

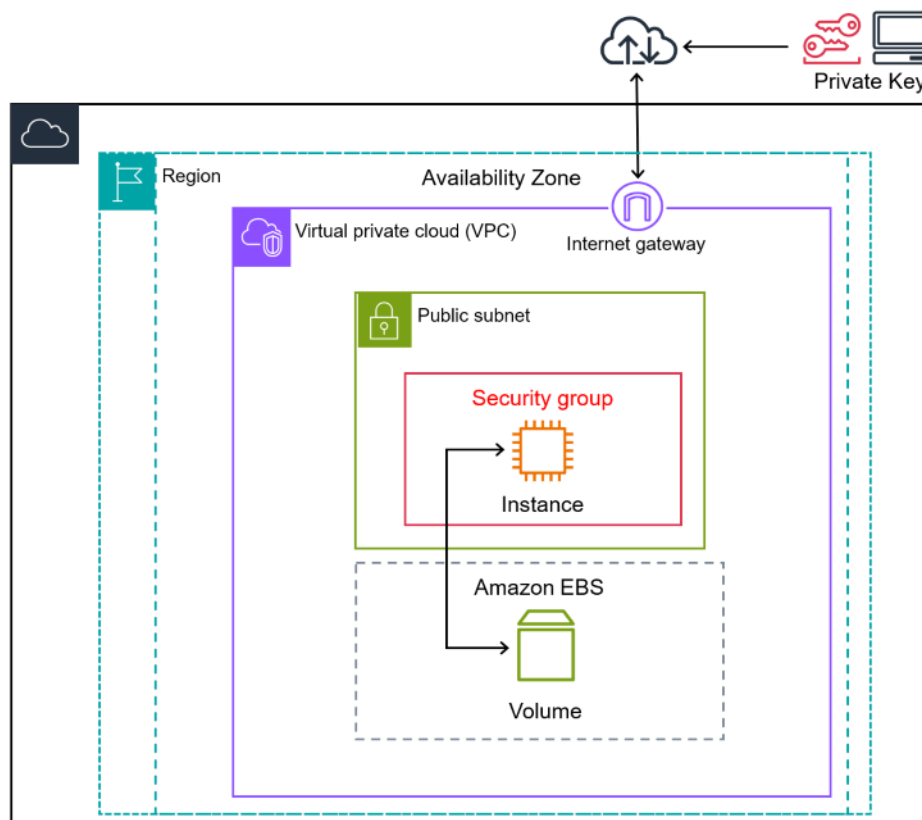
## Case study on Amazon EC2 and learn about Amazon EC2 web services.

### What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. You can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, you can reduce capacity (scale down) again.

### Basic Architecture of Amazon EC2

The following diagram shows a basic architecture of an Amazon EC2 instance deployed within an Amazon Virtual Private Cloud (VPC). In this example, the EC2 instance is within an Availability Zone in the Region. The EC2 instance is secured with a security group, which is a virtual firewall that controls incoming and outgoing traffic. A private key is stored on the local computer and a public key is stored on the instance. Both keys are specified as a key pair to prove the identity of the user. In this scenario, the instance is backed by an Amazon EBS volume. The VPC communicates with the internet using an internet gateway.



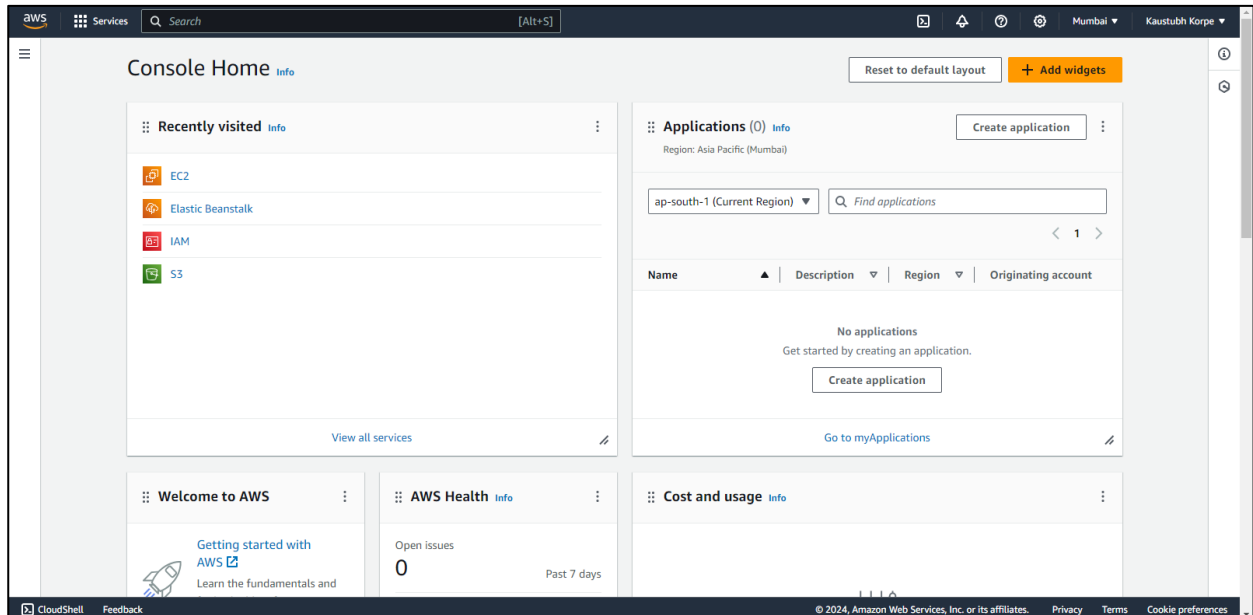
## Features of Amazon EC2

Amazon EC2 provides the following high-level features:

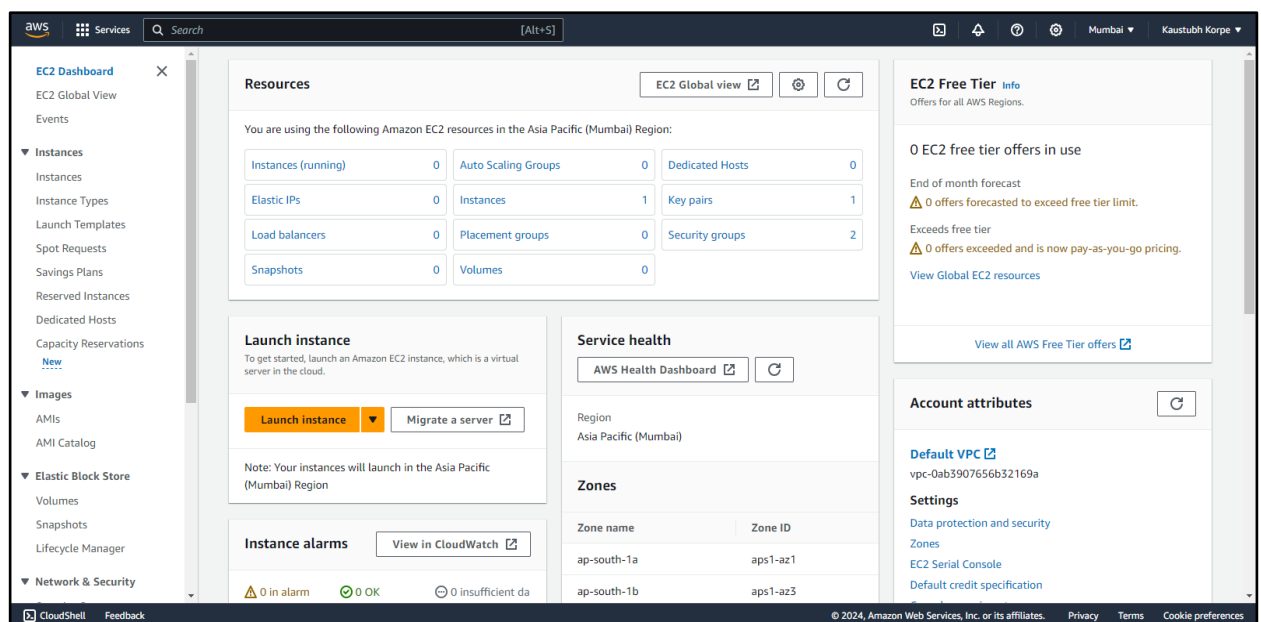
- **Instances:** Virtual servers.
- **Amazon Machine Images (AMIs):** Preconfigured templates for your instances that package the components you need for your server (including the operating system and additional software).
- **Instance types:** Various configurations of CPU, memory, storage, networking capacity, and graphics hardware for your instances.
- **Key pairs:** Secure login information for your instances. AWS stores the public key and you store the private key in a secure place.
- **Instance store volumes:** Storage volumes for temporary data that is deleted when you stop, hibernate, or terminate your instance.
- **Amazon EBS volumes:** Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS).
- **Regions, Availability Zones, Local Zones, AWS Outposts, and Wavelength Zones:** Multiple physical locations for your resources, such as instances and Amazon EBS volumes.
- **Security groups:** A virtual firewall that allows you to specify the protocols, ports, and source IP ranges that can reach your instances, and the destination IP ranges to which your instances can connect.
- **Elastic IP addresses:** Static IPv4 addresses for dynamic cloud computing.
- **Tags:** Metadata that you can create and assign to your Amazon EC2 resources.
- **Virtual private clouds (VPCs):** Virtual networks you can create that are logically isolated from the rest of the AWS Cloud. You can optionally connect these virtual networks to your own network.

# Launching an Instance in Amazon EC2

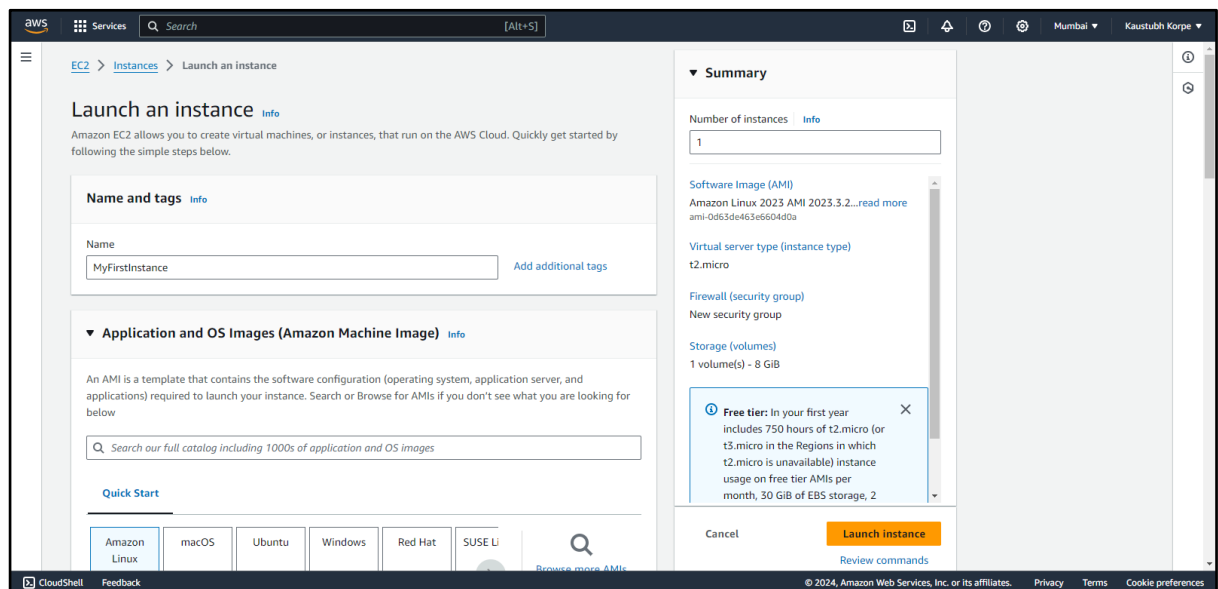
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.



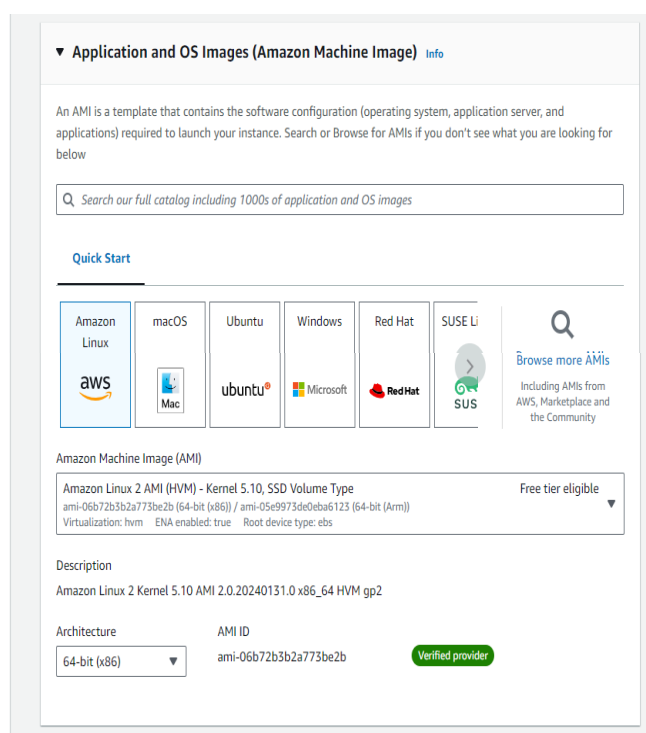
2. From the EC2 console dashboard, in the **Launch instance** box, choose **Launch instance**.



