How to Install LEMP Stack on Ubuntu 20.04 Server/ Desktop

edit from: https://www.linuxbabe.com/ubuntu/install-lemp-stack-ubuntu-20-04-server-desktop

This tutorial is going to show you how to install LEMP stack (Nginx, MariaDB, and PHP7.4) on Ubuntu 20.04. A software stack is a set of software tools bundled together. LEMP stands for Linux, Nginx (Engine-X), MariaDB/MySQL and PHP, all of which are open source and free to use. It is the most common software stack that powers dynamic websites and web applications. Linux is the operating system; Nginx is the web server; MariaDB/MySQL is the database server and PHP is the server-side scripting language responsible for generating dynamic web pages.

Prerequisites

To follow this tutorial, you need an Ubuntu 20.04 OS running on your local computer or on a remote server.
If you are looking for a VPS (Virtual Private Server), then you can register an account at DigitalOcean via this special link to get $50 free credit. (For new users only). If you are already a DigitalOcean user, then you can register an account on Vultr via this special link to get $50 free credit (for new users only).
And if you need to set up LEMP stack with a domain name, I recommend buying domain names from NameCheap because the price is low and they give whois privacy protection free for life.

Step 1: Update Software Packages

Before we install the LEMP stack, it's a good practice to update repository and software packages by running the following commands on your Ubuntu 20.04 OS.

```
apt update
```

```
apt upgrade
```

Step 2: Install Nginx Web Server

Nginx is a high-performance web server and very popular these days. It also can be used as a reverse proxy and caching server. Enter the following command to install Nginx Web server.

```
apt install nginx
```

After it's installed, we can enable Nginx to auto-start at boot time by running the following command.

```
systemctl enable nginx
```

Then start Nginx with this command:

```
systemctl start nginx
```

Now check out its status.

```
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 2020-04-10 14:11:43 UTC; 3s ago
       Docs: man:nginx(8)
    Process: 8533 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 8545 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 8549 (nginx)
```

```
      Tasks: 3 (limit: 9451)
     Memory: 3.9M
     CGroup: /system.slice/nginx.service
             ├─8549 nginx: master process /usr/sbin/nginx -g
daemon on; master_process on;
             ├─8550 nginx: worker process
             └─8551 nginx: worker process
```

"**Enabled**" indicates that auto-start at boot time is enabled and we can see that Nginx is running. You can also see how much RAM Nginx is using from the output. If the above command doesn't immediately quit after running. You need to press "**q**" to make it quit.

