



Home > Shell Scripts > How To Format Shell Programs Using Shfmt In Linux

SHELL SCRIPTS ♦ LINUX ♦ OPENSOURCE ♦ PROGRAMMING ♦ SCRIPTING ♦ UTILITIES

# How To Format Shell Programs Using Shfmt In Linux

Written by Karthick | Published: July 10, 2021 | 11901 views

3 comments11♥️👍👎🔖📧

KEEP IN TOUCH

- FACEBOOK
- TW
- LINKEDIN
- YO
- EMAIL
- RE
- RSS

In this guide, we will be discussing what is Shfmt, how to install Shfmt in Linux, and finally how to format shell programs using Shfmt in Linux.

## Table of Contents

- 1. Introduction
- 2. Install Shfmt in Linux
- 3. Format shell programs using Shfmt in Linux
  - 3.1. Custom Indentation
  - 3.2. Diff style output
  - 3.3. List scripts to be formatted
  - 3.4. Write output to file
  - 3.5. Find shell scripts in the path
  - 3.6. Shfmt can detect errors too
  - 3.7. Shfmt in external editors
- Conclusion

## 1. Introduction

If you are from a programming background, you may know the terms such as formatting, linting, etc.

There are important supporting tools that comes with every code editor for making our life easy while we code.

Similarly, for shell scripts, we have **shfmt**. shfmt is used to format, parse and interpret your shell scripts. Shfmt supports Bash, mksh, and Posix shells.

Shfmt is highly configurable and can format your code in several fashion and conventions.

You can install and use Shfmt tool from command line to work with your shell scripts. Also there are plugins available in popular text editors that will use shfmt to format your shell scripts.

## 2. Install Shfmt in Linux

Shfmt is available as snap application. If your distribution has snap installed, you can install shfmt using command:

```
$ sudo snap install shfmt
```

The another way to install Shfmt is by using the following one-liner command:

```
$ curl -sS https://webinstall.dev/shfmt | bash
```

The above two methods can be used irrespective of what distribution you are running. There are a few distributions where you can install shfmt from their package repositories.

To Install shfmt in Alpine Linux, run:

```
$ sudo apk add shfmt
```

Install shfmt in Arch Linux, EndeavourOS and Manjaro Linux:

```
$ sudo pacman -S shfmt
```

Install shfmt in FreeBSD.

```
$ sudo pkg install devel/shfmt
```

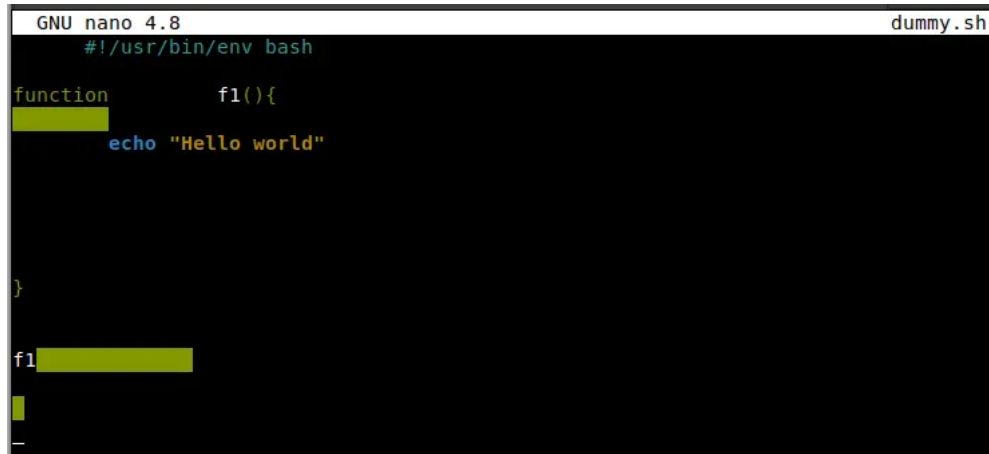
```
$ sudo pkg install shfmt
```

### 3. Format shell programs using Shfmt in Linux

If this is the first time you use Shfmt, start by running the `help` command to get an idea of what options shfmt supports:

```
$ shfmt --help
```

Now, allow me to show you an example. I have created a shell script with no proper formatting.



Sample script

SEARCH

KEEP IN TOUCH

- f

FACEBOOK
- X

TW
- in

LINKEDIN
- ▶

YO
- ✉

EMAIL
- 🍷

RE
- 📡

RSS

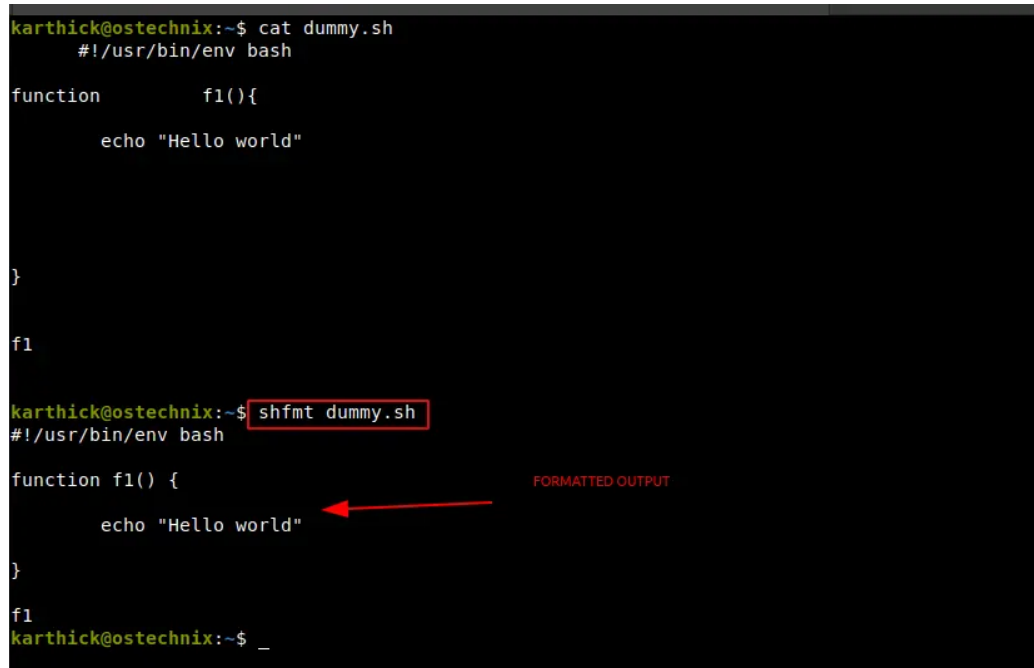
Take a look at the above image. It looks weird, right?

