

## Assignment-11:

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

Steps:-

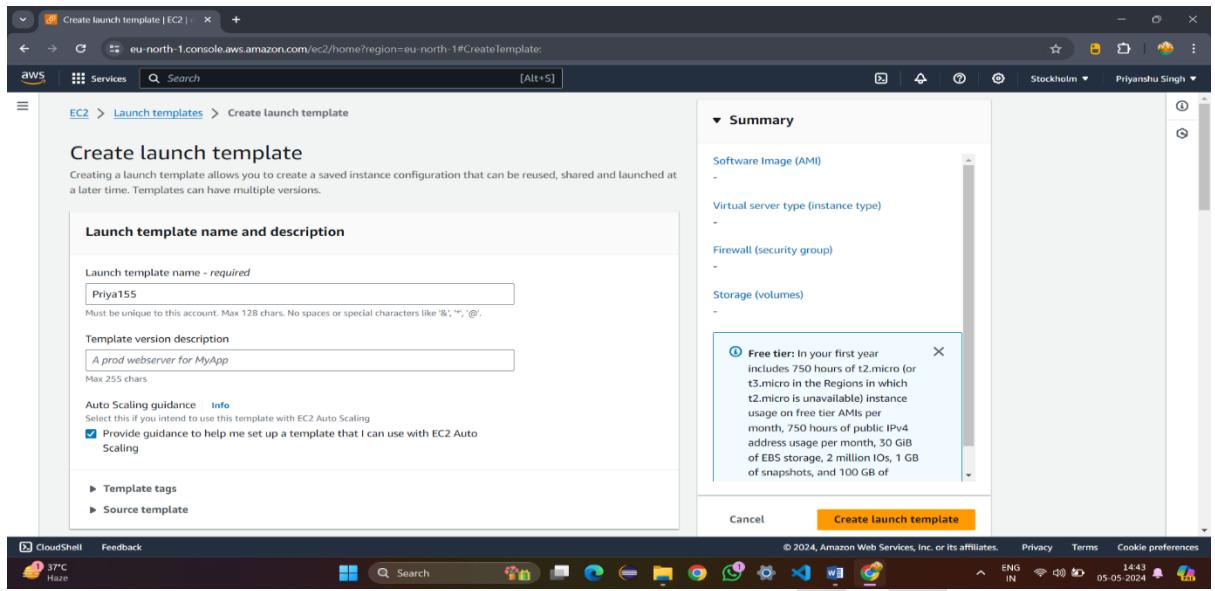
- At first go to EC2 then in Instances click on Launch Templates.

The screenshot shows the AWS EC2 Dashboard for the Europe (Stockholm) Region. On the left sidebar, under 'Instances', 'Launch Templates' is selected. The main content area displays various EC2 resources: Instances (running) 0, Auto Scaling Groups 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 4, Load balancers 0, Placement groups 0, Security groups 5, Snapshots 0, and Volumes 0. Below this, there's a 'Launch instance' section with a 'Launch instance' button and a 'Migrate a server' button. To the right, there's a 'Service health' section showing 'AWS Health Dashboard' and a note that the service is operating normally. Further right is the 'EC2 Free Tier' info section, which indicates 0 EC2 free tier offers in use, an end-of-month forecast, and offers exceeding the free tier limit. At the bottom, there's an 'Account attributes' section and a 'View all AWS Free Tier offers' link.

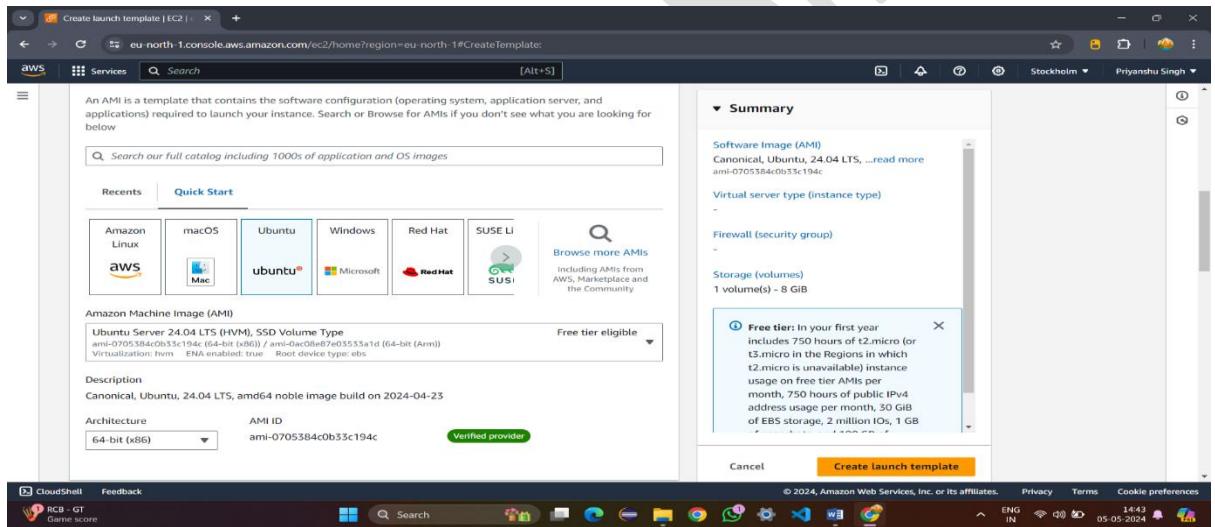
- Now click on Create launch template.

The screenshot shows the 'Launch Templates' page in the AWS EC2 console. The left sidebar shows 'Instances' selected. The main area displays a table with one entry: 'Launch Templates (1/1)'. The table includes columns for 'Launch Template ID', 'Launch Template Name', 'Default Version', 'Latest Version', 'Create Time', and 'Created By'. A prominent orange 'Create launch template' button is located at the top right of the table area.

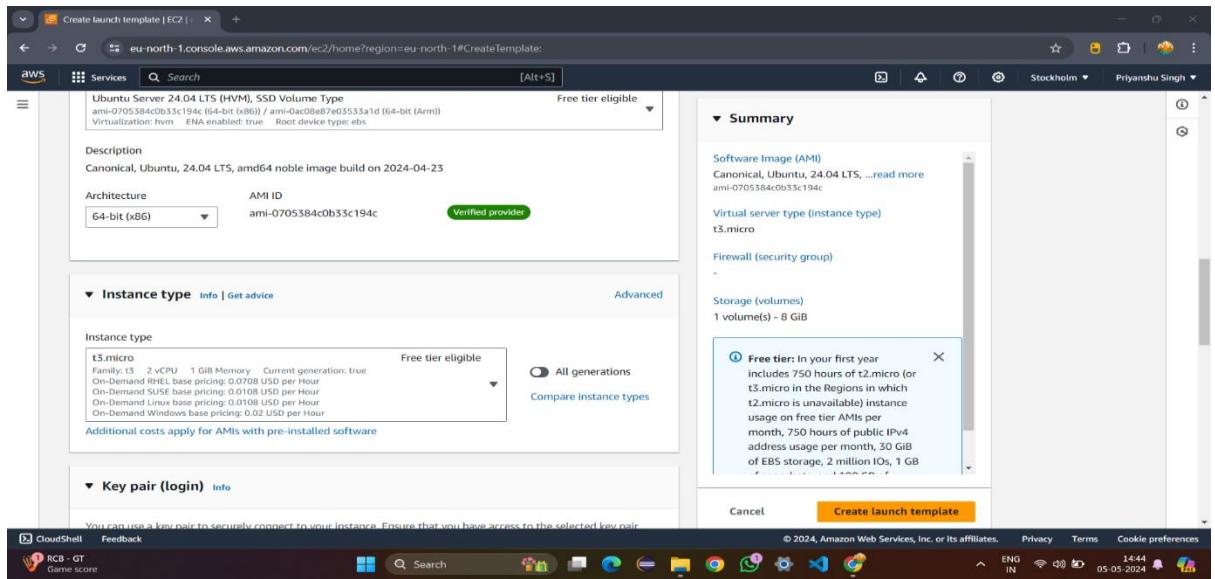
- After it give Launch template name and then click on checkbox for Auto Scaling guidance.



