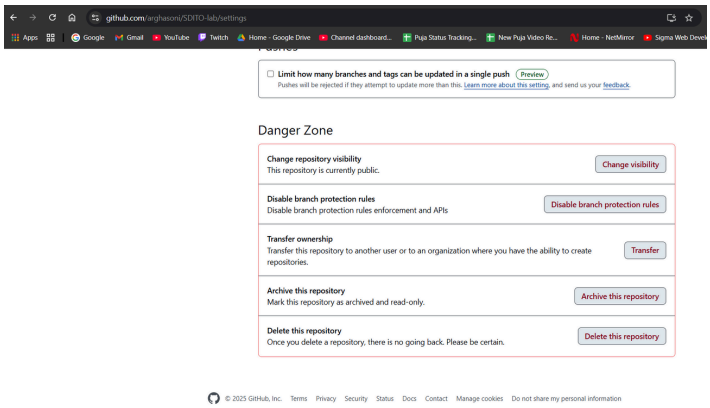
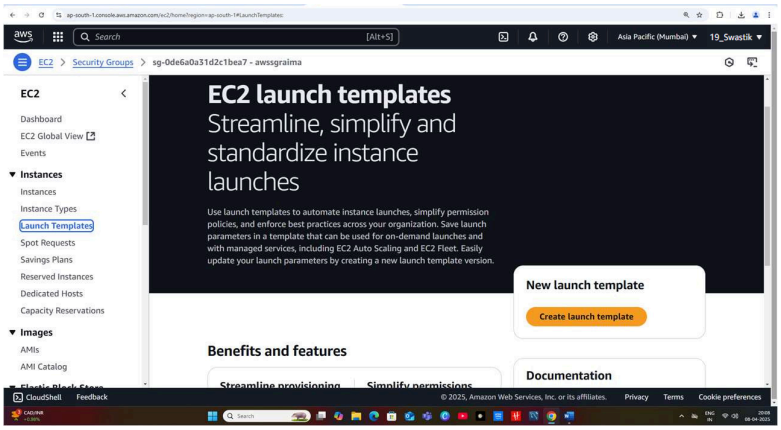
ASSIGNMENT NO :- 11

TITLE :- Build scaling plans in AWS that balance the load on different EC2 instances.

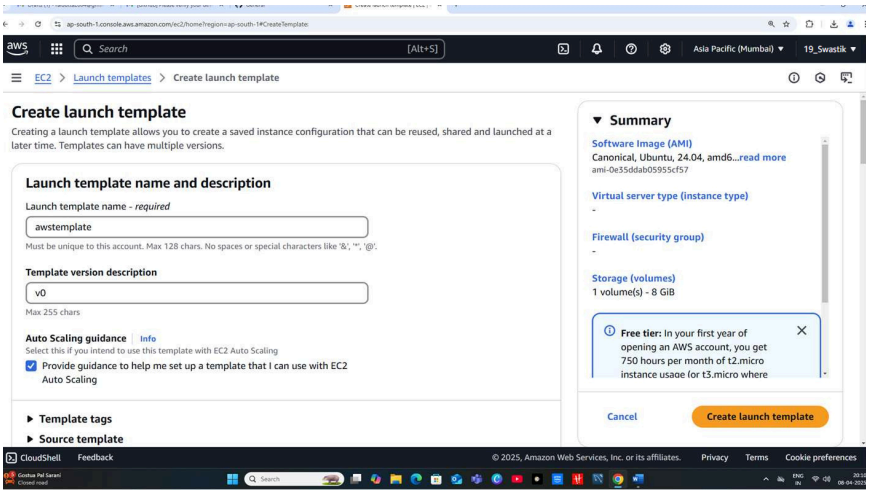
STEP 1 :- Check if your GitHub Repository is public or not .



