**What is Cloud Computing?**

**Cloud computing** is an internet-based computing service in which large groups of remote servers are networked to allow centralized data storage, and online access to computer services or resources

Using cloud computing, organizations can use shared computing and storage resources rather than building, operating, and improving infrastructure on their own.

Cloud computing is a model that enables the following features.

* Users can provision and release resources on-demand.
* Resources can be scaled up or down automatically, depending on the load.
* Resources are accessible over a network with proper security.
* Cloud service providers can enable a pay-as-you-go model, where customers are charged based on the type of resources and per usage.

## Types of Clouds

There are three types of clouds − Public, Private, and Hybrid cloud.

### Public Cloud

In public cloud, the third-party service providers make resources and services available to their customers via Internet. Customer’s data and related security is with the service providers’ owned infrastructure.

### Private Cloud

A private cloud also provides almost similar features as public cloud, but the data and services are managed by the organization or by the third party only for the customer’s organization. In this type of cloud, major control is over the infrastructure so security related issues are minimized.

### Hybrid Cloud

A hybrid cloud is the combination of both private and public cloud. The decision to run on private or public cloud usually depends on various parameters like sensitivity of data and applications, industry certifications and required standards, regulations, etc.

## Cloud Service Models

There are three types of service models in cloud − IaaS, PaaS, and SaaS.

### IaaS

IaaS stands for **Infrastructure as a Service**. It provides users with the capability to provision processing, storage, and network connectivity on demand. Using this service model, the customers can develop their own applications on these resources.

### PaaS

PaaS stands for **Platform as a Service**. Here, the service provider provides various services like databases, queues, workflow engines, e-mails, etc. to their customers. The customer can then use these components for building their own applications. The services, availability of resources and data backup are handled by the service provider that helps the customers to focus more on their application's functionality.

### SaaS

SaaS stands for **Software as a Service**. As the name suggests, here the third-party providers provide end-user applications to their customers with some administrative capability at the application level, such as the ability to create and manage their users. Also some level of customizability is possible such as the customers can use their own corporate logos, colors, etc.

### Load Balancing

**Load balancing** simply means to hardware or software load over web servers, that improver's the efficiency of the server as well as the application. Following is the diagrammatic representation of AWS architecture with load balancing.

**Elastic Load Balancing** can dynamically grow and shrink the load-balancing capacity to adjust to traffic demands and also support sticky sessions to address more advanced routing needs

### **Amazon Cloud-front**

It is responsible for content delivery, i.e. used to deliver website. It may contain dynamic, static, and streaming content using a global network of edge locations. Requests for content at the user's end are automatically routed to the nearest edge location, which improves the performance.

### **Elastic Load Balancer**

It is used to spread the traffic to web servers, which improves performance. AWS provides the Elastic Load Balancing service, in which traffic is distributed to EC2 instances over multiple available zones, and dynamic addition and removal of Amazon EC2 hosts from the load-balancing rotation.

Elastic Load Balancing can dynamically grow and shrink the load-balancing capacity as per the traffic conditions.

### **Security Management**

Amazon’s Elastic Compute Cloud (EC2) provides a feature called security groups, which is similar to an inbound network firewall, in which we have to specify the protocols, ports, and source IP ranges that are allowed to reach your EC2 instances.

Each EC2 instance can be assigned one or more security groups, each of which routes the appropriate traffic to each instance. Security groups can be configured using specific subnets or IP addresses which limits access to EC2 instances.

### **Elastic Caches**

Amazon Elastic Cache is a web service that manages the memory cache in the cloud. In memory management, cache has a very important role and helps to reduce the load on the services, improves the performance and scalability on the database tier by caching frequently used information.

### **Amazon RDS**

Amazon RDS (Relational Database Service) provides a similar access as that of MySQL, Oracle, or Microsoft SQL Server database engine. The same queries, applications, and tools can be used with Amazon RDS.

It automatically patches the database software and manages backups as per the user’s instruction. It also supports point-in-time recovery. There are no up-front investments required, and we pay only for the resources we use.

### **Hosting RDMS on EC2 Instances**

Amazon RDS allows users to install RDBMS (Relational Database Management System) of your choice like MySQL, Oracle, SQL Server, DB2, etc. on an EC2 instance and can manage as required.

Amazon EC2 uses Amazon EBS (Elastic Block Storage) similar to network-attached storage. All data and logs running on EC2 instances should be placed on Amazon EBS volumes, which will be available even if the database host fails.

Amazon EBS volumes automatically provide redundancy within the availability zone, which increases the availability of simple disks. Further if the volume is not sufficient for our databases needs, volume can be added to increase the performance for our database.

Using Amazon RDS, the service provider manages the storage and we only focus on managing the data.

### **Storage & Backups**

AWS cloud provides various options for storing, accessing, and backing up web application data and assets. The Amazon S3 (Simple Storage Service) provides a simple web-services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web.

Amazon S3 stores data as objects within resources called **buckets**. The user can store as many objects as per requirement within the bucket, and can read, write and delete objects from the bucket.

Amazon EBS is effective for data that needs to be accessed as block storage and requires persistence beyond the life of the running instance, such as database partitions and application logs.

Amazon EBS volumes can be maximized up to 1 TB, and these volumes can be striped for larger volumes and increased performance. Provisioned IOPS volumes are designed to meet the needs of database workloads that are sensitive to storage performance and consistency.

Amazon EBS currently supports up to 1,000 IOPS per volume. We can stripe multiple volumes together to deliver thousands of IOPS per instance to an application.

### **Auto Scaling**

The difference between AWS cloud architecture and the traditional hosting model is that AWS can dynamically scale the web application fleet on demand to handle changes in traffic.

In the traditional hosting model, traffic forecasting models are generally used to provision hosts ahead of projected traffic. In AWS, instances can be provisioned on the fly according to a set of triggers for scaling the fleet out and back in. Amazon Auto Scaling can create capacity groups of servers that can grow or shrink on demand.

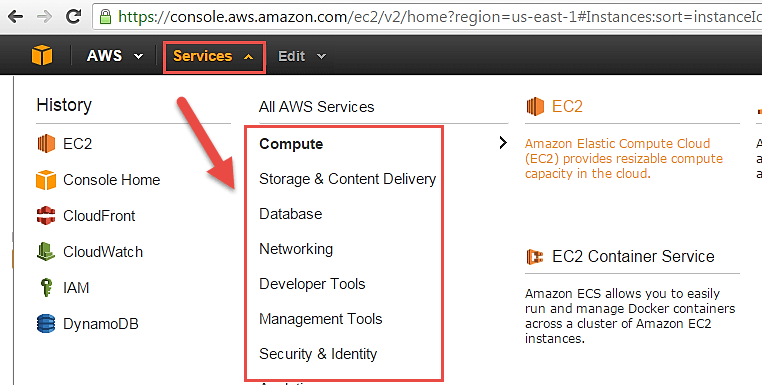
**how to launch an on-demand EC2 instance in AWS Cloud**

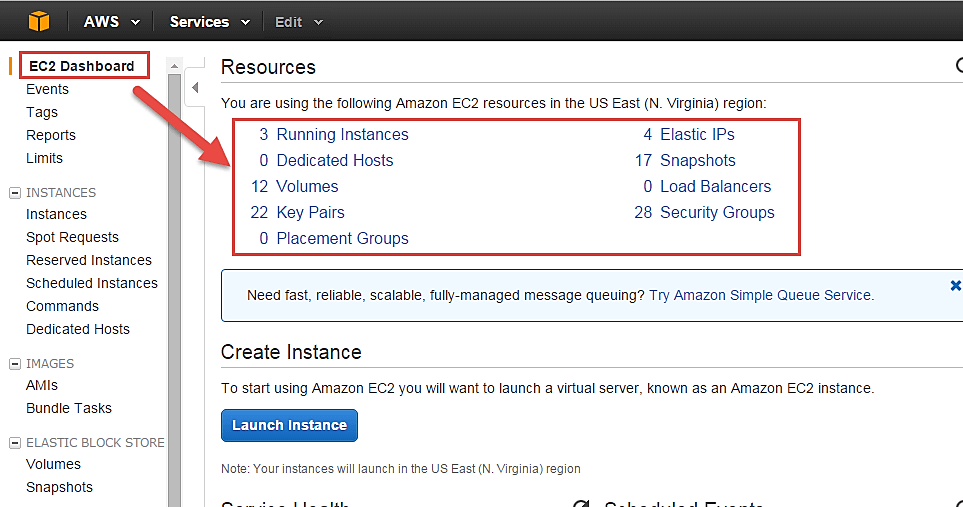
**Login and access to AWS services**

**Step 1)** In this step,

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

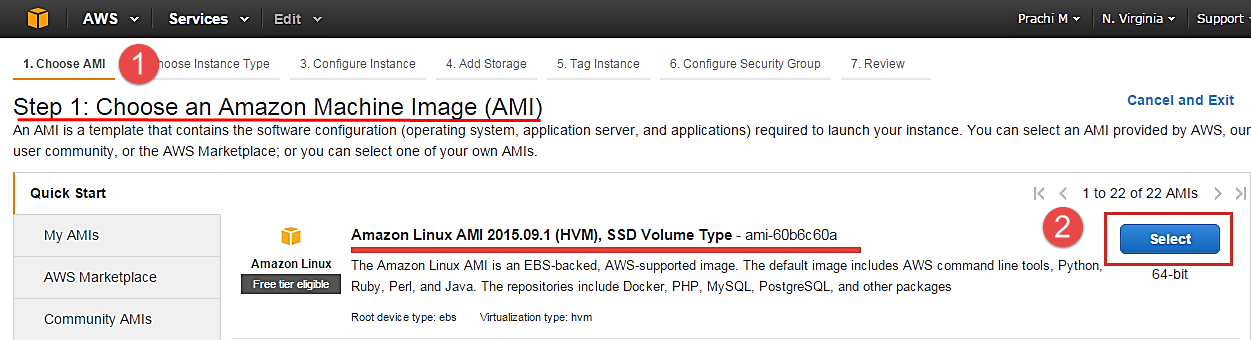
**Step 3)** In this step

* 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,

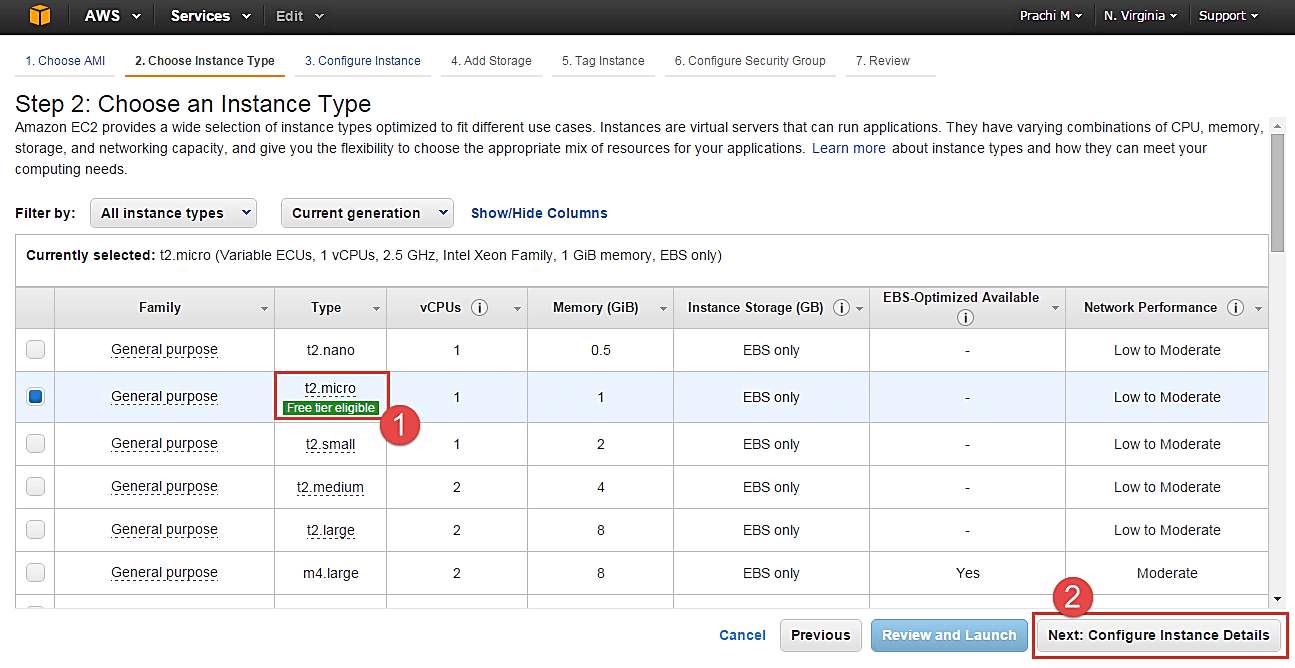
1. You will be asked to choose an AMI of your choice. (An AMI is an Amazon Machine Image. It is a template basically of an Operating System platform which you can use as a base to create your instance). Once you launch an EC2 instance from your preferred AMI, the instance will automatically be booted with the desired OS. (We will see more about AMIs in the coming part of the tutorial).
2. Here we are choosing the default Amazon [Linux](https://www.guru99.com/unix-linux-tutorial.html) (64 bit) AMI.



## Choose Instance Types

**Step 1)** In the next step, you have to choose the type of instance you require based on your business needs.

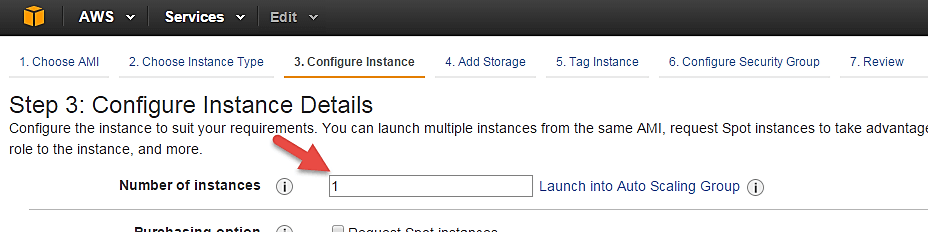
1. We will choose t2.micro instance type, which is a 1vCPU and 1GB memory server offered by AWS.
2. Click on "Configure Instance Details" for further configurations



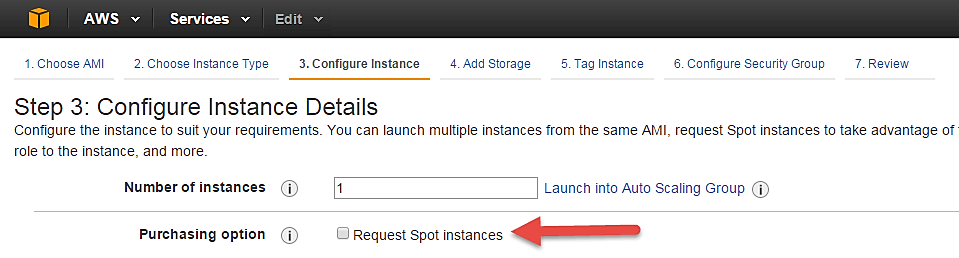
* In the next step of the wizard, enter details like no. of instances you want to launch at a time.
* Here we are launching one instance.

## Configure Instance

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



**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).

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA8.png)

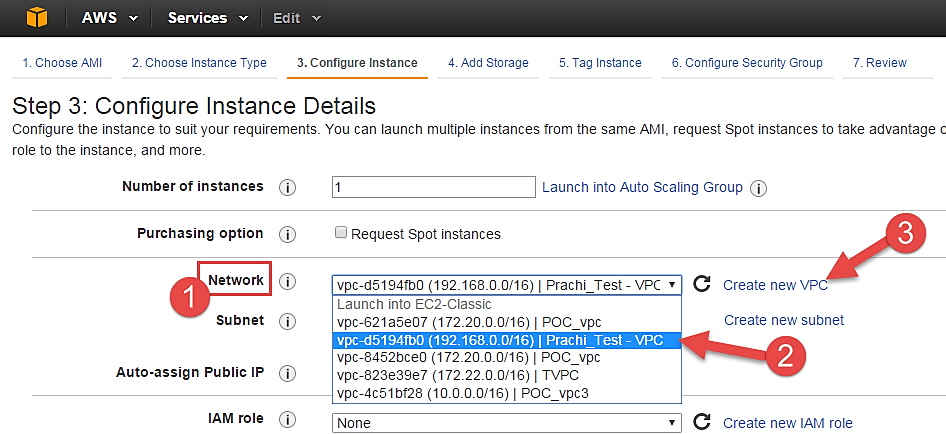
**Step 3)** Next, we have to configure some basic networking details for our EC2 server.

* 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,

1. Network section will give a list of VPCs available in our platform.
2. Select an already existing VPC
3. You can also create a new VPC

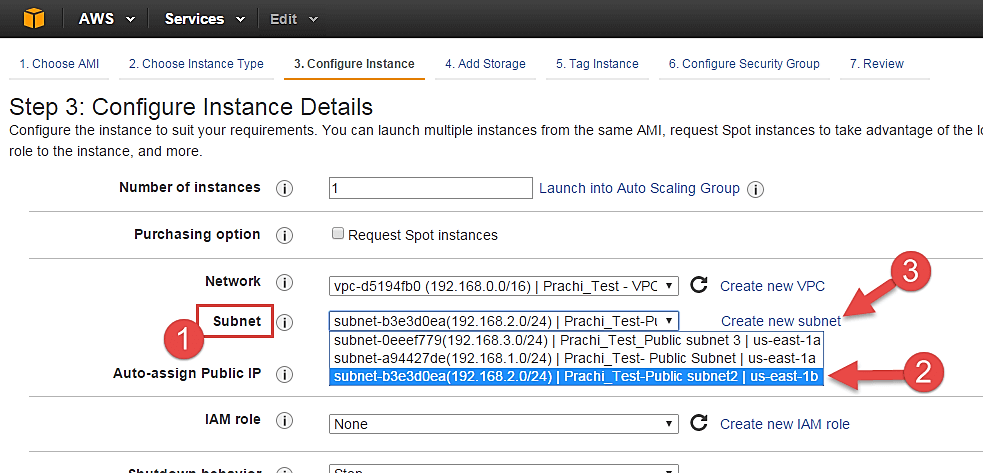
Here I have selected an already existing VPC where I want to launch my instance.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA9.png)

**Step 4)** In this step,

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

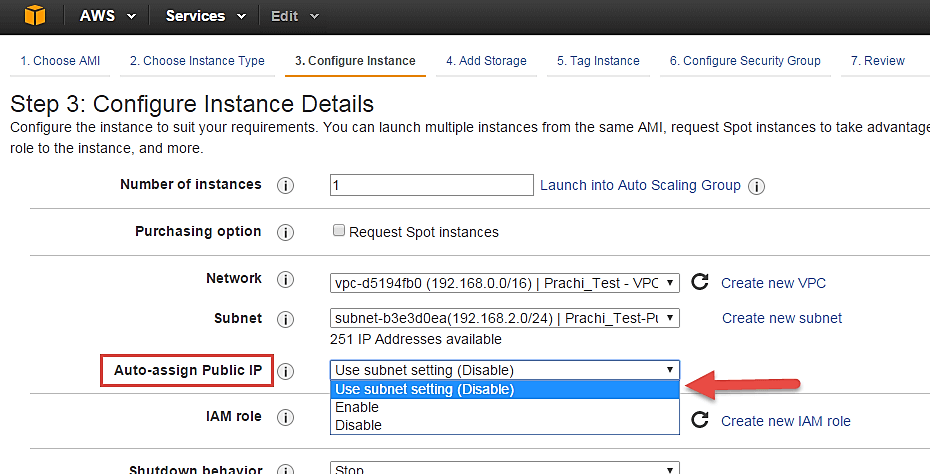
1. Under Subnets, you can choose the subnet where you want to place your instance.
2. I have chosen an already existing public subnet.
3. You can also create a new subnet in this step.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA10.png)

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

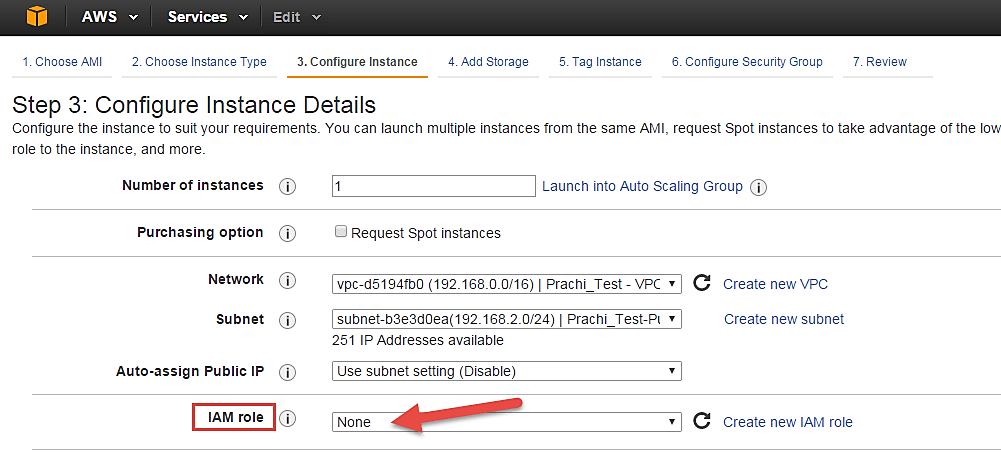
**Step 5)** In this step,

* 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.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA11.png)

