

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

Dockerfile, создание docker image.

Dockerfile

Представим, приложение уже работает на вашей машине, но еще не имеет образа. Для того, чтобы получить docker образ, нам необходимо описать его.

Создадим инструкцию для сборки образа

В папке с проектом(мы рассматриваем node проект) создадим файлы:

```
touch Dockerfile
```

Для того, чтобы в докер не попали ненужные файлы - рядом создадим `dockerignore` и впишем в него всё, что нам не нужно.

```
touch .dockerignore
```

Пример Dockerfile:

```
#Выберем базовый образ из которого мы будем создавать наш проект
FROM node:10-apline

# Укажем папку внутри докера, которая будет являться домашней при выполнении различных команд
```

```
WORKDIR /usr/src/app
```

```
# Скопируем из папки с проектом все файлы по маске package*.json в папку WORKDIR
```

```
COPY package*.json ./
```

```
# Запускаем установку пакетов npm
```

```
RUN npm install
```

```
# Копируем всё что осталось в папке с проектом в папку WORKDIR
```

```
COPY . .
```

```
# Указываем порт, через который будет доступен наш проект
```

```
EXPOSE 3000
```

```
# Указываем команду которая будет работать в тот момент, когда запустится контейнер
```

```
CMD["npm", "start"]
```

Список ключей используемых при создании образа

Ключ	Назначение
------	------------

FRO M	Ука зыв аем на осн ове кот оро го из обр азо в буд ем соз дав ать нов ый обр аз
MAI NTA INE R	Ука зыв аем авт ора соз дан ног о обр аза

RUN	Зап ука ем ком анд ы в вну три кон тей нер а нео бхд имы е для раб оты обр аза
-----	--

СМ D	Ком анд а кот ора я буд ет вып олн ена при зап уск е кон тей нер а(м оже т быт ь тол ько одн а)
---------	---

EXP OSE	Гов ори т док еру, что кон тей нер слу шае т на опр еде лен ном пор ту
ENV	Ука зыв аем с как им пер еме нны м окр уже ние м нео бхо дим о соз дав ать кон тей нер

ADD	Копирует файлы, папки, URL и добавляет их в файловую систему образа, может распаковать архив
COPY	Копирует файлы, папки из и по указанному пути

ENT RYP OIN T	Поз вол яет зад ава ть ком анд у зап уск а кон тей нер а, при это м при ста рте ука зыв ать клю чи зап уск а этог о кон тей нер а, пер еоп ред еля ет СМ D
------------------------	--

USE R	Ука зывает имя пользо вателя под кот орым буд ет про исхо дить вып олне ние ком анд RUN , CMD, ENTER POINT Т вну три кон тей нер а
----------	--

WO RKD IR	Ука зыв аем дир ект ори ю вну три кот оро й буд ет про исх оди ть вып олн ени е ком анд RUN , CM D, ENT RYP OIN T, COP Y, ADD
-----------------	--

ARG	Определением переменных, которые могут быть переданы билдеру при запуске <code>docker build</code> используется ключ <code>--build-arg</code> <code><variable>=<value></code>
-----	---

Билдим проект

Создаем билд с помощью команды:

```
docker build . -t firstimage
```

Ожидаем завершения команды. Для этого может потребоваться много времени.

В случае, если билд завершился неудачно, смотрим что за ошибка и правим Dockerfile

После завершения: мы можем посмотреть на наш `firstimage` в списке, который можно получить с помощью команды:

```
docker image
```

Запуск нашего проекта

Чтобы запустить наш проект - необходимо указать несколько параметров:

```
docker run -p 80:3000 firstimage
```

-p - говорит о том, что мы мапируем(прим. ред. mapping - сопоставление) системный 80 порт в порт(помните внутри контейнера использовали EXPOSE 3000) внутри контейнера.

Заходим на <http://localhost> и убеждаемся, что наше приложение доступно внутри контейнера.

Revision #4

Created 22 September 2021 11:53:26 by gasick

Updated 8 November 2023 16:10:47 by ivan.serov37@gmail.com