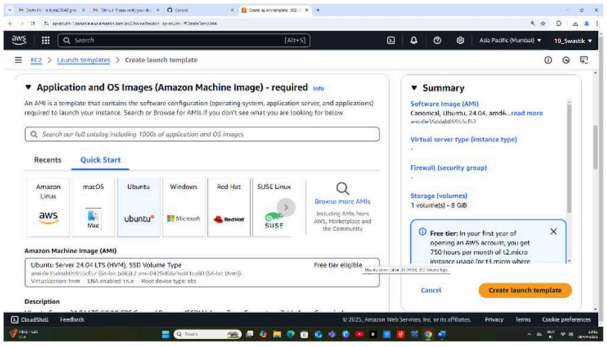
STEP 2 :- Login to your AWS Account and then go to Launch Template and click on Create Launch Template



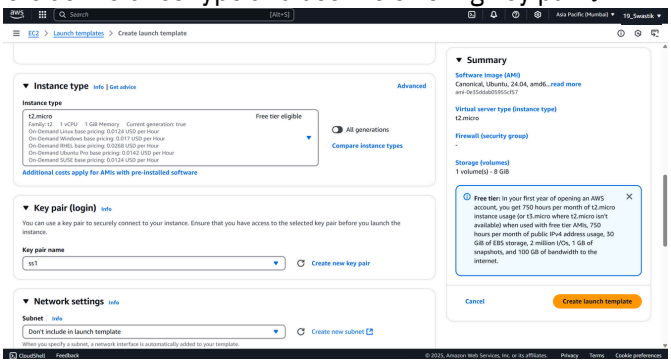
STEP 3:- Give your Template name and version description and select Auto Scaling guidance.



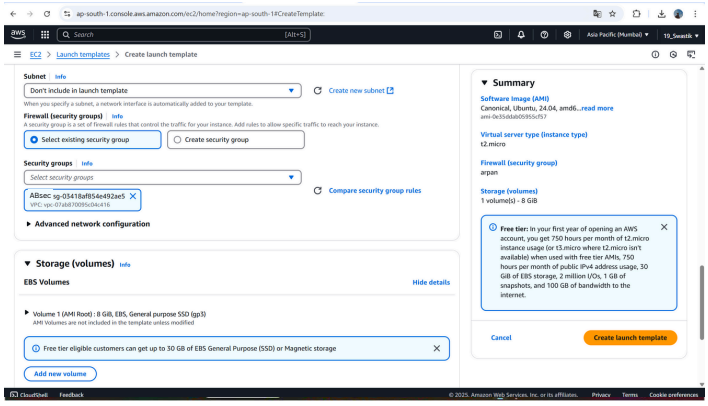
>> Select Ubuntu in Quickstart



>> Select t2.micro as Instance type and use the existing key pair you created

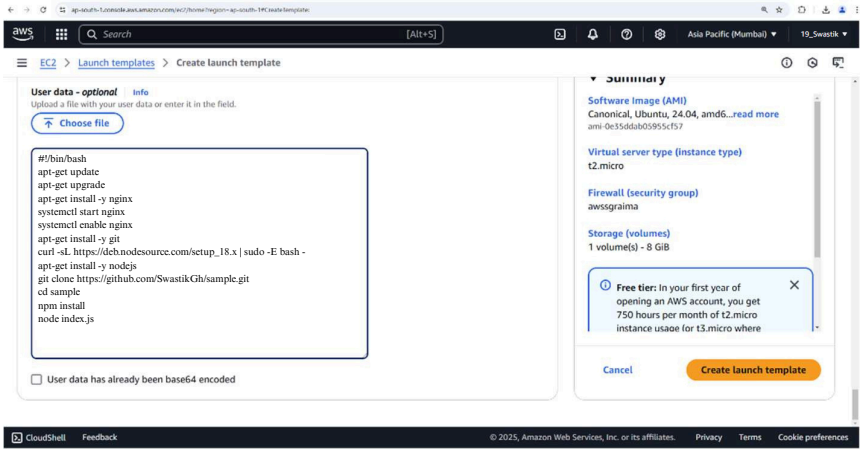


>>

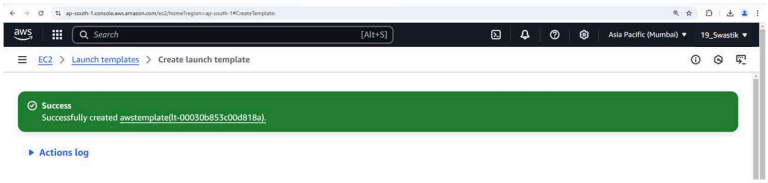


>> then click on Advanced Details

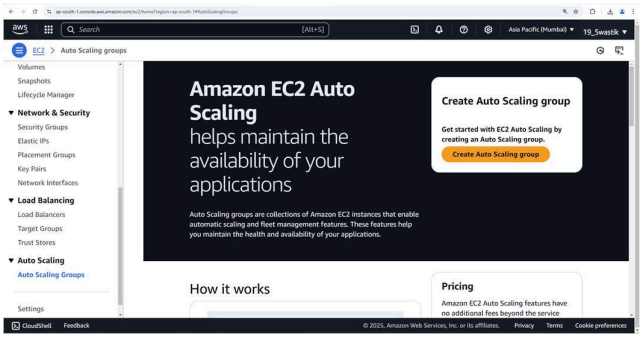
>> write the command line as written below and add your Repository link in Git Clone <Repository Link>



