**Create an Amazon EC2 Instance**



An **EC2 instance** is a virtual server in Amazon Web services terminology.EC2 is a web service where an AWS subscriber can request and provision a compute server in AWS cloud.

An **on-demand** EC2 instance is an offering from AWS where the subscriber/user can rent the virtual server per hour and use it to deploy his/her own applications.

The instance will be charged per hour with different rates based on the type of the instance chosen. AWS provides multiple instance types for the respective business needs of the user.

You can rent an EC2 instance what do you want and terminate it whenever you want.Your choose.P ay as much as you use.You will learn this tutorial;

Login and access to AWS services

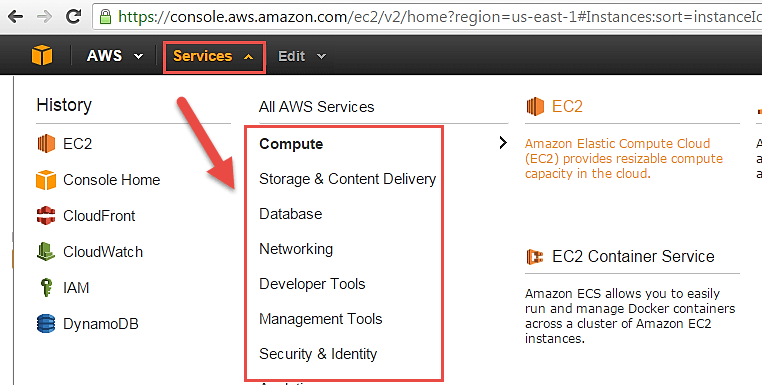
1. Choose AMI
2. Choose Instance Types
3. Configure Instance
4. Add Storage
5. Tag Instance
6. Configuring Security Groups
7. Review Instances
8. Creating a EIP and connecting to your instance.

Login and access to AWS services

**Step 1)** In this step,

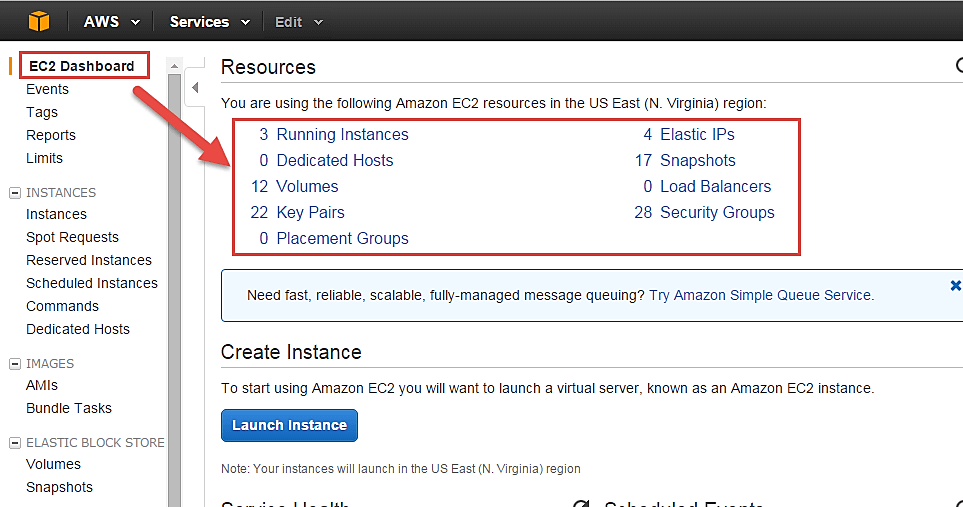
Login to your AWS account and go to the AWS Services tab at the top left corner.

You will see all of the AWS Services categorized as per their area viz. Compute, Storage, Database, etc. For creating an EC2 instance, we have to choose Compute EC2 as in the next step.


     An Amazon EBS-backed instance with an additional Amazon Elastic Block Store (EBS) volume
    

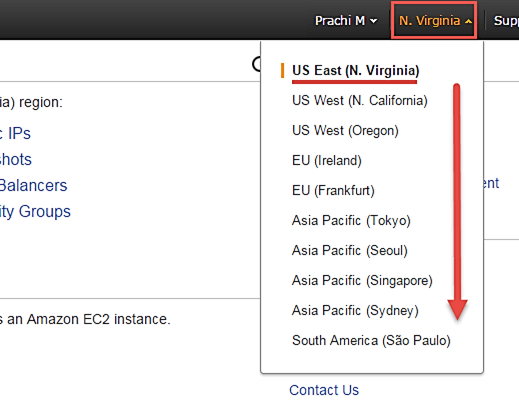
Open all the services and click on EC2 under Compute services. This will launch the dashboard of EC2.

Here is the EC2 dashboard. Here you will get all the information in gist about the AWS EC2 resources running.



**Step 2)** On the top right corner of the EC2 dashboard, choose the AWS Region in which you want to provision the EC2 server.

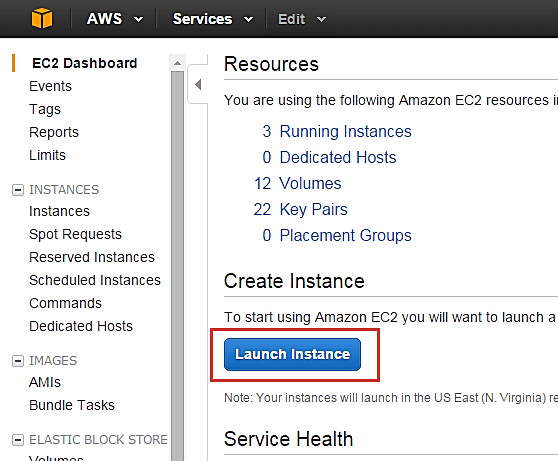
Here we are selecting N. Virginia. AWS provides 10 Regions all over the globe.Its increasing  over time.Because Amazon opens new regions all over the world day after day.



**Step 3)** In this step

Once your desired Region is selected, come back to the EC2 Dashboard.

Click on ‘Launch Instance’ button in the section of Create Instance (as shown below).



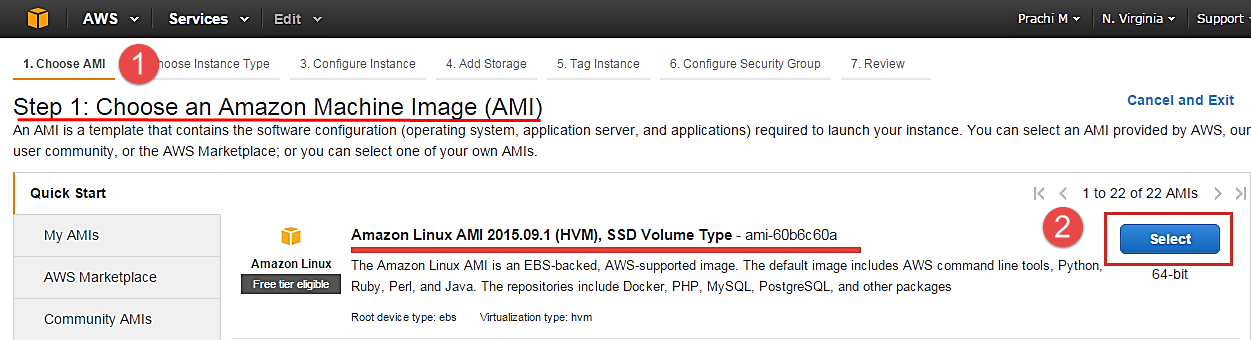
Instance creation wizard page will open as soon as you click ‘Launch Instance’.

Choose AMI

**Step 1)** In this step we will do,

An AMI is an Amazon Machine Image. It is a template basically of an Operating System platform which you can use as a base to create your instance.In Amazon web services has got a lot of ready AMI.You can choose what do you want.(Redhat,Centos,Windows etc.)

