

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙ-
СКОЙ ФЕДЕРАЦИИ**

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

Кафедра инфокоммуникаций

Курсы Dev-Ops

Отчет по лабораторной работе №1

Тема: «Знакомство с Docker»

Выполнил студент группы

ИВТ-б-о-21-1

Богадунов В.И. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель Воронкин Р.А.

(подпись)

Ставрополь 2023

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

Ход работы:

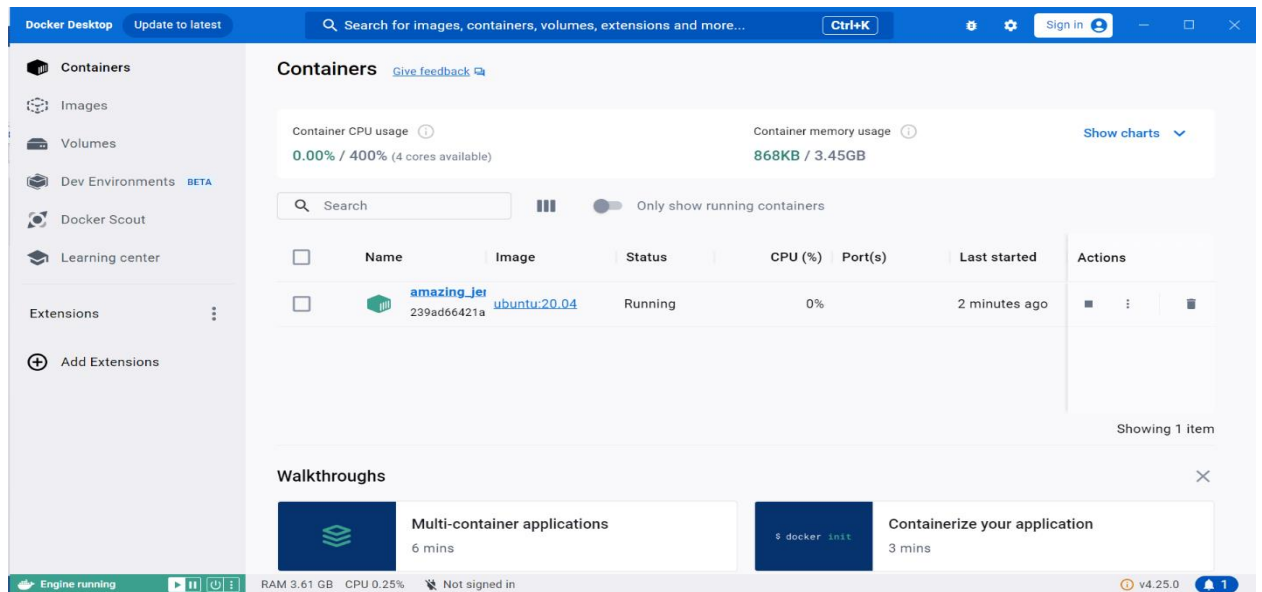


Рисунок 1 – Установленный Docker

```
PS C:\Users\Admin> docker

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
run           Create and run a new container from an image
exec         Execute a command in a running container
ps           List containers
build        Build an image from a Dockerfile
pull         Download an image from a registry
push         Upload an image to a registry
images       List images
login        Log in to a registry
logout       Log out from a registry
search       Search Docker Hub for images
version      Show the Docker version information
info         Display system-wide information

Management Commands:
builder      Manage builds
buildx*     Docker Buildx (Docker Inc., v0.11.2-desktop.5)
compose*    Docker Compose (Docker Inc., v2.23.0-desktop.1)
container   Manage containers
context      Manage contexts
dev*        Docker Dev Environments (Docker Inc., v0.1.0)
extension*  Manages Docker extensions (Docker Inc., v0.2.20)
image       Manage images
```

