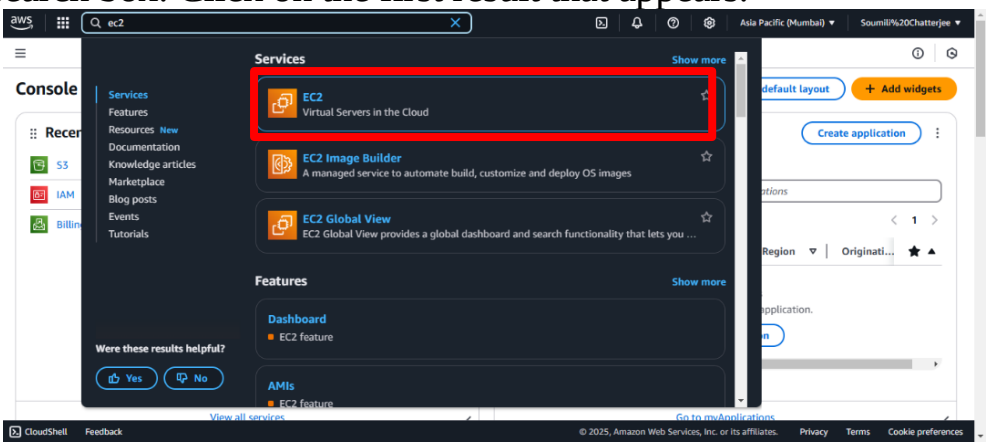
# Assignment No. – 7

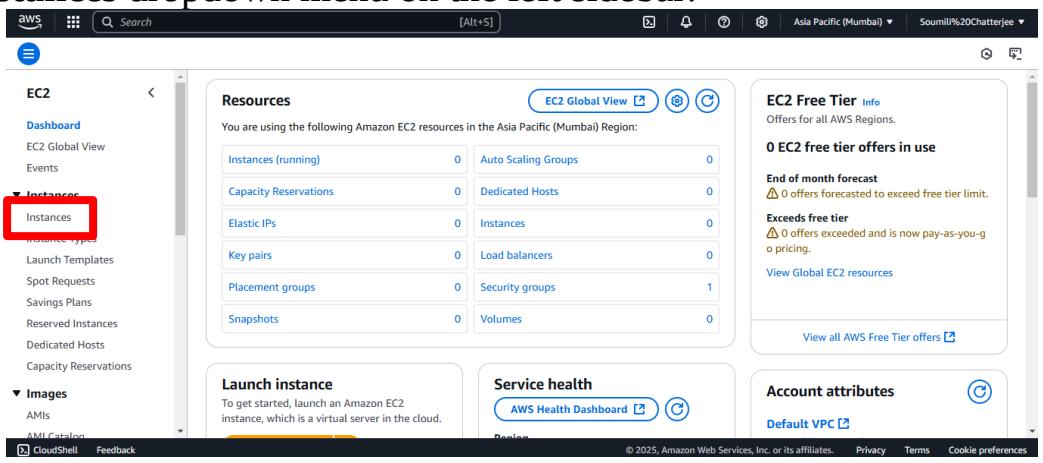
**Problem Statement:** Hosting a website on EC2.

## Procedure:

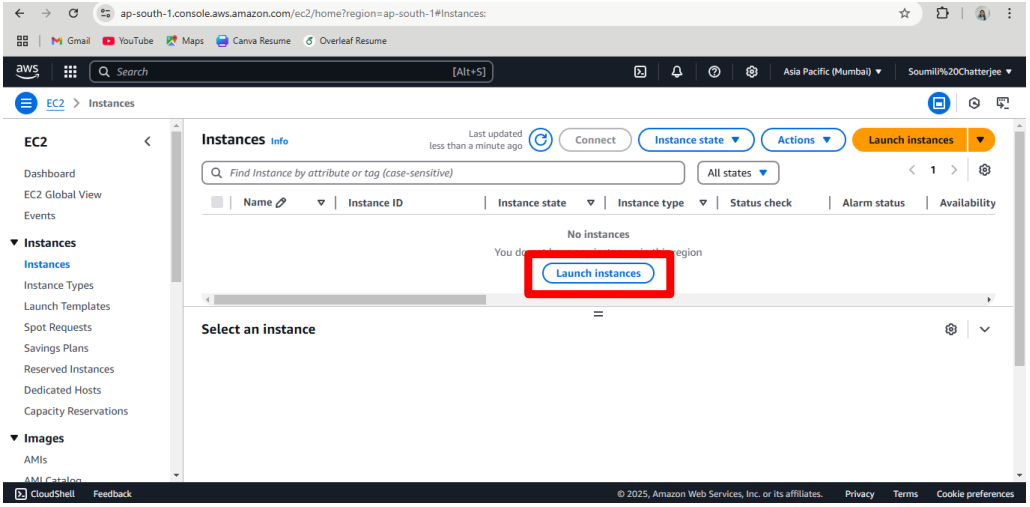
1. Login to your AWS account as a root user. Then search "**EC2**" in the search box. Click on the first result that appears.



