|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
|  | Институт информационных технологий (ИТ) |
|  | Кафедра инструментального и прикладного программного обеспечения (ИиППО) |

|  |  |  |  |
| --- | --- | --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №7** | | | |
| **по дисциплине** | | | |
| **«Настройка и администрирование сервисного программного обеспечения» на тему**  **«Создание контейнера с веб сервером Nginx в Docker»**  **Вариант 20** | | | |
| Выполнил студент группы ИКБО-20-19 | | Московка А.А. | |
|  | |  | |
| Принял  *Старший преподаватель* | | Матчин В.Т. | |
| Практические работы выполнены | «13» апреля 2021 г. | | Московка А.А.  (подпись студента) | |
| «Зачтено» | «14» апреля 2021 г. | | Матчин В.Т.  (подпись руководителя) | |
|  |  | |  | |

Москва 2021

**Оглавление**

[Практическая работа №7: «Создание контейнера с веб сервером Nginx в Docker» 3](#_Toc69213484)

[Практическое задание №1 3](#_Toc69213485)

[Практическое задание №2 9](#_Toc69213486)

[Практическое задание №3 16](#_Toc69213487)

[Практическое задание №4 21](#_Toc69213488)

[Практическое задание №5 27](#_Toc69213489)

# Практическая работа №7: «Создание контейнера с веб сервером Nginx в Docker»

**Цель работы:** получить навыки сборки Docker-контейнера с использованием Docker-composer.

**Выполнение работы**

*server/index.html (не изменяется на протяжении всей работы)*

=DOCKER-COMPOSE READY=

## Практическое задание №1

**Задание:** повторить пример, который был рассмотрен в разделе «Общие сведения». Проект назвать docker\_py.

**Выполнение задания:**

***client.py***

import urllib.request

fp = urllib.request.urlopen("http://localhost:1234/")

encodedContent = fp.read()

decodedContent = encodedContent.decode("utf8")

print(decodedContent)

fp.close()

***client/Dockerfile***

FROM python:latest

ADD client.py /client/

WORKDIR /client/

***server.py***

import http.server

import socketserver

handler = http.server.SimpleHTTPRequestHandler

with socketserver.TCPServer(("", 1234), handler) as httpd:

httpd.serve\_forever()

***server/Dockerfile***

FROM python:latest

ADD server.py /server/

ADD index.html /server/

WORKDIR /server/

***docker-compose.yml***

version: "3.8"

services:

server:

build: server/

command: python ./server.py

ports:

- 1234:1234

client:

build: client/

command: python ./client.py

network\_mode: host

depends\_on:

