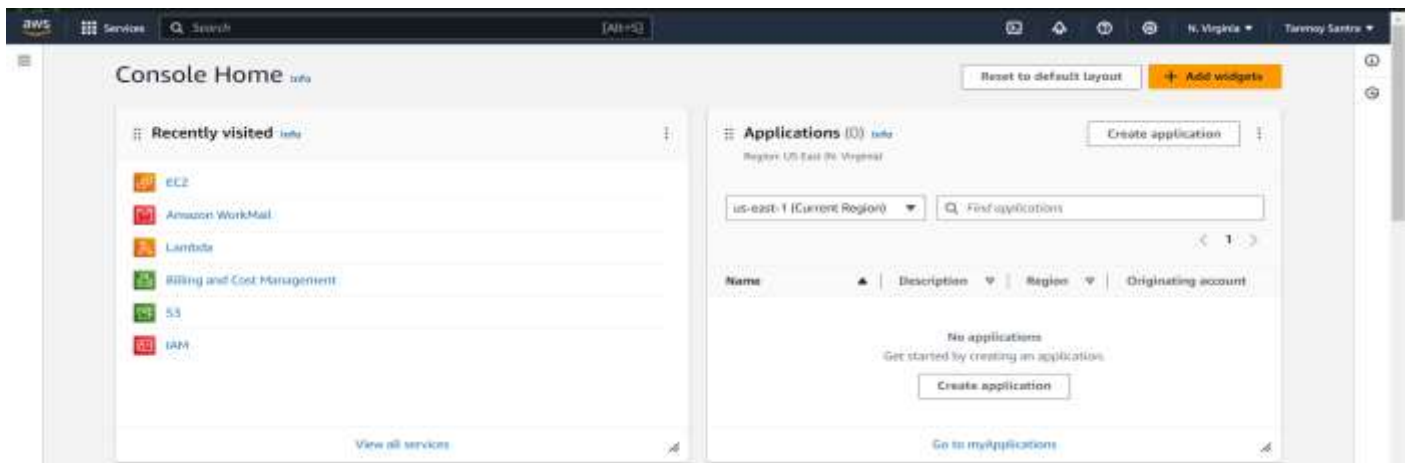


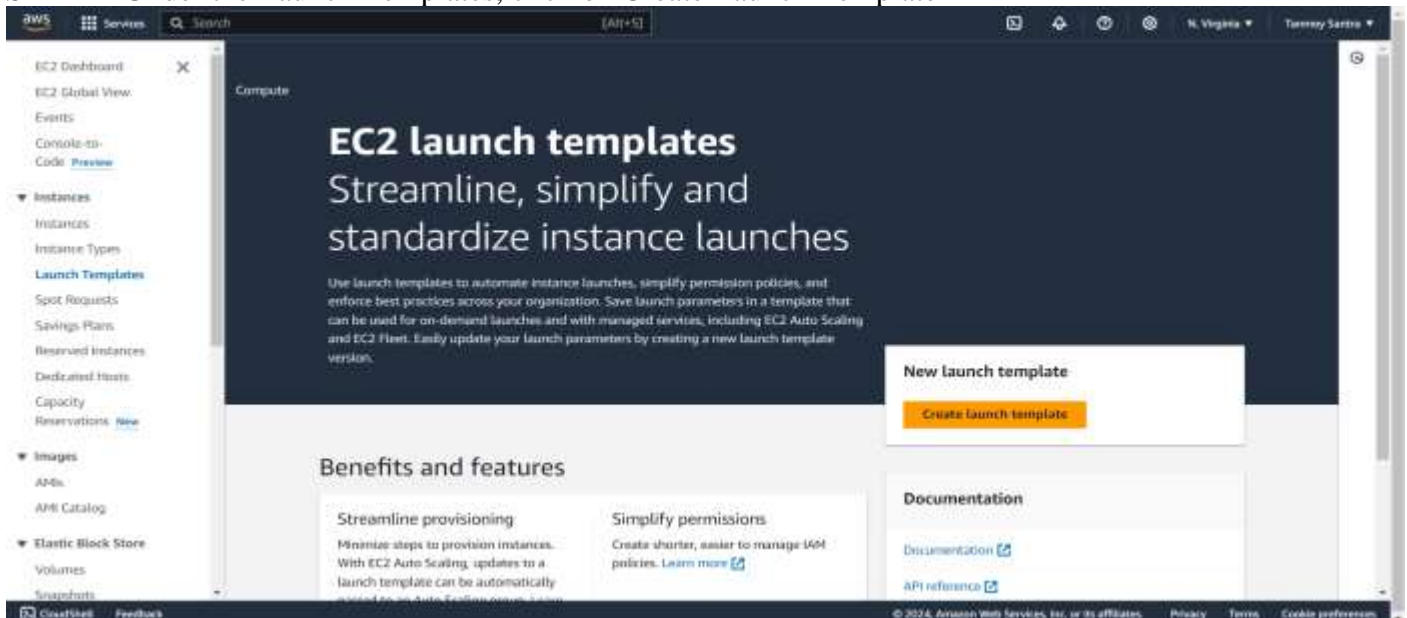
ASSIGNMENT – 11

PROBLEM STATEMENT -Build scaling plans in AWS