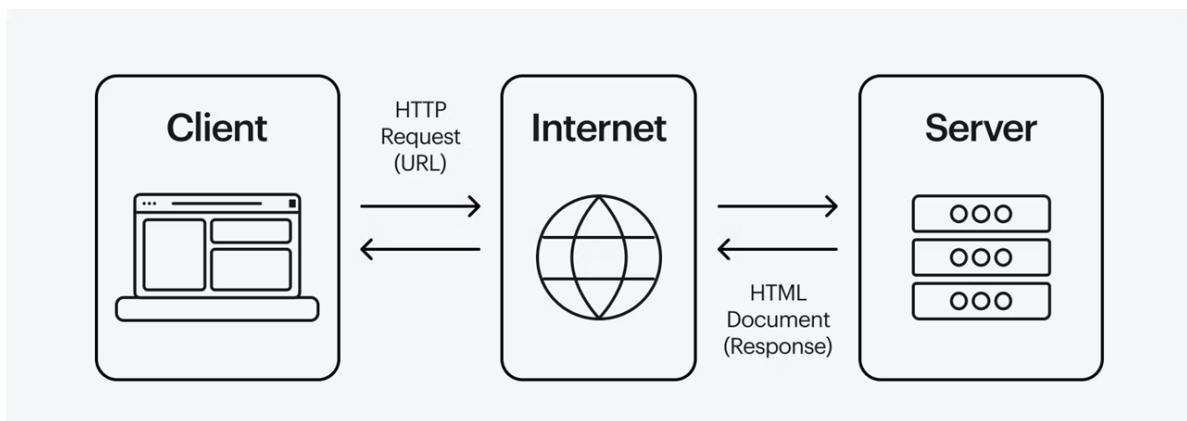


Web Server

A web server is software (sometimes the term also includes the hardware) that stores website files and sends them to a browser when requested. It listens for requests over the HTTP or HTTPS protocol and responds with the requested content, such as HTML pages, images, or files.

How a Web Server Works

When a user enters a website address in a web browser, the browser sends an HTTP request to the web server, the web server receives the request, locates the requested web page or file, and then sends it back to the browser, which displays the content to the user.



Static Website

A static website is a website whose content remains the same for every visitor and is delivered directly by the web server without generating content dynamically. It is built using HTML, CSS, JavaScript, images, and other fixed files.

Features of a Static Website

- Loads very fast because the server just sends ready-made files without any extra processing.
- Is simple and cheap to host since it does not need a database or backend logic.
- Is generally more secure because there is no server-side code that can be exploited.
- Is harder to update automatically, since every content change needs a manual file edit.

HTTPD Server

Apache HTTP Server (HTTPD) is a powerful and open-source web server developed by the Apache Software Foundation. It is widely used to host websites and web applications on various operating systems, including Linux, Windows, and macOS.

HTTPD receives requests from web browsers and delivers the requested web pages, images, and other web resources to users. It supports hosting both static websites and dynamic web applications, making it suitable for a wide range of web hosting needs.

One of the key features of Apache HTTPD is its modular architecture. Administrators can enable or disable modules to add specific functionality such as security features, URL rewriting, virtual hosting, and support for scripting languages. Its flexibility, stability, and extensive community support have made it one of the most popular web servers in the world.

Apache HTTPD is commonly used in development, testing, and production environments to provide reliable web services and efficiently handle client requests.

Installing Apache HTTPD Server

Before hosting a website, Apache HTTPD must be installed and configured on the server. The installation process varies depending on the Linux distribution being used.

- On Red Hat Enterprise Linux (RHEL), Rocky Linux, AlmaLinux, and CentOS, Apache can be installed using the dnf or yum package manager.
- The package is commonly named **httpd**.

```
root@angtsering:~# yum -y install httpd
Updating Subscription Management repositories.
Last metadata expiration check: 0:08:25 ago on Sat 01 Aug 2026 05:23:32 PM +0545.
Dependencies resolved.
=====
Package                        Architecture      Version           Repository        Size
=====
Installing:
httpd                          x86_64            2.4.63-13.el10    AppStream          56 k
```

- After installing the HTTPD package, the service is usually not running and may not be enabled to start automatically at boot.
- Before starting the HTTPD service, verify whether the package is installed on the system by running:

```
root@angtsering:~# rpm -q httpd
httpd-2.4.63-13.el10.x86_64
root@angtsering:~#
```

