РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

«Основы работы с Docker»

Отчет по лабораторной работе

по дисциплине «Анализ данных»

Выполнил студент группы ИВТ-б-о-21-1

Шайдеров Дмитрий Викторович.

«13» ноября 2023г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: научиться использовать основные команды Docker для управления контейнерами и понимать их назначение.

Порядок выполнения работы:

Задача 1: Основы Docker

Загрузите образ Ubuntu с Docker Hub.

```
user@Shayderov:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
aece8493d397: Pull complete
Digest: sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
user@Shayderov:~$
```

Рисунок 1 – Загрузка образа ubuntu

Создайте и запустите контейнер на основе этого образа.

```
user@Shayderov:~$ docker run -it ubuntu
```

Рисунок 2 - Запуск контейнера

Войдите в созданный контейнер и выполните команду ls , чтобы просмотреть файлы внутри контейнера.

```
root@f2b4d4fca238:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@f2b4d4fca238:/#
```

Рисунок 3 - Выполнение команды ls внутри контейнера

Задача 2: Управление контейнерами и образами

Загрузите образ Nginx с Docker Hub.

```
user@Shayderov:~$ docker pull nginx:latest
latest: Pulling from library/nginx
578acb154839: Pull complete
e398db710407: Pull complete
85c41ebe6d66: Pull complete
7170a263b582: Pull complete
8f28d06e2e2e: Pull complete
6f837de2f887: Pull complete
c1dfc7e1671e: Pull complete
Digest: sha256:86e53c4c16a6a276b204b0fd3a8143d86547c967dc8258b3d47c3a21bb68d3c6
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

Рисунок 4 – Загрузка образа ngnix

Создайте контейнер на основе этого образа и пробросьте порт 8080 контейнера на порт 80 хоста.

```
user@Shayderov:~$ docker run -p 8080:80 -d nginx
fda9231f09220787954c6c3752666f80d915c2c4d4985b4b9ba1563864bd13fd
```

Рисунок 5 - Создание контейнера и проброс порта

Посмотрите список активных контейнеров и убедитесь, что ваш контейнер работает.

```
user@Shayderov:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED             STATUS             PORTS                  NAMES
fda9231f0922   nginx     "/docker-entrypoint.…"   22 seconds ago      Up 22 seconds      0.0.0.0:8080->80/tcp   sweet_antonelli
f2b4d4fca238   ubuntu    "/bin/bash"              About an hour ago   Up About an hour                          intelligent_murdock
```

Рисунок 6 - Список активных контейнеров

Остановите и удалите контейнер.

```
user@Shayderov:~$ docker stop sweet_antonelli
sweet_antonelli
user@Shayderov:~$ docker rm sweet_antonelli
sweet_antonelli
```

Рисунок 7 - Остановка и удаление контейнера

Задача 3: Мониторинг и управление контейнерами

Запустите контейнер с именем "my_container".

```
user@Shayderov:~$ docker run --name my_container -d nginx
e19c997f2c558b7724243344935e9470b16f927d27f4956d91f3ec565148a830
```

Рисунок 8 – Запуск контейнера

Используя команду docker ps , убедитесь, что контейнер запущен.

```
user@Shayderov:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED             STATUS             PORTS     NAMES
e19c997f2c55   nginx     "/docker-entrypoint.…"   53 seconds ago      Up 53 seconds      80/tcp    my_container
f2b4d4fca238   ubuntu    "/bin/bash"              About an hour ago   Up About an hour              intelligent_murdock
```

Рисунок 9 - Список активных контейнеров

Остановите контейнер.

```
user@Shayderov:~$ docker stop my_container
my_container
```

Рисунок 10 - Остановка контейнера

Проверьте его статус снова и убедитесь, что он остановлен.

```
user@Shayderov:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED             STATUS                      PORTS     NAMES
e19c997f2c55   nginx     "/docker-entrypoint.…"   About a minute ago  Exited (0) 19 seconds ago             my_container
f2b4d4fca238   ubuntu    "/bin/bash"              About an hour ago   Up About an hour                      intelligent_murdock
```

Рисунок 11 - Просмотр активных контейнеров

Удалите контейнер.

```
user@Shayderov:~$ docker rm my_container
my_container
```

Рисунок 12 - Удаление контейнера

Задача 4: Удаление образов и оптимизация дискового пространства

Загрузите образы Ubuntu и Alpine с Docker Hub.

```
user@Shayderov:~$ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
96526aa774ef: Pull complete
Digest: sha256:eece025e432126ce23f223450a0326fbebde39cdf496a85d8c016293fc851978
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
```

Рисунок 13 - Загрузка образа alpine

Создайте контейнеры на основе обоих образов.

