06-06-2023

**DIFFERENT WEB SERVERS AVAILABLE IN THE MARKET SPACE**

Every Website sits on a computer known as a Web server. This server is always connected to the internet. Every Web server that is connected to the Internet is given a unique address made up of a series of four numbers between 0 and 255 separated by periods. For example, 68.178.157.132 or 68.122.35.127.

1. **APACHE HTTP SERVER**

This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is an open-source software and can be installed on almost all operating systems including Linux, Unix, Windows, FreeBSD, Mac OS X and more. About 60% of the web server machines run the Apache Web Server.

**Working of Apache:**

* Apache is not any physical server; it is software that executes on the server. However, we define it as a web server.
* Its objective is to build a connection among the website visitor browsers (Safari, Google Chrome, Firefox, etc.) and the server. Apache can be defined as cross-platform software, so it can work on Windows servers and UNIX.
* When any visitor wishes for loading a page on our website, the homepage, for instance, or our "About Us" page, the visitor's browser will send a request on our server.
* Apache will return a response along with each requested file (images, files, etc.). The client and server communicate by HTTP protocol, and Apache is liable for secure and smooth communication among t both the machines.
* Apache is software that is highly customizable. It contains the module-based structure. Various modules permit server administrators for turning additional functionality off and on.
* Apache includes modules for caching, security, password authentication, URL rewriting, and other purposes. Also, we can set up our own configuration of the server with the help of a file known as .htaccess. It is a supported configuration file of Apache.

**Features of Apache:**

* **HTTP/1.1 and HTTP/2 support:** Apache supports the latest HTTP protocols, enabling efficient and secure communication between clients and servers.
* **Virtual hosting**: Apache allows hosting multiple websites on a single server by using virtual hosts. Each virtual host can have its own configuration and website content.
* **URL rewriting:** Apache provides powerful URL rewriting capabilities, allowing you to manipulate URLs for search engine optimization (SEO) or to create user-friendly URLs.
* **Authentication and access control**: Apache supports various authentication mechanisms, such as basic, digest, and form-based authentication. It also offers access control based on IP addresses, user groups, or custom rules.
* **Dynamic content:** Apache integrates well with scripting languages like PHP, Perl, and Python, allowing you to generate dynamic web pages.
* **SSL/TLS encryption:** Apache supports secure connections using SSL/TLS protocols, enabling HTTPS communication for secure transactions.
* **Modules and extensions:** Apache have a modular architecture, allowing you to extend its functionality through modules. There is a wide range of modules available for tasks like caching, proxying, compression, and more.

**Apache advantages and disadvantages:**

Pros:

* It is open-source, making it completely free to use and customizable.
* The software is generally reliable, and has frequent patches to improve its performance and reduce security vulnerabilities.
* It is beginner-friendly and relatively easy to learn.
* The software works across multiple platforms.
* Apache works well with programming languages such as PHP and Python.
* The software is compatible with WordPress websites.
* There is a large community of developers and support if you run into any issues.

Cons:

* It can struggle with high traffic numbers.
* Modifying the default configurations can leave you vulnerable to security threats.
* You’ll need to update the software frequently

1. **NGINX WEB SERVER:**

Nginx is a lightweight and high-performance web server. It's known for its scalability and ability to handle concurrent connections efficiently. Nginx is often used as a reverse proxy, load balancer, or caching server, and it's commonly used to serve static content.

NGINX provides various services such as reverse proxy, load balancer, and rate limit network services. Reverse proxying is useful if we have multiple web services listening on various ports and we need a single public endpoint to reroute requests internally. This would allow us to host multiple domain names on port 80 while using a combination of different NodeJS, Go and java to power separate web services behind the scenes. Nginx can handle the logging, blacklisting, load balancing and serving static files while the web services focus on what they need to do.

