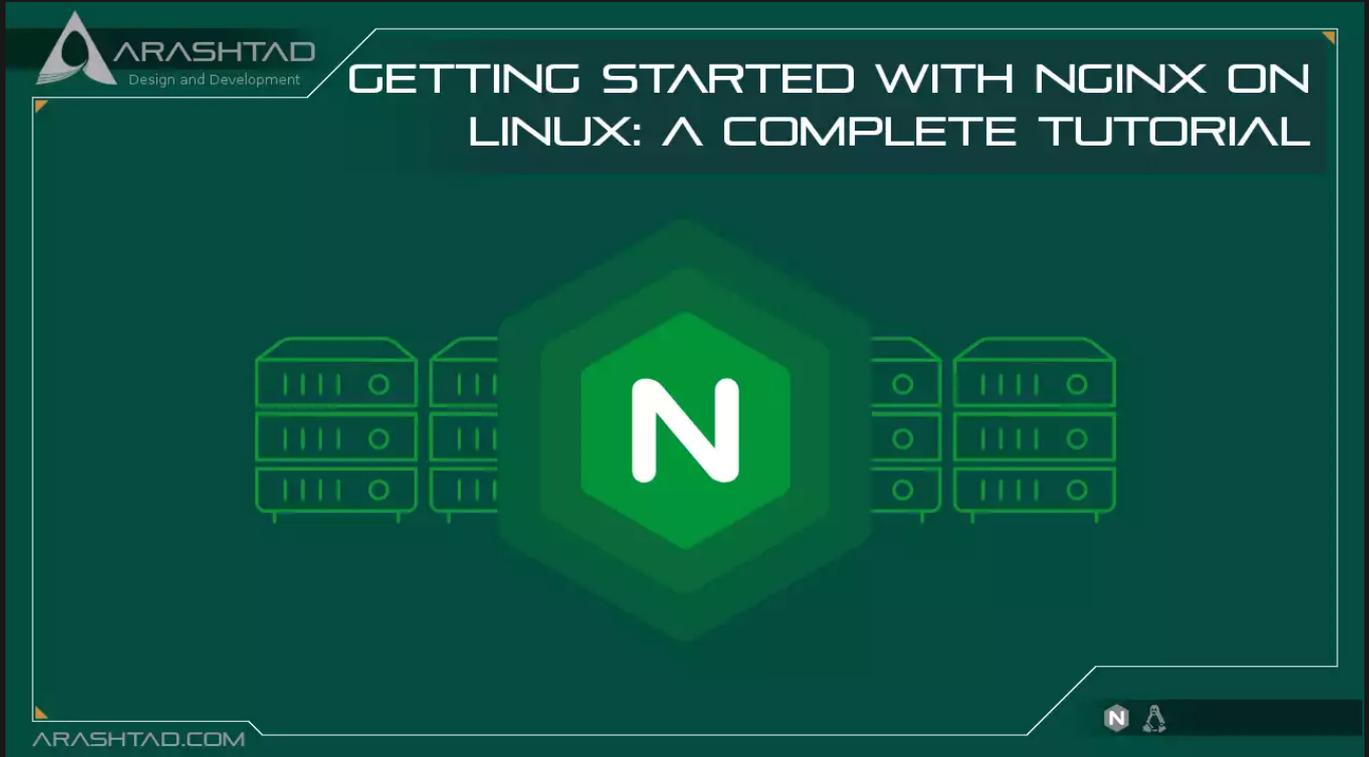


## Getting Started with Nginx on Linux: a Complete Tutorial

No comments



Nginx

*Nginx pronounced Engine-Ex, is a popular and open-source, lightweight, and high-performance web server software that also acts as a reverse proxy, load balancer, mail proxy, and HTTP cache. Nginx is easy to configure in order to serve static web content or to act as a proxy server. It can be deployed to also serve dynamic content on the network using FastCGI, SCGI handlers for scripts, WSGI application servers or Phusion Passenger modules, and it can serve as a software load balancer. Nginx uses an asynchronous event-driven approach, rather than threads, to handle requests. Nginx's modular event-driven architecture can provide predictable performance under high loads.*

