Part 1: EC2 with ELB and ASG

Objective: Learn how to create a scalable and highly available web application environment using Amazon EC2 instances, ELB, and ASG.

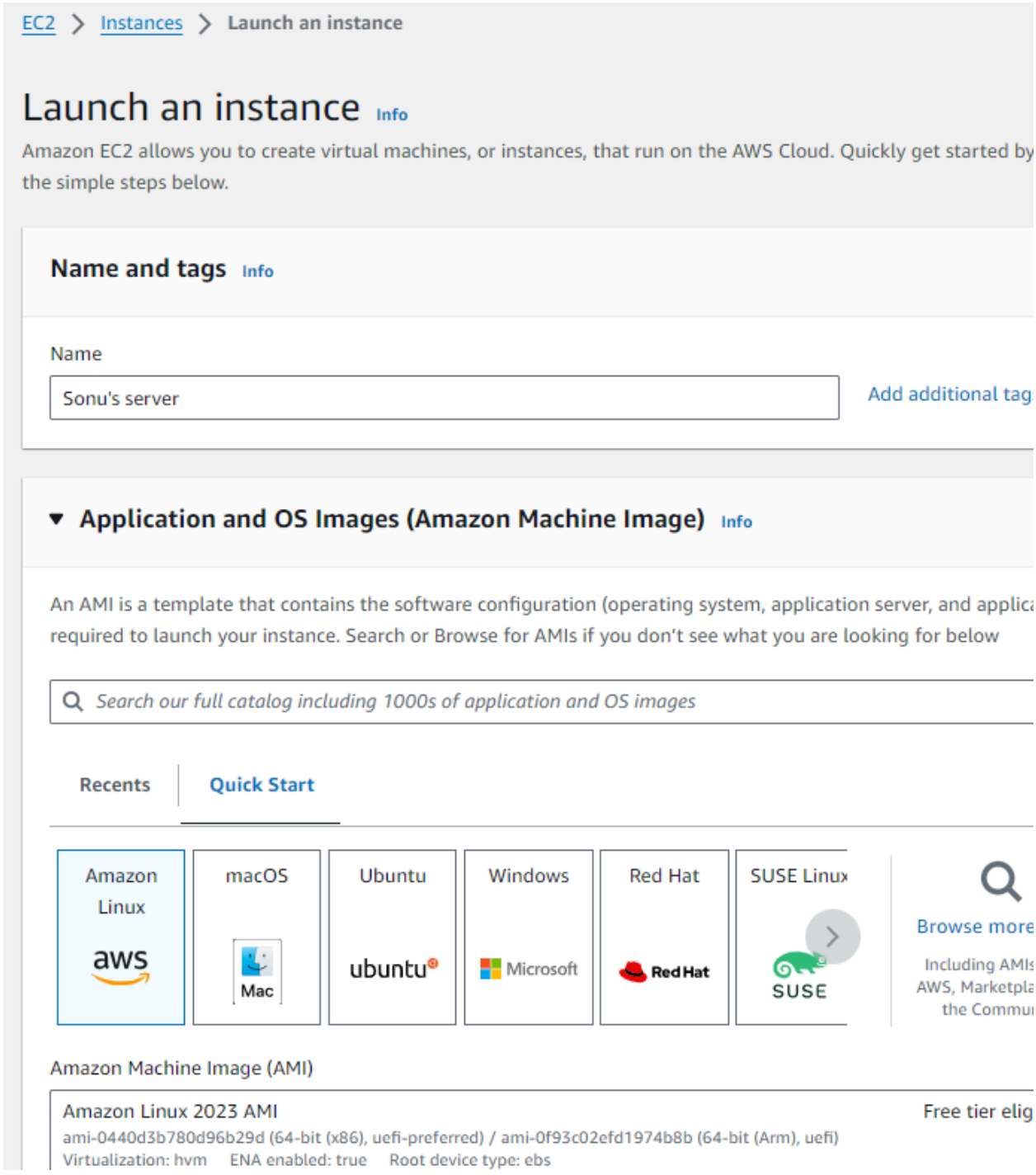
Approach:

1. **Launch EC2 Instances**: Start by launching two or more EC2 instances. These instances will run a simple web application (e.g., a "Hello World" page or any basic web service).
2. **Configure Load Balancer**: Set up an Elastic Load Balancer (ELB) to distribute incoming web traffic across your EC2 instances. This step ensures high availability and fault tolerance.
3. **Set Up Auto Scaling Group (ASG)**: Create an ASG that uses the launched EC2 instances. Configure ASG policies to automatically scale the number of instances up or down based on criteria like CPU usage or network traffic.
4. **Test Your Setup**: Simulate traffic to test the scaling policies and the load balancer. Observe how ASG adds or removes instances and how ELB distributes traffic.
5. **Verify Website Functionality**: Ensure that the website hosted on EC2 instances remains accessible and functional during scaling operations.

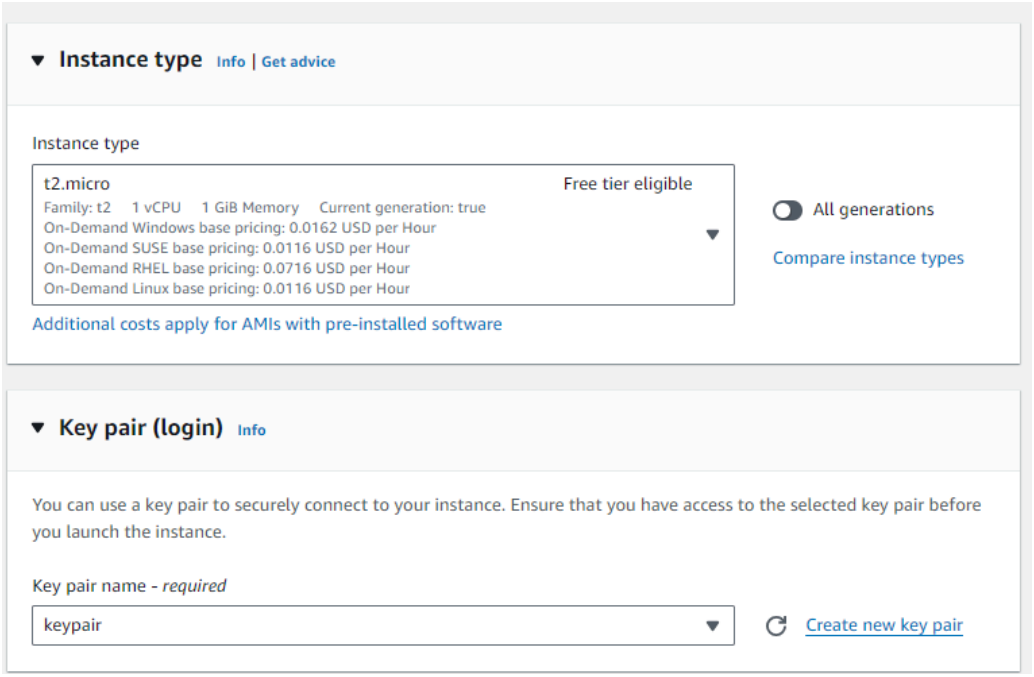
Goal: By the end of this lab, students will have a hands-on understanding of setting up a load-balanced and auto-scaled web application using AWS services.

Steps Involved:

1) **Launch EC2 instance**
   Provide the name of instance

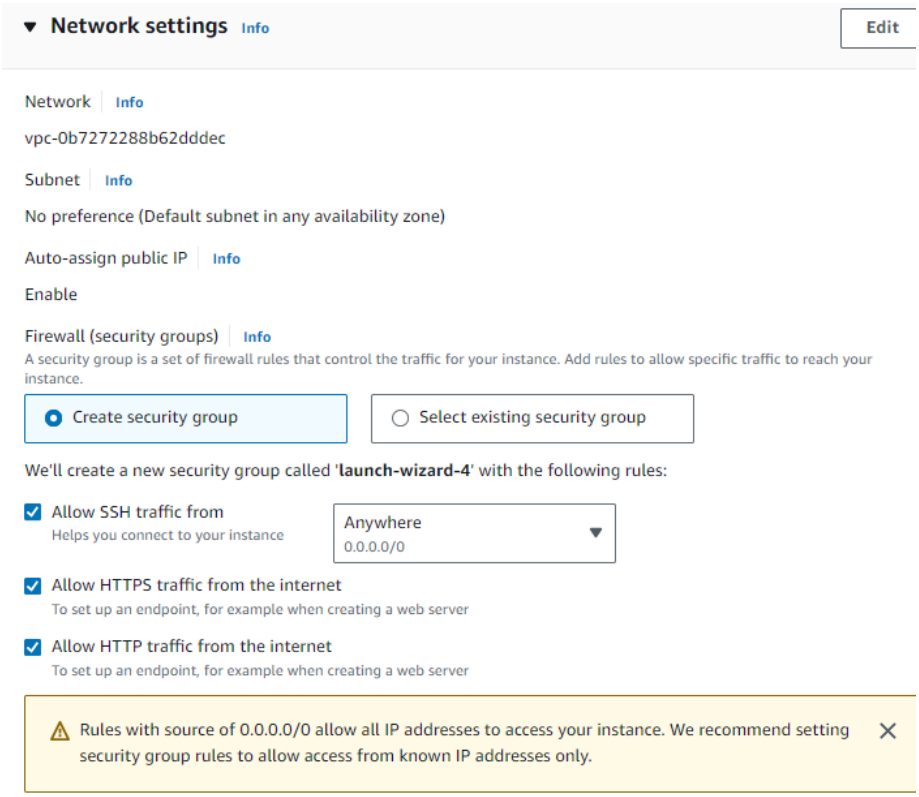


2) Select t2.micro instance type. Add new keypair or from an existing key pair.