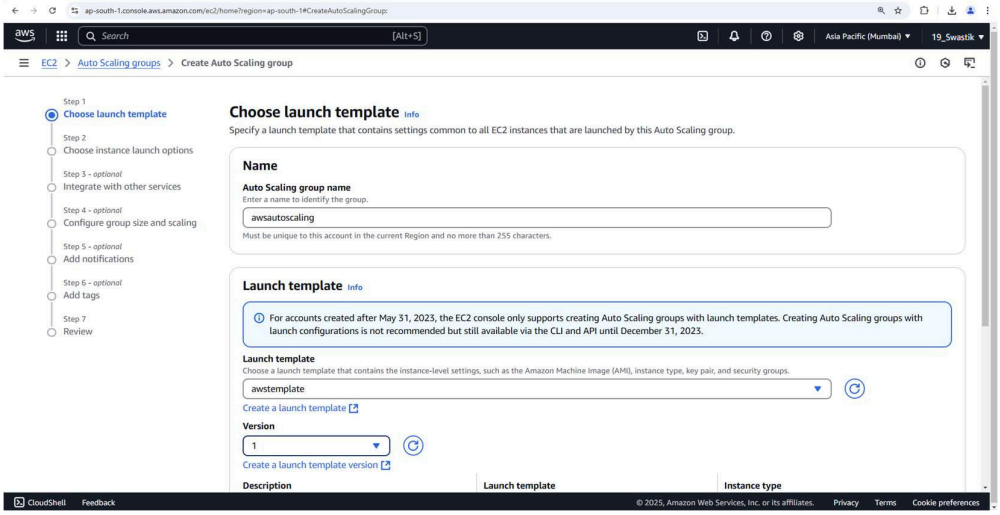
>> Your Template is created successfully.



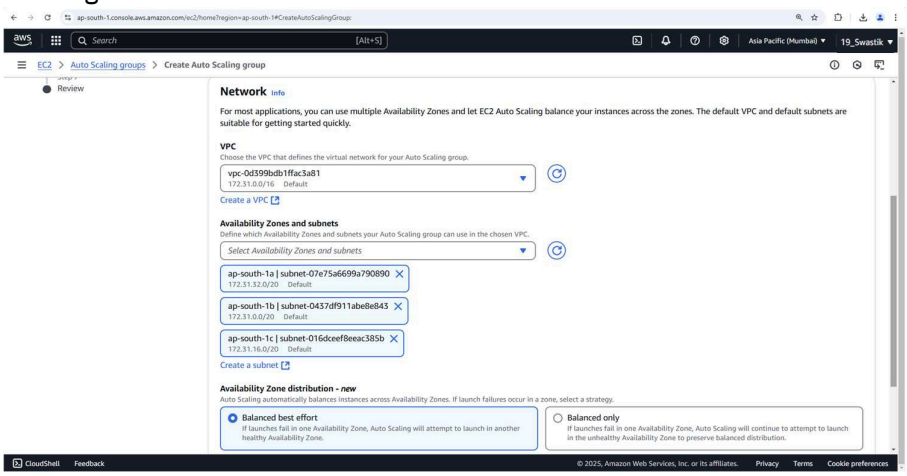
STEP 5:- Click on Auto Scaling Groups and then click on Create Auto Scaling groups.



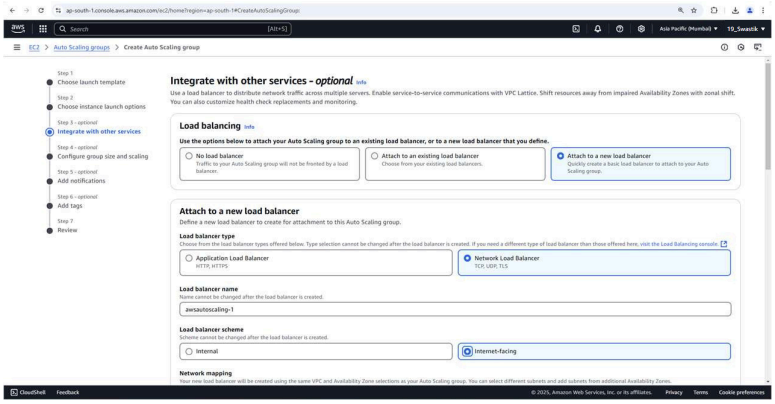
>> Give a name and select your created Template and Give the version as Latest (1) and click on Next



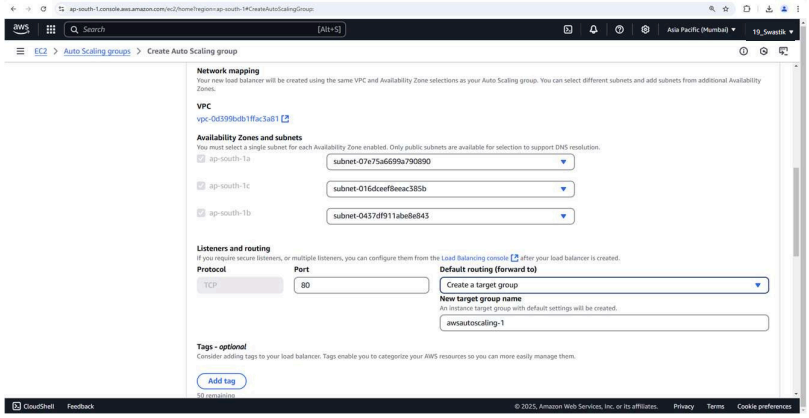
>>In Network region Select the Three Available Zones as servers and click on next



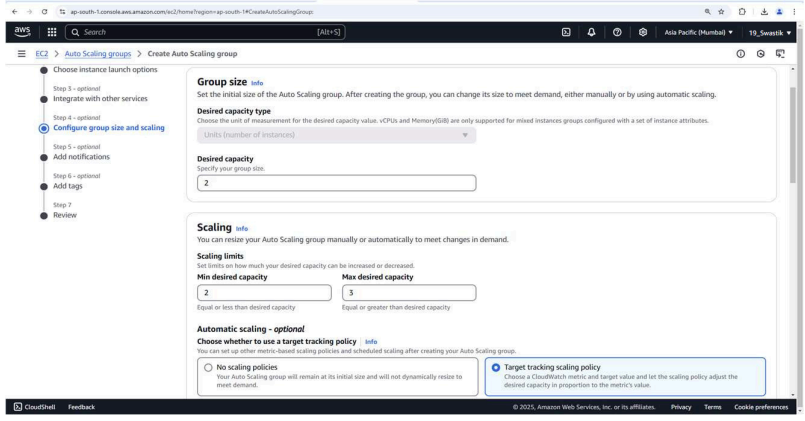
>> Select Attach a new load balancer and then select Network load balancer and then select Internet Facing



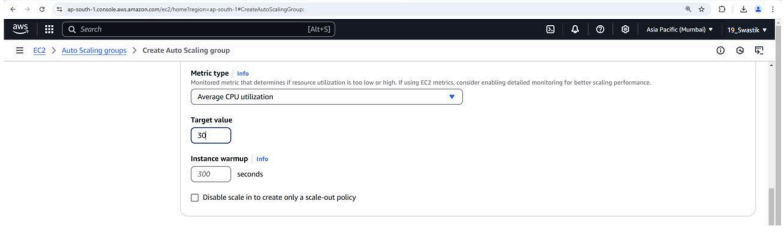
>> click on Create Target Group and select a target group name and then click on next



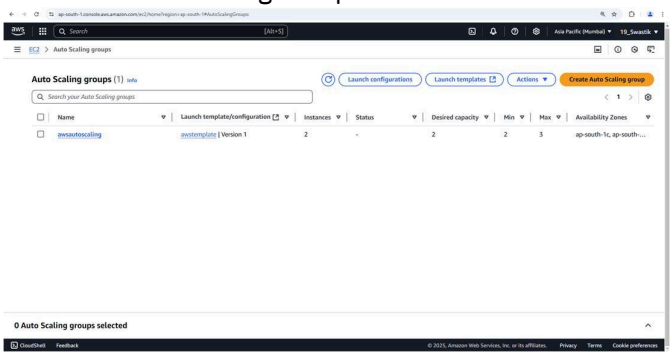
>> Now select your Desired capacity , Min desired and max desired capacity and then select Target Tracking Scaling capacity.



