

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

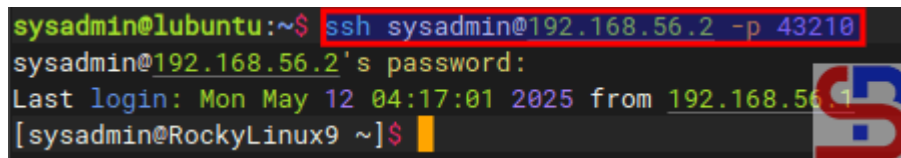
By default, you must use the format below to forward a port

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 connection. The prompt is 'sysadmin@lubuntu:~\$'. The command 'ssh sysadmin@192.168.56.2 -p 43210' is entered and highlighted with a red box. 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 yellow cursor. A red and blue 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

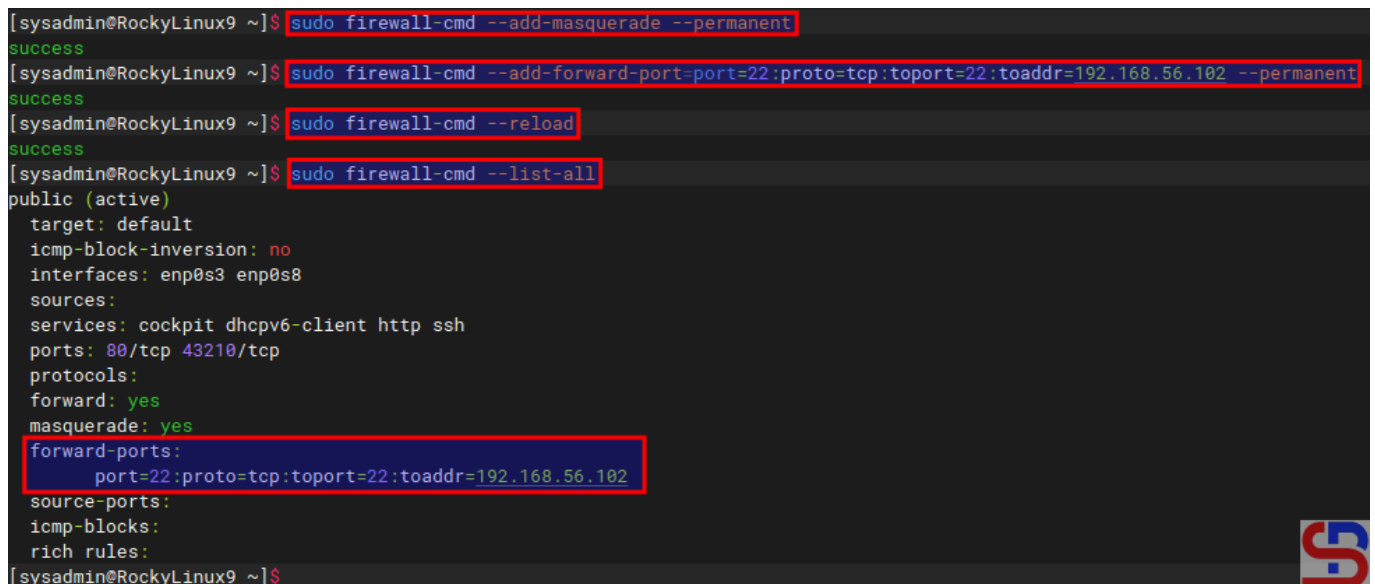
B. Forward to a different IP/device

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
```

A terminal window screenshot from a RockyLinux9 system. The user is sysadmin. The commands and their outputs are as follows:
1. Command: `sudo firewall-cmd --add-masquerade --permanent`
Output: `success`
2. Command: `sudo firewall-cmd --add-forward-port=port=22:proto=tcp:toport=22:toaddr=192.168.56.102 --permanent`
Output: `success`
3. Command: `sudo firewall-cmd --reload`
Output: `success`
4. Command: `sudo firewall-cmd --list-all`
Output: `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:`
The prompt returns to `[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
youtube.com
musaamin.web.id
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?

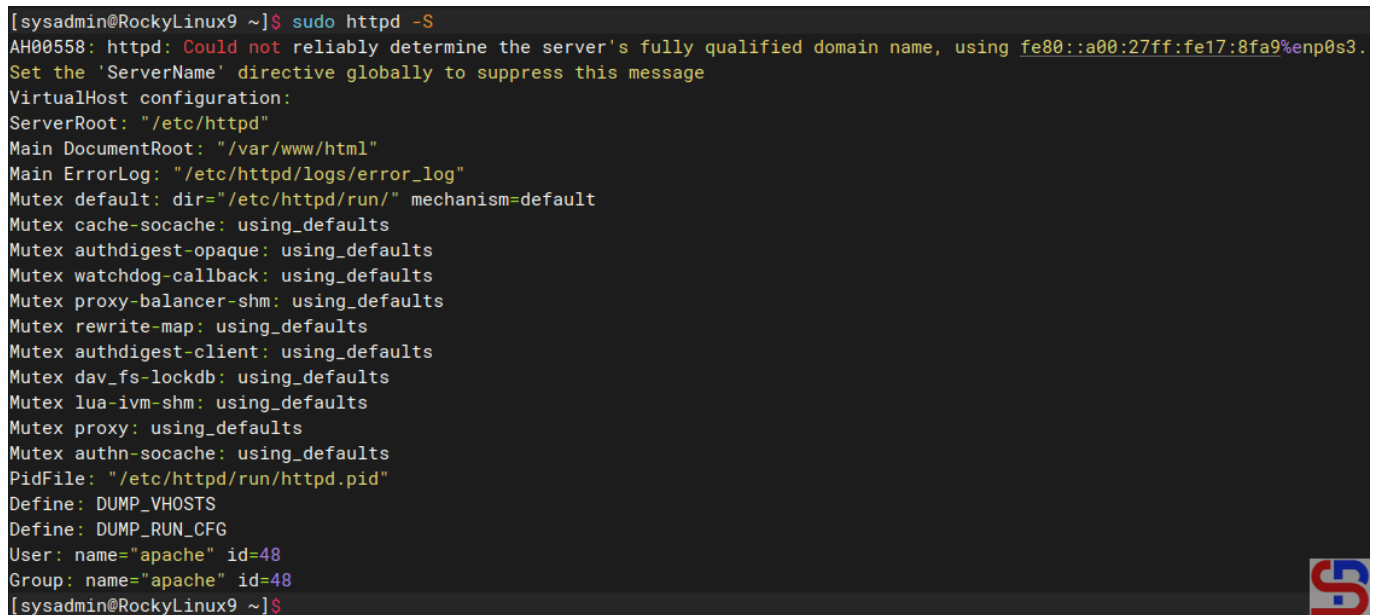
Solution

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`



This is for website1.com domain



Site website1.com:8080

B. IP-based virtual hosts

The meaning of IP-based is that you use a different IP address for each website. For example, you have 2 IPs and 2 domains, where IP **192.168.56.2 is for site1.com**, and IP **192.168.56.104 is for site2.com**. This article will use a server that has 2 IPs, as shown below:

```
[root@RockyLinux9 sysconfig]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:17:8f:a9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.104/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s3
        valid_lft 576sec preferred_lft 576sec
    inet6 fe80::a00:27ff:fe17:8fa9/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:38:ad:88 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.2/24 brd 192.168.56.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe38:ad88/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@RockyLinux9 sysconfig]#
```

Using 2 NICs in a server

1. Create the directories and the files

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

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

WARNING

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

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

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

2. Change ownership

Change the ownership of the folders:

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

3. Configuration of virtual hosts

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

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

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

