

LOAD BALANCER

Introduction

AWS load balancers are a critical component of Amazon Web Services (AWS) that distribute incoming application traffic to multiple targets. This can improve the performance, reliability, and availability of applications hosted on AWS

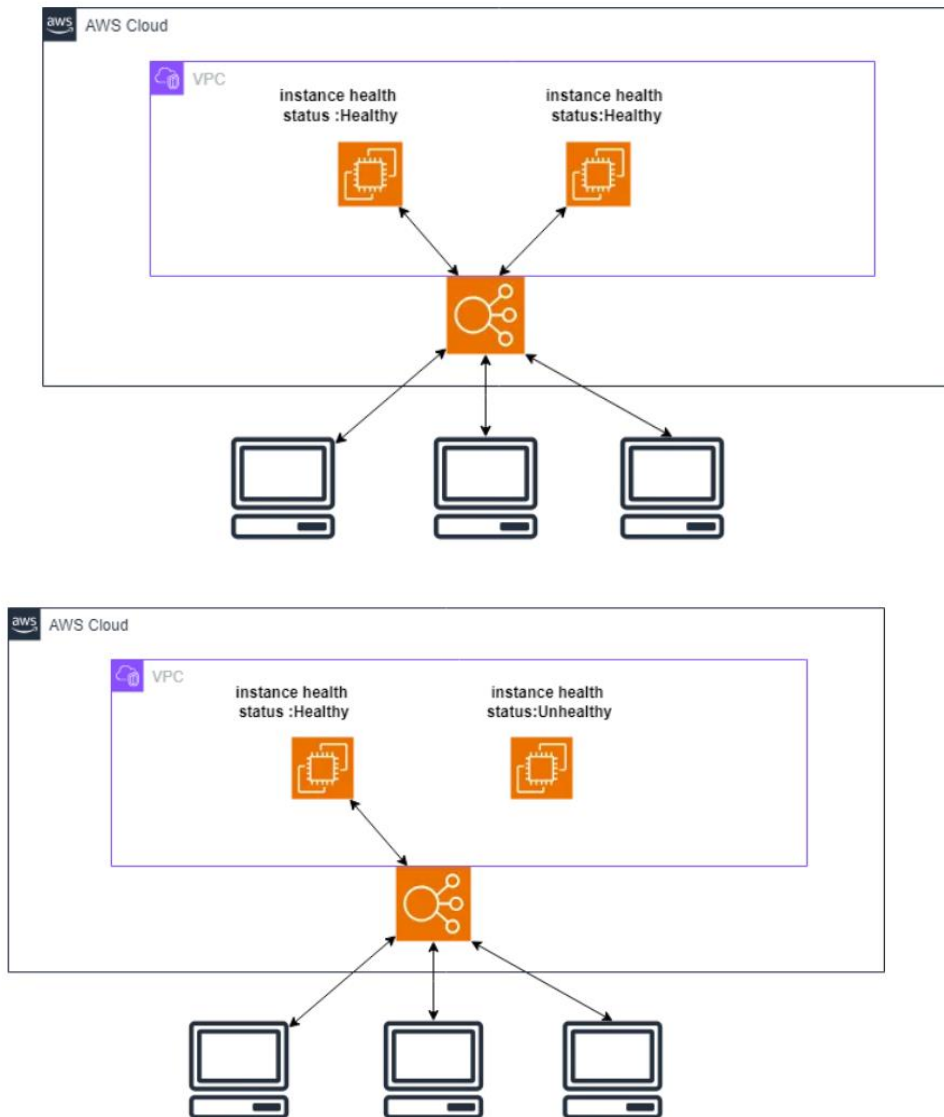
What is Elastic Load Balancing?

- ELB is a service that automatically distributes incoming application traffic and scale resources to meet traffic demands.
- ELB helps in adjusting capacity according to incoming application and network traffic.
- ELB can be enabled within a single availability zone or across multiple availability zones to maintain consistent application performance.

What are ELB features?

- Detection of unhealthy EC2 instances.
- Spreading EC2 instances across healthy channels only.
- Centralized management of SSL certificates.
- Optional public key authentication.
- Support for both IPv4 and IPv6.
- ELB accepts incoming traffic from clients and routes requests to its registered targets.
- When an unhealthy target or instance is detected, ELB stops routing traffic to it and resumes only when the instance is healthy again.
- ELB monitors the health of its registered targets and ensures that the traffic is routed only to healthy instances.
- ELB's are configured to accept incoming traffic by specifying one or more listeners. A listener is a process that checks for connection requests.
- Listeners are configured with a protocol and port number from the client to the ELB and vice-versa i.e., back from ELB to the client.

