

# How to Set Up Firewall Policies using Uncomplicated Firewall (UFW)

Learn how to set up and configure firewall policies using Uncomplicated Firewall (UFW), a user-friendly interface for managing iptables on Linux systems.

# Contents

01	Introduction	3
02	Prerequisites	3
03	Install UFW	3
04	Configure Firewall Policies using UFW	5
05	Set Up Firewall Rules using UFW	6
06	Allow Specific Networking Ports	6
07	Deny Connection Requests to Specific Server Ports	7
08	Allow Network Connections to System Services	8
09	Set Up Directional UFW rules	9
010	Delete UFW Firewall Rules	10
011	Apply Firewall Rule Comments using UFW	12
012	Conclusion	14

# Introduction

Uncomplicated Firewall (UFW) is a network packet filtering application that runs on Linux servers and mostly Debian-based operating systems such as Ubuntu. UFW filters network packets based on the server interfaces, ports, and services to protect the system from internal or external security threats. To ensure improved control of a system's networking environment, use UFW to add a security layer of access to the system services.

This article explains how to set up firewall policies using UFW on a Vultr Cloud Server. You will explore the firewall policies, set up sample rules, and modify the direction of network requests to secure your server.

## Prerequisites

Before you start:

- Deploy a [Vultr Ubuntu server](#) to use as the management workstation.
- Access the server [using SSH](#) as a [non-root user with sudo privileges](#).
- [Update the server](#).

## Install UFW

1. Verify the UFW application status and verify that it's not available on the server. By default, the application is available on most server distributions but inactive.

CONSOLE

```
$ sudo ufw status
```

If you receive a `Status: inactive` message, UFW is unavailable on the server. Install it using the default `APT` package manager.

2. Install the UFW application package on the server.

CONSOLE

```
$ sudo apt install ufw
```

3. Enable UFW to start at boot time.

CONSOLE

```
$ sudo systemctl enable ufw
```

4. Add the default SSH port `22` to the UFW rules table to keep the remote session active.

CONSOLE

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

5. Start UFW.

CONSOLE

```
$ sudo ufw enable
```

6. View the UFW system service status and verify that it's actively running.

CONSOLE

```
$ sudo systemctl status ufw
```

Output:

```
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset:
   enabled)
   Active: active (exited) since Mon 2024-01-08 15:58:32 UTC; 2 months 20 days
   ago
```

```
Docs: man:ufw(8)
Main PID: 237 (code=exited, status=0/SUCCESS)
CPU: 4ms

Jan 08 15:58:32 ubuntu systemd[1]: Starting Uncomplicated firewall...
Jan 08 15:58:32 ubuntu systemd[1]: Finished Uncomplicated firewall.
```

## Configure Firewall Policies using UFW

UFW filters server traffic by detecting both incoming and outgoing network directions. By default, all outgoing connections are allowed through the firewall table, but incoming connections are blocked depending on the configured policies. Follow the steps below to configure UFW firewall policies to filter incoming and outgoing network connections on the server.

1. Allow all outgoing connections from the server.

CONSOLE

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

2. Deny all incoming connections to the server.

CONSOLE

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

The above command blocks all incoming connections to the server unless specified in the UFW table. Denying incoming connection requests to the server without any active rules blocks access to server using ports such as the SSH port `22`.

3. Deny all network forwarding requests on the server.

CONSOLE

```
$ sudo ufw default deny forward
```

The above command blocks all network forwarding requests on the server. For example, if the server runs as a NAT gateway, all forwarding connections are blocked when the above rule is active.

## Set Up Firewall Rules using UFW

UFW filters server network traffic based on available rules in the firewall table. While firewall policies explicitly operate on outgoing, incoming, and forwarded traffic, firewall rules operate on specific network ports on the server. Follow the sections below to allow, deny, or specify the direction of connection requests on the server.

## Allow Specific Networking Ports

1. Allow incoming connections to an essential port such as the HTTP port `80`.

CONSOLE

```
$ sudo ufw allow 80
```

2. View the UFW table and verify that the new firewall rule is successful.

CONSOLE

```
$ sudo ufw status
```

Output:

```
Status: inactive
```

To	Action	From
--	-----	----
22/tcp	ALLOW	Anywhere

3. Allow another port such as the SSH port `22` and define TCP as the connection protocol.

```
CONSOLE
```

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

The above rule allows all connection requests to the TCP port `22` and blocks any connections with another scheme such as UDP.

4. Allow the UDP DNS port `53`.

```
CONSOLE
```

```
$ sudo ufw 53/udp
```

The above rule enables connections to the UDP port `53` and blocks any non-matching rules such as TCP.

5. View the firewall table.

```
CONSOLE
```

```
$ sudo ufw status
```

6. Reload the UFW rules to apply the new firewall table changes.

```
CONSOLE
```

```
$ sudo ufw reload
```

## Deny Connection Requests to Specific Server Ports

1. Deny connection requests to an internal special service port such as the MySQL port `3306`.

```
CONSOLE
```

```
$ sudo deny 3306
```

2. View the UFW rules table.

```
CONSOLE
```

```
$ sudo ufw status
```

3. Reload the UFW rules to apply the firewall table changes.

```
CONSOLE
```

```
$ sudo ufw reload
```

## Allow Network Connections to System Services

UFW filters network connection requests based on the available system services and the target application service name. Specific services may run with a variation of ports that can be blocked by the firewall. Follow the steps below to allow network connections to the system services available on the server.