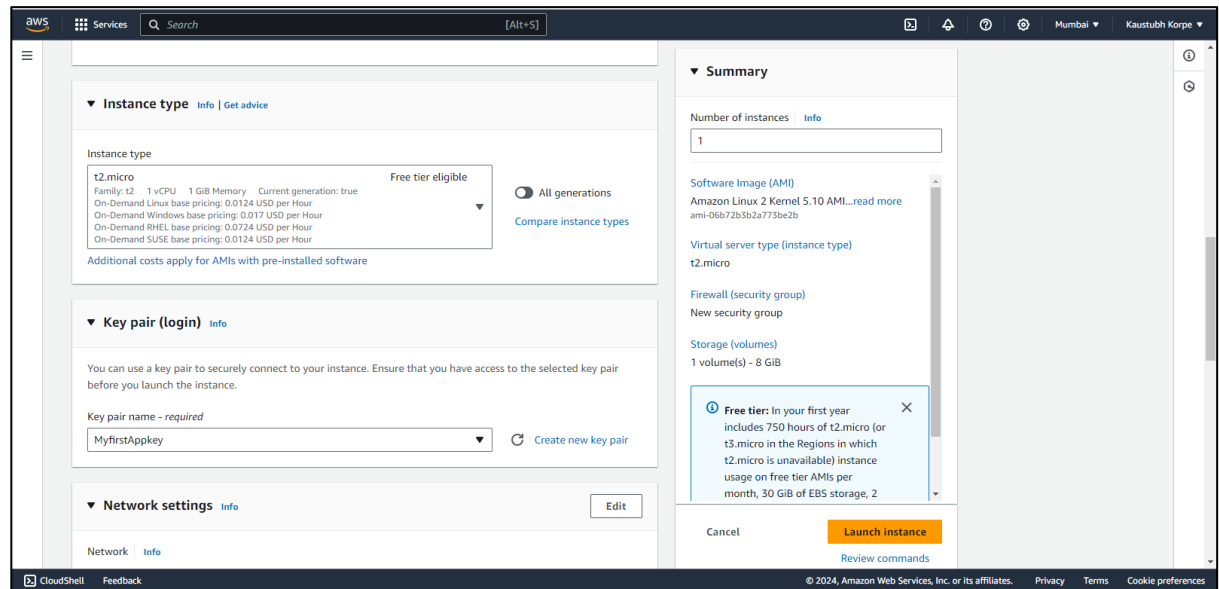
3. Under **Name and tags**, for **Name**, enter a descriptive name for your instance.



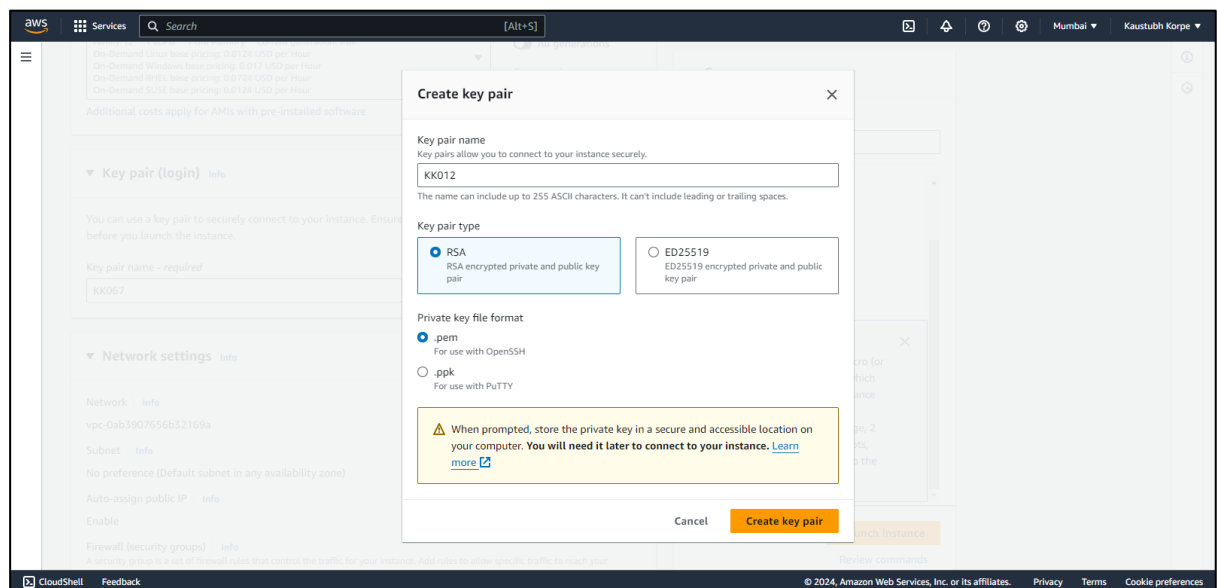
4. Under **Application and OS Images (Amazon Machine Image)**, do the following:
- Choose **Quick Start**, and then choose Amazon Linux. This is the operating system (OS) for your instance. From **Amazon Machine Image (AMI)**, select an HVM version of Amazon Linux 2.
  - Notice that these AMIs are marked **Free Tier eligible**. An *Amazon Machine Image (AMI)* is a basic configuration that serves as a template for your instance.



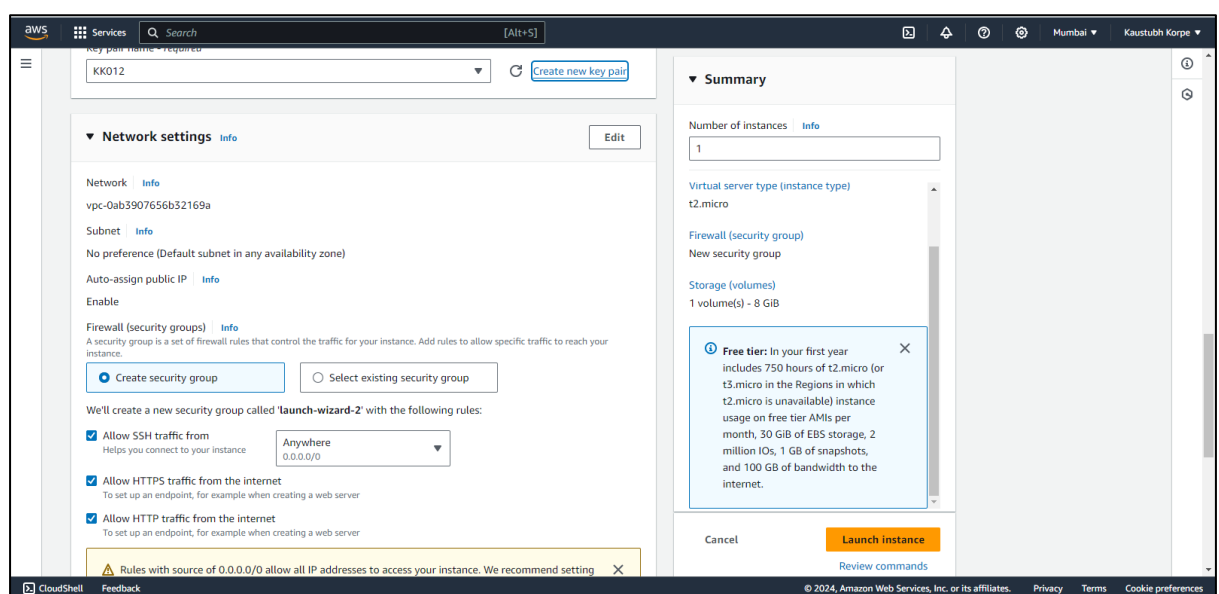
5. Under **Instance type**, from the **Instance type** list, you can select the hardware configuration for your instance. Choose the t2.micro instance type, which is selected by default. The t2.micro instance type is eligible for the Free Tier. In Regions where t2.micro is unavailable, you can use a t3.micro instance under the Free Tier.



