**What is Load Balancing ?**

Load balancing is the process of distributing incoming network traffic or computational workloads across multiple servers, systems, or resources to ensure no single resource is overwhelmed. This improves responsiveness, availability, and reliability of applications and services.

### **How Load Balancing Is Done**

Load balancing is done by placing a **load balancer** (a device or software) between clients and backend servers. It acts as a **traffic manager**, receiving requests from clients and intelligently forwarding them to one of the available backend servers based on a defined algorithm or policy.

### Step-by-Step Overview:

1. Client Sends Request : A user (or system) makes a request to a single IP address (e.g., example.com).
2. Load Balancer Receives the Request : The request first goes to the load balancer, not directly to the backend servers.
3. Choose a Server Using a Load Balancing Algorithm  
   * Examples:  
     + Round Robin
     + Least Connections
     + IP Hash
     + Weighted Algorithms
4. Forward the Request to the Selected Server : The load balancer forwards the request to the chosen backend server.
5. Server Processes Request and Sends Response : The selected backend server processes the request and sends the response back to the load balancer.
6. Load Balancer Sends Response to Client: The load balancer forwards the server's response back to the client, often without the client knowing which server handled the request.

### **Example Use Case**

Imagine a website with 3 backend web servers:

1. A user visits the website.
2. The DNS points to a load balancer (e.g., Nginx).
3. The load balancer checks which of the 3 servers is least busy.
4. It forwards the request to Server 2.
5. Server 2 handles it, sends the result back through the load balancer.
6. The user gets a fast and seamless response.

**Different strategies for Load Balancing**

### 1. Round Robin

Distributes incoming requests to servers sequentially in a circular order. After the last server, it starts again from the first.

Best for: Servers with similar specs and workloads.

### 2. Least Connections

Sends requests to the server with the fewest active connections at the moment.

Best for: Applications with long-lived or uneven requests.

### 3. IP Hash

Uses a hash of the client’s IP address to determine which server will handle the request, ensuring the same client is sent to the same server each time.

Best for: Session persistence (sticky sessions).

### 

### 4. Least Response Time

Routes traffic to the server with the lowest average response time and the fewest active connections.

Best for: Time-sensitive applications needing fast performance.

### 5. Random

Selects a server at random for each request, without any specific pattern.

Best for: Simple setups or low traffic systems where even distribution is enough.

### **Round Robin Load Balancing**

Round Robin is a load balancing strategy that distributes client requests evenly across a group of servers by assigning each incoming request to the next server in a fixed, cyclic order.

**How It Works:** When a request arrives, the load balancer forwards it to the next server in the list. After reaching the last server, it loops back to the first, repeating the cycle continuously. It does not consider the current load or capacity of the servers.

**Example:** Suppose you have three servers: A, B, and C. Incoming requests are distributed like this:

* Request 1 → Server A
* Request 2 → Server B
* Request 3 → Server C
* Request 4 → Server A
* Request 5 → Server B  
   …and so on.

### 

### 

### 

### 

### **Least Connections Load Balancing**

**Definition:** Least Connections is a load balancing strategy that directs incoming requests to the server with the **fewest active connections** at the moment, aiming to balance the load based on real-time server usage.

**How It Works:** The load balancer continuously monitors how many active connections each server is handling. When a new request arrives, it sends the request to the server currently managing the least number of connections, helping to prevent overloading busy servers.

**Example:** Imagine three servers with active connections:

* Server A: 10 connections
* Server B: 3 connections
* Server C: 7 connections

A new incoming request will be forwarded to **Server B** because it has the fewest active connections.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **Random Load Balancing**

**Definition:** Random load balancing distributes incoming requests by selecting a server **at random** from the pool of available servers.

**How It Works:** For each incoming request, the load balancer randomly picks one server to handle it, without considering the current load or server capacity.

**Example:** If there are three servers—A, B, and C—a request might go to:

* Request 1 → Server B
* Request 2 → Server A
* Request 3 → Server C
* Request 4 → Server B  
   …and so on, completely at random.