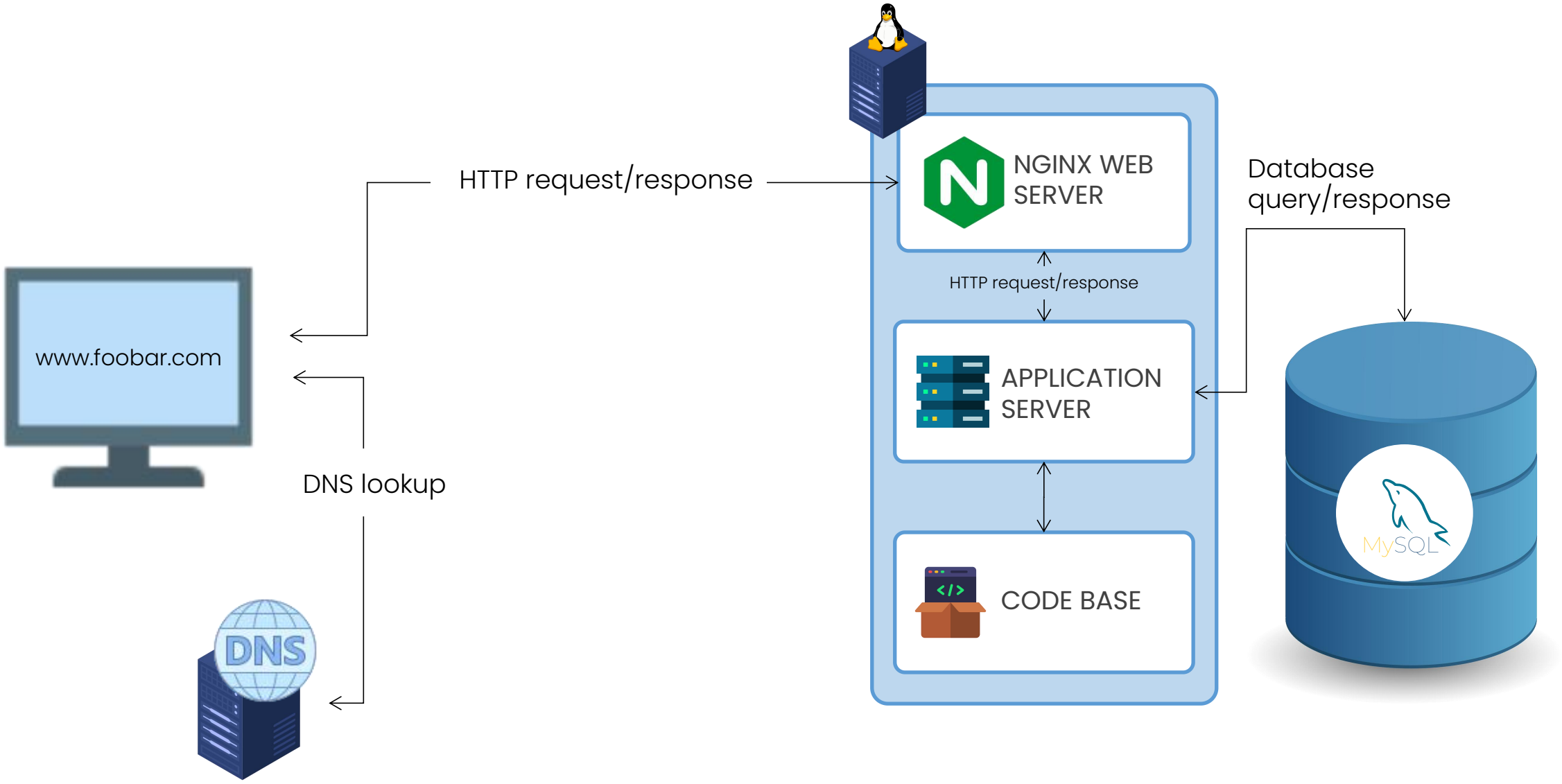

HTTP request/response
www.foobar.com
DNS lookup
DNS
NGINX WEB SERVER
HTTP request/response
APPLICATION SERVER
CODE BASE
Database query/response
MySQL

