LOAD BALANCING

Introduction to scaling

Scaling refers to the ability of a system to handle increasing amounts of traffic or workload by distributing it across multiple resources, such as servers or network devices. Load balancing is a technique used to evenly distribute incoming network traffic or workload across multiple servers, ensuring optimal resource utilization, maximizing throughput, minimizing response time, and avoiding overload on any single server.

When a system scales, it means it can accommodate more users, requests, or data without sacrificing performance or reliability. There are two primary types of scaling in load balancing:

Vertical Scaling (Scale-Up): In vertical scaling, also known as scaling up, additional resources, such as CPU, memory, or storage, are added to a single server to handle increased load. This approach has limits, as there's only so much capacity a single server can handle, and it can become expensive or impractical beyond a certain point.

Horizontal Scaling (Scale-Out): Horizontal scaling involves adding more servers or instances to the existing infrastructure to distribute the load. This approach is often more cost-effective and scalable because it allows for the addition of resources incrementally as demand grows. Load balancers play a crucial role in horizontal scaling by evenly distributing incoming requests or traffic across multiple servers, ensuring that no single server is overwhelmed.

Load balancers use various algorithms to determine how to distribute traffic effectively, such as round-robin, least connections, IP hash, or weighted round-robin, among others. Additionally, load balancers can perform health checks on servers to ensure they're capable of handling requests and remove unhealthy servers from the pool to prevent them from receiving traffic until they recover.

Overall, scaling in the concept of load balancing is essential for ensuring high availability, reliability, and performance of web applications, services, or any system that experiences varying levels of demand over time.

ELASTIC LOAD BALANCERS

ELB stands for Elastic Load Balancing, which is a service provided by Amazon Web Services (AWS) for distributing incoming application or network traffic across multiple targets, such as EC2 instances, containers, IP addresses, or Lambda functions. ELB automatically scales its load-balancing capacity based on incoming traffic, providing high availability and fault tolerance for applications.

There are four main types of Load Balancers offered by AWS:

Classic Load Balancer (CLB):

The Classic Load Balancer provides basic load-balancing functionality and is ideal for applications that were built within the EC2-Classic network. It operates at both the application and network layers of the OSI model and distributes traffic across multiple EC2 instances.

Application Load Balancer (ALB):

The Application Load Balancer operates at the application layer (Layer 7) of the OSI model and is optimized to handle HTTP and HTTPS traffic. It supports advanced features such as content-based routing, host-based routing, and path-based routing, making it suitable for modern web applications with multiple microservices.

Network Load Balancer (NLB):

The Network Load Balancer operates at the transport layer (Layer 4) of the OSI model and is designed to handle high-throughput, low-latency traffic, including TCP, UDP, and TLS traffic. It is capable of handling millions of requests per second and is commonly used for high-performance applications and services.

Gateway Load Balancers (GWLB):

GWLB in AWS are a unique type of load balancer designed to simplify the deployment, scaling, and management of virtual appliances such as firewalls, intrusion detection and prevention systems, and deep packet inspection systems. They provide a way to transparently deploy, scale, and manage virtual appliances while maintaining high availability and fault tolerance.

ELB automatically distributes incoming traffic across multiple Availability Zones within a single region to ensure high availability and fault tolerance. It also performs health checks on its registered targets and routes traffic only to healthy instances, helping to maintain the availability and reliability of applications.

Overall, Elastic Load Balancing simplifies the process of deploying and managing scalable and highly available applications on AWS by providing a flexible and reliable load-balancing solution.

What's the difference between application, network, and gateway load balancing?

Application load balancer (ALB), network load balancer (NLB), and gateway load balancer (GLB) are three types of load balancers used in the cloud. Load balancing is the process of distributing network traffic equally across a pool of resources supporting an application. Modern applications

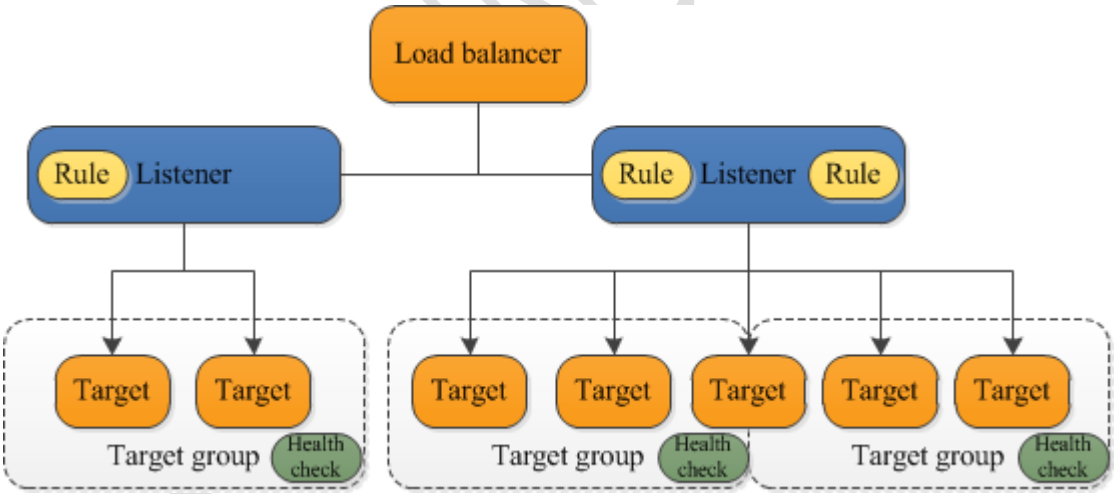
process millions of users simultaneously. These high-traffic volumes require many resource servers with duplicate data. To redirect application traffic, ALBs examine the requested content, such as HTTP headers or SSL session IDs. NLBs examine IP addresses and other network information to redirect traffic optimally. GLBs act as a transparent network gateway (a single entry and exit point for all traffic) and distribute traffic while scaling your virtual appliances with the demand.

