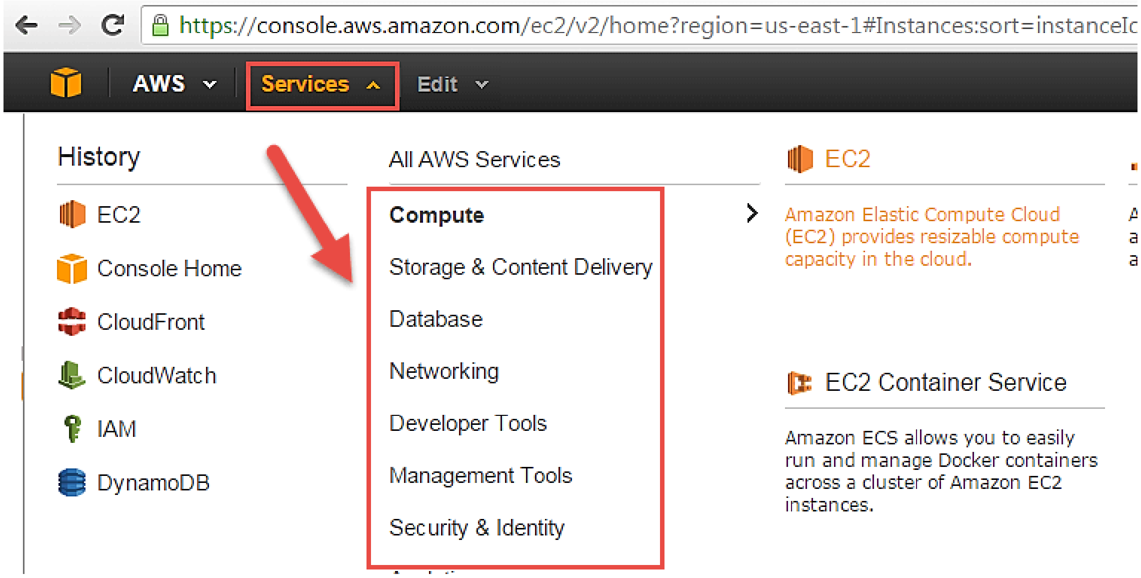
Lab-2 Creating and managing Amazon EC2 Instance

Step 1)

- 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.

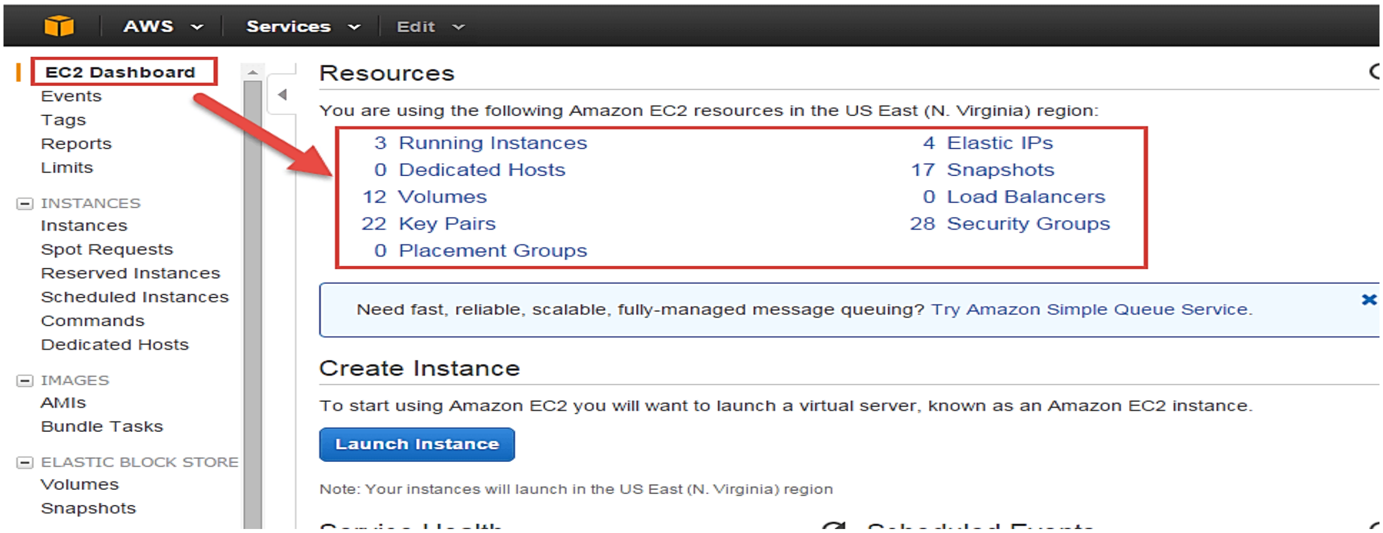


https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceIc

Amazon Elastic Compute Cloud (EC2) provides resizable compute capacity in the cloud.

Amazon ECS allows you to easily run and manage Docker containers across a cluster of Amazon EC2 instances.

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



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

3 Running Instances	4 Elastic IPs
0 Dedicated Hosts	17 Snapshots
12 Volumes	0 Load Balancers
22 Key Pairs	28 Security Groups
0 Placement Groups	

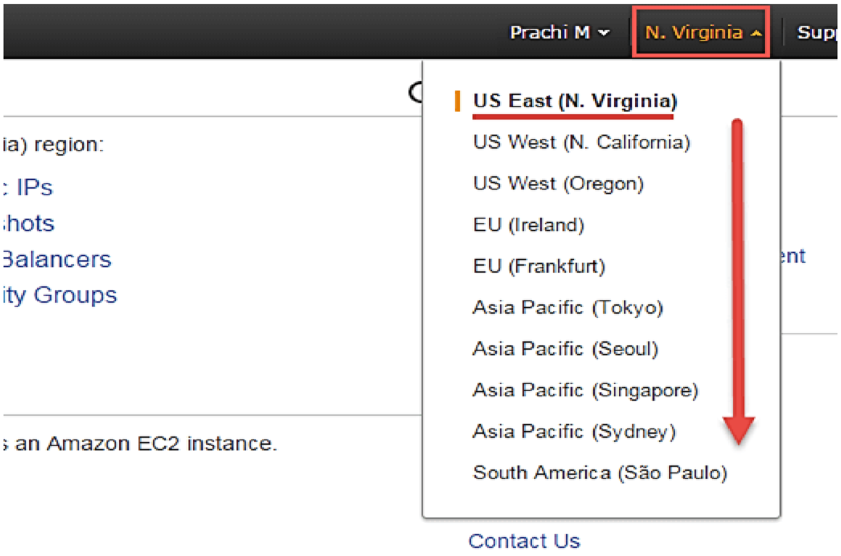
Need fast, reliable, scalable, fully-managed message queuing? Try Amazon Simple Queue Service.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