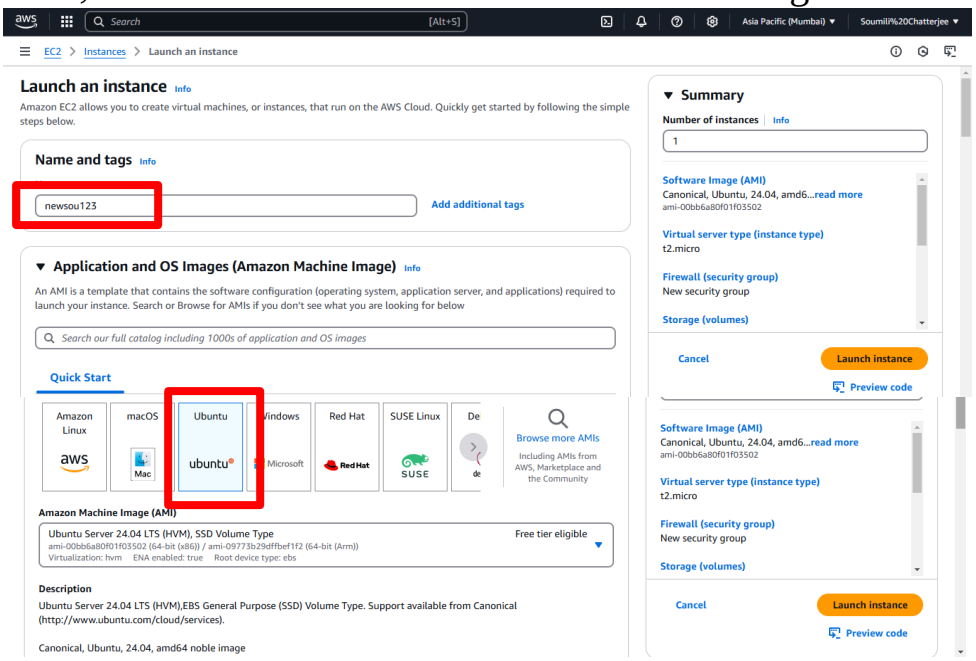

2. Click on "**EC2 Dashboard**" and then click on **Instances** under **Instances** dropdown menu on the left sidebar.

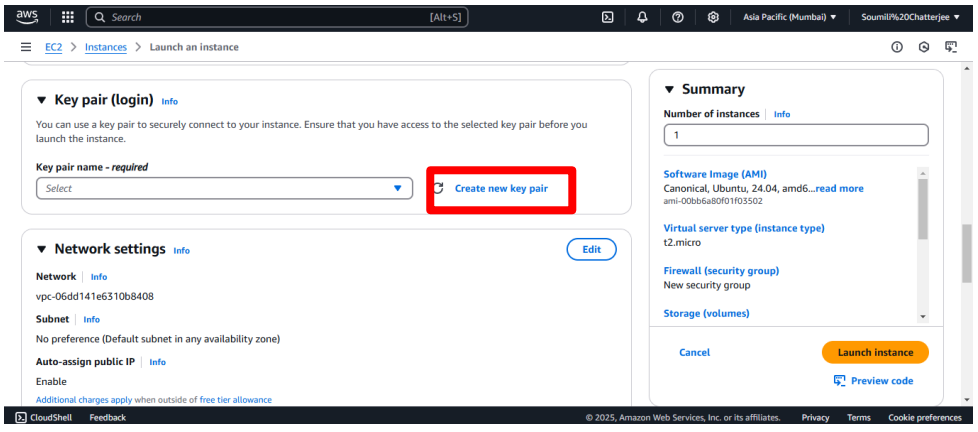


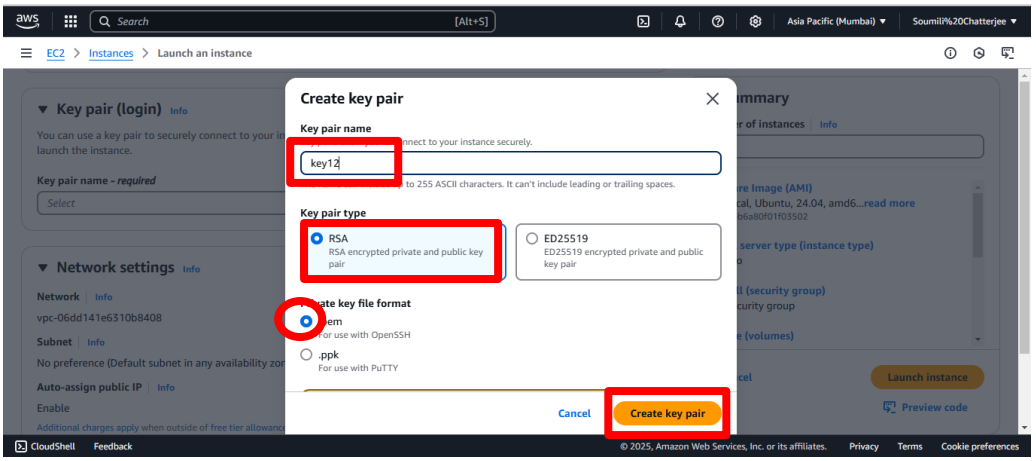

3. Next, click on **Launch instances** button.

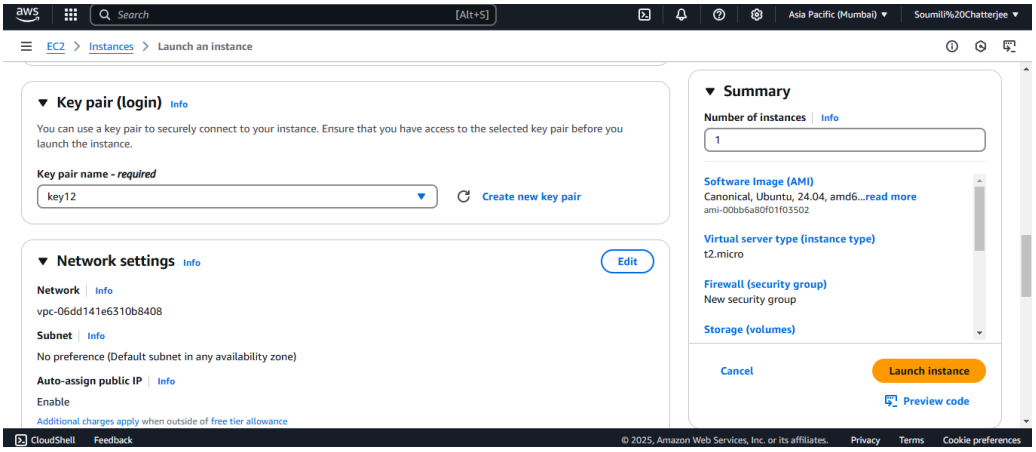


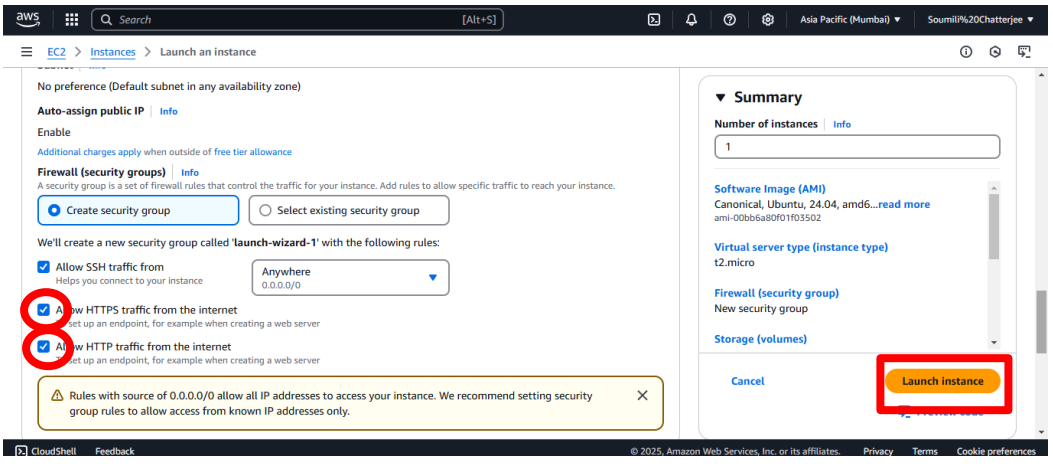
4. Now, customize the instance you want to launch:
    a. Set the **unique instance** name.
    b. Select **Ubuntu** as OS.
    c. Next, go to **key pair**(login) section:
        i. Click on **create new key pair.**
        ii. Enter the **name of key pair**.
        iii. Select **RSA** as **Key pair type**.
        iv. Select "**.pem**" as file format.
        v. Create the key pair.
        vi. A key pair file will be downloaded automatically. 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.