- server

***docker-compose build***

*PS C:\Users\frayz\IdeaProjects\root> docker-compose build*

*Building server*

*failed to get console mode for stdout: The handle is invalid.*

*[+] Building 36.1s (10/10) FINISHED*

*=> [internal] load build definition from Dockerfile 0.1s*

*=> => transferring dockerfile: 1.54kB 0.0s*

*=> [internal] load .dockerignore 0.0s*

*=> => transferring context: 2B 0.0s*

*=> [internal] load metadata for docker.io/library/python:latest 2.3s*

*=> [auth] library/python:pull token for registry-1.docker.io 0.0s*

*=> [1/4] FROM docker.io/library/python:latest@sha256:8bd2e361ad8575ae80 32.5s*

*=> => resolve docker.io/library/python:latest@sha256:8bd2e361ad8575ae80a 0.0s*

*=> => sha256:8bd2e361ad8575ae80a6a3e556a524d44421cb5fa6b 2.36kB / 2.36kB 0.0s*

*=> => sha256:b1aa63f57d3c525bf3061f9afec537997039a31b118 8.33kB / 8.33kB 0.0s*

*=> => sha256:8bf9c589d5f9475f1fcc050e02308d6b4eeb86eab 50.40MB / 50.40MB 8.5s*

*=> => sha256:ea848ad42f0d1676b3a0ce709208ba3fa154cd533 10.00MB / 10.00MB 2.3s*

*=> => sha256:1e3b89f69bb6ada672153256bd88d74ae60571f6755 2.22kB / 2.22kB 0.0s*

*=> => sha256:4c70e46d8b5f0e1d77797b7a0565b7a316bd8c7c024 7.83MB / 7.83MB 1.7s*

*=> => sha256:48fe137f8d2641315dd2f93c46dfdb671ce7399e 51.84MB / 51.84MB 11.4s*

*=> => sha256:4b13f6ed9b0cab6a0e5a875bd6ecb04c0b0bc1 192.34MB / 192.34MB 22.3s*

*=> => sha256:ba85279f50e08583b48ba123c12b2618296d16a877 6.15MB / 6.15MB 11.0s*

*=> => extracting sha256:8bf9c589d5f9475f1fcc050e02308d6b4eeb86eab7752ef9 5.3s*

*=> => sha256:59a18d8c368051ee692493c36203a73dca768ce3 19.15MB / 19.15MB 13.8s*

*=> => sha256:c610993f70c6a49b50bf6a97bcc98e2e224986ca717277 231B / 231B 11.7s*

*=> => sha256:a9afc028cd665f38b486bb73669a8ddbca7e328947 2.16MB / 2.16MB 12.4s*

*=> => extracting sha256:4c70e46d8b5f0e1d77797b7a0565b7a316bd8c7c024f5cca 0.6s*

*=> => extracting sha256:ea848ad42f0d1676b3a0ce709208ba3fa154cd53370c50e7 0.6s*

*=> => extracting sha256:48fe137f8d2641315dd2f93c46dfdb671ce7399edc1f44f1 5.8s*

*=> => extracting sha256:4b13f6ed9b0cab6a0e5a875bd6ecb04c0b0bc1cd98a1600f 8.1s*

*=> => extracting sha256:ba85279f50e08583b48ba123c12b2618296d16a877a1881a 0.3s*

*=> => extracting sha256:59a18d8c368051ee692493c36203a73dca768ce3ea28a400 0.8s*

*=> => extracting sha256:c610993f70c6a49b50bf6a97bcc98e2e224986ca7172772c 0.0s*

*=> => extracting sha256:a9afc028cd665f38b486bb73669a8ddbca7e328947a9a4c7 0.2s*

*=> [internal] load build context 0.0s*

*=> => transferring context: 1.00kB 0.0s*

*=> [2/4] ADD server.py /server/ 0.9s*

*=> [3/4] ADD index.html /server/ 0.0s*

*=> [4/4] WORKDIR /server/ 0.0s*

*=> exporting to image 0.1s*

*=> => exporting layers 0.0s*

*=> => writing image sha256:50d0ddd5acc268d4871c572fc111fbc0c20146a1dee74 0.0s*

*=> => naming to docker.io/library/docker\_py\_server 0.0s*

*Successfully built 50d0ddd5acc268d4871c572fc111fbc0c20146a1dee742c4634dee0b1bb01744*

*Building client*

*failed to get console mode for stdout: The handle is invalid.*

*[+] Building 2.3s (8/8) FINISHED*

*=> [internal] load build definition from Dockerfile 0.0s*

*=> => transferring dockerfile: 332B 0.0s*

*=> [internal] load .dockerignore 0.0s*

*=> => transferring context: 2B 0.0s*

*=> [internal] load metadata for docker.io/library/python:latest 0.7s*

*=> CACHED [1/3] FROM docker.io/library/python:latest@sha256:8bd2e361ad85 0.0s*

*=> [internal] load build context 0.0s*

*=> => transferring context: 1.22kB 0.0s*

*=> [2/3] ADD client.py /client/ 0.0s*

*=> [3/3] WORKDIR /client/ 0.7s*

*=> exporting to image 0.7s*

*=> => exporting layers 0.7s*

*=> => writing image sha256:0d645f9d24764736e73c78220d92fcc9ccfeff1d1fc7a 0.0s*

*=> => naming to docker.io/library/docker\_py\_client 0.0s*

*Successfully built 0d645f9d24764736e73c78220d92fcc9ccfeff1d1fc7a12cb2ad9b24ac60d283*

***docker-compose up***

*PS C:\Users\frayz\IdeaProjects\root> docker-compose up*

*Recreating docker\_py\_server\_1 ... done*

*Recreating docker\_py\_client\_1 ... done*

*Attaching to docker\_py\_server\_1, docker\_py\_client\_1*

*server\_1 | 172.19.0.1 - - [29/Mar/2021 21:06:14] "GET / HTTP/1.1" 200 -*

*client\_1 | =DOCKER-COMPOSE READY=*

*docker\_py\_client\_1 exited with code 0*

## Практическое задание №2

Создать аналогичный сервис как в примере, рассмотренном в разделе «Общие сведения», но вместо языка Python использовать язык JavaScript. Проект назвать docker\_js.

