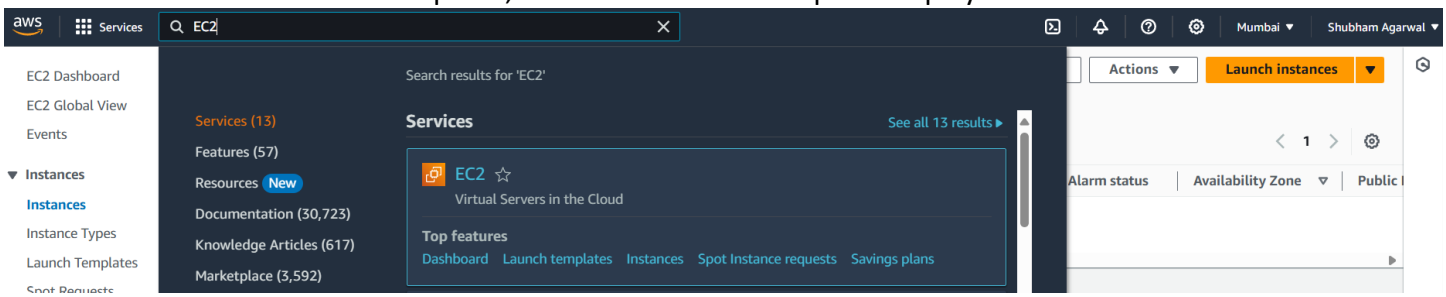
Assignment No : 11

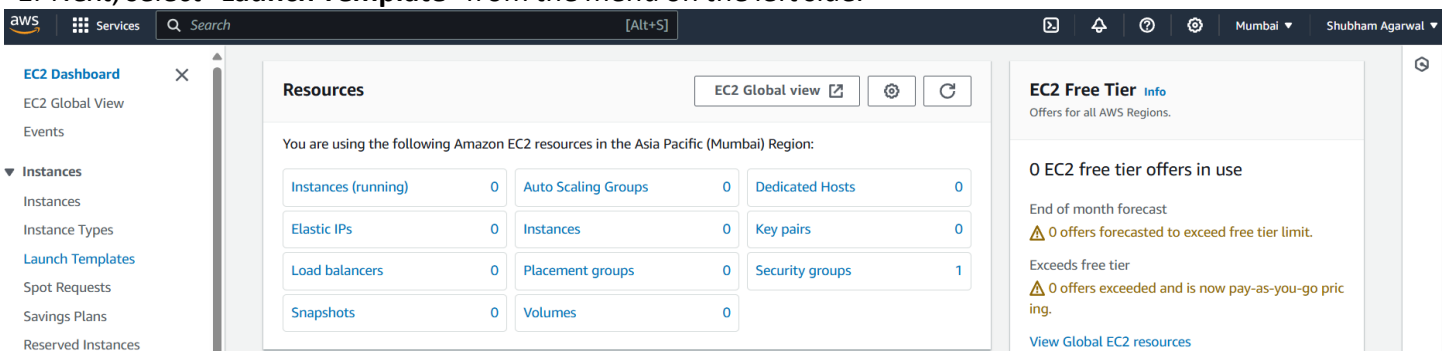
Problem Statement : Build scaling plans in AWS that balance load on different EC2 instances.

Steps:-

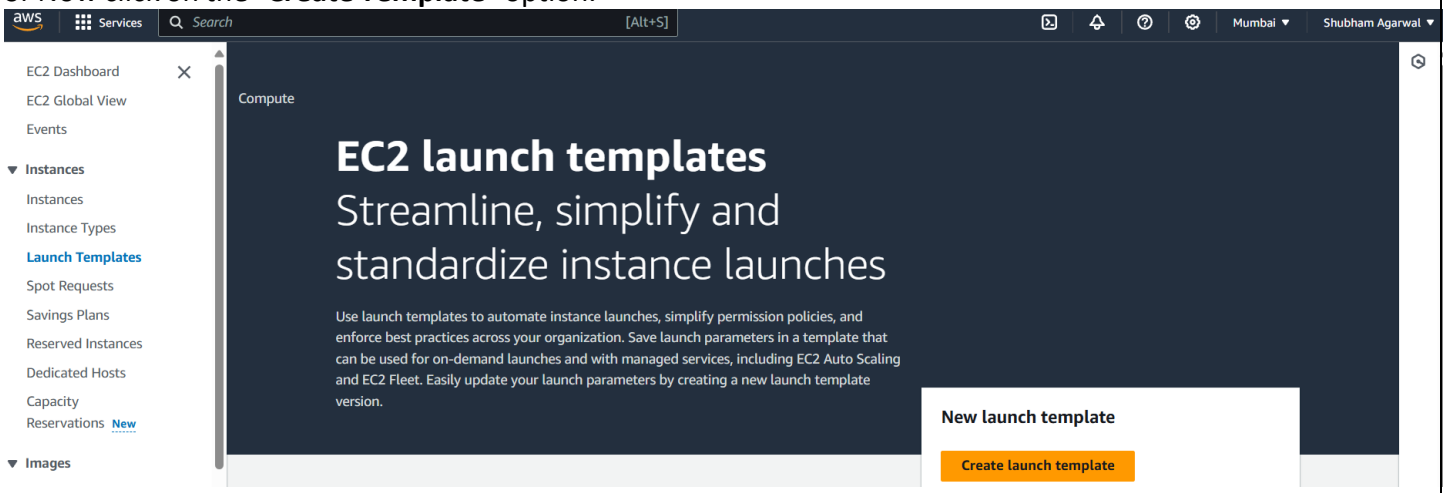
1. Please access AWS and look up EC2, then select the initial option displayed.



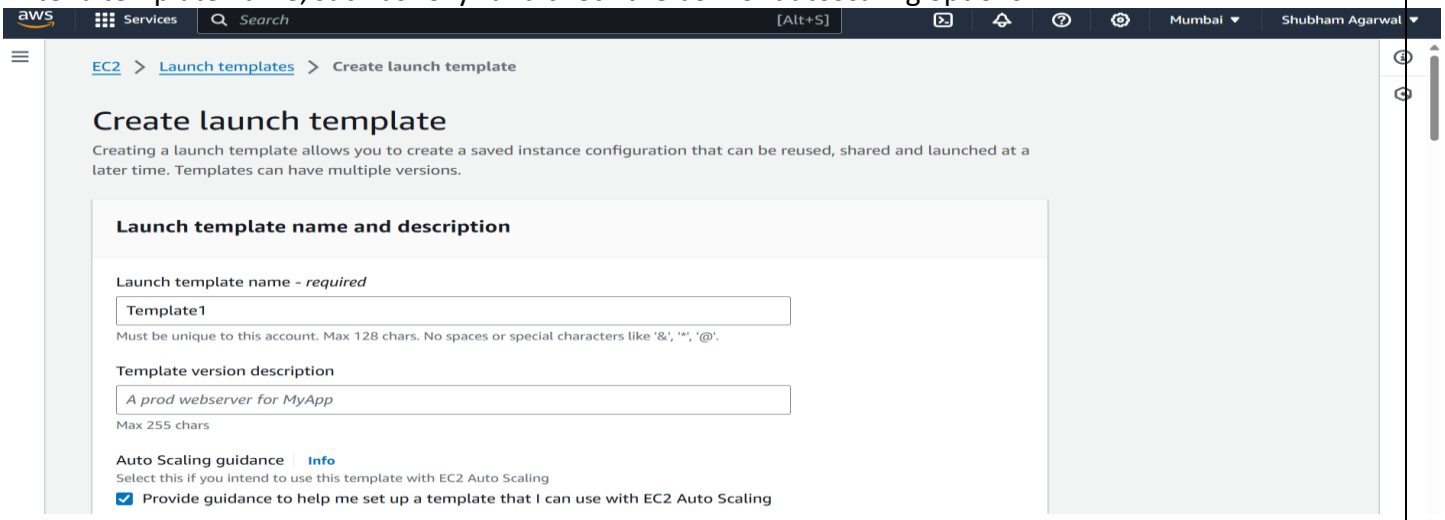
2. Next, select "**Launch Template**" from the menu on the left side.



