

[Home](#) » [Fedora](#) » **Install Fail2ban with Firewalld on Fedora Linux 35**

Install Fail2ban with Firewalld on Fedora Linux 35

Last Updated on: Saturday, January 29, 2022 by Joshua James

[Fail2ban](#) is an intrusion prevention software framework that protects computer servers from primarily brute-force attacks, banning bad user agents, banning URL scanners, and much more. Fail2ban achieves this by reading access/error logs of your server or web applications. Fail2ban is coded in the python programming language.

The following tutorial will teach you **how to install Fail2ban on Fedora 35 Workstation or Server and some basic setup and tips.**

Table of Contents



Prerequisites

- **Recommended OS:** [Fedora Linux 35](#).
- **User account:** A user account with sudo or root access.

Update Operating System

Update your **Fedora** operating system to make sure all existing packages are up to date:

Click To Copy!

```
sudo dnf upgrade --refresh -y
```

The tutorial will be using the **sudo command** and **assuming you have sudo status.**

To verify sudo status on your account:

[Click To Copy!](#)

```
sudo whoami
```

Example output showing sudo status:

[Click To Copy!](#)

```
[joshua@fedora ~]$ sudo whoami  
root
```

To set up an existing or new sudo account, visit our tutorial on [Adding a User to Sudoers on Fedora](#).

To use the **root account**, use the following command with the root password to log in.

[Click To Copy!](#)

```
su
```

Install Dependency Required

Before you proceed with the installation, run the following command to install or check that the package dnf-plugins-core is installed on your Fedora desktop.

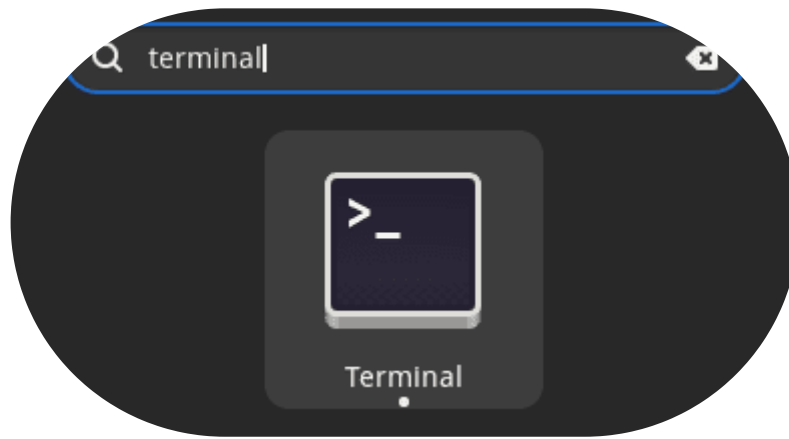
[Click To Copy!](#)

```
sudo dnf install dnf-plugins-core -y
```

By default, this should be installed.

The tutorial will utilize the terminal, which can be found in your show applications menu.

Example:



Install & Configure Firewallld

By default, Fedora comes with firewallld installed. To verify this, use the following command:

Click To Copy!

```
dnf info firewallld
```

Example output:

```
joshua@fedora:~  
Last metadata expiration check: 0:13:20 ago on Tue 11 Jan 2022 09:32:06.  
Installed Packages  
Name       : firewallld  
Version    : 1.0.1  
Release    : 2.fc35  
Architecture : noarch  
Size       : 2.0 M  
Source     : firewallld-1.0.1-2.fc35.src.rpm  
Repository : @System  
From repo  : fedora  
Summary    : A firewall daemon with D-Bus interface providing a dynamic  
           : firewall  
URL        : http://www.firewalld.org  
License    : GPLv2+  
Description : firewallld is a firewall service daemon that provides a dynamic  
           : customizable firewall with a D-Bus interface.  
[joshua@fedora ~]$
```