**Working of Nginix:**

1. Receive and Process Requests: Nginx runs as a daemon process, listening on specified ports (usually port 80 for HTTP and port 443 for HTTPS). When a client sends an HTTP request to the server, Nginx receives it and starts processing.
2. Configuration Parsing: Nginx reads and parses its configuration file, typically located at "/etc/nginx/nginx.conf". The configuration file defines various server settings, including listening ports, server blocks, and proxy configurations.
3. Virtual Host Matching: Nginx matches the incoming request to the appropriate virtual host based on the requested domain name or IP address. Each virtual host has its own configuration, allowing Nginx to serve different websites on the same server.
4. Request Processing: Once the appropriate virtual host is determined, Nginx processes the request based on the configuration for that particular virtual host. This includes applying any URL rewriting rules, handling static content, or forwarding the request to backend servers.
5. Reverse Proxy and Load Balancing: If Nginx is configured as a reverse proxy, it can forward the request to one or more backend servers based on load-balancing algorithms. Nginx can distribute incoming requests across multiple backend servers to optimize resource utilization and improve performance.
6. Static File Serving: If the requested resource is a static file (e.g., HTML, CSS, JavaScript, images), Nginx can serve it directly from the file system without involving backend servers. Nginx's high-performance file-serving capabilities make it efficient in delivering static content.
7. Caching: Nginx can cache responses from backend servers to improve performance and reduce the load on the backend infrastructure. It can cache both static and dynamic content based on cache configuration settings, such as cache control headers, cache expiration rules, and cache sizes.
8. SSL/TLS Termination: When serving HTTPS requests, Nginx can handle SSL/TLS encryption and decryption. It can terminate the SSL/TLS connection, decrypt the incoming request, and forward it to the backend servers over an unencrypted channel. Nginx can also perform SSL certificate management and configuration.
9. Response Generation and Delivery: Nginx generates the response based on the processing steps performed and any content generated by backend servers. It sends the response back to the client over the established connection.
10. Logging and Error Handling: Throughout the request handling process, Nginx logs relevant information, such as access logs and error logs. This helps in troubleshooting issues, monitoring server activity, and analyzing traffic patterns.

**Features of Nginix:**

* **High Performance:** Nginx is designed to handle a large number of concurrent connections and high traffic loads efficiently. It uses an asynchronous, event-driven architecture that allows it to handle requests with low memory footprint and minimal resource usage. This makes it well-suited for serving static content and handling high-volume websites.
* **Reverse Proxy:** Nginx acts as a reverse proxy, which means it can receive client requests and forward them to backend servers. It provides features like load balancing, SSL/TLS termination, caching, and request routing. Reverse proxying helps distribute traffic among multiple servers and improves performance, security, and availability.
* **Load Balancing:** Nginx supports load balancing across multiple backend servers to distribute incoming requests and optimize resource utilization. It offers various load-balancing algorithms, including round-robin, least connections, IP hash, and more. Load balancing helps improve performance, handle traffic spikes, and provide fault tolerance.
* **HTTP and HTTPS Support:** Nginx supports both HTTP and HTTPS protocols, allowing you to serve both regular and secure connections. It can handle SSL/TLS encryption and provides options for certificate management and configuration of security protocols and ciphers. Nginx also offers automatic SSL certificate installation and renewal using Let's Encrypt.
* **Caching:** Nginx includes an HTTP cache module that allows you to cache static and dynamic content, reducing the load on backend servers and improving response times. It supports flexible cache configuration, including cache control headers, cache purging, and cache expiration rules.
* **URL Rewriting:** Nginx provides powerful URL rewriting capabilities, allowing you to manipulate URLs for SEO purposes or create user-friendly URLs. It supports regular expressions and provides flexible rewriting rules for modifying URLs as they pass through the server.
* **Virtual Hosting:** Nginx allows hosting multiple websites on a single server using virtual hosts. Each virtual host can have its own configuration, domains, and website content. Nginx supports name-based and IP-based virtual hosting, enabling you to serve different websites based on the requested hostname or IP address.
* **WebSocket and HTTP/2 Support:** Nginx has native support for WebSocket communication, allowing you to handle real-time, bidirectional communication between clients and servers. It also supports HTTP/2, the latest version of the HTTP protocol, which offers improved performance, multiplexing, and server push capabilities.
* **Easy Configuration:** Nginx uses a simple and intuitive configuration syntax that allows for easy setup and management. The configuration files are human-readable and organized into blocks, making it convenient to define server behavior, locations, and other settings.
* **Modules and Extensibility:** Nginx has a modular architecture that allows extending its functionality through third-party modules. There is a wide range of community-developed modules available, offering additional features like advanced logging, authentication mechanisms, content compression, and more.

**NGINX advantages and disadvantages:**

Pros:

* High in performance- Nginix has been marked to record high performance than Apache
* Handle more clients with a smaller number of processes- Nginix require very less memory as far more clients can be supported with minimum or no thread. Hence very little memory is utilized as compared to Apache which needs one thread/process or each client.

