**What is AWS EC2**

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the [Amazon Web Services](https://howtodoinjava.com/aws/aws-amazon-web-service/) (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster.

Amazon EC2 allows us to get compute power in cloud through the launching of *virtual servers called instances*. When you launch an instance, you can make use of the compute as you wish, just as you would with an on-premises server. Because you are paying for the computing power of the instance, you are charged per hour while the instance is running. When you stop the instance, you are no longer charged.

There are two main things when launching instances on AWS :

* the amount of virtual hardware dedicated to the instance and
* the software loaded on the instance.

These two dimensions of new instances are controlled, respectively, by the Instance Type and the AMI - Amazon Machine Image.

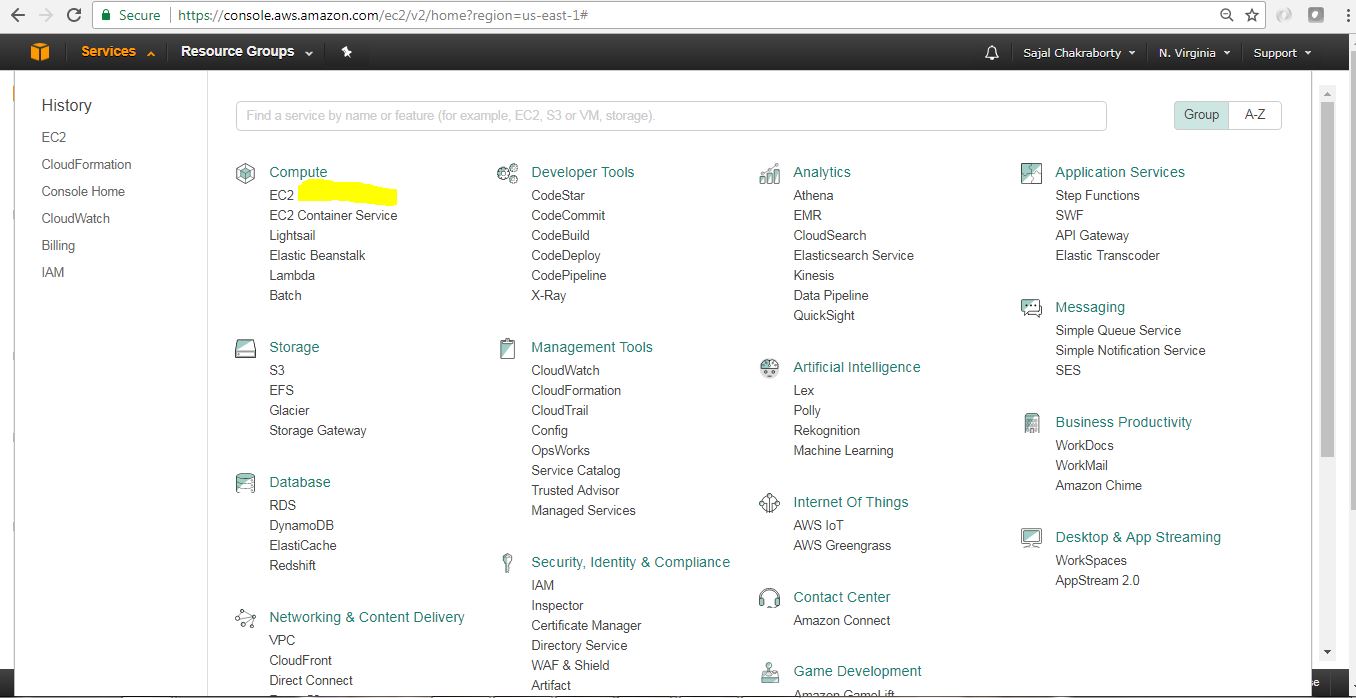
**Prerequisites**

Before starting up this exercise we need below setup as prerequisites.

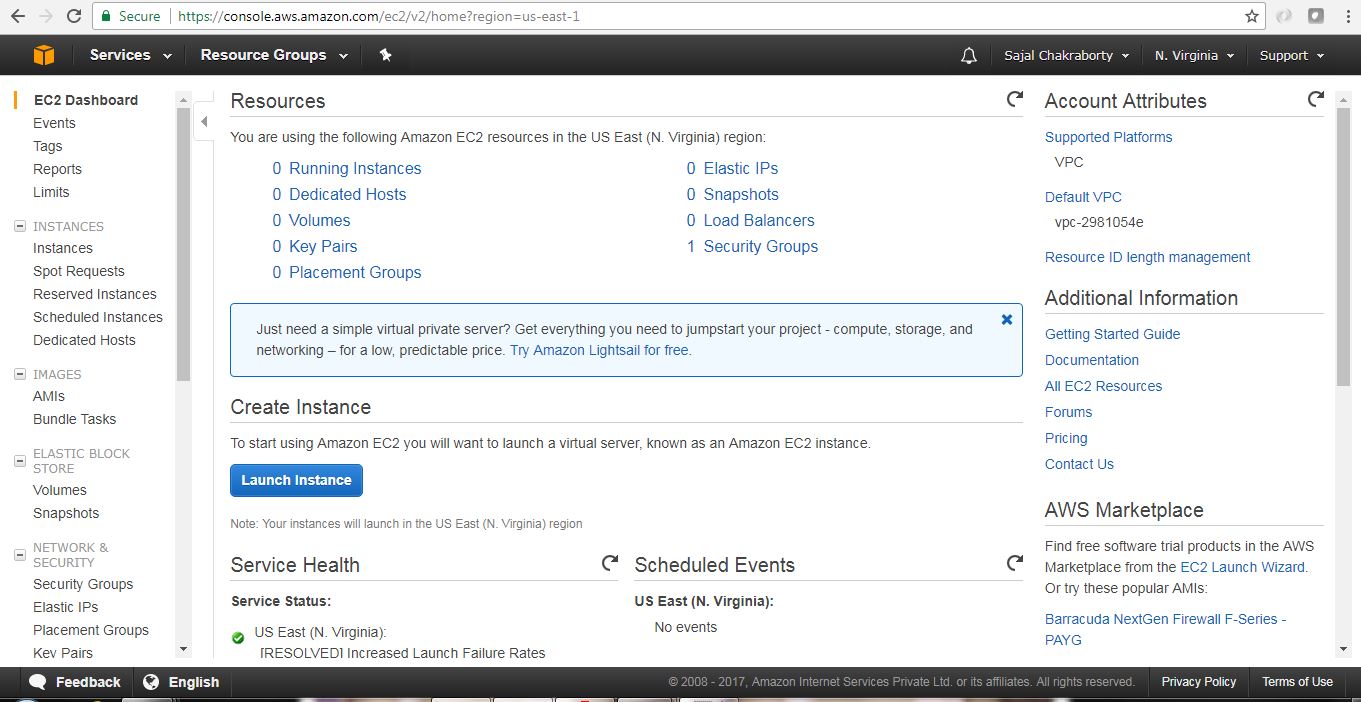
* **AWS account** – Must have to open one AWS account to do this exercise.
* **SSL Client – PuTTY** – This should be installed in Local M/C including PuTTYgen, PUTTY, Pageantapplications. Full set of applications related to PutTTY can be downloaded from [this link](https://the.earth.li/~sgtatham/putty/latest/w64/putty.zip). Just unzip to a convenient location.

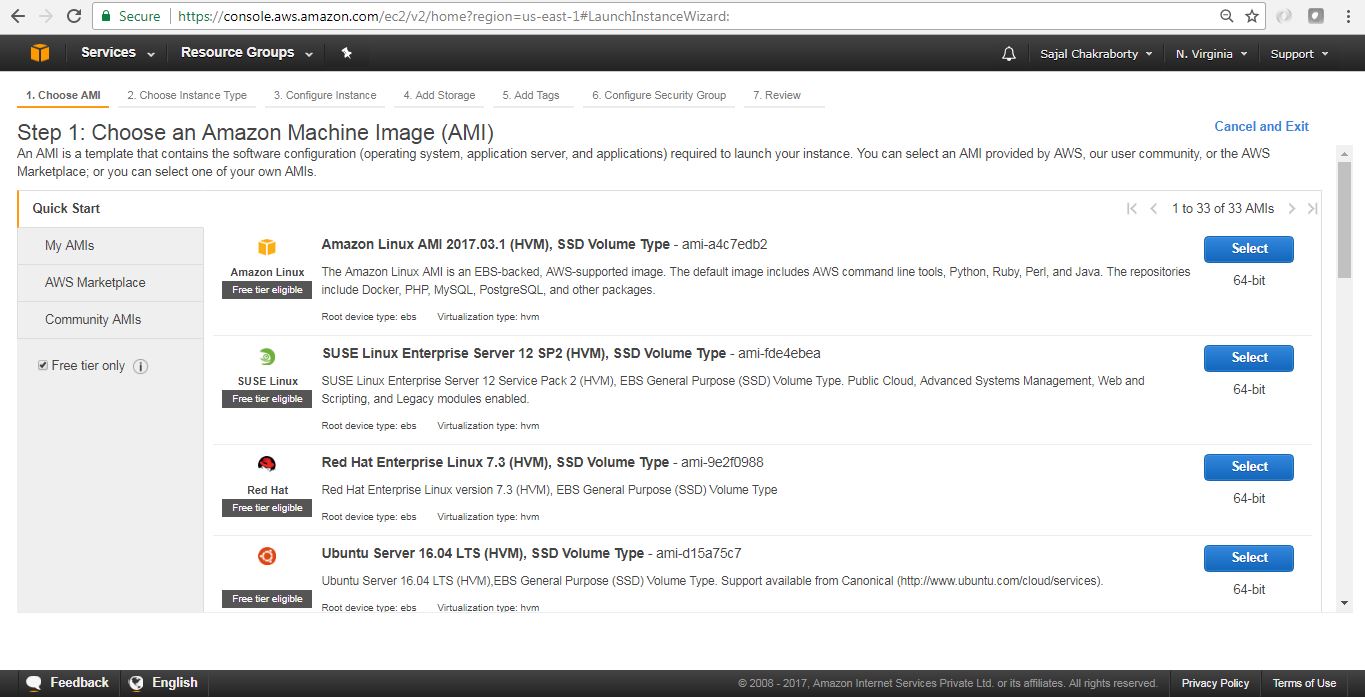
So, let’s get started by creating an AWS EC2 instance.

