## What is a “Document Root”?

In RHEL (and CentOS/Fedora too), the **default document root** is:

***/var/www/html***

This means — if you open your browser and type:

`http://your-server-ip/`

Apache will show files from:

***/var/www/html/***

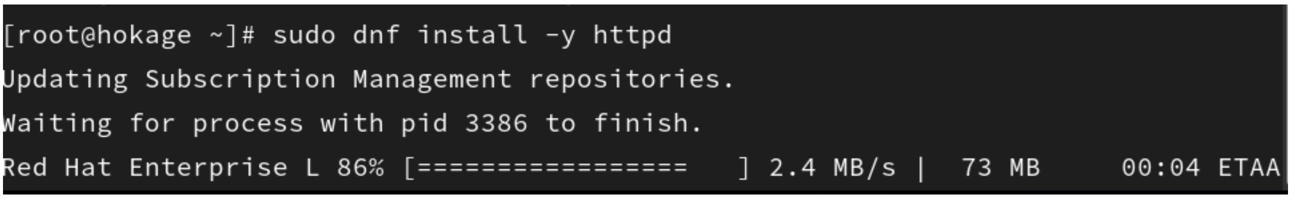
2□ What is `index.html`?

`index.html` is the **default homepage file** that Apache (or any web server) looks for automatically. So if `/var/www/html` contains a file named `index.html`, Apache will show it first.

**Install httpd?**

# install

sudo dnf install -y httpd

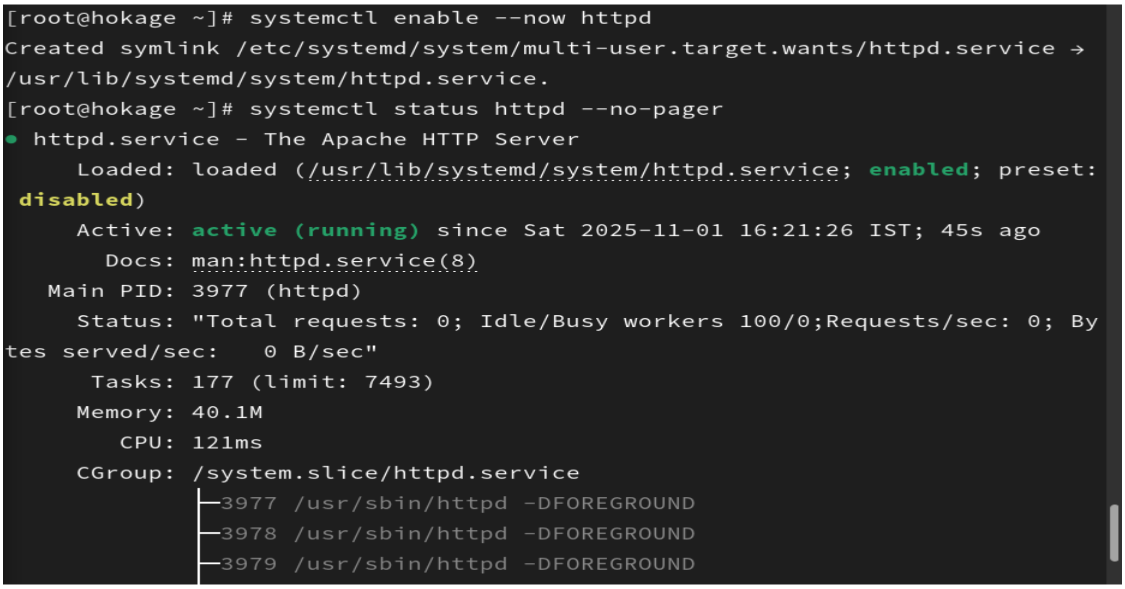


```
[root@hokage ~]# sudo dnf install -y httpd
Updating Subscription Management repositories.
Waiting for process with pid 3386 to finish.
Red Hat Enterprise L 86% [=================   ] 2.4 MB/s |  73 MB     00:04 ETAA
```

# enable & start

sudo systemctl enable --now httpd

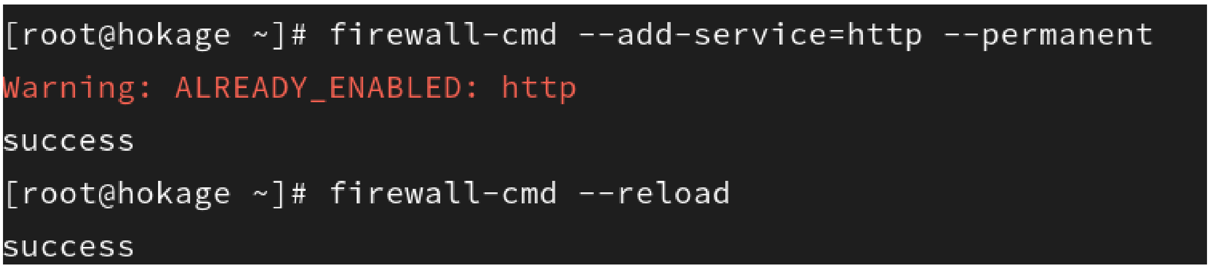
# check status



```
[root@hokage ~]# systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service →
/usr/lib/systemd/system/httpd.service.
[root@hokage ~]# systemctl status httpd --no-pager
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset:
 disabled)
     Active: active (running) since Sat 2025-11-01 16:21:26 IST; 45s ago
       Docs: man:httpd.service(8)
   Main PID: 3977 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; By
tes served/sec:   0 B/sec"
      Tasks: 177 (limit: 7493)
     Memory: 40.1M
        CPU: 121ms
     CGroup: /system.slice/httpd.service
             ├─3977 /usr/sbin/httpd -DFOREGROUND
             ├─3978 /usr/sbin/httpd -DFOREGROUND
             ├─3979 /usr/sbin/httpd -DFOREGROUND
```

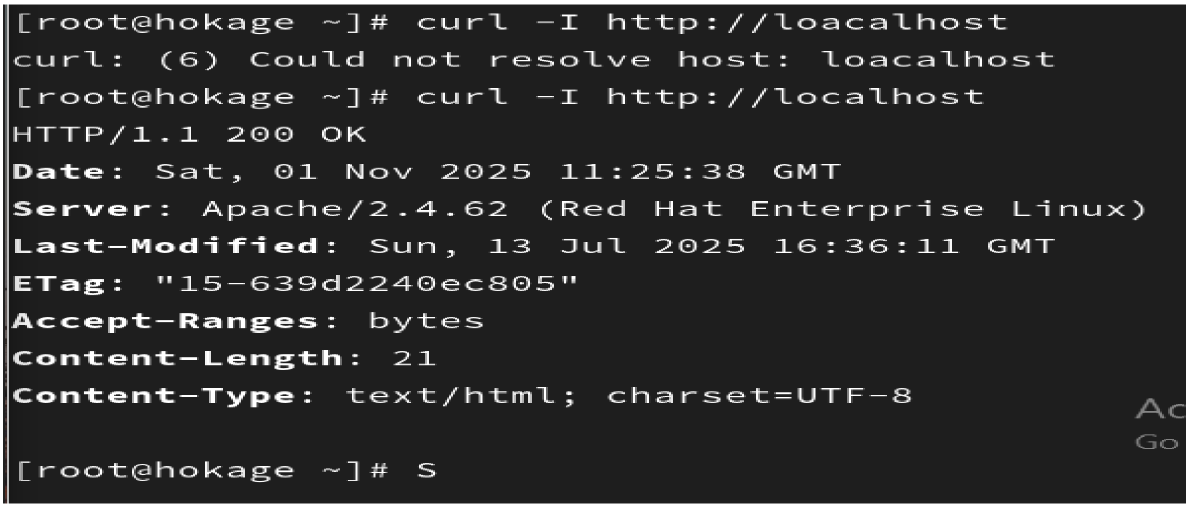
sudo systemctl status httpd --no-pager

sudo firewall-cmd --add-service=http --permanent sudo firewall-cmd –reload



```
[root@hokage ~]# firewall-cmd --add-service=http --permanent
Warning: ALREADY_ENABLED: http
success
[root@hokage ~]# firewall-cmd --reload
success
```

curl -I http://localhost

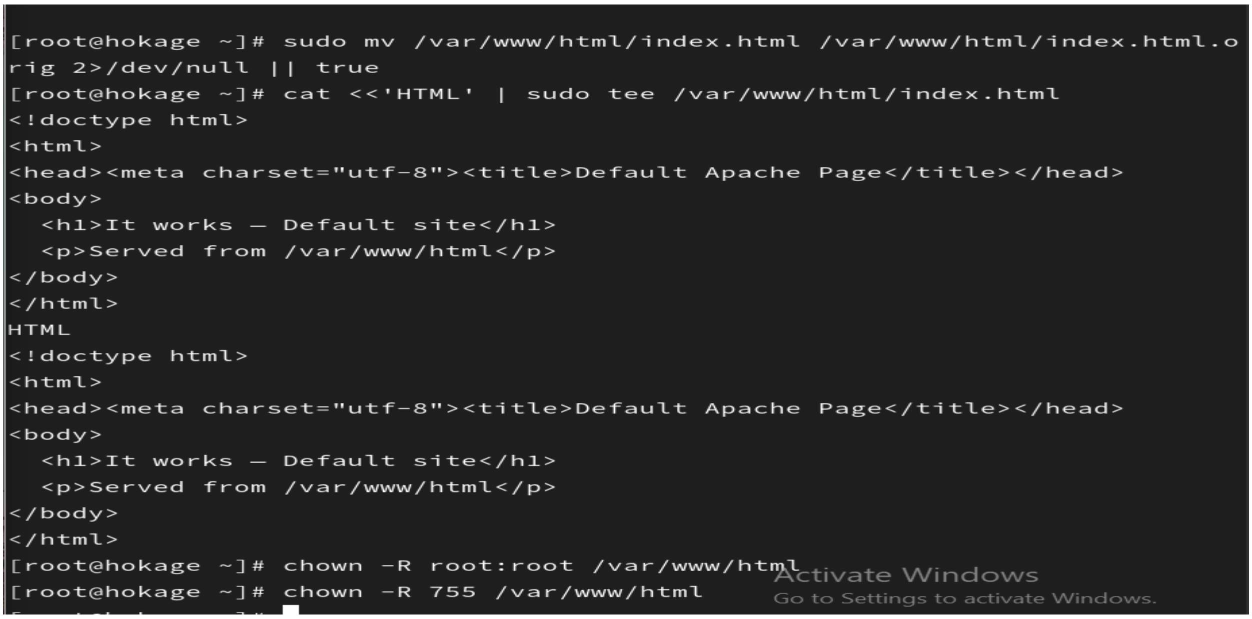


```
[root@hokage ~]# curl -I http://loacalhost
curl: (6) Could not resolve host: loacalhost
[root@hokage ~]# curl -I http://localhost
HTTP/1.1 200 OK
Date: Sat, 01 Nov 2025 11:25:38 GMT
Server: Apache/2.4.62 (Red Hat Enterprise Linux)
Last-Modified: Sun, 13 Jul 2025 16:36:11 GMT
ETag: "15-639d2240ec805"
Accept-Ranges: bytes
Content-Length: 21
Content-Type: text/html; charset=UTF-8

[root@hokage ~]# s
```

Qs1--

"Create an index.html page in the default document root" means:

answer:Create a file named `index.html` **inside** `/var/www/html`.

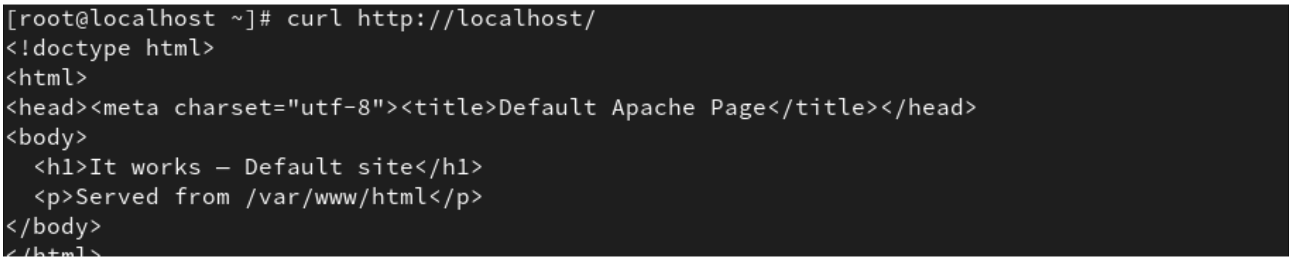


```
[root@hokage ~]# sudo mv /var/www/html/index.html /var/www/html/index.html.o
rig 2>/dev/null || true
[root@hokage ~]# cat <<'HTML' | sudo tee /var/www/html/index.html
<!doctype html>
<html>
<head><meta charset="utf-8"><title>Default Apache Page</title></head>
<body>
  <h1>It works — Default site</h1>
  <p>Served from /var/www/html</p>
</body>
</html>
HTML
<!doctype html>
<html>
<head><meta charset="utf-8"><title>Default Apache Page</title></head>
<body>
  <h1>It works — Default site</h1>
  <p>Served from /var/www/html</p>
</body>
</html>
[root@hokage ~]# chown -R root:root /var/www/html
[root@hokage ~]# chown -R 755 /var/www/html
```

Activate Windows
Go to Settings to activate Windows.

curl http://localhost/

Qs2--Access the website using mysamplewebsite.itfs  means:



```
[root@localhost ~]# curl http://localhost/
<!doctype html>
<html>
<head><meta charset="utf-8"><title>Default Apache Page</title></head>
<body>
  <h1>It works – Default site</h1>
  <p>Served from /var/www/html</p>
</body>
</html>
```

answer:should be able to **open your website in a web browser** (like Chrome or Firefox) **by typing the *domain name `mysamplewebsite.itfs`*** — **instead of using the IP address** (like `192.168.133.128`).

## Make the hostname resolve (for local testing)

On the machine from which you will access the site (your browser or the VM), add an entry to `/etc/hosts`.

If testing from the same VM:

echo "127.0.0.1  mysamplewebsite.itfs www.mysamplewebsite.itfs" | sudo tee -a /etc/hosts
here i am using a windows for see the output so
On your **Windows host system**:

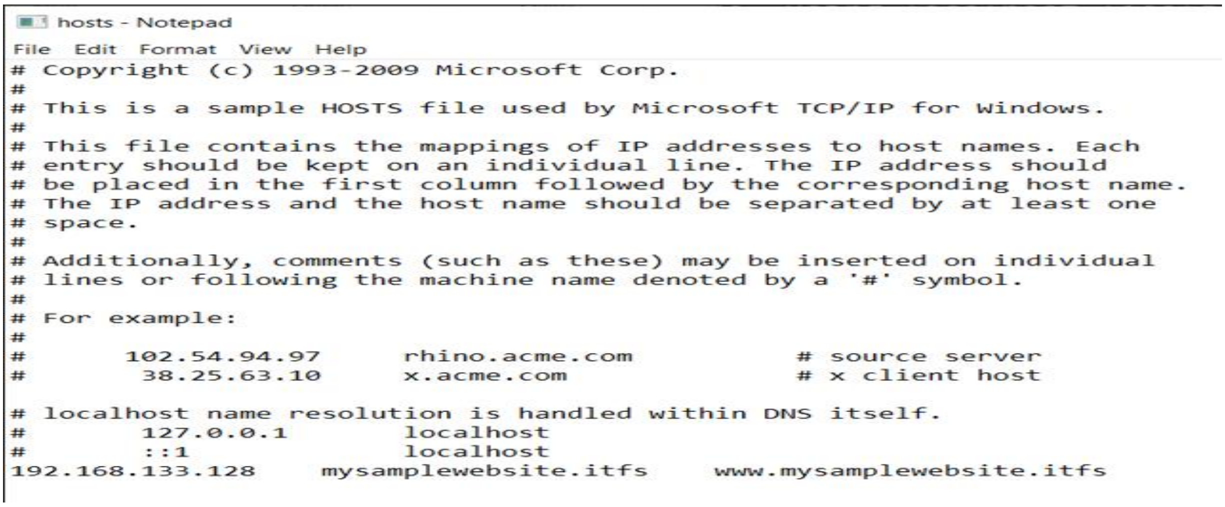
1. Press **Start → Notepad → Right-click → Run as Administrator**
2. Open this file:

```
C:\Windows\System32\drivers\etc\hosts
```

3. Add this line at the bottom (replace IP with your RHEL IP):

```
192.168.182.129   mysamplewebsite.itfs www.mysamplewebsite.itfs
```

4. Save the file.



```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
192.168.133.128    mysamplewebsite.itfs    www.mysamplewebsite.itfs
```

## Clear DNS cache

In Windows Command Prompt (as Administrator):

```
ipconfig /flushdns
```

**Why:**
This clears old DNS results so your system picks up the new `/etc/hosts` entry immediately.

### Qs3--Created the Custom Virtual Host File

answer:Apache configuration files are located in:

**`/etc/httpd/conf/`**

and

**`/etc/httpd/conf.d/`**

The main file is:

***`/etc/httpd/conf/httpd.conf`***

But instead of editing this main file directly (which is risky),
we created a **new custom configuration file** for your website inside the `conf.d` folder:

***`/etc/httpd/conf.d/mysamplewebsite.conf`***

## How

You created this file manually using a text editor like `vi` or `nano`.

Example:

```
vi /etc/httpd/conf.d/mysamplewebsite.conf
```

Inside it, you wrote:

```
<VirtualHost *:80>

    ServerName mysamplewebsite.itfs

    DocumentRoot /var/www/mysamplewebsite

</VirtualHost>
```

✅**Purpose:**
This tells Apache to serve your new domain `mysamplewebsite.itfs`
and fetch files from `/var/www/mysamplewebsite`.

### □ Why:

Instead of editing the global Apache configuration,
you made a **custom config file** — this is safer, cleaner, and easier to manage multiple sites.

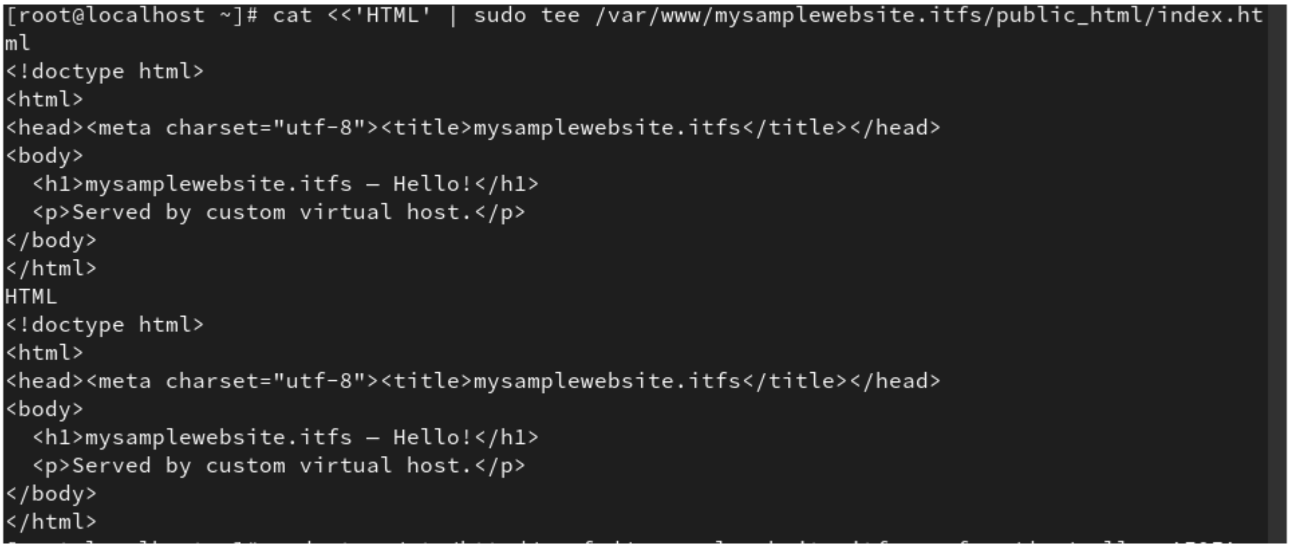
## Create a custom Virtual Host for `mysamplewebsite.itfs`

created a vhost file under `/etc/httpd/conf.d/`.

1. Create document root for this site:

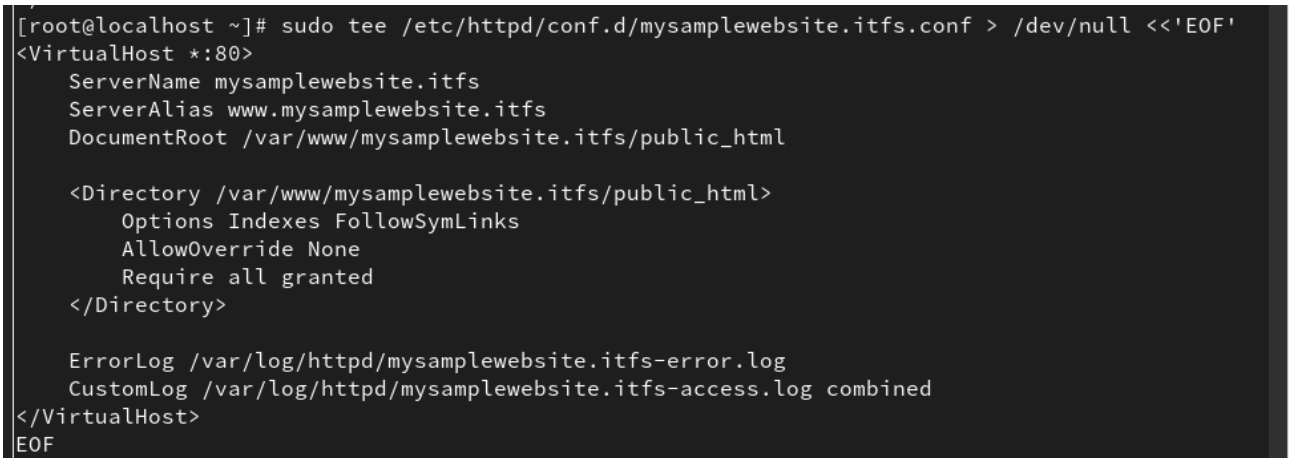
```
sudo mkdir -p /var/www/mysamplewebsite.itfs/public_html

sudo chown -R root:root /var/www/mysamplewebsite.itfs

sudo chmod -R 755 /var/www/mysamplewebsite.itfs
```

2. Create index for the vhost:



```
[root@localhost ~]# cat <<'HTML' | sudo tee /var/www/mysamplewebsite.itfs/public_html/index.ht
ml
<!doctype html>
<html>
<head><meta charset="utf-8"><title>mysamplewebsite.itfs</title></head>
<body>
  <h1>mysamplewebsite.itfs – Hello!</h1>
  <p>Served by custom virtual host.</p>
</body>
</html>
HTML
<!doctype html>
<html>
<head><meta charset="utf-8"><title>mysamplewebsite.itfs</title></head>
<body>
  <h1>mysamplewebsite.itfs – Hello!</h1>
  <p>Served by custom virtual host.</p>
</body>
</html>
```

3. Create virtual host config:

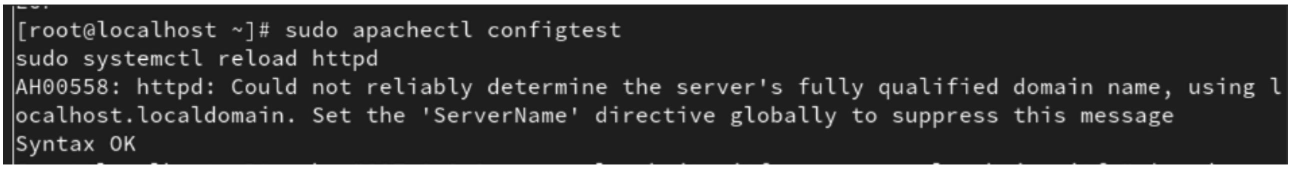


```
[root@localhost ~]# sudo tee /etc/httpd/conf.d/mysamplewebsite.itfs.conf > /dev/null <<'EOF'
<VirtualHost *:80>
    ServerName mysamplewebsite.itfs
    ServerAlias www.mysamplewebsite.itfs
    DocumentRoot /var/www/mysamplewebsite.itfs/public_html

    <Directory /var/www/mysamplewebsite.itfs/public_html>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>

    ErrorLog /var/log/httpd/mysamplewebsite.itfs-error.log
    CustomLog /var/log/httpd/mysamplewebsite.itfs-access.log combined
</VirtualHost>
EOF
```

4. Test Apache config and reload:

```
sudo apachectl configtest

sudo systemctl reload httpd
```

If `apachectl configtest` says `Syntax OK`, proceed.



```
[root@localhost ~]# sudo apachectl configtest
sudo systemctl reload httpd
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using l
ocalhost.localdomain. Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

## Qs4--Where and How To Took Backup of Configuration Files

answer:Backup Before Creating a Virtual Host

If there was already a configuration file:

```
cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.bak
```

✅This creates a copy named `httpd.conf.bak` in the same folder.

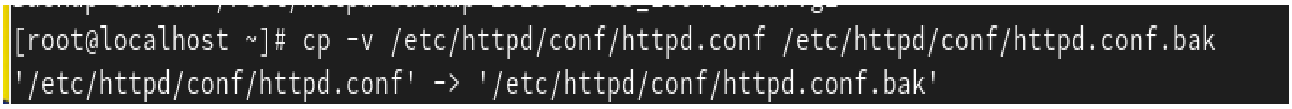
If something goes wrong, you can restore it:

**`mv /etc/httpd/conf/httpd.conf.bak /etc/httpd/conf/httpd.conf`**

## Backup of Your Custom Virtual Host

After creating `mysamplewebsite.conf`, you can back it up too:

***cp /etc/httpd/conf.d/mysamplewebsite.conf /etc/httpd/conf.d/mysamplewebsite.conf.bak***



```
[root@localhost ~]# cp -v /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.bak
'/etc/httpd/conf/httpd.conf' -> '/etc/httpd/conf/httpd.conf.bak'
```

### Backup `/etc/hosts` (on Linux or Windows)

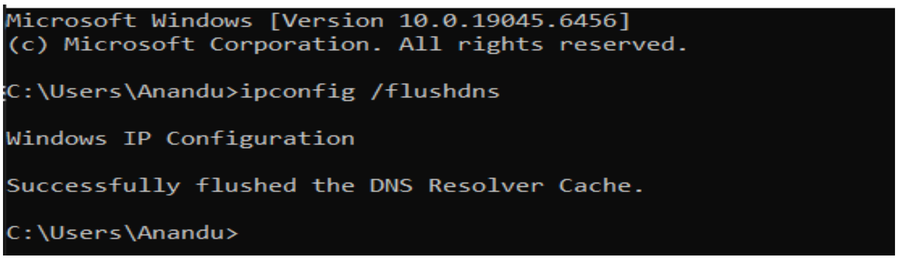
If you are editing `/etc/hosts` (to add the line for `mysamplewebsite.itfs`), take a backup like this:

□ On Linux:

```
cp /etc/hosts /etc/hosts.bak
```

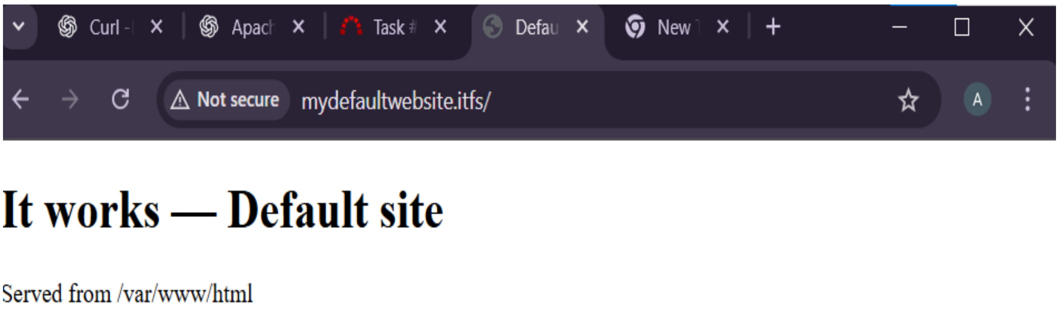
□ On Windows:

***Copy*** *`C:\Windows\System32\drivers\etc\hosts`*
to your Desktop or Documents folder before editing.



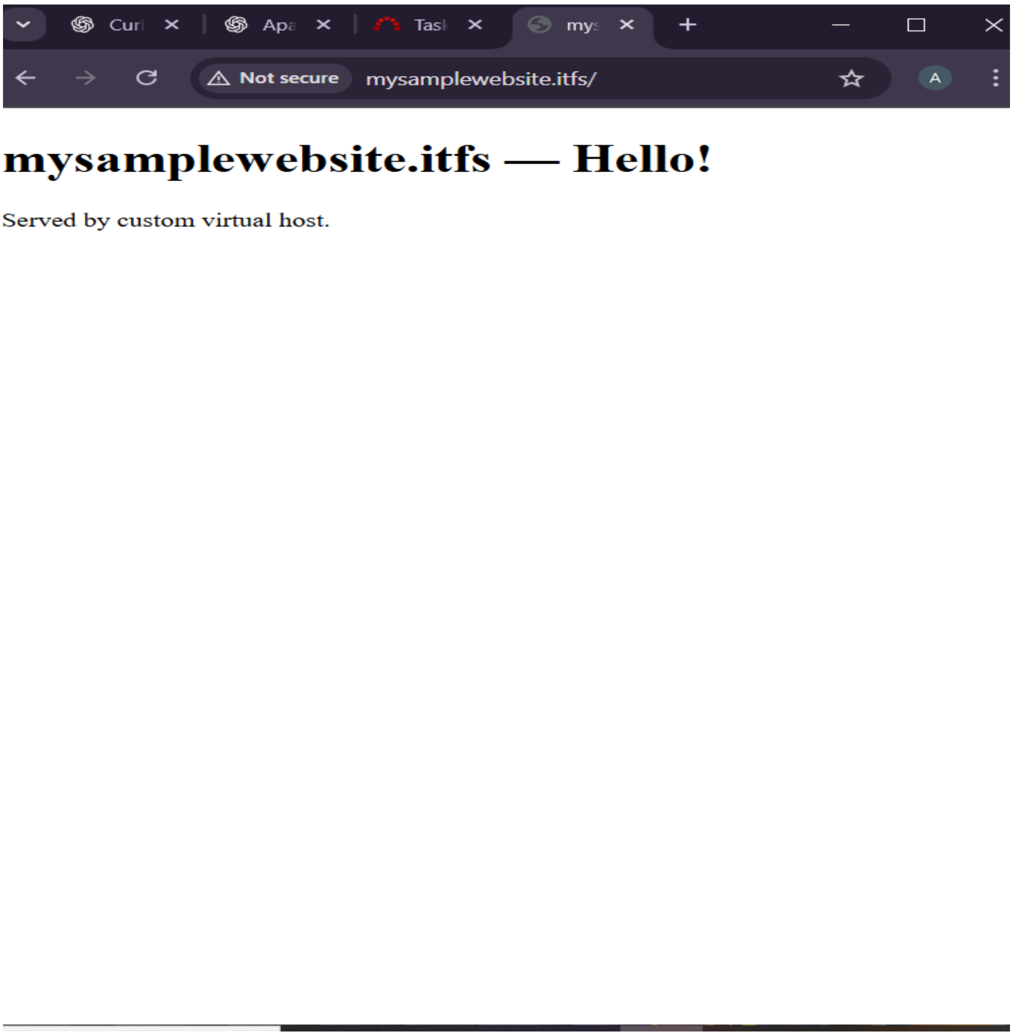
```
Microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Anandu>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\Anandu>
```

**RESULT**



# It works — Default site

Served from /var/www/html



mysamplewebsite.itfs — Hello!

Served by custom virtual host.

# Apache Default Files, Document Root, and Modules (Explained Simply)

## □ 1 Apache Default Files

When you install Apache (`httpd` package) on RHEL/Rocky Linux/CentOS, it automatically creates several configuration files and folders.

Let's understand what each main one does:

| File/Directory | Path | Description |
|---|---|---|
| **Main config file** | `/etc/httpd/conf/httpd.conf` | This is the main configuration file Apache reads at startup. It contains global settings, like server admin, ports, default document root, etc. |
| **Extra config directory** | `/etc/httpd/conf.d/` | This directory holds additional configuration files (like custom virtual hosts, SSL configs, etc.). Every `.conf` file here is loaded automatically when Apache starts. |

| File/Directory | Path | Description |
|---|---|---|
| **Module config directory** | `/etc/httpd/conf.modules.d/` | Contains config files that load Apache modules (like SSL, PHP, rewrite, etc.). |
| **Default website folder** | `/var/www/html` | This is the default folder Apache uses to serve web pages if no custom virtual host is defined. |
| **Log files** | `/var/log/httpd/access_log`, `/var/log/httpd/error_log` | These files store access and error logs for troubleshooting. |
| **Service control file** | `/usr/lib/systemd/system/httpd.service` | Defines how Apache runs as a system service. |

## □ 2 Document Root (Very Important)

□ What It Means:

The **DocumentRoot** is the directory on the server where Apache looks for website files (like HTML, CSS, JS, etc.).

□ Default Path:

```
/var/www/html
```

If you open your browser and type your server's IP (for example `http://192.168.133.128`), Apache looks inside `/var/www/html/index.html` and displays it.

□ Example inside **`httpd.conf`**:

```
DocumentRoot "/var/www/html"

<Directory "/var/www/html">

    AllowOverride None

    Require all granted

</Directory>
```

This means:
□ Files in `/var/www/html` will be publicly visible when accessed via the browser.

□ When we created a virtual host earlier:

We changed the DocumentRoot to `/var/www/mysamplewebsite`
so Apache serves our custom site instead of the default one.

## □ 3 Apache Modules

□ What They Are:

Modules are **add-ons or plug-ins** that extend Apache's functionality.

They let Apache handle things like:

- PHP pages
- SSL/HTTPS connections
- URL rewriting
- Authentication
- Compression, etc.

□ Location:

Configuration for loaded modules is in:

```
/etc/httpd/conf.modules.d/
```

□ Example Module File:

`00-base.conf`, `00-mpm.conf`, `00-systemd.conf`, etc.

□ Example Command to See All Enabled Modules:

```
httpd -M
```

✓Sample Output:

```
Loaded Modules:
 core_module (static)
 so_module (static)
 http_core_module (static)
 mpm_prefork_module (shared)
 authz_core_module (shared)
 mime_module (shared)
 dir_module (shared)
 alias_module (shared)
 ...
```

Each of these modules performs a special job.
For example:

| **Module Name** | **Function** |
|---|---|
| `mod_ssl` | Enables HTTPS support |
| `mod_rewrite` | Allows rewriting URLs |
| `mod_dir` | Serves index.html automatically |
| `mod_alias` | Maps URLs to filesystem paths |
| `mod_mime` | Handles file types (HTML, CSS, JS, etc.) |