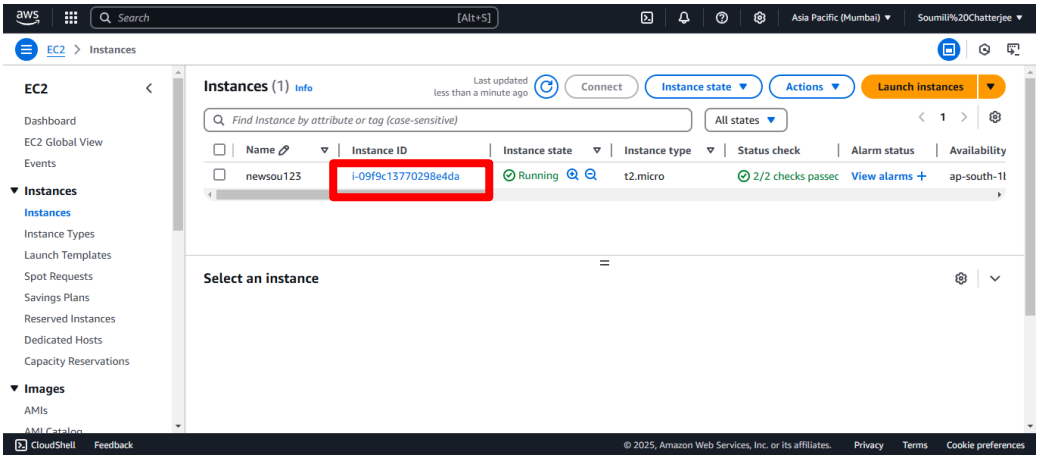





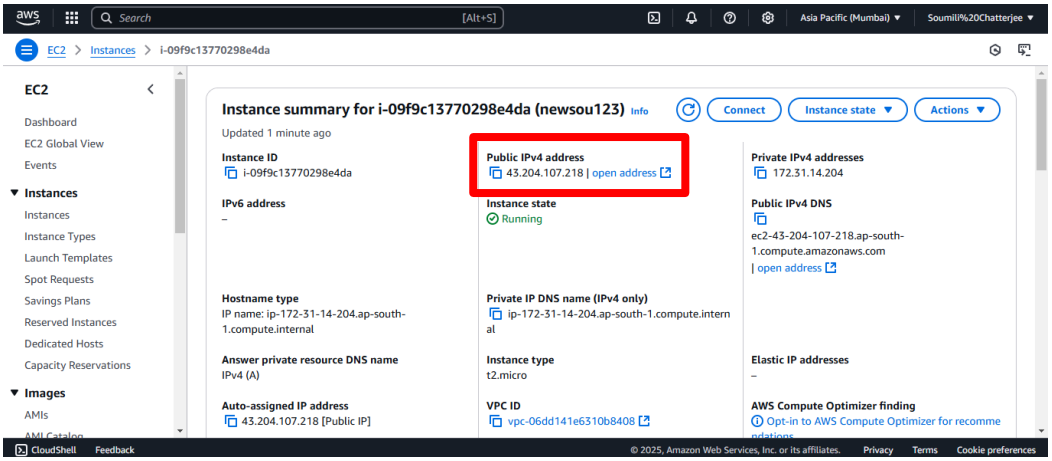




5. Check whether your **newly created instance** is running or not in the **Instances** page. Now, click on the **Instance ID** of the server.

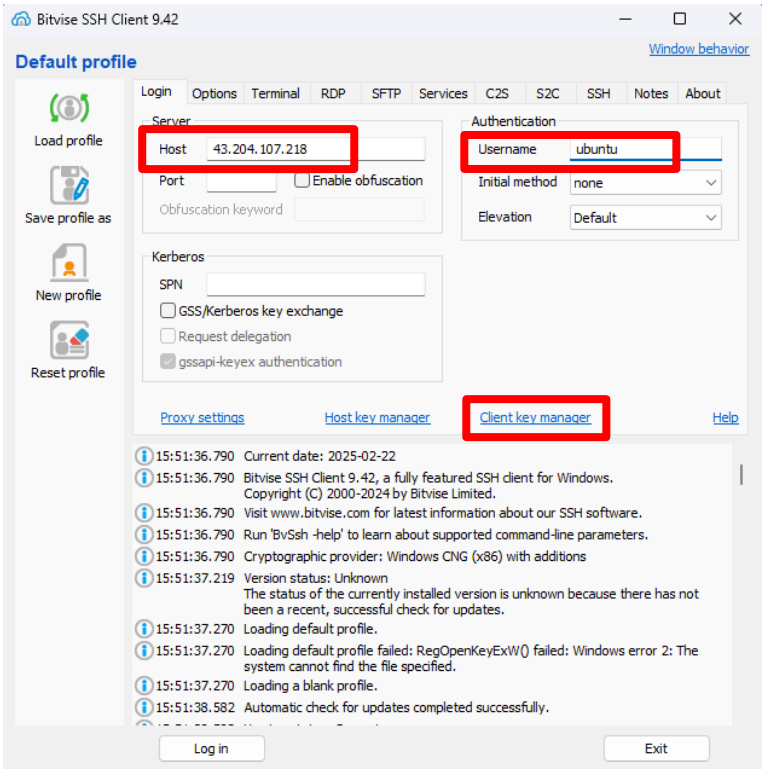


6. Copy the **Public IPv4** address.



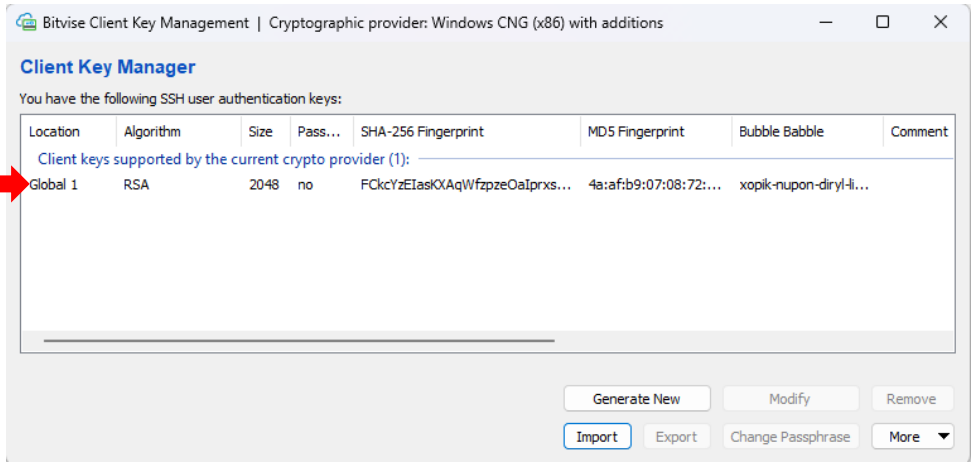

7. Now, for the next steps we require "**Bitvise SSH client**". Download and install it in your local computer.

8. Now, open the **Bitvise SSH Client**. Paste the **copied IPv4** address in the **Host** section. Set **user name** to **ubuntu**. Then, click on the **client key manager** link below the authentication section.