Cons:

* Difficulties in module creation
* Difficulties in supporting HTTP/1.0 with backend communication

1. **MICROSOFT INTERNET INFORMATION SERVICES (IIS):**

IIS works through a variety of standard languages and protocols. HTML is used to create elements such as text, buttons, image placements, direct interactions/behaviors and hyperlinks. The Hypertext Transfer Protocol (HTTP) is the basic communication protocol used to exchange information between web servers and users. HTTPS -- HTTP over Secure Sockets Layer (SSL) -- uses Transport Layer Security or SSL to encrypt the communication for added data security. The File Transfer Protocol (FTP), or its secure variant, FTPS, can transfer files. Additional supported protocols include the Simple Mail Transfer Protocol (SMTP), to send and receive email, and the Network News Transfer Protocol, to deliver articles on USENET.

**Working of IIS:**

* IIS is a web server provided by Microsoft for Windows servers. It's designed to integrate well with other Microsoft technologies and provides features like ASP.NET hosting,

Windows authentication, and easy management through the Windows Server interface.

* An IIS web server accepts requests from remote client computers and returns the appropriate response.
* This basic functionality allows web servers to share and deliver information across local area networks (LAN), such as corporate intranets, and wide area networks (WAN), such as the Internet.
* A web server can deliver information to users in several forms, such as static webpages coded in HTML; through file exchanges as downloads and uploads; and text documents, image files and more.

**Features of IIS:**

1. Integrated with Windows: IIS is tightly integrated with the Windows operating system, making it easy to install, configure, and manage. It takes advantage of Windows security, authentication, and management features, providing a familiar environment for Windows administrators.
2. Hosting Options: IIS supports hosting various types of web applications, including static websites, dynamic websites built with ASP.NET or PHP, and other server-side technologies. It provides the necessary infrastructure to handle HTTP and HTTPS requests and supports multiple protocols like HTTP, FTP, SMTP, and more.
3. ASP.NET Support: IIS offers excellent support for hosting ASP.NET applications, Microsoft's web development framework. It includes the necessary components and runtime environments to execute ASP.NET code and offers integration with other Microsoft technologies like SQL Server and Active Directory.
4. Management and Administration: IIS provides multiple ways to manage and administer web servers. It offers a graphical management console called Internet Information Services (IIS) Manager, which provides a user-friendly interface for configuring and monitoring server settings, websites, and application pools. Additionally, IIS can be managed using command-line tools and PowerShell cmdlets.
5. Scalability and Performance: IIS is designed to handle high traffic and scale as per the demands of the applications it hosts. It supports features like application pools, which isolate applications for better stability and performance. IIS can also take advantage of Windows Server features like clustering and load balancing to distribute traffic across multiple servers.
6. Security Features: IIS includes various security features to protect hosted websites and applications. It supports authentication mechanisms such as Windows authentication, Basic authentication, Digest authentication, and client certificate authentication. IIS also provides tools for managing SSL certificates, enabling secure HTTPS communication.
7. URL Rewrite and Redirect: IIS offers URL rewriting and redirection capabilities, allowing you to manipulate URLs for SEO purposes, create user-friendly URLs, or redirect requests to different locations. This feature helps in optimizing website URLs and managing redirects during website migrations or changes.
8. Logging and Monitoring: IIS provides comprehensive logging and monitoring capabilities to track server activity and diagnose issues. It generates detailed logs for access, errors, and other server events. Additionally, IIS can integrate with tools like Windows Performance Monitor and third-party monitoring solutions for performance analysis.
9. Extensibility: IIS supports extensibility through modules, allowing you to add additional functionality to the web server. Microsoft offers a wide range of modules for features like caching, compression, request filtering, and more. It also provides a development framework for building custom modules to meet specific requirements.
10. Application Pool Isolation: IIS utilizes application pools to isolate web applications from one another. Each application pool runs as a separate process, providing process isolation, improved security, and better stability. If one application encounters an issue, it does not affect other applications running in different pools.

**IIS advantages and disadvantages:**

Pros:

* The graphical user interface (GUI) helps new users. IIS has good integration with Performance Monitor, which provides for easy access to extensive usage statistics.

Cons:

* IIS is not robust and can easily be made to 'hang' so that the server must be rebooted to recover. It is certainly unacceptable for an NT Service to fail in this way.
* IIS can only be configured using the GUI, which can be a complex and laborious manual process in a production environment. There seems to be no way to encapsulate all of the settings in a single configuration file, as you can with Apache, which is essential for configuration management.
* IIS lacks flexibility in many areas. For example, there is almost no way to control how URLs are parsed.
* Remote administration and configuration of IIS is almost impossible, as this can only be done using a crippled web-interface that, of course, requires IIS to be running and correctly configured.
* IIS is only available for Windows.

