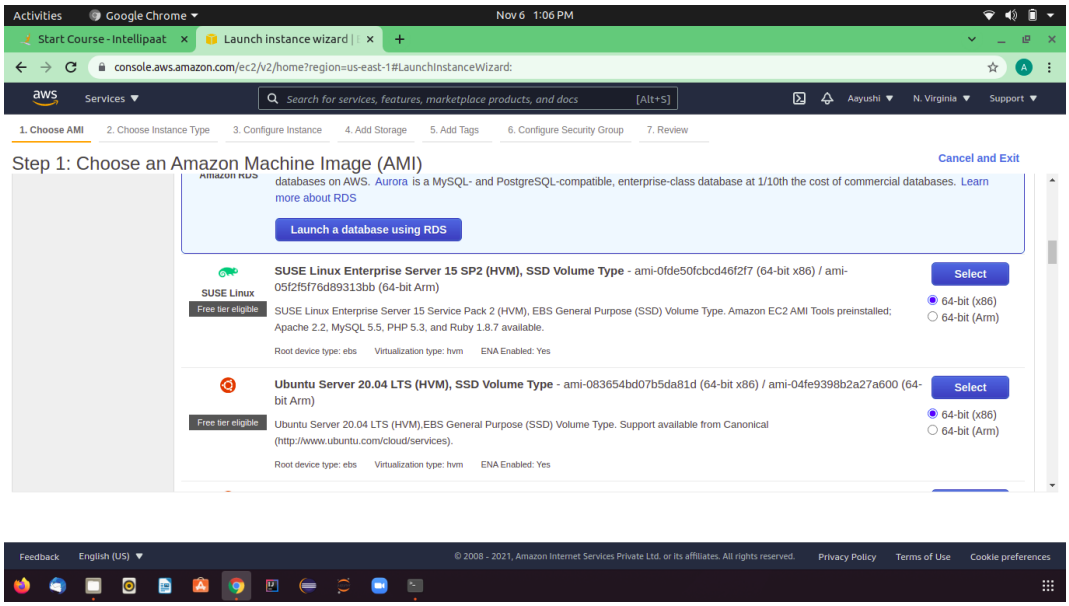
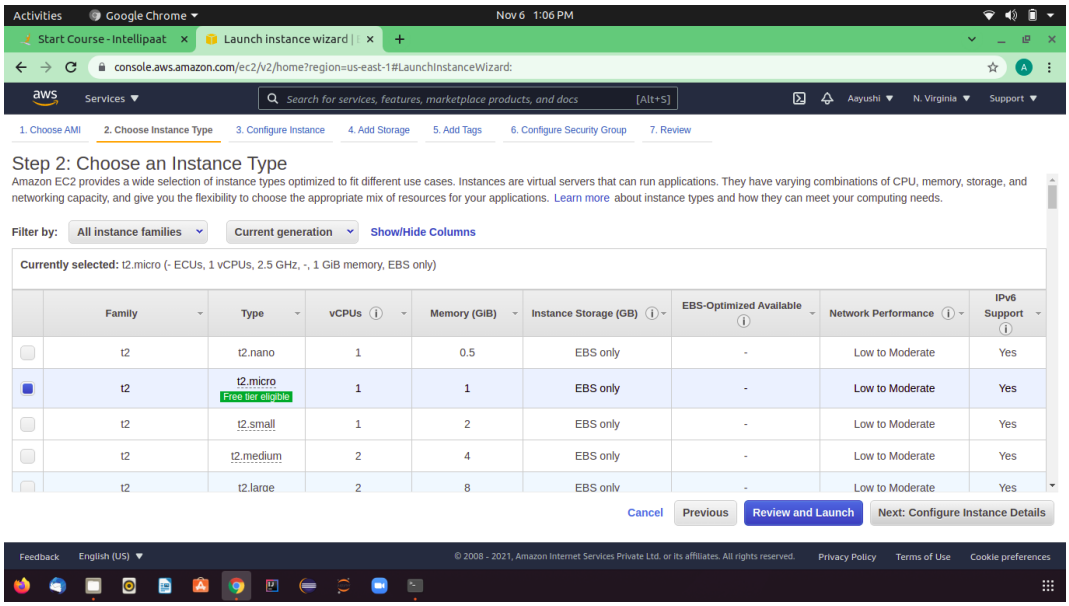
EC2 Assignment 1:

Create an instance in us-east-1a(N.Virginia) with an Ubuntu OS with website Hello World

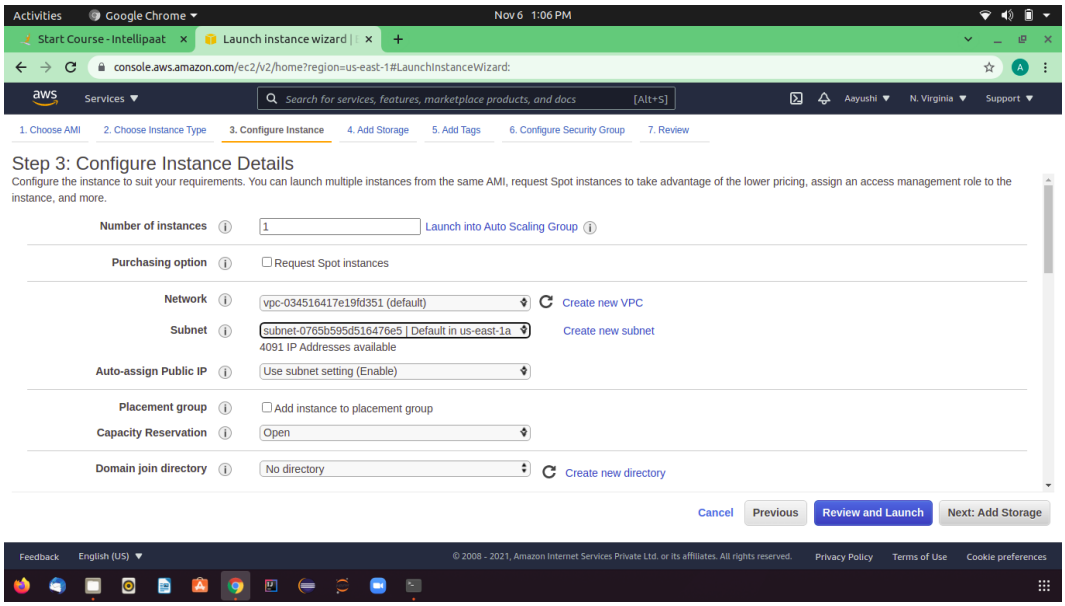
Create an instance:



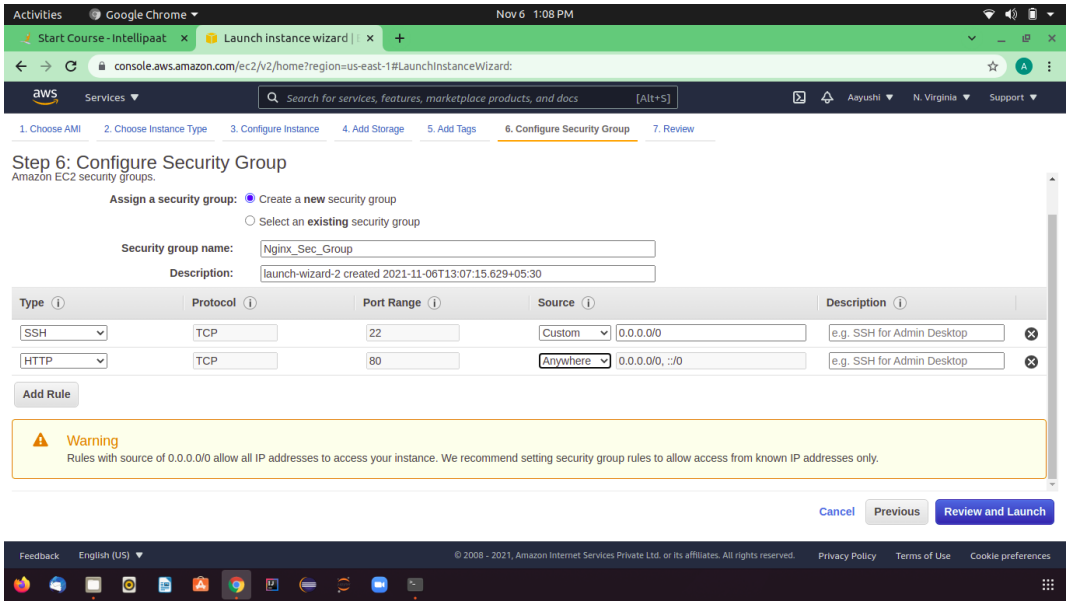
Choose t2.micro as instance type



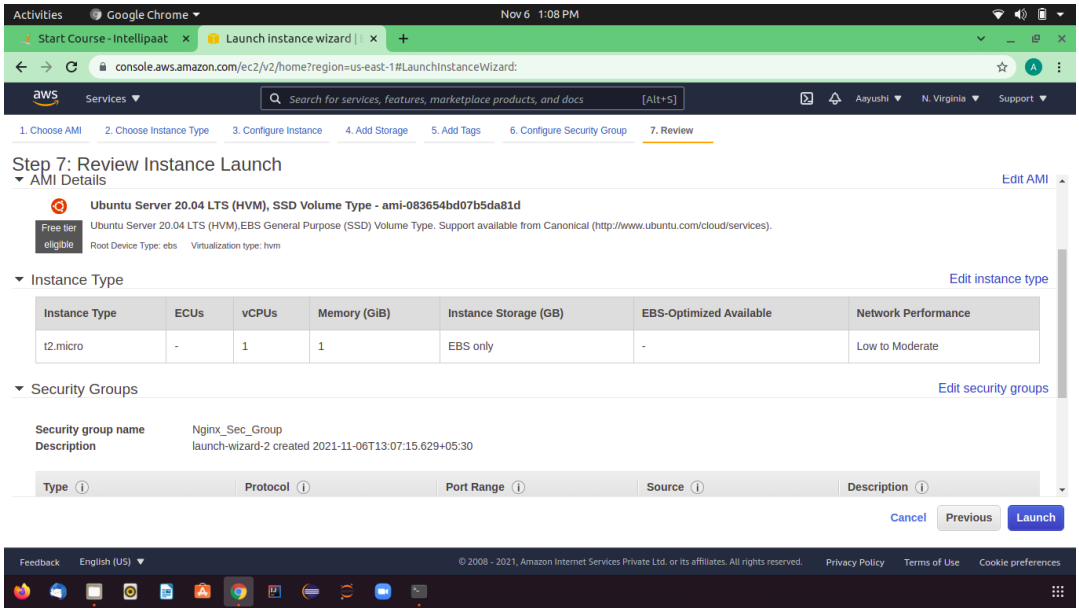
Configure the Instance Details
Select us-east-1a



