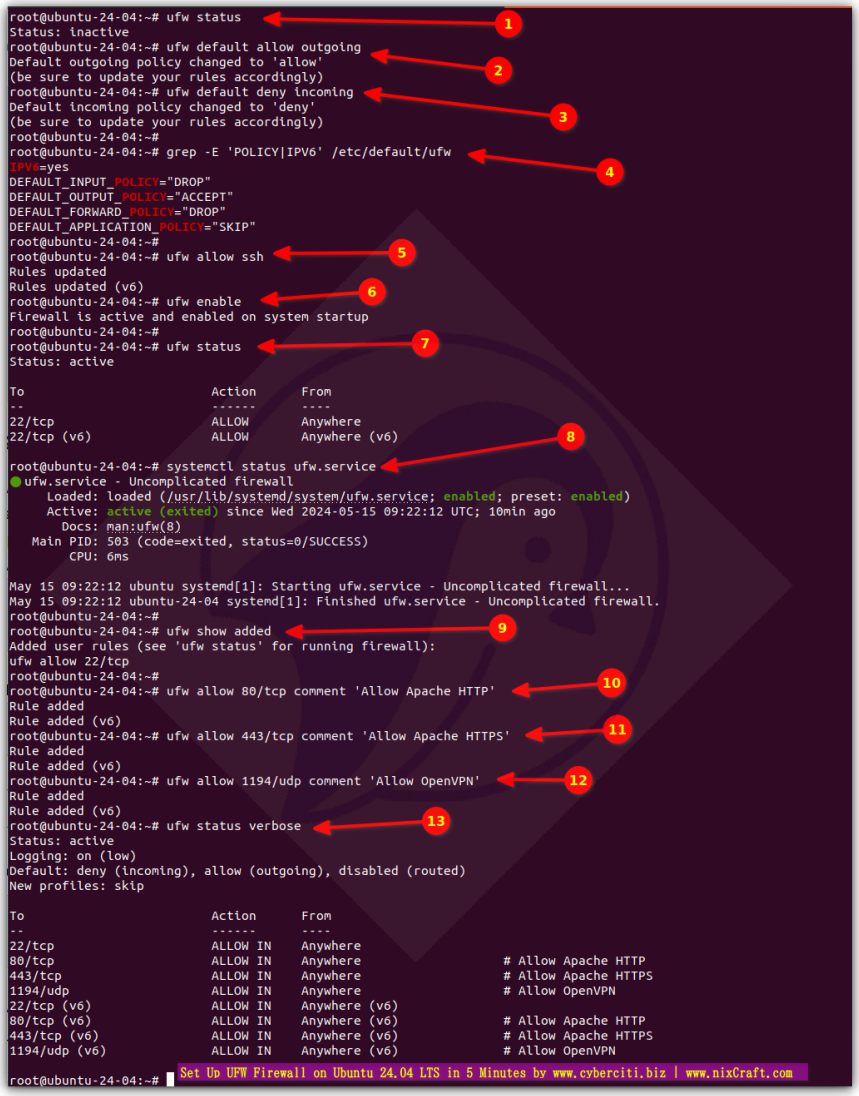
How to Set Up UFW Firewall on Ubuntu 24.04 LTS in 5 Minutes

cyberciti.biz/faq/how-to-set-up-ufw-firewall-on-ubuntu-24-04-lts-in-5-minutes

Author: Vivek Gite Last updated: May 15, 2024 0 comments May 15, 2024



A Ubuntu 24.04 LTS comes with UFW (uncomplicated firewall) that protects the desktop or server against unauthorized access. UFW is an easy-to-use frontend app for a Linux packet filtering system called Netfilter or nftables. Traditionally, Netfilter/nftables rules are set up or configured using the iptables command or nft command by developers and sysadmins. However, new Ubuntu Linux users and developers unfamiliar with firewall concepts find Netfilter/nft syntax confusing. Hence, the ufw project provides an easy-to-use frontend for Ubuntu 24.04 LTS Linux server and desktop. This quick guide makes setting up UFW on Ubuntu 24.04 LTS extremely simple. It provides step-by-step instructions for developers and sysadmins to secure their servers efficiently.

