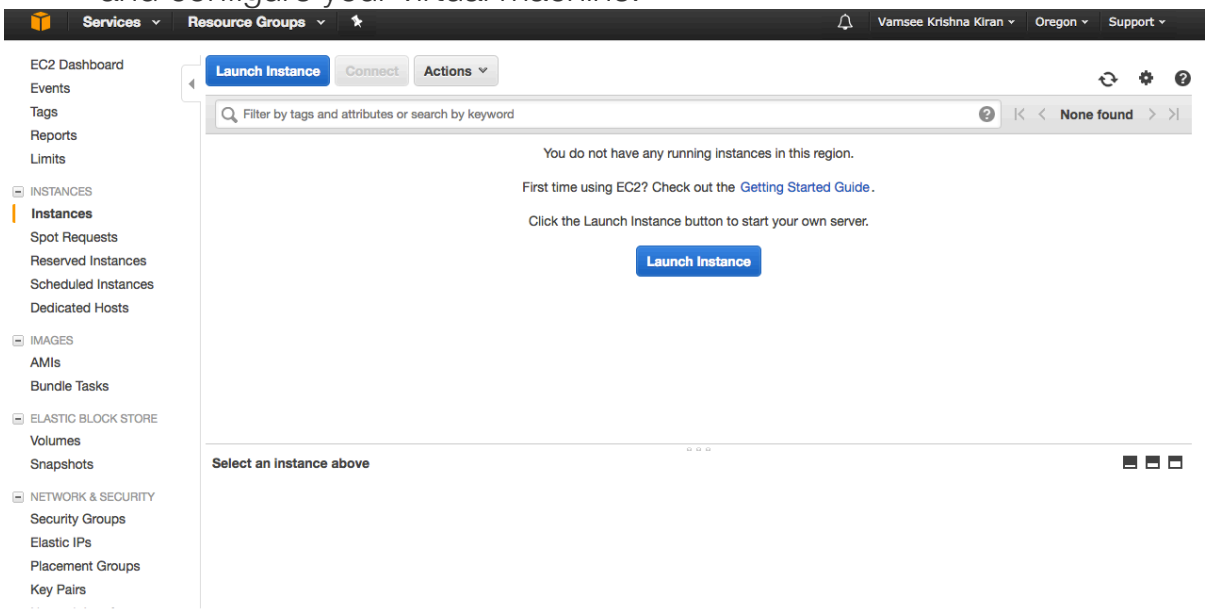
Launch a Linux VM

Step 1: Launch an Amazon EC2 Instance

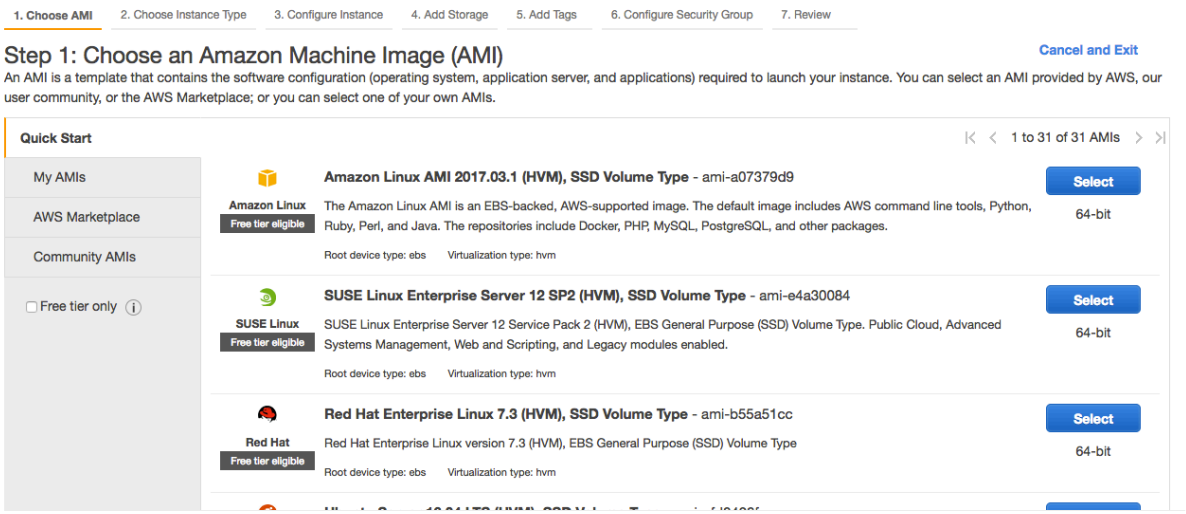
a. Open Amazon EC2 console and then click **Launch Instance** to create and configure your virtual machine.



