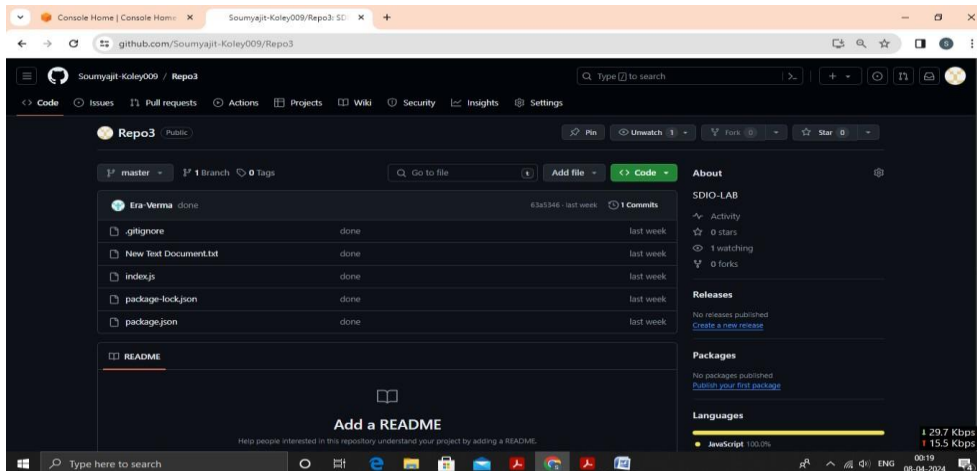


ASSIGNMENT No:- 9

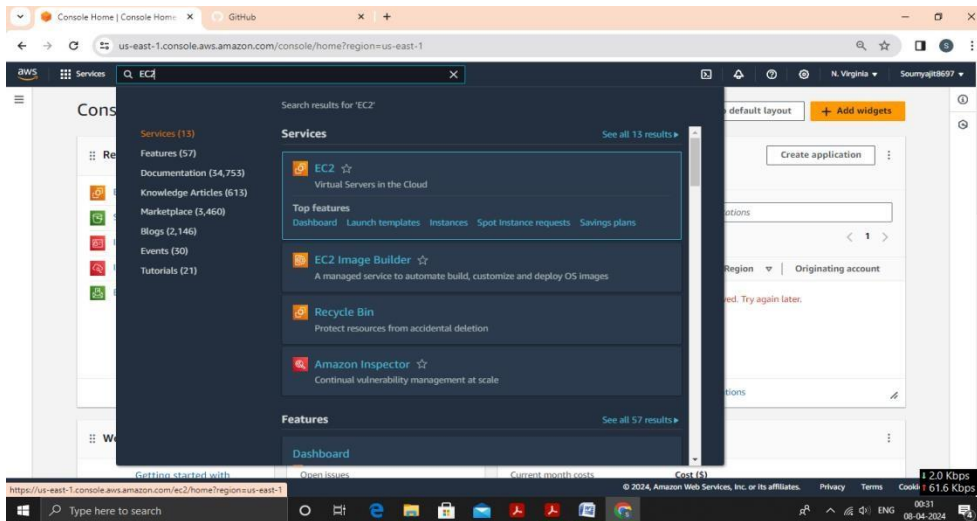
Problem Statement: Deploy a project from GitHub to EC2.

The steps are as follows: -

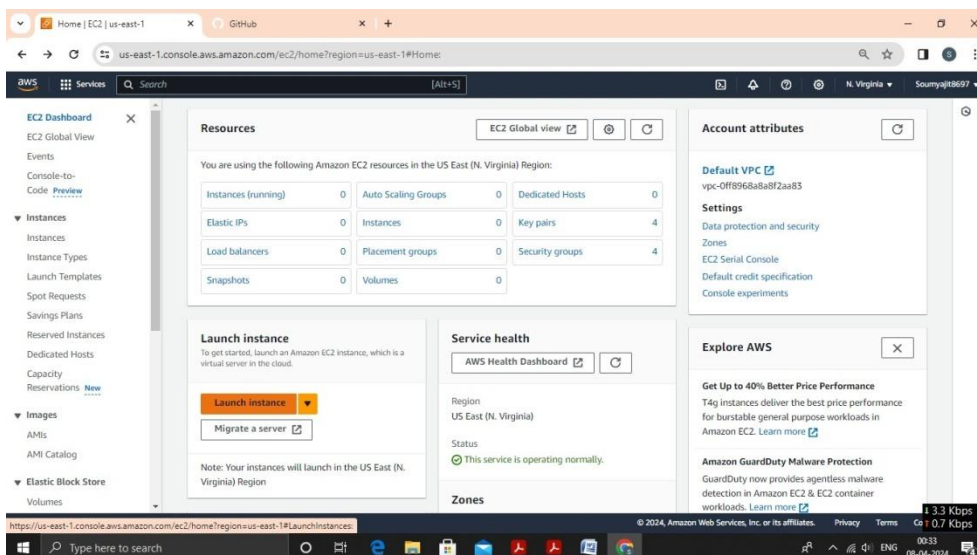
1) At first upload file(repo) to github.



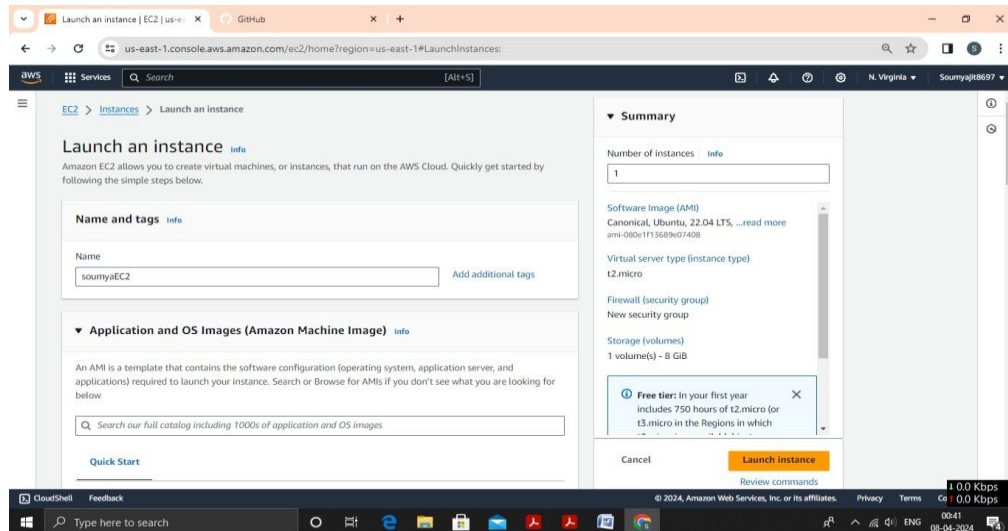
2. Access your AWS account and utilize the search box to look for EC2, then proceed to click on the first option displayed.



