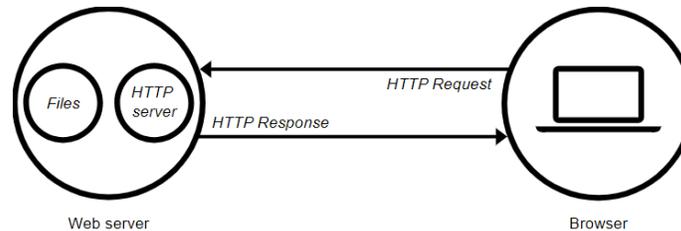




**WEBSERVERS**

## ❖ WEB SERVERS:

- A web server is a network service that **serves content (web pages)** to a client over the web.
- Web servers are also known as **HTTP (Hypertext Transport Protocol)** servers.
- **The Popular web servers are:** Apache HTTP and NGINX Server.



- To publish a website, you need either a static or a dynamic web server.

## STATIC WEB SERVER:

- A static web server, or stack, consists of a computer (hardware) with an HTTP server (software). We call it "static" because the server sends its hosted files as-is to your browser.

## DYNAMIC WEB SERVER:

- A dynamic web server consists of a static web server plus extra software, most commonly an application server and a database.
- We call it "dynamic" because the application server updates the hosted files before sending content to your browser via the HTTP server.



## ❖ **APACHE HTTP WEB SERVER:**

- Apache is an **open-source** web server developed by the **Apache Software Foundation (ASF)**.
- A web server is a network service that serves content to a client over the web. This typically means web pages.
- This is a **solid and stable** web server that has been around for years.
- It is also an option to use the **SSL protocol**, making **website safe and secure**.

## **MAIN CONFIGURATION FILES:**

<b>Package</b>	: httpd
<b>Document root location</b>	: /var/www/html
<b>Configuration Files</b>	: /etc/httpd/conf/httpd.conf /etc/httpd/conf.d/ssl.conf
<b>Auxiliary Directory</b>	: /etc/httpd/conf.d/
<b>Default web page</b>	: /etc/httpd/conf.d/welcome.conf
<b>Modules Location</b>	: /usr/lib64/httpd/modules
<b>Log Files Location</b>	: /var/log/httpd/
<b>Log Files</b>	: access_log, error_log
<b>Service / Daemon</b>	: httpd
<b>Ports</b>	: http-80 https/ssl-443

## ➤ **INSTALLING APACHE-HTTPD ON CENTOS9 / RHEL 9:**

### **STEP 1: Setting Hostname (FQDN)**

```
#hostname webservers.example.com  
#vim /etc/hostname  
webservers.example.com  
#vim /etc/hosts  
Ip-Address webservers.example.com webservers  
#bash
```

### **STEP2: Security-Enhanced Linux is being disabled or in permissive mode.**

```
#sed -i 's/SELINUX=.*SELINUX=disabled/g' /etc/selinux/config  
#setenforce 0
```

### **STEP 3: Installing HTTPD Server**

```
#yum update -y  
#yum install httpd -y  
#yum installed httpd  
#httpd -v
```

### **STEP 4: Start & Enable HTTPD Service**

```
#systemctl start httpd  
#systemctl enable httpd  
#systemctl status httpd
```

## STEP 5: Testing default web page

Connect with a web browser to

**http://server\_IP/\_host\_name**

### NOTE:

- If the `/var/www/html/` directory is empty or does not contain an **index.html** or **index.htm** file, Apache displays the Red Hat Enterprise Linux Test Page.
- If `/var/www/html/` contains HTML files with a different name, you can load them by entering the URL to that file, such as **http://server\_IP/\_host\_name/example.html**

### VIEWING LOG FILES:

- Log files can be very useful when trying to troubleshoot a problem with the system.
- **Apache log file location is:**

```
#cd /var/log/httpd
```

```
#ls
```

**access\_log:** A log file that records all events related to client applications and user access to a resource on a computer.

```
#tail -f access_log
```

**error\_log:** A file that contains detailed records of error conditions a computer software encounters when it's running.

```
#tail -f error_log
```

## **Making Sample Web page:**

Create a sample html file in the document root location

```
#cd /var/www/html  
  
#vim index.html  
  
<html>  
  
<body bgcolor=red text=blue>  
  
<marquee><H1> WELCOME TO SYSGEEKS...! </H1></marquee>  
  
</body>  
  
</html>  
  
#apachectl graceful
```

**NOTE:** Apache Graceful Restart means, Reloading the Apache Configuration (httpd.conf) without restarting the Web Server.

## **Testing a web page:**

Connect with a web browser to

**http://server\_IP/\_host\_name**



## ❖ **NGINX WEBSERVER:**

- NGINX is **open-source software** for **web serving, reverse proxying, caching, load balancing, media streaming**, and more.
- By default, NGINX act as a Web Server.
- It is faster than Apache webserver!
- NGINX is:
  - High performance
  - High Concurrency
  - Low Resource Usage

## ➤ **NGINX AS A WEB SERVER:**

- It is a fastest web server; the scalable underlying architecture has proved ideal for many web tasks beyond serving content.
- It supports all the components of the modern Web, including WebSocket, HTTP/2, gRPC, and streaming of multiple video formats (HDS, HLS, RTMP, and others).