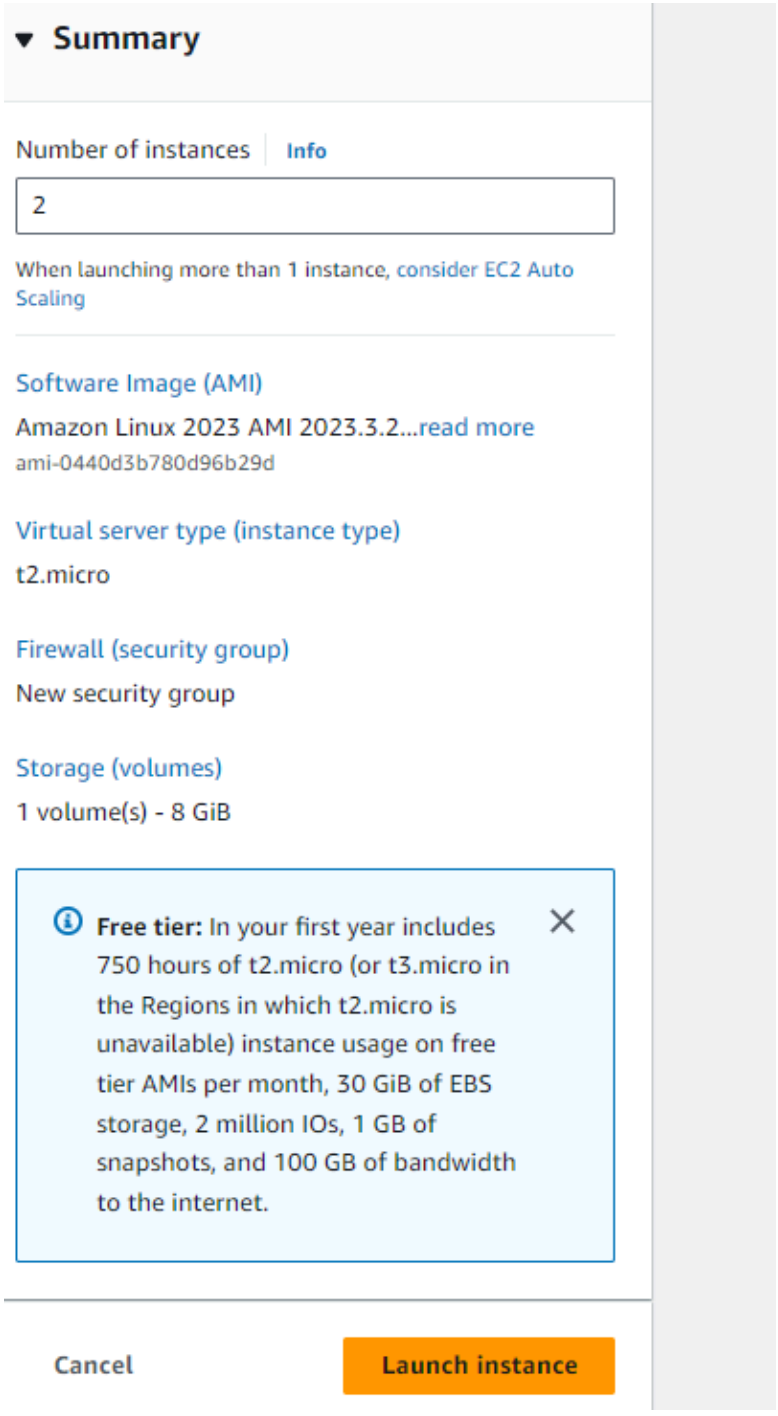


3) Network Settings

Security Group launch-wizard-14 is created allowing SSH, HTTPS and HTTP traffic from anywhere.



4) Two instances with same configuration are to be created.



5) At Advanced Detail section below script is attached and rest configuration are left as it is.

Upload a file with your user data or enter it in the field.

Choose file

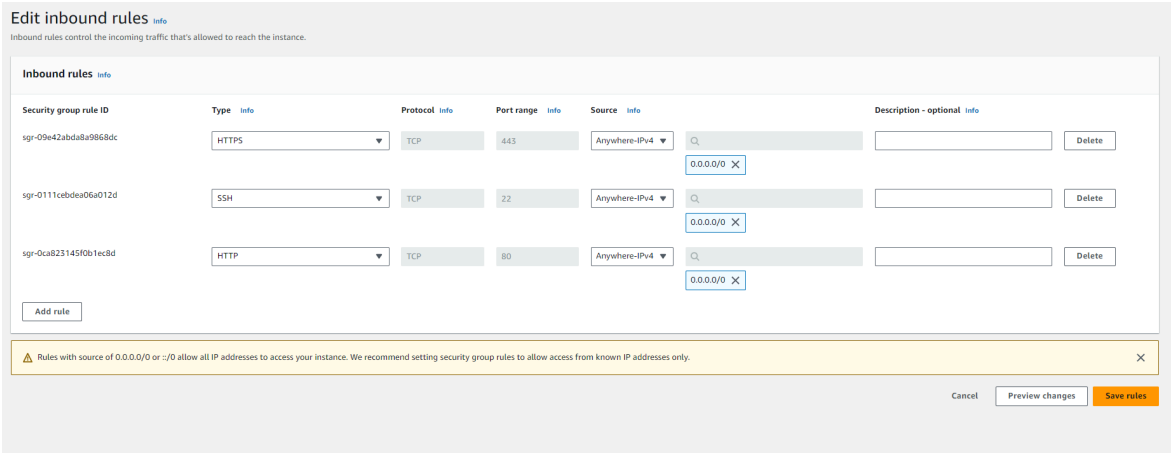
```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1> Hello from $(hostname -f)<a/h1>" > /var/www/html/index.html
S
```

☐ User data has already been base64 encoded

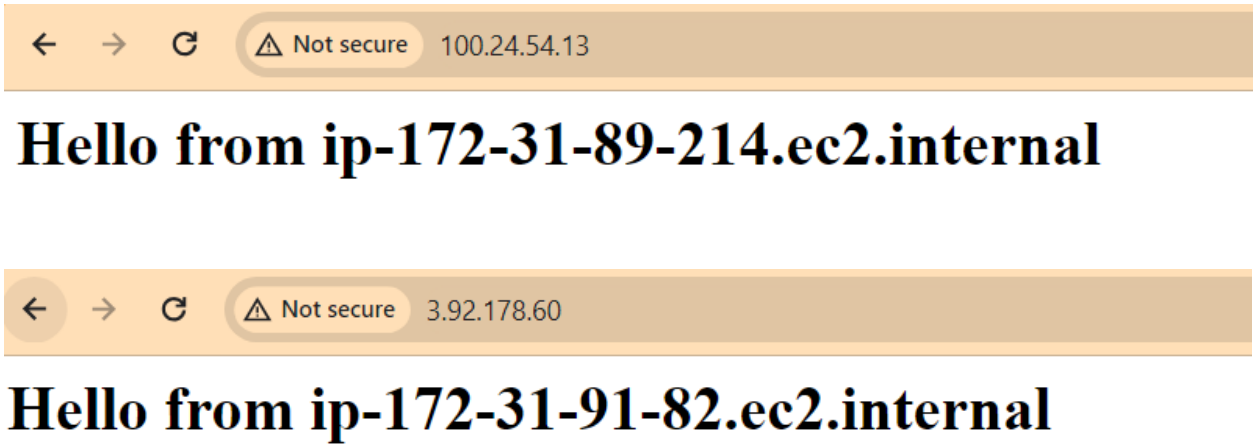
6) Click on “Launch Instance”

7) Successful instance creation

8) Update the created security group inbound rule as the ec2 instance is not working while trying to run from the Public IPv4 address of the created EC2 instance.

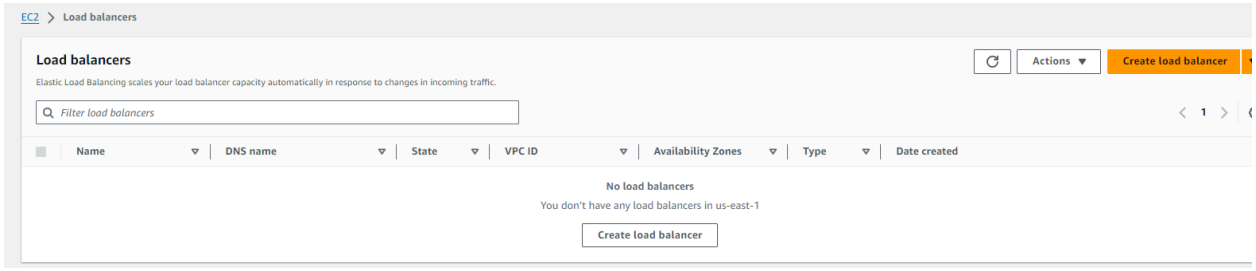


9) Both instances public IPv4 addresses worked in web browser successfully.

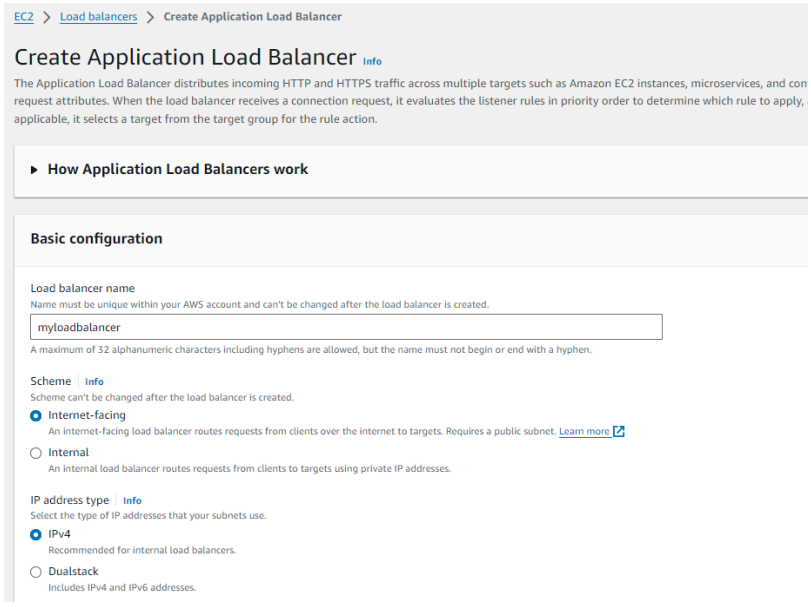


Configuring Load Balancer

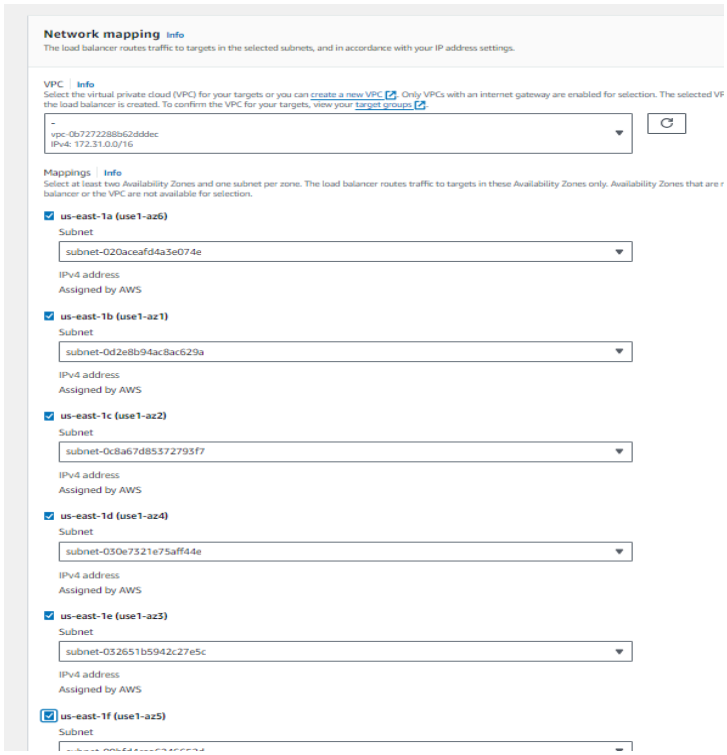
10) Creating Load Balancer



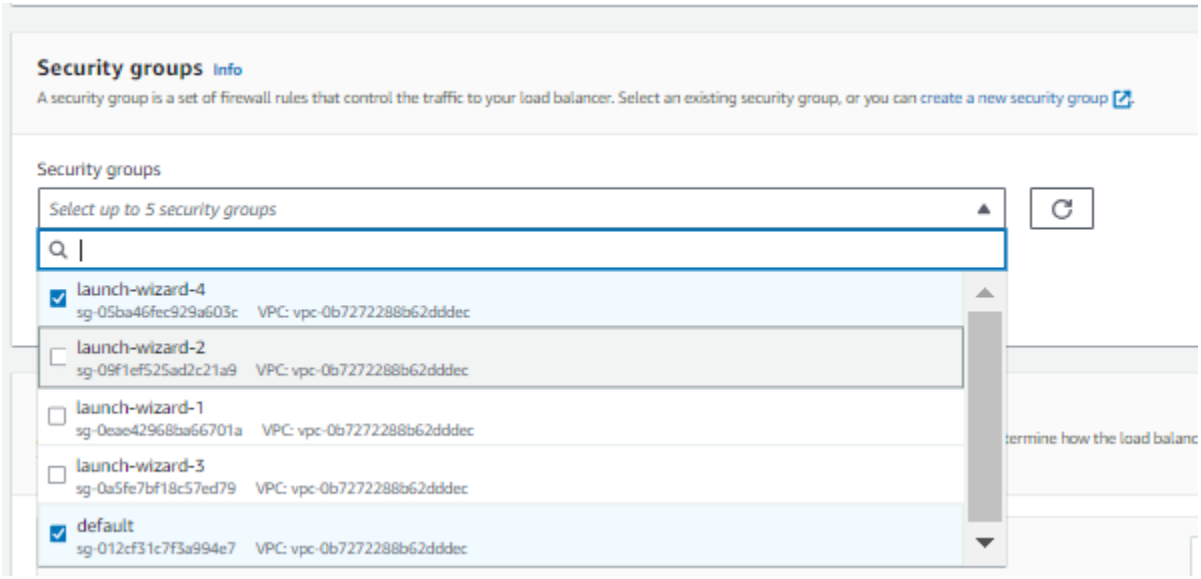
11) We select Application Load Balancer and provide the required configuration.



