МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«**МИРЭА – Российский технологический университет**»

РТУ МИРЭА

Институт комплексной безопасности и цифровых технологий (ИКБ)
Кафедра КБ-14 «Цифровые технологии обработки данных»

Администрирование баз данных

Практическая работа №2

Выполнили студенты 3 курса

Зарин Н.Н.

Луговой И.И.

БСБО-05-20

Москва 2023

Для выполнения задания был использован удаленный арендованный сервер.

Задание 1.

1. Установите Docker на локальной машине.

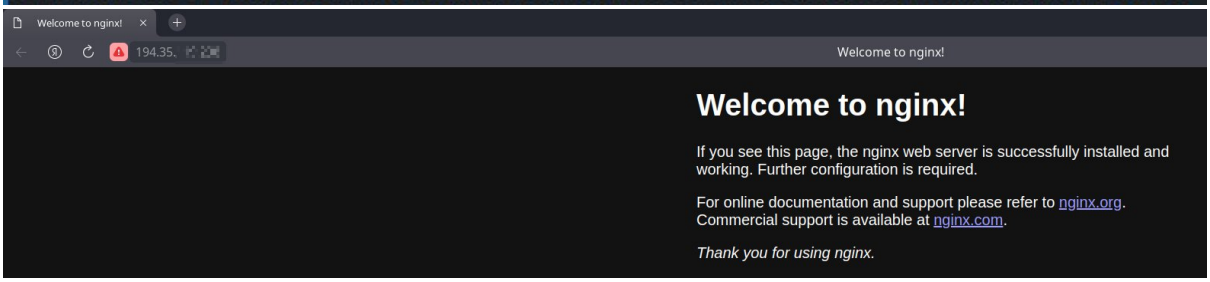
```
root@1162623-ci71069:~# docker -v
Docker version 24.0.2, build cb74dfc
```

2. Используя Docker Hub, загрузите образ Nginx

```
root@1162623-ci71069:~# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
f03b40093957: Pull complete
eed12bbd6494: Pull complete
fa7eb8c8eee8: Pull complete
7ff3b2b12318: Pull complete
0f67c7de5f2c: Pull complete
831f51541d38: Pull complete
Digest: sha256:af296b188c7b7df99ba960ca614439c99cb7cf252ed7bbc23e90cfda59092305
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

3. Запустите контейнер Nginx в фоновом режиме

```
root@1162623-ci71069:~# docker run -d -it -p 80:80 nginx
cf9d6d86c9f7e0a65d2f8752d0bf57a3f31c49cf09d1c853f63b80a1c72577a7
```



4. Вывести список работающих контейнеров

```
root@1162623-ci71069:~# docker ps -a -n 1
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                   NAMES
cf9d6d86c9f7   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp, :::80->80/tcp   focused_albattani
```

5. Остановите действующий контейнер

```
root@1162623-ci71069:~# docker stop cf9d6d86c9f7
cf9d6d86c9f7
```

6. Переименуйте остановленный контейнер на имя webhost

```
root@1162623-ci71069:~# docker rename cf9d6d86c9f7 webhost
root@1162623-ci71069:~#
```

7. Заново запустите контейнер с именем webhost

```
root@1162623-ci71069:~# docker start webhost
webhost
```

8. Выведите логи контейнера webhost

