

```
devops@ubuntu-server:~$ sudo apt install nginx
[sudo] password for devops:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgd3 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libxpm4 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  libgd3 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libxpm4 nginx nginx-common
  nginx-core
0 upgraded, 11 newly installed, 0 to remove and 53 not upgraded.
Need to get 864 kB of archives.
After this operation, 2,950 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
devops@ubuntu-server:~$ sudo systemctl start nginx
[sudo] password for devops:
sudo: a password is required
devops@ubuntu-server:~$ systemctl status nginx
• nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-09-15 10:05:01 +0330; 24min ago
     Docs: man:nginx(8)
   Process: 24107 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited>
   Process: 24108 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=>
 Main PID: 24200 (nginx)
   Tasks: 2 (limit: 2220)
  Memory: 3.6M
    CPU: 29ms
   CGroup: /system.slice/nginx.service
           └─24200 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─24203 "nginx: worker process"

Sep 15 10:05:01 ubuntu-server systemd[1]: Starting A high performance web server and a reverse proxy>
Sep 15 10:05:01 ubuntu-server systemd[1]: Started A high performance web server and a reverse proxy>

devops@ubuntu-server:~$ mkdir -p /var/www/mysite
mkdir: cannot create directory '/var/www/mysite': Permission denied
devops@ubuntu-server:~$ sudo mkdir -p /var/www/mysite
[sudo] password for devops:
devops@ubuntu-server:~$
```

```
devops@ubuntu-server:~$ curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
devops@ubuntu-server:~$
```



-2

```
sudo mkdir -p /var/www/mysite
```

```
sudo nano /var/www/mysite/index.html
```

```
GNU nano 6.2 /var/www/mysite/index.html
<!DOCTYPE html>
<html>
  <head>
    <title>My First Nginx Site</title>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

[Read 9 lines]

Help Write Out Where Is Cut Execute Location M-U Undo
Exit Read File Replace Paste Justify Go To Line M-E Redo

```
root /var/www/mysite;

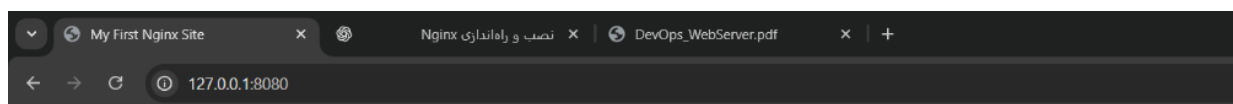
# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

# pass PHP scripts to FastCGI server
#
#location ~ \.php$ {
#    include snippets/fastcgi-php.conf;
```

```
devops@ubuntu-server:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
devops@ubuntu-server:~$ sudo systemctl restart nginx
devops@ubuntu-server:~$
```



Hello World

```
devops@ubuntu-server:~$ cd ~
devops@ubuntu-server:~$ mkdir myflaskapp
devops@ubuntu-server:~$ cd myflaskapp
devops@ubuntu-server:~/myflaskapp$ mkdir app.py
devops@ubuntu-server:~/myflaskapp$ _
```

```
GNU nano 6.2 app.py

from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello Worldddd"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=3000)
```

[Wrote 10 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo

```
root /var/www/mysite;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    proxy_pass http://127.0.0.1:3000;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    try_files $uri $uri/ =404;
}

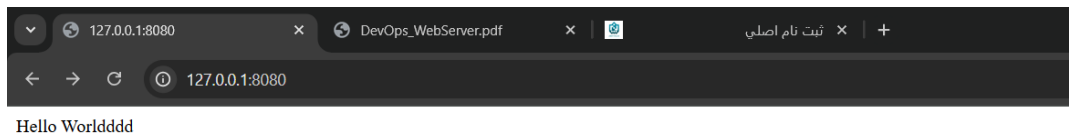
# pass PHP scripts to FastCGI server
#
#location ~ /\.php$ {
#    include snippets/fastcgi-php.conf;
#}
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo

```

devops@ubuntu-server:~/myflaskapp$ python3 app.py
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:3000
* Running on http://10.0.2.15:3000
Press CTRL+C to quit
127.0.0.1 - - [15/Sep/2025 11:42:50] "GET / HTTP/1.0" 200 -

