

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 Computeà 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.

The screenshot displays the AWS Management Console's EC2 Dashboard. The top navigation bar includes the AWS logo, 'AWS' dropdown, 'Services' dropdown, and 'Edit' dropdown. The left-hand navigation pane lists various services, with 'EC2 Dashboard' highlighted and a red arrow pointing to it. Below this, categories like INSTANCES, IMAGES, and ELASTIC BLOCK STORE are expanded. The main content area, titled 'Resources', shows a summary of EC2 resources in the 'US East (N. Virginia)' region. A red box highlights the following counts:

Resources in 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	

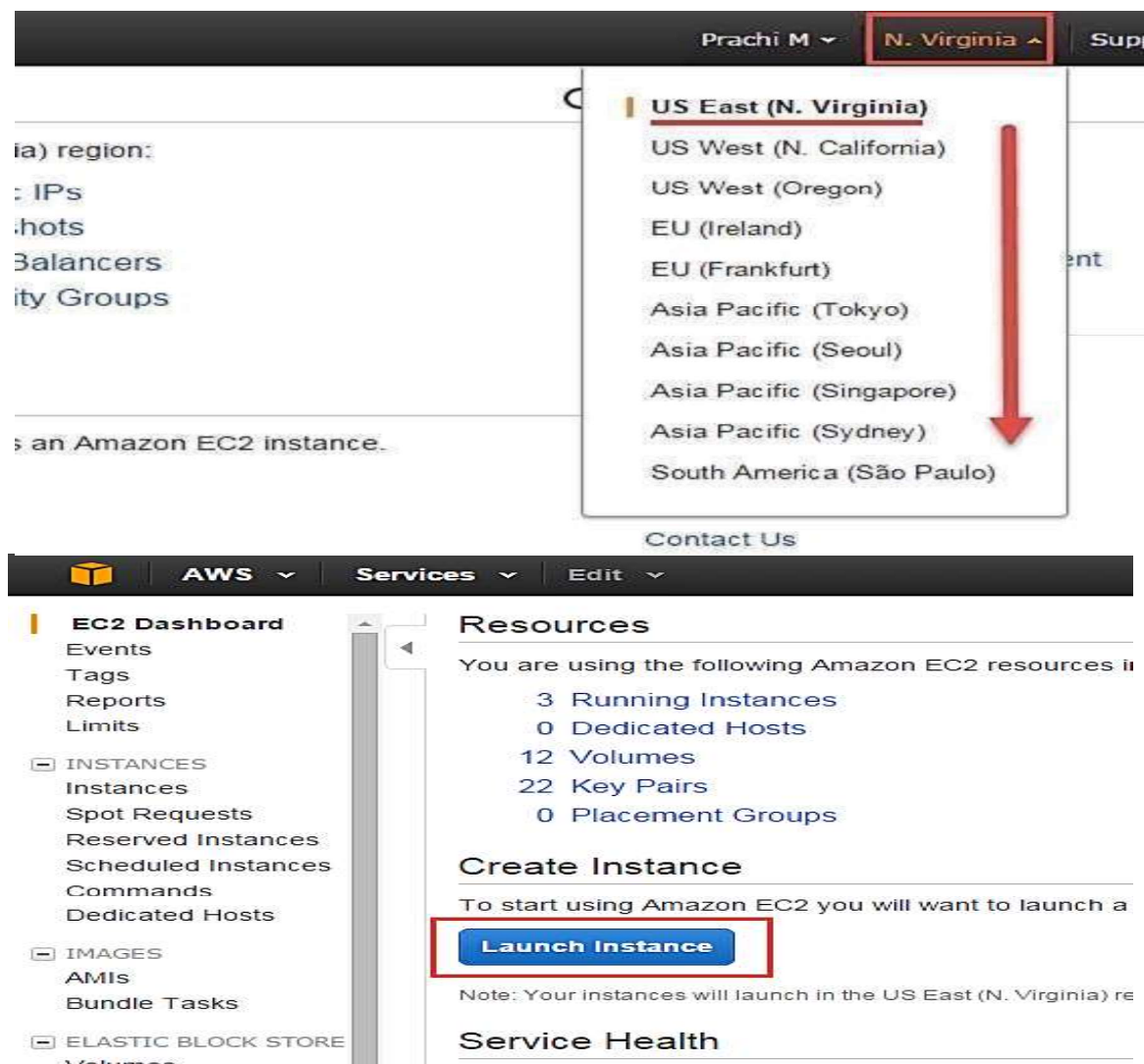
Below the resource summary, there is a promotional banner for Amazon Simple Queue Service and a 'Create Instance' section with a 'Launch Instance' button. A note at the bottom states: 'Note: Your instances will launch in the US East (N. Virginia) region.'

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).