*In this tutorial, we are going to get started with Nginx on Linux and use the terminal commands to install and configure a test on it. You will get familiar with all the codes and commands for setting Nginx up and running on your operating system.*

## What you need to get started:

1. This tutorial is based on Linux. If you are working with Ubuntu 20.04 Linux or Linux Mint, or any other OS of the Linux family, you have a suitable operating system for the following tutorial.
2. A user account with sudo or root privileges.
3. Access to a terminal window/command line

## Getting Started with Nginx

### 1. Installation

First off, you need to update software repositories. This helps make sure that the latest updates and patches are installed. Open a terminal window and enter the following: `sudo apt-get update` Now, to install Nginx from Ubuntu repository, enter the following command in the terminal: `sudo apt-get install nginx` If you are on Fedora, you should instead enter this command to install Nginx. `sudo dnf install nginx` And if you are on CentOS or RHEL, the installation is done using this command: `sudo yum install epel-release && yum install nginx` finally, we test the installation success by entering: `nginx -v` If the installation has been successful, You should get a result like this:  
`nginx version: nginx/1.18.0 (Ubuntu)`

### 2. Controlling the Nginx Service

Next, we should get familiar with the controlling commands. Using these commands, you will be able to start, enable, stop and disable the Nginx. First off, we should check the status of Nginx service. To do so, you can use the following command: `sudo systemctl status nginx` And you can see the result:

```
mohamad@mohamad-Lenovo-G510:~$ sudo systemctl status nginx
[sudo] password for mohamad:
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: ena
   Active: active (running) since Tue 2022-08-09 19:59:52 +0430; 1h 21min ago
     Docs: man:nginx(8)
   Process: 30735 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_proce
   Process: 30736 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (
 Main PID: 30737 (nginx)
    Tasks: 5 (limit: 4475)
   Memory: 4.2M
   CGroup: /system.slice/nginx.service
           └─30737 nginx: master process /usr/sbin/nginx -g daemon on; master
             └─30738 nginx: worker process
               └─30739 nginx: worker process
                 └─30740 nginx: worker process
                   └─30741 nginx: worker process

Aug 09 19:59:52 mohamad-Lenovo-G510 systemd[1]: Starting A high performance web
Aug 09 19:59:52 mohamad-Lenovo-G510 systemd[1]: Started A high performance web
lines 1-18/18 (END)
```

As you can it is activated and up and running. If it is not activated, you can first start by entering this command in the terminal: `sudo systemctl start nginx` And then, you will be able to enable it using the following command: `sudo systemctl enable nginx` If you want to stop the Nginx web service, you can first stop it: `sudo systemctl stop nginx` And then disable it: `sudo systemctl disable nginx` Also, if you want to reload the Nginx web service, you can use the following command: `sudo systemctl reload nginx` And for a hard restart, there is a command as below: `sudo systemctl restart nginx`

### 3. UnComplicated Firewall Commands:

Nginx needs access through the system's firewall. To do this, Nginx installs a set of profiles for the Ubuntu default ufw (Uncomplicated Firewall). To display the available Nginx profiles use this command: `sudo ufw app list` And you can see the result. Neglect the results other than that of Nginx.