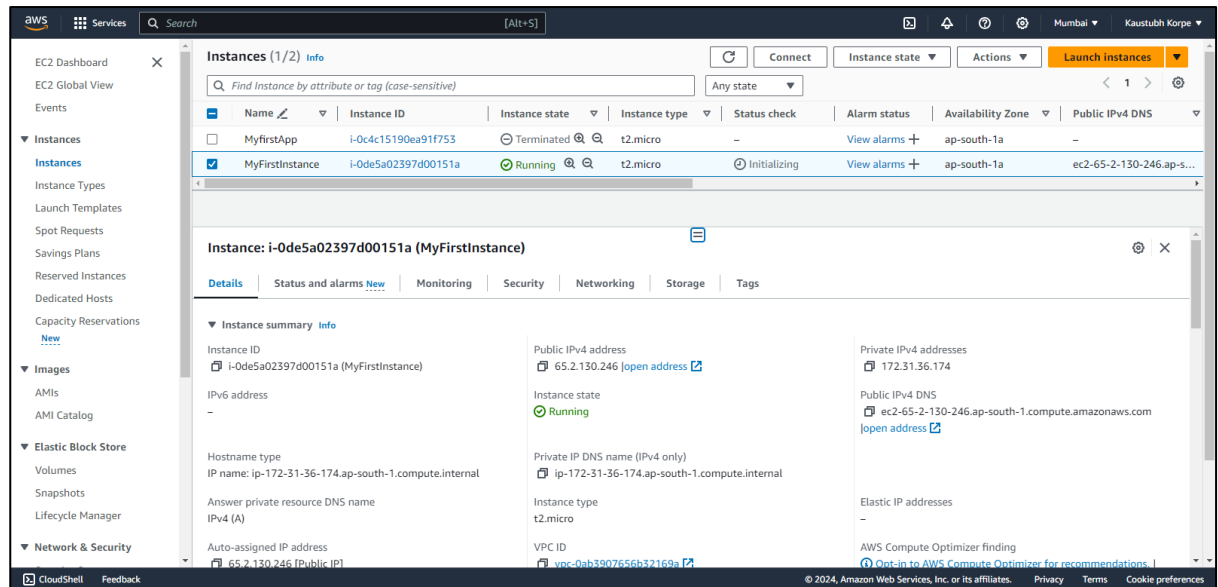
6. Under **Key pair (login)**, for **Key pair name**, choose the key pair that you created when getting set up.



7. Next to **Network settings**, choose **Edit**. For **Security group name**, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:
  - a. Choose **Select existing security group**.
  - b. From **Common security groups**, choose your security group from the list of existing security groups.
8. Keep the default selections for the other configuration settings for your instance.
9. Review a summary of your instance configuration in the **Summary** panel, and when you're ready, choose **Launch instance**.

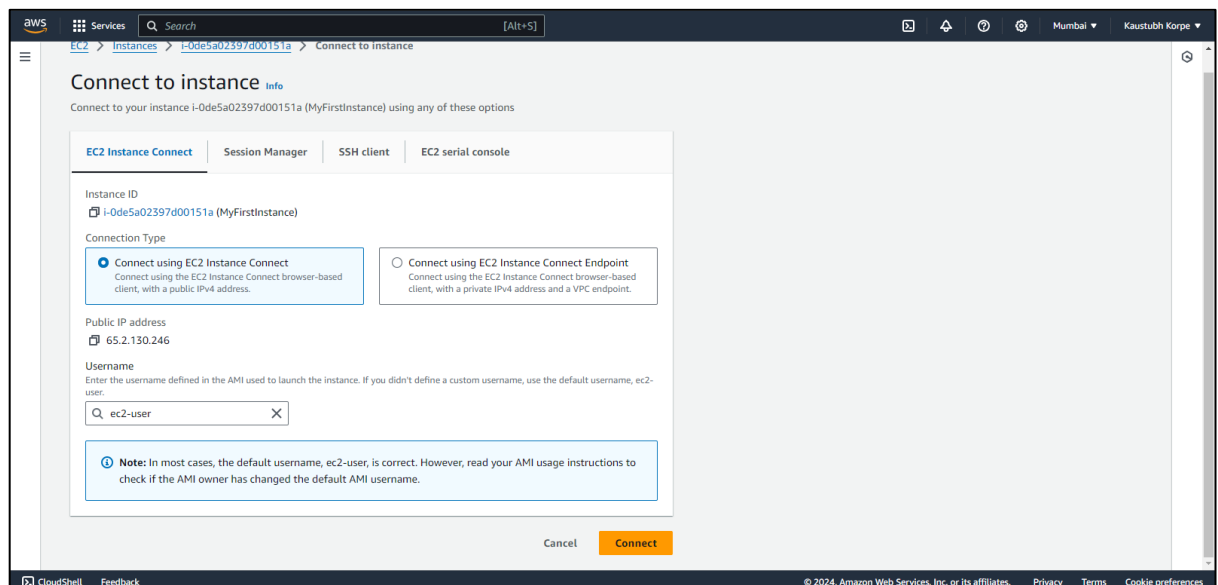


10. A confirmation page lets you know that your instance is launching. Choose **View all instances** to close the confirmation page and return to the console.
11. On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is pending. After the instance starts, its state changes to running and it receives a public DNS name. If the **Public IPv4 DNS** column is hidden, choose the settings icon (⚙️) in the top-right corner, toggle on **Public IPv4 DNS**, and choose **Confirm**.
12. It can take a few minutes for the instance to be ready for you to connect to it. Check that your instance has passed its status checks; you can view this information in the **Status check** column.