**Steps to Create EC2 Instance**

1. Login to [AWS console](https://console.aws.amazon.com/console) and open EC2 home screen – Once log in and click Services menu in the top left corner of the home screen, we will need to click the EC2 Link under compute section. The EC2 landing page will look like :[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/EC2_0_landingScreen.jpg)Figure 1.0 : Services Menu with Ec2 Link

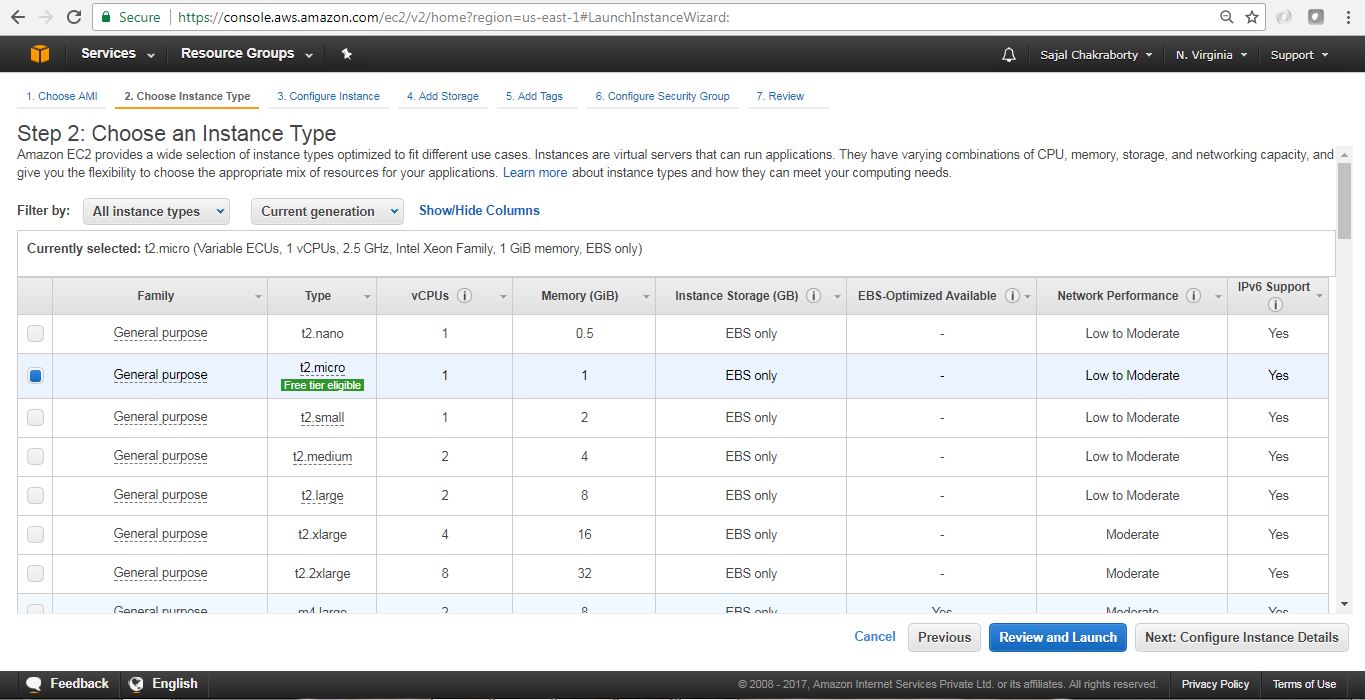
This is how EC2 Home page looks like. This is the place we land when we click EC2 link in the previous step (Figure 1.0).

[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_1.jpg)Figure 1.1 : EC2 Home Screen

1. **Launch a new virtual server** Now we need to click the Launch Instance button from the EC2 Home page.This will start the instance creation wizard which will guide us through the following steps mainly:
   * Selecting an OS
   * Choosing the size of your virtual server
   * Choosing the Extra attached storage we need for this instance
   * Configuring details
   * Reviewing your input and selecting a key pair for SSH
2. **Select AMI** – AMI stands for **Amazon Machine Images**, which is kind of a blue print of the instance that we will create, it tells about the Operating System of the Instance as well as the basic softwares that will be pre-installed. So in this step, while choosing AMI, we are choosing the underlying Operating System(OS) and the preinstalled software bundles that would be available in the instance upfront.We need to choose AMI based on our need. We will choose only those AMIs which falls under Free Tire Eligible type. An AMI is the basis your virtual server starts from. AMIs are offered by AWS, by thirdparty providers, and by the community. We can also create custom AMI based on our need. Some AMIs are chargeable. Here for this exercise we will choose Ubuntu Server 16.04 LTS(HVM) AMI.[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_2.jpg)Figure 3.0 : Select AMI

Click on the Select button beside the AMI to proceed with the sect step.

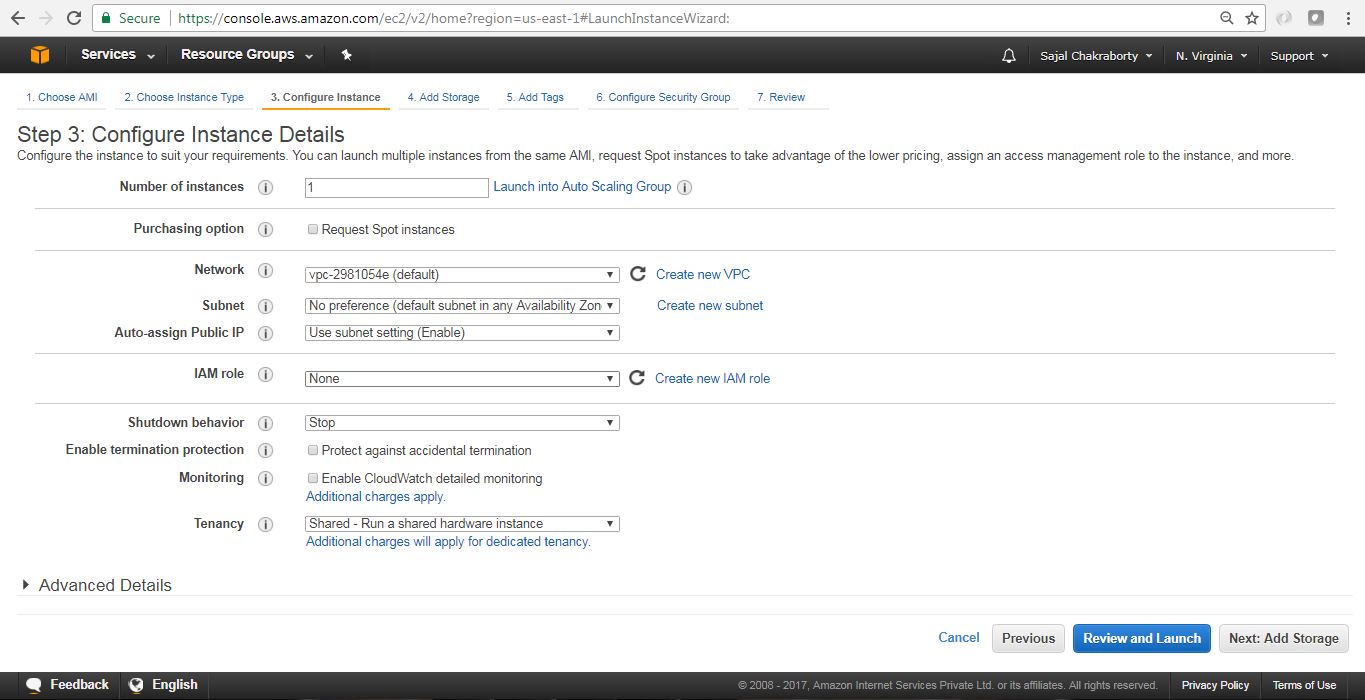
1. **Select Instance Type** – It’s now time to choose the computing power needed for your virtual server. Instance Type indicates the computing power of the instance by allocating the amount of virtual hardware dedicated to the instance. On AWS, computing power is classified into instance types. An instance type primarily describes the number of vCPUs and the amount of memory an instance will be allocated to once created. We will choose only the Free Tire Eligible Instance Type (t2.micro) which provides 1 vCPU and 1 GB of Memory. AWS has already defined many [instance types](https://aws.amazon.com/ec2/instance-types/), we will now choose the t2.micro instance type for this exercise. Once instance type is selected, click on button Next: Configure Instance Details in the bottom right corner of the page to proceed with the next steps.

[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_3.jpg)Figure 4.0 : Instance Type Selection

1. **Instance Details page** This will look like below. Here we will take all the default values and proceed with clicking Next: Add Storage button at the bottom right corner of the page. Before proceeding to the next section please note few important things that we can modify here:
   * Number of Instances we are going to create in this wizard.
   * We can select the Spot Instance option, Spot Instances are idle compute capacity that AWS makes available based on bid prices from customers.
   * Networking/Subnet/Public IP setting – This is the VPC under which the instance would be created, We can choose existing VPC, or create a new VPC here. VPC is itself a large topic which is out of scope of this article. Please follow official AWS [documentation](https://aws.amazon.com/documentation/vpc/)for the same.
   * Also we can configure, Shutdown Behavior, Termination Protection Flag(To avoid accidental Termination), Tenancy(Common vs dedicated H/W), Cloud Watch Detailed Monitoring. Here some of them are chargeable option.

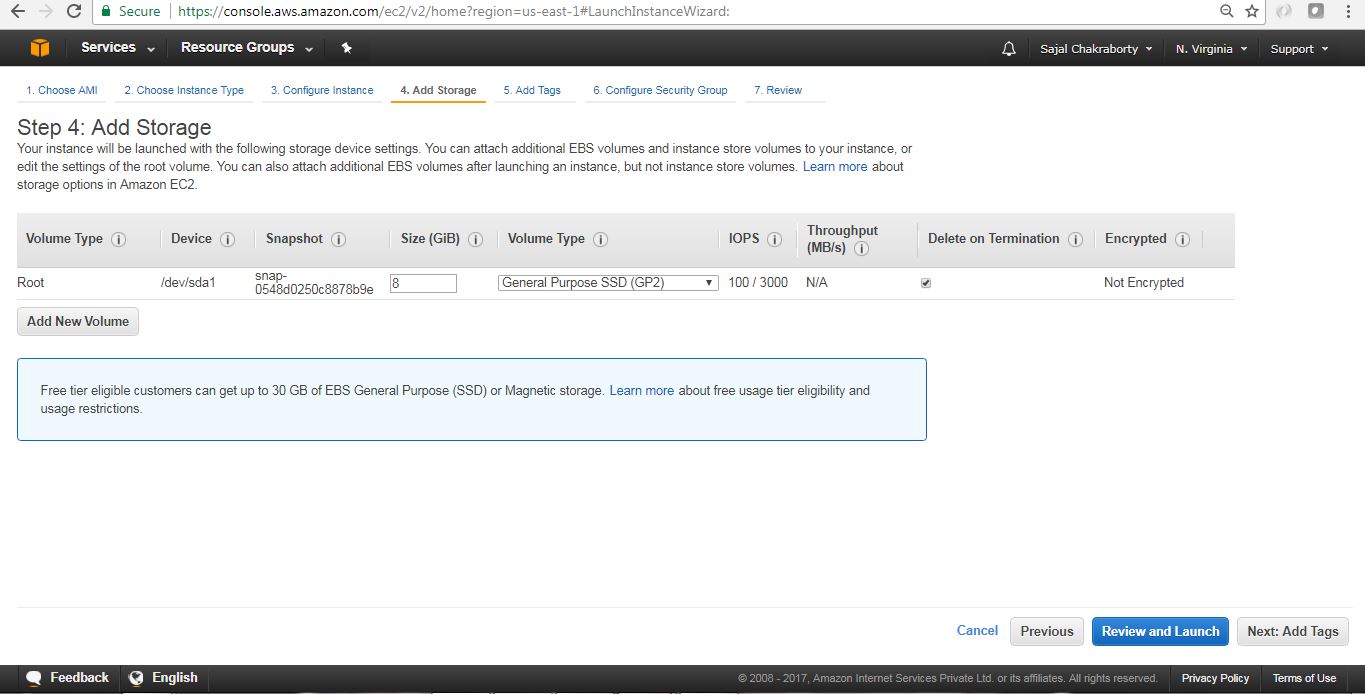
We can Change those later also once the Instance has been created.

