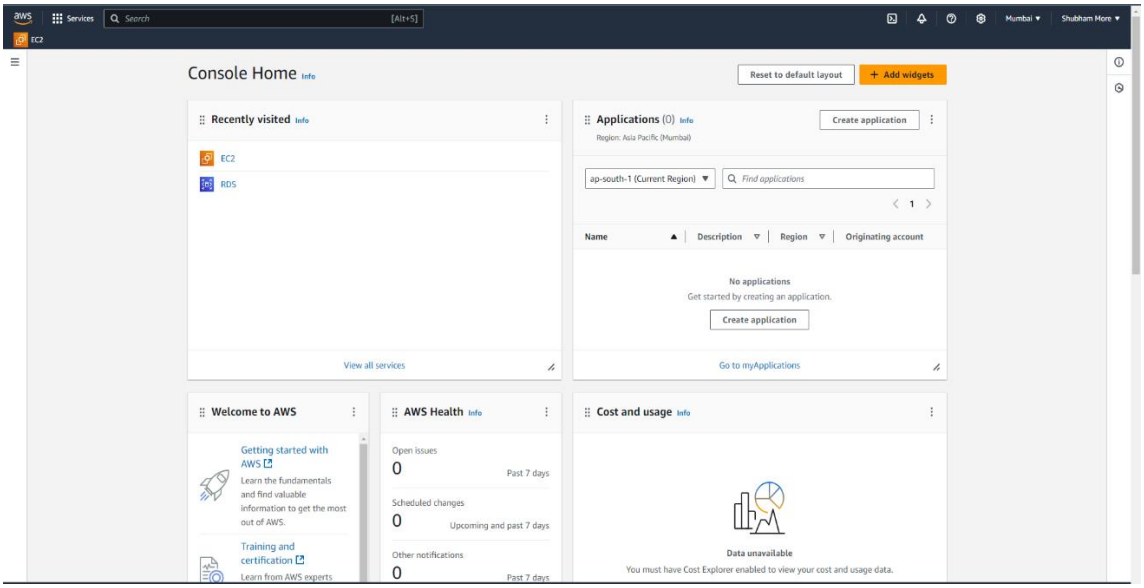
PRACTICAL - 5

Aim: Working in AWS to demonstrate to provisioning and Scaling of a website.

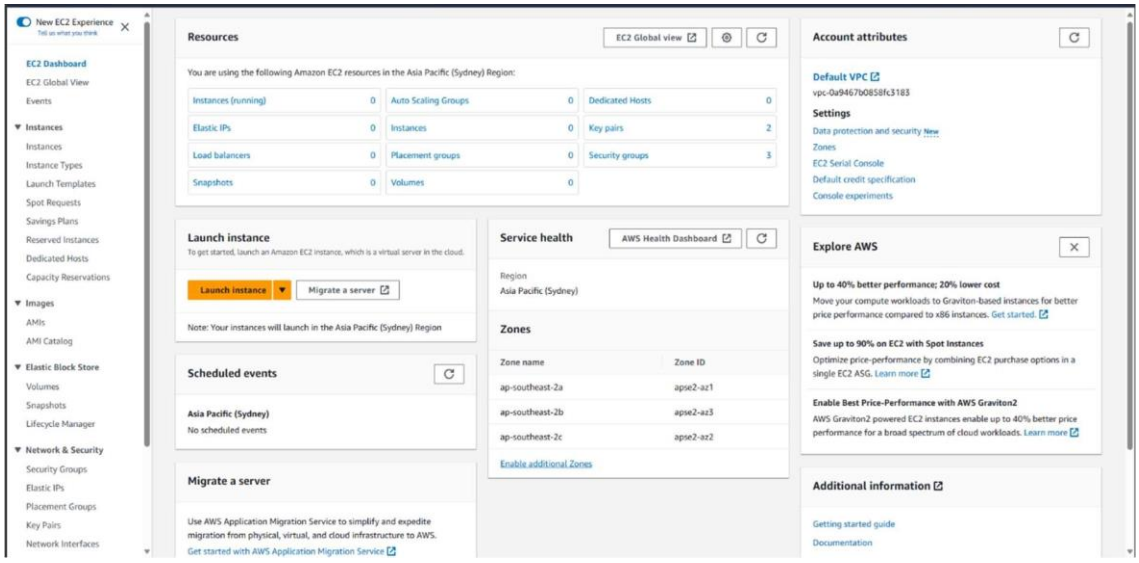
Practical Requirements:

i) AWS (EC2)
ii) Bitvise SSH Server (s/w)

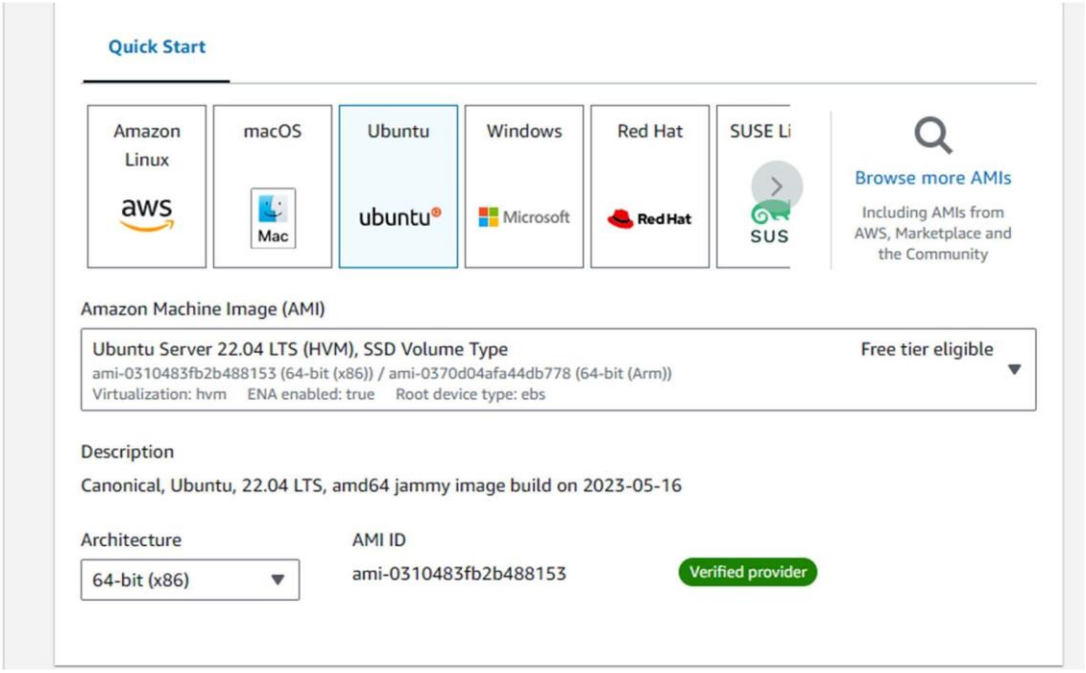
Step 1: Open the AWS Management Console and select EC2 service.



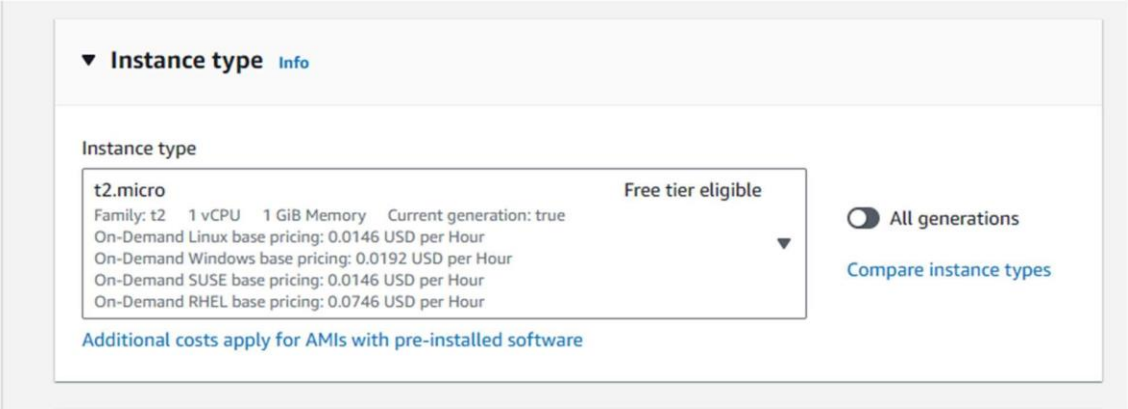
Step 2 : Click on Launch instance button to create a new instance.