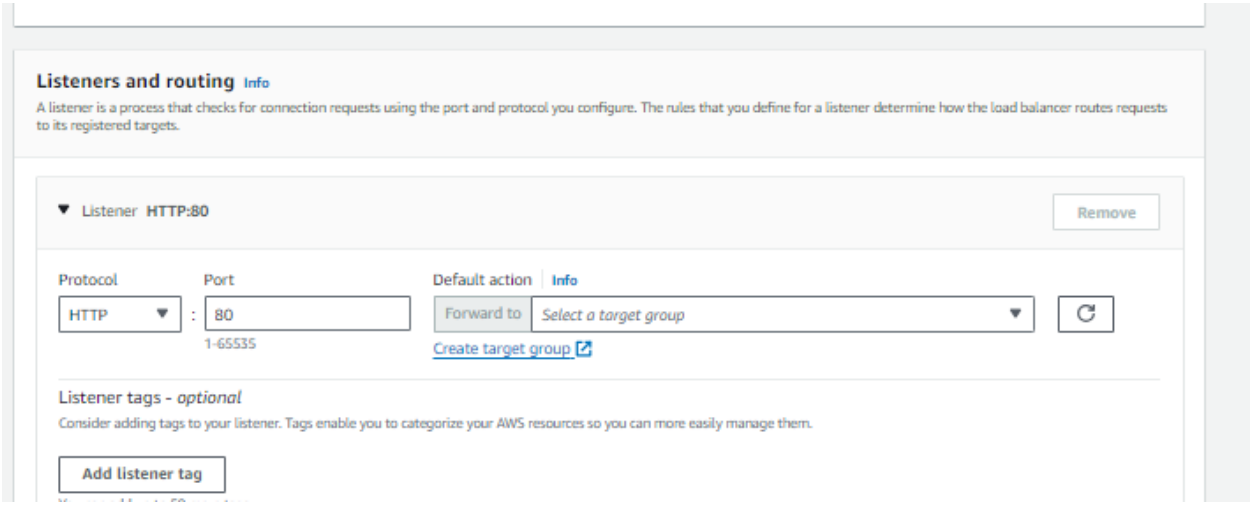
12) Select all Network Mapping options.



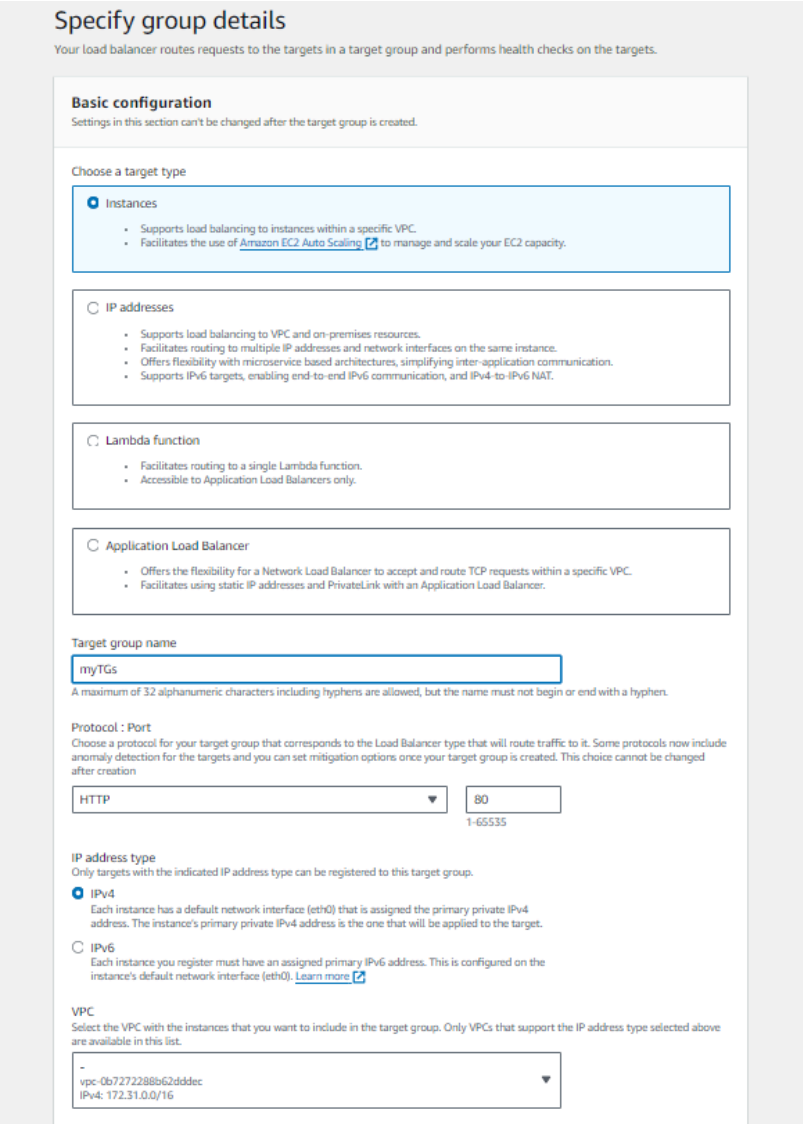
13) Application Load Balancer Security Groups



14) Creating a Target Group

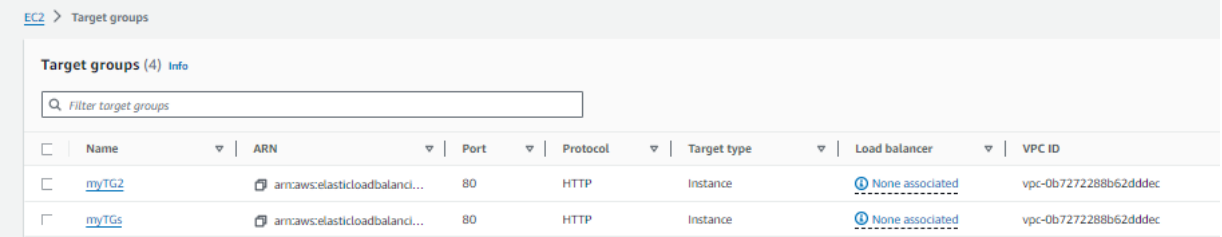


15) Provide the group details name for group -1 and for group-2.

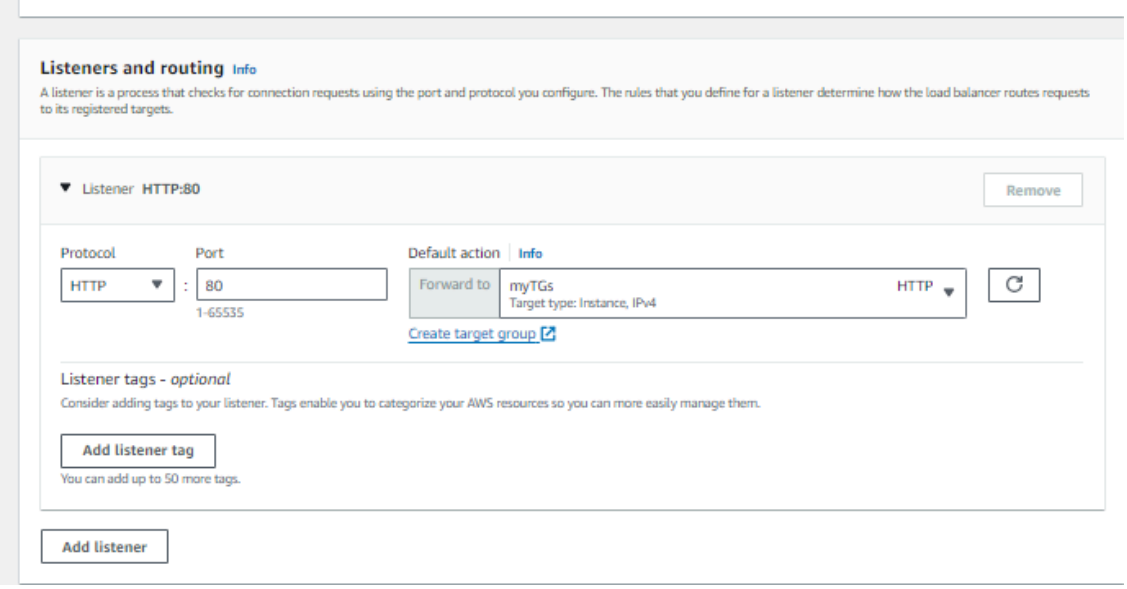


16) Configuration Review and Target Group Creation.

