

Name - Priyanshu Lapkale
Class - TY B
Roll No - 322067
PRN - 22220008

Network Load Balancer on EC2 Instance

- NLB serves as the single point of contact for clients. The load balancer distributes incoming traffic across multiple targets, such as Amazon EC2 instances. This increases the availability of your application. You add one or more listeners to your load balancer.
- For example, it is like a toll on a highway that divides all the incoming traffic into multiple lanes, so that all the pressure doesn't come on a single lane. (Here cars i.e. traffic is nothing but users/clients and one lane is one instance and toll is load balancer)

Let's make a NLB that distribute our traffic -

First make few EC2 instances i.e. lanes for which we can make NLB -

Instances (4) Info											
Find Instance by attribute or tag (case-sensitive)											
Any state											
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	
	icy-server-b	I-03487eb04fb112e90	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-13-201-55-98.ap-s...	13.201.55.98	-	
	icy-server-a	I-0dd77cda925ada228	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-15-206-81-19.ap-s...	15.206.81.19	-	

This is **icy-server-a** :



This is server A

This is **icy-server-b** :



This is server B

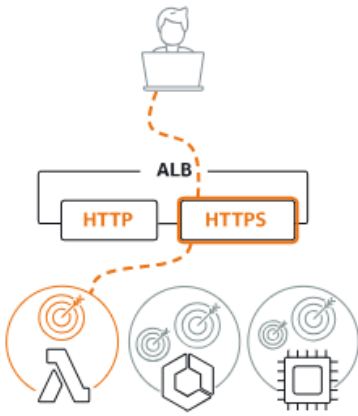
Now let's create Network Load Balancer -

- From the navigation bar on the left select *Load Balancer*.
- Then click on *Create load balancer*

- Then select Network Load Balancer

Load balancer types

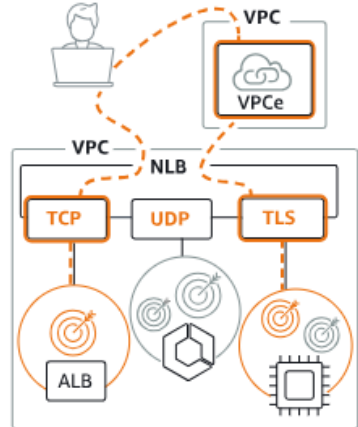
Application Load Balancer [Info](#)



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**

1. Basic configuration -

- Give a suitable name for you network load balancer.
- Select Internet-facing scheme, as we want to routes requests from clients over the internet to target.

- For IP address typer select IPv4

► How Network 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.

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**
Recommended for internal load balancers.
☐ **Dualstack**
Includes IPv4 and IPv6 addresses.

2. Network Mapping -

For this part if you want to create your VPC you can but i'll be going with default. Select atleast one availabilty zone for mapping, the NLB will route traffic only to targets in the selected Availability Zone.

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-039948e0d48e3f441
IPv4: 172.31.0.0/16

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.

☒ **ap-south-1a (aps1-az1)**
Subnet

IPv4 address

☐ **ap-south-1b (aps1-az3)**

☐ **ap-south-1c (aps1-az2)**

3. Security Groups - Let's create a new SG. Click on create a new security group option.

- Give suitable name for you SG. e.g. icy-nlb-sg (Network Load Balancer Security Group for icy server)
- Set HTTP and SSH for inbound rule with source as Anywhere IPv4. (Anywhere because client will be approaching our NLB not instances)

EC2 > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info

Name cannot be edited after creation.

Description Info

VPC Info

Inbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional	
HTTP	TCP	80	Anywhere... <input type="text" value="0.0.0.0"/>		Delete
SSH	TCP	22	Anywhere... <input type="text" value="0.0.0.0"/>		Delete

- Now hit create security group and go back refresh and add it to security group.

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 groups 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 rules 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

default
 sg-03833f0e3d40563f9 VPC: vpc-039948e0d48e3f441

icy-nlb-sg
 sg-0e21ed5621dcf596f VPC: vpc-039948e0d48e3f441

4. Listeners and routing - This is one of the most important step. Here we want to define for which kind of traffic where we want to forward them.(define the target group)

- First create a target group. Click on *Create target group*
- Choose target type - Instances (As we want our ec2 instances as targets)
- Give a suitable name for target group e.g. icy-ec2-tg
- Select protocol as TCP - as incoming traffic will be from TCP
- IP address type - IPv4
- VPC - if you've created other VPC select accordingly but i'll be going with default

- Health Check - Set them as HTTP

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.

Target group name

icy-ec2-t

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

TCP ▼

80

1-65535


IP address type

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

☒ **IPv4**

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

☐ **IPv6**

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#) 

VPC

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

-
vpc-039948e0d48e3f441
IPv4: 172.31.0.0/16 ▼

Health checks

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

Health check protocol

HTTP ▼

Health check path

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

/

Up to 1024 characters allowed.