```
root@1162623-ci71069:~# docker logs -f webhost
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/06/04 15:39:01 [notice] 1#1: using the "epoll" event method
2023/06/04 15:39:01 [notice] 1#1: nginx/1.25.0
2023/06/04 15:39:01 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/06/04 15:39:01 [notice] 1#1: OS: Linux 5.4.0-150-generic
2023/06/04 15:39:01 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/06/04 15:39:01 [notice] 1#1: start worker processes
2023/06/04 15:39:01 [notice] 1#1: start worker process 28
46.148.[illegible] - - [04/Jun/2023:15:39:35 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 YaBrowser/23.3.3.764 Yowser/2.5 Safari/537.36" "-"
2023/06/04 15:39:35 [error] 28#28: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 46.148.[illegible], server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "194.35.[illegible]", r
eferrer: "http://194.35.[illegible]/"
46.148.[illegible] - - [04/Jun/2023:15:39:35 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://194.35.[illegible]/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 YaBrowser/23.3.3.764 Yowser/
2.5 Safari/537.36" "-"
2023/06/04 15:39:52 [error] 28#28: *2 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 46.148.[illegible], server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "194.35.[illegible]"
46.148.[illegible] - - [04/Jun/2023:15:39:52 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 YaBrowser/23.3.3.764 Yowser/2.5 Safari/537.36" "-
"
46.148.[illegible] - - [04/Jun/2023:15:39:52 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 YaBrowser/23.3.3.764 Yowser/2.5 Safari/537.36" "-"
2023/06/04 15:41:50 [notice] 1#1: signal 3 (SIGQUIT) received, shutting down
2023/06/04 15:41:50 [notice] 28#28: gracefully shutting down
2023/06/04 15:41:50 [notice] 28#28: exiting
2023/06/04 15:41:50 [notice] 28#28: exit
2023/06/04 15:41:50 [notice] 1#1: signal 17 (SIGCHLD) received from 28
2023/06/04 15:41:50 [notice] 1#1: worker process 28 exited with code 0
2023/06/04 15:41:50 [notice] 1#1: exit
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: IPv6 listen already enabled
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/06/04 15:43:42 [notice] 1#1: using the "epoll" event method
2023/06/04 15:43:42 [notice] 1#1: nginx/1.25.0
2023/06/04 15:43:42 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/06/04 15:43:42 [notice] 1#1: OS: Linux 5.4.0-150-generic
2023/06/04 15:43:42 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/06/04 15:43:42 [notice] 1#1: start worker processes
2023/06/04 15:43:42 [notice] 1#1: start worker process 21
```

Задание 2.

1. Загрузите образ Ubuntu из Docker Hub.

```
root@1162623-ci71069:~# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
837dd4791cdc: Pull complete
Digest: sha256:ac58ff7fe25edc58bdf0067ca99df00014dbd032e2246d30a722fa348fd799a5
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

2. Запустите контейнер с Ubuntu из загруженного образа и подключитесь к нему с помощью команды docker exec -it.

```
root@1162623-ci71069:~# docker run -d -it ubuntu
4fb63503897a9bbc4edd41d375c35112cf5d3afbadae23d1c68a8c8f96099873
root@1162623-ci71069:~# docker exec -it 4fb63503897a /bin/bash
root@4fb63503897a:/#
```

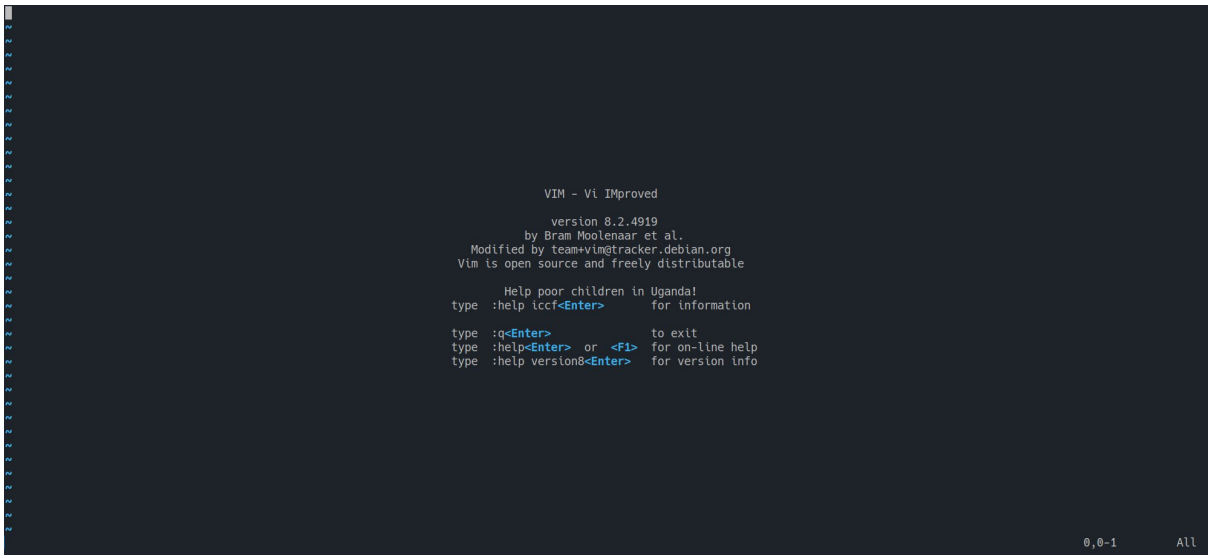
3. Используя команды Ubuntu, установите в контейнере несколько пакетов:, vim, curl, htop, ping.

