**CONFIGURING AUTO SCALING GROUPS**

**Ex.No : 06 Name : S.Yuvasri**

**Date: Roll.No: 23MCA039**

**AIM:**

To configure auto scaling groups in aws and follows the operation

* Create an Auto Scaling Group with a specified desired capacity, minimum, and maximum instances.
* Manually increase the CPU utilization on a Linux server using the yes command.
* Observe how the Auto Scaling Group reacts to the increased load by scaling up the number of instances.

**THEORETICAL BACKGROUND:**

**CPU Utilization**

CPU utilization refers to a computer’s usage of processing resources, or the amount of work handled by a CPU. Actual CPU utilization varies depending on the amount and type of managed computing tasks. Certain tasks require heavy CPU time, while others require less because of non-CPU resource requirements.

**Auto Scaling groups**

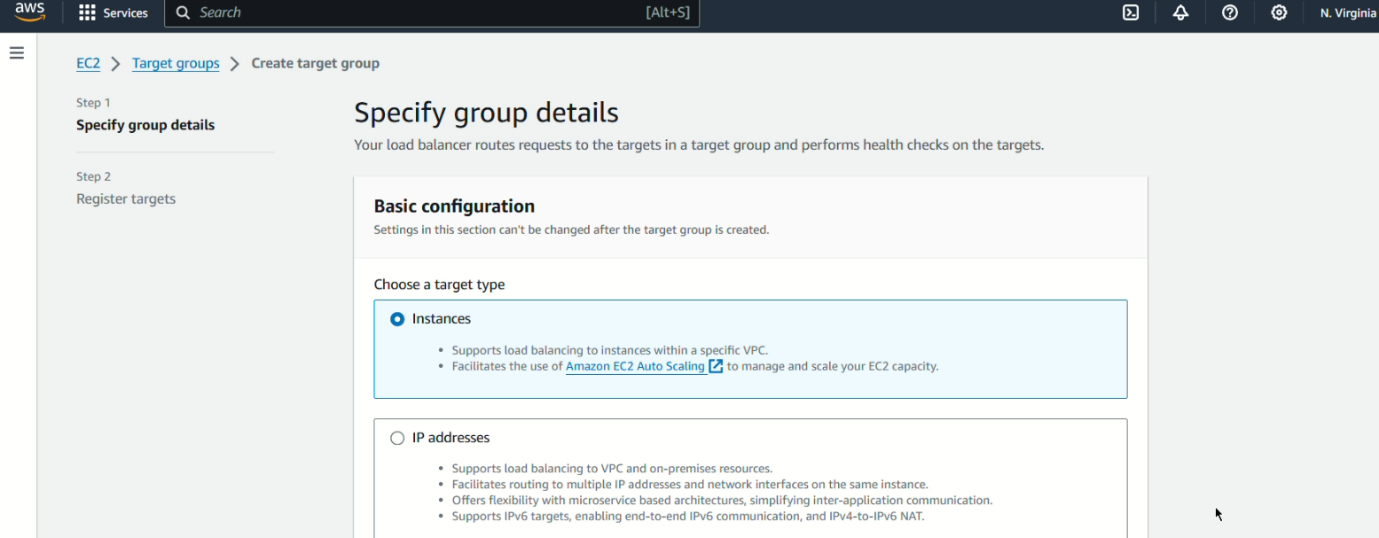
AutoScaling in AWS is the advanced cloud computing feature that provides automatic resource management based on the server’s load. The Resources associated with a server cluster generally scale up and scale down dynamically through mechanisms such as load balancer, AutoScaling groups, Amazon Machine Image (AMI), EC2 Instances, and Snapshots. The AWS AutoScaling feature helps in managing the pick time load in business.

When the traffic would start increasing, the AWS autoscaling service would automatically initiate the launch of another instance with the same configuration of your current server with the help of the AMI of your server.