Explanation:

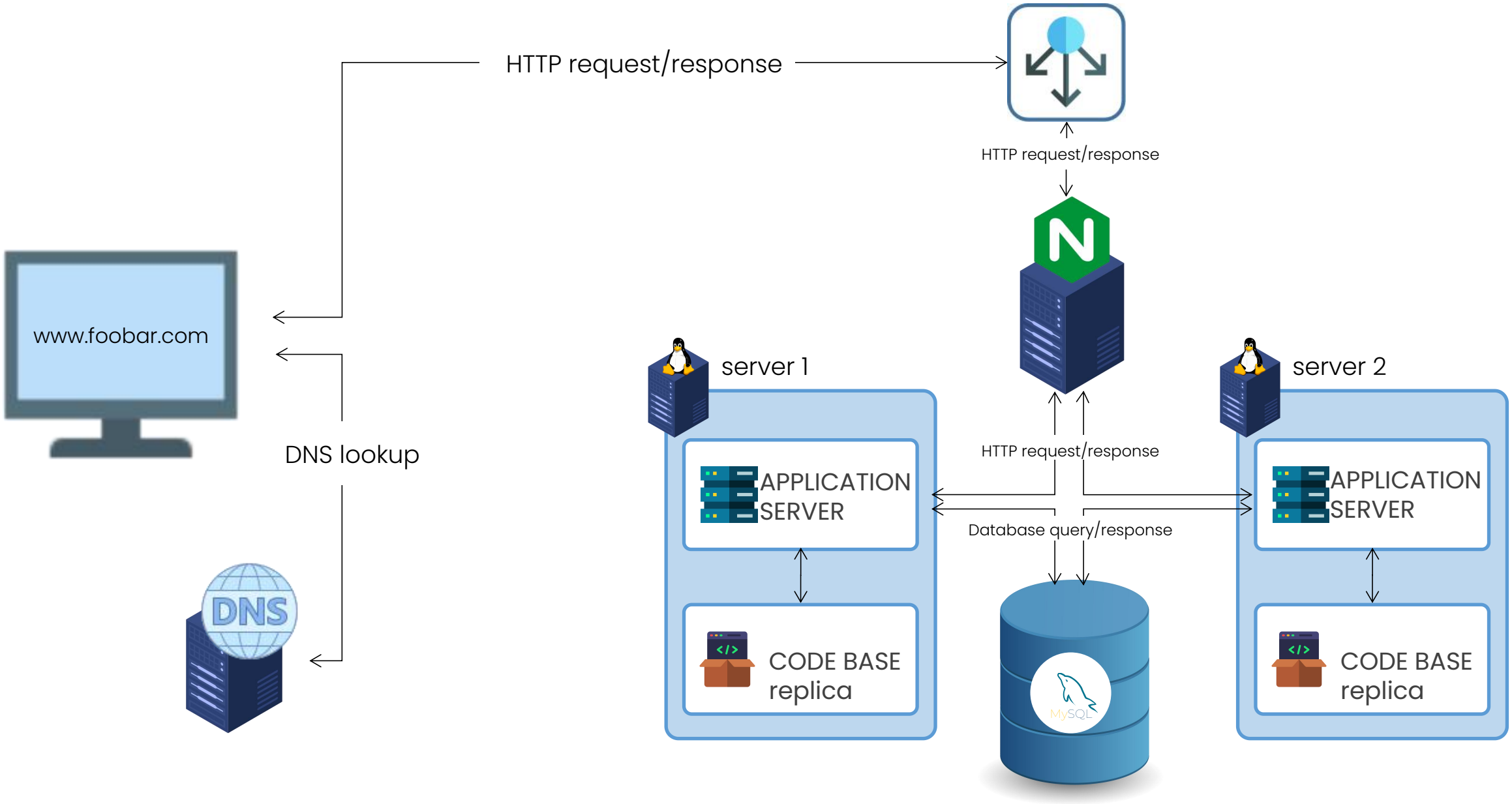
- The user's computer sends an HTTP request to the Nginx web server on the server.
- Nginx forwards the request to the application server.

