

Advance AWS

AWS Project- 4

Student:

Kishore Shinde

Teacher:

Mrs. Vinolin Jeremiah

Course:

Advance AWS Cloud Computing with DevOps
Fundamentals

Institute:

Lets Upgrade

Project 04:

Working with Elastic IP's

We will be doing it in 6 steps:

STEP A: Creating a Linux Instance

STEP B: Connecting a Linux Instance

STEP C: Installing Apache Server

STEP D: Creating Elastic IP

STEP E: Associating Elastic IP

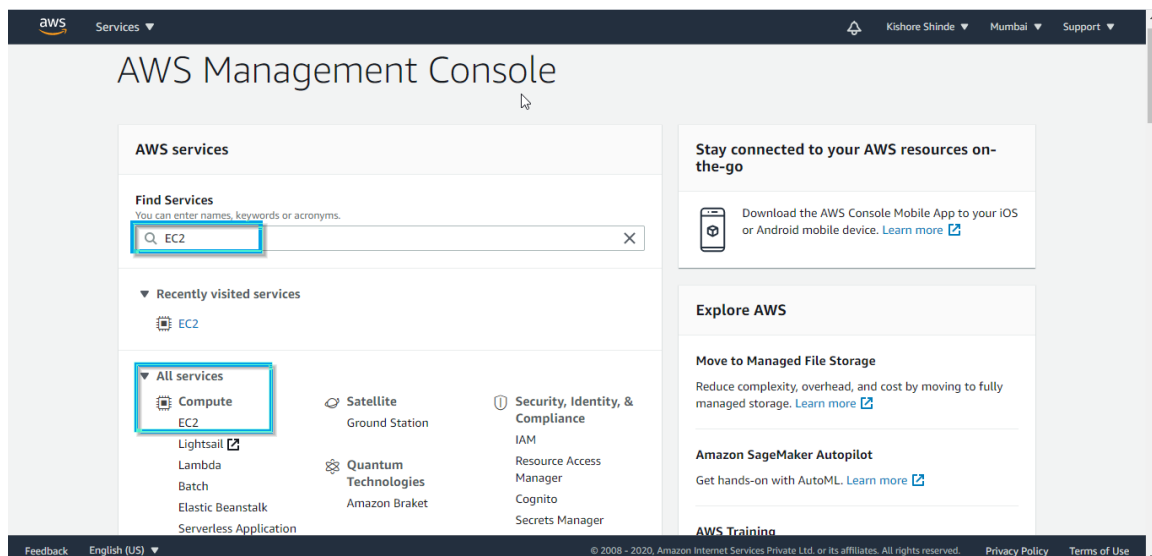
STEP F: Terminating a Linux Instance

STEP G: Releasing/Deleting Elastic IP

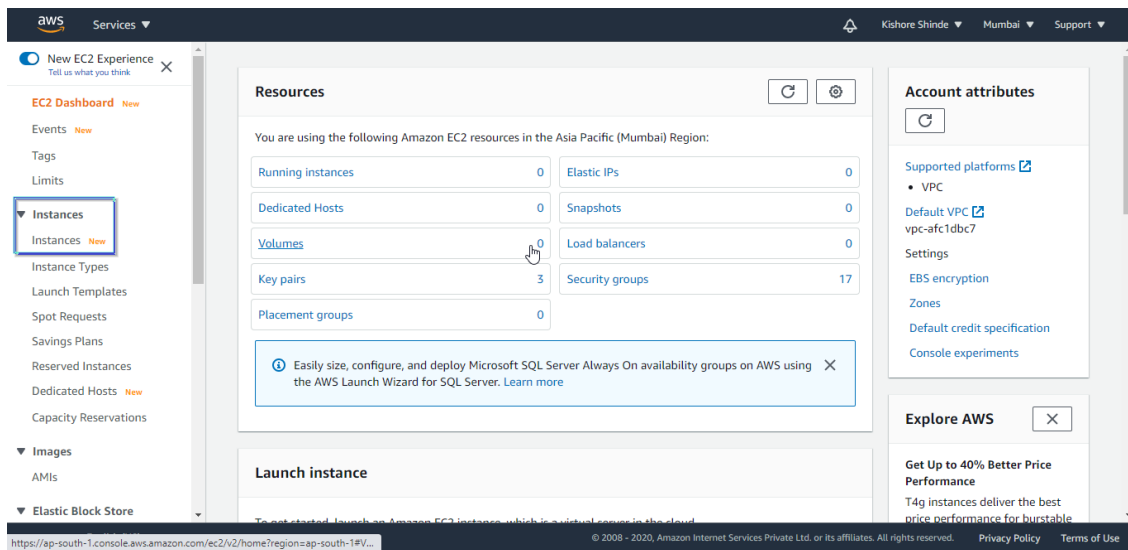
STEP A: Creating a Linux Instance

Steps for launching a new Linux Instance:

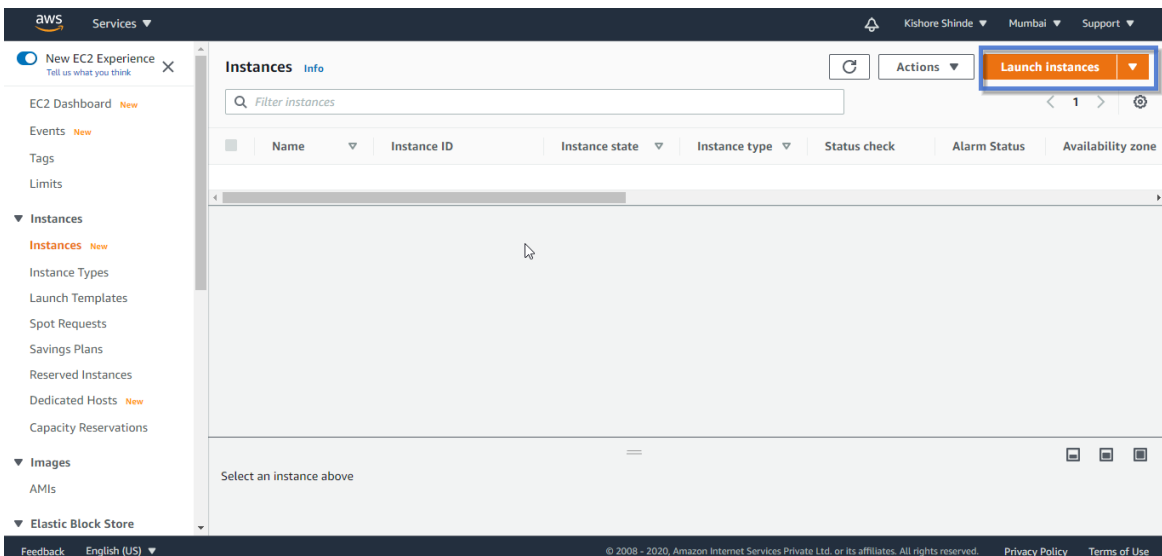
1. From the AWS Management Console, you can either find EC2 service or click on All services ->Compute->EC2



2. Once you are in EC2 console from the EC2 Dashboard on the left, select Instances

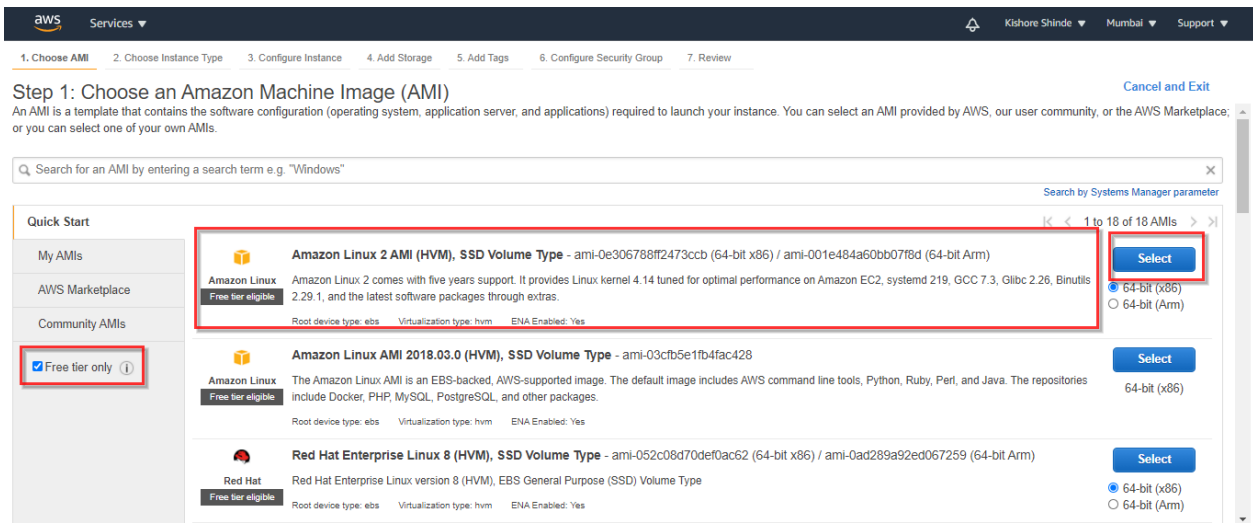


3. From the Instances Dashboard, Select Launch Instance at the right



Once you click on Launch Instance, a wizard will start which has 7 steps for creating the instance.

