

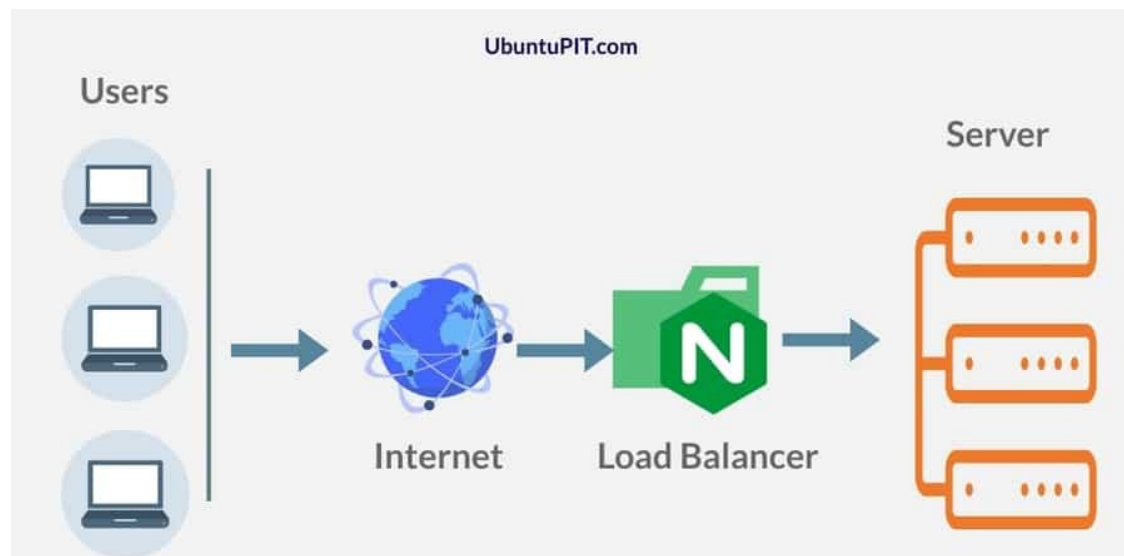
How to Install Nginx Web Server on Ubuntu Linux: A Tutorial for Beginners

By Jahid Onik

The Nginx (Pronounced as Engine-X) is a few software that can act both as a web server and a proxy server. You can serve web content through the Nginx server. Through the proxy and reverse proxy features of the Nginx server, you can also use it as a router. Traffic balancing, caching, and other server operations can be done by the Nginx server. You can install the Nginx server in any cloud server, [docker engine](#), or local Linux machine. The algorithm of the Nginx server is very beneficial for them who have a commercial server. As Ubuntu runs most of the web servers globally, you can install the Nginx server on your Ubuntu machine for better performance.

Nginx Web Server on Ubuntu Linux

A configured [Nginx web server](#) serving both static and dynamic contents will always be more efficient than a similar Apache server. You must have heard that Nginx is faster than Apache server; well, they are not lying at all. The Nginx web server can handle and serve resources faster and maintain a large number of concurrent requests. However, you can run both Apache and Nginx servers at the same machine. You need to specify a different IP address or port for each web server.



As I have mentioned earlier that the Nginx server can perform both proxy and reverse proxy configuration; let us discuss a bit on that. We all know that a proxy is a general server setting that can hide the client's identity. In proxy settings, the web-server can't find the identity of the client.

In a reverse proxy configuration, the client wouldn't find which server it is connecting to. Now, the Nginx server performs the reverse proxy settings to maintain the security checkup and incoming traffic. As the reverse proxy server can be a load balancer, the Nginx's reverse proxy server can be used to maintain a large web server.

