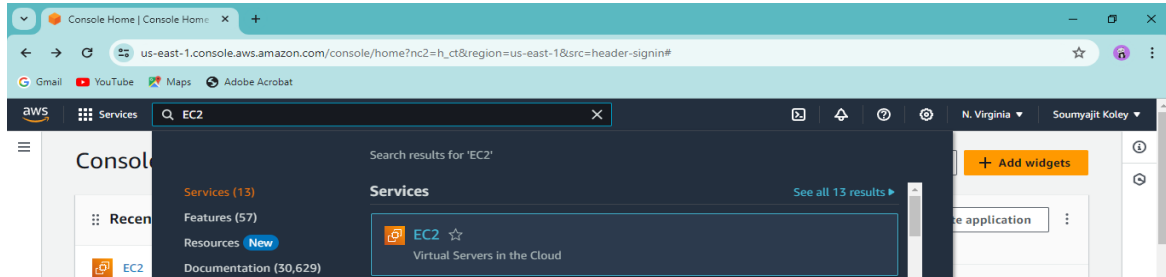


Assignment 11

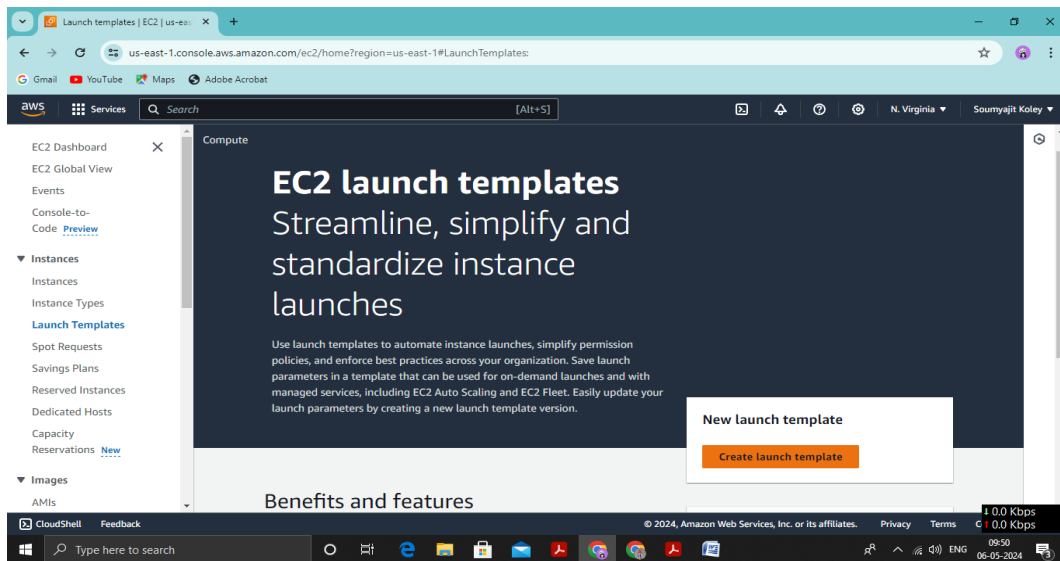
Problem Statement: Build scaling plans in AWS that balance load on different EC2 instances.

The steps are as follows:

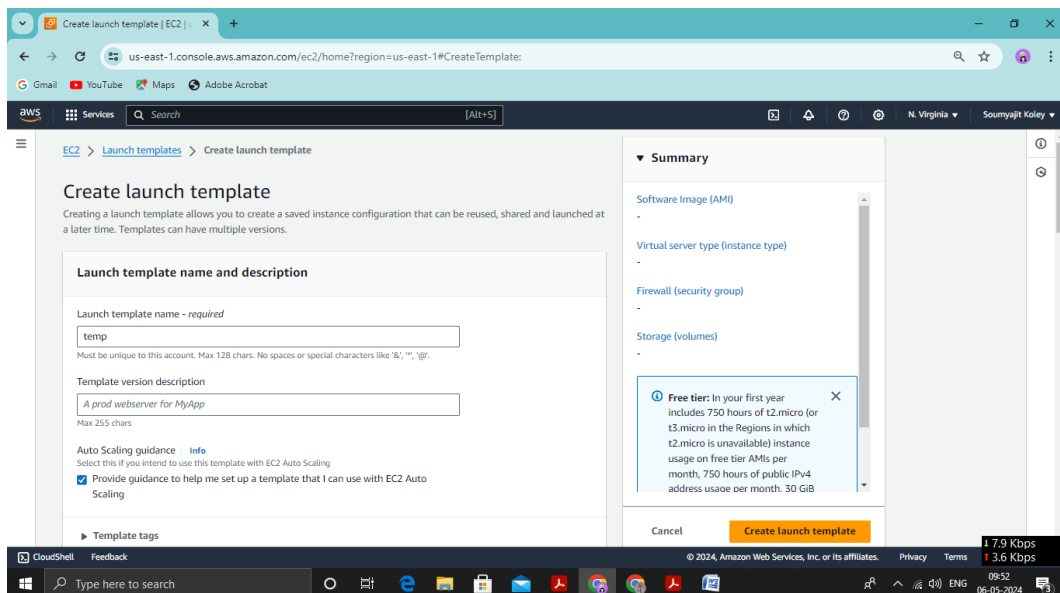
1. Please access AWS and look up EC2, then select the initial option displayed.



