Practical No: 01

Practical Title: Case study on Amazon EC2 and learn about Amazon EC2 web services.

Objectives:

- To learn Amazon EC2 web services
- To study on Amazon EC2 and learn about Amazon EC2 web services.

Hardware Requirements:

- Pentium IV with latest configuration

Software Requirements:

- Ubuntu 20.04

Theory:

An EC2 instance is nothing but a virtual server in Amazon Web services terminology. It stands for Elastic Compute Cloud. It is a web service where an AWS subscriber can request and provision a compute server in AWS cloud.

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

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

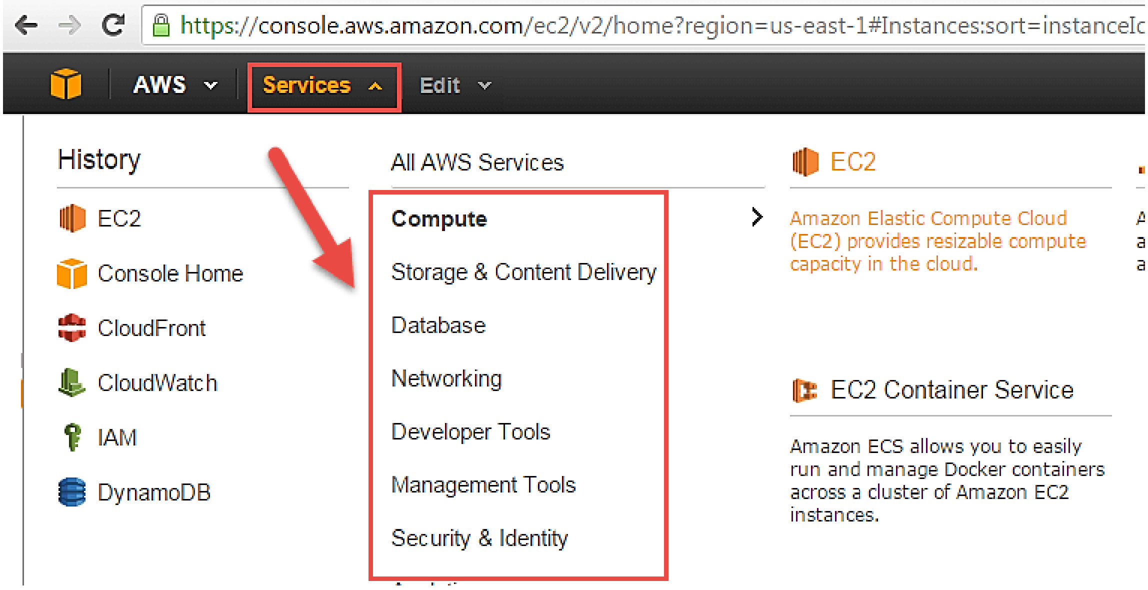
Thus, you can rent an instance based on your own CPU and memory requirements and use it as long as you want. You can terminate the instance when it's no more used and save on costs. This is the most striking advantage of an on-demand instance- you can drastically save on your CAPEX.

Let us see in detail 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 à EC2 as in the next step.

