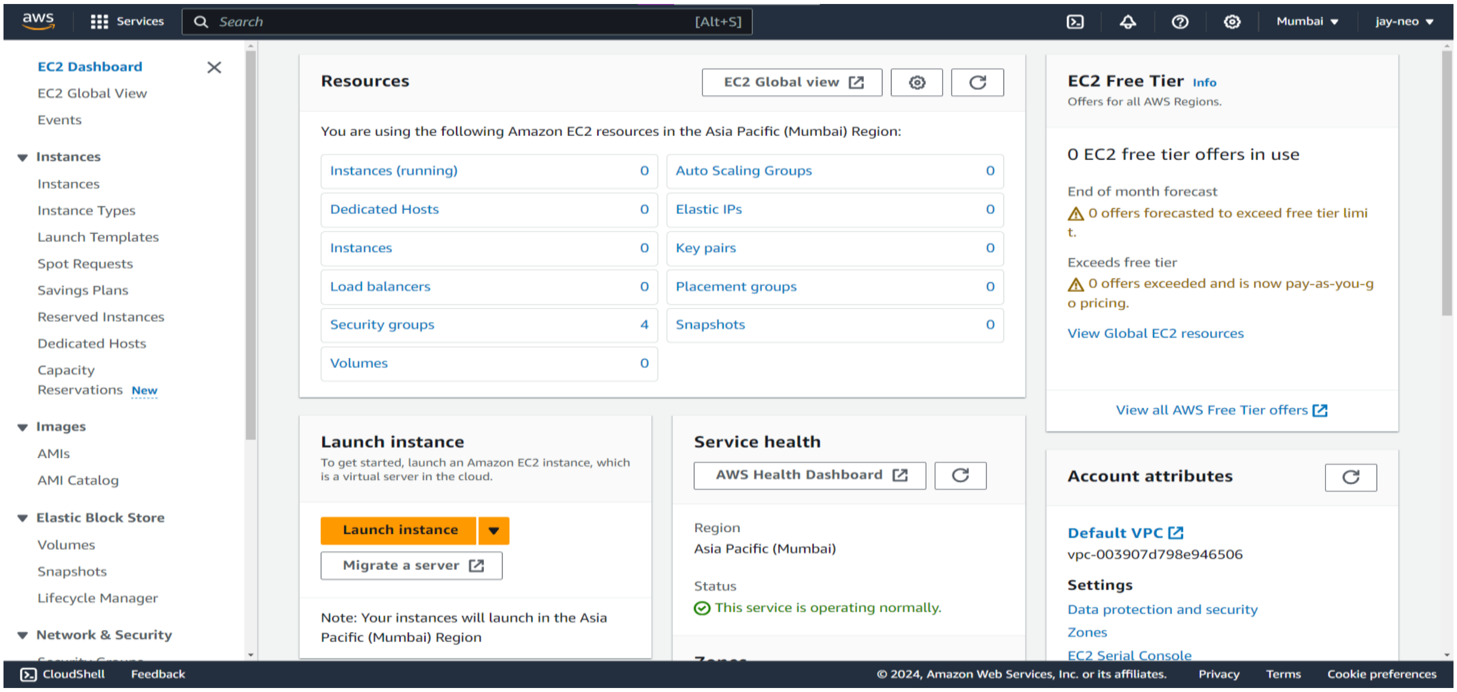
Assignment 7

Problem Statement: Host a website on EC2 service of AWS.

Initializing an EC2 instance

Step 1: Login to the AWS console, and search for EC2. Open the first EC2 link. We are directed to a page, where we can view details of all the instances we create, or are running.



Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	0	Auto Scaling Groups	0
Dedicated Hosts	0	Elastic IPs	0
Instances	0	Key pairs	0
Load balancers	0	Placement groups	0
Security groups	4	Snapshots	0
Volumes	0		

EC2 Free Tier Info

Offers for all AWS Regions.

0 EC2 free tier offers in use

End of month forecast

⚠ 0 offers forecasted to exceed free tier limit.

Exceeds free tier

⚠ 0 offers exceeded and is now pay-as-you-go pricing.

View Global EC2 resources

View all AWS Free Tier offers

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the Asia Pacific (Mumbai) Region

Service health

Region
Asia Pacific (Mumbai)

Status
✓ This service is operating normally.