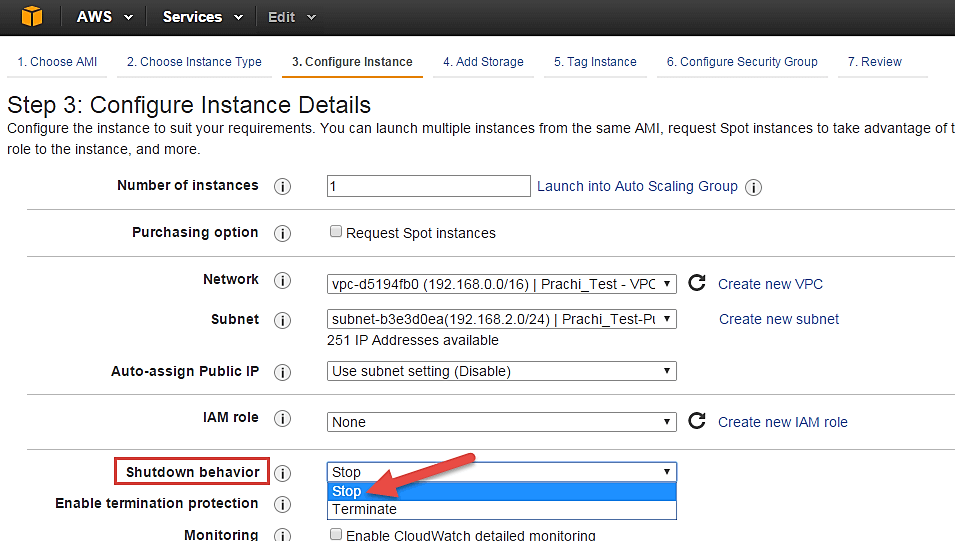
**Step 6)** In this step,

* 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.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA12.png)

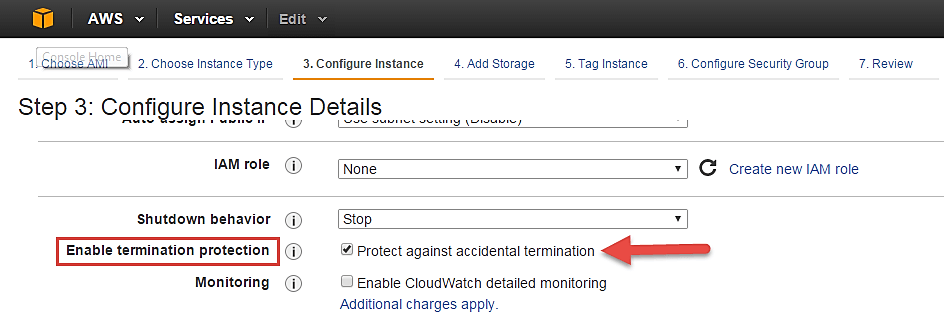
**Step 7)** In this step, you have to do following things

* 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.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA13.png)

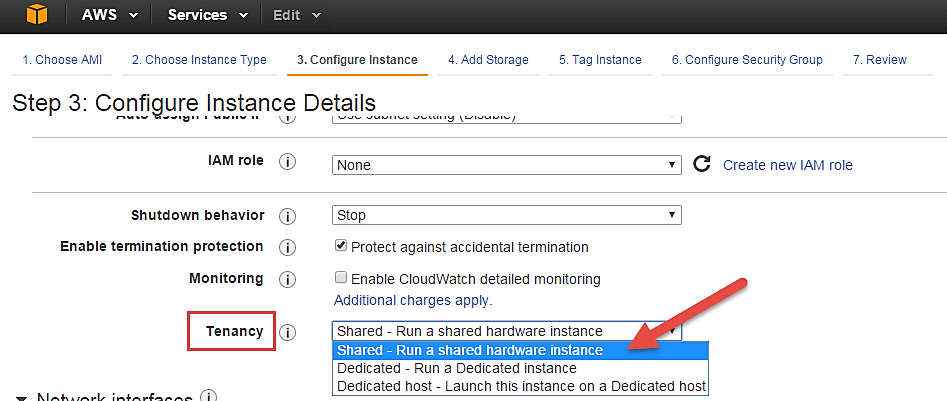
**Step 8)** In this step,

* 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.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA14.png)

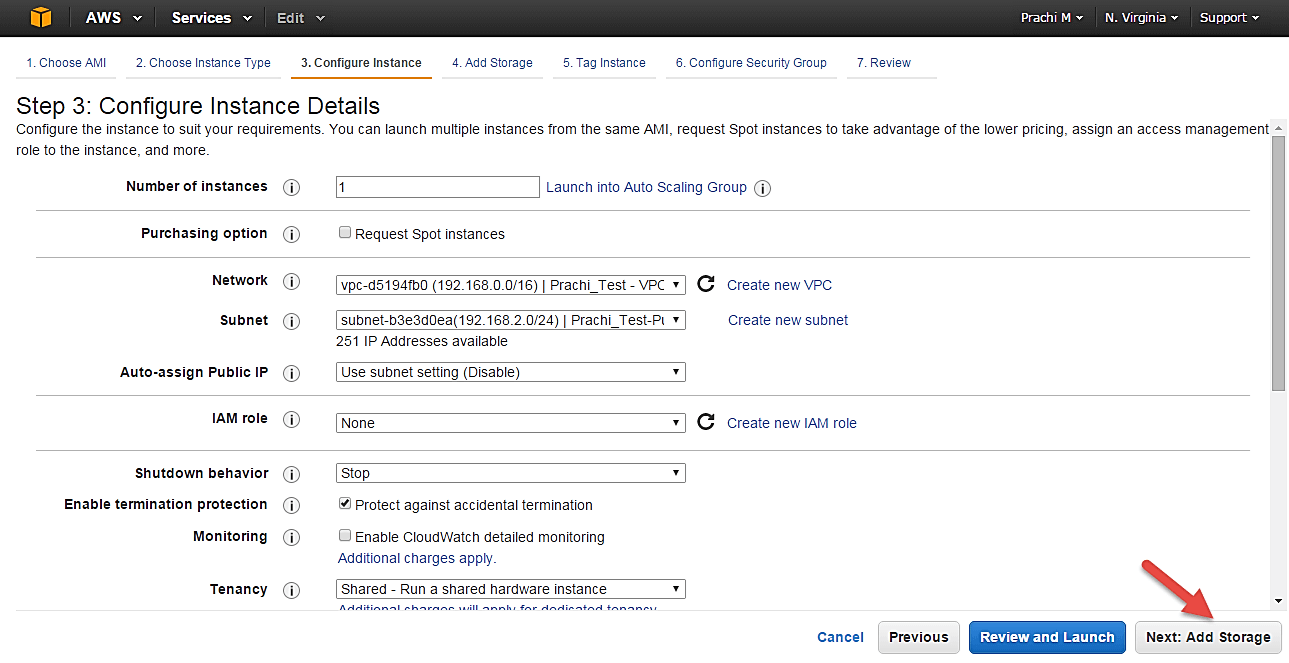
**Step 9)** In this step,

* 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.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA15.png)

**Step 10)** In this step,

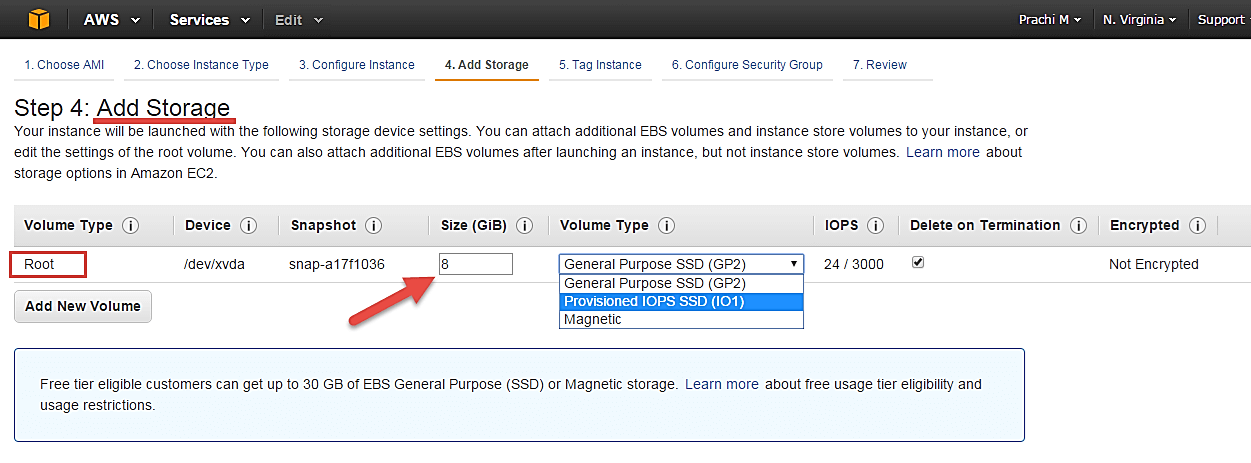
* Click on 'Add Storage' to add data volumes to your instance in next step.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA16.png)

## 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.

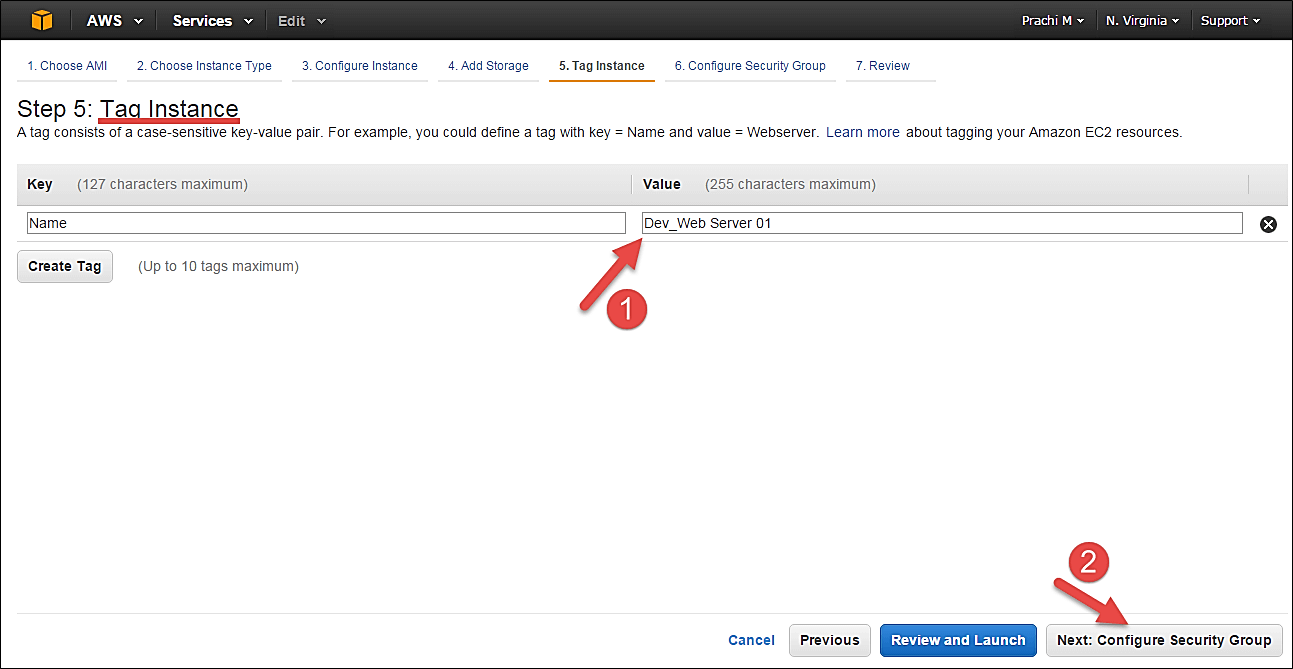
[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA17.png)

## 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.

1. Here we have tagged the instance as a **Dev\_Web server 01**
2. Go to configure Security Groups later

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA18.png)

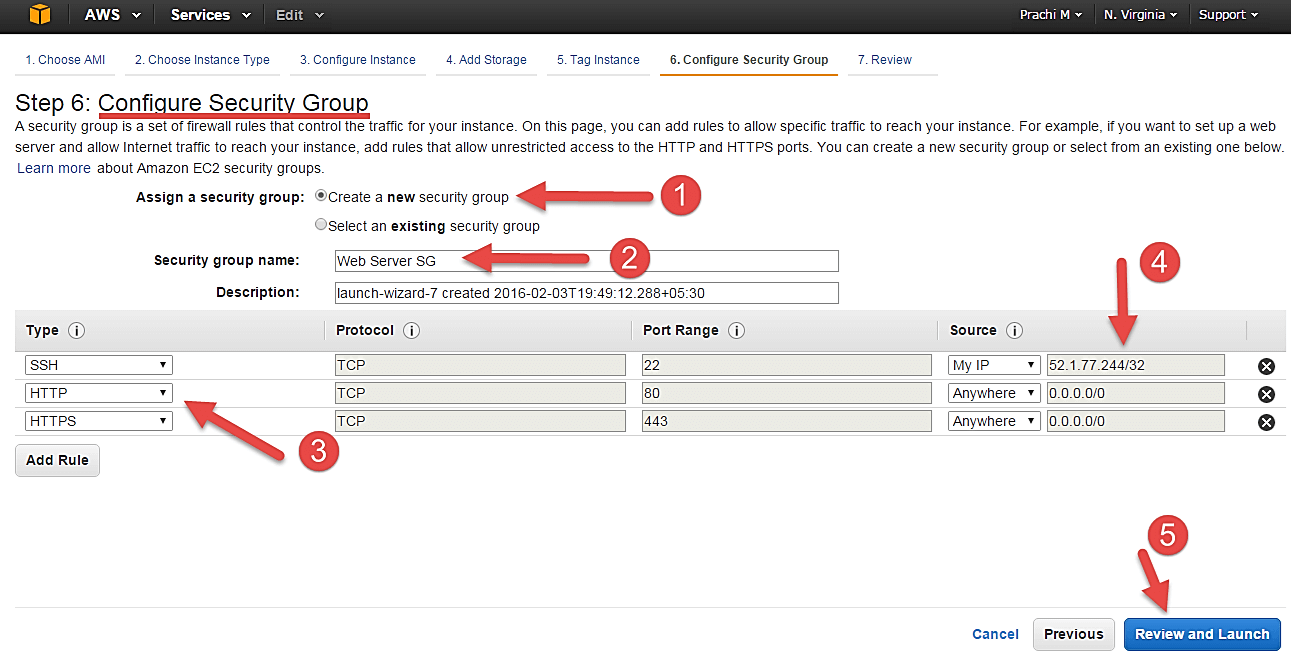
## Configuring Security Groups

**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.

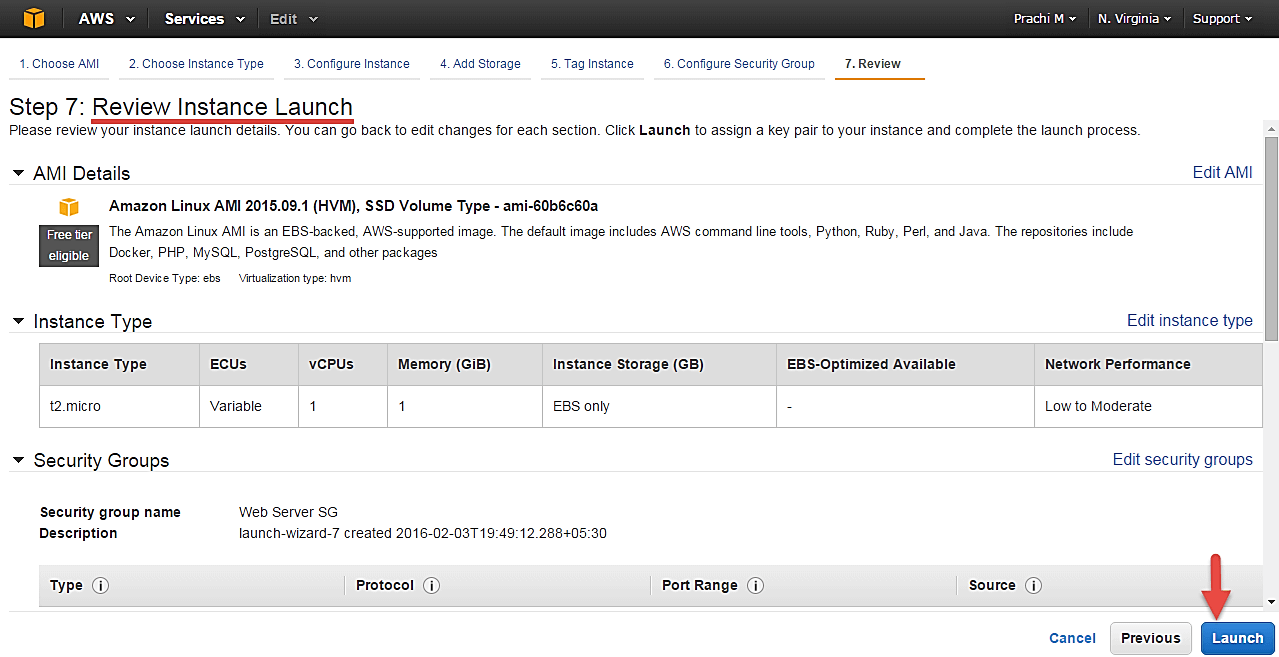
* Since our server is a webserver=, we will do following things

1. Creating a new Security Group
2. Naming our SG for easier reference
3. Defining protocols which we want enabled on my instance
4. Assigning IPs which are allowed to access our instance on the said protocols
5. Once, the firewall rules are set- Review and launch

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA19.png)

## Review Instances

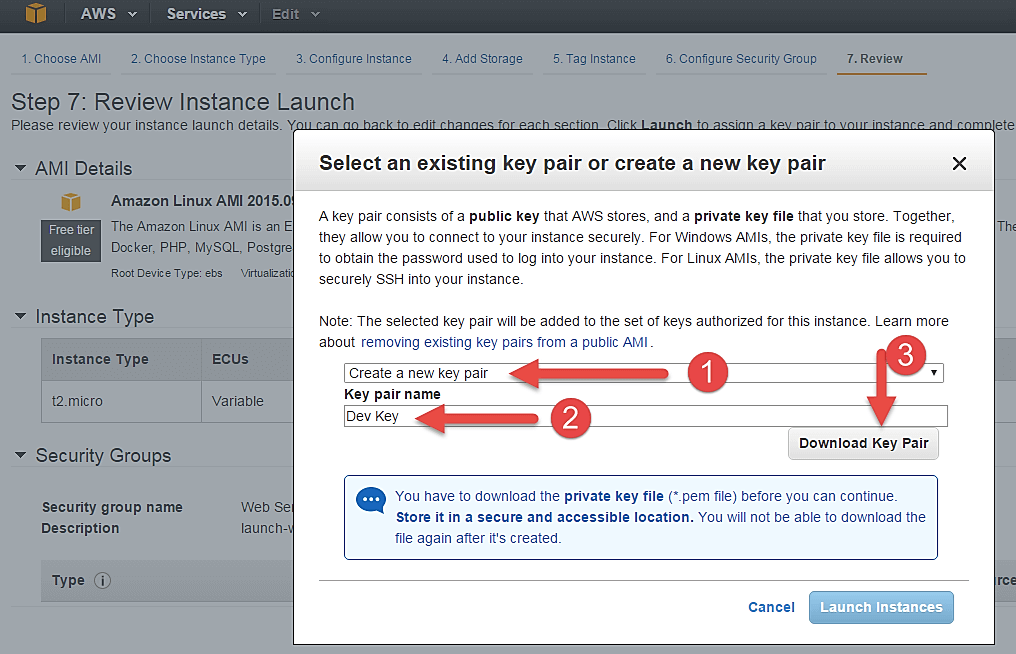
**Step 1)** In this step, we will review all our choices and parameters and go ahead to launch our instance.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA20.png)

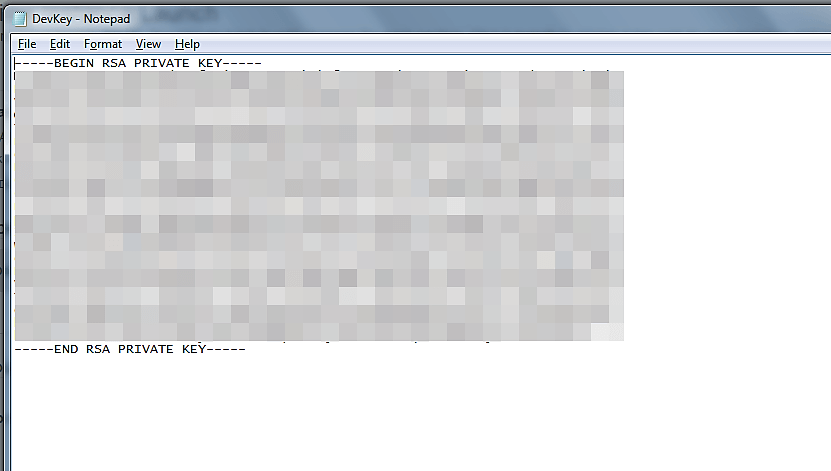
**Step 2)** In the next step you will be asked to create a key pair to login to you an instance. A key pair is a set of public-private keys.

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.