**Выполнение задания:**

***client.js***

*const request = require('request');*

*request('http://localhost:1234/', function (error, response, body) {*

*console.log(body);*

*});*

***client/Dockerfile***

*FROM node:latest*

*ADD client.js /client/*

*WORKDIR /client/*

*RUN npm init --yes && npm install request --save*

***server.js***

*var http = require('http'),*

*fs = require('fs');*

*fs.readFile('./index.html', function (err, html) {*

*if (err) {*

*throw err;*

*}*

*http.createServer(function(request, response) {*

*response.writeHeader(200, {"Content-Type": "text/html"});*

*response.write(html);*

*response.end();*

*}).listen(1234);*

*});*

***server/Dockerfile***

*FROM node:latest*

*ADD server.js /server/*

*ADD index.html /server/*

*WORKDIR /server/*

***docker-compose.yml***

*version: "3.8"*

*services:*

*server:*

*build: server/*

*command: node ./server.js*

*ports:*

*- 1234:1234*

*client:*

*build: client/*

*command: node ./client.js*

*network\_mode: host*

*depends\_on:*

*- server*

***docker-compose build***

*PS C:\Users\frayz\IdeaProjects\root> docker-compose build*

*Building server*

*failed to get console mode for stdout: The handle is invalid.*

*[+] Building 34.4s (10/10) FINISHED*

*=> [internal] load build definition from Dockerfile 0.0s*

*=> => transferring dockerfile: 122B 0.0s*

*=> [internal] load .dockerignore 0.0s*

*=> => transferring context: 2B 0.0s*

*=> [internal] load metadata for docker.io/library/node:latest 2.4s*

*=> [auth] library/node:pull token for registry-1.docker.io 0.0s*

*=> [internal] load build context 0.0s*

*=> => transferring context: 446B 0.0s*

*=> [1/4] FROM docker.io/library/node:latest@sha256:35b326a83fc2d8d3f34b 31.3s*

*=> => resolve docker.io/library/node:latest@sha256:35b326a83fc2d8d3f34b2 0.0s*

*=> => sha256:d66c11f62793ec5d00678a387b0279781b6b84190e7 2.21kB / 2.21kB 0.0s*

*=> => sha256:1dfe7e1e1bffb84b5330514880896199259b01ebe 45.38MB / 45.38MB 6.4s*

*=> => sha256:df86b11a61e4b8cbb37a4a570baf51ae6bbc8cbf2 11.29MB / 11.29MB 2.3s*

*=> => sha256:006eb4a69e602dcb61f4680d7e4a0d4853902062ab1 4.34MB / 4.34MB 1.3s*

*=> => sha256:35b326a83fc2d8d3f34b2b72c62e25eed19a3a83c501650 776B / 776B 0.0s*

*=> => sha256:2530199d91c24f9d1843db28709e9d2d47b0ef59d9b 7.83kB / 7.83kB 0.0s*

*=> => sha256:6c1c07de8af095e25b339c838c0a02c16fda79738 49.79MB / 49.79MB 9.8s*

*=> => sha256:1a51928b48bb8cc9f8c0a8fbde38cae9366136 214.35MB / 214.35MB 20.3s*

*=> => sha256:b5f37d5733917b359da3aa160ce11fec312603d30b8 4.19kB / 4.19kB 6.6s*

*=> => extracting sha256:1dfe7e1e1bffb84b5330514880896199259b01ebe2b9d531 3.9s*

*=> => sha256:b7dc29d3a908f8bd56e4fbfa1d72818bd5128232 33.74MB / 33.74MB 12.0s*