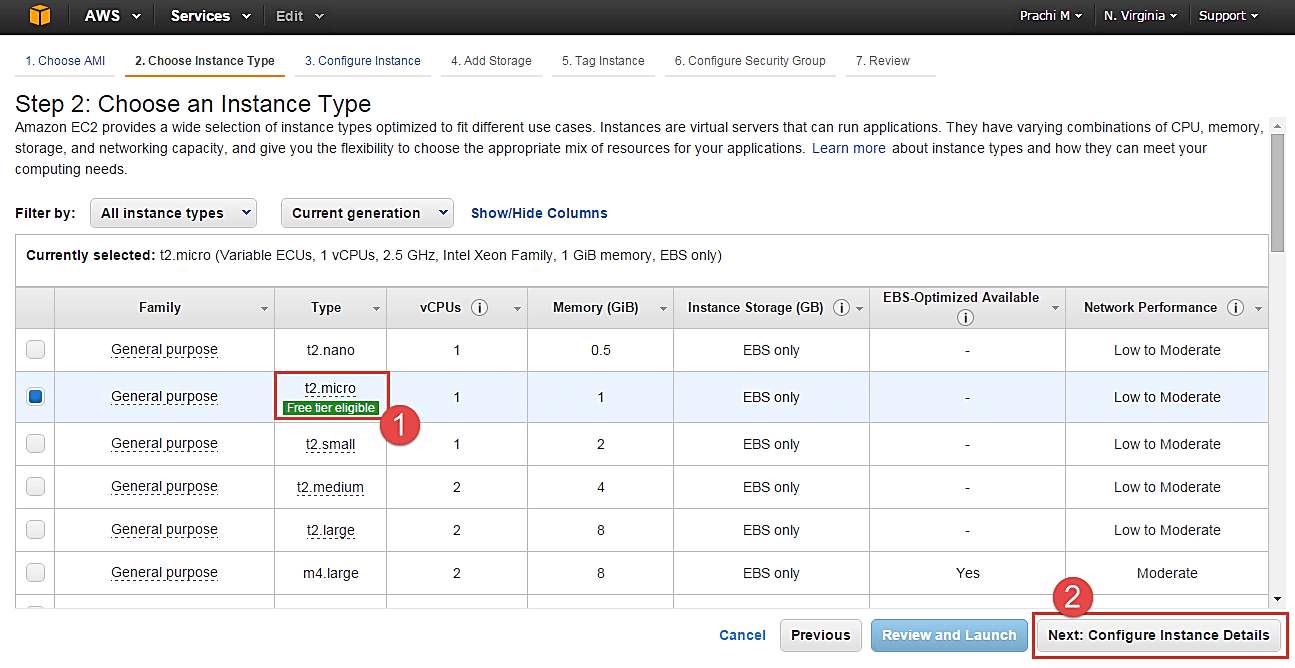
Here we are choosing the default Amazon Linux (64 bit) AMI.



Choose Instance Types

In the next step, you have to choose the type of instance you require based on your business needs.

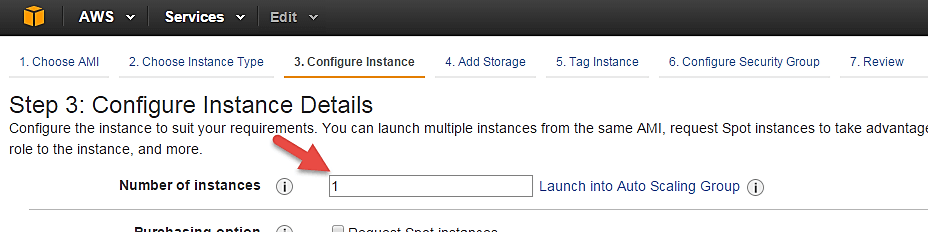
We will choose t2.micro instance type, because its free, which is a 1vCPU and 1GB memory server offered by AWS.AWS Free Tier includes 750 hours of Linux and Windows t2.micro instances each month for one year. To stay within the Free Tier, use only EC2 Micro instances.

Click on “Configure Instance Details” for further configurations. 

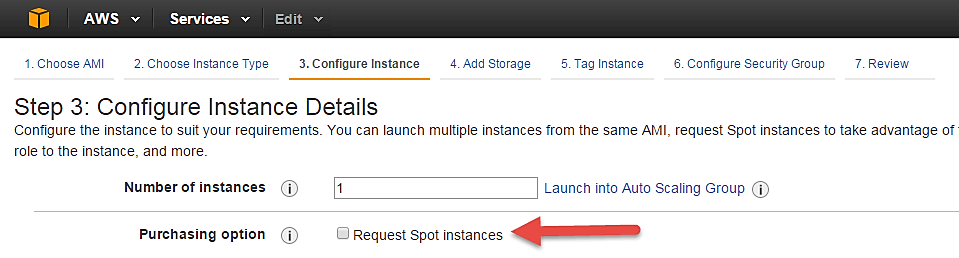
Next step,We will configure this instance details.

Configure Instance

**Step 1)** No. of instances- you can provision up to 20 instances at a time. Here we are launching one instance. Whatever you want.



**Step 2)** Under Purchasing Options, keep the option of ‘Request Spot Instances’ unchecked as of now. (This is done when we wish to launch Spot instances instead of on-demand ones. We will come back to Spot instances in the later part of the tutorial).



**Step 3)** Next, we have to configure some basic networking details for our EC2 server.

In this steps we configure VPC(Virtual Private Cloud) Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

You have to decide here, in which VPC (Virtual Private Cloud) you want to launch your instance and under which subnets inside your VPC. It is better to determine and plan this prior to launching the instance. Your AWS architecture set-up should include IP ranges for your subnets etc. pre-planned for better management. (We will see how to create a new VPC in Networking section of the tutorial.

Subnetting should also be pre-planned. E.g.: If it’s a web server you should place it in the public subnet and if it’s a DB server, you should place it in a private subnet all inside your VPC.

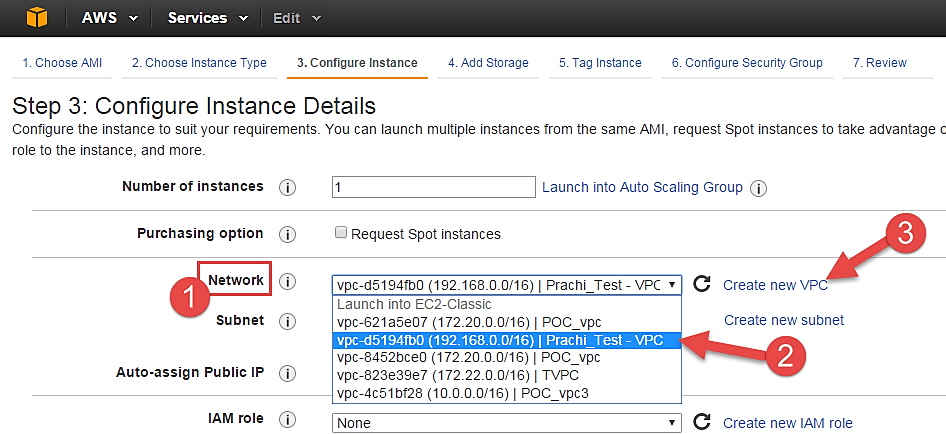
Below,

Network section will give a list of VPCs available in our platform.

Select an already existing VPC

You can also create a new VPC

Here I have selected an already existing VPC where I want to launch my instance.



