

### Установка Docker:

```
sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common -y
```

установка ключа для репозитория:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

установка репозитория:

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

установка докера:

```
sudo apt update; sudo apt install docker-ce -y
```

Добавляем себя в группу docker, чтобы выполнять команды без sudo:

```
sudo usermod -a -G docker $USER (выйти, зайти, чтобы сработало. Проверка: id)
```

проверка статуса докера:

```
ps ax | grep docker
```

запуск докера:

```
sudo systemctl start docker.service
```

```
mkdir wordpress && cd wordpress
```

```
mkdir nginx-conf
```

```
nano nginx-conf/nginx.conf
```

```
server {  
  
    listen 80;  
  
    listen [::]:80;  
  
    server_name example.com www.example.com;  
  
    index index.php index.html index.htm;  
  
    root /var/www/html;
```

```
location ~ /\.well-known/acme-challenge {

    allow all;

    root /var/www/html;

}


location / {

    try_files $uri $uri/ /index.php$is_args$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;

}


location ~ /\.ht {

    deny all;

}


location = /favicon.ico {

    log_not_found off; access_log off;
```

```

    }

    location = /robots.txt {

        log_not_found off; access_log off; allow all;

    }

    location ~* \.(css|gif|ico|jpeg|jpg|js|png)$ {

        expires max;

        log_not_found off;

    }

}

```

nano .env

```
MYSQL_ROOT_PASSWORD=123
```

```
MYSQL_USER=user1
```

```
MYSQL_PASSWORD=123
```

nano .dockerignore

```
.env
```

```
.git
```

```
docker-compose.yml
```

```
.dockerignore
```

Установка

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.26.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

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

nano docker-compose.yml

```
version: '3'
```

```
services:

  db:

    image: mysql:8.0

    container_name: db

    restart: unless-stopped

    env_file: .env

    environment:

      - MYSQL_DATABASE=wordpress

    volumes:

      - dbdata:/var/lib/mysql

    command: '--default-authentication-plugin=mysql_native_password'

    networks:

      - app-network
```

```
wordpress:

  depends_on:

    - db

  image: wordpress:5.1.1-fpm-alpine

  container_name: wordpress

  restart: unless-stopped

  env_file: .env

  environment:

    - WORDPRESS_DB_HOST=db:3306

    - WORDPRESS_DB_USER=$MYSQL_USER

    - WORDPRESS_DB_PASSWORD=$MYSQL_PASSWORD
```

```
- WORDPRESS_DB_NAME=wordpress

volumes:

- wordpress:/var/www/html

networks:

- app-network

webserver:

depends_on:

- wordpress

image: nginx:1.15.12-alpine

container_name: webserver

restart: unless-stopped

ports:

- "80:80"

volumes:

- wordpress:/var/www/html

- ./nginx-conf:/etc/nginx/conf.d

- certbot-etc:/etc/letsencrypt

networks:

- app-network

volumes:

certbot-etc:

wordpress:

dbdata:

networks:
```

app-network:

driver: bridge

docker-compose up -d

```
root@ubuntu-server: ~/wordpress
f3f5ff008d73: Pull complete
dd7054d6d0c7: Pull complete
70b5d4e8750e: Pull complete
cdc4a7b43bdd: Pull complete
a0608f8959e0: Pull complete
5d23e721608f: Pull complete
a564ada530a5: Pull complete
339565d00e89: Pull complete
a11a06843fd5: Pull complete
92f6d4aa041d: Pull complete
Digest: sha256:a43f6e7e7f3a5e5b90f857fbed4e3103ece771b19f0f75880f767cf66bbb6577
Status: Downloaded newer image for mysql:8.0
Pulling wordpress (wordpress:5.1.1-fpm-alpine)...
5.1.1-fpm-alpine: Pulling from library/wordpress
bdf0201b3a05: Pull complete
eae6561c50e0: Pull complete
f591654c6ada: Pull complete
f205a7399250: Pull complete
be168a59f377: Pull complete
073a30dhac02: Pull complete
d94cea587efe: Pull complete
6ed026e6424c: Pull complete
695dbe186452: Pull complete
79f0d83843fe: Pull complete
f57cd0d319e8: Pull complete
838d818ea958: Pull complete
0f06001e8eff: Pull complete
42bae694670e: Pull complete
a85be5b9eb59: Pull complete
537d24b2e974: Pull complete
Digest: sha256:4d6db7f7b368d6ade40510f27ac6d16e2b690b49bbdb237435f1f21234d8436a
Status: Downloaded newer image for wordpress:5.1.1-fpm-alpine
Pulling webserver (nginx:1.15.12-alpine)...
1.15.12-alpine: Pulling from library/nginx
e7c96db7181b: Pull complete
264026b2e255: Pull complete
a71634c5d29: Pull complete
5595887eb81: Pull complete
Digest: sha256:97a226f6cab6823027c0704a9346a890ffb0cacde06bc19bbc234c8720673555
Status: Downloaded newer image for nginx:1.15.12-alpine
Creating db ... done
Creating wordpress ... done
Creating webserver ... done
root@ubuntu-server:~/wordpress#
```