*=> => sha256:95a7126f7e3379374de2bfb08df9cde51602ead837 2.38MB / 2.38MB 10.3s*

*=> => sha256:3e906ddd6c299b9444e06f587688293aceaf4669baadf0 294B / 294B 10.6s*

*=> => extracting sha256:df86b11a61e4b8cbb37a4a570baf51ae6bbc8cbf231f5640 0.8s*

*=> => extracting sha256:006eb4a69e602dcb61f4680d7e4a0d4853902062ab123026 0.3s*

*=> => extracting sha256:6c1c07de8af095e25b339c838c0a02c16fda79738e6c03e2 4.7s*

*=> => extracting sha256:1a51928b48bb8cc9f8c0a8fbde38cae9366136a64f8eb27c 8.7s*

*=> => extracting sha256:b5f37d5733917b359da3aa160ce11fec312603d30b8b42a1 0.0s*

*=> => extracting sha256:b7dc29d3a908f8bd56e4fbfa1d72818bd51282329a83d126 1.6s*

*=> => extracting sha256:95a7126f7e3379374de2bfb08df9cde51602ead837876a76 0.1s*

*=> => extracting sha256:3e906ddd6c299b9444e06f587688293aceaf4669baadf0f8 0.0s*

*=> [2/4] ADD server.js /server/ 0.4s*

*=> [3/4] ADD index.html /server/ 0.0s*

*=> [4/4] WORKDIR /server/ 0.0s*

*=> exporting to image 0.1s*

*=> => exporting layers 0.0s*

*=> => writing image sha256:73da5c8798b11234ee9fbc0e65c3813ea995ffa47d585 0.0s*

*=> => naming to docker.io/library/docker\_js\_server 0.0s*

*Successfully built 73da5c8798b11234ee9fbc0e65c3813ea995ffa47d585a4e8374a6c5ec9f2d8a*

*Building client*

*failed to get console mode for stdout: The handle is invalid.*

*[+] Building 1.0s (8/8) FINISHED*

*=> [internal] load build definition from Dockerfile 0.0s*

*=> => transferring dockerfile: 97B 0.0s*

*=> [internal] load .dockerignore 0.0s*

*=> => transferring context: 2B 0.0s*

*=> [internal] load metadata for docker.io/library/node:latest 0.7s*

*=> [internal] load build context 0.0s*

*=> => transferring context: 458B 0.0s*

*=> CACHED [1/3] FROM docker.io/library/node:latest@sha256:35b326a83fc2d8 0.0s*

*=> [2/3] ADD client.js /client/ 0.0s*

*=> [3/3] WORKDIR /client/ 0.0s*

*=> [4/4] RUN npm init --yes && npm install request --save 3.2s*

*=> exporting to image 0.1s*

*=> => exporting layers 0.0s*

*=> => writing image sha256:ece2f93f27685ec3cf22c79a26b4c7ed7157bb24ce4bd 0.0s*

*=> => naming to docker.io/library/docker\_js\_client 0.0s*

*Successfully built ece2f93f27685ec3cf22c79a26b4c7ed7157bb24ce4bd4eedd7e1768017266f9*

***docker-compose up***

*# docker-compose up*

*Recreating docker\_js\_server\_1 ... done*

*Recreating docker\_js\_client\_1 ... done*

*Attaching to docker\_js\_server\_1, docker\_js\_client\_1*

*client\_1 | =DOCKER-COMPOSE READY=*

*docker\_js\_client\_1 exited with code 0*

## Практическое задание №3

**Задание:** создать сервис как в примере, рассмотренном в разделе «Общие сведения», но вместо сервера на Python использовать веб сервер Apache. Проект назвать docker\_apache.

