Create a simple web application with multiple web servers and implement load balancing using AWS services. The project will involve setting up EC2 instances, configuring an Elastic Load Balancer (ELB),testing the load balancing functionality, autoscaling the servers

1.Launching Two EC2 Instances

Here we are going to launch two Ec2 Instances in US-east-1a & Us-east-1b

EC2 Specifications

Name-Webserver 1

Name-Webserver2

AMI-Amazon Linux 2023

Instance type-t2.micro

Key pair- p11.pem for Webserver1

Key pair- p12.pem for Webserver2

Networking

VPC- Default VPC

Subnet – Subnet in us-east-1a for Websever1

Subnet – Subnet in us-east-1b for Websever2

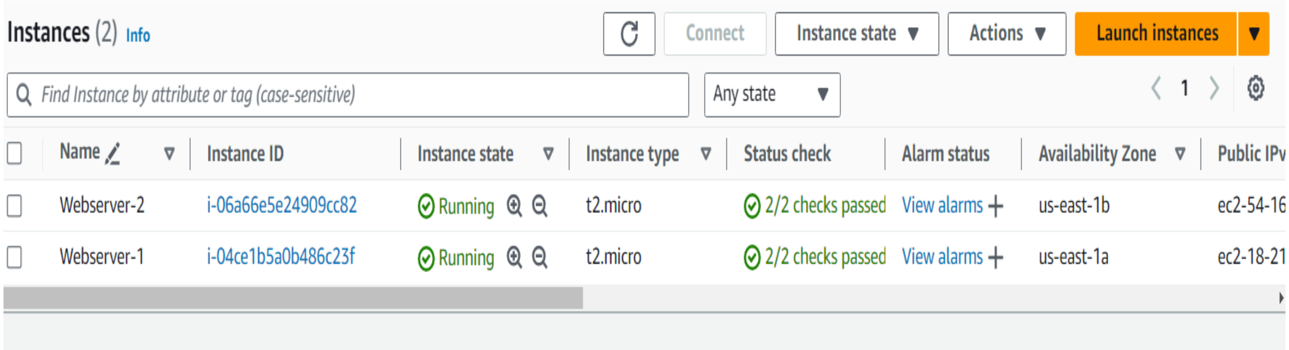
Security group

Create an SG which has SSH and HTTP access and attach it to both the instances.

Storage-

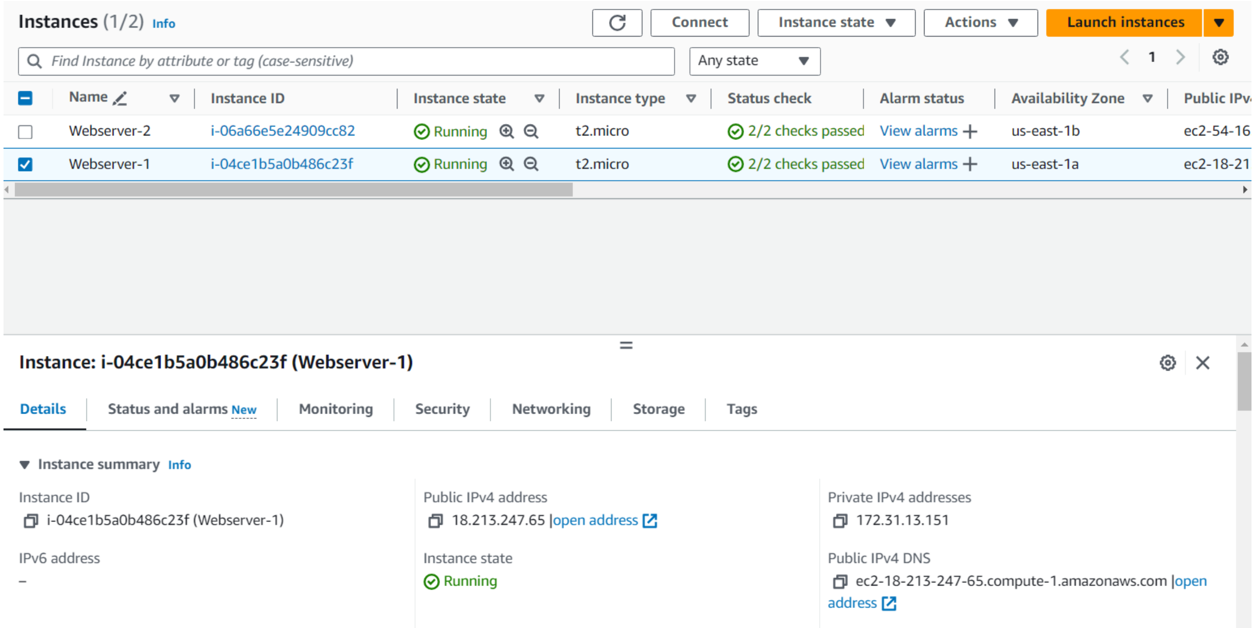
Keep default storage = 8GB as we are launching Linux instance.

Once launched wait till passing 2/2 check



Instances (2) Info

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
☐	Webserver-2	i-06a66e5e24909cc82	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-54-16
☐	Webserver-1	i-04ce1b5a0b486c23f	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-18-21



Instances (1/2) Info

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
☐	Webserver-2	i-06a66e5e24909cc82	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-54-16
☑	Webserver-1	i-04ce1b5a0b486c23f	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-18-21

Instance: i-04ce1b5a0b486c23f (Webserver-1)

▼ Instance summary Info

Instance ID
i-04ce1b5a0b486c23f (Webserver-1)

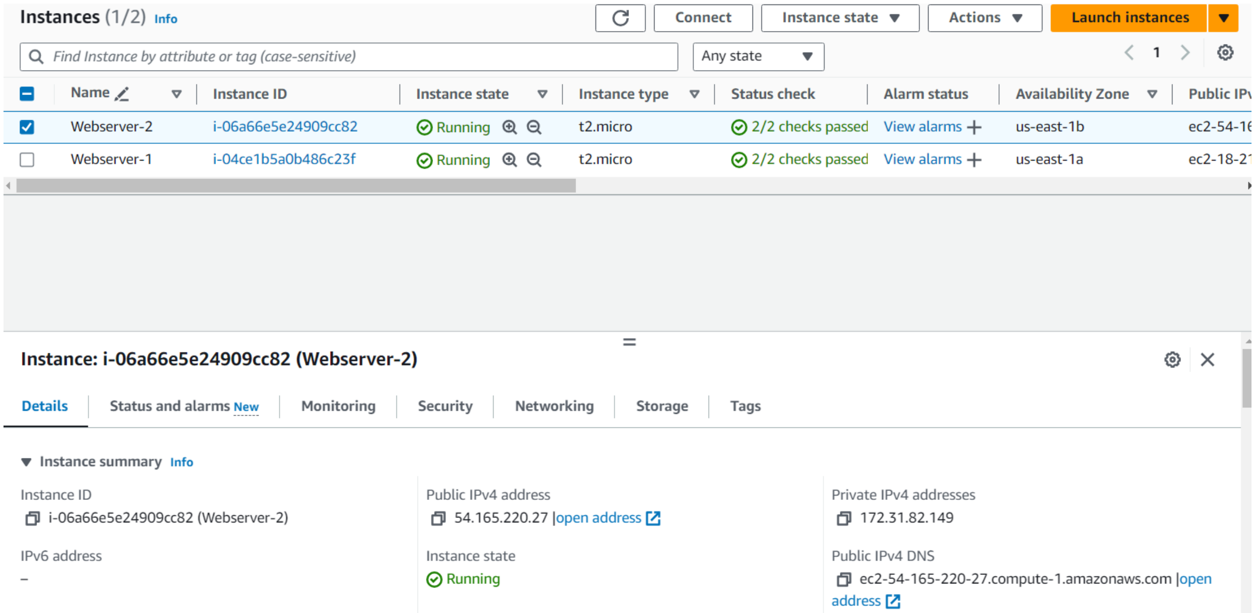
Public IPv4 address
18.213.247.65 |open address

Private IPv4 addresses
172.31.13.151

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-18-213-247-65.compute-1.amazonaws.com |open address



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
☑	Webserver-2	i-06a66e5e24909cc82	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us-east-1b	ec2-54-16
□	Webserver-1	i-04ce1b5a0b486c23f	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us-east-1a	ec2-18-2

Instance: i-06a66e5e24909cc82 (Webserver-2)

▼ Instance summary Info

Instance ID
i-06a66e5e24909cc82 (Webserver-2)

Public IPv4 address
54.165.220.27 |open address

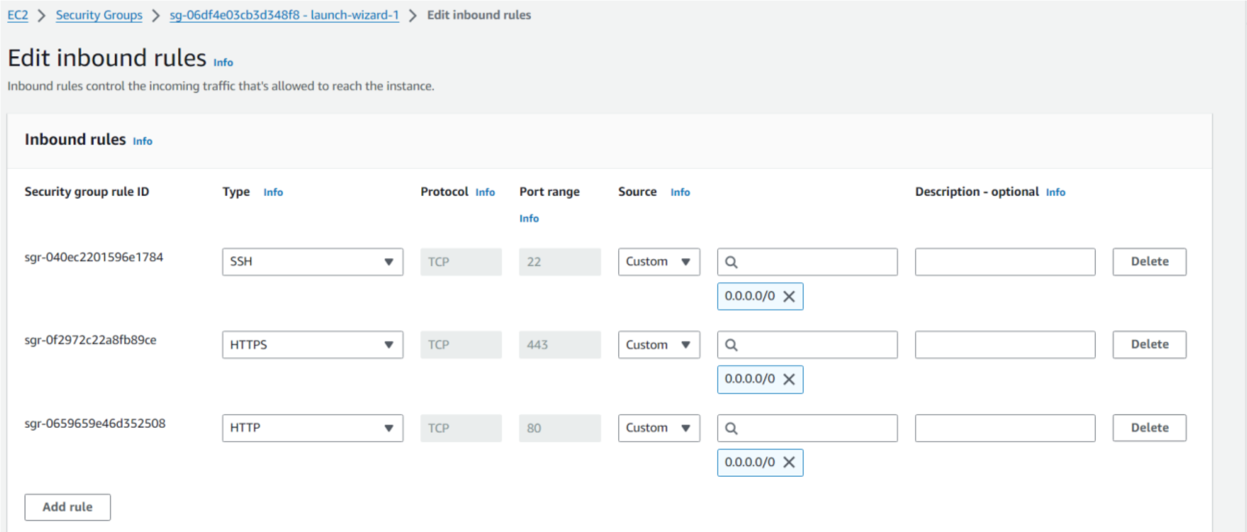
Private IPv4 addresses
172.31.82.149

IPv6 address
–

Instance state
✓ Running

Public IPv4 DNS
ec2-54-165-220-27.compute-1.amazonaws.com |open address

Security group


EC2 > Security Groups > sg-06df4e03cb3d348f8 - launch-wizard-1 > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

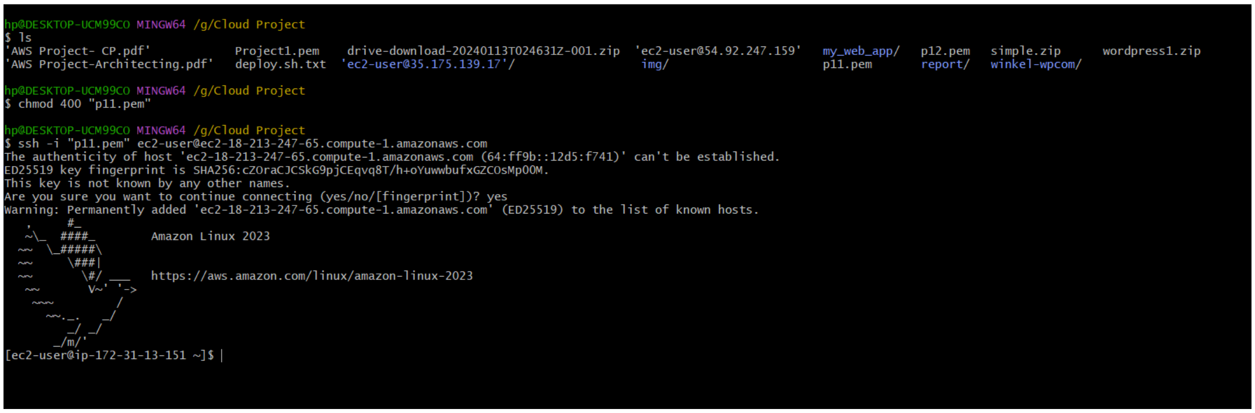
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-040ec2201596e1784	SSH ▼	TCP	22	Custom ▼	0.0.0.0/0 ✕		Delete
sgr-0f2972c22a8fb89ce	HTTPS ▼	TCP	443	Custom ▼	0.0.0.0/0 ✕		Delete
sgr-0659659e46d352508	HTTP ▼	TCP	80	Custom ▼	0.0.0.0/0 ✕		Delete

SSH to Webserver 1 & install Apache webserver on it

chmod 400 "p11.pem"

ssh -i "p11.pem" ec2-user@ec2-18-213-247-65.compute-1.amazonaws.com

you will get access to Webserver-1

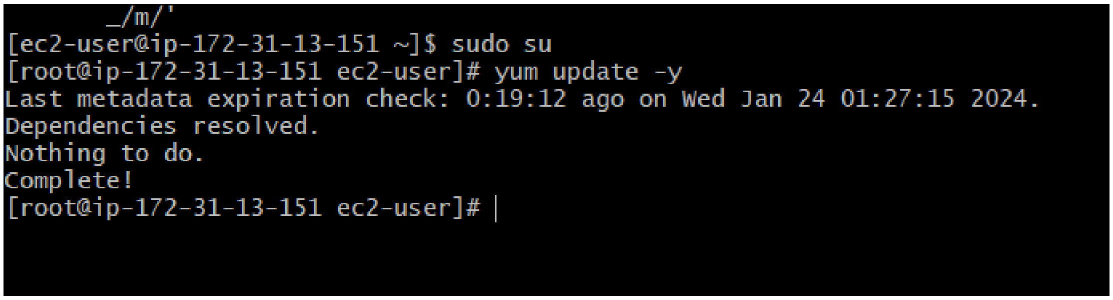


```
hp@DESKTOP-UCM99CO MINGW64 /g/Cloud Project
$ ls
'AWS Project- CP.pdf'            Project1.pem    drive-download-20240113T024631Z-001.zip  'ec2-user@54.92.247.159'   my_web_app/   p12.pem   simple.zip      wordpress1.zip
'AWS Project-Architecting.pdf'   deploy.sh.txt  'ec2-user@35.175.139.17'/                  img/                      p11.pem       report/   winkel-wpcom/

hp@DESKTOP-UCM99CO MINGW64 /g/Cloud Project
$ chmod 400 "p11.pem"

hp@DESKTOP-UCM99CO MINGW64 /g/Cloud Project
$ ssh -i "p11.pem" ec2-user@ec2-18-213-247-65.compute-1.amazonaws.com
The authenticity of host 'ec2-18-213-247-65.compute-1.amazonaws.com (64:ff9b::12d5:f741)' can't be established.
ED25519 key fingerprint is SHA256:cZOraCJCSkG9pjCEqvq8T/h+oYuwwbufxGZCOsMp0OM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-213-247-65.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-13-151 ~]$ |
```

