

# How to Configure Firewalld to be Port Forwarding?

written by sysadmin | 21 June 2025

Port forwarding is a networking technique used to redirect communication requests from one port number to another port number, typically across a network boundary such as a router or firewall. This technique can be used with Firewalld, available in RockyLinux, or derivative distros from RHEL such as AlmaLinux, CentOS, and others.

## Problem

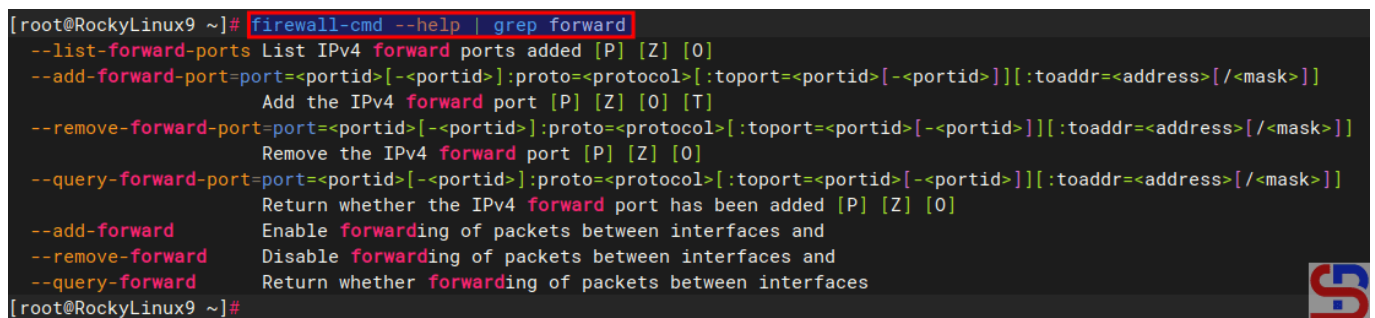
How to configure Firewalld to be port forwarding?

## Solution

If you want to see the command in firewalls to run port forwarding, type the below command:

```
firewall-cmd --help | grep forward
```

```
[root@RockyLinux9 ~]# firewall-cmd --help | grep forward
--list-forward-ports List IPv4 forward ports added [P] [Z] [0]
--add-forward-port=port=<portid>[-<portid>]:proto=<protocol>[:toport=<portid>[-<portid>]][:toaddr=<address>[/<mask>]]
    Add the IPv4 forward port [P] [Z] [0] [T]
--remove-forward-port=port=<portid>[-<portid>]:proto=<protocol>[:toport=<portid>[-<portid>]][:toaddr=<address>[/<mask>]]
    Remove the IPv4 forward port [P] [Z] [0]
--query-forward-port=port=<portid>[-<portid>]:proto=<protocol>[:toport=<portid>[-<portid>]][:toaddr=<address>[/<mask>]]
    Return whether the IPv4 forward port has been added [P] [Z] [0]
--add-forward      Enable forwarding of packets between interfaces and
--remove-forward   Disable forwarding of packets between interfaces and
--query-forward    Return whether forwarding of packets between interfaces
```



The commands in firewalld for port forwarding

There are 2 methods of port forwarding: forward the connection of a port to one IP/device and forward the connection of a port to a different IP/device.

### A. Forward to the same IP/device

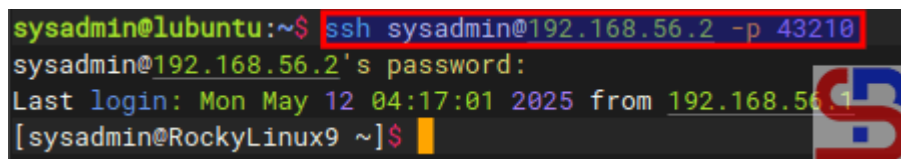
By default, you must use the format below to forward a port

in a device:

```
firewall-cmd --add-forward-port=port=port-  
number:proto=tcp|udp|sctp|dccp:toport=port-number
```

You can add an option **--permanent** if you want the rule to remain after reloading or rebooting the system. For example, you have a server with IP 192.168.56.2 where port 22 on the server is closed so to access the server via SSH must use port 43210. If you follow [this article](#), then you must type the command below to access the server:

```
ssh sysadmin@192.168.56.2 -p 43210
```

A terminal window showing an SSH session. The prompt is 'sysadmin@lubuntu:~\$' and the command 'ssh sysadmin@192.168.56.2 -p 43210' is entered. The output shows 'sysadmin@192.168.56.2's password:', 'Last login: Mon May 12 04:17:01 2025 from 192.168.56.1', and the prompt '[sysadmin@RockyLinux9 ~]\$' with a cursor. A red box highlights the command. A logo is visible on the right side of the terminal output.

```
sysadmin@lubuntu:~$ ssh sysadmin@192.168.56.2 -p 43210  
sysadmin@192.168.56.2's password:  
Last login: Mon May 12 04:17:01 2025 from 192.168.56.1  
[sysadmin@RockyLinux9 ~]$
```

Access the server via SSH using the port

However, by implementing a port forwarding you can access the server without typing the port. Let's say, the firewall is in the device, then on the device open port 43210 using the command:

```
sudo firewall-cmd --add-port=43210/tcp --permanent  
sudo firewall-cmd --reload
```

In the file **/etc/sshd/sshd\_config**, change the port to be as below:

```
Port 43210
```

After that restart SSH by using the command:

```
sudo systemctl restart sshd
```

After that, type the commands below to configure the forwarding port in the firewall:

```
firewall-cmd --add-masquerade --permanent
firewall-cmd --add-forward-port=port=22:proto=tcp:toport=43210 --permanent
firewall-cmd --reload
firewall-cmd --list-all
```

```
[root@RockyLinux9 ~]# firewall-cmd --add-masquerade --permanent
success
[root@RockyLinux9 ~]# firewall-cmd --add-forward-port=port=22:proto=tcp:toport=43210 --permanent
success
[root@RockyLinux9 ~]# firewall-cmd --reload
success
[root@RockyLinux9 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: cockpit dhcpv6-client http ssh
  ports: 80/tcp 43210/tcp
  protocols:
  forward: yes
  masquerade: yes
  forward-ports:
    port=22:proto=tcp:toport=43210:toaddr=
  source-ports:
  icmp-blocks:
  rich rules:
[root@RockyLinux9 ~]#
```

The commands to configure firewalld to be port forwarding

type the command below to access the server via SSH:

```
ssh sysadmin@192.168.56.2
```

You should be able to enter the server without having to type the 43210 port as shown below:

```
sysadmin@ubuntu:~$ ssh sysadmin@192.168.56.2
sysadmin@192.168.56.2's password:
Last login: Mon May 12 04:21:13 2025 from 192.168.56.1
[sysadmin@RockyLinux9 ~]$
```

Access the server via SSH without writing the port

## B. Forward to a different IP/device

By default, use the format below to forward a port to a different IP/device:

```
firewall-cmd --add-forward-port=port=port-  
number:proto=tcp|udp|sctp|dccp:toport=port-number:toaddr=ip_address
```

If you want the rule to stay in place after a system reboot or reload, you can add a **--permanent** option. As an illustration, suppose you have a server with IP address 192.168.56.2 and port 22 is available. You would like users who access port 22 to forward to port 22 with IP address 192.168.56.102. Use the command below to configure firewalls:

```
firewall-cmd --add-masquerade --permanent  
sudo firewall-cmd --add-forward-  
port=port=22:proto=tcp:toport=22:toaddr=192.168.56.102 --permanent  
firewall-cmd --reload  
firewall-cmd --list-all
```

```
[sysadmin@RockyLinux9 ~]$ sudo firewall-cmd --add-masquerade --permanent  
success  
[sysadmin@RockyLinux9 ~]$ sudo firewall-cmd --add-forward-port=port=22:proto=tcp:toport=22:toaddr=192.168.56.102 --permanent  
success  
[sysadmin@RockyLinux9 ~]$ sudo firewall-cmd --reload  
success  
[sysadmin@RockyLinux9 ~]$ sudo firewall-cmd --list-all  
public (active)  
target: default  
icmp-block-inversion: no  
interfaces: enp0s3 enp0s8  
sources:  
services: cockpit dhcpv6-client http ssh  
ports: 80/tcp 43210/tcp  
protocols:  
forward: yes  
masquerade: yes  
forward-ports:  
  port=22:proto=tcp:toport=22:toaddr=192.168.56.102  
source-ports:  
icmp-blocks:  
rich rules:  
[sysadmin@RockyLinux9 ~]$
```