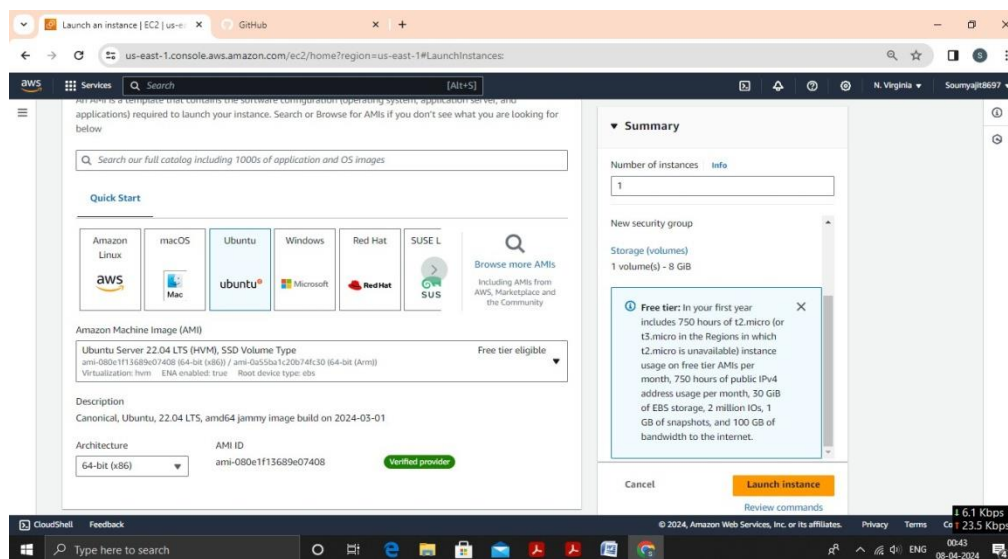
3. Click on “Launch instance”.



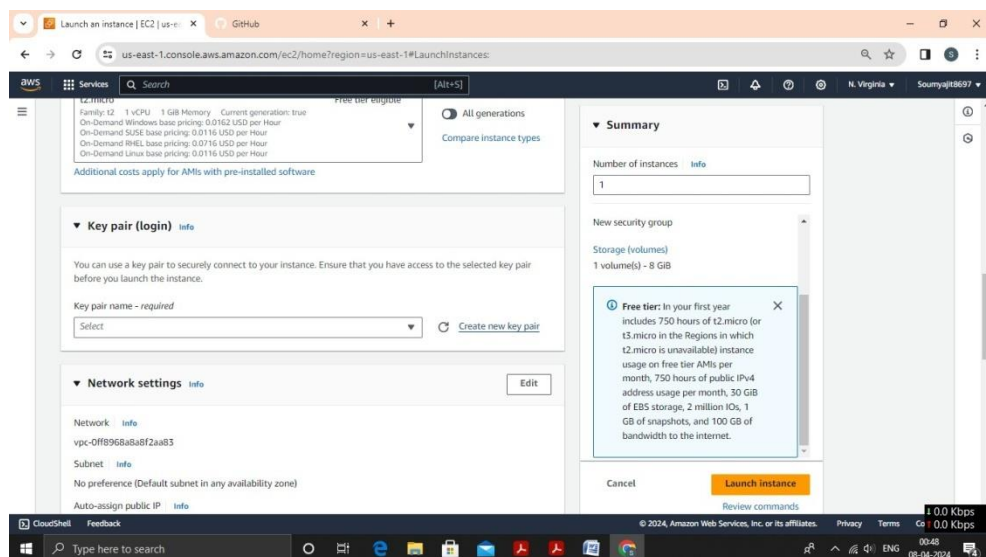
4. For the subsequent step, ensure to assign an appropriate name to the EC2 instance. (For instance, here it is -soumyaEC2)



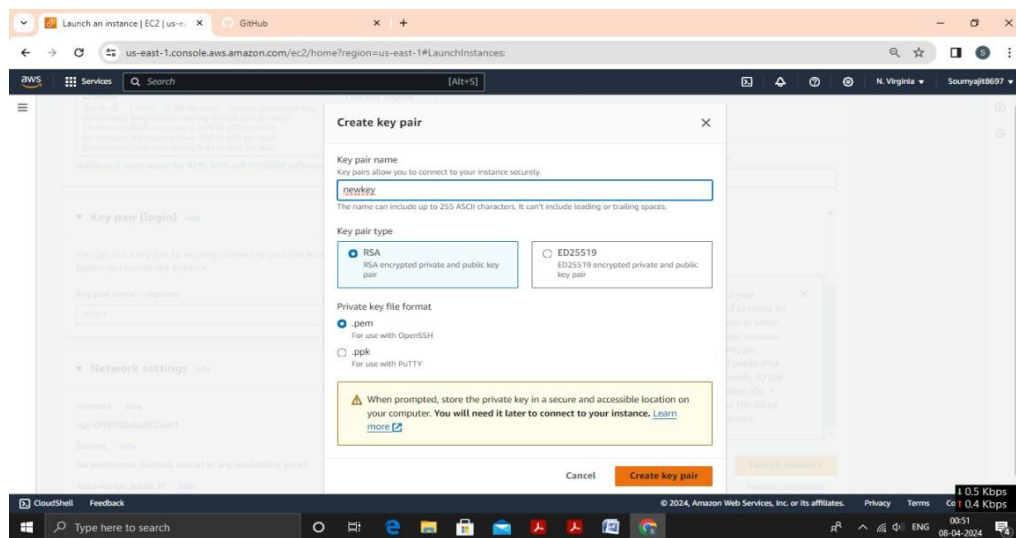
5. Choose **Ubuntu** from the available AMIs.



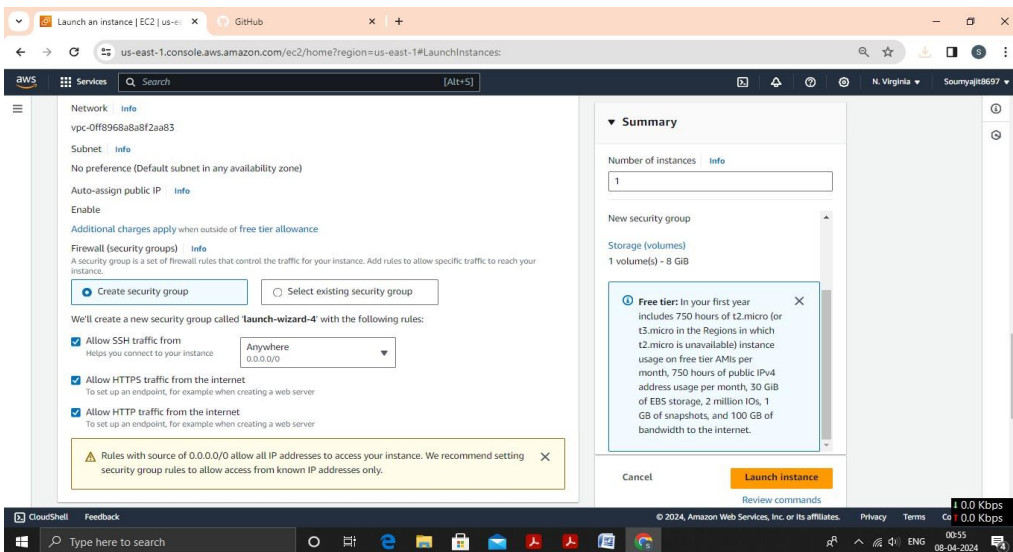
6. Generate a new key pair or alternatively, utilize an existing one. A new key pair is generated by selecting "Create new key pair".