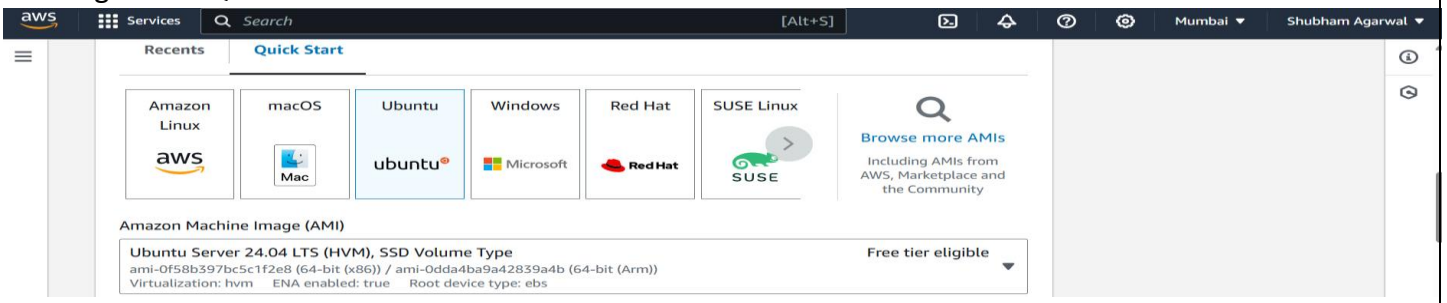
3. Now click on the "**Create Template**" option.



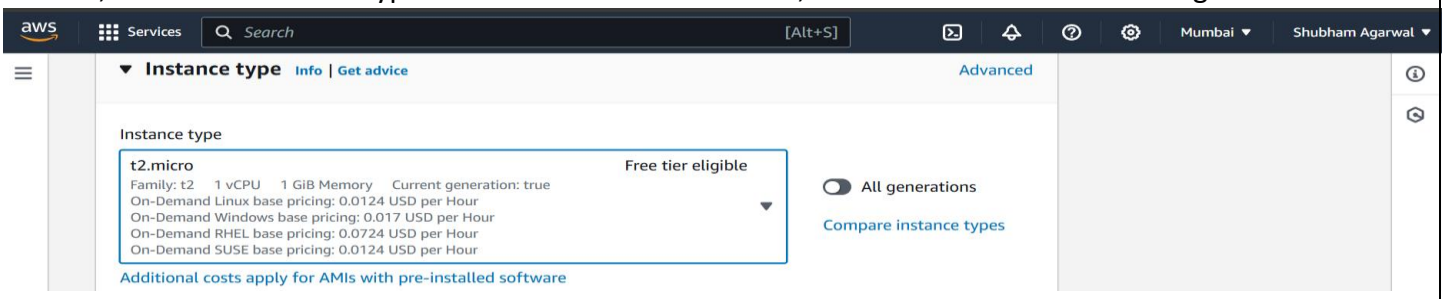
4. Enter a template name, such as "Sky" and check the box for autoscaling options.



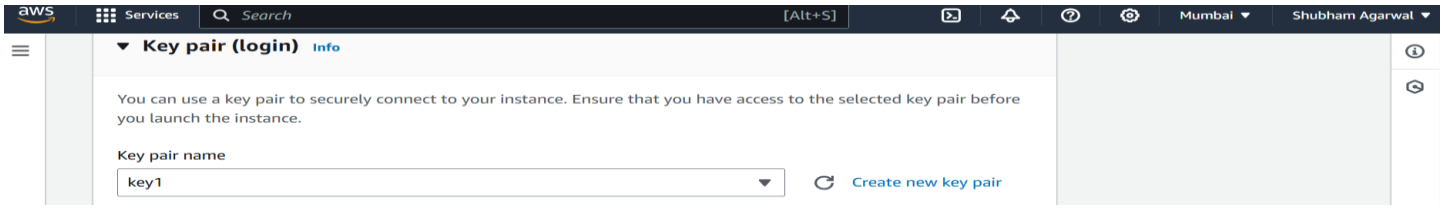
5. Navigate to "**Quick Start**" and choose "**Ubuntu**" from the list of available AMIs.



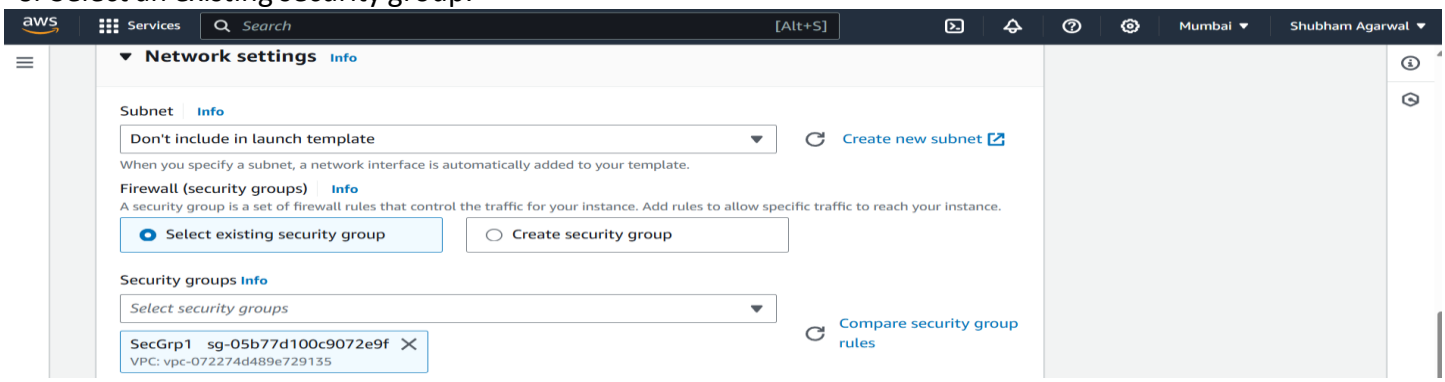
6. Next, select the instance type—either t2.micro or t3.micro, both of which are free tier eligible.



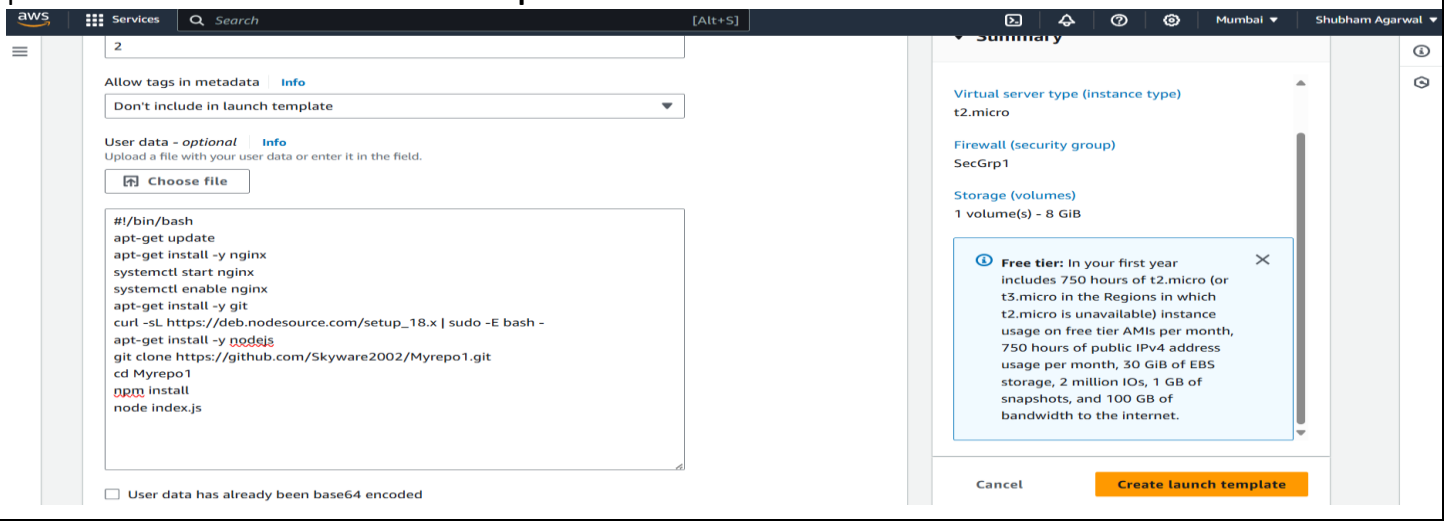
7. Choose either an existing key pair or create a new one if it doesn't exist. For example, use "**keypair1**" as the key pair name.