docker-compose ps

```
root@ubuntu-server: ~/wordpress
image: wordpress:5.1.1-fpm-alpine
container_name: wordpress
restart: unless-stopped
env_file: .env
environment:
  - WORDPRESS_DB_HOST=db:3306
  - WORDPRESS_DB_USER=$MYSQL_USER
  - WORDPRESS_DB_PASSWORD=$MYSQL_PASSWORD
  - WORDPRESS_DB_NAME=wordpress
volumes:
  - wordpress:/var/www/html
networks:
  - app-network

webserver:
  depends_on:
    - wordpress
  image: nginx:1.15.12-alpine
  container_name: webserver
  restart: unless-stopped
  ports:
    - "80:80"
  volumes:
    - wordpress:/var/www/html
    - ./nginx-conf:/etc/nginx/conf.d
    - certbot-etc:/etc/letsencrypt
  networks:
    - app-network

volumes:
  certbot-etc:
  wordpress:
  dbdata:

networks:
  app-network:
    driver: bridge

root@ubuntu-server:~/wordpress# docker-compose ps

```

Name	Command	State	Ports
db	docker-entrypoint.sh --def ...	Up	3306/tcp, 33060/tcp
webserver	nginx -g daemon off;	Up	0.0.0.0:80->80/tcp, :::80->80/tcp
wordpress	docker-entrypoint.sh php-fpm	Up	9000/tcp

```
root@ubuntu-server:~/wordpress#
```

После запуска ваших контейнеров вы можете завершить процесс установки WordPress через веб-интерфейс.

2.

docker network ls

```
root@ubuntu-server: ~/wordpress
- WORDPRESS_DB_USER=${MYSQL_USER}
- WORDPRESS_DB_PASSWORD=${MYSQL_PASSWORD}
- WORDPRESS_DB_NAME=wordpress
volumes:
- wordpress:/var/www/html
networks:
- app-network

webserver:
depends_on:
- wordpress
image: nginx:1.15.12-alpine
container_name: webserver
restart: unless-stopped
ports:
- "80:80"
volumes:
- wordpress:/var/www/html
- ./nginx-conf:/etc/nginx/conf.d
- certbot-etc:/etc/letsencrypt
networks:
- app-network

volumes:
certbot-etc:
wordpress:
dbdata:

networks:
app-network:
driver: bridge
root@ubuntu-server:~/wordpress# docker-compose ps
Name Command State Ports
-----
db docker-entrypoint.sh --def ... Up 3306/tcp, 33060/tcp
webserver nginx -g daemon off; Up 0.0.0.0:80->80/tcp, :::80->80/tcp
wordpress docker-entrypoint.sh php-fpm Up 9000/tcp
root@ubuntu-server:~/wordpress# docker network ls
NETWORK ID NAME DRIVER SCOPE
d9e61cb429f2 bridge bridge local
e5c9d18ed87c host host local
047099f4f4cda none null local
31d20badfafe wordpress_app-network bridge local
root@ubuntu-server:~/wordpress#
```

mkdir mariadb

```
docker pull mariadb:10.8
```

или

```
docker run --rm -ti --net learn-networking --name container1 mariadb:10.8
```

Второй контейнер:

```
docker run --rm -ti --net learn-networking --name container2 phpmyadmin
```

nc -lp 1234 (в терминале второго контейнера)

nc container2 1234 (в терминале первого контейнера)