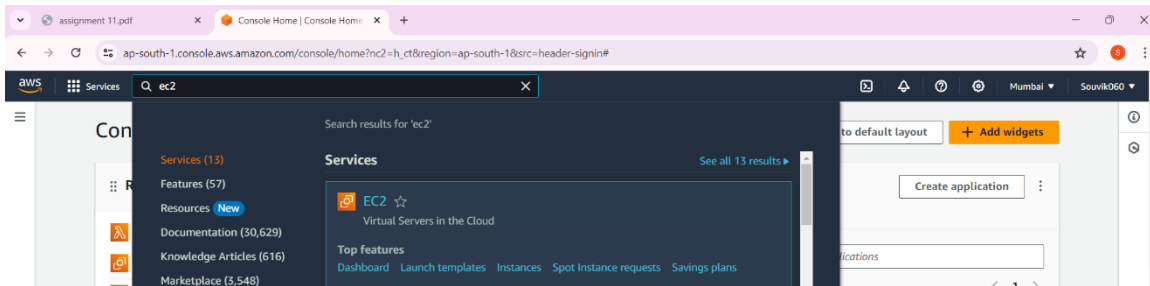


Assignment 11

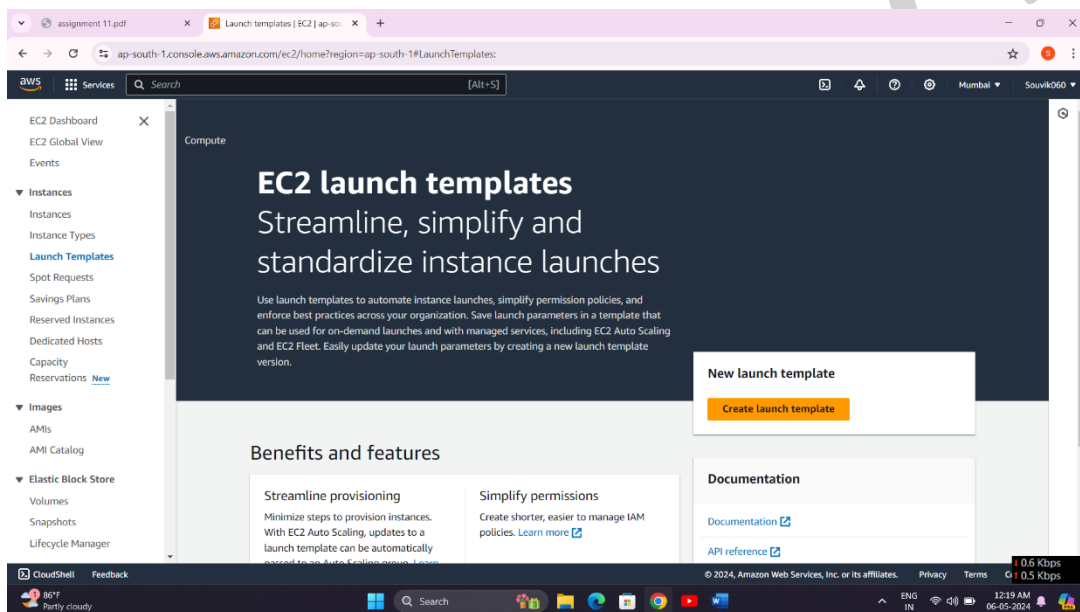
Problem Statement: Build scaling plans in AWS that balance load on different EC2 instances.

The steps are as follows:

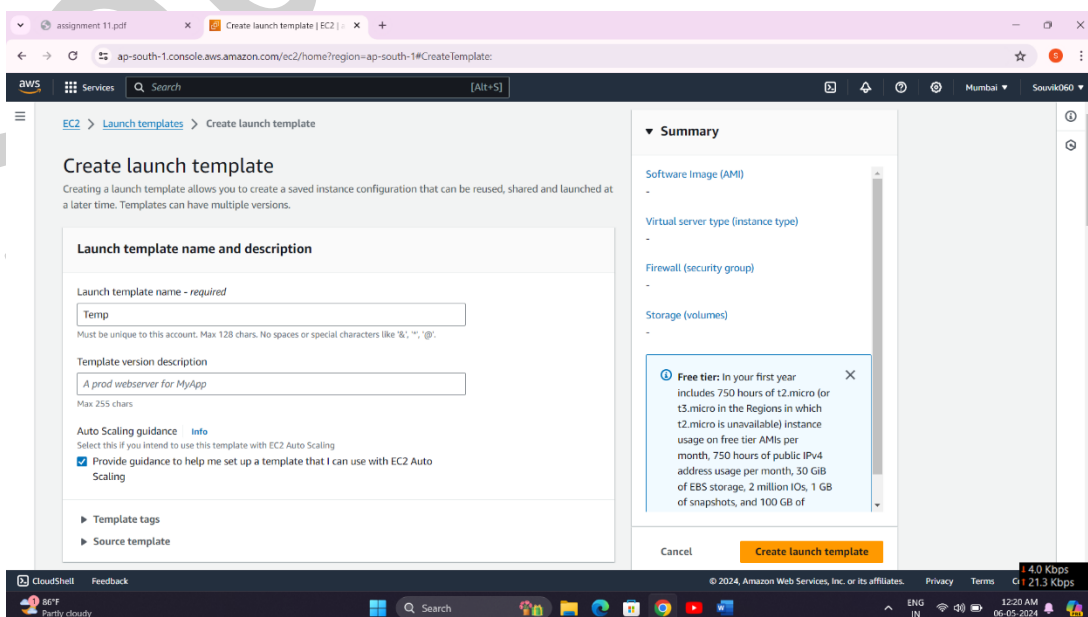
1. Please access AWS and look up EC2, then select the initial option displayed.