**Выполнение задания:**

***client.py***

*import urllib.request*

*fp = urllib.request.urlopen("http://localhost:1234/")*

*encodedContent = fp.read()*

*decodedContent = encodedContent.decode("utf8")*

*print(decodedContent)*

*fp.close()*

***client/Dockerfile***

*FROM python:latest*

*ADD client.py /client/*

*WORKDIR /client/*

***server/Dockerfile***

*FROM httpd:latest*

*COPY index.html /usr/local/apache2/htdocs/*

*WORKDIR /server/*

***docker-compose.yml***

*version: "3.8"*

*services:*

*server:*

*build: server/*

*ports:*

*- 1234:80*

*client:*

*build: client/*

*command: python ./client.py*

*network\_mode: host*

*depends\_on:*

*- server*

***docker-compose build***

*PS C:\Users\frayz\IdeaProjects\root> docker-compose build*

*Building server*

*failed to get console mode for stdout: The handle is invalid.*

*[+] Building 5.7s (9/9) FINISHED*

*=> [internal] load build definition from Dockerfile 0.0s*

*=> => transferring dockerfile: 116B 0.0s*

*=> [internal] load .dockerignore 0.0s*

*=> => transferring context: 2B 0.0s*

*=> [internal] load metadata for docker.io/library/httpd:latest 2.3s*

*=> [auth] library/httpd:pull token for registry-1.docker.io 0.0s*

*=> [internal] load build context 0.0s*

*=> => transferring context: 31B 0.0s*

*=> [1/3] FROM docker.io/library/httpd:latest@sha256:3769da9c8d8207503611 3.0s*

*=> => resolve docker.io/library/httpd:latest@sha256:3769da9c8d8207503611 0.0s*

*=> => sha256:375e8fe6c4f65ab5e30a9ce89c4692ae09004663c 24.48MB / 24.48MB 1.9s*

*=> => sha256:3769da9c8d8207503611bf86b60daf3754d84dcad2e 1.86kB / 1.86kB 0.0s*

*=> => sha256:6a1c6f11135e9f65d9092b639eb04c96b1c74f1fb86 1.37kB / 1.37kB 0.0s*

*=> => sha256:4ede4372e89be2def611d4e8164570aaa4ab1b7dee8 8.71kB / 8.71kB 0.0s*

*=> => sha256:ea331039465cfbf59234136a425f85fe7d5d06593f4805a 176B / 176B 0.2s*

*=> => sha256:fa84f6b2362267adee26638a1ffeda38dd32f8059ed 2.79MB / 2.79MB 0.9s*

*=> => extracting sha256:ea331039465cfbf59234136a425f85fe7d5d06593f4805a4 0.0s*

*=> => sha256:bba002695c79c816c7d601b841ec3e1884ed306e6340a54 297B / 297B 0.7s*

*=> => extracting sha256:fa84f6b2362267adee26638a1ffeda38dd32f8059ed3410f 0.3s*

*=> => extracting sha256:375e8fe6c4f65ab5e30a9ce89c4692ae09004663c0216308 0.9s*

*=> => extracting sha256:bba002695c79c816c7d601b841ec3e1884ed306e6340a54c 0.0s*

*=> [2/3] COPY index.html /usr/local/apache2/htdocs/ 0.2s*

*=> [3/3] WORKDIR /server/ 0.0s*

*=> exporting to image 0.0s*

*=> => exporting layers 0.0s*

*=> => writing image sha256:4107b9c2c5148aed06c90e3a506bf675125e7b42e49aa 0.0s*

*=> => naming to docker.io/library/docker\_apache\_server 0.0s*

*Successfully built 4107b9c2c5148aed06c90e3a506bf675125e7b42e49aa9a383560edd8c3f532f*

*Building client*

*failed to get console mode for stdout: The handle is invalid.*

*[+] Building 1.7s (9/9) FINISHED*

*=> [internal] load build definition from Dockerfile 0.0s*

*=> => transferring dockerfile: 99B 0.0s*

*=> [internal] load .dockerignore 0.0s*

*=> => transferring context: 2B 0.0s*

*=> [internal] load metadata for docker.io/library/python:latest 1.5s*

*=> [auth] library/python:pull token for registry-1.docker.io 0.0s*

*=> [internal] load build context 0.0s*

*=> => transferring context: 235B 0.0s*

*=> [1/3] FROM docker.io/library/python:latest@sha256:8bd2e361ad8575ae80a 0.0s*

*=> CACHED [2/3] ADD client.py /client/ 0.0s*

*=> CACHED [3/3] WORKDIR /client/ 0.0s*

*=> exporting to image 0.0s*

*=> => exporting layers 0.0s*

*=> => writing image sha256:ad0b807ce9ef6bcb884e3a906889f7c3aca639c3bc788 0.0s*

*=> => naming to docker.io/library/docker\_apache\_client 0.0s*

*Successfully built ad0b807ce9ef6bcb884e3a906889f7c3aca639c3bc788375313035587f35b1b9*