Now we will proceed to add EBS (Elastic Block Store) volume details associated with this instance. Configure Instance Details screen will look like

[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_4.jpg)Figure 5.0 : Configure Instance Details

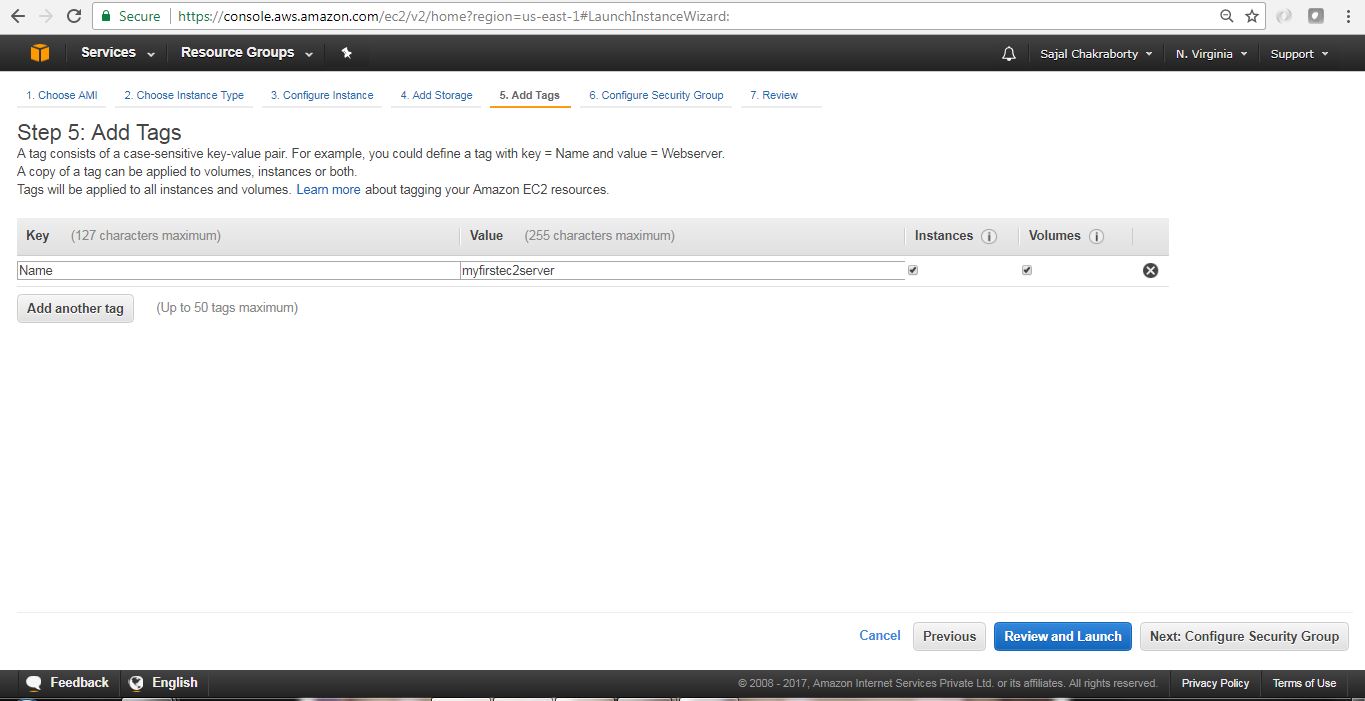
1. **Add EBS Storage** – EBS stands for Elastic block storage. It is basically network-attached storage attached your virtual server. This page will help us to configure Storage that will be associated with the EC2 instance that we are going to launch. We will choose default values to avoid extra cost. Make sure you choose the capacity which falls under Free Tire Eligibility to avoid incurring extra cost.

Review and click Next: Add Tags button at the bottom right corner of the page to proceed. The EBS volume configuration page will look like

[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_5.jpg)Figure 6.0 : EBS volume configuration

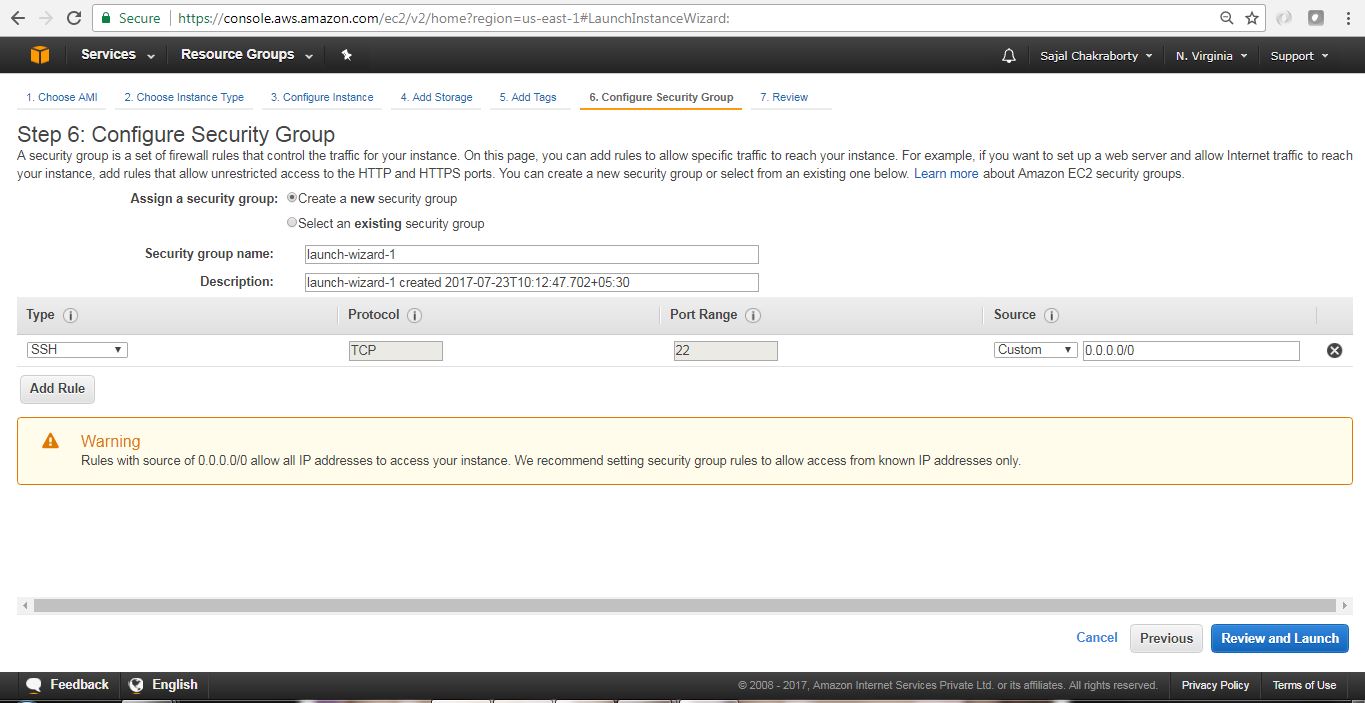
1. **Add Tags** – Tags are kind of an identifier of any AWS resources, with Tags we can easily locate the resource in future and also it helps us to classify the resources. Tags help you to organize resources on AWS. We can add max 50 tags to a particular resources. It is a simple Key value pair associated with the resource.

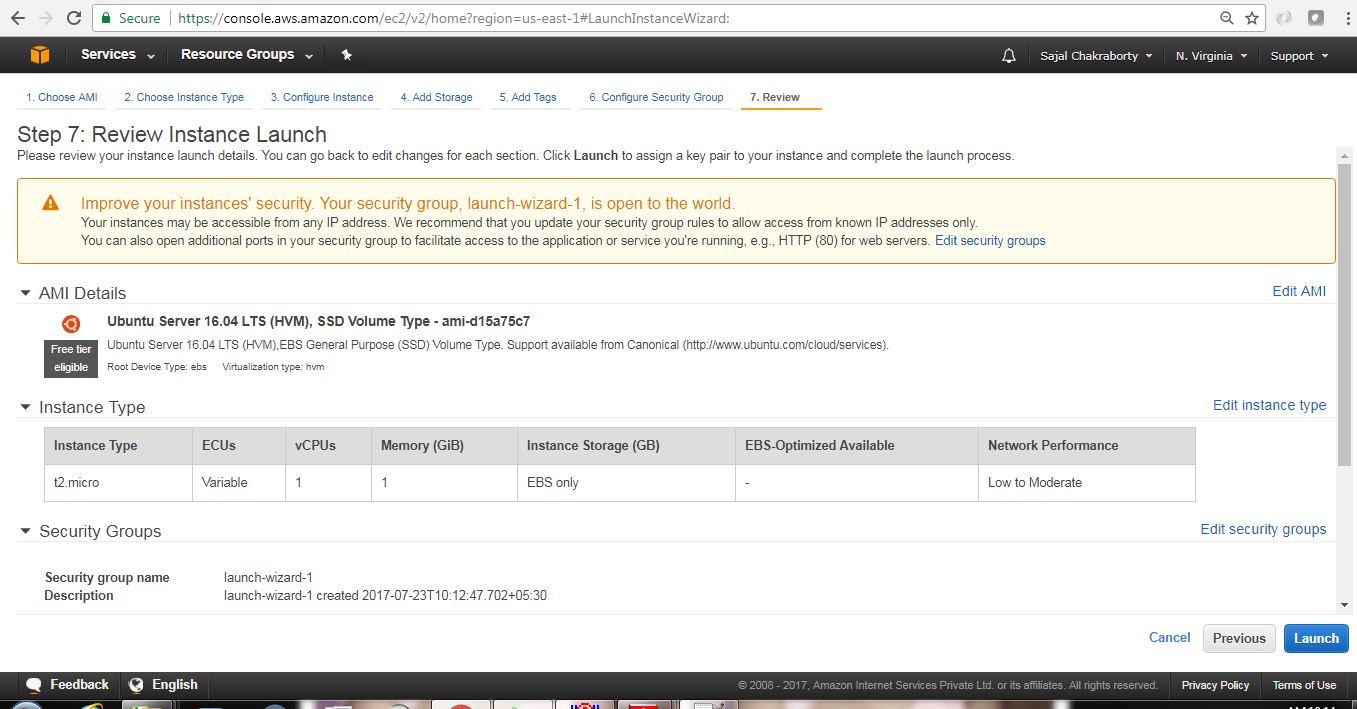
In this step we will add one tag Name with value myfirstec2server. Once tag is created, we will proceed with configuring security group by clicking the “Next: Configure Security Group” button at the bottom right corner of the page. Add Tag screen will look like:

[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_6.jpg)Figure 7.0 : EC2 Add Tag

