# 0. Simple web stack

**What is a server:**
A server is a dedicated computer or system responsible for managing network resources and providing services to other computers, known as clients. In this context, the server is the overall machine that hosts the web server, application server, and database.

**What is the role of the domain name:**
The domain name (e.g., foobar.com) serves as a human-readable alias for the server's IP address (8.8.8.8). It allows users to access the website using a memorable name instead of remembering and typing the IP address directly.

**What type of DNS record www is in www.foobar.com:**
The DNS record for "www" in www.foobar.com is typically a CNAME (Canonical Name) record. It is used to alias the domain name to another domain, in this case, pointing www.foobar.com to foobar.com.

**What is the role of the web server:**
The web server (Nginx in this case) handles incoming HTTP requests from users' browsers and serves static content like HTML, CSS, and images. It can also manage SSL/TLS encryption and handle redirections.

**What is the role of the application server:**
The application server executes the server-side logic of the web application. It processes dynamic content, interacts with databases, and performs tasks requested by users. It works in conjunction with the web server to generate and deliver complete web pages.

**What is the role of the database:**
The database (MySQL in this case) stores and manages the structured data used by the web application. It stores information such as user accounts, content, and other relevant data. The application server interacts with the database to retrieve or update information as needed.

**What is the server using to communicate with the computer of the user requesting the website:**
The server communicates with the user's computer through the HTTP (Hypertext Transfer Protocol) or HTTPS (HTTP Secure) protocol. These protocols define how data is transmitted between the server and the user's browser.

# Issues with the Infrastructure:

**Single Point of Failure (SPOF):**
The infrastructure has a potential single point of failure if the server goes down, impacting all components. To mitigate this, redundancy measures, such as using multiple servers or cloud-based solutions, should be considered.

**Downtime during Maintenance:**
Deploying new code or performing maintenance tasks, especially restarting the web server, can result in downtime. Implementing strategies like load balancing and rolling deployments can help minimize service interruptions during updates.

**Cannot Scale if too much Incoming Traffic:**
With only one server, scalability becomes a challenge, and the infrastructure may struggle to handle a significant increase in incoming traffic. Scaling options, such as load balancing and horizontal scaling, should be explored to address this limitation.