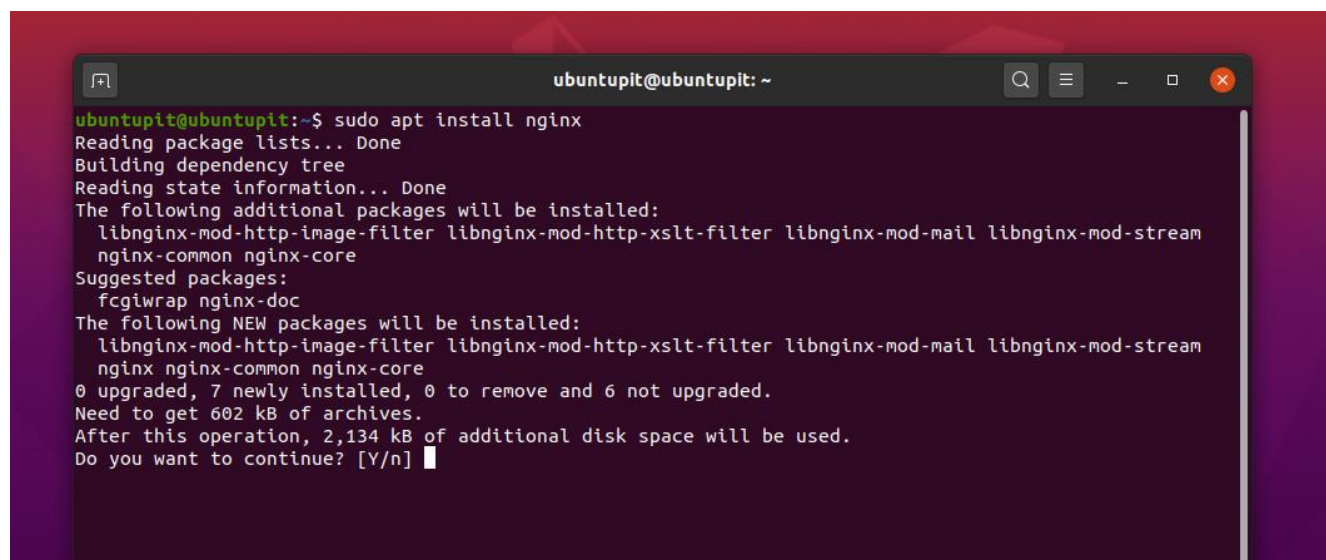
If your web server has a million users, you can use the reverse proxy feature through the Nginx server. In this post, we will see how to install the Nginx server in Ubuntu Linux. Some primary key features of the Nginx server is pointed below.

- Lightweight Server
- Requires Low Memory
- CPU Optimization
- Maximum Performance on Low Power
- Proxy and Reverse Proxy Server
- Application Gateway
- Supports All Major Protocols

Step 1: Install Nginx Web Server in Ubuntu Linux

Installing the Nginx web server is a straightforward process in Ubuntu. You need to update your software repository. Then install the Nginx web server by running the `apt` command given below. First, open your terminal shell copy and paste the following shell commands in your Ubuntu terminal.

```
sudo apt update
sudo apt install nginx
```



```

ubuntupit@ubuntupit: ~
ubuntupit@ubuntupit:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  nginx-common nginx-core
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  nginx nginx-common nginx-core
0 upgraded, 7 newly installed, 0 to remove and 6 not upgraded.
Need to get 602 kB of archives.
After this operation, 2,134 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

After installing the Nginx web server, you have to start the web server on your Ubuntu machine. To start the Nginx web server, you can use the `systemctl` command to start and check the server status.

```

sudo systemctl start nginx
sudo systemctl status nginx

```

Now, you can check the version of your Nginx web server on your Ubuntu machine.

```

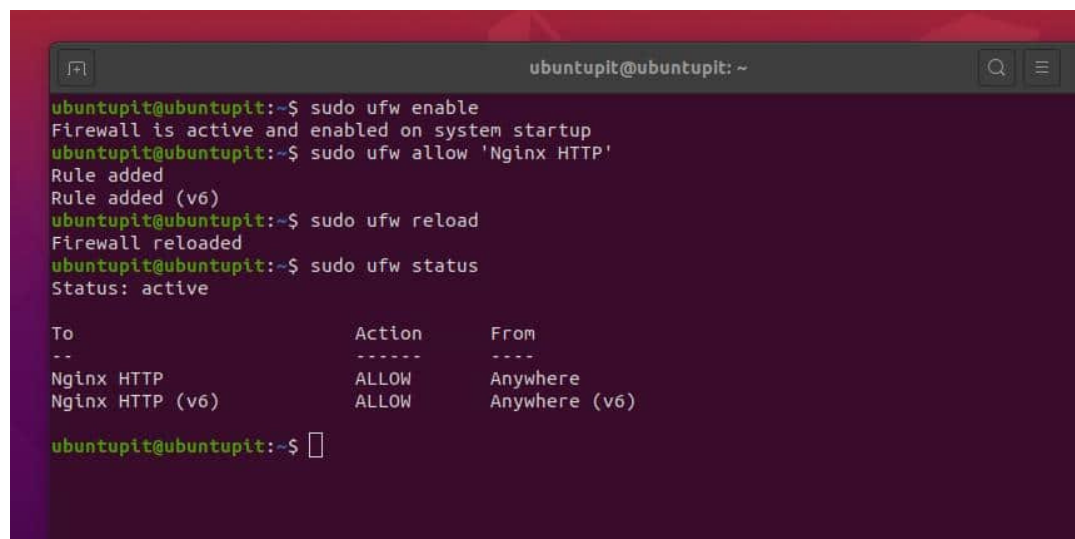
sudo dpkg -l nginx

```

Step 2: Configure Firewall Settings for the Nginx Server

Ubuntu uses the [Uncomplicated Firewall \(UFW\)](#) firewall protection to make your system secure. In this step, we have to configure the Nginx web server to the UFW firewall. We have to add a rule to allow the Nginx web server in the accepting firewall list.

First, you have to enable the UFW firewall features in your Ubuntu machine. Then you have to allow the Nginx webserver to the UFW settings. Then you can reload the check the status of your UFW firewall configuration. If you have done everything correctly, you will see that a new rule is added to the firewall configuration.