Architecture Diagram



What are ELB types of load balancers ?

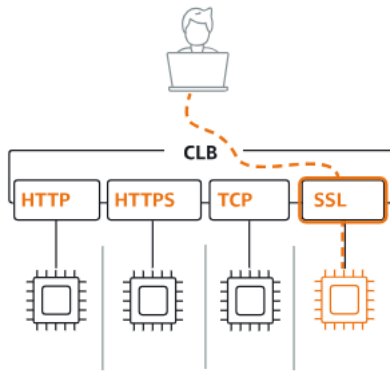
- Classic Load Balancer
- Application Load Balancer
- Network Load Balancer
- Gateway Load Balancer[Available in specific regions only]

Classic Load balancer

A Classic Load Balancer (CLB) is one of the earlier versions of load balancers provided by AWS under the Elastic Load Balancing (ELB) service.

If we have equal numbers of instances in both AZ then classic can be used.

Classic Load Balancer [Info](#)



Features of Classic Load Balancer:

Layer 4 and Layer 7 Support:

- Distributes traffic based on TCP/SSL (Layer 4) and HTTP/HTTPS (Layer 7) protocols.
- Can inspect and route HTTP headers for Layer 7 load balancing.

Basic Routing Capabilities:

- Routes traffic to EC2 instances based on their health and availability.
- No advanced features like content-based routing or host/path-based routing (available in ALBs).

Health Checks:

- Monitors the health of targets using TCP connections, HTTP requests, or HTTPS requests.
- Redirects traffic away from unhealthy instances.

SSL Termination:

- Can handle SSL certificates for HTTPS traffic, offloading the SSL processing from backend instances.

Static IPs:

- Does not support static IP addresses. Instead, it uses domain names that resolve to dynamic IPs.

Limited Scaling Capabilities:

- Automatically scales with the traffic load but has limitations compared to ALB or NLB in handling modern web application traffic.

EXECUTION

Launch the 2 ec2 instances with the httpd user data

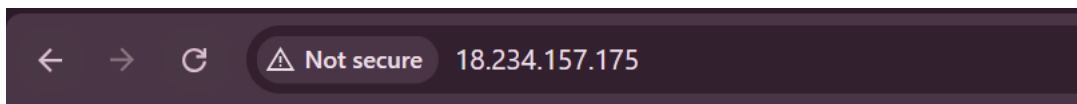
Find Instance by attribute or tag (case-sensitive)

All states

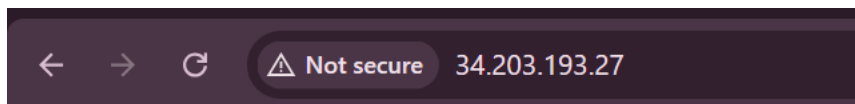
Instance state = running

Clear filters

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
<input type="checkbox"/>	Server-2	i-0f2b794fa623d9630	Running	t2.micro	2 Initializing	View alarms +	us-east-1b	ec2-34-203-193-27.co...	34.203.193.27
<input type="checkbox"/>	Server-1	i-0dc573120a02c077d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-18-234-157-175.co...	18.234.157.175



Welcome to the aws



Welcome to aws[2]

Now creating the classic Load balancer

- Go to the Loadbalancer[ec2 feature] in aws
- Select Classic loadbalancer

The Classic Load Balancer distributes incoming application traffic across multiple EC2 instance targets in multiple Availability Zones. This increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.

Load balancer name

ClassicLB

Scheme | Info

- Internet-facing

- ☐ Internal

- Serves internal traffic.
- Has private IP addresses.
- DNS name is publicly resolvable.

VPC | Info

<https://creativecommons.org/licenses/by-nc-sa/4.0/>



Availability Zones

- Subnet

▼

IPv4 address

Assigned by AWS

- Subnet

▼

IPv4 address

Assigned by AWS

Security groups



▼ Listener HTTP:80

Instance HTTP:80

Remove

Listener protocol

Listener port

1-65535

Instance protocol

Instance port

HTTP

80

1-65535

[Add listener](#)

Health checks [Info](#)

Your load balancer automatically performs health checks to test the availability of all registered instances. Traffic is only routed to healthy instances, which is determined on their response to the health check.