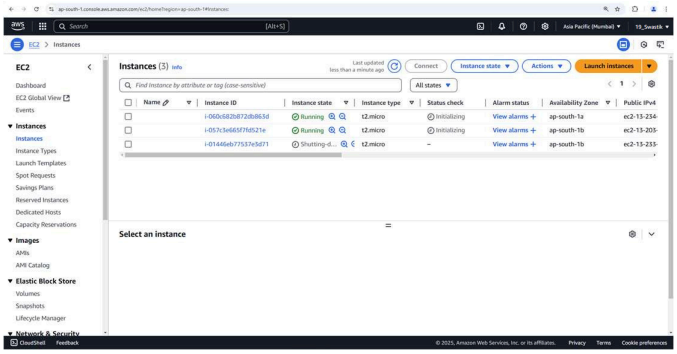
>> Give the Target Value and give Instance Runup and then click on Next (3times)



>>Review and click on Create Auto Scaling Group



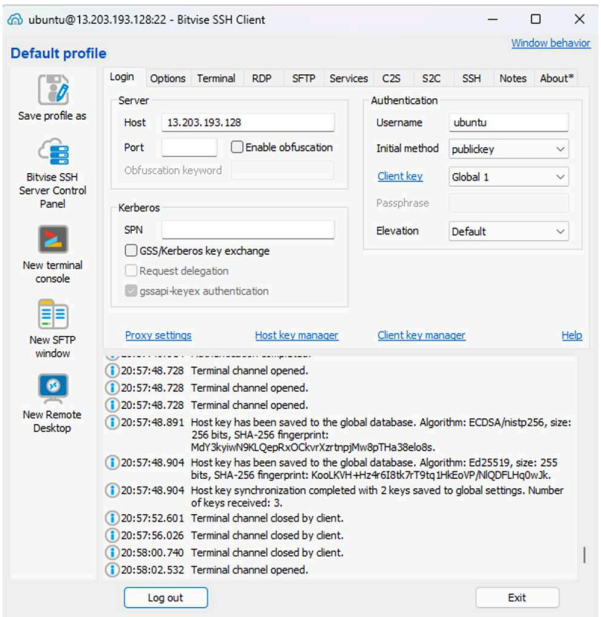
STEP 5:- As you can see the Instances are automatically created if you will delete one instance then the 3rd one will automatically generated as 3 servers are given



STEP 6:- Select any one Instance and Copy its IPV4 address and paste it in Incognito website with port number(IP Address : 4000)

Hellooo!! This is Argha from MCKVIE

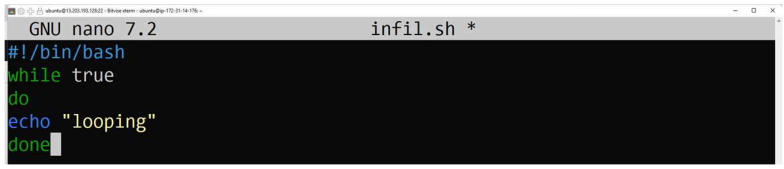
STEP 7:- Paste the address in Bitvise Server and give the required field of username , Initial method and Client key then click on LOG IN and go to the New Terminal



>> Then write the following command in terminal to open a shell file

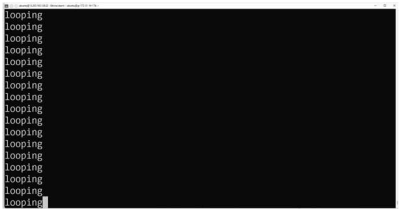
```
ubuntu@ip-172-31-14-176:~$ nano infil.sh
```