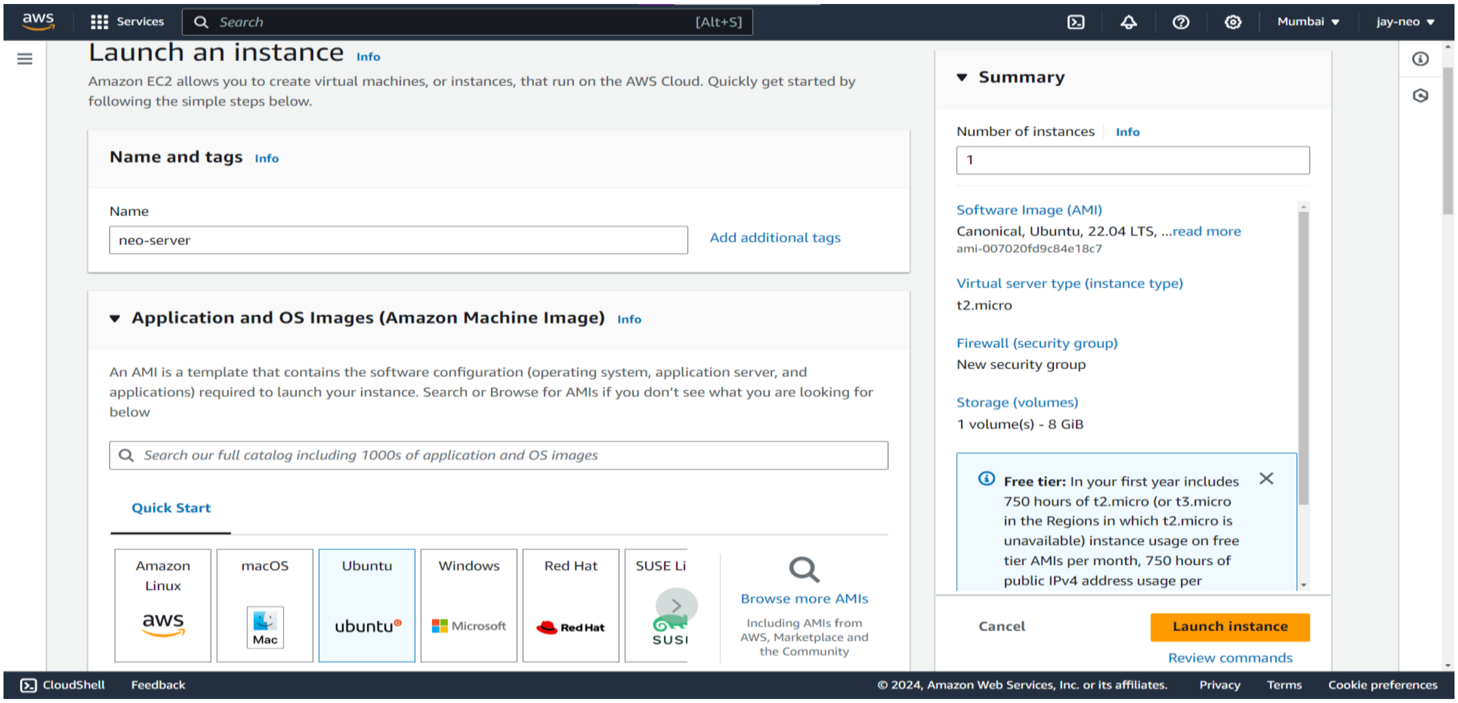
Account attributes

Default VPC
vpc-003907d798e946506

Settings
Data protection and security
Zones
EC2 Serial Console

© 2024, Amazon Web Services, Inc. or its affiliates.

Step 2: Click on the Launch Instance button. On the new window that opens, set a name for the instance, and select the Ubuntu platform.



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: neo-server

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Summary

Number of instances: 1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-007020fd9c84e18c7

Virtual server type (instance type)
t2.micro

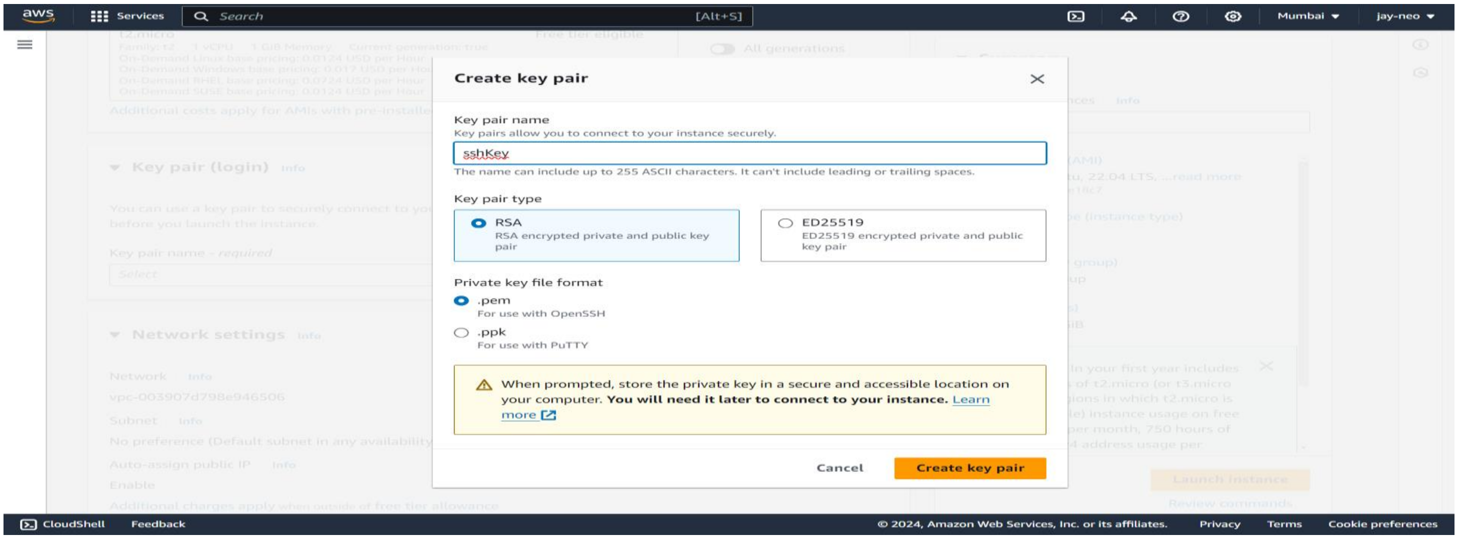
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per

© 2024, Amazon Web Services, Inc. or its affiliates.

Step 3: Create a new key pair, if does not exist already. For this click on Create new key pair.