Ping target

The health check ping is sent using the protocol and port you specify. If using HTTP/HTTPS protocol, you must also provide the destination path.

Ping protocol	Ping port	Ping path
HTTP	80	/index.html
	1-65535	

► Advanced health check settings

Instances (2)

[Remove](#)[Add instances](#)

You can add instances to register as targets of the load balancer. Alternatively, after your load balancer is created, you can add it to an Amazon EC2 Auto Scaling group to ensure you maintain the correct number of instances to handle the load for your application. For maximum fault tolerance, we recommend maintaining approximately equivalent numbers of instances in each Availability Zone.

Filter instances

<

1

>

<input type="checkbox"/>	Instance ID	Name	State	Security groups	Zone
<input type="checkbox"/>	i-0f2b794fa623d9630	Server-2	Running	default	us-east-1b
<input type="checkbox"/>	i-0dc573120a02c077d	Server-1	Running	default	us-east-1b

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](#)

ClassicLB

- Internet-facing

Network mapping [Edit](#)

VPC [vpc-0ecc6f611c2c2d571](#)

- us-east-1a [subnet-09669c833e50484b0](#)
- us-east-1b [subnet-09b29c7107b9f6723](#)
- us-east-1d [subnet-0ae8c95671bf41520](#)

Security groups [Edit](#)

- default [sg-0241aa1cc34a39d5a](#)

Listeners and routing [Edit](#)

- HTTP:80

Health checks [Edit](#)

HTTP:80/index.html

- Timeout: 2 seconds
- Interval: 5 seconds
- Unhealthy threshold: 2
- Healthy threshold: 10

Instances [Edit](#)

2 instances added

- 2 instances in us-east-1b

Attributes [Edit](#)

- Cross-zone load balancing: On
- Connection draining: On
- Connection draining timeout: 300 seconds

Tags [Edit](#)

None

Next click “create”

Load balancers (1/1)

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

1 match

ClassicLB

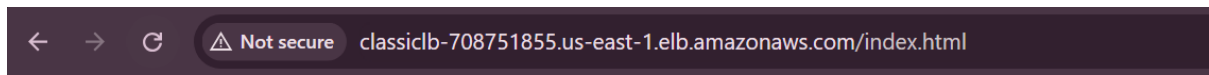
X

Clear filters

DNS name copied

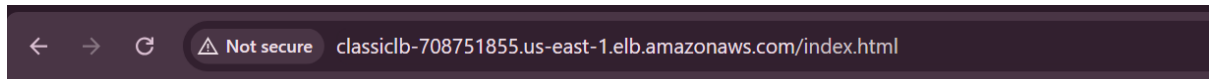
<input checked="" type="checkbox"/>	Name	State	VPC ID	Availability Zones	Type
<input checked="" type="checkbox"/>	<div>ClassicLB</div> <div> ClassicLB-708751855.us-ea... </div>	–	vpc-0ecc6f611c2c2d571	3 Availability Zones	classic

Copy the DNS name and paste it on the browser



Welcome to the aws

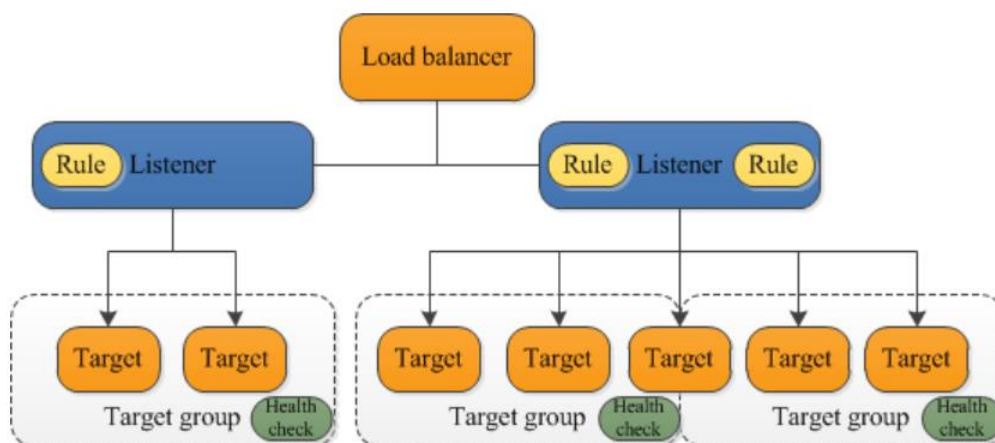
Refresh the browser



Welcome to aws[2]