Step 2: Configure your Instance

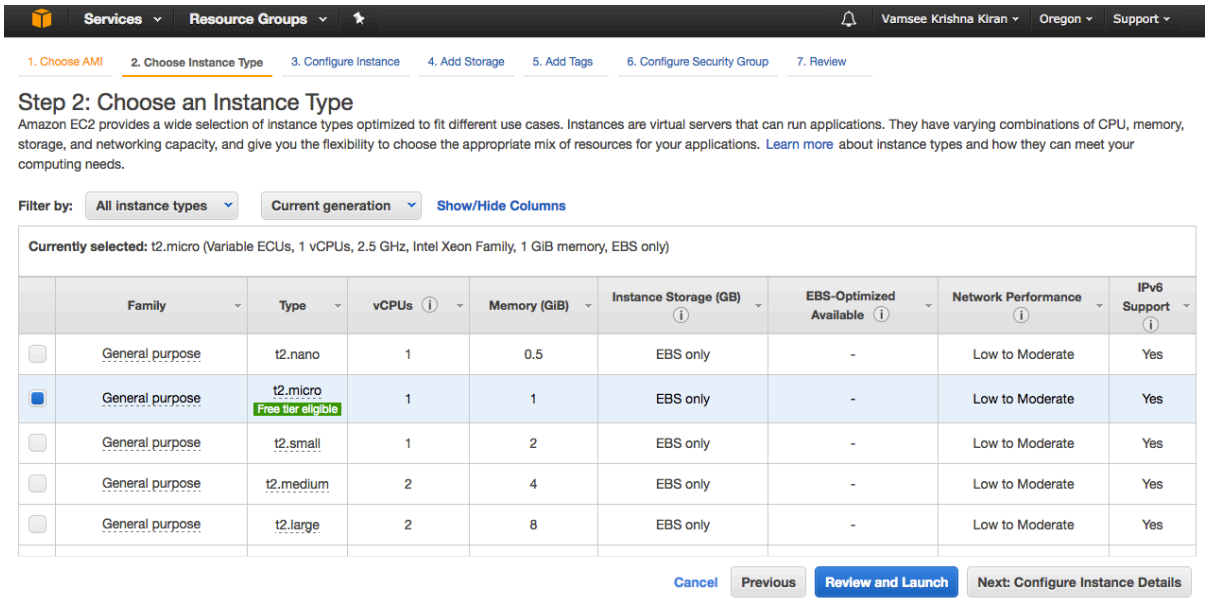
a. With Amazon EC2, you can specify the software and specifications of the instance you want to use. In this screen, you are shown options to choose an Amazon Machine Image (AMI), which is a template that contains the software configuration (e.g. an operating system, an application server, and applications). From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud.

For this tutorial, find *Amazon Linux AMI* and click **Select**.



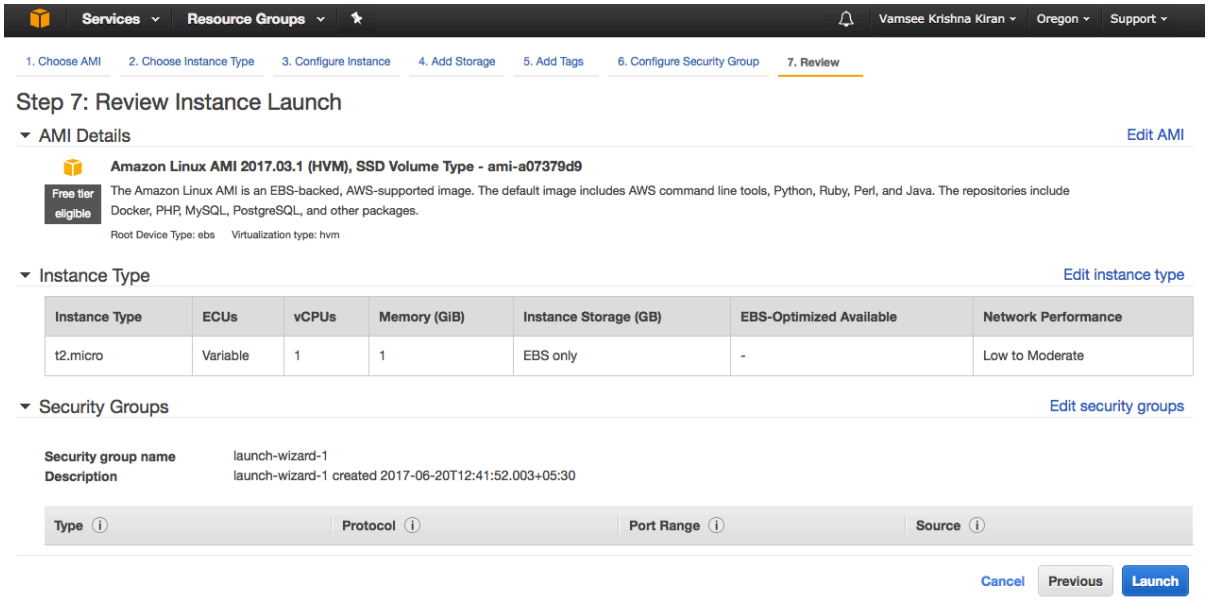
b. You will now choose an instance type. Instance types comprise of varying combinations of CPU, memory, storage, and networking capacity so you can choose the appropriate mix for your applications. For more information, see Amazon EC2 Instance Types.

The default option of *t2.micro* should already be checked. This instance type is covered within the Free Tier and offers enough compute capacity to tackle simple workloads. Click **Review and Launch** at the bottom of the page.



c. You can review the configuration, storage, tagging, and security settings that have been selected for your instance. While you have the option to customize these settings, we recommend accepting the default values for this tutorial.

