

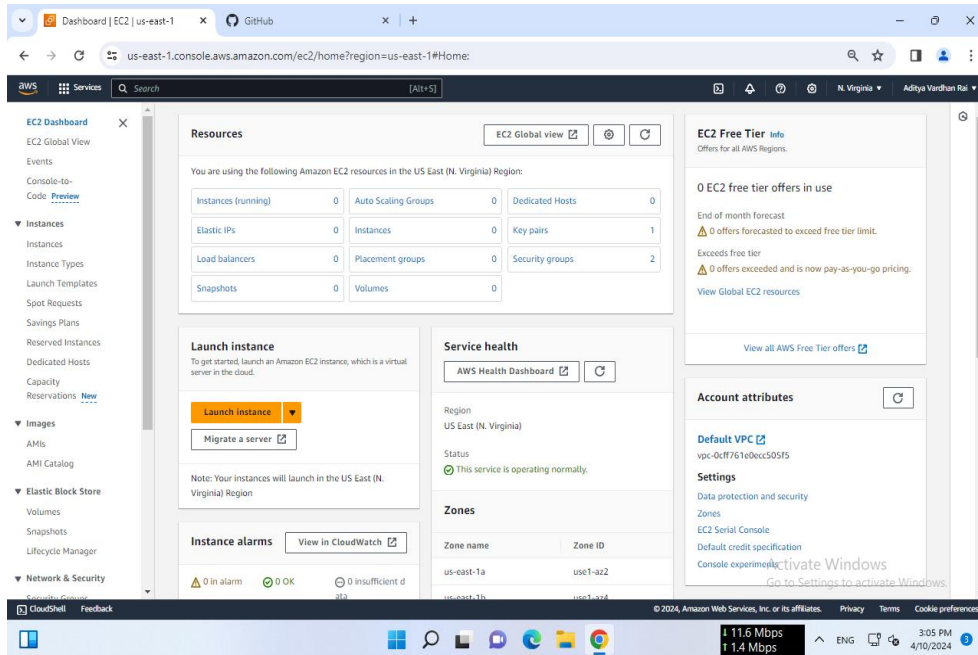
Assignment-9

PROBLEM STATEMENT :

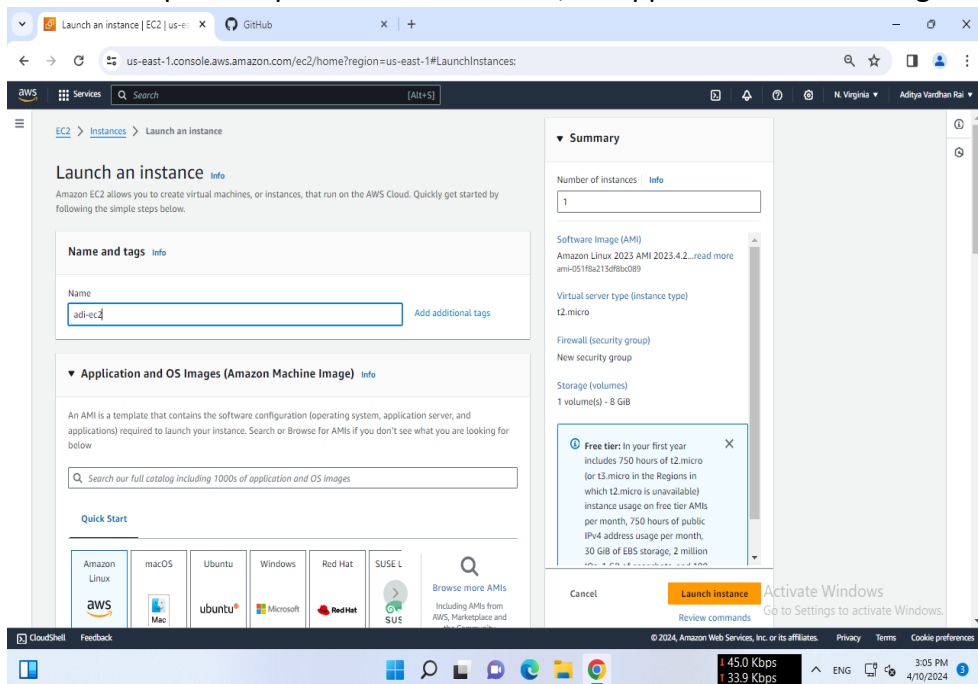
Deploy a project from GitHub to EC2.