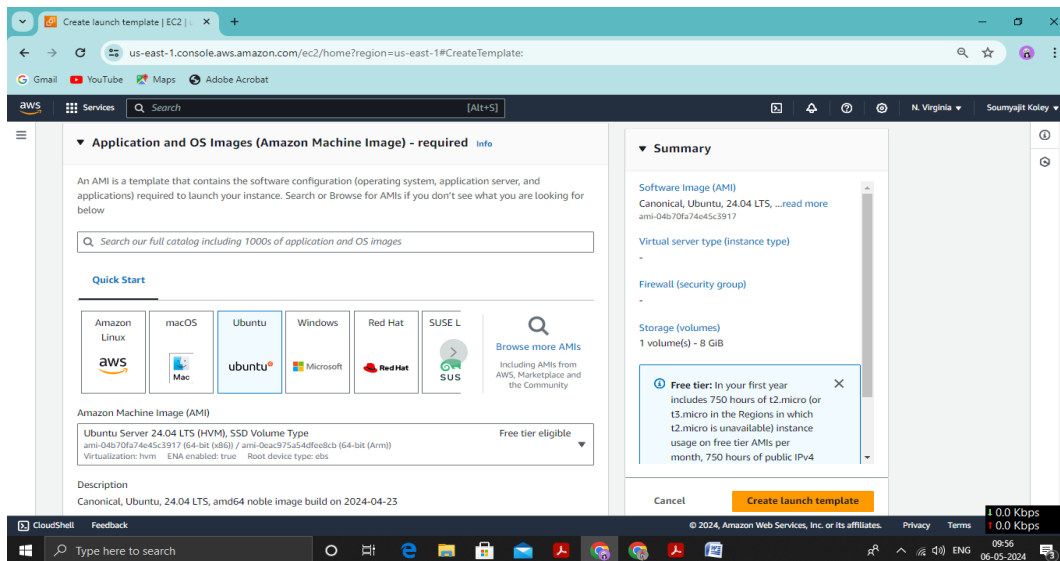
2. Next, select "Launch Template" from the menu on the left side. Now click on the "Create Launch Template" option.



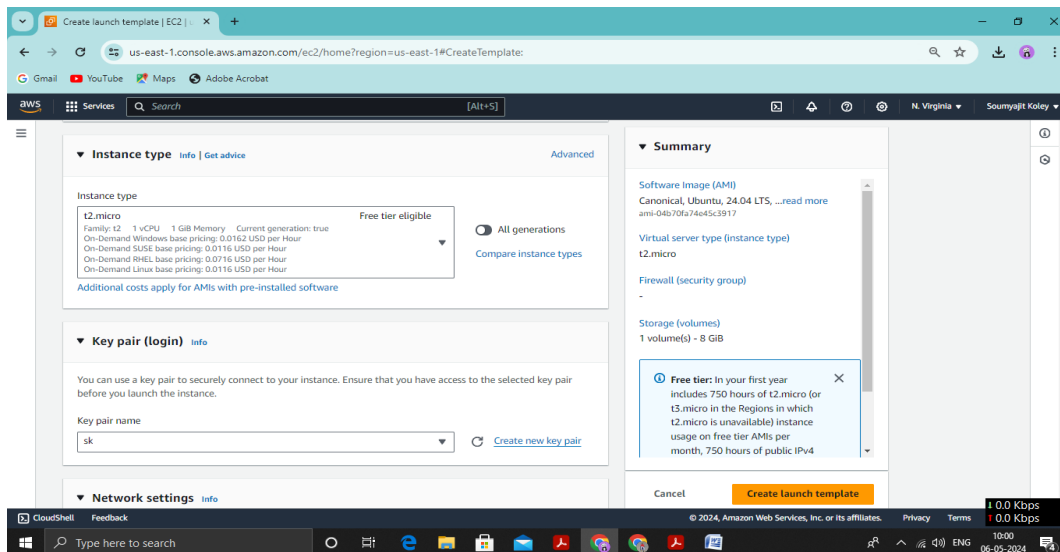
3. Enter a template name, such as "Temp" and check the box for autoscaling options.



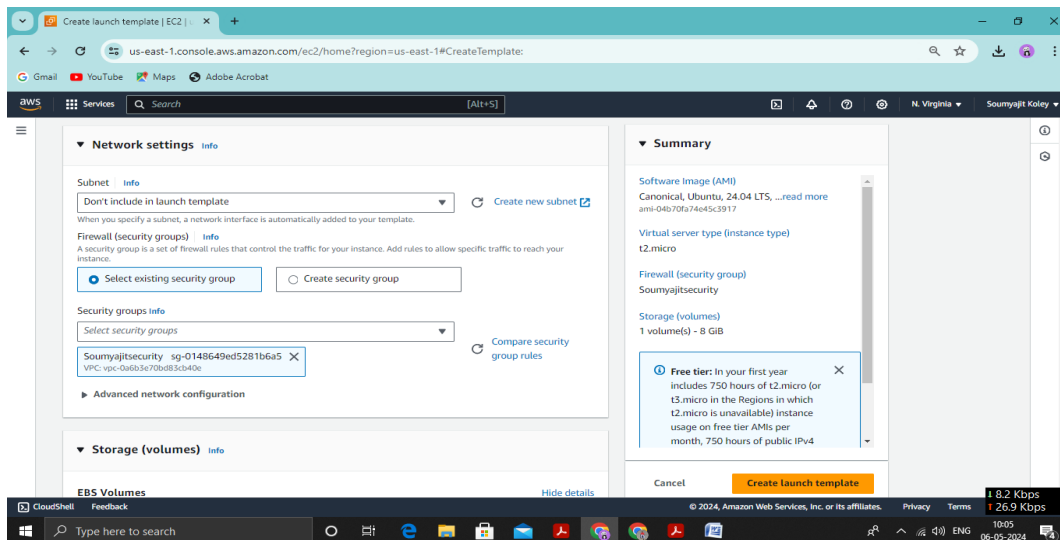
4. Navigate to "Quick Start" and choose "Ubuntu" from the list of available AMIs.