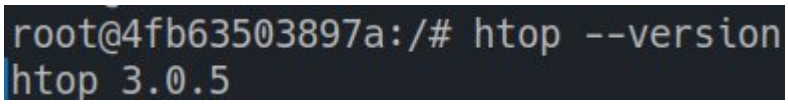
```
root@4fb63503897a:/# apt install vim curl htop iputils-ping
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ca-certificates libbrotli1 libcap2-bin libcurl4 libexpat1 libgpm2 libldap-2.5-0 libldap-common libmpdec3 libnghttp2-14 libnl-3-200 libnl-genl-3-200 libpam-cap libpsl5
  libpython3.10 libpython3.10-minimal libpython3.10-stdlib libreadline8 librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db libsodium23 libsqlite3-0 libssh-4 media-types
  openssl publicsuffix readline-common vim-common vim-runtime xxd
Suggested packages:
  lm-sensors lsof strace gpm libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal libsasl2-modules-ldap libsasl2-modules-otp libsasl2-modules-sql readline-doc ctags
  vim-doc vim-scripts
The following NEW packages will be installed:
  ca-certificates curl htop iputils-ping libbrotli1 libcap2-bin libcurl4 libexpat1 libgpm2 libldap-2.5-0 libldap-common libmpdec3 libnghttp2-14 libnl-3-200 libnl-genl-3-200
  libpam-cap libpsl5 libpython3.10 libpython3.10-minimal libpython3.10-stdlib libreadline8 librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db libsodium23 libsqlite3-0
  libssh-4 media-types openssl publicsuffix readline-common vim vim-common vim-runtime xxd
0 upgraded, 36 newly installed, 0 to remove and 0 not upgraded.
Need to get 17.8 MB of archives.
After this operation, 69.1 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

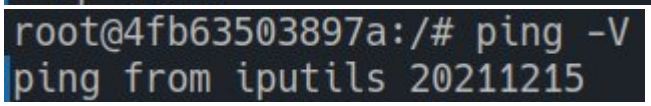
4. Остановите контейнер

```
root@1162623-ci71069:~# docker stop 4fb63503897a
4fb63503897a
```

5. Запустите его заново в интерактивном режиме и убедитесь что все пакеты установлены







Задание 3.

1. Выведите информацию о сети каждого существующего контейнера Ubuntu:

```
"NetworkSettings": {
    "Bridge": "",
    "SandboxID": "401bf8f4d15dbb61809b7c0de45ccba5061933d3113fb08c756297705cb6bcf0",
    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {},
    "SandboxKey": "/var/run/docker/netns/401bf8f4d15d",
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID": "7ad26d24d1a4d6fa1ef1db69f4757a68fabddb5cc9ebb24ea76d37b0270c1e7f",
    "Gateway": "172.17.0.1",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "172.17.0.4",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "MacAddress": "02:42:ac:11:00:04",
    "Networks": {
        "bridge": {
            "IPAMConfig": null,
            "Links": null,
            "Aliases": null,
            "NetworkID": "a7e6c601e216bdab41f17e7f0b15abb275adcc3d2505baa513f0dacb828765bd",
            "EndpointID": "7ad26d24d1a4d6fa1ef1db69f4757a68fabddb5cc9ebb24ea76d37b0270c1e7f",
            "Gateway": "172.17.0.1",
            "IPAddress": "172.17.0.4",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "MacAddress": "02:42:ac:11:00:04",
            "DriverOpts": null
        }
    }
}
```


Nginx:

```
"NetworkSettings": {
    "Bridge": "",
    "SandboxID": "338f0dad4667738e025b3caefed7ce238551b1f0e64c61ba883bfc09ee590ebe",
    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {
        "80/tcp": [
            {
                "HostIp": "0.0.0.0",
                "HostPort": "80"
            },
            {
                "HostIp": "::",
                "HostPort": "80"
            }
        ]
    },
    "SandboxKey": "/var/run/docker/netns/338f0dad4667",
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID": "ec1922f0b306d5801dee379eea8544b42d1757c25c9d4e0a3aa5a30f879ca89a",
    "Gateway": "172.17.0.1",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "172.17.0.3",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "MacAddress": "02:42:ac:11:00:03",
    "Networks": {
        "bridge": {
            "IPAMConfig": null,
            "Links": null,
            "Aliases": null,
            "NetworkID": "a7e6c601e216bdab41f17e7f0b15abb275adcc3d2505baa513f0dacb828765bd",
            "EndpointID": "ec1922f0b306d5801dee379eea8544b42d1757c25c9d4e0a3aa5a30f879ca89a",
            "Gateway": "172.17.0.1",
            "IPAddress": "172.17.0.3",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "MacAddress": "02:42:ac:11:00:03",
            "DriverOpts": null
        }
    }
}
```


2. Выведите все существующие сети в docker

```
root@1162623-ci71069:~# docker network ls
NETWORK ID     NAME                DRIVER    SCOPE
a7e6c601e216   bridge              bridge    local
14cb40463824   host                host      local
b42bf7ae8696   mongo_default       bridge    local
b885b5c5c3f9   none                null      local
f6ddc6394f83   postgres_default    bridge    local
```

3. Создайте свою собственную сеть типа Bridge

```
root@1162623-ci71069:~# docker network create -d bridge my_network
f65ece40511424d5bf8517fd2d6a41f95aa68cdc574ac40445686890486d1412
```

4. Подключите туда два созданных контейнера

```
root@1162623-ci71069:~# docker ps -a -n 2
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                   NAMES
4fb63503897a   ubuntu    "/bin/bash"              16 minutes ago   Up 11 minutes                                           affectionate_sammet
cf9d6d86c9f7   nginx     "/docker-entrypoint.…"   28 minutes ago   Up 24 minutes   0.0.0.0:80->80/tcp, :::80->80/tcp   webhost
root@1162623-ci71069:~#
root@1162623-ci71069:~# docker network connect my_network webhost
root@1162623-ci71069:~# docker network connect my_network affectionate_sammet
```

5. При помощи утилиты ping проверьте что контейнеры видят друг друга
Ubuntu:


```
"NetworkSettings": {
    "Bridge": "",
    "SandboxID": "863e569781eb81e25647ad6b9d0fcf08de0403a8ec927521f726052593733412",
    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {},
    "SandboxKey": "/var/run/docker/netns/863e569781eb",
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID": "9bbbea50b8a52def03f497b1cd8a8da431aab2c76db028d296cbb3d82861080f",
    "Gateway": "172.17.0.1",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "172.17.0.4",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "MacAddress": "02:42:ac:11:00:04",
    "Networks": {
        "bridge": {
            "IPAMConfig": null,
            "Links": null,
            "Aliases": null,
            "NetworkID": "a7e6c601e216bdab41f17e7f0b15abb275adcc3d2505baa513f0dacb828765bd",
            "EndpointID": "9bbbea50b8a52def03f497b1cd8a8da431aab2c76db028d296cbb3d82861080f",
            "Gateway": "172.17.0.1",
            "IPAddress": "172.17.0.4",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "MacAddress": "02:42:ac:11:00:04",
            "DriverOpts": null
        },
        "my_network": {
            "IPAMConfig": {},
            "Links": null,
            "Aliases": [
                "4fb63503897a"
            ],
            "NetworkID": "f65ece40511424d5bf8517fd2d6a41f95aa68cdc574ac40445686890486d1412",
            "EndpointID": "84d7ef3c0b8ec08d5d47be2595c05647d4240e76942fb83da851942541f9e320",
            "Gateway": "172.20.0.1",
            "IPAddress": "172.20.0.3",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "MacAddress": "02:42:ac:14:00:03",
            "DriverOpts": {}
        }
    }
}
```


Nginx:

```
"NetworkSettings": {
    "Bridge": "",
    "SandboxID": "dba57b107bc2a8956d8061f8ae503579f8c9b022680754becadec79b8b8aed05",
    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {
        "80/tcp": [
            {
                "HostIp": "0.0.0.0",
                "HostPort": "80"
            },
            {
                "HostIp": "::",
                "HostPort": "80"
            }
        ]
    },
    "SandboxKey": "/var/run/docker/netns/dba57b107bc2",
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID": "25f97dc32f6fcf721dcb5a5a06140d2b7f4645f9b267fe422c2b4230776fa14c",
    "Gateway": "172.17.0.1",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "172.17.0.3",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "MacAddress": "02:42:ac:11:00:03",
    "Networks": {
        "bridge": {
            "IPAMConfig": null,
            "Links": null,
            "Aliases": null,
            "NetworkID": "a7e6c601e216bdab41f17e7f0b15abb275adcc3d2505baa513f0dacb828765bd",
            "EndpointID": "25f97dc32f6fcf721dcb5a5a06140d2b7f4645f9b267fe422c2b4230776fa14c",
            "Gateway": "172.17.0.1",
            "IPAddress": "172.17.0.3",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "MacAddress": "02:42:ac:11:00:03",
            "DriverOpts": null
        },
        "my_network": {
            "IPAMConfig": {},
            "Links": null,
            "Aliases": [
                "cf9d6d86c9f7"
            ],
            "NetworkID": "f65ece40511424d5bf8517fd2d6a41f95aa68cdc574ac40445686890486d1412",
            "EndpointID": "22dfe9e043f277f8121d339d1c710e3a890cc2d0cc2ed8a43567d1ea5173d0fc",
            "Gateway": "172.20.0.1",
            "IPAddress": "172.20.0.2",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "MacAddress": "02:42:ac:14:00:02",
            "DriverOpts": {}
        }
    }
}
```

```
root@1162623-ci71069:~# docker exec -it 4fb63503897a /bin/bash
root@4fb63503897a:/# ping 172.20.0.2
PING 172.20.0.2 (172.20.0.2) 56(84) bytes of data.
64 bytes from 172.20.0.2: icmp_seq=1 ttl=64 time=0.194 ms
64 bytes from 172.20.0.2: icmp_seq=2 ttl=64 time=0.083 ms
64 bytes from 172.20.0.2: icmp_seq=3 ttl=64 time=0.097 ms
^C
--- 172.20.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2035ms
rtt min/avg/max/mdev = 0.083/0.124/0.194/0.049 ms
root@4fb63503897a:/#
```

Задание 4.

1. Создайте Dockerfile:
    a. Задайте свой образ на основе исходного образа Ubuntu
    b. Установите пакет nano
    c. Запустите команду "echo 'Hello, World!' > index.html" внутри контейнера
    d. Задайте рабочую директорию /usr/local/apache2/htdocs/
    e. Копируйте файл index.html в указанную директорию
    f. Установите пакет apache2
    g. Запустите Apache в foreground mode (чтобы контейнер не завершался сразу после запуска)

```
  GNU nano 4.8
FROM ubuntu

RUN apt update && apt upgrade -y
RUN apt install nano -y
RUN mkdir -p /var/www/html/
RUN echo 'Hello, World!' > /var/www/html/index.html
RUN apt install apache2 -y

CMD apache2ctl -D FOREGROUND
```

2. Соберите образ используя Dockerfile

```
root@1162623-ci71069:~/lab2# docker build -f Dockerfile .
[+] Building 17.3s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 248B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/ubuntu:latest
 => [1/6] FROM docker.io/library/ubuntu
 => CACHED [2/6] RUN apt update && apt upgrade -y
 => CACHED [3/6] RUN apt install nano -y
 => [4/6] RUN mkdir -p /var/www/html/
 => [5/6] RUN echo 'Hello, World!' > /var/www/html/index.html
 => [6/6] RUN apt install apache2 -y
 => exporting to image
 => => exporting layers
 => => writing image sha256:58a7d3809dac02559e3dad73c7d898ede60dd12faa0c72e10da870f021c1ed07
root@1162623-ci71069:~/lab2#
```

3. Запустите контейнер используя ваш образ