that balances the load on different EC2 instances.

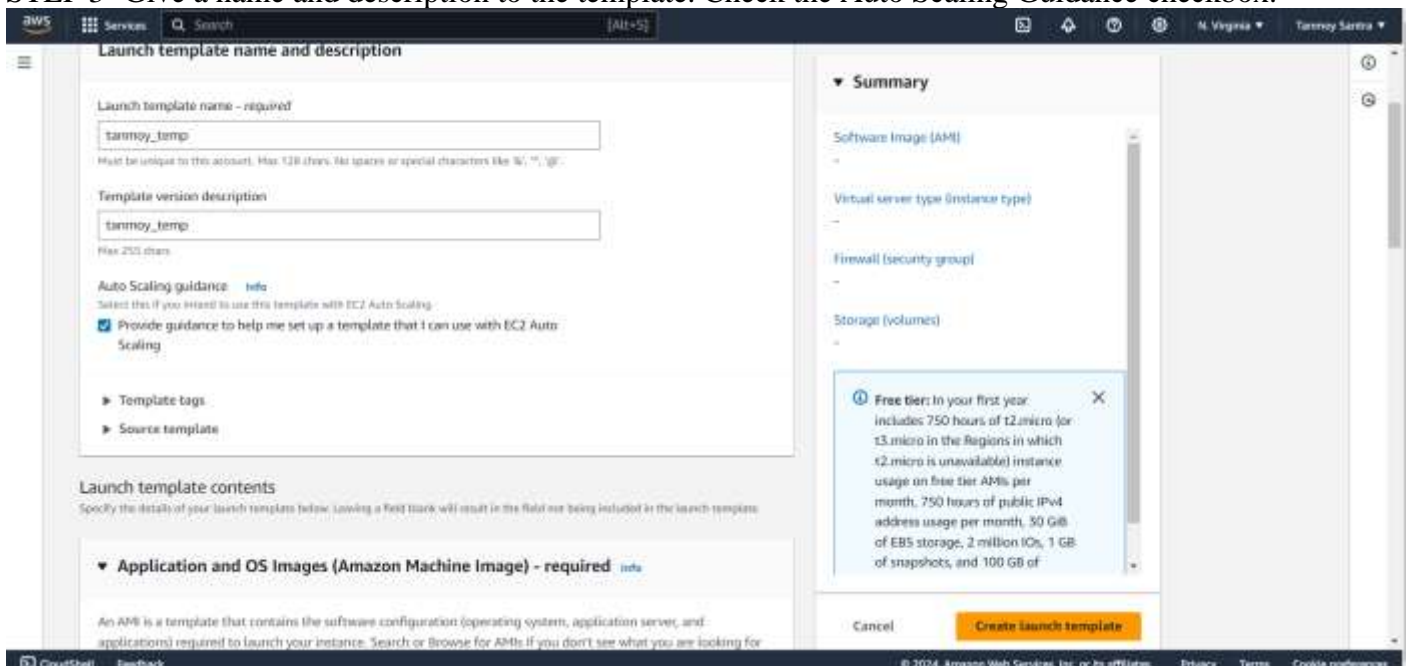
STEP 1- From AWS home screen, select EC2 option.