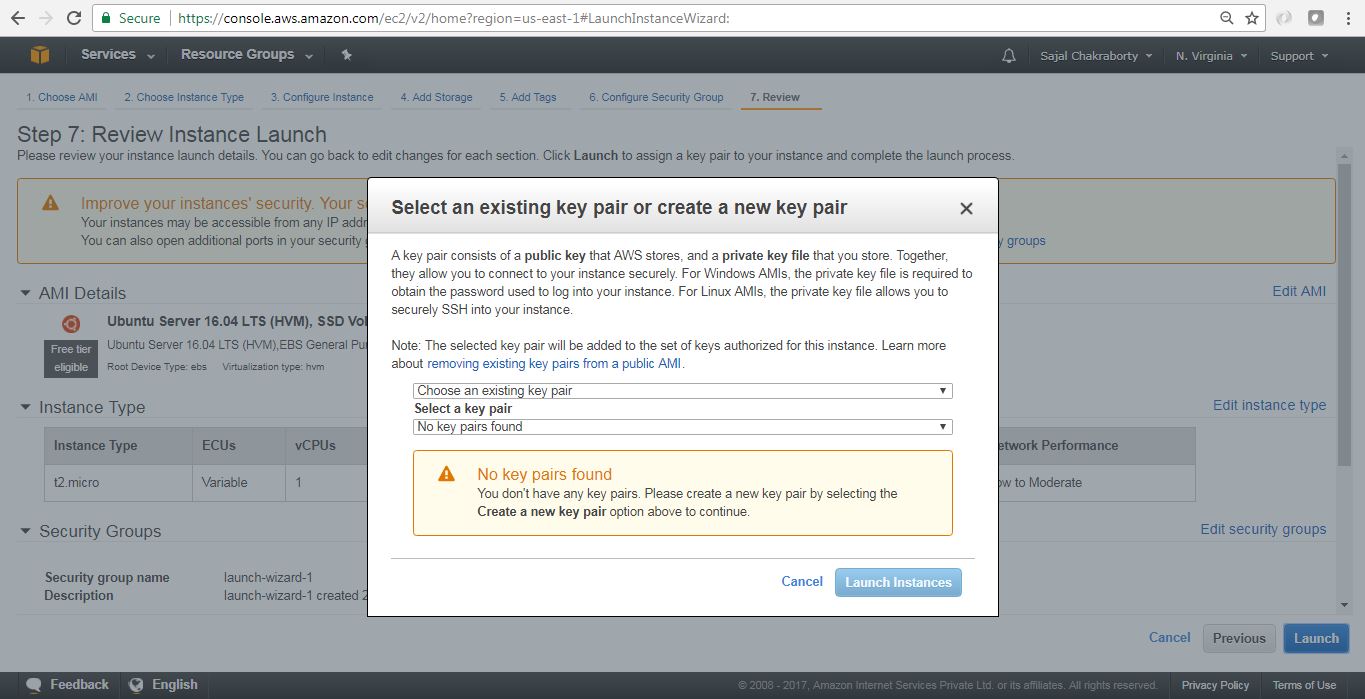
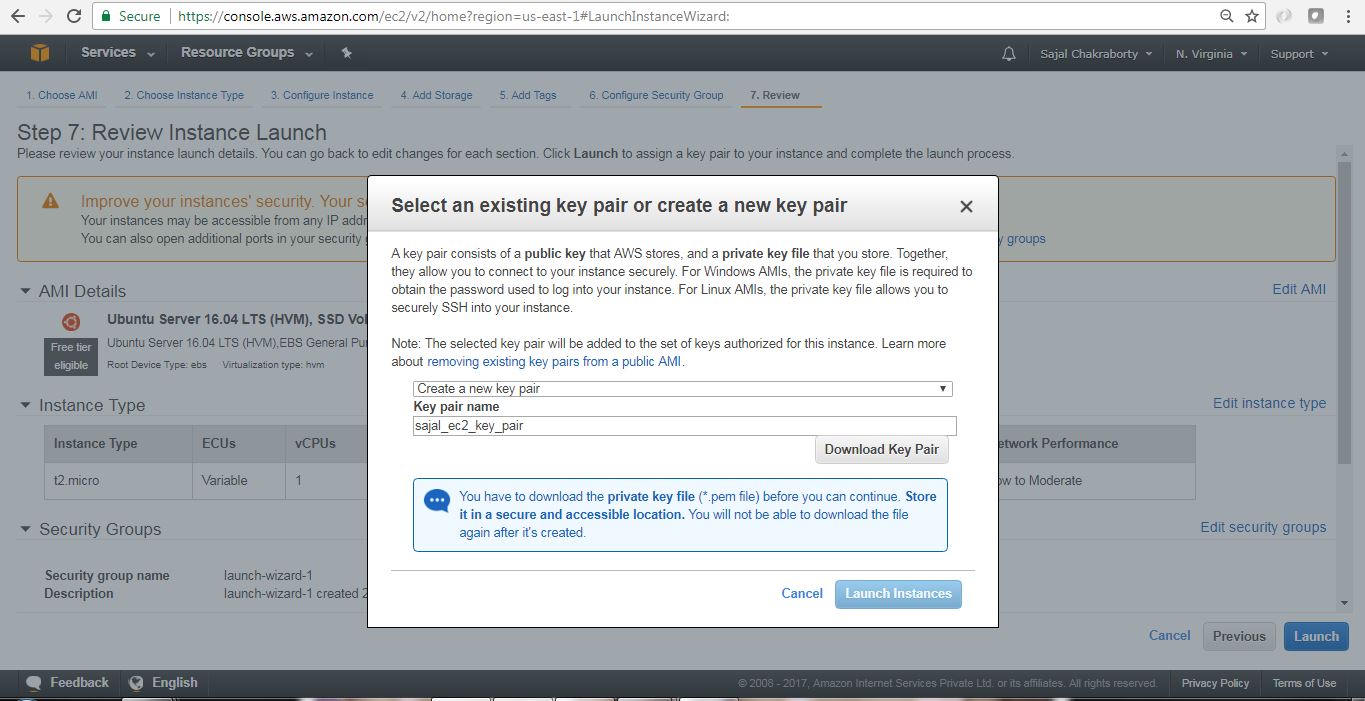
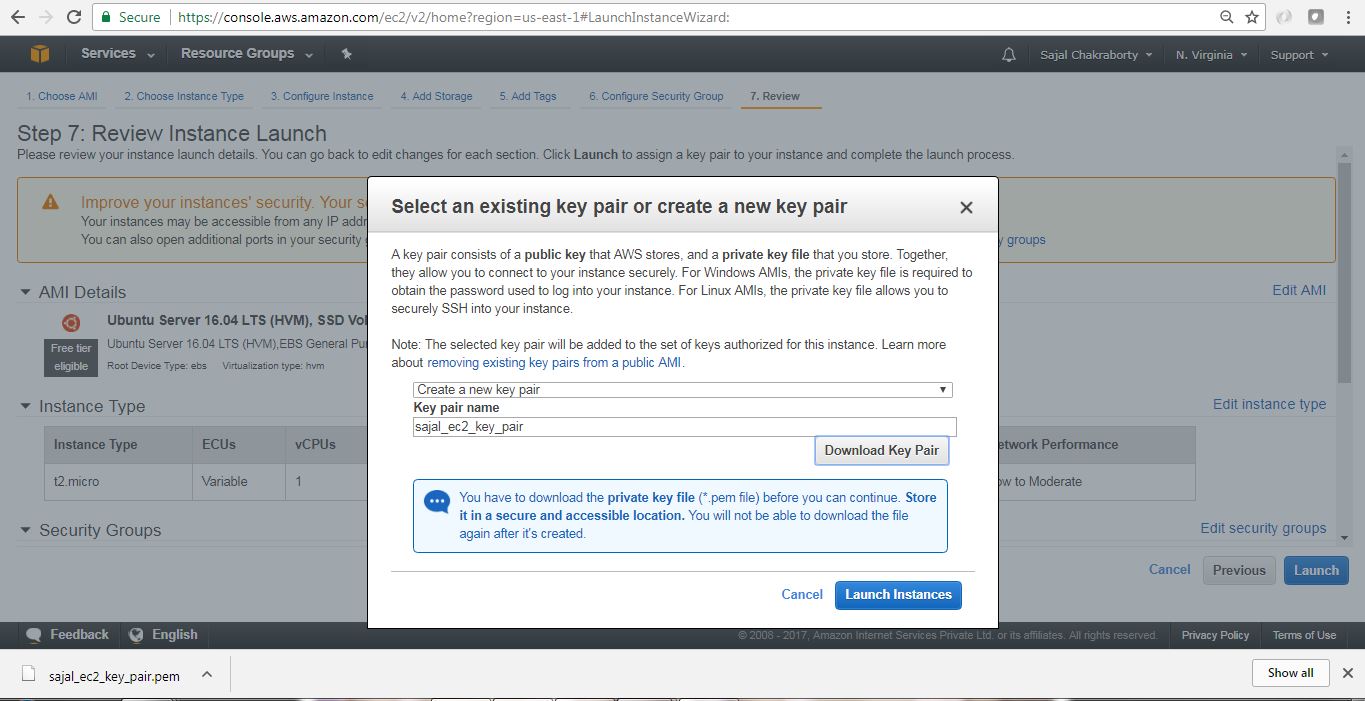
1. **Configure Security Group** – This section is to define a firewall that helps to secure our virtual server. A [security group](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html) is a virtual stateful firewall that controls inbound and outbound network traffic to AWS resources and Amazon EC2 instances. All Amazon EC2 instances must be launched into a security group. If a security group is not specified at launch, then the instance will be launched into the default security group for the Amazon VPC. The default security group allows communication between all resources within the security group, allows all outbound traffic, and denies all other traffic.In this screen we can create/reuse security groups based on our need.

In this example, assuming that we don’t have any security group created, We will choose the default option and will proceed. To proceed with next page, we will click on the Review and Launch button at the bottom right corner of the page. The Security Group configuration screen will look like:

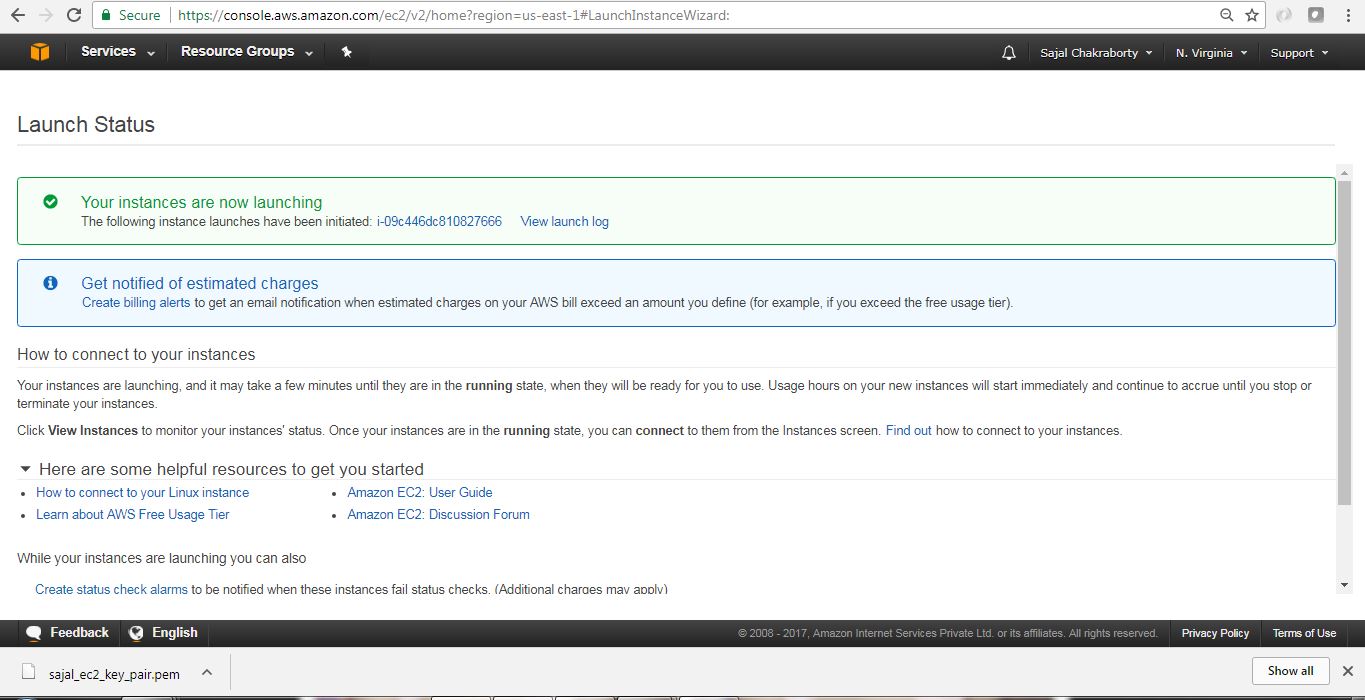
[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_7.jpg)Figure 8.0 : Security Group Configuration Screen

1. **Review and associate Key pair** – We will now review all the configurations that we have selected so far in this screen and to proceed further we will click on Launch button at the bottom right corner of the page.[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/EC2_8.jpg)Figure 9.0 : Review Launch configurations