8. Select an existing security group.

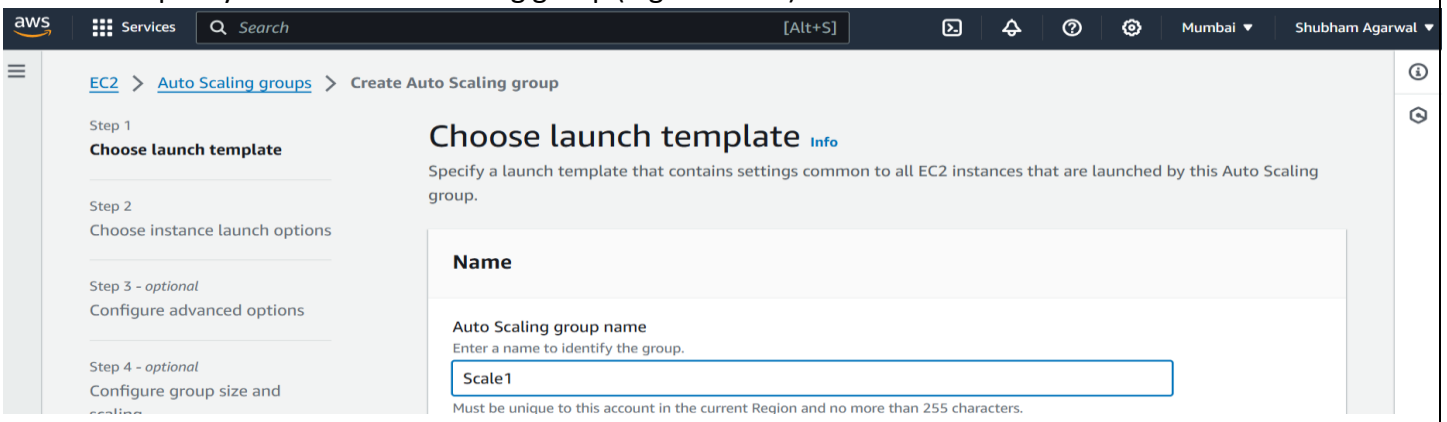


9. Expand the "**Advanced details**" section, navigate to "**User data**", and input the provided code. Then proceed to click on "**Create launch template**".

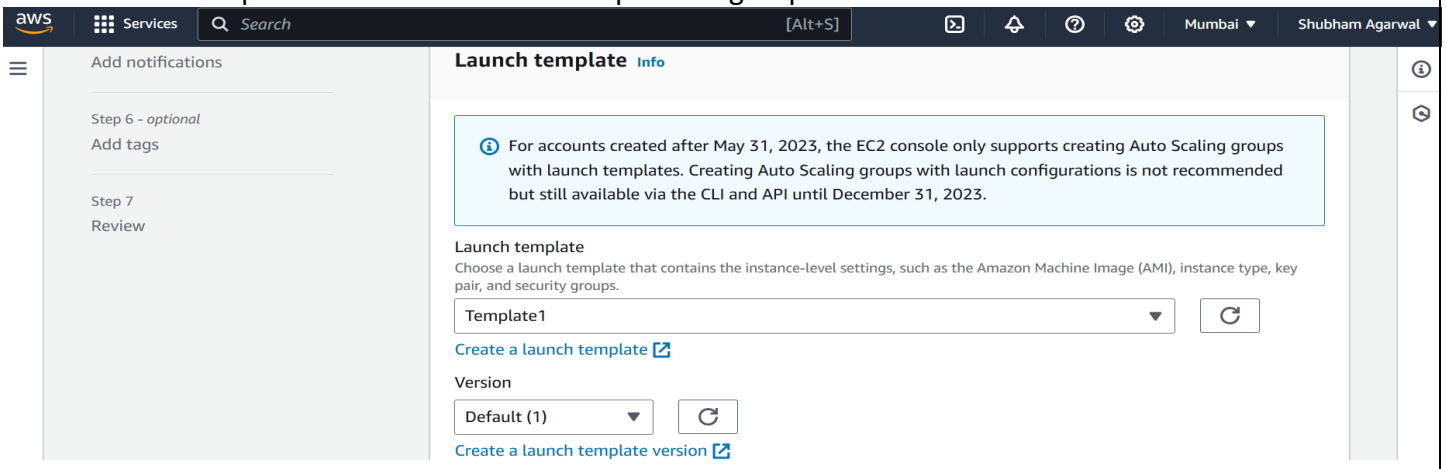


10. Once the launch template has been successfully created, navigate to the left pane and search for "**Auto Scaling Groups**". Then, select "**Create Auto Scaling Group**".

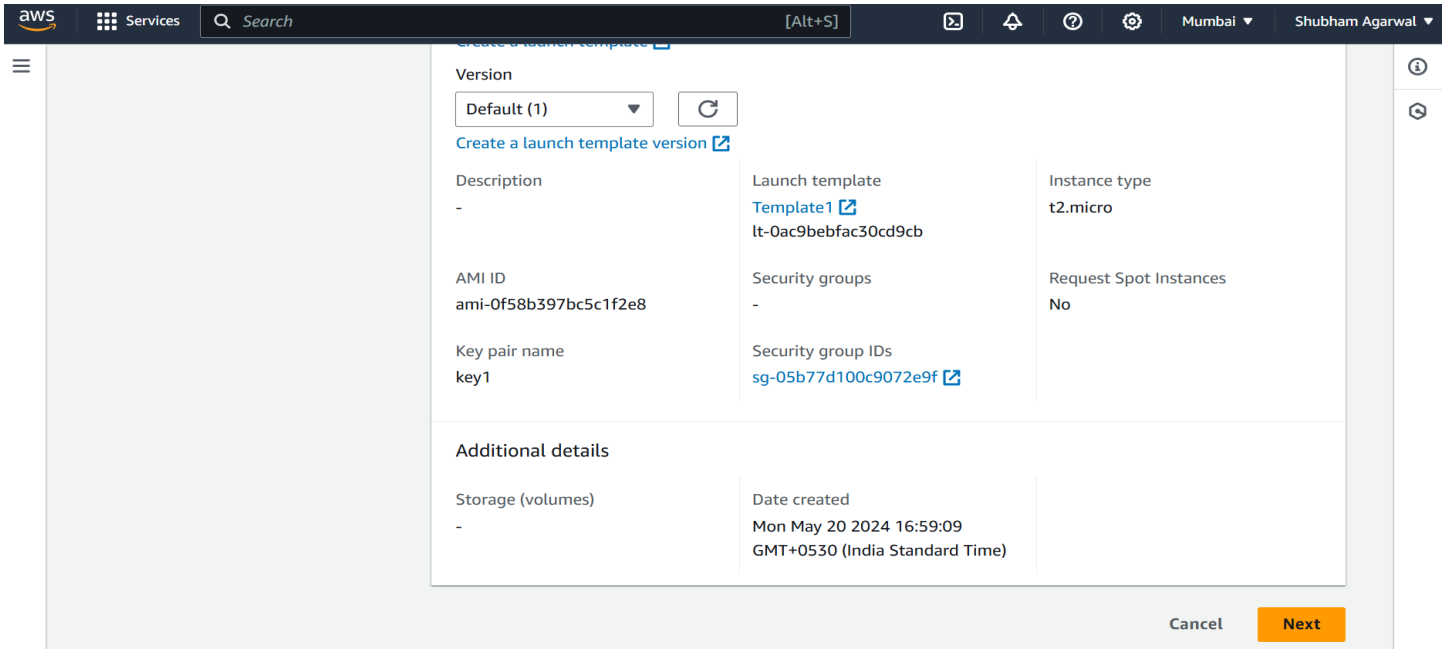
11. Please specify a name for the scaling group (e.g. "**Scale1**").



