

LEMP Deployment on Ubuntu using Nginx Webserver

Introduction

This project demonstrates how to deploy a **LEMP stack (Linux, Nginx, MySQL, PHP)** on an **Ubuntu EC2 instance**. It covers setting up the Nginx web server, installing PHP and database support, and hosting a simple PHP application.

Prerequisites

- AWS account with an Ubuntu EC2 instance (t2.micro recommended)
- SSH key (.pem) to connect
- Security group with SSH (22) and HTTP (80) open
- Basic Linux knowledge and a web browser for testing

Steps to Deploy LEMP Stack

Step 1: Connect to EC2 Instance

Go to the folder where your SSH key (.pem) is saved, open Git Bash there, and run

```
ubuntu@ip-172-31-32-125: ~
HP@LAPTOP-HFSFVC80 MINGW64 /d/dhanashri_workspace/ssh key
$ ssh -i "key.pem" ubuntu@ec2-54-81-125-12.compute-1.amazonaws.com
The authenticity of host 'ec2-54-81-125-12.compute-1.amazonaws.com (54.81.125.12)' can't be established.
ED25519 key fingerprint is SHA256:AtMuog2YwNj2ZI7Cr+LsS6WrEXwqLa1XHbc/2sd1VLQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-81-125-12.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sat Aug 23 13:45:04 UTC 2025
```

Step 2: Update System and Install Nginx

1. Update the System

```
sudo apt update
```

```
ubuntu@ip-172-31-32-125: ~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
```

2. Install the nginx Server

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

```
ubuntu@ip-172-31-32-125:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 5 not upgraded.
```

3. Start, enable and check status of server

```
sudo systemctl start nginx
sudo systemctl enable nginx
sudo systemctl status nginx
```

```
ubuntu@ip-172-31-32-125:~$ sudo systemctl start nginx
ubuntu@ip-172-31-32-125:~$ sudo systemctl enable nginx
Synchronizing state of nginx.service with sysv service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
ubuntu@ip-172-31-32-125:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
  Active: active (running) since Sat 2025-08-23 13:50:41 UTC; 2min 2s ago
    Docs: man:nginx(8)
 Main PID: 3021 (nginx)
   Tasks: 3 (limit: 1008)
  Memory: 2.4M (peak: 5.1M)
    CPU: 27ms
   CGroup: /system.slice/nginx.service
           ├─3021 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
           ├─3023 "nginx: worker process"
           └─3024 "nginx: worker process"

Aug 23 13:50:41 ip-172-31-32-125 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server
Aug 23 13:50:41 ip-172-31-32-125 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-32-125:~$
```

Step 3: Install mysql

1. Search for mysql

```
sudo apt search mysql
```

```
ubuntu@ip-172-31-32-125:~$ sudo apt search mysql
Sorting... Done
Full Text Search... Done
akonadi-backend-mysql/noble 4:23.08.5-0ubuntu3 all
  MySQL storage backend for Akonadi
asterisk-mysql/noble 1:20.6.0~dfsg+~cs6.13.40431414-2build5 amd64
  MySQL database protocol support for the Asterisk PBX
audiofile/noble 0.95.4 all
```

2. Install mysql-server

```
sudo apt install mysql-server -y
```

```
ubuntu@ip-172-31-32-125:~$ sudo apt install mysql-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libht
  libmecab2 libprotobuf-lite32t64 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic
  mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libio-compress-brotli-perl libbusiness-isbn-
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libht
  libmecab2 libprotobuf-lite32t64 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic
```

Step 4: Install PHP

1. Search for php

```
sudo apt search php
```

```
ubuntu@ip-172-31-32-125:~$ sudo apt install php
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php8.3 php-common php8.3 php8.3-cli php8.3-common php8.3-opcache php8.3-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
```

