

ALX PROJECT

0x09-web_infrastructure_design

0-simple_web_stack

- A user wants to access the website hosted at `www.foobar.com`.

1. Components:

- **Server:** We'll use a single server to host all components.
- **Web Server (Nginx):** Nginx will handle incoming HTTP requests and serve static content (like HTML, CSS, and images).
- **Application Server:** This server will execute dynamic code (e.g., PHP, Python, or Node.js) and generate dynamic content.
- **Application Files (Code Base):** The application files (your code) reside on the server and are executed by the application server.
- **Database (MySQL):** MySQL stores data (e.g., user accounts, blog posts, etc.).

2. Domain Name and DNS:

- **Domain Name (foobar.com):** The domain name is the human-readable address (like a signpost) for our website.
- **Role of Domain Name:** It provides a memorable way for users to access our site.
- **DNS Record for www:** The DNS record for `www.foobar.com` is an A record pointing to the server's IP address (e.g., `8.8.8.8`).

3. Roles of Components:

- **Web Server (Nginx):**
 - Serves static files (HTML, CSS, images).
 - Handles SSL/TLS encryption (HTTPS).
- **Application Server:**
 - Executes dynamic code (e.g., PHP, Python, or Node.js).
 - Generates dynamic content (e.g., personalized pages).
- **Database (MySQL):**
 - Stores and manages data (e.g., user profiles, blog posts).
 - Accessed by the application server.

4. Communication:

- When a user requests `www.foobar.com`, the following happens:
 - DNS resolves `www.foobar.com` to the server's IP (`8.8.8.8`).
 - User's browser sends an HTTP request to the server.
 - Nginx (web server) handles the request.
 - If it's a static file (e.g., an image), Nginx serves it directly.
 - If it's dynamic (e.g., a user profile), Nginx forwards the request to the application server.
 - The application server processes the request, retrieves data from the database, and generates a response.
 - The response is sent back to the user's browser.

5. Issues with this Infrastructure:

- **Single Point of Failure (SPOF):** If the server fails, the entire website goes down.
- **Downtime during Maintenance:** When deploying new code, the web server (Nginx) needs to be restarted, causing temporary downtime.
- **Scalability Limitations:** With only one server, handling high traffic becomes challenging. Scaling horizontally (adding more servers) is difficult.