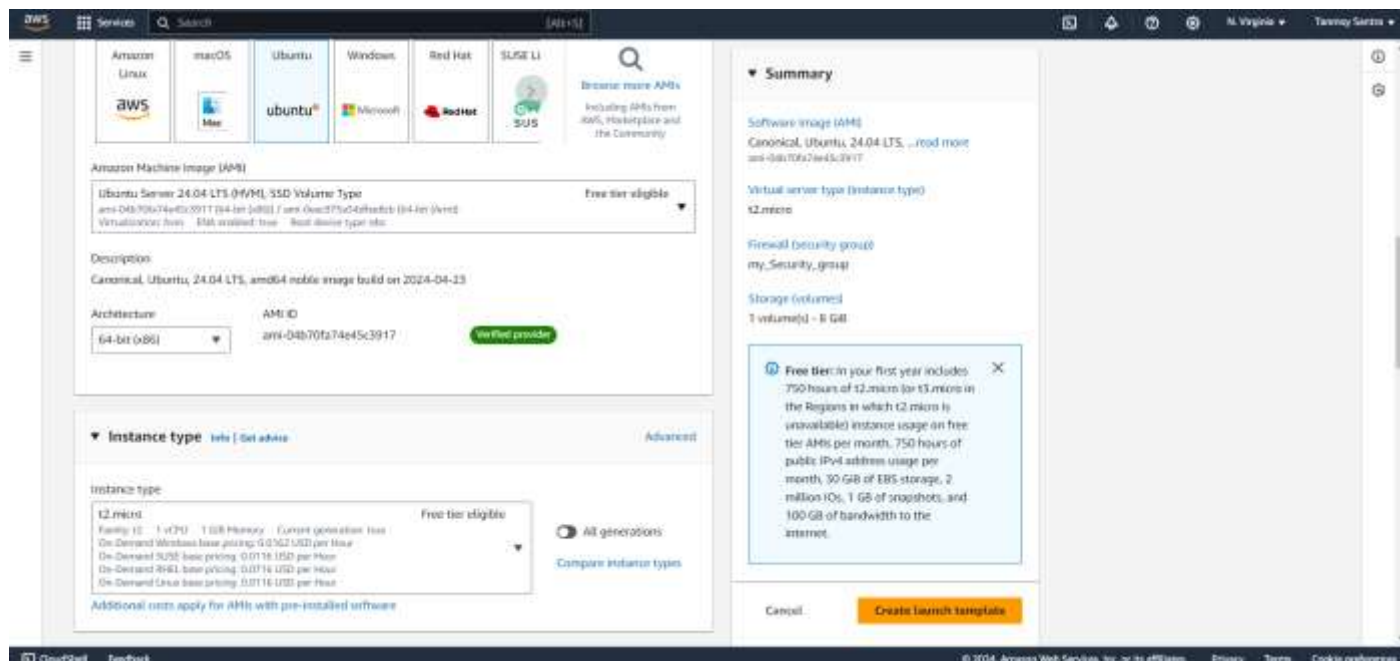
STEP 2- Under the Launch Templates, click on Create Launch Template



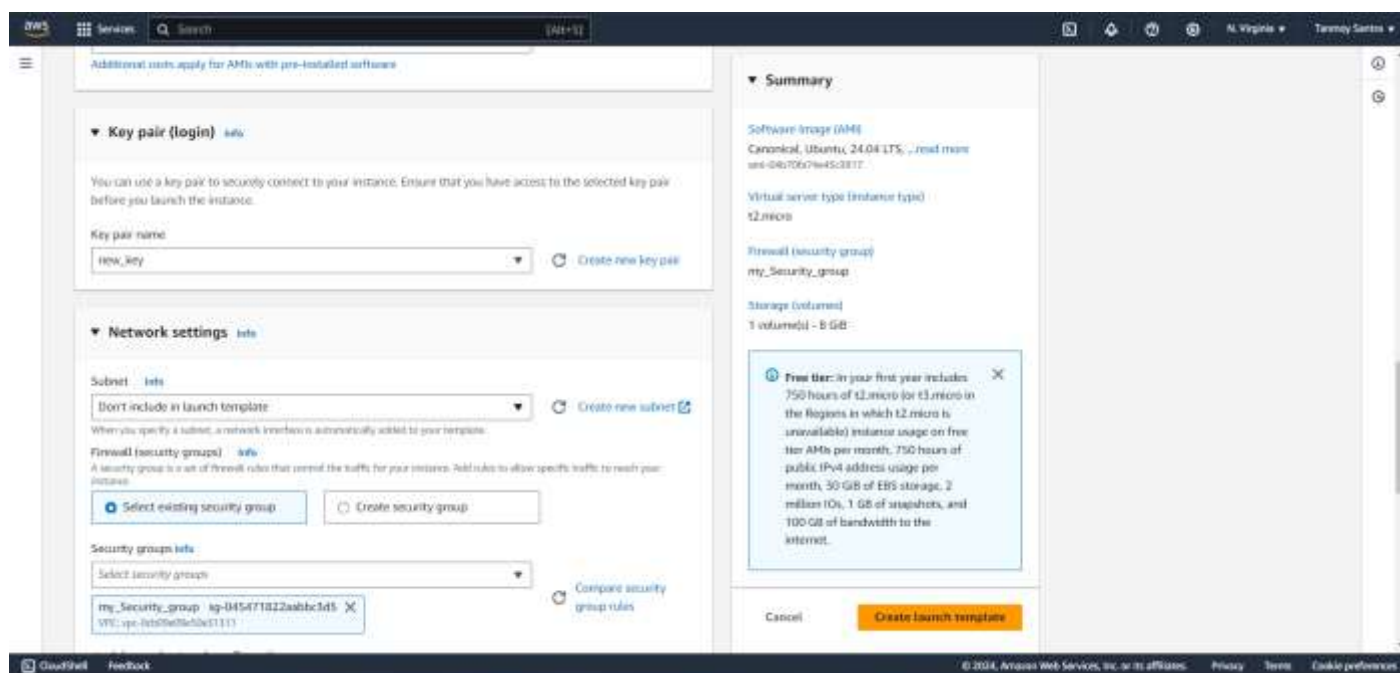
STEP 3- Give a name and description to the template. Check the Auto Scaling Guidance checkbox.



