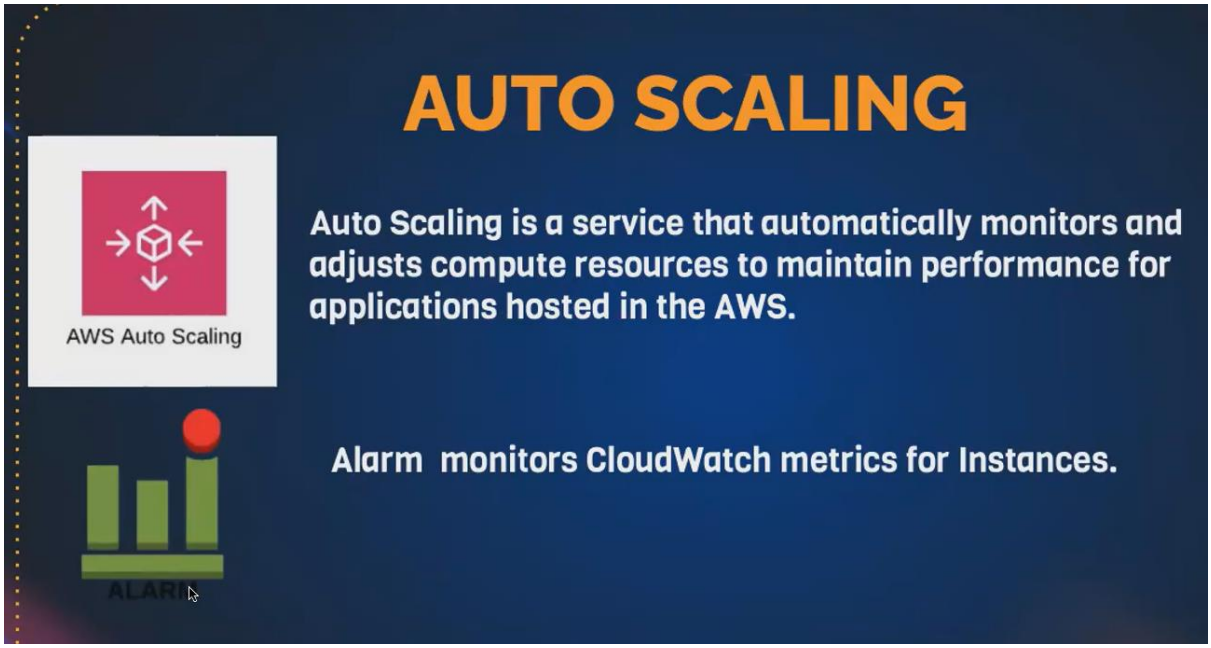
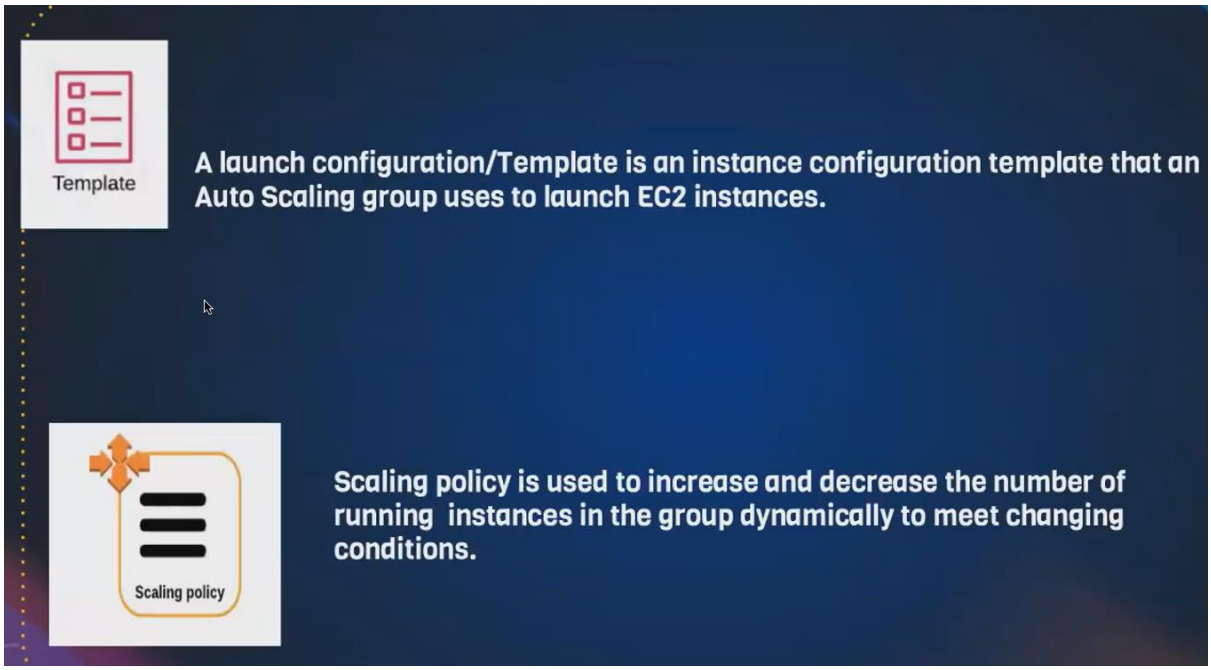
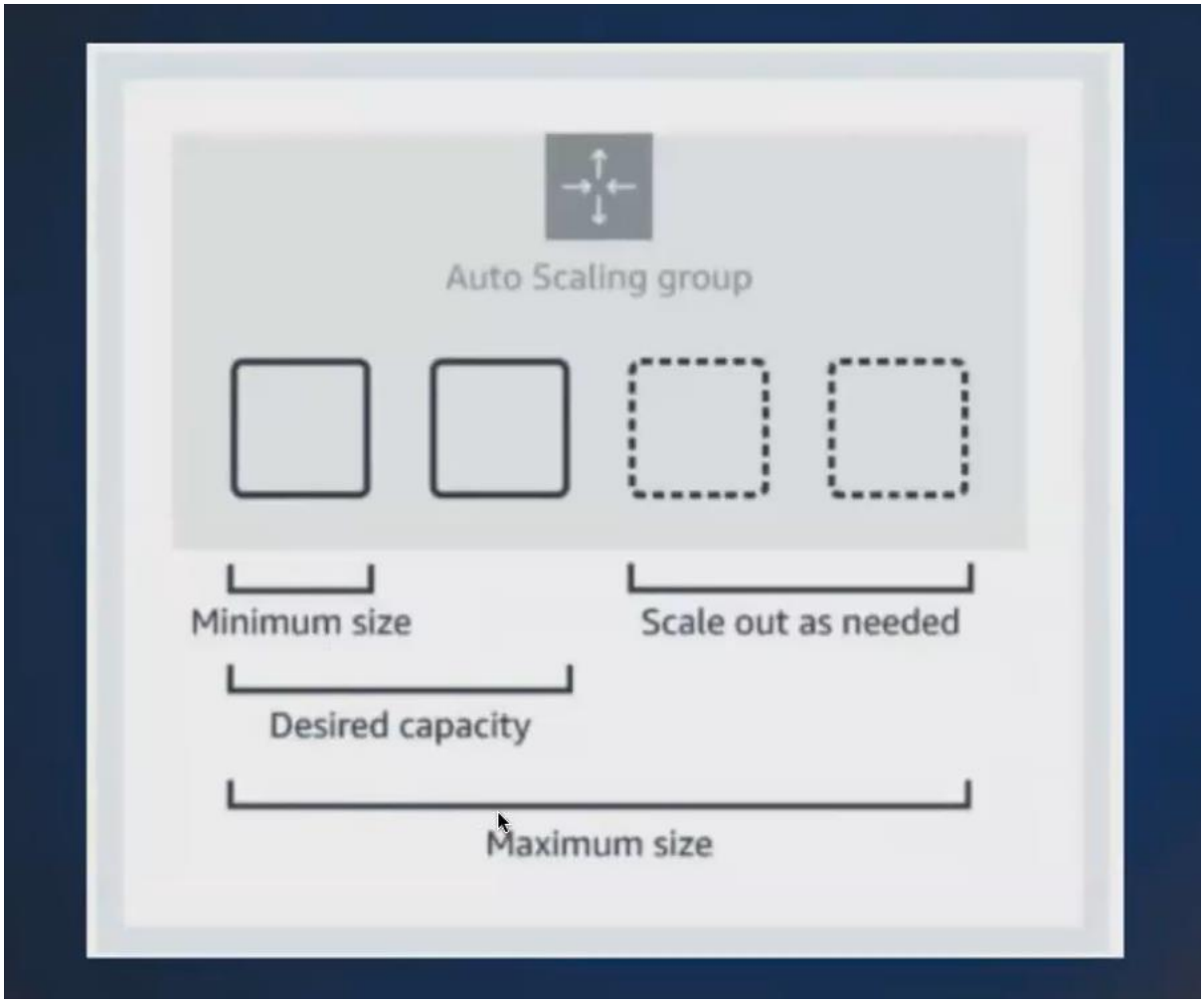
Day 5 - AWS - ELB, ASG, Cloud watch, SNS & SQS

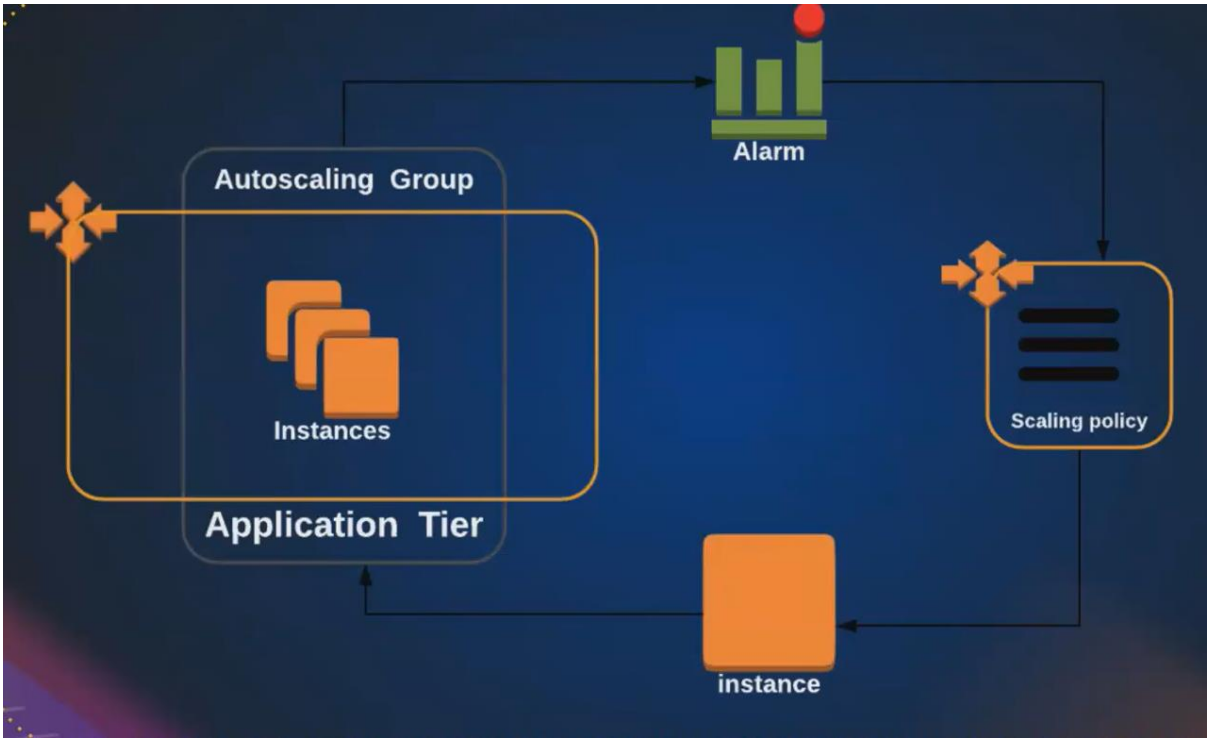
auto scaling group ,It intelligently analyses the request. Based on that, it will scale up. It will increase the Count of Ec2 or decrease the Count of Ec2 in a Aws account. mainly we go for this auto scaling group to automatically scale up and scale down so that you can, you can cut down on your cost.



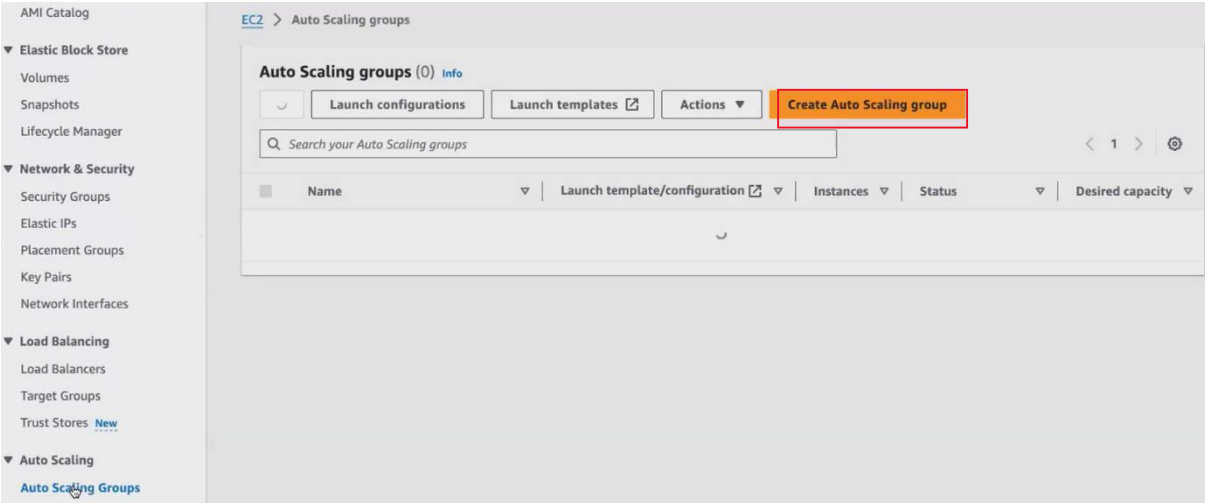




- **maximum capacity means, if I have to scale up so it will scale up till my maximum size.**
- **So if I give 4, your auto scaling group can only create up to 4 Ec2 do not go beyond that.**
- **if I have to scale down it will scale down to the desired capacity.**
- **our auto scaling group will maintain minimum of 2 ec2 always.**



This service comes under ec2 instance:



Before creating auto scaling group , create a launch template

Choose launch template

Step 2
Choose instance launch options

Step 3 - *optional*
Configure advanced options

Step 4 - *optional*
Configure group size and scaling

Step 5 - *optional*
Add notifications

Step 6 - *optional*
Add tags

Step 7
Review

Choose launch template Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.

MyASG

Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Select a launch template

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 reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

MyASGTemplate

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

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ Application and OS Images (Amazon Machine Image) - required Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

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

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible

ami-0f58b397bc5c1f2e8 (64-bit (x86)) / ami-0dda4ba9a42839a4b (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

EBS Volumes Hide details

Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp2))

AMI Volumes are not included in the template unless modified

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

▼ Resource tags Info

No resource tags are currently included in this template. Add a resource tag to include it in the launch template.

Add new tag

You can add up to 50 more tags.