*Virtual Private Cloud*

**Step 4)** In this step,

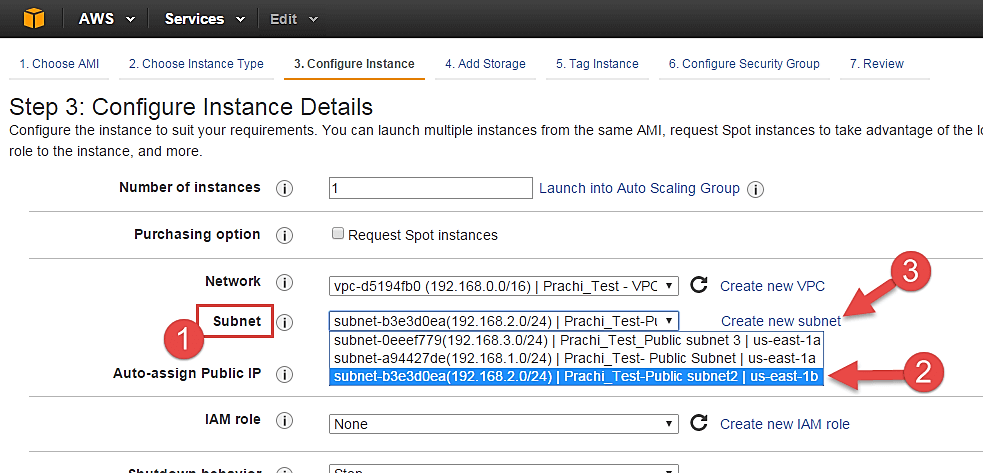
A VPC consists of subnets, which are IP ranges that are separated for restricting access.

Below,

Under Subnets, you can choose the subnet where you want to place your instance.

I have chosen an already existing public subnet.

You can also create a new subnet in this step.



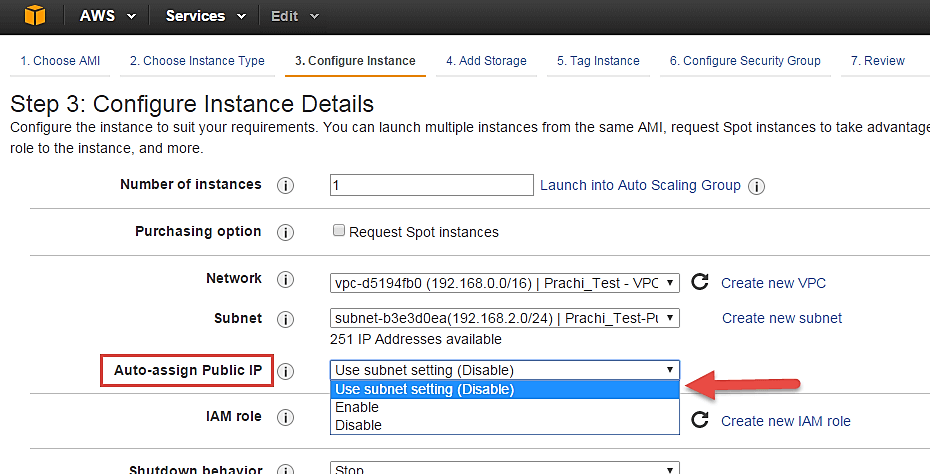
*choosing subnet*

Once your instance is launched in a public subnet, AWS will assign a dynamic public IP to it from their pool of IPs.

**Step 5)** In this step,

You can choose if you want AWS to assign it an IP automatically, or you want to do it manually later. You can enable/ disable ‘Auto assign Public IP’ feature here likewise.

Here we are going to assign this instance a static IP called as EIP (Elastic IP) later. So we keep this feature disabled as of now.

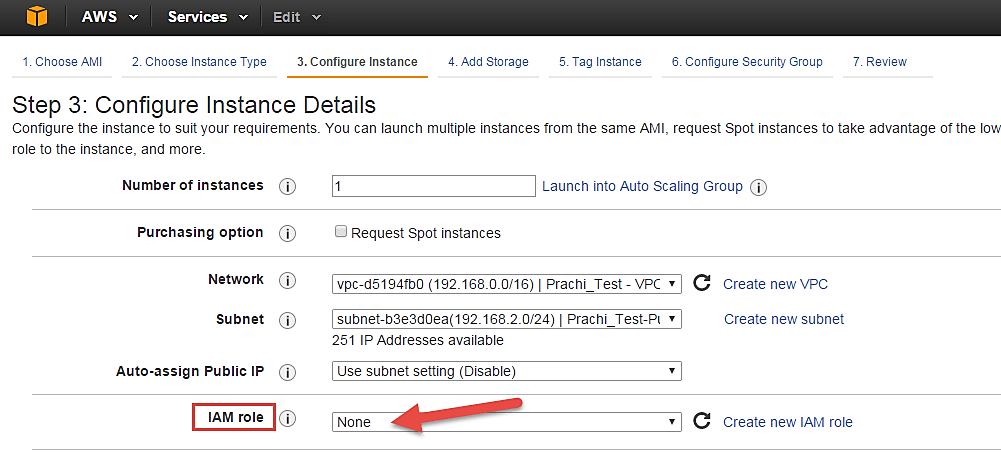


*assign public I*

**Step 6)** In this step,

In the following step, keep the option of IAM role ‘None’ as of now. We will visit the topic of IAM role in detail in IAM services.

An IAM *role* is similar to a user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials (password or access keys) associated with it. Instead, if a user assumes a role, temporary security credentials are created dynamically and provided to the user.

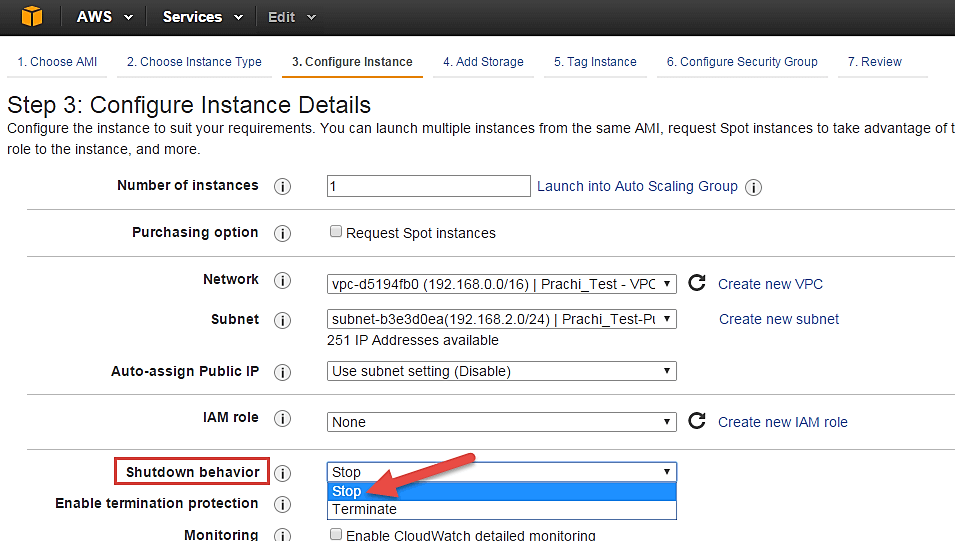


*IAM role*

**Step 7)** In this step, you have to do following things

Shutdown Behavior – when you accidently shut down your instance, you surely don’t want it to be deleted but stopped.

Here we are defining my shutdown behavior as Stop.

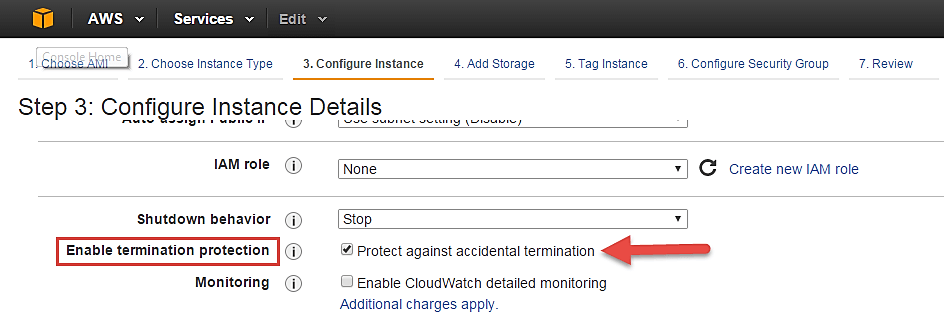


*shutdown behavior ec2*

**Step 8)** In this step,

In case, you have accidently terminated your instance, AWS has a layer of security mechanism. It will not delete your instance if you have enabled accidental termination protection.