Create key pair

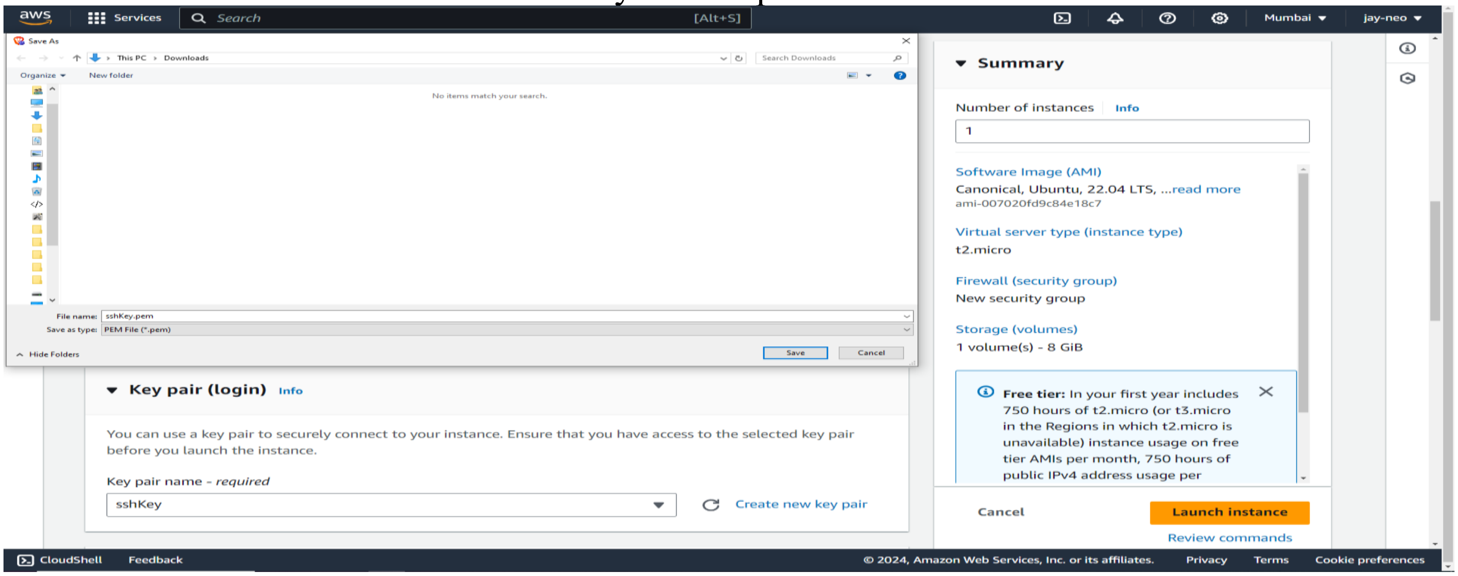
Key pair name
Key pairs allow you to connect to your instance securely.
sshKey
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type
● RSA
RSA encrypted private and public key pair
○ ED25519
ED25519 encrypted private and public key pair

Private key file format
● .pem
For use with OpenSSH
○ .ppk
For use with PuTTY

When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Step 4: The new popup appears, where we enter the name of the key pair, and leave all other options default (RSA key pair type, and the '.pem' extension).Then click on the Create key pair button. Save the file in **~/Downloads** on your computer.



File name: sshKey.pem
Save as type: PEM File (*.pem)

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*
sshKey
Create new key pair

Summary

Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-007020fd9c84e18c7

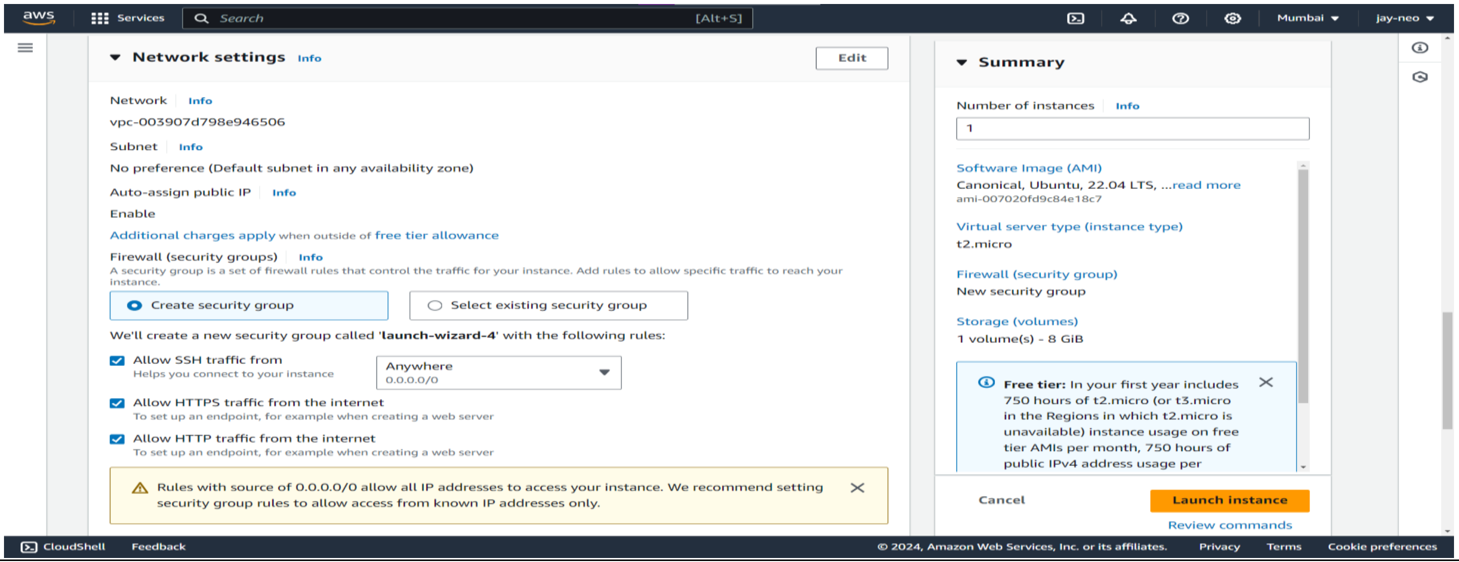
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per

Step 5: Then in **Network Settings**, Create security group and check all the checkboxes.