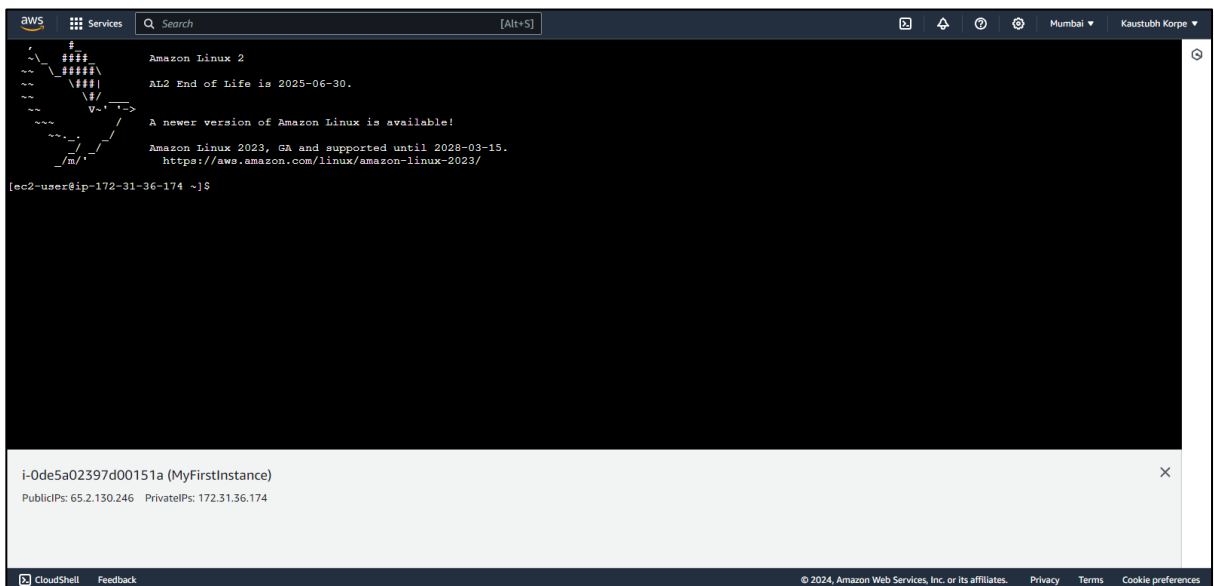


## Connecting and Installing Apache Server on the Instance

1. Select your instance and click connect as highlighted below.
2. Once, you click Connect, you will see a screen with the default username for a Linux 2 instance. Verify that **ec2-user** is showing in the username field and click **Connect**.



3. A new browser window will open and you will be connected to your instance like below.



4. Update the latest package available on the system by running following command

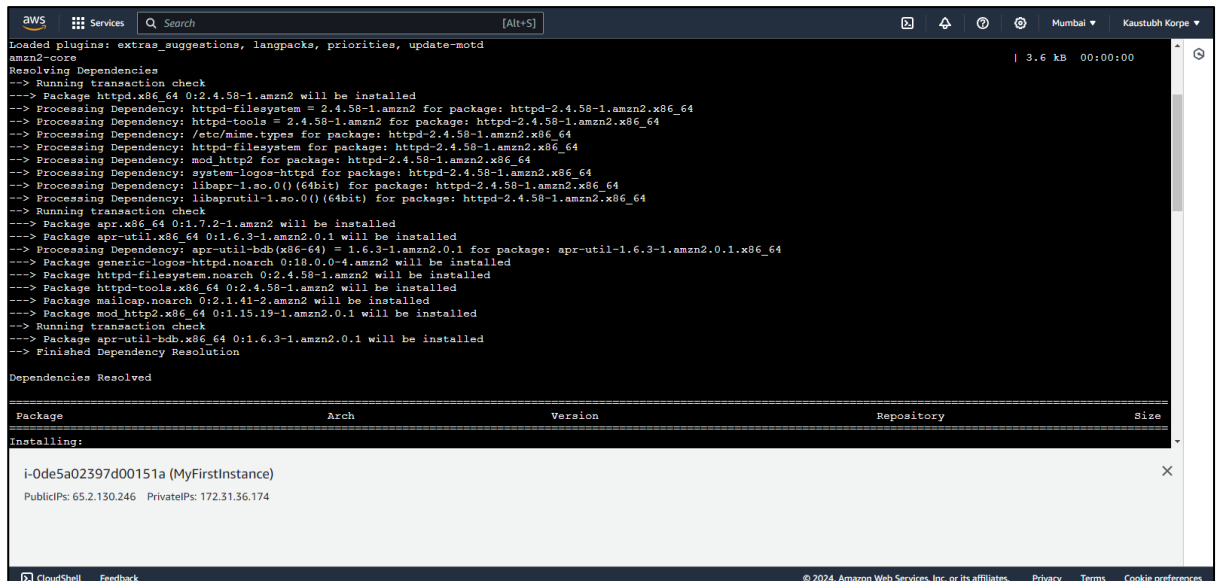
```
sudo yum update -y
```



5. Install Apache Web Server

```
sudo yum install -y httpd.x86_64
```





```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.58-1.amzn2 will be installed
--> Processing Dependency: httpd-filesystem = 2.4.58-1.amzn2 for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: httpd-tools = 2.4.58-1.amzn2 for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: /etc/pass.types for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.58-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.7.2-1.amzn2 will be installed
--> Package apr-util.x86_64 0:1.6.3-1.amzn2.0.1 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.3-1.amzn2.0.1 for package: apr-util-1.6.3-1.amzn2.0.1.x86_64
--> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpd-filesystem.noarch 0:2.4.58-1.amzn2 will be installed
--> Package httpd-tools.x86_64 0:2.4.58-1.amzn2 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
--> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

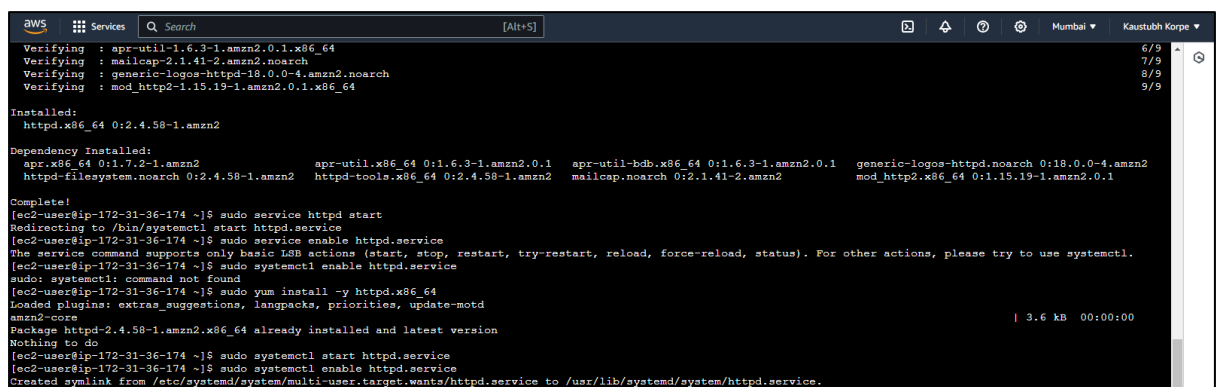
Package Arch Version Repository Size
Installing:
i-Ode5a02397d00151a (MyFirstInstance)
PublicIPs: 65.2.130.246 PrivateIPs: 172.31.36.174
```

