**What is a Web server?**

* A webserver is a computer system / software that delivers web content to the users over the internet.
* It processes requests sent by user’s web browser and serves up web pages, files, or other resources that makeup a website.
* **Working:**
  + **Client Request:** a user enters a website URL in their web server. This sends a request to the web server hosting that site.
  + **Process Request:** the web server receives the request and processes it. If the resource exists, the server receives it.
  + **Response:** the server sends the request content back to the user’s browser, which then displays the website
* **Key Features:**
  + **Content Delivery:** serves static content like HTML, CSS, JavaScript, Images and dynamic content generated by application servers / databases.
  + **Protocol Support:** work with HTTP / HTTPS protocols for secure and standard communication between the server and client.
  + **Request handling:** efficiently handle multiple client requests concurrently using techniques like multithreading / asynchronous handling.
  + **Virtual hosting:** allow hosting multiple websites on a single server, identified by different domain names.
  + **Security features:** supports SSL/TSL encryption, authentication mechanisms, and firewalls to safeguard data and resources.
  + **Logging and Monitoring:** record details of user requests and server performance for analytics and troubleshooting
  + **Customizability:** configurable settings to control website behaviour.
  + **Load Balancing:** distribute incoming requests across multiple servers to ensure high availability and performance during traffic spikes.
  + **Integration with applications:** work seamlessly with backend services and application services to handle business logic and database operations.
  + **Scalability:** easily scale to handle growing traffic or deploy in clustered setups for improved reliability.

**APACHE TOMCAT:**

* Apache Tomcat is an open-source web server and servlet container developed by the Apache Software Foundation.
* It is widely used for deploying java-based web applications and is specifically designed to execute Java Servlets and JavaServer Pages, which are components of server-side Java Programming.
* **Key Features:**
  + **Servlet and JSP Support:** provides a robust environment for running Java Servlet and JSPs, which are used to build dynamic web applications.
  + **WebSocket Support:** enables bi-directional communication between client and server, useful for real-time applications like chat systems and live notifications.
  + **Lightweight and Fast:** ideal for small-to-medium scale Java web applications due to it’s lightweight architecture and efficient resource management.
  + **Integration with Java EE:** easily integrates with other Java EE technologies, such as JavaBeans, JMS, JNDI.
  + **Highly Configurable:** offers extensive configuration options through XML files for fine-grained control over server behavior.
  + **Clustering and Load Balancing:** supports clustering for fault tolerance and load balancing , ensuring high availability and scalability for enterprise applications.
  + **Security Features:** comes with built-in security capabilities, including SSL / TSL encryption, realms for user authentication and access control.
  + **Modular Architecture:** allows developers to use only the necessary components, reducing overhead and complexity.

**Nginx:**

* Nginx is a powerful, high performance web server, reverse proxy server and load balancer.
* It is widely used for handling large volumes of traffic efficiently and is especially popular for its speed and low resource consumption.
* **Key features:**
  + **High Performance:** handlesa large number of simultaneous connections with low memory usage, makes it ideal for high traffic websites.
  + **Reverse Proxy:** acts as an intermediary between clients and servers, enabling load balancing, caching, and SSL termination.
  + **Load Balancing:** distributes the traffic across multiple servers to improve scalability and reliability.
  + **HTTPS Support:** provides more secure communication by supporting more web protocols like HTTPS.
  + **Static Content Serving:** Efficiently serves Static content like HTML, CSS, which reduces the load on the application servers.
  + **Dynamic Configuration:** allows seamless reloading of the configuration files without disrupting the active interaction, making updates easy.
  + **Event Driven Architecture:** uses asynchronous, non-blocking event handling to process multiple requests concurrently, enhancing it’s speed and efficiency.

**Difference Between Tomcat and Nginx:**

| **Tomcat** | Nginx |
| --- | --- |
| Primarily used for Java servlets and JSP | High performance web server, reverse proxy and load balancer |
| Runs Java based applications | Serves static content, reverse proxying, load balancing and caching |
| Slower than Nginx for static content, optimized for Java applications | Faster for Static Content, handles high concurrency efficiently |
| Threadbasedarchitecture | Event driven architecture |
| Limited Capabilities | Excellent reverse proxy functioning |
| Basic load balance functionality | Advanced Load balance |
| More memory consumption | Less memory consumption |