```



-4

```

GNU nano 6.2 app.py
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World 3000"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=3000)

```

[Wrote 10 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
 ^X Exit ^R Read File ^_ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo

```
devops@ubuntu-server:~$ mkdir ~/myflaskapp2
devops@ubuntu-server:~$ cd ~/myflaskapp2
devops@ubuntu-server:~/myflaskapp2$ nano app.py_
```

```
GNU nano 6.2 app.py
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello world 3001"

if __name__=="__main__":
    app.run(host="0.0.0.0", port=3001)_

[ Wrote 9 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo
```

```
GNU nano 6.2 /etc/nginx/sites-available/default *
#
upstream flask_servers {
    server 127.0.0.1:3000;
    server 127.0.0.1:3001;
}

server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: This is preferred because it enables support for
    #       temporary redirecting of users to the SSL site for
```

```

devops@ubuntu-server:~/myflaskapp$ python3 app.py &
[1] 25631
devops@ubuntu-server:~/myflaskapp$ * Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:3000
* Running on http://10.0.2.15:3000
Press CTRL+C to quit
^C
devops@ubuntu-server:~/myflaskapp$ cd ~
devops@ubuntu-server:~$ cd ~/myflaskapp2
devops@ubuntu-server:~/myflaskapp2$ python3 app.py &
[2] 25632
devops@ubuntu-server:~/myflaskapp2$ * Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:3001
* Running on http://10.0.2.15:3001
Press CTRL+C to quit

devops@ubuntu-server:~$ curl http://127.0.0.1
127.0.0.1 - - [15/Sep/2025 12:15:50] "GET / HTTP/1.0" 200 -
Hello World 3001devops@ubuntu-server:~$ curl http://127.0.0.1
127.0.0.1 - - [15/Sep/2025 12:16:27] "GET / HTTP/1.0" 200 -
Hello World 3000devops@ubuntu-server:~$ curl http://127.0.0.1
127.0.0.1 - - [15/Sep/2025 12:16:30] "GET / HTTP/1.0" 200 -
Hello World 3001devops@ubuntu-server:~$ curl http://127.0.0.1
127.0.0.1 - - [15/Sep/2025 12:16:31] "GET / HTTP/1.0" 200 -
Hello World 3000devops@ubuntu-server:~$

```

-5

sudo apt install apache2-utils -y

```

devops@ubuntu-server:~$ sudo htpasswd -c /etc/nginx/.htpasswd Yalda
sudo: command not found
devops@ubuntu-server:~$ sudo htpasswd -c /etc/nginx/.htpasswd Yalda
[sudo] password for devops:
New password:
Re-type new password:
Adding password for user Yalda
devops@ubuntu-server:~$ _

```

```

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    root /var/www/mysite;
    index index.html;
    try_files $uri $uri/ =404;
}

location /protected{
    root /var/www/mysite;
    index index.html;

    auth_basic "Restricted Area";
    auth_basic_user_file /etc/nginx/.htpasswd;
}

location ~ /\.hit{
    deny all;
}
# pass PHP scripts to FastCGI server
#
#location ~ \.php$ {
#    include snippets/fastcgi-php.conf;
#
#    # With php-fpm (or other unix sockets):
#    fastcgi_pass unix:/run/php/php7.4-fpm.sock;
#    # With php-cgi (or other tcp sockets):

```

devops@ubuntu-server:~\$ sudo nginx -t

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok

nginx: configuration file /etc/nginx/nginx.conf test is successful

devops@ubuntu-server:~\$ sudo systemctl restart nginx

devops@ubuntu-server:~\$