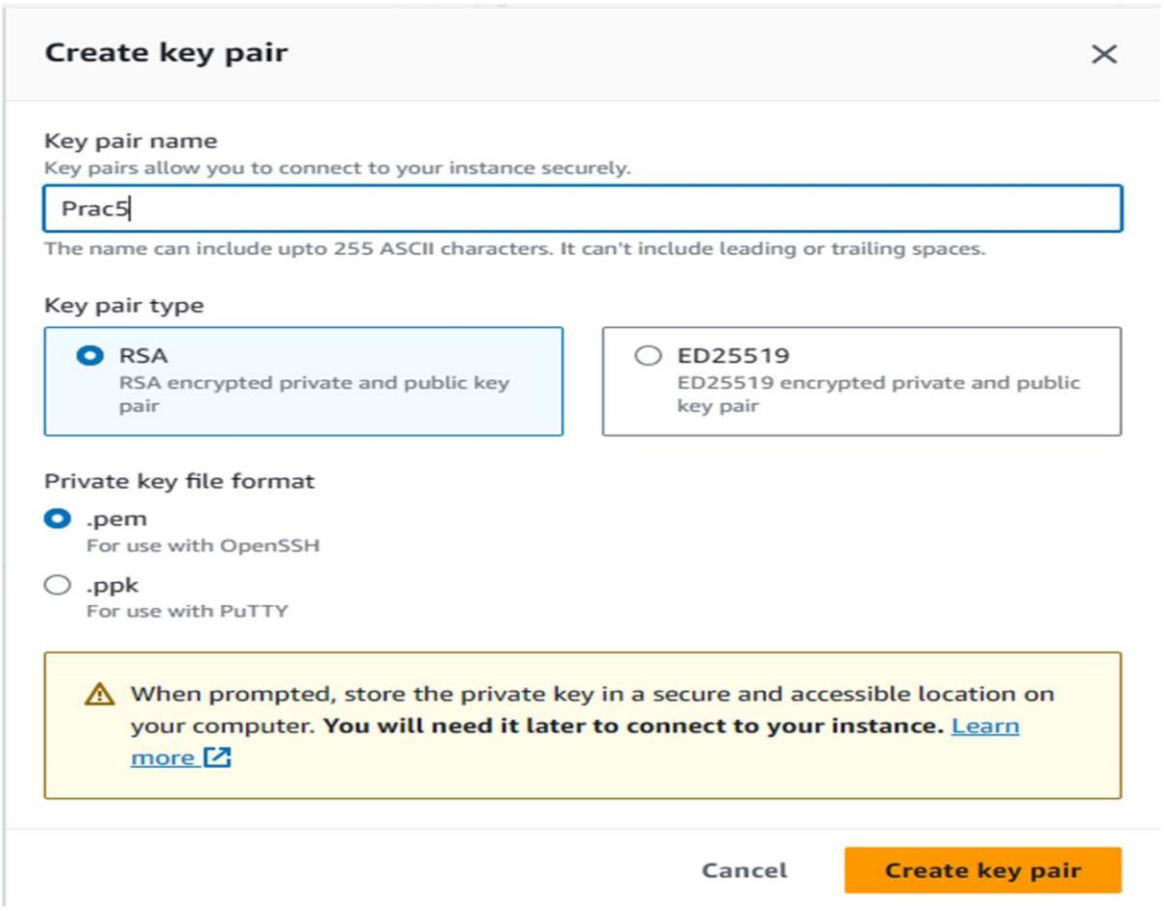
Step 3: Select the ubuntu server 22.04 LTS (HVM) to install ubuntu on instance.



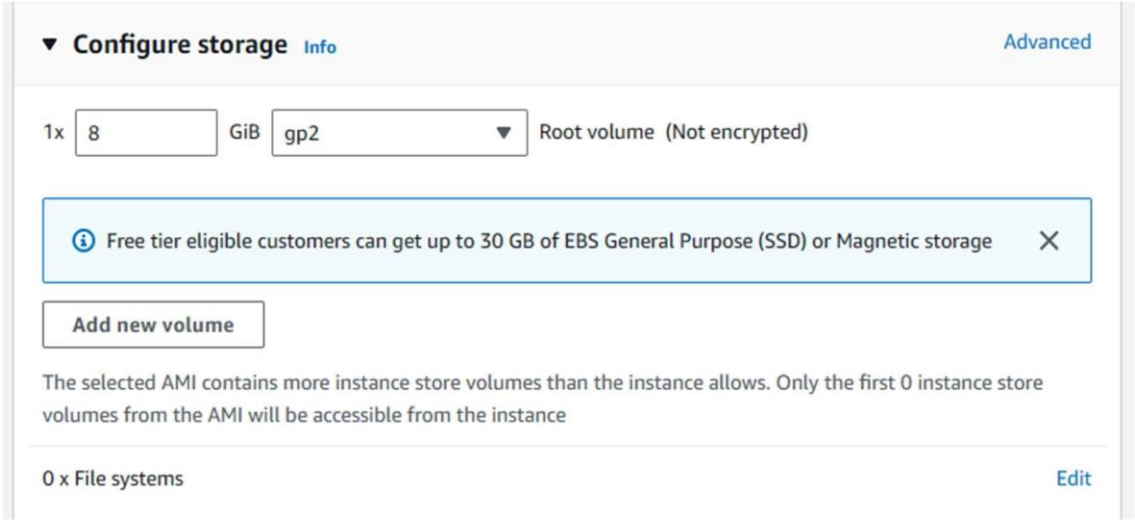
Step 4: Choose the default instance type (free tier eligible only)



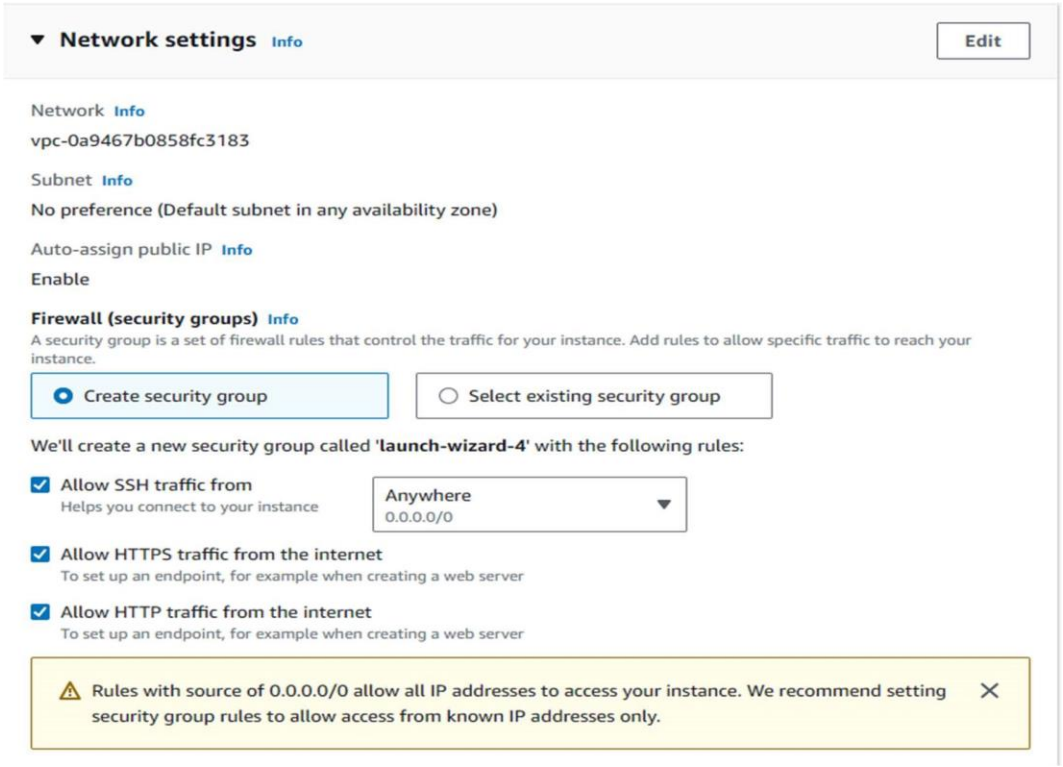
Step 5 : Now create key pair and select .pem file format and click on create key pair and download it.



