# 0x09. Web infrastructure design

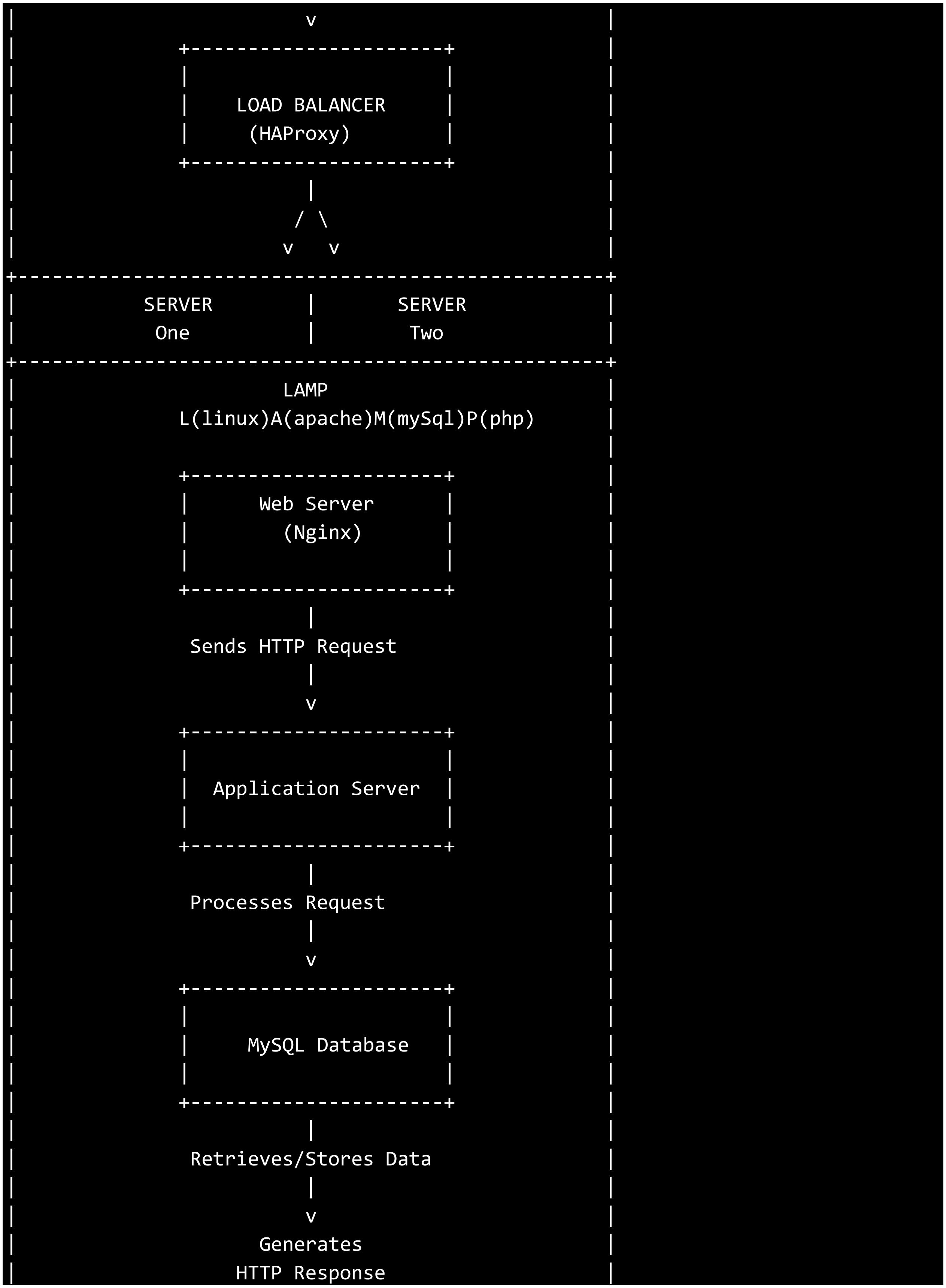
## 1. Distributed web infrastructure

## Table of Content

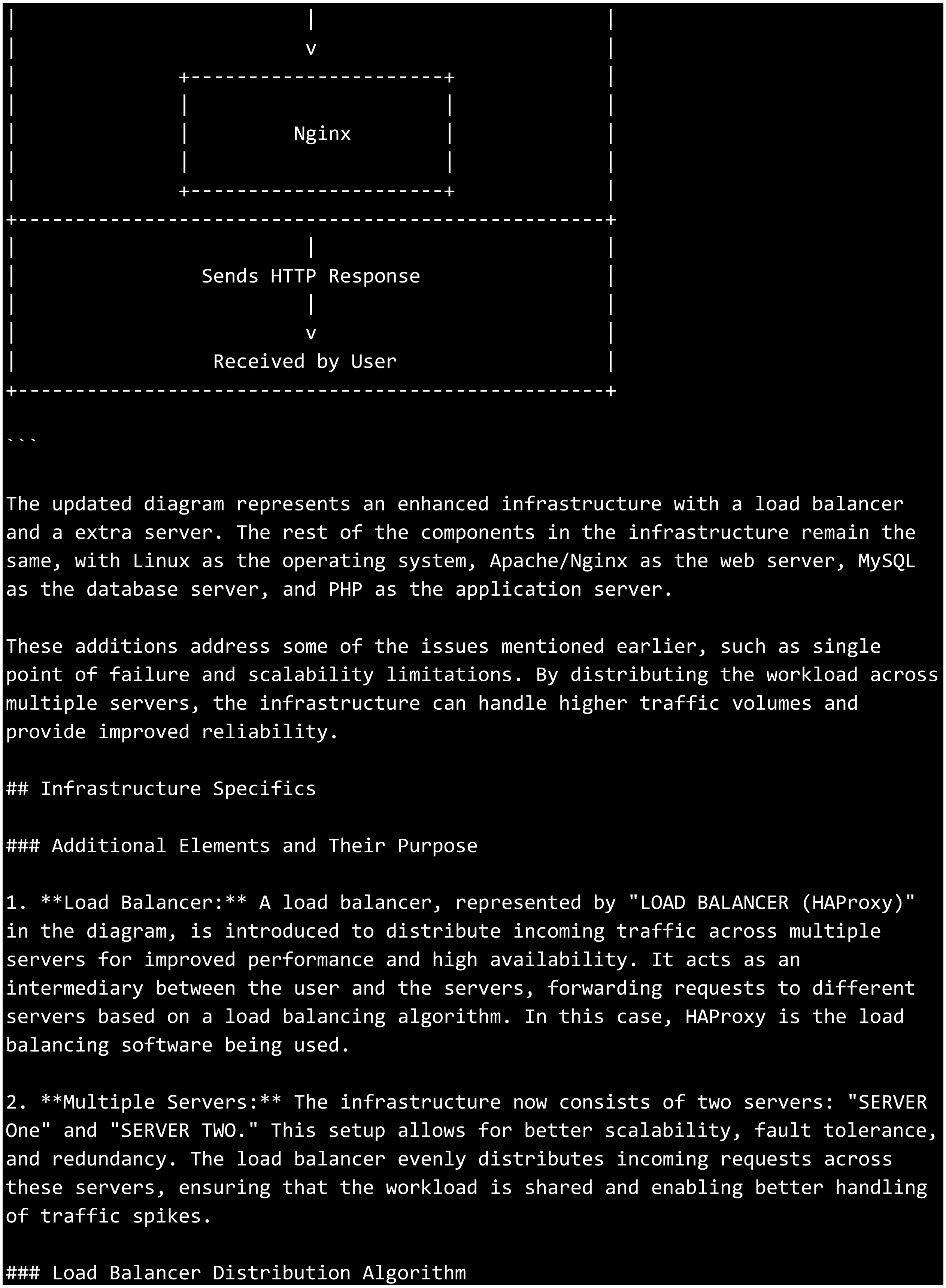
## Diagram Representation

```sql

+---------------------------------------------------+
|                      User                         |
+---------------------------------------------------+
|               Enters www.foobar.com               |
|                         |                         |
|                         v                         |
|              +----------------------+             |
|              |                      |             |
|              |          DNS         |             |
|              |                      |             |
|              +----------------------+             |
|                         |                         |
|               Resolves www.foobar.com             |
|                         |                         |
|                         v                         |
|               Resolved IP: 8.8.8.8                |
|                         |                         |
```



```
|                         v                         |
|              +----------------------+             |
|              |                      |             |
|              |    LOAD BALANCER     |             |
|              |     (HAProxy)        |             |
|              +----------------------+             |
|                         |                         |
|                        / \                        |
|                       v   v                       |
+---------------------------------------------------+
|           SERVER        |       SERVER            |
|            One          |        Two              |
+---------------------------------------------------+
|                       LAMP                        |
|              L(linux)A(apache)M(mySql)P(php)      |
|                                                   |
|              +----------------------+             |
|              |      Web Server      |             |
|              |        (Nginx)       |             |
|              |                      |             |
|              +----------------------+             |
|                         |                         |
|               Sends HTTP Request                  |
|                         |                         |
|                         v                         |
|              +----------------------+             |
|              |                      |             |
|              |  Application Server  |             |
|              |                      |             |
|              +----------------------+             |
|                         |                         |
|               Processes Request                   |
|                         |                         |
|                         v                         |
|              +----------------------+             |
|              |                      |             |
|              |     MySQL Database   |             |
|              |                      |             |
|              +----------------------+             |
|                         |                         |
|               Retrieves/Stores Data               |
|                         |                         |
|                         v                         |
|                     Generates                     |
|                   HTTP Response                   |
```



~~~
|                         |                         |
|                         v                         |
|              +----------------------+             |
|              |                      |             |
|              |         Nginx        |             |
|              |                      |             |
|              +----------------------+             |
+---------------------------------------------------+
|                         |                         |
|                Sends HTTP Response                |
|                         |                         |
|                         v                         |
|                 Received by User                  |
+---------------------------------------------------+

