

попробуем инструкцию с <https://admin812.ru/razvertyvanie-wordpress-s-nginx-php-fpm-i-mariadb-s-pomoshhyu-docker-compose.html>

```
sudo apt update && sudo apt-get upgrade
```

```
sudo apt install docker.io
```

```
sudo systemctl start docker
```

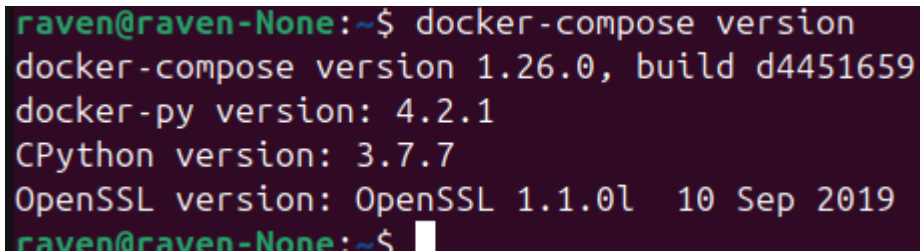
```
sudo systemctl enable docker
```

```
sudo apt install docker-compose
```

```
sudo wget -O /usr/local/bin/docker-compose
```

```
https://github.com/docker/compose/releases/download/1.26.0/docker-compose-Linux-x86\_64
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

A terminal window with a dark background and light-colored text. The prompt is 'raven@raven-None:~\$'. The command 'docker-compose version' has been executed, resulting in the following output: 'docker-compose version 1.26.0, build d4451659', 'docker-py version: 4.2.1', 'CPython version: 3.7.7', and 'OpenSSL version: OpenSSL 1.1.0l 10 Sep 2019'. The prompt is now 'raven@raven-None:~\$' with a cursor.

```
raven@raven-None:~$ docker-compose version
docker-compose version 1.26.0, build d4451659
docker-py version: 4.2.1
CPython version: 3.7.7
OpenSSL version: OpenSSL 1.1.0l 10 Sep 2019
raven@raven-None:~$
```

```
mkdir -p wordpress-compose
```

```
mkdir -p nginx/
```

```
mkdir -p logs/
```

```
mkdir -p logs/nginx
```

```
mkdir -p data/
```

```
mkdir -p data/html
```

```
mkdir -p data/mysql
```

```
nano ~/wordpress-compose/nginx/nginx.conf
```

Вставляем следующее:

```
server {
    listen 80;
    server_name site1.ru;
```

```
root /var/www/html;
```

```
index index.php;
```

```
access_log /var/log/nginx/site-access.log;
```

```
error_log /var/log/nginx/site-error.log;
```

```
location / {
```

```
    try_files $uri $uri/ /index.php?$args;
```

```
}
```

```
location ~ /\.php$ {
```

```
    try_files $uri =404;
```

```
    fastcgi_split_path_info ^(.+\.php)(/.+)$;
```

```
    fastcgi_pass wordpress:9000;
```

```
    fastcgi_index index.php;
```

```
    include fastcgi_params;
```

```
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
```

```
    fastcgi_param PATH_INFO $fastcgi_path_info;
```

```
}
```

```
}
```

```
nano docker-compose.yml
```

Вставляем следующее:

```
version: '3'
```

```
services:
```

```
  mysql:
```

```
    image: mariadb
```

```
    volumes:
```

```
      - ./data/mysql:/var/lib/mysql
```

```
    environment:
```

```

MYSQL_ROOT_PASSWORD: 4m3SxCpaQb*6LB#
MYSQL_DATABASE: db_wordpress
MYSQL_USER: user_wp
MYSQL_PASSWORD: 9UqfqeE9#hn2@t^
restart: always
wordpress:
  image: wordpress:php7.4-fpm-alpine
  volumes:
    - ./data/html:/var/www/html
  depends_on:
    - mysql
  environment:
    WORDPRESS_DB_HOST: mysql
    MYSQL_ROOT_PASSWORD: 4m3SxCpaQb*6LB#
    WORDPRESS_DB_NAME: db_wordpress
    WORDPRESS_DB_USER: user_wp
    WORDPRESS_DB_PASSWORD: 9UqfqeE9#hn2@t^
    WORDPRESS_TABLE_PREFIX: wp_
  links:
    - mysql
  restart: always
nginx:
  image: nginx:latest
  volumes:
    - ./nginx:/etc/nginx/conf.d
    - ./data/html:/var/www/html
  ports:
    - 8080:80
  links:
    - wordpress

pma:
  image: phpmyadmin/phpmyadmin
  restart: always
  links:
    - mysql:mysql
  ports:
    - 8081:80
  environment:
    MYSQL_ROOT_PASSWORD: 4m3SxCpaQb*6LB#

```

Sudo docker-compose up -d

```

raven@raven-None:~/wordpress-compose$ sudo docker-compose ps

```

Name	Command	State	Ports
wordpress-	docker-entrypoint.sh	Up	3306/tcp
compose_mysql_1	mariadb		
wordpress-	/docker-entrypoint.sh	Up	0.0.0.0:8080->80/tcp,:
compose_nginx_1	nginx ...		::8080->80/tcp
wordpress-compose_pma_1	/docker-entrypoint.sh	Up	0.0.0.0:8081->80/tcp,:
	apac ...		::8081->80/tcp
wordpress-	docker-entrypoint.sh	Up	9000/tcp
compose_wordpress_1	php-fpm		

```

raven@raven-None:~/wordpress-compose$

```

Результат:

