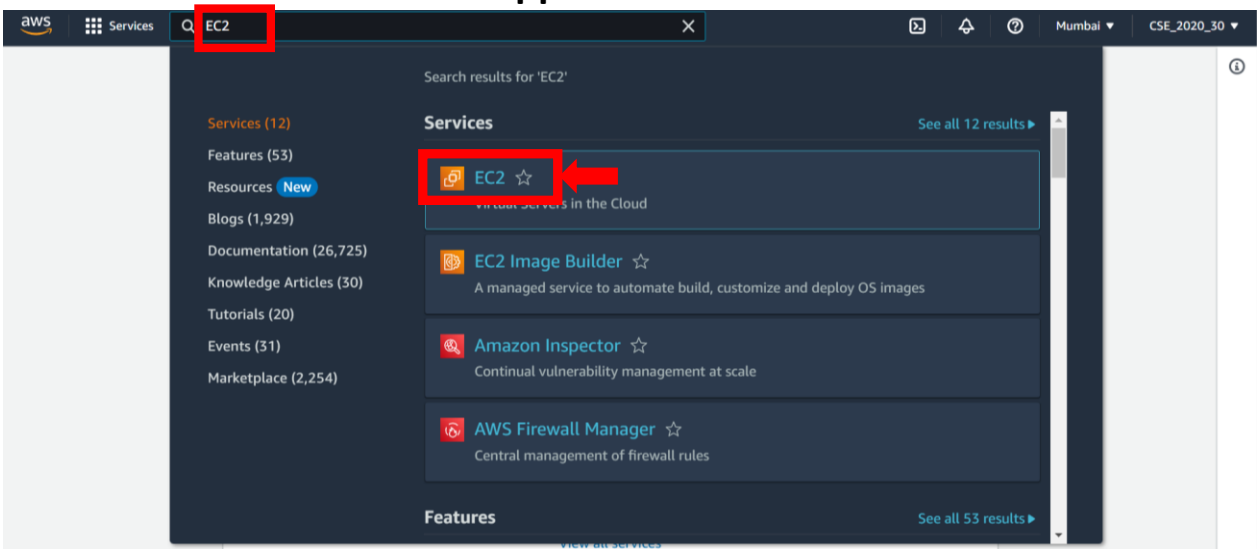
ASSIGNMENT-7

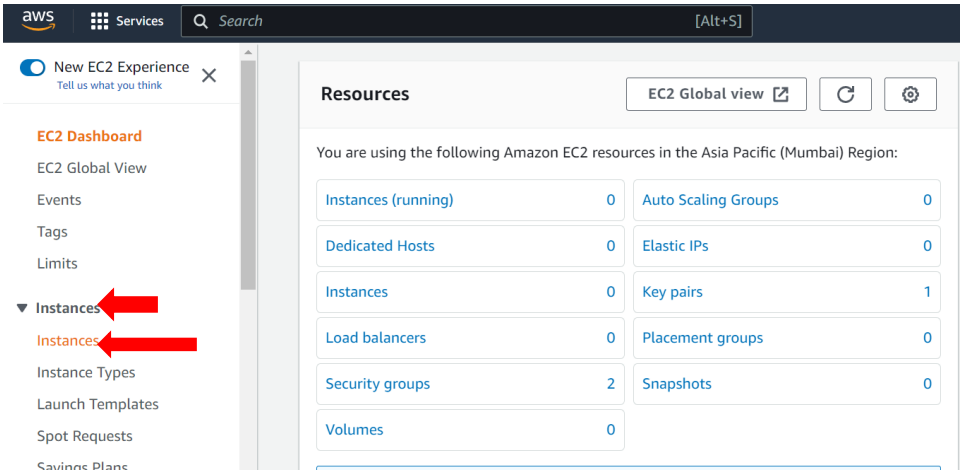
Problem Statement: Upload a static website on EC2.

