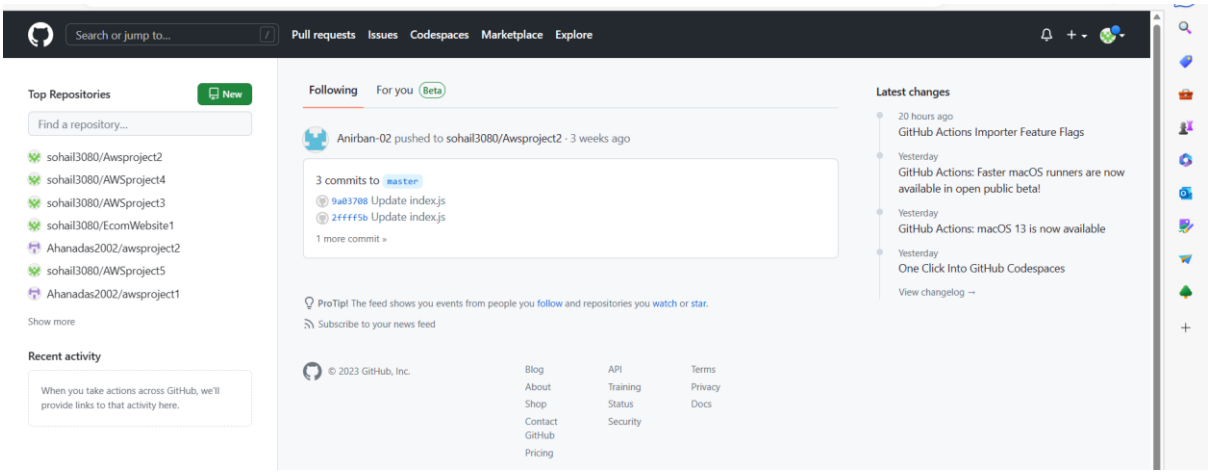
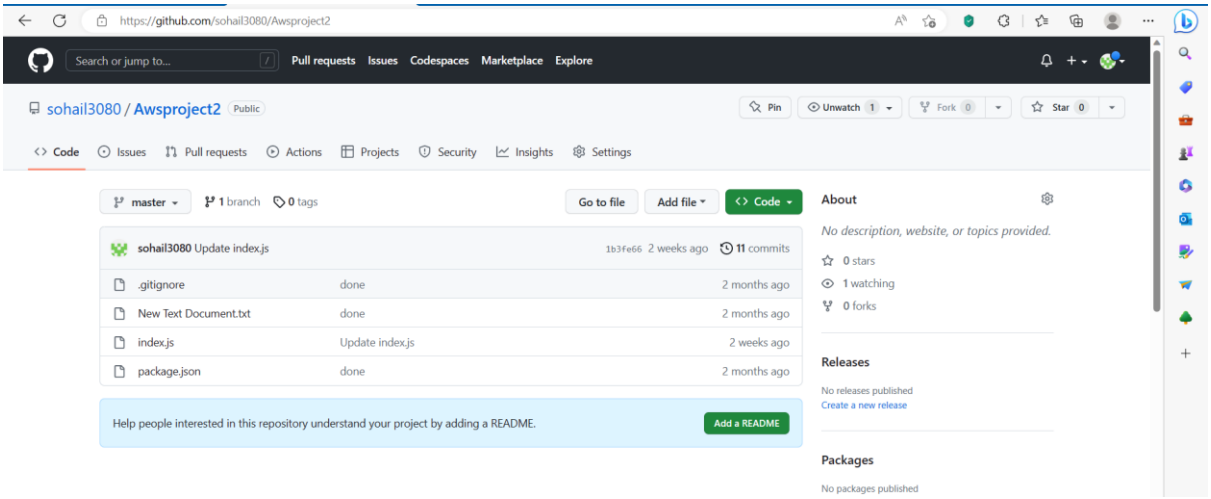
ASSIGNMENT 11

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

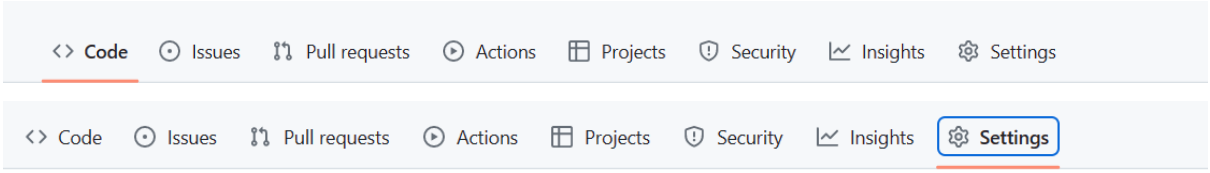
1. Sign in to your GitHub account.

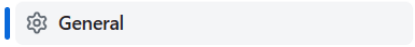


2. Open the Repository which you want to use and make sure it is public.



3. If the repository is not public, then go to Settings.

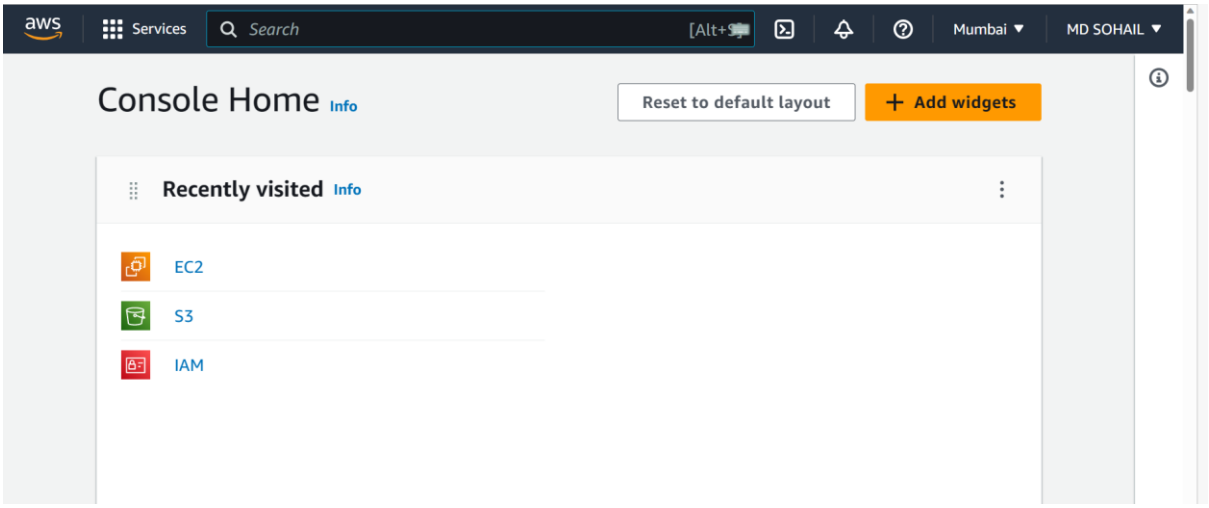




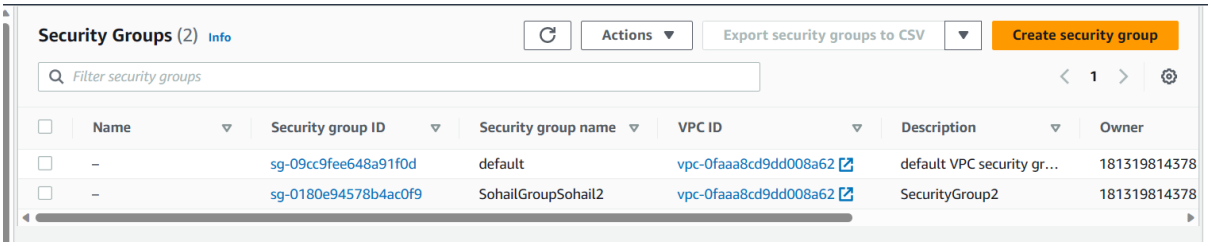
General

Next, scroll to the bottom in the Danger Zone. Click on Change visibility → Change to Public→I want to make this repository Public→I have read and understand these effects→Make this repository public. At last, give the Password. [My repository is already public so I have not followed these steps]

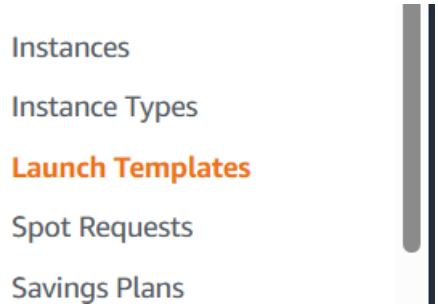
4. **Sign in** to your AWS account.