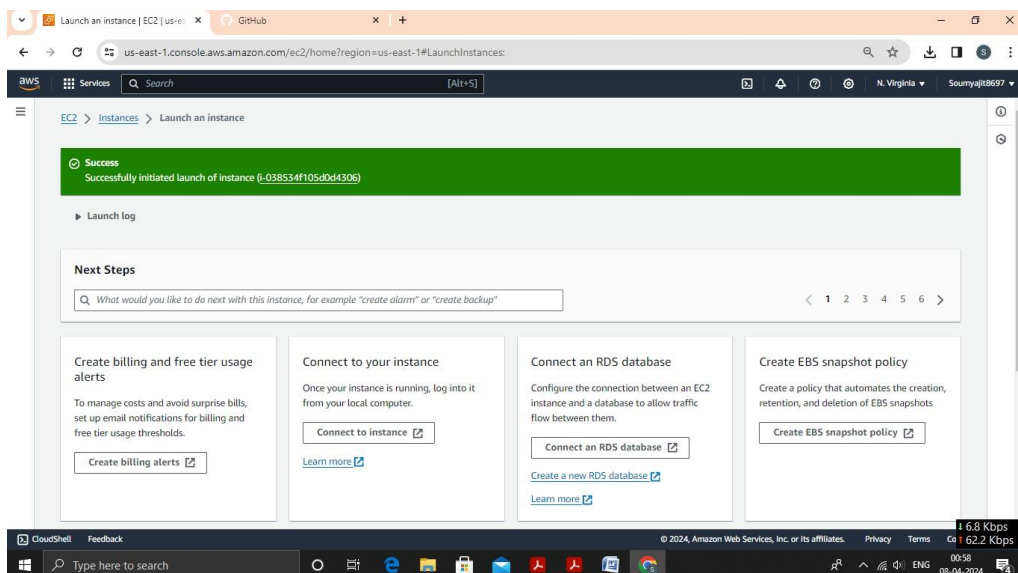
7. Then click on "create key pair".



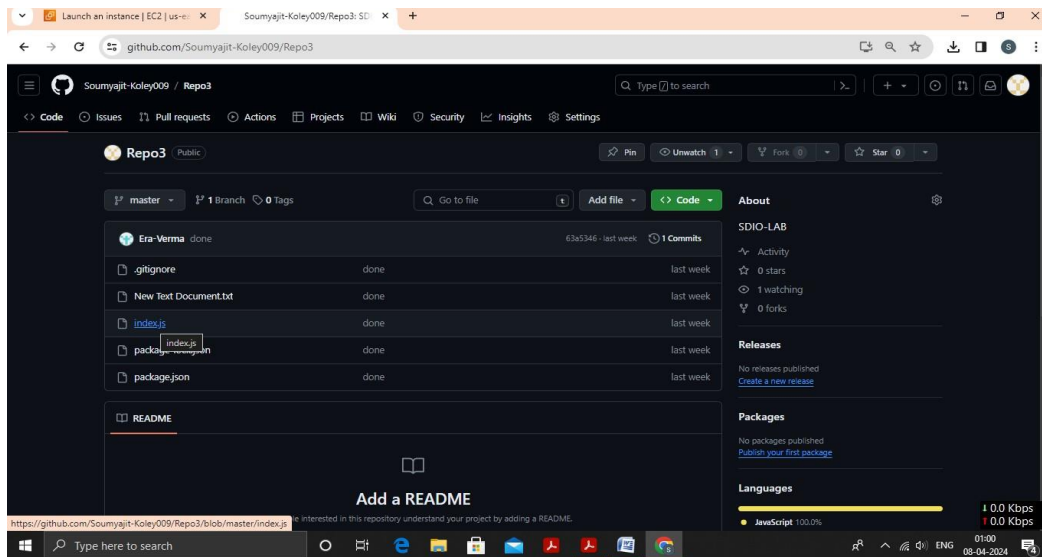
8. Select to create a server and ensure to enable all three protocols: SSH, HTTP, and HTTPS.



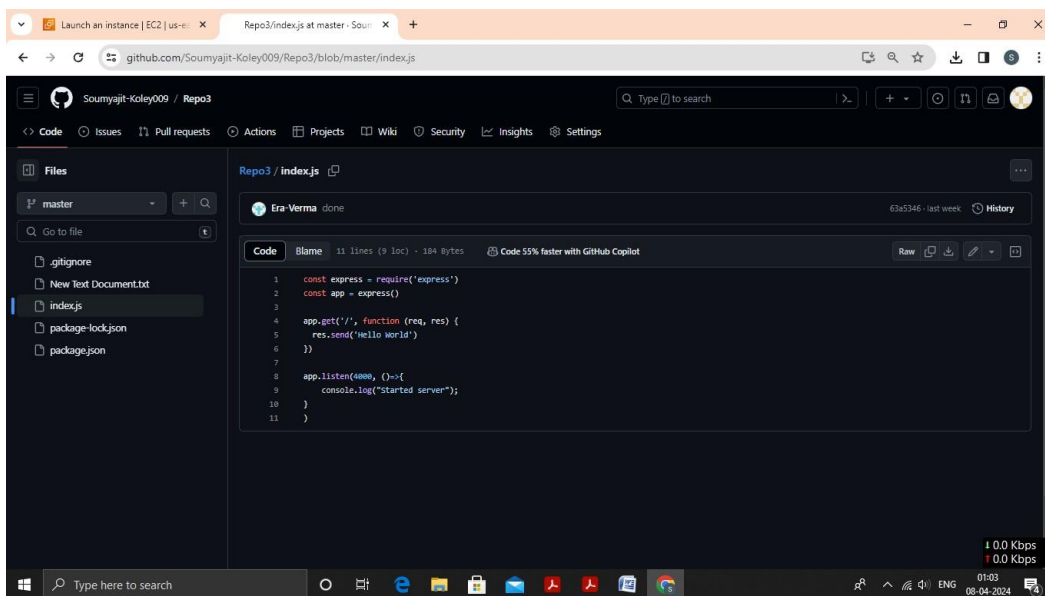
9. Finally, proceed by clicking on "Launch Instance".