```

ubuntupit@ubuntupit: ~
ubuntupit@ubuntupit:~$ sudo ufw enable
Firewall is active and enabled on system startup
ubuntupit@ubuntupit:~$ sudo ufw allow 'Nginx HTTP'
Rule added
Rule added (v6)
ubuntupit@ubuntupit:~$ sudo ufw reload
Firewall reloaded
ubuntupit@ubuntupit:~$ sudo ufw status
Status: active

To Action From
--
Nginx HTTP ALLOW Anywhere
Nginx HTTP (v6) ALLOW Anywhere (v6)

```

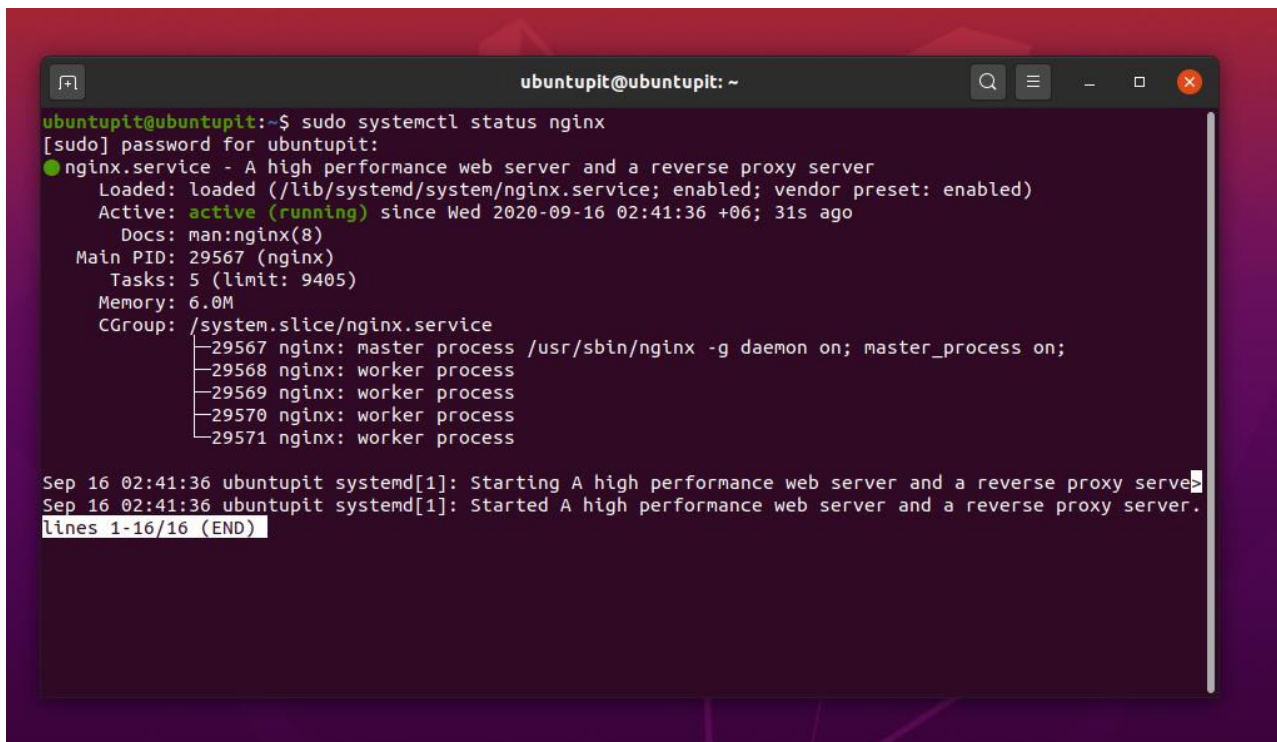
```