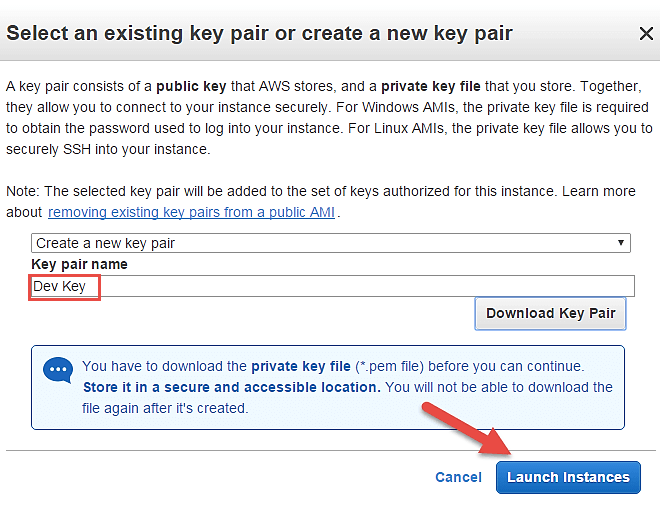
1. Create a new key pair
2. Give a name to your key
3. Download and save it in your secured folder

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA21.png)

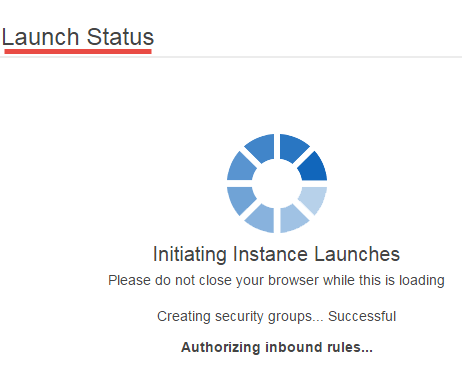
* When you download your key, you can open and have a look at your RSA private key.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA22.png)

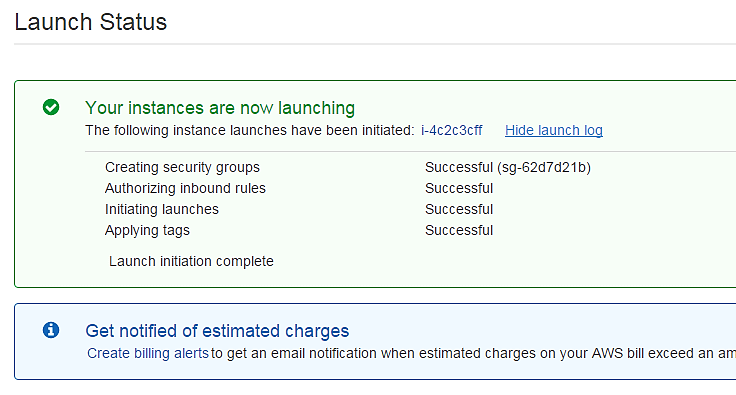
**Step 3)** Once you are done downloading and saving your key, launch your instance.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA23.png)

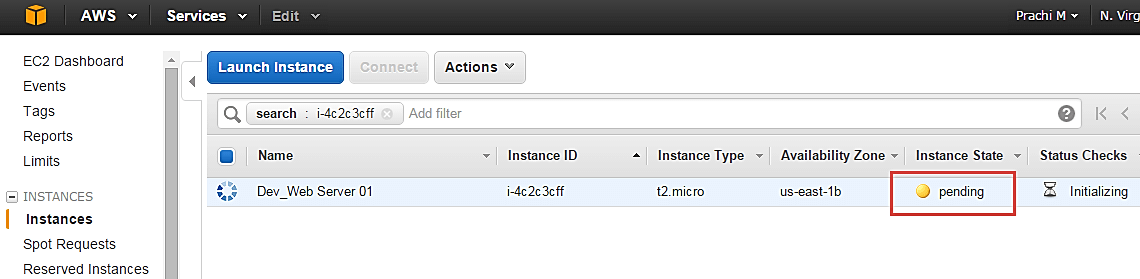
* You can see the launch status meanwhile.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA24.png)

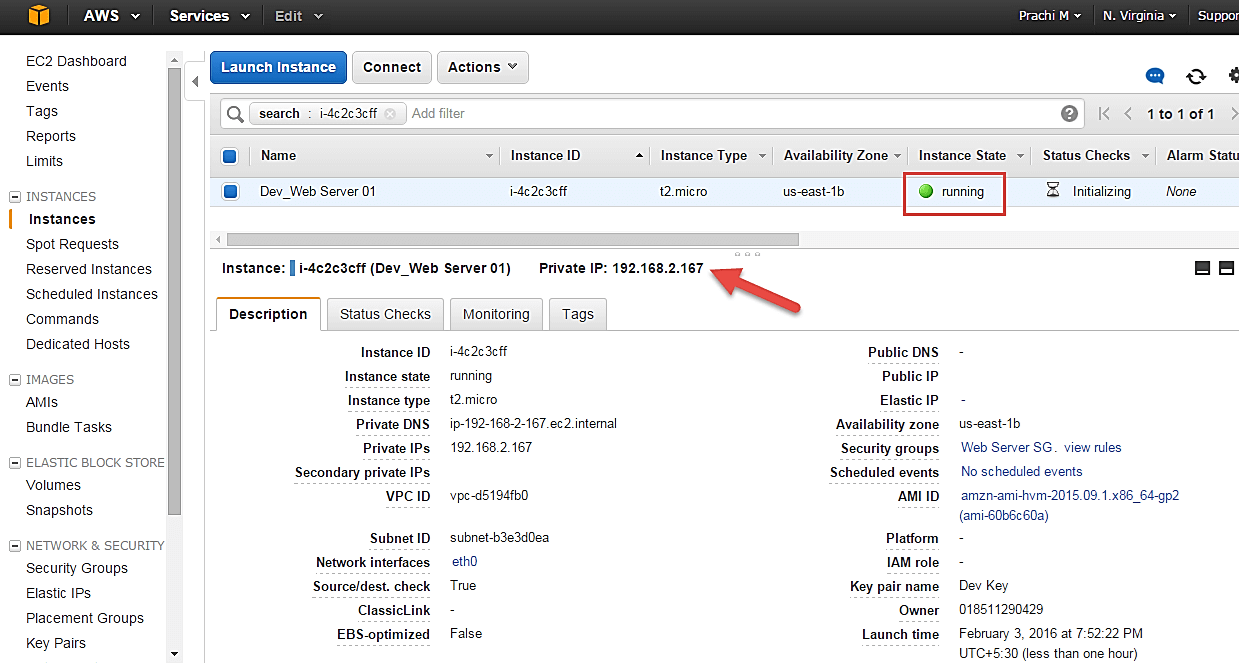
* You can also see the launch log.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA25.png)

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

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA26.png)

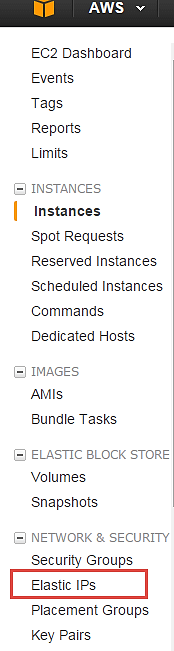
* Once your instance is up and running, you can see its status as 'Running' now.
* Note that the instance has received a Private IP from the pool of AWS.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA27.png)

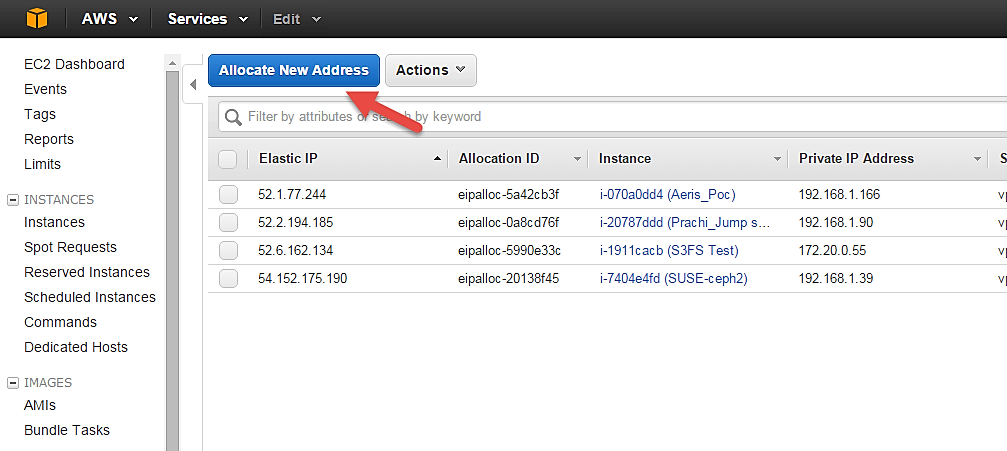
## Creating a EIP and connecting to your instance

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.

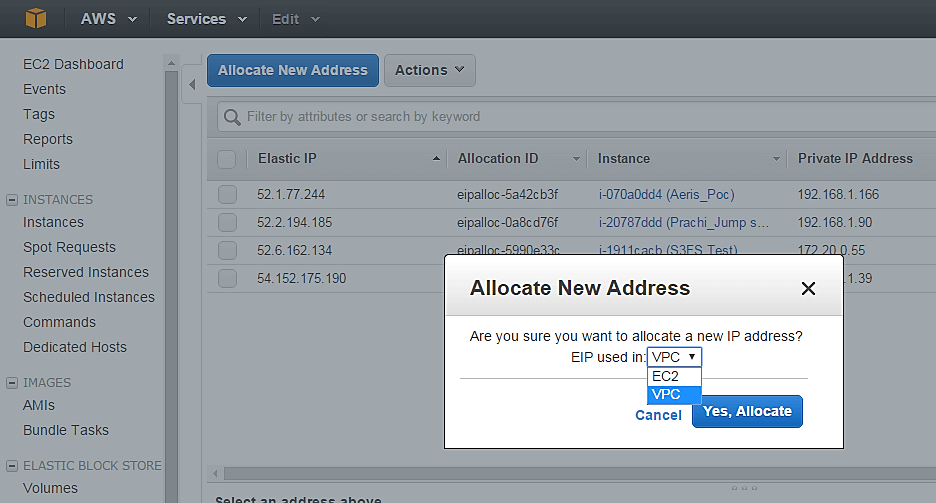
**Step 1)** On the left pane of EC2 Dashboard, you can go to 'Elastic IPs' as shown below.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA28.png)

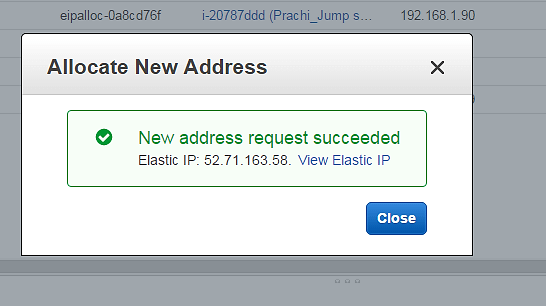
**Step 2)** Allocate a new Elastic IP Address.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA29.png)

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

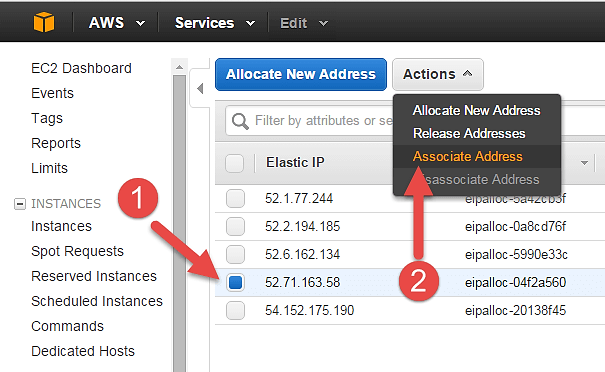
[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA30.png)

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

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA31.png)

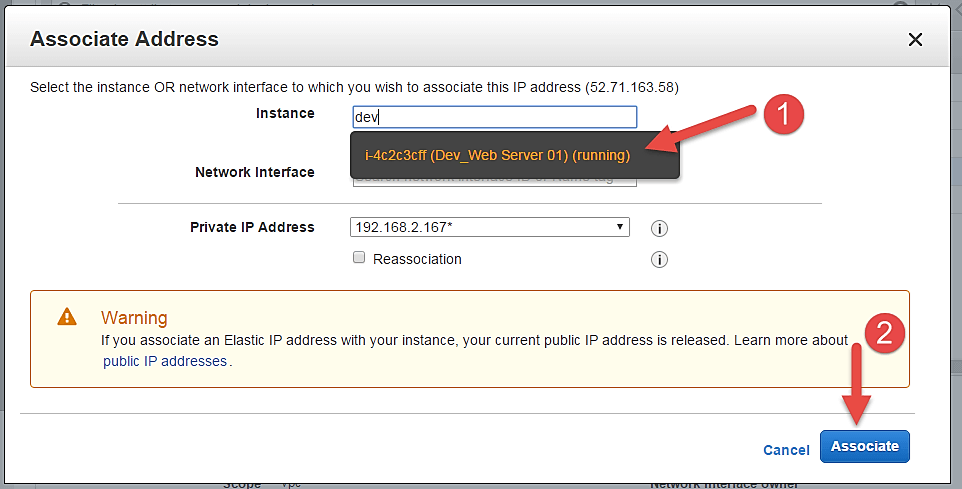
**Step 4)** Now assign this IP to your instance.

1. Select the said IP
2. Click on Actions **->** Associate Address

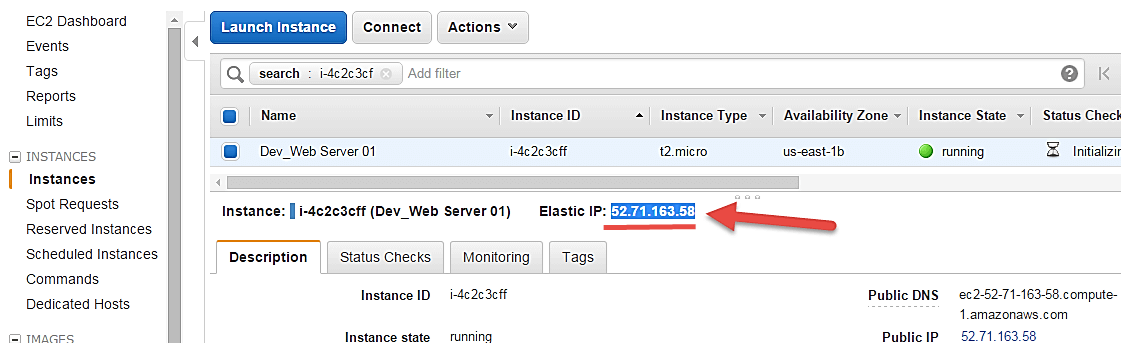
[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA32.png)

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

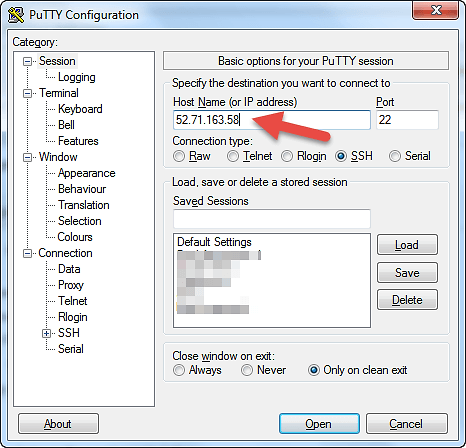
1. Search for your instance and
2. Associate the IP to it.

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA33.png)

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

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA34.png)

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

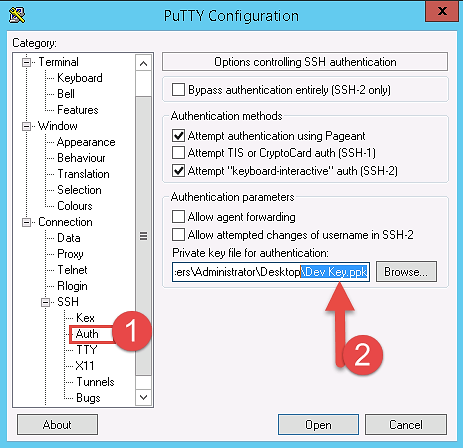
[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA35.png)

**Step 8)** In this step,

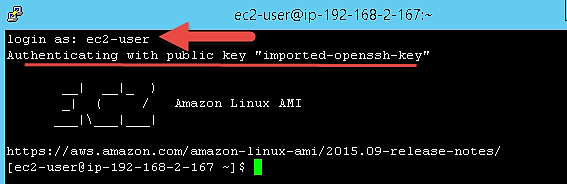
Add your private key in putty for secure connection

1. Go to Auth
2. Add your private key in .ppk (putty private key) format

Once done click on "Open" button

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA36.png)

* Once you connect, you will successfully see the [Linux](https://www.guru99.com/unix-linux-tutorial.html) 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).

[](https://cdn.guru99.com/images/3-2016/032816_0629_CreatinganA37.png)

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:**

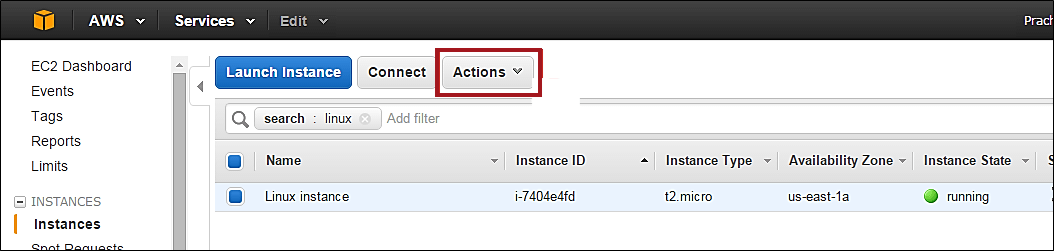
Thus, we saw in detail how to create an on-demand EC2 instance in this tutorial. Because it is an on-demand server, you can keep it running when in use and 'Stop' it when it's unused to save on your costs.

You can provision a [Linux](https://www.guru99.com/unix-linux-tutorial.html) or Windows EC2 instance or from any of the available AMIs in AWS Marketplace based on your choice of OS platform.

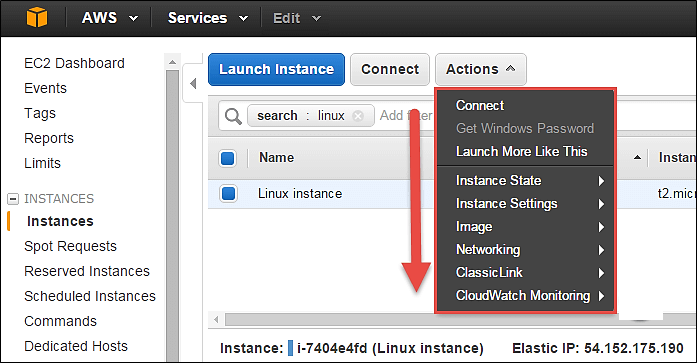
If your application is in production and you have to use it for years to come, you should consider provisioning a reserved instance to drastically save on your CAPEX.

### Checking the modification parameters

**Step 1)** On the EC2 Dashboard, select the instance whose configuration parameters you want to modify and Click on the "Actions" button as shown below.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth4.png)

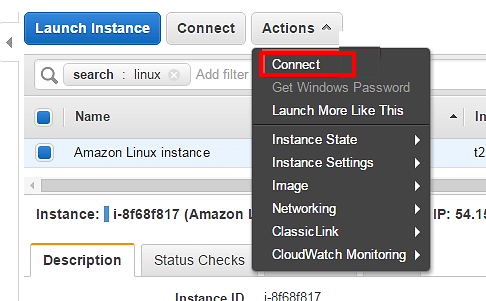
**Step 2)** As you click the button, the drop- down will show us all the areas where we can modify the instance characteristics.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth5.png)

### Viewing the connection details

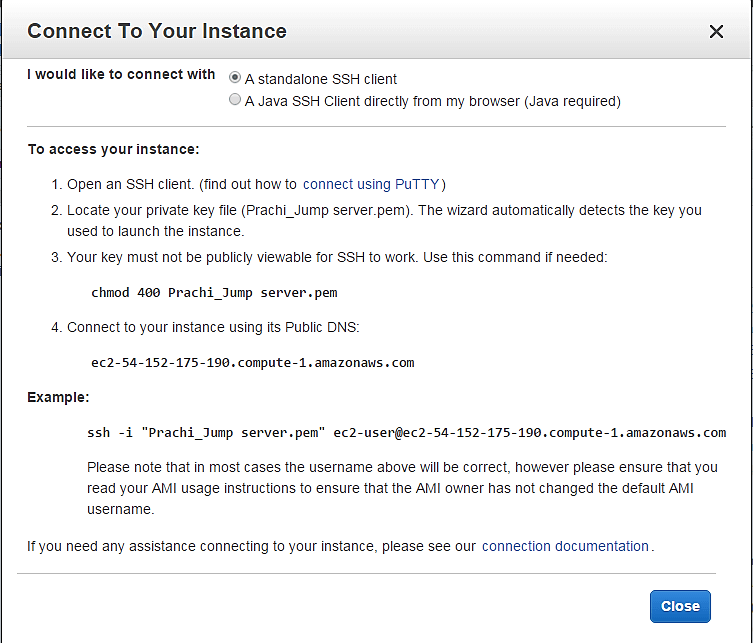
Connect option below will show us ways in which we can connect to an EC2 instance.

**Step 1)** Click on option 'Connect.'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth6.png)

You may choose to connect with a standalone SSH client or a [Java](https://www.guru99.com/java-tutorial.html) client. You will get a step-by-step procedure on how you can connect to your instance.

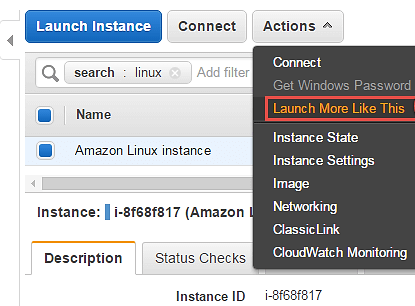
For this tutorial, we can see the connection methods for a [Linux](https://www.guru99.com/unix-linux-tutorial.html) instance.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth7.png)

### Launching multiple instances with the similar configuration

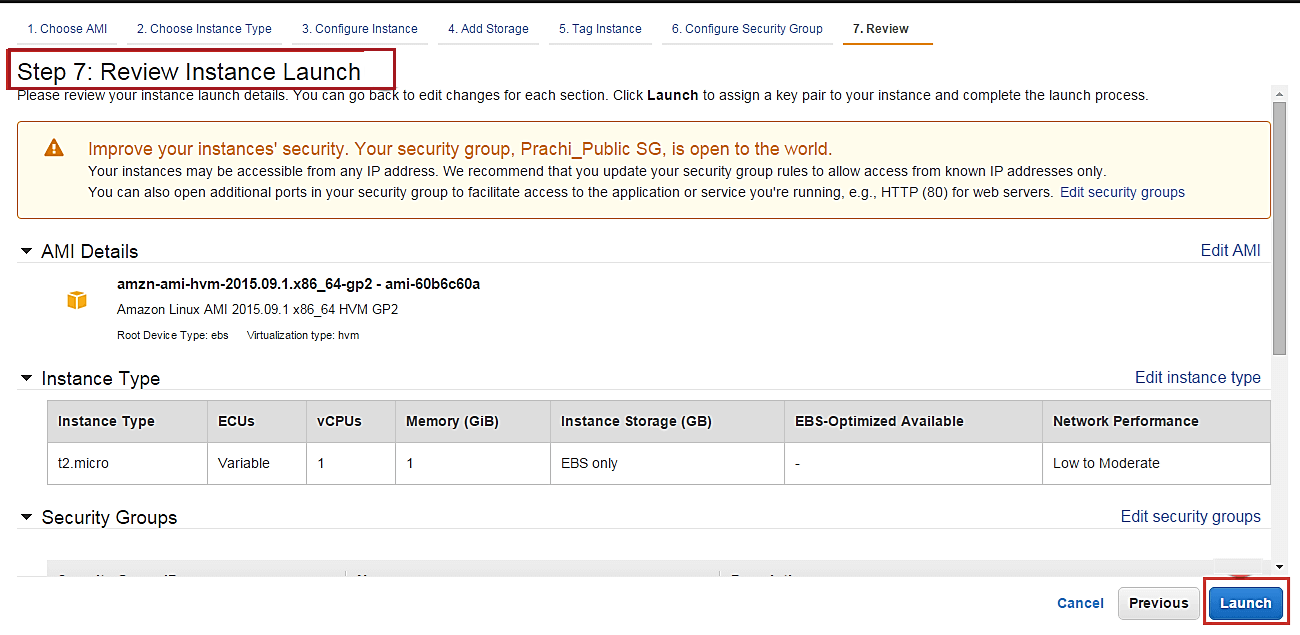
If you have a single EC2 instance running with a particular configuration, and you wish to quickly launch another instance in a one-click deployment, then 'Launch More Like This' option helps us do that.

