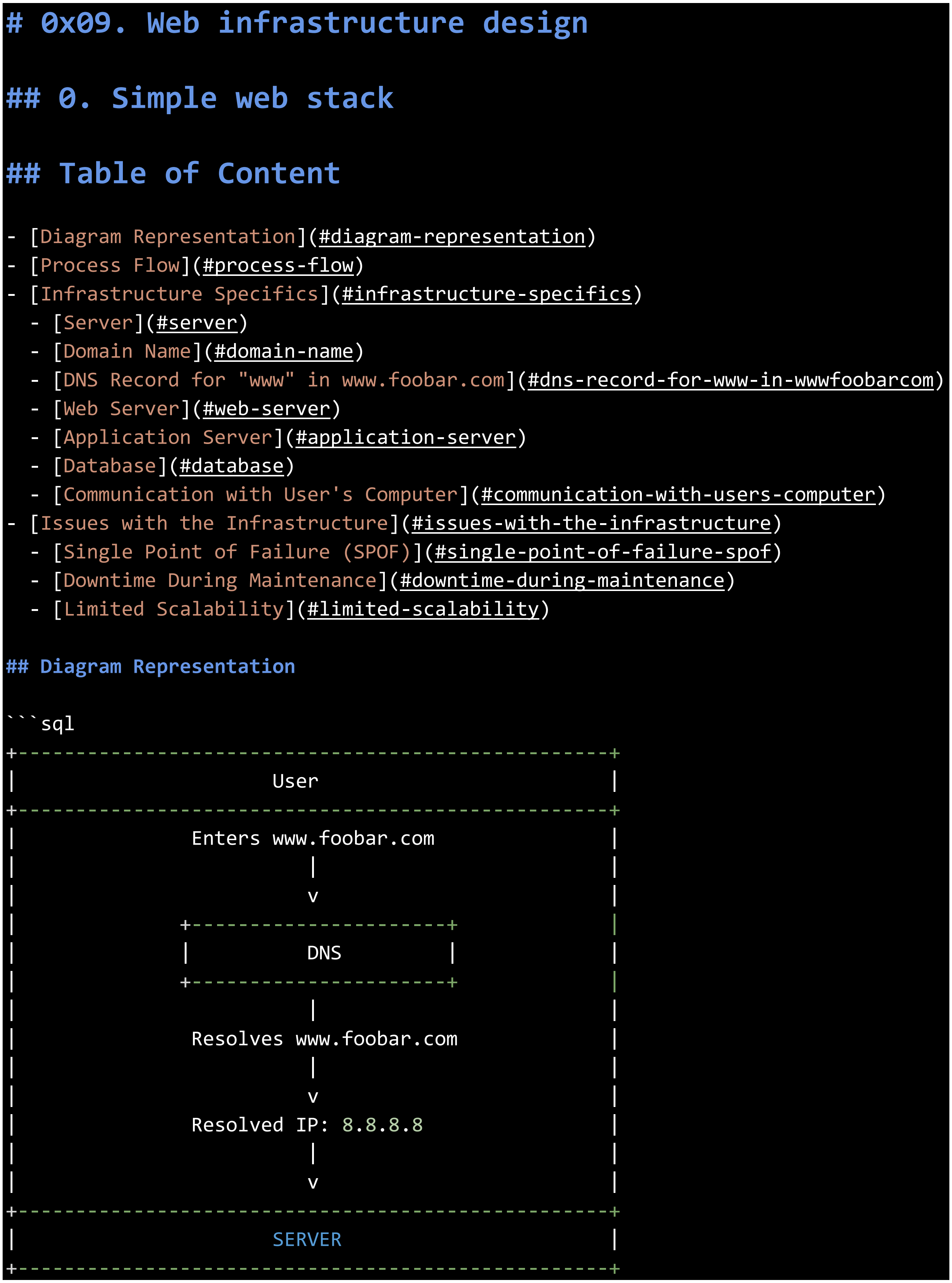


~~~
# 0x09. Web infrastructure design

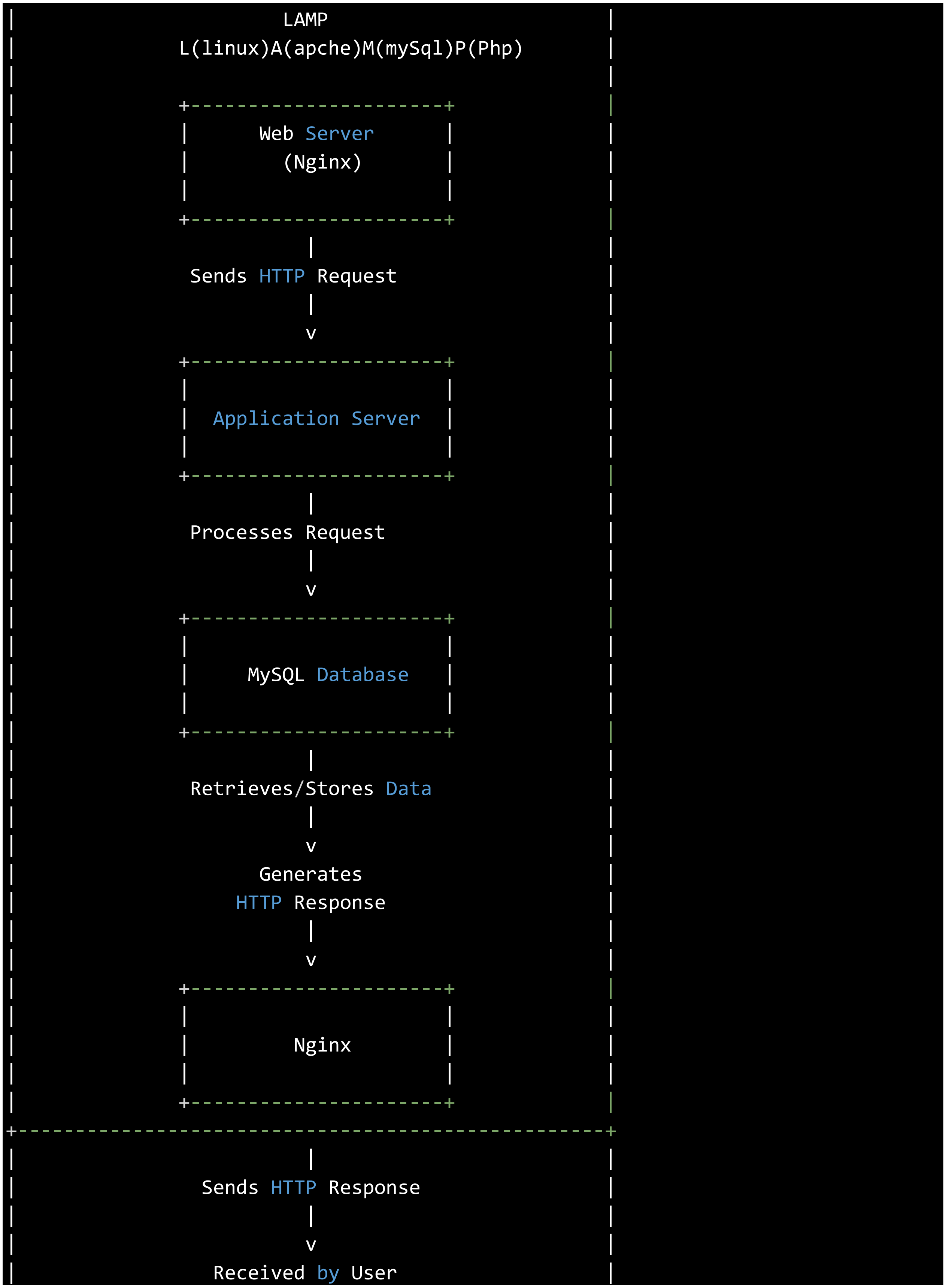
## 0. Simple web stack

## Table of Content

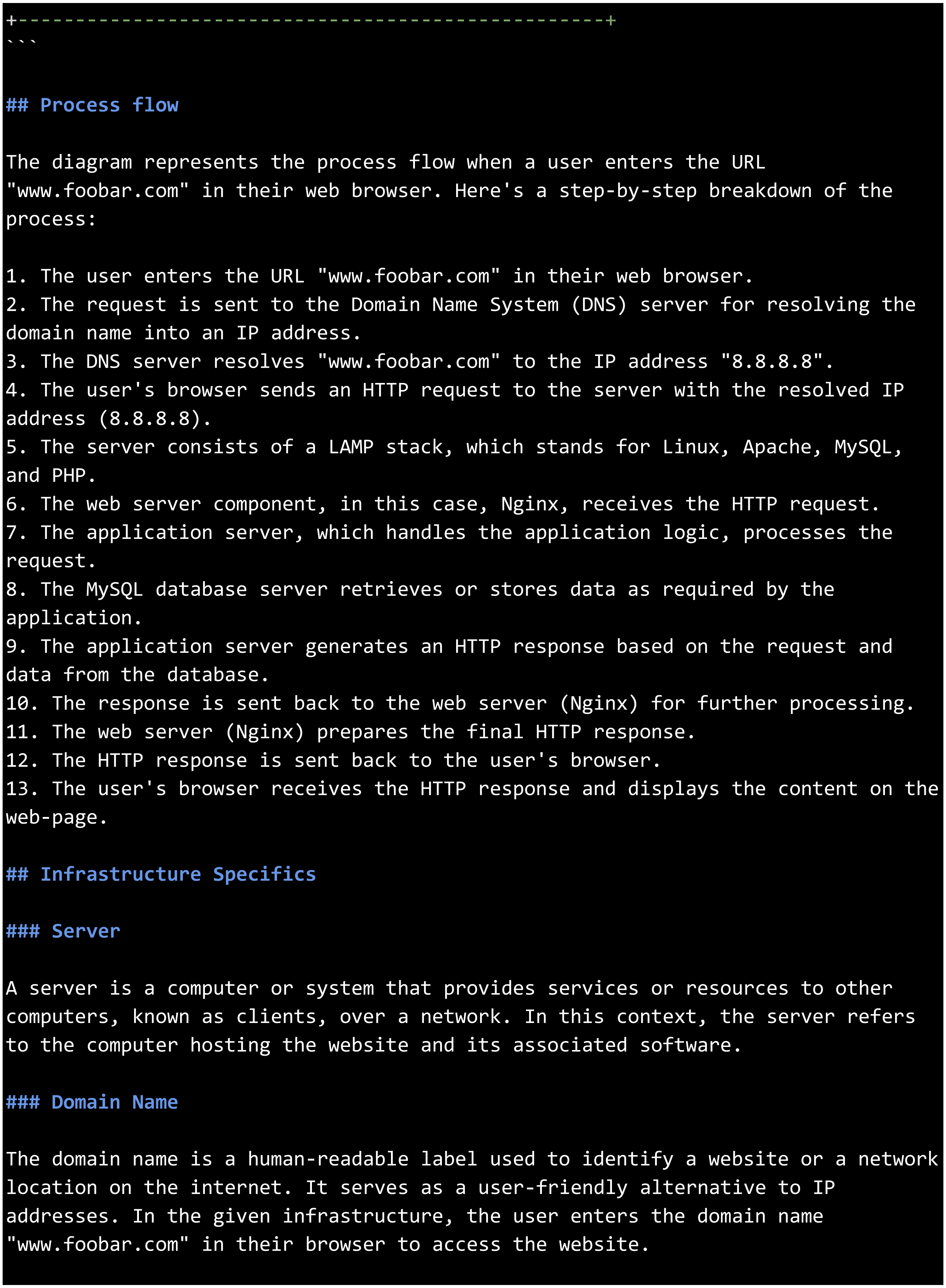
- [Diagram Representation](#diagram-representation)
- [Process Flow](#process-flow)
- [Infrastructure Specifics](#infrastructure-specifics)
  - [Server](#server)
  - [Domain Name](#domain-name)
  - [DNS Record for "www" in www.foobar.com](#dns-record-for-www-in-wwwfoobarcom)
  - [Web Server](#web-server)
  - [Application Server](#application-server)
  - [Database](#database)
  - [Communication with User's Computer](#communication-with-users-computer)
- [Issues with the Infrastructure](#issues-with-the-infrastructure)
  - [Single Point of Failure (SPOF)](#single-point-of-failure-spof)
  - [Downtime During Maintenance](#downtime-during-maintenance)
  - [Limited Scalability](#limited-scalability)

## Diagram Representation

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