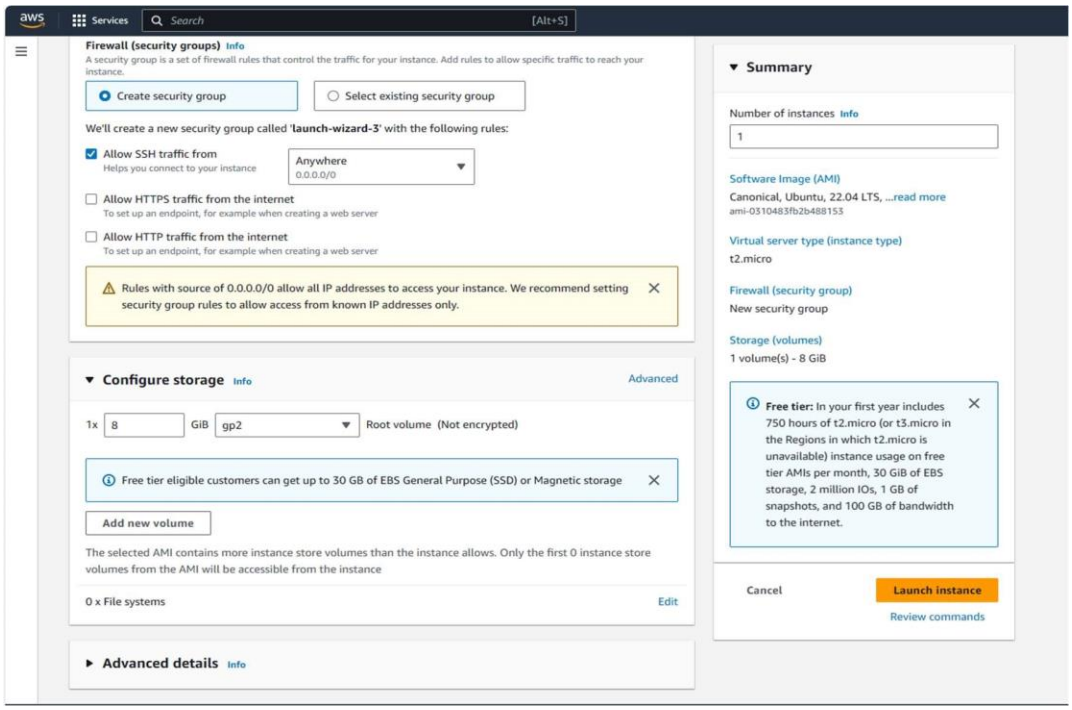
Step 6: In this 8 gb storage is sufficient.



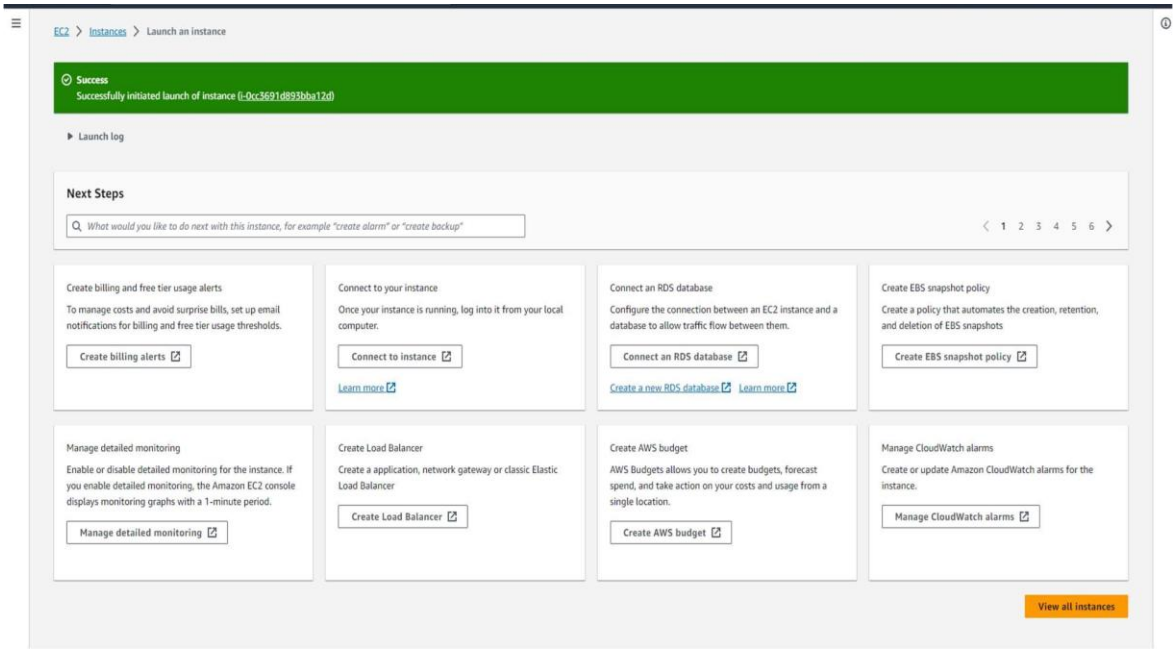
Step 7 : In this, select SSH, HTTPS, HTTP .



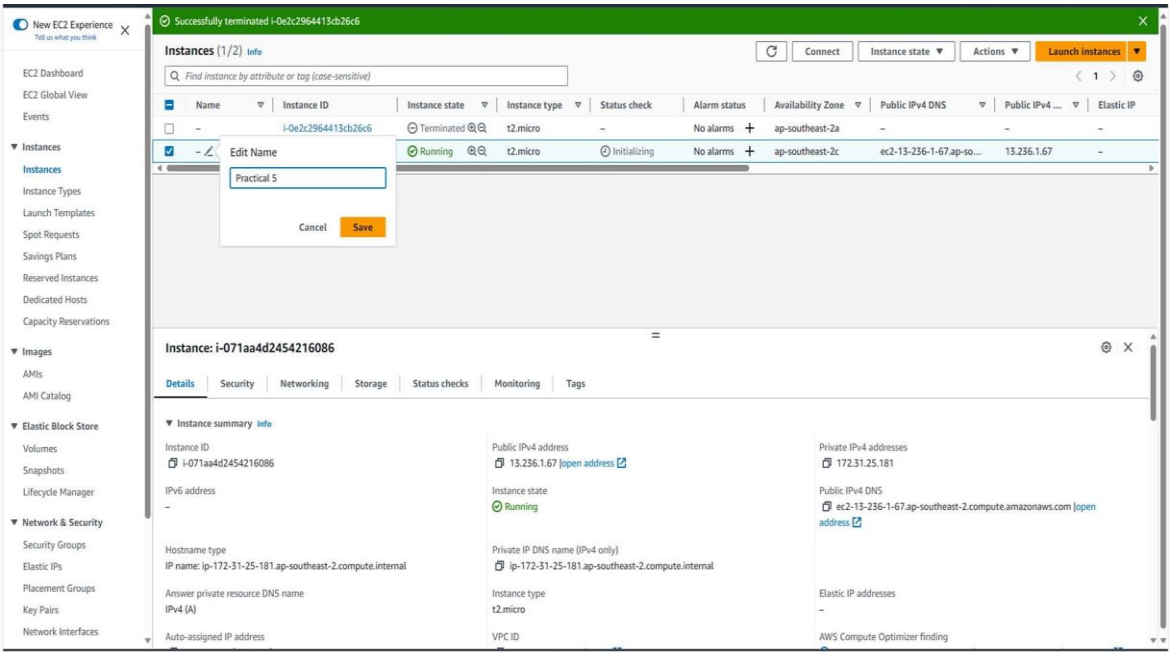
Step 8: Click on launch instance.



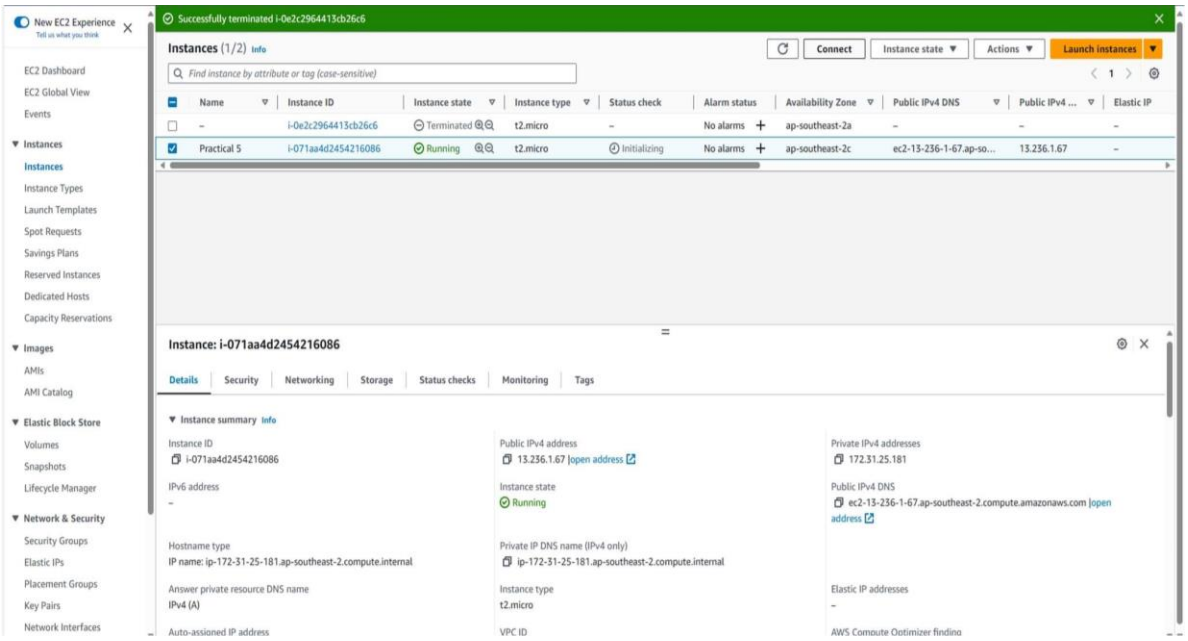
Step 9: Click on view all instances.