Procedure:

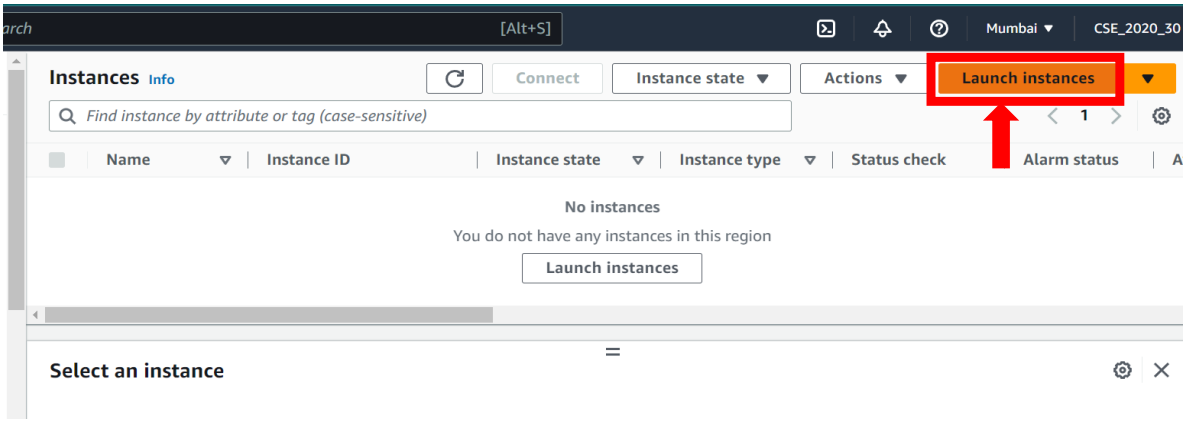
1. Login to your AWS account as root user. Then search **"EC2"** in the search box. **Click on the first result that appears**.



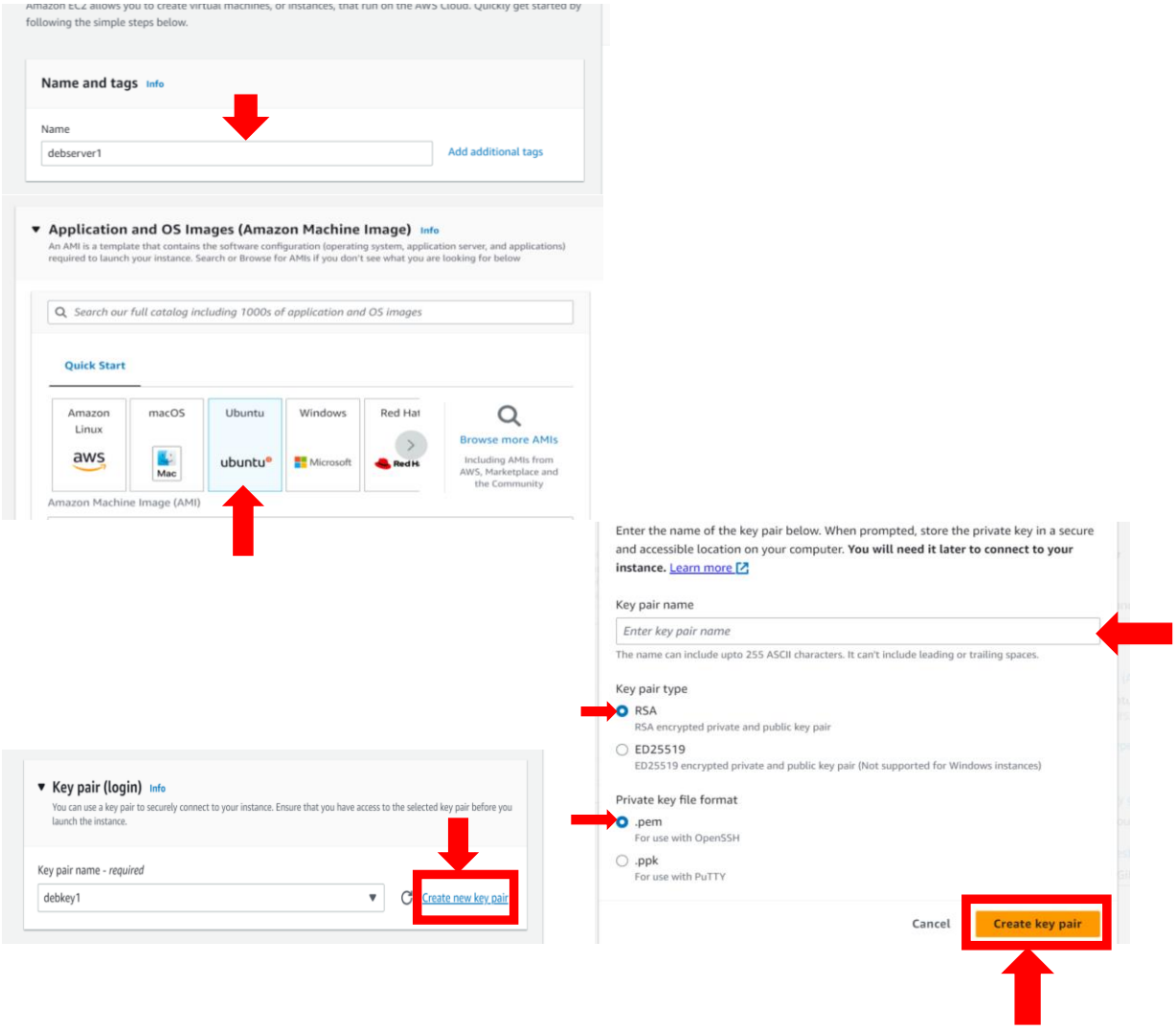
2. Click on **Instances dropdown menu** on the left sidebar. Then again click on **instances**.

