



# Ubuntu Shell


## How to Install Freenginx on Ubuntu (Ultimate Guide)

FEBRUARY 25, 2024 - 3 MINS READ



Freenginx gained a lot of attention after NGINX's core developer quit the company due to internal conflict and policy changes to initiate his own Freenginx project, a free and open-source version prioritizing community contributions over company interests.

Currently, it's new, and you can barely find any difference in usage, but

 **Tip Me** for fast improvements, fixes, and the introduction of exciting features in the future.

If you want to give it a try, then read this entire article to learn how to install Freenginx on Ubuntu from source with practical examples.

## How to Install Freenginx on Ubuntu

As of writing this article, the only available option to install Freenginx is via source code for Linux; for Windows users, the compressed file can be found in the download section.

In this article, we will focus on installing Freenginx on Ubuntu, so follow each step mentioned below one by one.

1. Open your terminal and execute the following command to update the system repository information:

```
$ sudo apt update
```

2. Install the required packages, dependencies, and libraries, along with the development tool for source code compilation.

```
$ sudo apt install -y wget build-essential libpcre3 libpcre3-dev  
zlib1g zlib1g-dev libssl-dev libgd-dev libxml2 libxml2-dev  
uuid-dev
```

3. Stop and remove the existing NGINX from your system to avoid potential conflict issues later.

```
$ sudo apt --purge remove nginx-*  
$ sudo apt autoremove
```

4. Now, visit the [official Freenginx download page](#) and download the source tarball file for the Freenginx version (mainline, stable, or legacy) you wish to install.

**nginx: download**

**Mainline version**

<a href="#">CHANGES</a>	<a href="#">nginx-1.25.3</a> <a href="#">pgp</a>	<a href="#">nginx/Windows-1.25.3</a> <a href="#">pgp</a>
-------------------------	--	--

**Stable version**

<a href="#">CHANGES-1.24</a>	<a href="#">nginx-1.24.0</a> <a href="#">pgp</a>	<a href="#">nginx/Windows-1.24.0</a> <a href="#">pgp</a>
------------------------------	--	--

**Legacy versions**

<a href="#">CHANGES-1.22</a>	<a href="#">nginx-1.22.1</a> <a href="#">pgp</a>	<a href="#">nginx/Windows-1.22.1</a> <a href="#">pgp</a>
<a href="#">CHANGES-1.20</a>	<a href="#">nginx-1.20.2</a> <a href="#">pgp</a>	<a href="#">nginx/Windows-1.20.2</a> <a href="#">pgp</a>
<a href="#">CHANGES-1.18</a>	<a href="#">nginx-1.18.0</a> <a href="#">pgp</a>	<a href="#">nginx/Windows-1.18.0</a> <a href="#">pgp</a>
<a href="#">CHANGES-1.16</a>	<a href="#">nginx-1.16.1</a> <a href="#">pgp</a>	<a href="#">nginx/Windows-1.16.1</a> <a href="#">pgp</a>
<a href="#">CHANGES-1.14</a>	<a href="#">nginx-1.14.2</a> <a href="#">pgp</a>	<a href="#">nginx/Windows-1.14.2</a> <a href="#">pgp</a>
<a href="#">CHANGES-1.12</a>	<a href="#">nginx-1.12.2</a> <a href="#">pgp</a>	<a href="#">nginx/Windows-1.12.2</a> <a href="#">pgp</a>
<a href="#">CHANGES-1.10</a>	<a href="#">nginx-1.10.3</a> <a href="#">pgp</a>	<a href="#">nginx/Windows-1.10.3</a> <a href="#">pgp</a>
<a href="#">CHANGES-1.8</a>	<a href="#">nginx-1.8.1</a> <a href="#">pgp</a>	<a href="#">nginx/Windows-1.8.1</a> <a href="#">pgp</a>
<a href="#">CHANGES-1.6</a>	<a href="#">nginx-1.6.3</a> <a href="#">pgp</a>	<a href="#">nginx/Windows-1.6.3</a> <a href="#">pgp</a>

**free nginx**

[english](#)  
[русский](#)

[news](#)  
[about](#)  
[download](#)  
[security](#)  
[documentation](#)  
[faq](#)  
[books](#)  
[support](#)

Alternatively, you can also download it from your terminal via the [wget](#) command.

```
$ wget https://freenginx.org/download/nginx-1.24.0.tar.gz
```