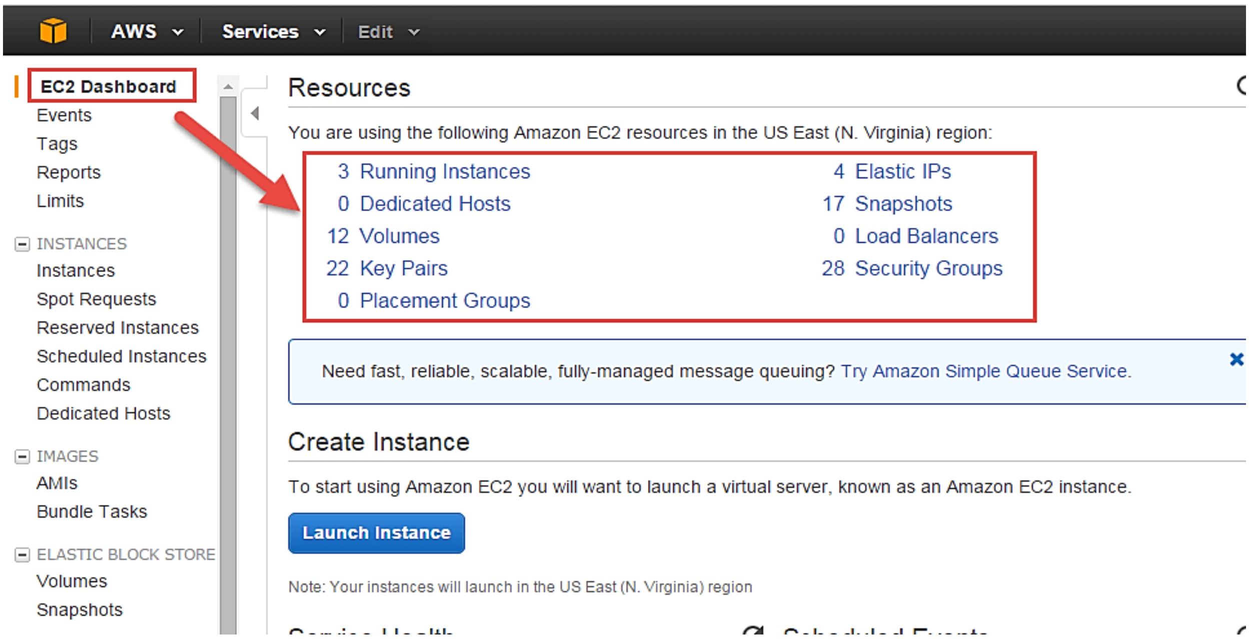


Amazon Elastic Compute Cloud (EC2) provides resizable compute capacity in the cloud.

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

- 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.