Let us format this script using shfmt like below:

```
$ shfmt dummy.sh
```

Shfmt will start to format the given script.

Here is the output of the above script before and after the optimization:



```
karthick@ostechnix:~$ cat dummy.sh
#!/usr/bin/env bash

function      f1(){
    echo "Hello world"
}

f1

karthick@ostechnix:~$ shfmt dummy.sh
#!/usr/bin/env bash

function f1() {
    echo "Hello world"
}

f1
karthick@ostechnix:~$ _
```

Format shell programs using shfmt

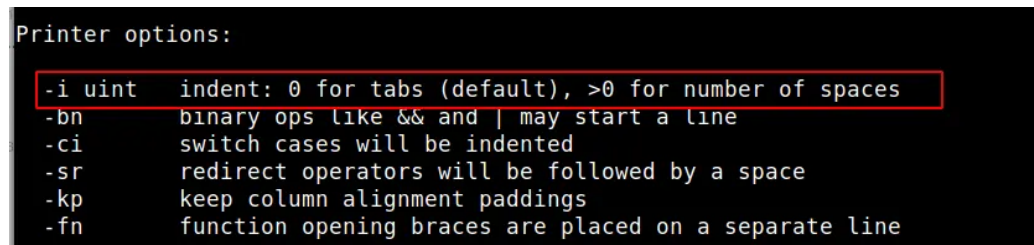
From the above output, you can see our script is nicely formatted.

You may wonder why the indentation is set to **tab**. This is default and modifiable.

### 3.1. Custom Indentation

I always follow [google shell script style docs](#) and set the indentation to **2 spaces** instead of tabs.

To set custom indentation use **-i** flag. Let's see what the help menu has to say about this **-i** flag:



```
Printer options:
-i uint    indent: 0 for tabs (default), >0 for number of spaces
-bn        binary ops like && and | may start a line
-ci        switch cases will be indented
-sr        redirect operators will be followed by a space
-kp        keep column alignment paddings
-fn        function opening braces are placed on a separate line
```

Custom indentation using shfmt

Any value greater than zero is passed with **-i** flag is the amount of spaces that will be used to intend.

```
$ shfmt -i 2 scriptname.sh
```

SEARCH

KEEP IN TOUCH

f FACEBOOK X TW  
in LINKEDIN y YO  
EMAIL RE  
RSS

```
karthick@ostechnix:~$ shfmt -i 2 dummy.sh
#!/usr/bin/env bash

function f1() {
    echo "Hello world"
}

f1
karthick@ostechnix:~$ _
```

Set indentation using shfmt

### 3.2. Diff style output

If you need information on what exactly has been formatted you can use `-d` flag. Take a look at the below image, it shows what is formatted in green color with `+` symbol.

```
karthick@ostechnix:~$ shfmt -d -i 2 dummy.sh
--- dummy.sh.orig
+++ dummy.sh
@@ -1,7 +1,7 @@
#!/usr/bin/env bash

-function f1() {
-echo    "Hello world"
-
+function f1() {
+ echo "Hello world"
+}

f1
```

Diff style output

[f](#) [FACEBOOK](#) [X](#) [TW](#)  
[in](#) [LINKEDIN](#) [YO](#)  
[EMAIL](#) [RE](#)  
[RSS](#)

### 3.3. List scripts to be formatted

You can use the `-l` flag which will get the list of scripts that needs to be formatted.

To demonstrate this, I am going to create three script files. I left the second and third scripts unformatted. And I have given a lot of space in the echo statement of the 2nd and 3rd files.

```
karthick@ostechnix:~/Downloads$ cat sample*.sh
#!/usr/bin/env bash

echo "Sample file 1"
#!/usr/bin/env bash

echo      "sample file 2"

#!/usr/bin/env bash
echo     "sample file 3"
```

Input scripts

Now if I run the shfmt command with the `-l` flag, it should pick my second sample 2 and 3 files only:

```
$ shfmt -l sample*.sh
sample2.sh
sample3.sh
```

```
karthick@ostechnix:~/Downloads$ shfmt -l sample*.sh
sample2.sh
sample3.sh
```

List of scripts to be formatted

By default, Shfmt writes the output of the formatted script to stdout, which is the terminal. If you wish to write the output to a file, you can use `-w` flag.

```
$ shfmt -w scriptname.sh
```

### 3.5. Find shell scripts in the path

With `-f` flag, shfmt will scan the given directory recursively and will find all the shell scripts.

It doesn't matter whether a script has the extension `.sh` or not. Shfmt can recognize all the shell scripts even if they don't have extensions.