This page explains how to set up a firewall with UFW on Ubuntu 24.04 LTS server or desktop.

Tutorial details	
Difficulty level	Easy
Root privileges	Yes
Requirements	Linux terminal
Category	Firewall
OS compatibility	Debian • Linux • Mint • Pop!_OS • Ubuntu
Est. reading time	5 minutes

The steps are as follows for setting up UFW:

Step 1 – Set Up default UFW policies

First see the current status:

```
$ sudo ufw status
```

By default, the firewall is not enabled. Hence, you will see something as follows on a newly installed system:

```
Status: inactive
```

The default policy firewall works excellent for servers and the desktop. It is always a good policy to close all ports on the server and open only the required TCP or UDP ports. Let us block all incoming connections and only allow outgoing connections from the Ubuntu 24.04 LTS cloud server:

```
$ sudo ufw default allow outgoing
$ sudo ufw default deny incoming
```

Make sure IPv6 support enabled too. Run the grep command as follows:

```
$ grep IPV6 /etc/default/ufw
```

Otherwise, edit the /etc/default/ufw file using a text editor (feel free to choose your preferred text editor):

```
$ sudo nano /etc/default/ufw
```

Set it as follows:

```
IPV6=yes
```

Here is how to verify everything again using the egrep command:

```
$ sudo grep -E 'POLICY|IPV6' /etc/default/ufw
```

That is all.

Step 2 – Open SSH TCP port 22 using the ufw

The next rational step is to allow incoming SSH connections on the default TCP port 22 as follows:

```
$ sudo ufw allow ssh
```

Say you are running the OpenSSH server on TCP port 1222, then:

```
$ sudo ufw allow 1222/tcp
```

You can limit ssh port access to combat bots as follows too:

```
$ sudo ufw limit ssh
```

See "How to limit SSH (TCP port 22) connections with ufw on Ubuntu Linux" for more information.

Step 3 – Turning on the firewall

That is all needed. Next, you need to turn on the firewall protection for your Ubuntu Linux 24.04 LTS machine. For example:

```
$ sudo ufw enable
```

You may need to confirm the operation if asked. To view the current firewall status, type the systemctl command as follows:

```
$ sudo ufw status
```

Please note that once UFW is enabled, it runs across system reboots. You can verify that easily using the systemctl command:

```
$ sudo systemctl status ufw.service
```

Outputs:

```
● ufw.service - Uncomplicated firewall
     Loaded: loaded (/usr/lib/systemd/system/ufw.service; enabled; preset:
enabled)
     Active: active (exited) since Wed 2024-05-15 09:22:12 UTC; 10min ago
       Docs: man:ufw(8)
   Main PID: 503 (code=exited, status=0/SUCCESS)
        CPU: 6ms

May 15 09:22:12 ubuntu systemd[1]: Starting ufw.service - Uncomplicated
firewall...
May 15 09:22:12 ubuntu-24-04 systemd[1]: Finished ufw.service - Uncomplicated
firewall.
root@ubuntu-24-04:~#
```