sudo ufw enable
sudo ufw allow 'Nginx HTTP'
sudo ufw reload
sudo ufw status

```

Step 3: Power the Nginx Web Server

If you are completely new to Nginx or have been using the Apache web server, I believe this step will help you get [fundamental commands of the Nginx web server](#). After installing the Nginx web server, you can now use the following terminal command-lines to start, stop, enable, reload, and restart the Nginx web server on your Ubuntu machine.

A terminal window titled 'ubuntuupit@ubuntuupit: ~' with search, menu, and window control icons in the title bar. The terminal shows the command 'sudo systemctl status nginx' and its output. The output indicates that the 'nginx.service' is loaded and active (running). It shows the service was started on Wed 2020-09-16 at 02:41:36. The main PID is 29567, and there are 5 tasks (limit 9405) using 6.0M of memory. The CGroup is '/system.slice/nginx.service'. A list of processes is shown: 29567 nginx: master process, and four worker processes (29568, 29569, 29570, 29571). At the bottom, two log messages from systemd are visible, and the terminal indicates 'lines 1-16/16 (END)'.

```
sudo systemctl stop nginx
sudo systemctl start nginx
sudo systemctl enable nginx
sudo systemctl restart nginx
sudo systemctl reload nginx
```

A few basic commands of the Nginx server are given below that might be helpful to maintain the Nginx web server on Ubuntu.

Check the Nginx server Status

```
nginx -t
```

Reload the Nginx Server

```
nginx -s
```

More Detailed Information About Nginx Server

```
nginx -V
```

Dump Full Nginx Server Configuration

```
nginx -T
```

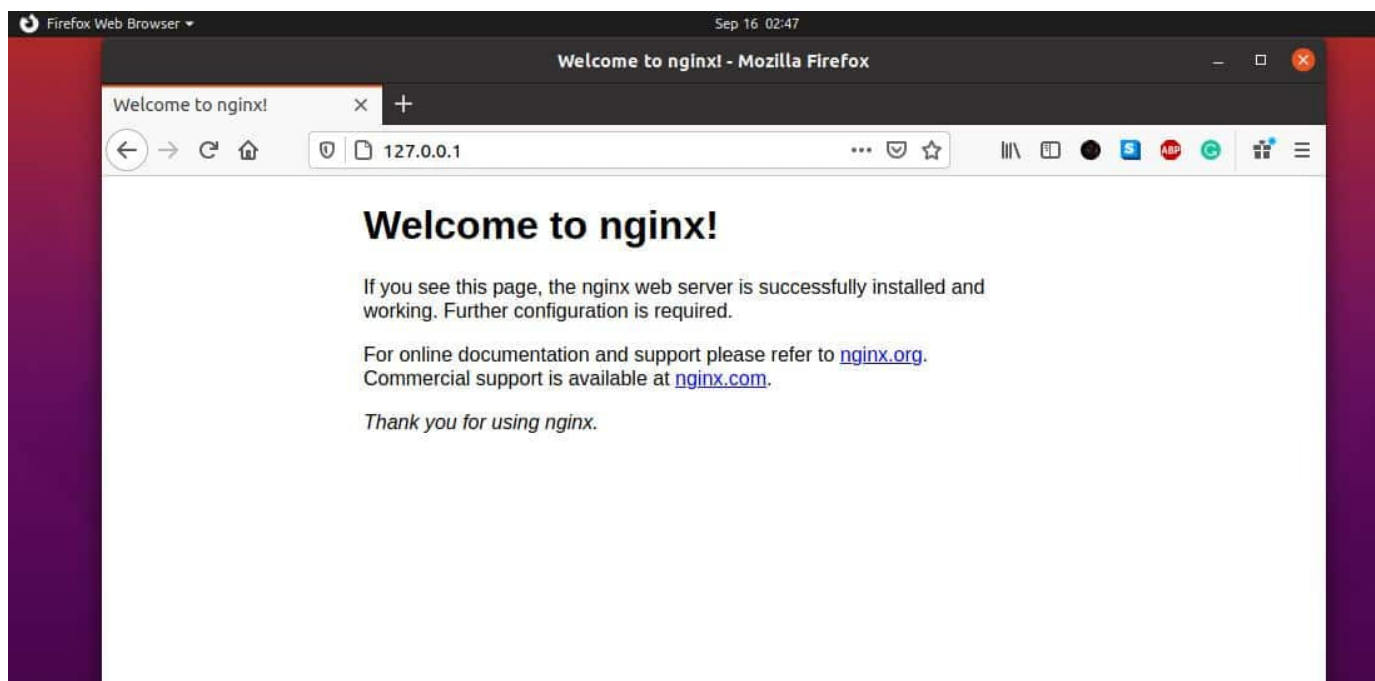
Step 4: Load the Nginx Web Server on Ubuntu Linux

Now in this step, we will find the web address that we can use to load the Nginx web server through a web browser. We can load the Nginx server on the localhost address, and the default Nginx port is 80. However, we can check and verify the web address by using the Net-tool commands. If you don't have the net tool installed inside your Ubuntu machine, install it first.

