

Если вы видите что-то необычное, просто сообщите мне.

Docker-compose ????.

Docker-compose - инструмент для совместного запуска нескольких контейнеров. Основой для docker-compose является yaml-файл с настройками. Удобство заключается в том, что единожды написанный docker-compose.yml можно легко перенести с устройства на устройство.

??????????

```
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
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

Третья строчка скрипта дает нам возможность использовать docker-compose от имени системы, если эта команда не будет выполнена. команда docker-compose потребует указания полноценного пути расположения.

??????? ?????????????? docker- compose

Пример взят с docs.microsoft.com

```
version: "3.7"

services:
  app:
    image: node:12-alpine
    #build: .
```

```
command: sh -c "yarn install && yarn run dev"
```

```
ports:
```

```
- 3000:3000
```

```
working_dir: /app
```

```
volumes:
```

```
- ./:/app
```

```
environment:
```

```
MYSQL_HOST: mysql
```

```
MYSQL_USER: root
```

```
MYSQL_PASSWORD: secret
```

```
MYSQL_DB: todos
```

```
networks:
```

```
app_net:
```

```
mysql:
```

```
image: mysql:5.7
```

```
volumes:
```

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

```
environment:
```

```
MYSQL_ROOT_PASSWORD: secret
```

```
MYSQL_DATABASE: todos
```

```
networks:
```

```
app_net:
```

```
volumes:
```

```
todo-mysql-data:
```

```
networks:
```

```
app_net:
```

```
driver: bridge
```

????????? ?????? ?????????????????? ? docker-compose:

Ключи	Применение
build	Указывает путь откуда брать Dockerfile для создания образа
args	Указание аргументов билда которые требуются во время создания образа

Ключи	Применение
command	Переписывает CMD указанную в Dockerfile
devices	Мапирование физических устройств в рабочий контейнер
depends_on	Указание зависимости сервисов друг от друга. Зависимый сервис будет запускаться последний
dns	Указанный в ручную DNS сервер
entrypoint	Переписывает ENTRYPOINT директиву из Dockerfile
env_file	Добавление переменных с помощью файла переменных окружения
environment	Явное указание переменных в контейнер
expose	Выставление портов контейнера, они будут доступны только <code>linked</code> сервисам, могут быть указаны только внутренние порты.
image	Указание образа, который будет взят для старта контейнера
links	Связывание контейнеров разных сервисах
logging	Настройки логирования для сервиса.
networks	Сеть к которой будет присодинен сервис
ports	Мапирование портов из контейнера в хост систему
volumes	Мапирование папок из хост системы в контейнер

Docker-compose cli

Сборка контейнера, имеет смысл если в `yaml` файле присутствует директива `build`. Можно использовать ключ `--no-cache`, в таком случае игнорируются кэшированные образы.

```
docker-compose build
```

Запуск приложения. В запуск можно добавить `--build` ключ, тогда docker-compose перебилдит незакешированные шаги.

```
docker-compose up
```

Docker-compose останавливает и удаляет запущенные контейнеры, согласно yaml конфигурации. То есть выполняется 2 команды docker: `docker stop` и `docker rm`

```
docker-compose stop
```

Отображает запущенные контейнеры принадлежащие yaml файлу.

```
docker-compose ps
```

“ ВНИМАНИЕ! Выполнение команда, в отличии от `docker`, должно быть с в папке с yaml файлом. Команда `docker` может выполняться из любой папки.

Revision #6

Created 2021-09-22 11:54:59 UTC by gasick

Updated 2023-11-08 16:10:47 UTC by gasick