THE PERFECT NGINX SERVER

CENTOS 8

LEMP STACK

NGINX | MARIADB | PHP

INSTALL | HARDEN | OPTIMIZE

UPDATED DECEMBER 2020

nginx version - 1.18

php 8.0

PACKAGE MANAGER COMMANDS

```
sudo dnf install package_name(s)

sudo dnf remove package_name(s)

sudo dnf search package_name

sudo dnf info packagename
```

NGINX MARIADB & PHP

INSTALL NGINX

```
sudo dnf update
```

Check versions that are available: The default version is 1.14 but we can install version 1.16

```
sudo dnf module list nginx
```

Enable and install the latest version - updated December 2020 - version 1.18

```
sudo dnf module reset nginx

sudo dnf module enable nginx:1.16

sudo dnf module list nginx

sudo dnf install nginx
```

DECEMBER 2020 UPDATE

Newer version of nginx available: 1.18 - to install:

```
sudo dnf module enable nginx:1.18
```

Start and enable the nginx service

```
sudo systemctl start nginx && sudo systemctl enable nginx

sudo systemctl status nginx
```

Fix bug:

```
sudo bash -c 'printf "[Service]\nExecStartPost=/bin/sleep 0.1\n" > /etc/systemd/system/nginx.service.d/override.conf'

sudo systemctl daemon-reload

sudo systemctl restart nginx
```

Open your browser and type in your server's ip address:

http://ip_address/

The default nginx page will be displayed.

INSTALL MARIADB

Install MariaDB

```
sudo dnf update

sudo dnf install mariadb-server


sudo systemctl start mariadb && sudo systemctl enable mariadb

sudo systemctl status mariadb
```

INSTALL PHP

IMPORTANT NOTICE - PHP 8.0

As of December 2020, I don't recommend php 8 and WordPress 5.6 be used on a production server.

Time is needed for the theme and plugin developers to release versions that are fully compatible with WordPress 5.6 and php 8.

For this reason, **test WordPress 5.6 and php8.0 on a test server first, not your production server.**

Your production server is not used to test new software releases!!!

INSTALL PHP 7.4

```
sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm

sudo dnf install https://rpms.remirepo.net/enterprise/remi-release-8.rpm

sudo dnf install yum-utils

sudo dnf module reset php

sudo dnf module install php:remi-7.4
```

A slightly easier method of installing lots of packages - command typed on a single line

```
sudo dnf install php-
{fpm,gd,json,mbstring,mysqlnd,xml,xmlrpc,opcache,cli,zip,soap,intl,bcmath,cur
l,ssh2}
```

Command typed on a single line

```
sudo dnf install php-{fpm,gd,json,mbstring,mysqlnd,xml,xmlrpc,opcache,cli,zip,soap,intl,bcmath,curl,ssh2}
```

Start and Enable the PHP service

```
sudo systemctl start php-fpm && sudo systemctl enable php-fpm

sudo systemctl status php-fpm
```

INSTALL PHP 8.0

```
sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm

sudo dnf install https://rpms.remirepo.net/enterprise/remi-release-8.rpm

sudo dnf install yum-utils

sudo dnf module reset php

sudo dnf module install php:remi-8.0
```

A slightly easier method of installing lots of packages

```
sudo dnf install php-
{fpm,gd,json,mbstring,mysqlnd,xml,xmlrpc,opcache,cli,zip,soap,intl,bcmath,cur
l,ssh2}
```

Command typed on a single line

```
sudo dnf install php-{fpm,gd,json,mbstring,mysqlnd,xml,xmlrpc,opcache,cli,zip,soap,intl,bcmath,curl,ssh2}
```

Start and Enable the PHP service

```
sudo systemctl start php-fpm && sudo systemctl enable php-fpm

sudo systemctl status php-fpm
```

CONFIGURE PHP - 7.4 and 8.0

```
cd /etc/php-fpm.d/
sudo cp www.conf www.conf.bak
sudo nano www.conf
```

Edit the www.conf file as per the video lectures.

To complete the installation, remove the php-fpm.conf file from the /etc/nginx/conf.d directory:

```
cd /etc/nginx/conf.d
sudo rm php-fpm.conf
cd /etc/nginx/default.d
sudo rm php.conf
sudo systemctl restart php-fpm
sudo systemctl reload nginx
```

SECURE THE MARIADB

```
sudo mysql_secure_installation
```

SECURE PHP

```
cd /etc/

sudo cp php.ini php.ini.bak

sudo nano php.ini
```

Changes:

```
allow_url_fopen = Off

cgi.fix_pathinfo=0

expose_php = Off
```

OPTIMZIE MARIADB

```
cd /etc/my.cnf.d

sudo cp mariadb-server.cnf mariadb-server.cnf.bak

sudo nano mariadb-server.cnf
```

```
# Performance Schema

performance_schema=ON

performance-schema-instrument='stage/%=ON'

performance-schema-consumer-events-stages-current=ON

performance-schema-consumer-events-stages-history=ON

performance-schema-consumer-events-stages-history-long=ON
```

```
sudo systemctl restart mariadb
```

UPDATE: url has changed, download the following Sys Schema from github.com to your home directory

```
cd

sudo dnf install wget zip unzip

wget https://github.com/FromDual/mariadb-sys/archive/master.zip
```

Extract the master.zip file:

```
unzip mariadb-sys-master.zip
```

Change to the extracted directory and install the Sys Schema, the schema filename has changed

```
cd mariadb-sys-master/

mysql -u root -p < ./sys_10.sql

cd
```

Delete the master.zip file and the mariadb-sys-master directory:

```
rm -rf mariadb-sys-master/ master.zip
```

Download MySQLTuner to your home directory:

```
cd

mkdir MySQLTuner/

cd MySQLTuner/

wget http://mysqltuner.pl/ -O mysqltuner.pl
```

Give MySQLTuner executable permissions:

```
chmod +x mysqltuner.pl
```

To run mysqltuner:

```
./mysqltuner.pl
```

Follow the video lectures at this point…

OPTIMIZE PHP: Both 7.4 and 8

PHP.INI

```
cd /etc/
sudo nano php.ini
```

Changes to php.ini

```
upload_max_filesize = 100M
post_max_size = 100M
max_execution_time = 30
max_input_time = 60
max_input_vars = 3000
memory_limit = 256M
```

OPCACHE

```
cd /etc/php.d/
sudo cp 10-opcache.ini 10-opcache.ini.bak
sudo nano 10-opcache.ini
```

Changes to 10-opcache.ini

```
opcache.enable=1
opcache.memory_consumption=192
opcache.interned_strings_buffer=16
opcache.max_accelerated_files=7963
opcache.validate_timestamps=0
```

Restart php-fpm

```
sudo systemctl restart php-fpm
```

Determine number of php files, refer to video for detailed instructions

```
cd /var/www
find . -type f -print | grep php | wc -l
```

Setting the Process Manager Values: default values as per server RAM

```
cd /etc/php-fpm.d/

sudo nano www.conf
```

process manager values

DIRECTIVE	1GIG RAM Server	2GIG RAM Server	4GIG RAM Server
pm	dynamic	dynamic	dynamic
pm.max_children	8	18	27
pm.start_servers	4	7	11
pm.min_spare_servers	2	4	6
pm.max_spare_servers	5	10	16
pm.max_requests	250	500	500

IMPORTANT NOTICE

Copying and then pasting commands from a pdf can produce errors when pasting the commands into your terminal emulator.

Confirm the accuracy & correctness of the command before pressing enter.