ALL ABOUT ELB AND IT'S TYPE

ELB in AWS stands for Elastic Load Balancing. It is a fully managed service that automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and more. The purpose of ELB is to ensure that no single server or resource gets overwhelmed with traffic, improving the availability and fault tolerance of applications.

Key Functions of ELB:

1. Traffic Distribution: ELB distributes incoming traffic across multiple backend servers (EC2 instances) to ensure even load distribution.
2. High Availability: It helps ensure that your application is always available, even if some servers fail, by rerouting traffic to healthy servers.
3. Scalability: ELB can automatically scale to handle increasing or decreasing traffic without requiring manual intervention.
4. Fault Tolerance: If one of your application instances becomes unhealthy, ELB can automatically reroute traffic to healthy instances.

Types of ELB in AWS:

1. Application Load Balancer (ALB):
    - Best for HTTP and HTTPS traffic.
    - Operates at the application layer (Layer 7 of the OSI model).
    - Supports advanced routing features like host-based and path-based routing.
2. Network Load Balancer (NLB):
    - Best for handling high-throughput, low-latency traffic.
    - Operates at the transport layer (Layer 4).
    - Handles millions of requests per second and supports static IP addresses.

3. Classic Load Balancer (CLB):
    - A legacy load balancer that can handle both Layer 4 and Layer 7 traffic.
    - AWS recommends using ALB or NLB instead of CLB for most use cases.
4. **Gateway Load Balancer (GLB)**
    - A load balancer that can handle Layer 3 traffic.
    - Layer 3 known as NETWORK Layer, follow the protocol GENEVE.
    - Use for tunneling when create tunnel.

APPLICATION LOAD BALANCER

ALB stands for **Application Load Balancer**, and it is one of the types of load balancers provided by **Elastic Load Balancing (ELB)** in AWS. ALB is designed to distribute HTTP and HTTPS traffic at the **application layer** (Layer 7 of the OSI model), making it ideal for modern web applications that require advanced routing features and fine-grained control over how requests are distributed.

Key Features and Benefits of ALB:

1. **Content-Based Routing:**
    - **Host-based routing:** ALB can route traffic based on the domain name in the request (e.g., www.example.com vs. api.example.com).
    - **Path-based routing:** ALB can route traffic based on the URL path (e.g., /api can go to one set of instances and /web to another).
2. **SSL/TLS Termination:**
    - ALB can offload SSL/TLS encryption/decryption (HTTPS), reducing the load on backend instances and improving performance.
3. **WebSocket Support:**

- ALB supports WebSocket connections, making it suitable for real-time applications like chat apps and live updates.

4. **Containerized Application Support:**
    - ALB integrates seamlessly with Amazon ECS (Elastic Container Service) and EKS (Elastic Kubernetes Service), making it easy to route traffic to containerized applications.
5. **Dynamic Target Registration:**
    - Targets (such as EC2 instances or containers) can be dynamically registered and deregistered based on health checks and scaling events.
6. **Advanced Request Routing:**
    - ALB supports routing based on specific HTTP headers, query parameters, or other request properties, enabling more sophisticated routing logic.
7. **Automatic Scaling:**
    - ALB scales automatically to handle varying amounts of incoming traffic, ensuring that applications can handle increased load without manual intervention.
8. **Improved Security:**
    - ALB can integrate with AWS WAF (Web Application Firewall) to protect against common web exploits and attacks.
    - Supports security groups and access control lists to restrict access to the load balancer.
9. **Health Checks:**
    - ALB can continuously monitor the health of backend instances and route traffic only to healthy targets.

Use Cases for ALB:

- **Microservices Architectures:** ALB can route traffic to different microservices based on the URL path, making it ideal for microservices-based applications.
- **Dynamic Content Delivery:** Applications that need to deliver different content types or perform user-specific routing can benefit from ALB's advanced routing capabilities.
- **Containerized Applications:** ALB works well with Amazon ECS and EKS for distributing traffic to containers.
- **Real-time Web Applications:** WebSocket support makes ALB suitable for real-time communication applications.

