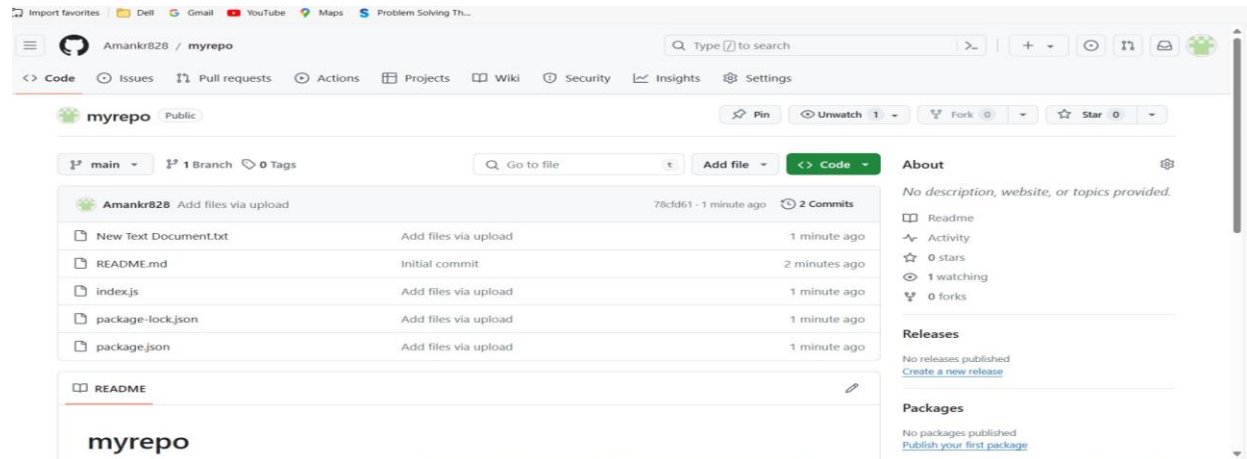


Assignment No : 9

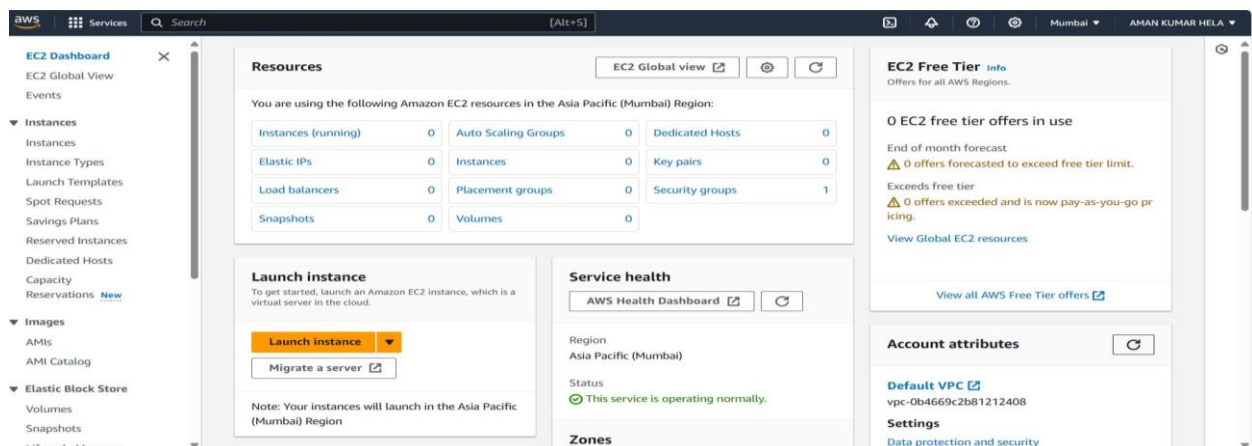
Problem Statement : Deploy a project from GitHub on EC2.