2. Check the version of php and install php-fpm

```
php --version
sudo apt install php8.3-fpm -y
```

```
ubuntu@ip-172-31-32-125:~$ php --version
PHP 8.3.6 (cli) (built: Jul 14 2025 18:30:55) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
  with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
ubuntu@ip-172-31-32-125:~$ sudo apt install php8.3-fpm -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

3. Start, enable and check status of php8.3-fpm

```
sudo systemctl start php8.3-fpm
sudo systemctl enable php8.3-fpm
sudo systemctl status php8.3-fpm
```

```
ubuntu@ip-172-31-32-125:~$ sudo systemctl start mysql php8.3-fpm
ubuntu@ip-172-31-32-125:~$ sudo systemctl enable mysql php8.3-fpm
Synchronizing state of mysql.service with sysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mysql
Synchronizing state of php8.3-fpm.service with sysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable php8.3-fpm
ubuntu@ip-172-31-32-125:~$ sudo systemctl status mysql php8.3-fpm
● mysql.service - MySQL Community Server
  Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
  Active: active (running) since Sat 2025-08-23 13:54:35 UTC; 6min ago
    Main PID: 3957 (mysqld)
      Status: "Server is operational"
     Tasks: 37 (limit: 1008)
    Memory: 351.3M (peak: 377.1M)
      CPU: 3.616s
     CGroup: /system.slice/mysql.service
             └─3957 /usr/sbin/mysqld
```

Step 5: Go into the default directory /var/www/html/ and create an index.html and index.php file for testing.

1. Create index.html and index.php

```
ubuntu@ip-172-31-32-125:~/var/www/html
ubuntu@ip-172-31-32-125:~$ cd /var/www/html/
ubuntu@ip-172-31-32-125:/var/www/html$ sudo vim index.html
ubuntu@ip-172-31-32-125:/var/www/html$ ls
index.html  index.nginx-debian.html
ubuntu@ip-172-31-32-125:/var/www/html$ sudo rm -rf index.html
ubuntu@ip-172-31-32-125:/var/www/html$ sudo vim index.html
ubuntu@ip-172-31-32-125:/var/www/html$ sudo vim index.php
ubuntu@ip-172-31-32-125:/var/www/html$ ...
```

2. Add content in index.html

```
ubuntu@ip-172-31-32-125:~/var/www/html
<h1> This is LEMP </h1>
~
~
~
~
~
~
~
```

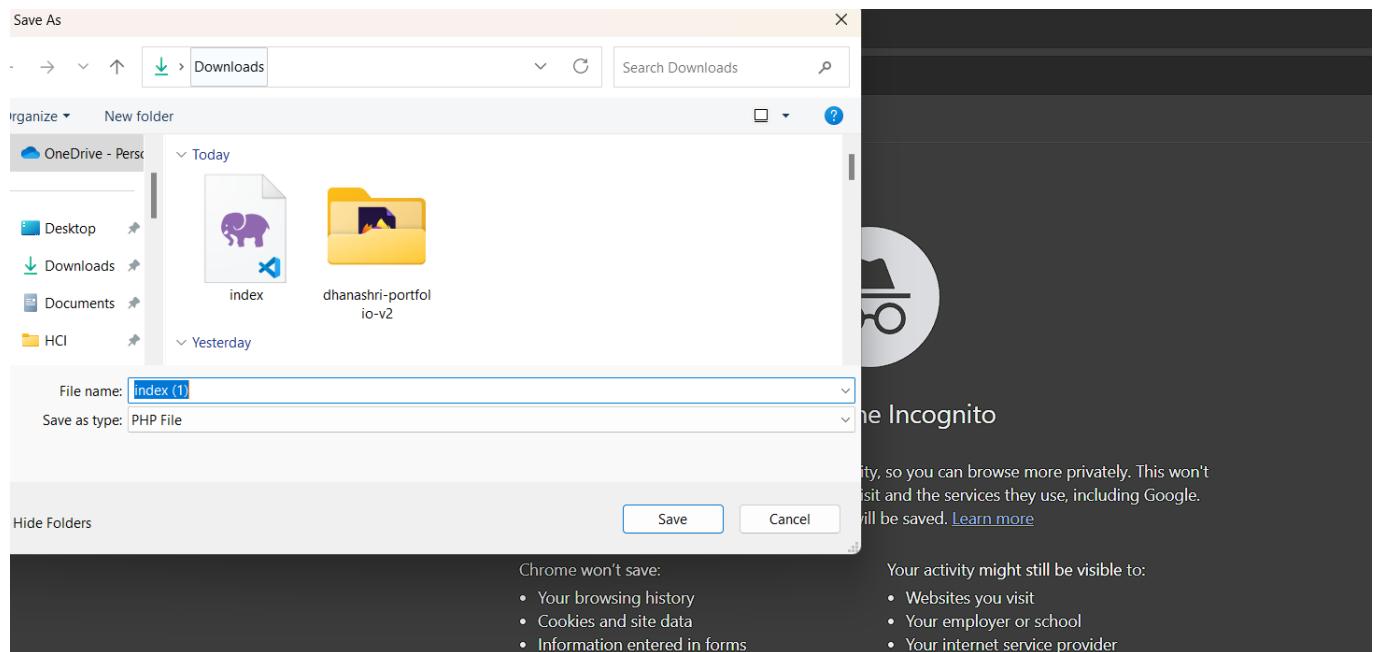
3. Add content in index.php