```
/dev/null
```

4. Check the configuration

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

```
sudo apachectl configtest
```

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

```
sudo systemctl reload httpd
```

WARNING

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

5. Check in the browser

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

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

Info

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

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

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

```
firewall-cmd --reload
```

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

`http://site1.com`

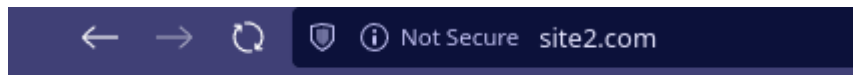


This is for site1.com domain



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
reintech.io
tecmint.com
httpd.apache.org
askubuntu.com
dewaweb.com
arubacloud.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
```

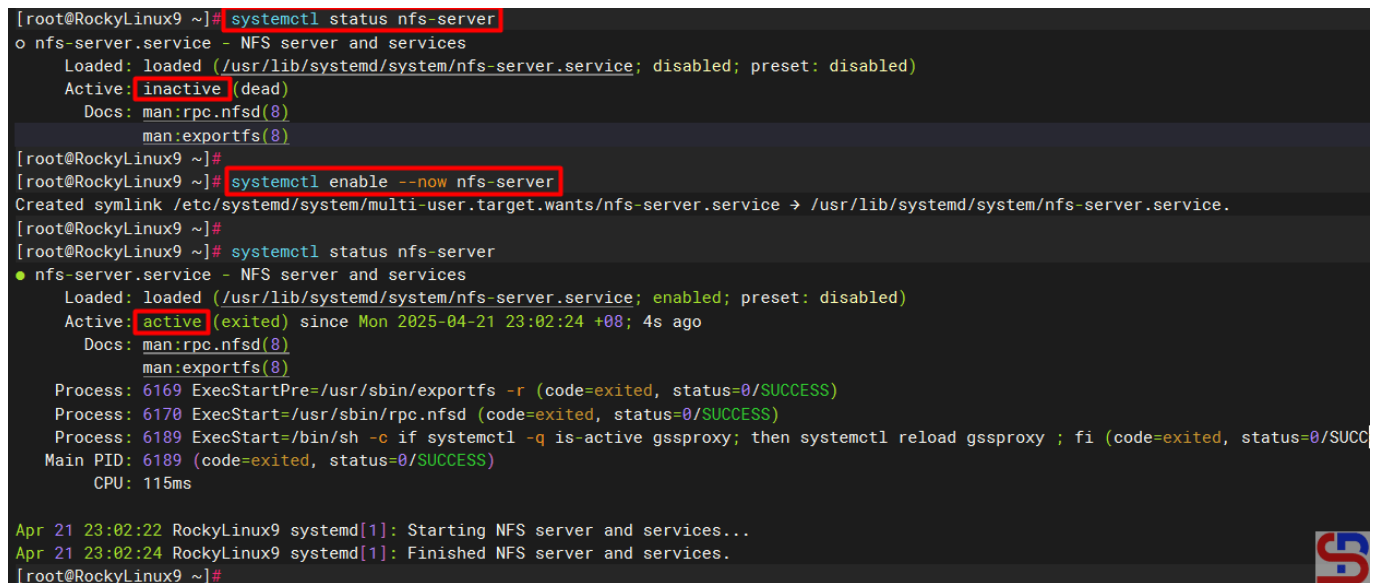
2. Check NFS status

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
```

A terminal window screenshot from RockyLinux9. The first command is `systemctl status nfs-server`, which shows the service is loaded but inactive. The second command is `systemctl enable --now nfs-server`, which creates a symlink and enables the service. The third command is `systemctl status nfs-server`, which shows the service is now active and running. The output includes details about the service's state, documentation, and the processes it has started.

```
[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
```

```
[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 ~]#
[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](https://www.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:~$
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:

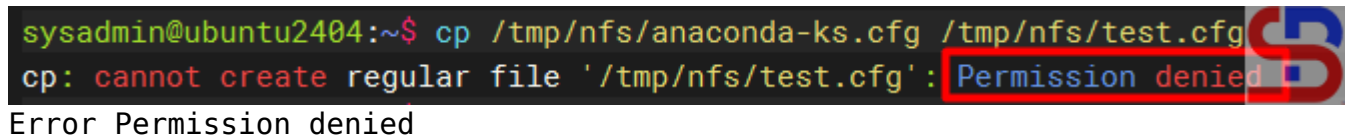
```
systemctl start nfs-mountd
```

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
```

A terminal window showing a user attempting to copy a file from /tmp/nfs/anaconda-ks.cfg to /tmp/nfs/test.cfg. The command is 'cp /tmp/nfs/anaconda-ks.cfg /tmp/nfs/test.cfg'. The output is 'cp: cannot create regular file '/tmp/nfs/test.cfg': Permission denied'. The error message is highlighted with a red box. To the right of the terminal output is a large blue and red 'S' logo.

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.

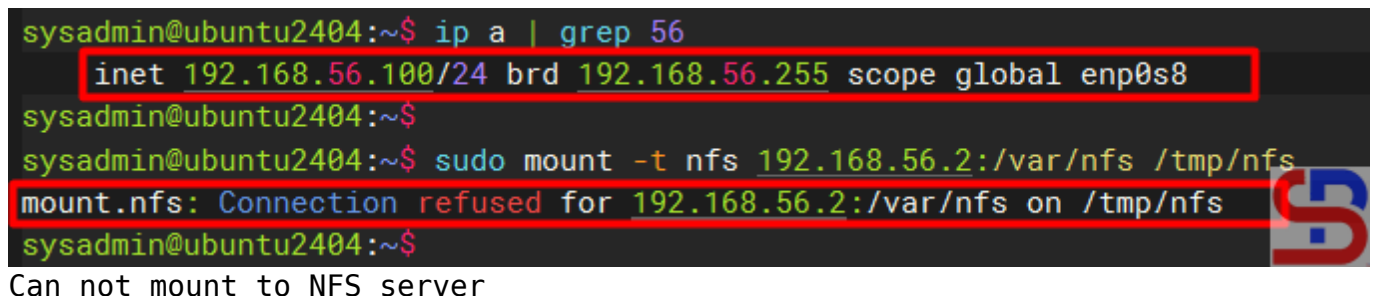
Note

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:~$
```

A terminal window showing two commands. The first is 'ip a | grep 56', which outputs 'inet 192.168.56.100/24 brd 192.168.56.255 scope global enp0s8'. The second is 'sudo mount -t nfs 192.168.56.2:/var/nfs /tmp/nfs', which outputs 'mount.nfs: Connection refused for 192.168.56.2:/var/nfs on /tmp/nfs'. Both the output lines are highlighted with red boxes. To the right of the terminal output is a large blue and red 'S' logo.

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
redhat.com
phoenixnap.com
howtoforge.com
youtube.com
docs.oracle.com
linux.die.net

[How to Install Nagios on RockyLinux?](#)

written by sysadmin | 21 June 2025

[The previous article](#) explained how to install the Nagios application on Ubuntu. This article will explain how to install the Nagios application on RockyLinux.

Problem

How to install Nagios on RockyLinux?

Solution

Below are the steps to install Nagios on RockyLinux and work on RockyLinux 9.5 and below. But I think these steps should apply to installing Nagios on RHEL and its derivatives, such as CentOS, AlmaLinux, and so on.