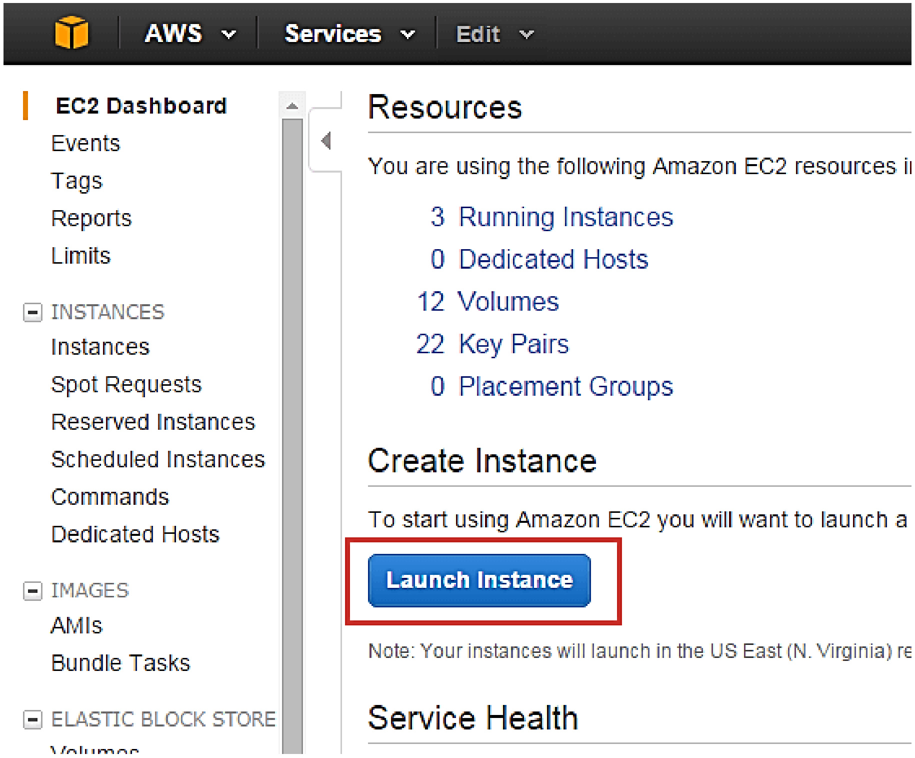
Note: Your instances will launch in the US East (N. Virginia) region

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.



Step 3)

- 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'.



Resources

You are using the following Amazon EC2 resources i

3 Running Instances
0 Dedicated Hosts
12 Volumes
22 Key Pairs
0 Placement Groups

Create Instance

To start using Amazon EC2 you will want to launch a

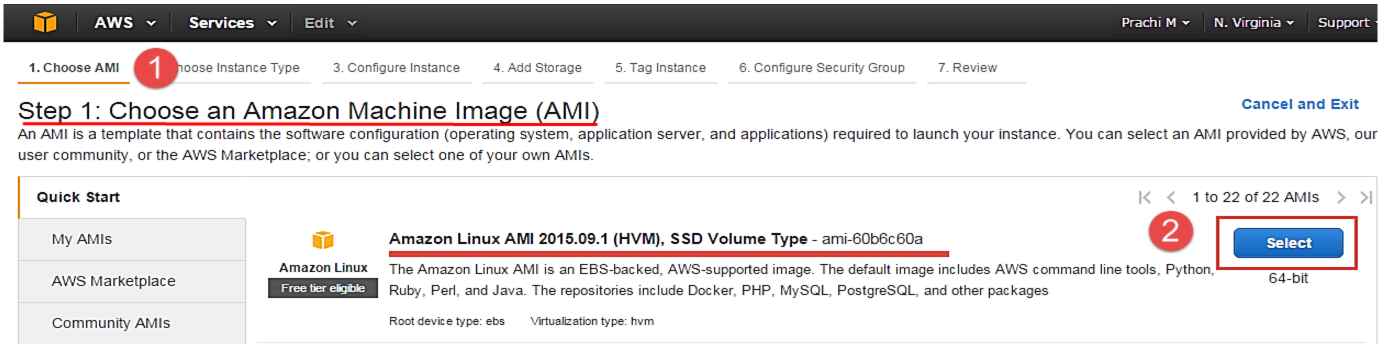
Note: Your instances will launch in the US East (N. Virginia) re

Service Health

Choosing 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 (64 bit) AMI.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-60b6c60a

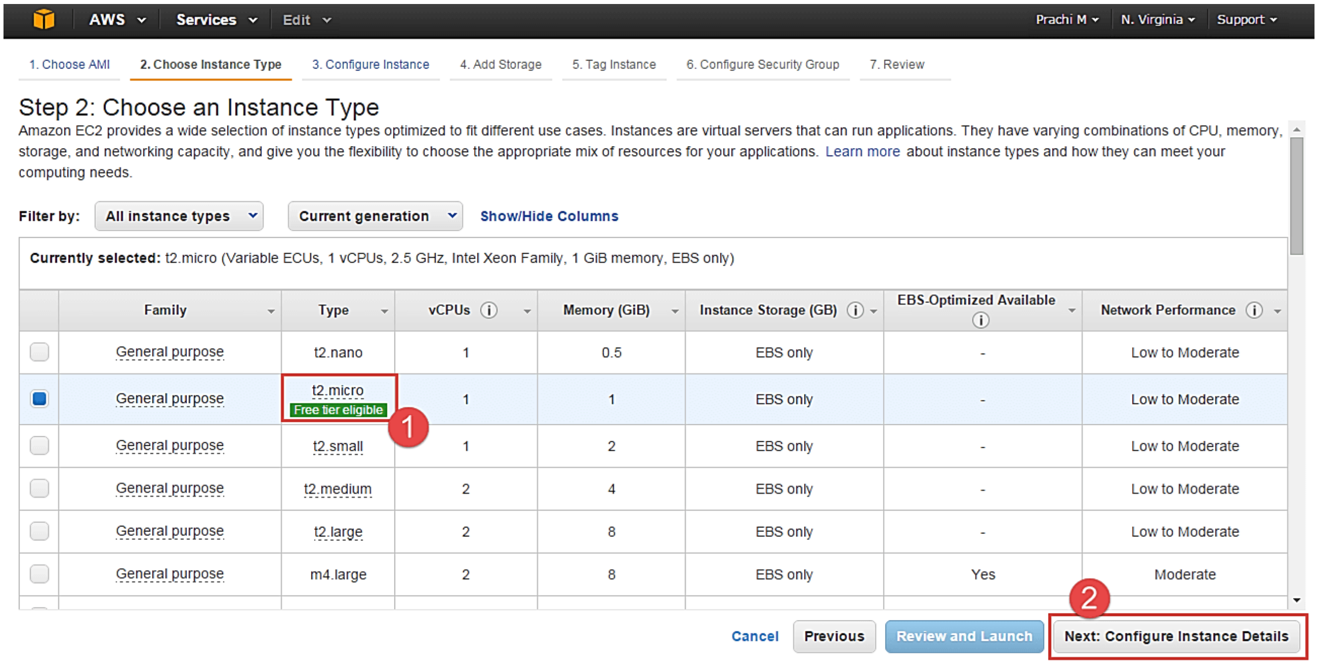
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages

Root device type: ebs Virtualization type: hvm

Choose EC2 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



Step 2: Choose an Instance Type

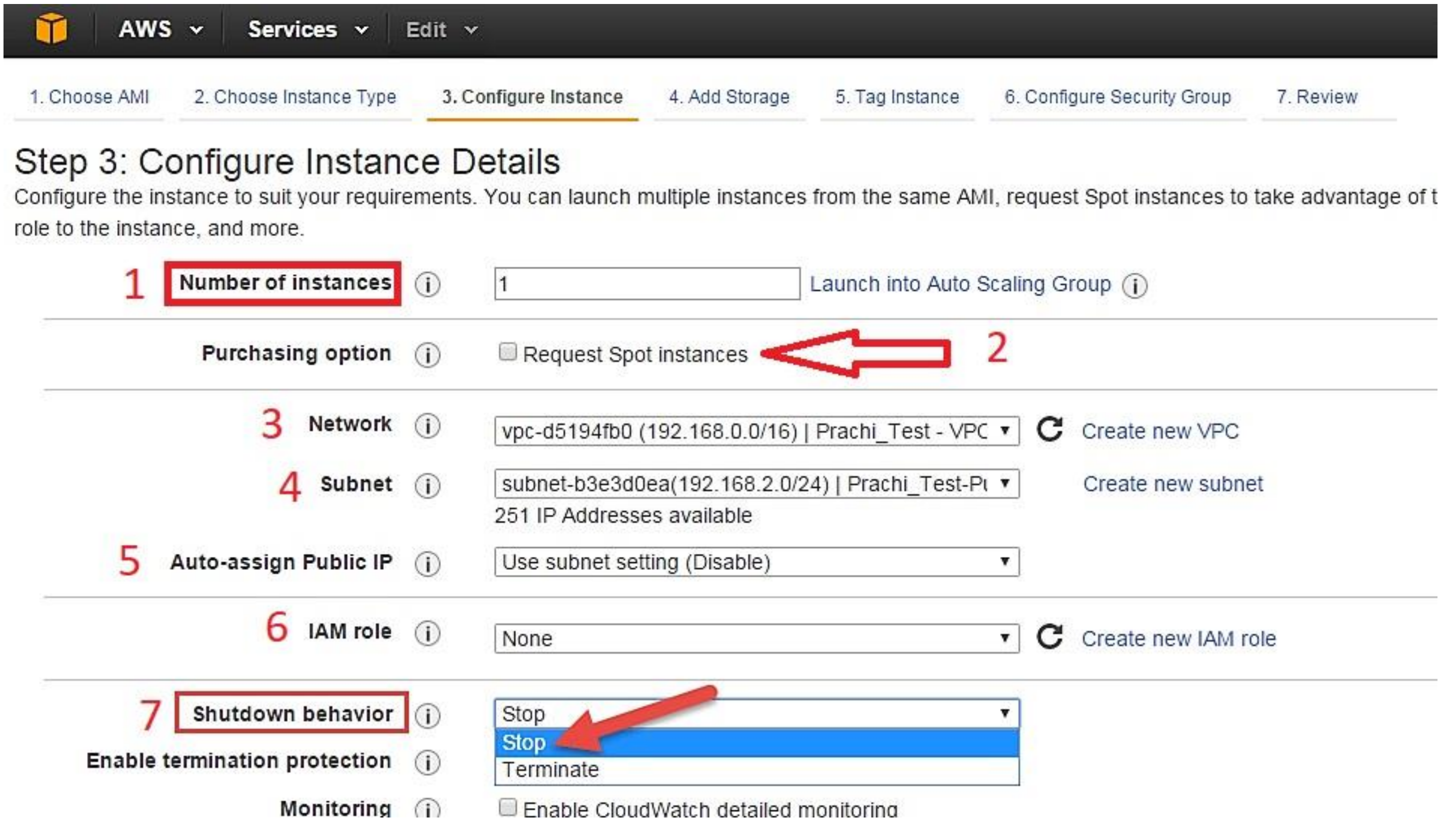
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
☐	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
☐	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
☐	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
☐	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
☐	General purpose	m4.large	2	8	EBS only	Yes	Moderate

Configure Instance

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



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of t role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pu
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

IAM role: None

Shutdown behavior: Stop
Stop
Terminate

Enable termination protection

Monitoring: □ Enable CloudWatch detailed monitoring

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.

- 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.

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.

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.

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.

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

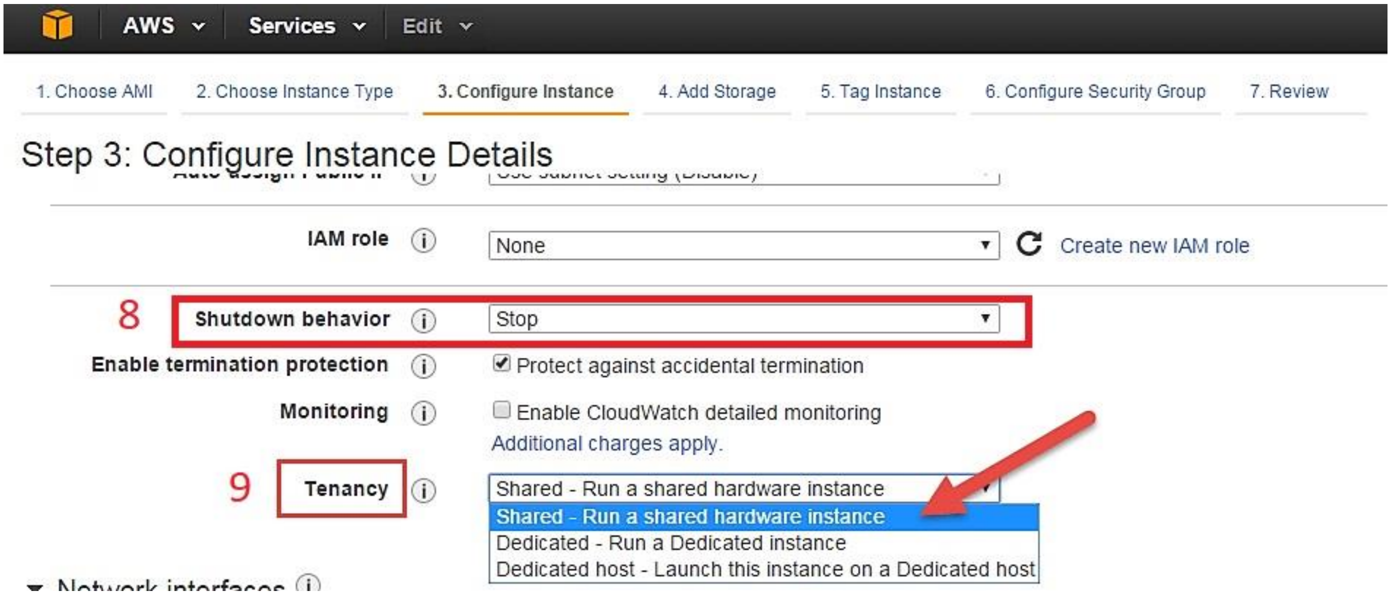
- Shutdown Behavior – when you accidently shut down your instance, you surely don't want it to be deleted but stopped.
- Here we are defining my shutdown behavior as Stop.

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.

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



Step 3: Configure Instance Details

IAM role: None — Create new IAM role

Shutdown behavior: Stop

Enable termination protection: ☑ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring
Additional charges apply.

