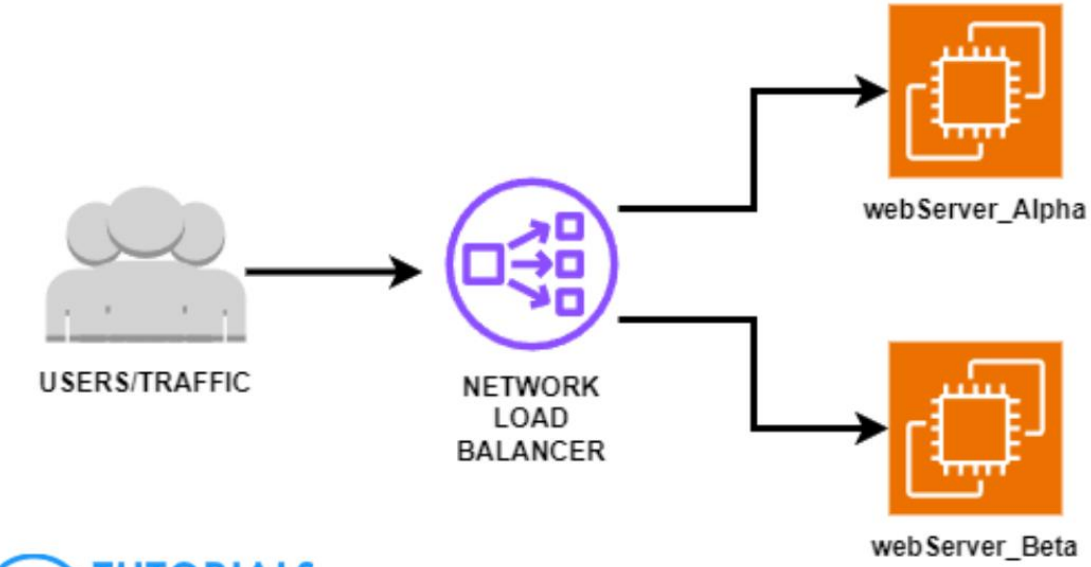
Guided Lab: Creating Your First Network Load Balancer

Description

A Network Load Balancer (NLB) is designed to handle millions of requests per second while maintaining ultra-low latencies, making it ideal for handling volatile traffic patterns. It operates at the connection level (Layer 4), routing connections between clients and targets within Amazon VPC based on IP protocol data. This lab will guide you through the steps to set up your first NLB, helping you understand its functionality and how it can be integrated into your infrastructure for better performance and reliability.



Prerequisites

This lab assumes you have experience creating an Amazon EC2 Instance and its basic fundamentals. If you find any gaps in your knowledge, consider taking the following labs:

- Creating an Amazon EC2 instance (Linux)
- Setting up a Web server on an EC2 instance
- Launching an EC2 Instance with User Data

Objectives

By the end of this guide, you will:

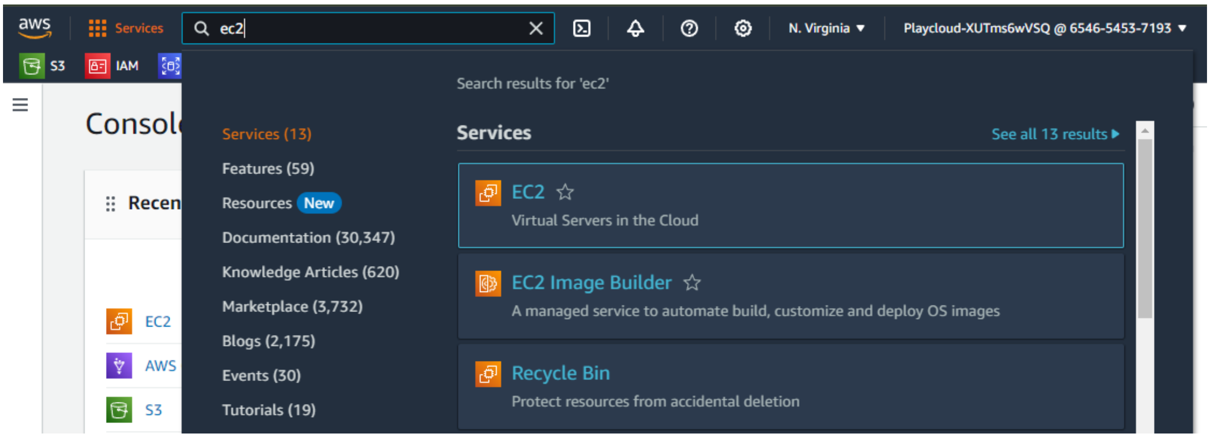
- Understand the fundamentals of AWS Network Load Balancers.
- Successfully create and configure a Network Load Balancer.
- Test the NLB to ensure it properly distributes traffic across multiple backend servers.

[Subscribe to access AWS PlayCloud Labs]

Lab Steps

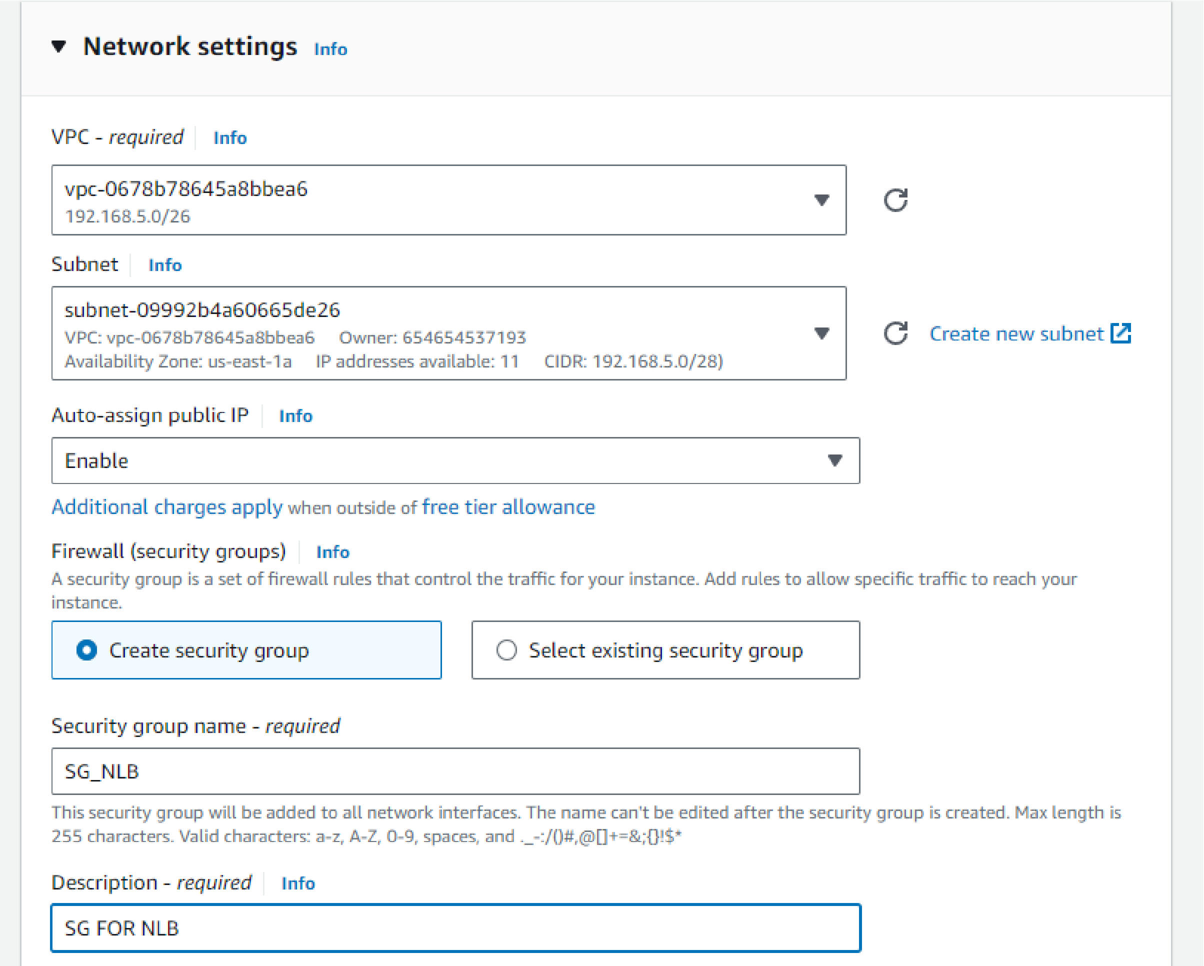
Creating two EC2 Instances

1. **Navigate the EC2 Dashboard.**



2. **Launch the first EC2 Instances using the following configurations:**

- Name: **webServer_Alpha**
- AMI: **Amazon Linux**
- Instance type: **t2.micro**
- Key pair: (**Please create a new one.**)
    - Key pair name: **web-server-key-pair**
    - Key pair type: **RSA**
    - Private key file format: **.pem**
- Network settings: **(**Click **"Edit")**
    - Subnet: (Choose the subnet that is in the AZ: **us-east-1a**)
    - Auto-assign public IP: Select **Enable**
    - Firewall (security groups): tick on the **Create security group**
        - Security group name – required: **SG_NLB**
        - Description – required: **SG FOR NLB**



VPC - *required* Info

vpc-0678b78645a8bbea6
192.168.5.0/26

Subnet Info

subnet-09992b4a60665de26
VPC: vpc-0678b78645a8bbea6 Owner: 654654537193
Availability Zone: us-east-1a IP addresses available: 11 CIDR: 192.168.5.0/28)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

Security group name - *required*

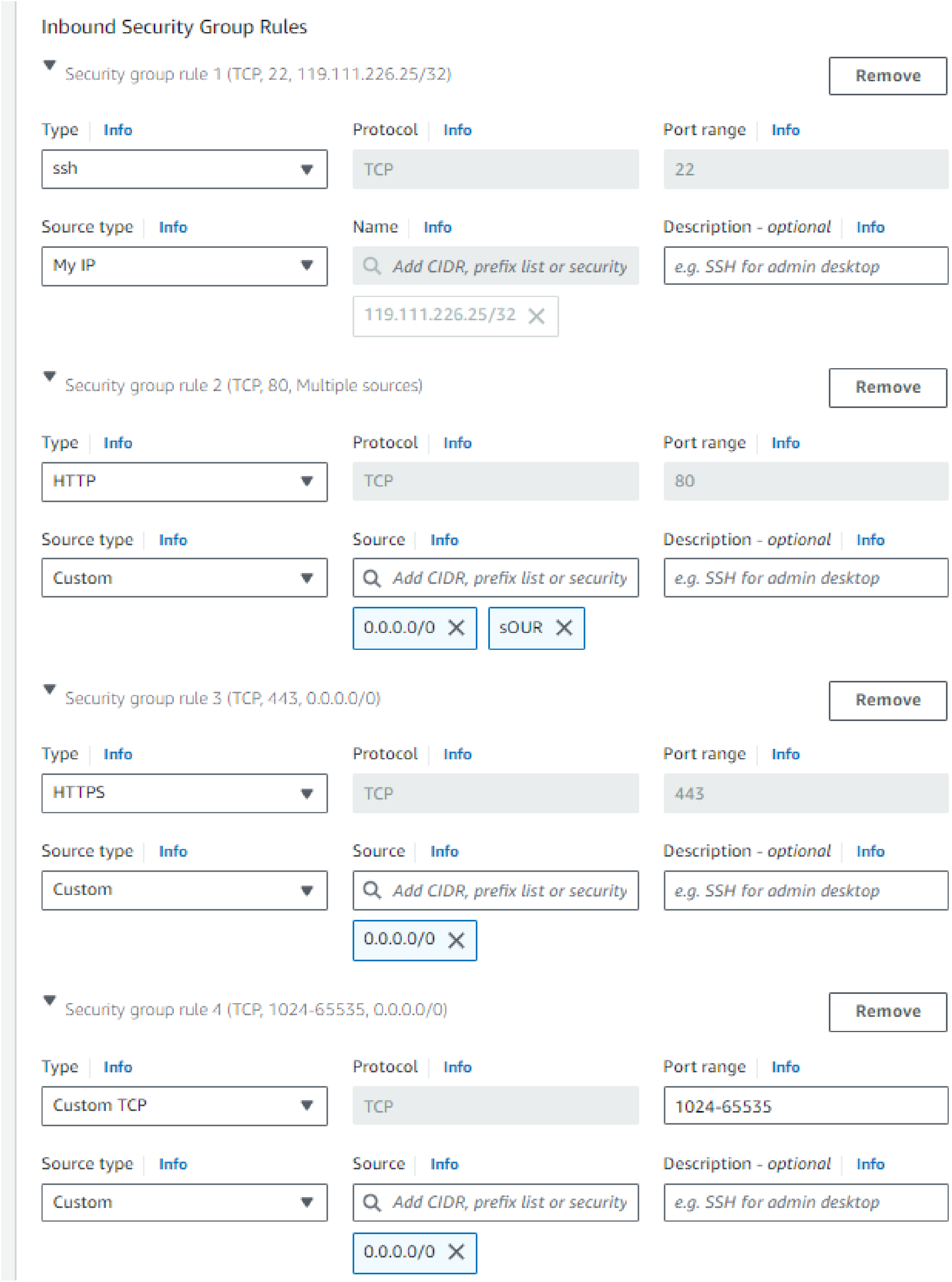
SG_NLB

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info

SG FOR NLB

- Add the following Inbound Security Group Rules:
  - Type: **ssh**
    - Source type: **My IP**
  - Type: **HTTPS**
    - Source type: **Custom**
    - Source: **0.0.0.0/0**
  - Type: **Custom TCP**
    - Port range: **1024-65535**
    - Source type: **Custom**
    - Source: **0.0.0.0/0**



Inbound Security Group Rules

Security group rule 1 (TCP, 22, 119.111.226.25/32)