Step1: Choose an Amazon Machine Image (AMI)



You can search the AMI e.g. Linux.... or can select the AMI from the list of the AMI's.

Note: Make sure you select the Free Tier only option so only free AMI's will be shown and you will not be charged.

Click on "Select" on the Amazon Linux 2 AMI e.g. Amazon Linux 2 AMI (HVM), SSD....

Step 2: Choose an Instance Type

Here you can select the Instance type. These are varying combinations of CPU, Memory, Storage and Networking capacity. The default instance type selected is "t2.micro" which is Free tier eligible. Let it be selected otherwise you will be charged for other instance type. You can even see the details of the selected instance type in **Currently Selected** e.g. : t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

aws Services

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

Step 2: Choose an Instance Type

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

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

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

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

Cancel Previous Review and Launch Next: Configure Instance Details

Step 3: Configure Instance Details

Here you can configure the instance that suits your requirement. You can launch multiple instances from the same AMI you can mention it in **Number of instances**.

In **Network** you can select the VPC or create new VPC, we will continue with default VPC, select or create new Subnet, we will continue with default subnet. We can select IAM role, we will continue with "None".

aws Services

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

Step 3: Configure Instance Details

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

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: ☐ Request Spot instances

Network: vpc-afc1dbc7 (default) Create new VPC

Subnet: No preference (default subnet in any Availability Zone) Create new subnet

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

Placement group: ☐ Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory Create new directory

IAM role: None Create new IAM role

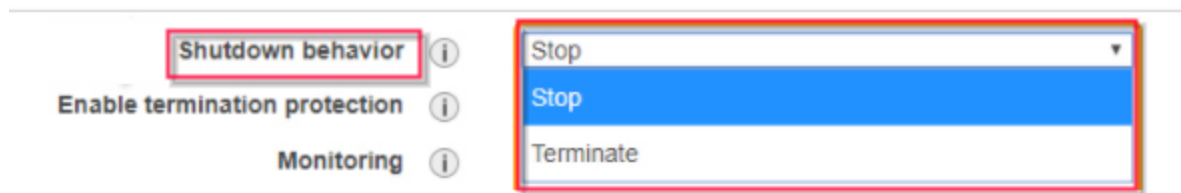
Shutdown behavior: Stop

Stop - Hibernate behavior: ☐ Enable hibernation as an additional stop behavior

Enable termination protection: ☐ Protect against accidental termination

Cancel Previous Review and Launch Next: Add Storage

In the Shutdown behavior you can select Stop or Terminate. It is an important option If you select Stop when the instance shuts down it will not be deleted but stopped.



Enable termination protection: If you check it will protect your instance from accidental termination.



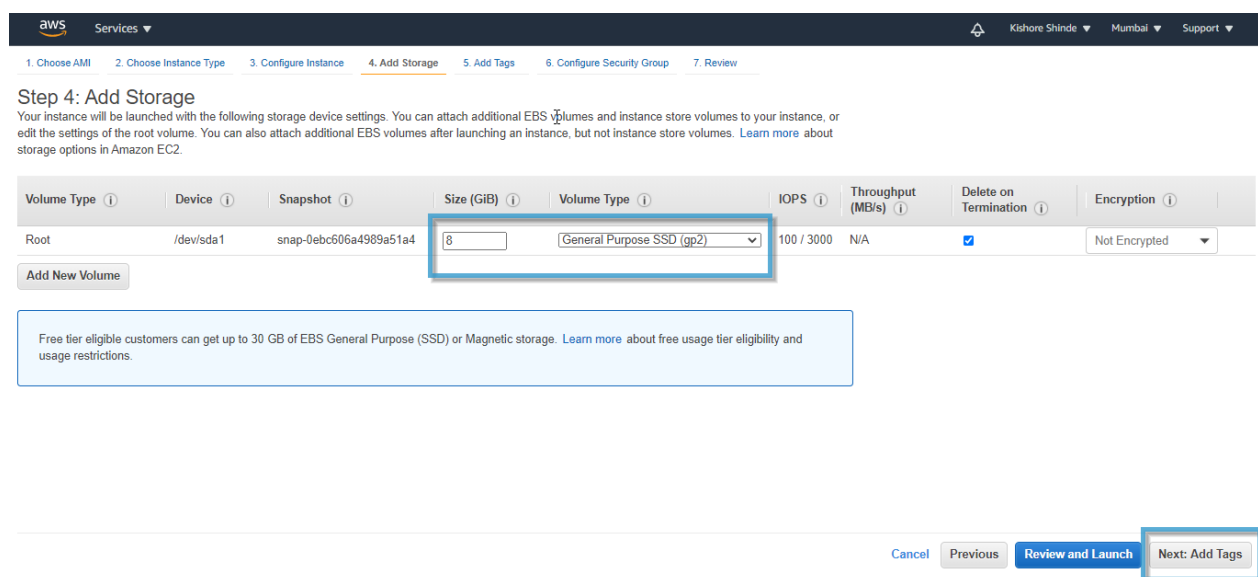
Now keep whatever is default don't change anything and click on Next: Add Storage

Step 4: Add Storage

Your Instance will be launched with the following storage device settings. You can attach additional EBS volumes. For now, keep the default Volume Type: Root and size: 8 GiB

Note: The default root partition in AWS for windows is 30 GB and for Linux is 8 GB you can change it.

Let the General-Purpose SSD (gp2) selection as it is. Delete on Termination checkbox will make sure the volume gets deleted as soon as the Instance is terminated. You can even add new EBS volume.



Click on Next: Add Tags

Step 5: Add Tags

Tags enable you to categorize your AWS resources in different ways. Each tag is a simple label consisting of customer-defined key and an optional value that can make it easier to manage, search for, and filter resources. For E.g. Key can be Name and Value can be Web Server (Linux-Apache).

aws Services

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)
Name	Linux -Apache

[Add another tag](#) (Up to 50 tags maximum)

Instances Volumes

Cancel Previous Review and Launch Next: Configure Security Group

You can add the tag or can continue to next step “Configure Security Group”

Step 6: Configure Security Group

A security group is similar to firewall. Here you can set the rules that can control traffic for your instance. For the current instance in the Type select “All Traffic” and in Source select “Anywhere”. It will show you a warning that the source anywhere will allow all IP addresses to access your instance you must select IP Addresses only. Ignore it for now and click on “Review and Launch”.

aws Services

Kishore Shinde Mumbai 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

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
All traffic	All	0 - 65535	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

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.

Cancel Previous **Review and Launch**

Step 7: Review

Here you can review all the selection you have done in previous steps and if required can go back and change them. You are able to review the AMI details, Instance Type/Details, Security Groups, Storage & Tags.

aws Services

Kishore Shinde Mumbai Support

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

Step 7: Review Instance Launch

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e306788ff2473ccb

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 2.28, Binutils 2.29.1, and the latest software packages through extras.

Root Device Type: ebs Virtualization type: hvm

Instance Type

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

Security Groups

Security group name: launch-wizard-20
 Description: launch-wizard-20 created 2020-10-09T11:29:05.049+05:30

Type	Protocol	Port Range	Source	Description
All traffic	All	All	0.0.0.0/0	
All traffic	All	All	::/0	

Instance Details

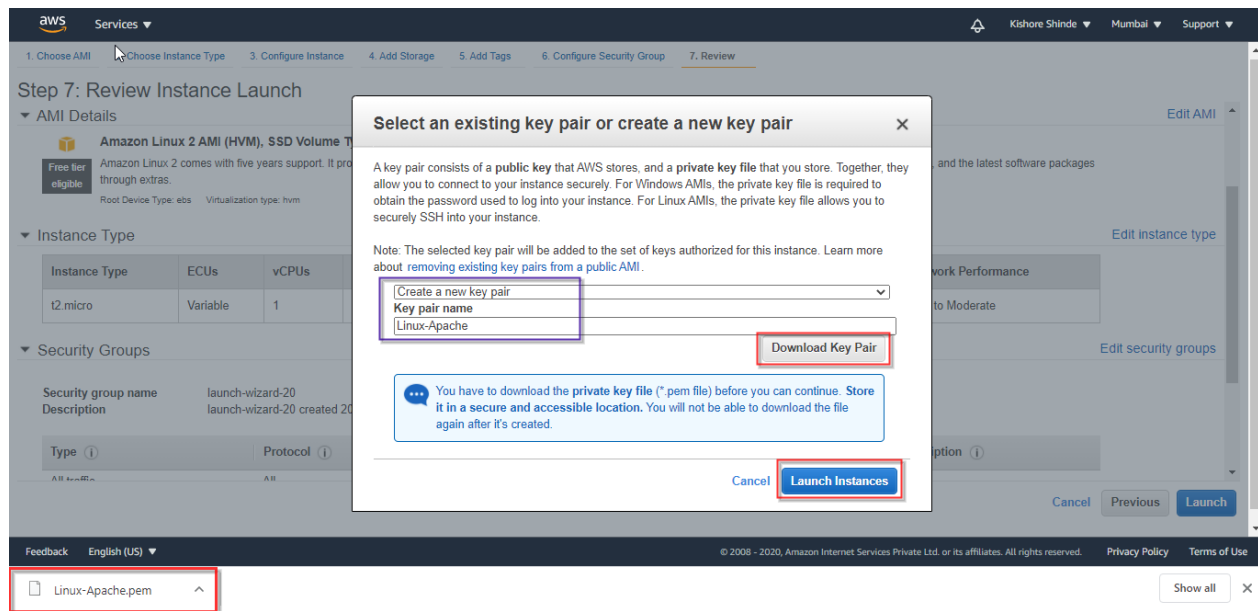
Storage

Tags

Cancel Previous **Launch**

Click on Launch once you have reviewed all the details.