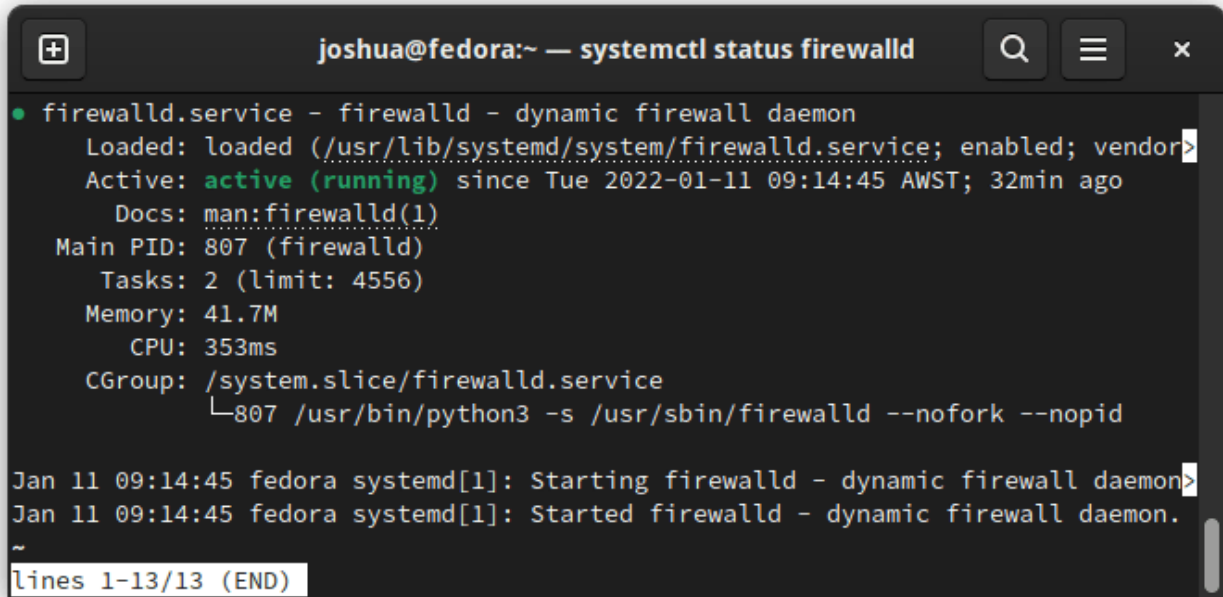
As you can see, this is installed on Fedora by default; also, it should be automatically enabled on your system.

To confirm this, use the following **systemctl** command:

[Click To Copy!](#)

```
systemctl status firewalld
```

Example output:

A terminal window titled 'joshua@fedora:~ — systemctl status firewalld' showing the status of the firewalld service. The output indicates the service is active and running. It lists details such as the service name, loaded state, active state with start time, documentation, main PID, tasks, memory usage, CPU usage, and CGroup. It also shows log messages for the service starting and starting successfully.

```
joshua@fedora:~ — systemctl status firewalld
• firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor>
  Active: active (running) since Tue 2022-01-11 09:14:45 AWST; 32min ago
  Docs: man:firewalld(1)
  Main PID: 807 (firewalld)
  Tasks: 2 (limit: 4556)
  Memory: 41.7M
  CPU: 353ms
  CGroup: /system.slice/firewalld.service
          └─807 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

Jan 11 09:14:45 fedora systemd[1]: Starting firewalld - dynamic firewall daemon>
Jan 11 09:14:45 fedora systemd[1]: Started firewalld - dynamic firewall daemon.
~
lines 1-13/13 (END)
```

Another handy trick with firewalld is to use the `firewall-cmd --state` command to verify if running or not:

[Click To Copy!](#)

```
sudo firewall-cmd --state
```

Example output:

[Click To Copy!](#)

```
running
```

If your firewalld is switched off, to start it use the following:

[Click To Copy!](#)

```
sudo systemctl start firewalld
```

To re-enable it to start on system boot, use the following:

[Click To Copy!](#)

```
sudo systemctl enable firewalld
```

Example output if successful:

[Click To Copy!](#)

```
Created symlink /etc/systemd/system/dbus-  
org.fedoraproject.Firewalld1.service →  
/usr/lib/systemd/system/firewalld.service.  
Created symlink /etc/systemd/system/multi-  
user.target.wants/firewalld.service →  
/usr/lib/systemd/system/firewalld.service.
```

If your firewall has been removed, you can re-install firewalld with the following command:

[Click To Copy!](#)

```
sudo dnf install firewalld
```

Finally, to verify the current rules before any new ones are added by fail2ban, list the existing ones to get familiar with firewalld:

[Click To Copy!](#)

```
sudo firewall-cmd --list-all
```

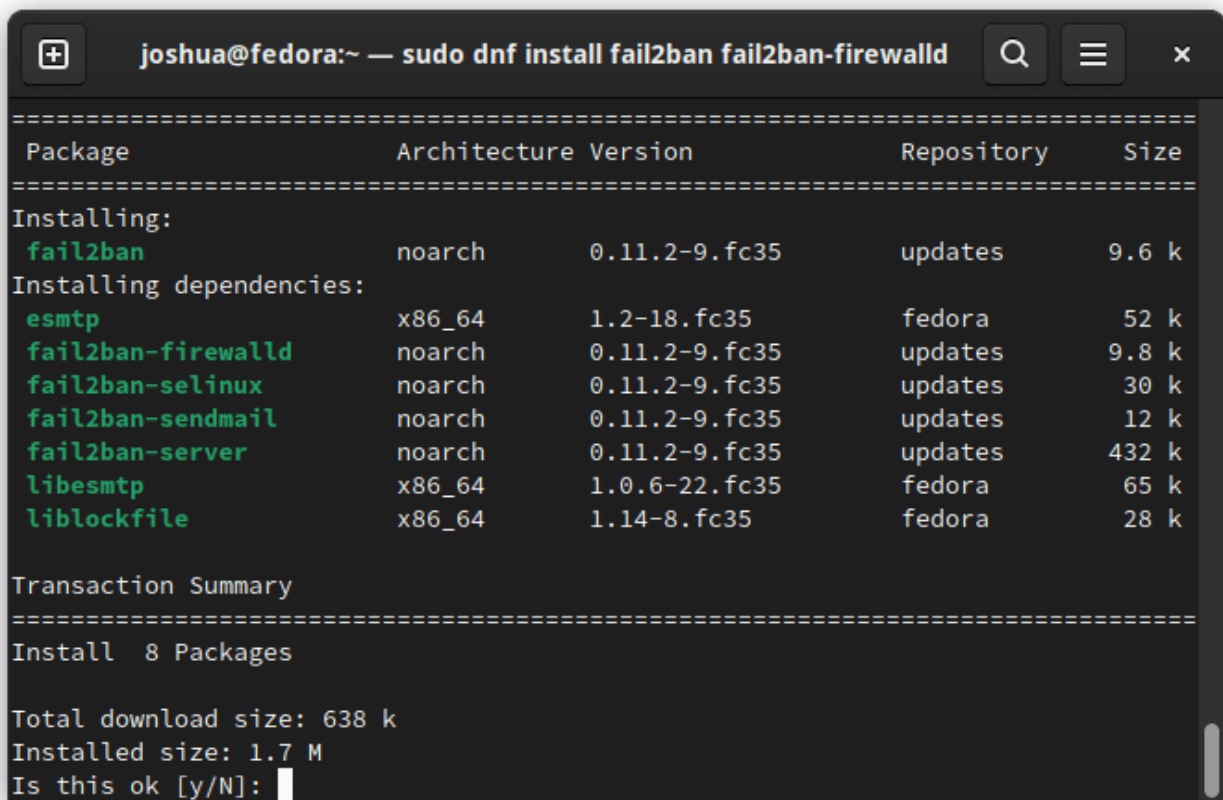
Install Fail2ban on Fedora

The next part of the tutorial is installing **fail2ban** and the addition package **fail2ban-firewalld**, which will correctly configure Fail2ban to work with firewalld for future use.

[Click To Copy!](#)

```
sudo dnf install fail2ban fail2ban-firewalld
```

Example output:



```
joshua@fedora:~ — sudo dnf install fail2ban fail2ban-firewalld
=====
Package                Architecture Version      Repository    Size
=====
Installing:
fail2ban                noarch       0.11.2-9.fc35 updates       9.6 k
Installing dependencies:
esmtplib                x86_64       1.2-18.fc35   fedora        52 k
fail2ban-firewalld      noarch       0.11.2-9.fc35 updates       9.8 k
fail2ban-selinux        noarch       0.11.2-9.fc35 updates       30 k
fail2ban-sendmail       noarch       0.11.2-9.fc35 updates       12 k
fail2ban-server         noarch       0.11.2-9.fc35 updates      432 k
libesmtplib             x86_64       1.0.6-22.fc35 fedora        65 k
liblockfile             x86_64       1.14-8.fc35   fedora        28 k

Transaction Summary
=====
Install 8 Packages

Total download size: 638 k
Installed size: 1.7 M
Is this ok [y/N]:
```

TYPE Y, then press the **ENTER KEY** to proceed with the installation.