Click **Launch** at the bottom of the page.



d. On the next screen you will be asked to choose an existing key pair or create a new key pair. A key pair is used to log into your instance (just like your house key is used to enter your home). Select **Create a new key pair** and give it the name **MyKeyPair**. Next click the **Download Key Pair** button.

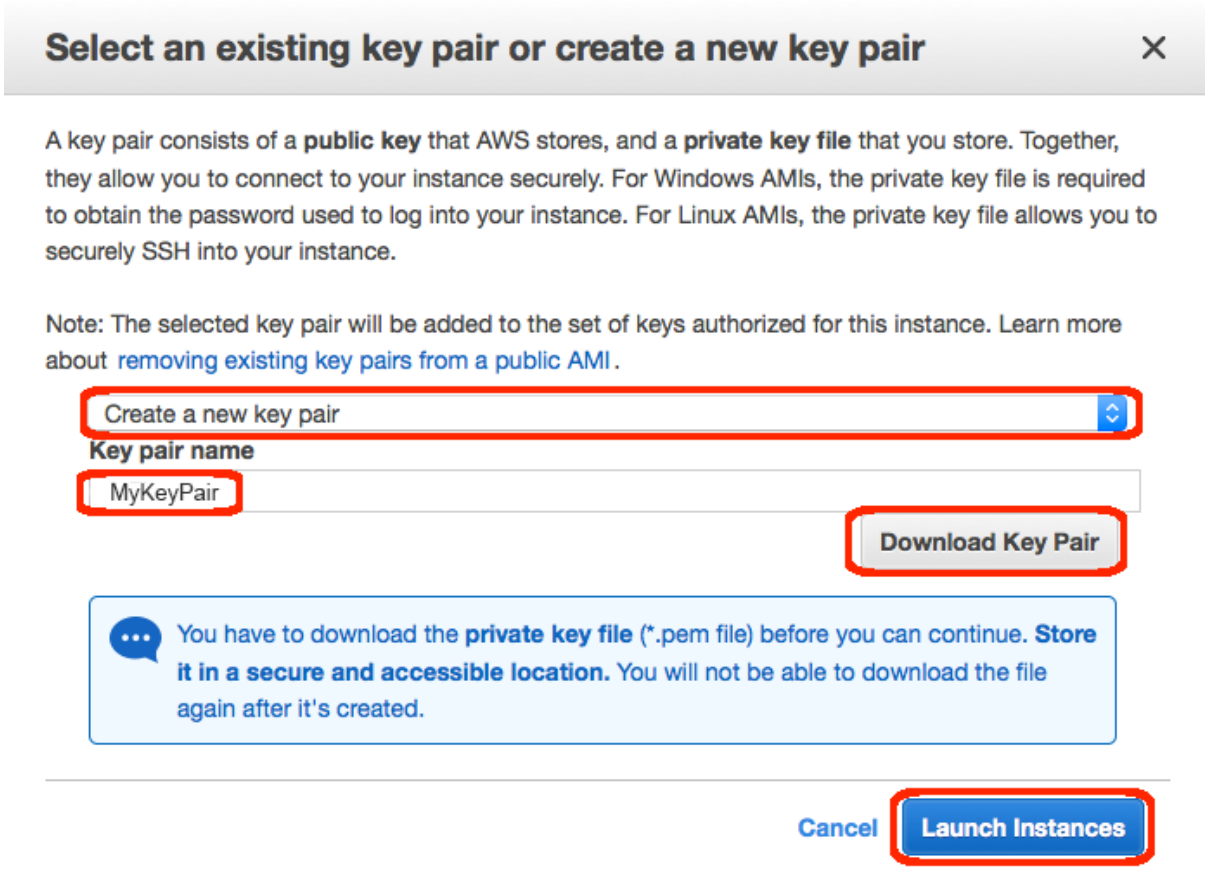
Be sure to save the key pair in a safe location on your computer. If you don't remember where you store your SSH private key (the file you are downloading), you won't be able to connect to your virtual machine.

Windows users: We recommend saving your key pair in your user directory in a sub-directory called .ssh
(ex. *C:\user\{yourusername}\.ssh\MyKeyPair.pem*).

Note: You can't use Windows Explorer to create a folder with a name that begins with a period unless you also end the folder name with a period. After you enter the name (.ssh.), the final period is removed automatically.