Add a forwarding port to a different IP in firewallld

If you type the command below:

```
ssh sysadmin@192.168.56.2
```

You will be forwarded to a server that uses IP 192.168.56.102 as shown below:

```
sysadmin@ubuntu:~$ ssh sysadmin@192.168.56.2
sysadmin@192.168.56.2's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Mon May 12 15:21:13 2025 from 192.168.56.2
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 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 53225sec preferred_lft 53225sec
    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 420sec preferred_lft 420sec
    inet6 fe80::a00:27ff:fe83:985/64 scope link
        valid_lft forever preferred_lft forever
sysadmin@Ubuntu2404:~$
```

Forward a port to another IP/device

## Note

To see rule forwarding is in the rule in the firewall, besides being able to use the **firewall-cmd --list-all** command, you can also use the command below:

```
sudo firewall-cmd --list-forward-ports
```

then you will see the results as shown below:

```
[sysadmin@RockyLinux9 ~]$ sudo firewall-cmd --list-forward-ports
port=22:proto=tcp:toport=22:toaddr=192.168.56.102
[sysadmin@RockyLinux9 ~]$
```

Using **--list-forward-ports** option

And if you want to delete a rule port forwarding in the firewall, then you can simply change the options **--add-forward-port** to **--remove-forward-port** so the command will

change like in the command below:

```
sudo firewall-cmd --add-forward-  
port=port=22:proto=tcp:toport=22:toaddr=192.168.56.102 --permanent
```

```
[sysadmin@RockyLinux9 ~]$ sudo firewall-cmd --list-forward-ports  
port=22:proto=tcp:toport=22:toaddr=192.168.56.102  
[sysadmin@RockyLinux9 ~]$  
[sysadmin@RockyLinux9 ~]$ sudo firewall-cmd --remove-forward-port=port=22:proto=tcp:toport=22:toaddr=192.168.56.102 --permanent  
success  
[sysadmin@RockyLinux9 ~]$ sudo firewall-cmd --reload  
success  
[sysadmin@RockyLinux9 ~]$ sudo firewall-cmd --list-forward-ports  
[sysadmin@RockyLinux9 ~]$  
[sysadmin@RockyLinux9 ~]$
```

Remove a forwarding port rule

## References

[docs.redhat.com](https://docs.redhat.com)

[youtube.com](https://youtube.com)

[musaamin.web.id](https://musaamin.web.id)

[faun.pub](https://faun.pub)

---

## [How to Configure Virtual Hosts in Apache on RockyLinux?](#)

written by sysadmin | 21 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
```



## 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
```

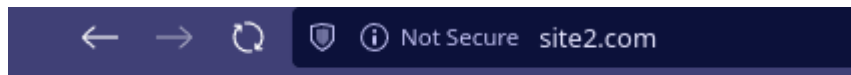


**This is for site1.com domain**



Site site1.com

```
http://site2.com
```



**This is for site2.com domain**



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
```

```

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 site1.com
192.168.56.104 site2.com
sysadmin@lubuntu:~$
sysadmin@lubuntu:~$ curl http://site1.com
<h1> This is for website1.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. 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 ~]$ sudo sh -c 'echo "ServerName localhost" >> /etc/httpd/conf/httpd.conf'
[sysadmin@RockyLinux9 ~]$ sudo systemctl reload httpd
[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](https://sandeepbansod.medium.com)
- [reintech.io](https://reintech.io)
- [tecmint.com](https://tecmint.com)
- [httpd.apache.org](https://httpd.apache.org)
- [askubuntu.com](https://askubuntu.com)
- [dewaweb.com](https://dewaweb.com)
- [arubacloud.com](https://arubacloud.com)

---

# [How to Configure Virtual Hosts in Apache on Ubuntu?](#)

written by sysadmin | 21 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?

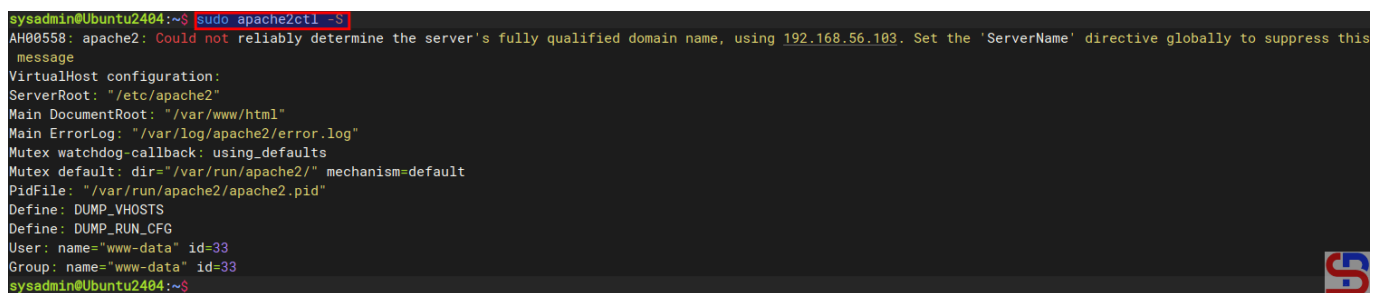
## Solution

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
```



```
sysadmin@Ubuntu2404:~$ sudo apache2ctl -S
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
VirtualHost configuration:
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex watchdog-callback: using_defaults
Mutex default: dir="/var/run/apache2/" mechanism=default
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="www-data" id=33
Group: name="www-data" id=33
sysadmin@Ubuntu2404:~$
```

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`



**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@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   lubuntu
192.168.56.102 site1.com
192.168.56.103 site2.com
sysadmin@ubuntu:~$
sysadmin@ubuntu:~$ curl http://site1.com
<h1> This is for site1.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. 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



## 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 **192.168.56.102** 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



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
192.168.56.102 site1.com
192.168.56.103 site2.com
sysadmin@ubuntu:~$
sysadmin@ubuntu:~$ curl http://site1.com
<h1> This is for site1.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/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
```



## 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:

```
sysadmin@Ubuntu2404:/etc/apache2$ sudo apache2ctl -S
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
VirtualHost configuration:
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex default: dir="/var/run/apache2/" mechanism=default
Mutex watchdog-callback: using_defaults
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="www-data" id=33
Group: name="www-data" id=33
sysadmin@Ubuntu2404:/etc/apache2$ sudo sh -c 'echo "ServerName localhost" >> /etc/apache2/apache2.conf'
sysadmin@Ubuntu2404:/etc/apache2$ sudo systemctl reload apache2
sysadmin@Ubuntu2404:/etc/apache2$ sudo apache2ctl -S
VirtualHost configuration:
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex default: dir="/var/run/apache2/" mechanism=default
Mutex watchdog-callback: using_defaults
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="www-data" id=33
Group: name="www-data" id=33
sysadmin@Ubuntu2404:/etc/apache2$
```

Remove error AH00558

## WARNING

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

## References

- [httpd.apache.org](http://httpd.apache.org)
- [phoenixnap.com](http://phoenixnap.com)
- [medium.com](https://medium.com)
- [digitalocean.com](https://digitalocean.com)
- [stackoverflow.com](https://stackoverflow.com)
- [serverfault.com](https://serverfault.com)
- [baeldung.com](https://baeldung.com)
- [askubuntu.com](https://askubuntu.com)

# How to Storage Mount Using Bind Mount on Docker?

written by sysadmin | 21 June 2025

[The previous article](#) explained how to make storage using volume in Docker. This article will explain how to make storage using Bind Mount.

## **Problem**

How to storage mount using bind mount on Docker?

