

* **For additional element, why you are adding it?**

One slave server and two master servers make up the new arrangement. Because the master servers will operate using an Active-Active configuration, every extra component must be added to our primary web infrastructure in the previous point. A load balancer will manage the load by distributing the inquiries using the Robin-Round method. To relieve the master servers' reading queries, a replica or slave server will eventually require its server.

* **What distribution algorithm your load balancer is configured with, and how it works**

Our load-balancer distributes its workload using the Round Robin method. In other words, the requested queries are delivered to each server in turn, one after the other. And the algorithm starts from the first server after submitting the request to the last server. This will result in an average server load distribution of roughly 50% on each of the two servers.

* **Is your load-balancer enabling an Active-Active or Active-Passive setup, explain the difference between both**

Our load-balancer is enabling an Active-Active setup.

Usually, the Active-Active cluster consists of two or more nodes that are both actively performing the same kind of services simultaneously. By assigning jobs to several servers, they aim to establish load balancing to prevent overload. The service time and process throughput can be improved because multiple servers (nodes) are available to use.

On the other hand, the Active-Passive architecture likewise consists of at least two nodes (servers), but not all of them will be active simultaneously. While one node is active in this arrangement, the other nodes (failover servers) are waiting passively to become active as a backup in case the primary server (the one that is being used actively) becomes disconnected or unable to provide service. With this configuration, just like in the Active-Active setup, the primary and failover nodes must have the same server configuration so that clients won't be able to distinguish between them when the failover server assumes control of the operation (Villanueva, 2017).

* **How a database Primary-Replica (Master-Slave) cluster works**

An archive A system known as Primary-Replica (Master-Replica) allows data from one database server (the master) to be duplicated to one or more other computers or database servers (the slaves), ensuring that all users have access to the same level of information. Thanks to a distributed database created through this approach, users can easily access data without interacting with one another.

a repository Data from one database server (the master) can be duplicated to one or more additional computers or database servers (the slaves) using a technique called Primary-Replica (Master-Replica), guaranteeing that all users have access to the same level of information. This method creates a distributed database that users may simply access without interacting with one another.

Replication is carried out similarly to asynchronous replication by delivering data from the client to the model server, followed by a confirmation order to the client, who then approves the copying of data to the replicas at an undetermined or observed rate (Lutkevich, 2020)

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

The replica database is synchronized to the primary database, regarded as the authoritative source. One of the most important differences between the application's primary and replica nodes is this. Writing only occurs at the central node since the "real" data is kept; it serves as the information's keeper. On the other hand, reading is restricted to the slave or replica node. The architecture's main objective is to safeguard site dependability. When a site has a lot of traffic, a replica node prevents the master node from getting overwhelmed with reading and writing requests. This lessens the total stress on the system and prevents it from failing (Theodorus, 2020).

## **Issues with the distributed web infrastructure**