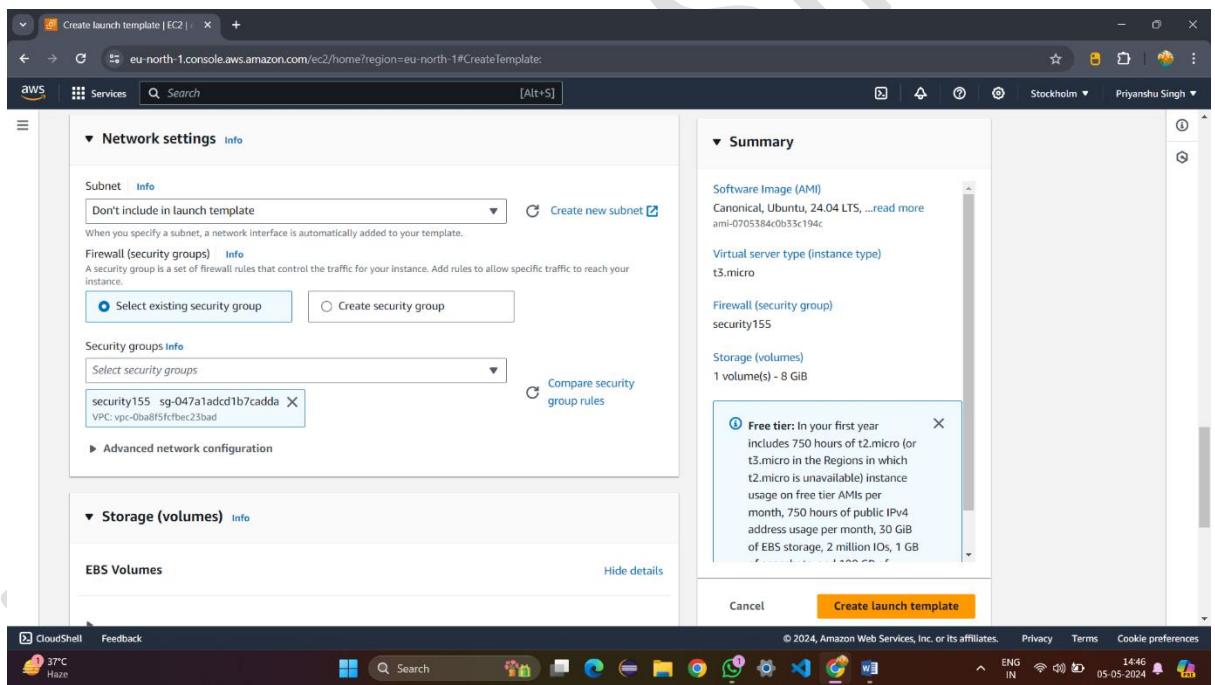
#### 4. Then in quick start select ubuntu.



#### 5. Now select instance type –either t2.micro or t3.micro.



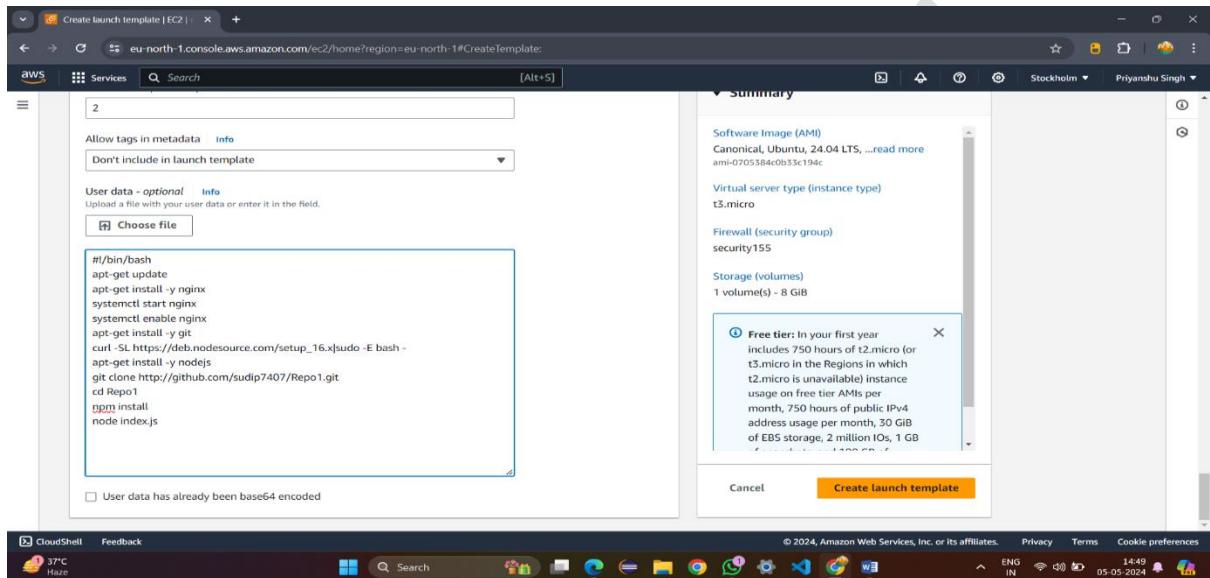
## 6. After it select existing security group.