Next it will ask you to Select an existing key pair or create a new key pair.

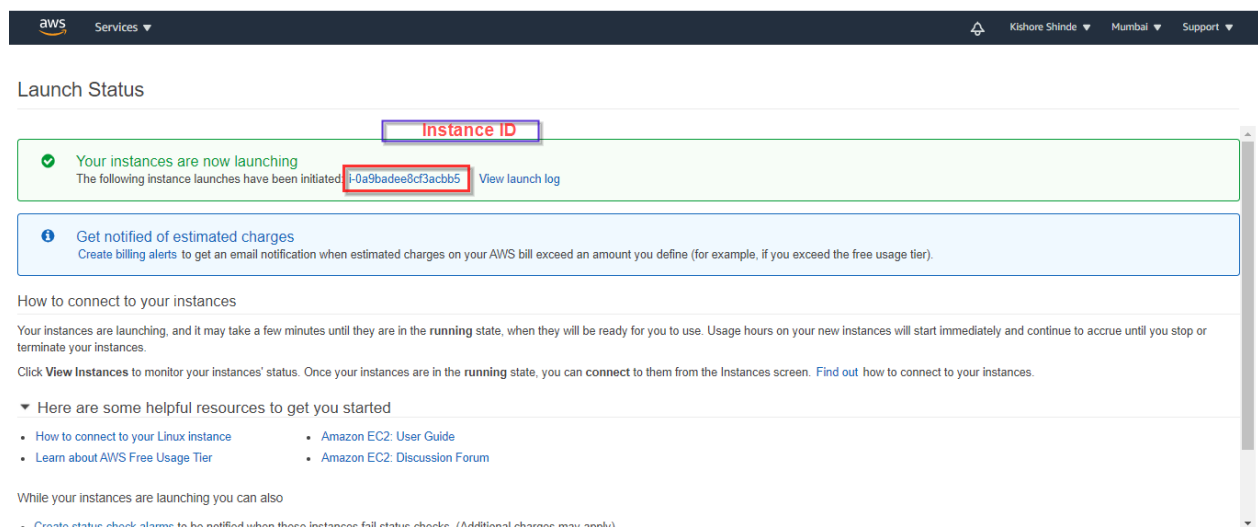


You can select existing key pair if you have one. For now, select “Create new pair” give key pair a name and download the keypair. We have given name to key pair as *Linux-Apache* and downloaded it *Linux-Apache.pem*.

Note: Please keep it safe, it will be required to connect to the instance otherwise you can't.

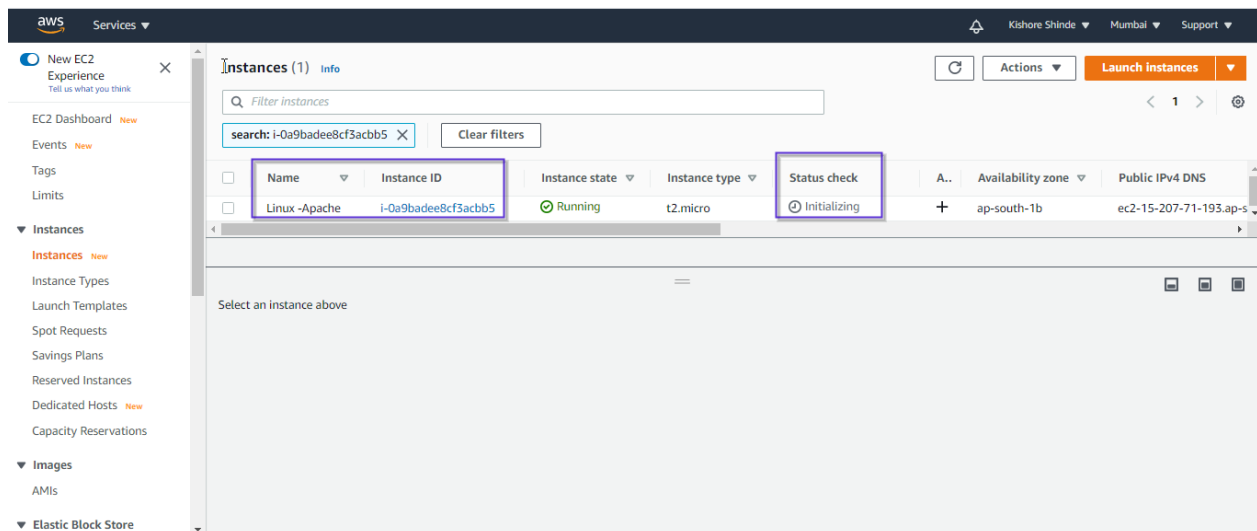
Click on Launch Instance.

In the next screen you will be able to see “Your instance is launching”. You will be able to see the instance id that is initiated for launch.

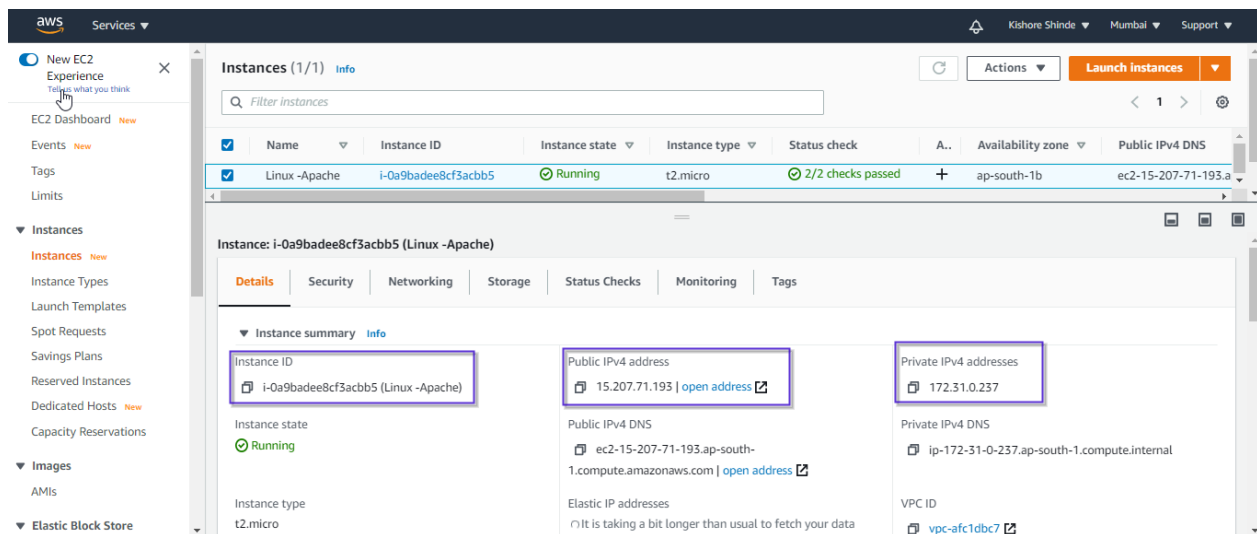


You can click on the instance id which will take you to Instances Dashboard.

Here you will see the Linux instance is created which will be initially showing Instance State as “Pending”. Wait till the Status check shows 2/2 checks passed now it is Initializing and Instance State becomes “Running”.



