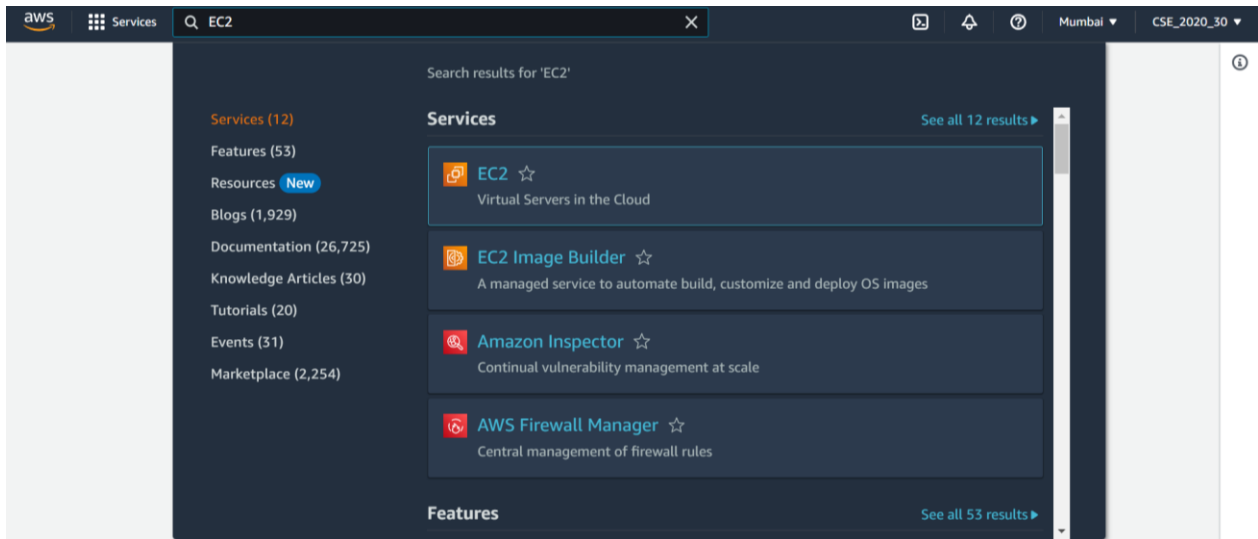


ASSIGNMENT-7

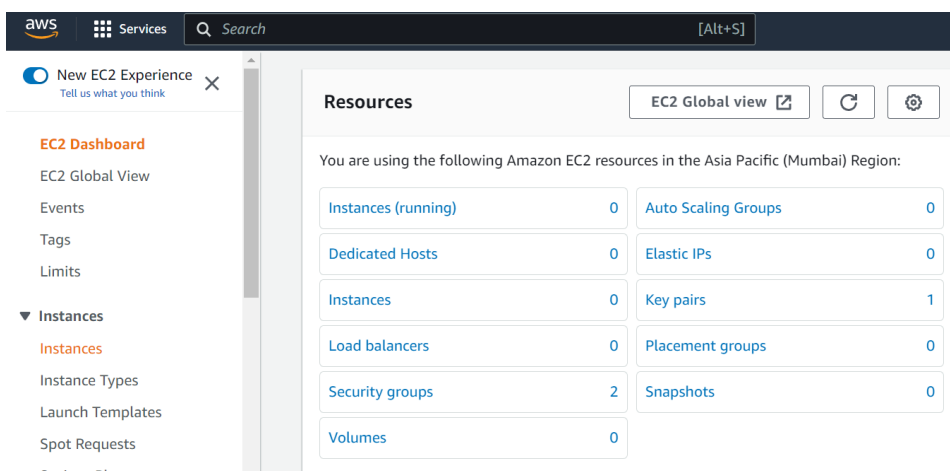
Problem Statement: Upload a static website on EC2.