Type: ssh	Protocol: TCP	Port range: 22
Source type: My IP	Name: Add CIDR, prefix list or security 119.111.226.25/32	Description - *optional*: *e.g. SSH for admin desktop*

Security group rule 2 (TCP, 80, Multiple sources)

Type: HTTP	Protocol: TCP	Port range: 80
Source type: Custom	Source: Add CIDR, prefix list or security 0.0.0.0/0 sOUR	Description - *optional*: *e.g. SSH for admin desktop*

Security group rule 3 (TCP, 443, 0.0.0.0/0)

Type: HTTPS	Protocol: TCP	Port range: 443
Source type: Custom	Source: Add CIDR, prefix list or security 0.0.0.0/0	Description - *optional*: *e.g. SSH for admin desktop*

Security group rule 4 (TCP, 1024-65535, 0.0.0.0/0)

Type: Custom TCP	Protocol: TCP	Port range: 1024-65535
Source type: Custom	Source: Add CIDR, prefix list or security 0.0.0.0/0	Description - *optional*: *e.g. SSH for admin desktop*

- Click the dropdown for **Advanced details**
  - Scroll down and in the **user data**, paste the following:

#!/bin/bash

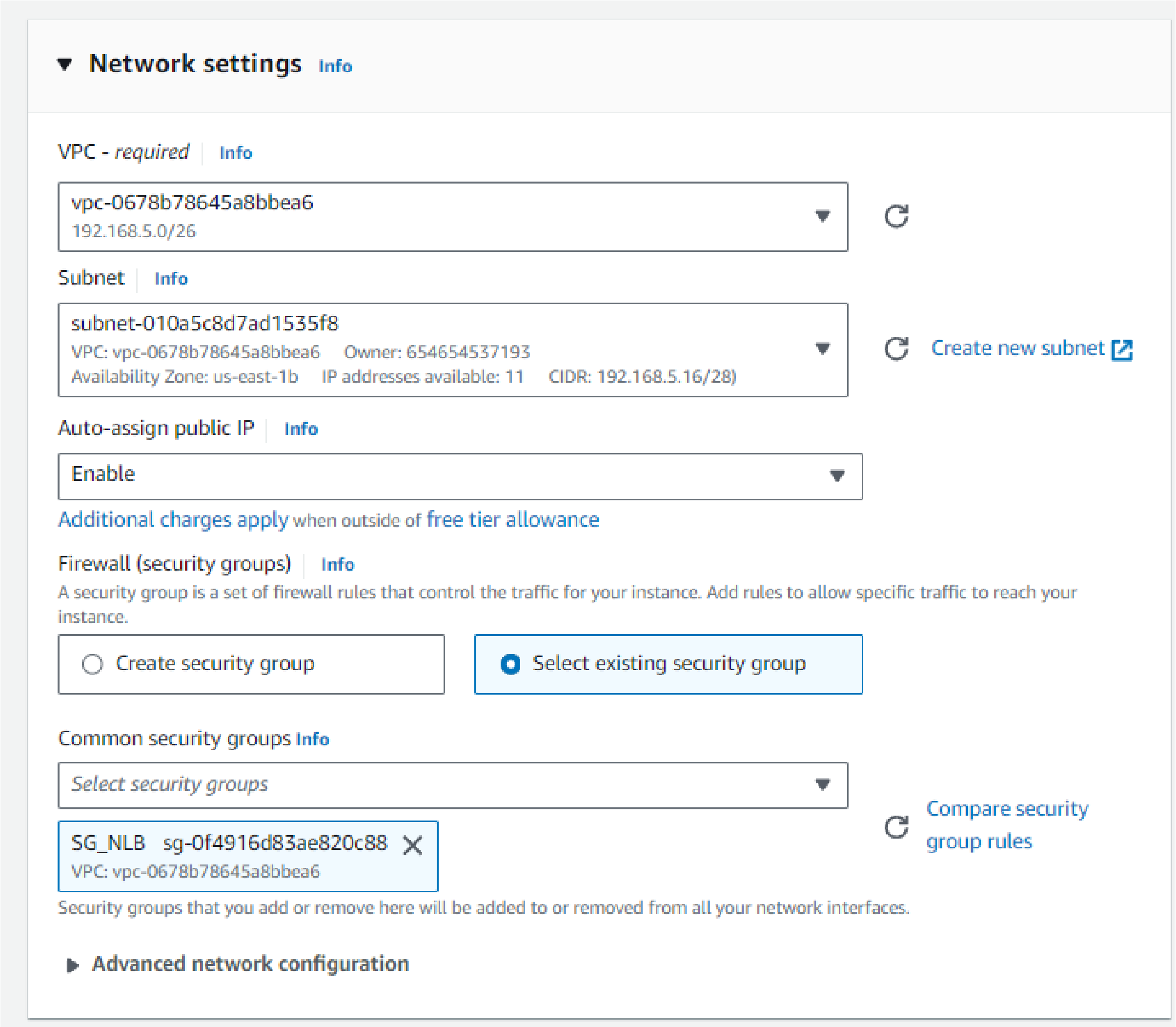
yum update -y

```
yum install -y httpd

systemctl start httpd

systemctl enable httpd

echo "<h1>Welcome to the webServer_Alpha</h1>" > /var/www/html/index.html
```

- Click **Launch instance**

3. **Launch the second EC2 Instances using the following configurations:**

- Name: **webServer_Beta**
- AMI: **Amazon Linux**
- Instance type: **t2.micro**
- Key pair: select the **web-server-key-pair** we created from the first instance
- Network settings: (Click “**Edit**”)
- Subnet: (Choose the subnet that is in the AZ: **us-east-1b**)
- Auto-assign public IP: Select **Enable**
- Firewall (security groups): tick on the **Select existing security group**, choose **SG_NLB**



▼ Network settings Info

VPC - *required* Info

vpc-0678b78645a8bbea6
192.168.5.0/26

Subnet Info

subnet-010a5c8d7ad1535f8
VPC: vpc-0678b78645a8bbea6 Owner: 654654537193
Availability Zone: us-east-1b IP addresses available: 11 CIDR: 192.168.5.16/28)

Create new subnet

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group ● Select existing security group

Common security groups Info

Select security groups

SG_NLB sg-0f4916d83ae820c88 ✕
VPC: vpc-0678b78645a8bbea6

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

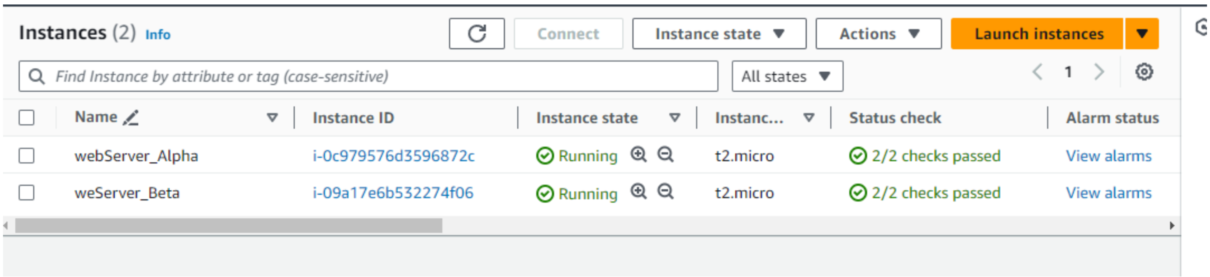
▶ Advanced network configuration

- Click the dropdown for **Advanced details**
  - Scroll down and in the **user data**, paste the following:

```
#!/bin/bash

yum update -y

yum install -y httpd

systemctl start httpd

systemctl enable httpd

echo "<h1>Welcome to the webServer_Beta</h1>" > /var/www/html/index.html
```

- Click **Launch instance**

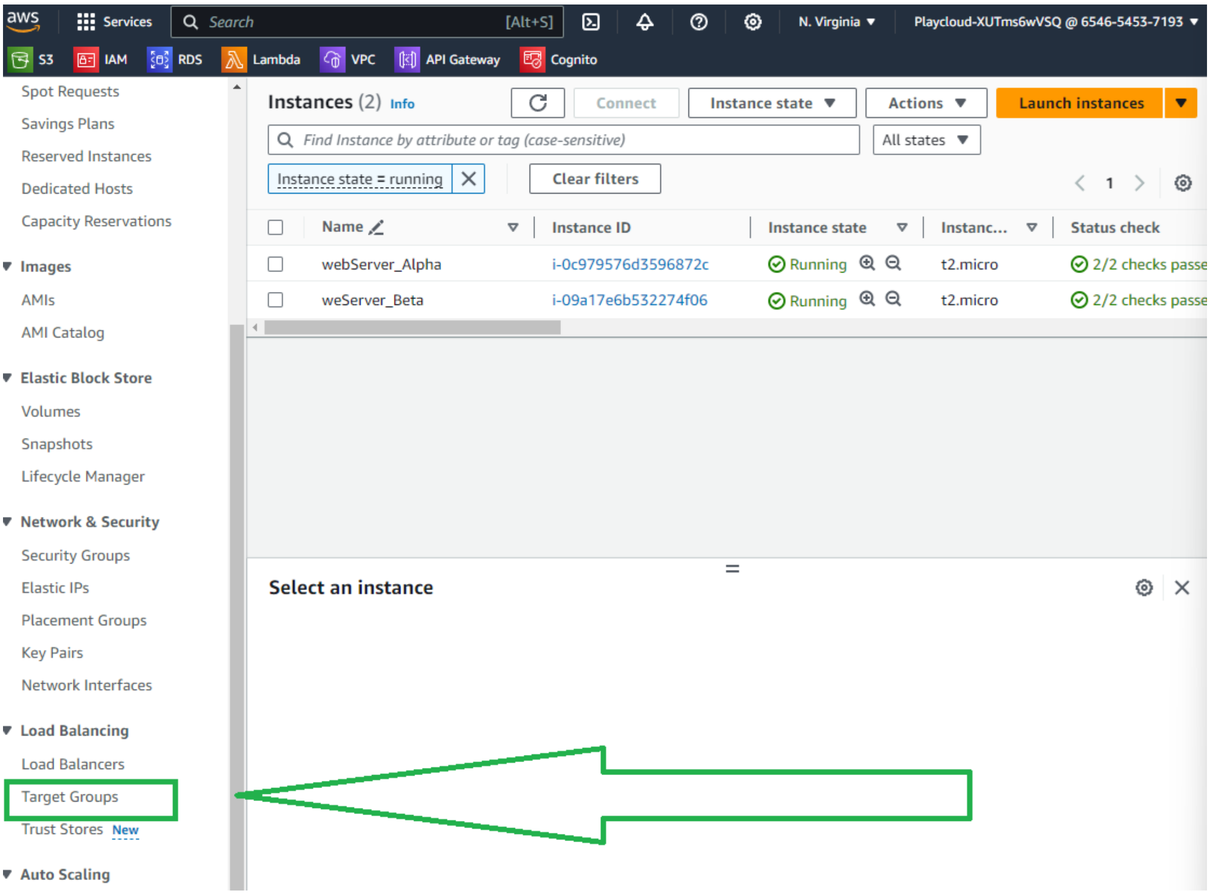
4. Wait for your instances to be in **running** state and **2/2 checks passed** status.



□	Name	Instance ID	Instance state	Instanc...	Status check	Alarm status
□	webServer_Alpha	i-0c979576d3596872c	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms
□	weServer_Beta	i-09a17e6b532274f06	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms

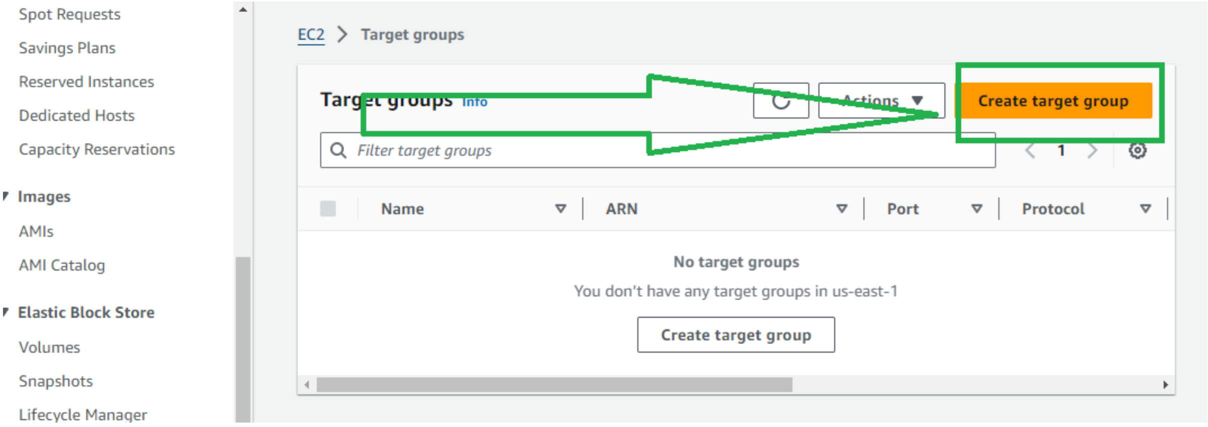
Setting Up Target Group

1. Go to the EC2 Dashboard, scroll down from the left sidebar and under ‘**Load Balancing**‘, select ‘**Target Groups**‘.



