

Firewall with ufw

ufw (Uncomplicated Firewall) is a software, which helps modifying iptable rules on your linux machine. You can add rules which then are applied to the iptables of your system, to modify the IP traffic. Here some examples:

- Deny all incoming traffic
- Allow incoming traffic only on specific ports
- Allow incoming traffic only on specific ip addresses
- Allow incoming traffic only on specific interfaces
- Deny outgoing traffic to a specific ip address

Before you start: SSH

The default rules of ufw is:

- Block all incoming traffic
- Allow all outgoing traffic

This means: **when you are on a machine via SSH and start ufw without adding a SSH rule, you will lock yourself out.** Therefore, in that case, see how you add a rule to allow incoming traffic on your SSH port, before enabling ufw.

Status and enabling the firewall

After installig ufw on your device, you can check the status via:

```
sudo ufw status
```

```
# or with more information
```

```
sudo ufw status verbose
```

The status command will tell you if ufw is active and list your current rules. Additionally with the verbose parameter, you get some more information, like the log level, the preset (the rules which are applied, when no rule matches) or what happens when a new profile is installed, but no rule matches (A profile is a name, e.g. "OpenSSH", which represents a port, e.g. 22).

Adding rules

The simplest rules are *allow*, *deny* and *reject*.

allow will allow connections explicitly, while *deny* and *reject* will prevent connections. The difference is, that *deny* will just drop the connection, while *reject* answers with a negative response.

You can also always use names of services (e.g. "ssh") or port numbers (e.g. 8080). Here are some examples:

```
# Allow incoming SSH traffic
sudo ufw allow in ssh

# Allow outgoing HTTP traffic
sudo ufw allow out http

# Deny outgoing TCP Port 22 (SSH) traffic
sudo ufw deny out 22/tcp

# Allow Port 22 traffic from a specific ip address (range)
sudo ufw allow from 192.168.1.0/24 to any port 22 proto tcp
# traffic from the ip addr 192.168.1.0/24 (in this example a range)
# to any ip address of this device
# port 22 with the tcp protocol

# Allow incoming http requests only on a specific ethernet interface
sudo ufw allow in on enp4s0 from any to any port 8080 proto tcp
```

You may ask the same question as I...

Why is it sometimes so short and sometimes so verbose?

The reason, as far my research is correct, is that the fully syntax is (according of the manual page):

```
ufw
  [--dry-run]
  [rule]
  [delete]
```

```
[insert NUM]
[prepend]
allow|deny|reject|limit
[in|out [on INTERFACE]]
[log|log-all]
[proto PROTOCOL]
[from ADDRESS [port PORT | app APPNAME ]]
[to ADDRESS [port PORT | app APPNAME ]]
[comment COMMENT]
```

I would simplify it as:

```
ufw
allow|deny|reject|limit
[in|out [on INTERFACE]]
[proto PROTOCOL]
[from ADDRESS [port PORT | app APPNAME ]]
[to ADDRESS [port PORT | app APPNAME ]]
[comment COMMENT]
```

Therefore you need something like "to any" before the port or "proto tcp", instead of "22/tcp".

The first few examples are a shorthand for simple rules.

Deleting rules

Lets say you added a rule using "allow in 22". How do you delete it now? You can delete it by describing the rule again, but with the keyword "delete" in front of it.

```
sudo ufw delete allow in 22
```

The delete command must be as specific as the command which created the rule.

A note about docker

There is one thing about docker. In docker you can work with ports, e.g. this command would open port 8080

```
docker run -p 8080:80 nginx
```

To allow this, docker can modify the iptables directly. Thus opening ports on your system without being visible or managed by ufw. You can accept it, if you want to have ports open anyways, when you open them using docker, or as far as I have seen, you can config docker accordingly. But this is not handled in this page.

Revision #1

Created 2026-09-06 20:56:25 UTC by Matthias

Updated 2026-09-06 21:35:49 UTC by Matthias