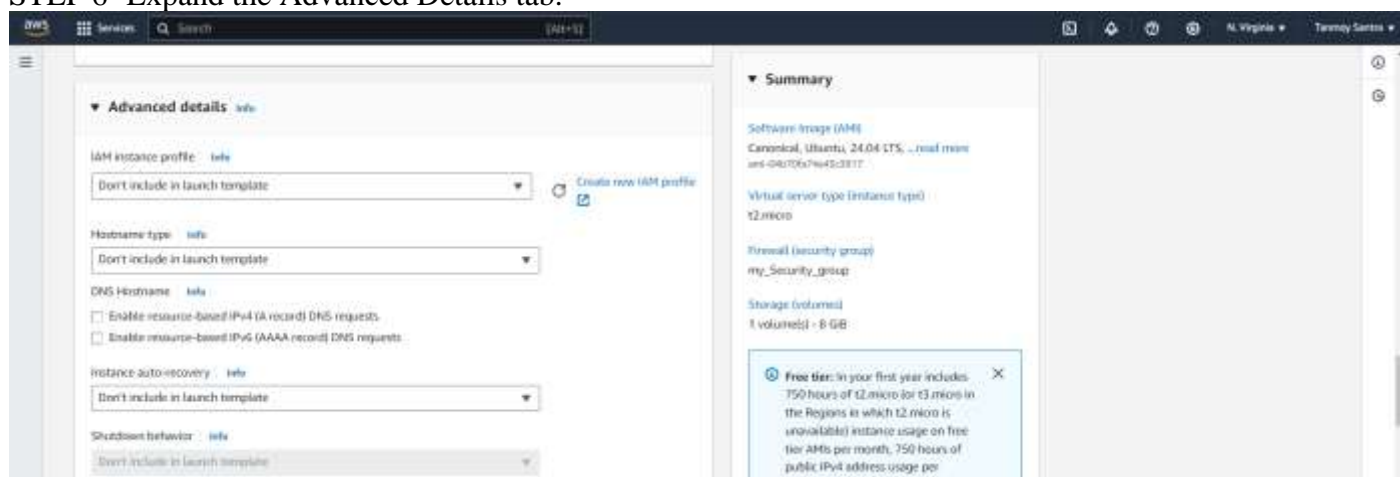
STEP 4- Select Ubuntu & under instance type, select t2.micro .