Comparison with Other Load Balancers:

- **ALB** vs **NLB (Network Load Balancer):** While ALB is application-layer focused (HTTP/HTTPS traffic, Layer 7), **NLB** operates at the transport layer (Layer 4) and is better suited for high-throughput, low-latency needs (such as TCP traffic).
- **ALB** vs **CLB (Classic Load Balancer):** ALB provides more advanced routing and better support for modern web applications. CLB is a legacy option and is generally less flexible than ALB.

Conclusion:

Application Load Balancer (ALB) is a powerful, flexible load balancing solution in AWS for routing HTTP and HTTPS traffic at the application layer. It is especially useful for complex, modern applications that require intelligent routing, support for containers, and the ability to scale automatically with traffic.

NETWORK LOAD BALANCER

NLB stands for **Network Load Balancer**, and it is another type of load balancer offered by **Elastic Load Balancing (ELB)** in AWS. NLB operates at the **transport layer** (Layer 4) of the OSI model, which makes it capable of handling high-throughput, low-latency traffic, and is ideal for use cases where performance is critical.

Key Features and Benefits of NLB:

1. **High Performance & Low Latency:**
    - NLB is designed to handle millions of requests per second while maintaining ultra-low latency. This makes it suitable for applications requiring high performance, such as gaming, IoT (Internet of Things), or financial services.
2. **Handles TCP and UDP Traffic:**
    - NLB supports both **TCP** and **UDP** traffic, which makes it a good choice for non-HTTP-based protocols. For example, it can be used for applications running over database connections or real-time applications like VoIP (Voice over IP).
3. **Static IP Address:**
    - NLB can provide a **static IP address** for your application, which is useful if you need a fixed IP that clients can use to connect to your service. Additionally, it can support **Elastic IPs** (EIP) for better network management and accessibility.
4. **Cross-Zone Load Balancing:**
    - NLB supports **cross-zone load balancing**, allowing traffic to be evenly distributed across targets in different availability zones (AZs). This increases the resiliency and availability of the service.
5. **Health Checks:**
    - NLB supports health checks to monitor the health of registered targets (like EC2 instances). If a target becomes unhealthy, traffic is automatically routed to healthy instances.
6. **TLS Termination:**
    - NLB supports **TLS (Transport Layer Security) termination**, meaning it can decrypt incoming encrypted traffic before forwarding it to backend instances, which helps offload some of the encryption tasks from your backend resources.

7. **Scalability:**
    - NLB is highly scalable and can automatically adjust to fluctuations in incoming traffic, handling sudden spikes in load without requiring manual intervention.
8. **Integration with AWS Services:**
    - NLB works seamlessly with other AWS services, such as **Auto Scaling**, **Amazon EC2**, and **Amazon ECS**, providing better flexibility and control over your infrastructure.
9. **Security Features:**
    - NLB integrates with **Security Groups** and **Access Control Lists (ACLs)** to control access to your load balancer and backend resources.
    - It also supports **AWS WAF** (Web Application Firewall) for security at the application layer if necessary.

Use Cases for NLB:

- **High-Throughput Applications:** NLB is ideal for applications that require large-scale traffic handling and high-performance networking, such as gaming platforms, media streaming, and financial services.
- **Non-HTTP Traffic:** Since NLB supports both TCP and UDP, it is well-suited for applications using protocols other than HTTP/HTTPS, such as database connections (e.g., MySQL, PostgreSQL) or real-time communication.
- **IoT (Internet of Things):** NLB can efficiently distribute traffic for IoT applications, where devices might generate massive amounts of data that need to be processed by backend servers.
- **VPN and Internal Applications:** It can be used for VPN (Virtual Private Network) connections or for internal services requiring fast, reliable network traffic distribution.

