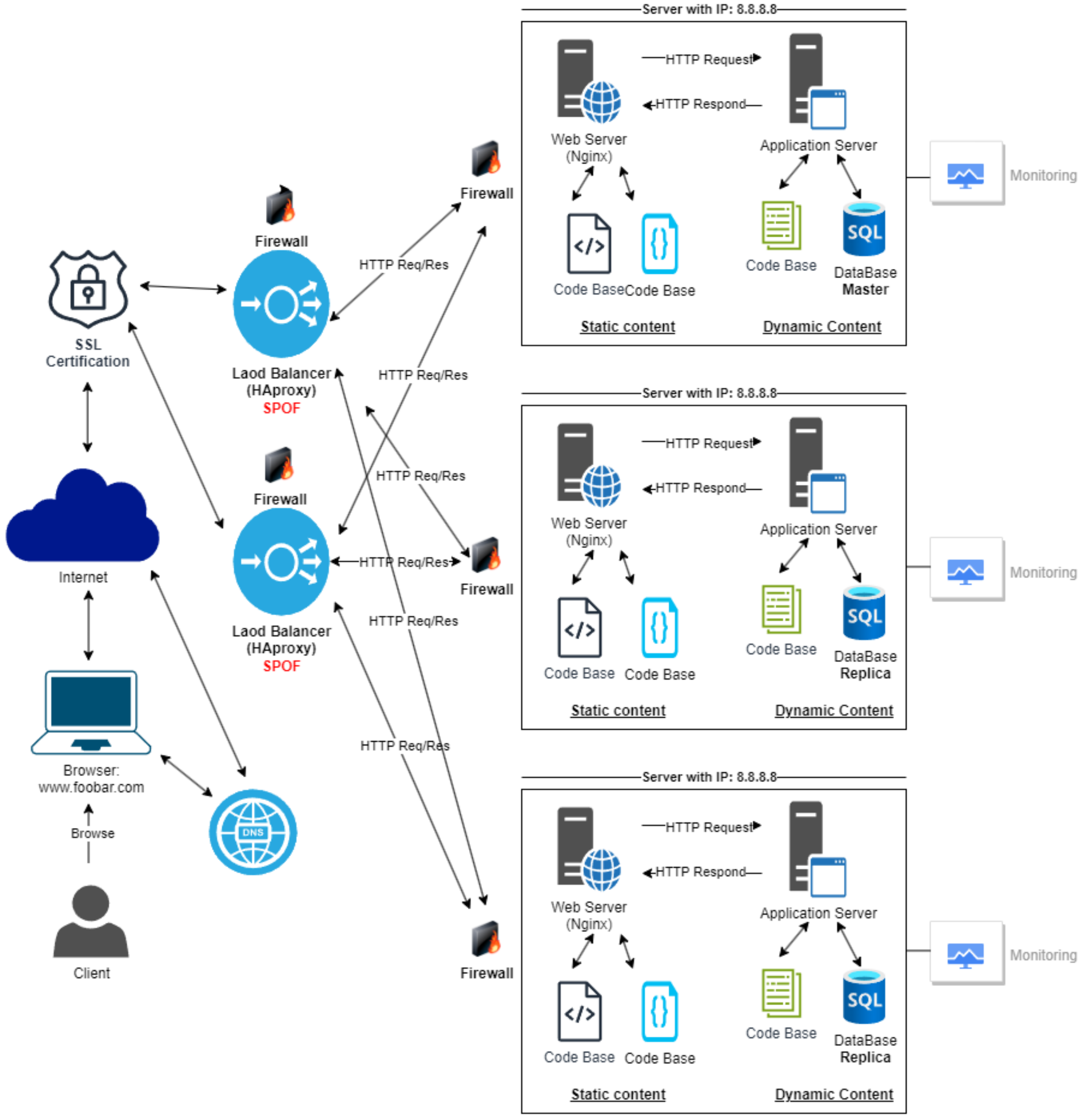
Application Server vs. Web Server

Whiteboard Diagram



Explanation

Scenario:

Designing an infrastructure that includes both an Application Server and a Web Server, with load balancing for improved performance.

Components:

1. Load Balancer (HAproxy):

- **Purpose:**
  - Distributes incoming traffic between the Web Server and Application Server.
  - Enhances scalability and provides fault tolerance.
- **Configuration:**
  - Configured as a cluster with another HAproxy for high availability.

2. Web Server (Nginx):

- **Purpose:**
  - Handles incoming HTTP requests from users.
  - Serves static content directly to users.
- **Addition:**
  - Introducing Nginx as a dedicated web server to efficiently handle static content, ensuring faster response times.

3. Application Server:

- **Purpose:**
  - Executes the application logic, handling dynamic content.
  - Processes requests that require server-side logic.
- **Addition:**
  - Introducing a separate application server to efficiently process dynamic content, enhancing overall performance.

4. Database:

- **Purpose:**
  - Stores and manages website data.
- **Addition:**
  - Represented separately, emphasising a distinct server for data storage and retrieval.

5. Cluster Configuration:

- **Purpose:**
  - Ensures high availability and reliability.
  - Distributes incoming traffic evenly between the two load balancers.
- **Addition:**
  - Configuring HAproxy as a cluster to avoid a single point of failure and improve system resilience.

Explanation:

- **Load Balancer:**
  - **Why**: Added for load distribution, fault tolerance, and high availability.
- **Web Server:**
  - **Why**: Introduced for handling static content efficiently and improving user experience.
- **Application Server:**
  - **Why**: Added for processing dynamic content, separating concerns, and enhancing overall system performance.
- **Database:**
  - **Why**: Represents a dedicated server for data storage and retrieval.