Dockerfile

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mariadb-php-lemp-stack-on-debian-10>

1. Скачиваем необходимые инструменты

* vim
* установка веб-сервера Nginx
* установить систему баз данных default-mysql-server php7.3
* Установка PHP для обработки.

1. php7.3-mysql
2. php-mbstring
3. php-json
4. hp7.3
5. php7.3-fpm - PHP fastCGI process manager", и сказать Nginx, чтобы он передавал запросы PHP этому программному обеспечению для обработки.

* wordpress
* openssl

1. Скачиваем phpmyadmin с ссылки, присваиваем имя, распаковываем, перемещаем
2. Настройка Nginx для использования PHP-процессора
   1. Затем назначьте владельцу каталога переменную окружения $USER, которая должна ссылаться на текущего системного пользователя:

sudo chown -R $USER:$USER /var/www/your\_domain

* 1. default

1. конфигурация, которая прослушивает порт *80* и обслуживает файлы из веб-корня, который вы только что создали. Он будет отвечать только на запросы к хосту или IP-адресу, предоставленному после *server\_name* этого, и любые файлы, заканчивающиеся на *.php* этом, будут обработаны *php-fpm* до того, как Nginx отправит результаты пользователю.
2. Активируйте свою конфигурацию, связавшись с конфигурационным файлом из *sites-enabled* каталога Nginx:

sudo ln -s /etc/nginx/sites-available/your\_domain /etc/nginx/sites-enabled*/*

1. Генерируем самоподписные коды SSL (.crt - сертификат CA; .key - ключ CA, приватный ключ)
2. Редактируем wp-config.php (wp-config-sample.php.)

[*https://zacompom.ru/lessons/fajl-wp-config-php-podrobnoe-opisanie-nastroek-wordpress.html*](https://zacompom.ru/lessons/fajl-wp-config-php-podrobnoe-opisanie-nastroek-wordpress.html)

[*https://codex.wordpress.org/Редактирование\_wp-config.php*](https://codex.wordpress.org/Редактирование_wp-config.php)

[*https://wpschool.ru/wpconfig-tricks/*](https://wpschool.ru/wpconfig-tricks/)

23: define( 'DB\_NAME', '<объект> ' );

1. Создание пользователей MySQL/MariaDB и предоставление прав доступа (\*.sql)

<https://www.dmosk.ru/miniinstruktions.php?mini=mysql-user>

<https://obu4alka.ru/sozdaem-polzovatelya-s-pravami-root-v-mysql.html>

* Создайте пользователя для phpMyAdmin

CREATE USER 'phpmyadmin'@'localhost' IDENTIFIED BY 'some\_pass';

(GRANT <тип привилегий> ON <объект> TO <пользователь> [IDENTIFIED BY <пароль>] <доп. опции>;)

GRANT ALL PRIVILEGES ON \*.\* TO 'phpmyadmin'@'localhost' WITH GRANT OPTION;

* Следующая инструкция SQL создает базу данных с именем "testDB":

CREATE DATABASE testDB;

(Перед созданием базы данных убедитесь, что у вас есть права администратора. После создания базы данных ее можно проверить в списке баз данных со следующей командой SQL: Показать базы данных)

* Разрешить пользователю root подключаться к MySQL без sudo

UPDATE user set plugin='' WHERE User='root';

* После чего надо обновить привилегии пользователей

FLUSH PRIVILEGES;

1. И config.inc.php

<https://php-myadmin.ru/doc/config.html>

1. Настраиваем autoindex
2. Прописываем start.sh

<http://ruunix.ru/1502-phpmyadmin-ustanovka-reshenie-problem-pri-ustanovke-i-nastrojke.html>

1. EXPOSE 80 443
2. ENTRYPOINT bash

Tar – архив

Для создания архива стандартная комбинация *cvzf*

Для распаковки *xvf*

Давайте разберемся, что все эти флаги означают:

* **c** — создать архив в linux
* **x** — извлечь файлы из архива
* **v** — показать подробную информацию о процессе работы (без него мы бы не увидели, какие файлики запаковались / распаковались)
* **f** — файл для записи архива
* **z** — сжатие

<https://kapeli.com/cheat_sheets/Dockerfile.docset/Contents/Resources/Documents/index> инструкции (команды) по докеру

<https://evilinside.ru/docker-cleanup-ochistka-docker/> хорошая ссылка про чистку