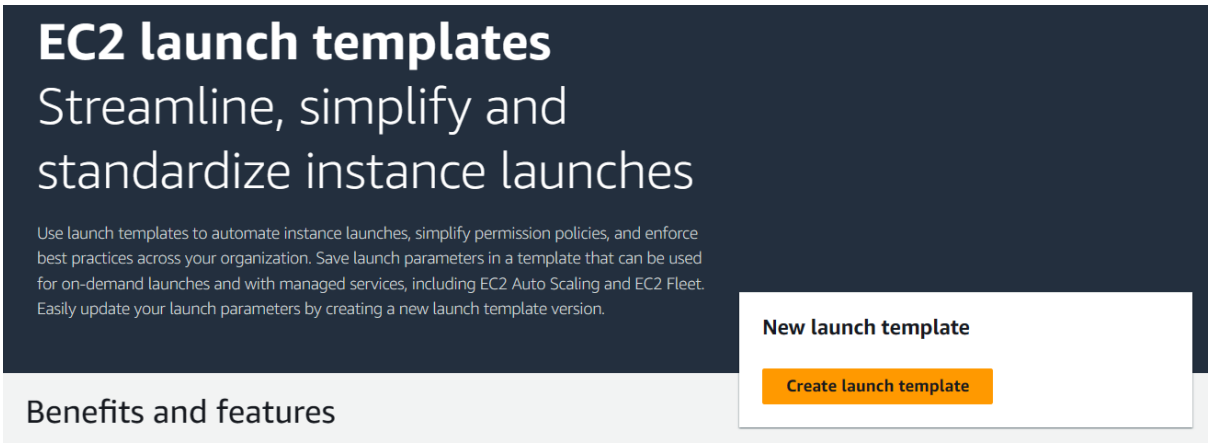


5. Go to EC2. Make sure you have already created a Security Group (i.e security group other than the default one).

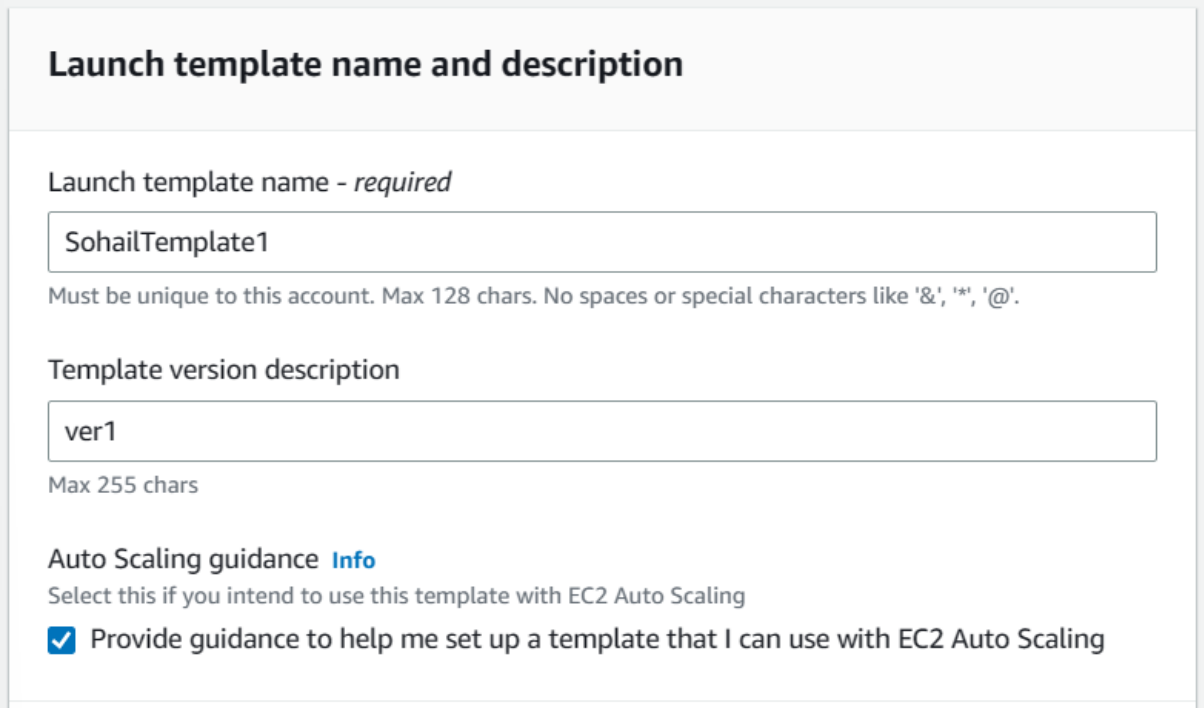


6. Next, Click on **Launch Templates** on the left sidebar. Next, Click on **Create launch template**.

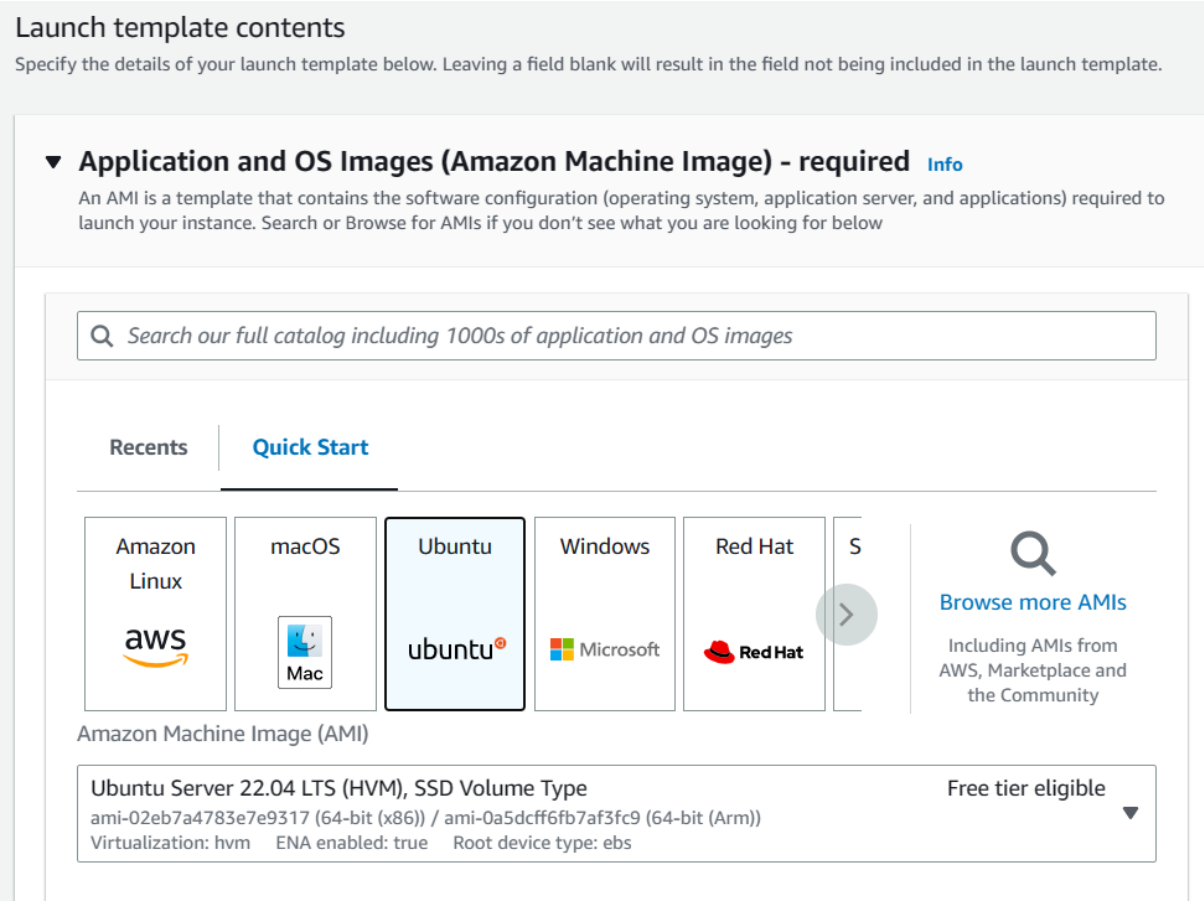




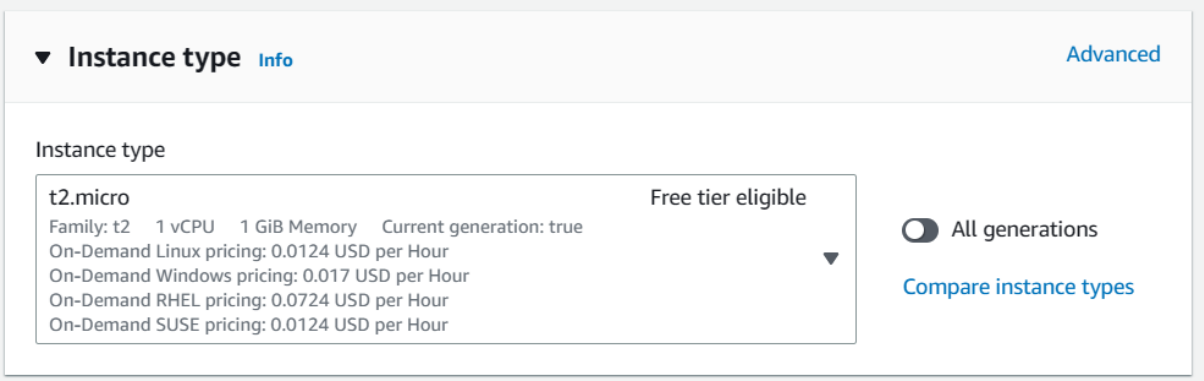
7. Now, Enter the **Launch template name** and **Template version description**. Also, you may select Provide guidance box.



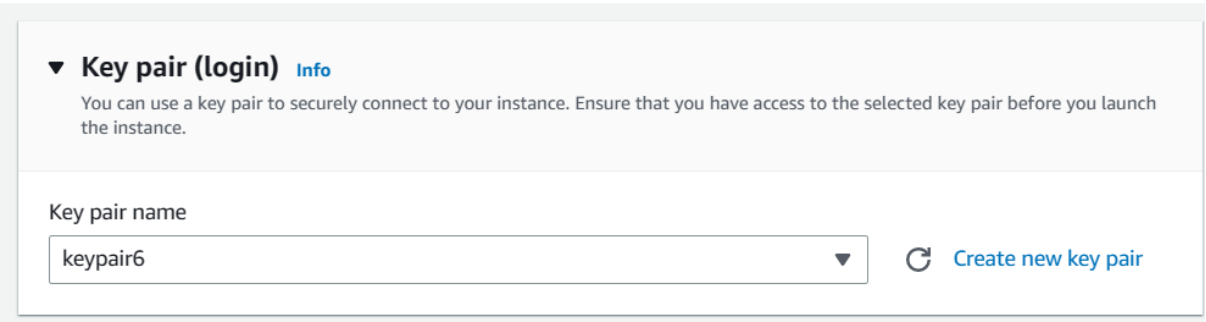
8. Next, within the Launch Template Contents, Click on Quick Start tab. Within the Quick start tab, select Ubuntu.



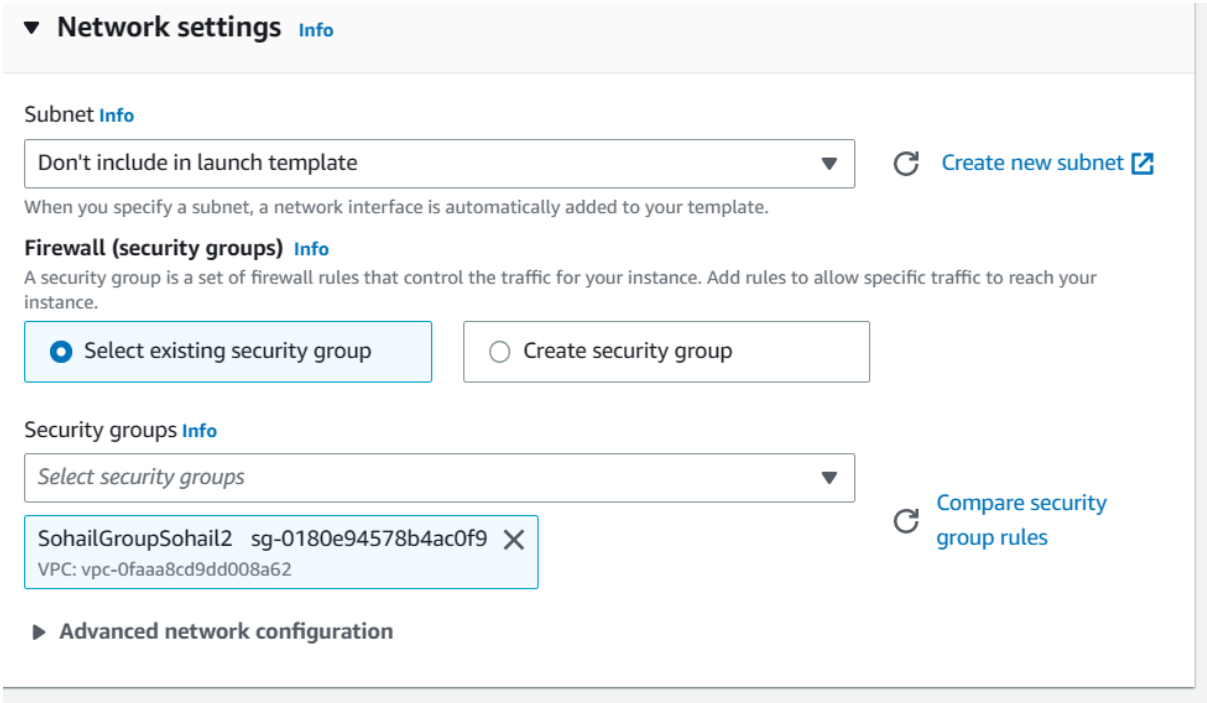
9. Select the Instance type as t2.micro.



10. Select a key pair which you have already created. If not, create a new one.



11. Within the Network Settings, Click on **Select existing security group**.



12. Next, Click on Advance details and scroll down to the bottom of it until you see the User data field and write the following commands in there.