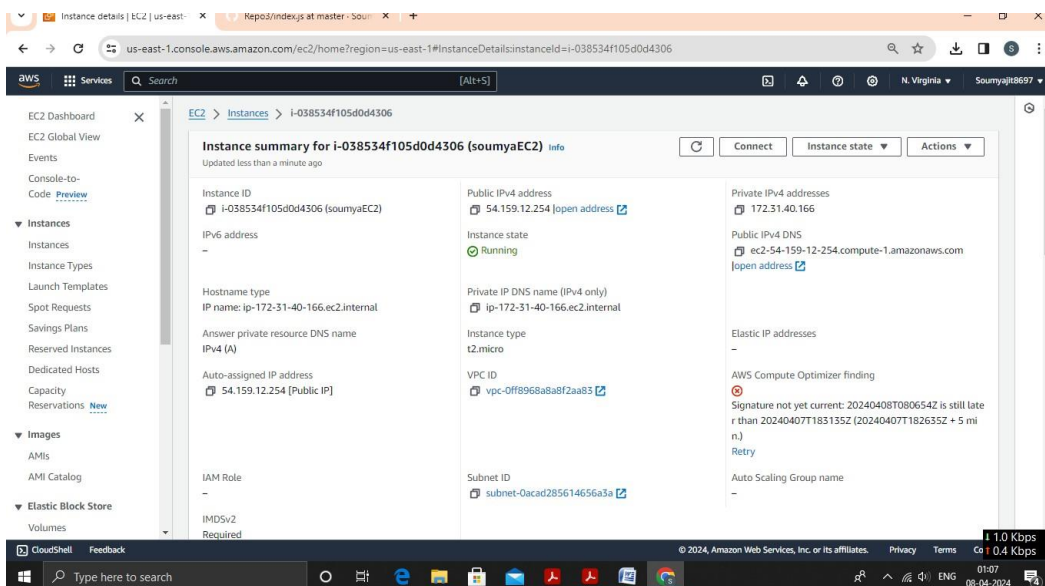
10. Access GitHub and navigate to your repository that is currently being worked on, containing the `index.js` file.



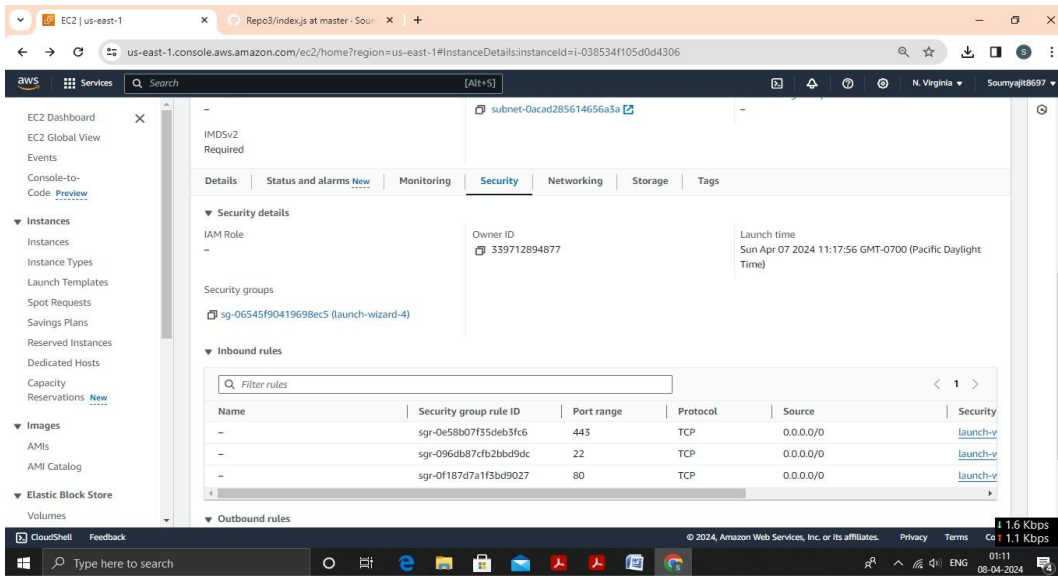
11. Access the "index.js" file.



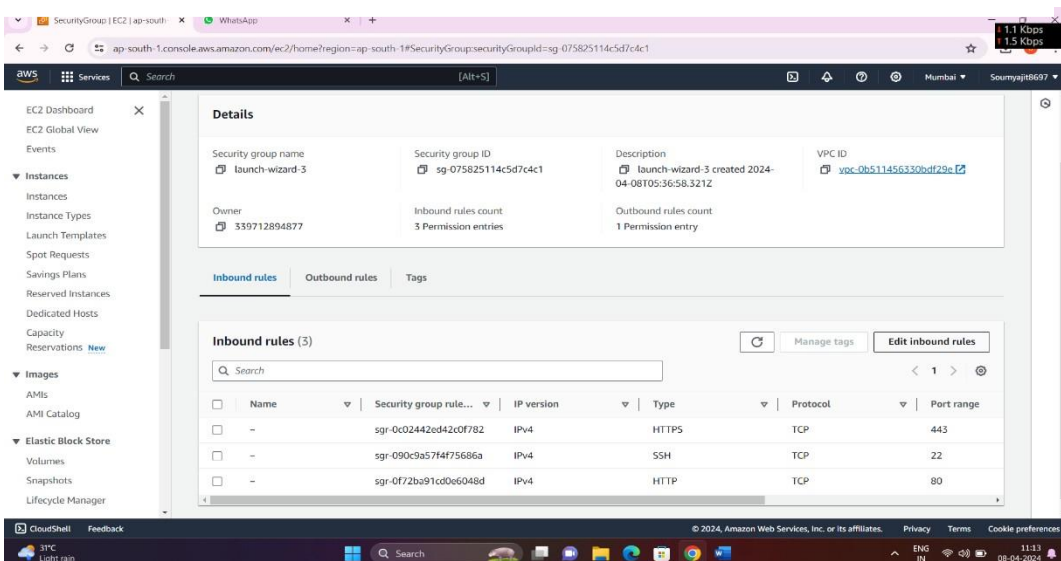
12. Return to the AWS tab currently open in your system, then navigate to **instances** and select **security**.



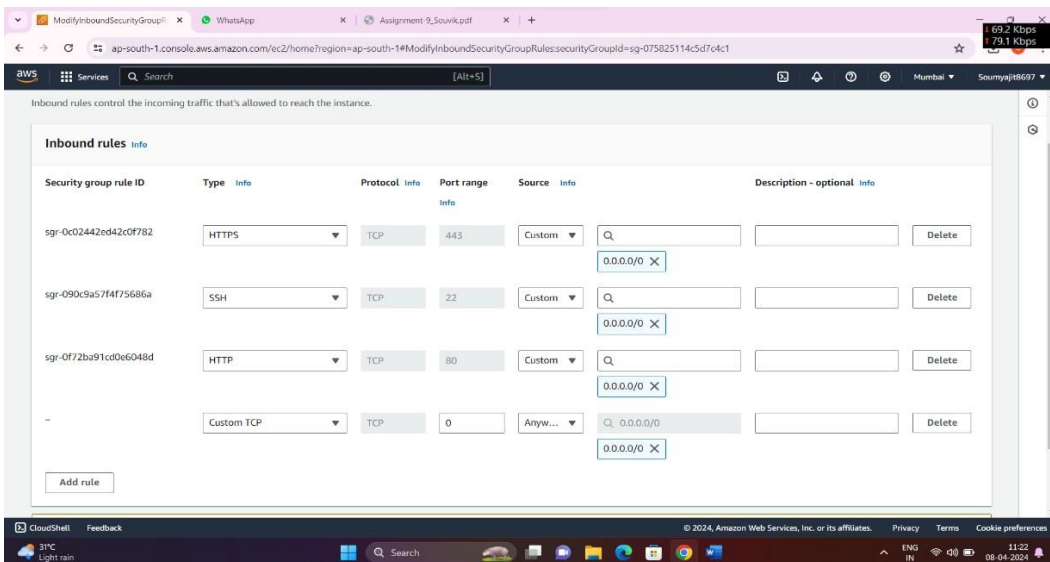
13. Select the security group and observe that three protocols have been chosen.