Here we are checking the option for further protecting our instance from accidental termination.

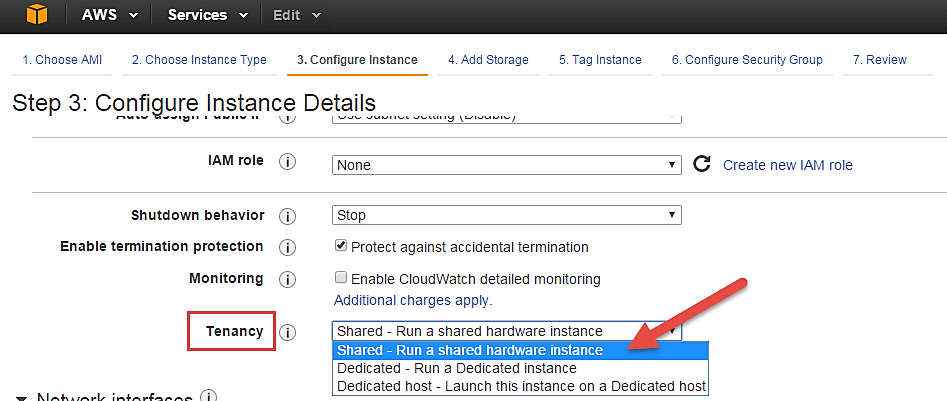


*termination protection*

**Step 9)** In this step,

Under Monitoring- you can enable Detailed Monitoring if your instance is a business critical instance. Here we have kept the option unchecked. AWS will always provide Basic monitoring on your instance free of cost. We will visit the topic of monitoring in AWS Cloud Watch part of the tutorial.

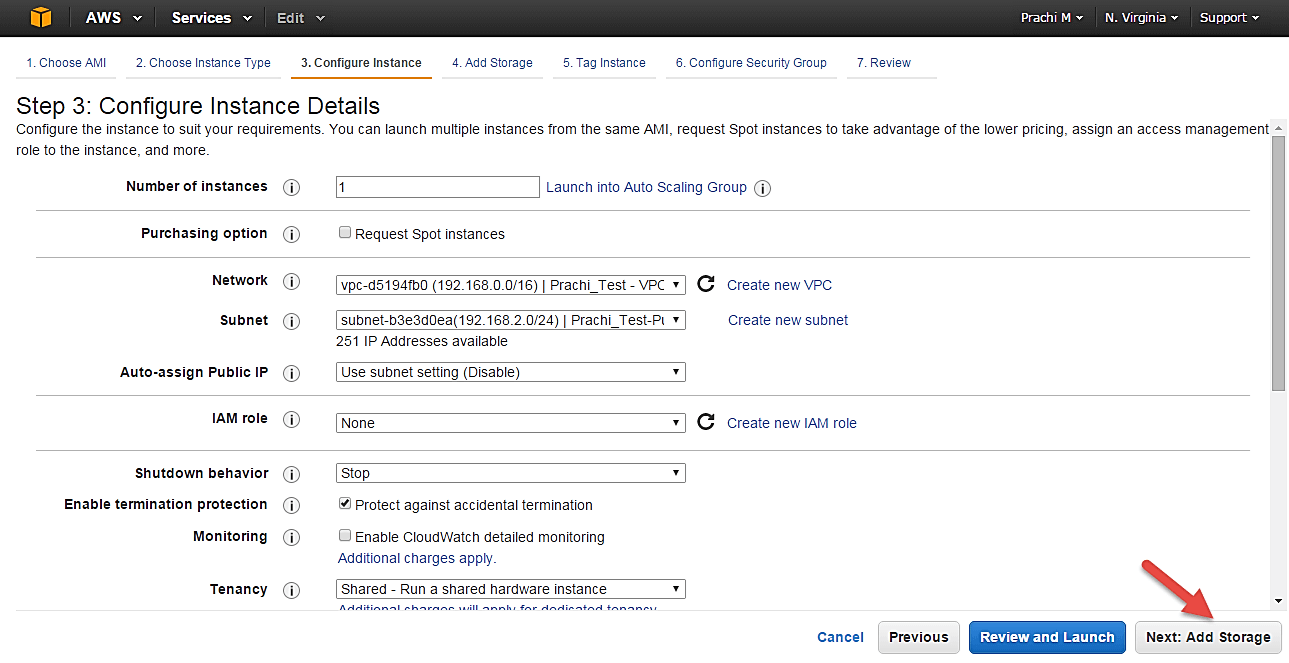
Under Tenancy- select the option if shared tenancy. If your application is a highly secure application, then you should go for dedicated capacity. AWS provides both options.



*tenancy*

**Step 10)** In this step,

Click on ‘Add Storage’ to add data volumes to your instance in next step.



*Add Storage*

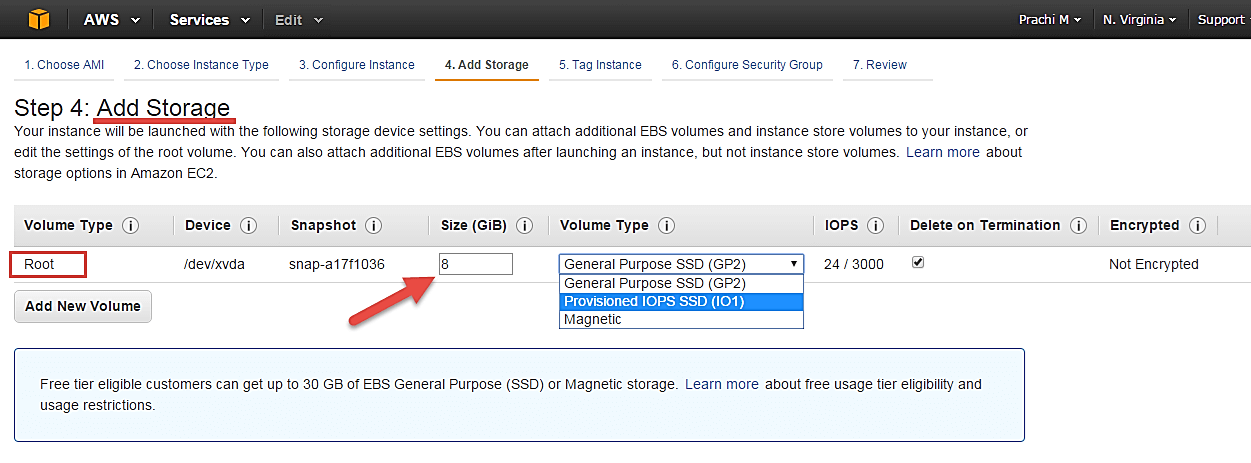
Add Storage

**Step 1)** In this step we do following things,

In the Add Storage step, you’ll see that the instance has been automatically provisioned a General Purpose SSD root volume of 8GB. ( Maximum volume size we can give to a General Purpose volume is 16GB)

You can change your volume size, add new volumes, change the volume type, etc.

AWS provides 3 types of EBS volumes- Magnetic, General Purpose SSD, Provisioned IOPs. You can choose a volume type based on your application’s IOPs needs.