6. Start Apache Server by running following command:

```
sudo yum install -y httpd.x86_64
```

7. Configure Apache to run on system boot by running following command:

```
sudo systemctl enable httpd.service
```



```
Verifying : apr-util-1.6.3-1.amzn2.0.1.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 : mod_http2-1.15.19-1.amzn2.0.1.x86_64 9/9

Installed:
httpd.x86_64 0:2.4.58-1.amzn2

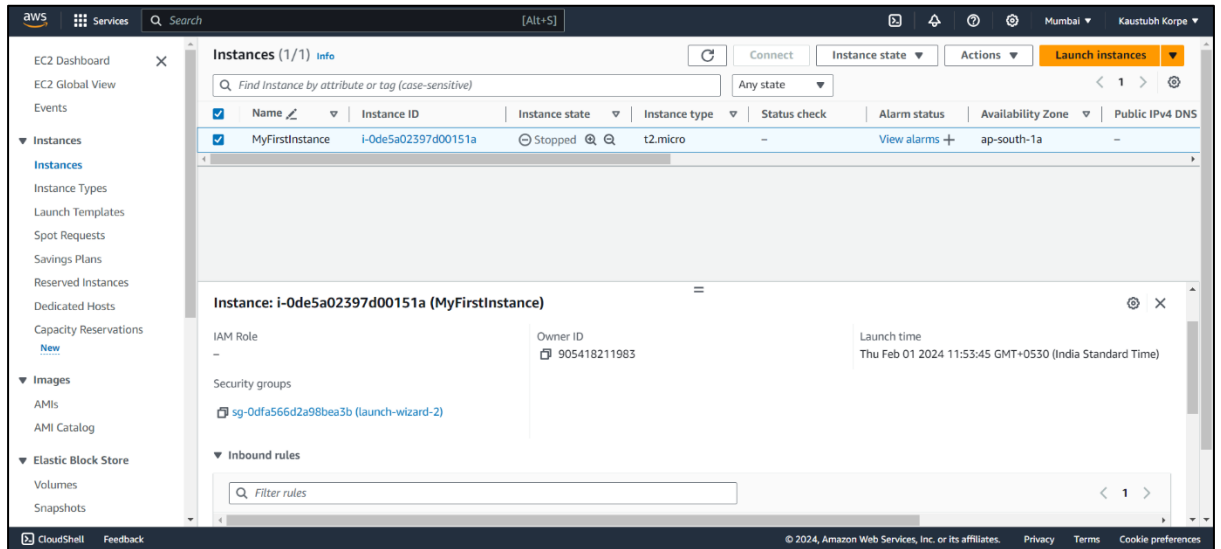
Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2 apr-util.x86_64 0:1.6.3-1.amzn2.0.1 apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1 generic-logos-httpd.noarch 0:18.0.0-4.amzn2
httpd-filesystem.noarch 0:2.4.58-1.amzn2 httpd-tools.x86_64 0:2.4.58-1.amzn2 mailcap.noarch 0:2.1.41-2.amzn2 mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
[ec2-user@ip-172-31-36-174 ~]$ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[ec2-user@ip-172-31-36-174 ~]$ sudo service enable httpd.service
The service command supports only basic LSB actions (start, stop, restart, try-restart, reload, force-reload, status). For other actions, please try to use systemctl.
[ec2-user@ip-172-31-36-174 ~]$ sudo systemctl enable httpd.service
sudo: systemctl: command not found
[ec2-user@ip-172-31-36-174 ~]$ sudo yum install -y httpd.x86_64
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Package httpd-2.4.58-1.amzn2.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-36-174 ~]$ sudo systemctl start httpd.service
[ec2-user@ip-172-31-36-174 ~]$ sudo systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
```

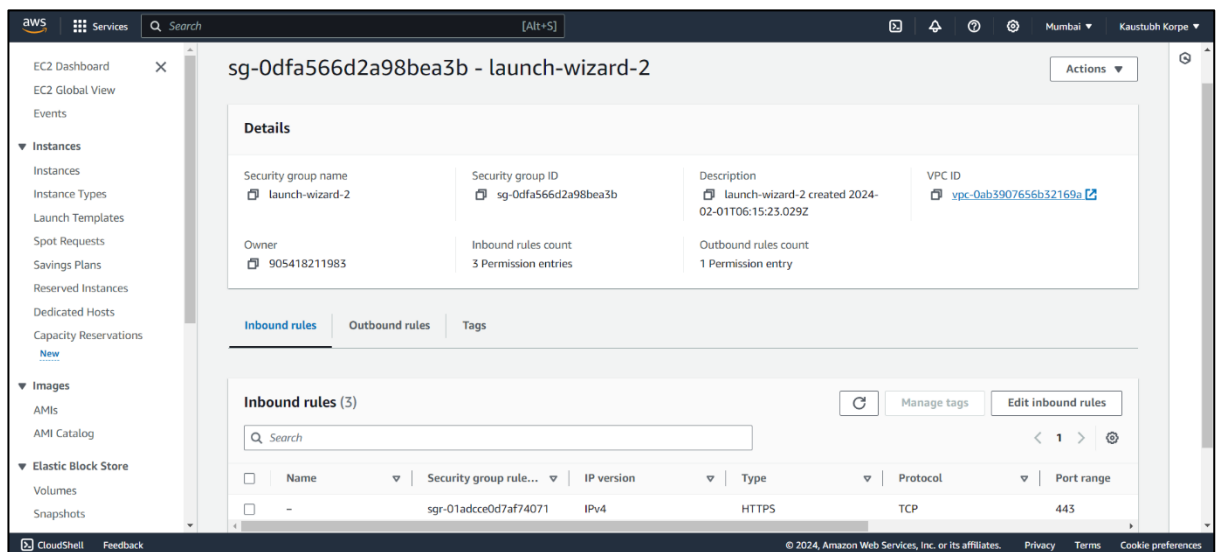
8. At this moment, your Apache web server is already installed and started in your Linux 2 instance. But you can't access it right now. if you try to hit the public IP you will get an error like **This site can't be reached**. Because our EC2 instance doesn't allow web traffic yet.

Change the Security Group of the instance to allow ports 80 and 443

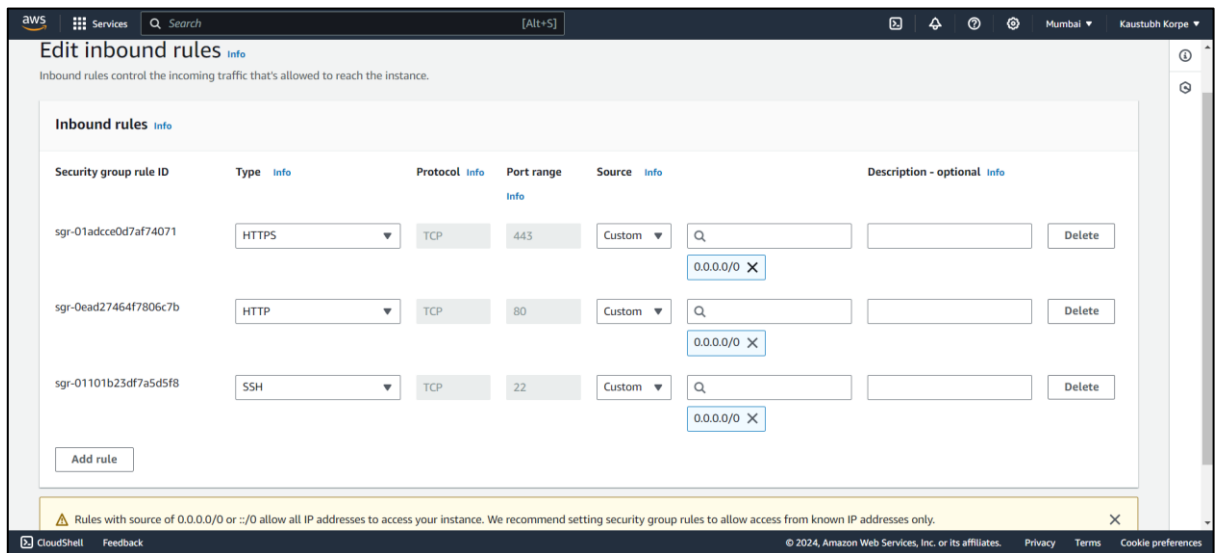
9. Click on your instance ID to see the instance details. Scroll down and click on the **Security Tab** and you should see the security group like below.



