ALX Project Web infrastructure design

Task 1. Definitions and Explanations

For every additional element, why are adding it

Adding a new server so that we can be able to add a load balancer to handle too much incoming traffic and enable us to eliminate a single point of failure which could occur by having just one server.

- **What distribution algorithm your load balancer is configured with and how it works**
  Our load balancer employs the Round Robin algorithm, which establishes connections in a sequential manner, unless a server is inaccessible. Incoming requests are distributed among servers in a rotational order, processing each request in sequence. Once all servers have received a request, the cycle begins anew from the first server. This algorithm is chosen when servers possess equal specifications and persistent connections are not extensively utilized.

- **Is your load-balancer enabling an Active-Active or Active-Passive setup, explain the difference between both**
  We utilize an Active-Active setup for our load balancers. In this configuration, both load balancers are always actively operational. This approach ensures availability and redundancy. If one load balancer experiences an issue, the other can seamlessly take over the workload, ensuring uninterrupted service.

  On the other hand, an Active-Passive setup involves having one server active while the other remains in a passive state. The passive server comes into action only when the active server fails. It acts as a standby, activated as needed to maintain service continuity.
- **How a database Primary-Replica (Master-Slave) cluster works**
  The primary (master) database server handles writing operations and manages data modifications.
  Replicas (slave) database servers are copies of the primary and are used mainly for read operations.
  Replicas receive updates from the primary transaction log and keep their data synchronized.
- **What is the difference between the Primary node and the Replica node in regard to the application**
  The **Primary node** handles write operations and is the authoritative source for data changes. Applications interact with the Primary for data modification queries (e.g., INSERT, UPDATE, DELETE). It ensures data integrity and manages transactions.

  The **Replica node** is primarily used for read operations. Applications can query the Replica for data retrieval (e.g., SELECT queries). Replicas receive updates from the Primary's transaction log and provide improved read scalability and redundancy.

what the issues

- **Where are SPOF**

  DNS Server Failure

  Web Server (Nginx) Failure

  Application Server (App Server) Failure

  Database Server (MySQL) Failure

- **Security issues (no firewall, no HTTPS)**
  No Firewall: Allows potential denial of service attacks, breaches through unsecured ports, and data exfiltration.
  No HTTPS: Exposes sensitive data due to unencrypted communication, putting passwords and information at risk.
- **No monitoring**
  Inability to identify and respond to system issues promptly.
  Missed opportunities for performance optimization and issue prevention.
  Reduced visibility into system health and anomalies.