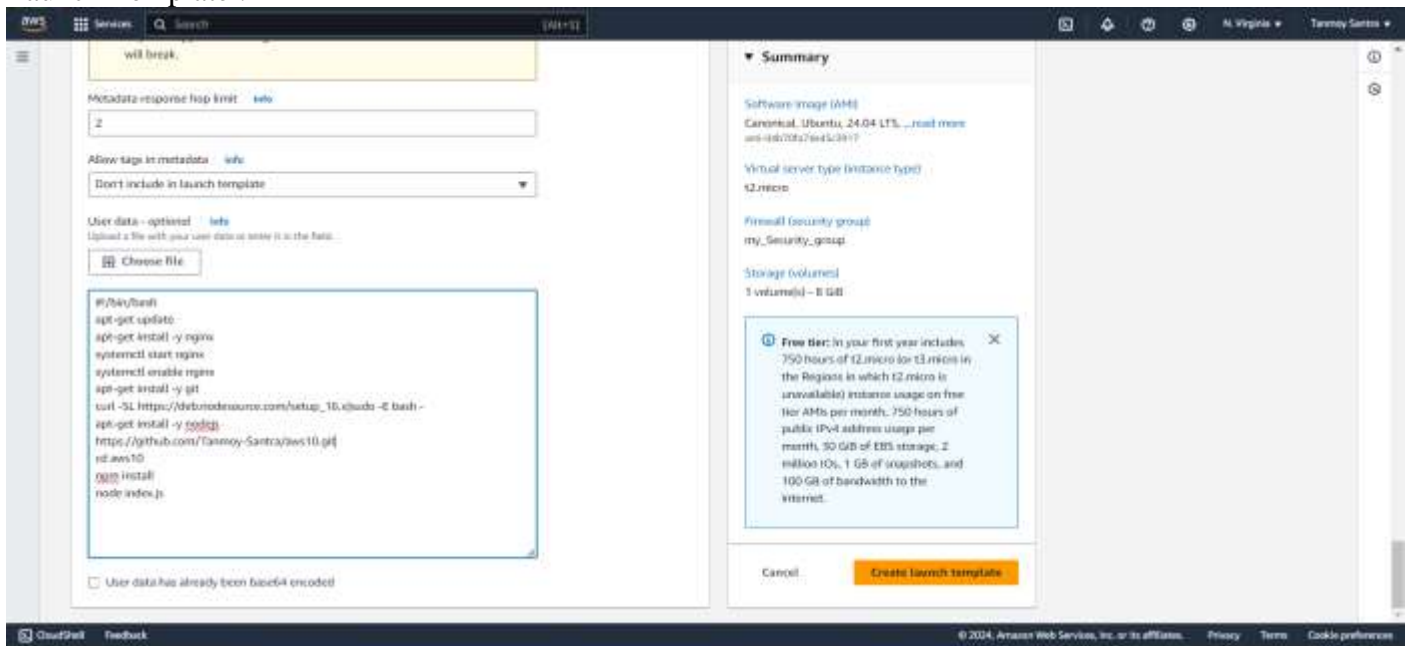
STEP 5- Under key pair, select an existing key and select the user created Security group.



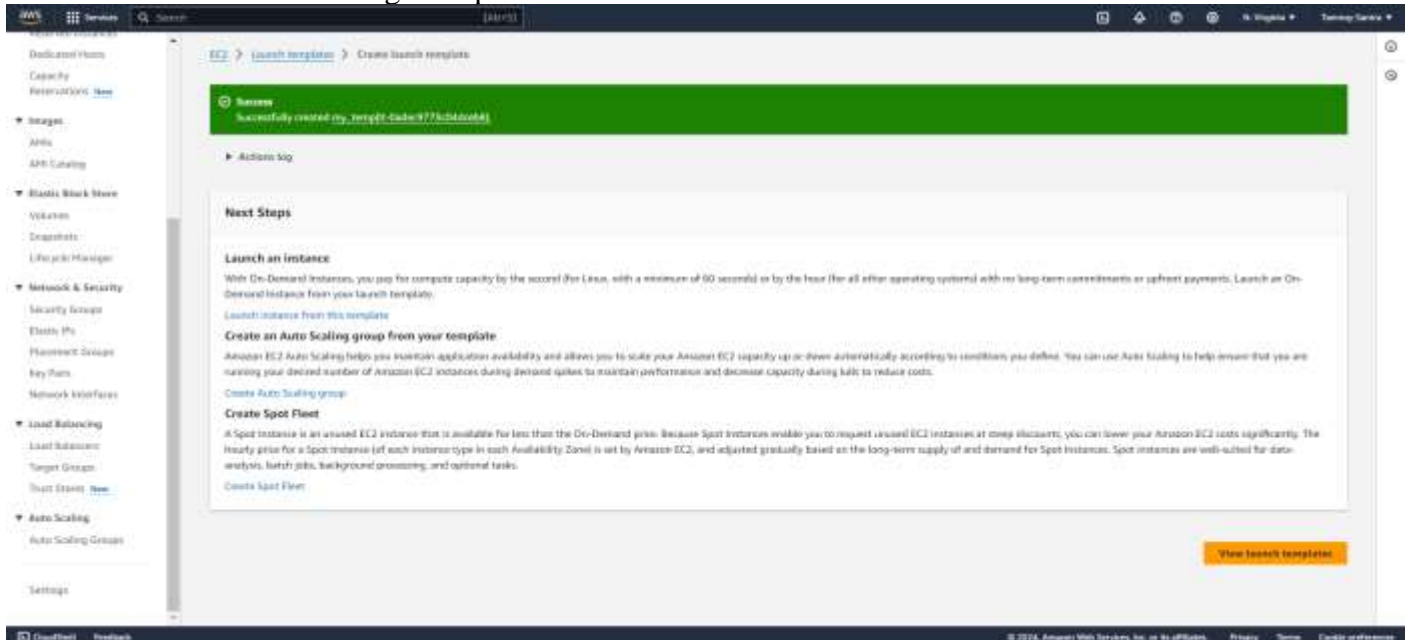
STEP 6- Expand the Advanced Details tab.



