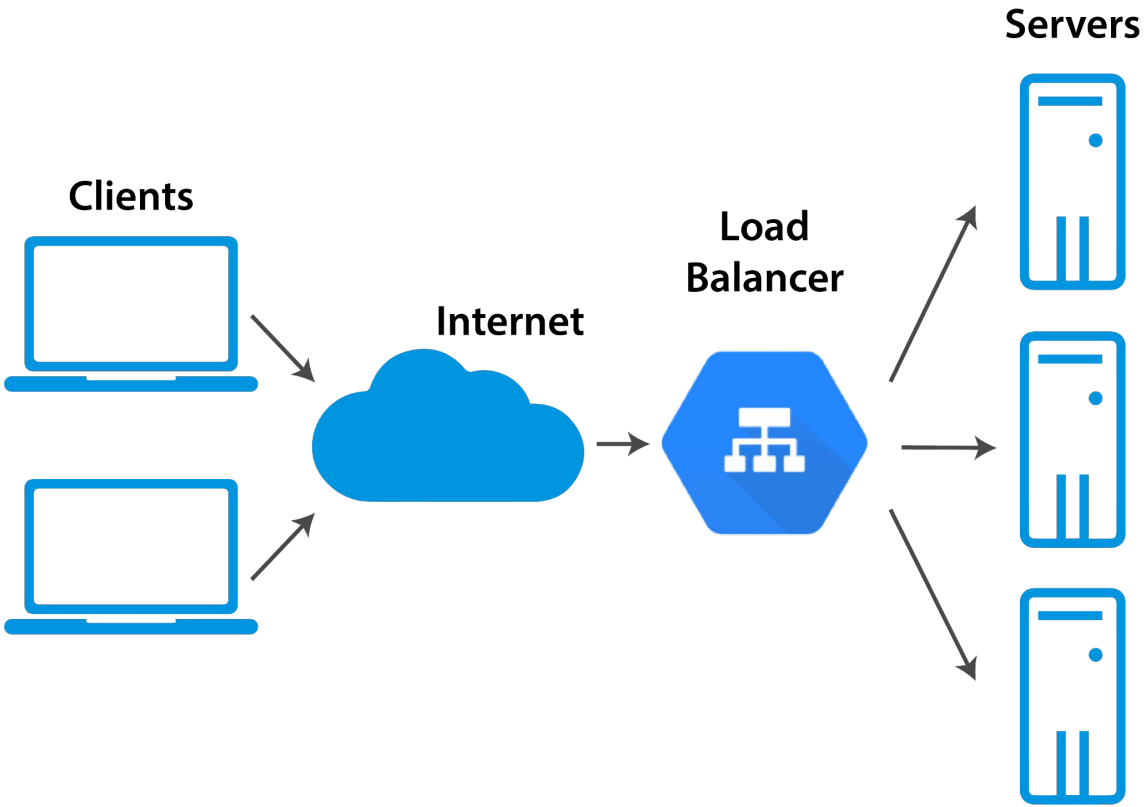
Load Balancing

...

What is Load Balancing?

- Load balancing enables our service to scale and stay highly available when the traffic load increases.
- Load balancers act as a single point of contact for all the client requests.
- Load balancers:
  - Distribute heavy traffic load across the servers running in the cluster
  - Avert the risk of all the traffic converging to a single or a few machines in the cluster.
  - If a server goes down, the load balancer automatically routes the future requests to other running server nodes in the cluster.