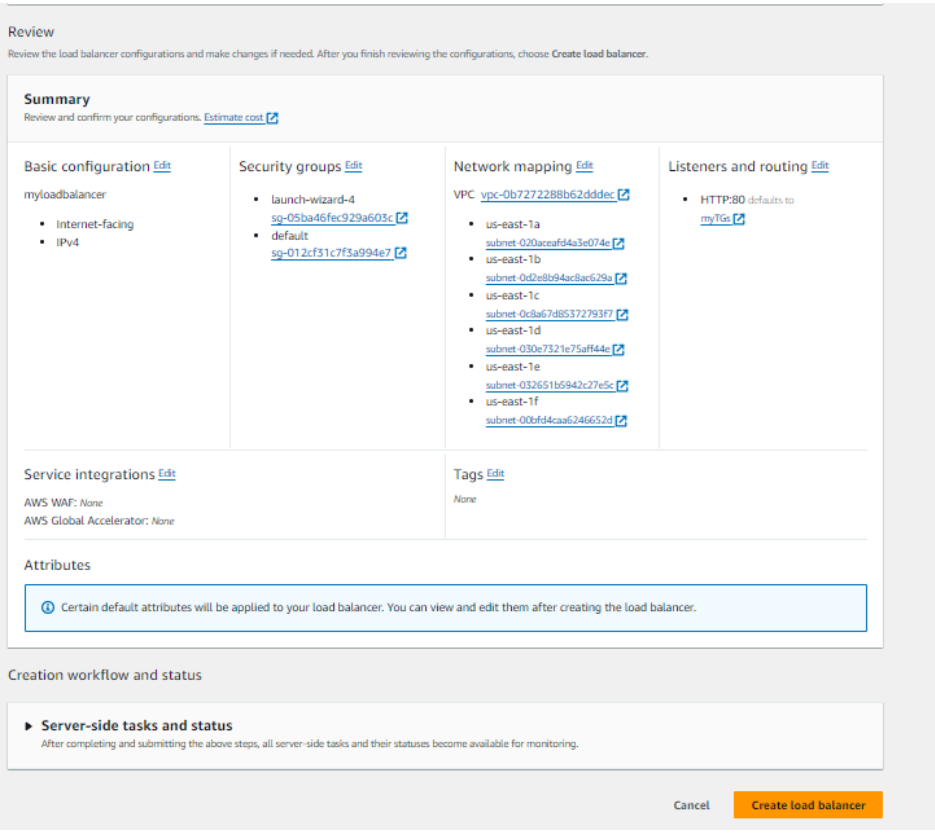
Successful Target Group Creation



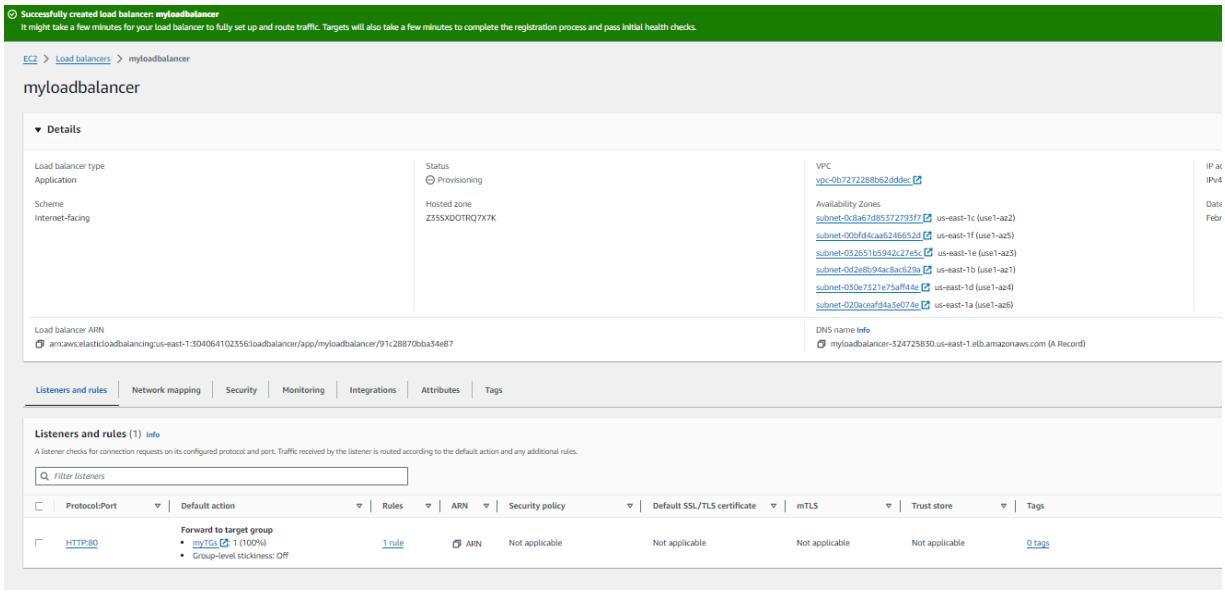
17) Now, the load balancer configuration can be continued. In Listeners and routing section,a recently created target group is selected.



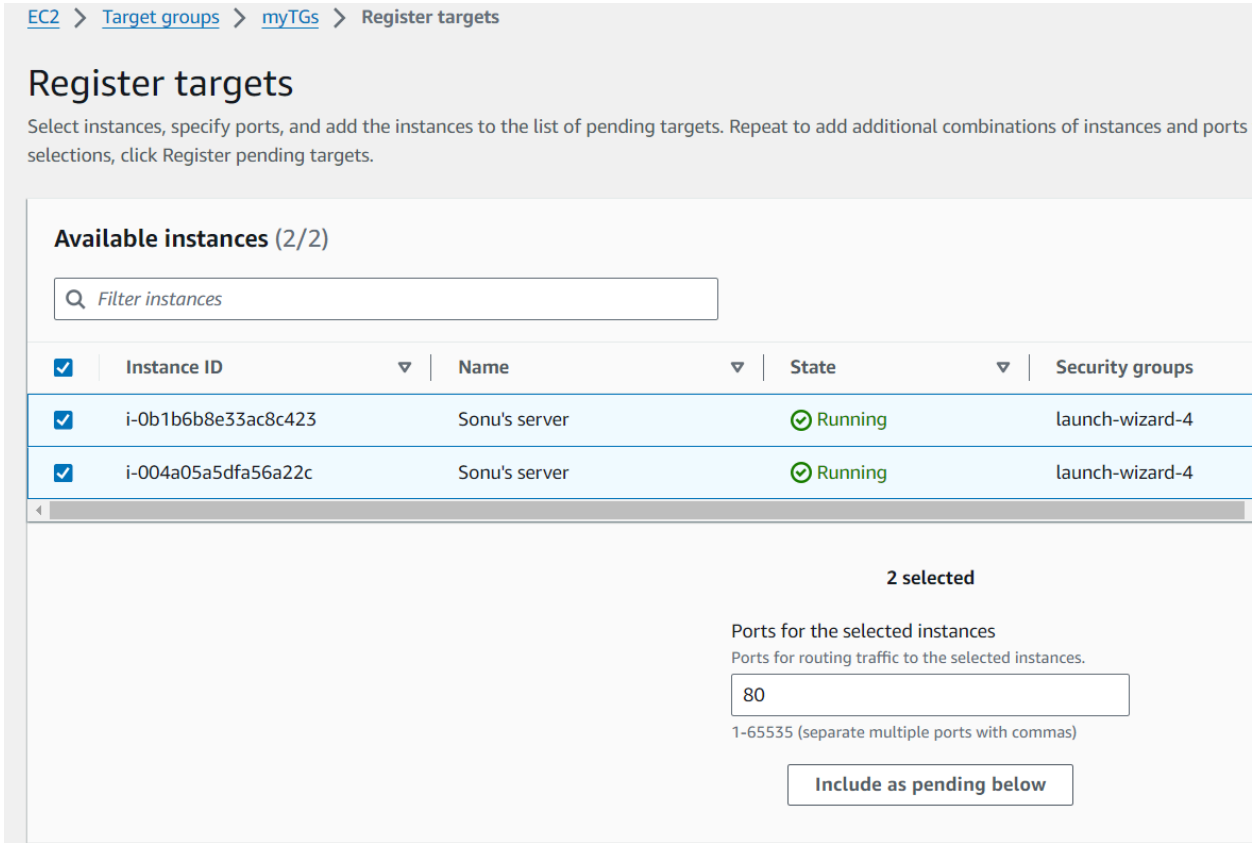
18) Summary and Load Balancer Creation.Click on Create load balancer.



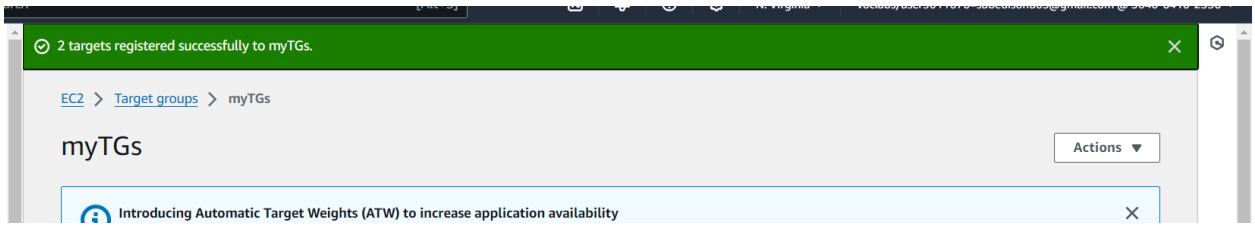
19) Successful creation of Application Load Balancer.



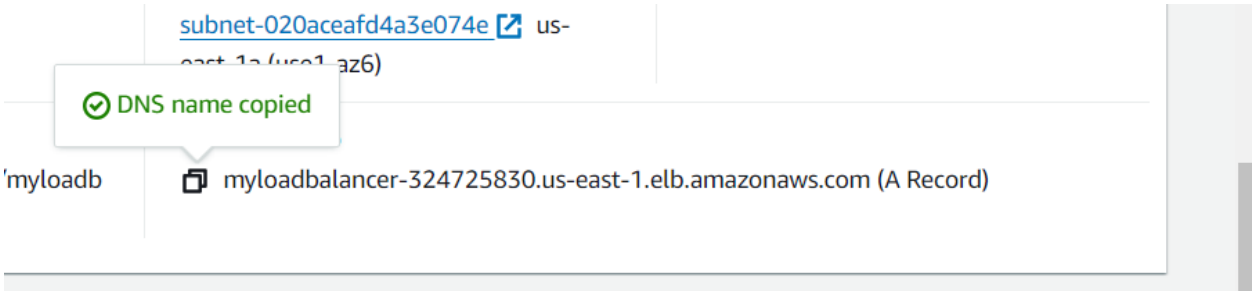
20) Registering targets



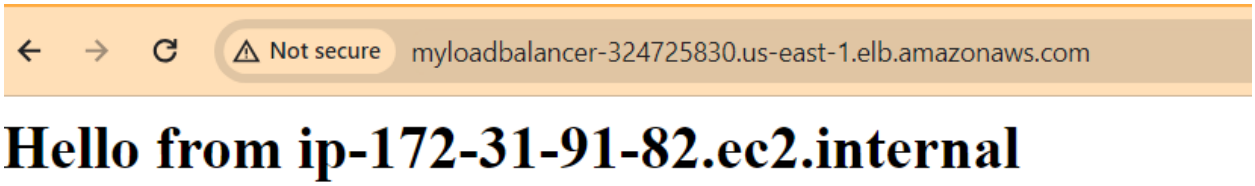
21) The targets has been successfully registered.



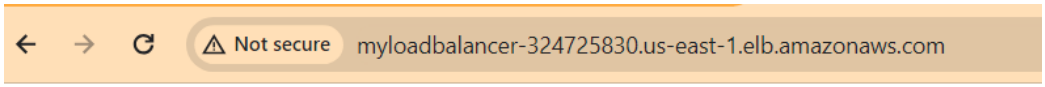
21) After target group has been registered,copy the DNS name of load balancer and access in web browser.