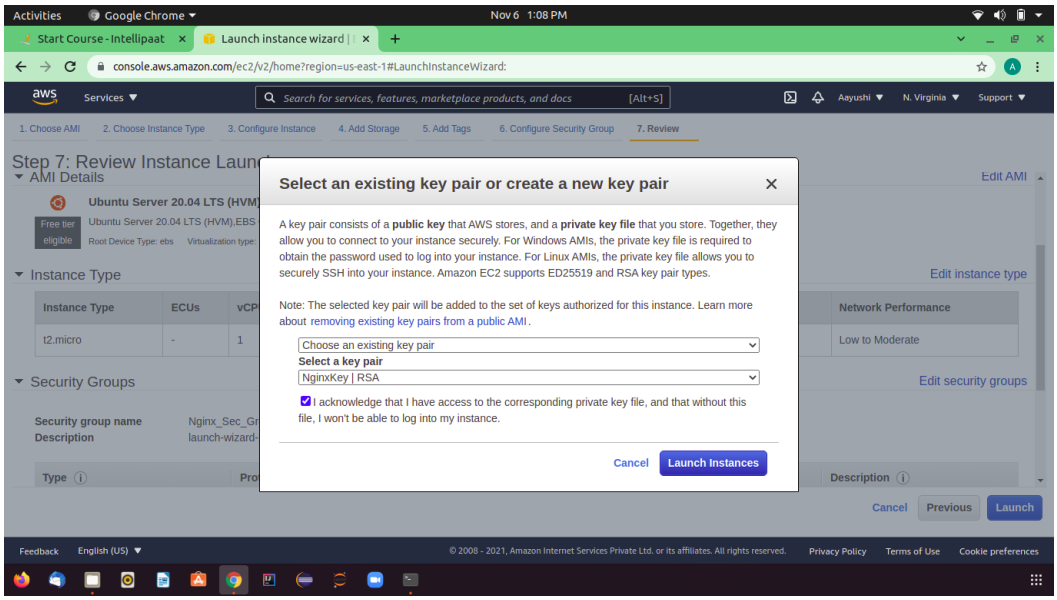
Add security group



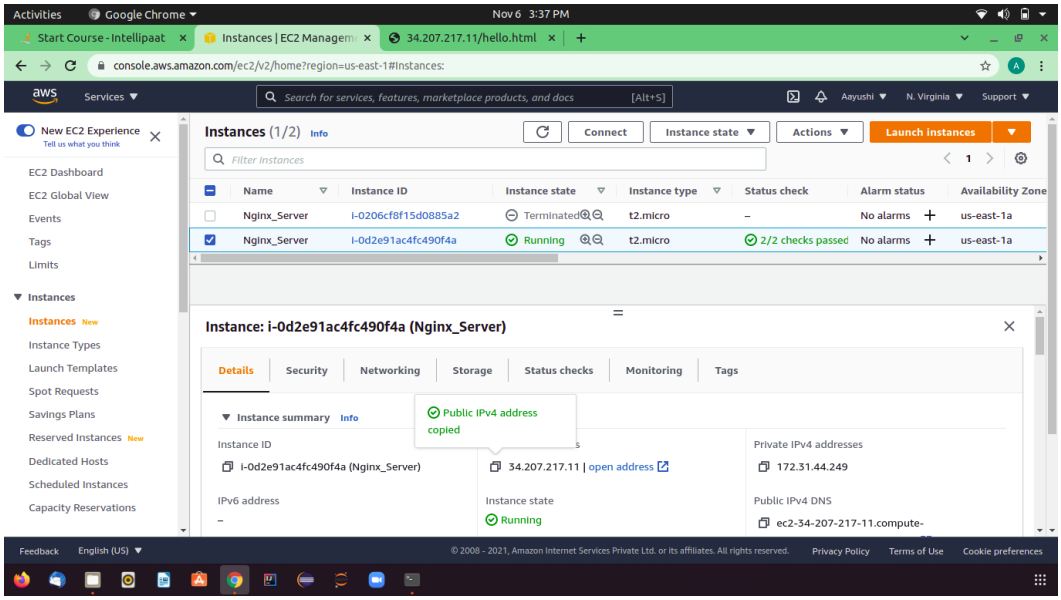
Now take review



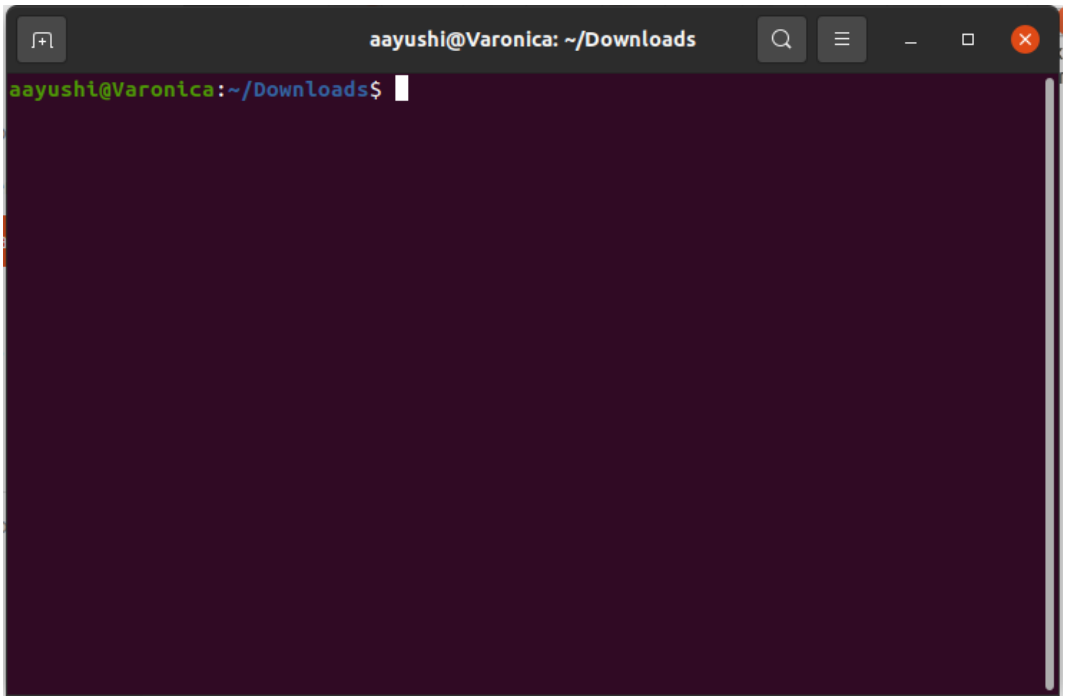
Create key pair and download key



Launch the instance



Go to the command prompt



Now go to the root using sudo su command

Now get the permission of private key
Using command chmod 400 Key name.pem

```
s# chmod 400 NginxKey.pem
```

This is the explanation of the SSH command:

- ssh: Command to use SSH protocol
- -i: Flag that specifies an alternate identification file to use for public key authentication.
- username: Username that uses your instance
- ip-address: IP address given to your instance

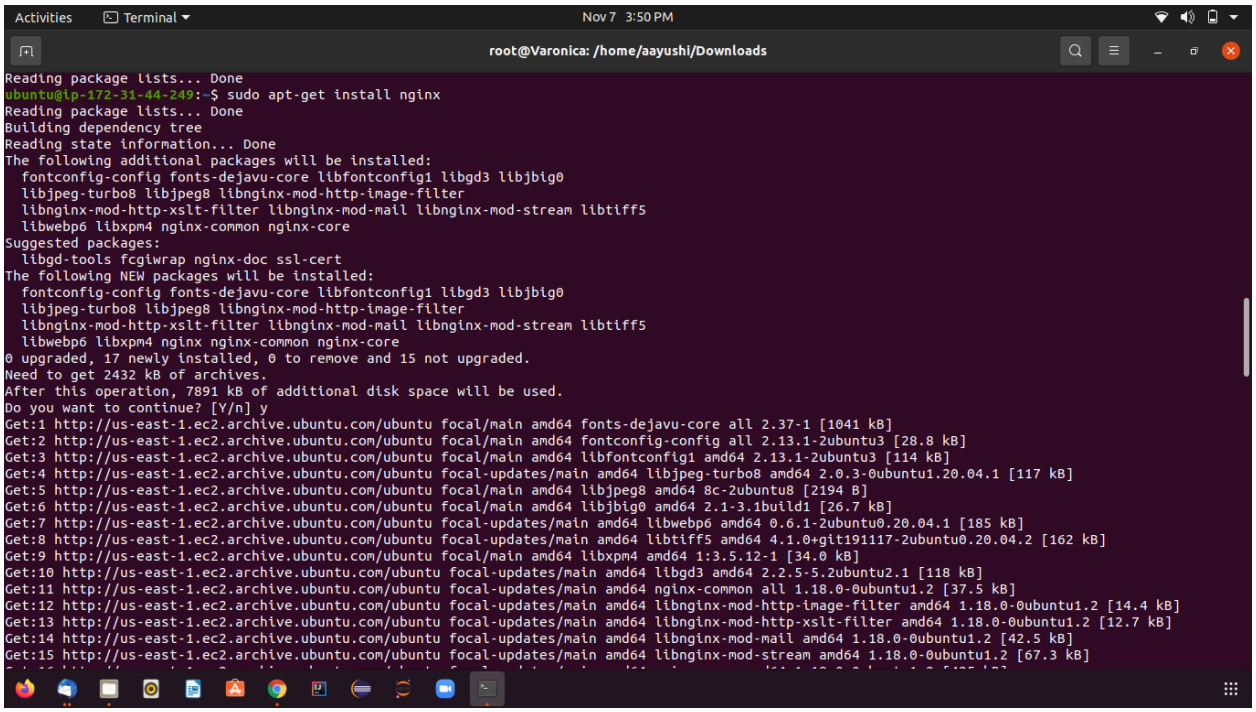
Using command `ssh -i key name.pem username@Public-ip-address`

```
root@Varonica:/home/aayushi/Downloads# ssh -i "NginxKey.pem" ubuntu
@ec2-34-207-217-11.compute-1.amazonaws.com
The authenticity of host 'ec2-34-207-217-11.compute-1.amazonaws.com
 (34.207.217.11)' can't be established.
ECDSA key fingerprint is SHA256:zwCEEKP5DCEEfPJgYgvf8+ffBeUTdnk/B7f
SoCOzMUY.
Are you sure you want to continue connecting (yes/no/[fingerprint])
? yes
Warning: Permanently added 'ec2-34-207-217-11.compute-1.amazonaws.c
om,34.207.217.11' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

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

  System information as of Sat Nov  6 09:55:51 UTC 2021

  System load:  0.21              Processes:             101
  Usage of /:   17.7% of 7.69GB   Users logged in:       0
```