Procedure:

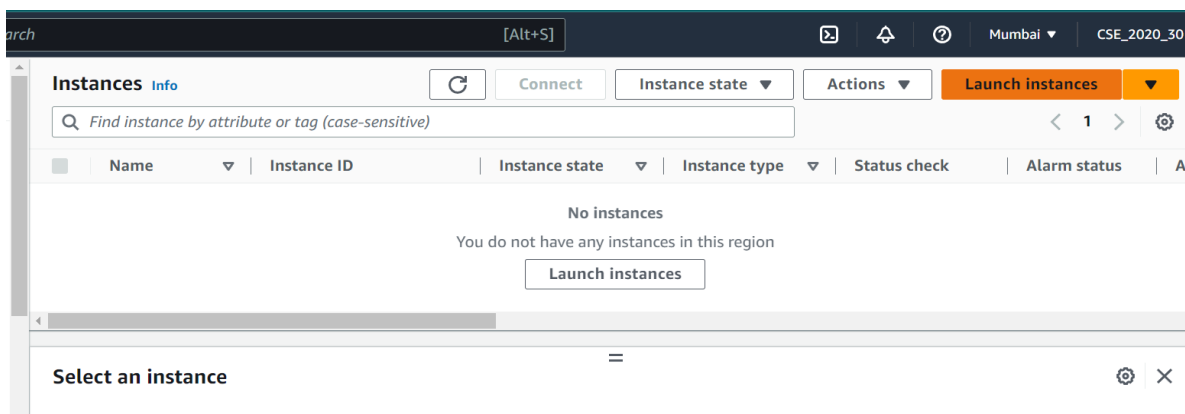
1. Login to your AWS account as root user. Then search “EC2” in the search box. Click on the first result that appears.



2. Click on Instances dropdown menu on the left sidebar. Then again click on instances.



3. Next click on Launch instances button.



4. Now customize the instance you want to launch.

- a. Set the unique instance name.
- b. Select Ubuntu as OS.
- c. Next go to key pair(login) section.
 - i. Click on create new key pair
 - ii. Enter the name of key pair.
 - iii. Select RSA as Key pair type.
 - iv. Select “. pem” as file format.
 - v. Create the key pair.
 - vi. Save the automatically downloaded file. It will be required later.
- d. Now select the newly created key pair from the dropdown selection.
- e. Go at the bottom of the network settings section and check the
 - i. Allow HTTP traffic box.
 - ii. Allow HTTPS traffic box.
- f. Next Click on Launch Instance button on the right side.

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

debserver1

Add additional tags

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

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

Browse more AMIs

Amazon Machine Image (AMI)

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

debkey1

Create new key pair

Enter the name of the key pair below. 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](#)

Key pair name

Enter key pair name

The name can include upto 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 (Not supported for Windows instances)

Private key file format

☒ .pem

For use with OpenSSH

☐ .ppk

For use with PuTTY

Cancel

Create key pair

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-2' with the following rules:

- ☒ Allow SSH traffic from Anywhere (0.0.0.0/0)
Helps you connect to your instance
- ☒ 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

Summary

Number of instances [Info](#)
1

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 available)

[Cancel](#) [Launch instance](#)

- Now check whether your newly created instance is running or not in the instances page. Note it will take a few seconds to show the running status. (From Pending)

Instances (1) [Info](#) [Refresh](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Actions
<input type="checkbox"/>	debserver1	i-097ad2f7005b61255	Running	t2.micro	-	No alarms	Connect Instance state Actions

Select an instance

- Now click on the Instance ID of the server.

EC2 > **Instances** > **i-097ad2f7005b61255**

Instance summary for i-097ad2f7005b61255 (debserver1) [Info](#)
Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state](#) [Actions](#)

Instance ID i-097ad2f7005b61255 (debserver1)	Public IPv4 address 13.233.93.219 open address
IPv6 address -	Instance state Running

- Copy the Public IPv4 address.
- Now for the next steps we require **Bitvise SSH client**. Download it and install in your local pc.
- Now open the Bitvise SSH Client.

Bitvise SSH Client 9.27 [Window behavior](#)

Default profile

[Load profile](#) [Save profile as](#) [New profile](#) [Reset profile](#)

Login **Options** **Terminal** **RDP** **SFTP** **Services** **C2S** **S2C** **SSH** **Notes** **About**

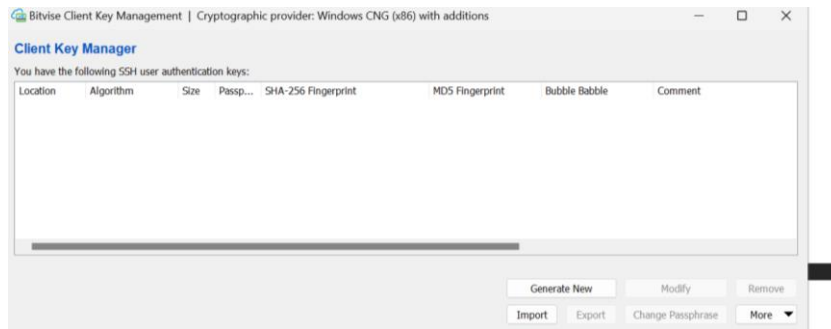
Server
Host:
Port: ☐ Enable obfuscation
Obfuscation keyword:

Authentication
Username:
Initial method: none
Elevation: Default

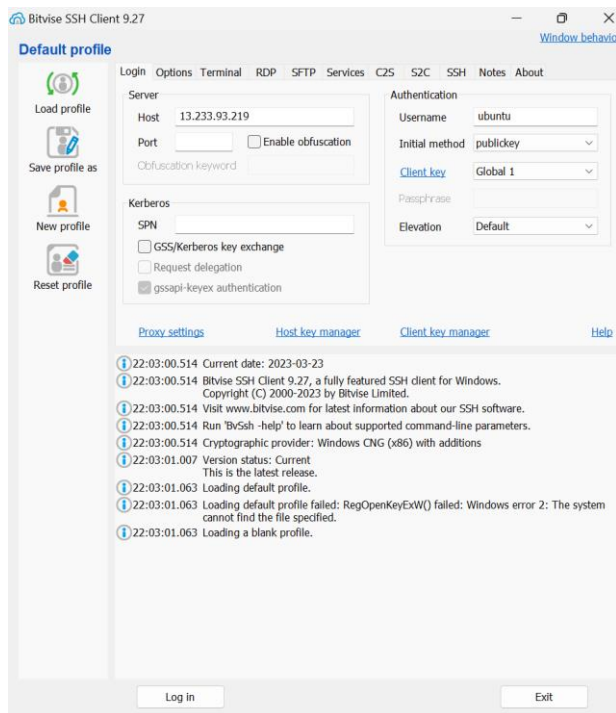
Kerberos
SPN:
☐ GSS/Kerberos key exchange
☐ Request delegation
☒ gssapi-keyex authentication

[Proxy settings](#) [Host key manager](#) [Client key manager](#) [Help](#)

10. Paste the copied IPv4 address in the Host section.
11. Set user name to ubuntu.
12. Click on the client key manager link below the authentication section.
It will open another pop-up window. There click on import button.
Select the previously downloaded .pem file. Click on import. Then close the Client key manager window.

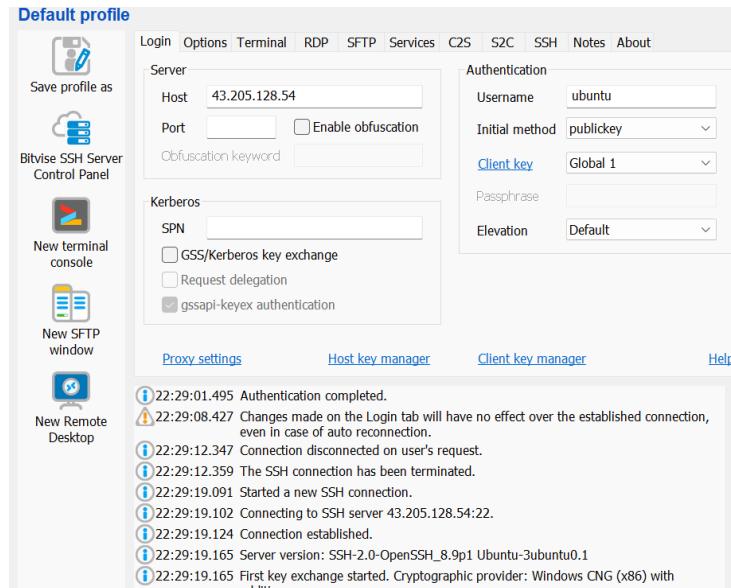


13. Now set initial method to public key.
14. Set Client Key to Global 1.



15. Now click on the Log In button at the bottom of the Window.
Click on Accept and Save button on the pop-up.
One of many ways in which you can know that whether you have successfully logged in is if your Log In button has changed to Log Out.

16. Now newly created options will arise on the left sidebar on successful login. Click on the new terminal console to open terminal of our server.



17. Enter the following commands:

a. **sudo apt-get update**

b. **sudo apt-get upgrade**

(Remember to press Y and then Enter when prompted)

(After the process is completed a new box/window appears. But just press Enter to continue.)

