ALX Project
Web infrastructure design

Task 1.
Definitions and Explanations.

1. **For every additional element, why we are adding it:** We are adding a new server to incorporate a load balancer, aiming to address two key objectives: managing high incoming traffic effectively and mitigating the risk of a single point of failure associated with relying on a single server.
2. **What distribution algorithm your load balancer is configured with and how it works:** Our load balancer employs the Round Robin algorithm, which follows a sequential order when connecting to servers, provided they are not offline. Requests are distributed to servers in a sequential manner, progressing from one server to the next. Once the last server is reached, the sequence restarts with the first server.
3. **Is your load-balancer enabling an Active-Active or Active-Passive setup, explain the difference between both:** The load balancer facilitates an Active-Active setup, where both nodes (servers) actively run the same service concurrently. In contrast, an Active-Passive setup involves not all nodes being simultaneously active. For a two-node configuration, if the first node is active, the second node remains passive or on standby. The primary distinction lies in performance: Active-Active clusters utilize the resources of all servers in regular operations, while an Active-Passive cluster only engages the backup server during failover events.
4. **How a database Primary-Replica (Master-Slave) cluster works:** Master-Slave replication involves the propagation of data from one designated database server (the master) to one or more additional database servers (the slaves). The master server records updates, and these changes are subsequently replicated to the slave servers. Synchronization can occur either synchronously, if changes are made simultaneously to the master and slaves, or asynchronously, if changes are queued and written at a later time.
5. **What is the difference between the Primary node and the Replica node in regard to the application:** A replica node functions as a duplicate of the primary node, offering redundant copies of the application codebase. This duplication serves two primary purposes: first, it acts as a safeguard against hardware failures, ensuring continued availability of the application in case the primary node encounters issues. Second, replica nodes increase capacity to handle read requests, such as searching or retrieving documents, thereby enhancing the system's overall performance and responsiveness.