Design of a Multi-Server Web Infrastructure with Split Components and Load Balancing:

\*\*Components\*\*:

1. \*\*Server 1 (Web Server)\*\*:

- Hosts the website's static content, such as HTML files, images, and CSS.

2. \*\*Server 2 (Application Server)\*\*:

- Hosts the website's dynamic codebase, executes application logic, and serves dynamic content.

3. \*\*Server 3 (Database Server)\*\*:

- Stores structured data, handles database operations, and provides data to the application server.

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

- Configured as a cluster with two nodes for load balancing and high availability.

\*\*Specifics\*\*:

\*\*Additional Elements\*\*:

- \*\*Server 1 (Web Server)\*\*: Splitting the web server from the application server enhances security and performance. Serving static content separately ensures faster response times for users and reduces the risk of application server overload.

- \*\*Server 2 (Application Server)\*\*: Separating the application server from the web server allows for better resource allocation, scalability, and easier maintenance.

- \*\*Server 3 (Database Server)\*\*: A dedicated database server provides optimal data management, isolation from other services, and ensures data integrity.

- \*\*Load Balancer (HAproxy)\*\*: Configuring HAproxy as a cluster with two nodes improves reliability. It balances user requests between the web and application servers, ensuring high availability and preventing overloading of a single server.

\*\*Why Split Components\*\*:

- Splitting components into separate servers enhances security, performance, and scalability. Each server can be optimized for its specific role, making the infrastructure more resilient to high traffic and easier to maintain.

\*\*Why Load Balancing\*\*:

- Load balancing distributes incoming user requests across multiple servers to ensure even resource utilization, minimize downtime, and enhance performance.

- Clustering the load balancer adds redundancy, further increasing availability by providing failover in case one node fails.

This design optimizes the infrastructure by isolating components onto dedicated servers, improving performance and security. The load balancer cluster ensures reliability and high availability.