
# Nginx

Eleonora Biolo

Haaga-Helia University of Applied Sciences
Exchange program, Double Degree
Project
2023/2024

## Abstract

**Author(s)**
Eleonora Biolo

**Degree**
Bachelor of Business Information Technology, Double Degree, Exchange

**Report/Thesis Title**
Nginx

**Number of pages and appendix pages**
11

How to install nginx and how to make a customised web server.

## Table of contents

## Introduction

As part of the Linux server ICI003AS2AE-3001 course, we will be asked to develop a small project with the aim of demonstrating the skills learned during the course and applying them independently.

I decided to install the Nginx web server. Nginx is a free web server software that handles http and https protocol requests quickly and efficiently. As it is open source, it is free. Nginx not only acts as a web server, but is also used as a proxy, i.e. to process client requests and pass them on to other servers, thereby balancing the load (number of requests) between several servers. It is capable of managing large-scale simultaneous connections, enabling large-scale deployments.

This web server is supported by all the major operating systems, including Linux, Windows, MacOs and others. It is most commonly used with Linux because it is stable, flexible and has a large support community. What's more, this web server uses very little memory and its configuration is very flexible. Nginx offers a number of other features, such as use with other web applications, but I'll be concentrating on creating a web server.

## Commands and results

I'm going to install the Nginx web server on my virtual machine, Oracle VM Virtualbox, which hosts my Linux Server version 22.04.3 LTS.

The first step is to launch the server and update it with the *sudo apt-get update* and *sudo apt-get upgrade* commands.

```
ele@eleserver:~$ sudo apt-get update _
```
Figure 1 - Update command

```
ele@eleserver:~$ sudo apt-get upgrade _
```
Figure 2 - Upgrade command

Once these two commands have been completed, we can go on to install Nginx with the following command *sudo apt install nginx*

```
ele@eleserver:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjbig0 libjpeg-turbo8
  libjpeg8 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4
  nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjbig0 libjpeg-turbo8
  libjpeg8 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 nginx
  nginx-common nginx-core
0 upgraded, 20 newly installed, 0 to remove and 4 not upgraded.
Need to get 2,692 kB of archives.
After this operation, 8,345 kB of additional disk space will be used.
Do you want to continue? [Y/n] _
```
Figure 3 - Nginx installation

To check whether the program has been installed, you can use the following command to find out the status of the program: *sudo service nginx status*. As you can see in the image below, nginx is active. You can also check which version of nginx you have installed with the command *nginx -v*.

```
ele@eleserver:~$ sudo service nginx status
● 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 Sun 2023-11-26 16:52:42 UTC; 2min 24s ago
       Docs: man:nginx(8)
    Process: 27436 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited>
    Process: 27437 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=>
   Main PID: 27532 (nginx)
      Tasks: 4 (limit: 4557)
     Memory: 5.5M
        CPU: 92ms
     CGroup: /system.slice/nginx.service
             ├─27532 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─27535 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" >
             ├─27536 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" >
             └─27537 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" >

Nov 26 16:52:42 eleserver systemd[1]: Starting A high performance web server and a reverse proxy se>
Nov 26 16:52:42 eleserver systemd[1]: Started A high performance web server and a reverse proxy ser>
lines 1-18/18 (END)_
```
Figure 4 - Check the status of nginx

```
ele@eleserver:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
```
Figure 5 - Check which version of nginx we have installed

Since we're in a server, so we don't have a graphical interface, we need to install a text-based web browser. In this case, we're going to install lynx. To do this, issue the command *sudo apt-get install lynx*.

```
ele@eleserver:~$ sudo apt-get install lynx
```

Figure 6 - Lynx installation

Now we need to know our ip address so that we can launch the nginx web server. Type the command *ip a*

```
ele@eleserver:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 08:00:27:c5:1b:5d brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 86084sec preferred_lft 86084sec
    inet6 fe80::a00:27ff:fec5:1b5d/64 scope link
       valid_lft forever preferred_lft forever
```

Figure 7 - Ip address

The ip address is 10.0.2.15, so you can launch the web server using the command *lynx http://10.0.2.15*

```
ele@eleserver:~$ lynx http://10.0.2.15_
```