Steps to deploy project :

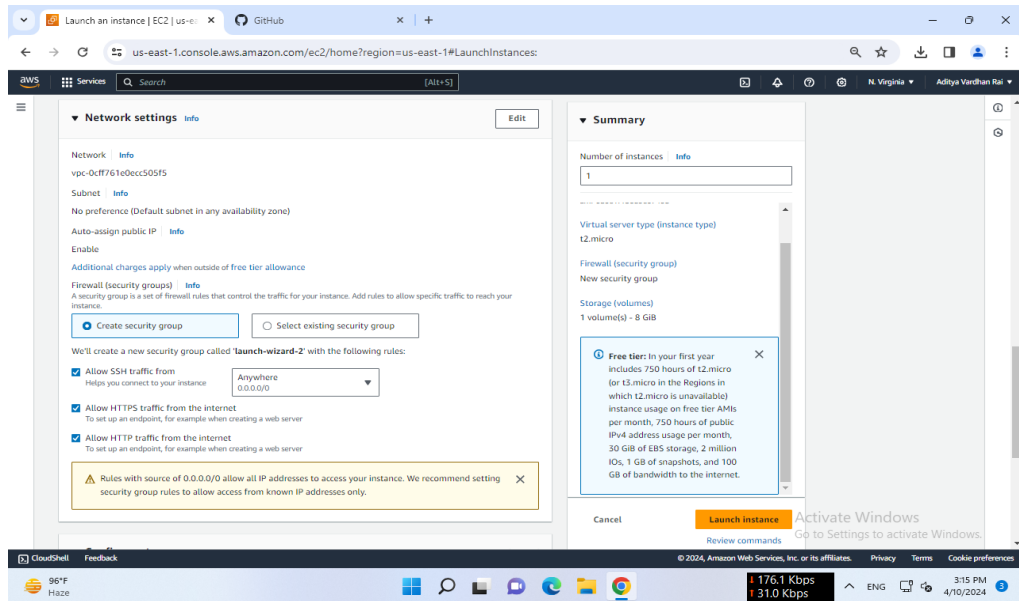
1. Sign up for an AWS account, search for 'EC2', select it and click on launch instance.



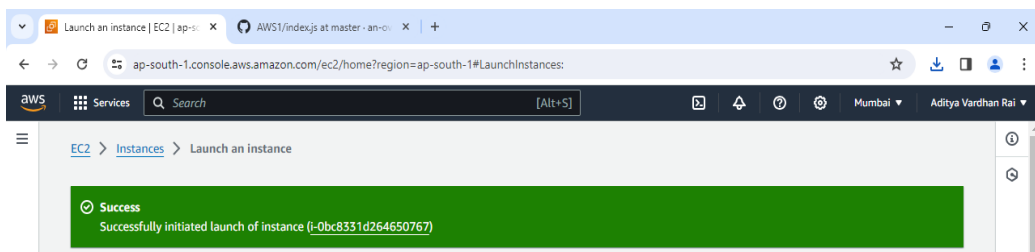
2. Fill up the required details->'Name', in 'Application and OS Images' select 'ubuntu'.



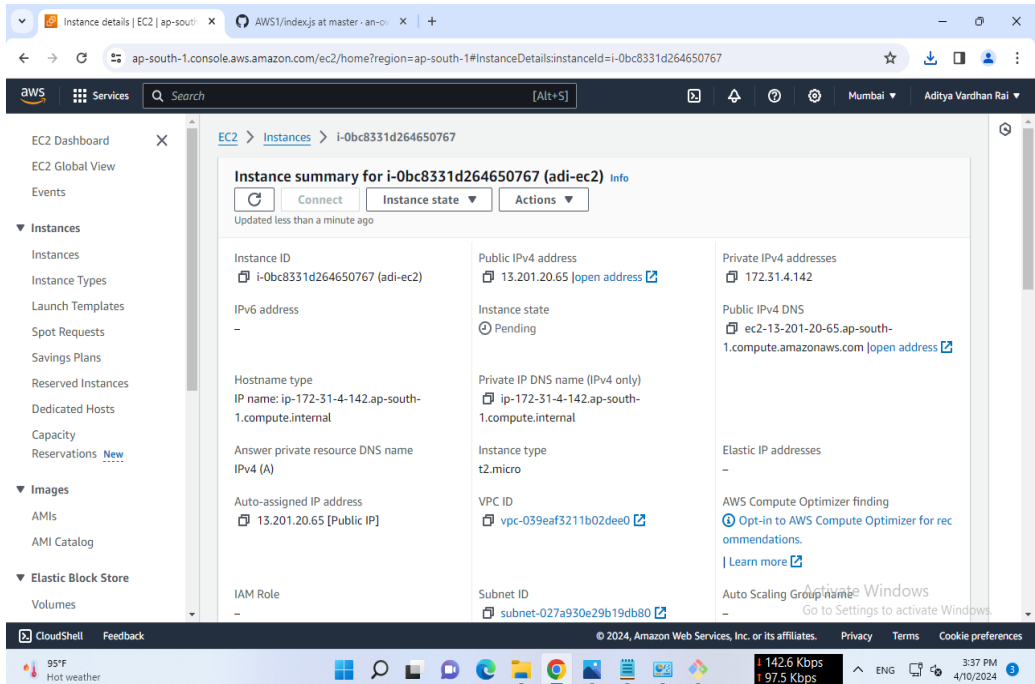
Under 'Network settings', tick off all the three checkboxes then click on 'Launch instance'.



