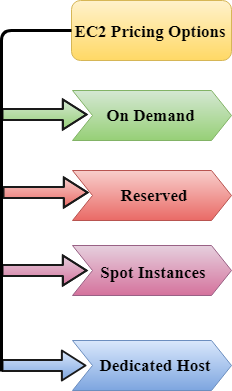
EC2

* EC2 stands for Amazon Elastic Compute Cloud.
* Amazon EC2 is a web service that provides resizable compute capacity in the cloud.
* Amazon EC2 reduces the time required to obtain and boot new user instances to minutes rather than in older days, if you need a server then you had to put a purchase order, and cabling is done to get a new server which is a very time-consuming process. Now, Amazon has provided an EC2 which is a virtual machine in the cloud that completely changes the industry.
* You can scale the compute capacity up and down as per the computing requirement changes.
* Amazon EC2 changes the economics of computing by allowing you to pay only for the resources that you actually use. Rather than you previously buy physical servers, you would look for a server that has more CPU capacity, RAM capacity and you buy a server over 5 year term, so you have to plan for 5 years in advance. People spend a lot of capital in such investments. EC2 allows you to pay for the capacity that you actually use.
* Amazon EC2 provides the developers with the tools to build resilient applications that isolate themselves from some common scenarios.

EC2 Pricing Options



On Demand

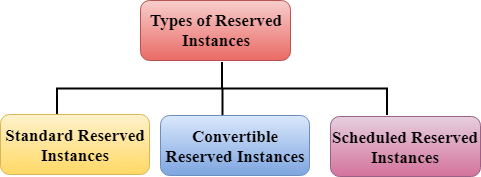
* It allows you to pay a fixed rate by the hour or even by the second with no commitment.
* Linux instance is by the second and windows instance is by the hour.
* On Demand is perfect for the users who want low cost and flexibility of Amazon EC2 without any up-front investment or long-term commitment.
* It is suitable for the applications with short term, spiky or unpredictable workloads that cannot be interrupted.
* It is useful for the applications that have been developed or tested on Amazon EC2 for the first time.
* On Demand instance is recommended when you are not sure which instance type is required for your performance needs.

Reserved

* It is a way of making a reservation with Amazon or we can say that we make a contract with Amazon. The contract can be for 1 or 3 years in length.
* In a Reserved instance, you are making a contract means you are paying some upfront, so it gives you a significant discount on the hourly charge for an instance.
* It is useful for applications with steady state or predictable usage.
* It is used for those applications that require reserved capacity.
* Users can make up-front payments to reduce their total computing costs. For example, if you pay all your upfronts and you do 3 years contract, then only you can get a maximum discount, and if you do not pay all upfronts and do one year contract then you will not be able to get as much discount as you can get If you do 3 year contract and pay all the upfronts.

Types of Reserved Instances:

* Standard Reserved Instances
* Convertible Reserved Instances
* Scheduled Reserved Instances



Standard Reserved Instances

* It provides a discount of up to 75% off on demand. For example, you are paying all up-fronts for 3 year contract.
* It is useful when your Application is at the steady-state.

Convertible Reserved Instances

* It provides a discount of up to 54% off on demand.
* It provides the feature that has the capability to change the attributes of RI as long as the exchange results in the creation of Reserved Instances of equal or greater value.
* Like Standard Reserved Instances, it is also useful for the steady state applications.

Scheduled Reserved Instances

* Scheduled Reserved Instances are available to launch within the specified time window you reserve.
* It allows you to match your capacity reservation to a predictable recurring schedule that only requires a fraction of a day, a week, or a month.

Spot Instances

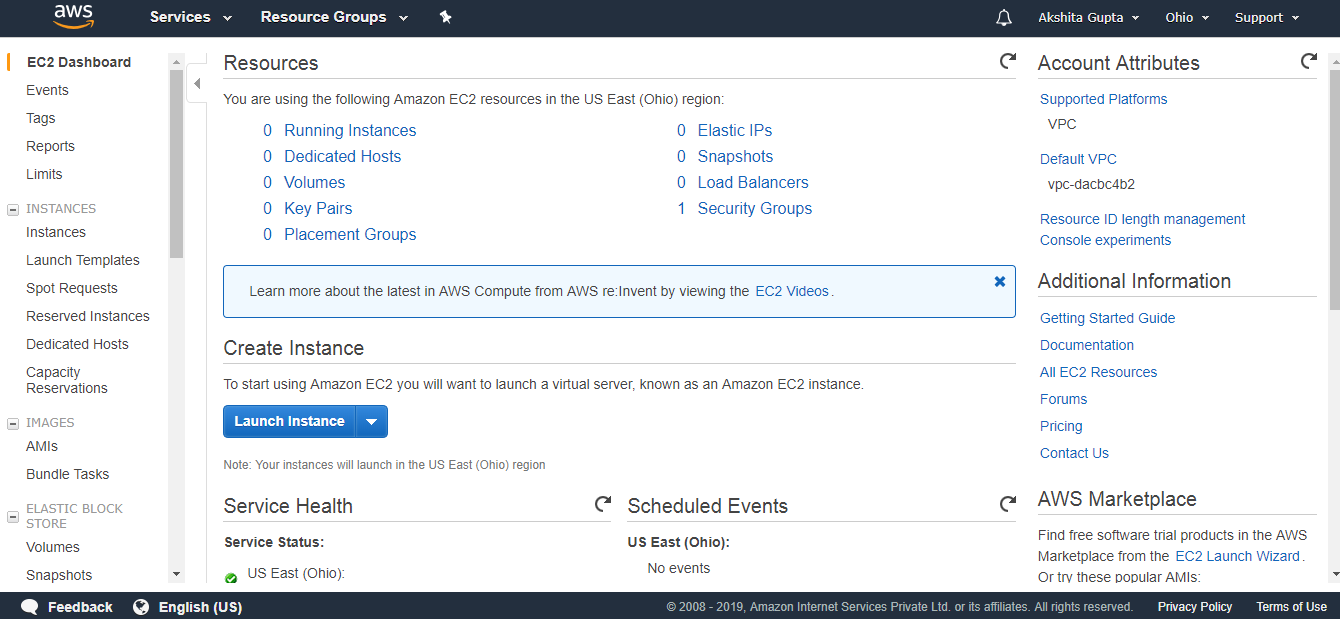
* It allows you to bid for a price whatever price that you want for instance capacity, and providing better savings if your applications have flexible start and end times.
* Spot Instances are useful for those applications that have flexible start and end times.
* It is useful for those applications that are feasible at very low compute prices.
* It is useful for those users who have an urgent need for large amounts of additional computing capacity.
* EC2 Spot Instances provide less discounts as compared to On Demand prices.
* Spot Instances are used to optimize your costs on the AWS cloud and scale your application's throughput up to 10X.
* EC2 Spot Instances will continue to exist until you terminate these instances.

Dedicated Hosts

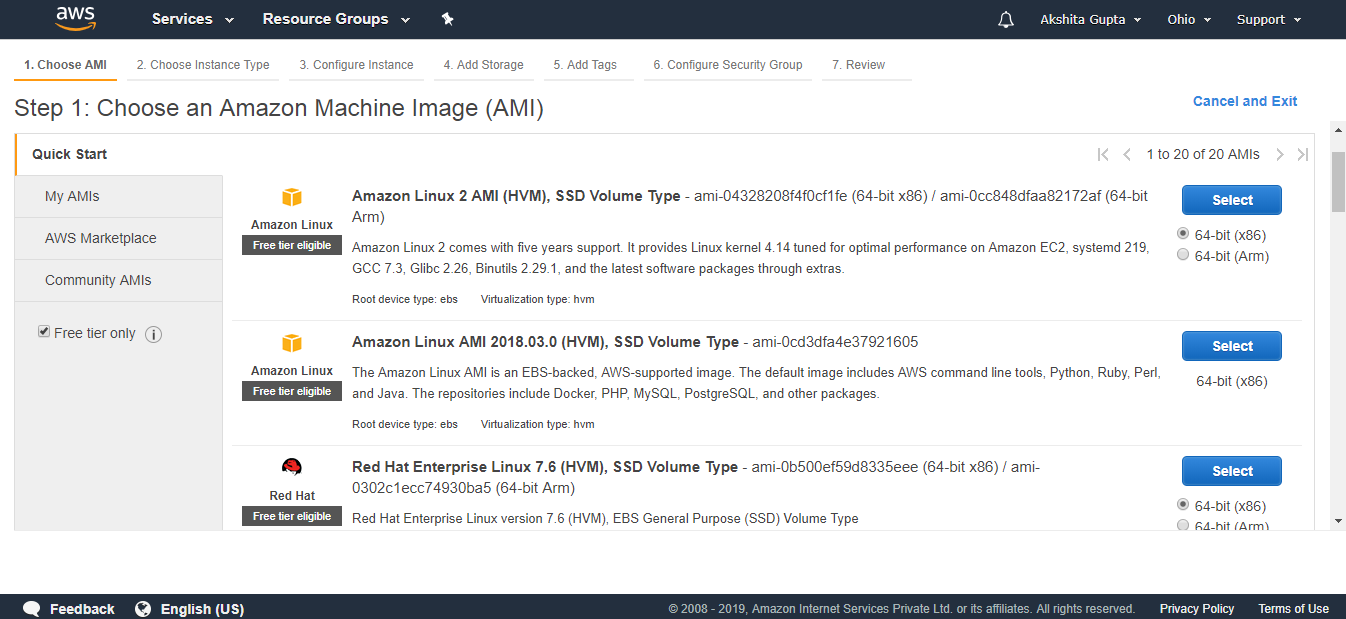
* A dedicated host is a physical server with EC2 instance capacity which is fully dedicated to your use.
* The physical EC2 server is the dedicated host that can help you to reduce costs by allowing you to use your existing server-bound software licenses. For example, Vmware, Oracle, SQL Server depending on the licenses that you can bring over to AWS and then they can use the Dedicated host.
* Dedicated hosts are used to address compliance requirements and reduces host by allowing to use your existing server-bound server licenses.
* It can be purchased as a Reservation for up to 70% off On-Demand price.

Creating an EC2 instance

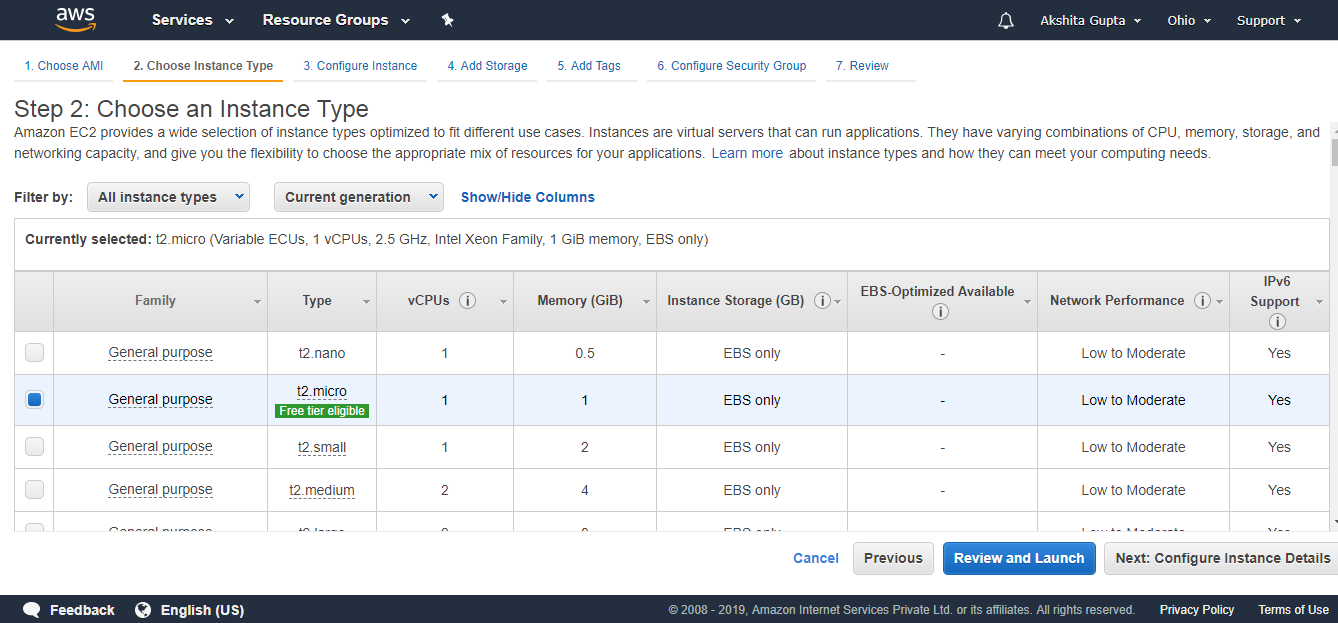
* Sign in to the AWS Management Console.
* Click on the EC2 service.
* Click on the **Launch Instance** button to create a new instance.