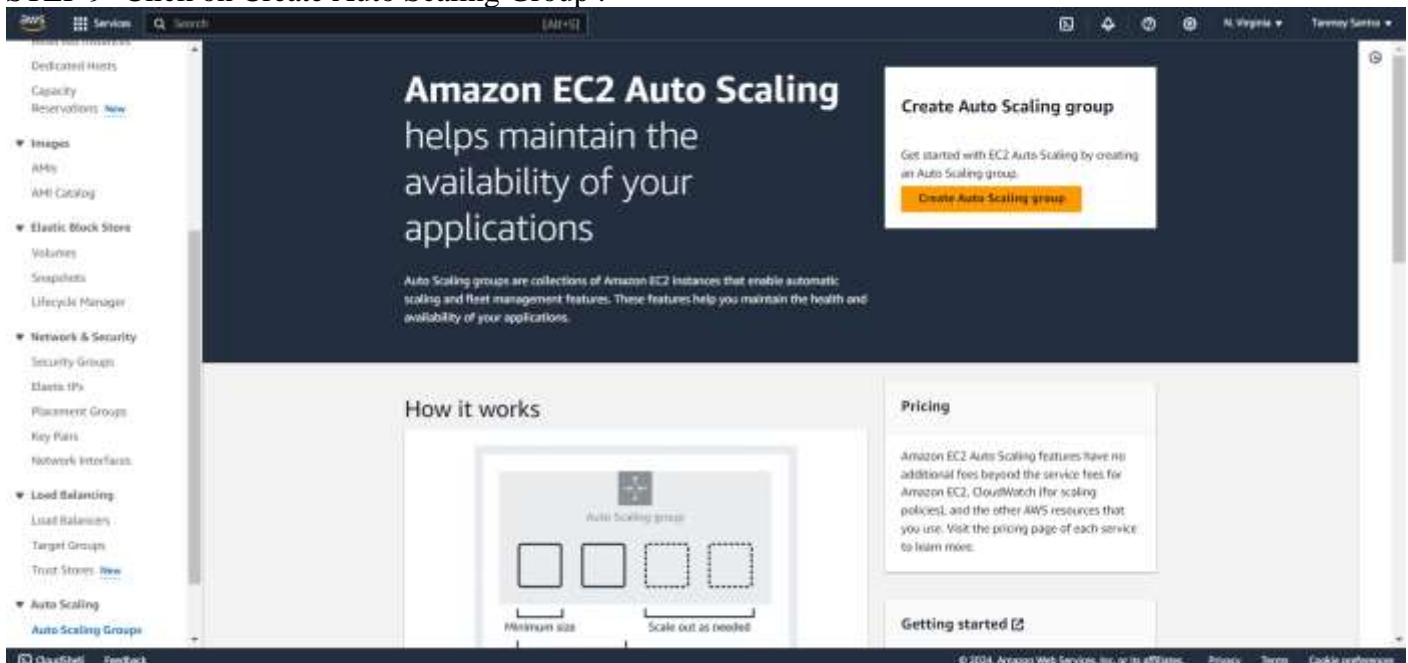
STEP 7- Scroll down to the bottom, in the bash console type the following commands:Then click on Create Launch Template .



STEP 8-Click on Auto Scaling Group .



STEP 9- Click on Create Auto Scaling Group .



STEP 10- Give a name and select the newly created Template. Then, click on Next .

STEP 11- In Network tab select all the available zones. Then click on next.

STEP 12- Select Attach a new load balancer , select Application Load Balancer & select Internet-Facing .

STEP 13- Give the port no. 4000 & select Create a target group. Then select No VPC Lattice Service.

