SHRADHA SAHU

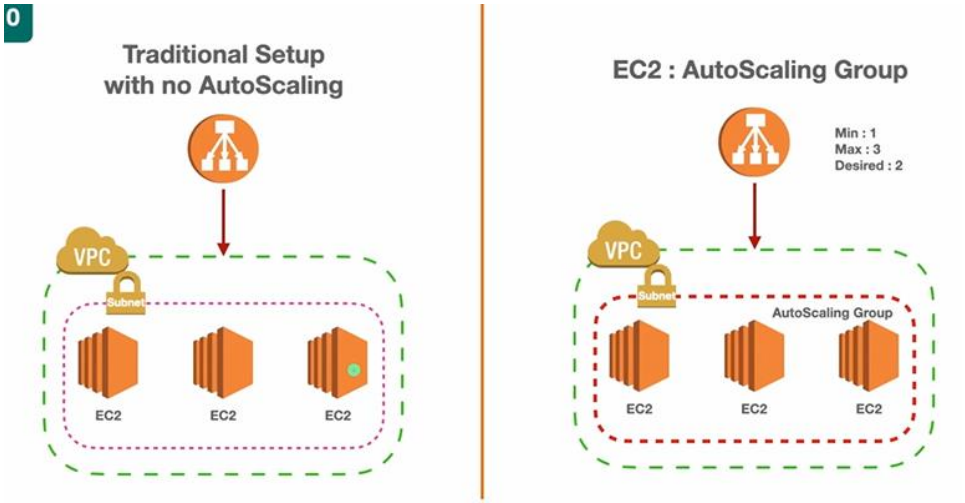
EC2 AUTO SCALING

INTRODUCTION:

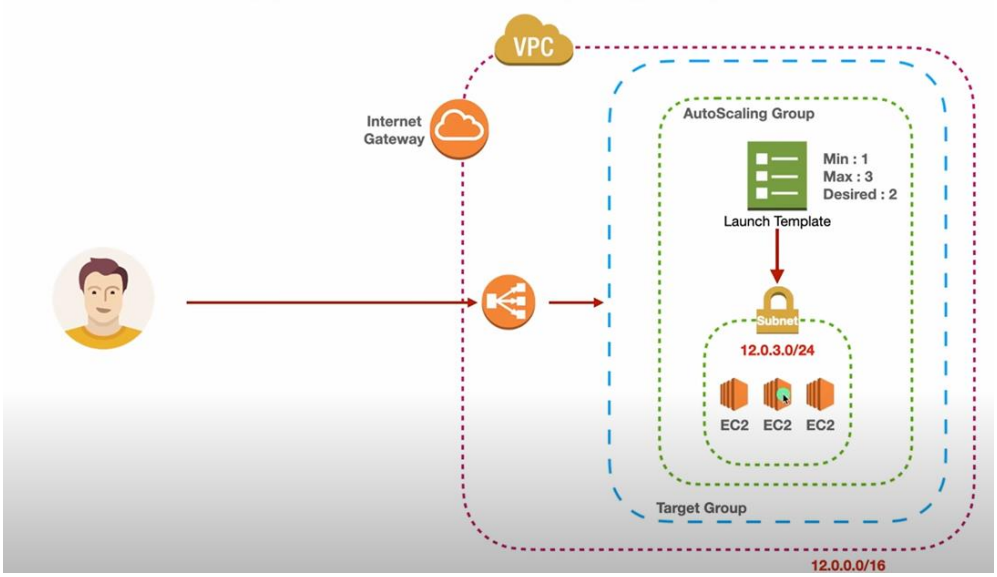
Implemented EC2 Auto Scaling to dynamically scale instances as per application needs.

What is Amazon EC2 Auto Scaling?

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called *Auto Scaling groups*. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instance. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.



ARCHITECTURE:



SERVICES USED:

- **Application Load Balancer** - An Application Load Balancer functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action. You can configure listener rules to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups.
- **VPC (Virtual Private Cloud) -** which is a virtual network infrastructure provided by AWS. It enables you to create a logically isolated section of the AWS cloud where you can launch resources such as EC2 instances, RDS databases, and more.
- **Internet Gateway (IGW)** - is a horizontally scaled, redundant, and highly available component in AWS that allows communication between resources in a VPC and the internet. An Internet Gateway is a virtual router that connects a VPC to the internet. It provides a target for traffic destined for the public internet from instances in the VPC and a source for traffic originating from the internet and intended for instances in the VPC.

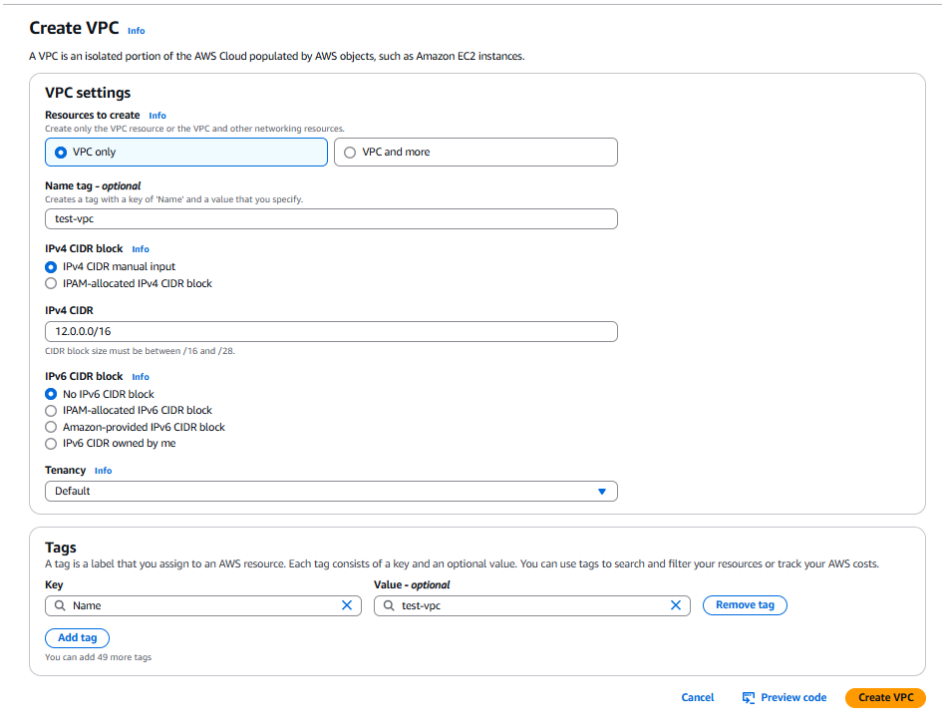
IMPLEMENTATION:

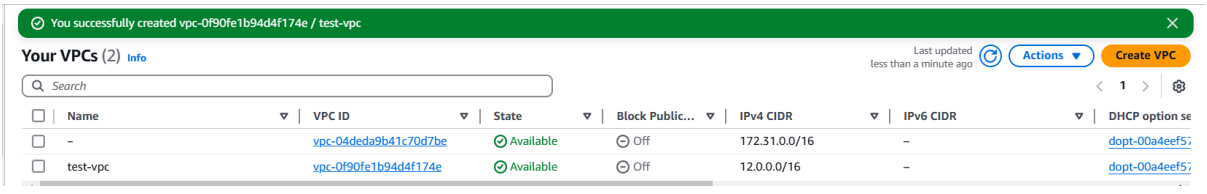
STEP1) Create VPC

By default, when you create a new AWS account, a default VPC (Virtual Private Cloud) is created for you in each AWS region. The default VPC comes preconfigured with several default settings, including an Internet Gateway and a default subnet in each Availability Zone within the region. This means that you can launch your resources in the default VPC without having to worry about configuring networking settings.

-Here, we will create custom VPC

- Click on “Create VPC” ->select VPC only -> IPV4 CIDR block (12.0.0.0/16)
- name- “**test-vpc”**

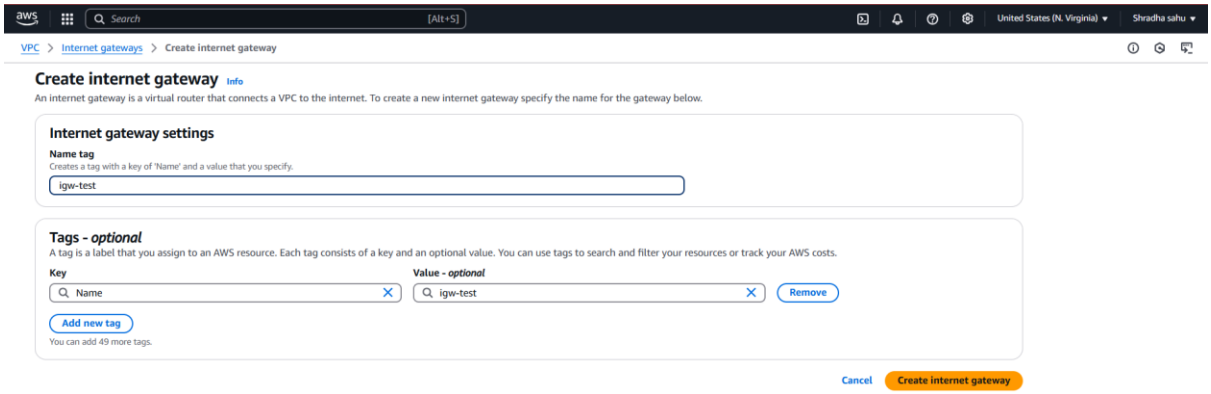


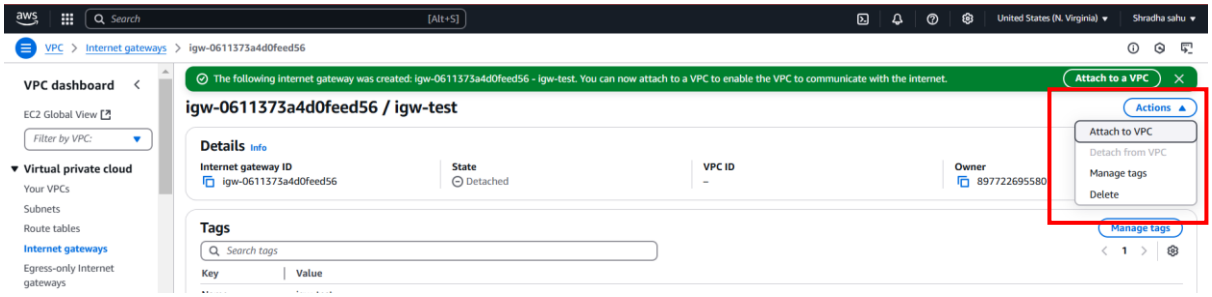


- test-vpc created

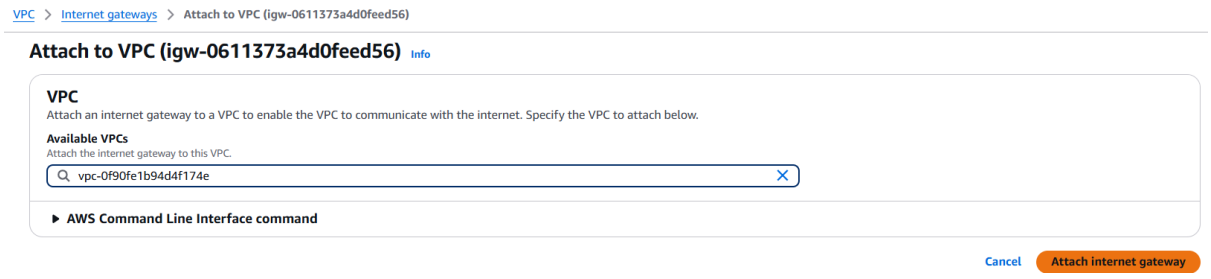
STEP2) Create an Internet gateway to your Custom VPC

