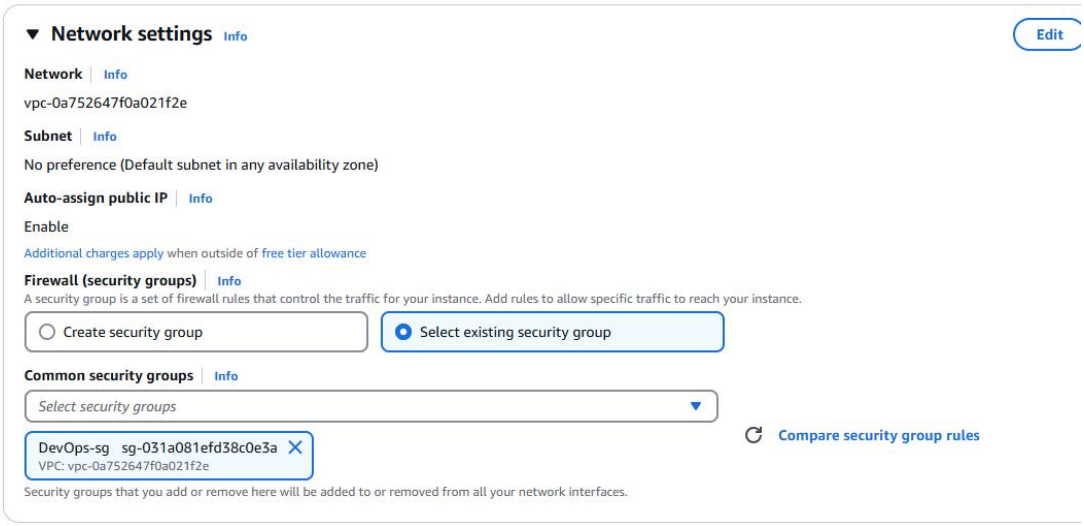
AWS EC2 conclusion:

Load balancer -> Autoscaling group
Application Loadbalancer
Network Loadbalancer
Gateway Loadbalancer

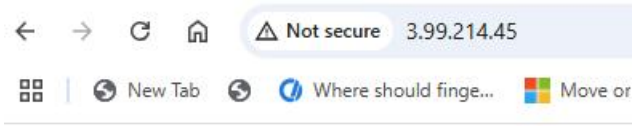
Application LB -> operates at Layer 7, ex: http, https, microservices, advanced routing, path-based-routing.

Network Load balancer: at OSI Layer 4 --> Transport layer --> whenever you want to provide ultra high performance with lowest latency --> Gaming application, video streaming, IOT application -> wherever latency should be minimum
Gateway LB --> works at Layer 3 Network layer: whenever you want to go with Third-party communications like VPN. VPNs, Firewalls, where security concerns are high. Requires high security
Firewall (VPN) ---> Gateway Loadbalancer
| |
S1 S2



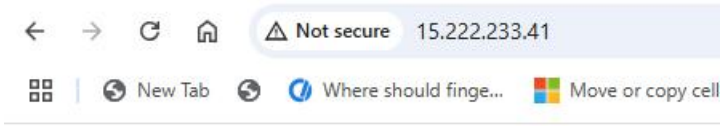

```
#! /bin/bash
sudo su
yum install httpd -y
cd /var/www/html
echo "<html><h1>Banking Server 1</h1></html>" > index.html
systemctl start httpd
```

```
#! /bin/bash
sudo su
yum install httpd -y
cd /var/www/html
echo "<html><h1>Banking Server_2</h1></html>" > index.html
systemctl start httpd
```



Banking Service 1

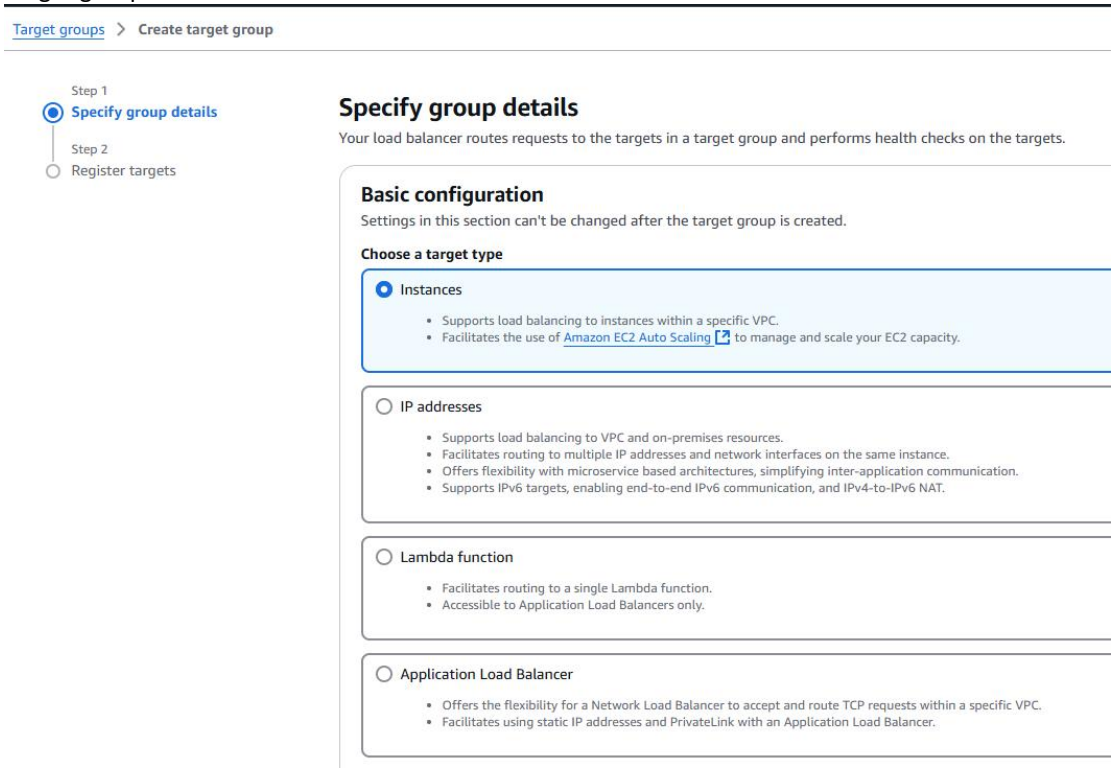
It should be Banking Server 1 and 2 because these are Monolithic applications not Microservices



Banking Service_2

Network LoadBalancer: it is TCP, UDP
Application LoadBalancer: it is HTTP, HTTPS

Target group:



Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you created. This choice cannot be changed after creation

TCP

80

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

- IPv4
  Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.
- IPv6
  Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). Learn more

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

-
vpc-0a752647f0a021f2e
IPv4 VPC CIDR: 172.31.0.0/16

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/

Up to 1024 characters allowed.