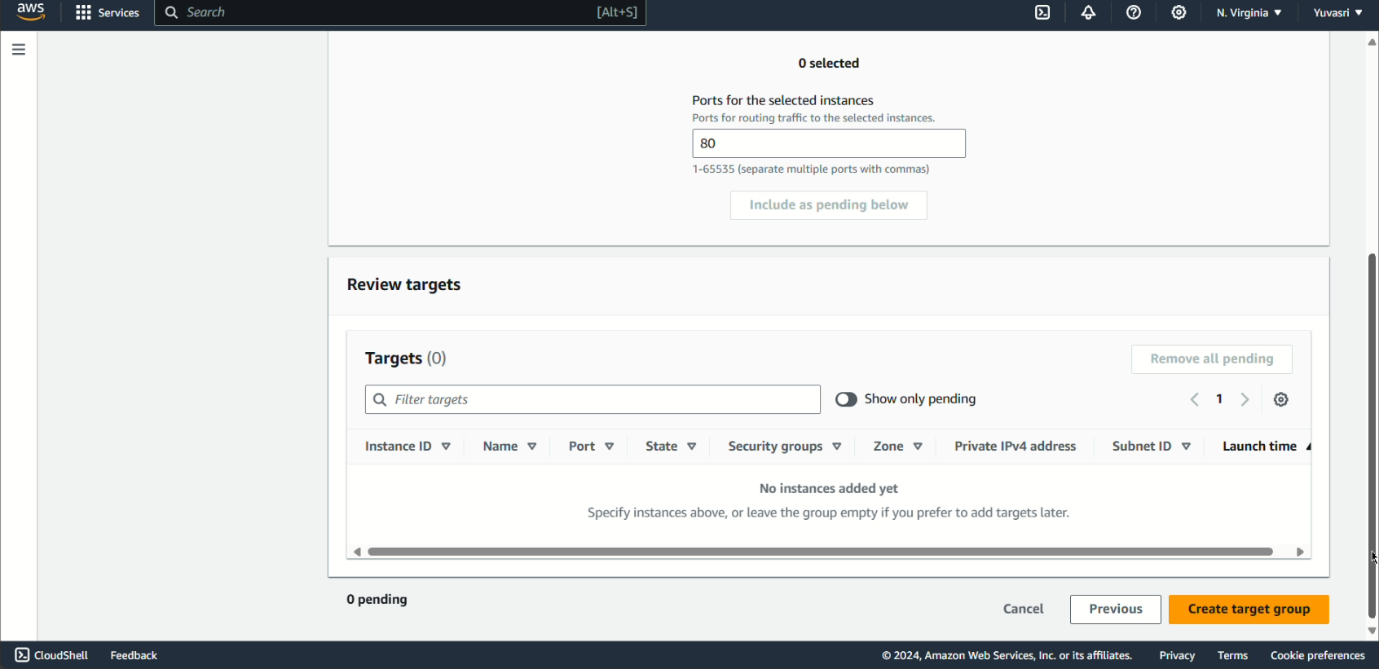
**STEPS INVOLVED:**

**1) Create an Auto Scaling Group with a specified desired capacity, minimum, and maximum instances.**

**Step 1:** Create onr tatrget group . In the creation , choose instances and give name to the target group.

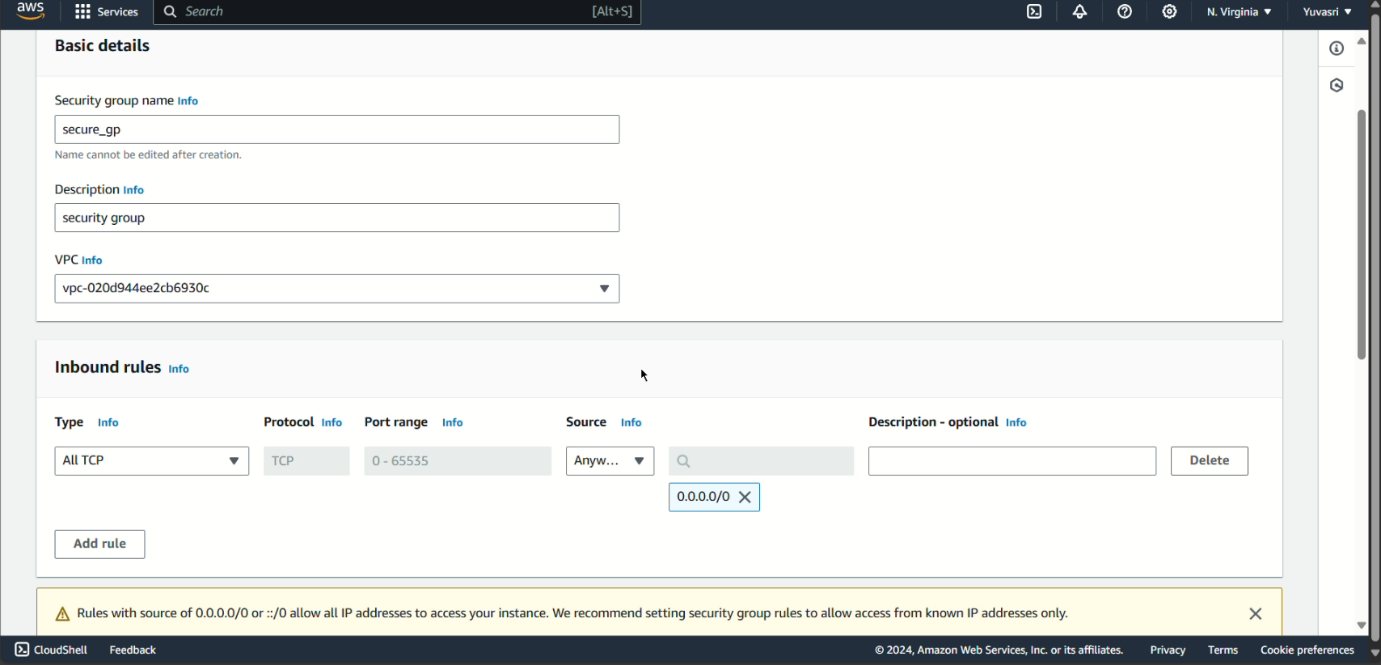


**Step 2:** After that no instances is created and click “create target group”

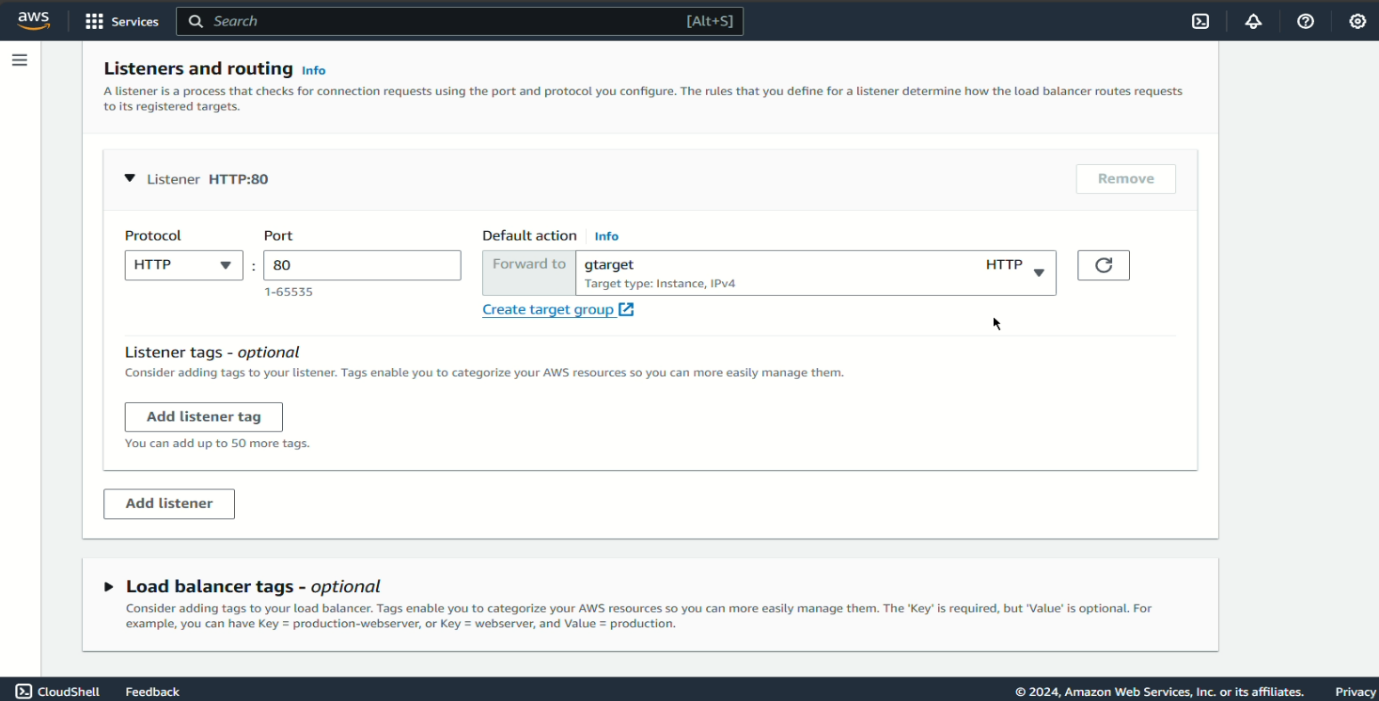


**Step 3:** Create load balancer go to load balancer and click create load balancer . Give name to the load balancer and click any three availability zones.

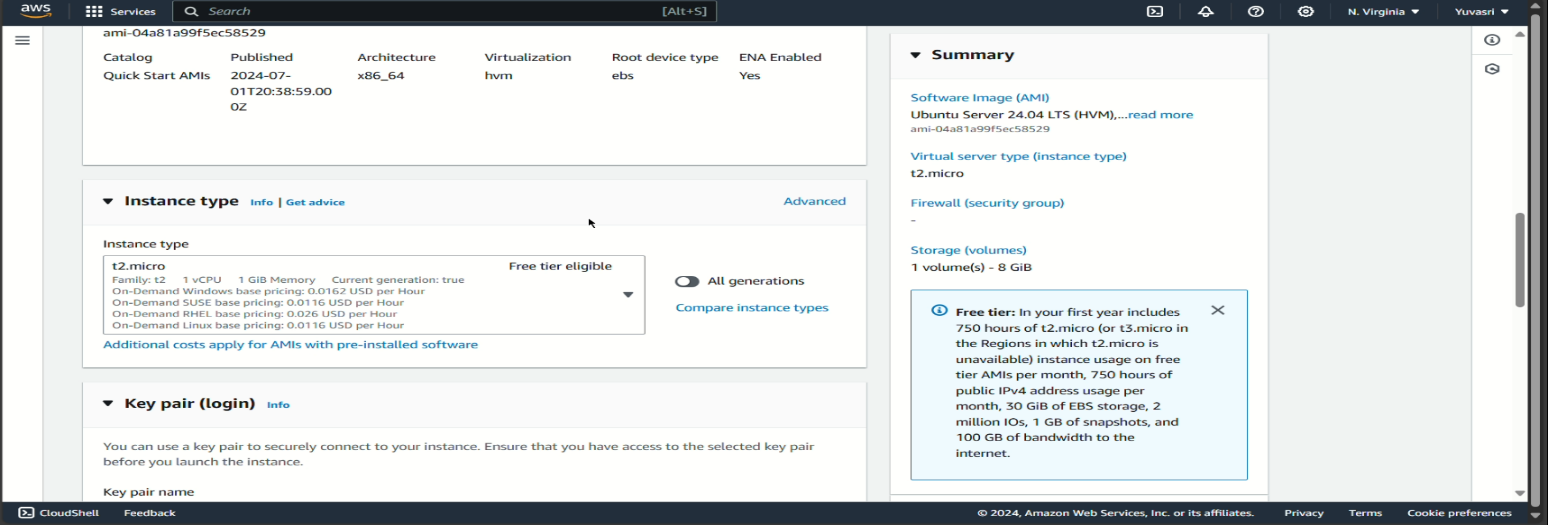
**Step 4:** Create one security group for the load balancer. In the security group , inbound rule is “All TCP” and source is “Anywhere IPV4”.



