



How To Set Up the UFW Firewall on Linux

Published: 29 December 2021 – 7 min. read

COMMAND LINE

LINUX



Nicholas Xuan Nguyen

Read [more tutorials](#) by Nicholas Xuan Nguyen!



Your Job!

Your Company!

\$50,000 – \$100,000

**Get Started
Today!**

Table of Contents

Prerequisites

Installing UFW and Enabling IPv6 Connection

Configuring Default Policies for Firewall Rules

Allowing SSH Connections on the UFW Firewall

Allowing HTTP and HTTPS Connections

Allowing Connections from Specific Port Range and IP Address

Allowing Traffic from a Specific Network Interface

Deleting UFW Firewall Rules

Resetting the UFW Firewall

Looks like you're offline!

Without a firewall, there are no rules or restrictions on your network traffic and that leads to a number of negative consequences. Linux system comes with a default firewall configuration tool, which is Uncomplicated Firewall (UFW). But how do you set up a UFW firewall? Sit back and relax, this tutorial has got you covered!

In this tutorial, you'll learn how to configure UFW and set up a firewall on your Linux system to secure your network and ward-off malicious acts.

Ready? Read on to get started!

Prerequisites

This tutorial will be a hands-on demonstration. If you'd like to follow along, be sure you have the following:

- An Ubuntu machine – This tutorial uses Ubuntu 20.04 LTS, but other Linux distributions will work.

Related: [How to Install Ubuntu 20.04 \[Step-by-Step\]](#).

- Root privileges to your machine.

Looks like you're offline!

Even though UFW comes packaged with your Ubuntu system, UFW is not installed by default. Install UFW first with the `apt` package manager and configure it to allow connections over IPv6.

Related: [How to use the Ansible apt Module to Manage Linux Packages](#)

1. Open your terminal and run the `apt update` command below to update your local package index. The command accepts all prompts (`-y`) during the update for less user intervention.

```
sudo apt update -y
```

```
root@ubuntu:~# sudo apt update -y
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [344 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1,065 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [196 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/restricted i386 Packages [20.5 kB]
```

Looks like you're offline!

2. Next, run the below command to install UFW (`install ufw`) on your system while accepting all prompts (`-y`) during the installation.

```
sudo apt install ufw -y
```

```
root@ubuntu:~# sudo apt install ufw -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  iptables libip6tc2 libnetfilter-conntrack3 libnfnetlink0 libnftnl11
Suggested packages:
  firewallld nftables
The following NEW packages will be installed:
  iptables libip6tc2 libnetfilter-conntrack3 libnfnetlink0 libnftnl11 ufw
0 upgraded, 6 newly installed, 0 to remove and 141 not upgraded.
Need to get 670 kB of archives.
After this operation, 4,056 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libip6tc2 amd64 1.8.
-3ubuntu2 [19.2 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libnfnetlink0 amd64
0.1.3build1 [13.8 kB]
```

Installing UFW on Ubuntu

Looks like you're offline!

3. Open the UFW configuration file (*/etc/default/ufw*) with your favorite text editor. UFW supports IPv6, but you need to make sure that the firewall is configured to accept connections over IPv6.

If you only have IPv4 enabled, you're still leaving yourself open to IPv6 attacks.

4. Scroll down to the **IPV6** variable and set the value to **yes**, as shown below, then save the changes and exit the editor

Looks like you're offline!

```
# Set to yes to apply rules to support IPv6 (no means only IPv6 on loopback
# accepted). You will need to 'disable' and then 'enable' the firewall for
# the changes to take affect.
IPV6=yes

# Set the default input policy to ACCEPT, DROP, or REJECT. Please note that if
# you change this you will most likely want to adjust your rules.
DEFAULT_INPUT_POLICY="DROP"
```

Enabling IPV6 in the UFW Configuration File

5. Finally, run the command below to disable and re-enable UFW. The command restarts the UFW service so the changes can take effect.

After the command completes, your firewall can now write both IPv4 and IPv6 firewall rule sets.

```
sudo ufw disable && sudo ufw enable
```

Configuring Default Policies for Firewall Rules

If you're just getting started with UFW, it's recommended to set up a default policy for your rules. The default policies are applied to a chain that doesn't have any specific rules

Looks like you're offline!

Set up UFW to deny all incoming connections and allow all outgoing connections. As a result, anyone trying to reach your machine from the outside world is denied, while you can still freely connect to any website or server.