>> Then write the following code in the shell file

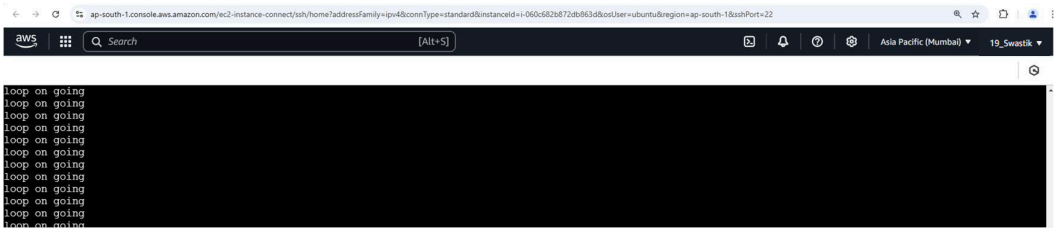


>> Compile and run the File Using the following command

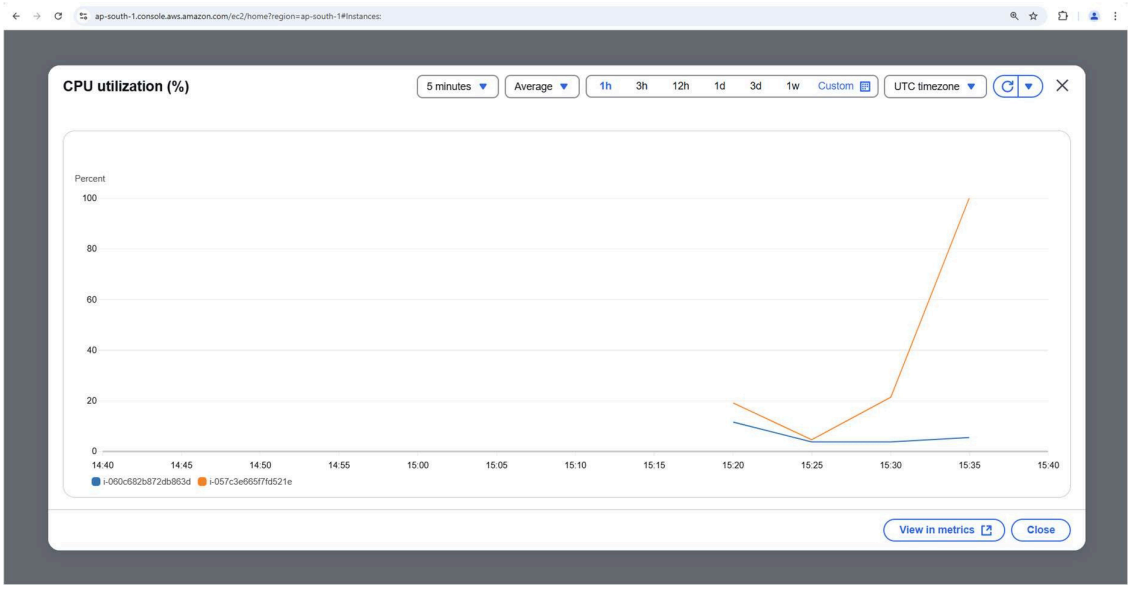
```
ubuntu@ip-172-31-14-176:~$ chmod +x infil.sh
ubuntu@ip-172-31-14-176:~$ ./infil.sh
```



STEP 6:- Select other Instance and click on Connect and Then write the same command and same code in terminal and then run it.



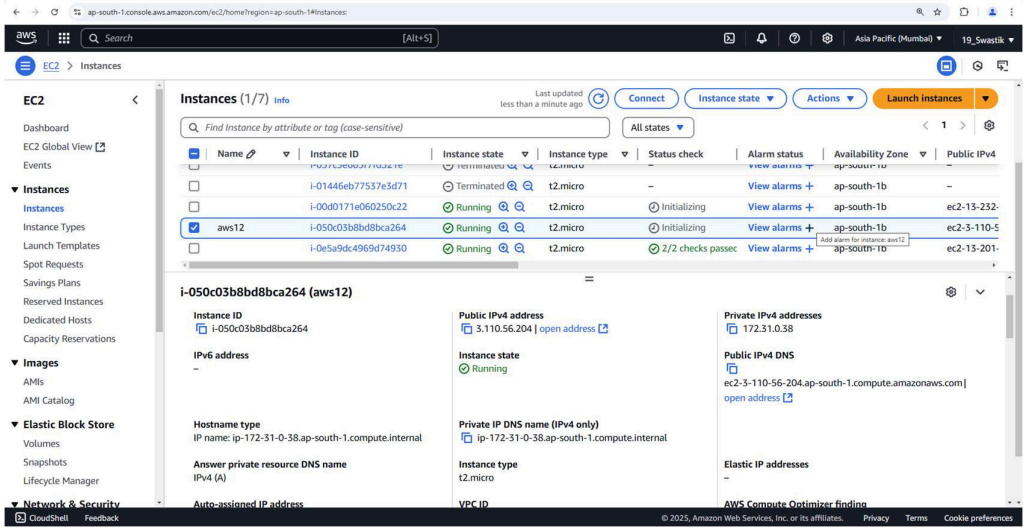
>> Then click on both the instances and you can see the graph of both