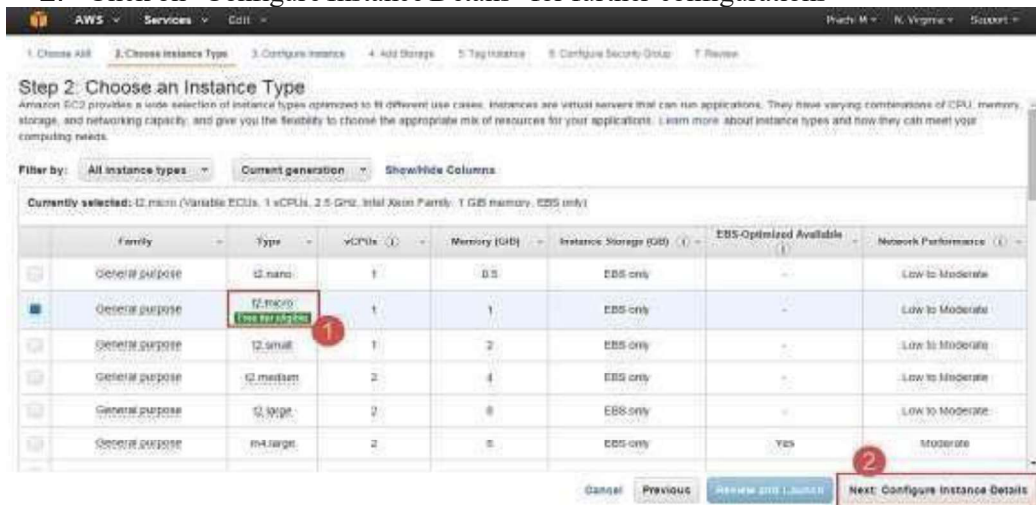
- Instance creation wizard page will open as soon as you click 'LaunchInstance'. Choose AMI
- Step 1) In this step we will do,
1. You will be asked to choose an AMI of your choice. (An AMI is an Amazon Machine Image. It is a template basically of an Operating System platform which you can use as a base 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.



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.

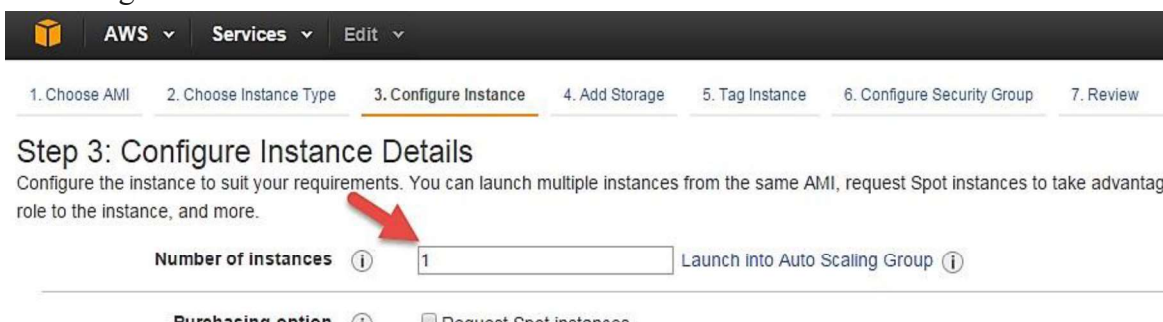
1. We will choose t2.micro instance type, which is a 1vCPU and 1GB memory server offered by AWS.
2. Click on "Configure Instance Details" for further configurations



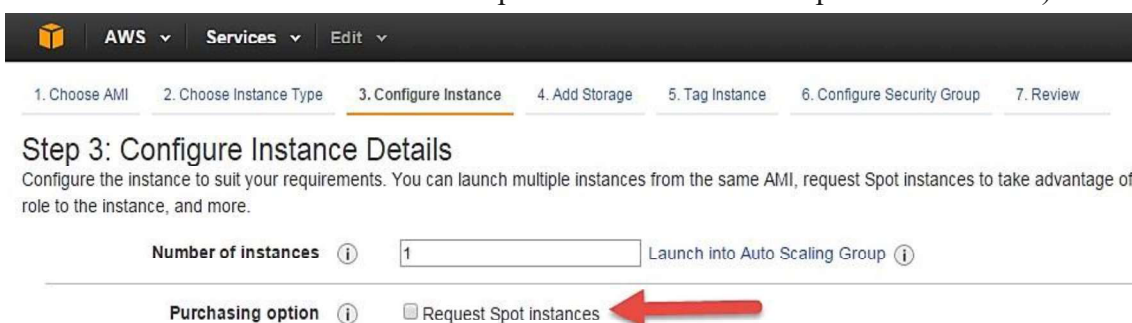
- In the next step of the wizard, enter details like no. of instances you want to launch at a time.
- Here we are launching one

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).



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 of the role to the instance, and more.

Number of instances ① 1 Launch into Auto Scaling Group ①

Purchasing option ① ☐ Request Spot instances

Network ① vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC

Subnet ① vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC

Auto-assign Public IP ① ☐ None

IAM role ① 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 role to the instance, and more.

Number of instances ① 1 Launch into Auto Scaling Group ①

Purchasing option ① ☐ Request Spot instances

Network ① vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC

Subnet ① subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Public subnet2 | us-east-1b

Auto-assign Public IP ① ☐ None

IAM role ① 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.

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

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-Pi Create new subnet
251 IP Addresses available

Auto-assign Public IP Use subnet setting (Disable) 

IAM role Enable  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

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

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

Cancel Previous **Review and Launch** Next: Add Storage 

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 amount.

The screenshot displays the AWS Management Console interface for an EC2 instance. The instance is named 'Dev_Web Server 01' and is in the 'running' state. A red arrow points to the private IP address '192.168.2.167' in the 'Private IP' field. The console shows various details about the instance, including its ID, type, availability zone, and associated security groups.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
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

Description	Status Checks	Monitoring	Tags
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-d51341b0		
Subnet ID	subnet-b3e3d1aa		
Network interfaces	eni-07f0		
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	amazon-ami-hvm-2015.09.1.x86_64-gp2 (ami-60b6c50a)
Platform	-
IAM role	-
Key pair name	Dev Key
Owner	016611290429
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.