Network settings Info

Network Info
vpc-003907d798e946506

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-4**' with the following rules:

☑ Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0

☑ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

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.

Summary

Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-007020fd9c84e18c7

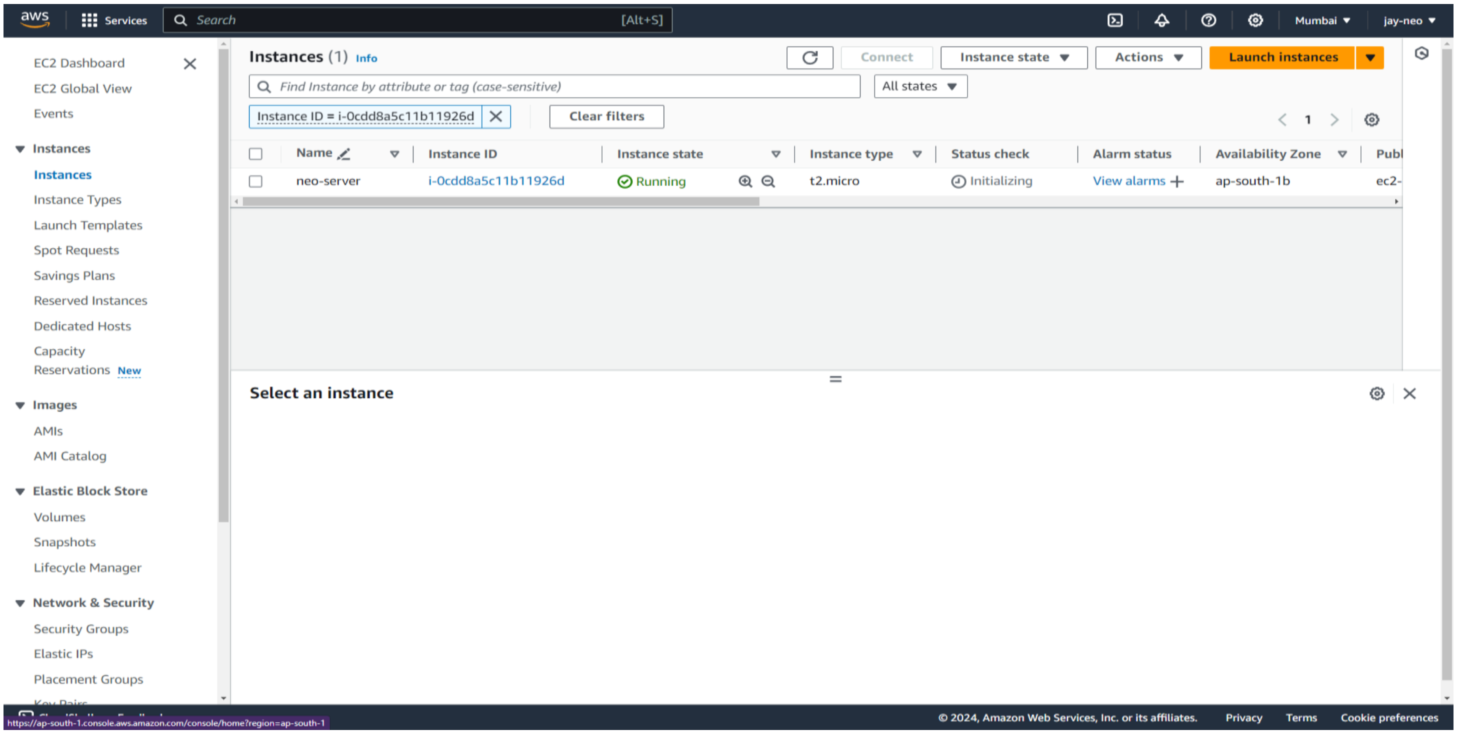
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per

Step 6: Leave all other options default and click on Launch instance. A success message is shown on a new window. Click on View all instances. After, a window showing all instances is open.