```
ubuntu@ip-172-31-32-125:~/var/www/html
<?php
    phpinfo();
?>
~
~
```

Step 6: Restart nginx, mysql, php8.3-fpm

```
ubuntu@ip-172-31-32-125:~/var/www/html
ubuntu@ip-172-31-32-125:/var/www/html$ sudo systemctl restart nginx mysql php8.3-fpm
ubuntu@ip-172-31-32-125:/var/www/html$ |
```

Step 7: Copy the Public IP Address and paste in any browser for testing deployment.



On Ubuntu LEMP (Nginx + PHP), if your .php file is getting downloaded instead of executing, it means Nginx is not configured to process PHP through PHP-FPM.

Step 8: Fix for PHP Download Issue in Nginx

1. Change Directory to `/etc/nginx/sites-enabled` and go to default file.

```
ubuntu@ip-172-31-32-125: /etc/nginx/sites-enabled
ubuntu@ip-172-31-32-125: /var/www/html$ cd /etc/nginx/sites-enabled/
ubuntu@ip-172-31-32-125: /etc/nginx/sites-enabled$ sudo vim default
ubuntu@ip-172-31-32-125: /etc/nginx/sites-enabled$ sudo systemctl restart nginx mysql php8.3-fpm
```

2. Inside the default Nginx config file, uncomment the PHP location block and update the PHP-FPM

version to match your installed PHP version.

```
ubuntu@ip-172-31-32-125:/etc/nginx/sites-enabled
#
# Self signed certs generated by the ssl-cert package
# Don't use them in a production server!
#
# include snippets/snakeoil.conf;

root /var/www/html;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

# pass PHP scripts to FastCGI server
#
#location ~ \.php$ {
#    include snippets/fastcgi-php.conf;
#
#    # With php-fpm (or other unix sockets):
#    fastcgi_pass unix:/run/php/php8.3-fpm.sock;
#    # With php-cgi (or other tcp sockets):
#    fastcgi_pass 127.0.0.1:9000;
#}

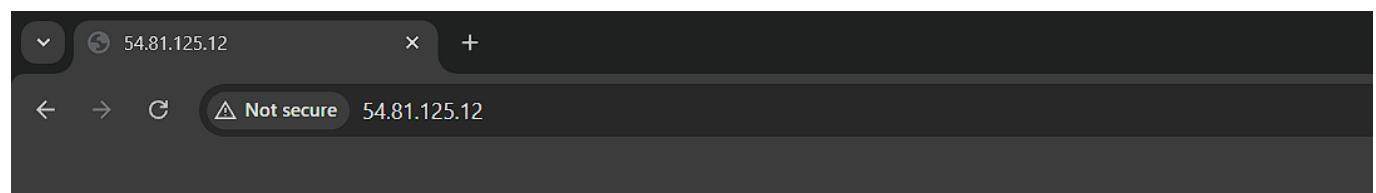
# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
#}