1. Download the packages

Install the packages needed to install Nagios using the command below:

```
yum install -y httpd php php-devel gcc glibc glibc-common gd gd-devel make  
net-snmp-* wget zip unzip php-mysqlnd php-mysql*
```

2. Create a user and a group

Create a user and group for Nagios using the commands:

```
useradd nagios  
groupadd nagcmd  
usermod -G nagcmd nagios  
usermod -G nagcmd apache
```

3. Download Nagios

Use the commands below to download Nagios, where at the time of this writing (February 2025), the latest version of Nagios is version 4.5.9:

```
cd /tmp  
wget  
https://github.com/NagiosEnterprises/nagioscore/archive/refs/heads/master.zip  
-O nagios.zip  
unzip nagios.zip  
cd nagioscore-master/
```

4. Install Nagios

By default, Linux will create a Nagios folder in the /usr/local folder to save Nagios configuration files. So, use the following commands to install Nagios:

```
./configure
```

Info

If you want to save all Nagios files in a non-default folder, for example, in the /data folder, then use the following command: **./configure -prefix=/data/nagios**

After that, run the following commands:

```
make all
make install
make install-init
make install-commandmode
make install-config
make install-webconf
```

5. Create the password

Create a password for the user to access the Nagios application. Usually, nagiosadmin is a popular username for Nagios, but you can create another username.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

```
[root@RockyLinux9 nagioscore-master]# htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[root@RockyLinux9 nagioscore-master]#
```

Create a password for the nagiosadmin user

Info

If you installed Nagios in a non-default folder, for example, in the /data folder, execute the below command: **htpasswd -c /data/nagios/etc/htpasswd.users nagiosadmin**

6. Download Nagios Plugins

Plugins are compiled executables or scripts (Perl, shell, Python, PHP, Ruby, etc.) that can be run from a command line to check the status of a host or service. Nagios Core uses the results from plugins to determine the current status of hosts and services on your network. As of this writing (February 2025), the latest version of Nagios plugins is version 2.4.12. You can check the latest version of Nagios plugins on this site. Run the following commands to download Nagios plugins:

```
cd /tmp
wget
https://github.com/nagios-plugins/nagios-plugins/archive/refs/heads/master.zip
p -O nagios-plugins.zip
unzip nagios-plugins.zip
cd nagios-plugins-master/
```

7. Install Nagios Plugins

After that, install Nagios plugins using the following commands:

```
./tools/setup
sudo ./configure --with-nagios-user=nagios --with-nagios-group=nagios
sudo make
sudo make install
```

8. Check the configuration

After installing Nagios and Nagios plugins, run the following command to check the configuration of Nagios:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Info

If you installed Nagios in a non-default folder, for example, in the /data folder, execute the below command: /data/nagios/bin/nagios -v /data/nagios/etc/nagios.cfg

and make sure there is no error like in the image below:

```
[root@RockyLinux9 nagios-plugins-master]# sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.9
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-12-19
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@RockyLinux9 nagios-plugins-master]#
```

Check the Nagios configuration

9. Turn on the services

Turn on the services using the commands below:

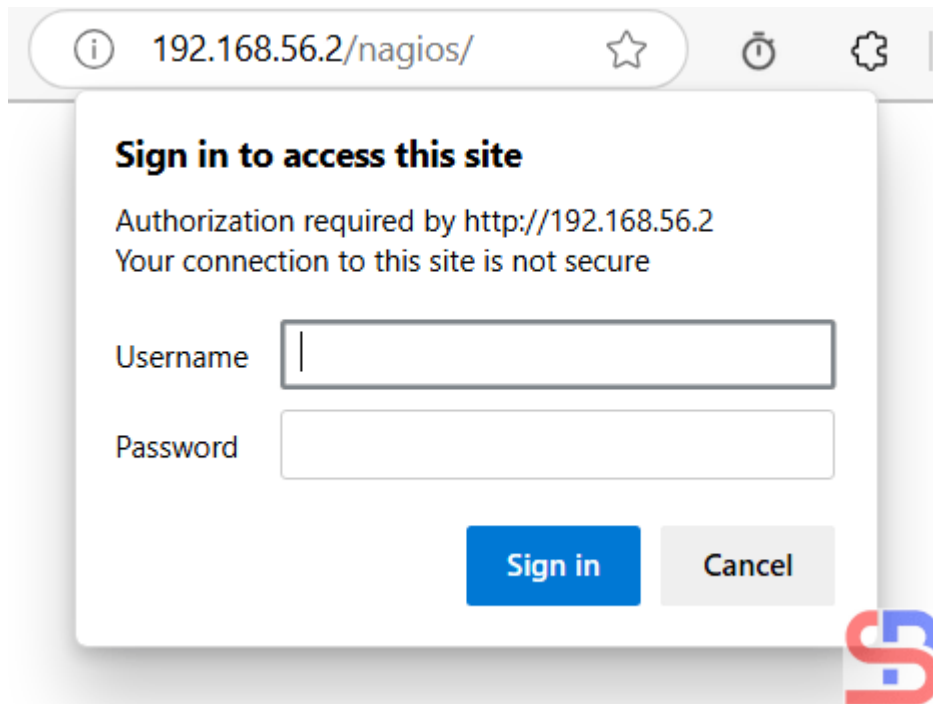
```
cp /lib/systemd/system/nagios.service /etc/systemd/system/
systemctl start httpd
systemctl start nagios
systemctl enable httpd
systemctl enable nagios
```

10. Check the application

Open your browser, and type in your browser:

`http://your_ip_address_server/nagios`

And there should be a display like the image below:

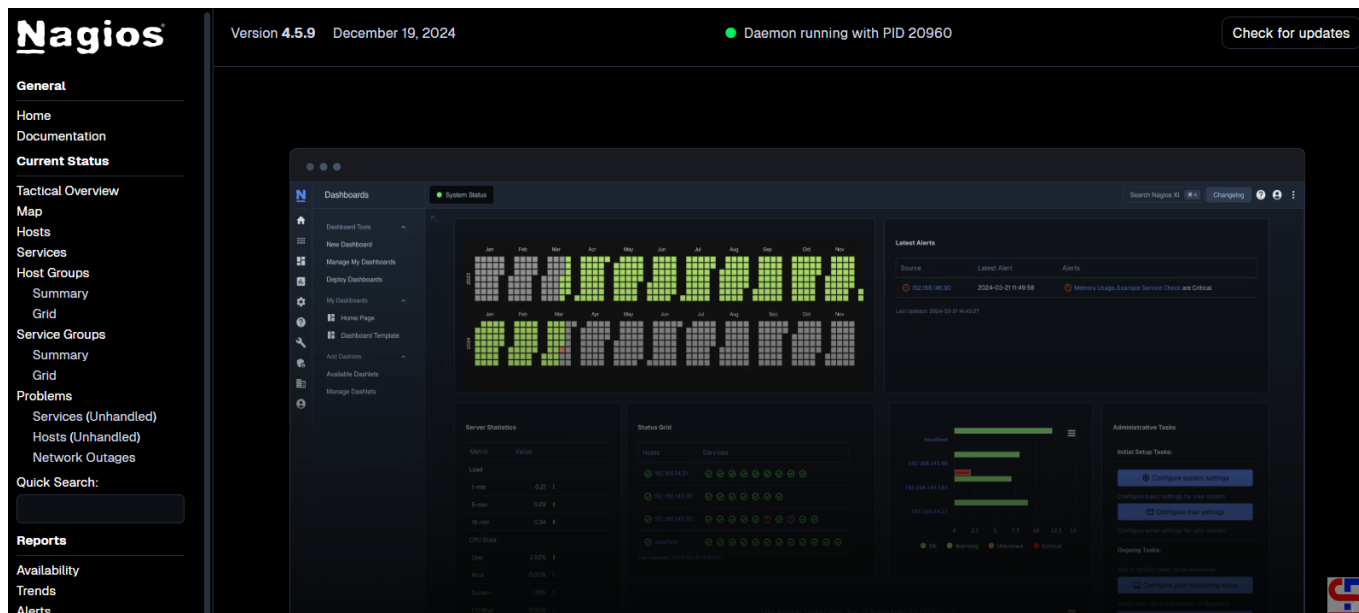


Open the Nagios application

If you don't see the image like the above image in your browser, maybe the Firewall/IPTables is still on in your server. Run the following commands:

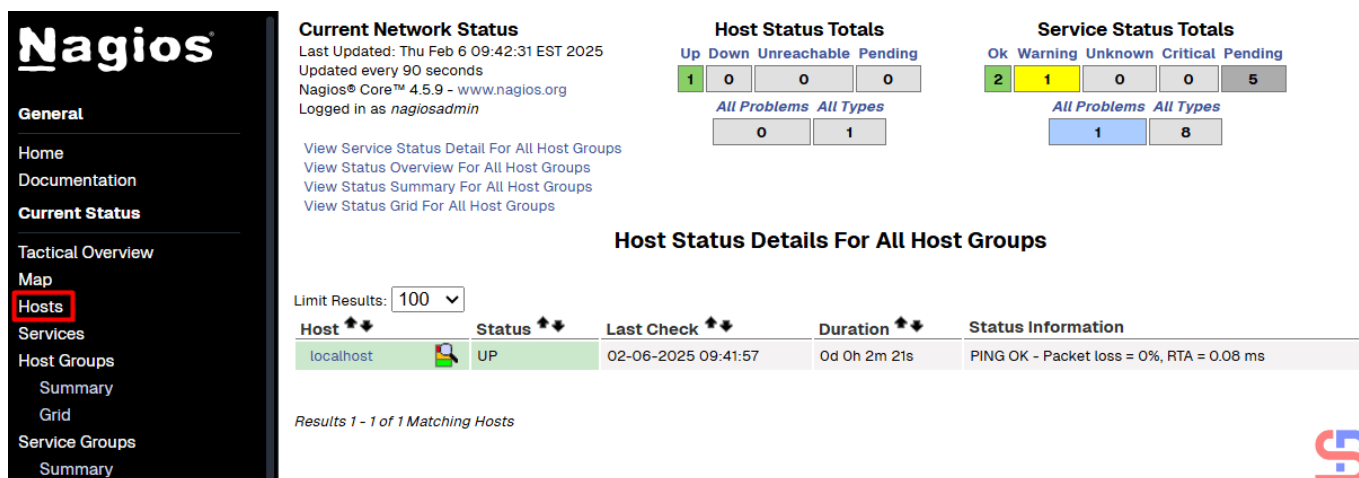
```
firewall-cmd --zone=public --add-port=80/tcp --permanent
firewall-cmd --reload
sed -i 's/SELINUX=.* /SELINUX=disabled/g' /etc/selinux/config
setenforce 0
```

Back to your browser again, and it should work now. Insert the username (**nagiosadmin**) and the password for Nagios. If the username and the password are right, the Nagios application will appear like this:



Open the Nagios application

If you want to know which hosts are being monitored by Nagios, click **Hosts**. Nagios will display the hosts that are being monitored:



Hosts monitored by Nagios

From the picture above, it can be seen that currently, Nagios is only monitoring the Nagios server or localhost. If you want to know which services are being monitored by Nagios, click **Services**. Nagios will display the services that are being monitored:

Nagios

General

- Home
- Documentation
- Current Status**
- Tactical Overview
- Map
- Hosts
- Services**
- Host Groups
 - Summary
 - Grid
- Service Groups
 - Summary
 - Grid
- Problems
 - Services (Unhandled)
 - Hosts (Unhandled)

Current Network Status
 Last Updated: Thu Feb 6 09:46:10 EST 2025
 Updated every 90 seconds
 Nagios® Core™ 4.5.9 - www.nagios.org
 Logged in as nagiosadmin

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
7	1	0	0	0

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	02-06-2025 09:45:42	0d 0h 5m 28s	1/4	OK - load average: 0.00, 0.26, 0.35
	Current Users	OK	02-06-2025 09:41:20	0d 0h 6m 6s+	1/4	USERS OK - 1 users currently logged in
	HTTP	WARNING	02-06-2025 09:44:57	0d 0h 1m 13s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 7897 bytes in 0.001 second response time
	PING	OK	02-06-2025 09:42:35	0d 0h 6m 6s+	1/4	PING OK - Packet loss = 0%, RTA = 0.11 ms
	Root Partition	OK	02-06-2025 09:43:12	0d 0h 6m 6s+	1/4	DISK OK - free space: / 15158 MiB (87.15% inode=99%):
	SSH	OK	02-06-2025 09:43:50	0d 0h 6m 6s+	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	OK	02-06-2025 09:44:27	0d 0h 6m 6s+	1/4	SWAP OK - 100% free (2047 MB out of 2047 MB)
	Total Processes	OK	02-06-2025 09:45:05	0d 0h 6m 6s+	1/4	PROCS OK: 35 processes with STATE = RSZDT

Results 1 - 8 of 8 Matching Services

Services monitored by Nagios

From the picture above, you can see that Nagios monitored 8 services for the Nagios server or localhost.

Note

If you have a domain/subdomain and want to use that domain/subdomain for the Nagios application, create a virtual host on your web server. For example, I have the domain sysadminpedia.com and want to use the subdomain nagios.sysadminpedia.com for the Nagios application. So, I created the script below in the file

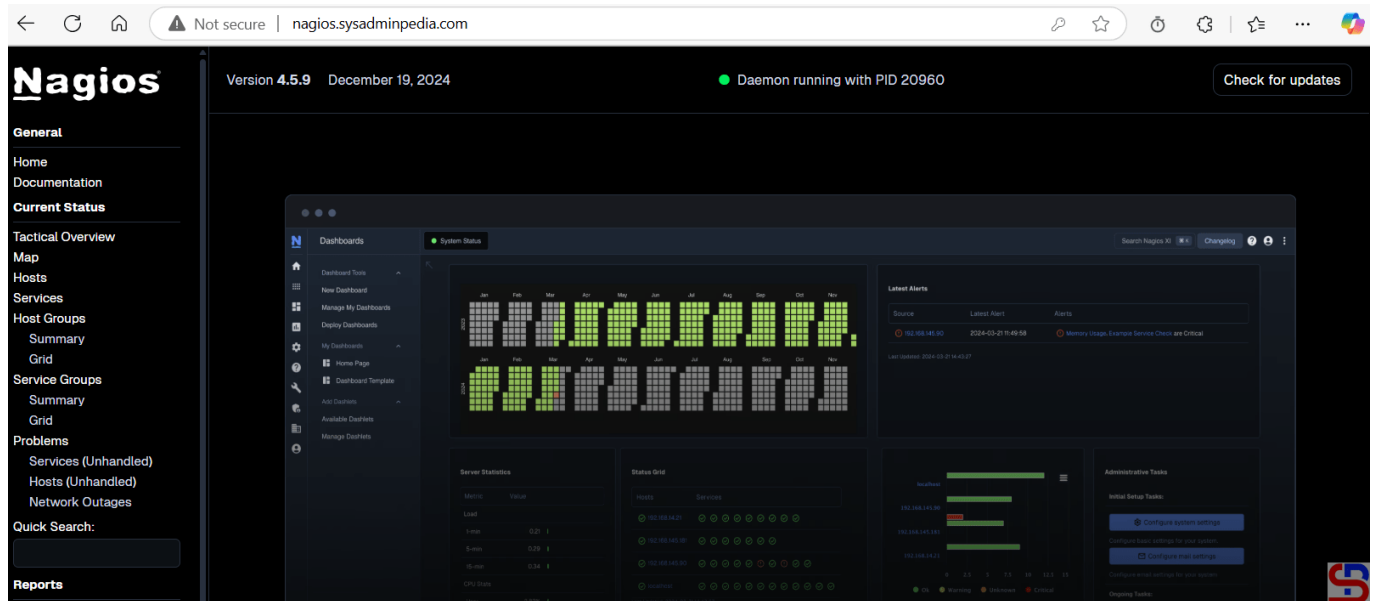
/etc/httpd/conf.d/nagios.sysadminpedia.com.conf:

```
<VirtualHost *:80>
    ServerName nagios.sysadminpedia.com
    ServerAdmin sysadmin@nagios.sysadminpedia.com
    DocumentRoot /usr/local/nagios/share
    <Directory /usr/local/nagios/share>
        Options -Indexes +FollowSymLinks
        AllowOverride All
    </Directory>

    ErrorLog /var/log/httpd/nagios.sysadminpedia.com-error.log
    CustomLog /var/log/httpd/nagios.sysadminpedia.com-access.log combined
</VirtualHost>
```