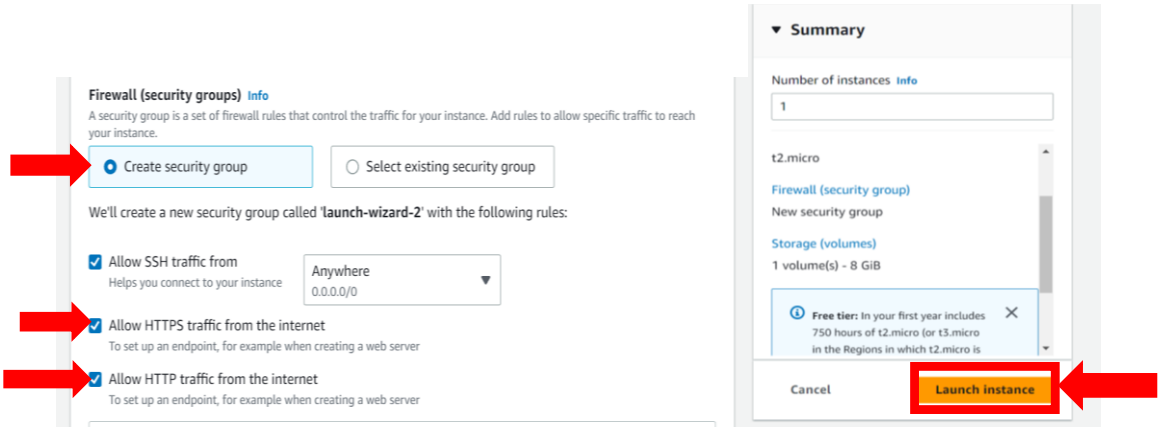


3. Next click on **Launch instances** button.

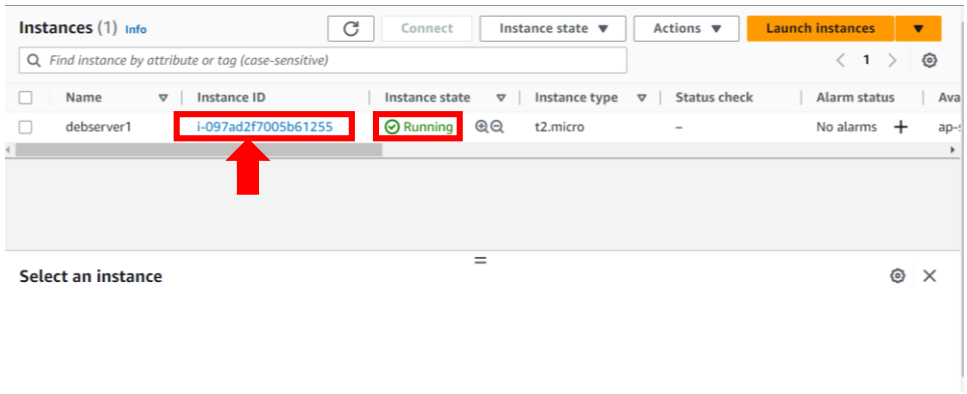


4. Now customize the instance you want to launch.
    a. Set the **unique instance name**.
    b. Select **Ubuntu** as OS.
    c. Next go to key pair(login) section.
        i. Click on **create new key pair**
        ii. Enter the **name** of key pair.
        iii. Select **RSA** as Key pair type.
        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.



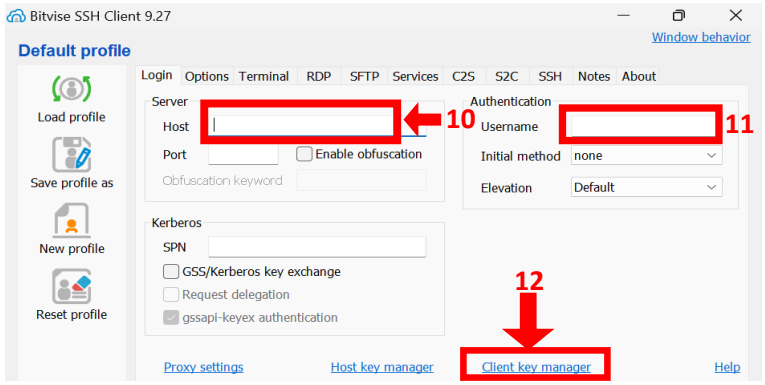


5. Now **check** whether your newly created instance is **running** or not in the instances page. **Note it will take a few seconds to show the running status. (From Pending)**

