

How to Configure Virtual Hosts in Apache on RockyLinux?

written by sysadmin | 18 June 2025

[The previous article](#) explained how to create virtual hosts in Ubuntu. This article will explain how to configure virtual hosts in Apache on Rocky Linux or derivatives of RHEL, such as AlmaLinux, CentOS, and so on.

Problem

How to configure virtual hosts in Apache on RockyLinux?

Solution

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

```
yum install -y httpd
```

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

```
sudo httpd -S
```

```
[sysadmin@RockyLinux9 ~]$ sudo httpd -S
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::a00:27ff:fe17:8fa9%enp0s3.
Set the 'ServerName' directive globally to suppress this message
VirtualHost configuration:
ServerRoot: "/etc/httpd"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/etc/httpd/logs/error_log"
Mutex default: dir="/etc/httpd/run/" mechanism=default
Mutex cache-socache: using_defaults
Mutex authdigest-opaque: using_defaults
Mutex watchdog-callback: using_defaults
Mutex proxy-balancer-shm: using_defaults
Mutex rewrite-map: using_defaults
Mutex authdigest-client: using_defaults
Mutex dav_fs-lockdb: using_defaults
Mutex lua-ivm-shm: using_defaults
Mutex proxy: using_defaults
Mutex authn-socache: using_defaults
PidFile: "/etc/httpd/run/httpd.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="apache" id=48
Group: name="apache" id=48
[sysadmin@RockyLinux9 ~]$
```

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

3. Configuration of virtual hosts

Unlike Ubuntu and its derivatives, which use the `sites-available` and `sites-enabled` folders in configuring virtual hosts, by default, RockyLinux and its derivatives do not use both folders, but the virtual hosts configuration is inserted into the `/etc/httpd/conf.d/` folder. Therefore, type the command below to create two domains in the virtual hosts:

```
echo "<VirtualHost *:80>" | sudo tee /etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    ServerName website1.com" | sudo tee -a /etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    ServerAlias www.website1.com" | sudo tee -a /etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    ServerAdmin webmaster@website1.com" | sudo tee -a /etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    DocumentRoot /var/www/html/website1.com" | sudo tee -a /etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    ErrorLog logs/website1-error.log" | sudo tee -a /etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    CustomLog logs/website1-access.log combined" | sudo tee -a /etc/httpd/conf.d/website1.com.conf > /dev/null
echo "</VirtualHost>" | sudo tee -a /etc/httpd/conf.d/website1.com.conf > /dev/null

echo "<VirtualHost *:80>" | sudo tee /etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    ServerName website2.com" | sudo tee -a /etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    ServerAlias www.website2.com" | sudo tee -a /etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    ServerAdmin webmaster@website2.com" | sudo tee -a /etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    DocumentRoot /var/www/html/website2.com" | sudo tee -a /etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    ErrorLog logs/website2-error.log" | sudo tee -a /etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    CustomLog logs/website2-access.log combined" | sudo tee -a /etc/httpd/conf.d/website2.com.conf > /dev/null
echo "</VirtualHost>" | sudo tee -a /etc/httpd/conf.d/website2.com.conf > /dev/null
```

WARNING

You can change `*:80` to your IP server like **192.168.56.2:80**.

4. 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 apachectl configtest
```

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

```
sudo systemctl reload httpd
```

WARNING

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

5. 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:

```
192.168.56.2  website1.com website2.com
```

Info

Change IP 192.168.56.2 with your RockyLinux IP server.

By default, Rockylinux activates the firewall, so you have to open the HTTP port using the command below:

```
firewall-cmd --add-service=http --permanent  
firewall-cmd --reload
```

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

`http://website1.com`



This is for website1.com domain



Site website1.com

`http://website2.com`



This is for website2.com domain



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
192.168.56.2 website1.com website2.com
sysadmin@lubuntu:~$
sysadmin@lubuntu:~$ curl http://website1.com
<h1> This is for website1.com domain</h1>
sysadmin@lubuntu:~$
sysadmin@lubuntu:~$ curl http://website2.com
<h1> This is for website2.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/httpd/conf.d/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 logs/website1-error.log
    CustomLog logs/website1-access.log combined
</VirtualHost>

```

Don't forget to open the 8080 port on the Rockylinux server and reload Apache using the command below:

```

firewall-cmd --add-port=8080/tcp --permanent
firewall-cmd --reload
sudo systemctl reload apache

```

Open your browser and type the command below:

http://website1.com:8080



This is for website1.com domain



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 **192.168.56.2** is for **site1.com**, and IP **192.168.56.104** is for **site2.com**. This article will use a server that has 2 IPs, as shown below:

```
[root@RockyLinux9 sysconfig]# 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
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:17:8f:a9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.104/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s3
        valid_lft 576sec preferred_lft 576sec
    inet6 fe80::a00:27ff:fe17:8fa9/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:38:ad:88 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.2/24 brd 192.168.56.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe38:ad88/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@RockyLinux9 sysconfig]#
```