```
|                       LAMP                        |
|              L(linux)A(apche)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                  |
```



~~~~
+---------------------------------------------------+
```

## Process flow

The diagram represents the process flow when a user enters the URL
"www.foobar.com" in their web browser. Here's a step-by-step breakdown of the
process:

1. The user enters the URL "www.foobar.com" in their web browser.
2. The request is sent to the Domain Name System (DNS) server for resolving the
domain name into an IP address.
3. The DNS server resolves "www.foobar.com" to the IP address "8.8.8.8".
4. The user's browser sends an HTTP request to the server with the resolved IP
address (8.8.8.8).
5. The server consists of a LAMP stack, which stands for Linux, Apache, MySQL,
and PHP.
6. The web server component, in this case, Nginx, receives the HTTP request.
7. The application server, which handles the application logic, processes the
request.
8. The MySQL database server retrieves or stores data as required by the
application.
9. The application server generates an HTTP response based on the request and
data from the database.
10. The response is sent back to the web server (Nginx) for further processing.
11. The web server (Nginx) prepares the final HTTP response.
12. The HTTP response is sent back to the user's browser.
13. The user's browser receives the HTTP response and displays the content on the
web-page.

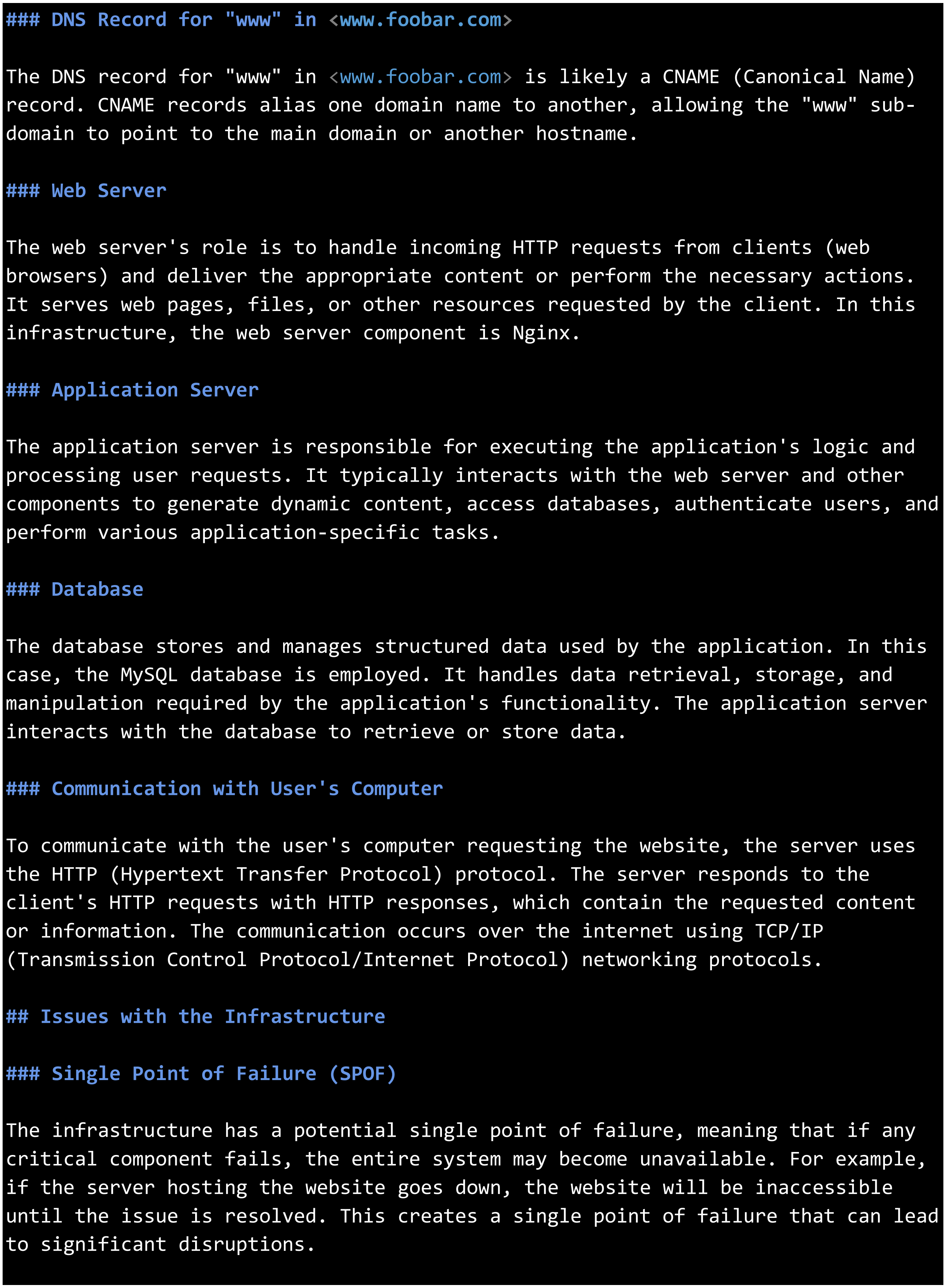
## Infrastructure Specifics

### Server

A server is a computer or system that provides services or resources to other
computers, known as clients, over a network. In this context, the server refers
to the computer hosting the website and its associated software.

### Domain Name

The domain name is a human-readable label used to identify a website or a network
location on the internet. It serves as a user-friendly alternative to IP
addresses. In the given infrastructure, the user enters the domain name
"www.foobar.com" in their browser to access the website.
~~~~