Output:

```

ubuntushell@linux:~$ wget https://freenginx.org/download/nginx-1.24.0.tar.gz
--2024-02-16 21:58:56-- https://freenginx.org/download/nginx-1.24.0.tar.gz
Resolving freenginx.org (freenginx.org)... 144.76.182.204
Connecting to freenginx.org (freenginx.org)|144.76.182.204|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1112471 (1.1M) [application/octet-stream]
Saving to: 'nginx-1.24.0.tar.gz'

nginx-1.24.0.tar.gz      100%[=====>] 1.06M  376KB/s   in 2.9s

2024-02-16 21:59:00 (376 KB/s) - 'nginx-1.24.0.tar.gz' saved [1112471/1112471]

ubuntushell@linux:~$

```

5. Extract the downloaded tarball file and then move it into the extracted directory.

```
$ tar -zxvf nginx-*.tar.gz && cd nginx*/
```

Output:

```

nginx-1.24.0/auto/lib/openssl/conf
nginx-1.24.0/auto/lib/openssl/make
nginx-1.24.0/auto/lib/openssl/makefile.bcc
nginx-1.24.0/auto/lib/openssl/makefile.msvc
nginx-1.24.0/auto/lib/libxslt/conf
nginx-1.24.0/auto/lib/libgd/conf
nginx-1.24.0/auto/lib/libatomic/conf
nginx-1.24.0/auto/lib/libatomic/make
nginx-1.24.0/auto/lib/google-perftools/conf
nginx-1.24.0/auto/lib/geoip/conf
nginx-1.24.0/auto/cc/acc
nginx-1.24.0/auto/cc/bcc
nginx-1.24.0/auto/cc/cc
nginx-1.24.0/auto/cc/clang
nginx-1.24.0/auto/cc/conf
nginx-1.24.0/auto/cc/gcc
nginx-1.24.0/auto/cc/icc
nginx-1.24.0/auto/cc/msvc
nginx-1.24.0/auto/cc/name
nginx-1.24.0/auto/cc/owc
nginx-1.24.0/auto/cc/sunc
ubuntushell@linux:~/nginx-1.24.0$

```

6. Configure the custom path for the NGINX configuration file, access, and error log path with some NGINX modules using configure flags.

```
$ ./configure --prefix=/var/www/html --sbin-path=/usr/sbin/
nginx --conf-path=/etc/nginx/nginx.conf --http-log-path=/var/
log/nginx/access.log --error-log-path=/var/log/nginx/
error.log --with-pcre --lock-path=/var/lock/nginx.lock --pid-
path=/var/run/nginx.pid --with-http_ssl_module --with-
http_image_filter_module=dynamic --modules-path=/etc/nginx/
modules --with-http_v2_module --with-stream=dynamic --with-
http_addition_module --with-http_mp4_module
```

Output:

```
checking for PCRE JIT support ... found
checking for OpenSSL library ... found
checking for zlib library ... found
checking for GD library ... found
checking for GD WebP support ... found
creating objs/Makefile

Configuration summary
+ using system PCRE library
+ using system OpenSSL library
+ using system zlib library

nginx path prefix: "/var/www/html"
nginx binary file: "/usr/sbin/nginx"
nginx modules path: "/etc/nginx/modules"
nginx configuration prefix: "/etc/nginx"
nginx configuration file: "/etc/nginx/nginx.conf"
nginx pid file: "/var/run/nginx.pid"
nginx error log file: "/var/log/nginx/error.log"
nginx http access log file: "/var/log/nginx/access.log"
nginx http client request body temporary files: "client_body_temp"
nginx http proxy temporary files: "proxy_temp"
nginx http fastcgi temporary files: "fastcgi_temp"
nginx http uwsgi temporary files: "uwsgi_temp"
nginx http scgi temporary files: "scgi_temp"

ubuntushell@linux:~/nginx-1.24.6$
```

7. Once the custom configuration is completed, proceed to compile the NGINX source code and install it using the following two commands:

```
$ make  
$ sudo make install
```

