Infrastructure Second Task

For every additional element, why you are adding it?

We added the load balancer to distribute the load between the three servers and by doing that we're making sure if one of the servers fails, the other can handle the situation.

Also, we added a Master-slave database configuration and the purpose of it is to have synchronous databases that contain the same data at the same time. By doing this, if the database fails, we can do our queries using the other databases.

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

Round robin load balancing is a load balancing technique that cyclically forwards client requests via a group of servers to effectively balance the server load. It works best when these servers have similar computational and storage capabilities.

Round robin is the easiest way to balance the server load and provide simple fault tolerance. In this method, multiple identical servers are set up to deliver the same services or applications. Although they all have the same internet domain name, each server has a unique IP address. A load balancer keeps a list of all the unique IP addresses linked with the internet domain name.

Round robin operates on a simple mechanism. With round robin network load balancing, connection requests are cyclically moved between servers. The requests are sequenced based on the order they're received. Let's take an example to help you understand how this works.

When servers receive session requests associated with an internet domain name, the requests are assigned randomly or in a rotating sequence. For example, the first request receives server 1's IP address; the second gets server 2's IP address, and so on. These requests resume at server 1 once all servers are assigned an access request in a cycle. This sequential movement of client requests helps keep the load balanced even during high traffic.

In a nutshell, round-robin network load balancing rotates connection requests among web servers in the order they're received. Consider an organization with a cluster of three servers: A, B, and C.

Server A receives the first request.

Server B receives the second request.

Server C receives the third request.

According to this directive, the load balancer continues to send requests to servers. This distributes the server load evenly, allowing the balancer to manage high traffic.

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

Active-active

An active-active high availability cluster distributes workloads evenly across all nodes, ensuring load balancing. An active-passive setup involves not all nodes being active, with the other node(s) on standby to take over if the active node fails, ensuring service continuity without load distribution.

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

Replication Topology: A replication topology defines the relationship between the source database and replica databases. Different replication topologies exist, including master-slave, master-master, and multi-level replication.

Source Database: The source database is the primary database that holds the original and authoritative copy of the data. It is responsible for processing read and write operations and propagating data changes to the replicas.

Replica Databases: Replica databases are copies of the source database. They receive and apply data changes from the source database to keep their data in sync. Replica databases can be located on the same server as the source database or distributed across different servers for improved performance and availability.

Data Replication: Data replication involves capturing and transmitting data changes from the source database to the replicas. When a write operation (such as an insert, update, or delete) occurs on the source database, we record the change in a replication or transaction log.

Replication Process: The replication process reads the changes recorded in the replication log and applies them to the replica databases. Depending on the database system and replication technology, we do database replication through various mechanisms, such as log-based, statement-based, or trigger-based.

Consistency and Synchronization: Maintaining data consistency and synchronization is crucial in database replication. Depending on the replication method, we can use various techniques to ensure we apply data changes to the replicas in the same order and with the same consistency as the source database.

What is the difference between the Primary node and the Replica node regarding the application?

Primary Node:

Write Operations: The primary node is the main server responsible for handling write operations, such as insert, update, and delete queries. Any changes to the data occur on the primary node first.

Consistency: In synchronous replication setups, the primary node ensures that changes made to the data are immediately propagated to the replica nodes. This ensures data consistency across the entire database cluster.

High Availability: While the primary node is handling write operations, it also plays a crucial role in maintaining high availability. If the primary node fails, a replica node can be promoted to take over as the new primary node to ensure uninterrupted service.

Replica Node:

Read Operations: Replica nodes are primarily used for handling read operations. Applications can distribute read queries across replica nodes to improve read performance and handle a larger number of concurrent read requests.

Redundancy and Fault Tolerance: Replica nodes provide redundancy and fault tolerance. If the primary node fails, one of the replica nodes can be promoted to become the new primary node, ensuring continuity of service.

Scalability: Replica nodes contribute to scalability by distributing read operations, allowing for better utilization of resources and improved overall performance.

Where are SPOF؟

The load balancer here is an SPOF, if it fails the whole system will fail.

Security issues (no firewall, no HTTPS)?

We're using HTTP not HTTPS so our communication is not encrypted, and a hacker can attack it and we're not using a firewall so some unwanted traffic can get in.

No monitoring

Our system is not monitored using anything, so if a server is about to fail, we'll not know until it fails for real, our system works but we can predicate any failures, and we can't determine anything.

Name: Amr Mohamed Mahdi Alnas
Cohort: 17