Figure 8 - Launch web browser for access to our web server

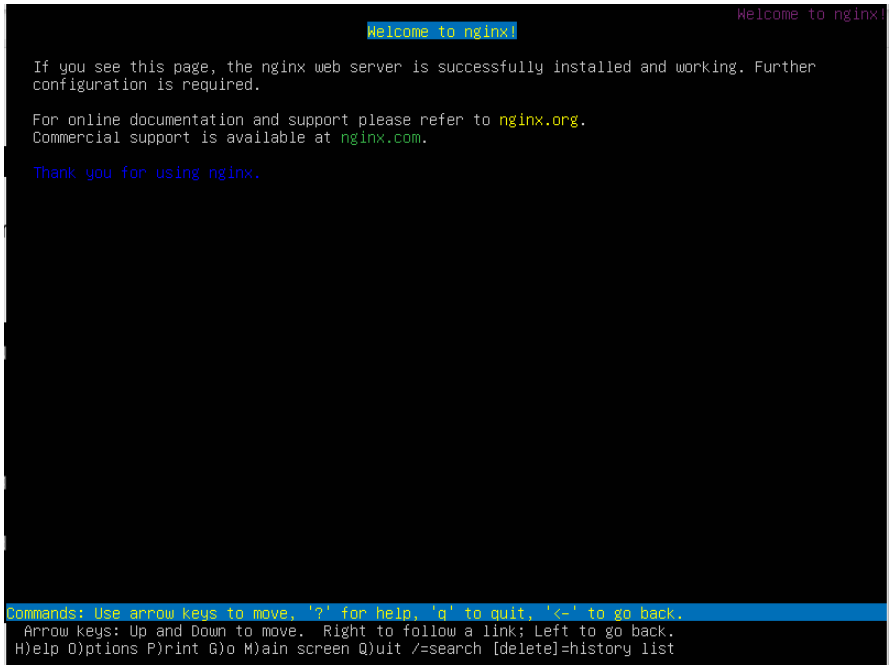


Figure 9 - Web server

In the /var/www folder we can create a folder called "mysite" in which we will create an html file, index.html. With the following code:

```
<html>
<head>Eleonora's Web Site</head>
<body>
  <h1>Welcome to Eleonora's
homepage</h1>
</body>
</html>
```

This file will allow us to display the nginx server with a custom page containing the text above.

In order to go to the /var/www folder, we're going to do the command *cd /var/www* and then create the mysite folder with *sudo mkdir mysite*. Enter this folder (*cd mysite*) and create the file *with sudo nano index.html*.

```
ele@eleserver:~$ cd /var/www
ele@eleserver:/var/www$ pwd
/var/www
```

Figure 10 - Go to the www folder and check where we are

```
ele@eleserver:/var/www$ sudo mkdir mysite
```

Figure 11 - Creation of "mysite" folder

```
ele@eleserver:/var/www$ ls
html  mysite
```

Figure 12 - Check that the folder has been created

```
ele@eleserver:/var/www$ cd mysite/
```

Figure 13 - Go to mysite folder

```
ele@eleserver:/var/www/mysite$ sudo nano index.html
```

Figure 14 - Create and edit the "index.html" file

```
ele@eleserver:/var/www/mysite$ cat index.html
<html>
<head>Eleonora's Web Site</head>
<body>
  <h1>Welcome to Eleonora's homepage</h1>
</body>
</html>
```

Figure 15 – Check the content of the file

Once we have created the web server's home page we need to create a new configuration in the /etc/nginx/sites-availables folder. To do this we move to this folder, *cd /etc/nginx/sites-availables* and create a file, *sudo nano eleonora.fi.* In this file we will write the following configuration

```
server {
    listen 80;
    server_name eleonora.fi www.eleonora.fi;

    location / {
        root /var/www/mysite;
        index index.html;
        try_files $uri $uri/ $uri.html =404;
    }
}
```

```
ele@eleserver:/etc/nginx/sites-available$ cd
ele@eleserver:~$ cd /etc/nginx/sites-available/
ele@eleserver:/etc/nginx/sites-available$ sudo nano eleonora.fi
```

Figure 16 - Move to the sites-available folder and create the configuration file

```
ele@eleserver:/etc/nginx/sites-available$ cat eleonora.fi
server {
        listen 80;
        server_name eleonora.fi www.eleonora.fi;

        location / {
                root /var/www/mysite;
                index index.html;
                try_files $uri $uri/ $uri.html =404;
        }
}
ele@eleserver:/etc/nginx/sites-available$
```

Figure 17 - Contents of the eleonora.fi configuration file

This configuration allows the server to listen for requests on port 80 of the eleonora.fi domain or www.eleonora.fi, it will look for the file in the /var/www/mysite folder and if this file is not found it will return a 404 error.