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

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

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

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

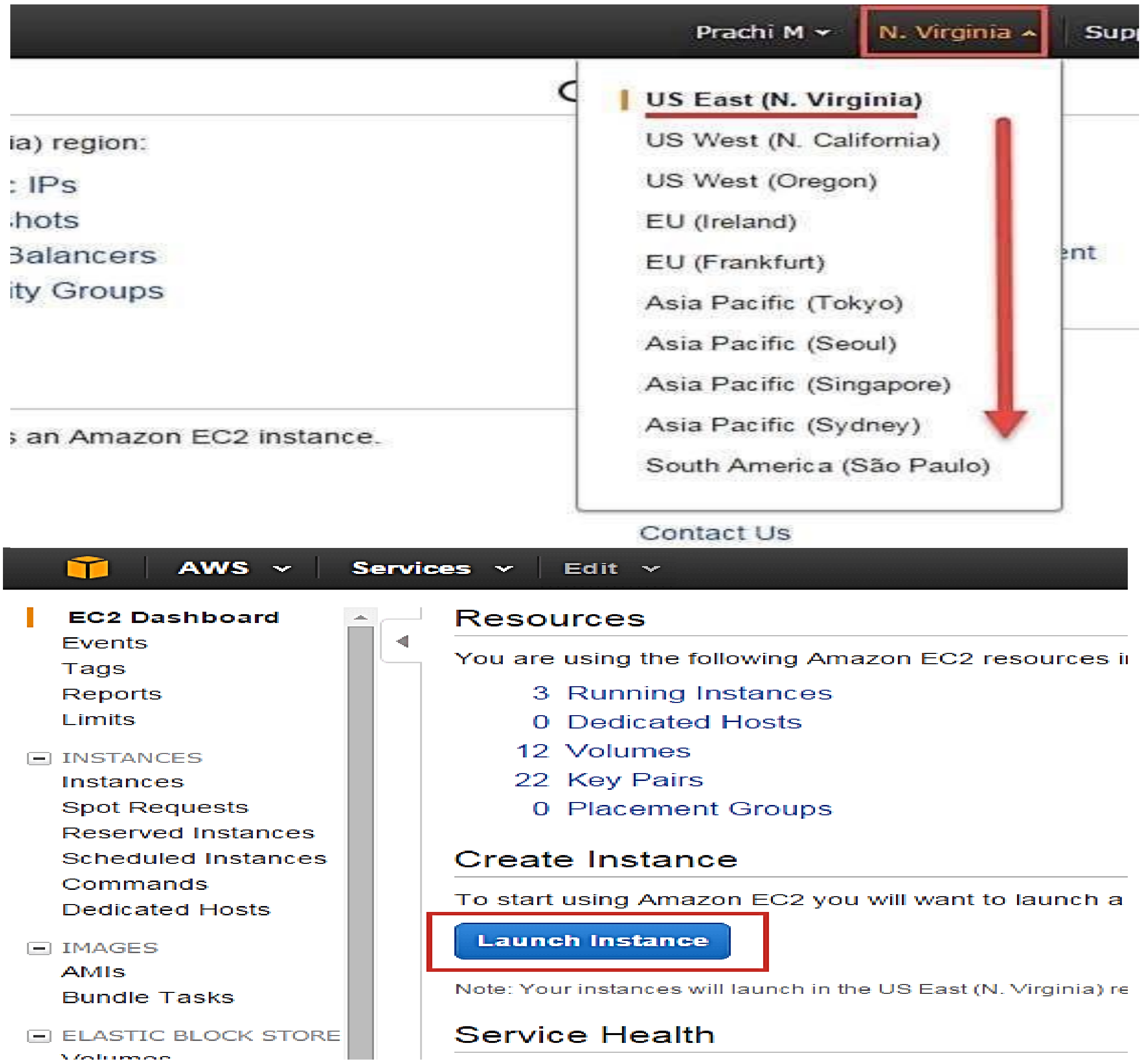
Note: Your instances will launch in the US East (N. Virginia) region

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) 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).



s an Amazon EC2 instance.

You are using the following Amazon EC2 resources i

To start using Amazon EC2 you will want to launch a

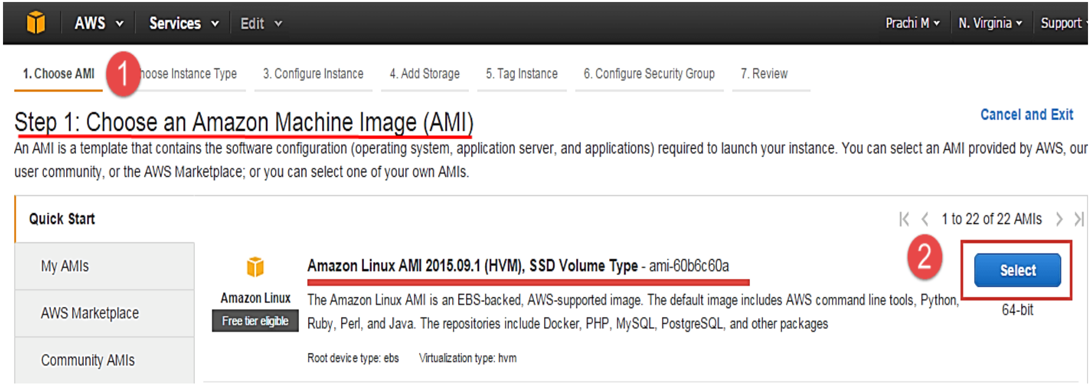
Note: Your instances will launch in the US East (N. Virginia) re

- 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 (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

