How to Configure Virtual Hosts in Apache on Ubuntu?

written by sysadmin | 16 June 2025

Virtual hosts are a feature on a web server, such as Apache or Nginx, to run more than one site on a server. By using this feature, you can easily configure multiple domains on a server and save on operational costs because you only need one server or a public IP. This article will explain how to configure virtual hosts in Apache on Ubuntu.

Problem

How to configure virtual hosts in Apache on Ubuntu?

Solution

Before starting the configuration, make sure that on the Ubuntu server, the Apache application is installed by using the command:

```
apt update
apt install -y apache2
```

To see the default settings of Apache in Ubuntu, type the command below:

```
sudo apache2ctl -S
```



Display default Apache configuration

2 types of virtual hosts can be used, name-based and IP-based, and the difference between the two can be seen in the image below:

Aspect	Name-Based	IP-Based
Definition	Uses the domain name (hostname) to distinguish between websites on the same IP.	Uses different IP addresses for each website hosted on the server.
How it Works	Server identifies the requested site by the hostname in the Host header.	Server identifies the site based on the destination IP address.
IP Requirements	Only one IP address is needed for multiple sites.	Each site requires its own unique IP address.
SSL Compatibility	May require SNI (Server Name Indication) to support multiple SSL certificates.	Easier SSL management, as each site can have its own SSL certificate.
Resource Efficiency	More efficient, as multiple sites share the same IP.	Less efficient, as each site requires its own IP.
Isolation	Sites share the same IP, so there is no isolation between them at the IP level.	Sites are isolated at the IP level, which can be beneficial for network management.
Performance	Slightly slower with SSL if SNI is not supported, otherwise similar.	No performance hit for SSL, as each site has its own IP.
Use Cases	Ideal for hosting many websites on the same server, especially for shared hosting.	Ideal for scenarios requiring multiple SSL certificates or when isolation is necessary.

Comparison of name-based and IP-based in virtual hosts

WARNING
This article uses a private IP, not a public IP.

A. name-based virtual hosts

The meaning of name-based is that you have many websites or domains, but you only have one IP. For example, you have 2 domain names: **website1.com** and **website2.com,** but you only have 1 IP, which is **192.168.56.100**. Here are the steps to get all three domains to use the same IP:

1. Create the directories and the files

By default, Apache uses the **/var/www/html** folder as its rootdocument, as shown in the image above. However, to make it easier to configure it, you should create a folder for each of these websites, as shown in the image below:

```
sudo mkdir -p /var/www/html/website1.com/
sudo mkdir -p /var/www/html/website2.com/
```

WARNING
You can change the above directory to another directory, but for the next steps, you have to follow the directory you created.

After that, create an index.html file for each domain:

```
sudo sh -c 'echo "<h1> This is for website1.com domain</h1>" >
/var/www/html/website1.com/index.html'
sudo sh -c 'echo "<h1> This is for website2.com domain</h1>" >
/var/www/html/website2.com/index.html'
```

2. Change ownership

Change the ownership of the folders:

```
sudo chown -R www-data:www-data /var/www/html/website1.com/
sudo chown -R www-data:www-data /var/www/html/website2.com/
sudo chmod -R 755 /var/www/html
```