***docker-compose up***

*PS C:\Users\frayz\IdeaProjects\root>docker-compose up*

*Creating network "docker\_apache\_default" with the default driver*

*Creating docker\_apache\_server\_1 ... done*

*Creating docker\_apache\_client\_1 ... done*

*Attaching to docker\_apache\_server\_1, docker\_apache\_client\_1*

*server\_1 | AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.22.0.2. Set the 'ServerName' directive globally to suppress this message*

*server\_1 | AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.22.0.2. Set the 'ServerName' directive globally to suppress this message*

*server\_1 | [Mon Mar 29 22:32:56.520209 2021] [mpm\_event:notice] [pid 1:tid 140715728663680] AH00489: Apache/2.4.46 (Unix) configured -- resuming normal operations*

*server\_1 | [Mon Mar 29 22:32:56.520599 2021] [core:notice] [pid 1:tid 140715728663680] AH00094: Command line: 'httpd -D FOREGROUND'*

*client\_1 | =DOCKER-COMPOSE READY=*

*server\_1 | 172.22.0.1 - - [29/Mar/2021:22:32:56 +0000] "GET / HTTP/1.1" 200 22*

*docker\_apache\_client\_1 exited with code 0*

## Практическое задание №4

**Задание:** создать сервис как в примере, рассмотренном в разделе «Общие сведения», но вместо сервера на Python использовать веб сервер Nginx. Проект назвать docker\_nginx.