▶ **Advanced health check settings**

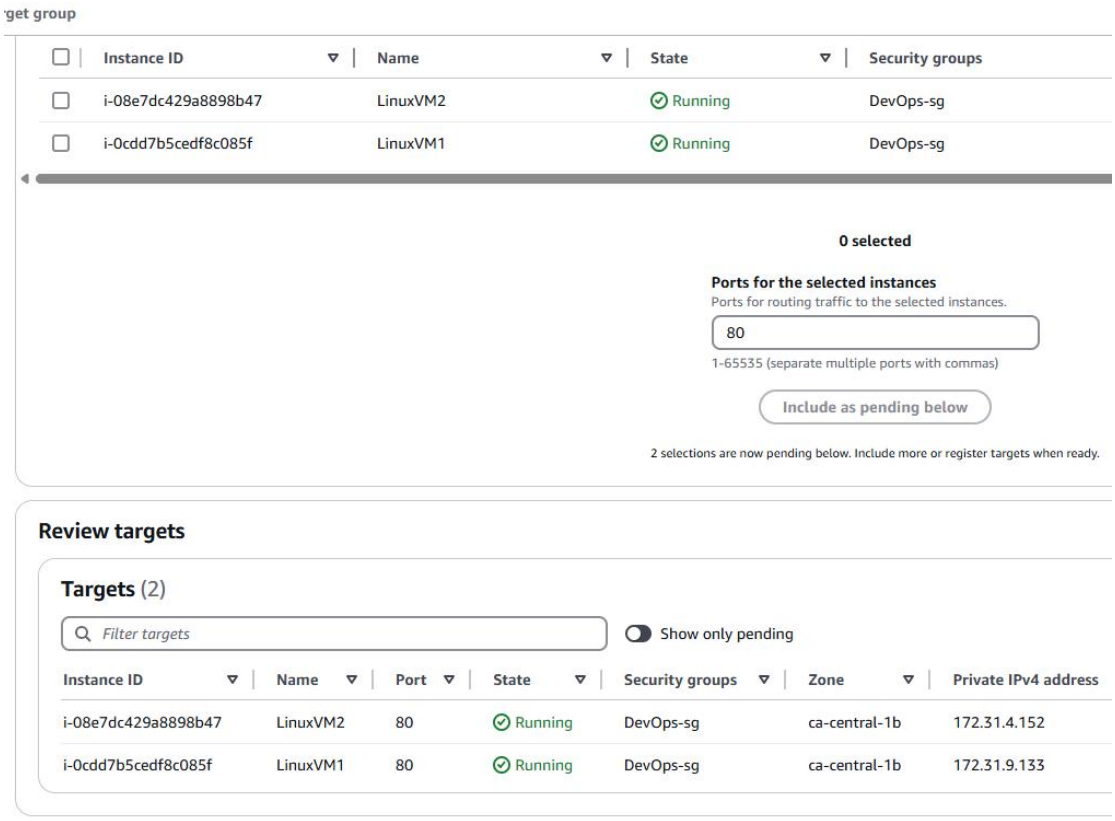
Attributes

ⓘ Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

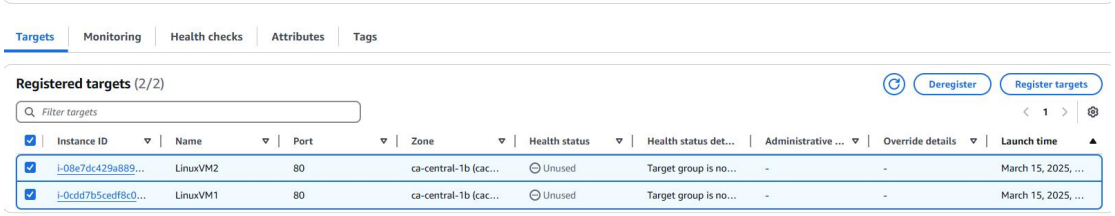
▶ **Tags - *optional***

Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

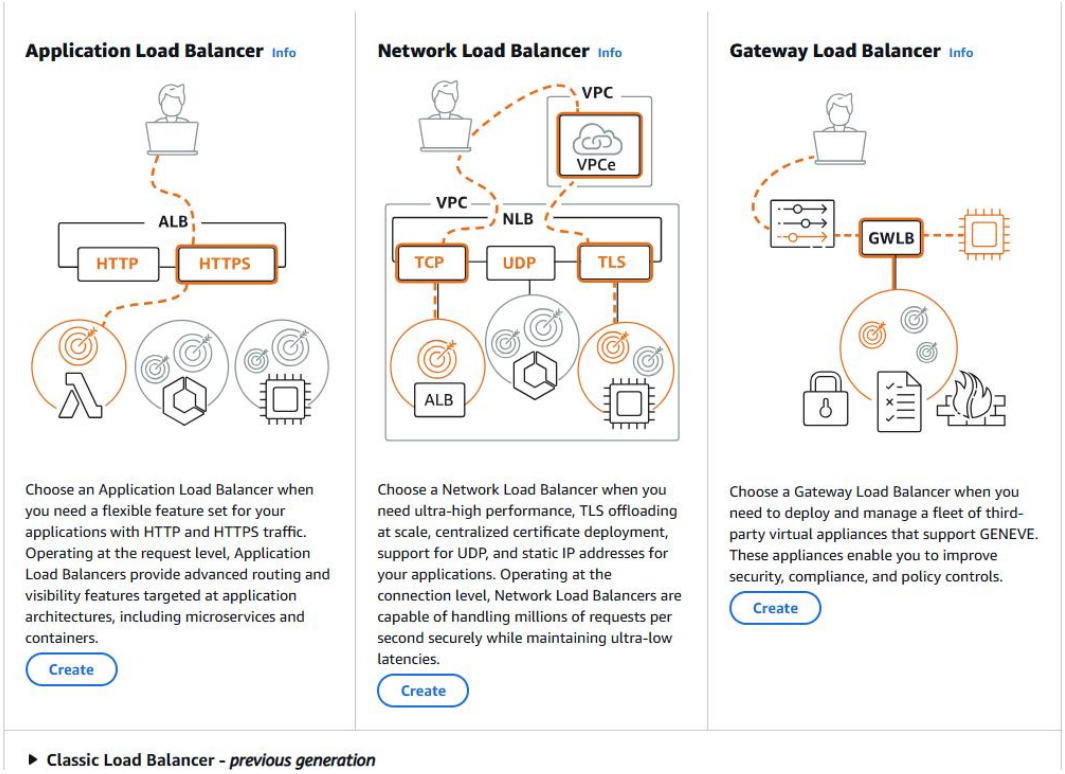
Cancel Next



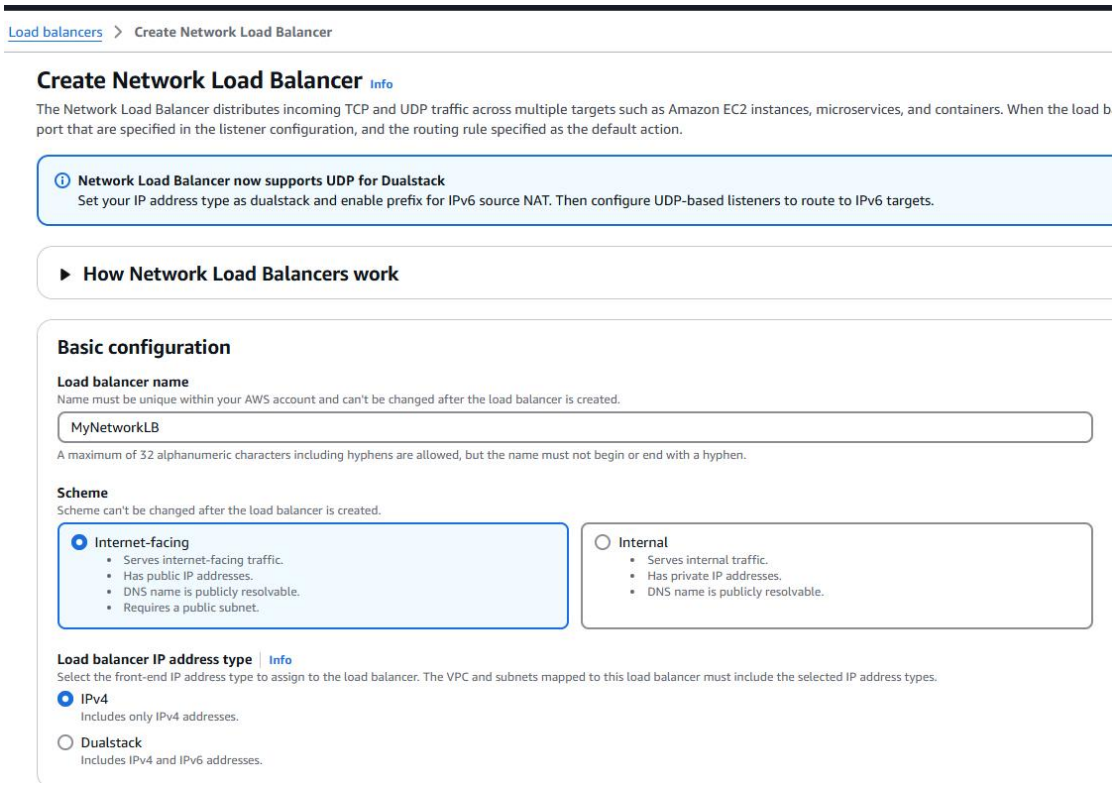
Register targets