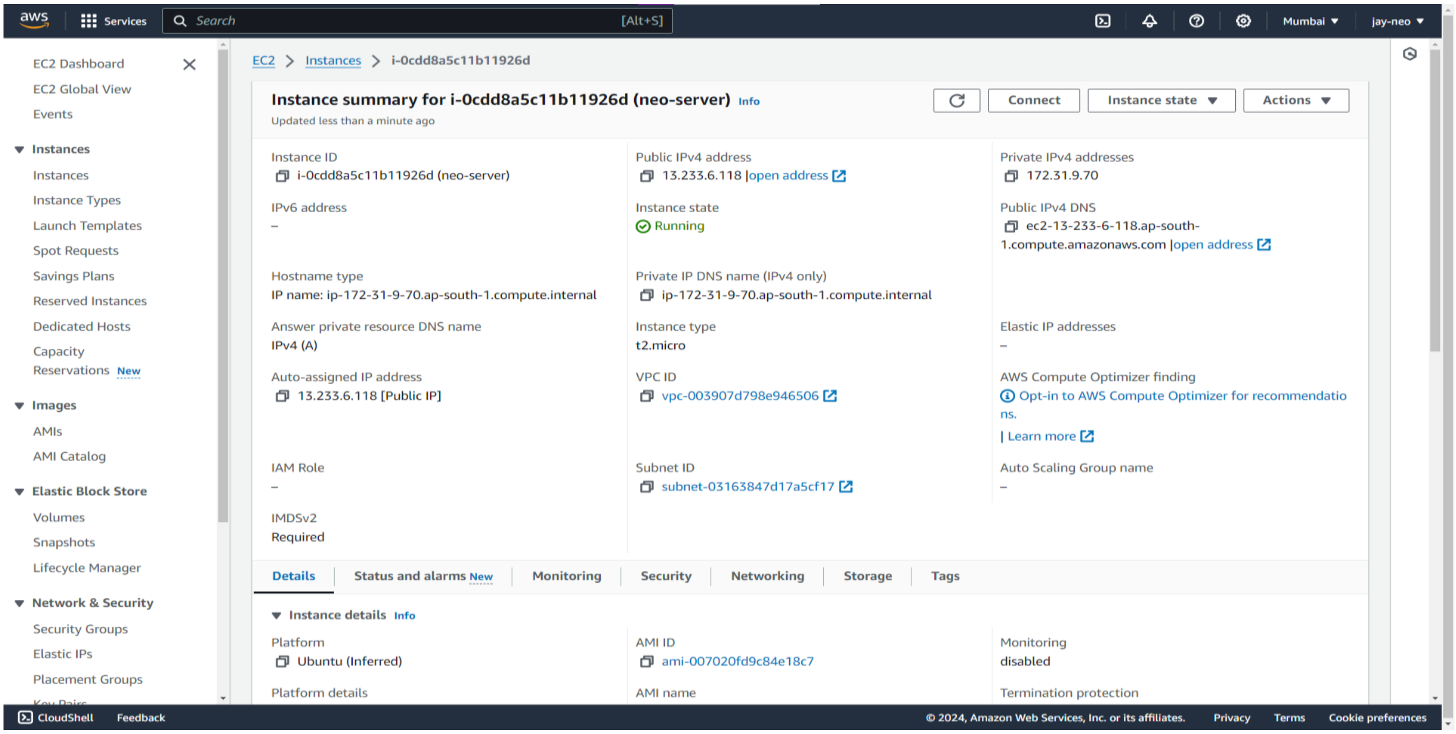


Instances (1) Info

Instance ID = i-0cdd8a5c11b11926d

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
□	neo-server	i-0cdd8a5c11b11926d	✓ Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-

Step 7: Now click on the Instance ID (hidden in the image, but it is actually visible on the website). Copy the public IPv4 address as visible from the new interface that opens.



Instance summary for i-0cdd8a5c11b11926d (neo-server) Info

Updated less than a minute ago

Instance ID: i-0cdd8a5c11b11926d (neo-server)

Public IPv4 address: 13.233.6.118 |open address

Private IPv4 addresses: 172.31.9.70

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-13-233-6-118.ap-south-1.compute.amazonaws.com |open address

Hostname type: IP name: ip-172-31-9-70.ap-south-1.compute.internal

Private IP DNS name (IPv4 only): ip-172-31-9-70.ap-south-1.compute.internal

Answer private resource DNS name: IPv4 (A)

Instance type: t2.micro

Elastic IP addresses: –

Auto-assigned IP address: 13.233.6.118 [Public IP]

VPC ID: vpc-003907d798e946506

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role: –

Subnet ID: subnet-03163847d17a5cf17

Auto Scaling Group name: –

IMDSv2: Required

Instance details Info

Platform: Ubuntu (Inferred)