Tag Instance

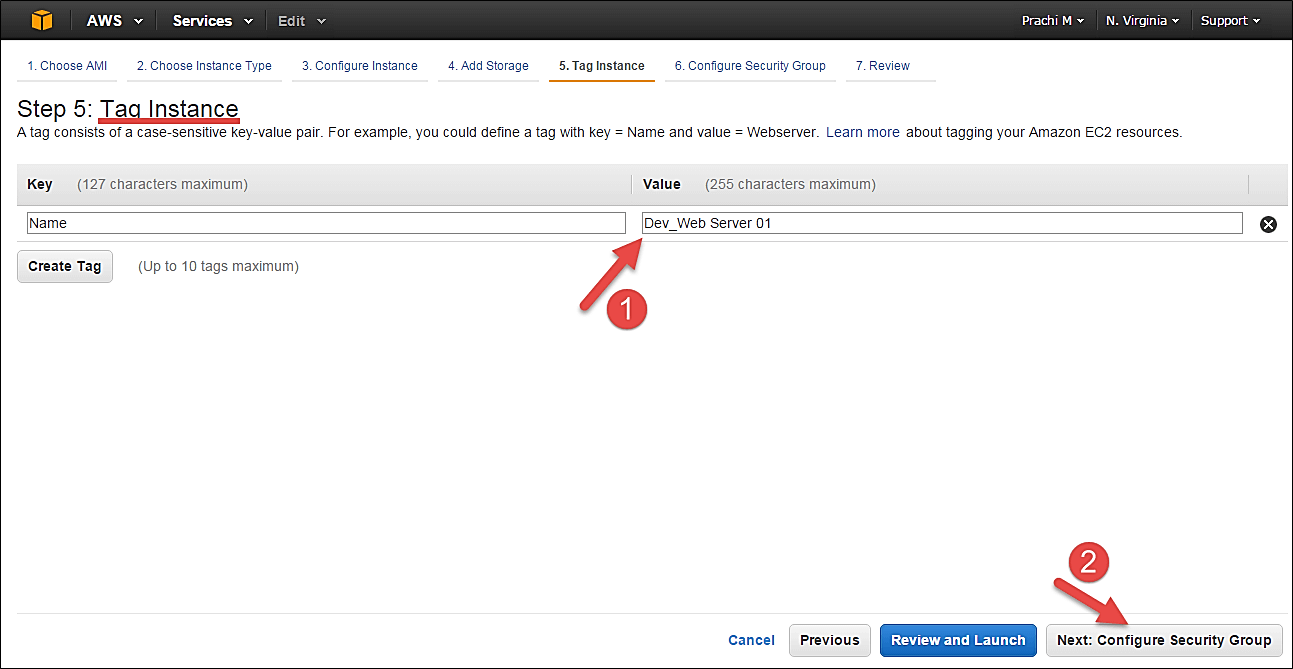
**Step 1)** In this step

you can tag your instance with a key-value pair. This gives visibility to the AWS account administrator when there are lot number of instances.

The instances should be tagged based on their department, environment like Dev/SIT/Prod. Etc. this gives a clear view of the costing on the instances under one common tag.

Here we have tagged the instance as a **Dev\_Web server 01**

Go to configure Security Groups later



*Tagging*

**Step 1)** In this next step of configuring Security Groups, you can restrict traffic on your instance ports. This is an added firewall mechanism provided by AWS apart from your instance’s OS firewall.

You can define open ports and IPs.

Since our server is a webserver=, we will do following things

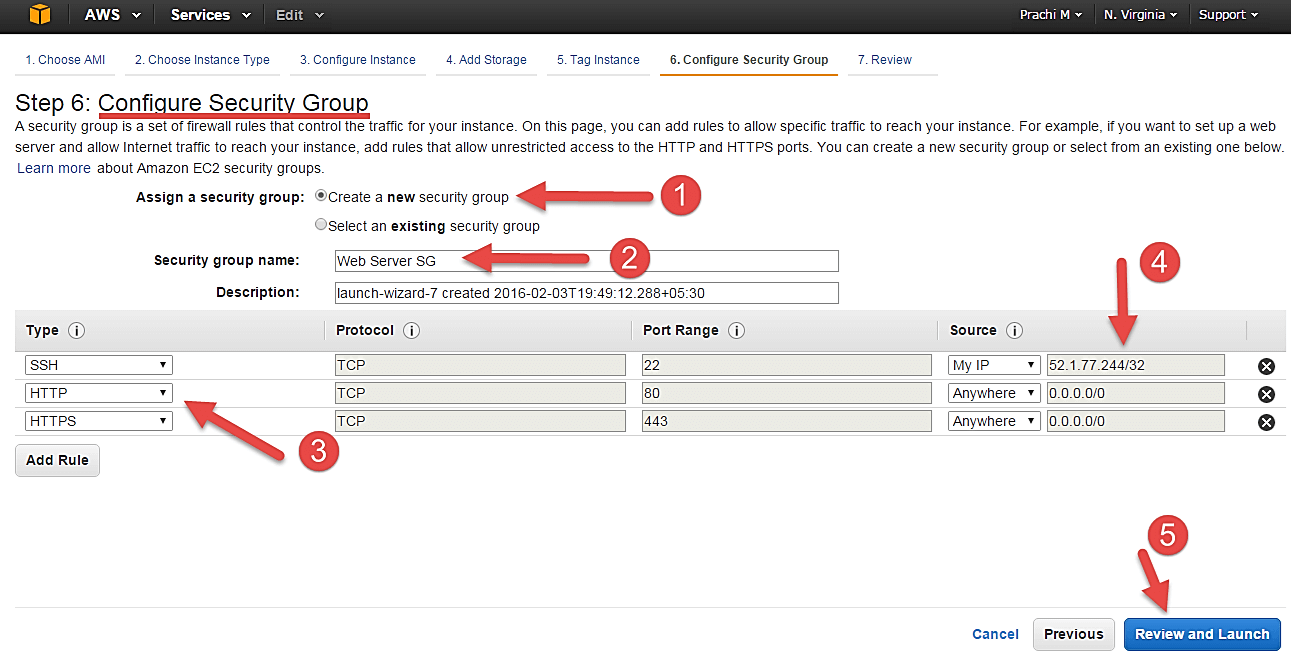
Creating a new Security Group

Naming our SG for easier reference

Defining protocols which we want enabled on my instance

Assigning IPs which are allowed to access our instance on the said protocols

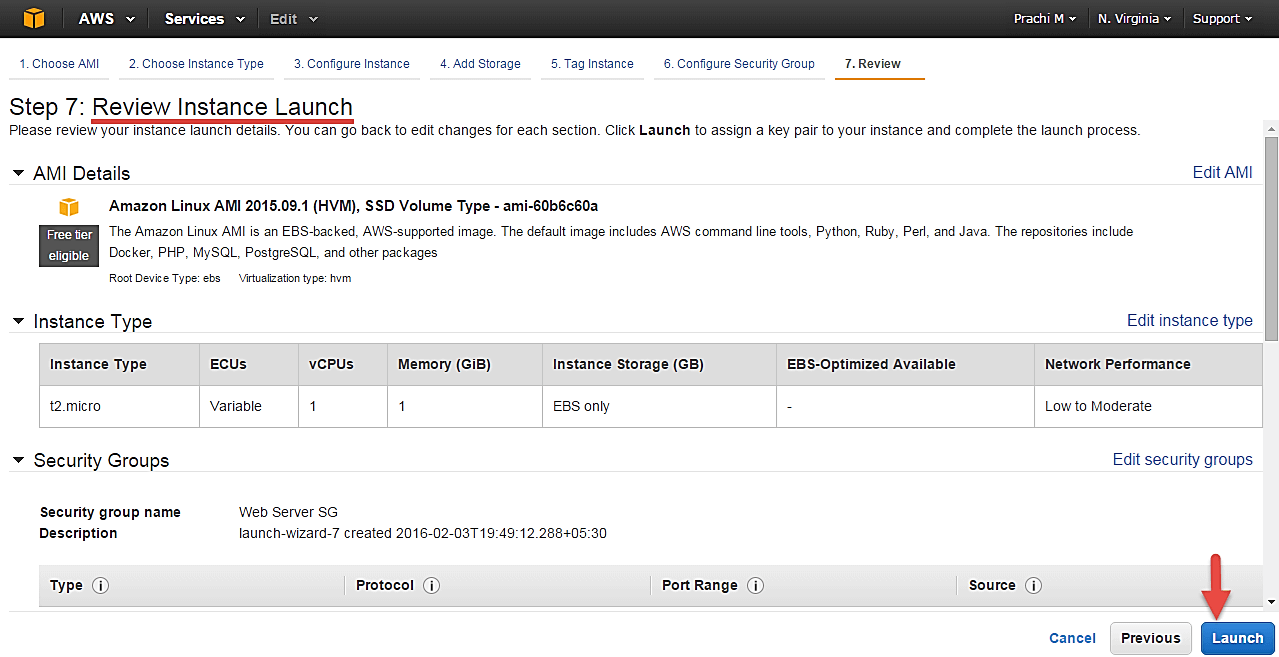
Once, the firewall rules are set- Review and launch