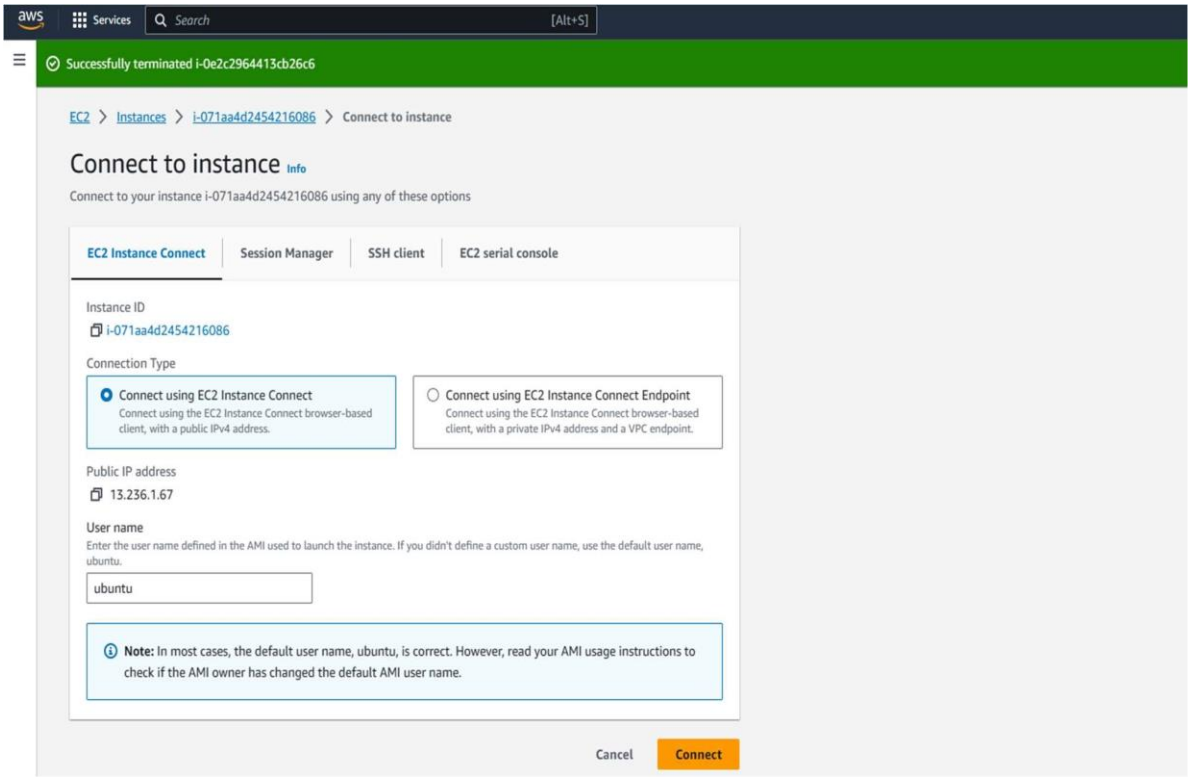


Step 10 : Then at this page your instance will be created and give any name to it to identify better be in future for example Practical 5



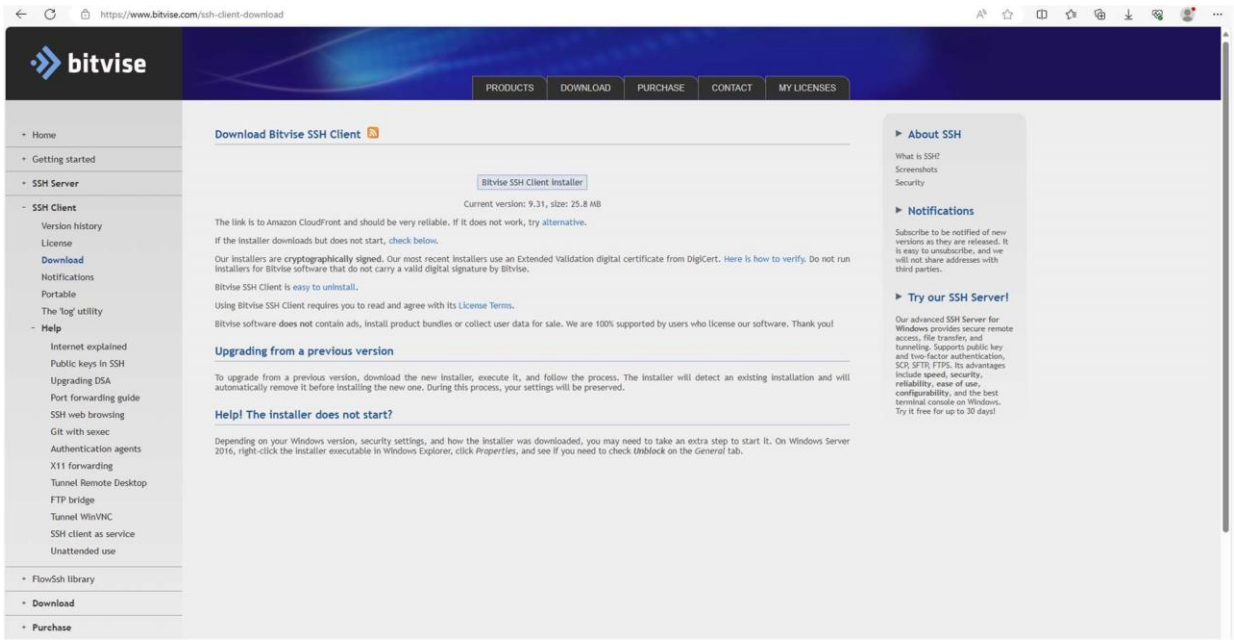
Step 11 : Then select that instance and click on connect at top to see your public IP and instance information.



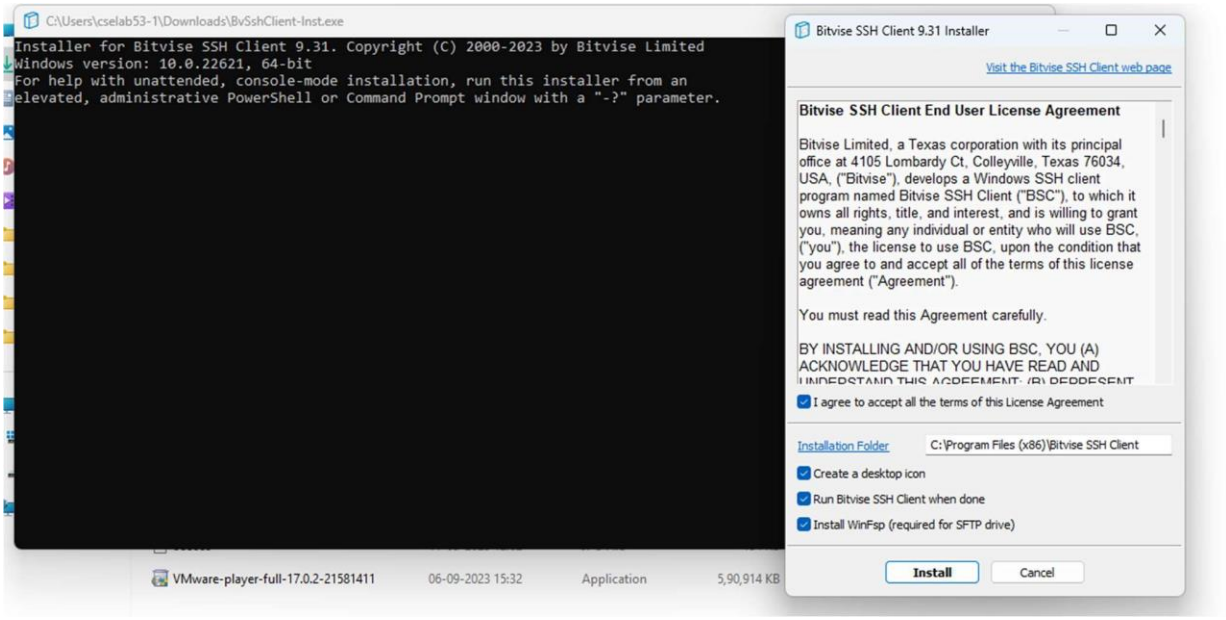


:

Step 12 : Then download Bitvise SSH client 9.31 from https://www.bitvise.com/ssh-clientdownload



Step 13: Click on install and install it.