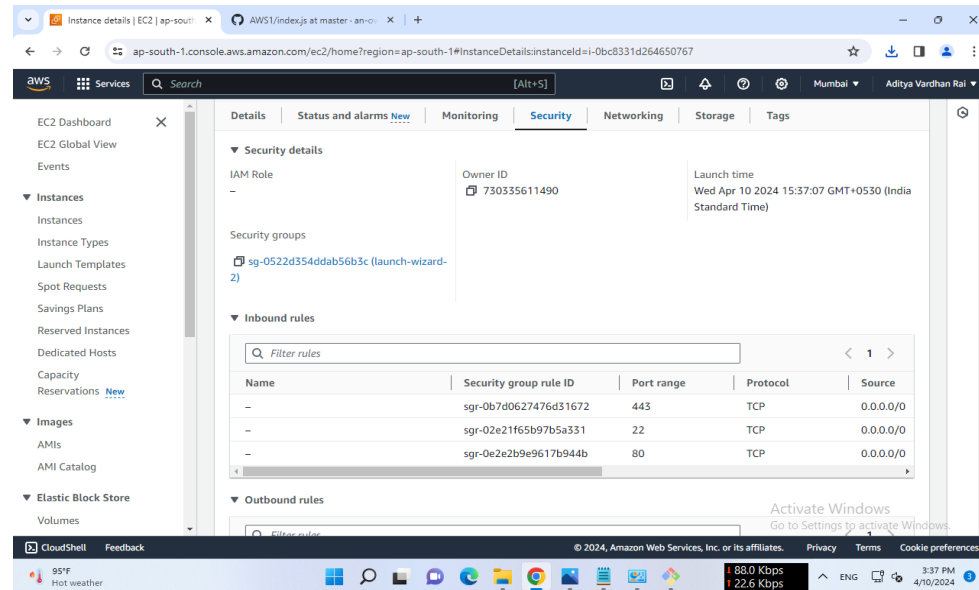
3. "Dynamicip_EC2" instance is successfully created and then click on 'Instances'.



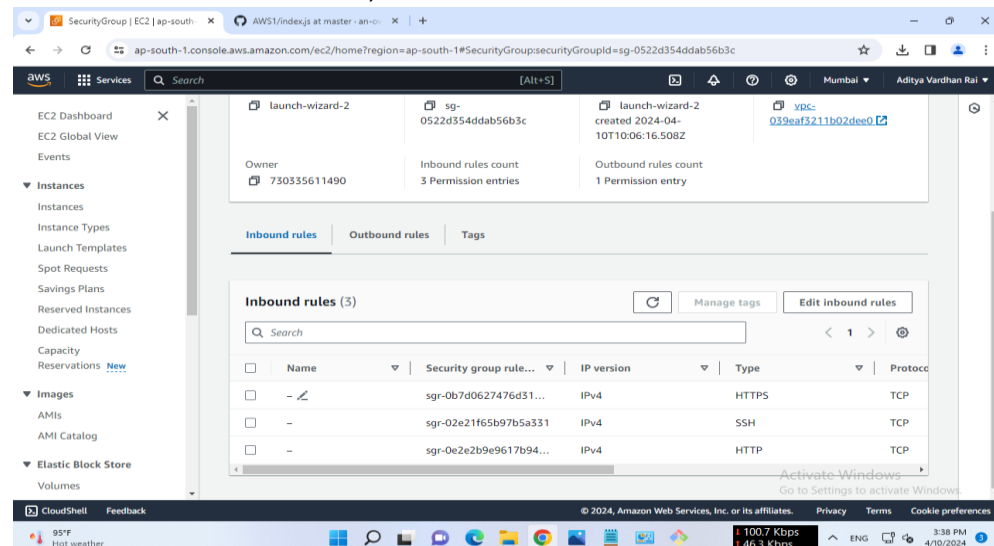
4. Now check whether "Dynamicip_EC2" is running or not then click on "Instance ID".