- The application server interacts with the MySQL database by sending queries and receiving responses.

- The application server processes the request, generates an HTTP response, and sends it back to Nginx.

- Nginx forwards the response to the user's computer, completing the request-response cycle.

- The domain name is associated with the server through a DNS lookup, allowing users to access the website using the www.foobar.com address.


HTTP request/response
HTTP request/response
www.foobar.com
DNS lookup
DNS
server 1
server 2
APPLICATION SERVER
APPLICATION SERVER
HTTP request/response
Database query/response
CODE BASE replica
CODE BASE replica
MySQL

Explanation:

User Accessing the Website:
The user wants to access the website hosted at www.foobar.com.

Server:
The server is a physical or virtual machine responsible for hosting the web infrastructure.
It has an IP address (e.g., 8.8.8.8) assigned to it.
The server acts as the central component of the infrastructure, handling incoming requests and serving the website's content.

Domain Name:
The domain name (e.g., foobar.com) is the human-readable address used to identify the website.
It provides a more user-friendly way to access the website instead of using the server's IP address directly.

DNS Record:
The www record in www.foobar.com is a DNS (Domain Name System) record.
It is a type of DNS record called a CNAME (Canonical Name) record.
The www record points to the server's IP address (e.g., 8.8.8.8), mapping the domain name to the server.

Web Server (Nginx):
The web server (e.g., Nginx) handles HTTP requests and responses.
It acts as an intermediary between the user's web browser and the application server.
The web server is responsible for serving static files, handling SSL/TLS encryption, load balancing, and caching.

Application Server:
The application server hosts the application code base.
It executes the dynamic logic of the website and generates responses based on user requests.
The application server communicates with the web server to process requests and send responses back.

Application Files:

The application files contain the website's codebase, including HTML, CSS, JavaScript, and server-side scripts.
These files are stored on the server and are accessed and executed by the application server when processing user requests.

Database (MySQL):

The database (e.g., MySQL) stores and manages the website's data.
It provides a structured way to store and retrieve information needed by the application.
The application server communicates with the database to perform operations such as storing user data, retrieving content, etc.
Communication with User's Computer:

The server communicates with the user's computer over the internet using the HTTP protocol.
When the user requests a webpage, their computer sends an HTTP request to the server's IP address.
The server processes the request and sends an HTTP response back to the user's computer, containing the requested webpage content.