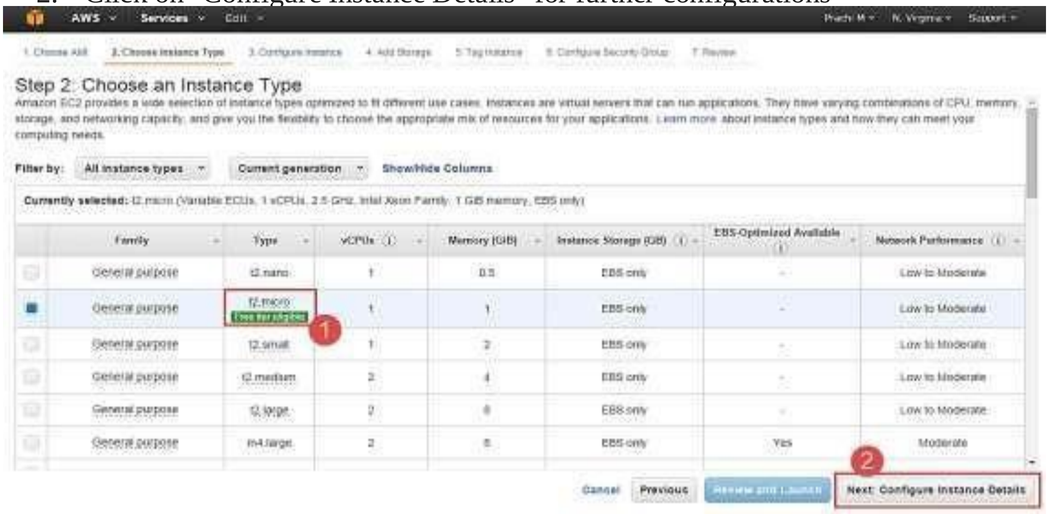
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages

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 memoryserver 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.

Filter by: All instance types Current generation Show/Hide Columns

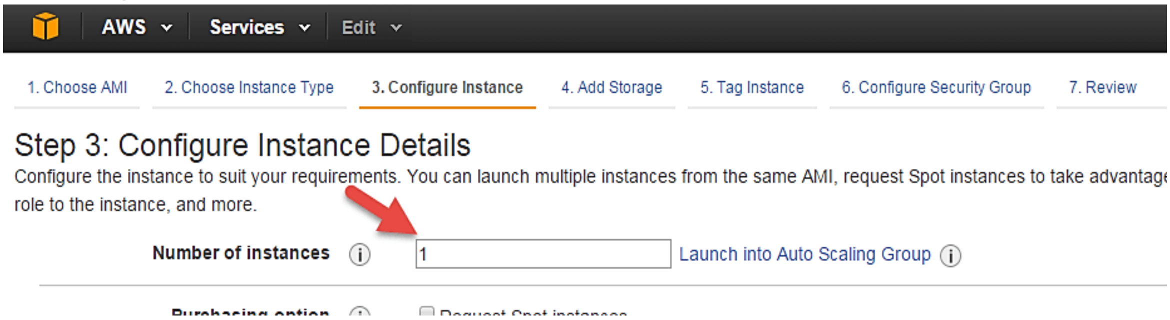
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GB memory, EBS only)

	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

- 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 launchingone instance.



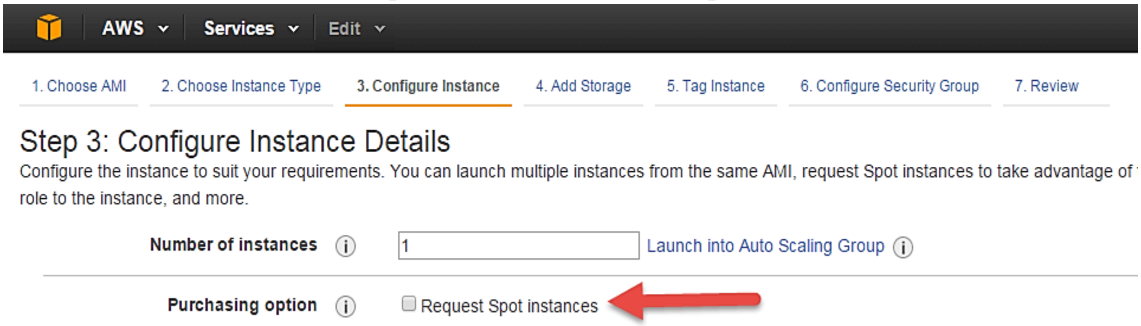
Step 3: Configure Instance Details

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

Number of instances 1

Purchasing option □ Request Spot instances

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: 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 role to the instance, and more.