*Security Group*

Review Instances

**Step 1)** In this step, we will review all our choices and parameters and go ahead to launch our instance.



*Launch*

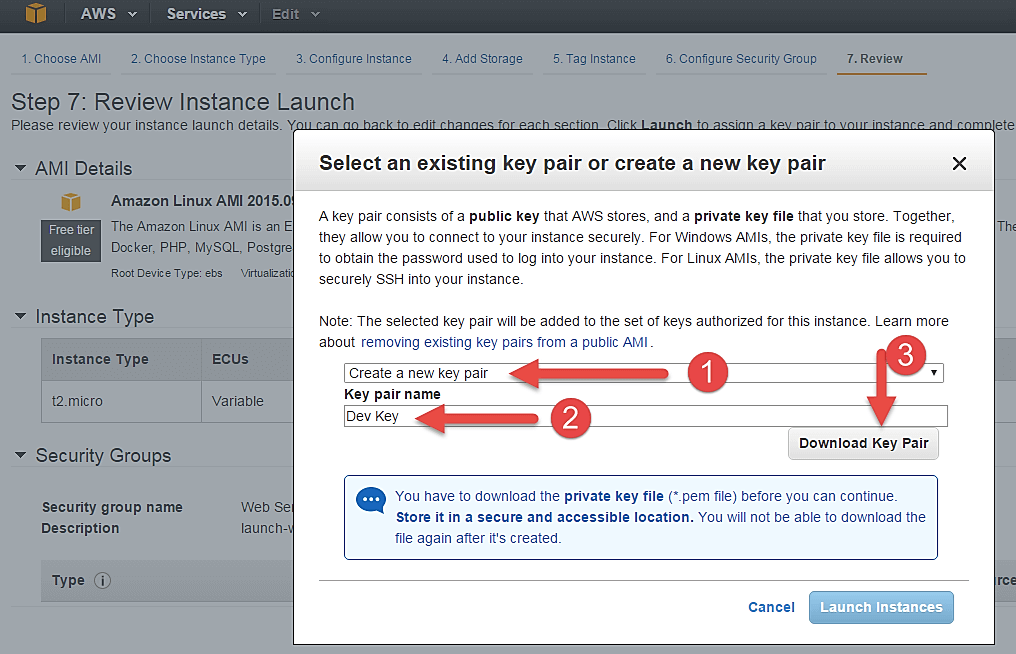
**Step 2)** In the next step you will be asked to create a key pair to login to you an instance. A key pair is a set of public-private keys.

AWS stores the private key in the instance, and you are asked to download the public key. Make sure you download the key and keep it safe and secured; if it is lost you cannot download it again.

Create a new key pair

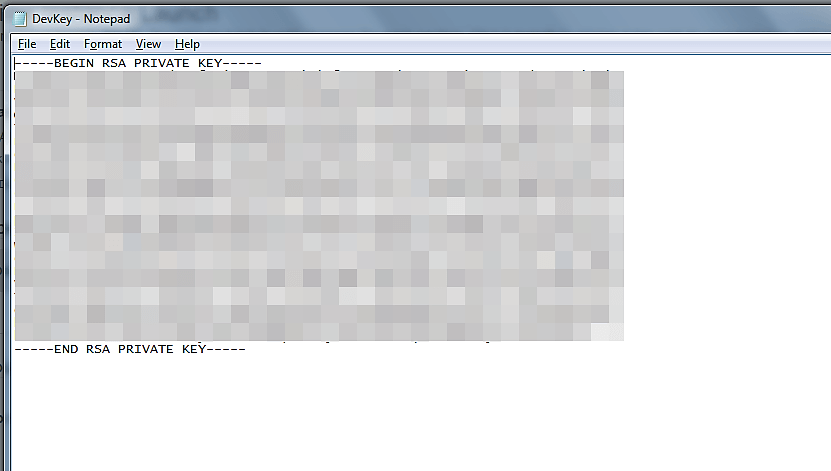
Give a name to your key

Download and save it in your secured folder



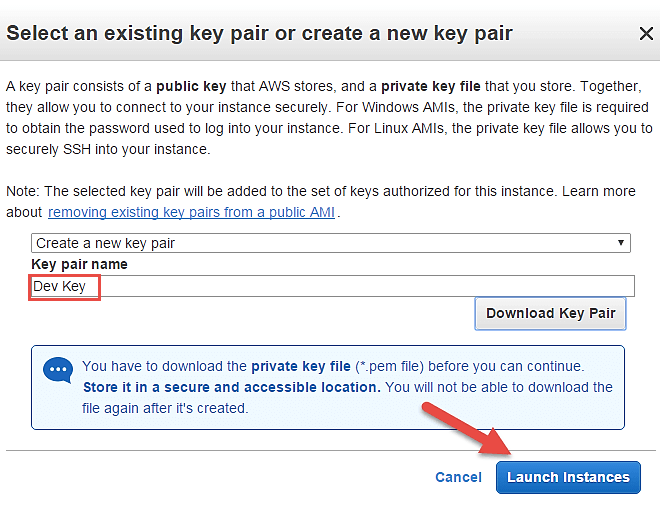
*Key Pair*

When you download your key, you can open and have a look at your RSA private key.

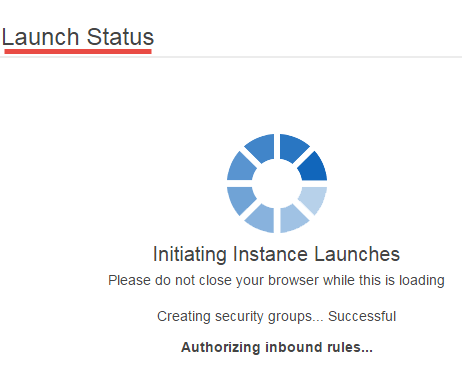


*RSA Key*

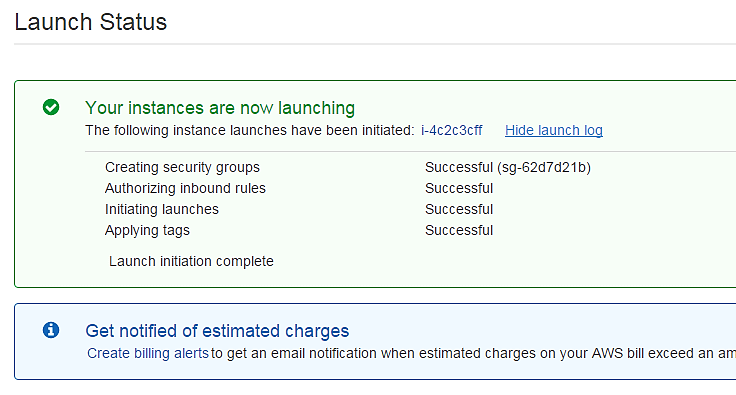
**Step 3)** Once you are done downloading and saving your key, launch your instance.



