Task 2

Secured and monitored web infrastructure

Recently added to the infrastructure:

- 3 firewalls.
- 1 SSL certificate to serve **www.foobar.com** over HTTPS.
- 3 monitoring clients (data collector for Sumologic or other monitoring services).

Reasons to add it:

● 3 Firewalls

Firewall is a network security device that monitors and filters incoming and outgoing network traffic, based on a previously established security policies. And it usually connected between the internal network and the external network such as the Internet.

In our infrastructure, 3 firewalls were added, the 1st firewall was added before the first server which is Load-Balancer server, the other 2 firewalls were added before each internal server to add more security layer to our network infrastructure.

● 1 SSL certificate

SSL certificate stands for Secure Sockets Layer, is a digital certificate that authenticates a website's identity and enables an encrypted link between a web server and a web browser.

You can know that's the website we're visiting is secured using SSL by the presence of padlock icon next to the URL in the address bar in the browser.

In our infrastructure, one SSL added to keeps internet connection secure and prevent criminals and hackers from reading or modifying information transferred between user's browser and our web server by encrypting the HTTP requests using the HTTPs, secured HTTP protocol.

● 3 Monitoring Clients

Monitoring Clients is a real-time software monitoring tool that watch computer metrics, record them, and emit an alert if something is unusual or that could make the computer not work properly happens.

Web stack monitoring can be broken down into 2 categories:

- ➢ **Application Monitoring**: getting data about the running software and making sure it is behaving as expected.
- ➢ **Server Monitoring**: getting data about the virtual or physical server and making sure they are not overloaded (could be CPU, memory, disk or network overload).

Monitoring tools collect data through various methods, including:

- **Agent-based Monitoring:** Installing lightweight agents on servers to collect and send data.
- **SNMP (Simple Network Management Protocol):** Monitoring network devices using SNMP-enabled agents.
- **Log Parsing:** Analyzing log files generated by servers and applications.
- **API Integration:** Connecting directly to APIs provided by applications or services to gather metrics.

Monitoring Web Server QPS (Queries Per Second):

- **QPS** stands for Queries Per Second, measuring the rate of queries or requests processed by a server.
- **Steps to Monitor Web Server QPS:**
  1. **Choose a Monitoring Tool:** Select a monitoring tool that supports web server metrics.
  2. **Configure Monitoring Agent: I**nstall and configure the monitoring agent on the web server.
  3. **QPS Metrics:** Specify the QPS metrics to be monitored (e.g., HTTP requests/sec).
  4. **Set Maximum Limits:** Establish acceptable QPS maximum limits for alerting.
  5. **Dashboard Setup:** Create a dashboard to visualize QPS trends and fluctuations.
  6. **Alerting Configuration:** Configure alerts to notify administrators if QPS exceeds defined **maximum limits**.
  7. **Regular Review:** Regularly review monitoring data to identify patterns or rare and unexpected errors.
  8. **Optimize Performance:** Use monitoring insights to optimize server performance and handle increased QPS effectively.

issues are with this infrastructure:

1. **Terminating SSL at the Load Balancer:**
   - **Issue:** When SSL termination occurs at the load balancer, the communication between the load balancer and backend servers is in plaintext. This means that the data is no longer encrypted beyond the load balancer.
   - **Security Concerns:** Sensitive information, such as user credentials, may be vulnerable to interception if the communication between the load balancer and backend servers is not secured. This undermines the end-to-end encryption provided by SSL/TLS, potentially exposing data to malicious actors.

2. **Single MySQL Server Accepting Writes:**
   - **Issue:** Having only one MySQL server capable of accepting writes poses a SPOF single point of failure.
   - **High Availability Concerns:** If the sole writable MySQL server experiences downtime or fails, the entire system may become unavailable for write operations. This configuration lacks redundancy, exposing data integrity and system reliability in danger.
   - **Scalability Challenges:** It limits the ability to distribute write operations across multiple servers, hindering scalability as the application grows in terms of user base or workload.

3. **Uniformity of Server Components (Database, Web Server, Application Server):**
   - **Issue:** Deploying servers with identical components can lead to several problems.
   - **Limited Flexibility:** Different components may have varying resource requirements. Uniform servers may lead to over-provisioning certain resources for some components and under-provisioning for others, impacting overall system efficiency.
   - **Scalability Challenges:** As the application scales, different components might need to scale independently. Uniformity may hinder the ability to allocate resources optimally based on the specific needs of each server type.
   - **Difficulty in Specialization:** Certain components, such as databases or caching servers, may benefit from specialized configurations. Uniformity might limit the ability to tailor each server type for optimal performance in its specific role.