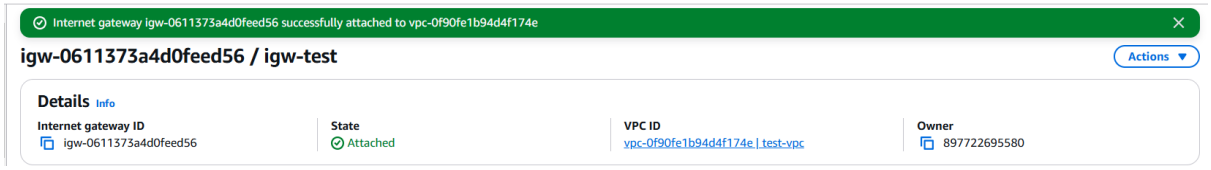
- Name- "**igw-test**" ->click on "Create Internet gateway"





- igw-test created but it is not attached to custom VPC
- Attach IGW to VPC by clicking on "Attach to VPC" and select the VPC to attach.

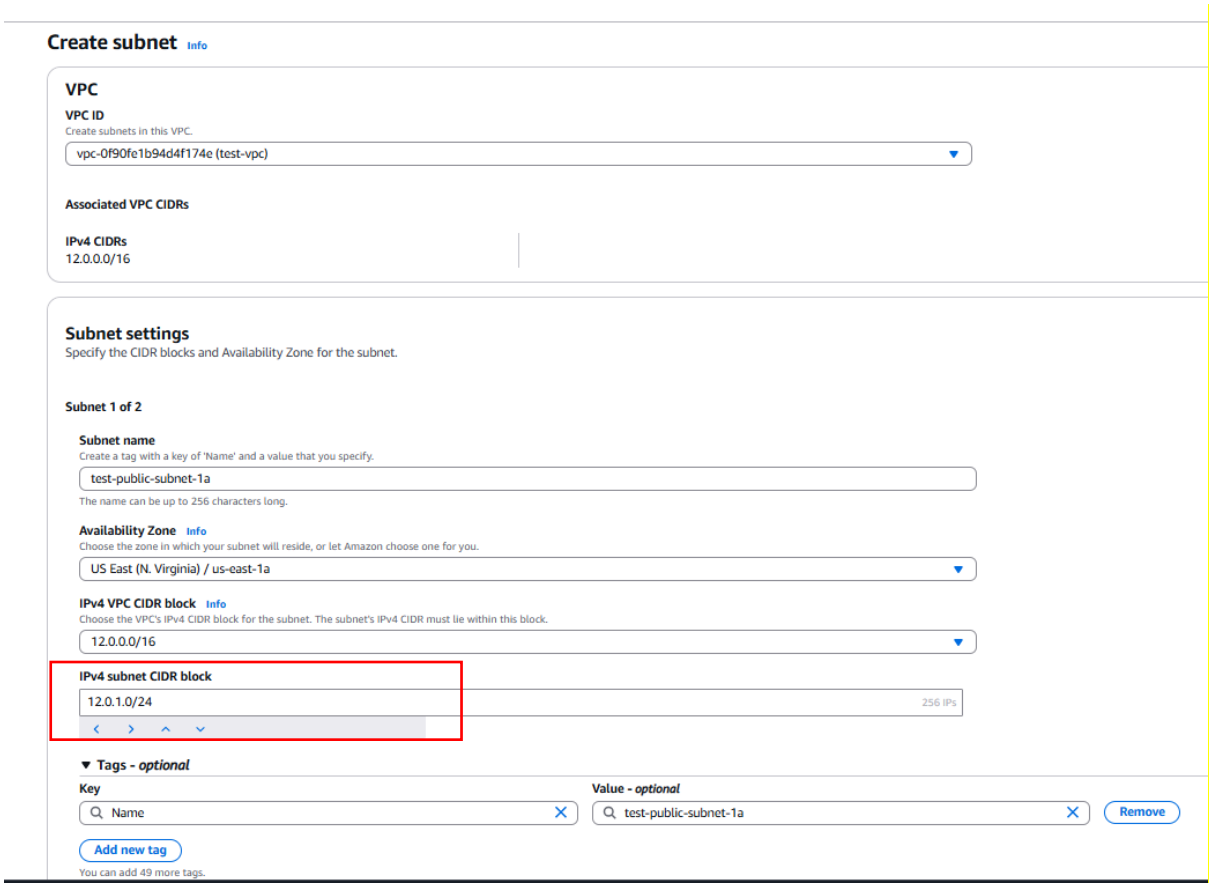


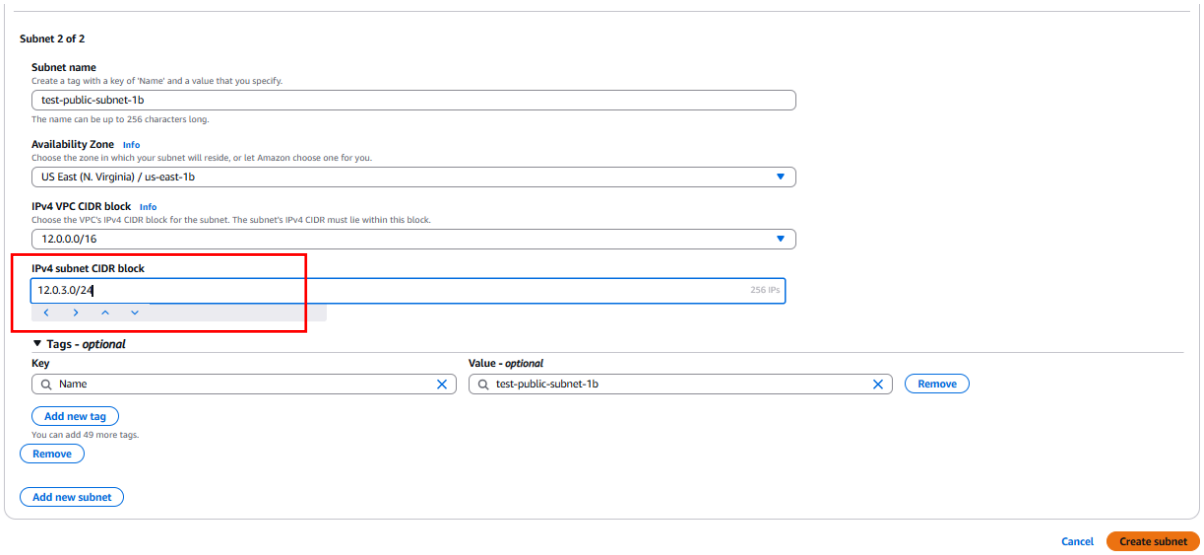


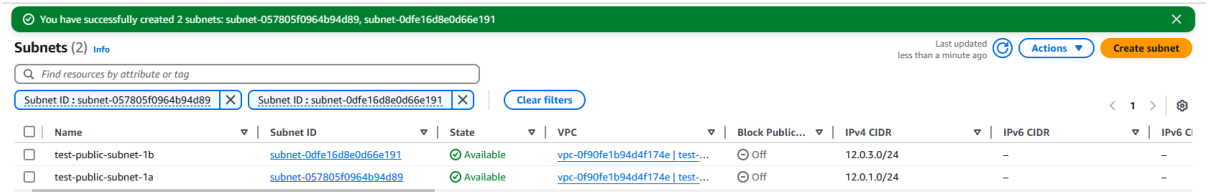
- "igw-test" is now attached to "**test-vpc".**

STEP3) Create public subnets

- Click on Create subnet, select your custom VPC, Give name to subnet, select Availability zone according your convince and assign IPV4 CDIR block to this subnet.
- Create 2 public subnets (“***test-public-subnet-1a***” and “***test-public-subnet-1b”***) in different availability zones (“***us-east-1a***” and “***us-east-1b***”) and IPV4 CDIR blocks (**12.0.1.0/24** and **12.0.3.0/24**)