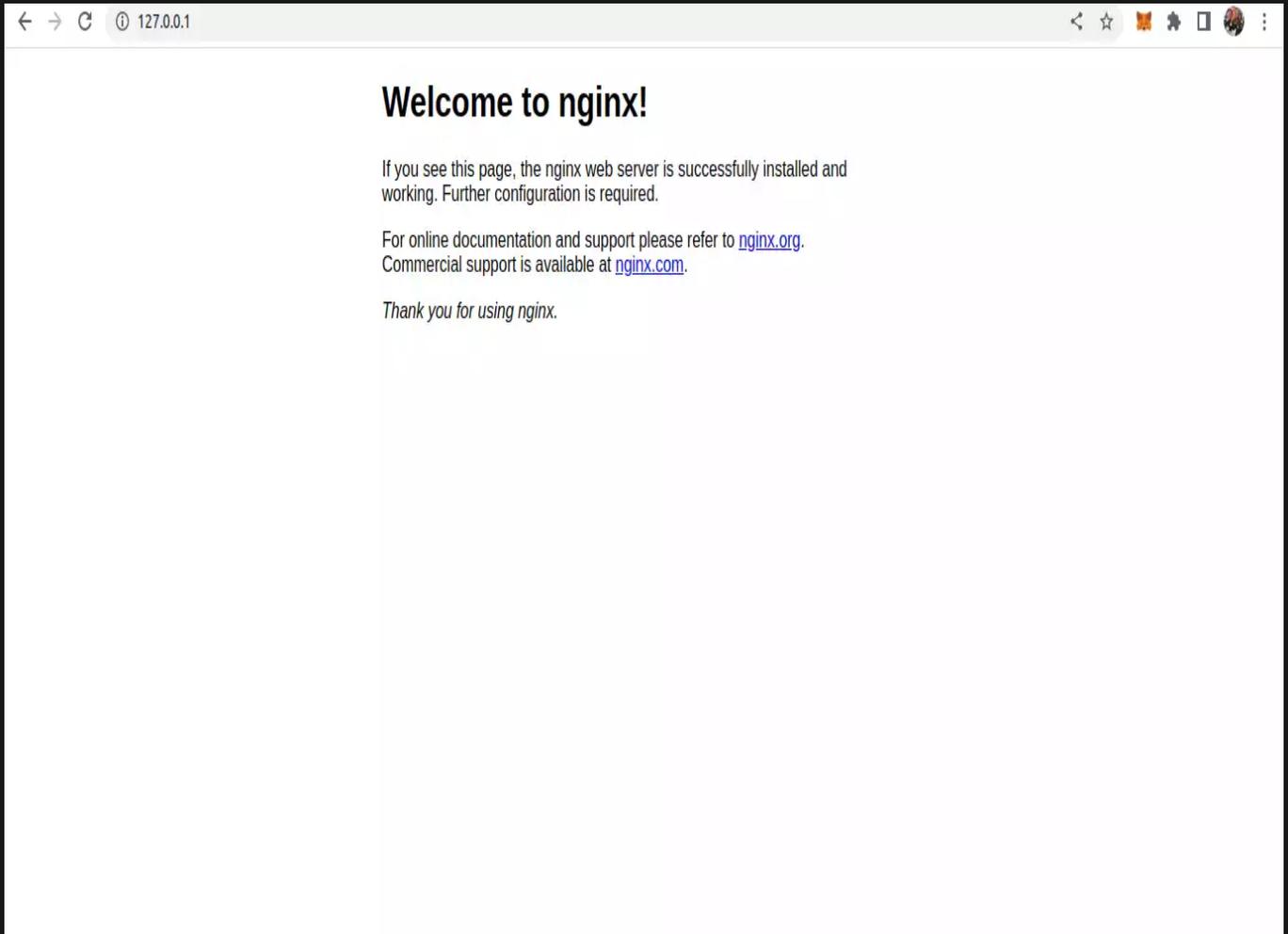
```
mohamad@mohamad-Lenovo-G510:~$ sudo ufw app list
Available applications:
CUPS
Nginx Full
Nginx HTTP
Nginx HTTPS
mohamad@mohamad-Lenovo-G510:~$
```

To get Nginx access through the default Ubuntu firewall, enter the following: `sudo ufw allow 'nginx http'` Then you need to refresh the firewall settings by entering: `sudo ufw reload` For https traffic, enter: `sudo ufw allow 'nginx https'` And for both you can use: `sudo ufw`

```
allow 'nginx full'
```

#### 4. Running a Test

To begin running a test, you should first make sure that the Nginx service is running, by checking the status as mentioned earlier. Open a web browser, and enter the following web address: `http://127.0.0.1` And you should be able to see the following result containing a page with a welcome statement.