□	Name	Instance ID	Instance state	Instanc...	Status check
□	webServer_Alpha	i-0c979576d3596872c	✓ Running	t2.micro	✓ 2/2 checks passe
□	weServer_Beta	i-09a17e6b532274f06	✓ Running	t2.micro	✓ 2/2 checks passe

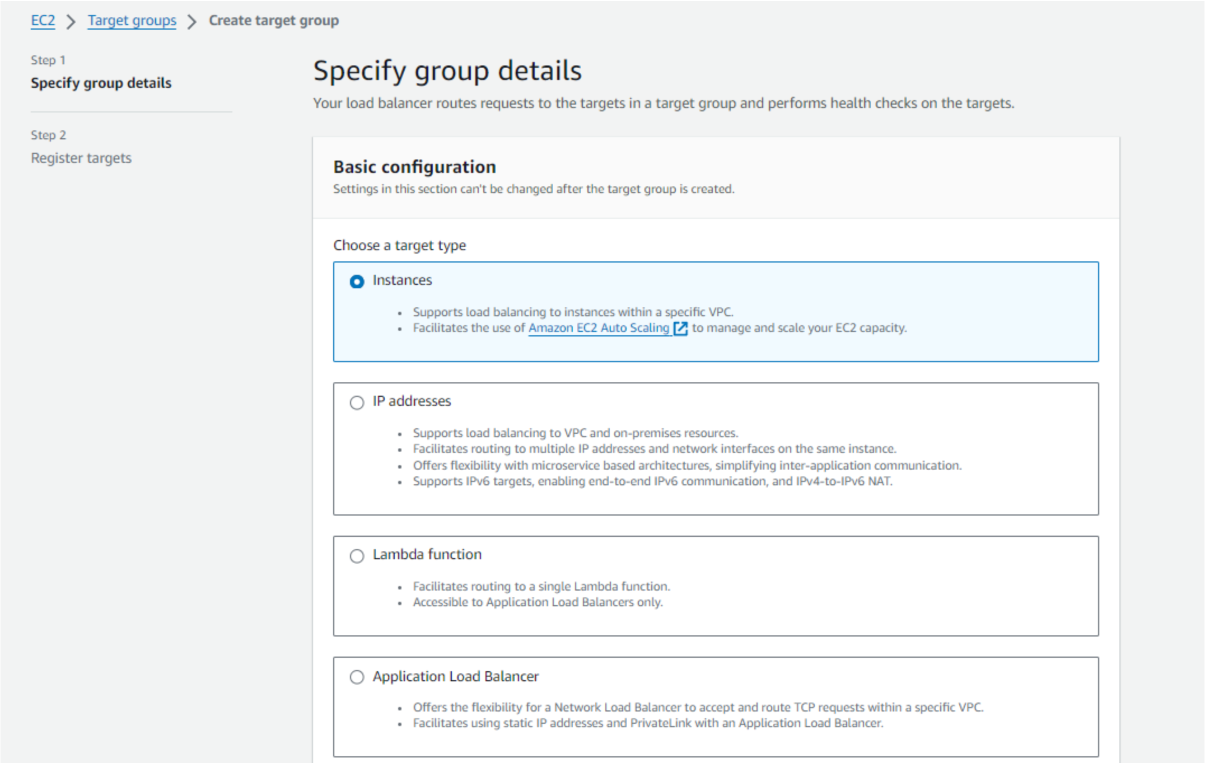
2. Click ‘**Create target group**‘



No target groups

You don't have any target groups in us-east-1

3. Choose ‘**Instances**‘ as the target type.



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.

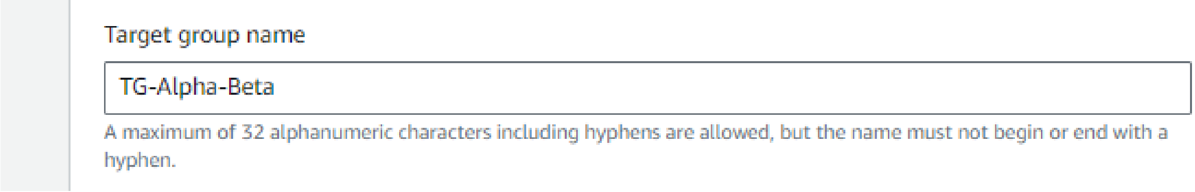
○ Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

○ Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

4. Name the Target group (for example **TG-Alpha-Beta)**

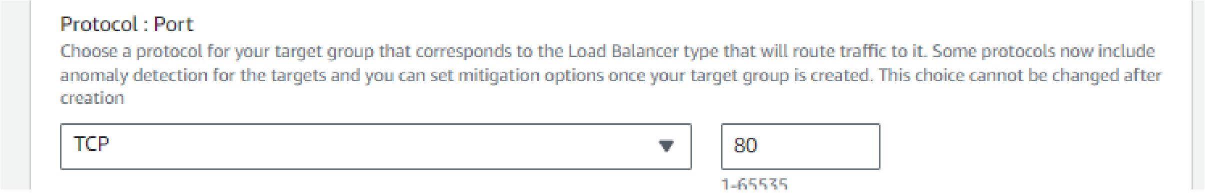


Target group name

TG-Alpha-Beta

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

5. Specify protocol as (**TCP**) and port as (**80**).



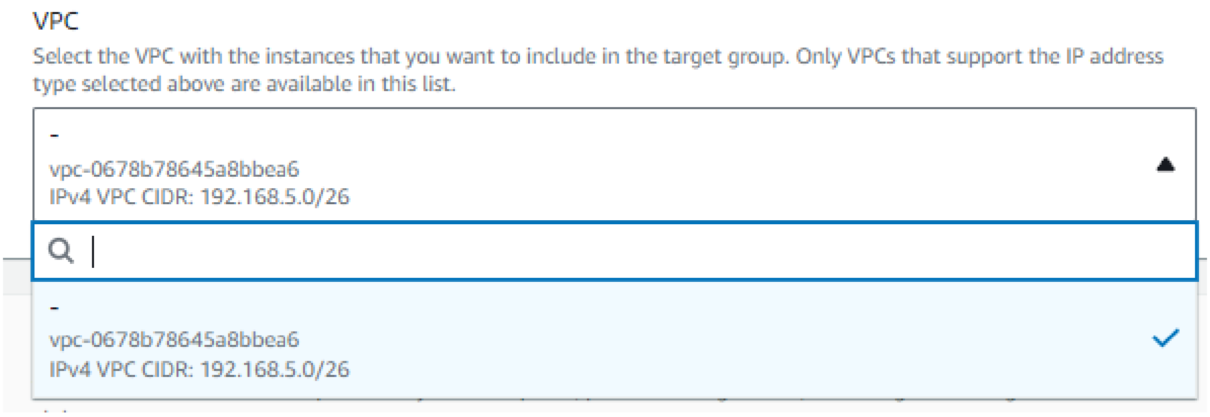
Protocol : Port

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

TCP ▼ 80

1-65535

6. Under **VPC**, Select the default VPC given.



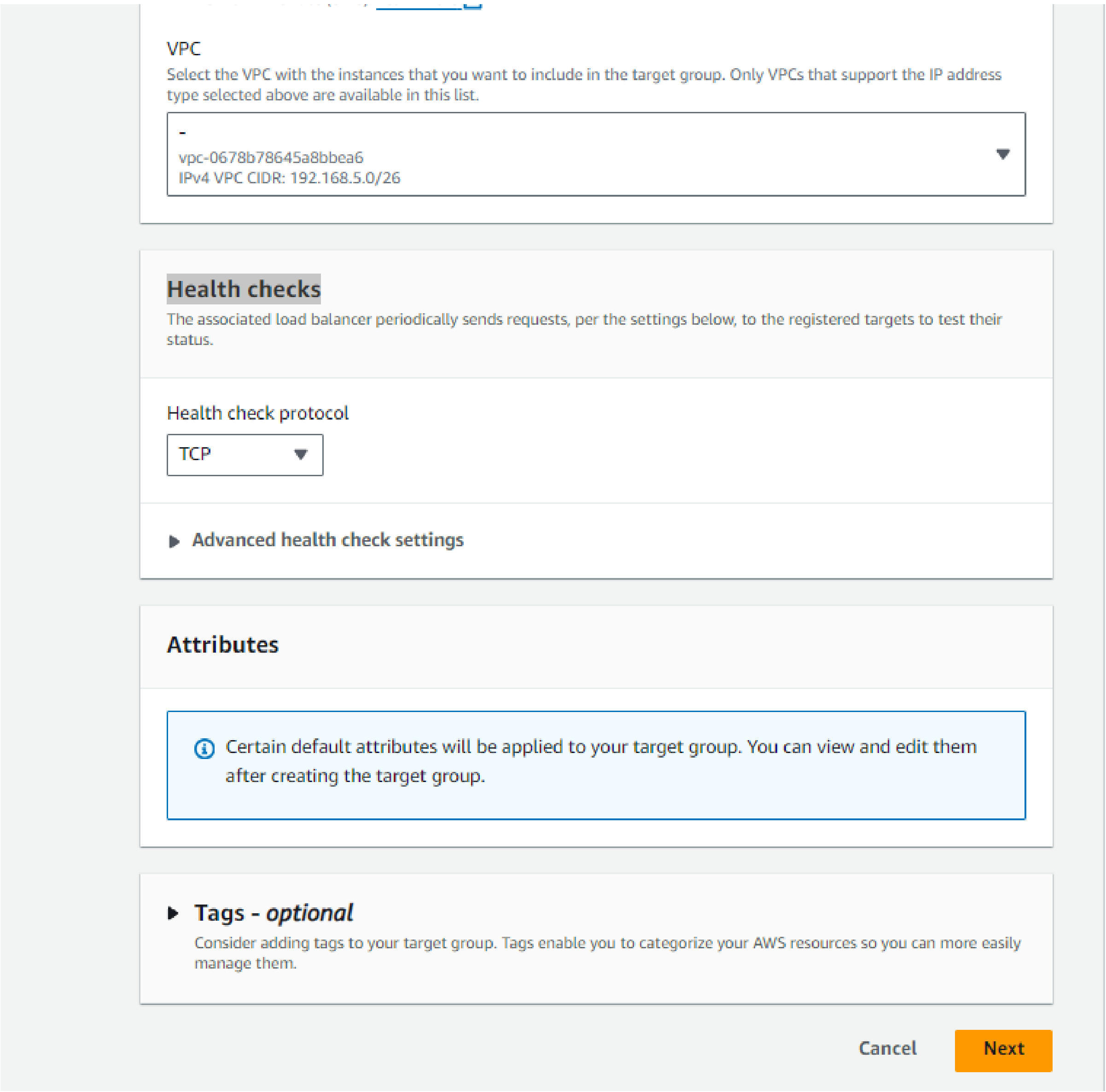
VPC

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-0678b78645a8bbea6
IPv4 VPC CIDR: 192.168.5.0/26

-
vpc-0678b78645a8bbea6
IPv4 VPC CIDR: 192.168.5.0/26 ✓

7. For the **Health checks,** under **Health check protocol,** select **TCP.**



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-0678b78645a8bbea6
IPv4 VPC CIDR: 192.168.5.0/26

Health checks

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

Health check protocol: TCP

▶ Advanced health check settings

Attributes

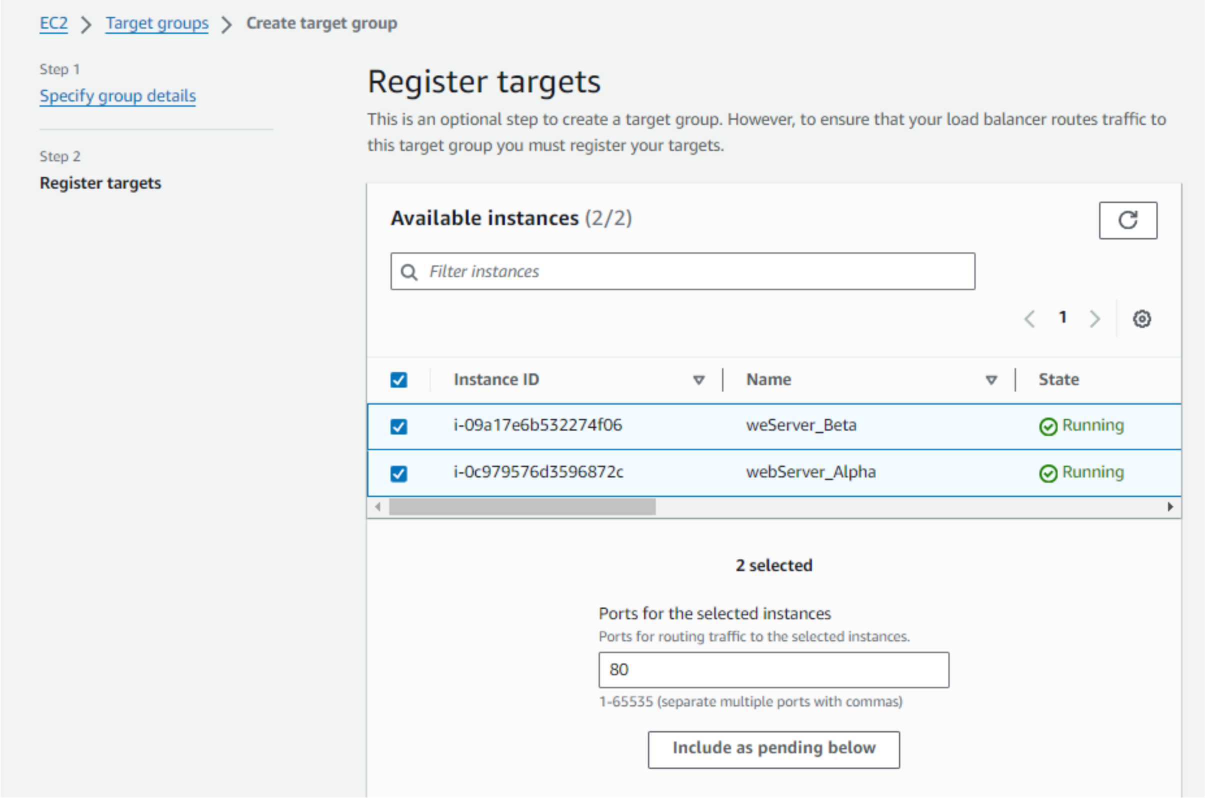
Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

