

WEB APPLICATION ARCHITECTURE

Building the architecture for an e-commerce website begins with understanding the end-users—customers who will interact with the platform. Users will send requests, such as retrieving product details or placing orders, which the application processes to deliver the required information. Developing such an application involves carefully planning what features to include, where they should be implemented, and how resources can be allocated efficiently. By focusing on user needs and aligning the application with robust design principles, we can create a system that is reliable, scalable, and user-friendly.

To understand the components:

Load balancer: contributes to handling traffic from users to multiple web servers. It ensures optimal performance by balancing the load between servers (as the name suggests) so that no single server gets overwhelmed with excess traffic (thereby avoids crashing). Amazon uses an Elastic Load Balancer from AWS to deal with traffic during peak sales, which has the ability to auto scale and dynamically add or remove instances based on traffic and distributes traffic amongst those multiple instances.

Auto-scaling VMs/Containers: handles user requests. Scaled up or down depending on the traffic that the application receives. (To understand through an example: consider a supermarket, if there are many customers, many billing counters will be open, if there are few they cut down the number of open billing counters.) Nginx, often used as a web server or reverse proxy, plays a key role in efficiently handling HTTP requests, serving static files, and forwarding client requests to backend servers. It can also act as a software load balancer and optimize performance through caching and SSL support, making it ideal for managing high web traffic in e-commerce platforms.

Cache: It deals with storing frequently accessed data on a temporary basis making it faster to retrieve without needing to access the database. Amazon caches product name, price and image so that repeated requests for the same product are served faster and also reduces the load on the database.

Database: Structured data storage that holds essential information like user accounts, product details, inventory and much more. Amazon uses multiple databases like mysql, postgresql, mariaDB, oracle, etc.

Storage: Large files not typically stored in the database, such as product reviews and database backups. For example, product reviews, backups of database etc. Amazon uses S3 bucket as their storage choice.

Security: This includes using firewalls and intrusion detection systems (IDS) to prevent unauthorized access, SSL/TLS encryption for secure communication, and encrypting sensitive data both in transit and at rest. Access controls are implemented through role-based policies and secure authentication methods, while regular security audits and vulnerability scans help maintain system integrity

Monitoring: To track the performance of the application, it continuously monitors web servers, error rates, response times etc to enable easy and efficient functioning of the application. Sends alerts and reports to the development team so any issues can be fixed quickly thereby minimizing loss to both the host and the customer. Example: Prometheus is one such monitoring tool

Together, these components create a user-friendly platform capable of handling peak loads, providing seamless performance, and delivering a superior customer experience.