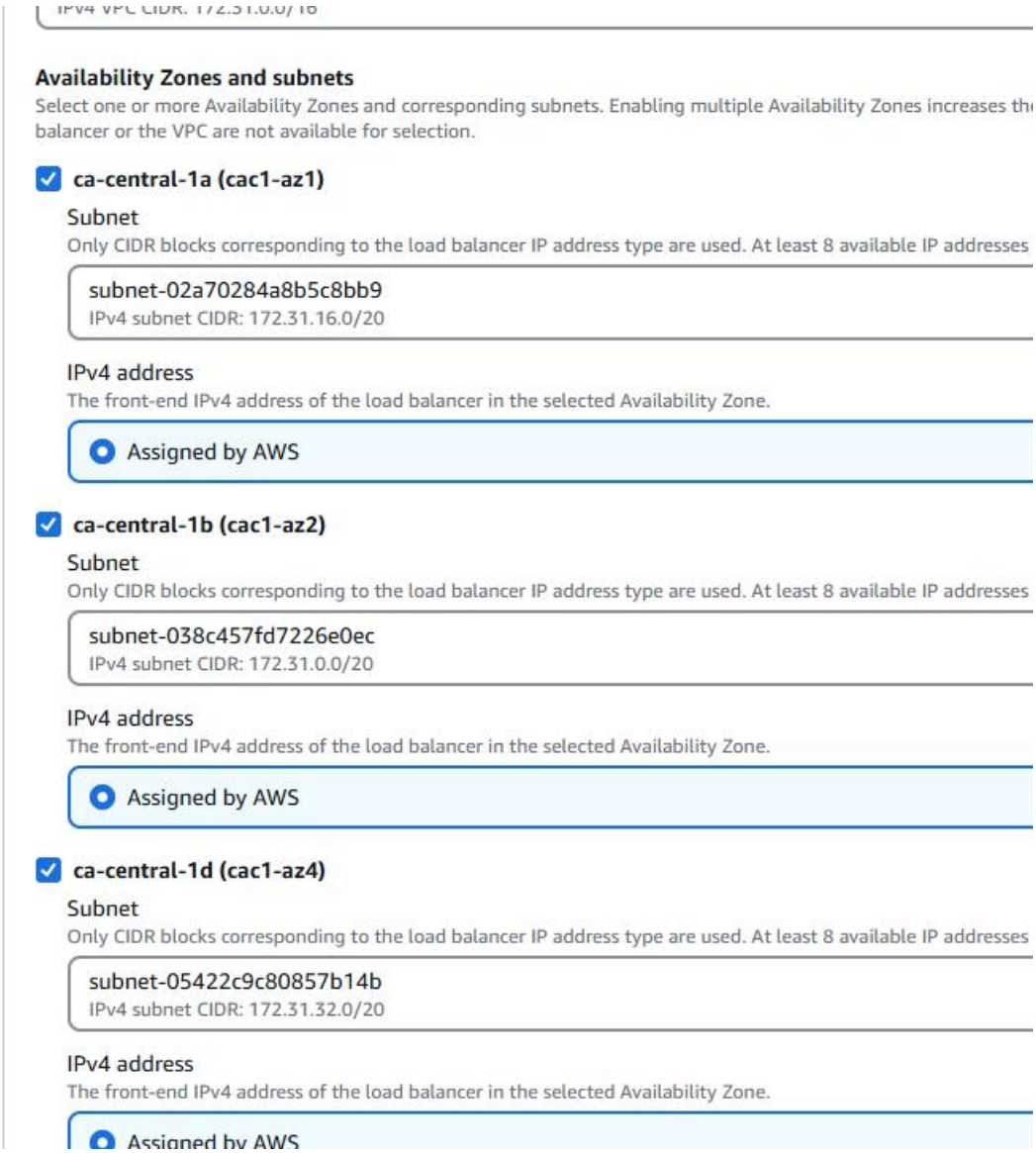


Now go to Load Balancer



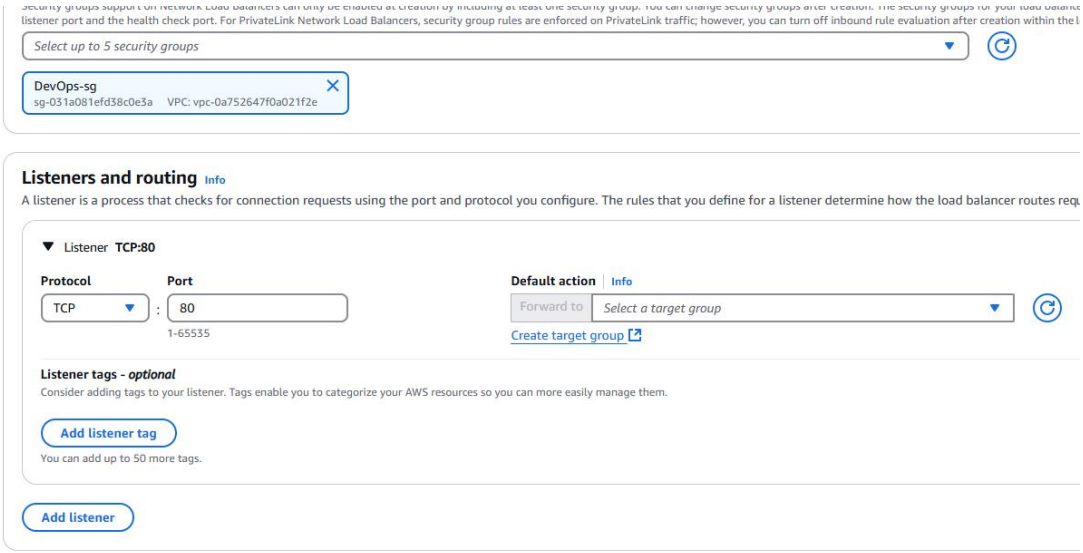
Create Network Loadbalancer



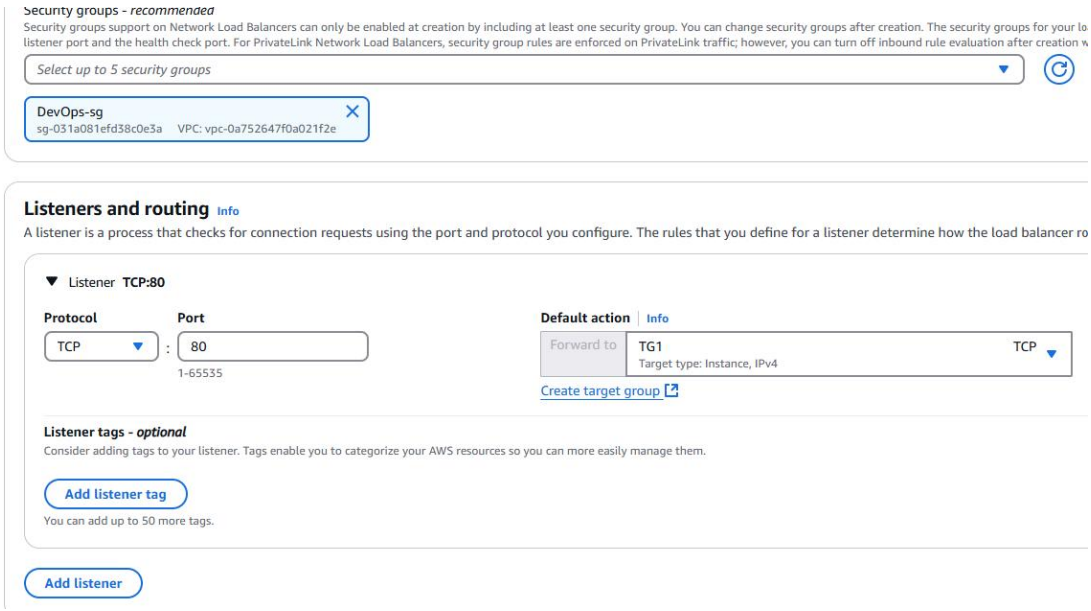


Security group selected

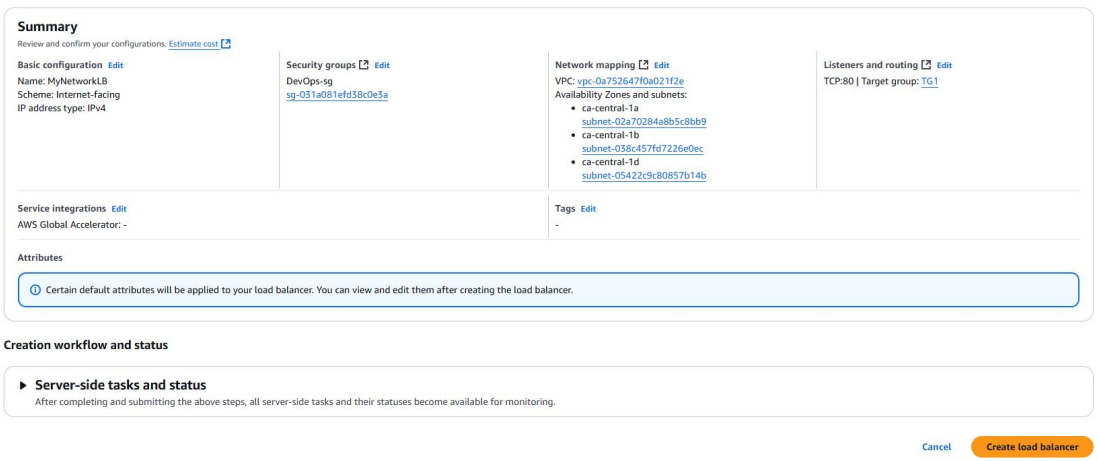
And TCP selected



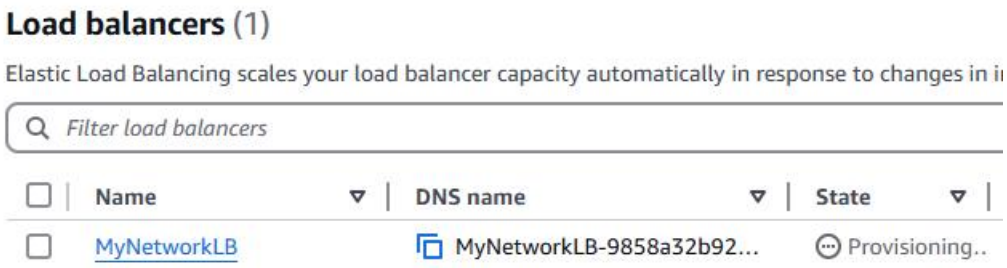
Select Target group

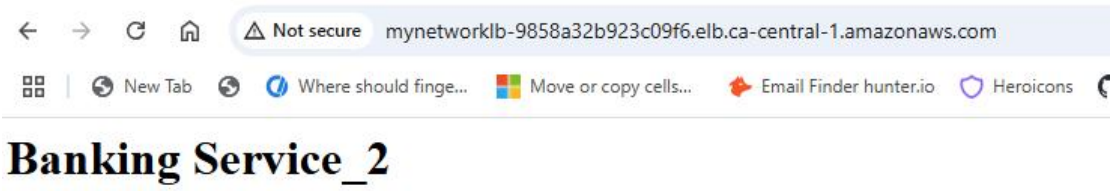


Create load balancer