By default, fail2ban will not be active, so you must start it manually with the following **systemctl** command:

[Click To Copy!](#)

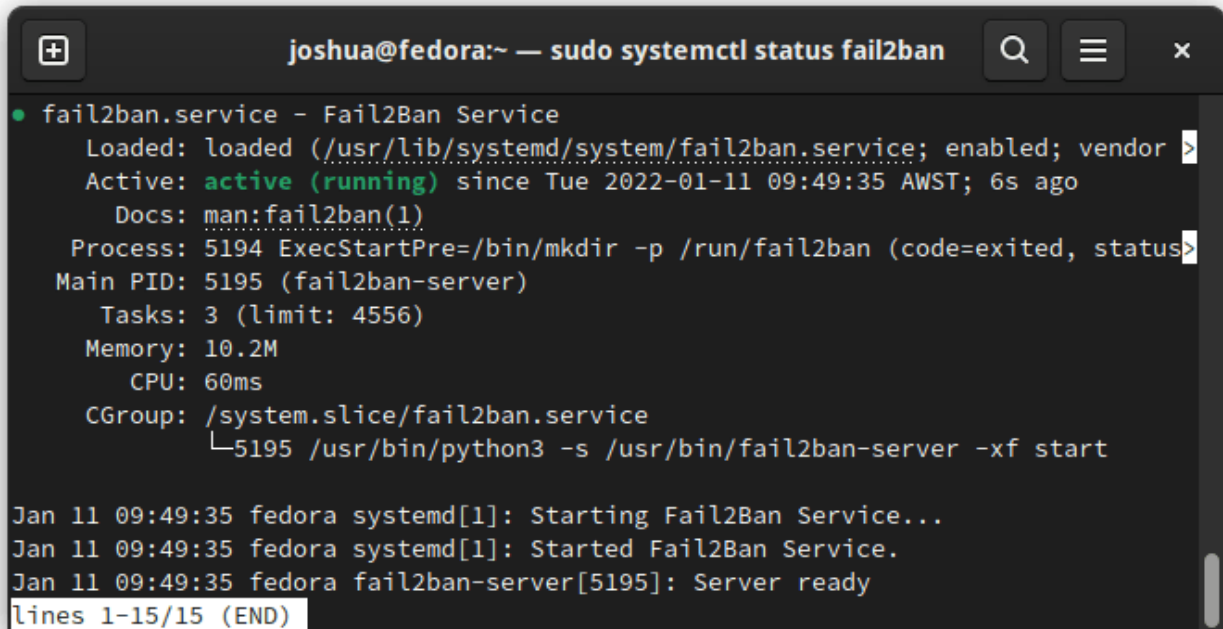
```
sudo systemctl enable fail2ban --now
```

Verify the status with the following command:

[Click To Copy!](#)

```
sudo systemctl status fail2ban
```

Example output:

A terminal window titled 'joshua@fedora:~ — sudo systemctl status fail2ban' showing the status of the fail2ban service. The output indicates the service is loaded, enabled, and active (running) since Tue 2022-01-11 09:49:35 AWST. It shows the process details for fail2ban-server and the start logs from Jan 11 09:49:35.

```
joshua@fedora:~ — sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; vendor
   Active: active (running) since Tue 2022-01-11 09:49:35 AWST; 6s ago
     Docs: man:fail2ban(1)
   Process: 5194 ExecStartPre=/bin/mkdir -p /run/fail2ban (code=exited, status
 Main PID: 5195 (fail2ban-server)
    Tasks: 3 (limit: 4556)
   Memory: 10.2M
      CPU: 60ms
   CGroup: /system.slice/fail2ban.service
           └─5195 /usr/bin/python3 -s /usr/bin/fail2ban-server -xf start

Jan 11 09:49:35 fedora systemd[1]: Starting Fail2Ban Service...
Jan 11 09:49:35 fedora systemd[1]: Started Fail2Ban Service.
Jan 11 09:49:35 fedora fail2ban-server[5195]: Server ready
lines 1-15/15 (END)
```

Lastly, verify the version and build of fail2ban:

[Click To Copy!](#)

```
fail2ban-client --version
```

Example output:

[Click To Copy!](#)

```
Fail2Ban v0.11.2
```

How to Configure Fail2ban

After completing the installation, the next step is setup and basic configuration.

Fail2ban comes with two configuration files which are located

in **/etc/fail2ban/jail.conf** and the default fail2ban jail **/etc/fail2ban/jail.d/00-**

firewalld.conf. Do not modify these files. The original set-up files are your originals and will be replaced in any update to Fail2ban in the future.

Now you may wonder how you set up Fail2ban if you update and lose your settings. Simple, we create copies ending in **.local** instead of **.conf** as fail2ban will always read **.local** files first before loading **.conf**.

To do this, use the following commands.