Number of instances 1

Purchasing option □ Request Spot instances

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

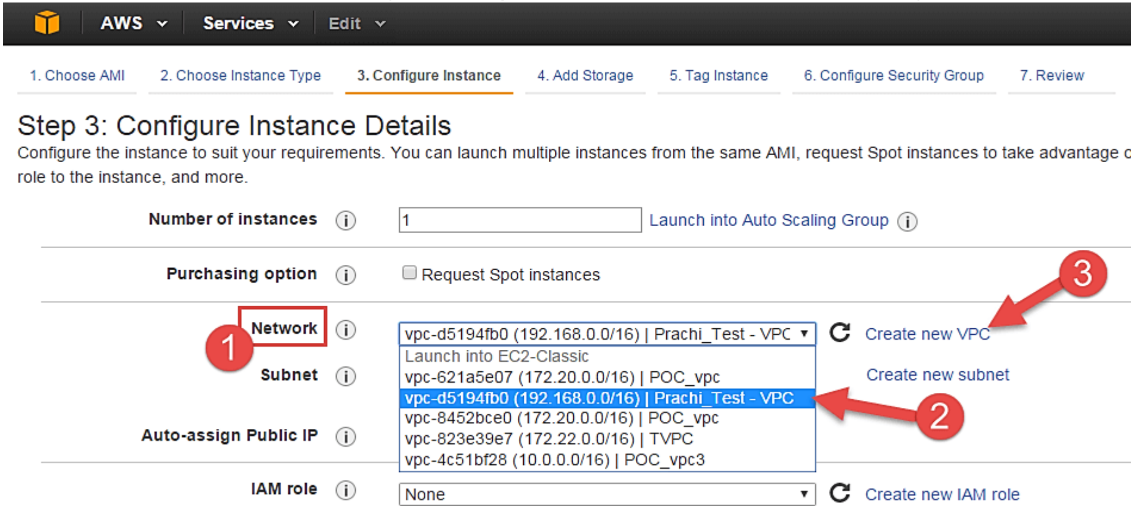
- You have to decide here, in which VPC (Virtual Private Cloud) you want to launch your instance and under which subnets inside your VPC. It is better to determine and plan this prior to launching the instance. Your AWS architecture set-up should include IP ranges for your subnets etc. pre-planned for 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.



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 c 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
- Launch into EC2-Classic
- vpc-621a5e07 (172.20.0.0/16) | POC_vpc
- vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC
- vpc-8452bce0 (172.20.0.0/16) | POC_vpc
- vpc-823e39e7 (172.22.0.0/16) | TVPC
- vpc-4c51bf28 (10.0.0.0/16) | POC_vpc3

