

Advance AWS

AWS Assessment Project 1

Student:

Kishore Shinde

Teacher:

Mrs. Vinolin Jeremiah

Course:

Advance AWS Cloud Computing with DevOps
Fundamentals

Institute:

Lets Upgrade

Project: Deploying a Highly Available Web Application and Bastion Host in AWS

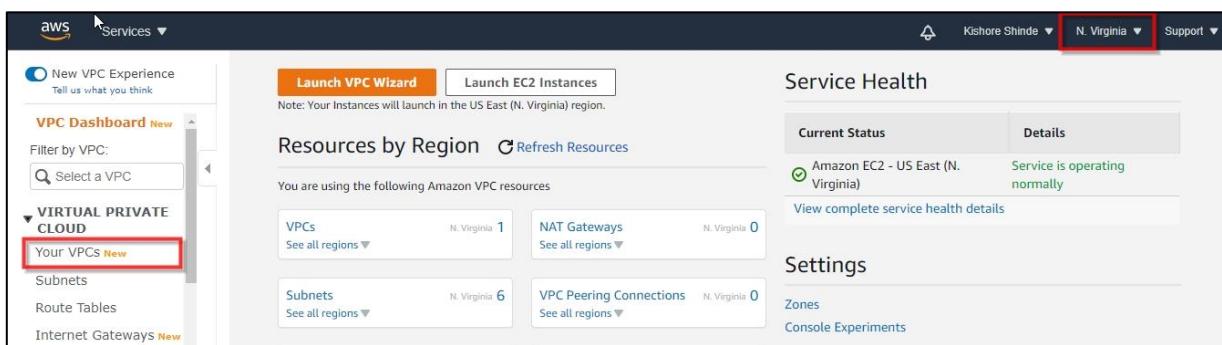
Step 1	Create a VPC with a Public & Private Subnet
Step 2	Create Internet gateway and associate with the Public Subnet
Step 3	Creating NAT Gateway & associate with Private Subnet
Step 4	Launch Bastion Host in Public Subnet
Step 5	Creating a Security Group for the Load Balancer
Step 6	Launch two Web Servers securely in Private Subnet
Step 7	SSH into Web Servers through Bastion Server using RSA private key, Install Apache, Host Page(index.html) on Web Servers
Step 8	Creating additional Public Subnets, each in different availability zone and in same VPC
Step 9	Creating an Application Load Balancer with multiple subnets
Step 10	Checking the health of Load Balancer & Testing DNS
Step 11	Testing High Availability

STEP 1 : Create a VPC with a Public & Private Subnet

- Log into AWS Management Console
- Click on Services and Search for VPC
- Select VPC & do following tasks

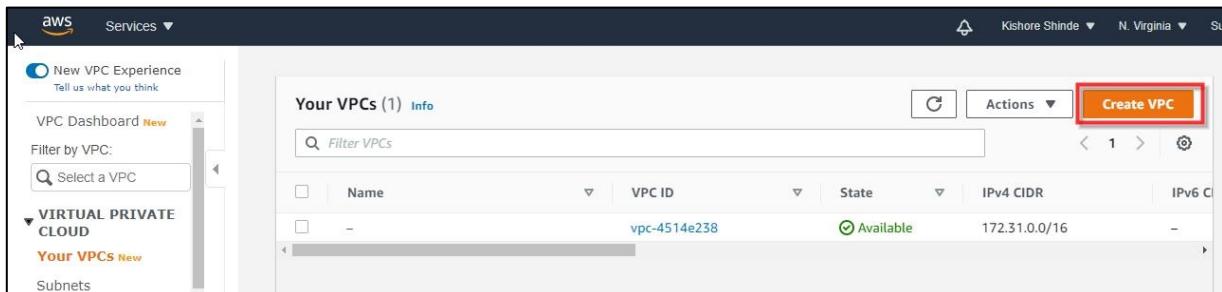
Task 1 : Create VPC

- Make sure you have selected US East (N. Virginia) us-east-1 region.



The screenshot shows the AWS VPC Dashboard. At the top right, the region is set to "N. Virginia". On the left sidebar, under "VIRTUAL PRIVATE CLOUD", the "Your VPCs" link is highlighted with a red box. In the center, there's a "Resources by Region" summary with sections for VPCs, NAT Gateways, Subnets, and VPC Peering Connections. A "Service Health" panel indicates that "Amazon EC2 - US East (N. Virginia)" is operating normally. At the bottom right of the dashboard, there's a "Create VPC" button.

- Click on Your VPCs



The screenshot shows the "Your VPCs" list page. The "Your VPCs (1)" header is visible, and the "Actions" dropdown menu has a "Create VPC" button highlighted with a red box. The table lists one VPC entry: "vpc-4514e238" with state "Available" and IPv4 CIDR "172.31.0.0/16".

- Click on Create VPC

AWS Services ▾ Kishore Shinde ▾ N. Virginia ▾ Support ▾

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

MyVPC

IPv4 CIDR block Info
10.0.0.0/16

IPv6 CIDR block Info
 No IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy Info
Default

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - optional
Name MyVPC Remove Add new tag

You can add 49 more tags.

Cancel **Create VPC**

Feedback English (US) ▾ © 2006 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Name tag : MyVPC
- IPv4 CIDR block : 10.0.0.0/16
- IPv6 CIDR block : No IPv6 CIDR block
- Tenancy : Default
- Click on Create VPC

New VPC Experience Tell us what you think

You successfully created **vp-05fc7fcfd94476a6 / MyVPC**

VPC Dashboard New Filter by VPC: Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs New Subnets Route Tables Internet Gateways New Egress Only Internet Gateways New Carrier Gateways New DHCP Options Sets New Elastic IPs New Managed Prefix Lists New

VPC > Your VPCs > vp-05fc7fcfd94476a6 / MyVPC Actions ▾

Details <small>Info</small>			
VPC ID	State	DNS hostnames	DNS resolution
vp-05fc7fcfd94476a6	Available	Disabled	Enabled
Tenancy	DHCP options set	Route table	Network ACL
Default	dopt-587ae922	rtb-0f67197124a070128	acl-040d5c6ed20ff3685
Default VPC	IPv4 CIDR	IPv6 pool	IPv6 CIDR (Network Border Group)
No	10.0.0.0/16	-	-
Owner ID			
391321345174			

The VPC is now created.

Task 2 : Create Public & Private Subnet

1. Go to Subnets in the left panel of the VPC page

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with 'Subnets' highlighted by a red box. The main area has a table titled 'Subnets' with columns: Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, and IPv6 CIDR. There are six rows of data. At the top right of the main area, there's a 'Create subnet' button also highlighted with a red box.

2. Creating Public Subnet, Click on Create subnet

The screenshot shows the 'Create subnet' wizard. It has fields for 'Name tag' (MyPublicSubnet), 'VPC' (MyVPC), 'Availability Zone' (us-east-1a), and 'IPv4 CIDR block' (10.0.0.0/24). The 'Create' button at the bottom right is highlighted with a red box.

- Name tag : MyPublicSubnet
- VPC : Select MyVPC
- Availability Zone : us-east-1a
- IPv4 CIDR block : 10.0.0.0/24
- Click on Create

Create subnet

The screenshot shows a confirmation message: 'The following Subnet was created:' followed by the Subnet ID: subnet-053f85d7bb6fb903f. A 'Close' button at the bottom right is highlighted with a red box.

You will get a message Subnet was created click on Close

3. Enable Auto Assign Public IP to instances created within this subnet

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with 'VIRTUAL PRIVATE CLOUD' and 'Your VPCs'. In the main area, a table lists subnets. A context menu is open over 'MyPublicSubnet', with 'Actions' expanded. The 'Modify auto-assign IP settings' option is highlighted with a red box.

- Select **MyPublicSubnet**, Click on **Actions**
- Click on **Modify auto-assign IP settings**

The screenshot shows the 'Modify auto-assign IP settings' page. It has two sections: 'Auto-assign IPv4' (with a checked checkbox) and 'Auto-assign Co-IP' (with an unchecked checkbox). At the bottom right are 'Cancel' and 'Save' buttons, with 'Save' highlighted with a red box.

- Select **Enable auto-assign public IPv4 address**
- Click on **Save**

Note: Now the instances launched inside **MyPublicSubnet** will have Public IPs assigned to them by default.

4. Create Private subnet, Click on Create Subnet

The screenshot shows the 'Create subnet' page. It has several input fields: 'Name tag' (MyPrivateSubnet), 'VPC*' (MyVPC), 'Availability Zone' (No preference), and 'IPv4 CIDR block*' (10.0.1.0/24). At the bottom right are 'Cancel' and 'Create' buttons, with 'Create' highlighted with a red box.

- Name tag: **MyPrivateSubnet**

- VPC : Select **MyVPC**
- Availability Zone : **No preference**
- IPv4 CIDR block : **10.0.1.0/24**
- Click on **Create**

You will get a message subnet was created click on **Close**.

You will see two subnets created (Public & Private).

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with options like New VPC Experience, VPC Dashboard, Filter by VPC, and a list of VPCs. The main area has tabs for Create subnet, Actions, and a search bar. A table lists subnets with columns for Name, Subnet ID, State, VPC, IPv4 CIDR, and Available IPv4. Two subnets are visible: MyPublicSubnet (subnet-053f85d7bb6fb903f) and MyPrivateSubnet (subnet-0ec7936b7bd5f7340). Both are in the available state, belong to vpc-05fc7fcfd94476a6 | MyVPC, and have an IPv4 CIDR of 10.0.0.0/24. The MyPrivateSubnet row is selected and highlighted with a blue border.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4
MyPublicSubnet	subnet-053f85d7bb6fb903f	available	vpc-05fc7fcfd94476a6 MyVPC	10.0.0.0/24	251
MyPrivateSubnet	subnet-0ec7936b7bd5f7340	available	vpc-05fc7fcfd94476a6 MyVPC	10.0.1.0/24	251

STEP 2 : Create Internet Gateway & associate with Public Subnet

An internet gateway is a virtual router that connects VPC to the internet.

1. Go to **Internet Gateways** in the left panel of VPC page
2. Click on **Create Internet Gateway**

The screenshot shows the AWS Internet Gateways page. On the left, there's a sidebar with options like New VPC Experience, VPC Dashboard, Filter by VPC, and a list of Internet Gateways. The main area has tabs for Internet gateways (1/1), Actions, and a search bar. A table lists the gateway igw-d41732af, which is attached to the VPC vpc-4514e238. A red box highlights the 'Create internet gateway' button in the top right corner.

Name	Internet gateway ID	State	VPC ID
-	igw-d41732af	Attached	vpc-4514e238

To create new internet gateway specify the name for the gateway below:

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

MyIGW

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - optional

Name MyIGW Remove

Add new tag

You can add 49 more tags.

Create internet gateway

- Name : MyIGW
- Click on Create internet gateway

An Internet gateway is created. Now let us attach VPC.

