

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное
учреждение высшего образования**
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Кафедра инфокоммуникаций
Институт цифрового развития

ОТЧЁТ

по лабораторной работе №1

Дисциплина: «Конфигурационное распределенное управление ПО»

Тема: «Основы работы с Docker»

Вариант 2

Выполнил: студент 3 курса,

группы ИВТ-б-о-21-1

Богдан Александр Анатольевич

Ставрополь 2023

Цель занятия: Научить студентов использовать основные команды Docker для управления контейнерами и понимать их назначение.

Практическая часть:

Задание 1. Основы Docker

- Загрузите образ Ubuntu с Docker Hub.

```
root@LAPTOP-8N8C4IMT:~# 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
```

- Создайте и запустите контейнер на основе этого образа.

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

```
root@LAPTOP-8N8C4IMT:~# docker run -it ubuntu
root@4ea3d74ea216:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@4ea3d74ea216:/#
```

Задание 2. Управление контейнерами и образами

- Загрузите образ Nginx с Docker Hub.

```
root@4ea3d74ea216:/# root@LAPTOP-8N8C4IMT:~# docker pull nginx:latest
latest: Pulling from library/nginx
1f7ce2fa46ab: Pull complete
9b16c94bb686: Pull complete
9a59d19f9c5b: Pull complete
9ea27b074f71: Pull complete
c6edf33e2524: Pull complete
84b1ff10387b: Pull complete
517357831967: Pull complete
Digest: sha256:10d1f5b58f74683ad34eb29287e07dab1e90f10af243f151bb50aa5dbb4d62ee
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

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

```
super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker run -p 8080:80 -d nginx
786d3d5768b779e4ea3308ba281304ed79e4af496be440291920c121e557bc75
```

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

```
root@LAPTOP-8N8C4IMT:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                  NAMES
cb42a1fd15ad   nginx    "/docker-entrypoint..." 56 seconds ago Up 54 seconds  0.0.0.0:8080->80/tcp   adoring_panini
root@LAPTOP-8N8C4IMT:~#
```

- Остановите и удалите контейнер.

```
root@LAPTOP-8N8C4IMT:~# docker stop adoring_panini
adoring_panini
root@LAPTOP-8N8C4IMT:~# docker rm adoring_panini
adoring_panini
```

Задание 3. Мониторинг и управление контейнерами

- Запустите контейнер с именем "my_container".

```
root@LAPTOP-8N8C4IMT:~# docker run --name my_container -d nginx
c4825e043351ee8346982735ee55ecc8d166b0a1253500091182723d7e7bb259
```

- Используя команду `docker ps`, убедитесь, что контейнер запущен.

```
root@LAPTOP-8N8C4IMT:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                  NAMES
c4825e043351   nginx    "/docker-entrypoint..." 30 seconds ago Up 29 seconds  80/tcp                my_container
root@LAPTOP-8N8C4IMT:~#
```

- Остановите контейнер.

```
root@LAPTOP-8N8C4IMT:~# docker stop my_container
my_container
```

- Проверьте его статус снова и убедитесь, что он остановлен.

```
root@LAPTOP-8N8C4IMT:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                  NAMES
root@LAPTOP-8N8C4IMT:~#
```

- Удалите контейнер.

```
root@LAPTOP-8N8C4IMT:~# docker rm my_container
my_container
```

Задание 4. Удаление образов и оптимизация дискового пространства

- Загрузите образы Ubuntu и Alpine с Docker Hub.

```
root@LAPTOP-8N8C4IMT:~# docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
96526aa774ef: Pull complete
Digest: sha256:eece025e432126ce23f223450a0326fbbebde39cdf496a85d8c016293fc851978
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
```

- Создайте контейнеры на основе обоих образов.

```
root@LAPTOP-8N8C4IMT:~# docker run --name c_alpine -d alpine
8e3da2704e8ee8f12461f99802f761f0a9386b355ca5fa55d435d6f67afaed7f
root@LAPTOP-8N8C4IMT:~# docker run --name c_ubuntu -d ubuntu
65aeca520deba9062a834b8fd238f52fba4c901a728b2560fa626c45515548e7
```

- Убедитесь, что контейнеры запущены и работают.