10. Click on the Security Group ID link-> Click on Edit Inbound rules.

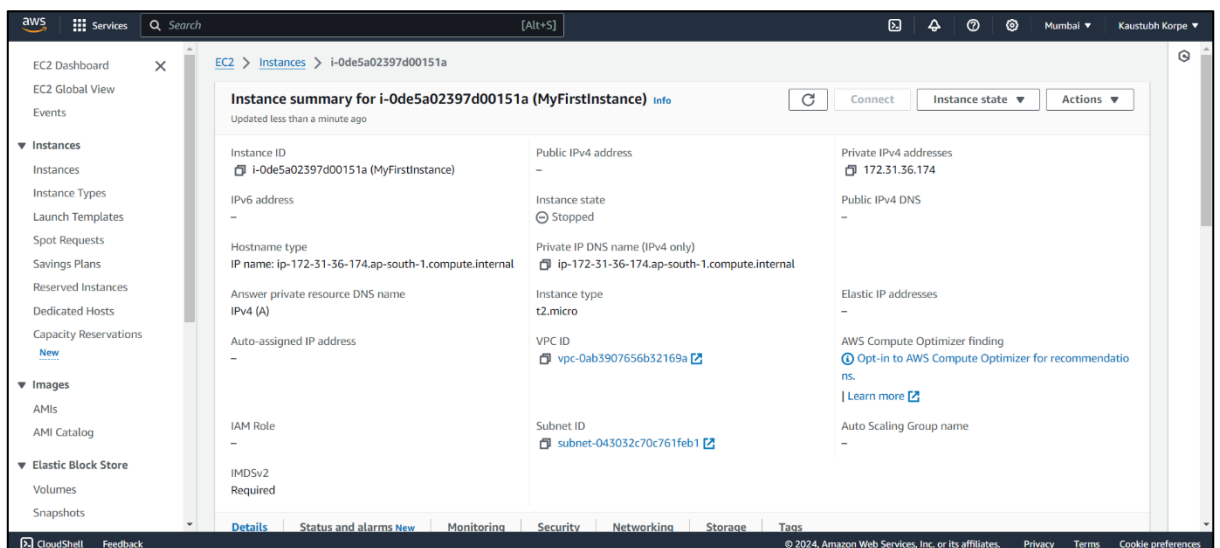


11. Use the **Add rule** button to add more rules one by one. Specify rules for HTTP and HTTPS Web traffic from anywhere like above.

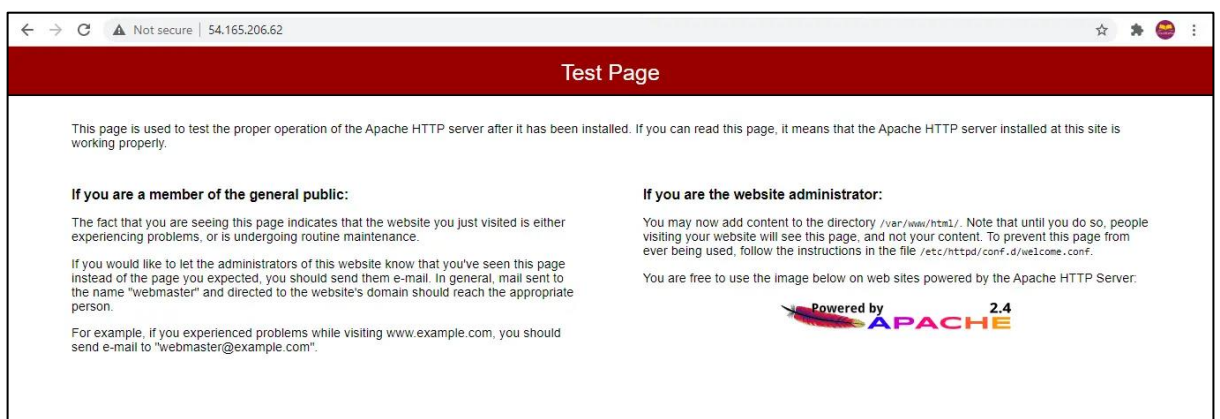


## 12. Verify the Installation

We have installed and started the Apache web server. Our instance allows web traffic now, it's time to grab the public IP or public DNS of the instance.



## 13. Click on the **open address** and you should be able to see the Apache default page below.



## 14. Customize the web page

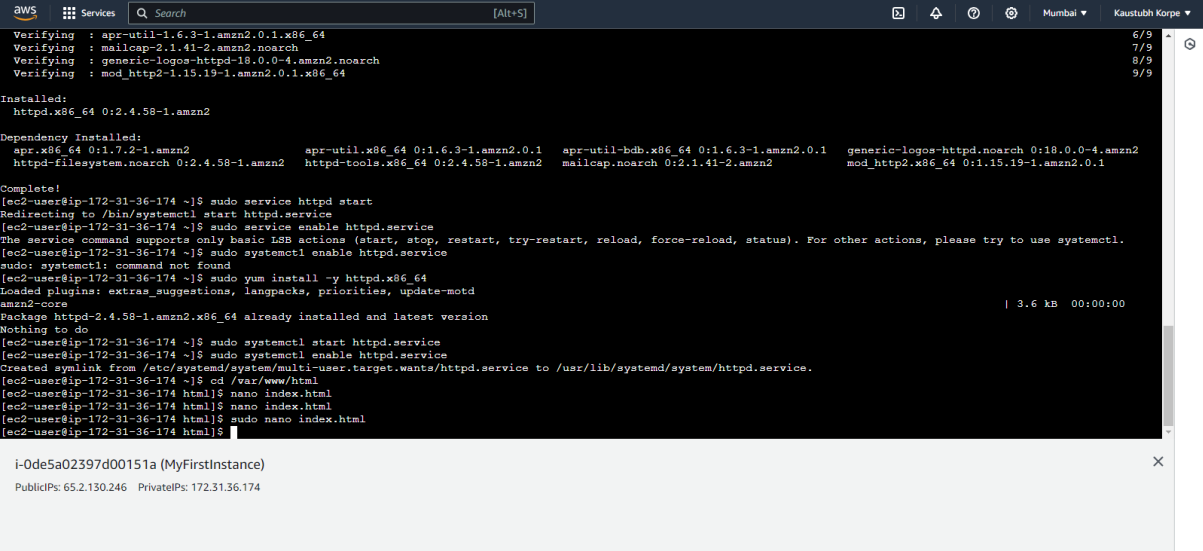
We have seen the default Apache page served by Apache. Let's modify that.