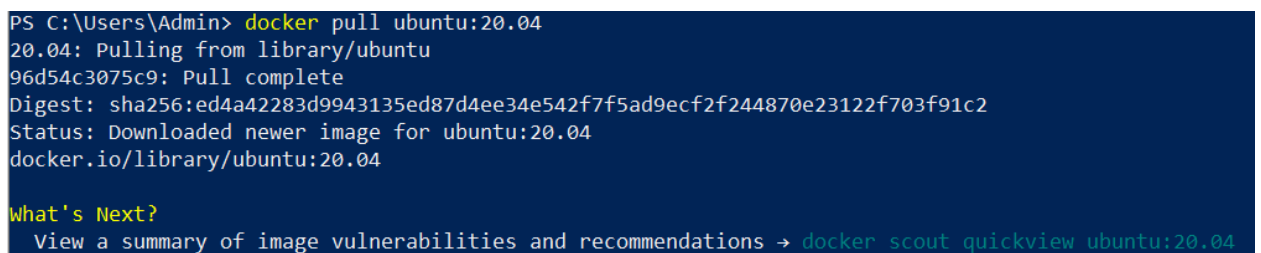
Рисунок 2 – Проверка установки Docker Практическая часть

Задача 1: Основы Docker

Цель: Познакомиться с основами Docker и командами для работы с контейнерами.

Задача:

- Загрузите образ Ubuntu с Docker Hub.
- Создайте и запустите контейнер на основе этого образа.
- Войдите в созданный контейнер и выполните команду `ls`, чтобы просмотреть файлы внутри контейнера.



```
PS C:\Users\Admin> docker pull ubuntu:20.04
20.04: Pulling from library/ubuntu
96d54c3075c9: Pull complete
Digest: sha256:ed4a42283d9943135ed87d4ee34e542f7f5ad9ecf2f244870e23122f703f91c2
Status: Downloaded newer image for ubuntu:20.04
docker.io/library/ubuntu:20.04

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu:20.04
```

Рисунок 3 – Создание контейнера с Ubuntu



```
PS C:\Users\Admin> docker run -it ubuntu:20.04
root@239ad66421a2:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@239ad66421a2:/#
```

Рисунок 4 – Вход в контейнер и вывод команды `ls`

Задача 2: Управление контейнерами и образами.

Цель: Освоить команды для управления контейнерами и образами Docker.

Задача:

- Загрузите образ Nginx с Docker Hub.
- Создайте контейнер на основе этого образа и пробросьте порт 8080 контейнера на порт 80 хоста.
- Посмотрите список активных контейнеров и убедитесь, что ваш контейнер работает.
- Остановите и удалите контейнер.

```

PS C:\Users\Admin> docker pull nginx
Using default tag: 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

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview nginx
PS C:\Users\Admin> docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
docker/welcome-to-docker  latest         c1f619b6477e   9 days ago     18.6MB
nginx                latest         c20060033e06   2 weeks ago    187MB
ubuntu              20.04         bf40b7bc7a11   6 weeks ago    72.8MB
PS C:\Users\Admin>

```

Рисунок 5 – Установка образа Nginx

```

PS C:\Users\Admin> docker run -p 8080:80 nginx
/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: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/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
2023/11/17 03:48:36 [notice] 1#1: using the "epoll" event method
2023/11/17 03:48:36 [notice] 1#1: nginx/1.25.3
2023/11/17 03:48:36 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/11/17 03:48:36 [notice] 1#1: OS: Linux 5.15.90.1-microsoft-standard-WSL2
2023/11/17 03:48:36 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/11/17 03:48:36 [notice] 1#1: start worker processes
2023/11/17 03:48:36 [notice] 1#1: start worker process 29
2023/11/17 03:48:36 [notice] 1#1: start worker process 30
2023/11/17 03:48:36 [notice] 1#1: start worker process 31
2023/11/17 03:48:36 [notice] 1#1: start worker process 32

```

Рисунок 6 – Создание контейнера на основе образа Nginx и проброс порта 8080 на порт 80 хоста










<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	 amazing_jer 239ad66421a	ubuntu:20.04	Exited	0%		17 minutes ago	  
<input type="checkbox"/>	 peaceful_m 0c33504bc21	nginx	Running	0%	8080:80 	3 minutes ago	  