▶ Advanced details Info

▼ Summary

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-05e00961530ae1b55

Virtual server type (instance type)

t2.micro

Firewall (security group)

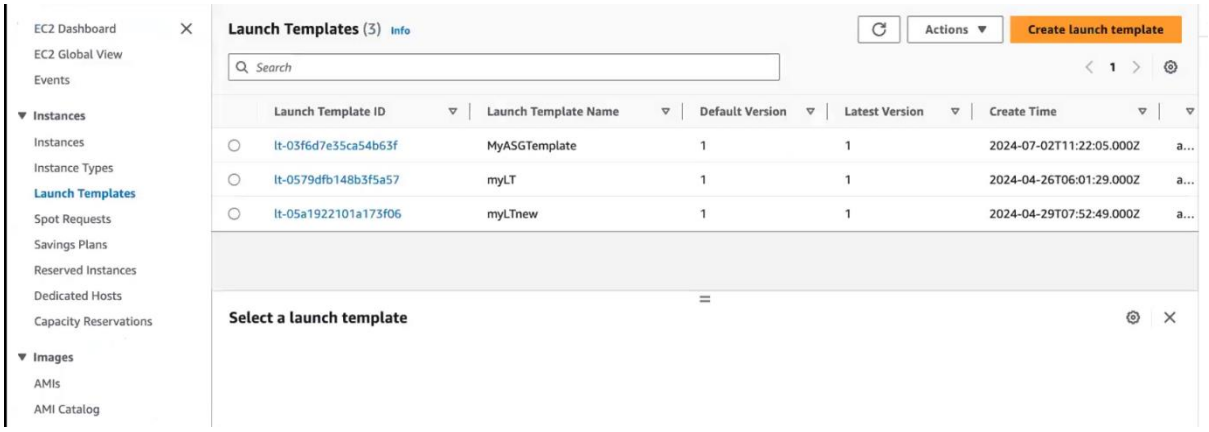
-

Storage (volumes)

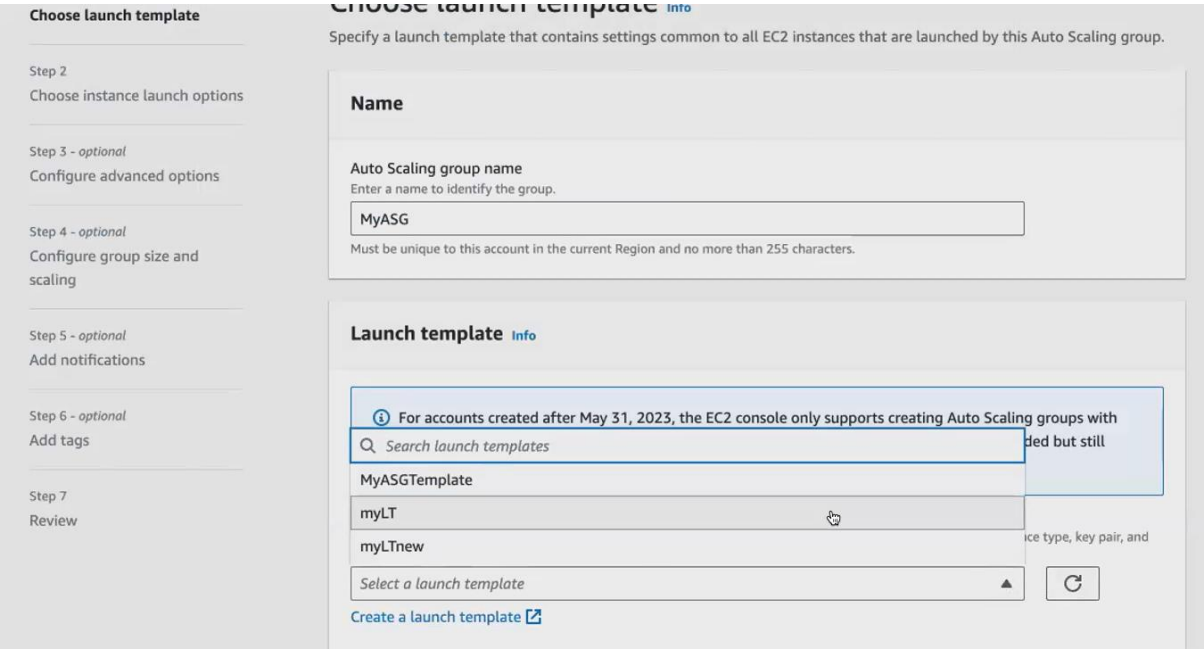
1 volume(s) - 8 GiB

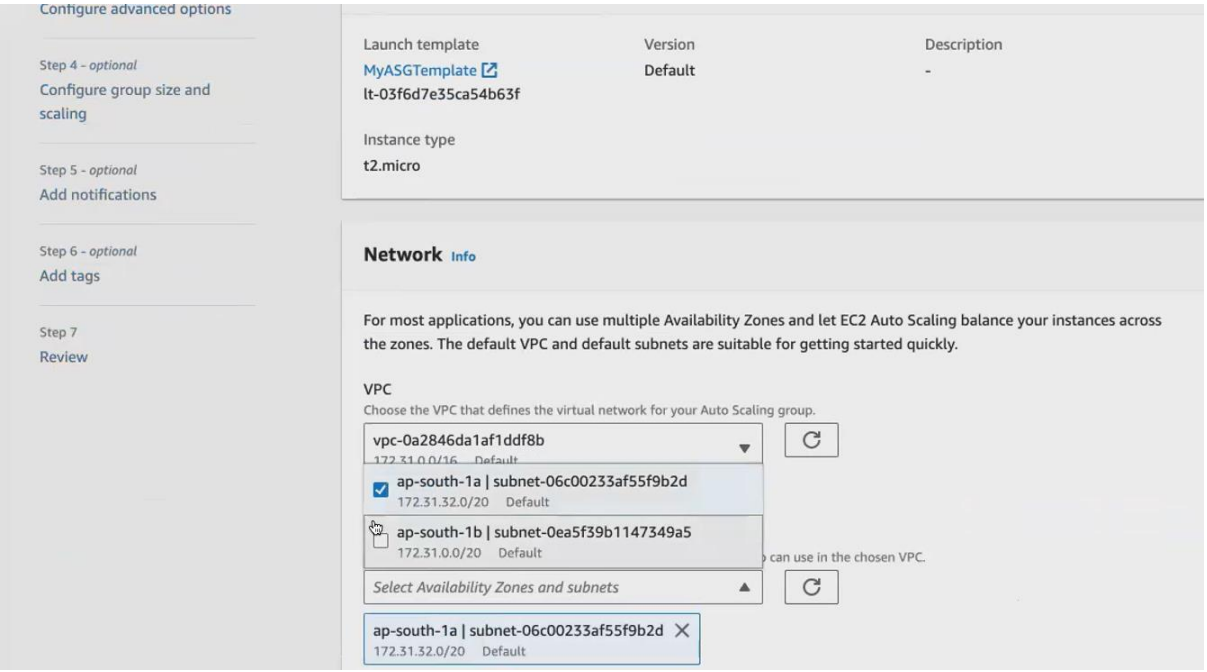
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the

Cancel | Create launch template

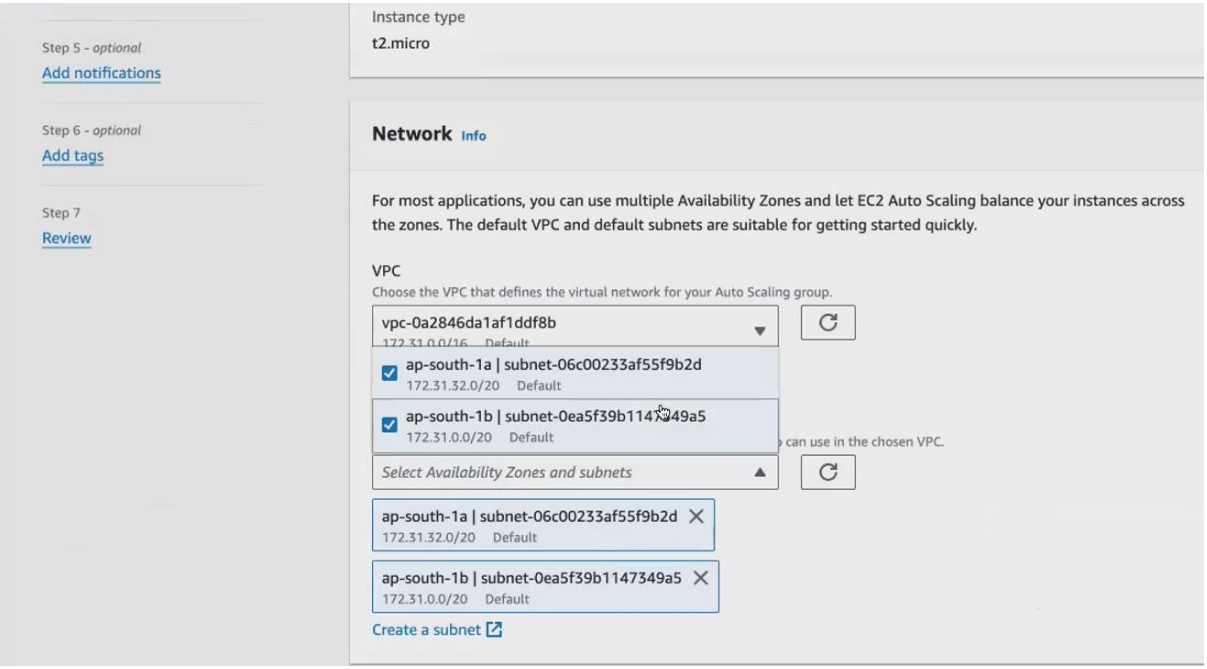


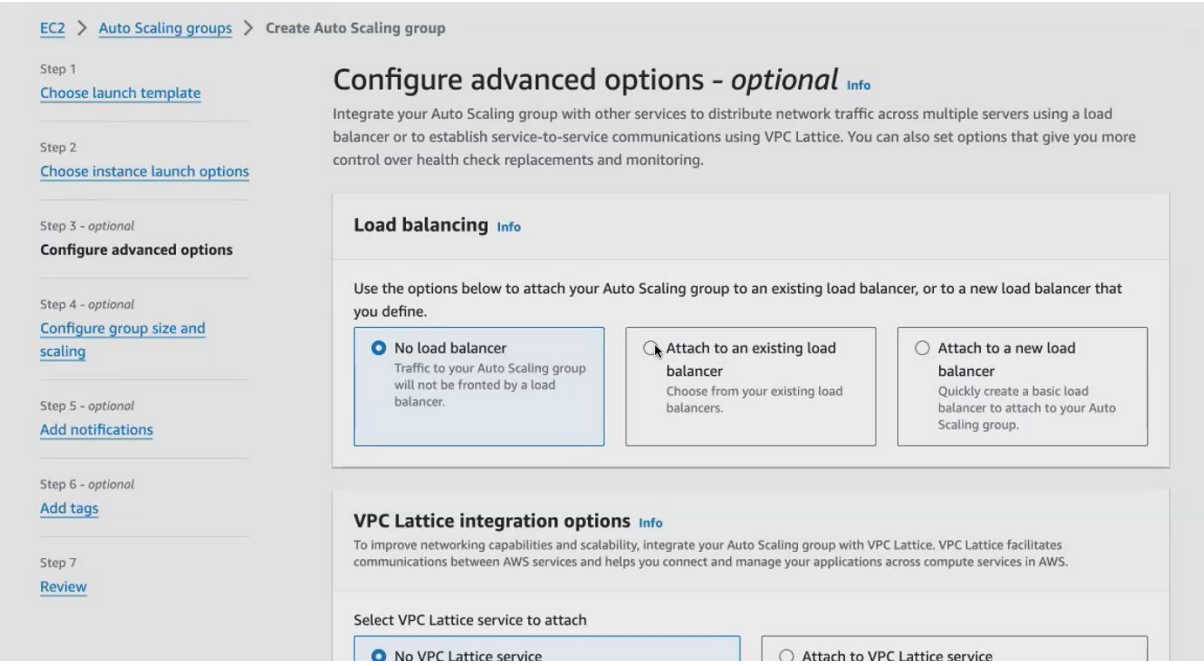
Template created. Again go to auto scaling option.here select our template name



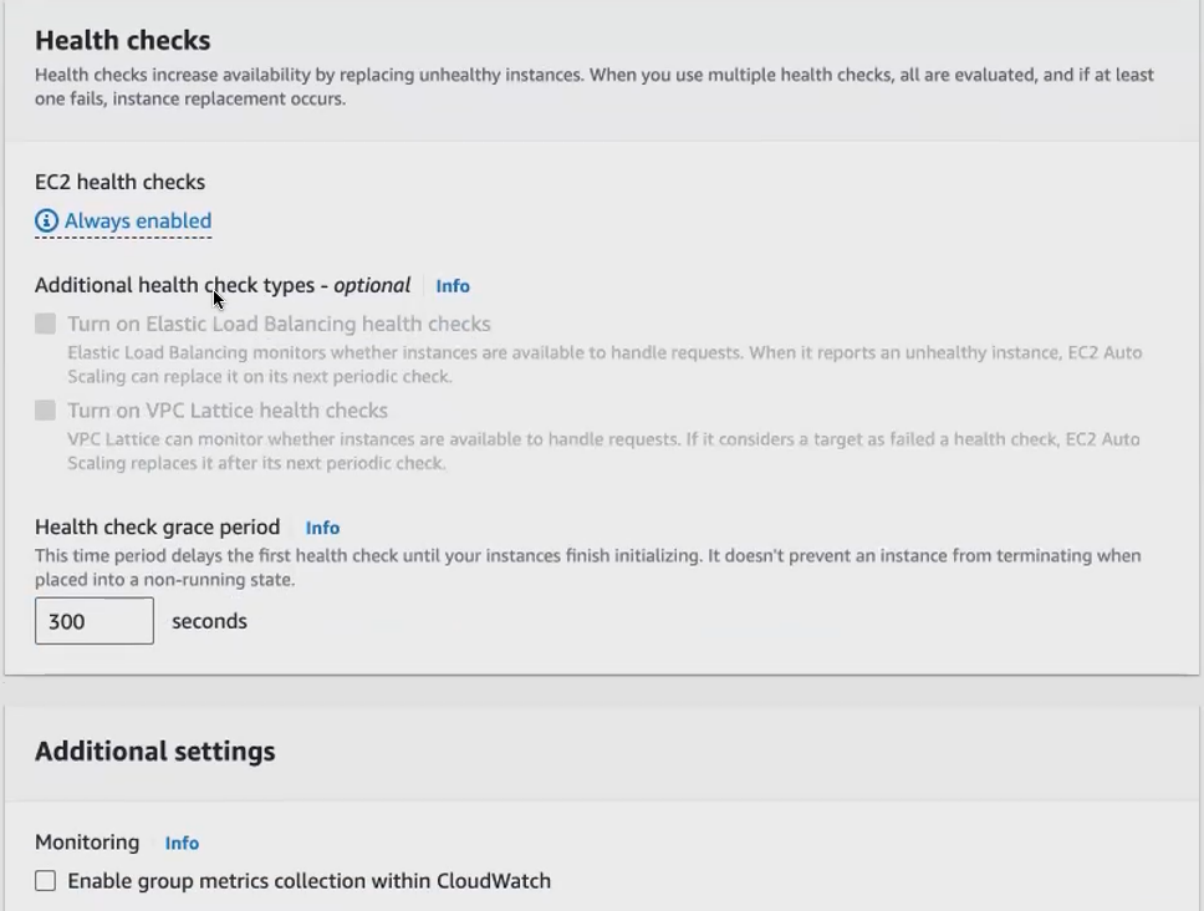


ec2 create default vpc..select 2 a.z





If we want load balance attach it here.we don't need



why health check? Suppose del instance ,it again regain it.so it is always enable

EC2 health checks

Always enabled

Additional health check types - *optional* Info

Turn on Elastic Load Balancing health checks

Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

Turn on VPC Lattice health checks

VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.

