

Installing and setting up Php and Nginx on Ubuntu 20.04

In this guide we will cover steps to install PHP 7.4 in Ubuntu 20.04 and set up php-fpm to serve that traffic with the help of Nginx.

PHP is a general-purpose scripting language. It is mostly popular for web development. It can be used to develop dynamic and interactive websites. It was among the first server-side languages that could be embedded into HTML, making it easier to add functionality to web pages without needing to call external files for data. PHP is among the easier programming languages to learn.

It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server. Many popular CMS and frameworks such as WordPress, Magento, and Laravel are written in PHP.

At the time of writing this article, the default Ubuntu 20.04 repositories include PHP 7.4 version.

Prerequisites

To follow along this guide, ensure that you have the following:

1. Updated Ubuntu 20.04 server that is connected to the internet
2. Root access to the server or user with sudo access
3. You should be comfortable with using Linux terminal to execute commands.

Steps to install and and configure PHP in Ubuntu 20.04

1. Ensure that the server is up to date
2. Install php 7.4 on the server
3. Installing Nginx
4. Configure Nginx to serve php file

1. Ensure that the server is up to date

Before proceeding, let us ensure that our server is updated. Use these commands to 1. Update the server repositories, 2. Upgrade server packages to the latest releases:

```
# Update repos
sudo apt update
```

```
# Upgrade packages
sudo apt -y upgrade
```

2. Install php 7.4 on the server

Since the default repositories contain php version 7.4, let's install it with this command:

```
sudo apt install php -y
```

Confirm the installed version with the command `php -v`:

```
~$ php -v
PHP 7.4.3 (cli) (built: Aug 13 2021 05:39:12) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
```

In this set up, we intend to serve php files with Nginx. Nginx doesn't have built-in support for processing PHP files. We'll use PHP-FPM ("fastCGI process manager") to handle the PHP files.

Nginx will not work if Apache is installed and running. Since we will be using Nginx, let's use this command to remove apache if it is installed in our system:

```
$ sudo apt remove apache2 -y
```

Let's now install php-fpm with this command:

```
sudo apt install php-fpm -y
```

Once the installation is completed, the FPM service will start automatically. The service is added to systemd as php7.4-fpm. To check the status of the service, run:

```
$ sudo systemctl status php7.4-fpm
• php7.4-fpm.service - The PHP 7.4 FastCGI Process Manager
Loaded: loaded (/lib/systemd/system/php7.4-fpm.service; enabled;
vendor preset: enabled)
Active: active (running) since Tue 2021-09-28 16:18:41 UTC; 3min 18s
ago
Docs: man:php-fpm7.4(8)
Main PID: 56092 (php-fpm7.4)
```

```
Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic:
0req/sec"
Tasks: 3 (limit: 4710)
Memory: 7.1M
CGroup: /system.slice/php7.4-fpm.service
└─56092 php-fpm: master process (/etc/php/7.4/fpm/php-fpm.conf)
└─56113 php-fpm: pool www
└─56114 php-fpm: pool www
```

```
Sep 28 16:18:41 ip-172-26-11-229 systemd[1]: Starting The PHP 7.4
FastCGI Process Manager...
Sep 28 16:18:41 ip-172-26-11-229 systemd[1]: Started The PHP 7.4
FastCGI Process Manager.
```

You should see output like above.

3. Installing Nginx

Now that PHP and php-fpm are installed in the system, we need to install nginx. Nginx will allow us to set up a virtual host that we can use to serve traffic from a domain.

Install nginx using this command:

```
sudo apt install nginx -y
```

Once the installation is done, Nginx will be started by default. Confirm that it is working as expected by checking its status. You should see output like this:

```
$ sudo 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 Tue 2021-09-28 16:28:30 UTC; 1min 2s
ago
Docs: man:nginx(8)
Main PID: 56990 (nginx)
Tasks: 3 (limit: 4710)
Memory: 3.7M
CGroup: /system.slice/nginx.service
└─56990 nginx: master process /usr/sbin/nginx -g daemon on;
master_process on;
└─56991 nginx: worker process
```

└─56992 nginx: worker process

Sep 28 16:28:30 ip-172-26-11-229 systemd[1]: Starting A high performance web server and a reverse pr>

Sep 28 16:28:30 ip-172-26-11-229 systemd[1]: Started A high performance web server and a reverse pro>

To confirm that Nginx is actually running and is reachable, use this command:

```
curl http://server_ip/
```

Where server_ip is the ip of your server. If by any reason you do not know the public ip of your server use this to check:

```
curl -4 icanhazip.com
```

You should see the Nginx default page.

