Task 1:

Web Server (Nginx):

Purpose: Serves static content, handles incoming requests.

Why Use It: Efficiently serves static files, offloading processing from the application server.

Application Server:

Purpose: Executes application logic, processes dynamic content.

Why Use It: Separates application logic from the web server, allowing for scalability and modularity.

Load Balancer (HAproxy):

Purpose: Distributes incoming traffic across multiple application servers.

Why Use It: Improves performance, ensures high availability, and enables scalability.

Distribution Algorithm: Round Robin- Routes traffic sequentially to each server in a circular order.

Active-Active or Active-Passive: Active-Active- All servers actively handle requests, promoting load distribution and high availability.

Database Server (MySQL):

Purpose: Stores and manages application data.

Why Use it: Centralized data storage, efficient data management.

Primary-Replica Cluster: Implements a Master-Slave replication model.

How it Works: The primary node (master) receives write operations and replicates them to replica nodes (slaves). This ensures data consistency and availability.

Difference Between Primary and Replica: The primary node accepts write operations, while replica nodes handle read operations. This distribution optimizes data access for the application.

Issues with this Infrastructure:

Single Point of Failure (SPOF):

The Load Balancer is a potential SPOF. If it fails, incoming traffic won't be properly distributed, affecting availability.

Security Issues:

No firewall is mentioned, exposing servers to potential security threats.

Lack of HTTPS can compromise data integrity and user security.

No Monitoring:

Without monitoring tools, it's challenging to identify and address performance issues, potential failures, or security breaches in real-time.