New VPC Experience Tell us what you think

VIRTUAL PRIVATE CLOUD Your VPCs Subnets Route Tables Internet Gateways Egress Only Internet Gateways Carrier Gateways DHCP Options Sets Elastic IPs Managed Prefix Lists

The following internet gateway was created: igw-0d3bac8fd73341042. You can now attach to a VPC to enable the VPC to communicate with the internet.

igw-0d3bac8fd73341042 / MyIGW

Details Info

Internet gateway ID: igw-0d3bac8fd73341042 State: Detached VPC ID: Owner: 391321345174

Actions Attach to VPC Detach from VPC Manage tags Delete

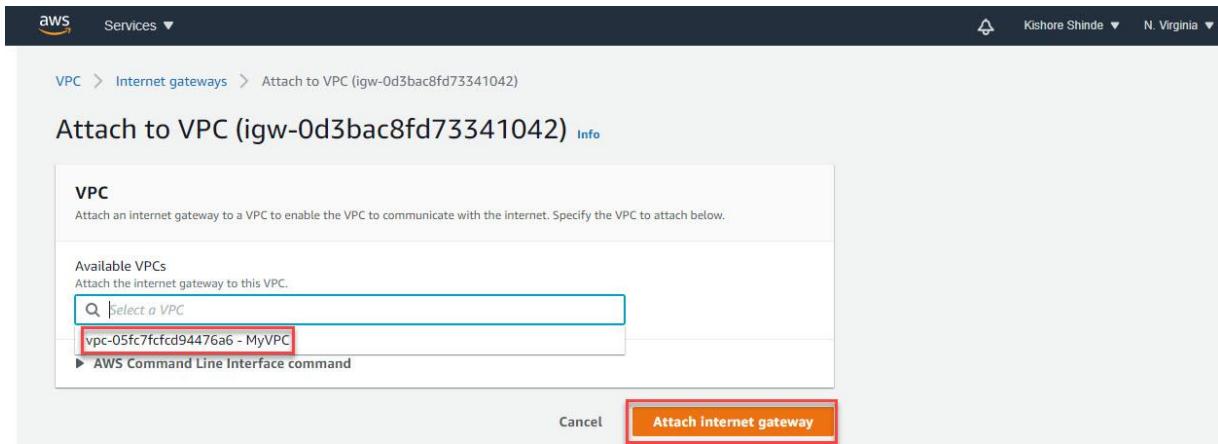
Tags

Search tags

Key Value

Name MyIGW

- Click on Attach to VPC at the top
- or
- Click on Actions and select Attach to VPC



- Select **MyVPC**
- Click on **Attach internet gateway**

You can see the internet gateway (**MyIGW**) is attached to VPC (**MyVPC**)

The screenshot shows the 'Internet Gateways' details page for the gateway igw-0d3bac8fd73341042. A success message at the top states: 'Internet gateway igw-0d3bac8fd73341042 successfully attached to vpc-05fc7fcfd94476a6'. The 'Details' tab is selected, showing the following information:

Internet gateway ID igw-0d3bac8fd73341042	State Attached	VPC ID vpc-05fc7fcfd94476a6 MyVPC	Owner 391321345174
--	-------------------	---	-----------------------

The 'Tags' section shows a single tag: Name = MyIGW. There is a 'Manage tags' button next to the tags table.

Task 4 : Create Public Route Table and Configure

We will create a route table and attach a public subnet to it. Instances launched within this subnet will have access to the Internet.

1. Go to **Route Tables** in the left panel of the VPC page

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Route Tables' section, the 'Create route table' button is highlighted with a red box. The main pane displays a table of existing route tables, with two entries visible: 'rtb-0f67197124a070128' and 'rtb-db9de0a5'. The table includes columns for Name, Route Table ID, Explicit subnet association, Edge associations, Main, and VPC ID.

2. Click on **Create route table**

The screenshot shows the 'Create route table' wizard. In the 'Name tag' field, 'PublicRouteTable' is entered and highlighted with a red box. In the 'VPC*' dropdown, 'vpc-05fc7fcfd94476a6' is selected and highlighted with a red box. Below the dropdown, a list of VPCs is shown, with 'vpc-05fc7fcfd94476a6' and 'MyVPC' highlighted with a red box. At the bottom right, the 'Create' button is highlighted with a red box.

- Name tag : **PublicRouteTable**
- VPC : Select **MyVPC**
- Click on **Create**

You will get a message Route Table was created click on **Close**.

3. To attach Internet Gateway, select **PublicRouteTable**

The screenshot shows the AWS VPC Route Tables interface. On the left sidebar, under 'Route Tables', 'Edit routes' is highlighted. The main area displays a table of routes. The first route, 'rtb-0ace56882584baad5' (Name: PublicRouteTable), has its entire row highlighted with a red box. Below the table, tabs for 'Summary', 'Routes' (which is selected and highlighted with a red box), 'Subnet Associations', 'Edge Associations', 'Route Propagation', and 'Tags' are visible. At the bottom, a table shows route details: Destination 10.0.0.0/16, Target local, Status active, and Propagated No. A red box highlights the 'Edit routes' button at the top of the 'Routes' section.

4. In the **Routes** tab, Click on **Edit routes**

5. On the next screen click on **Add route**

The screenshot shows the 'Edit routes' screen. At the top, it says 'Route Tables > Edit routes'. Below is a table with columns: Destination, Target, Status, and Propagated. The first row has '10.0.0.0/16' in the Destination column and 'local' in the Target column. The second row has '0.0.0.0/0' in the Destination column and 'igw-' in the Target dropdown menu. A red box highlights the '0.0.0.0/0' entry. Another red box highlights the 'igw-' dropdown. At the bottom right, there are 'Cancel' and 'Save routes' buttons, with 'Save routes' highlighted by a red box. A note at the bottom left says '* Required'.

- Destination : Enter **0.0.0.0/0**
- Target : Select **Internet Gateway**, and then from the list select **MyIGW**
- Click on **Save routes**

You will get a message Routes successfully edited. Click on **Close**.

Note : We have created the route table, edited the routes and attached the internet gateway, now the resources inside the public subnet can communicate to the internet.

Now we will associate the public subnet to the route table.

- Select the route table : **PublicRouteTable**

The screenshot shows the AWS VPC Route Tables page. On the left, there's a sidebar with 'Route Tables' selected under 'VIRTUAL PRIVATE CLOUD'. In the main area, a table lists route tables. One row for 'PublicRouteTable' is selected and highlighted with a blue border. A context menu is open over this row, with the 'Edit subnet associations' option highlighted by a red box. Other options in the menu include 'Set Main Route Table', 'Delete Route Table', 'Edit edge associations', 'Edit route propagation', 'Edit routes', and 'Add/Edit Tags'. Below the table, tabs for 'Summary', 'Routes', 'Subnet Associations', 'Edge Associations', 'Route Propagation', and 'Tags' are visible. The 'Routes' tab is selected.

- Click on Actions & select **Edit subnet associations**
- On the next screen, select **MyPublicSubnet**

This screenshot shows the 'Edit subnet associations' dialog for the 'PublicRouteTable'. At the top, it says 'Associated subnets' and shows 'subnet-053f85d7bb6fb903f'. Below is a table listing subnets:

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0ec7936b7bd5f7340 MyPrivateSubnet	10.0.1.0/24	-	Main
subnet-053f85d7bb6fb903f MyPublicSubnet	10.0.0.0/24	-	Main

At the bottom, there are 'Cancel' and 'Save' buttons. The 'Save' button is highlighted with a red box.

- Click on **Save**

Once the configurations are completed, it will look like below :

The screenshot shows the AWS VPC Route Tables page. On the left, there's a sidebar with 'Route Tables' selected. The main area displays a table of route tables. One row is highlighted with a red box, showing 'PublicRouteTable' as the name, 'rtb-0ace56882584baad5' as the Route Table ID, and 'Explicit subnet association' under Subnet Associations. Another row is also highlighted with a red box, showing 'rtb-0f67197124a070128'. Below the table, a section titled 'Route Table: rtb-0ace56882584baad5' shows a 'Routes' tab selected. It lists two routes: one for '10.0.0.0/16' target 'local' status 'active' and propagated 'No'; and another for '0.0.0.0/0' target 'igw-0d3bac8fd73341042' status 'active' and propagated 'No'.

You can see a subnet associated with **PublicRouteTable**.

- Check it is not the **Main** route table
- Check the internet gateway attached is also active

You can also see there is another existing route table already available for **MyVPC**.

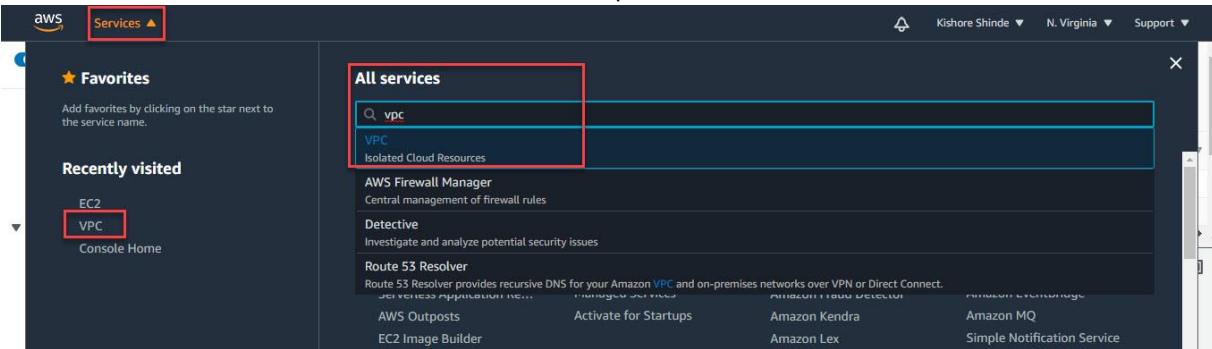
It is the **Main** route table created at the time the VPC was created. We will use it when we create **NAT Gateway**.

Note : Make sure the **PublicRouteTable** is not the **Main** route table.

Step 3 : Creating NAT Gateway & associate with Private Subnet

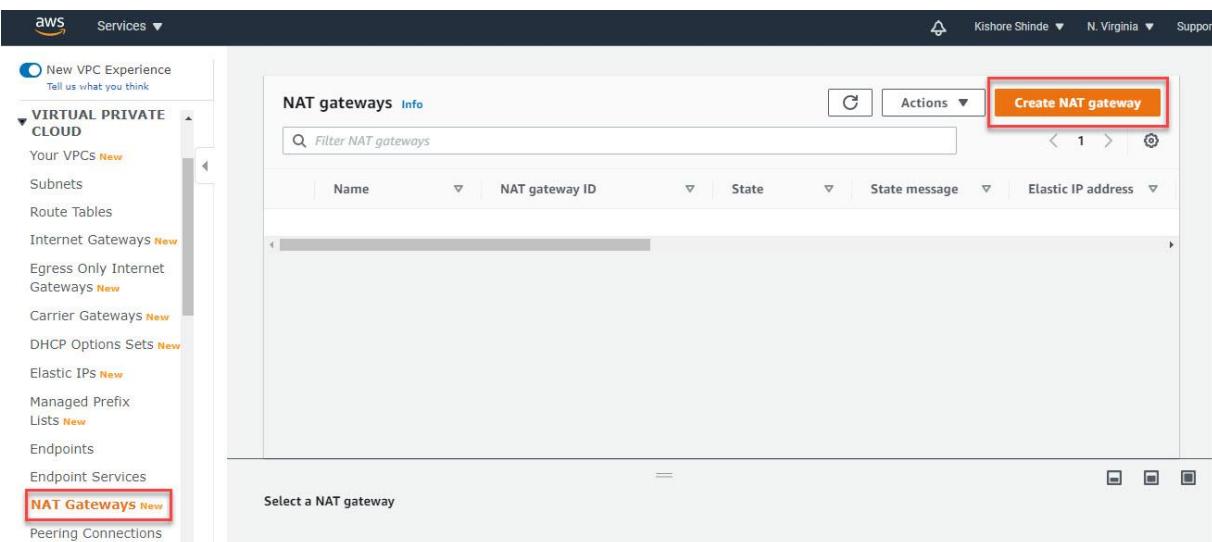
Task 1 : Create NAT Gateway

1. Click on Services at the left top



2. Select VPC from the Recently visited or Search for VPC under All services and select it

3. Select NAT Gateways



4. Click on Create NAT gateway

5. Create NAT gateway :

The screenshot shows the 'Create NAT gateway' configuration page in the AWS VPC service. The 'Name - optional' field contains 'MyNAT'. The 'Subnet' dropdown is set to 'subnet-053f85d7bb6fb903f (MyPublicSubnet)'. The 'Elastic IP allocation ID' dropdown contains 'eipalloc-0036637cd46467a9b'. A blue button labeled 'Allocate Elastic IP' is visible next to the dropdown. Below these fields is a 'Tags' section with one tag named 'Name' with value 'MyNAT'. At the bottom right is an orange 'Create NAT gateway' button.

- Name : **MyNAT**
- Subnet : Select **MyPublicSubnet**
- Elastic IP allocation ID : Click on **Allocate Elastic IP**
An elastic ip is allocated to the NAT gateway.
- Click on **Create NAT gateway**.

Note : 1. We are creating the NAT gateway in Public subnet because when you have to give internet access to private subnet the NAT Gateway has to be in public subnet.
2. We are allocating the Elastic IP because we don't want the configuration to be changed.

6. We will see the NAT gateway is created.

- Wait for the **State** to change from **Pending** to **Available**.

