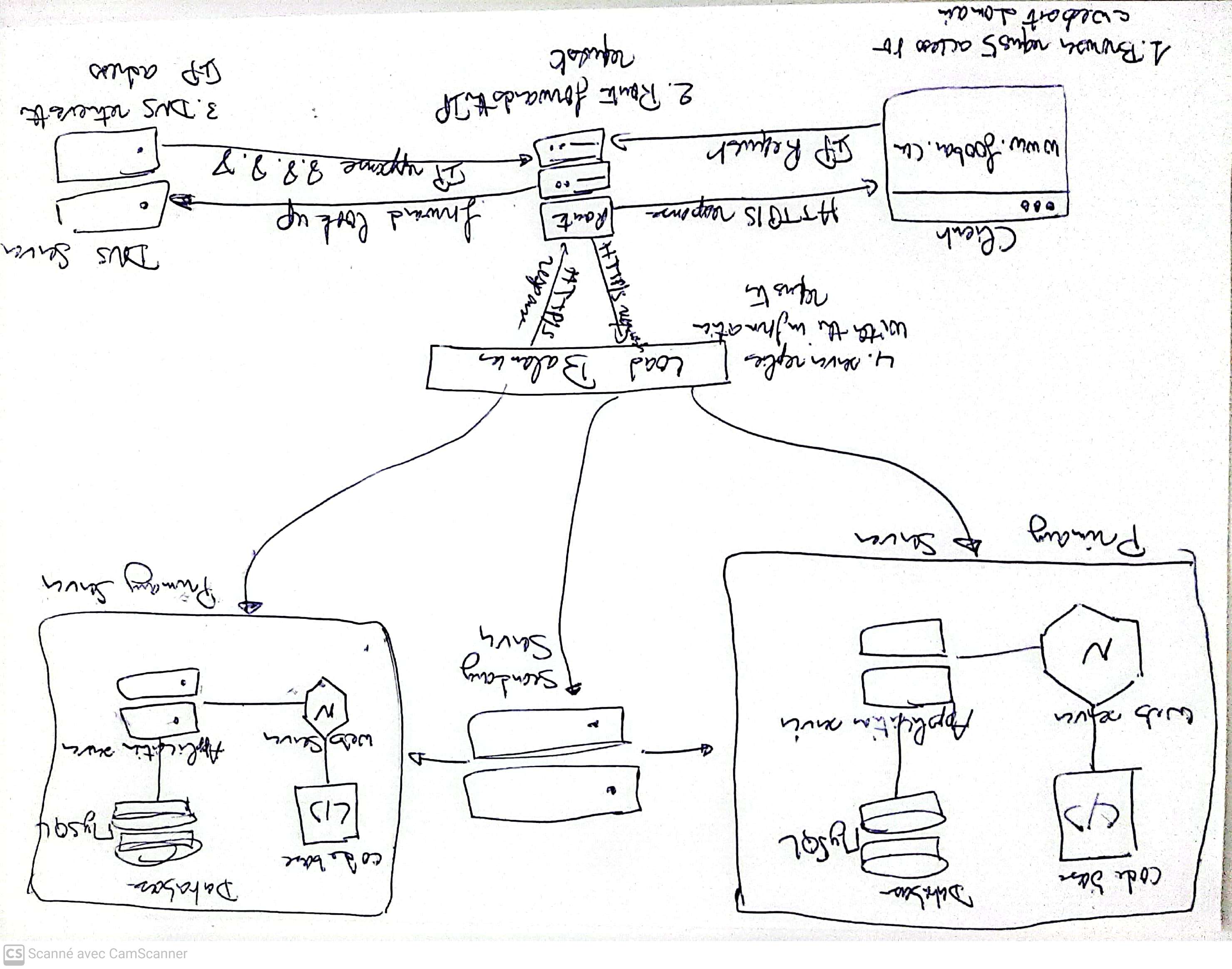
1. Distributed web infrastructure



The new configuration is composed of two master servers and one slave-servers. As the master servers are going to be working based on an Active-Active setup, their configuration must be identical, therefore we need to add every additional element as the simple web infrastructure we had in the previous point. The load is going to be managed through a load-balancer, which distributes the queries according to a Robin-Round algorithm. Finally, an additional server will be needed to serve a replica or slave server, helping to unload the masters’ servers reading queries.

The distribution algorithm that the load balancer is configured with and how it works: Our load-balancer is using a Round Robin algorithm distribution. Meaning the queries requested are distributed to every server sequentially one after another. And after sending the request to the last server, the algorithm starts from the first server. This will bring on average and approximately, to a server load distribution of 50% on each of the two servers configurations.

Is the load-balancer enabling an Active-Active or Active-Passive setup, what is the difference between both: Our load-balancer is enabling an Active-Active setup.

The Active-Active cluster is typically made up of at least two nodes, both actively running the same type of services at the same time. Their purpose is to achieve load balancing by distributing tasks to different servers in order to prevent overload. As there is more than one server (nodes) available to severe, the service time and process throughput can have improvements.

On the other hand, the Active-Passive setup is also made up of at least two nodes (servers); however, not all nodes are going to be active simultaneously. In this configuration, while one node is active, the other nodes (failover servers) are passively waiting to be active as a backup in case the primary server (the one being in use actively) is disconnected or unable to serve. Under this configuration, and as in the Active-Active setup, it is important that primary and failover nodes have the exact server configuration, so clients will not be able to tell the difference when the failover server takes over the operation.

How a database Primary-Replica (Master-Slave) cluster works: A database Primary-Replica (Master-Replica) is a mechanism that enables data of one database server (the master) to be replicated or be copied to one or more computers or database servers (the slaves), in order all users share the same level of information. This process leads to a distributed database in which users can quickly access data without interfering with each other.

The database replication process can either be synchronous or asynchronous. In the first one, the replication process is done from the client-server to the model server and then replicated to all the replica servers before the client is notified about the data replication. This method of replication may take longer to verify, however, all data was copied before proceeding.

As in the asynchronous replication process, replication is done by sending data from the client to the model server, followed by a confirmation order to the client, who finally gives permission of copying to the replicas at an unspecified or monitored pace.

What is the difference between the primary node and the replica node in regard to the application:

One of the main differences between the primary node and the replica node, regarding the application, is that the primary database is regarded as the authoritative source, while the replica database is synchronized to it. The primary node serves as the keeper of information, here the “real” data is kept, and then writing only happens here. On the other hand, reading only occurs in the replica or slave node. This architecture’s purpose is due to safeguard site reliability. In case a site receives a lot of traffic, a replica node prevents the overloading of the master node with reading and writing requests. This eases the load of the entire system preventing it to collapse.

Issues with the Infrastructure:

a) Single Points of Failure (SPOF):

Although we have added redundancy in the form of multiple servers, there are still some potential SPOFs in this infrastructure. For example, if the load balancer, it can cause service disruption or downtime.

b) Security Issues:

The infrastructure lacks essential security measures such as a firewall and HTTPS. This leaves the system vulnerable to unauthorized access, data breaches, and potential security threats.

c) Lack of Monitoring:

The absence of a monitoring system means there is no real-time visibility into the performance, availability, and health of the servers, load balancer, and database cluster. Monitoring is crucial for identifying issues, optimizing performance, and ensuring the overall stability of the infrastructure.