Now, if the system does not have a graphical interface, the Nginx Welcome page can be loaded in the terminal using curl: `sudo apt-get install curl` By entering the following command, you should be able to see the Welcome page contents in the terminal: `curl 127.0.0.1` And the result is as expected:

```
mohamad@mohamad-Lenovo-G510: ~  
File Edit View Search Terminal Help  
mohamad@mohamad-Lenovo-G510:~$ curl 127.0.0.1  
<!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>
```

## 5. Configuring a Server Block

In Nginx, a server block is a configuration that works as its own server. By default, Nginx has one server block preconfigured. It is located at `/var/www/html`. However, it can be configured with multiple server blocks for different sites.

Note that in this tutorial, we will use `test_domain.com` for the domain name. This may be replaced with your own domain name.

In a terminal, create a new directory by entering the following command: `sudo mkdir -p /var/www/test_domain.com/html` Use `chmod` to configure ownership and permission rules: `sudo chown -R $USER:$USER /var/www/test_domain.com` `sudo chmod -R 755 /var/www/test_domain.com` Open `index.html` for editing in a text editor of your choice (we will use the Nano text editor): `sudo nano /var/www/test_domain.com/html/index.html` You will see the HTML code like below in it. You edit it if you like, but we will keep it this way.

```
mohamad@mohamad-Lenovo-G510: ~
File Edit View Search Terminal Help
GNU nano 4.8 /var/www/test_domain.com/html/index.html
<html>
  <head>
    <title>Welcome to test_domain.com!</title>
  </head>
  <body>
    <h1>This message confirms that your Nginx server block is working. Great >
  </body>
</html>

[ Read 8 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line
```

Press CTRL+o to write the changes, then CTRL+x to exit.

Now, open the configuration file for editing: `sudo nano /etc/nginx/sites-available/test_domain.com` Enter the following code in it: `server {`  
`listen 80;`  
`root /var/www/test_domain.com/html;`  
`index index.html index.htm index.nginx.debian.html;`  
`server_name test_domain.com www.test_domain.com;`  
`location / {`  
`try_files $uri $uri/ =404;`  
`}`  
`}` So that you have the following result:

```
mohamad@mohamad-Lenovo-G510: ~
File Edit View Search Terminal Help
GNU nano 4.8 /etc/nginx/sites-available/test_domain.com Modified
server {
    listen 80;

    root /var/www/test_domain.com/html;
    index index.html index.htm index.nginx.debian.html;

    server_name test_domain.com www.test_domain.com;
    location / {
        try_files $uri $uri/ =404;
    }
}

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line
```

Press CTRL+o to write the changes, then CTRL+x to exit.