Restart the web server, open your browser, and type your domain/subdomain for Nagios, and it should be like the image

below:



Using a domain or a subdomain for the Nagios application

Info

If you installed Nagios in a non-default folder, for example, in the /data folder, you can copy the script above, but you must change the word /usr/local to /data.

References

support.nagios.com
tecmint.com
statusengine.org

How to Make a Linux User Have the sudo Function?

written by sysadmin | 21 June 2025

SUDO stands for “**SuperUser DO**” and it is a program for Unix-like computer operating systems that enables users to run programs with the security privileges of another user, by default, the superuser. With sudo, a normal user can install or delete an application, change the server network, or even

reboot or shut down the server.

Problem

How to make a Linux user have the sudo function?

Solution

This article will explain how to make a Linux user have the sudo function on RockyLinux/AlmaLinux/CentOS, Ubuntu/Debian, and OpenSUSE distros. For example, you want to add the user john to these distros and want that user to be able to use the sudo function. As far as I know, there are two methods to do it:

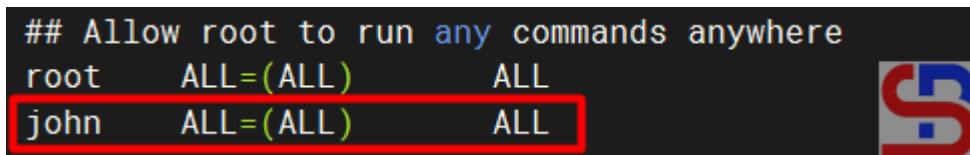
1. Change the sudoers file

Open the /etc/sudoers file or use the command below:

```
visudo
```

Add to the file the user name as in the image below:

```
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
john    ALL=(ALL)    ALL
```



Add the user in the sudoers file

After that, save the file and then try to add a new user using the user john, if there is a display like the image below:

```
[john@RockyLinux9 ~]$ sudo systemctl restart sshd
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for john:
```

```
[john@RockyLinux9 ~]$
```



Choose number 1

Then select number 1, and the user should successfully add a new user as in the image above.

2. Add the user to the sudo group

Add the user to the sudo group, where the name of this sudo group can vary in each distro. To see the name of the sudo group, look in the sudoers file and look for a sentence similar to '**Allows people in group to execute any command**'. For example, in RockyLinux and OpenSUSE, the name of the sudo group is **wheel**, **sudo** in Ubuntu, and don't forget to make sure to uncomment the section as in the image below:

```
## Allows people in group wheel to run all commands  
%wheel  ALL=(ALL)      ALL
```



Check the sudo group in the sudoers file

Then type the command below so that a user can use sudo:

RockyLinux & OpenSUSE

```
usermod -aG wheel john
```

```
[root@RockyLinux9 ~]# usermod -aG wheel john
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# su - john
Last login: Wed Jan 15 05:51:59 EST 2025 on pts/0
[john@RockyLinux9 ~]$ sudo adduser edward
[sudo] password for john:
[john@RockyLinux9 ~]$
```



Add the user to the sudo group

Ubuntu/Debian

```
usermod -aG sudo john
```

Note

The two methods above can provide the sudo feature to a user on Linux so that the user can run commands that can only be executed by root if the user uses the sudo command by writing down the password. However, if you want the bob user not to have to enter a password when running the sudo command, then in the sudoers file, type the script below:

```
bob                ALL=(ALL)        NOPASSWD: ALL
```

Use the command below if you want the robin user to only be able to perform reboot commands using sudo, but not other commands using sudo:

```
robin              ALL=(ALL)        /usr/sbin/reboot
```

```
[robin@RockyLinux9 ~]$ sudo systemctl restart sshd
[sudo] password for robin:
Sorry, user robin is not allowed to execute '/bin/systemctl restart sshd' as root on RockyLinux9.
[robin@RockyLinux9 ~]$
```



Give the partial sudo function to the user

References

en.wikipedia.org
askubuntu.com
phoenixnap.com

How to Open And Close a Port on RockyLinux Server?

written by sysadmin | 21 June 2025

By default, the RockyLinux/AlmaLinux/CentOS distro provides two firewalls, iptables and firewalld. This article will explain how to open and close a port using Firewalld on the distro. If you have opened and closed a port using Firewalld, you don't need to open and close a port in iptables.

Problem

How to open and close a port on the RockyLinux server?

Solution

A. Check the Firewalld status

By default, the Firewalld package is installed automatically using the command:

```
systemctl status firewalld
```

```
[root@RockyLinux9 ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-01-10 02:17:26 EST; 52min ago
     Docs: man:firewalld(1)
    Main PID: 650 (firewalld)
      Tasks: 2 (limit: 4672)
     Memory: 42.4M
        CPU: 3.490s
    CGroup: /system.slice/firewalld.service
            └─650 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

Jan 10 02:17:19 RockyLinux9 systemd[1]: Starting firewalld - dynamic firewall daemon...
Jan 10 02:17:26 RockyLinux9 systemd[1]: Started firewalld - dynamic firewall daemon.
[root@RockyLinux9 ~]#
```



Check the status of Firewalld

From the picture above, you can see that the firewall on the server is already running. If the Firewalld is not already running, use the command below:

```
systemctl enable --now firewalld
```

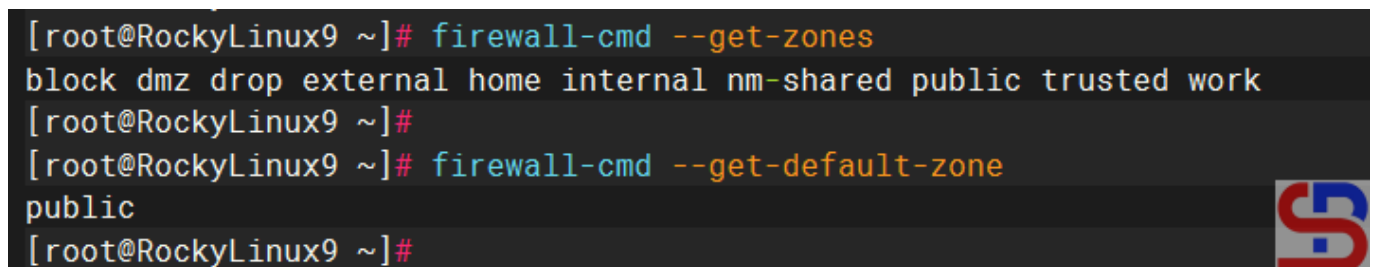
But if on your server there is no firewall package, you can install it using the command below:

```
yum install -y firewalld
```

