0x09. Web infrastructure design Project

Tasks

1-distributed_web_infrastructure.docx

- What distribution algorithm your load balancer is configured with and how it works
  - The HAProxy load-balancer is enabling an *Active-Passive* setup rather than an *Active-Active* setup. In an *Active-Active* setup, the load balancer distributes workloads across all nodes in order to prevent any single node from getting overloaded. Because there are more nodes available to serve, there will also be a marked improvement in throughput and response times. On the other hand, in an *Active-Passive* setup, not all nodes are going to be active (capable of receiving workloads at all times). In the case of two nodes, for example, if the first node is already active, the second node must be passive or on standby. The second or the next passive node can become an active node if the preceding node is inactive.
- Is your load-balancer enabling an Active-Active or Active-Passive setup, explain the difference between both
  - The load balancer enables active setup where both servers are actively running the same kind of services simeltaniously
- How a database Primary-Replica (Master-Slave) cluster works
  - A *Primary-Replica* setup configures one server to act as the *Primary* server and the other server to act as a *Replica* of the *Primary* server. However, the *Primary* server is capable of performing read/write requests whilst the *Replica* server is only capable of performing read requests. Data is synchronized between the *Primary* and *Replica* servers whenever the *Primary* server executes a write operation.
- What is the difference between the Primary node and the Replica node in regard to the application
  - The *Primary* node is responsible for all the write operations the site needs whilst the *Replica* node is capable of processing read operations, which decreases the read traffic to the *Primary* node.

Issues

- Cannot scale if there's too much incoming traffic.
  It would be hard to scale this infrastructure becauses one server contains the required components. The server can quickly run out of resources or slow down when it starts receiving a lot of requests

- There are multiple SPOF (Single Point Of Failure) in this infrastructure.
  For example, if the MySQL database server is down, the entire site would be down.
- Downtime when maintenance needed.
  When we need to run some maintenance checks on any component, they have to be put down or the server has to be turned off. Since there's only one server, the website would be experiencing a downtime.