

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра теоретической и прикладной информатики

Лабораторная работа №1 по дисциплине «Разработка web-приложений и распределенных информационных систем»

Факультет:	ПМИ
Группа:	ПМ-92
Студент:	Иванов В. В.
Преподаватель:	Цыгулин А. А.

Новосибирск

2022

1. Задание

Создать Docker-контейнер и разместить его в репозитории Docker Hub. При загрузке и запуске контейнер должен работать как http-сервер и на заданном порту предоставлять доступ к странице сайта.

2. Ход работы

Для создания Docker-контейнера создаём каталог, в который помещаем три файла:

index.html

```
<html>
  <head>
    <title>lab1</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

server.js

```
var express = require('express');
var app = express();

app.get('/', function(req, res){
  res.sendFile('index.html')
})

app.listen(80);

console.log('Start Web Server');
```

Dockerfile

1. FROM debian:9.3-slim
2. RUN rm /bin/sh && ln -s /bin/bash /bin/sh
3. RUN apt-get update && apt-get install -y curl
4. ENV NVM_DIR /usr/local/nvm
5. ENV NODE_VERSION 8.9.4
6. RUN curl --silent -o- https://raw.githubusercontent.com/creationix/nvm/v0.31.2/install.sh | bash

```
7. RUN source ~/.bashrc && nvm install $NODE_VERSION && nvm alias default $NODE_VERSION && nvm use default
8. ENV NODE_PATH $NVM_DIR/v$NODE_VERSION/lib/node_modules
9. ENV PATH $NVM_DIR/versions/node/v$NODE_VERSION/bin:$PATH
10. RUN node -v
11. RUN npm -v
12. RUN npm install express
13. COPY . .
14. EXPOSE 80
15. CMD node server.js
```

Сервер работает на фреймворке **express**. Приложение слушает соединения на порте **80**, и на все запросы, адресованные корневому каталогу, выдает файл **index.html**.

В **Dockerfile** описываются инструкции для сборки контейнера. В строке 1 указывается образ, с которого наш образ будет унаследован. В строках 2-3 настраивается оболочка, обновляются пакеты и устанавливается **curl**. В строках 4-5 настраиваются переменные среды. В строках 6-9 устанавливаются **nvm** и **node**, а затем настраиваются используемые **node** переменные. В строках 12-13 устанавливается **express**, после чего содержимое каталога помещается в корень ОС. Затем настраивается порт и команда, которая будет выполнена при запуске контейнера.

Собираем контейнер в текущей директории:

```
sudo docker build -t webserver .
```

Запускаем контейнер:

```
sudo docker run -p 80:80 webserver
```

Переходим на localhost:80:



Теперь загрузим контейнер на Docker Hub. Для этого регистрируемся на сайте и выполняем авторизацию через терминал:

```
~/Docker/lab1 ➤ sudo docker login
[sudo] password for lenferdetroud:
Login with your Docker ID to push and pull images from Docker Hub. If you don't have
Username: lenferdetroud
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
~/Docker/lab1 ➤
```

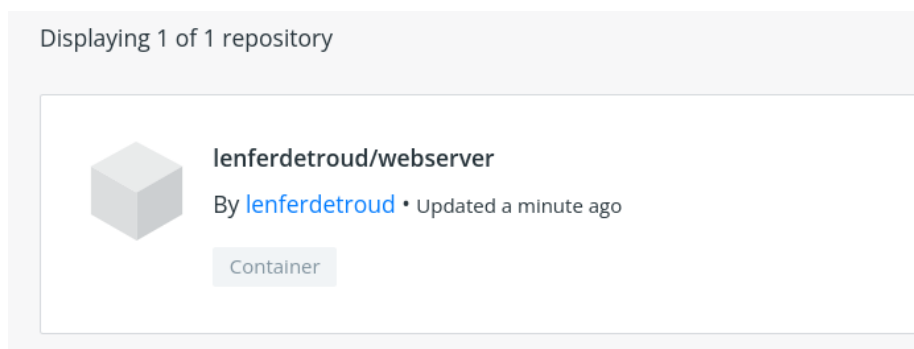
Аналогично собираем контейнер:

```
sudo docker build -t lenferdetroud/webserver .
```

Загружаем на Docker Hub:

```
~/Docker/lab1 ➤ sudo docker push lenferdetroud/webserver:latest
The push refers to repository [docker.io/lenferdetroud/webserver]
b332d5408aaf: Pushed
40db9c4d827b: Pushed
64aba420886e: Pushed
cac30d463859: Pushed
bdfb7fffec07: Pushed
17885b0acfad: Pushed
cbf4d3a36685: Pushed
014cf8bfcfb2d: Mounted from library/debian
latest: digest: sha256:807d2d1c793dc8bea0eedc75c081f09e9a03c6258e453bd391aec3c478b1d074 size: 1994
~/Docker/lab1 ➤
```

Контейнер появился в профиле:



Теперь для получения контейнера другому пользователю достаточно выполнить команду:

```
docker pull lenferdetroud/webserver
```

После чего контейнер можно запустить при помощи **docker run**.