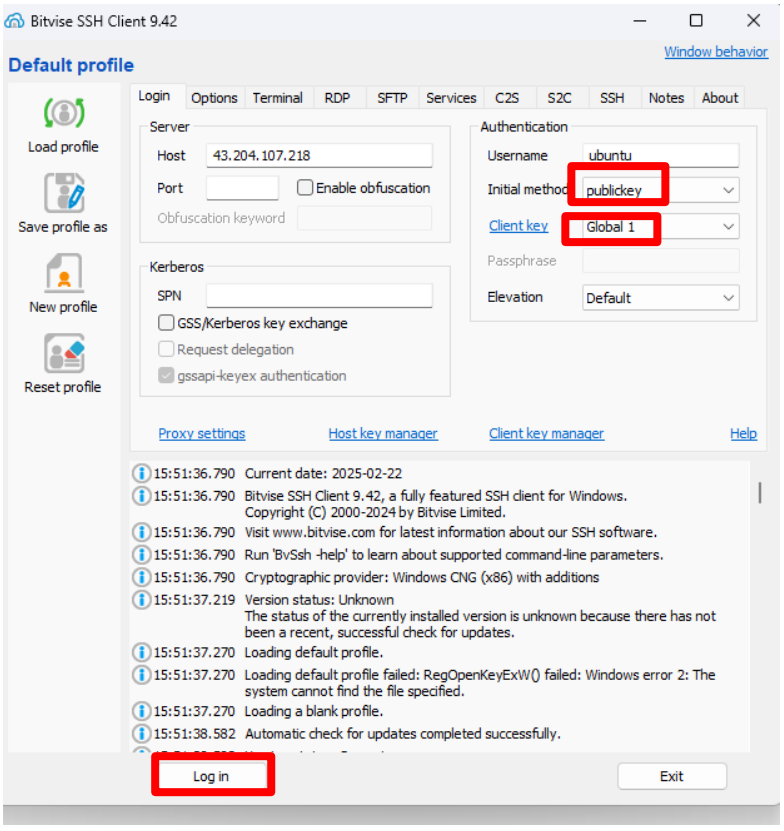



9. It will open another pop-up window. There click on **import** button.

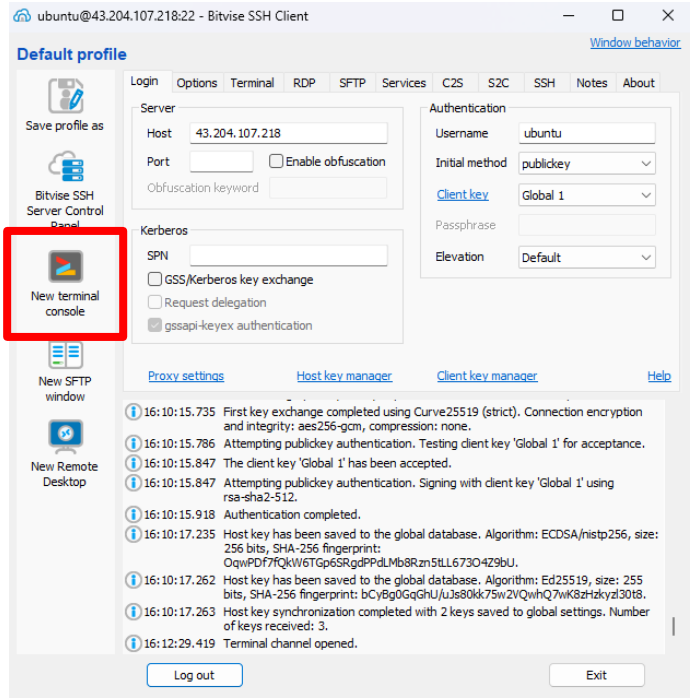
Select the previously downloaded **.pem** file. Click on **import**. Then close the Client key manager window.




10. Now, set **initial method** to **public key**. Set **Client Key** to **Global 1**. Now, click on the **Log In** button at the bottom of the Window.