User data - *optional* **Info**

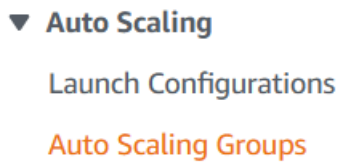
Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/sohail3080/Awsproject2.git
cd Awsproject2
npm install
node index.js
```

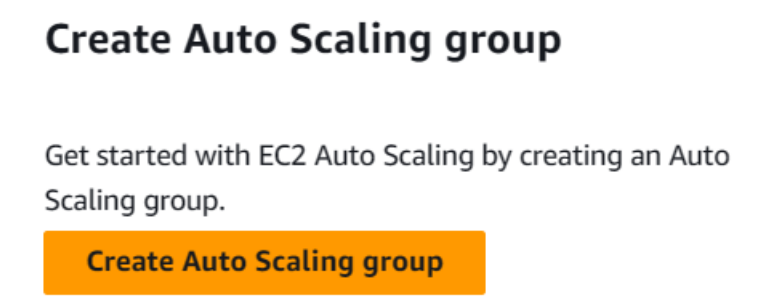
13. Click on Create Launch Template.

Template was successfully created.

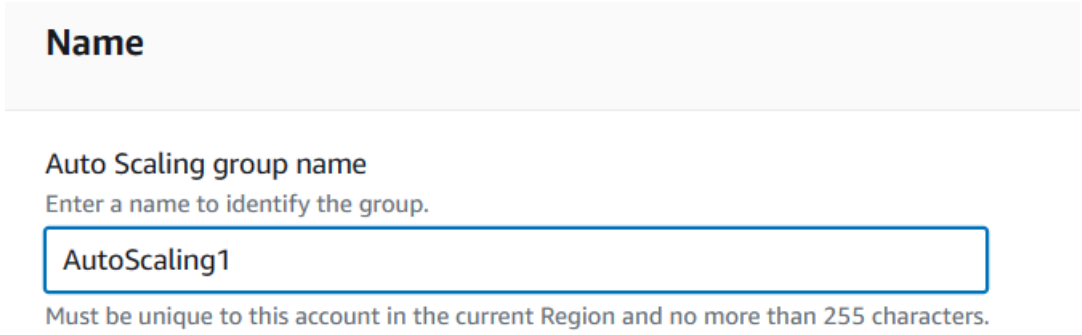
14. Next, Go to EC2 and in the left sidebar, Click on Auto Scaling Groups.



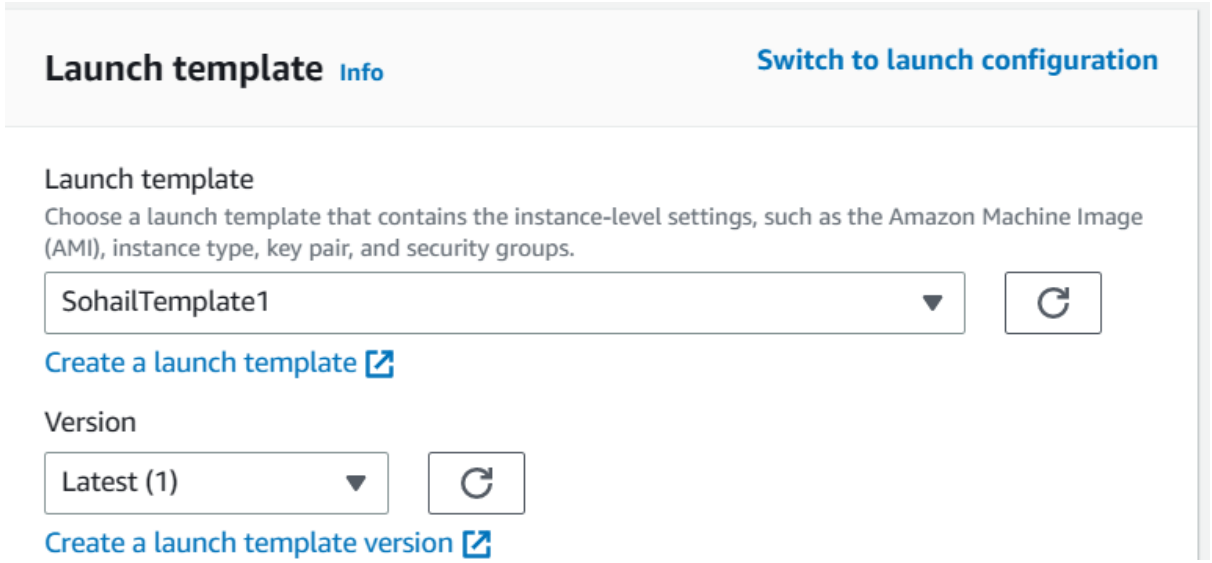
15. Now, Click on **Create Auto Scaling group**.



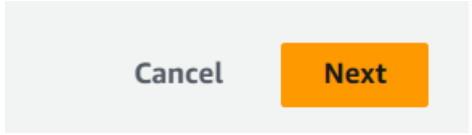
16. Give a Auto Scaling group name.



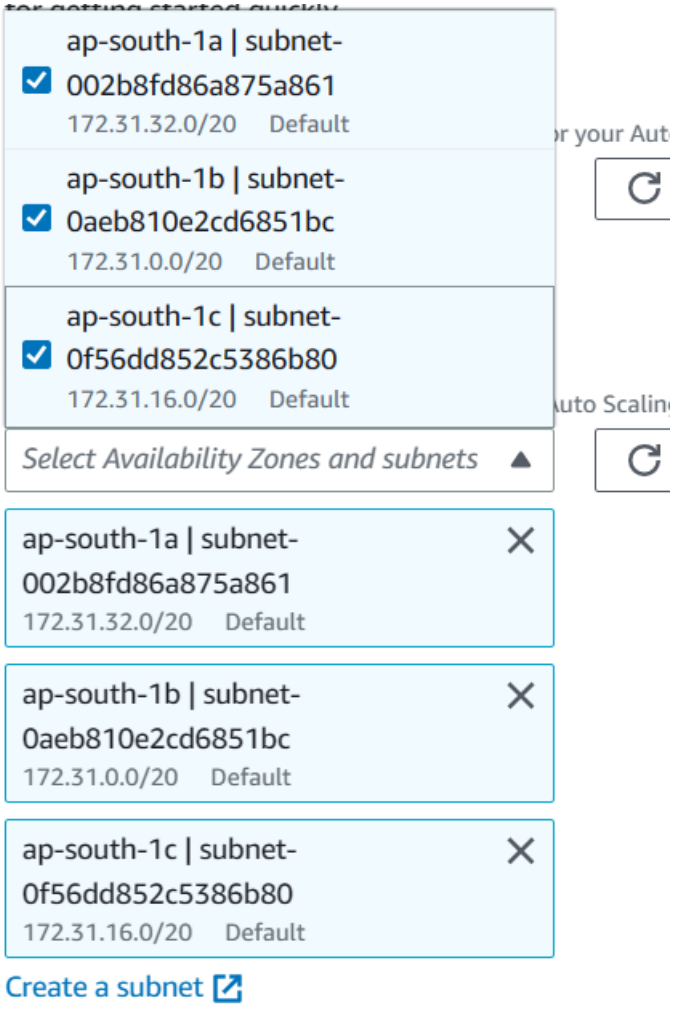
17. Within the Launch Template field, select the template you have created. Also select the Version to Lastest(1).



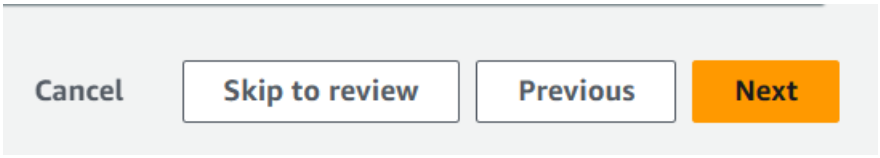
Then Click on Next.



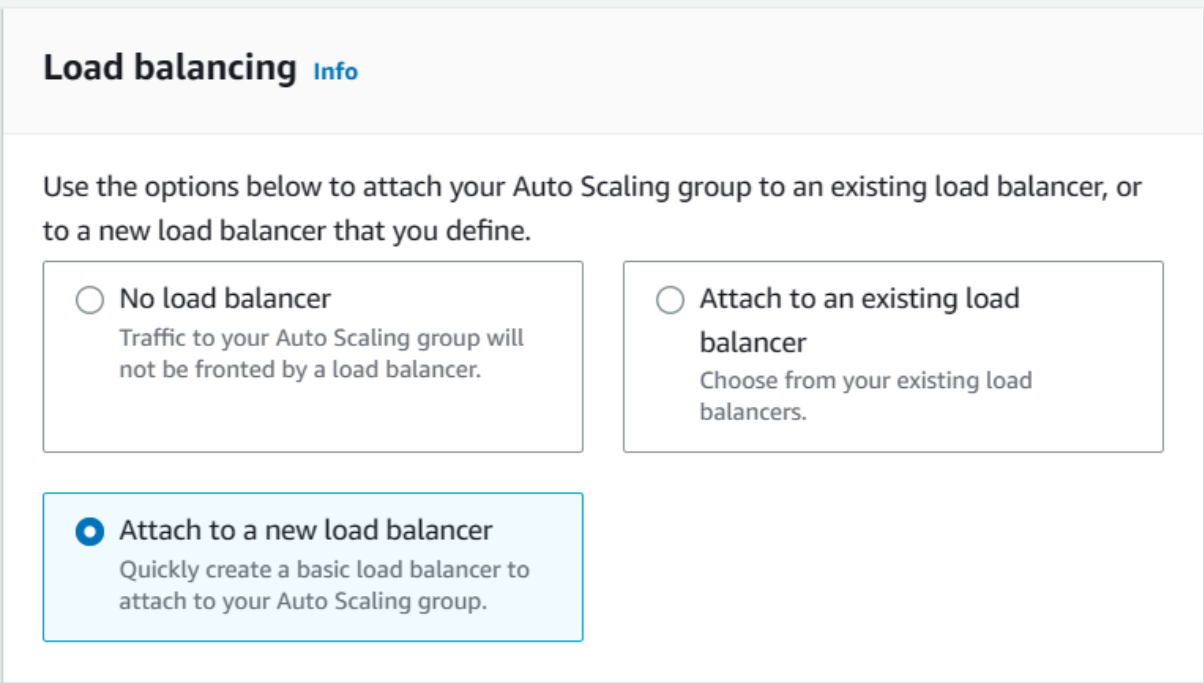
18. Now, within the Network field. Choose all Availability Zones an subnets.



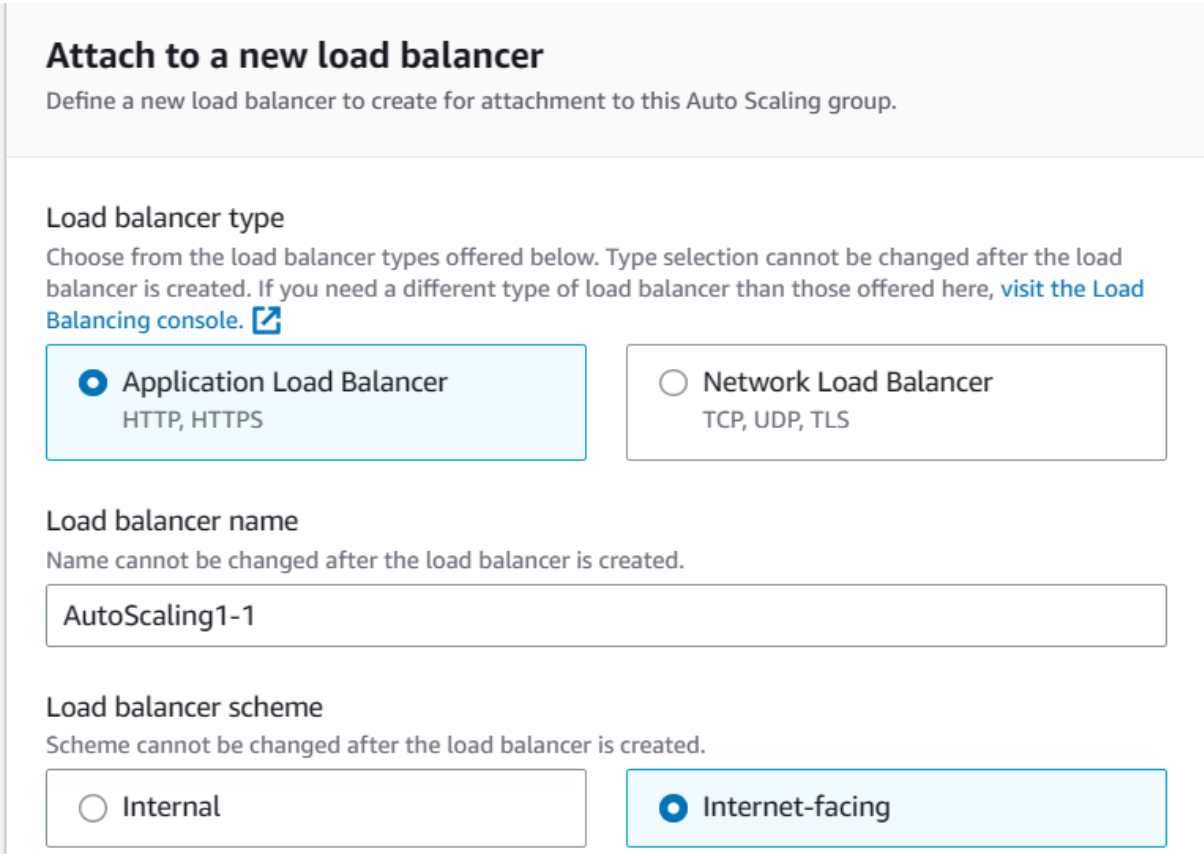
Then, Click on Next.