Steps :-

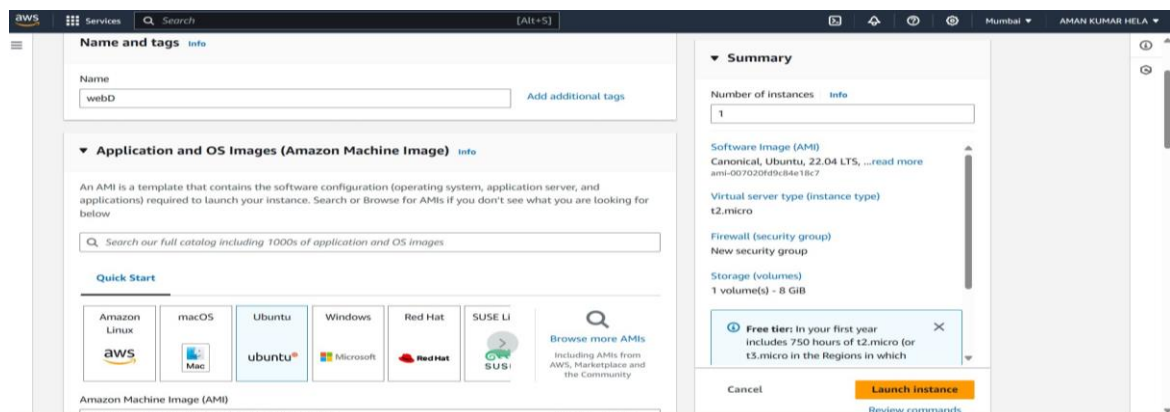
1. At first upload files(repo) to GitHub.



2. Now open EC2 and go to Launch Instance.



3. Now give name of server which should be unique and select Ubuntu application and OS image inside quick start.



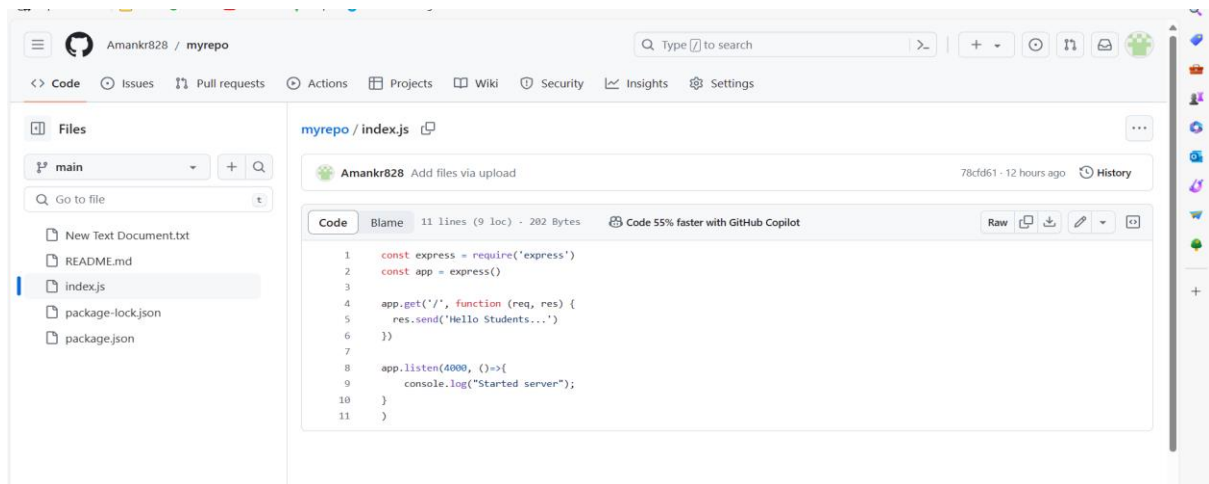
4. Now generate new key pair, give name(unique) and now click Create Key Pair and also download the document.

The screenshot shows the 'Create key pair' dialog in the AWS Management Console. It has a title bar with a close button. The main content area includes a 'Key pair name' section with a text input containing 'Ass009' and a note about character limits. Below is the 'Key pair type' section with two radio buttons: 'RSA' (selected) and 'ED25519'. The 'Private key file format' section has two radio buttons: '.pem' (selected) and '.ppk'. A yellow warning box at the bottom states: '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'. At the bottom right are 'Cancel' and 'Create key pair' buttons.