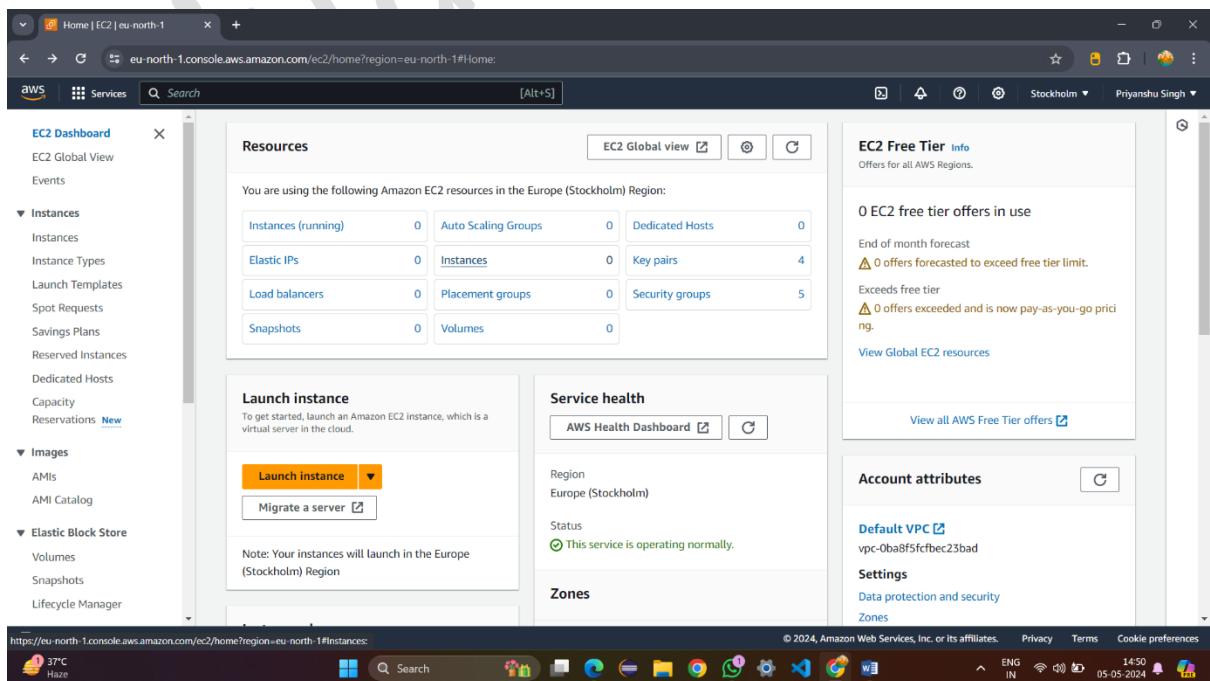
## 7. Now go to Advanced details and then in User data write given codes and then Create launch template.

- `#!/bin/bash`
- `apt -get update`
- `apt -get install -y nginx`
- `systemctl start nginx`
- `systemctl enable nginx`
- `apt -get install -y git`

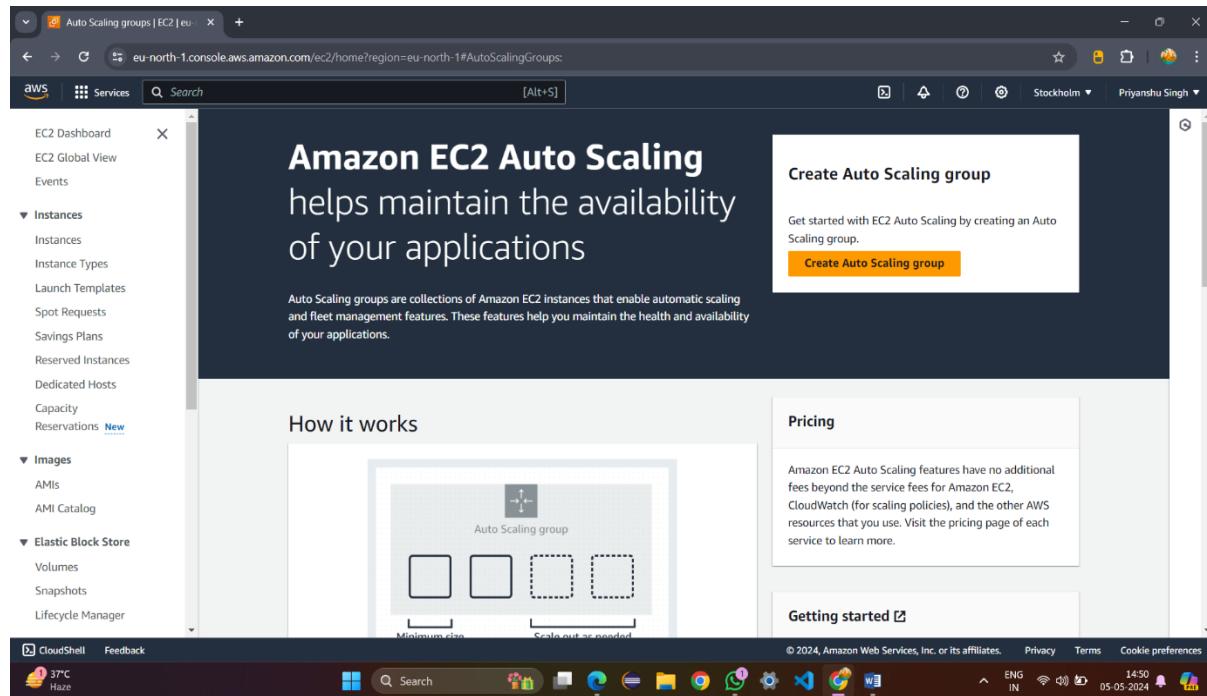
- curl -SL https://deb.nodesource.com/setup\_16.x|sudo -E bash
- 
- apt -get install -y nodejs
- git clone https://github.com/Rohan26Chakraborty/myrepo.git
- cd myrepo
- cd myrepo
- npm install
- node index.js



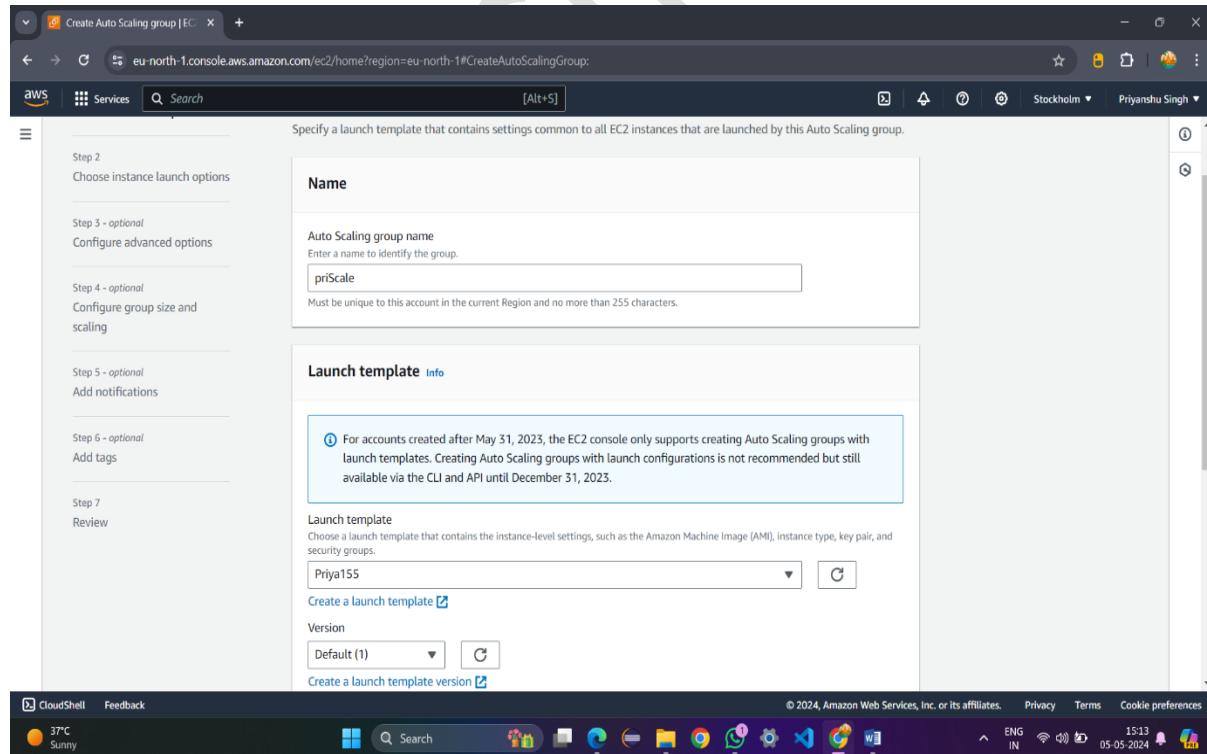
## 8. After creating launch template then click on Auto Scaling Groups in left side.