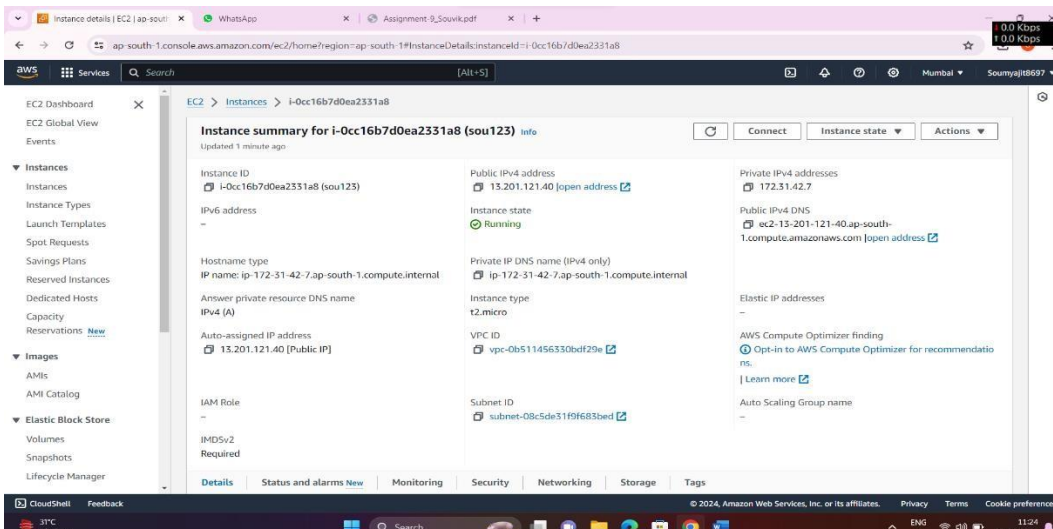
14. Select "Edit inbound rules."



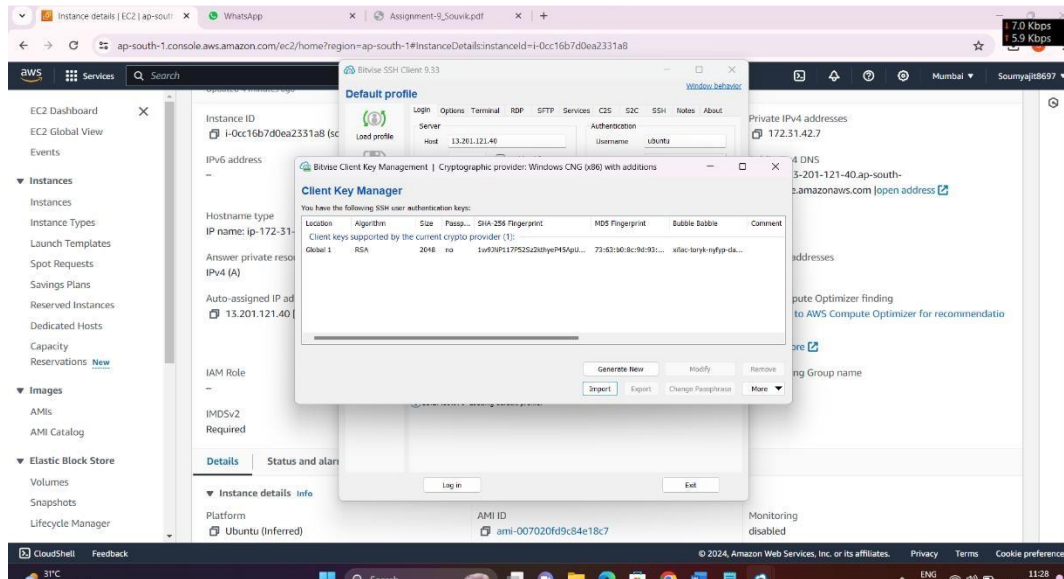
15. Then in inbound rules, click on "Add rule" select Custom TCP ,give port range 4000 and select 0.0.0.0/0. After it click on Save rules right side bottom corner.



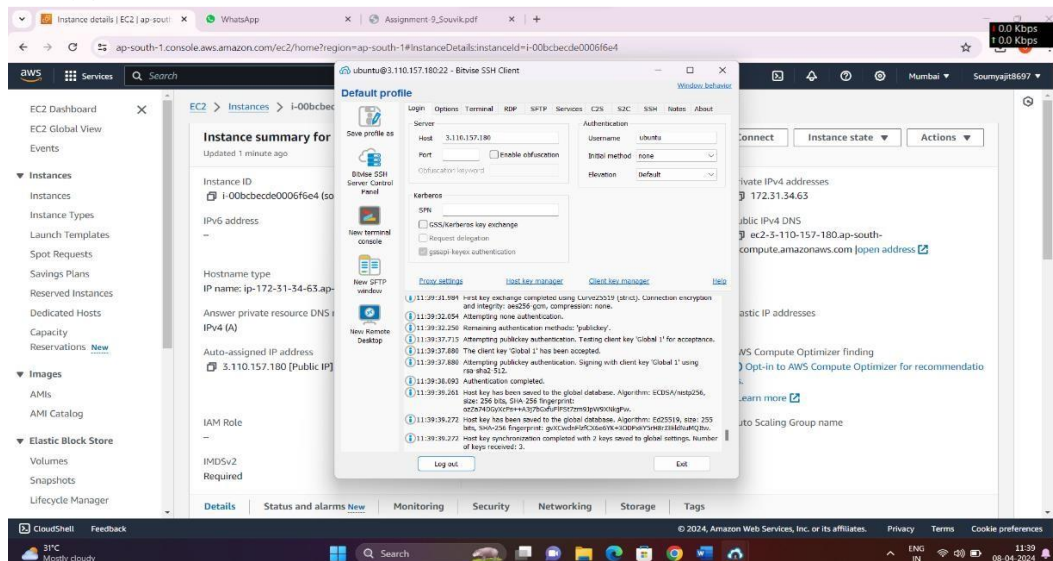
16. Now go back to EC2 recently created instance and copy the public IPv4 address.



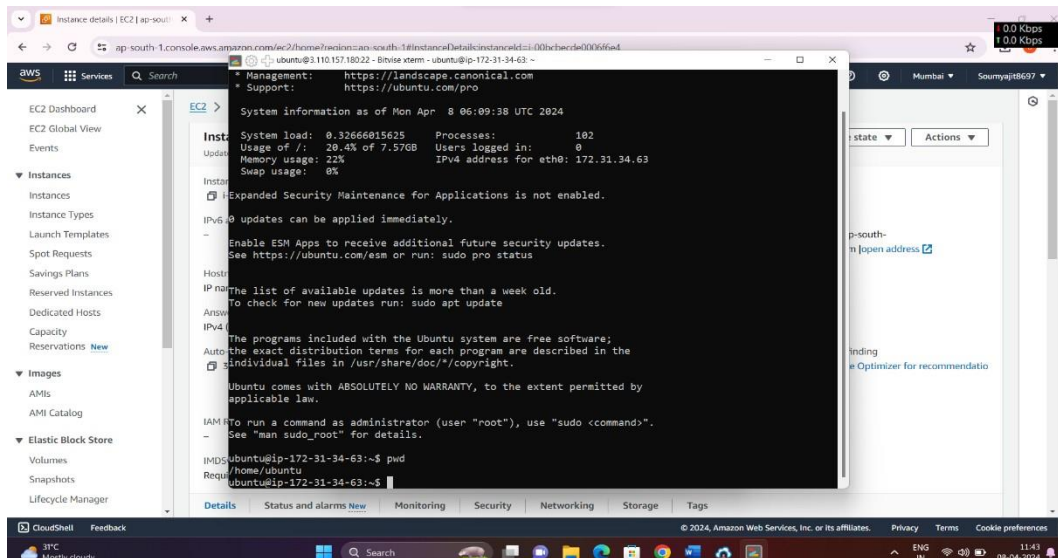
17. Now open Bitvise SSH Client, go to Client key management and import that downloaded or saved (previously) key.