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

3. Configuration of virtual hosts

Unlike Ubuntu and its derivatives, which use the sites-available and sites-enabled folders in configuring virtual hosts, by default, RockyLinux and its derivatives do not use both folders, but the virtual hosts configuration is inserted into the **/etc/httpd/conf.d/** folder. Therefore, type the command below to create two domains in the virtual hosts:

```
echo "<VirtualHost 192.168.56.2:80>" | sudo tee
/etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    ServerName website1.com" | sudo tee -a
/etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    ServerAlias www.website1.com" | sudo tee -a
/etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    ServerAdmin webmaster@website1.com" | sudo tee -a
/etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    DocumentRoot /var/www/html/website1.com" | sudo tee -a
/etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    ErrorLog logs/website1-error.log" | sudo tee -a
/etc/httpd/conf.d/website1.com.conf > /dev/null
echo "    CustomLog logs/website1-access.log combined" | sudo tee -a
```

```
/etc/httpd/conf.d/website1.com.conf > /dev/null
echo "</VirtualHost>" | sudo tee -a /etc/httpd/conf.d/website1.com.conf >
/dev/null

echo "<VirtualHost 192.168.56.104:80>" | sudo tee
/etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    ServerName website2.com" | sudo tee -a
/etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    ServerAlias www.website2.com" | sudo tee -a
/etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    ServerAdmin webmaster@website2.com" | sudo tee -a
/etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    DocumentRoot /var/www/html/website2.com" | sudo tee -a
/etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    ErrorLog logs/website2-error.log" | sudo tee -a
/etc/httpd/conf.d/website2.com.conf > /dev/null
echo "    CustomLog logs/website2-access.log combined" | sudo tee -a
/etc/httpd/conf.d/website2.com.conf > /dev/null
echo "</VirtualHost>" | sudo tee -a /etc/httpd/conf.d/website2.com.conf >
/dev/null
```

4. 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 apachectl configtest
```

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

```
sudo systemctl reload httpd
```

WARNING

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

5. 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:

```
192.168.56.2    site1.com
192.168.56.104 site2.com
```

Info

Change IP 192.168.56.2 & IP 192.168.56.104 with your RockyLinux IP server.

By default, Rockylinux activates the firewall, so you have
to open the HTTP port using the command below:

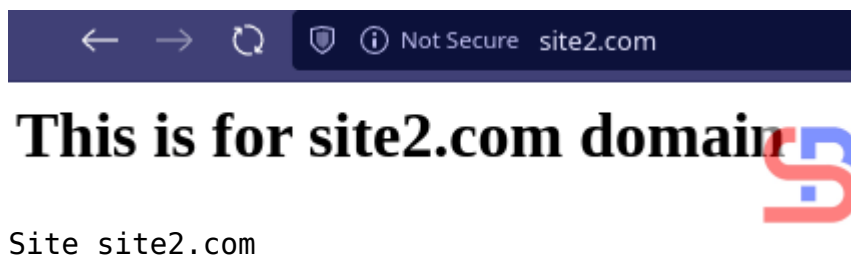
```
firewall-cmd --add-service=http --permanent
firewall-cmd --reload
```

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

<http://site1.com>



<http://site2.com>



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

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

```
sysadmin@ubuntu:~$ 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 ubuntu
192.168.56.2 site1.com
192.168.56.104 site2.com
sysadmin@ubuntu:~$
sysadmin@ubuntu:~$ curl http://site1.com
<h1> This is for website1.com domain</h1>
sysadmin@ubuntu:~$
sysadmin@ubuntu:~$ curl http://site2.com
<h1> This is for site2.com domain</h1>
sysadmin@ubuntu:~$
```



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/httpd/conf.d/website1.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 logs/site1-error.log CustomLog logs/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 firewall-cmd --add-port=8181/tcp --permanent
sudo firewall-cmd --reload
sudo systemctl reload apache
```

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

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



Note

If you want to remove the error like this:

AH00558: httpd: 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/httpd/conf/httpd.conf` and insert the script below:

```
ServerName localhost
```

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

```

[sysadmin@RockyLinux9 ~]$ sudo httpd -S
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::a00:27ff:fe17:8fa9%enp0s3. Set the 'ServerName' directive globally to suppress this message
VirtualHost configuration:
ServerRoot: "/etc/httpd"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/etc/httpd/logs/error_log"
Mutex authdigest-opaque: using_defaults
Mutex watchdog-callback: using_defaults
Mutex proxy-balancer-shm: using_defaults
Mutex rewrite-map: using_defaults
Mutex authdigest-client: using_defaults
Mutex dav_fs-lockdb: using_defaults
Mutex lua-ivm-shm: using_defaults
Mutex proxy: using_defaults
Mutex authn-socache: using_defaults
Mutex default: dir="/etc/httpd/run/" mechanism=default
Mutex cache-socache: using_defaults
PidFile: "/etc/httpd/run/httpd.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="apache" id=48
Group: name="apache" id=48
[sysadmin@RockyLinux9 ~]$
[sysadmin@RockyLinux9 ~]$ sudo sh -c 'echo "ServerName localhost" >> /etc/httpd/conf/httpd.conf'
[sysadmin@RockyLinux9 ~]$ sudo systemctl reload httpd
[sysadmin@RockyLinux9 ~]$
[sysadmin@RockyLinux9 ~]$ sudo httpd -S
VirtualHost configuration:
ServerRoot: "/etc/httpd"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/etc/httpd/logs/error_log"
Mutex dav_fs-lockdb: using_defaults
Mutex lua-ivm-shm: using_defaults
Mutex proxy: using_defaults
Mutex authn-socache: using_defaults
Mutex default: dir="/etc/httpd/run/" mechanism=default
Mutex cache-socache: using_defaults
Mutex authdigest-opaque: using_defaults
Mutex watchdog-callback: using_defaults
Mutex proxy-balancer-shm: using_defaults
Mutex rewrite-map: using_defaults
Mutex authdigest-client: using_defaults
PidFile: "/etc/httpd/run/httpd.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="apache" id=48
Group: name="apache" id=48
[sysadmin@RockyLinux9 ~]$

```

Remove error AH00558

References

sandeepbansod.medium.com
reintech.io
tecmint.com
httpd.apache.org
askubuntu.com
dewaweb.com
arubacloud.com