Step 1 – Installing Nginx

Because Nginx is available in Ubuntu’s default repositories, it is possible to install it from these repositories using the apt packaging system.

Since this is our first interaction with the apt packaging system in this session, we will update our local package index so that we have access to the most recent package listings. Afterwards, we can install nginx:

* sudo apt update
* sudo apt install ngnix
* nginx -v

After accepting the procedure, apt will install Nginx and any required dependencies to your server.

Step 2 – Adjusting the Firewall

Before testing Nginx, the firewall software needs to be adjusted to allow access to the service. Nginx registers itself as a service with ufw upon installation, making it straightforward to allow Nginx access.

List the application configurations that ufw knows how to work with by typing:

* sudo ufw app list

You should get a listing of the application profiles:

Output

Available applications:

Nginx Full

Nginx HTTP

Nginx HTTPS

OpenSSH

As you can see, there are three profiles available for Nginx:

* Nginx Full: This profile opens both port 80 (normal, unencrypted web traffic) and port 443 (TLS/SSL encrypted traffic)
* Nginx HTTP: This profile opens only port 80 (normal, unencrypted web traffic)
* Nginx HTTPS: This profile opens only port 443 (TLS/SSL encrypted traffic)

It is recommended that you enable the most restrictive profile that will still allow the traffic you’ve configured. Since we haven’t configured SSL for our server yet in this guide, we will only need to allow traffic on port 80.

Note: ufw mean uncomplicated firewall.

You can enable this by typing:

* sudo ufw allow 'Nginx HTTP'

You can verify the change by typing:

* sudo ufw status

You should see HTTP traffic allowed in the displayed output:

Output

Status: active

To Action From

-- ------ ----

OpenSSH ALLOW Anywhere

Nginx HTTP ALLOW Anywhere

OpenSSH (v6) ALLOW Anywhere (v6)

Nginx HTTP (v6) ALLOW Anywhere (v6)

Step 3 – Checking your Web Server

At the end of the installation process, Ubuntu 18.04 starts Nginx. The web server should already be up and running.

We can check with the systemd init system to make sure the service is running by typing:

* systemctl status nginx

Output

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 Fri 2018-04-20 16:08:19 UTC; 3 days ago

Docs: man:nginx(8)

Main PID: 2369 (nginx)

Tasks: 2 (limit: 1153)

CGroup: /system.slice/nginx.service

├─2369 nginx: master process /usr/sbin/nginx -g daemon on; master\_process on;

└─2380 nginx: worker process

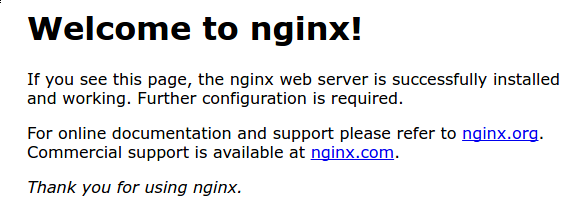
As you can see above, the service appears to have started successfully. However, the best way to test this is to actually request a page from Nginx.

curl -4 icanhazip.com >> by this you would be getting ip address.

When you have your server’s IP address, enter it into your browser’s address bar:

http://your\_server\_ip

You should see the default Nginx landing page:



This page is included with Nginx to show you that the server is running correctly.

Step 4 – Managing the Nginx Process

Now that you have your web server up and running, let’s review some basic management commands.

To stop your web server, type:

* sudo systemctl stop nginx

To start the web server when it is stopped, type:

* sudo systemctl start nginx

To stop and then start the service again, type:

* sudo systemctl restart nginx

If you are simply making configuration changes, Nginx can often reload without dropping connections. To do this, type:

* sudo systemctl reload nginx

By default, Nginx is configured to start automatically when the server boots. If this is not what you want, you can disable this behavior by typing:

* sudo systemctl disable nginx

To re-enable the service to start up at boot, you can type:

* sudo systemctl enable nginx