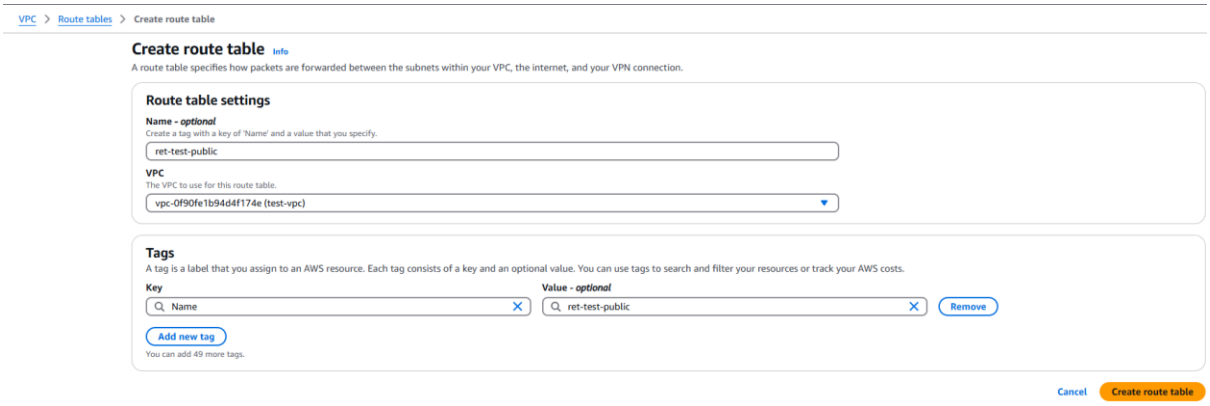




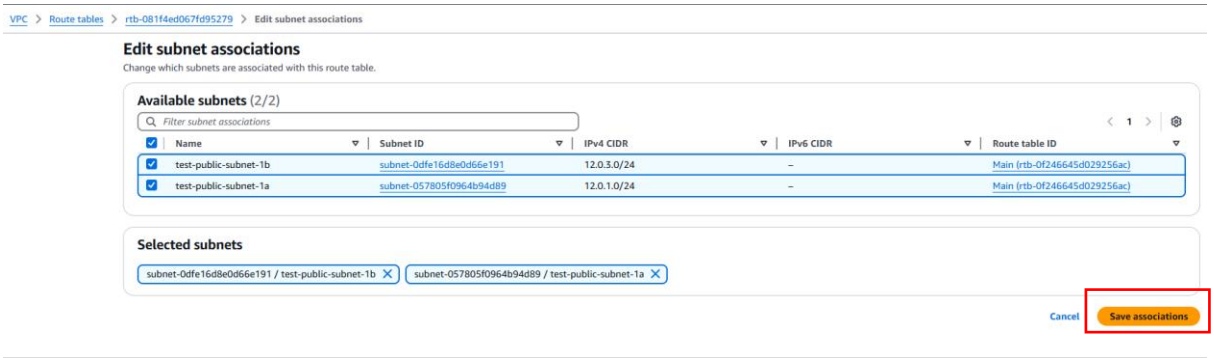


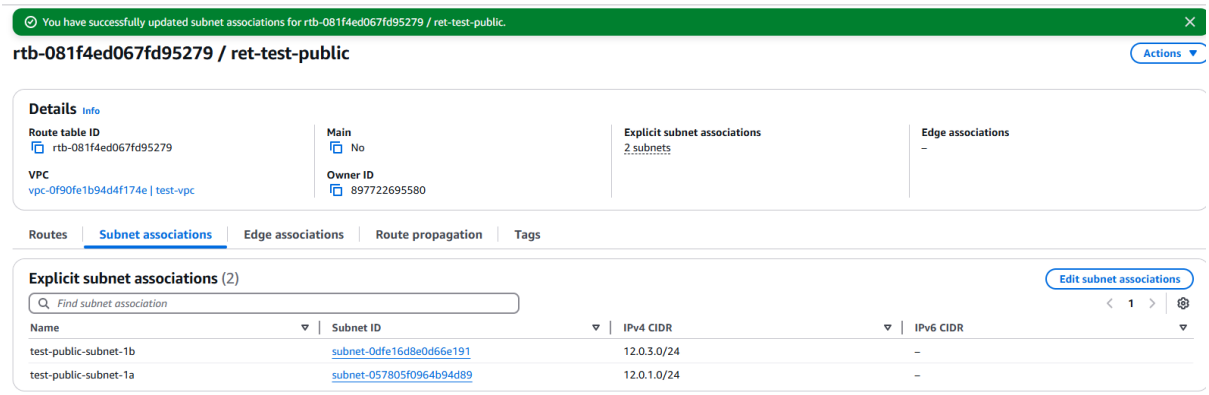
- subnets created for custom vpc

STEP4) Create Route table (“**ret-test-public**”) for public subnet.



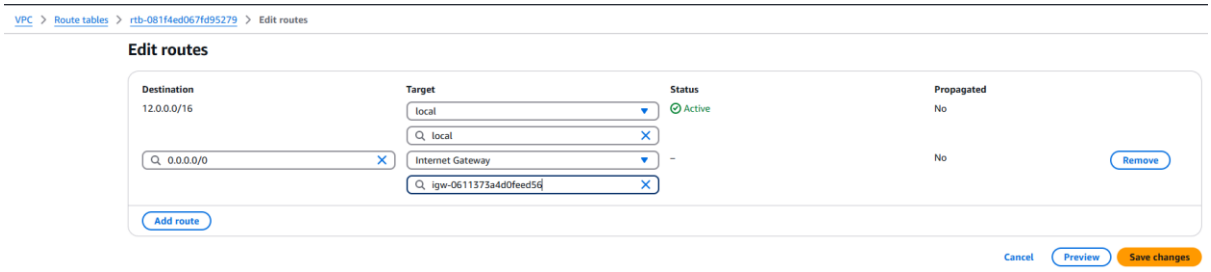
- Associate “**ret-test-public**” with public subnet-> Click on “subnet association” ->“Edit subnet association”-> click on public subnet.





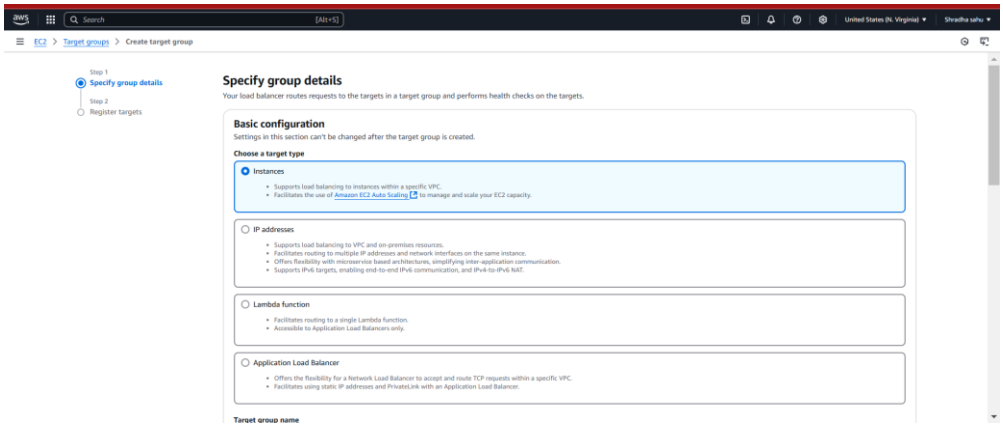
- For directing traffic within the VPC public route is associated with public subnet and public subnet needs internet access.

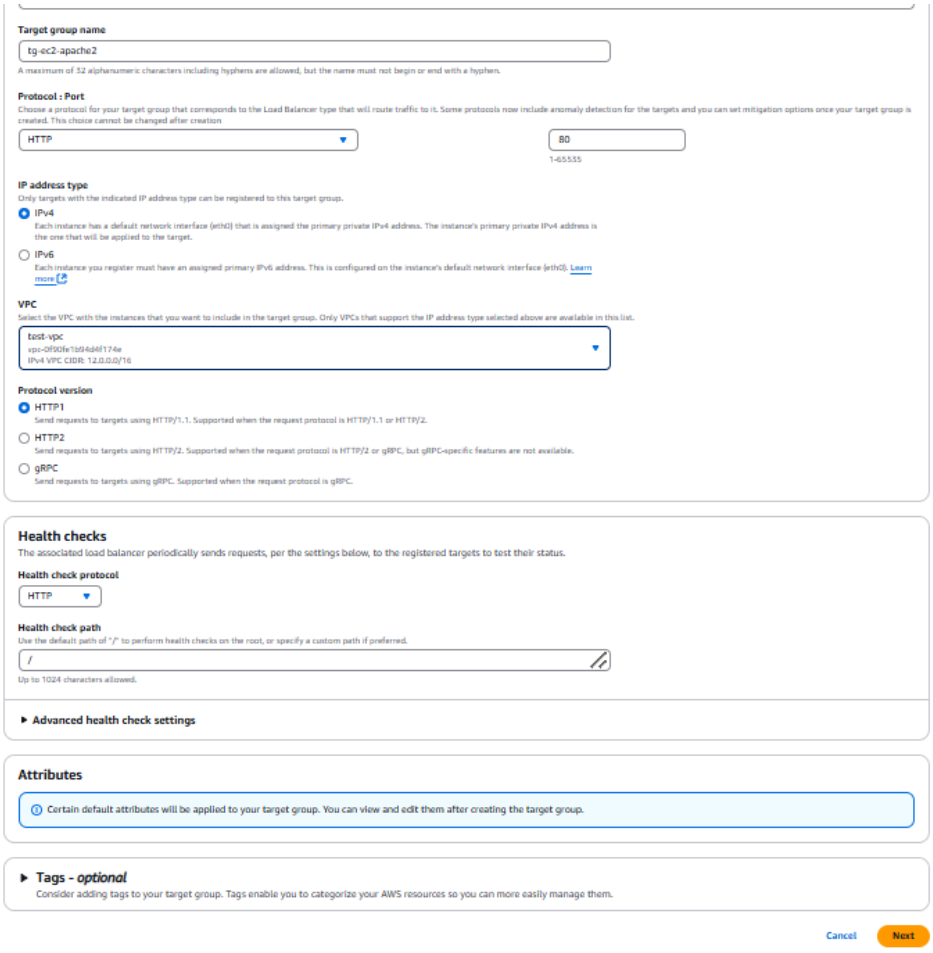
- Edit route by associating it with internal gateway.



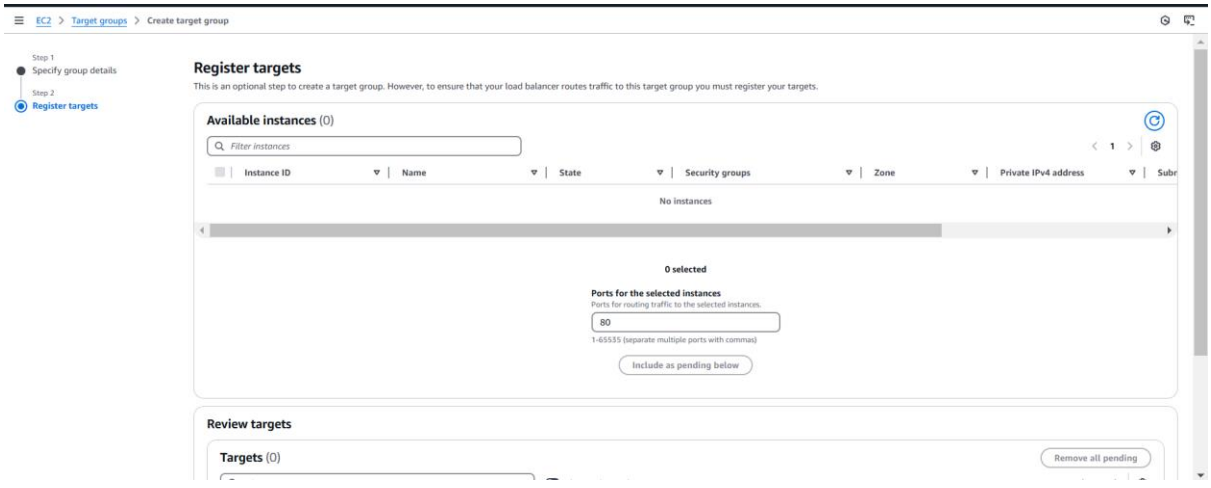
STEP 5) Create Target group

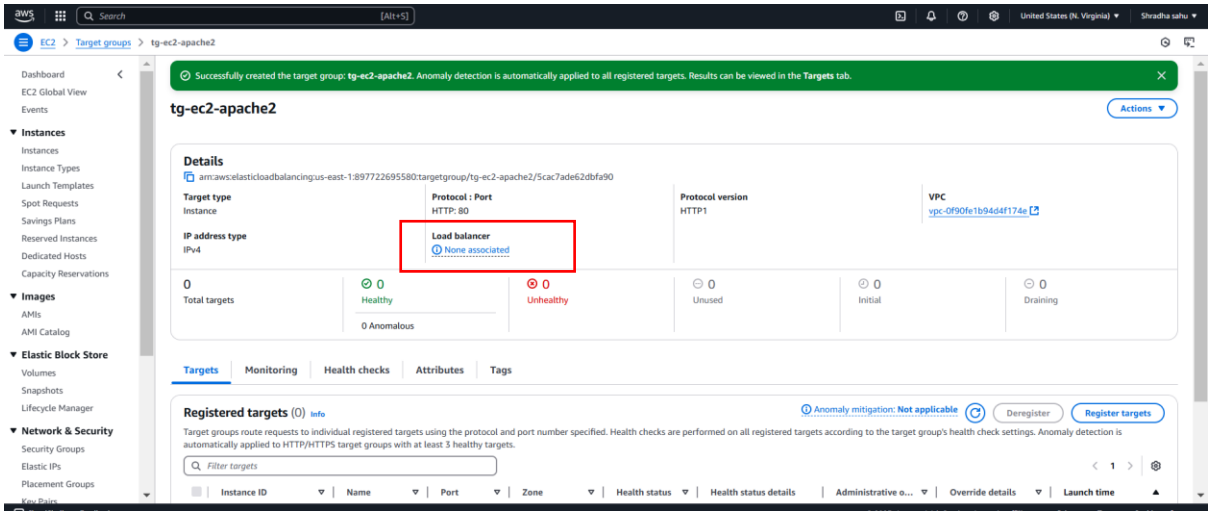
- Target group is responsible for pointing the EC2 instances.
- Target group name ("**tg-ec2-apache2**")->Protocol (**HTTP**)





- Blank target group is created as instances will be created via Autoscale policy.