## **Solution**

A bind mount is a method of hosting a directory or file that is directly mounted into a container. Since they aren't isolated by Docker, both non-Docker processes on the host and container processes can modify the mounted files simultaneously. So, you can change the data from the host, and those changes will be reflected in the container. This method is useful for development environments where code must be updated and tested in real-time. There are 2 ways when use bind mount:

### **A. using -v option**

This option (using **-v** or **--volume**) uses three fields, separated by colon characters (:). The fields must be in the correct order, and the meaning of each field is not immediately obvious. To use this option, use the format below:

```
docker run -d --name container_name -v  
/path/folder/in/server:/path/folder/in/container[:opts]  
image_name:tag
```

## INFO

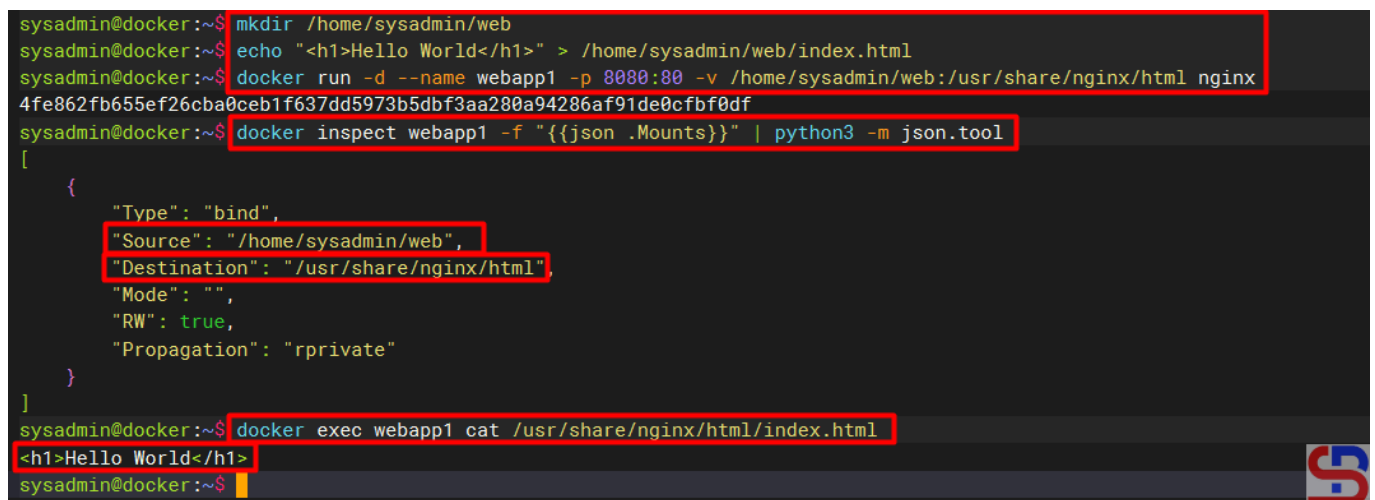
opts is the abbreviation for options like readonly, private, z, Z, and so on. The third field is optional, and is separated by colon characters. You can see the options and their descriptions [on this page](#).

To see more detailed information about mounting a container, use the format below:

```
docker inspect container_name -f "{{json .Mounts}}" | python3 -m json.tool
```

For example, you want to create a container with a nginx webserver that can be accessed with port **8080** with sources in folder **/home/sysadmin/web** and destination to the folder **/usr/share/nginx/html** on container, so first make a web folder in the server, and create a simple site then create a container using the command below:

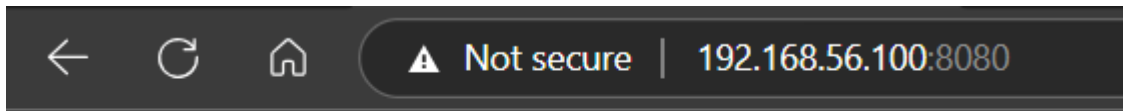
```
mkdir /home/sysadmin/web
echo "<h1>Hello World</h1>" > web/index.html
docker run -d --name webapp1 -p 8080:80 -v
/home/sysadmin/web:/usr/share/nginx/html nginx
docker inspect webapp1 -f "{{json .Mounts}}" | python3 -m json.tool
docker exec webapp1 cat /usr/share/nginx/html/index.html
```

A terminal window showing a series of Docker commands and their outputs. The commands are: 1. mkdir /home/sysadmin/web; 2. echo "<h1>Hello World</h1>" > /home/sysadmin/web/index.html; 3. docker run -d --name webapp1 -p 8080:80 -v /home/sysadmin/web:/usr/share/nginx/html nginx; 4. docker inspect webapp1 -f "{{json .Mounts}}" | python3 -m json.tool; 5. docker exec webapp1 cat /usr/share/nginx/html/index.html. The output of the inspect command is a JSON object showing the mount configuration: [{"Type": "bind", "Source": "/home/sysadmin/web", "Destination": "/usr/share/nginx/html", "Mode": "", "RW": true, "Propagation": "rprivate"}]. The output of the exec command is "<h1>Hello World</h1>".

```
sysadmin@docker:~$ mkdir /home/sysadmin/web
sysadmin@docker:~$ echo "<h1>Hello World</h1>" > /home/sysadmin/web/index.html
sysadmin@docker:~$ docker run -d --name webapp1 -p 8080:80 -v /home/sysadmin/web:/usr/share/nginx/html nginx
4fe862fb655ef26c0a0ceb1f637dd5973b5dbf3aa280a94286af91de0c9bf0df
sysadmin@docker:~$ docker inspect webapp1 -f "{{json .Mounts}}" | python3 -m json.tool
[
  {
    "Type": "bind",
    "Source": "/home/sysadmin/web",
    "Destination": "/usr/share/nginx/html",
    "Mode": "",
    "RW": true,
    "Propagation": "rprivate"
  }
]
sysadmin@docker:~$ docker exec webapp1 cat /usr/share/nginx/html/index.html
<h1>Hello World</h1>
sysadmin@docker:~$
```

Execute the commands

You can see from the image above that the index.html file in the container only displays the hello world sentence, so when you open the browser then the displays in the browser as shown below:



# Hello World



The display of the website

You can change the appearance of the website in the `index.html` file on the server or in the container. For example, you add the script below to the file on the server:

```
echo "<h2>Additional from server</h2>" >> web/index.html
```

Likewise, if you add the script below to the container:

```
docker exec webapp1 bash -c "echo Additional from container >> /usr/share/nginx/html/index.html"
```

Both scripts will be included in the `index.html` file and will be displayed on the website as shown below:



# Hello World

## Additional from the server

Additional from container



Display of the website after additional scripts

## B. Using --mount Option

In general, these options are more explicit and verbose. This option consists of multiple key-value pairs, separated by commas, and each consisting of a **<key>=<value>** tuple. The biggest difference is that the `-v` syntax combines all the options in one field, while the `--mount` syntax separates them. The `--mount` syntax is more verbose than `-v` or `--volume`, but the order of the keys is not significant, and it is easier to understand. To use this option, use the format below:

```
docker run -d --name container_name --mount
type=type_mount,source=/path/folder/in/server,destination=/path/folder/in/con
tainer[,<key>=<value>...] image_name:tag
```

### INFO

The third field is optional, and is separated by commas like `readonly`, `bind-propagation`, and so on. You can see the options and their descriptions [on this page](#).

For example, you want to create a container with an Apache web server that can be accessed on port **8081** with sources in the folder **/home/sysadmin/mount** and destination to the folder **/usr/local/apache2/htdocs** on the container, so first make the mount folder in the server, then make the container using the command below:

```
mkdir /home/sysadmin/mount
echo "<h1>Learn --mount option</h1>" > mount/index.html
docker run -d --name webapp2 -p 8081:80 --mount
type=bind,source=/home/sysadmin/mount,destination=/usr/local/apache2/htdocs
httpd
docker inspect webapp2 -f "{{json .Mounts}}" | python3 -m json.tool
docker exec webapp2 cat /usr/local/apache2/htdocs/index.html
```

```
sysadmin@docker:~$ mkdir /home/sysadmin/mount
sysadmin@docker:~$ echo "<h1>Learn --mount option</h1>" > mount/index.html
sysadmin@docker:~$ docker run -d --name webapp2 -p 8081:80 --mount type=bind,source=/home/sysadmin/mount,destination=/usr/local/apache2/htdocs httpd
5376279999b798a6f70f4ced5de48b07e387202984327421936d682a0fdcb35
sysadmin@docker:~$ docker inspect webapp2 -f "{{json .Mounts}}" | python3 -m json.tool
[
  {
    "Type": "bind",
    "Source": "/home/sysadmin/mount",
    "Destination": "/usr/local/apache2/htdocs",
    "Mode": "",
    "RW": true,
    "Propagation": "rprivate"
  }
]
sysadmin@docker:~$ docker exec webapp2 cat /usr/local/apache2/htdocs/index.html
<h1>Learn --mount option</h1>
sysadmin@docker:~$
```

Execute the commands

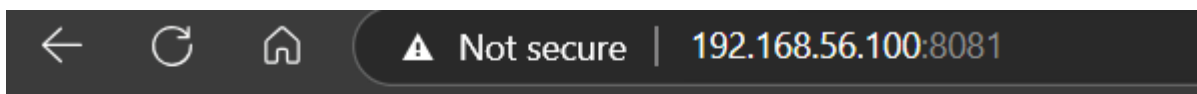
Like using the `-v` option, you can change the appearance of the website in the `index.html` file on the server or in the container. For example, you add the script below to the file on the server:

```
echo "<h2>Additional from server</h2>" >> mount/index.html
```

Likewise, if you add the script below to the container:

```
docker exec webapp2 bash -c "echo Additional from container >> /usr/local/apache2/htdocs/index.html"
```

Both scripts will be included in the `index.html` file and will be displayed on the website as shown below:



# Learn --mount option

## Additional from server

Additional from container



Display of the website after additional scripts

## Note

The only difference between the two options above is how the commands are executed; otherwise, the results would be identical.