1. Allow network connections to a system service such as the Nginx web server.

```
CONSOLE
```

```
$ sudo ufw allow nginx-full
```

The above rule allows network connections to all ports associated with the Nginx system service.

2. Allow a service such as FTP through the firewall regardless of the system service daemon.

```
CONSOLE
```

```
$ sudo ufw allow ftp
```

3. View the UFW rules table to verify the new firewall changes.

```
CONSOLE
```

```
$ sudo ufw status
```

4. Reload the UFW rules to apply the new configuration changes.

```
CONSOLE
```

```
$ sudo ufw reload
```

## Set Up Directional UFW rules

Directional firewall rules define the source and destination of network traffic on the server. Follow the steps below to set up directional UFW rules that operate on specific network interfaces.

1. Allow incoming requests to the SSH port `22` from your public IP address.

```
CONSOLE
```

```
$ sudo ufw allow from 192.0.2.100 to any port 22
```

The above rule accepts SSH connection requests to the server but only from your public IP Address. UFW blocks all connections from other IP Addresses to the same SSH port.

2. Allow connection requests from a specific network interface such as `enp8s0` to the HTTP port `80`.

```
CONSOLE
```

```
$ sudo ufw allow in on enp8s0 proto tcp to any port 80
```

The above firewall rule accepts all connection requests to the HTTP port `80` from the server network interface `enp8s0`. UFW blocks all connection requests that don't match the interface as the source.

3. Allow outgoing connections through the public interface `enp1s0` on port `443`.

CONSOLE

```
$ sudo ufw allow out on enp1s0 proto tcp to any port 443
```

4. Deny connection requests to the SSH port to all hosts on the public network IP `192.0.2.100` while accepting connections from other addresses such as the local network address.

CONSOLE

```
$ sudo ufw deny from 192.0.2.500 to any port 22
```

5. View the UFW rules table.

CONSOLE

```
$ sudo ufw status
```

6. Reload the UFW rules to apply the firewall table changes.

CONSOLE

```
$ sudo ufw reload
```

## Delete UFW Firewall Rules

Depending on your server environment, you can remove UFW firewall rules from the filtering table and use the default UFW policies associated with the

connection type. Follow the steps below to remove multiple firewall rules set up by UFW.

1. View the UFW rules table and verify the target rule numbers to remove.

CONSOLE

```
$ sudo ufw status
```

Output:

```
Status: active
```

To	Action	From
--	-----	----
22	ALLOW	Anywhere
21/tcp	ALLOW	Anywhere
80/tcp on enp8s0	ALLOW	Anywhere
22 (v6)	ALLOW	Anywhere (v6)
21/tcp (v6)	ALLOW	Anywhere (v6)
80/tcp (v6) on enp8s0	ALLOW	Anywhere (v6)

Based on the above output, remove the HTTP port `80` directional rule with number `3` and the FTP service rule with number `2` from the firewall table.

2. Remove a firewall rule number from the UFW table. For example `3` to remove the HTTP port `80` rule.

CONSOLE

```
$ sudo ufw delete 3
```

Enter `y` when prompted to delete the firewall rule from the UFW table.

```
Deleting:
allow 80/tcp
Proceed with operation (y|n)? y
```

3. Remove another firewall rule number from the table. For example, rule number `5` to remove the FTP service.

```
CONSOLE
```

```
$ sudo ufw delete 5
```

4. View the UFW rules table to verify the new firewall changes.

```
CONSOLE
```

```
$ sudo ufw status
```

Output:

```
Status: active
```

To	Action	From
--	-----	----
22	ALLOW	Anywhere
22 (v6)	ALLOW	Anywhere (v6)
21/tcp (v6)	ALLOW	Anywhere (v6)
80/tcp (v6) on enp8s0	ALLOW	Anywhere (v6)

Based on the above output, UFW removes all firewall rules that match the entry number. Any similar entries are not affected by the delete operation such as the IPV6 connection rules because they don't match the target rule number.

## Apply Firewall Rule Comments using UFW

Firewall rules are numerical or text-based depending on your target connection type. Apply firewall rule comments to identify specific firewall table entries depending on the target effect by following the steps below.

1. Apply a comment to the custom SSH firewall rule that permits your public IP address. For example, `My secure public IP SSH Connection`.

```
CONSOLE
```

```
$ sudo ufw allow from 192.0.2.100 to any port 22 comment "My secure public IP SSH Connection"
```

2. Apply a comment on a special port such as 3306 for the MySQL port. For example, MySQL database server port for external access.

CONSOLE

```
$ sudo ufw allow 3306/tcp comment "MySQL database server port for external access"
```

3. Reload the UFW rules table to apply the configuration changes.

CONSOLE

```
$ sudo ufw reload
```

4. View the UFW rules table and verify the new firewall rule comments.

CONSOLE

```
$ sudo ufw status
```

Output:

```
Status: active

To Action From
--
22 ALLOW 192.0.2.100 # My secure
public IP SSH Connection
3306/tcp ALLOW Anywhere # MySQL
database server port for external access
3306/tcp (v6) ALLOW Anywhere (v6) # MySQL
database server port for external access
```

## Conclusion

---

You have set up firewall policies and rules using UFW on a cloud server. Depending on your networking environment, UFW filters network requests on the server while an advanced firewall such as the Vultr Firewall further tightens your server security by filtering requests before forwarding them to the server. Modify the configuration files in the UFW data directory `/etc/ufw` to set up advanced firewall rules which include before or after specific server events. For more information about UFW, visit the application manual page using the `man ufw` command.



VULTR