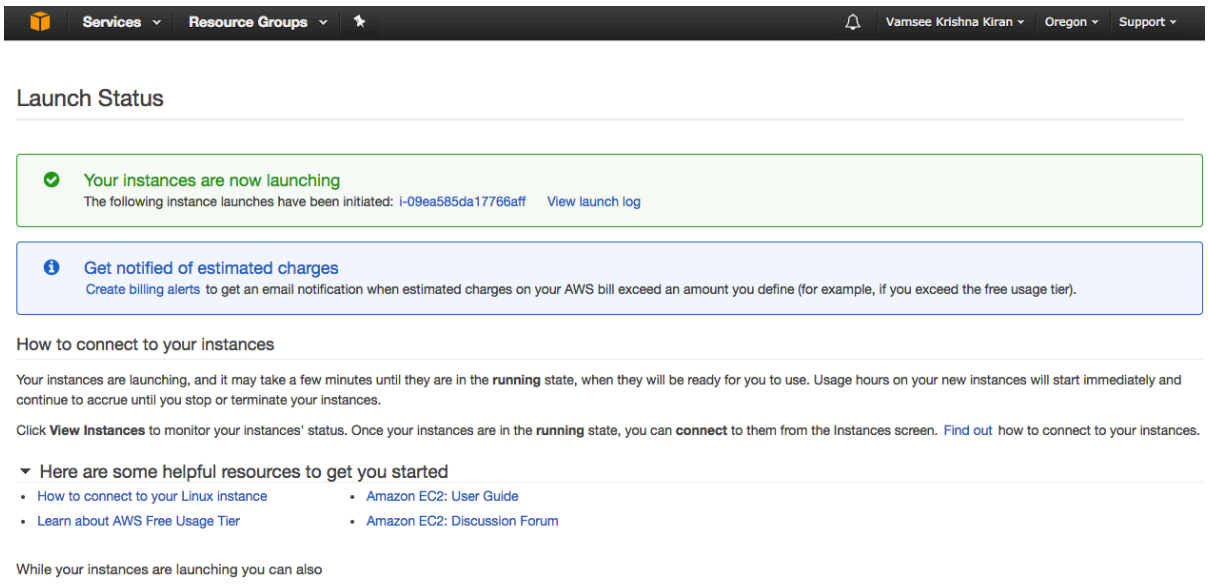
Mac/Linux users: We recommend saving your key pair in the .ssh sub-directory from your home directory (ex. *~/.ssh/MyKeyPair.pem*).

Note: On Mac, the key pair is downloaded to your Downloads directory by default. To move the key pair into the .ssh sub-directory, enter the following command in a terminal window: **mv ~/Downloads/MyKeyPair.pem ~/.ssh/MyKeyPair.pem**

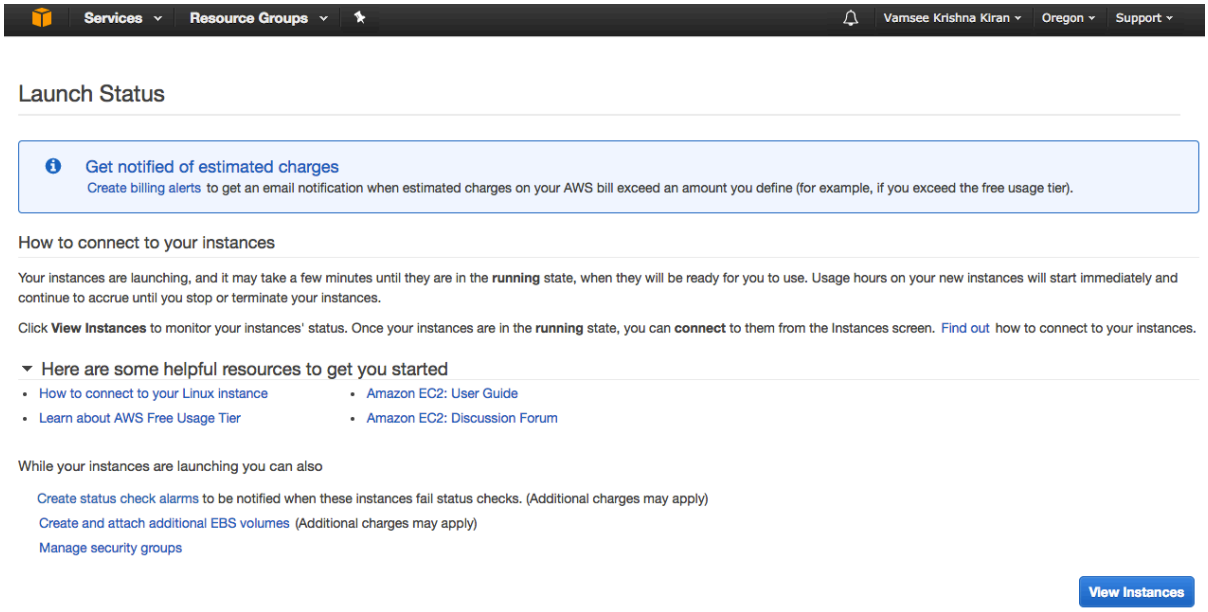


After you have stored your key pair, click **Launch Instance** to start your Linux instance.

Note: It will take a few minutes to launch your instance.