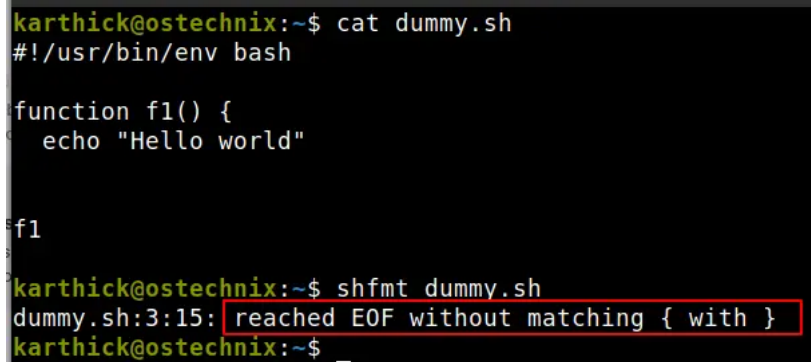
```
$ shfmt -f /home/ostechnix
/home/ostechnix/.config/envman/load.sh
/home/ostechnix/.local/bin/webi
/home/ostechnix/Downloads/sampleshell
/home/ostechnix/dummy.sh
```

[f FACEBOOK](#) [X TW](#)[in LINKEDIN](#) [YO](#)[EMAIL](#) [RE](#)[RSS](#)

### 3.6. Shfmt can detect errors too

Shfmt is not just for formatting shell scripts. It can also spot the errors in the scripts too.

Take a look at the below image, where my braces to terminate the function are missing and shfmt spots it.

A terminal window with a black background and green text. The user is at the prompt 'karthick@ostechnix:~\$'. They run 'cat dummy.sh' and the content of the file is displayed: '#!/usr/bin/env bash', 'function f1() {', ' echo "Hello world"', and 'f1'. Then they run 'shfmt dummy.sh' and the output is 'dummy.sh:3:15: reached EOF without matching { with }', which is highlighted with a red box. The prompt returns to 'karthick@ostechnix:~\$ \_'.

Shfmt can detect syntax errors in scripts

You can also use the bash built-in syntax check feature by using `-n` flag which will validate your code and list all the errors.

```
$ bash -vn scriptname.sh
```

### 3.7. Shfmt in external editors

In real-time, you can use any one of the text editors of your choice such as Vim, Atom, Sublime Text, Vs code, etc. Some of these text editors might have native support to format bash scripts.

Each code editor has a plugin that integrates with shfmt. Go to the GitHub official repository (link given at the end) and at the bottom of the page, you will see the list of plugins for each text editor that uses shfmt underneath.

## Conclusion

In this guide, we have seen what is Shfmt and how to format Shell programs using Shfmt with

This website uses cookies to improve your experience. By using this site, we will assume that you're OK with it. [Accept](#) [Read More](#)

organized. Integrate with text editors and you will have a very smooth workflow in creating the scripts.

Resource:

- [Shfmt GitHub Repository](#)

Related read:

- [ShellCheck – A Free Utility To Find Bugs In Your Shell Scripts](#)
- [Fix “Exec format error” When Running Scripts With run-parts Command](#)
- [How To Run All Scripts In A Directory In Linux](#)
- [How To Parse And Pretty Print JSON With Linux Commandline Tools](#)

Featured image from [Pixabay](#).

- FORMAT SHELL SCRIPT
- LINUX
- PROGRAMMING
- SCRIPTING
- SHELL SCRIPTS
- SHFMT

SEARCH

KEEP IN TOUCH

- f

FACEBOOK

X

TW
- in

LINKEDIN

▶

YO
- ✉

EMAIL

RE
- 📡

RSS

3 comments

11

f


X

in

🗨

📧

✉



KARTHICK

Karthick is a passionate software engineer who loves to explore new technologies. He is a public speaker and loves writing about technology especially about Linux and opensource.

🌐

Previous post

Next post

How To Setup Firewall With UFW On Linux

Introduction To Ansible Automation Platform

YOU MAY ALSO LIKE

How To Run All Scripts In A Directory...

Published: April 4, 2020

Bash Scripting – Associative Array Explained With Examples

Published: October 6, 2021

Bash Scripting – For Loop Explained With Examples

Published: September 21, 2021



3 COMMENTS




JALAL HAJIGHOLAMALI


🕒 July 11, 2021 - 9:35 am

Hi,  
Thanks a lot  
useful article

REPLY



I like the article but why am I getting this:  
user@Flash:~/scripts\$ shfmt -i 2 -d -w am  
open .am559725908: permission denied  
when I try to write the file?

SK

REPLY

🕒 July 13, 2021 - 11:00 pm

Looks like your current doesn't have proper permission to the file. Did you try with sudo?

LEAVE A COMMENT

Your Comment

Name\*

Email\*

☐ Save my name, email, and website in this browser for the next time I comment.  
\* By using this form you agree with the storage and handling of your data by this website.







SUBMIT

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

How To Setup Firewall With UFW On Linux

Written by Karthick | Published: July 6, 2021 | Updated: December 25, 2021 | 9.2K views

🗨 3 comments

19 ❤️      

This guide explains what is UFW, how to install UFW in Linux, and how to setup firewall with UFW on various Linux operating systems.

Table of Contents

Introduction

What is UFW?

1. Install UFW in Linux

1.1. Enable, start, and stop UFW service

3. Setup firewall with UFW on Linux








3.1. Getting help

3.2. Set default rules

3.2.1. Check status of UFW firewall rules

SEARCH








KEEP IN TOUCH

-  FACEBOOK
-  TWITTER
-  LINKEDIN
-  YOUTUBE
-  EMAIL
-  REDDIT
-  RSS

SEARCH

Type and hit enter...