Health check grace period Info

This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

300 seconds

Additional settings

Monitoring Info

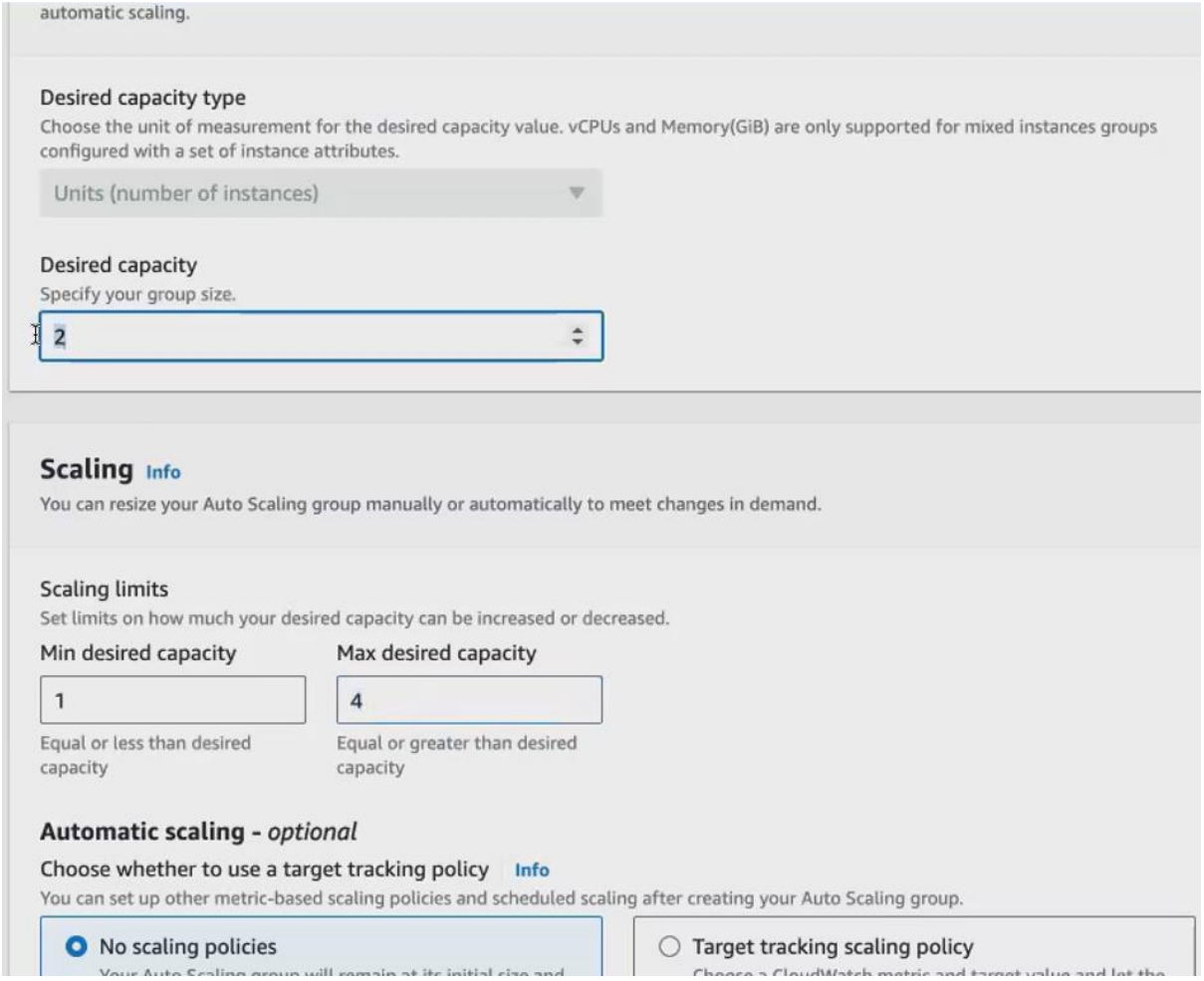
Enable group metrics collection within CloudWatch

Default instance warmup Info

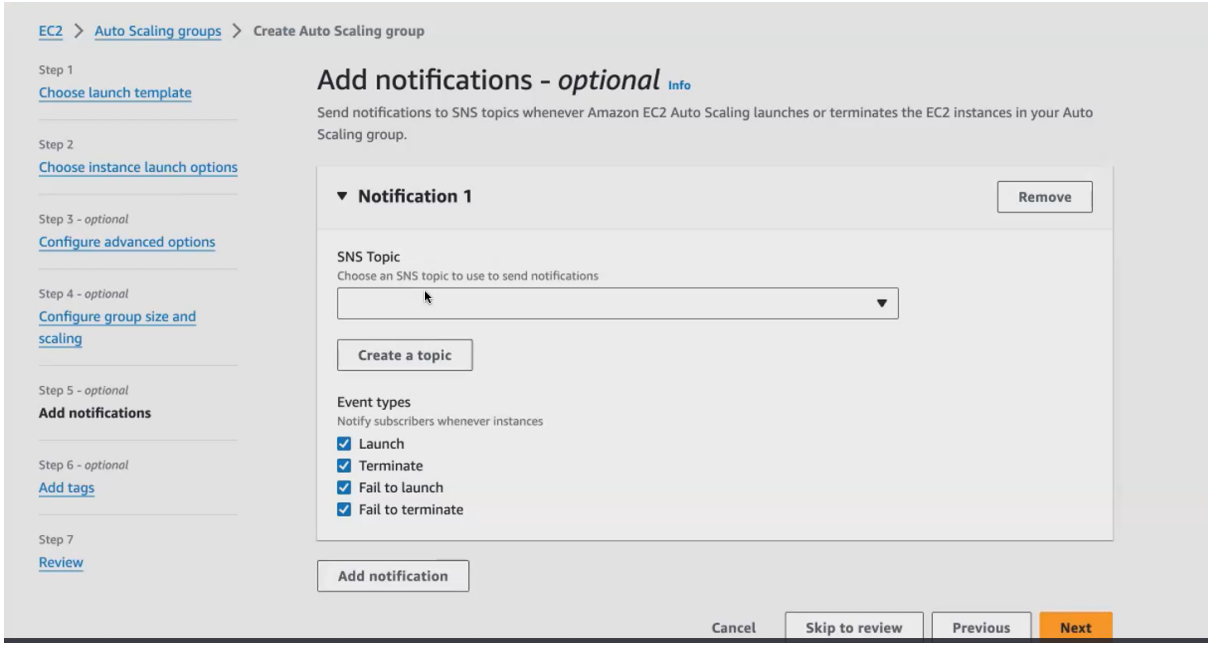
The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.

Enable default instance warmup

Cancel Skip to review Previous Next



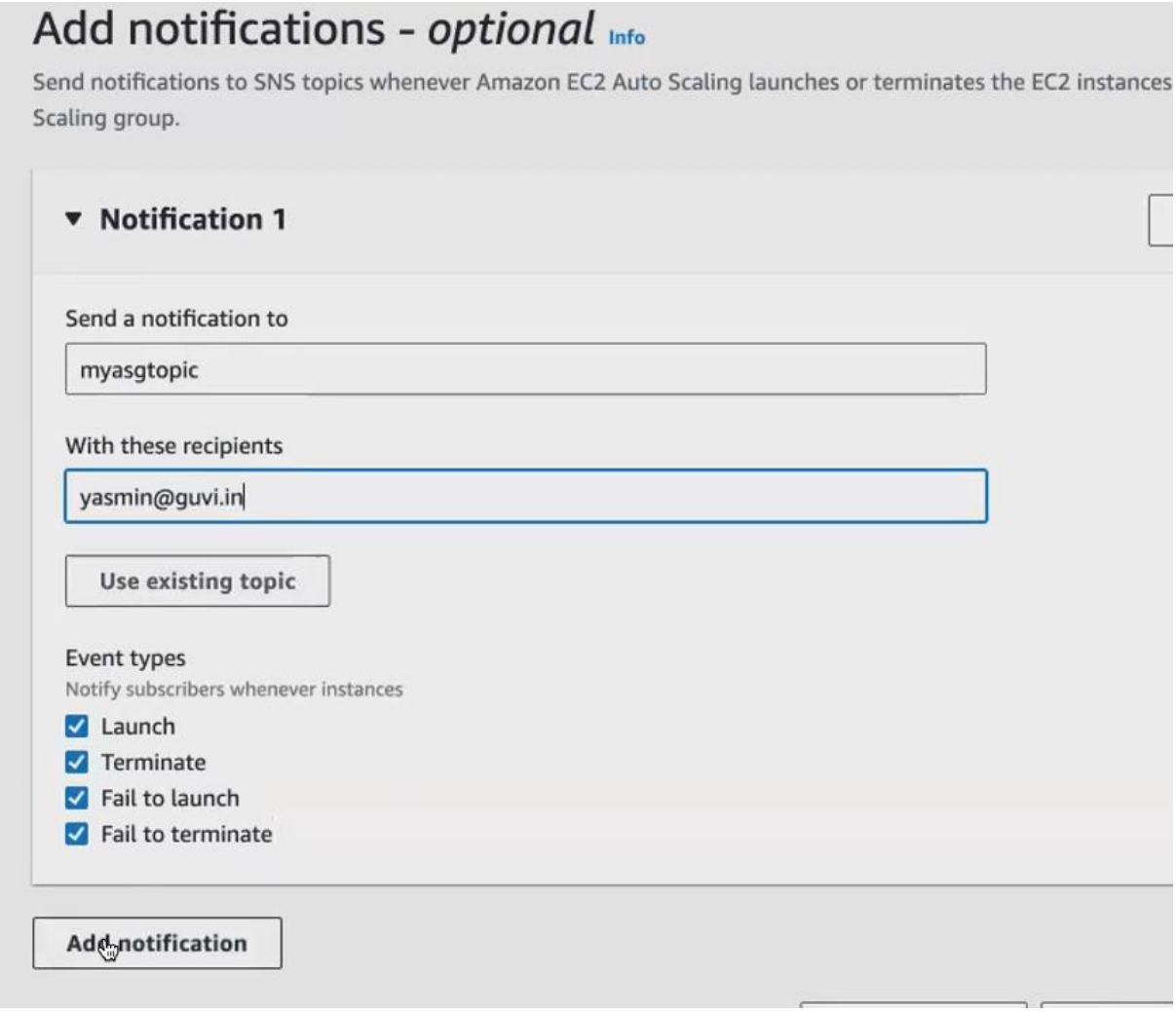
Always D.C IS 2 ..MIN AND max our choose



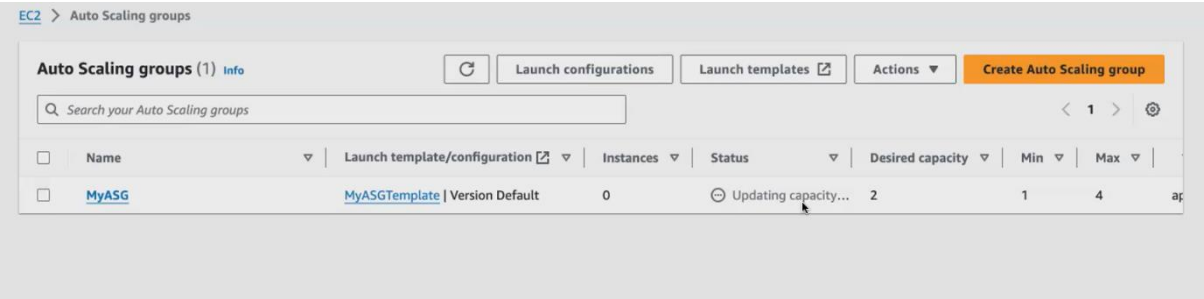
if you want, you can also add notification. So this will. This will create a notification. Notification means if any. If as I told you, your auto scaling group will be connected with your cloud watch, monitoring tool cloud watch is a monitoring tool. We will be covering up. And this monitor, this auto scaling group, will send you a notification.

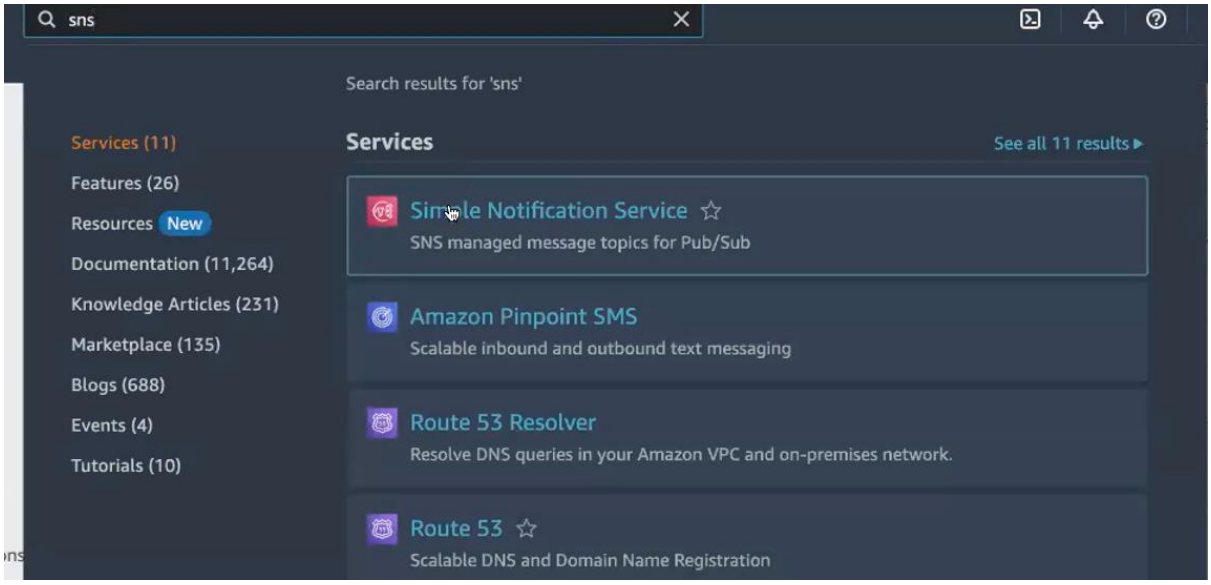
or what will happen it will, you can tell you. Can you can trigger a event. So if anyone easy to mission goes down so I can. I can tell my auto scaling group to launch a new one terminate the old one.

whenever there is a change happening in my auto scaling group. I wanted to be notified. If I wanted to be notified. I can create a topic here. Topic means so since we don't. We haven't created any Sns topic, SNS means simple notification service and aws, which helps you to send the notification via email.

