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

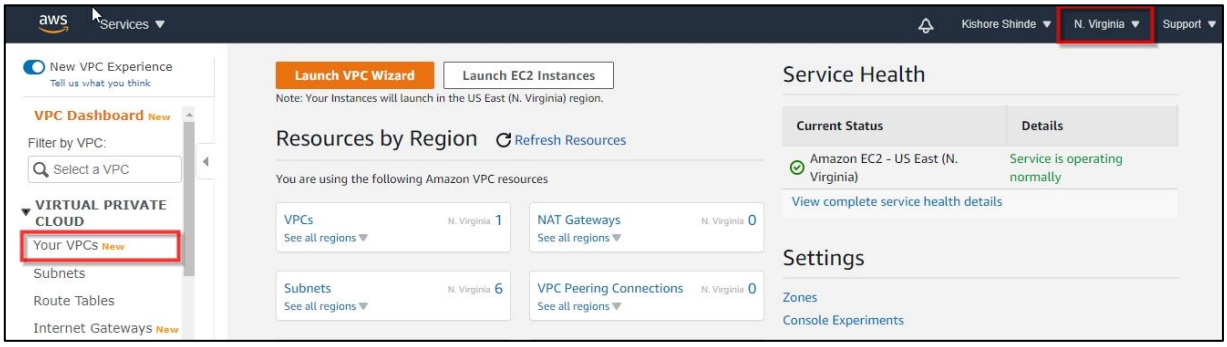
Step 1	Create a VPC with a Public & Private Subnet
Step 2	Create Internet gateway and associate with the Public Subnet
Step 3	Creating NAT Gateway & associate with Private Subnet
Step 4	Launch Bastion Host in Public Subnet
Step 5	Creating a Security Group for the Load Balancer
Step 6	Launch two Web Servers 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

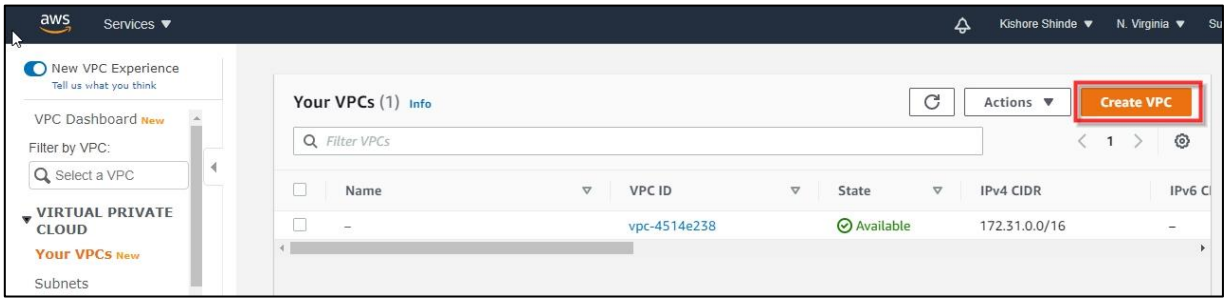
- Log into **AWS Management Console**
- Click on **Services** and Search for VPC
- Select **VPC** & do following tasks

Task 1 : Create VPC

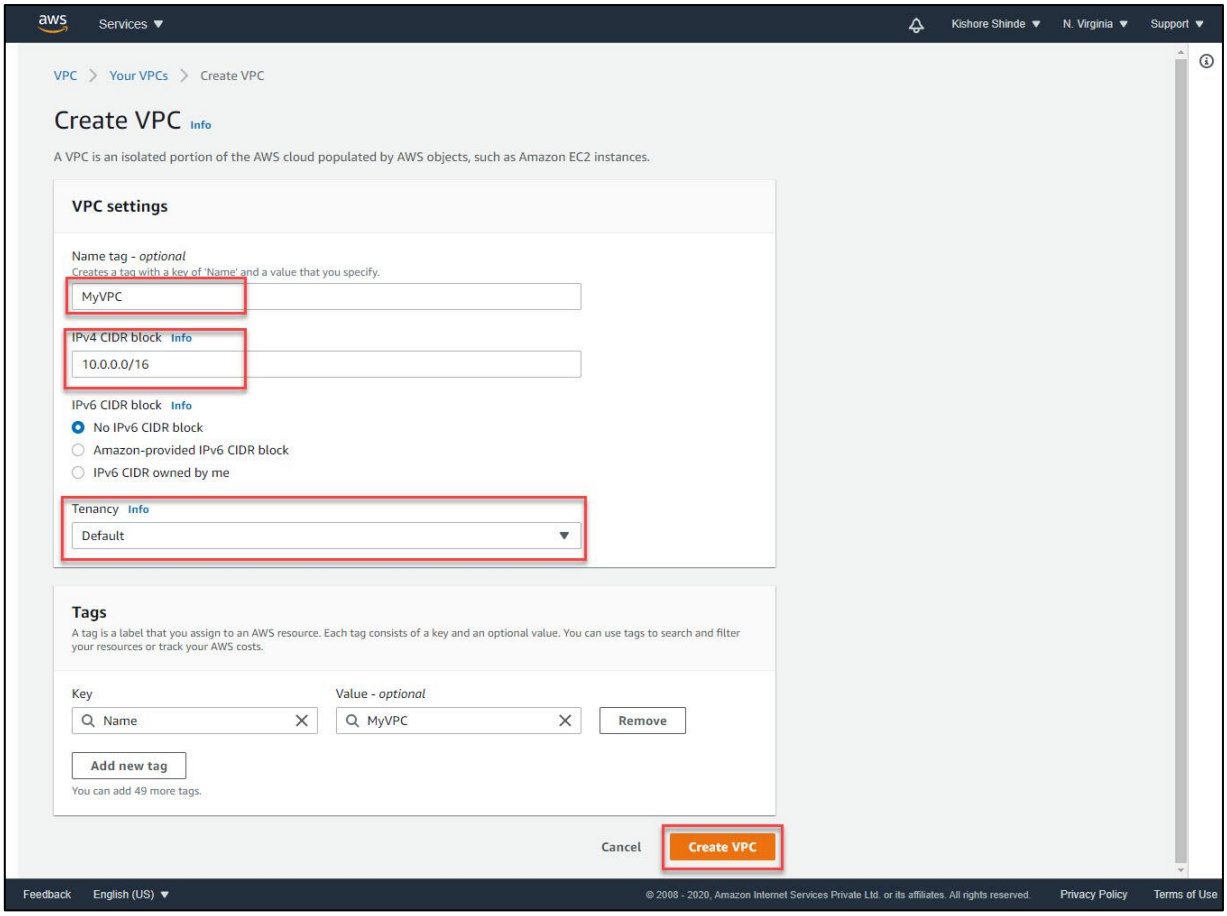
- Make sure you have selected **US East (N. Virginia) us-east-1** region.



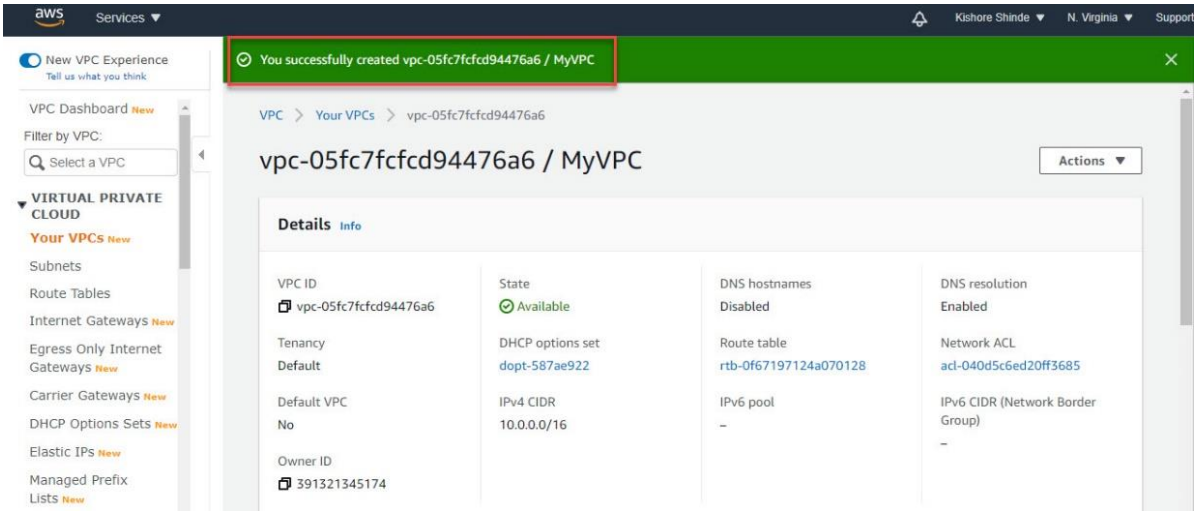
- Click on **Your VPCs**



- Click on **Create VPC**



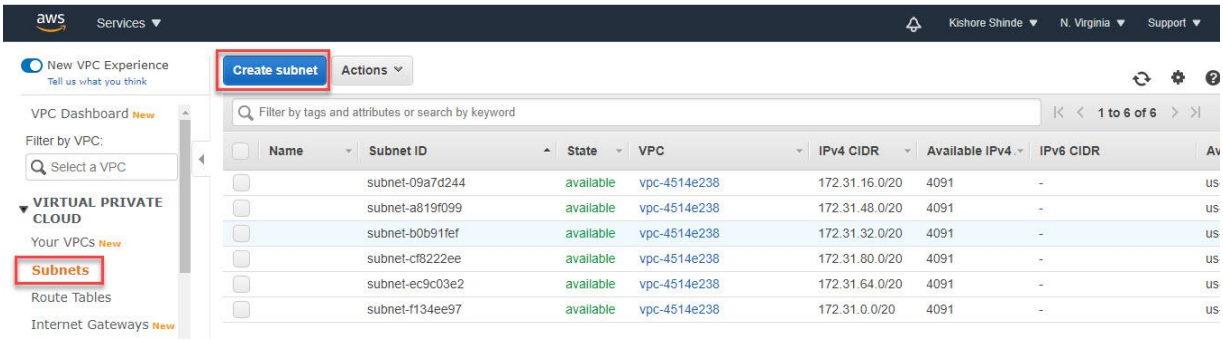
- Name tag : MyVPC
- IPv4 CIDR block : 10.0.0.0/16
- IPv6 CIDR block : No IPv6 CIDR block
- Tenancy : Default
- Click on Create VPC



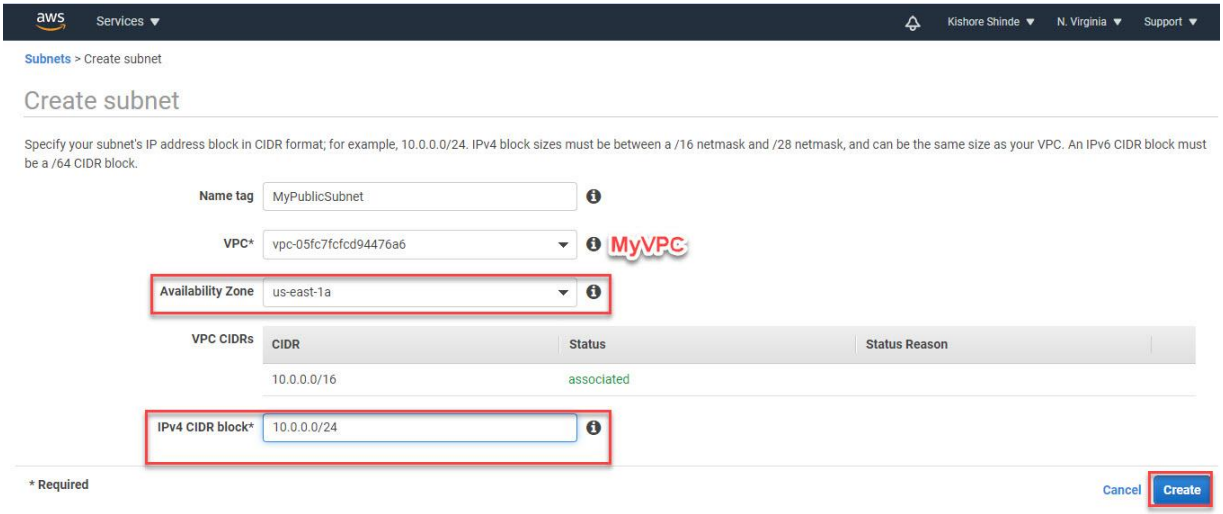
The VPC is now created.

Task 2 : Create Public & Private Subnet

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



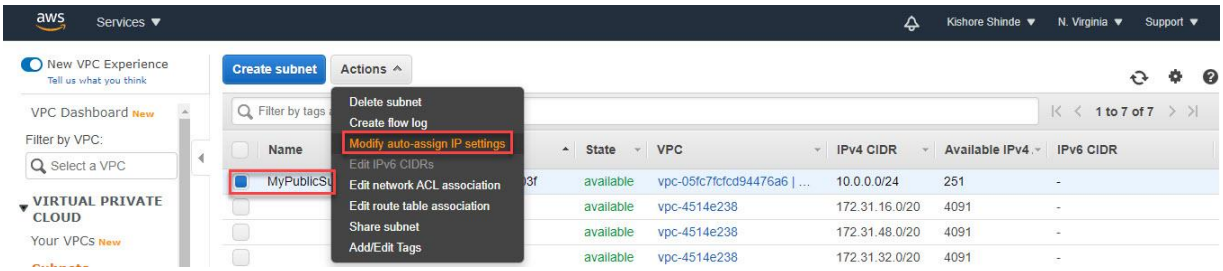
2. Creating Public Subnet, Click on **Create subnet**



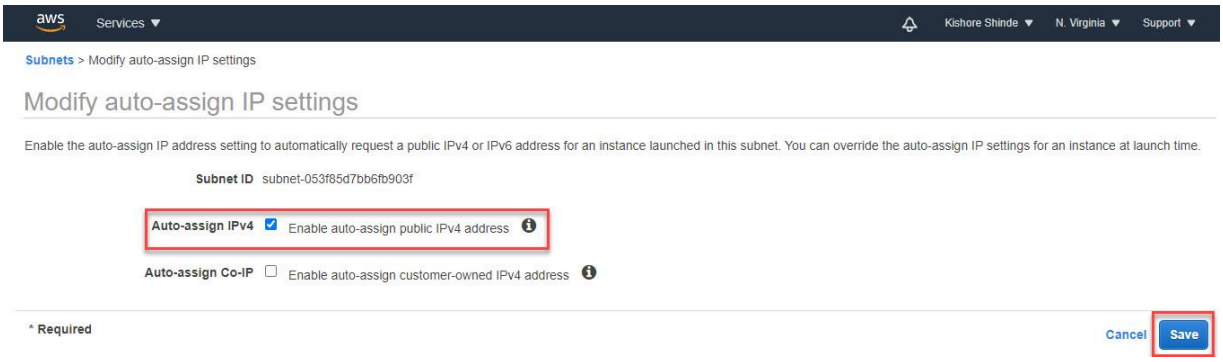
- Name tag : **MyPublicSubnet**
- VPC : Select **MyVPC**
- Availability Zone : **us-east-1a**
- IPv4 CIDR block : **10.0.0.0/24**
- Click on **Create**

You will get a message Subnet was created click on **Close**

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



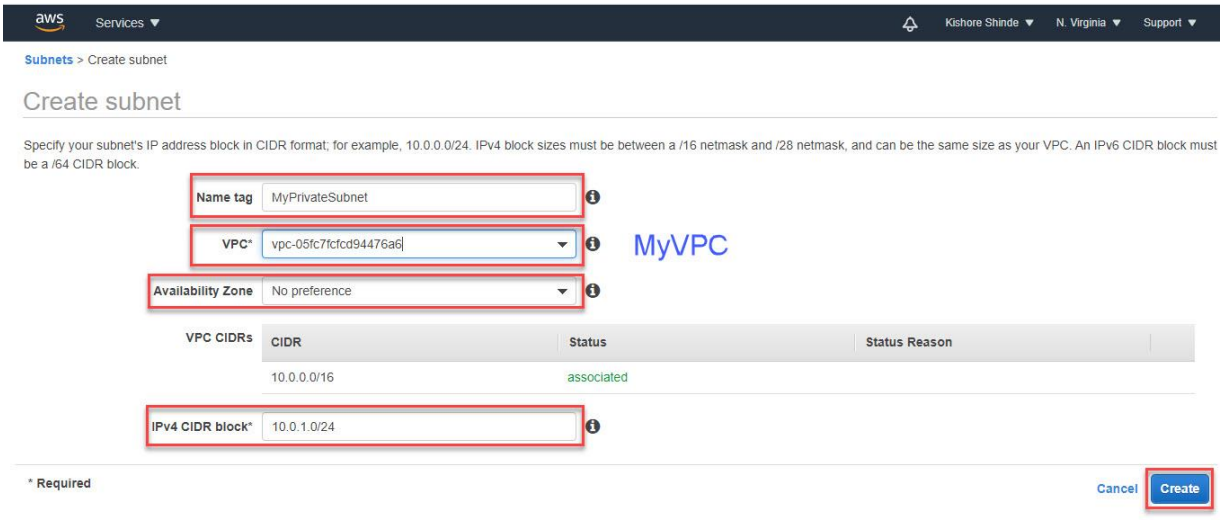
- Select **MyPublicSubnet**, Click on **Actions**
- Click on **Modify auto-assign IP settings**



- 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**

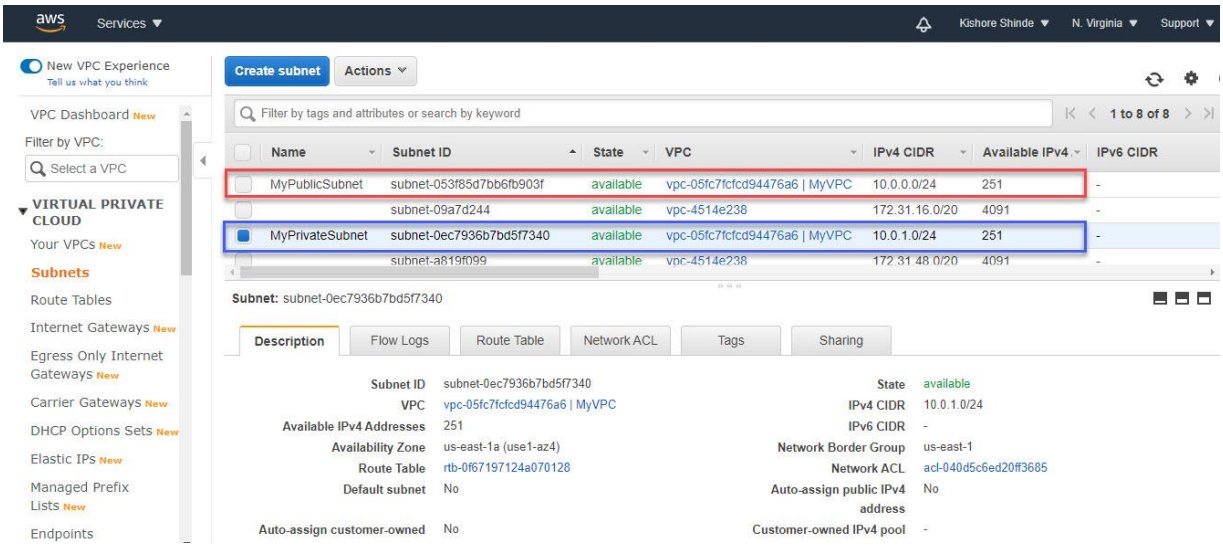


- Name tag: **MyPrivateSubnet**

- 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.**

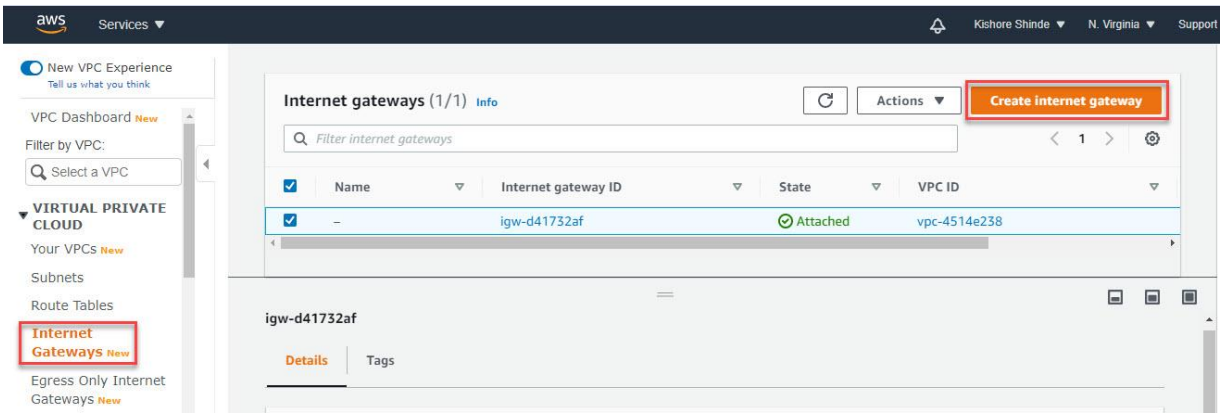
You will see two subnets created (Public & Private).



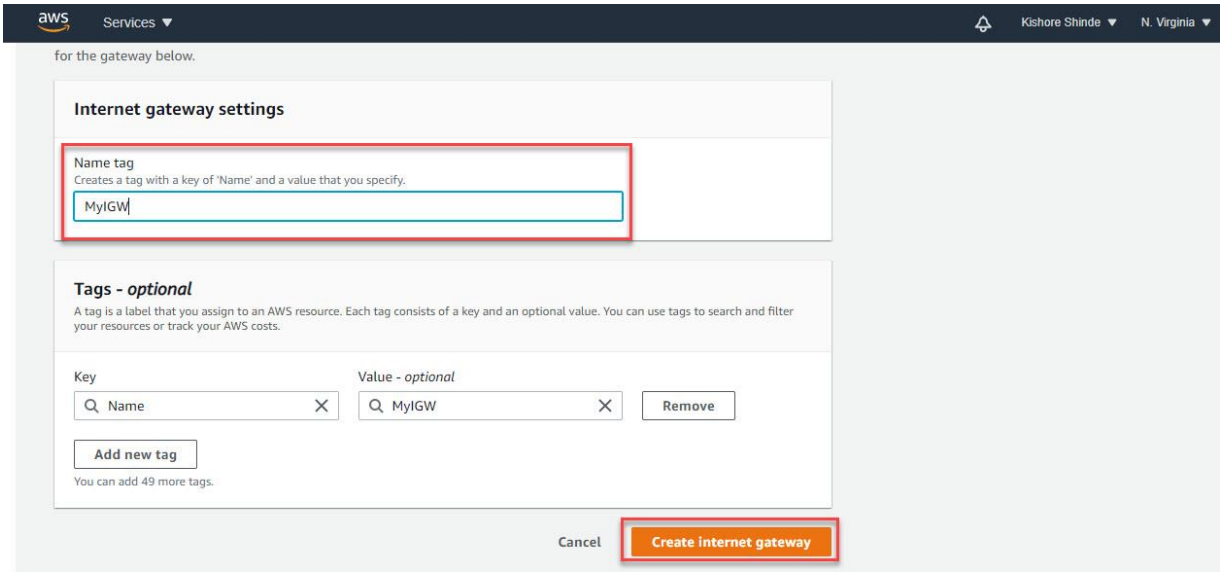
STEP 2 : Create Internet Gateway & associate with Public Subnet

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**

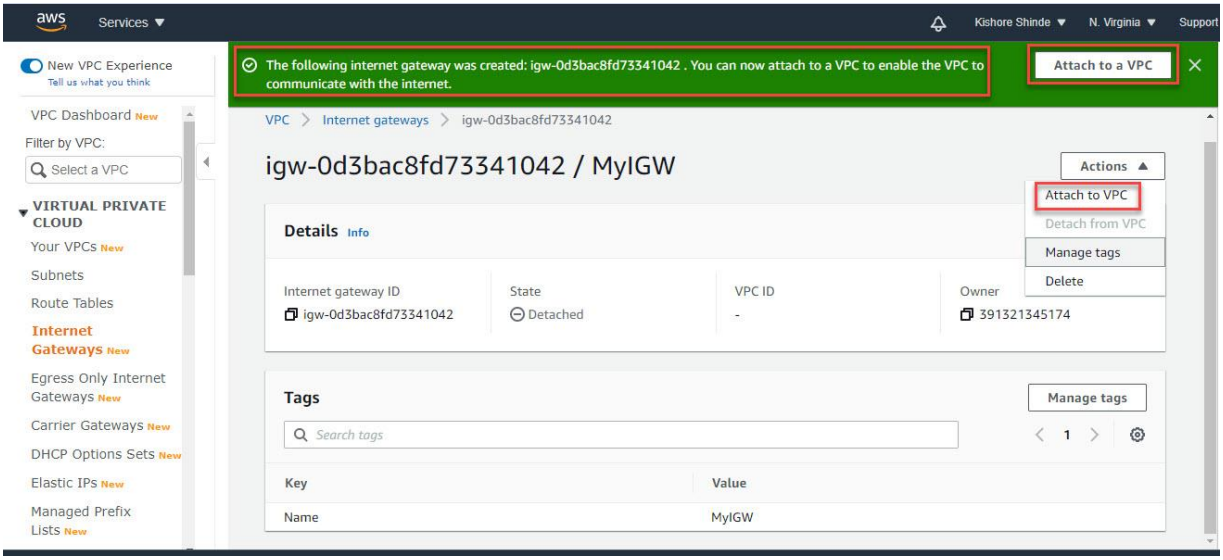


To create new internet gateway specify the name for the gateway below:



- Name : **MyIGW**
- Click on **Create internet gateway**

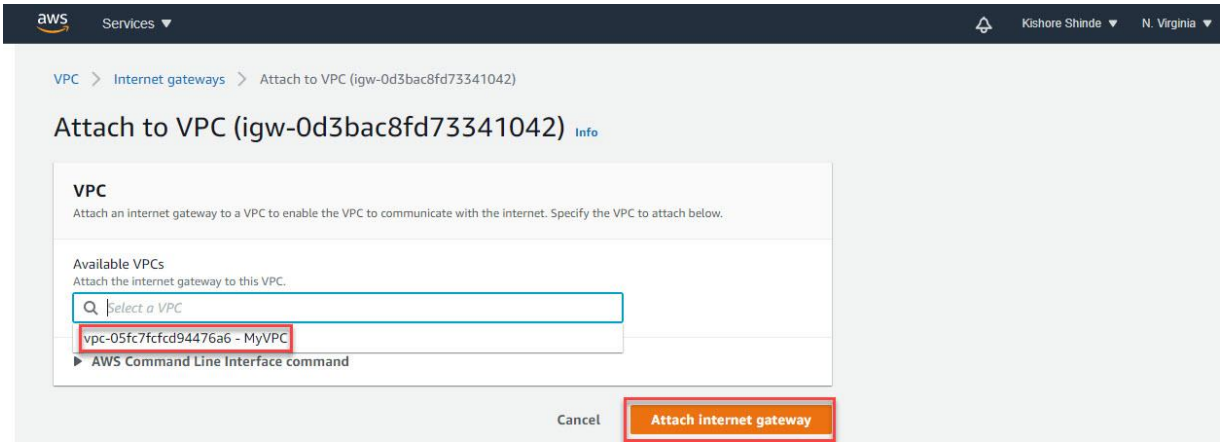
An Internet gateway is created. Now let us attach VPC.