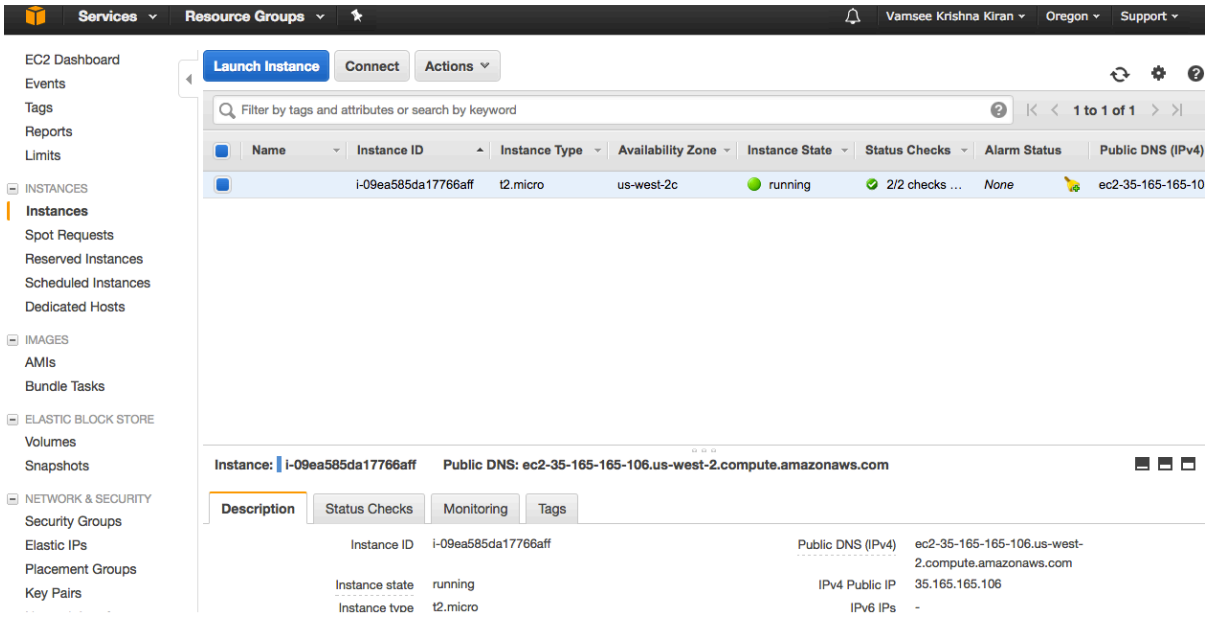


e. Click **View Instances** on the next screen to view your instances and see the status of the instance you have just started.



f. Make note of the *Public IP* address of your AWS instance, you will need this to connect to the instance in Step 3 part c.

Note: If your instance is still starting up, the *Public IP* address may not be shown yet. The *Instance State* column will show you if the instance is running yet, and the *Status Checks* column will tell you if the instance has passed the 2 checks to make sure it is done provisioning. You can refresh these values by pressing the refresh button on the right just above the table.



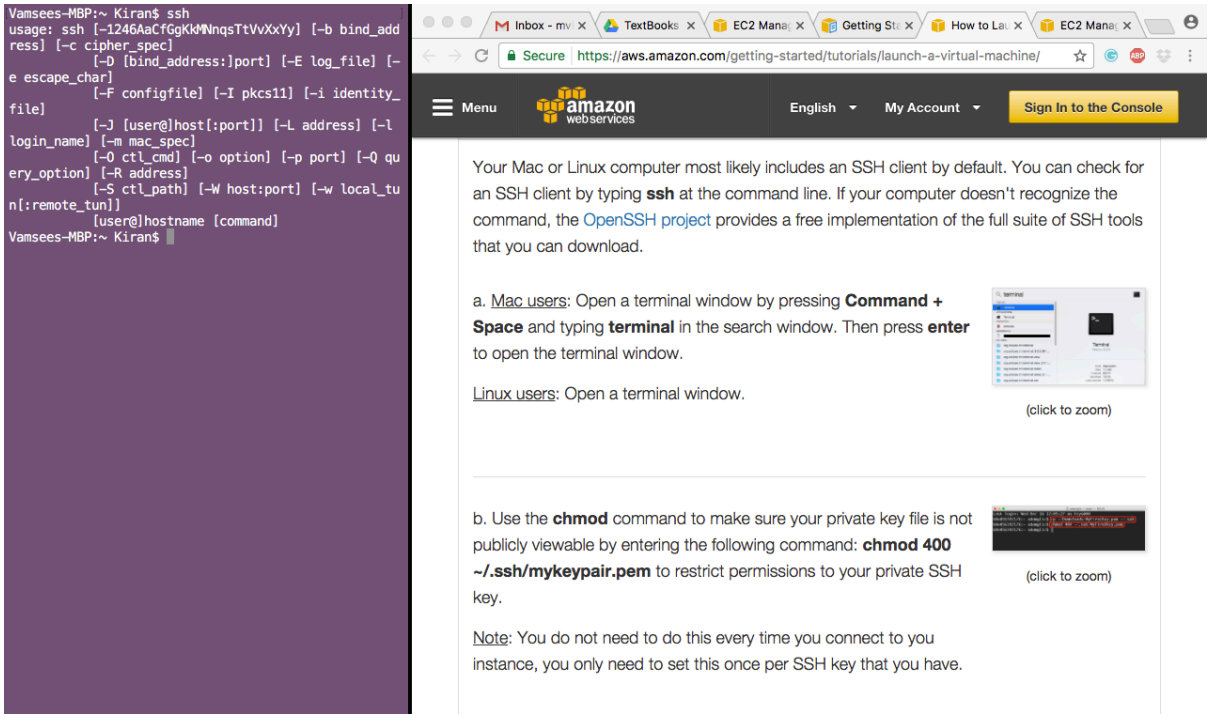
Step 3: Connect to your Instance

Your Mac or Linux computer most likely includes an SSH client by default. You can check for an SSH client by typing **ssh** at the command line. If your

computer doesn't recognize the command, the OpenSSH project provides a free implementation of the full suite of SSH tools that you can download.