The screenshot shows the 'Listeners and routing' configuration page in the AWS Management Console. The 'Protocol' is set to HTTP and the 'Port' is 4000. The 'Default routing (forward to)' dropdown is set to 'Create a target group'. Below this, the 'New target group name' is 'targetgroup/123-1'. The 'VPC Lattice integration options' section shows 'No VPC Lattice service' selected.

STEP 14- Check the Turn on Elastic Load Balancing Health checks checkbox. Give the Health Check Grace Period of 240 seconds. Click on NEXT.

The screenshot shows the 'Health checks' configuration page in the AWS Management Console. The 'Turn on Elastic Load Balancing health checks' checkbox is checked. The 'Health check grace period' is set to 240 seconds. The 'Additional settings' section shows 'Monitoring' and 'Default instance warmup' options.

STEP 15-Under Desired capacity, give a size of 2.Under Scaling, give min capacity 2 & max capacity 3.

The screenshot shows the 'Configure group size and scaling' configuration page in the AWS Management Console. The 'Desired capacity' is set to 2. The 'Scaling' section shows 'Min desired capacity' set to 2 and 'Max desired capacity' set to 3.

STEP 16-Select Target tracking scaling policy . And give the instance warmup time of 240 seconds. Then click on Next .

The screenshot shows the 'Automatic scaling - optional' configuration page in the AWS Management Console. 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'. The 'Instance warmup' is set to '240 seconds'. The 'Disable scale-in to create only a scale-out policy' checkbox is unchecked.

STEP 17- Click on Next .

The screenshot shows the 'Add notifications - optional' configuration page. The 'Add notification' button is visible. The 'Next' button is highlighted in orange.

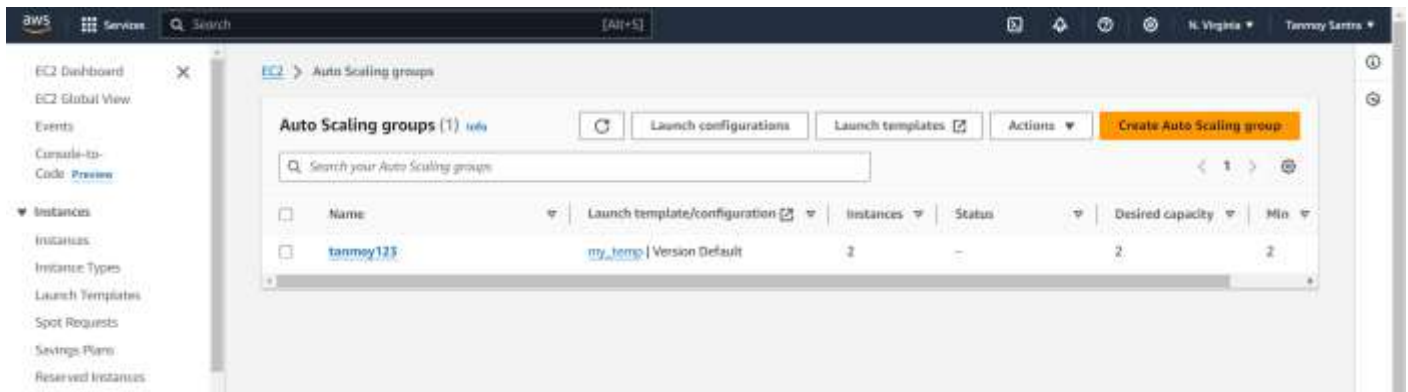
STEP 18- Click on Next .

The screenshot shows the 'Add tags - optional' configuration page. The 'Add tag' button is visible. The 'Next' button is highlighted in orange.

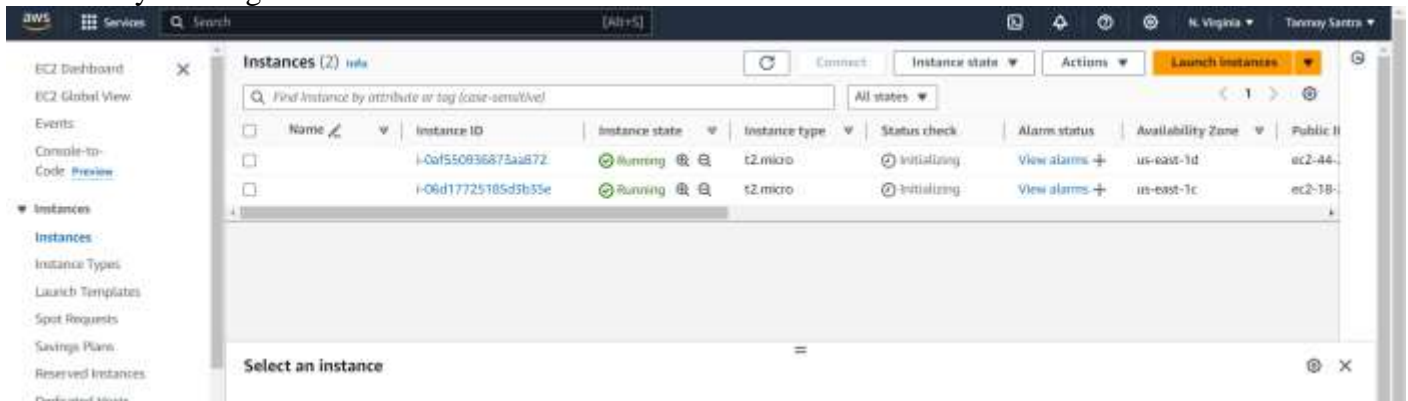
STEP 19- Review all the data of the group to be created and click on Create Auto Scaling Group .