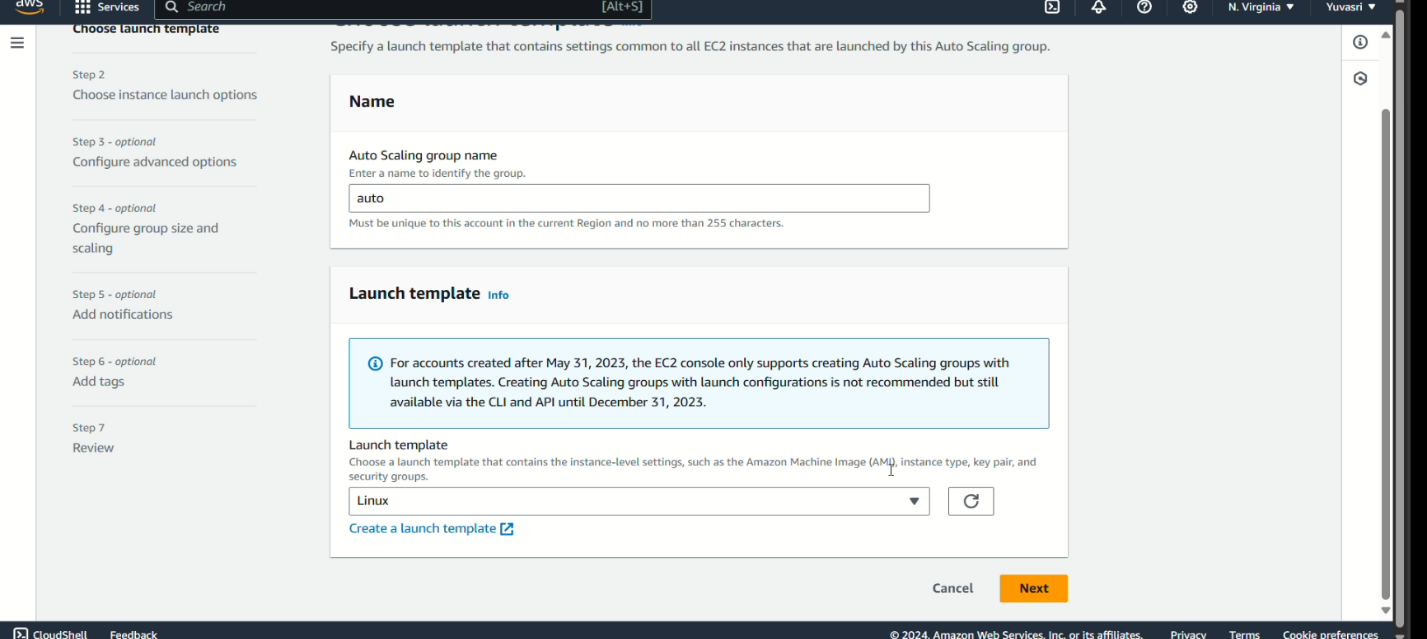
**Step 5:** In the listeners and routing column add our created target group name . And click “Create load balancer”.



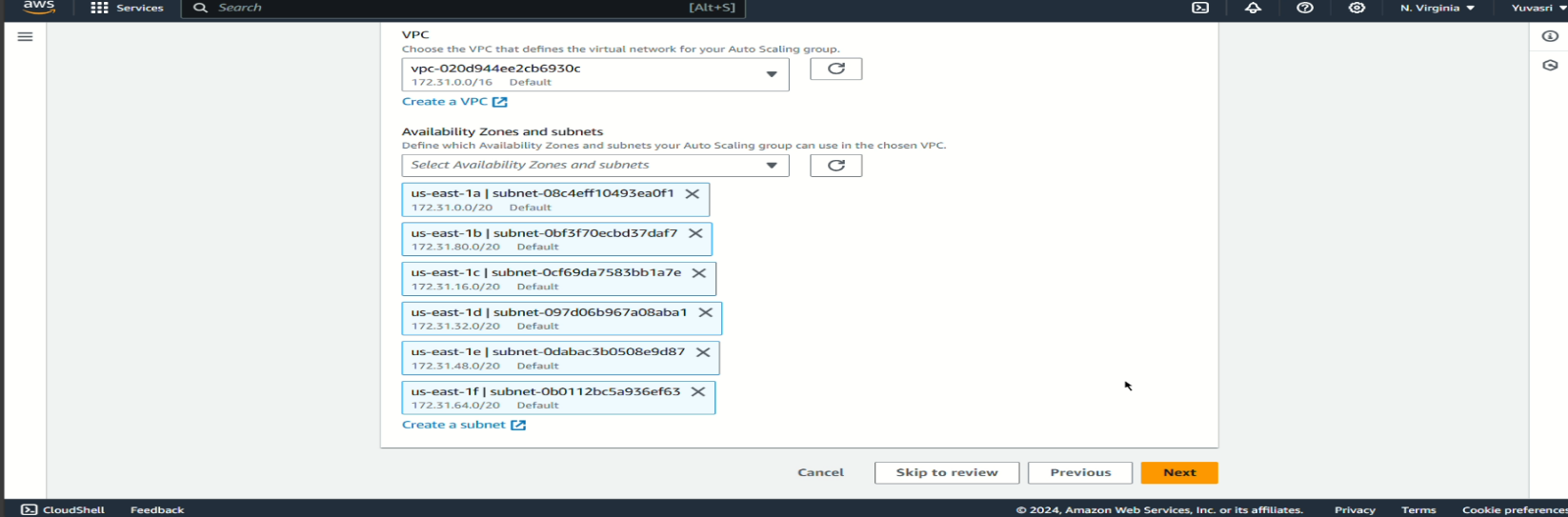
**Step 6:** Go to Launch template for creating one linux template. In the launch template creation click one ubuntu server and instance type is “t2.micro, create one new key pair and in the network settings select our existing security group while creating load balancer and click “Create launch template”.