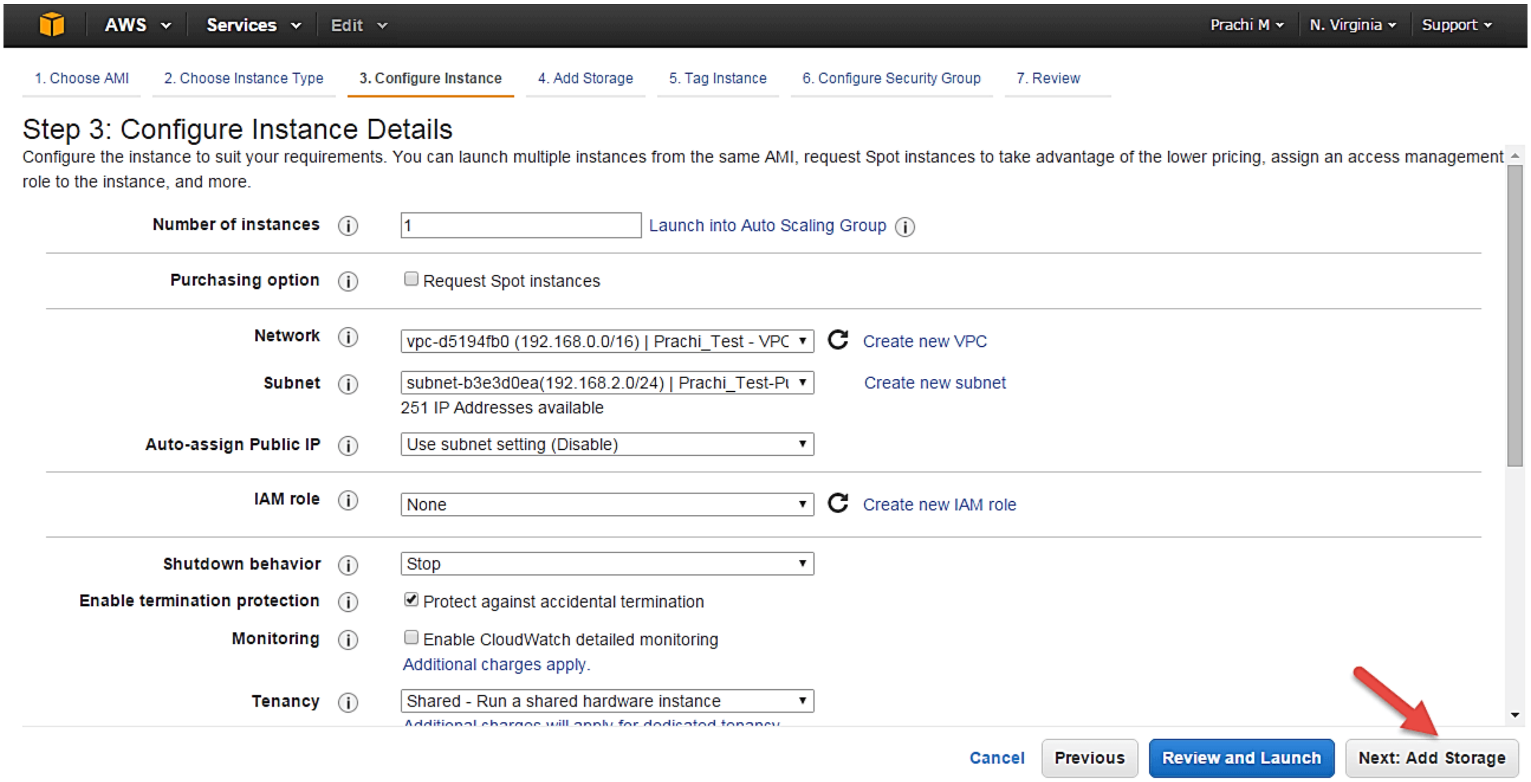
Tenancy: Shared - Run a shared hardware instance

- Shared - Run a shared hardware instance
- Dedicated - Run a Dedicated instance
- Dedicated host - Launch this instance on a Dedicated host

Network interfaces

Step 10) In this step,

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



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC — Create new VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pu — Create new subnet
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

IAM role: None — Create new IAM role

Shutdown behavior: Stop

Enable termination protection: ☑ Protect against accidental termination

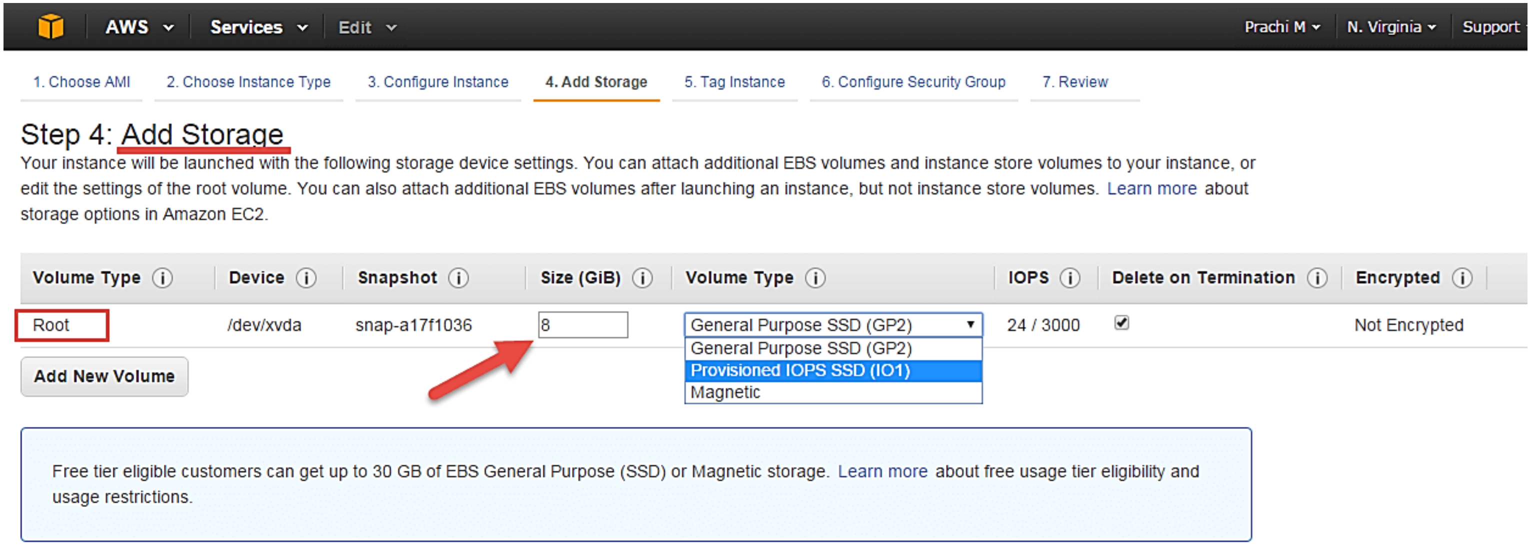
Monitoring: □ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy: Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.