Step 4 – Opening (allow) TCP or UDP ports

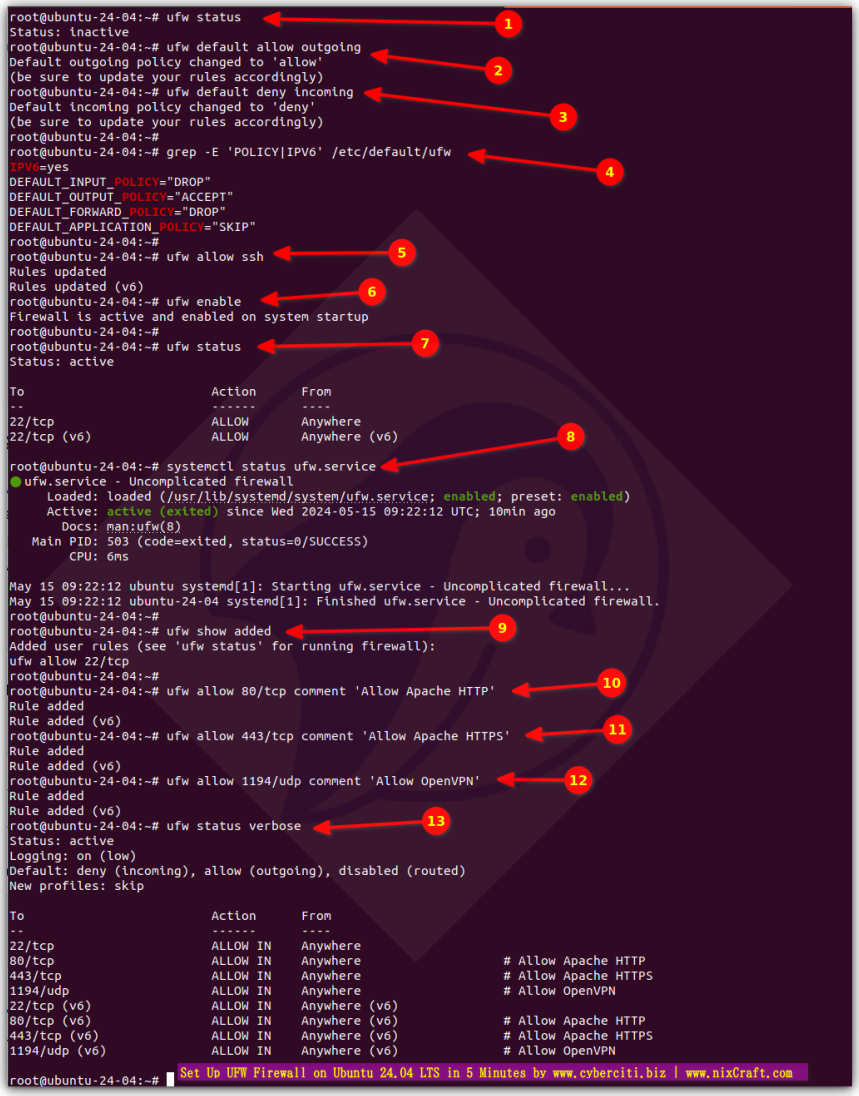
After setting up a firewall policy and opening TCP port 22 for SSH, the next step is to open additional service ports based on the requirements of your application. For example, you may need to open TCP ports 80 and 443 for Nginx, Apache, or Lighttpd web servers:

```
$ sudo ufw allow 80/tcp comment 'Allow Apache HTTP'
$ sudo ufw allow 443/tcp comment 'Allow Apache HTTPS'
```

Here is how to open the OpenVPN UDP port 1194, type:

```
$ sudo ufw allow 1194/udp comment 'Allow OpenVPN'
```

The ufw comment keywords adds comments, which act as an instrumental in understanding firewall rules.



UFW set and running on Ubuntu server version 24.04 LTS (click to enlarge)

Opening TCP and UDP port ranges

```
$ sudo ufw allow 3000:3200/tcp
$ sudo ufw allow 7000:8000/udp
```

Allowing connection from a single IP or CIDR

In this example, you want to allow ALL connections from an IP address called 8.8.8.8, enter:

```
$ sudo ufw allow from 8.8.8.8
```

Let us allow connections from an IP address called 123.1.2.3 to our port 25, enter:

```
$ sudo ufw allow from 123.1.2.3 to any port 25 proto tcp
```

And you can set destination IP 111.1.2.3 for port 25 too:

```
sudo ufw allow from 123.2.3.4 to 111.1.2.3 port 25 proto tcp
```

How to allow connection on specific interface

Open TCP port 22 for wg1 interface only:

```
$ sudo ufw allow in on wg1 to any port 22
```

Say you want to allow connection for TCP port 3306 on incusbr0 interface from 10.105.28.22, then add:

```
$ sudo ufw allow in on incusbr0 from 10.105.28.22 to any port 3306 proto tcp
```

Step 5 – Blocking TCP or UDP ports and connections

Do you want to close ports and block certain IP addresses? The syntax is as follows to deny access. In other words, simply ignoring access to port 23:

```
$ sudo ufw deny 23/tcp comment 'Block telnet'
```