3. Configuration of virtual hosts

By default, 2 directories are used to manage the many domains in the virtual hosts running on that server: the **sites-available** and **sites-enabled** directories located in the **/etc/apache2** directory. The sites-enabled directory contains all the configurations of the website (virtual host) that are available on the server but are not yet activated automatically. In contrast, the sites-enabled directory contains a symlink (symbolic link) to the configuration file that exists in the sites-available directory, and only the files that exist in the sites-enabled directory will be executed and activated by the web server if the web server is restarted or reloaded. Use the command below to create two websites on virtual hosts:

```
echo '<VirtualHost *:80>' | sudo tee /etc/apache2/sites-
available/website1.com.conf > /dev/null
```

```
echo '    ServerName website1.com' | sudo tee -a /etc/apache2/sites-
available/website1.com.conf > /dev/null
echo '    ServerAlias www.website1.com' | sudo tee -a /etc/apache2/sites-
available/website1.com.conf > /dev/null
echo '    ServerAdmin webmaster@website1.com' | sudo tee -a
/etc/apache2/sites-available/website1.com.conf > /dev/null
echo '    DocumentRoot /var/www/html/website1.com' | sudo tee -a
/etc/apache2/sites-available/website1.com.conf > /dev/null
echo '    ErrorLog ${APACHE_LOG_DIR}/website1-error.log' | sudo tee -a
/etc/apache2/sites-available/website1.com.conf > /dev/null
echo '    CustomLog ${APACHE_LOG_DIR}/website1-access.log combined' | sudo
tee -a /etc/apache2/sites-available/website1.com.conf > /dev/null
echo '</VirtualHost>' | sudo tee -a /etc/apache2/sites-
available/website1.com.conf > /dev/null

echo '<VirtualHost *:80>' | sudo tee /etc/apache2/sites-
available/website2.com.conf > /dev/null
echo '    ServerName website2.com' | sudo tee -a /etc/apache2/sites-
available/website2.com.conf > /dev/null
echo '    ServerAlias www.website2.com' | sudo tee -a /etc/apache2/sites-
available/website2.com.conf > /dev/null
echo '    ServerAdmin webmaster@website2.com' | sudo tee -a
/etc/apache2/sites-available/website2.com.conf > /dev/null
echo '    DocumentRoot /var/www/html/website2.com' | sudo tee -a
/etc/apache2/sites-available/website2.com.conf > /dev/null
echo '    ErrorLog ${APACHE_LOG_DIR}/website2-error.log' | sudo tee -a
/etc/apache2/sites-available/website2.com.conf > /dev/null
echo '    CustomLog ${APACHE_LOG_DIR}/website2-access.log combined' | sudo
tee -a /etc/apache2/sites-available/website2.com.conf > /dev/null
echo '</VirtualHost>' | sudo tee -a /etc/apache2/sites-
available/website2.com.conf > /dev/null
```

WARNING

You can change ***:80** to your IP server like **192.168.56.102:80**.

Then type the command below to enable the Virtual Hosts configuration:

```
sudo a2ensite website1.com.conf
sudo a2ensite website2.com.conf
```

Type the command below to disable the default virtual hosts configuration:

```
sudo a2dissite 000-default.conf
```

WARNING

If you want to change the configuration of virtual hosts, you have to **change it in the sites-available directory** and not in the sites-enabled directory.

5. Check the configuration

Use the command below to check whether there is an Apache configuration that is an error or not by using the command below:

```
sudo apache2ctl configtest
```

If there is no error, then reload Apache using the command below:

```
sudo systemctl reload apache2
```

WARNING

Use the command above if there is a change in the configuration of virtual hosts in each domain.

6. Check in the browser

Because this article uses a private IP, you must configure it in the hosts file before you check the browser. If you use Windows, change the hosts file in **C:\Windows\System32\drivers\etc\hosts** or in **/etc/hosts** if you use Linux. In the hosts file, add the below script:

```
192.168.56.102  website1.com website2.com
```

Info

Change IP 192.168.56.102 with your Ubuntu IP server.

If your Ubuntu server uses a firewall, type the command below to open the port for Apache:

```
sudo ufw allow 'Apache Full'
```

Open your browser and type each of these domains, then there should be a site displayed as in the image below:

```
http://website1.com
```



Site website1.com

```
http://website2.com
```



site website2.com

If you use Linux, you can use the command below to check the result:

```
curl http://website1.com
curl http://website2.com
```

```
sysadmin@lubuntu:~$ cat /etc/hosts
# Standard host addresses
127.0.0.1  localhost
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
# This host address
127.0.1.1  lubuntu
192.168.56.102 site1.com
192.168.56.103 site2.com
sysadmin@lubuntu:~$
sysadmin@lubuntu:~$ curl http://site1.com
<h1> This is for site1.com domain</h1>
sysadmin@lubuntu:~$
sysadmin@lubuntu:~$ curl http://site2.com
<h1> This is for site2.com domain</h1>
sysadmin@lubuntu:~$
```

Using the curl command

By default, websites work on the web server using port 80. But you can change port 80 to another port as long as the port is not used on the server. For example, if you want the website1.com site to use port **8080**, change the **/etc/apache2/sites-available/website1.com.conf** file and change its contents to something like this:

```
Listen 8080
<VirtualHost *:8080>
     ServerName website1.com
     ServerAlias www.website1.com
     ServerAdmin webmaster@website1.com
     DocumentRoot /var/www/html/website1.com
     ErrorLog ${APACHE_LOG_DIR}/website1-error.log
     CustomLog ${APACHE_LOG_DIR}/website1-access.log combined
</VirtualHost>
```