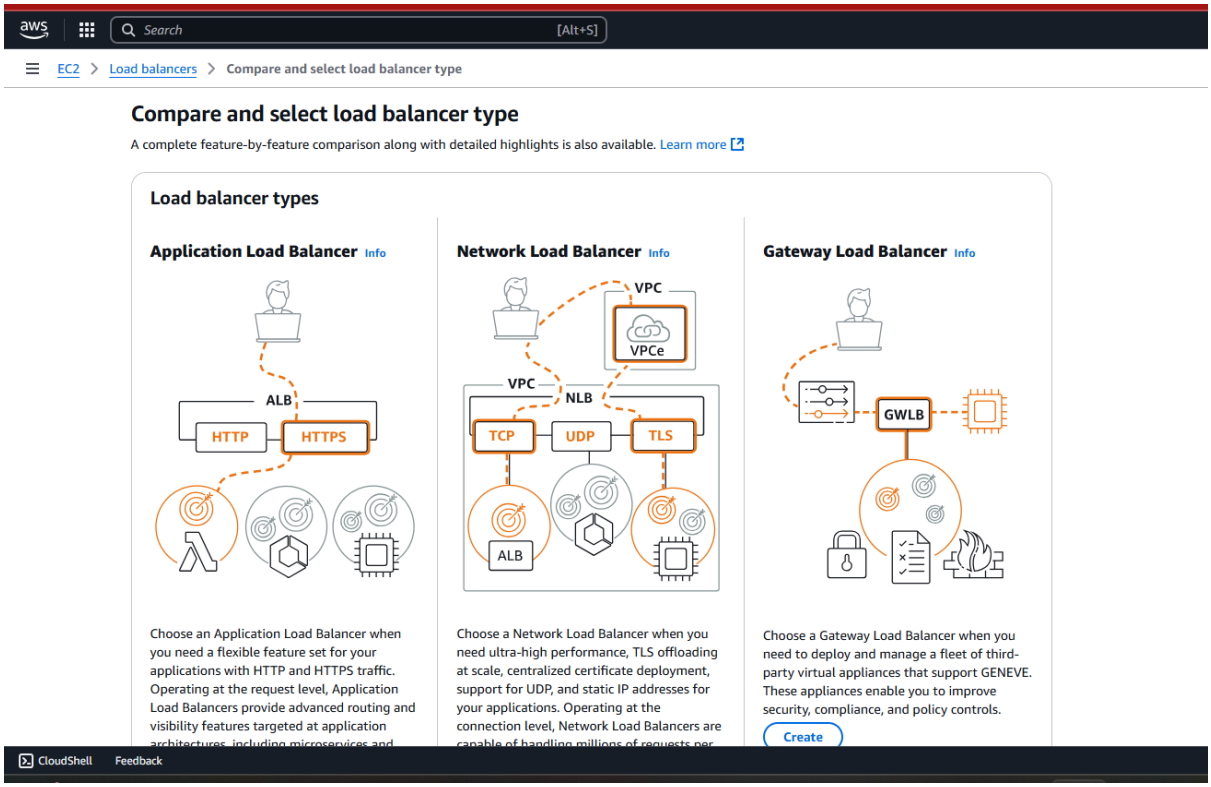


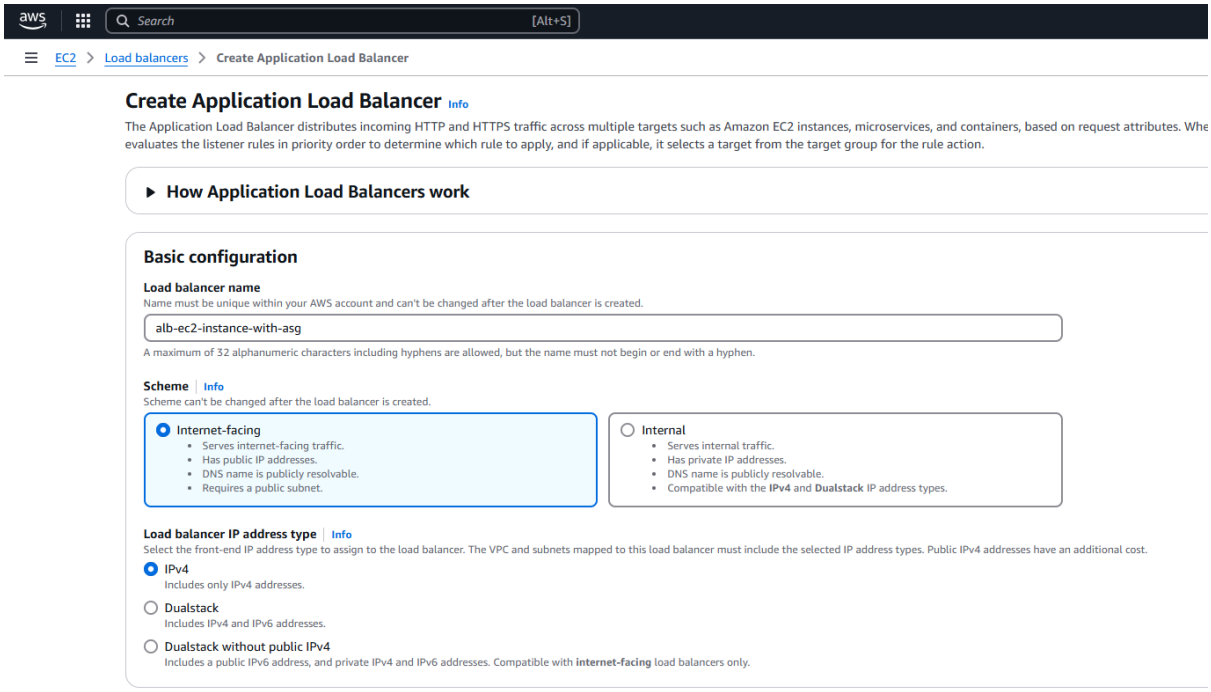


- Target group created.
- As we can see there is no load balancer associated to the target group.

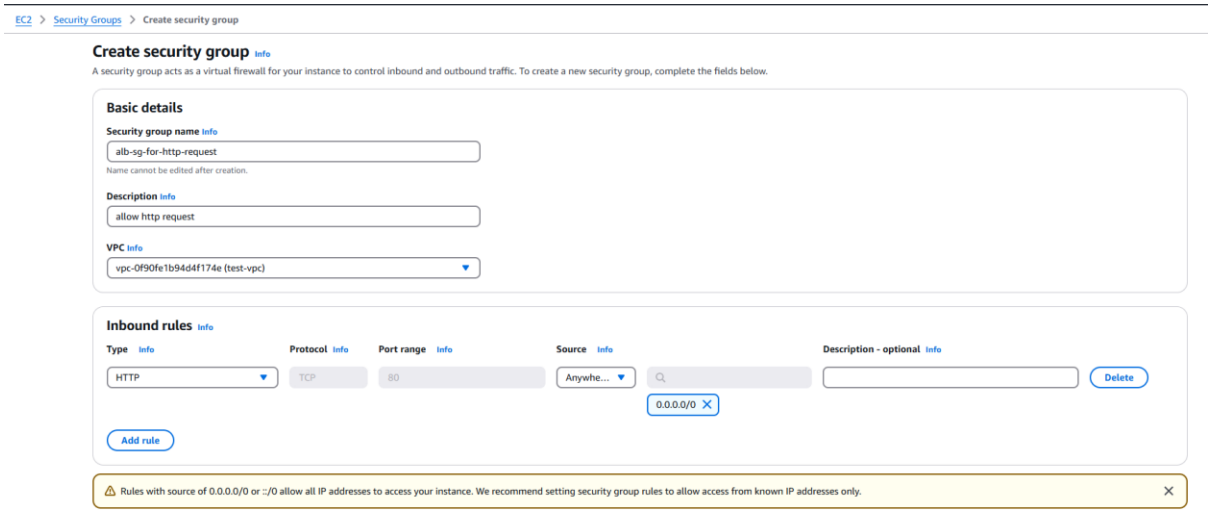
STEP 6) Create load balancer

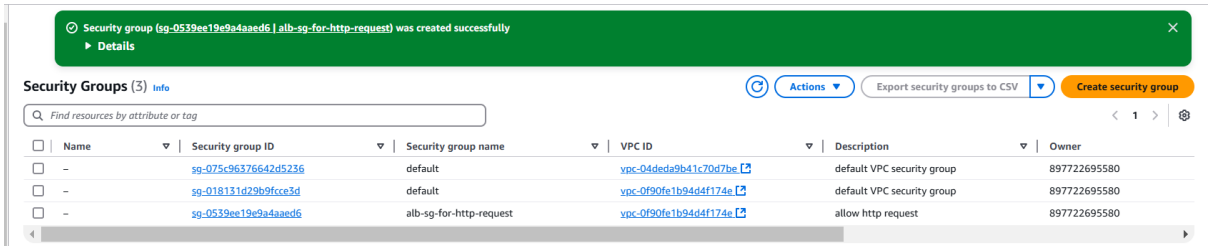
- Select and create Application load balancer (“***alb-ec2-instance-with-asg***”)->internet facing



