2. Secured and Monitored Web Infrastructure

Domain: www.foobar.com

- **DNS A Record** points to **Load Balancer IP**
- **Serves HTTPS traffic using SSL certificate**

Infrastructure Overview (3 Servers + Security & Monitoring)

Server	Role	Components Installed
Server 1	Web & App Server	Nginx, App Code, Monitoring Agent, Firewall
Server 2	Web & App Server	Nginx, App Code, Monitoring Agent, Firewall
Server 3	Load Balancer	HAProxy, SSL Termination, Monitoring Agent, Firewall
Database	MySQL (optional separate node or installed on one of the above servers)	

Security Enhancements

- **1. SSL Certificate**
  - Installed on **HAProxy (Load Balancer)**
  - **Purpose:** Encrypts all traffic from client to server via **HTTPS**
  - **Why?** Protects against:
    - Man-in-the-middle (MITM) attacks
    - Data interception

- **2. Firewalls (x3)**
  - One per server: Server 1, Server 2, Load Balancer
  - **Purpose:** Controls inbound/outbound traffic
  - **Typical rules:**

    - Allow: HTTP (80), HTTPS (443), SSH (22), MySQL (3306 internal only)
    - Deny all others by default
- **Why?** Reduces attack surface and prevents unauthorized access

Monitoring Setup

🔸 **3 Monitoring Clients**

- Installed on **each server**
- Uses a service like **Sumo Logic, Datadog, Prometheus, or ELK**
- **Purpose:** Collects logs, metrics (CPU, memory, disk, QPS, etc.), alerts

What is Monitoring Used For?

- Detecting performance bottlenecks
- Tracking uptime
- Getting alerts on errors or abnormal behavior
- Observing user activity & traffic trends

How Monitoring Works

- Each agent/collector:
    - Collects logs (e.g., Nginx logs, system logs)
    - Collects metrics (e.g., CPU usage, response times, QPS)
    - Sends data to a centralized monitoring server/cloud

How to Monitor QPS (Queries per Second)

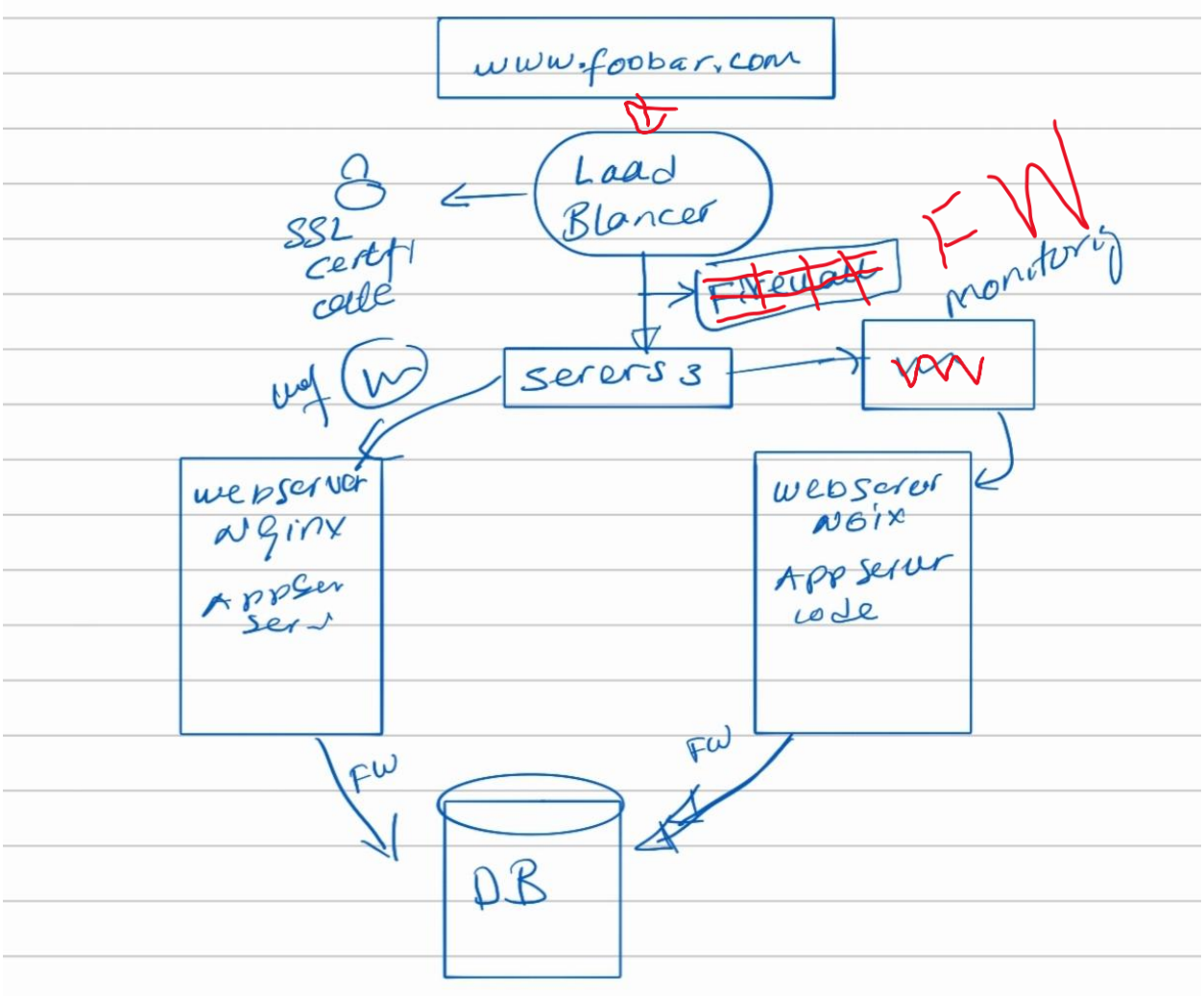
- Configure monitoring agent to track:
    - Nginx logs → parse requests per second
    - App logs or metrics endpoints
    - MySQL performance schema (for DB QPS)
- Tools: Prometheus + Grafana, Datadog, or Sumo Logic dashboards

Infrastructure Issues and Explanations

Issue	Explanation
SSL Termination at Load Balancer	If SSL is terminated at the load balancer and **not re-encrypted** internally, the traffic from load balancer to web servers is unencrypted (vulnerable on internal network). Best practice: use **SSL passthrough** or re-encrypt.
Single MySQL Write Node	One MySQL server handling all writes is a **SPOF (Single Point of Failure)**. If it crashes, no data can be written. It also becomes a **performance bottleneck**.
All-in-one Servers (App + Web + DB)	Violates separation of concerns. Makes scaling, maintaining, and securing each component harder. For example, updating MySQL might disrupt the web server on the same machine. Also, if one service is compromised, others are at risk.

Summary of Additional Elements and Why They're Added

Element	Purpose
Firewalls (x3)	Enforce access control to each server
SSL Certificate	Encrypts traffic to ensure secure communication
Monitoring Clients (x3)	Collect and send system, app, and network metrics/logs for performance and security visibility


www.foobar.com
Laad Blancer
SSL certif calle
Firewall
FW monitoring
serers 3
webserver nginx
Appser ser
webserer nGix
App server code
FW
FW
DB