AMI ID: ami-007020fd9c84e18c7

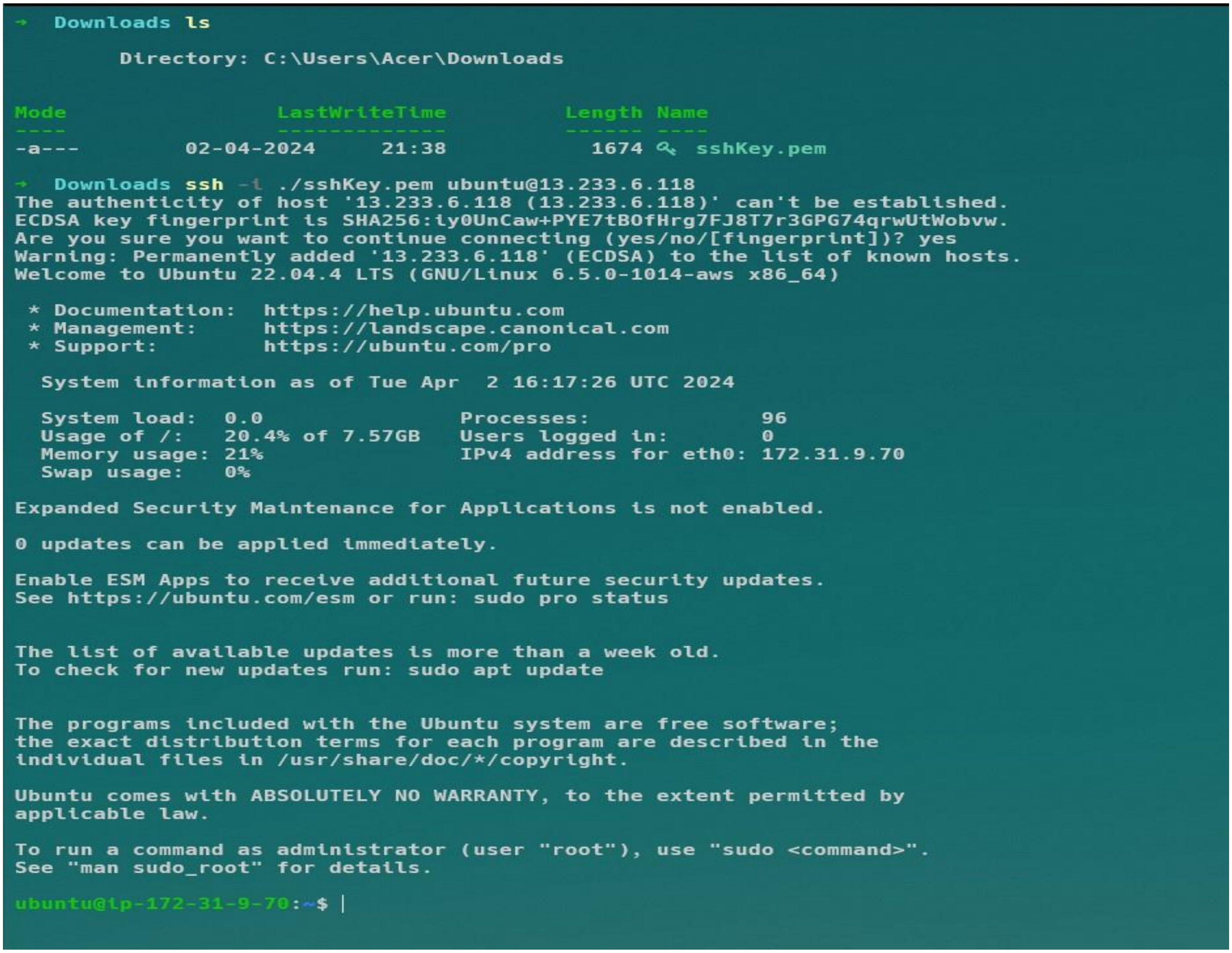
Monitoring: disabled

Platform details

AMI name

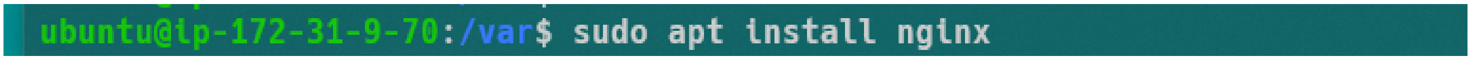
Termination protection

Step 8: Now, our EC2 service is ready to work with it. Firstly, to connect EC2 instance from out local machine, open Windows Terminal where shell is powershell and connect using ssh command. Here pass the ssh private key file using **-i** flag with ssh command.



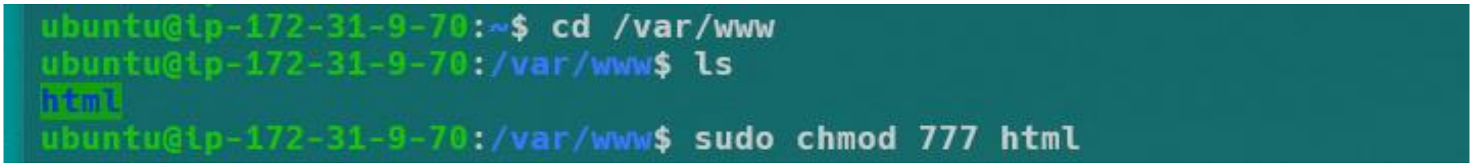
```
→  Downloads ls

        Directory: C:\Users\Acer\Downloads

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---         02-04-2024     21:38           1674   sshKey.pem

→  Downloads ssh -i ./sshKey.pem ubuntu@13.233.6.118
The authenticity of host '13.233.6.118 (13.233.6.118)' can't be established.
ECDSA key fingerprint is SHA256:iy0UnCaw+PYE7tBOfHrg7FJ8T7r3GPG74qrwUtWobvw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.233.6.118' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1014-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Tue Apr  2 16:17:26 UTC 2024

  System load:  0.0               Processes:             96
  Usage of /:   20.4% of 7.57GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for eth0: 172.31.9.70
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-9-70:~$ |
```

Step 9: Download **ngnix** package using apt package manager with super user permission.

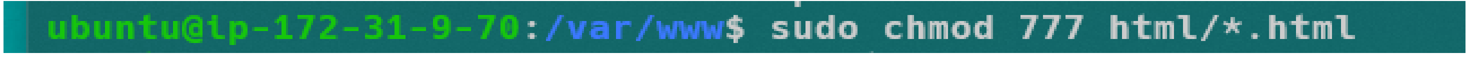


```
ubuntu@ip-172-31-9-70:/var$ sudo apt install nginx
```

Step 10: Give the full permission of **/var/www/html** folder and all html file inside it.

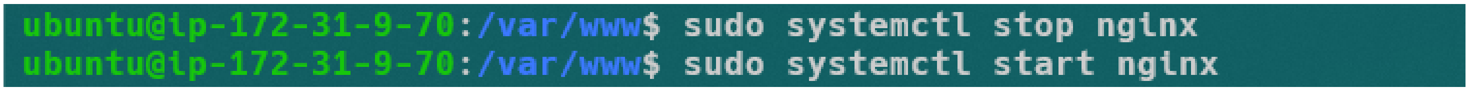


```
ubuntu@ip-172-31-9-70:~$ cd /var/www
ubuntu@ip-172-31-9-70:/var/www$ ls
html
ubuntu@ip-172-31-9-70:/var/www$ sudo chmod 777 html
```