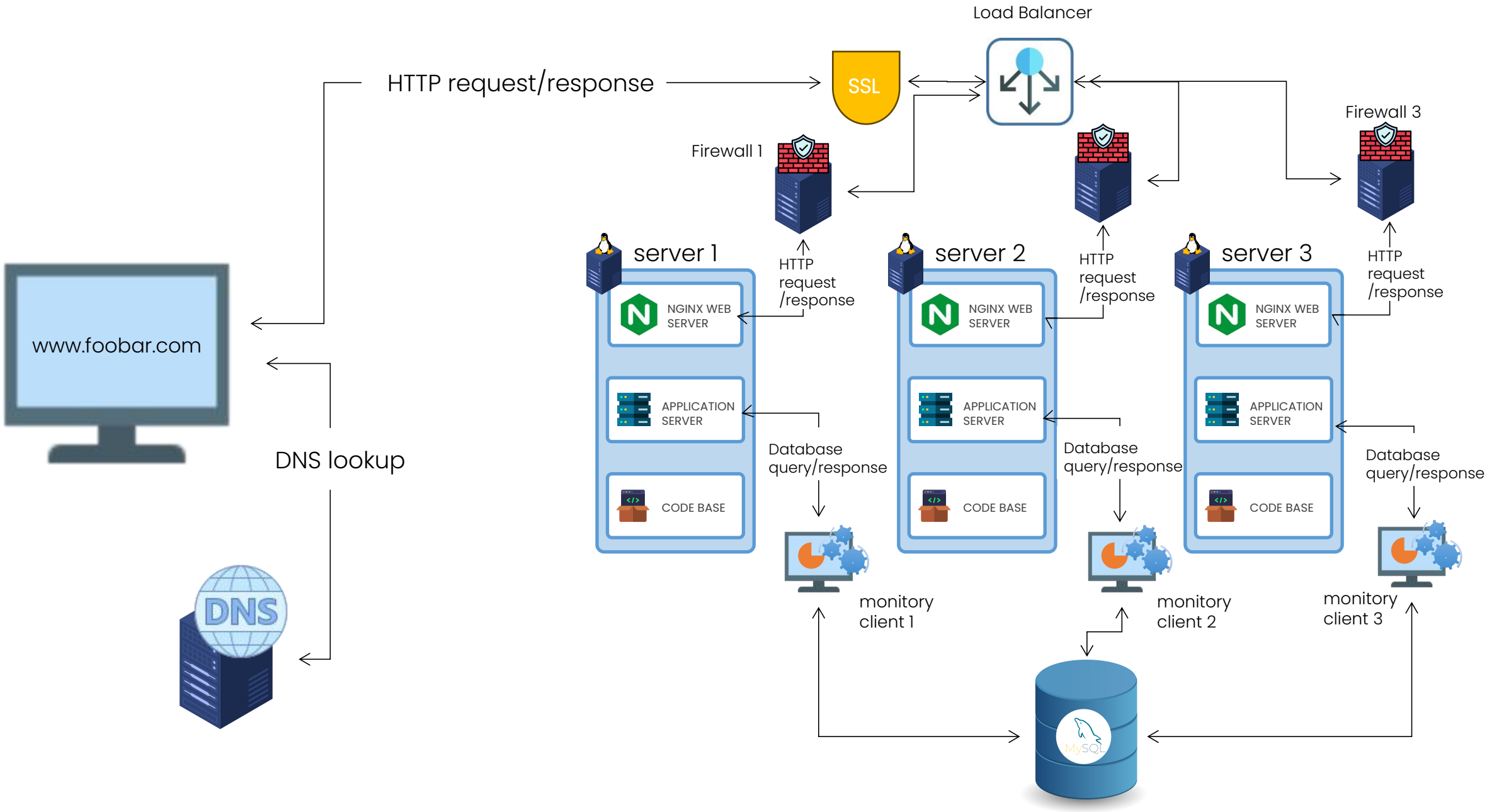
Issues with the Infrastructure:

SPOF (Single Point of Failure):

The infrastructure relies on a single server, which creates a single point of failure.
If the server experiences hardware or software issues, the entire website becomes inaccessible.
Downtime during Maintenance:

Performing maintenance tasks, such as deploying new code or restarting the web server, can cause downtime.
During maintenance, the website may be temporarily unavailable to users.
Scalability Limitations:

With a single server, the infrastructure may struggle to handle a large influx of incoming traffic.
If the website experiences high traffic, it may result in performance issues or even downtime.


Load Balancer
HTTP request/response
SSL
Firewall 1
Firewall 3
www.foobar.com
DNS lookup
DNS
server 1
server 2
server 3
NGINX WEB SERVER
APPLICATION SERVER
CODE BASE
HTTP request /response
Database query/response
monitory client 1
monitory client 2
monitory client 3
MySQL

Explanation:

- The user's computer sends an HTTP request to the load balancer for the domain name www.foobar.com.
- The load balancer distributes the request to Server 1 and Server 2 using a specific distribution algorithm (such as round-robin, least connections, etc.).
- Nginx on each server receives the request and forwards it to the application server.
- The application server interacts with the MySQL database and its replica by sending queries and receiving responses.
- The application server processes the request, generates an HTTP response, and sends it back to Nginx.
- Nginx forwards the response to the load balancer, which in turn forwards it to the user's computer, completing the request-response cycle.

