**Клюкина В, ПУОР22-2м**

**Задание на самостоятельную работу с контейнером**

В этом задании вы будете собирать собственный контейнер с веб-сервером и запустив его должны попасть на свой сайт.

Таким образом задание предполагает работу с отдельным контейнером (К), докерфайлом (Д) и хостовым линуксом (Х).

**Сдача отчетности**

1. Вам необходимо написать Dockerfile и создать в этой же директории согласно заданию необходимые файлы. Полученные файлы необходимо положить в свой git-репозиторий.
2. Собранный образ необходимо загрузить в докер-репозиторий, для этого следует зарегистрироваться на hub.docker.com, а затем предоставить ссылку на него и git-репозиторий преподавателю.

**Задание**

***Примечание:*** *некоторые команды могут иметь другой синтаксис в зависимости от выбранного базового образа*

*Выполняете оптимизацию образа за счет применения легковесного базового образа, уменьшения количества слоев и др.*

1. Выберите базовый образ (Д)

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

1. Выполнить обновление apt-кеша (Д)
2. Обновить все пакеты в контейнере (Д)
3. Установить веб сервер nginx (Д)
4. Очистить скачанный apt-cache (Д)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Удалить содержимое директории /var/www/ (Д)
2. Создать в директории /var/www/ директорию с именем вашего сайта и папку с картинками (my\_project/img) (Д)

Изображение выглядит как текст, снимок экрана, Цвет электрик, Шрифт

Автоматически созданное описание

1. Поместить из папки с докер файлом в директорию /var/www/ my\_project файл index.html. Файл нужно создать рядом с Dockerfile и поместить туда какую-нибудь строку приветствия (Х,Д)
2. Поместить из папки с докер файлом в директорию /var/www/my\_project/img файл img.jpg (поместить туда какую-нибудь картинку). (Х,Д)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Задать рекурсивно на папку /var/www/my\_project права "владельцу - читать, писать, исполнять; группе - читать, исполнять, остальным - читать, исполнять" (Д)
2. С помощью команды useradd создать пользователя <ваше имя> (Д)
3. С помощью команды groupadd создать группу <ваше имя> (Д)
4. С помощью команды usermod поместить пользователя <ваше имя> в группу <ваше имя> (Д)
5. Рекурсивно присвоить созданных пользователя и группу на папку /var/www/my\_project (Д)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Воспользоваться конструкцией (sed -i 's/ЧТО ЗАМЕНИТЬ/НА ЧТО ЗАМЕНИТЬ/g' имя\_файла) и заменить в файле /etc/nginx/sites-enabled/default следующую подстроку (/var/www/html) так, чтобы она соответствовала (/var/www/my\_project) (Д)
2. С помощью команды grep найти в каком файле задается пользователь (user), от которого запускается nginx (К)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. С помощью команды sed проделать операцию замены пользователя в файле, найденном в пункте 17 на вашего пользователя (Д)

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание





Формируем Dockerfile:

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

Итого:

FROM nginx:stable-alpine3.17-slim

RUN command apk update \

&& apk upgrade \

&& rm -rf /var/cache/apk/\* \

&& mkdir /var/www \

&& mkdir /var/www/my\_project/ \

&& mkdir /var/www/my\_project/img

COPY index.html /var/www/my\_project/

COPY img.jpg /var/www/my\_project/img/img.jpg

RUN chmod -R 755 /var/www/my\_project \

&& addgroup -S lera \

&& adduser -S lera -G lera \

&& chown -R lera:lera /var/www/my\_project

RUN sed -i '2s/nginx/lera lera/' /etc/nginx/nginx.conf \

&& sed -i '8s/usr\/share\/nginx\/html/var\/www\/my\_project/' /etc/nginx/conf.d/default.conf

EXPOSE 80

CMD ["nginx","-g","daemon off;"]

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

1. Соберите ваш контейнер: docker build -t test . (Х)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Проведите тест nginx командой nginx -t (К)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*nginx: the configuration file /etc/nginx/nginx.conf syntax is ok*

*nginx: configuration file /etc/nginx/nginx.conf test is successful*

1. Определите порт подключения. (Д)

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

1. Задайте в команды запуска веб-сервера. (Д)
2. Запустить ваш контейнер, но на этот раз передав аргументами нужные порты: docker run -ti -p 80:80 test

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Откройте в браузере 127.0.0.1:80. Вы должны получить вашу приветственную страницу. Сделайте скриншот страницы в браузере на фоне команды запуска контейнера. Скриншот положить в репозиторий (Х)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

1. Выполните оптимизацию образа за счет применения легковесного базового образа, уменьшения количества слоев и др.(Д) – выбран изначально
2. Выполните правильную постановку тега и загрузите ваш контейнер в docker hub. (Х)