▶ **Tags - *optional***

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

8. Click **Next.**

9. Register the newly created EC2 instances with this target group by selecting the two under Available instances.



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

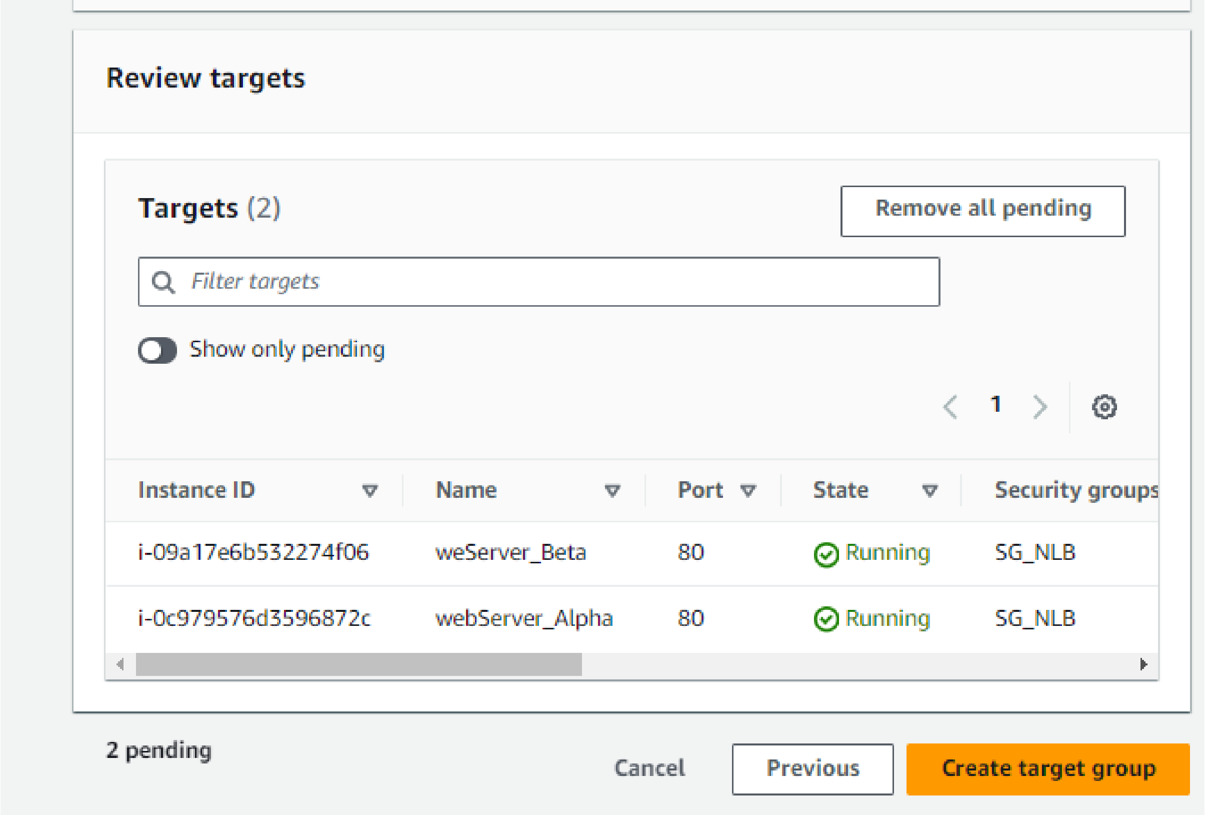
☑	Instance ID	Name	State
☑	i-09a17e6b532274f06	weServer_Beta	Running
☑	i-0c979576d3596872c	webServer_Alpha	Running

2 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)

Include as pending below

10. Click **Include as pending below** to add the two Instance in the Review targets



Review targets

Targets (2)

Remove all pending

Filter targets

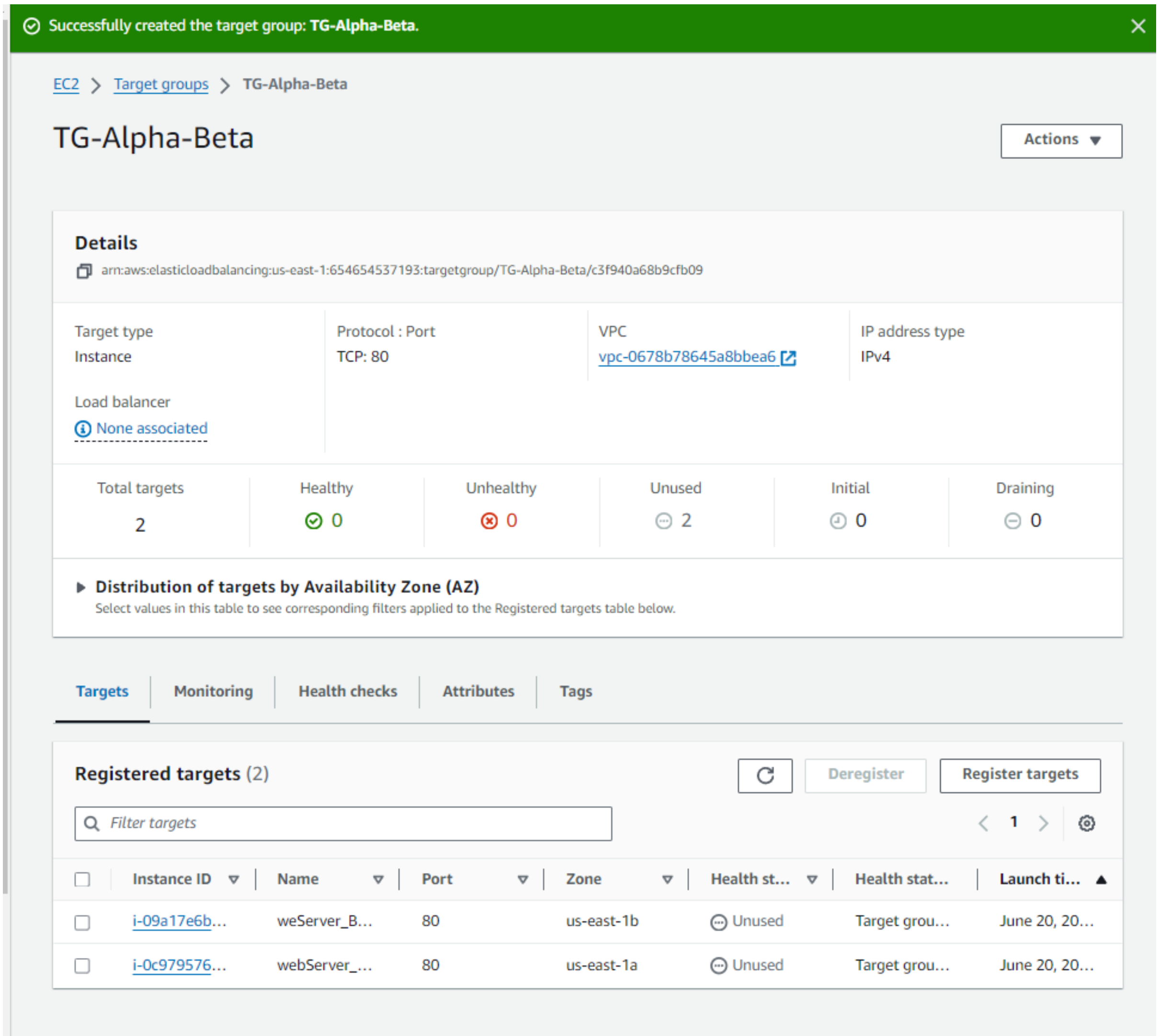
Show only pending

Instance ID	Name	Port	State	Security groups
i-09a17e6b532274f06	weServer_Beta	80	Running	SG_NLB
i-0c979576d3596872c	webServer_Alpha	80	Running	SG_NLB

2 pending

Cancel Previous Create target group

11. Click **Create target group**



Successfully created the target group: **TG-Alpha-Beta.**

EC2 > Target groups > TG-Alpha-Beta

TG-Alpha-Beta

Details

arn:aws:elasticloadbalancing:us-east-1:654654537193:targetgroup/TG-Alpha-Beta/c3f940a68b9cfb09

Target type	Protocol : Port	VPC	IP address type
Instance	TCP: 80	vpc-0678b78645a8bbea6	IPv4

Load balancer
None associated

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	0	0	2	0	0

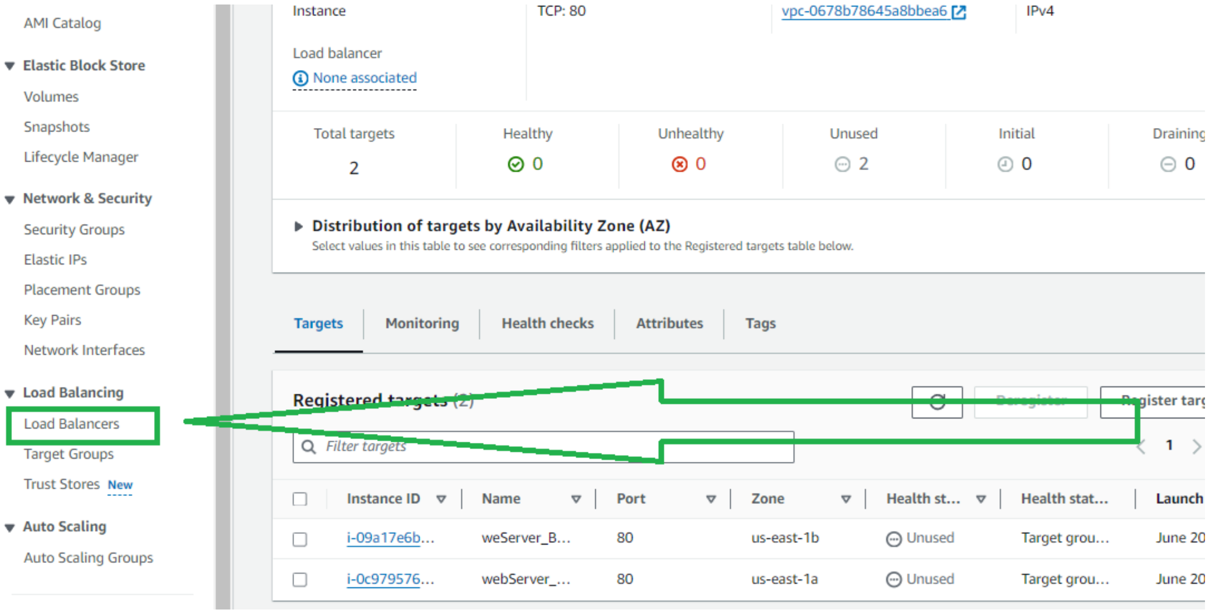
Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (2)

□	Instance ID	Name	Port	Zone	Health st...	Health stat...	Launch ti...
□	i-09a17e6b...	weServer_B...	80	us-east-1b	Unused	Target grou...	June 20, 20...
□	i-0c979576...	webServer_...	80	us-east-1a	Unused	Target grou...	June 20, 20...

Setting Up the Network Load Balancer

1. Navigate through the Left sidebar, under ‘**Load Balancing**‘, click ‘**Load Balancers’**



Instance | TCP: 80 | vpc-0678b78645a8bbea6 | IPv4

Load balancer
None associated

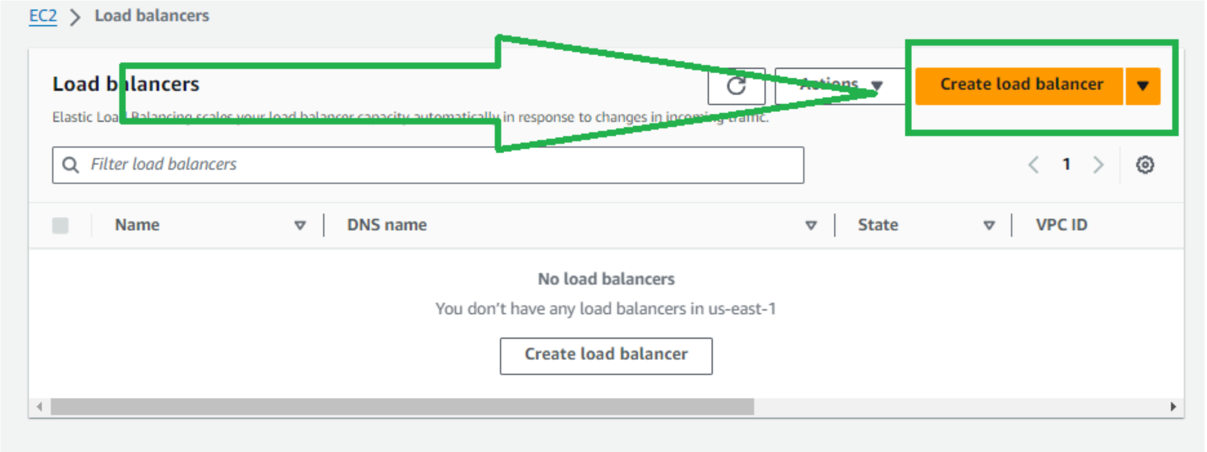
Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	0	0	2	0	0

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

Registered targets (2)

☐	Instance ID	Name	Port	Zone	Health st...	Health stat...	Launch
☐	i-09a17e6b...	weServer_B...	80	us-east-1b	Unused	Target grou...	June 20
☐	i-0c979576...	webServer_...	80	us-east-1a	Unused	Target grou...	June 20