***client.py***

*import urllib.request*

*fp = urllib.request.urlopen("http://localhost:1234/")*

*encodedContent = fp.read()*

*decodedContent = encodedContent.decode("utf8")*

*print(decodedContent*

*fp.close()*

***client/Dockerfile***

*FROM python:latest*

*ADD client.py /client/*

*WORKDIR /client/*

***server/Dockerfile***

*FROM nginx:latest*

*COPY index.html /usr/share/nginx/html*

*WORKDIR /server/*

***docker-compose.yml***

*version: "3.8"*

*services:*

*server:*

*build: server/*

*ports:*

*- 1234:80*

*client:*

*build: client/*

*command: python ./client.py*

*network\_mode: host*

*depends\_on:*

*- server*

***docker-compose build***

*PS C:\Users\frayz\IdeaProjects\root>docker-compose build*

*Building server*

*failed to get console mode for stdout: The handle is invalid.*

*[+] Building 9.4s (9/9) FINISHED*

*=> [internal] load build definition from Dockerfile 0.0s*

*=> => transferring dockerfile: 111B 0.0s*

*=> [internal] load .dockerignore 0.0s*

*=> => transferring context: 2B 0.0s*

*=> [internal] load metadata for docker.io/library/nginx:latest 2.2s*

*=> [auth] library/nginx:pull token for registry-1.docker.io 0.0s*

*=> [internal] load build context 0.0s*

*=> => transferring context: 59B 0.0s*

*=> [1/3] FROM docker.io/library/nginx:latest@sha256:b0ea179ab61c789ce759 6.8s*

*=> => resolve docker.io/library/nginx:latest@sha256:b0ea179ab61c789ce759 0.0s*

*=> => sha256:9bfb4f5453fefe7819d7bcc795d6d09bbb4e3a9a93e 1.57kB / 1.57kB 0.0s*

*=> => sha256:09de04de3c750e3ebcce85cc5a312e7ce2d298917 26.58MB / 26.58MB 5.4s*

*=> => sha256:b0c8a51e66287c43c855b9538e067be5c245bfe14e96f51 602B / 602B 0.4s*

*=> => sha256:b0ea179ab61c789ce759dbe491cc534e293428ad232 1.86kB / 1.86kB 0.0s*

*=> => sha256:08b11a3d692c1a2e15ae840f2c15c18308dcb079aa5320e 894B / 894B 0.4s*

*=> => sha256:b8cf2cbeabb915843204ceb7ef0055fecadd55c2b0c 7.73kB / 7.73kB 0.0s*

*=> => sha256:a0e0e6bcfd2cd773311862c3fb2e6da7ab49d3c127faa20 667B / 667B 0.7s*

*=> => sha256:4fcb23e29ba19bf305d0d4b35412625fea51e82292e 1.41kB / 1.41kB 0.7s*

*=> => extracting sha256:09de04de3c750e3ebcce85cc5a312e7ce2d29891740129ea 0.9s*

*=> => extracting sha256:b0c8a51e66287c43c855b9538e067be5c245bfe14e96f516 0.0s*

*=> => extracting sha256:08b11a3d692c1a2e15ae840f2c15c18308dcb079aa5320e1 0.0s*

*=> => extracting sha256:a0e0e6bcfd2cd773311862c3fb2e6da7ab49d3c127faa200 0.0s*

*=> => extracting sha256:4fcb23e29ba19bf305d0d4b35412625fea51e82292ec7312 0.0s*

*=> [2/3] COPY index.html /usr/share/nginx/html 0.2s*

*=> [3/3] WORKDIR /server/ 0.0s*

*=> exporting to image 0.1s*

*=> => exporting layers 0.0s*

*=> => writing image sha256:239a0b59d8af35c0103b627a7cc23e079341b5c5f4450 0.0s*

*=> => naming to docker.io/library/docker\_nginx\_server 0.0s*

*Successfully built 239a0b59d8af35c0103b627a7cc23e079341b5c5f44503ea9089c79690a5c3f0*

*Building client*

*failed to get console mode for stdout: The handle is invalid.*

*[+] Building 1.4s (9/9) FINISHED*

*=> [internal] load build definition from Dockerfile 0.0s*

*=> => transferring dockerfile: 99B 0.0s*

*=> [internal] load .dockerignore 0.0s*

*=> => transferring context: 2B 0.0s*

*=> [internal] load metadata for docker.io/library/python:latest 1.3s*

*=> [auth] library/python:pull token for registry-1.docker.io 0.0s*

*=> [internal] load build context 0.0s*

*=> => transferring context: 235B 0.0s*

*=> [1/3] FROM docker.io/library/python:latest@sha256:8bd2e361ad8575ae80a 0.0s*

*=> CACHED [2/3] ADD client.py /client/ 0.0s*

*=> CACHED [3/3] WORKDIR /client/ 0.0s*

*=> exporting to image 0.0s*

*=> => exporting layers 0.0s*

*=> => writing image sha256:ad0b807ce9ef6bcb884e3a906889f7c3aca639c3bc788 0.0s*

*=> => naming to docker.io/library/docker\_nginx\_client 0.0s*

*Successfully built ad0b807ce9ef6bcb884e3a906889f7c3aca639c3bc788375313035587f35b1b9*

***docker-compose up***

*PS C:\Users\frayz\IdeaProjects\root>docker-compose up*

*Creating network "docker\_nginx\_default" with the default driver*

*Creating docker\_nginx\_server\_1 ... done*

*Creating docker\_nginx\_client\_1 ... done*

*Attaching to docker\_nginx\_server\_1, docker\_nginx\_client\_1*

*server\_1 | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration*

*server\_1 | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/*

*server\_1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh*

*server\_1 | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf*

*server\_1 | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf*

*server\_1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh*

*server\_1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh*

*server\_1 | /docker-entrypoint.sh: Configuration complete; ready for start up*

*server\_1 | 172.23.0.1 - - [29/Mar/2021:22:37:34 +0000] "GET / HTTP/1.1" 200 22 "-" "Python-urllib/3.9" "-"*

*client\_1 | =DOCKER-COMPOSE READY=*

*docker\_nginx\_client\_1 exited with code 0*

## Практическое задание №5

**Задание:** создать сервис в состав которого входят следующие компоненты apache, mysql, phpMyAdmin. Проект назвать docker\_phpmyadmin.