# Virtual Host configuration for example.com
-- INSERT --
```

Step 9: Restart nginx, mysql, php8.3-fpm

```
ubuntu@ip-172-31-32-125:/etc/nginx/sites-enabled
ubuntu@ip-172-31-32-125:/var/www/html$ cd /etc/nginx/sites-enabled/
ubuntu@ip-172-31-32-125:/etc/nginx/sites-enabled$ sudo vim default
ubuntu@ip-172-31-32-125:/etc/nginx/sites-enabled$ sudo systemctl restart nginx mysql php8.3-fpm
```

Step 10: Testing the Deployment



This is LEMP

The screenshot shows a web browser window with the title "PHP 8.3.6 - phpinfo()". The URL in the address bar is "54.81.125.12/index.php". The page content is a table titled "PHP Version 8.3.6" with a "php" logo in the top right corner. The table contains numerous rows of configuration parameters, such as "System" (Linux ip-172-31-32-125 6.14.0-1011-aws #11-24.04.1-Ubuntu SMP Fri Aug 1 02:07:25 UTC 2025 x86_64), "Build Date" (Jul 14 2025 18:30:55), "Server API" (FPM/FastCGI), and "Configuration File (php.ini) Path" (/etc/php/8.3/fpm). The "Additional .ini files parsed" row lists several files under /etc/php/8.3/fpm/conf.d/. The bottom of the table shows PHP API (20230831), PHP Extension (20230831), and Zend Extension (420230831).

PHP Version 8.3.6	
System	Linux ip-172-31-32-125 6.14.0-1011-aws #11-24.04.1-Ubuntu SMP Fri Aug 1 02:07:25 UTC 2025 x86_64
Build Date	Jul 14 2025 18:30:55
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/fpm
Loaded Configuration File	/etc/php/8.3/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/fpm/conf.d
Additional .ini files parsed	/etc/php/8.3/fpm/conf.d/10-opcache.ini, /etc/php/8.3/fpm/conf.d/10-pdo.ini, /etc/php/8.3/fpm/conf.d/20-calendar.ini, /etc/php/8.3/fpm/conf.d/20-cgi-fcgi.ini, /etc/php/8.3/fpm/conf.d/20-discard.ini, /etc/php/8.3/fpm/conf.d/20-fpm.ini, /etc/php/8.3/fpm/conf.d/20-gettext.ini, /etc/php/8.3/fpm/conf.d/20-fileinfo.ini, /etc/php/8.3/fpm/conf.d/20-xml.ini, /etc/php/8.3/fpm/conf.d/20-phar.ini, /etc/php/8.3/fpm/conf.d/20-posix.ini, /etc/php/8.3/fpm/conf.d/20-readline.ini, /etc/php/8.3/fpm/conf.d/20-shmop.ini, /etc/php/8.3/fpm/conf.d/20-sockets.ini, /etc/php/8.3/fpm/conf.d/20-sysvmsg.ini, /etc/php/8.3/fpm/conf.d/20-sysvsem.ini, /etc/php/8.3/fpm/conf.d/20-sysvshm.ini, /etc/php/8.3/fpm/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831

Summary

In this project, we deployed a **LEMP** stack on Ubuntu using the Nginx webserver. We connected to an EC2 instance via SSH, installed and configured **Nginx, PHP, and MySQL**, and set up the default directory `/var/www/html/` with test HTML and PHP files. We also configured Nginx to work with PHP-FPM to ensure PHP files execute properly instead of downloading.