



# Install and configure a WEB server

---

UBUNTU 22.04

ESPRIT 2023/2024

## **Requirements:**

- A newly installed Ubuntu 22.04
- Internet connection
- root privilege

## **Goals:**

- Install Apache Web server
- Set up name based virtual host
- Set up IP based virtual host
- Set up port based virtual host

## **Introduction :**

The Apache web server is a popular method for serving websites on the internet. As of 2019, it is estimated to serve 29% of all active websites and it offers robustness and flexibility for developers. Using Apache, an administrator can set up one server to host multiple domains or sites off of a single interface or IP by using a matching system. Each domain or individual site — known as a “virtual host” — that is configured using Apache will direct the visitor to a specific directory holding that site’s information. This is done without indicating that the same server is also responsible for other sites. This scheme is expandable without any software limit as long as your server can handle the load. The basic unit that describes an individual site or domain is called a virtual host. How to set up Apache virtual hosts on an Ubuntu 22.04 server. During this process, you’ll learn how to serve different content to different visitors depending on which domains they are requesting using Name Based, IP Based and Port Based virtual hosts. The World Wide Web commonly known as the web is a system of interlinked hypertext documents accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks.

### ❖ HTTP protocol :

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems.

HTTP functions as a request-response protocol in the client-server computing model. A web browser, for example, may be the client and an application running on a computer hosting a web site may be the server. The client submits an HTTP request message to the server.

### ❖ Port numbers :

A port is associated with an IP address of the host, as well as the type of protocol used for communication. The purpose of ports is to uniquely identify different applications or processes running on a single computer and thereby enable them to share a single physical connection to a packet-switched network like the Internet.

By default, HTTP uses port 80 and HTTPS uses port 443, but a URL like `http://www.example.com:8080/path/` specifies that the web resource be served by the HTTP server on port 8080.

## INSTALL APACHE ON UBUNTU 22.04

You will need to have Apache installed in order to work through these steps. If you haven't already done so, you can get Apache installed on your server through the apt package manager:

```
$ sudo apt update
$ sudo apt install apache2
```

### I. SET UP NAME BASED VIRTUAL HOST

#### I.1. Create the directory structure :

The first step that you are going to take is to make a directory structure that will hold the site data that you will be serving to visitors.

Your document root (the top-level directory that Apache looks at to find content to serve) will be set to individual directories under the **/var/www** directory. You will create a directory here for both of the virtual hosts you plan on making.

Within each of these directories, you will create a **public\_html** folder that will hold your actual files. This gives you some flexibility in your hosting.

```
-virtual-machine:~/Desktop$ sudo mkdir -p /var/www/esprit1.com/public_html
```

```
-virtual-machine:~/Desktop$ sudo mkdir -p /var/www/esprit2.com/public_html
```

#### I.2. Grant permissions :

Now you have the directory structure for your files, but they are owned by your root user. If you want our regular user to be able to modify files in your web directories, you can change the ownership

```
-virtual-machine:~/Desktop$ sudo chown -R $USER:$USER /var/www/esprit1.com/public_html
```

```
-virtual-machine:~/Desktop$ sudo chown -R $USER:$USER /var/www/esprit2.com/public_html
```

You should also modify your permissions to ensure that read access is permitted to the general web directory and all of the files and folders it contains so that pages can be served correctly:

```
-virtual-machine:~/Desktop$ sudo chmod -R 755 /var/www
```

Your web server should now have the permissions it needs to serve content, and your user should be able to create content within the necessary folders.