1. **LITESPEED WEB SERVER:**

LiteSpeed is a high-performance, lightweight web server known for its excellent scalability and speed. It's designed to be a drop-in replacement for Apache and supports features like HTTP/2 and WebSocket. LiteSpeed is often chosen for its performance benefits and compatibility with existing Apache configurations.

**Features of LiteSpeed Web Server:**

LiteSpeed Web Server has superior performance in terms of both raw speed and scalability. LiteSpeed is faster than Apache while serving PHP contents. It’s an ideal hosting platform for serving WordPress, Joomla and Drupal based websites as it easily increases PHP’s performance and security.

* **Apache Compatibility:** LiteSpeed Web Server is compatible with the commonly used Apache web server and its features such as mod rewrite, .htaccess, and mod security. LiteSpeed Web Server can even load the Apache configuration files, allowing it to work as a direct replacement for Apache. It is also compatible with the leading hosting control panels, such as cPanel, Plesk, and DirectAdmin.
* **Performance & Scalability:** LiteSpeed Web Server increases the performance & scalability of web hosting platforms through its unique event driven architecture. It has the capability of serving thousands of clients simultaneously with minimum usage of server resources such as memory and CPU. The uniquely developed & optimized code of LiteSpeed Web Server increases PHP performance as well as serving static websites faster than Apache. It has the capability of handling sudden spikes in traffic as well as helping to manage against DDOS attacks without any DDOS mitigation hardware.
* **Security:** LiteSpeed Web Server is compatible with Apache’s mod security feature and has built-in anti-DDoS capability. With customizable features such as per-IP connections and bandwidth throttling, IPs that make too many connections or requests or ask for too much bandwidth are blocked, stopping attackers before they overrun your server.
* **Cost Effective:** Switching to LiteSpeed from Apache is an easy task and very cost effective in terms of support costs. LiteSpeed’s licensing costs are quite low compared to any hardware upgrades required to optimize a server running Apache. Additionally, LiteSpeed’s unique features help reduce support costs by providing an optimized, secure, and stable hosting platform. This in turn gives support technicians and administrators peace of mind instead of dealing with an overloaded server.

**Working of LiteSpeed Web Server:**

1. Architecture: LiteSpeed Web Server follows an event-driven architecture, leveraging asynchronous I/O and a highly optimized codebase to efficiently handle concurrent connections and maximize resource utilization.
2. Listener and Worker Processes: LSWS utilizes multiple listener processes to accept incoming connections and hand them over to worker processes. Listener processes are responsible for listening to network ports and accepting client requests, while worker processes handle the actual processing of these requests.
3. Request Processing: When a client sends an HTTP request to LSWS, a listener process accepts the connection and passes the request to an available worker process. The worker process then processes the request according to the server configuration and the requested resource.
4. Cache and Acceleration: LSWS includes a built-in caching system that can cache static and dynamic content, reducing the load on backend application servers. It also supports caching of database queries, which can greatly improve the performance of dynamic websites and web applications.
5. Event-driven Processing: LSWS utilizes an event-driven model where worker processes are event-driven and handle multiple connections concurrently. This model allows for efficient resource management and minimizes the overhead associated with thread or process creation.
6. LiteSpeed API: LSWS provides an API that enables developers to extend its functionality or integrate it with other systems. The API allows for customizing server behavior, implementing plugins, and creating application-specific optimizations.
7. HTTP/2 and TLS Support: LSWS supports the latest HTTP/2 protocol, enabling faster and more efficient communication between clients and the server. It also provides native support for SSL/TLS encryption, allowing for secure HTTPS connections and improved website security.
8. Compatibility with Apache: LiteSpeed Web Server is designed to be a drop-in replacement for Apache HTTP Server. It can seamlessly integrate with existing Apache configurations, including Apache mod\_rewrite rules, .htaccess files, and Apache-compatible modules. This makes the transition from Apache to LSWS relatively straightforward.
9. Management and Monitoring: LSWS provides a web-based administration panel called LiteSpeed WebAdmin Console. The console allows administrators to manage server settings, monitor server performance, configure virtual hosts, set up security features, and access server logs.
10. Performance and Scalability: The primary goal of LSWS is to deliver high performance and scalability. It aims to handle a large number of concurrent connections efficiently and optimize resource utilization. LSWS achieves this through its event-driven architecture, advanced caching mechanisms, and optimized codebase.

