Design of a Three-Server Web Infrastructure for www.foobar.com:

\*\*Components\*\*:

\*\*Server 1 (Primary Server)\*\*:

- Hosting primary components of the web infrastructure.

\*\*Server 2 (Replica Server)\*\*:

- Acting as a replica or backup for redundancy and data consistency.

\*\*Web Server (Nginx)\*\*:

- Handling incoming HTTP requests and serving static content.

\*\*Application Server\*\*:

- Hosting the website's dynamic codebase, processing application logic, and serving dynamic content.

\*\*Load Balancer (HAproxy)\*\*:

- Distributing incoming traffic between the two servers for load balancing and high availability.

\*\*Application Files (Code Base)\*\*:

- Storing the website's code, including server-side scripts, templates, and logic.

\*\*Database (MySQL)\*\*:

- Storing structured data and handling database operations.

\*\*Specifics\*\*:

\*\*Additional Elements\*\*:

- \*\*Replica Server\*\*: Added for redundancy. In case the primary server fails, the replica server can take over to ensure high availability.

- \*\*Load Balancer (HAproxy)\*\*: Added to evenly distribute incoming user traffic between the primary and replica servers. This load balancing ensures efficient resource utilization and high availability.

- \*\*Database Primary-Replica Cluster\*\*: To enhance database reliability and performance, the database is configured as a Primary-Replica (Master-Slave) cluster. This allows for data replication and failover capabilities.

- \*\*Code Base\*\*: Application files are included to host the website's code and serve dynamic content.

\*\*Load Balancer Distribution Algorithm\*\*:

- The load balancer (HAproxy) can be configured with various distribution algorithms, such as round-robin, least connections, or IP hash, depending on the requirements.

- Round-robin, for example, evenly distributes incoming requests to each server in a circular fashion. This ensures that no server is overwhelmed with traffic.

\*\*Active-Active vs. Active-Passive\*\*:

- In this design, the load balancer enables an Active-Active setup, which means that both the primary and replica servers actively handle user traffic simultaneously.

- Active-Active provides load balancing and high availability, with both servers serving requests and sharing the traffic load.

- In contrast, an Active-Passive setup would involve the replica server remaining idle unless the primary server fails, making it a passive backup.

\*\*Database Primary-Replica (Master-Slave) Cluster\*\*:

- In a Primary-Replica cluster, the Primary node (Master) handles write operations (e.g., INSERT, UPDATE), while the Replica node (Slave) replicates data from the Primary node and serves read operations.

- The Primary node is responsible for maintaining data integrity and consistency. Changes made here are propagated to the Replica node.

- The Replica node is used to scale read operations and provides redundancy. It ensures data availability in case the Primary node experiences issues.

\*\*Difference Between Primary and Replica Nodes\*\*:

- In regard to the application, the Primary node handles write operations and database updates. Any data changes, such as user registrations or content updates, are performed on the Primary node.

- The Replica node primarily serves read operations and query requests. It provides faster response times for retrieving data and ensures data availability even if the Primary node fails.

\*\*Issues\*\*:

1. \*\*Single Points of Failure (SPOF)\*\*:

- The web servers, application servers, and database all have SPOF. If any of these components fail, it can lead to downtime.

2. \*\*Security Issues\*\*:

- There's no mention of a firewall, which is essential for protecting the infrastructure from unauthorized access and attacks.

- HTTPS is not mentioned, leaving the website vulnerable to data interception and compromising user privacy.

3. \*\*No Monitoring\*\*:

- Lack of monitoring tools means there is no proactive way to detect and address issues before they impact users. Monitoring is crucial for maintaining system health and performance.

To address these issues, you should consider adding redundancy and failover mechanisms for web servers and databases, implementing a firewall and HTTPS for security, and setting up monitoring and alerting tools to ensure system health and early issue detection.