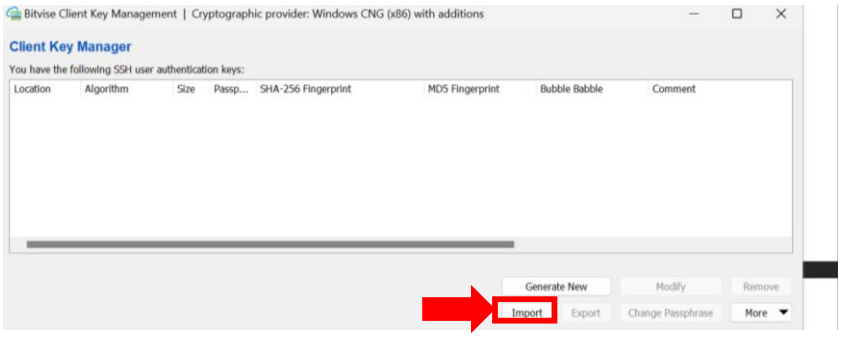


6. Now **click** on the **Instance ID** of the server.

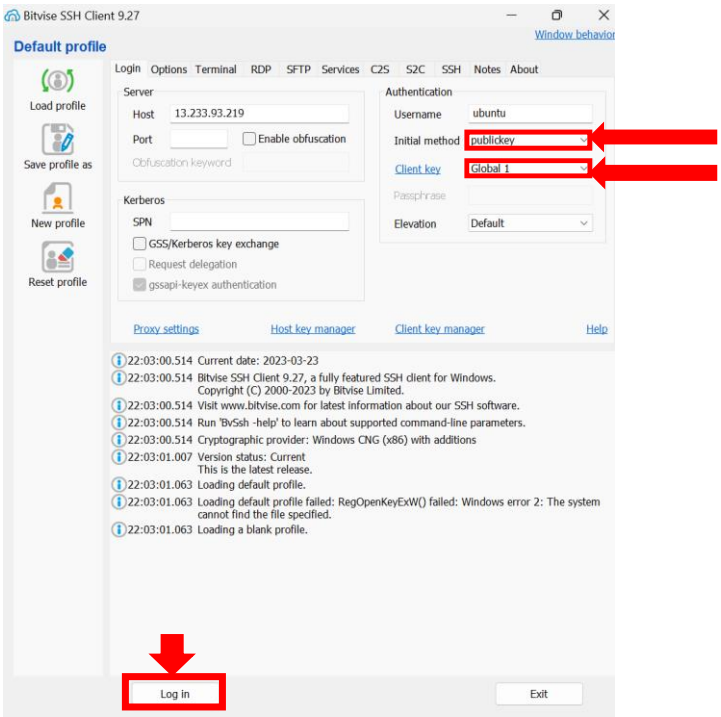
7. **Copy the Public IPv4 address**.
8. Now for the next steps we require **Bitvise SSH client**. Download it and install in your **Personal Computer**(local).
9. Now **open** the Bitvise SSH Client.