## References

[youtube.dimas-maryanto.com](https://youtube.dimas-maryanto.com)

[youtube.com](https://youtube.com)

[docs.docker.com](https://docs.docker.com)

[docker.com](https://docker.com)

[linkedin.com](https://linkedin.com)

---

# [How to Install and Configure NFS on Linux?](#)

written by sysadmin | 21 June 2025

NFS or Network File Sharing is a protocol that allows you to share directories and files with other Linux clients over a network. Similar to locally created folders, an NFS file share is accessible when mounted on a client computer. When you have limited disk space and need to share public data between client machines, NFS is especially helpful.

## Problem

How to install and configure NFS on Linux?

## Solution

This article will explain how to install and configure NFS on 3 Linux distros: Rockylinux, Ubuntu, and OpenSuse and this article should work in each of their derivatives of the three distros.

## **A. On the server**

Following are the steps to install and configure NFS:

### **1. Install NFS**

I install NFS in the server with IP 192.168.56.2, and to install the NFS application on the Linux server, use the command below:

#### **RockyLinux**

```
sudo dnf install -y nfs-utils
```

#### **Ubuntu**

```
sudo apt update -y  
sudo apt-get install -y nfs-kernel-server
```

#### **OpenSUSE**

```
sudo zypper install -y nfs-kernel-server nfs-utils
```

### **2. Check NFS status**

Type the command below to check the NFS status:

```
systemctl status nfs-server
```

If you see the NFS status is still not on, use the command below to turn on the NFS service:

```
sudo systemctl enable --now nfs-server
```

```

[root@RockyLinux9 ~]# systemctl status nfs-server
o nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:rpc.nfsd(8)
           man:exportfs(8)
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# systemctl enable --now nfs-server
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /usr/lib/systemd/system/nfs-server.service.
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# systemctl status nfs-server
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: disabled)
   Active: active (exited) since Mon 2025-04-21 23:02:24 +08; 4s ago
     Docs: man:rpc.nfsd(8)
           man:exportfs(8)
   Process: 6169 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
   Process: 6170 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
   Process: 6189 ExecStart=/bin/sh -c if systemctl -q is-active gssproxy; then systemctl reload gssproxy ; fi (code=exited, status=0/SUCCESS)
  Main PID: 6189 (code=exited, status=0/SUCCESS)
    CPU: 115ms

Apr 21 23:02:22 RockyLinux9 systemd[1]: Starting NFS server and services...
Apr 21 23:02:24 RockyLinux9 systemd[1]: Finished NFS server and services.
[root@RockyLinux9 ~]#

```

Check the NFS service status

Sometimes you have to check the **nfs-mountd** service using the command below:

```
sudo systemctl status nfs-mountd
```

If the service is not on the server, then use the command below to turn on the service:

```
sudo systemctl start nfs-mountd
```

### 3. Check the rpcbind status

Make sure that the **rpcbind** service is actively used by NFS for the mapping port. Use the command below to check the status of the service:

```
sudo systemctl status rpcbind
```

If the service is not active, use the command below to start the service:

```
sudo systemctl enable -now rpcbind
```

#### 4. Check NFS and Portmap

To see if NFS and portmap (Portmap is a server that converts RPC program numbers into DARPA protocol port numbers. It must be running to make RPC calls) are running on the server, use the command below:

```
sudo rpcinfo -p
```

```
[root@RockyLinux9 ~]# rpcinfo -p
  program vers proto  port  service
  100000    4   tcp    111   portmapper
  100000    3   tcp    111   portmapper
  100000    2   tcp    111   portmapper
  100000    4   udp    111   portmapper
  100000    3   udp    111   portmapper
  100000    2   udp    111   portmapper
  100024    1   udp   34897  status
  100024    1   tcp   59199  status
  100005    1   udp   20048  mountd
  100005    1   tcp   20048  mountd
  100005    2   udp   20048  mountd
  100005    2   tcp   20048  mountd
  100005    3   udp   20048  mountd
  100005    3   tcp   20048  mountd
  100003    3   tcp    2049   nfs
  100003    4   tcp    2049   nfs
  100227    3   tcp    2049   nfs_acl
  100021    1   udp   42222  nlockmgr
  100021    3   udp   42222  nlockmgr
  100021    4   udp   42222  nlockmgr
  100021    1   tcp   44893  nlockmgr
  100021    3   tcp   44893  nlockmgr
  100021    4   tcp   44893  nlockmgr
[root@RockyLinux9 ~]#
```

Check whether NFS and portmap run in the server or not

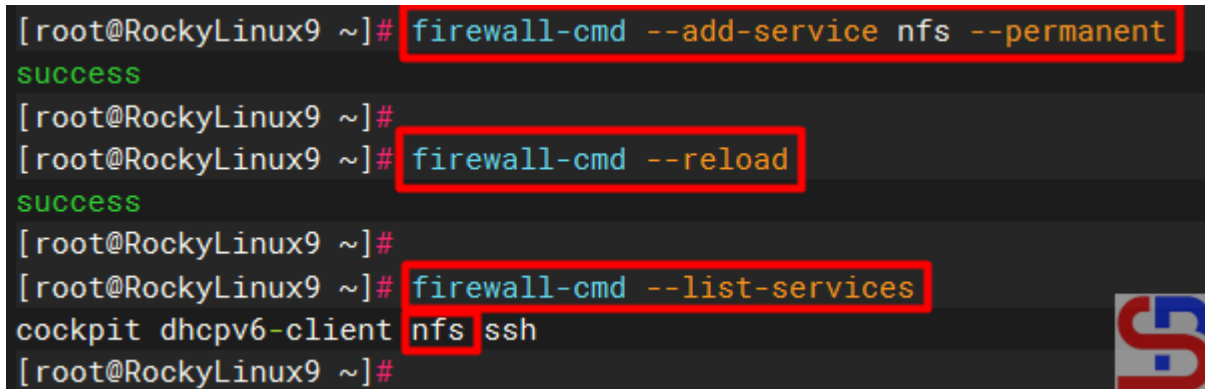
#### 5. Configure firewall

If you still turn on the firewall on Linux, use the command below to open the NFS port (Port NFS is TCP Port 2049):

##### RockyLinux & OpenSUSE

```
firewall-cmd --add-service nfs --permanent
```

```
firewall-cmd --reload
firewall-cmd --list-services
```

A terminal window screenshot from RockyLinux9. The prompt is [root@RockyLinux9 ~]#. The first command is firewall-cmd --add-service nfs --permanent, which returns success. The second command is firewall-cmd --reload, which also returns success. The third command is firewall-cmd --list-services, which returns cockpit dhcpv6-client nfs ssh. The nfs service is highlighted with a red box. There is a logo in the bottom right corner of the terminal window.

```
[root@RockyLinux9 ~]# firewall-cmd --add-service nfs --permanent
success
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# firewall-cmd --reload
success
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# firewall-cmd --list-services
cockpit dhcpv6-client nfs ssh
[root@RockyLinux9 ~]#
```

Open the NFS port in RockyLinux

### Ubuntu

```
sudo ufw allow nfs
sudo ufw status verbose
```

Use the command below to open the rpcbind port (rpcbind port is TCP Port 111):

### Rockylinux & OpenSUSE

```
firewall-cmd --add-port=111/tcp --permanent
firewall-cmd --reload
firewall-cmd --list-ports
```

### Ubuntu

```
sudo ufw allow 111
sudo ufw status verbose
```

## 6. Make a folder sharing

Create a folder to collect NFS files and folders and I make it in the folder **/var/nfs** using the command below:

```
mkdir /var/nfs
```