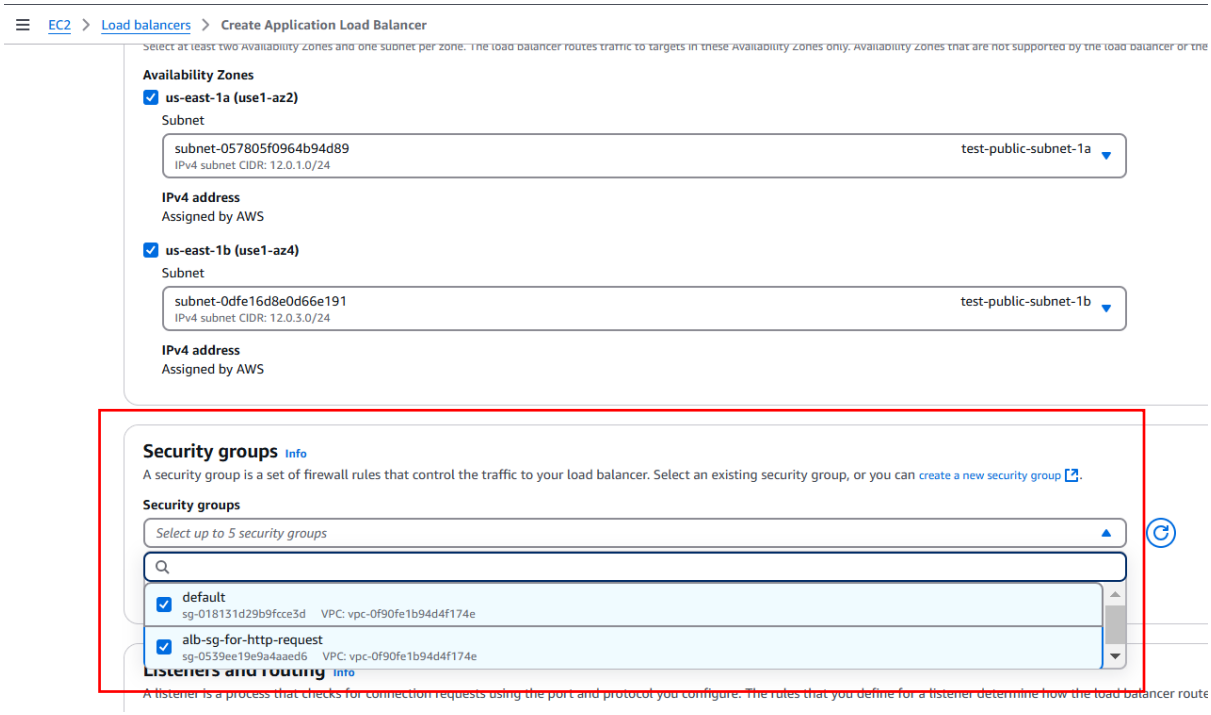


- Create a new security group (“***alg-sg-for-http-request***”) in addition to the default security group.

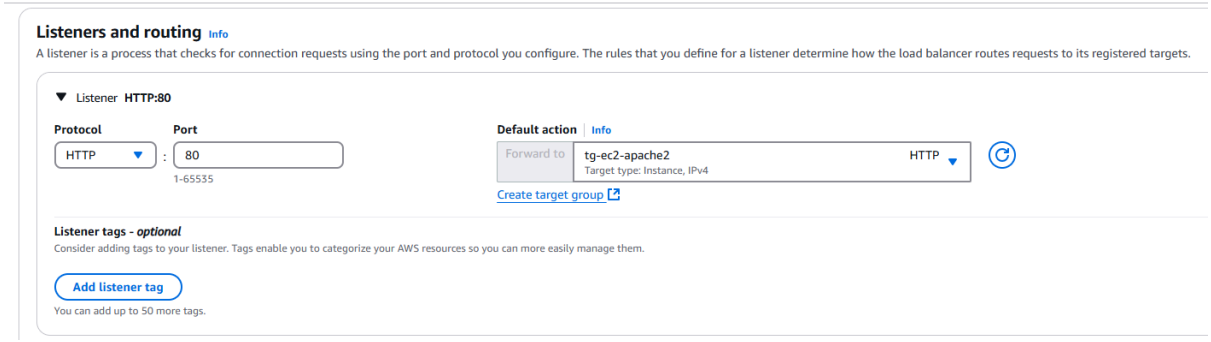




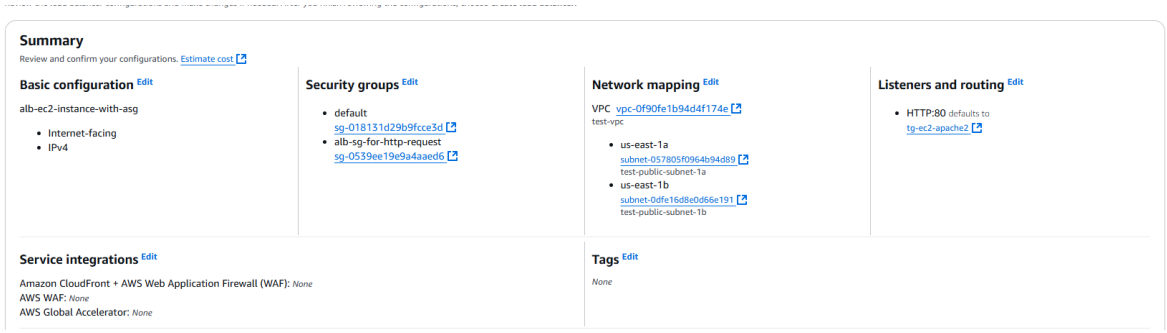
- Select the security groups



- Load balancer needs to point at the target group ("tg-ec2-apache2")
- A ***listener*** checks for connection requests from clients, using the protocol(HTTP) and port (80) that you configure.



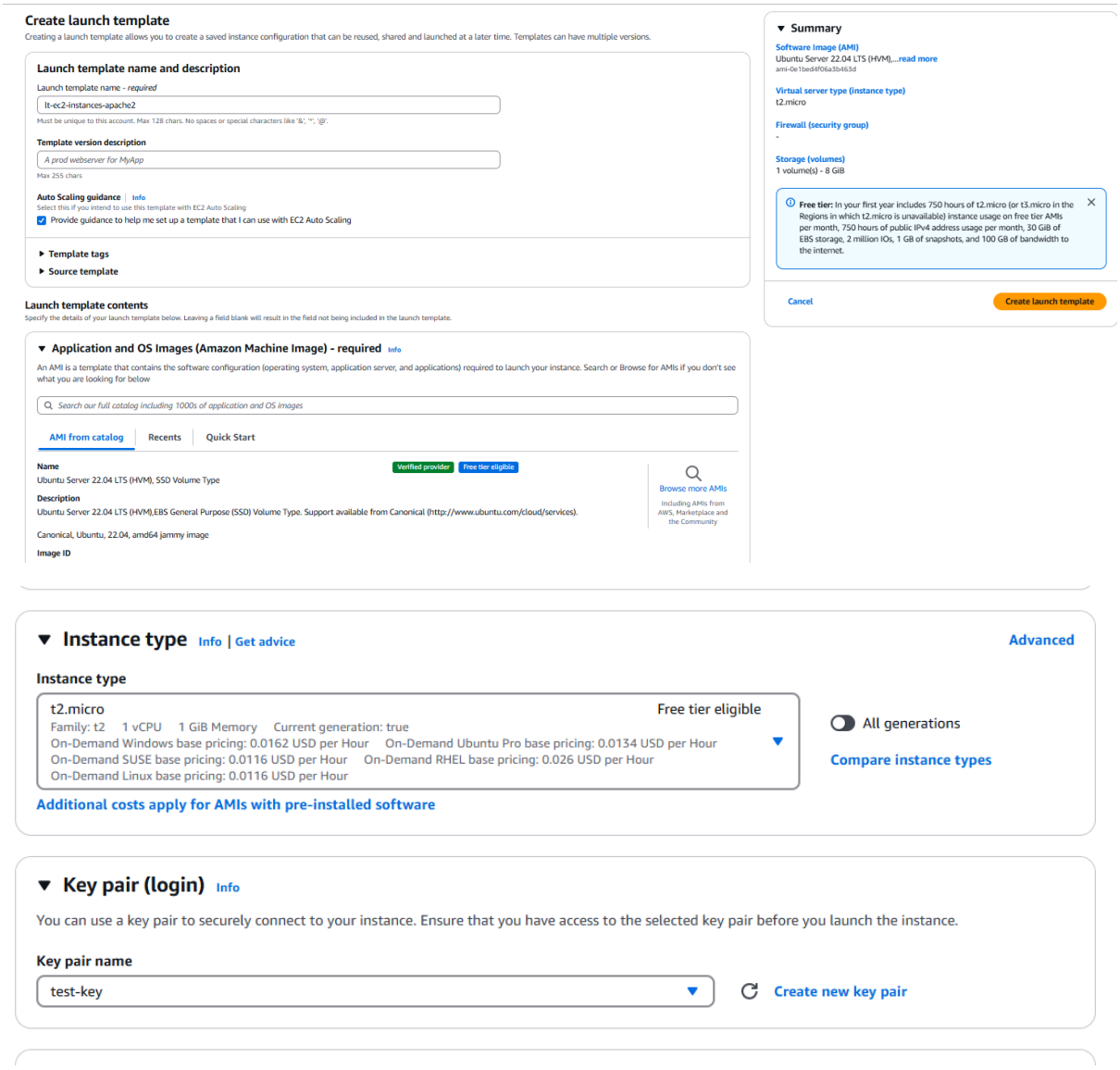
Summary:



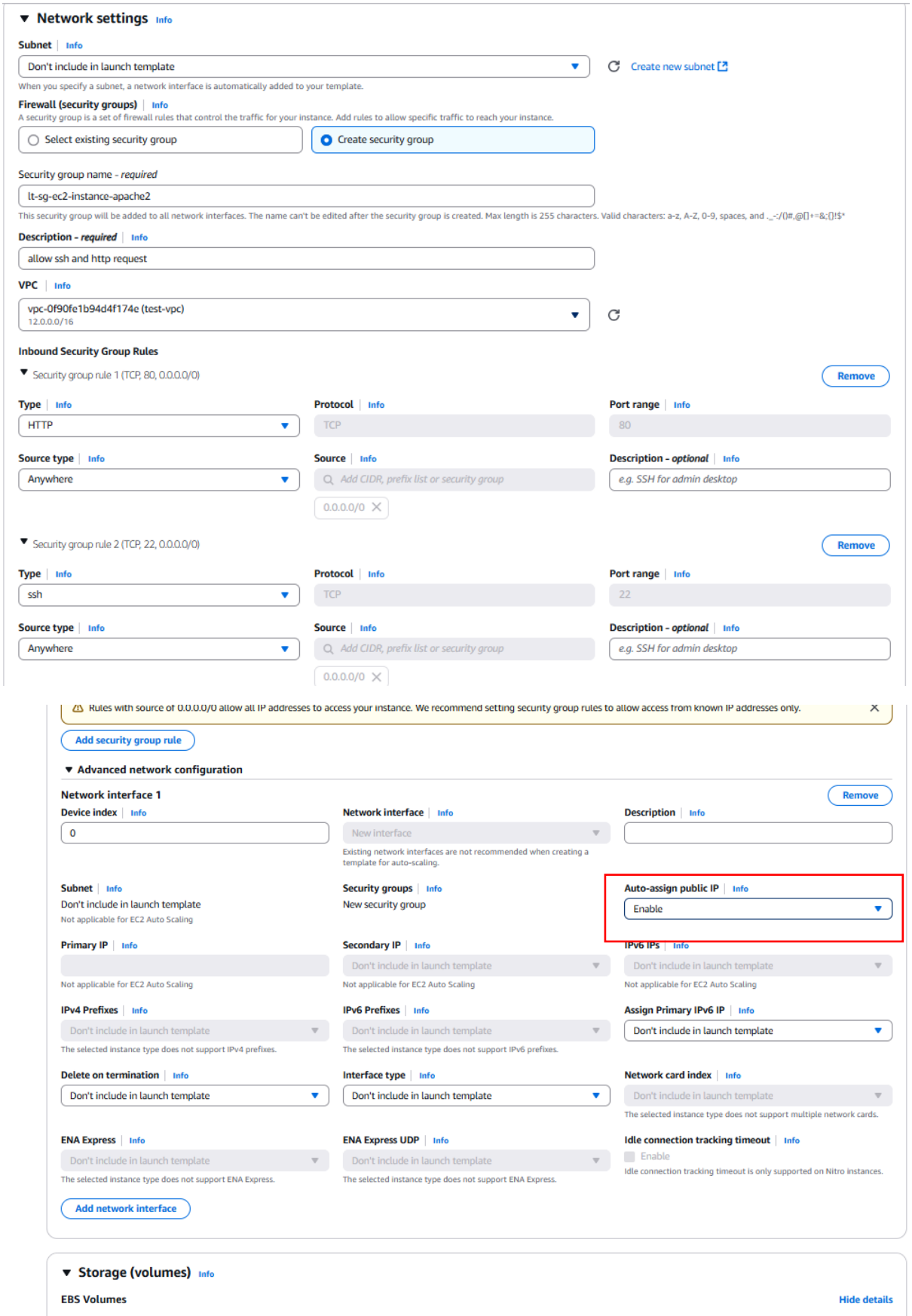
Load balancer created.

STEP 7) Create Auto-scaling group

- Launch template name (“**lt-ec2-instance-apache2**”).

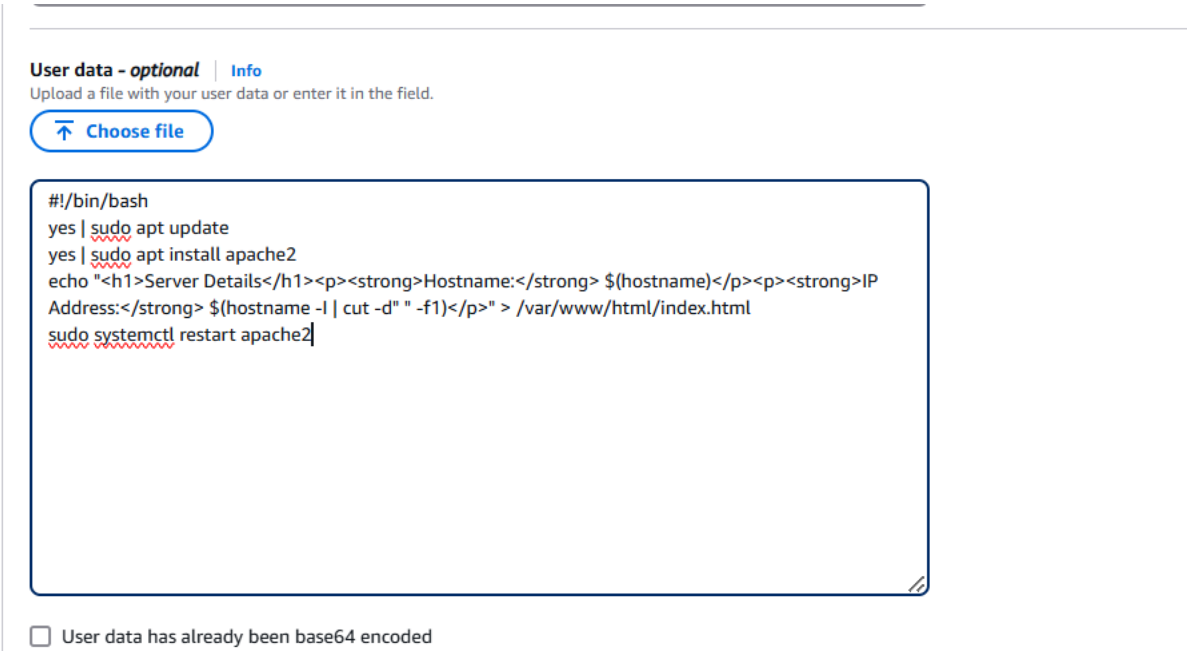


Network settings:

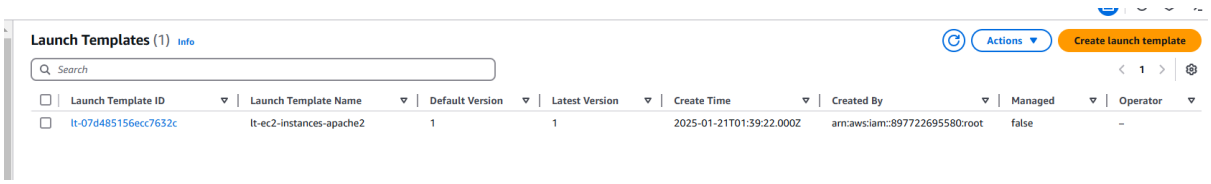


In advanced details under user data, paste the code to install apache server in ec2.

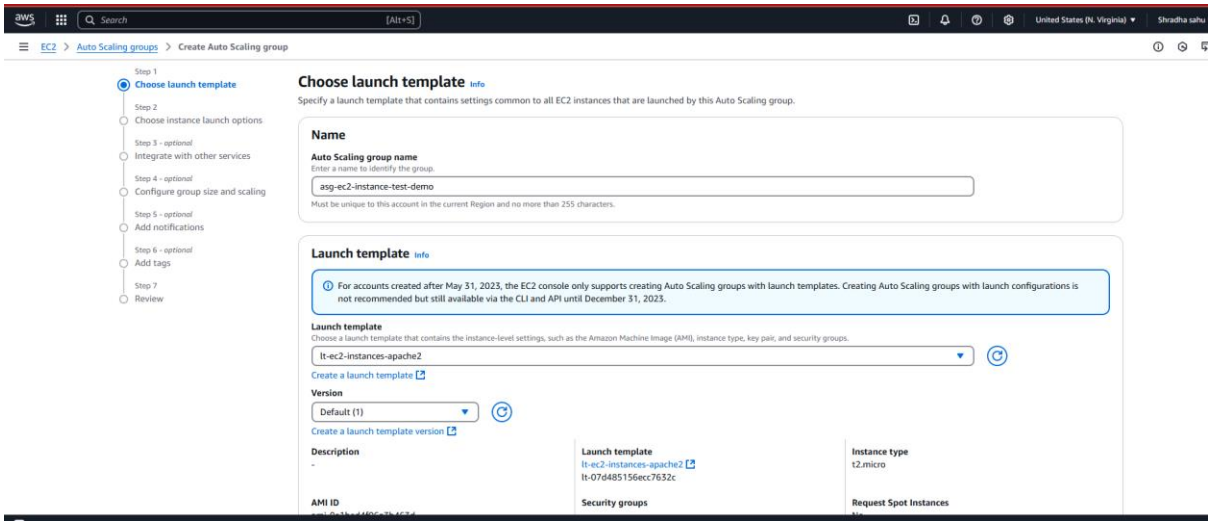
```
#!/bin/bash
yes | sudo apt update
yes | sudo apt install apache2
echo "<h1>Server Details</h1><p><strong>Hostname:</strong> $(hostname)
</p><p><strong>IP Address:</strong> $(hostname -I | cut -d" " -f1)</p>" >
/var/www/html/index.html
sudo systemctl restart apache2
```



- Launch template create



- Get back to the auto scaling group page after the launch template is created
- Auto Scaling group name (“***asg-ec2-instance-test-demo***“)->select the launch template created.



Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0f90fe1b94d4f174e (test-vpc)
12.0.0.0/16

Create a VPC

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-east-1a | subnet-057805f0964b94d89 (test-public-subnet-1a)
12.0.1.0/24

us-east-1b | subnet-0dfe16d8e0d66e191 (test-public-subnet-1b)
12.0.3.0/24

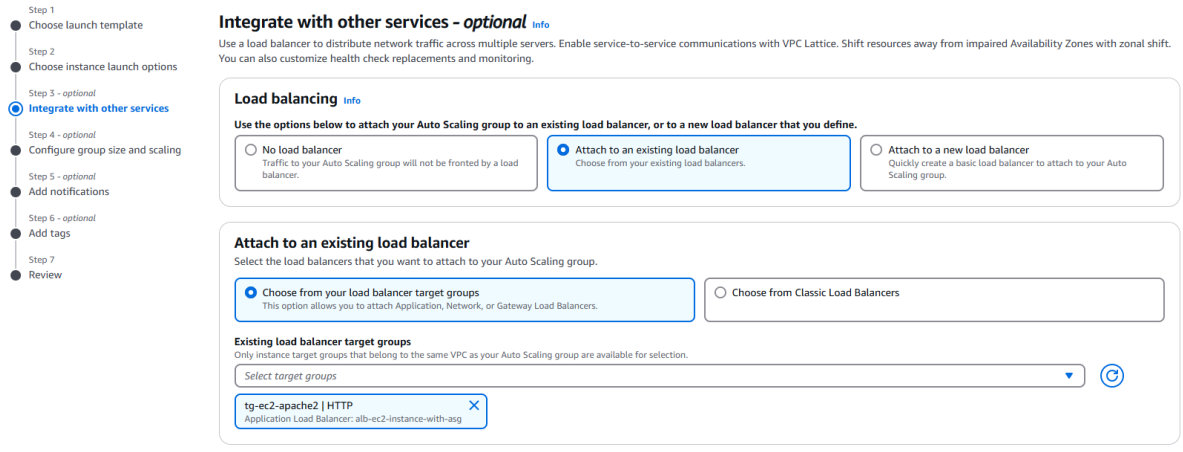
Create a subnet

Availability Zone distribution - *new*

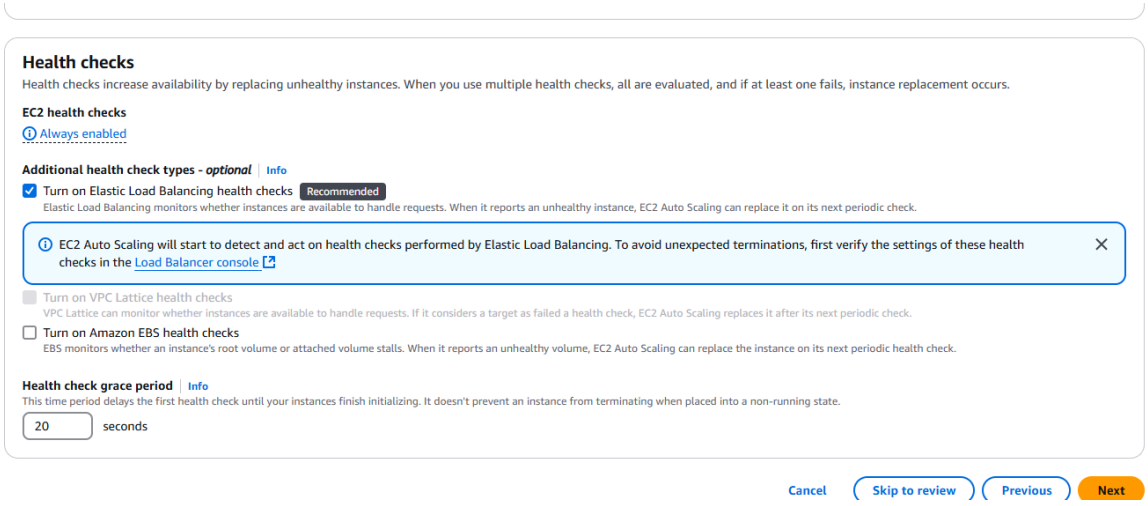
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