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**.

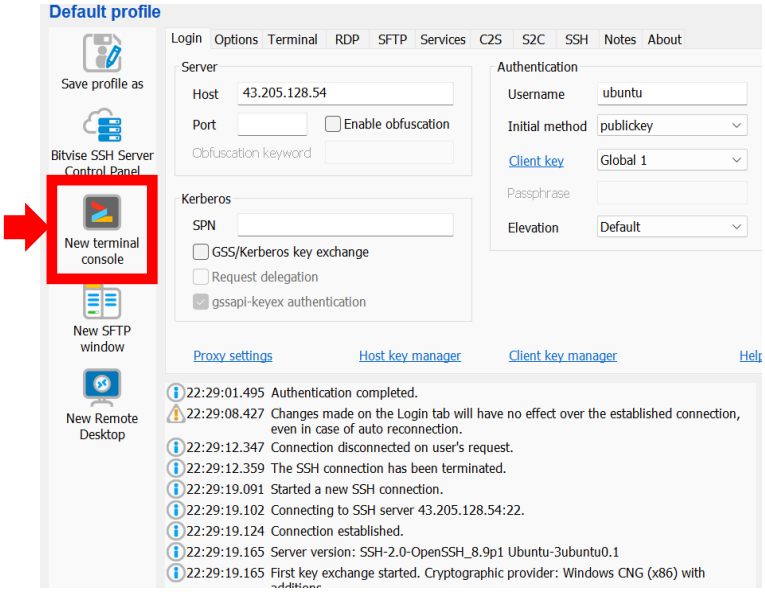


13. Now set **initial method** to **public key**.
14. Set **Client Key** to **Global 1**.



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.)

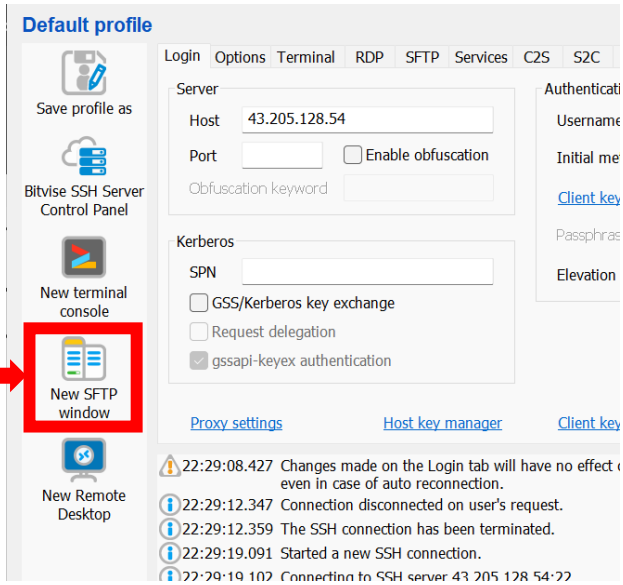
c. **sudo apt-get install nginx**

(Remember to press Y and then Enter when prompted)