- Click on **Attach to a VPC** at the top

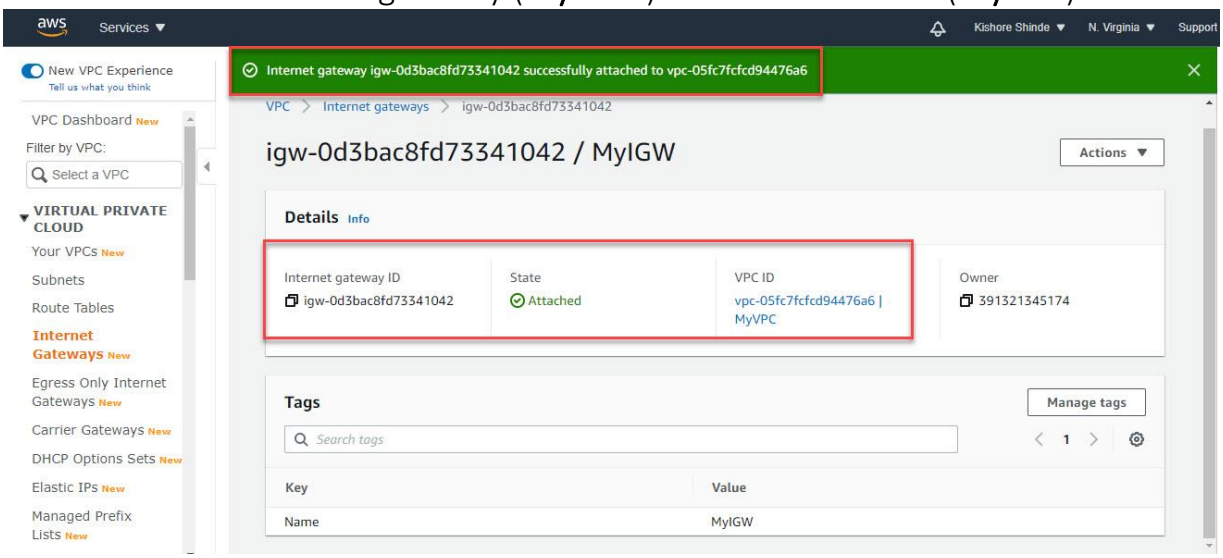
or

- Click on **Actions** and select **Attach to VPC**



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

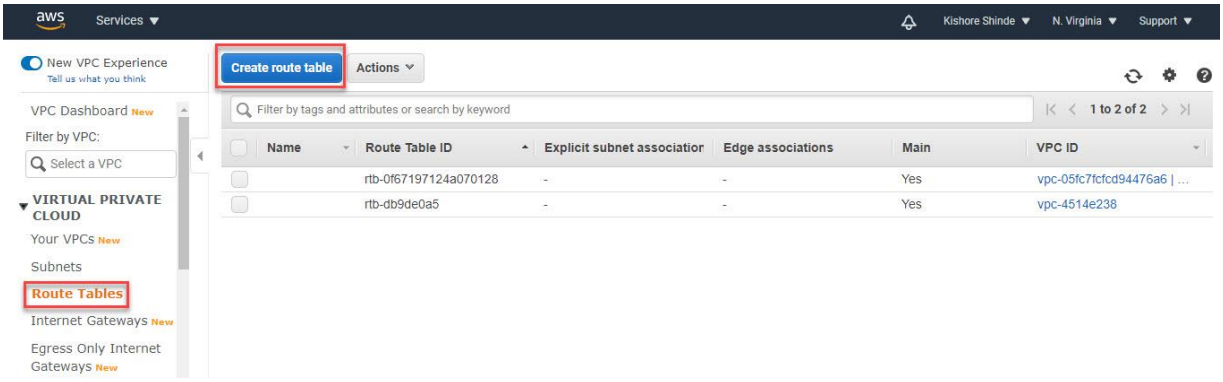
You can see the internet gateway (**MyIGW**) is attached to VPC (**MyVPC**)



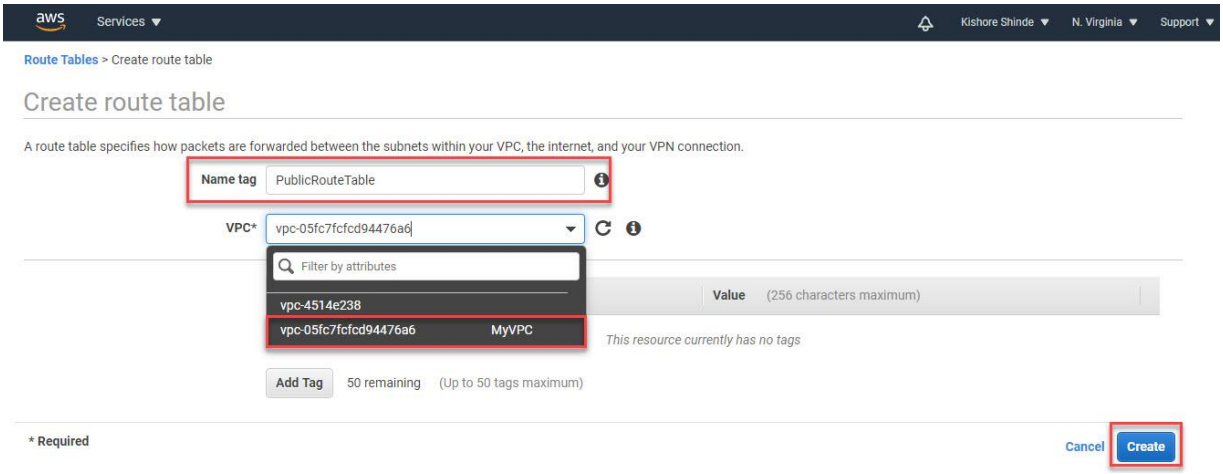
Task 4 : Create Public Route Table and Configure

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



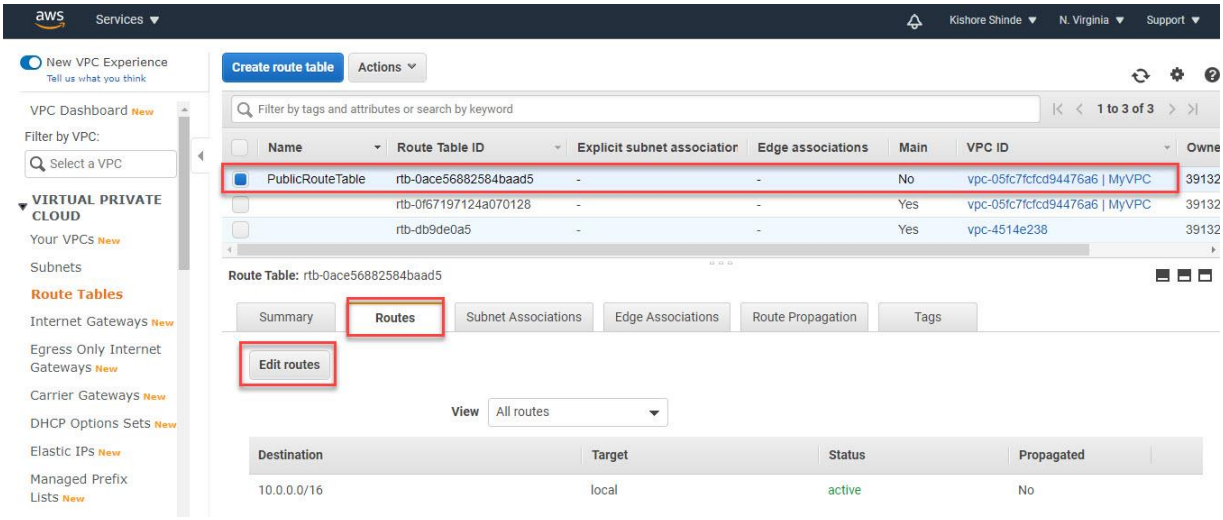
2. Click on **Create route table**



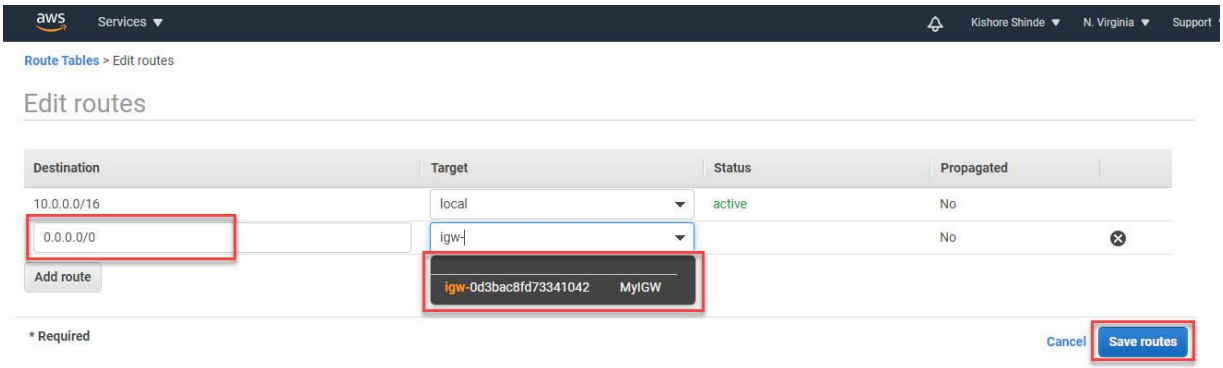
- Name tag : **PublicRouteTable**
- VPC : Select **MyVPC**
- Click on **Create**

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

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



4. In the **Routes** tab, Click on **Edit routes**
5. On the next screen click on **Add route**



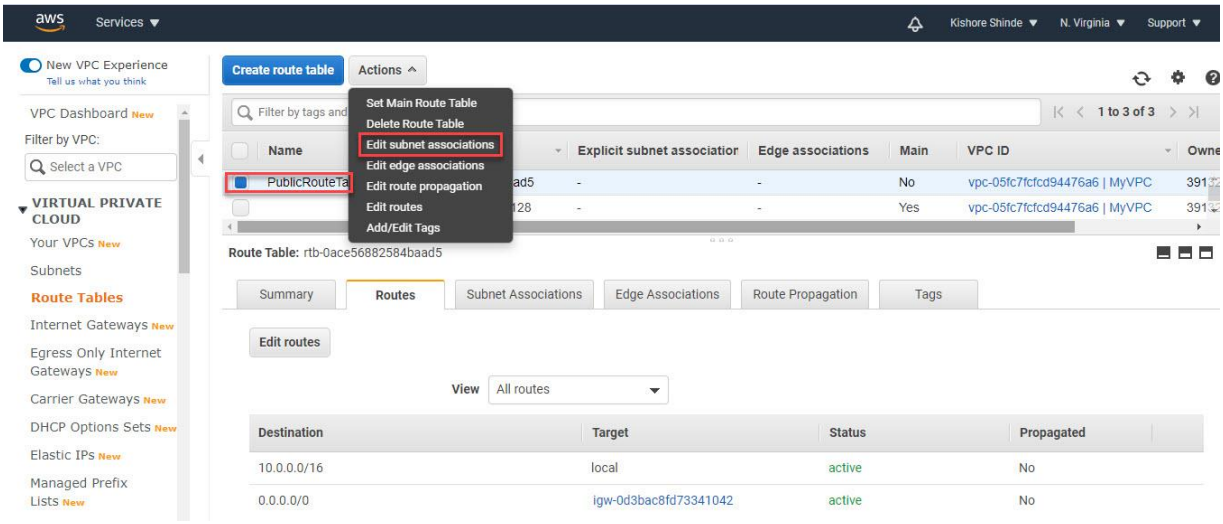
- Destination : Enter **0.0.0.0/0**
- Target : Select **Internet Gateway**, and then from the list select **MyIGW**
- Click on **Save routes**

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

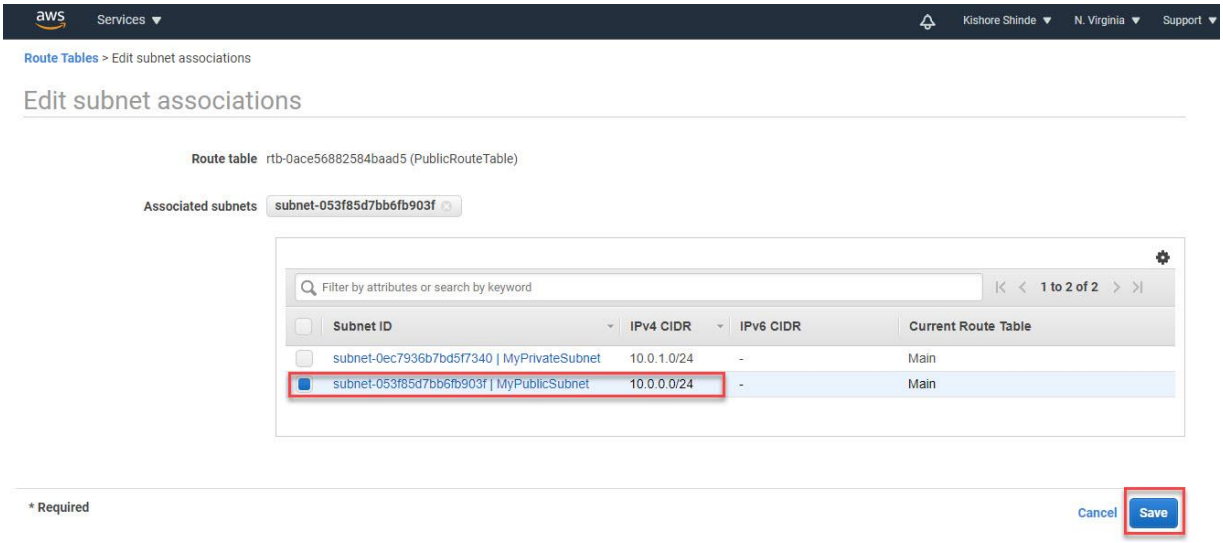
***Note** : We have created the route table, edited the routes and attached the internet gateway, now the resources inside the public subnet can communicate to the internet.*

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

- Select the route table : **PublicRouteTable**

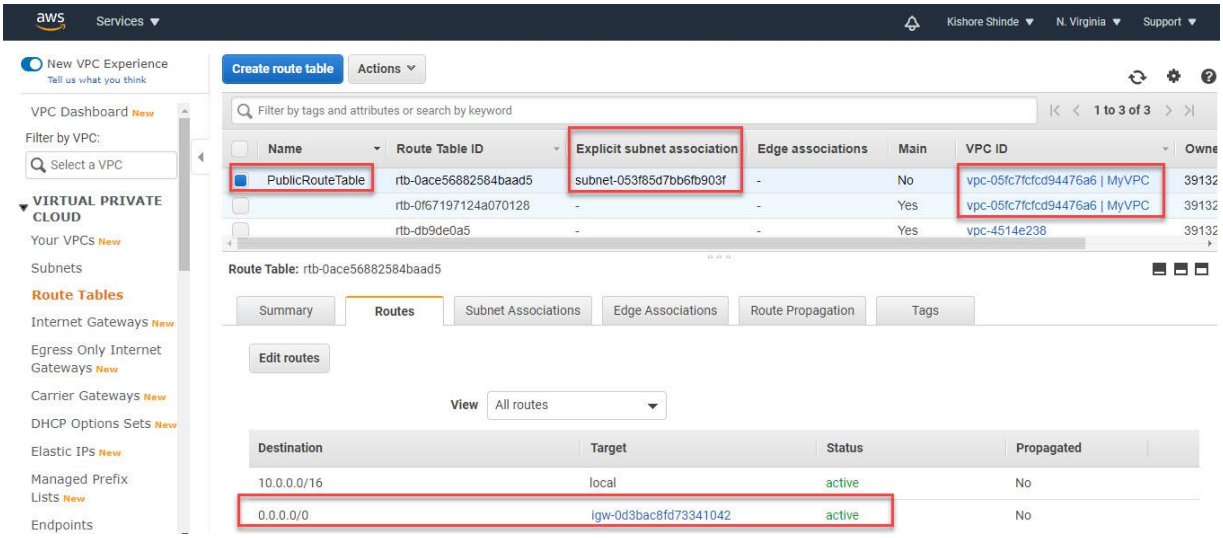


- Click on Actions & select **Edit subnet associations**
- On the next screen, select **MyPublicSubnet**



- Click on **Save**

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



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.**

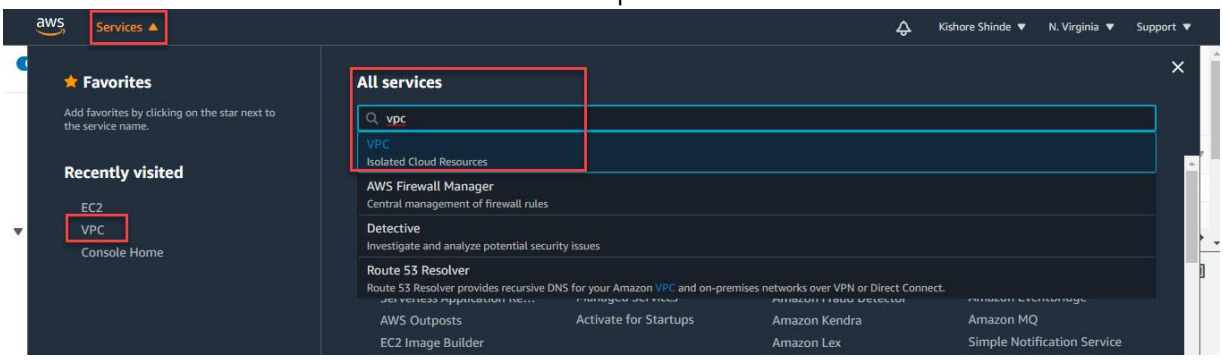
It is the **Main** route table created at the time the VPC was created. We will use it when we create **NAT Gateway**.

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

Step 3 : Creating NAT Gateway & associate with Private Subnet

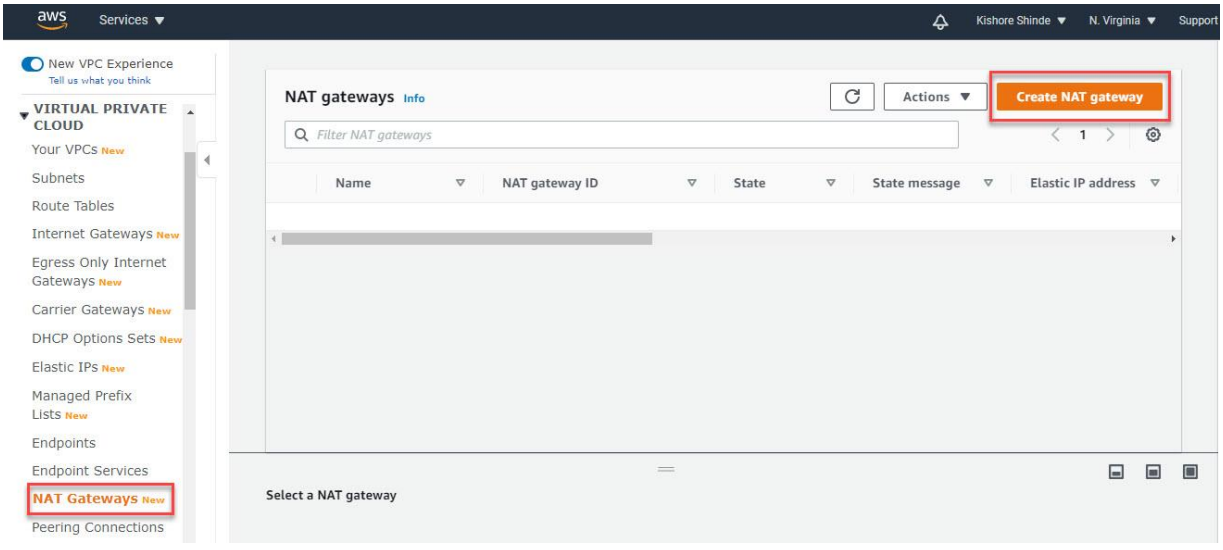
Task 1 : Create NAT Gateway

1. Click on **Services** at the left top



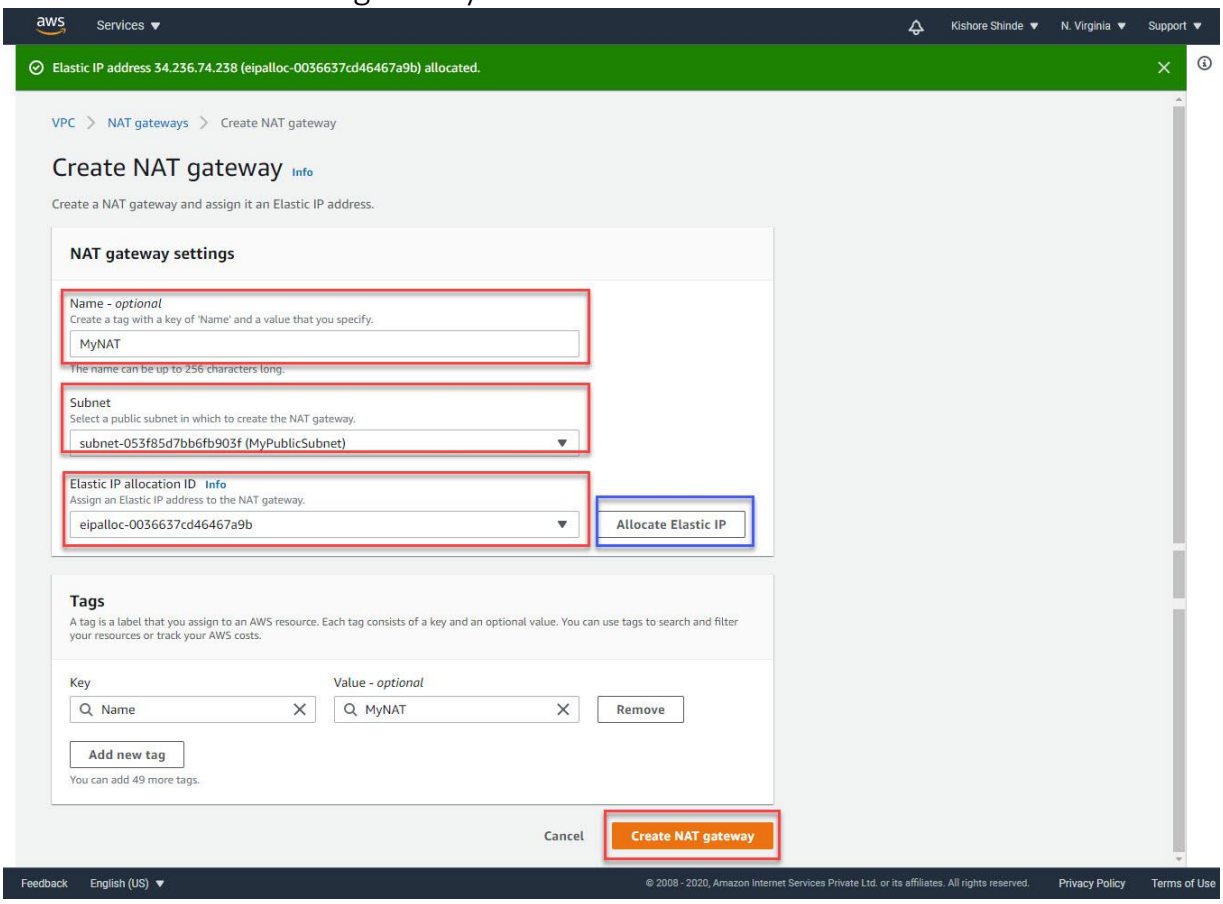
2. Select **VPC** from the **Recently visited** or Search for VPC under **All services** and select it

3. Select **NAT Gateways**



4. Click on **Create NAT gateway**

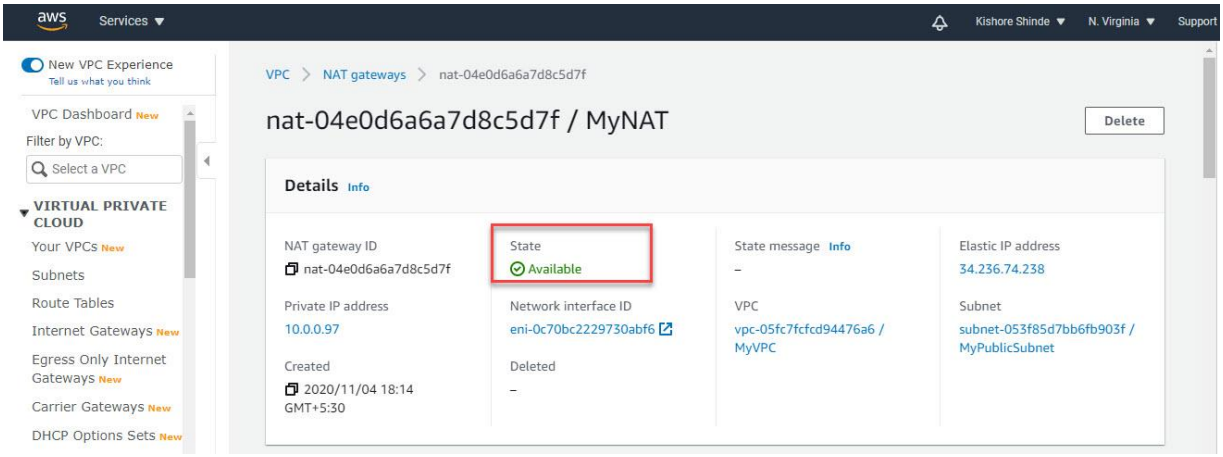
5. Create NAT gateway :



- 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.