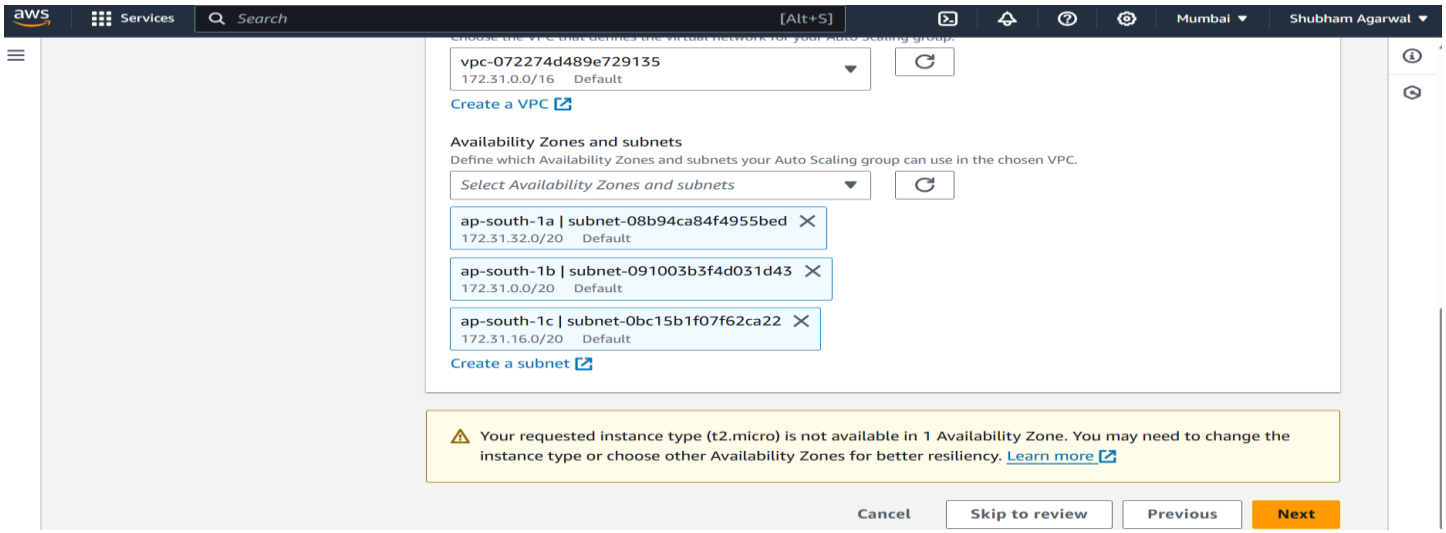
12. Select the template that was created in the preceding steps.



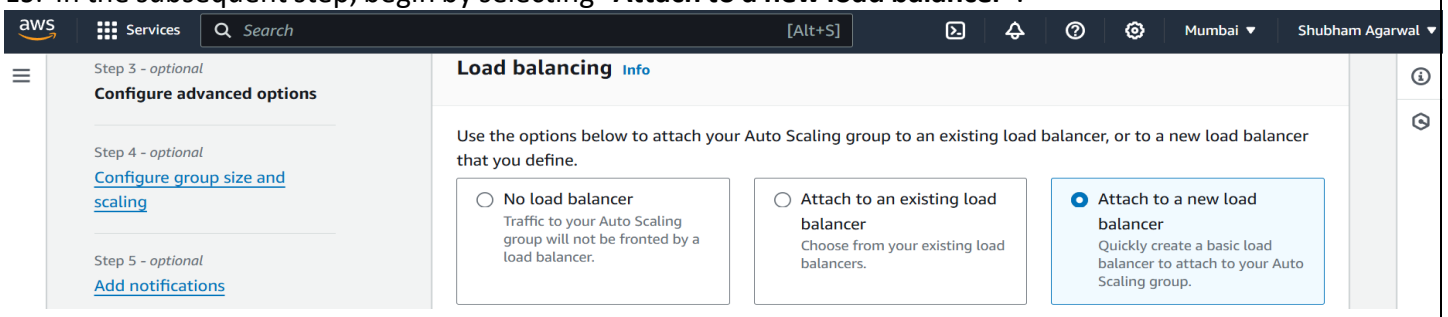
13. Proceed to click on "**Next**".



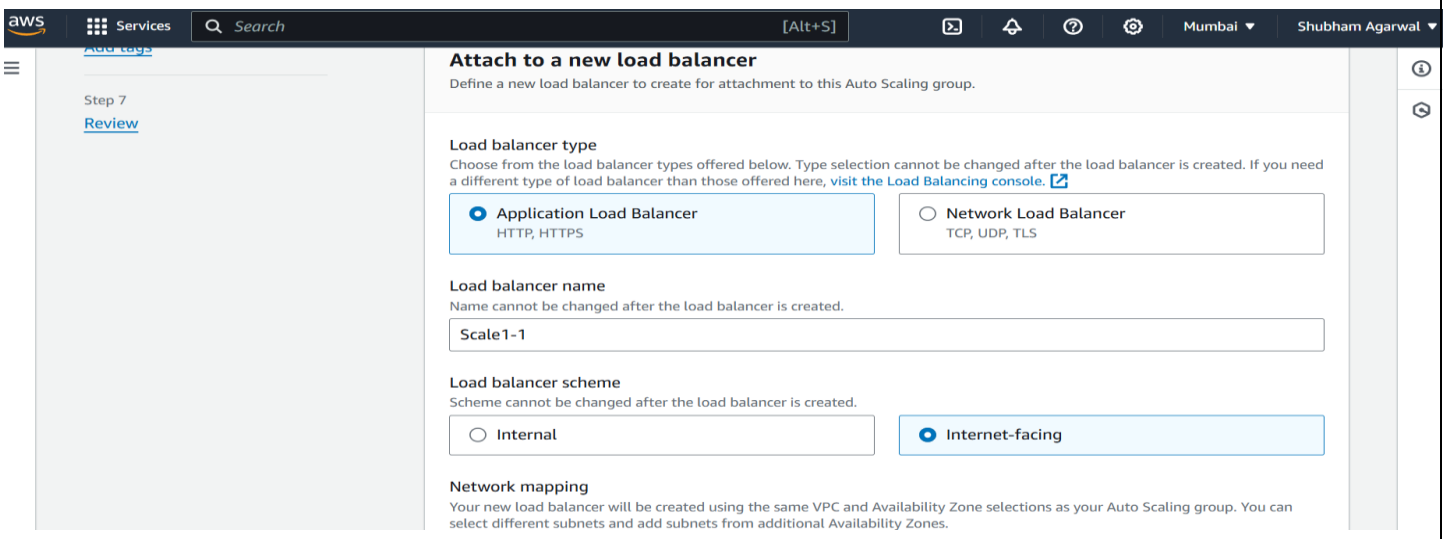
14. In the following step, choose all available availability zones and subnets, then proceed by clicking "**Next**".