We will create our own index.html in the **Document Root** folder which is **/var/www/html** in our case.

**Document Root:** Document Root is the directory from which Apache looks for and serves web files on your request. So, we will create an index.html in **/var/www/html** folder.

Use the *nano* command to create the file and put content into it.

```
sudo nano index.html
```



```
Verifying : apr-util-1.6.3-1.amzn2.0.1.x86_64
Verifying : mailcap-2.1.41-2.amzn2.noarch
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch
Verifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64

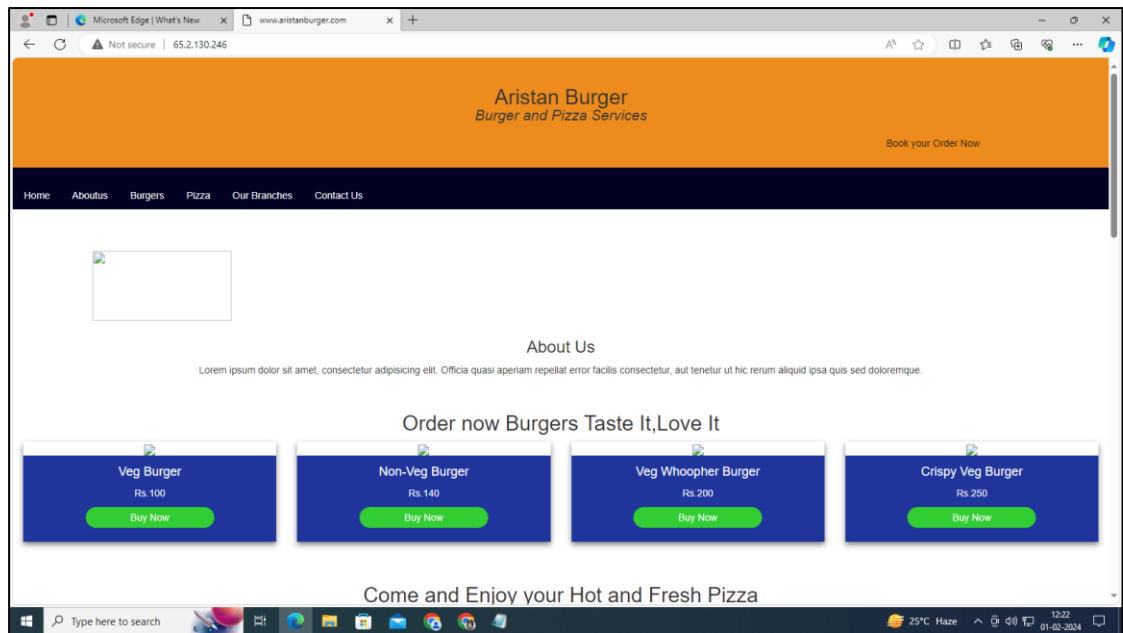
Installed:
  httpd.x86_64 0:2.4.58-1.amzn2

Dependency Installed:
  apr.x86_64 0:1.7.2-1.amzn2          apr-util.x86_64 0:1.6.3-1.amzn2.0.1  apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1  generic-logos-httpd.noarch 0:18.0.0-4.amzn2
  httpdfilesystem.noarch 0:2.4.58-1.amzn2  httpd-tools.x86_64 0:2.4.58-1.amzn2      mailcap.noarch 0:2.1.41-2.amzn2      mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
[ec2-user@ip-172-31-36-174 ~]$ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[ec2-user@ip-172-31-36-174 ~]$ sudo service enable httpd.service
The service command supports only basic LSB actions (start, stop, restart, try-restart, reload, force-reload, status). For other actions, please try to use systemctl.
[ec2-user@ip-172-31-36-174 ~]$ sudo systemctl enable httpd.service
sudo: systemctl: command not found
[ec2-user@ip-172-31-36-174 ~]$ sudo yum install -y httpd.x86_64
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package httpd-2.4.58-1.amzn2.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-36-174 ~]$ sudo systemctl start httpd.service
[ec2-user@ip-172-31-36-174 ~]$ sudo systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-36-174 ~]$ cd /var/www/html
[ec2-user@ip-172-31-36-174 html]$ nano index.html
[ec2-user@ip-172-31-36-174 html]$ sudo nano index.html
[ec2-user@ip-172-31-36-174 html]$
```

Save and Close File.

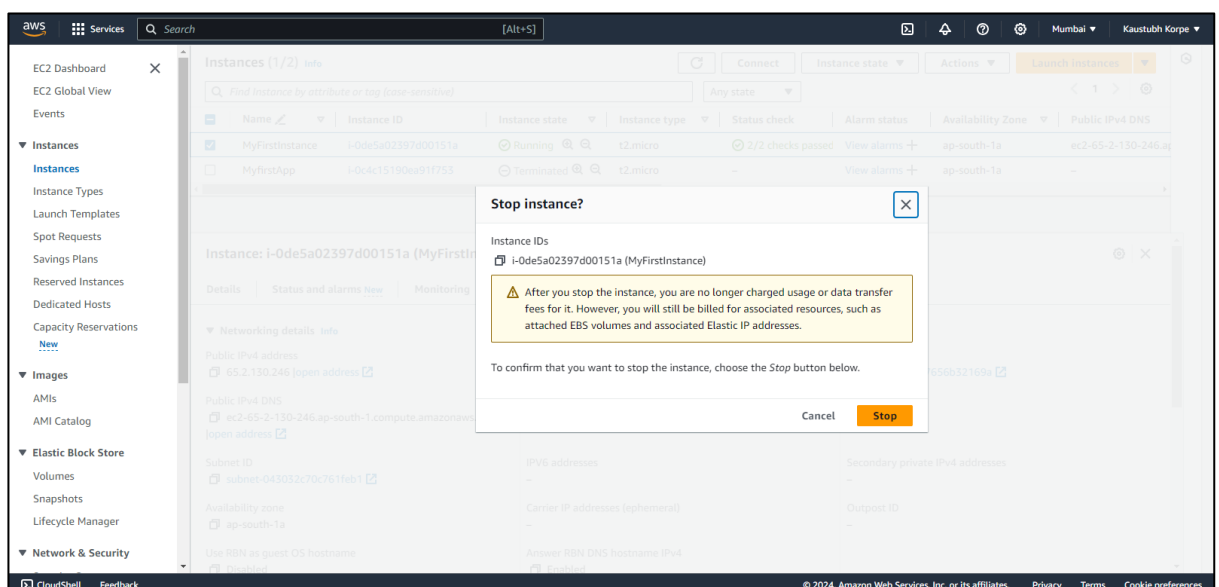
## 15. View the customized web page: This time when you enter **public IP** or DNS into the browser, you will see your customized page



## Stopping the EC2 Instance

To stop the EC2 Instance click the stop Instance option from the console by clicking the instance id.

Following window or dialog box will appear to stop the instance.



By clicking the stop option, the instance will stop working until we launch it again.