KEEP IN TOUCH

-  FACEBOOK
-  TWITTER
-  LINKEDIN
-  YOUTUBE
-  EMAIL
-  REDDIT
-  RSS

[3.2.2. Add rules](#)


[3.2.3. Delete UFW firewall rules](#)


[3.2.4. Enable, disable and reload UFW firewall rules](#)


[3.2.5. Adding policy for port ranges](#)


SEARCH


KEEP IN TOUCH


FACEBOOK


TW

LINKEDIN

YO

EMAIL

RE

RSS

## Introduction

Security is a serious business. Whether you are running your Linux operating system in data centers or on your desktop, you should secure your operating system against all possible threats.

In fact, servers running in the corporate environment will be well protected. Most corporate companies invests millions of dollars to secure their infrastructure.

There will be a separate network team, firewall team, security team to protect your environment and Linux servers. This will not be the case when you run Linux on your desktops or servers.

You should be aware of how to secure your Linux machines with the right tools. One such tool is UFW.

## What is UFW?

UFW, stands for **Uncomplicated Firewall**, is a firewall program that comes preinstalled by default with Ubuntu-based distributions.

Why UFW instead of iptables? You might wonder.

If you don't know already, Netfilter is a packet filtering system that ships with a Linux kernel and iptables are used to manipulate net filters with a set of commands.

Getting comfortable with iptables may take time and could be a daunting task. To make the firewall management easy, there are many front-ends to iptables are created. UFW is one of them.

UFW is a command line front-end to manage iptables. It provides a framework for managing and manipulating netfilter firewall.

UFW is available by default in all Ubuntu installations after 8.04 LTS version.

There is also a graphical front-end for UFW named **Gufw**. We will discuss about it in a separate guide. In this article, our focus will be on using ufw from command line.

Without further ado, let us go ahead and see how to install and setup UFW firewall on Linux.

## 1. Install UFW in Linux

UFW comes preinstalled with most of Debian-based and Arch-based distributions. To check if UFW is installed or not, run the following command:

```
$ which ufw
/usr/sbin/ufw
```

```
$ ufw version
ufw 0.36
Copyright 2008-2015 Canonical Ltd.
```

If it is not installed on your distribution, you can install it using your distribution's default package manager.

To install UFW in Alpine Linux, run:



```
$ sudo apk add ufw
```

Install UFW in Arch Linux and its variants such as EndeavourOS and Manjaro Linux:

```
$ sudo pacman -S ufw
```

Install ufw in Debian, Ubuntu and its derivatives:

```
$ sudo apt update
$ sudo apt install ufw
```

Install UFW in Fedora:

```
$ sudo dnf install ufw
```

UFW is available in [EPEL] repository for Enterprise Linux operating systems such as RHEL, CentOS, AlmaLinux and Rocky Linux.

Enable [EPEL] repository and install UFW in RHEL, CentOS, AlmaLinux, Rocky Linux like below:

```
$ sudo dnf install epel-release
```

```
$ sudo dnf install ufw
```

Install UFW in openSUSE:

```
$ sudo zypper install ufw
```

### 1.1. Enable, start, and stop UFW service

In Debian-based systems, UFW daemon will be started and enabled automatically.

Run the following command to check the status of the UFW service:








```
$ systemctl status ufw
```

Sample output:

```
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/usr/lib/systemd/system/ufw.service; enabled; vendor prese>
   Active: active (exited) since Mon 2021-07-05 20:08:01 IST; 44s ago
     Docs: man:ufw(8)
           man:ufw-framework(8)
           file:///usr/share/doc/ufw/README
   Process: 21690 ExecStart=/usr/libexec/ufw/ufw-init start (code=exited, stat>
   Main PID: 21690 (code=exited, status=0/SUCCESS)
      CPU: 169ms
```

SEARCH

KEEP IN TOUCH

-  **FACEBOOK**
-  **TW**
-  **LINKEDIN**
-  **YO**
-  **EMAIL**
-  **RE**
-  **RSS**

```
Jul 05 20:08:01 ostechnix systemd[1]: Starting Uncomplicated firewall...
Jul 05 20:08:01 ostechnix systemd[1]: Finished Uncomplicated firewall.
```

The other way way to check if UFW service is enabled and active:

```
$ systemctl is-enabled ufw
enabled
```

```
$ systemctl is-active ufw
active
```

If UFW service is not started automatically after installation, run the following command to start UFW service:

```
$ sudo systemctl start ufw
```

Ufw should also be enabled to automatically started between system reboots.

```
$ sudo systemctl enable ufw
```

Or, you can combine both commands into one to enable and start the UFW service in one go like below:

```
$ sudo systemctl enable --now ufw
```

To stop UFW service, simply run:

```
$ sudo systemctl stop ufw
```

## 3. Setup firewall with UFW on Linux

### 3.1. Getting help

If you're new to UFW, the first thing to do after installing it is to refer the help section and man page of UFW to get the basic idea about UFW usage.

```
$ ufw --help
```

```
$ man ufw
```

If you forgot the syntax or need a reference for a particular feature of ufw, these two commands will be very handy.