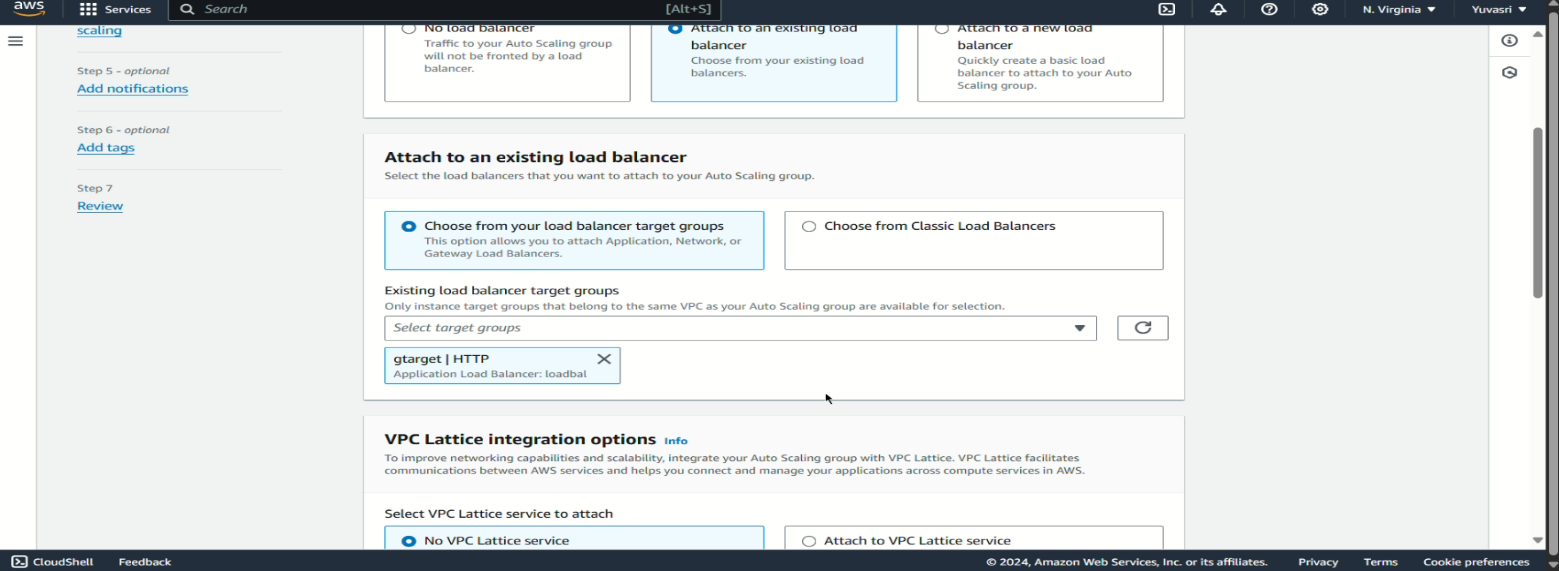
**Step 7:** Create auto scaling groups. Click create autoscaling groups and give name for the groups and in the launch template column select our launch template name and click “Next”



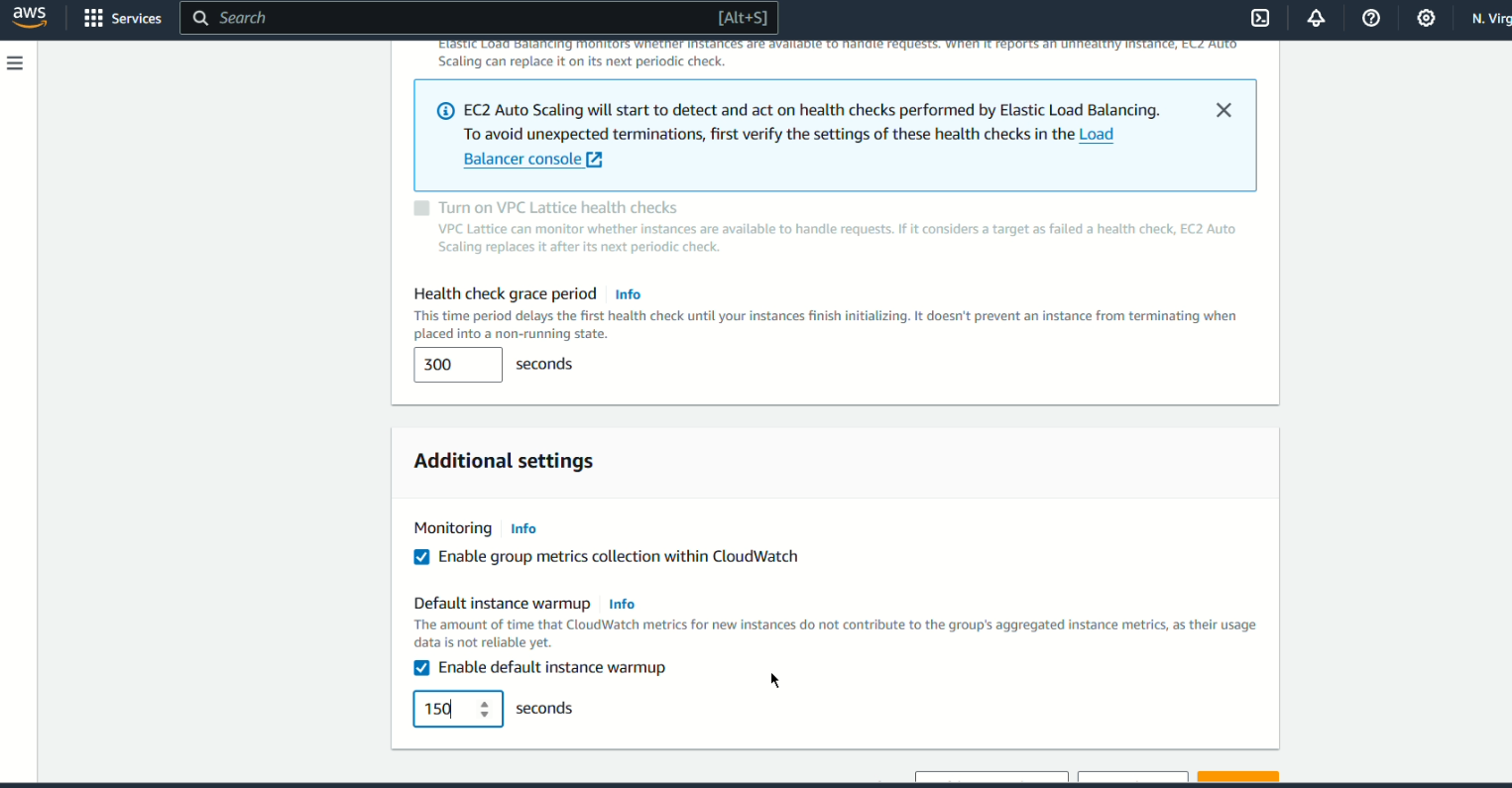
**Step 8:** Select one default vpc and check the availability zones any three or select all and click “Next”.