IP from one of the instance,

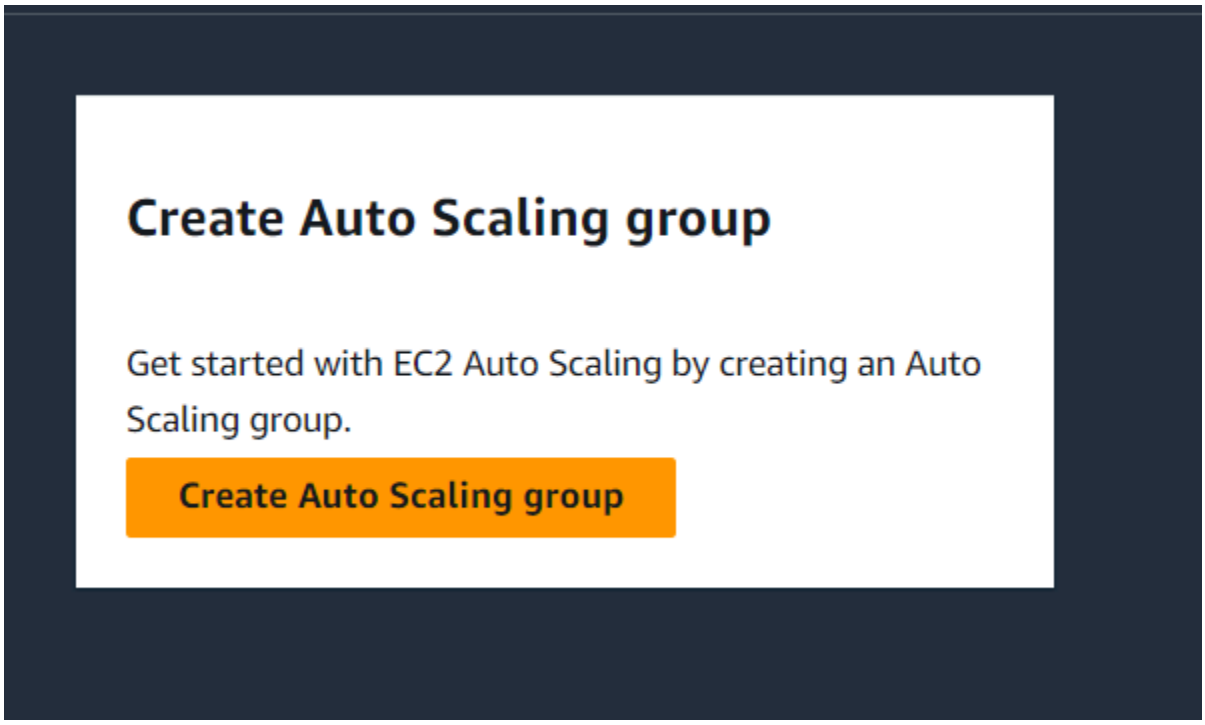


Ip from next instance

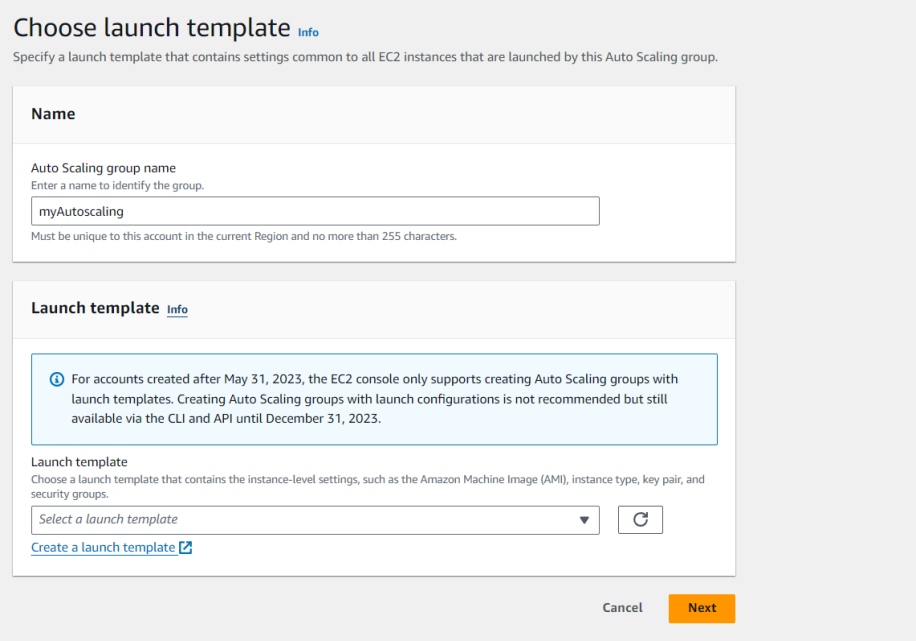


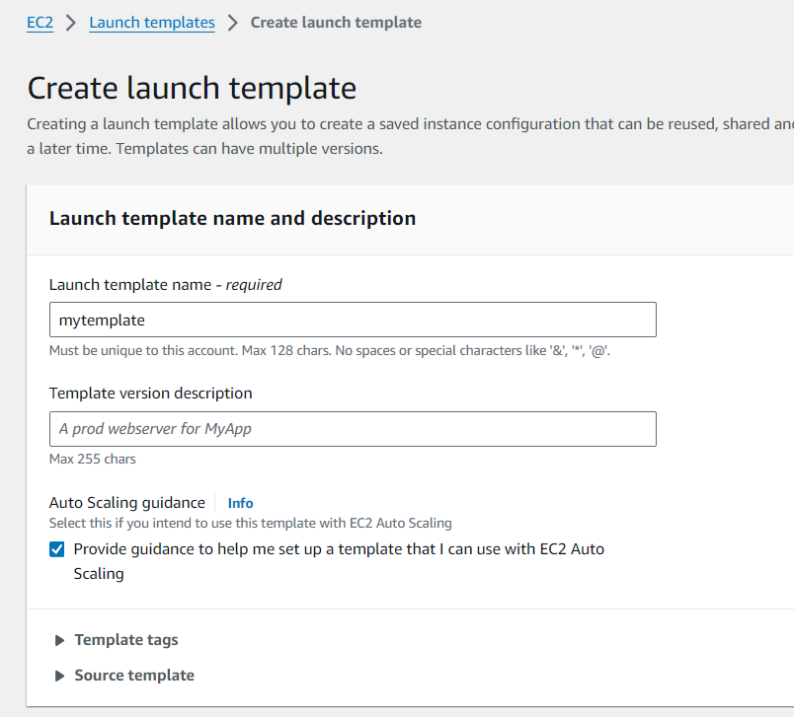
Hello from ip-172-31-89-214.ec2.internal

22) Creating an Auto Scaling Group

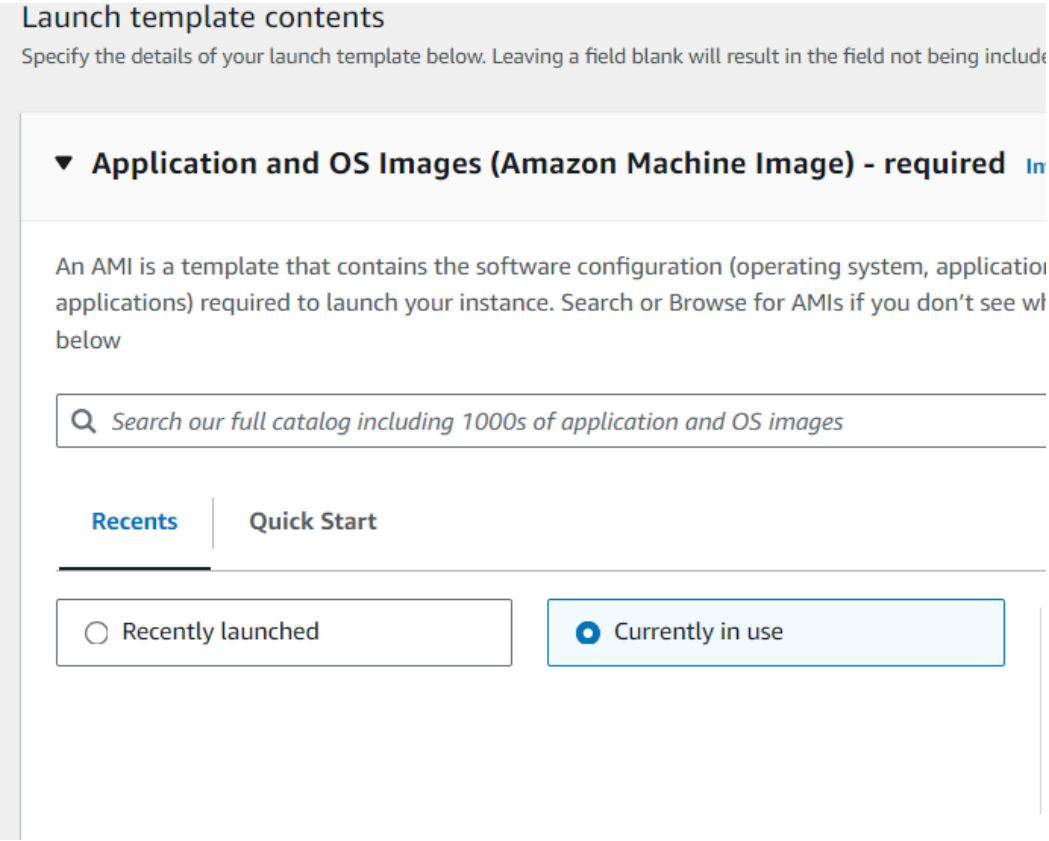


23) In the launch template section, Choose launch template, “Create a launch template” is selected and new template is created.