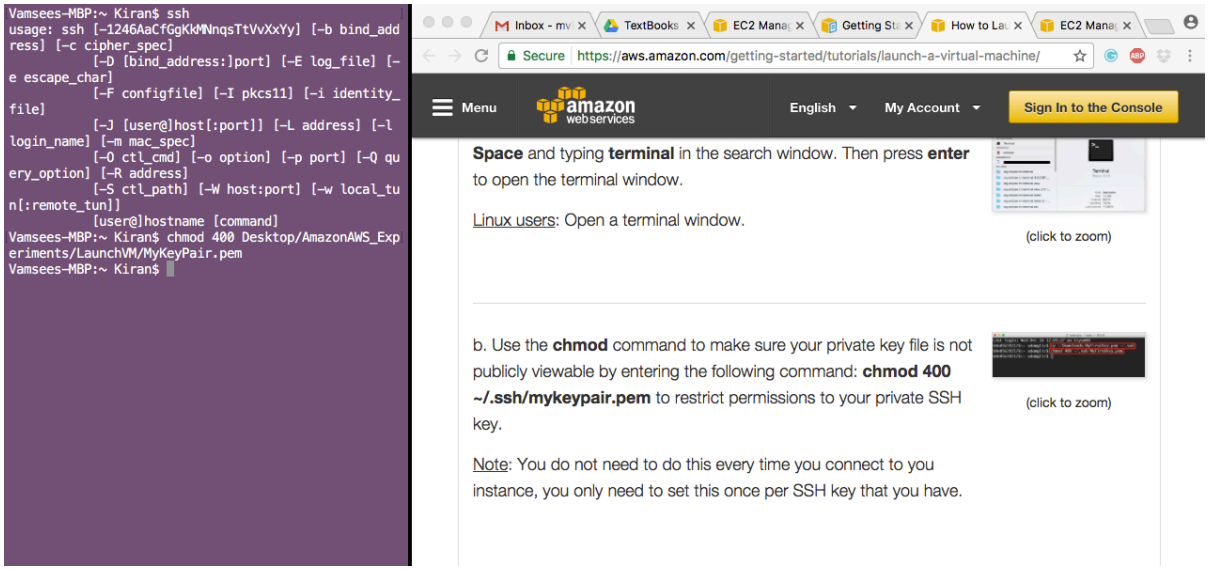
a. Mac users: Open a terminal window by pressing **Command + Space** and typing **terminal** in the search window. Then press **enter** to open the terminal window.

Linux users: Open a terminal window.



b. Use the **chmod** command to make sure your private key file is not publicly viewable by entering the following command: **chmod 400 ~/.ssh/mykeypair.pem** to restrict permissions to your private SSH key.

Note: You do not need to do this every time you connect to you instance, you only need to set this once per SSH key that you have.



c. Use SSH to connect to your instance. In this case the user name is ec2-user, the SSH key is stored in the directory we saved it to in step 2 part d, and the IP address is from step 2 part f. The format is *ssh -i {full path of your .pem file} ec2-user@{instance IP address}*.

Windows users: Enter **ssh -i 'c:\Users*yourusername*\.ssh\MyKeyPair.pem' ec2-user@*{IP_Address}*** (ex. *ssh -i 'c:\Users\adamglic\.ssh\MyKeyPair.pem' ec2-user@52.27.212.125*)

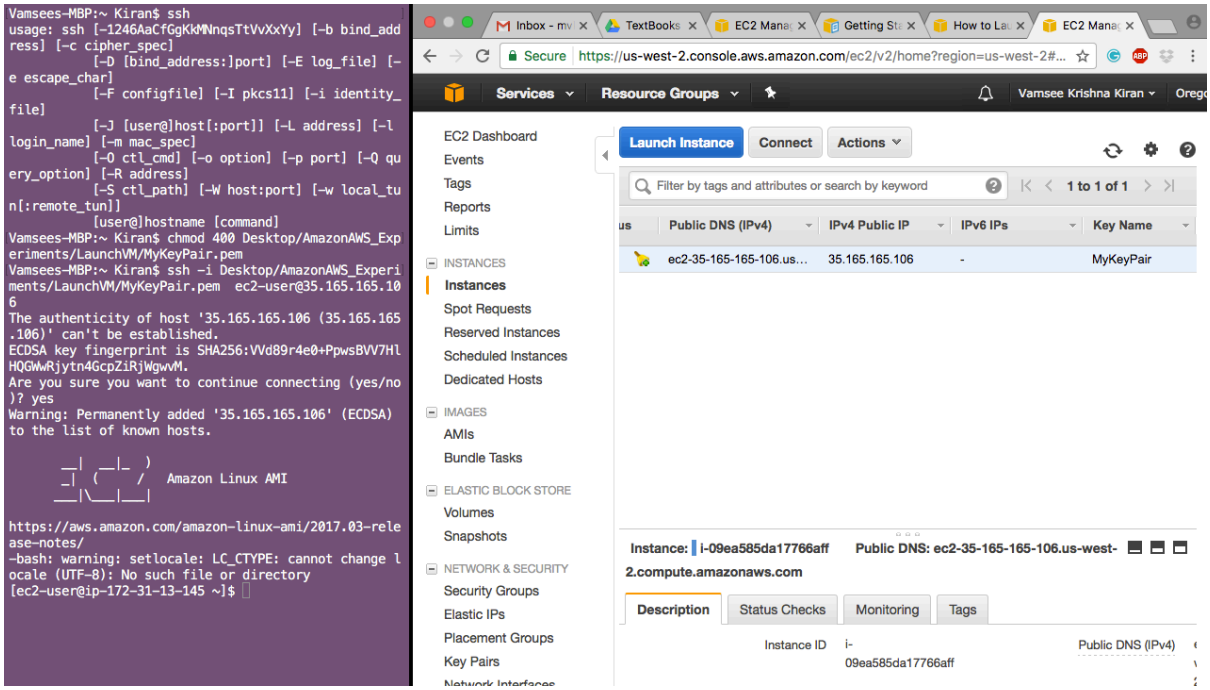
Mac/Linux users: Enter **ssh -i ~/.ssh/MyKeyPair.pem ec2-user@*{IP_Address}*** (ex. ssh -i ~/.ssh/MyKeyPair.pem ec2-user@52.27.212.125)