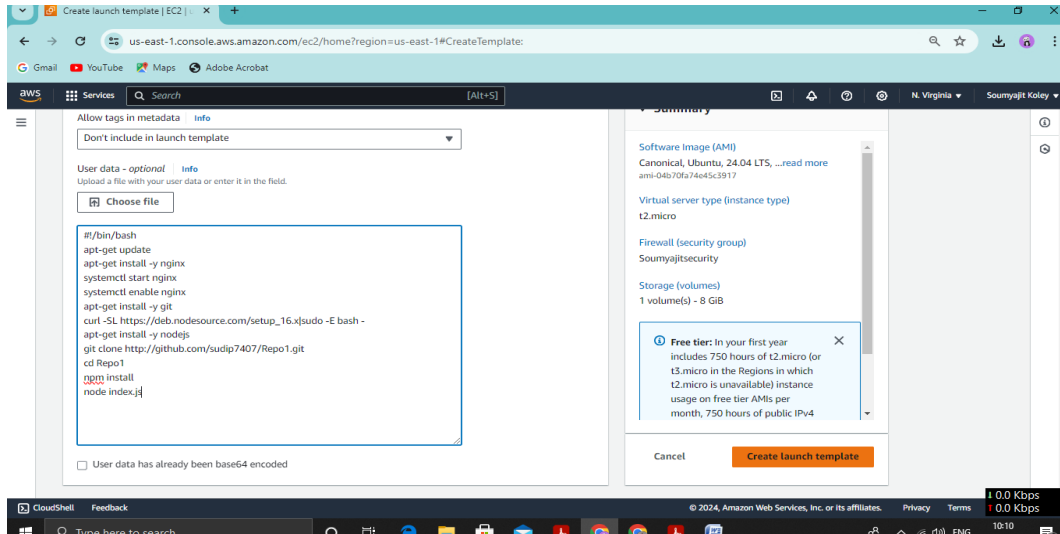
5. Next, select the instance type—either t2.micro or t3.micro, both of which are free tier eligible. Then choose either an existing key pair or create a new one if it doesn't exist. For example, use "SK" as the key pair name.



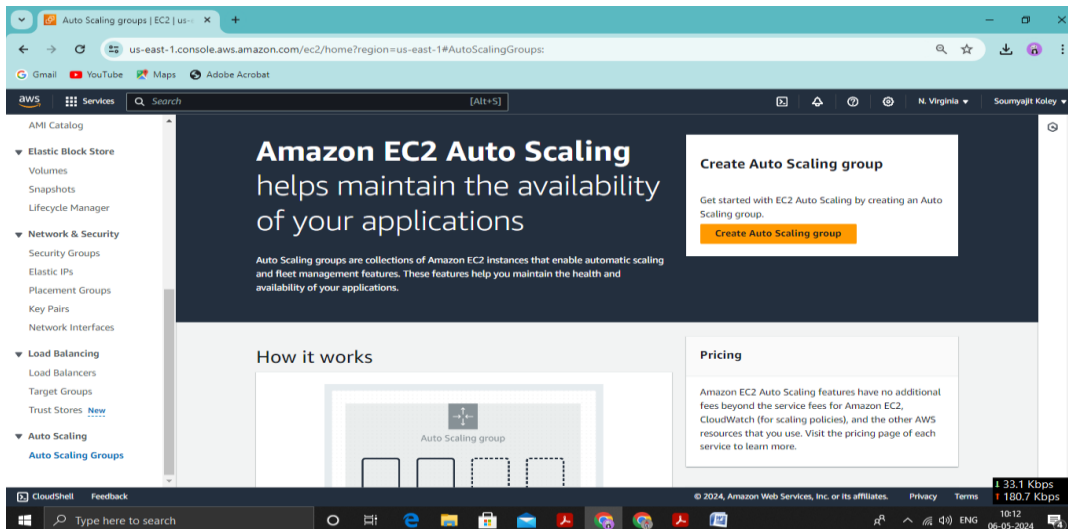
6. Select an existing security group, such as "Soumyajitsecurity" which is already in place.