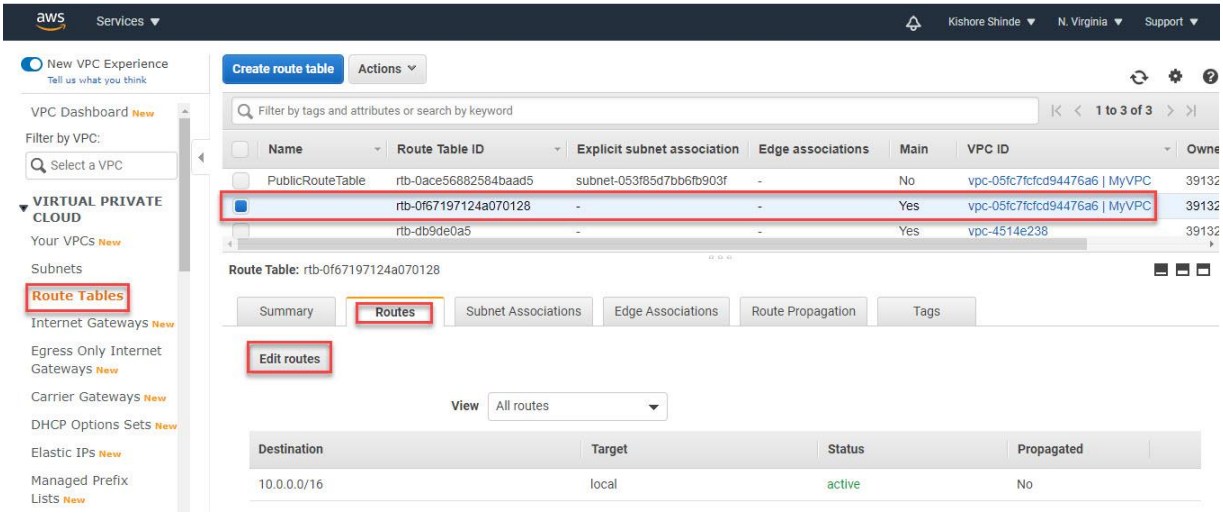
6. We will see the NAT gateway is created.



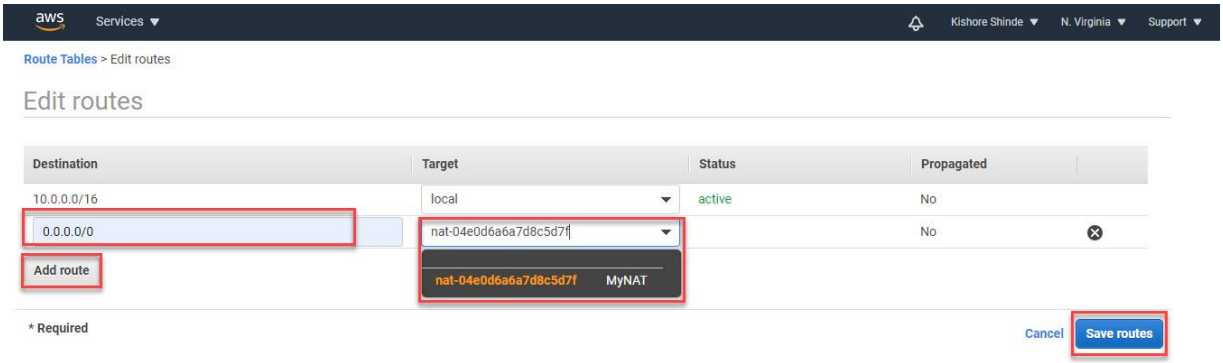
- Wait for the **State** to change from **Pending** to **Available**.

Task 2 : Update Route Table and Configure NAT Gateway

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



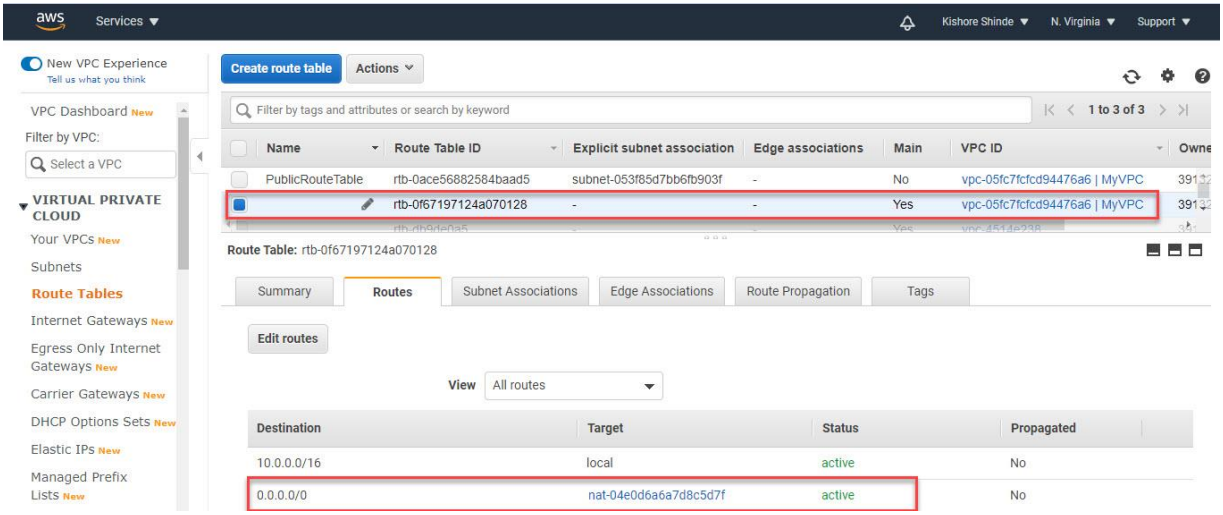
2. You can see two route tables available for **MyVPC**
3. Select the **Main** route table (*Main : Yes*)
4. In **Routes** tab click on **Edit Routes**
5. In next page **Edit Routes** :
    - Click on **Add route**



- 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.

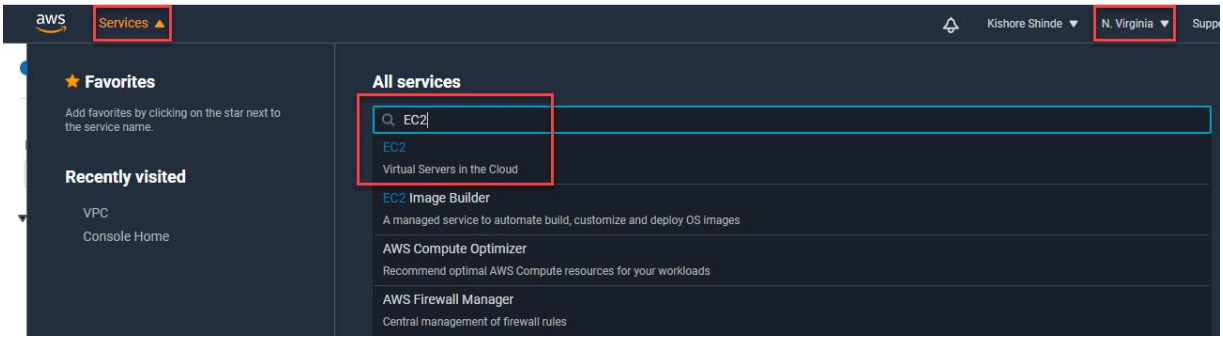


Note : *We have created a private subnet and NAT gateway. The private subnet is attached to a route table, to route traffic via NAT gateway to the internet.*

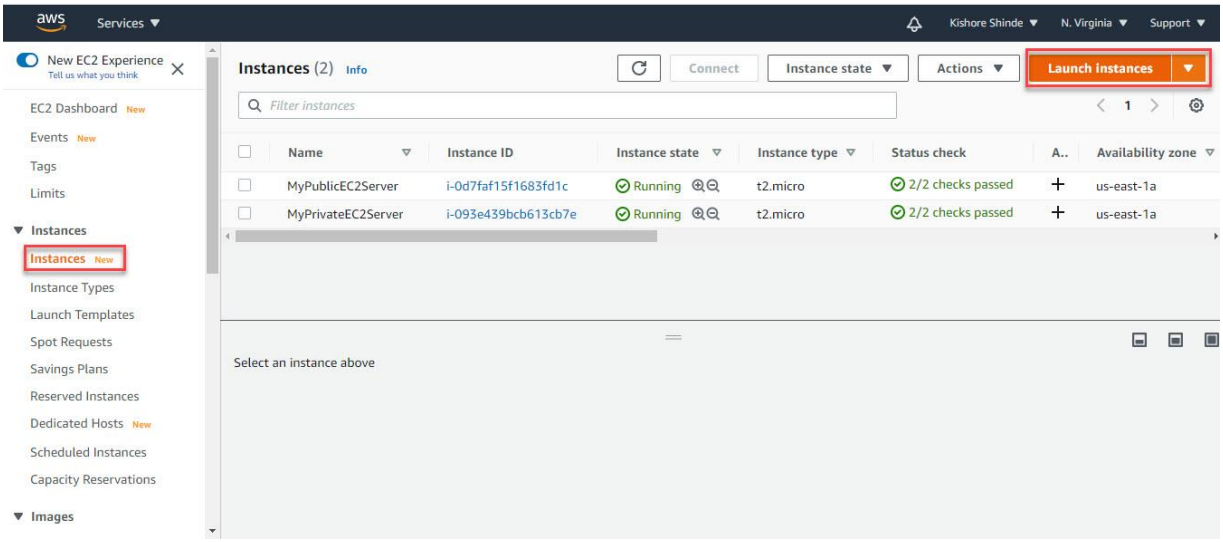
Step 4 Launch Bastion Host in Public Subnet

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.

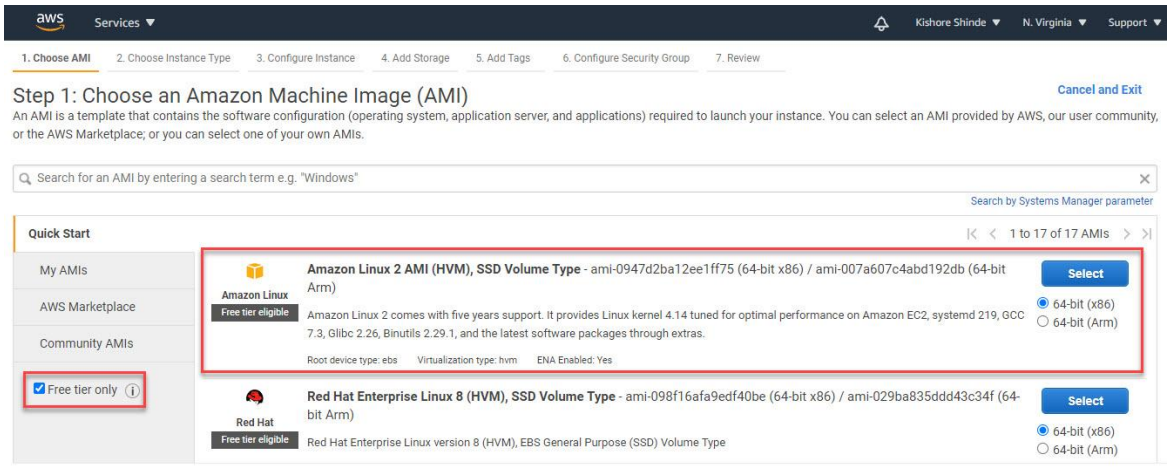
1. Make sure you have selected **N. Virginia** region
2. Click on **Services** on the top and search for **EC2** & select it



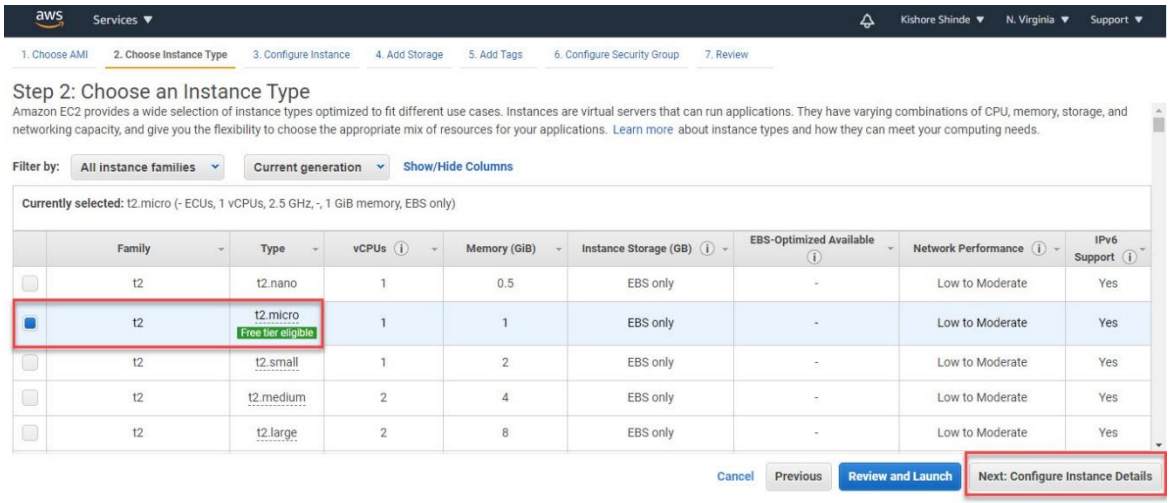
3. Go to **Instances** & click on **Launch Instances**



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

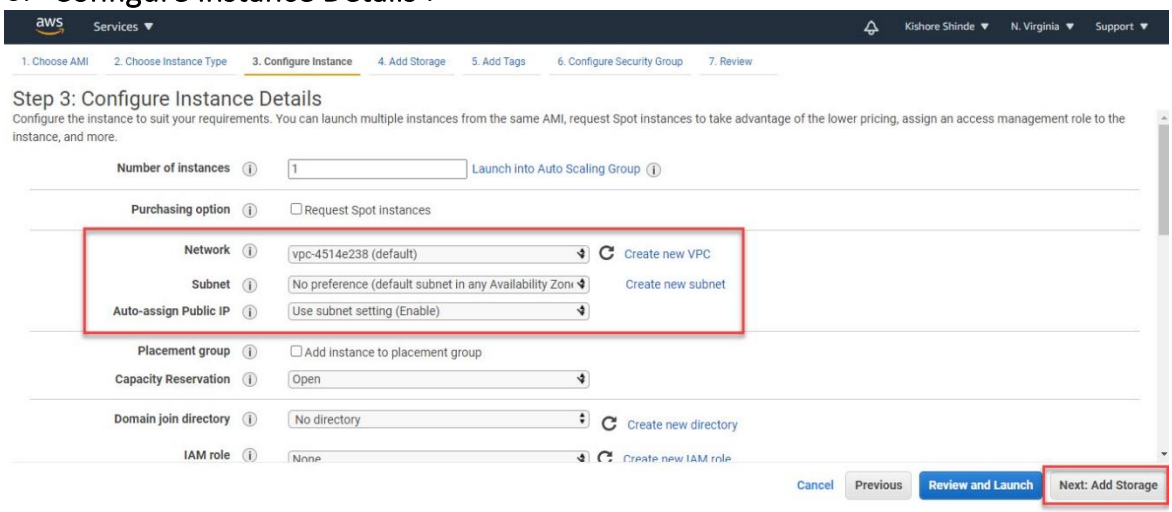


5. **Instance Type** : Select **t2.micro** (free-tier eligible)



Click on **Next : Configure Instance Details**

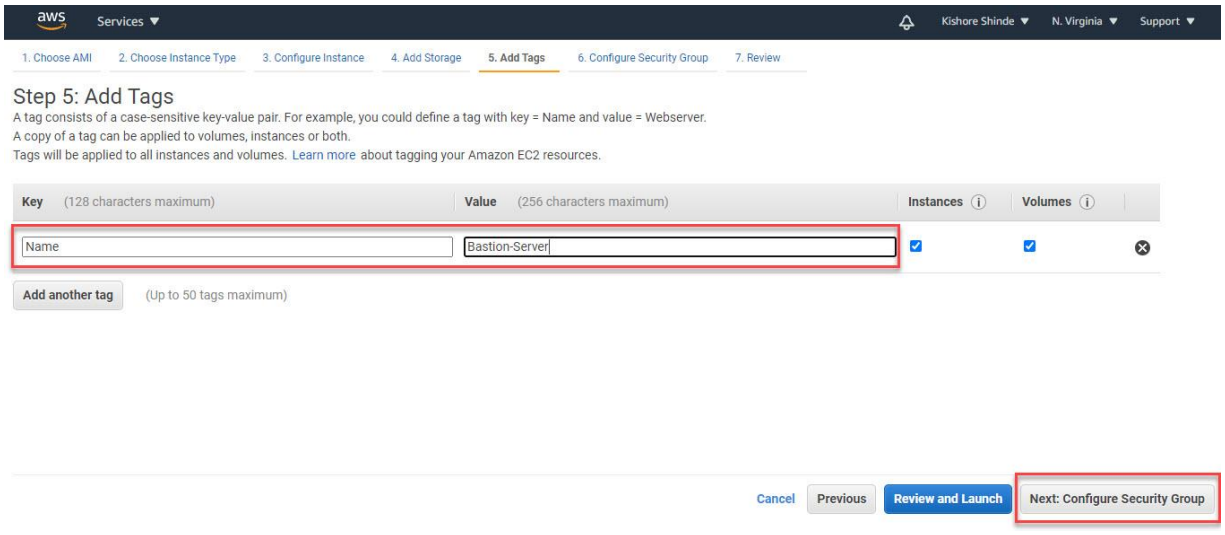
6. Configure Instance Details :



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

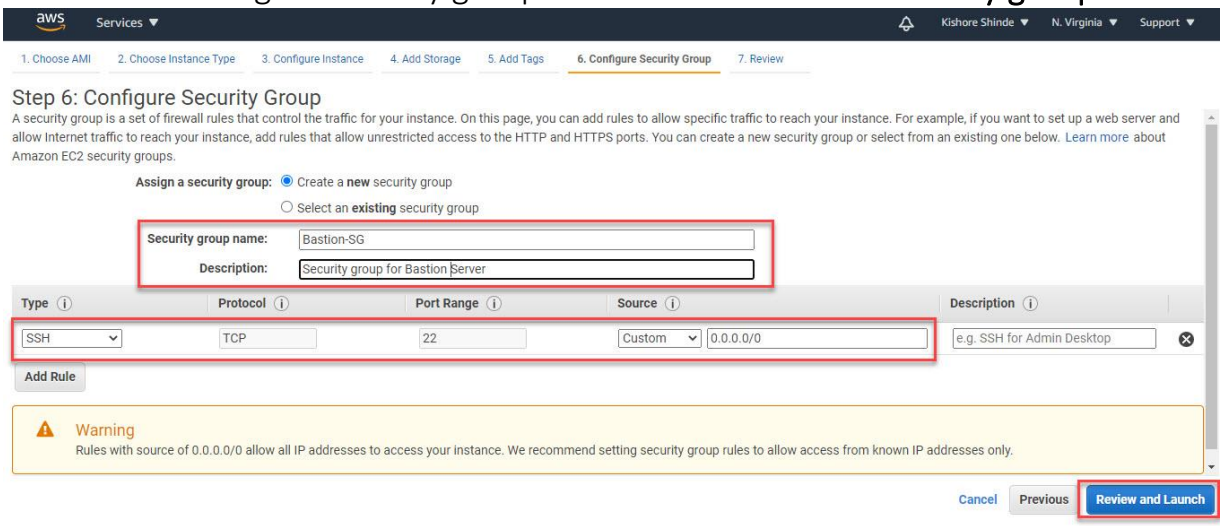
7. **Add Storage** : No need to change anything in this step and click on **Next : Add Tags**

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



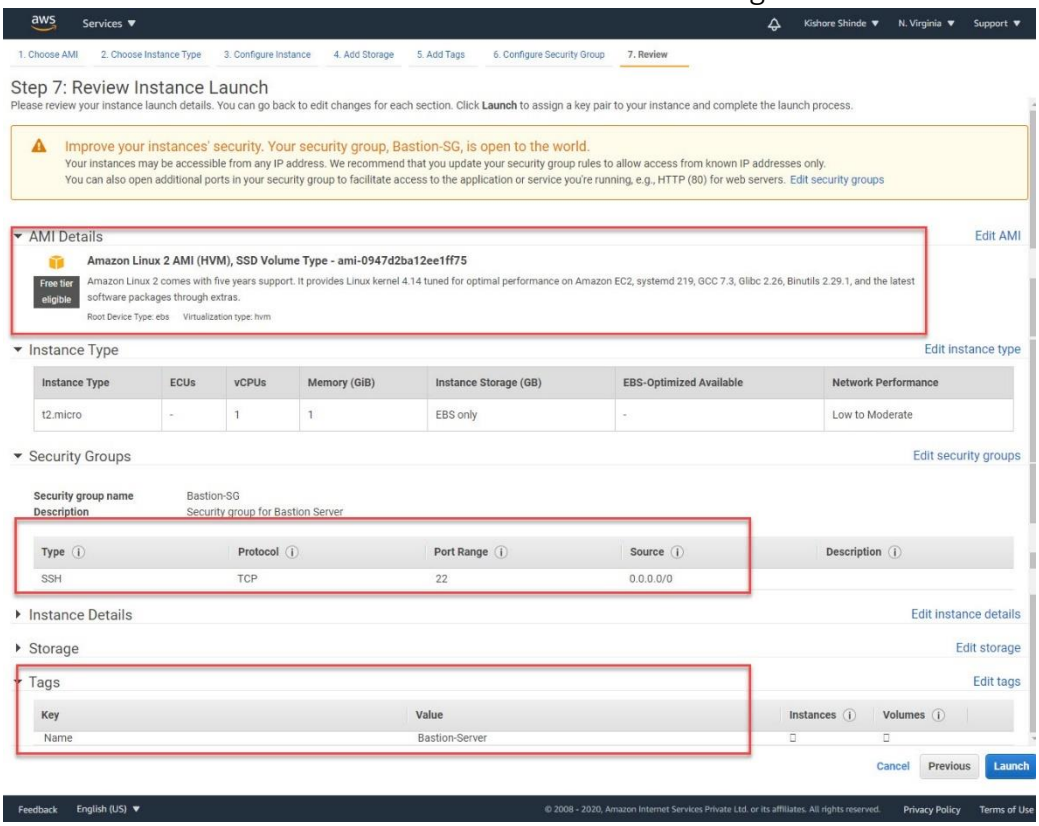
- Key : **Name**
- Value : **Bastion-Server**
- Click on **Next : Configure Security Group**