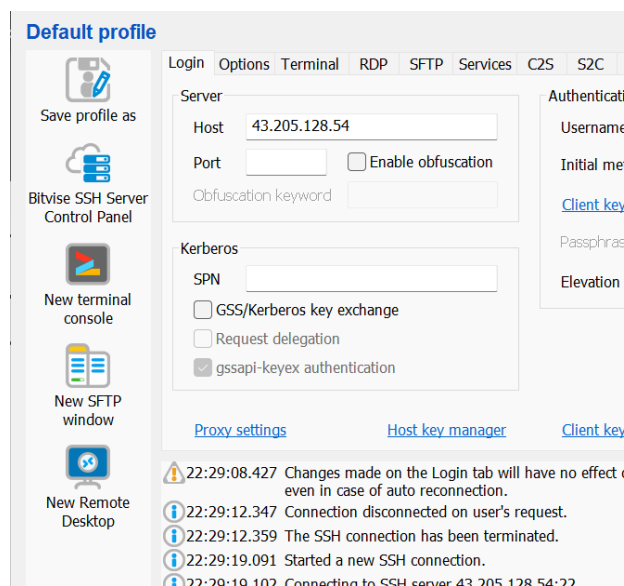
c. **sudo apt-get install nginx**

(Remember to press Y and then Enter when prompted)

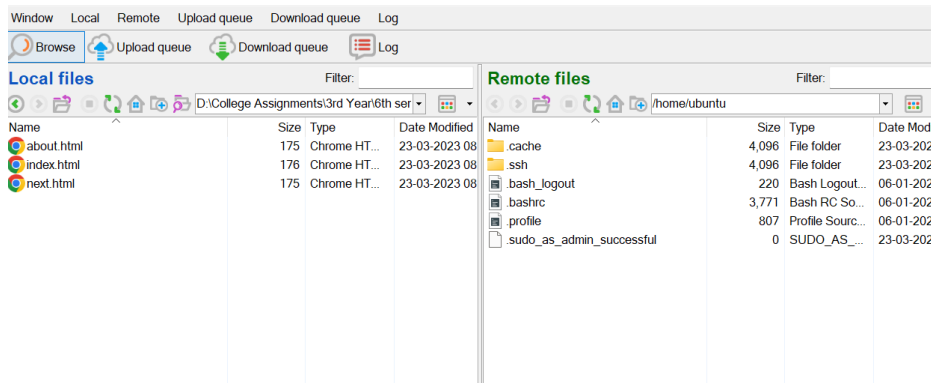
(After the process is completed a new box/window appears. But just press Enter to continue.)

18. Now minimize the console.

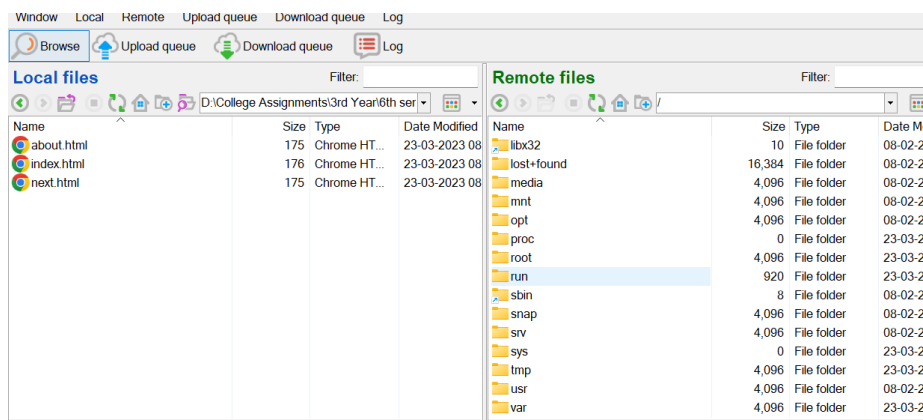
19. Click on the new SFTP window icon on the left sidebar.