```
sudo apt install net-tools
ifconfig
```

As we can see that the localhost web address is shown in the shell, we can type the address and hit the Enter button to load the Nginx web server's welcome page on our Ubuntu machine.

```
http://127.0.0.1/
```



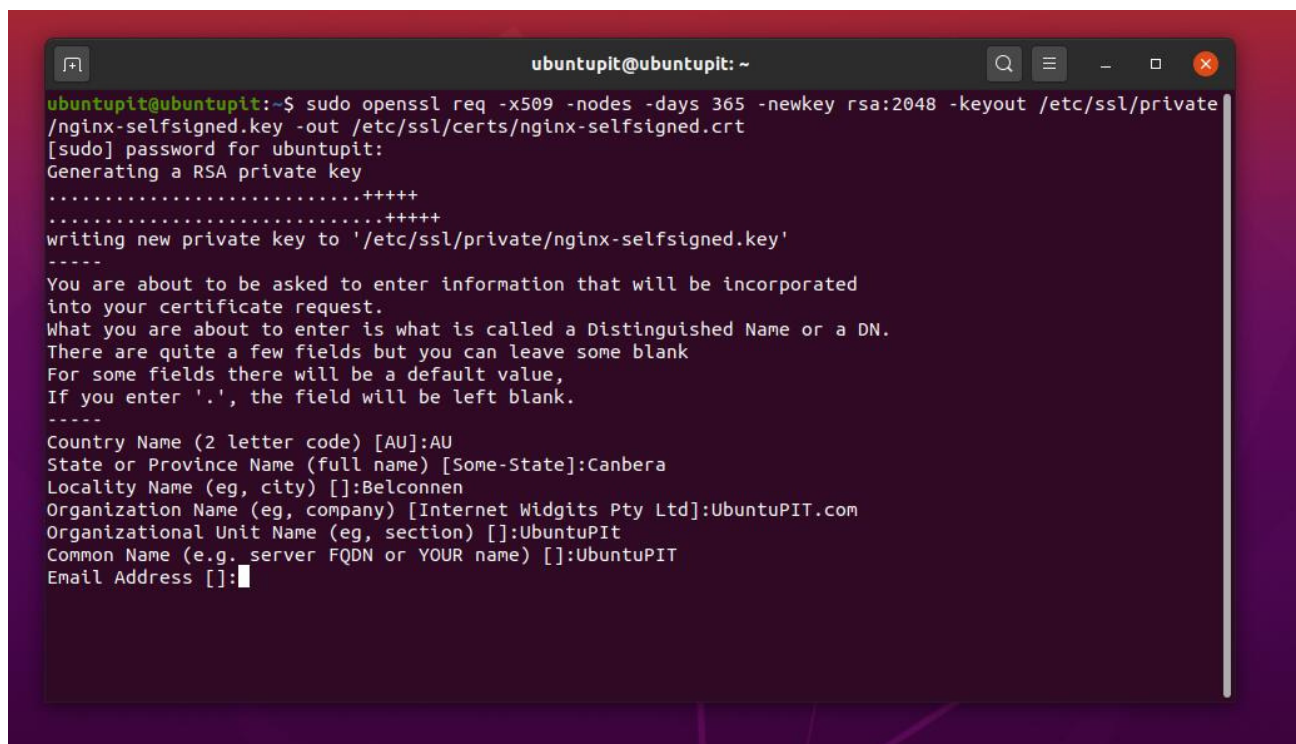
Step 5: Enable HTTPS on the Web Server

The default web address of any web server is always an HTTP address. If you want to make your web server more secure, you can generate a self-signed CSR key for your web server. You can go through this post to know more about [generating a CSR \(Certificate Signing Request\) in Linux](#). However, you can locate the CSR file inside the `/etc/ssl` directory of your Linux filesystem. First, make a directory to store your private CSR key then access root permission to the path.