9. **Configure Security Group** :
    - Assign a security group – Select **Create a new security group**



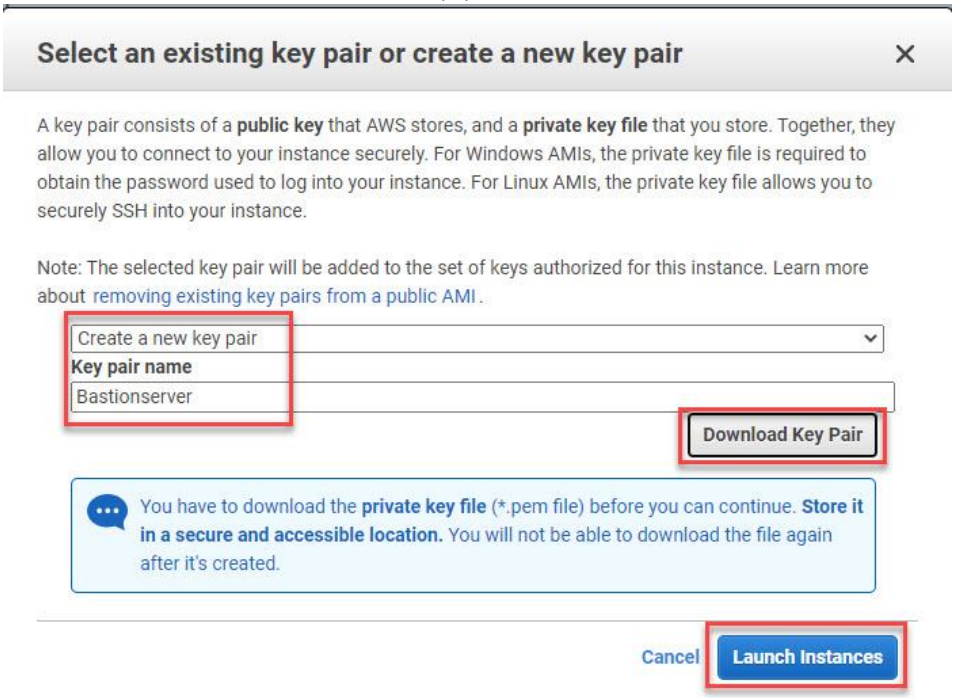
    - Security Group Name : **Bastion-SG**
    - Description : **Security group for Bastion Server**
    - Make sure you select type : **SSH** (*will be default selected*)
    - Source : **Custom – 0.0.0.0/0**
    - Click on **Review and Launch**

10. **Review Instance Launch** : Review all the settings



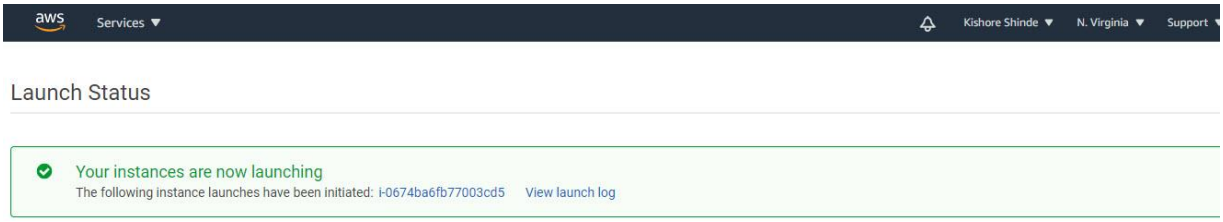
    - Click on **Launch**

10. **Key Pair** :
    - Create a new key pair : **Bastionserver**



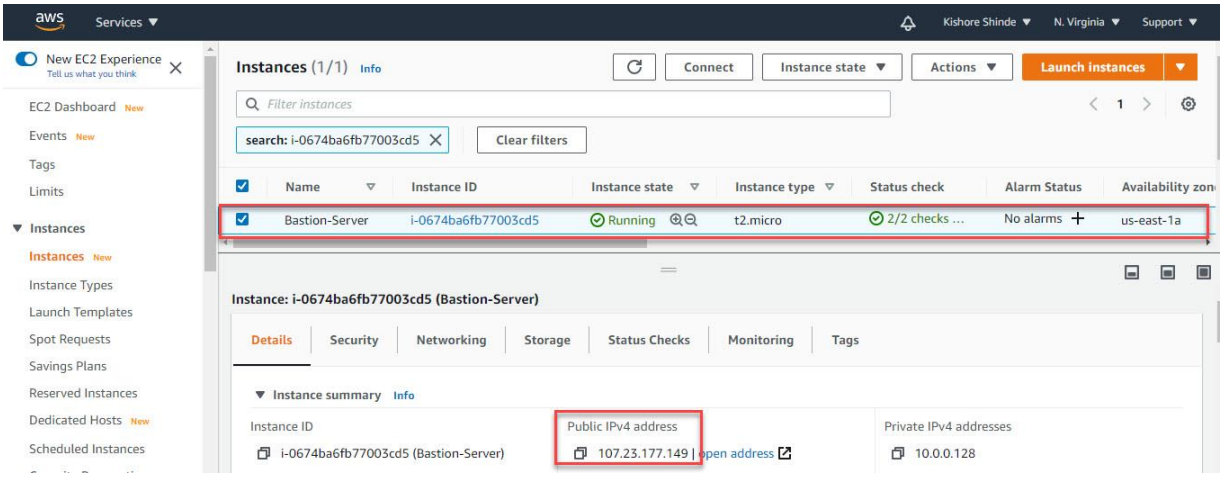
    - Click on **Download Key Pair**
    - Click on **Launch Instances**

11. **Launch Status :**
    - Select the instance id



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

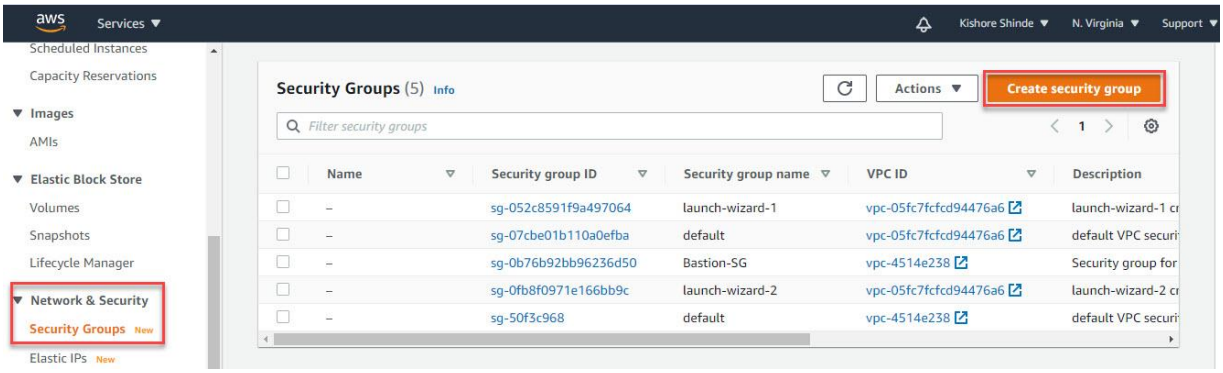
12. **Bastion Host Details** : Bastion Host is launched successfully.



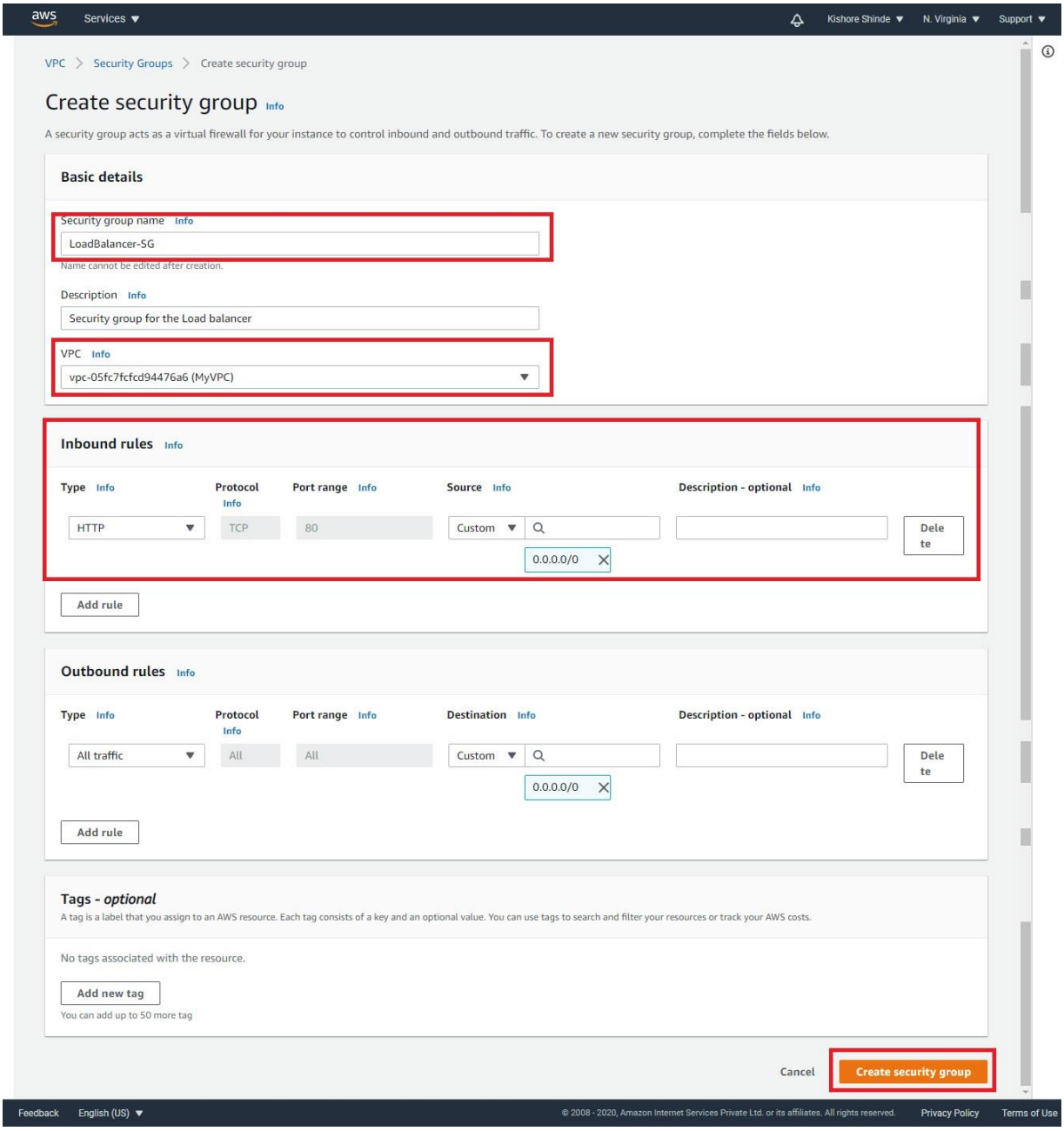
- Bastion Host Associated Public IP : **107.23.177.149**

Step 5 Creating a Security Group for the Load Balancer

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

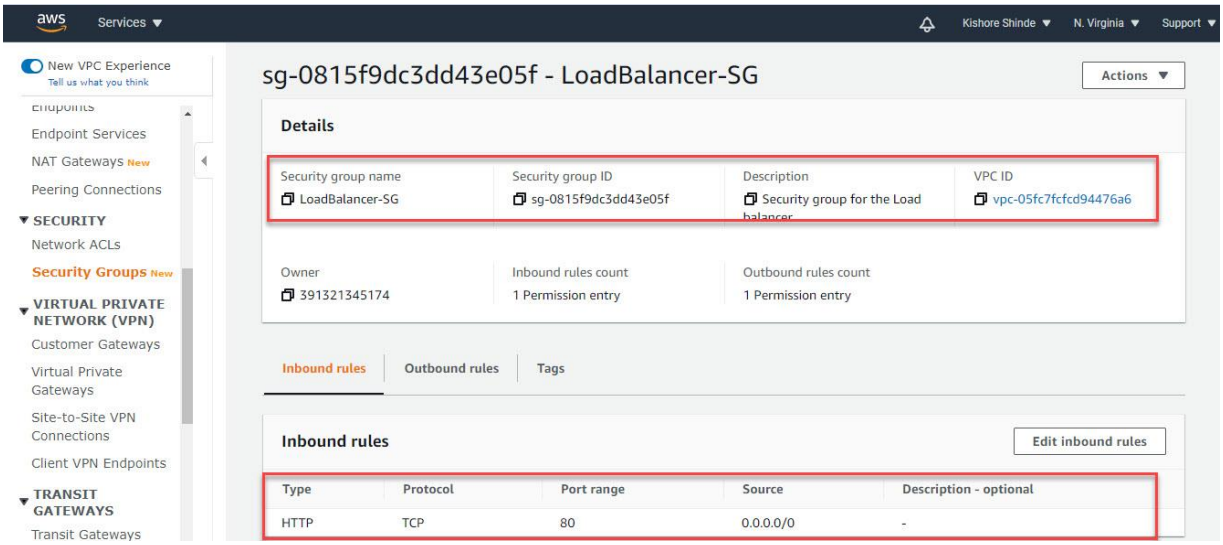


2. Click on **Create security group**



- 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**

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



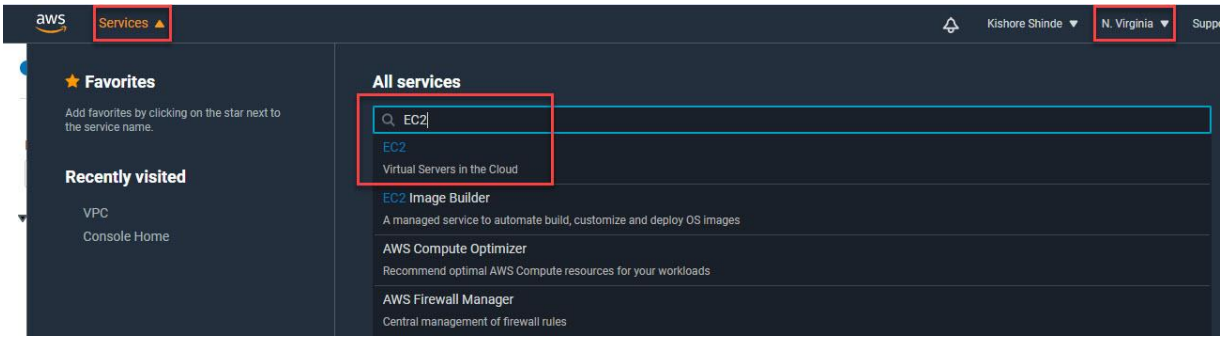
- Check the **Inbound rules** with added HTTP rule

Step 6 Launch two Web Servers securely in Private Subnet

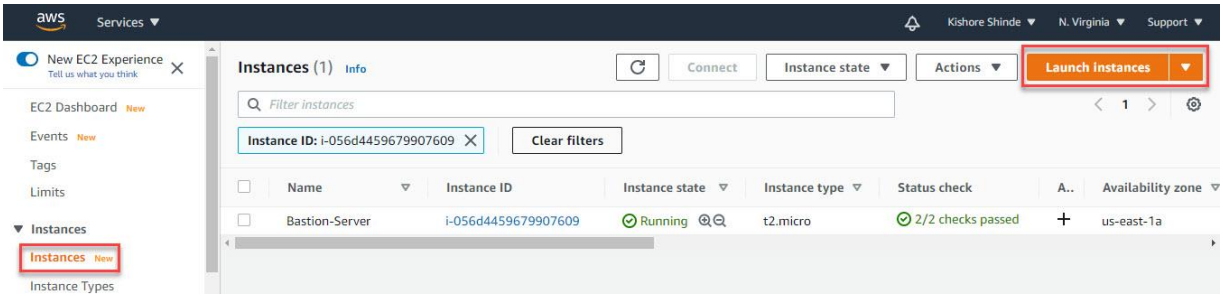
We have created a private subnet and NAT gateway. The private subnet is attached to a route table to route traffic via NAT gateway to the internet. Now let us create two webservers in private subnet.

Task 1 : Launching Web Server 1

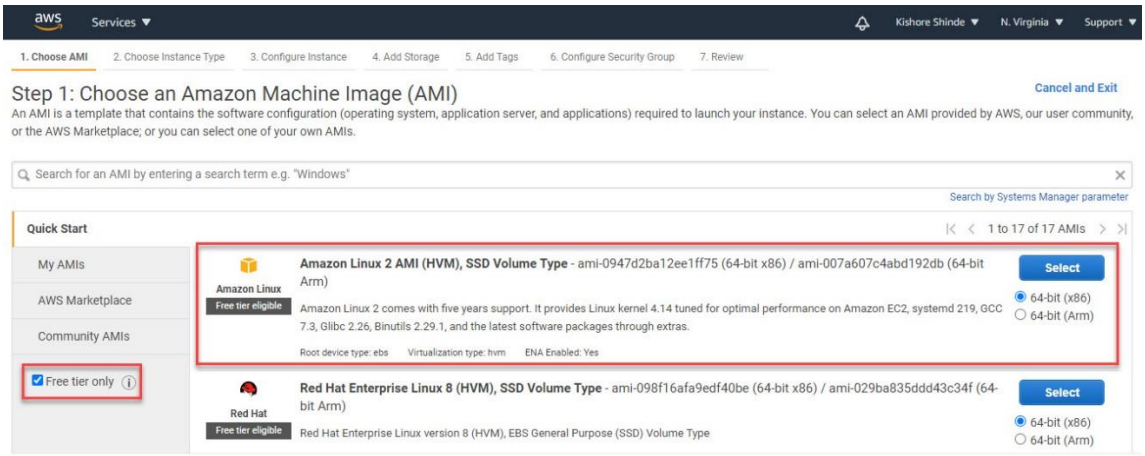
1. Make sure you have selected **N. Virginia** region
2. Click on **Services** on the top and search for **EC2** & select it



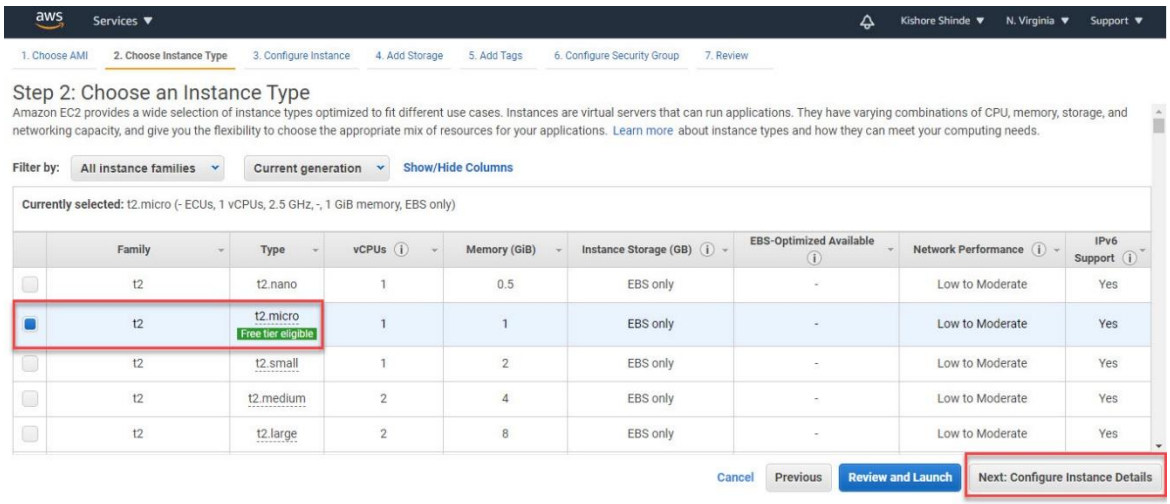
3. Go to **Instances** & click on **Launch Instances**



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

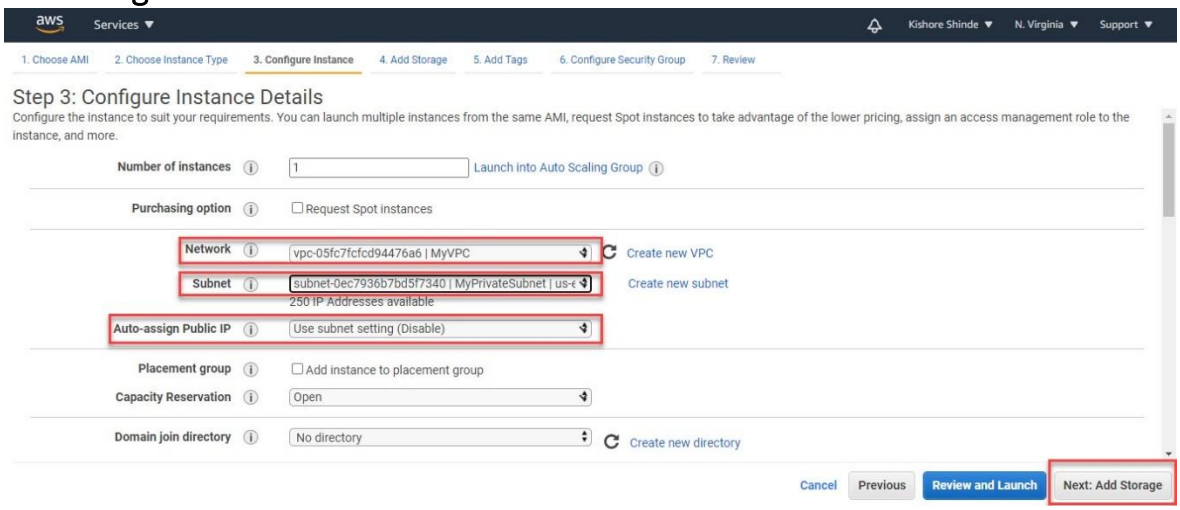


5. **Instance Type** : Select **t2.micro** (free-tier eligible)



Click on **Next : Configure Instance Details**

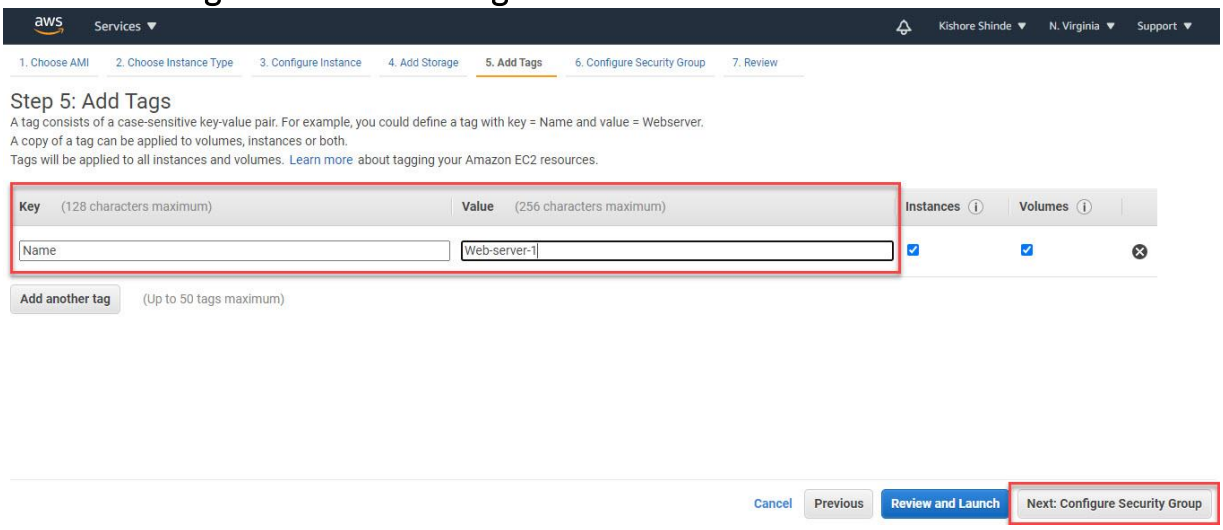
6. **Configure Instance Details** :