APPLICATION LOAD BALANCER

- Application Load Balancer operates at layer 7 (application layer) and allows defining routing rules based on content across multiple services or containers running on one or more EC2 instances.
- scales the load balancer as traffic to the application changes over time.
- can scale to the vast majority of workloads automatically.
- supports health checks, used to monitor the health of registered targets so that the load balancer can send requests only to the healthy targets.



An Application Load Balancer functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model.

After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action.

You can configure listener rules to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups.

You can configure the routing algorithm used at the target group level. The default routing algorithm is round robin; alternatively, you can specify the least outstanding requests routing algorithm.

You can add and remove targets from your load balancer as your needs change, without disrupting the overall flow of requests to your application. Elastic Load Balancing scales your load balancer as traffic to your application changes over time. Elastic Load Balancing can scale to the vast majority of workloads automatically.

You can configure health checks, which are used to monitor the health of the registered targets so that the load balancer can send requests only to the healthy targets.

Benefits of migrating from a Classic Load Balancer

Using an Application Load Balancer instead of a Classic Load Balancer has the following benefits:

- Support for Path conditions. You can configure rules for your listener that forward requests based on the URL in the request. This enables you to structure your application as smaller services, and route requests to the correct service based on the content of the URL.
- Support for Host conditions. You can configure rules for your listener that forward requests based on the host field in the HTTP header. This enables you to route requests to multiple domains using a single load balancer.
- Support for routing based on fields in the request, such as HTTP header conditions and methods, query parameters, and source IP addresses.
- Support for routing requests to multiple applications on a single EC2 instance. You can register an instance or IP address with multiple target groups, each on a different port.
- Support for redirecting requests from one URL to another.

- Support for returning a custom HTTP response.
- Support for registering targets by IP address, including targets outside the VPC for the load balancer.
- Support for registering Lambda functions as targets.
- Support for the load balancer to authenticate users of your applications through their corporate or social identities before routing requests.
- Support for containerized applications. Amazon Elastic Container Service (Amazon ECS) can select an unused port when scheduling a task and register the task with a target group using this port. This enables you to make efficient use of your clusters.
- Support for monitoring the health of each service independently, as health checks are defined at the target group level and many CloudWatch metrics are reported at the target group level. Attaching a target group to an Auto Scaling group enables you to scale each service dynamically based on demand.
- Access logs contain additional information and are stored in compressed format.
- Improved load balancer performance.

EXECUTION

Created an application load balancer

ARC zonal shift for Application Load Balancers has changed

- Use of Amazon Application Recovery Controller (ARC) zonal shift now requires the Application Load Balancer attribute **ARC zonal shift integration** to be **enabled**.
- ARC zonal shift now supports cross-zone enabled Application Load Balancers.

Load balancers (1/1)

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

Filter load balancers

<input checked="" type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
<input checked="" type="checkbox"/>	ApplicationLB	ApplicationLB-132063633.u...	Active	vpc-0ecc6f611c2cd571	4 Availability Zones	application	December 15, 2024, 20:18 (UTC+05:30)

Attached target group

Load balancer: ApplicationLB

Details | **Listeners and rules** | Network mapping | Resource map - new | Security | Monitoring | Integrations | Attributes | Capacity - new | Tags

Listeners and rules (1)

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

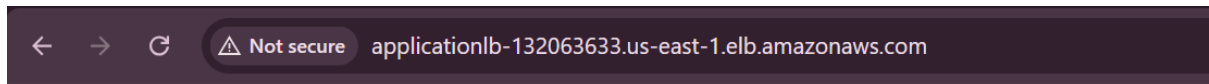
Filter listeners

<input type="checkbox"/>	Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS
<input type="checkbox"/>	HTTP:80	Forward to target group <ul style="list-style-type: none"> images: 1 (100%) Target group stickiness: Off 	1 rule	ARN	Not applicable	Not applicable	Not applicab