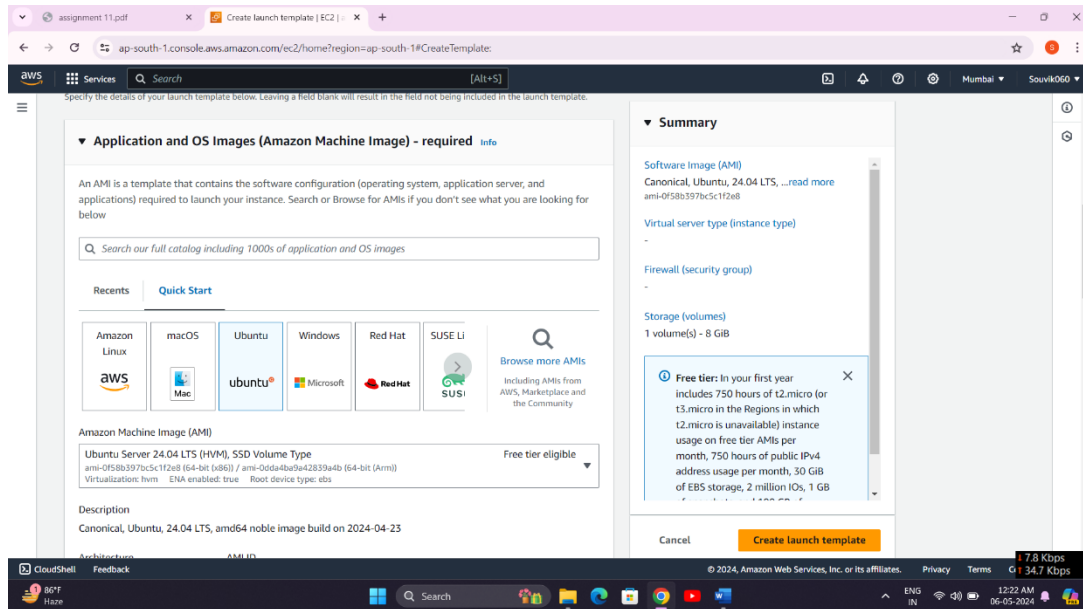
2. Next, select "Launch Template" from the menu on the left side. Now click on the "Create Launch Template" option.



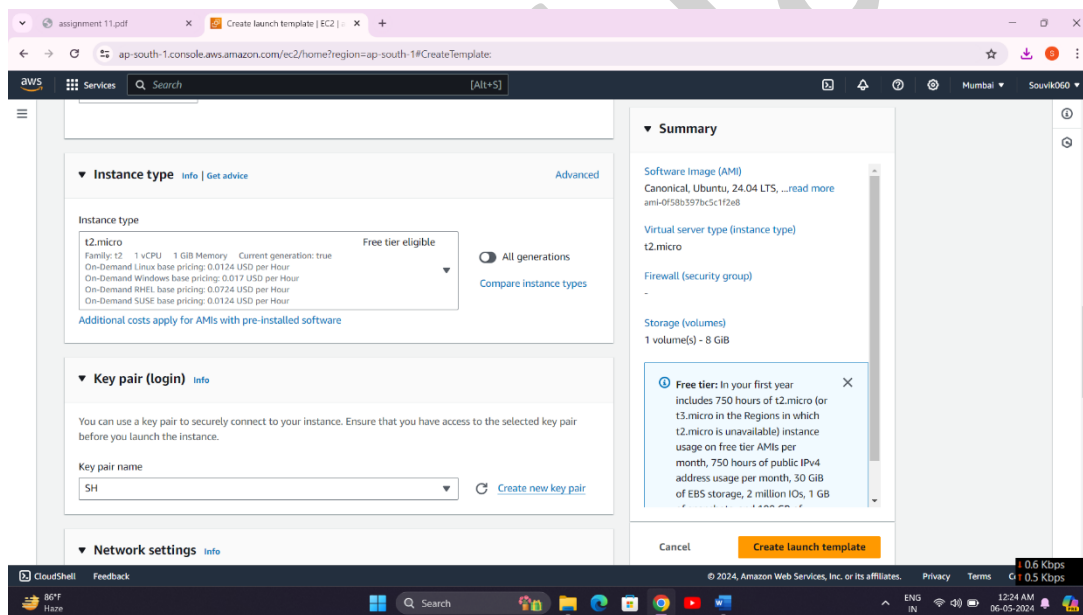
3. Enter a template name, such as "Temp" and check the box for autoscaling options.



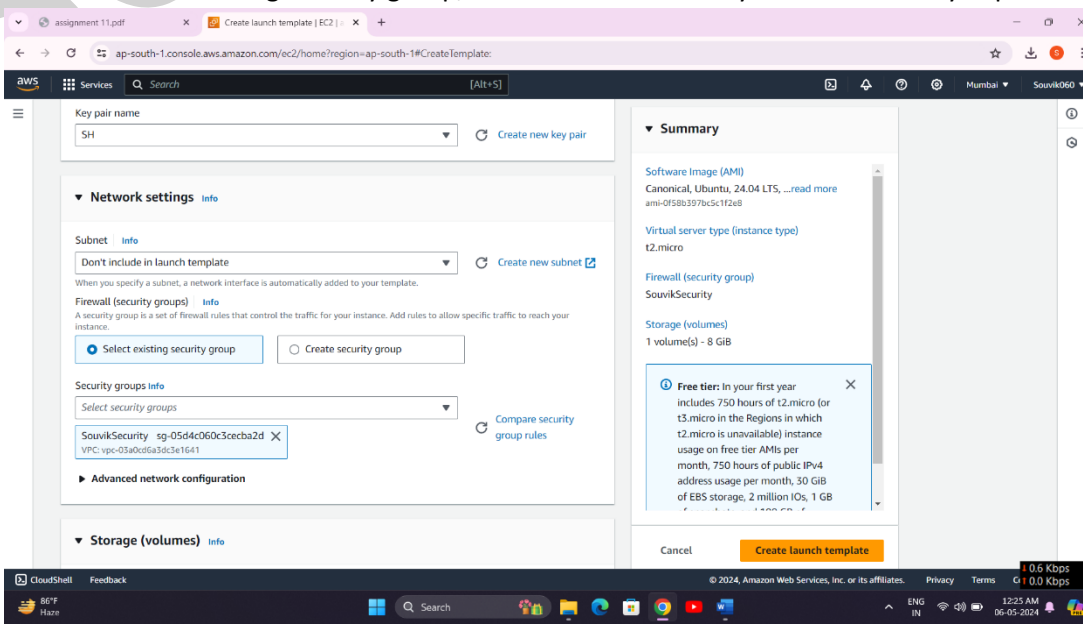
4. Navigate to "Quick Start" and choose "Ubuntu" from the list of available AMIs.