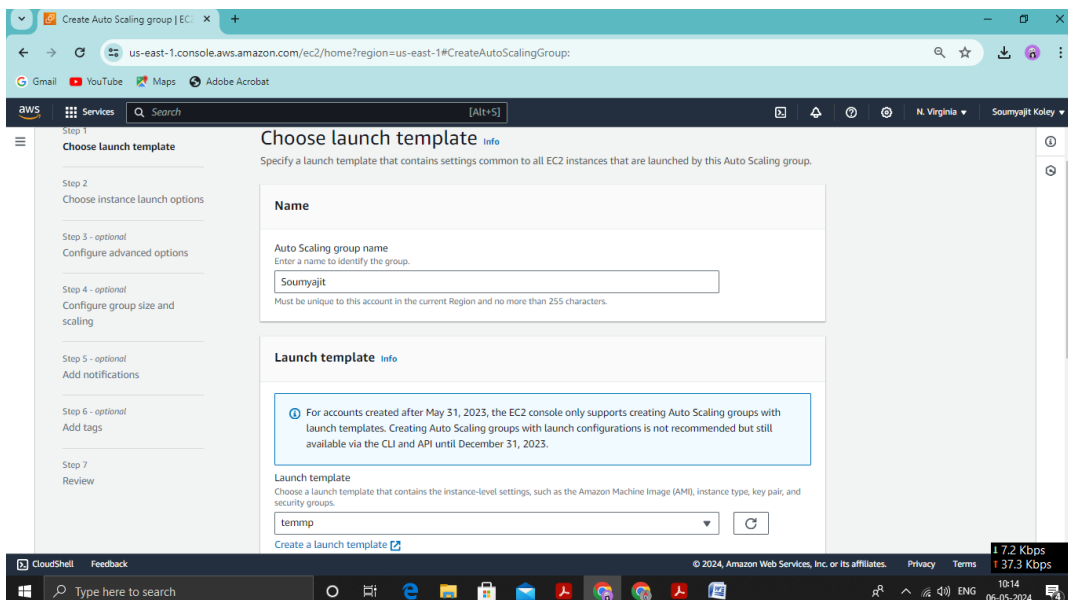
7. Expand the "Advanced details" section, navigate to "User data", and input the provided code. Then proceed to click on "Create launch template".



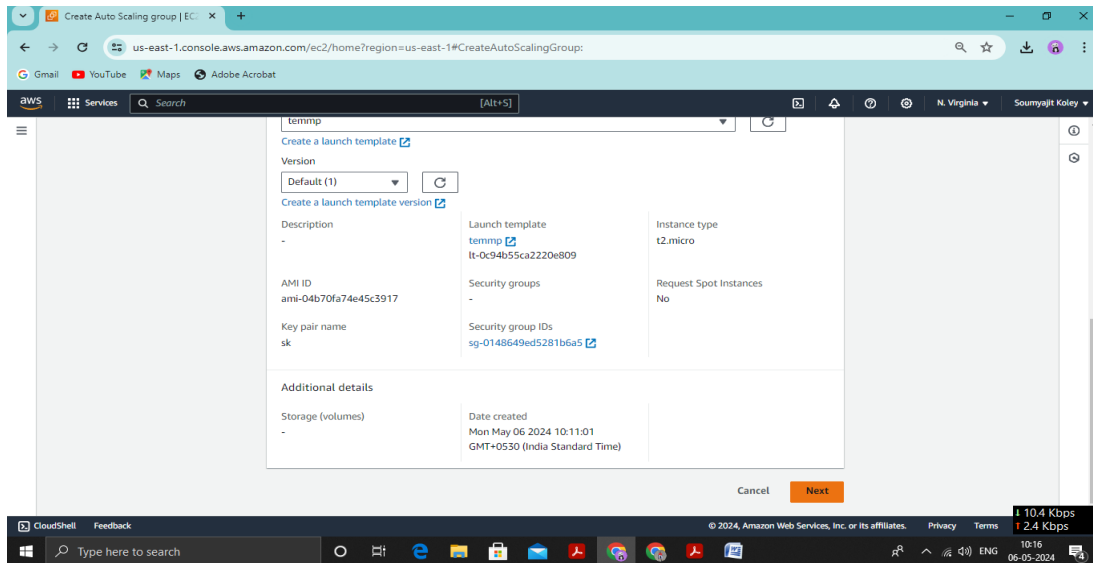
8. Once the launch template has been successfully created, navigate to the left pane and search for "Auto Scaling Groups". Then, select "Create Auto Scaling Group".