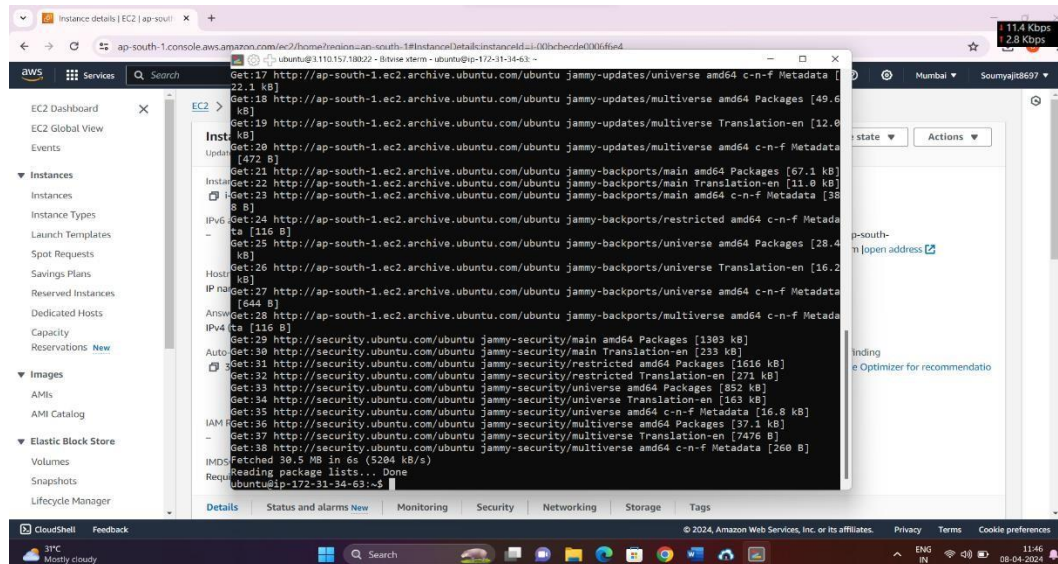
18. After that paste copied IPv4 public address in host and then login and also do Accept and Save, also set publickey, global1 then click on ok.



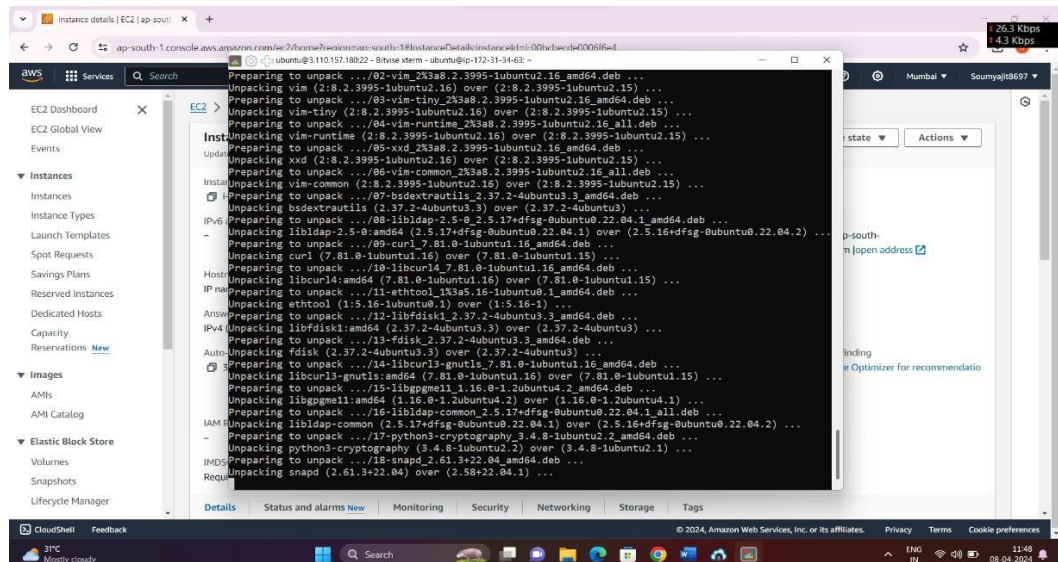
19. Now open a new terminal and write pwd to check working directory, we are in ubuntu.



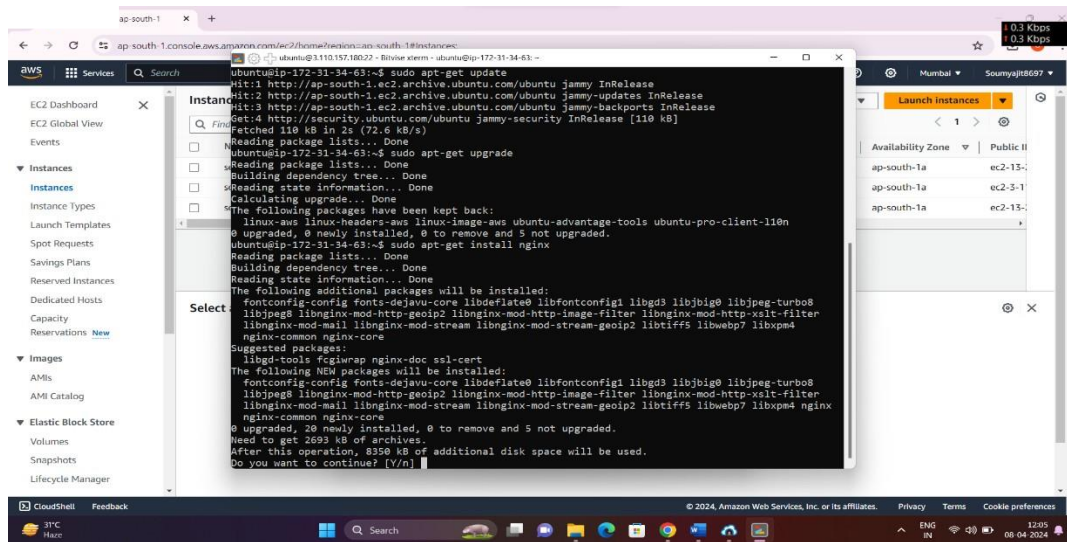
20. Now write command sudo apt-get update to fetch all packages