2. Click ‘**Create Load Balancer**‘

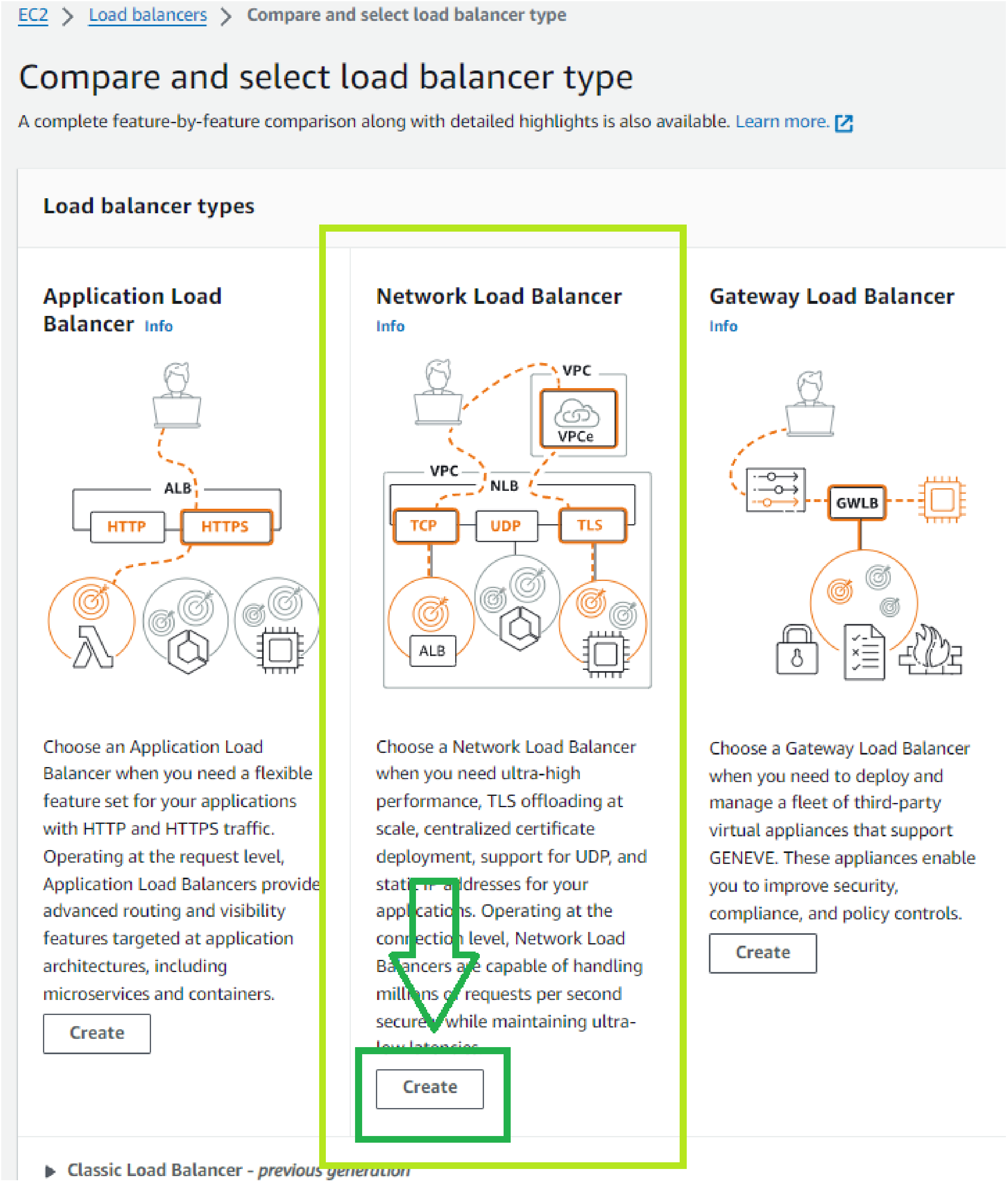


Load balancers
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

☐	Name	DNS name	State	VPC ID

No load balancers
You don't have any load balancers in us-east-1
Create load balancer

3. Select ‘**Network Load Balancer**‘ by clicking **Create** under it.



EC2 > Load balancers > Compare and select load balancer type

Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. Learn more.

Load balancer types

Application Load Balancer Info

Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create

Network Load Balancer Info

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

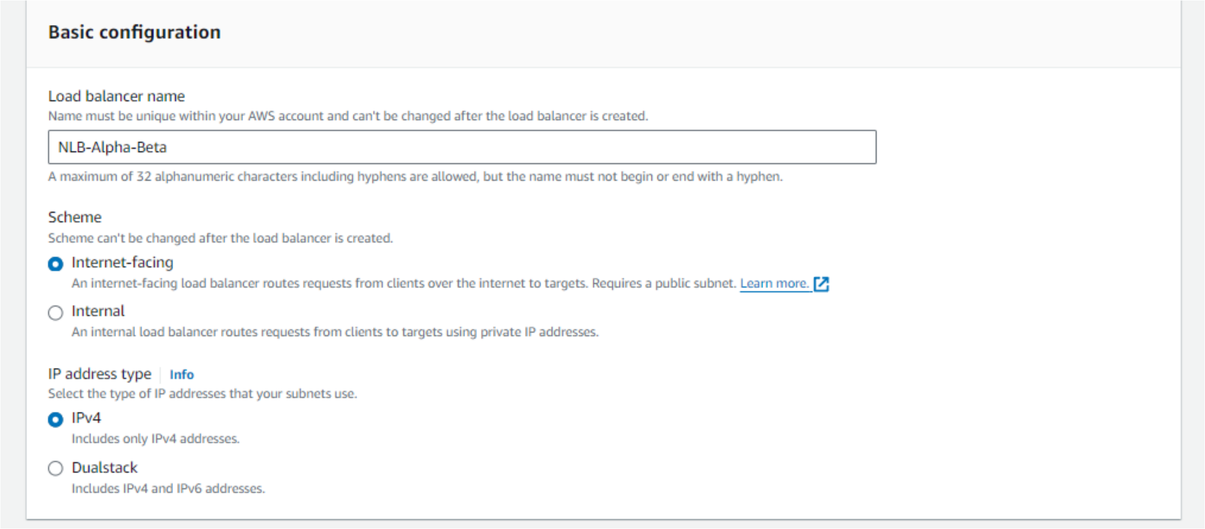
Gateway Load Balancer Info

Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

▶ Classic Load Balancer - *previous generation*

4. Enter a name for the Load Balancer (for example **NLB-Alpha-Beta**)



Basic configuration

Load balancer name

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

NLB-Alpha-Beta

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

Scheme

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.

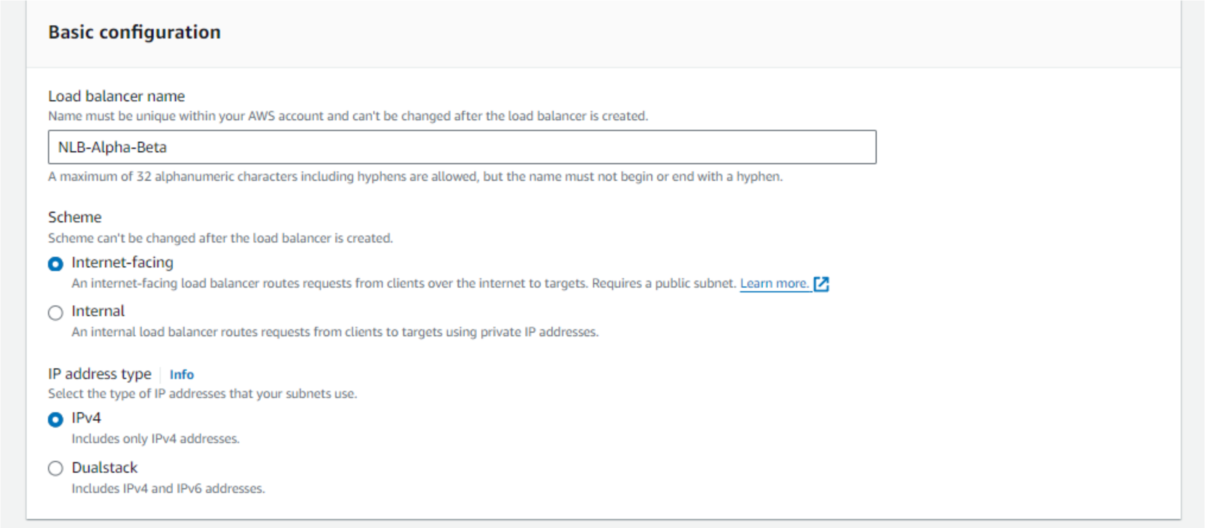
● IPv4

Includes only IPv4 addresses.

○ Dualstack

Includes IPv4 and IPv6 addresses.

5. For the Scheme, select **Internet-facing**, and for teh IP address type is **IPv4**



Basic configuration

Load balancer name

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

NLB-Alpha-Beta

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

Scheme

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.

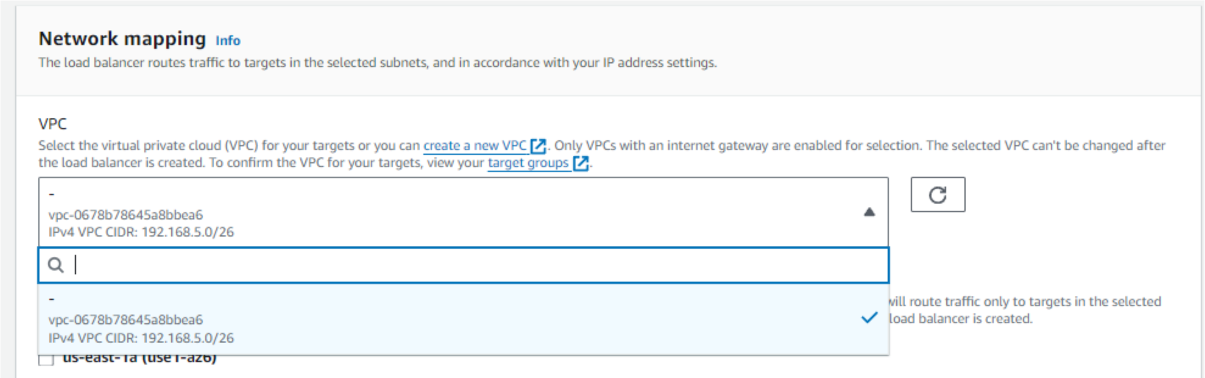
● IPv4

Includes only IPv4 addresses.

○ Dualstack

Includes IPv4 and IPv6 addresses.

6. In the **Network mapping**, select the same **default VPC** that is given



Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC

Select the virtual private cloud (VPC) for your targets or you can create a new VPC. Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your target groups.

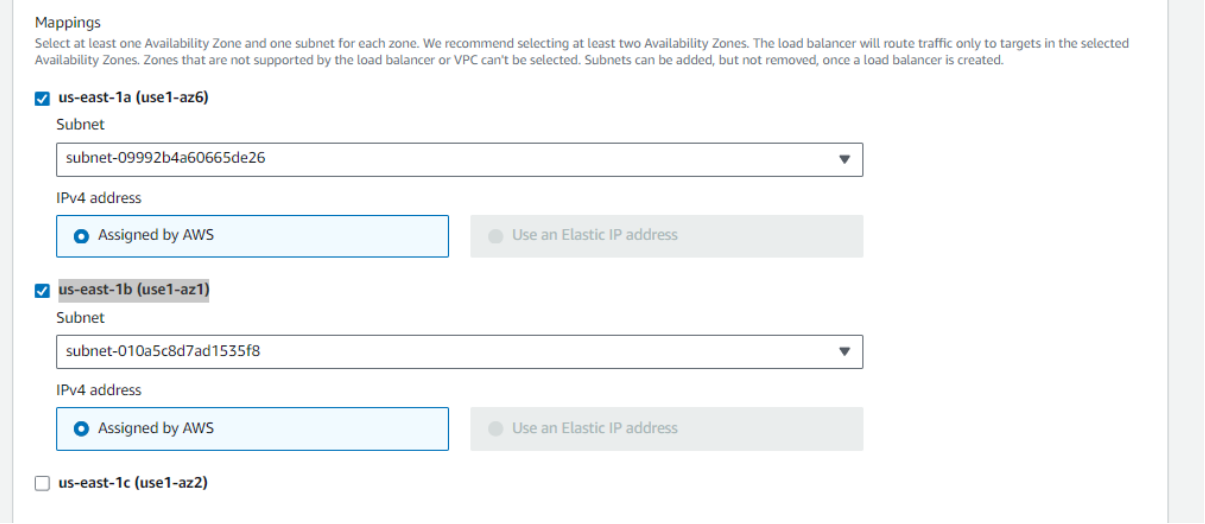
-
vpc-0678b78645a8bbea6
IPv4 VPC CIDR: 192.168.5.0/26

-
vpc-0678b78645a8bbea6 ✓
IPv4 VPC CIDR: 192.168.5.0/26

will route traffic only to targets in the selected load balancer is created.

□ us-east-1a (use1-az6)

7. Select the mappings with **us-east-1a** and **us-east-1b** on them. This will be the subnets which the NLB will operate.



Mappings

Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Zones that are not supported by the load balancer or VPC can't be selected. Subnets can be added, but not removed, once a load balancer is created.

