



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

**Институт кибербезопасности и цифровых технологий
Цифровая кафедра**

ОТЧЁТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ
Практика по получению профессиональных умений и опыта
профессиональной деятельности

Тема практики: «Создание собственного контейнера»

Отчет представлен к
рассмотрению:

«__» _____ 2023
г.

Студент группы: ИВБО-07-21

(Подпись) Стока И.П.

Отчет утвержден.
Допущен к защите:

Руководитель практики от кафедры

«__» _____ 2023 г.

(Подпись) А.Т. Тарланов

Москва 2023 г.

СОДЕРЖАНИЕ

Введение.....	3
Создание машины.....	4
Создание docker контейнеров	5
Приготовления.....	5
Создание контейнера № 1	5
Создание контейнера № 2	6
Создание контейнера № 3	7
Заключение.....	9

Введение

Темой практики является создание собственного контейнера в Docker. Docker – программное обеспечение для автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений. Docker позволяет, упаковать приложение со всем его окружением и зависимостями в контейнер.

Создание машины

Создание машины производится на сервисе Базис от Ростелекома. После выделения необходимых мощностей: CPU, memory size и другие, создаем виртуальную машину (Рисунок 1).

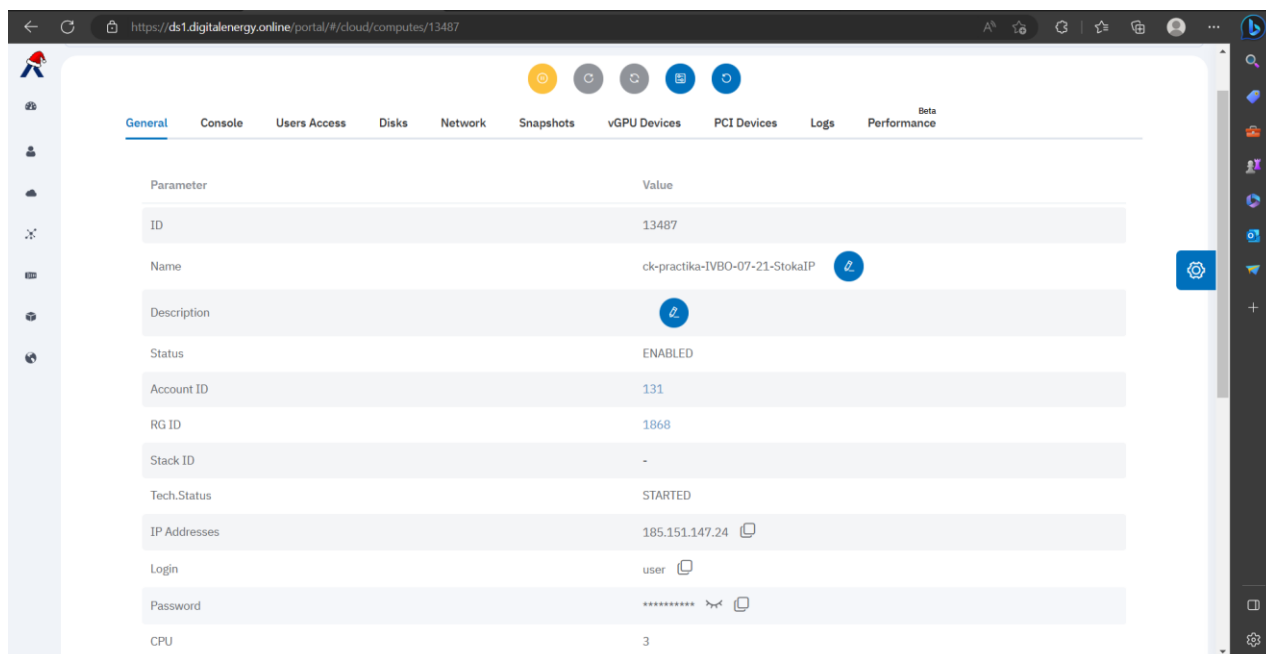


Рисунок 1 – Виртуальная машина

Создание docker контейнеров

Приготовление

Для начала скачает сам docker (Рисунок 2).

```
user@ck-practika--07-21-toka:~$ curl -fsSL https://get.docker.com -o get-docker.sh
user@ck-practika--07-21-toka:~$ _
```