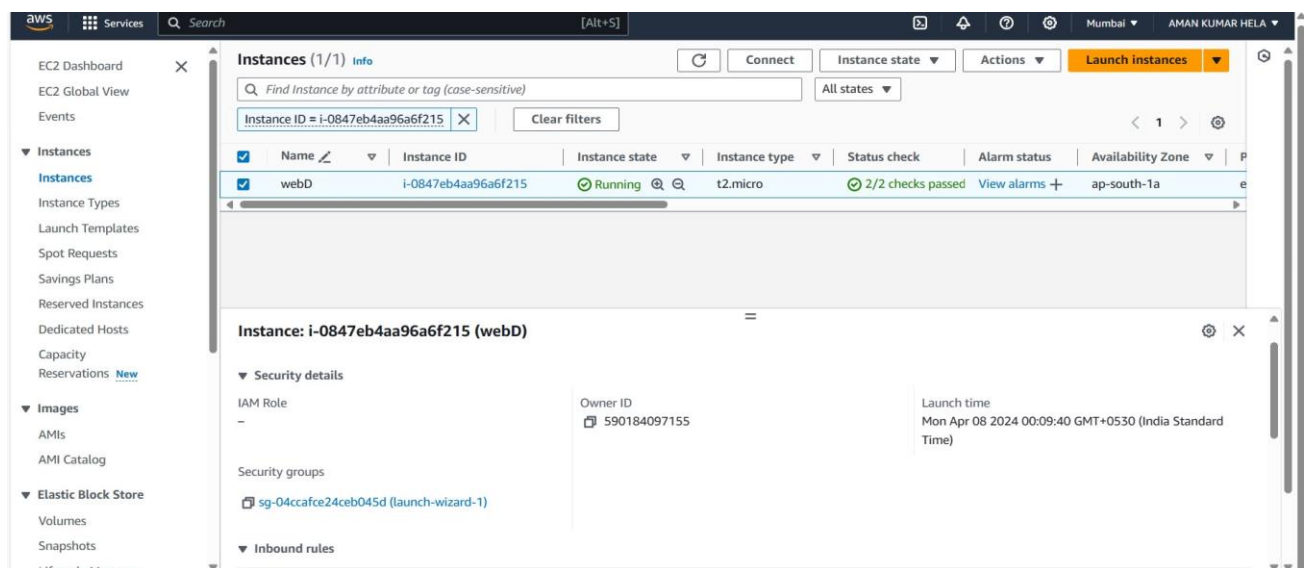
5. After that select Create security group and click all checkboxes bellow it. After it click on Launch Instance.

The screenshot shows the 'Firewall (security groups)' step of the AWS 'Launch Instance' wizard. It offers 'Create security group' (selected) or 'Select existing security group'. Below, it lists rules to be added: 'Allow SSH traffic from Anywhere', 'Allow HTTPS traffic from the internet', and 'Allow HTTP traffic from the internet'. A yellow warning box notes that rules with source 0.0.0.0/0 allow all IP addresses. The 'Configure storage' section shows '1x 8 GIB gp2' for the root volume. A 'Free tier' notice is at the bottom left. On the right, a 'Summary' panel shows instance details like 'Number of instances: 1', 'Software Image (AMI): Canonical, Ubuntu, 22.04 LTS', 'Virtual server type (instance type): t2.micro', and 'Firewall (security group): New security group'. At the bottom right are 'Cancel' and 'Launch instance' buttons, with a 'Review commands' link below.

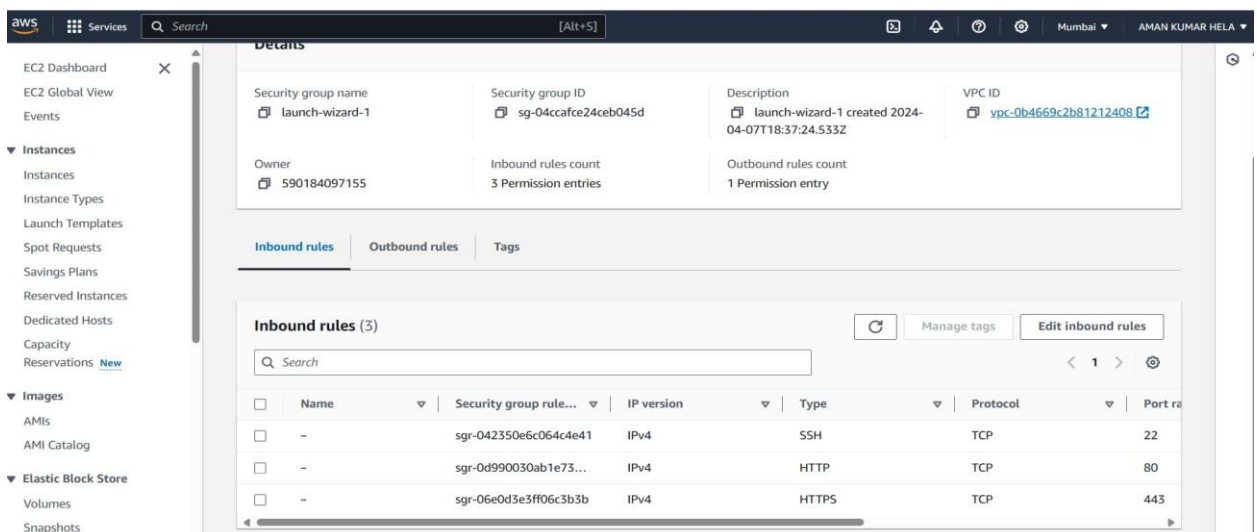
6. Now go to GitHub that project and then to index.js ,there you can see 4000 in app.listen which is port no that will be added to custom TCP.