[Click To Copy!](#)

```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

By default, **jail.local** is set up to use **IPTABLES**. To keep things simple, instead of using the **00-firewalld.conf/jail** and creating your rules from scratch, open **jail.local** and **go to line 208 and replace**:

Open jail.local:

[Click To Copy!](#)

```
sudo nano /etc/fail2ban/jail.local
```

Find Old code (IPTABLES):

[Click To Copy!](#)

```
banaction = iptables-multiport  
banaction_allports = iptables-allports
```

Replace with (FIREWALLD):

[Click To Copy!](#)

```
banaction = firewallcmd-rich-rules[actiontype=]  
banaction_allports = firewallcmd-rich-rules[actiontype=]
```


Next, the tutorial will run over some settings that you can use or modify to your liking. Note that most settings are commented out; the tutorial will uncomment the lines in question or modify the existing ones in the example settings.

Remember, these are optional settings, and you can set whatever you like if you know more about fail2ban and have the confidence.

Ban Time Increment

The first setting you will come across is **Ban time increments**. You should enable this every time the attacker returns. It will increase the ban time, saving your system from constantly re-banning the same IP if your ban time lengths are minor; for example, 1 hour, you would want this to be longer if the attacker returns x5 times.

The tutorial recommends uncommenting the following multiplier line for a good range of continual banning of malicious IP addresses to keep the list growing too large on infrequent attackers.

[Click To Copy!](#)

```
# following example can be used for small initial ban time
(bantime=60) - it gr>
# for bantime=60 the multipliers are minutes and equal: 1 min, 5
min, 30 min, 1>

bantime.multipliers = 1 5 30 60 300 720 1440 2880  ## UNCOMMENT
THIS LINE##
```

This is the most effective at short ban times then increasing, but you can change these numbers or even pick another multiplier system altogether in the configuration.

Whitelist IPs in Fail2ban

Next in the list, we come across whitelisting options, uncomment the following and address any IP addresses you want to be whitelisted.

[Click To Copy!](#)

```
ignoreip = 127.0.0.1/8 ::1 180.53.31.33 (example IP address)
```

Make sure to space or comma between the IP addresses. You can whitelist IP ranges as well.

Default Ban Time Set-Up

Ban time defaults are 10 minutes with 10 minutes finder on 5 retries. An explanation of this is Fail2ban jail with filtering will ban your attacker for 10 minutes after it has retried the same attack in 10 minutes (find time) x 5 times (retries). You can set some default ban settings here.

However, when you get to jails, it's advised to set different ban times as some banning should automatically be longer than others, including retries that should be less or more.

E-Mail set up with Fail2ban

You can set an e-mail address for Fail2ban to send reports. The default **action = % (action_mw)s** that bans the offending IP and sends an e-mail with a whois report for you to review. However, in your action.d folder, other e-mail options exist for reporting to not only yourself but sending out e-mails to blacklist providers and the attacker's ISP to report.

Example below:

[Click To Copy!](#)

```
# Destination email address used solely for the interpolations in
# jail.{conf,local,d/*} configuration files.
destemail = admin@example.com

# Sender e-mail address used solely for some actions
sender = fail2ban@example.com
```

Note, by default, Fail2ban uses **sendmail MTA** for email notifications. You can change this to the **mail function** by doing the following:

Change from:

[Click To Copy!](#)

```
mta = sendmail
```

Change to:

[Click To Copy!](#)

```
mail = sendmail
```

Fail2ban Jails

Next, we come to jails. You can set pre-defined jails with filters and actions created by the community covering many popular server applications. You can make custom jails or find external ones on various **gists** and community websites; however, we will set up the default Fail2ban package jails.

Default set up for all the jails as per the picture below. Notice how nothing is enabled.

Example below:

[Click To Copy!](#)

```
[apache-badbots]
# Ban hosts which agent identifies spammer robots crawling the web
# for email addresses. The mail outputs are buffered.
port      = http,https
logpath   = %(apache_access_log)s
bantime   = 48h
maxretry  = 1
```

So, we have an Apache 2 HTTP server, and like filter/ban bad bots, all you need to do is add `enabled = true` as the example below.

[Click To Copy!](#)

```
[apache-badbots]
# Ban hosts which agent identifies spammer robots crawling the web
# for email addresses. The mail outputs are buffered.
enabled   = true
port      = http,https
```

```
logpath    = %(apache_access_log)s
bantime    = 48h
maxretry   = 1
```

Notice how the max retry equals 1, and the ban time is 48H. This is an individual max retry and bans length setting for this jail that will automatically increase with the ban multiplier we set up earlier in the guide. If any of the filters are missing, you can add them as an example.

[Click To Copy!](#)