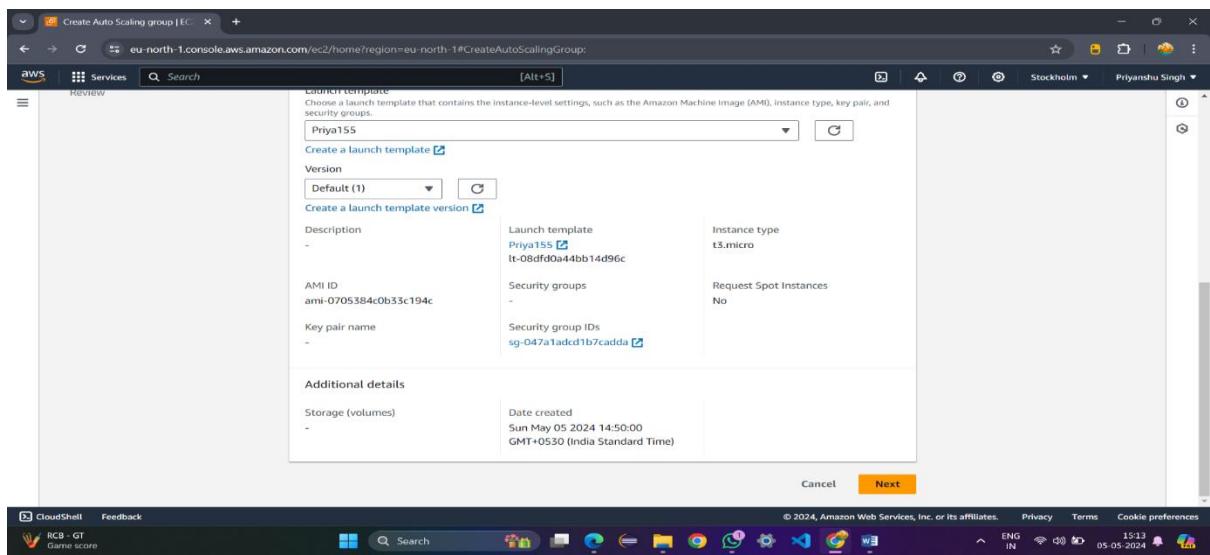
## 9. Click on Create Auto Scaling group.



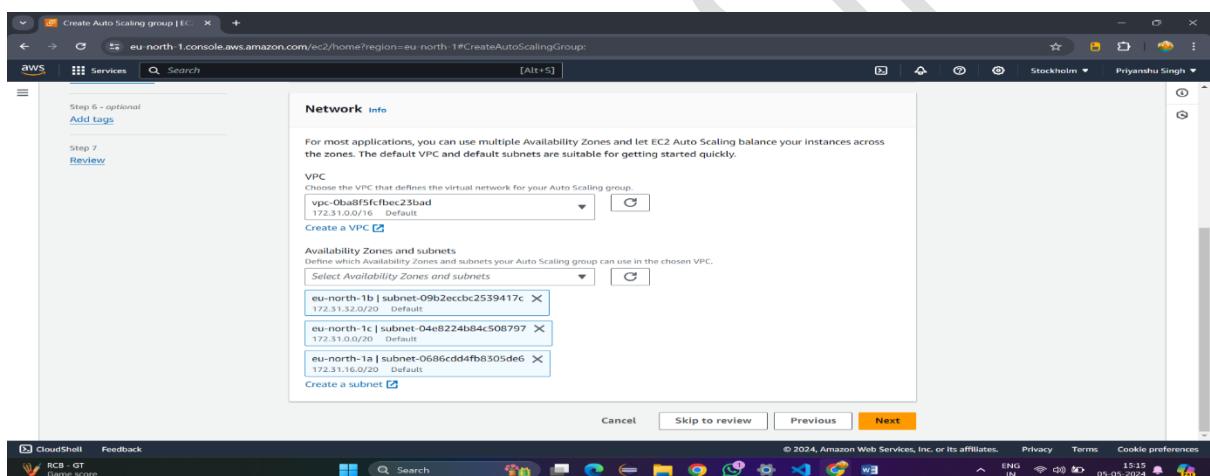
## 10. Now give Auto scaling group name and then select your recently launched template.



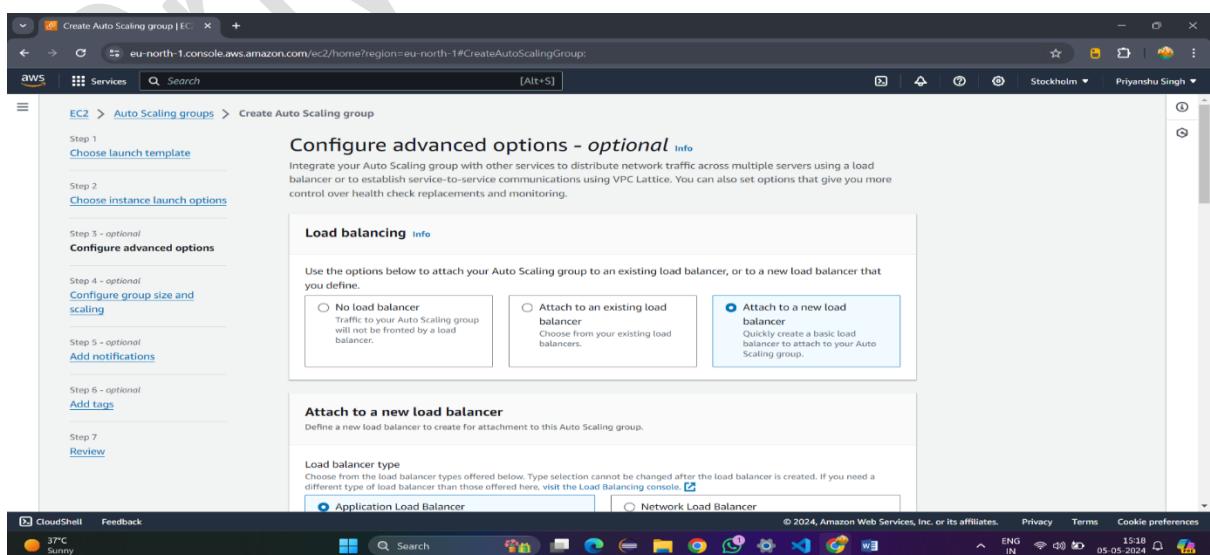
## 11. After it click on Next.