```
sudo mkdir /etc/ssl/private
sudo chmod 700 /etc/ssl/private
```

Now, make a CSR request for your Nginx web server from your Ubuntu machine. Use the following command line to generate a CSR. Fill all the required information to complete the certificate signing request.

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt
```



Then you have to configure your Nginx web server for SSL certification. You can do this by editing the SSL configuration script. To make that change open the SSL configuration script by using your favorite script editor.

```
sudo gedit /etc/nginx/conf.d/ssl.conf
```

Now copy and paste the following script inside the SSL configuration script. Then save and exit the file.

```
server {
listen 443 http2 ssl;
listen [::]:443 http2 ssl;

UbuntuPIT http://127.0.0.1/;

ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
```

```

ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
ssl_dhparam /etc/ssl/certs/dhparam.pem;
}

root /usr/share/nginx/html;

location / {
}

error_page 404 /404.html;
location = /404.html {
}

error_page 500 502 503 504 /50x.html;
location = /50x.html {
}
}

```

Finally, make a little effort to redirect your Nginx webserver to HTTP to HTTPS. Open the SSL redirection configuration script add the following script into the file.

```
sudo vi /etc/nginx/default.d/ssl-redirect.conf
```

You can copy and paste the following line into your SSL redirect script.

```
return 301 https://$host$request_uri/;
```

Now to see the impact, restart your Nginx web server on Ubuntu.

```
sudo systemctl restart nginx
```

Now visit the HTTPS web address of your Nginx web server.

```
https://127.0.0.1
```

Step 6: Customize the Nginx Web Server on Ubuntu

Till now, we have seen how to install, configure, and load the Nginx web server; now, we will see how to customize the Nginx web server. We can assign a new web address and port to the Nginx configuration script.

First, we need to locate the Nginx installation path, which is `/var/www/html`. Don't forget to run all the commands as a root user. Let assume our new web address will be `www.mysamplesite101.com`. Here, we will be configuring the Nginx settings to add the new URL to the Nginx web server.

Now, we have to make a directory to host and add a new URL address. Then grant the access permission to the directory.

```

sudo mkdir -p /var/www/mysamplesite101.com/html
sudo chown -R $USER:$USER /var/www/mysamplesite101.com/html
sudo chmod -R 755 /var/www/mysamplesite101.com

```

Then open the `index.html` file of your new address to add some basic HTML lines for the webpage. You can use any script editor to edit the `index.html` script. The HTML code is given below; feel free to copy, paste, and replace the web address.

```
sudo gedit /var/www/mysamplesite101.com/html/index.html
```

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Welcome to UbuntuPIT NGINX SERVER TEST!</title>
5 <style>
6   body {
7     width: 35em;
8     margin: 0 auto;
9     font-family: Tahoma, Verdana, Arial, sans-serif;
10  }
11 </style>
12 </head>
13 <body>
14 <h1>Welcome to UbuntuPIT nginx server!</h1>
15 <p>If you see this page, the nginx web server is successfully installed and
16 working. Further configuration is required.</p>
17
18 <p>For online documentation and support please refer to
19 <a href="http://ubuntupit.com/">ubuntupit.com</a>.<br>
20 Commercial support is available at
21 <a href="http://ubuntupit.com/">ubuntupit.com</a>.</p>
22
23 <p><em>Thank you for using nginx.</em></p>
24 </body>
25 </html>

```

```

<!DOCTYPE html>
<html>
<head>
<title>Welcome to UbuntuPIT NGINX SERVER TEST!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to UbuntuPIT nginx server!</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
ubuntupit.com.