```
[apache-noscript]
enabled = true
port    = http,https
logpath = %(apache_error_log)s
```

Change above the following example below:

[Click To Copy!](#)

```
[apache-noscript]
enabled = true
port    = http,https
logpath = %(apache_error_log)s
bantime = 1d
maxretry = 3
```

Next, you would like to have different actions than specified in your default set up in `/etc/fail2ban/jail.local`, additional actions you can find in `action.d` directory. Different actions from this directory can be easily set up by following directions inside those action configuration lines in the files, remembering to rename them first to **.jail** over **.conf**, and then adding the following to your jail setup.

[Click To Copy!](#)

```
[apache-botsearch]
enabled = true
port    = http,https
logpath = %(apache_error_log)s
banaction = action_mw
```

```
cloudflare  
bantime = 72h  
maxretry = 1
```

As you above, the example **added action_mw**, so it automatically bans as per our default action and emails us a report with whois, then the following action, if you use Cloudflare, it will ban the IP address on the Cloudflare service as well.

Remember, Cloudflare needs setting up before use. Read the action.d file **cloudflare.conf**.

Once you are happy with your set-up, do the following command to restart fail2ban to load your new jails.

[Click To Copy!](#)

```
sudo systemctl restart fail2ban
```

Examples of using Fail2ban-client

Now that you are up and running with Fail2ban, you need to know some basic operating commands. We do this by using the fail2ban-client command. You may need to have sudo privileges, depending on your setup.

Ban an IP address:

[Click To Copy!](#)

```
sudo fail2ban-client set apache-botsearch banip <ip address>
```

Unban an IP address:

[Click To Copy!](#)

```
sudo fail2ban-client set apache-botsearch unbanip <ip address>
```

Command to bring up the help menu if you need to find additional settings or get help on a particular one.

[Click To Copy!](#)

```
sudo fail2ban-client -h
```

Checking Firewalld and Fail2ban

By default, firewalld should be configured to automatically be banning any IP that fail2ban actions a ban on. To see if this is indeed working correctly, use the following command:

A quick test is the located in your jail **[SSHD]** and placing **enabled = true** even if you are not using this jail as it is just a test then using the following ban command:

[Click To Copy!](#)

```
sudo fail2ban-client set sshd banip 192.155.1.7
```

Now list the firewall list rich rules as follows:

[Click To Copy!](#)

```
firewall-cmd --list-rich-rules
```

Example output:

[Click To Copy!](#)

```
rule family="ipv4" source address="192.155.1.7" port port="ssh"  
protocol="tcp" reject type="icmp-port-unreachable"
```

As you can see, fail2ban and firewalld are working correctly for a live environment.

How to Monitor Fail2ban Logs

Many common mistakes are setting up jails and walking away without testing or monitoring them. Reviewing logs is essential, which the fail2ban log is in its default path **`/var/log/fail2ban.log`**.

If you have a server receiving decent traffic, an excellent command to watch live to see any issues and keep an eye on it as you work in other servers is to use the **tail -f** command below.

[Click To Copy!](#)

```
tail -f /var/log/fail2ban.log
```

The command can come in handy for spot-checking without diving into logging.

Another option is to print the last X amount of lines. For example, X is replaced with 30 to print 30 lines by adding the **-n 30** flag.

[Click To Copy!](#)

```
tail -f /var/log/fail2ban.log -n 30
```

These are just some examples of reading logs, and grep can also be helpful.

How to Remove (Uninstall) Fail2ban

If you no longer require Fail2ban, to remove it from your system, use the following command:

[Click To Copy!](#)

```
sudo dnf autoremove fail2ban fail2ban-firewalld
```

Note, this will also remove all the unused dependencies installed with Fail2ban.

Comments and Conclusion

The tutorial has shown you the basics of installing Fail2ban on the Fedora 35 system and setting up some jails with the filters available.

Overall, Fail2ban is a potent tool when configured and maintained correctly, and you can set it up in many different ways from what the tutorial has shown. All servers or desktops require different settings and configurations. Fail2ban is actively developed and is a solid choice to deploy on your server in these times where attacks are becoming so frequent.

For further information, visit the [Fail2ban official documentation](#).

 [Fedora](#)

 [Fail2ban, Fedora 35, Firewalld](#)

< [Install Deepin Desktop Environment \(DDE\) on Fedora Linux 35](#)

> [Install ClamAV on Fedora Linux 35](#)