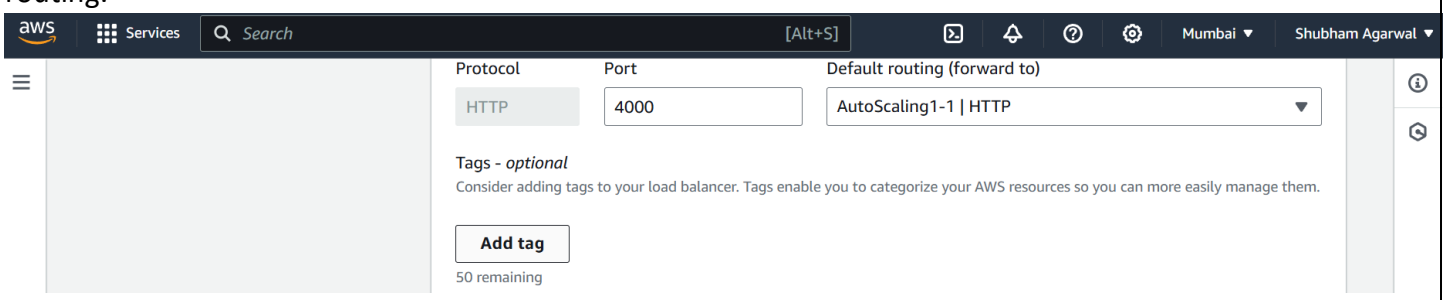
15. In the subsequent step, begin by selecting "**Attach to a new load balancer**".



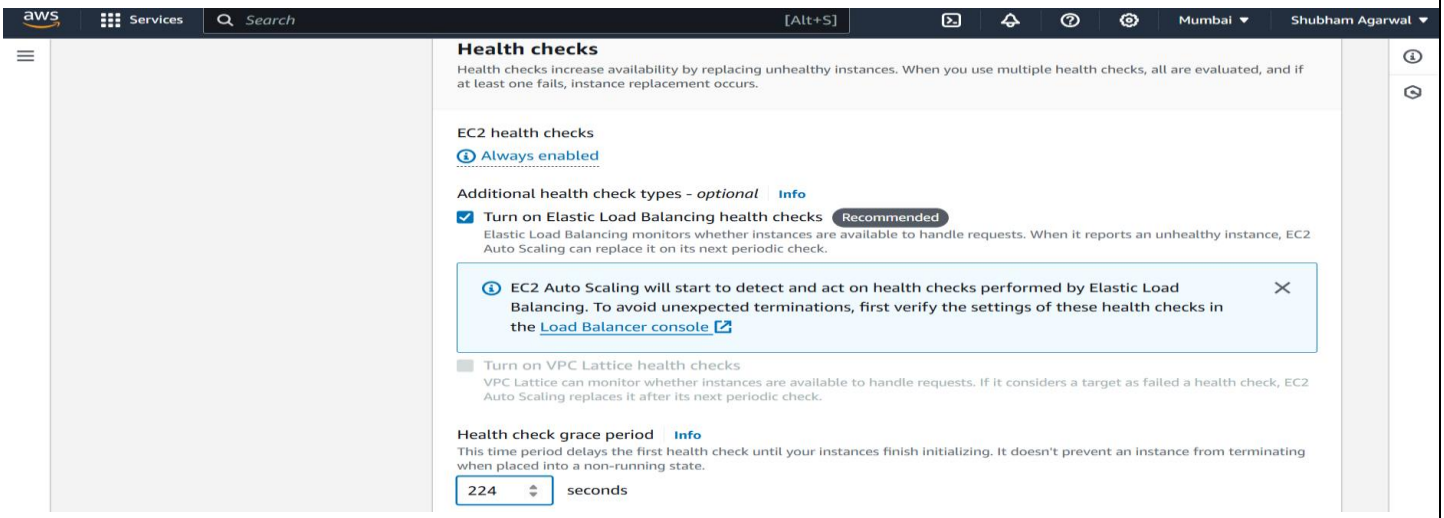
16. Select "**Application Load Balancer**" as the load balancer type and "**Internet-facing**" as the load balancer scheme.



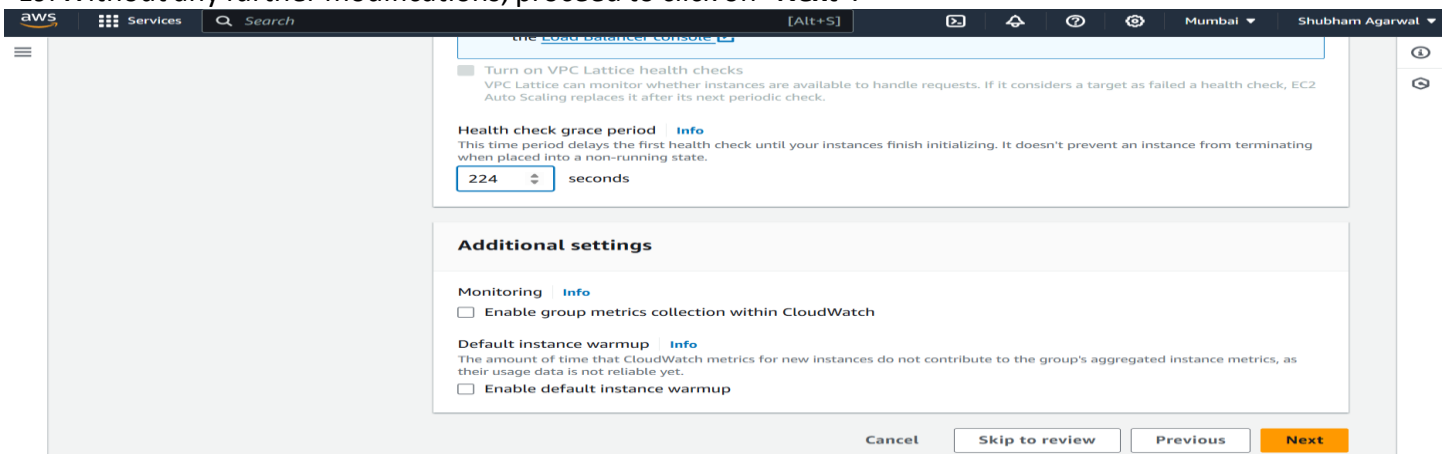
17. Modify the HTTP port number from 80 to 4000 and designate the scaling group created for default routing.



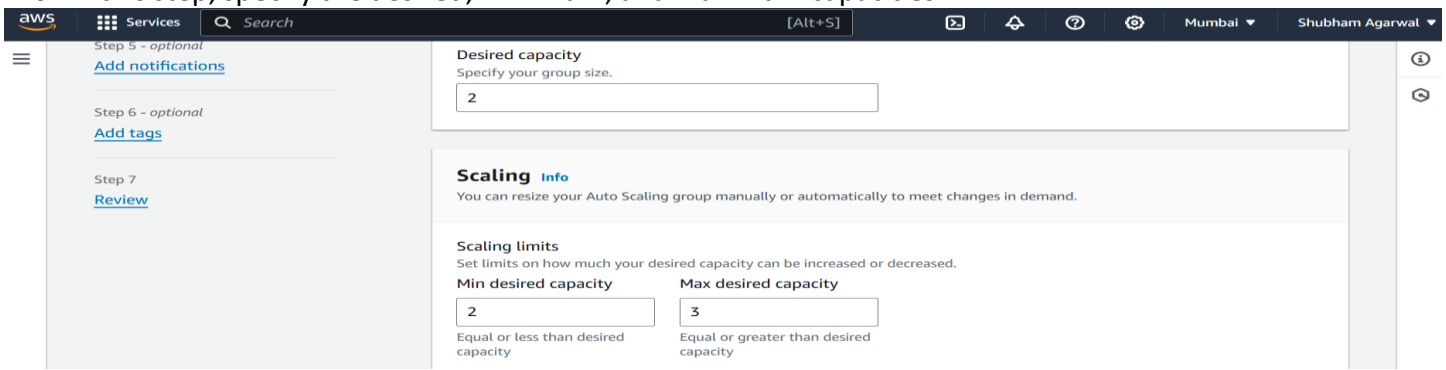
18. Enable the checkbox to activate health checks and specify a "**health check grace period**", set here to 224 seconds.