- (●) Balanced best effort
  If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.
- ( ) Balanced only
  If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

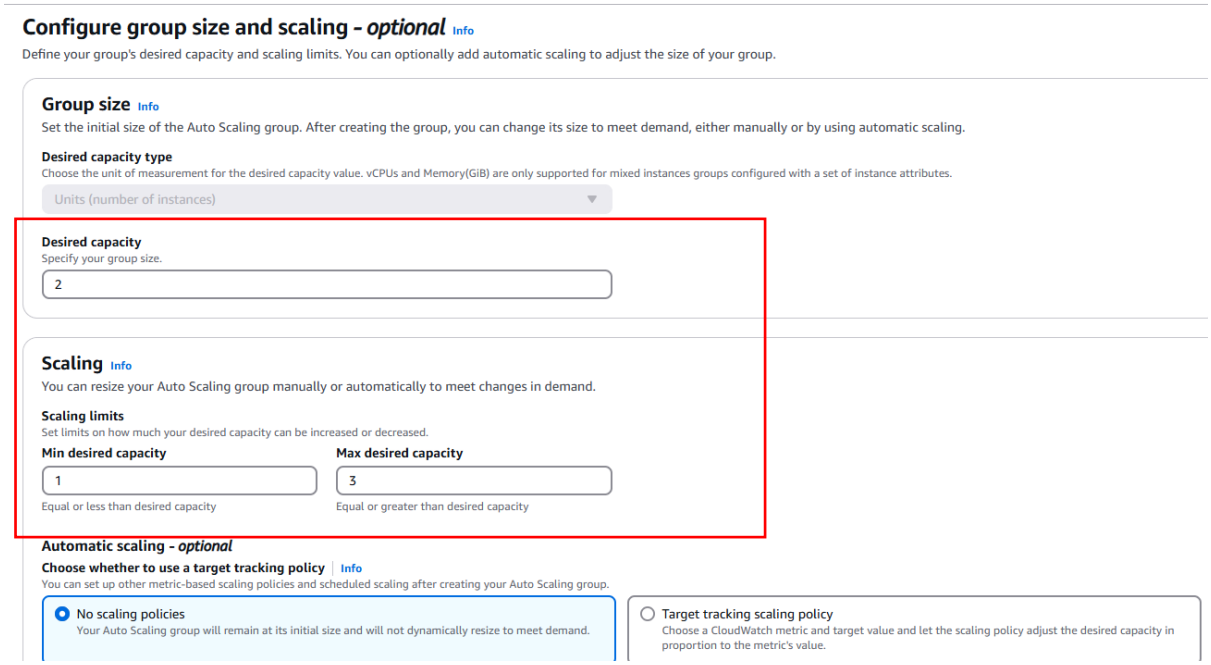
Cancel | Skip to review | Previous | Next

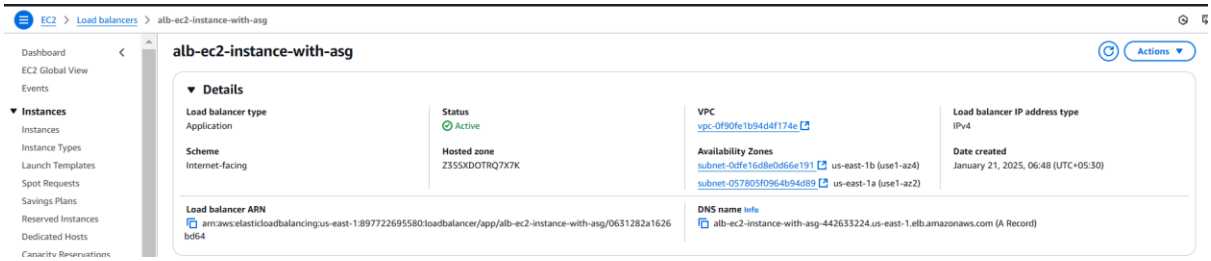


Enable health check

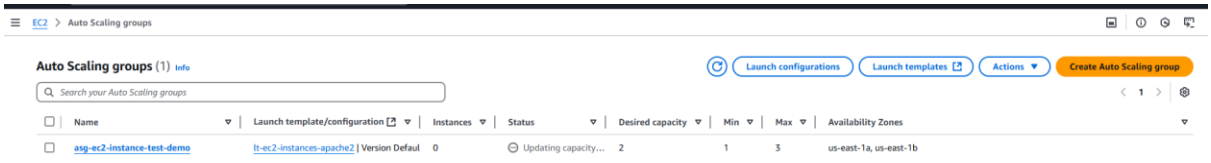


- Configure capacity:
  ->Desired-2, Min-1, Max-3

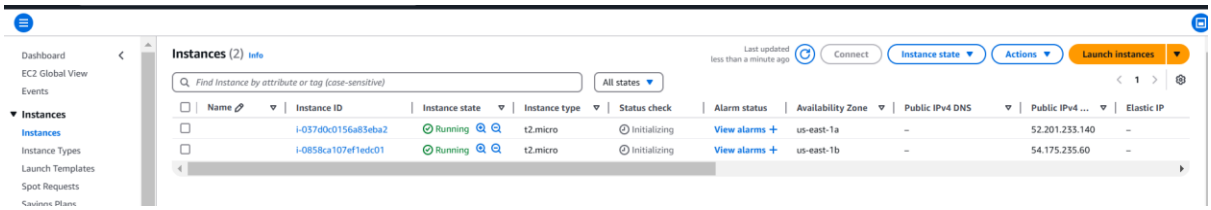




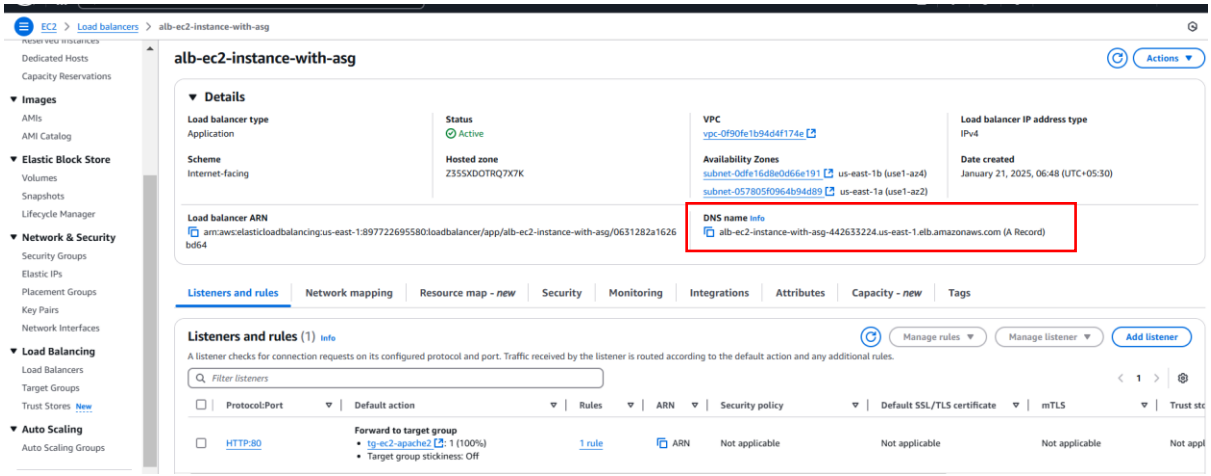
- Autoscaling group created

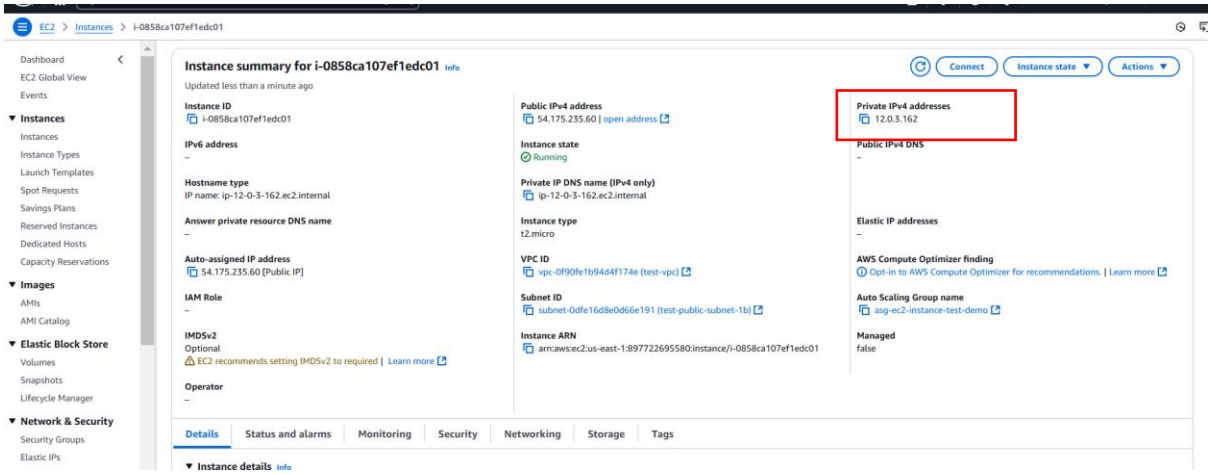


- Autoscaling group will automatically start creating EC2 instances based on the desired capacity(here, 2).
- Check desired capacity ec2 instance provisioned.