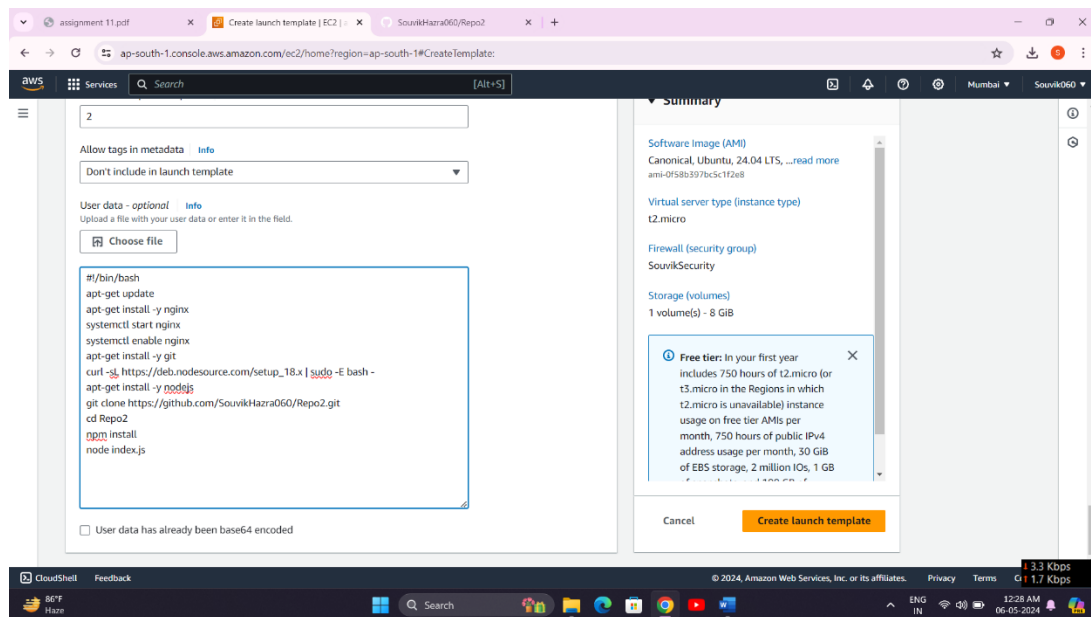
5. Next, select the instance type—either t2.micro or t3.micro, both of which are free tier eligible. Then choose either an existing key pair or create a new one if it doesn't exist. For example, use "SH" as the key pair name.



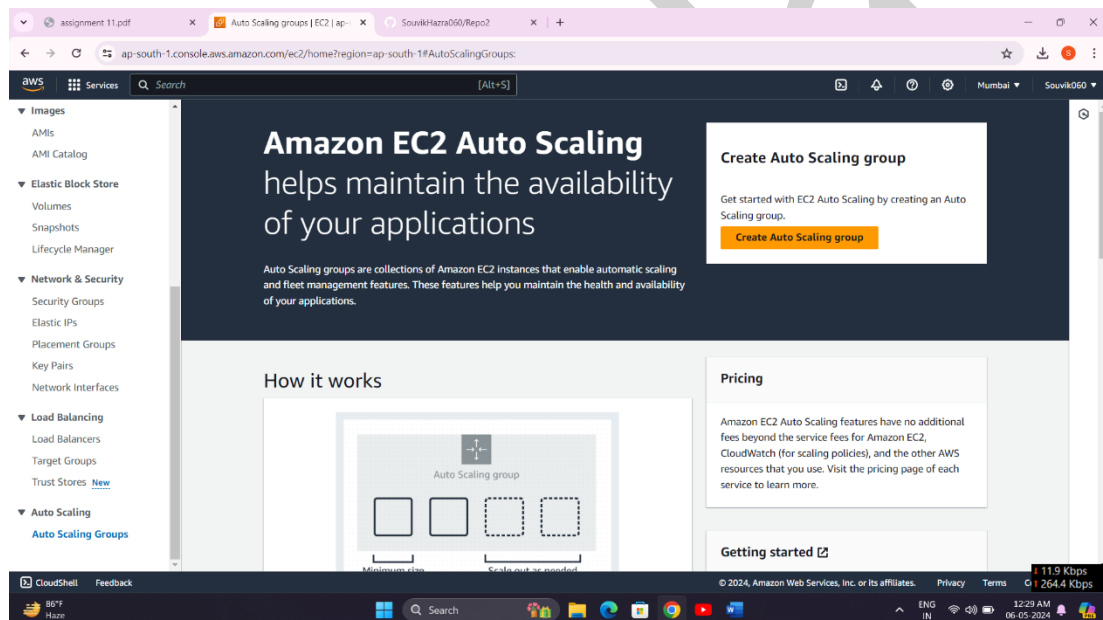
6. Select an existing security group, such as "SouvikSecurity" which is already in place.



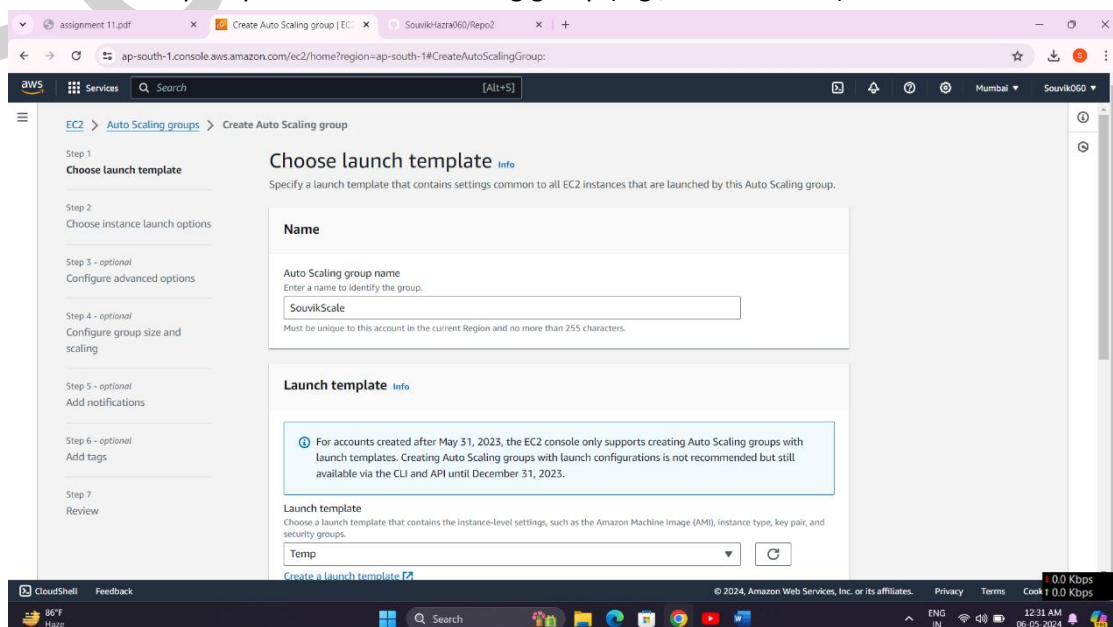
7. Expand the "Advanced details" section, navigate to "User data", and input the provided code. Then proceed to click on "Create launch template".