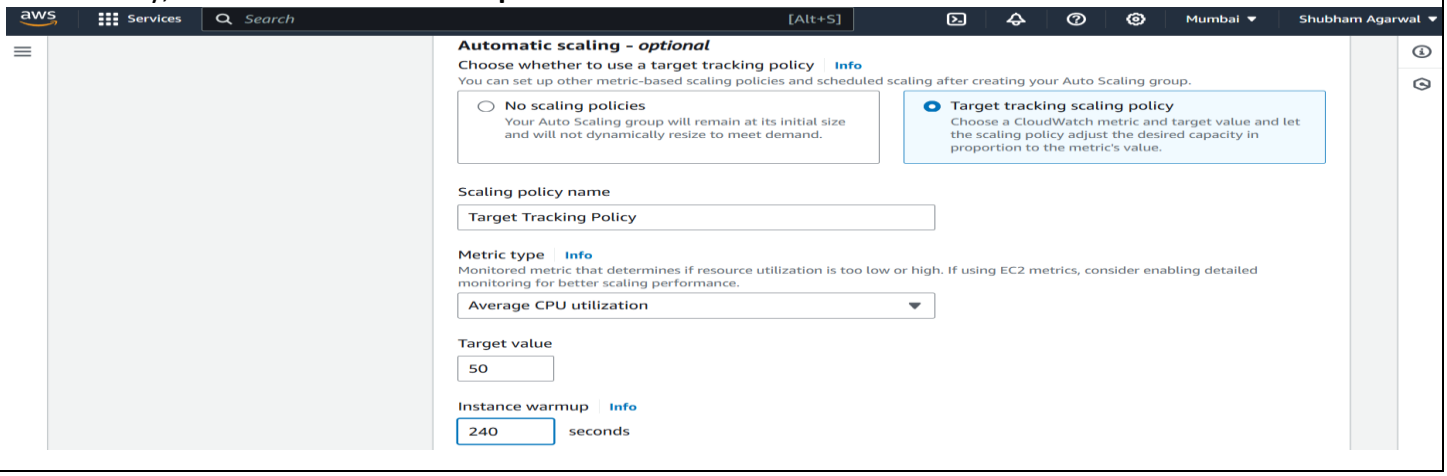
19. Without any further modifications, proceed to click on "**Next**".



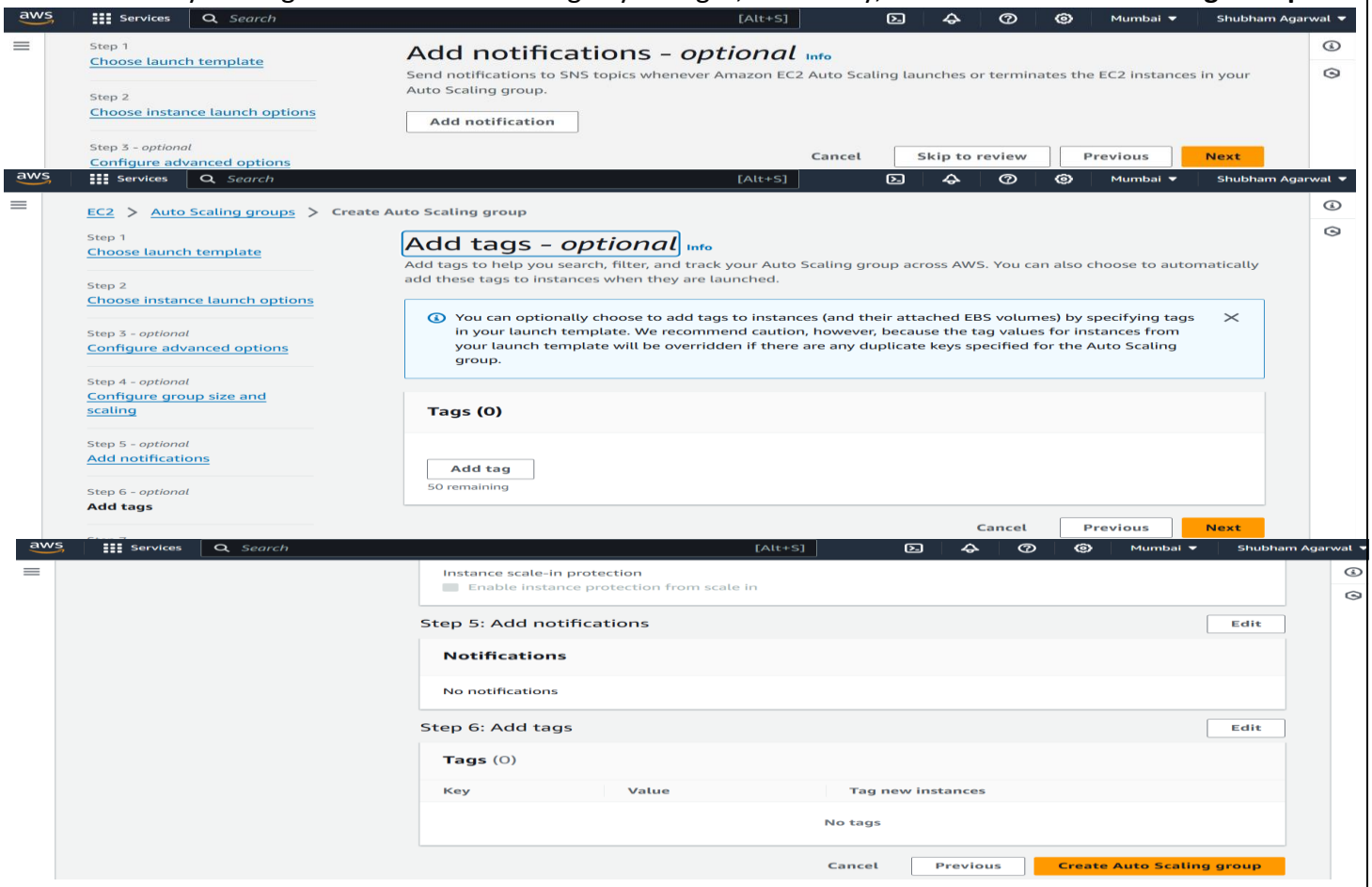
20. In this step, specify the desired, minimum, and maximum capacities.



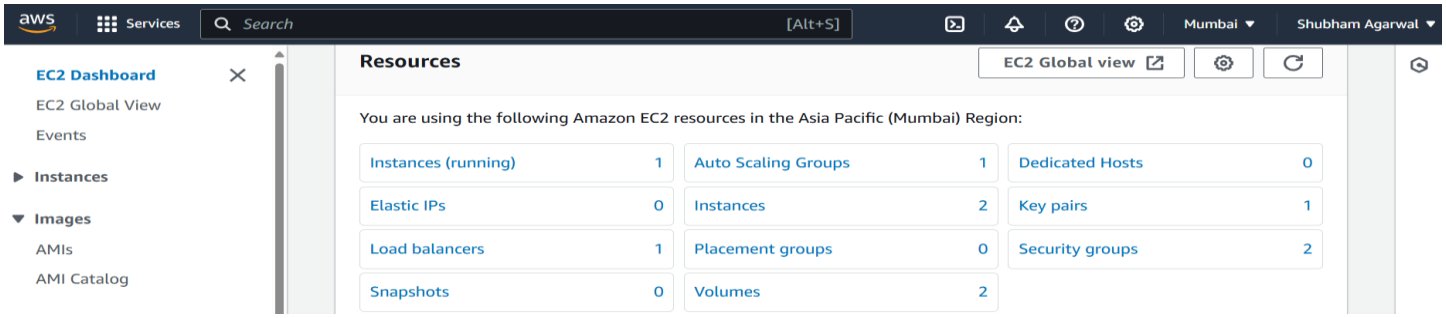
21. Next, opt for the "**Target Tracking Scaling Policy**" and configure the **CPU utilization target value** to 50. Additionally, set the **instance warm-up time** to 240 seconds.