### DNS Record for "www" in <www.foobar.com>

The DNS record for "www" in <www.foobar.com> is likely a CNAME (Canonical Name) record. CNAME records alias one domain name to another, allowing the "www" sub-domain to point to the main domain or another hostname.

### Web Server

The web server's role is to handle incoming HTTP requests from clients (web browsers) and deliver the appropriate content or perform the necessary actions. It serves web pages, files, or other resources requested by the client. In this infrastructure, the web server component is Nginx.

### Application Server

The application server is responsible for executing the application's logic and processing user requests. It typically interacts with the web server and other components to generate dynamic content, access databases, authenticate users, and perform various application-specific tasks.

### Database

The database stores and manages structured data used by the application. In this case, the MySQL database is employed. It handles data retrieval, storage, and manipulation required by the application's functionality. The application server interacts with the database to retrieve or store data.

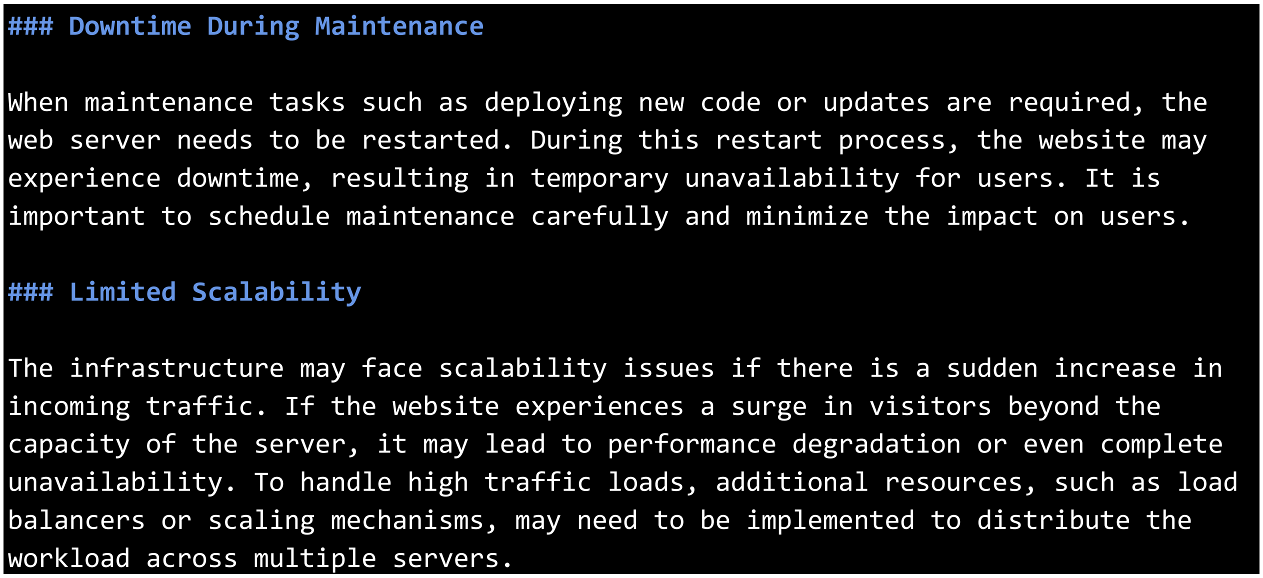
### Communication with User's Computer

To communicate with the user's computer requesting the website, the server uses the HTTP (Hypertext Transfer Protocol) protocol. The server responds to the client's HTTP requests with HTTP responses, which contain the requested content or information. The communication occurs over the internet using TCP/IP (Transmission Control Protocol/Internet Protocol) networking protocols.

## Issues with the Infrastructure

### Single Point of Failure (SPOF)

The infrastructure has a potential single point of failure, meaning that if any critical component fails, the entire system may become unavailable. For example, if the server hosting the website goes down, the website will be inaccessible until the issue is resolved. This creates a single point of failure that can lead to significant disruptions.



### Downtime During Maintenance

When maintenance tasks such as deploying new code or updates are required, the web server needs to be restarted. During this restart process, the website may experience downtime, resulting in temporary unavailability for users. It is important to schedule maintenance carefully and minimize the impact on users.

### Limited Scalability

The infrastructure may face scalability issues if there is a sudden increase in incoming traffic. If the website experiences a surge in visitors beyond the capacity of the server, it may lead to performance degradation or even complete unavailability. To handle high traffic loads, additional resources, such as load balancers or scaling mechanisms, may need to be implemented to distribute the workload across multiple servers.