☑ us-east-1a (use1-az6)

Subnet: subnet-09992b4a60665de26

IPv4 address: ● Assigned by AWS ○ Use an Elastic IP address

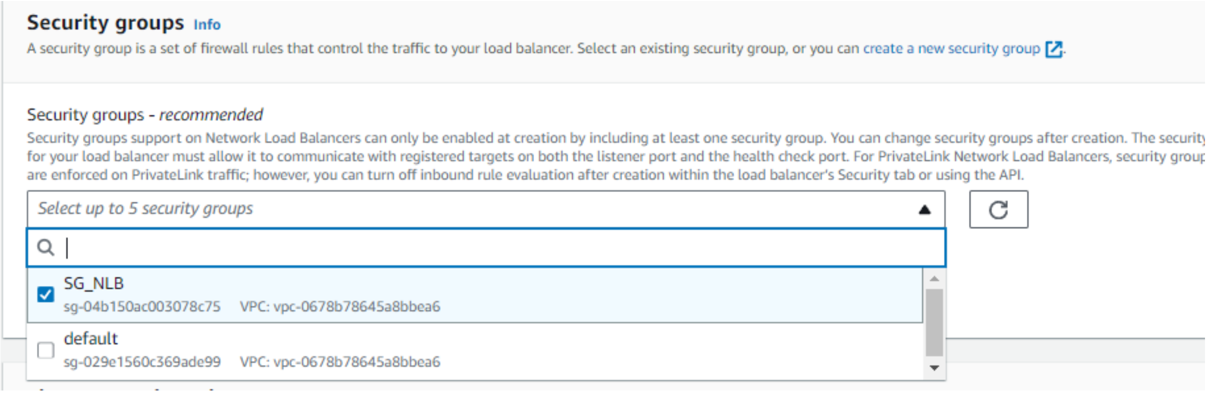
☑ us-east-1b (use1-az1)

Subnet: subnet-010a5c8d7ad1535f8

IPv4 address: ● Assigned by AWS ○ Use an Elastic IP address

□ us-east-1c (use1-az2)

8. Under **Security groups**, select the Security group we created a while ago and unselect the **default** to delete it from the list



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 - *recommended*

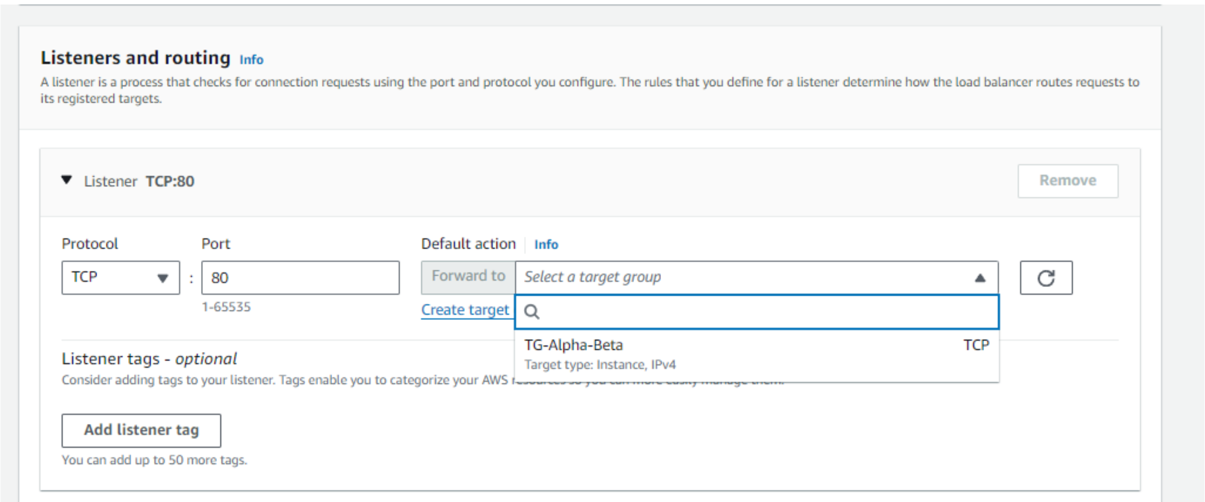
Security groups support on Network Load Balancers can only be enabled at creation by including at least one security group. You can change security groups after creation. The security for your load balancer must allow it to communicate with registered targets on both the listener port and the health check port. For PrivateLink Network Load Balancers, security group are enforced on PrivateLink traffic; however, you can turn off inbound rule evaluation after creation within the load balancer's Security tab or using the API.

Select up to 5 security groups

☑ SG_NLB sg-04b150ac003078c75 VPC: vpc-0678b78645a8bbea6

□ default sg-029e1560c369ade99 VPC: vpc-0678b78645a8bbea6

9. Next, on the **Listeners and routing,** select the target group we created.
*Ensure that the protocol is **TCP** and Port is **80***



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 requests to its registered targets.

▼ Listener TCP:80

Protocol: TCP

Port: 80 (1-65535)

Default action: Forward to *Select a target group*

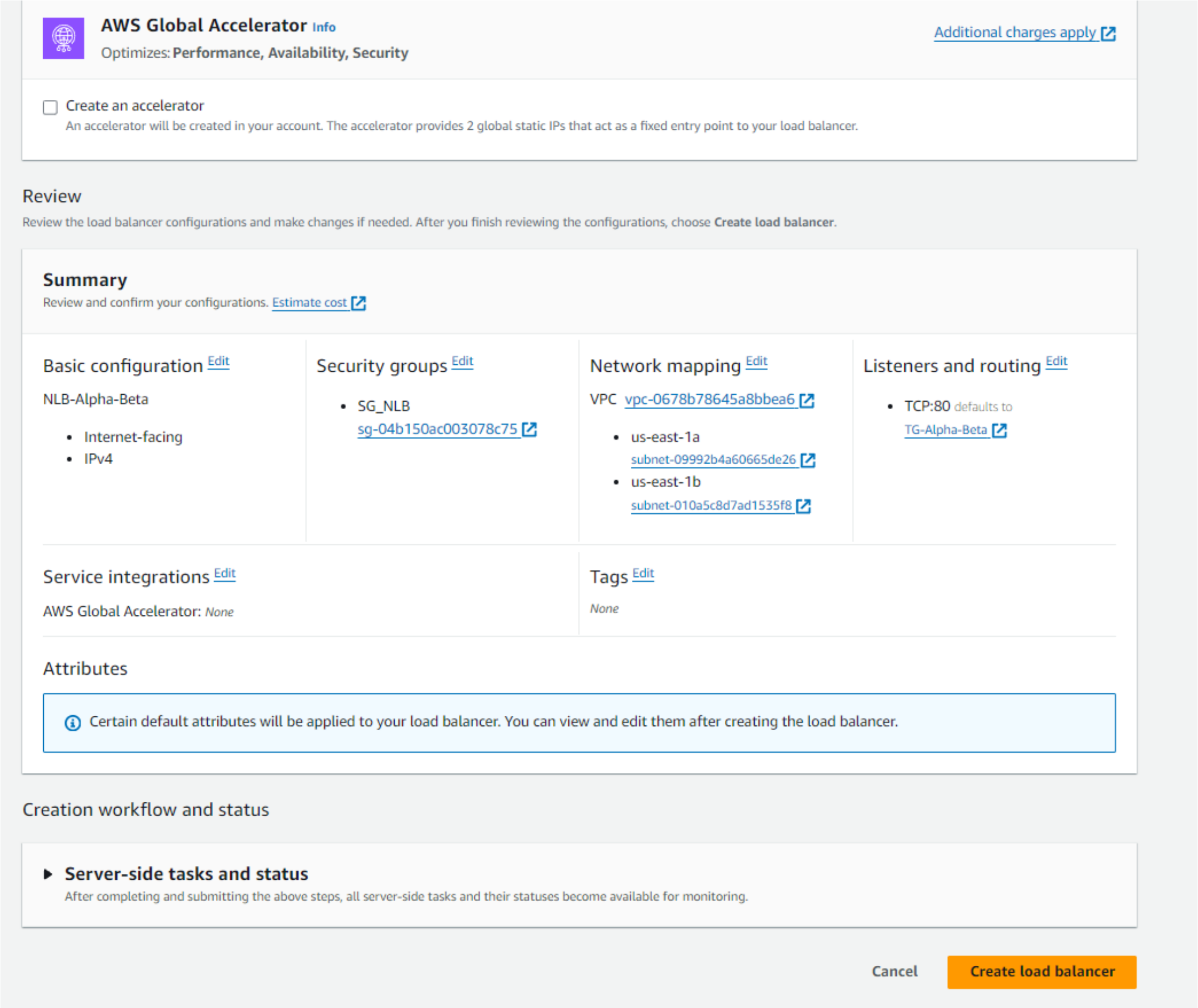
TG-Alpha-Beta TCP
Target type: Instance, IPv4

Listener tags - *optional*

Consider adding tags to your listener. Tags enable you to categorize your AWS

You can add up to 50 more tags.

10. Scroll down to the very bottom



AWS Global Accelerator Info

Optimizes: **Performance, Availability, Security**

Additional charges apply

□ Create an accelerator

An accelerator will be created in your account. The accelerator provides 2 global static IPs that act as a fixed entry point to your load balancer.

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

NLB-Alpha-Beta

- Internet-facing
- IPv4

Security groups Edit

- SG_NLB
  sg-04b150ac003078c75

Network mapping Edit

VPC vpc-0678b78645a8bbea6

- us-east-1a
  subnet-09992b4a60665de26
- us-east-1b
  subnet-010a5c8d7ad1535f8

Listeners and routing Edit

- TCP:80 defaults to
  TG-Alpha-Beta

Service integrations Edit

AWS Global Accelerator: *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.

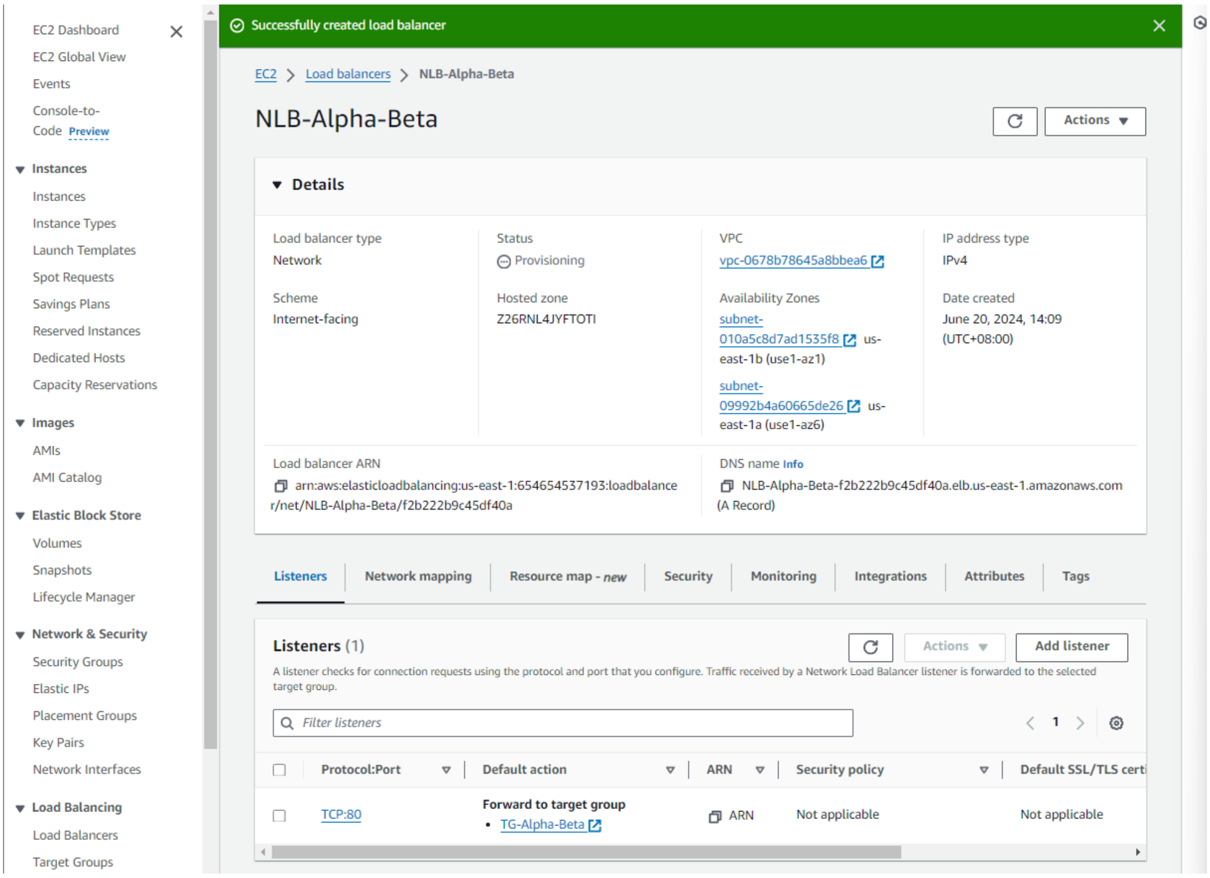
Creation workflow and status

▶ **Server-side tasks and status**

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

11. Click **Create load balancer**

12. The output would be:



Successfully created load balancer

NLB-Alpha-Beta

Details

Load balancer type: Network

Status: Provisioning

VPC: vpc-0678b78645a8bbea6

IP address type: IPv4

Scheme: Internet-facing

Hosted zone: Z26RNL4JYFTOTI

Availability Zones: subnet-010a5c8d7ad1535f8 us-east-1b (use1-az1); subnet-09992b4a60665de26 us-east-1a (use1-az6)

Date created: June 20, 2024, 14:09 (UTC+08:00)

Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:654654537193:loadbalancer/net/NLB-Alpha-Beta/f2b222b9c45df40a

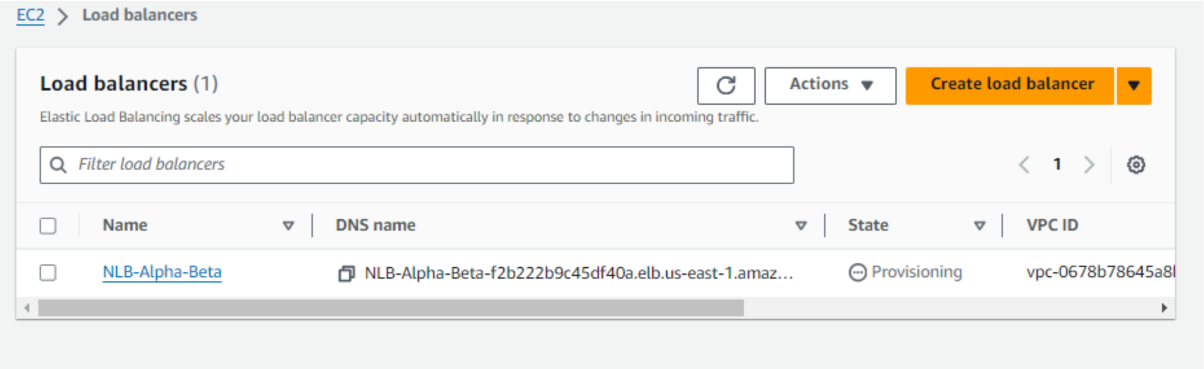
DNS name Info: NLB-Alpha-Beta-f2b222b9c45df40a.elb.us-east-1.amazonaws.com (A Record)

Listeners (1)

A listener checks for connection requests using the protocol and port that you configure. Traffic received by a Network Load Balancer listener is forwarded to the selected target group.