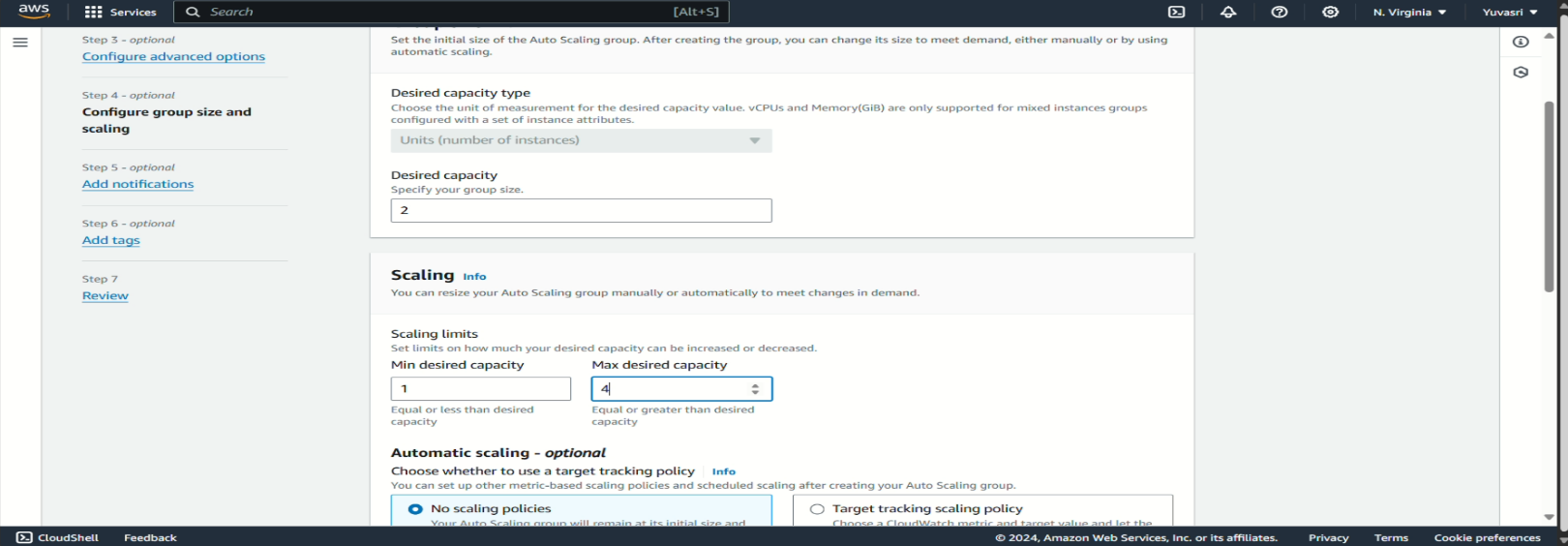
**Step 9:** Click “Attach to existing load balancer” and choose from your load balancer target group. In that select our target group.



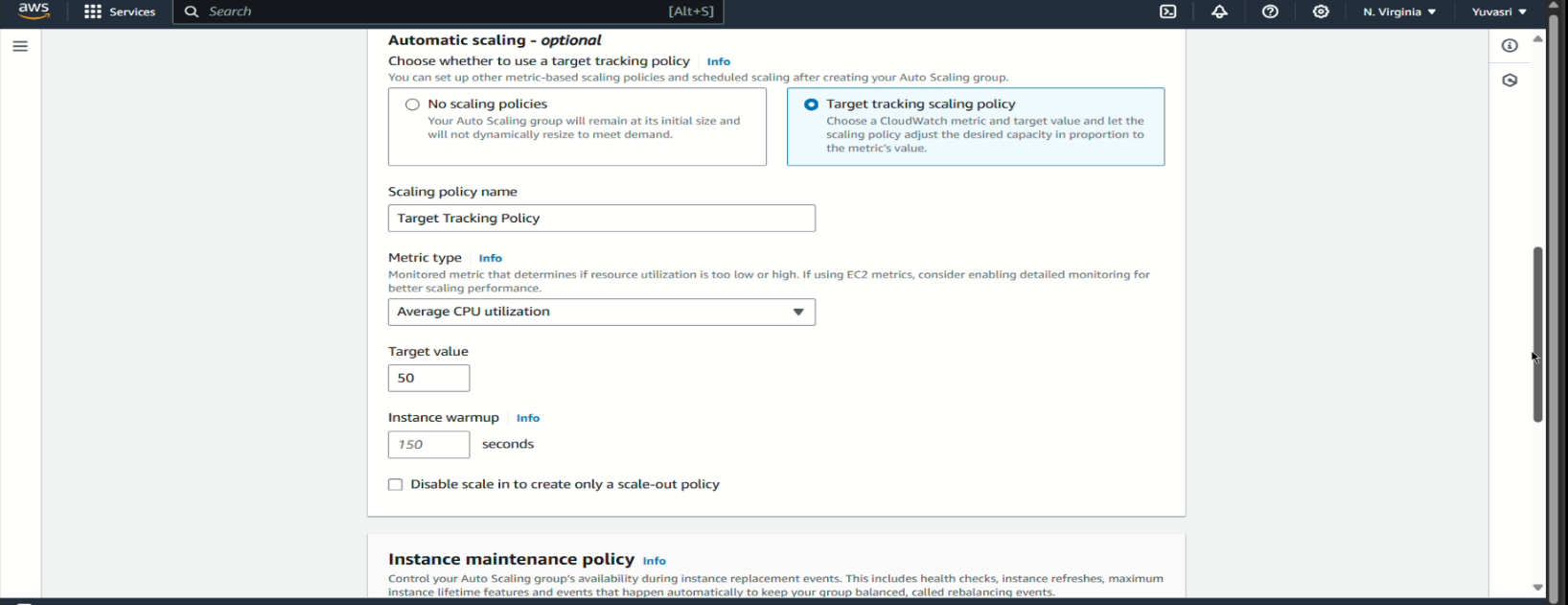
**Step 10:** In EC2 health checks , click “Turn on the elastic load balancing health checks” and health grade is 300 seconds. In additional settings select the two checkboxes and give 150 seconds and click “Next”.