Install the Nginx
Using command: sudo apt-get update
Sudo apt-get install nginx

```
:~$ sudo apt-get update
```

```
. Done
:~$ sudo apt-get install nginx
```



Adjust the firewall
By using: sudo ufw app list

```
ubuntu@ip-172-31-44-249:~$ sudo ufw app list
Available applications:
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
  OpenSSH
```

sudo ufw allow 'Nginx HTTP'

```
ubuntu@ip-172-31-44-249:~$ sudo ufw allow 'Nginx HTTP'
Rules updated
Rules updated (v6)
```

Checking the web server
Using command: systemctl status nginx

```
:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2021-11-06 07:49:27 UTC; 1min 25s ago
       Docs: man:nginx(8)
   Main PID: 1954 (nginx)
      Tasks: 2 (limit: 1154)
     Memory: 4.9M
     CGroup: /system.slice/nginx.service
             ├─1954 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─1955 nginx: worker process
```

```
ubuntu@ip-172-31-44-249:~$ curl -4 icanhazip.com
34.207.217.11
```

When you have your server's IP address, enter it into your browser's address bar:

http://your_server_ip

Setting up server blocks

Create the directory for as follows, using the -p flag to create any necessary parent directories:
sudo mkdir -p /var/www/html

```
:~$ sudo mkdir -p /var/www/html
```

Next, assign ownership of the directory with the $USER environment variable:
sudo chown -R $USER:$USER /var/www/htmt

```
:~$ sudo chown -R $USER:$USER /var/www/html
```