Output:

```
    || cp conf/uwsgi_params '/etc/nginx'  
cp conf/uwsgi_params \  
    '/etc/nginx/uwsgi_params.default'  
test -f '/etc/nginx/scgi_params' \  
    || cp conf/scgi_params '/etc/nginx'  
cp conf/scgi_params \  
    '/etc/nginx/scgi_params.default'  
test -f '/etc/nginx/nginx.conf' \  
    || cp conf/nginx.conf '/etc/nginx/nginx.conf'  
cp conf/nginx.conf '/etc/nginx/nginx.conf.default'  
test -d '/var/run' \  
    || mkdir -p '/var/run'  
test -d '/var/log/nginx' \  
    || mkdir -p '/var/log/nginx'  
test -d '/var/www/html/html' \  
    || cp -R html '/var/www/html'  
test -d '/var/log/nginx' \  
    || mkdir -p '/var/log/nginx'  
test -d '/etc/nginx/modules' \  
    || mkdir -p '/etc/nginx/modules'  
test ! -f '/etc/nginx/modules/nginx_http_image_filter_module.so' \  
    || mv '/etc/nginx/modules/nginx_http_image_filter_module.so' \  
        '/etc/nginx/modules/nginx_http_image_filter_module.so.old'  
cp objs/nginx_http_image_filter_module.so '/etc/nginx/modules/nginx_http_image_filter_module.so'  
test ! -f '/etc/nginx/modules/nginx_stream_module.so' \  
    || mv '/etc/nginx/modules/nginx_stream_module.so' \  
        '/etc/nginx/modules/nginx_stream_module.so.old'  
cp objs/nginx_stream_module.so '/etc/nginx/modules/nginx_stream_module.so'  
make[1]: Leaving directory '/home/ubuntushell/nginx-1.24.0'  
ubuntushell@linux:~/nginx-1.24.0$
```

8. Freenginx is now successfully installed; you can check its version and configure options.

```
$ nginx -v  
# OR  
$ nginx -V
```

Output:

```
ubuntushell@linux:~/nginx-1.24.0$ nginx -v
nginx version: nginx/1.24.0
ubuntushell@linux:~/nginx-1.24.0$ nginx -V
nginx version: nginx/1.24.0
built by gcc 11.4.0 (Ubuntu 11.4.0-1ubuntu1~22.04)
built with OpenSSL 3.0.2 15 Mar 2022
TLS SNI support enabled
configure arguments: --prefix=/var/www/html --sbin-path=/usr/sbin/nginx --conf-path=/etc/nginx/nginx
.conf --http-log-path=/var/log/nginx/access.log --error-log-path=/var/log/nginx/error.log --with-pcr
e --lock-path=/var/lock/nginx.lock --pid-path=/var/run/nginx.pid --with-http_ssl_module --with-http_
image_filter_module=dynamic --modules-path=/etc/nginx/modules --with-http_v2_module --with-stream=dy
namic --with-http_addition_module --with-http_mp4_module
ubuntushell@linux:~/nginx-1.24.0$
```

9. Now, stop the nginx if it's already running, and then create a Freenginx systemd unit file at `/lib/systemd/system` path for managing Freenginx.

```
$ nginx -s stop
```

```
$ sudo vi /lib/systemd/system/nginx.service
```

And then paste this script:

```
[Unit]
Description=The Freenginx HTTP and reverse proxy server
After=syslog.target network-online.target remote-fs.target nss-lookup.t
Wants=network-online.target

[Service]
Type=forking
PIDFile=/var/run/nginx.pid
ExecStartPre=/usr/sbin/nginx -t
ExecStart=/usr/sbin/nginx
ExecReload=/usr/sbin/nginx -s reload
ExecStop=/bin/kill -s QUIT $MAINPID
PrivateTmp=true
```

```
[Install]
WantedBy=multi-user.target
```

Now, save the file by first pressing **Esc** to exit the editing mode, then type **:** followed by writing and executing the **wq!** command.

10. That's it! You can now manage Freenginx using the systemctl command, with the following examples of systemctl commands available for managing Freenginx:

Command	Description
---------	-------------

<code>sudo systemctl status nginx</code>	Check the status of Freenginx.
<code>sudo systemctl enable nginx</code>	Autostart the Freenginx at boot.
<code>sudo systemctl start nginx</code>	Start the Freenginx server.
<code>sudo systemctl stop nginx</code>	Stop the Freenginx server.
<code>sudo systemctl disable nginx</code>	Disable autostart for Freenginx at boot.



## Related Posts