Task 2 : Update Route Table and Configure NAT Gateway

1. Go to **Route Tables** in the left panel.

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
PublicRouteTable	rtb-0ace56882584baad5	subnet-053f85d7bb6fb903f	-	No	vpc-05fc7fcfd94476a6 MyVPC	39132
	rtb-0f67197124a070128	-	-	Yes	vpc-05fc7fcfd94476a6 MyVPC	39132
	rtb-db9de0a5	-	-	Yes	vpc-4514e238	39132

2. You can see two route tables available for **MyVPC**
3. Select the **Main** route table (**Main : Yes**)
4. In **Routes** tab click on **Edit Routes**
5. In next page **Edit Routes** :
 - Click on **Add route**

Route Tables > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	nat-04e0d6a6a7d8c5d7f		No
Add route	nat-04e0d6a6a7d8c5d7f MyNAT		

* Required

Cancel Save routes

- Destination : Enter 0.0.0.0/0
- Target : Select **NAT Gateway**, then select **MyNAT**
- Click on **Save routes**

6. You will get a message “Routes successfully edited”. Click on **Close**.

7. NAT Gateway is configured. In **Routes** tab, Check the NAT Gateway is active.

New VPC Experience
Tell us what you think

VPC Dashboard New

Filter by VPC:
Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs New

Subnets

Route Tables

Internet Gateways New

Egress Only Internet Gateways New

Carrier Gateways New

DHCP Options Sets New

Elastic IPs New

Managed Prefix Lists New

Create route table Actions ▾

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
PublicRouteTable	rtb-0ace56882584baad5	subnet-053f85d7bb6fb903f	-	No	vpc-05fc7fcfd94476a6 MyVPC	39132
rtb-0f67197124a070128	-	-	-	Yes	vpc-05fc7fcfd94476a6 MyVPC	39132
rtb-dh91feba5	-	-	-	Yes	vpc-4514ea238	39132

Route Table: rtb-0f67197124a070128

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View All routes

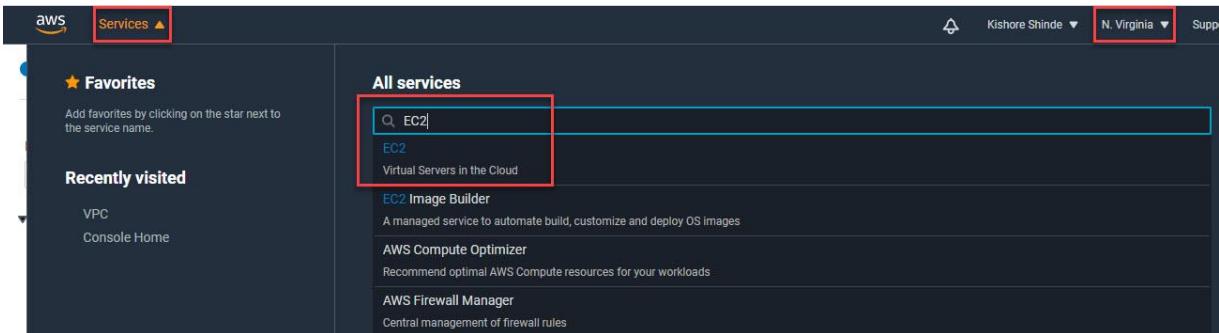
Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	nat-04e0d6a6a7d8c5d7f	active	No

Note : We have created a private subnet and NAT gateway. The private subnet is attached to a route table, to route traffic via NAT gateway to the internet.

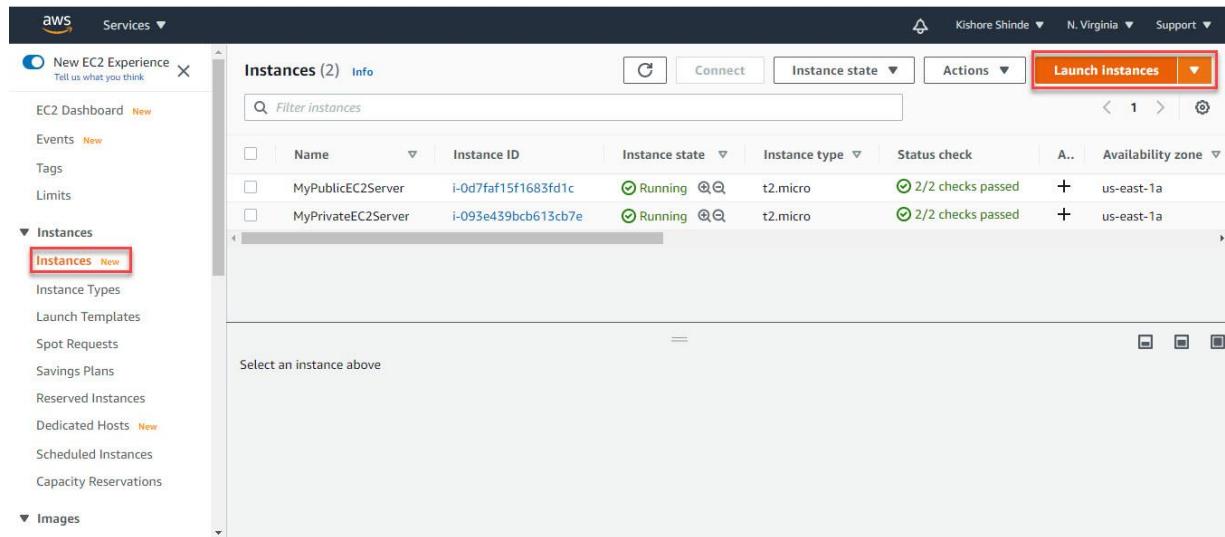
Step 4 Launch Bastion Host in Public Subnet

Bastion Host is a computer that acts like a proxy server that allows the client machine to connect to the remote server. It filters the incoming traffic and prevents unwanted connections entering the network.

1. Make sure you have selected **N. Virginia** region
2. Click on **Services** on the top and search for **EC2** & select it



3. Go to **Instances** & click on **Launch Instances**



4. Choose AMI : Choose the Amazon Linux 2 AMI (HVM), SSD Volume Type click on the Select button.

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.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start

	My AMIs	AWS Marketplace	Community AMIs
<input checked="" type="checkbox"/> Free tier only (i)	Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0947d2ba12ee1ff75 (64-bit x86) / ami-007a607c4abd192db (64-bit Arm)		
	Free tier eligible	Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.	
		Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	
			Select
		Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-098f16afa9edf40be (64-bit x86) / ami-029ba835ddd43c34f (64-bit Arm)	
		Free tier eligible Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type	
			Select
			64-bit (x86) 64-bit (Arm)

5. Instance Type : Select t2.micro (free-tier eligible)

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 families (i) Current generation (i) Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, 1 GiB memory, EBS only)

	Family	Type	vCPUs (i)	Memory (GiB) (i)	Instance Storage (GB) (i)	EBS-Optimized Available (i)	Network Performance (i)	IPv6 Support (i)
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

Click on Next : Configure Instance Details

6. Configure Instance Details :

The screenshot shows the 'Configure Instance Details' step of the AWS EC2 instance creation wizard. The 'Network' section is highlighted with a red box, showing settings for 'vpc-4514e238 (default)', 'No preference (default subnet in any Availability Zone)', and 'Use subnet setting (Enable)'. Below this, the 'Placement group' and 'Capacity Reservation' sections are shown. At the bottom right, buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Storage' are visible, with 'Next: Add Storage' highlighted by a red box.

- As we are creating Bastion Host it will be in Public Subnet so let all settings be default.
- Click on **Next : Add Storage**

7. Add Storage : No need to change anything in this step and click on **Next : Add Tags**

8. Add Tags : Click on **Add Tag**

The screenshot shows the 'Add Tags' step of the AWS EC2 instance creation wizard. A single tag is being added with the key 'Name' and value 'Bastion-Server'. The 'Instances' and 'Volumes' checkboxes are checked. At the bottom right, buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Security Group' are visible, with 'Next: Configure Security Group' highlighted by a red box.

- Key : Name
- Value : Bastion-Server
- Click on **Next : Configure Security Group**

9. Configure Security Group :

- Assign a security group – Select Create a new security group

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:
Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom	0.0.0.0/0

Add Rule

Warning
 Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Review and Launch

- Security Group Name : **Bastion-SG**
- Description : **Security group for Bastion Server**
- Make sure you select type : **SSH** (*will be default selected*)
- Source : **Custom – 0.0.0.0/0**
- Click on **Review and Launch**

10. Review Instance Launch : Review all the settings

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 2 AMI (HVM), SSD Volume Type - ami-0947d2ba12ee1ff5
 Free tier eligible
 Root Device Type: ebs Virtualization type: hvm

Instance Type
 t2.micro - 1 vCPU, 1 GiB Memory, EBS only, Low to Moderate Network Performance

Security Groups
 Bastion-SG (Security group for Bastion Server)

Tags
 Name: Bastion-Server

Launch

- Click on **Launch**

10. Key Pair :

- Create a new key pair : Bastionserver

Select an existing key pair or create a new key pair X

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](#).

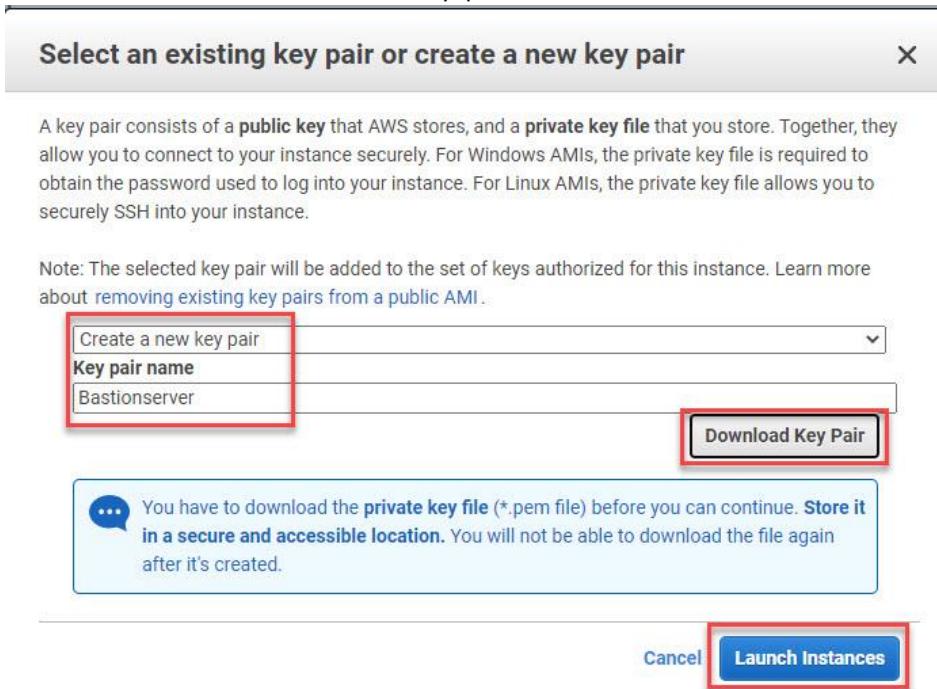
Create a new key pair

Key pair name
Bastionserver

Download Key Pair

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.

Cancel Launch Instances



- Click on **Download Key Pair**
- Click on **Launch Instances**

11. Launch Status :

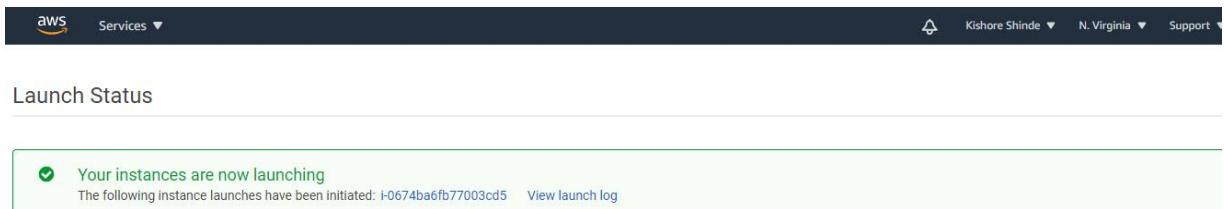
- Select the instance id

aws Services ▾ Kishore Shinde ▾ N. Virginia ▾ Support ▾

Launch Status

Your instances are now launching
The following instance launches have been initiated: i-0674ba6fb77003cd5 [View launch log](#)

- Wait for 1-2 minutes (until Bastion Server Instance state changes from **pending** to **running** and Status check : **2/2 checks passed**)



12. Bastion Host Details : Bastion Host is launched successfully.

The screenshot shows the AWS EC2 Instances page. A single instance, "Bastion-Server" (Instance ID: i-0674ba6fb77003cd5), is listed. The instance is running, of type t2.micro, and has 2/2 checks passing with no alarms. It is located in the us-east-1a availability zone. The Public IPv4 address is highlighted with a red box and is shown as 107.23.177.149.