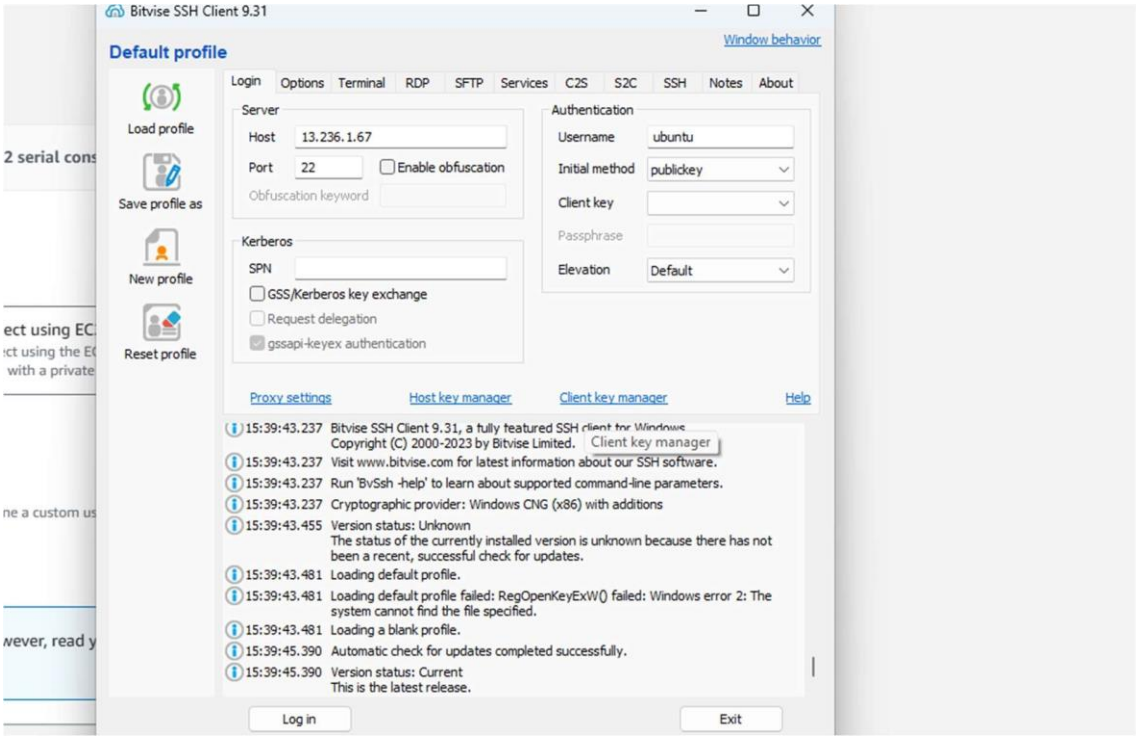


:

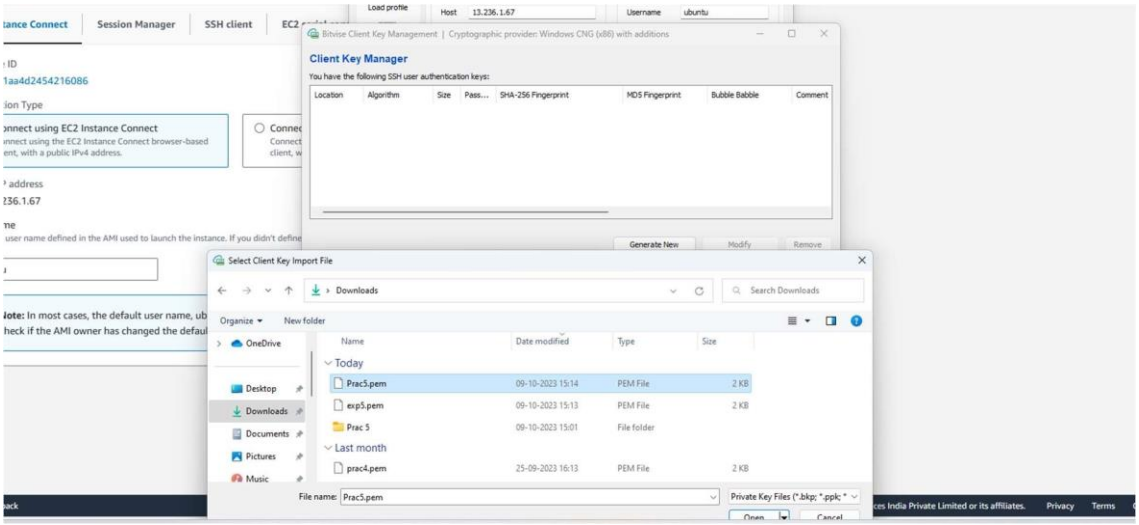
Step 14 Then open Bitvise SSH client and copy the public address of your instance from the EC2 instance connect section ex: 13.236.1.67. and paste it on the Bitvise Host text area column as shown in SS.

Then put the username as ubuntu, at initial method option select public key and elevation as default.

Then Click on Client key manager link.

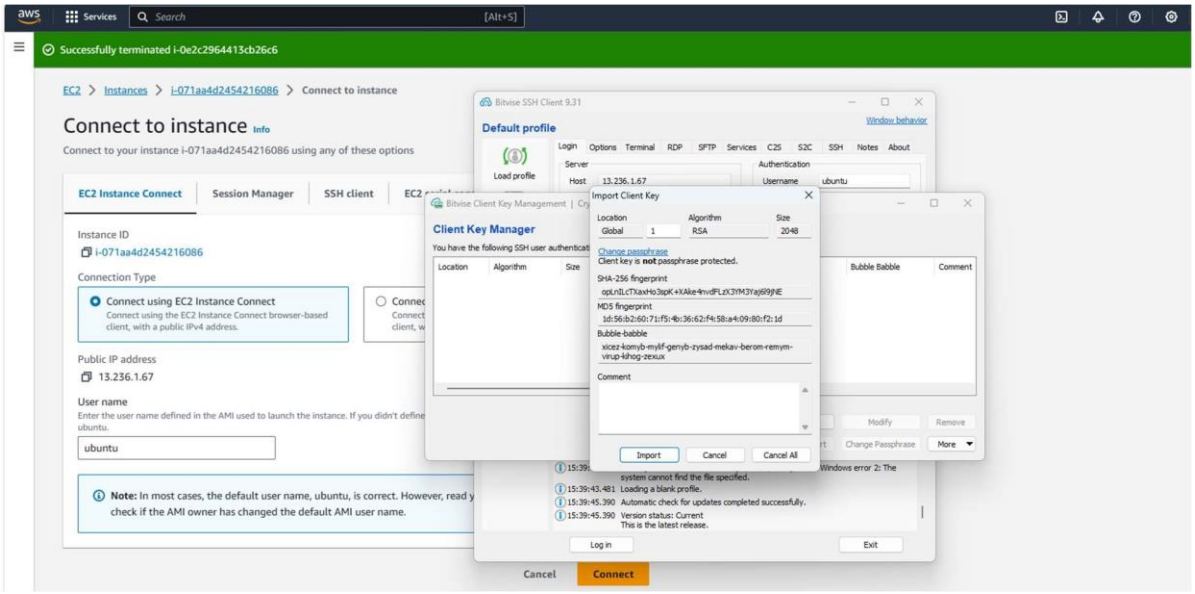


Step 15: Open the .pem file that we have downloaded at step no 5 (create key pair).

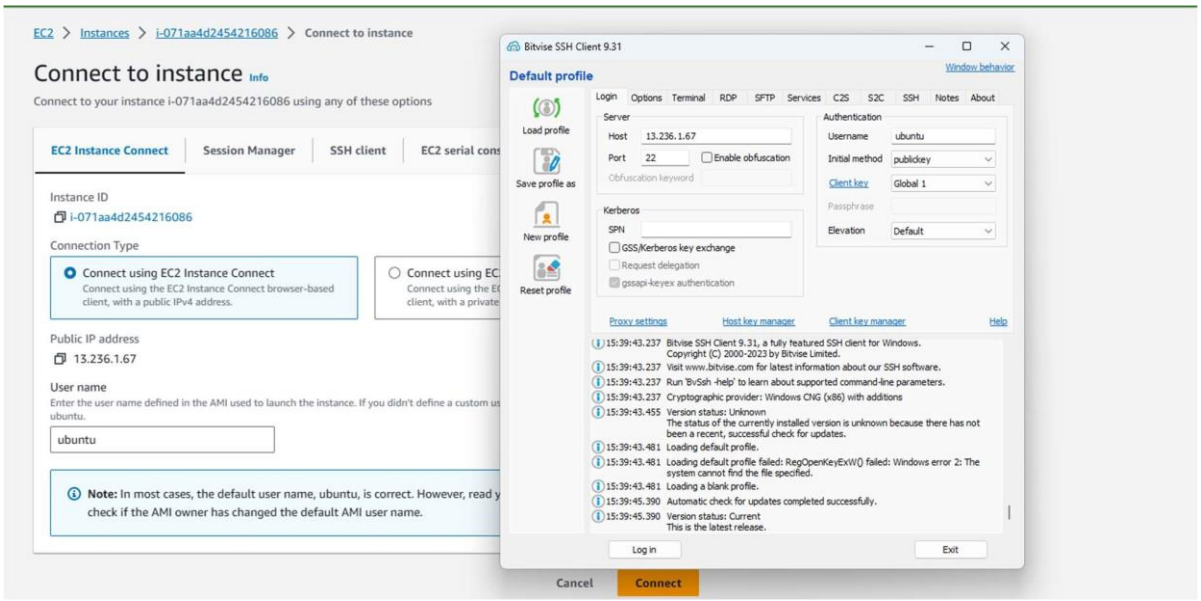


:

Step 16 : Then click on import button.