### I.3. Create Demo Pages for each Virtual Host

For demonstration purposes, you'll make an **index.html** page for each site. Let's begin with **esprit1.com**. You can open up an index.html file in a text editor, in this case you'll use gedit:

```
-virtual-machine:~/Desktop$ sudo gedit /var/www/esprit1.com/public_html/index.html
```

Within this file, create an HTML document that indicates the site it is connected to, like the following:



```
index.html
/var/www/esprit1.com/public_html

1 <html>
2   <head>
3     <title>Welcome to Esprit1.com!</title>
4   </head>
5   <body>
6     <h1>Success! The esprit1.com virtual host is working!</h1>
7   </body>
8 </html>
```

Do the same thing for Esprit2 →

```
-virtual-machine:~/Desktop$ sudo gedit /var/www/esprit2.com/public_html/index.html
```



```
index.html
/var/www/esprit2.com/public_html

1 <html>
2   <head>
3     <title>Welcome to Esprit2.com!</title>
4   </head>
5   <body>
6     <h1>Success! The esprit2.com virtual host is working!</h1>
7   </body>
8 </html>
```

### I.4. Create new Virtual Host files

Virtual host files are the files that specify the actual configuration of your virtual hosts and dictate how the Apache web server will respond to various domain requests.

Apache comes with a default virtual host file called 000-default.conf that you can use as a jumping off point. You are going to copy it over to create a virtual host file for each of your domains.

You will start with one domain, configure it, copy it for our second domain, and then make the few further adjustments needed. The default Ubuntu configuration requires that each virtual host file end in .conf.

Start by copying the file for the first domain:

```
-virtual-machine:~/Desktop$ sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/esprit1.com.conf
```

Open the new file in your editor with root privileges:

```
-virtual-machine:~/Desktop$ sudo gedit /etc/apache2/sites-available/esprit1.com.conf
```

With comments removed, the file will look similar to this:

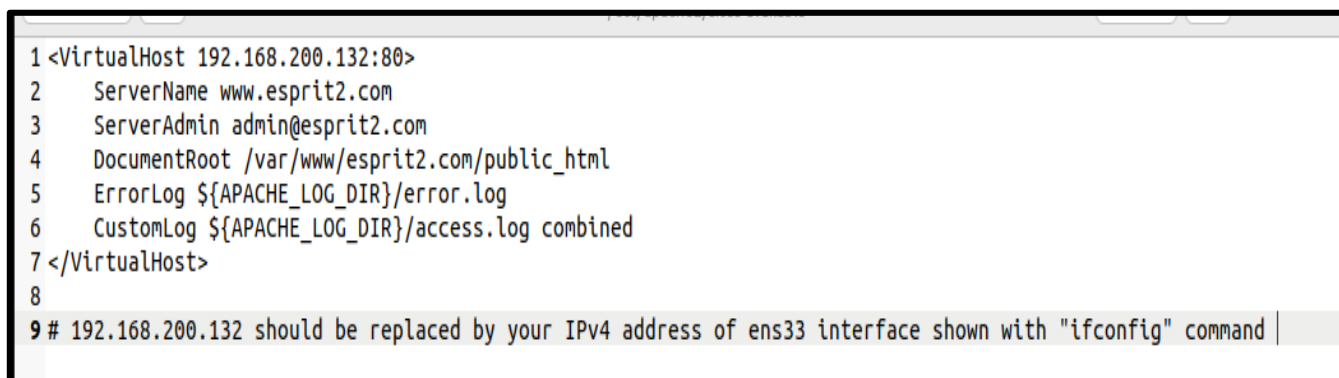


```
*esprit1.com.conf
/etc/apache2/sites-available

1 <VirtualHost 192.168.200.132:80>
2     ServerName www.esprit1.com
3     ServerAdmin admin@esprit1.com
4     DocumentRoot /var/www/esprit1.com/public_html
5     ErrorLog ${APACHE_LOG_DIR}/error.log
6     CustomLog ${APACHE_LOG_DIR}/access.log combined
7 </VirtualHost>
8
9 # 192.168.200.132 should be replaced by your IPv4 address of ens33 interface shown with "ifconfig" command
```