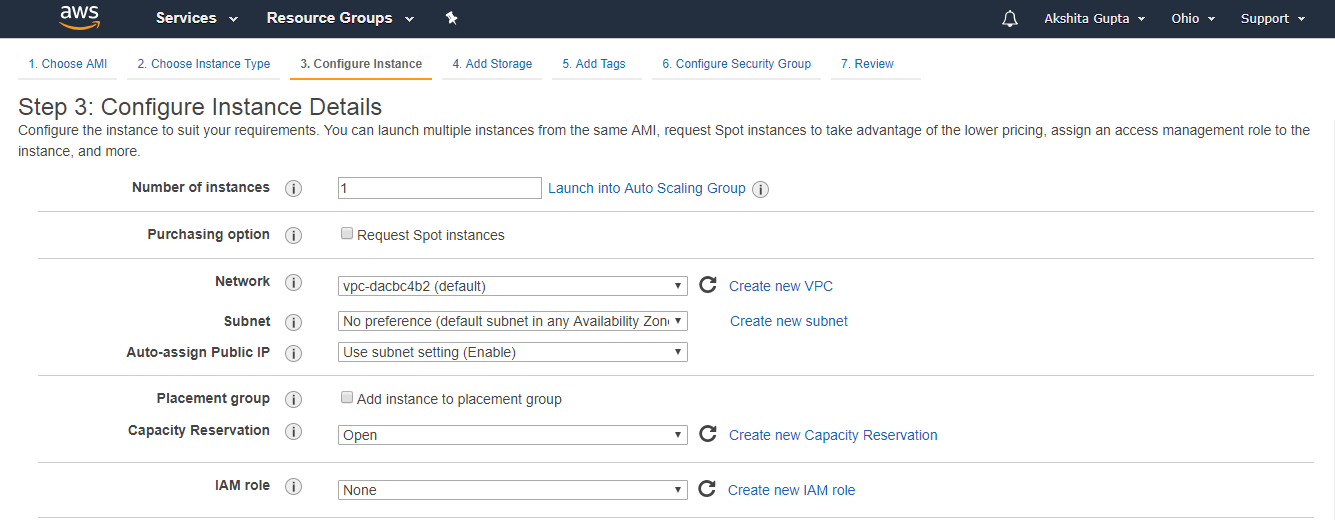
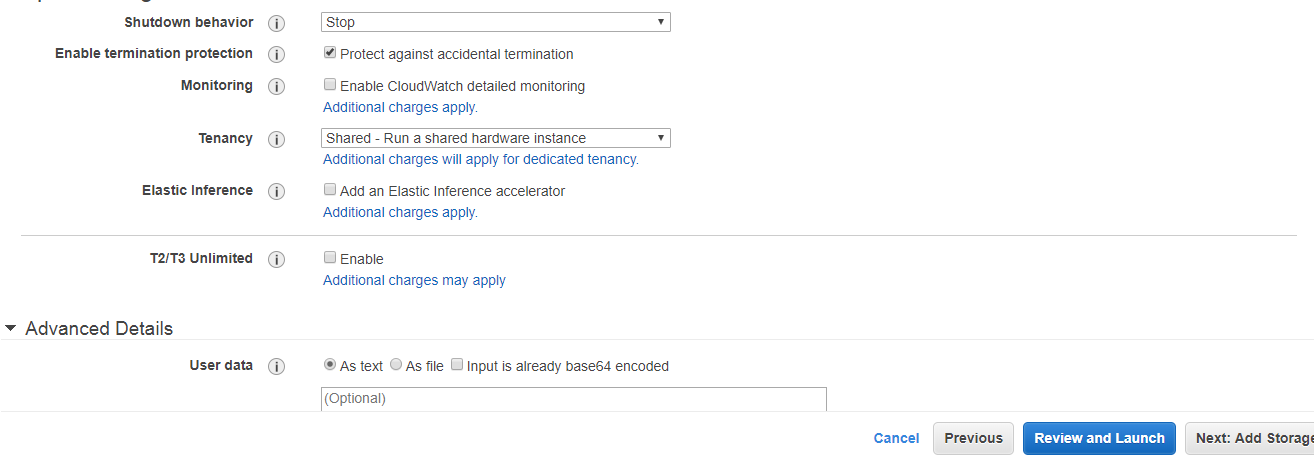
* Now, we have different Amazon Machine Images. These are the snapshots of different virtual machines. We will be using Amazon Linux AMI 2018.03.0 (HVM) as it has built-in tools such as java, python, ruby, perl, and especially AWS command line tools.



* Choose an Instance Type, and then click on the Next. Suppose I choose a t2.micro as an instance type.



* The main setup page of EC2 is shown below where we define setup configuration.

**Where,**

**Number of Instances:** It defines how many EC2 instances you want to create. I leave it as 1 as I want to create only one instance.

**Purchasing Option:** In the purchasing option, you need to set the price, request from, request to, and persistent request. Right now, I leave it as unchecked.

**Tenancy:** Click on the **Shared-Run a shared hardware instance** from the dropdown menu as we are sharing hardware.

**Network:** Choose your network, set it as default, i.e., **vpc-dacbc4b2 (default)** where vpc is a virtual private cloud where we can launch the AWS resources such as EC2 instances in a virtual cloud.

**Subnet:** It is a range of IP addresses in a virtual cloud. In a specified subnet, you can add new AWS resources.

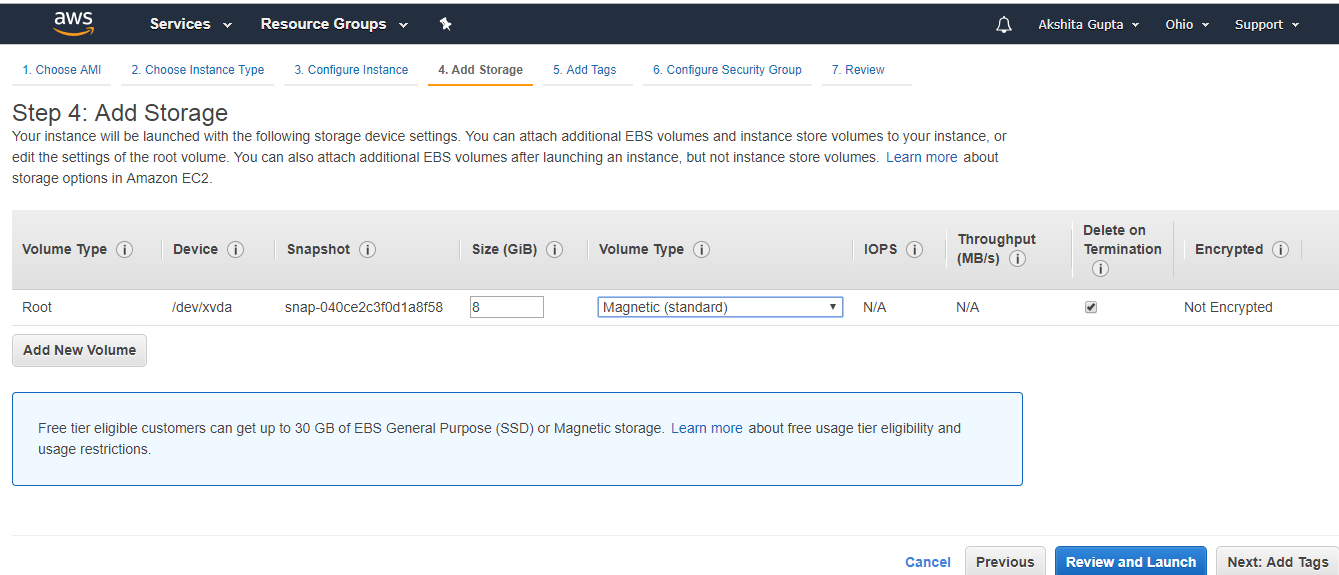
**Shutdown behavior:** It defines the behavior of the instance type. You can either stop or terminate the instance when you shut down the Linux machine. Now, I leave it as Stop.

**Enable Termination Protection:** It allows the people to protect against the accidental termination.

**Monitoring:** We can monitor things such as CPU utilization. Right now, I uncheck the Monitoring.

**User data:** In Advanced details, you can pass the bootstrap scripts to EC2 instance. You can tell them to download PHP, Apache, install the Apache, etc.

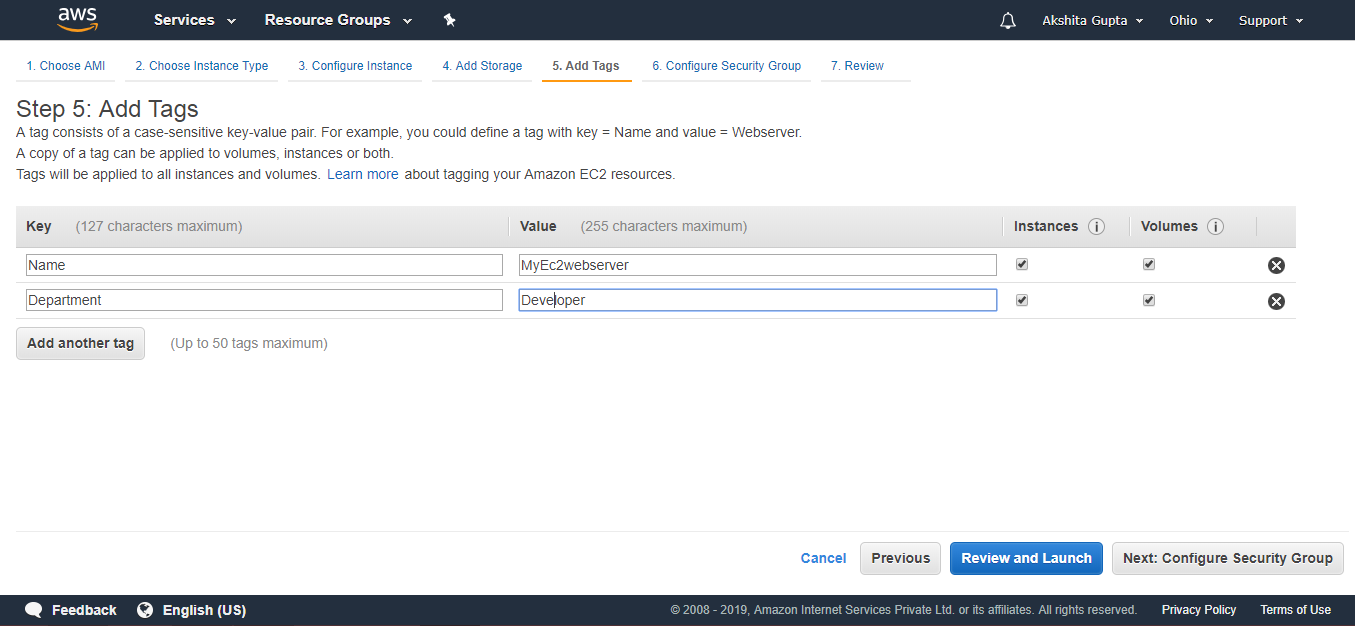
* Now, add the EBS volume and attach it to the EC2 instance. Root is the default EBS volume. Click on the **Next.**



**Volume Type:** We select the Magnetic (standard) as it is the only disk which is bootable.

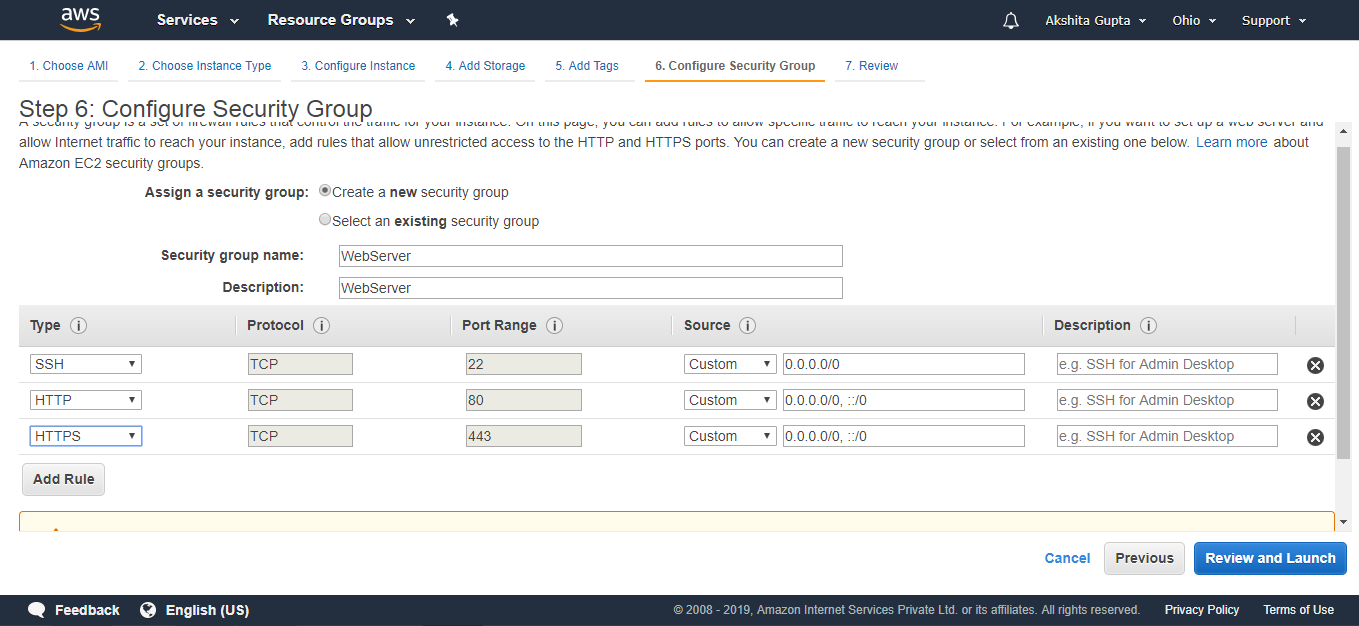
**Delete on termination:** It is checked means that the termination of an EC2 instance will also delete EBS volume.