Execute Following commands on Webserver1

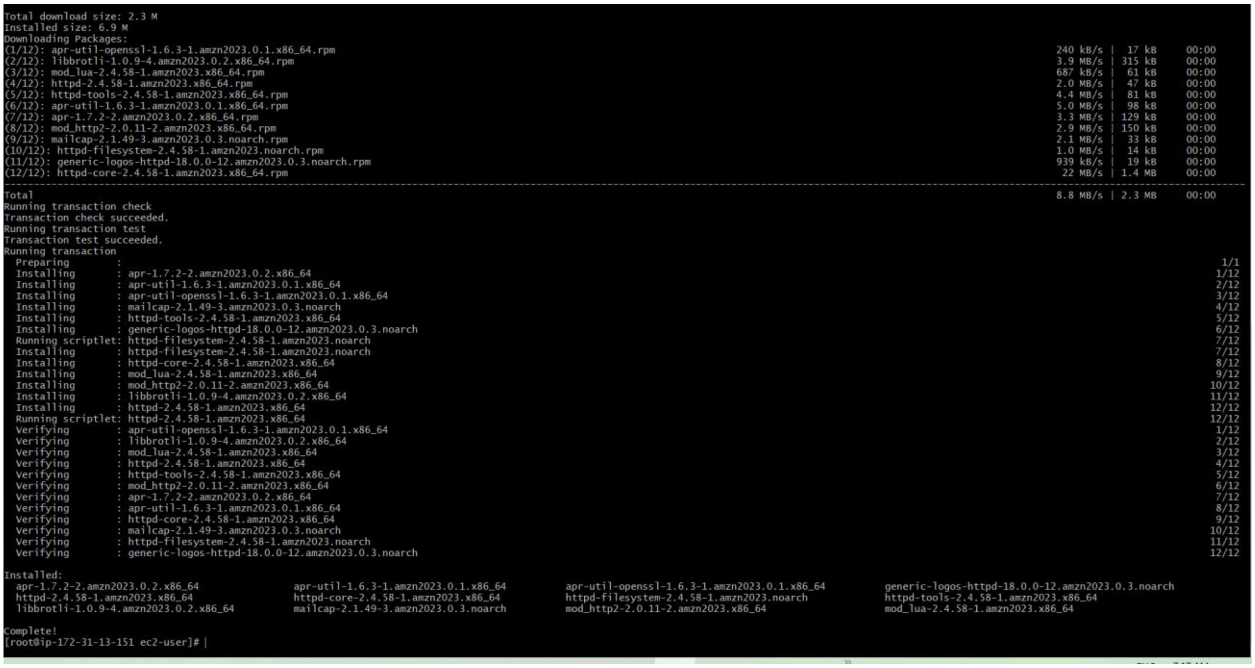
sudo su

yum update –y



```
        _/m/'
[ec2-user@ip-172-31-13-151 ~]$ sudo su
[root@ip-172-31-13-151 ec2-user]# yum update -y
Last metadata expiration check: 0:19:12 ago on Wed Jan 24 01:27:15 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-13-151 ec2-user]# |
```

yum install -y httpd



```
Total download size: 2.3 M
Installed size: 6.9 M
Downloading Packages:
(1/12): apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64.rpm                                            240 kB/s |  17 kB     00:00
(2/12): libbrotli-1.0.9-4.amzn2023.0.2.x86_64.rpm                                                   3.9 MB/s | 315 kB     00:00
(3/12): mod_lua-2.4.58-1.amzn2023.x86_64.rpm                                                        687 kB/s |  61 kB     00:00
(4/12): httpd-2.4.58-1.amzn2023.x86_64.rpm                                                          2.0 MB/s |  47 kB     00:00
(5/12): httpd-tools-2.4.58-1.amzn2023.x86_64.rpm                                                    4.4 MB/s |  81 kB     00:00
(6/12): apr-util-1.6.3-1.amzn2023.0.1.x86_64.rpm                                                    5.0 MB/s |  98 kB     00:00
(7/12): apr-1.7.2-2.amzn2023.0.2.x86_64.rpm                                                         3.3 MB/s | 129 kB     00:00
(8/12): mod_http2-2.0.11-2.amzn2023.x86_64.rpm                                                      2.9 MB/s | 150 kB     00:00
(9/12): mailcap-2.1.49-3.amzn2023.0.3.noarch.rpm                                                    2.1 MB/s |  33 kB     00:00
(10/12): httpd-filesystem-2.4.58-1.amzn2023.noarch.rpm                                              1.0 MB/s |  14 kB     00:00
(11/12): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm                                      939 kB/s |  19 kB     00:00
(12/12): httpd-core-2.4.58-1.amzn2023.x86_64.rpm                                                     22 MB/s | 1.4 MB     00:00
--------------------------------------------------------------------------------------------------------------------------------------
Total                                                                                               8.8 MB/s | 2.3 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                          1/1
  Installing       : apr-1.7.2-2.amzn2023.0.2.x86_64                                                                         1/12
  Installing       : apr-util-1.6.3-1.amzn2023.0.1.x86_64                                                                    2/12
  Installing       : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64                                                            3/12
  Installing       : mailcap-2.1.49-3.amzn2023.0.3.noarch                                                                    4/12
  Installing       : httpd-tools-2.4.58-1.amzn2023.x86_64                                                                    5/12
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                                       6/12
  Running scriptlet: httpd-filesystem-2.4.58-1.amzn2023.noarch                                                               7/12
  Installing       : httpd-filesystem-2.4.58-1.amzn2023.noarch                                                               7/12
  Installing       : httpd-core-2.4.58-1.amzn2023.x86_64                                                                     8/12
  Installing       : mod_lua-2.4.58-1.amzn2023.x86_64                                                                        9/12
  Installing       : mod_http2-2.0.11-2.amzn2023.x86_64                                                                     10/12
  Installing       : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                                                                  11/12
  Installing       : httpd-2.4.58-1.amzn2023.x86_64                                                                         12/12
  Running scriptlet: httpd-2.4.58-1.amzn2023.x86_64                                                                         12/12
  Verifying        : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64                                                            1/12
  Verifying        : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                                                                   2/12
  Verifying        : mod_lua-2.4.58-1.amzn2023.x86_64                                                                        3/12
  Verifying        : httpd-2.4.58-1.amzn2023.x86_64                                                                          4/12
  Verifying        : httpd-tools-2.4.58-1.amzn2023.x86_64                                                                    5/12
  Verifying        : mod_http2-2.0.11-2.amzn2023.x86_64                                                                      6/12
  Verifying        : apr-1.7.2-2.amzn2023.0.2.x86_64                                                                         7/12
  Verifying        : apr-util-1.6.3-1.amzn2023.0.1.x86_64                                                                    8/12
  Verifying        : httpd-core-2.4.58-1.amzn2023.x86_64                                                                     9/12
  Verifying        : mailcap-2.1.49-3.amzn2023.0.3.noarch                                                                   10/12
  Verifying        : httpd-filesystem-2.4.58-1.amzn2023.noarch                                                              11/12
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                                      12/12

Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64            apr-util-1.6.3-1.amzn2023.0.1.x86_64       apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64      generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  httpd-2.4.58-1.amzn2023.x86_64             httpd-core-2.4.58-1.amzn2023.x86_64        httpd-filesystem-2.4.58-1.amzn2023.noarch         httpd-tools-2.4.58-1.amzn2023.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64      mailcap-2.1.49-3.amzn2023.0.3.noarch       mod_http2-2.0.11-2.amzn2023.x86_64                mod_lua-2.4.58-1.amzn2023.x86_64

Complete!
[root@ip-172-31-13-151 ec2-user]# |
```

mkdir temp

cd temp

chown ec2-user:ec2-user temp/

chmod 700 temp/

exit

exit

scp -i p11.pem simple.zip ec2-user@18.213.247.65:/home/ec2-user/temp/

login again to webserver

ssh -i "p11.pem" ec2-user@ec2-18-213-247-65.compute-1.amazonaws.com

cd temp

you will find sample.zip file

unzip simple.zip

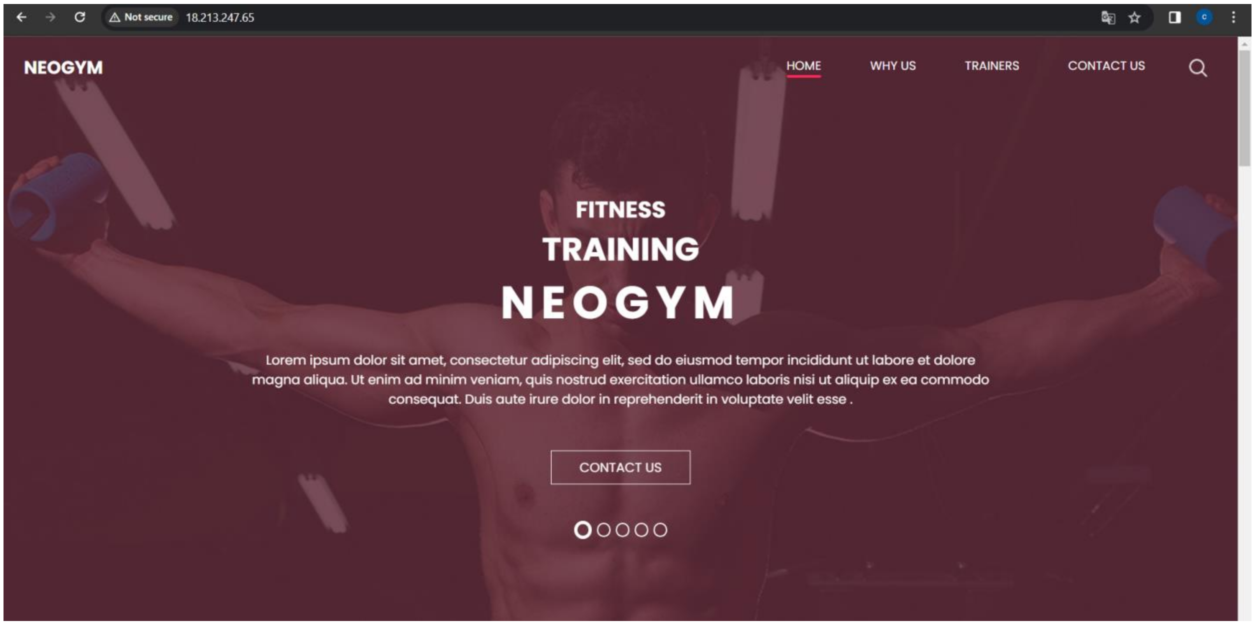
```
cd simple
```

```
mv * /var/www/html
```

```
systemctl enable httpd
```

```
systemctl start httpd
```

copy public ip of webserver1 and see whether webpage is accessible or not



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse .

We successfully installed web application on webserver1

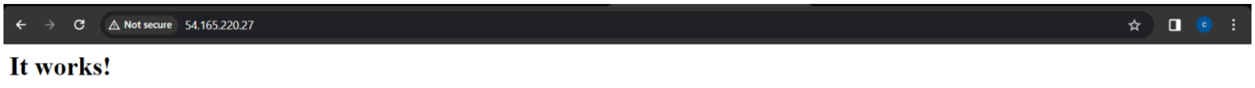
Repeat this process to install web application on Webserver2

```
chmod 400 "p12.pem"
```

```
ssh -i "p12.pem" ec2-user@ec2-54-165-220-27.compute-1.amazonaws.com
```

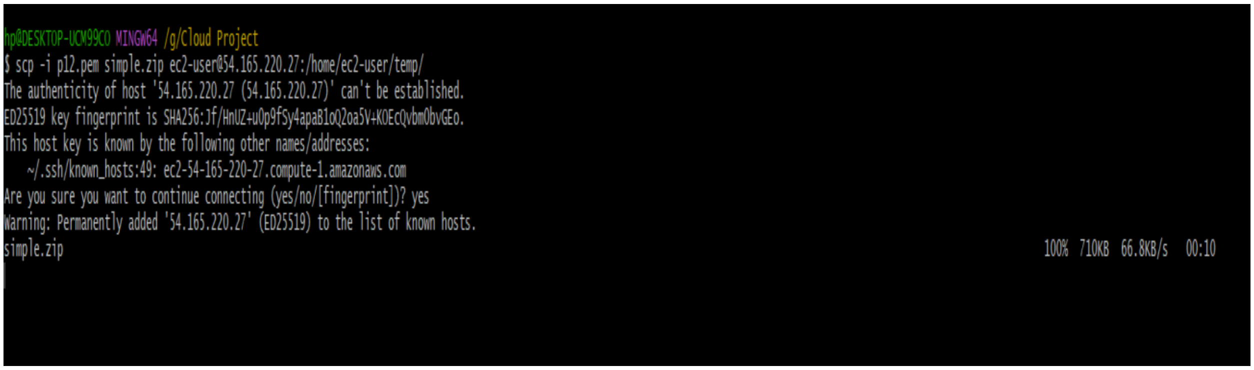
rest of the procedure will be same as per Webserver1

As we just started httpd without installing web app it looks like



It works!

scp -i p12.pem simple.zip ec2-user@54.165.220.27:/home/ec2-user/temp/



```
hp@DESKTOP-UCM99CO MINGW64 /g/Cloud Project
$ scp -i p12.pem simple.zip ec2-user@54.165.220.27:/home/ec2-user/temp/
The authenticity of host '54.165.220.27 (54.165.220.27)' can't be established.
ED25519 key fingerprint is SHA256:Jf/HnUZ+u0p9fSy4apaB1oQ2oa5V+KOEcQvbm0bvGEo.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:49: ec2-54-165-220-27.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.165.220.27' (ED25519) to the list of known hosts.
simple.zip                                                                        100%  710KB  66.8KB/s   00:10
```

Log in to webserver2

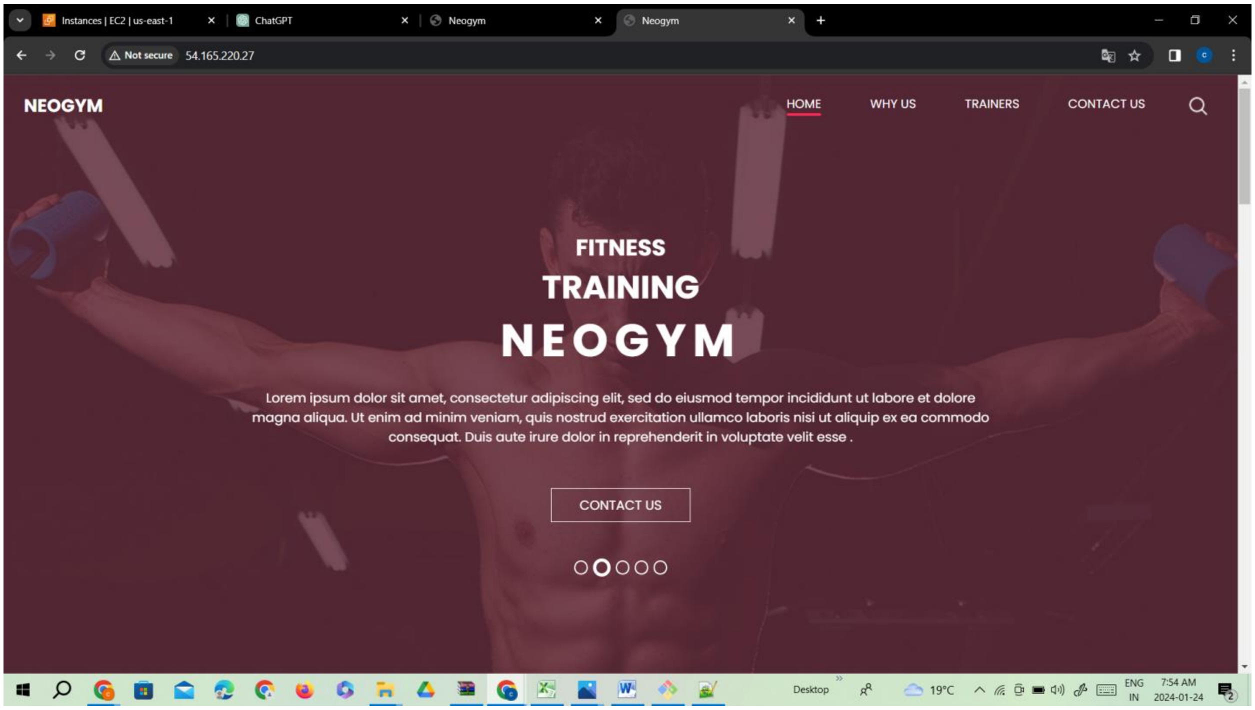
sudo su

cd temp/

unzip simple.zip

cd simple

mv * /var/www/html

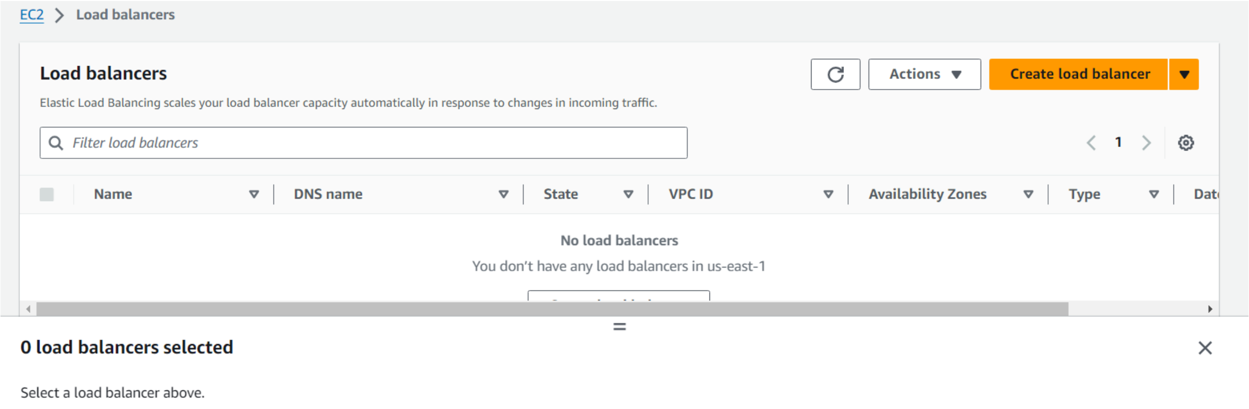


FITNESS
TRAINING
NEOGYM

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse .

Configuring load balancer

Create load balancer



Load balancers

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

□	Name	DNS name	State	VPC ID	Availability Zones	Type	Dat

No load balancers

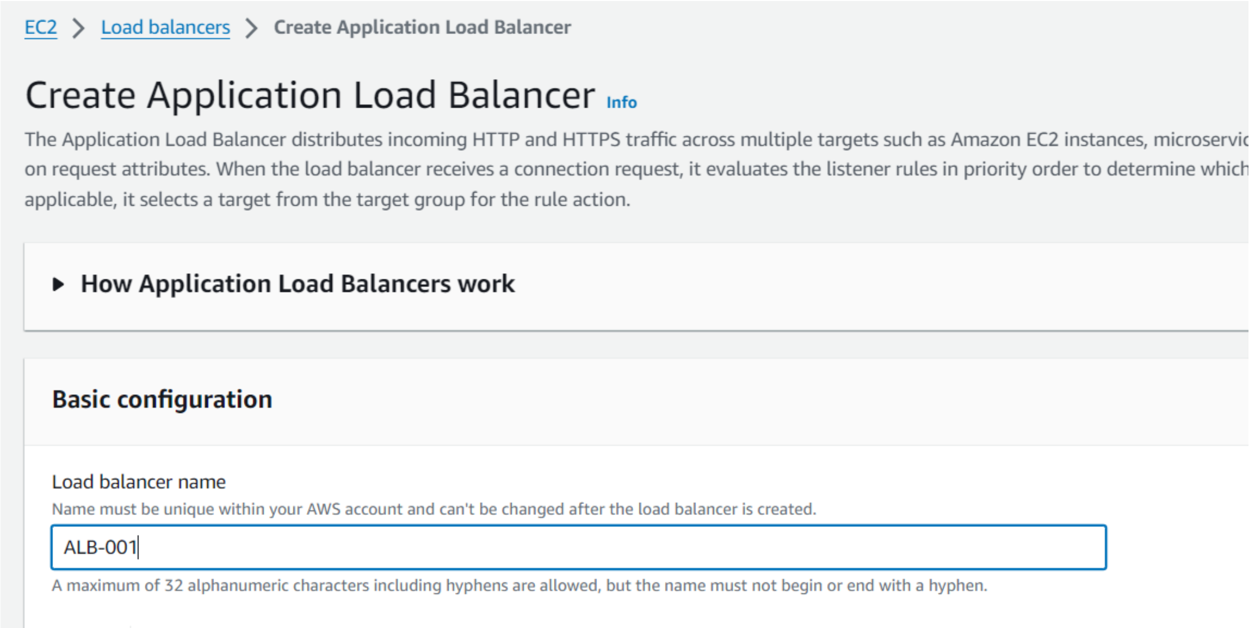
You don't have any load balancers in us-east-1

0 load balancers selected

Select a load balancer above.