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



Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/xvda	snap-a17f1036	8	General Purpose SSD (GP2)	24 / 3000	☑	Not Encrypted

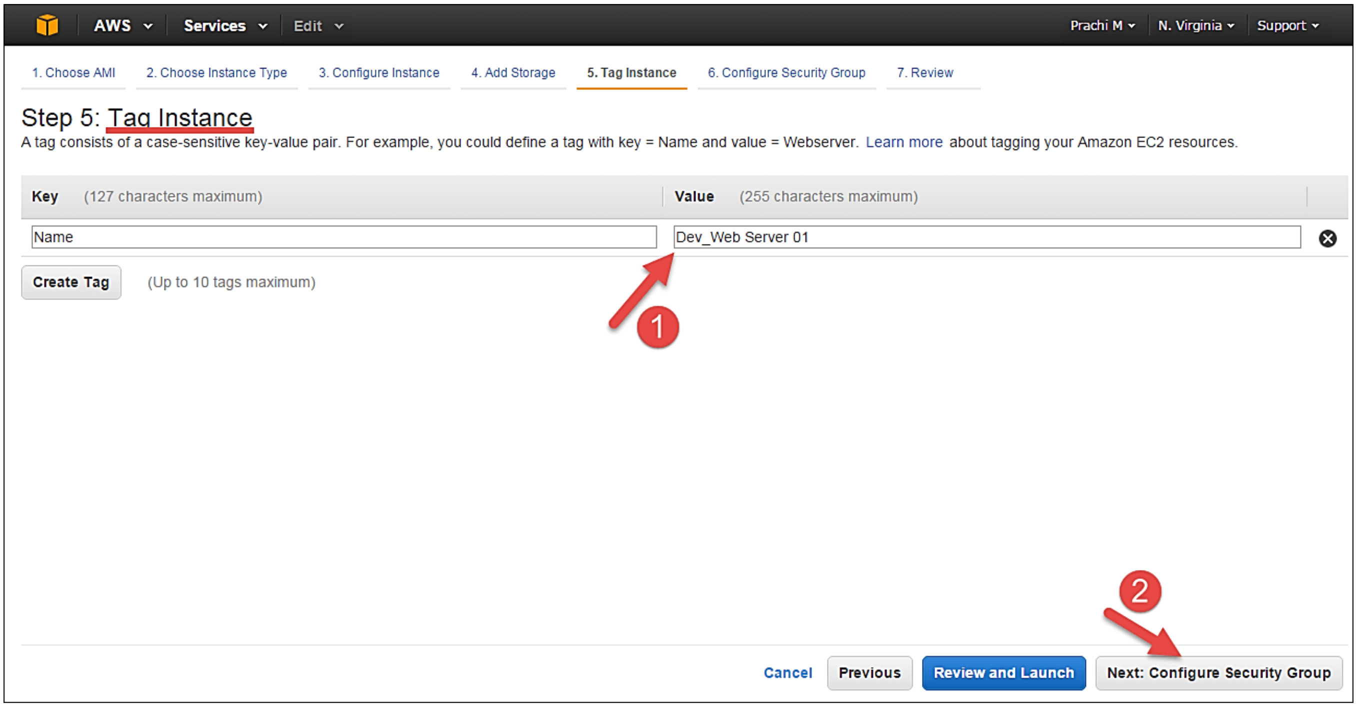
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

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



Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

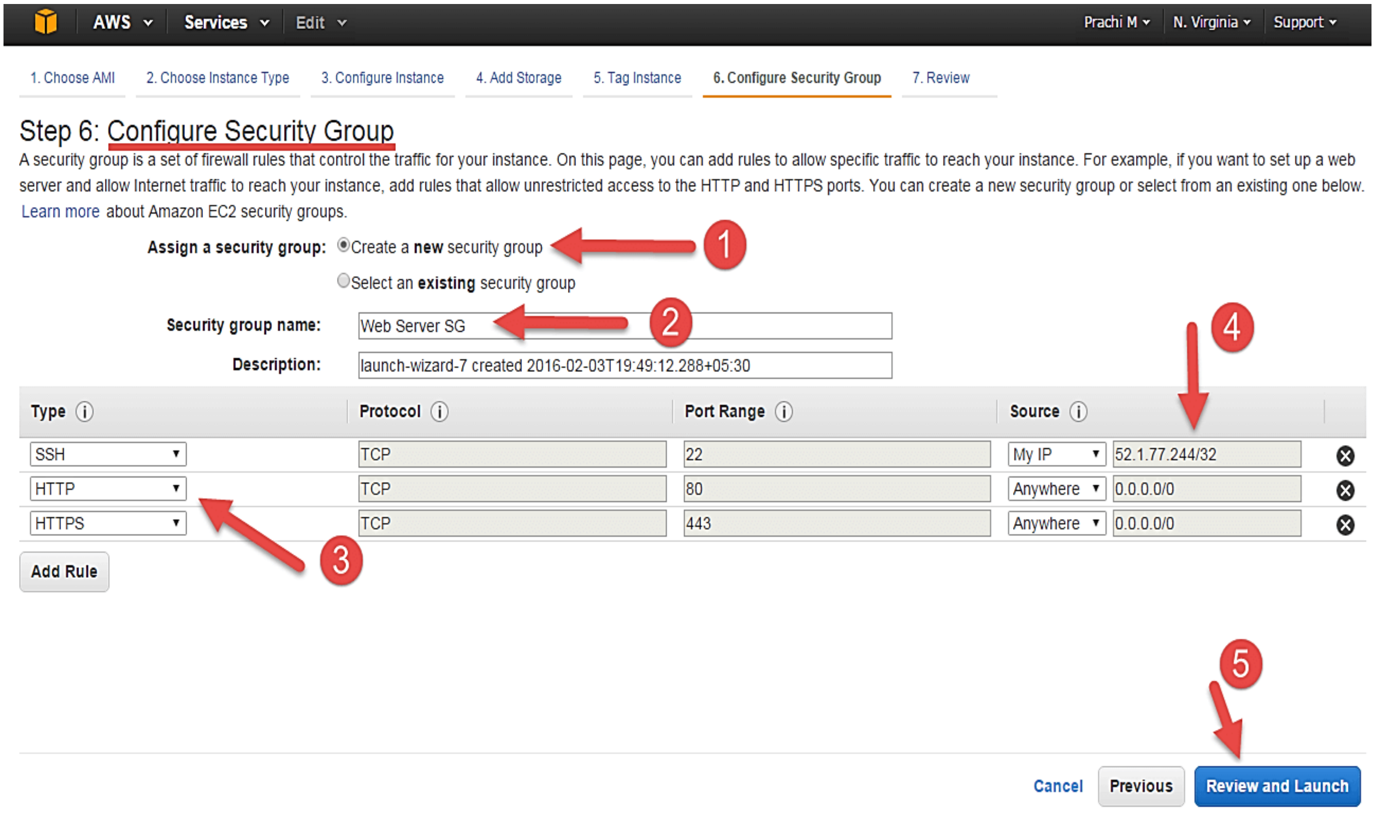
Key (127 characters maximum)	Value (255 characters maximum)
Name	Dev_Web Server 01