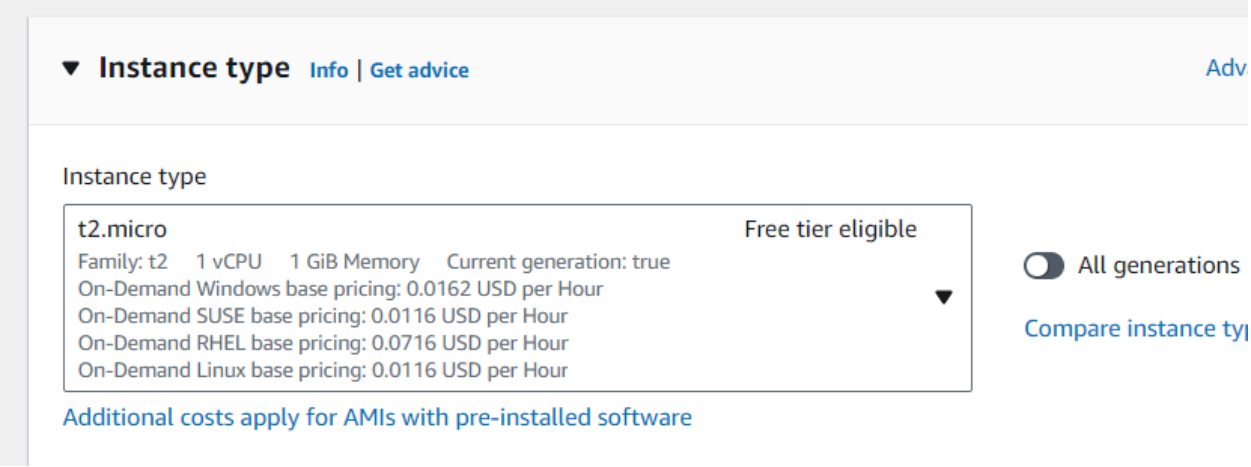


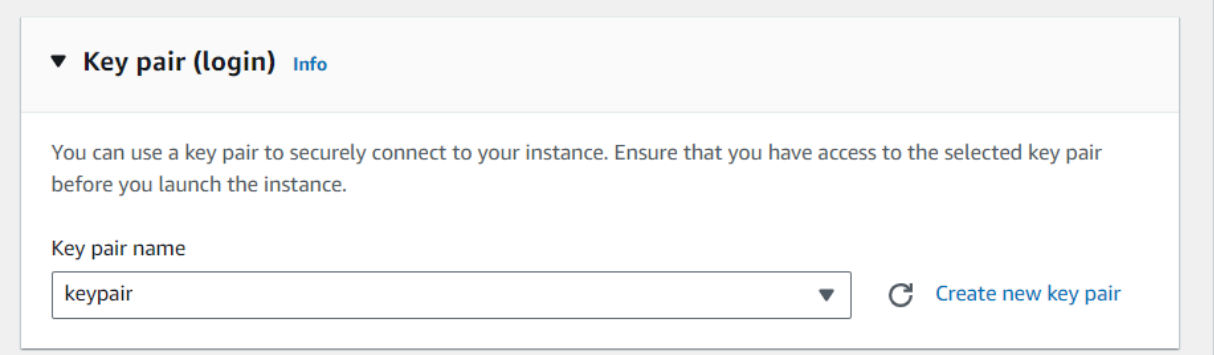


Application and OS Image
24) AMI in current use is selected

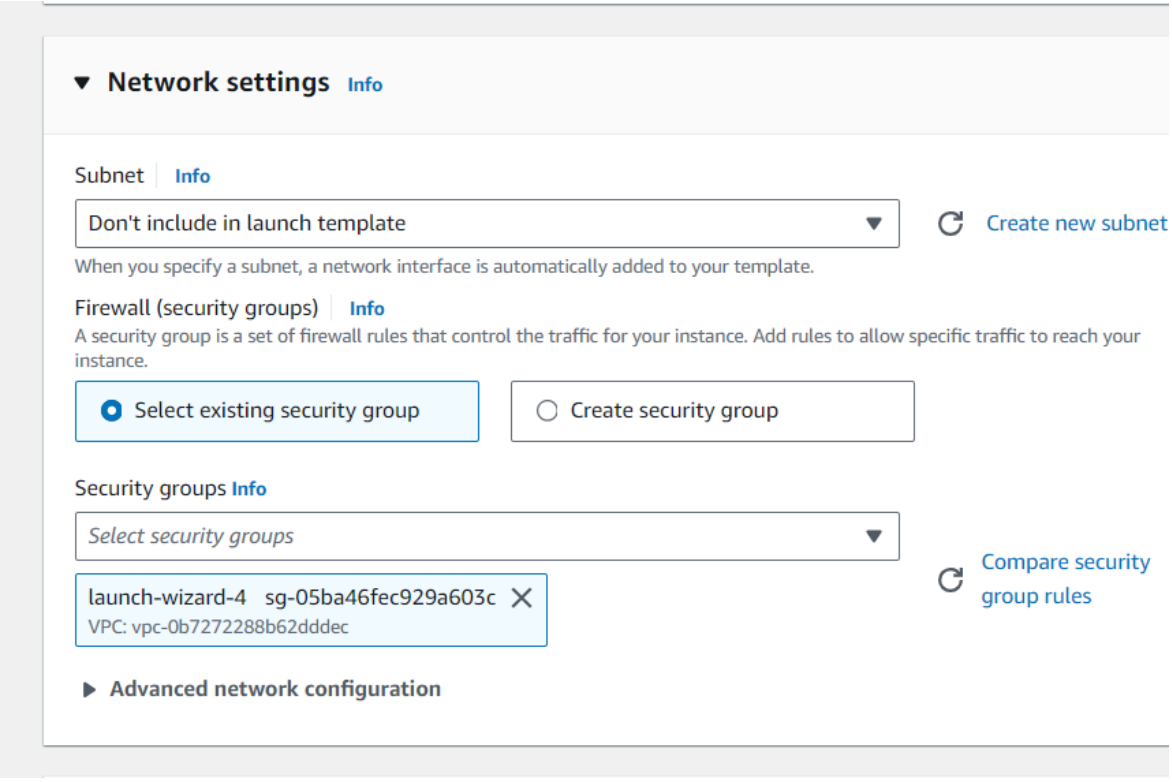


25) For instance type, t2.micro is selected and for key pair, existing key pair is selected



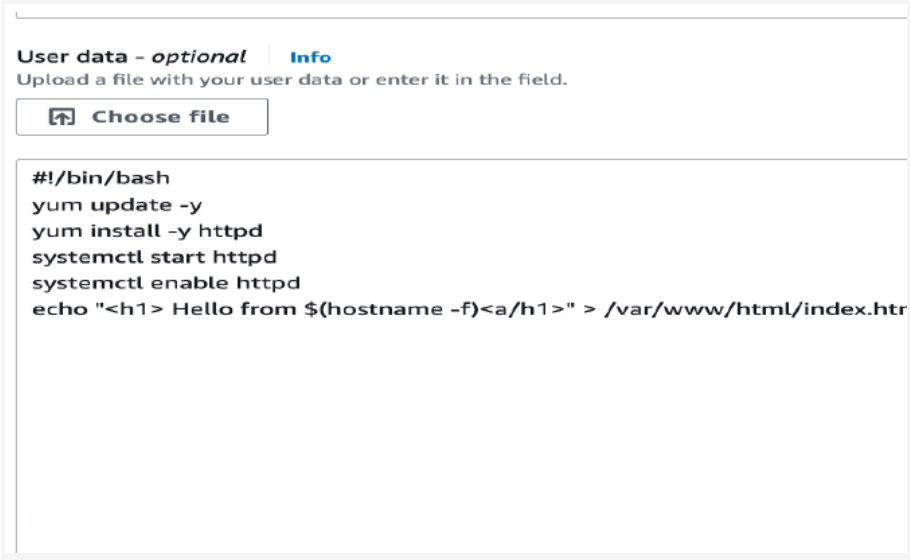


26) Network Settings
Existing Security Groups are selected and other configurations are kept the same as it.

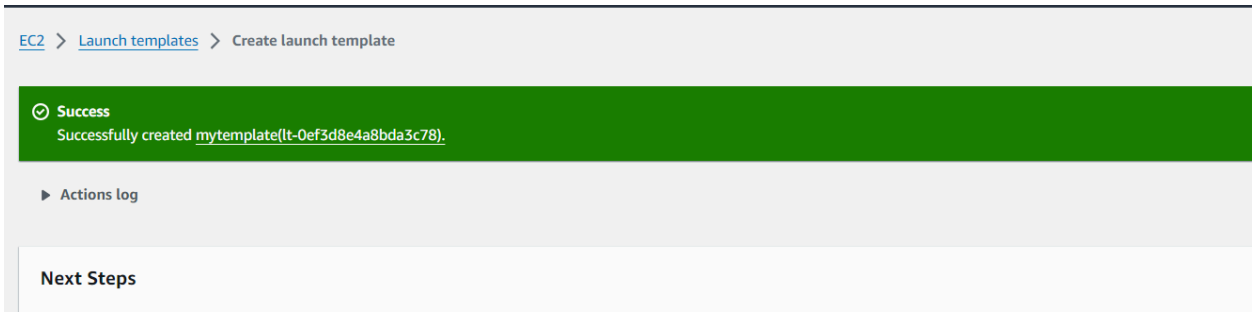


27) Bash script is added as per the previous instance configurations

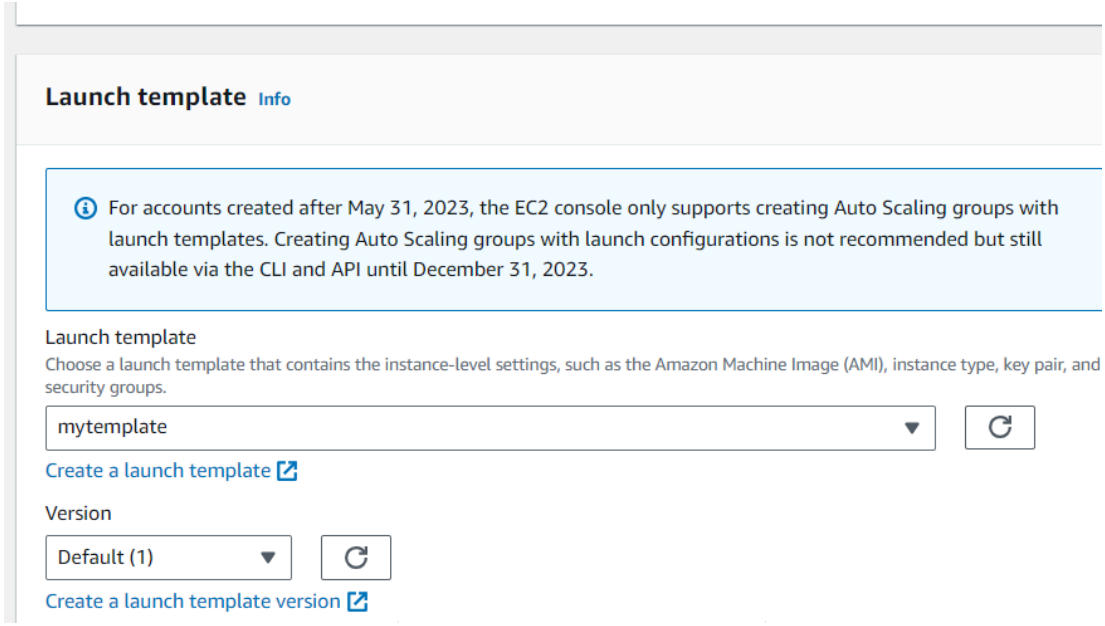
```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1> Hello from $(hostname -f)<a/h1>" >
/var/www/html/index.html
```



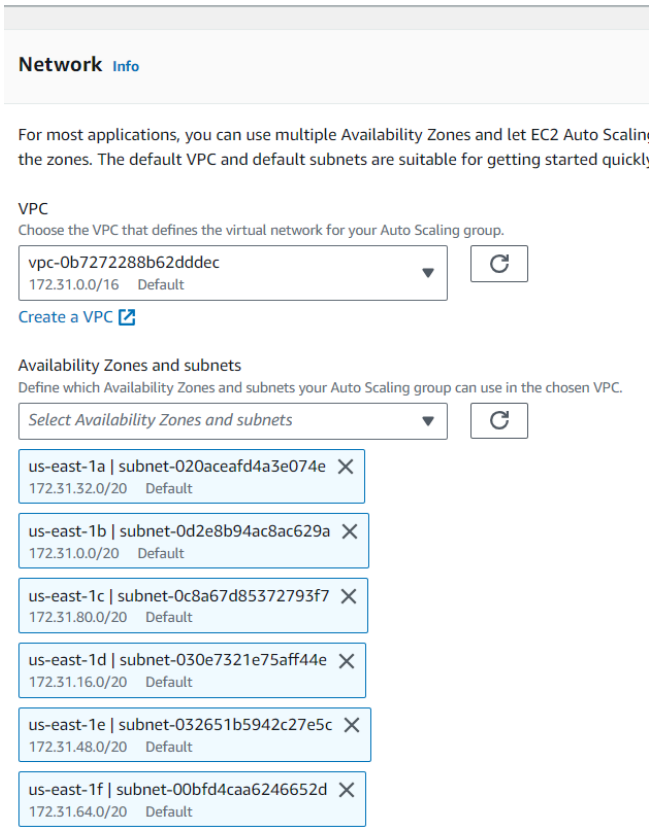
28) Launch template successfully created.