Subnet

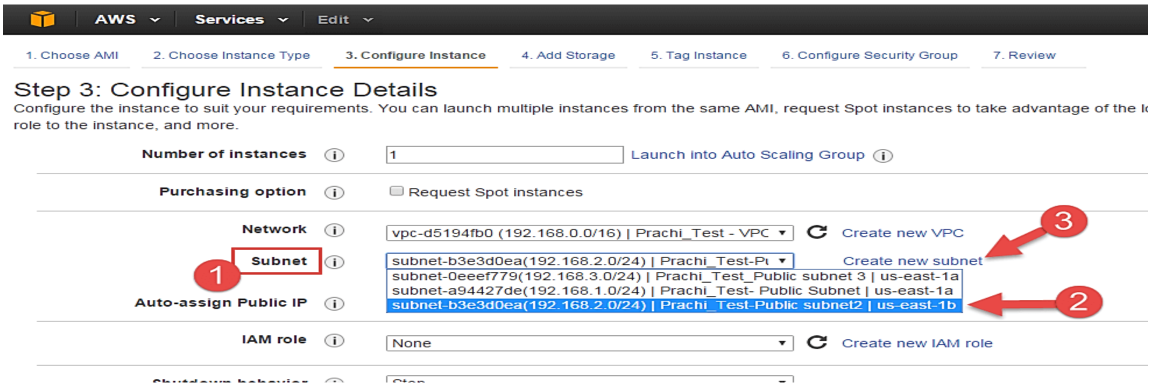
Auto-assign Public IP

IAM role: None

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 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 l 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-Pι
- subnet-0eeef779(192.168.3.0/24) | Prachi_Test_Public subnet 3 | us-east-1a
- subnet-a94427de(192.168.1.0/24) | Prachi_Test- Public Subnet | us-east-1a
- subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Public subnet2 | us-east-1b

Auto-assign Public IP

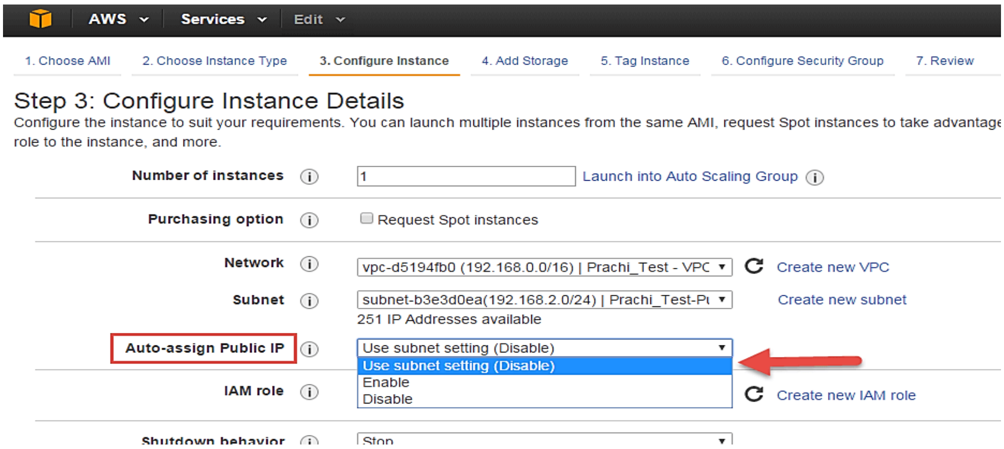
IAM role: None

- 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.



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 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

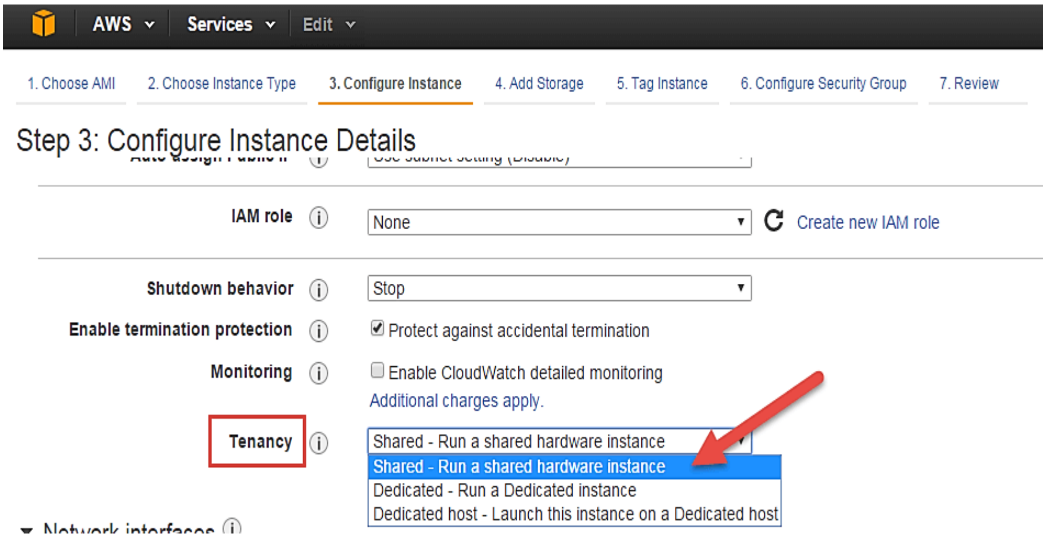
Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pu — Create new subnet
251 IP Addresses available