Configure 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



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

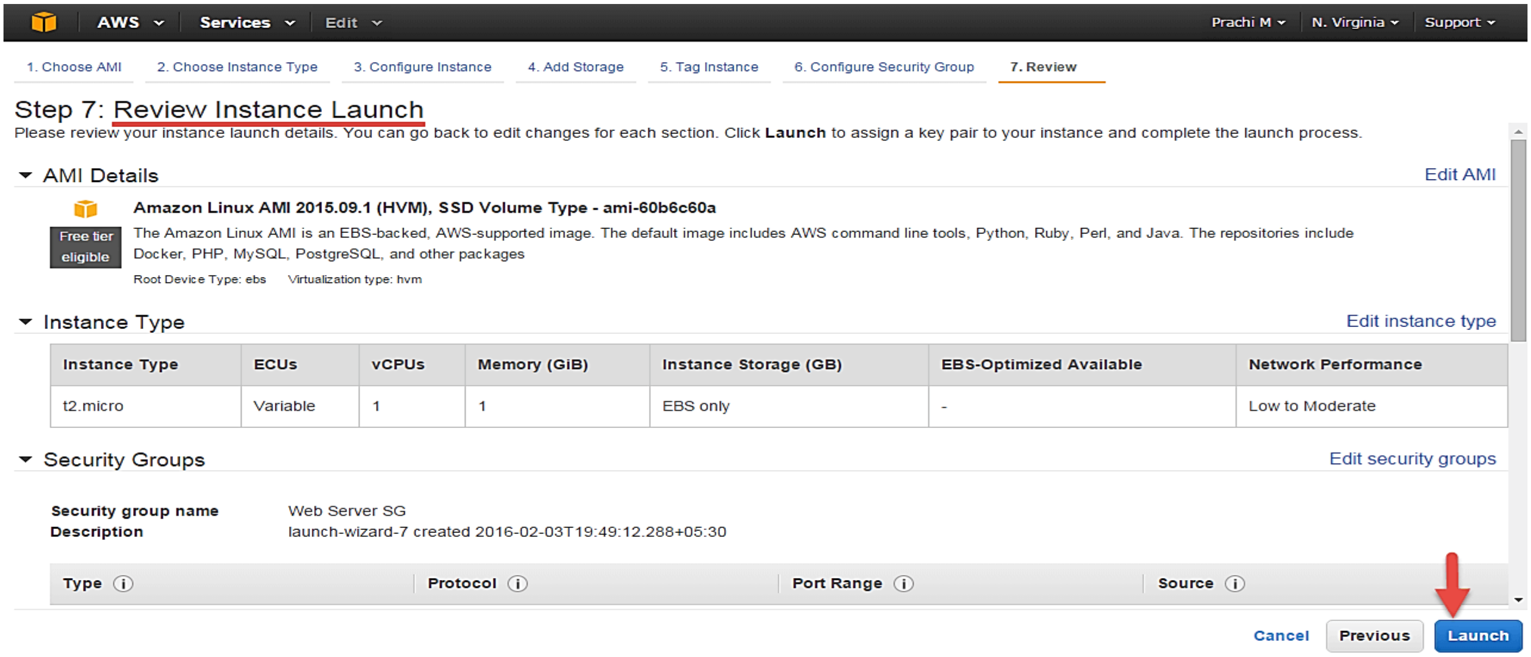
Security group name: Web Server SG

Description: launch-wizard-7 created 2016-02-03T19:49:12.288+05:30

Type	Protocol	Port Range	Source	
SSH	TCP	22	My IP	52.1.77.244/32
HTTP	TCP	80	Anywhere	0.0.0.0/0
HTTPS	TCP	443	Anywhere	0.0.0.0/0

Review Instances

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



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-60b6c60a

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

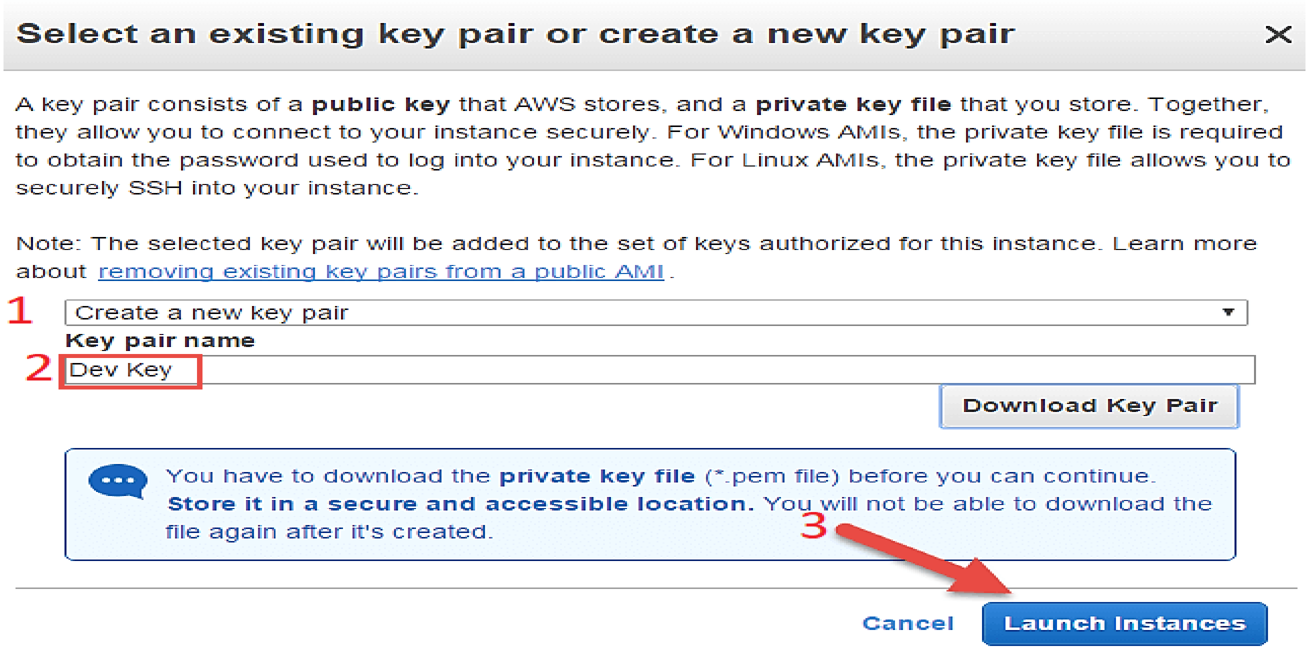
Security group name Web Server SG
Description launch-wizard-7 created 2016-02-03T19:49:12.288+05:30

Type	Protocol	Port Range	Source

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 and Download and save it in your secured folder
3. Launch the instance once file downloaded



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

1 Create a new key pair

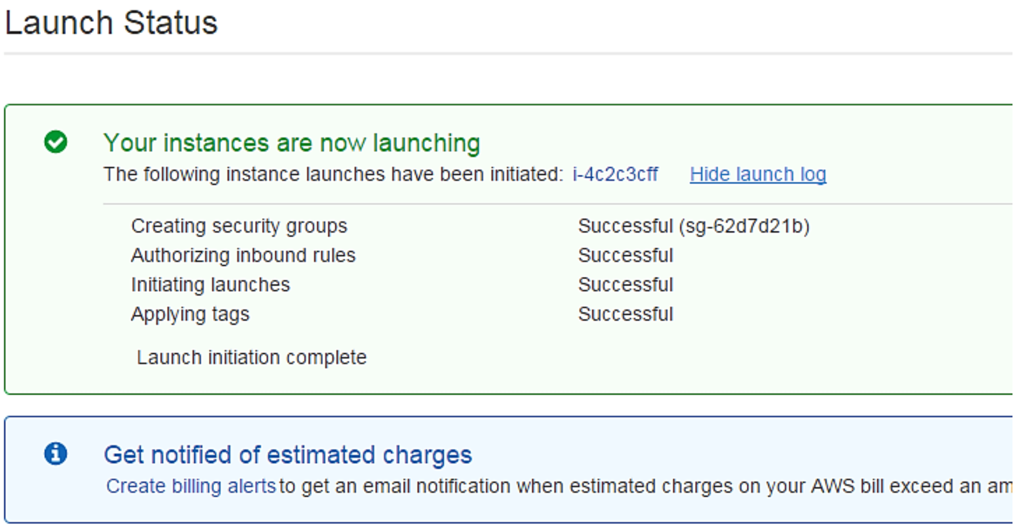
Key pair name

2 Dev Key

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

You can also see the launch log.