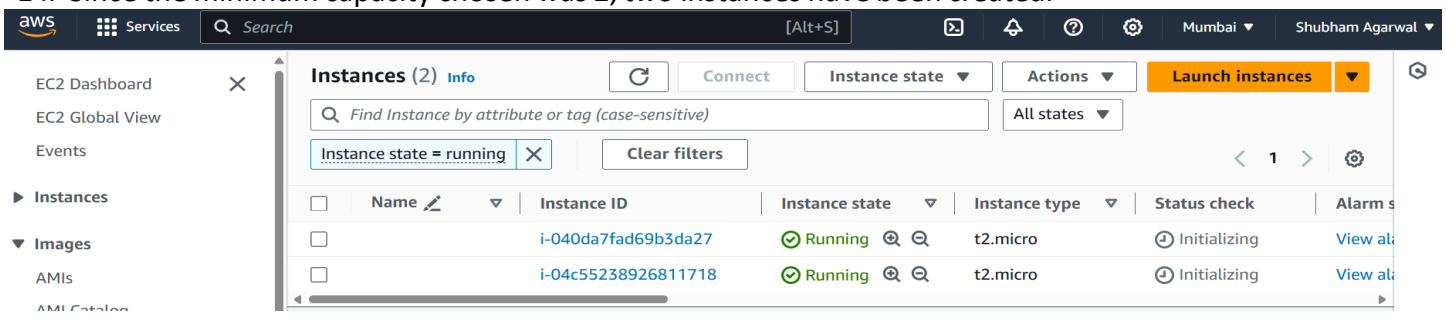
22. Proceed by clicking "**Next**" without making any changes, and finally, select "**Create auto Scaling Group**".



23. After the auto-scaling group is created, return to the EC2 dashboard and navigate to the "**Instances**" section for running instances.

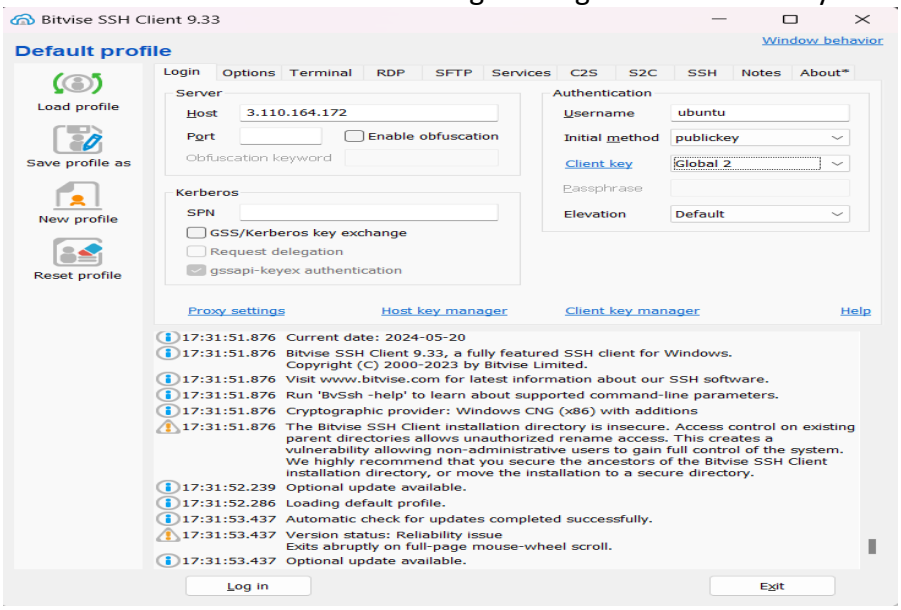


24. Since the minimum capacity chosen was 2, two instances have been created.

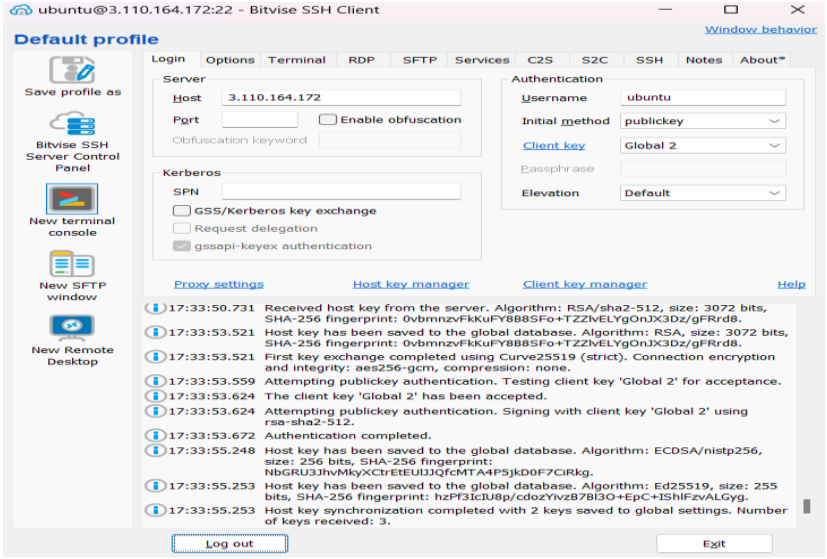


25. Choose any one of the instance IDs and copy the public IPv4 address.

26. Launch the Bitwise SSH Client and log in using the IPv4 address you copied earlier.



27. After successfully logging in, open a new terminal console from the left pane.

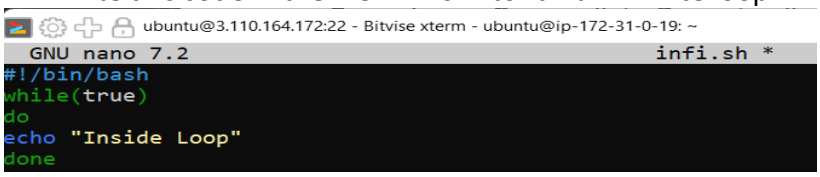


28. Now write the commands in the terminal as follows: -

→***sudo nano infi.sh*** (creates a .sh file)

```
ubuntu@ip-172-31-0-19:~$ sudo nano infi.sh
```

→Write this code in the file “infi.sh” to run an infinite loop.



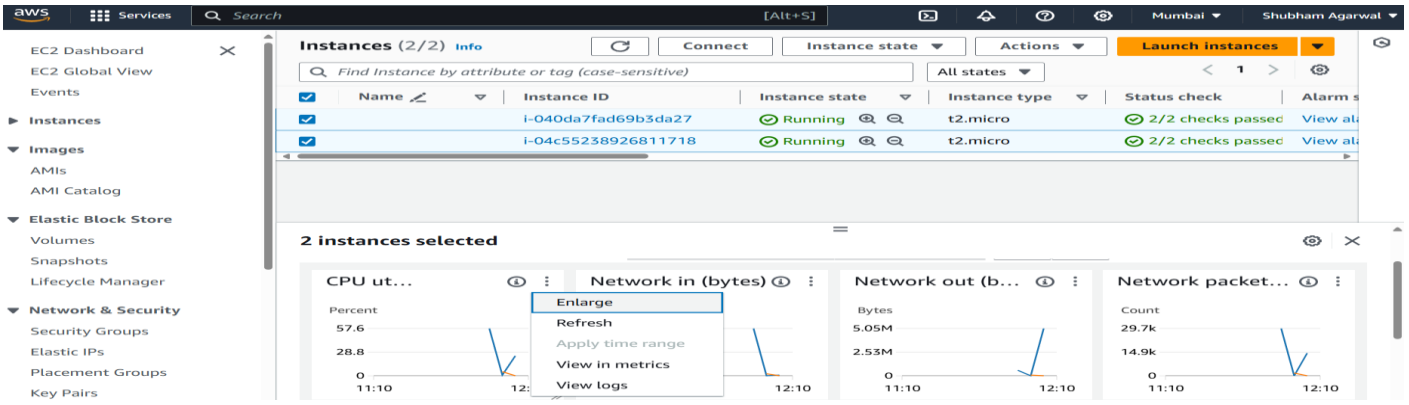
→Press Ctrl+X and write ‘y’ to save the file.

