**What is Load Balancing?**

Load balancing is the process of distributing incoming network traffic across multiple servers (or resources) so no single server gets overwhelmed. Think of it like having several checkout counters at a supermarket instead of one.

The main goals:

* Improve **performance** (faster response time)
* Increase **reliability** (no downtime if one server fails)
* Scale easily (add/remove servers as needed)

Load balancing can happen at different layers of the OSI model (mostly Layer 4 - Transport and Layer 7 - Application).

**Types of Load Balancing**

**1. Hardware vs Software Load Balancing**

* **Hardware Load Balancers**
  + Dedicated appliances (e.g., F5 BIG-IP, Citrix ADC).
  + Very fast, but expensive and less flexible.
* **Software Load Balancers**
  + Run on standard servers (e.g., HAProxy, Nginx, Envoy, AWS ELB, Azure Load Balancer).
  + Cheaper, flexible, cloud-friendly.

**2. Layer 4 vs Layer 7 Load Balancing**

* **Layer 4 (Transport Layer)**
  + Makes decisions based on TCP/UDP info (IP address, port).
  + Faster, lightweight, but less “smart.”
  + Example: Distribute traffic evenly across backend servers regardless of content.
* **Layer 7 (Application Layer)**
  + Makes decisions based on the actual content of the request (URL, headers, cookies).
  + Allows advanced routing: send image requests to one server group, API requests to another.
  + Example: AWS Application Load Balancer, Nginx reverse proxy.

**Load Balancing Algorithms (Techniques)**

1. **Round Robin**
   * Requests go to servers in rotation.
   * Simple, but doesn’t consider server load.
2. **Weighted Round Robin**
   * Like round robin, but gives more requests to powerful servers.
3. **Least Connections**
   * Directs traffic to the server with the fewest active connections.
   * Good when sessions vary in length.
4. **Least Response Time**
   * Sends traffic to the server that has the fewest active connections **and** lowest latency.
5. **IP Hash**
   * Uses a hash of the client’s IP address to pick a server.
   * Ensures the same client goes to the same server (good for session persistence).
6. **Random with Two Choices**
   * Picks two servers at random and sends traffic to the one with fewer connections.
   * Balances speed and fairness.

**Load Balancing in Practice**

* **Health Checks:** Load balancers regularly check if a server is alive and healthy before sending it traffic.
* **Failover:** If one server fails, traffic is automatically routed to the others.
* **Sticky Sessions (Session Persistence):** Some apps need the same user to always hit the same server (e.g., shopping carts). Load balancers handle this with cookies or IP hashing.
* **Global Load Balancing (GSLB):** Spreads traffic across servers in different geographic regions using DNS or Anycast routing.