**Step 1)** Click on 'Launch More Like This.'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth8.png)

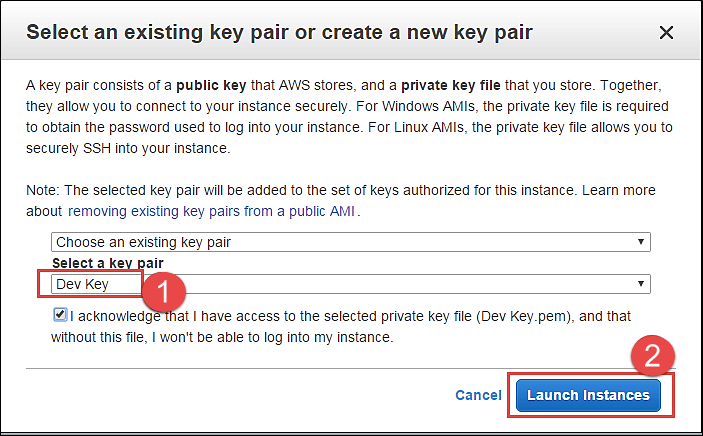
You will be straightaway directed to the review instance details page of the launch instance wizard. Here we can verify all the details once more.

**Step 2)** On review instance details page Click on button 'Launch.'

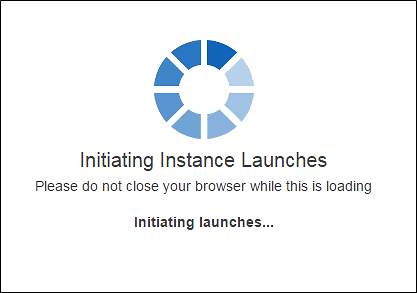
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth9.png)

**Step 3)** In this window,

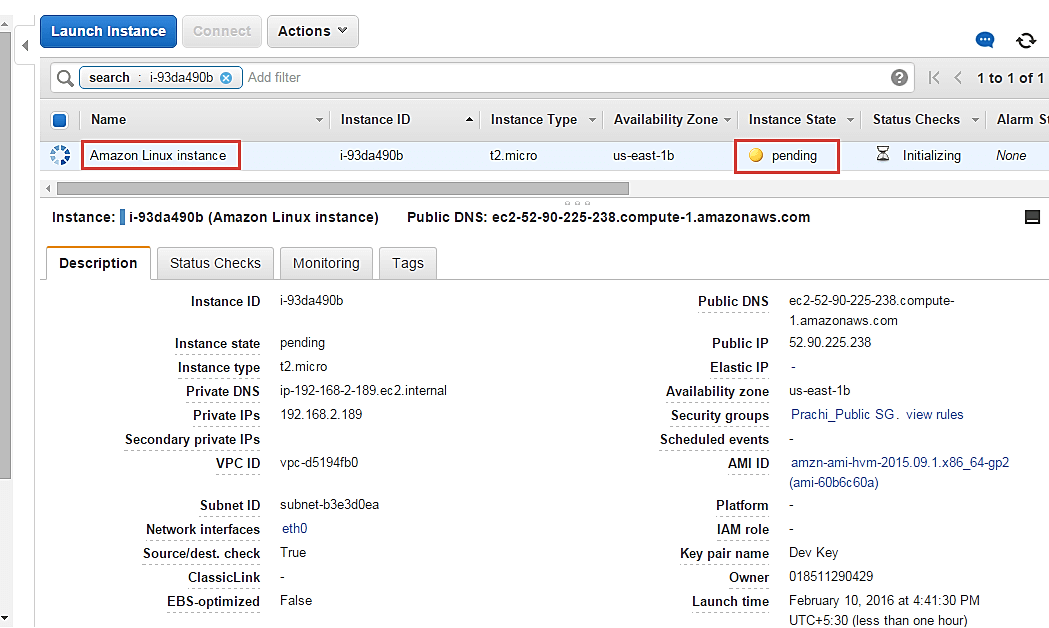
1. Select an existing key pair
2. Click on "Launch Instance."

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth10.png)

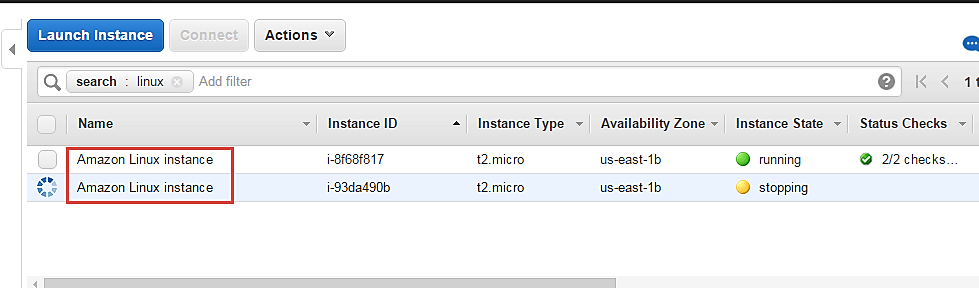
Instance launch progress can be seen as below.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth11.png)

You can see below that a new instance is in a pending state before creation.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth12.png)

You can see that the new instance has the same tag as well.

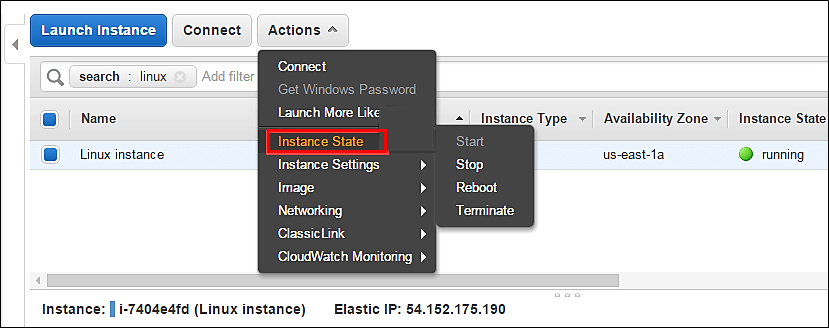
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth13.png)

### Changing the instance state

You can change the instance state on the fly from the Management Console on a single click.

**Step 1)** In this step, Click on 'Instance State' under actions.

* Stop – you can stop the running instance
* Reboot – you can reboot the instance
* Terminate – you can delete the instance permanently

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth14.png)

## Changing instance settings

Here you can change a lot of instance settings like security groups, termination protection, etc.

Let's us see each one in detail.

### Creating tags

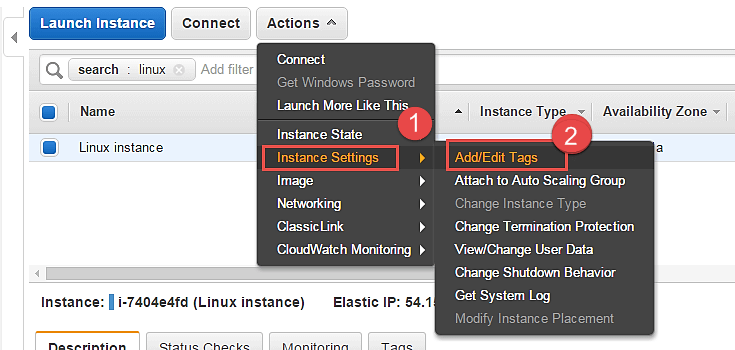
Add/Edit Tags – You can add or edit the tags assigned to the instance. Tagging makes it easier for the business owner of the AWS account to keep a track of the instances especially if there are multiple environments.

AWS admins should assign each instance a tag based on the segregation e.g.: tagging all the instances in the production environment as 'Prod' or tagging the instances belonging to a department with the department initials etc. Tagging is a very effective method to track the costing of the instances as well.

Let's see how to change tags

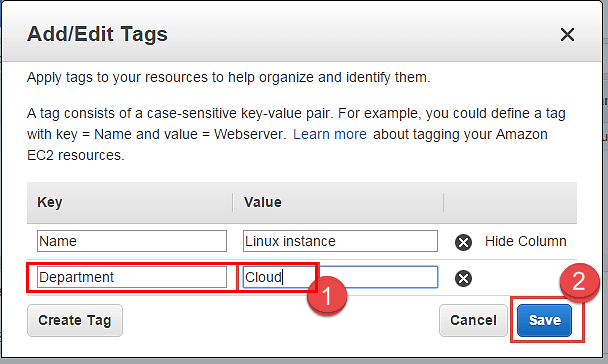
**Step 1)** In this step,

1. Click on instance setting
2. Click on 'Add/Edit Tags.'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth15.png)

**Step 2)** A tag is just a key-value pair.

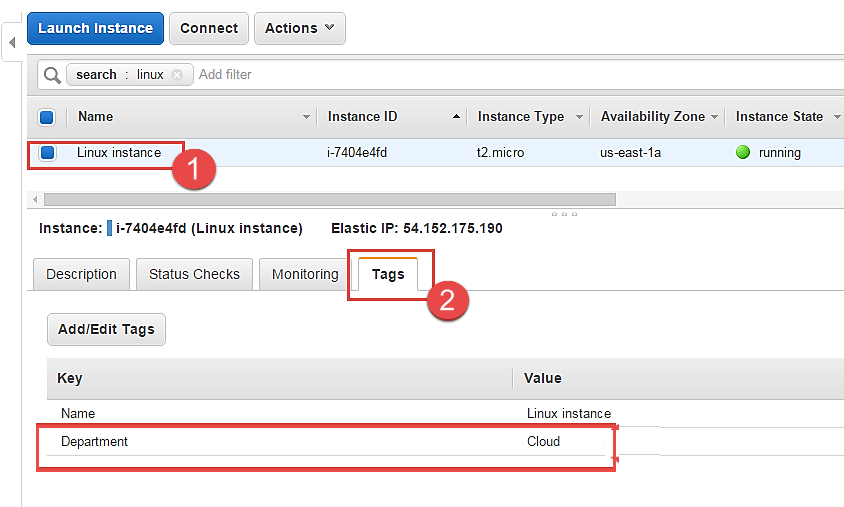
1. So we have assigned a new tag as Department and added its value as Cloud.
2. Click on Save

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth16.png)

**Step 3)** Come back to the EC2 Dashboard and

1. Select your instance again
2. Select the tab of 'Tags'

Note that the new tag as "Department" with value as Cloud has appeared under Tags.

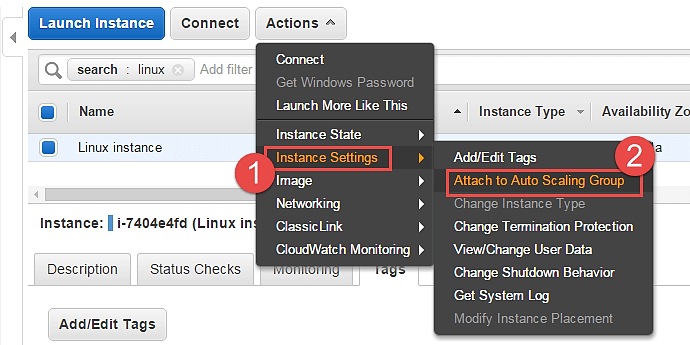
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth17.png)

### Attaching to Auto Scaling Group

An EC2 instance can be attached to an Auto Scaling Group on the fly.

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

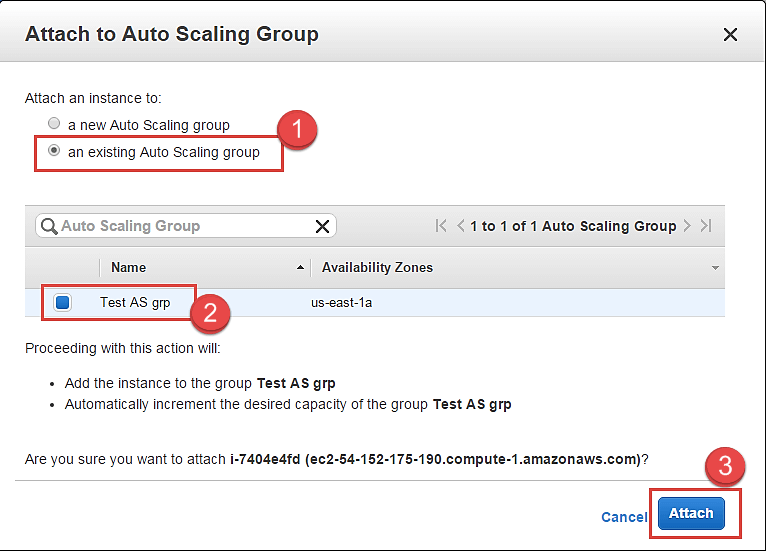
1. Click on 'Instance Settings'
2. Click on 'Attach to Auto Scaling Group.'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth18.png)

**Step 2)** In this step,

1. Attach an instance to an existing AS group. You can also create a new AS group in this step.
2. Select one AS group from the list of already existing groups.
3. Click on 'Attach'.

This action will attach your instance to an auto-scaling group in your environment.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth19.png)

### Changing instance type

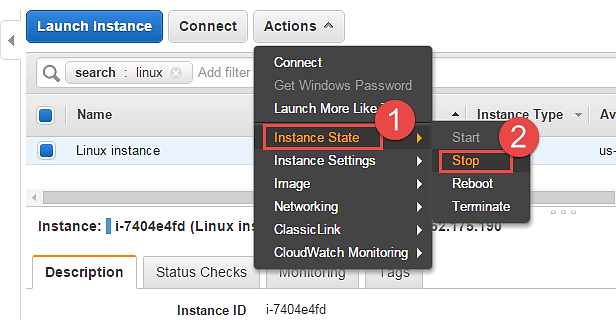
You can change the instance-type of your instance if you desire higher configuration instance as per your application requirement. This can be done to vertically scale your instance and provide you with more compute/memory capacity.

Let's see how to do this.

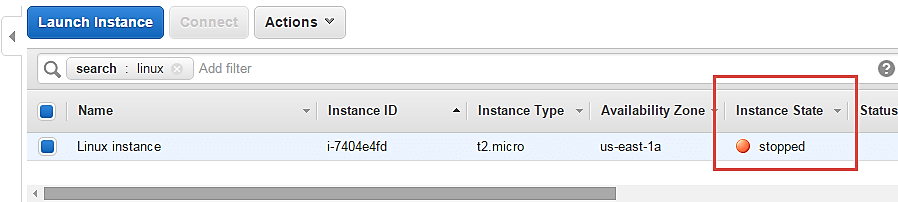
You cannot change an instance type if it's a running server. You have to stop it before doing so.

**Step 1)** In this step,

1. Go to 'Instance State'
2. Click on 'Stop'. This will stop the instance.

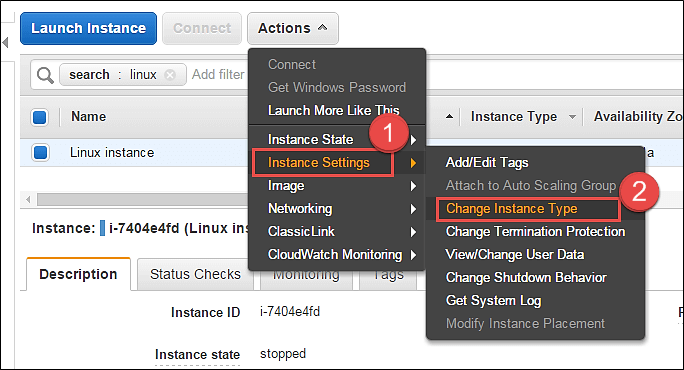
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth20.png)

Note that the instance state is now in "stopping" mode on the EC2 Dashboard. You change an instance type now.

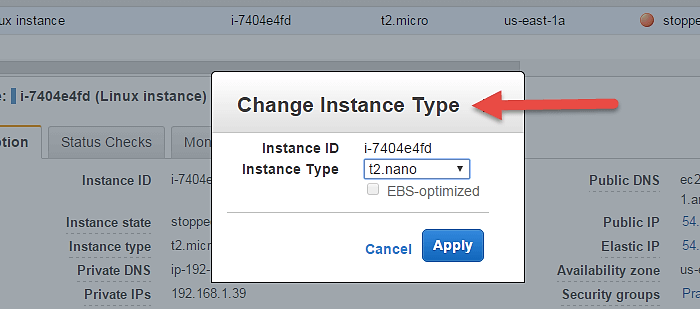
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth21.png)

**Step 2)** In this step,

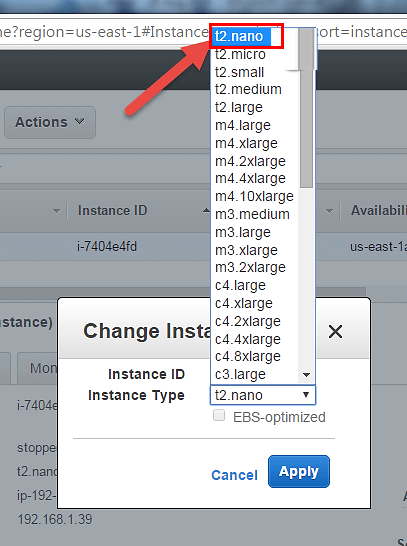
1. Go to 'Instance Settings'
2. Click on 'Change Instance Type'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth22.png)

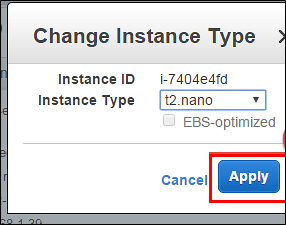
A Change Instance Type pop-up will appear.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth23.png)

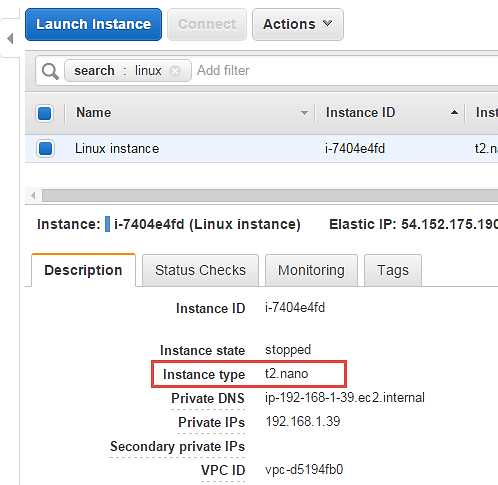
**Step 3)** You can select from a range of EC2 available instance types. For this tutorial, we are changing it to t2.nano just for the sake of demonstration.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth24.png)

**Step 4)** Select t2.nano and hit 'Apply'.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth25.png)

Notice on the EC2 Dashboard, your instance type has been changed to the said type automatically.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth26.png)

You can now start your instance and continue on the operations on that. There will be no change in other configuration parameters and also your existing installations on the server will remain intact.

### Enabling termination protection

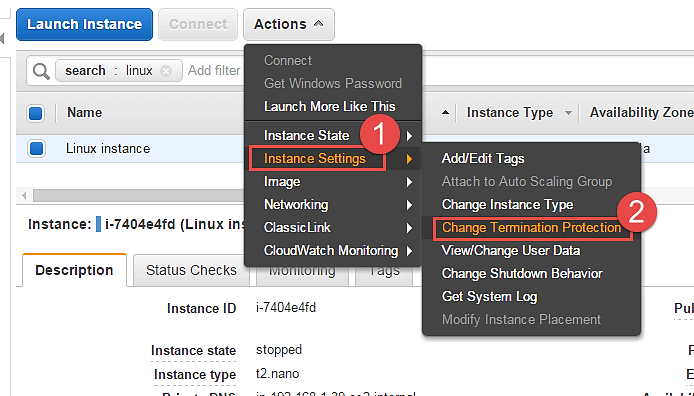
An instance should always have termination protection enabled especially on production servers. This will ensure that your EC2 instance is not getting accidently terminated.

AWS will add an additional level of security in case you happen to accidently hit the instance terminate option.

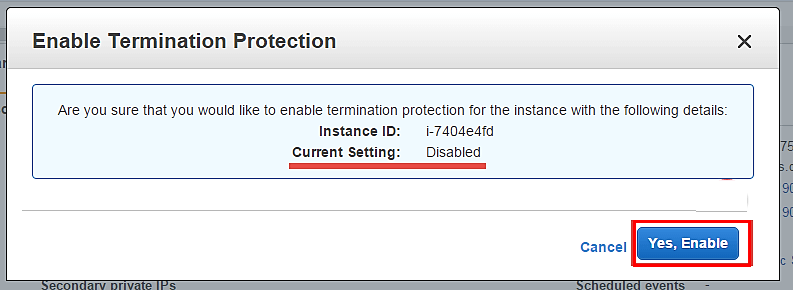
Let's see how to enable termination protection.