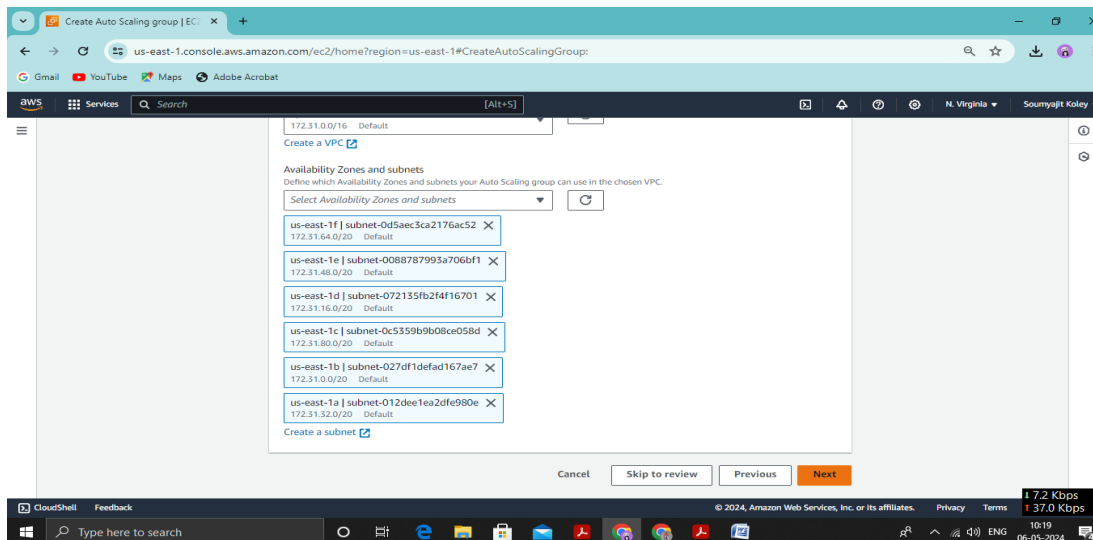
9. Please specify a name for the scaling group (e.g., "Soumyajit").



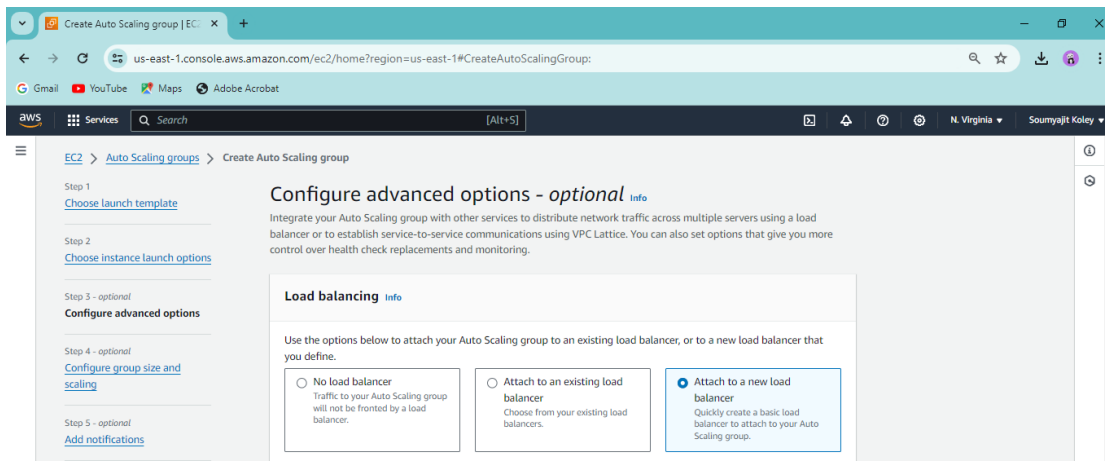
10. Select the template that was created in the preceding steps (e.g. Temp). Proceed to click on "Next".



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.

The screenshot shows the 'Attach to a new load balancer' step in the AWS Management Console. The page title is 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:'. The left sidebar shows 'Step 6 - optional' and 'Step 7 - Review'. The main content area is titled 'Attach to a new load balancer' and includes the following sections:

- Load balancer type:** Two radio buttons are present. 'Application Load Balancer' (HTTP, HTTPS) is selected. 'Network Load Balancer' (TCP, UDP, TLS) is unselected.
- Load balancer name:** A text input field contains 'Soumyajit-1'.
- Load balancer scheme:** Two radio buttons are present. 'Internet-facing' is selected. 'Internal' is unselected.
- Network mapping:** A text input field contains 'vpc-0a6b3e70bd83cb40e'.

The bottom of the screen shows a Windows taskbar with various application icons and a system clock displaying '10:22 06-05-2024'.

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

The screenshot shows the 'Listeners and routing' step in the AWS Management Console. The page title is 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:'. The left sidebar shows 'Step 6 - optional' and 'Step 7 - Review'. The main content area is titled 'Listeners and routing' and includes the following sections:

- Protocol:** A dropdown menu is set to 'HTTP'.
- Port:** A text input field contains '4000'.
- Default routing (forward to):** A dropdown menu is set to 'Select new or existing target group'.
- Tags - optional:** A section with an 'Add tag' button and a note '50 remaining'.

The bottom of the screen shows a Windows taskbar with various application icons and a system clock displaying '10:22 06-05-2024'.

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".

The screenshot shows the 'Health checks' step in the AWS Management Console. The page title is 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:'. The left sidebar shows 'Step 6 - optional' and 'Step 7 - Review'. The main content area is titled 'Health checks' and includes the following sections:

- EC2 health checks:** A section with a note 'Always enabled'.
- Additional health check types - optional:** A section with a note 'Recommended' and a checkbox 'Turn on Elastic Load Balancing health checks' which is checked.
- Health check grace period:** A text input field contains '224' seconds.
- Additional settings:** A section with a note 'info' and a checkbox 'Enable group metrics collection within CloudWatch' which is unchecked.

The bottom of the screen shows a Windows taskbar with various application icons and a system clock displaying '10:22 06-05-2024'.