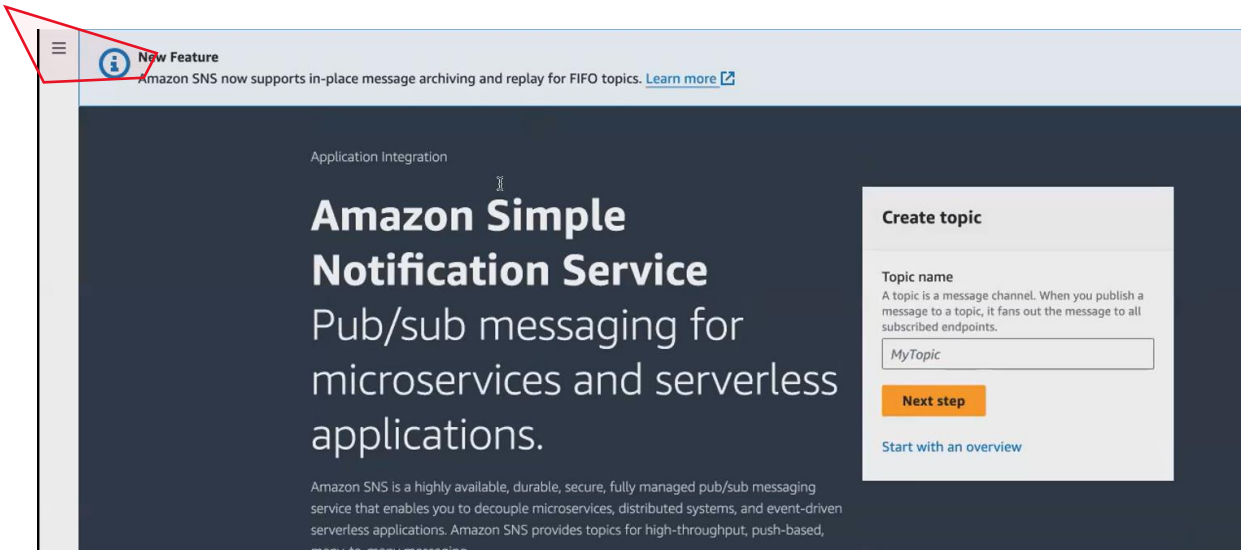


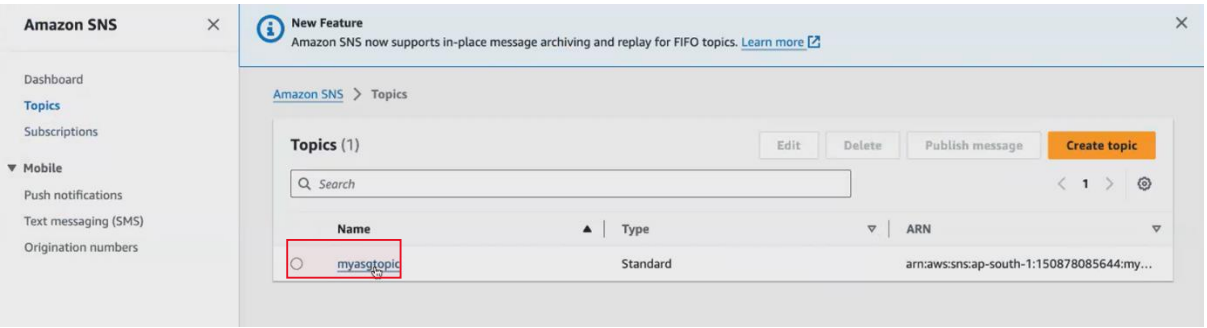
Add tags nothing has change Click create auto scale .as is created

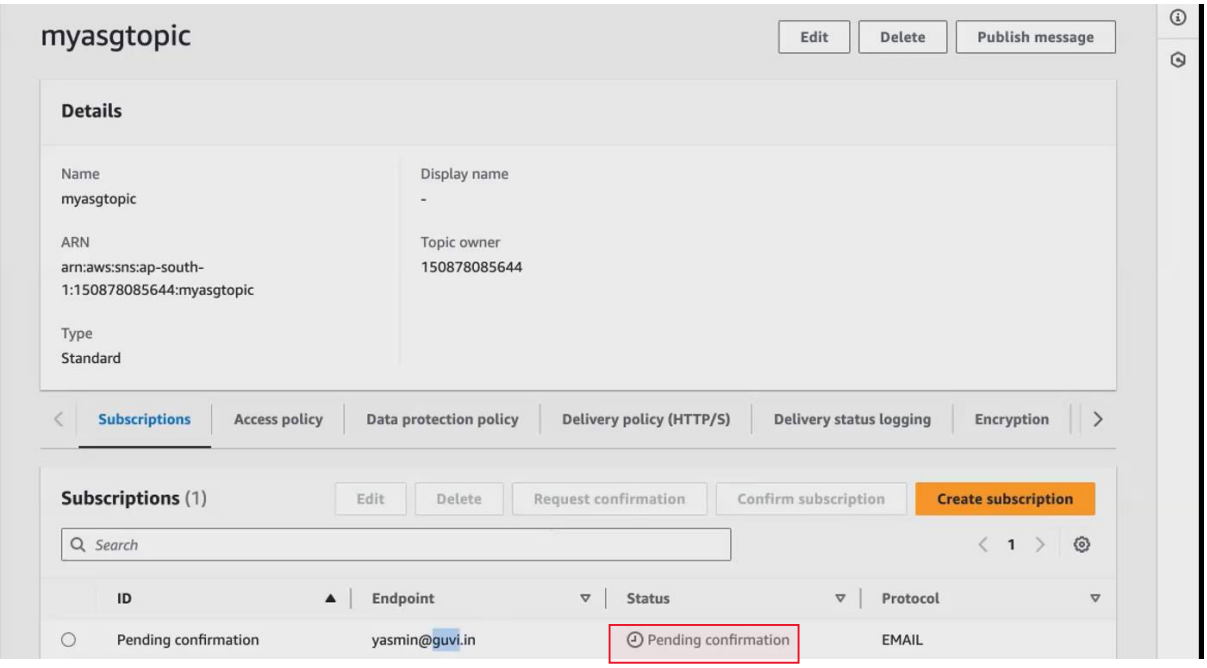




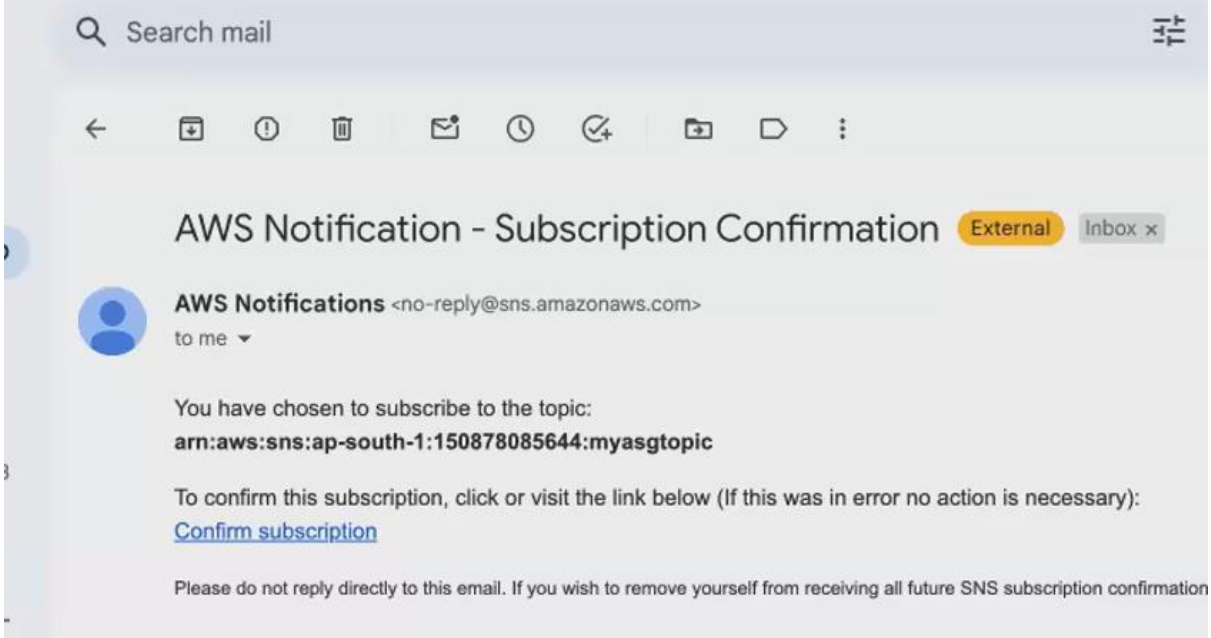
Check SNS is send notification or not

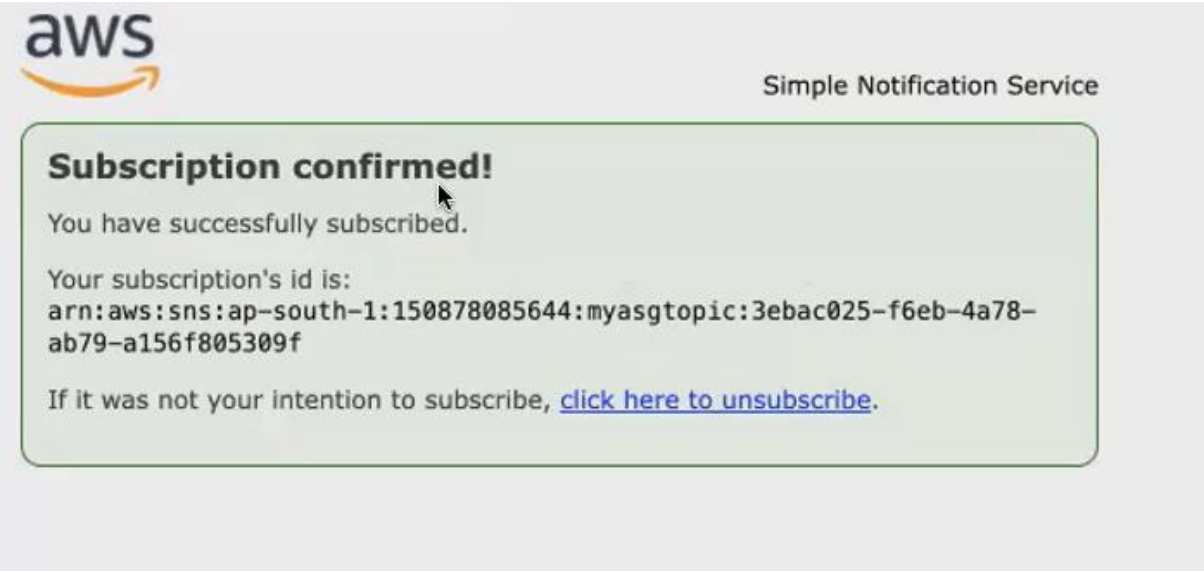


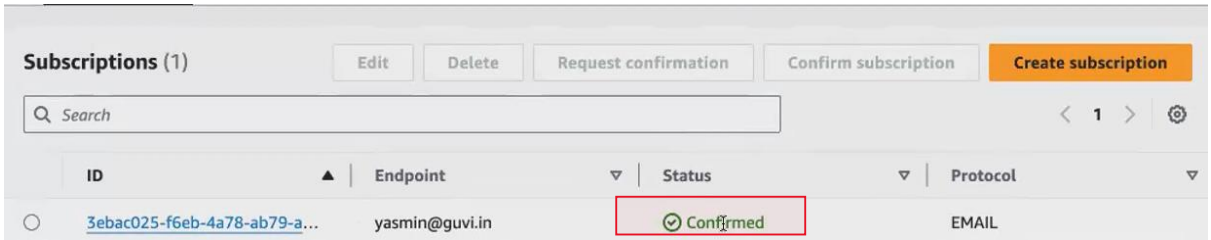


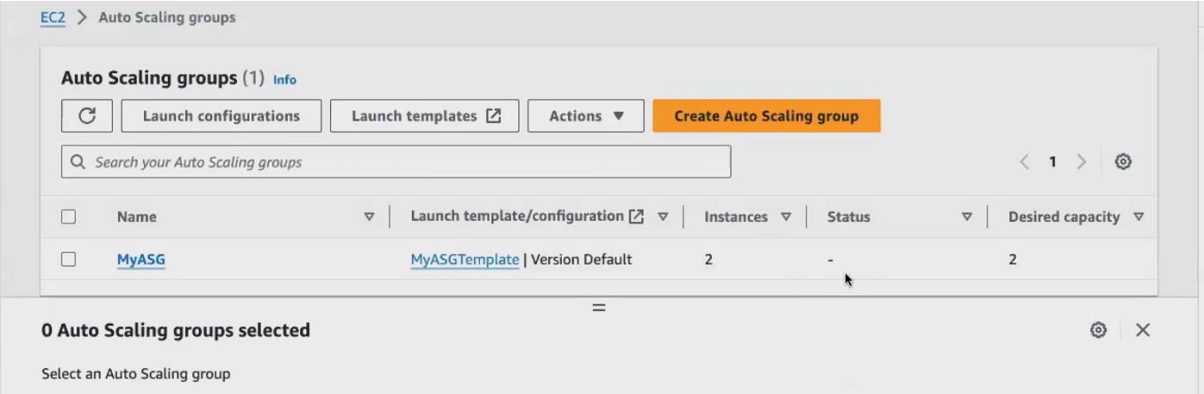


Not confirmed still is bending..goto our mail give confirmation









We create 2 instance via auto scaling. For some reason, if you delete your Ec2. automatically, your Asg will recreate another one will recreate a new Ec2.

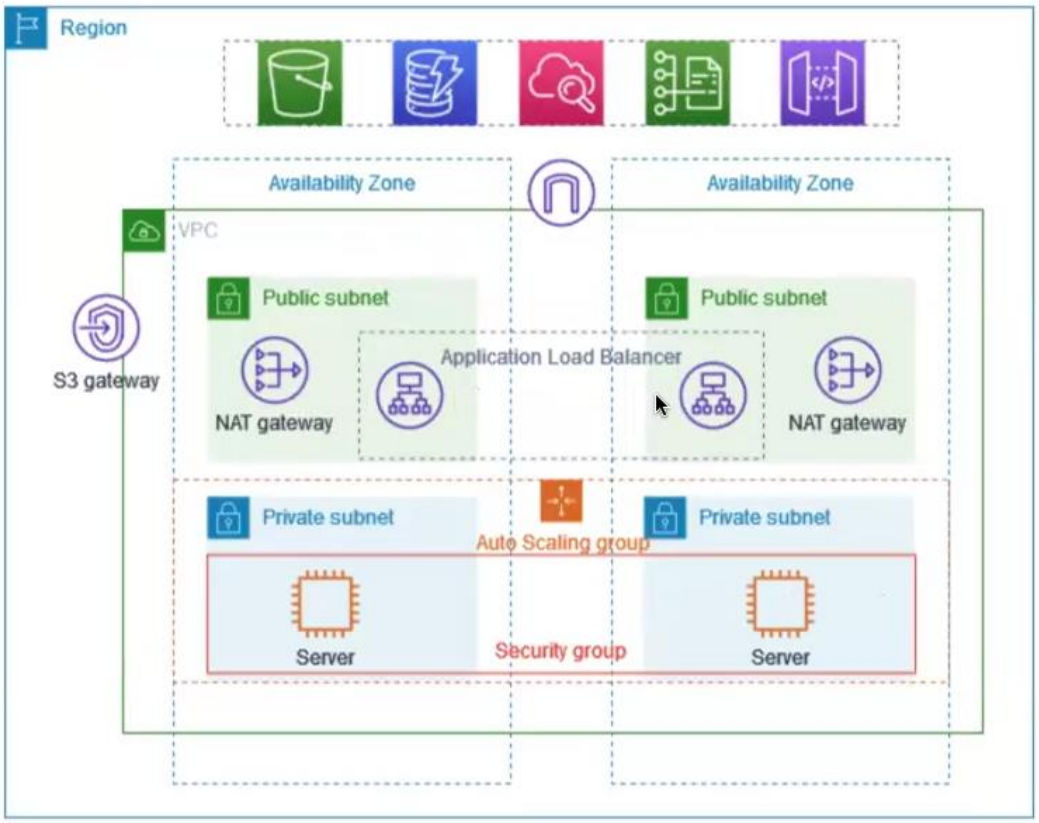
now I terminated an Ec2. So I've automatically got a notification from my auto scaling group on the termination via mail.

After termination ec2 ,auto scaling group automatically create one more ec2 .becz default is 2 instance

We will go and create the Vpc. And then we will launch all the components [auto scaling group and load balacer]

1. **vpc created**
2. **inside vpc create auto scaling for private subnet**
3. **inside vpc Create load balancer for public sunnet**

Architect of VPC



Now we are launching Vpc in a manual way. You can launch the same Vpc Via aws cli terraform. So usually we use terraform to launch this Vpc. we don't go and manually configure the Vpc. We only use terraform or cloud formation to launch the Vpc.