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

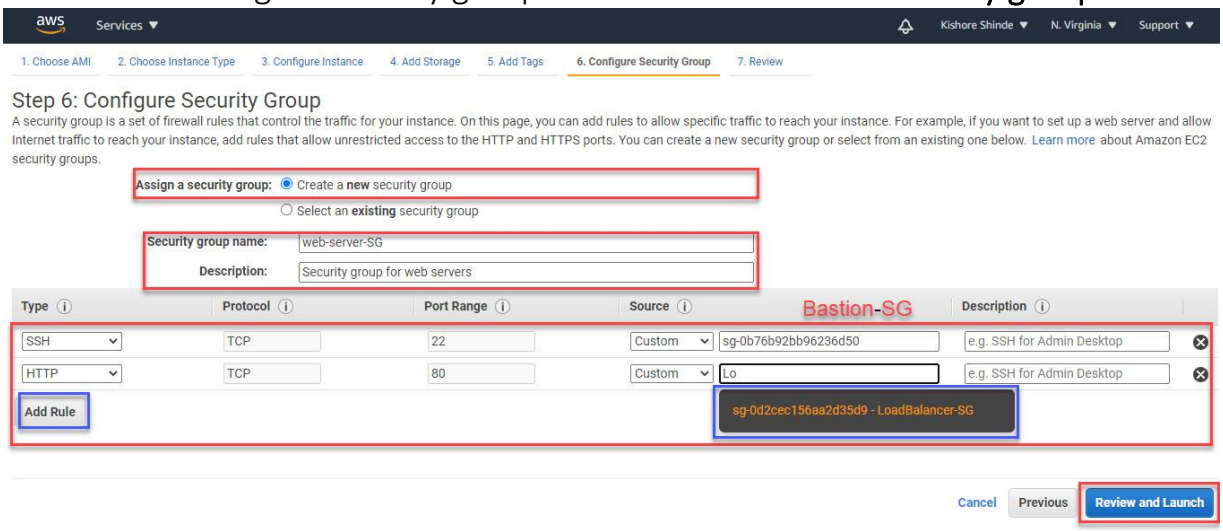
7. **Add Storage** : No need to change anything in this step and click on **Next : Add Tags**

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



- 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**



- 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**

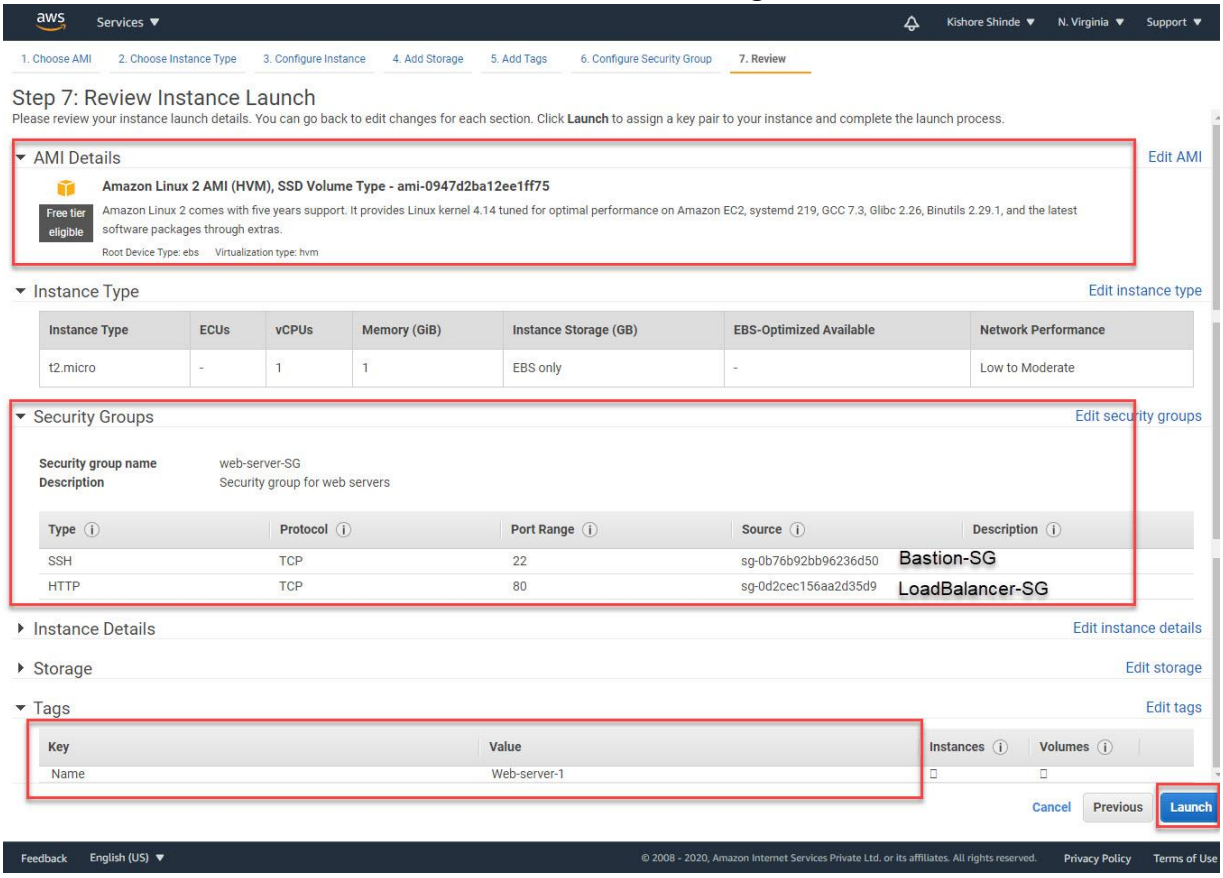
Note : *On port 22, we select **Bastion -SG** security group as its source to allow SSH connection to web servers from only bastion server by restricting the public SSH connection.*

- 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.*

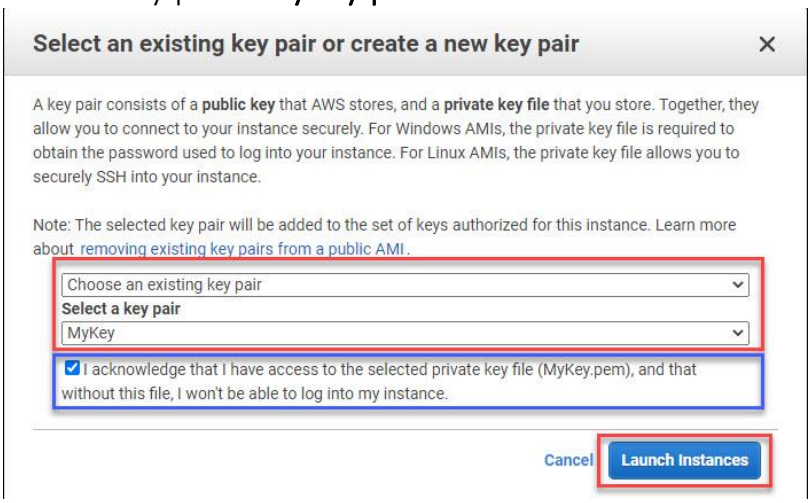
- Click on **Review and Launch**

10. **Review Instance Launch** : Review all the settings



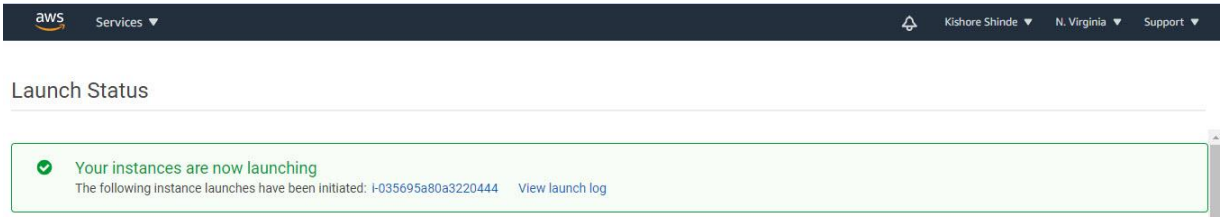
- Click on **Launch**

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



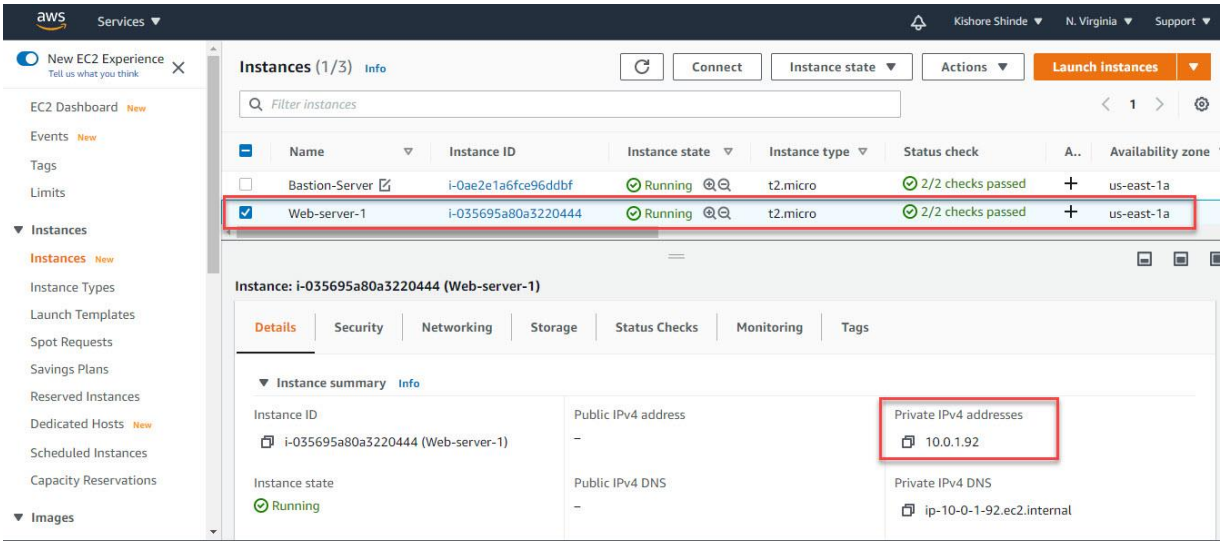
- Select : I acknowledge.. checkbox
- Click on **Launch Instances**

11. Launch Status :
    - Select the instance id



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

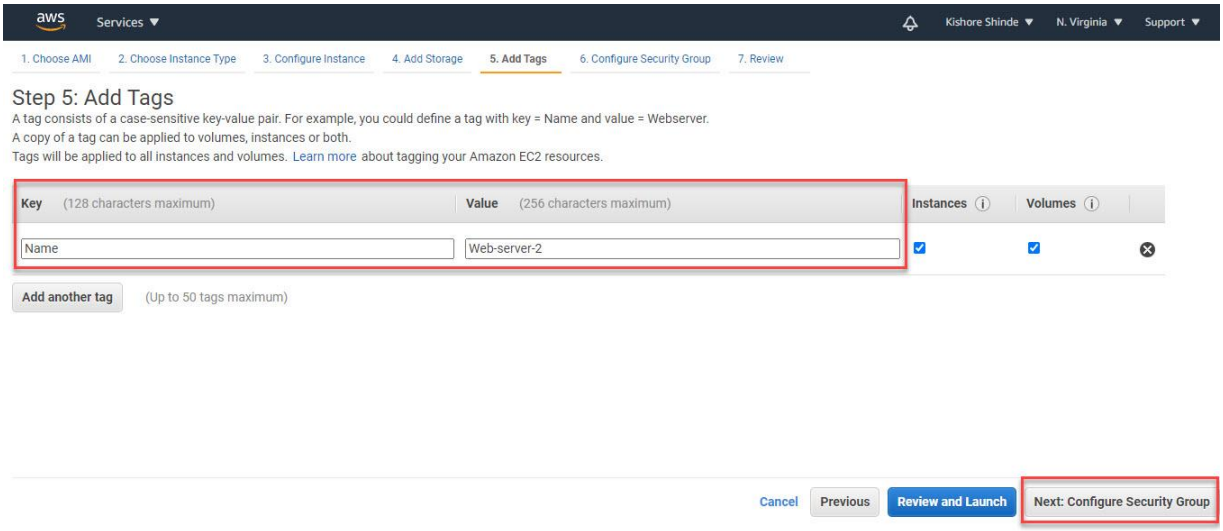
12. **Web-server-1 Details** : You will see new instances Web-server-1 running along with Bastion-server created in the earlier step.



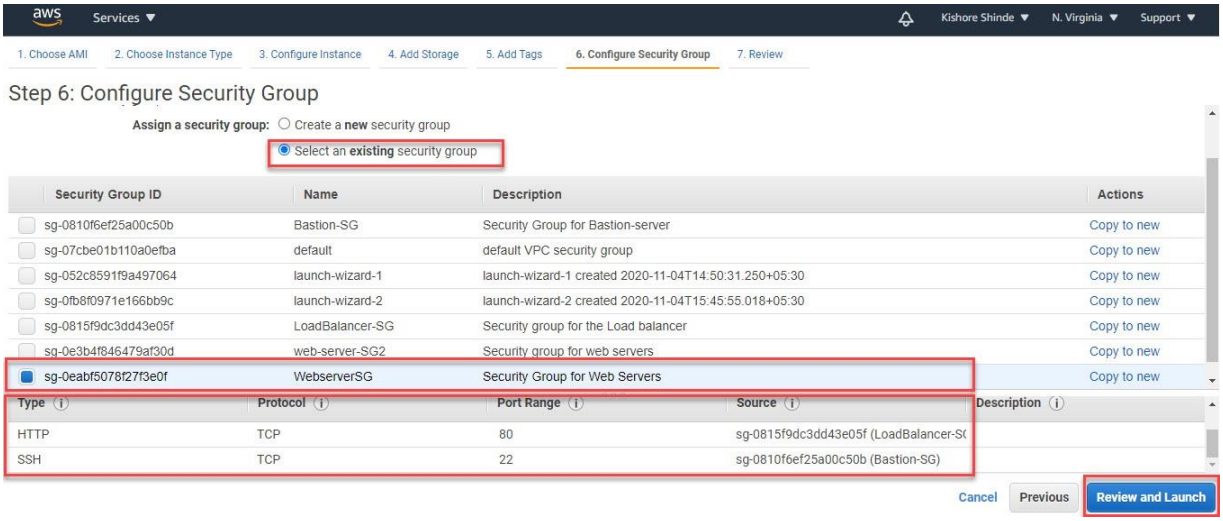
    - **Web-server-1 Private IP : 10.0.1.92**

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**

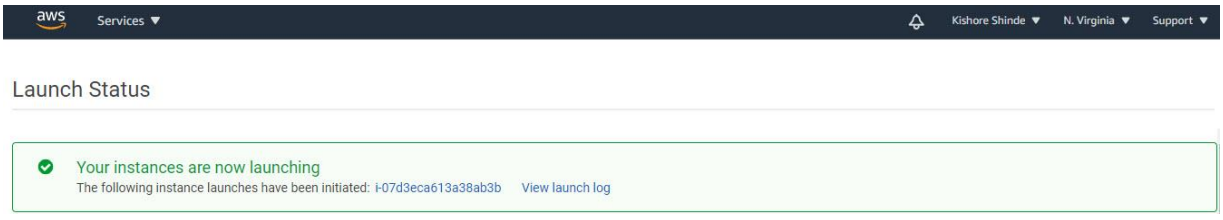


14. **Step 6** : **Configure Security Group** section, select existing group **WebserverSG**



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

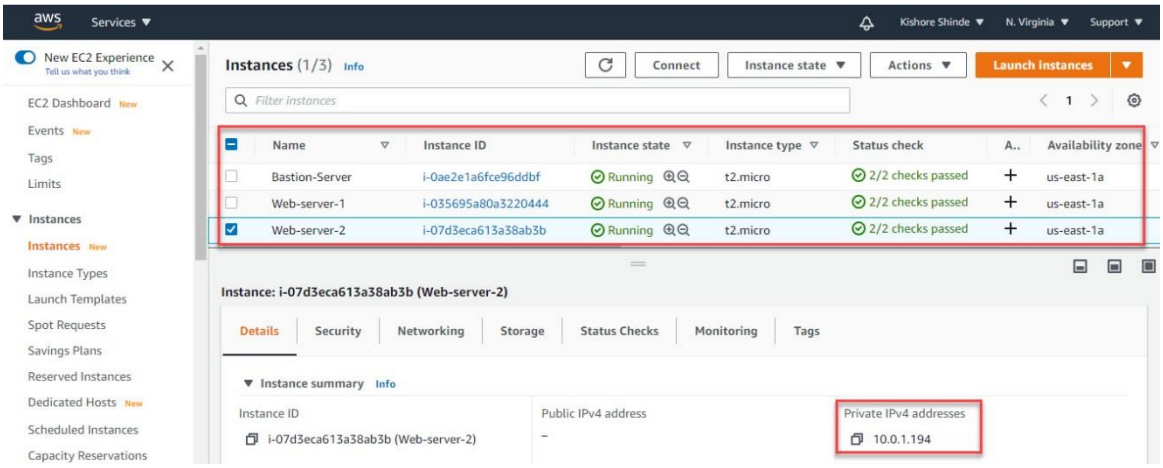
16. In **Launch Status** we can see Your instances are now launching.



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



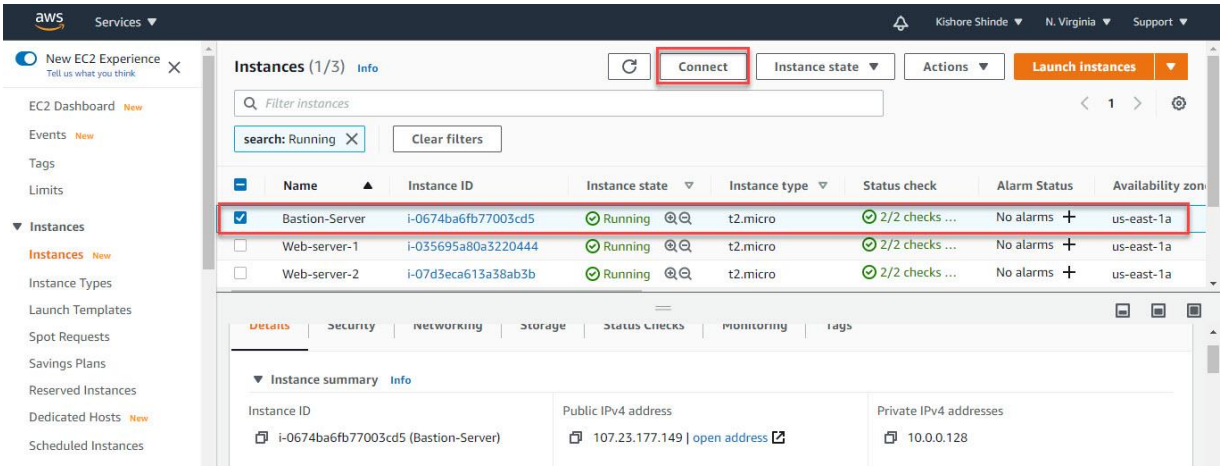
a. Web-server-1 Private IP : 10.0.1.92

b. Web-server-2 Private IP : 10.0.1.194

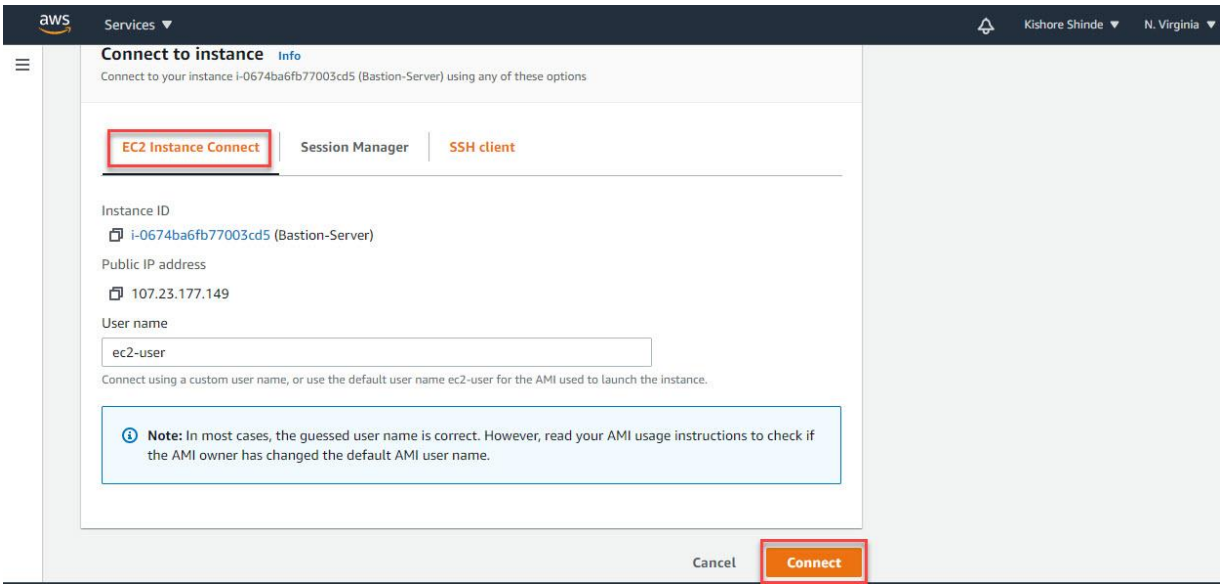
Step 7 : SSH into Web Servers through Bastion Server using RSA private key, Install Apache, Host Page(index.html) on both Web Servers

Task 1 : Connecting to Bastion-Server

1. Go To **EC2 Dashboard** & Click on **Instances**

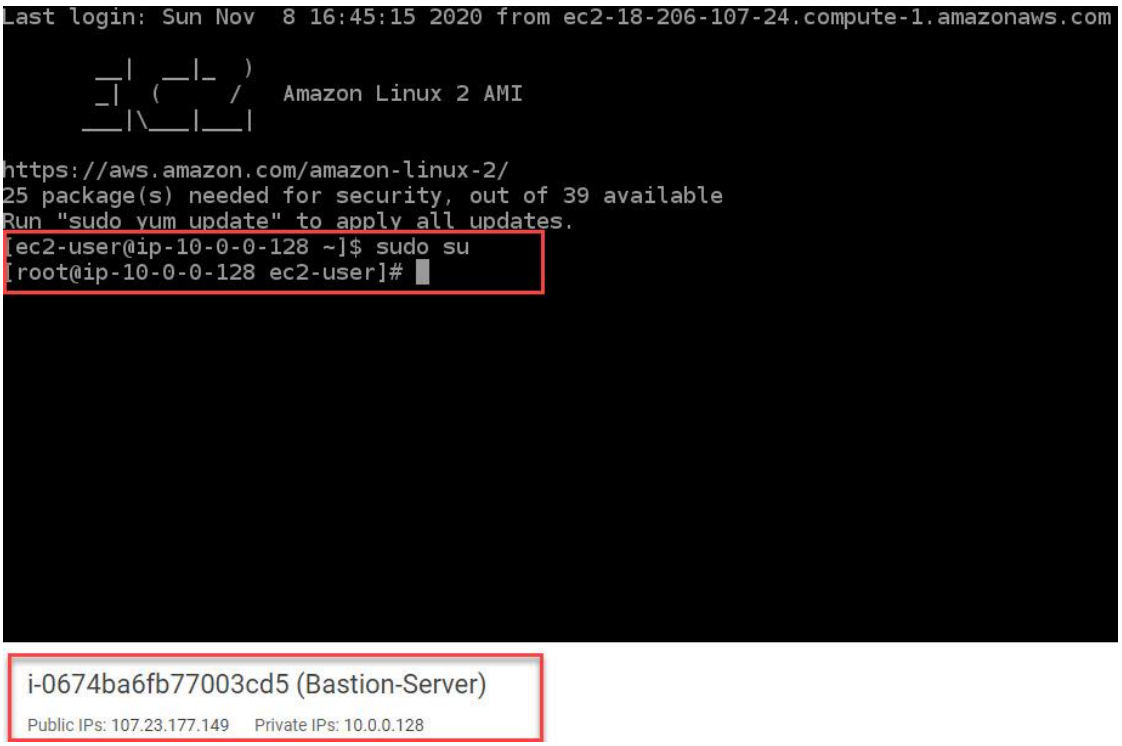


2. From the list of instances, select **Bastion-Server** and click on **Connect.**
3. Connect the instance through **EC2 Instance Connect.**



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

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

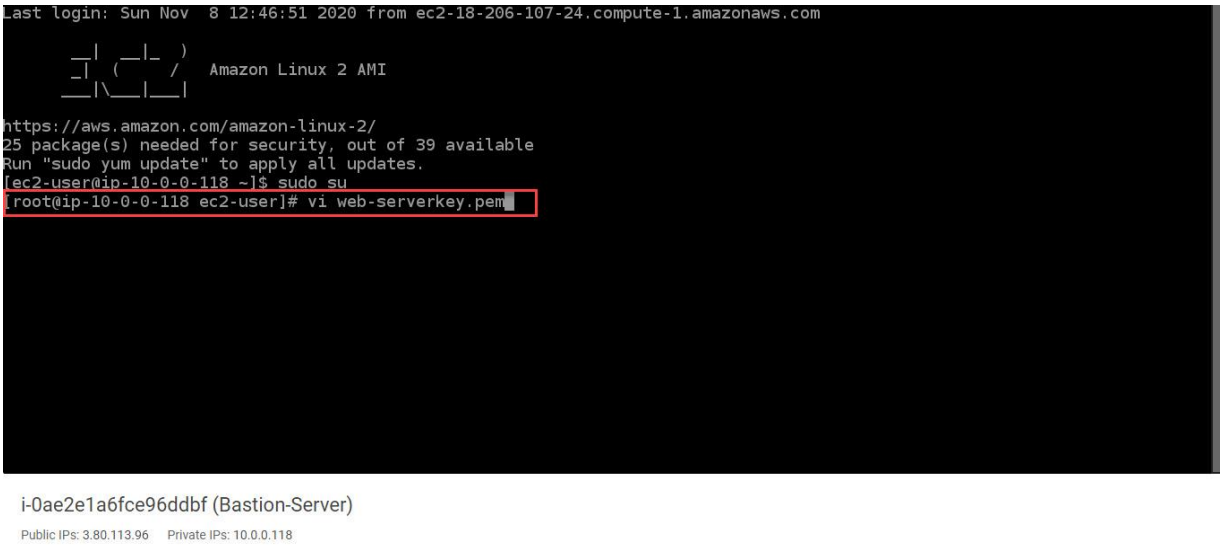


Task 2 : Connect Web-servers through Bastion-Server

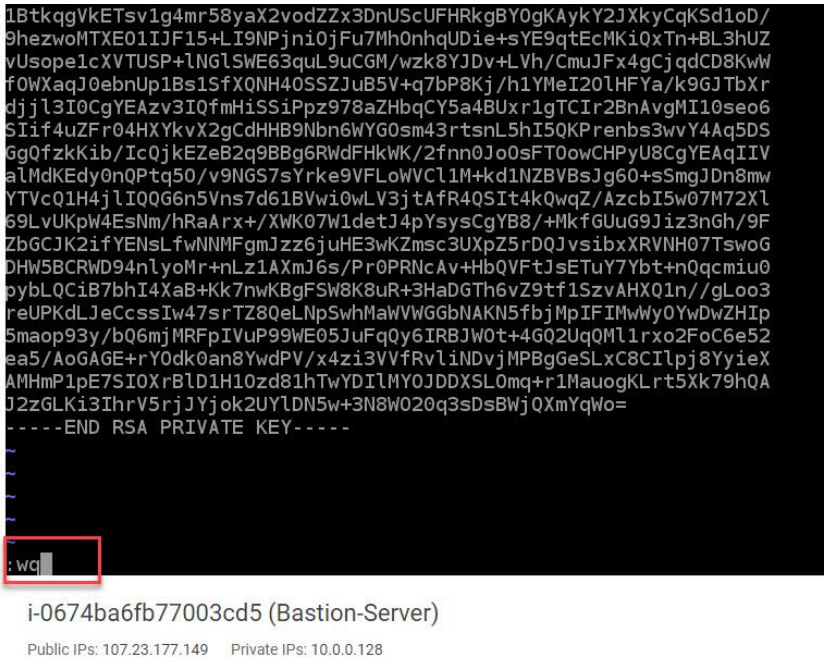
1. To connect to the web servers via Bastion-server, we will need the web server key which we used to launch the web servers.(MyKey.pem)
2. Open the MyKey.pem file on your local system and copy it's content.