- Bastion Host Associated Public IP : 107.23.177.149

Step 5 Creating a Security Group for the Load Balancer

1. On the EC2 Dashboard, Scroll down the left panel and Select **Security Groups**

The screenshot shows the AWS EC2 Security Groups page. It lists five security groups: launch-wizard-1, default, Bastion-SG, launch-wizard-2, and default. The "Create security group" button is highlighted with a red box.

2. Click on **Create security group**

AWS Services ▾ Kishore Shinde ▾ N. Virginia ▾ Support ▾ ⓘ

VPC > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name <small>Info</small>	LoadBalancer-SG
Name cannot be edited after creation.	
Description <small>Info</small>	Security group for the Load balancer
VPC <small>Info</small>	vpc-05fc7fcfd94476a6 (MyVPC)

Inbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
HTTP	TCP	80	Custom	Q 0.0.0.0/0 X
Delete				

[Add rule](#)

Outbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Destination <small>Info</small>	Description - optional <small>Info</small>
All traffic	All	All	Custom	Q 0.0.0.0/0 X
Delete				

[Add rule](#)

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tag

[Cancel](#) [Create security group](#)

Feedback English (US) ▾ © 2008–2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Security Group Name : **LoadBalancer-SG**
- Description : **Security group for the Load balancer**
- VPC : **MyVPC**
- Inbound rules :
 - Click on **Add rule**
 - Type : Select **HTTP**, make sure Protocol is **TCP** and Port range is **80**
- Source : **Custom** and select **0.0.0.0/0**
- Outbound rules : **Leave as default**
- Tags-optional : **Leave as default**
- Click on **Create security group**

- The security group for the load balancer will be created.

The screenshot shows the AWS Security Groups console. On the left, there's a navigation pane with categories like VPCs, Endpoint Services, NAT Gateways, Peering Connections, SECURITY (Network ACLs, Security Groups), VIRTUAL PRIVATE NETWORK (VPN) (Customer Gateways, Virtual Private Gateways, Site-to-Site VPN Connections, Client VPN Endpoints), and TRANSIT GATEWAYS (Transit Gateways). The main area displays a security group named "sg-0815f9dc3dd43e05f - LoadBalancer-SG". The "Details" section includes fields for Security group name (LoadBalancer-SG), Security group ID (sg-0815f9dc3dd43e05f), Description (Security group for the Load balancer), and VPC ID (vpc-05fc7fcfd94476a6). Below this, there are sections for Owner (391321345174), Inbound rules count (1 Permission entry), and Outbound rules count (1 Permission entry). The "Inbound rules" tab is selected, showing a single rule: Type (HTTP), Protocol (TCP), Port range (80), Source (0.0.0.0/0), and Description - optional (empty). An "Edit inbound rules" button is visible at the top right of the rules table.

- Check the Inbound rules with added HTTP rule

Step 6 Launch two Web Servers securely in Private Subnet

We have created a private subnet and NAT gateway. The private subnet is attached to a route table to route traffic via NAT gateway to the internet. Now let us create two web servers in private subnet.

Task 1 : Launching Web Server 1

- Make sure you have selected **N. Virginia** region
- Click on **Services** on the top and search for **EC2** & select it

The screenshot shows the AWS Services console. At the top, there's a search bar with "EC2" typed into it. Below the search bar, there's a list of services under "All services": EC2 (Virtual Servers in the Cloud), EC2 Image Builder (A managed service to automate build, customize and deploy OS images), AWS Compute Optimizer (Recommend optimal AWS Compute resources for your workloads), and AWS Firewall Manager (Central management of firewall rules). To the left, there's a sidebar with "Favorites" (Add favorites by clicking on the star next to the service name) and "Recently visited" (VPC, Console Home).

- Go to **Instances** & click on **Launch Instances**

The screenshot shows the AWS EC2 Instances page. A single instance, 'Bastion-Server' (Instance ID: i-056d4459679907609), is listed as 'Running'. The 'Launch Instances' button at the top right is highlighted with a red box.

4. Choose AMI : Choose the Amazon Linux 2 AMI (HVM), SSD Volume Type click on the Select button.

The screenshot shows the 'Choose AMI' step of the EC2 wizard. It lists two AMIs: 'Amazon Linux 2 AMI (HVM), SSD Volume Type' and 'Red Hat Enterprise Linux 8 (HVM), SSD Volume Type'. The first one is selected, and its 'Select' button is highlighted with a red box. A checkbox for 'Free tier only' is also highlighted with a red box.

5. Instance Type : Select t2.micro (free-tier eligible)

The screenshot shows the 'Choose Instance Type' step of the EC2 wizard. It lists several t2 instance types: t2.nano, t2.micro (selected and highlighted with a red box), t2.small, t2.medium, and t2.large. The 'Review and Launch' button at the bottom right is highlighted with a red box.

Click on Next : Configure Instance Details

6. Configure Instance Details :

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

Purchasing option: Request Spot Instances

Network: vpc-05fc7fcfd94476a6 | MyVPC

Subnet: subnet-0ec7936b7bd5f7340 | MyPrivateSubnet | us-
250 IP Addresses available

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

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory

Cancel Previous Review and Launch Next: Add Storage

- Network : **MyVPC**
- Subnet : **MyPrivateSubnet**
- Auto-assign Public IP : **Use subnet setting (Disable)**
- Leave all the other fields be default.
- Click on **Next : Add Storage**

7. Add Storage : No need to change anything in this step and click on **Next : Add Tags**

8. Add Tags : Click on **Add Tag**

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key: Name (128 characters maximum) Value: Web-server-1 (256 characters maximum)

Instances: Volumes:

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

- Key : **Name**
- Value : **Web-server-1**
- Click on **Next : Configure Security Group**

9. Configure Security Group :

- Assign a security group – Select **Create a new security group**

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:
Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom	sg-0b76b92bb96236d50 e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	Lo e.g. SSH for Admin Desktop

Add Rule

Bastion-SG

sg-0d2cec156aa2d35d9 - LoadBalancer-SG

Cancel Previous Review and Launch

- Security Group Name : web-server-SG
- Description : **Security group for web servers**
- Make sure you select Type : **SSH** (*will be default selected*)
- Source : **Custom** – type Bastion and select **Bastion-SG**

Note : *On port 22, we select **Bastion -SG** security group as its source to allow SSH connection to web servers from only bastion server by restricting the public SSH connection.*

- Click on Add Rule
- Select Type : **HTTP**
- Source : **Custom** – type Load and select **LoadBalancer-SG**

Note : *On port 80, we select **LoadBalancer-SG** as its source to serve the traffic coming through the load balancer.*

- Click on **Review and Launch**

10. Review Instance Launch : Review all the settings

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 2 AMI (HVM), SSD Volume Type - ami-0947d2ba12ee1ff75
Free tier eligible

Instance Type

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

Security Groups

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	sg-0b76b92bb96236d50	Bastion-SG
HTTP	TCP	80	sg-0d2cec156aa2d35d9	LoadBalancer-SG

Tags

Key	Value	Instances	Volumes
Name	Web-server-1		

Buttons: Edit AMI, Edit instance type, Edit security groups, Edit tags, Cancel, Previous, **Launch**

- Click on **Launch**

10.Key Pair :

- Choose an existing key pair
- Select a key pair : **MyKey.pem**

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](#).

Choose an existing key pair

Select a key pair

MyKey

I acknowledge that I have access to the selected private key file (MyKey.pem), and that without this file, I won't be able to log into my instance.

Cancel **Launch Instances**

- Select : I acknowledge.. checkbox
- Click on **Launch Instances**

11. Launch Status :

- Select the instance id

The screenshot shows the AWS Launch Status page. At the top, there's a green success message: "Your instances are now launching. The following instance launches have been initiated: i-035695a80a3220444" with a link to "View launch log". Below this, there's a table with two rows:

Name	Instance ID	Instance state	Status check	Actions
Bastion-Server	i-0ae2e1afce96ddbf	Running	2/2 checks passed	[Launch]
Web-server-1	i-035695a80a3220444	Running	2/2 checks passed	[Launch]

- Wait for 1-2 minutes (until Bastion Server Instance state changes from **pending** to **running** and Status check : **2/2 checks passed**)

12. Web-server-1 Details : You will see new instances Web-server-1 running along with Bastion-server created in the earlier step.

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the "Instances" section, "Web-server-1" is selected. The main pane displays the instance details for "Web-server-1" (i-035695a80a3220444). The "Details" tab is active, showing the following information:

Instance ID	Public IPv4 address	Private IPv4 addresses
i-035695a80a3220444 (Web-server-1)	-	10.0.1.92
Instance state	Public IPv4 DNS	Private IPv4 DNS
Running	-	ip-10-0-1-92.ec2.internal

- Web-server-1 Private IP : 10.0.1.92

13. Repeat the above steps of **Web-server-1**, from **Step 1** to **Step 4** to create **Web-server-2**. We will need 2 instances in private subnet.

14. On **Step 5**: Add Tags page, Click on **Add Tag** button, and enter below details.

Name: **Web-server-2**

AWS Services ▾ Kishore Shinde ▾ N. Virginia ▾ Support ▾

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

Step 5: Add Tags
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances ⓘ	Volumes ⓘ
Name	Web-server-2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Add another tag (Up to 50 tags maximum)		<input type="button" value="X"/>	

Cancel Previous Review and Launch Next: Configure Security Group

14. Step 6 : Configure Security Group section, select existing group WebserverSG

AWS Services ▾ Kishore Shinde ▾ N. Virginia ▾ Support ▾

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

Step 6: Configure Security Group
Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
sg-0810f6ef25a00c50b	Bastion-SG	Security Group for Bastion-server	<input type="button" value="Copy to new"/>
sg-07cbe01b110a0ebfa	default	default VPC security group	<input type="button" value="Copy to new"/>
sg-052c8591f9a497064	launch-wizard-1	launch-wizard-1 created 2020-11-04T14:50:31.250+05:30	<input type="button" value="Copy to new"/>
sg-0fb8f0971e166bb9c	launch-wizard-2	launch-wizard-2 created 2020-11-04T15:45:55.018+05:30	<input type="button" value="Copy to new"/>
sg-0815f9dc3dd43e05f	LoadBalancer-SG	Security group for the Load balancer	<input type="button" value="Copy to new"/>
sg-0e3b4f846479af30d	web-server-SG2	Security group for web servers	<input type="button" value="Copy to new"/>
sg-0eabf5078f27f3e0f	WebserverSG	Security Group for Web Servers	<input type="button" value="Copy to new"/>

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
HTTP	TCP	80	sg-0815f9dc3dd43e05f (LoadBalancer-SG)	
SSH	TCP	22	sg-0810f6ef25a00c50b (Bastion-SG)	

Cancel Previous Review and Launch

15.Click on **Review and Launch** and then Click on **Launch** and select a key pair, **MyKey.pem**, and Launch the instance.

16.In **Launch Status** we can see Your instances are now launching.