If you use the firewall in the Ubuntu server, don’t forget to open port 8080 using the command below:

```
sudo ufw allow 8080
```

Reload Apache and open it in the browser by typing the command:

http://website1.com:8080



Site website1.com:8080

B. IP-based virtual hosts

The meaning of IP-based is that you use a different IP address for each website. For example, you have 2 IPs and 2 domains, where IP **192.168.56.102 is for site1.com,** and IP **192.168.56.103 is for site2.com**. This article will use a server that has 2 IPs, as shown below:

```
sysadmin@Ubuntu2404:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:29:a3:f1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.103/24 metric 100 brd 192.168.56.255 scope global dynamic enp0s3
       valid_lft 573sec preferred_lft 573sec
    inet6 fe80::a00:27ff:fe29:a3f1/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:83:09:85 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 metric 100 brd 192.168.56.255 scope global dynamic enp0s8
       valid_lft 565sec preferred_lft 565sec
    inet6 fe80::a00:27ff:fe83:985/64 scope link
       valid_lft forever preferred_lft forever
sysadmin@Ubuntu2404:~$
```

Using 2 NICs in a server

1. Create the directories and the files

By default, Apache uses the **/var/www/html** folder as its rootdocument, as shown in the image above. However, to make it easier to configure it, you should create a folder for each of these websites, as shown in the image below:

```
sudo mkdir -p /var/www/html/site1.com/
sudo mkdir -p /var/www/html/site2.com/
```

WARNING
You can change the above directory to another directory, but for the next steps, you have to follow the directory you created.

After that, create an index.html file for each domain:

```
sudo sh -c 'echo "<h1> This is for site1.com domain</h1>" >
/var/www/html/site1.com/index.html'
sudo sh -c 'echo "<h1> This is for site2.com domain</h1>" >
/var/www/html/site2.com/index.html'
```

2. Change ownership

Change the ownership of the folders:

```
sudo chown -R www-data:www-data /var/www/html/site1.com/
sudo chown -R www-data:www-data /var/www/html/site2.com/
sudo chmod -R 755 /var/www/html
```

3. Configuration of virtual hosts

By default, 2 directories are used to manage the many domains in the virtual hosts running on that server: the **sites-available** and **sites-enabled** directories located in the **/etc/apache2** directory. The sites-enabled directory contains all the configuration of the website (virtual host) that is available on the server, but is not yet activated automatically while the sites-enabled directory contains a symlink (symbolic link) to the configuration file that exists in the sites-available directory and only the files that exist in the site-enabled directory will be executed and activated by the web server if the webserver is restarted or reloaded. Use the command below to create a virtual hosts directory:

```
echo '<VirtualHost 192.168.56.102:80>' | sudo tee /etc/apache2/sites-
available/site1.com.conf > /dev/null
echo '    ServerName site1.com' | sudo tee -a /etc/apache2/sites-
available/site1.com.conf > /dev/null
echo '    ServerAlias www.site1.com' | sudo tee -a /etc/apache2/sites-
```

```
available/site1.com.conf > /dev/null
echo '    ServerAdmin webmaster@site1.com' | sudo tee -a /etc/apache2/sites-
available/site1.com.conf > /dev/null
echo '    DocumentRoot /var/www/html/site1.com' | sudo tee -a
/etc/apache2/sites-available/site1.com.conf > /dev/null
echo '    ErrorLog ${APACHE_LOG_DIR}/site1-error.log' | sudo tee -a
/etc/apache2/sites-available/site1.com.conf > /dev/null
echo '    CustomLog ${APACHE_LOG_DIR}/site1-access.log combined' | sudo tee -
a /etc/apache2/sites-available/site1.com.conf > /dev/null
echo '</VirtualHost>' | sudo tee -a /etc/apache2/sites-
available/site1.com.conf > /dev/null

echo '<VirtualHost 192.168.56.103:80>' | sudo tee /etc/apache2/sites-
available/site2.com.conf > /dev/null
echo '    ServerName site2.com' | sudo tee -a /etc/apache2/sites-
available/site2.com.conf > /dev/null
echo '    ServerAlias www.site2.com' | sudo tee -a /etc/apache2/sites-
available/site2.com.conf > /dev/null
echo '    ServerAdmin webmaster@site2.com' | sudo tee -a /etc/apache2/sites-
available/site2.com.conf > /dev/null
echo '    DocumentRoot /var/www/html/site2.com' | sudo tee -a
/etc/apache2/sites-available/site2.com.conf > /dev/null
echo '    ErrorLog ${APACHE_LOG_DIR}/site2-error.log' | sudo tee -a
/etc/apache2/sites-available/site2.com.conf > /dev/null
echo '    CustomLog ${APACHE_LOG_DIR}/site2-access.log combined' | sudo tee -
a /etc/apache2/sites-available/site2.com.conf > /dev/null
echo '</VirtualHost>' | sudo tee -a /etc/apache2/sites-
available/site2.com.conf > /dev/null
```