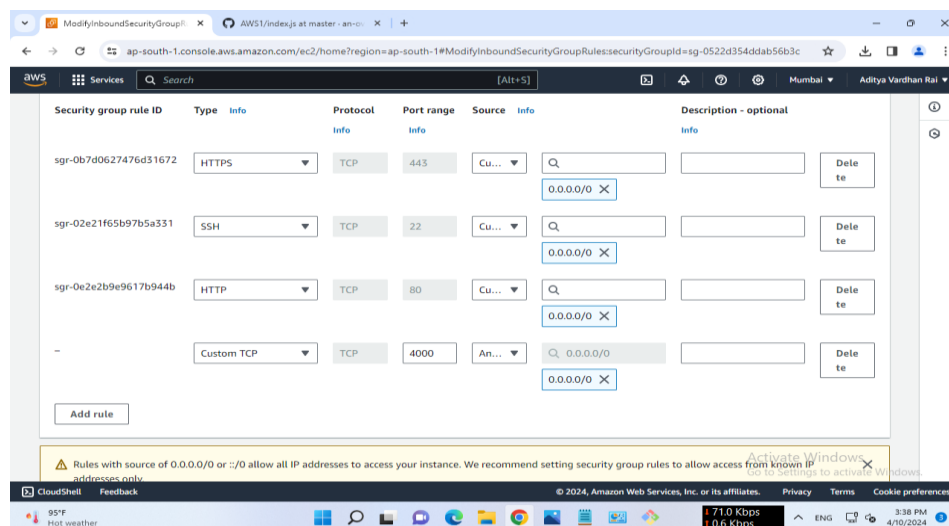
5. Then go to “Security” and click on link under “Security groups” to get the edit option.



6. Then in “Inbound rules”, click on “Edit inbound rules”.



7. Click on “Add rule”.

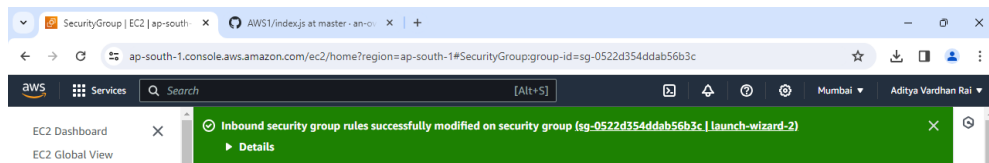


8. Set the “Port range” to 4000 , in “Source” set “0.0.0.0/0” & click on “Save rules”.

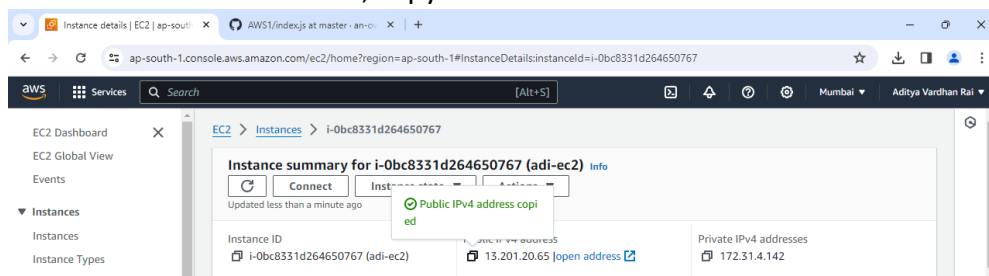
The screenshot shows the 'Inbound rules' configuration page for a security group. It lists three existing rules: sgr-035cce9e964384fbd (HTTPS, port 443), sgr-047e17b302581bb74 (HTTP, port 80), and sgr-009b8cce3e2b87fa (SSH, port 22). A new rule is being added at the bottom with the following settings: Type: Custom TCP, Protocol: TCP, Port range: 4000, Source: Anywhere... (0.0.0.0/0). A warning message at the bottom states: 'Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-035cce9e964384fbd	HTTPS	TCP	443	Custom	Q 0.0.0.0/0 X
sgr-047e17b302581bb74	HTTP	TCP	80	Custom	Q 0.0.0.0/0 X
sgr-009b8cce3e2b87fa	SSH	TCP	22	Custom	Q 0.0.0.0/0 X
-	Custom TCP	TCP	4000	Anywhere...	Q 0.0.0.0/0 X