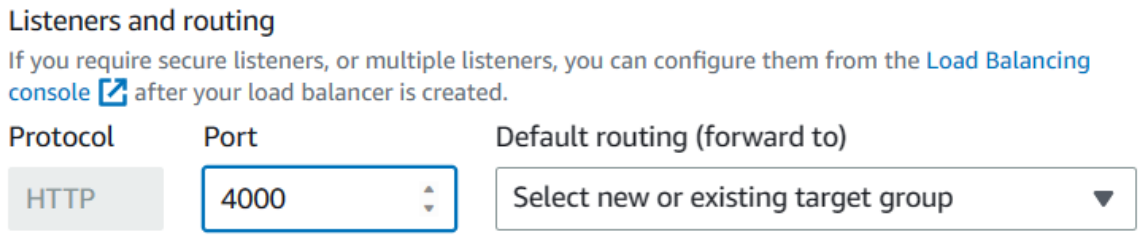
19. Next, within the Load balancing field, select **Attach to a new load balancer**.



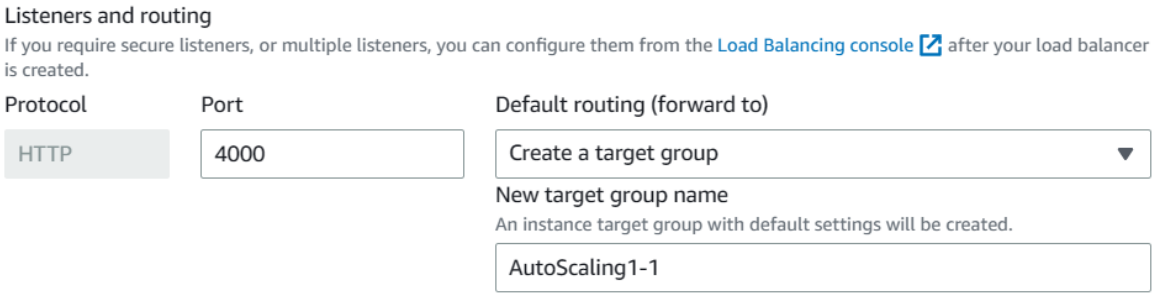
20. Select the Load balancer type as Application Load Balancer(default). Enter the Load balancer name (default AutoScaling1-1). Select the Load balancer scheme as Internet-facing.



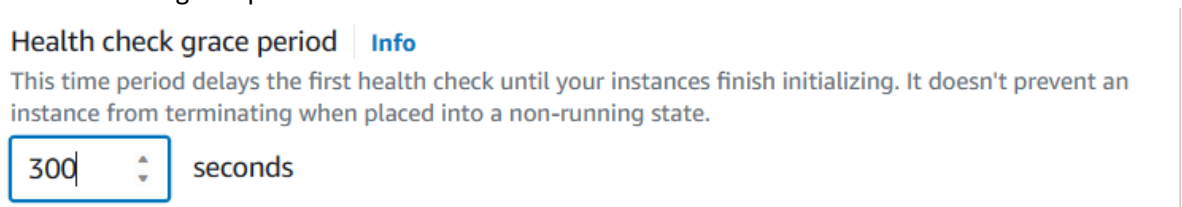
21. Enter the Port number of Listeners and routing as 4000. [According to our project we have taken 4000]



22. Now, Click on Select a new or existing target group → Create a target group. New target group name should be AutoScaling1-1 according to our project.

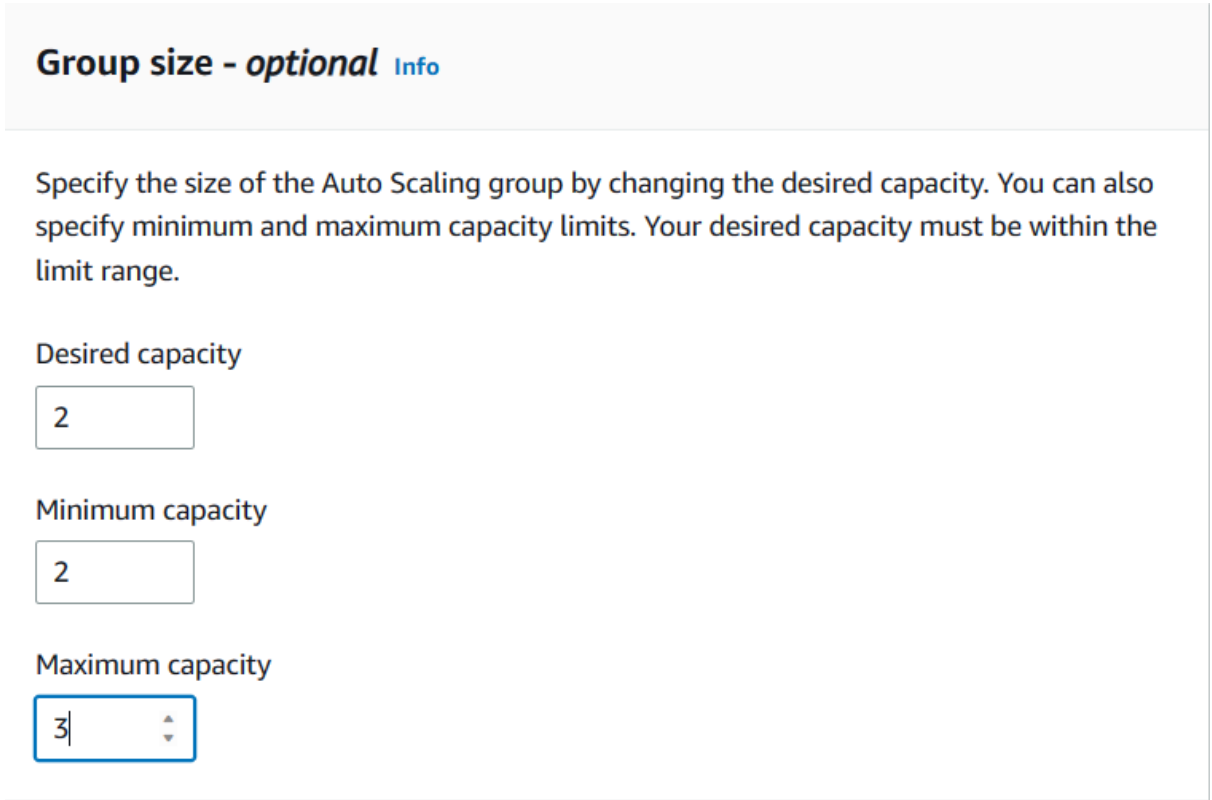


23. Set Health check grace period to 300 seconds.

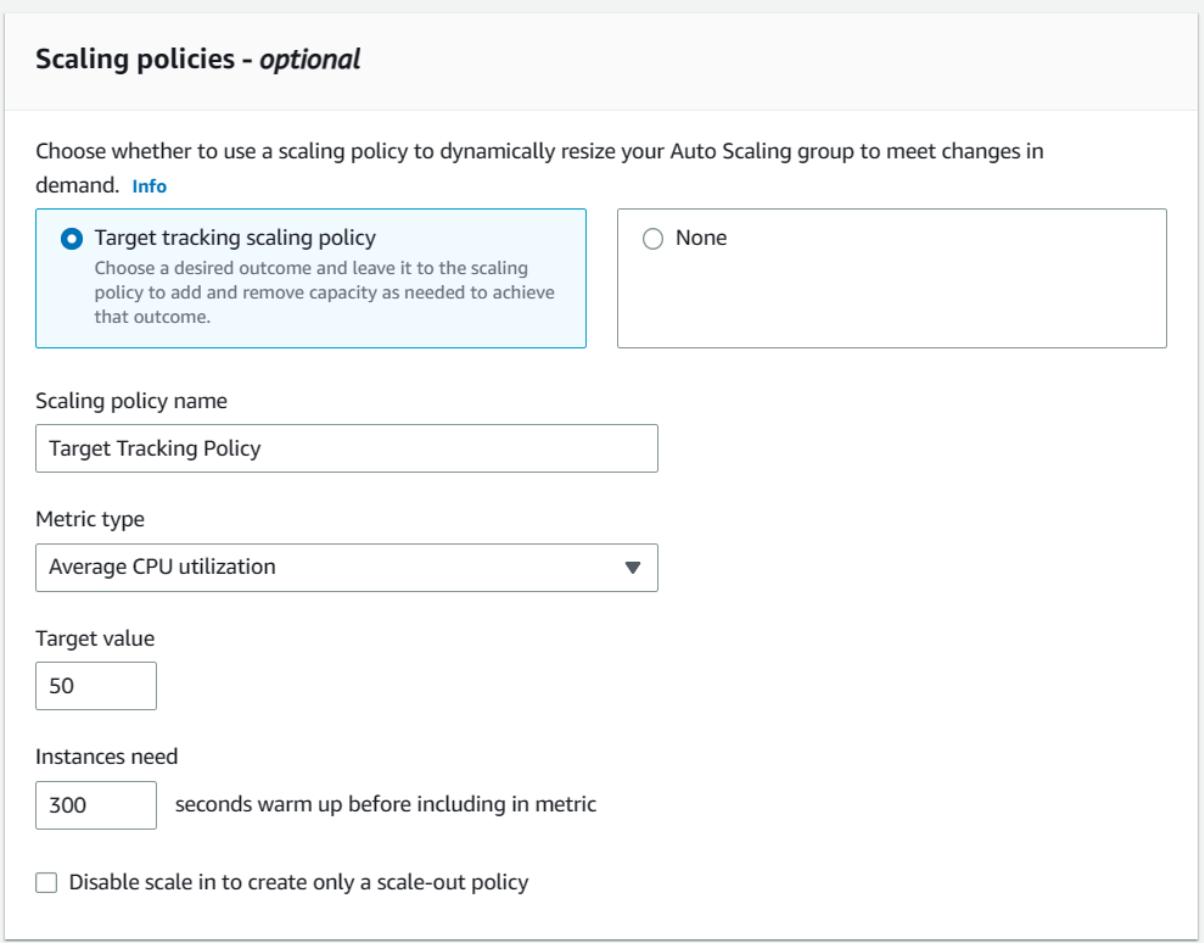


Next, Click on Next.

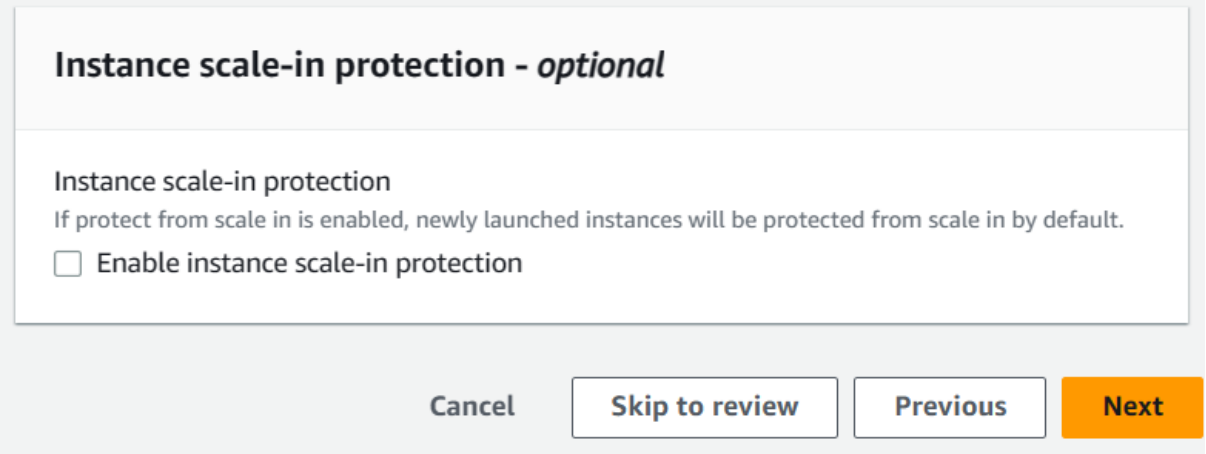
24. In the Group size field.
Set Desired capacity to 2, Minimum capacity to 2, Maximum capacity to 3.



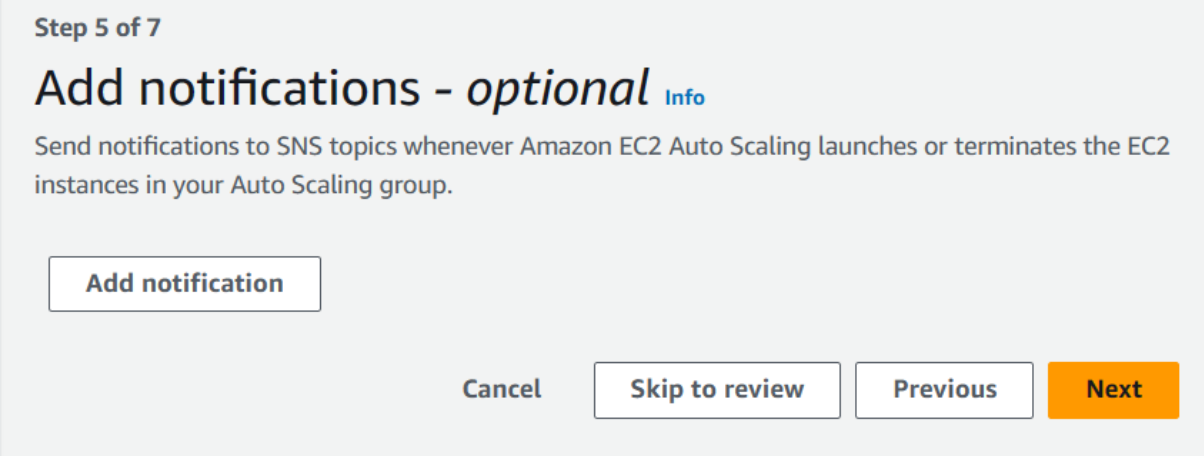