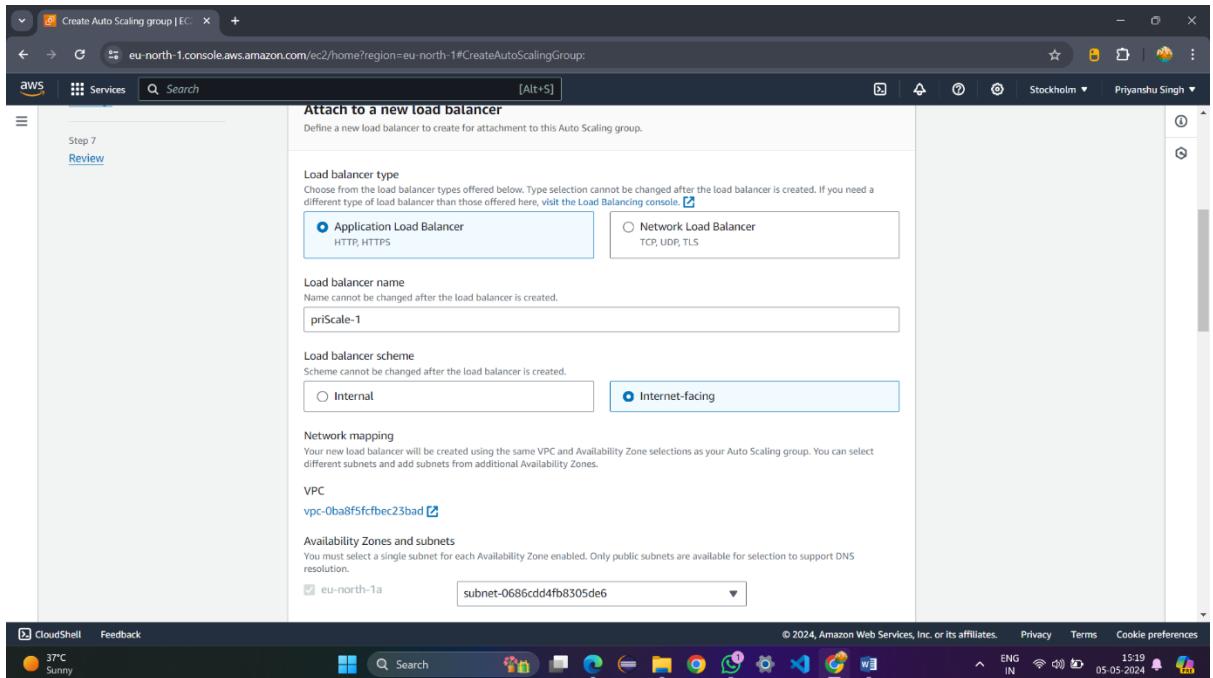
12. After it select availability zones and subnets then click on Next.



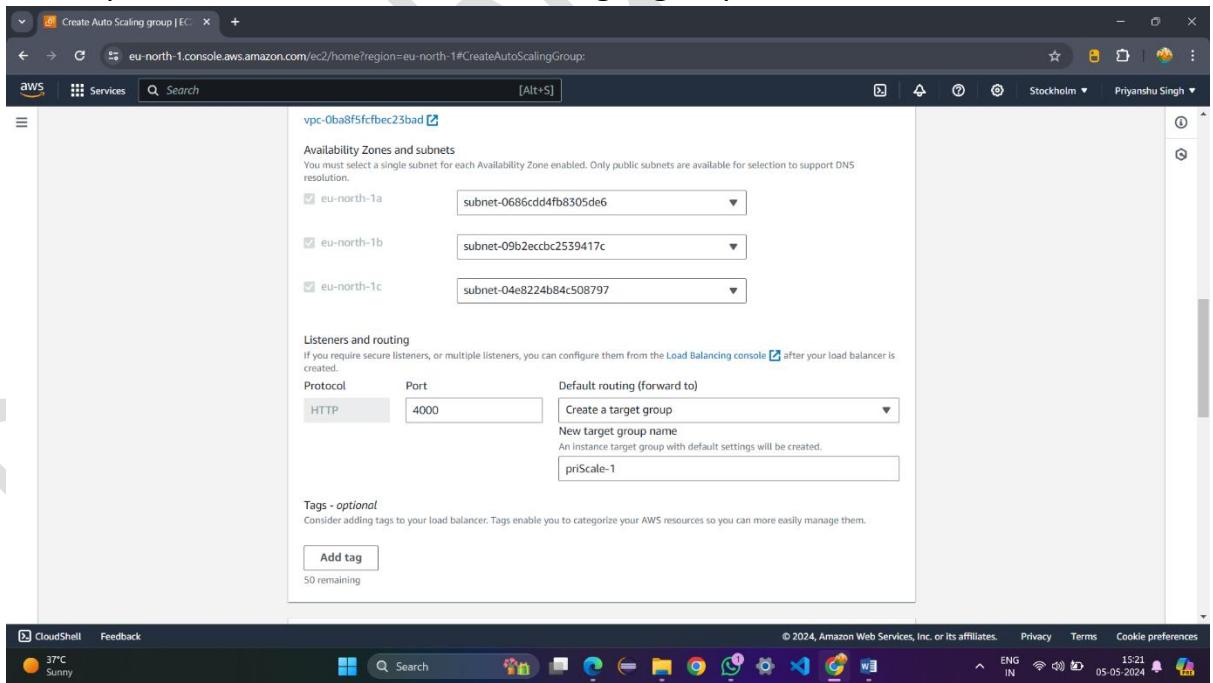
13. Click on Attach to a new load balancer .