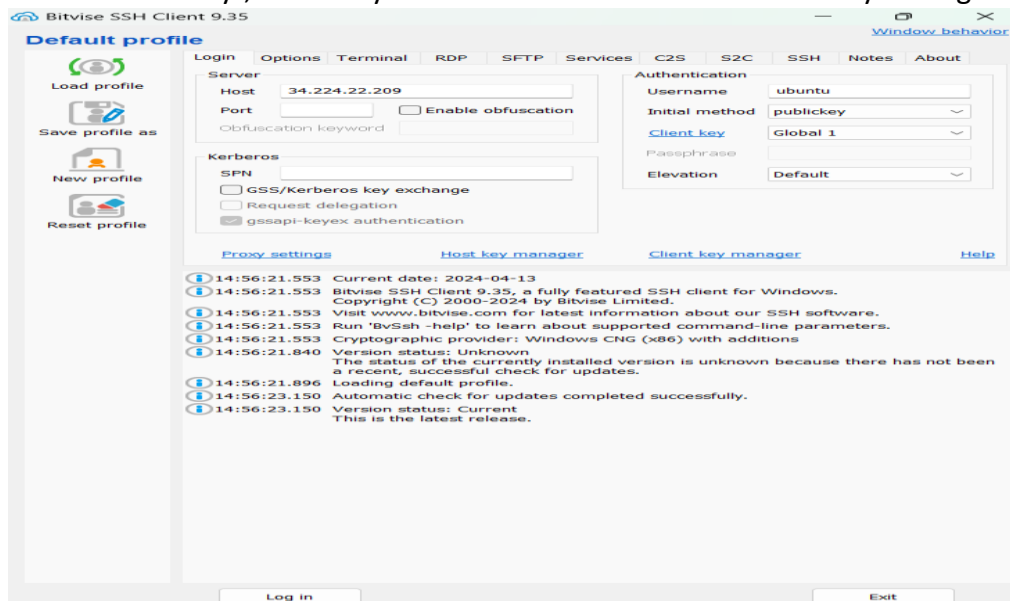
9. Rule has been successfully saved.



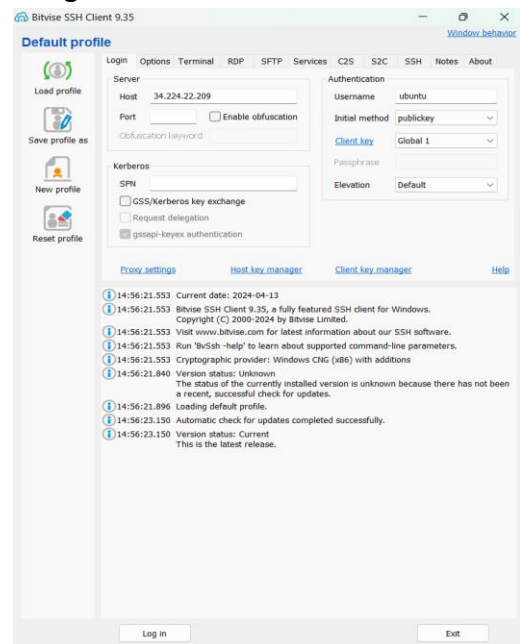
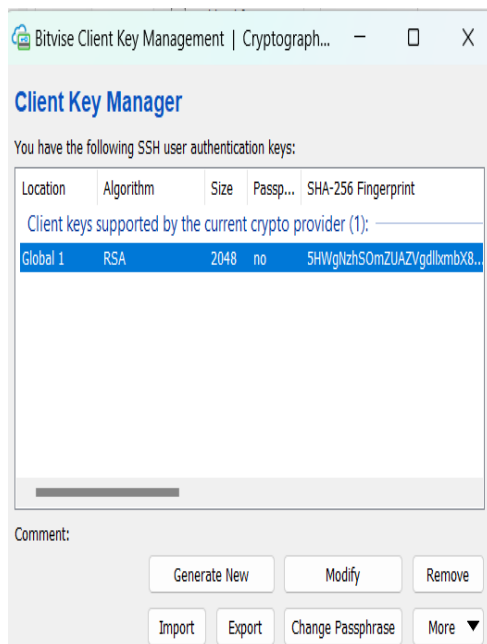
10. Go back to the Instance, copy the “Public IPv4 address”.