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

- Use subnet setting (Disable)
- Enable
- Disable

IAM role — Create new IAM role

Shutdown behavior: Stop



Step 3: Configure Instance Details

IAM role: None — Create new IAM role

Shutdown behavior: Stop

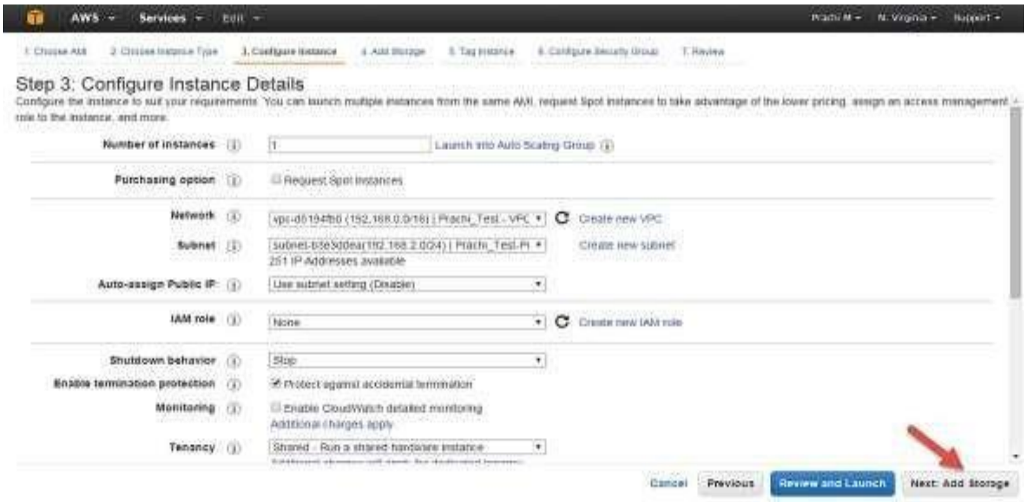
Enable termination protection: ☑ Protect against accidental termination

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 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)