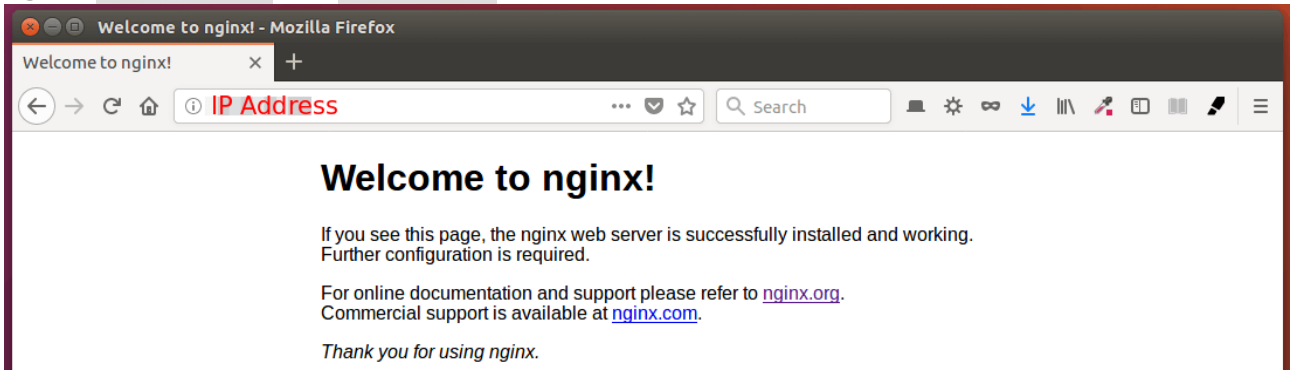
Check Nginx version.

```
nginx -v
```

Output:

```
nginx version: nginx/1.17.9 (Ubuntu)
```

Now type in the public IP address of your Ubuntu 20.04 server in the browser address bar. You should see the "Welcome to Nginx" Web page, which means Nginx Web server is running properly. If you are installing LEMP on your local Ubuntu 20.04 computer, then type `127.0.0.1` or `localhost` in the browser address bar.



If the connection is refused or failed to complete, there might be a firewall preventing incoming requests to TCP port 80. If you are using iptables firewall, then you need to run the following command to open TCP port 80.

```
iptables -I INPUT -p tcp --dport 80 -j ACCEPT
```

If you are using UFW firewall, then run this command to open TCP port 80.

```
ufw allow http
```

Finally, we need to make `www-data` (Nginx user) as the owner of web directory. By default, it's owned by the root user.

```
chown www-data:www-data /usr/share/nginx/html -R
```

Step 3: Install MariaDB Database Server