Run the `ufw` command below to deny all incoming connections by default.

```
sudo ufw default deny incoming
```

```
root@ubuntu:~# sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
```

Denying Incoming Network Traffics

Now run the following command to allow all outgoing connections by default.

```
sudo ufw default allow outgoing
```

```
root@ubuntu:~# sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

Allowing Outgoing Network Traffics

Looks like you're offline!

Allowing SSH Connections on the UFW Firewall

You've just set up default policies on your UFW firewall to deny all incoming traffic, and the "allow all-deny all" rule is a good setting for a regular user. But what if you're running a server? You'll need to allow specific traffic in and out. Allowing SSH connection on your UFW firewall will do the trick to allow specific traffic in and out.

Related: [A Windows Guy in a Linux World: Setting up SSH in Linux](#)

Looks like you're offline!

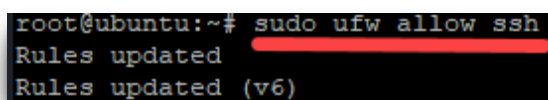
You'll set up an SSH server that allows incoming SSH connections on port 22. But why port 22 and not any other port? On Unix-like systems, the SSH daemon listens on port 22 by default, so it's a good practice to use the default SSH port to make your life a bit easier.

1. Run the below commands to install the OpenSSH server (`install openssh-server`) on your system and start an OpenSSH server (`start ssh`).

```
sudo apt install openssh-server -y
sudo systemctl start ssh
```

2. Now run the command below to allow incoming SSH connections. Without specifying port 22 will be enough as UFW knows what port is for SSH.

```
sudo ufw allow ssh
```



```
root@ubuntu:~# sudo ufw allow ssh
Rules updated
Rules updated (v6)
```

Allowing SSH connection

Looks like you're offline!

the file on your text editor, scroll down to `ssh` and see the port number (22) is part of the service description, as shown below.

```
ftp      21/tcp
fsp      21/udp      fsp
ssh      22/tcp      # SSH Remote Login Protocol
telnet   23/tcp
smtp     25/tcp      mail
time     37/tcp      timserver
time     37/udp      timserver
whois    43/tcp      nicname
tacacs   49/tcp      # Login Host Protocol (TACACS)
tacacs   49/udp
domain   53/tcp      # Domain Name Server
domain   53/udp
bootps   67/udp
```

Previewing the `/etc/services` file

But perhaps you prefer to specify the port number (22) to allow for SSH. If so, run the following command instead.

```
sudo ufw allow 22
```

3. Now run the below command to enable UFW.

Looks like you're offline!

Type Y in the confirmation prompt, as shown below, and press Enter to continue running the command. UFW will now start filtering packets on your system.

```
root@ubuntu:~# sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? Y
Firewall is active and enabled on system startup
```

Enabling UFW

4. Finally, run either of the below commands to check the status of your UFW firewall.

```
## Displays more detailed information, such as the interface and
## the packet's current progress
sudo ufw status verbose
## Shows each rule with a number and the corresponding allow or deny sta
## The numbered mode is useful when you are trying to delete a rule set
sudo ufw status numbered
```

If you run the command with the `verbose` option, you'll see an output similar to the one below:

- **Status: active** – Indicates the firewall is currently running.

Looks like you're offline!

- **Default: deny (incoming), allow (outgoing), disabled (routed)** – Indicates that the default policy is to deny all incoming connections and allow all outgoing connections.
- **New profiles: skip** – Indicates the firewall is currently using the default set of rules.

```
root@ubuntu:~# sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
22/tcp ALLOW IN Anywhere
22/tcp (v6) ALLOW IN Anywhere (v6)
```

Checking verbose UFW firewall status

If you run the command with the `numbered` option instead, you'll see the output below. You can see a list of numbered rules and their corresponding **ALLOW** or **DENY** status.

```
root@ubuntu:~# sudo ufw status numbered
Status: active

To Action From
--
[ 1] 22/tcp ALLOW IN Anywhere
[ 2] 22/tcp (v6) ALLOW IN Anywhere (v6)
```

Looks like you're offline!

Allowing HTTP and HTTPS Connections

At this point, you've only allowed SSH connections on your UFW firewall, but that limits your server's capabilities. Allow other types of connections, such as HTTP or HTTPS, and add more rules to the UFW firewall.

Run either of the following commands to allow incoming HTTP connections.