AWS Services ▾ Kishore Shinde ▾ N. Virginia ▾ Support ▾

Launch Status

✓ Your instances are now launching
The following instance launches have been initiated: i-07d3eca613a38ab3b [View launch log](#)

17.Click on the instance id & go to EC2 Dashboard – You can see the Web-server-2 is running and the **Private IP** is **10.0.1.194**

18. Now you will see three servers running namely :

- Bastion-server,
- Web-server-1
- Web-server-2

The screenshot shows the AWS EC2 Instances page. The left sidebar has 'Instances' selected. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	A..	Availability zone
Bastion-Server	i-0ae2e1a6fce96ddbf	Running	t2.micro	2/2 checks passed	+	us-east-1a
Web-server-1	i-035695a80a3220444	Running	t2.micro	2/2 checks passed	+	us-east-1a
Web-server-2	i-07d3eca613a38ab3b	Running	t2.micro	2/2 checks passed	+	us-east-1a

Below the table, the details for 'Web-server-2' are shown:

Instance: i-07d3eca613a38ab3b (Web-server-2)

Details Security Networking Storage Status Checks Monitoring Tags

Instance summary Info

Instance ID i-07d3eca613a38ab3b (Web-server-2)	Public IPv4 address -	Private IPv4 addresses 10.0.1.194
---	--------------------------	--------------------------------------

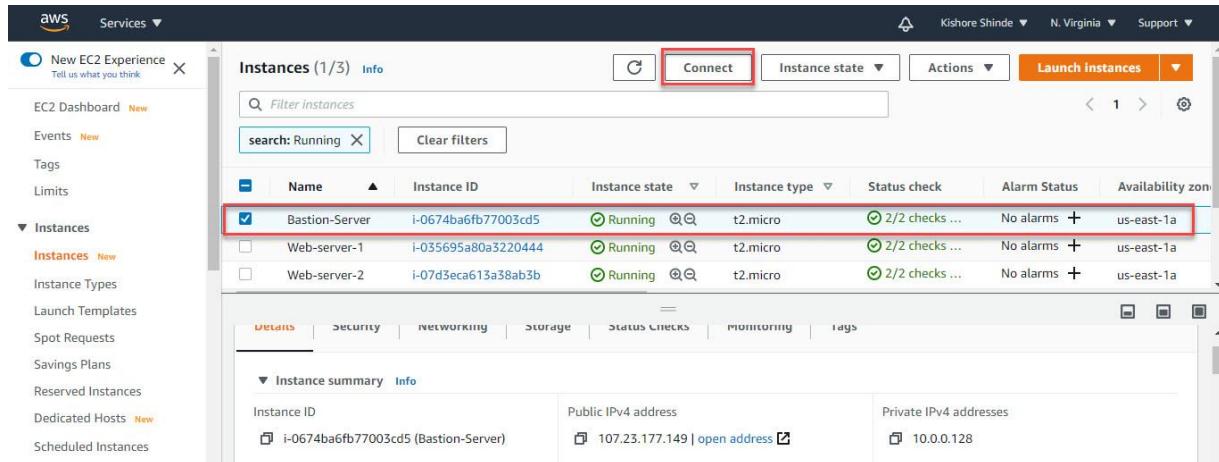
a. Web-server-1 Private IP : 10.0.1.92

b. Web-server-2 Private IP : 10.0.1.194

**Step 7 : SSH into Web Servers through Bastion Server using RSA private key,
Install Apache, Host Page(index.html) on both Web Servers**

Task 1 : Connecting to Bastion-Server

1. Go To EC2 Dashboard & Click on Instances

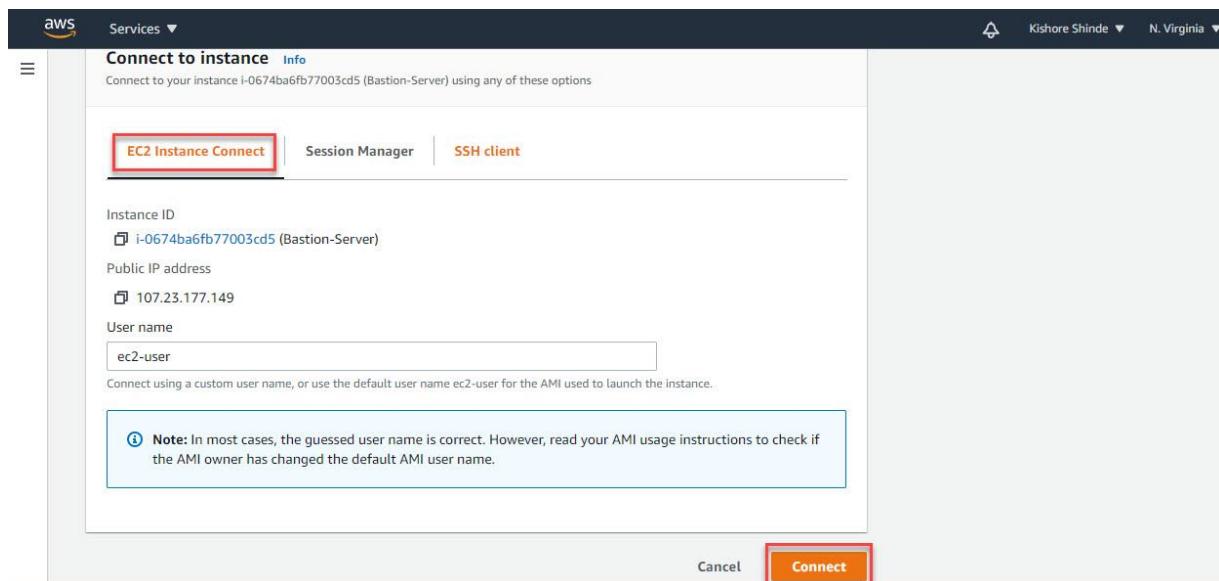


The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, 'Instances' is selected. In the main area, there is a table titled 'Instances (1/3)'. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm Status, and Availability zone. There are three rows:

Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone
Bastion-Server	i-0674ba6fb77003cd5	Running	t2.micro	2/2 checks ...	No alarms	us-east-1a
Web-server-1	i-035695a80a3220444	Running	t2.micro	2/2 checks ...	No alarms	us-east-1a
Web-server-2	i-07d3eca613a38ab3b	Running	t2.micro	2/2 checks ...	No alarms	us-east-1a

Below the table, there are tabs for Details, Security, Networking, Storage, Status Checks, Monitoring, and Tags. Under the Details tab, there is an 'Instance summary' section with fields for Instance ID, Public IPv4 address, and Private IPv4 addresses.

2. From the list of instances, select **Bastion-Server** and click on **Connect**.
3. Connect the instance through EC2 Instance Connect.



The screenshot shows the 'Connect to instance' dialog for the Bastion-Server instance. At the top, it says 'Connect to your instance i-0674ba6fb77003cd5 (Bastion-Server) using any of these options'. Below this, there are three tabs: 'EC2 Instance Connect' (highlighted with a red box), 'Session Manager', and 'SSH client'. The 'EC2 Instance Connect' tab is active. It displays the Instance ID (i-0674ba6fb77003cd5), Public IP address (107.23.177.149), and User name (ec2-user). A note below states: 'Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.' A note at the bottom left says: 'Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.' At the bottom right, there are 'Cancel' and 'Connect' buttons, with 'Connect' highlighted with a red box.

4. Click on **Connect**.
5. You will be connected to the server.

- Type : sudo su – You will be switched to super/root user

```
Last login: Sun Nov  8 16:45:15 2020 from ec2-18-206-107-24.compute-1.amazonaws.com
[ec2-user@ip-10-0-0-128 ~]$ sudo su
[ec2-user@ip-10-0-0-128 ~]$
```

i-0674ba6fb77003cd5 (Bastion-Server)

Public IPs: 107.23.177.149 Private IPs: 10.0.0.128

Task 2 : Connect Web-servers through Bastion-Server

- To connect to the web servers via Bastion-server, we will need the web server key which we used to launch the web servers.(MyKey.pem)
- Open the MyKey.pem file on your local system and copy it's content.

MyKey.pem - Notepad

File Edit Format View Help

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAI D+5m8b32NufVoGa1XBnkxqOYW14Yrh oWhdxeo77eWFbeR4g
jqj1GYSQppR221OKWfCksU+A85vyiR6hbhZy2qh0ct6vD5zuYQ24A1aGrfrK8jm
rLQx7Lqm77XdfUesN91Pql+7SFeij1MRFJ+RrqFvn9yMGQIPN4hvbbXwcvTZwDY
iIc6cjG57cinCrNcjNpd43w98xOY2ZYEL0CsdEzk05BLLwsLyFV6tduA7RwsOBO
+EUFMTqshkBWk6F1zPUsinZsDHXKUXP0v/46RF16FgNS9qsuhRAop560zPYvQXH6
JsgY802120mcBFr3NyBwdB3SwZ2g/F/kwNENRQIDAQABoIBAF5ezR+hu6+fxwhy
8pydo4/XlWAERgs3V7qlgDYCXIoVBMrF8dY56w1zGxi6TLJQT1ANh06DgcMxUbrQV
1BtkqgVkJETsv1g4mr58yaX2vodZZx3DnUScUFHRkgBY0gKAykY2JXkyCqKsd1oD
9hezwMTXE01IJF15+LISNPjnioJfu7MhOnhqUDie+sYE9qtEcMkiQxTn+BL3hUZ
vUsope1cXTTUSP+1NG1sWE63quL9uCGM/wzK8YJDv+Lvh/CmuJFx4gCjqdCD8KwW
fOWXaqJ0ebnUp1Bs1SfxQNH4OSSZjubSV+q7bP8kj/h1YMe1201HFYa/k9GJtbXr
djj13I0CgYEAzv3IQfmHiSSiPpz978aZHbqCY5a4BUXr1gTCIr2BnAvgMI1seo6
SIif4ZFr4HXKykvX2gCdHHB9Nb6WYG0sm43rtsnL5hI5QKPrebns3vvY4Aq5DS
GgQfzkKib/IcqjkEzeB2qB8g6RwdFHkWk/2fnnoJoOsFTOowCHPyU8CgYEaqIIV
alMdKEdy0nQPtq50/v9NGS7sYrke9VFLoWVC11M+kd1NZBVBSJg60+sSmgJDn8mw
YTVCq1H4j1IQQG6n5Vns7d61BVw10wLV3jttaFR4QSIt4kQwqZ/AzcbISw07M72X1
69LvUKpW4EsNm/hRaArx+/XWK07W1detJ4pYsysCgYB8/+MkfGUuG9Jjz3nGh/9F
ZbGCJ2iFYENsLfwNNMFgmJzz6juHE3wKZmsc3UXpZ5rDQJvsibxXRVNH07tswoG
DHW5BCRWd94n1yoMr+nLz1AXmJ6s/Pr0PRNcAv+HbQVFTjsEtUy7Ybt+nQqcmiu0
pybLQCiB7bhIAxAb+KK7nwKBgFSw8K8Ur+3HaDGTh6VZrtf1SzvAHXQIn//gLo03
reUPKdLJeCcssIw47srT2ZQeLNpSwMaWVWGGbNAKN5fbjMpIFIMwlyOYwDwZHIp
5maop93y/bQ6mjMRFpIVuP99WE05JuFqy6IRBJWot+4GQ2UqqM11rxo2FoC6e52
ea5/AoGAGE+rY0dk0an8YwdPV/x4zi3VVfRvlNdVjMPBgGeSLxC8CI1pj8YieX
AMHmP1pE75IOXrB1d1H0zd81hTwYDI1MYOJDDXSL0mq+r1MauogKLrt5Xk79hQA
J2zGLKi3IhrV5rjJYjok2UY1DN5w+3N8W020q3sDsBWjQXmYqWo=
-----END RSA PRIVATE KEY-----
```

3. Go to the Bastion-server which we have connected and create a file named **web-serverkey.pem**. Below is the command :

- **vi web-serverkey.pem**

```
Last login: Sun Nov  8 12:46:51 2020 from ec2-18-206-107-24.compute-1.amazonaws.com
[ec2-user@ip-10-0-0-118 ~]$ sudo su
[ec2-user@ip-10-0-0-118 ec2-user]# vi web-serverkey.pem
```

i-0ae2e1a6fce96ddbf (Bastion-Server)

Public IPs: 3.80.113.96 Private IPs: 10.0.0.118

- press **i** & paste the content that you have copied from **MyKey.pem** here.
- Press **Esc key** & type :**wq** & press enter.