```
user@Shayderov:~$ docker run --name container_1 -d ubuntu
864e95ea1c6caa4de5e597ae73464e3aede82ef4982cdaffac0fa6f1cbb7703f
user@Shayderov:~$ docker run --name container_2 -d alpine
2082d45490e901531fe424904a64830b0156d8560f7ab2128b1a37840732f1c3
```

Рисунок 14 - Создание контейнеров

Убедитесь, что контейнеры запущены и работают.

```
user@Shayderov:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED              STATUS                      PORTS     NAMES
2082d45490e9   alpine    "/bin/sh"     About a minute ago   Exited (0) 35 seconds ago             container_2
864e95ea1c6c   ubuntu    "/bin/bash"   2 minutes ago        Exited (0) 39 seconds ago             container_1
f2b4d4fca238   ubuntu    "/bin/bash"   About an hour ago    Up About an hour                      intelligent_murdock
```

Рисунок 15 - Список контейнеров

Удалите образ Ubuntu.

```
user@Shayderov:~$ docker rmi -f ubuntu
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
```

Рисунок 16 - Удаление образа ubuntu

Проверьте, что образ Ubuntu больше не существует, но образ Alpine остался на системе.

```
user@Shayderov:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED         STATUS                     PORTS     NAMES
2082d45490e9   alpine         "/bin/sh"     7 minutes ago   Exited (0) 6 minutes ago             container_2
864e95ea1c6c   e4c58958181a   "/bin/bash"   7 minutes ago   Exited (0) 3 minutes ago             container_1
```

Рисунок 17 - Просмотр контейнеров

Задача 5: Взаимодействие с контейнером

Запустите контейнер с именем "my_container" в фоновом режиме.

```
user@Shayderov:~$ docker run --name my_container -d nginx
983fc2d8e19f81b90a273490f7d1cd575c6a4e6f70e7e0d12edd821bab87f5d7
```

Рисунок 18 – Запуск контейнера

Используя команду docker exec , выполните команду ls -l /app внутри контейнера.

```
user@Shayderov:~$ docker exec my_container ls -l /app
ls: cannot access '/app': No such file or directory
user@Shayderov:~$ docker exec my_container ls -l
total 64
lrwxrwxrwx   1 root root    7 Oct 30 00:00 bin -> usr/bin
drwxr-xr-x   2 root root 4096 Sep 29 20:04 boot
drwxr-xr-x   5 root root  340 Nov 15 18:26 dev
drwxr-xr-x   1 root root 4096 Nov  1 05:12 docker-entrypoint.d
-rwxrwxr-x   1 root root 1620 Nov  1 05:11 docker-entrypoint.s
drwxr-xr-x   1 root root 4096 Nov 15 18:26 etc
drwxr-xr-x   2 root root 4096 Sep 29 20:04 home
lrwxrwxrwx   1 root root    7 Oct 30 00:00 lib -> usr/lib
lrwxrwxrwx   1 root root    9 Oct 30 00:00 lib32 -> usr/lib32
lrwxrwxrwx   1 root root    9 Oct 30 00:00 lib64 -> usr/lib64
lrwxrwxrwx   1 root root   10 Oct 30 00:00 libx32 -> usr/libx3
drwxr-xr-x   2 root root 4096 Oct 30 00:00 media
drwxr-xr-x   2 root root 4096 Oct 30 00:00 mnt
drwxr-xr-x   2 root root 4096 Oct 30 00:00 opt
dr-xr-xr-x 309 root root    0 Nov 15 18:26 proc
drwx------   2 root root 4096 Oct 30 00:00 root
drwxr-xr-x   1 root root 4096 Nov 15 18:26 run
lrwxrwxrwx   1 root root    8 Oct 30 00:00 sbin -> usr/sbin
drwxr-xr-x   2 root root 4096 Oct 30 00:00 srv
dr-xr-xr-x  11 root root    0 Nov 15 18:26 sys
drwxrwxrwt   1 root root 4096 Nov  1 05:12 tmp
drwxr-xr-x   1 root root 4096 Oct 30 00:00 usr
drwxr-xr-x   1 root root 4096 Oct 30 00:00 var
```

Рисунок 19 - Выполнение команды ls -l

Выполните команду ps aux внутри контейнера, чтобы увидеть список запущенных процессов.

```
root@293de6cca461:/# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.3  0.0   4624  3612 pts/0    Ss   19:19   0:00 /bin/bash
root           9  0.0  0.0   7060  1668 pts/0    R+   19:20   0:00 ps aux
root@293de6cca461:/#
```

Рисунок 20 - Выполнение команды ps aux

Остановите и удалите контейнер.