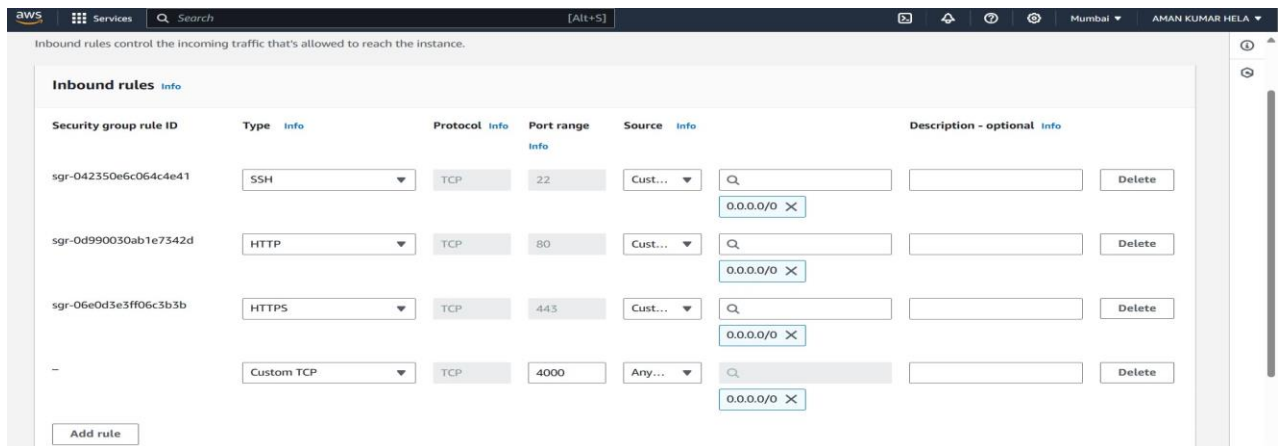
7. After it go back to EC2 instances then select that instance newly made and click on Security tab and then Security groups.



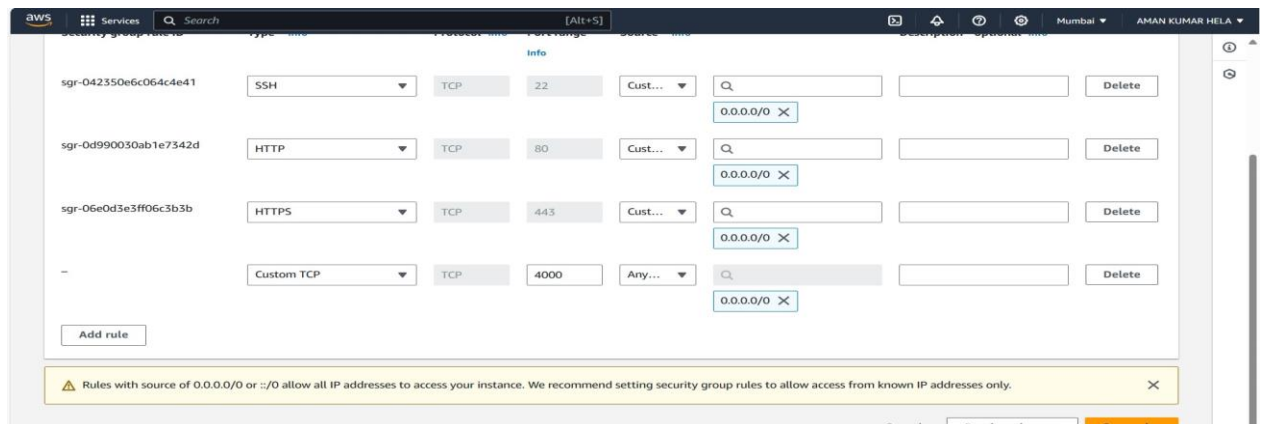
8. After in Inbound rules section click on Edit inbound rules.