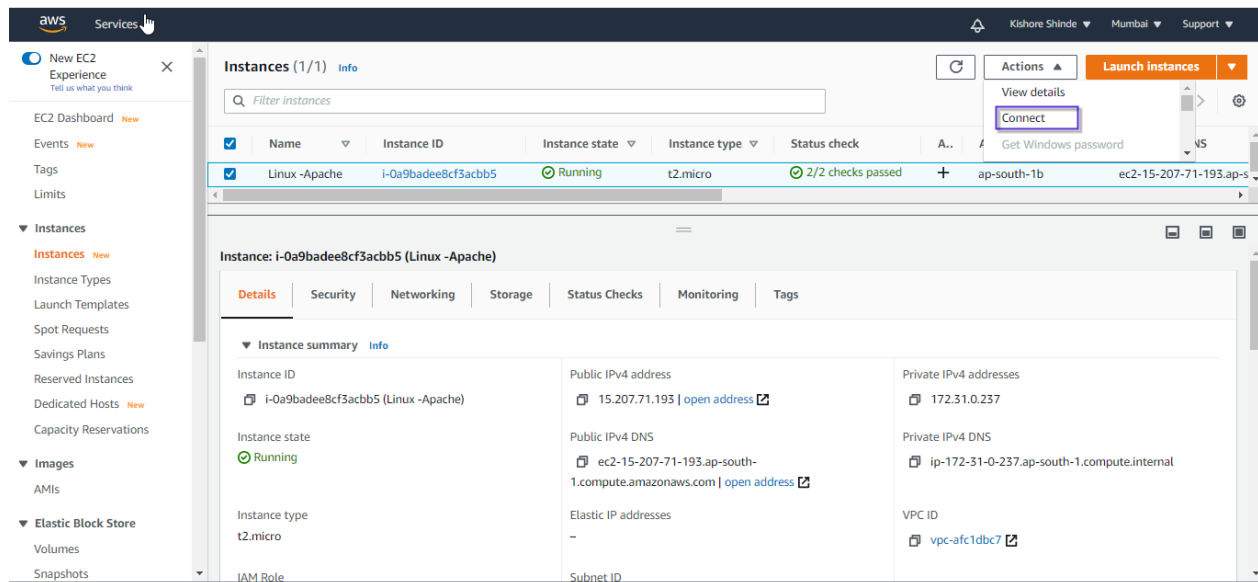
Now select the instance you will be able to see the additional details of the instance like Public IPv4 address, Private IPv4 address, Public IPv4 DNS, Private IPv4 DNS etc. You can also check the Security, Networking, Storage details etc.



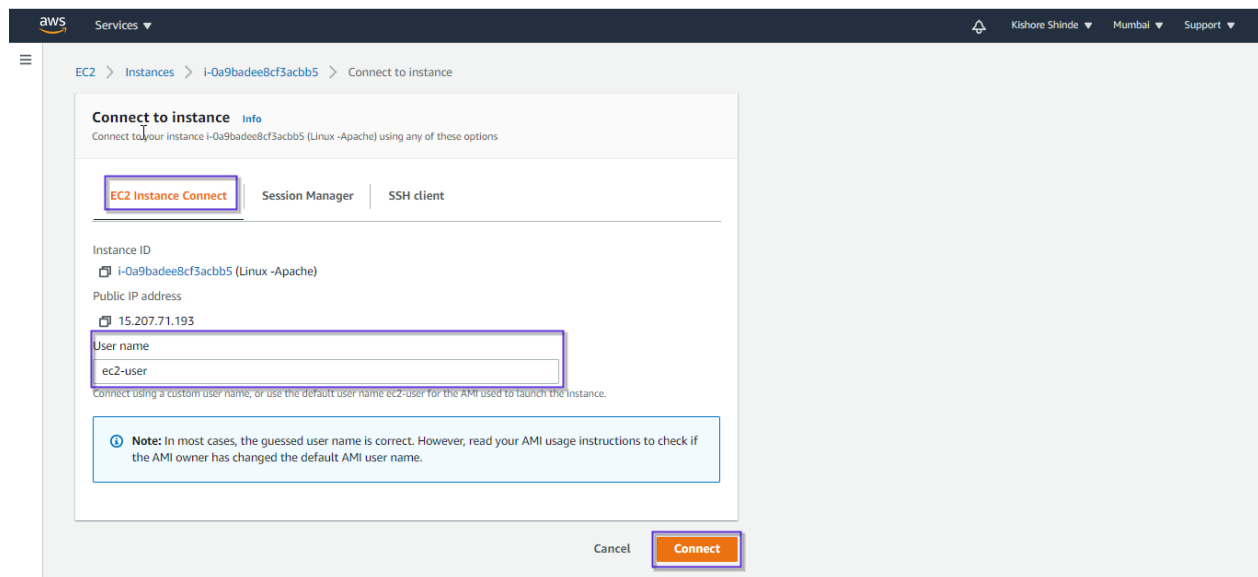
So now your Linux instance is up and running.

Step B: Connecting to Linux Instance

Following are the steps to connect to the instance. Select the instance, Click on Action menu at the top, from the list select Connect.



On the next screen you will see EC2 Instance Connect (default selected), Session Manager, SSH client Tabs. We will connect by EC2 Instance Connect



Let the default User Name “ec2-user” as it is and click on Connect.”

The Linux instance will be connected. Once the instance is started you will be able to see the \$ prompt.

2. yum -y update

- this command will download & update it to latest version

```
Verifying : rpm-plugin-systemd-inhibit-4.11.3-40.amzn2.0.5.x86_64 3/25
Verifying : rpm-libs-4.11.3-40.amzn2.0.5.x86_64 4/25
Verifying : ec2-net-utils-1.4-3.amzn2.noarch 5/25
Verifying : pam-1.1.8-23.amzn2.0.1.x86_64 6/25
Verifying : python2-boto-1.18.6-1.amzn2.0.1.noarch 7/25
Verifying : kernel-4.14.198-152.320.amzn2.x86_64 8/25
Verifying : rpm-4.11.3-40.amzn2.0.5.x86_64 9/25
Verifying : p11-kit-trust-0.23.21-2.amzn2.0.1.x86_64 10/25
Verifying : kernel-tools-4.14.198-152.320.amzn2.x86_64 11/25
Verifying : rpm-build-libs-4.11.3-40.amzn2.0.5.x86_64 12/25
Verifying : awscli-1.18.147-1.amzn2.0.1.noarch 13/25
Verifying : ec2-net-utils-1.4-2.amzn2.noarch 14/25
Verifying : rpm-4.11.3-40.amzn2.0.4.x86_64 15/25
Verifying : awscli-1.18.107-1.amzn2.0.1.noarch 16/25
Verifying : p11-kit-trust-0.23.19-1.amzn2.x86_64 17/25
Verifying : rpm-build-libs-4.11.3-40.amzn2.0.4.x86_64 18/25
Verifying : p11-kit-0.23.19-1.amzn2.x86_64 19/25
Verifying : pam-1.1.8-22.amzn2.x86_64 20/25
Verifying : kernel-tools-4.14.193-149.317.amzn2.x86_64 21/25
Verifying : rpm-libs-4.11.3-40.amzn2.0.4.x86_64 22/25
Verifying : rpm-plugin-systemd-inhibit-4.11.3-40.amzn2.0.4.x86_64 23/25
Verifying : python2-boto-1.17.31-1.amzn2.0.1.noarch 24/25
Verifying : python2-rpm-4.11.3-40.amzn2.0.4.x86_64 25/25

Installed:
kernel.x86_64 0:4.14.198-152.320.amzn2

Updated:
awscli.noarch 0:1.18.147-1.amzn2.0.1          ec2-net-utils.noarch 0:1.4-3.amzn2          kernel-tools.x86_64 0:4.14.198-152.320.amzn2
p11-kit.x86_64 0:0.23.21-2.amzn2.0.1         p11-kit-trust.x86_64 0:0.23.21-2.amzn2.0.1                 pam.x86_64 0:1.1.8-23.amzn2.0.1
python2-boto.noarch 0:1.18.6-1.amzn2.0.1     python2-rpm.x86_64 0:4.11.3-40.amzn2.0.5         rpm.x86_64 0:4.11.3-40.amzn2.0.5
rpm-build-libs.x86_64 0:4.11.3-40.amzn2.0.5   rpm-libs.x86_64 0:4.11.3-40.amzn2.0.5         rpm-plugin-systemd-inhibit.x86_64 0:4.11.3-40.amzn2.0.5

Complete!
[root@ip-172-31-0-237 ec2-user]#
```

i-0a9badee8cf3acbb5 (Linux -Apache)

Public IPs: 15.207.71.193 Private IPs: 172.31.0.237

All the updates will be installed & you will see a Complete message above the \$ prompt.

