0-simple_web_stack

Step-by-Step Explanation:

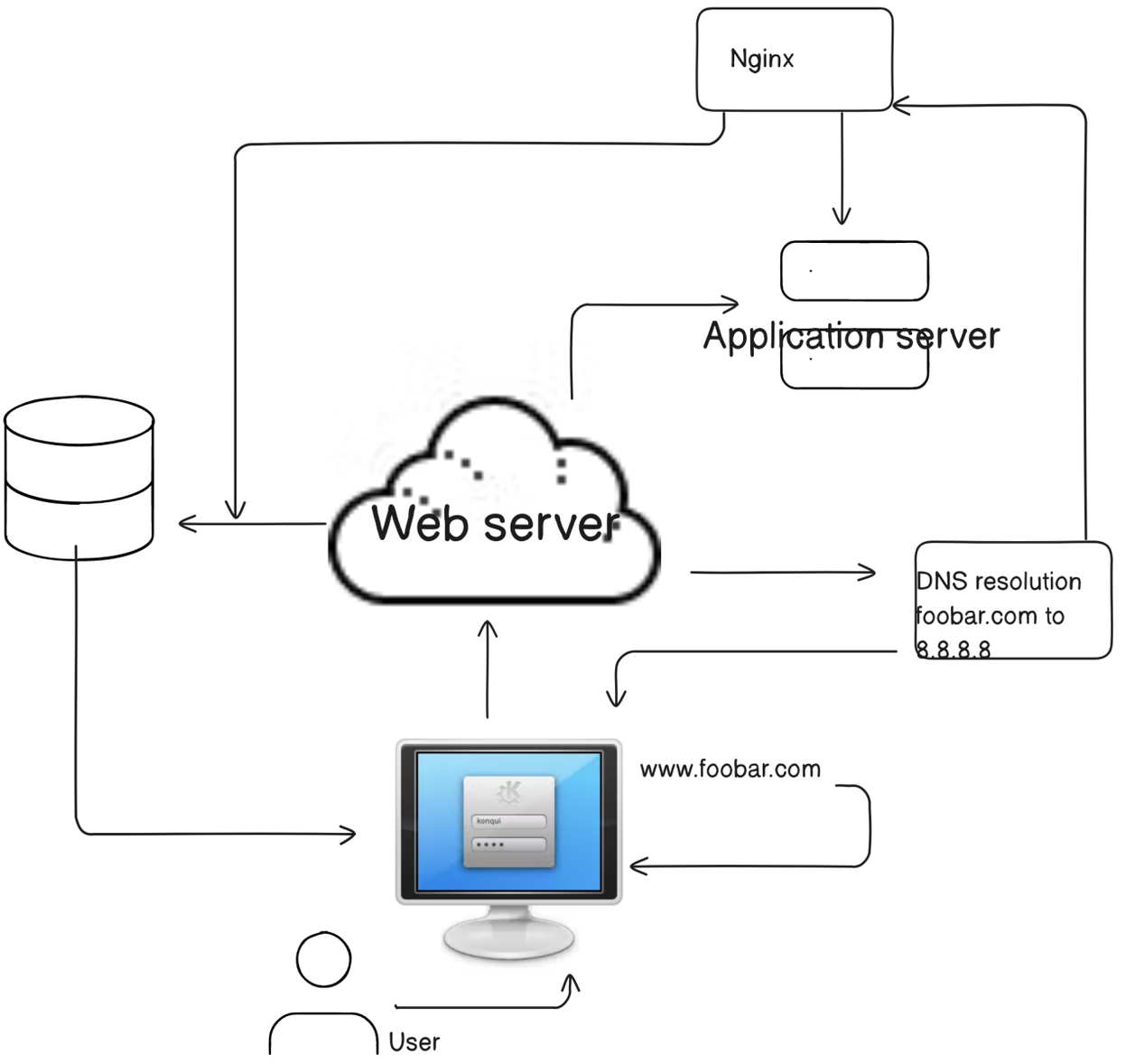
1. **Starting Point - User Request:**
   - A user opens a web browser and types `www.foobar.com` .
   - The browser sends a request to the internet to find the server associated with this domain.
2. **Domain Name and DNS Record:**
   - The domain `foobar.com` is registered with a DNS provider.
   - The DNS has an 'A' record for `www` that points to the IP address `8.8.8.8` .
   - **Role of Domain Name:** It acts as an easy-to-remember address for a website, which gets translated to an IP address.
   - **DNS Record Type:** `www` is an 'A' record, which maps the domain name to an IPv4 address.
3. **Reaching the Server:**
   - The IP address `8.8.8.8` directs the request to your server.
   - **What is a Server:** It's a powerful computer hosting resources and services - in this case, your website.
4. **Web Server - Nginx:**
   - The server runs Nginx, a web server software.
   - **Role of Web Server:** It receives the request, looks for the requested webpage, and sends it back to the user's browser. It serves static content like HTML, CSS, and JavaScript files.
5. **Application Server:**
   - If the request needs dynamic content, Nginx communicates with the application server.
   - The application server runs your website's backend code (e.g., PHP).
   - **Role of Application Server:** It processes business logic, user authentication, interacts with the database, and dynamically generates web pages.
6. **Database - MySQL:**
   - The application server retrieves or stores data in the MySQL database as needed.
   - **Role of Database:** Stores and manages data (like user profiles, posts, etc.) that the application needs to function.
7. **Communication Protocol:**
   - The server communicates with the user's computer using HTTP/HTTPS protocol.
8. **Issues with This Infrastructure:**
   - **Single Point of Failure (SPOF):** If the server goes down, the entire website is inaccessible.

- **Downtime for Maintenance:** When updating code or performing maintenance, the website may need to go offline temporarily.
- **Scalability Issues:** If traffic increases significantly, this single server setup might not handle the load efficiently.

Diagram on the Whiteboard:

Let's create a simple diagram to illustrate this setup:

1. User's computer with a browser requesting `www.foobar.com` .
2. The DNS system resolving `www.foobar.com` to `8.8.8.8` .
3. The server with its components: Nginx, application server, application files, and MySQL database.


Nginx
Application server
Web server
DNS resolution
foobar.com to
8.8.8.8
www.foobar.com
User