> Note: if you started a Linux instance that isn't Amazon Linux, there may by a different user name that is used. common user names include *ec2-user*, *root*, *ubuntu*, and *fedora*. If you are unsure what the login user name is, check with your AMI provider.

You'll see a response similar to the following:

The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' can't be established. RSA key fingerprint is 1f:51:ae:28:df:63:e9:d8:cf:38:5d:87:2d:7b:b8:ca:9f:f5:b1:6f. Are you sure you want to continue connecting (yes/no)?

Type **yes** and press **enter**.

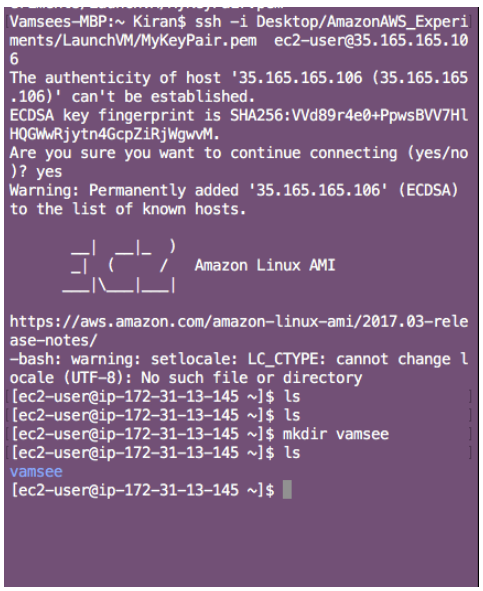


You'll see a response similar to the following:

Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA) to the list of known hosts.

You should then see the welcome screen for your instance and you are now connected to your AWS Linux virtual machine in the cloud. The welcome screen is nothing but the command prompt of the Linux machine that you just launched.

Now you can access the terminal of the virtual machine you launched. For example, you can create some directories etc. on the VM as follows:



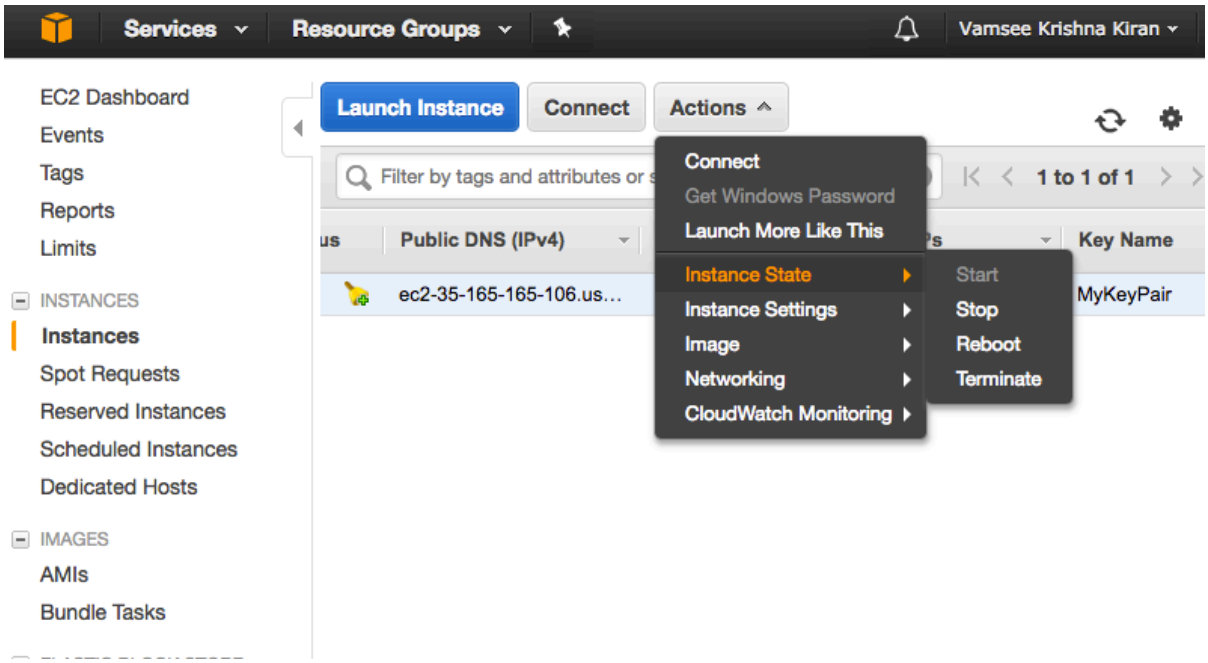
To come out of the VM terminal, type exit. This will exit from terminal of VM by closing the connection to VM. You will come back to your host machine terminal now.

```
https://aws.amazon.com/amazon-linux-ami/2017.03-rele
ase-notes/
-bash: warning: setlocale: LC_CTYPE: cannot change l
ocale (UTF-8): No such file or directory
[ec2-user@ip-172-31-13-145 ~]$ ls
[ec2-user@ip-172-31-13-145 ~]$ ls
[ec2-user@ip-172-31-13-145 ~]$ mkdir vamsee
[ec2-user@ip-172-31-13-145 ~]$ ls
vamsee
[ec2-user@ip-172-31-13-145 ~]$ exit
logout
Connection to 35.165.165.106 closed.
Vamsees-MBP:~ Kiran$
```