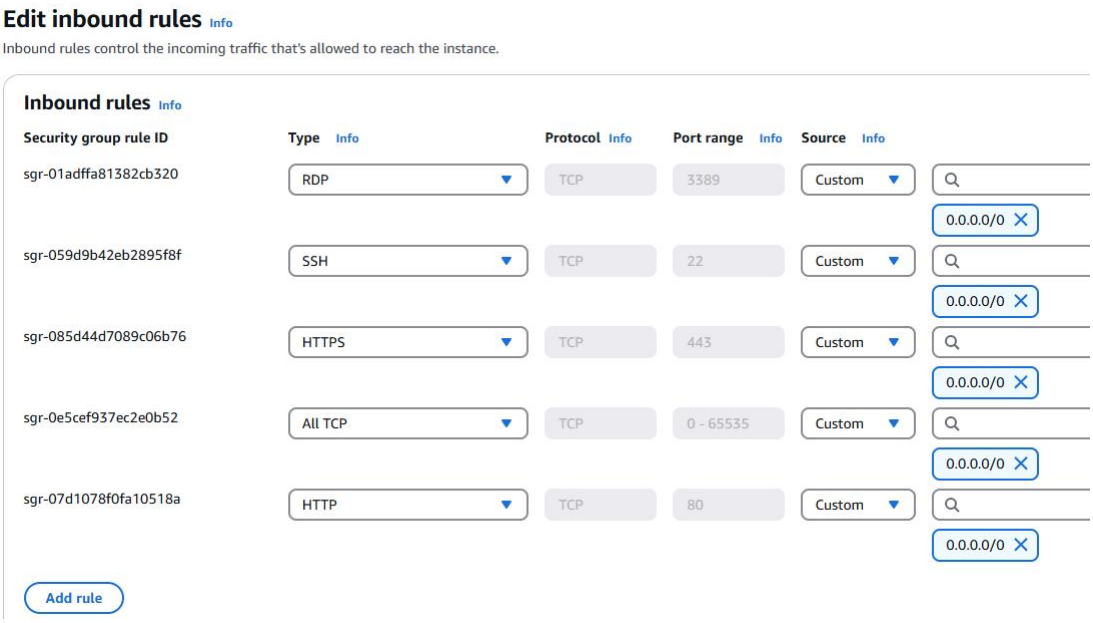


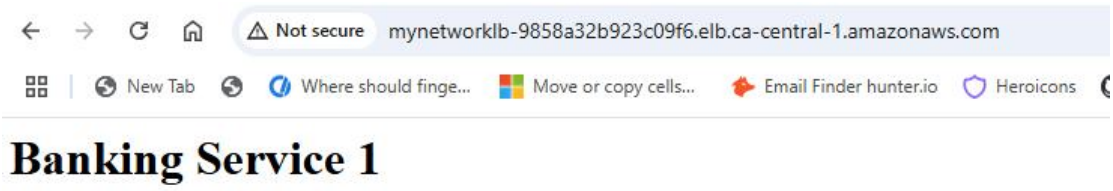
Currently Provisioning





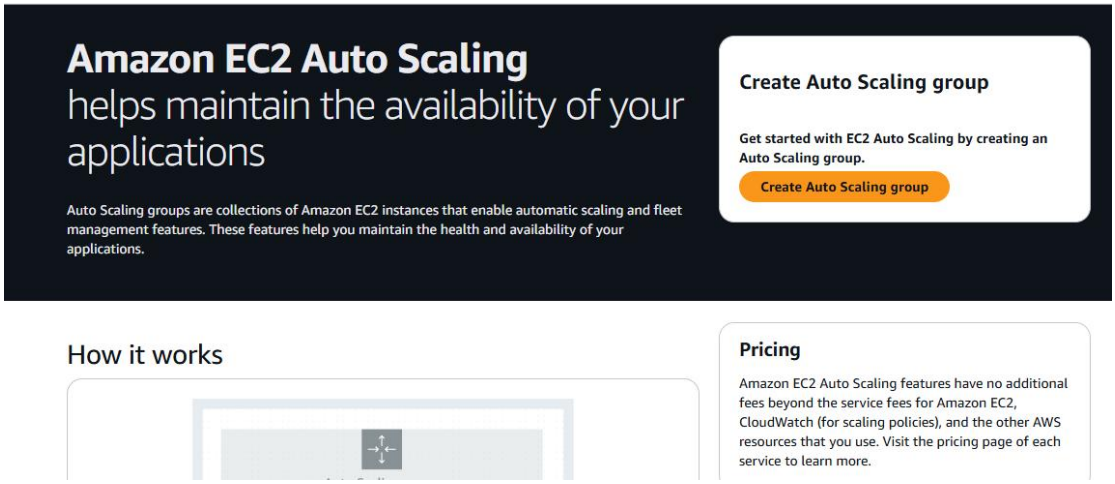
Security groups -> inbound rules
May be add All TCP -> 0.0.0.0/0



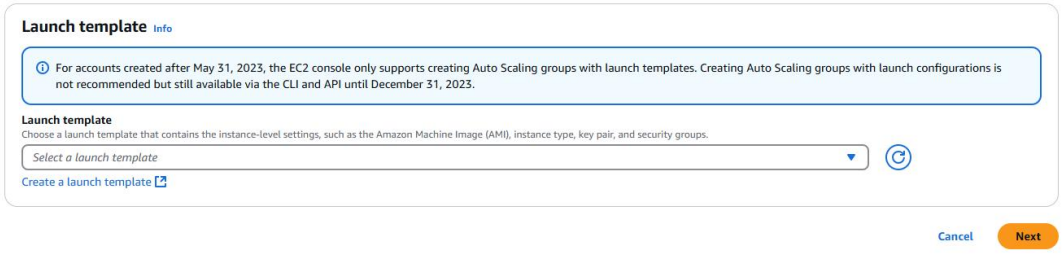


Why we have to go with Auto-scaling?
If you want to modify infrastructure based on the demand, auto-scaling comes into picture