3. clear

- This command will clear the screen

4. yum install httpd

- this command will install Apache Server

```
[root@ip-172-31-0-237 ec2-user]# yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.46-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.46-1.amzn2 for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.46-1.amzn2 for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.46-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.6.3-5.amzn2.0.2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpd-filesystem.noarch 0:2.4.46-1.amzn2 will be installed
--> Package httpd-tools.x86_64 0:2.4.46-1.amzn2 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
--> Package mod_http2.x86_64 0:1.15.14-2.amzn2 will be installed
--> Running transaction check
--> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

```

total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.6.3-5.amzn2.0.2.x86_64 1/9
  Installing : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64 2/9
  Installing : apr-util-1.6.1-5.amzn2.0.2.x86_64 3/9
  Installing : httpd-tools-2.4.46-1.amzn2.x86_64 4/9
  Installing : generic-logos-httpd-18.0.0-4.amzn2.noarch 5/9
  Installing : mailcap-2.1.41-2.amzn2.noarch 6/9
  Installing : httpd-filesystem-2.4.46-1.amzn2.noarch 7/9
  Installing : mod_http2-1.15.14-2.amzn2.x86_64 8/9
  Installing : httpd-2.4.46-1.amzn2.x86_64 9/9
  Verifying : apr-util-1.6.1-5.amzn2.0.2.x86_64 1/9
  Verifying : httpd-filesystem-2.4.46-1.amzn2.noarch 2/9
  Verifying : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64 3/9
  Verifying : httpd-tools-2.4.46-1.amzn2.x86_64 4/9
  Verifying : mod_http2-1.15.14-2.amzn2.x86_64 5/9
  Verifying : apr-1.6.3-5.amzn2.0.2.x86_64 6/9
  Verifying : mailcap-2.1.41-2.amzn2.noarch 7/9
  Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch 8/9
  Verifying : httpd-2.4.46-1.amzn2.x86_64 9/9

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.x86_64 0:1.6.1-5.amzn2.0.2  apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2  generic-logos-httpd.noarch 0:18.0.0-4.amzn2
  httpd-filesystem.noarch 0:2.4.46-1.amzn2  httpd-tools.x86_64 0:2.4.46-1.amzn2  mailcap.noarch 0:2.1.41-2.amzn2  mod_http2.x86_64 0:1.15.14-2.amzn2

complete!
root@ip-172-31-0-237 ec2-user]#

```

When prompted Press “Y” for confirmation.

You will see a Complete message above the \$ prompt once Apache server is installed.

5. systemctl start httpd

- this command will start the Apache Server

```

[root@ip-172-31-0-237 ec2-user]# systemctl start httpd
[root@ip-172-31-0-237 ec2-user]#

```

6. systemctl enable httpd

- this command will enable the Apache Server

```

[root@ip-172-31-0-237 ec2-user]# systemctl start httpd
[root@ip-172-31-0-237 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-172-31-0-237 ec2-user]#

```

7. systemctl status httpd

- this command will check the status of the Apache Server

```

[root@ip-172-31-0-237 ec2-user]# systemctl start httpd
[root@ip-172-31-0-237 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-172-31-0-237 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2020-10-09 07:16:43 UTC; 4min 23s ago
     Docs: man:httpd.service(8)
  Main PID: 14321 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
   CGroup: /system.slice/httpd.service
           └─14321 /usr/sbin/httpd -DFOREGROUND
             └─14322 /usr/sbin/httpd -DFOREGROUND
               └─14323 /usr/sbin/httpd -DFOREGROUND
                 └─14324 /usr/sbin/httpd -DFOREGROUND
                   └─14325 /usr/sbin/httpd -DFOREGROUND
                     └─14326 /usr/sbin/httpd -DFOREGROUND