11. Click on **Accept and Save** button on the pop-up. 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.




12. Enter the following commands:

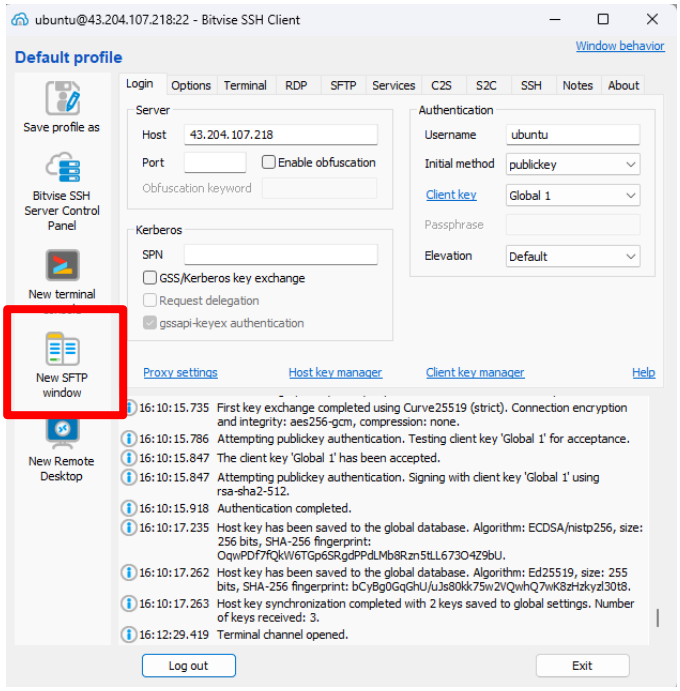
    a. **sudo apt-get update**

    b. **sudo apt-get upgrade**

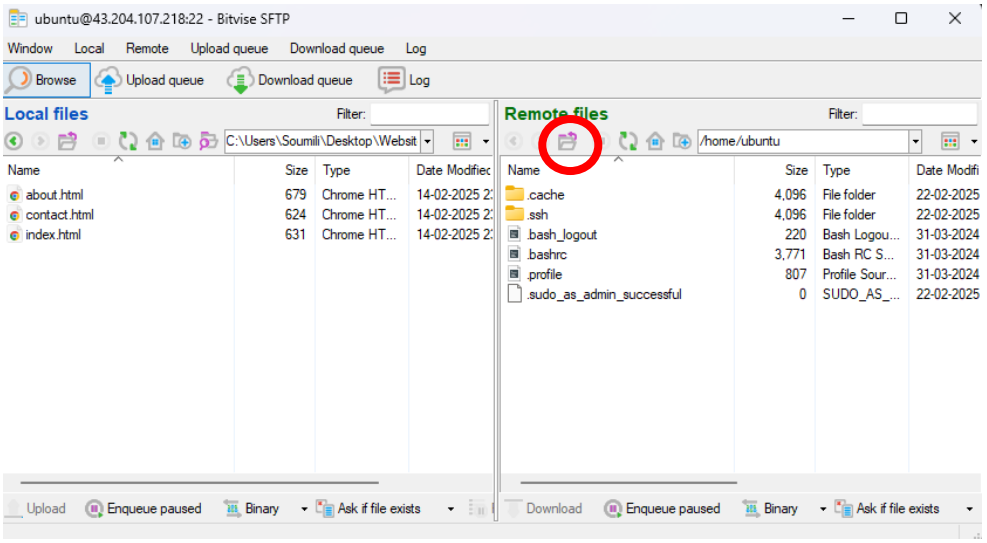
    c. **sudo apt-get install nginx**

    [Remember to **press Y** and then **Enter when prompted**.]

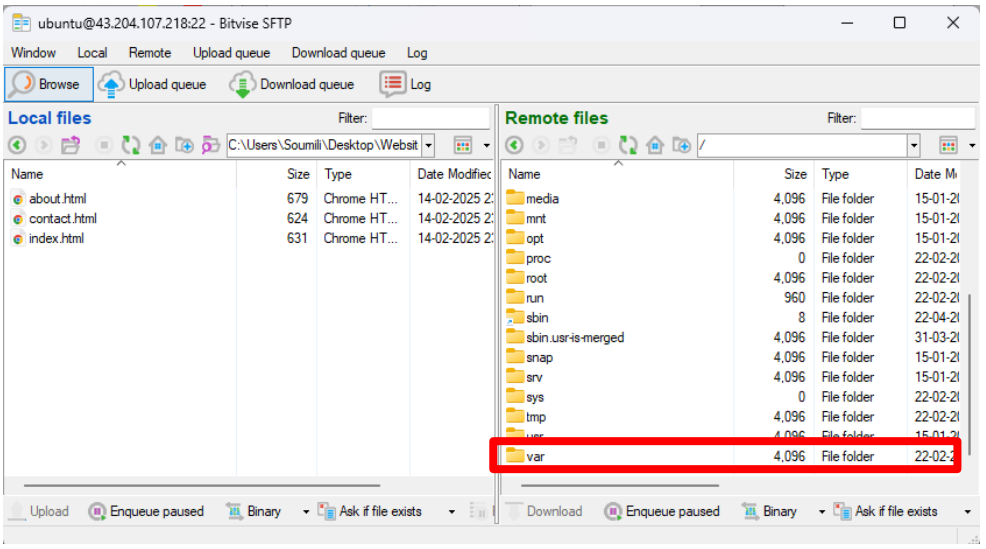
13. Now, minimize the console. Click on the **new SFTP window** icon on the left sidebar.