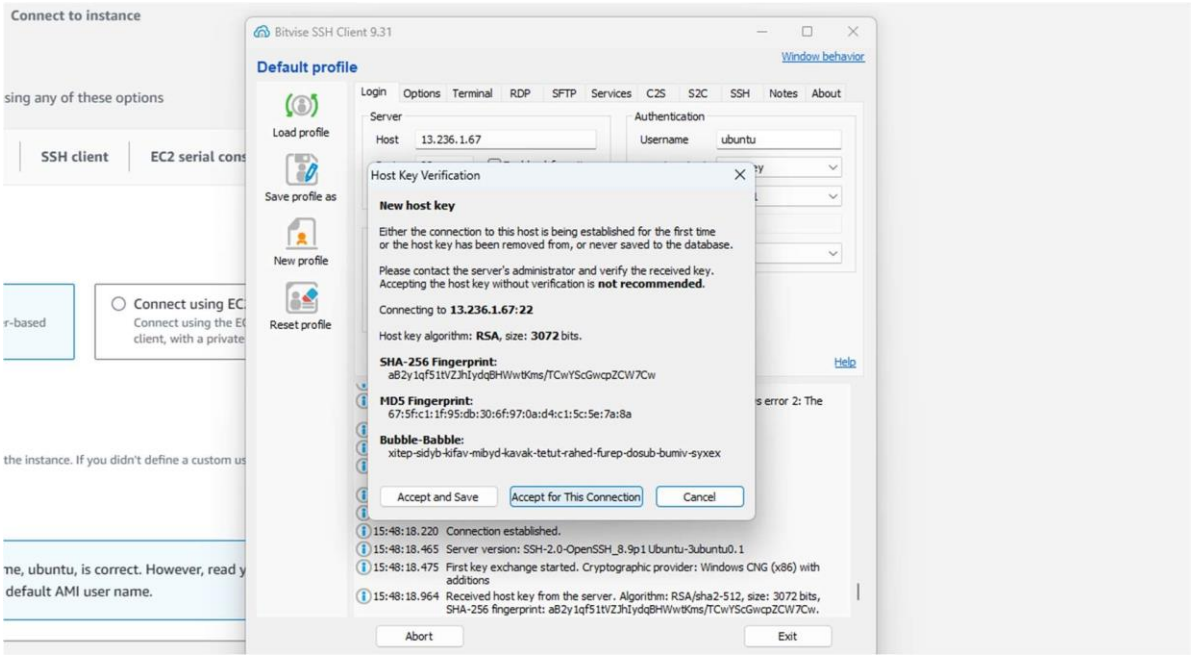


Step 17: Now at this step select the client key as Global 1 and click on log in button.

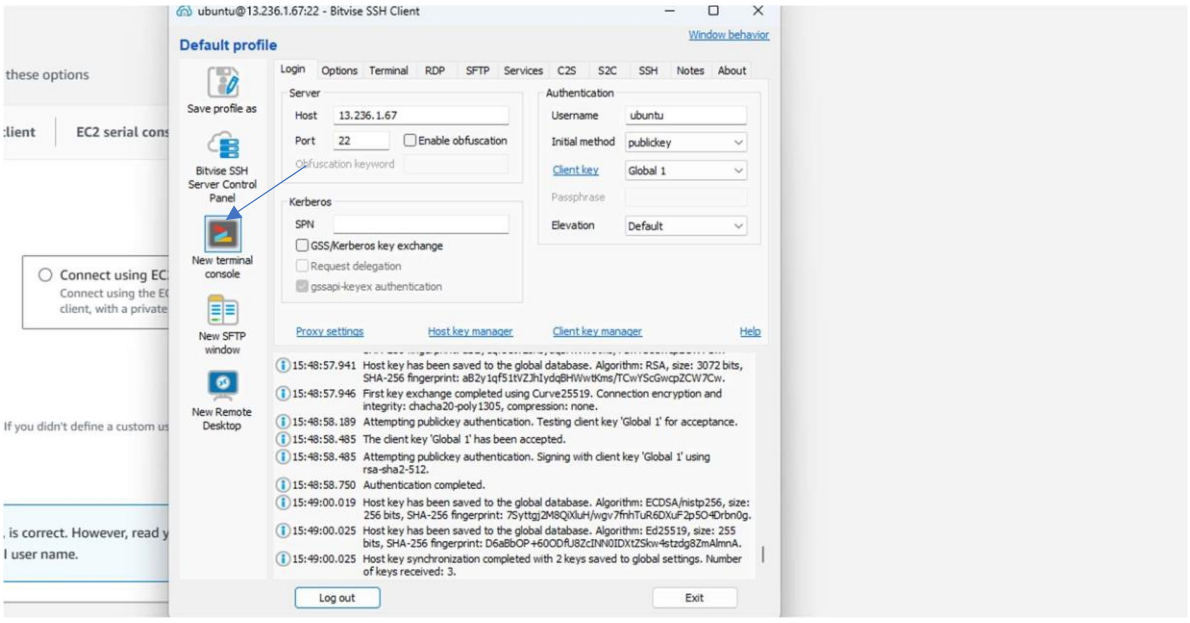


:

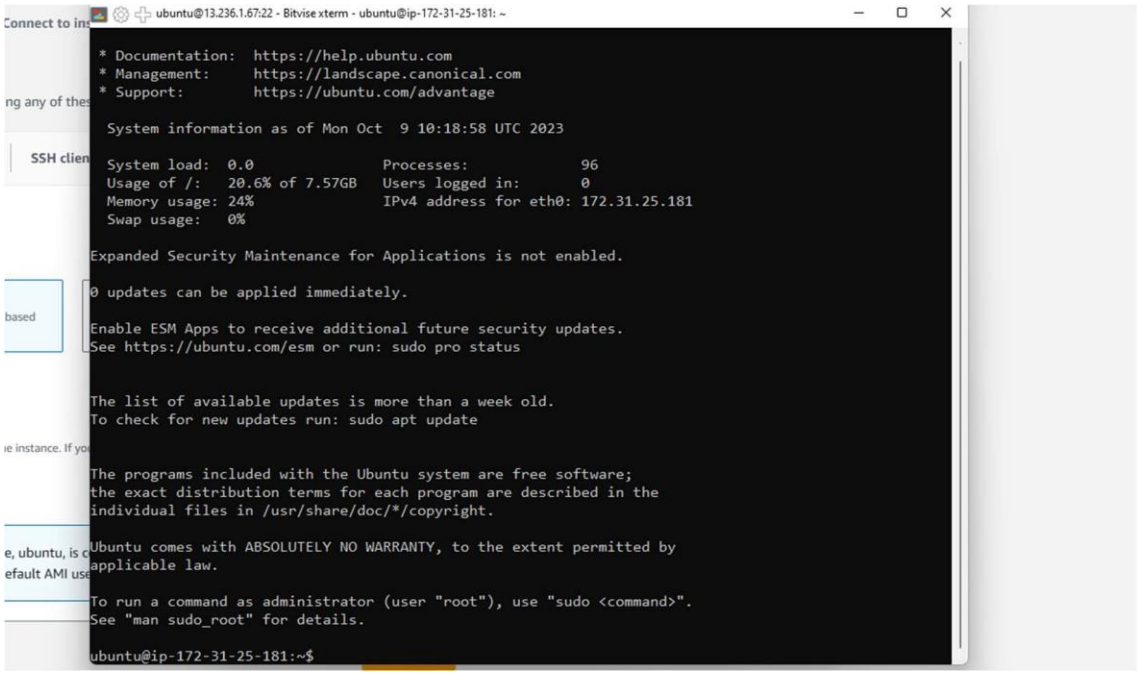
Step 18 Click on Accept and Save.



