On the whiteboard, we'll outline the following components:

1. Load Balancer: The load balancer acts as the entry point for incoming traffic, distributing requests across multiple servers. We'll terminate SSL at the load balancer level for enhanced security and performance.
2. Web Servers: We'll have three web servers that handle incoming requests from the load balancer. These servers will serve the web pages and communicate with the application servers.
3. Application Servers: The application servers will execute the website's codebase, handle the application logic, and generate dynamic content. They will communicate with the web servers for request processing and retrieve data from the database.
4. Database Servers: We'll include multiple MySQL database servers configured in a cluster, utilizing a master-master replication setup. This ensures redundancy and allows multiple servers to handle write operations.
5. Firewalls: We'll add three firewalls to enhance the security of the infrastructure. Firewalls act as barriers, filtering network traffic based on predefined rules and policies. They help protect the servers and the network from unauthorized access and potential threats.
6. SSL Certificate: We'll install an SSL certificate on the load balancer to serve the website over HTTPS. HTTPS encrypts the traffic between the user's browser and the web server, ensuring data confidentiality and integrity.
7. Monitoring Clients: We'll include three monitoring clients, which act as data collectors for a monitoring service like Sumo Logic. These clients gather metrics, logs, and other data from the servers and send it to the monitoring service for analysis and visualization.

Now, let's address the specifics of this infrastructure:

- Firewalls: Firewalls are added to provide an additional layer of security. They control incoming and outgoing network traffic based on predefined rules, protecting the servers from unauthorized access and potential threats.
- Traffic served over HTTPS: HTTPS encrypts the communication between the user's browser and the web server, ensuring that sensitive data transmitted over the network remains secure and cannot be intercepted or tampered with by malicious actors.
- Monitoring: Monitoring is used to track the health, performance, and availability of the infrastructure and applications. It helps identify issues, proactively detect anomalies, and gather insights for capacity planning and optimization.
- Monitoring Data Collection: The monitoring clients installed on the servers collect various data, including metrics, logs, and events. They send this data to the monitoring service, such as Sumo Logic, where it is processed, analyzed, and presented in a meaningful way to administrators and operators.

- Monitoring Web Server QPS: To monitor the web server's queries per second (QPS), you can leverage the monitoring tool's metrics collection capabilities. It should provide QPS-related metrics, which you can track and analyze to understand the server's performance and identify any spikes or abnormalities.

Now, let's discuss the issues with this infrastructure:

1. Terminating SSL at the load balancer level: While terminating SSL at the load balancer improves performance and offloads the encryption workload from the web servers, it means that the communication between the load balancer and web servers may be unencrypted. To ensure end-to-end security, it's generally recommended to encrypt traffic between the load balancer and web servers as well.
2. Single MySQL server accepting writes: Having only one MySQL server capable of accepting writes represents a single point of failure. If this server fails, write operations will be unavailable, impacting the availability and functionality of the website. To address this, implementing a multi-master replication setup with multiple write-capable MySQL servers would provide redundancy and improve fault tolerance.
3. Identical components across servers: Having servers with the same components (database, web server, and application server) might lead to a lack of fault isolation. If there is a critical issue with one component, it could affect all servers simultaneously. Introducing diversity in the infrastructure by using different hardware or software versions for certain components can mitigate this risk.