Step 4: Terminate Your Instance

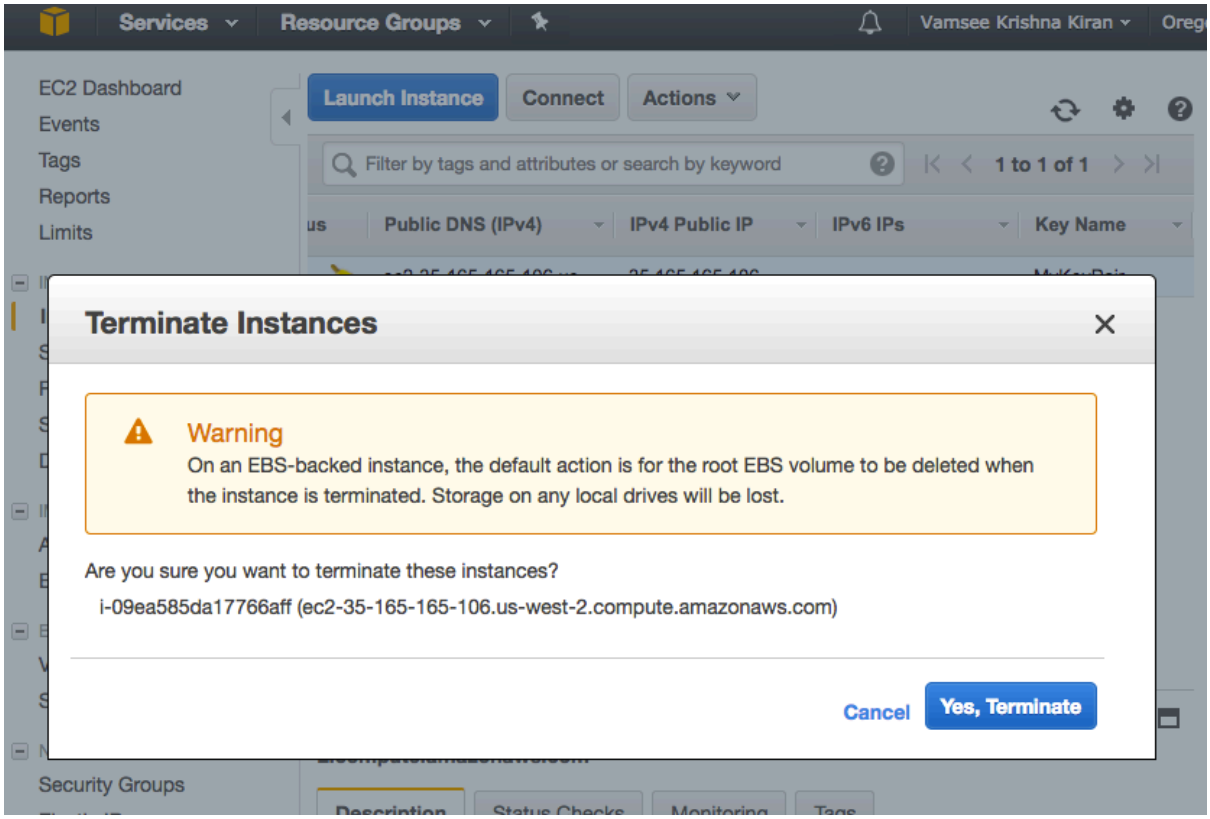
You can easily terminate the instance from the EC2 console. In fact, it is a best practice to terminate instances you are no longer using so you don't keep getting charged for them.

a. Back on the EC2 Console, select the box next to the instance you created. Then click the **Actions** button, navigate to *Instance State,* and click **Terminate**.



b. You will be asked to confirm your termination - select **Yes, Terminate.**

Note: This process can take several seconds to complete. Once your instance has been terminated, the Instance State will change to *terminated* on your EC2 Console.


Services
Resource Groups
Vamsee Krishna Kiran
EC2 Dashboard
Events
Tags
Reports
Limits
Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 1 of 1
Public DNS (IPv4)
IPv4 Public IP
IPv6 IPs
Key Name
Terminate Instances
Warning
On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.
Are you sure you want to terminate these instances?
i-09ea585da17766aff (ec2-35-165-165-106.us-west-2.compute.amazonaws.com)
Cancel
Yes, Terminate
Security Groups