```

1BtkcqgVkJETsv1g4mr58yaX2vodZZx3DnUScUFHRkgBY0gKAykY2JXkyCqKSd1oD/
9hezwMTXE01IJF15+LI9NPjnj0jFu7Mh0nhqUDie+sYE9qtEcMKiQxTn+BL3hUZ
vUsope1cXVTUSP+lNGlSwE63quL9uCGM/wzk8YJDv+Lvh/CmuJFx4gCjqdCD8KwW
f0WXaqj0ebnUp1Bs1SfxQNH40SSZJuB5V+q7bP8Kj/h1YMeI20lHFYa/k9GJTbXr
djjl3IOcgYEAzv3I0fmHiSSiPpz978aZHbqCY5a4BUxr1gTCIr2BnAvgMI10seo6
SIif4uZFr04HXYkvX2gCdHHB9Nbn6WYG0sm43rtsnL5hI50KPrenebs3wvY4Aq5DS
GgQfzkkib/IcQjkEZB2qBBg6RWdFHkWk/2fnm0jo0sFT0owCHPyU8CgYEAgIIIV
alMdKEdy0nQPtg50/v9NGS7sYrke9VFLowVCl1M+kdlNZBVbsJg60+sSmgJDn8mw
YTVCQ1H4j1IQQG6n5Vns7d61BVwi0wLV3jtAfR4QSIt4kQwqZ/AzcbI5w07M72Xl
69LvUKpW4EsNm/hRaArx+/XWK07W1detJ4pYsysCgYB8/+MkfGUuG9Jiz3nGh/9F
ZbGCJk2ifYElsfwNNMFgmJzz6juHE3wKZmsc3UXpZ5rDQjvsibxXRVNH07TtwoG
DHW5BCRWd94nlyoMr+nLz1AXmJ6s/Pr0PRNcAv+HbQVFtjsETuY7Ybt+nQcmiu0
pybLQCib7bhI4XaB+Kk7nwKBgFSW8K8uR+3HaDGTh6vZ9tf1SzvAHXQ1n//gLoo3
reUPKdLJeCcss1w47srTZ8QeLNpSwMaWWGGbNAKN5fbjMpIFIMwW0YwDwZHIp
5maop93y/bQ6mjMRFpIVuP99WE05JuFq0y6IRBJW0t+4GQ2UqQMl1rxo2FoC6e52
ea5/AoGAGE+rY0dk0an8YwdPV/x4zi3VVfRvlNDvjMPBggEsLxC8C1lpj8YieX
AMHmP1pE7SI0XrBld1H10zd81hTwYDILMY0JDDXSL0mq+r1MauogKLrt5Xk79hQA
J2zGLKi3IhrV5rjJYjok2UYldN5w+3N8W020q3DsBWjQXmYqWo=
-----END RSA PRIVATE KEY-----

```

:wq

i-0674ba6fb77003cd5 (Bastion-Server)

Public IPs: 107.23.177.149 Private IPs: 10.0.0.128

- A file with **web-serverkey.pem** will be created
- Now we will have to change the permission of the pem file. Below is the command :
 - **chmod 400 web-serverkey.pem**
 - This command will give read-only permission to file.
 - Now we can connect/login to the web servers using this file with the below command:
 - **ssh -i web-serverkey.pem ec2-user@10.0.1.92**

j-0674ba6fb77003cd5 (Bastion-Server)

Public IPs: 107.23.177.149 Private IPs: 10.0.0.128

- You can see we are connected to our first Webserver(Web-server-1) with **Private ip : 10.0.1.92**

6. Now we will update Web Server & install Apache Server on it.

```
[root@ip-10-0-0-128 ec2-user]# ssh -i web-serverkey.pem ec2-user@10.0.1.92
Last login: Sun Nov  8 17:09:48 2020 from 10.0.0.128
```

Amazon Linux 2 AMI

```
https://aws.amazon.com/amazon-linux-2/
26 package(s) needed for security, out of 40 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-1-92 ~]$ sudo su
[root@ip-10-0-1-92 ec2-user]# yum update -y
```

- On Web-server-1 type following commands :
 - `sudo su` – to become root user
 - `yum update -y` – to install latest packages
 - `yum install httpd -y` – to install Apache
 - `systemctl start httpd` – to start Apache
 - `systemctl enable httpd` – to enable Apache

```

Installed:
httpd.x86_64 0:2.4.46-1.amzn2

Dependency Installed:
apr.x86_64 0:1.6.3-5.amzn2.0.2
apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2
httpd-filesystem.noarch 0:2.4.46-1.amzn2
mailcap.noarch 0:2.1.41-2.amzn2

Complete!
[root@ip-10-0-1-92 ec2-user]# systemctl start httpd
[root@ip-10-0-1-92 ec2-user]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-10-0-1-92 ec2-user]#

```

- Now Apache is installed on your Web-server-1 and it's running.

7. Now let us create a homepage with name **index.html** on your **Web-Server1**, before that change the directory & then a create homepage.

- cd /var/www/html** – Will move to default folder of web server
- echo “REQUEST HANDLING BY SERVER 1”> index.html**