1. **Vpc creation :**

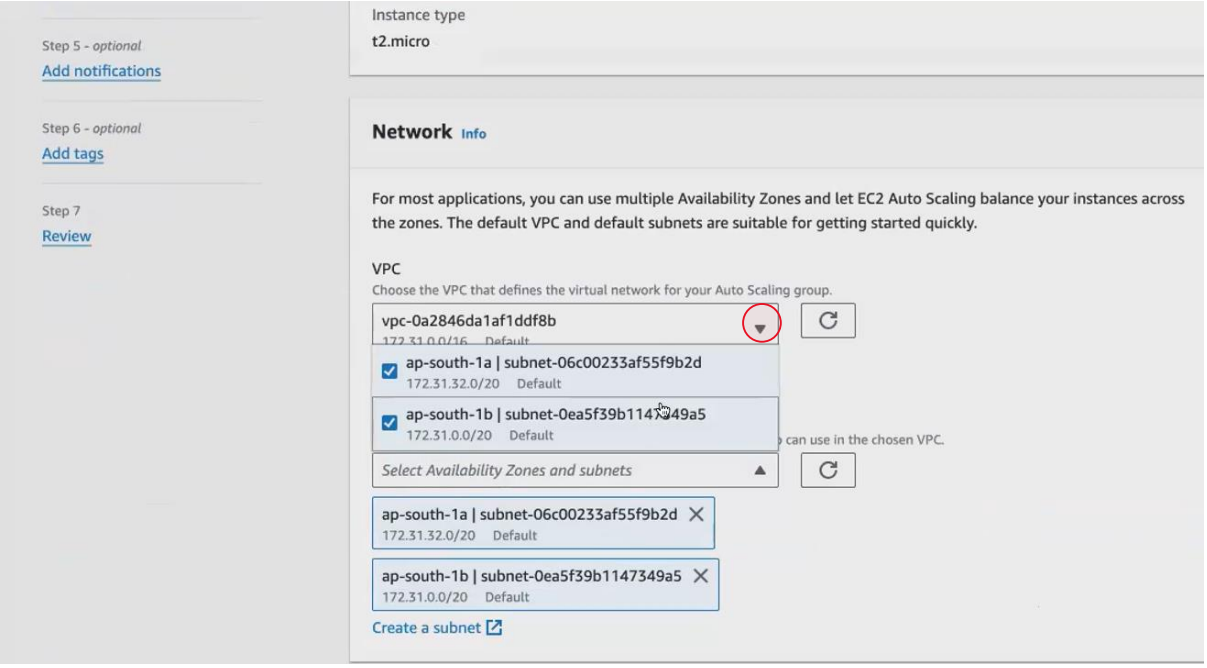
Search vpc and click it.create vpc. In day4 AWS notes refer how to create vpc

https://us06web.zoom.us/rec/play/FKm1k2Zrvv4OUj0zqcsLERlhzAh1802t6siH2kJoJSvQUi2BhTPM-ULB_M2329EOzVNRfRGs-xngsk-x.OtX6e9DkO8t4_Oy_?hasValidToken=false&canPlayFromShare=true&from=share_recording_detail&continueMode=true&componentName=rec-play&originRequestUrl=https%3A%2F%2Fus06web.zoom.us%2Frec%2Fshare%2FEOWS2ccsUK7rDNqsL4rLd0hjSlHOu-0mCePgMVGUud8iVEW1YIuqkQNH24DxxXEq.bwYrcpi6DlhiZw55

- **vpc created**
- **inside vpc create auto scaling for private subnet**
- **inside vpc Create load balancer for public sunnet**

2. **inside vpc create auto scaling**

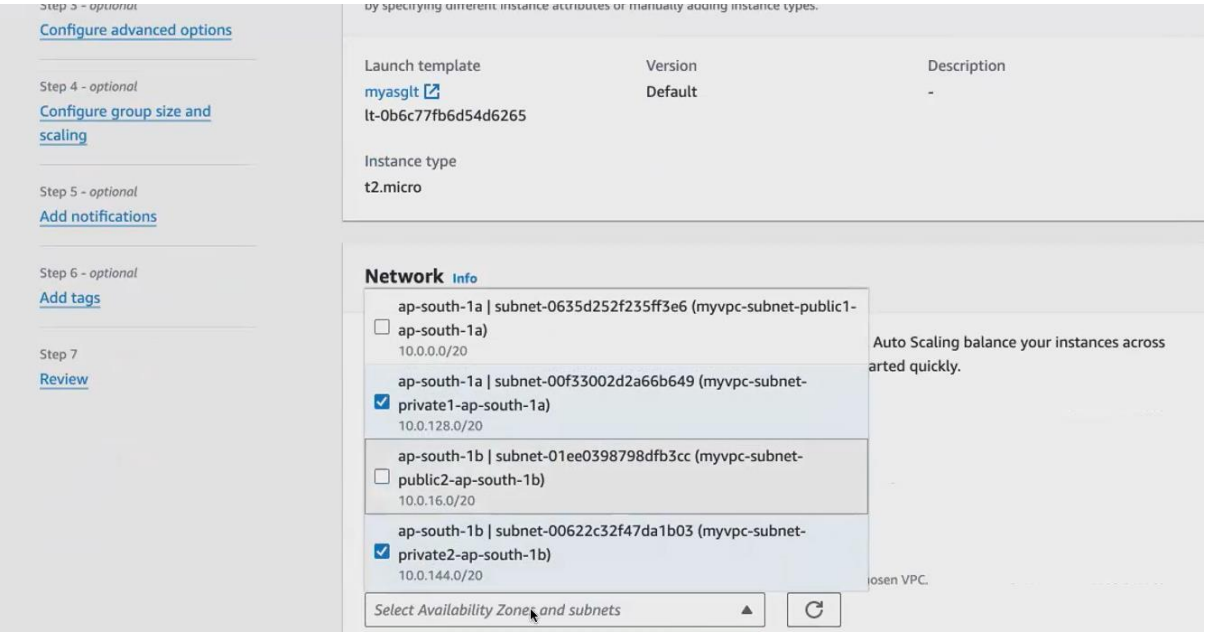
before creating auto scaling we give default vpc..but now give our 2 public subnets.

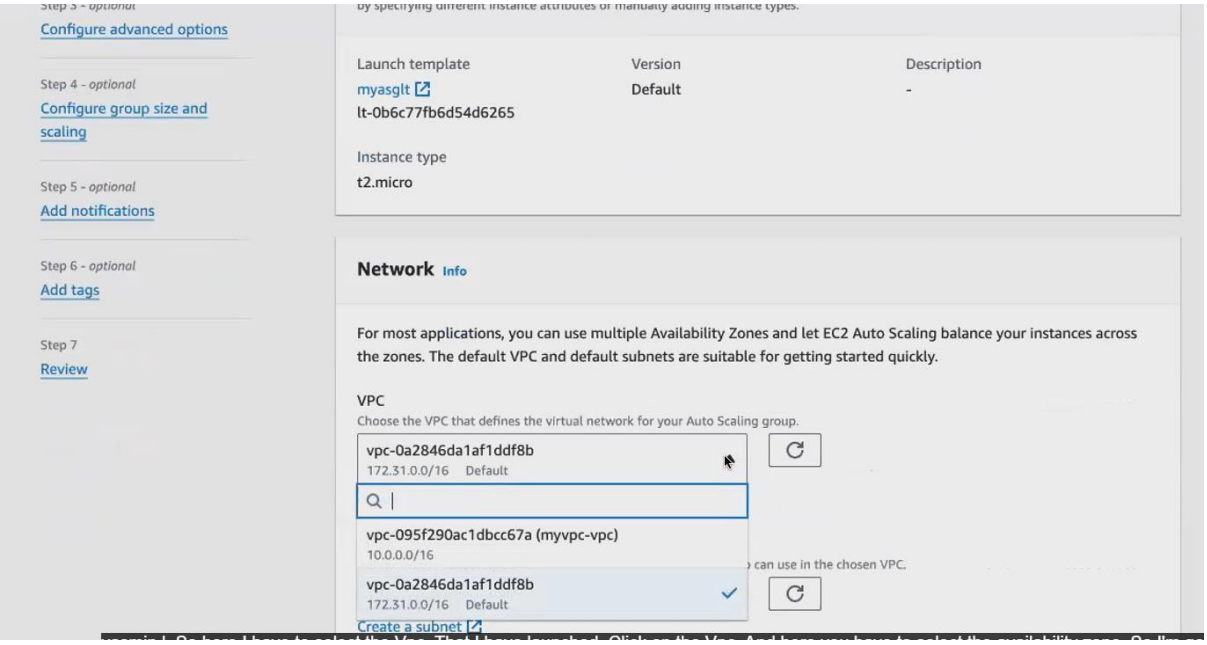


Remaining steps (refer auto scaling creation steps)

In our vpc ,now create auto scaling for private subnet

3. **<u>In our vpc Create load balancer for public sunnet</u>**

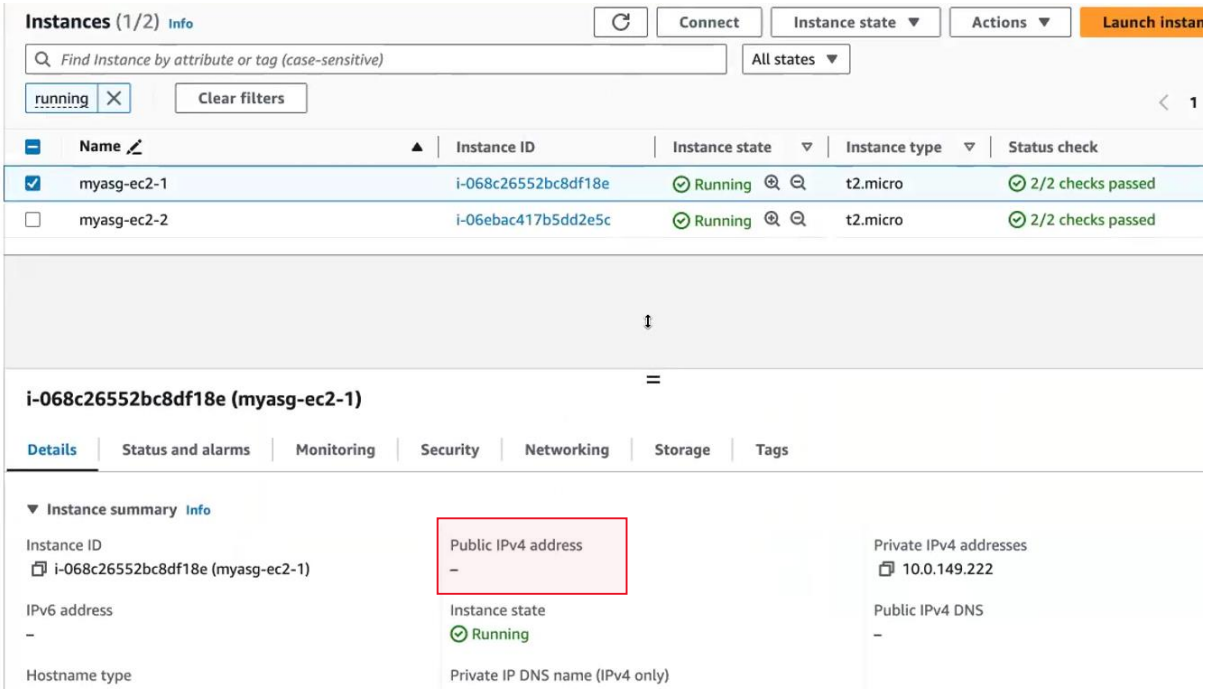




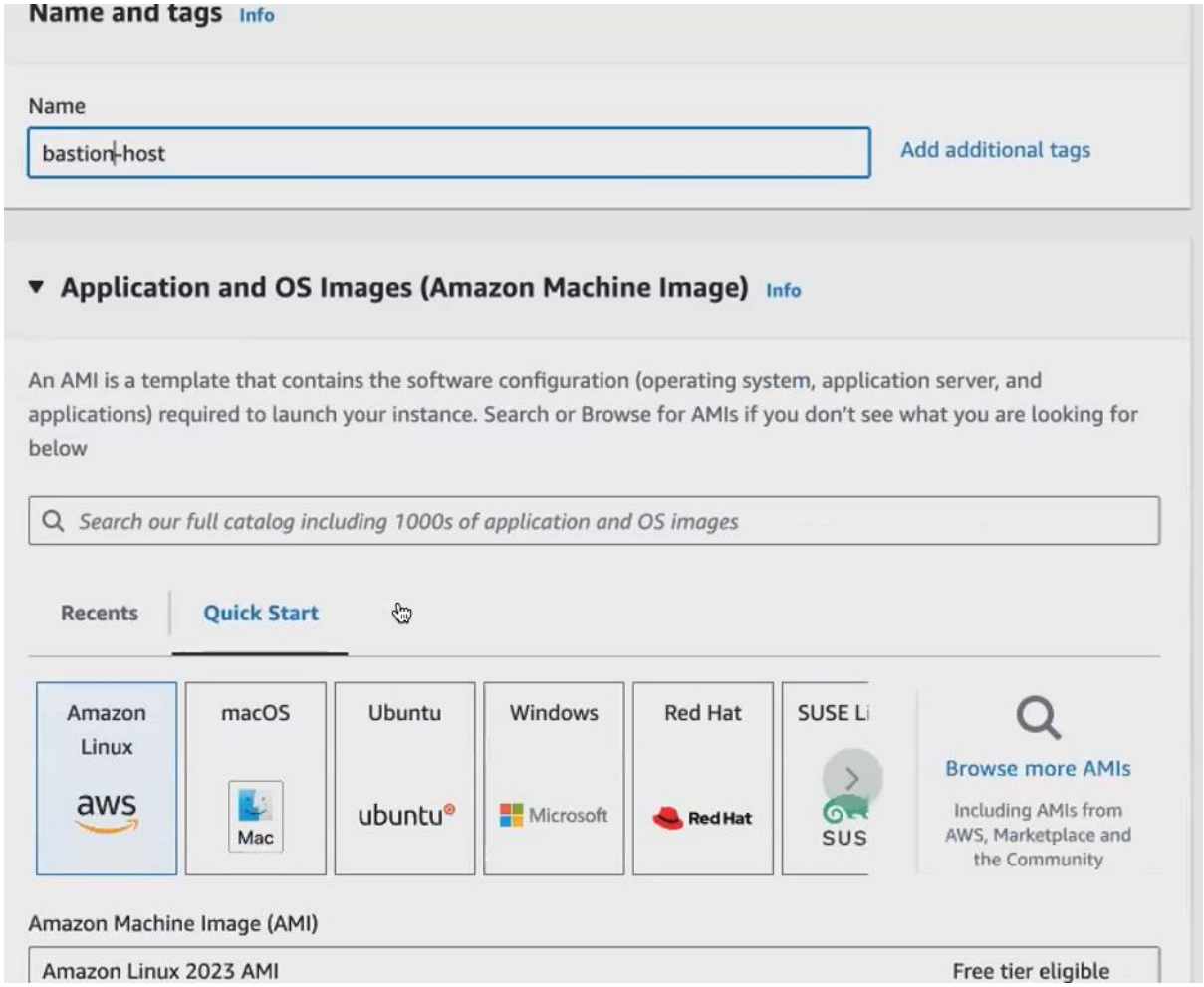
- **you cannot access the private Ec. 2 machine. You will not be able to login to the private Ec2.So what we do we will <u>create a bastion host</u>. We call it as bastion host, so which will using that bastion host. That is also an Ec 2 mission only using that bastion host, we will connect with our private Ec commission.**
- **so you have to create an Ec2 in the public subnet and via that Ec2 you'll be. You'll be launching the application inside your private subnet. Since your you cannot connect directly with your private subnet, we cannot connect**