Commercial support is available at
ubuntupit.com.</p>
<p>Thank you for using nginx.</p>

</body>
</html>

You must know that the Nginx web server has a configuration script where all the allowed web address names and links are stored. You have to enlist your URL address and grant permission to the address to let the Nginx web server load your URL.

```
sudo gedit /etc/nginx/sites-available/mysamplesite101.com
sudo ln -s /etc/nginx/sites-available/mysamplesite101.com /etc/nginx/sites-enabled/
sudo ln -s /etc/nginx/sites-available/mysamplesite101.com /etc/nginx/sites-enabled/
```

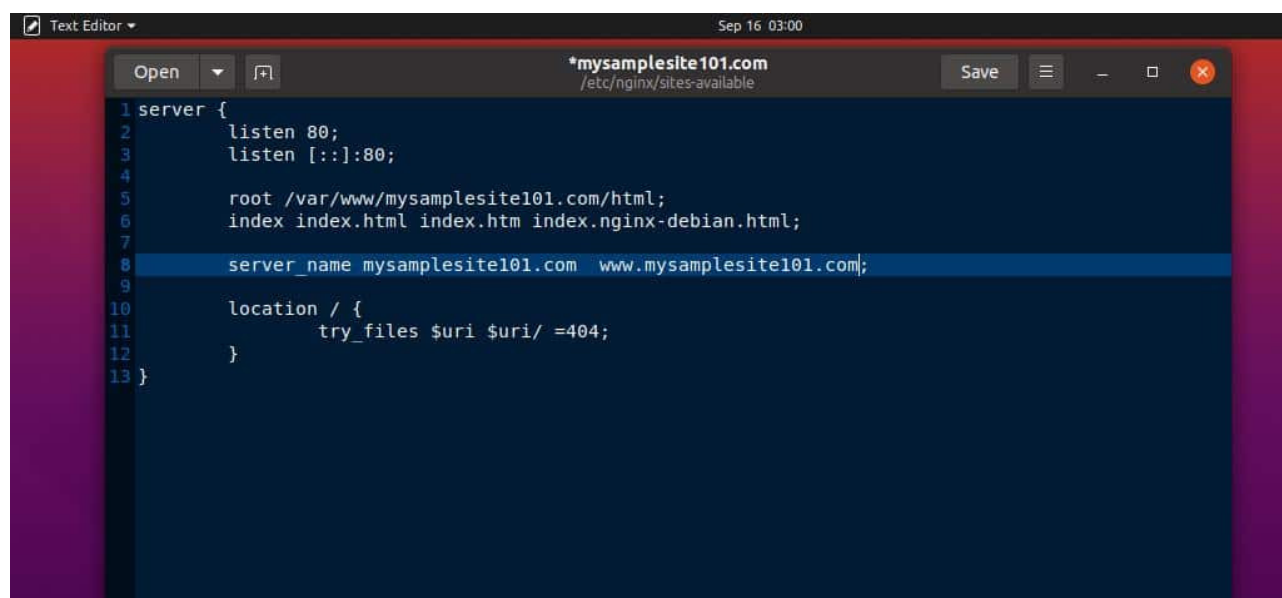
First, open the index script to see the available site list of Nginx on your Ubuntu machine. Then add your site name and domain to the script. You can use the following command lines to do the job.

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

mysamplesite101.com mysamplesite101.com;

root /var/www/mysamplesite101.com;
index index.html;