- Check the current status of the HTTPD service by running:

```
root@angtsering:~# systemctl status httpd
○ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
```

- To start the service immediately and enable it to start automatically after every system reboot, run:

```
root@angtsering:~# systemctl enable httpd --now
Created symlink '/etc/systemd/system/multi-user.target.wants/httpd.service' → '/usr/lib/systemd/system/httpd.service'.
root@angtsering:~#
```

```
root@angtsering:~# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2026-08-01 18:24:30 +0545; 37s ago
```

- The **--now** option performs two actions in a single command:
 1. Starts the HTTPD service immediately.
 2. Enables the HTTPD service to start automatically at system boot.

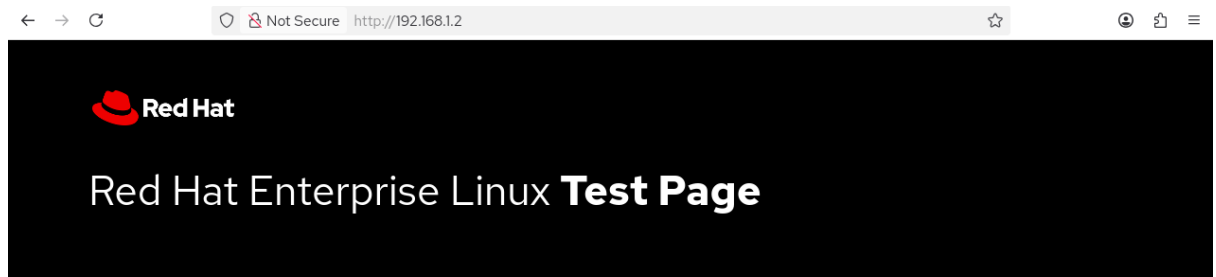
If the HTTPD service status is displayed as Active (running), perform the following steps:

- Verify that the web server is responding and displaying the default web page:

```
root@angtsering:~# curl localhost
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Test Page for the HTTP Server on Red Hat Enterprise Linux</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <style type="text/css">
      /**/</pre></div><div data-bbox="114 278 878 465" data-label="Image"><img alt="Screenshot of a web browser showing the Red Hat Enterprise Linux Test Page. The page has a black header with the Red Hat logo and the text 'Red Hat Enterprise Linux Test Page'. Below the header, a message states: 'This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page, it means that the HTTP server installed at this site is working properly.' The browser's address bar shows 'http://localhost/#contact'."/></div><div data-bbox="147 479 815 513" data-label="List-Group"><ul><li>• To allow access to the web server from a browser, open the HTTP and HTTPS services in the firewall:</li></ul></div><div data-bbox="181 531 873 555" data-label="Text"><pre>root@angtsering:~# firewall-cmd --permanent --add-service=http --add-service=https --zone=public
success</pre></div><div data-bbox="147 560 626 577" data-label="List-Group"><ul><li>• Reload the firewall configuration to apply the changes:</li></ul></div><div data-bbox="181 594 472 630" data-label="Text"><pre>root@angtsering:~# firewall-cmd --reload
success
root@angtsering:~#</pre></div><div data-bbox="147 637 883 689" data-label="List-Group"><ul><li>• To check the system's IP address, run <b>ip addr</b> or <b>ip a</b>, then identify the active network interface (such as <b>ens160</b>, <b>ens192</b>, <b>ens33</b>, or <b>eth0</b>) and note the IP address displayed next to the <b>inet</b> field.</li></ul></div><div data-bbox="118 697 803 906" data-label="Text"><pre>root@angtsering:~# ip a
1: lo: &lt;LOOPBACK,UP,LOWER_UP&gt; mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens160: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether 00:0c:29:df:49:58 brd ff:ff:ff:ff:ff:ff
   altname enp3s0
   altname enx000c29df4958
   inet 192.168.1.2/24 brd 192.168.1.255 scope global noprefixroute ens160
       valid_lft forever preferred_lft forever
   inet6 2400:1a00:3b47:87a3:20c:29ff:fedf:4958/64 scope global dynamic noprefixroute
       valid_lft 1077sec preferred_lft 1077sec
   inet6 fe80::20c:29ff:fedf:4958/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
root@angtsering:~# █</pre></div>
```

- In the above output:
ens160 is the network interface name.
192.168.1.2 is the IP address of the system.
- Access the web server from a browser using the server's IP address:



This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page, it means that the HTTP server installed at this site is working properly.