How an application load balancer works

ALBs distribute incoming traffic across multiple targets, such as EC2 instances. For example, an ecommerce application has a product directory, a shopping cart, and checkout functions. The ALB sends requests for browsing products to servers that contain images and videos but don't need to maintain open connections. By comparison, it sends shopping cart requests to servers that maintain many client connections and save cart data for a long time.

The ALB has a listener component that checks for connection requests from clients. You can define rules for a listener that determine how the load balancer routes requests to its registered targets. A target group sorts registered targets into groups. You can define rules to route common traffic to an entire group. For example, you can create a target group for general requests and other target groups for requests to the microservices for your application.

The following diagram shows how an ALB works.



How a network load balancer works

NLBs distribute traffic based on network conditions. For example, if you have multiple database servers with duplicate data, the NLB routes traffic based on predetermined server IP addresses or server availability.

The NLB monitors the health of its registered targets and routes traffic only to the healthy targets. 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. Each individual TCP connection is routed to a single target for the life of the connection. Similarly, you can also route a UDP flow consistently to a single target throughout its lifetime.

How a gateway load balancer works

With a GLB, you can deploy, manage, and scale virtual appliances, such as intrusion detection and prevention, firewalls, and deep packet inspection systems. It creates a single entry and exit point for all appliance traffic and scales your virtual appliances with demand. You can also use it to exchange traffic across virtual private cloud (VPC) boundaries.

In the GLB, you establish rules using route tables. Depending on the rules that you set up, it selects different target groups to forward traffic to. It receives IP packets and forwards traffic to specific target groups.

Difference Between Application, Network and Gateway Load Balancers

	Application load balancer (ALB)	**Network load balancer (NLB)**	**Gateway load balancer (GLB)**
OSI layer	Works on layer 7, the application layer	Works on layer 4, the transport layer	Works on the network layer, layer 3, and layer 7.
Target types	Works with IP, instance, and lambda target types	Works with IP, instance, and ALB target types	Works with IP and instance target types
Proxy behavior	Ends connection	Ends connection	Doesn't terminate the flow
Protocols	Supports HTTP, HTTPS, and gRPC protocols	Supports TCP, UDP, and TLS protocols	Supports IP-based routing
Algorithms	Round-robin.	Flow hash	Routing table lookup

When to use these Load Balancers

An ALB (Application Load Balancers) is a good choice when you need flexible application-level traffic management and routing. It's best with microservices, containerized environments, and

web applications. Its features—such SSL termination, session persistence, and content-based routing—enable it to offer assistance with complex routing scenarios.

NLB (Network Load Balancers) is best for high-performance, low-latency, and scalable network-level balancing. Applications that distribute traffic on the transport layer use NLBs, especially considering its reliability. Gaming systems, media streaming services, and major IoT systems use NLBs.

GLB(Gateway Load Balancers) is ideal when you're balancing on the network gateway level. For example, a GLB works well if you manage traffic between cloud and on-premises environments or across different regions. Because it combines OSI layers 3 and 4 balancing, it can route traffic between distinct regions and networks. Because it supports IP-based routing, it can distribute traffic across virtual gateways, so it can offer high scalability and availability.

Components of load balancing

Load balancing involves distributing incoming network traffic or workload across multiple servers, ensuring optimal resource utilization, maximizing throughput, minimizing response time, and avoiding overload on any single server. Here are the key components and types of load balancing:

Components of Load Balancing:

Client: The client initiates a connection request to the load balancer to access a service or application.

Load Balancer: The load balancer sits between the client and the backend servers, intercepting incoming traffic and distributing it across multiple servers based on predefined algorithms and configurations.

Backend Servers: These are the servers that actually handle the client requests. They host the services or applications being accessed by the clients.

Health Checks: Load balancers periodically check the health of backend servers to ensure they are capable of handling requests. Unhealthy servers may be temporarily removed from the pool to prevent them from receiving traffic until they recover.