```

[root@ip-10-0-1-92 ec2-user]# echo "REQUEST HANDLING BY SERVER1">>index.html
[root@ip-10-0-1-92 ec2-user]# exit
exit
[ec2-user@ip-10-0-1-92 ~]$ exit
Logout
Connection to 10.0.1.92 closed.
[root@ip-10-0-0-128 ec2-user]#

```

i-0674ba6fb77003cd5 (Bastion-Server)

Public IPs: 107.23.177.149 Private IPs: 10.0.0.128

- Type “**exit**” to come out of the root user
 - Again type “**exit**” to come out of the instance back to Bastion-Server
8. Repeat the **Task 2** steps from **5 to 7** for second web server (**Web-server-2**) **with its respective private IP** and also make sure to change the content of **index.html** for Web-server-2 to “ **REQUEST HANDLING BY SERVER 2**”
9. After the above step we are ready with our web servers having Apache Server running and having home pages.

STEP 8 Creating additional Public Subnets, each subnet in different availability zone and in same VPC

1. We have already created one Public subnet at the start with following details :

- Name : MyPublicSubnet
- VPC : MyVPC
- Availability Zone : us-east-1a
- IPv4 CIDR : 10.0.0.0/24

2. Now we will be creating two more Public subnet in two different availability zones.

3. Navigate to **VPC ->Subnets**

4. Click on **Create subnet**

The screenshot shows the 'Create subnet' form in the AWS Management Console. The 'Name tag' field contains 'MyPublicSubnet1'. The 'VPC*' dropdown shows 'vpc-05fc7fcfd94476a6' with 'MyVPC' next to it. The 'Availability Zone' dropdown is set to 'us-east-1b' and is highlighted with a red box. The 'VPC CIDRs' table shows a single entry: 'CIDR' 10.0.0.0/16 and 'Status' associated. The 'IPv4 CIDR block*' field contains '10.0.4.0/24'. At the bottom, there is a note '* Required' and a 'Create' button.

- Name : MyPublicSubnet1
- VPC : Select MyVPC
- Availability Zone : us-east-1b
- IPv4 CIDR block : 10.0.4.0/24

5. Click on **Create**. Your subnet will be created.

6. Now similarly create one more subnet with following details :

- Name : MyPublicSubnet
- VPC : MyVPC
- Availability Zone : us-east-1e
- IPv4 CIDR block : 10.0.6.0/24

- After creating the subnets we can see three public subnets created in different availability zones and in same VPC(**MyVPC**)

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with options like New VPC Experience, VIRTUAL PRIVATE CLOUD, Subnets, Route Tables, Internet Gateways, and Egress Only Internet Gateways. The Subnets section is selected. In the main area, there's a table titled 'Create subnet' with columns: Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, IPv6 CIDR, Availability Zone, and Av. Three subnets are listed: MyPublicSubnet2, MyPublicSubnet1, and MyPublicSubnet, all in the 'available' state. They are part of the 'MyVPC' VPC. Their IPv4 CIDRs are 10.0.6.0/24, 10.0.4.0/24, and 10.0.0.0/24 respectively. They are located in the 'us-east-1e', 'us-east-1b', and 'us-east-1a' availability zones. A red box highlights these three subnets.

- Next step is to create an **Application Load Balancer**.

Note: *The purpose of creating more subnets in different availability zone is to protect our application from the any location failures.*

STEP 9 Creating an Application Load Balancer with multiple subnets

Now we will be creating new load balancer for our webservers.

- In the **EC2 console**, scroll down to **Load Balancing** and select **Load Balancer** under it.

The screenshot shows the AWS EC2 Load Balancing console. On the left, there's a sidebar with Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces) and Load Balancing (Load Balancers, Target Groups). The Load Balancers option is selected and highlighted with a red box. In the main area, there's a table with columns: Name, DNS name, State, VPC ID, Availability Zones, and Type. A message at the bottom says 'You do not have any load balancers in this region.' A red box highlights the 'Create Load Balancer' button.

- Click on **Create Load Balancer**.

- In **Select load balancer type**,

Select load balancer type

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers (new), and Classic Load Balancers. Choose the load balancer type that meets your needs. Learn more about which load balancer is right for you.

Application Load Balancer

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Network Load Balancer

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your application. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Classic Load Balancer

PREVIOUS GENERATION for HTTP, HTTPS, and TCP

Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

[Learn more >](#)

[Create](#)

[Create](#)

[Create](#)

- Under **Application Load Balancer,(HTTP/HTTPS)** Click on **Create**

Note : We are selecting *HTTP/HTTPS* as we will be testing High availability of the web application.

4. In Configure Load Balancer, enter the following details

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name	ApplicationLB
Scheme	<input checked="" type="radio"/> Internet-facing <input type="radio"/> Internal
IP address type	IPv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80

Add listener

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC	vpc-05fc7fcfd94476a6 (10.0.0.0/16) MyVPC
Availability Zones	<input checked="" type="checkbox"/> us-east-1a subnet-053f85d7bb6fb903f (MyPublicSubnet) <input checked="" type="checkbox"/> us-east-1b subnet-03a49a1135f507fb3 (MyPublicSubnet1) <input checked="" type="checkbox"/> us-east-1e subnet-0c32629efc1861230 (MyPublicSubnet2)

Add-on services

[Cancel](#) [Next: Configure Security Settings](#)

- Name : **ApplicationLB**
- Scheme : Select **internet-facing**

- VPC : Select **MyVPC**
- IP address type : **ipv4**
- Listeners : **Default (HTTP : 80)**
- Availability Zones : Select 3 Availability zones and in subnet select **MyPublicSubnet(us-east-1a)** , **MyPublicSubnet1(us-east-1b)**, **MyPublicSubnet2(us-east-1e)**
- Click on **Next: Configure Security Settings**

Note: We must specify the availability zones in which your load balancer needs to be enabled, making it routing the traffic only to the targets launched in those availability zones. You must include subnets from a minimum of two Availability zones to make the Load balancer **Highly Available**.

5. **Configure Security Settings** : Will see a message “Improve your load balancer’s security. Your load balancer is not using any secure listener.” Leave it as it is.

The screenshot shows the AWS Load Balancer configuration interface. At the top, there's a navigation bar with the AWS logo, 'Services ▾', and user information 'Kishore Shinde ▾ N. Virginia ▾ Support ▾'. Below the navigation bar, a progress bar indicates the current step: '1. Configure Load Balancer' (disabled), '2. Configure Security Settings' (selected), '3. Configure Security Groups' (disabled), '4. Configure Routing' (disabled), '5. Register Targets' (disabled), and '6. Review' (disabled). The main content area is titled 'Step 2: Configure Security Settings'. A red box highlights a warning message: '⚠ Improve your load balancer's security. Your load balancer is not using any secure listener.' followed by a detailed description: 'If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under Basic Configuration section. You can also continue with current settings.' At the bottom right, there are buttons for 'Cancel', 'Previous', and 'Next: Configure Security Groups' (which is highlighted with a red border).

Click on **Next: Configure Security Groups**

6. Configure Security Groups :

- Assign a security group: select, **Select an existing security group**

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group:

- Create a new security group
- Select an existing security group

Security Group ID	Name	Description	Actions
sg-0810f6ef25a00c50b	Bastion-SG	Security Group for Bastion-server	Copy to new
sg-07cbe01b110a0e0fa	default	default VPC security group	Copy to new
sg-052c8591f9a497064	launch-wizard-1	launch-wizard-1 created 2020-11-04T14:50:31.250+05:30	Copy to new
sg-0fb8f0971e166bb9c	launch-wizard-2	launch-wizard-2 created 2020-11-04T15:45:55.018+05:30	Copy to new
sg-0815f9dc3dd43e05f	LoadBalancer-SG	Security group for the Load balancer	Copy to new
sg-0e3b4f846479af30d	web-server-SG2	Security group for web servers	Copy to new
sg-0eabf5078f27f3e0f	WebserverSG1	Security Group for Web Servers	Copy to new

Cancel Previous Next: Configure Routing

- From the existing security group list, select **LoadBalancer-SG**
- Click on **Next: Configure Routing**

7. Configure Routing :

- Target group : **New target group** (selected by default)
- Name : **WebApp-TG**
- Target type : Select **Instance** (selected by default)
- Protocol : **HTTP**
- Port : **80**

Step 4: Configure Routing
Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. The target group you specify in this step will apply to all of the listeners configured on this load balancer; you can edit the listeners and add listeners after the load balancer is created.

Target group

Target group: New target group
Name: WebApp-TG
Target type: Instance
Protocol: HTTP
Port: 80
Protocol version:
 HTTP1: Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
 HTTP2: Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
 gRPC: Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

Protocol: HTTP
Path: /index.html

Advanced health check settings

Cancel Previous **Next: Register Targets**

- Health checks :
 - Protocol : select **HTTP**
 - Path : type **/index.html**

Note: The load balancer periodically sends pings, attempts connections, or sends requests to test the EC2 instances. These tests are called **health checks**.

In the Path we have mentioned **/index.html** it will create an **index.html** in the root directory of the Apache web servers (**/var/www/html**) to pass this health check.

- Click on **Next: Register Targets**

8. Register Targets

- Under the Instances select the two instance we created in private subnet (**Web-server-1 & Web-server-2**) & click on **Add to registered**.

Step 5: Register Targets

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

Registered targets

To deregister instances, select one or more registered instances and then click Remove.

Instance	Name	Port	State	Security groups	Zone
i-035695a80a3220444	Web-server-1	80	running	WebserverSG1	us-east-1a
i-07d3eca613a38ab3b	Web-server-2	80	running	WebserverSG1	us-east-1a

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered on port 80

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-035695a80a3220444...	Web-server-1	running	WebserverSG1	us-east-1a	subnet-0ec7936b7bd5f7340	10.0.1.0/24
i-07d3eca613a38ab3b...	Web-server-2	running	WebserverSG1	us-east-1a	subnet-0ec7936b7bd5f7340	10.0.1.0/24

Cancel Previous Next: Review

- You can see the instances will get added under **Registered targets**
- Click on **Next: Review**.

9. Review : Review all the settings

Step 6: Review

Please review the load balancer details before continuing

Load balancer

Name ApplicationLB
Scheme internet-facing
Listeners Port:80 - Protocol:HTTP
IP address type ipv4
VPC vpc-05fc7fcfd0447656 (MyVPC)
Subnets subnet-053fb5d70b6fb903f (MyPublicSubnet), subnet-03a49a1135f507fb3 (MyPublicSubnet1)

Step 6: Review

Routing

Target group New target group
Target group name WebApp-TG
Port 80
Target type instance
Protocol HTTP
Protocol version HTTP1
Health check protocol HTTP
Path /index.html
Health check port traffic port
Healthy threshold 5
Unhealthy threshold 2
Timeout 5
Interval 30
Success codes 200

Targets

Instances i-035695a80a3220444 (Web-server-1):80, i-07d3eca613a38ab3b (Web-server-2):80
--

Add-on services

AWS Global Accelerator Disabled

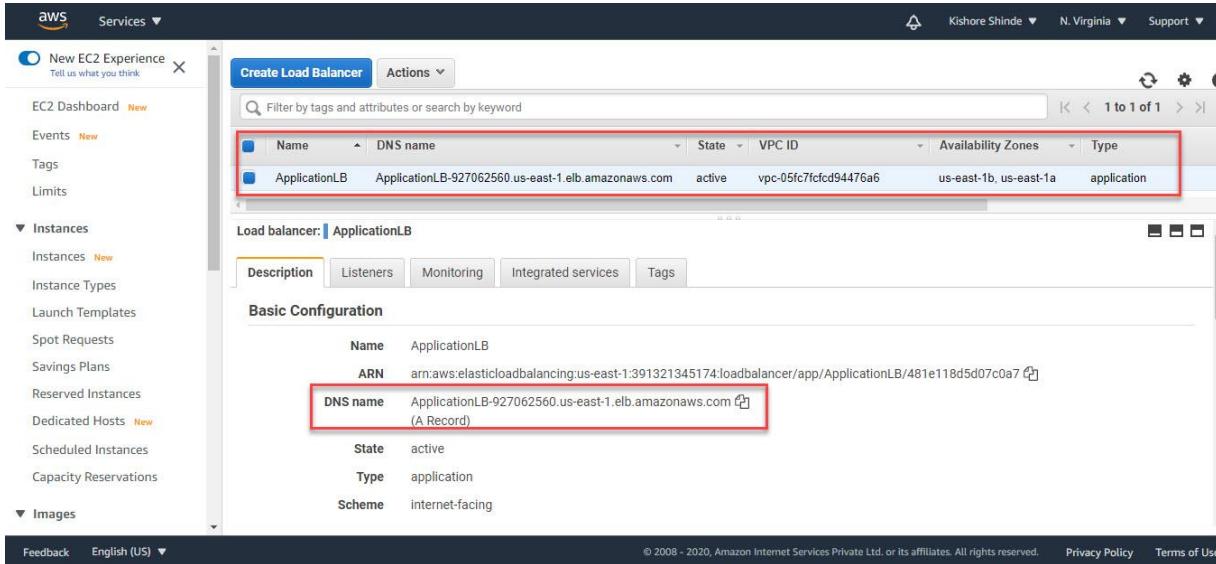
Create

- Click on **Create**

10. You can see the Load Balancer Creation Status : Successfully created load balancer . Click on Close.

11. You have successfully created the Application Load Balancer. Please wait for till the ALB status to become Active.

12. Load Balancer Details :



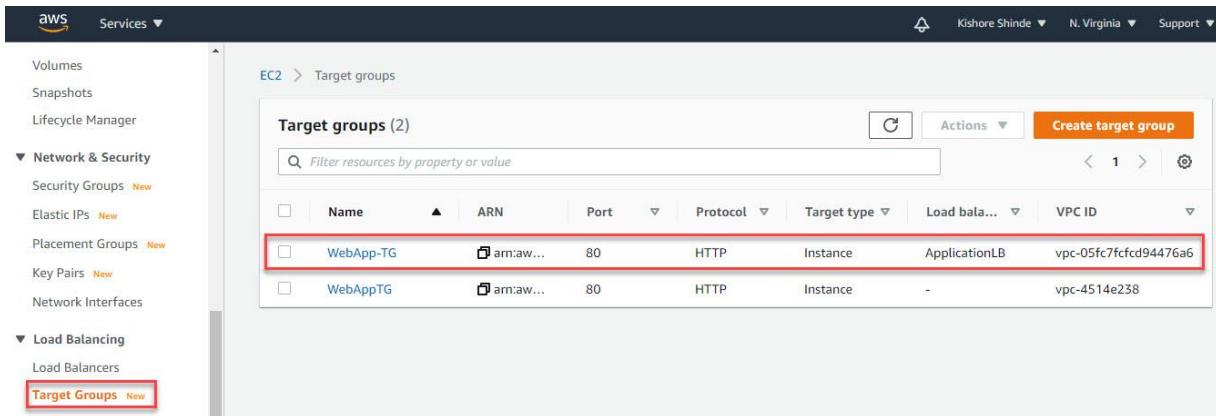
The screenshot shows the AWS EC2 Load Balancers page. On the left, there's a sidebar with options like EC2 Dashboard, Instances, and Images. The main area shows a table of load balancers with one row highlighted. The table has columns for Name, DNS name, State, VPC ID, Availability Zones, and Type. The highlighted row shows 'ApplicationLB' as the Name, 'ApplicationLB-927062560.us-east-1.elb.amazonaws.com' as the DNS name, 'active' as the state, 'vpc-05fc7fcfd94476a6' as the VPC ID, 'us-east-1b, us-east-1a' as the Availability Zones, and 'application' as the Type. Below the table, there's a detailed view for 'ApplicationLB' with tabs for Description, Listeners, Monitoring, Integrated services, and Tags. Under 'Basic Configuration', it shows the Name, ARN, DNS name (which is also highlighted with a red box), State, Type, and Scheme. The DNS name is listed as 'ApplicationLB-927062560.us-east-1.elb.amazonaws.com (A Record)'.

- Name : ApplicationLB
- DNS Name : ApplicationLB-927062560.us-east1.elb.amazonaws.com

Step 10 Checking the health of Load Balancer & Testing DNS

Task1 : Checking the health of Load Balancer

1. Go to EC2 Dashboard scroll down the left section and click on Target Groups under Load Balancing.