B. Check the zones

Firewalld uses zones and services, compared to iptables, which use chains and rules. Zones are a collection of rules that have been set for what network connections should be permitted based on the level of confidence in the network connected to the system. We can determine the name of the network interface and the network source into zones. To see the zones in firewalld and which zone is the default, use the command below:

```
firewall-cmd --get-zones
firewall-cmd --get-default-zone
```

A terminal window screenshot from a RockyLinux9 system. The prompt is [root@RockyLinux9 ~]#. The first command is firewall-cmd --get-zones, which outputs: block dmz drop external home internal nm-shared public trusted work. The second command is firewall-cmd --get-default-zone, which outputs: public. The terminal has a dark background with light-colored text. A small logo is visible in the bottom right corner of the terminal window.

```
[root@RockyLinux9 ~]# firewall-cmd --get-zones
block dmz drop external home internal nm-shared public trusted work
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# firewall-cmd --get-default-zone
public
[root@RockyLinux9 ~]#
```

Show all zones in Firewalld

From the picture above, there are 9 zones, and the explanation can be seen in the picture below, which is sorted from the most trusted

Zone Name	Description
Trusted	This zone accepts all the incoming traffic. You can use this zone to manage the traffic on a trusted network because it will not filter anything.
Home	This zone is designed for only the home network. It permits only selected incoming traffic and reject all.
Work	This zone designed for only the work (corporate) networking. It permits only selected incoming traffic and reject all.
Internal	This zone intended to design for the internal network. It permits only what is allowed and rejects all.
Public	This zone rejects all the incoming traffic, except what is granted. Using with the default zone, we can add any newly network interfaces on it. It is designed to use only the public places.
External	This zone designed for outgoing traffic forwarded with masquerading is enabled. Also, we can use this for NAT
Dmz	This zone designed to use the demilitarized zone with limited public access. It permits only selected incoming traffic and reject all.
Block	This zone designed to reject all incoming traffic with an ICMP-host-prohibited message is returned. It permits only outgoing traffic.
Drop	This zone designed to drop all incoming traffic with no notification like ICMP errors. It is purely used in high secure places.

The zones in Firewallld (Image credit for linuxteck.com)

To view all settings for all zones, use the following command:

```
firewall-cmd --list-all-zones
```

```
[root@RockyLinux9 ~]# firewall-cmd --list-all-zones
```

```
block
```

```
target: %%REJECT%%  
icmp-block-inversion: no  
interfaces:  
sources:  
services:  
ports:  
protocols:  
forward: yes  
masquerade: no  
forward-ports:  
source-ports:  
icmp-blocks:  
rich rules:
```

```
dmz
```

```
target: default  
icmp-block-inversion: no  
interfaces:  
sources:  
services: ssh  
ports:  
protocols:  
forward: yes  
masquerade: no  
forward-ports:  
source-ports:  
icmp-blocks:  
rich rules:
```



View all the settings in Firewallld

But, if you want to view all settings in a specific zone, for example, a public zone, use the following command:

```
firewall-cmd --zone=public --list-ports
```

C. Open the Port

Now, if you want to open port 43210 with TCP protocol, use the command below:

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

```
[root@RockyLinux9 ~]# firewall-cmd --add-port=43210/tcp --permanent
success
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# firewall-cmd --reload
success
[root@RockyLinux9 ~]#
```



Open the port

Use the command below to see the ports that have been opened:

```
firewall-cmd --list-ports
```

```
[root@RockyLinux9 ~]# firewall-cmd --list-ports
43210/tcp
[root@RockyLinux9 ~]#
```



List all opened ports

D. Open the port from a certain IP

If you want to open a port from a certain IP, for example, you only allow IP 192.168.56.100 to access port 22 on this server, then use the command below:

```
firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 source
address=192.168.56.100 port port=22 protocol=tcp accept'
firewall-cmd --reload
firewall-cmd --list-rich-rules
```

```
[root@RockyLinux9 ~]# sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="192.168.56.100" port port="22" protocol="tcp" accept'
success
[root@RockyLinux9 ~]# firewall-cmd --reload
success
[root@RockyLinux9 ~]# firewall-cmd --list-rich-rules
rule family="ipv4" source address="192.168.56.100" port port="22" protocol="tcp" accept
[root@RockyLinux9 ~]#
```



Allow the IP to a certain port

If you want to reject a host with IP 192.168.56.100 to access port 22, use the command below:

```
sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source
address="192.168.56.100" port port="22" protocol="tcp" reject'
firewall-cmd --reload
```

firewall-cmd --list-rich-rules

```
[root@RockyLinux9 ~]# sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="192.168.56.100" port port="22" protocol="tcp" reject'
success
[root@RockyLinux9 ~]# firewall-cmd --reload
success
[root@RockyLinux9 ~]# firewall-cmd --list-rich-rules
rule family="ipv4" source address="192.168.56.100" port port="22" protocol="tcp" reject
[root@RockyLinux9 ~]#
```

Block the IP to a certain port

E. Close the port from a certain IP

If you want to close a port from a certain IP, for example, you block a host with IP 192.168.56.100 from accessing port 22 on this server, then use the command below:

```
sudo firewall-cmd --permanent --remove-rich-rule='rule family="ipv4" source address="192.168.56.100" port port="22" protocol="tcp" accept'
firewall-cmd --reload
firewall-cmd --list-rich-rules
```

```
[root@RockyLinux9 ~]# firewall-cmd --list-rich-rules
rule family="ipv4" source address="192.168.56.100" port port="22" protocol="tcp" accept
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# sudo firewall-cmd --permanent --remove-rich-rule='rule family="ipv4" source address="192.168.56.100" port port="22" protocol="tcp" accept'
success
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# firewall-cmd --reload
success
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# firewall-cmd --list-rich-rules
[root@RockyLinux9 ~]#
```

Remove the IP to a certain port

INFO

In short, if you want to delete the rich rule, then change the option `--add-rich-rule` to `--remove-rich-rule`.

F. Close the port

Use the command below to close the newly opened port 43210:

```
firewall-cmd --remove-port=43210/tcp --permanent
firewall-cmd --reload
```

```

[root@RockyLinux9 ~]# firewall-cmd --list-ports
43210/tcp
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# firewall-cmd --remove-port=43210/tcp --permanent
success
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# firewall-cmd --reload
success
[root@RockyLinux9 ~]#
[root@RockyLinux9 ~]# firewall-cmd --list-ports

```



Close the port in Firewallld

G. Open the service

Apart from using ports, Firewallld can also open and close services on the server. To see the services that have been opened, type the command below:

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

```

[root@RockyLinux9 ~]# firewall-cmd --list-services
cockpit dhcpv6-client ssh
[root@RockyLinux9 ~]#

```



List all opened services

You can see in the picture above that the distro only opens 3 services. If you want to open the SMTP service, use the command below:

```

firewall-cmd --add-service=smtp --permanent
firewall-cmd --reload