* Now, Add the Tags and then click on the Next.

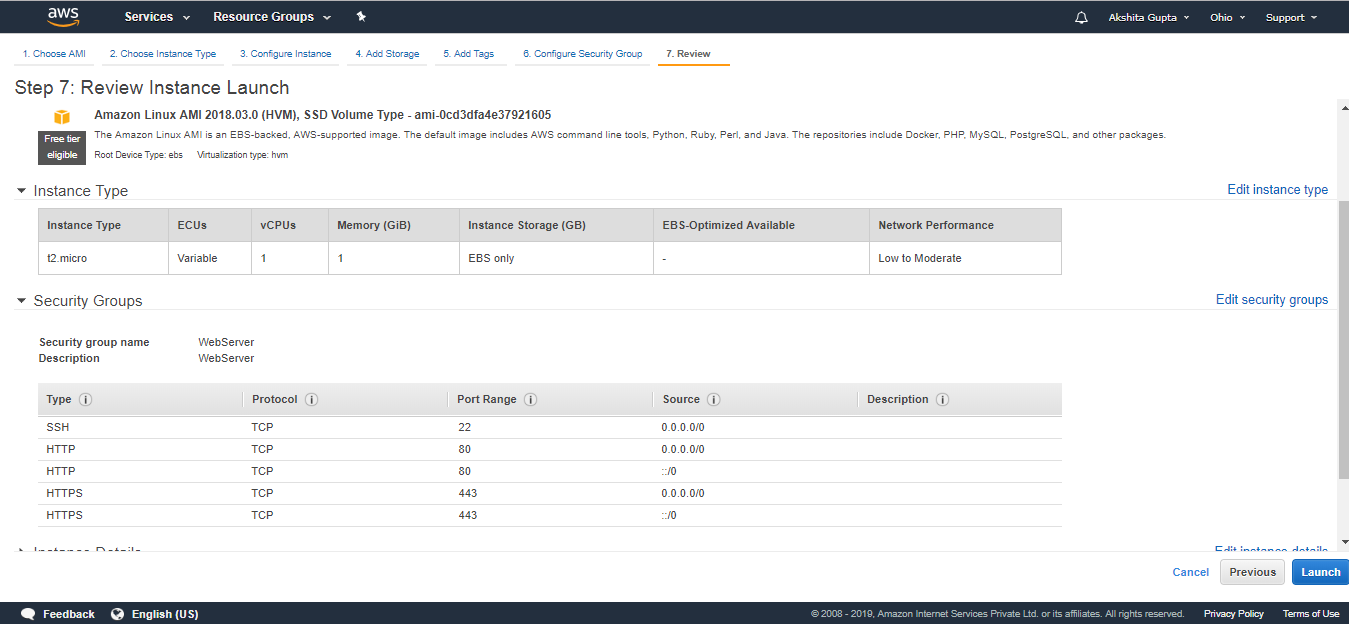


In the above screen, we observe that we add two tags, i.e., the name of the server and department. Create as many tags as you can as it reduces the overall cost.

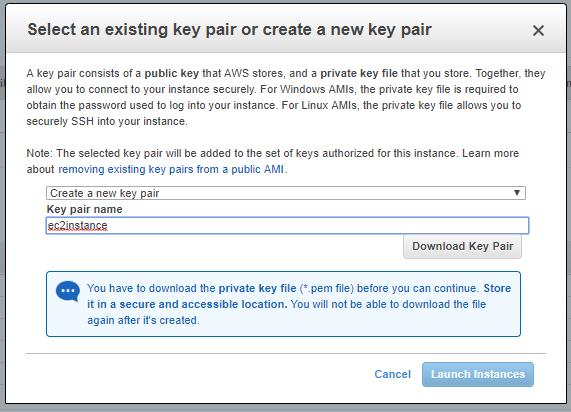
* Configure Security Group. The security group allows some specific traffic to access your instance.



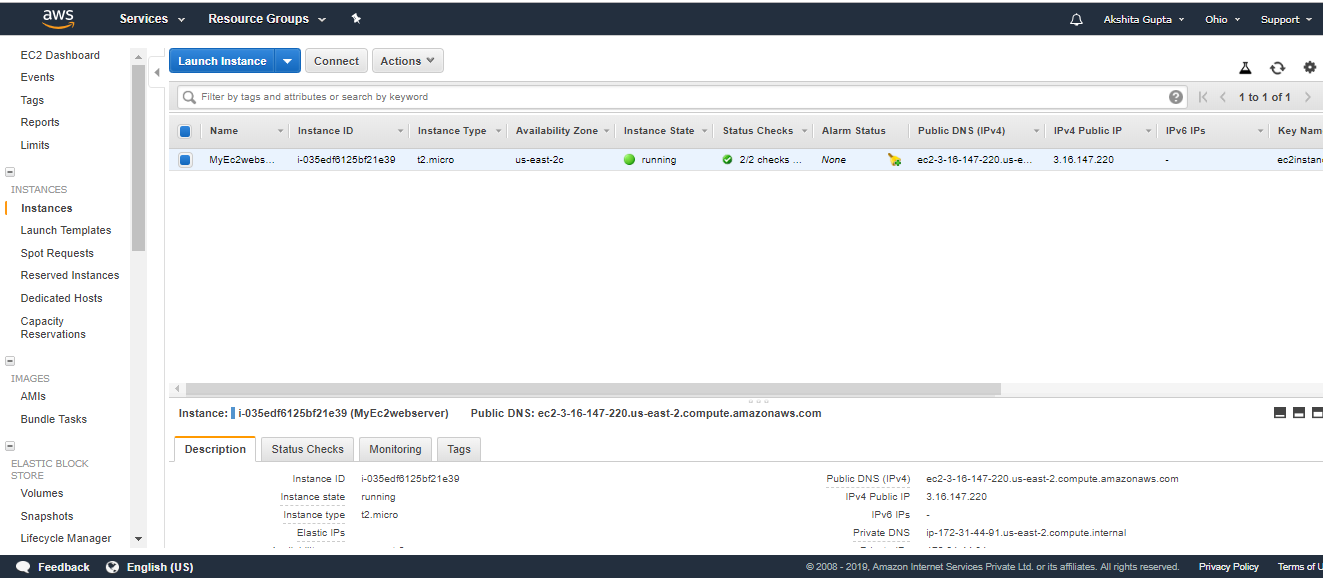
* Review an EC2 instance that you have just configured, and then click on the Launch button.



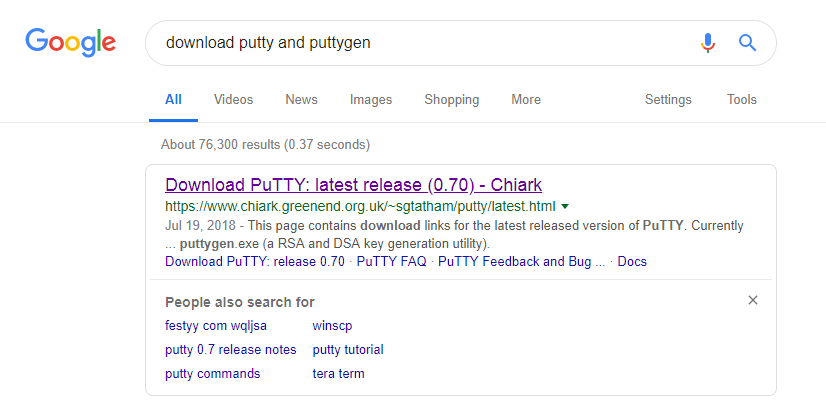
* Create a new key pair and enter the name of the key pair. Download the Key pair.



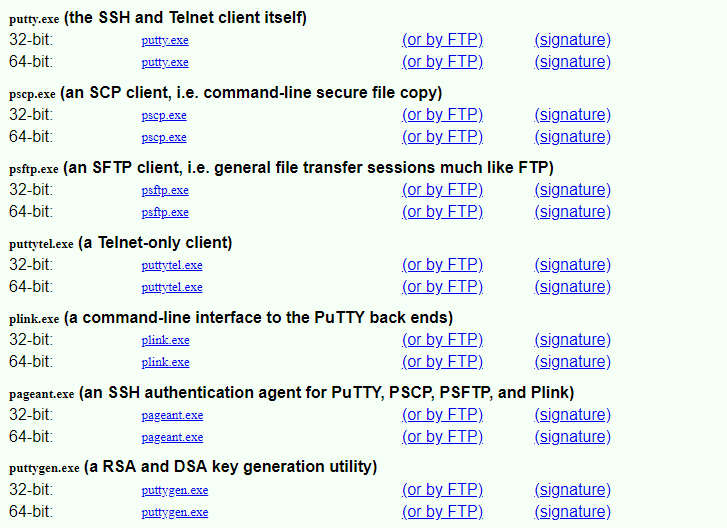
* Click on the Launch Instances button.



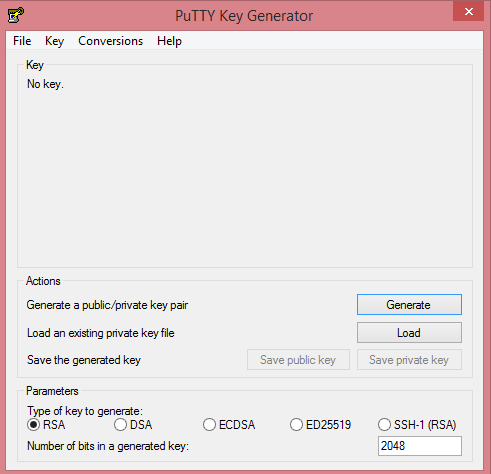
* To use an EC2 instance in Windows, you need to install both **Putty** and **PuttyKeyGen.**
* Download the **Putty** and **PuttyKeyGen.**



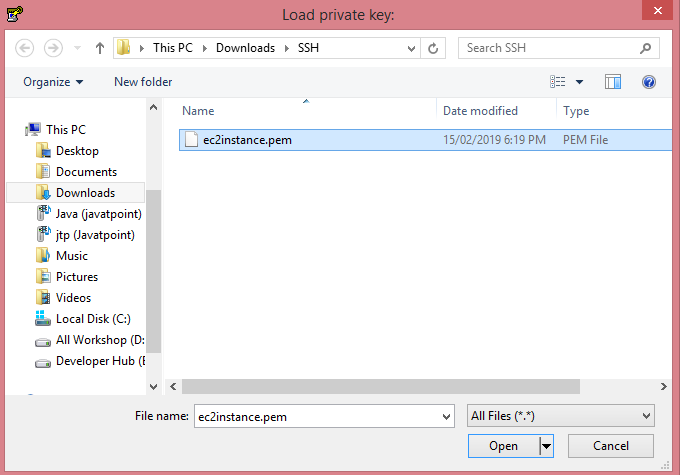
* Download the putty.exe and puttygen.exe file.



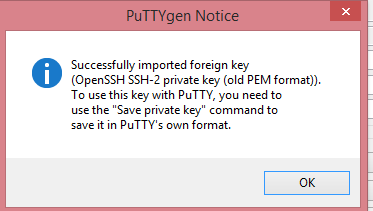
* In order to use the key-pair which we have downloaded previously, we need to convert the pem file to ppk file. Puttygen is used to convert the pem file to ppk file.
* Open the Puttygen software.
* Click on the Load.



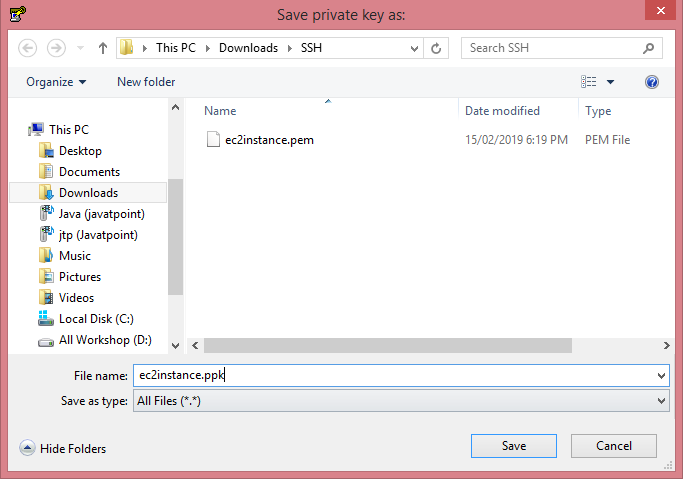
* Open the key-pair file, i.e., ec2instance.pem.



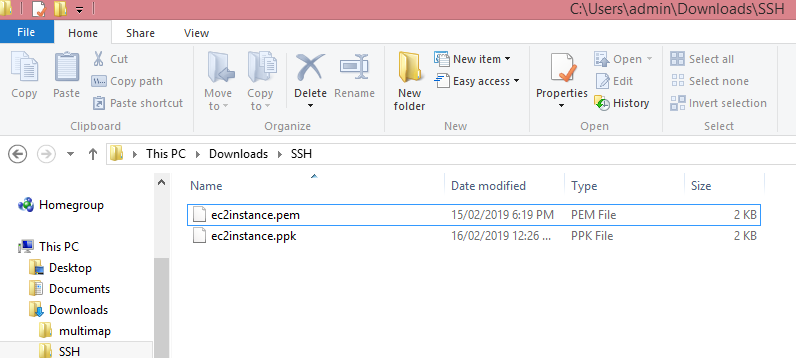
* Click on the OK button.