Oct 09 07:16:43 ip-172-31-0-237.ap-south-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Oct 09 07:16:43 ip-172-31-0-237.ap-south-1.compute.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-172-31-0-237 ec2-user]#

```

Now we can see the Apache server is active and running.

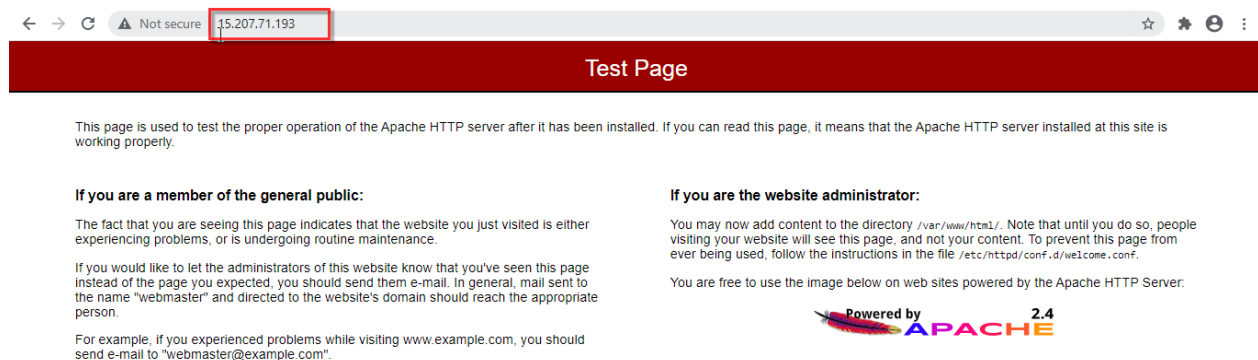
Disconnect the server.

Now to check that the Apache Server is installed properly. Copy the public address from the Instance details & paste it in your Internet browser (e.g. Chrome/Internet Explorer/Firefox) or click on open. In our current scenario our Linux server public IP is: 15.207.71.193

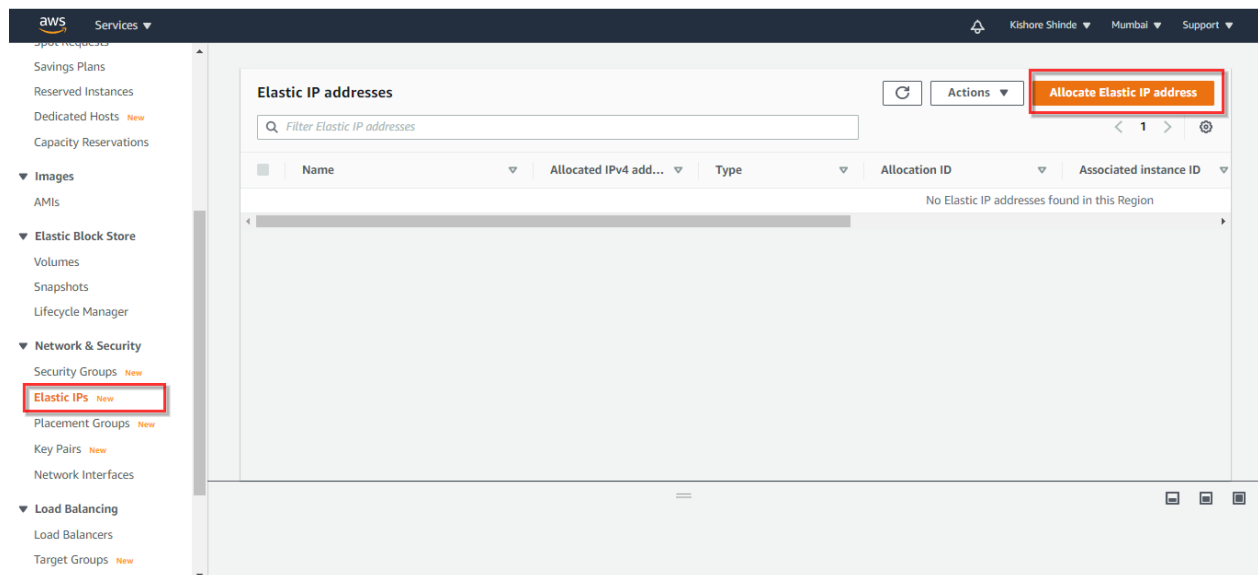
The screenshot shows the AWS Management Console interface. On the left, the 'Instances' menu is expanded. The main panel displays the details for the instance 'Linux-Apache' (ID: i-0a9badee8cf3acbb5). The instance is in a 'Running' state. The 'Instance summary' section is visible, showing the following details:

- Instance ID:** i-0a9badee8cf3acbb5 (Linux -Apache)
- Instance state:** Running
- Instance type:** t2.micro
- Public IPv4 address:** 15.207.71.193 | [open address](#)
- Public IPv4 DNS:** ec2-15-207-71-193.ap-south-1.compute.amazonaws.com | [open address](#)
- Private IPv4 addresses:** 172.31.0.237
- Private IPv4 DNS:** ip-172-31-0-237.ap-south-1.compute.internal
- VPC ID:** vpc-afc1dbc7

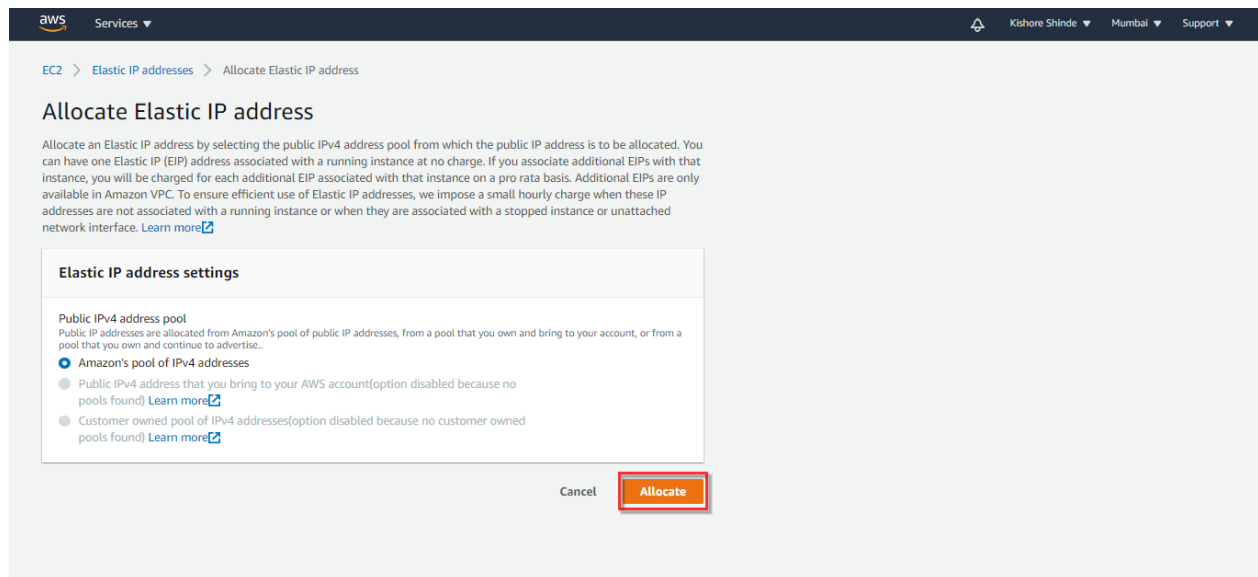
You should be able to see the Apache web server installed on Linux instance.



STEP D: Creating Elastic IP