```

```

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

```

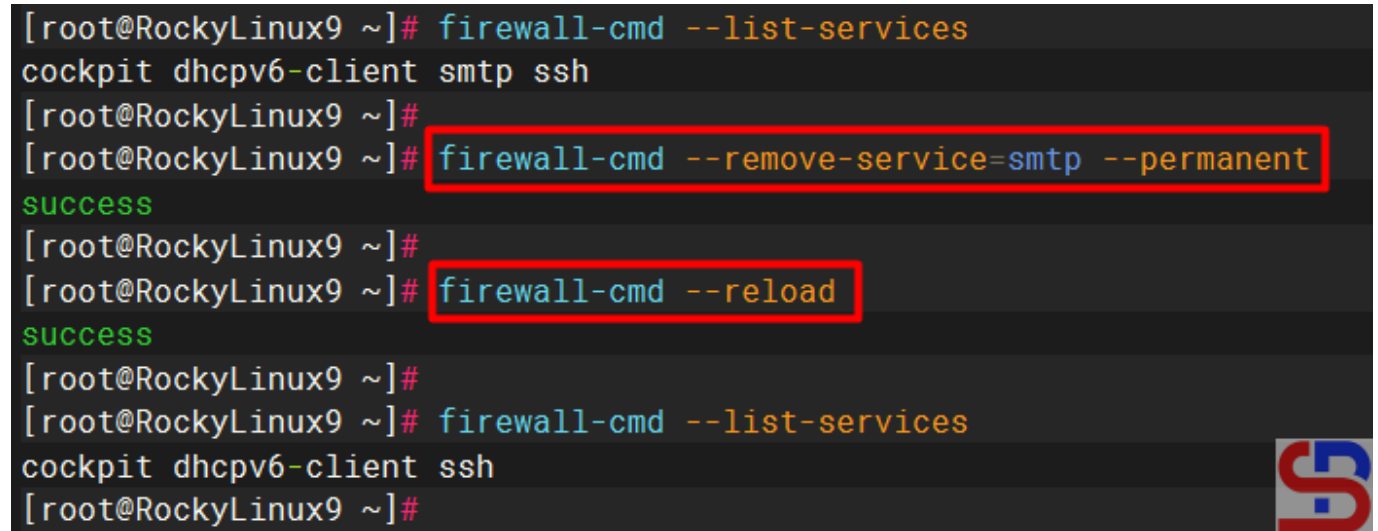


Add the service to the firewall

H. Close the service

To delete the SMTP service in Firewalld, use the command below:

```
firewall-cmd --remove-service=smtp --permanent
firewall-cmd --reload
```

A terminal window screenshot from RockyLinux9. The prompt is [root@RockyLinux9 ~]#. The first command is firewall-cmd --list-services, which outputs cockpit dhcpv6-client smtp ssh. The second command is firewall-cmd --remove-service=smtp --permanent, which is highlighted with a red box and outputs success. The third command is firewall-cmd --reload, which is also highlighted with a red box and outputs success. The fourth command is firewall-cmd --list-services, which outputs cockpit dhcpv6-client ssh. The prompt is [root@RockyLinux9 ~]# at the end. A small logo is visible in the bottom right corner of the terminal window.

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

Close the service in Firewalld

Note

If you use the OpenSUSE distro, you can use the above commands to open and close a port, like in the image below:

```
opensuse15:~ # systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-01-10 06:14:05 EST; 5min ago
     Docs: man:firewalld(1)
   Main PID: 833 (firewalld)
    Tasks: 2 (limit: 1125)
     CPU: 23.153s
   CGroup: /system.slice/firewalld.service
           └─833 /usr/bin/python3 /usr/sbin/firewalld --nofork --nopid

Jan 10 06:13:57 opensuse15 systemd[1]: Starting firewalld - dynamic firewall daemon...
Jan 10 06:14:05 opensuse15 systemd[1]: Started firewalld - dynamic firewall daemon.
opensuse15:~ #
opensuse15:~ # firewall-cmd --add-port=43210/tcp --permanent
success
opensuse15:~ #
opensuse15:~ # firewall-cmd --reload
success
opensuse15:~ #
opensuse15:~ # firewall-cmd --list-ports
43210/tcp
opensuse15:~ #
opensuse15:~ # firewall-cmd --remove-port=43210/tcp --permanent
success
opensuse15:~ #
opensuse15:~ # firewall-cmd --reload
success
opensuse15:~ #
opensuse15:~ # firewall-cmd --list-ports
opensuse15:~ #
```



The Firewalld commands in OpenSUSE

References

[redhat.com](https://www.redhat.com)
[greenwebpage.com](https://www.greenwebpage.com)
[inmotionhosting.com](https://www.inmotionhosting.com)
[baeldung.com](https://www.baeldung.com)
[musaamin.web.id](https://www.musaamin.web.id)

[How to Install gcloud on RockyLinux?](#)

written by sysadmin | 21 June 2025

If you use GCP in daily operations, it is recommended to use

the commands in the CLI known as gcloud. This is because many commands can only be executed using gcloud rather than using the Console in the browser.

Problem

How to install gcloud on RockyLinux?

Solution

Before you access GCP and run GCP commands through your server, you must first install gcloud on your server.

A. Install gcloud

As far as I know, there are 2 methods to install gcloud on RockyLinux/AlmaLinux/CentOS, and both methods recommend using a user other than root.

1. Using the script

Before you download the script, install the packages using the command below:

```
yum install tar curl
```

Use the command below to download and install the script:

```
curl https://sdk.cloud.google.com | bash
```

Then you will see a display like the one below:

```
[root@RockyLinux9 ~]# curl https://sdk.cloud.google.com | bash
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 443 100 443    0     0  938      0 --:--:-- --:--:-- --:--:--  932
Downloading Google Cloud SDK install script: https://dl.google.com/dl/cloudsdk/channels/rapid/install_google_cloud_sdk_bash
##### 100.0%
Running install script from: /tmp/tmp.75bU1NQIeX/install_google_cloud_sdk_bash
which curl
curl -# -f https://dl.google.com/dl/cloudsdk/channels/rapid/google-cloud-sdk.tar.gz
##### 100.0%

Installation directory (this will create a google-cloud-sdk subdirectory) (/root):
mkdir -p /root
tar -C /root -zxvf /tmp/tmp.aRGolzrmtE/google-cloud-sdk.tar.gz
google-cloud-sdk/.install/.download/
google-cloud-sdk/.install/core.manifest
google-cloud-sdk/.install/core.snapshot.json
google-cloud-sdk/.install/gcloud-deps.manifest
```

Install gcloud using the script

Wait until it's finished, and you will see a display like the one below:

```
Modify profile to update your $PATH and enable shell command completion?
Do you want to continue (Y/n)? Y

The Google Cloud SDK installer will now prompt you to update an rc file to bring the Google Cloud CLIs into your environment.

Enter a path to an rc file to update, or leave blank to use [/home/sysadmin/.bashrc]:
Backing up [/home/sysadmin/.bashrc] to [/home/sysadmin/.bashrc.backup].
[/home/sysadmin/.bashrc] has been updated.

==> Start a new shell for the changes to take effect.

For more information on how to get started, please visit:
https://cloud.google.com/sdk/docs/quickstarts

[sysadmin@RockyLinux9 ~]$
```

Installation complete

From the image above, you are asked to create a new SSH connection so that the effect can be seen, and type the command below:

```
gcloud version
```

However, you can use the command below:

```
source /home/sysadmin/.bashrc
```

So you don't need to create a new SSH connection to run the gcloud version command, which results in the image below:

```
Modify profile to update your $PATH and enable shell command completion?

Do you want to continue (Y/n)? Y

The Google Cloud SDK installer will now prompt you to update an rc file to bring the Google Cloud CLIs into your environment.

Enter a path to an rc file to update, or leave blank to use [/home/sysadmin/.bashrc]:
Backing up [/home/sysadmin/.bashrc] to [/home/sysadmin/.bashrc.backup].
[/home/sysadmin/.bashrc] has been updated.

==> Start a new shell for the changes to take effect.

For more information on how to get started, please visit:
https://cloud.google.com/sdk/docs/quickstarts

[sysadmin@RockyLinux9 ~]$ source /home/sysadmin/.bashrc
[sysadmin@RockyLinux9 ~]$
[sysadmin@RockyLinux9 ~]$ gcloud version
Google Cloud SDK 504.0.1
bq 2.1.11
bundled-python3-unix 3.11.9
core 2024.12.19
gcloud-crc32c 1.0.0
gsutil 5.33
[sysadmin@RockyLinux9 ~]$
```