► **Advanced health check settings**

You can also explore advance health check options, but i'll leave them as default. Now click on *Next*

- Now select which instance you want to target and click on *Include as pending below*

- Then review it once and click on *Create target group*

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)

Filter instances

<input type="checkbox"/>	Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
<input type="checkbox"/>	i-03487eb04fb112e90	icy-server-b	Running	launch-wizard-1	ap-south-1a	172.31.39.219	subnet-0760f17eee42a793d	February 11, 2024, 2
<input type="checkbox"/>	i-0dd77cda925ada228	icy-server-a	Running	launch-wizard-1	ap-south-1a	172.31.42.0	subnet-0760f17eee42a793d	February 11, 2024, 2

0 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

2 selections are now pending below. Include more or register targets when ready.

Review targets

Targets (2)

Filter targets

Show only pending

Remove all pending

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
i-03487eb04fb112e90	icy-server-b	80	Running	launch-wizard-1	ap-south-1a	172.31.39.219	subnet-0760f17eee42a793d	February 11, 2024, 22:04 (UTC+05:30)
i-0dd77cda925ada228	icy-server-a	80	Running	launch-wizard-1	ap-south-1a	172.31.42.0	subnet-0760f17eee42a793d	February 11, 2024, 22:02 (UTC+05:30)

2 pending

Cancel

Previous

Create target group

Now go back and select this TG that we created as target group-

Listeners and routing

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

Remove

Protocol

TCP

Port

80

1-65535

Default action

Info

Forward to

icy-ec2-t

Target type: Instance, IPv4

TCP

↺

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.

Add listener tag

You can add up to 50 more tags.

Add listener

At last scroll down at the bottom and click on *Create 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](#)

icy-nlb

- Internet-facing
- IPv4

Security groups [Edit](#)

- default
 - [sg-03833f0e3d40563f9](#)
- icy-nlb-sg
 - [sg-0e21ed5621dcf596f](#)

Network mapping [Edit](#)

VPC [vpc-039948e0d48e3f441](#)

- ap-south-1a
 - [subnet-0760f17eee42a793d](#)

Listeners and routing [Edit](#)

- TCP:80 defaults to [icy-ec2-t](#)

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.

Cancel

Create load balancer

Finally our NLB is created. But we have to make sure it is working fine, go to the target group tab and check health status of all the instance. If they all are healthy it's a good sign. If they are Unhealthy go to there SG and make sure all rules are logically correct and meet your goal.

EC2 > Target groups > icy-ec2-t

icy-ec2-t

Actions

Details

arn:aws:elasticloadbalancing:ap-south-1:471112644790:targetgroup/icy-ec2-t:ade1986f26f806ae

Target type

Instance

Protocol : Port

TCP: 80

VPC

[vpc-039948e0d48e3f441](#)

IP address type

IPv4

Load balancer

[icy-nlb](#)

Total targets

2

Healthy

2

Unhealthy

0

Unused

0

Initial

0

Draining

0

► Distribution of targets by Availability Zone (AZ)

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

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (2)

Filter targets

Refresh

Deregister

Register targets

< 1 >

Instance ID

Name

Port

Zone

Health status

Health status details

Launch time

[i-03487eb04fb112e90](#)

icy-server-b

80

ap-south-1a

Healthy

February 11, 2024, 22:04 (...)

[i-0dd77cda925ada228](#)

icy-server-a

80

ap-south-1a

Healthy

February 11, 2024, 22:02 (...)

8 / 11

Then also check status of our newly create NLB, make sure it's status is active -

EC2 > Load balancers > icy-nlb

icy-nlb

Actions

Details

Load balancer type

Network

Scheme

Internet-facing

Status

Active

Hosted zone

ZVDDRBQ08TROA

VPC

vpc-039948e0d48e3f441

Availability Zones

subnet-0760f17eee42a793d ap-south-1a (aps1-az1)

IP address type

IPv4

Date created

February 11, 2024, 23:14 (UTC+05:30)

Load balancer ARN

arn:aws:elasticloadbalancing:ap-south-1:471112644790:loadbalancer/net/icy-nlb/098298d62083fe48

DNS name

Info

icy-nlb-098298d62083fe48.elb.ap-south-1.amazonaws.com (A Record)

Listeners

Network mapping

Security

Monitoring

Integrations

Attributes

Tags

Listeners (1)

Actions

Add listener

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.