After verifying that the default Apache web page is accessible using the server's IP address, you can configure a custom domain name for easier access and testing purposes. This can be done by adding an entry in the `/etc/hosts` file, which maps your server's IP address to a domain name.

- Edit the `/etc/hosts` file and add your domain name: **`vim /etc/hosts`**

```
# Loopback entries; do not change.
# For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.example.org foo
# 192.168.1.13 bar.example.org bar
192.168.1.2 angtsering.com
~
```

The `/etc/hosts` file is used for local hostname resolution and allows the system to translate a domain name into an IP address without requiring a DNS server.

- Verify the configuration:

```
# Loopback entries; do not change.
# For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.example.org foo
# 192.168.1.13 bar.example.org bar
192.168.1.2 angtsering.com
~
```

- After adding the entry, you can access the web server using the domain name instead of the IP address.



This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page, it means that the HTTP server installed at this site is working properly.

To host your own content, website files should be placed in the default web root directory `/var/www/html/`. Any file stored in this directory can be accessed through the web browser using the server's IP address or configured domain name.

- Create a simple web page and verify the content locally:

```
root@angtsering:~# echo "Hello ,This is my first Web Hosting Project">>/var/www/html/index.html
root@angtsering:~# curl angtsering.com
Hello ,This is my first Web Hosting Project
root@angtsering:~#
```

- Open a web browser and access the website using the server's IP address or configured domain name:



Now, To host own static website

- Navigate to the default web root directory (`/var/www/html`) and create the default web page file, `index.html`, which will be served automatically by Apache HTTPD when users access the website.

```
root@angtsering:/var/www/html# ll
total 836
-rwxr--r--. 1 root root 828238 Aug  1 18:42 ang.jpeg
-rw-r--r--. 1 root root  22154 Aug  1 18:40 index.html
root@angtsering:/var/www/html#
```

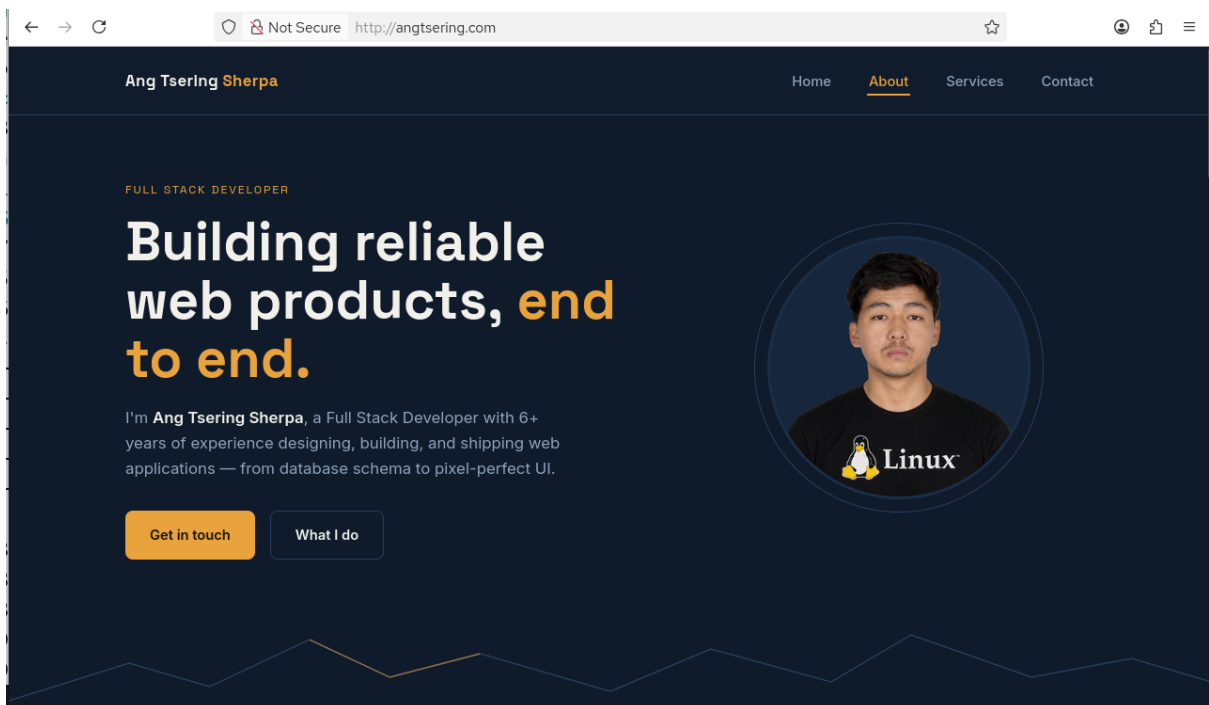
*The main web page file should be named **index.html** because Apache HTTPD looks for this file by default in the web root directory `/var/www/html/`.*