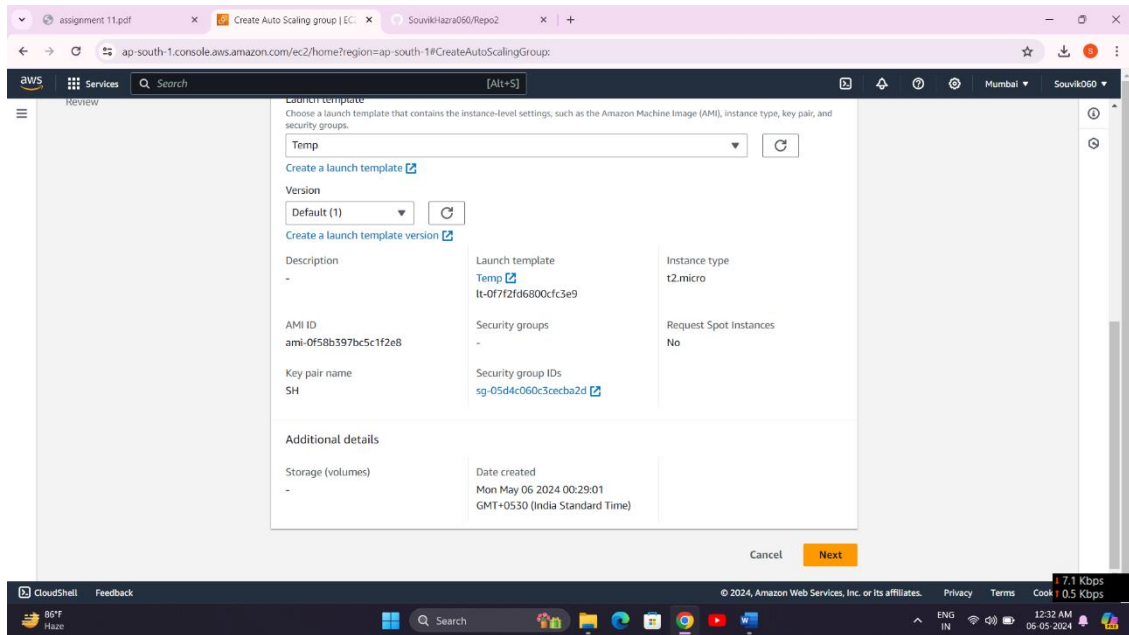
8. Once the launch template has been successfully created, navigate to the left pane and search for "Auto Scaling Groups". Then, select "Create Auto Scaling Group".



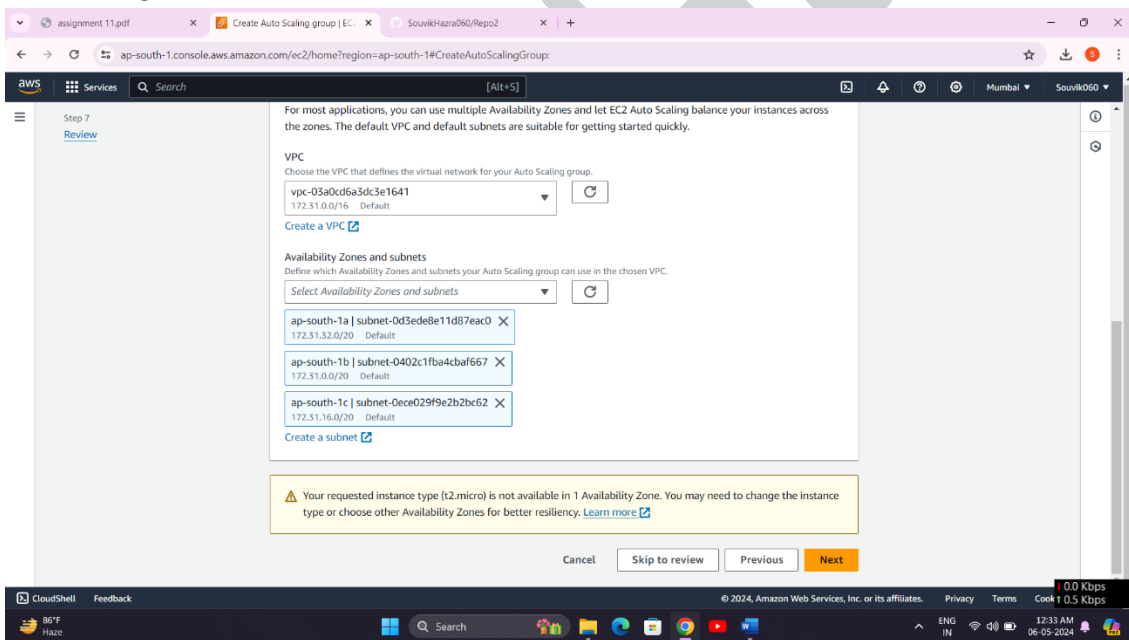
9. Please specify a name for the scaling group (e.g., "SouvikScale").



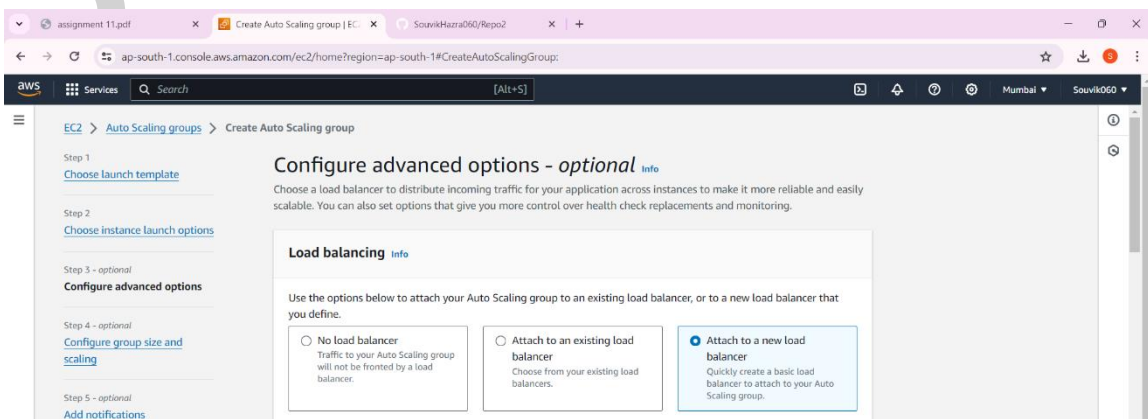
10. Select the template that was created in the preceding steps (e.g. Temp). Proceed to click on "Next".