Here is how to deny all connections from an IP address called 1.1.1.1, enter:

```
$ sudo ufw deny from 1.1.1.1
```

How about clock IP/subnet (CIDR) called 1.42.42.32/27, enter:

```
$ sudo ufw deny from 1.42.42.32/27
```

Finally, deny access to 1.1.1.4 (say bad guys or hacker IP address) on port 22? Try:

```
$ sudo ufw deny from 1.1.1.4 to any port 22 proto tcp
```

Testing ufw rules for syntax error

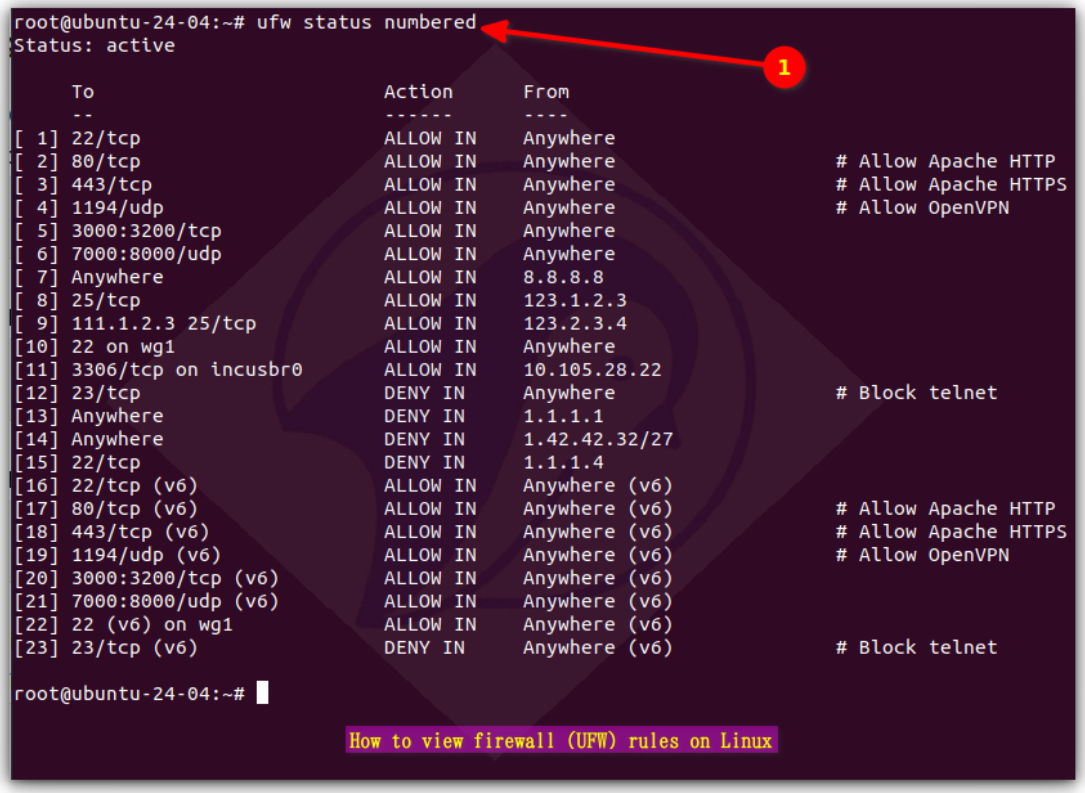
Pass the `--dry-run` option to test a ufw rule. This means not modifying anything, just displaying the firewall changes. This is great for testing rules before insertion or deletion. The syntax:

```
$ sudo ufw --dry-run rule
$ sudo ufw --dry-run allow from 192.168.1.0/24 to any port 3128 proto tcp
# make some error and see what it doe
$ sudo ufw --dry-run allow from 192.168.1.1000/24 to any port 3128 proto tcp
Outputs:
ERROR: Bad source address
$ sudo ufw --dry-run allow from 192.168.1.1/24 to any port 3128 proto tcp_fake_proto
Outputs:
ERROR: Unsupported protocol 'tcp_fake_proto'
```