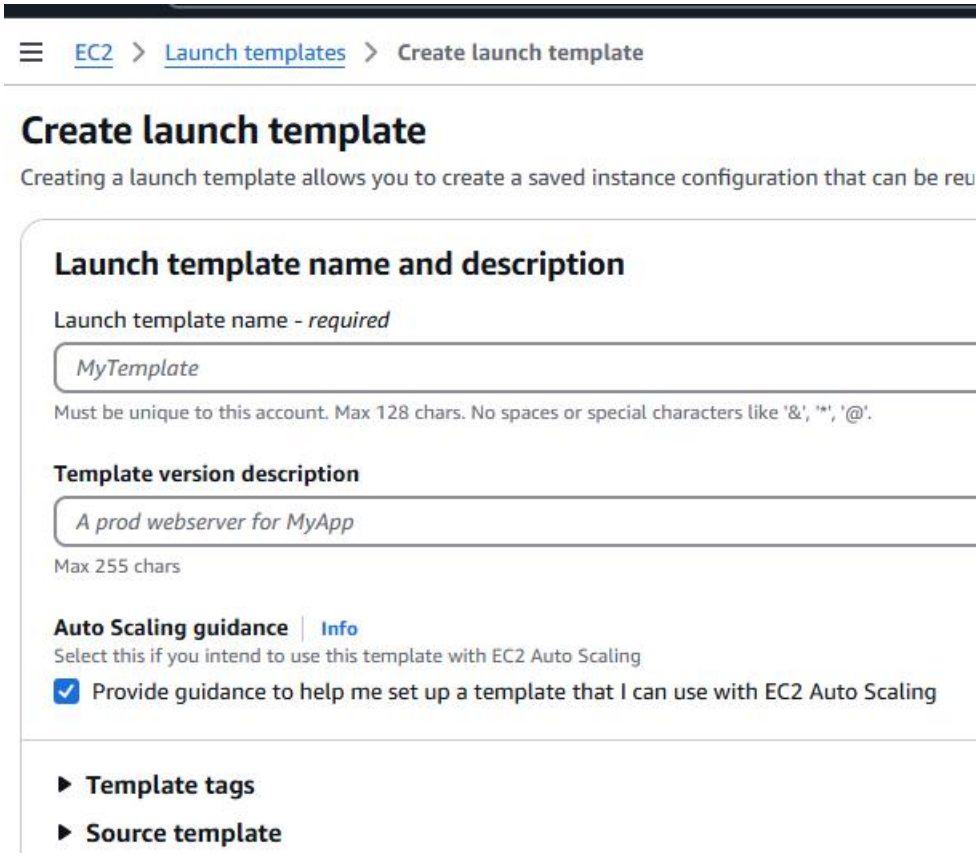
Amazon EC2 Auto Scaling



Create Auto Scaling Group



Click ‘Create a launch template’


EC2 > Launch templates > Create launch template
Create launch template
Creating a launch template allows you to create a saved instance configuration that can be reu
Launch template name and description
Launch template name - required
MyTemplate
Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.
Template version description
A prod webserver for MyApp
Max 255 chars
Auto Scaling guidance | Info
Select this if you intend to use this template with EC2 Auto Scaling
Provide guidance to help me set up a template that I can use with EC2 Auto Scaling
Template tags
Source template

MyTemplate

Must be unique to this account. Max 128 chars. No spaces or sp

Template version description

my_template-v1.0

Max 255 chars

Auto Scaling guidance | Info

Select this if you intend to use this template with EC2 Auto Sca

☑ Provide guidance to help me set up a template that

▶ **Template tags**

▶ **Source template**

Launch template contents

Specify the details of your launch template below. Leaving a field b

▼ Application and OS Images (Amazo

An AMI is a template that contains the software config
don't see what you are looking for below

Search our full catalog including 1000s of applica

Recents | Quick Start

◉ Currently in use

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-0cc3a9edb87c91b53 (64-bit (x86), uefi-preferred) / ami-092dc7d756444f58e (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term supp
performance execution environment to develop and run your cloud applications.

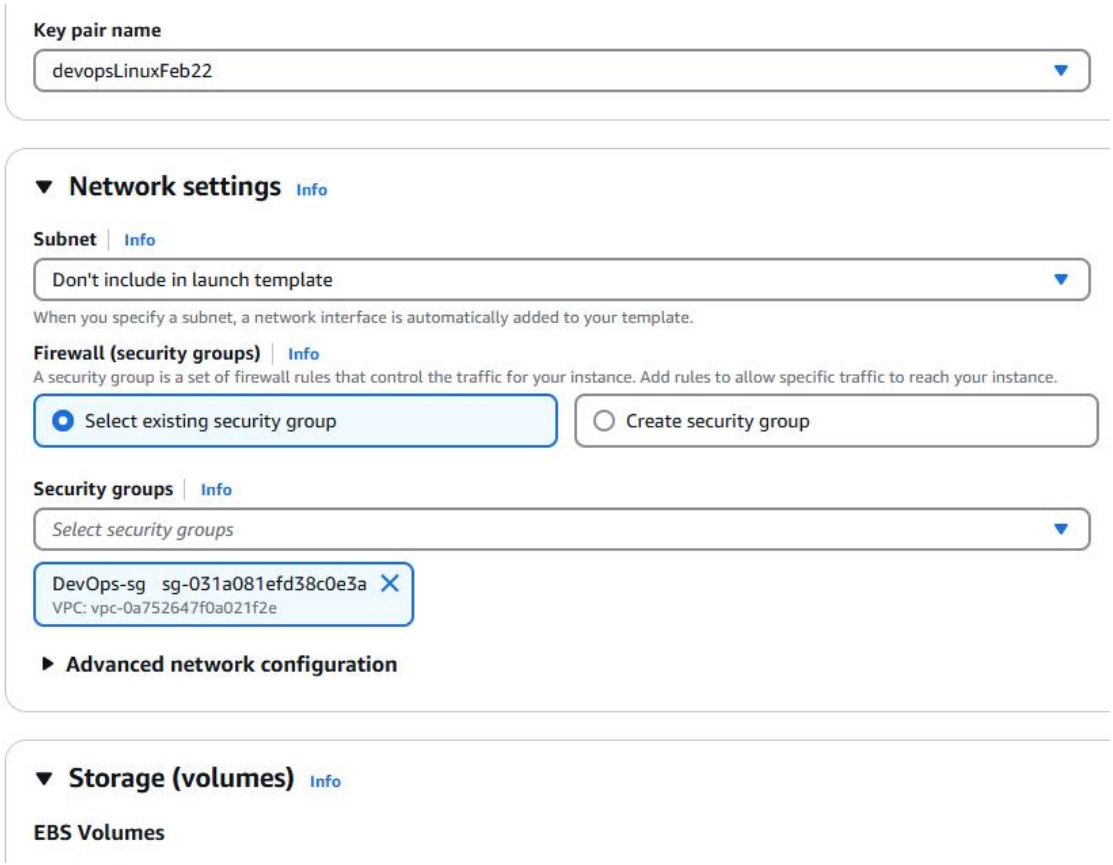
Amazon Linux 2023 AMI 2023.6.20250303.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date
64-bit (x86)	uefi-preferred	ami-0cc3a9edb87c91b53	2025-03-04