**Step 11:** In the desired capacity give 2 and minimum desired capacity is 1 and maximum desired capacity is 4.

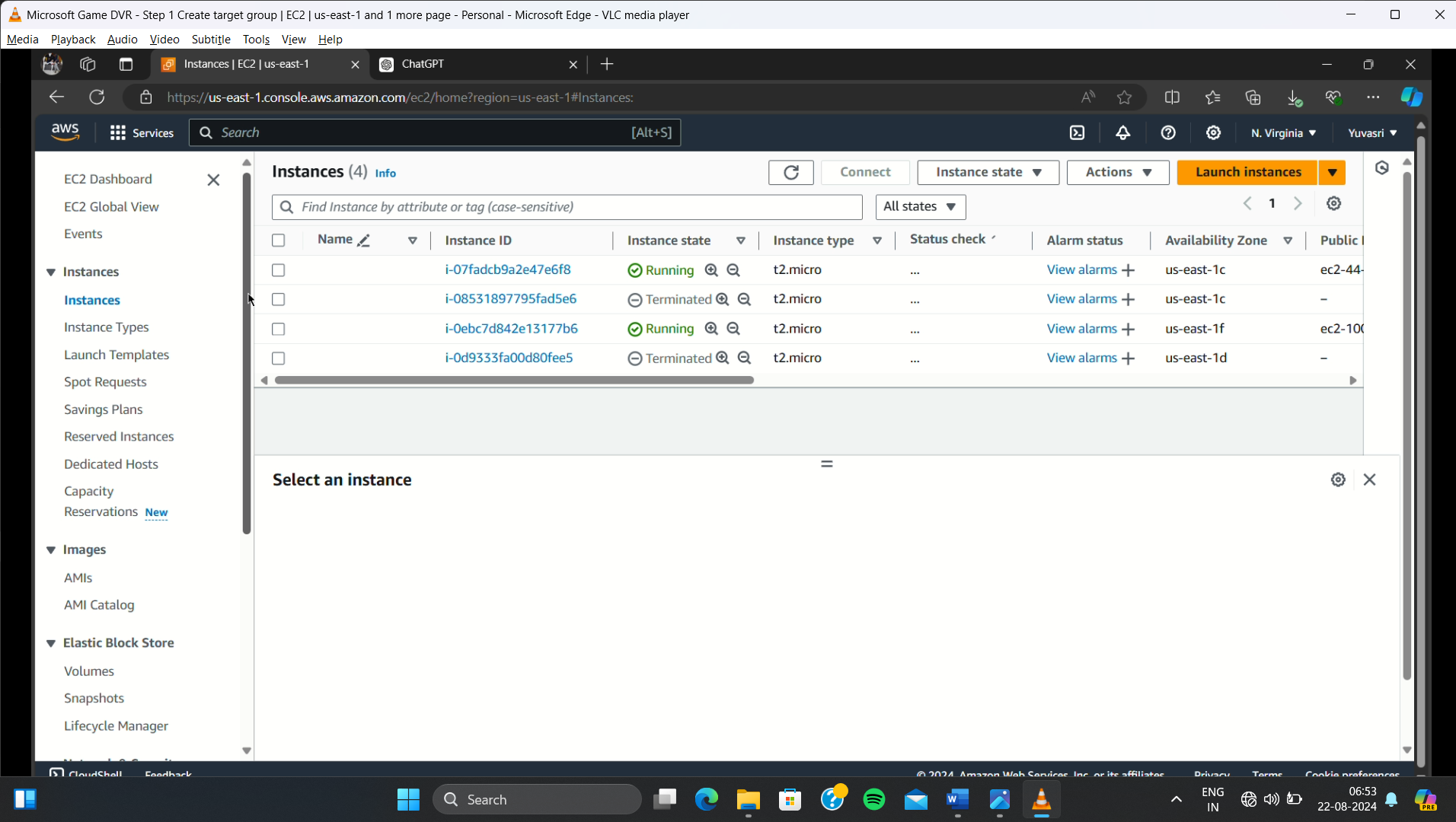


**Step 12:** In the auto scaling choose “Target tracking scaling policy” and metric type is averge CPU Utilization and choose “no policy” and click next. And click create auto scaling groups”.



**2) Manually increase the CPU utilization on a Linux server using the yes command.**

**Step 1:** After creating auto scaling groups there are two instances running connect the two instances.

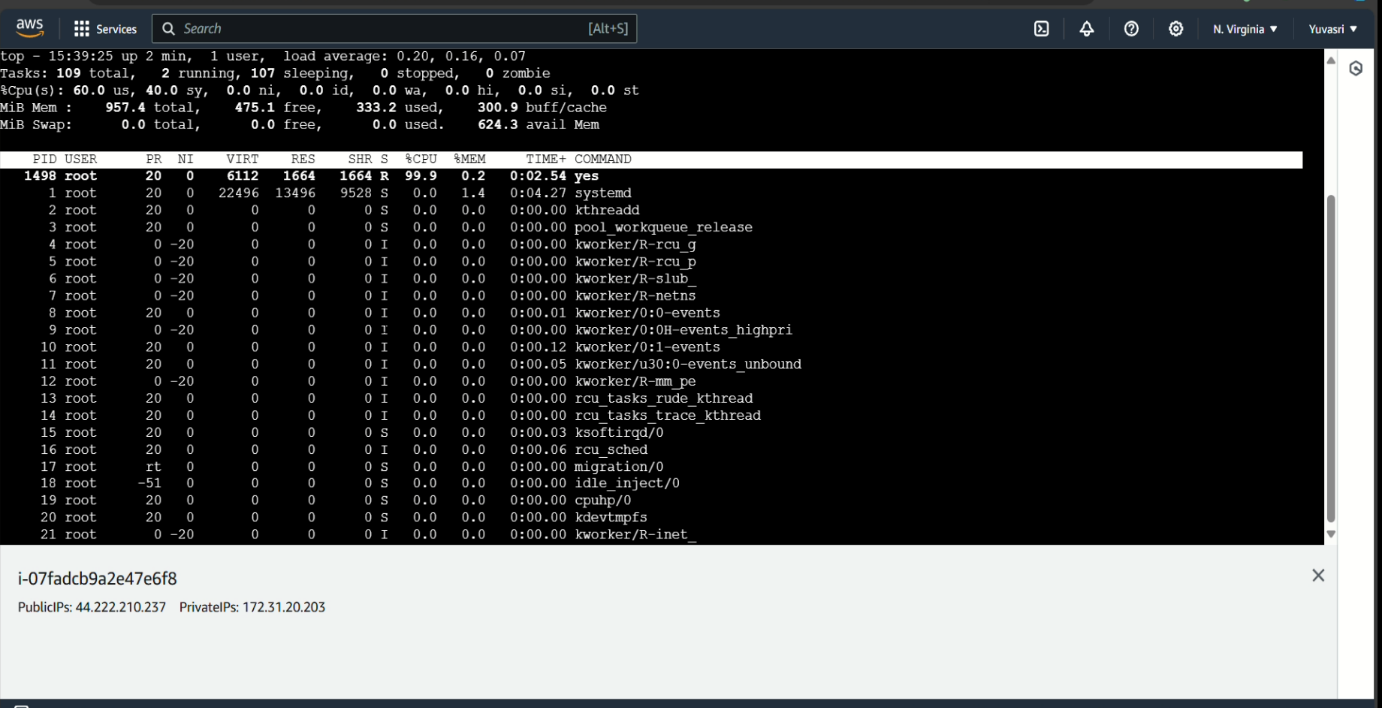


**Step 2:** In the instances give the following commands to manually increase the CPU utilization.

**sudo su //Go to the root directory**

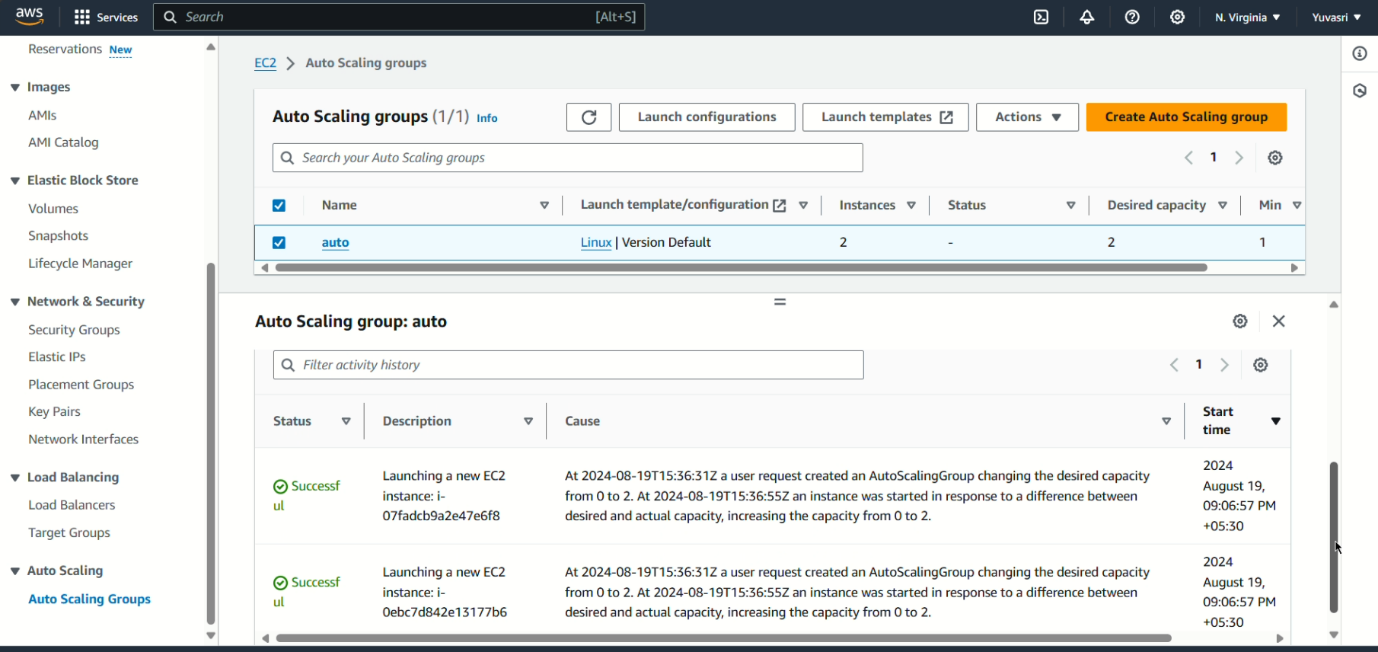
**yes > /dev/null & //Increase the CPU utilization**

