

Homework_Lesson_22

Цель: получить практический опыт написания Dockerfile, развертывания приложений с использованием Docker-compose

Задание 1: Вам необходимо написать Dockerfile для создания контейнера с приложением веб-сервера на основе образа Ubuntu 20.04. Приложение должно быть запущено на порту 8080 и должно отдавать статические файлы из каталога /app/static.

Шаги, которые необходимо выполнить:

1. Создайте новый файл Dockerfile в пустой директории на вашем локальном компьютере.
 2. Напишите инструкцию FROM, которая указывает базовый образ Ubuntu 20.04.
 3. Установите необходимые зависимости с помощью инструкции RUN.
- Установите пакеты nginx и curl, а также создайте каталог /app/static.
4. Скопируйте файл конфигурации nginx из вашего локального каталога внутрь контейнера с помощью инструкции COPY.
 5. Скопируйте статические файлы из каталога /app/static на вашем локальном компьютере внутрь контейнера с помощью инструкции COPY.
 6. Используйте инструкцию EXPOSE для открытия порта 8080.
 7. Используйте инструкцию CMD для запуска команды nginx с указанием пути к файлу конфигурации, который вы скопировали на шаге 4.
 8. Сохраните файл Dockerfile и соберите образ с помощью команды docker build.
 9. Запустите контейнер из образа с помощью команды docker run и проверьте, что веб-сервер отдает статические файлы из каталога /app/static на порту 8080.

Подготовим Dockerfile и конфиг nginx

```
Dockerfile X
dz22 > Dockerfile > ...
1  # Шаг 2: Указываем базовый образ
2  FROM ubuntu:20.04
3
4  # Шаг 3: Устанавливаем зависимости
5  RUN apt-get update && \
6      apt-get install -y nginx curl && \
7      mkdir -p /app/static
8
9  # Шаг 4: Копируем конфигурацию Nginx
10 COPY nginx.conf /etc/nginx/nginx.conf
11
12 # Шаг 5: Копируем статические файлы
13 COPY ./static /app/static
14
15 # Шаг 6: Открываем порт 8080
16 EXPOSE 8080
17
18 # Шаг 7: Запускаем Nginx
19 CMD ["nginx", "-g", "daemon off;"]
```

```

dz22 > nginx.conf
1  # Главная секция для Nginx
2  events {}
3
4  # Основной блок для конфигурации http-сервера
5  http {
6      server {
7          listen 8080;
8          server_name localhost;
9
10         # Указываем корневую папку для статических файлов
11         root /app/static;
12
13         # Обрабатываем запросы на index.html
14         index index.html;
15
16         # Настроим путь для статических файлов
17         location / {
18             try_files $uri $uri/ =404;
19         }
20
21         # Логирование ошибок
22         error_log /var/log/nginx/error.log;
23         access_log /var/log/nginx/access.log;
24     }
25 }

```

Подготовим простой html файл

```

dz22 > static > index.html > ...
1  <!DOCTYPE html>
2  <html lang="ru">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Пример страницы</title>
7  </head>
8  <body>
9      <header>
10         <h1>Липиков Павел Добро пожаловать! </h1>
11     </header>
12
13     <main>
14         <section>
15             <h2>О нас</h2>
16         </section>
17
18         <section>
19             <h2>Раздел
20             <ul>
21                 <li>Подраздел</li>
22                 <li>Подраздел</li>
23                 <li>Подраздел</li>
24             </ul>
25         </section>
26     </main>
27
28     <footer>
29         <p>© 2025 Компания. Все права защищены.</p>
30     </footer>

```


Соберем Docker образ с нашим веб-сервером:

```
sttewie@sttewie-VMware-Virtual-Platform:~/dz22$ sudo docker build -t my-nginx-app .
[+] Building 1.1s (9/9) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 585B                             0.0s
=> [internal] load metadata for docker.io/library/ubuntu:20.04  1.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                    0.0s
=> [1/4] FROM docker.io/library/ubuntu:20.04@sha256:8e5c4f0285ecbb4ead070431d29b576a530d3166df73ec44affc1cd27555141b  0.0s
=> [internal] load build context                                0.0s
=> => transferring context: 823B                                  0.0s
=> CACHED [2/4] RUN apt-get update && apt-get install -y nginx curl && mkdir -p /app/static  0.0s
=> [3/4] COPY nginx.conf /etc/nginx/nginx.conf                 0.0s
=> [4/4] COPY ./static /app/static                             0.0s
=> exporting to image                                           0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:49a5f2bf7b5f3085d641810081f85aad34a6e00f1bc0a0d1939d73253b1f85bb  0.0s
=> => naming to docker.io/library/my-nginx-app                 0.0s
sttewie@sttewie-VMware-Virtual-Platform:~/dz22$
```