Create an Application load balancer

Load balancer name- ALB-001



EC2 > Load balancers > Create Application Load Balancer

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservic on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which applicable, it selects a target from the target group for the rule action.

▸ How Application Load Balancers work

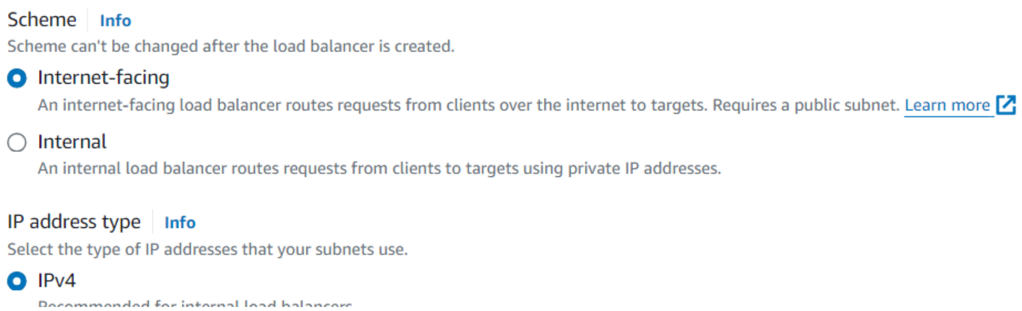
Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

ALB-001

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



Scheme Info

Scheme can't be changed after the load balancer is created.

● Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more

○ Internal

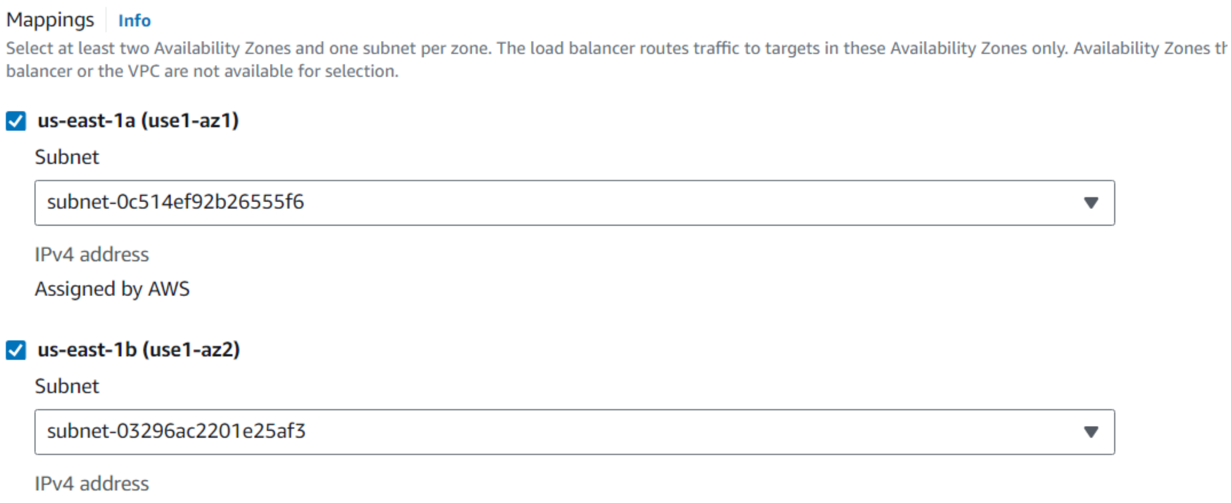
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info

Select the type of IP addresses that your subnets use.

● IPv4

Recommended for internal load balancers



Mappings Info

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones th
balancer or the VPC are not available for selection.

☑ us-east-1a (use1-az1)

Subnet

subnet-0c514ef92b26555f6

IPv4 address

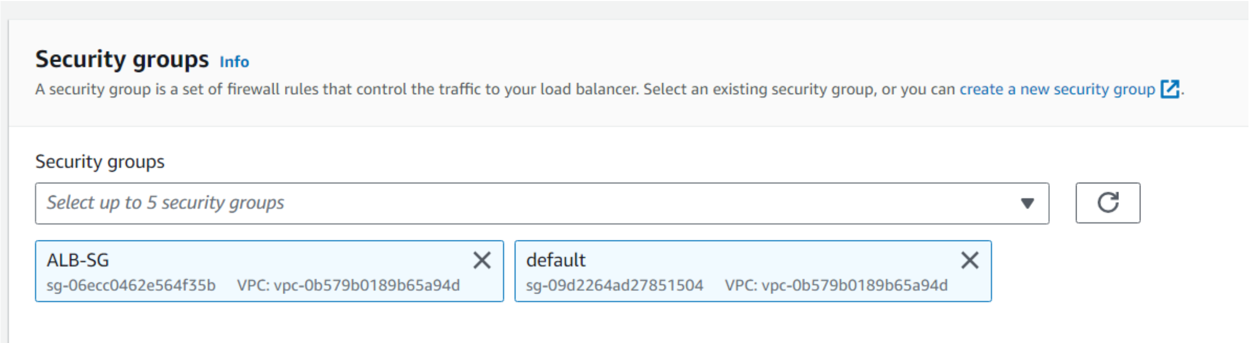
Assigned by AWS

☑ us-east-1b (use1-az2)

Subnet

subnet-03296ac2201e25af3

IPv4 address



Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group.

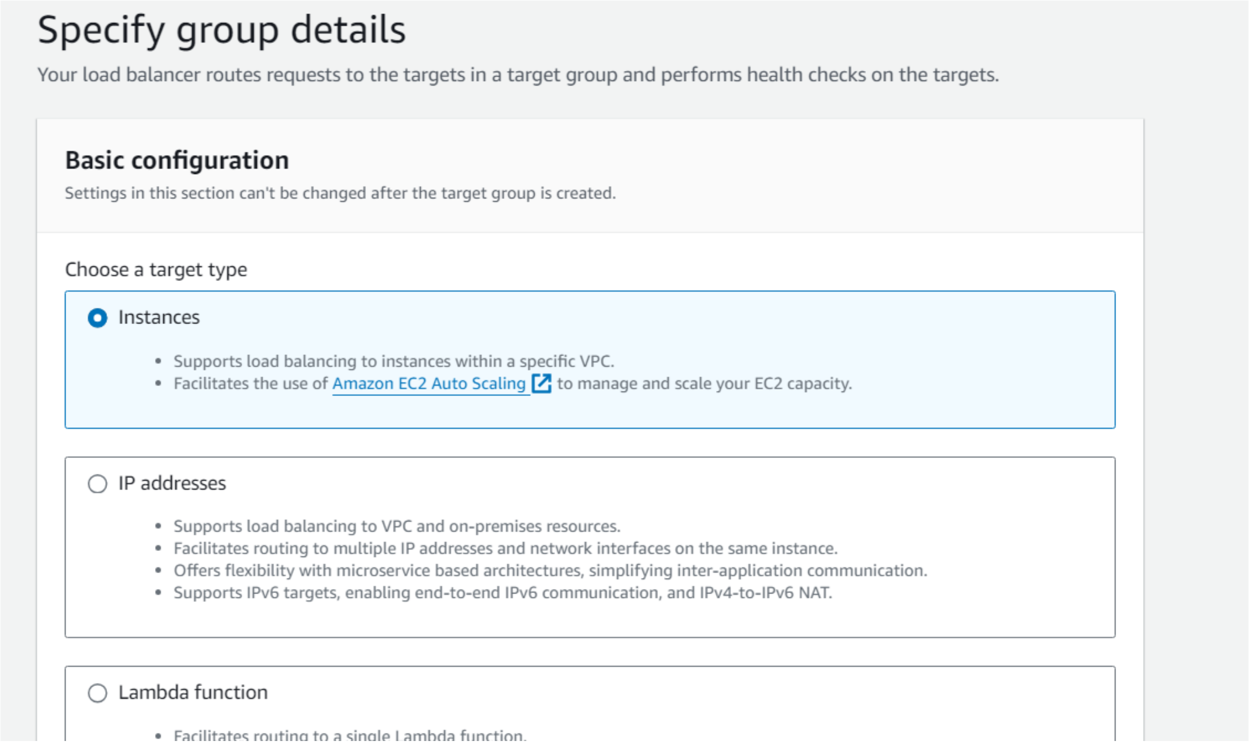
Security groups

Select up to 5 security groups

ALB-SG
sg-06ecc0462e564f35b VPC: vpc-0b579b0189b65a94d

default
sg-09d2264ad27851504 VPC: vpc-0b579b0189b65a94d

Create Target group



Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

● Instances

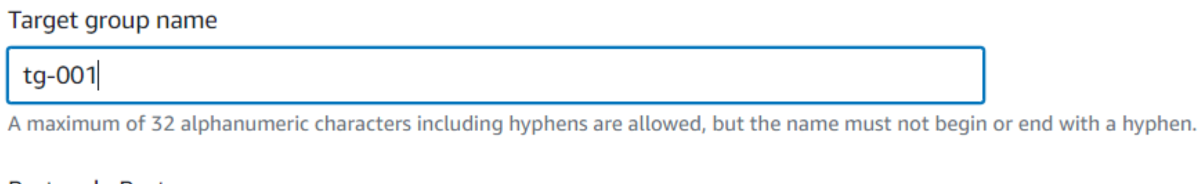
- Supports load balancing to instances within a specific VPC.
- Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.

○ IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

○ Lambda function

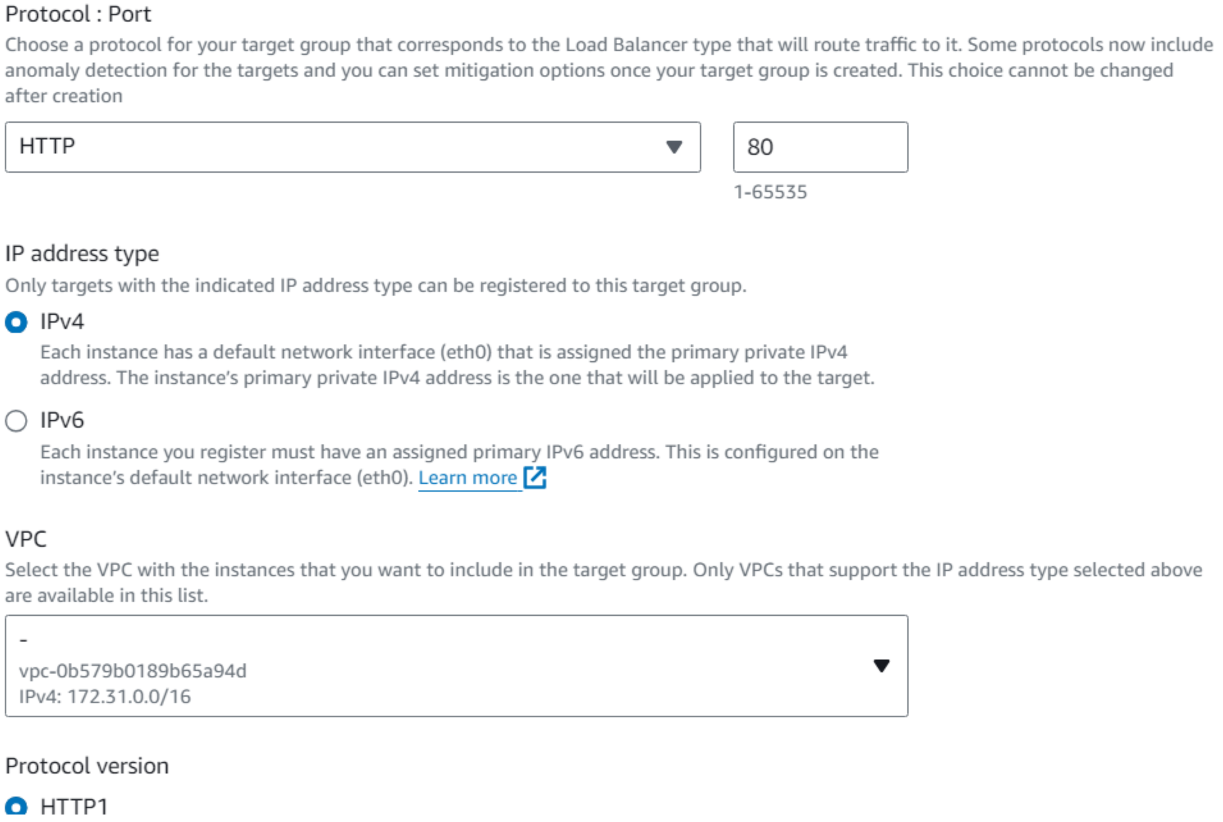
- Facilitates routing to a single Lambda function.



Target group name

tg-001

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 can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP ▼ 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

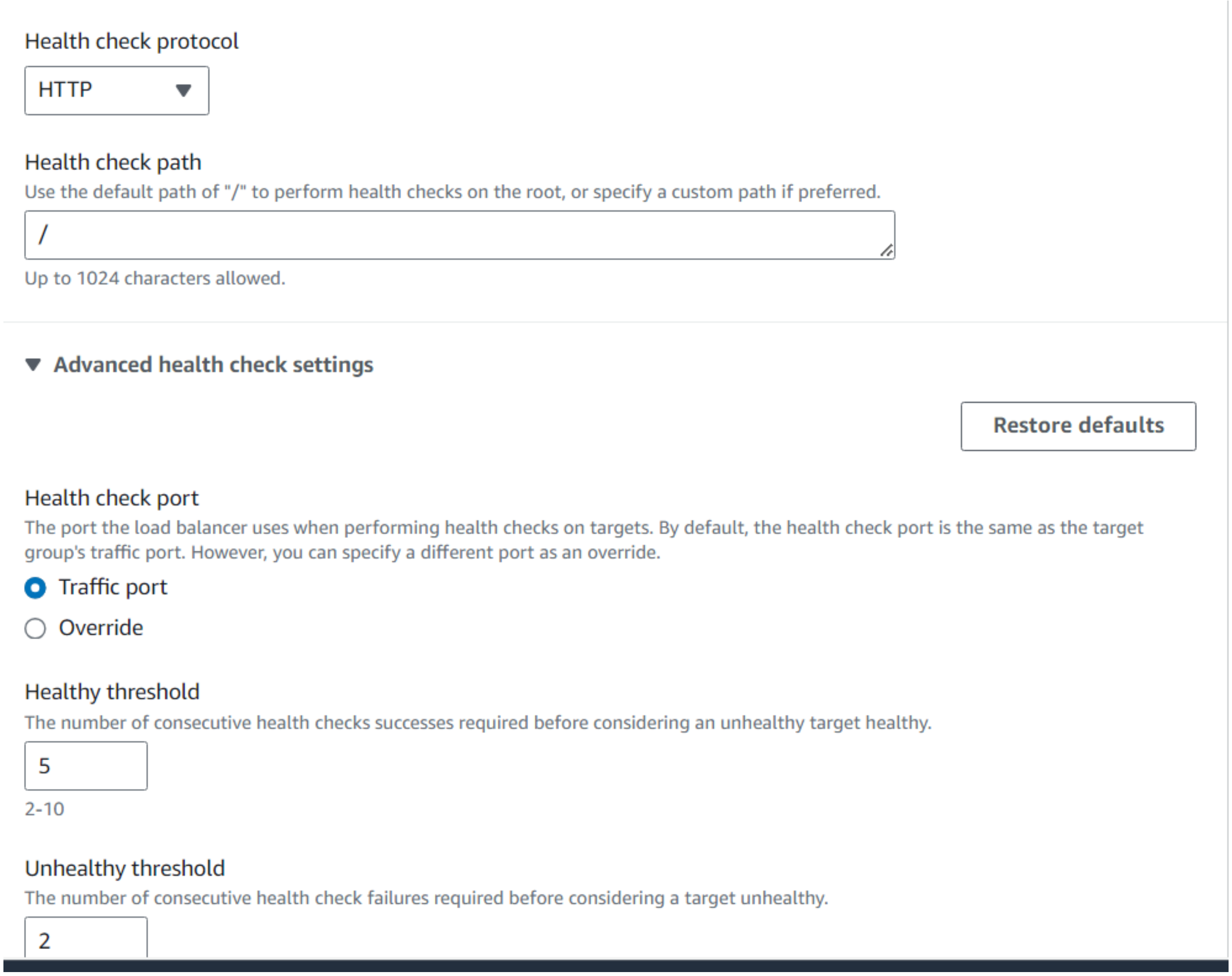
VPC

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-0b579b0189b65a94d
IPv4: 172.31.0.0/16 ▼

Protocol version

● HTTP1



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**

Health check port

The port the load balancer uses when performing health checks on targets. By default, the health check port is the same as the target group's traffic port. However, you can specify a different port as an override.