**top //Monitor the CPU utilization**

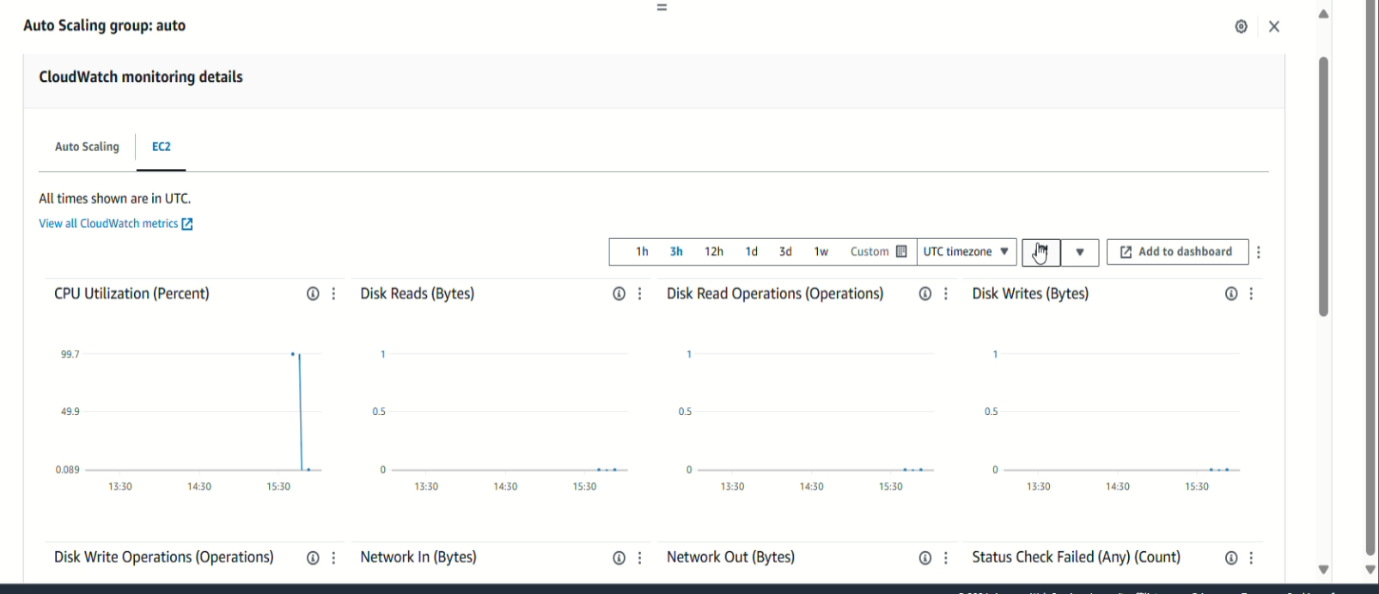
****

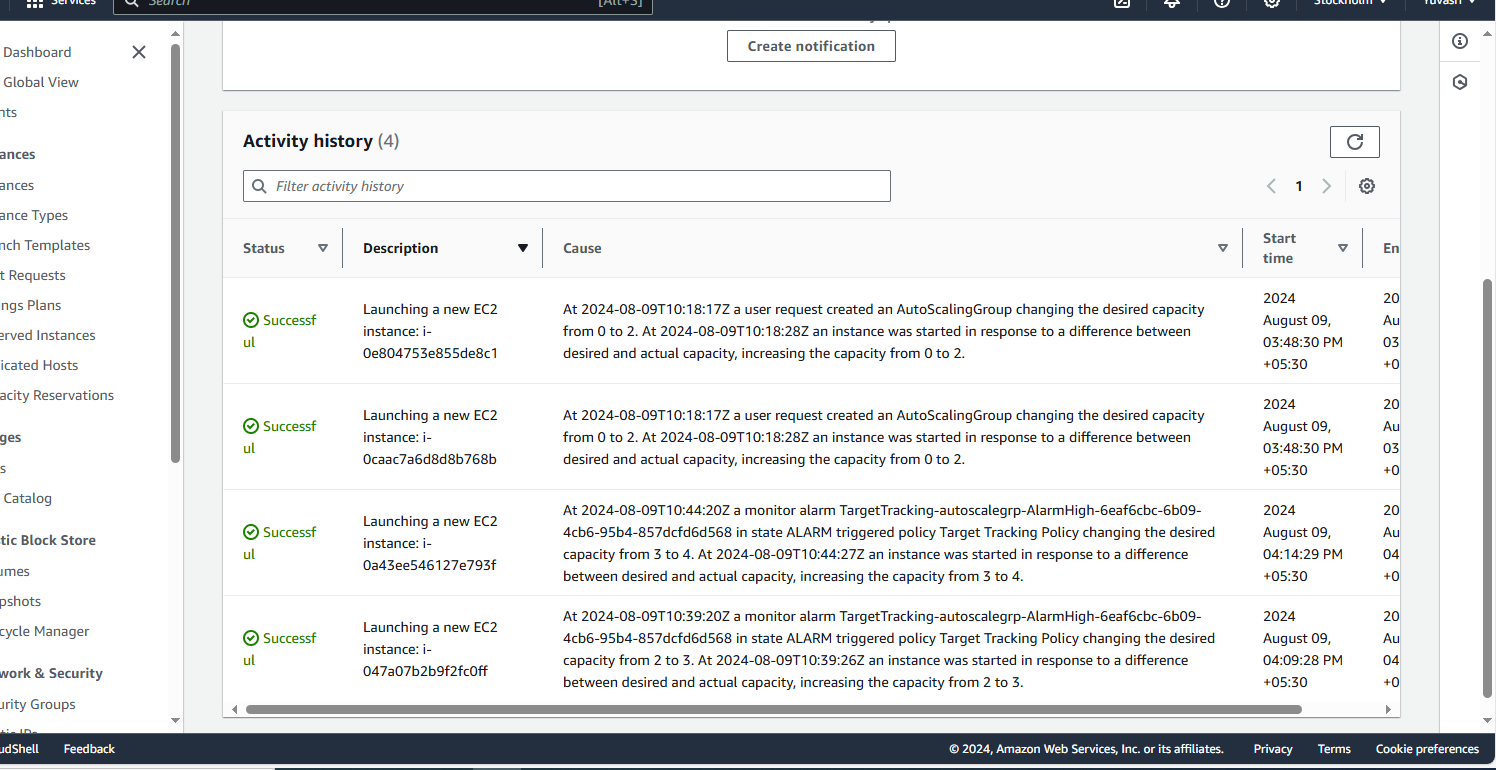
**3) Observe how the Auto Scaling Group reacts to the increased load by scaling up the number of instances.**