The screenshot shows the AWS EC2 Target groups page. On the left, there's a sidebar with options like Volumes, Snapshots, Lifecycle Manager, Network & Security, Load Balancing, and Target Groups. The 'Target Groups' option is highlighted with a red box. The main area shows a table of target groups with two rows. The table has columns for Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. The first row is highlighted with a red box and shows 'WebApp-TG' as the Name, 'arn:aws:lambda:us-east-1:927062560:target/WebApp-TG' as the ARN, '80' as the Port, 'HTTP' as the Protocol, 'Instance' as the Target type, 'ApplicationLB' as the Load balancer, and 'vpc-05fc7fcfd94476a6' as the VPC ID. The second row shows 'WebAppTG' as the Name, 'arn:aws:lambda:us-east-1:927062560:target/WebAppTG' as the ARN, '80' as the Port, 'HTTP' as the Protocol, 'Instance' as the Target type, '-' as the Load balancer, and 'vpc-4514e238' as the VPC ID.

2. Click on the target group – WebApp-TG

The screenshot shows the AWS Elastic Load Balancing console with the target group 'WebApp-TG' selected. The left sidebar shows navigation options like Volumes, Snapshots, Lifecycle Manager, Network & Security, Load Balancing, and Auto Scaling. The main area displays the 'Basic configuration' tab, which includes fields for Target type (instance), Protocol (HTTP : 80), VPC (vpc-05fc7fcfcd94476a6), and Load balancer (ApplicationLB). The 'Health check settings' tab is also visible, showing protocol (HTTP), port (traffic-port), healthy threshold (5 consecutive successes), unhealthy threshold (2 consecutive failures), timeout (5 seconds), and success codes (200). The 'Attributes' tab shows settings for Stickiness (Disabled) and Deregistration delay (300 seconds).

- Under the **Health check settings** you can see the **Healthy** and **Unhealthy thresholds**.

3. Now let us check the status of the attached targets. Click on **Targets** tab.

The screenshot shows the AWS EC2 Target Groups console. On the left, there's a navigation sidebar with options like EC2 Dashboard, Events, Tags, Limits, Instances (with sub-options like Instances, Instance Types, Launch Templates, etc.), and Images. The main area shows the 'Target groups' page for 'WebApp-TG'. The 'Basic configuration' section is highlighted with a red box, showing details like Target type: instance, Protocol: Port HTTP: 80, VPC: vpc-05fc7fcfd94476a6, and Load balancer: ApplicationLB. Below it is a table titled 'Registered targets (2)' with two entries: 'i-035695a80a3220444' (Web-server-1) and 'i-07d3eca613a38ab3b' (Web-server-2), both marked as healthy.

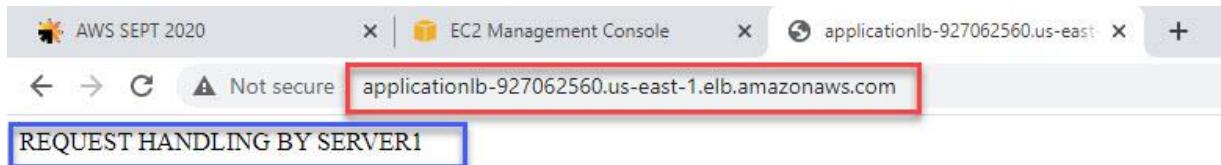
- Under the **Registered targets** you can see both the instances are **healthy**.

Task 2 : Testing the DNS

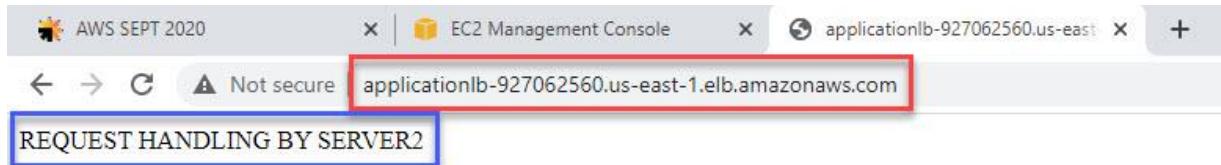
- Click on Load Balancers under Load Balancing.

The screenshot shows the AWS Load Balancers console. The left sidebar has sections for Elastic Block Store, Network & Security, Load Balancing (with 'Load Balancers' selected), and Auto Scaling. The main area shows a list of load balancers, with 'ApplicationLB' selected. The 'Basic Configuration' section at the bottom shows the Name: ApplicationLB, ARN: arm:aws:elasticloadbalancing:us-east-1:391321345174:loadbalancer/app/ApplicationLB/481e118d5d07c0a7, and DNS name: ApplicationLB-927062560.us-east-1.elb.amazonaws.com (A Record). The 'DNS name' field is highlighted with a red box.

- Select the Load Balancer – ApplicationLB.
- Copy the DNS name : **ApplicationLB-927062560.us-east-1.elb.amazonaws.com** & paste it into the browser.
- You should be able to see the REQUEST HANDLING BY SERVER 1.



5. Refresh the browser a couple of times to see the requests being served from both servers.



Once you see both the outputs **REQUEST HANDLING BY SERVER 1** & **REQUEST HANDLING BY SERVER 2** our DNS is tested and it also shows that the load is shared between the two web servers via Application Load Balancer.

Step 11 : Testing High Availability

1. To test the High Availability, we will have to make one of the instances unhealthy and test whether we get response from the other server.
2. If your instance is unhealthy then its status would be one of the following:
 - Stopping
 - Stopped
 - Terminating
 - Terminated
3. Navigate to the EC2 Dashboard and select **Web-server-1**. Click on **Instance State** & click on **Stop instance**.

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, the 'Instances' link is highlighted with a red box. The main table lists three instances:

Name	Instance ID	Instance state
<input checked="" type="checkbox"/> Web-server-1	i-035695a80a3220444	Running
<input type="checkbox"/> Web-server-2	i-07d3eca613a38ab3b	Running
<input type="checkbox"/> Bastion-Server	i-0674ba6fb77003cd5	Running

The 'Actions' menu at the top right has a 'Stop instance' option, which is also highlighted with a red box.

- Click on Stop for confirmation.



2. You can see the Web-server-1 instance state is Stopping.

The screenshot shows the AWS EC2 Instances page again. The 'Instances' link in the sidebar is highlighted with a red box. The main table now shows the 'Web-server-1' instance in the 'Stopping' state:

Name	Instance ID	Instance state	Instance type	Status check	Alarm Status
<input checked="" type="checkbox"/> Web-server-1	i-035695a80a3220444	Stopping	t2.micro	2/2 checks ...	No alarms
<input type="checkbox"/> Web-server-2	i-07d3eca613a38ab3b	Running	t2.micro	2/2 checks ...	No alarms
<input type="checkbox"/> Bastion-Server	i-0674ba6fb77003cd5	Running	t2.micro	2/2 checks ...	No alarms

3. Now we will check the status of **Targets**. Navigate to the Load Balancing on the left panel, and select Target Groups under it.

The screenshot shows the AWS EC2 Target Groups console. On the left sidebar, under the 'Instances' section, 'Target Groups' is selected. In the main content area, the 'WebApp-TG' target group is displayed. The 'Targets' tab is active. The 'Registered targets' table shows two entries:

Instance ID	Name	Port	Zone	Status	Status details
i-035695a80a3220444	Web-server-1	80	us-east-1a	unused	Target is in the stopped state
i-07d3eca613a38ab3b	Web-server-2	80	us-east-1a	healthy	

4. Click on **Targets** tab you can see **Web-server-1** status is changed from **healthy** to **unused** and in **Status details** it shows “**Target is in stopped state**”.

5. Navigate to Load Balancers. Select ApplicationLB, under Description tab below. Copy the DNS name.

The screenshot shows the AWS Load Balancers console. Under the 'Load Balancing' section, 'Load Balancers' is selected. A table lists one load balancer:

Name	DNS name	State	VPC ID	Availability Zones	Type
ApplicationLB	ApplicationLB-927062560.us...	active	vpc-05fc7fcfd94476a6	us-east-1b, us-east-1a...	application

In the 'Description' tab, the 'DNS name' field is highlighted with a red box and contains the value 'ApplicationLB-927062560.us-east-1.elb.amazonaws.com'. A 'Copied' button is visible next to the field.

Paste it into your browser. You should see the response "**REQUEST HANDLING BY SERVER 2**". This is the response from Web-server-2.

A screenshot of a web browser window. The address bar shows the URL `applicationlb-927062560.us-east-1.elb.amazonaws.com`. The page content is a single line of text: "REQUEST HANDLING BY SERVER2".

6. Even if you refresh a few times, you will continue to see the response only from Web-server-2
7. Now let us stop the **Web-server-2** and check.
8. Before that let us restart **Web-server-1**

A screenshot of the AWS EC2 Instances page. The left sidebar shows navigation options like EC2 Dashboard, Events, Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, and Images. The main content area shows a table of instances:

Name	Instance ID	Instance state
Web-server-1	i-035695a80a3220444	Stopped
Web-server-2	i-07d3ca613a38ab3b	Running
Bastion-Server	i-0674ba6fb77003cd5	Running

The "Actions" dropdown menu for the stopped instance includes "Stop instance", "Start instance" (which is highlighted with a red box), "Reboot instance", and "Hibernate instance". Below the table, there is a detailed view for the selected instance (Web-server-1):

Instance: i-035695a80a3220444 (Web-server-1)

Details Security Networking Storage Status Checks Monitoring Tags

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-035695a80a3220444 (Web-server-1)	-	10.0.1.92
Instance state	Public IPv4 DNS	Private IPv4 DNS
Stopped	-	-

9. Once it is started let us stop **Web-server-2**.

Instances (1/3) Info

Name	Instance ID	Instance state	Actions
Web-server-1	i-035695a80a3220444	Running	Stop instance Reboot instance Hibernate instance
<input checked="" type="checkbox"/> Web-server-2	i-07d3eca613a38ab3b	Running	Start instance Terminate instance
Bastion-Server	i-0674ba6fb77003cd5	Running	Stop instance Reboot instance Hibernate instance

Instance: i-07d3eca613a38ab3b (Web-server-2)

Details Security Networking Storage Status Checks Monitoring Tags

Instance summary Info

Instance ID <input type="checkbox"/> i-07d3eca613a38ab3b (Web-server-2)	Public IPv4 address -	Private IPv4 addresses <input type="checkbox"/> 10.0.1.194
Instance state Running	Public IPv4 DNS -	Private IPv4 DNS <input type="checkbox"/> ip-10-0-1-194.ec2.internal

10. Now let us will check the status of Targets. Navigate to the Load Balancing on the left panel, and select Target Groups under it. Select the target group WebApp-TG and go to Targets Tab.

EC2 > Target groups > WebApp-TG

WebApp-TG

arn:aws:elasticloadbalancing:us-east-1:391321345174:targetgroup/WebApp-TG/65aad98f5e084e15

Basic configuration

Target type instance	Protocol : Port HTTP : 80	VPC vpc-05fc7fcfd94476a6	Load balancer ApplicationLB
	Protocol version HTTP1		

Targets

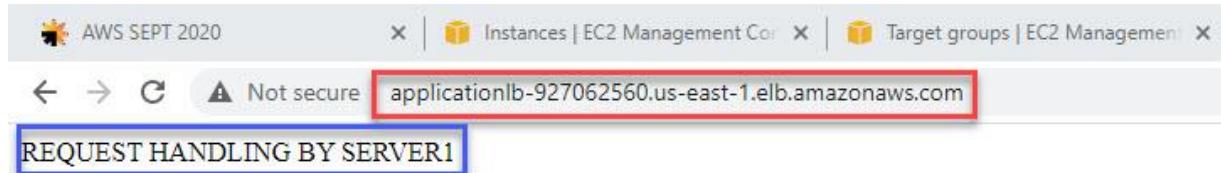
Group details Targets Monitoring Tags

Registered targets (2)

Instance ID	Name	Port	Zone	Status	Status details
i-035695a80a3220444	Web-server-1	80	us-east-1a	healthy	
i-07d3eca613a38ab3b	Web-server-2	80	us-east-1a	unused	Target is in the stopped state

- You can see the status of Web-server-2 which we stopped is unused and Web-server-1 is healthy

11. Now when we access the DNS we will get the response from Web-server-1



12. Now when we restart the Web-server-2 and check the health of instances or web servers it will show both **healthy**.

A screenshot of the AWS EC2 Target Groups page. The target group is named "WebApp-TG". The "Basic configuration" section shows the target type as "instance", protocol as "HTTP : 80", and VPC as "vpc-05fc7fcfd94476a6". The "Registered targets" section lists two instances: "Web-server-1" and "Web-server-2", both of which are marked as "healthy".

Instance ID	Name	Port	Zone	Status
i-035695a80a3220444	Web-server-1	80	us-east-1a	healthy
i-07d3eca613a38ab3b	Web-server-2	80	us-east-1a	healthy

13. Now let us **Terminate** one web server or instance and check.

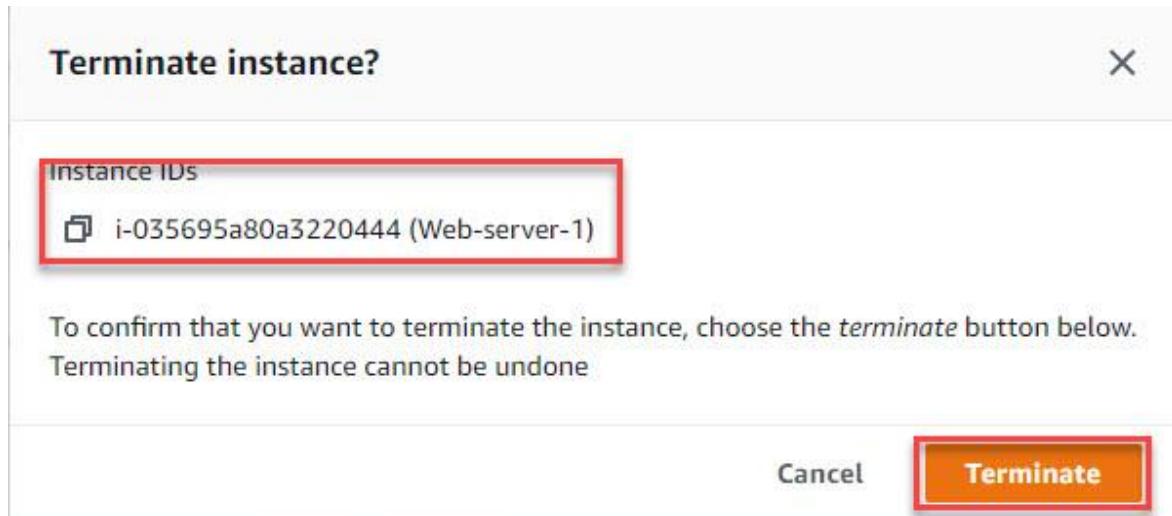
14. Go to **EC2 Dashboard** and select **Web-server-1** instance and click on **Terminate**

The screenshot shows the AWS EC2 Instances page. There are three instances listed:

Name	Instance ID	Instance state
Web-server-1	i-035695a80a3220444	Running
Web-server-2	i-07d3eca613a38ab3b	Running
Bastion-Server	i-0674ba6fb77003cd5	Running

The 'Actions' dropdown menu for the selected instance (Web-server-1) includes options: Stop instance, Reboot instance, Hibernate instance, and Terminate instance. The 'Terminate instance' option is highlighted and surrounded by a red box.

15. Just confirm the Web-server-1 termination.



- Click on Terminate.

The screenshot shows the AWS EC2 Instances page. A success message at the top says "Successfully terminated i-035695a80a3220444". The main table lists three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm Status
Web-server-1	i-035695a80a3220444	Shutting-down	t2.micro	-	No alarms
Web-server-2	i-07d3eca613a38ab3b	Running	t2.micro	2/2 checks ...	No alarms
Bastion-Server	i-0674ba6fb77003cd5	Running	t2.micro	2/2 checks ...	No alarms

Details for instance i-035695a80a3220444 (Web-server-1) are shown, indicating it is shutting down.

16. Once the **Web-server-1** is terminated let us once again check the health of the instances.

The screenshot shows the AWS Target Groups page for the "WebApp-TG" target group. The "Targets" tab is selected. The "Registered targets" section shows one target:

Instance ID	Name	Port	Zone	Status
i-07d3eca613a38ab3b	Web-server-2	80	us-east-1a	healthy

- You can see the **Web-server-1** target is deleted and **Web-server-2** is present and **healthy**.

17. Now let us check the DNS again. Copy the DNS name and paste in browser.

The screenshot shows a browser window with the URL "applicationlb-927062560.us-east-1.elb.amazonaws.com". The response from the server is "REQUEST HANDLING BY SERVER2".

- We can see we are getting the reply from **Web-server-2**.

Now we have tested the **High Availability** after **stopping & terminating** the instances.

Completion and Conclusion

1. We have launched a **Bastion server** and **two web-servers**. We were able to SSH into the servers via Bastion Server successfully.
2. We launched an Application Load Balancer and associated our web servers with the load balancer.
3. We tested the load sharing between web servers.
4. We successfully tested the high availability of the web application by making one of the web servers unhealthy.

xxx---AWS Assessment Project 1 Ends Here---xxx