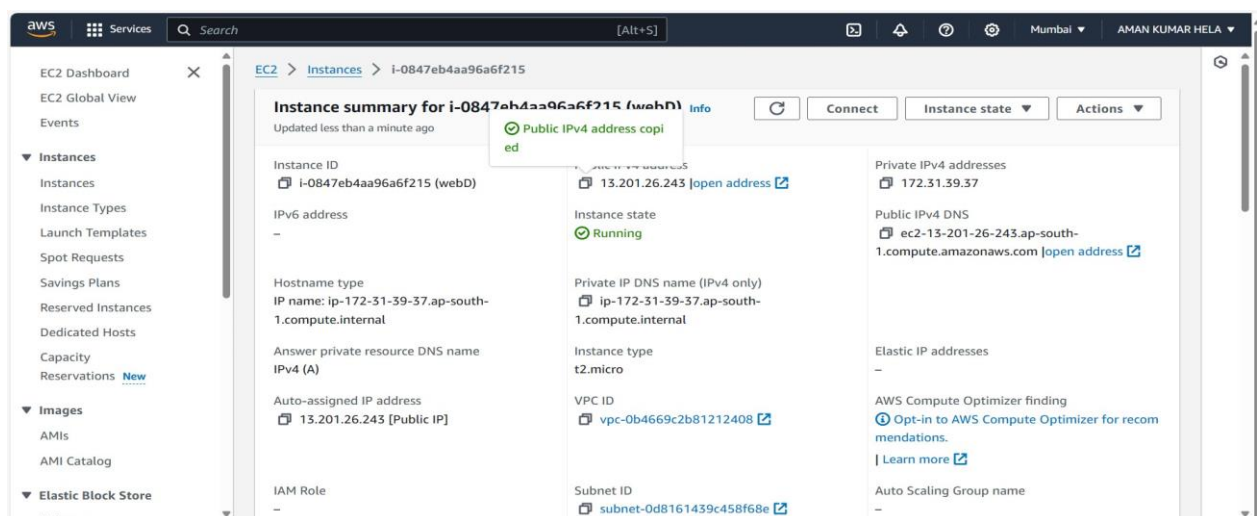
9. Then in inbound rules, click on Add rule select Custom TCP ,give port range 4000 and select 0.0.0.0/0.



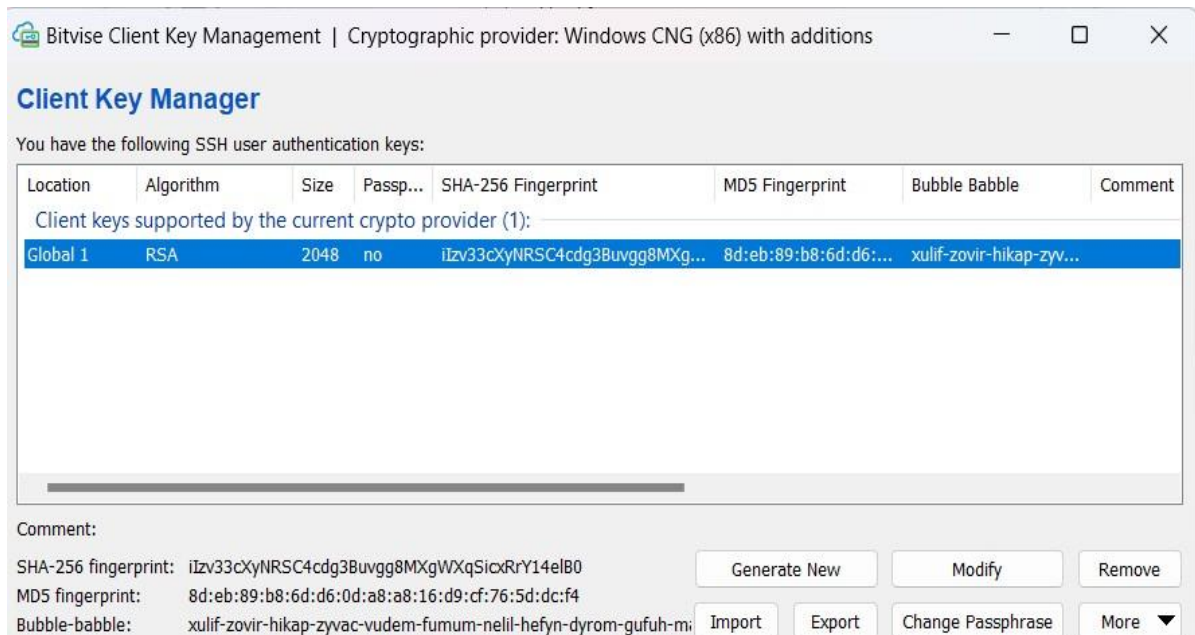
10. After it click on Save rules.