After that, copy the file(s) and folder(s) that you want to

share into the folder as shown below:

```
[root@RockyLinux9 ~]# mkdir /var/nfs
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# cp -R * /var/nfs/
[root@RockyLinux9 ~]# ls -al /var/nfs
total 544
drwxr-xr-x. 3 root root 103 Apr 22 03:17 .
drwxr-xr-x. 20 root root 4096 Apr 22 03:17 ..
-rw-----. 1 root root 1321 Apr 22 03:17 anaconda-ks.cfg
-rw-r--r--. 1 root root 79793 Apr 22 03:17 download.htm
-rw-r--r--. 1 root root 103917 Apr 22 03:17 image.jpeg
-rw-r--r--. 1 root root 358300 Apr 22 03:17 quickmail.zip
drwxr-xr-x. 3 root root 21 Apr 22 03:17 uploads
```

Copy the file(s) and folder(s) into the folder sharing

## 7. Define an Export File

To grant access to NFS clients, you need to define an export file and it is typically located at `/etc/exports`. Use the format below to define an export file:

`/folder/path`      `accessible-host-ip-address(options)`

The options you can use can be seen in the image below:

Option	Description
ro,rw	Read-only, Read-write (default)
rw=list	Hosts in the list can do rw, others ro only
root_squash	Maps UID 0 and GID 0 to the value of anonuid and anongid (default)
no_root_squash	Allow root access
all_squash	Maps all UID and GID to anonymous one
subtree_check	Check that the accessed file is in the appropriate filesystem and in the exported tree.
no_subtree_check	Disables subtree checking
anonuid=xxx	Related to root_squash
anongid=xxx	Related to root_squash
secure	Require remote access from privileged port
insecure	Allow remote access from any port
noaccess	Prevent access to this dir and it's subdir

Options in NFS (Image credit for [slideplayer.com](http://slideplayer.com))

You can use more than one option like (rw, sync, no\_subtree\_check). By default, NFS uses the **ro** option where the client can only read the file or folder in the folder sharing. In this article, I only want the folder sharing can only be accessed by users who only use IP 192.168.56.0/24 and the folder can be changed by the users, then use the command below to enter the script into the exports file:

```
sudo echo "/var/nfs          192.168.56.0/24(rw)" > /etc/exports
```

Then change the permissions so that the files and folders in the folder sharing can be changed using the command below:

### **RockyLinux & OpenSUSE**

```
chown -R nobody:nobody /var/nfs
sudo chmod -R 775 /var/nfs
```

## **Ubuntu**

```
chown -R nobody:nogroup /var/nfs  
sudo chmod -R 775 /var/nfs
```

### **8. Export exports file**

Use the command below to make the folder sharing available to the clients:

```
sudo exportfs -r
```

Use the command below to view the exports file:

```
showmount -e
```

To see which hosts access file sharing, use the command below:

```
sudo netstat -an | grep 2049
```

## **B. On the client**

Following are the steps to install and configure NFS:

### **1. Install NFS client**

Use the command below to install the NFS client:

#### **RockyLinux**

```
sudo dnf install -y nfs-utils
```

#### **Ubuntu**

```
sudo apt-get install -y nfs-common
```

#### **OpenSUSE**

```
zypper install -y nfs-client*
```

## 2. Check the ports in the NFS server

Use the command below to check whether the client can access the ports (port 2049 and 111) in the NFS server or not (the IP server NFS is 192.168.56.2):

```
rpcinfo -p 192.168.56.2
```

```
sysadmin@ubuntu2404:~$ rpcinfo -p 192.168.56.2
```

program	vers	proto	port	service
100000	4	tcp	111	portmapper
100000	3	tcp	111	portmapper
100000	2	tcp	111	portmapper
100000	4	udp	111	portmapper
100000	3	udp	111	portmapper
100000	2	udp	111	portmapper
100024	1	udp	44911	status
100024	1	tcp	54479	status
100005	1	udp	20048	mountd
100005	1	tcp	20048	mountd
100005	2	udp	20048	mountd
100005	2	tcp	20048	mountd
100005	3	udp	20048	mountd
100005	3	tcp	20048	mountd
100003	3	tcp	2049	nfs
100003	4	tcp	2049	nfs
100227	3	tcp	2049	nfs_acl
100021	1	udp	57003	nlockmgr
100021	3	udp	57003	nlockmgr
100021	4	udp	57003	nlockmgr
100021	1	tcp	41379	nlockmgr
100021	3	tcp	41379	nlockmgr
100021	4	tcp	41379	nlockmgr

```
sysadmin@ubuntu2404:~$
```

Check the connection between the client to the NFS server

## 2. Make and mount a folder

Make the folder where we want to mount the NFS shares from the server, for example, I made a folder in **/tmp/nfs**:

```
mkdir /tmp/nfs
```

After that the mount folder with the NFS server using the format below:

```
sudo mount -t nfs 192.168.56.2:/var/nfs /tmp/nfs
```

```
sysadmin@ubuntu2404:~$ mkdir /tmp/nfs
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ sudo mount -t nfs 192.168.56.2:/var/nfs /tmp/nfs
sysadmin@ubuntu2404:~$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	97M	1.1M	96M	2%	/run
/dev/mapper/ubuntu--vg-ubuntu--lv	9.8G	4.8G	4.6G	52%	/
tmpfs	481M	0	481M	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
/dev/sda2	1.7G	95M	1.5G	6%	/boot
tmpfs	97M	12K	97M	1%	/run/user/1000
192.168.56.2:/var/nfs	17G	1.6G	16G	10%	/tmp/nfs

```
sysadmin@ubuntu2404:~$
```

Mount the folder to the folder-sharing

### INFO

You can use the `-v` option so that the above command becomes:

```
sudo mount -v -t nfs 192.168.56.2:/var/nfs /tmp/nfs
```

to display the logs when mounting so that you can know if there is an error when mounting.

You should access the folder sharing on the NFS server as shown below:

```

sysadmin@ubuntu2404:~$ sudo mount -t nfs 192.168.56.2:/var/nfs /tmp/nfs
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ df -h

```

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	97M	1.1M	96M	2%	/run
/dev/mapper/ubuntu--vg-ubuntu--lv	9.8G	4.3G	5.0G	47%	/
tmpfs	481M	0	481M	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
/dev/sda2	1.7G	95M	1.5G	6%	/boot
tmpfs	97M	12K	97M	1%	/run/user/1000
192.168.56.2:/var/nfs	17G	1.7G	16G	10%	/tmp/nfs

```

sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ ls /tmp/nfs
anaconda-ks.cfg  download.htm  image.jpg  quickmail.zip  uploads
sysadmin@ubuntu2404:~$

```

Access to the NFS server

You can use the command below to see the NFS client connection:

```
sudo mount | grep -i nfs
```

```

sysadmin@ubuntu2404:~$ sudo mount | grep -i nfs
192.168.56.2:/var/nfs on /tmp/nfs type nfs4 (rw,relatime,vers=4.2,rsize=131072,wsiz=131072,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.56.100,local_lock=none,addr=192.168.56.2)
sysadmin@ubuntu2404:~$

```

Check the status of the NFS client

#### 4. Simulation test

Try to do the simulation by changing the file name in the folder sharing. I try to rename the download.htm file to index.html using the command below:

```
sudo mv /tmp/nfs/download.htm /tmp/nfs/index.html
```

The file was successfully changed as shown below:

```
sysadmin@ubuntu2404:~$ ls /tmp/nfs
anaconda-ks.cfg download.htm image.jpeg quickmail.zip uploads
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ sudo mv /tmp/nfs/download.htm /tmp/nfs/index.html
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ ls /tmp/nfs
anaconda-ks.cfg image.jpeg index.html quickmail.zip uploads
sysadmin@ubuntu2404:~$
```

Rename the file in NFS

### 5. Configure the fstab file

To keep the folder sharing is still connected in the client after the client is rebooted, configure the `/etc/fstab` file using the command below:

```
echo '192.168.56.2:/var/nfs /tmp/nfs nfs rw 0 0' | sudo tee -a /etc/fstab
```