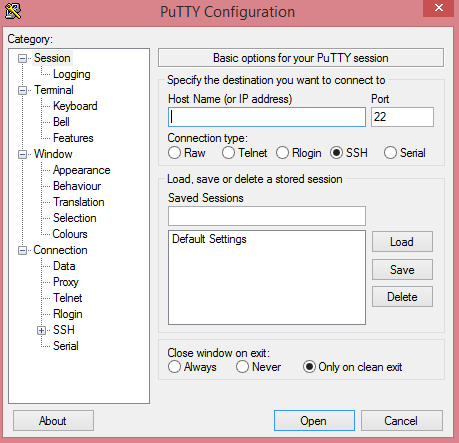
* Click on the Save private key. Change the file extension from pem to ppk.



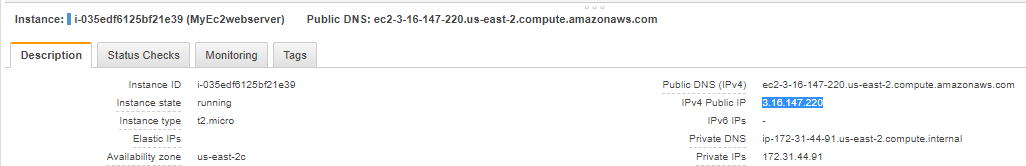
* Click on the Save button.
* Move to the download directory where the ppk file is downloaded.



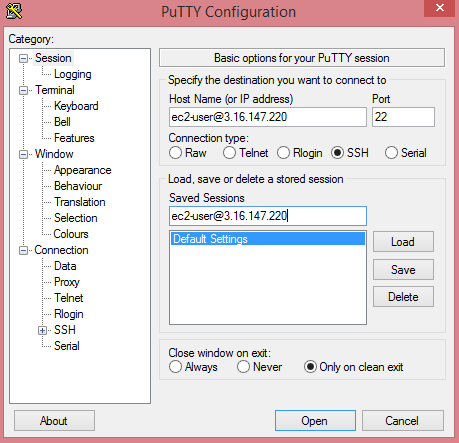
* Open the Putty.



* Move to the EC2 instance that you have created and copy its IP address.

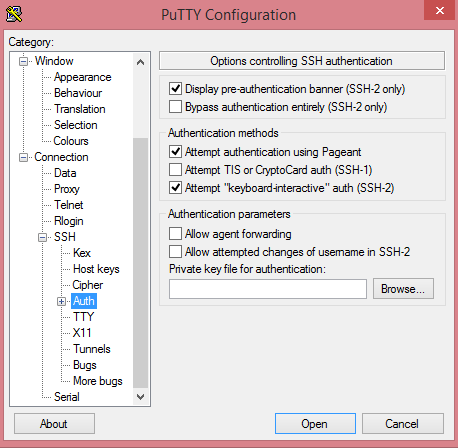


* Now, move to the Putty configuration and type ec2user@, and then paste the IP address that you have copied in a previous step. Copy the Host Name in Saved Sessions.

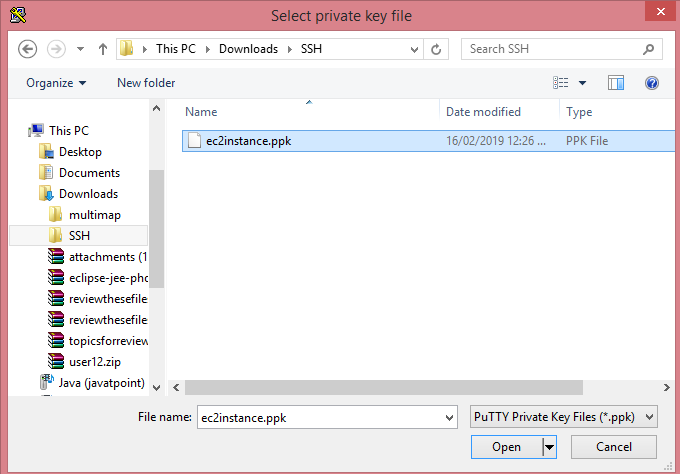


Now, your Host Name is saved in the default settings.

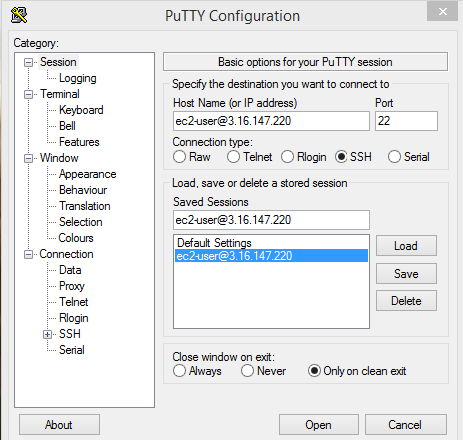
* Click on the SSH category appearing on the left side of the Putty, then click on the Auth.



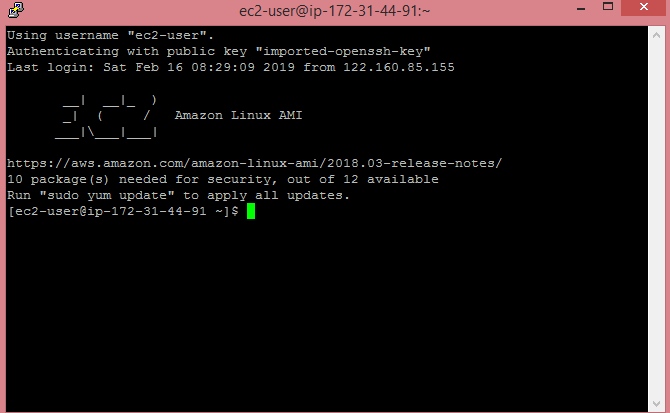
* Click on the Browse to open the ppk file.



* Move to the Session category, click on the Save to save the settings.

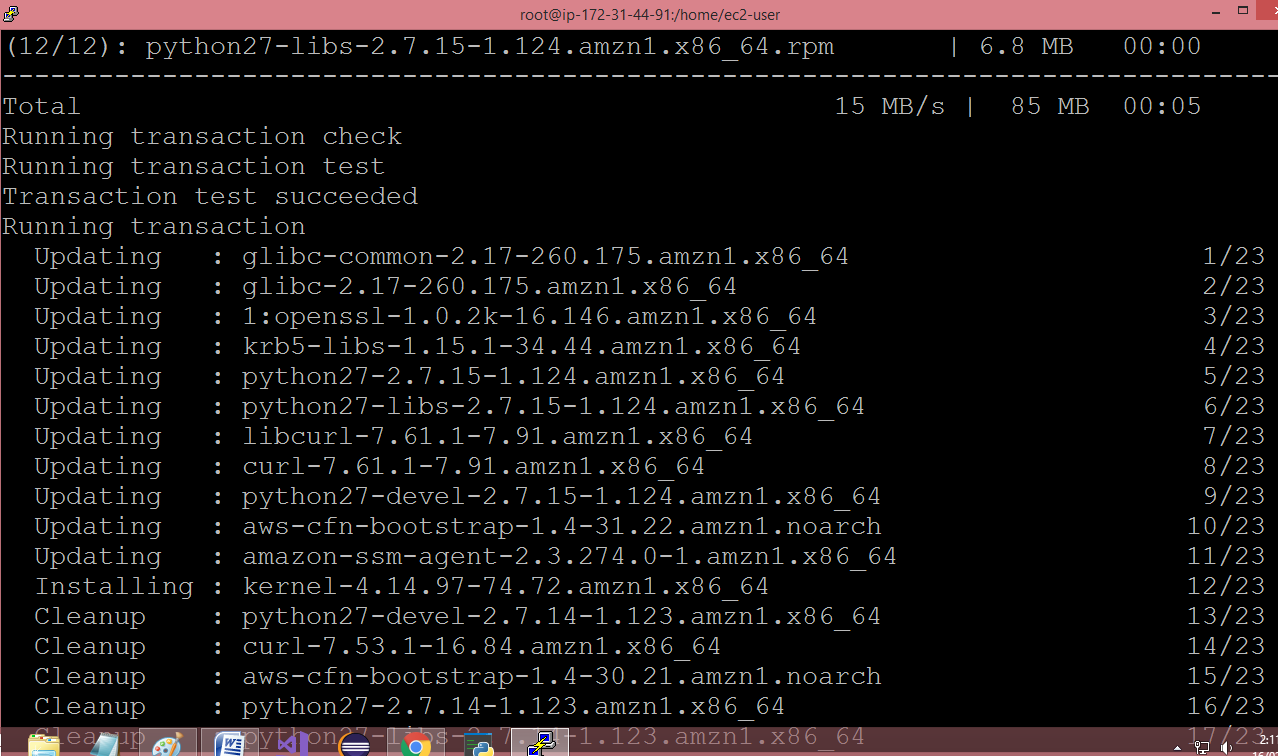


* Click on the Open button to open the Putty window.



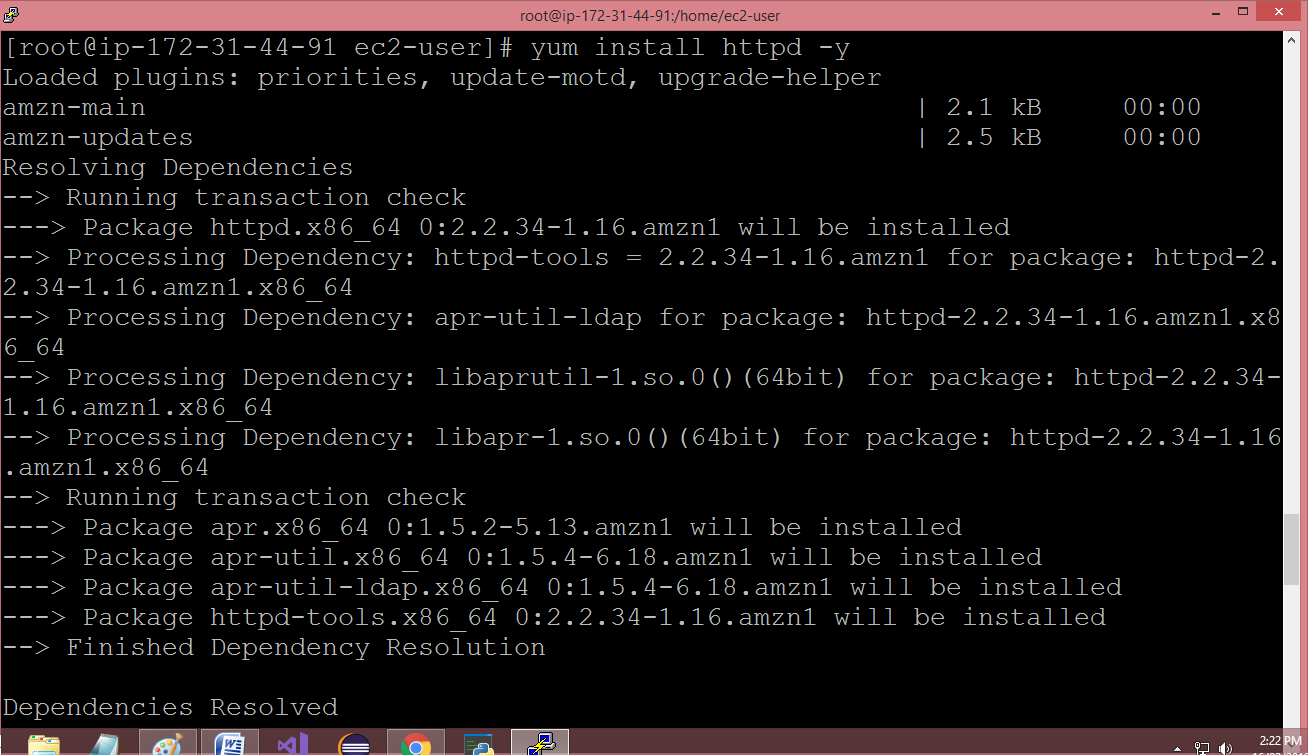
The above screen shows that we are connected to the EC2 instance.

* Run the command **sudo su,** and then run the command **yum update** -y to update the EC2 instance.



Note: sudo su is a command which is used to provide the privileges to the root level.

* Now, we install the apache web server to ensure that an EC2 instance becomes a web server by running a command **yum install httpd** -y.



