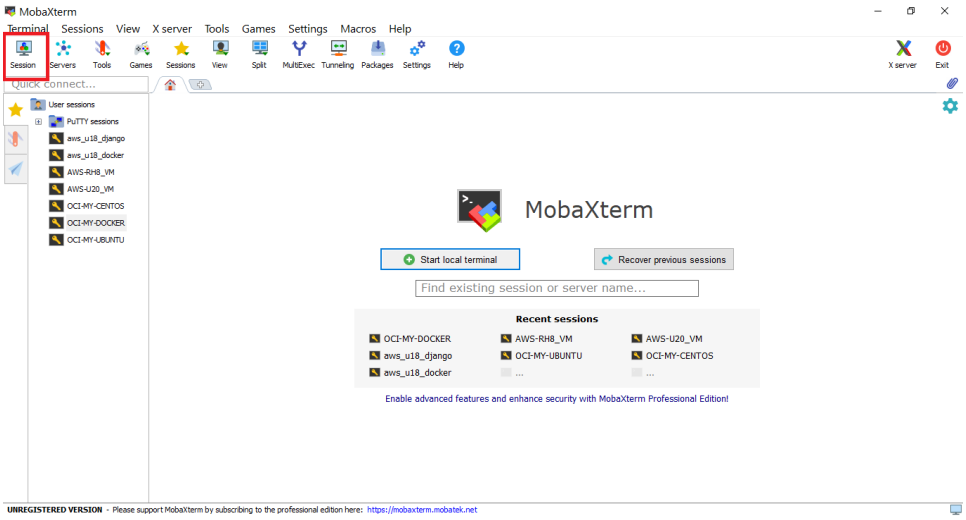


Ensure your new instance is selected (**u20_vm**), then on the "**Details**" tab, note the value for "**Public IPv4 DNS**". I usually keep the instance's name, public IP, private IP and public IPv4 DNS stored for easy access. We will need this to connect to the instance.

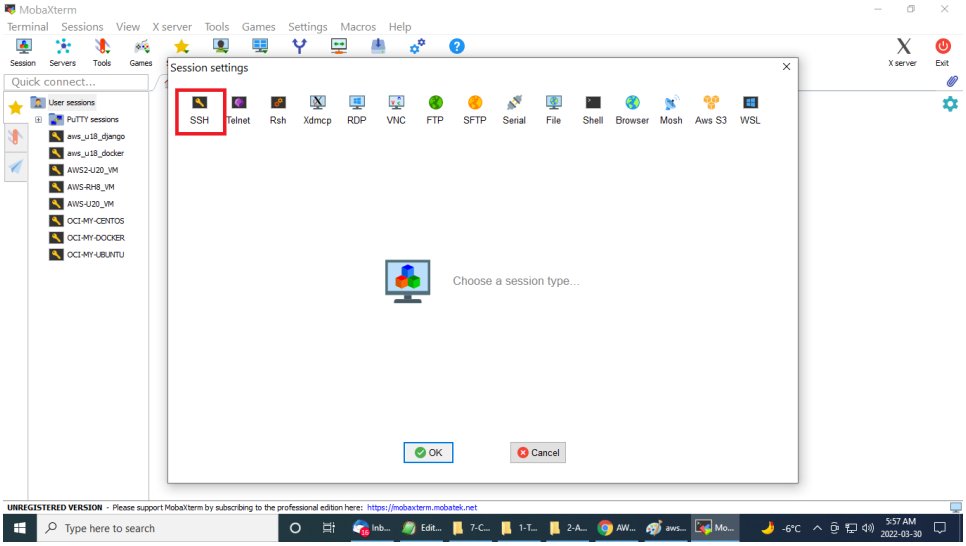


Connect to Ubuntu 20

Now you will need an SSH client to connect to your instance. I am on Windows 10 and have installed "GitBash" which includes an SSH client. If you do not want to install GitBash, I also use "MobaXterm Portable" and I find it to be a great tool and easy to use. Using **MobaXterm**, I would first create a session by clicking the "**Session**" button in the top left corner:

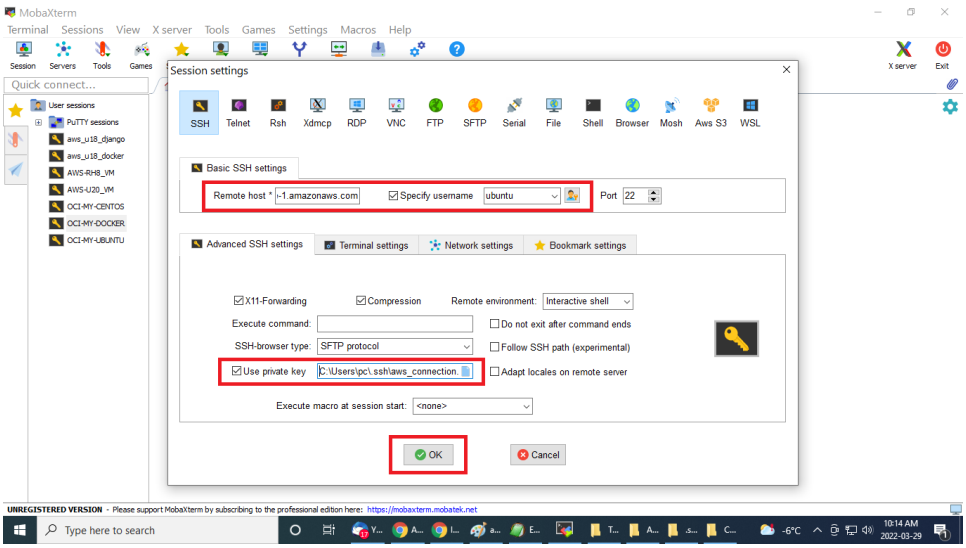


Followed by clicking the "**SSH**" button.