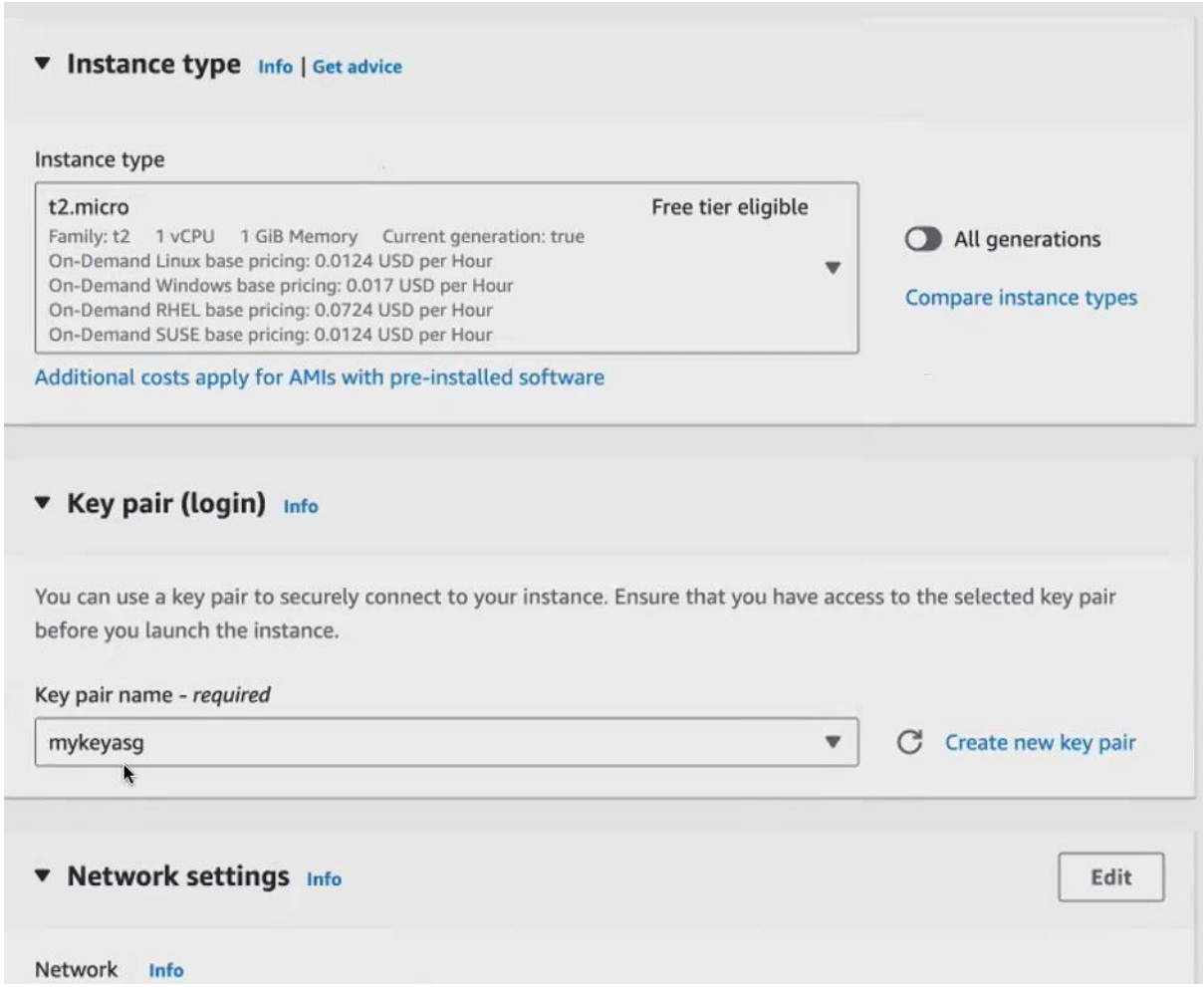
remotely to the private subnet. We use the bastion host via that bastion host. We can talk to them

- directly we can’t connect with private subnet. Via bastion host only we can connect with private subnet.
- so, after doing, after launching the application, we launch the application load balancer in the public subnet, and we will see how to access the application.
- You don't have a public IP here because this is a private Ec2. You don't have a public IP here [we create a private vpc]. You will not be able to connect with this. Ec2. So in order to install or run any application inside this private Ec2, you cannot do it directly. Rather We will be using another Ec2 which will act as a bastion host. So via this Ec2, we will login to our private ec2. Okay.so, we call this EC2 as bastion host.
- In below public id not shown. So we create another ec2 for bastion host.

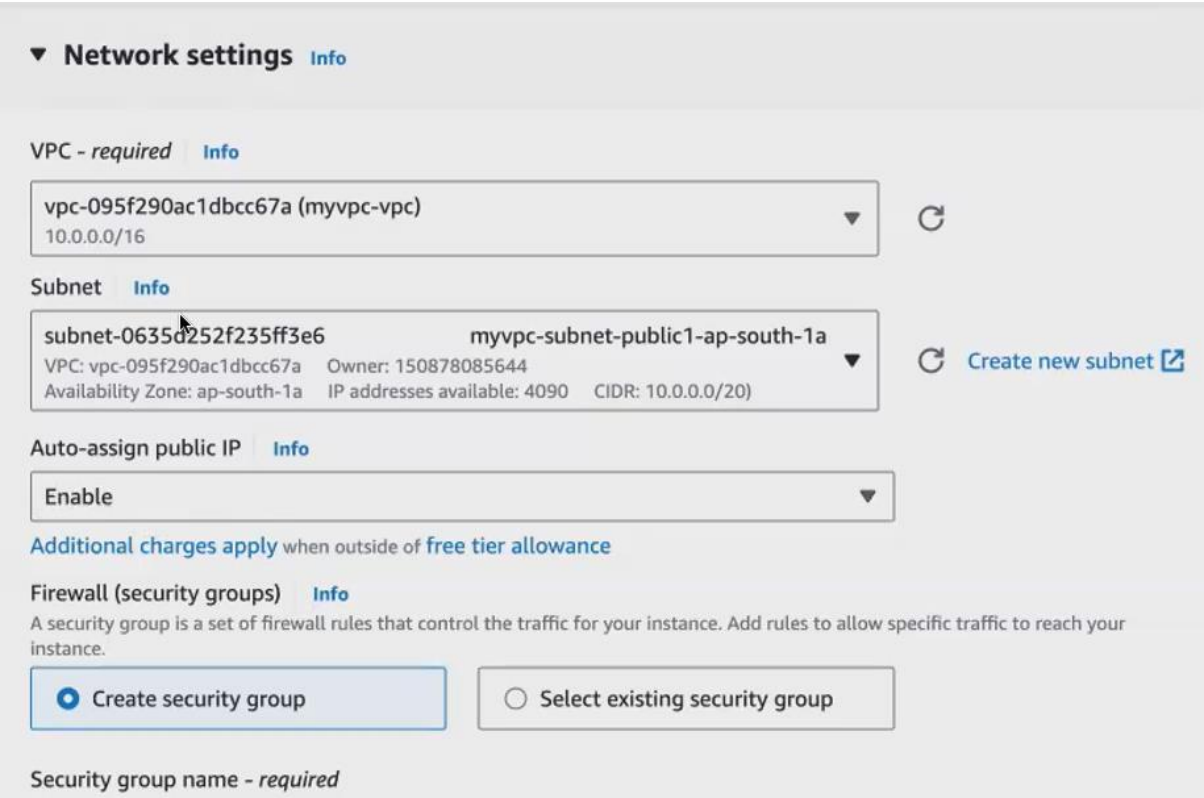


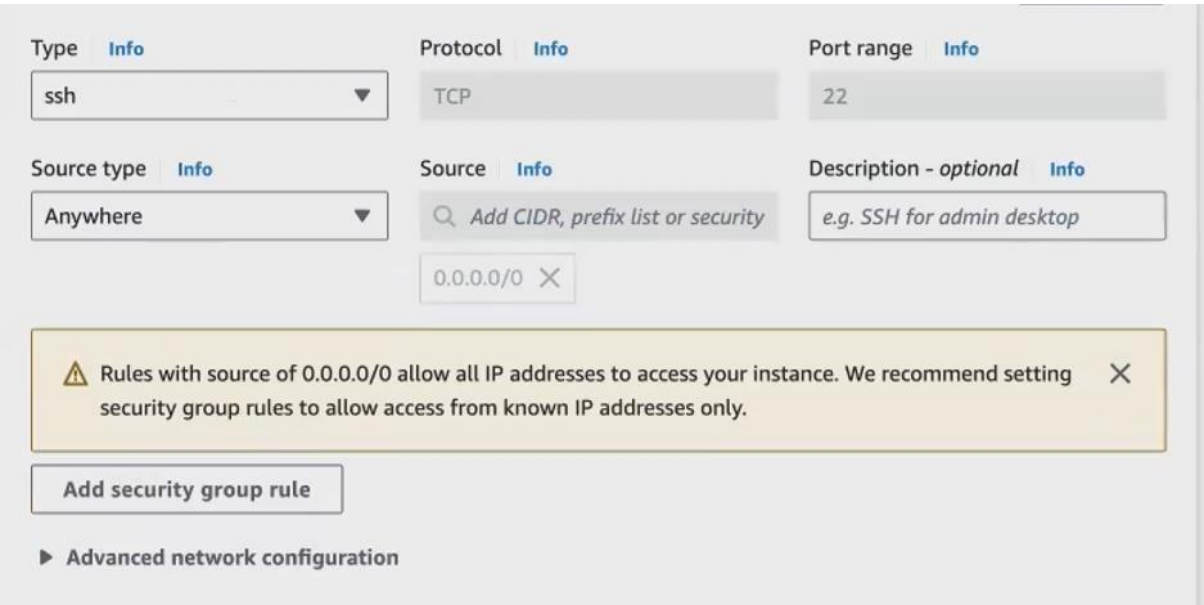
create another ec2 for bastion host



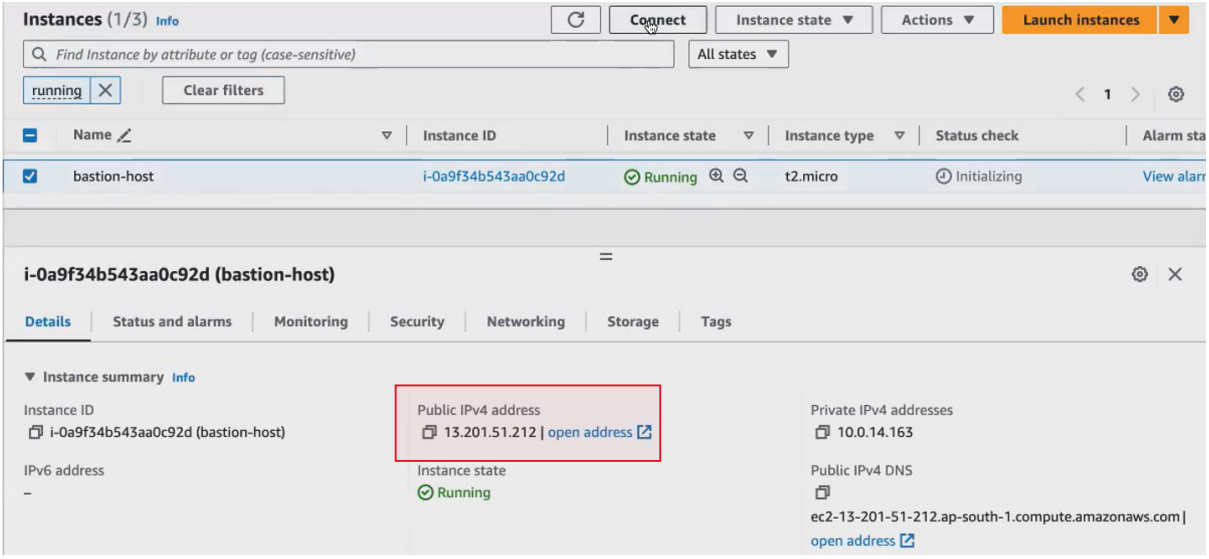


- **Your private Ec2 and your bastion host should share the same pem Key. Okay, make sure you select the same pem key or use the same key that you have used for the private Ec2.So while creating your auto scaling group, it asked you to create a keypad so you can either create a new key pair or you can use the old one itself. But whatever key pair that you're connecting with your private Ec2 make sure you attach the same key here with your bastion host also.**



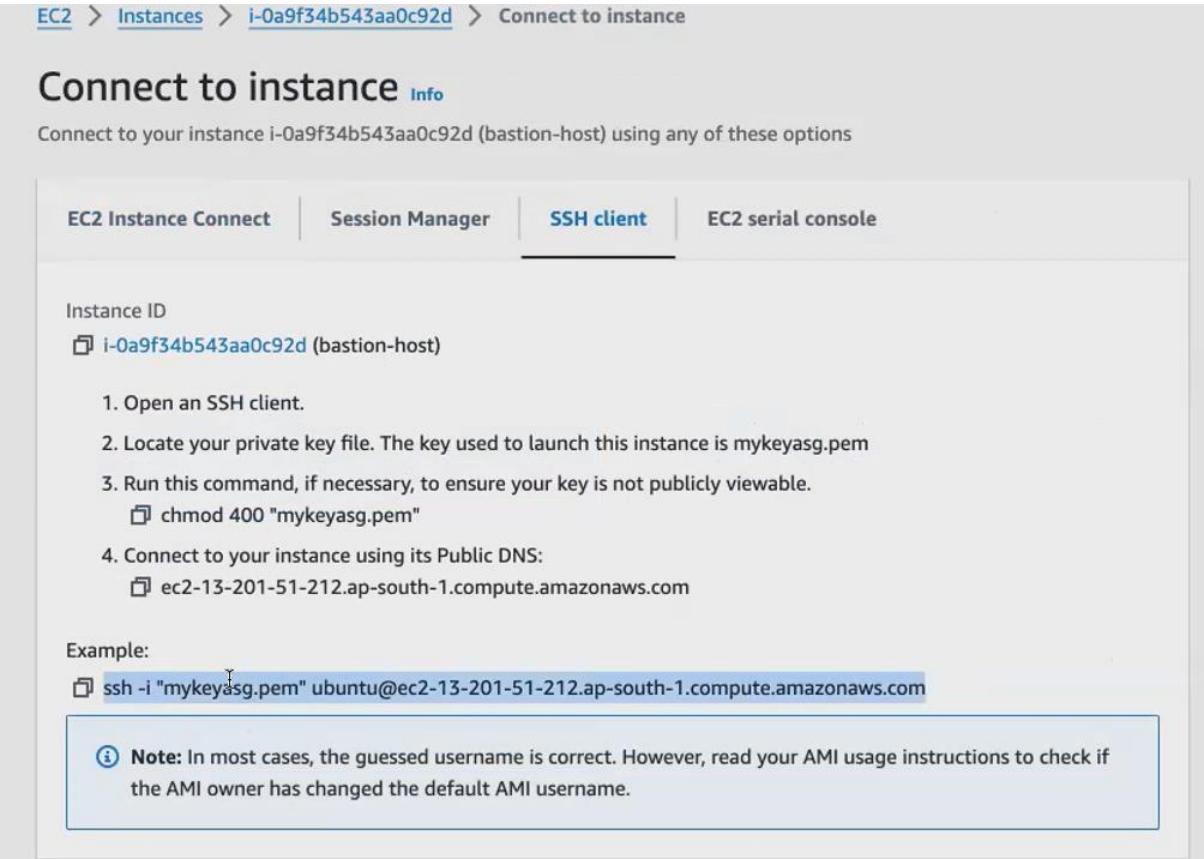


Click launch instance for bastion host .its public



I will be moving my pem key to this bastion host **only then, from the bastion host I can connect with my private ec2. Okay, so to connect with your private ec2. Your bastion host should have the pem key like how you have it in your downloads folder. You have the pem key in your downloads folder, so that with the pem key your laptop can connect with your Ec2 Same thing you are going to do with the pem key. Your bastion host can connect with your private ec2.**

Connect with bastion host ec2. After connect into terminal goto download folder →give chmod comd → copy ssh cmd in terminal