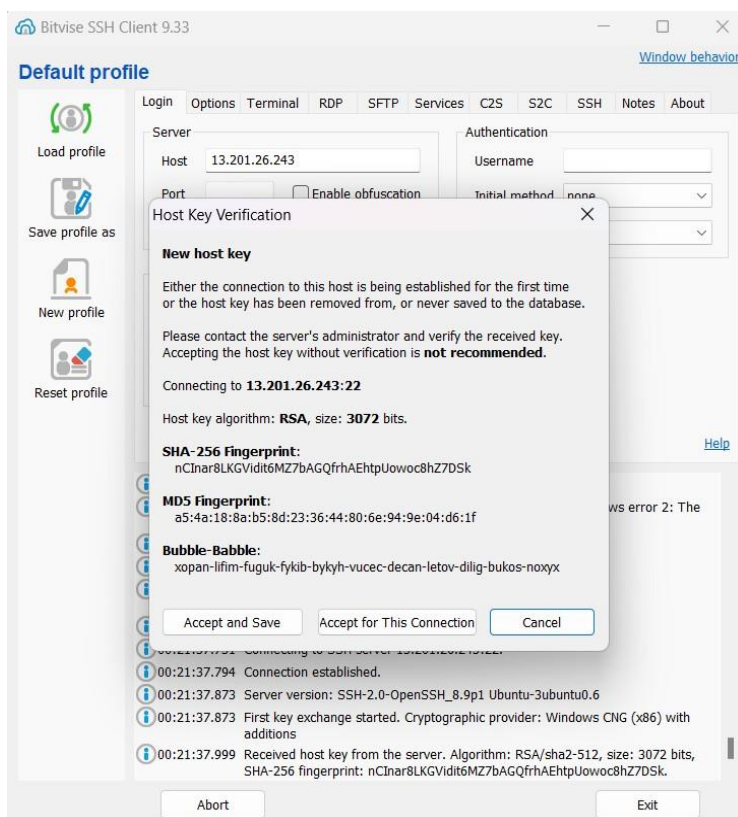
11. Now go back to EC2 recently created instance and copy the public IPv4 address.



12. Now open Bitwise SSH, go to Client key manager and import that downloaded key.



13. After it paste that copied IPv4 public address in host and then login and also do Accept and Save, also set publickey as global 2 .



14. Now open new terminal and write pwd, we are in ubuntu.


```
ubuntu@13.201.26.243:22 - Bitvise xterm - ubuntu@ip-172-31-39-37: ~
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Sun Apr 7 18:53:08 UTC 2024

System load: 0.080078125 Processes: 96
Usage of /: 20.4% of 7.57GB Users logged in: 0
Memory usage: 21% IPv4 address for eth0: 172.31.39.37
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

Updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-39-37:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-39-37:~$
```

15. Now write command “sudo apt-get update” to fetch all packages.

```
ubuntu@ip-172-31-39-37:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-39-37:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1519 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [293 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1643 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [275 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1060 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [241 kB]
```