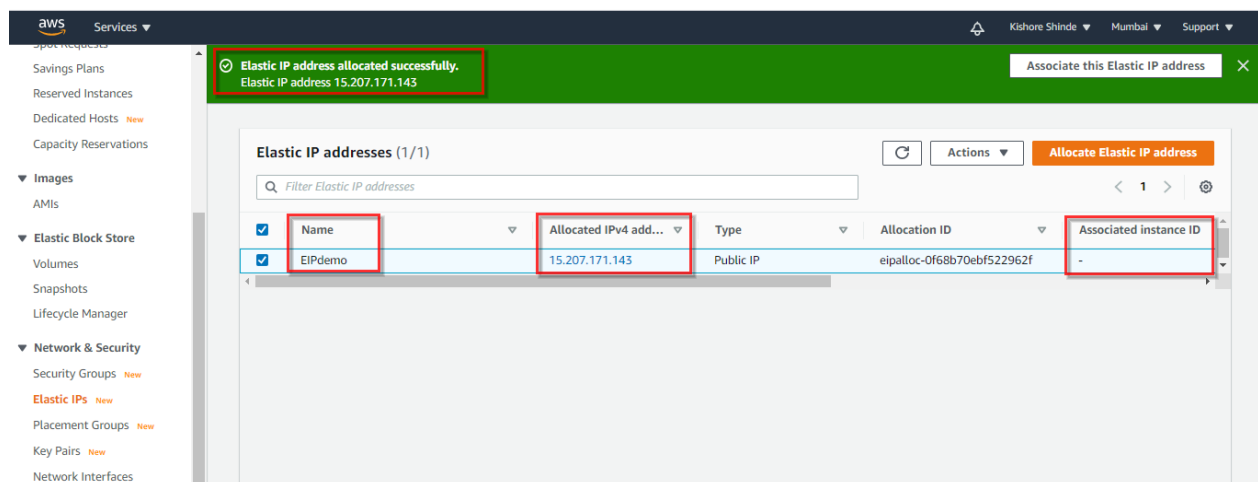


Select **Elastic IPs** from the left pane. You will be able to see the Elastic IP addresses dashboard. Click on **Allocate Elastic IP address**



Click on **Allocate**.

You will be taken to Elastic IP Dashboard and you will see that **Elastic IP is allocated successfully**.

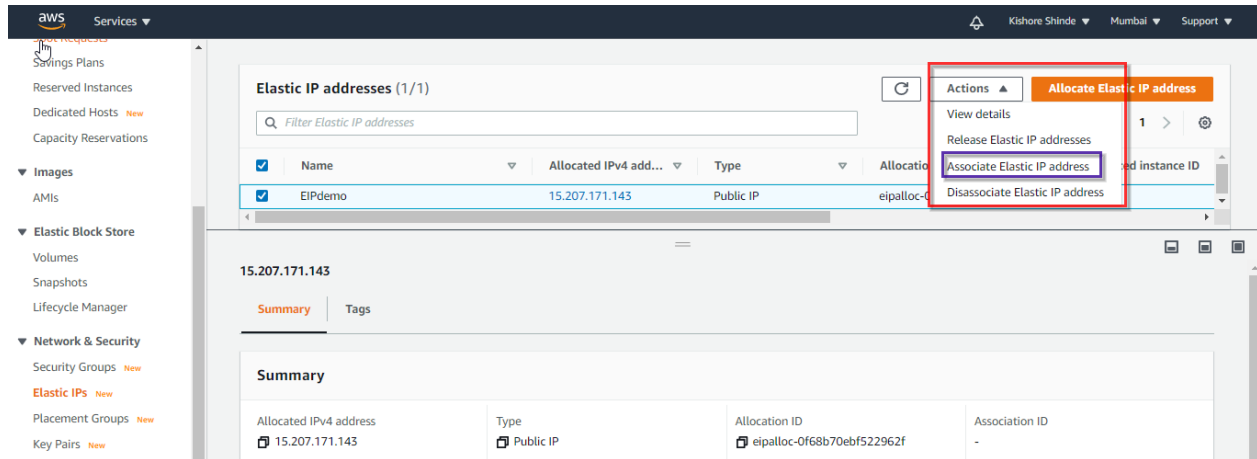


Give it a name e.g. EIPdemo, you can see the allocated IP (15.207.171.143). You can see the Associated instance ID is blank as we not yet associated it with any instance.

Note: Make sure you associate the Elastic IP with an instance if you keep it idle you will be charged.

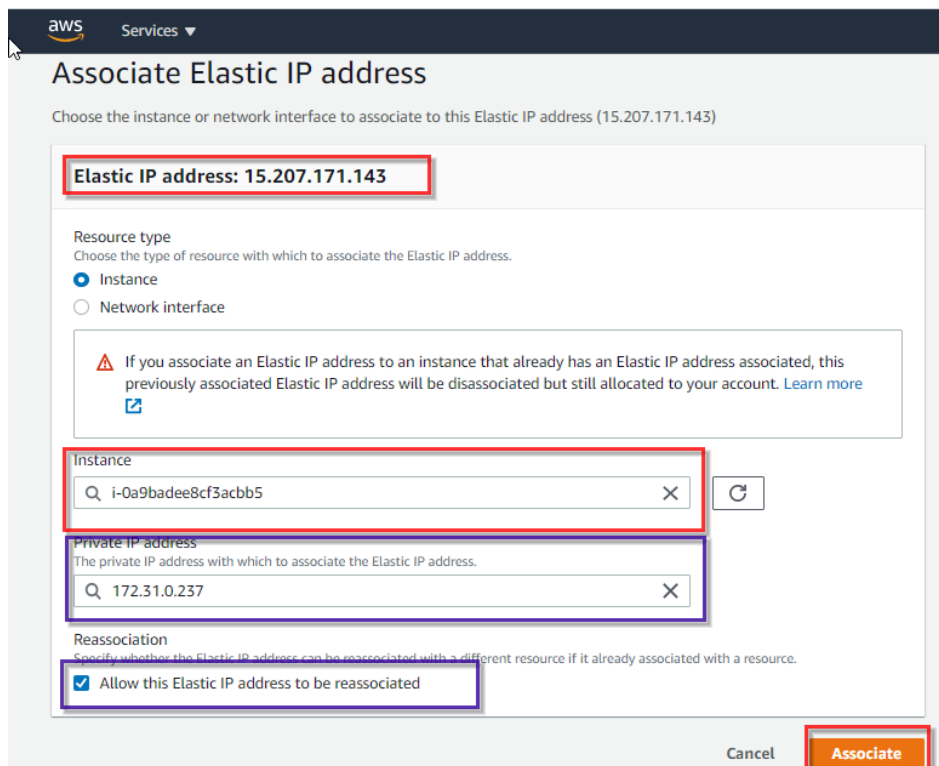
STEP E: Associating Elastic IP

Select the Elastic IP you want to associate to an instance.



Click on **Actions** then click on **Associate Elastic IP address**.

On the Associate Elastic IP address screen, in **Resource type** select **Instance** (selected by default).



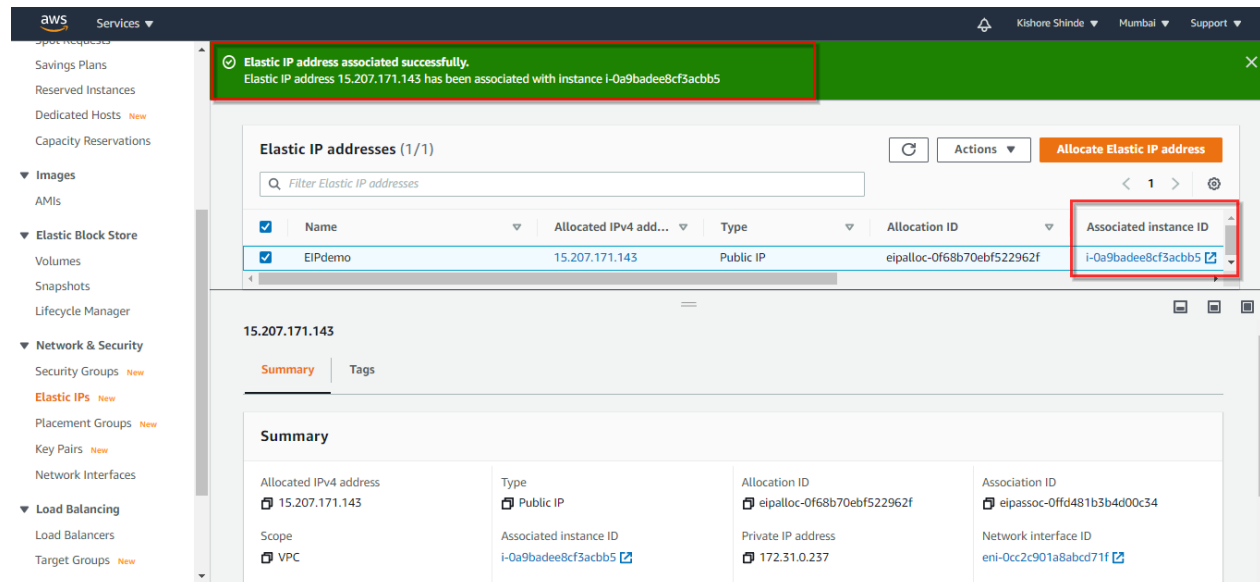
From the **Instance** select the Instance to which you want to associate the Elastic IP.

Private IP address: Will come automatically just select it.

Reassociation: Check **Allow this Elastic IP address to be reassociated**. This will make sure that once you terminate the instance to which Elastic IP is associated you can then associate the Elastic IP with another instance.

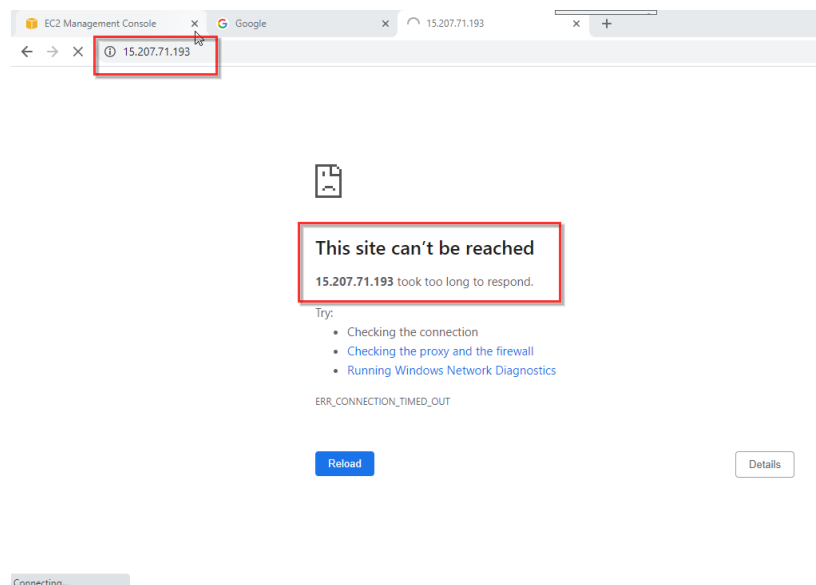
Click on **Associate** to associate the Elastic IP.

It will take you back to the Elastic IP dashboard.

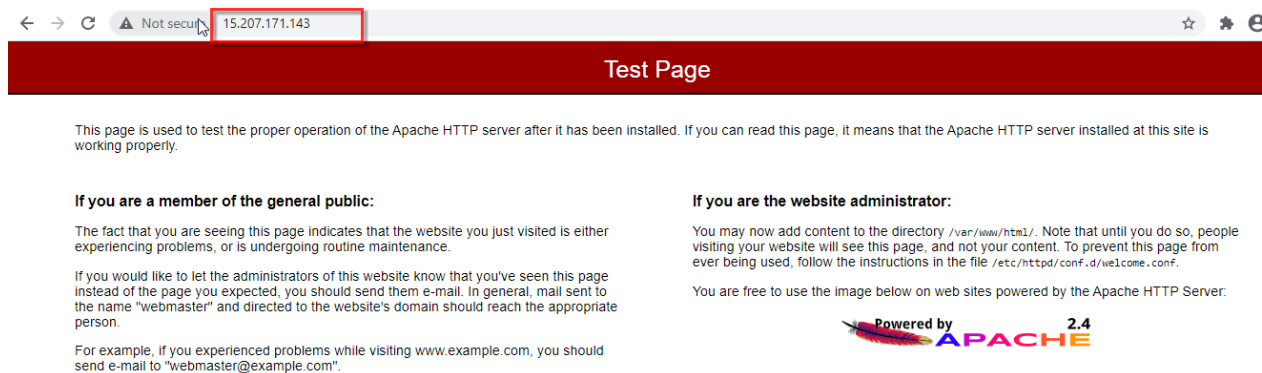


