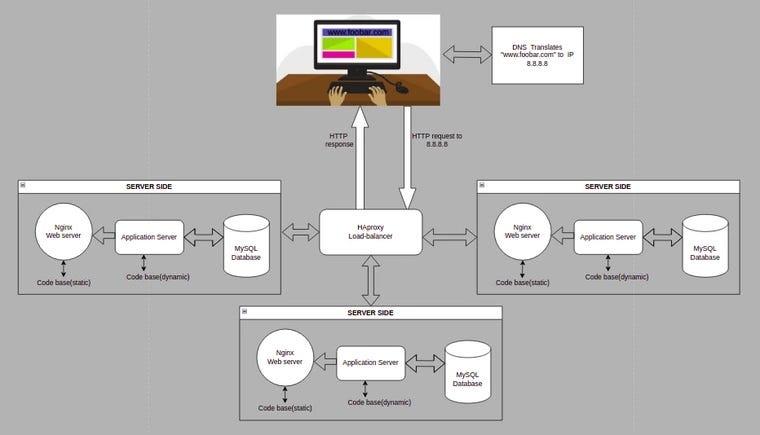
### **Distributed web infrastructure**



For every additional element, why you are adding it

**Server:** Serves a portion of the users when traffic is distributed. When traffic starts to grow, and I mean really GROW, When millions of users at once are making requests for websites like Google and Facebook, One web server won’t be able to handle all these corresponding requests, so we have to use two (or many more) servers. A new problem arises — when a user makes a request, will the content come from webserver 1 or web server 2? For this exact reason, these types of websites have a Load Balancer (which is actually also a server).

**Load balancer:** Distributes traffic using round robin technique thus eliminating single

point of failure. The purpose of the Load Balancer is to distribute incoming traffic across multiple servers, which increases the efficiency, reliability, and availability of your site. If one web server crashes all of a sudden, this special server(Load balancer) automatically redirects the traffic to the remaining web servers.

What distribution algorithm your load balancer is configured with and how it works

The Load Balancer has different algorithms for how it divides up the workload, such as:

**Round Robin (most common)** — Requests are distributed across the group of servers sequentially. Request 1 is directed to server 1, request 2 to server 2, and so forth.

**Least Connections** — Before redirecting a request to a server, the Load Balancer computes which server has the least connections, and then sends the request to there.

**IP Hash** — The IP address of the client is used to determine which server the request will be directed to. For example, all IP addresses from 100.100.100.100–400.400.400.400 will be sent to server 3. (IP Hash load balancing uses an algorithm that takes the source and destination IP address of the client and server to generate a unique hash key. This key is used to allocate the client to a particular server. They are assigned individually as they connect to the server and once assigned a certain server, the Client will always connect to that particular server)

Is your load-balancer enabling an Active-Active or Active-Passive setup, explain the difference between both

It enables an Active-Active setup; both nodes (servers) are actively running the same kind of service simultaneously. In an Active-Passive setup, not all nodes are going to be active. In the case of two nodes, if the first node is already active, the second node must be passive or on standby. The key difference between these two architectures is performance. Active-active clusters give you access to the resources of all your servers during normal operation. In an active-passive cluster, the backup server only sees action during failover.

In an active-passive configuration, the server load balancer recognizes a failed node and redirects traffic to the next available node. In an active-active configuration, the load balancer spreads out the workload’s traffic among multiple nodes.

How a database Primary-Replica (Master-Slave) cluster works

It enables data from one database server (the master) to be replicated to one or more other

database servers (the slaves). The master logs the updates, which then ripple through the

slaves. If the changes are made to the master and slave at the same time, it is synchronous. If

changes are queued up and written later, it is asynchronous. It is usually used to spread read

access on multiple servers for scalability, although it can also be used for other purposes such

as for failover, or analyzing data on the slave in order not to overload the master

What is the difference between the Primary node and the Replica node in regard to the application

The difference between the Primary node and the Replica node in regard to the application is that-, the primary node is regarded as the authoritative source, and the replica node (also known as slave) databases are synchronized to it(Master).

Issues with the infrastructure:

1. Where are SPOF

Having one load balancer seem to be the single point of failure in this infrastructure

2. Security issues (no firewall, no HTTPS)

The 2 security issues are communicating over insecure HTTP protocol (transfers data in plain

texts) rather than HTTPS which is secure (encrypts data). This makes it possible for attackers to

hijack the process and gain access to sensitive information.

3. No monitoring

Since we can’t fix or improve what we can’t measure, monitoring is essential. It will help detect

security threats and reduce downtime.