**Выполнение задания:**

***Dockerfile***

*FROM php:7.3-apache*

*RUN docker-php-ext-install mysqli pdo\_mysql*

*RUN apt-get update \*

*&& apt-get install -y libzip-dev \*

*&& apt-get install -y zlib1g-dev \*

*&& rm -rf /var/lib/apt/lists/\* \*

*&& docker-php-ext-install zip*

***docker-compose.yml***

*version: "3.1"*

*services:*

*www:*

*build: .*

*ports:*

*- "8001:80"*

*volumes:*

*- ./www:/var/www/html/*

*links:*

*- db*

*networks:*

*- default*

*db:*

*image: mysql:8.0*

*ports:*

*- "3306:3306"*

*command: --default-authentication-plugin=mysql\_native\_password*

*environment:*

*MYSQL\_DATABASE: myDb*

*MYSQL\_USER: user*

*MYSQL\_PASSWORD: test*

*MYSQL\_ROOT\_PASSWORD: test*

*volumes:*

*- ./dump:/docker-entrypoint-initdb.d*

*- ./conf:/etc/mysql/conf.d*

*- persistent:/var/lib/mysql*

*networks:*

*- default*

*phpmyadmin:*

*image: phpmyadmin/phpmyadmin*

*links:*

*- db:db*

*ports:*

*- 8000:80*

*environment:*

*MYSQL\_USER: user*

*MYSQL\_PASSWORD: test*

*MYSQL\_ROOT\_PASSWORD: test*

*volumes:*

*persistent:*

***index.php***

*<html>*

*<head>*

*<title>Apache and MySQL web application</title>*

*<meta charset="utf-8">*

*<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">*

*<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>*

*<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>*

*</head>*

*<body>*

*<div class="container">*

*<?php*

*$conn = mysqli\_connect('db', 'user', 'test', "myDb");*

*$query = 'SELECT \* From Person';*

*$result = mysqli\_query($conn, $query);*

*echo '<table class="table table-striped">';*

*echo '<thead><tr><th></th><th>id</th><th>name</th></tr></thead>';*

*while($value = $result->fetch\_array(MYSQLI\_ASSOC)){*

*echo '<tr>';*

*echo '<td><a href="#"><span class="glyphicon glyphicon-search"></span></a></td>';*

*foreach($value as $element){*

*echo '<td>' . $element . '</td>';*

*}*

*echo '</tr>';*

*}*

*echo '</table>';*

*$result->close();*

*mysqli\_close($conn);*

*?>*

*</div>*

*</body>*

*</html>*

***docker-compose up***

*PS C:\Users\frayz\IdeaProjects\root>docker-compose up -d*

*Creating network "apachemysqlproject\_default" with the default driver*

*Creating volume "apachemysqlproject\_persistent" with default driver*

*Building www*

*failed to get console mode for stdout: The handle is invalid.*

*[+] Building 1.7s (8/8) FINISHED*

*=> [internal] load build definition from Dockerfile 0.0s*

*=> => transferring dockerfile: 281B 0.0s*

*=> [internal] load .dockerignore 0.0s*

*=> => transferring context: 2B 0.0s*

*=> [internal] load metadata for docker.io/library/php:7.3-apache 1.5s*

*=> [auth] library/php:pull token for registry-1.docker.io 0.0s*

*=> [1/3] FROM docker.io/library/php:7.3-apache@sha256:63fa3e80042995e05c 0.0s*

*=> CACHED [2/3] RUN docker-php-ext-install mysqli pdo\_mysql 0.0s*

*=> CACHED [3/3] RUN apt-get update && apt-get install -y libzip-dev 0.0s*

*=> exporting to image 0.0s*

*=> => exporting layers 0.0s*

*=> => writing image sha256:d7949600d17308a2e7490817061e87963caff9e279142 0.0s*

*=> => naming to docker.io/library/apachemysqlproject\_www 0.0s*

*Successfully built d7949600d17308a2e7490817061e87963caff9e2791426a04f3776366f0bccd7*

*WARNING: Image for service www was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compose up --build`.*

*Creating apachemysqlproject\_db\_1 ... done*

*Creating apachemysqlproject\_www\_1 ... done*

*Creating apachemysqlproject\_phpmyadmin\_1 ... done*

**Вывод:** в ходе данной практической работы были получены навыки сборки Docker-контейнера с использованием Docker-composer, а также умения работы с консольным приложением и системного мышления.

**Список использованных источников**

1. Сейерс, Э. Х. Docker на практике / Э. Х. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. — ISBN 978-5-97060-772-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/131719 (дата обращения: 27.03.2021). — Режим доступа: для авториз. пользователей.
2. Кочер, П. С. Микросервисы и контейнеры Docker : руководство / П. С. Кочер ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2019. — 240 с. — ISBN 978-5-97060-739-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123710 (дата обращения: 27.03.2021). — Режим доступа: для авториз. пользователей.
3. Docker — URL: https://docs.docker.com (дата обращения: 04.04.2021).