→***sudo chmod 777 infi.sh*** (to provide all permission to the file)

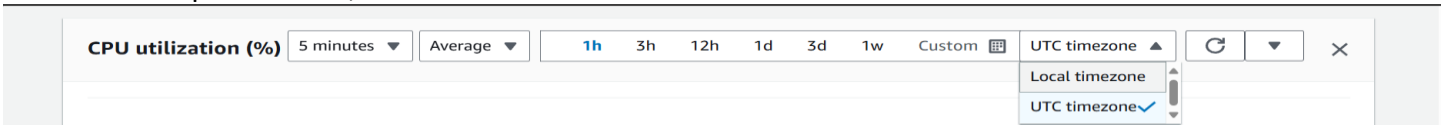
```
ubuntu@ip-172-31-0-19:~$ sudo nano infi.sh
ubuntu@ip-172-31-0-19:~$ sudo chmod 777 infi.sh
```

→***sh infi.sh*** (Run the .sh file)

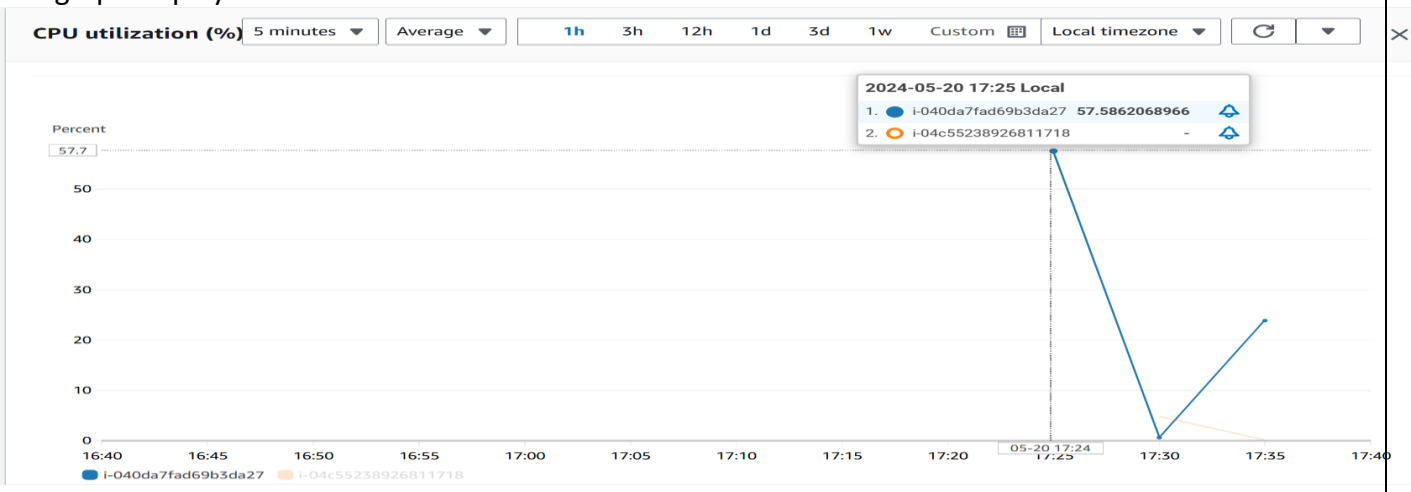
29. Return to AWS and select both running instances. Below, locate the monitoring options, and choose "**CPU utilization**". Then, enlarge the view.



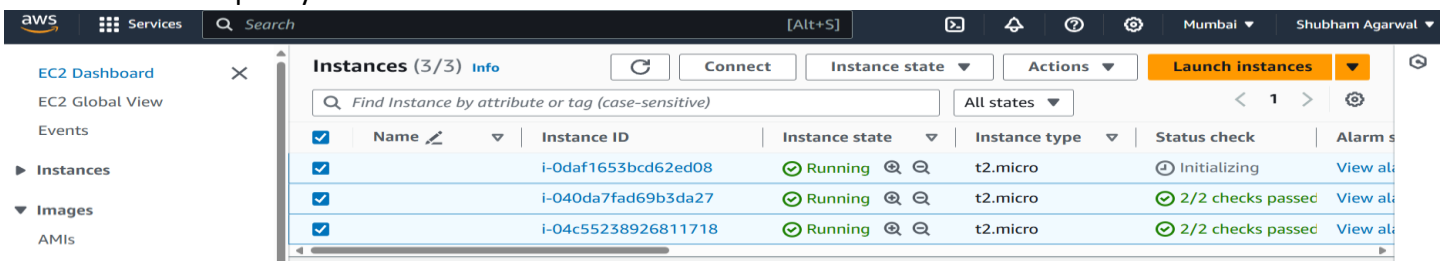
30. From the panel above, select "**Local timezone**."

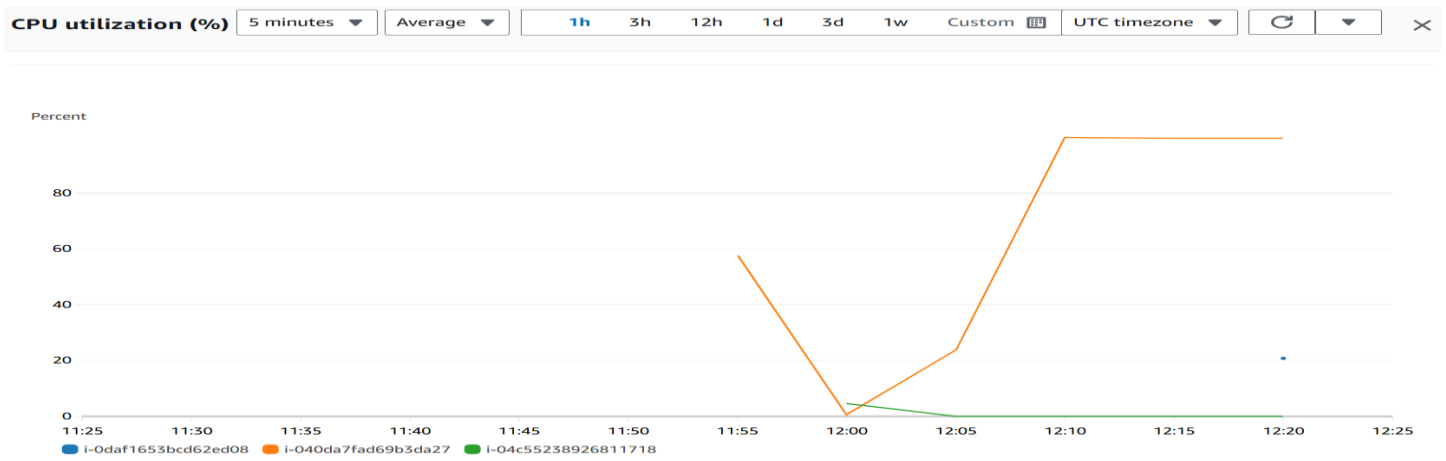


31. The graph displays CPU utilization for both instances.



When the CPU utilization exceeds the limit for both instances, another instance is created, as we have set the maximum capacity to 3.




CPU utilization (%)
5 minutes
Average
1h
3h
12h
1d
3d
1w
Custom
UTC timezone
Percent
80
60
40
20
0
11:25
11:30
11:35
11:40
11:45
11:50
11:55
12:00
12:05
12:10
12:15
12:20
12:25
i-0daf1653bcd62ed08
i-040da7fad69b3da27
i-04c55238926811718