## **MAIN CONFIGURATION FILES:**

<b>Package</b>	: nginx
<b>Document root location</b>	: /usr/share/nginx/html/
<b>Configuration Files</b>	: /etc/nginx/nginx.conf
<b>Log Files Location</b>	: /var/log/nginx/
<b>Log Files</b>	: access_log, error_log
<b>Service / Daemon</b>	: nginx
<b>Ports</b>	: httpd-80

## ➤ **INSTALLING NGINX ON CENTOS9 / RHEL 9:**

### **STEP 1: Setting Hostname (FQDN)**

```
#hostname webservr.example.com  
  
#vim /etc/hostname  
webservr.example.com  
  
#vim /etc/hosts  
  
Ip-Address webservr.example.com webservr  
  
#bash
```

### **STEP2: Security-Enhanced Linux is being disabled or in permissive mode.**

```
#sed -i 's/SELINUX=.*//SELINUX=disabled/g' /etc/selinux/config  
  
#setenforce 0
```

### **STEP 3: Installing NGINX Server**

```
#yum update -y  
  
#yum install nginx -y
```

### **STEP 4: Start & Enable NGINX Service**

```
#systemctl start nginx  
  
#systemctl enable nginx  
  
#systemctl status nginx
```

### **STEP 5: Testing default web page**

Connect with a web browser to **http://server\_IP/\_/host\_name**

## **CONFIGURING NGINX AS A WEB SERVER:**

It provides different content for different domains.

### **STEP 1: Edit the /etc/nginx/nginx.conf file:**

```
server {  
    server_name example.com;  
    root    /var/www/example.com/;  
    access_log /var/log/nginx/example.com/access.log;  
    error_log /var/log/nginx/example.com/error.log;  
}
```

### **NOTE:**

The **access\_log** directive defines a separate access log file for this domain.

The **error\_log** directive defines a separate error log file for this domain.

### **STEP 2: Create the root directories for both domains:**

```
# mkdir -p /var/www/example.com/
```

```
#cd /var/www/example.com/
```

```
#vim index.html
```

```
<html>
```

```
<body bgcolor=blue text=red>
```

```
<marquee><H1> WELCOME TO SYSGEEKS...! </H1></marquee>
```

```
</body>
```

```
</html>
```

### **STEP 3: Create a log directory for the domain:**

```
# mkdir /var/log/nginx/example.com/
```

#### **STEP 4: Restart the nginx service:**

```
# systemctl restart nginx
```

#### **Testing a web page**

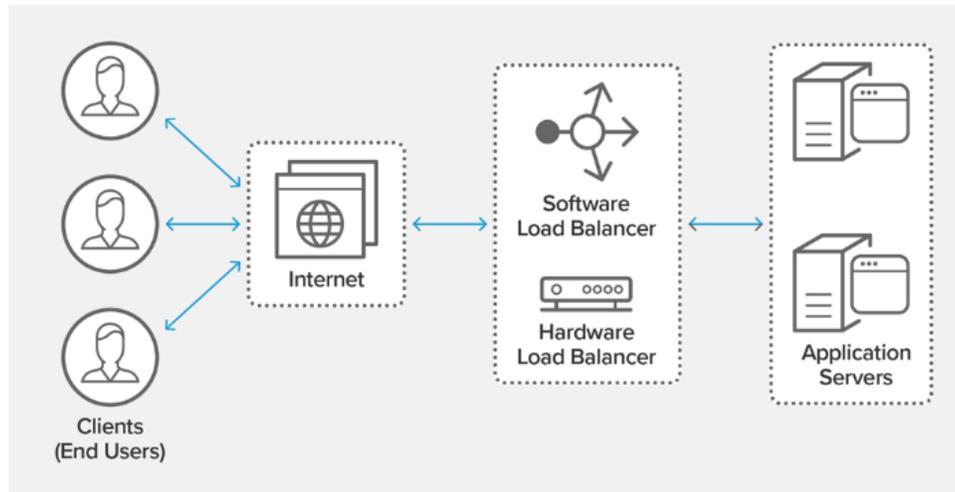
Connect with a web browser to **http://server\_IP/\_host\_name**

#### **➤ NGINX AS A REVERSE PROXY:**

- A reverse proxy is a type of proxy server that typically sits behind the firewall in a private network and directs client requests to the appropriate backend server.
- It provides additional level of abstraction and control to ensure the smooth flow of network traffic between clients and servers.
- Common uses for a reverse proxy server include:
  - Load balancing
  - Web acceleration
  - Security and anonymity

#### **NGINX AS A LOAD BALANCER:**

- It is used to efficiently distributing incoming network traffic across a group of backend servers.
- It maximizes speed and capacity utilization and ensures that no one server is overworked.
- A load balancer performs the following functions:
  - Distributes client requests or network load efficiently across multiple servers
  - Ensures high availability and reliability by sending requests only to servers that are online
  - Provides the flexibility to add or subtract servers as demand dictates



## **LOAD BALANCING ALGORITHMS:**

### **ROUND ROBIN:**

- Requests are distributed across the group of servers sequentially.

### **LEAST CONNECTIONS:**

- A new request is sent to the server with the fewest current connections to clients. The relative computing capacity of each server is factored into determining which one has the least connections.

### **LEAST TIME:**

- Sends requests to the server selected by a formula that combines the fastest response time and fewest active connections. Exclusive to NGINX Plus.

### **HASH:**

- Distributes requests based on a key you define, such as the client IP address or the request URL. NGINX Plus can optionally apply a consistent hash to minimize redistribution of loads if the set of upstream servers' changes.

### **IP HASH:**

- The IP address of the client is used to determine which server receives the request.

### **RANDOM WITH TWO CHOICES:**

- Picks two servers at random and sends the request to the one that is selected by then applying the Least Connections algorithm (or for NGINX Plus the Least Time algorithm, if so configured).