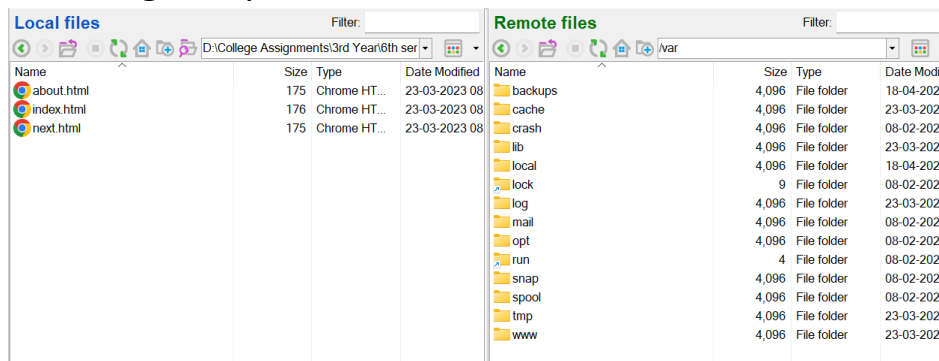
20. Select the folder where you have kept HTML files of your website on the local files section. Just keep it open.



21. Now click the Up button (2 times) on the Remote Files section. You will be able to see a bunch of folders. Scroll down and open the last folder named “var”.

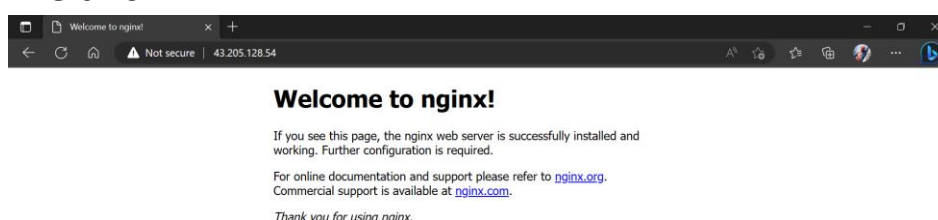


22. Now again open the last folder in it named “www”.



23. Open the only folder named “html” and keep it open. You will see a default html already present.

You can check whether nginx is working by pasting our previously copied IPv4 address of our server instance in a different browser. It will show something like this.



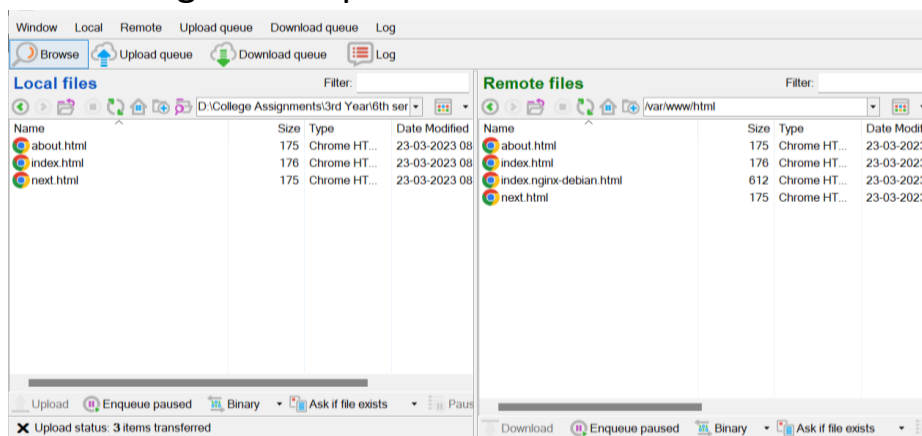
24. We actually need to transfer our local html file here in this open folder of the remote server. However, we do not have such permissions for this folder. To give such permission we need to go back to the terminal console and give the required permissions to the folder.

25. Now type the following commands in the terminal.

- a. **cd /**
- b. **cd var/www/**
- c. **sudo chmod 777 html**

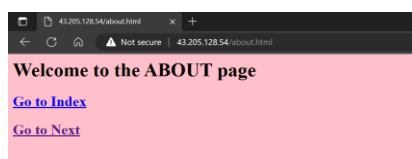
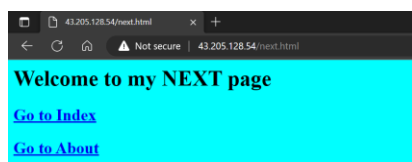
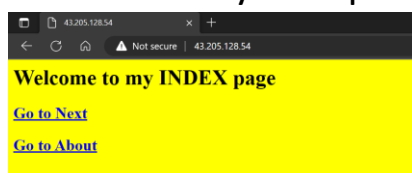
Now the permission (Read, Write, Execute) of the folder is successfully granted.

26. Now drag and drop all the files from local to remote.



Remember you must have the opening html named “index.html” in order to show the opening html page by the web server.

27. Finally open the website from any browser or device by using the public IPv4 address that you copied.



We now have successfully hosted a static website on an AWS EC2 sever.