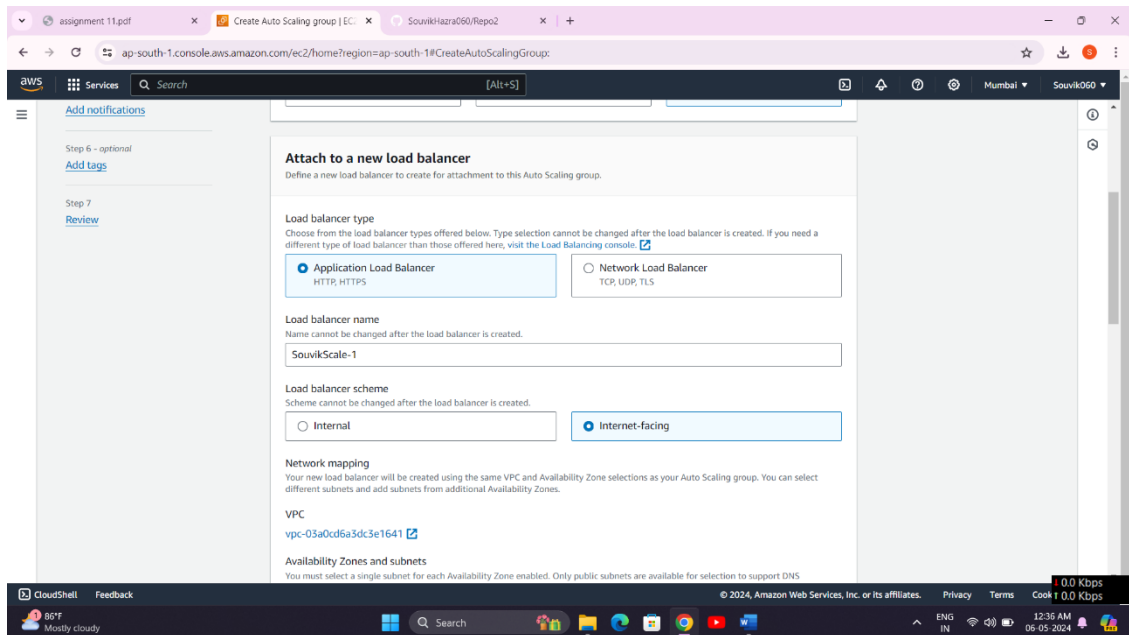
11. In the following step, choose all available availability zones and subnets, then proceed by clicking "Next".



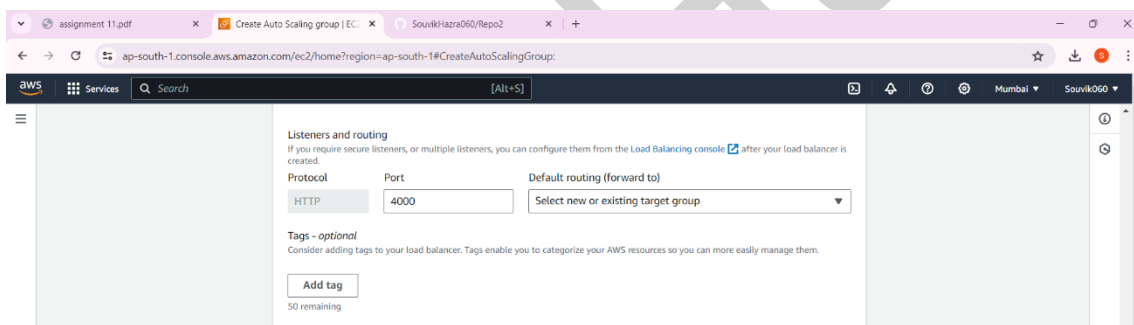
12. In the subsequent step, begin by selecting "Attach to a new load balancer".



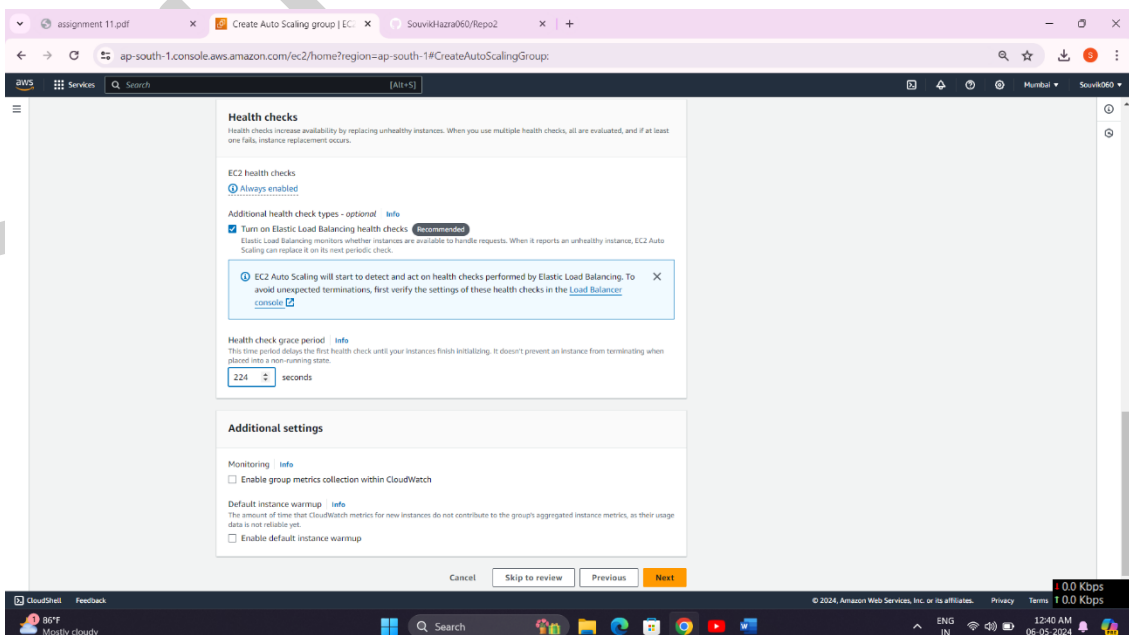
13. Select "Application Load Balancer" as the load balancer type and "Internet-facing" as the load balancer scheme.



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



15. Enable the checkbox to activate health checks and specify a "health check grace period", set here to 224 seconds. Without any further modifications, proceed to click on "Next".