```
## HTTP connection uses port 80 (not secure)
sudo ufw allow 80
sudo ufw allow http
```

```
root@ubuntu:~# sudo ufw allow 80
Rule added
Rule added (v6)
root@ubuntu:~# sudo ufw allow http
Rule added
Rule added (v6)
root@ubuntu:~# █
```

Allowing HTTP connections

Now, run either of the commands below to allow incoming HTTPS connections.

Looks like you're offline!

```
## HTTP connection uses port 443 (secure)
sudo ufw allow 443
```

```
root@ubuntu:~# sudo ufw allow https
Rule added
Rule added (v6)
root@ubuntu:~# sudo ufw allow 443
Rule added
Rule added (v6)
root@ubuntu:~#
```

Allowing incoming HTTPS connections.

Allowing Connections from Specific Port Range and IP Address

Some applications use multiple ports in order to provide their services. And perhaps you have a range of ports to open or you need to allow connection from a specific IP address. In that case, add more UFW firewall rules.

Run the commands below to allow incoming connections on ports 5001 to 5009. You always should specify the protocol (`tcp` or `udp`) after the port range that the rules apply to because not all ports are used by both protocols.

Looks like you're offline!

```
sudo ufw allow 5001:5010/tcp
sudo ufw allow 5001:5010/udp
```

```
root@ubuntu:~# sudo ufw allow 5001:5010/tcp
Rule added
Rule added (v6)
root@ubuntu:~# sudo ufw allow 5001:5010/udp
Rule added
Rule added (v6)
```

Allowing traffic on 5001:5010 port range

Run the below command instead if you prefer to allow SSH connections from a specific IP address. The command allows SSH connections (port 22) only from the 192.168.1.2 IP address.

```
sudo ufw allow from 192.168.1.2 to any port 22
```

```
root@ubuntu:~# sudo ufw allow from 192.168.1.2 to any port 22
Rule added
root@ubuntu:~#
```

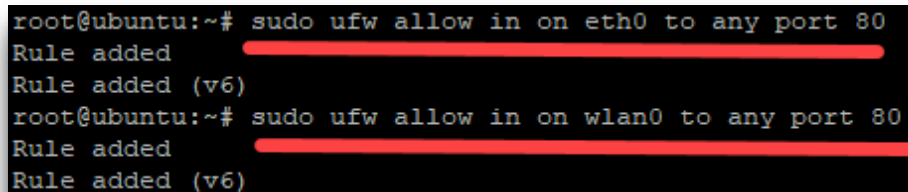
Allowing SSH Connections from Specific IP Address

Looks like you're offline!

UFW also lets you allow traffic on a specific network interface only, such as `eth0` is the first Ethernet interface and `wlan0` is the first Wi-Fi interface.

Run either of the commands below to allow HTTP connections only on the `eth0` and `wlan0` interfaces.

```
## Allow HTTP connection only on the eth0 interface
sudo ufw allow in on eth0 to any port 80
## Allow HTTP connection only on the wlan0 interface
sudo ufw allow in on wlan0 to any port 80
```

A terminal window with a black background and white text. It shows two commands being executed to allow HTTP traffic on specific interfaces. The first command is 'sudo ufw allow in on eth0 to any port 80', followed by two lines of output: 'Rule added' and 'Rule added (v6)'. The second command is 'sudo ufw allow in on wlan0 to any port 80', followed by two lines of output: 'Rule added' and 'Rule added (v6)'. A green cursor is visible at the end of the second output line.

```
root@ubuntu:~# sudo ufw allow in on eth0 to any port 80
Rule added
Rule added (v6)
root@ubuntu:~# sudo ufw allow in on wlan0 to any port 80
Rule added
Rule added (v6)
```

Allowing traffic on a specific interface

Deleting UFW Firewall Rules

Looks like you're offline!

number or the name of the rule to delete.

1. Run the below command to get a numbered list of the rules added to UFW.

```
sudo ufw status numbered
```

Note the rule's number or name in the output, like the one below.