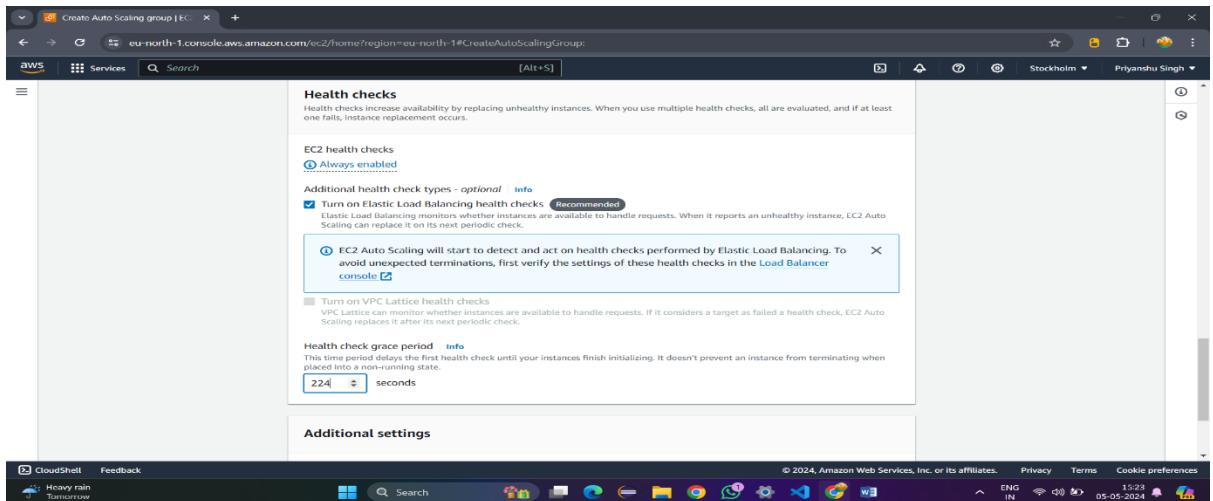
14. Now select Application Load Balancer as load balancer type and Internet-facing as Load balancer scheme.



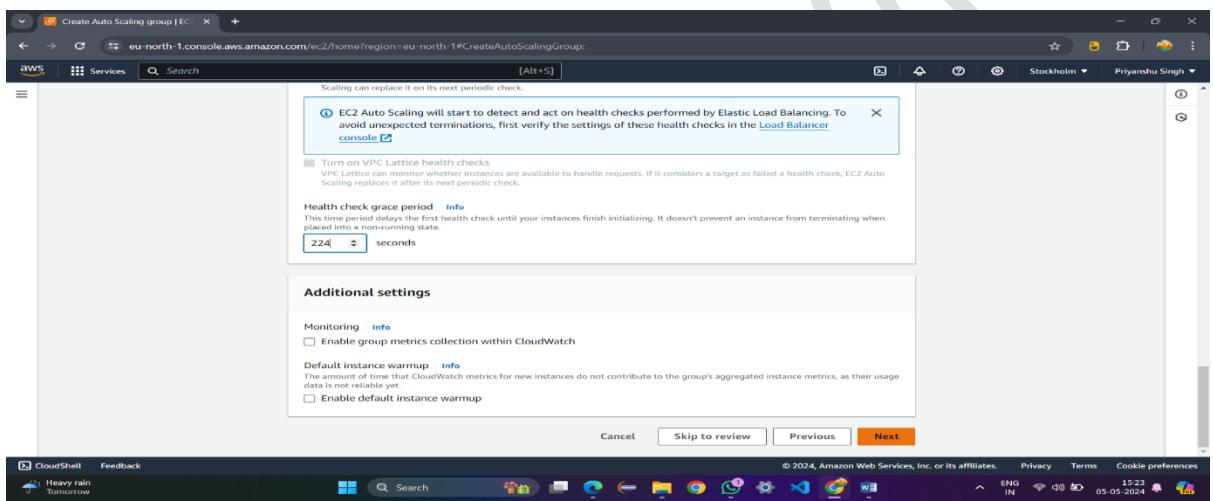
15. Select port no 4000 and select New target group name.



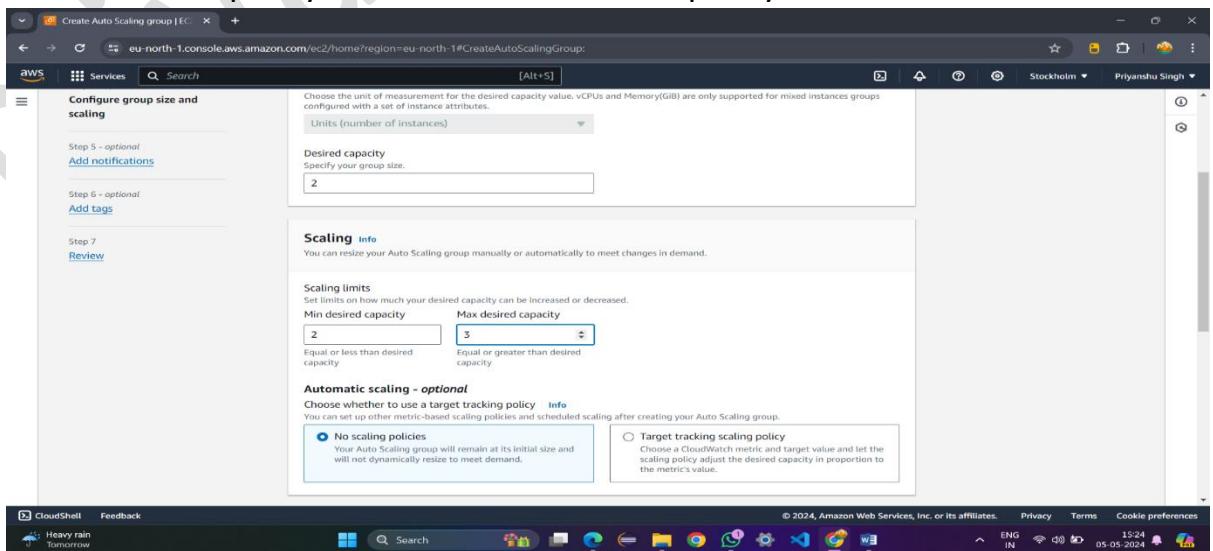
16. Now turn on Elastic load balancing health check and in health check grace period give 224 seconds.



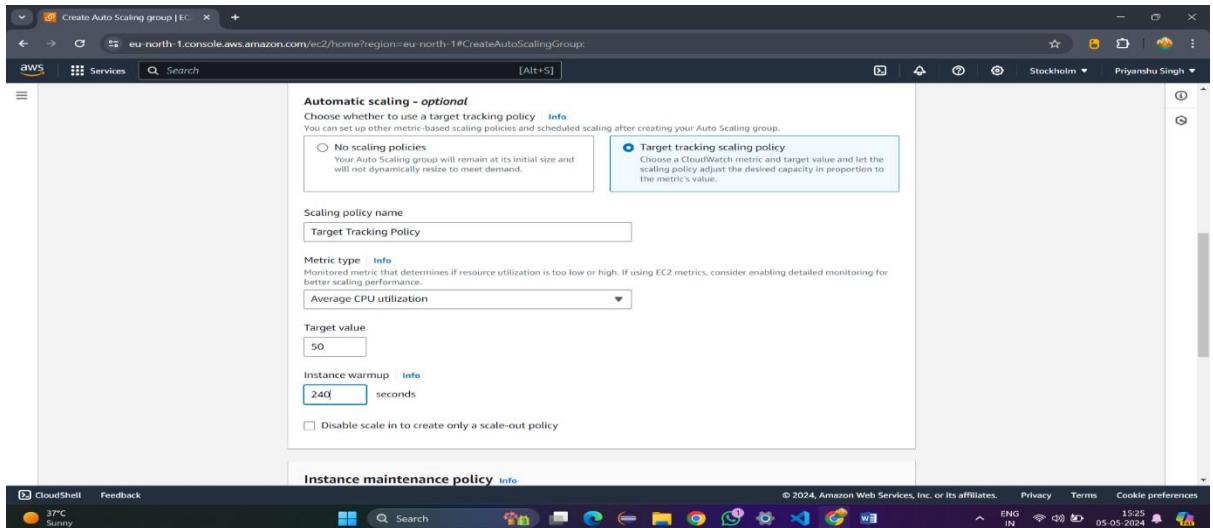
17. After it click to next.