* Run the command **cd /var/www/html.**

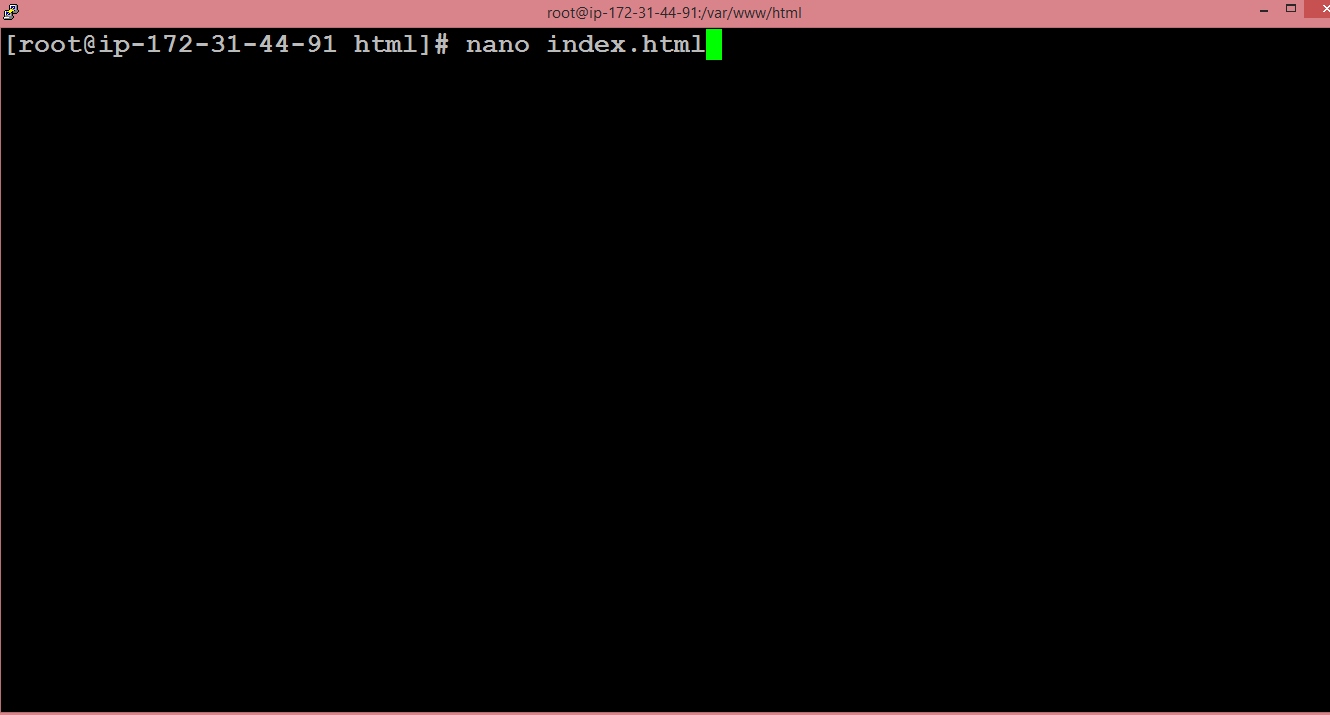


* To list the files available in the html folder, run the command **ls.**

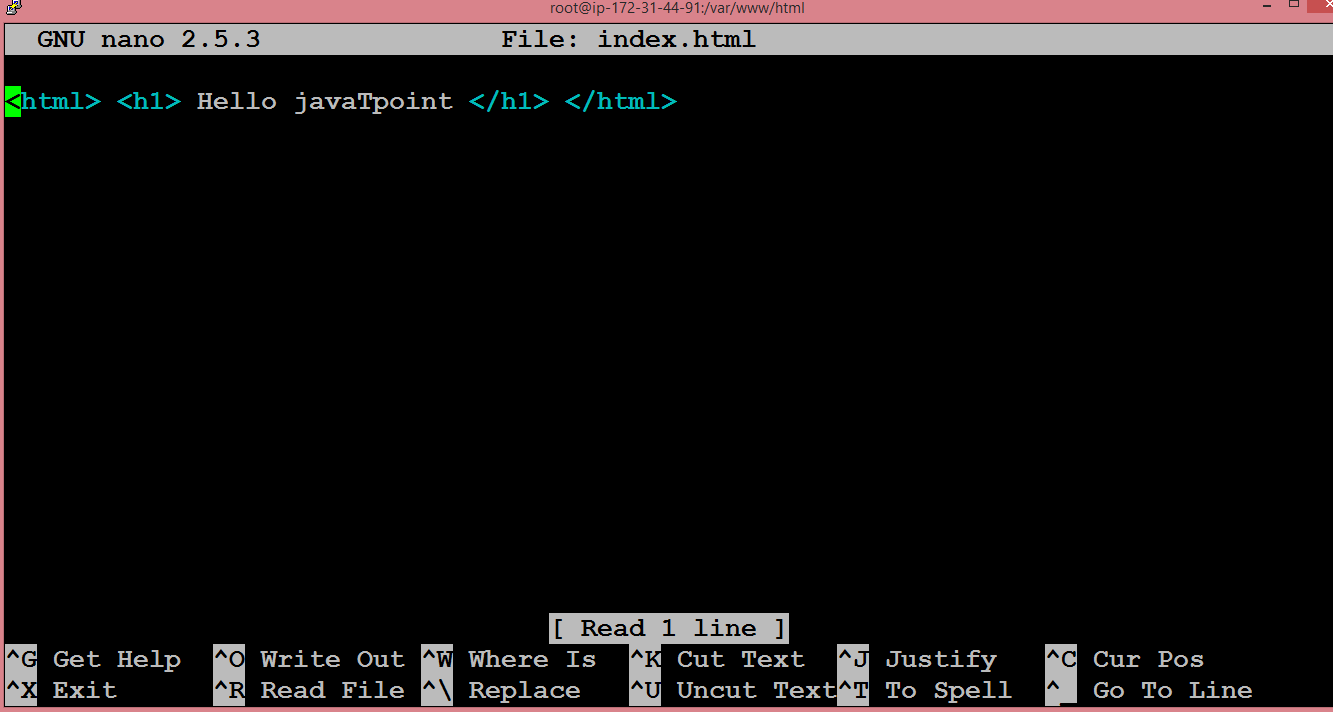


We observe that on running the command **ls,** we do not get any output. It means that it does not contain any file.

* We create a text editor, and the text editor is created by running the command **nano index.html** where index.html is the name of the web page.

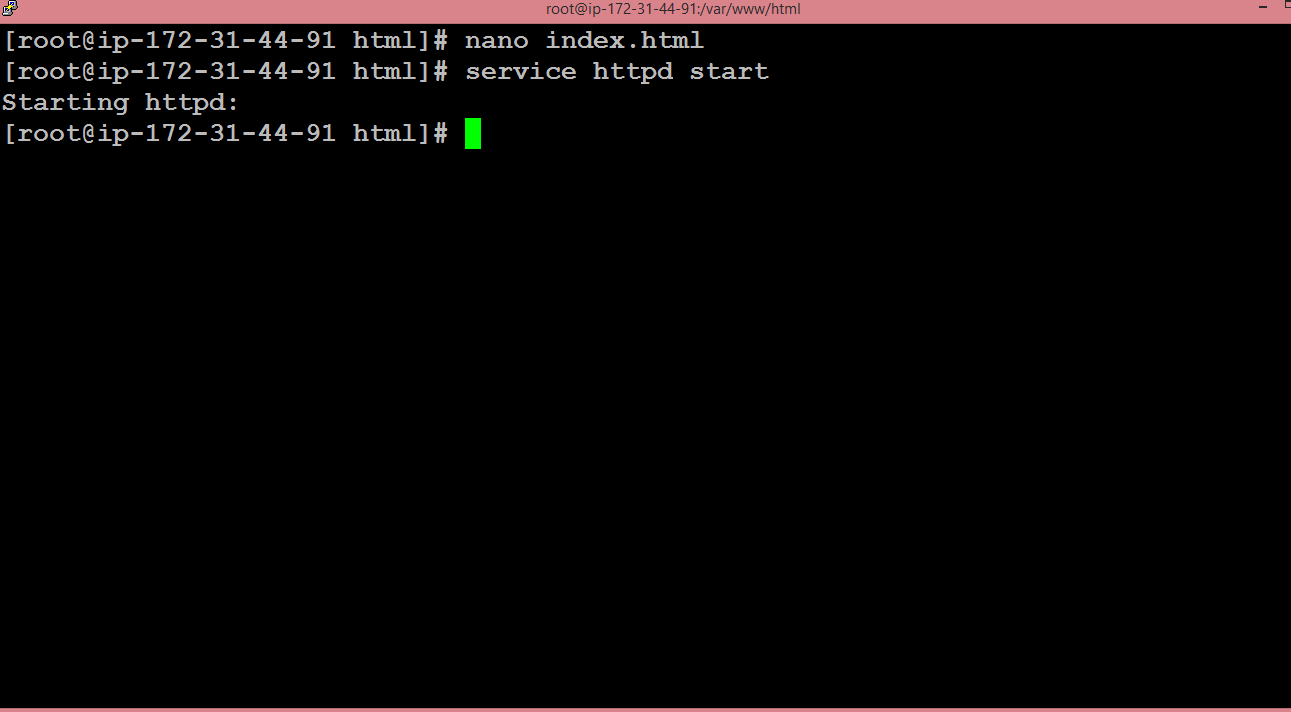


* The text editor is shown below where we write the HTML code.



After writing the HTML code, press ctrl+X to exit, then press 'Y' to save the page and press Enter.

* Start the Apache server by running the command **service httpd start.**

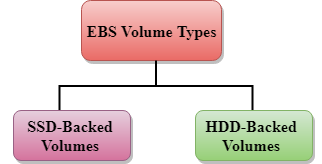


* Go to the web browser and paste the IP address which is used to connect to your EC2 instance. You will see the web page that you have created.

What is EBS?

* EBS stands for **Elastic Block Store**.
* EC2 is a virtual server in a cloud while EBS is a virtual disk in a cloud.
* Amazon EBS allows you to create storage volumes and attach them to the EC2 instances.
* Once the storage volume is created, you can create a file system on the top of these volumes, and then you can run a database, store the files, applications or you can even use them as a block device in some other way.
* Amazon EBS volumes are placed in a specific availability zone, and they are automatically replicated to protect you from the failure of a single component.
* EBS volume does not exist on one disk, it spreads across the Availability Zone. EBS volume is a disk which is attached to an EC2 instance.
* EBS volume attached to the EC2 instance where windows or Linux is installed known as Root device of volume.

EBS Volume Types

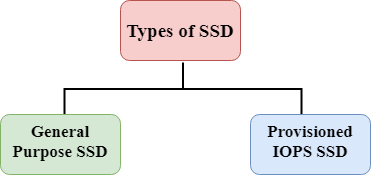


Amazon EBS provides two types of volume that differ in performance characteristics and price. EBS Volume types fall into two parts:

* SSD-backed volumes
* HDD-backed volumes

SSD

* SSD stands for solid-state Drives.
* In June 2014, SSD storage was introduced.
* It is a general purpose storage.
* It supports up to 4000 IOPS which is quite very high.
* SSD storage is very high performing, but it is quite expensive as compared to HDD (Hard Disk Drive) storage.
* SSD volume types are optimized for transactional workloads such as frequent read/write operations with small I/O size, where the performance attribute is IOPS.



**SSD is further classified into two parts:**

* General Purpose SSD
* Provisioned IOPS SSD

General Purpose SSD

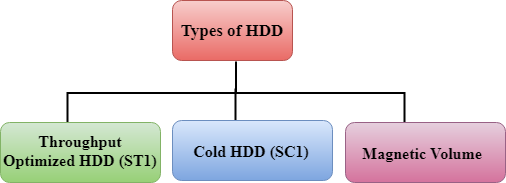
* General Purpose SSD is also sometimes referred to as a GP2.
* It is a General purpose SSD volume that balances both price and performance.
* You can get a ratio of 3 IOPS per GB with up to 10,000 IOPS and the ability to burst up to 3000 IOPS for an extended period of time for volumes at 3334 GiB and above. For example, if you get less than 10,000 IOPS, then GP2 is preferable as it gives you the best performance and price.

Provisioned IOPS SSD

* It is also referred to as IO1.
* It is mainly used for high-performance applications such as intense applications, relational databases.
* It is designed for I/O intensive applications such as large relational or NOSQL databases.
* It is used when you require more than 10,000 IOPS.

HDD

* It stands for Hard Disk Drive.
* HDD based storage was introduced in 2008.
* The size of the HDD based storage could be between 1 GB to 1TB.
* It can support up to 100 IOPS which is very low.



Throughput Optimized HDD (st1)

* It is also referred to as ST1.
* Throughput Optimized HDD is a low-cost HDD designed for those applications that require higher throughput up to 500 MB/s.
* It is useful for those applications that require the data to be frequently accessed.
* It is used for Big data, Data warehouses, Log processing, etc.
* It cannot be a boot volume, so it contains some additional volume. For example, if we have Windows server installed in a C: drive, then C drive cannot be a Throughput Optimized Hard disk, D: drive or some other drive could be a Throughput Optimized Hard disk.
* The size of the Throughput Hard disk can be 500 GiB to 16 TiB.
* It supports up to 500 IOPS.

Cold HDD (sc1)

* It is also known as SC1.
* It is the lowest cost storage designed for the applications where the workloads are infrequently accessed.
* It is useful when data is rarely accessed.
* It is mainly used for a File server.
* It cannot be a boot volume.
* The size of the Cold Hard disk can be 500 GiB to 16 TiB.
* It supports up to 250 IOPS.