### 3.2. Set default rules

Using UFW, you can create firewall rules (or policies) to allow or deny a specific service. Through these policies, you instruct the UFW what port, service, IP addresses, and interfaces

[!\[\]\(066cb4a00c9d9f40edb6f87372ec6f08\_img.jpg\) FACEBOOK](#)[!\[\]\(aceb1790ece33f2eac474d4a9431c6d6\_img.jpg\) TW](#)[!\[\]\(b9742ff0bb3da904abeeee81c2bcb456\_img.jpg\) LINKEDIN](#)[!\[\]\(26cddea01ddf7f002af4ba779c4999ee\_img.jpg\) YO](#)[!\[\]\(b78e2d0769ad682766c36e077fde3d60\_img.jpg\) EMAIL](#)[!\[\]\(1adebd97b172010e8ebc985144647a7c\_img.jpg\) RE](#)[!\[\]\(eff7520f80aa06fb7298beb68337d76d\_img.jpg\) RSS](#)

There are default policies that come with ufw. The default policy will drop all incoming connections and allow all outgoing connections.

**IMPORTANT:** If you are setting up ufw in a remote server, make sure you've allowed the ssh port or service before enabling ufw firewall.

Default incoming policy will deny all incoming connections. So if you didn't configure the rules to allow SSH, you will be locked out of the remote system and you can not log in to the system. This is not the case when you running ufw on your local system.

The default policies are defined in the `/etc/default/ufw` file. Here is the contents of the file:

File: /etc/default/ufw

```
1 # /etc/default/ufw
2 #
3
4 # Set to yes to apply rules to support IPv6 (no means only IPv6 on loopback
5 # accepted). You will need to 'disable' and then 'enable' the firewall for
6 # the changes to take affect.
7 IPV6=yes
8
9 # Set the default input policy to ACCEPT, DROP, or REJECT. Please note that if
10 # you change this you will most likely want to adjust your rules.
11 DEFAULT_INPUT_POLICY="DROP"
12
13 # Set the default output policy to ACCEPT, DROP, or REJECT. Please note that if
14 # you change this you will most likely want to adjust your rules.
15 DEFAULT_OUTPUT_POLICY="ACCEPT"
16
17 # Set the default forward policy to ACCEPT, DROP or REJECT. Please note that
18 # if you change this you will most likely want to adjust your rules
19 DEFAULT_FORWARD_POLICY="DROP"
20
21 # Set the default application policy to ACCEPT, DROP, REJECT or SKIP. Please
22 # note that setting this to ACCEPT may be a security risk. See 'man ufw' for
23 # details
24 DEFAULT_APPLICATION_POLICY="SKIP"
25
```

Ufw default policies

From here, we can set default policies.

Alternatively, We can use the `ufw allow` command to set default policies for incoming and outgoing commands:

```
$ sudo ufw default deny incoming
```

```
$ sudo ufw default allow outgoing
```

3.2.1. Check status of UFW firewall rules

To check if default policies are active, run the following command:

```
$ sudo ufw status
```

Sample output:

```
Status: active

To Action From
--
SSH ALLOW Anywhere
```

SEARCH

KEEP IN TOUCH

f

FACEBOOK

X

TW

in

LINKEDIN

▶

YO

✉

EMAIL

🐙

RE

📡

RSS

SSH (v6)	ALLOW	Anywhere (v6)
ff02::fb mDNS	ALLOW	Anywhere (v6)

And for more verbose status information use this command:

```
$ sudo ufw status verbose
```

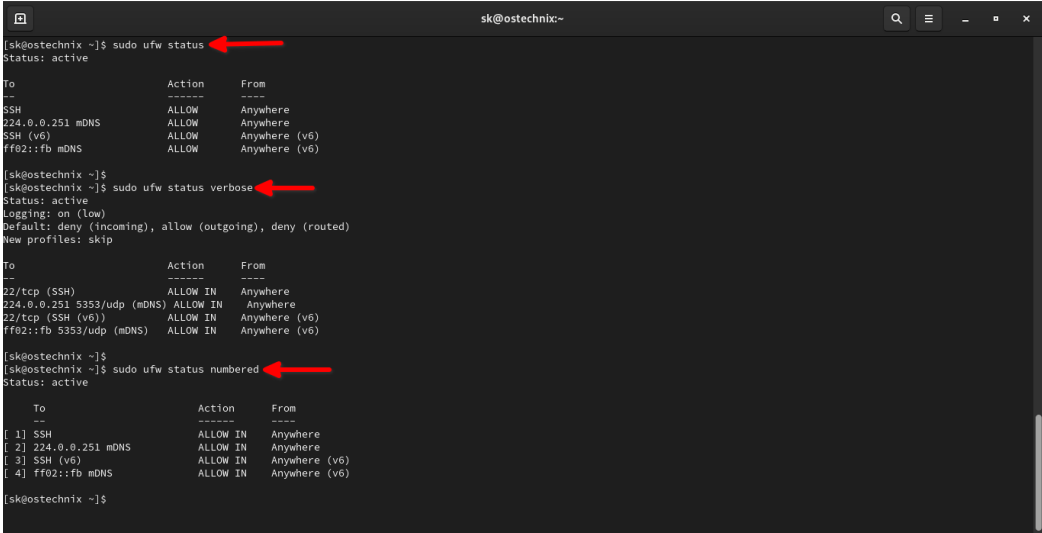
Sample output:

Status: active  
Logging: on (low)  
Default: deny (incoming), allow (outgoing), deny (routed)  
New profiles: skip

To	Action	From
--	-----	----
22/tcp (SSH)	ALLOW IN	Anywhere
224.0.0.251 5353/udp (mDNS)	ALLOW IN	Anywhere
22/tcp (SSH (v6))	ALLOW IN	Anywhere (v6)
ff02::fb 5353/udp (mDNS)	ALLOW IN	Anywhere (v6)

To view the numbered format, run:

```
$ sudo ufw status numbered
```



View UFW status

3.2.2. Add rules

Let me take SSH as an example to demonstrate how to add firewall rules with ufw command. Take a look at the below commands:

```
$ sudo ufw allow ssh
```

```
$ sudo ufw allow 22
```

SEARCH

KEEP IN TOUCH

- f

FACEBOOK

X

TW
- in

LINKEDIN

▶

YO
- ✉

EMAIL

RE
- 📡

RSS

```
$ sudo ufw allow 22/tcp
```

```
$ sudo ufw allow 2222/tcp
```

All commands serves the same purpose.