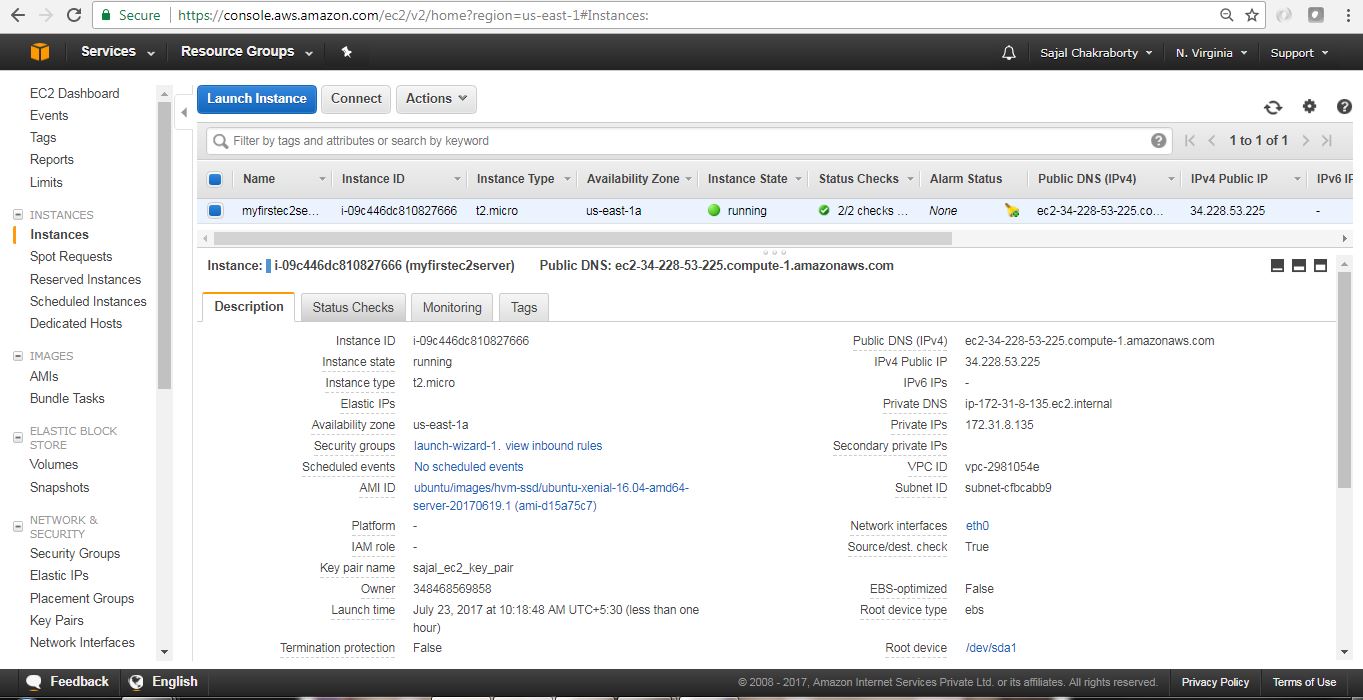
Once we proceed, we will have to select the Key pair which will be used for authentication while connecting with the instance.

1. **Create a new Key Pair** -Logging in to your virtual server requires a key. We use a key instead of a password to authenticate. We will create a new Key pair by choosing appropriate option from drop down and will have to provide a name of the key-pair file and then we need to download that file by clicking the Download Key Pair button in the page. A key is much more secure than a password, and using keys for SSH is mandatory for virtual servers running Linux on AWS. In our case we have chosen Ubuntu Linux, so creating a Key-pair is a must for us here. In this step if we create a new Key-pair, we get one file of type .pem extension. Here are the steps of creating the Key pairs. Instructions are self describing and very easy to follow.[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/EC2_9.jpg)Figure 10.0 : Key Pair Creation Step 1[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_10.jpg)Figure 10.1 : Key Pair Creation Step 2[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_11.jpg)Figure 10.2 : Key Pair Creation Step 3

Once we have created and downloaded the Key pair we need to proceed by clicking Launch Instances button in the **Figure 10.2** page.

1. **Finish Creation Process** – Once we have clicked Launch Instances button in the last page where we have created Key pairs, we will start the actual instance creation step as below.[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_13.jpg)Figure 11.0 : Instance Creating Last step

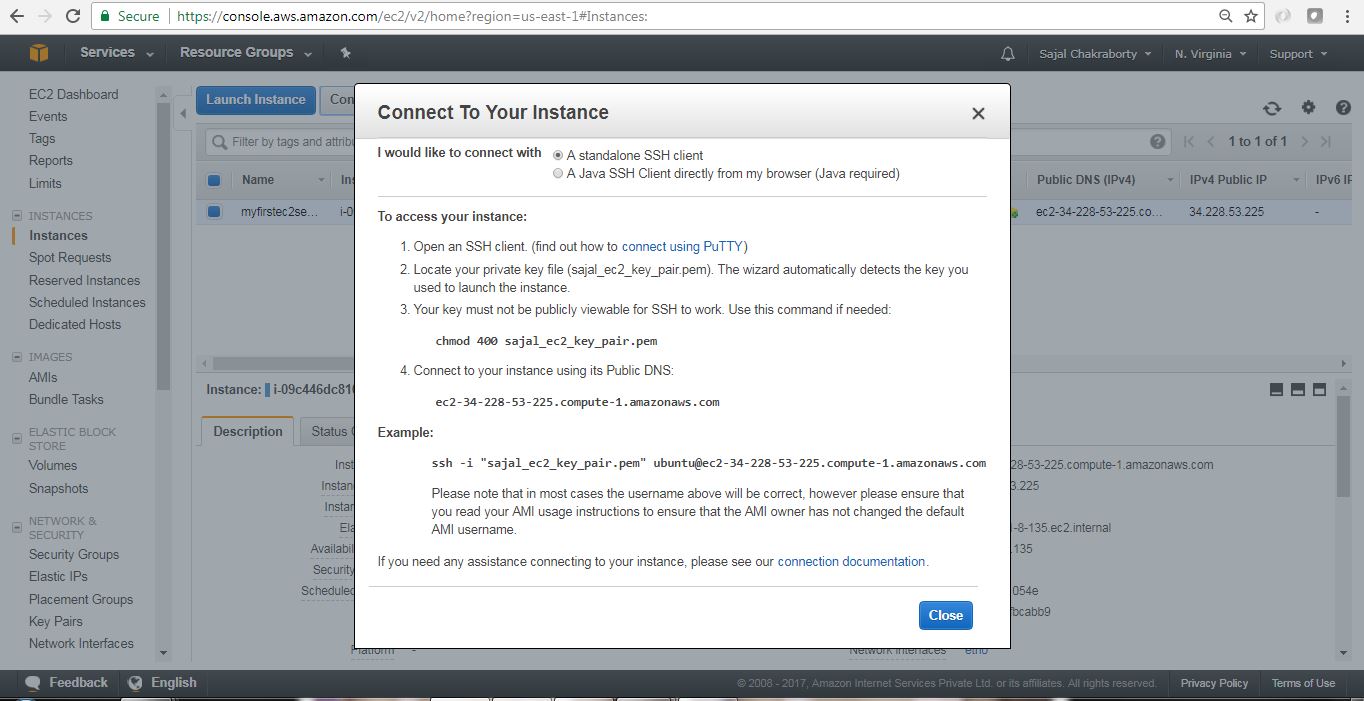
Finally this is the screen where we will see the details of the instance that we have just created.

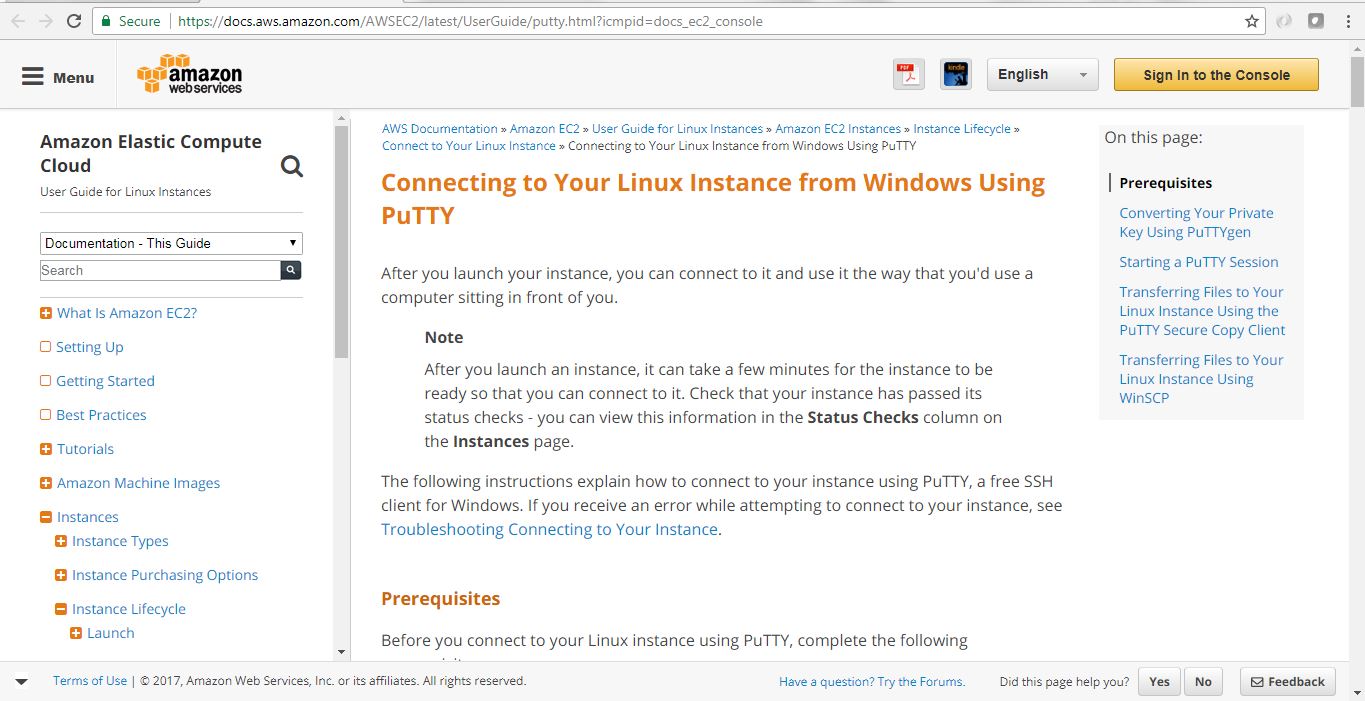
[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_14.jpg)Figure 11.1 : Instance Created

Now we have created our first EC2 Instance. Next we will connect to this instance from our local workstation and will install one software (Linkchecker) in EC2 instance and will learn to use that software.

**Connect to EC2 Instance**

AWS has provided a very good [documentation](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html?icmpid=docs_ec2_console) in this regard. Here are the basic steps for configuring and connecting to the instance through putty.