Filter listeners

< 1 >

Protocol:Port

Default action

ARN

Security policy

Default SSL/TLS certificate

ALPN policy

Tags

TCP:80

Forward to target group

icy-ec2-t

ARN

Not applicable

Not applicable

None

0 tags

Copy the DNS name and paste it in new tab. If it is running like this then you are all set -

This is server B

To check whether is targetting both instances we'll run a script to send continous request on our load balancer.

```
#!/bin/bash
nlwb="icy-nlb-098298d62083fe48.elb.ap-south-1.amazonaws.com"
for((i=0;i<=1000;i++))
do
    curl ${nlwb}
done
```

9 / 11

In this script we'll send 100 request to our NLB.

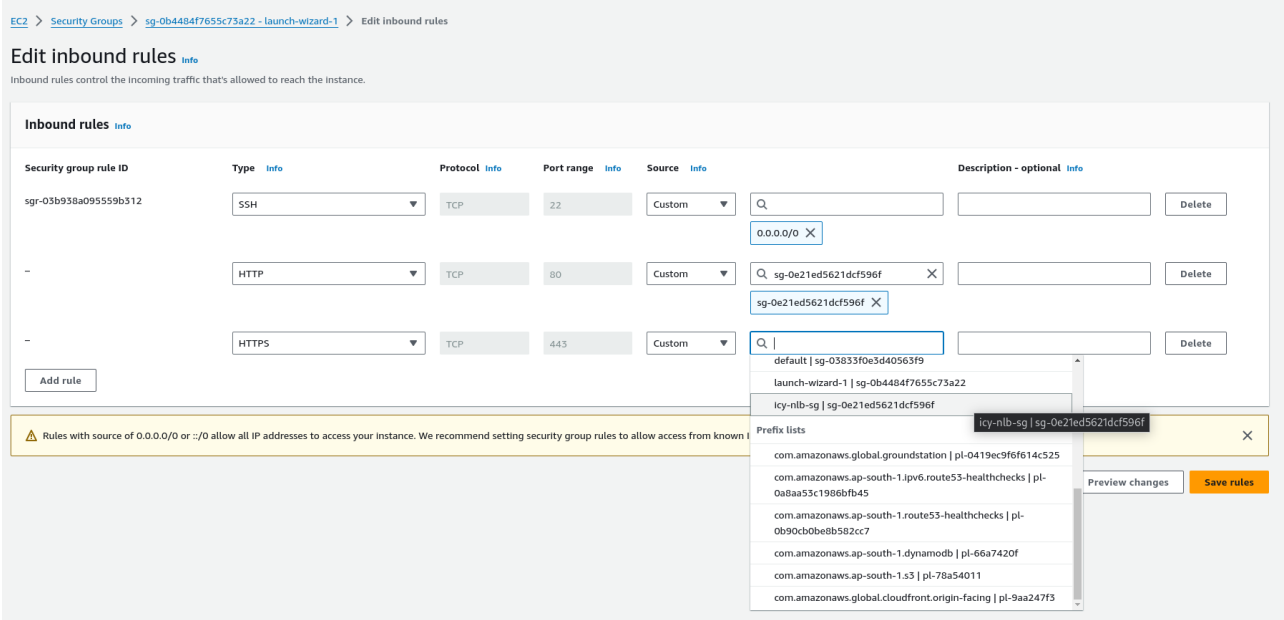
```
icyhotx@fedora:~/GitHub/Cloud-Computing-Course-work/Load Balancer for EC2$ bash load.sh
<h1>This is server A</h1>
<h1>This is server B</h1>
<h1>This is server A</h1>
<h1>This is server A</h1>
<h1>This is server A</h1>
<h1>This is server B</h1>
<h1>This is server A</h1>
<h1>This is server B</h1>
<h1>This is server B</h1>
<h1>This is server A</h1>
<h1>This is server A</h1>
<h1>This is server A</h1>
<h1>This is server A</h1>
<h1>This is server B</h1>
<h1>This is server A</h1>
<h1>This is server A</h1>
<h1>This is server B</h1>
<h1>This is server A</h1>
<h1>This is server B</h1>
<h1>This is server A</h1>
<h1>This is server B</h1>
<h1>This is server A</h1>
<h1>This is server B</h1>
```

You can see that with this our load balancer is working perfectly fine.

We still have one problem remaining. Our NLB is working fine but whenever if anyone try to access an ec2 instance with it's public IP, they are able to access it. We don't want that because then there won't be any point to create a Load Balancer if they are able to access our instance using public IPs.

To solve this problem we have to edit security group of our instance.

- Delete the existing rule for HTTP and HTTPS
- Add new rule for HTTP and HTTPS and set source as custom and set it to our NLB. (With this only our NLB will be able to access our instance and no other. Other can access them through NLB) Client -> NLB -> instanceA/instanceB



Here we can conclude that we've successfully created NLB in AWS for our EC2 instance.