● Traffic port

○ Override

Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

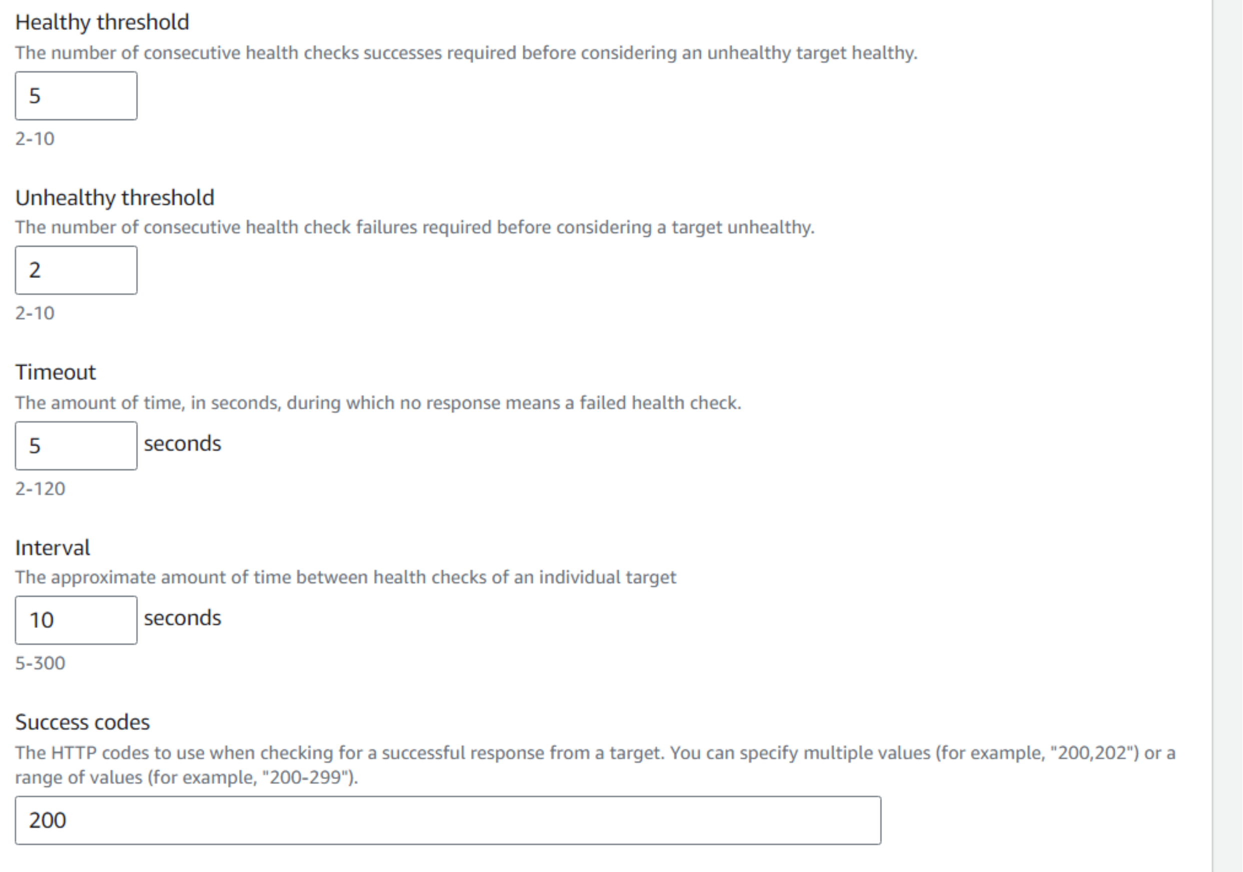
5

2-10

Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

2



Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

5

2-10

Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

2

2-10

Timeout

The amount of time, in seconds, during which no response means a failed health check.

5 seconds

2-120

Interval

The approximate amount of time between health checks of an individual target

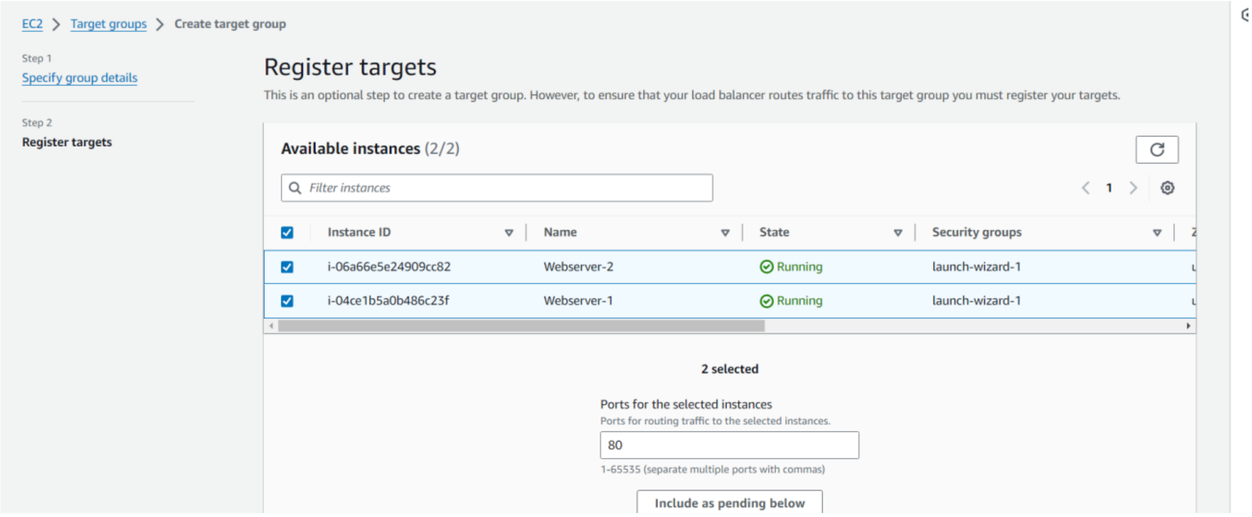
10 seconds

5-300

Success codes

The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

200



EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/2)

Filter instances

☑	Instance ID	Name	State	Security groups
☑	i-06a66e5e24909cc82	Webserver-2	✓ Running	launch-wizard-1
☑	i-04ce1b5a0b486c23f	Webserver-1	✓ Running	launch-wizard-1

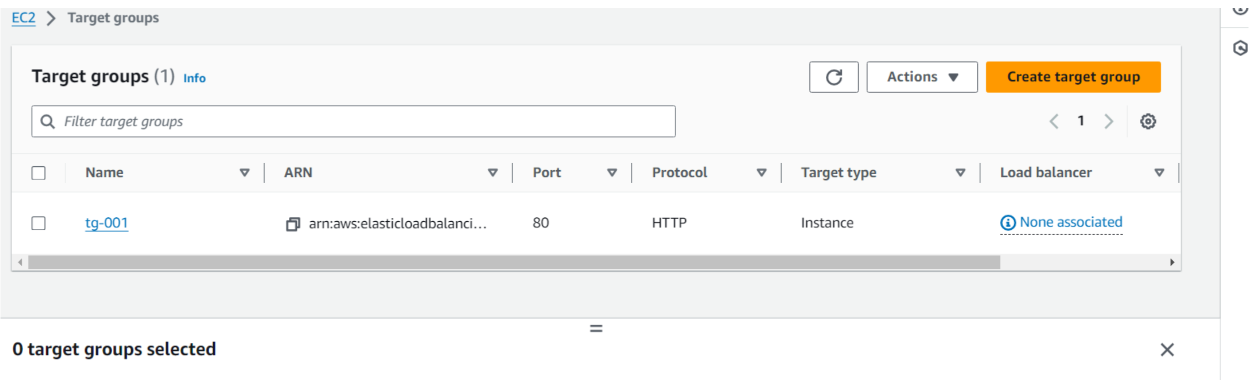
2 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

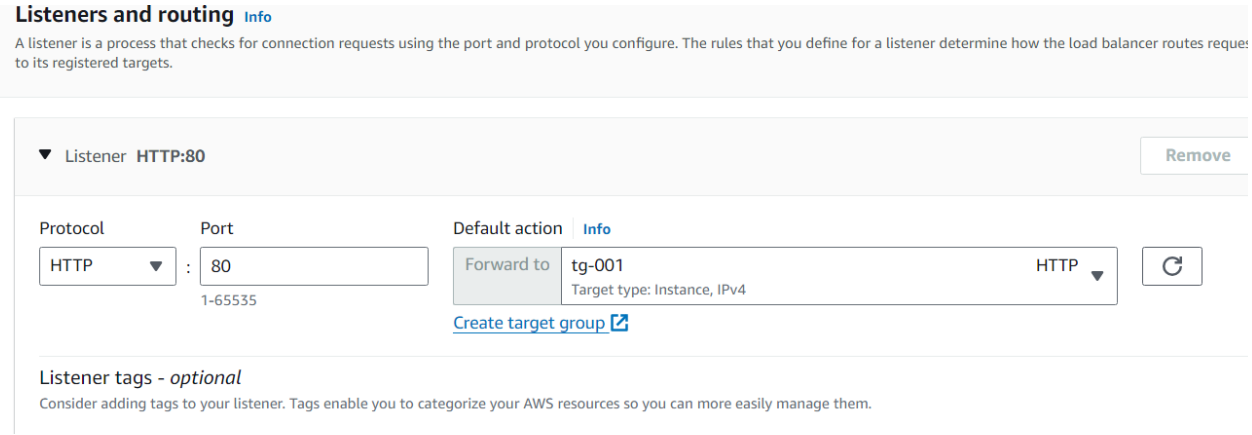


EC2 > Target groups

Target groups (1) Info

□	Name	ARN	Port	Protocol	Target type	Load balancer
□	tg-001	arn:aws:elasticloadbalanci…	80	HTTP	Instance	None associated

0 target groups selected



Listeners and routing Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes reques to its registered targets.

▼ Listener **HTTP:80**

Protocol: HTTP

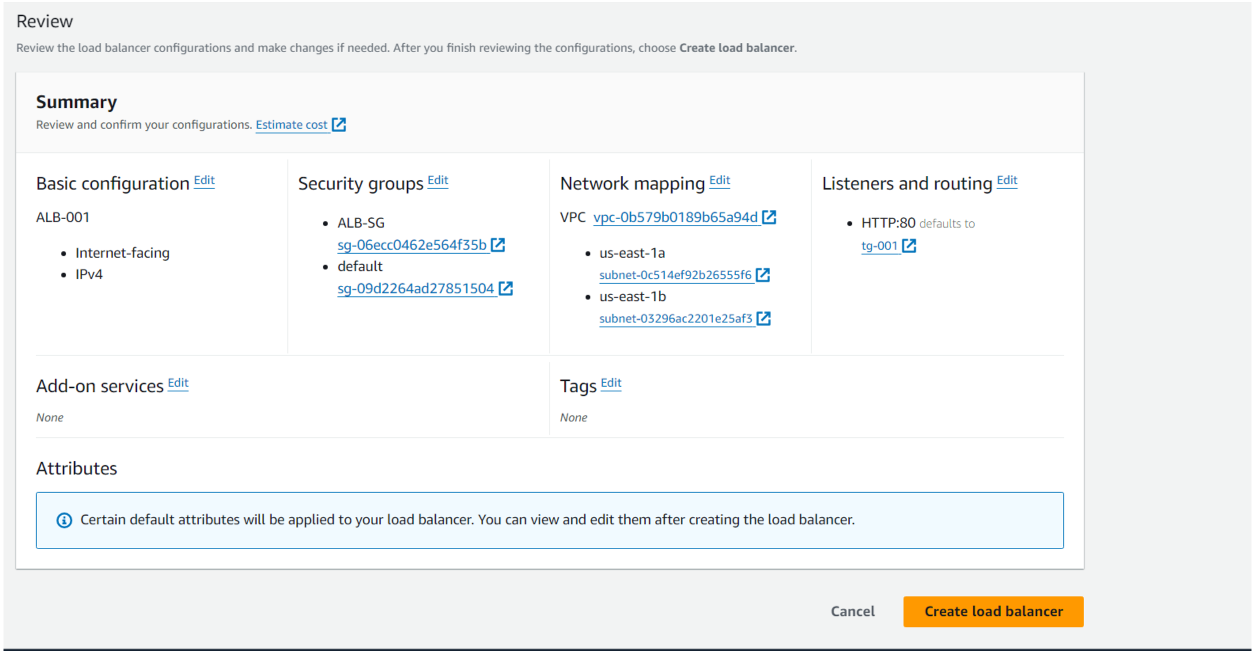
Port: 80
1-65535

Default action Info
Forward to tg-001 HTTP
Target type: Instance, IPv4

Create target group

Listener tags - *optional*

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



Review

Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose **Create load balancer**.

Summary

Review and confirm your configurations. Estimate cost

Basic configuration Edit

ALB-001

- Internet-facing
- IPv4

Security groups Edit

- ALB-SG
  sg-06ecc0462e564f35b
- default
  sg-09d2264ad27851504

Network mapping Edit

VPC vpc-0b579b0189b65a94d

- us-east-1a
  subnet-0c514ef92b26555f6
- us-east-1b
  subnet-03296ac2201e25af3

Listeners and routing Edit

- HTTP:80 defaults to
  tg-001

Add-on services Edit

None

Tags Edit

None

Attributes

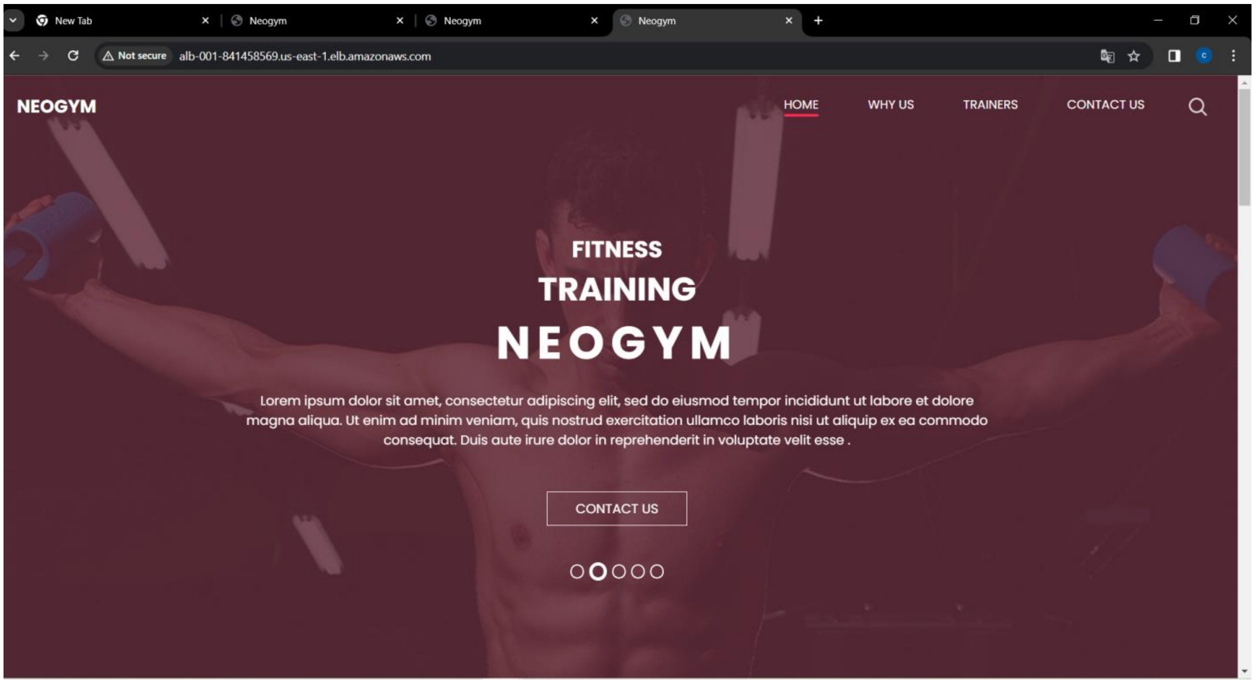
ⓘ Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

After creating load balancer wail till state is active

Once it is active

Copy ALB DNS name

Browse

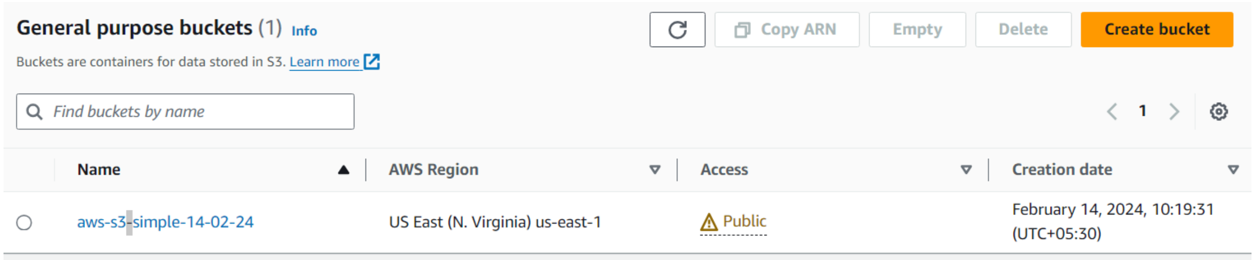


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse .

Auto Scaling group

To create Auto scaling group we have to create launch template

Here we stored application data in s3 bucket

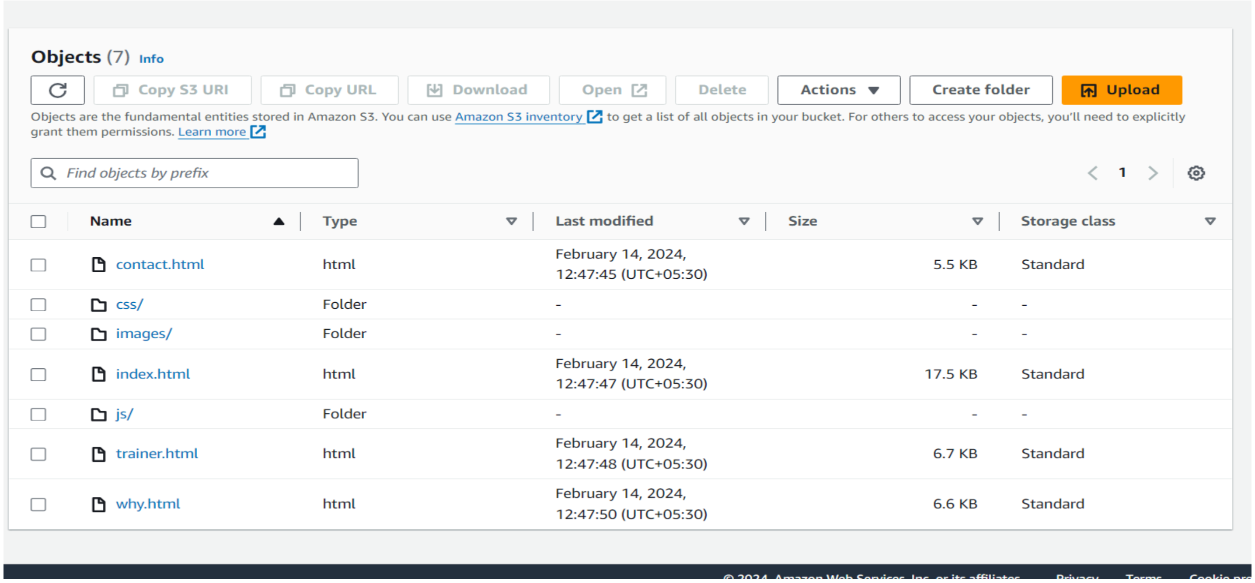


General purpose buckets (1) Info

Buckets are containers for data stored in S3. Learn more

Find buckets by name

	Name ▲	AWS Region ▽	Access ▽	Creation date ▽
○	aws-s3-simple-14-02-24	US East (N. Virginia) us-east-1	⚠ Public	February 14, 2024, 10:19:31 (UTC+05:30)



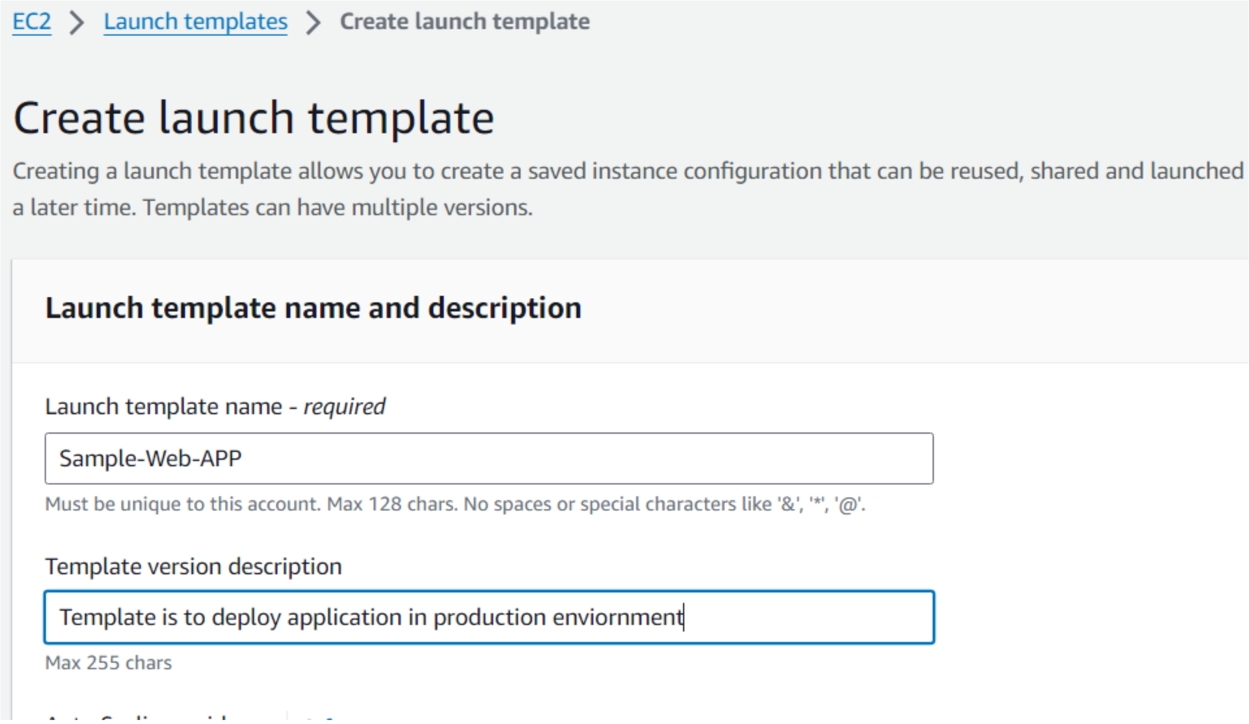
Objects (7) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Find objects by prefix

□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	contact.html	html	February 14, 2024, 12:47:45 (UTC+05:30)	5.5 KB	Standard
□	css/	Folder	-	-	-
□	images/	Folder	-	-	-
□	index.html	html	February 14, 2024, 12:47:47 (UTC+05:30)	17.5 KB	Standard
□	js/	Folder	-	-	-
□	trainer.html	html	February 14, 2024, 12:47:48 (UTC+05:30)	6.7 KB	Standard
□	why.html	html	February 14, 2024, 12:47:50 (UTC+05:30)	6.6 KB	Standard

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 a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

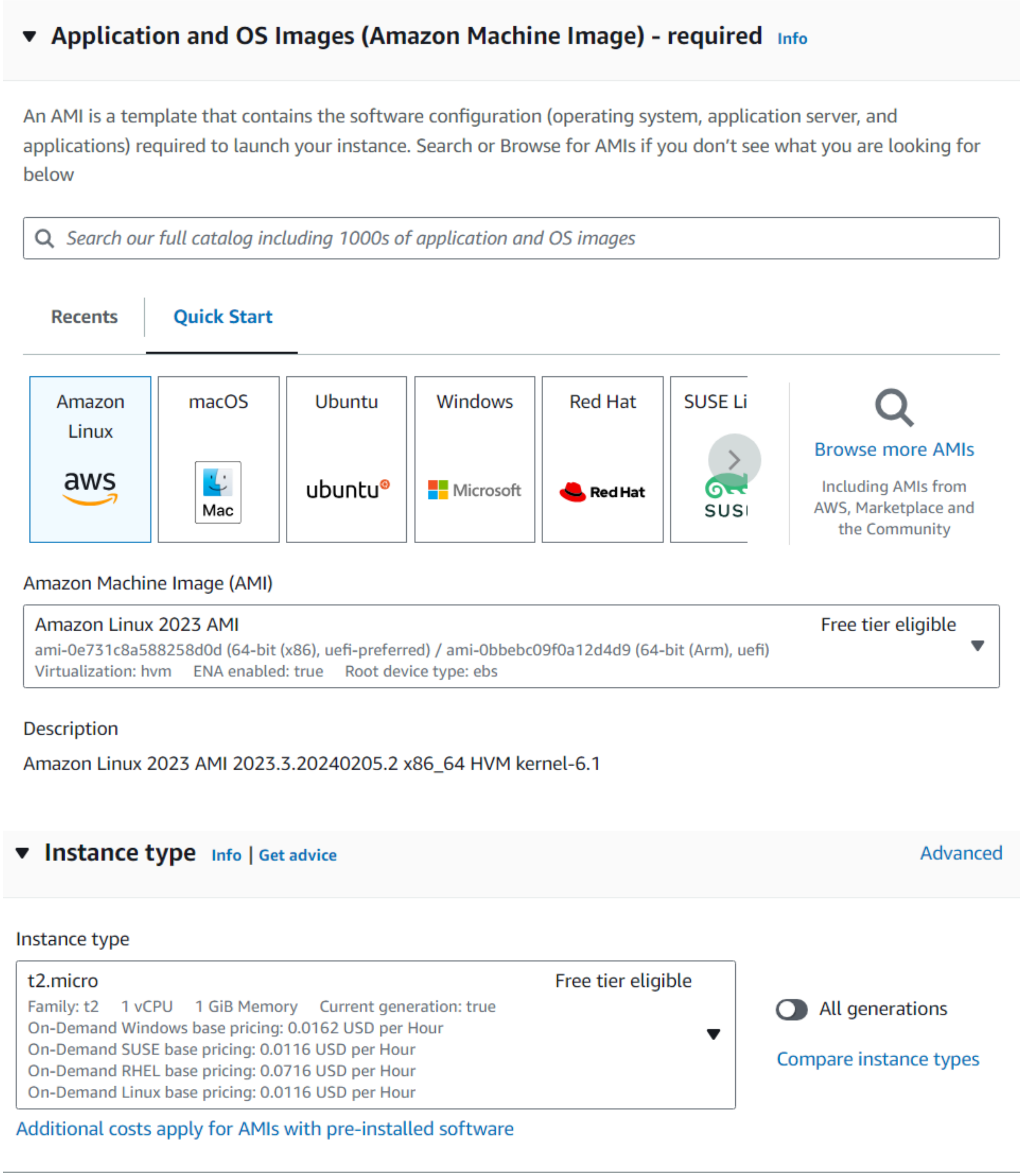
Sample-Web-APP

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

Template is to deploy application in production enviornment

Max 255 chars



▼ 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

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-0e731c8a588258d0d (64-bit (x86), uefi-preferred) / ami-0bbebc09f0a12d4d9 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

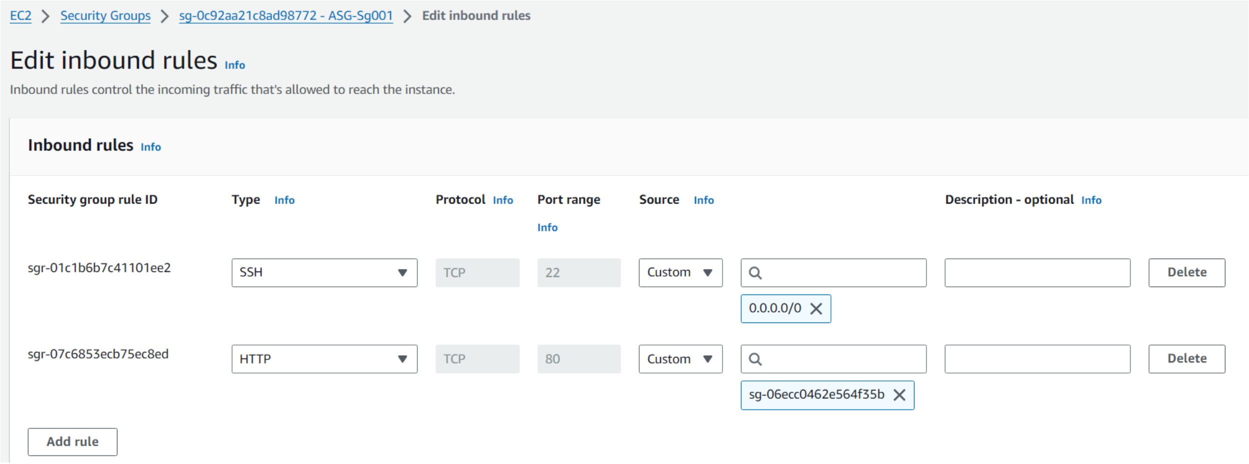
Amazon Linux 2023 AMI 2023.3.20240205.2 x86_64 HVM kernel-6.1

▼ Instance type Info | Get advice — Advanced

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Additional costs apply for AMIs with pre-installed software



EC2 > Security Groups > sg-0c92aa21c8ad98772 - ASG-Sg001 > Edit inbound rules

Edit inbound rules Info

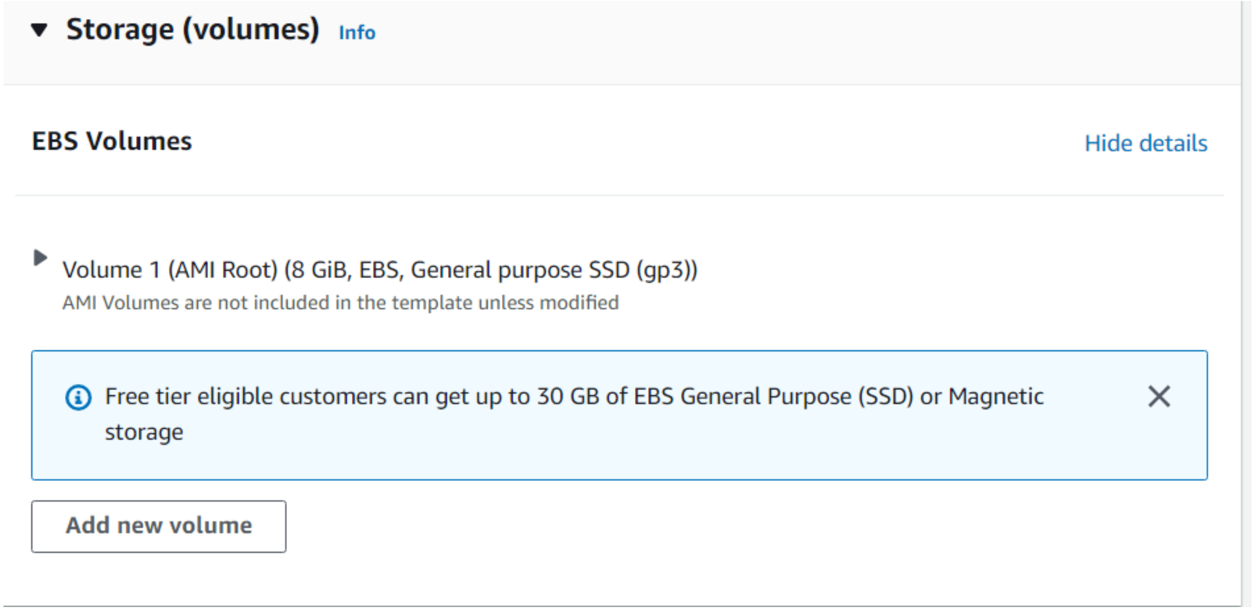
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-01c1b6b7c41101ee2	SSH ▼	TCP	22	Custom ▼	0.0.0.0/0 ✕		Delete
sgr-07c6853ecb75ec8ed	HTTP ▼	TCP	80	Custom ▼	sg-06ecc0462e564f35b ✕		Delete

- **sg-06ecc0462e564f35b** is a security group for load balancer which allows traffic from internet.

  Here we allow traffic from load balancer to our EC2 instances which are created from auto scaling group.



▼ Storage (volumes) Info

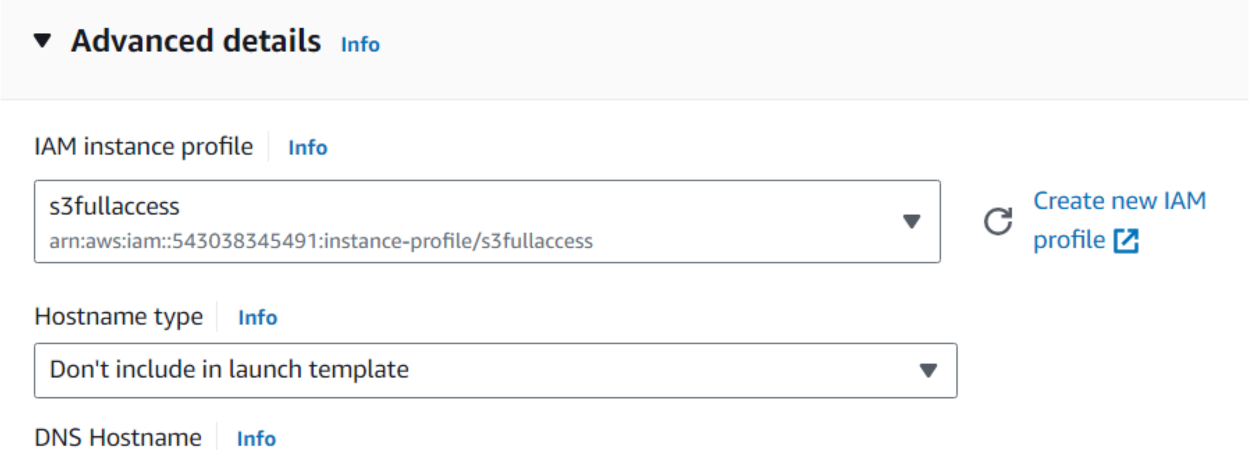
EBS Volumes

Hide details

▶ Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3))