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

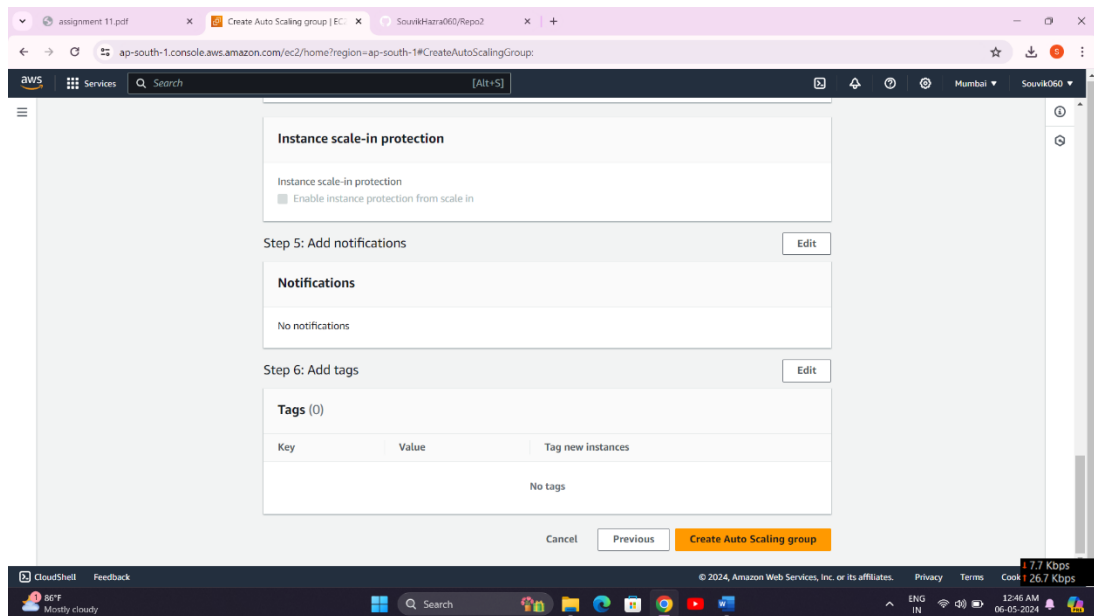
The screenshot shows the AWS Management Console interface for creating an Auto Scaling group. The browser tabs include 'assignment 11.pdf', 'Create Auto Scaling group | EC2', and 'SouvikHaza060/Repo2'. The URL is 'ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup:'. The left sidebar shows the navigation menu with steps 1 through 7. The main content area is titled 'Step 2: Configure group size and scaling'. It includes a 'Group size' section with a 'Desired capacity type' dropdown set to 'Units (number of instances)' and a 'Desired capacity' input field set to '2'. Below this is a 'Scaling' section with 'Scaling limits' input fields for 'Min desired capacity' (set to '2') and 'Max desired capacity' (set to '3'). The bottom of the screen shows a Windows taskbar with various application icons and system tray information.

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

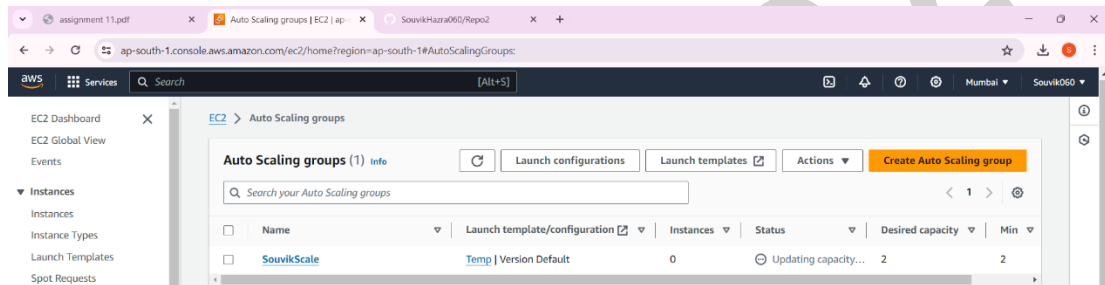
The screenshot shows the AWS Management Console interface for creating an Auto Scaling group, specifically the 'Automatic scaling - optional' section. The browser tabs and URL are the same as in the previous screenshot. The left sidebar shows the navigation menu with steps 1 through 7. The main content area is titled 'Step 3: Configure advanced options'. It includes a 'Automatic scaling - optional' section with a 'Target tracking scaling policy' selected. Below this, there is a 'Scaling policy name' input field set to 'Target Tracking Policy', a 'Metric type' dropdown set to 'Average CPU utilization', a 'Target value' input field set to '50', and an 'Instance warmup' input field set to '240' seconds. The bottom of the screen shows a Windows taskbar with various application icons and system tray information.

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

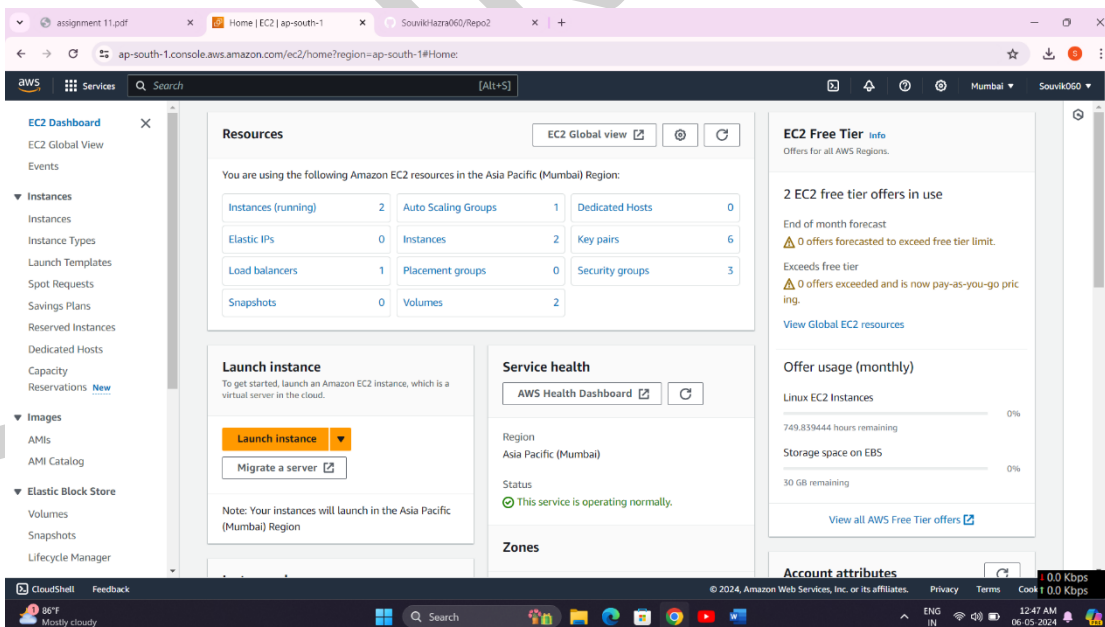
The screenshot shows the AWS Management Console interface for creating an Auto Scaling group, specifically the 'Add notifications - optional' section. The browser tabs and URL are the same as in the previous screenshots. The left sidebar shows the navigation menu with steps 1 through 3. The main content area is titled 'Step 3: Add notifications - optional'. It includes a section for 'Add notifications' with a button labeled 'Add notification'. At the bottom of the page, there are four buttons: 'Cancel', 'Skip to review', 'Previous', and 'Next'. The 'Next' button is highlighted in orange. The bottom of the screen shows a Windows taskbar with various application icons and system tray information.