[ 1 ] - In the first command, I am allowing all access to ssh service. UFW knows by default ssh listens to port 22. So when you use allow ssh service, it will also enforce the rule for port 22.

[ 2 ] - In the second command, I am explicitly telling to allow incoming connections for port 22.

[ 3 ] - The third command is the same as the second command. It allows all access to tcp port 22. Both TCP and UDP protocols are supported.

[ 4 ] - In the fourth command, I am allowing a custom ssh port (i.e. 2222) to accept the incoming connections.

You can use these four commands not only for ssh but for any services and ports. For instance, if you want to connect to PostgreSQL running at port 5433, then the rule should be added like below.

```
$ sudo ufw allow 5433
```

Similarly, we can use ufw deny command to reject incoming connections:

```
$ sudo ufw deny 5433
```

This command will deny traffic on port 5433.

### 3.2.3. Delete UFW firewall rules

To remove a rule or policy, you can use ufw delete command.

For instance, If you no longer wish to allow HTTP traffic, simply run:

```
sudo ufw delete allow 80
```

### 3.2.4. Enable, disable and reload UFW firewall rules

This is different than enabling and starting the UFW daemon. Starting the ufw systemd unit will not enforce your firewall rules. UFW has dedicated commands to enable, disable and reload firewall rules.

To make the rules effective, you have to run the following command:

```
$ sudo ufw enable  
Firewall is active and enabled on system startup
```

As I already mentioned, use the following command to view the status of UFW firewall rules:

```
$ sudo ufw status
```

[!\[\]\(008bfeb2de157dcb66edb3a8218c280e\_img.jpg\) FACEBOOK](#)[!\[\]\(135faf555a2da147cc447132eda26e60\_img.jpg\) TW](#)[!\[\]\(e03857cdd33a5ff23dbb9f5eebaa4497\_img.jpg\) LINKEDIN](#)[!\[\]\(d28209ff6e28188fea111756512e918d\_img.jpg\) YO](#)[!\[\]\(141489a9a09a5a55d166fd7134726d50\_img.jpg\) EMAIL](#)[!\[\]\(f3cd43c0876202a7cb76d17dba19e77d\_img.jpg\) RE](#)[!\[\]\(06456157f083c12e510a7643240746db\_img.jpg\) RSS](#)








Sample output:

```
Status: active

To Action From
--
SSH ALLOW Anywhere
224.0.0.251 mDNS ALLOW Anywhere
SSH (v6) ALLOW Anywhere (v6)
ff02::fb mDNS ALLOW Anywhere (v6)
```

SEARCH

KEEP IN TOUCH

-  FACEBOOK
-  TW
-  LINKEDIN
-  YO
-  EMAIL
-  RE
-  RSS

To disable the Firewall rules, run:

```
$ sudo ufw disable
Firewall stopped and disabled on system startup
```

**Please note:** The above command will only disable the firewall rules. The UFW daemon will be still running and enabled on reboots.

After adding any policy, reload the ufw for the policy to take effect using command:

```
$ sudo ufw reload
```

### 3.2.5. Adding policy for port ranges

You can add a policy for a range of ports instead of creating a policy for a single port:

```
$ sudo ufw allow 8000:8080/tcp
```

```
$ sudo ufw deny 8000:8080/tcp
```

### 3.2.6. Adding policy for specific IP addresses, subnets and ports

You can create more fine-grained rules with ufw. Let’s say if you want your server to be connected (ssh’ed) from a specific IP only, you can do so by adding the following rule.

```
$ sudo ufw allow from 192.168.156.2
```

```
$ sudo ufw allow from 192.168.156.2 to any port 2222
```

The first command allows specified IP to connect based on opened ports. The second command specifies that the user can connect to port 2222 only from 192.168.156.2.

To allow a group of IPs from the same subnet to connect to ssh, you can use the subnet while adding a rule, allowing all IP parts of that subnet to be connected to port 2222.

```
$ sudo ufw allow from 192.168.156.1/24 to any port 2222
```

### 3.2.7. Adding network interface policy

You can also create policies based on network interfaces. The following command will create a policy to accept connections for network interface `en01` to port `2222`.

SEARCH

```
$ sudo ufw allow in on en01 to any port 2222
```

### 3.2.8. Test rules without applying them using dry-run option

UFW has `--dry-run` option to test rules without actually applying them. For example, the following is what would be applied if opening the SSH port:

KEEP IN TOUCH

```
$ sudo ufw --dry-run allow ssh
```

f

FACEBOOK

X

TW

in

LINKEDIN

▶

YO

✉

EMAIL

🐙

RE

📡

RSS

sk@ostechnix~

🔍

☰

▢

✕

```
[sk@ostechnix ~]$ sudo ufw --dry-run allow ssh
[sudo] password for sk:
#filter
:ufw-user-input - [0:0]
:ufw-user-output - [0:0]
:ufw-user-forward - [0:0]
:ufw-before-logging-input - [0:0]
:ufw-before-logging-output - [0:0]
:ufw-before-logging-forward - [0:0]
:ufw-user-logging-input - [0:0]
:ufw-user-logging-output - [0:0]
:ufw-user-logging-forward - [0:0]
:ufw-after-logging-input - [0:0]
:ufw-after-logging-output - [0:0]
:ufw-after-logging-forward - [0:0]
:ufw-logging-deny - [0:0]
:ufw-logging-allow - [0:0]
:ufw-user-limit - [0:0]
:ufw-user-limit-accept - [0:0]
## RULES ##

## tuple ## allow tcp 22 0.0.0.0/0 any 0.0.0.0/0 SSH - in
-A ufw-user-input -p tcp --dport 22 -j ACCEPT -m comment --comment 'dapp_SSH'

## tuple ## allow udp 5353 224.0.0.251 any 0.0.0.0/0 mDNS - in
-A ufw-user-input -p udp -d 224.0.0.251 --dport 5353 -j ACCEPT -m comment --comment 'dapp_mDNS'

## tuple ## allow any 22 0.0.0.0/0 any 0.0.0.0/0 in
-A ufw-user-input -p tcp --dport 22 -j ACCEPT
-A ufw-user-input -p udp --dport 22 -j ACCEPT

## END RULES ##

## LOGGING ##
-A ufw-after-logging-input -j LOG --log-prefix "[UFW BLOCK] " -m limit --limit 3/min --limit-burst 10
-A ufw-after-logging-forward -j LOG --log-prefix "[UFW BLOCK] " -m limit --limit 3/min --limit-burst 10
-I ufw-logging-deny -m conntrack --ctstate INVALID -j RETURN -m limit --limit 3/min --limit-burst 10
```

Dry run UFW commands

As you can see in the above output, the `ufw` command only outputs the resulting rules, but not apply them when we add `--dry-run` option. This comes in handy when you want to test any firewall policies.

### 3.2.9. Add comment to each rule

You might have added several rules. After a particular number of rules (Say 50), you have no way of remembering what the rule is about.

In that case, you can add a comment to each rule like below:

```
$ sudo ufw allow 22 comment 'open port 22 for ssh'
```

The above command will allow all traffic to port `22` and adds a comment for the rule. This way you can easily find the purpose of a specific rule.

## 4. Which rule gets priority?

Priority is important when you are creating multiple rules for the same service/ports. Policy gets their priority in the order they created. Run the following command which will give you policy along with its priority.

```
$ sudo ufw status numbered
```


Sample output:


```
Status: active


      To                Action      From
      --                -
[ 1] 22                ALLOW IN   Anywhere
[ 2] 2222              ALLOW IN   Anywhere
[ 3] 2222              ALLOW IN   192.168.156.2
[ 4] 2222              DENY IN     192.168.157.0/24
[ 5] 22 (v6)           ALLOW IN   Anywhere (v6)
[ 6] 2222 (v6)         ALLOW IN   Anywhere (v6)
```


SEARCH


KEEP IN TOUCH


 **FACEBOOK**


 **TW**

 **LINKEDIN**

 **YO**

 **EMAIL**

 **RE**

 **RSS**

Take a look at [ 4 ] in the above output. Any connection to port 2222 from the subnet 192.168.157.0/24 should be dropped.

But when I try to connect from any of the machines from the same subnet, the connection will be allowed because the high priority has been given to [ 2 ].

To override this behavior you have to create rules with priority. You can delete the existing rule and add a new rule with priority and reload the service.

```
$ sudo ufw delete 4
Deleting:
deny from 192.168.157.0/24 to any port 2222
Proceed with operation (y|n)? y
Rule deleted
```

```
$ sudo ufw insert 2 deny from 192.168.157.0/24 to any port 2222
Rule inserted
```

```
$ sudo ufw reload
Firewall reloaded
```

```
$ sudo ufw status numbered
Status: active
To Action From
-- -
[ 1] 22 ALLOW IN Anywhere
[ 2] 2222 DENY IN 192.168.157.0/24
[ 3] 2222 ALLOW IN Anywhere
[ 4] 2222 ALLOW IN 192.168.156.2
[ 5] 22 (v6) ALLOW IN Anywhere (v6)
[ 6] 2222 (v6) ALLOW IN Anywhere (v6)
```

Take a look at above output. Priority is reassigned to [ 2 ]. Now if I try to connect to port 2222 from 192.168.157.0/24, my connection will be denied.

## 5. UFW logging



To disable UFW logging, run the following command:

```
$ sudo ufw logging off
Logging disabled
```

To enable UFW logging, run:

```
$ sudo ufw logging on
Logging enabled
```

[!\[\]\(17413706fd4997a1a4bdf85c6864eee1\_img.jpg\) FACEBOOK](#) [!\[\]\(f419710cbe076aa30a9c6c031b5cbe84\_img.jpg\) TW](#)[!\[\]\(faf942dc3e59ce8eb64b4ac481eca7e0\_img.jpg\) LINKEDIN](#) [!\[\]\(f6b0299e0b5e4340e509b71914970da0\_img.jpg\) YO](#)[!\[\]\(cf531ed27e91483460120fcc057b3901\_img.jpg\) EMAIL](#) [!\[\]\(34fde9b7c74442c0438f550a41236260\_img.jpg\) RE](#)[!\[\]\(d3102649f02e825ddb76dc3de0190154\_img.jpg\) RSS](#)

There is four-levels of logging, namely **low**, **medium**, **high**, and **full**. Depending upon the level you choose the logs will be generated under `/var/log/ufw.log` file. By default, the log level will be **low**.