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

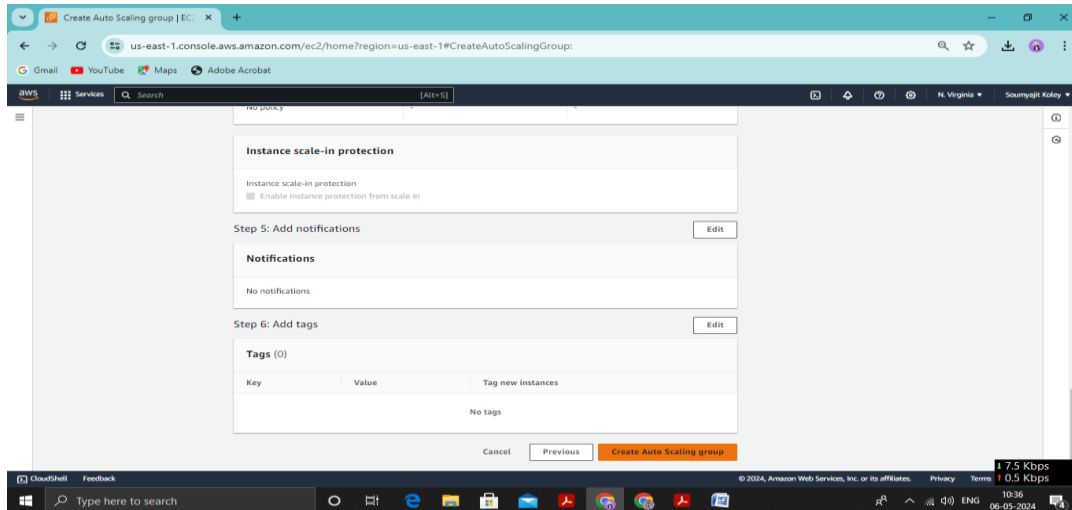
The screenshot shows the 'Create Auto Scaling group' wizard in the AWS console, specifically Step 4: 'Configure group size and scaling'. The left sidebar lists steps from Step 2 to Step 7. The main content area is divided into three sections: 'Group size', 'Desired capacity type', and 'Scaling info'. In the 'Group size' section, 'Units (number of instances)' is set to 2. In the 'Desired capacity type' section, 'Desired capacity' is set to 2. In the 'Scaling info' section, 'Min desired capacity' is set to 2 and 'Max desired capacity' is set to 3. The 'Automatic scaling - optional' section is visible at the bottom.

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.

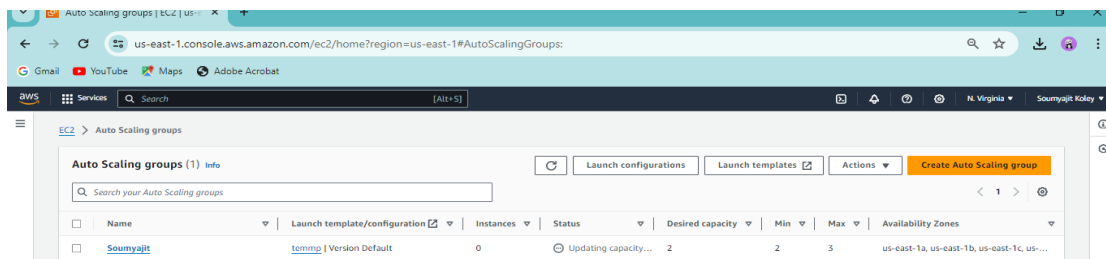
The screenshot shows the 'Create Auto Scaling group' wizard in the AWS console, specifically Step 5: 'Add notifications'. The left sidebar lists steps from Step 2 to Step 7. The main content area is divided into two sections: 'Automatic scaling - optional' and 'Instance maintenance policy'. In the 'Automatic scaling - optional' section, the 'Target tracking scaling policy' is selected. The 'Scaling policy name' is 'Target Tracking Policy', the 'Metric type' is 'Average CPU utilization', the 'Target value' is 50, and the 'Instance warmup' is 240 seconds. The 'Instance maintenance policy' section is visible at the bottom.

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

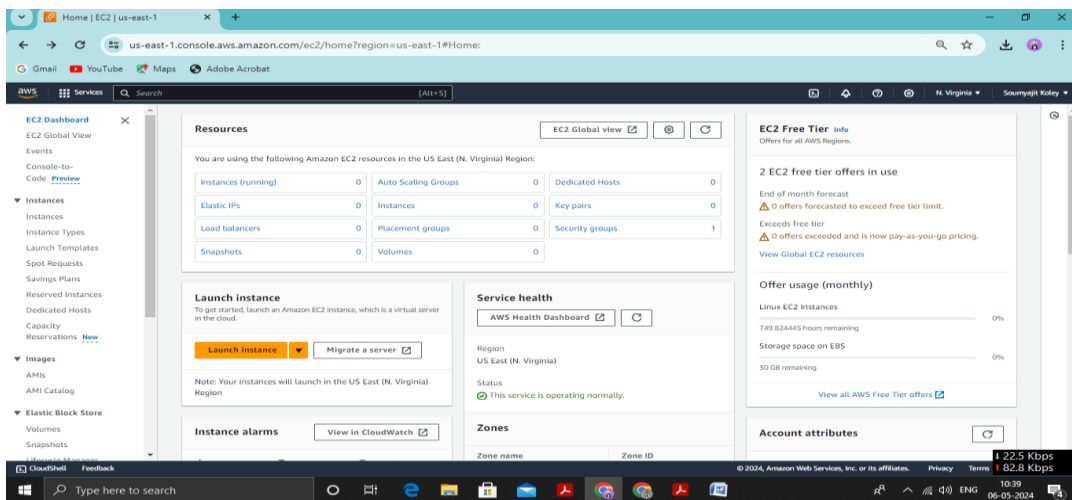
The screenshot shows the 'Create Auto Scaling group' wizard in the AWS console, specifically Step 6: 'Review'. The left sidebar lists steps from Step 1 to Step 5. The main content area is divided into two sections: 'Add notifications - optional' and 'Review'. In the 'Add notifications - optional' section, there is a button to 'Add notification'. In the 'Review' section, there are buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next'. The 'Next' button is highlighted in orange.