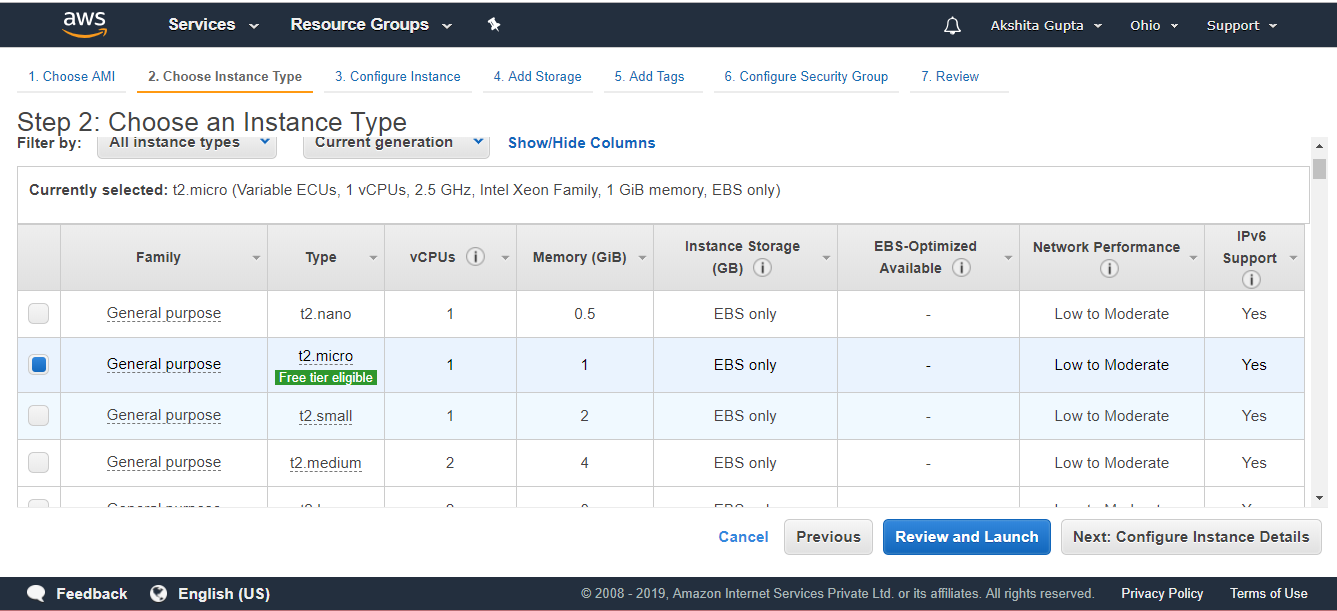
Magnetic Volume

* It is the lowest cost storage per gigabyte of all EBS volume types.
* It is ideal for the applications where the data is accessed infrequently
* It is useful for applications where the lowest storage cost is important.
* Magnetic volume is the only hard disk which is bootable. Therefore, we can say that it can be used as a boot volume.

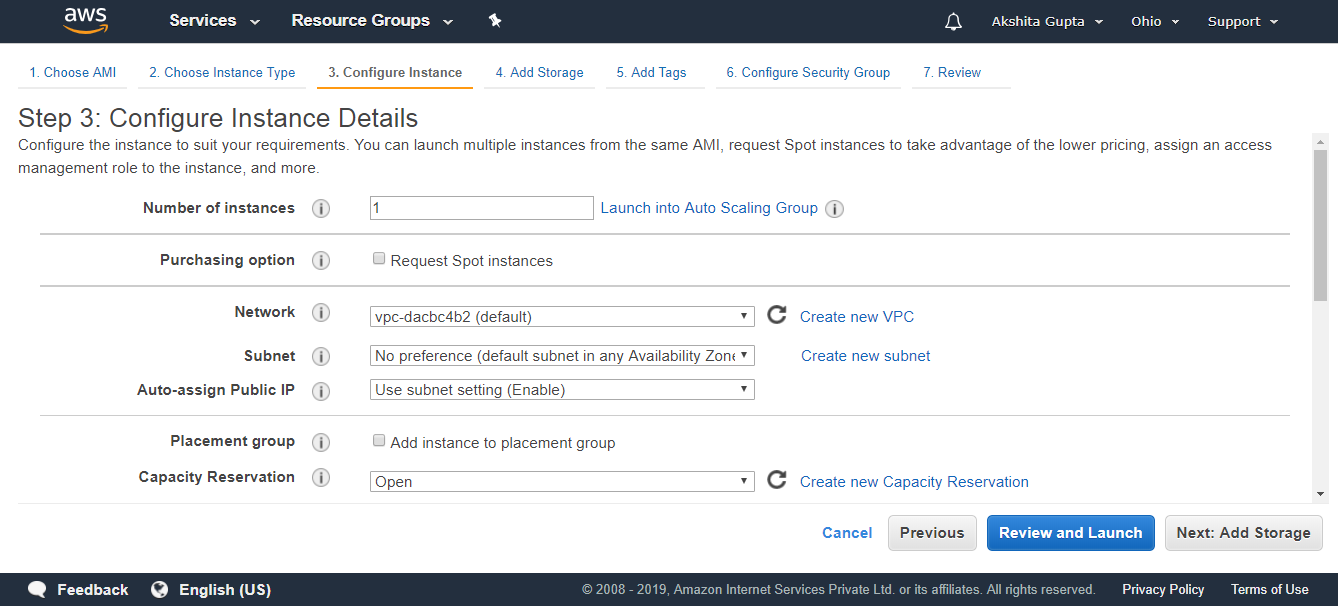
EBS Volume

In this topic, we will learn how to upgrade the EBS Volume in various ways.

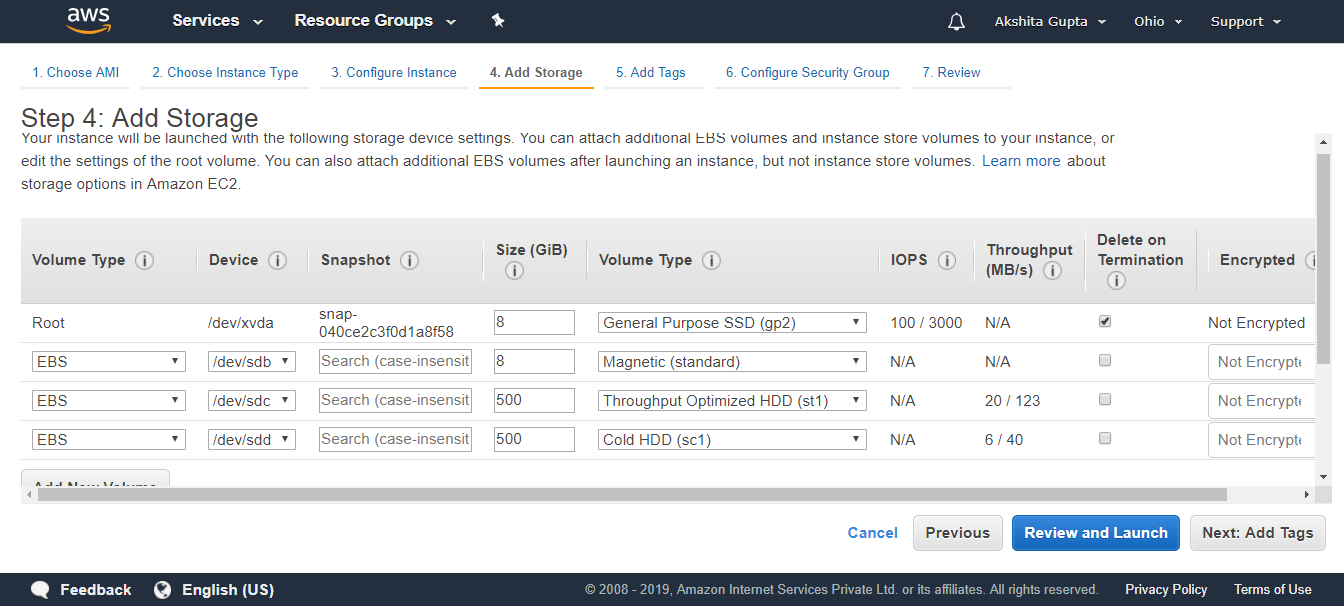
* Sign in to the AWS Management Console.
* Move to the EC2 service.
* Create an EC2 instance.
* Choose an Amazon Machine Image (AMI).
* Choose an Instance type. Suppose I choose a **t2.micro** type for our EC2 instance. Click on the **Next.**



* Configure Instance details. Keep all the details as default, and then click on the **Next.**

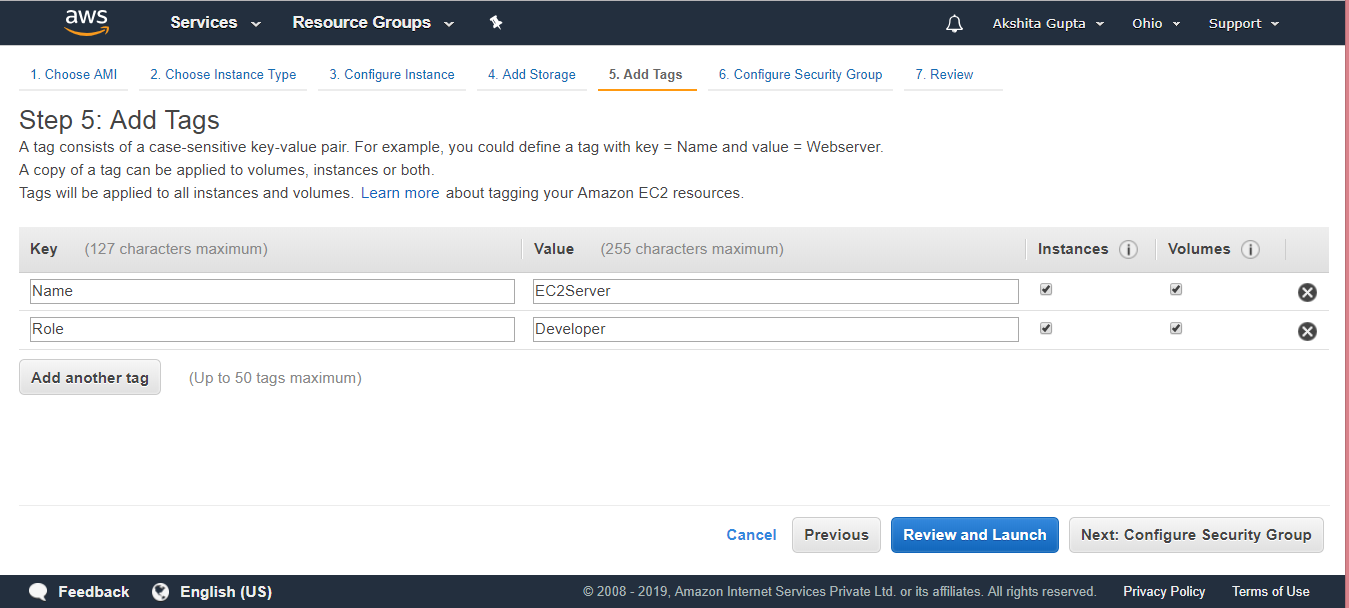


* Add the EBS Volume. **Root** is the default EBS volume used to store the operating system.

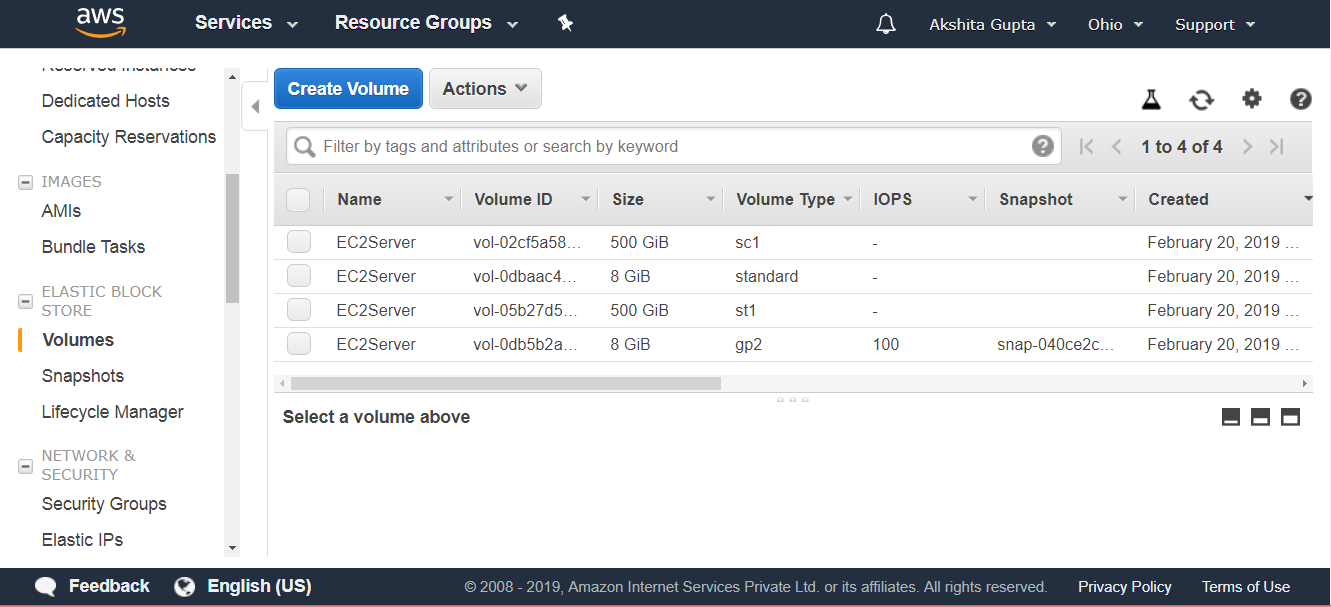


I create three EBS Volumes of type **Magnetic, Throughput Optimized HDD and Cold HDD.**

* Add tags.