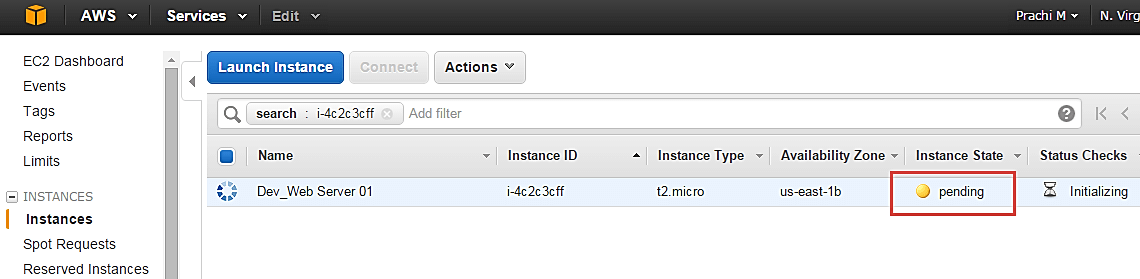
You can see the launch status meanwhile.



You can also see the launch log.

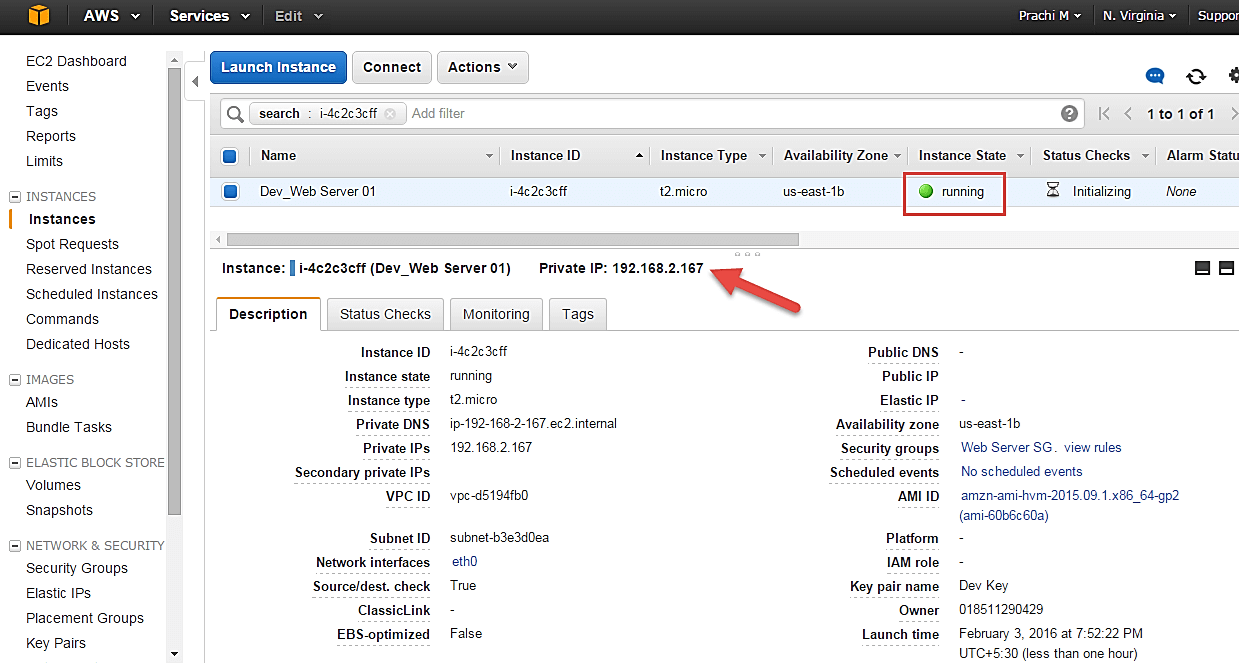


Click on the ‘Instances’ option on the left pane where you can see the status of the instance as ‘Pending’ for a brief while.



Once your instance is up and running, you can see its status as ‘Running’ now.

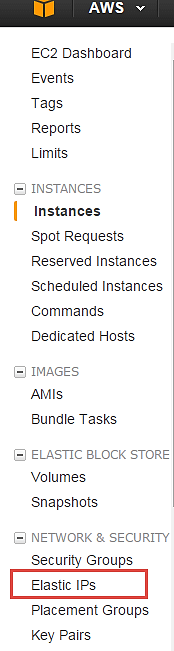
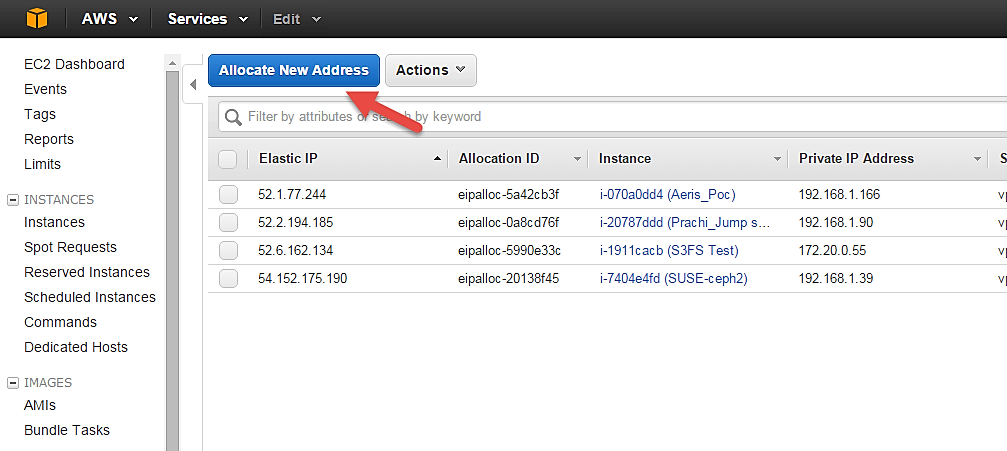
Note that the instance has received a Private IP from the pool of AWS.



Creating a EIP and connecting to your instance

An EIP is a static public IP provided by AWS. It stands for Elastic IP. Normally when you create an instance, it will receive a public IP from the AWS’s pool automatically. If you stop/reboot your instance, this public IP will change- it’dynamic. In order for your application to have a static IP from where you can connect via public networks, you can use an EIP.

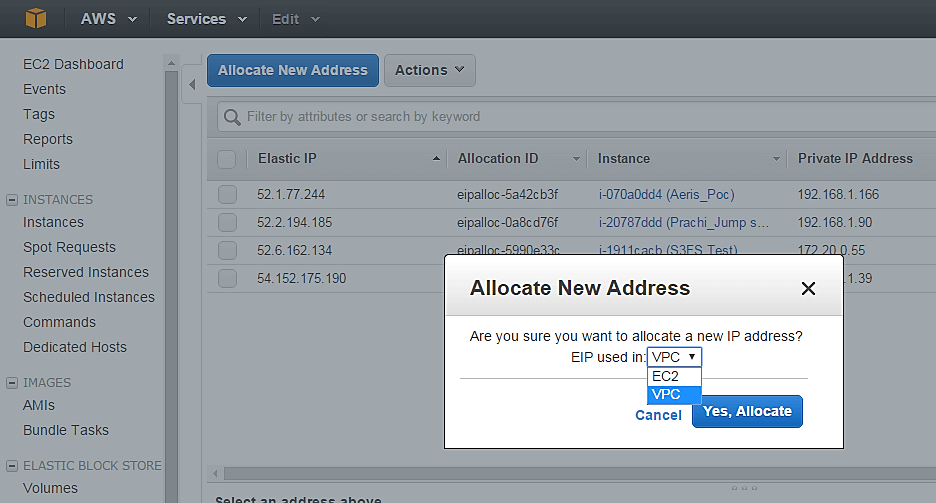
**Step 1)** On the left pane of EC2 Dashboard, you can go to ‘Elastic IPs’ as shown below.

