

## **Proxies:**

Proxies act as intermediaries between clients and servers in network communication.

### **Forward Proxies:**

- Role: Represents clients to access resources from the internet.
- Usage: Enhances privacy, filters content, and hides client identities.
- Example: Used in corporate networks to control outbound traffic.

### **Reverse Proxies:**

- Role: Represents servers, handles requests on behalf of them.
- Usage: Load balancing, security, caching, SSL termination.
- Example: Used by websites to manage incoming traffic and improve performance.

### **Differences between forward and reverse proxies:**

- Function:
  - Forward: Protects clients' identities, filters content.
  - Reverse: Offloads servers, enhances security, manages traffic.
- Access:
  - Forward: Clients access internet via proxy.
  - Reverse: Clients access servers via proxy.
- Visibility:
  - Forward: Servers don't know client identities.
  - Reverse: Clients don't know server details.
- Location:
  - Forward: Typically located in client networks.
  - Reverse: Typically located in front of servers.
- Use Cases:
  - Forward: Anonymity, content filtering.
  - Reverse: Load balancing, security, caching.
- Examples:
  - Forward: Used in home networks, corporate setups.
  - Reverse: Common for websites, web applications.

## **Load Balancers:**

- Load balancers distribute incoming network traffic across multiple servers to ensure efficient resource utilization and prevent overload.
- Purpose: Enhance system performance, optimize resource usage, and improve reliability.

## **Load Balancing Strategies:**

### **Round Robin:**

- Distributes traffic equally to each server in a cyclic manner.
- Simple to implement but doesn't consider server load or capacity.

### **Least Connections:**

- Routes traffic to the server with the fewest active connections.
- Suitable for balancing unevenly sized requests.

### **Least Response Time:**

- Sends traffic to the server with the lowest response time.
- Effective for minimizing user-perceived latency.

### **IP Hash:**

- Assigns clients to servers based on their IP addresses.
- Ensures consistent routing for the same client.

### **Weighted Round Robin:**

- Assigns a weight to each server to control the amount of traffic it receives.
- Allows allocating more resources to powerful servers.

### **Weighted Least Connections:**

- Accounts for server capacity by considering both weight and current connections.
- Ensures efficient resource distribution.

### **Random:**

- Randomly selects a server to handle each request.
- Simple but might not ensure even load distribution.