- After creating and saving the index.html file, restart the HTTPD service:

```
root@angtsering:~# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2026-08-01 19:49:49 +0545; 6s ago
 Invocation: 086c805dca41499c824f2d2e01938188
    Docs: man:httpd.service(8)
   Main PID: 11030 (httpd)
      Status: "Started, listening on: port 80"
     Tasks: 177 (limit: 10114)
    Memory: 15.7M (peak: 16M)
       CPU: 666ms
    CGroup: /system.slice/httpd.service
            └─11030 /usr/sbin/httpd -DFOREGROUND
              └─11031 /usr/sbin/httpd -DFOREGROUND
                └─11032 /usr/sbin/httpd -DFOREGROUND
                  └─11033 /usr/sbin/httpd -DFOREGROUND
                    └─11064 /usr/sbin/httpd -DFOREGROUND

Aug 01 19:49:48 angtsering systemd[1]: Starting httpd.service - The Apache HTTP Server...
Aug 01 19:49:48 angtsering (httpd)[11030]: httpd.service: Referenced but unset environment variable evaluates to an empty string
Aug 01 19:49:48 angtsering httpd[11030]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, setting 'ServerName' to ''
Aug 01 19:49:49 angtsering systemd[1]: Started httpd.service - The Apache HTTP Server.
Aug 01 19:49:49 angtsering httpd[11030]: Server configured, listening on: port 80
lines 1-22/22 (END)
```

Open a web browser and verify the static website by entering the server's IP address or configured domain name in the address bar. If the configuration is correct, the content of the index.html file will be displayed.



Steps to Access the Linux HTTPD Website from Windows

- Verify network connectivity from Windows by opening Command Prompt and pinging the Linux server's IP address to ensure both systems can communicate successfully.