Wait until you get 2/2 checks in Instance state.



Launch Status

Your instances are now launching

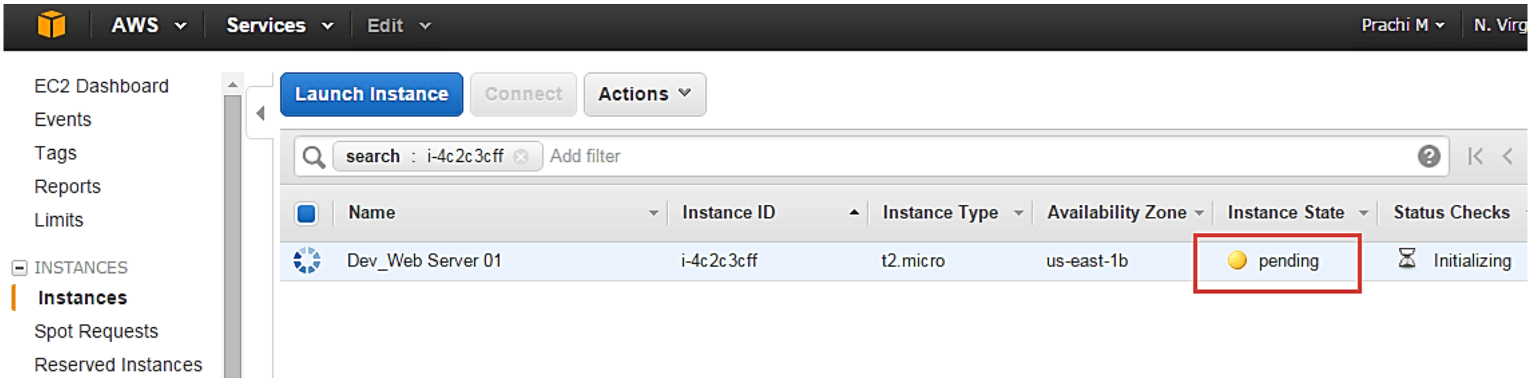
The following instance launches have been initiated: i-4c2c3cff Hide launch log

Creating security groups	Successful (sg-62d7d21b)
Authorizing inbound rules	Successful
Initiating launches	Successful
Applying tags	Successful

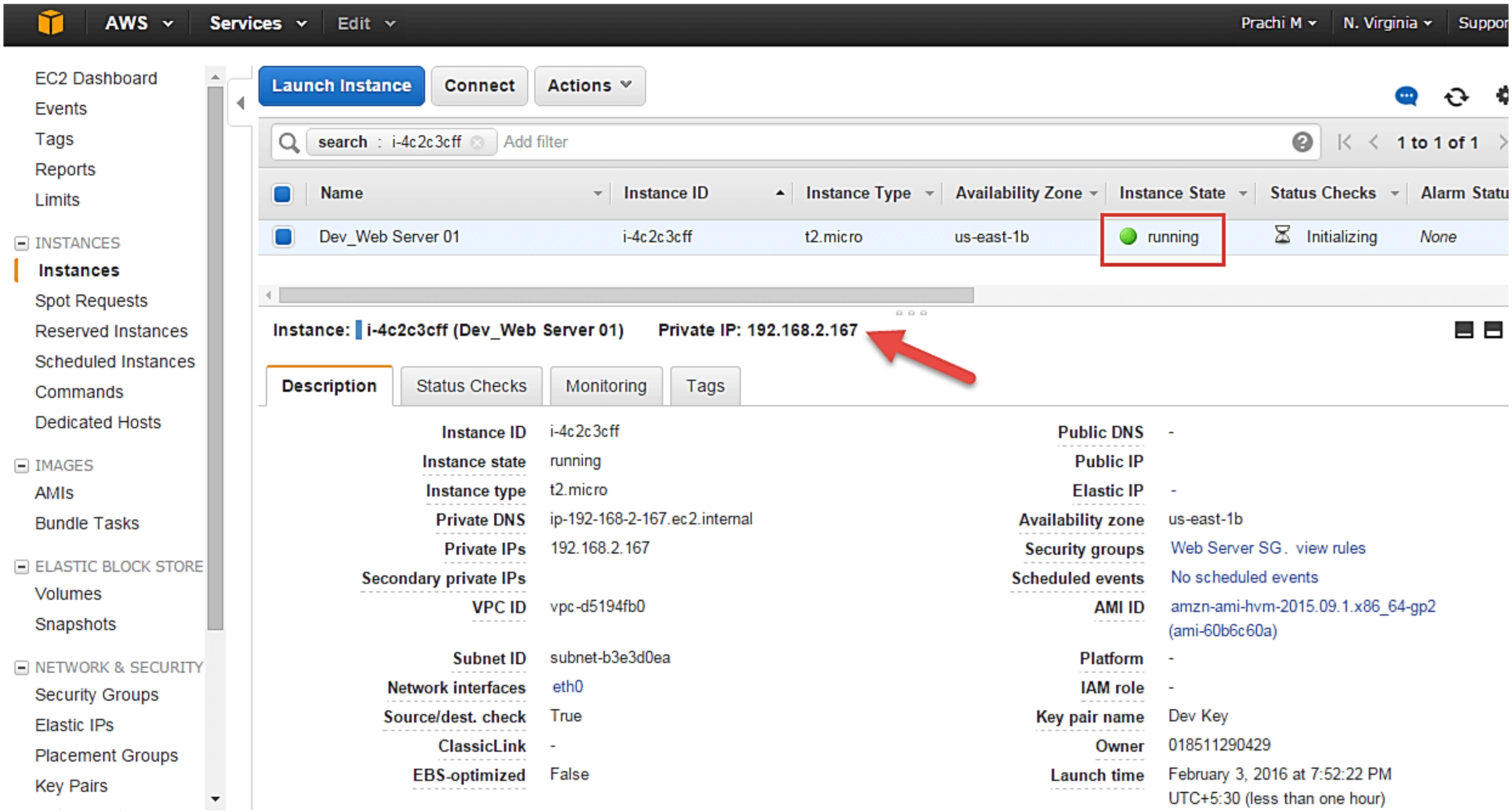
Launch initiation complete

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an am



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
Dev_Web Server 01	i-4c2c3cff	t2.micro	us-east-1b	● pending	Initializing