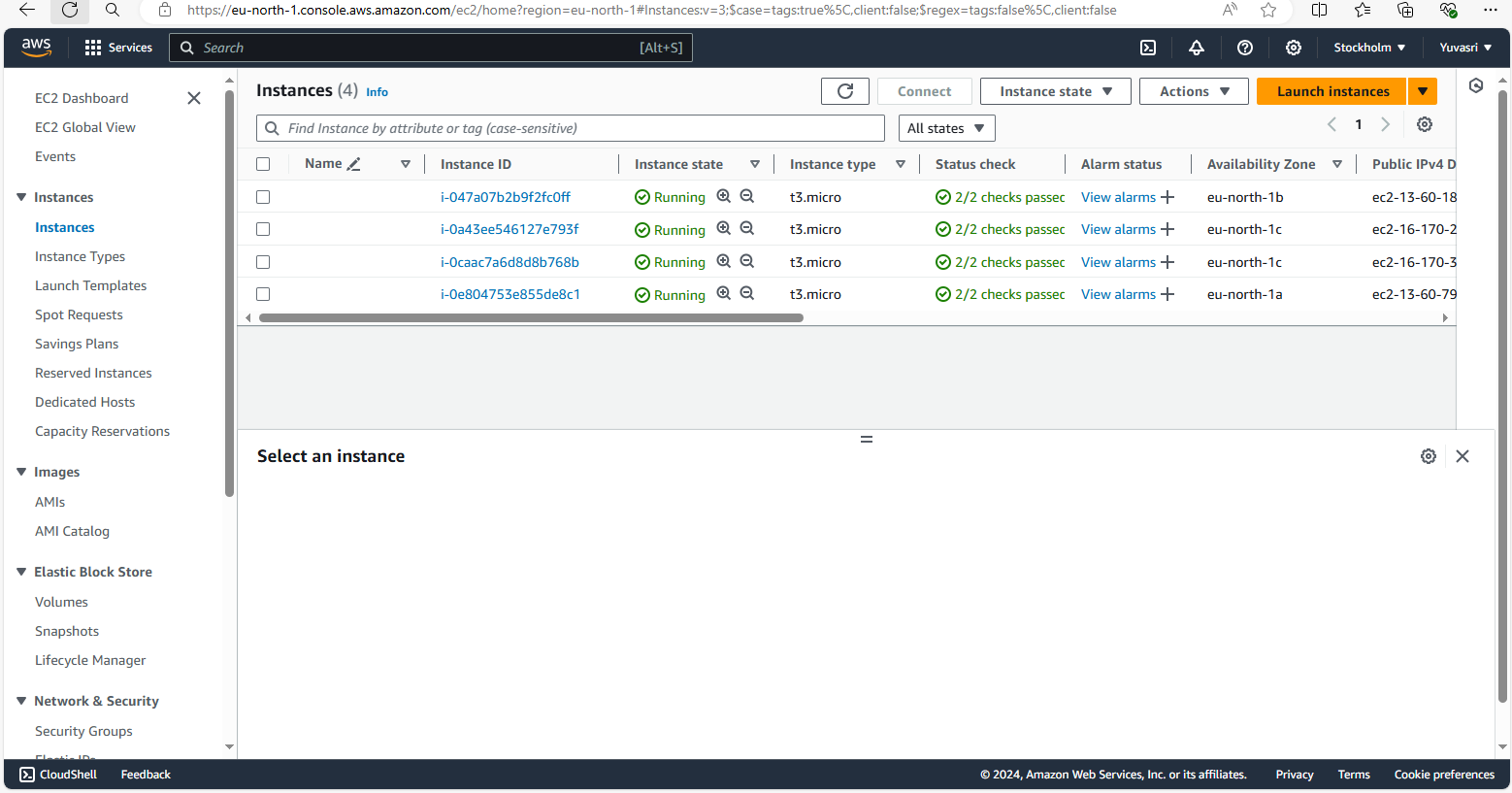
**Step 1:** Go to auto scaling groups , at first there are only 2 instance is running .



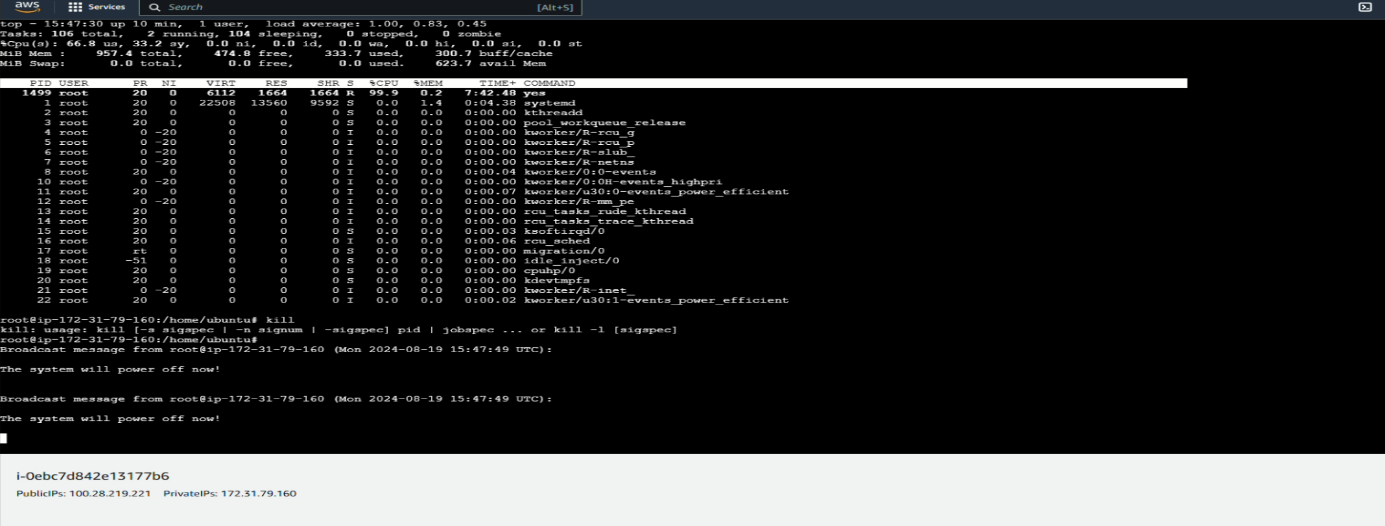
**Step 2:** Click our auto scaling group and go to monitoring. In the monitoring go to EC2 in the cloudwatch it displays the CPU Utilization .

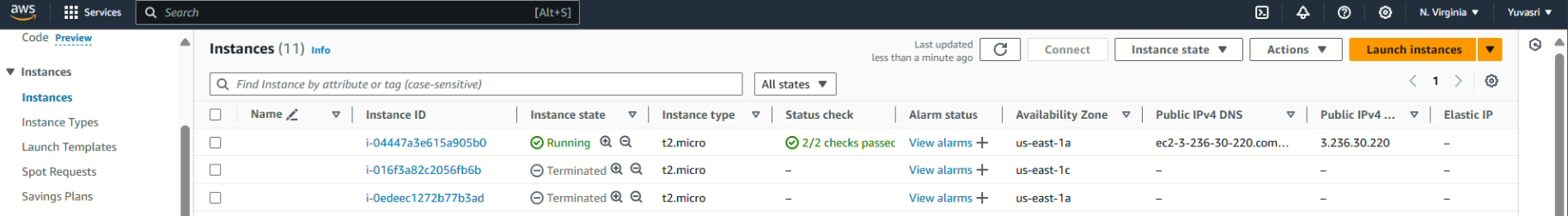


**Step 3:** After the CPU Utilization is increased to 90 and above there are two more instances is created and running in the activity area it is scaled out to our maximum desired capacity.



**Step 4:** For scaled in in the connecting linux instance give the command “Kill”. After that go to monitoring it will reduce the percentage below 20% at that time the instances are terminated our connecting instances is automatically power off and scaled in to only one instance is running .





**CONCLUSION:**

Thus the above steps for configuring auto scaling groups and monitor the scaled out process was executed successfully.