25. Now, Click on **Target tracking scaling policy**. Set the Target value to **50** and instance need to **300** seconds warm up before including in metric.



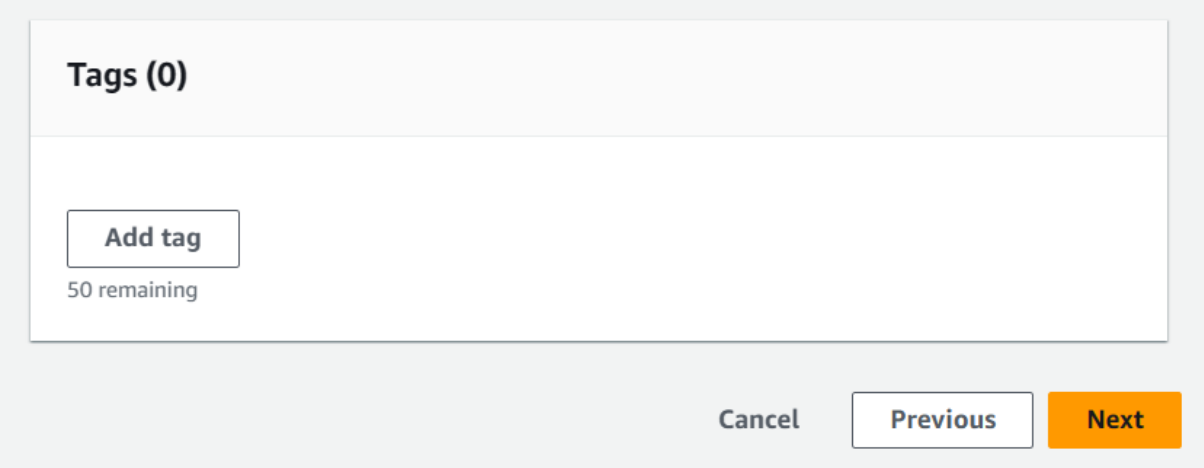
Click on Next.



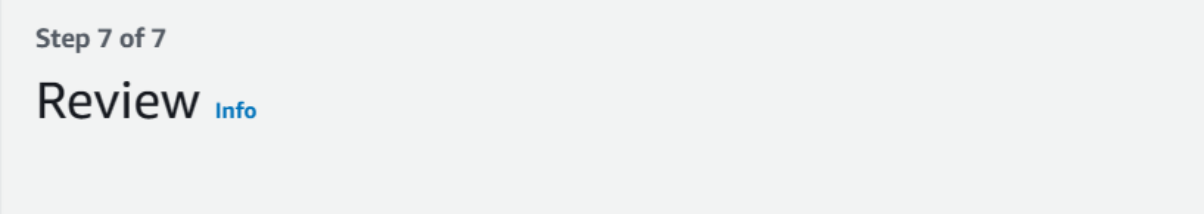
26. Click on Next.



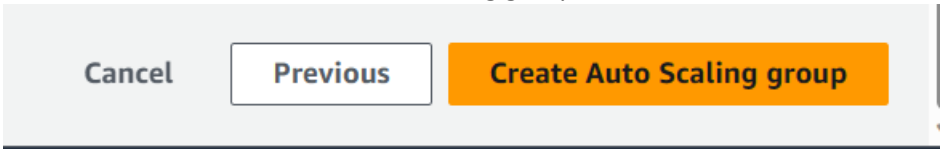
27. Click on Next.



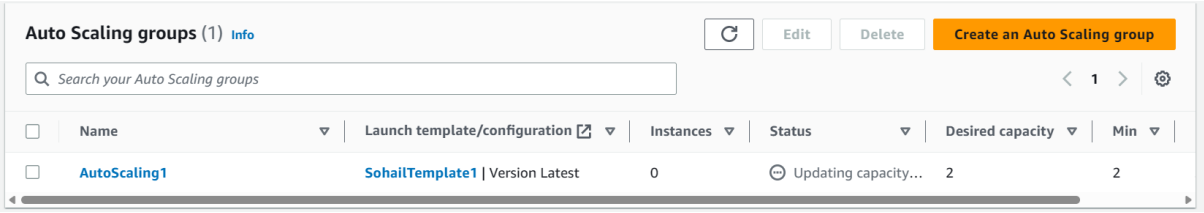
28. Scroll down on the Review page.



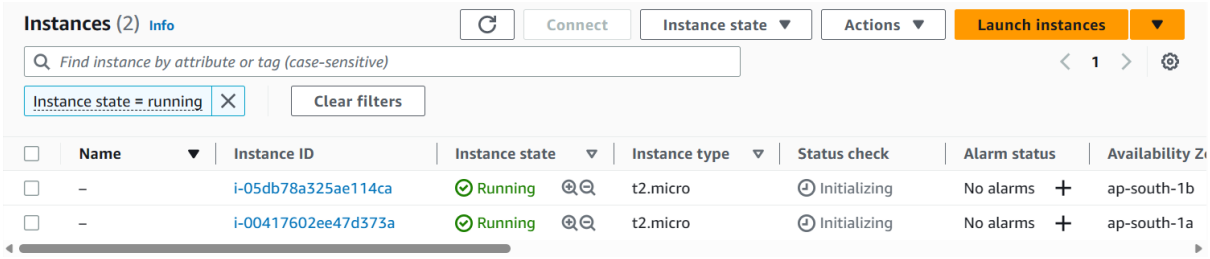
At the bottom, Select Create Auto Scaling group.



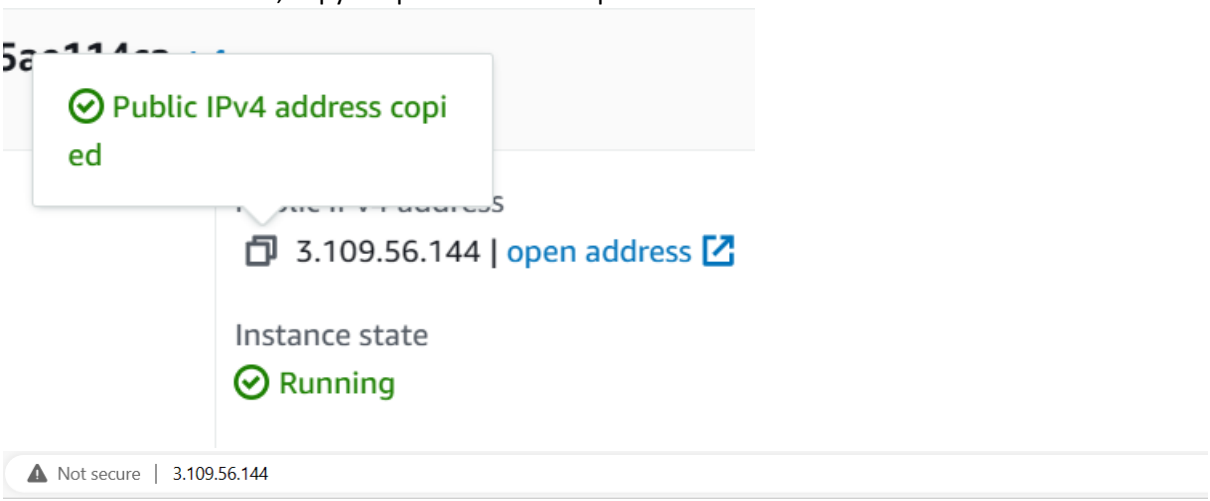
29. You can see the Auto Scaling Group was created successfully.



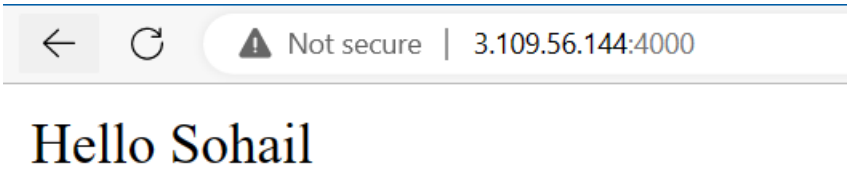
30. Go to EC2 Instances. You can see two instances running. [Wait, It may take some time]



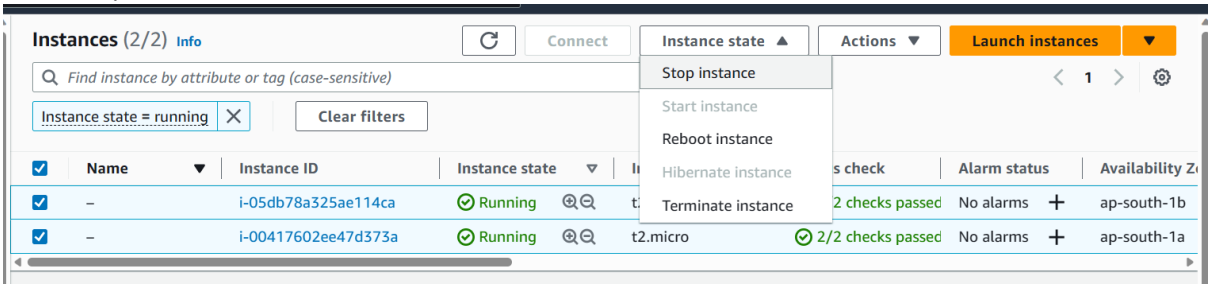
31. Go to the first instance, copy its Ipv4 address and paste it in the URL bar of the browser.



Add the Port Number 4000 at the end of the URL as follows.



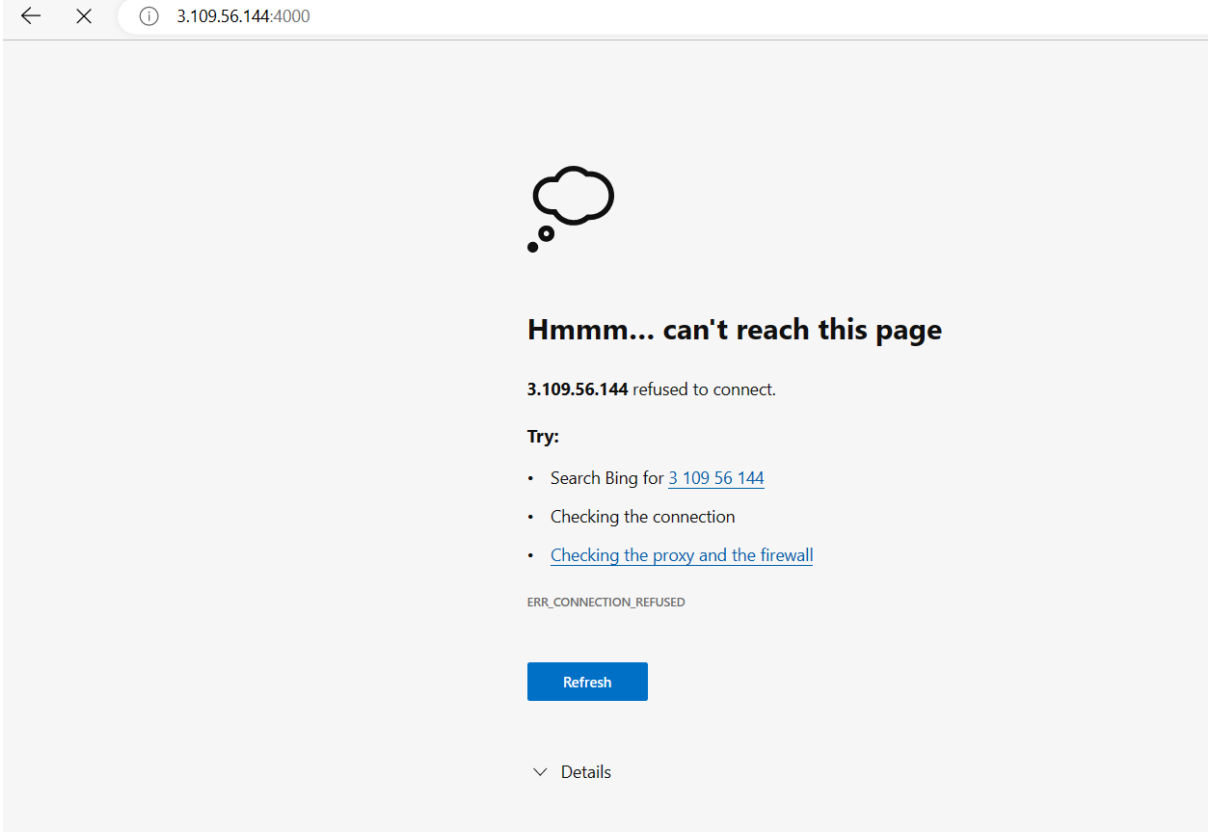