MariaDB is a drop-in replacement for MySQL. It is developed by former members of MySQL team who are concerned that Oracle might turn MySQL into a closed-source product. Enter the following command to install MariaDB on Ubuntu 20.04.

```
apt install mariadb-server mariadb-client
```

After it's installed, MariaDB server should be automatically stared. Use

systemctl to check its status.

```
systemctl status mariadb
```

Output:

```
● mariadb.service - MariaDB 10.3.22 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled;
vendor preset: enabled)
     Active: active (running) since Fri 2020-04-10 14:19:16 UTC;
18s ago
       Docs: man:mysqld(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 9161 (mysqld)
     Status: "Taking your SQL requests now..."
      Tasks: 31 (limit: 9451)
     Memory: 64.7M
     CGroup: /system.slice/mariadb.service
```

```
└─9161 /usr/sbin/mysqld
```

If it's not running, start it with this command:

```
systemctl start mariadb
```

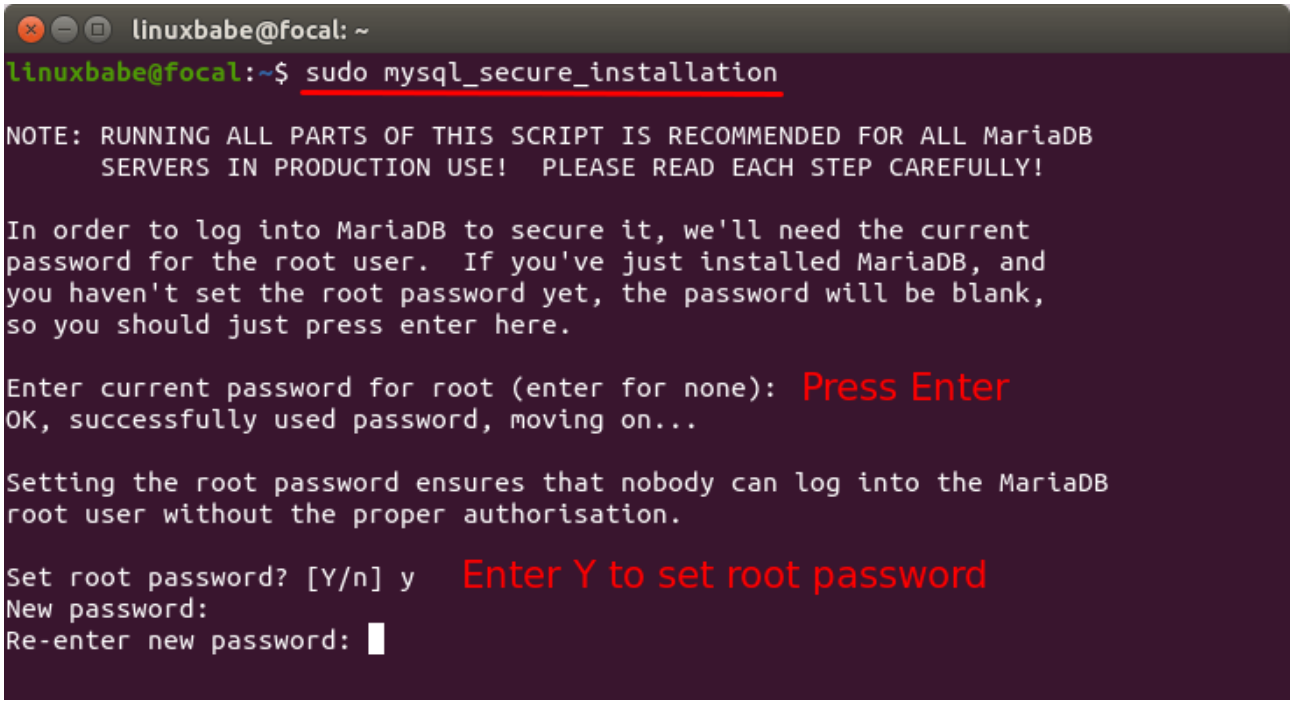
To enable MariaDB to automatically start at boot time, run

```
systemctl enable mariadb
```