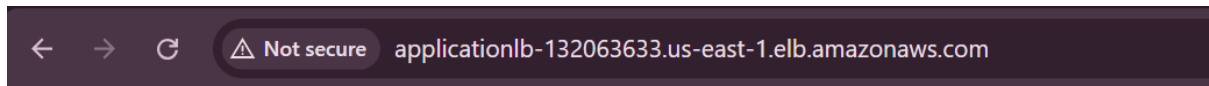
We can give path prefix also

```
[root@ip-172-31-45-138 html]# mkdir images
[root@ip-172-31-45-138 html]# ls
images  index.html
[root@ip-172-31-45-138 html]# cd images/
[root@ip-172-31-45-138 images]# vi index.html
[root@ip-172-31-45-138 images]#
```

Output

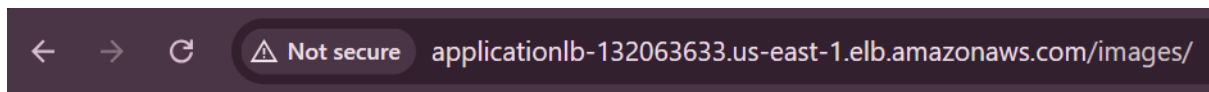


Welcome to aws[2]



Welcome to the aws

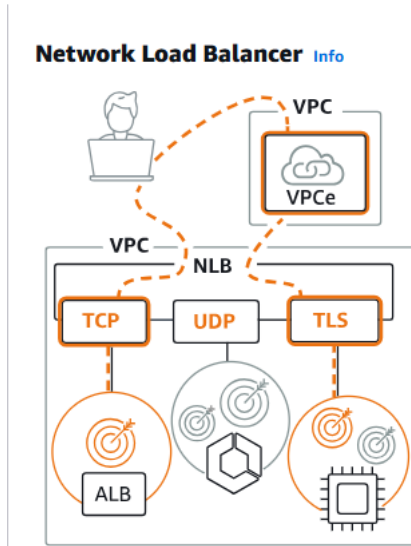
with the path prefix



Hello from Images!

NETWORK LOAD BALANCER

A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. It can handle millions of requests per second. After the load balancer receives a connection request, it selects a target from the target group for the default rule. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration



Network Load Balancer works on TCP/UDP Protocols

TCP/UDP Protocols works on the Transport layer that is Layer 4.

We can configure different Listeners.

It will work based on path prefix and rules configure .Target group will be attached to the load balancer, we can assign a elastic ip to NLB.

NLB will be used if we are not sure of hitting our application and if we are expecting sudden spikes in traffic then NLB can be used handle such situations.

EXECUTION

Created one Network Loadbalancer

Load balancers (1)

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

Filter load balancers

<input type="checkbox"/>	Name	<input checked="" type="checkbox"/> DNS name copied	State	VPC ID	Availability Zones	Type	Date created
<input type="checkbox"/>	Networkloadbalancer	Networkloadbalancer-fcda3...	Active	vpc-0ecc6f611c2c2d571	4 Availability Zones	network	December 15, 2024, 20:02 (UTC+05:30)

NLB

[Actions ▼](#)

Details

arn:aws:elasticloadbalancing:us-east-1:585008054730:targetgroup/NLB/143ed5731477cacb

Target type

Instance

Protocol : Port

TCP: 80

VPC

[vpc-0eccc6f611c2c2d571](#)

IP address type

IPv4

Load balancer

[Networkloadbalancer](#)

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)

[Deregister](#)[Register targets](#)

Filter targets

< 1 > ⚙

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status ...	Administrat...	Override de...	Launch time
<input type="checkbox"/>	i-0dc573120a0...	Server-1	80	us-east-1b (us...	✓ Healthy		⊖ No override	No override is ...	December 15, ...
<input type="checkbox"/>	i-0f2b794fa62...	Server-2	80	us-east-1b (us...	✓ Healthy		⊖ No override	No override is ...	December 15, ...

OUTPUT

← → ↻ ⚠ Not secure networkloadbalancer-fcda3761bcb7a9f9.elb.us-east-1.amazonaws.com

Welcome to the aws

← → ↻ ⚠ Not secure networkloadbalancer-fcda3761bcb7a9f9.elb.us-east-1.amazonaws.com

Welcome to aws[2]