**Step 1)** In this step,

1. Go to 'Instance Settings.'
2. Click on 'Change Termination Protection.'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth27.png)

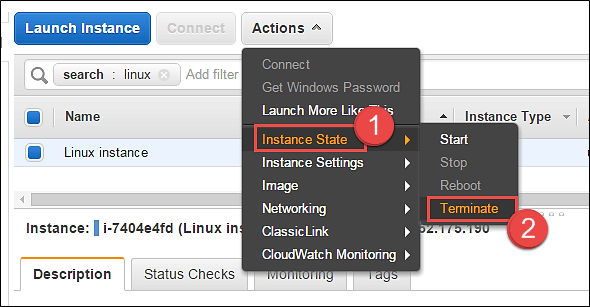
**Step 2)** Notice that the current setting on our instance is disabled. Click on "Yes,Enable".

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth28.png)

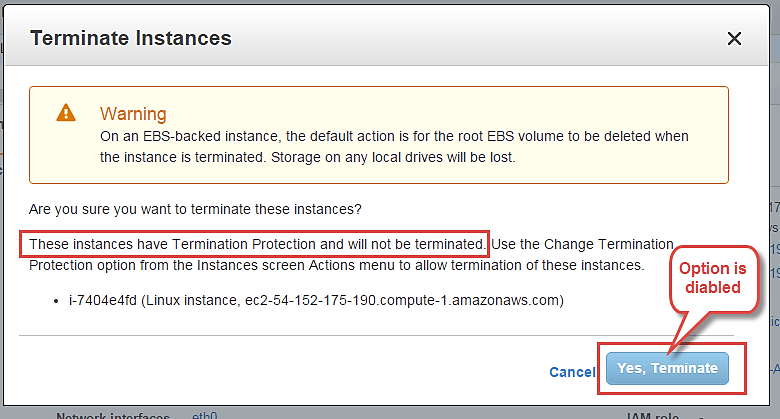
This has enabled Termination protection on our instance. We'll check to see if our instance gets deleted when we hit Terminate.

**Step 3)** In this step,

1. Select option 'Instance State' and then
2. Click on 'Terminate.'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth29.png)

AWS will immediately notify you that the EC2 instance has "termination protection enabled" and you will not be able to delete it. The 'Terminate' button below is disabled.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth30.png)

### Changing User Data

When you launch a new EC2 instance, you have the option to pass user data to an instance to run tasks at boot time automatically e.g. common configuration tasks, init scripts, etc.

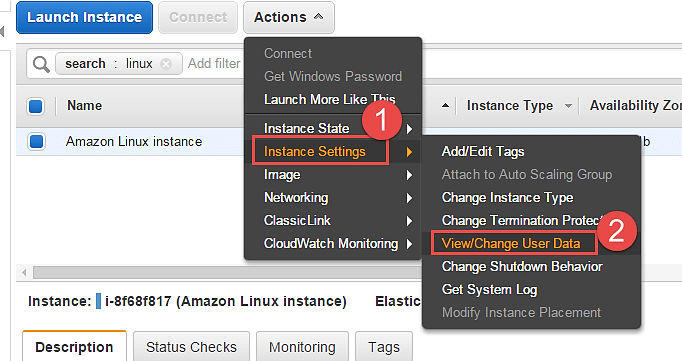
You can pass the user data in the form of shell scripts or cloud-init directives. This can be either plain text, as a file or as base64 encoded text for API calls.

Here we will see how we can edit these scripts.

You will have to stop the instance first, you will not be able to edit the instance userdata if it's running. On a stopped instance, perform below steps.

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

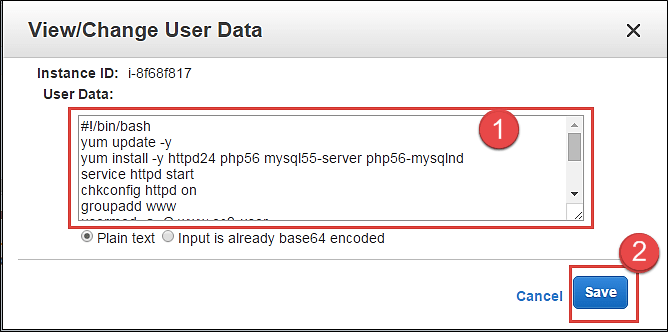
1. Go to 'Instance Settings'.
2. Click on 'View/Change User Data'.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth31.png)

Here for the purpose of demonstration, we have a shell script which installs LAMP stack on the server.

**Step 2)** In this step,

1. View/ modify your user data field.
2. Click on "Save" tab.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth32.png)

### Changing the shutdown behavior

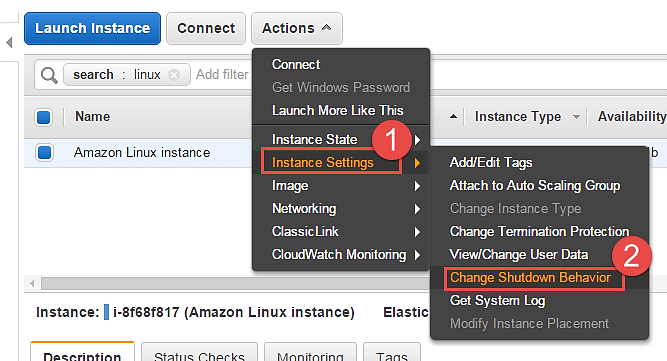
If ever you have accidently shutdown the instance via the OS console, you don't want AWS EC2 to actually terminate the instance.

For that, we can set up the shutdown behavior as 'Stop' instead of 'Terminate'. We can also do vice versa if the application requirement is as such.

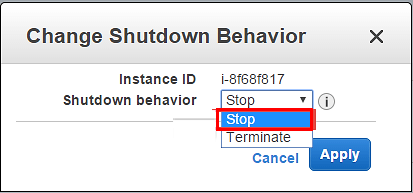
Let's see how to achieve this.

**Step 1)** In this step,

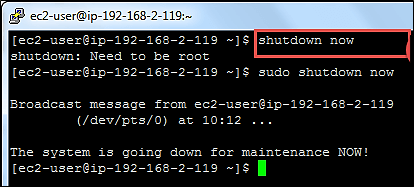
1. Go to 'Instance Settings'.
2. Click on 'Change Shutdown Behavior.'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth33.png)

**Step 2)** In this step, click on 'Stop' and then hit apply. The setting will be applied to the instance accordingly.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth34.png)

**Step 3)** Now when "stop" shutdown is initiated in the instance console via putty, it will not get terminated. It will simply shutdown normally.

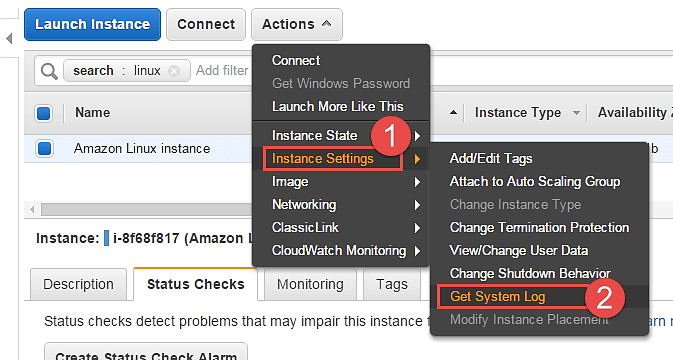
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth35.png)

### Viewing System Log

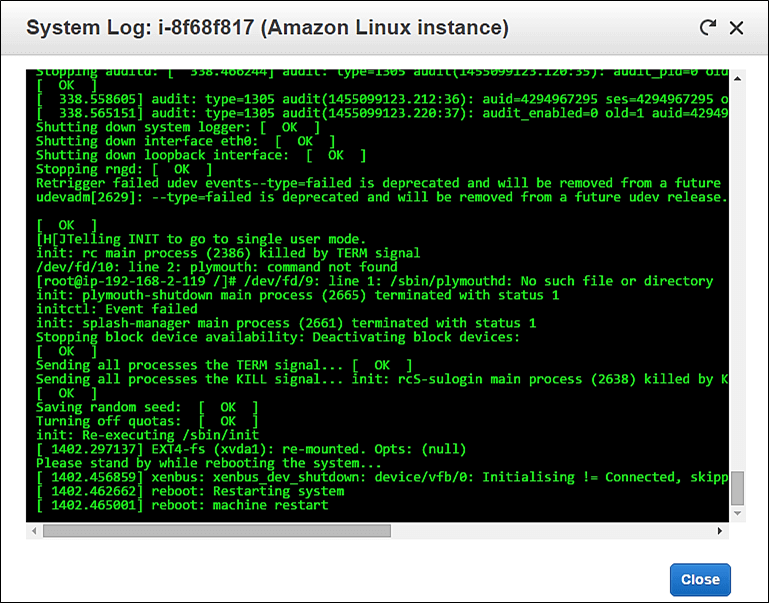
You can see the system log for any EC2 instance for troubleshooting purposes etc.

**Step 1)** In this step,

1. Go to 'Instance Settings'.
2. Click on 'Get System Log'.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth36.png)

You can see a separate window depicting the instance log details. Here we can see a snap of log when the instance was restarted.

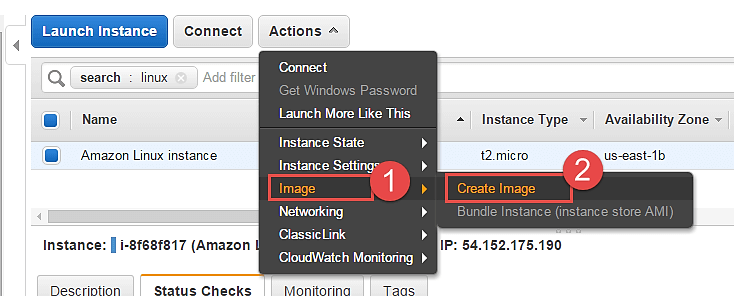
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth37.png)

## Creating an instance AMI

You can create an AMI of your EC2 instance for backup.

**Step 1)** In this step,

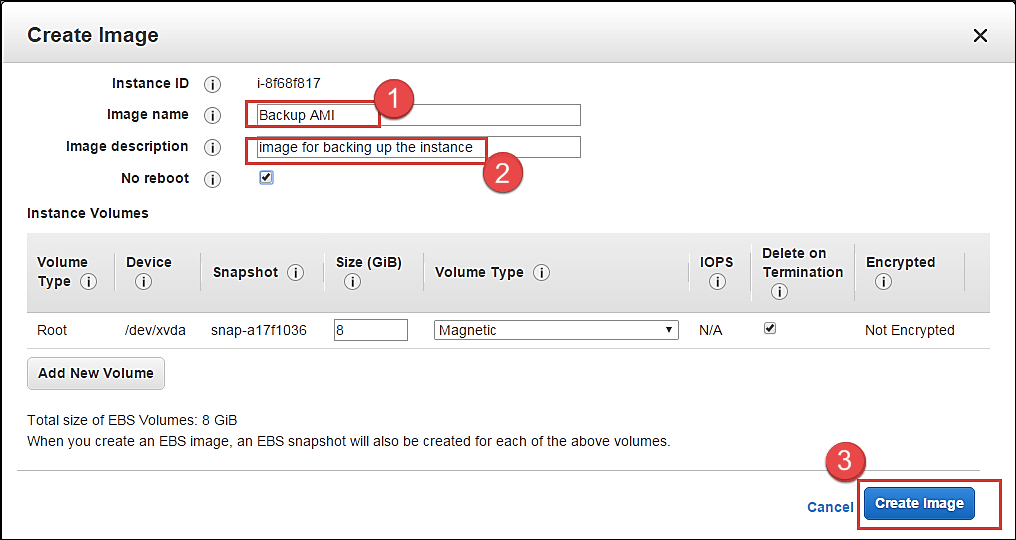
1. Go to 'Image'.
2. Click on 'Create Image'.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth38.png)

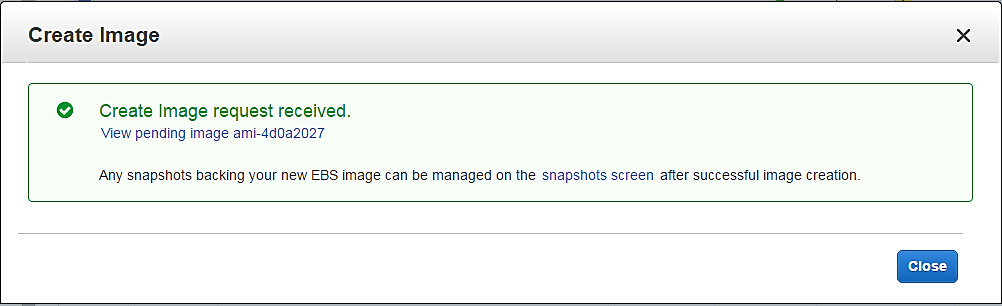
An image creation wizard will open.

**Step 2)** In this step,

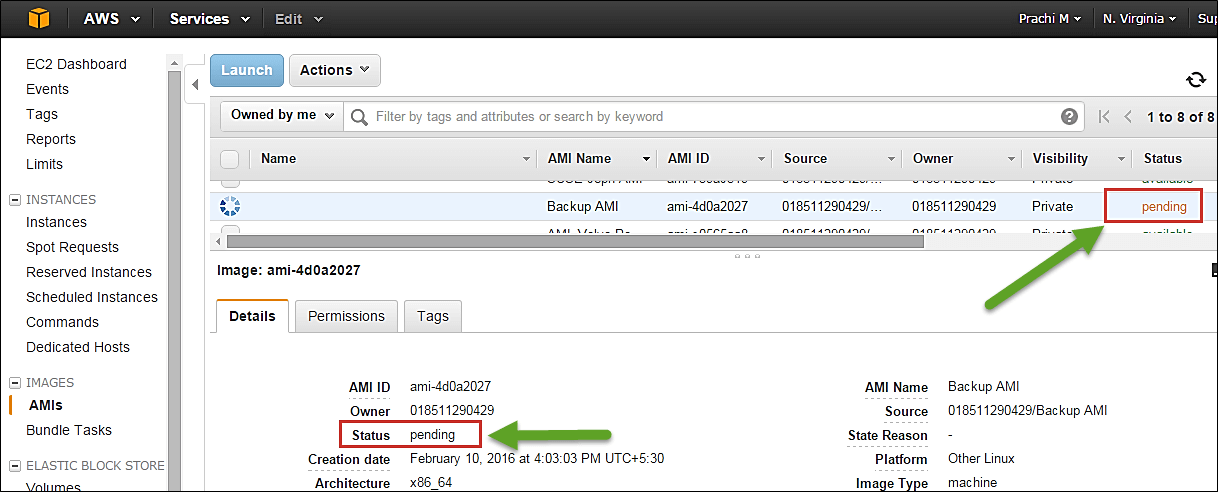
1. Add the image name
2. Give some friendly description for the AMI
3. Check the volumes and then hit 'Create Image' button.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth39.png)

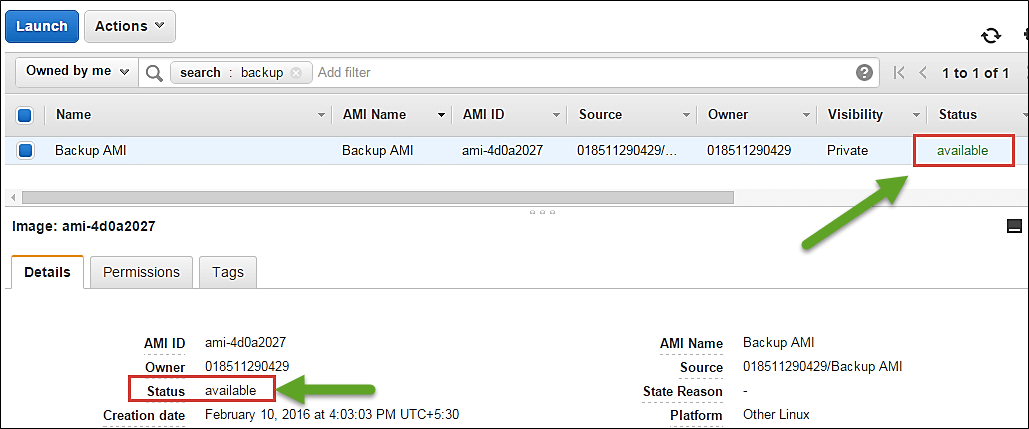
AWS will receive your create image request and will send a notification immediately.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth40.png)

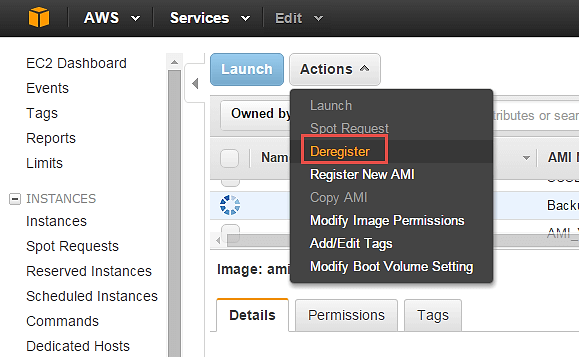
You can check the status of the request on the EC2 dashboard as 'pending' just like what is shown below.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth41.png)

After a while the status is "available" and you will have your AMI ready as a backup.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth42.png)

You can also de-register it from the dashboard once the backup is old.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth43.png)

## Changing the instance network settings

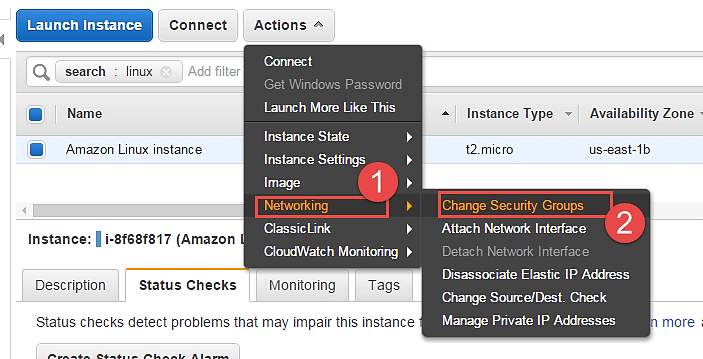
### Changing the Security Group

You can change the SG (Security Group) of an instance anytime. If you have another security group with different firewall rules, you can easily do so using the console.

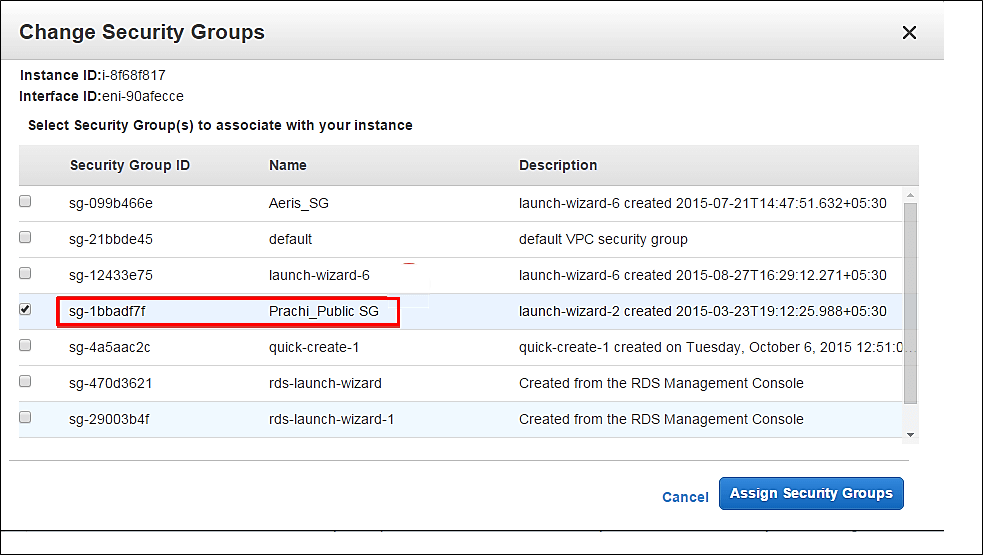
Let's see how.

**Step 1**) In this step,

1. Go to 'Networking'.
2. Click on 'Change Security Groups'.

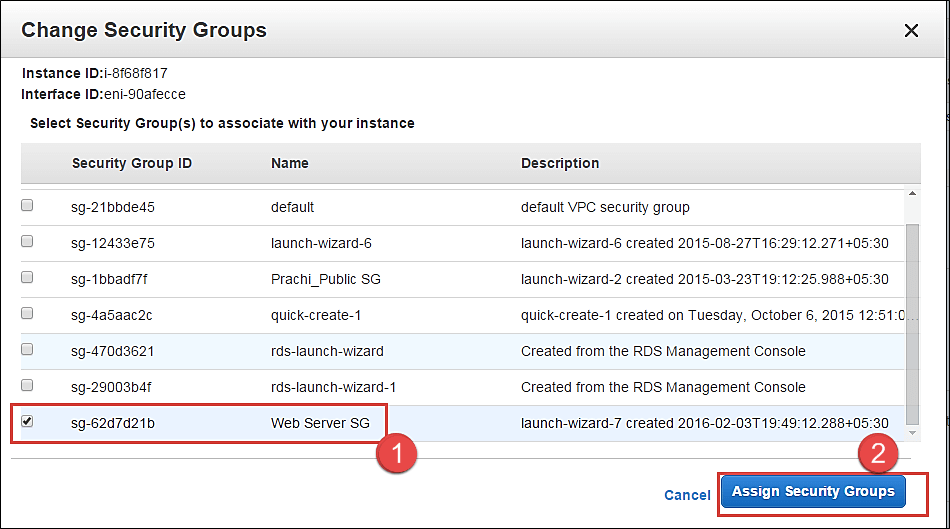
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth44.png)

**Step 2)** In the change security groups wizard, it will show the already existing SG on the instance along with a list of all the security groups in the region.