The screenshot shows the 'Review' step of the 'Create Auto Scaling Group' wizard. It displays the 'Replacement behavior' (No policy), 'Instance scale-in protection' (Enabled), 'Step 5: Add notifications' (No notifications), and 'Step 6: Add tags' (No tags). The 'Create Auto Scaling group' button is highlighted in orange.

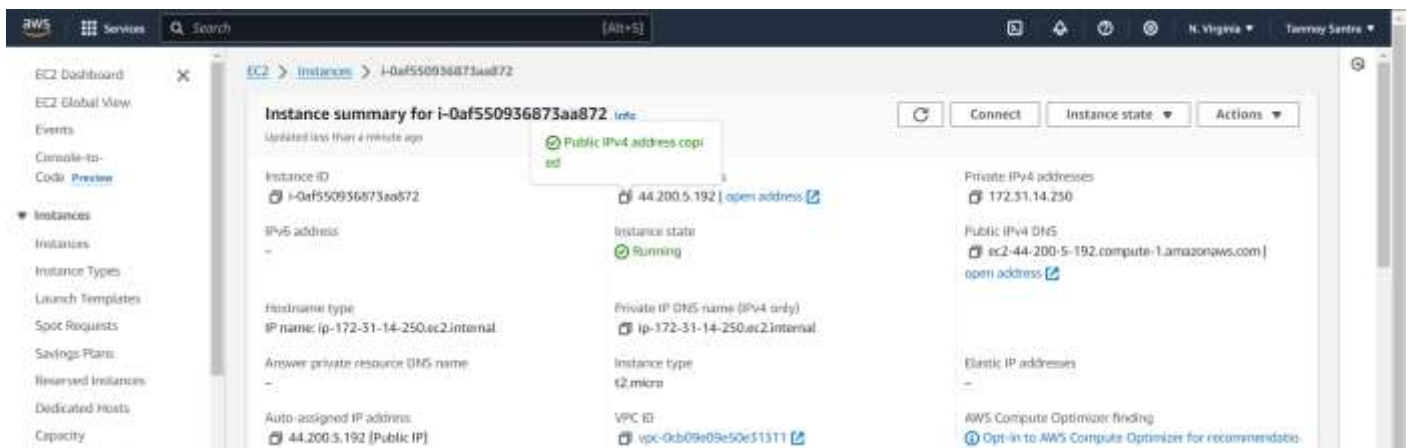
STEP 20- After creating the scaling group, go back to Instances from the left side menu.



STEP 21- Since the capacity was given as 2, two instances are created. Now open any one of the the instance by clicking on its id.



STEP 22- Copy its Public IPv4 Address.



STEP 23- Paste the copied address and click on Log in .



STEP 24- Click on New Terminal Console .

STEP 25- Type the command:

```
ubuntu@ip-172-31-14-250:~$ sudo nano infy.sh
```

STEP 26- Write the following code for an infinite loop in the infy.sh file.

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

STEP 27- Write the following commands in the terminal:

```
ubuntu@ip-172-31-14-250:~$ sudo chmod 777 infy.sh
ubuntu@ip-172-31-14-250:~$ sh infy.sh
```

```
ubuntu@44.200.5.192:22 - Bitvise xterm - ubuntu@ip-172-31-14-250: ~
inside loop
inside loop
inside loop
inside loop
inside loop
```


STEP 28- Select both the instances, then under monitoring go to CPU utilization and enlarge it.

STEP 29- The graph shows the CPU Utilization for both the instances.

When the CPU utilization exceed the limit for both the instances, a new instance will be created.