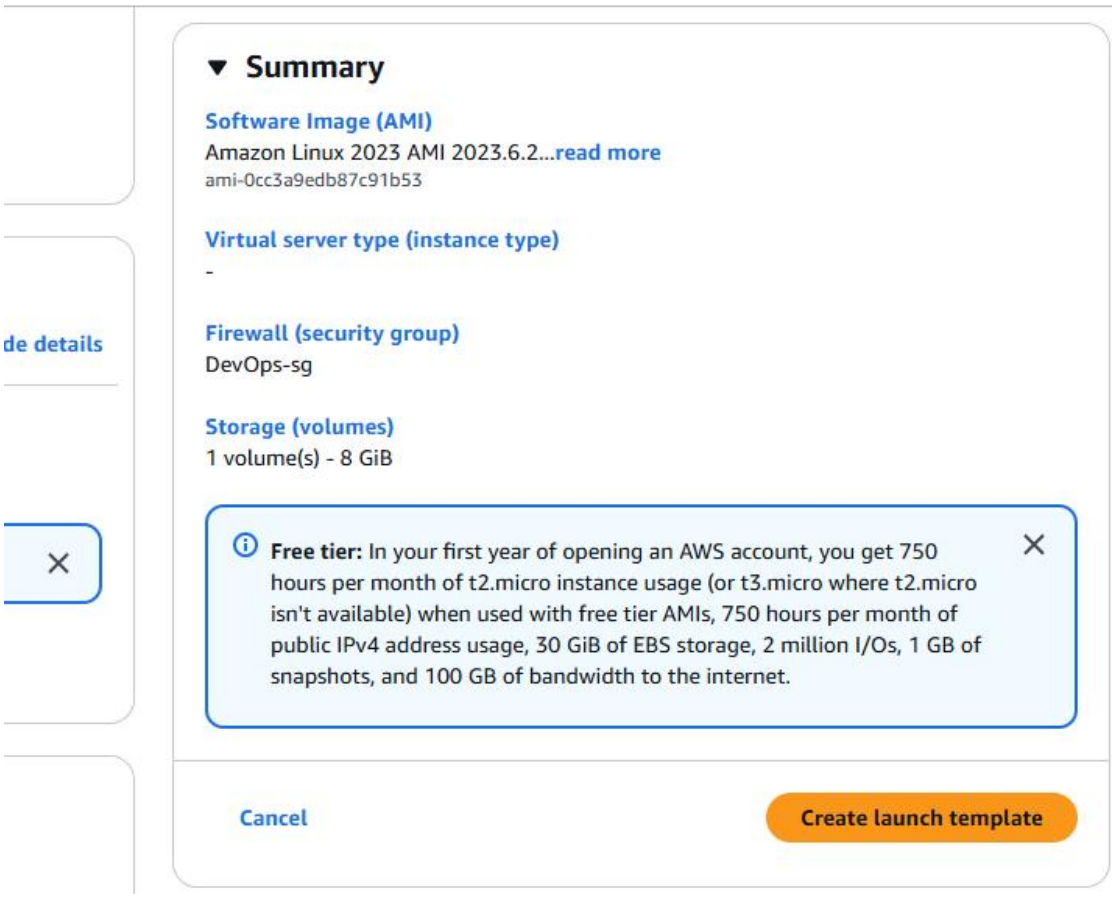
Instance type Info | Get advice

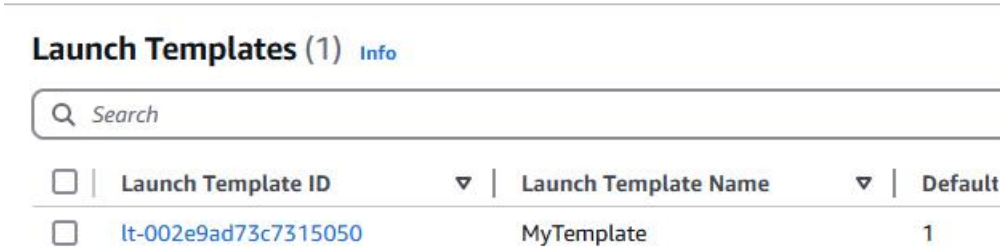
Instance type

Don't include in launch template

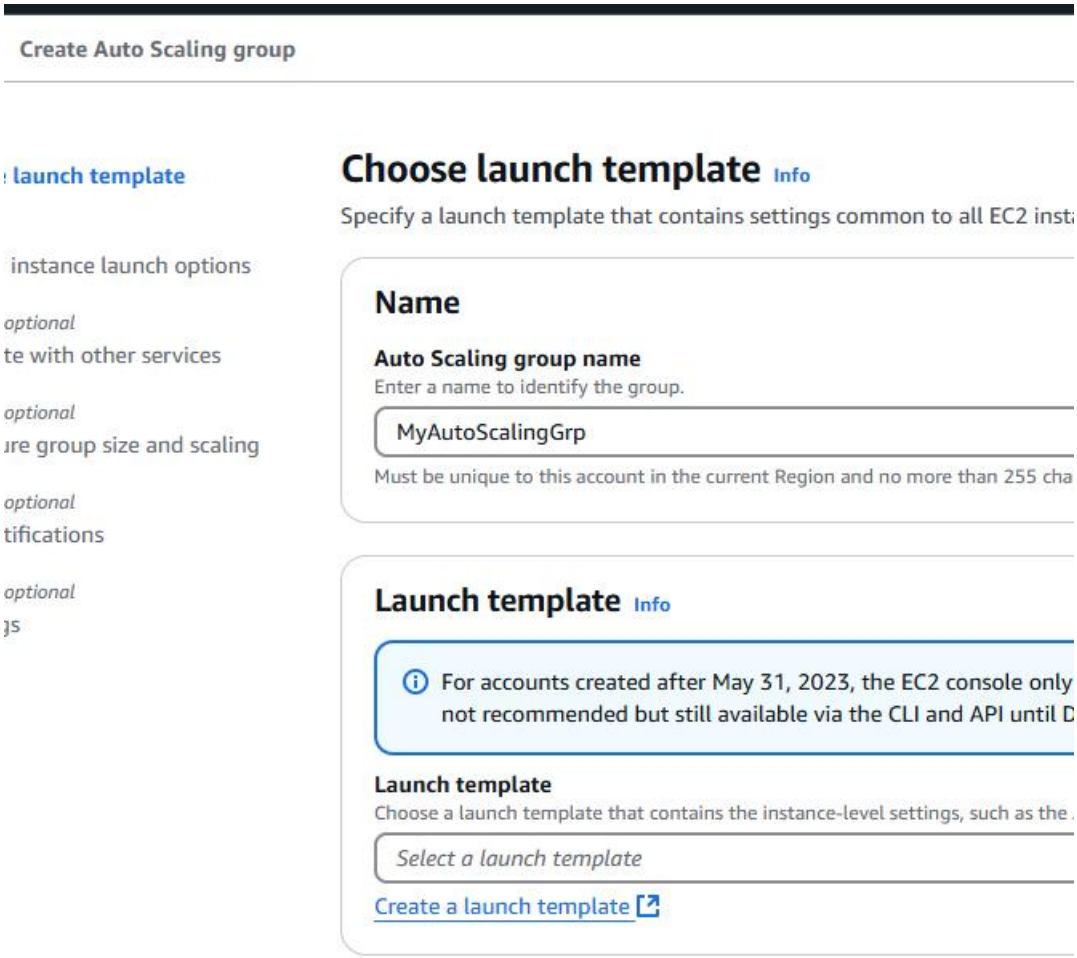


Configure Security groups, Keypair --> Create launch template

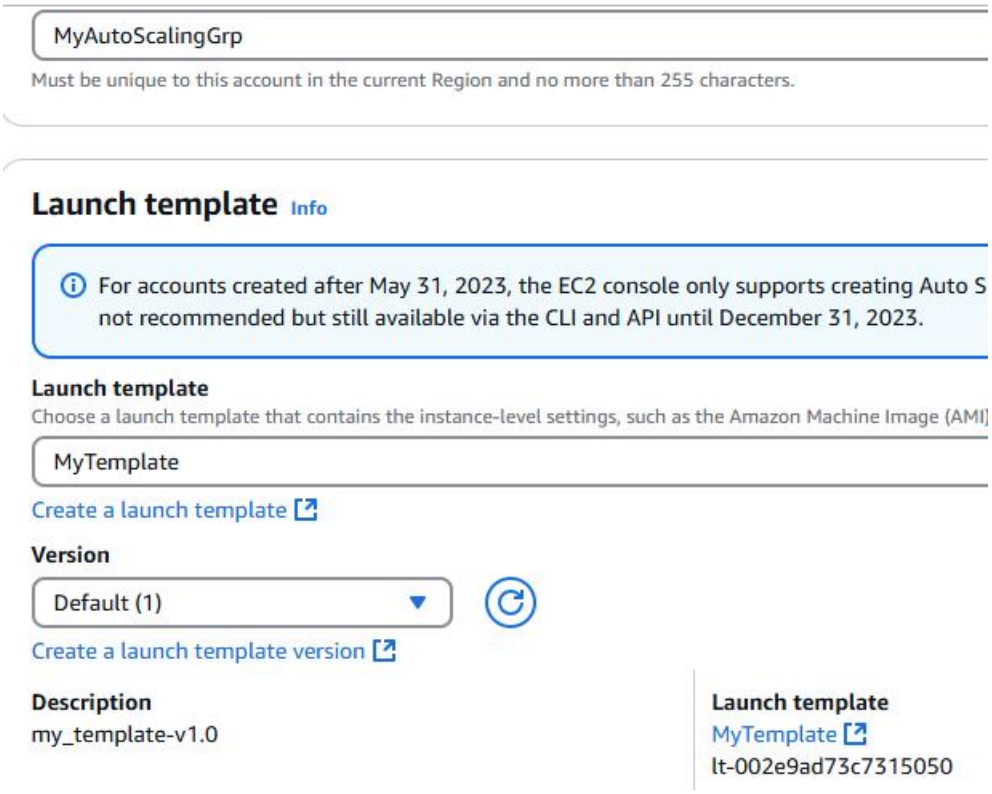




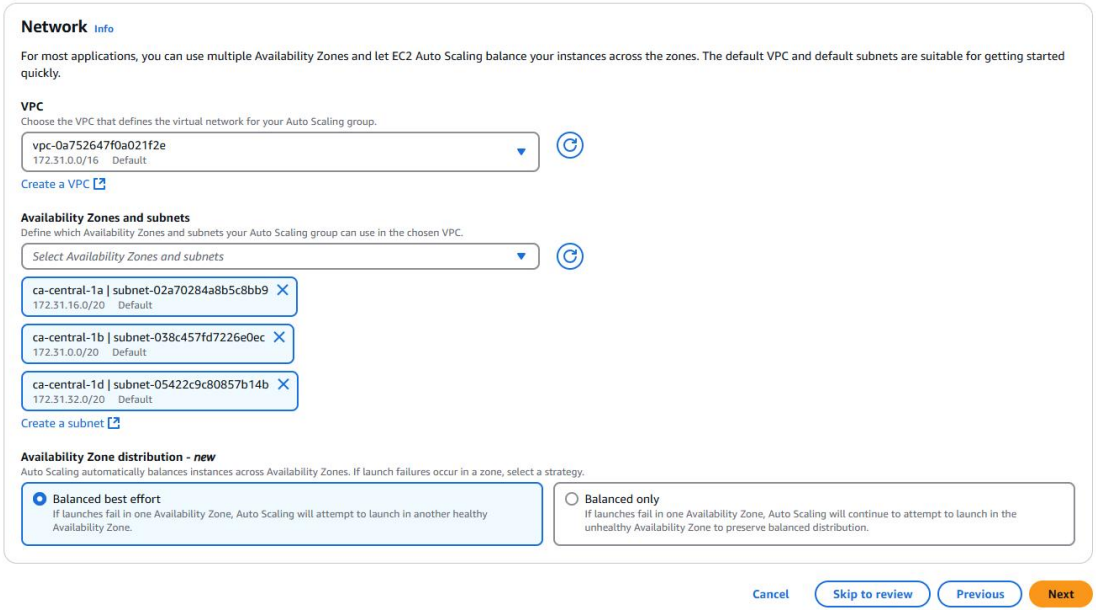
Go back to Create Auto Scaling Group



Select ‘MyTemplate’

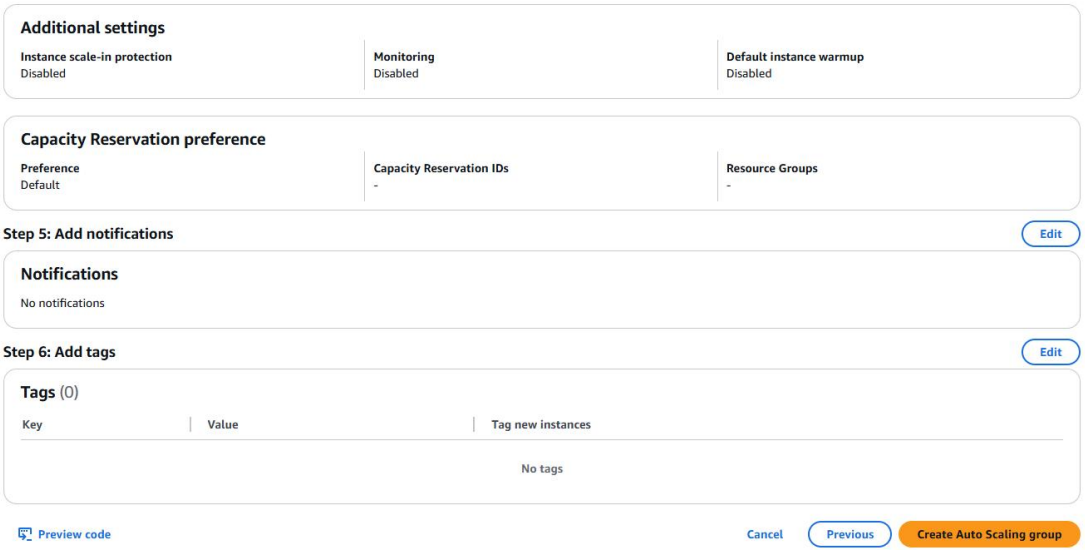


Click Next ---> Pick Availability Zones and Subnets

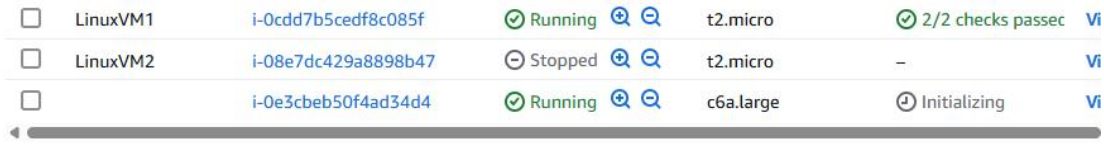


All others I want to keep it default ---> click Next

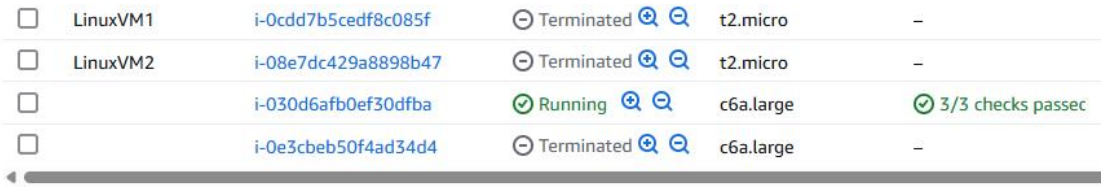
Review and create Auto Scaling Group



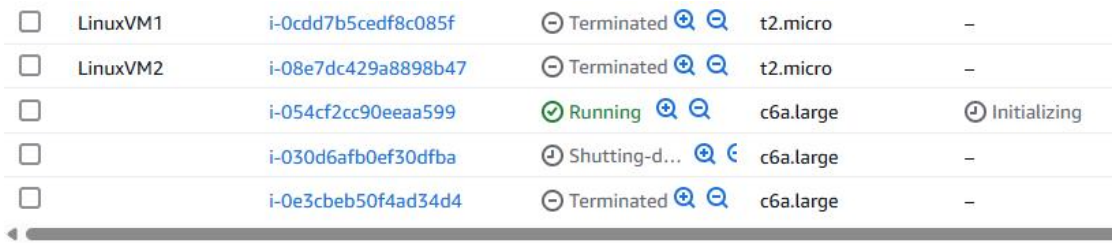
Go to EC2 Instances



We can see one more instance is being created automatically --> Auto scaling is creating this instance