```
root@LAPTOP-8N8C4IMT:~# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b82fa6380209	alpine	"/bin/sh"	7 minutes ago	Exited (0) 2 minutes ago		c_alpine
4a77b02528d9	ubuntu	"/bin/bash"	7 minutes ago	Exited (0) 2 minutes ago		c_ubuntu

- Удалите образ Ubuntu.

```
root@LAPTOP-8N8C4IMT:~# docker rmi -f ubuntu
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Deleted: sha256:e4c58958181a5925816faa528ce959e487632f4cfd192f8132f71b32df2744b4
```

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

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b82fa6380209	alpine	"/bin/sh"	8 minutes ago	Exited (0) 3 minutes ago		c_alpine
4a77b02528d9	e4c58958181a	"/bin/bash"	9 minutes ago	Exited (0) 3 minutes ago		c_ubuntu

Задание 5. Взаимодействие с контейнером

- Запустите контейнер с именем "my_container" в фоновом режиме.

```
root@LAPTOP-8N8C4IMT:~# docker run --name my_container -d nginx  
ae1c204188c6aa067c20e8cc602ec5ad03f0eafef3d308f2dd66976686db73c4
```

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

```
root@LAPTOP-8N8C4IMT:~# docker exec my_container ls -l  
total 64  
lrwxrwxrwx    1 root root    7 Nov 20 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 23 12:47 dev  
drwxr-xr-x    1 root root 4096 Nov 21 09:05 docker-entrypoint.d  
-rwxrwxr-x    1 root root 1620 Nov 21 09:05 docker-entrypoint.sh  
drwxr-xr-x    1 root root 4096 Nov 23 12:47 etc  
drwxr-xr-x    2 root root 4096 Sep 29 20:04 home  
lrwxrwxrwx    1 root root    7 Nov 20 00:00 lib -> usr/lib  
lrwxrwxrwx    1 root root    9 Nov 20 00:00 lib32 -> usr/lib32  
lrwxrwxrwx    1 root root    9 Nov 20 00:00 lib64 -> usr/lib64  
lrwxrwxrwx    1 root root   10 Nov 20 00:00 libx32 -> usr/libx32  
drwxr-xr-x    2 root root 4096 Nov 20 00:00 media  
drwxr-xr-x    2 root root 4096 Nov 20 00:00 mnt  
drwxr-xr-x    2 root root 4096 Nov 20 00:00 opt  
dr-xr-xr-x 303 root root    0 Nov 23 12:47 proc  
drwx-----   2 root root 4096 Nov 20 00:00 root  
drwxr-xr-x    1 root root 4096 Nov 23 12:47 run  
lrwxrwxrwx    1 root root    8 Nov 20 00:00 sbin -> usr/sbin  
drwxr-xr-x    2 root root 4096 Nov 20 00:00 srv  
dr-xr-xr-x   11 root root    0 Nov 23 12:47 sys  
drwxrwxrwt    1 root root 4096 Nov 21 09:05 tmp  
drwxr-xr-x    1 root root 4096 Nov 20 00:00 usr  
drwxr-xr-x    1 root root 4096 Nov 20 00:00 var  
root@LAPTOP-8N8C4IMT:~#
```

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

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

- Остановите и удалите контейнер.

```
root@LAPTOP-8N8C4IMT:~# docker stop my_container
my_container
root@LAPTOP-8N8C4IMT:~# docker rm my_container
my_container
```

Контрольные вопросы:

1. Что делает команда docker pull?

Используется для загрузки образа контейнера с Docker Hub или другого репозитория. Это важный этап при работе с контейнерами, так как он позволяет получить образ, который затем можно использовать для создания и запуска контейнеров.

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
```

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. Она позволяет вам возобновить выполнение контейнера, который был ранее приостановлен или остановлен.

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

`docker start my_container`

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

`docker start my_container`

15. Что делает команда docker stop?

Используется для остановки работающего контейнера. Это важная команда, которая позволяет контролировать жизненный цикл контейнеров.

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` ?

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` ?

Используется для удаления образов контейнеров с вашей системы. Это важная команда, которая позволяет освобождать дисковое пространство и управлять образами на вашей системе.

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
```

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

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

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

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

```
docker exec --help
```

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

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

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

```
docker run -d nginx
```

34. Как завершить выполнение контейнера без его удаления?

```
docker ps -s
```

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

```
docker stop my_container
```

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

```
docker rm $(docker ps -aq)
```

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

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

38. Как принудительно удалить контейнер с флагом -f?

Добавление опции -q выводит только ID контейнеров.

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

```
docker rm -f my_container
```

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

```
docker run --name postgres_container postgres
```

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

```
docker exec -it my_container
```

42. 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: в результате выполнения работы были изучены основные команды Docker для управления контейнерами.