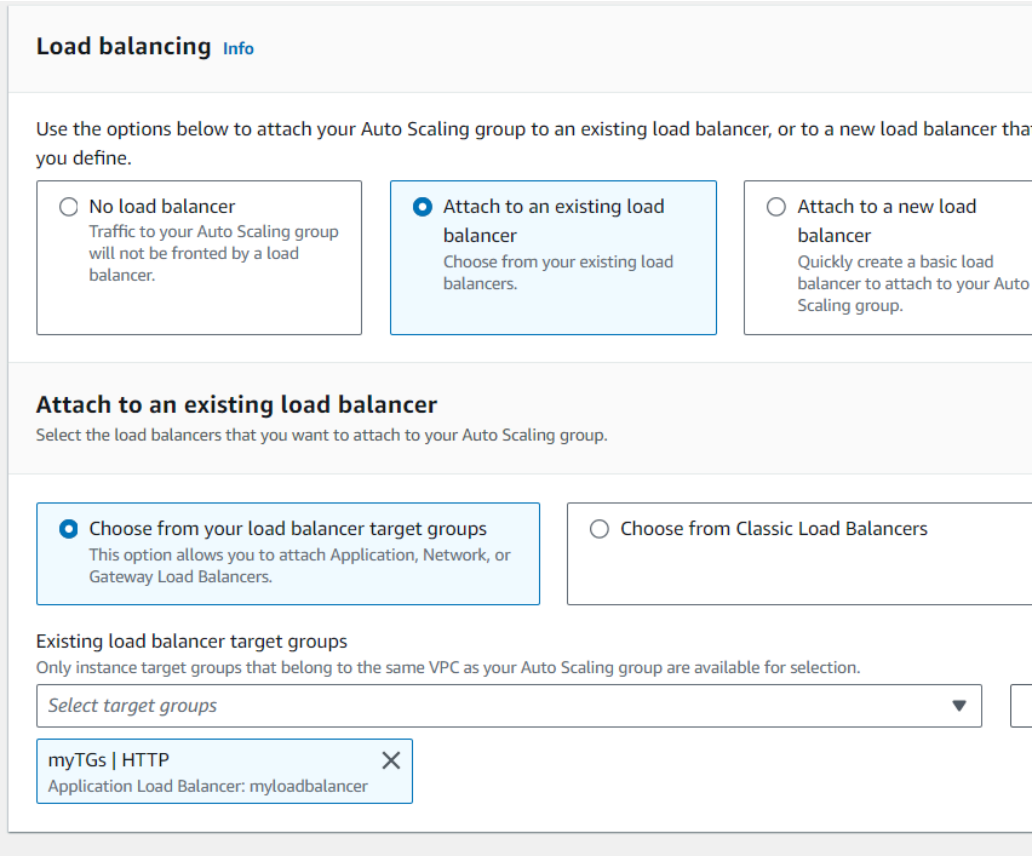
29) Now, the configuration of auto scaling is continued. Here a newly created launch template is selected.



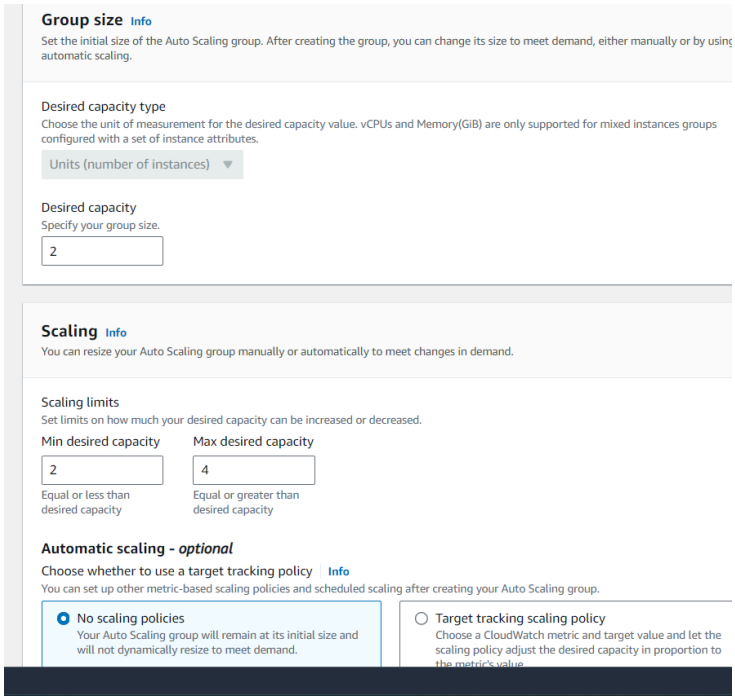
30) VPC along with all Availability Zones and subnets are selected .



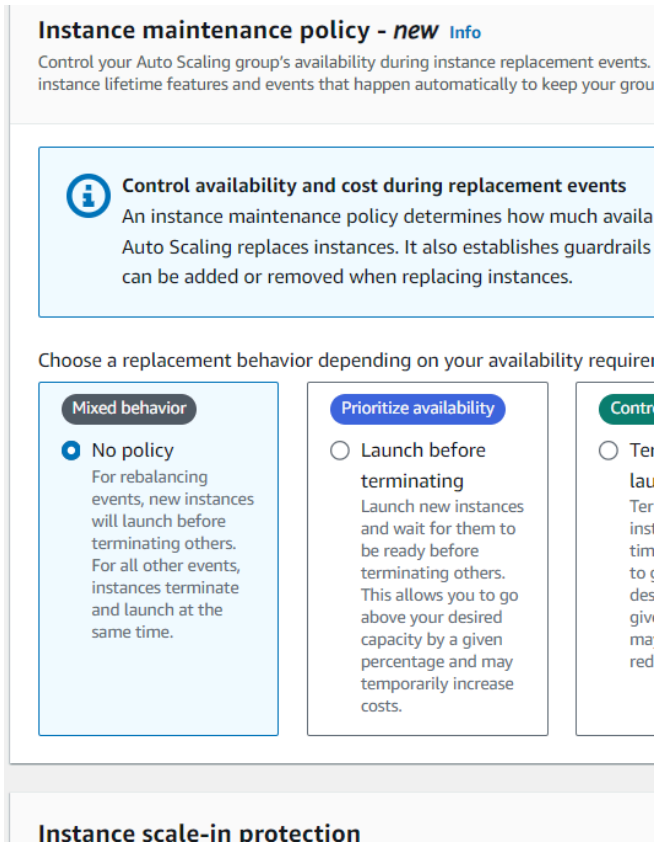
31) Configuring Advanced Options
Existing load balancer is chosen and it's target group is selected. Others advanced configuraton is kept as default.

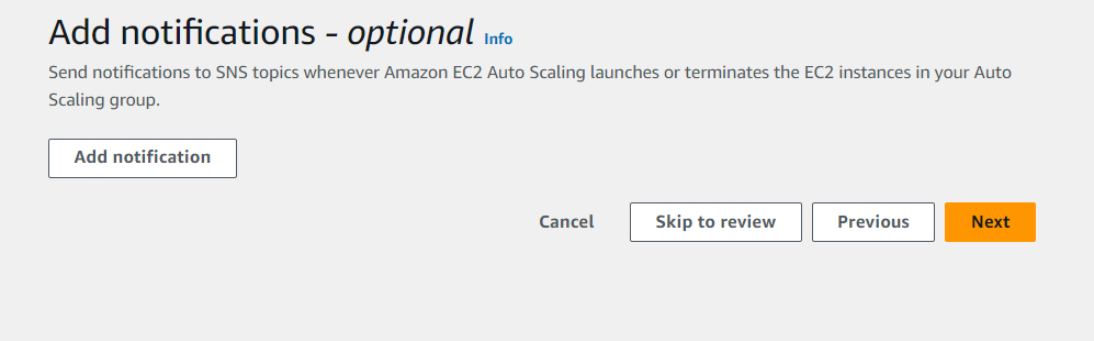


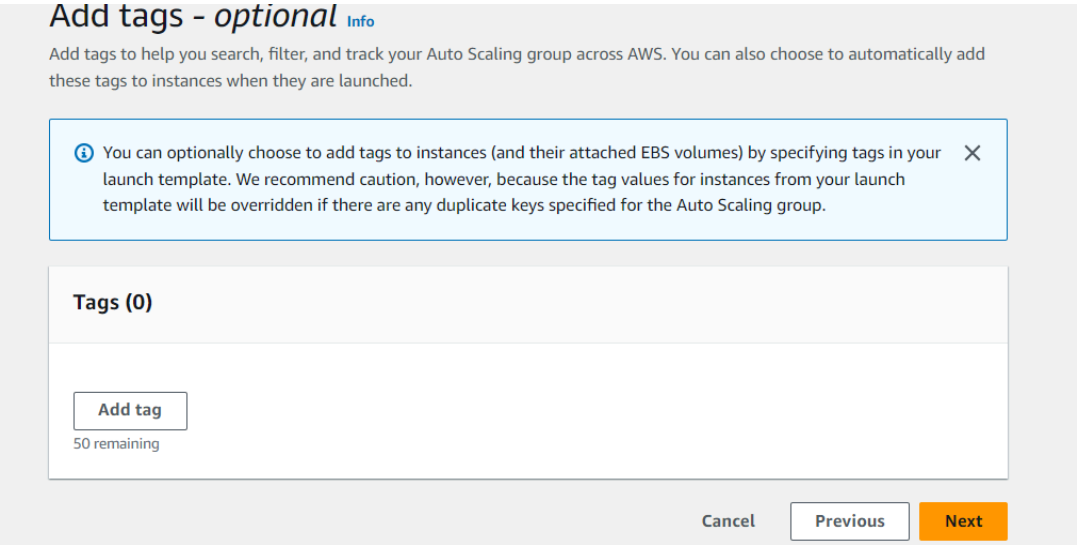
32) Group Size and Scaling Configuration is done.Here desired capacity of Group size is set as 2. In scaling, min desired capacity is set as 2 and max desired capacity as 4.



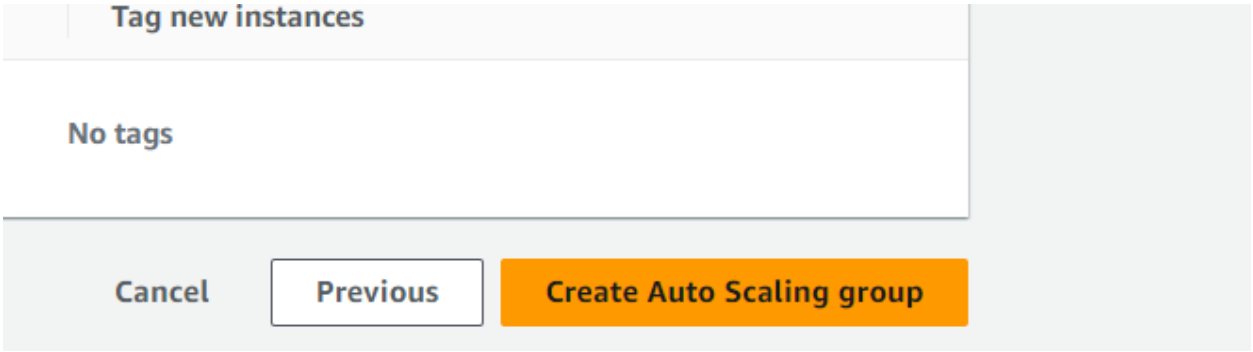
33) No Instance Maintenance Policy is selected. No notifications is added and no tags are added.



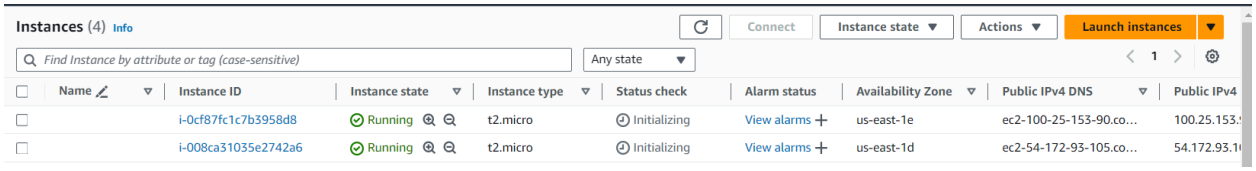


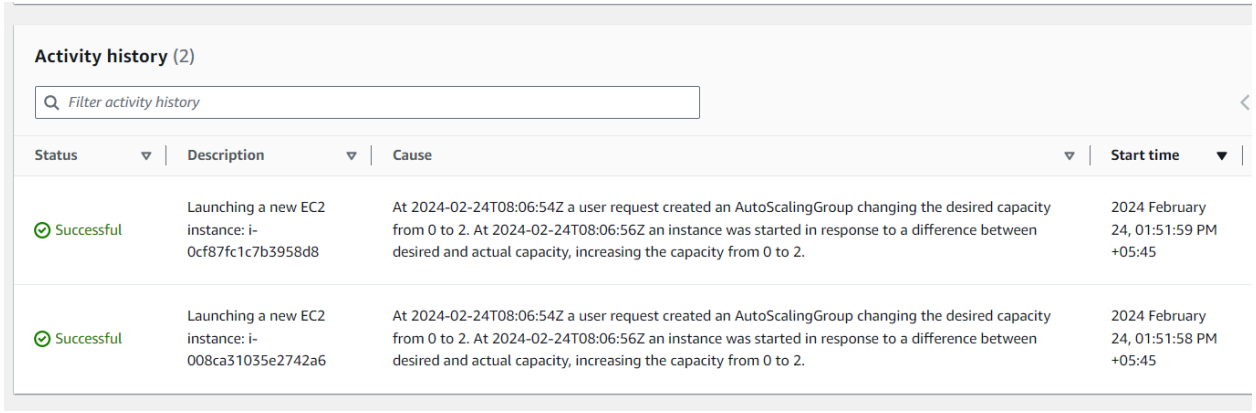


34) Now auto scaling group is configured and created.