Do the same thing for Esprit2 →

```
-virtual-machine:~/Desktop$ sudo gedit /etc/apache2/sites-available/esprit2.com.conf
```



```
*esprit2.com.conf
/etc/apache2/sites-available

1 <VirtualHost 192.168.200.132:80>
2     ServerName www.esprit2.com
3     ServerAdmin admin@esprit2.com
4     DocumentRoot /var/www/esprit2.com/public_html
5     ErrorLog ${APACHE_LOG_DIR}/error.log
6     CustomLog ${APACHE_LOG_DIR}/access.log combined
7 </VirtualHost>
8
9 # 192.168.200.132 should be replaced by your IPv4 address of ens33 interface shown with "ifconfig" command
```

## **I.5. Enable the new Virtual Host Files :**

Now that you have created your virtual host files, you must enable them. Apache includes some tools that allow you to do this.

You'll be using the a2ensite tool to enable each of your sites.

```
-virtual-machine:~/Desktop$ sudo a2ensite esprit1.com.conf
```

```
-virtual-machine:~/Desktop$ sudo a2ensite esprit2.com.conf
```

Next, disable the default site defined in **000-default.conf**:

```
virtual-machine:~/Desktop$ sudo a2dissite 000-default.conf
```

When you are finished, you need to restart Apache to make these changes take effect to verify the success of the restart.

```
virtual-machine:~/Desktop$ systemctl reload apache2
```

## I.6. Set UP local hosts File

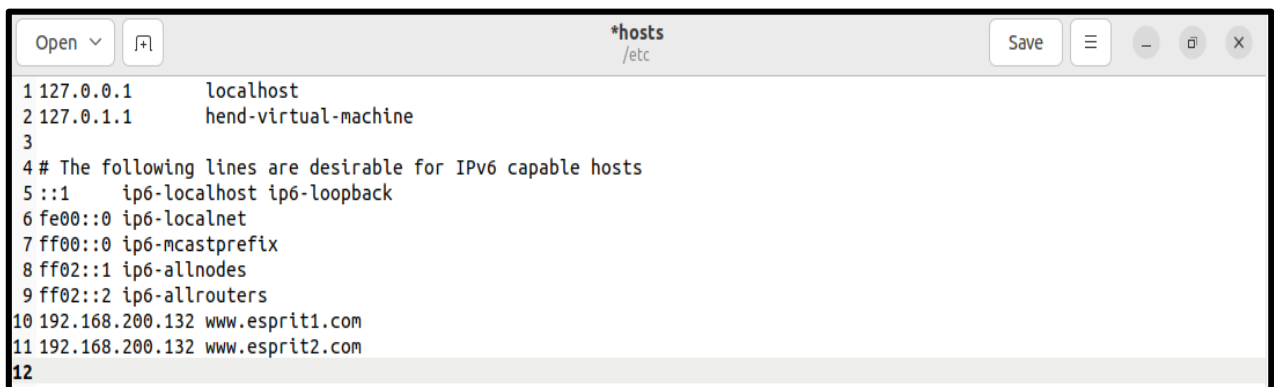
If you haven't been using actual domain names that you own to test this procedure and have been using some example domains instead, you can at least test the functionality of this process by temporarily modifying the hosts file on your local computer.

This will intercept any requests for the domains that you configured and point them to your VPS server, just as the DNS system would do if you were using registered domains. This will only work from your local computer though, and only for testing purposes.

```
virtual-machine:~/Desktop$ sudo gedit /etc/hosts
```

The details that you need to add are the IP address of your server followed by the domain you want to use to reach that server.

**Use your IPv4 address instead of <192.168.200.132>**

A screenshot of a text editor window titled '\*hosts /etc'. The window shows the contents of the /etc/hosts file. The text is as follows:

```
1 127.0.0.1    localhost
2 127.0.1.1    hend-virtual-machine
3
4 # The following lines are desirable for IPv6 capable hosts
5 ::1        ip6-localhost ip6-loopback
6 fe00::0    ip6-localnet
7 ff00::0    ip6-mcastprefix
8 ff02::1    ip6-allnodes
9 ff02::2    ip6-allrouters
10 192.168.200.132 www.esprit1.com
11 192.168.200.132 www.esprit2.com
12
```

This will direct any requests for **esprit1.com** and **esprit2.com** on our computer and send them to our server. This is what we want if we are not actually the owners of these domains in order to test our virtual hosts.