Once you have created this file, you need to create a link in the /etc/nginx/sites-enable folder pointing to this file. To do this, move to the sites-enable folder and issue the command *sudo ln -s ../sites-available/eleonora.fi eleonora.fi*. If this command doesn't work, you need to check the permissions and probably change them with *sudo chmod 664 eleonora.fi*, directly in the sites-available folder. You should also delete the default file in this same folder with the command *sudo rm default*

```
ele@eleserver:/etc/nginx/sites-enabled$ sudo rm default
```

Figure 18 - Remove the default file

```
ele@eleserver:/etc/nginx/sites-enabled$ sudo ln -s ../sites-available/eleonora.fi eleonora.fi
```

Figure 19 - Create a link in the sites-enable folder

```
ele@eleserver:/etc/nginx/sites-enabled$ ls -la
total 8
drwxr-xr-x 2 root root 4096 Nov 26 19:09 .
drwxr-xr-x 8 root root 4096 Nov 26 16:52 ..
lrwxrwxrwx 1 root root   30 Nov 26 19:09 eleonora.fi -> ../sites-available/eleonora.fi
```

Figure 20 - Check that the link has been created

Once these commands have been carried out, we need to add our ip address to the /etc/hosts file. This will allow us to reach our web server and not an external server. To do this, type the following command *sudo nano /etc/hosts* and add the following two lines :

```
ele@eleserver:~$ sudo nano /etc/hosts _
```

Figure 21 - Open the hosts file

```
  GNU nano 6.2                                /etc/hosts *
127.0.0.1 localhost
127.0.1.1 eleserver
# My web server_
10.0.2.15 eleonora.fi
10.0.2.15 www.eleonora.fi

# 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
```

Figure 22 - Add the two lines under # My web server

Before starting the web server, you need to restart it with sudo service nginx restart.

```
ele@eleserver:/etc/nginx/sites-enabled$ sudo service nginx restart
```

Figure 23 - Restart of nginx

You can use the *lynx eleonora.fi* and *lynx www.eleonora.fi* commands to access our web server.

```
ele@eleserver:~$ lynx www.eleonora.fi_
```

Figure 24 - Accessing the web server with the domain name www.eleonora.fi

```
ele@eleserver:~$ lynx eleonora.fi
```

Figure 25 - Accessing the web server with the domain name eleonora.fi


Figure 26 - The web server home page

## Issues

When I started the project I couldn't get an internet connection, so I had to modify my settings to be able to continue. I first tried using commands to activate the network, but it was a problem that had to be solved in the settings.

When I created the personalised home page I didn't succeed straight away. When creating the link between the eleonora.fi file in the sites-available and sites-enable folders, I had an error that prevented me from moving forward. So, I changed the permissions of the file in the sites-available folder and was then able to make the link. In addition, when I restarted my web server I had an error, the configuration folder was not reachable. To find out where the problem was, I looked in the /var/log/nginx/error.log file. I modified the eleonora.fi file that was causing the problem and then it worked. This error took up a lot of my time because I didn't know what modification to make.

Finally, I couldn't reach the web server via the domain name, I could only access it via my ip address. I then searched for and found the changes I needed to make to the hosts file in order to be able to use domain names.

Fun fact: I added an image to the web server's home page, but then I remembered that there's no graphical interface.

## Sources

DeepL Translate: The World's Most Accurate Translator. https://www.deepl.com/translator, 27th november 2023

DeepL Write: AI-Powered Writing Companion https://www.deepl.com/en/write, 27th november 2023

ChatGPT, openai, openai.com, 26th november 2023

« How to serve images with nginx ». Stack Overflow, https://stackoverflow.com/q/18954827, 26th november 2023

« Linux network troubleshooting - Commands and examples ». CodiLime, https://codilime.com/blog/linux-network-troubleshooting/, 26th november 2023

« Modify Your Hosts File ». Rackspace Technology Documentation, https://docs.rackspace.com/docs/modify-your-hosts-file, 26th november 2023

Nginx — Formation Mettre en place un serveur Web. grafikart.fr, https://grafikart.fr/tutoriels/nginx-692, 26th november 2023