GETTING STARTED
WITH
WEBSERVERS

➢ THE WEB SERVERS:

- A web server is a network service that serves content to a client over the web. This typically means web pages, but any other documents can be served as well.
- Web servers are also known as **HTTP** servers, as they use **the hypertext transport protocol (HTTP).**
- The popular web servers are:
  - Apache HTTPD
  - Nginx
  - Apache Tomcat
  - MS IIS

APACHE HTTP SERVER:

- Apache is an **open-source** web server developed by the **Apache Software Foundation (ASF).**
- The Apache HTTP Server ("httpd") was launched in 1995 and it has been the most popular web server on the Internet since April 1996.
- 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.

❖ HTTPD INSTALLATION AND CONFIGURATION:

PRE-REQUISITES:

Package name	**:** httpd
Main config file	**:** /etc/httpd/conf/httpd.conf
Document root location	**:** /var/www/html/
Default web page	**:** /etc/httpd/conf.d/welcome.conf
Log Files location	**:** /var/log/httpd access_log & error_log
Service / Daemon	**:** httpd
Module's location	**:** /usr/lib64/httpd/modules
Ports	**:** HTTP – 80

→ **Installing apache httpd package:**

#dnf install httpd -y

#rpm -q httpd

→ **To check version information:**

#httpd -v

→ **Reload the systemd manager configuration:**

#systemctl daemon-reload

→ **Start and enable the named service:**

#systemctl start httpd

#systemctl enable httpd

→ **Verify the status of the bind:**

#systemctl status httpd

→ **Verify the port number of http:**

#netstat -pantl

#netstat -pantl | grep -i http

→ **IP-Address and Hostname mapping details:**

#vim /etc/hosts

192.168.10.254 server.example.com server

→ **Default web page location:**

#cat /etc/httpd/conf.d/welcome.conf

WEBPAGE VERIFICATION:

Go to web browser type: http://server.example.com

http://192.168.10.254

NOTE: If the /var/www/html/ directory is empty / does not contain an index.html file, Apache displays the Red Hat Enterprise Linux Test Page.

❖ VIRTUAL HOSTING:

- The term Virtual Host refers to the practice of running more than one web site (such as sysgeeks.com and raju.com) on a single machine.
- Virtual hosts can be **"IP-based",** meaning that you have a different IP address for every web site, or **"name-based",** meaning that you have multiple names running on each IP address.

NAME BASED VIRTUAL HOST:

- Name-based virtual hosts enable Apache to serve different content for different domains that resolve to the IP address of the server.
- Name-based virtual hosting is usually simpler, since you need only configure your DNS server to map each hostname to the correct IP address and then configure the Apache HTTP Server to recognize the different hostnames.

→ **Edit the /etc/httpd/conf/httpd.conf file:**

#vim /etc/httpd/conf/httpd.conf

```
<VirtualHost *:80>
   DocumentRoot "/var/www/sysgeeks/"
   ServerName www.sysgeeks.com
   CustomLog /var/log/httpd/sysgeeks_access.log combined
   ErrorLog /var/log/httpd/sysgeeks_error.log
   DiretoryIndex index.html
</VirtualHost>
```

→ **Append a similar virtual host configuration for the example domain:**

```
<VirtualHost *:80>
   DocumentRoot "/var/www/raju/"
   ServerName www.raju.com
   CustomLog /var/log/httpd/raju_access.log combined
   ErrorLog /var/log/httpd/raju_error.log
   DiretoryIndex index.html
</VirtualHost>
```

→ **Create the document roots for both virtual hosts:**
#mkdir /var/www/html/sysgeeks
#mkdir /var/www/html/raju

→ **Create a different example file in each virtual host's document root:**

#echo "This Is Sysgeeks Website...!" > /var/www/html/sysgeeks/index.html
#echo "This Is Example Website...!" > /var/www/html/raju/index.html

→ **Adding Website names in /etc/hosts file:**
192.168.10.154 www.sysgeeks.com
192.168.10.254 www.raju.com

→ **To check the configuration for possible errors:**
#apachectl configtest

→ **To reload Apache Configuration:**
#apachectl graceful

Use a browser and connect to **http://www.sysgeeks.com**

http://www.raju.com

IP-BASED VIRTUAL HOSTING:

- IP-based virtual hosting is a method to apply different directives based on the IP address and port a request is received on.
- Most commonly, this is used to serve different websites on different ports or interfaces.

→ **Edit /etc/httpd/conf/httpd.conf file:**

<VirtualHost **192.168.10.100**:80>
DocumentRoot "/var/www/ram/"
ServerName www.ram.com
CustomLog /var/log/httpd/ram_access.log combined
ErrorLog /var/log/httpd/ram_error.log
DiretoryIndex index.html
</VirtualHost>

→ **Create the document roots for both virtual hosts:**
#mkdir /var/www/html/ram
#echo "This Is Ram Website...!" > /var/www/html/ram/index.html

→ **Edit /etc/hosts file on your local machine and add following line:**
#vim /etc/hosts
192.168.10.100 www.ram.com

→ **To check the configuration for possible errors:**
#apachectl configtest

→ **To reload Apache Configuration:**
#apachectl graceful

Use a browser and connect to **http://www.ram.com**

PORT BASED HOSTING:

- Setting up your web server software to listen for incoming connections on particular ports and rerouting traffic to different websites based on the port number constitutes configuring port-based web hosting.

→ **Website running with port number 8000:**

#vim /etc/httpd/conf/httpd.conf
Listen 8000
<VirtualHost *:8000>
ServerAdmin root@server.example.com
DocumentRoot /var/www/html/ramu
ServerName www.ramu.com
ErrorLog logs/server.ramu.com-error_log
CustomLog logs/server.ramu.com-access_log common
DirectoryIndex index.html
</ VirtualHost>

→ **Create the document roots for both virtual hosts:**
#mkdir /var/www/html/ramu
#echo "This Is Ramu Website...!" > /var/www/html/ramu/index.html

→ **Edit /etc/hosts file on your local machine and add following line:**
#vim /etc/hosts
192.168.10.254 www.ramu.com

→ **To check the configuration for possible errors:**
#apachectl configtest

→ **To reload Apache Configuration:**
#apachectl graceful

Use a browser and connect to **http://www.ramu.com:8000**

WEBSITE USER AUTHENTICATION:

- In some cases, we may need to secure our website so that only authenticated users can access the website.
- We can protect our website at the server level using Apache Password Authentication.

→ **Edit /etc/httpd/conf/httpd.conf file:**
#vim /etc/httpd/conf/httpd.conf

```
<Directory "/var/www/html/sysgeeks">
Options Indexes
AllowOverride AuthConfig
</Directory>
```

→ **Go to Document root location and create a .htaccess file:**

#cd /var/www/html/sysgeeks/

#vim .htaccess

```
AuthType Basic
AuthName "WEB RESTRICTED AREA..."
AuthUserFile /etc/httpd/conf/.htpasswd
Require valid-user          ### or          Require User Sachin
```

→ **Create a user for web authentication:**

#useradd jai

htpasswd -c /etc/httpd/conf/.htpasswd jai

→ **Edit /etc/hosts file on your local machine and add following line:**
#vim /etc/hosts
192.168.10.254 www.sysgeeks.com

→ **To check the configuration for possible errors:**
#apachectl configtest

→ **To reload Apache Configuration:**
#apachectl graceful

Use a browser and connect to **http://www.sysgeeks.com**

LOG FILE VERIFICATION:

- On Red Hat Enterprise Linux, the default location for Apache logs are **/var/log/httpd/.**

Access Log: This file stores information about incoming requests. You'll find details about each request such as the requested resource, response codes, time taken to generate the response, IP address of the client, and more.

#tail -f access_log

Error Log: This file contains diagnostic information about any errors were encountered while processing requests.

#tail -f error_log