How to copy that pemkey to bastion host

So you use scp, which means secure copy. So you use a command to copy the files from your local to the remote mission. Now, my local, I'm copying the pem key from my laptop to my bastion host . Okay, so we give scp. So after I you have to give your key identity file [-i] and what you wanted to copy in your remote mission. So if it is located in a different folder.

```
yasmin@abdul-personal ~ % cd Downloads
yasmin@abdul-personal Downloads % ls | grep  mykey
mykey.pem
mykeyasg.pem
yasmin@abdul-personal Downloads % pwd
/Users/yasmin/Downloads
yasmin@abdul-personal Downloads % chmod 400 mykeyasg
chmod: mykeyasg: No such file or directory
yasmin@abdul-personal Downloads % chmod 400 mykeyasg.pem
yasmin@abdul-personal Downloads % scp -i mykeyasg.pem mykeyasg.pem ubuntu@13.201.51.212:/home/ubuntu/
The authenticity of host '13.201.51.212 (13.201.51.212)' can't be established.
ED25519 key fingerprint is SHA256:5hC9vLJ4uz+x/2334M/4r6txnVHpM4u5z61cUq8S6ck.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.201.51.212' (ED25519) to the list of known hosts.
mykeyasg.pem
yasmin@abdul-personal Downloads % ssh -i mykeyasg.pem ubuntu@13.201.51.212
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Tue Jul  2 11:56:40 UTC 2024

  System load:  0.0048828125      Processes:             102
  Usage of /:   20.5% of 7.57GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for eth0: 10.0.14.163
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-14-163:~$ ls
mykeyasg.pem
ubuntu@ip-10-0-14-163:~$
```

want to attach a load, balancer application load balancer, we have to install some application inside, we should have an application running inside the private Ec 2 mission, so that we can access the application by a load balancer endpoint. So I will go and use the same user data. So let me create the script.

```
#!/bin/bash

# Installing Dependencies
echo "########################################"
echo "Installing packages."
echo "########################################"
sudo apt-get update > /dev/null
sudo apt-get install wget unzip apache2 -y > /dev/null
echo

# Start & Enable Service
echo "########################################"
echo "Start & Enable Apache2 Service"
echo "########################################"
sudo systemctl start apache2
sudo systemctl enable apache2
echo

# Creating Temp Directory
echo "########################################"
echo "Starting Artifact Deployment"
echo "########################################"
mkdir -p /tmp/webfiles
cd /tmp/webfiles
echo

wget https://www.tooplate.com/zip-templates/2098_health.zip > /dev/null
unzip 2098_health.zip > /dev/null
sudo cp -r 2098_health/* /var/www/html/
echo

# Bounce Service
echo "########################################"
echo "Restarting Apache2 service"
echo "########################################"
sudo systemctl restart apache2
echo

# Clean Up
echo "########################################"
echo "Removing Temporary Files"
echo "########################################"
rm -rf /tmp/webfiles
echo

sudo systemctl status apache2
ls /var/www/html/
~
~
```

```
ubuntu@ip-10-0-149-222:~$ vi website.sh
ubuntu@ip-10-0-149-222:~$ chmod +x website.sh
ubuntu@ip-10-0-149-222:~$ ./website.sh
##########################################
Installing packages.
##########################################
```

```
#####################################
Restarting Apache2 service
#####################################

#####################################
Removing Temporary Files
#####################################

● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2024-07-02 12:02:33 UTC; 30ms ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 2679 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 2684 (apache2)
      Tasks: 55 (limit: 1121)
     Memory: 4.8M
        CPU: 25ms
     CGroup: /system.slice/apache2.service
             ├─2684 /usr/sbin/apache2 -k start
             ├─2687 /usr/sbin/apache2 -k start
             └─2688 /usr/sbin/apache2 -k start
```

```
ubuntu@ip-10-0-149-222:~$ exit
logout
Connection to 10.0.149.222 closed.
ubuntu@ip-10-0-14-163:~$
```

now the application is running. You can see Apache server is installed, and it is running the application. Now we will come out of this private ec2 .We will again go and launch. We will connect with another private, because we have 2 private Ec2 one we have connected, and we have also launched the application.

```
ubuntu@ip-10-0-14-163:~$ ssh -i mykeyasg.pem ubuntu@10.0.129.245
The authenticity of host '10.0.129.245 (10.0.129.245)' can't be established.
ED25519 key fingerprint is SHA256:jcapPzfUk6rngwEiJ4TaToVl81+8i6WVvqmFOxGdo2Y.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.129.245' (ED25519) to the list of known hosts.
```

```
#!/bin/bash

# Installing Dependencies
echo "#######################################"
echo "Installing packages."
echo "#######################################"
sudo apt-get update > /dev/null
sudo apt-get install wget unzip apache2 -y > /dev/null
echo

# Start & Enable Service
echo "#######################################"
echo "Start & Enable Apache2 Service"
echo "#######################################"
sudo systemctl start apache2
sudo systemctl enable apache2
echo

# Creating Temp Directory
echo "#######################################"
echo "Starting Artifact Deployment"
echo "#######################################"
mkdir -p /tmp/webfiles
cd /tmp/webfiles
echo

wget https://www.tooplate.com/zip-templates/2098_health.zip > /dev/null
unzip 2098_health.zip > /dev/null
sudo cp -r 2098_health/* /var/www/html/
echo

# Bounce Service
echo "#######################################"
echo "Restarting Apache2 service"
echo "#######################################"
sudo systemctl restart apache2
echo

# Clean Up
echo "#######################################"
echo "Removing Temporary Files"
echo "#######################################"
rm -rf /tmp/webfiles
echo

sudo systemctl status apache2
ls /var/www/html/
```

```
ubuntu@ip-10-0-129-245:~$ vi website.sh
ubuntu@ip-10-0-129-245:~$ chmod +x website.sh
ubuntu@ip-10-0-129-245:~$ ./website.sh
#########################################
Installing packages.
#########################################
```

```
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2024-07-02 12:05:30 UTC; 34ms ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 2647 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 2652 (apache2)
      Tasks: 55 (limit: 1121)
     Memory: 4.7M
        CPU: 27ms
     CGroup: /system.slice/apache2.service
             ├─2652 /usr/sbin/apache2 -k start
             ├─2655 /usr/sbin/apache2 -k start
             └─2656 /usr/sbin/apache2 -k start

Jul 02 12:05:30 ip-10-0-129-245 systemd[1]: Starting The Apache HTTP Server...
Jul 02 12:05:30 ip-10-0-129-245 systemd[1]: Started The Apache HTTP Server.
'ABOUT THIS TEMPLATE.txt'   css   fonts   images   index.html   js   news-detail.html
ubuntu@ip-10-0-129-245:~$ exit
logout
Connection to 10.0.129.245 closed.
ubuntu@ip-10-0-14-163:~$
```

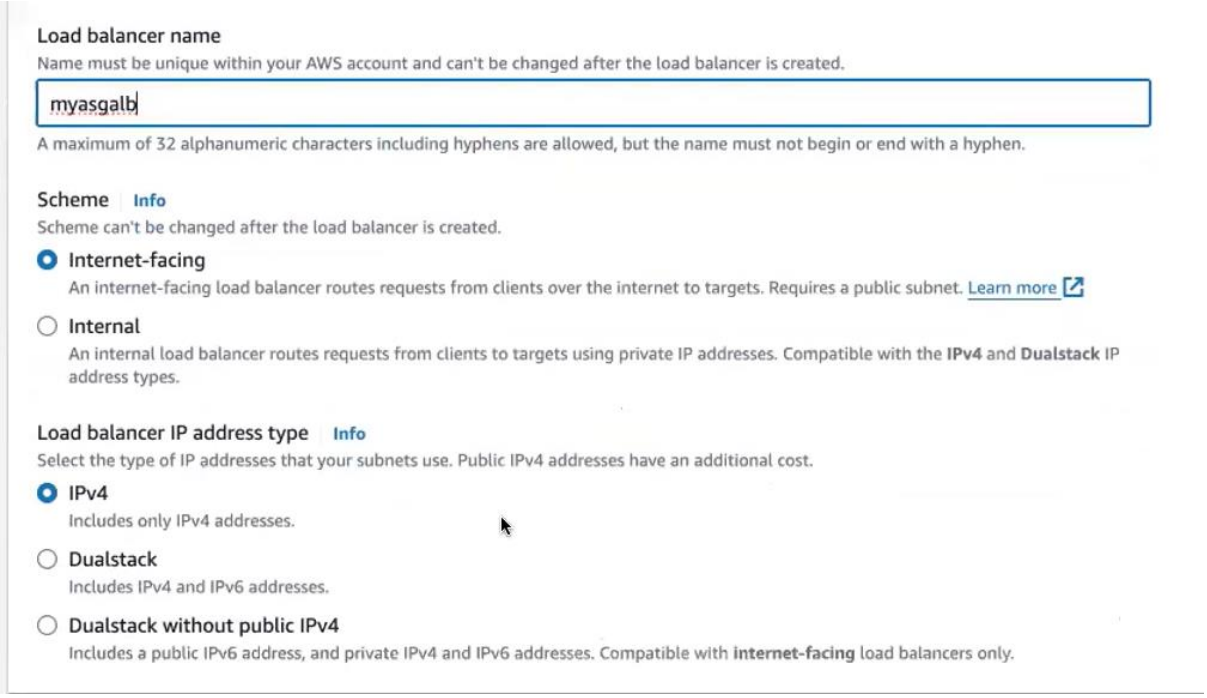
```
ubuntu@ip-10-0-14-163:~$ ssh -i mykeyasg.pem ubuntu@10.0.149.222

The authenticity of host '10.0.149.222 (10.0.149.222)' can't be established.
ED25519 key fingerprint is SHA256:oD69r+OmqRetmgxlg/dvWiVzGVHmGWolfRHHur9osq8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.149.222' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Tue Jul  2 11:59:37 UTC 2024

  System load:  0.0               Processes:             96
  Usage of /:   20.5% of 7.57GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for eth0: 10.0.149.222
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-149-222:~$
```

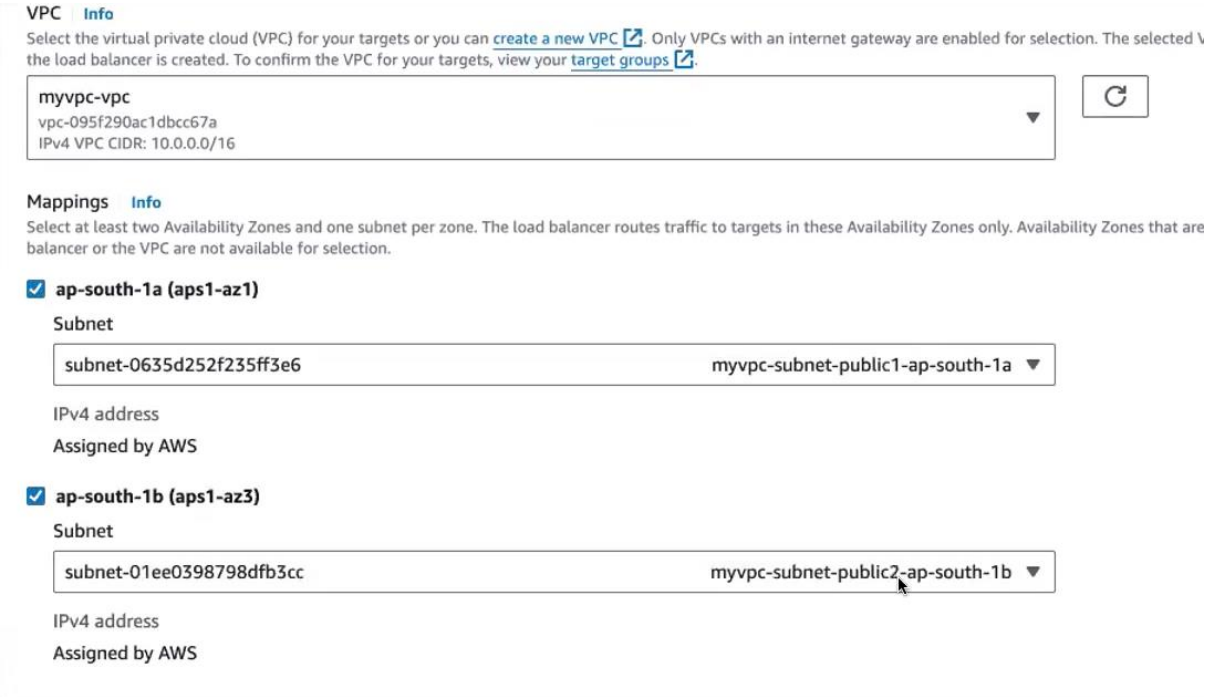
Now 2 application running .for creating load balancer we create 2 ec2 & 2 appli server.

launch the load balancer. So now I have created the Auto Scaling group. I have launched the private Ec2. So, in the private Ec2, my application is running here. Now we are going to launch the application load balancer on top of my application, load balancer, and creating inside the public. Submit because my

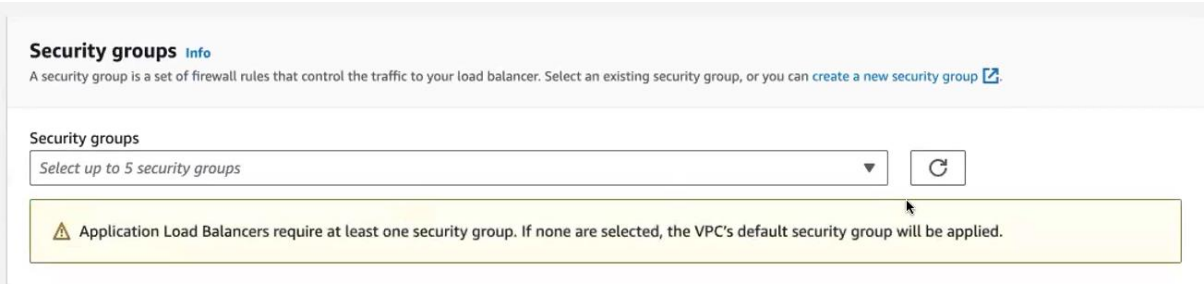
application load balancer is the endpoint gives me the endpoint. I have to access the endpoint in the browser. So, I am. I'm creating. I'm putting my load balancer in the public sub. Only then I can connect or talk with my load balancer through the Internet. Okay? So, what we what we do, we go to the load balancer.