WARNING

If you want to change the configuration of virtual hosts, you have to **change it in the sites-available directory** and not in the sites-enabled directory.

Then type the command below to enable the Virtual Hosts configuration:

```
sudo a2ensite site1.com.conf
sudo a2ensite site2.com.conf
```

Type the command below to disable the default virtual hosts configuration:

```
sudo a2dissite 000-default.conf
```

5. Check the configuration

Use the command below to check whether there is an Apache configuration that is an error or not by using the command below:

```
sudo apache2ctl configtest
```

If there is no error, then reload Apache using the command below:

```
sudo systemctl reload apache2
```

WARNING
Use the command above if there is a change in the configuration of virtual hosts in each domain.

6. Check in the browser

Because this article uses a private IP, you must configure it in the hosts file before you check the browser. If you use Windows, change the hosts file in **C:\Windows\System32\drivers\etc\hosts** or in **/etc/hosts** if you use Linux. In the hosts file, add the below script:

```
192.168.56.102  site1.com
192.168.56.103   site2.com
```

Info
Change IP 192.168.56.102 and IP 192.168.56.103 with your Ubuntu IP server.

If your Ubuntu server uses a firewall, type the command below to open the port for Apache:

```
sudo ufw allow 'Apache Full'
```

Open your browser and type each of these domains, then there should be a site displayed as in the image below:

http://site1.com



Site site1.com

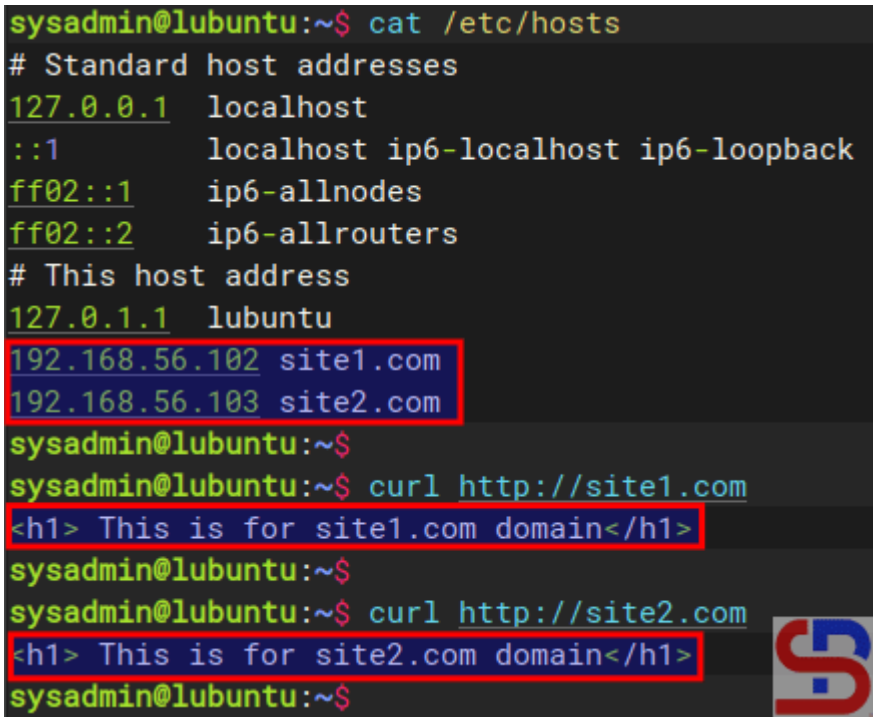
http://site2.com



Site site2.com

If you use Linux, you can use the command below to check the result:

```
curl http://site1.com
curl http://site2.com
```