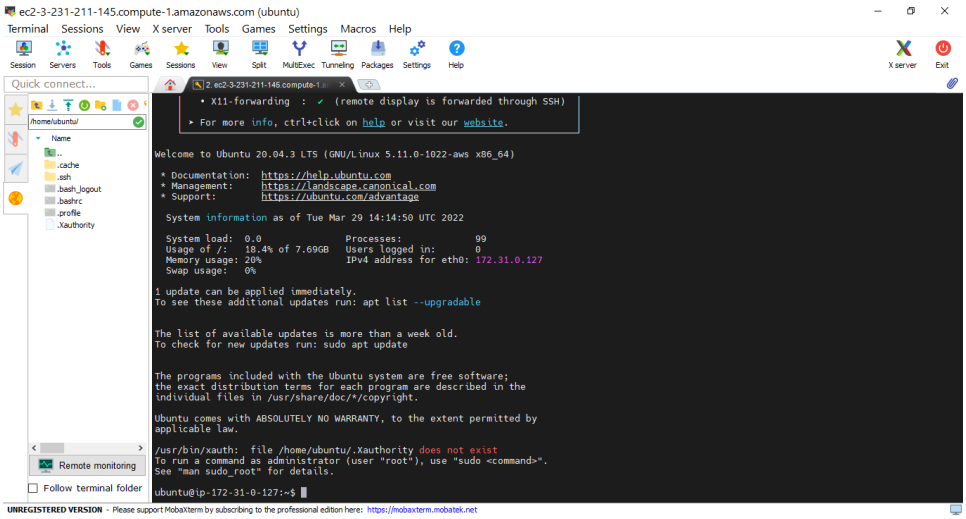


For "**Remote host:**", enter your "**Public IPv4 DNS**" from your instance's "**Details**" tab. Also specify the username of "**ubuntu**". That is the default username for every Ubuntu instance created on AWS.

Then, under "**Advanced SSH Settings**" select the key that you downloaded earlier (**aws_connection.pem**). Finally, click **OK**.



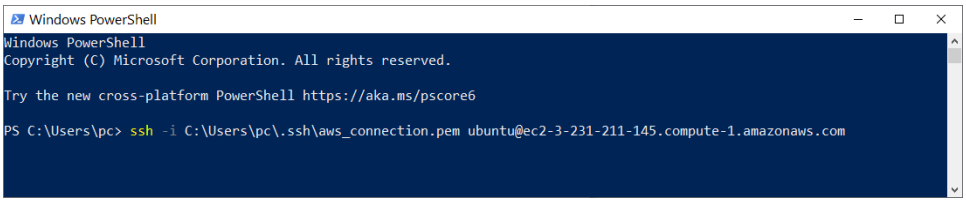
Your session should open to your newly created EC2 instance running Ubuntu 20.

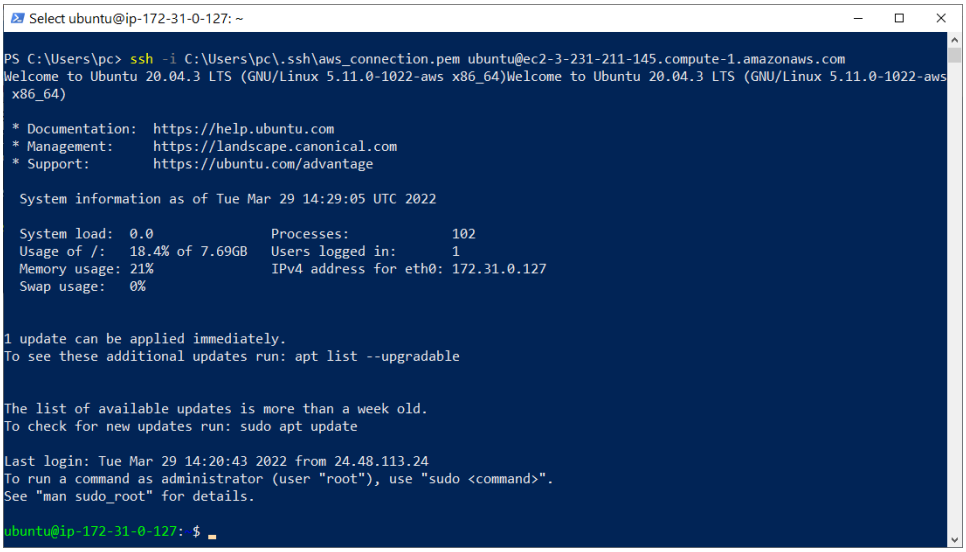


If you installed "GitBash", or have another SSH client, you can open either PowerShell or a Windows command prompt. Then, from the command line, enter the following (**NOTE**: make sure you enter your connection details):

ssh -i /path/my-key-pair.pem my-instance-user-name@my-instance-public-dns-name

ssh -i C:\Users\pc\.ssh\aws_connection.pem ubuntu@ec2-3-231-211-145.compute-1.amazonaws.com





I hope you've enjoyed this tutorial.

Please note that the free tier allows for 750 hours per month of Amazon EC2. You can create many EC2 instances but beware of the limit. If you go over that limit, you will pay the cost. My advice to you is to shutdown your instance/s after you've done your work.

If you want to become more proficient in a Linux environment you can see my other tutorials here.