



# Elastic Load Balancing

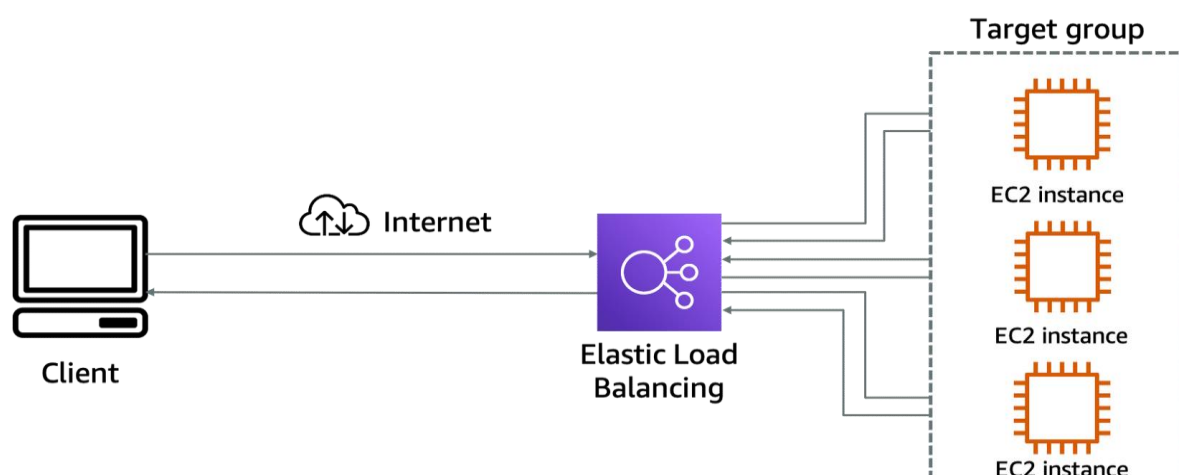
**The Elastic Load Balancing (ELB) service can distribute incoming application traffic across EC2 instances, containers, IP addresses, and Lambda functions.**



Load balancing refers to the process of distributing tasks across a set of resources. In the case of the Employee Directory application, the resources are EC2 instances that host the application, and the tasks are the requests being sent. You can use a load balancer to distribute the requests across all the servers hosting the application.

To do this, the load balancer needs to take all the traffic and redirect it to the backend servers based on an algorithm. The most popular algorithm is round robin, which sends the traffic to each server one after the other.

A typical request for an application starts from a client's browser. The request is sent to a load balancer. Then, it's sent to one of the EC2 instances that hosts the application. The return traffic goes back through the load balancer and back to the client's browser. Although it is possible to install your own software load balancing solution on EC2 instances, AWS provides the ELB service for you.





## ELB features

The ELB service provides a major advantage over using your own solution to do load balancing. Mainly, you don't need to manage or operate ELB. It can distribute incoming application traffic across EC2 instances, containers, IP addresses, and Lambda functions. Other key features include the following:

**Hybrid mode** – Because ELB can load balance to IP addresses, it can work in a hybrid mode, which means it also load balances to on-premises servers.

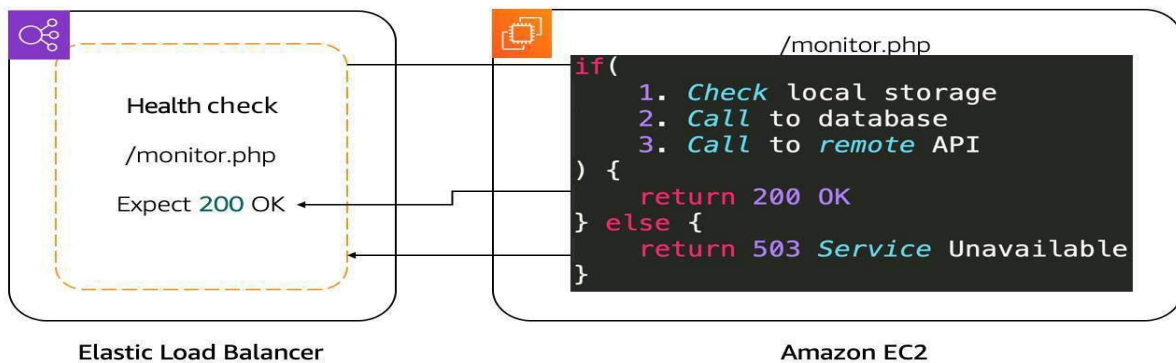
**High availability** – ELB is highly available. The only option you must ensure is that the load balancer's targets are deployed across multiple Availability Zones.

**Scalability** – In terms of scalability, ELB automatically scales to meet the demand of the incoming traffic. It handles the incoming traffic and sends it to your backend application.

## Health checks

Monitoring is an important part of load balancers because they should route traffic to only healthy EC2 instances. That's why ELB supports two types of health checks as follows:

- Establishing a connection to a backend EC2 instance using TCP and marking the instance as available if the connection is successful.
- Making an HTTP or HTTPS request to a webpage that you specify and validating that an HTTP response code is returned.
- Taking time to define an appropriate health check is critical. Only verifying that the port of an application is open doesn't mean that the application is working. It also doesn't mean that making a call to the home page of an application is the right way either.