Now run the post installation security script.

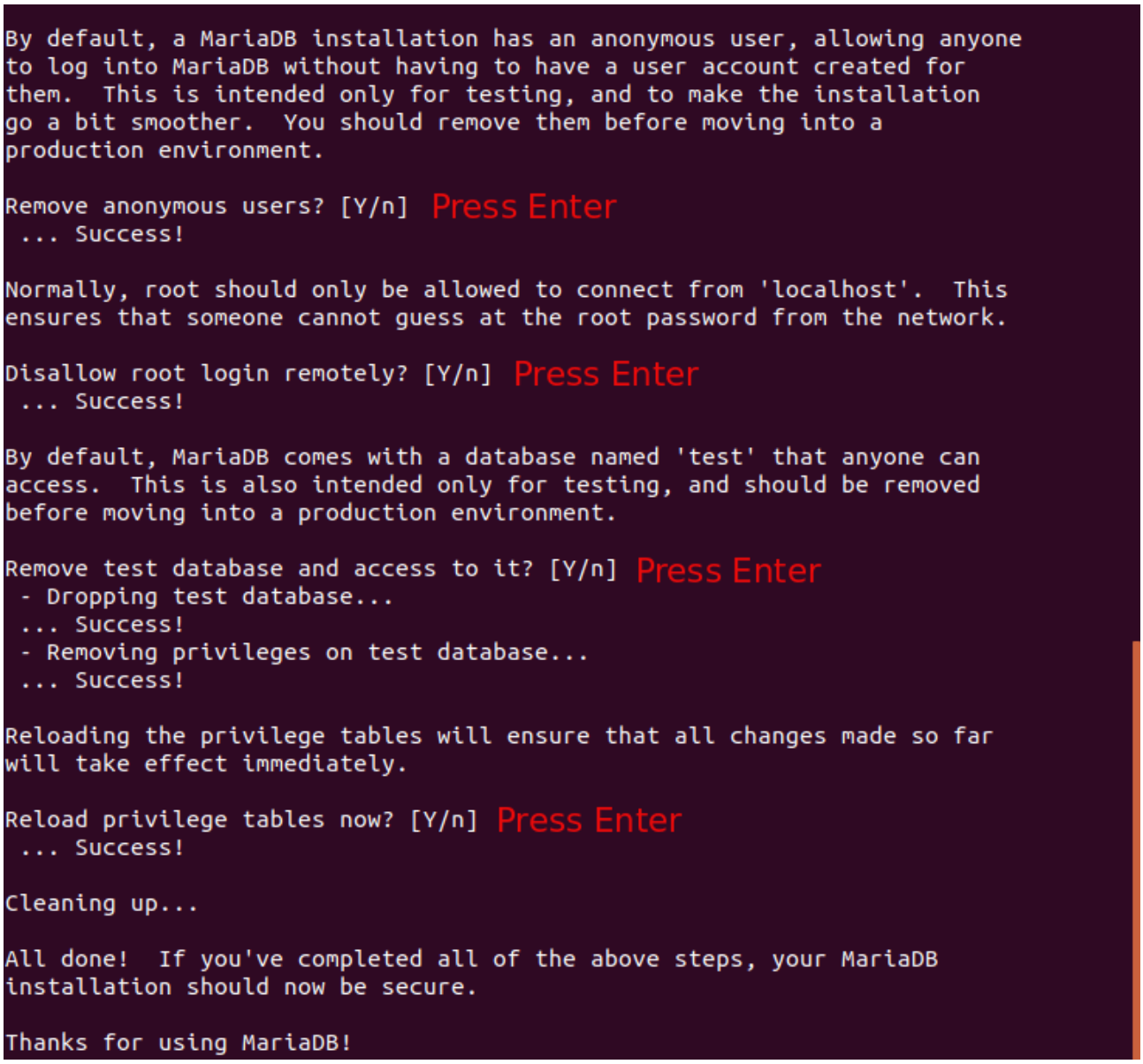
```
mysql_secure_installation
```

When it asks you to enter MariaDB root password, press Enter key as the root password isn't set yet. Then enter y to set the root password for MariaDB server.



Next, you can press Enter to answer all remaining questions, which will remove anonymous user, disable remote root login and remove test database. This step is a basic requirement for MariaDB database

security. (Notice that Y is capitalized, which means it is the default answer.)



By default, the MaraiDB package on Ubuntu uses `unix_socket` to authenticate user login, which basically means you can use username and password of the OS to log into MariaDB console. So you can run the following command to login without providing MariaDB root password.

```
mariadb -u root
```

To exit, run

```
exit;
```

Check MariaDB server version information.

```
mariadb —version
```

As you can see, we have installed MariaDB 10.3.22.

```
mariadb  Ver 15.1 Distrib 10.3.22-MariaDB, for debian-linux-gnu
(x86_64) using readline 5.2
```

Step 4: Install PHP7.4

PHP7.4 is included in Ubuntu 20.04 repository and has a minor performance improvement over PHP7.3. Enter the following command to install PHP7.4 and some common extensions.

```
apt install php7.4 php7.4-fpm php7.4-mysql php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline php7.4-mbstring php7.4-xml php7.4-gd php7.4-curl
```

PHP extensions are commonly needed for content management systems (CMS) like WordPress. For example, if your installation lacks `php7.4-xml`, then some of your WordPress site pages may be blank and you can find an error in Nginx error log like:

```
PHP message: PHP Fatal error:  Uncaught Error: Call to undefined function xml_parser_create()
```

Installing these PHP extensions ensures that your CMS runs smoothly. Now start php7.4-fpm.

```
systemctl start php7.4-fpm
```

Enable auto-start at boot time.

```
systemctl enable php7.4-fpm
```

Check status:

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

Sample output:

```
● php7.4-fpm.service - The PHP 7.4 FastCGI Process Manager
     Loaded: loaded (/lib/systemd/system/php7.4-fpm.service;
enabled; vendor pr>
     Active: active (running) since Fri 2020-04-10 14:40:26 UTC;
12s ago
       Docs: man:php-fpm7.4(8)
    Process: 21019 ExecStartPost=/usr/lib/php/php-fpm-socket-
helper install /ru>
   Main PID: 21012 (php-fpm7.4)
     Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0,
Traffic: 0req>
      Tasks: 3 (limit: 9451)
     Memory: 9.4M
     CGroup: /system.slice/php7.4-fpm.service
             ├─21012 php-fpm: master process (/etc/php/7.4/fpm/
php-fpm.conf)
             ├─21017 php-fpm: pool www
             └─21018 php-fpm: pool www
```

If the above command doesn't immediately quit after running. You need to press "**q**" to make it quit.