We manually create LinuxVM1 and LinuxVM2, third is automatic
Now I go back to EC2 instances and I see that one more instance is automatically created. This is fault-tolerance of Auto scaling



I terminate one more instance, which initializes another instance

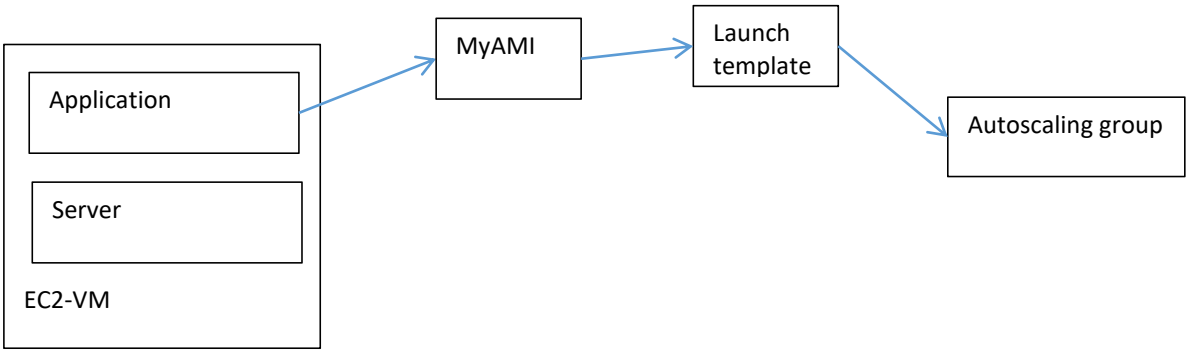


AutoScaling groups:
It is used to adjust the capacity required to handle the load
If number of requests are increasing, then servers must also be increased to give smooth experience for customers and similarly if requests are decreasing then number of servers should be decreased to manage cost. In that case, we can go with AutoScaling group.
1. Fault-tolerance
2. Cost management
3. High availability

---> To create AutoScaling group, we use Launch Template - it is used to specify configuration required to launch new VM whenever needed