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



Advanced details

IAM instance profile

s3fullaccess
arn:aws:iam::543038345491:instance-profile/s3fullaccess

Hostname type

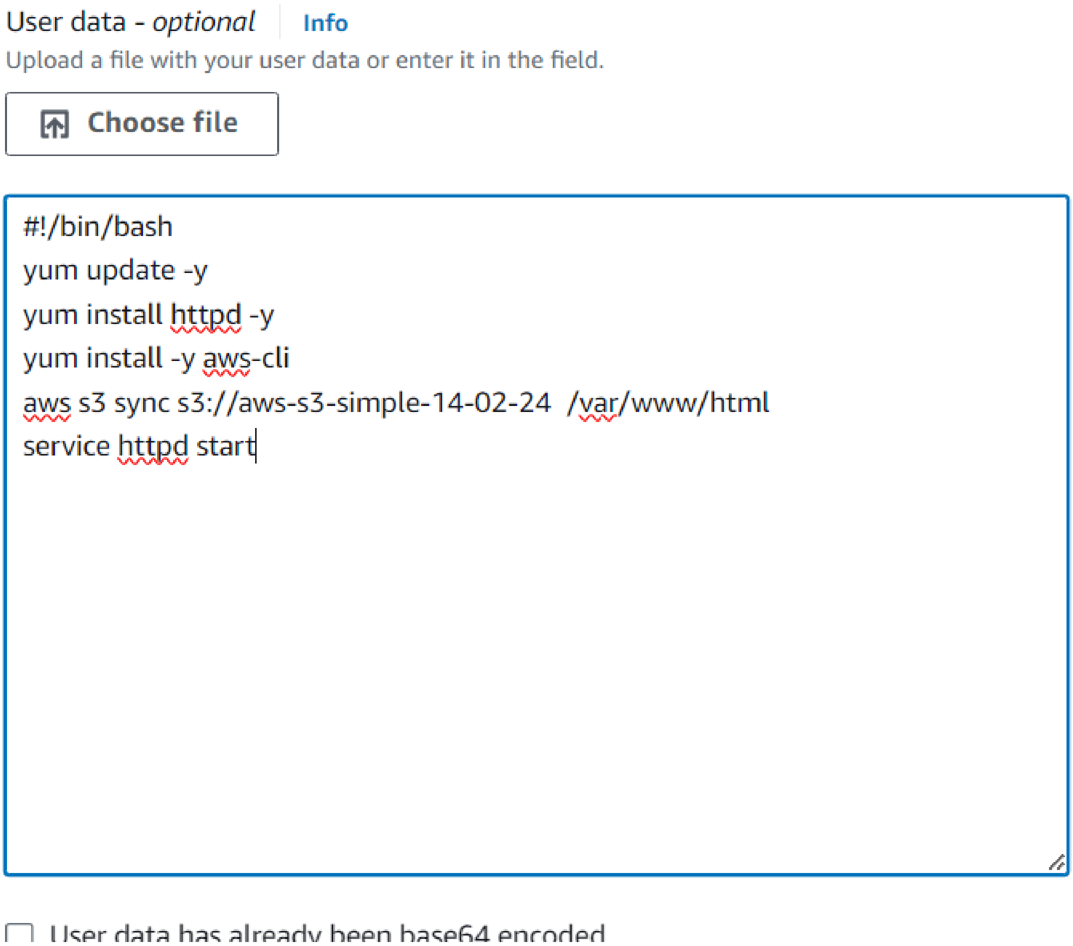
Don't include in launch template

DNS Hostname

This roles gives permission to ec2 instance to communicate with s3 & access data from s3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "s3-object-lambda:*"
            ],
            "Resource": "*"
        }
    ]
}
```

We Have to add user data bash script

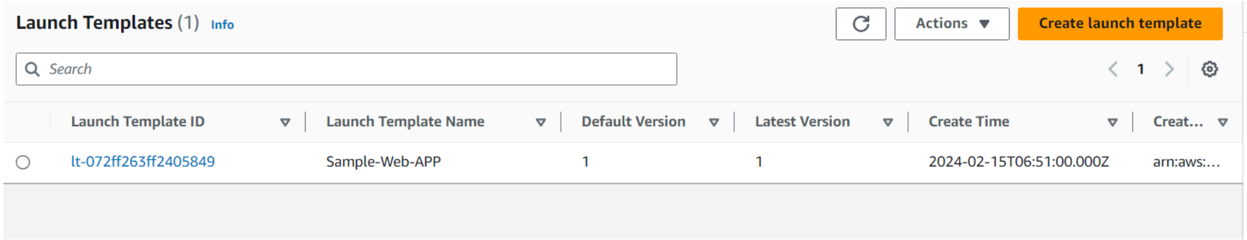


Upload a file with your user data or enter it in the field.

```
#!/bin/bash
yum update -y
yum install httpd -y
yum install -y aws-cli
aws s3 sync s3://aws-s3-simple-14-02-24  /var/www/html
service httpd start
```

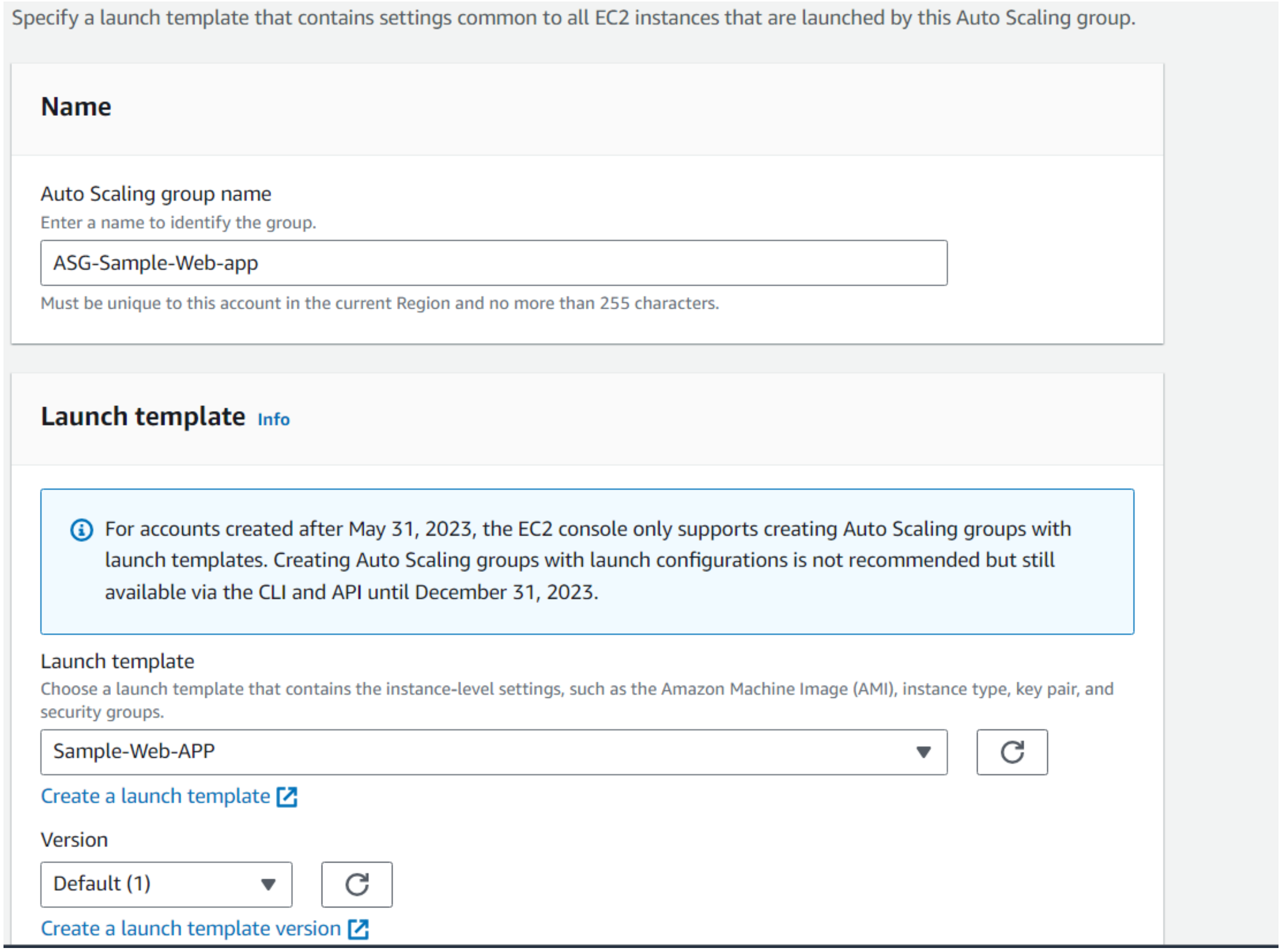
□ User data has already been base64 encoded