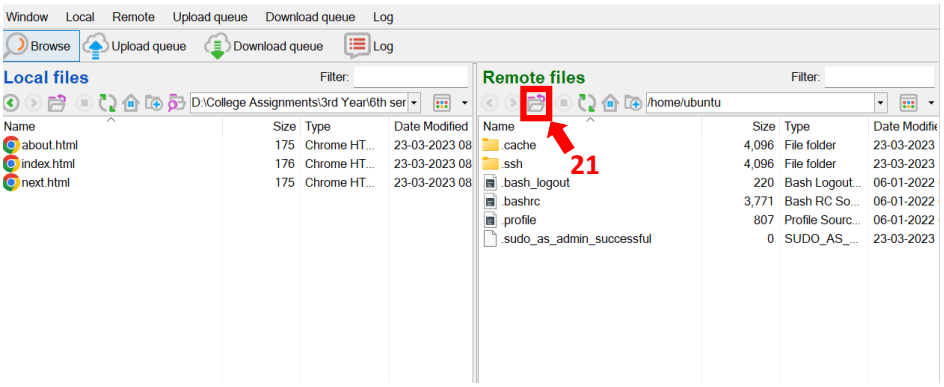
(After the process is completed a new box/window appears. But just press Enter to continue.)

18. Now **minimize** the console.

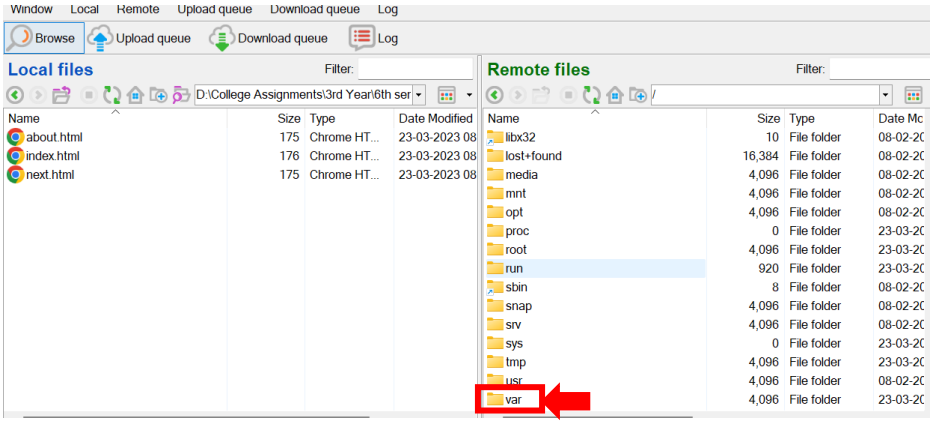
19. Click on the new **SFTP window** icon on the left sidebar.



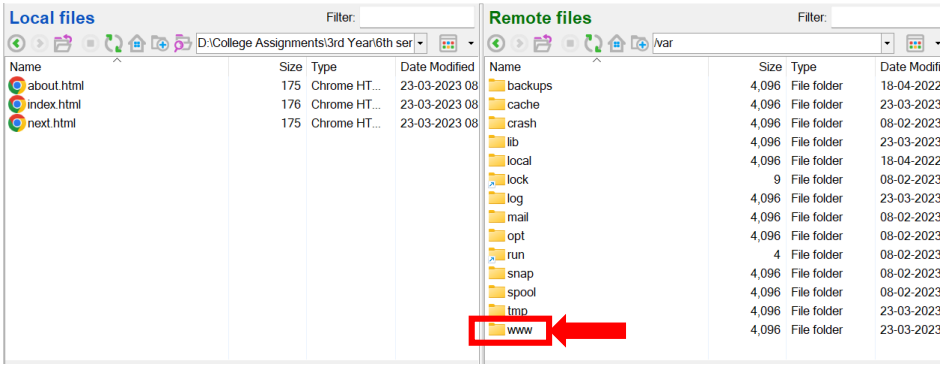
20. **Select the folder where you have kept HTML files of your website on the local files section**. Just **keep it open**.