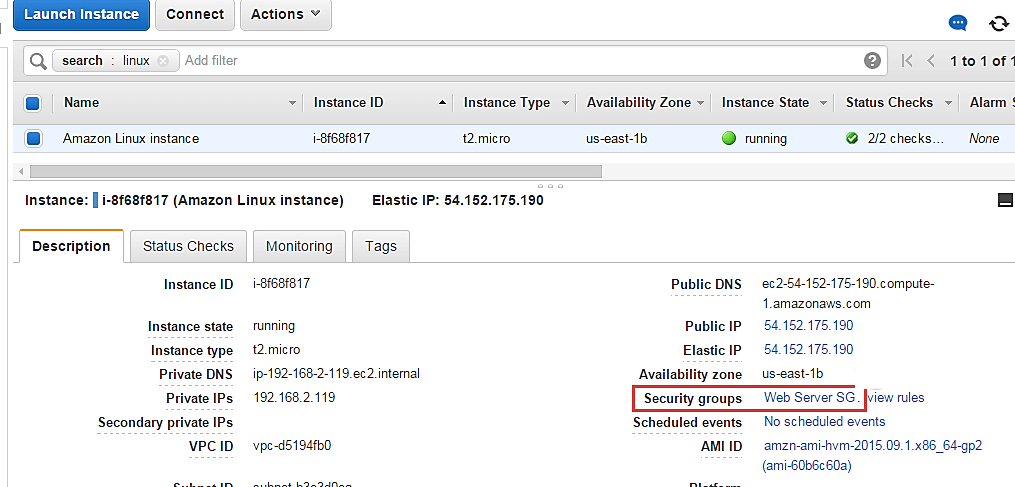
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth45.png)

**Step 3)** In this step,

1. Tick the box against your desired SG
2. Click on 'Assign Security Groups' button.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth46.png)

**Step 4)** On the EC2 Dashboard, you can see that the SG of the instance has been changed. The instance will now send/receive traffic based on the new SG settings.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth47.png)

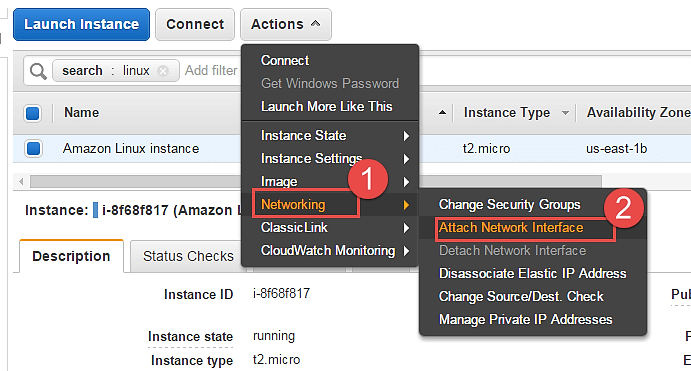
You can also add multiple security groups.

### Adding a Network Interface

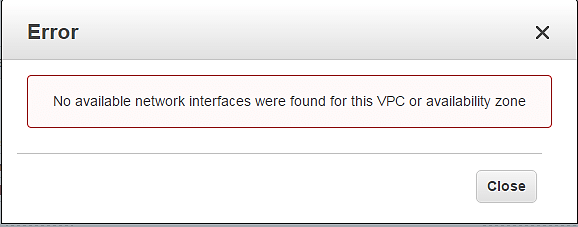
A network interface is like another NIC card to an instance. It will have another set of IPs additional to the already existing primary Network Interface.

**Step 1)** In this step,

1. Go to 'Networking'.
2. Click on 'Attach Network Interface'.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth48.png)

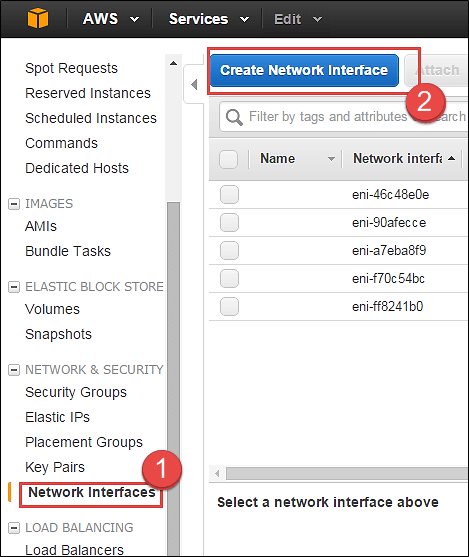
You will get an error prompt if you don't have a Network Interface already created.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth49.png)

Let's see how to create a Network Interface quickly.

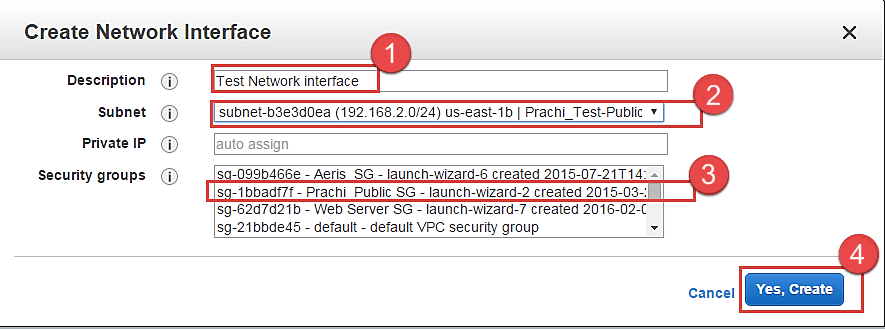
**Step 2)** In this step,

1. Go to EC2 Dashboard, and click on 'Network Interfaces' on the left pane.
2. Click on 'Create Network Interface' button.

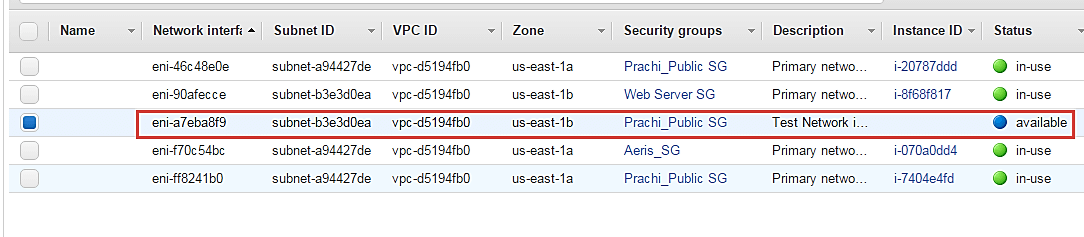
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth50.png)

**Step 3)** In this step,

1. Add a description for your network interface
2. Select the subnet where you want to create your network interface. Keep the auto assign the private IP option default
3. Security groups are applied to a network interface of an instance, so here you will get an option for the same. Select your desired SG
4. Once you're done entering the details, click on 'Create.'

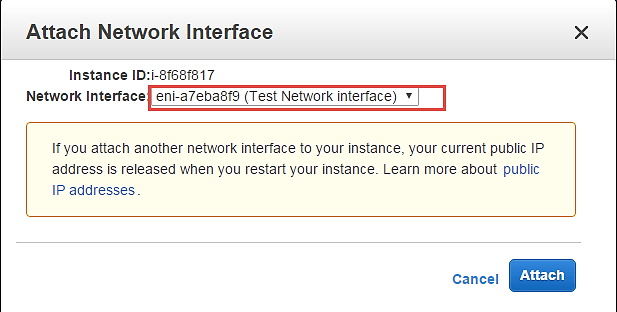
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth51.png)

Now you can come back to the EC2 Dashboard and check that your network interface is getting created.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth52.png)

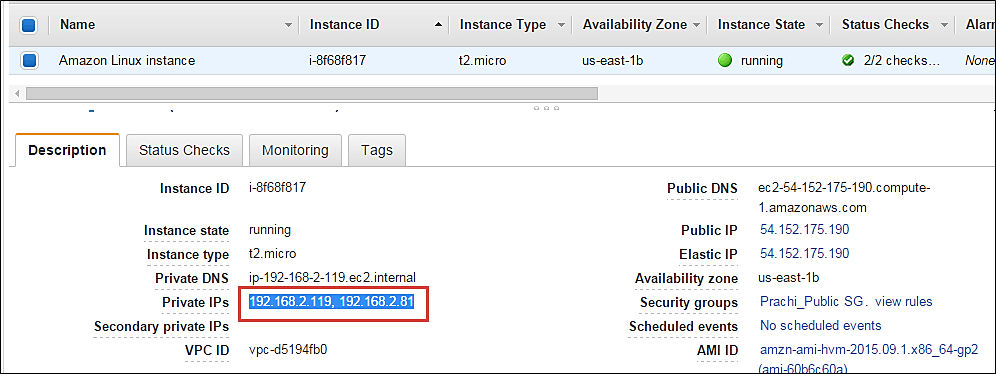
Now come back on Step 2) and go ahead with selecting your available interface which we just created and attach it to the instance.

Now as you can see the network interface which we just created is enlisted below automatically.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth53.png)

Your network interface will be attached to the instance immediately.

We can come back to the EC2 Dashboard and check our instance now. Note that the instance has 2 private IPs belonging to 2 network interfaces.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth54.png)

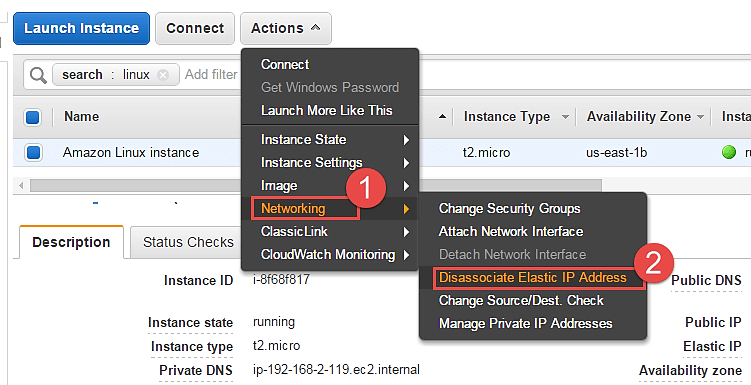
### Dissociating EIP

An Elastic IP is a static Public IP.

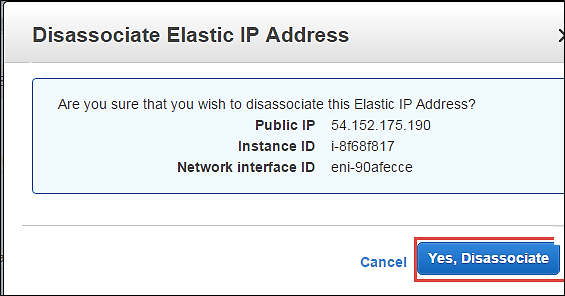
You can dissociate an EIP directly from the instance dashboard.

**Step 1)** In this step

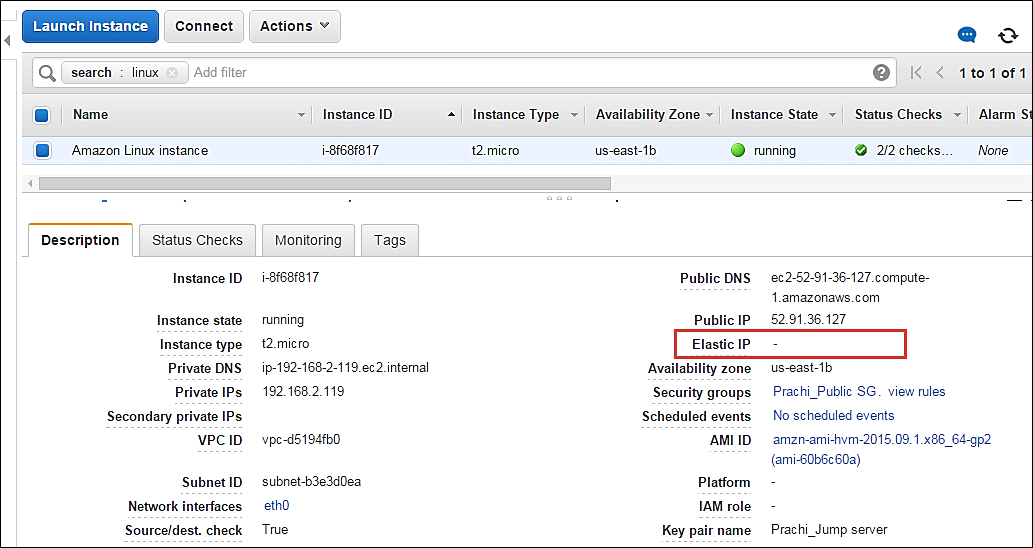
1. Click on 'Networking.'
2. Click on 'Dissociate Elastic IP Address.'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth55.png)

**Step 2)** Click on the button of dissociate, once we have verified the instance id and the EIP.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth56.png)

Check below that the instance dashboard now shows the EIP field blank.

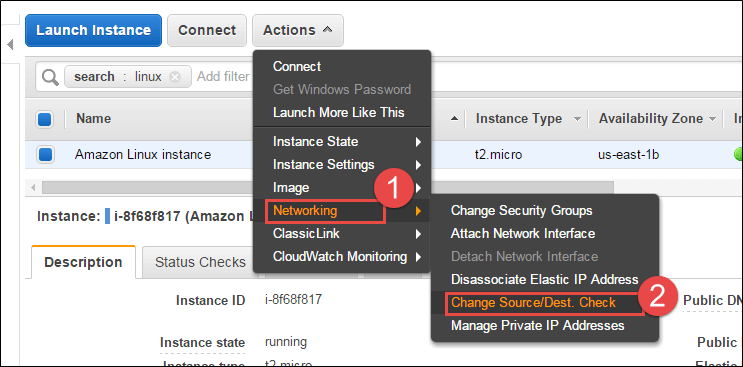
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth57.png)

### Changing Source/Destination check

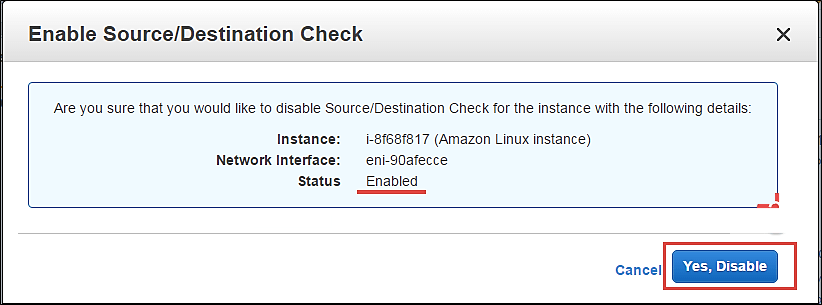
The Source/Destination Check attribute controls whether source/destination checking is enabled on the instance. Disabling this attribute enables an instance to handle network traffic that isn't specifically destined for the instance. For example, instances running services such as network address translation, routing, or a firewall should set this value to disabled.

**Step 1)** In this step,

1. Click on 'Networking.'
2. Click on 'change Source/Dust. Check'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth58.png)

**Step 2)** Click on 'Disable'. If it is disabled already, you can enable it in this step.

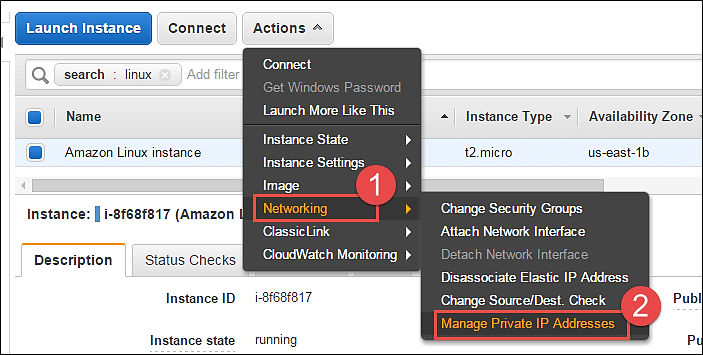
[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth59.png)

### Managing private IP addresses

You can assign multiple private IP addresses to a single instance if that is your application architecture's design. The maximum no of IPs you can assign of course depends on the EC2 instance type.

**Step 1)** In this step,

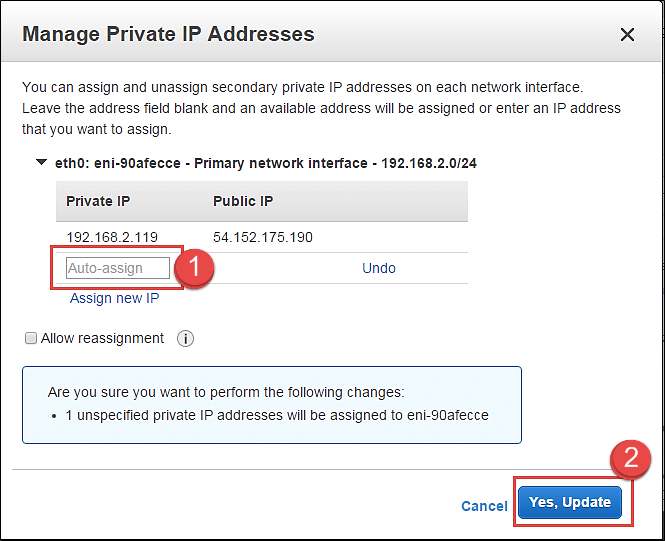
1. Click on 'Networking.'
2. Click on 'Manage Private IP addresses.'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth60.png)

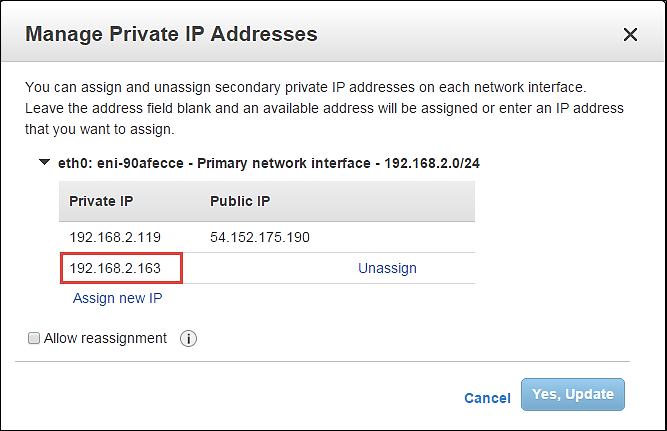
You will be redirected to a new window to assign a secondary IP address to your instance.

**Step 2)** In this step,

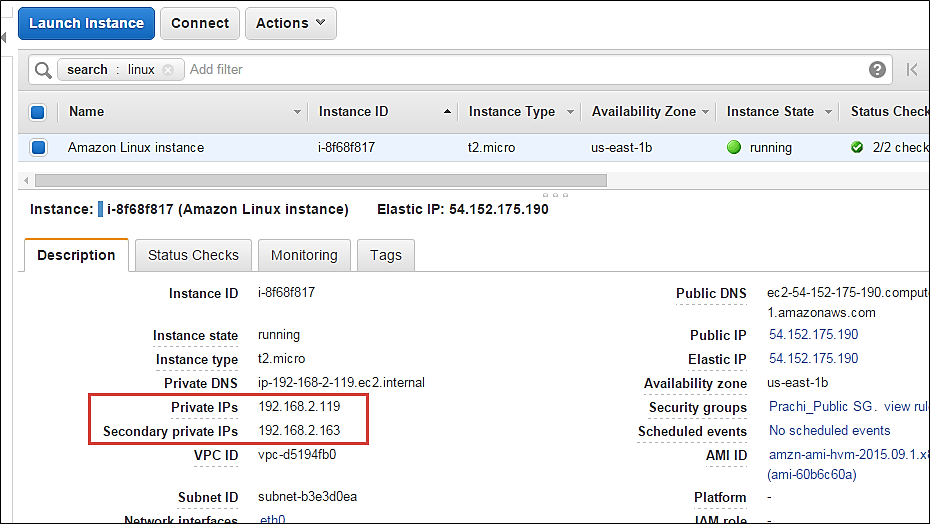
1. Here we are leaving the field blank. This will enable AWS to auto-assign any available private IP to our instance.
2. Click on 'Update.'

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth61.png)

Note that an IP has been automatically assigned here.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth62.png)

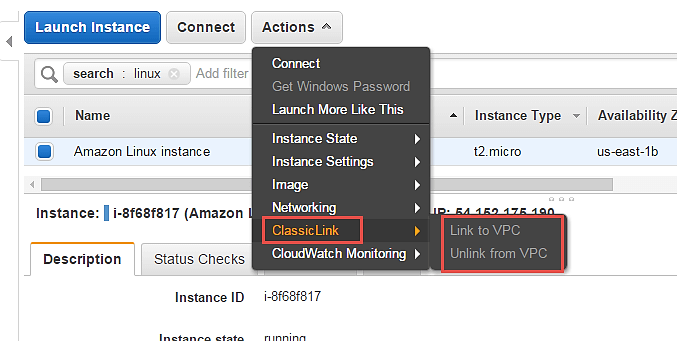
Also, come back to the EC2 dashboard and notice the 2 private IPs assigned. These are 2 IPs on a single network interface.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth63.png)

### Enabling/disabling ClassicLink to a VPC

If your instance is provisioned in EC2 – Classic, which is a deployment mode in AWS where resources are provisioned out of a VPC; then you can link your instance to a VPC environment as shown below.

The options below are disabled for us as our instance is already in a VPC.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth64.png)

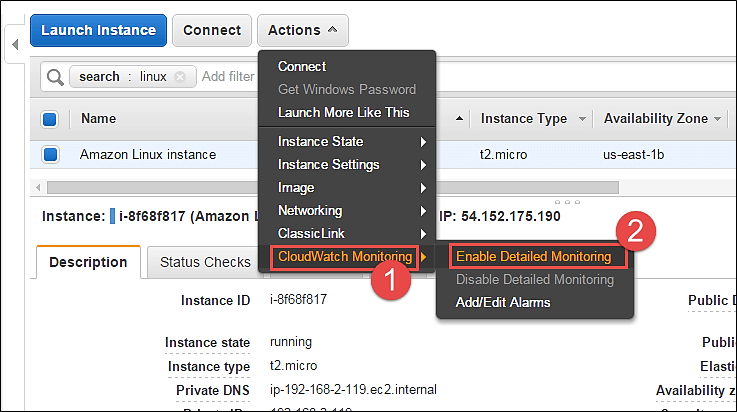
### Enabling detailed CloudWatch monitoring

AWS will by default have basic CloudWatch monitoring enabled on all its resources. However, if our instances are production instances, we may wish to enable detailed monitoring on them with additional costs of course.