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**



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

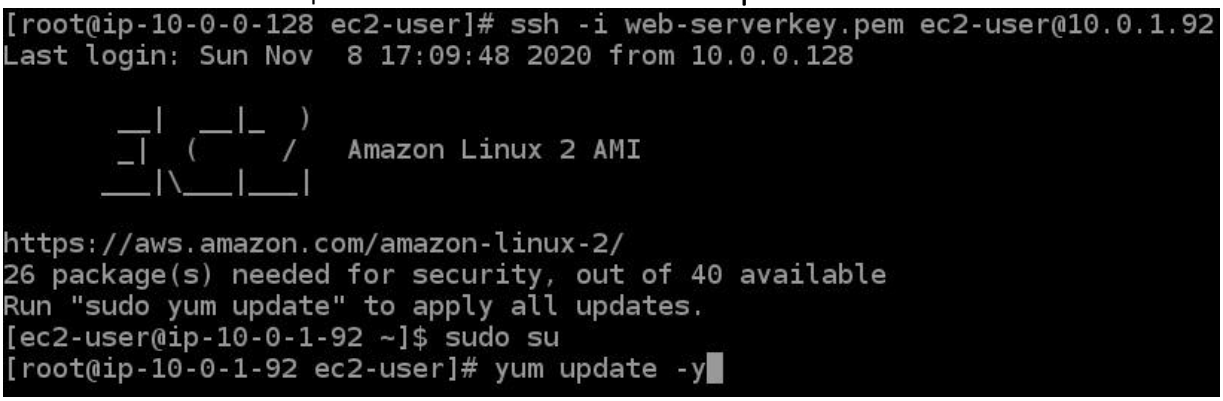


- A file with **web-serverkey.pem** will be created

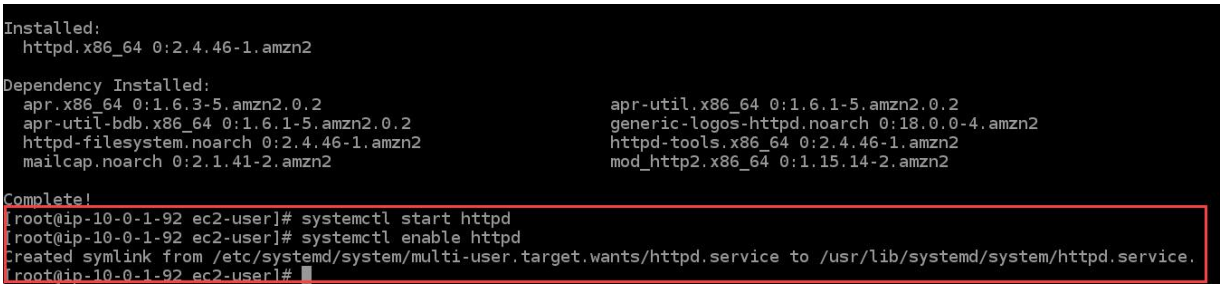
4. Now we will have to change the permission of the pem file. Below is the command :
    - chmod 400 web-serverkey.pem
    - This command will give read-only permission to file.
5. Now we can connect/login to the web servers using this file with the below command:
    - ssh -i web-serverkey.pem ec2-user@10.0.1.92

- 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.

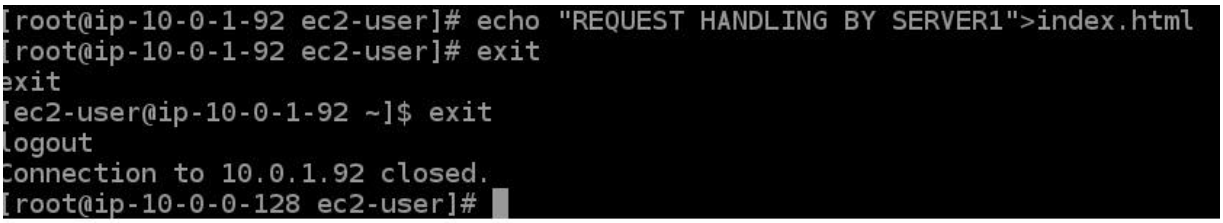


- On Web-server-1 type following commands :
  - **sudo su** – to become root user
  - **yum update -y** – to install latest packages
  - **yum install httpd -y** – to install Apache
  - **systemctl start httpd** – to start Apache
  - **systemctl enable httpd** – to enable Apache



- 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**

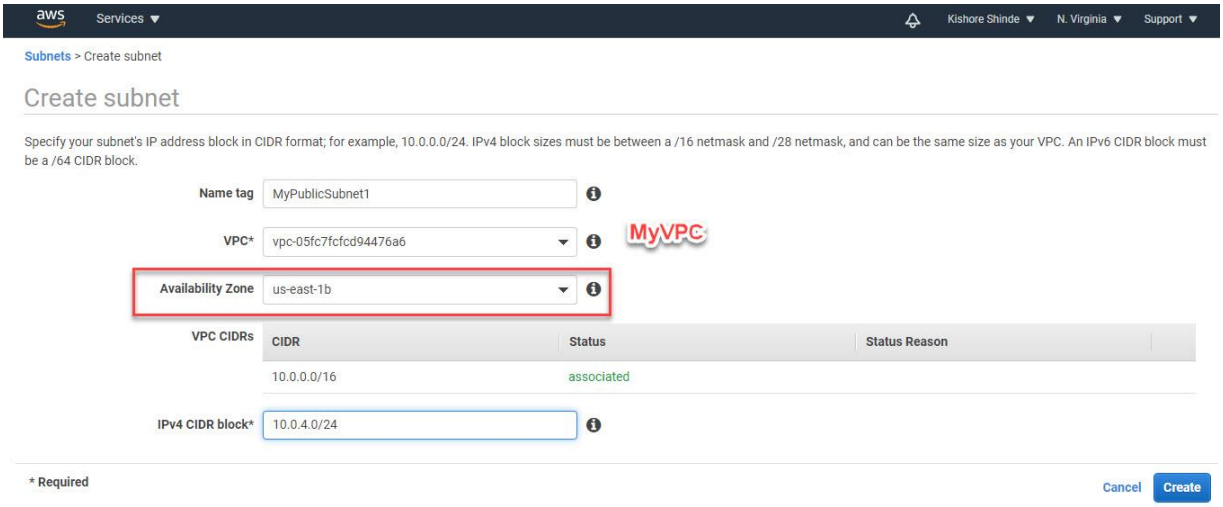


- Type "**exit**" to come out of the root user
- Again type "**exit**" to come out of the instance back to Bastion-Server

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.

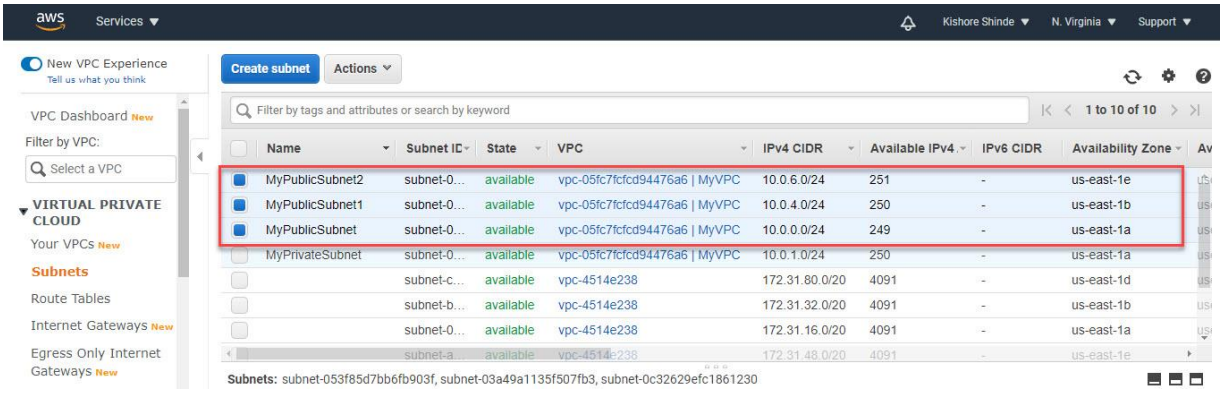
STEP 8 Creating additional Public Subnets, each subnet in different availability zone and in same VPC

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**



    - 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**

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



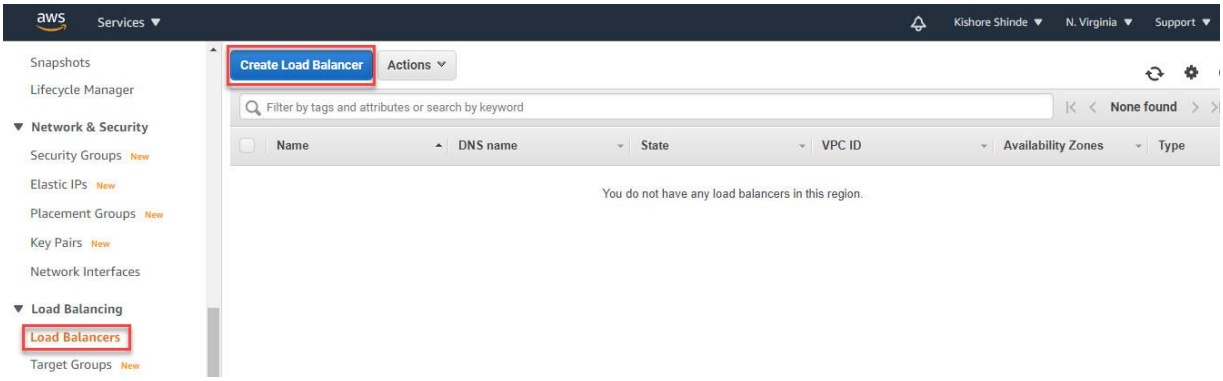
8. Next step is to create an **Application Load Balancer**.

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

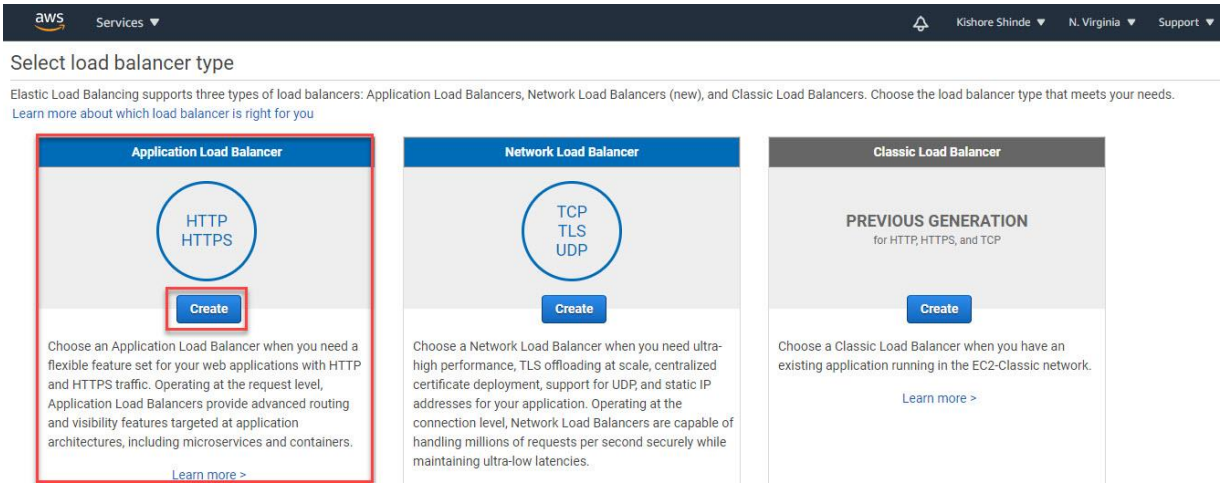
Now we will be creating new load balancer for our webservers.

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



2. Click on **Create Load Balancer.**

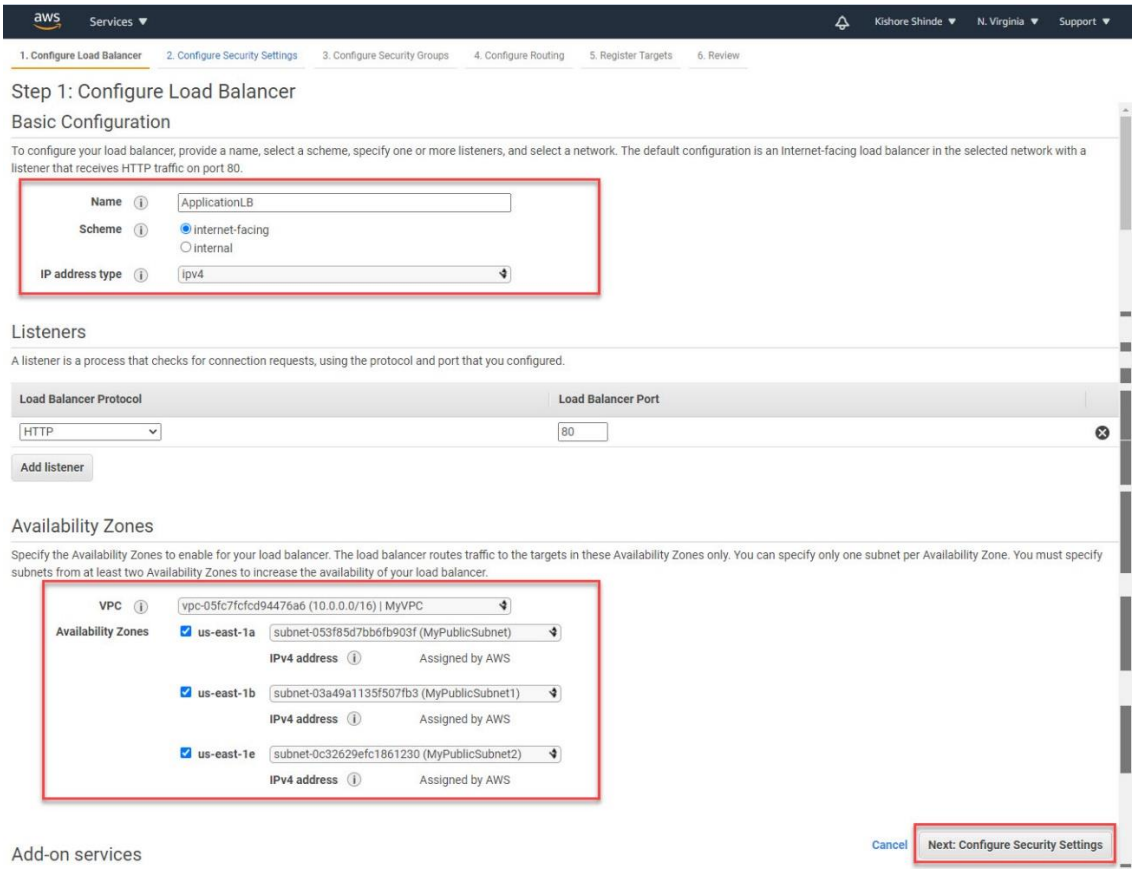
3. In **Select load balancer type**,



- 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

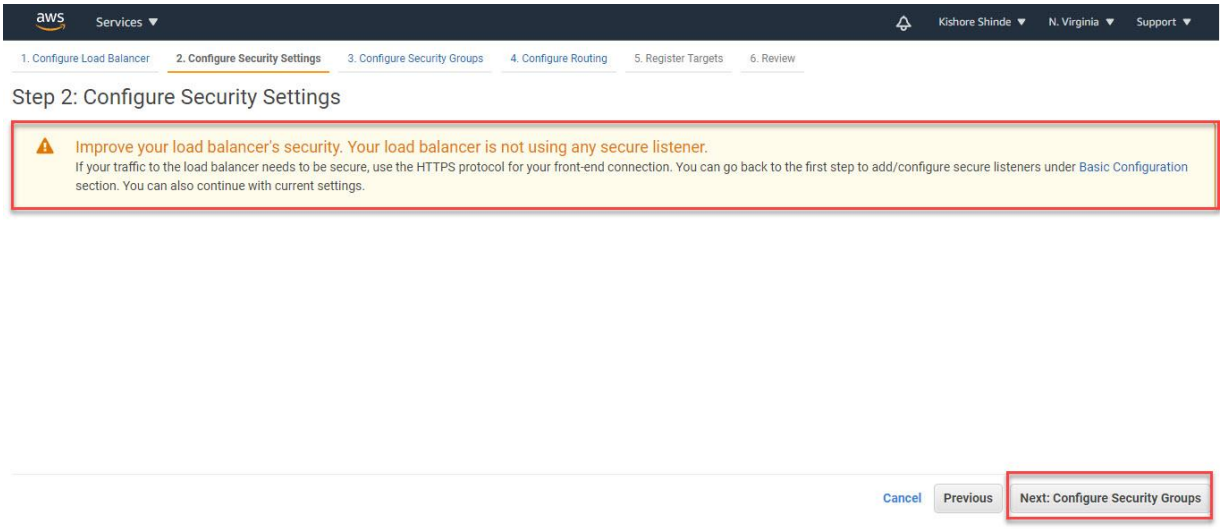


- Name : **ApplicationLB**
- Scheme : Select **internet-facing**

- 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****.*

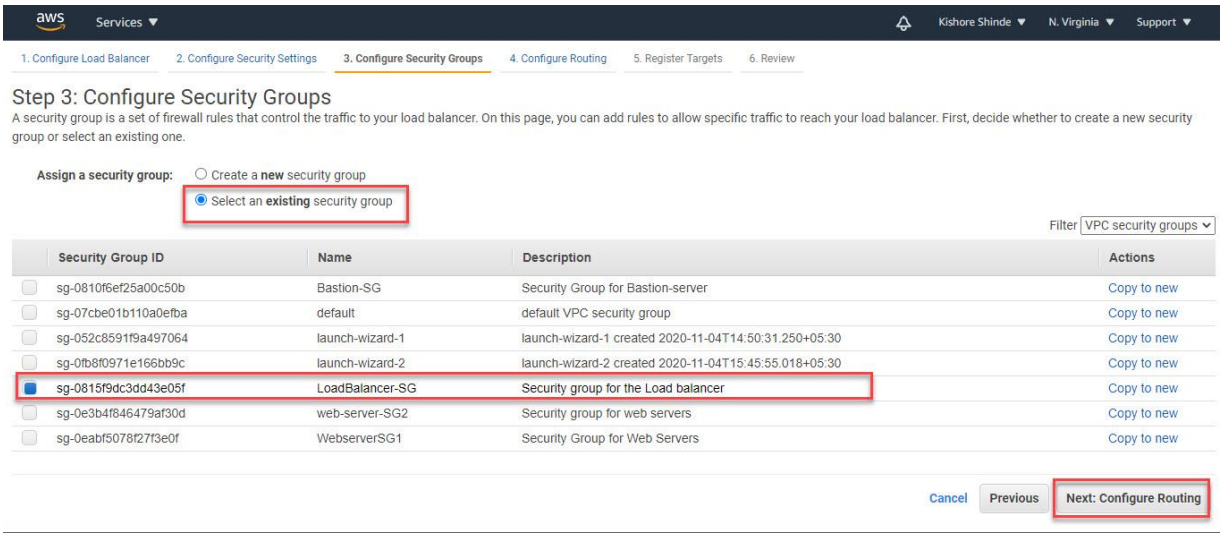
5. **Configure Security Settings** : Will see a message "*Improve your load balancer's security. Your load balancer is not using any secure listener*." Leave it as it is.