Check the result of the installation

2. Using the Repository

You have to add the Google Cloud SDK repository to your server using the following command:

```
sudo tee -a /etc/yum.repos.d/google-cloud-sdk.repo << EOM
[google-cloud-cli]
name=Google Cloud CLI
baseurl=https://packages.cloud.google.com/yum/repos/cloud-sdk-el9-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=0
gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOM
```

After that, install gcloud using the command below:

```
yum install google-cloud-sdk
```

After the installation finishes, run the following command to test the gcloud command:

```
gcloud version
```

B. Connect to GCP

After you install gcloud on your server, type the command below:

```
gcloud init
```

Then there will be a display like the image below:

```
[sysadmin@RockyLinux9 ~]$ gcloud init
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

You must sign in to continue. Would you like to sign in (Y/n)? Y

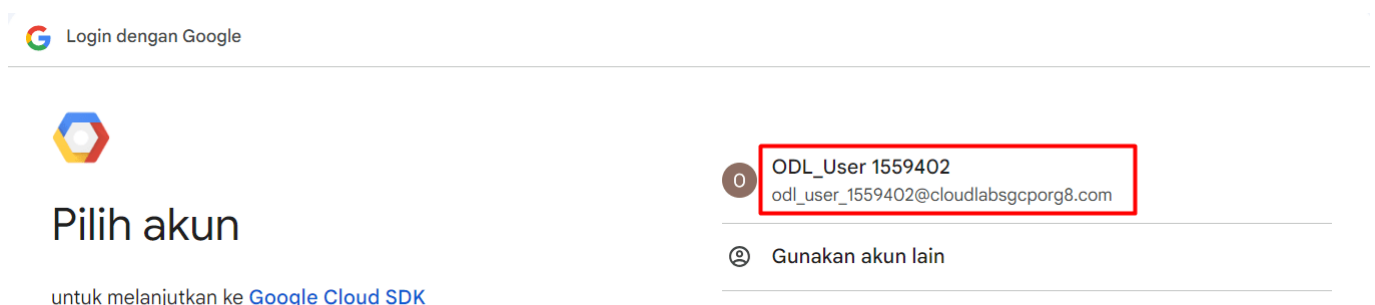
Go to the following link in your browser, and complete the sign-in prompts:

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Fsdsk.cloud.google.com%2Fauthcode.html&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=d2JSQAga1WPHQzFXNj5AaQnyXvU16&prompt=consent&token_usage=remote&access_type=offline&code_challenge=JA4vnnvBK9ZHCrJ9WQ240aHXUoszw91xkBHhNb1VN7Dw&code_challenge_method=S256

Once finished, enter the verification code provided in your browser: █
```

Click the link

Click the **Ctrl+Click** button in the red box to open the link in a browser, or if you have difficulty, copy what is in the red box and place it in your browser so you will see a display like the one below:



Click the account

Click on the Google account that will access GCP, then there will be a display like the image below:



Sign in to Google Cloud SDK

odl_user_1559402@cloudlabsgcporg8.com

By continuing, Google will share your name, email address, language preference, and profile picture with Google Cloud SDK. See Google Cloud SDK's Privacy Policy and Terms of Service.

You can manage Sign in with Google in your [Google Account](#).

Cancel

Continue



Click the Continue button

Click the **Continue** button, then the display below will appear:



Google Cloud SDK wants to access your Google Account

odl_user_1559402@cloudlabsgcporg8.com

This will allow **Google Cloud SDK** to:

- See, edit, configure, and delete your Google Cloud data and see the email address for your Google Account. ⓘ
- View and sign in to your Google Cloud SQL instances ⓘ
- View and manage your Google Compute Engine resources ⓘ
- View and manage your applications deployed on Google App Engine ⓘ

Make sure you trust Google Cloud SDK

[Learn why you're not seeing links to Google Cloud SDK's Privacy Policy or Terms of Service](#)

Review Google Cloud SDK's Privacy Policy and Terms of Service to understand how Google Cloud SDK will process and protect your data.

To make changes at any time, go to your [Google Account](#).

Learn how Google helps you [share data safely](#).

Cancel

Allow



Click the Allow button

Click the **Allow** button, then the display below will appear:



Sign in to the gcloud CLI

You are seeing this page because you ran the following command in the gcloud CLI from this or another machine. If this is not the case, close this tab.

```
gcloud auth login --no-launch-browser
```

Enter the following verification code in gcloud CLI on the machine you want to log into. This is a credential **similar to your password** and should not be shared with others.

```
4/0AanRRruchiESKnvxMD0H4Ds5LcSFkfAXgo5  
SwDxgHetI-Nftseo4ebZab4TwnivEeqjh9w
```

Copy

You can close this tab when you're done.



Click the Copy button

Click the **Copy** button, and paste it into the CLI on your server as in the image below:

```
Go to the following link in your browser, and complete the sign-in prompts:

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Fsdg.cloud.google.com%2Fauthcode.html&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo_email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine_admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice_login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts_reauth&state=d2JSQAga1WPHQzfXNJ5AaQnyXvU16&prompt=consent&token_usage=remote&access_type=offline&code_challenge=JA4vnyv8K9ZHCrJ9WQ248aHXUoszw91xkBIhNb1VN7Dw&code_challenge_method=S256

Once finished, enter the verification code provided in your browser: 4/0AanRRuchiESKnxvMD0H4Ds5LcSFkFAXgo5SwDxgHetI-Nftseo4ebZab4TwnivEeqjh9w
You are signed in as: [odl_user_1559402@cloudlabsgcporg8.com].

Pick cloud project to use:
[1] clgcporg8-0883
[2] Enter a project ID
[3] Create a new project
Please enter numeric choice or text value (must exactly match list item): 1

Your current project has been set to: [clgcporg8-0883].

Do you want to configure a default Compute Region and Zone? (Y/n)? Y

Which Google Compute Engine zone would you like to use as project default?
If you do not specify a zone via a command line flag while working with Compute Engine resources, the default is assumed.
[1] us-east1-b
[2] us-east1-c
```

Paste the code

Select the project and configure the zone as in the image above. After that, the gcloud configuration is complete.

C. Test gcloud

Now, try gcloud to access your GCP. I try to list my virtual machine in GCP using the below command:

```
gcloud compute instances list
```

Then the display below will appear:

```
[sysadmin@RockyLinux9 ~]$ gcloud compute instances list
NAME          ZONE          MACHINE_TYPE  PREEMPTIBLE  INTERNAL_IP  EXTERNAL_IP  STATUS
my-first-vm    us-west1-a    e2-medium     10.138.15.229 35.247.67.92  RUNNING
```

Display virtual machine in GCP using gcloud

If you get a display like the image above, you have successfully used your gcloud to access your GCP.

Note

If you have many projects on your GCP, you can choose one of these projects as the starting point for your gcloud on GCP. You can switch projects using the command:

```
gcloud config set project PROJECT_ID
```

Change **PROJECT_ID** to the project ID you want to switch to.

References

liquidweb.com

cloud.google.com

bacancytechnology.com