Step 5: Create an Nginx Server Block

An Nginx server block is like a virtual host in Apache. We will not use the default server block because it's inadequate to run PHP code and if we modify it, it becomes a mess. So remove the `default` symlink in `sites-enabled` directory by running the following command. (It's still available as `/etc/nginx/sites-available/default`.)

```
rm /etc/nginx/sites-enabled/default
```

Then use a command-line text editor like Nano to create a brand new server block file under **/etc/nginx/conf.d/** directory.

```
nano /etc/nginx/conf.d/default.conf
```

Paste the following text into the file. The following snippet will make Nginx listen on IPv4 port 80 and IPv6 port 80 with a catch-all server name.

```
server {
  listen 80;
  listen [::]:80;
  server_name _;
  root /usr/share/nginx/html/;
  index index.php index.html index.htm index.nginx-debian.html;

  location / {
    try_files $uri $uri/ /index.php;
  }

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

 # A long browser cache lifetime can speed up repeat visits to
your page
  location ~* \.(jpg|jpeg|gif|png|webp|svg|woff|woff2|ttf|css|js|
ico|xml)$ {
       access_log        off;
       log_not_found     off;
       expires           360d;
  }

  # disable access to hidden files
  location ~ /\.ht {
      access_log off;
      log_not_found off;
      deny all;
  }
}
```

Save and close the file. (To save a file in Nano text editor, press Ctrl+O, then press Enter to confirm. To exit, press Ctrl+X.)

Then test Nginx configurations.

```
nginx -t
```

If the test is successful, reload Nginx.

```
systemctl reload nginx
```

Step 6: Test PHP

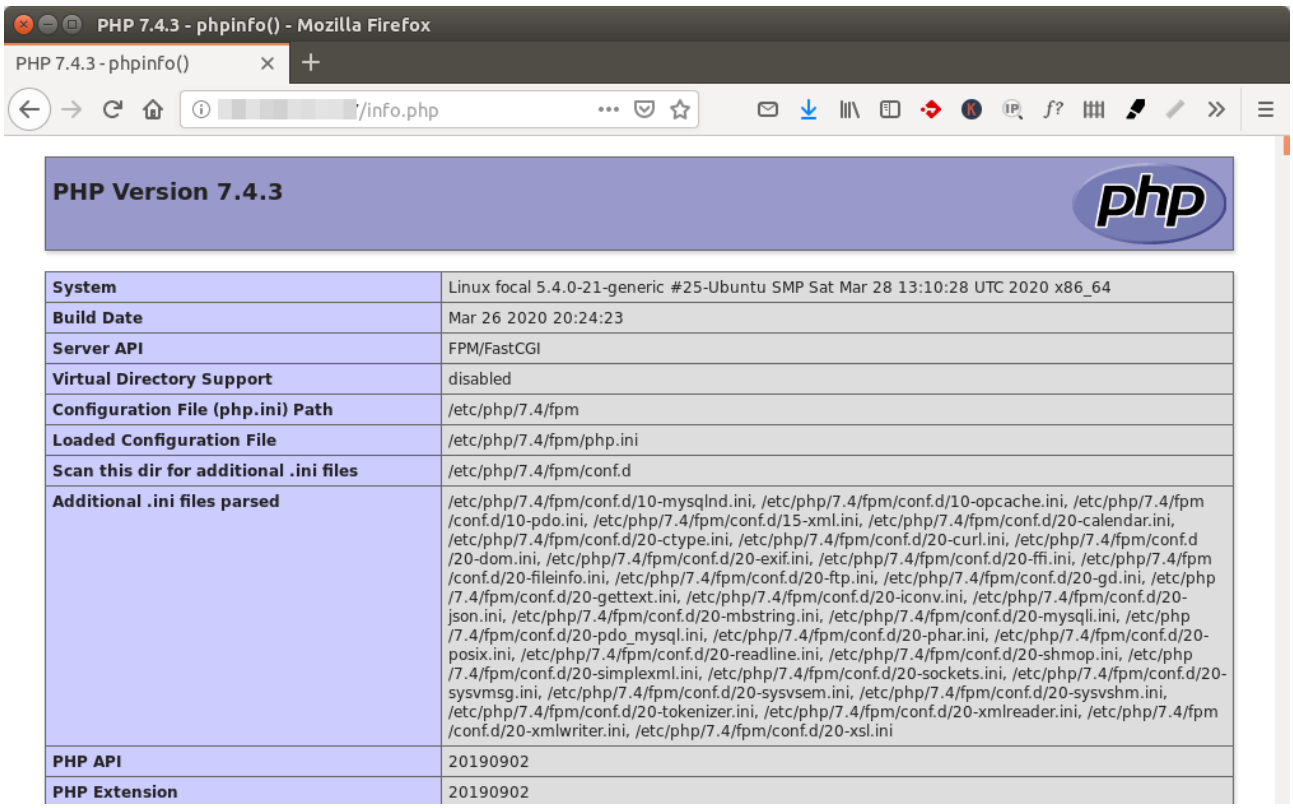
To test PHP-FPM with Nginx Web server, we need to create a info.php file in the webroot directory.

```
nano /usr/share/nginx/html/info.php
```