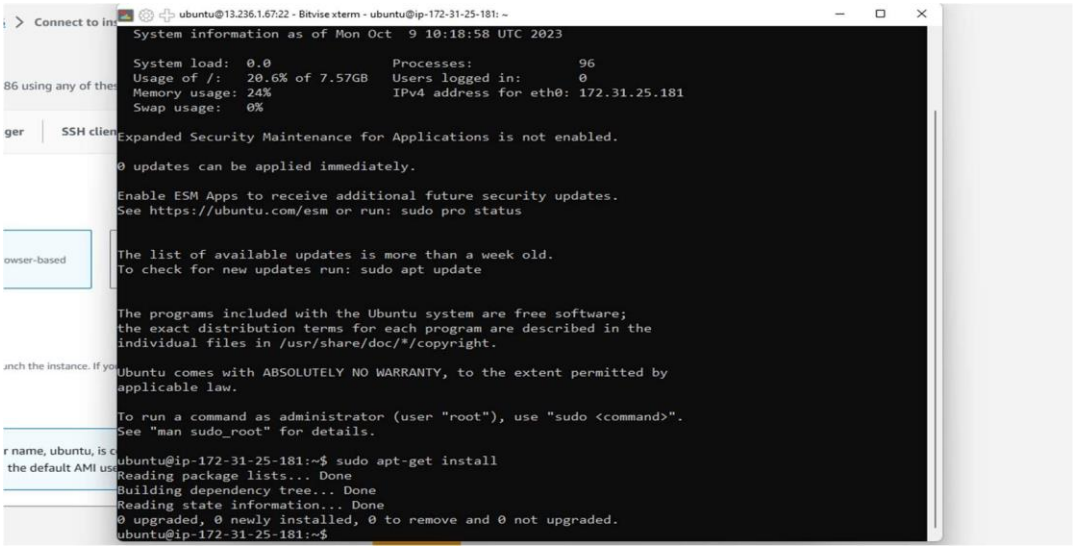
Step 19: Click on the new terminal window option to open the ubuntu OS.



Now our ubuntu OS window will Open like this.



Step 20: Use the following command to update your system before initiating a new installation sudo apt-get install or sudo apt install update depends on your system here the system is updated successfully.



```
E: Failed to fetch http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu/pool/main/a/apache2/apache2-u
tils_2.4.52-1ubuntu4.5_amd64.deb  404  Not Found [IP: 13.210.201.60 80]
E: Failed to fetch http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu/pool/main/a/apache2/apache2_2
.4.52-1ubuntu4.5_amd64.deb  404  Not Found [IP: 13.210.201.60 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
root@ip-172-31-25-181:~# apt-get update
Hit:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
```

Step 21: Switch to root user to install the apache server , for this , use following command.

sudo su -

```
ubuntu@ip-172-31-25-181:~$ sudo su -
root@ip-172-31-25-181:~#
```

Step 22: To install the apache2 use the following command.

Apt install apache2

```
ubuntu@ip-172-31-25-181:~$ sudo apt-get install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-25-181:~$ sudo su -
root@ip-172-31-25-181:~# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

Step 23: Then we need to install te nginx server in your root user for this, use following command it will be installed.

Sudo apt-get install nginx

```
root@ip-172-31-25-181:~# sudo apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

Step 24 To switch in /var/www/html folder use the following command cd

/var/www/html

:

Now to list files in folder use following command

ls

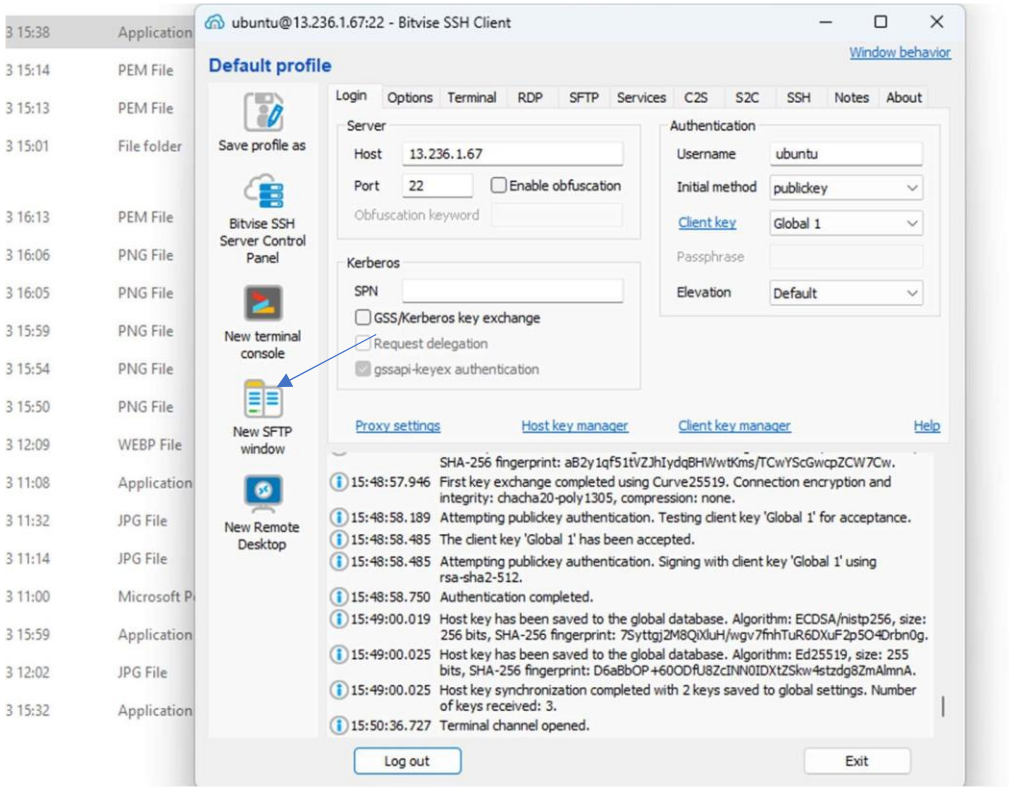
```
root@ip-172-31-25-181:~# cd /var/www/html
root@ip-172-31-25-181:/var/www/html# ^C
-bash: :s^C: substitution failed
root@ip-172-31-25-181:/var/www/html# ^C
root@ip-172-31-25-181:/var/www/html# ls
index.html  index.nginx-debian.html
root@ip-172-31-25-181:/var/www/html#
```

Step 25: To check where the apache2 server is installed or not, Simply paste the public instance IP on our web browser and search it.

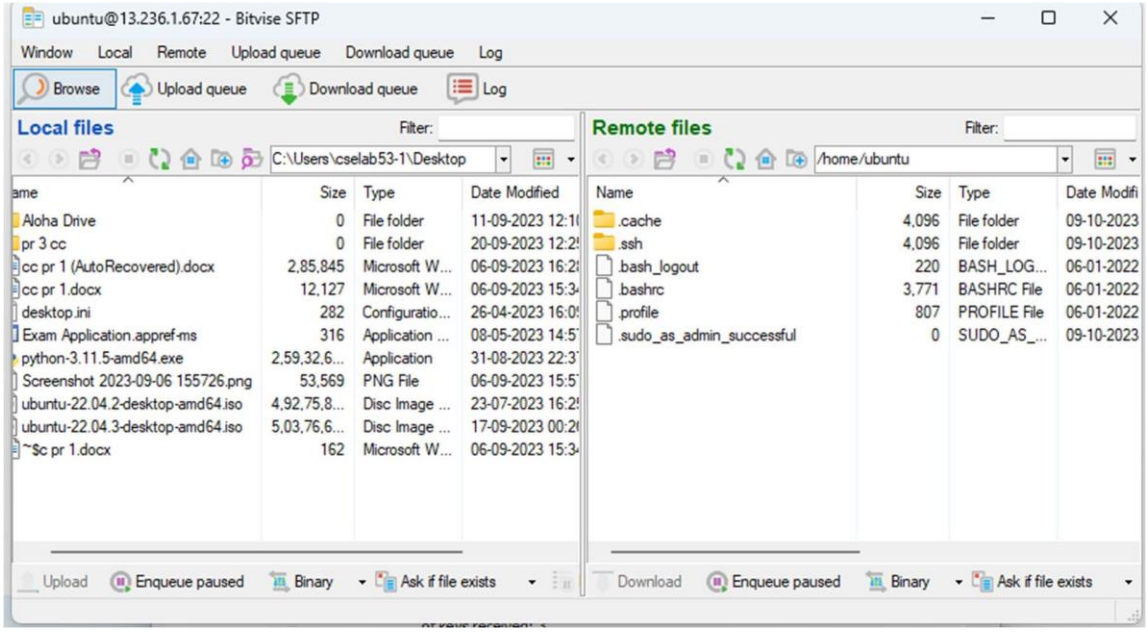
If the address got displayed by apache page then it is installed successfully.

