**ELB - Elastic Load Balancer Learning Material**

**1. ELB Service Introduction**  
Amazon **Elastic Load Balancer (ELB)** automatically distributes incoming traffic across multiple EC2 instances to ensure that no single instance is overwhelmed. It helps improve the availability and fault tolerance of your application by ensuring that traffic is routed to healthy instances.

**Key Benefits of ELB**:

* **Scalability**: Automatically adjusts to handle varying amounts of traffic.
* **Fault Tolerance**: Distributes traffic only to healthy instances, ensuring high availability.
* **Flexible**: Supports both HTTP/HTTPS and TCP traffic.

ELB is essential for applications that need to handle large amounts of traffic or ensure high availability.

**2. Types of Load Balancers**  
AWS offers three types of load balancers in the ELB family, each designed for specific use cases:

* **Application Load Balancer (ALB)**:
  + **Layer 7** (Application Layer) load balancer.
  + Routes traffic based on HTTP/HTTPS requests, such as URL paths or host headers.
  + Ideal for web applications and microservices.
* **Network Load Balancer (NLB)**:
  + **Layer 4** (Transport Layer) load balancer.
  + Routes traffic based on IP protocol data, such as TCP or UDP.
  + Ideal for high-performance applications that require low latency and high throughput.
  + Can handle millions of requests per second and maintain high availability.
  + Best suited for load balancing of non-HTTP(S) traffic, such as gaming applications or IoT.
* **Classic Load Balancer (CLB)**:
  + This is the older generation load balancer.
  + Works at both Layer 4 and Layer 7, but does not have all the advanced features of ALB and NLB.
  + Useful for applications that need basic load balancing, typically used for legacy applications.

**3. User Data Concept**  
**User Data** is a feature that allows you to specify a script or command to be executed when an EC2 instance is launched. This can be used to automate tasks like installing software, configuring the server, or setting up services, all during the instance initialization phase.

* **Example Use Cases**:
  + Automatically installing Apache or Nginx on a web server when the instance starts.
  + Configuring environment variables or application settings when the instance boots.

You can define user data as a shell script or cloud-init configuration during EC2 instance launch. For example, a simple shell script can be provided as user data:

#!/bin/bash

yum install -y httpd

service httpd start

This script installs and starts the Apache web server.

**4. Application Load Balancer (ALB)**  
The **Application Load Balancer (ALB)** is used for HTTP/HTTPS traffic and provides advanced routing capabilities.

* **Key Features**:
  + **Host-based Routing**: Routes traffic to different EC2 instances based on the host name (e.g., api.example.com or www.example.com).
  + **Path-based Routing**: Routes traffic based on URL paths (e.g., /images or /products).

**How ALB Works**:

* A client sends an HTTP request to the load balancer.
* The ALB examines the URL and based on the routing rules, it directs the request to the appropriate EC2 instance.
* The backend EC2 instance processes the request and returns the response to the client through the load balancer.

**5. Network Load Balancer (NLB)**  
The **Network Load Balancer (NLB)** is used for routing traffic at the **Layer 4** (Transport Layer), making it suitable for handling high-performance, low-latency applications.

* **Key Features**:
  + **TCP/UDP Load Balancing**: Routes non-HTTP traffic such as TCP or UDP requests.
  + **High Performance**: Supports millions of requests per second, ideal for applications requiring low latency.
  + **Static IP**: NLB provides a static IP address for your application, making it easier to manage network traffic.

**How NLB Works**:

* NLB directs traffic based on IP protocol data, such as TCP or UDP.
* It is designed for applications that require very fast and high-throughput connections, such as gaming, real-time applications, or IoT devices.

**Key Differences Between ALB and NLB:**

| **Feature** | **Application Load Balancer (ALB)** | **Network Load Balancer (NLB)** |
| --- | --- | --- |
| **Layer** | Layer 7 (Application Layer) | Layer 4 (Transport Layer) |
| **Protocol** | HTTP, HTTPS | TCP, UDP |
| **Routing Type** | Host-based, Path-based | IP-based routing (TCP/UDP) |
| **Use Case** | Web applications, microservices | High-performance, low-latency apps |
| **Performance** | Optimized for HTTP(S) traffic | Optimized for high throughput |
|  |  |  |

This material provides a solid understanding of **Elastic Load Balancer (ELB)**, including the different types of load balancers (ALB, NLB), user data configuration, and how they work to distribute traffic across multiple EC2 instances.

**Hands-On**

In this section, students will practice configuring and using ELB:

1. **Create an Application Load Balancer**:
   * Create a new ALB, configure listener rules, and attach EC2 instances.
   * Test host-based and path-based routing.
2. **Create a Network Load Balancer**:
   * Create a new NLB, configure listeners for TCP/UDP traffic, and attach EC2 instances.
   * Test high-performance traffic routing.
3. **Set Up User Data on EC2 Instances**:
   * Launch EC2 instances with user data scripts to automate software installation and configuration.