```
root@ubuntu:~# sudo ufw status numbered
Status: active

      To      Action      From
      --      -
[ 1] 22/tcp    ALLOW IN    Anywhere
[ 2] 80/tcp    ALLOW IN    Anywhere
[ 3] 443/tcp   ALLOW IN    Anywhere
[ 4] 5001:5010/tcp ALLOW IN    Anywhere
[ 5] 5001:5010/udp ALLOW IN    Anywhere
[ 6] 22        ALLOW IN    192.168.1.2
[ 7] 80 on eth0 ALLOW IN    Anywhere
```

Previewing all the rules

2. Next, run the command below to delete rule number 4, which is the 5001:5010/tcp port range.

Looks like you're offline!

```
root@ubuntu:~# sudo ufw delete 4
Deleting:
allow 5001:5010/tcp
Proceed with operation (y|n)? y
Rule deleted
```

Deleting a Rule by Rule Number

3. Run the below command to **delete** a rule by its actual name with the **allow** status. In this example, you would delete the **http** rule by running the following command.

```
sudo ufw delete allow http
```

```
root@ubuntu:~# sudo ufw delete allow http
Rule deleted
Rule deleted (v6)
root@ubuntu:~#
```

Deleting a Rule by Rule Name (http)

4. Now run the following command to **delete** a rule by specifying a port number (**443**) with the **allow** status.

```
sudo ufw delete allow 443
```

```
root@ubuntu:~# sudo ufw delete allow 443
Rule deleted
Rule deleted (v6)
root@ubuntu:~#
```

Deleting a Rule by Port Number (443)

Looks like you're offline!

5. Finally, re-run the following command as you did in step one to list all rules.

```
sudo ufw status numbered
```

As show can see below, the rules for the `5001:5010/tcp` port range, the `http`, and the `443` port are now gone.

```
root@ubuntu:~# sudo ufw status numbered
Status: active

      To Action      From
      --
[ 1] 22/tcp      ALLOW IN    Anywhere
[ 2] 80          ALLOW IN    Anywhere
[ 3] 5001:5010/udp ALLOW IN    Anywhere
[ 4] 22          ALLOW IN    192.168.1.2
[ 5] 80 on eth0    ALLOW IN    Anywhere
[ 6] 80 on wlan0   ALLOW IN    Anywhere
[ 7] 22/tcp (v6)  ALLOW IN    Anywhere (v6)
[ 8] 5001:5010/udp (v6) ALLOW IN    Anywhere (v6)
[ 9] 80 (v6) on eth0 ALLOW IN    Anywhere (v6)
[10] 80 (v6) on wlan0 ALLOW IN    Anywhere (v6)
```

Checking the firewall rules

Resetting the UFW Firewall

There might be times when you need to reset UFW to its defaults, such as after configuring a large set of rules. An update may change your configuration, requiring you to re-configure UFW and possibly start over from scratch.

Looks like you're offline!

Run the `ufw reset` command below to reset all of your firewall rules to their default settings. This command disables UFW and deletes all of your current firewall rules.

```
sudo ufw reset
```

Type 'Y' and press Enter to continue resetting your UFW firewall.

```
root@ubuntu:~# sudo ufw reset
Resetting all rules to installed defaults. This may disrupt existing ssh
connections. Proceed with operation (y|n)? y
Backing up 'user.rules' to '/etc/ufw/user.rules.20211216_173828'
Backing up 'before.rules' to '/etc/ufw/before.rules.20211216_173828'
Backing up 'after.rules' to '/etc/ufw/after.rules.20211216_173828'
Backing up 'user6.rules' to '/etc/ufw/user6.rules.20211216_173828'
Backing up 'before6.rules' to '/etc/ufw/before6.rules.20211216_173828'
Backing up 'after6.rules' to '/etc/ufw/after6.rules.20211216_173828'
```

Resetting UFW

After the reset is complete, you will have a fresh installation of UFW fully disabled, and even your default policies are gone.

Now run the below command to re-enable UFW start configuring your firewall rules from scratch.

```
sudo ufw enable
```

If you decide you don't want to use UFW anymore, then there's no need to re-enable it. Or run the command below to ensure UFW is disabled.

Looks like you're offline!

```
sudo ufw disable
```

```
root@ubuntu:~# sudo ufw disable  
Firewall stopped and disabled on system startup
```

Disabling UFW firewall

Conclusion

Throughout this tutorial, you've realized that setting up a firewall is not too daunting when using UFW. You should now have a good understanding of how to set up and implement your own rules with UFW on Ubuntu.

Now, why not build on this newfound knowledge by learning more about UFW and Docker Security on a Linux machine?