* **AWS provided steps –**In the instance details page, we can click Connect button to view the pop up like this. This will give us the required steps and information regarding connecting to the instance, Also we can go to the actual AWS documentation by going to the link in the pop up – connect using PuTTY link.[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_15.jpg)Connect Pop up Details

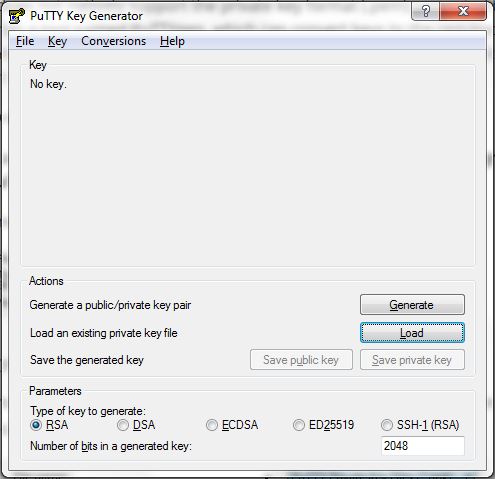
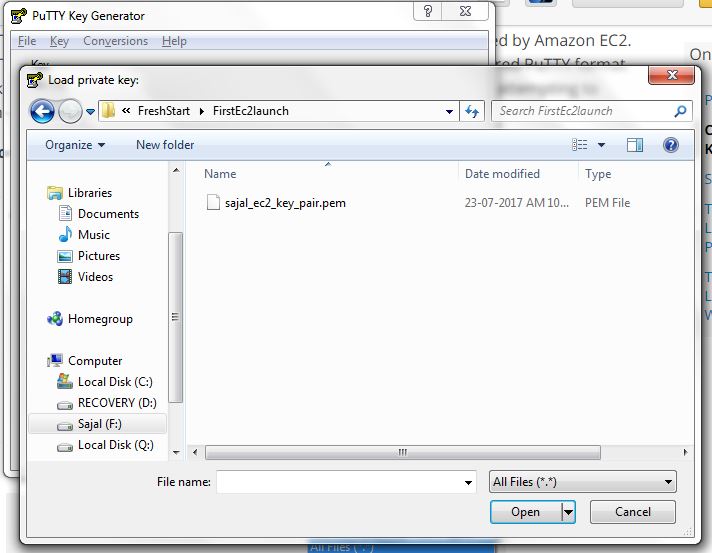
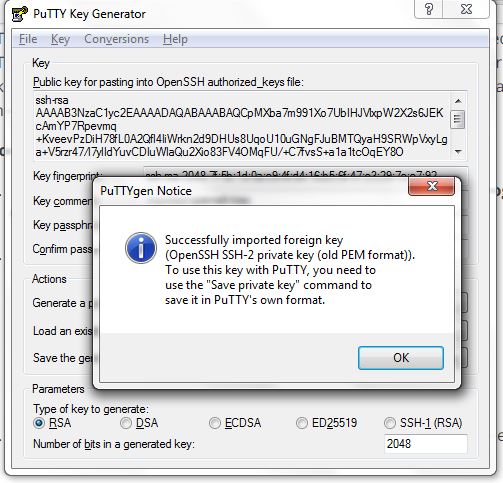
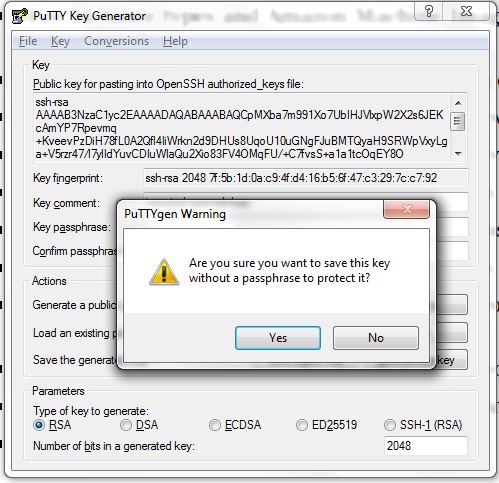
[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_16.jpg)Official AWS steps for PuTTY in Windows

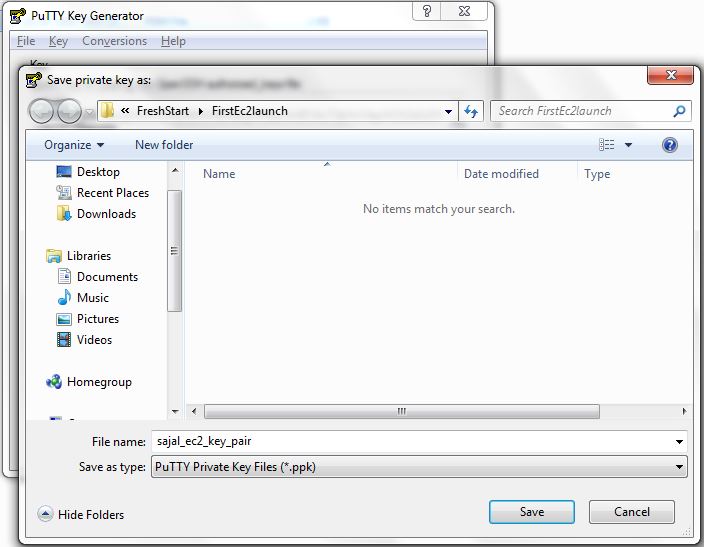
* **Use PuTTYgen to convert the Key File** – Now we need the .pem file we downloaded while creating EC2 instance. Now we will open PuTTYgen app to convert this .pem file to .pkk file which will be used by PuTTY to login to the instance terminal.

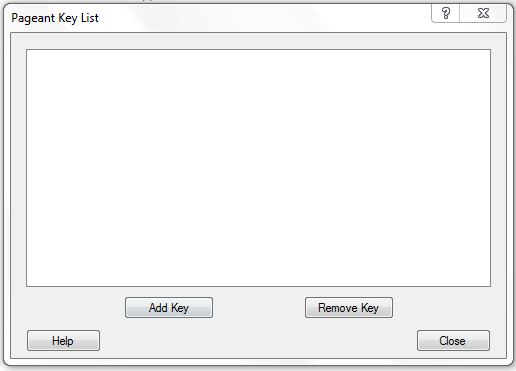
In the PuTTYgen application follow the below step to convert the .pem file.

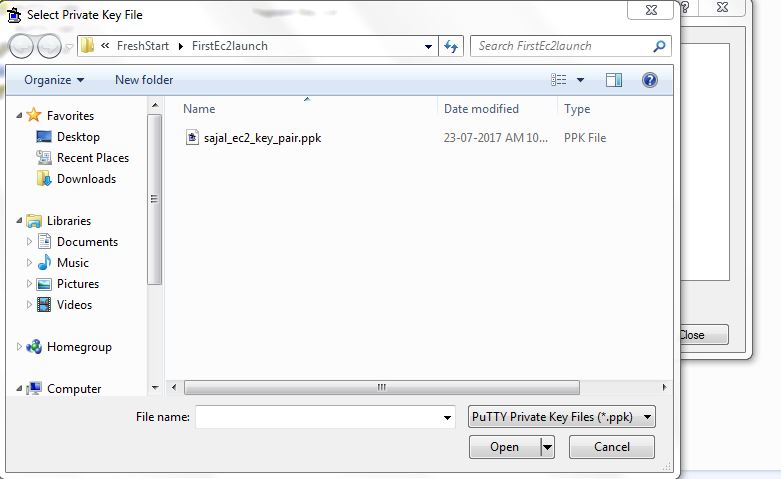
* 1. Run the application PuTTYgen.
  2. Select RSA radio button under Type of Key to Generate.
  3. Click Load.
  4. Because PuTTYgen displays only \*.pkk files, you need to switch the file extension of the File Name field to All Files.
  5. Select the .pem file [in my case it is sajal\_ec2\_key\_pair.pem], and click Open.
  6. Confirm the dialog box.
  7. Click Save Private Key. Ignore the warning about saving the key without a passphrase.

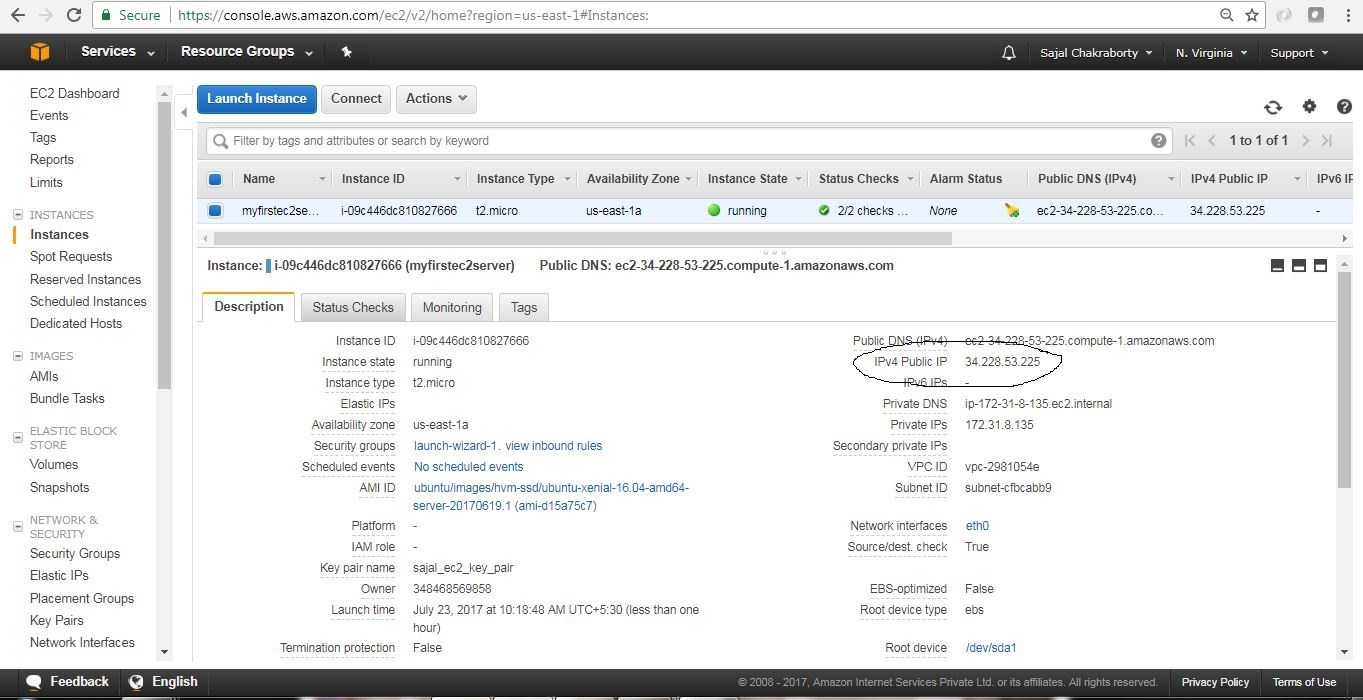
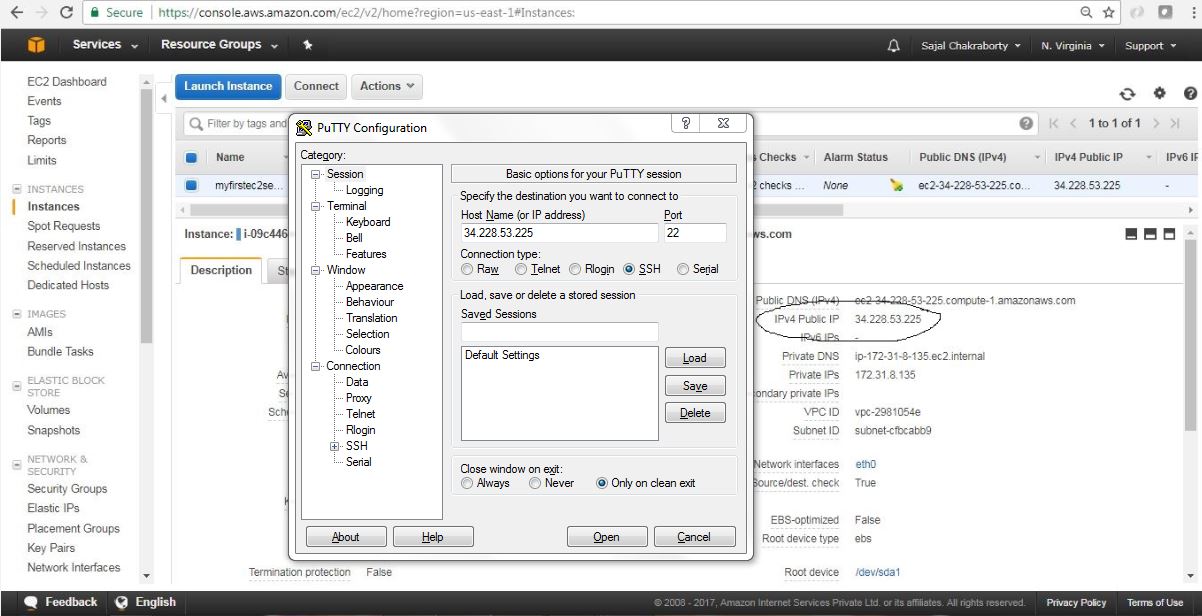
Here are few screen shots for those steps.