32. Now, Stop both the instances for sometime.



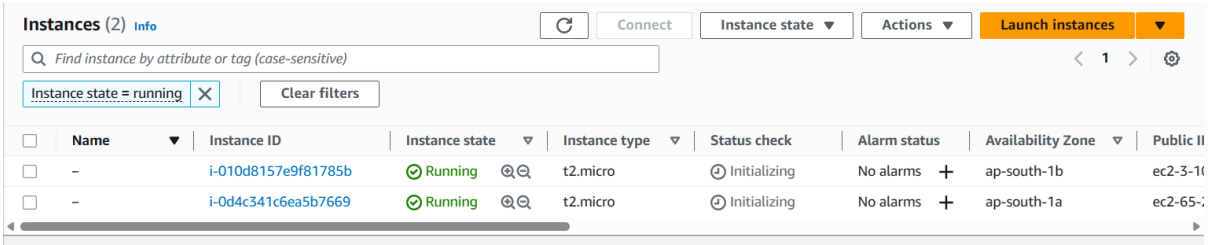
Click on Stop.

instances, choose the *Stop* button below.

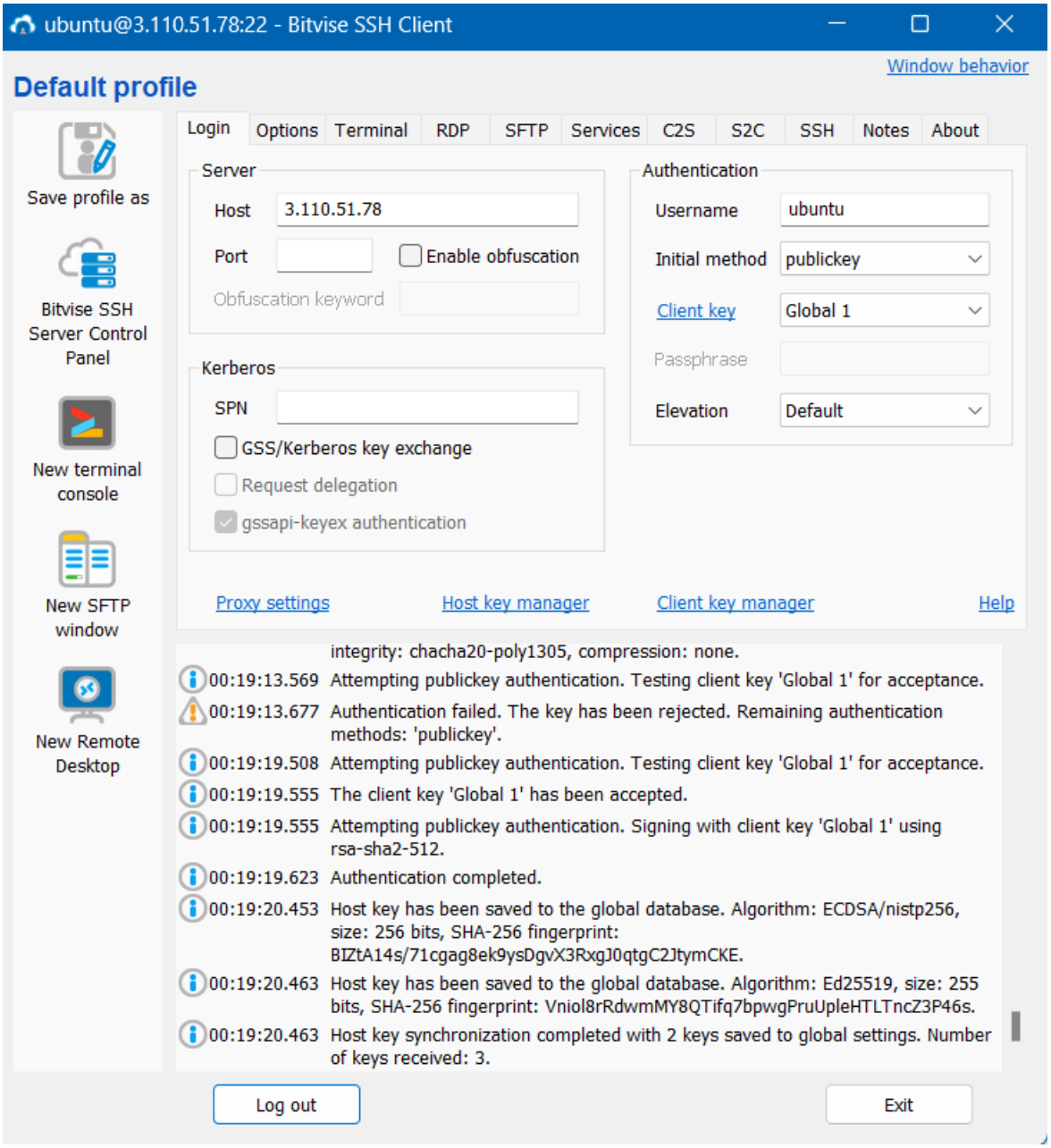
33. As the servers have stopped, our page will not show anymore.



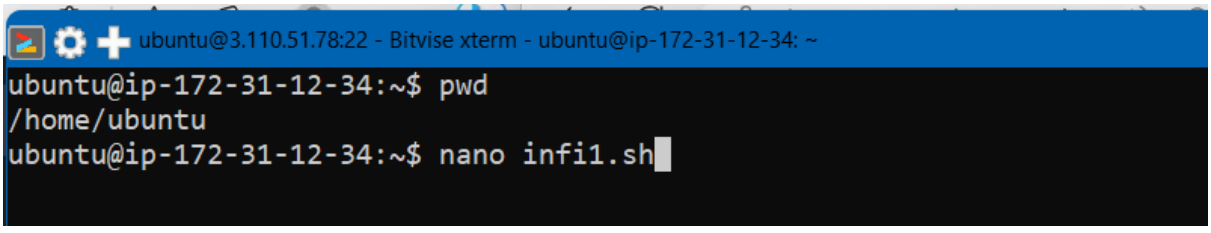
34. After some time, new instances got automatically created.



35. Click on the first instance and connect It to Bitvise SSH Client.

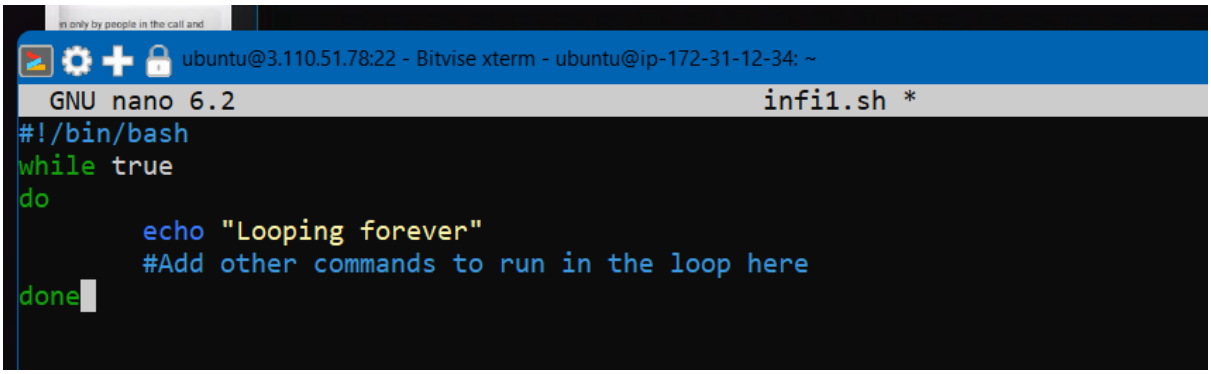


36. Open New terminal console.
    Type the following commands respectively.
    a) pwd
    b) nano infi1.sh



    c) #!/bin/bash
    while true
    do
        echo "Looping forever"
        #Add other commands to run in the loop here

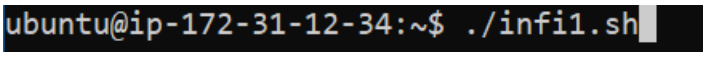
done



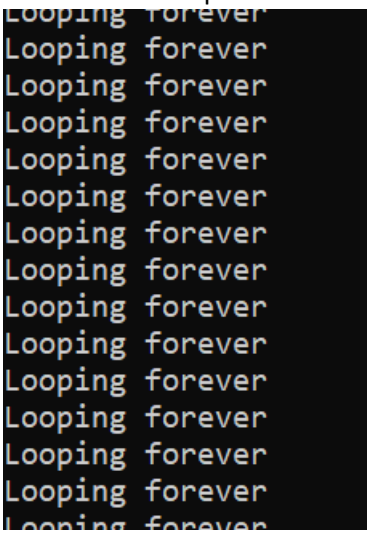
Now Press Ctrl+X, Ctrl+Y and Enter respectively.

d)chmod +x infi1.sh

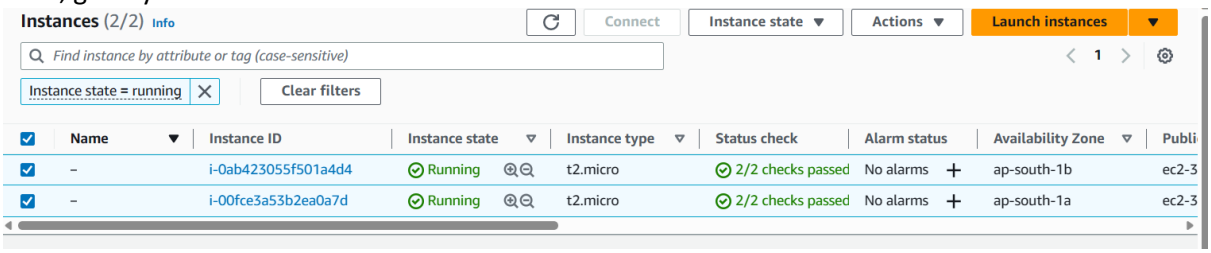
e) ./infi1.sh



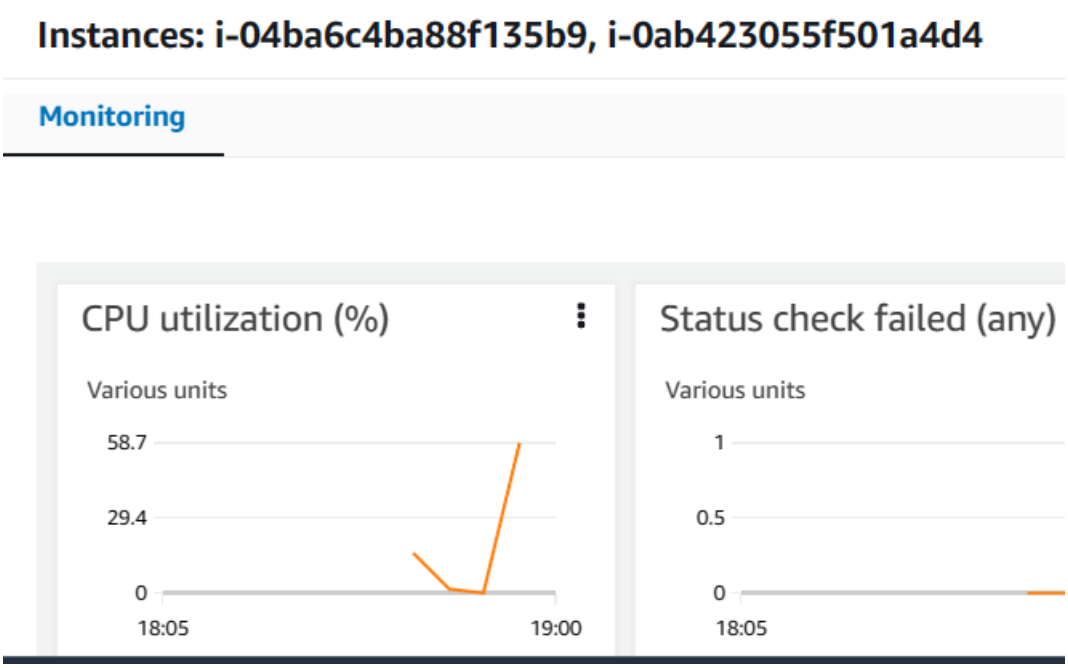
Runs Infinite Loop.



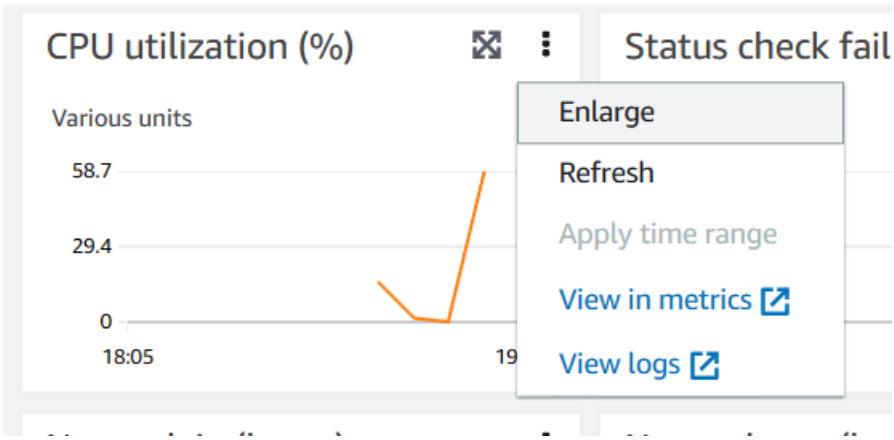