Specifics about the infrastructure:

Load balancer (HAproxy):
Added to distribute incoming traffic across multiple servers, ensuring better performance and availability.
Database Replica: Added to achieve database replication and provide redundancy. It allows for load balancing of database queries and provides high availability in case of failure.

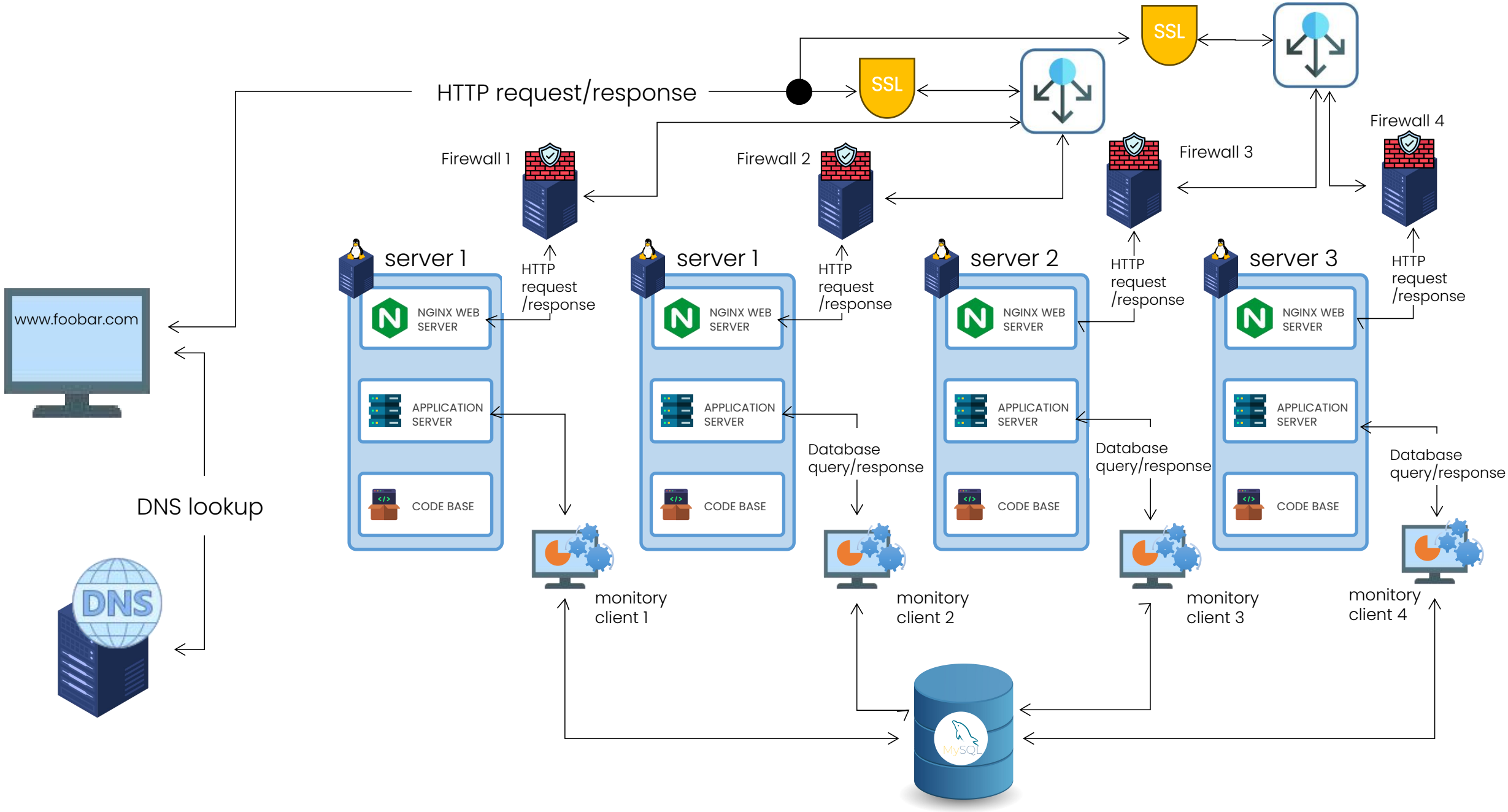
Distribution algorithm:
The load balancer is configured with a specific distribution algorithm (e.g., round-robin, least connections) to evenly distribute incoming requests among the available servers.