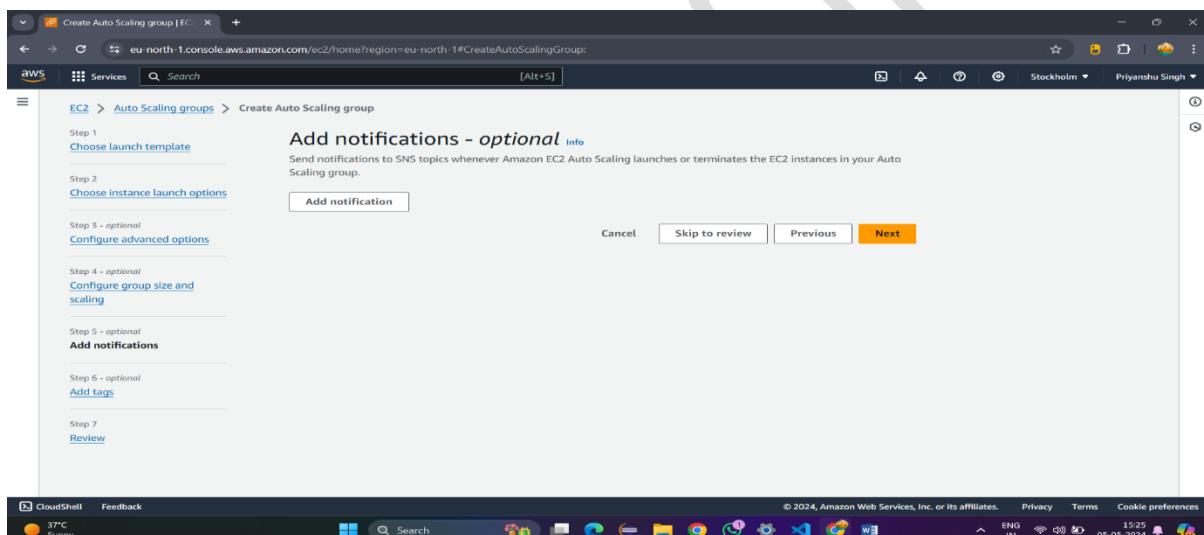
18. After next select 2 in Desired capacity in group size and in scaling 2 in Min desired capacity and 3 in Max desired capacity.



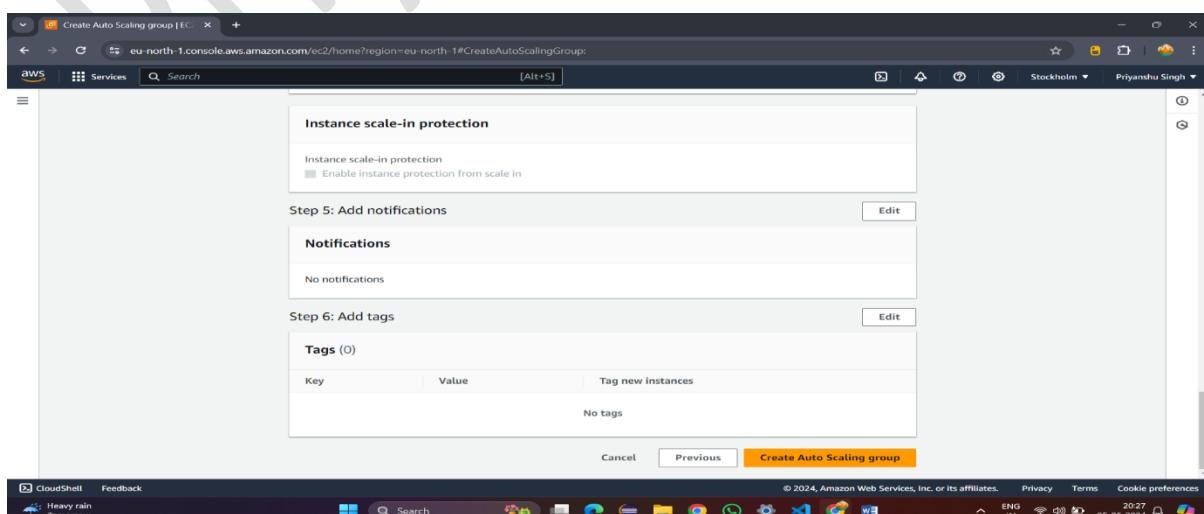
19. Next in automatic scaling select Target tracking scaling policy and give 240 in Instance warmup



20. After it click on next, then again next.



21. Now click on Create Auto Scaling group.



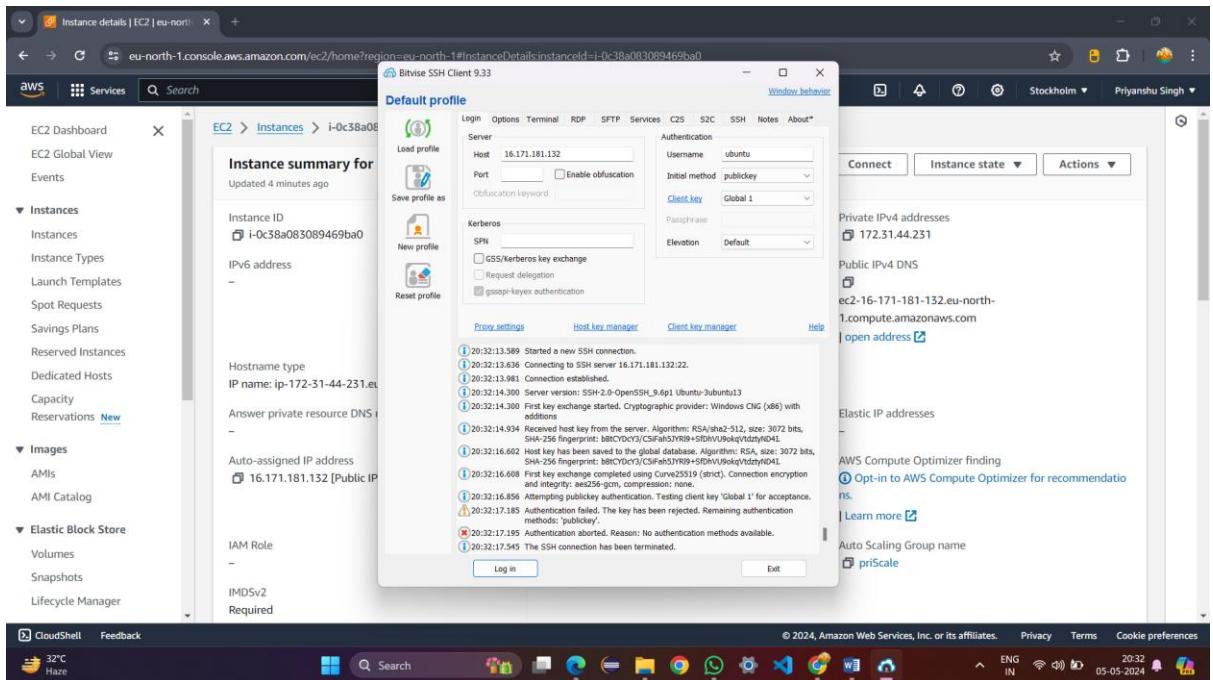
22. After it come back to EC2 dashboard and go to Instance and here you can see 2 instances running as minimum capacity 2 chosen.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main area displays a table titled 'Instances (2) Info' with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. Two instances are listed: one with Instance ID i-0c38a083089469ba0 and another with i-0268a7aa92585a21e, both in the 'Running' state. Below the table is a modal window titled 'Select an instance'.