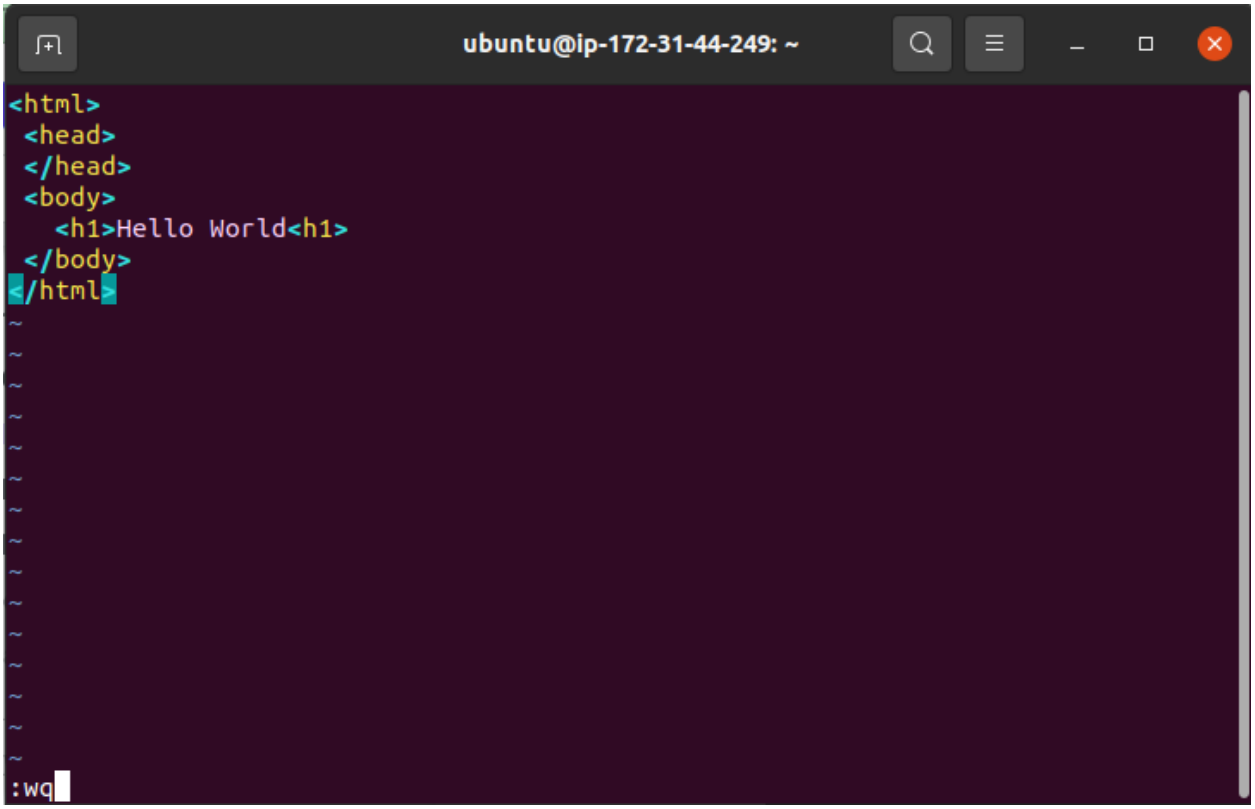
To ensure that your permissions are correct and allow the owner to read, write, and execute the files while granting only read and execute permissions to groups and others, you can input the following command:
sudo chmod -R 755 /var/www/html

```
ubuntu@ip-172-31-44-249:~$ sudo  chmod -R 755 /var/www/html
ubuntu@ip-172-31-44-249:~$
```

Next, create a file hello.html page using vi:
vi /var/www/html/hello.html

```
:~$
:~$ vi /var/www/html/hello.html
```

Inside, add the following sample of html:



In order for Nginx to serve this content, it's necessary to create a server block with the correct directives. Instead of modifying the default configuration file directly, let's make a new one at:
sudo nano /etc/nginx/sites-available/html

```
:~$ sudo vi /etc/nginx/sites-available/html
:~$
```

Paste in the following configuration block, which is similar to the default, but updated for our new directory and domain name:

```
server
{
        listen 80;
        listen [::]:80;

        root /var/www/html;
        hello hello.html hello.htm hello.nginx-debian.html;

        server_name html www.html;

        location /
          {
             try_files $uri $uri/ =404;
          }
}
```

sudo ln -s /etc/nginx/sites-available/html /etc/nginx/sites-enabled/

```
:~$
:~$ sudo ln -s /etc/nginx/sites-available/html /etc/nginx/sites-enabled/
```

sudo vi /etc/nginx/nginx.conf

```
:~$
:~$ sudo vi /etc/nginx/nginx.conf
```

Find the `server_names_hash_bucket_size` directive and remove the `#` symbol to uncomment the line. If you are using nano, you can quickly search for words in the file by pressing `CTRL` and `w`.

Next, test to make sure that there are no syntax errors in any of your Nginx files:
sudo nginx -t

```
ubuntu@ip-172-31-44-249:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

If there aren’t any problems, restart Nginx to enable your changes:
sudo systemctl restart nginx

```
ubuntu@ip-172-31-44-249:~$
ubuntu@ip-172-31-44-249:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-44-249:~$
```

Goto the Web server
http://34.207.217.11/hello.html