Click on **Next: Configure Security Groups**

6. **Configure Security Groups** :

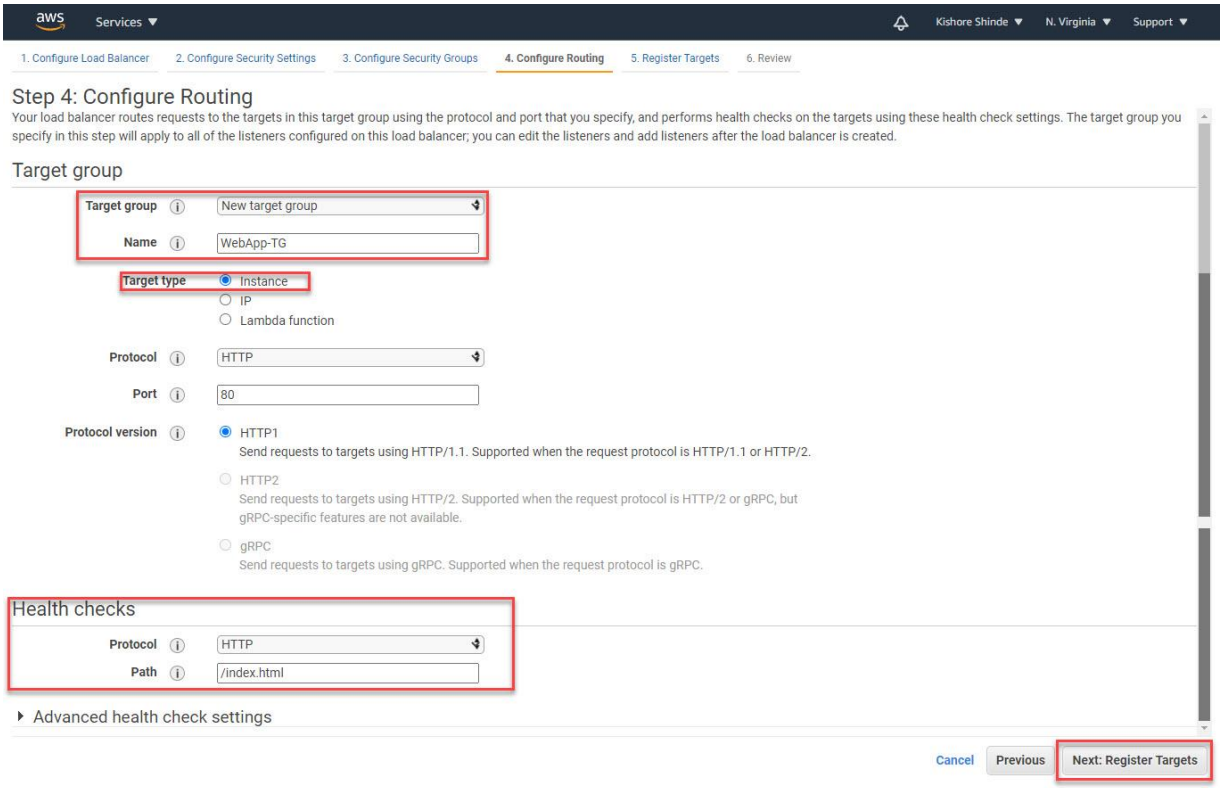
    - Assign a security group: select, **Select an existing security group**



    - 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**



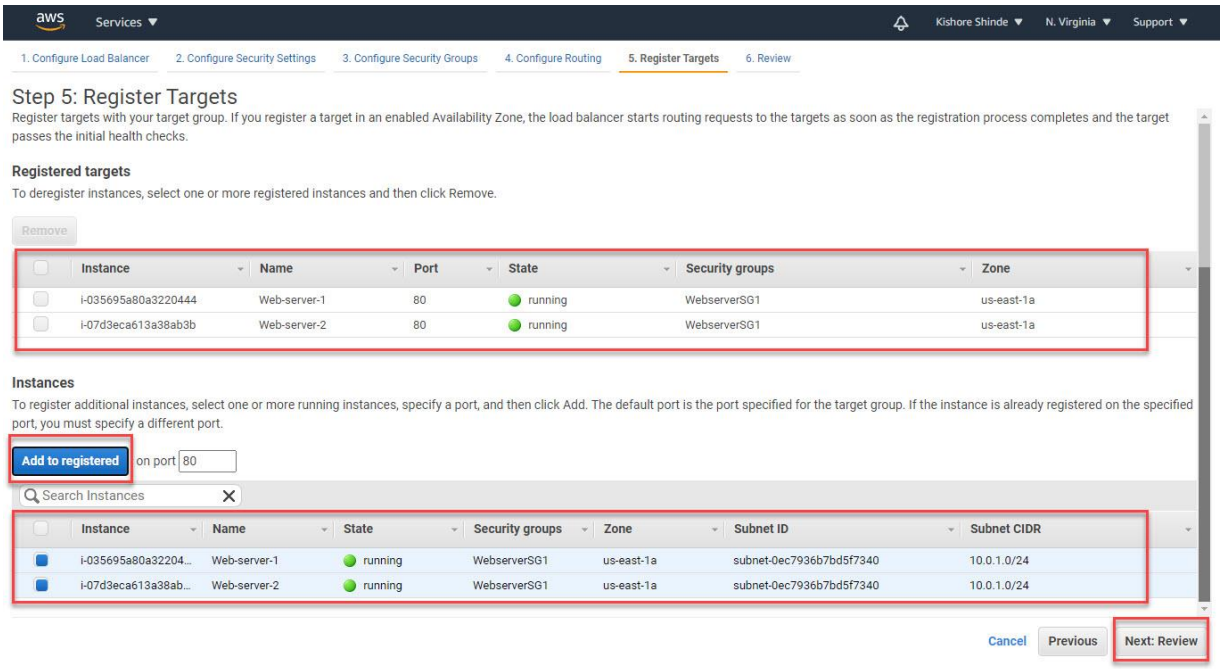
- Health checks :
  - Protocol : select **HTTP**
  - Path : type **/index.html**

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**

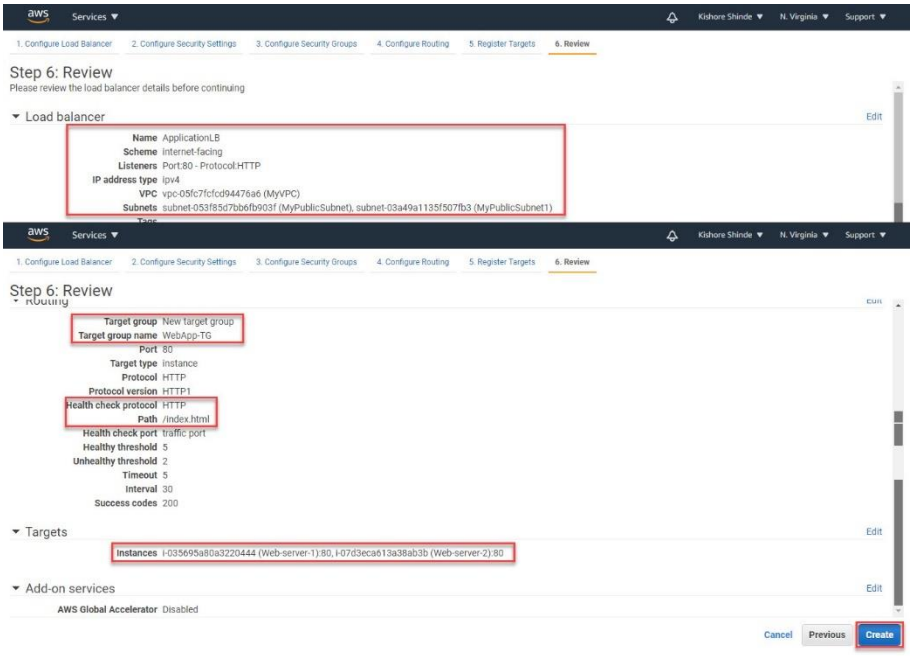
8. Register Targets

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



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

9. **Review :** Review all the settings

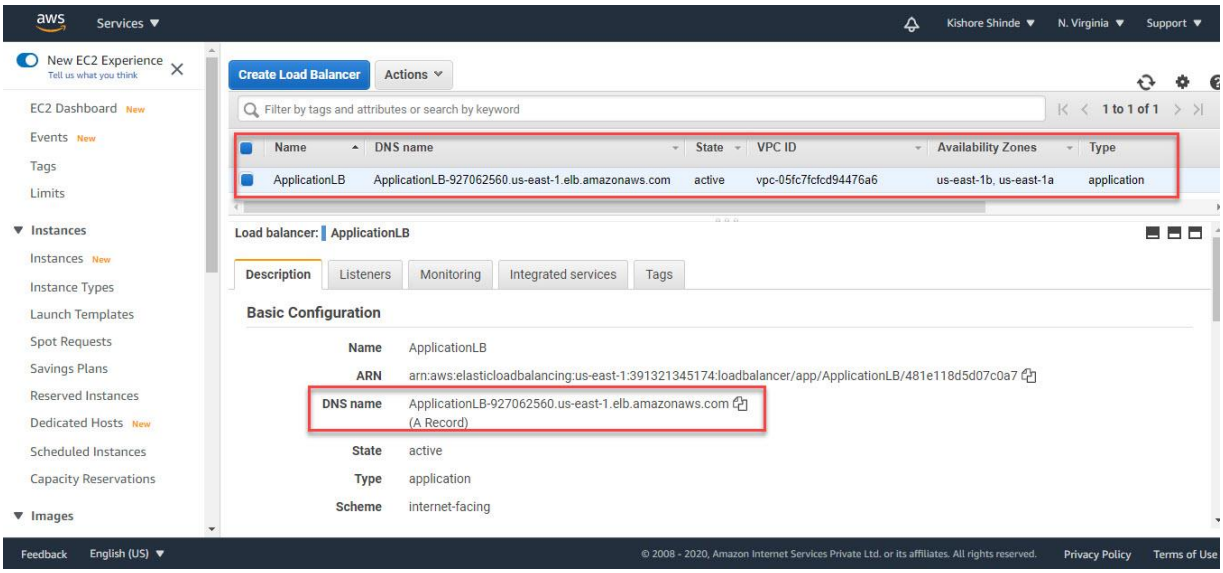


- Click on **Create**

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** :

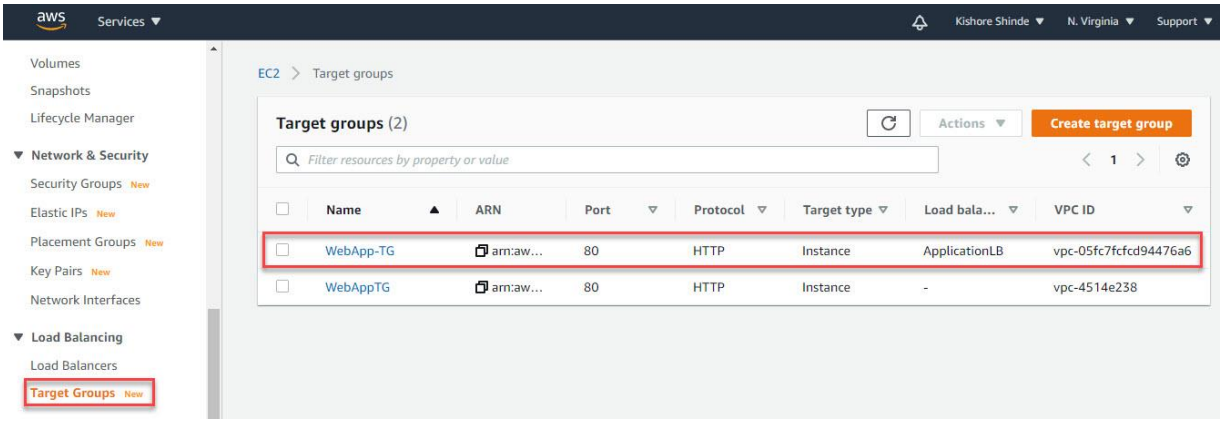


- Name : **ApplicationLB**
- DNS Name : **ApplicationLB-927062560.us-east1.elb.amazonaws.com**

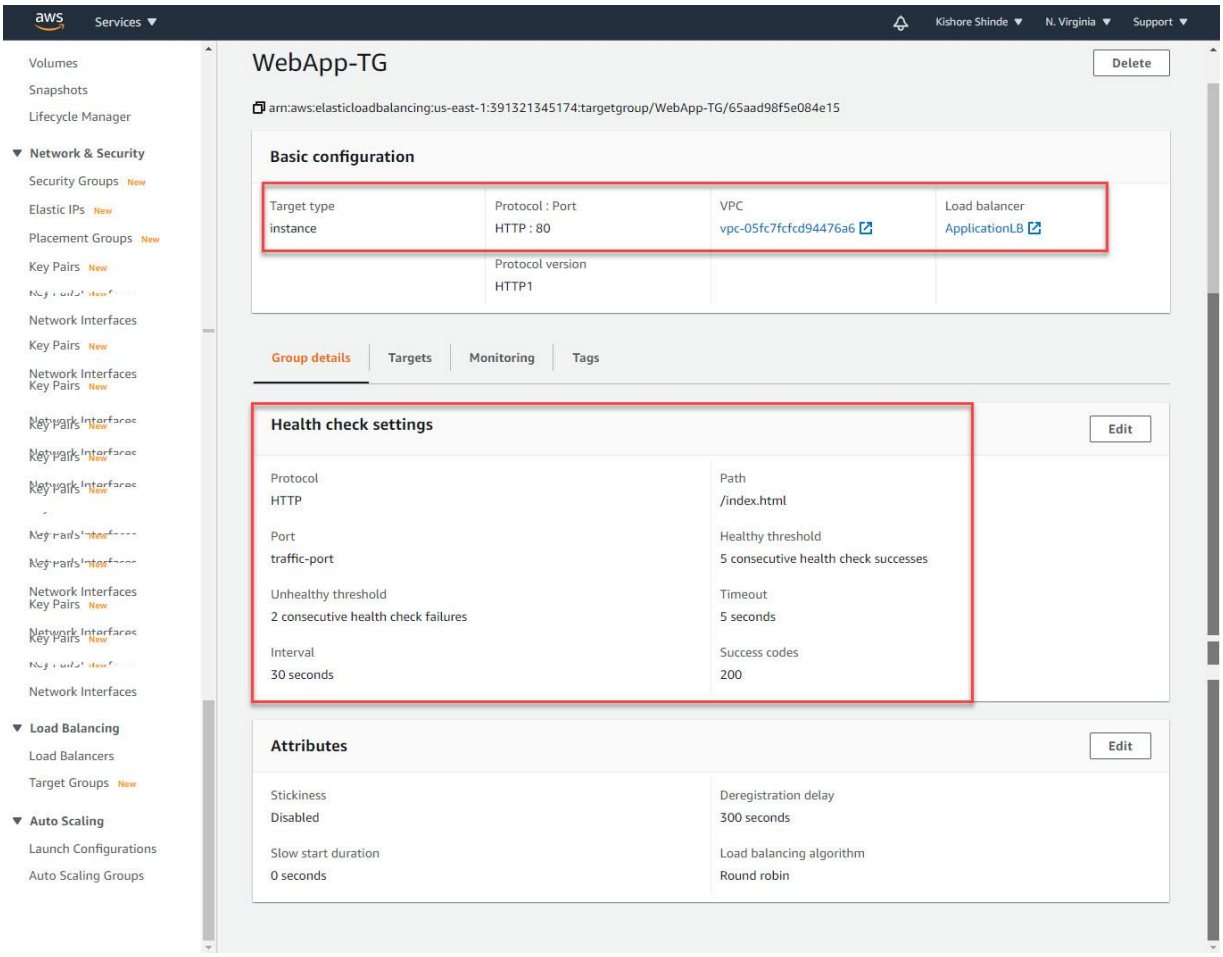
Step 10 Checking the health of Load Balancer & Testing DNS

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.**

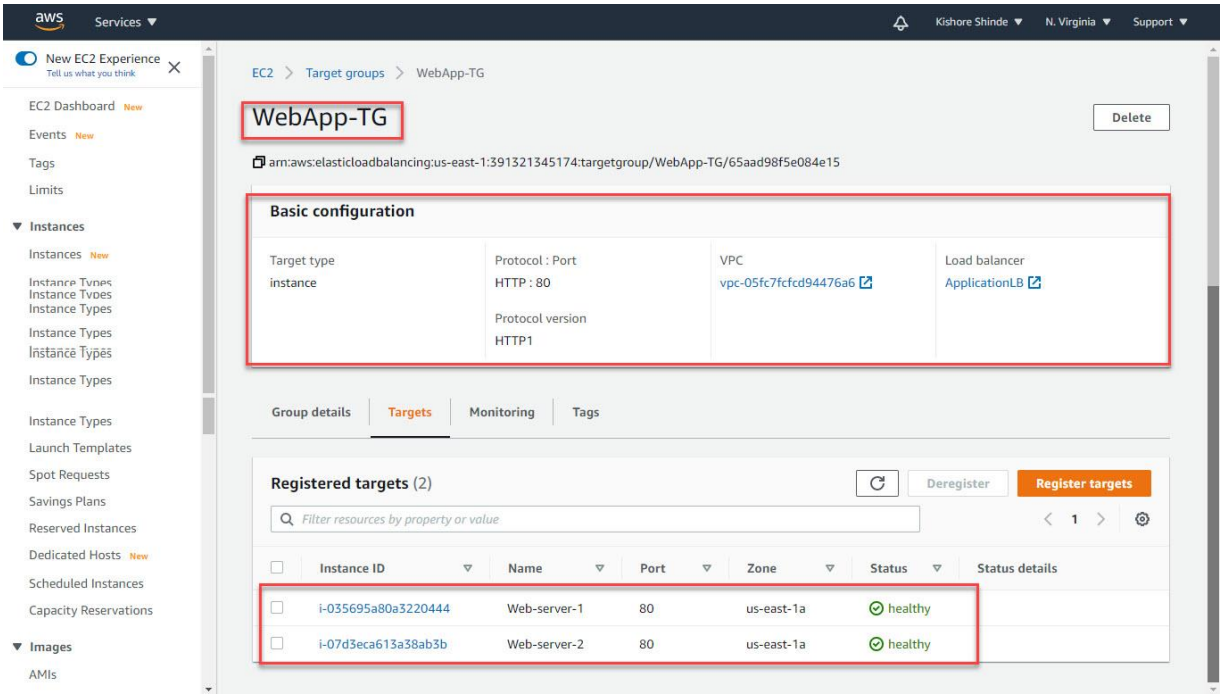


2. Click on the target group – **WebApp-TG**



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

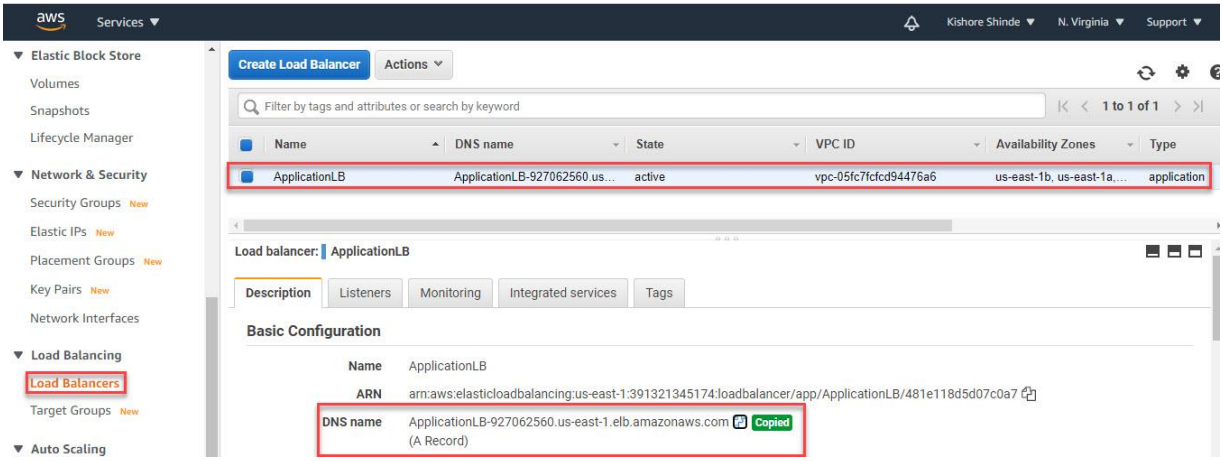
3. Now let us check the status of the attached targets. Click on **Targets** tab.



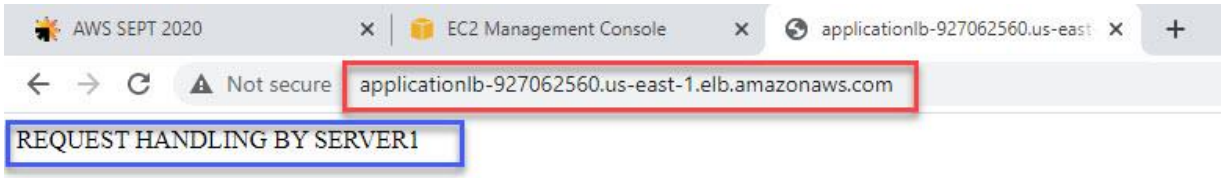
- Under the **Registered targets** you can see both the instances are **healthy.**

Task 2 : Testing the DNS

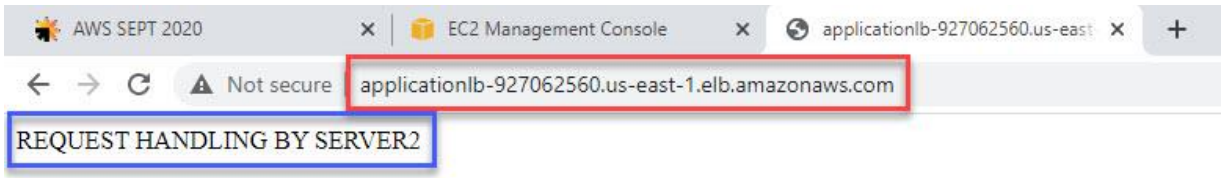
1. Click on **Load Balancers** under **Load Balancing.**



2. Select the Load Balancer – **ApplicationLB.**
3. Copy the DNS name : **ApplicationLB-927062560.us-east-1.elb.amazonaws.com** & paste it into the browser.
4. 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.



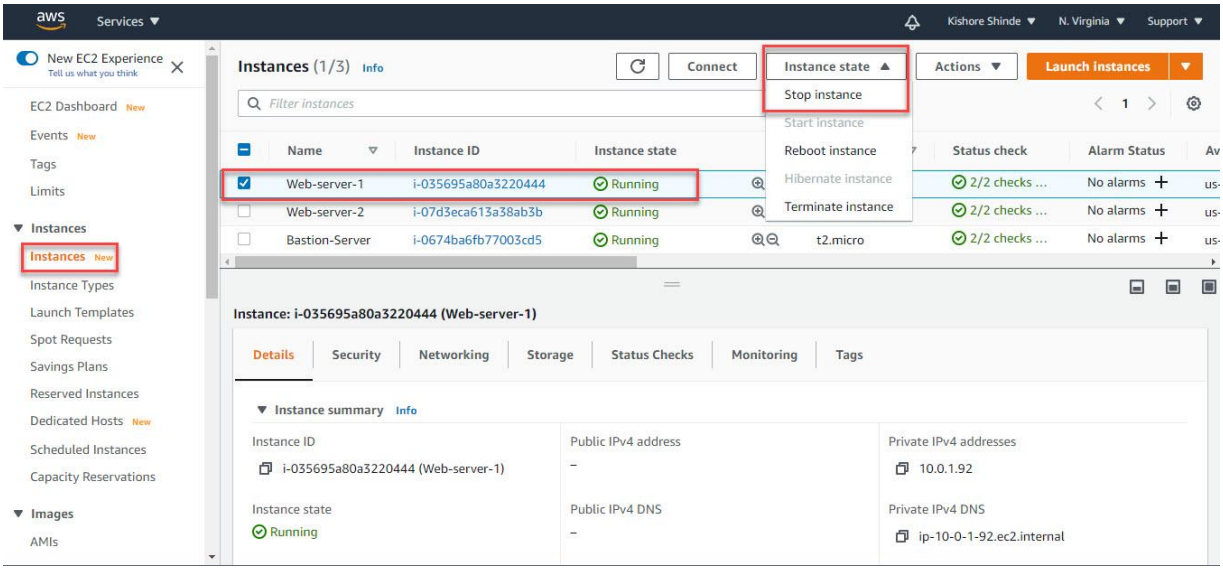
Once you see both the outputs **REQUEST HANDLING BY SERVER 1** & **REQUEST HANDLING BY SERVER 2** our DNS is tested and it also shows that the load is shared between the two web servers via Application Load Balancer.

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 it's status would be one of the following:

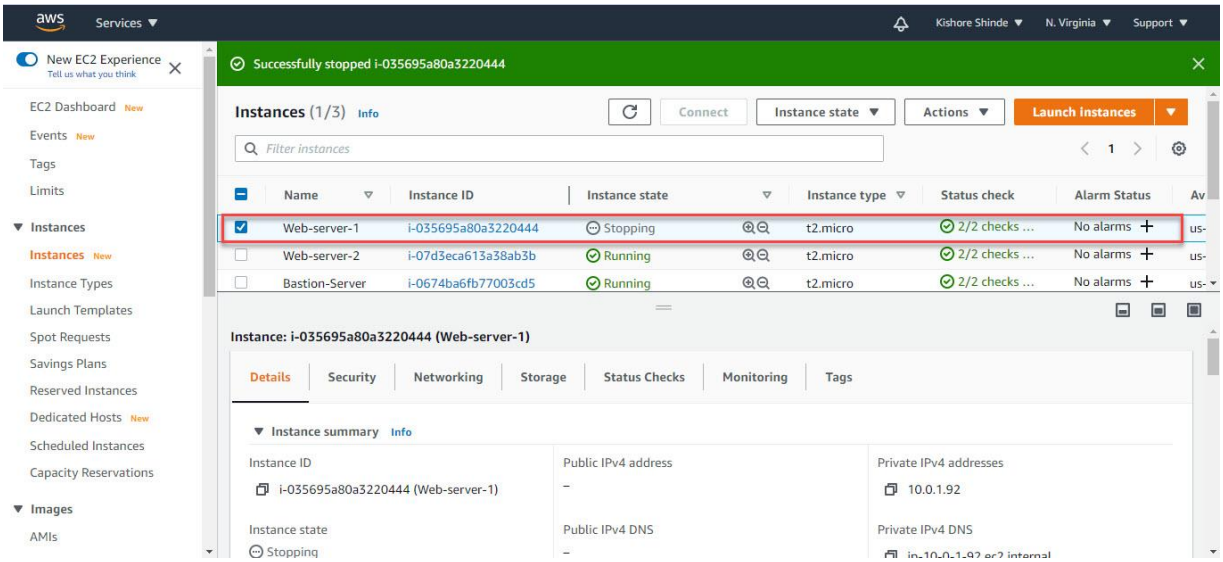
- Stopping
- Stopped
- Terminating
- Terminated

3. Navigate to the **EC2 Dashboard** and select **Web-server-1**. Click on **Instance State** & click on **Stop instance**.