NLB vs ALB:

- **NLB (Network Load Balancer)** operates at **Layer 4 (Transport Layer)** and is better for **non-HTTP traffic** (e.g., TCP, UDP), providing ultra-low latency and high performance.
- **ALB (Application Load Balancer)** operates at **Layer 7 (Application Layer)** and is best for **HTTP/HTTPS traffic**, offering advanced features such as content-based routing, WebSocket support, and SSL termination.

Conclusion:

The **Network Load Balancer (NLB)** is a robust, high-performance load balancing solution suitable for handling large-scale, low-latency, and high-throughput traffic. It is ideal for applications that need to manage non-HTTP traffic, such as database connections, real-time communications, and network-intensive services. It also provides high availability, scalability, and enhanced security features for a variety of workloads in AWS.

CLASSIC LOAD BALANCER

CLB stands for Classic Load Balancer, which is the legacy load balancing solution provided by Elastic Load Balancing (ELB) in AWS. CLB operates at both the application layer (Layer 7) and the transport layer (Layer 4) of the OSI model, making it a versatile, albeit older, choice for load balancing traffic.

Key Features of Classic Load Balancer (CLB):

1. Layer 4 and Layer 7 Support:
    - CLB can handle HTTP/HTTPS traffic (Layer 7) as well as TCP traffic (Layer 4). This allows it to support a wide range of use cases.
2. Basic Load Balancing:
    - CLB is designed for simple load balancing, particularly in earlier AWS environments. While it is functional, it lacks the advanced routing capabilities that newer load balancers, like Application Load Balancer (ALB) or Network Load Balancer (NLB), offer.

3. Sticky Sessions (Session Persistence):
    - CLB supports sticky sessions, which means it can route requests from the same client to the same backend instance to maintain session continuity. This is useful for stateful applications.
4. SSL Termination:
    - CLB can perform SSL termination, meaning it can handle the decryption of HTTPS traffic, offloading the work from backend instances.
5. Health Checks:
    - CLB offers health checks to monitor the health of the instances behind the load balancer. If an instance is found to be unhealthy, the traffic will be redirected to healthy instances.
6. Automatic Scaling:
    - Like other AWS load balancers, CLB integrates with Auto Scaling, automatically adjusting the number of instances based on the traffic load.

When to Use CLB:

CLB is suitable for simpler use cases or applications that:

- Are using legacy setups.
- Do not require advanced routing (such as host-based or path-based routing).
- Rely on older AWS architecture or tools.
- Handle basic HTTP, HTTPS, or TCP traffic without needing high performance or complex load balancing logic.

Comparison with ALB and NLB:

- Classic Load Balancer (CLB) is being phased out in favor of the more feature-rich Application Load Balancer (ALB) and Network Load Balancer (NLB).

- ALB is more appropriate for modern web applications with complex routing requirements (e.g., microservices, containerized applications), operating at the application layer (Layer 7).
- NLB is best for high-performance needs, especially for non-HTTP protocols like TCP or UDP, and operates at the transport layer (Layer 4).

Why Use CLB Today?

- Although AWS encourages using ALB and NLB for most new applications, CLB is still functional and can be used for legacy systems that haven't migrated to newer load balancing solutions.
- Some users with simpler needs, such as straightforward web applications that don't require advanced routing, may still find CLB adequate.

Conclusion:

While Classic Load Balancer (CLB) served as the first ELB option for AWS customers, it is now considered a legacy service with limited features compared to ALB and NLB. AWS recommends using ALB or NLB for new workloads, as they provide more advanced capabilities, better performance, and more flexibility for modern application architectures.

GATEWAY LOAD BALANCER

Gateway Load Balancer (GLB) is a service in AWS that allows you to deploy, scale, and manage third-party virtual appliances like firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS) at scale in your virtual private cloud (VPC). It is specifically designed to enable customers to integrate third-party network appliances for high-performance traffic inspection and management, without the need to manually manage traffic routing.