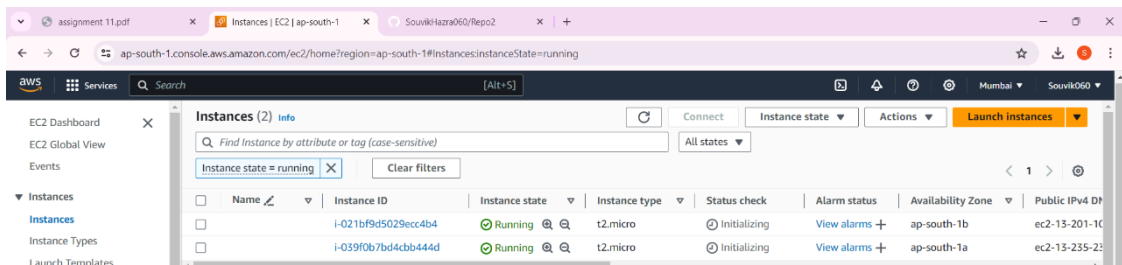
19. Our Auto Scaling Group is created successfully.



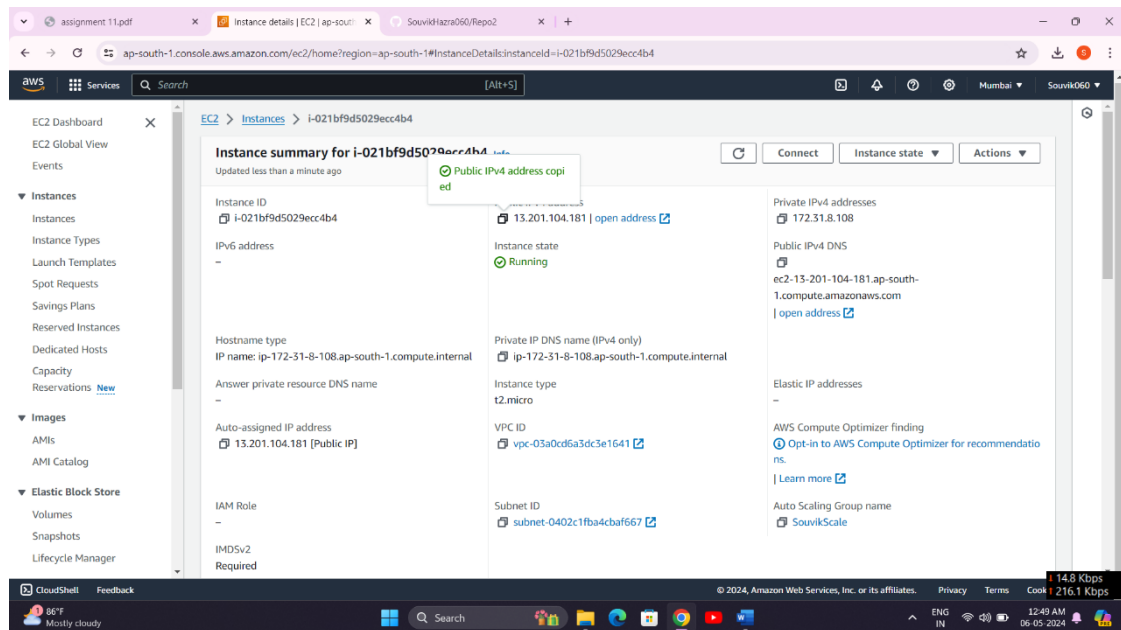
20. After the auto-scaling group is created, return to the EC2 dashboard and navigate to the "Instances" section for running instances.



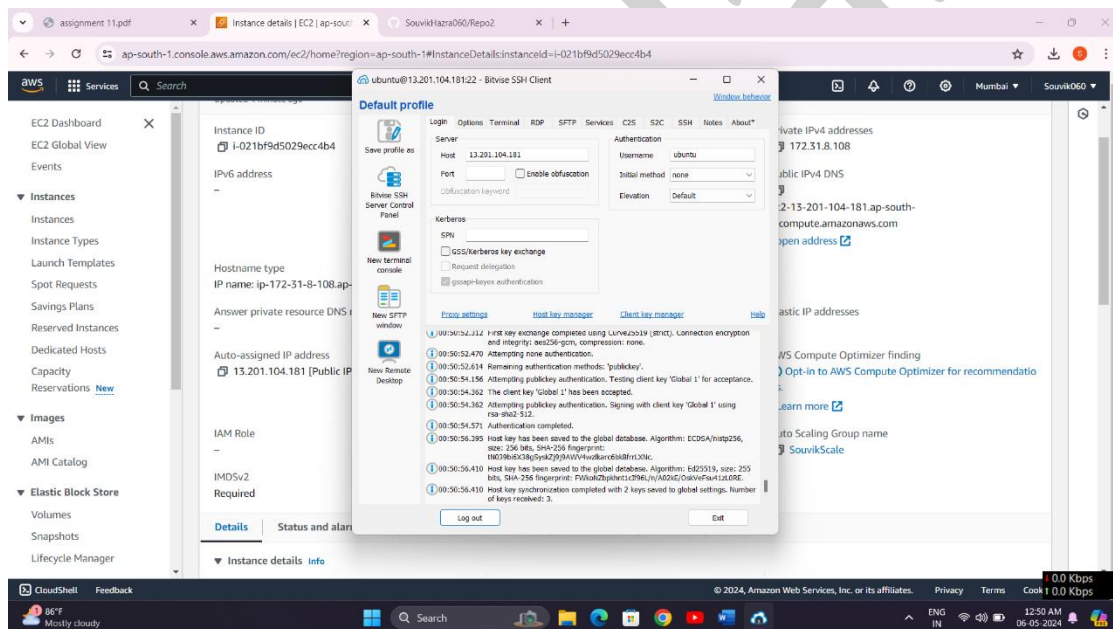
21. Since the minimum capacity chosen was 2, two instances have been created.



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



23. Launch the Bitwise SSH Client and log in using the IPv4 address we copied earlier.



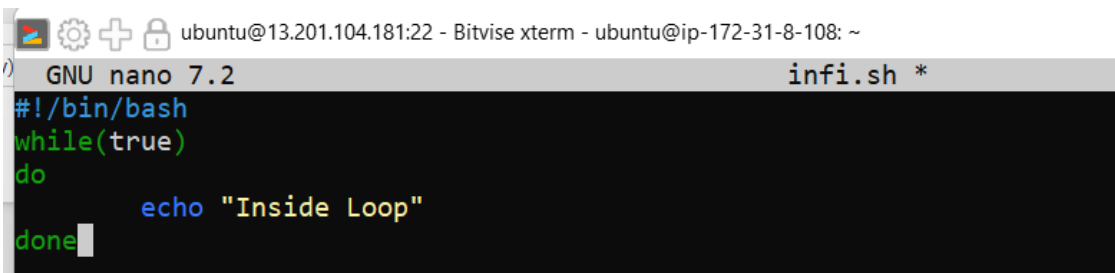
24. After successfully logging in, open a new terminal console from the left pane.

And now write the commands in the terminal as follows:

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

```
ubuntu@ip-172-31-8-108:~$ sudo nano infi.sh
```

25. →Write this following code in the file "infi.sh" to run an infinite loop. Press CTRL+X,Y then enter for save the file



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

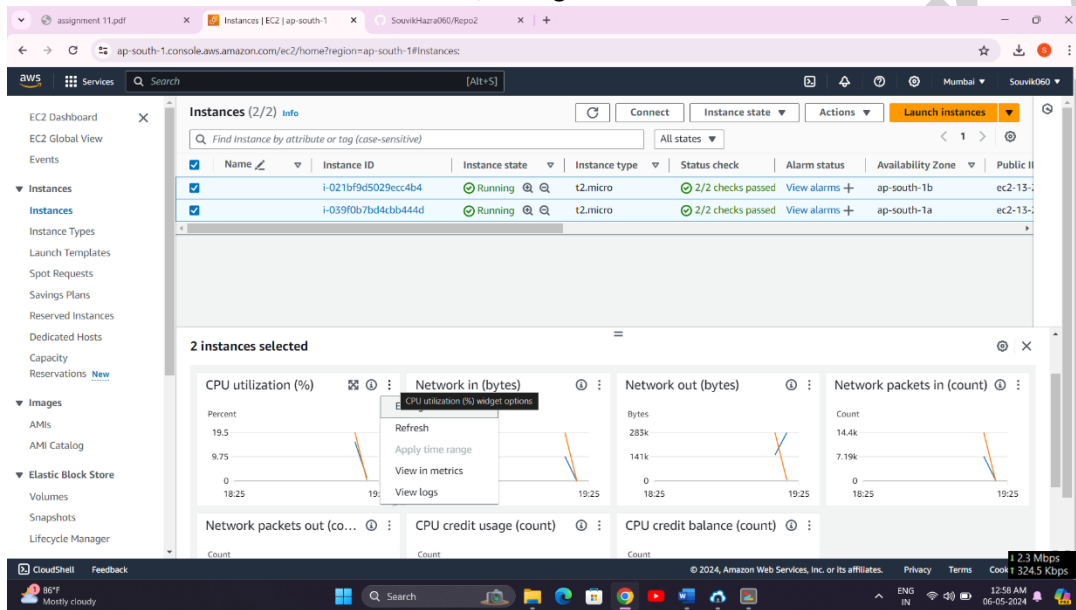

```
ubuntu@ip-172-31-8-108:~$ sudo chmod 777 infi.sh
```

27. →sh infi.sh (Run the .sh file infinite time)

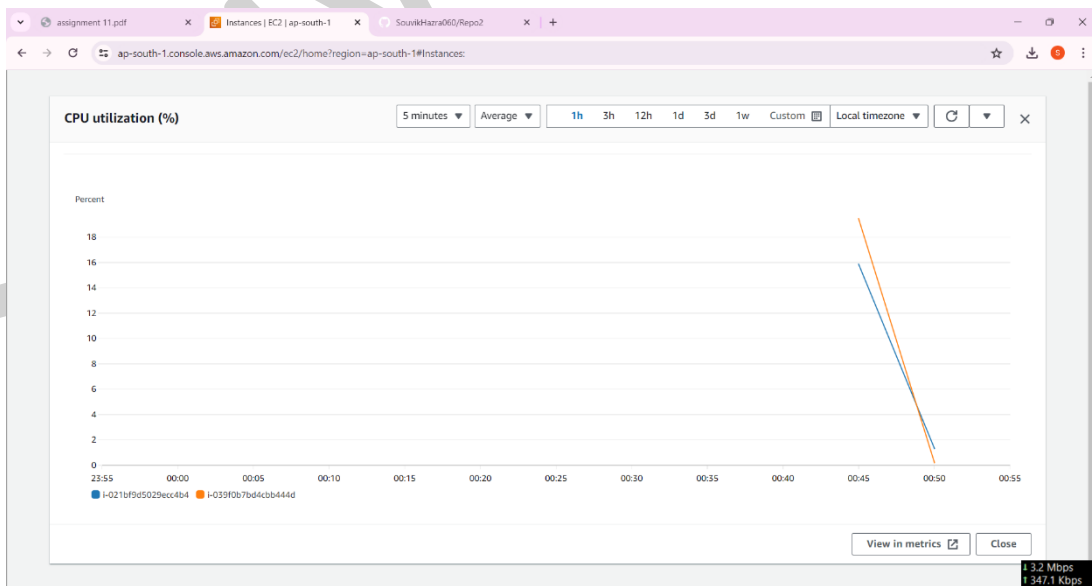
```
ubuntu@ip-172-31-8-108:~$ sh infi.sh
```

[illegible]

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



29. From the panel above, select "Local timezone." The graph displays CPU utilization for both instances.



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