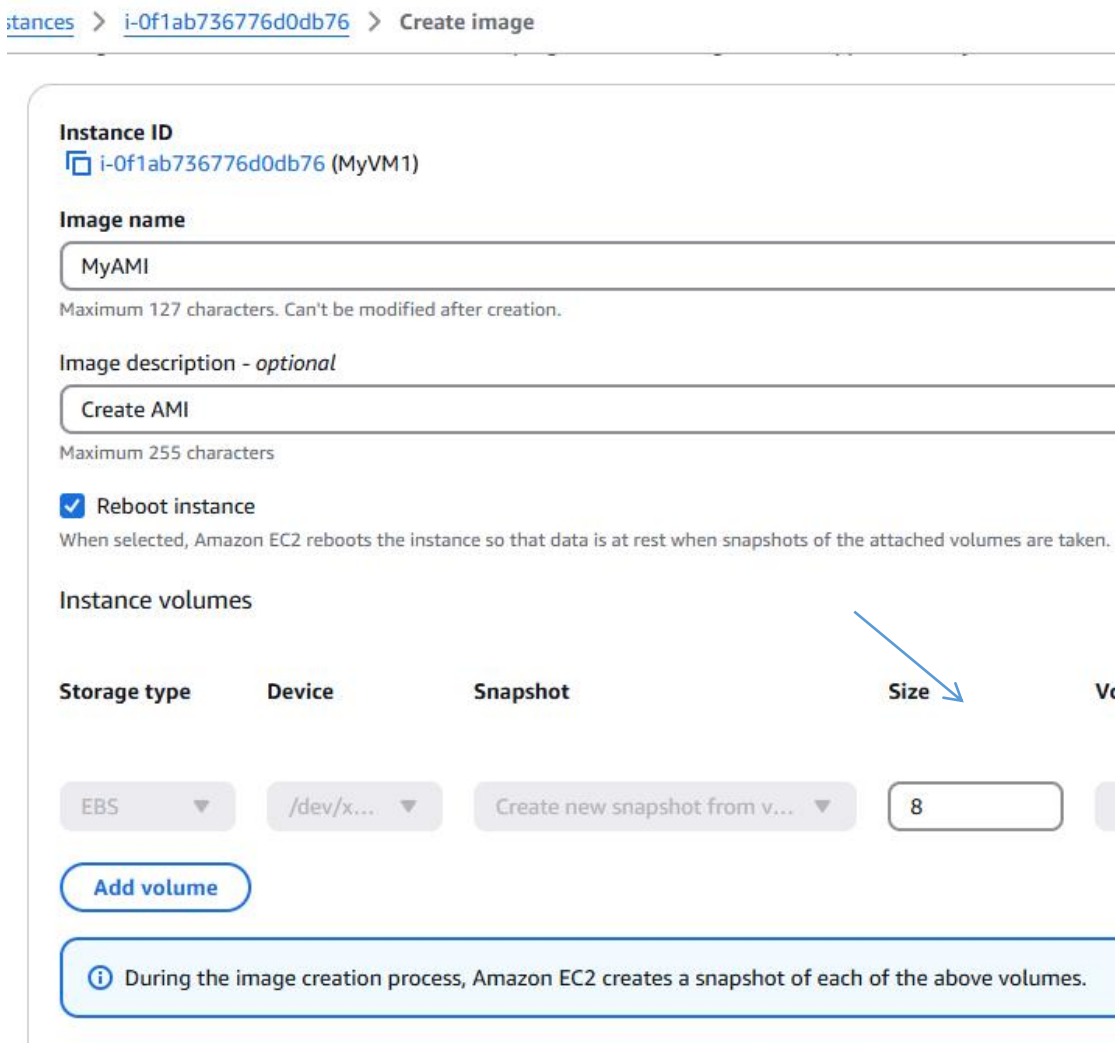
==========

How an application will be deployed into a new VM created by AutoScaling group?
Kubernetes is the default option. However, lets look into custom AMI for now
Using custom AMI ---> what's AMI? Amazon Machine Image
User data script
Kubernetes cluster



Different types of AWS EC2 instances:
General purpose
Compute optimized
Storage optimized
Memory optimized
Accelerated computing
High performance computing

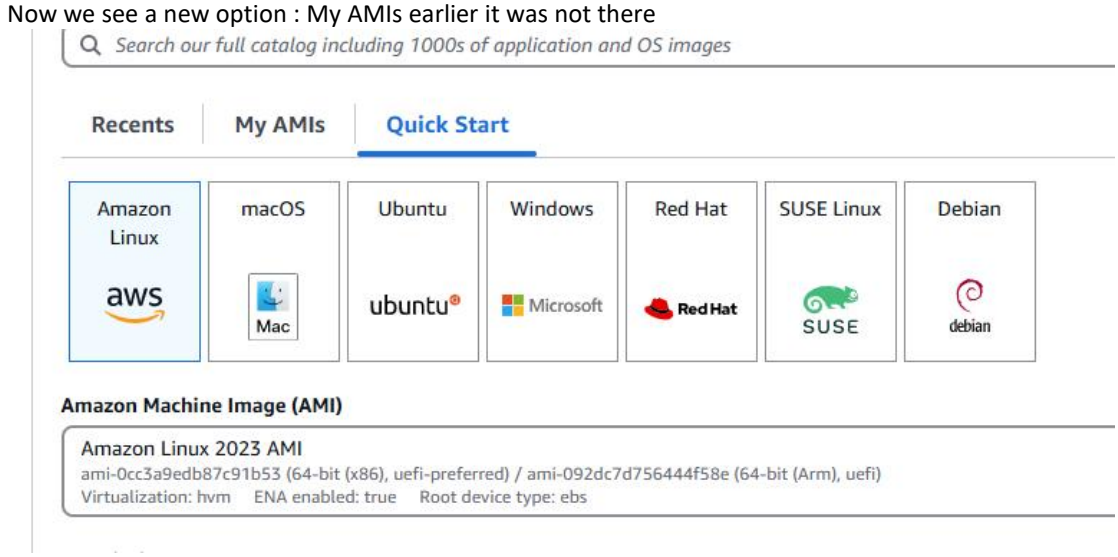
Select the instance ---> Actions --> Image and templates --> Create image



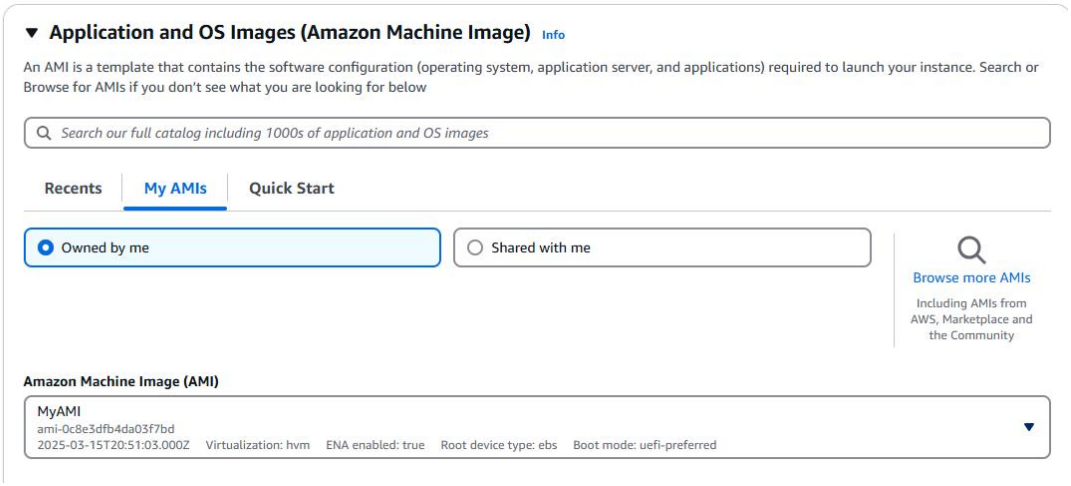
Click Create Image

Now click on Launch Instances

Now we see a new option : My AMIs earlier it was not there

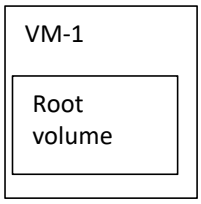


I can select MyAMI



Launch template looks like a new Launch instance only so it can go into AutoScaling groups if we want to deploy our application (from AMI) on all the automatically created EC2 instances

For every VM we create, by default root volume is there. We can add additional volumes.



I create a new VM2 with new Root volume. Yes, we can attach new volume to VMs. I can copy data from RV of VM2 into additional volume. Later, I will move Additional volume from VM2 into VM1. I think if VM1 and VM2 use different .pem files (keys), we got to do key replacement using a temporary VM . When I copy data from VM2 RV into Add volume on VM2, it will be according to private key of VM2. Whenever we try to access volumes, it will match public and private keys. In VM1, when we attach a volume it will compare with the private keys. Say if VM2 pem file is lost, create a new temporary VM ---> detach RV from VM1 attach in temp VM then copy from add vol into RV from VM1. The RV will now have IPs associated with the new pem file of temp VM. Now VM1 can use the temp VM pem file to access RV.