Запускаем контейнер:

```
sttewie@sttewie-VMware-Virtual-Platform:~/dz22$ sudo docker run -d -p 8080:8080 my-nginx-app
c00086df16df57fc5871d644936b6a61873408b0a57ff99939fbc071b6d35230
sttewie@sttewie-VMware-Virtual-Platform:~/dz22$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
c00086df16df   my-nginx-app   "nginx -g 'daemon of..." 26 seconds ago Up 25 seconds 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   silly_chatterjee
sttewie@sttewie-VMware-Virtual-Platform:~/dz22$
```

И проверяем его работу

← → ↻ http://localhost:8080/ 

Лапиков Павел Добро пожаловать!

О нас

Раздел

- Подраздел
- Подраздел
- Подраздел

© 2025 Компания. Все права защищены.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS DOCKER
sttewie@sttewie-VMware-Virtual-Platform:~/dz22$ sudo docker ps
[sudo] password for sttewie:
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
c00086df16df   my-nginx-app   "nginx -g 'daemon of..." 15 minutes ago Up 15 minutes 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   silly_chatterjee
sttewie@sttewie-VMware-Virtual-Platform:~/dz22$
```

Задание 2 – развертывание приложения с помощью Docker-compose

1. Создайте новый файл docker-compose.yml в пустой директории на вашем локальном компьютере.

2. Напишите инструкцию version в версии 3.

3. Определите сервис для базы данных PostgreSQL. Назовите его "db".

Используйте образ postgres:latest, задайте переменные окружения

POSTGRES_USER, POSTGRES_PASSWORD и POSTGRES_DB для установки пользовательского имени, пароля и имени базы данных соответственно.

4. Определите сервис для веб-сервера на основе образа NGINX. Назовите его "web". Используйте образ nginx:latest. Определите порт, на котором должен работать сервер, с помощью инструкции ports. Задайте путь к файлам конфигурации NGINX внутри контейнера, используя инструкцию volumes.

5.* Определите ссылку на сервис базы данных в сервисе веб-сервера.

Используйте инструкцию links.

6. Сохраните файл docker-compose.yml и запустите приложение с помощью команды docker-compose up.

7. Проверьте, что приложение работает, перейдя в браузере на localhost:80.

```

docker-compose.yml X
docker-compose.yml
1  version: '3'
2
3  >Run All Services
4  services:
5      >Run Service
6      db:
7          image: postgres:latest
8          container_name: my-postgres
9          environment:
10             POSTGRES_USER: user
11             POSTGRES_PASSWORD: password
12             POSTGRES_DB: mydatabase
13          volumes:
14             - pgdata:/var/lib/postgresql/data
15          restart: always
16
17      >Run Service
18      web:
19          image: nginx:latest
20          container_name: my-nginx
21          ports:
22             - "80:80"
23          volumes:
24             - ./nginx.conf:/etc/nginx/nginx.conf
25             - ./static:/app/static
26          depends_on:
27             - db
28          links:
29             - db
30
31  volumes:
32     pgdata:

```

```

nginx.conf X
nginx.conf
1  server {
2      listen 80;
3      server_name localhost;
4
5      # Устанавливаем корневую директорию для файлов
6      root /usr/share/nginx/html;
7
8      # Индексный файл
9      index index.html;
10
11     # Настройка обработки запросов
12     location / {
13         try_files $uri $uri/ =404;
14     }
15
16     # Логирование
17     access_log /var/log/nginx/access.log;
18     error_log /var/log/nginx/error.log;
19 }
20
21

```

```

sttewie@sttewie-VMware-Virtual-Platform:~/docker-compose$ sudo docker ps

```

```

[sudo] password for sttewie:

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a43b133edae5	postgres:latest	"docker-entrypoint.s..."	44 seconds ago	Up 44 seconds	5432/tcp	my-postgres

Что-то не получилось, не поднялся хост