4. Configure Nginx to serve php file

Now that we have Nginx set up and running in our server, let's make it serve php files. In this example we will configure Nginx to serve the php info page.

To achieve this, we need to set up a virtual host in Nginx serving traffic for a specific domain (we will use example.citizix.com). For this to work you need to update the DNS records for that domain in your dns host to have A records point to that domain or subdomain. Consult your DNS providers documentation for the specifics of how to do that.

Confirm that the domain can resolve correctly to the server using the dig command:

```
> dig -t A example.citizix.com
```

```
; <<>> DiG 9.10.6 <<>> -t A example.citizix.com
```

```
;; global options: +cmd
```

```
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10212
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
```

```
;; OPT PSEUDOSECTION:
```

```
; EDNS: version: 0, flags:; udp: 512
```

```
;; QUESTION SECTION:
```

```
;example.citizix.com. IN A
```

```
;; ANSWER SECTION:
```

```
example.citizix.com. 300 IN A 172.67.197.11
```

```
;; Query time: 828 msec
```

```
;; SERVER: 192.168.157.227#53(192.168.157.227)
```

```
;; WHEN: Tue Sep 28 20:42:05 EAT 2021
```

```
;; MSG SIZE rcvd: 80
```

Now we need to follow these steps to have our simple script working:

1. Create the directory to store our static content (we will use /var/www/example.citizix.com)
2. Ensure the created directory is accessible by the Nginx user
3. Add an index file /var/www/example.citizix.com/index.php with content to show php info to the directory (/var/www/example.citizix.com)

Edit the file:

Add this content:

Save and close the file.

4. Create nginx configuration in the nginx default directory /etc/nginx/conf.d. 1. We will create the file /etc/nginx/conf.d/example.citizix.com.conf with the required content.

Add this content:

```
server {
listen 80;
server_name example.citizix.com;

root /var/www/example.citizix.com;
index index.php;

location ~ /\.php$ {
fastcgi_intercept_errors on;
include fastcgi_params;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
include snippets/fastcgi-php.conf;
fastcgi_pass unix:/run/php/php7.4-fpm.sock;
```

```
}  
}
```

Confirm that nginx config is correct with this command:

```
$ sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Now you need to restart nginx and php-fpm using these commands:


```
sudo systemctl restart nginx  
sudo systemctl restart php7.4-fpm
```

Confirm that the services came up:

```
sudo systemctl status nginx  
sudo systemctl status php7.4-fpm
```

You should see that the services restarted successfully!

If all goes well, you should see the php info page as shown in the screenshot below:



PHP Version 7.4.3	
System	Linux ip-172-26-11-229 5.4.0-1018-aws #18-Ubuntu SMP Wed Jun 24 01:15:00 UTC 2020 x86_64
Build Date	Aug 13 2021 05:39:12
Server API	FPFI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/fpm
Loaded Configuration File	/etc/php/7.4/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/fpm/conf.d
Additional .ini files parsed	/etc/php/7.4/fpm/conf.d/10-opcache.ini, /etc/php/7.4/fpm/conf.d/10-pdo.ini, /etc/php/7.4/fpm/conf.d/20-calendar.ini, /etc/php/7.4/fpm/conf.d/20-ctype.ini, /etc/php/7.4/fpm/conf.d/20-exif.ini, /etc/php/7.4/fpm/conf.d/20-fileinfo, /etc/php/7.4/fpm/conf.d/20-ftp.ini, /etc/php/7.4/fpm/conf.d/20-gettext.ini, /etc/php/7.4/fpm/conf.d/20-iconv.ini, /etc/php/7.4/fpm/conf.d/20-json.ini, /etc/php/7.4/fpm/conf.d/20-phar.ini, /etc/php/7.4/fpm/conf.d/20-posix.ini, /etc/php/7.4/fpm/conf.d/20-readline.ini, /etc/php/7.4/fpm/conf.d/20-shmop.ini, /etc/php/7.4/fpm/conf.d/20-sockets.ini, /etc/php/7.4/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-sysvshm.ini, /etc/php/7.4/fpm/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS
PHP Extension Build	API20190902.NTS
Database Driver	

Citizix – PHP 7.4 Info page

That's all! Up to this point we have been able to install php 7.4 in Ubuntu 20.04 and serve simple file with Nginx.