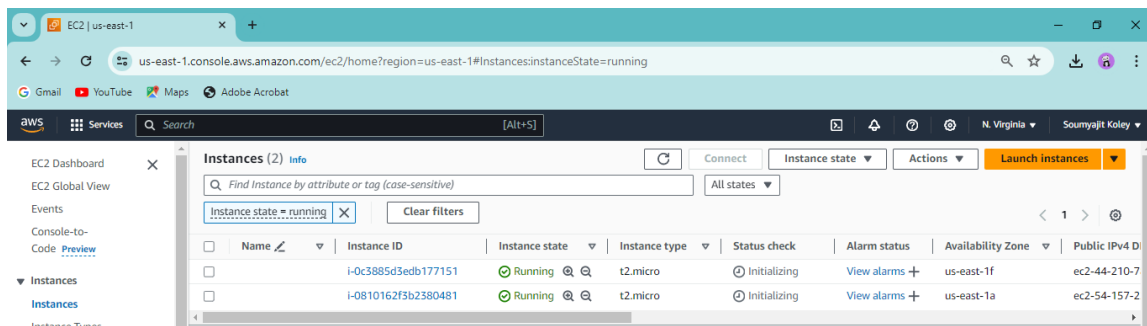
19. Our Auto Scaling Group is created successfully.



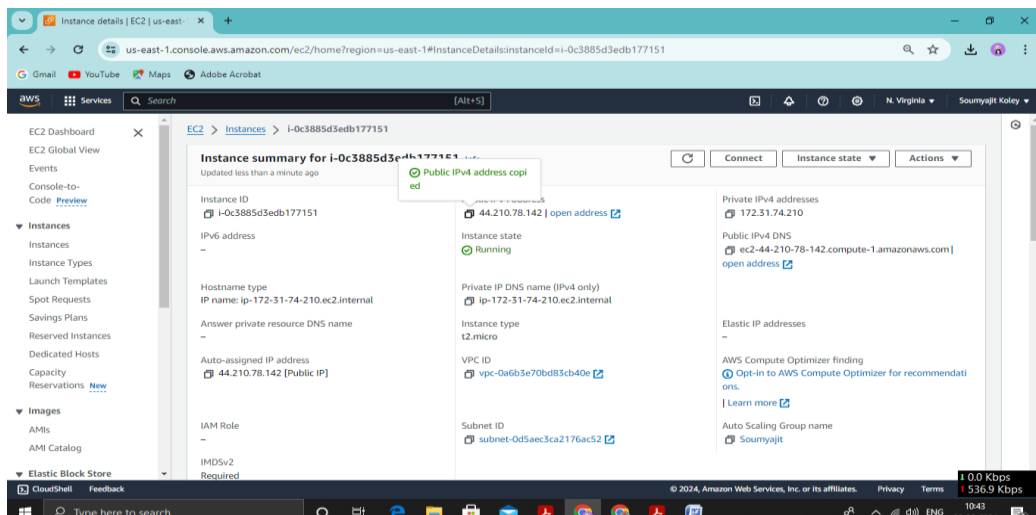
20. After the auto-scaling group is created, return to the EC2 dashboard and navigate to the "Instances" section for running instances.



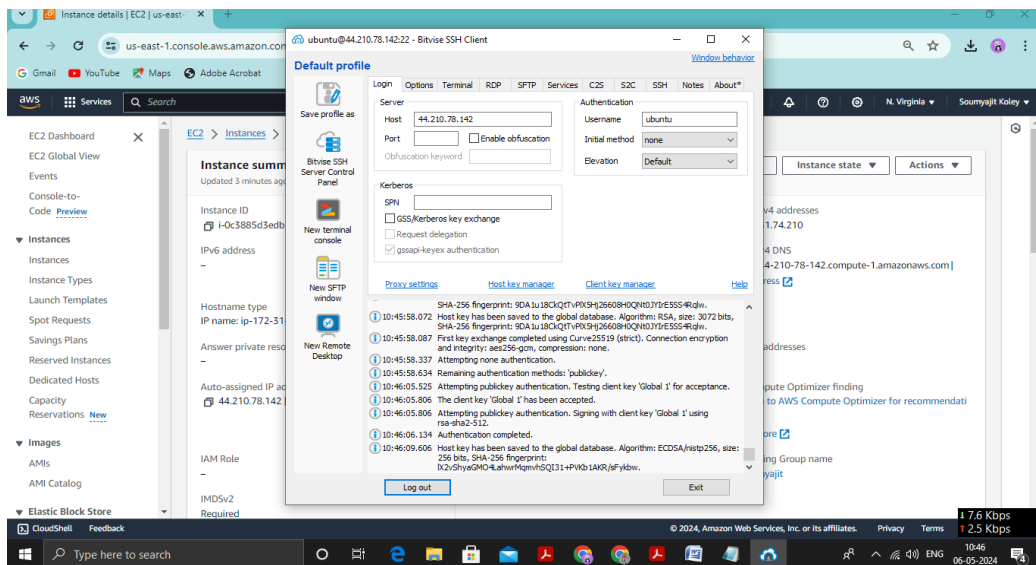
21. Since the minimum capacity chosen was 2, two instances have been created.