Click create load balancer →

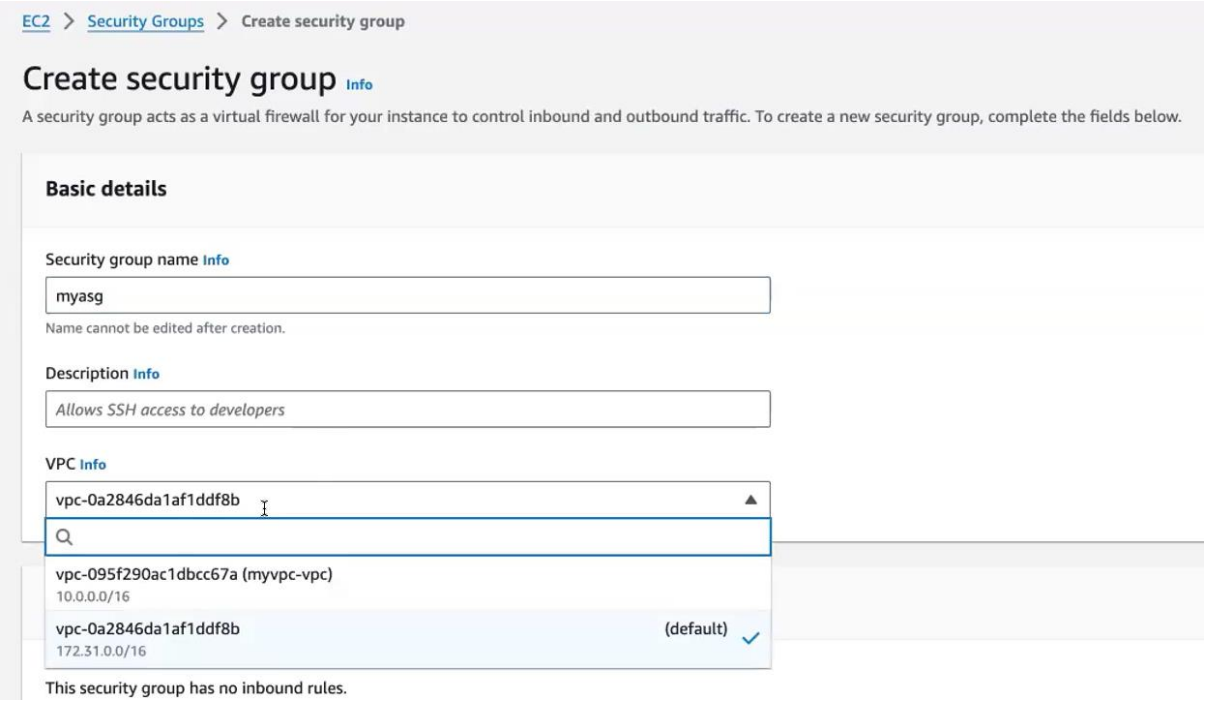




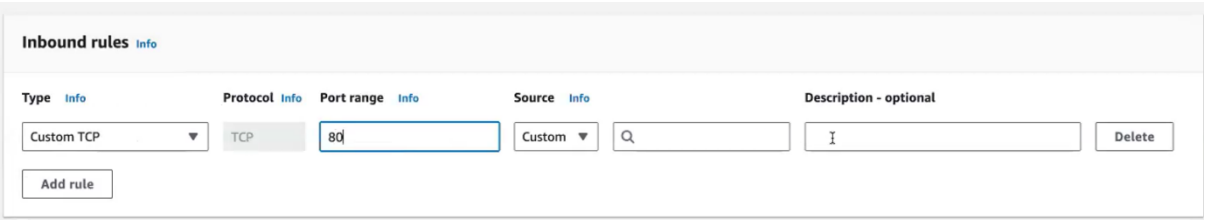
Put the subnet only in public then only we access the application via browser



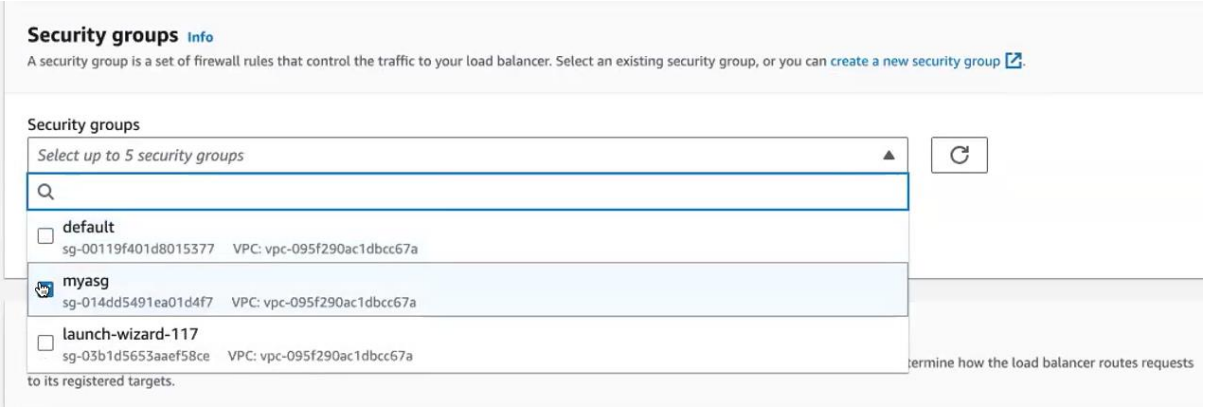
Remove the default security group ..then create new security group



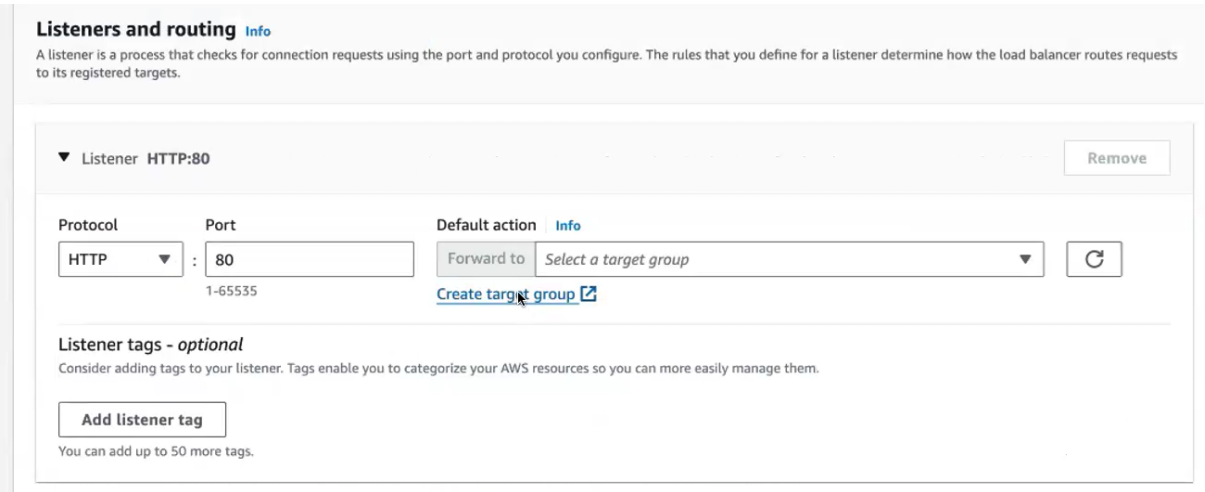
In description give same myasg .add inbound rule

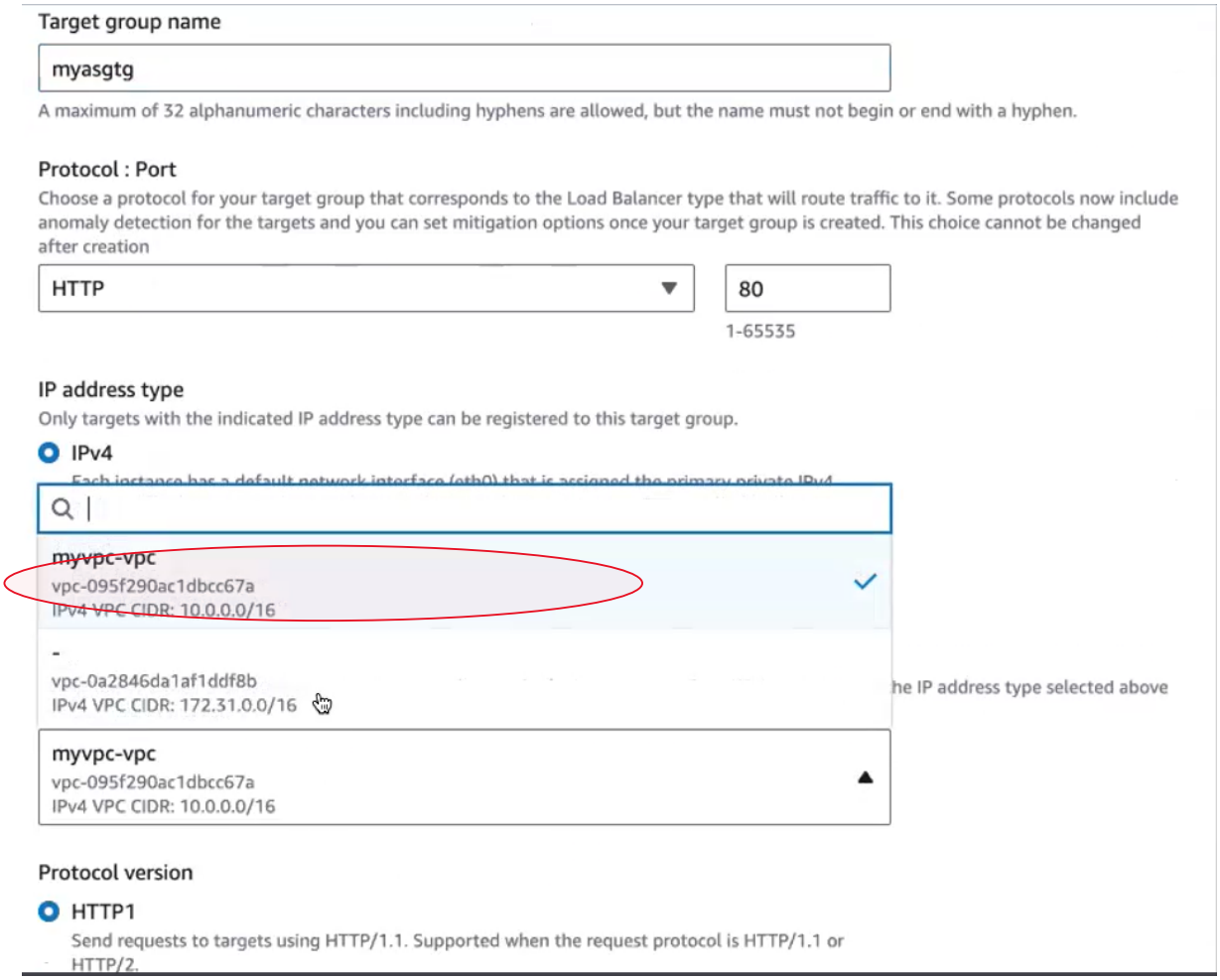


Once refresh it ,created s.g available..click this..now s,g created

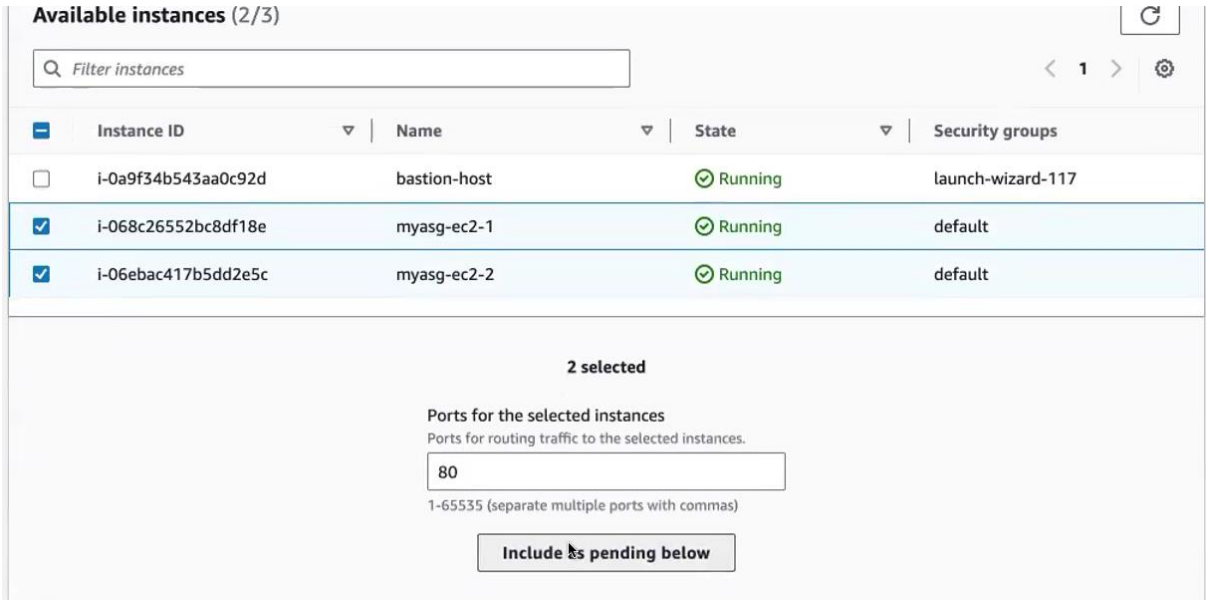


<u>After this create target group</u>



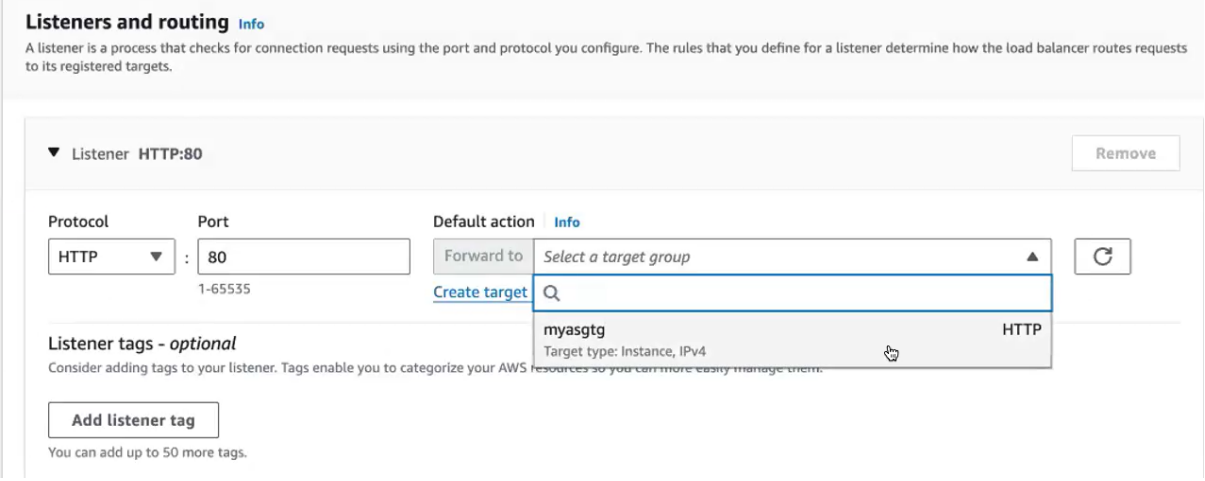


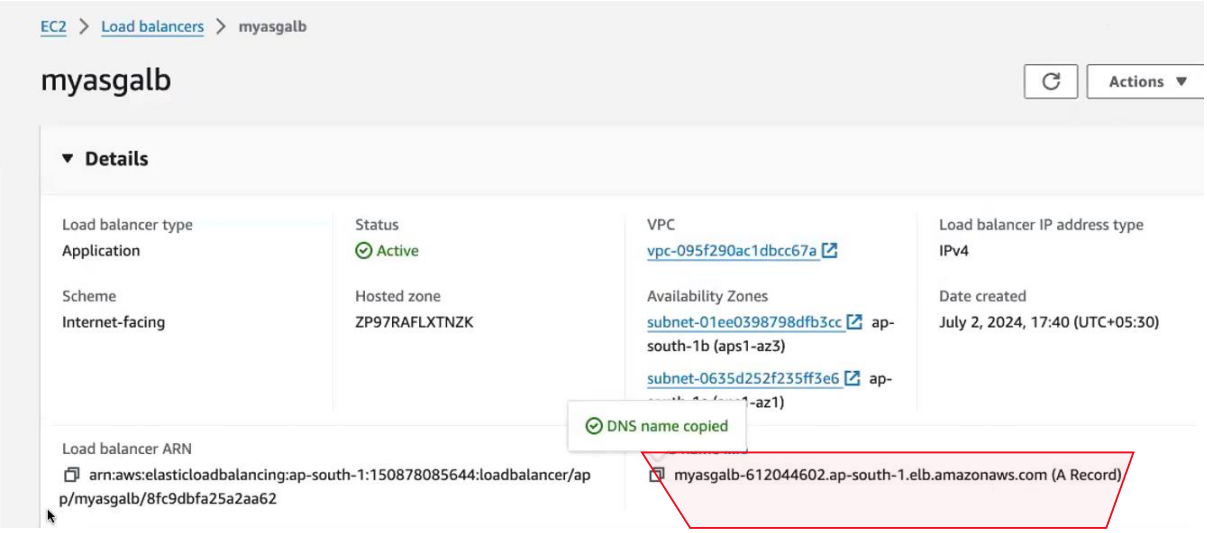
Select only auto scaling group .give port no 80



Click create target group. Now app is running

After creating target group came to application creation page ,refresh target group. Created target group shown .select this. Click create load balancer.





Copy the DNS paste it in browser. Application will running

now you're you're running the application in the private instance only. but you're connecting it via the load balancer, so that your private instance is much more secure. So you have launched, you have access to everything because your private instance is provided from the Internet. You've connected it through the bastion host and from the bastion host you have installed everything. You have launched your application inside your private

documentation for this.

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-example-private-subnets-nat.html