1. Distributed web infrastructure

In this three-server web infrastructure, we host the website [www.foobar.com](<http://www.foobar.com/>). This modification enhances our infrastructure's reliability and resilience. Let's explore the changes:

- Adding Another Server: We’ve introduced an additional server for each component (Web server, Application server, and Database server). This setup minimizes the risk of a Single Point of Failure (SPOF). Both servers run simultaneously, ensuring uninterrupted service even if one server encounters issues.
- Load Balancer: The Load Balancer acts as a traffic cop, distributing incoming requests among multiple servers. Its role is to optimize resource utilization, improve responsiveness, and enhance overall performance and availability.

LOAD BALANCER

The Load Balancer serves as the intermediary between clients and servers. It receives HTTP requests and redirects them to the appropriate server. Load balancers analyze incoming requests and determine distribution methods based on factors such as server load, health, or algorithms.

Load Balancer Algorithms:

- *Round Robin:* Distributes traffic evenly in a circular fashion, ensuring relatively equal distribution.
- *Least Connections:* Redirects HTTP requests to the server with the fewest connections.

These are just a few examples of load balancer algorithms. In our case, assuming both servers have the same capacity, we can use *Round Robin*. However, if load-based distribution is preferred, *Least Connections* can be a good choice.

Load Balancer Modes:

Load balancers can operate in either of these two modes:

- *Active-Active:* All resources linked to the load balancer are active and running. It distributes traffic among resources using the selected distribution algorithm. This mode offers high scalability, utilization, and fault tolerance but can be complex to manage.
- *Active-Passive:* In this mode, one resource runs while others are on standby. They come into play when the primary resource fails. While offering high availability, it has a failover time, causing brief service interruptions.

PRIMARY-REPLICA DATABASES

Also known as a master-slave database cluster, this setup aims for high availability and fault tolerance in database systems.

- The primary database handles all operations (CRUD), while the replica is read-only.
- The primary server maintains a log (Binary log) of changes and

communicates with replica servers to keep them updated.

- Replica servers receive updates from the primary server and apply changes for synchronization, potentially located in different locations to enhance user experience.

ISSUES

Here are potential Single Points of Failure (SPOFs) and security risks that this infrastructure may face:

- FIREWALL: Security devices that monitor traffic based on predetermined security rules. In this infrastructure, two firewalls are essential: one between the client and the ISP (client-side firewall) and one between the ISP and the server (network firewall).
- HTTPS: HyperText Transfer Protocol Secure, a protocol that applies encryption to secure data exchange between the user’s browser and the web server, ensuring confidentiality.
- MONITORING: Real-time tracking of the system’s health, performance, and security is vital to address potential issues promptly.