```
~~~

The updated diagram represents an enhanced infrastructure with a load balancer and a extra server. The rest of the components in the infrastructure remain the same, with Linux as the operating system, Apache/Nginx as the web server, MySQL as the database server, and PHP as the application server.

These additions address some of the issues mentioned earlier, such as single point of failure and scalability limitations. By distributing the workload across multiple servers, the infrastructure can handle higher traffic volumes and provide improved reliability.

## Infrastructure Specifics

### Additional Elements and Their Purpose

1. **Load Balancer:** A load balancer, represented by "LOAD BALANCER (HAProxy)" in the diagram, is introduced to distribute incoming traffic across multiple servers for improved performance and high availability. It acts as an intermediary between the user and the servers, forwarding requests to different servers based on a load balancing algorithm. In this case, HAProxy is the load balancing software being used.

2. **Multiple Servers:** The infrastructure now consists of two servers: "SERVER One" and "SERVER TWO." This setup allows for better scalability, fault tolerance, and redundancy. The load balancer evenly distributes incoming requests across these servers, ensuring that the workload is shared and enabling better handling of traffic spikes.

### Load Balancer Distribution Algorithm

```
The specific load balancer distribution algorithm can vary based on
configuration, but common algorithms include Round Robin, Least Connections, and
IP Hash. The algorithm determines how the load balancer selects a server to
handle each incoming request. For example, Round Robin distributes requests in a
cyclic manner, Least Connections selects the server with the fewest active
connections, and IP Hash uses the client's IP address to determine the server.
The chosen algorithm depends on factors like traffic patterns, server capacities,
and session persistence requirements.

### Active-Active vs. Active-Passive Setup

The load balancer in this infrastructure enables an Active-Passive setup. In an
Active-Active setup, all servers actively handle requests simultaneously, sharing
the workload. However, in an Active-Passive setup, one server (active) handles
incoming traffic while the other server(s) (passive) are on standby, ready to
take over if the active server fails. This configuration provides redundancy and
failover capabilities but may underutilize resources during normal operation.

### Database Primary-Replica (Master-Slave) Cluster

A database Primary-Replica (or Master-Slave) cluster involves having a primary
database node and one or more replica nodes. The primary node handles read and
write operations and acts as the authoritative source of data. The replica nodes
replicate data from the primary node and serve as backups or read-only replicas.
Replication ensures data redundancy, scalability, and fault tolerance.

### Role Difference Between Primary Node and Replica Node

The primary node in a database cluster handles read and write operations, manages
data modifications, and acts as the main database server. It receives updates and
replicates the changes to the replica nodes. The replica nodes, on the other
hand, do not accept write operations but instead replicate data from the primary
node. They can serve read requests, improving read performance and providing
failover capabilities in case the primary node becomes unavailable.

## Issues with the Infrastructure

### Single Point of Failure (SPOF)

The infrastructure still faces potential single points of failure. If the load
balancer or any of the servers (primary or replica) fail, it can disrupt the
availability of the website. Introducing redundancy, such as additional load
balancers or server clusters, can help mitigate this issue.
```

```
### Security Issues

The infrastructure lacks certain security measures. Not having a firewall leaves
the system vulnerable to unauthorized access or malicious attacks. Additionally,
the absence of HTTPS (HTTP Secure) encryption exposes user data to potential
eavesdropping and manipulation. Implementing a firewall and enabling HTTPS
through SSL/TLS certificates are crucial for protecting sensitive information and
securing communications.

### Lack of Monitoring

The absence of proper monitoring tools and practices can hinder the ability to
detect and address performance issues, bottlenecks, or potential failures in
real-time. Monitoring is essential for maintaining system health, identifying
anomalies, and ensuring proactive management and troubleshooting. Implementing
monitoring solutions can provide valuable insights into the infrastructure's
performance, resource utilization, and overall health.
```