**Step 1)** In this step,

1. Click on 'CloudWatch Monitoring'
2. Click on 'Enable Detailed Monitoring'

You can also add/edit alarms to alert you for attributes in your CloudWatch monitoring metrics.

[](https://cdn.guru99.com/images/3-2016/032816_0542_Modifyingth65.png)

**Summary**

Thus, we saw in this tutorial, how to enable/modify various attributes in AWS for the instance configuration from the Management Console after it is launched.

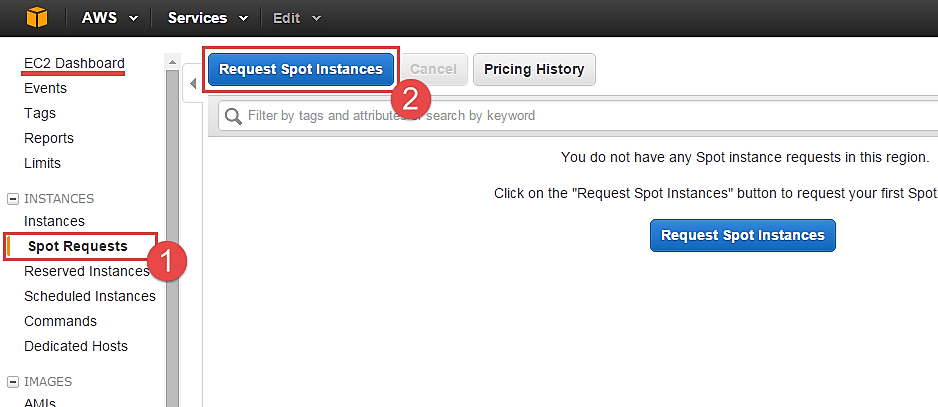
AWS provides many more configuration options via CLI/API.

### Creating a Spot Request

In order to launch a spot instance, you have to first create a Spot Request.

Follow the steps below to create a Spot Request.

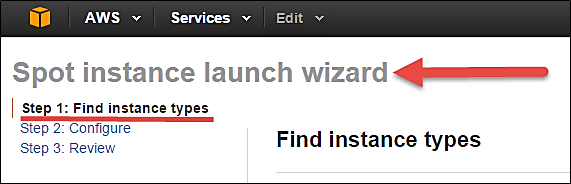
1. On the EC2 Dashboard select 'Spot Requests' from the left pane under Instances.
2. Click on the button 'Request Spot Instances" as shown below.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp4.png)

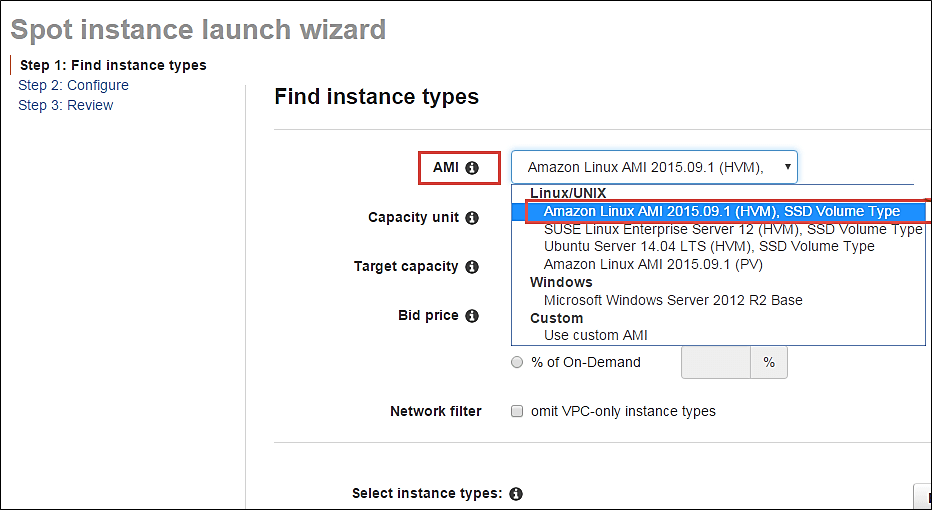
Spot instance launch wizard will open up. You can now go ahead with selecting the parameters and the instance configuration.

#### Find Instance Types

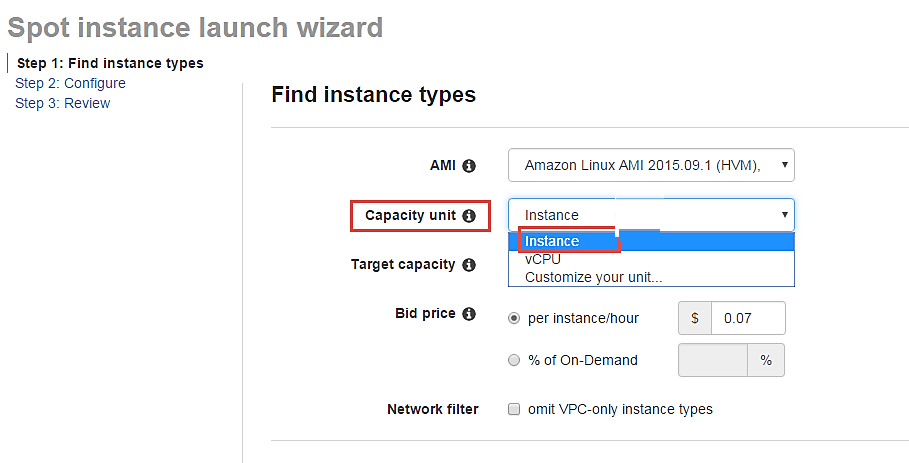
The first step for spot instance is to "Find instance types."

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp5.png)

**Step 1)** Select an AMI- an AMI is a template consisting of the OS platform and software to be installed in the instance. Select your desired AMI from the existing list. We are selecting Amazon Linux AMI for this tutorial.

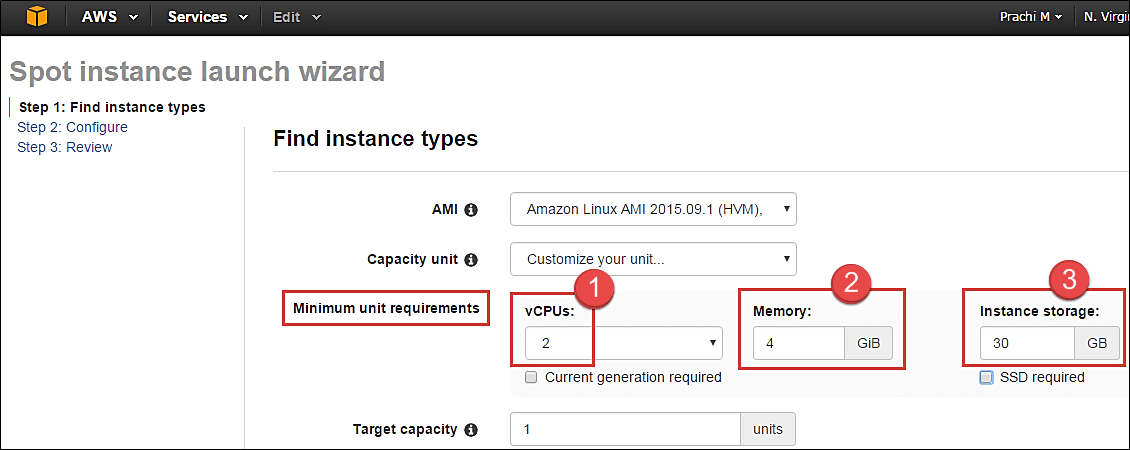
[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp6.png)

**Step 2)** Capacity Unit- a Capacity Unit is your application requirement. You may decide to launch an instance based on the instance type, vCPU or custom configuration like your choice of vCPU/memory/storage requirements. Here we are selecting an Instance.

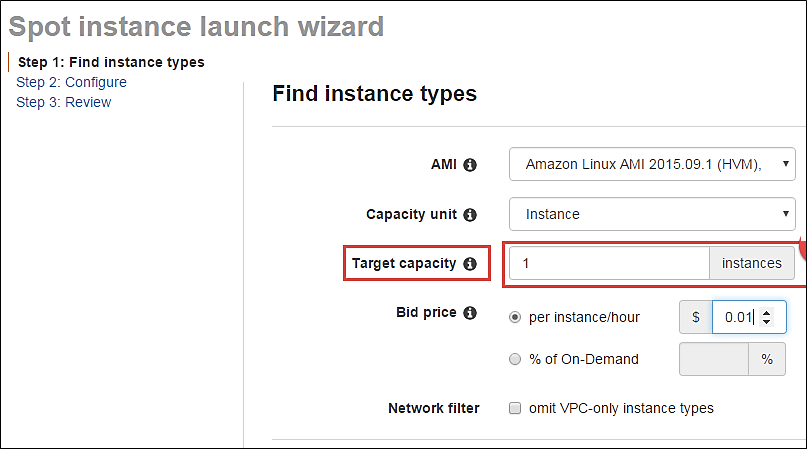
[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp7.png)

If you wish to customize the capacity, you can add your choice of

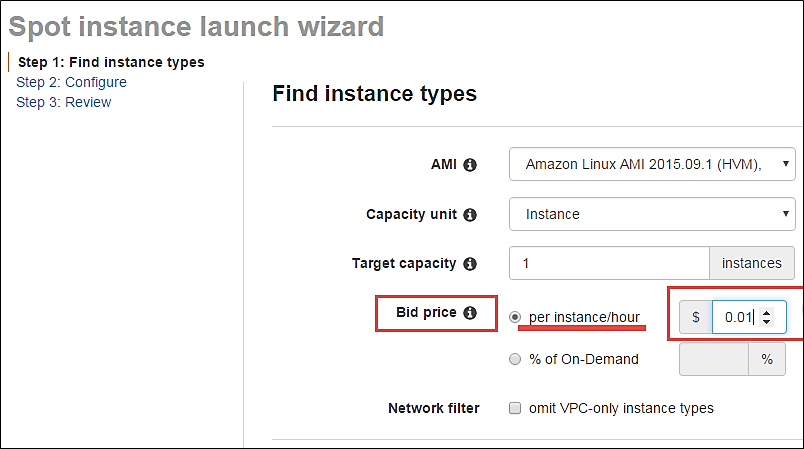
1. vCPU,
2. Memory and
3. Instance storage as below.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp8.png)

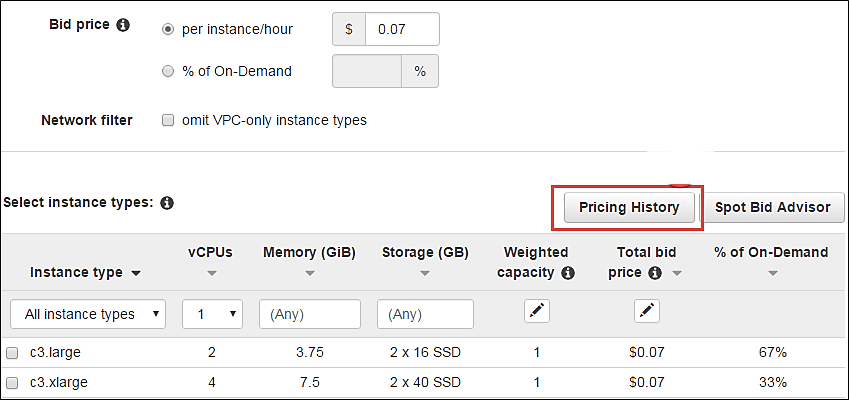
**Step 3)** Target Capacity depicts how many spot instances you wish to maintain in your request. Here we are selecting one.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp9.png)

**Step 4)** Bid Price – this is the maximum price we are ready to pay for the instance. We are going to set a particular price per instance/hour. This is the simplest to calculate based on our business requirement. We will see ahead how we should determine the bid price so that our bid price always remains high and doesn't exceed the spot price so that our instance keeps running.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp10.png)

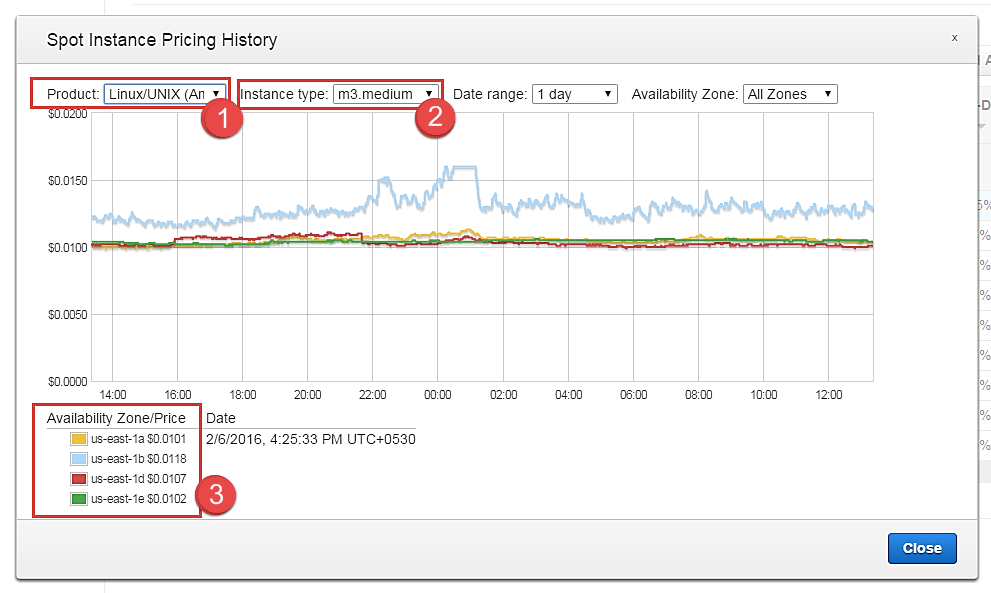
**j**ust below the bid price you can see a button of Pricing History. Click on that as shown below.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp11.png)

Here in Pricing History, we can see a graph depicting instance pricing trends with historical data. You can select the parameters and get an idea of the pricing of our desired instance over a period of time.

1. Select the product. We have selected our Linux AMI.
2. Select the instance type. We have selected m3.medium.
3. Note the average prices for over a day here.

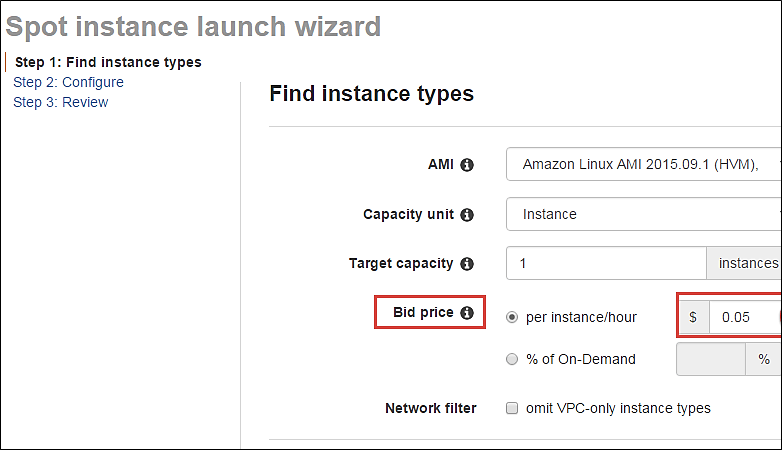
Thus, from the chart below, we can see that the instance type that we are planning to provision lies in the pricing range of $0.01xx, and it seems that Availability Zone 'us-east 1a' has the lowest price.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp12.png)

cont. to step 4.

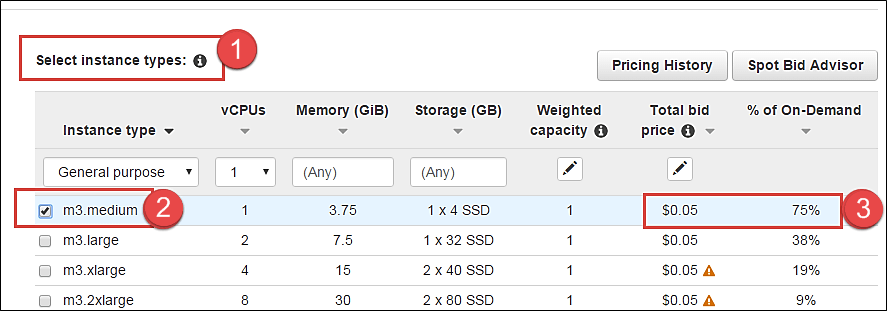
So let's come back to our step of quoting a bid price.

For the sake of maintaining our instance always available and if it falls within our budget, we can quote a higher bid price. Here we have quoted a slightly higher price of $0.05.

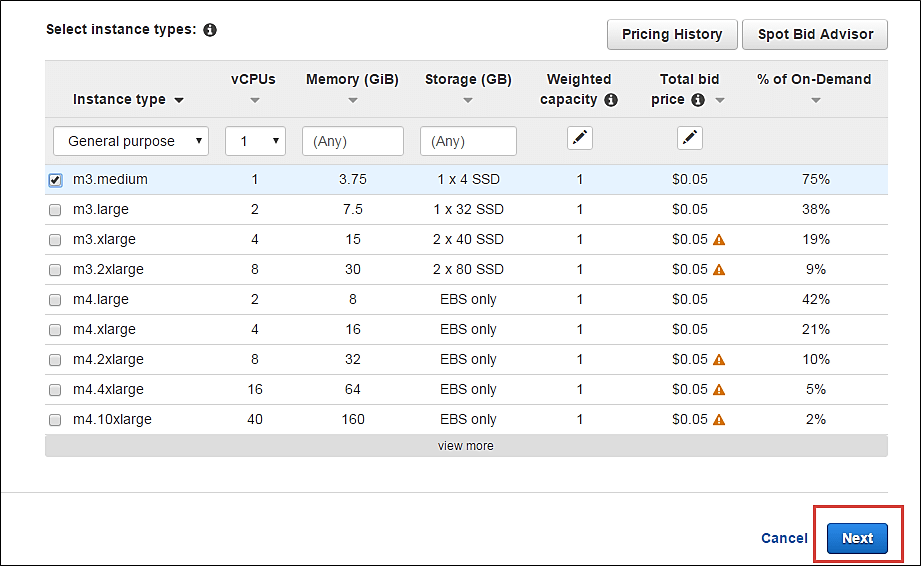
[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp13.png)

You can see some trends in the wizard itself.

1. Note the instance types section
2. Select the instance type that we are planning to provision
3. Note the price that we are planning to bid. % of on-demand shows us that our quoted price is 75% of the on-demand price for the same instance type. This means we are saving 25% per hour as compared to an on-demand instance. You can further lower the price and save costs drastically.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp14.png)

**Step 5)** Once we are done looking at the trends and quoting our bid price, click on next.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp15.png)

#### Configure the Spot instance

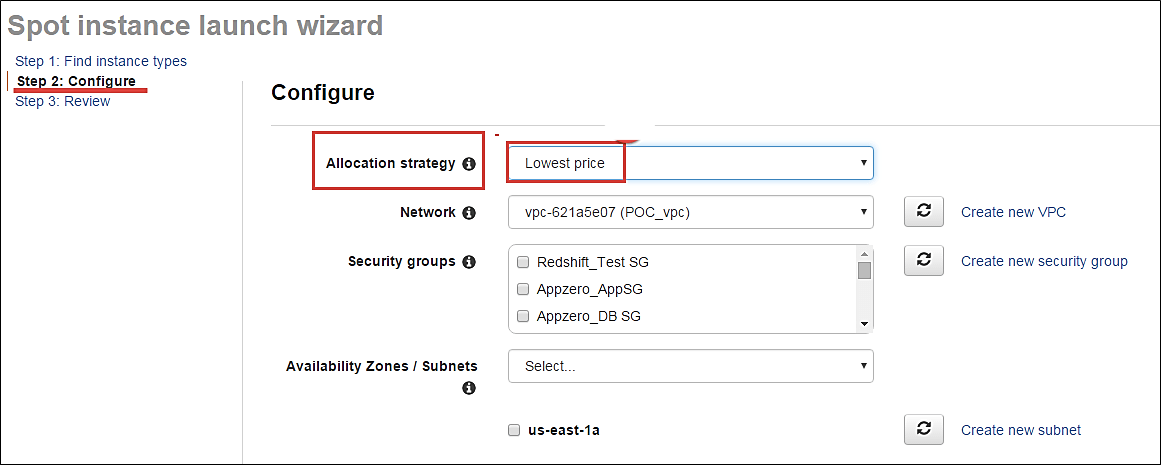
Our next step is to configure the instance, in this step of the wizard, we'll configure instance parameters like VPC, subnets, etc.

Let's take a look.

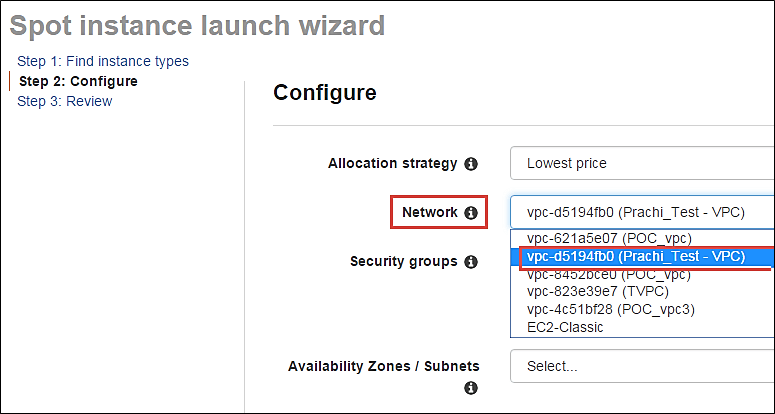
**Step 1)** Allocation Strategy – it determines how your spot request is fulfilled from the AWS's spot pools. There are two types of strategies:

* Diversified – here, spot instances are balanced across all the spot pools
* Lowest price – here, spot instances are launched from the pool which has lowest price offers

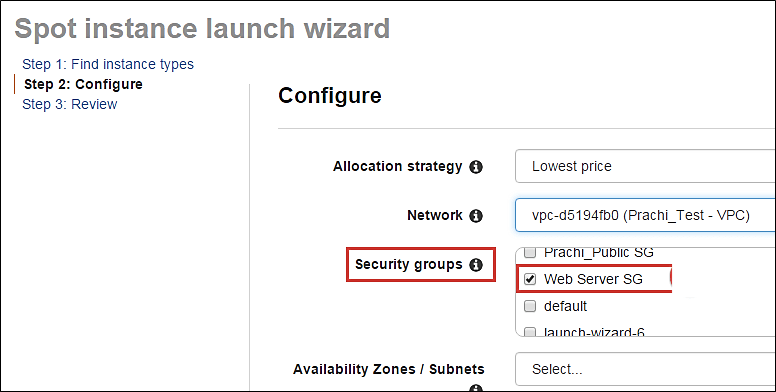
For this tutorial, we'll select Lowest Price as our allocation strategy.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp16.png)

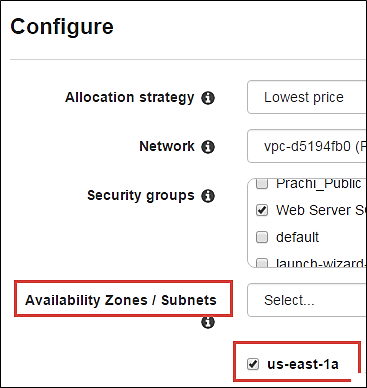
**Step 2)** Select the VPC- we'll select from the list of available VPCs that we have created earlier. We can also create a new VPC in this step.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp17.png)

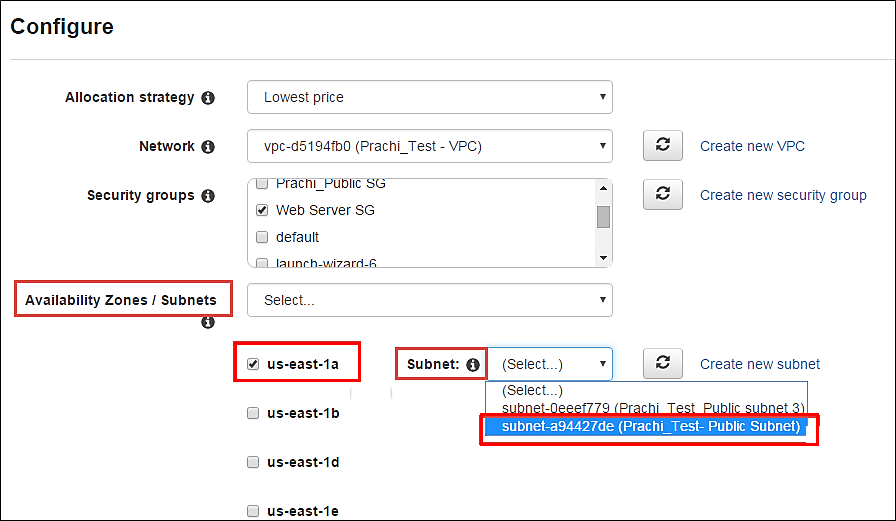
**Step 3)** Next we'll select the security group for the instance. We can select an already existing SG or create a new one.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp18.png)

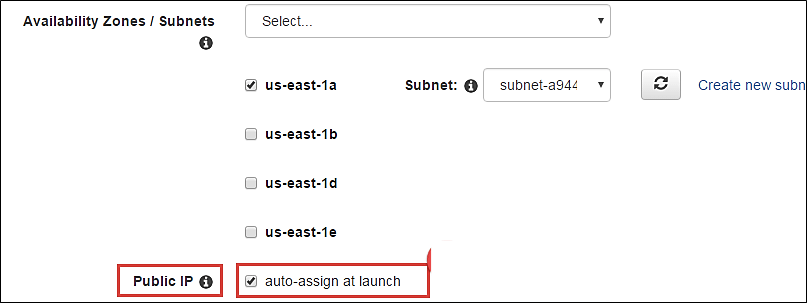
**Step 4)** Availability Zone- we'll select the AZ where we want to place our instance based on our application architecture. We are selecting AZ- us-east-1a.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp19.png)

**Step 5)** Subnets- we are going to select the subnet from our list of already available list.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp20.png)

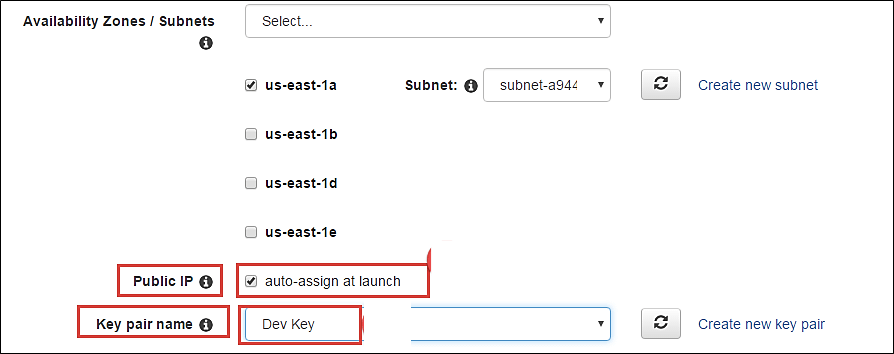
**Step 6)** Public IP- we'll choose to assign the instance a public IP as soon as it launches. In this step, you can choose if you want AWS to assign it an IP automatically, or you want to do it manually later. You can enable/ disable 'Auto assign Public IP' feature here likewise.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp21.png)

**Step 7)** Key pair- A key pair is a set of public-private keys.

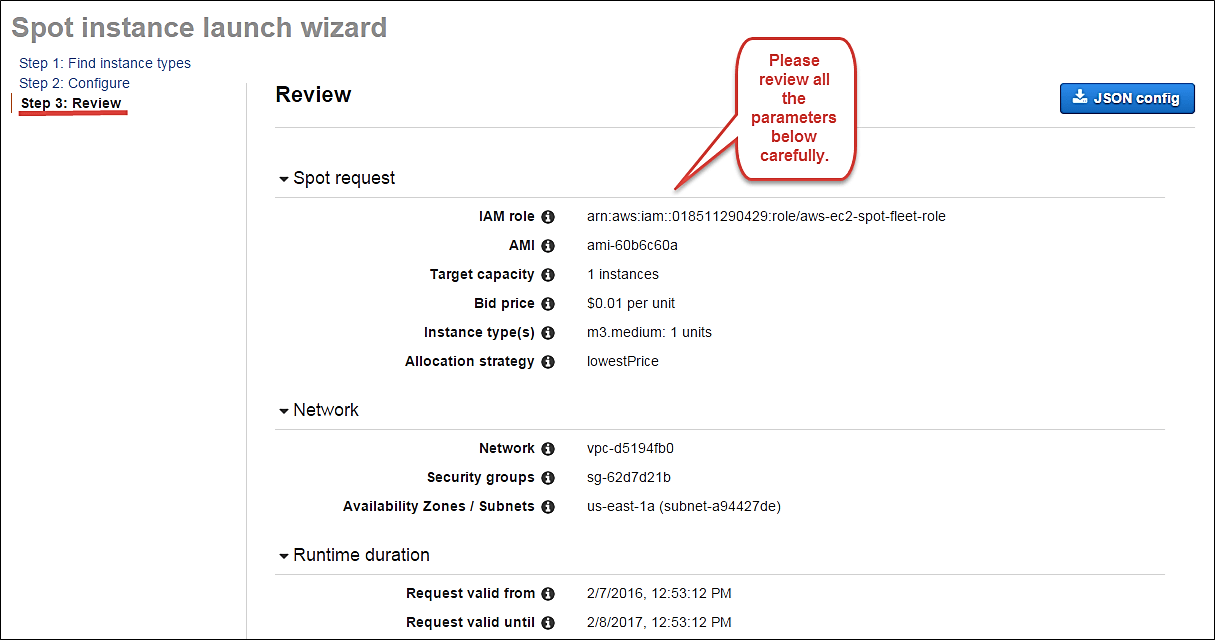
AWS stores the private key in the instance, and you are asked to download the public key. Make sure you download the key and keep it safe and secured**; if it is lost you cannot download it again.**

After selecting public IP, here we are selecting a key which we already have created in our last tutorial.

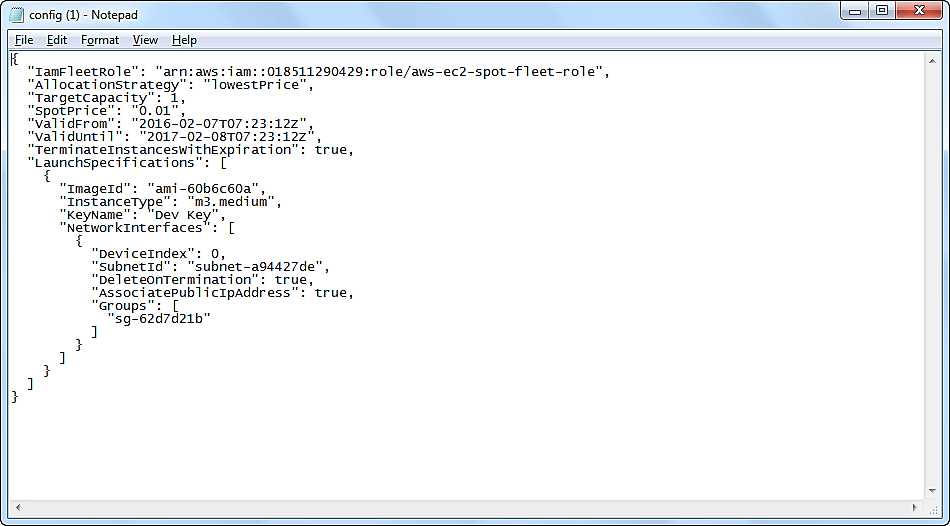
[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp22.png)

#### Reviewing your Spot instance

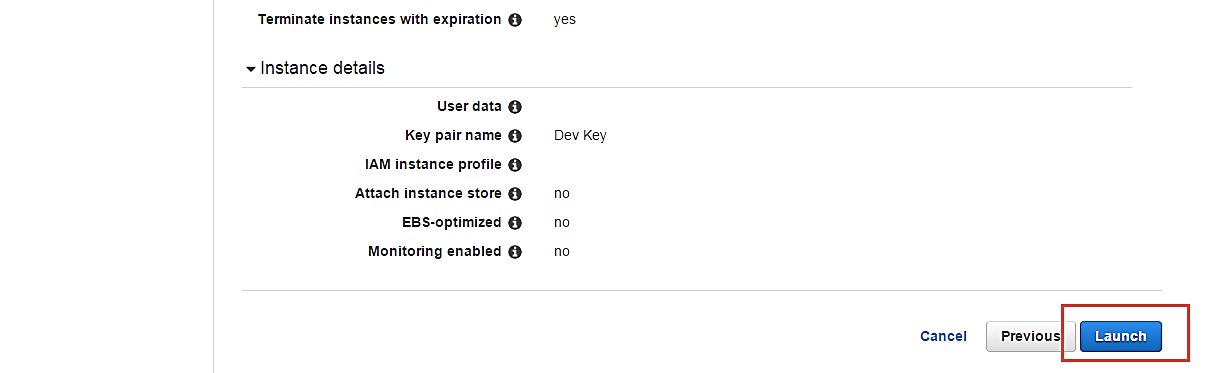
Once we are done configuring our spot instance request in the 2 steps earlier in our wizard, we'll take a look at the overall configuration.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp23.png)

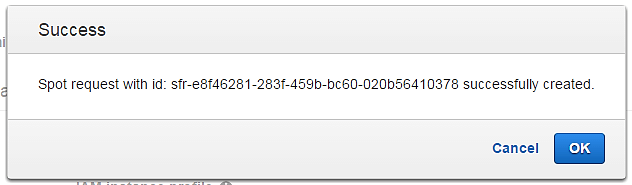
1. **We can also download a JSON file with all the configurations. Below is our JSON file.**

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp24.png)

After we are done reviewing, we can proceed with the launching by clicking the Launch button as shown below.

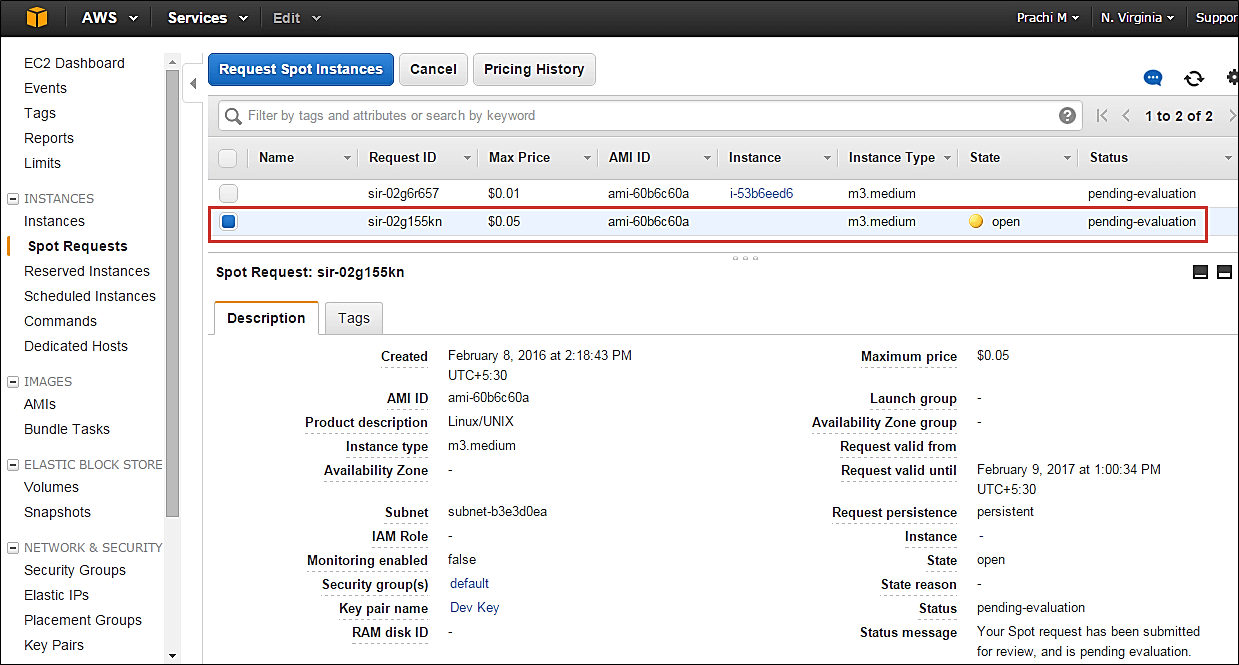
[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp25.png)

Once we select Launch, we can see a notification about the request getting created.

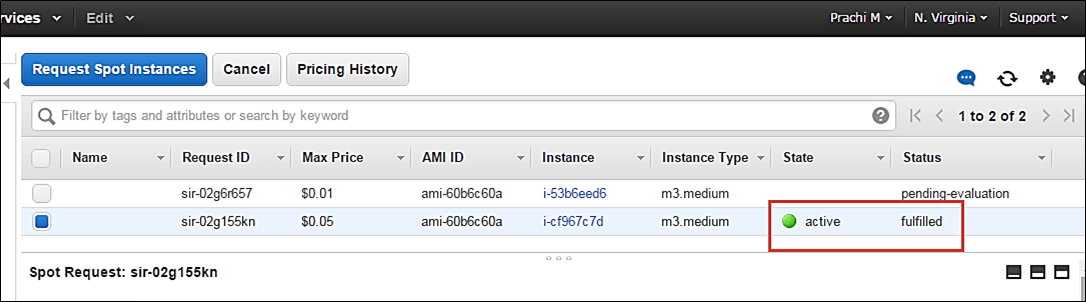
[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp26.png)

The spot request creation wizard will close, and the page will automatically direct back to the EC2 Dashboard.

You can see as shown below that the State of our request is 'open' which means that it is getting evaluated from the AWS's side. AWS EC2 will check if the required instance is available in its spot pool.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp27.png)

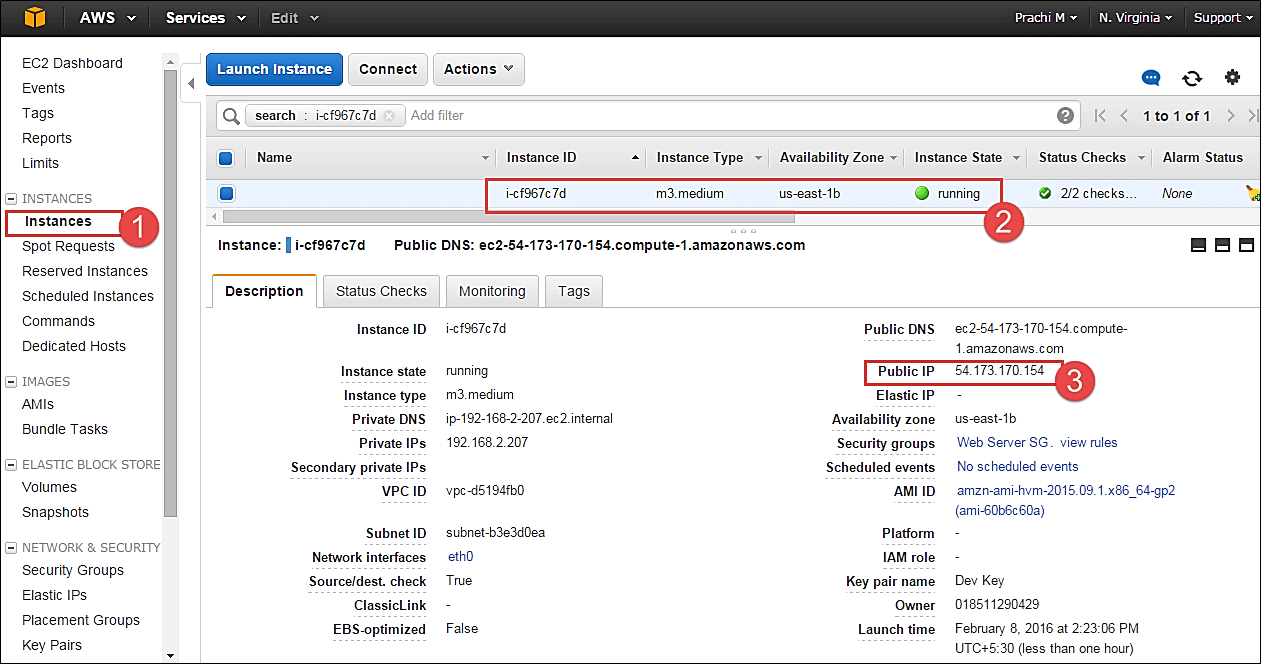
After a couple of minutes, you can see that the state is changed to 'active', and now our spot request is successfully fulfilled. You can note the configuration parameters below.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp28.png)

### Connecting to the spot instance

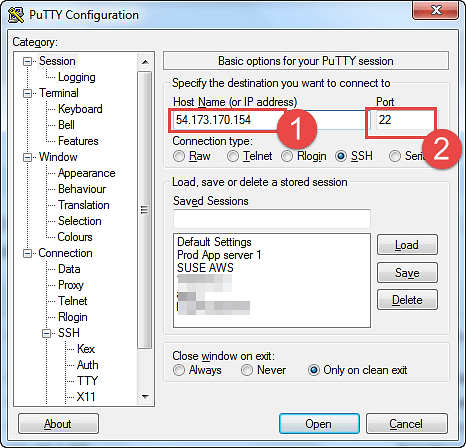
**Step 1)** In this step, you will notice following things

1. on the EC2 Dashboard, click on Instances in the left pane
2. Here you can see that the spot instance is running
3. Note down its Public IP

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp29.png)

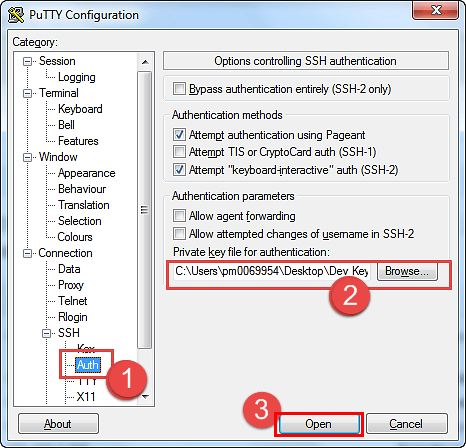
**Step 2)** Open putty in your computer.

1. Connect via the said Public IP that we noted above
2. Assign the connecting port as 22 for ssh access

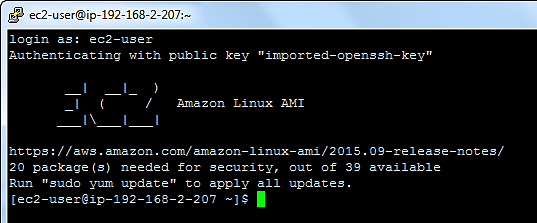
[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp30.png)

**Step 3)** In this step,

1. go to Auth Browse and
2. add our .ppk key to connect to the instance
3. Click open button

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp31.png)

You can now connect to your instance.

[](https://cdn.guru99.com/images/3-2016/032816_0656_CreatingaSp32.png)

**Summary**

Here, we saw how to create a Spot Instance request successfully by determining our bid price.

Spot instances are a great way to save on costs for instances which are not application critical. A common example would be to create a fleet of spot instances for a task such as image processing or video encoding. In such cases, you can keep a cluster of instances under a load balancer.

If the bid price exceeds the spot price and your instance is terminated from AWS's side, you can have other instances doing the processing job for you. You can leverage Auto scaling for this scenario. Avoid using Spot instances for business critical applications like databases etc.