16. Write “sudo apt-get upgrade” to upgrade all outdated packages.

```
ubuntu@ip-172-31-39-37:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  linux-aws linux-headers-aws linux-image-aws ubuntu-advantage-tools ubuntu-pro-client-110n
The following packages will be upgraded:
  apt apt-utils bash bsdxtrautils bsduutils coreutils curl dpkg eject ethtool fdisk libapt-pkg6.0
  libblkid1 libcurl3-gnutls libcurl4 libexpat1 libfdisk1 libgpgme11 libldap-2.5-0 libldap-common
  libmount1 libsmartcols1 libuuid1 mount python3-cryptography python3-update-manager snapd
  update-manager-core update-notifier-common util-linux uuid-runtime vim vim-common vim-runtime
  vim-tiny xxd
36 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
Need to get 43.9 MB of archives.
```

17. Now write command “sudo apt install nginx” to install webserver.

```
The following packages have been kept back:
  linux-aws linux-headers-aws linux-image-aws ubuntu-advantage-tools ubuntu-pro-client-110n
The following packages will be upgraded:
  apt apt-utils bash bsdxtrautils bsduutils coreutils curl dpkg eject ethtool fdisk libapt-pkg6.0
  libblkid1 libcurl3-gnutls libcurl4 libexpat1 libfdisk1 libgpgme11 libldap-2.5-0 libldap-common
  libmount1 libsmartcols1 libuuid1 mount python3-cryptography python3-update-manager snapd
  update-manager-core update-notifier-common util-linux uuid-runtime vim vim-common vim-runtime
  vim-tiny xxd
36 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
Need to get 43.9 MB of archives.
After this operation, 1584 kB disk space will be freed.
Do you want to continue? [Y/n] y
```

18. Now to execute javascript, we need to install nodejs. Write command

“ curl -SL https://deb.nodesource.com/setup_16.x | sudo -E bash – ”.

```
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 110 kB in 1s (129 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
curl is already the newest version (7.81.0-1ubuntu1.16).
gnupg is already the newest version (2.2.27-3ubuntu2.1).
apt-transport-https is already the newest version (2.4.12).
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://deb.nodesource.com/node_16.x nodistro InRelease [12.1 kB]
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://deb.nodesource.com/node_16.x nodistro/main amd64 Packages [7462 B]
Fetched 19.6 kB in 2s (11.8 kB/s)
Reading package lists... Done
```

19. Now to install write “sudo apt-get install -y nodejs”.

```

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 5 not upgraded.
Need to get 27.5 MB of archives.
After this operation, 128 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_16.x nodistro/main amd64 nodejs amd64 16.20.2-1nodesource1 [27.5 MB]
Fetched 27.5 MB in 2s (17.0 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 65516 files and directories currently installed.)
Preparing to unpack .../nodejs_16.20.2-1nodesource1_amd64.deb ...
Unpacking nodejs (16.20.2-1nodesource1) ...
Setting up nodejs (16.20.2-1nodesource1) ...
Processing triggers for man-db (2.10.2-1) ...

```

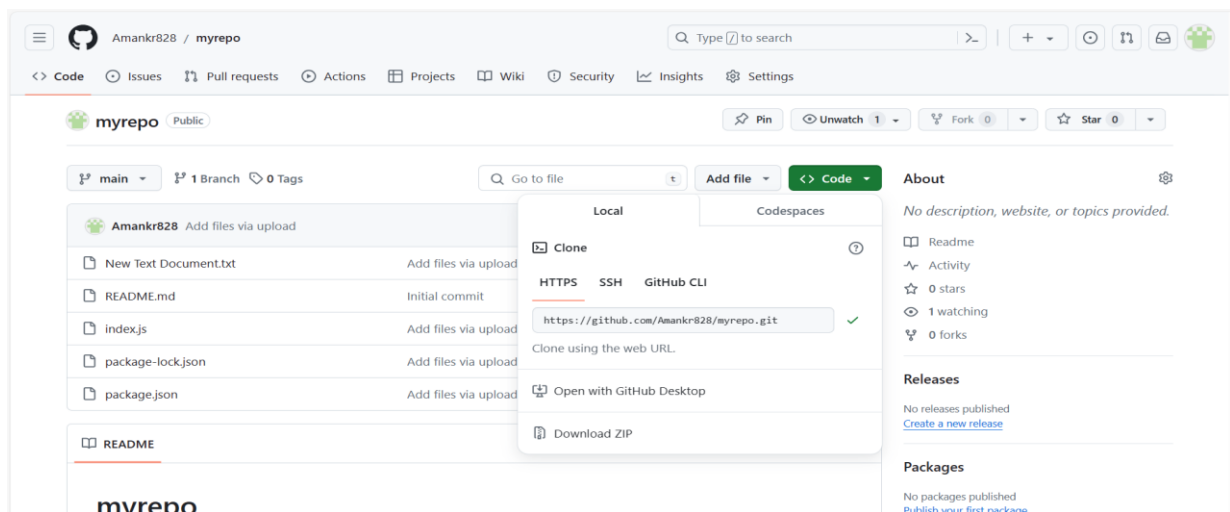
20. Write “node --version” to see what version of nodejs installed.

```

ubuntu@ip-172-31-39-37:~$ node --version
v12.22.9