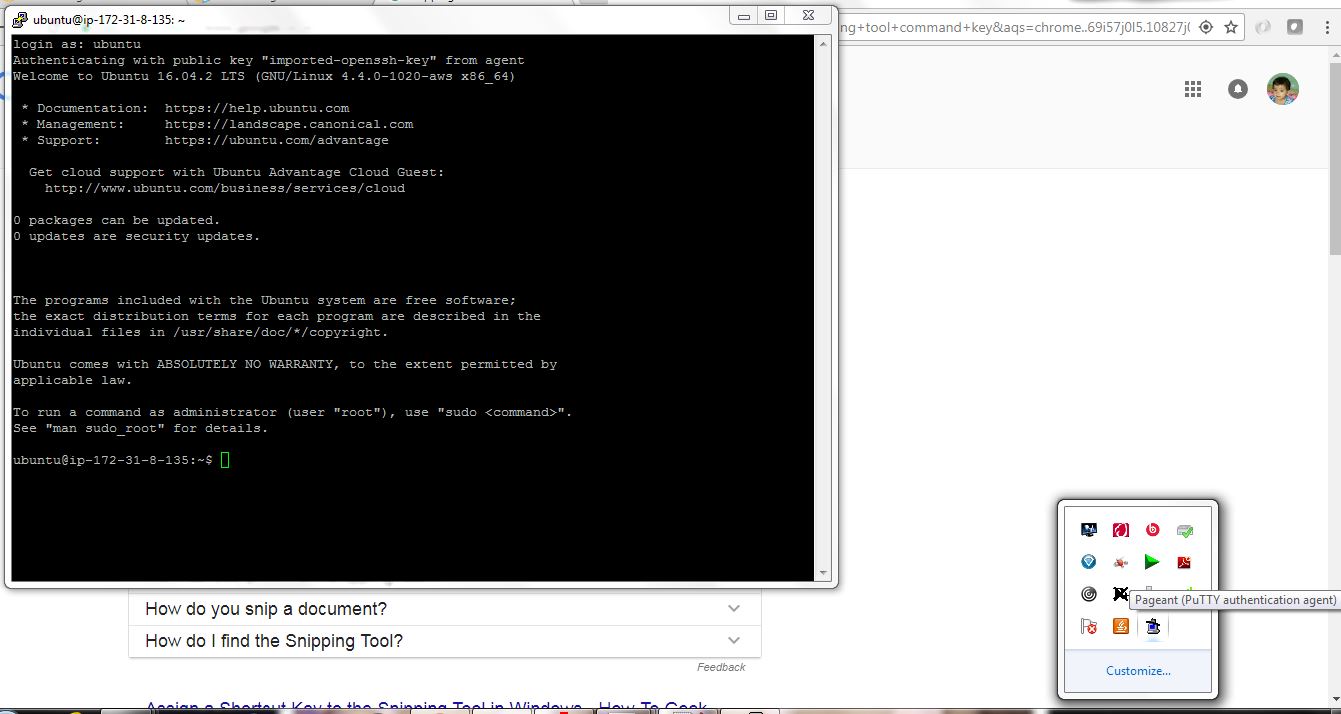
[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/putty1.jpg)PuTTYgen Steps[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/putty2.jpg)PuTTYgen Steps[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/putty3.jpg)PuTTYgen Steps[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/putty4.jpg)PuTTYgen Steps

[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/putty5.jpg)PuTTYgen Steps

* Start Pageant.EXE and select the .pkk file. Once Pageant stated we will need to add the Key by selecting the .pkk file we have already created. Here are the sample steps for this.[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/PAgent1.jpg)PAGENT step 1

[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/PAgent2.jpg)PAGENT step 1

* **Login to EC2 Instance** – Once we have Pageant.exe running and added the key .pkk file, Open Putty.exe and give public IP address of the Instance for connect. We can get the public ip fro the instance details screen in the AWS EC2 page. Now we need to give user name as **ubuntu** in the login as prompt. If everything goes well this would be authenticated and log in to the instance.[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/EC2_14-1.jpg)Public IP address for AWS console[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/Putty-With-Public-IP.jpg)Putty With Public IP

[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/putty7.jpg)Putty Login

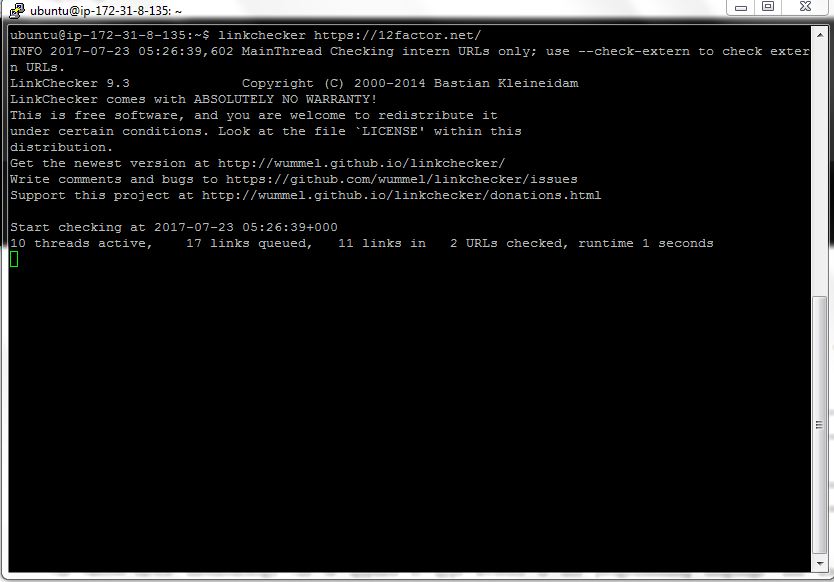
So now we are able to SSH connect to EC2 instance through putty.

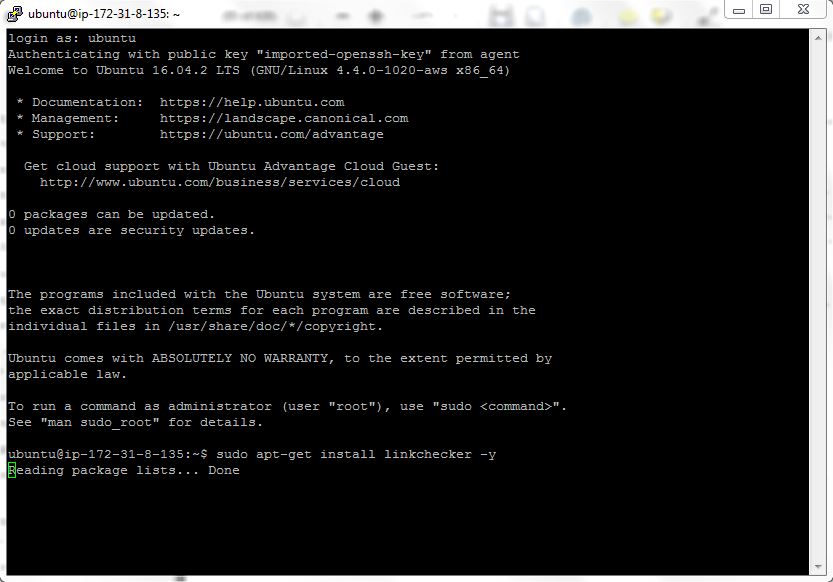
**Installing softwares on EC2 Instance**

Now we will install linkchecker software and will use that to do some testing.

* open putty and login ti the terminal as described above and enter the command sudo apt-get install linkchecker -y in the terminal. This will install the linkchecker software in the instance.
* Now test the linkchecker by simply providing some URL like linkchecker https://...

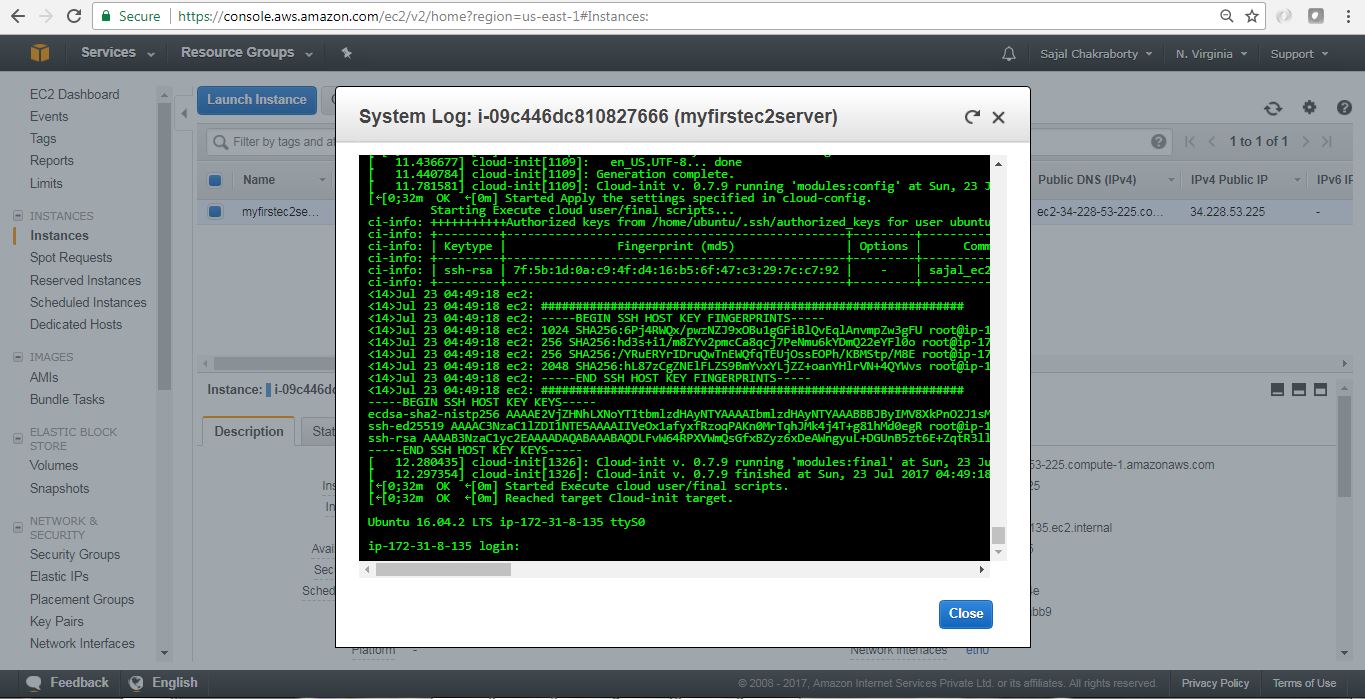
Here is the sample screen related to link checker install and usage.