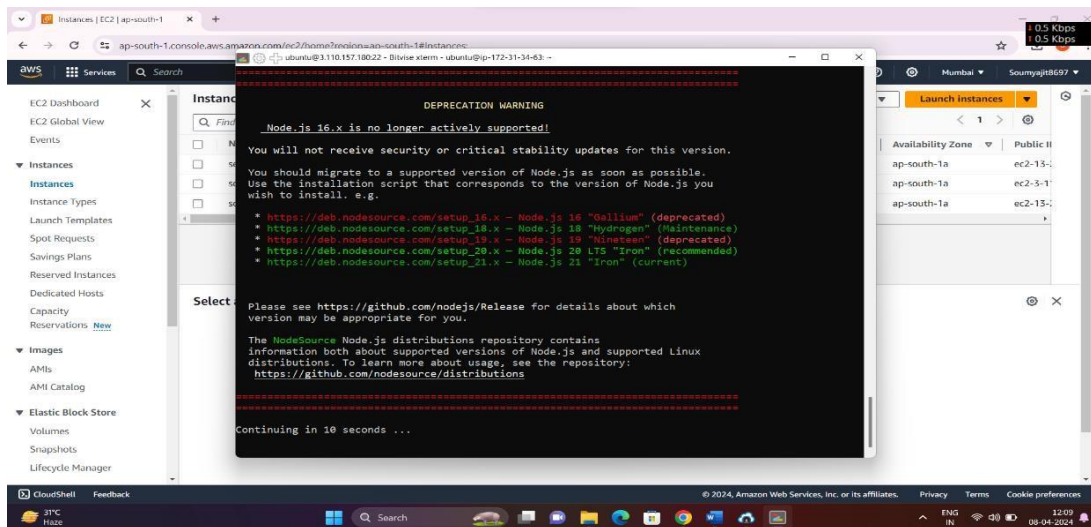
21. Write sudo apt-get upgrade to upgrade all outdated packages.



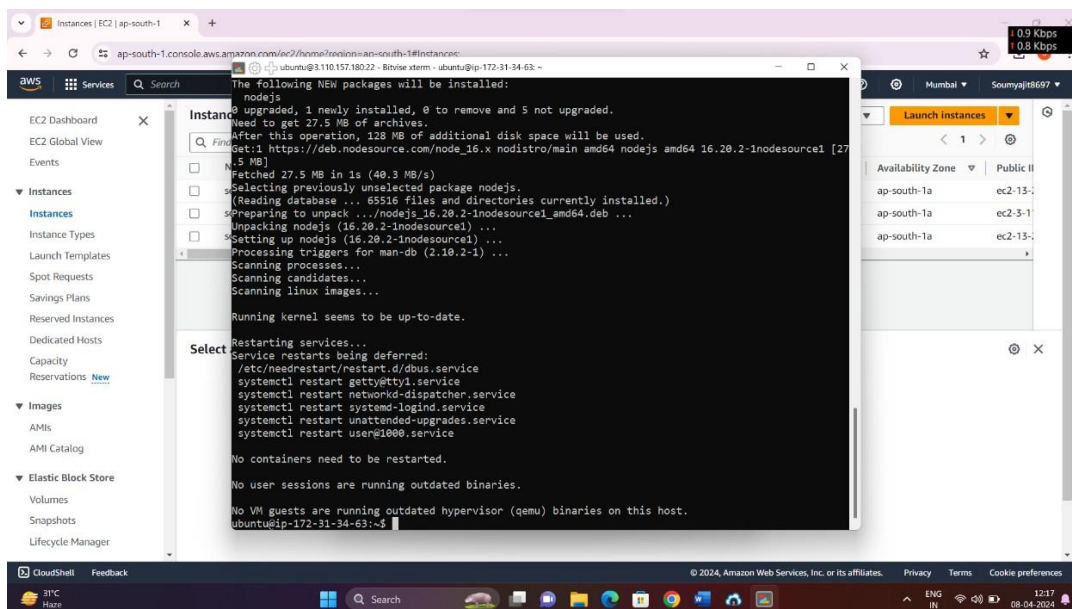
22. Now write command `sudo apt-get install nginx` to install webserver.



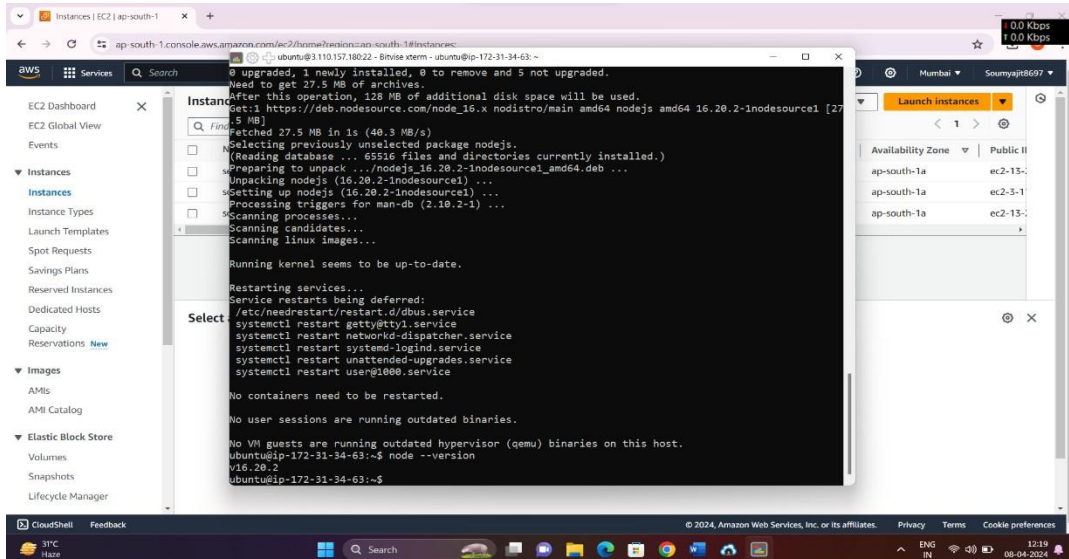
23. Now to execute javascript, we need to install nodejs. For this just write the command `curl -sL https://deb.nodesource.com/setup_16.x | sudo -E bash -`



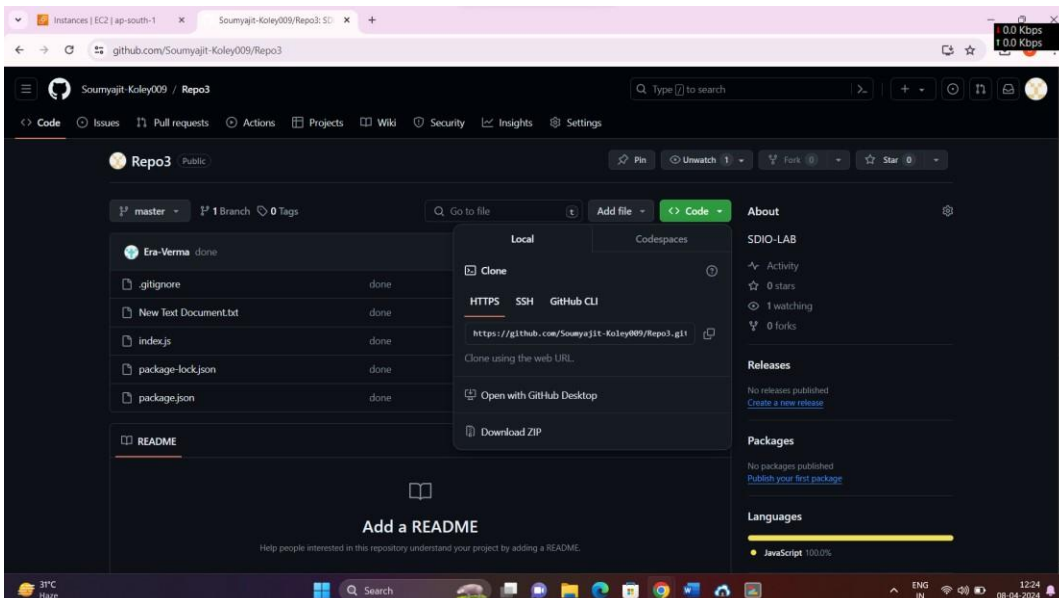
24. Now to install nodejs write the command `sudo apt install nodejs`.