```
#!/bin/bash

yum update -y

yum install httpd -y

yum install -y aws-cli

aws s3 sync s3://aws-s3-simple-14-02-24  /var/www/html

service httpd start
```



Launch Templates (1) Info

	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Creat...
○	lt-072ff263ff2405849	Sample-Web-APP	1	1	2024-02-15T06:51:00.000Z	arn:aws:...



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.

ASG-Sample-Web-app

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.

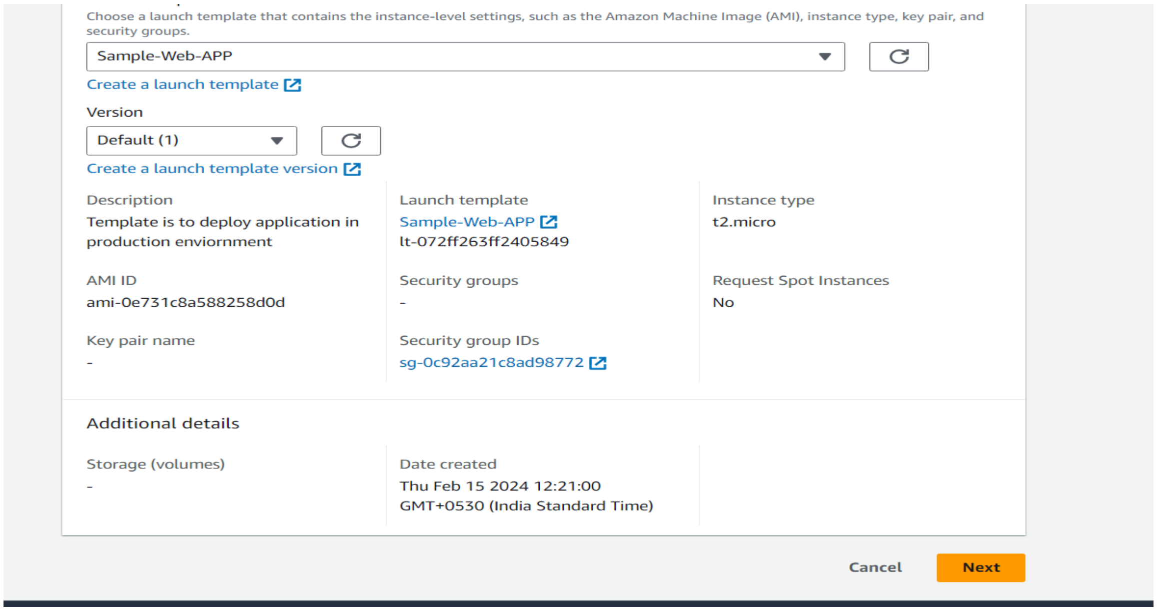
Sample-Web-APP ▼

Create a launch template

Version

Default (1) ▼

Create a launch template version



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

Sample-Web-APP

Create a launch template

Version

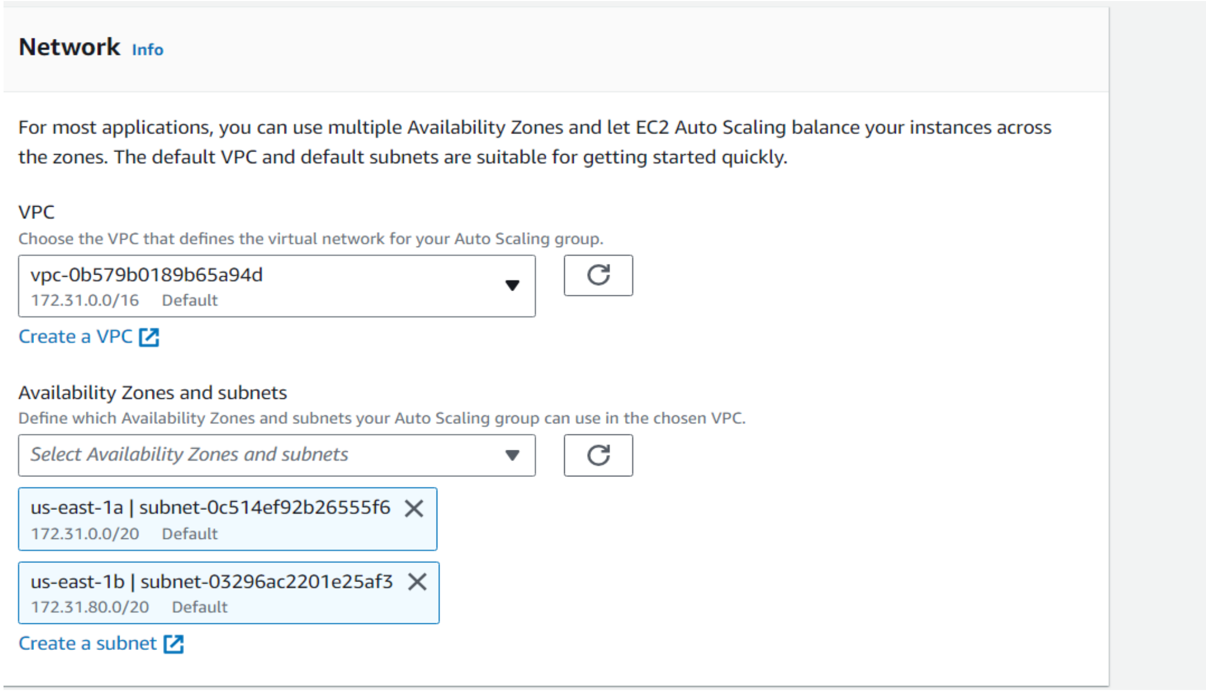
Default (1)

Create a launch template version

Description Template is to deploy application in production enviornment	Launch template Sample-Web-APP lt-072ff263ff2405849	Instance type t2.micro
AMI ID ami-0e731c8a588258d0d	Security groups -	Request Spot Instances No
Key pair name -	Security group IDs sg-0c92aa21c8ad98772	

Additional details

Storage (volumes) -	Date created Thu Feb 15 2024 12:21:00 GMT+0530 (India Standard Time)



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-0b579b0189b65a94d
172.31.0.0/16 Default

Create a VPC

Availability Zones and subnets

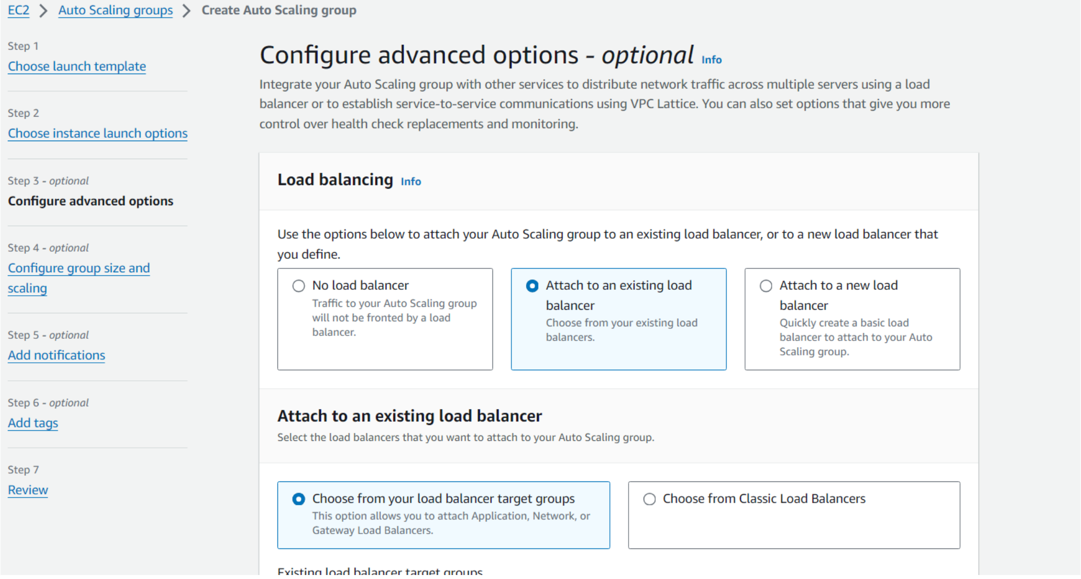
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-east-1a | subnet-0c514ef92b26555f6
172.31.0.0/20 Default

us-east-1b | subnet-03296ac2201e25af3
172.31.80.0/20 Default

Create a subnet



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
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

Configure advanced options - *optional* Info

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over health check replacements and monitoring.

Load balancing Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

○ No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

● Attach to an existing load balancer
Choose from your existing load balancers.

○ Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

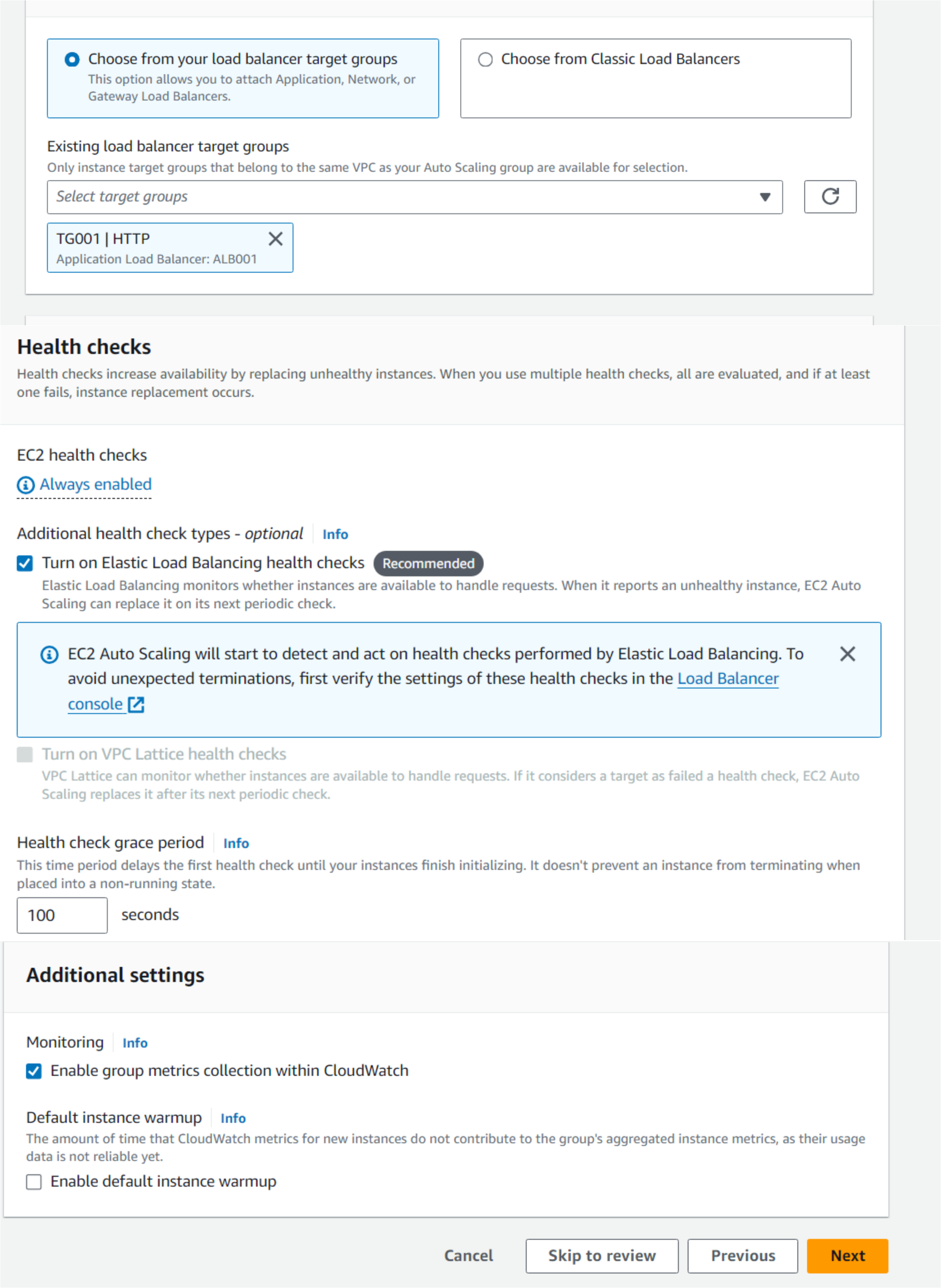
Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

● Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

○ Choose from Classic Load Balancers

Existing load balancer target groups



● Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

○ Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups ▼

TG001 | HTTP ✕
Application Load Balancer: ALB001

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks
ⓘ Always enabled

Additional health check types - *optional* Info

☑ Turn on Elastic Load Balancing health checks Recommended
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.

ⓘ EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing. To avoid unexpected terminations, first verify the settings of these health checks in the Load Balancer console ✕

□ 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.

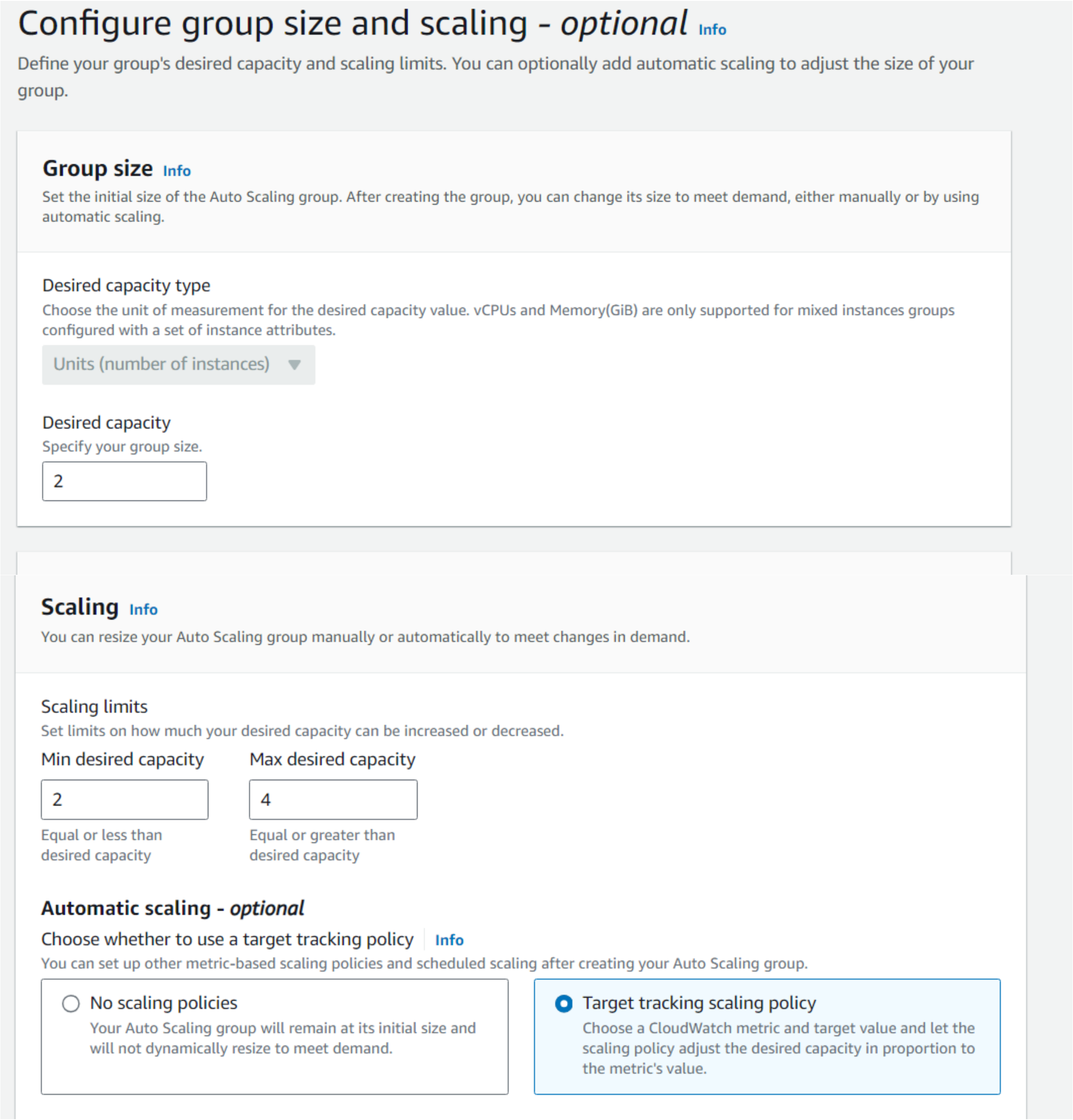
Health check grace period Info
This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

100 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



Configure group size and scaling - *optional* Info

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size Info

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances) ▼

Desired capacity

Specify your group size.

2

Scaling Info

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity	Max desired capacity
2	4
Equal or less than desired capacity	Equal or greater than desired capacity

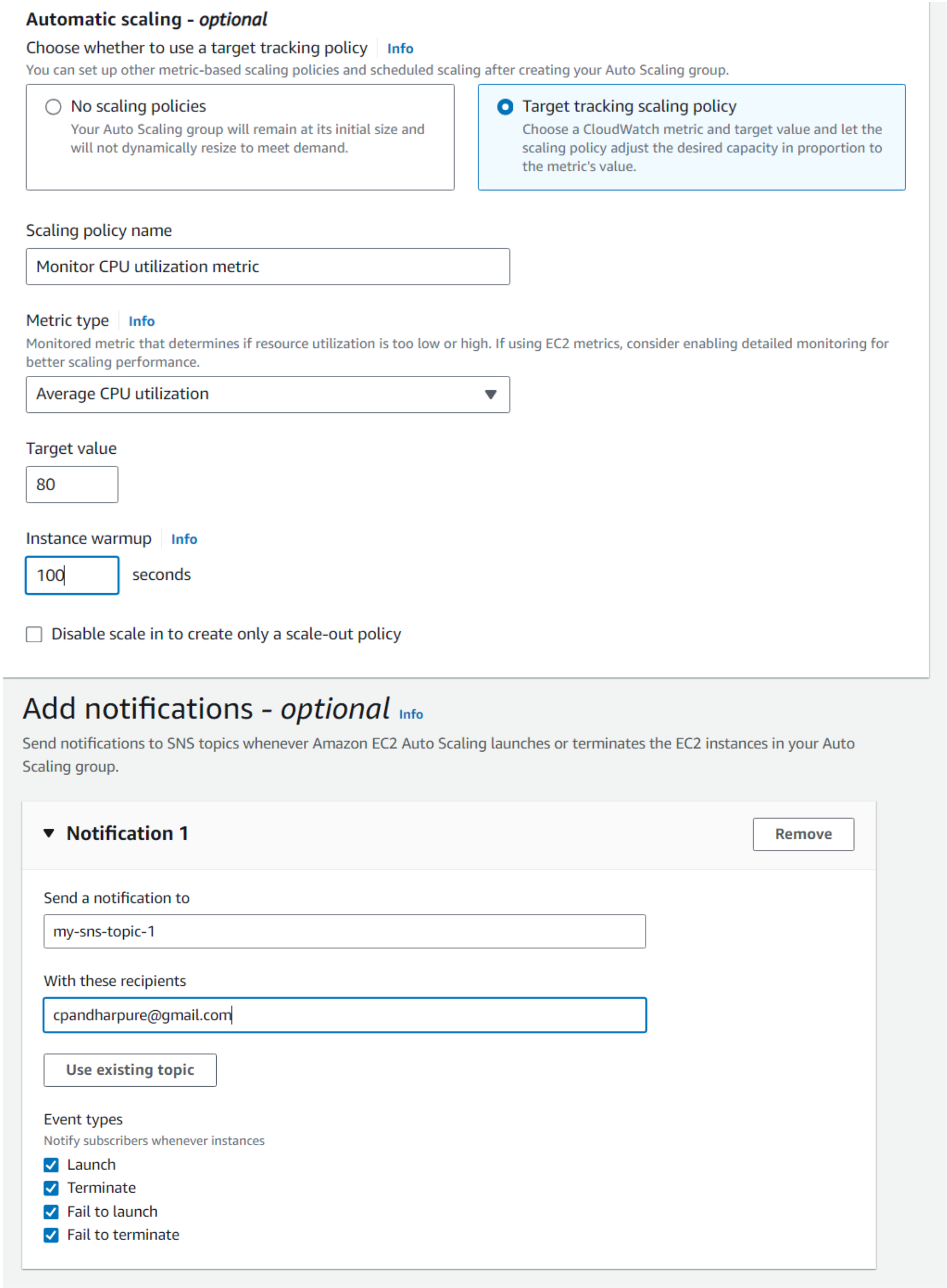
Automatic scaling - *optional*

Choose whether to use a target tracking policy Info

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

○ No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

● Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.



Automatic scaling - *optional*

Choose whether to use a target tracking policy Info

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

○ No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

● Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Scaling policy name

Monitor CPU utilization metric

Metric type Info

Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization ▼

Target value

80

Instance warmup Info

100 seconds

□ Disable scale in to create only a scale-out policy

Add notifications - *optional* Info

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

▼ **Notification 1** Remove

Send a notification to

my-sns-topic-1

With these recipients

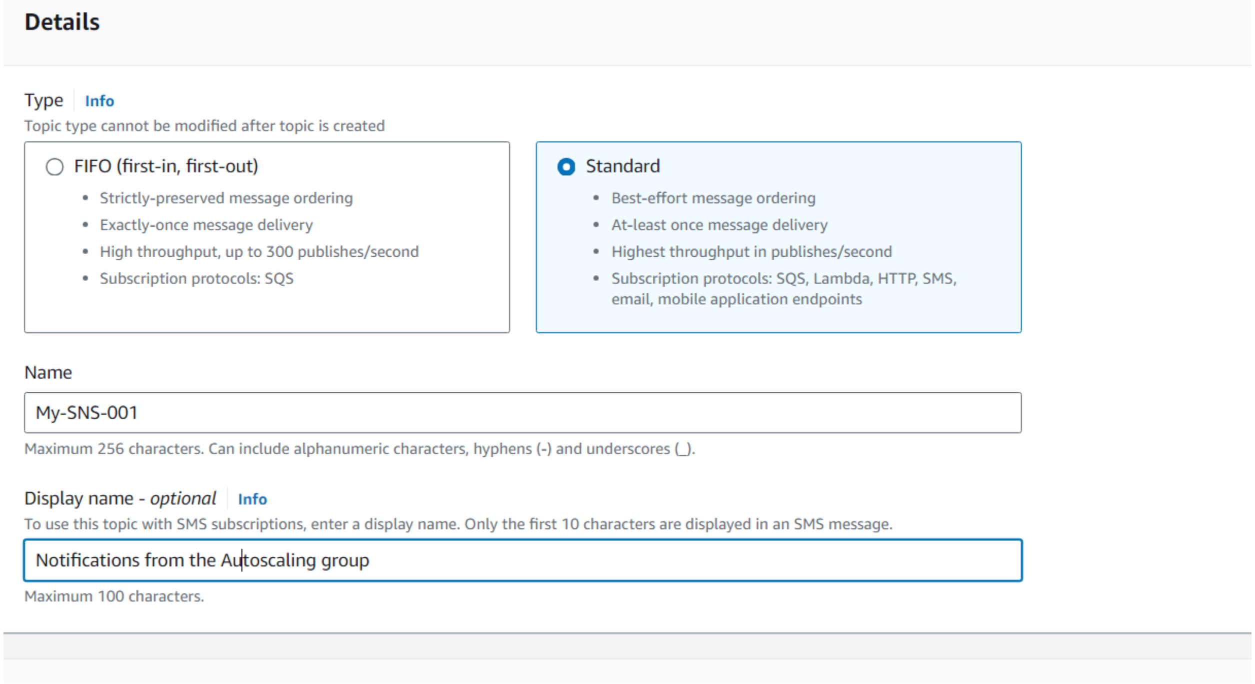
cpandharpure@gmail.com

Use existing topic

Event types

Notify subscribers whenever instances

☑ Launch
☑ Terminate
☑ Fail to launch
☑ Fail to terminate



Details

Type | Info

Topic type cannot be modified after topic is created

○ FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

● Standard

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

My-SNS-001

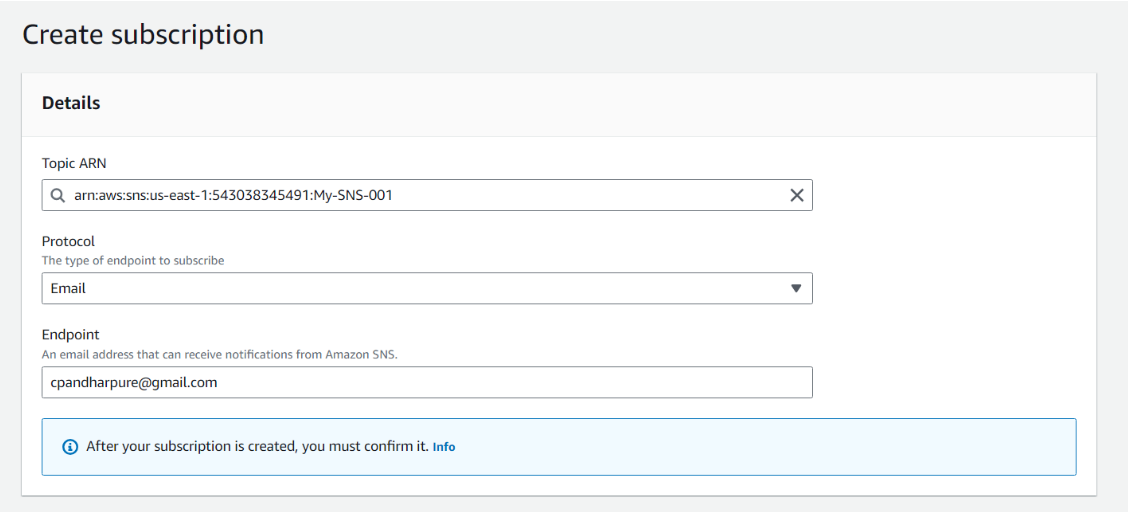
Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - *optional* | Info

To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

Notifications from the Autoscaling group

Maximum 100 characters.



Create subscription

Details

Topic ARN

arn:aws:sns:us-east-1:543038345491:My-SNS-001

Protocol

The type of endpoint to subscribe

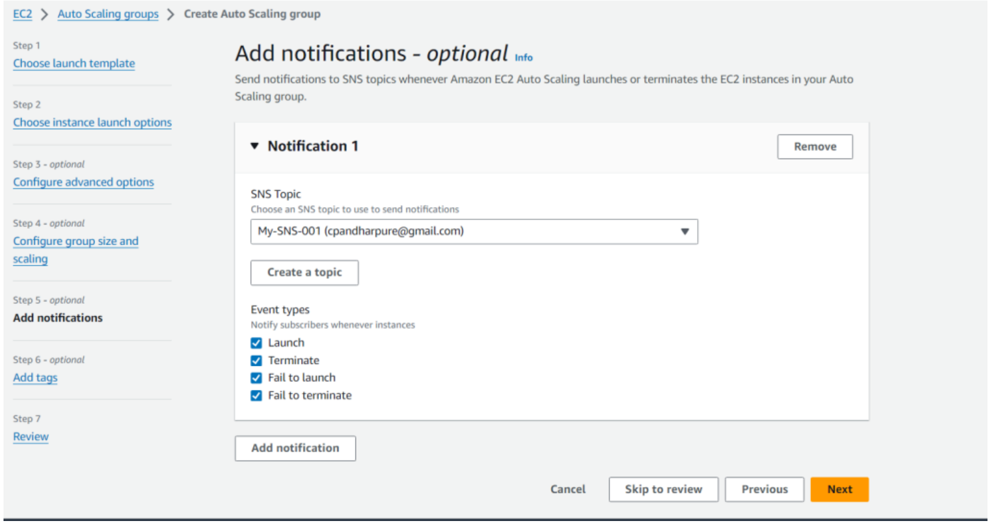
Email

Endpoint

An email address that can receive notifications from Amazon SNS.

cpandharpure@gmail.com

After your subscription is created, you must confirm it. Info



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
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

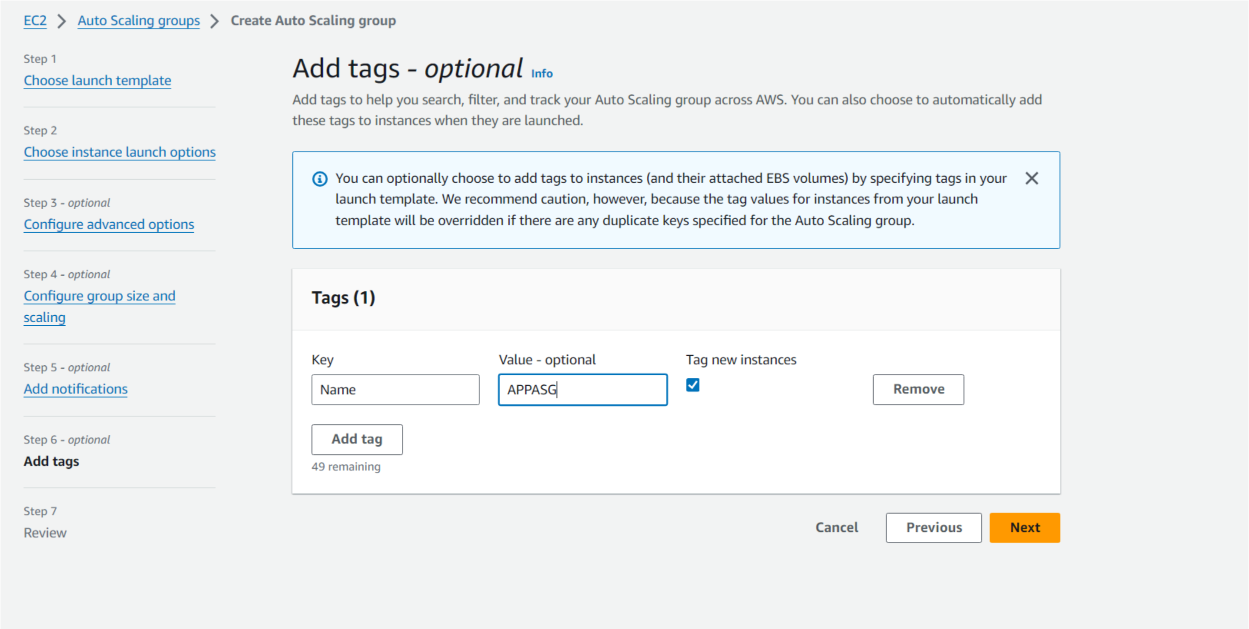
Add notifications - *optional* Info

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

▼ Notification 1

SNS Topic
Choose an SNS topic to use to send notifications
My-SNS-001 (cpandharpure@gmail.com)

Event types
Notify subscribers whenever instances
☑ Launch
☑ Terminate
☑ Fail to launch
☑ Fail to terminate



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
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

Add tags - *optional* Info

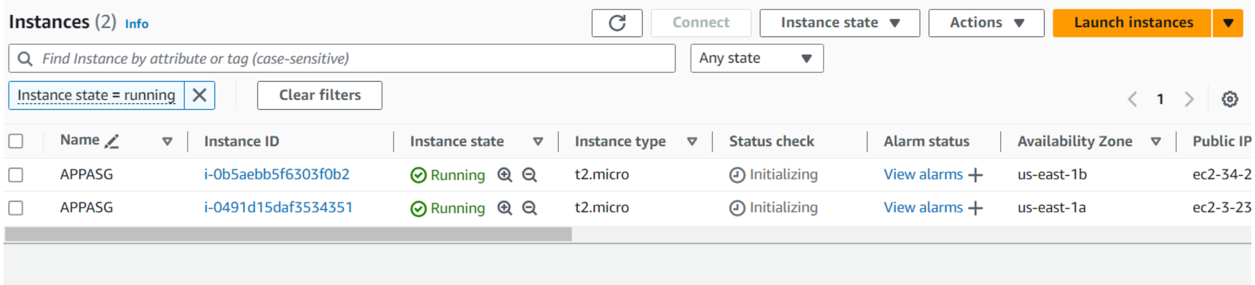
Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.

Tags (1)

Key	Value - optional	Tag new instances
Name	APPASG	☑

49 remaining



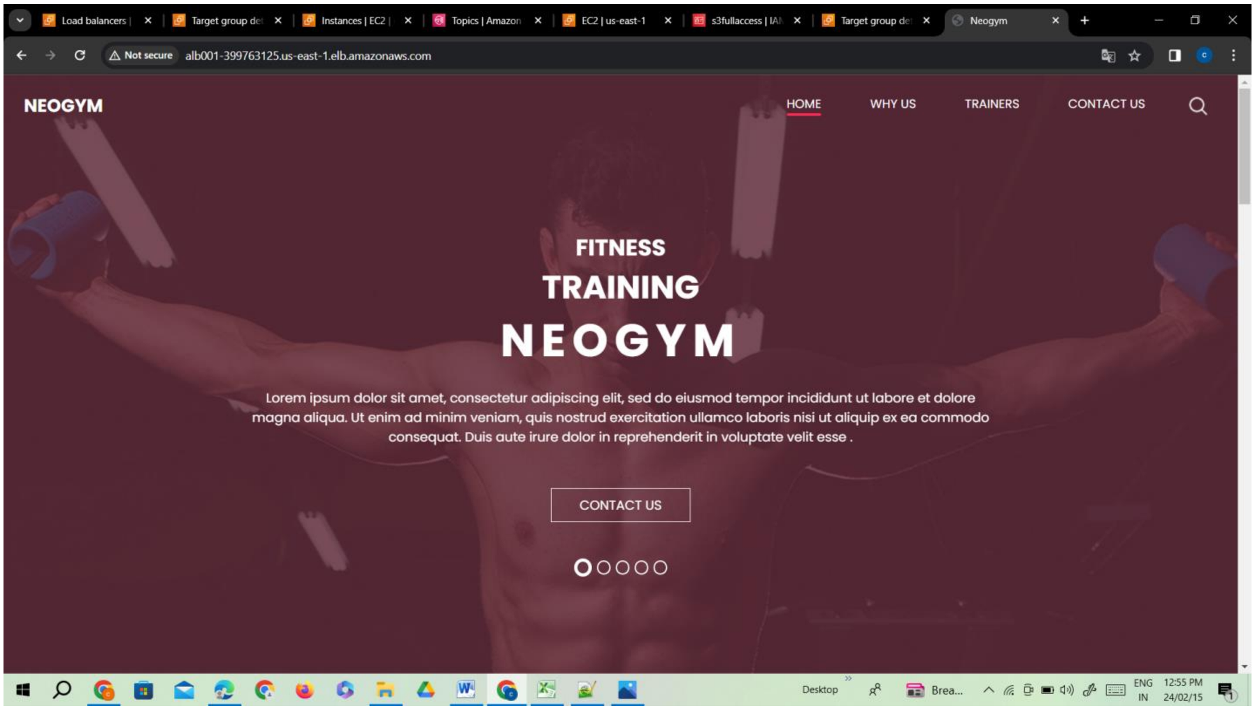
Instances (2) Info

Find Instance by attribute or tag (case-sensitive)

Any state

Instance state = running

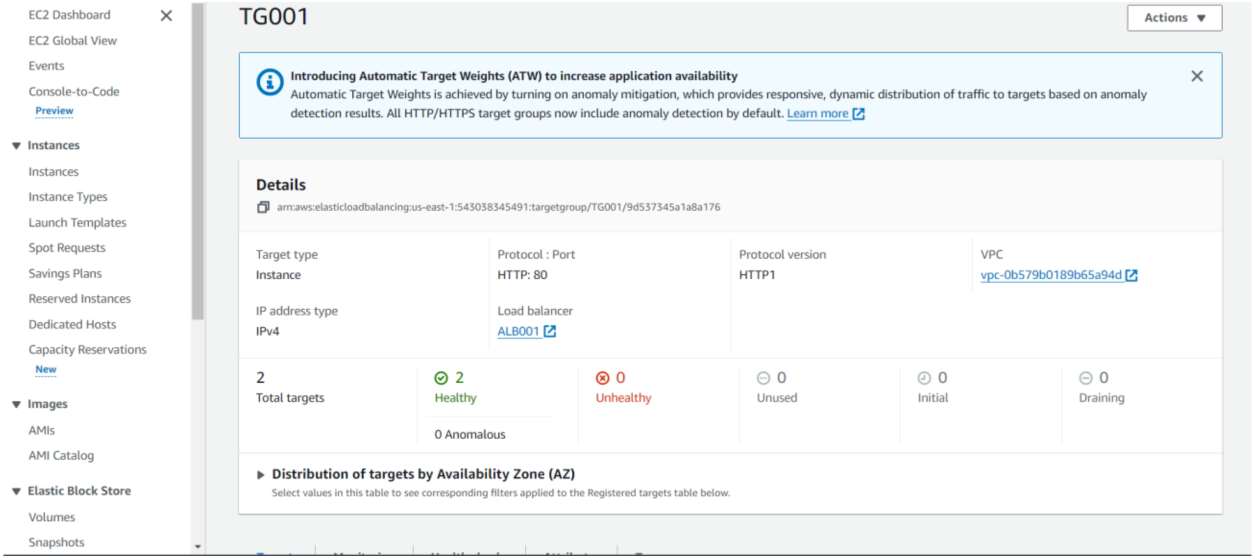
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
APPASG	i-0b5aebb5f6303f0b2	✓ Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-34-2
APPASG	i-0491d15daf3534351	✓ Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-3-23



FITNESS
TRAINING
NEOGYM

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse .

0000



TG001

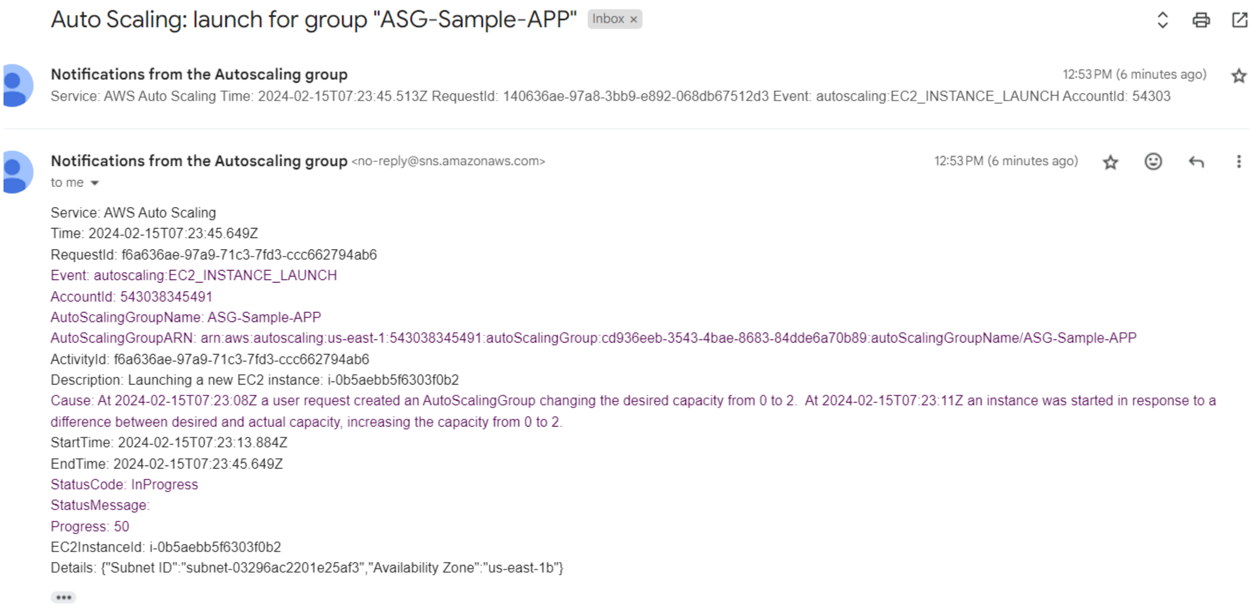
Introducing Automatic Target Weights (ATW) to increase application availability
Automatic Target Weights is achieved by turning on anomaly mitigation, which provides responsive, dynamic distribution of traffic to targets based on anomaly detection results. All HTTP/HTTPS target groups now include anomaly detection by default. Learn more

Details
arn:aws:elasticloadbalancing:us-east-1:543038345491:targetgroup/TG001/9d537345a1a8a176

Target type: Instance
Protocol : Port: HTTP: 80
Protocol version: HTTP1
VPC: vpc-0b579b0189b65a94d
IP address type: IPv4
Load balancer: ALB001

2	✓ 2	⊗ 0	0	0	0
Total targets	Healthy 0 Anomalous	Unhealthy	Unused	Initial	Draining

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.



Auto Scaling: launch for group "ASG-Sample-APP" Inbox ×

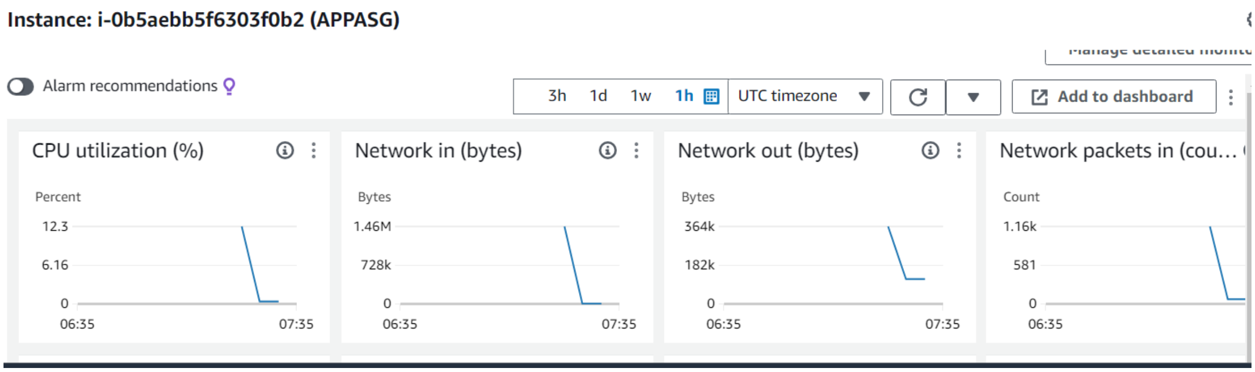
Notifications from the Autoscaling group 12:53 PM (6 minutes ago)

Service: AWS Auto Scaling Time: 2024-02-15T07:23:45.513Z RequestId: 140636ae-97a8-3bb9-e892-068db67512d3 Event: autoscaling:EC2_INSTANCE_LAUNCH AccountId: 54303

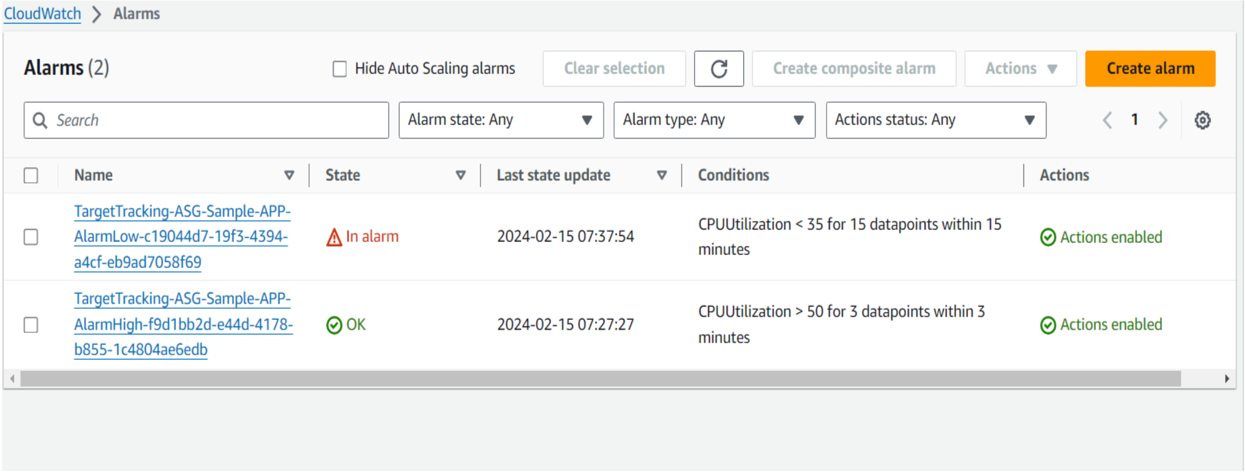
Notifications from the Autoscaling group <no-reply@sns.amazonaws.com> 12:53 PM (6 minutes ago)

to me