Using the curl command

By default, websites work on the web server using port 80. But you can change port 80 to another port as long as the port is not used on the server. So, if you want the site1.com site to use port **8181**, change the **/etc/apache2/sites-available/site1.com.conf** file and change its contents to something like this:

```
Listen 8181
<VirtualHost 192.168.56.102:8181>
     ServerName site1.com
     ServerAlias www.site1.com
     ServerAdmin webmaster@site1.com
     DocumentRoot /var/www/html/site1.com
     ErrorLog ${APACHE_LOG_DIR}/site1-error.log
     CustomLog ${APACHE_LOG_DIR}/site1-access.log combined
</VirtualHost>
```

If you use the firewall in your Ubuntu server, don’t forget to open port 8181 using the command below:

```
sudo ufw allow 8181
```

Reload Apache and open it in the browser by typing the command:

```
http://site1.com:8181
```




```
Site site1.com:8181
```

Note

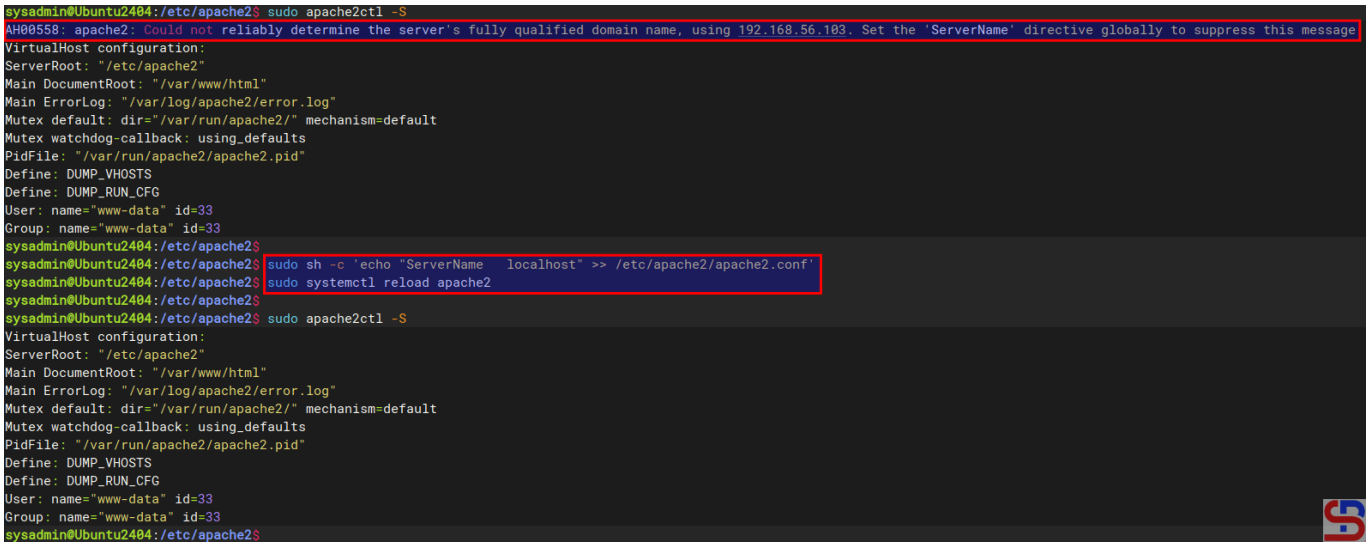
If you want to remove the error like this:

```
AH00558: apache2: Could not reliably determine the server’s fully qualified domain name, using 192.168.56.103. Set the ‘ServerName’ directive globally to suppress this message
```

Go to the **/etc/apache2/apache2.conf** and insert the script below:

```
ServerName localhost
```

Reload the Apache, and the error will disappear, like in the image below:



Remove error AH00558

WARNING
You can change the localhost to your domain name, like website1.com or another domain name.

References

httpd.apache.org
phoenixnap.com
medium.com
digitalocean.com
stackoverflow.com
serverfault.com
baeldung.com
askubuntu.com