23. Now click on any instance and then copy public IPv4 address.

The screenshot shows the AWS EC2 Instance details page for instance i-0c38a083089469ba0. The left sidebar is identical to the previous screenshot. The main content area is titled 'Instance summary for i-0c38a083089469ba0'. It shows various details about the instance, including its IP address (16.171.181.132), state (Running), and VPC information. A tooltip 'Public IPv4 address copied' is visible over the IP address field. The bottom of the screen shows a taskbar with various icons and system status.

24. Open Bitvise ssh client and then paste that public IPv4 address and then in client key manager import that generated key and then do login.



25. Open New terminal console and then write command sudo nano infi.sh. A new .sh file will be created. Now write this code to run an infinite loop.

```
#!/bin/bash
while(true)
do
    echo "Inside Loop"
done
```

```
GNU nano 7.2
#!/bin/bash
while(true)
do
    echo "Inside Loop"
done
```

26. Then do ctrl+x then y then click enter. And write command sudo chmod 777 infi.sh to provide all permissions to file and then to run give sh infi.sh

```
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Sat May 4 16:23:14 UTC 2024

System load: 0.08 Temperature: -273.1 C
Usage of /: 29.2% of 6.71GB Processes: 115
Memory usage: 57% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 172.31.33.32

Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
3 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

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.

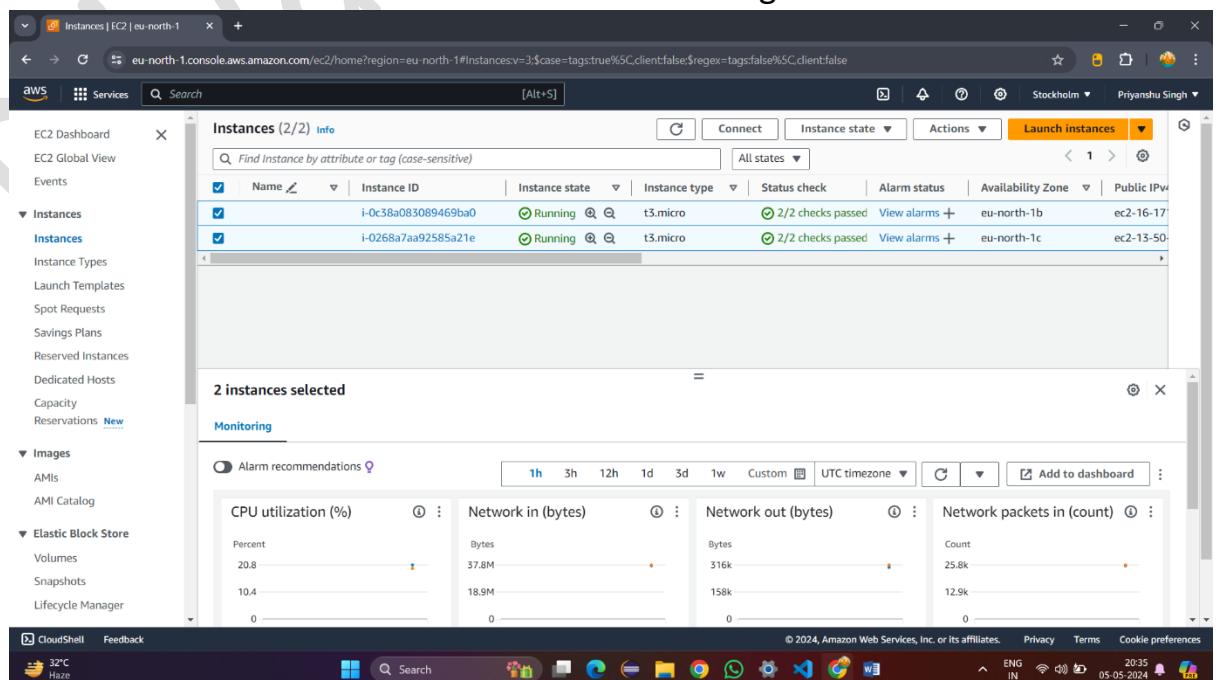
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-33-32:~$ sudo nano infi.sh
ubuntu@ip-172-31-33-32:~$ sudo chmod 777 infi.sh
ubuntu@ip-172-31-33-32:~$ sh infi.sh
```

27. Here you can see infinite loop running.

```
Inside Loop
```

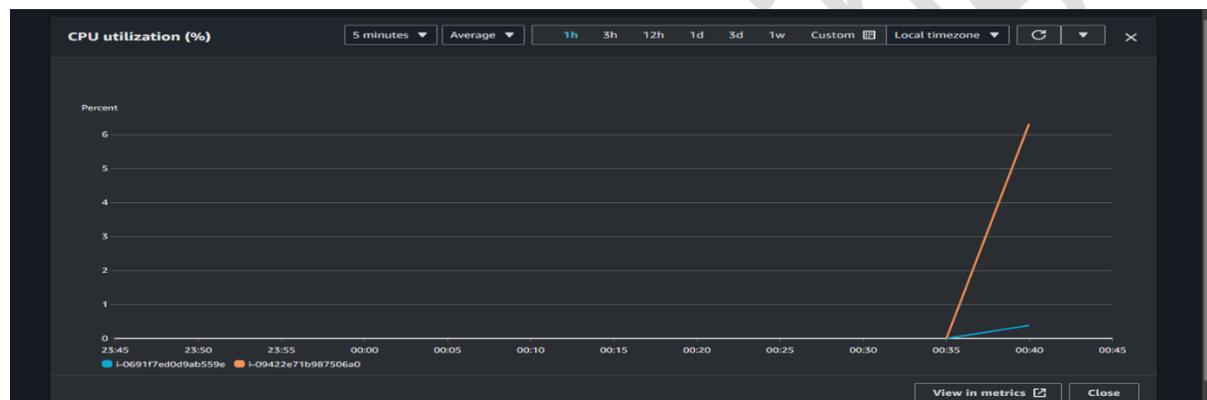
28. Now go back to instances and then select both of them and then in below click on CPU utilization and then click Enlarge.



29. Now select Local Timezone.



30. The graph displays CPU utilization for both instances. When CPU utilization for both instances exceeds then another instance will be created as we have set maximum capacity to 3.



31. At last in instance we can see another instance created.