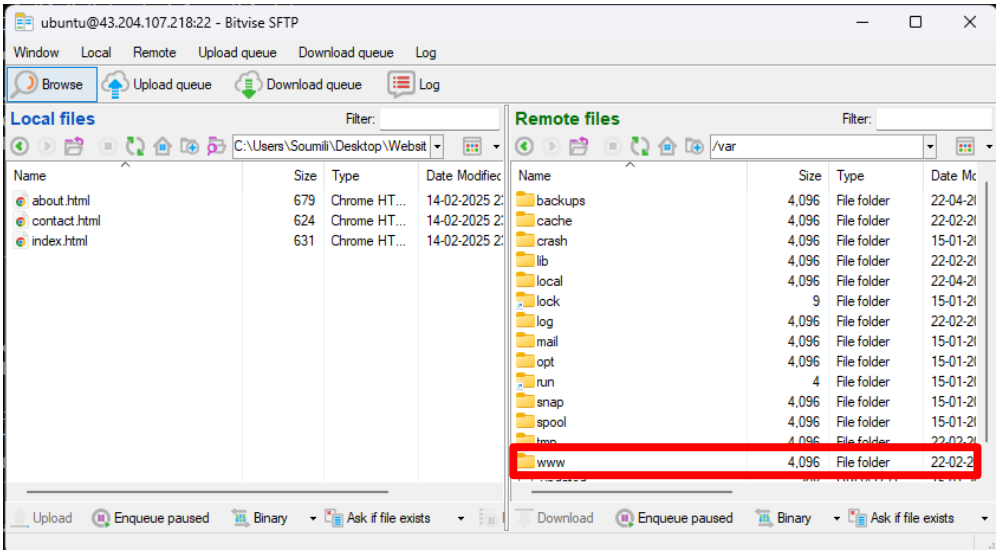

14. Select the folder where you have kept HTML files of your website on the local files section. Just keep it open. Now, click the **Up button** (2 times) on the Remote Files section.



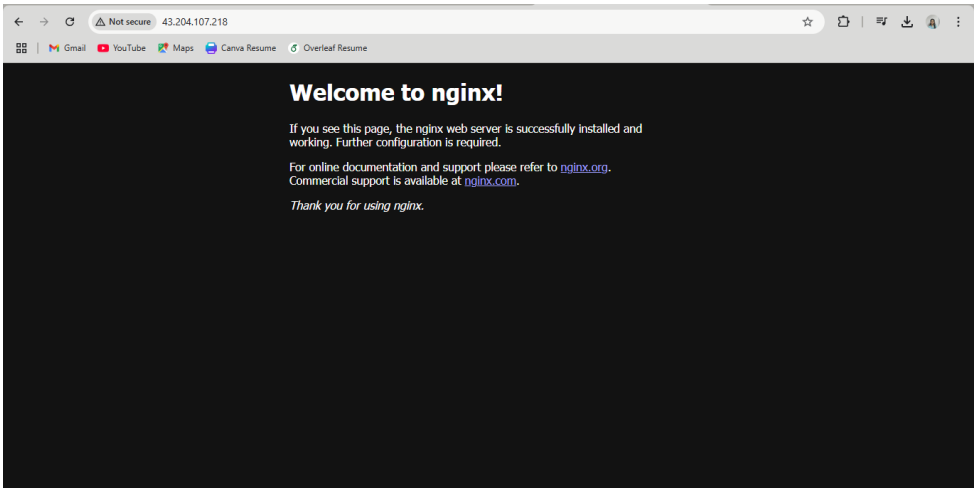

15. You will be able to see a bunch of folders. Scroll down and open the last folder named "**var**".




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



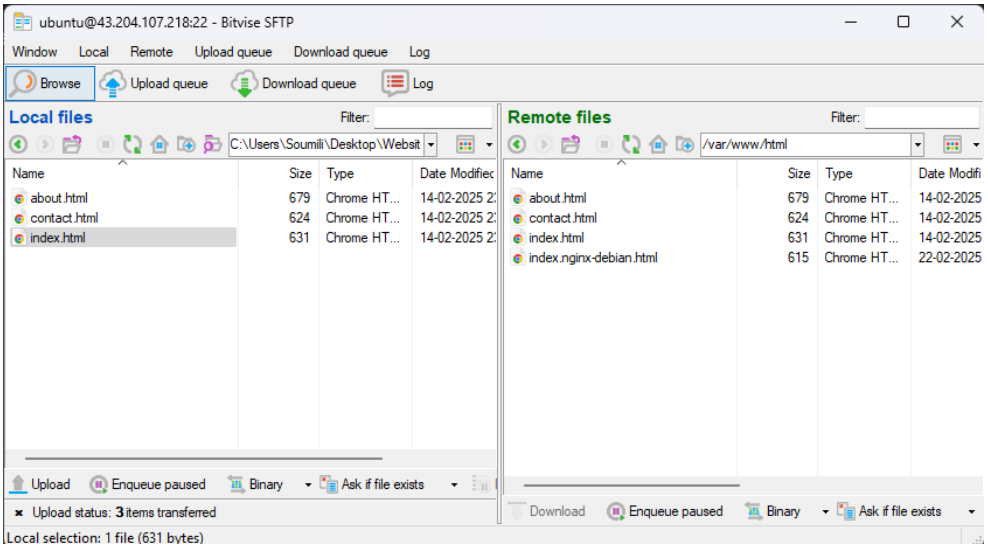

17. Open the only folder named "**html**" and keep it open. You will see a **default html** already present. You can check whether nginx is working by pasting our previously copied IPv4 address of our server instance in a different browser. It will show something like this:



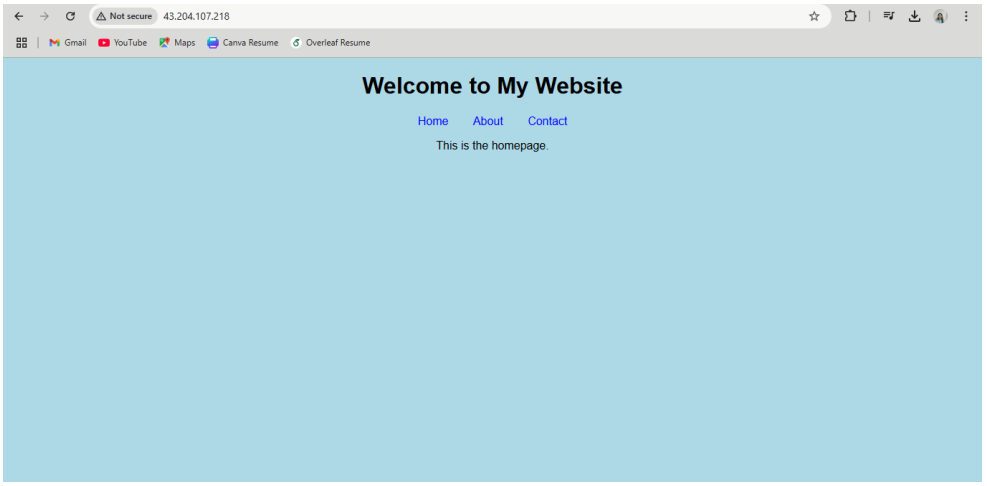

18. We actually need to transfer our local html files in the open folder of the remote server. However, we do not have such permissions for this folder. To give such permission we need to go back to the terminal console and give the required permissions to the folder.
19. Now type the following commands in the terminal.
    a. **cd /**
    b. **cd var/www/**
    c. **sudo chmod 777 html**

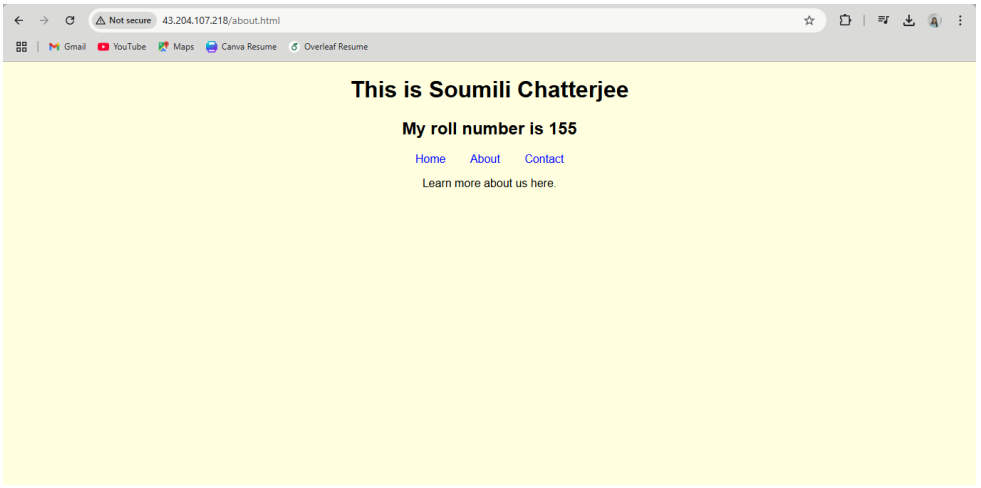
Now the permission (**Read, Write, Execute**) of the folder is successfully granted.

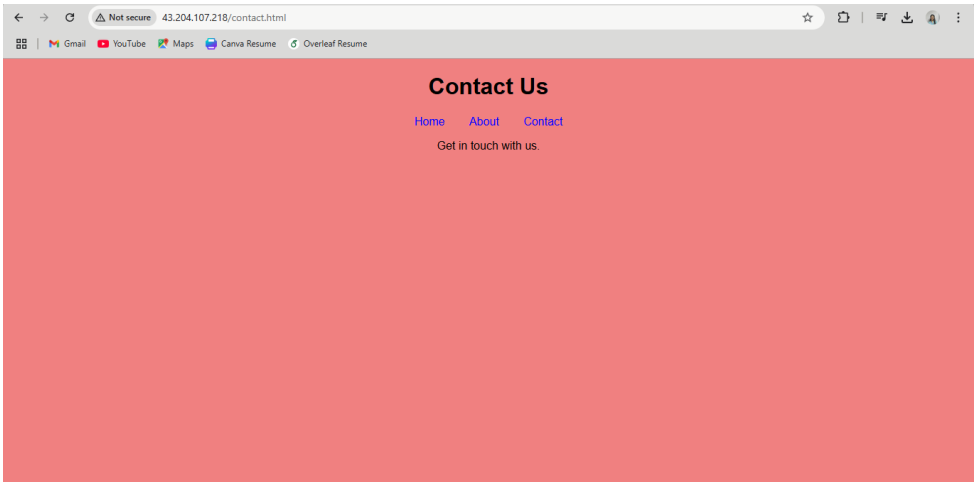
20. Now, drag and drop **all the files** from **local** to **remote**.




21. Finally, open the website from any browser or device by using the **public IPv4 address** that is copied.




We now have successfully hosted a website on **AWS EC2 server**.