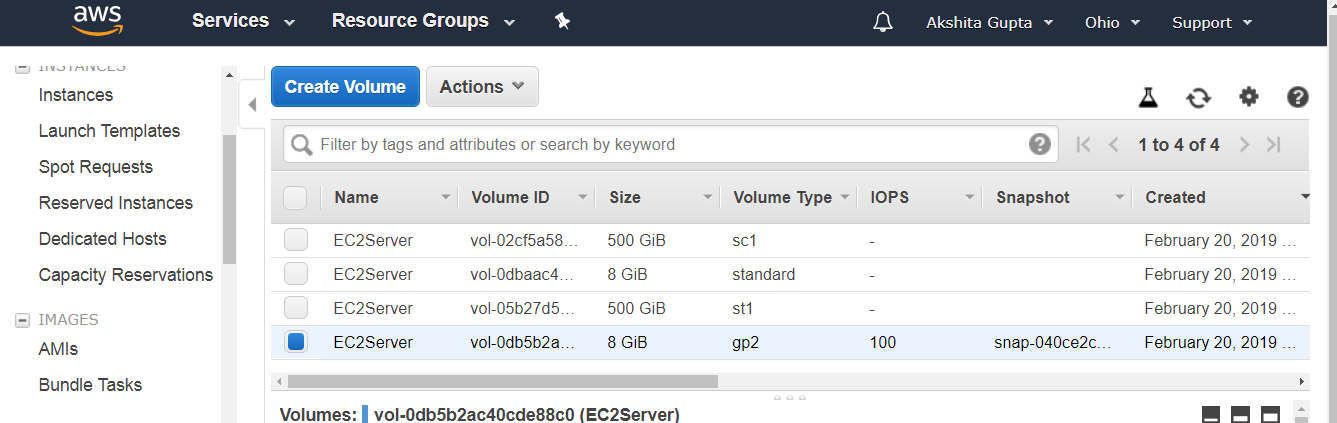


* Configure Security Group. Click on the **Review and Launch.**
* Click on the **Launch** button.
* Move to the Volumes appearing on the left side of the console.

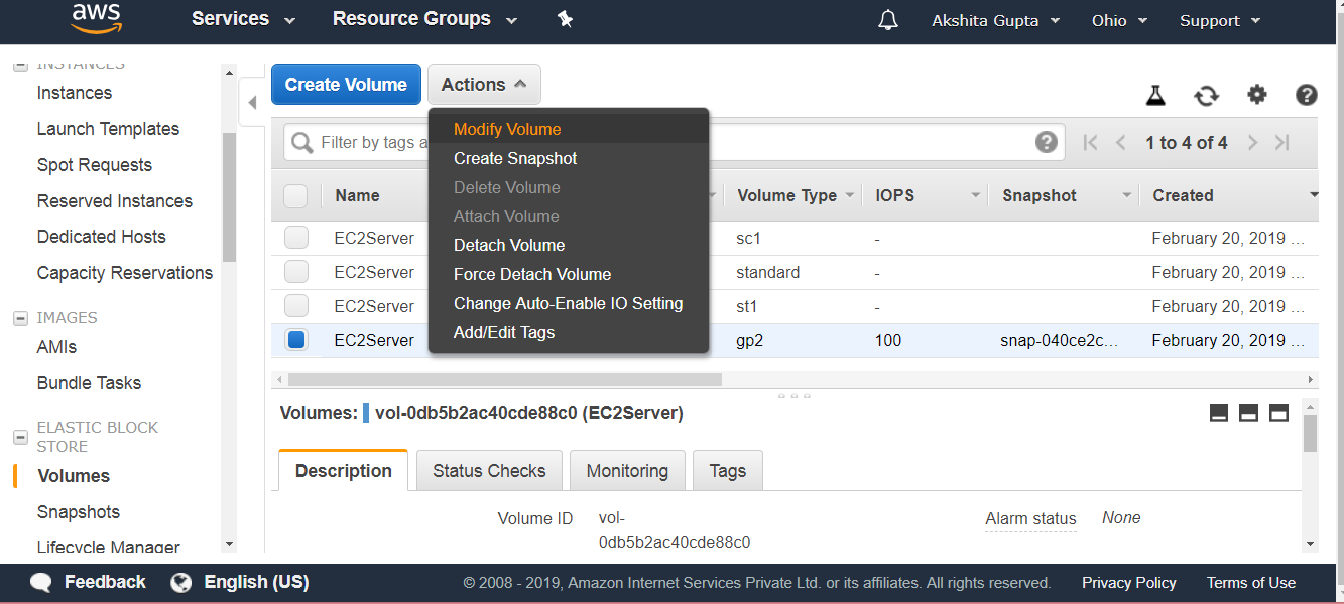
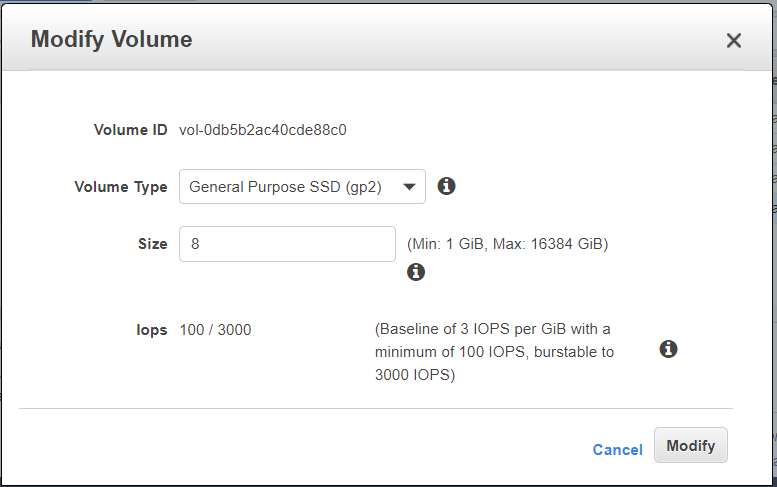


From the above screen, we observe that a single EC2 instance is associated with four EBS Volumes type.

* We can also modify the volume. Suppose I want to modify the volume type of **gp2,** check the EBS Volume that you want to modify.

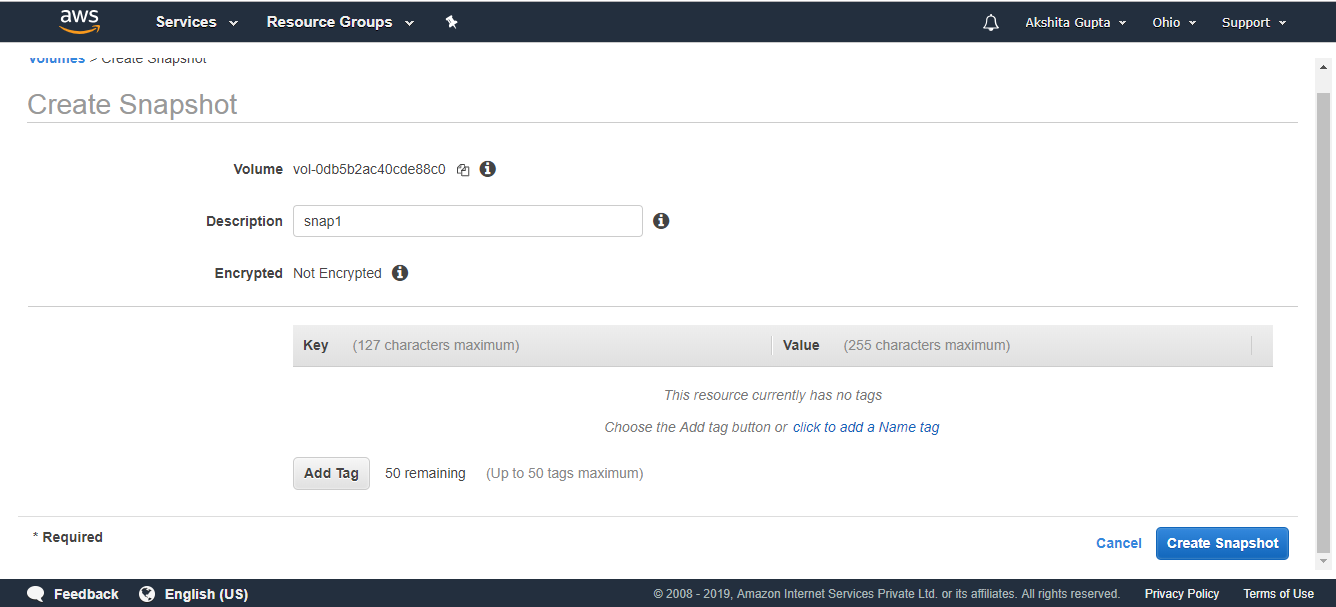


* Click on the **Actions** dropdown menu, click on the **Modify Volume.**

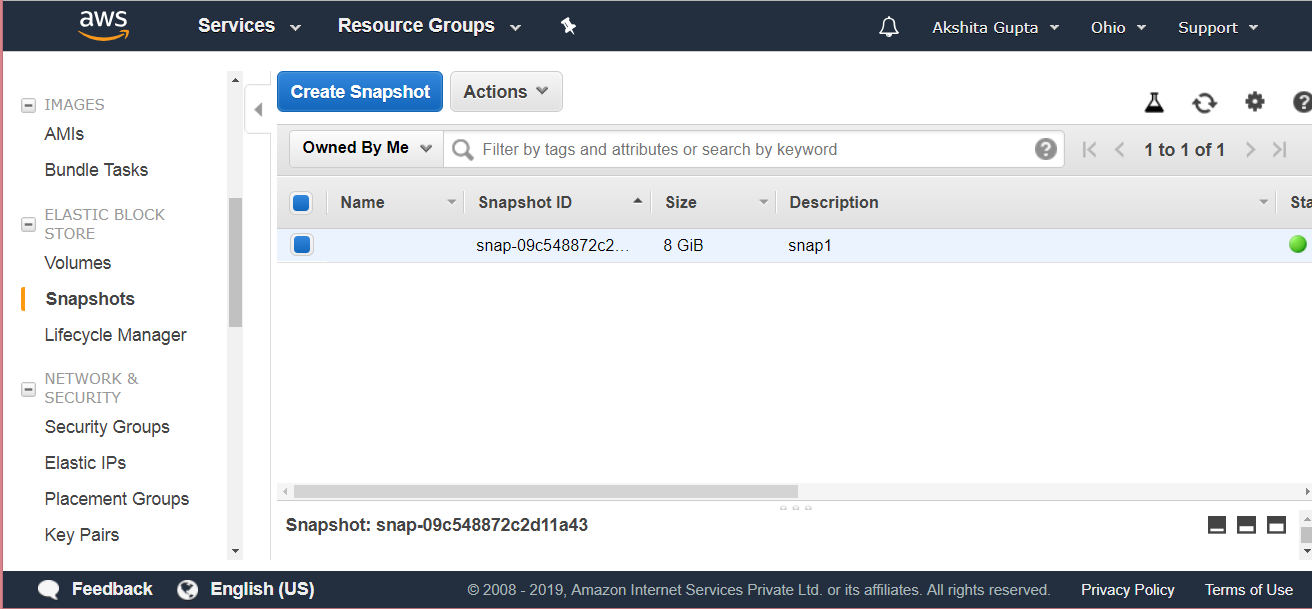
  


From the above screen, we observe that we can also modify the volume type. Standard cannot be modified as it is the oldest storage available. Therefore, we can say that all the volumes are modifiable apart from the magnetic storage volume.

* Suppose I want to create another EBS Volume or EC2 instance in another Availability Zone. In order to achieve this, we first need to create a snapshot. Move to the **Actions** dropdown menu, then click on the **Create Snapshot.**