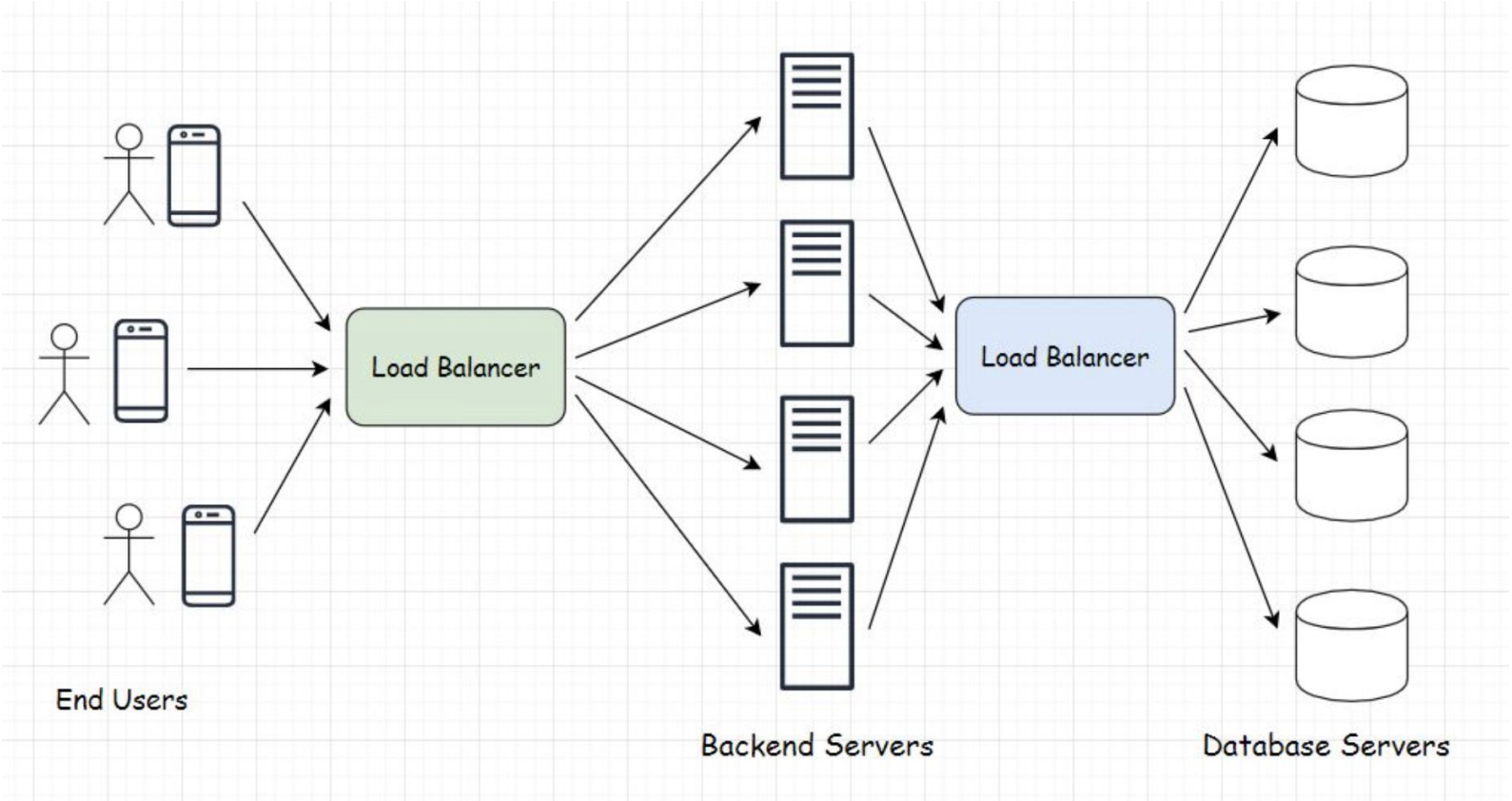
What is Load Balancing?



Load Balancers: Multiple parts of stack

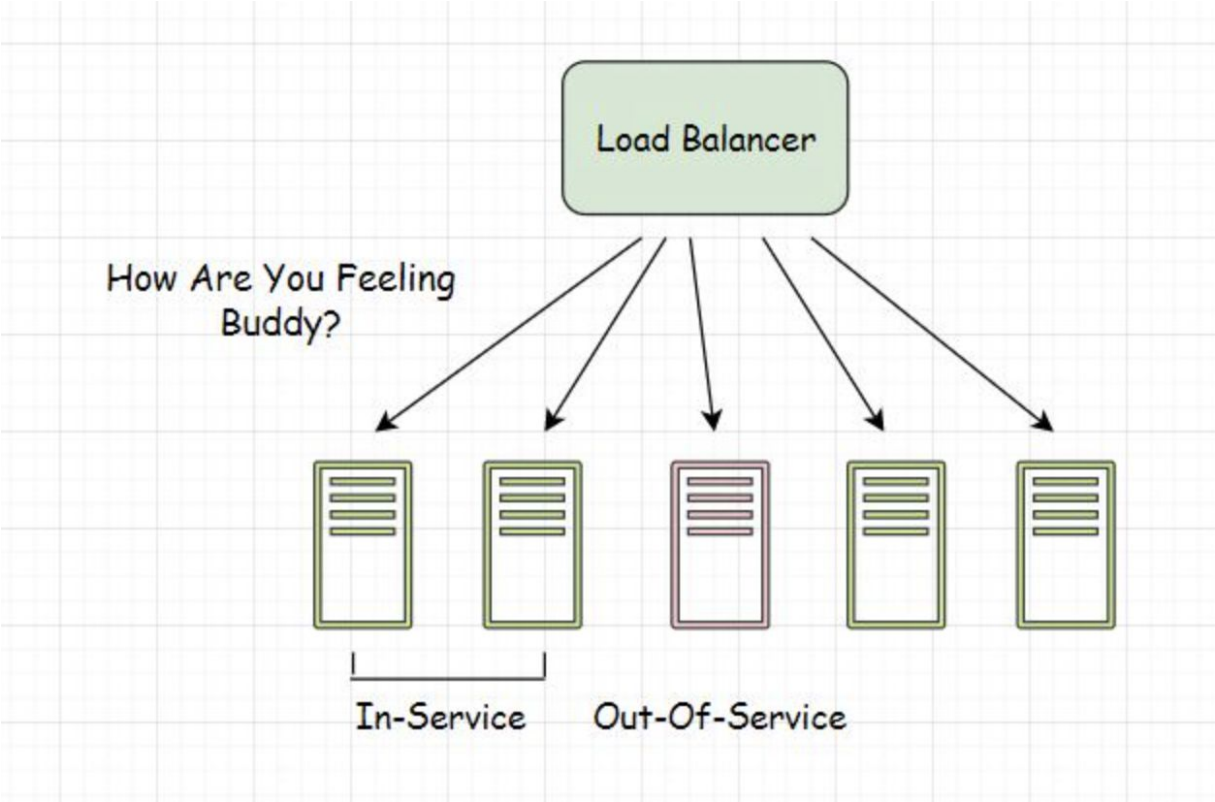
- LBs can be set up application component level to efficiently manage traffic directed towards any application component
  - backend application server
  - database component
  - message queue
- Uniformly spread the request load across the machines in the cluster powering that particular application component.

