What is a server?

A server is like a computer that stores and shares data or services with other computers over a network. In our case, it's the machine hosting our website files and serving them to people who want to see our site.

What is the role of the domain name?

A domain name is the web address people use to find our website. It's like the street address for a house on the internet. Instead of using numbers like IP addresses, we use names that are easier for people to remember, like foobar.com.

What type of DNS record is "www" in www.foobar.com?

The "www" in www.foobar.com is like a nickname for the main domain. It's a type of record in the DNS (Domain Name System) called a CNAME, which helps direct traffic from the "www" subdomain to the main domain.

What is the role of the web server?

The web server handles requests from people's web browsers when they want to see our website. It's like the waiter at a restaurant, taking orders and bringing back what people asked for. In our setup, we're using a web server called Nginx to do this job.

What is the role of the application server?

The application server is where the dynamic parts of our website live. It's like the chef in a kitchen, taking the orders from the waiter (web server) and preparing the food (dynamic content) for the customers (users). In our setup, it could be something like Gunicorn for handling Python code.

What is the role of the database?

The database is like the pantry where we keep all the ingredients (data) for our website. It stores things like user information, blog posts, and product details. MySQL, in our case, is the software that helps us organize and manage this data.

What is the server using to communicate with the computer of the user requesting the website?

The server communicates with the user's computer using a protocol called HTTP (Hypertext Transfer Protocol). It's like the language they use to talk to each other over the internet. When a user's browser wants to see a page on our website, it sends an HTTP request to our server, which then sends back the requested webpage.

What is the issue with Single Point of Failure (SPOF) in this infrastructure?

SPOF means that if anything goes wrong with our single server, like a hardware failure or software crash, the entire website will go down. We don't have a backup or another server to take over, so if this one fails, the website becomes unavailable.

What is the issue with downtime when maintenance is needed, such as deploying new code requiring the web server to be restarted?

When we need to update the website or deploy new features, we have to restart the web server. During this time, the website is offline, and users can't access it. This downtime can frustrate users and disrupt their experience on the site.

What is the issue with scalability if there is too much incoming traffic?

Our infrastructure can't handle a sudden increase in visitors. If too many people try to access the website at once, it could slow down or crash. We don't have a way to quickly add more resources or servers to manage the extra load, so the website may struggle to keep up with demand.