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**".

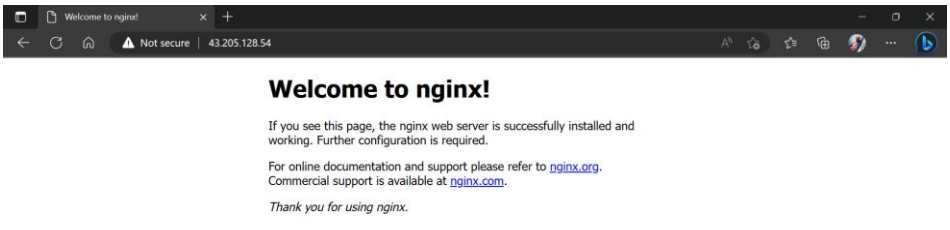


22. Now again **open** the **last folder** in it named "**www**".



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.



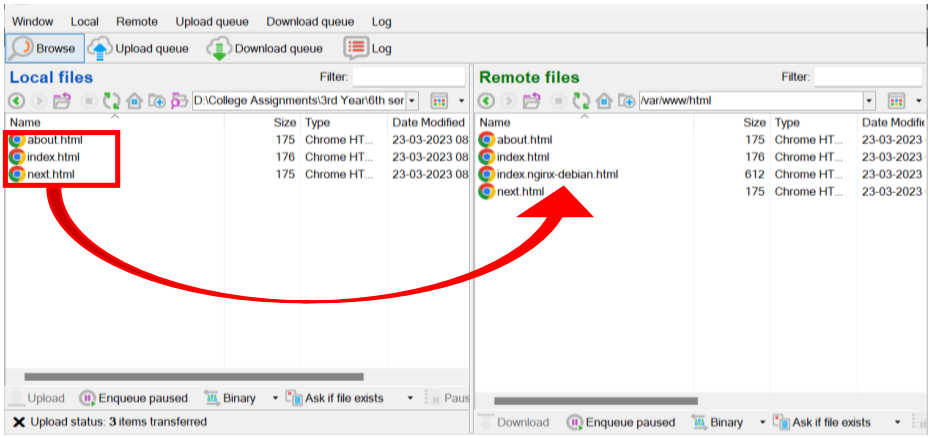
24. We actually need to transfer our local html file here in this open folder of the remote server. However, we do not have such permissions for this folder. To give such permission we need to go back to the terminal console and give the required permissions to the folder.

25. Now type the **following commands** in the **terminal** (which we minimized).

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

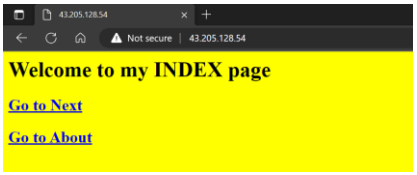
Now the permission (Read, Write, Execute) of the folder is successfully granted.

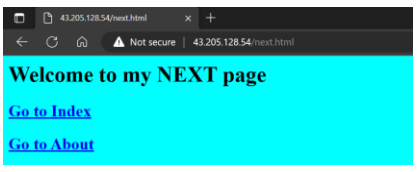
26. Now drag and drop all the files from <u>Local Files</u> to <u>Remote Files</u>.

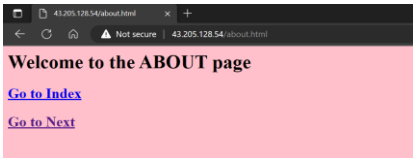


<u>Remember</u> you must have the <u>opening html</u> named "<u>index.html</u>" in order to show the opening html page by the web server.

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.