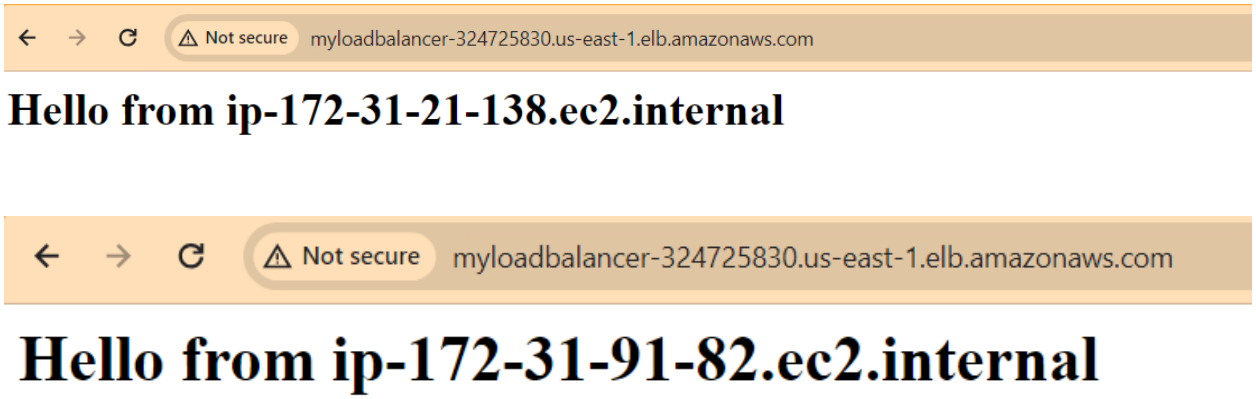


35) Now we can notice two Instances are in pending state automatically.In the activity tab of auto scaling, we can observe the activity history of the EC2 instances.



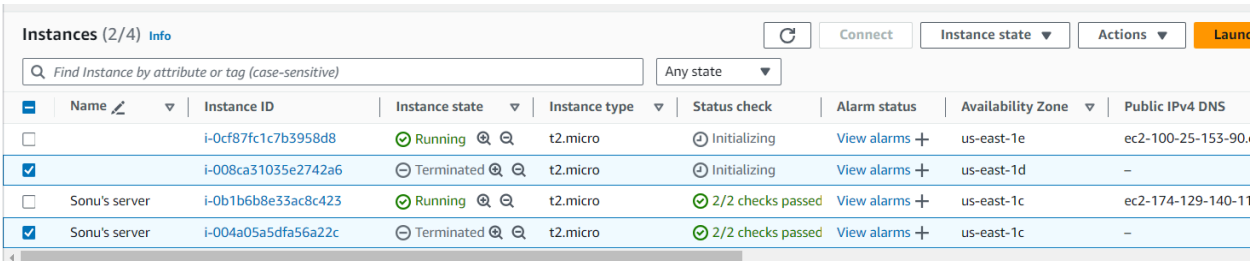


36) When the DNS of the load balancer is refreshed even multiple times, it works.



37) Testing of Auto-Scaling

For that, we terminate the instance.
We notice that the other instance is running automatically.

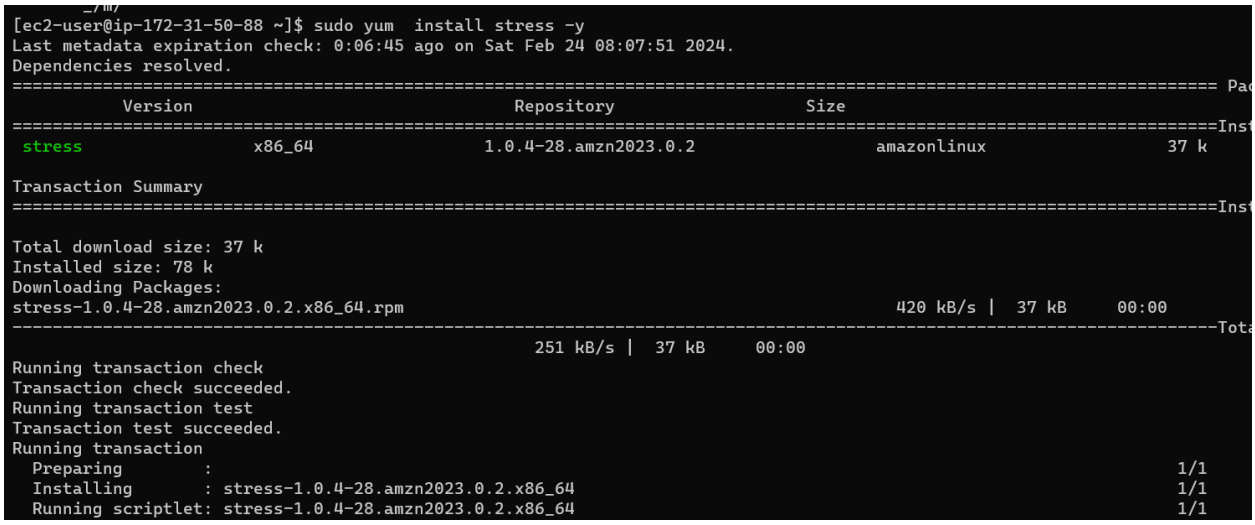


38) Now connect to the instance via SSH.

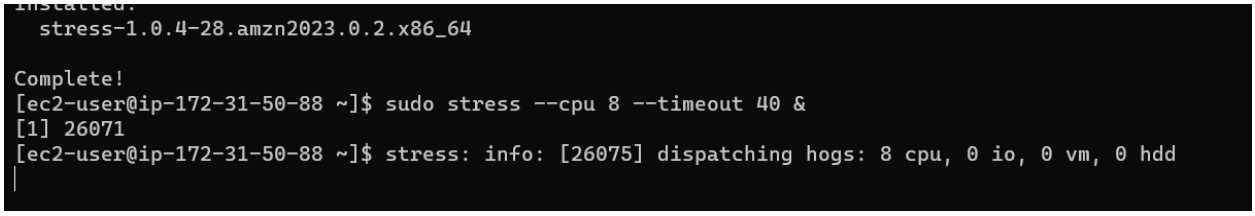
```
PS C:\Users\subed> cd Downloads
PS C:\Users\subed\Downloads> ssh -i "keypair.pem" ec2-user@100.25.153.90
The authenticity of host '100.25.153.90 (100.25.153.90)' can't be established.
ED25519 key fingerprint is SHA256:/Tw0viw+C761FAp/Ub9GkoPEl5AAhMtwgx6aeVgbHkc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '100.25.153.90' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-50-88 ~]$
```

Stress test

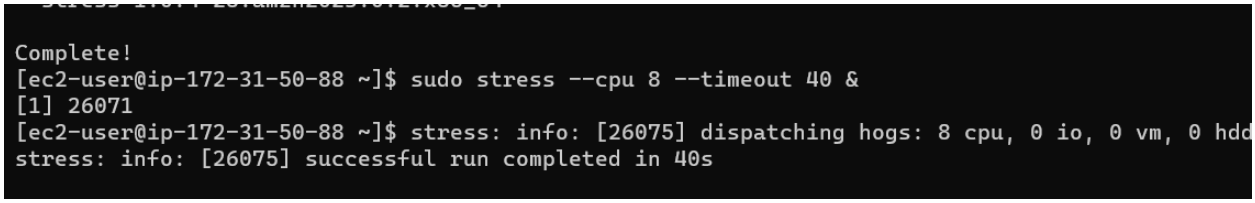
39) Stress test is installed.

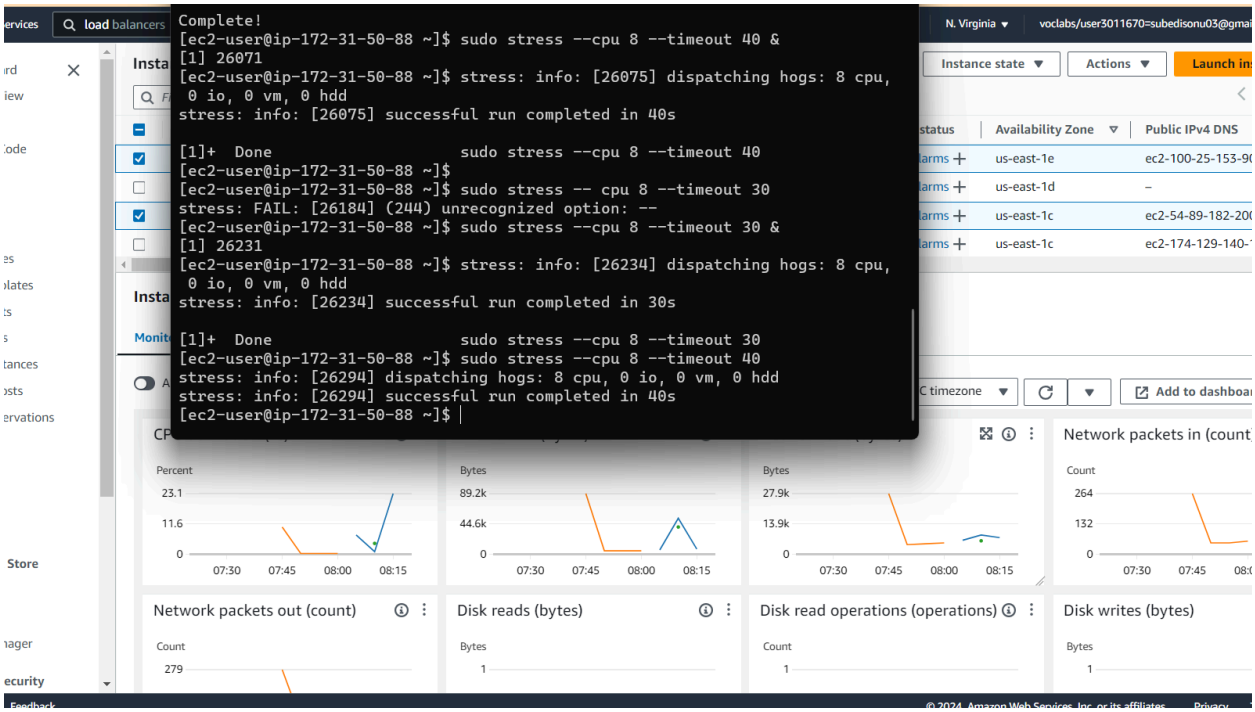


40) Now the stress test is performed.



41)Completion of stress test





42) Monitoring the instances

The CPU utilization graph of two instances is shown as. It's in increasing order.