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Statu
Dev_Web Server 01	i-4c2c3cff	t2.micro	us-east-1b	● running	Initializing	*None*

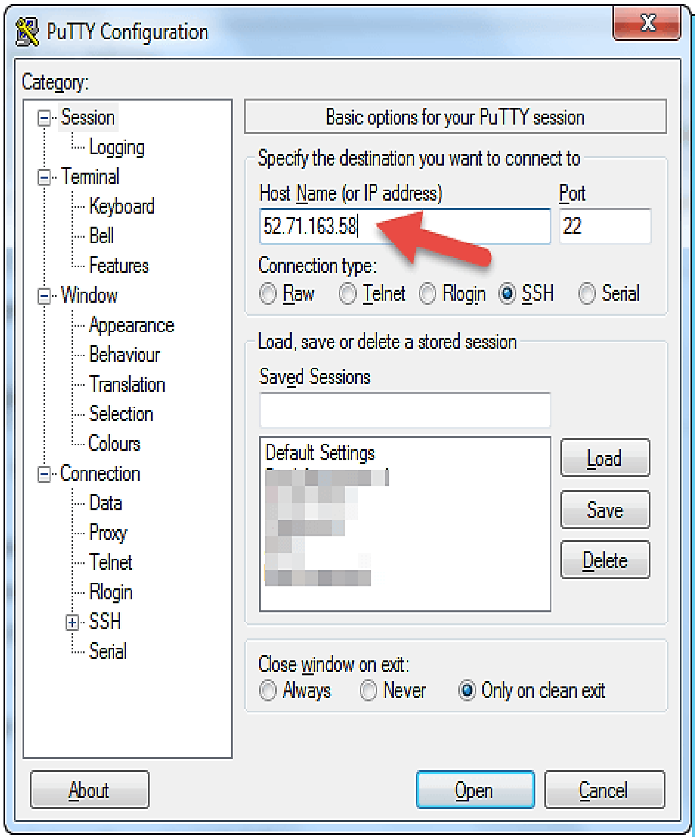
Instance: i-4c2c3cff (Dev_Web Server 01) **Private IP: 192.168.2.167**

Instance ID	i-4c2c3cff	**Public DNS**	-
Instance state	running	**Public IP**	
Instance type	t2.micro	**Elastic IP**	-
Private DNS	ip-192-168-2-167.ec2.internal	**Availability zone**	us-east-1b
Private IPs	192.168.2.167	**Security groups**	Web Server SG. view rules
Secondary private IPs		**Scheduled events**	No scheduled events
VPC ID	vpc-d5194fb0	**AMI ID**	amzn-ami-hvm-2015.09.1.x86_64-gp2 (ami-60b6c60a)
Subnet ID	subnet-b3e3d0ea	**Platform**	-
Network interfaces	eth0	**IAM role**	-
Source/dest. check	True	**Key pair name**	Dev Key
ClassicLink	-	**Owner**	018511290429
EBS-optimized	False	**Launch time**	February 3, 2016 at 7:52:22 PM UTC+5:30 (less than one hour)

Logging In to EC2 instance

Enter Public Ip of instance
Add your private key in putty for secure connection

1. Go to Auth
2. Add your private key in .ppk (putty private key) format. You will need to convert pem file from AWS to ppk using puttygen
3. Once done click on "Open" button
4. Once you connect, you will successfully see the Linux prompt.
5. Please note that the machine you are connecting from should be enabled on the instance Security Group for SSH (like in the steps above).



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 52.71.163.58

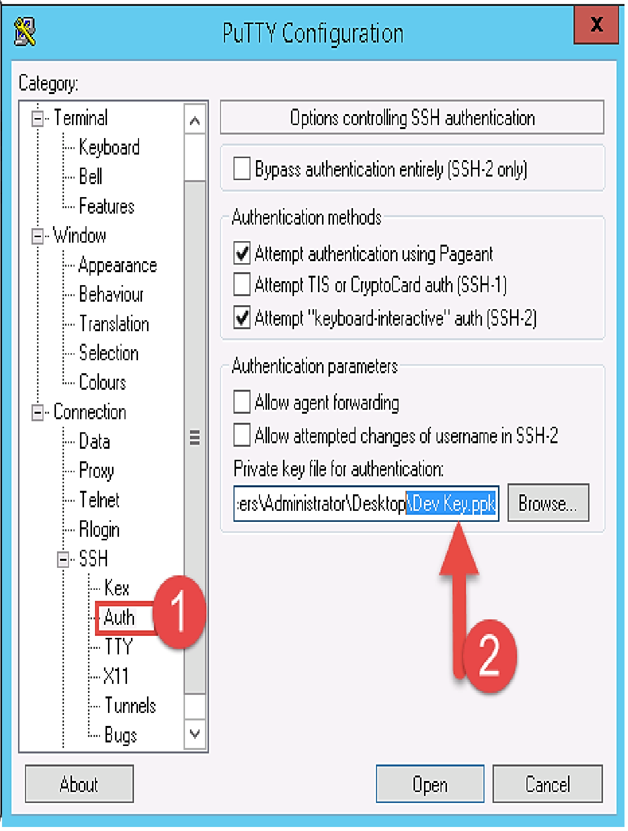
Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit



PuTTY Configuration

Options controlling SSH authentication

☐ Bypass authentication entirely (SSH-2 only)

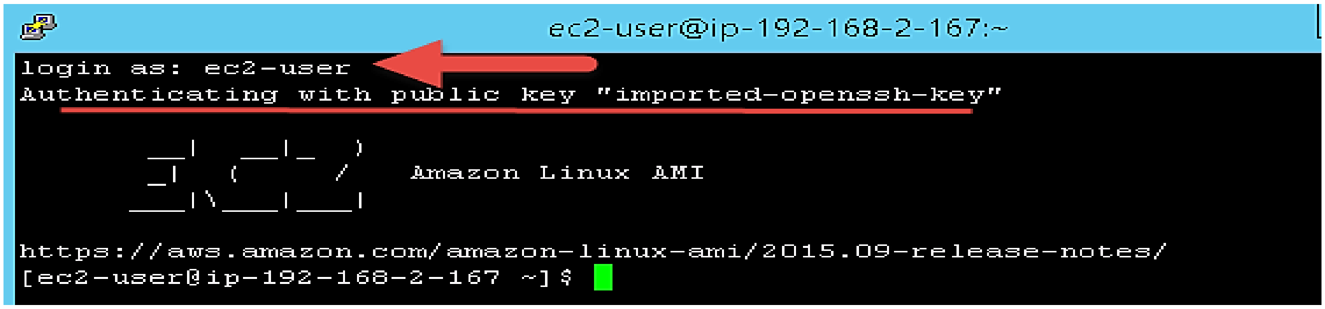
Authentication methods

☑ Attempt authentication using Pageant
☐ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

☐ Allow agent forwarding
☐ Allow attempted changes of username in SSH-2

Private key file for authentication: ers\Administrator\Desktop\Dev Key.ppk



```
ec2-user@ip-192-168-2-167:~
login as: ec2-user
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/
[ec2-user@ip-192-168-2-167 ~]$
```

That's all you have done