Server is anything that is ready to handle requests 24\*7 as it’s connected with the internet all the time. I will make it as easy as possible to understand the difference. So, we’ll start with an example.

Let’s say you have a website that gets 1000 visitors a day. And your budget to host your website is $10 per month. In that case, you will get one Shared hosting or tiny VPS to host your website.

Now, You are hosting your website on a single server that responds your Web requests (Apache/Nginx), Processes your data(PHP/Python) and handles your database queries(MySQL/MariaDB/Postgres)

All of these softwares are installed in a same machine and a single machine handles Web requests, Application processes and Database queries. It means, a single machine has **Web server, Application server and Database server**.

Now, you made progress and your website is getting 1,000,000 visitors per month. In this case, to maximize the uptime and to improve monitoring, you will host your database on different machine and call it a **Database server.**Because it handles all your database queries.

Now, your traffic is 1.5M visitors every month and you want to maximize your revenue by processing their data which will require too much processing. It will increase load on your server that handles your web requests and application processes.

So, you will host your application on another machine that handle requests from the web server. It will become your **Application server** and the server that is handling your web requests will become your dedicated **Web server**.

Now, you have one machine that handles your web requests which is called **Web server**. You have one server that runs all the processes your application requires in order to work properly which is called your **Application server**. And you have one server that handles database queries and we call it **Database server**.

Our user base increased to 50 million users. Now, we came to know that a single web server isn’t enough to handle those requests. What we’ll do next is create two more servers, One will act as a **Load balancer** and the second will be another web server.

As we have a different database server, different application server, all we have to do is to host our application frontend on the second web server just like it is on the first web server. And we can configure the load balancer to send a request to both of them one by one (Or there are other algorithms that you can use).

Now you get better monitoring and can find out bugs and unusual activities on each and every part of your web application. To wrap it up, I am going to give you differences in a single lines.

1. **Web Server**: Web server handles web requests sent by visitors visiting your website. Web server runs on Apache, Nginx, Microsoft IIS, etc.
2. **Application server**: Application server is the server that works between Web server and database server and basically manages, processes the data.
3. **Database Server**: Database server handles database queries. It runs on MySQL, PostgreSQL, MariaDB, etc.

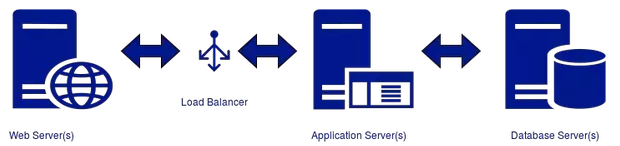
**What is an Application Server?**

An Application Server or shortly an APP SERVER is a combination of hardware and software of a computer in a network (Maybe restricted to increase the security) that provides the business logic for an application.

**What is a Web Server?**

A Web Server or Internet server is a combination of hardware and software of a computer in a system that delivers content or services to end users over the Internet & Intranet by using HTTP, HTTPS, FTP and other protocols.

In a three tier architecture (like the below image), web server will receive all the requests from user and forward only the business requests to application server.



**The static assets (like CSS, JS, Common images) will be served from your web server itself**. The use of Load Balancer is to distribute the load between multiple application servers. **Application server will be responsible for only business requests (like Login, Fetching details and etc,. ). Database server will have the data which runs business**.

Please note, application server can only accessed via web server, database server can only accessed by application server. This architecture is used to increase the security of the application.

We want to serve all the web requests through port 80. We have different type of content. Suppose you are running a Web application which serves static js, CSS, images as well as dynamic web pages. Assuming it is developed on tomcat.

Tomcat is not very fast and efficient in serving static files i.e Js, CSS and images. Nginx is pretty efficient in doing so that is, it takes lesser resources to serve static files as compared to tomcat.

Now, only one web server can be on the front because you want to serve every request through port 80.

A simple configuration in this case would be to keep nginx on port 80 to serve static content while reverse proxy the dynamic requests to tomcat. The user only sees one server on the front. Tomcat can be running behind the scenes on another port not directly accessible through internet.

Веб сервер нужен что бы отвечать на запросы клиентов – обрабатывает HTTP/HTTPs запросы. Существует **nginx** и **apache**. Какой на сайте используется сервер можно увидеть в **DevTools > Network > Response > Headers > Server**

**Apache** работает по такой схеме, что для каждого нового подключения (юзера на сайте) он создает процесс, который отжирает оперативную память, а так же ресурсы процессора

**Nginx** работает по схеме Мастер процесса и воркеров. Мастер процесс плодит ворекеры. 1 воркер, как правило, занимает 1 ядро сервера. 1 воркер уже способен обслужить десятки тысяч юзеров, что выйгрывает перед Apache