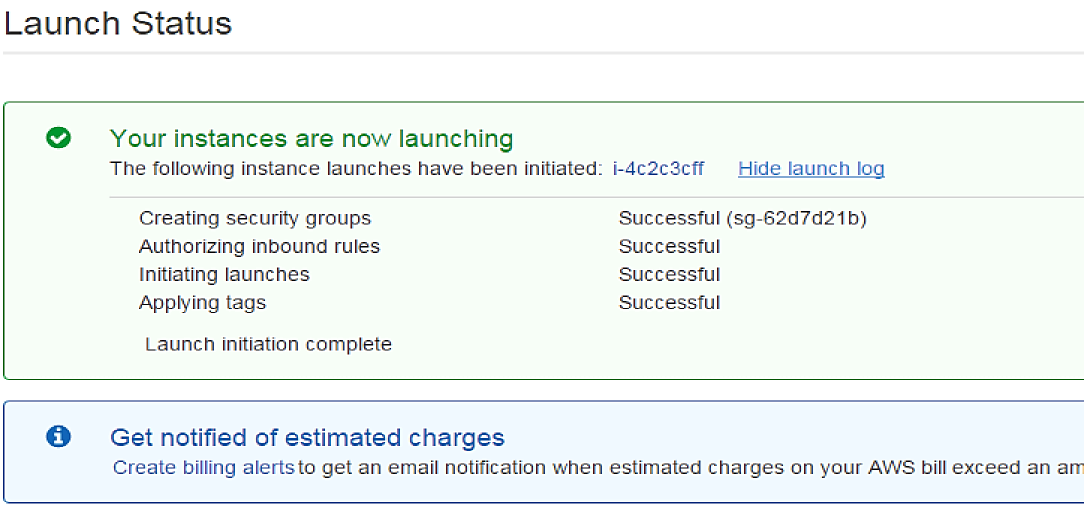
IAM role: None — Create new IAM role

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



Launch Status

Your instances are now launching

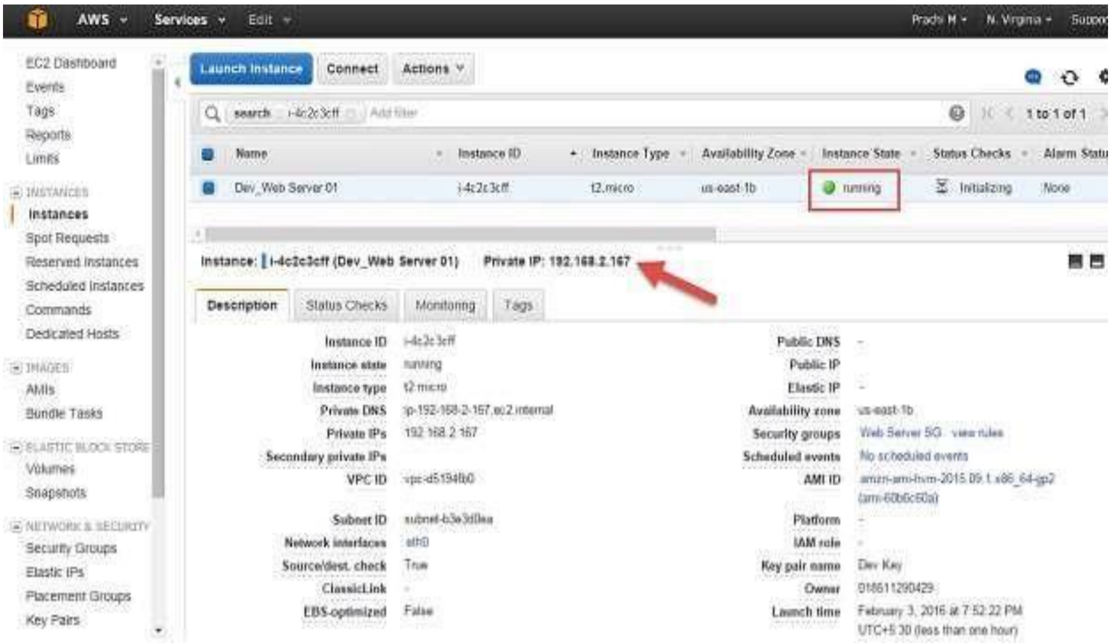
The following instance launches have been initiated: i-4c2c3cff Hide launch log

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	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
Instance state: running
Instance type: t2.micro
Private DNS: ip-192-168-2-167.ec2.internal
Private IPs: 192.168.2.167
Secondary private IPs:
VPC ID: vpc-d51940b0
Subnet ID: subnet-b3e3d0ea
Network interfaces: eth0
Source/dest. check: True
ClassicLink: -
EBS-optimized: False

Public DNS: -
Public IP:
Elastic IP: -
Availability zone: us-east-1b
Security groups: Web Server SG. view rules
Scheduled events: No scheduled events
AMI ID: amzn-ami-hvm-2015.09.1.x86_64-gp2 (ami-60b6c60a)
Platform: -
IAM role: -
Key pair name: Dev Key
Owner: 018611290429
Launch time: February 3, 2016 at 7:52:22 PM UTC+5:30 (less than one hour)

Conclusion:

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