## I.7. Test your Results :

Now that you have your virtual hosts configured, you can test your setup by going to the domains that you configured in your web browser:



## II. SET UP IP BASED VIRTUAL HOST

To setup IP based virtual hosting, you must have more than one IP address assigned to your server or your Linux machine.

It can be on a single NIC card , For example: ens33:1, ens33:2, ens33:3 ... so forth. Multiple NIC cards can also be attached.

```
virtual-machine:~/Desktop$ sudo ifconfig ens33:1 192.168.200.133/24
```

Now, check that you have two IP Address with « **Ifconfig** » command

```
virtual-machine:~/Desktop$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.200.132  netmask 255.255.255.0  broadcast 192.168.200.255
    inet6 fe80::e583:700d:bd31:72dc  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:3b:c4:eb  txqueuelen 1000  (Ethernet)
    RX packets 462409  bytes 669704934 (669.7 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 49213  bytes 3098558 (3.0 MB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

ens33:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.200.133  netmask 255.255.255.0  broadcast 192.168.200.255
    ether 00:0c:29:3b:c4:eb  txqueuelen 1000  (Ethernet)

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 693  bytes 83482 (83.4 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 693  bytes 83482 (83.4 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

As you can see in above output, two IPs 192.168.200.132 (ens33) and 192.168.200.133 (ens33:1) is attached to the server, both IPs are assigned to the same physical network device (ens33).

Now, assign a specific IP to receive http requests, you can simply do it by changing the directive in **/etc/apache2/sites-available/esprit3.com.conf** file.



```
1 <VirtualHost 192.168.200.133:80>
2     ServerName www.esprit3.com
3     ServerAdmin admin@esprit3.com
4     DocumentRoot /var/www/esprit3.com/public_html
5     ErrorLog ${APACHE_LOG_DIR}/error.log
6     CustomLog ${APACHE_LOG_DIR}/access.log combined
7 </VirtualHost>
```

**Note :** don't forget to enable the new virtual machine, to create a new directory and the testing page (index.html) for the new web site www.esprit3.com  
Don't forget to add esprit3.com in the /etc/hosts file.

Check the result :



### III. SET UP PORT BASED VIRTUAL HOST

The default port number for HTTP is 80. However, most web servers can be configured to operate on almost any port number, provided the port number is not in use by any other program on the server.

For example, a server may host the website www.esprit2.com. However, if the owner wishes to operate a second site, and does not have access to the domain name configuration for their domain name, and/or owns no other IP addresses which could be used to serve the site from, they could instead use another port number, for example, www.esprit.com:8080 for port 8080.

Open Apache main configuration file with your chosen editor and add **listen 8080** parameter

```
-virtual-machine:~/Desktop$ sudo gedit /etc/apache2/ports.conf
```

Add the following virtual directive at the bottom of the file. Save and close the file.



In **/etc/apache2/sites-available/esprit4.com.conf** file, specify parameters like bellow :



```
1 <VirtualHost 192.168.200.132:8080>
2 ServerName www.esprit4.com
3 ServerAdmin admin@esprit4.com
4 DocumentRoot /var/www/esprit4.com/public_html
5 ErrorLog ${APACHE_LOG_DIR}/error.log
6 CustomLog ${APACHE_LOG_DIR}/access.log combined
7 </VirtualHost>
```

**Note :** don't forget to enable the new virtual machine, to create a new directory and the testing page (index.html) for the new web site [www.esprit4.com](http://www.esprit4.com)  
Don't forget to add esprit4.com in the /etc/hosts file.

Check the result :