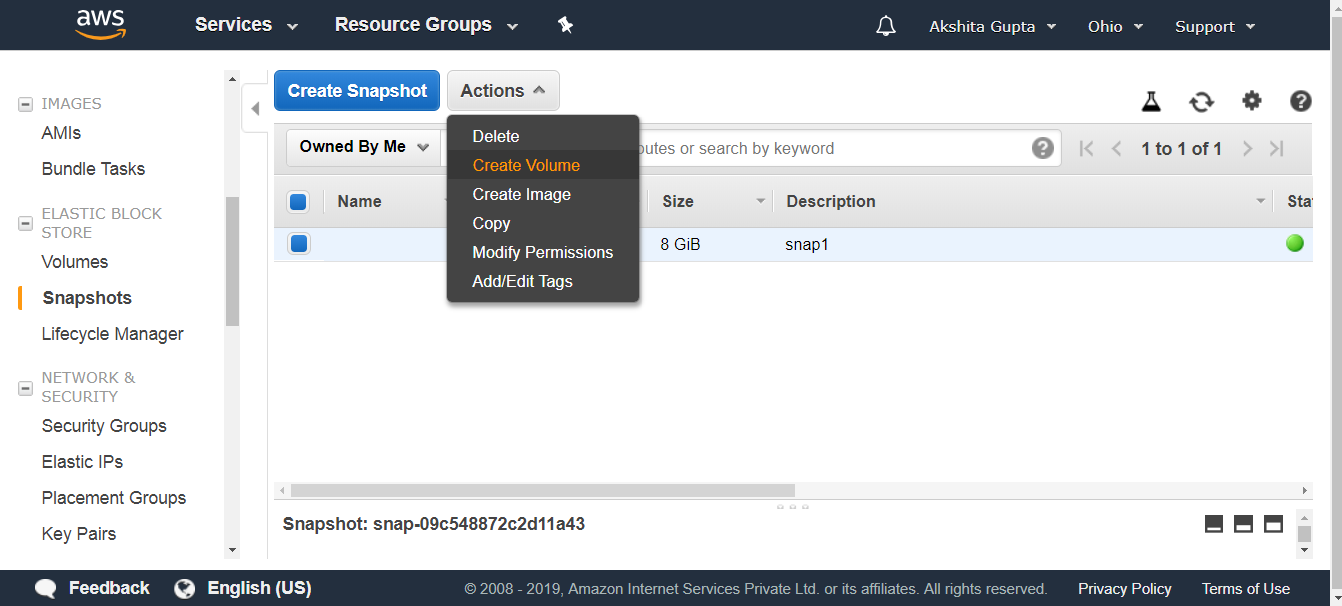
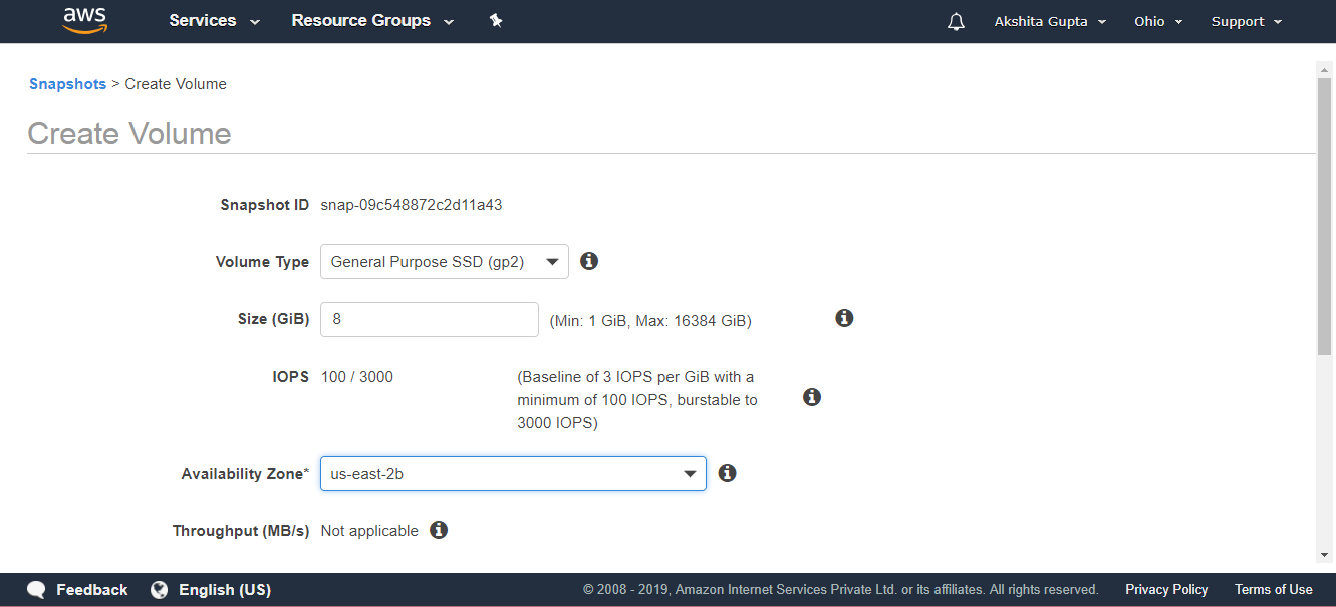


Move to the snapshot appearing on the left side of the console.



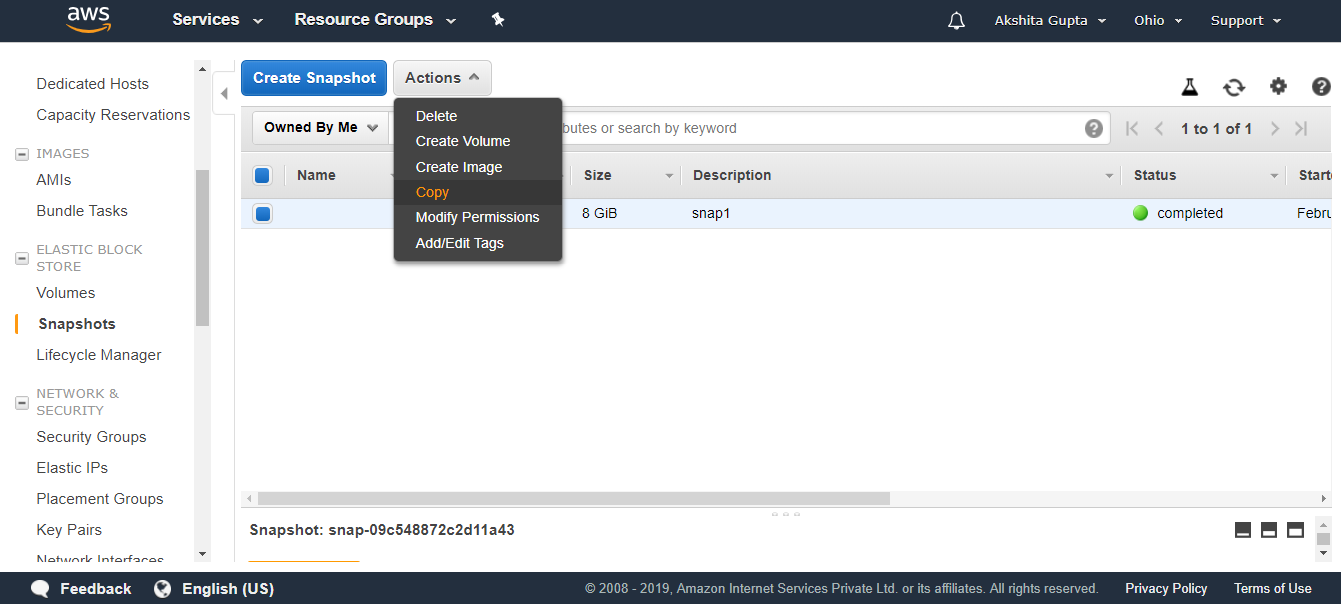
The above screen shows the snapshot that you have created just now.

Move to the **Actions** dropdown menu, click on the **Create Volume.**

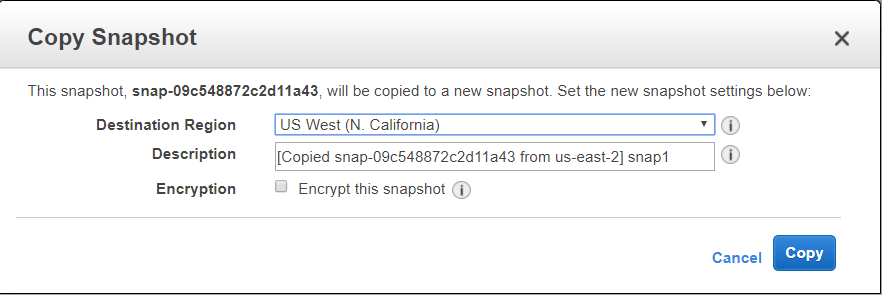
  


From the above screen, we observe that we can change the Availability zone of a Volume for a snapshot that we have just created. Therefore, we have created a new EBS Volume with different Availability zone, **i.e., us-east-2b.**

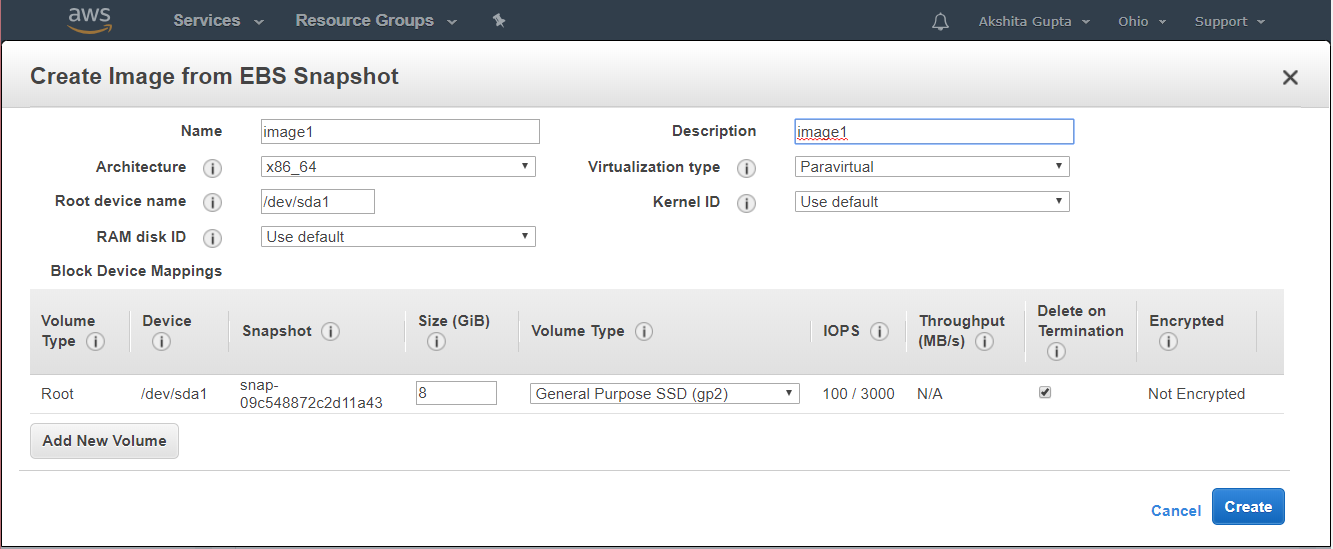
* You can also move an EC2 instance from one region to another region. In order to achieve this, the following steps are to be taken:
  + Create a Snapshot.
  + Move to the **Actions** dropdown menu, click on the copy.



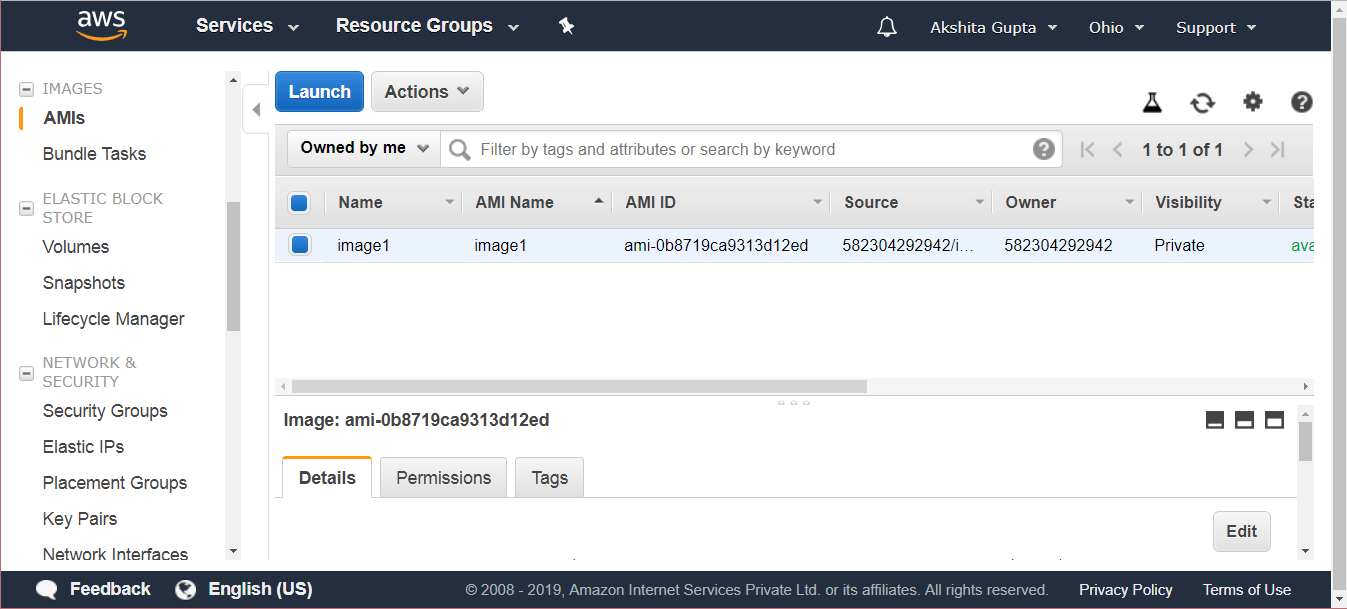
* The screen shown below shows that the snapshot has been copied to a new region, i.e., US West.



* Once the snapshot is in a different region, create the image of the snapshot by clicking on the **Actions** dropdown menu, and then click on the **Create Image.**



* Click on the **Create** button.
* Move to the AMI appearing on the left side of the console. An AMI displays the image that you have just created.



* Click on the **Launch**button.

Note: If you delete an EC2 instance, then root device is deleted, and other EBS Volumes are not deleted.

Important points to be noted:

* Volumes exist on EBS. Volume is a Virtual Hard Disk which stores the operating system.
* Snapshots exist on S3. However if you look at a bucket and searches for the snapshots in a bucket, you will not be able to see the snapshots. Therefore, we can say that the snapshot has no visibility.
* Snapshots are a point in time copies of volumes. It's just like taking a photograph of the hard disk at a point of time.
* Snapshots are incremental means the blocks that have been changed since your last snapshot are moved to S3.
* The first snapshot always takes some time to create.
* To create a snapshot for Amazon EBS Volumes that serve as root devices, you should stop the instance before taking the snapshot.
* However, you can take a snapshot while the instance is running.
* You can create AMI from snapshots.
* You can change the EBS Volume size and storage type.
* The Volume will always be in the same availability zone as the EC2 instance.
* To move an EBS Volume from one region to another, take a snapshot of it, and then copy it to the new region.