	Protocol:Port	Default action	ARN	Security policy	Default SSL/TLS certi
□	TCP:80	Forward to target group • TG-Alpha-Beta	ARN	Not applicable	Not applicable

13. Navigate back to the **Load Balancers** and wait for it to go from **Provisioning** to **Active** (click the refresh button occasionally)

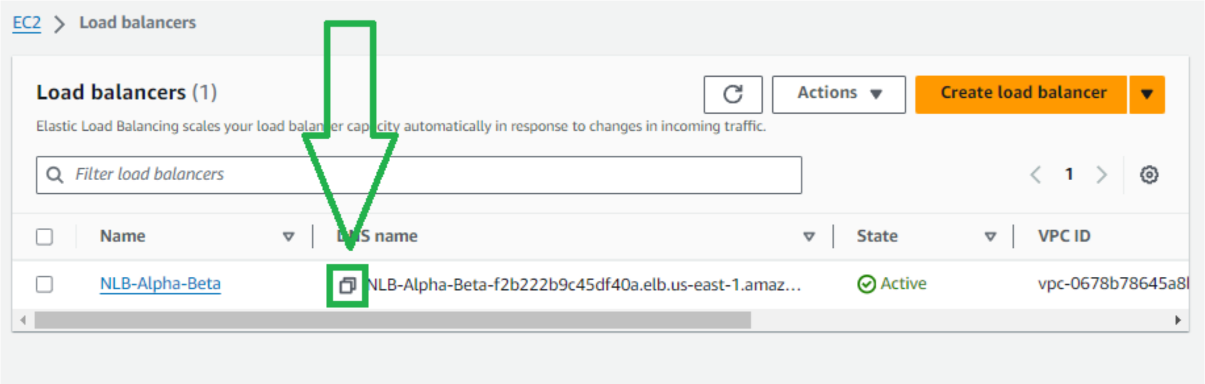


Load balancers (1)

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

	Name	DNS name	State	VPC ID
□	NLB-Alpha-Beta	NLB-Alpha-Beta-f2b222b9c45df40a.elb.us-east-1.amaz...	Provisioning	vpc-0678b78645a8l

14. Once **Active** , copy the **DNS name** of your NLB



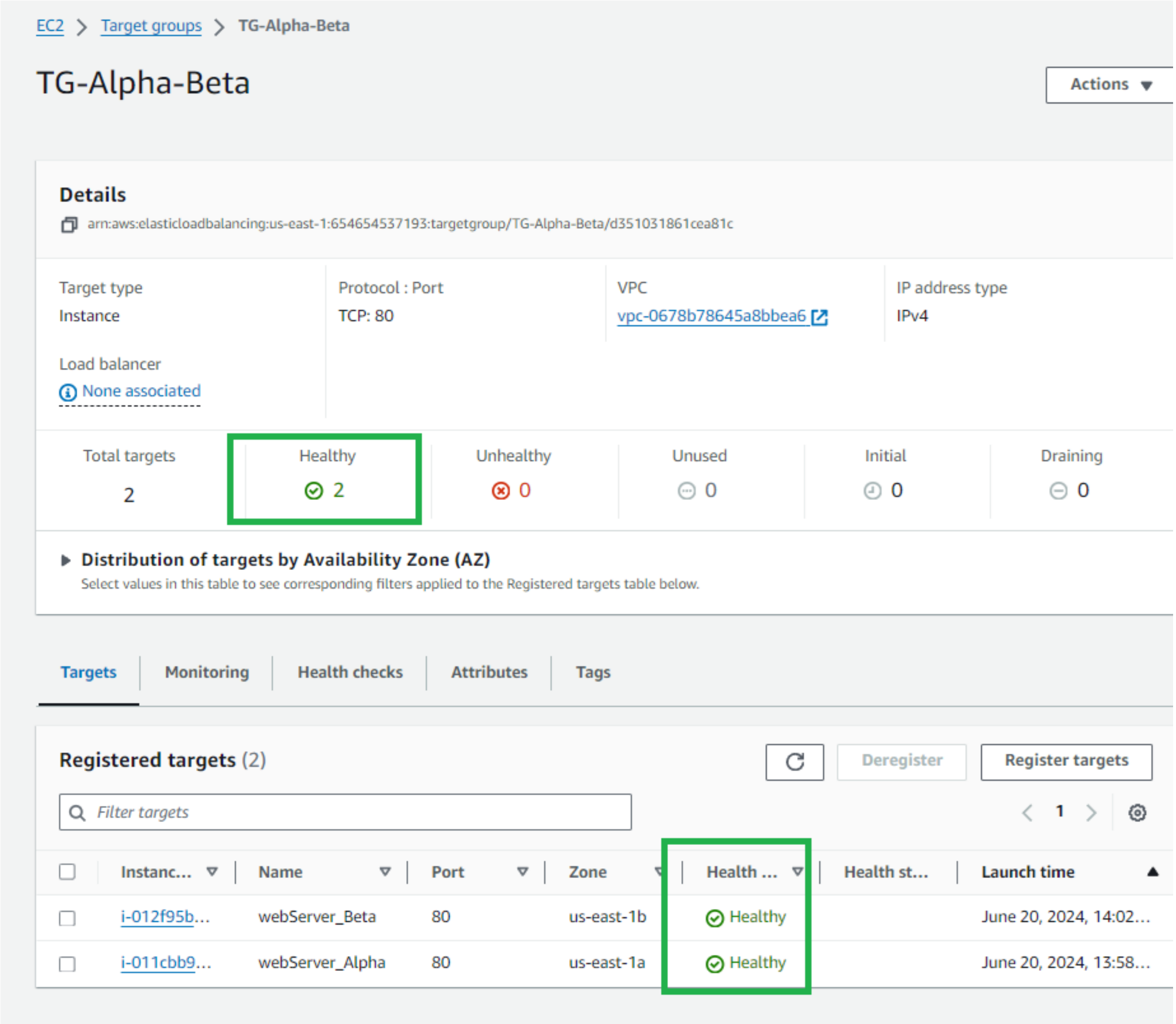
EC2 > Load balancers

Load balancers (1)

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

□	Name	DNS name	State	VPC ID
□	NLB-Alpha-Beta	NLB-Alpha-Beta-f2b222b9c45df40a.elb.us-east-1.amaz…	✓ Active	vpc-0678b78645a8l

15. Also, navigate back to the target group we created and ensure that the Registered targets are **Healthy**



EC2 > Target groups > TG-Alpha-Beta

TG-Alpha-Beta

Details

arn:aws:elasticloadbalancing:us-east-1:654654537193:targetgroup/TG-Alpha-Beta/d351031861cea81c

Target type: Instance

Protocol : Port: TCP: 80

VPC: vpc-0678b78645a8bbea6

IP address type: IPv4

Load balancer: None associated

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	✓ 2	☒ 0	0	0	0

▸ **Distribution of targets by Availability Zone (AZ)**

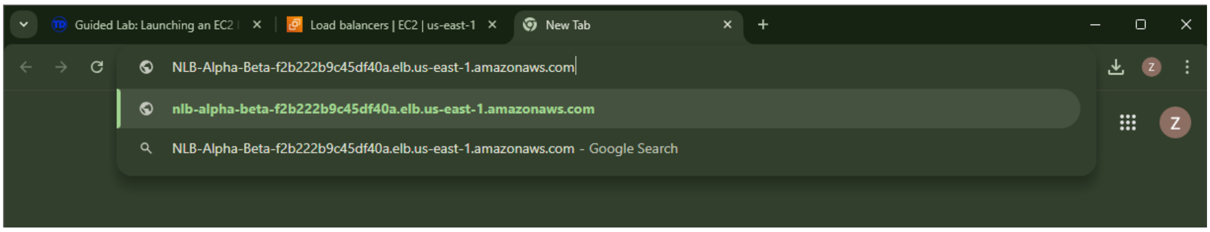
Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (2)

□	Instanc…	Name	Port	Zone	Health …	Health st…	Launch time
□	i-012f95b…	webServer_Beta	80	us-east-1b	✓ Healthy		June 20, 2024, 14:02…
□	i-011cbb9…	webServer_Alpha	80	us-east-1a	✓ Healthy		June 20, 2024, 13:58…

Testing the Network Load Balancer

1. In your browser add a new tab and Paste the DNS of your NLB you copied in the previous step.

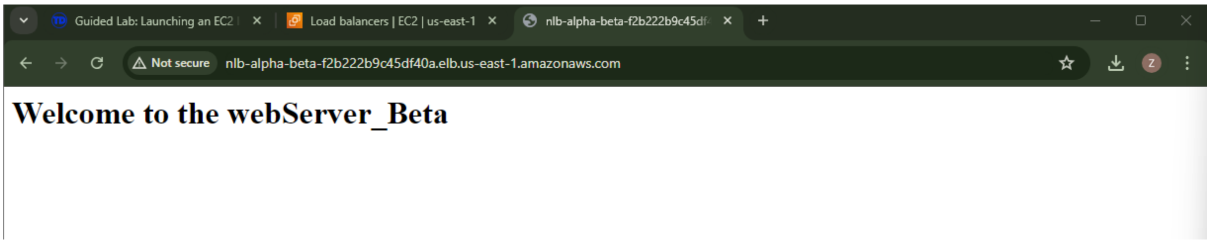

Guided Lab: Launching an EC2
Load balancers | EC2 | us-east-1
New Tab

NLB-Alpha-Beta-f2b222b9c45df40a.elb.us-east-1.amazonaws.com

nlb-alpha-beta-f2b222b9c45df40a.elb.us-east-1.amazonaws.com

NLB-Alpha-Beta-f2b222b9c45df40a.elb.us-east-1.amazonaws.com - Google Search

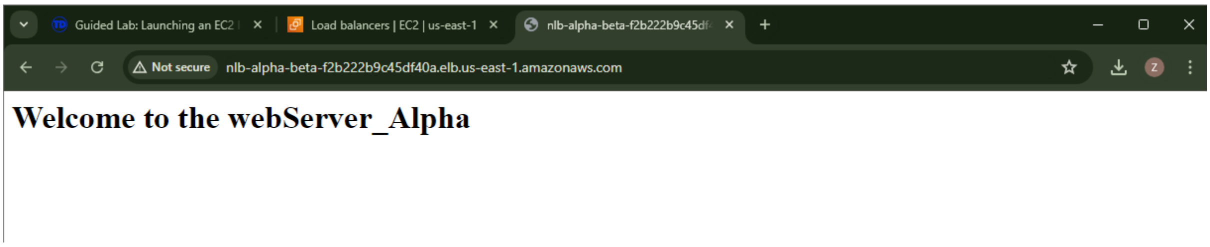
2. You should see either of the images below:


Guided Lab: Launching an EC2
Load balancers | EC2 | us-east-1
nlb-alpha-beta-f2b222b9c45df
Not secure

nlb-alpha-beta-f2b222b9c45df40a.elb.us-east-1.amazonaws.com

Welcome to the webServer_Beta

or


Guided Lab: Launching an EC2
Load balancers | EC2 | us-east-1
nlb-alpha-beta-f2b222b9c45df
Not secure

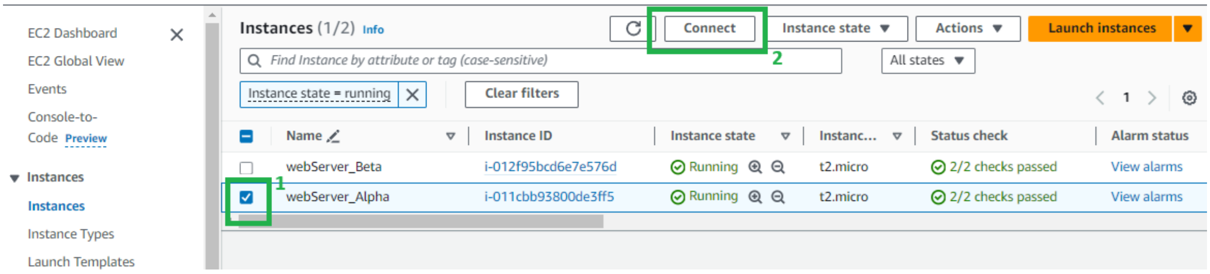
nlb-alpha-beta-f2b222b9c45df40a.elb.us-east-1.amazonaws.com