Key Features and Benefits of Gateway Load Balancer:

1. **Traffic Steering for Virtual Appliances:**

- **Gateway Load Balancer** is used to direct traffic to virtual appliances (such as firewalls, threat detection, or monitoring systems) deployed in your VPC. It operates at the **network layer (Layer 3)** and provides a seamless way to handle traffic through these appliances without complex configuration.

2. **Easy Integration with Third-Party Appliances:**
    - GLB allows you to easily deploy and scale third-party appliances like firewalls, deep packet inspection tools, and other security services in your AWS environment. It integrates directly with **AWS Marketplace** appliances, simplifying the deployment process.
3. **Scalability and Flexibility:**
    - GLB automatically scales to accommodate high levels of traffic and adjusts the number of appliance instances as needed, ensuring that performance remains high even during traffic spikes.
4. **Transparent Traffic Flow:**
    - GLB makes it easy to insert network appliances into the traffic path. It acts as a transparent gateway that inspects, processes, and forwards the traffic without disrupting the overall network flow.
5. **Support for Stateful and Stateless Appliances:**
    - It works with both **stateful** appliances (which maintain session information) and **stateless** appliances (which don't track session states), making it flexible for different types of workloads.
6. **Simplified Architecture:**
    - GLB simplifies traffic routing by automatically handling tasks like managing IP addresses, scaling, and monitoring. This reduces the operational complexity compared to manually setting up custom routing configurations for appliances.
7. **High Availability and Fault Tolerance:**
    - Gateway Load Balancer provides **high availability** by distributing traffic across multiple instances of your virtual appliances and

supporting failover capabilities. If one appliance becomes unavailable, traffic is automatically routed to a healthy appliance.

8. **Health Checks for Appliances:**
    - GLB performs **health checks** on your appliances to ensure that traffic is only forwarded to healthy instances. If an appliance is detected to be unhealthy, the load balancer will automatically redirect traffic to a healthy one.
9. **Cost Efficiency:**
    - By using GLB, you avoid over-provisioning your virtual appliances, as the load balancer can dynamically scale based on traffic demand. This reduces costs by ensuring you're using only the necessary resources.

How Gateway Load Balancer Works:

1. **Traffic Insertion**:
    - GLB is placed at the edge of your VPC and directs traffic to appliances via a **Target Group**. The load balancer ensures that traffic is correctly steered to the appropriate appliance.
2. **Health Monitoring**:
    - GLB continuously monitors the health of the appliances using health checks. If an appliance fails, traffic is automatically routed to a healthy instance, ensuring minimal disruption.
3. **Forwarding Traffic**:
    - After processing by the appliance (such as a firewall or IDS), the traffic is then forwarded back into the VPC to its intended destination.
4. **Traffic Inspection and Logging**:
    - Appliances can inspect and log the traffic before it reaches its final destination, allowing you to maintain network security, perform packet analysis, or implement other security measures.

Use Cases for Gateway Load Balancer:

1. **Security Appliance Integration:**
    - When you need to insert third-party security appliances, such as **firewalls**, **IDS/IPS**, or **Web Application Firewalls (WAFs)** into your traffic flow, GLB provides a simple way to scale and manage these services.
2. **Traffic Inspection at Scale:**
    - For applications or workloads requiring deep packet inspection or advanced threat detection, GLB can efficiently distribute traffic to appliances that provide these services.
3. **High-Performance Network Security:**
    - Organizations with high-performance security requirements, such as enterprises or financial institutions, can use GLB to handle large volumes of traffic while ensuring that it is thoroughly inspected by security appliances.
4. **Regulatory Compliance and Monitoring:**
    - Many industries (e.g., finance, healthcare) require traffic inspection for compliance reasons. GLB can route traffic through appliances that perform logging, monitoring, or other security checks necessary to meet regulatory requirements.