37. Now, go to your Instances in AWS and select both of them.



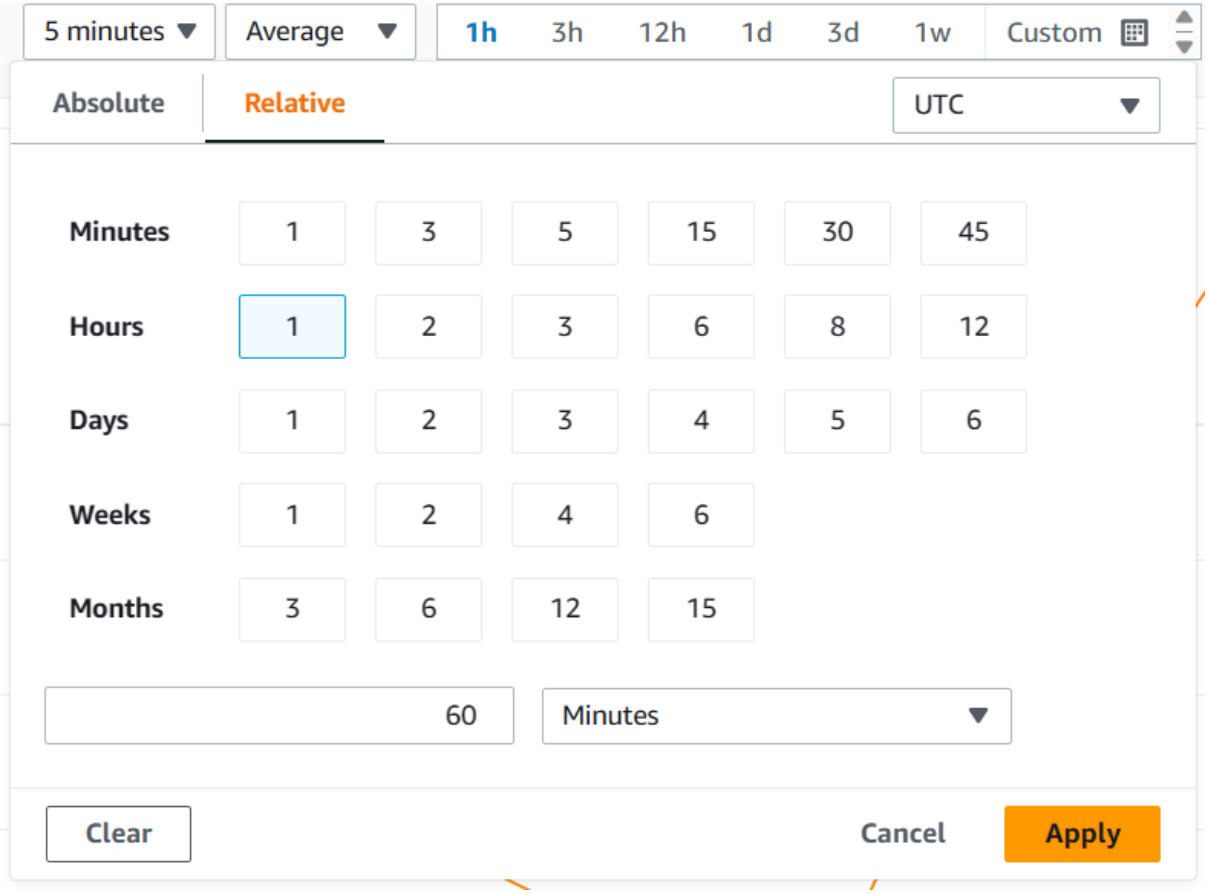
38. Click on Monitoring.



And in CPU utilization (%), click on Enlarge.

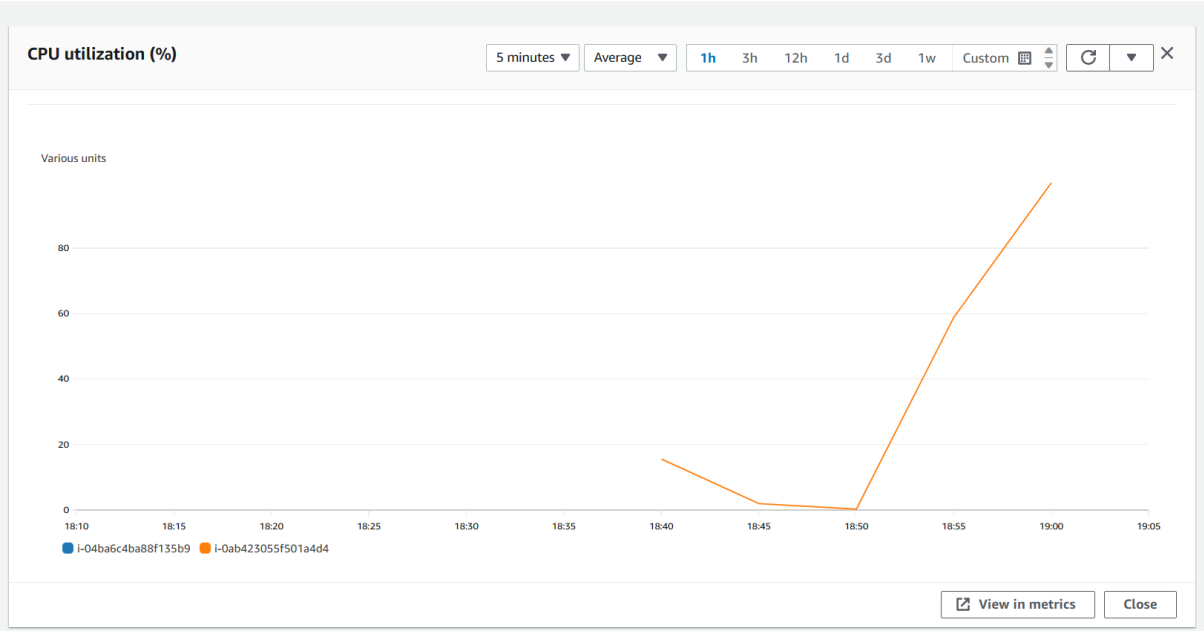


39. Click on Custom→Select UTC on the right upper side.



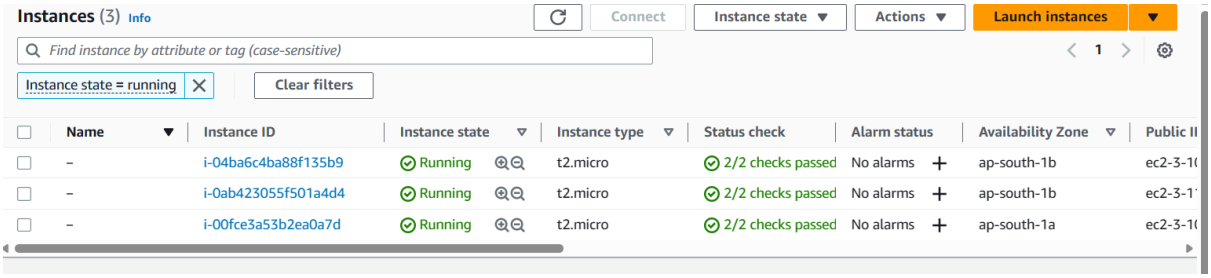
Click on **Apply**.

40. As you can see, according to what we have set, the CPU utilization (%) is more than 50, so a new instance should be created.

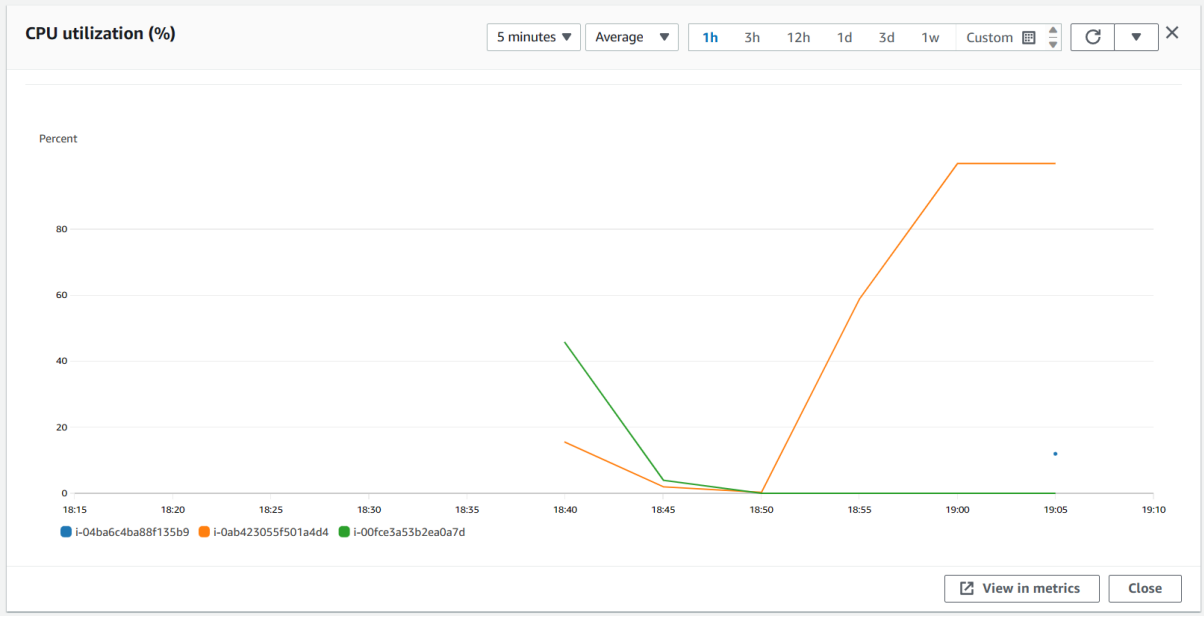


Let us check whether a new Instance is created or not.

41. As you can see, we have a new instance.

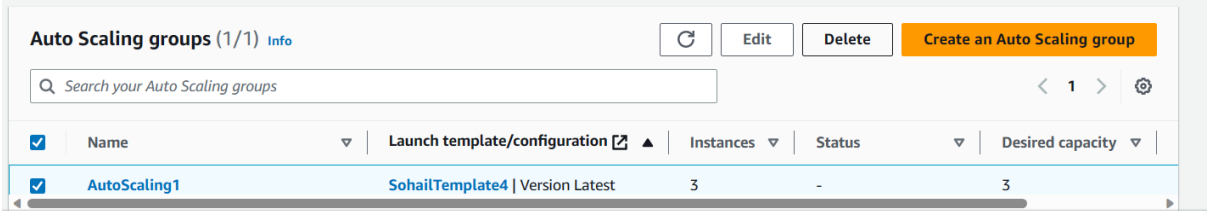


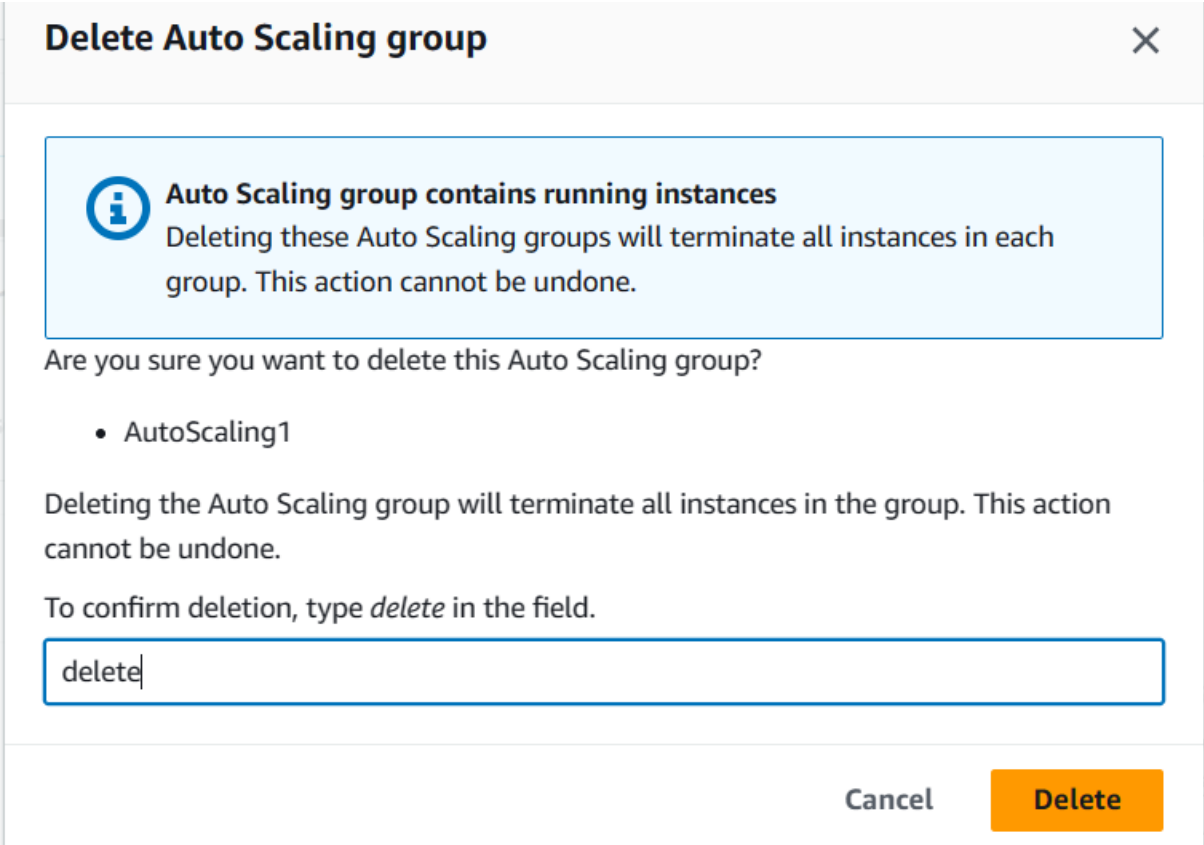
42. Now, select all the instances and enlarge the CPU utilization. Here you can see the graph of them.



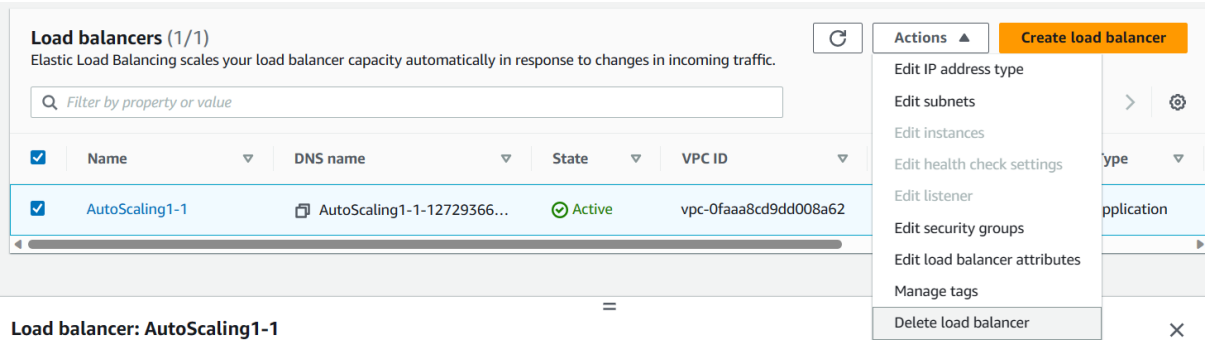
Hence, When the load was above the set value, it was balanced out by creating a new Instance.

43. To close
    a) first, we have to remove Auto Scaling group.