**LSWS advantages and disadvantages:**

Pros:

* The hardware required is not too big.
* Easy to learn
* Provides excellent customer server and community.
* Transactions are easier that Apache format.

Cons:

* Not open-source, requires specific licence to use.
* Limited community and ecosystem
* Dependency on LiteSpeed-Specific Features
* Compatibility with legacy systems
* Reduced customization.

1. **CADDY WEB SERVER:**

Caddy is a modern, open-source web server written in Go. It emphasizes ease of use and automatic HTTPS configuration, making it popular for developers and small-scale deployments. Caddy is known for its simplicity, automatic TLS certificates, and support for various plugins.

Caddy was first released in 2015 and developed by Carl Drechsler. The project is hosted on GitHub and has received funding from the Internet Infrastructure Foundation (IFF). The open-source, cross-platform web server runs on Linux, macOS, Windows, BSD, and Solaris. It’s also licensed under the Apache License 2.0. Caddy has several features that set it apart. Notably, it supports automatic HTTPS using Let’s Encrypt. This feature means that you can access all websites served by Caddy over a secure HTTPS connection. Caddy also comes with a web-based administration interface. This tool makes it easy to manage multiple websites on a single server. Caddy is highly extensible, and you can customize it to fit almost any need.

**Features of Caddy Web Server:**

1. Automatic HTTPS: One of the standout features of Caddy is its automatic HTTPS configuration. By default, Caddy automatically obtains and manages SSL/TLS certificates for your websites using Let's Encrypt. This makes it incredibly easy to enable secure HTTPS connections without any manual certificate configuration.
2. Easy Configuration: Caddy uses a simple and human-readable configuration language. The configuration file follows a straightforward structure, allowing you to define site-specific settings, routes, proxy configurations, and more. Caddy's configuration is designed to be easy to understand and modify, even for users who are not familiar with web server administration.
3. HTTP/2 Support: Caddy has native support for HTTP/2, the latest version of the HTTP protocol. HTTP/2 brings improvements such as multiplexing, server push, and header compression, resulting in faster and more efficient communication between clients and the server.
4. Reverse Proxy: Caddy can function as a powerful reverse proxy, allowing you to route requests to different backend servers or services based on various criteria. It supports load balancing, URL rewriting, path-based routing, and other advanced proxy features.
5. WebSockets Support: Caddy includes native support for WebSockets, enabling real-time, bidirectional communication between clients and servers. It allows you to proxy WebSocket connections, handle upgrades, and integrate WebSockets into your applications easily.
6. Extensible with Plugins: Caddy is designed to be extensible through plugins. It offers a plugin ecosystem that allows developers to add additional functionality to the server. There are a variety of official and community-developed plugins available, ranging from caching and compression to authentication and rate limiting.
7. Static File Serving: Caddy can serve static files efficiently. It supports automatic content compression, range requests, caching headers, and configurable caching strategies. This makes it suitable for hosting static websites or delivering static assets with optimal performance.
8. Automatic HTTP/2 Server Push: Caddy can automatically analyze web pages and initiate server pushes for critical resources, such as CSS and JavaScript files, to reduce page load times. This feature utilizes HTTP/2's server push capability to proactively deliver assets to clients.
9. On-Demand TLS: Caddy's on-demand TLS feature allows the server to generate and obtain certificates for domains only when needed, reducing the upfront effort and certificate management overhead. This feature is particularly useful for dynamic environments or when hosting sites with wildcard domains.
10. Logging and Metrics: Caddy provides logging capabilities to track requests and server activity. It can log in various formats, including JSON, Common Log Format (CLF), and more. Additionally, Caddy supports integration with external metrics systems like Prometheus, allowing you to monitor server performance and collect valuable insights.

**Working of Caddy:**

* A Caddy web server works by proxying requests from clients to backend servers. Caddy can be configured to serve websites directly from its file system, or it can proxy requests to other web servers.
* When a client makes a request, Caddy will first check its cache to see if the requested content is available.
* If so, Caddy will serve the content from its cache.
* If not, Caddy will forward the request to the appropriate backend server and return the response to the client.
* You can configure Caddy to serve static files, such as HTML, CSS, and JavaScript.
* Alternatively, you can use it as a reverse proxy for dynamic content generated by backend servers.
* Caddy is also capable of load-balancing requests between multiple backend servers.

**Caddy advantages and disadvantages:**

Pros:

* Ease of use
* Flexibility
* High performance

Cons:

* A lack of documentation.
* A limited feature set.