You can use the below command to set to the suitable level you want:

```
$ sudo ufw logging [ high | medium | Full | Low ]
```

To check the status of logging and log level, run the `ufw status` command and look for logging entry.

```
$ sudo ufw status verbose
Status: active
Logging: on (high)
Default: deny (incoming), allow (outgoing), deny (routed)
New profiles: skip
```

## 6. Application profiles

When you install any packages using using your package manager (E.g. `apt` or `pacman`), an application profile that defines rules for that package will be created in `ufw`.

For example, if you are installing an OpenSSH server using `apt`, then profile will be created for port 22. All application profiles are stored under `/etc/ufw/applications.d` directory.

To get the list of application profiles, run the following command:

```
$ sudo ufw app list
```

**Sample output:**

```
Available applications:
CUPS
OpenSSH
```

This is a test machine. I have installed only OpenSSH. So you see only two profiles.

To get detailed information about that profile and what policy it enforces, run the following command:

```
$ sudo ufw app info 'OpenSSH'
```

Sample output:

```
Profile: OpenSSH
Title: Secure shell server, an rshd replacement
Description: OpenSSH is a free implementation of the Secure Shell protocol.
Port:
22/tcp
```

## 7. Reset UFW firewall to default policy

If you wish to clean all the rules you created and reset to default, you can do that by running the `ufw reset` command.

```
$ sudo ufw reset
```

Sample output:

```
Resetting all rules to installed defaults. Proceed with operation (y|n)? y
Backing up 'user.rules' to '/etc/ufw/user.rules.20210705_131655'
Backing up 'before.rules' to '/etc/ufw/before.rules.20210705_131655'
Backing up 'after.rules' to '/etc/ufw/after.rules.20210705_131655'
Backing up 'user6.rules' to '/etc/ufw/user6.rules.20210705_131655'
Backing up 'before6.rules' to '/etc/ufw/before6.rules.20210705_131655'
Backing up 'after6.rules' to '/etc/ufw/after6.rules.20210705_131655'
```

When you run this command your current rules will be backed up before resetting to the default profile.

Ufw also has many graphical front-ends. One of them is **Gufw**.

## 8. Gufw, a graphical front-end to UFW

Some of you may not be comfortable with command line mode. Fortunately, there is a graphical front-end for UFW available.

Gufw is a graphical front-end application to manage the Uncomplicated Firewall (UFW) in Linux. Gufw is mainly developed to install and configure firewall for Linux desktops.

Refer the following guide to learn how to install and configure Gufw on Linux:

- [How To Setup Firewall With Gufw On Linux Desktop](#)

## Conclusion

In this guide, we have discussed what is UFW, how to install and setup UFW firewall on Linux with example commands.

Now it is your turn to test ufw on your machine. I suggest test ufw in any Virtual machine before implementing it on your desktop or server.

Resource:

- [Ubuntu documentation](#)

[f FACEBOOK](#)[X TW](#)[in LINKEDIN](#)[▶ YO](#)[✉ EMAIL](#)[RE](#)[RSS](#)

- [How To Improve The Linux System's Security Using Firejail](#)

[FIREWALL](#)

[IPTABLES](#)

[LINUX](#)

[NETFILTER](#)

[OPEN SOURCE](#)

[SECURITY](#)

[UBUNTU](#)

[UFW](#)

[UNCOMPLICATED FIREWALL](#)

SEARCH

3 comments

19

[f](#)

[x](#)

[in](#)

[g](#)

[t](#)

[e](#)

[m](#)



KARTHICK

Karthick is a passionate software engineer who loves to explore new technologies. He is a public speaker and loves writing about technology especially about Linux and opensource.

KEEP IN TOUCH

[f](#) FACEBOOK

[X](#) TW

[in](#) LINKEDIN

[y](#) YO

[e](#) EMAIL

[r](#) RE

[RSS](#)

Previous post

Next post

How To Find All Installed Fonts From Commandline In Linux

How To Format Shell Programs Using Shfmt In Linux

YOU MAY ALSO LIKE

How To Setup Firewall With Gufw On Linux...

Published: July 22, 2021

3 COMMENTS

JASPER BRANNIGAN

REPLY

July 7, 2021 - 10:05 am

Do people still use “ufw”? All the cool kids switched to the far superior “firewalld” years ago.

PARIDE

REPLY

July 20, 2023 - 4:09 pm

Do people still use “firewalld”? All the cool kids is remained to the far ssuperior “iptables/ebtables” from the begin.

SK

REPLY

July 9, 2021 - 11:00 pm

I can’t speak for others. But I do use UFW in my Ubuntu server.

LEAVE A COMMENT

Your Comment

SEARCH

Name\*

Email\*








☐ Save my name, email, and website in this browser for the next time I comment.

\* By using this form you agree with the storage and handling of your data by this website.

SUBMIT

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

KEEP IN TOUCH

-  FACEBOOK
-  TW
-  LINKEDIN
-  YO
-  EMAIL
-  RE
-  RSS

ABOUT OSTECHNIX

OSTechNix (Open Source, Technology, Nix\*) regularly publishes the latest news, how-to articles, tutorials and tips & tricks about free and opensource software and technology.



Archives

Select Month

POPULAR POSTS

- 1

Yt-dlp Commands: TI  
Tutorial For Begin  
Published: September :
- 2

Linus Torvalds Defi  
Kernel's Removal o  
Maintainers  
Published: October 24
- 3

How To Fix Busybox  
Error On Ubuntu  
Published: August 6, :

[About](#) [Contact Us](#) [Privacy Policy](#) [Sitemap](#) [Terms and Conditions](#)

OSTechNix © 2024. All Rights Reserved. This site is licensed under [CC BY-NC 4.0](#).