location / {
try_files $uri $uri/ =404;
}
}
```

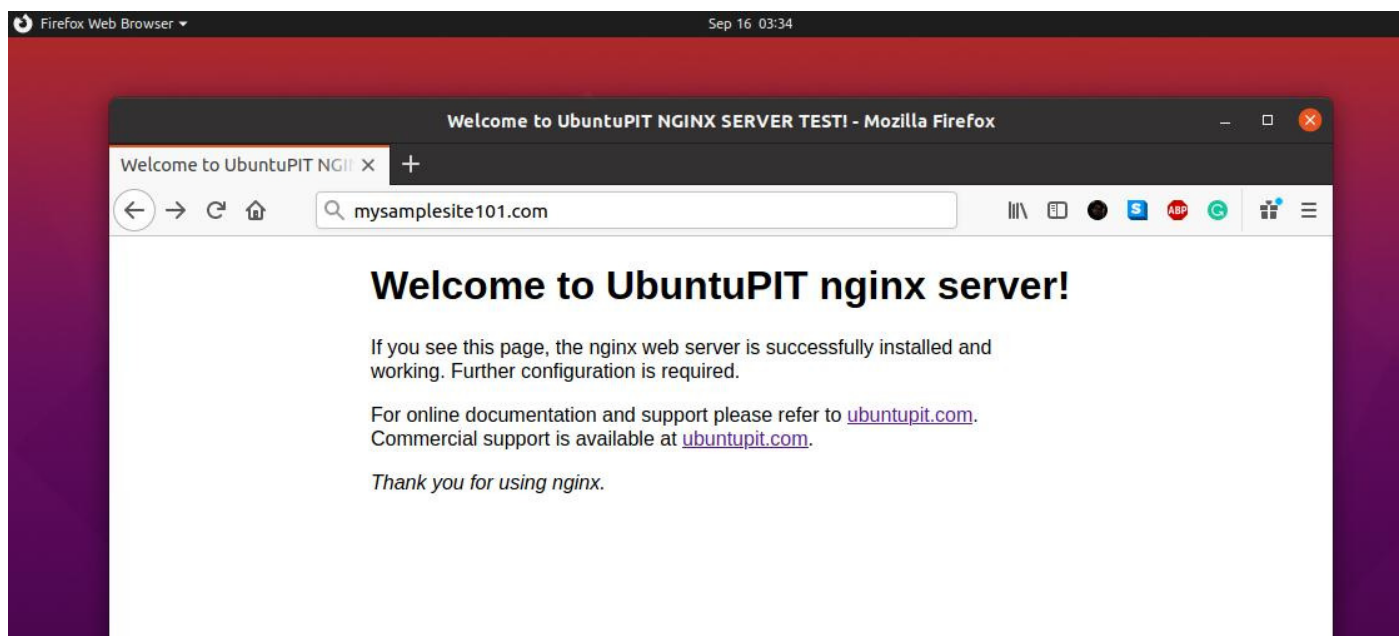


Now restart the Nginx web server on your Ubuntu machine.

```
sudo systemctl restart nginx
nginx -t
```

Here is some crucial directory address of the Nginx web server that might help you locate and edit the configuration script, available site list, error log, and access log on Ubuntu.

Now type your customized URL in the address bar of your web browser. If you can load the following page, your configuration is functioning perfectly.



```
/etc/nginx/nginx.conf
/etc/nginx/sites-available
/etc/nginx/sites-enabled
/var/log/nginx/access.log
/var/log/nginx/error.log
```

Remove the Nginx Web Server from Ubuntu Linux

Several specific reasons might cause you to take your Nginx web server down from your Ubuntu machine; however, if you want to uninstall and remove the Nginx web server and its configuration files from your machine. First, you need to stop the running progress of the Nginx server on your machine. Follow the terminal commands to stop all Nginx services on your Ubuntu machine.

```
service nginx quit
systemctl quit nginx
```

Now, use the following apt commands to uninstall the Nginx web server from your machine.

```
sudo apt-get remove nginx-full nginx-common
sudo apt-get purge nginx nginx-common
```

After the uninstallation process is done, you can use the following terminal command lines to remove the Nginx configuration directories from your Ubuntu machine.

```
sudo apt-get autoremove
rm -rf /etc/nginx
```

As you are dealing with a server configuration, you might be interested in something that can increase the loading speed of your site. While the traditional HTTP protocol loads files individually from the server, the [HTTP2 protocol](#) can combine and load all the resources from your server as a combined binary file. By using the HTTP2 method, you can make your server's speed better, decrease the loading time, and fix the timeout error.

Ending Words

Before installing the Nginx server in your Ubuntu machine, make sure that you actually need the Nginx server for your website or local business. You may also check the Nginx plus features. If you are an Nginx web server enthusiast, you can go through this post to know [essential things about the Nginx web server](#).

I have demonstrated the fundamental thoughts of a proxy and a reverse proxy server in the entire post. I have also described the steps of how to install and configure the Nginx web server on Ubuntu. You can also install the Nginx server on your Ubuntu server.

If you love this post, please share it with your friends and the Linux community. We also encourage you to write down your opinions in the comment section.