Рисунок 2 – Установка docker

Проверим работоспособность после установки, создав контейнер hello-world (Рисунок 3).

```
Digest: sha256:fffb13da98453e0f04d33a6eee5bb8e46ee50d08ebe17735fc0779d0349e889e9
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

user@ck-practika--07-21-toka:/home$
```

Рисунок 3 – Проверка работоспособности

Создание контейнера № 1

Посредством Dockerfile создаётся контейнер, который будет пинговать по адресу 8.8.8.8, дополнительно передадим команде ping параметры для отправки только 5 пакетов (Рисунок 4).

```
GNU nano 6.2 Dockerfile *
FROM alpine

RUN ping -c 5 8.8.8.8
```

Рисунок 4 – Dockerfile контейнера ping

Далее необходимо сбилдить и запустить контейнер (Рисунок 5).

```
user@ck-practika--07-21-toka:~/docker1$ sudo docker run ping
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=57 time=17.679 ms
64 bytes from 8.8.8.8: seq=1 ttl=57 time=17.299 ms
64 bytes from 8.8.8.8: seq=2 ttl=57 time=17.438 ms
64 bytes from 8.8.8.8: seq=3 ttl=57 time=17.256 ms
64 bytes from 8.8.8.8: seq=4 ttl=57 time=17.439 ms
```

Рисунок 5 – Запуск контейнера

Создание контейнера № 2

Необходимо создать контейнер, который копирует файл в нужную директорию и в дальнейшем выводит его содержимое (Рисунок 6).

```
GNU nano 6.2 Dockerfile
FROM alpine

WORKDIR /home/user

COPY ./docker1/text .

CMD [ "cat", "/home/user/text" ]
```

Рисунок 6 – Dockerfile контейнера copyfile

Далее строится контейнер и запускается (Рисунок 7).

```

user@ck-practika--07-21-toka:~$ sudo docker build -t copyfile .
[+] Building 0.8s (8/8) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 127B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/alpine:latest 0.5s
=> [1/3] FROM docker.io/library/alpine@sha256:124c7d2707904eea7431fffe91 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 83B 0.0s
=> CACHED [2/3] WORKDIR /home/user 0.0s
=> [3/3] COPY ./docker1/text . 0.1s
=> exporting to image 0.1s
=> => exporting layers 0.1s
=> => writing image sha256:3ee2e367b17962b682d69db15086ee0be06300bd7fb19 0.0s
=> => naming to docker.io/library/copyfile 0.0s
user@ck-practika--07-21-toka:~$ sudo docker run copyfile
hello world!
user@ck-practika--07-21-toka:~$ _

```

Рисунок 7 – Запуск контейнера

Создание контейнера № 3

Необходимо создать контейнер, выполняющий команду `sl`. Нужно обратить внимание на менеджера пакетов в `Dockerfile`, так как там используется `apk` менеджер, а также вместо `install` используется `add` (Рисунок 8).

```

GNU nano 6.2 Dockerfile *
FROM alpine

RUN apk update && apk add sl

CMD [ "sl" ]

```

Рисунок 8 – Dockerfile контейнера train

Далее необходимо сбилдить контейнер (Рисунок 9).

```

user@ck-practika--07-21-toka:~/docker3$ sudo docker build -t train .
[+] Building 3.3s (6/6) FINISHED
=> [internal] load .dockerignore                                0.1s
=> => transferring context: 2B                                  0.0s
=> [internal] load build definition from Dockerfile             0.1s
=> => transferring dockerfile: 93B                              0.0s
=> [internal] load metadata for docker.io/library/alpine:latest 0.9s
=> CACHED [1/2] FROM docker.io/library/alpine@sha256:124c7d2707904eea743 0.0s
=> [2/2] RUN apk update && apk add sl                          2.2s
=> exporting to image                                           0.1s
=> => exporting layers                                          0.1s
=> => writing image sha256:9d16aef412996b4b1322dbfef2d68921915b1fde9b1f7 0.0s
=> => naming to docker.io/library/train                        0.0s
user@ck-practika--07-21-toka:~/docker3$ _

```

Рисунок 9 – Build контейнера train

Далее запустим контейнер (Рисунок 10).



Рисунок 10 – Запуск контейнера

Заключение

На практике были получены навыки и знания использования контейнеризации, на примере программного обеспечения docker, в сервисе Базис от Ростелекома. Получены навыки работы с образами, в частности alpine, создание Dockerfile и его конфигурации, знакомство с файловым менеджером, а также работа с Docker Hub.