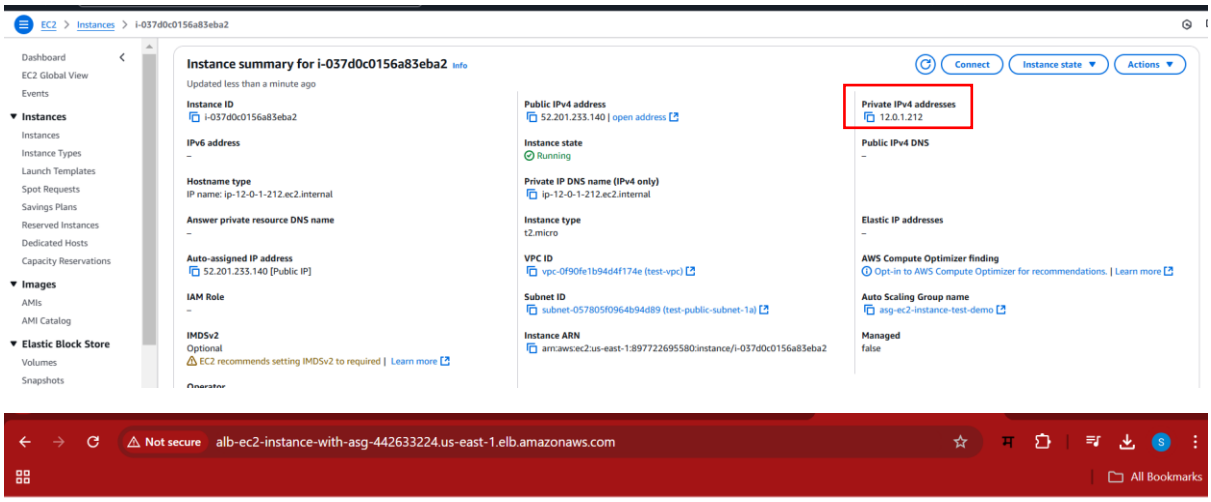


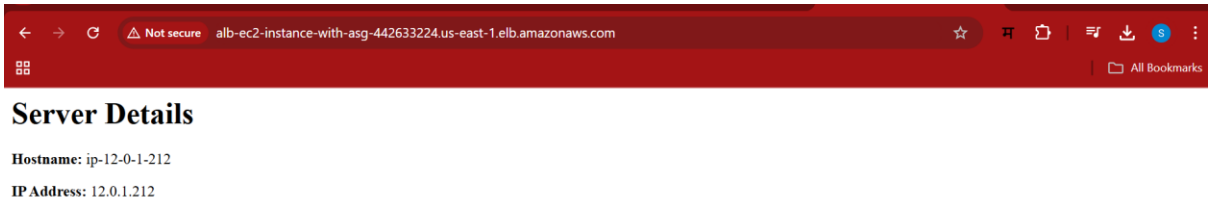
- To check, go to load balancer created and copy the dns name and paste it in new tab





- We are able to access and see the server details

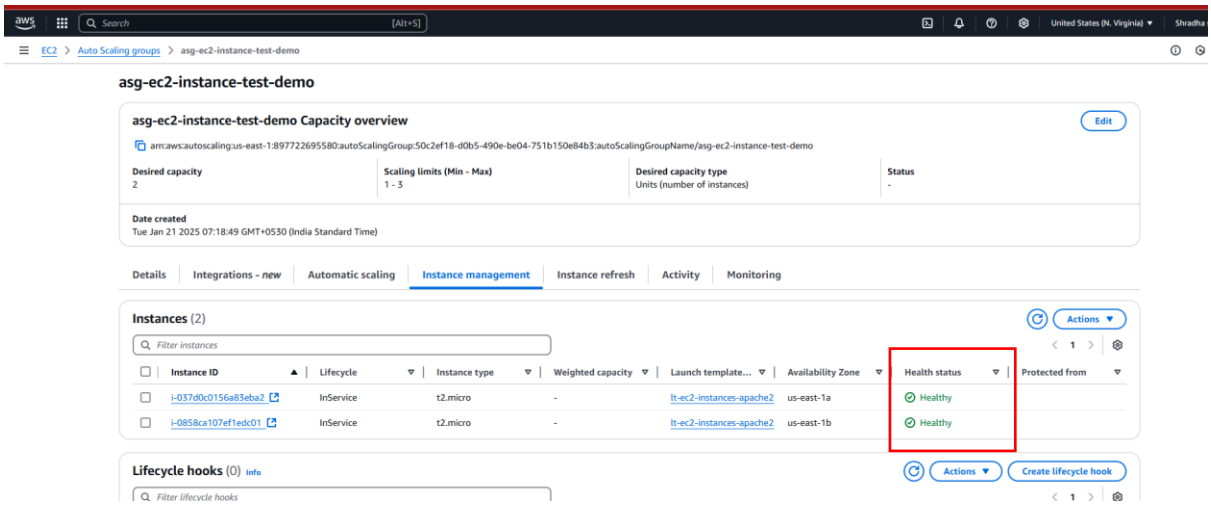




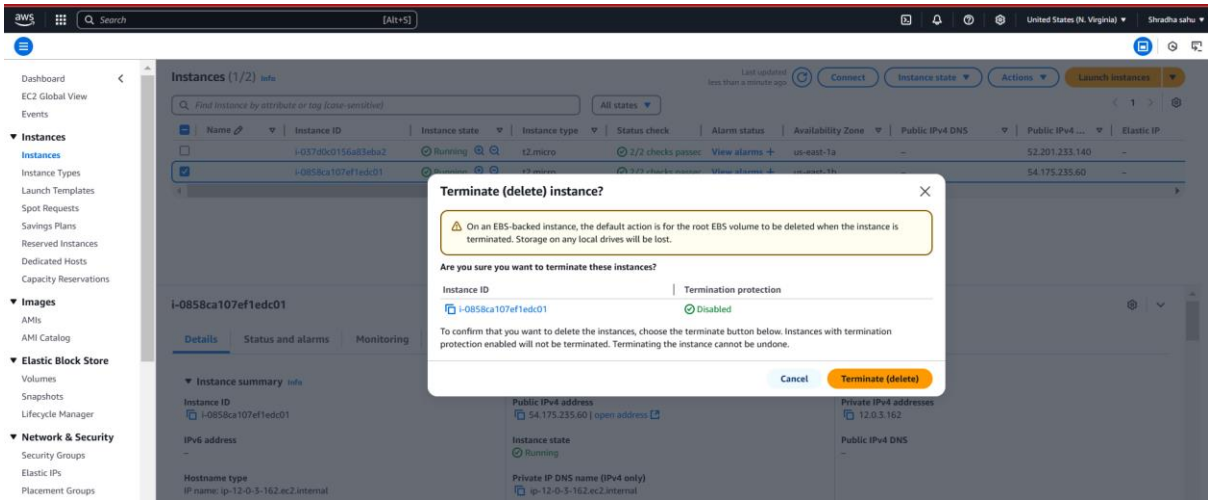
- The ec2 instance private ip and the server details private ip given is same for both ec2 servers

Check how autoscaling works:

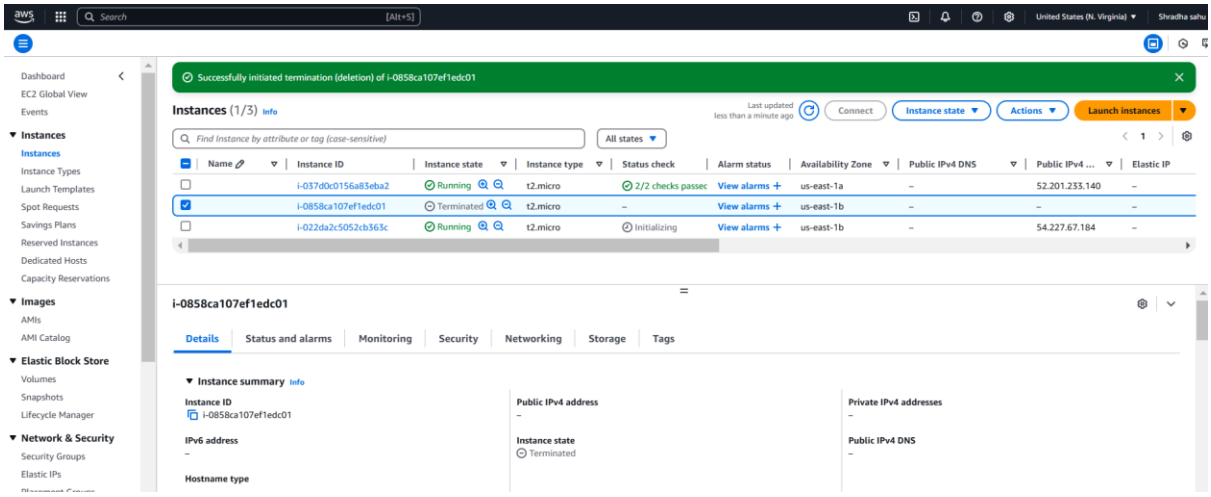
- In Autoscaling both servers are shown in healthy state



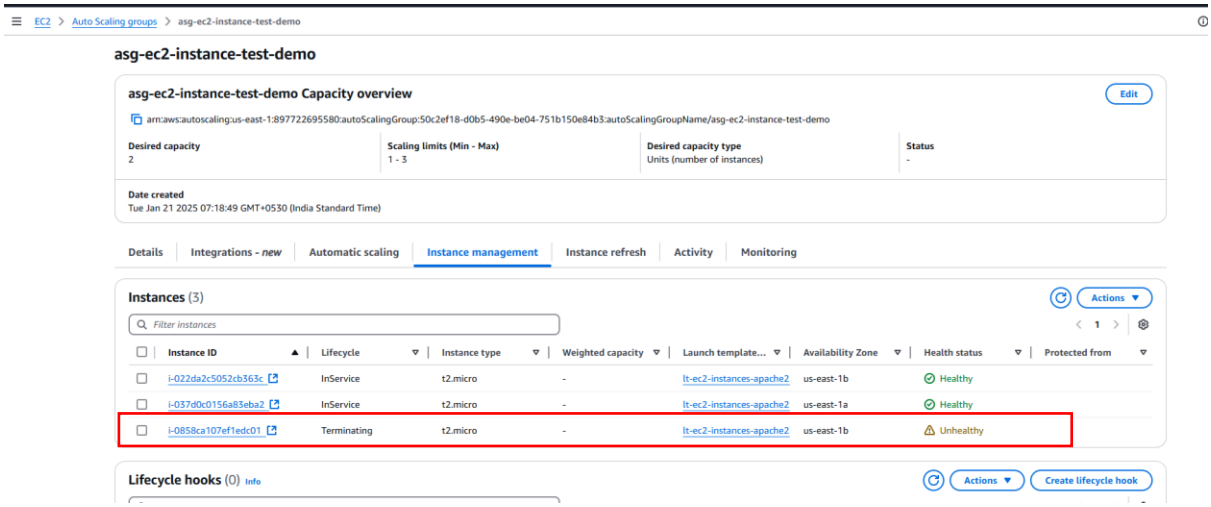
- Now, manually terminate one instance,



- It has started provisioning new instance



- We can see the terminated instance showing unhealthy state



- Autoscaling group has automatically created EC2 instances based on the desired capacity(here, 2) even after the one instance is terminated.