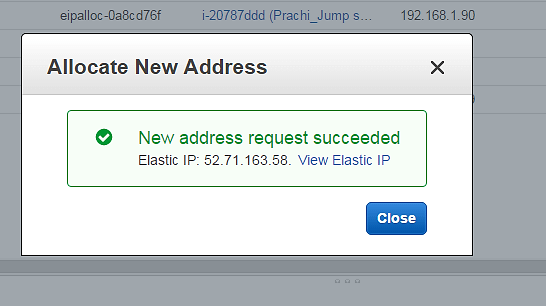
 

*Elastic IP*

**Step 2)** Allocate a new Elastic IP Address.

**Step 3)** Allocate this IP to be used in a VPC scope.

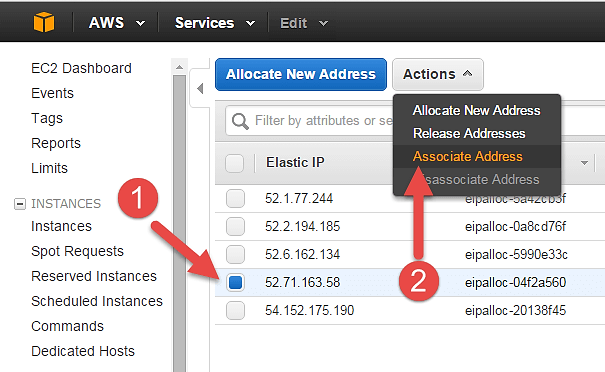


Your request will succeed if you don’t have 5 or more than 5 EIPs already in your account. 

**Step 4)** Now assign this IP to your instance.

Select the said IP

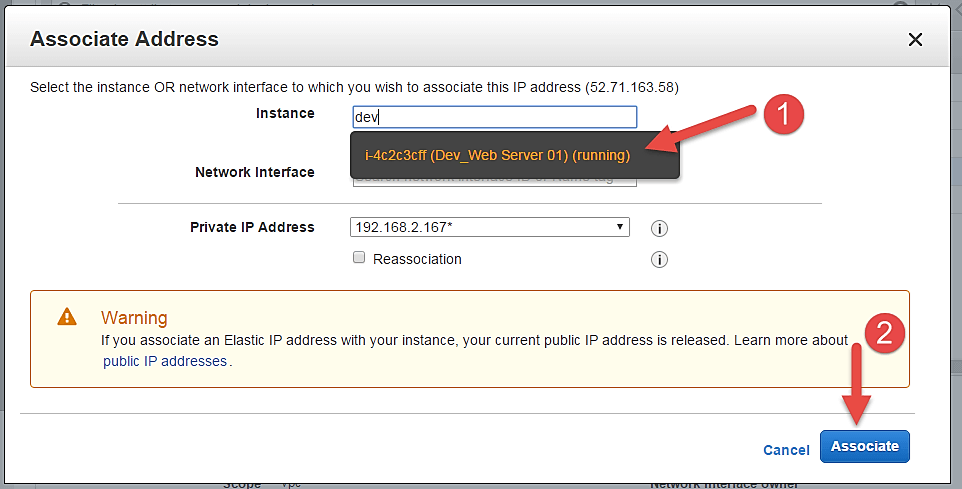
Click on Actions **->** Associate Address



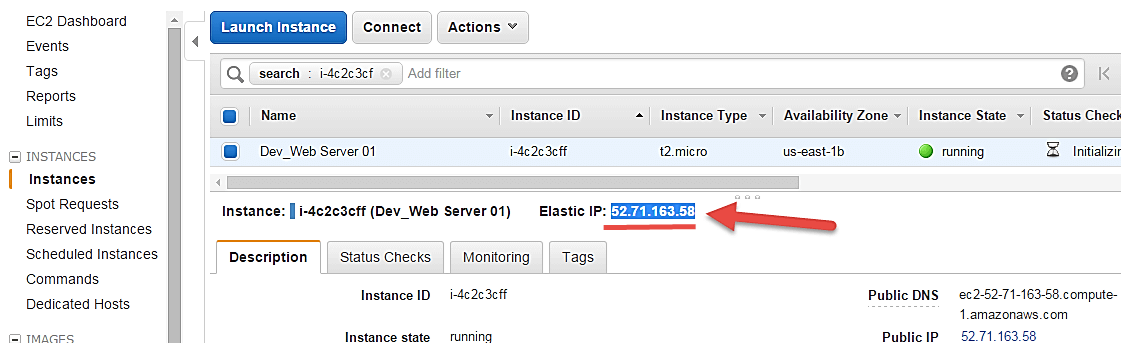
**Step 5)** In the next page,

Search for your instance and

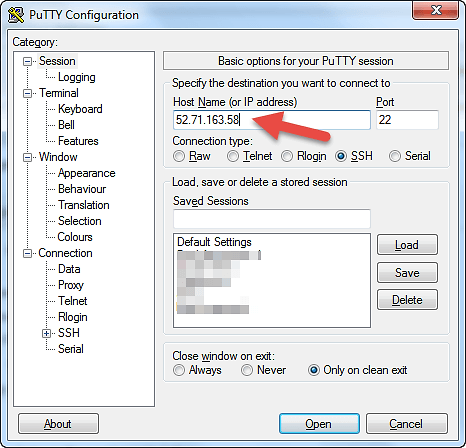
Associate the IP to it.



**Step 6)** Come back to your instances screen, you’ll see that your instance has received your EIP.



**Step 7)** Now open putty from your programs list and add your same EIP in there as below.



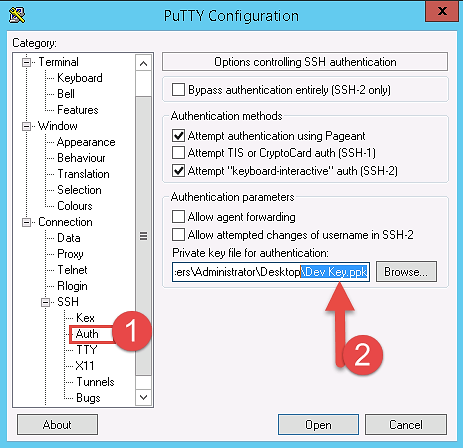
**Step 8)** In this step,

Add your private key in putty for secure connection

Go to Auth

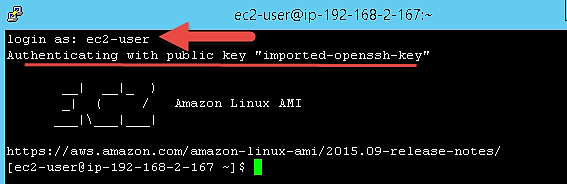
Add your private key in .ppk (putty private key) format

Once done click on “Open” button



Once you connect, you will successfully see the Linux prompt.

Please note that the machine you are connecting from should be enabled on the instance Security Group for SSH (like in the steps above).



Once you become familiar with the above steps for launching the instance, it becomes a matter of 2 minutes to launch the same!

You can now use your on-demand EC2 server for your applications.

**Summary:**

Finally, we saw in detail how to create an on-demand EC2 instance in this tutorial. Because it is an on-demand server, you can keep it running when in use and ‘Stop’ it when it’s unused to save on your costs.

You can provision a Linux or Windows EC2 instance or from any of the available AMIs in AWS Marketplace based on your choice of OS platform.

If your application is in production and you have to use it for years to come, you should consider provisioning a reserved instance to drastically save on your CAPEX.