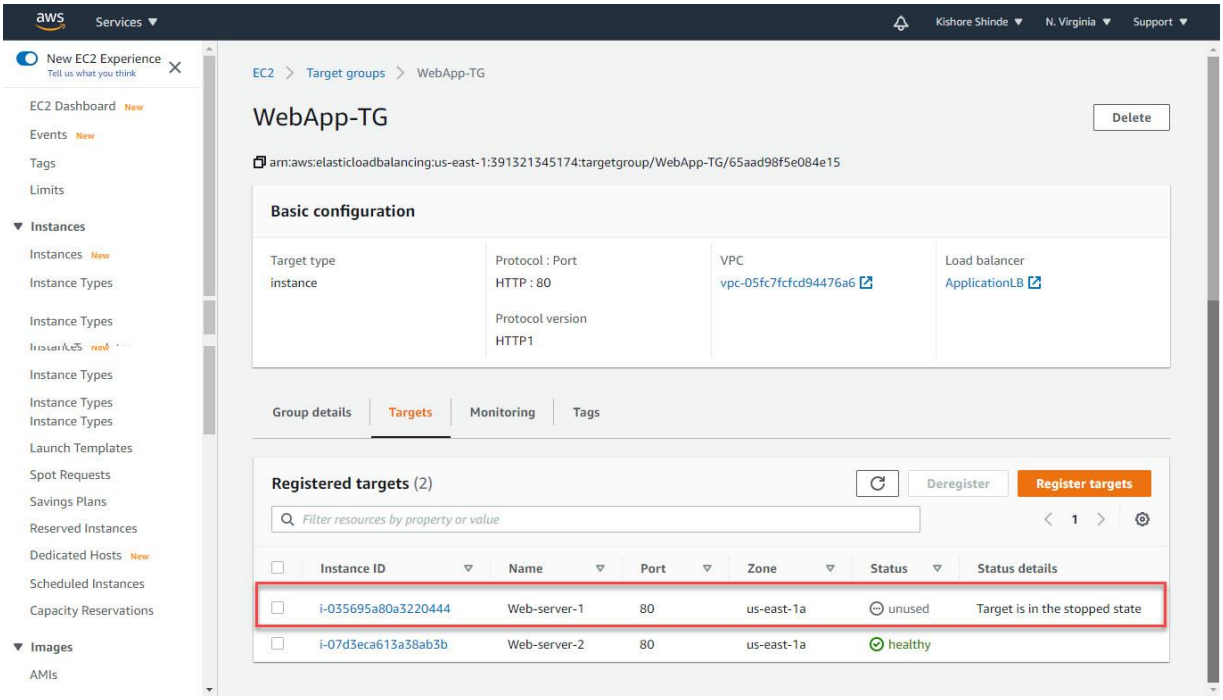


- Click on **Stop** for confirmation.

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

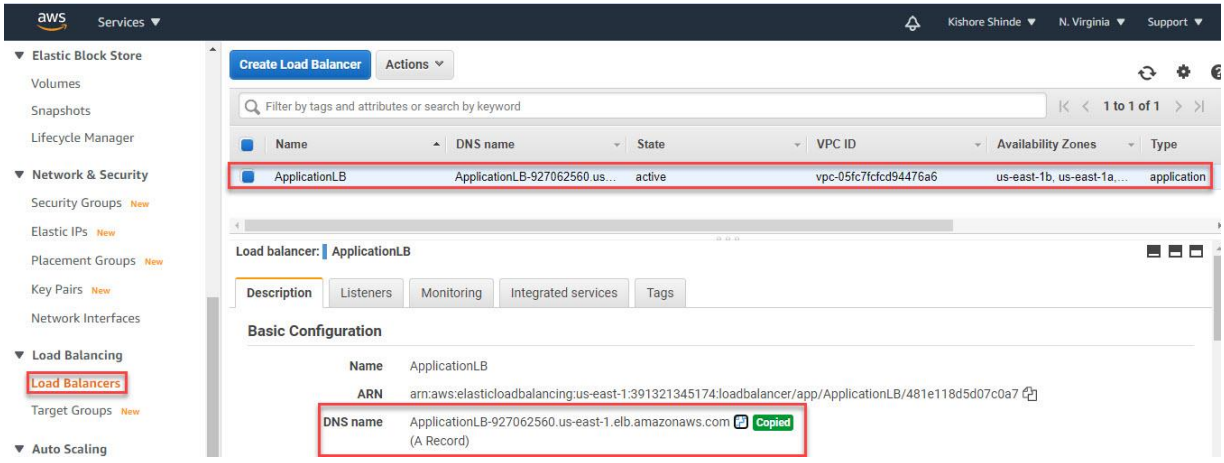


3. Now we will check the status of **Targets**. Navigate to the **Load Balancing** on the left panel, and select **Target Groups** under it.

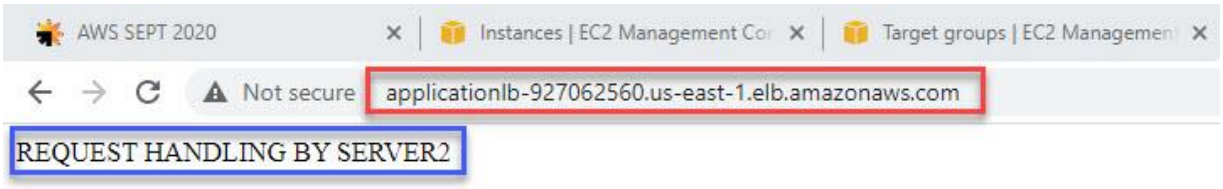


4. Click on **Targets** tab you can see **Web-server-1** status is changed from **healthy** to **unused** and in **Status details** it shows "**Target is in stopped state**".

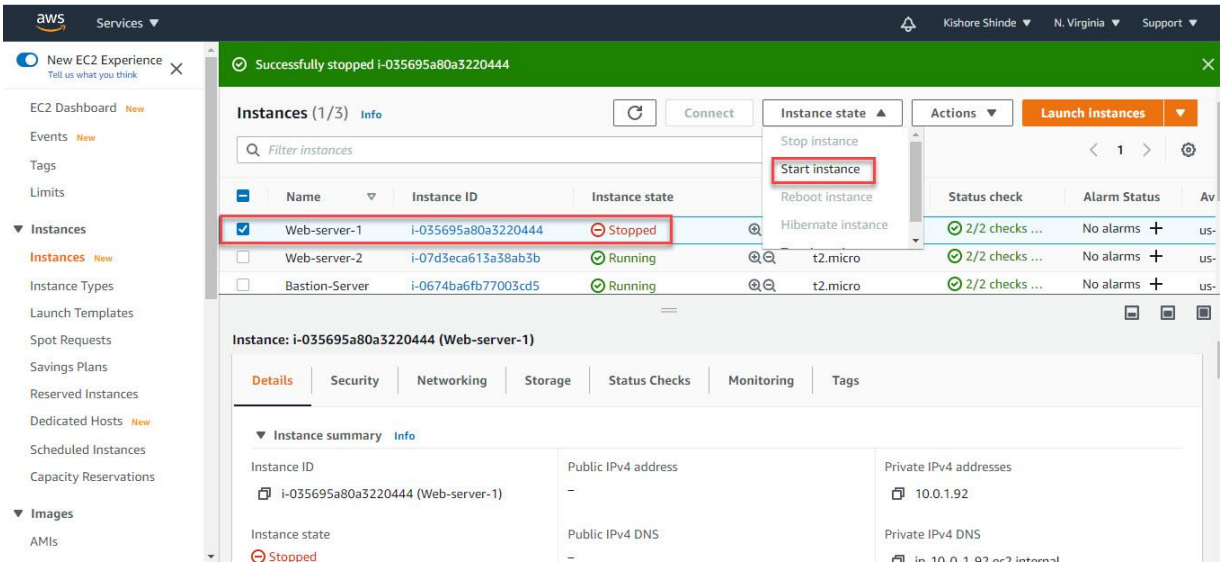
5. Navigate to **Load Balancers**. Select **ApplicationLB, under Description** tab below. Copy the **DNS name.**



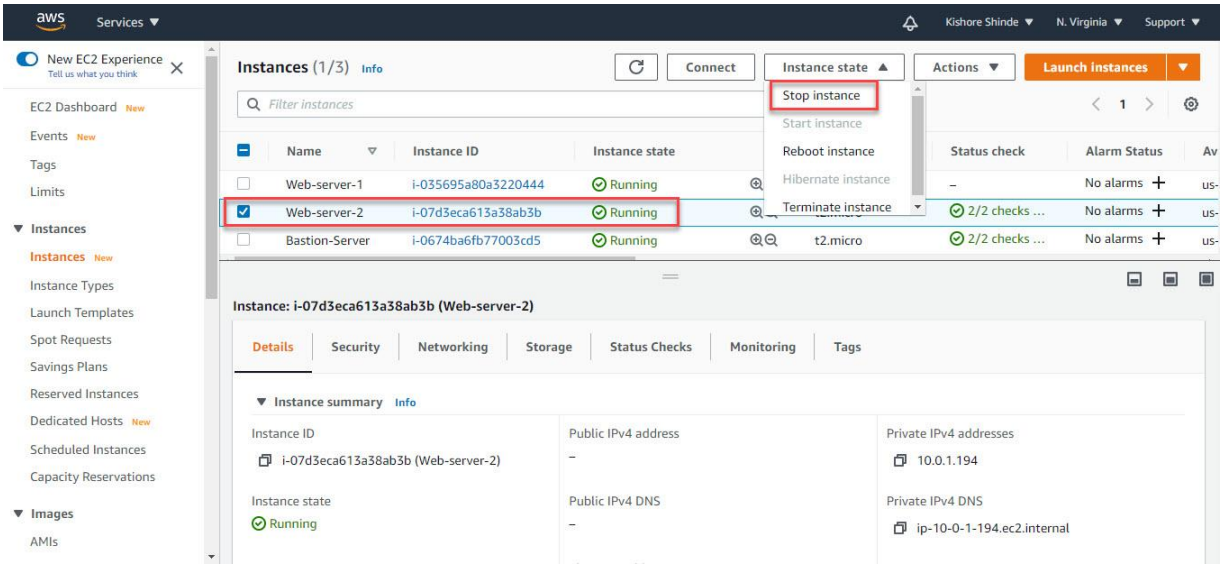
Paste it into your browser. You should see the response "**REQUEST HANDLING BY SERVER 2**". This is the response from Web-server-2.



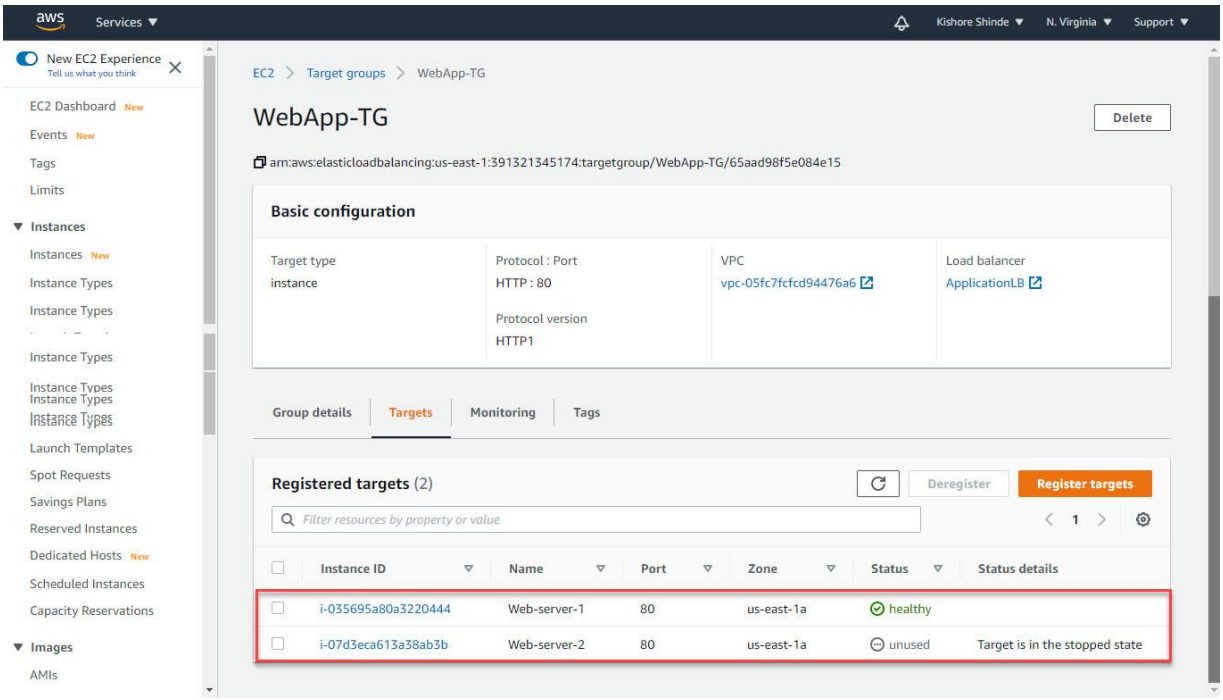
6. Even if you refresh a few times, you will continue to see the response only from Web-server-2
7. Now let us stop the **Web-server-2** and check.
8. Before that let us restart **Web-server-1**



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

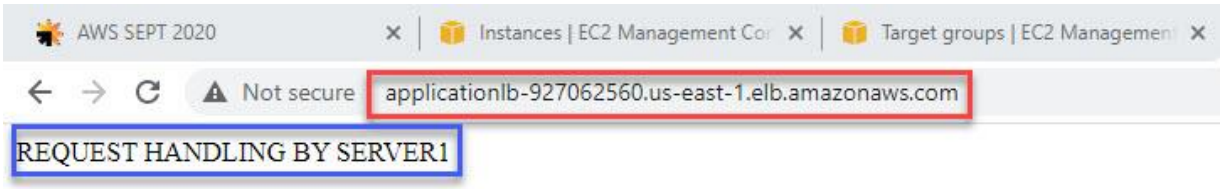


10. Now let us will check the status of **Targets**. Navigate to the **Load Balancing** on the left panel, and select **Target Groups** under it. Select the target group **WebApp-TG** and go to **Targets** Tab.

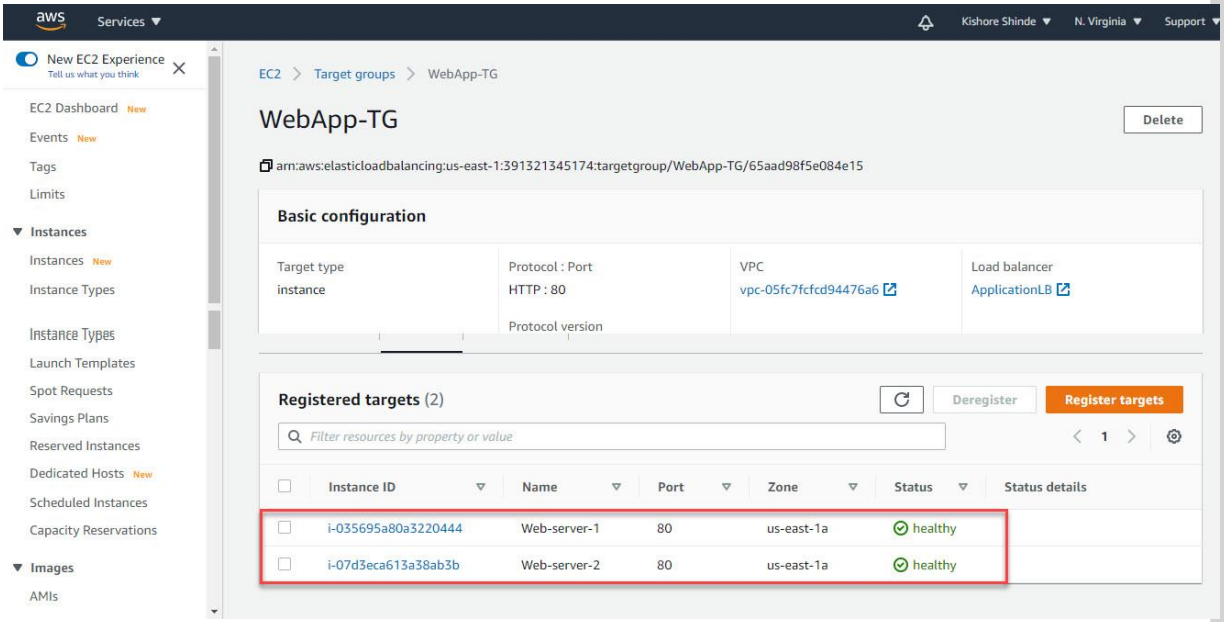


- You can see the status of **Web-server-2** which we stopped is **unused** and **Web-server-1** is **healthy**

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

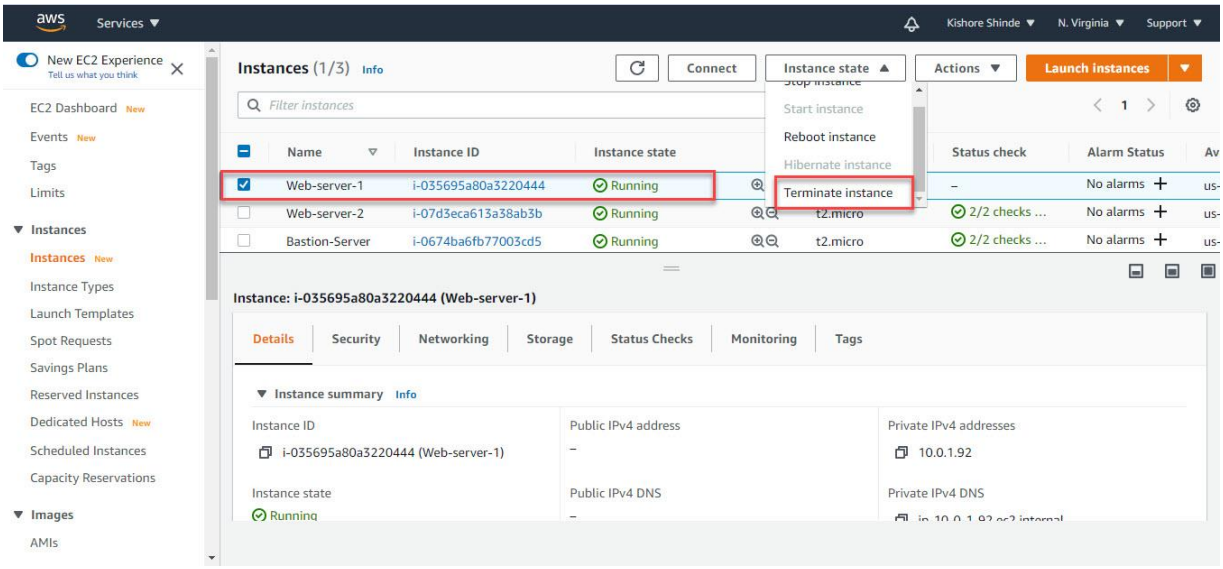


12. Now when we restart the Web-server-2 and check the health of instances or web servers it will show both **healthy**.



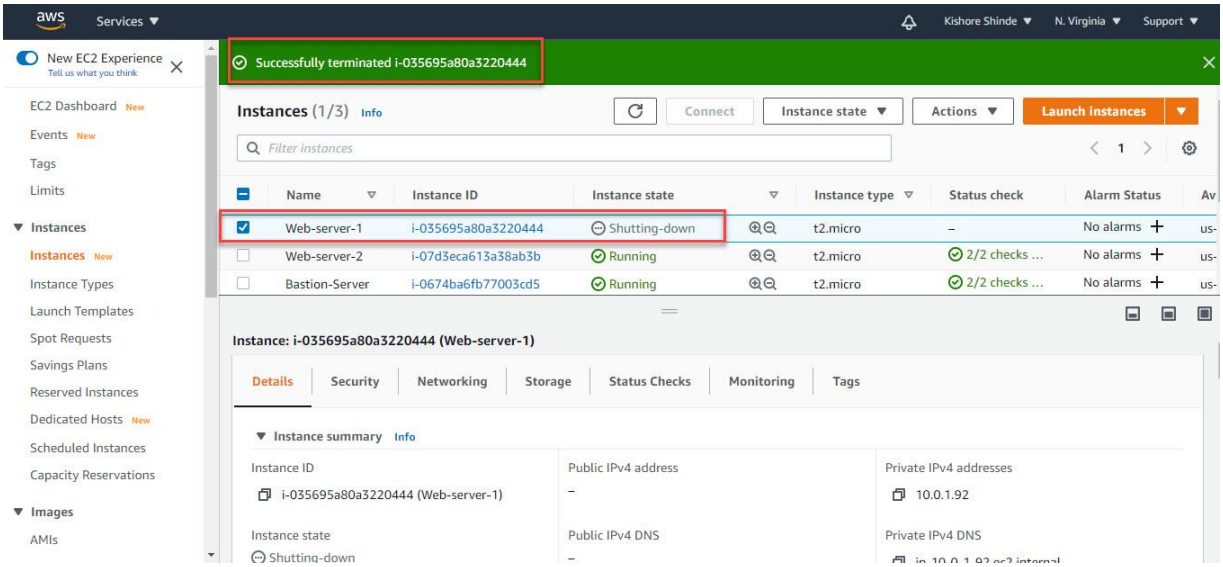
13. Now let us **Terminate** one web server or instance and check.

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

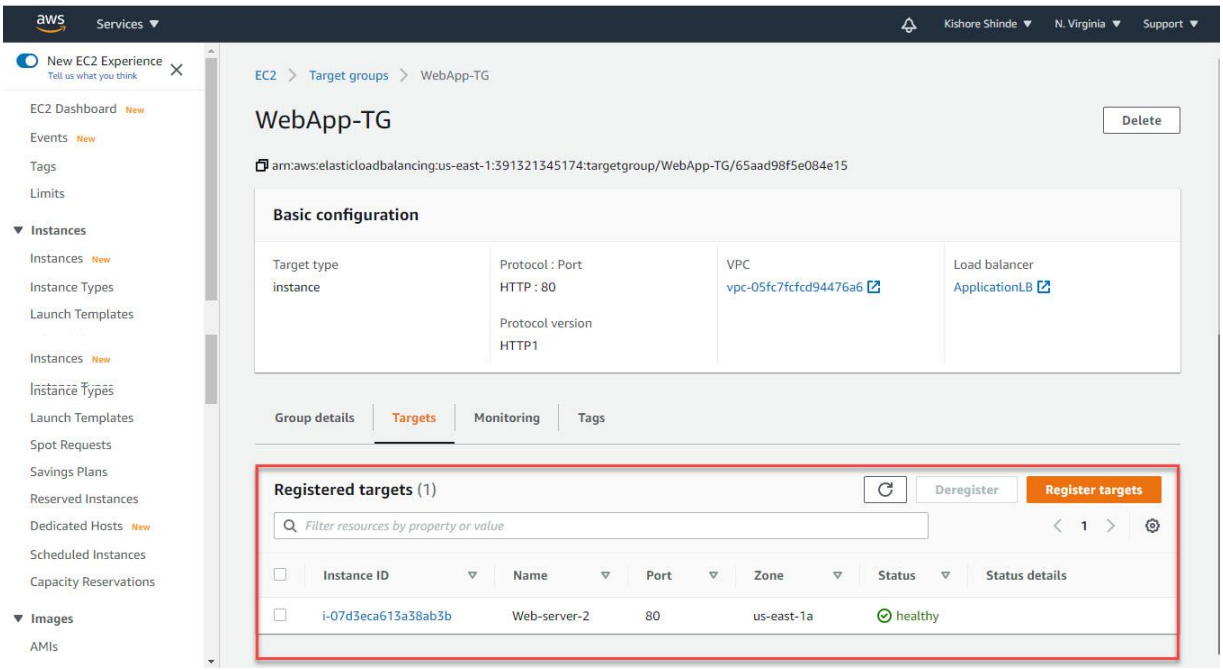


15.Just confirm the Web-server-1 **termination**.

- Click on **Terminate**.

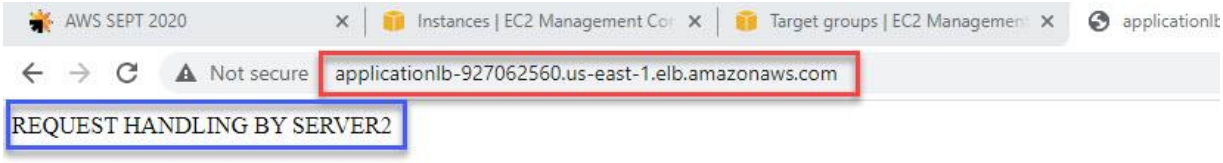


16. Once the **Web-server-1** is terminated let us once again check the health of the instances.



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

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



- 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