Step 6 – Viewing firewall rules

You can see firewall status as numbered list of RULES:

```
$ sudo ufw status numbered
```



Click to enlarge image

Step 7 – Deleting ufw firewall rules

Get list all of the current rules in a numbered list format:

```
$ sudo ufw status numbered
```

Outputs:

```
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere                   # Allow
Apache HTTP
[ 3] 443/tcp                    ALLOW IN    Anywhere                   # Allow
Apache HTTPS
[ 4] 1194/udp                   ALLOW IN    Anywhere                   # Allow
OpenVPN
[ 5] 3000:3200/tcp              ALLOW IN    Anywhere
[ 6] 7000:8000/udp              ALLOW IN    Anywhere
[ 7] Anywhere                   ALLOW IN    8.8.8.8
[ 8] 25/tcp                     ALLOW IN    123.1.2.3
[ 9] 111.1.2.3 25/tcp           ALLOW IN    123.2.3.4
[10] 22 on wg1                  ALLOW IN    Anywhere
[11] 3306/tcp on incusbr0       ALLOW IN    10.105.28.22
[12] 23/tcp                     DENY IN     Anywhere                   # Block
telnet
[13] Anywhere                   DENY IN     1.1.1.1
[14] Anywhere                   DENY IN     1.42.42.32/27
[15] 22/tcp                     DENY IN     1.1.1.4
[16] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[17] 80/tcp (v6)                ALLOW IN    Anywhere (v6)              # Allow
Apache HTTP
[18] 443/tcp (v6)               ALLOW IN    Anywhere (v6)              # Allow
Apache HTTPS
[19] 1194/udp (v6)              ALLOW IN    Anywhere (v6)              # Allow
OpenVPN
[20] 3000:3200/tcp (v6)         ALLOW IN    Anywhere (v6)
[21] 7000:8000/udp (v6)         ALLOW IN    Anywhere (v6)
[22] 22 (v6) on wg1             ALLOW IN    Anywhere (v6)
[23] 23/tcp (v6)                DENY IN     Anywhere (v6)              # Block
telnet
```

To remove firewall rule # 22 type the command:

```
$ sudo ufw delete 22
$ sudo ufw status numbered
```

See how to delete a UFW firewall rule on Ubuntu / Debian Linux tutorial for further information.

Step 8 – Stopping and removing UFW

If you no longer need ufw, here is how to disable it:

```
$ sudo ufw disable
$ sudo ufw reset
```

Step 9 – View the firewall logs

By default all UFW entries are logged into /var/log/ufw.log file. Use the grep/less/more and other commands to view the ufw logs. For examples:

```
$ sudo journalctl -u ufw.service
```

Let us print a list of all IP address trying to log in via SSH port but dropped by the UFW:

```
$ journalctl -u ufw.service -g 'DPT=22' |\
grep -E -o 'SRC=([0-9]{1,3}[\.]){3}[0-9]{1,3}' |\
awk -F'=' '{ print $2 }' | sort -u
```

Finally, here is how to display the list of rules:

```
$ sudo ufw show listening
```

Outputs:

```
tcp:
tcp6:
  22 * (systemd)
   [16] allow 22/tcp

udp:
  68 10.83.200.36 (systemd-networkd)
   [ 7] allow from 8.8.8.8
   [13] deny from 1.1.1.1
   [14] deny from 1.42.42.32/27
```

Also:

```
$ sudo ufw show added
```

Summing up

Wasn't that easy? You've now learned how to safeguard your Ubuntu 24.04 LTS Linux server. For further guidance, consult the documentation online or access them using the man command (`ufw help` command).

```
$ man ufw
$ ufw help
```

This entry is **13** of **13** in the **Uncomplicated Firewall (UFW)** series. Keep reading the rest of the series:

😳 Was this helpful? Please add a comment to show your appreciation or feedback.

Hi! 🤠

I'm Vivek Gite, and I write about Linux, macOS, Unix, IT, programming, infosec, and open source. Subscribe to my RSS feed or email newsletter for updates.