

How it works :

Additional Elements and Their Purposes

1. Load Balancer (HAProxy)

- **Purpose:** Distributes incoming traffic across multiple servers to ensure no single server bears too much load
- **Distribution Algorithm:** Configured with Round Robin algorithm, which:
 - Distributes requests sequentially across the server pool
 - Works well when all servers have similar capabilities
 - Provides simple and fair distribution of traffic
- **Active-Active Setup:**
 - Both servers actively handle traffic simultaneously
 - Provides better load distribution and redundancy
 - Maximizes resource utilization of all servers

2. Multiple Servers

- **Purpose:** Provides redundancy and increased capacity
- Each server contains:
 - Nginx web server
 - Application server
 - Application codebase
 - MySQL database (configured in Primary-Replica setup)

3. Database Primary-Replica Cluster

- **Primary Node (Master):**
 - Handles all write operations
 - Logs changes in a binary log
 - Sends updates to replica nodes
- **Replica Node (Slave):**
 - Replicates data from the primary node

- Primarily handles read operations
- Helps distribute database load
- Can be promoted to primary if the primary fails

Infrastructure Issues

1. Single Points of Failure (SPOF)

- **Load Balancer:** Single load balancer is a SPOF - if it fails, the entire system becomes inaccessible
- **Primary Database:** If the primary database fails, write operations will be interrupted until the replica is promoted
- **No Redundant Network Paths:** Single network path to each server could cause isolation if it fails

2. Security Issues

- **No Firewall:** Leaves servers vulnerable to unauthorized access and various cyber threats
- **No HTTPS:** Data transmission is not encrypted, exposing:
 - User data to potential interception
 - Website to man-in-the-middle attacks
 - Lack of authentication between client and server

3. Monitoring Deficiencies

- **No Monitoring System:** Unable to:
 - Detect and respond to issues proactively
 - Track system performance and health
 - Identify potential bottlenecks
 - Alert administrators of anomalies

Differences Between Primary and Replica Nodes

Primary Node (Master)

- Handles all write operations
- Maintains the authoritative copy of the database
- Changes are logged and replicated to slaves
- Critical for data consistency

Replica Node (Slave)

- Primarily handles read operations
- Reduces load on the primary database
- Can experience replication lag
- Provides redundancy for read operations

Active-Active vs Active-Passive Setup Explanation

Active-Active

- Currently implemented in this design
- Both servers actively handle traffic
- Provides better resource utilization
- Offers improved load distribution

Active-Passive

- Alternative setup where one server remains on standby
- Passive server only takes over if active server fails
- Less efficient resource utilization but simpler to manage
- Typically used when consistency is more important than performance