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-172-31-74-210:~$ sudo nano infi.sh
ubuntu@ip-172-31-74-210:~$
```

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

```
GNU nano 7.2 infi.sh *
#!/bin/bash
while(true)
do
echo "Inside loop"
done
```

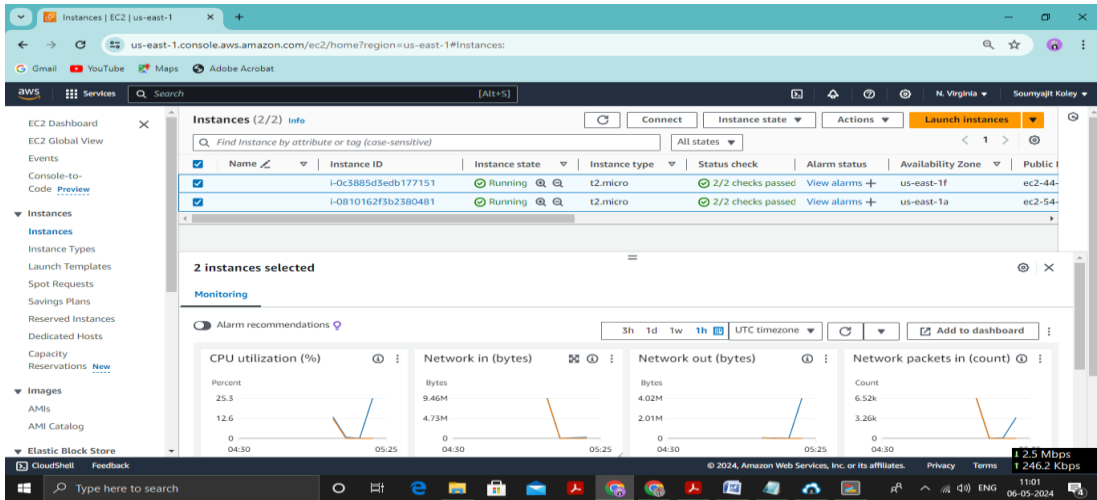
26. →sudo chmod 777 infi.sh (to provide all permission to the file)

```
ubuntu@ip-172-31-74-210:~$ sudo chmod 777 infi.sh
ubuntu@ip-172-31-74-210:~$
```

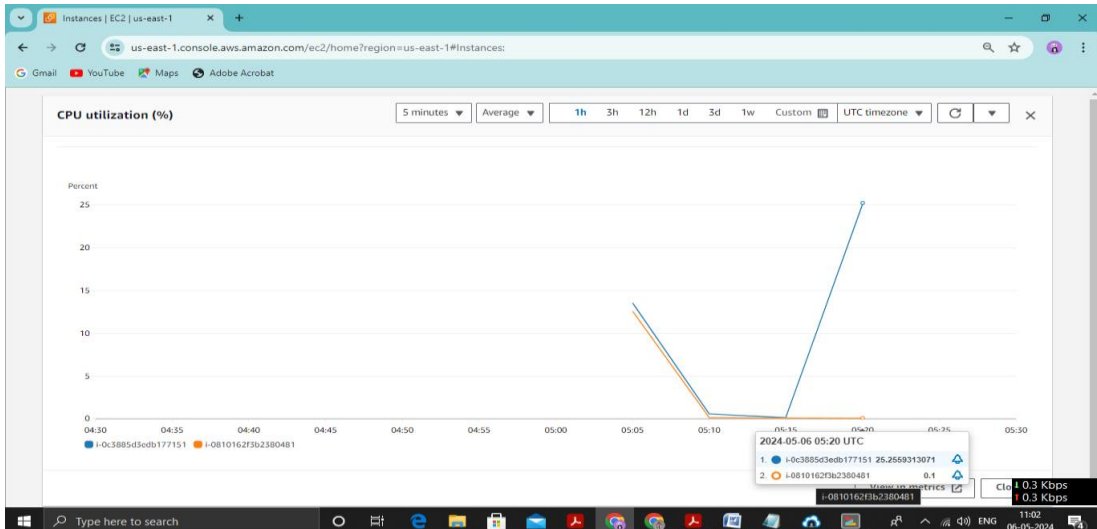

27. →sh infi.sh (Run the .sh file infinite time)

```
ubuntu@44.210.78.142:22 - Bitvise xterm - ubuntu@ip-172-31-74-210: ~
Inside loop
Inside loop
Inside loop
Inside loop
Inside loop
Inside loop
Inside loop
Inside loop
Inside loop
Inside loop
Inside loop
```

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.