```
root@293de6cca461:/# user@Shayderov:~$ docker stop my_container
my_container
user@Shayderov:~$ docker rm my_container
my_container
```

Рисунок 21 - Остановка и удаление контейнера

Контрольные вопросы:

1. Что делает команда docker pull?

Команда docker pull в Docker используется для загрузки образа контейнера с Docker Hub или другого репозитория.

2. Какой синтаксис используется для загрузки образа с Docker Hub с помощью docker pull?

docker pull <имя_образа>:<тег>

3. Как можно просмотреть список всех доступных образов на системе с помощью docker images?

docker images

Эта команда выведет список всех образов, которые находятся на вашей системе, включая их имена, теги, размер и ID.

4. Какой ключ используется для просмотра образов в формате таблицы с docker images?

docker images --format "table {{.Repository}}\t{{.Tag}}\t{{.Size}}"

5. Как создать и запустить контейнер с использованием docker run?

docker run [опции] <имя_образа> [команда] [аргументы]

6. Как пробросить порт при запуске контейнера с docker run?

docker run -p 8080:80 nginx

7. Как изменить имя контейнера при его создании с помощью docker run?

docker run --name my_container -d nginx

8. Как создать контейнер в фоновом режиме с docker run?

docker run -d nginx

9. Какая команда используется для просмотра активных контейнеров на системе?

docker ps

10. Какие опции могут использоваться с docker ps для отображения остановленных контейнеров?

docker ps -a

11. Как можно просмотреть список всех контейнеров, включая остановленные, с docker ps?

docker ps -a

12. Что делает команда docker start?

Команда docker start в Docker используется для запуска остановленных контейнеров.

13. Какой синтаксис используется для запуска остановленного контейнера с docker start?

docker start [опции] <имя_или_ID_контейнера>

14. Как запустить контейнер в фоновом режиме с docker start?

docker start -d my_container

15. Что делает команда docker stop?

Команда docker stop в Docker используется для остановки работающего контейнера.

16. Как остановить контейнер по его имени с помощью docker stop?

docker stop my_container

17. Как принудительно остановить контейнер с docker stop?

docker stop -f my_container

18. Что делает команда docker rm?

Команда docker rm в Docker используется для удаления контейнера, который был остановлен.

19. Как удалить контейнер по его ID с использованием docker rm?

docker rm 1234567890

20. Как удалить несколько контейнеров сразу с docker rm?

docker rm container1 container2

21. Что делает команда docker rmi?

Команда docker rmi в Docker используется для удаления образов контейнеров с вашей системы.

22. Как удалить Docker-образ по его имени и тегу с помощью docker rmi?

docker rmi ubuntu:20.04

23. Как удалить несколько Docker-образов сразу с docker rmi?

docker rmi image1 image2

24. Как выполнить команду внутри работающего контейнера с docker exec?

docker exec [опции] <имя_или_ID_контейнера> <команда> [аргументы]

25. Как выполнить команду внутри контейнера в интерактивном режиме с docker exec?

docker exec -it my_container /bin/bash

26. Как выполнить команду с использованием определенного пользователя внутри контейнера с docker exec?

docker exec -u 1000 my_container whoami

27. Какой ключ используется для запуска команды в фоновом режиме с docker exec?

docker exec -d my_container my_command

28. Как выполнить команду внутри контейнера с именем вместо ID с docker exec?

docker exec -it $(docker ps -q -f "name=my_container") /bin/bash

29. Как передать аргументы при выполнении команды с docker exec?

docker exec [опции] <имя_или_ID_контейнера> <команда> [аргументы]

30. Как проверить список доступных команд и опций для docker exec?

docker exec --help

31. Как передать переменную окружения в контейнер при его запуске?

docker run -e MYSQL_ROOT_PASSWORD=my-secret-pw mysql

32. Какой ключ используется для запуска контейнера в фоновом режиме с командой docker run?

docker run -d nginx

33. Как проверить статус выполнения контейнеров на системе с помощью docker ps?

docker ps -s

34. Как завершить выполнение контейнера без его удаления?

docker stop my_container

35. Каким образом можно удалить все остановленные контейнеры с системы?

docker rm $(docker ps -aq)

36. Что делает опция -a при использовании docker ps?

Добавление опции -a позволяет просматривать все контейнеры, включая те, которые были остановлены.

37. Что означает опция -q при выполнении docker ps?

Добавление опции –q выводит только ID контейнеров.

38. Как принудительно удалить контейнер с флагом -f?

docker rm -f my_container

39. Какой Docker-образ и какую команду можно использовать для создания контейнера с базой данных PostgreSQL?

docker run –name postgres_container postgres

40. Какой ключ используется для выполнения команды внутри контейнера в интерактивном режиме?

docker exec -it my_container <команда>

41. Какой ключ можно использовать для передачи ID пользователя при выполнении команды внутри контейнера?

С опцией -u мы указываем ID пользователя, от имени которого будет выполнена команда.

Вывод: были изучены основные команды Docker для управления контейнерами.