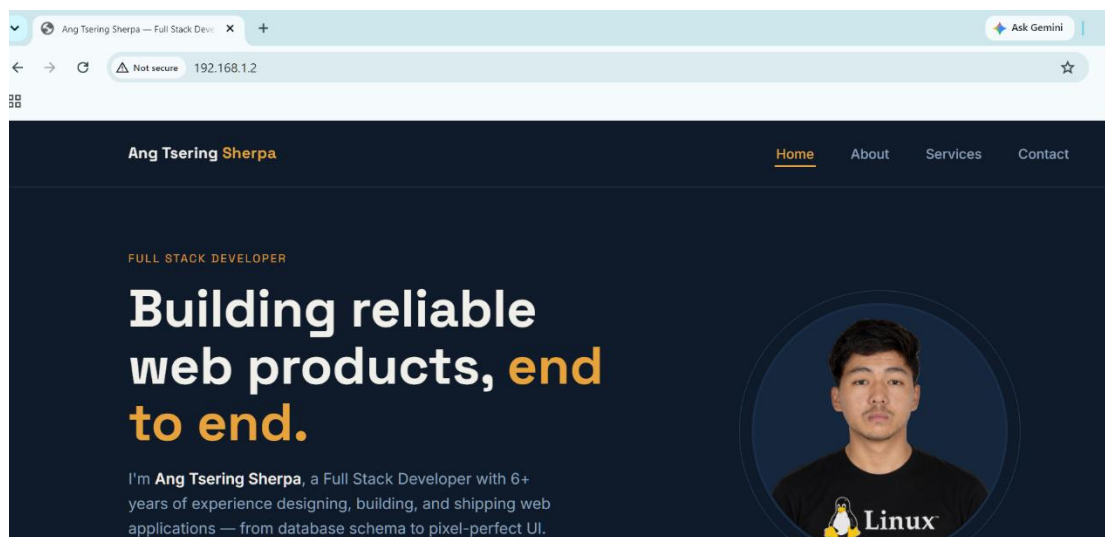
```
C:\Users\adawa>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=1ms TTL=64
Reply from 192.168.1.2: bytes=32 time<1ms TTL=64
Reply from 192.168.1.2: bytes=32 time<1ms TTL=64
Reply from 192.168.1.2: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Note: If successful replies are received, the Windows system can reach the Linux server over the network.

- Open a web browser on the Windows system and enter the Linux server's IP address in the address bar to access the hosted website.



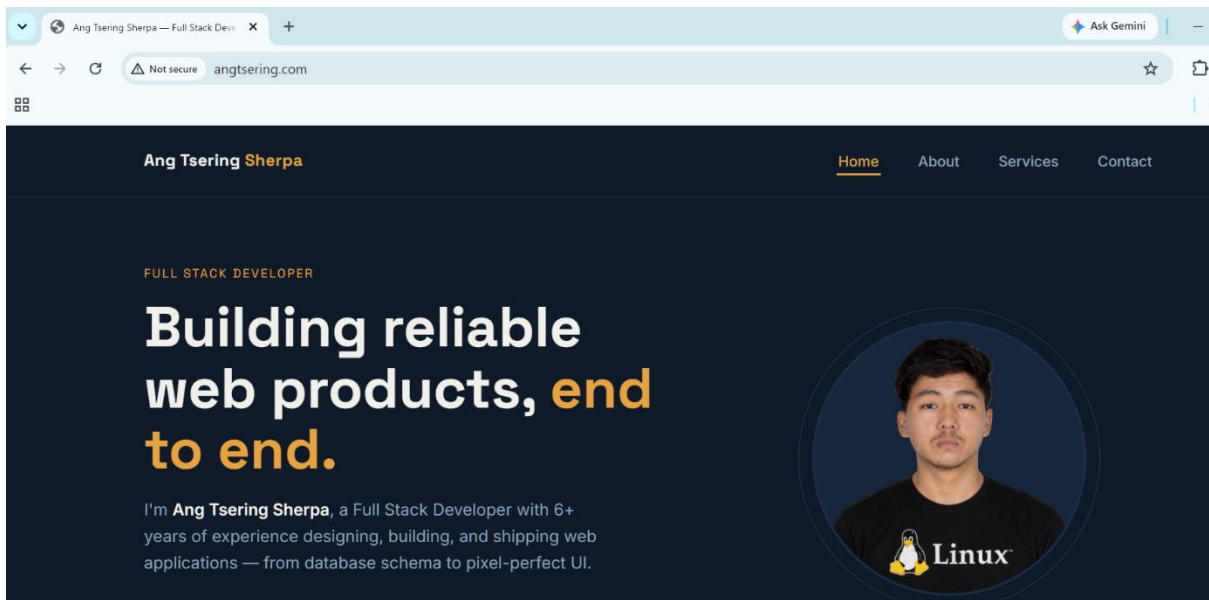
If the HTTPD service is running and the firewall is configured correctly, the website hosted in /var/www/html will be displayed.

Accessing the Website Using a Custom Domain Name

- Open the hosts file located at **C:\Windows\System32\drivers\etc\hosts** using Notepad with administrative privileges, add an entry that maps the Linux server's IP address to the desired domain name, save the file, and then access the website through the configured domain name in a web browser.

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
0.0.0.0 genuine-software.autodesk.com
0.0.0.0 genuine-software1.autodesk.com
0.0.0.0 genuine-software2.autodesk.com
0.0.0.0 api.genuine-software.autodesk.com
0.0.0.0 api.ase-cdn-stg.autodesk.com
192.168.1.2 angtsering.com
```

- After saving the file, open a web browser and navigate:



Conclusion:

This is how you can host a static website using Apache HTTPD by configuring the web server, creating an index.html file in /var/www/html, allowing web traffic through the firewall, and accessing the website from both Linux and Windows systems using the server's IP address or a custom domain name through a web browser.