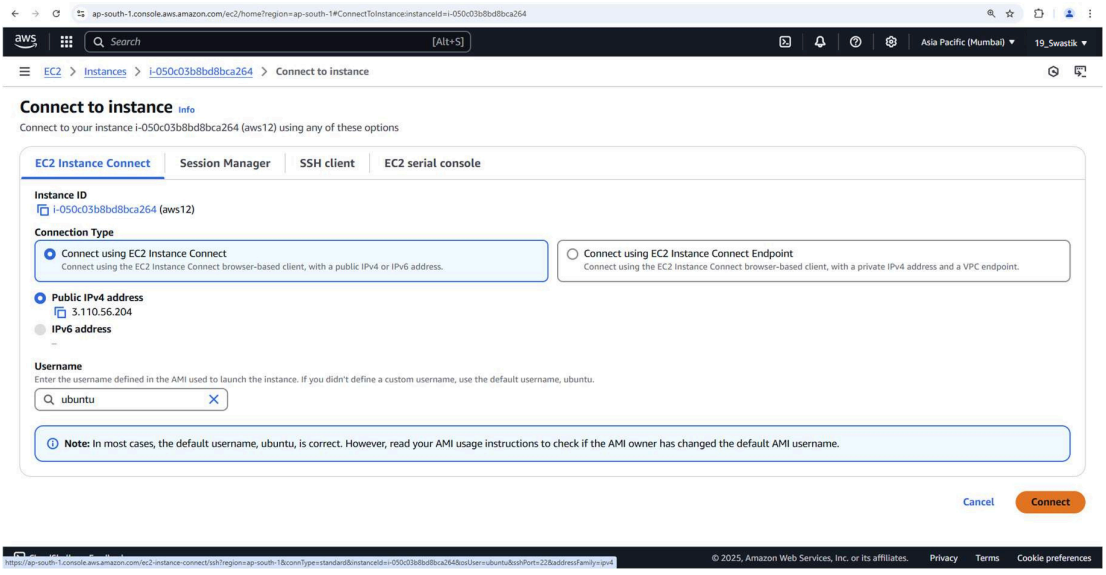
ASSIGNMENT NO :- 12

TITLE :- Deploy and run the project in AWS without using the port.

STEP 1 :- Create a new EC2 Instance



STEP 2 :- Connect the instance with the server.



STEP 4:- Wite the following command in terminal.

```
ubuntu@ip-172-31-0-38:~$ cd ..
ubuntu@ip-172-31-0-38:/home$ cd ..
ubuntu@ip-172-31-0-38:/$ cd etc
ubuntu@ip-172-31-0-38:/etc$ cd nginx
ubuntu@ip-172-31-0-38:/etc/nginx$ cd sites-available
ubuntu@ip-172-31-0-38:/etc/nginx/sites-available$ ls
default
ubuntu@ip-172-31-0-38:/etc/nginx/sites-available$ sudo nano default
ubuntu@ip-172-31-0-38:/etc/nginx/sites-available$ sudo chmod 777 default
ubuntu@ip-172-31-0-38:/etc/nginx/sites-available$ sudo systemctl restart nginx
ubuntu@ip-172-31-0-38:/etc/nginx/sites-available$
```

>> Now Open the default file and write the following code.

```
#       location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
#               try_files $uri $uri/ =404;
#       }
        location / {
                proxy_pass http://localhost:4000;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection 'Upgrade';
                proxy_set_header Host $host;
                proxy_cache_bypass $http_upgrade;
        }
        # pass PHP scripts to FastCGI server
        #
        #location ~ \.php$ {
```

STEP 5:- Now copy the IP address of the instance and check the server without PORT number.

Swastik