Load Balancers: Multiple parts of stack



Load Balancers and Health Checks

- LBs want to always route requests to a **running** machine
  - regularly perform health checks on the machines in the cluster.



Load Balancing Approaches

- Types:
  - DNS Load Balancing
  - Hardware-based Load Balancing
  - Software-based Load Balancing

DNS Load Balancing

- Configuring a domain in the Domain Name System (DNS) such that client requests to the domain are distributed across a group of server machines
- Nameservers do round-robin (just cycling through) returning various servers
- Downside: no health checks; could send to broken server
- Upside: the Nameserver manages it; you don’t have it

Hardware Load Balancing

- A hardware device with a specialized operating system that distributes web application traffic across a cluster of application servers.
- Distributes traffic according to customized rules
    - deployed in on-premises data centers
- Upsides: highly performant
- Downsides: have to work with datacenters to configure.
- Large companies use these!

Software Load Balancing

- Software load balancers can be installed on commodity hardware and VMs (i.e. in AWS)
  - They are more cost-effective and offer more flexibility to the developers.
- Software load balancers can be upgraded and provisioned easily compared to hardware.
  - Load Balancers as a Service (LBaaS):
    - enable you to directly plug in load balancers into your application without much setup.
- Software load balancers consider many parameters:
  - data hosted by the servers
  - CPU and memory utilization
  - load on the network, etc.
- They perform health checks on the servers to keep an updated list of running machines.

Software Load Balancing: Routing Algos

- **Round robin**
  - Sends IP addresses of machines sequentially to the clients.
  - Parameters such as the server load, CPU consumption, and so on are not considered when sending the IP addresses to the clients.
- **Weighted round robin**
  - based on the server's compute and traffic handling capacity, weights are assigned to them.
  - And then, based on server weights, traffic is routed to them using the round robin algorithm.

Software Load Balancing: Routing Algos

- **Least connections**
  - Traffic is routed to the machine with the least open connections of all the machines in the cluster.
  - Used when the server has long opened connections like persistent connections in a gaming application.
- Approach 1: Assume that all the requests will consume an equal amount of server resources, and the traffic is routed to the machine with the least open connections based on this assumption.
    - In this scenario, there is a possibility that the machine with the least open connections might already be processing requests demanding most of its CPU power. Routing more traffic to this machine would not be a good idea.
- Approach 2: the CPU utilization and the request processing time of the chosen machine are also considered before routing the traffic to it.
  - Machines with the shortest request processing time, least CPU utilization, and the least open connections are suitable candidates to process future client requests.

Software Load Balancing: Routing Algos

- **Random**
  - Traffic is randomly routed to the servers.
- No way to targetedly attack this
- Simple to implement

Software Load Balancing: Routing Algos

- **Hash** the source IP where the request is coming from and the request URL are hashed to route the traffic to the backend servers.
    - Hashing the source IP ensures that a client's request with a certain IP will always be routed to the same server.
- Server has already processed the initial client requests and holds the client's data in its local memory
    - There is no need for it to fetch the client session data from the session memory of the cluster and process the request. This reduces latency.
- Hashing the client IP also enables the client to re-establish the connection with the same server that was processing its request in case the connection drops.
- Hashing a URL ensures that the request with that URL always hits a certain cache that already has data on it.
    - This is to ensure that there is no cache miss.
- This also averts the need for duplicating data in every cache