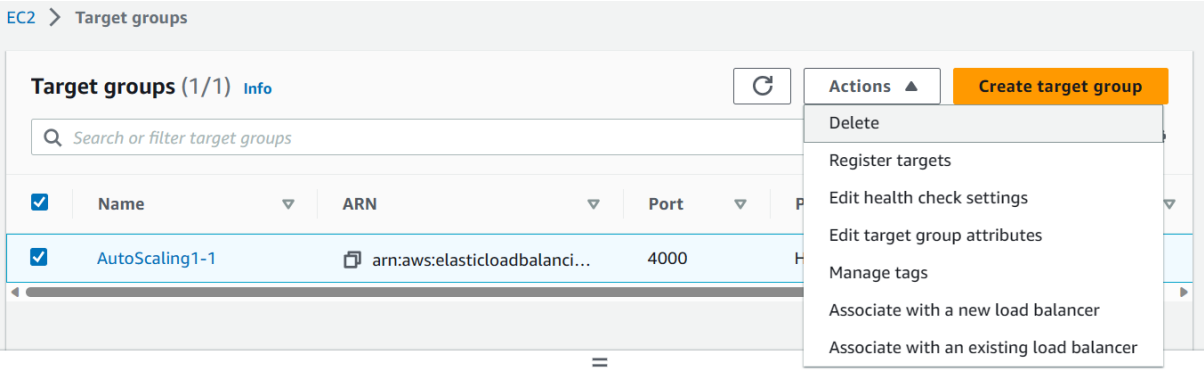




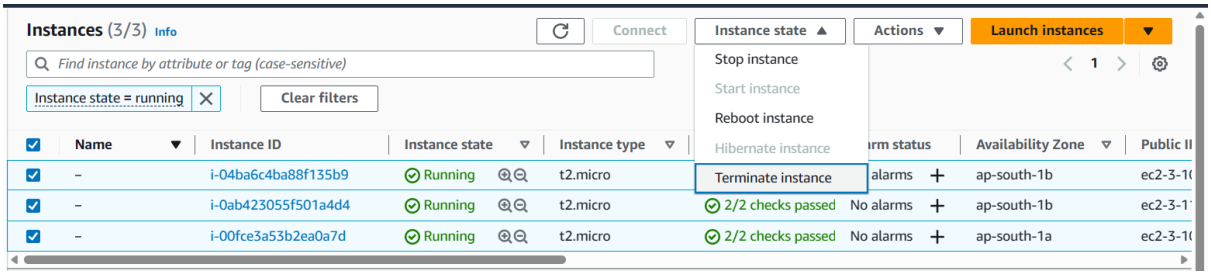
b) Then Load Balancer

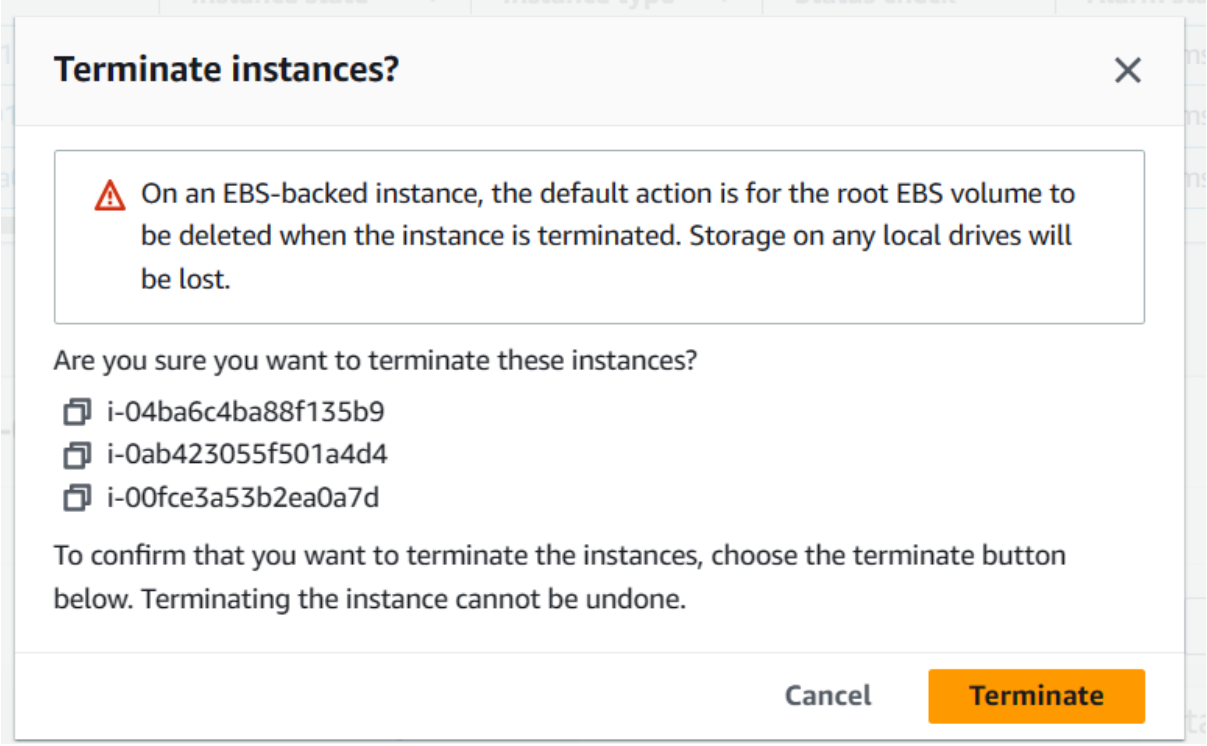


c) Then Target Group.



d) And at last EC2.


Instances (3/3) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
Instance state = running
Clear filters
Stop instance
Start instance
Reboot instance
Hibernate instance
Terminate instance
Name
Instance ID
Instance state
Instance type
Availability Zone
i-04ba6c4ba88f135b9
Running
t2.micro
2/2 checks passed
No alarms
ap-south-1b
i-0ab423055f501a4d4
Running
t2.micro
2/2 checks passed
No alarms
ap-south-1b
i-00fce3a53b2ea0a7d
Running
t2.micro
2/2 checks passed
No alarms
ap-south-1a


Terminate instances?
On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.
Are you sure you want to terminate these instances?
i-04ba6c4ba88f135b9
i-0ab423055f501a4d4
i-00fce3a53b2ea0a7d
To confirm that you want to terminate the instances, choose the terminate button below. Terminating the instance cannot be undone.
Cancel
Terminate