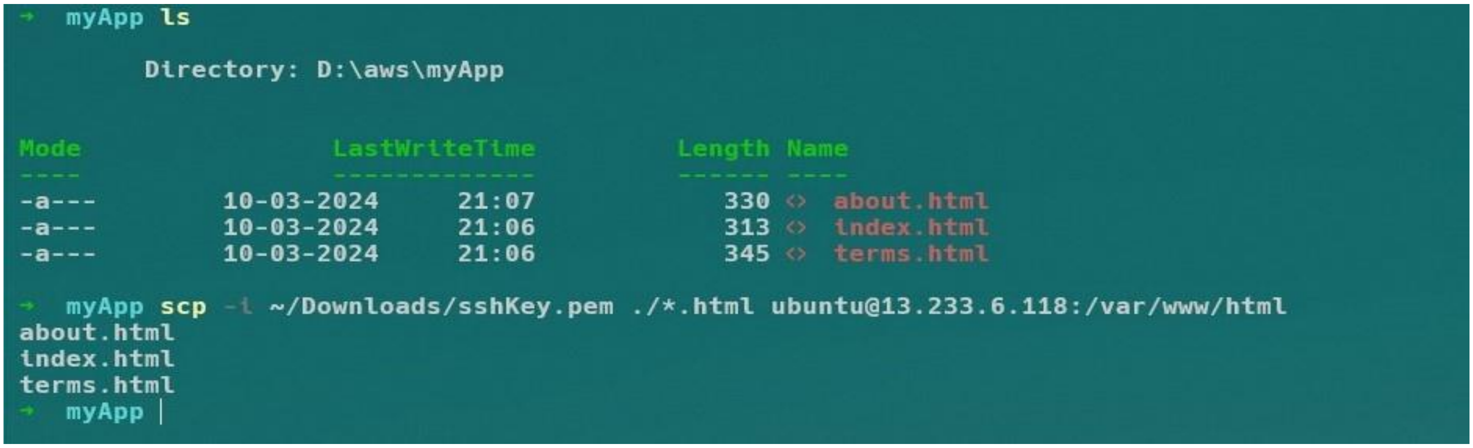
```
ubuntu@ip-172-31-9-70:/var/www$ sudo chmod 777 html/*.html
```

Step 11: If nginx running or dead, we restart the nginx using **systemctl** command.



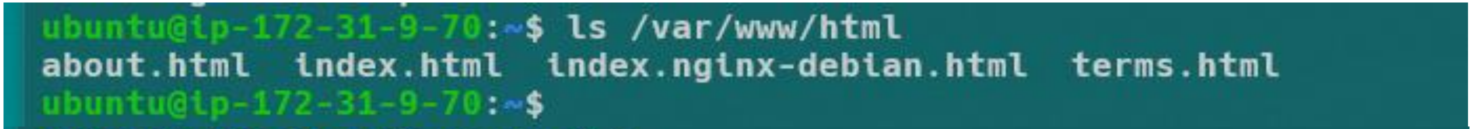
```
ubuntu@ip-172-31-9-70:/var/www$ sudo systemctl stop nginx
ubuntu@ip-172-31-9-70:/var/www$ sudo systemctl start nginx
```

Step 12: Now, from our local machine copy web application code from local machine to EC2 instance using **scp** command. Here we reuse the html files that are already created from Assignment 6, that are index.html, about.html, terms.html. Open another Windows Terminal from our local machine.

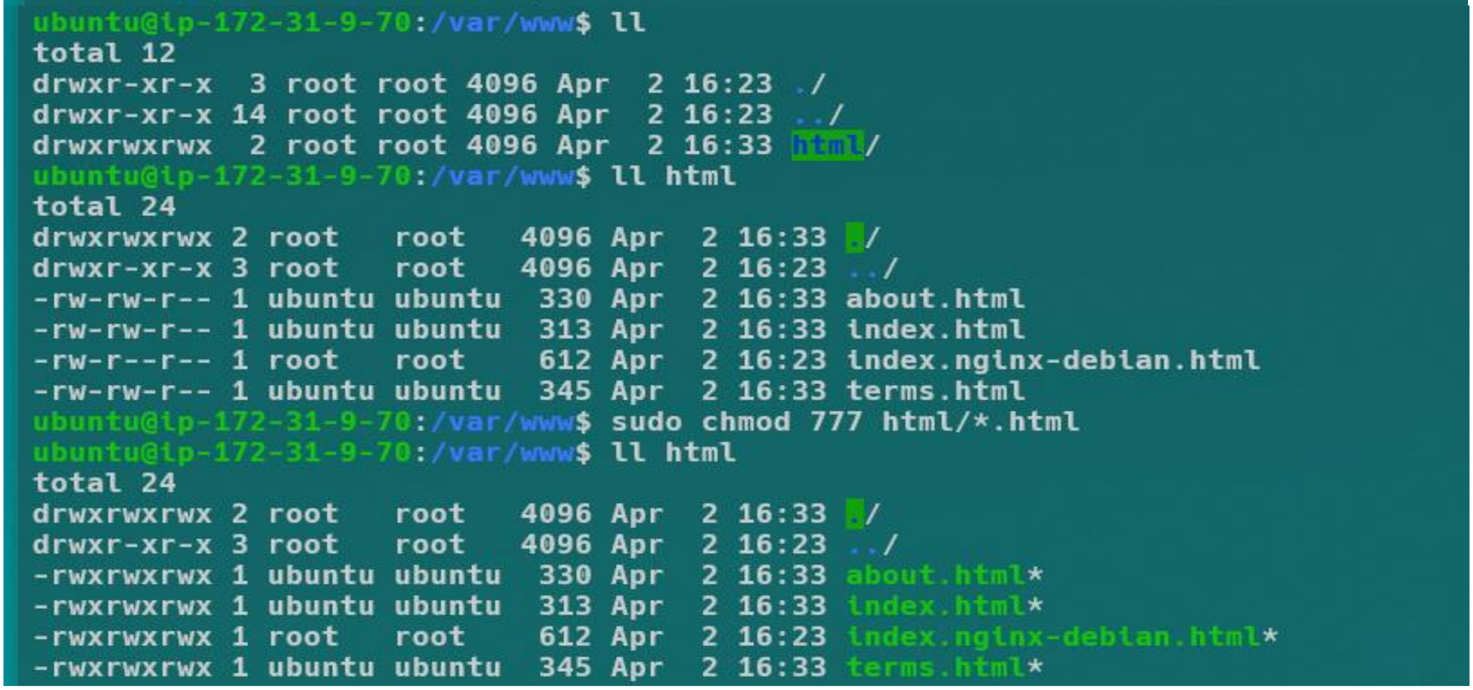


```
→  myApp ls

        Directory: D:\aws\myApp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---         10-03-2024     21:07            330 <> about.html
-a---         10-03-2024     21:06            313 <> index.html
-a---         10-03-2024     21:06            345 <> terms.html

→  myApp scp -i ~/Downloads/sshKey.pem ./*.html ubuntu@13.233.6.118:/var/www/html
about.html
index.html
terms.html
→  myApp |
```