Welcome to the webServer_Alpha

3. Now, lets add some traffic to one of your instances to see if our NLB is really working fine.

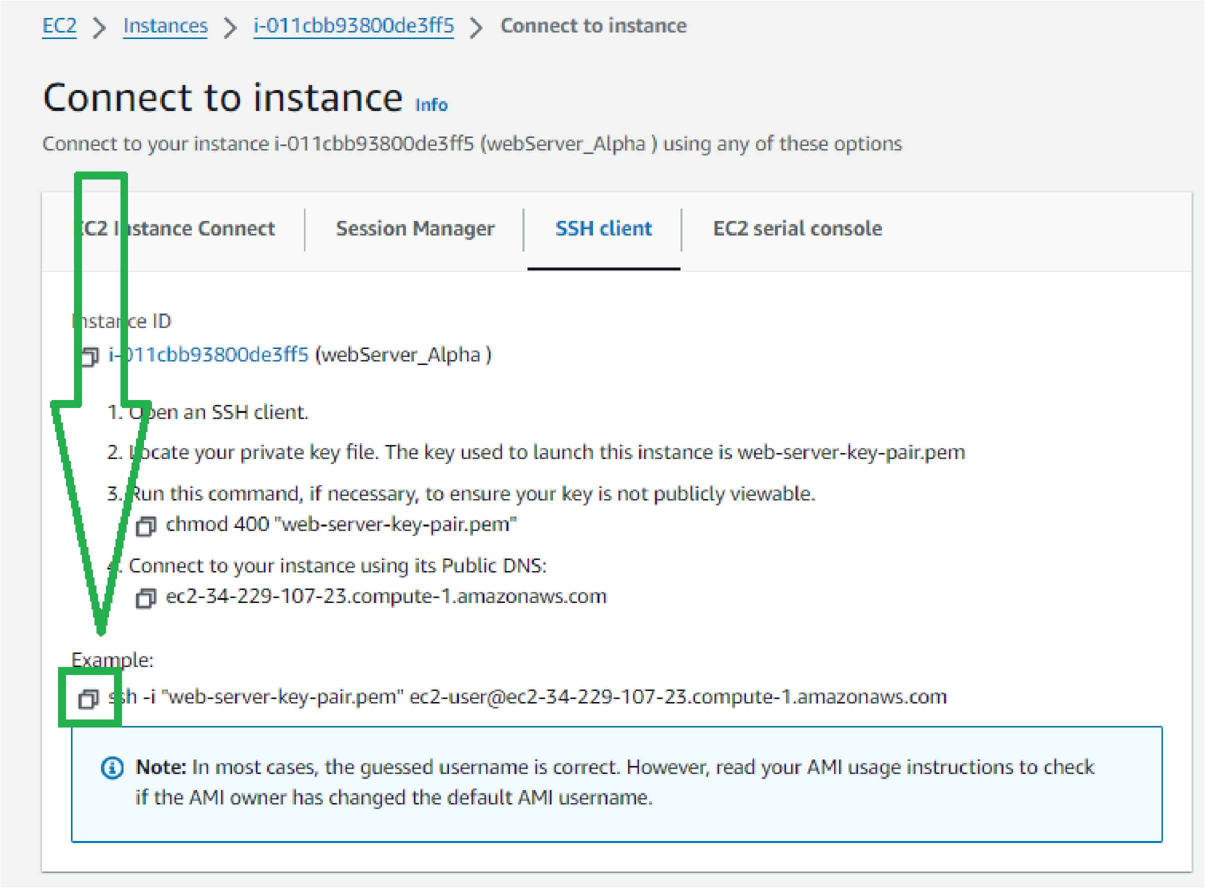
4. Connect to one of the created instance either the **webServer_Beta** or **webServer_Alpha** via SSH.

- Navigate back to the Instances
- Select one of the instances and click **Connect**



	Name	Instance ID	Instance state	Instanc...	Status check	Alarm status
□	webServer_Beta	i-012f95bcd6e7e576d	Running	t2.micro	2/2 checks passed	View alarms
☑	webServer_Alpha	i-011cbb93800de3ff5	Running	t2.micro	2/2 checks passed	View alarms

- Copy the ssh command:


EC2 > Instances > i-011cbb93800de3ff5 > Connect to instance

Connect to instance Info

Connect to your instance i-011cbb93800de3ff5 (webServer_Alpha) using any of these options

Instance ID

i-011cbb93800de3ff5 (webServer_Alpha)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is web-server-key-pair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "web-server-key-pair.pem"
4. Connect to your instance using its Public DNS:
   ec2-34-229-107-23.compute-1.amazonaws.com

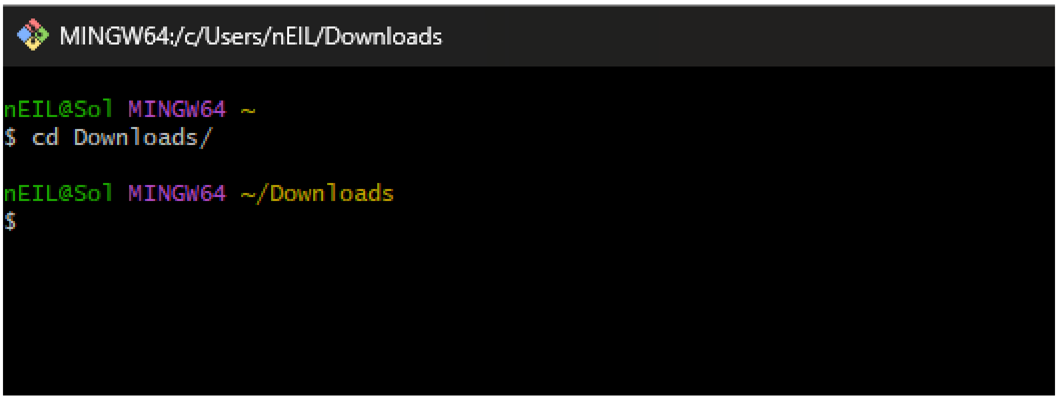
Example:

ssh -i "web-server-key-pair.pem" ec2-user@ec2-34-229-107-23.compute-1.amazonaws.com

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

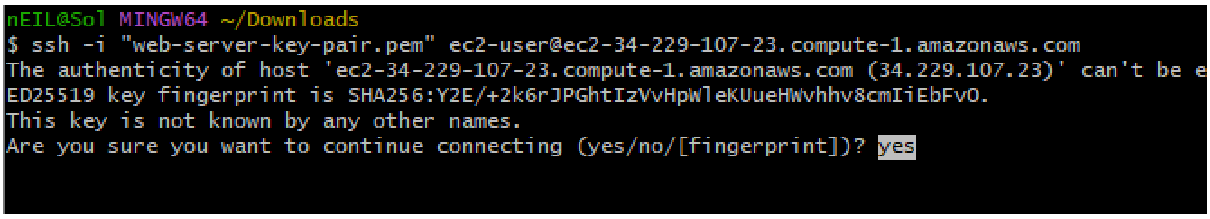
- Now, open your **terminal** or **GitBash**
  - change the directory where the **web-server-keypair.pem** is downnloaded. Usually its in the Downloads folder

cd <directory>


MINGW64:/c/Users/nEIL/Downloads

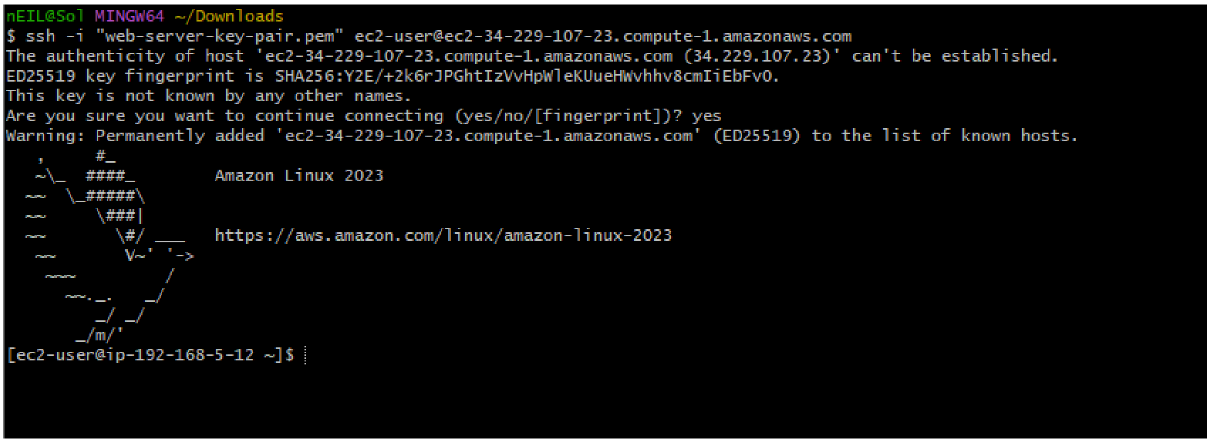
```
nEIL@Sol MINGW64 ~
$ cd Downloads/

nEIL@Sol MINGW64 ~/Downloads
$
```

- If a question will pop up like the image below, just type **yes** and hit **enter**



```
nEIL@Sol MINGW64 ~/Downloads
$ ssh -i "web-server-key-pair.pem" ec2-user@ec2-34-229-107-23.compute-1.amazonaws.com
The authenticity of host 'ec2-34-229-107-23.compute-1.amazonaws.com (34.229.107.23)' can't be e
ED25519 key fingerprint is SHA256:Y2E/+2k6rJPGhtIzVvHpWleKUueHWvhhv8cmIiEbFv0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

- You will then be connected to your instance:

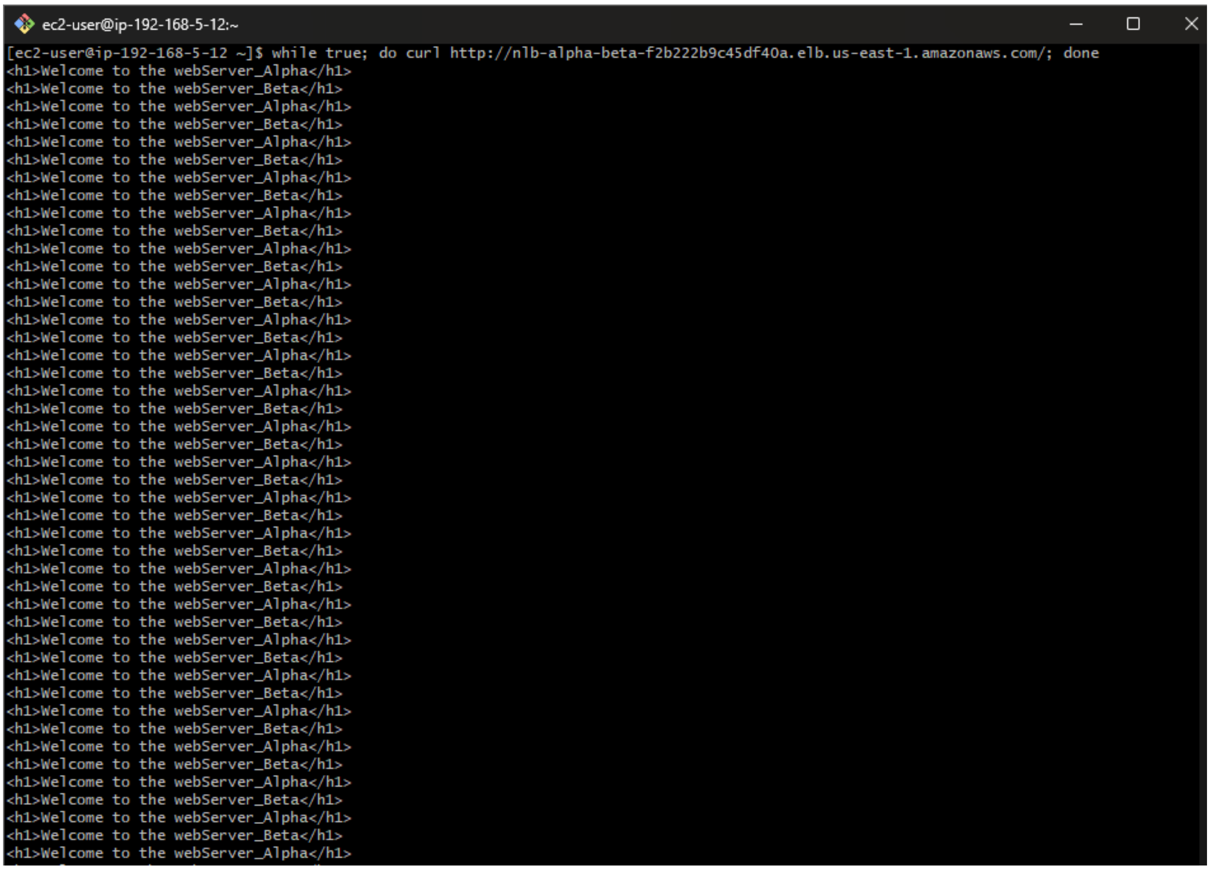


```
nEIL@Sol MINGW64 ~/Downloads
$ ssh -i "web-server-key-pair.pem" ec2-user@ec2-34-229-107-23.compute-1.amazonaws.com
The authenticity of host 'ec2-34-229-107-23.compute-1.amazonaws.com (34.229.107.23)' can't be established.
ED25519 key fingerprint is SHA256:Y2E/+2k6rJPGhtIzVvHpWleKUueHWvhhv8cmIiEbFvO.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-229-107-23.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-192-168-5-12 ~]$
```

- Lasty, Paste the following command

while true; do curl http://<NLB-DNS-NAME>; done

Ensure that the NLB-DNS-NAME is correct



```
[ec2-user@ip-192-168-5-12 ~]$ while true; do curl http://nlb-alpha-beta-f2b222b9c45df40a.elb.us-east-1.amazonaws.com/; done
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
<h1>Welcome to the webServer_Beta</h1>
<h1>Welcome to the webServer_Alpha</h1>
```

Do you notice how the welcome message change from **webServer_Alpha** to **webServer_Beta**? This means that the Network Load Balancer we created are working as intended

That’s it! Congratulations! You have created a functional Network Load Balancer that effectively distributes incoming traffic across multiple backend servers. This setup enhances the fault tolerance of your applications by ensuring no single server bears too much load. Experiment further by adjusting settings like health check intervals and thresholds to see how they impact the performance of your NLB.

This lab serves as a foundational exercise in understanding and utilizing AWS Load Balancers to improve application scalability and reliability.

One last thing! It is a good practice to clean up the resources created during this lab. Not only will it make you a better professional, but you will also become a more organized person. Happy learning!