You can see the Elastic IP is associated successfully and in Associated Instance you can see the instance id of the Linux instance with which we have associated.

Now if you check the Linux instance with the public IP (15.207.71.193) that was created with it. You will not be able to access the Apache Server.

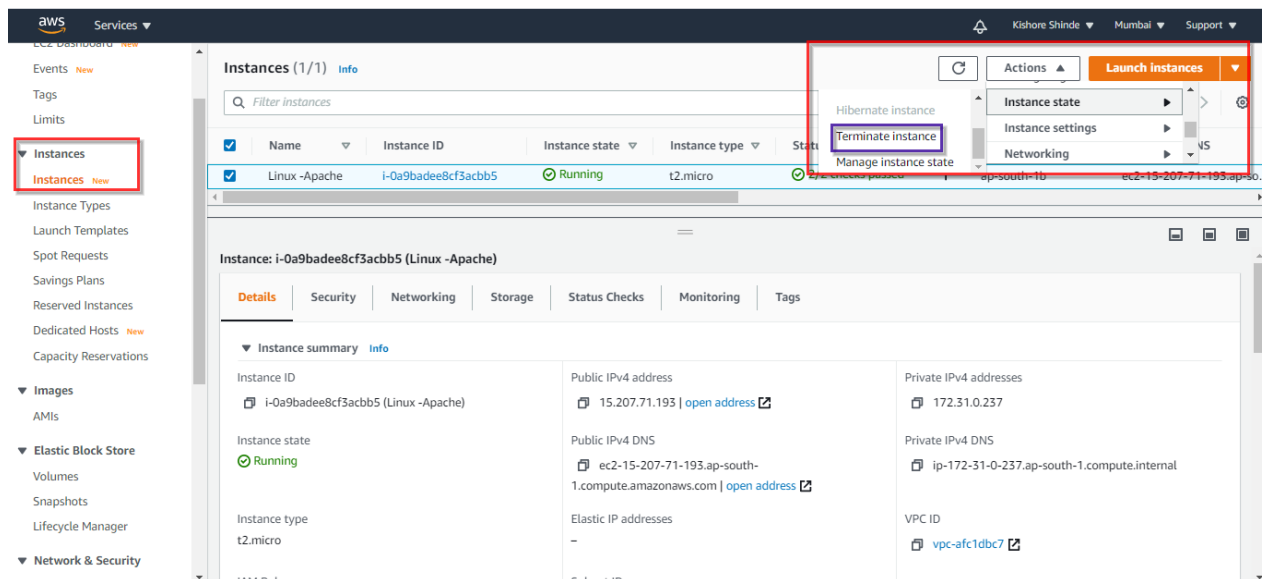


To access the Linux Instance/Apache Server you will have to check it with newly associated Elastic IP (15.2017.171.143).



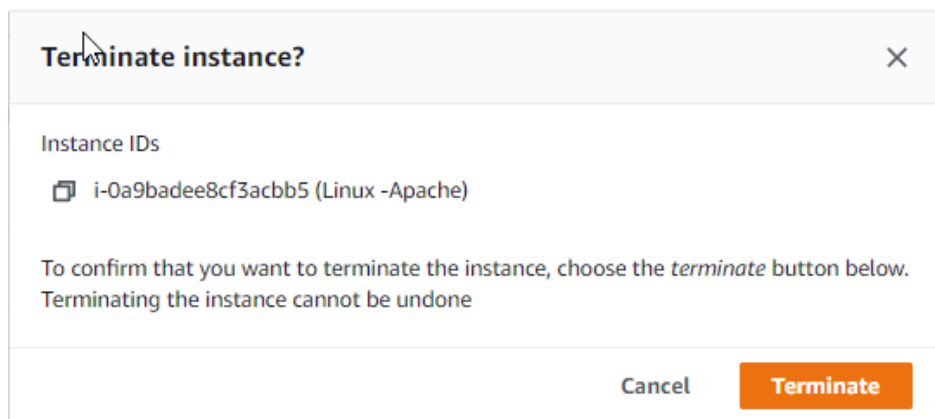
STEP F: Terminating a Linux Instance

To Terminate the instance, go to Instances.

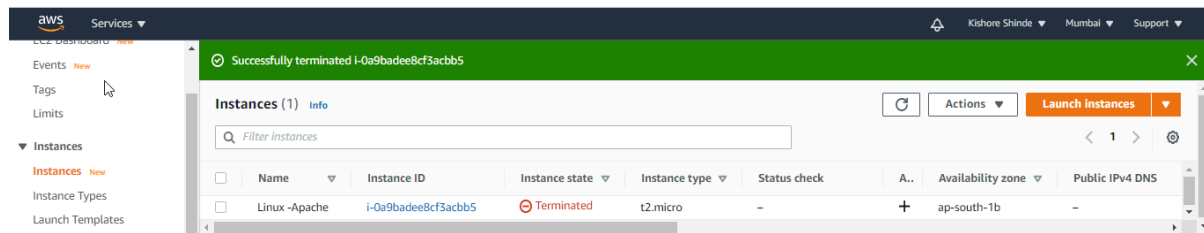


Click on Actions->Select Instance State->Click on Terminate instance.

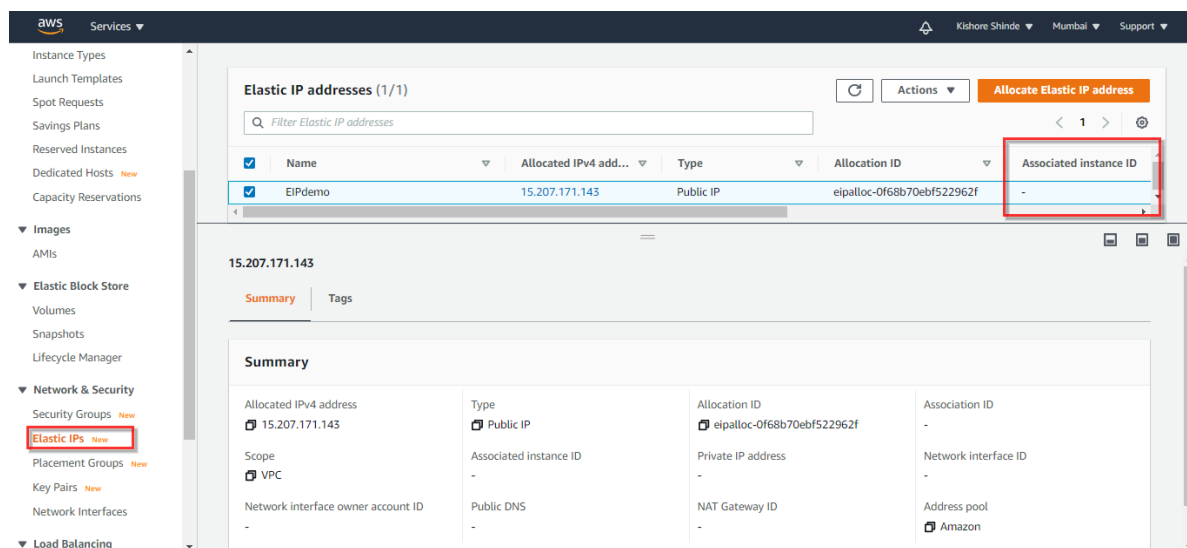
It will ask you for Termination permission click on Terminate.



The instance will be terminated, the instance status will show terminated.

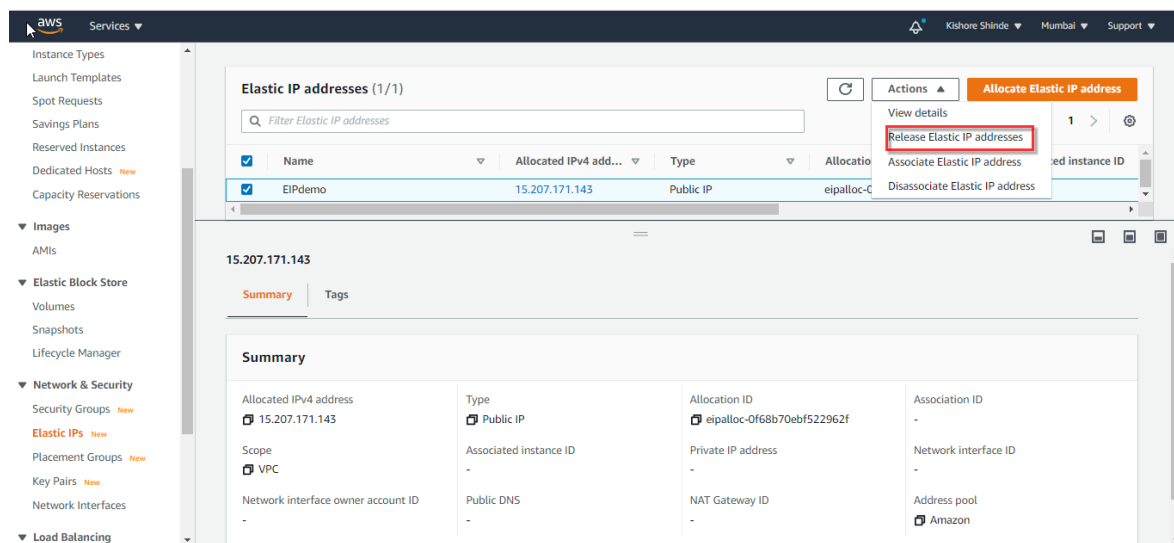


Go To Elastic IPs. You can check that as soon as the instance is terminated the Associate Instance ID is removed and the IP is available to be associated with new instance.



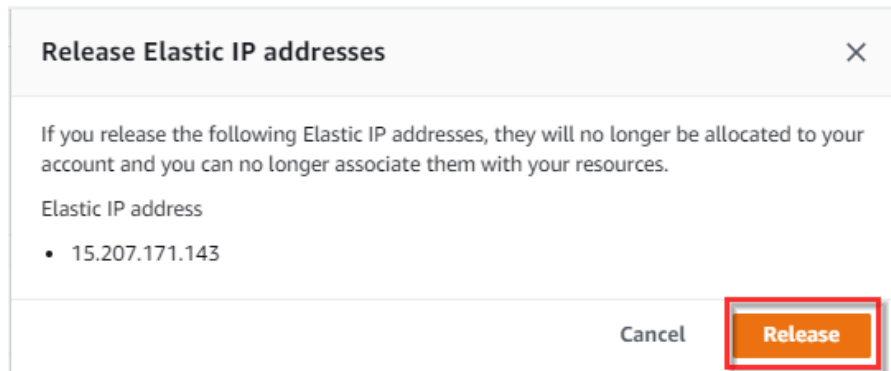
STEP G: Releasing/Deleting Elastic IP

To Release or Delete the Elastic IP. Click on Action.

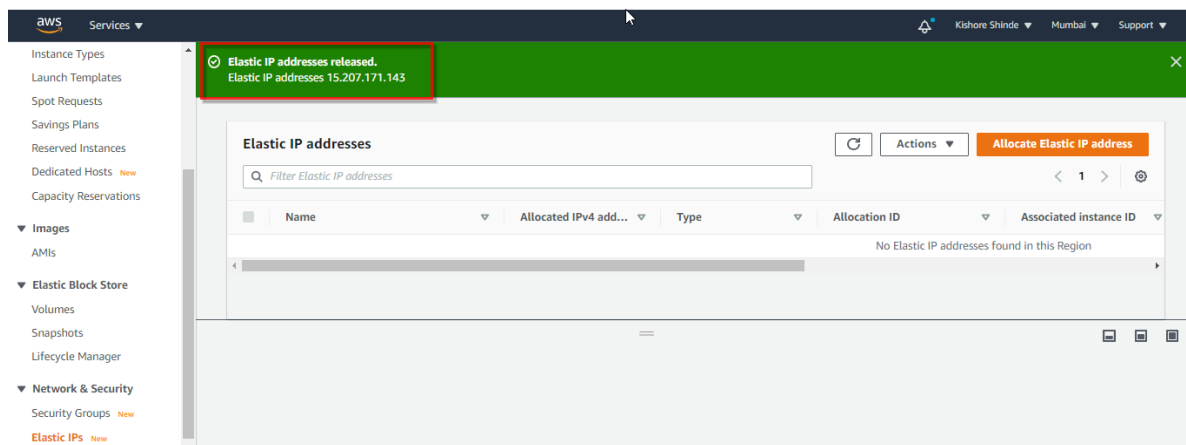


Then Click on **Release Elastic IP address**.

It will ask you for permission. Click on **Release**.



You will be taken back to Elastic IP dashboard.



You will see the Elastic IP is released.

Project 4 is completed.