```
[root@RockyLinux9 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0  4.0M   0% /dev
tmpfs           385M   0  385M   0% /dev/shm
tmpfs           154M  3.1M  151M   2% /run
/dev/mapper/r1_rockylinux9-root 17G  1.8G   16G  11% /
/dev/sda1       1014M  395M  620M  39% /boot
192.168.56.12:/var/nfs 10G  3.4G  5.8G  37% /tmp/nfs
tmpfs           77M    0   77M   0% /run/user/1000
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# sudo echo '192.168.56.12:/var/nfs /tmp/nfs nfs rw 0 0' | sudo tee -a /etc/fstab
192.168.56.12:/var/nfs /tmp/nfs nfs rw 0 0
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Sep 19 07:29:32 2024
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/r1_rockylinux9-root / xfs defaults 0 0
UUID=066eb699-fd9c-45ae-bba8-6c220e767ed7 /boot xfs defaults 0 0
/dev/mapper/r1_rockylinux9-swap none swap defaults 0 0
192.168.56.12:/var/nfs /tmp/nfs nfs rw 0 0
[root@RockyLinux9 ~]#
```

Insert the script to fstab file

### C. Errors and solutions

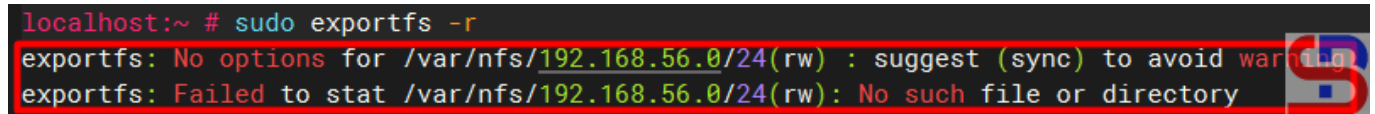
Below are errors that often appear and their solutions:

## 1. No options for /var/nfs

Sometimes when you run the **exportfs -r** command, there is an error as below:

```
exportfs: No options for /var/nfs/192.168.56.0/24(rw) : suggest (sync) to avoid warning
exportfs: Failed to stat /var/nfs/192.168.56.0/24(rw): No such file or directory
```

```
localhost:~ # sudo exportfs -r
exportfs: No options for /var/nfs/192.168.56.0/24(rw) : suggest (sync) to avoid warning
exportfs: Failed to stat /var/nfs/192.168.56.0/24(rw): No such file or directory
```



Error failed to stat

To eliminate the error, check in the **/etc/exports** file and you have to fix the writing in the file from:

```
/var/nfs/192.168.56.0/24(rw)
```

changed into

```
/var/nfs 192.168.56.0/24(rw)
```

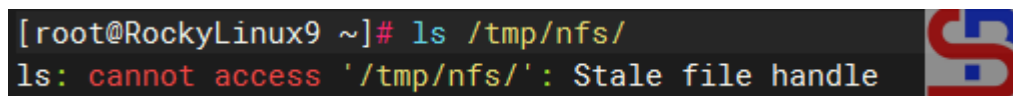
After that, run the **exportfs -r** command again and the error should disappear.

## 2. Error Stale file handle

When you want to connect a client to the NFS server there is an error like the below (usually this happens if there is an error like number 1 or other causes on the NFS server):

Stale file handle

```
[root@RockyLinux9 ~]# ls /tmp/nfs/
ls: cannot access '/tmp/nfs/': Stale file handle
```



Stale file handle error

To solve this error you have to unmount on the side of the client and then mount back as shown below:

```
[root@RockyLinux9 ~]# ls /tmp/nfs/
ls: cannot access '/tmp/nfs/': Stale file handle
[root@RockyLinux9 ~]# sudo umount -f /tmp/nfs
[root@RockyLinux9 ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                   4.0M         0  4.0M   0% /dev
tmpfs                       385M         0  385M   0% /dev/shm
tmpfs                       154M     3.1M  151M   2% /run
/dev/mapper/rl_rockylinux9-root 17G     1.8G   16G  11% /
/dev/sda1                   1014M     395M   620M  39% /boot
tmpfs                       77M         0    77M   0% /run/user/1000
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# sudo mount -t nfs 192.168.56.12:/var/nfs /tmp/nfs
[root@RockyLinux9 ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                   4.0M         0  4.0M   0% /dev
tmpfs                       385M         0  385M   0% /dev/shm
tmpfs                       154M     3.1M  151M   2% /run
/dev/mapper/rl_rockylinux9-root 17G     1.8G   16G  11% /
/dev/sda1                   1014M     395M   620M  39% /boot
tmpfs                       77M         0    77M   0% /run/user/1000
192.168.56.12:/var/nfs     10G     3.4G   5.8G  37% /tmp/nfs
[root@RockyLinux9 ~]# ls /tmp/nfs/
bin  get-docker.sh
[root@RockyLinux9 ~]#
```

Solve the stale file handle error

### 3. RPC: Program not registered

When typing the **showmount -e** command on the NFS server there is an error as below:

```
clnt_create: RPC: Program not registered
```

```
localhost:~ # showmount -e
clnt_create: RPC: Program not registered
Error Program Not Registered
```

The solution is that you have to run the command below so that the nfs-mountd service runs on the server:

```
systemctl start nfs-mountd
```

#### 4. Permission denied

When you want to connect to the NFS server or when you want to change the file in the NFS, there is an error like this:

Permission denied

```
sysadmin@ubuntu2404:~$ cp /tmp/nfs/anaconda-ks.cfg /tmp/nfs/test.cfg
cp: cannot create regular file '/tmp/nfs/test.cfg': Permission denied
Error Permission denied
```

The solution is to check the exports file on the NFS server and make sure that the folder has been given permissions as in step 5 in the server section.

#### Note

If you want to block an IP address of a host so the host can't access the NFS server, use the command below to block the IP host:

```
sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="192.168.56.100" port port="2049" protocol="tcp" reject'
sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="192.168.56.100" port port="111" protocol="tcp" reject'
sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="192.168.56.100" port port="2049" protocol="udp" reject'
sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="192.168.56.100" port port="111" protocol="udp" reject'
sudo firewall-cmd --reload
sudo firewall-cmd --list-rich-rules
```

and should the client with IP 192.168.56.100 not be able to access the folder sharing as shown in the image below:

```
sysadmin@ubuntu2404:~$ ip a | grep 56
inet 192.168.56.100/24 brd 192.168.56.255 scope global enp0s8
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ sudo mount -t nfs 192.168.56.2:/var/nfs /tmp/nfs
mount.nfs: Connection refused for 192.168.56.2:/var/nfs on /tmp/nfs
sysadmin@ubuntu2404:~$
```

Can not mount to NFS server

If you want to delete an IP address of a host then the option **--add-rich-rule** becomes **--remove-rich-rule** so that the command becomes as command below:

```
sudo firewall-cmd --permanent --remove-rich-rule='rule family="ipv4" source
address="192.168.56.100" port port="2049" protocol="tcp" reject'
sudo firewall-cmd --permanent --remove-rich-rule='rule family="ipv4" source
address="192.168.56.100" port port="111" protocol="tcp" reject'
sudo firewall-cmd --permanent --remove-rich-rule='rule family="ipv4" source
address="192.168.56.100" port port="2049" protocol="udp" reject'
sudo firewall-cmd --permanent --remove-rich-rule='rule family="ipv4" source
address="192.168.56.100" port port="111" protocol="udp" reject'
sudo firewall-cmd --reload
sudo firewall-cmd --list-rich-rules
```

#### WARNING

In my experience, you can't immediately block a client to NFS if the client is still connected to the NFS. You have to wait until the client disconnects to the NFS server, either the host reboots or others.

## References

[bluexp.netapp.com](http://bluexp.netapp.com)  
[redhat.com](http://redhat.com)  
[phoenixnap.com](http://phoenixnap.com)  
[howtoforge.com](http://howtoforge.com)  
[youtube.com](http://youtube.com)  
[docs.oracle.com](http://docs.oracle.com)  
[linux.die.net](http://linux.die.net)

---

## [How to Connect Storage Mount Using Volume Mount on Docker?](#)

written by sysadmin | 21 June 2025

[The previous article](#) explained how to access the database installed in a container. But you must know that if you delete the container, it will automatically delete the

database too. Therefore, you must create a storage to store a database so that if you delete the container, either intentionally or unintentionally, the database will remain. By default, Docker supports the following types of storage mounts for storing data outside of the container's writable layer: volume mounts, bind mounts, and tmpfs mounts, each with its unique characteristics and use cases. This article will explain how to connect storage mount using volume mount. You can see a summary of the three types of storage [here](#).

## **Problem**

How to connect storage mount using volume mount on Docker?

## **Solution**

Volumes are persistent storage mechanisms managed by the Docker daemon. It is stored in the filesystem on the host in `/var/lib/docker/volumes` folder, but to interact with the data in the volume, you must mount the volume to a container. Volumes are ideal for performance-critical data processing and long-term storage needs and are also suitable for sharing data between containers since volumes can be attached to multiple containers.

### **A. List, remove and create the volume**

To see the volume on the server, use the command below:

```
docker volume ls
```

To remove a volume, use the command below:

```
docker volume rm volume_name
```

Use the command below to remove all unused volumes:

```
docker volume prune
```

By default, there is no volume if there is no container on the server. But if there is a container that uses storage like a container that uses a database image, the volume will form automatically. To see which containers that use volume can use the command below:

```
docker container inspect webapp postgresdb mysql db -f '{{ .Name }} => {{json .Mounts}}'
```

```
sysadmin@docker:~$ docker volume ls
DRIVER      VOLUME NAME
sysadmin@docker:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
5467510f142e2879394c382f2b58853739126e87871aedabf2d88b0922e61c
sysadmin@docker:~$ docker run -d --name webapp1 nginx
sysadmin@docker:~$ docker run -d --name db_mysql -e MYSQL_ROOT_PASSWORD='q1w2e3r4' mysql
f0e968bf2533a98c74d16f19b81d977b66d440e663820e8171106d23a2b2b553
sysadmin@docker:~$ docker volume ls
DRIVER      VOLUME NAME
local      198cb03b7a38d6b6840af7e4a676c5a7310f96c6fb4a03b83961b2405afeeb9a
sysadmin@docker:~$ docker container inspect $(docker ps -a -q) -f '{{ .Name }} => {{json .Mounts}}'
/db_mysql => [{"Type":"volume","Name":"198cb03b7a38d6b6840af7e4a676c5a7310f96c6fb4a03b83961b2405afeeb9a","Source":"/var/lib/docker/volumes/198cb03b7a38d6b6840af7e4a676c5a7310f96c6fb4a03b83961b2405afeeb9a/_data","Destination":"/var/lib/mysql","Driver":"local","Mode":"","RW":true,"Propagation":""}]
/webapp1 => []
sysadmin@docker:~$
```

Display the container that uses volume

You can create a new volume using the format below:

```
docker volume create volume_name
```

Use the command below if you want to create a mysql\_vol volume:

```
docker volume create mysql_vol
```

To see this volume information in detail, use the command below:

```
docker volume inspect mysql_vol
```

```
sysadmin@docker:~$ docker volume inspect mysql_vol
[
  {
    "CreatedAt": "2025-04-16T16:00:28Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/mysql_vol/_data",
    "Name": "mysql_vol",
    "Options": null,
    "Scope": "local"
  }
]
sysadmin@docker:~$
```



Inspect the volume

## B. Connect storage mount to a container

After that, make a new container where the container mounts to the volume you created using the format below:

```
docker run -d --name container_name -v volume_name:/path/folder/in/container
image_name
```

You can see more detailed information on mounting a container, using the format below:

```
docker inspect container_name -f "{{json .Mounts}}" | python3 -m json.tool
```

For example, use the command below if you want to create a db\_mysql container with a password q1w2e3r4 using mysql\_vol volume in the folder /var/lib/mysql :

```
docker run -d --name db_mysql -e MYSQL_ROOT_PASSWORD='q1w2e3r4' -v
mysql_vol:/var/lib/mysql mysql
docker inspect db_mysql -f "{{json .Mounts}}" | python3 -m json.tool
```

```
sysadmin@docker:~$ docker run -d --name db_mysql -e MYSQL_ROOT_PASSWORD='q1w2e3r4' -v mysql_vol:/var/lib/mysql mysql
6ad689b31c9d559aac3b360c23434b591bf650fbec017629ff26b273da5e7d7e
sysadmin@docker:~$
sysadmin@docker:~$ docker inspect db_mysql -f "{{json .Mounts}}" | python3 -m json.tool
[
  {
    "Type": "volume",
    "Name": "mysql_vol",
    "Source": "/var/lib/docker/volumes/mysql_vol/_data",
    "Destination": "/var/lib/mysql",
    "Driver": "local",
    "Mode": "z",
    "RW": true,
    "Propagation": ""
  }
]
sysadmin@docker:~$
```

Run the container and check the mount

### C. Simulate remove the container

After that, try to enter the container by using the command below:

```
docker exec -it db_mysql mysql -uroot -pq1w2e3r4
```

Create a database and fill it as shown in the image below:

```

sysadmin@docker:~$ docker exec -it db_mysql mysql -uroot -pq1w2e3r4
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE lab_db;
Query OK, 1 row affected (0.064 sec)

mysql> use lab_db
Database changed
mysql> CREATE TABLE students (
->     id INT AUTO_INCREMENT PRIMARY KEY
->     name VARCHAR(100) NOT NULL,
->     age INT NOT NULL
-> );
Query OK, 0 rows affected (0.175 sec)

mysql> INSERT INTO students (name, age) VALUES ('Alice', 21);
Query OK, 1 row affected (0.115 sec)

mysql> select * from students;
+----+-----+-----+
| id | name  | age  |
+----+-----+-----+
|  1 | Alice |  21  |
+----+-----+-----+
1 row in set (0.002 sec)

mysql> \q
Bye

```

Create a database

Then, try to simulate by deleting the container and creating a new container where the data is put into the mysql\_vol volume in the /var/lib/mysql folder using the command below:

```

docker stop db_mysql
docker rm db_mysql
docker run -d --name db_mysql_new -e MYSQL_ROOT_PASSWORD='q1w2e3r4' -v
mysql_vol:/var/lib/mysql mysql

```



Access the container by using the command:

```
docker exec -it db_mysql_new mysql -uroot -pq1w2e3r4
```

The lab\_db database should still be accessible using the container you just created, as shown in the image below:

```
sysadmin@docker:~$ docker stop db_mysql
db_mysql
sysadmin@docker:~$ docker rm db_mysql
db_mysql
sysadmin@docker:~$ docker run -d --name db_mysql_new -e MYSQL_ROOT_PASSWORD='q1w2e3r4' -v mysql_vol:/var/lib/mysql mysql
b09d93a86523177327c20da04323cbad45c138867a63bd24d3a0707b151c2af2
sysadmin@docker:~$ docker exec -it db_mysql_new mysql -uroot -pq1w2e3r4
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use lab_db
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from students;
+----+-----+-----+
| id | name | age |
+----+-----+-----+
|  1 | Alice |  21 |
+----+-----+-----+
1 row in set (0.015 sec)

mysql> 
```

The database can still be accessed

## D. Share data in the volume among containers

Let's do a simulation to share the data that is in the volume with more than one container on a single server. In this article, 2 containers will be created that are connected to a mysql\_vol volume. Type the command below to make 2 containers:

```
docker run -d --name db_mysql1 -e MYSQL_ROOT_PASSWORD='q1w2e3r4' -v
mysql_volume:/var/lib/mysql mysql
docker run -d --name db_mysql2 -e MYSQL_ROOT_PASSWORD='q1w2e3r4' -v
mysql_volume:/var/lib/mysql mysql
docker ps
```

```
sysadmin@docker:~$ docker run -d --name db_mysql1 -e MYSQL_ROOT_PASSWORD='q1w2e3r4' -v mysql_vol:/var/lib/mysql mysql
c1ea3cf4e789f13b3d9ccd5df8e4f08e11f26209ca68752c1559a09fae276938
sysadmin@docker:~$
sysadmin@docker:~$ docker run -d --name db_mysql2 -e MYSQL_ROOT_PASSWORD='q1w2e3r4' -v mysql_vol:/var/lib/mysql mysql
62e62f11319726962f8c697a2b0bd9ba361331e516f6a85078f08cc465c4855a
sysadmin@docker:~$
sysadmin@docker:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
62e62f113197   mysql    "docker-entrypoint.s..." 9 seconds ago  Up 8 seconds  3306/tcp, 33060/tcp  db_mysql2
c1ea3cf4e789   mysql    "docker-entrypoint.s..." 21 seconds ago Up 21 seconds  3306/tcp, 33060/tcp  db_mysql1
```

Create 2 containers

After that, type the command below to create a database and insert the data via container db\_mysql1:

```
docker exec db_mysql1 bash -c "mysql -uroot -pq1w2e3r4 -e \"CREATE DATABASE db_school; USE db_school; CREATE TABLE students (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, age INT NOT NULL); INSERT INTO students (name, age) VALUES ('bob', 21), ('John', 22);select * from students;\""
```

Then there will be a display below:

```
sysadmin@docker:~$ docker exec db_mysql1 bash -c "mysql -uroot -pq1w2e3r4 -e \"CREATE DATABASE db_school; USE db_school; CREATE TABLE students (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, age INT NOT NULL); INSERT INTO students (name, age) VALUES ('bob', 21), ('John', 22);select * from students;\"
mysql: [Warning] Using a password on the command line interface can be insecure.
id      name  age
1       bob   21
2       John  22
sysadmin@docker:~$
```

Execute the command to create a database and insert data

From the picture above, there are 2 data in the student table in the db\_school database. Type the command below to add data to the table from container db\_mysql2:

```
docker exec db_mysql1 bash -c "mysql -uroot -pq1w2e3r4 -e \"USE db_school; INSERT INTO students (name, age) VALUES ('Laura', 20), ('Andrew', 23);select * from students;\""
```

```
sysadmin@docker:~$ docker exec db_mysql1 bash -c "mysql -uroot -pq1w2e3r4 -e \"USE db_school; INSERT INTO students (name, age) VALUES ('Laura', 20), ('Andrew', 23);select * from students;\"
mysql: [Warning] Using a password on the command line interface can be insecure.
id      name  age
1       bob   21
2       John  22
3       Laura 20
4       Andrew 23
sysadmin@docker:~$
```

Execute the command to add data

In the picture above, there are 4 data in the student table so that the Docker volume data can be shared between containers on a server well.

## Note

You can see the picture below to see the difference between the three storage:

FEATURES	Volume	Bind Mount	tmpfs Mount
Persistence	Yes (data persists after container stops)	Depends on the host structure	No (data lost when container stops)
Performance	Good (optimized for Docker)	Good (depends on host disk speed)	Excellent (in-memory storage)
Use Case	Production data, stateful applications	Development, real-time file updates	Temporary data, caching
Management	Managed by Docker, easier to back up	User-managed, requires manual handling	User-managed, temporary & ephemeral
Security	More isolated from the host filesystem	Direct access to host, higher risk	Isolated to container, secure
Backup/Restore	Easily backed up & restored with Docker Commands	Manual backup required	Not applicable (ephemeral)

Types of storage in Docker (Image credit from [linkedin.com](https://www.linkedin.com))

## References

- [docs.docker.com](https://docs.docker.com)
  - [medium.com](https://medium.com)
  - [linkedin.com](https://www.linkedin.com)
  - [hub.docker.com](https://hub.docker.com)
  - [youtube.com](https://www.youtube.com)
-

# How to Download a Package With Its Dependencies in RockyLinux?

written by sysadmin | 21 June 2025

To install a package on RockyLinux distro which is a derivative of RHEL and derivatives such as AlmaLinux or CentOS and others, just type **yum install** or **dnf install** and follow by the name of the package to be installed, then the package will be directly installed immediately on the server. But sometimes I just want to download a package with its dependencies for some purposes like installing into the server that can't connect to the internet or just for some experiments.

## Problems

How to download a package with its dependencies in RockyLinux?

## Solution

There are 2 methods to download a package with all its dependencies without installing the package on the server and both methods work successfully on RockyLinux9:

### A. Using **downloadonly** plugin

To execute this method, use the format below:

```
yum install --downloadonly --downloadaddir=/folder/path/in/linux package_name -y
```

For example, suppose you want to download nginx packages along with their dependencies, and all the packages are contained in the **/tmp/nginx** folder, use the following command:

```
yum install --downloadonly --downloadaddir=/tmp/nginx nginx -y
```

After you run the above command, the nginx package and its dependencies files should be in the /tmp/nginx folder as shown below:

```
[root@RockyLinux9 ~]: yum install --downloadonly --downloaddir=/tmp/nginx nginx -y
Last metadata expiration check: 0:24:06 ago on Fri 18 Apr 2025 11:10:24 PM +08.
Dependencies resolved.
=====
Package                Architecture          Version              Repository           Size
=====
Installing:
nginx                  x86_64                2:1.20.1-20.e19.0.1  appstream            36 k
Installing dependencies:
nginx-core             x86_64                2:1.20.1-20.e19.0.1  appstream            566 k

Transaction Summary
=====
Install 2 Packages

Total download size: 601 k
Installed size: 1.7 M
YUM will only download packages for the transaction.
Downloading Packages:
(1/2): nginx-1.20.1-20.e19.0.1.x86_64.rpm           58 kB/s | 36 kB   00:00
(2/2): nginx-core-1.20.1-20.e19.0.1.x86_64.rpm     323 kB/s | 566 kB 00:01
-----
Total                                               212 kB/s | 601 kB 00:02
Complete!
The downloaded packages were saved in cache until the next successful transaction.
You can remove cached packages by executing 'yum clean packages'.
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# ls /tmp/nginx/
nginx-1.20.1-20.e19.0.1.x86_64.rpm  nginx-core-1.20.1-20.e19.0.1.x86_64.rpm
[root@RockyLinux9 ~]#
```

Download a nginx file using downloadonly option

By default, if you forget to type the `--downloaddir` option, it will be stored in the `/var/cache/dnf/appstream-*/packages` folder.

## B. Using yumdownloader

Run the command below to install the yum-utils package:

```
yum install yum-utils
```

Then to execute this method, use the format below:

```
yumdownloader --destdir=/path/in/linux --resolve package_name -y
```

For example, suppose you want to download a vim package and it is stored in the `/root/vim` folder, then use the command

below:

```
yumdownloader --destdir=/root/vim --resolve vim -y
```

After you run the above command, the vim package and its dependencies files should be in the `/root/vim` folder as shown below:

```
[root@RockyLinux9 ~]# yumdownloader --destdir=/root/vim --resolve vim -y
Last metadata expiration check: 0:17:29 ago on Fri 18 Apr 2025 11:10:24 PM +08.
(1/4): vim-filesystem-8.2.2637-21.e19.noarch.rpm          13 kB/s | 9.3 kB    00:00
(2/4): gpm-libs-1.20.7-29.e19.x86_64.rpm                1.7 kB/s | 20 kB   00:11
(3/4): vim-enhanced-8.2.2637-21.e19.x86_64.rpm          52 kB/s | 1.7 MB   00:34
(4/4): vim-common-8.2.2637-21.e19.x86_64.rpm           57 kB/s | 6.6 MB   01:57
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# ls /root/vim
gpm-libs-1.20.7-29.e19.x86_64.rpm      vim-enhanced-8.2.2637-21.e19.x86_64.rpm
vim-common-8.2.2637-21.e19.x86_64.rpm vim-filesystem-8.2.2637-21.e19.noarch.rpm
```

Download a vim package using yumdownloader

By default, if you forget to type the `--destdir` option, the package and its dependencies will be saved in the folder where the command is run. If for example the command is executed in `/root` then the package and its dependencies files will be stored in the `/root` folder.

### Warning

You can download more than one package either using the first method or the second method, but this is not recommended because there will be mixing between dependencies files in one folder so that it makes confusion. So it is recommended to download only one package in one folder.

### Note

If you have installed a package on the server and you want to download the package along with the dependencies files, but you use the `downloadonly` plugin then you can't download the package like in the image below:

```
[root@RockyLinux9 ~]# yum install --downloadonly --downloadaddir=/tmp/httpd httpd -y
Last metadata expiration check: 1:23:27 ago on Fri 18 Apr 2025 11:10:24 PM +08.
Package httpd-2.4.62-1.el9_5.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@RockyLinux9 ~]#
```



Failed to download a package

so you have to remove the package from the server first and then you can download the package by running the previous command. However, you can still download packages that are already installed on the server if you use yumdownloader. And if you want to install the downloaded package along with its dependency files then go to the downloaded folder and type the command below:

```
rpm -ivh *
```

so you can install the package easily and quickly like in the image below:

```
[root@RockyLinux9 ~]# cd vim
[root@RockyLinux9 vim]#
[root@RockyLinux9 vim]# ls
gpm-libs-1.20.7-29.el9.x86_64.rpm      vim-enhanced-8.2.2637-21.el9.x86_64.rpm
vim-common-8.2.2637-21.el9.x86_64.rpm  vim-filessystem-8.2.2637-21.el9.noarch.rpm
[root@RockyLinux9 vim]#
[root@RockyLinux9 vim]# rpm -ivh *
Verifying... ##### [100%]
Preparing... ##### [100%]
Updating / installing...
 1:vim-filessystem-2:8.2.2637-21.el9 ##### [ 25%]
 2:vim-common-2:8.2.2637-21.el9 ##### [ 50%]
 3:gpm-libs-1.20.7-29.el9 ##### [ 75%]
 4:vim-enhanced-2:8.2.2637-21.el9 ##### [100%]
[root@RockyLinux9 vim]#
```



Install a package with its dependencies

## References

[access.redhat.com](https://access.redhat.com)  
[dbpilot.net](https://dbpilot.net)  
[ostechnix.com](https://ostechnix.com)