**In AWS, a Load Balancer automatically distributes incoming traffic across multiple targets (like EC2 instances, containers, IP addresses) to ensure:**

* **High availability**
* **Fault tolerance**
* **Scalability**

**🧱 Types of Load Balancers in AWS (via ELB)**

1. **Application Load Balancer (ALB)**
   * **Works at Layer 7 (HTTP/HTTPS)**
   * **Smart routing: based on URL, host, headers, etc.**
   * **Best for web apps & microservices**
2. **Network Load Balancer (NLB)**
   * **Works at Layer 4 (TCP/UDP)**
   * **Super fast, handles millions of requests per second**
   * **Best for gaming, real-time apps, or low-latency workloads**
3. **Gateway Load Balancer (GWLB)**
   * **Works at Layer 3 (IP)**
   * **Used with 3rd-party appliances (firewalls, monitoring)**
   * **Best for security & traffic inspection tools**

**🔧 Steps to Create an Application Load Balancer (ALB)**

**Step 1: Open EC2 Dashboard**

* Go to AWS Console → EC2 → Load Balancers → **Create Load Balancer**

**Step 2: Select Load Balancer Type**

* Choose **Application Load Balancer**

**Step 3: Configure Load Balancer**

* Name: my-alb
* Scheme: **Internet-facing** (for public access)
* IP type: IPv4
* Listeners: Add **HTTP on port 80**

**Step 4: Select VPC & Subnets**

* Choose your **VPC**
* Select at least **2 public subnets** (in different AZs)

**Step 5: Configure Security Group**

* Create or select a security group
  + Allow **inbound HTTP (80)** or **HTTPS (443)**

**Step 6: Configure Target Group**

* Target type: **Instance** (or IP/Lambda)
* Protocol: HTTP
* Port: 80
* Health checks: Path = /, protocol = HTTP

**🌐 Step 7: Test on Internet**

1. Go to EC2 → Load Balancers
2. Copy the **DNS name** (e.g., my-alb-123456.ap-south-1.elb.amazonaws.com)
3. Paste in browser: