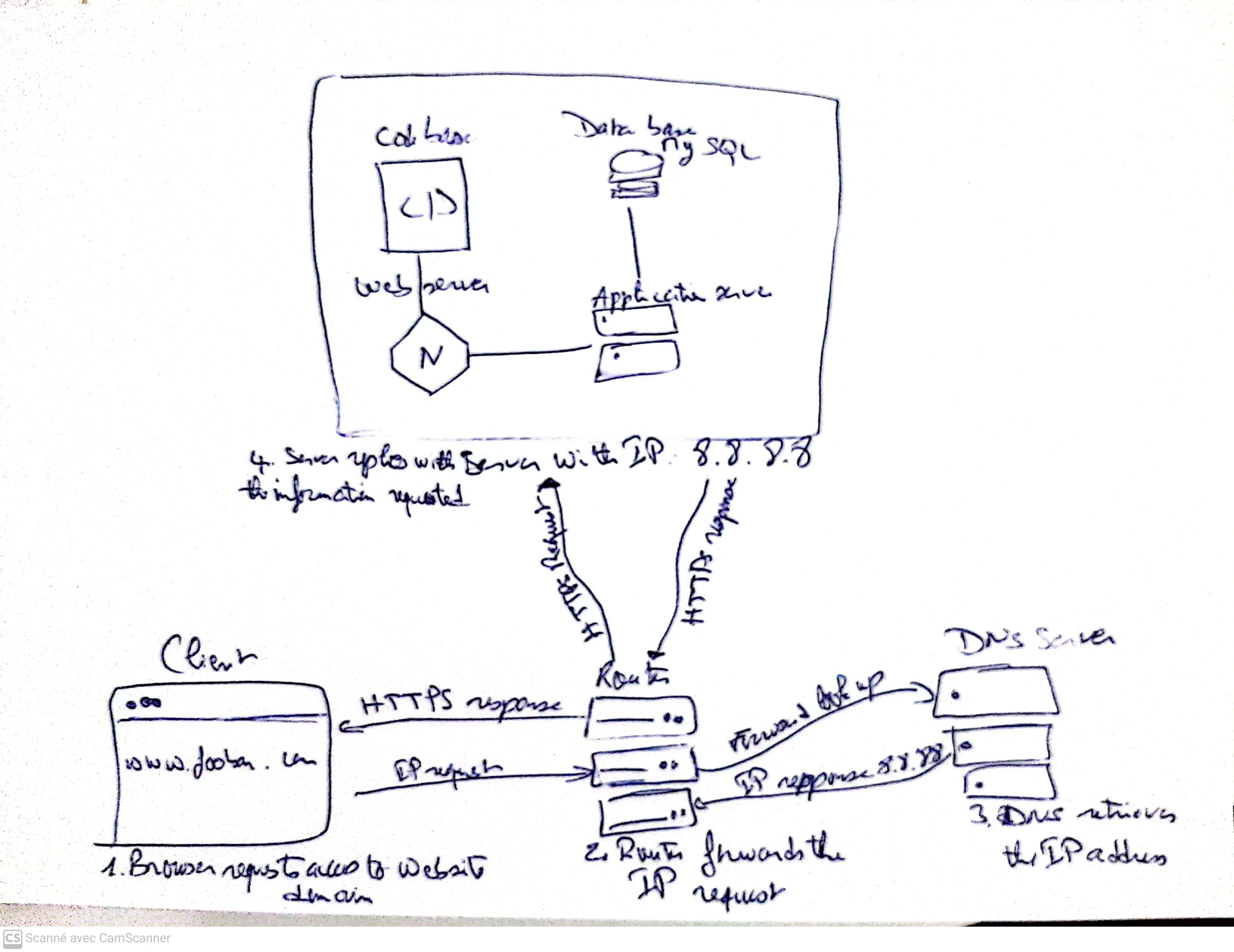
### Task 0 : Simple web stack



To design a one-server web infrastructure that hosts the website reachable via www.foobar.com, let's start with the user's request to access the website. When a user wants to access the website hosted on the one-server web infrastructure, they enter the URL "www.foobar.com" in their web browser. The domain name "foobar.com" acts as a user-friendly representation of the server's IP address, which in this case is 8.8.8.8. The DNS record associated with the domain name includes a CNAME (Canonical Name) record for the "www" subdomain, ensuring that requests for "www.foobar.com" are directed to the server's IP address. Upon receiving the request, the server, which is a physical or virtual machine, plays a central role in the entire process. It houses all the components required for the web infrastructure. The web server, in this case, Nginx, receives the HTTP request from the user's browser and listens on port 80 (HTTP) or 443 (HTTPS). It acts as the intermediary between the user and the application server. The application server, running the website's code, processes the dynamic content, interacts with the web server, and generates a response based on the user's request. The application files, including the code base and other necessary files, are stored on the server and executed by the application server. If the website requires data storage, a MySQL database is employed to handle structured data. When the server receives the request, it communicates with the user's computer via the internet using HTTP. It processes the request, generates a response, and sends it back to the user's computer, allowing the website to be displayed in their web browser.

1. User Request: A user wants to access the website www.foobar.com.
2. Domain Name: The domain name, foobar.com, serves as a human-readable address for the website. It helps users easily identify and access the website. The "www" in www.foobar.com is a subdomain and typically represents the web server for the domain.
3. DNS Record: The www in www.foobar.com represents a DNS CNAME (Canonical Name) record. It is used to alias the www subdomain to the main domain. In this case, the DNS record for www.foobar.com should be configured as a CNAME record pointing to the server's IP address, which is 8.8.8.8 in this example.
4. Server: A server is a computer or a system that hosts and delivers resources, such as websites, to other computers over a network, in this case, the Internet. It provides the necessary infrastructure to store and serve the website's files and data.
5. Web Server (Nginx): The web server is responsible for handling incoming HTTP requests from clients, such as web browsers, and delivering the appropriate responses. In this infrastructure, we are using Nginx as the web server software.
6. Application Server: The application server hosts the website's codebase and is responsible for executing the application logic and generating dynamic content. It processes requests received from the web server and communicates with the database server, if necessary, to fetch or store data.
7. Application Files: The application files contain the website's codebase, including HTML, CSS, JavaScript, and any server-side code. These files are hosted on the application server and are responsible for generating the website's pages and responding to user requests.
8. Database (MySQL): The database stores and manages the website's data, such as user information, content, and settings. MySQL is used as the database management system in this infrastructure.
9. Communication with User's Computer: To communicate with the user's computer, the server uses the Internet Protocol (IP) to establish a connection. When the user's computer sends a HTTP request to access www.foobar.com, the request is transmitted over the network using IP packets, and the server responds with the appropriate content.

Issues with the Infrastructure:

a) SPOF (Single Point of Failure):

This infrastructure has a single server, which means it has a single point of failure. If the server goes down, the website becomes inaccessible. This creates a risk of downtime and potential loss of business.

b) Downtime during Maintenance:

When performing maintenance tasks, such as deploying new code, the web server needs to be restarted. During this downtime, the website may become unavailable to users. This can result in a negative user experience and potential loss of revenue.

c) Limited Scalability:

With only one server in the infrastructure, it becomes challenging to handle a large influx of incoming traffic. As the traffic grows, the server may become overloaded, causing slow response times or even crashes. The lack of scalability can hinder the website's performance and scalability.