Step 13: Now to go to EC2 instance see the **/var/www/html** directory. Give the all permission of all html file exist there.

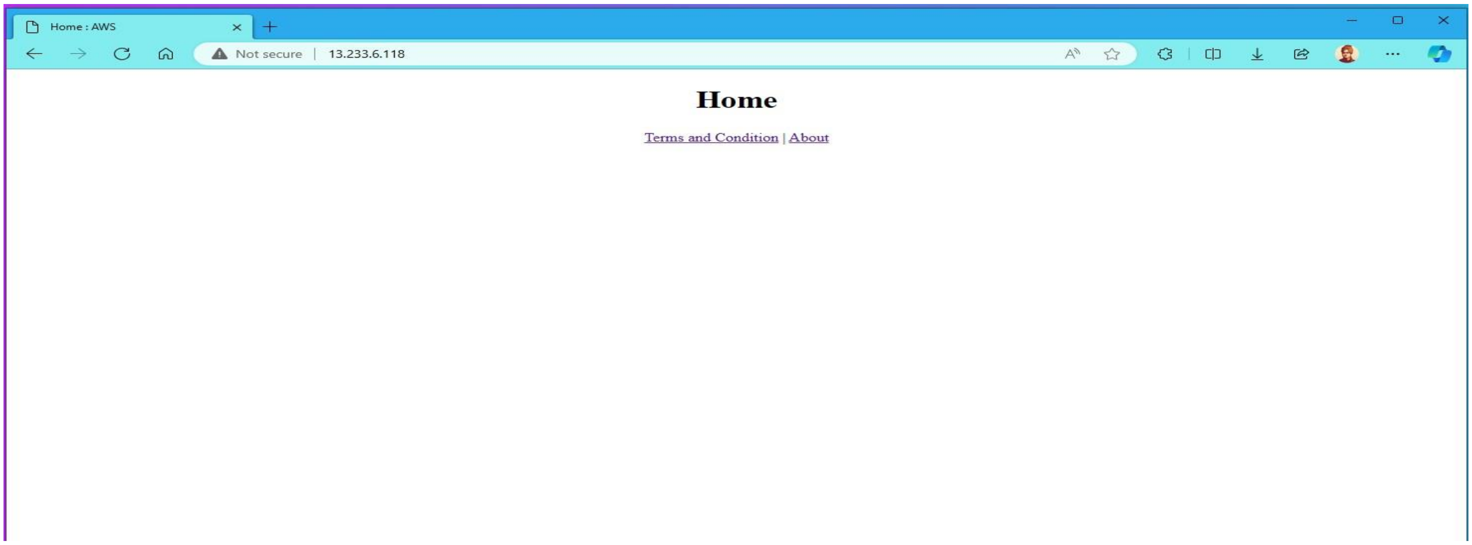


```
ubuntu@ip-172-31-9-70:~$ ls /var/www/html
about.html  index.html  index.nginx-debian.html  terms.html
ubuntu@ip-172-31-9-70:~$
```



```
ubuntu@ip-172-31-9-70:/var/www$ ll
total 12
drwxr-xr-x  3 root root 4096 Apr  2 16:23 ./
drwxr-xr-x 14 root root 4096 Apr  2 16:23 ../
drwxrwxrwx  2 root root 4096 Apr  2 16:33 html/
ubuntu@ip-172-31-9-70:/var/www$ ll html
total 24
drwxrwxrwx 2 root   root   4096 Apr  2 16:33 ./
drwxr-xr-x 3 root   root   4096 Apr  2 16:23 ../
-rw-rw-r-- 1 ubuntu ubuntu  330 Apr  2 16:33 about.html
-rw-rw-r-- 1 ubuntu ubuntu  313 Apr  2 16:33 index.html
-rw-r--r-- 1 root   root    612 Apr  2 16:23 index.nginx-debian.html
-rw-rw-r-- 1 ubuntu ubuntu  345 Apr  2 16:33 terms.html
ubuntu@ip-172-31-9-70:/var/www$ sudo chmod 777 html/*.html
ubuntu@ip-172-31-9-70:/var/www$ ll html
total 24
drwxrwxrwx 2 root   root   4096 Apr  2 16:33 ./
drwxr-xr-x 3 root   root   4096 Apr  2 16:23 ../
-rwxrwxrwx 1 ubuntu ubuntu  330 Apr  2 16:33 about.html*
-rwxrwxrwx 1 ubuntu ubuntu  313 Apr  2 16:33 index.html*
-rwxrwxrwx 1 root   root    612 Apr  2 16:23 index.nginx-debian.html*
-rwxrwxrwx 1 ubuntu ubuntu  345 Apr  2 16:33 terms.html*
```

Step 14: Now finally go to the public IP of our EC2 instance to see the web application.



Home

Terms and Condition | About