```
root@1162623-ci71069:~/lab2# docker run -d -it -p 8080:80 929ee05a9265
d2edd93188668ce4955366ccadeb034580287b4b8a4befefdd025feeb73a1a8f
root@1162623-ci71069:~/lab2#
```

4. Проверьте, что ваш веб-сервер работает корректно, открыв http://localhost в вашем браузере

Hello, World!

5. Остановите и удалите контейнер.

```
root@1162623-ci71069:~/lab2# docker ps -a -n 1
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                                       NAMES
d2edd9318866   929ee05a9265   "/bin/sh -c 'apache2…"   21 seconds ago   Up 20 seconds   0.0.0.0:8080->80/tcp, :::8080->80/tcp   dazzling_dijkstra
root@1162623-ci71069:~/lab2#
root@1162623-ci71069:~/lab2# docker stop d2edd9318866
d2edd9318866
root@1162623-ci71069:~/lab2#
```

6. Добавьте в Dockerfile команду, которая устанавливает пакет curl

```
  GNU nano 4.8
FROM ubuntu

RUN apt update && apt upgrade -y
RUN apt install nano -y
RUN mkdir -p /var/www/html/
RUN echo 'Hello, World!' > /var/www/html/index.html
RUN apt install apache2 -y
RUN apt install curl -y

CMD apache2ctl -D FOREGROUND
```

7. Пересоберите образ и запустите контейнер заново

```
root@1162623-ci71069:~/lab2# docker build -f Dockerfile .
[+] Building 3.3s (11/11) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 272B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/ubuntu:latest
 => [1/7] FROM docker.io/library/ubuntu
 => CACHED [2/7] RUN apt update && apt upgrade -y
 => CACHED [3/7] RUN apt install nano -y
 => CACHED [4/7] RUN mkdir -p /var/www/html/
 => CACHED [5/7] RUN echo 'Hello, World!' > /var/www/html/index.html
 => CACHED [6/7] RUN apt install apache2 -y
 => [7/7] RUN apt install curl -y
 => exporting to image
 => => exporting layers
 => => writing image sha256:e6fa4add3ead77dcf074894a60b424ab19bc2a11a8788e88dc40be0d1a2c2bd4
root@1162623-ci71069:~/lab2#
```

8. Проверьте, что пакет curl установлен внутри контейнера

```
root@1162623-ci71069:~/lab2# docker ps -a -n 3
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS             PORTS                                   NAMES
a05f2a1578c5   e6fa4add3ead   "/bin/sh -c 'apache2…"   8 seconds ago   Up 7 seconds       0.0.0.0:8080->80/tcp, :::8080->80/tcp   reverent_proskuriakova
4fb63503897a   ubuntu         "/bin/bash"              2 hours ago     Up About an hour                                           affectionate_sammet
cf9d6d86c9f7   nginx          "/docker-entrypoint.…"   2 hours ago     Up About an hour   0.0.0.0:80->80/tcp, :::80->80/tcp       webhost
root@1162623-ci71069:~/lab2# docker exec -it a05f2a1578c5 /bin/bash
root@a05f2a1578c5:/# curl --version
curl 7.81.0 (x86_64-pc-linux-gnu) libcurl/7.81.0 OpenSSL/3.0.2 zlib/1.2.11 brotli/1.0.9 zstd/1.4.8 libidn2/2.3.2 libpsl/0.21.0 (+libidn2/2.3.2) libssh/0.9.6/openssl/zlib nghttp2/1.
43.0 librtmp/2.3 OpenLDAP/2.5.14
Release-Date: 2022-01-05
Protocols: dict file ftp ftps gopher gophers http https imap imaps ldap ldaps mqtt pop3 pop3s rtmp rtsp scp sftp smb smbs smtp smtps telnet tftp
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM NTLM_WB PSL SPNEGO SSL TLS-SRP UnixSockets zstd
root@a05f2a1578c5:/#
```