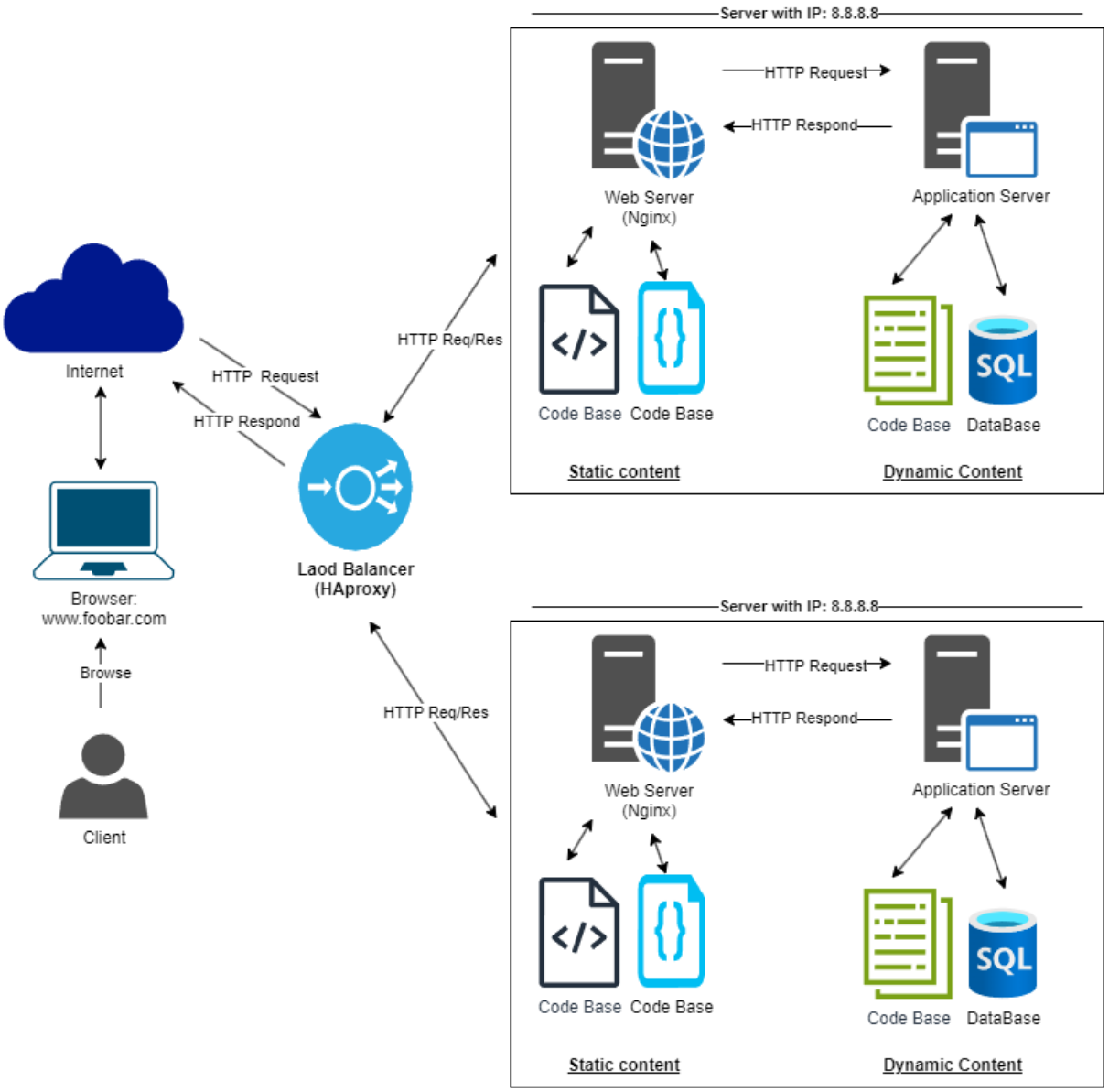
Three-Server Web Infrastructure Design



Components:

1. Load Balancer (HAproxy):

- **Purpose:** Introduces load balancing to distribute incoming traffic across multiple servers, ensuring optimal resource utilisation and preventing a single point of failure.
- **Distribution Algorithm:** Configured with a Round Robin distribution algorithm, evenly distributing requests among the available servers.
- **Setup:** Enables an Active-Active setup, allowing all servers to handle requests simultaneously.

2. Web Server (Nginx):

- **Purpose:** Handles incoming HTTP requests from users.
- **Additional Element:** Introducing Nginx as a dedicated web server for efficient handling of static content and forwarding dynamic content requests to the application server.
- **Distribution Algorithm:** No specific distribution algorithm, as Nginx receives requests from the load balancer.

3. Application Server:

- **Purpose:** Executes the application logic, handling dynamic content generation.
- **Additional Element:** Introducing a separate application server to efficiently process and generate dynamic content, enhancing the overall performance of the web infrastructure.

4. Application Files (Code Base):

- **Purpose:** Contains the source code and files constituting the website's application.
- **Additional Element:** Including a set of application files to be executed by the application server.

5. Database (MySQL):

- **Purpose:** Stores and manages website data.
- **Additional Element:** Introducing a dedicated MySQL database to handle data storage and retrieval for the application.
- **Cluster Setup:** Configured as a Primary-Replica (**Master-Slave**) cluster for high availability.
- **Primary-Replica Setup Explanation:**
  - The Primary node (**Master**) is the main database server that handles both read and write operations.
  - Replica nodes (**Slaves**) are read-only copies of the data, serving read requests and providing redundancy.
  - Ensures data consistency, fault tolerance, and load distribution.

Issues with the Infrastructure:

1. Single Point of Failure (SPOF):

- The load balancer is a potential single point of failure. If it fails, the entire system could become inaccessible.
- Lack of redundancy for the load balancer.

2. Security Issues:

- No firewall mentioned in the design, exposing servers and databases to potential security threats.
- Absence of HTTPS leaves the communication between users and servers unencrypted, posing security risks.

3. No Monitoring:

- Lack of monitoring tools or services to track the health and performance of the infrastructure.
- Absence of monitoring may result in undetected issues, affecting overall system reliability.