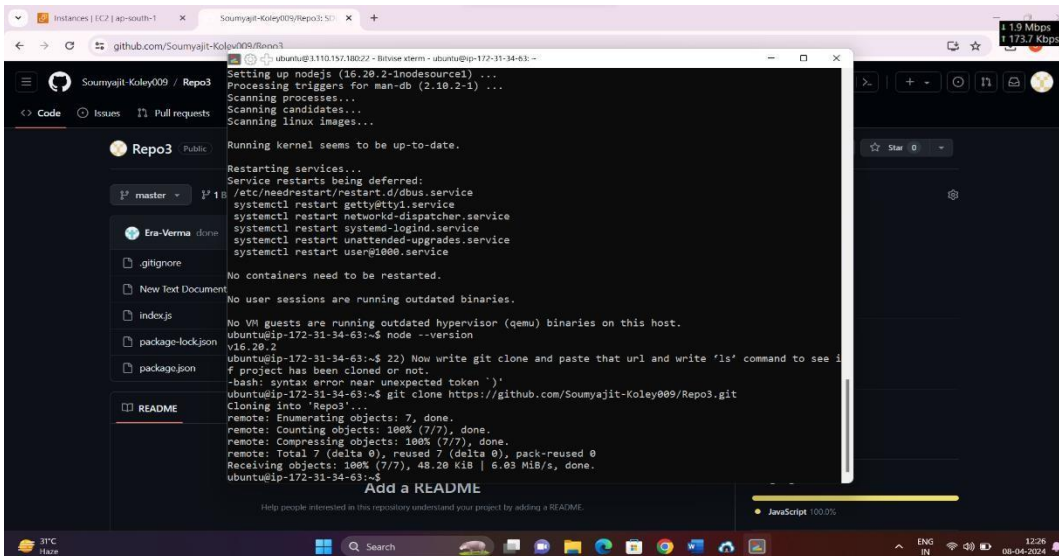
25. Write `node --version` to see what version of nodejs installed.



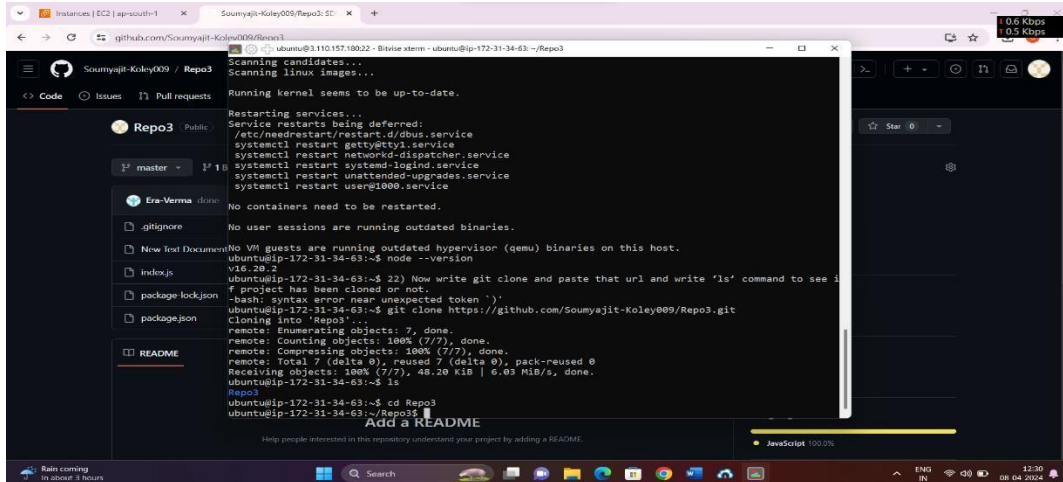
26. Now go back to github enter into repository and in Code section copy Https url.



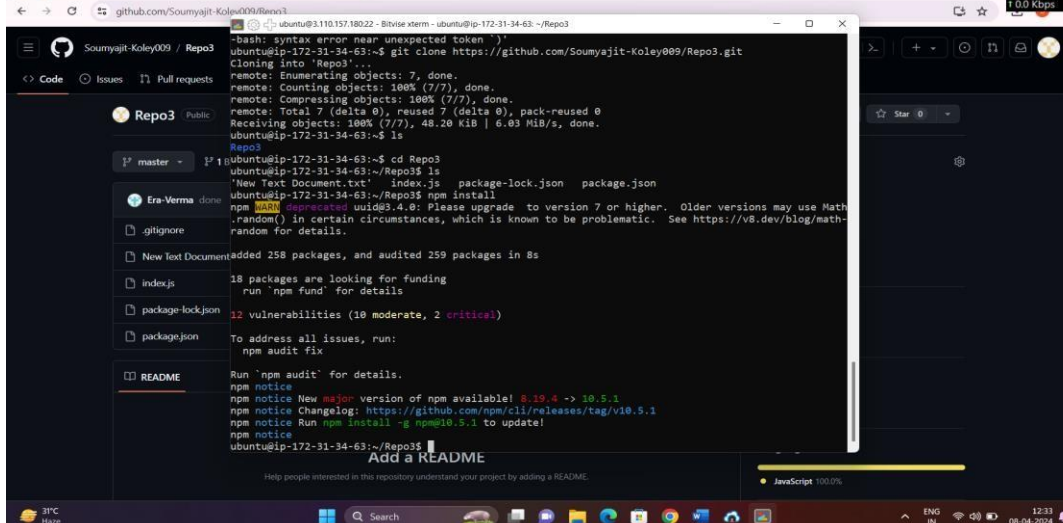
27. Now write git clone and paste that url and write 'ls' command to see if project has been cloned or not.



28. After it write command `cd (project name)` to enter into project and then `ls` to see what files have been uploaded.



29. Now to execute node command we have to install node packet manager(npm). So write `npm install`.



30. Now write `node index.js`. Server will started.



31. After it again copy public IPv4 address in EC2 instance and then paste it in another tab (Incognito) url section. We get "Welcome to nginx!"



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

32. At last in url at end write `:4000` to get our website .



33. Now to close server in new terminal do `(ctrl+c)` to stop server and at last logout. In this way we have deployed a project from GitHub to EC2