

## Задание:

Установить в виртуальную машину или VDS Docker, настроить набор контейнеров через docker compose по инструкции по ссылке: <https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-docker-compose-ru>. Часть с настройкой certbot и HTTPS опустить, если у вас нет настоящего домена и белого IP.

\* Запустить два контейнера, связанные одной сетью (используя документацию). Первый контейнер БД (например, образ mariadb:10.8), второй контейнер — phpmyadmin. Получить доступ к БД в первом контейнере через второй контейнер (веб-интерфейс phpmyadmin).

Результат:

Текст команд, которые применялись при выполнении задания. При наличии: часть конфигурационных файлов, которые решают задачу. Скриншоты результата запуска контейнеров (веб-интерфейс).

Присылаем в формате текстового документа: задание и команды для решения (без вывода). Формат — PDF (один файл на все задания).

## Решение:

### 1. Устанавливаю Docker:

- `sudo apt update`
- `sudo apt install apt-transport-https ca-certificates curl software-properties-common`
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
- `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"`
- `sudo apt update`
- `apt-cache policy docker-ce`
- `sudo apt install docker-ce`
- `sudo systemctl status docker` — проверяю статус Docker.

### 2. Устанавливаю Docker compose:

- `sudo curl -L https://github.com/docker/compose/releases/download/1.29.2/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose`
- `sudo chmod +x /usr/local/bin/docker-compose`
- `docker-compose --version` — проверяю версию

### 3. Настраиваю конфигурацию веб сервера:

- `mkdir wordpress && cd wordpress` — создаю директорию wordpress и перехожу в нее.
- `mkdir nginx-conf` — создаю директорию для файла конфигурации
- `nano nginx-conf/nginx.conf` — открываю файл конфигурации
- Добавляю настройки в файл:

```
server {
    listen 80;
    listen [::]:80;

    server_name _;

    index index.php index.html index.htm;

    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;
}
}
```

Сохраняю

#### 4. Настраиваю переменные среды:

- `nano .env` — открываю файл `env`, добавляю в него и сохраняю:  
MYSQL\_ROOT\_PASSWORD=123456  
MYSQL\_USER=admin  
MYSQL\_PASSWORD=123456
- `nano .dockerignore` — открываю файл, добавляю в него и сохраняю:  
.env  
docker-compose.yml  
.dockerignore

### 5. Определяю службы с помощью Docker Compose:

- `nano docker-compose.yml` — открываю файл, добавляю в него и сохраняю:

```
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
  networks:
    - app-network
```

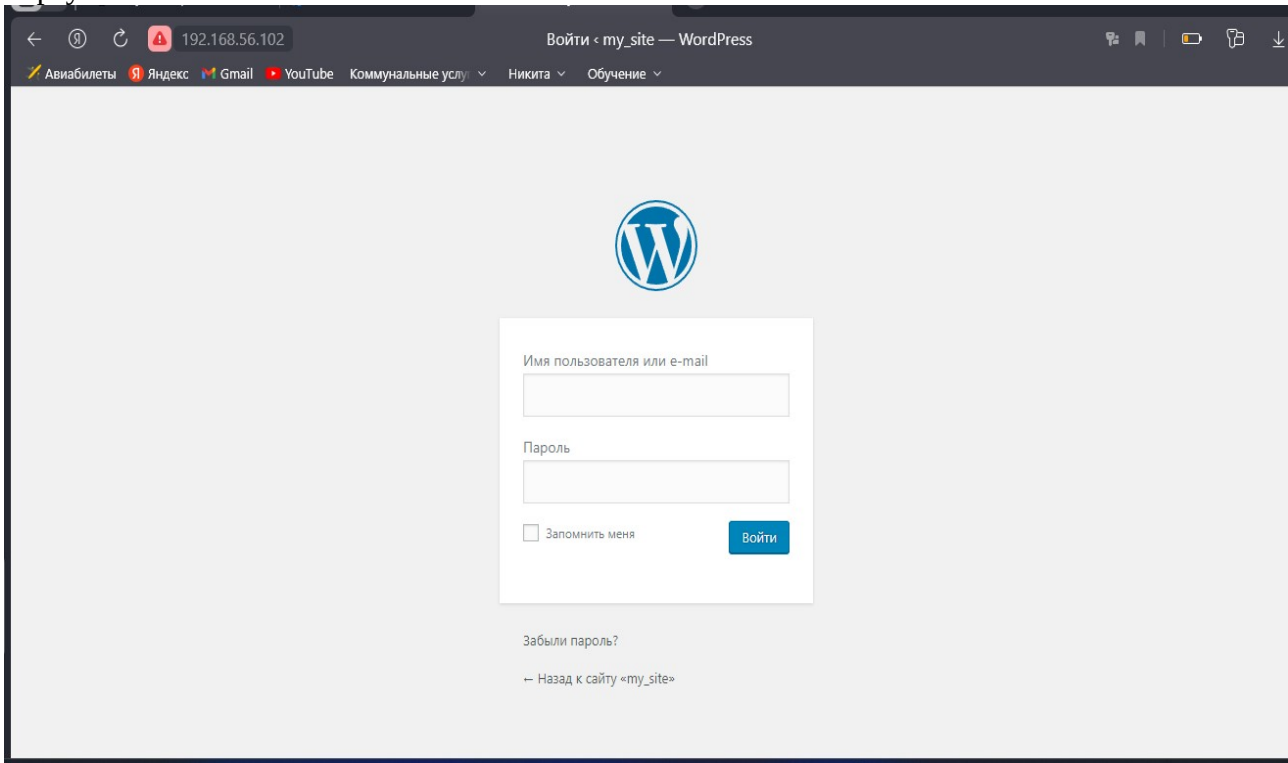
```
volumes:
  wordpress:
  dbdata:

networks:
  app-network:
    driver: bridge
```

6. Создаю контейнеры и проверяю работоспособность:

- `docker-compose up -d` — собираю.

В браузере на хостовой машине я могу подключиться по ip адресу виртуалки, а в виртуальной по localhost:



В моем случае я уже зарегистрировался и сделал первичную настройку wordpress.