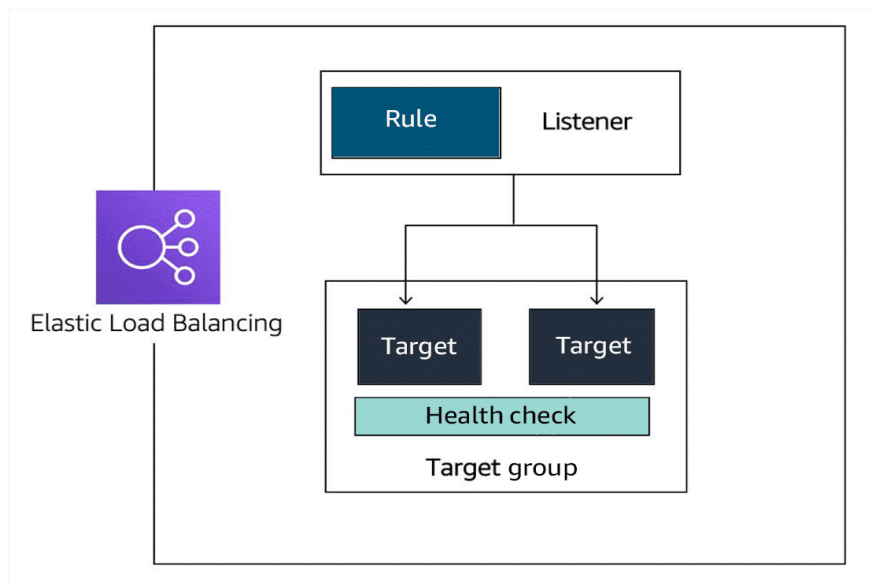
After determining the availability of a new EC2 instance, the load balancer starts sending traffic to it. If ELB determines that an EC2 instance is no longer working, it stops sending traffic to it and informs Amazon EC2 Auto Scaling. It is the responsibility of Amazon EC2 Auto Scaling to remove that instance from the group and replace it with a new EC2 instance. Traffic is only sent to the new instance if it passes the health check.

If Amazon EC2 Auto Scaling has a scaling policy that calls for a scale down action, it informs ELB that the EC2 instance will be terminated. ELB can prevent Amazon EC2 Auto Scaling from terminating an EC2 instance until all connections to the instance end. It also prevents any new connections.



## ELB components

The ELB service is made up of three main components: rules, listeners, and target groups.



### 1. Rule

To associate a target group to a listener, you must use a rule. Rules are made up of two conditions. The first condition is the source IP address of the client. The second condition decides which target group to send the traffic to.

### 2. Listener

The client connects to the listener. This is often called client side. To define a listener, a port must be provided in addition to the protocol, depending on the load balancer type.

### 3. Target group




The backend servers, or server side, are defined in one or more target groups. This is where you define the type of backend you want to direct traffic to, such as EC2 instances, Lambda functions, or IP addresses. Also, a health check must be defined for each target group.



## Types of load balancers

We will cover three types of load balancers: Application Load Balancer (ALB), Network Load Balancer (NLB), and Gateway Load Balancer (GLB).

Key points about each type:

 Application Load Balancer	<ul style="list-style-type: none"><li>- User authorization</li><li>- Rich metrics and logging</li><li>- Redirects</li><li>- Fixed response</li></ul>
 Network Load Balancer	<ul style="list-style-type: none"><li>- TCP and User Datagram Protocol (UDP) connection based</li><li>- Source IP preservation</li><li>- Low latency</li></ul>
 Gateway Load Balancer	<ul style="list-style-type: none"><li>- Health checks</li><li>- Gateway Load Balancer Endpoints</li><li>- Higher availability for third-party virtual appliances</li></ul>



## **1) Application Load Balancer**

For our Employee Directory application, we are using an Application Load Balancer. An Application Load Balancer functions at Layer 7 of the Open Systems Interconnection (OSI) model. It is ideal for load balancing HTTP and HTTPS traffic. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply. It then routes traffic to targets based on the request content.

Primary features:

- Routes traffic based on request data
- Sends responses directly to the client
- Uses TLS offloading
- Authenticates users
- Secures traffic
- Supports sticky sessions

## **2) Network Load Balancer**

A Network Load Balancer is ideal for load balancing TCP and UDP traffic. It functions at Layer 4 of the OSI model, routing connections from a target in the target group based on IP protocol data.

Primary features:

- Routes requests from the same client to the same target.
- Offers low latency for latency-sensitive applications.
- Preserves the client-side source IP address.
- Automatically provides a static IP address per Availability Zone (subnet).
- Lets users assign a custom, fixed IP address per Availability Zone (subnet).
- Uses Amazon Route 53 to direct traffic to load balancer nodes in other zones.



### **3) Gateway Load Balancer**

A Gateway Load Balancer helps you to deploy, scale, and manage your third-party appliances, such as firewalls, intrusion detection and prevention systems, and deep packet inspection systems. It provides a gateway for distributing traffic across multiple virtual appliances while scaling them up and down based on demand.

Primary Features:

- Ensures high availability and reliability by routing traffic through healthy virtual appliances.
- Can be monitored using CloudWatch metrics.
- Can deploy a new virtual appliance by selecting it in the AWS Marketplace.
- Connects internet gateways, virtual private clouds (VPCs), and other network resources over a private network.



## Selecting between ELB types

You can select between the ELB service types by determining which feature is required for your application. The following table presents a list of some of the major features of load balancers.

Feature	ALB	NLB	GLB
Load Balancer Type	Layer 7	Layer 4	Layer 3 gateway and Layer 4 load balancing
Target Type	IP, instance, Lambda	IP, instance, ALB	IP, instance
Protocol Listeners	HTTP, HTTPS	TCP, UDP, TLS	IP
Static IP and Elastic IP Address		Yes	
Preserve Source IP Address	Yes	Yes	Yes
Fixed Response	Yes		
User Authentication	Yes		