Active-Active vs. Active-Passive setup:
The load balancer enables an Active-Passive setup, where only one server actively handles incoming requests while the others remain passive. In an Active-Active setup, multiple servers actively handle requests simultaneously.

Database Primary-Replica (Master-Slave) cluster:
The primary node in the database cluster is responsible for processing write operations (e.g., insert, update) and forwarding the changes to the replica node. The replica node synchronizes with the primary node and serves read operations to offload the primary node and provide redundancy.
Difference between Primary and Replica nodes:

Difference between Primary and Replica nodes:
The primary node in the database cluster handles write operations and serves as the authoritative source of data. The replica node replicates the data from the primary node and primarily serves read operations to distribute the load and improve performance.
Issues with this infrastructure:

Single Point of Failure (SPOF): If any component (server, load balancer, database) fails, it can disrupt the availability of the website.
Lack of security measures: There is no mention of a firewall or HTTPS, leaving the infrastructure vulnerable to security threats.
Absence of monitoring: Monitoring tools are necessary to track the performance, health, and resource utilization of the servers, load balancer, and database, allowing proactive maintenance and issue detection.


HTTP request/response
SSL
SSL
Firewall 1
Firewall 2
Firewall 3
Firewall 4
www.foobar.com
DNS lookup
DNS
server 1
server 1
server 2
server 3
NGINX WEB SERVER
APPLICATION SERVER
CODE BASE
HTTP request /response
Database query/response
monitory client 1
monitory client 2
monitory client 3
monitory client 4
MySQL

Explanation:

- The user's computer sends an HTTPS request to Firewall 1 for the domain name www.foobar.com.
- The request passes through multiple firewalls for security and access control purposes.
- The load balancer terminates SSL/TLS encryption using an SSL certificate (SSL) and distributes the request to Server 1, Server 2, and Server 3.
- Nginx web servers on each server handle the request, interact with the application servers, and communicate with the MySQL database.
- Monitoring clients collect monitoring data from the web servers and the database to track performance, health, and resource utilization.

Specifics about the infrastructure:

Firewalls:

Added to provide network security by filtering and controlling incoming and outgoing traffic based on predefined rules.
SSL Certificate: Used to enable secure communication (HTTPS) between the user's computer and the web infrastructure, encrypting the data in transit.

Monitoring:

Monitoring tools (Sumologic or others) are used to collect data from the web servers and the database, allowing real-time monitoring of performance, availability, and potential issues.
Monitoring Data Collection: The monitoring clients gather data from the web servers and the database by capturing metrics, logs, and events related to system performance, errors, and usage patterns.

Issues with the infrastructure:

- Terminating SSL at the load balancer level: This can be an issue if there is a need for end-to-end encryption, especially when traffic is sent between the load balancer and the web servers.
- Single MySQL server capable of accepting writes: It poses a single point of failure and can limit scalability and high availability. Implementing a primary-replica (master-slave) cluster with multiple MySQL replicas can address this issue.
- Identical server components: Having all servers with the same components (database, web server, and application server) can lead to a lack of redundancy and increase the risk of a single point of failure. Introducing diversity in the infrastructure can enhance resilience.