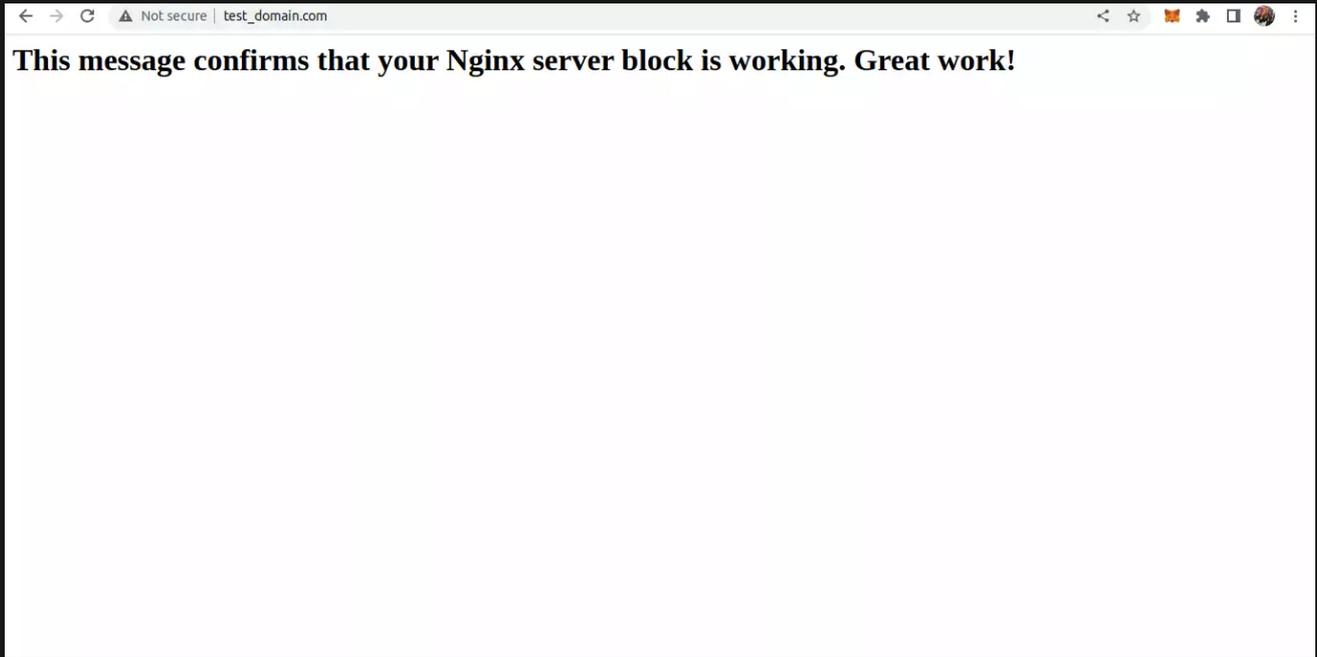
Next, create a symbolic link between the server block and the startup directory by entering the following: `sudo ln -s /etc/nginx/sites-available/test_domain.com /etc/nginx/sites-enabled` Afterward, you should restart Nginx by running the following command: `sudo systemctl restart nginx` Then, open /etc/hosts for editing: `sudo nano /etc/hosts` You will see the following result:

```
mohamad@mohamad-Lenovo-G510: ~  
File Edit View Search Terminal Help  
GNU nano 4.8 /etc/hosts  
127.0.0.1 localhost  
127.0.1.1 mohamad-Lenovo-G510  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
  
[ Read 9 lines ]  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Next, enter this command after the last line: `127.0.1.1 test_domain.com`  
`www.test_domain.com` So that it becomes like this:

```
mohamad@mohamad-Lenovo-G510: ~  
File Edit View Search Terminal Help  
GNU nano 4.8 /etc/hosts Modified  
127.0.0.1 localhost  
127.0.1.1 mohamad-Lenovo-G510  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
127.0.1.1 test_domain.com www.test_domain.com  
  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Now if you open a browser window and navigate to `test_domain.com` (or the domain name you configured in Nginx). You should see the message you entered in HTML file you opened with nano. Notice that there were already an HTML script in there and we didn't change it. But anyway, if you have changed the HTML file, you will see the result of your edition different from ours:



## Conclusion

In this tutorial, we have provided the guidelines for installing, starting, and setting Nginx up and running on Linux. Also we tested the configuration and in the end, we configured an Nginx server block. We hope you enjoyed this quick Nginx configuration tutorial and it has been helpful for you.

## Join Arashtad Community

### Follow Arashtad on Social Media

We provide variety of content, products, services, tools, tutorials, etc. Each social profile according to its features and purpose can cover only one or few parts of our updates. We can not upload our videos on SoundCloud or provide our eBooks on Youtube. So, for not missing any high quality original content that we provide on various social networks, make sure you follow us on as many social networks as you're active in. You can find out Arashtad's profiles on different social media services.



### Get Even Closer!

Did you know that only one universal Arashtad account makes you able to log into all Arashtad network at once? Creating an Arashtad account is free. Why not to try it? Also, we have regular updates on our newsletter and feed entries. Use all these beneficial free features to get more involved with the community and enjoy the many products, services, tools, tutorials, etc. that we provide frequently.

[SIGN UP](#)[NEWSLETTER](#)[RSS FEED](#)