:

Step 26 : Then click on the ‘New SFTP window’ button on Bitvise SSH client



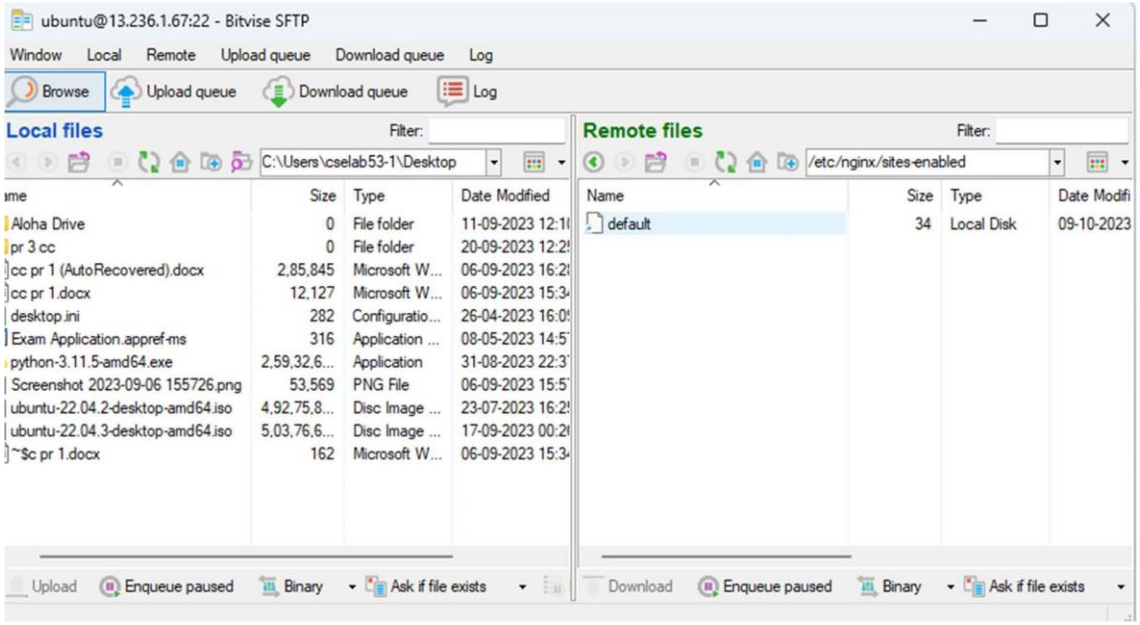
After clicking on new SFTP window the file transfer protocol will open.



:

Step 27 Then at search bar of SFTP open the path as:

/etc/nginx/sites-enabled



In the /etc/nginx/sites-enabled there is only one file is available that default file

So /etc/nginx/sites-enabled folder having the restriction to the user to perform read and write operations on it.

Step 28: Then to exceed restrictions open /etc/nginx directory at root user for this firstly use

use following command to go in root again sudo

su –

and then use cd

/etc/nginx

```
root@ip-172-31-25-181:/var/www/html# sudo su -
root@ip-172-31-25-181:~# cd /etc/nginx
root@ip-172-31-25-181:/etc/nginx#
```

:

Step 29 Use the following command for giving the permission to /etc/nginx.

sudo chmod 777

Now /etc/nginx root is allowing to perfrom read and write operations on it.

We can check that the read and write operations are allowed or not by creating file with .conf extension.

Step 30: To delete the index.html file from the /var/www/html folder use following command.

rm index.html

The index.html file is deleted now

To see whether the index.html file is deleted or not, use list command again

ls

```
root@ip-172-31-25-181:~# cd /var/www/html
root@ip-172-31-25-181:/var/www/html# ls
index.html  index.nginx-debian.html
root@ip-172-31-25-181:/var/www/html# rm index.html
root@ip-172-31-25-181:/var/www/html# ls
index.nginx-debian.html
root@ip-172-31-25-181:/var/www/html#
```

We can see that indx.html is now deleted.

Step 31: Now create a new as “index.html” file again. For this use following command

sudo nano index.html

```
root@ip-172-31-25-181:/var/www/html# sudo nano index.html
```

Step 32: Now the blank index.html is getting created. So we can add our html content over here.

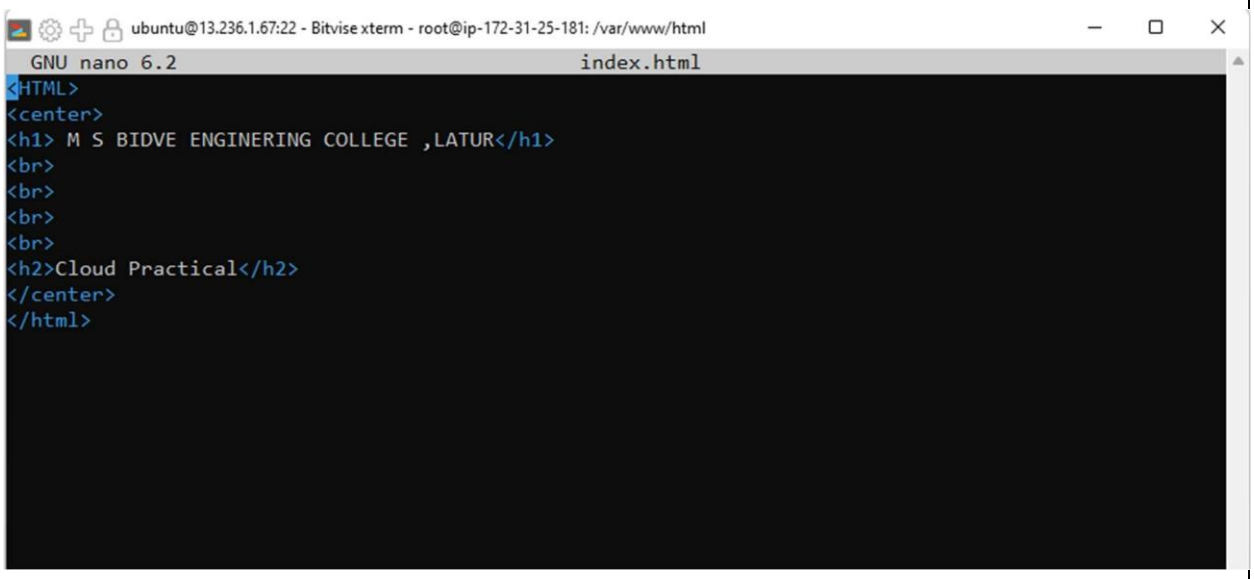
Example:

<HTML>

<h1> M S BIDVE ENGINEERING COLLEGE, LATUR </h1>

<h2>Cloud Practical </h2>

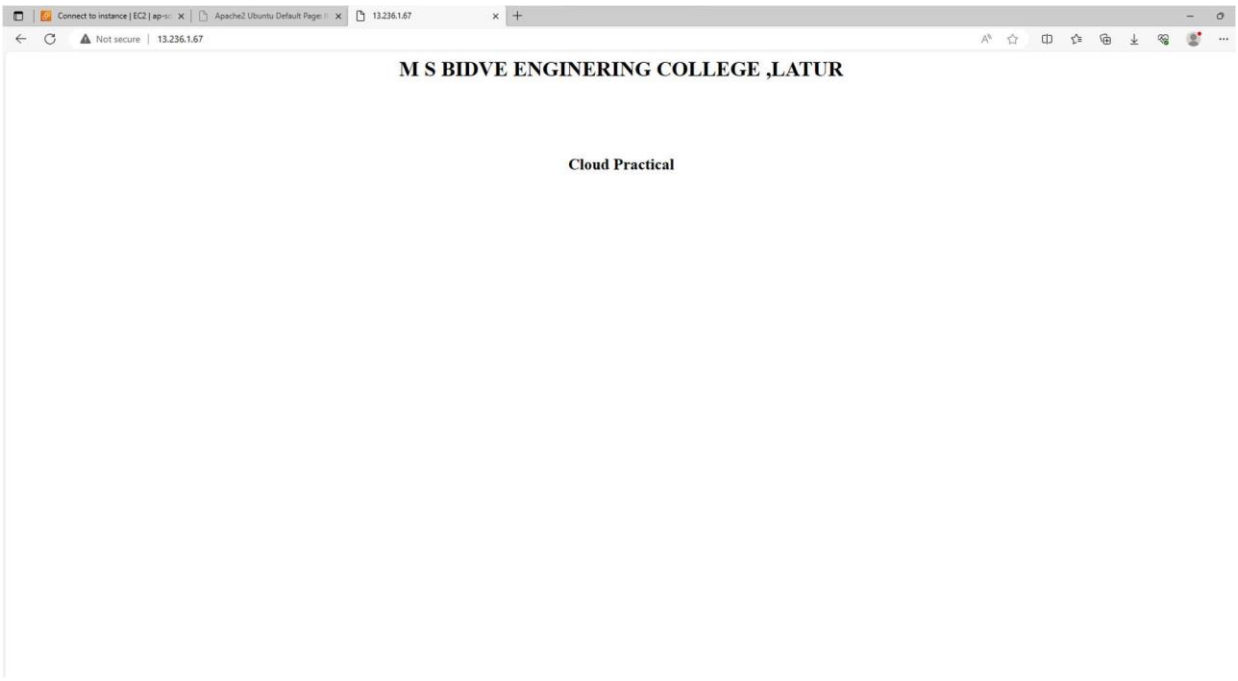
</html>



Save this code in index.html for this use ctrl + X command (^X) on the index.html

So now the index.html file is sorted.

Step 33: So to access our web site paste your instance IP address in web browser



Conclusion: We are able to perform the Provisioning and Scaling of the websites using AWS