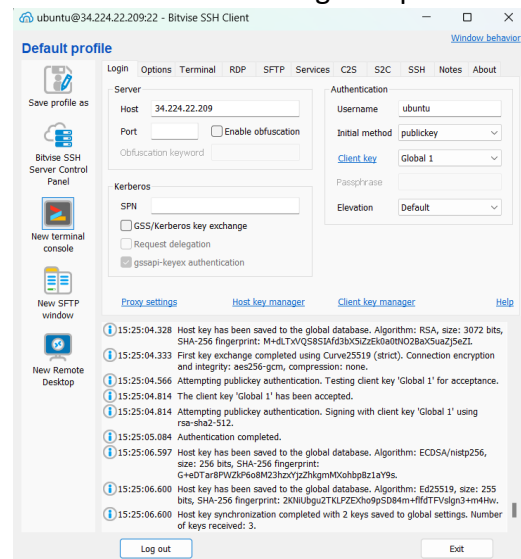
11. In “Bitwise SSH Client”, paste the “Public IPv4 address” in “Host” and under “Authentication tab” give the username as Ubuntu, Initial method as “Public key”, Client key as “Global1”. Then click on “Client key manager”.



12. Remove any previously selected key if any, then click on “Import” & select the key with which instance was created. Then click on Log in.



13. After successful “Log in” open a “New Terminal Console”.



14. In the console, type the following commands in sequential order.

```
ubuntu@34.224.22.209:22 - Bitvise xterm - ubuntu@ip-172-31-27-221: ~
```

```
ubuntu@ip-172-31-27-221:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-27-221:~$ sudo apt-get update
Fetched 30.7 MB in 6s (5382 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-27-221:~$ sudo apt-get upgrade
```

```
No containers need to be restarted.
```

```
No user sessions are running outdated binaries.
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-27-221:~$ sudo apt-get install nginx
```

```

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-27-221:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
ubuntu@ip-172-31-27-221:~$ curl -SL https://deb.nodesource.com/setup_16.x | sudo -E bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
 100 19.6 kB    0 19.6 kB    0  24.5 kB/s    0:00  0:00  0:00  24.5 kB/s
Reading package lists... Done
2024-04-13 10:28:31 - Repository configured successfully. To install Node.js, run: apt-get install nodejs
ubuntu@ip-172-31-27-221:~$ sudo apt install nodejs

No containers need to be restarted.

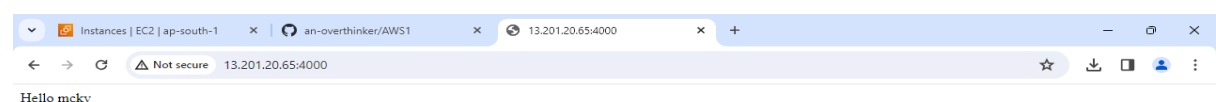
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-27-221:~$ nodejs -v
v16.20.2
ubuntu@ip-172-31-27-221:~$ git clone https://github.com/Rupsa1037/Rup_AWS1.git
Cloning into 'Rup_AWS1'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), 48.21 KiB | 8.03 MiB/s, done.
ubuntu@ip-172-31-27-221:~$ ls
Rup_AWS1
ubuntu@ip-172-31-27-221:~$ cd Rup_AWS1/
ubuntu@ip-172-31-27-221:~/Rup_AWS1$ ls
'New Text Document.txt'  index.js  package-lock.json  package.json
ubuntu@ip-172-31-27-221:~/Rup_AWS1$ npm install

Run `npm audit` for details.
npm notice
npm notice New major version of npm available! 8.19.4 -> 10.5.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.5.2
npm notice Run npm install -g npm@10.5.2 to update!
npm notice
ubuntu@ip-172-31-27-221:~/Rup_AWS1$ npm -v
8.19.4
ubuntu@ip-172-31-27-221:~/Rup_AWS1$ node index.js
Started server

```

15. Now copy the “Public IPv4 address” & paste it on a new tab. Add “:4000” at the end and then press enter.



Therefore, the project has been successfully deployed from GitHub to EC2.