Paste the following PHP code into the file.

```
<?php phpinfo(); ?>
```

Save and close the file. Now in the browser address bar, enter server-ip-address/info.php. Replace sever-ip-address with your actual IP. If you follow this tutorial on your local computer, then type 127.0.0.1/info.php or localhost/info.php.
You should see your server's PHP information. This means PHP scripts can run properly with Nginx web server.



Congrats! You have successfully installed Nginx, MariaDB and PHP7.4 on Ubuntu 20.04. For your server's security, you should delete `info.php` file now to prevent hacker seeing it.

```
sudo rm /usr/share/nginx/html/info.php
```

Troubleshooting Tip

If you encounter errors, you can check the Nginx error log (`/var/log/nginx/error.log`) to find out what's wrong.

Nginx Automatic Restart

If for any reason your Nginx process is killed, you need to run the following command to restart it.

```
systemctl restart nginx
```

Instead of manually typing this command, we can make Nginx automatically restart by editing the `nginx.service` systemd service unit.

To override the default systemd service configuration, we create a separate directory.

```
mkdir -p /etc/systemd/system/nginx.service.d/
```

Then create a file under this directory.

```
nano /etc/systemd/system/nginx.service.d/restart.conf
```

Add the following lines in the file, which will make Nginx automatically restart **5 seconds** after a failure is detected. The default value of `RetartSec` is **100ms**, which is too small. Nginx may complain that "start request repeated too quickly" if `RestartSec` is not big enough.

```
[Service]
Restart=always
RestartSec=5s
```

Save and close the file. Then reload systemd.

```
systemctl daemon-reload
```

To check if this would work, kill Nginx with:

```
pkill nginx
```

Then check Nginx status. You will find Nginx automatically restarted.

```
systemctl status nginx
```

Next Step

I hope this tutorial helped you **install LEMP stack on Ubuntu 20.04 LTS**. You may want to install phpMyAdmin, which is a free and open-source web-based database management tool.

- How to Install phpMyAdmin with Nginx (LEMP) on Ubuntu 20.04 LTS

You can also install WordPress on top of the LEMP stack to create your own website or blog.

- Install WordPress on Ubuntu 20.04 with Nginx, MariaDB, PHP7.4 (LEMP)

As always, if you found this post useful, then subscribe to our free newsletter to get more tips and tricks. Take care 🙂