```

21. Now go back to GitHub and in code copy HTTPS URL.



22. Now write git clone and paste that URL and write ls to see if project has been cloned or not.

```

ubuntu@ip-172-31-39-37:~$ git clone https://github.com/Amankr828/myrepo.git
Cloning into 'myrepo'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), 49.58 KiB | 7.08 MiB/s, done.
ubuntu@ip-172-31-39-37:~$ ls
myrepo
ubuntu@ip-172-31-39-37:~$ cd myrepo

```


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

```
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), 49.58 KiB | 7.08 MiB/s, done.
ubuntu@ip-172-31-39-37:~$ ls
myrepo
ubuntu@ip-172-31-39-37:~$ cd myrepo
ubuntu@ip-172-31-39-37:~/myrepo$ ls
'New Text Document.txt'  README.md  index.js  package-lock.json  package.json
```

24. Now to execute node command we have to install node packet manager(npm). So write “npm install”.

```
ubuntu@ip-172-31-39-37:~/myrepo$ npm install
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math
.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-
random for details.

added 258 packages, and audited 259 packages in 13s

18 packages are looking for funding
  run `npm fund` for details

12 vulnerabilities (10 moderate, 2 critical)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
ubuntu@ip-172-31-39-37:~/myrepo$
```

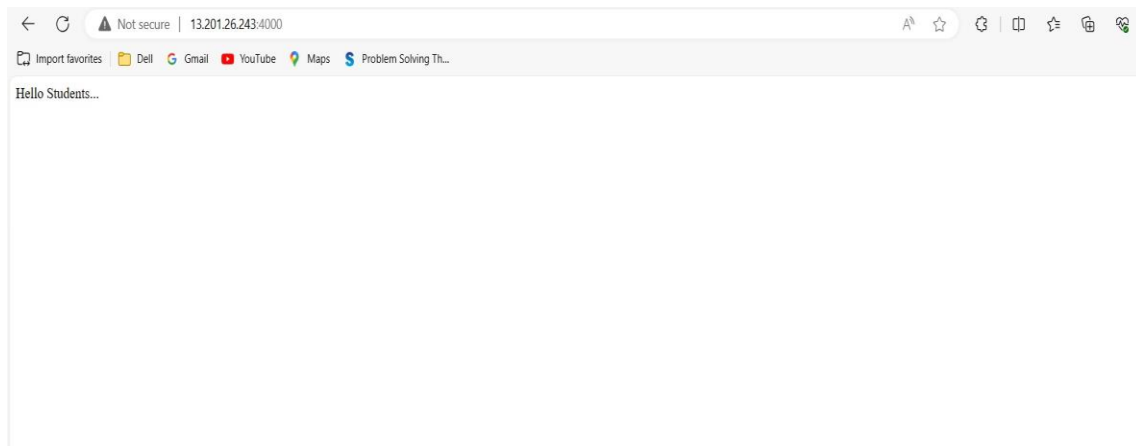
25. Now write node `index.js`.

```
ubuntu@ip-172-31-39-37:~/myrepo$ node index.js
Started server
```

26. After it again copy public IPv4 address in EC2 instance and then paste it in another tab URL section.



27. At last in URL at end write :4000 to get your website.



28. Now to close server in new Terminal do (ctrl + c) to stop server and at last logout.

```
Run `npm audit` for details.  
ubuntu@ip-172-31-39-37:~/myrepo$ node index.js  
Started server  
^C  
ubuntu@ip-172-31-39-37:~/myrepo$
```

■ In this way we have deployed a project from GitHub to EC2.