```
Service: AWS Auto Scaling
Time: 2024-02-15T07:23:45.649Z
RequestId: f6a636ae-97a9-71c3-7fd3-ccc662794ab6
Event: autoscaling:EC2_INSTANCE_LAUNCH
AccountId: 543038345491
AutoScalingGroupName: ASG-Sample-APP
AutoScalingGroupARN: arn:aws:autoscaling:us-east-1:543038345491:autoScalingGroup:cd936eeb-3543-4bae-8683-84dde6a70b89:autoScalingGroupName/ASG-Sample-APP
ActivityId: f6a636ae-97a9-71c3-7fd3-ccc662794ab6
Description: Launching a new EC2 instance: i-0b5aebb5f6303f0b2
Cause: At 2024-02-15T07:23:08Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2.  At 2024-02-15T07:23:11Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.
StartTime: 2024-02-15T07:23:13.884Z
EndTime: 2024-02-15T07:23:45.649Z
StatusCode: InProgress
StatusMessage:
Progress: 50
EC2InstanceId: i-0b5aebb5f6303f0b2
Details: {"Subnet ID":"subnet-03296ac2201e25af3","Availability Zone":"us-east-1b"}
```



Instance: i-0b5aebb5f6303f0b2 (APPASG)



□	Name	State	Last state update	Conditions	Actions
□	TargetTracking-ASG-Sample-APP-AlarmLow-c19044d7-19f3-4394-a4cf-eb9ad7058f69	⚠ In alarm	2024-02-15 07:37:54	CPUUtilization < 35 for 15 datapoints within 15 minutes	✓ Actions enabled
□	TargetTracking-ASG-Sample-APP-AlarmHigh-f9d1bb2d-e44d-4178-b855-1c4804ae6edb	✓ OK	2024-02-15 07:27:27	CPUUtilization > 50 for 3 datapoints within 3 minutes	✓ Actions enabled