[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/putty9.jpg)Check 12factor website

[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/putty8.jpg)Install Software

**Checking Logs on EC2 Instance**

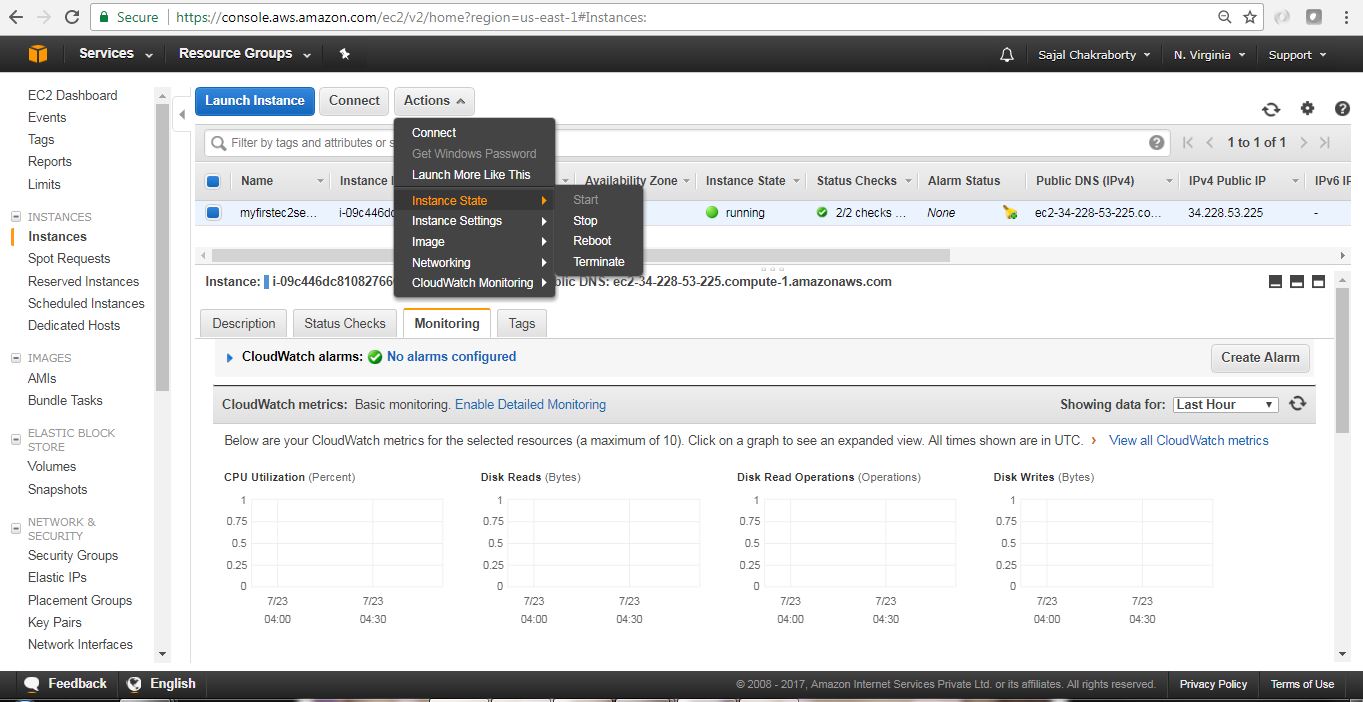
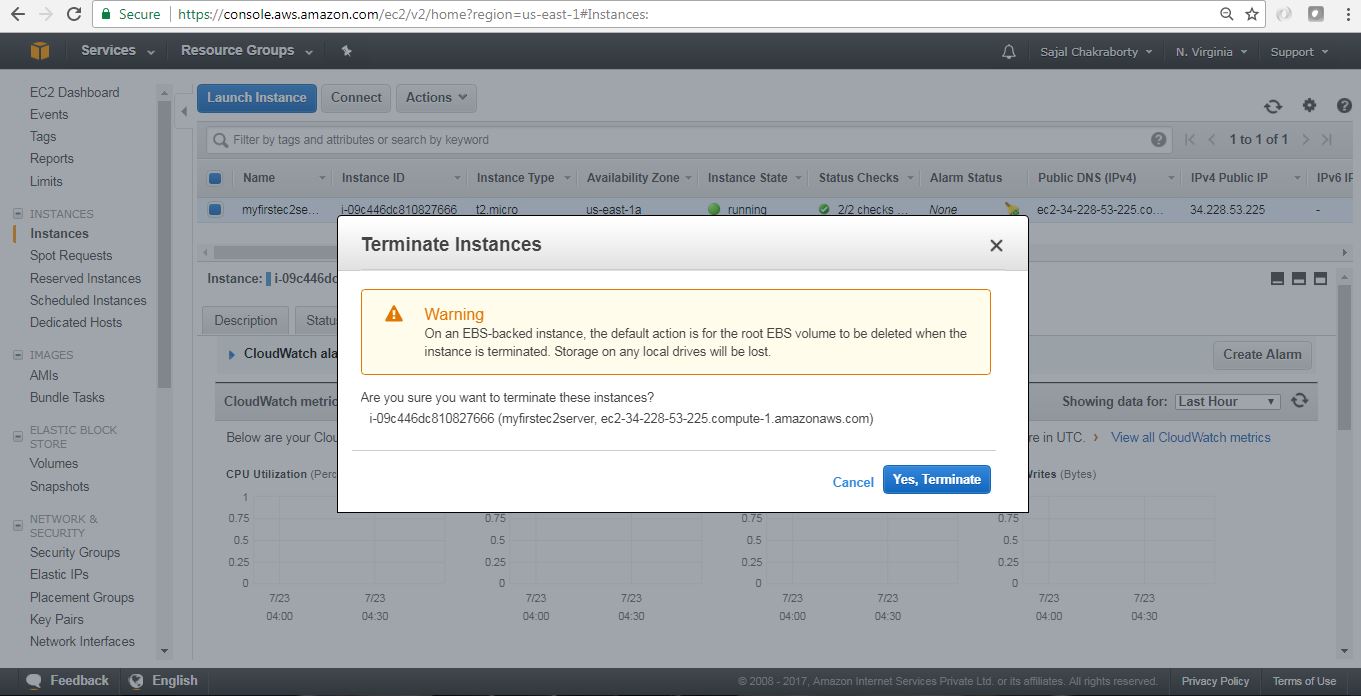
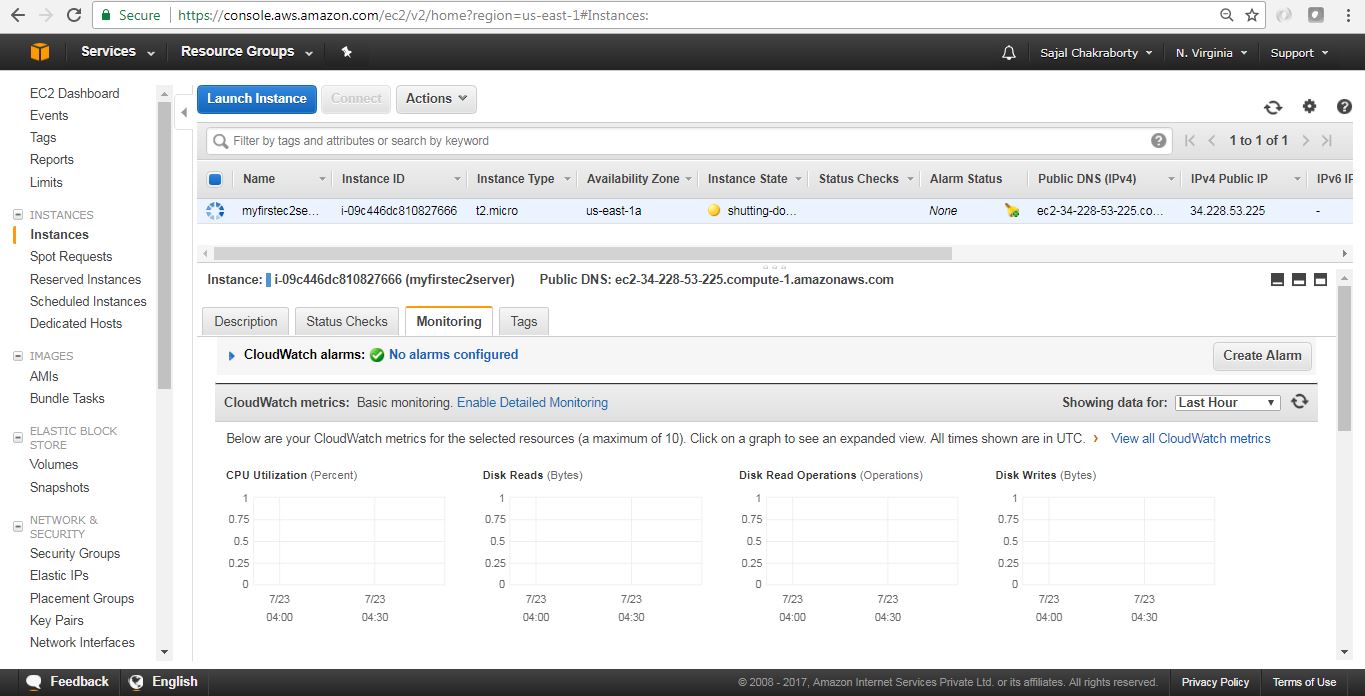
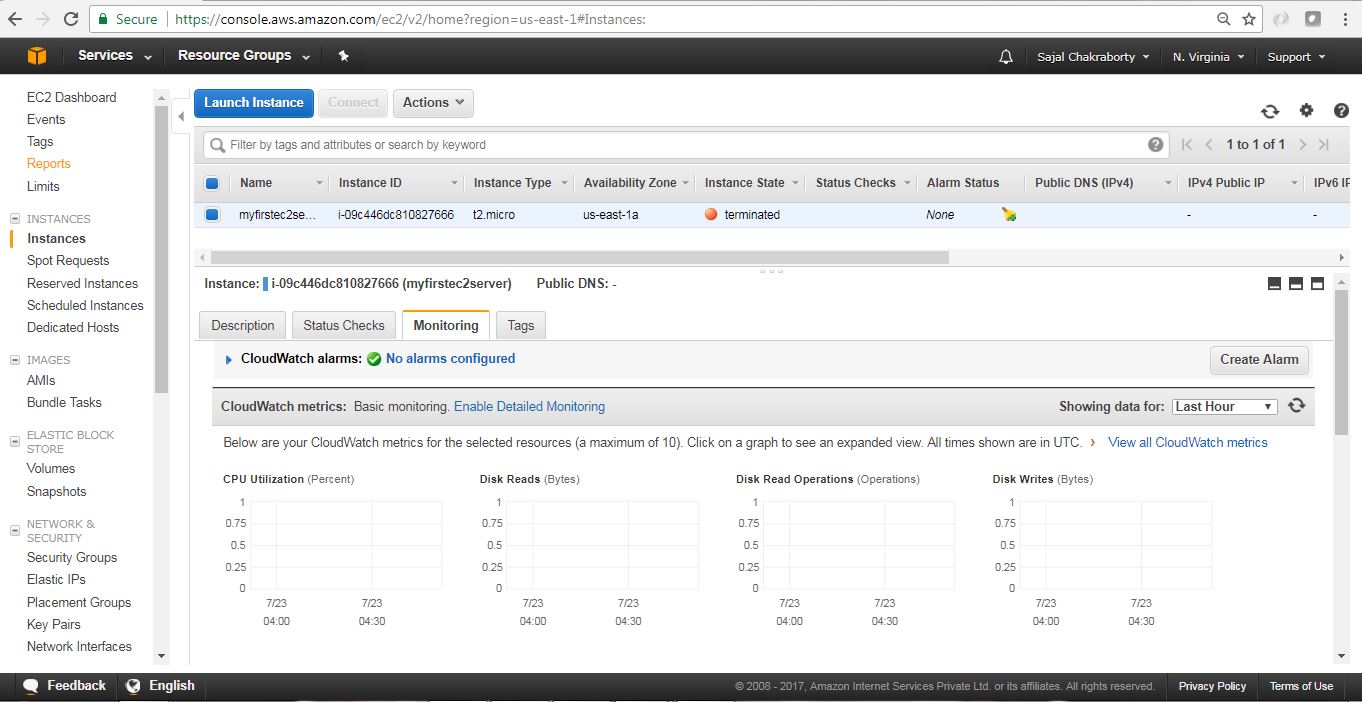
We can view the console log of EC2 instance from the AWS console itself by clicking on the **Actions**menu, choose **Instance Settings > Get System Log**. Here is the sample screen looks like:

[](https://cdn1.howtodoinjava.com/wp-content/uploads/2017/07/EC2_viewLogs.jpg)EC2 view System Logs

**How to Terminate EC2 Instance**

We can terminate EC2 instance by clicking the menu option **Actions > Instance State > Terminate**.

Here are the consequent steps related to termination of the instance. Please note that termination means Deletion/removal of the instance from AWS system. Also we can stop the instance. Please choose accordingly.

[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/EC2_Terminate.jpg)EC2 Instance Termination[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/EC2_Terminate1.jpg)EC2 Instance Termination[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/EC2_Terminate2.jpg)EC2 Instance Termination[](https://cdn2.howtodoinjava.com/wp-content/uploads/2017/07/EC2_Terminate3.jpg)EC2 Instance Terminationv