Рисунок 7 – Список активных контейнеров

```

PS C:\Users\Admin> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
f7573469d2d1   nginx    "/docker-entrypoint..." About a minute ago Exited (0) 41 seconds ago           great_galois
f89794772f87   ubuntu:20.04 "/bin/bash"            9 minutes ago   Exited (0) 9 minutes ago           nervous_shirley
PS C:\Users\Admin> docker rm f7573469d2d1
f7573469d2d1
PS C:\Users\Admin> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
f89794772f87   ubuntu:20.04 "/bin/bash"            10 minutes ago Exited (0) 10 minutes ago           nervous_shirley
PS C:\Users\Admin>

```

Рисунок 8 – Удаление контейнера

Задача 3: Мониторинг и управление контейнерами

Цель: Изучить команды мониторинга и управления контейнерами.

Задача:

Запустите контейнер с именем "my_container".

- Используя команду `docker ps`, убедитесь, что контейнер запущен.
- Остановите контейнер.
- Проверьте его статус снова и убедитесь, что он остановлен.
- Удалите контейнер.

```
PS C:\Users\Admin> docker run --name my_container
"docker run" requires at least 1 argument.
See 'docker run --help'.

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Create and run a new container from an image
PS C:\Users\Admin> docker run --name my_container -d ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
aece8493d397: Pull complete
Digest: sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Status: Downloaded newer image for ubuntu:latest
5880091b7b8b144b77561c2d53931c92a8e13d86c86d752053735221b94d261c
PS C:\Users\Admin> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
PS C:\Users\Admin> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
5880091b7b8b   ubuntu    "/bin/bash"   3 minutes ago   Exited (0) 3 minutes ago           my_container
f89794772f87   ubuntu:20.04    "/bin/bash"   29 minutes ago   Exited (0) 28 minutes ago           nervous_shirley
PS C:\Users\Admin> docker stop my_container
my_container
PS C:\Users\Admin>
```

Рисунок 9 – Запуск контейнера + остановка контейнера

```
PS C:\Users\Admin> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
5880091b7b8b   ubuntu    "/bin/bash"   16 minutes ago   Exited (0) 16 minutes ago           my_container
f89794772f87   ubuntu:20.04    "/bin/bash"   42 minutes ago   Exited (0) 41 minutes ago           nervous_shirley
PS C:\Users\Admin> docker rm 5880091b7b8b
5880091b7b8b
PS C:\Users\Admin> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
f89794772f87   ubuntu:20.04    "/bin/bash"   2 hours ago   Exited (0) 2 hours ago           nervous_shirley
PS C:\Users\Admin>
```

Рисунок 10 – Проверка статуса и удаление контейнера

Задача 4: Удаление образов и оптимизация дискового пространства

Цель: Освоить команды для удаления образов и оптимизации использования дискового пространства.

Задача:

- Загрузите образы Ubuntu и Alpine с Docker Hub.
- Создайте контейнеры на основе обоих образов.
- Убедитесь, что контейнеры запущены и работают.
- Удалите образ Ubuntu.
- Проверьте, что образ Ubuntu больше не существует, но образ Alpine остался на системе.

```
PS C:\Users\Admin> 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

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview alpine
PS C:\Users\Admin> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu
PS C:\Users\Admin> docker run -it ubuntu
root@8b81f0149a9d:/# exit
exit
PS C:\Users\Admin> docker run -it alpine
/ # exit
PS C:\Users\Admin> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7648669a78c0	alpine	"/bin/sh"	14 seconds ago	Exited (0) 8 seconds ago		hopeful_dubinsky
8b81f0149a9d	ubuntu	"/bin/bash"	25 seconds ago	Exited (0) 21 seconds ago		wonderful_perlman
f89794772f87	ubuntu:20.04	"/bin/bash"	2 hours ago	Exited (0) 2 hours ago		nervous_shirley

```
PS C:\Users\Admin>
```

Рисунок 11 – Загрузка образов Ubuntu и Alpine + создание контейнеров на основе обоих образов

```
PS C:\Users\Admin> docker start 7648669a78c0
7648669a78c0
PS C:\Users\Admin> docker start 8b81f0149a9d
8b81f0149a9d
PS C:\Users\Admin> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7648669a78c0	alpine	"/bin/sh"	4 minutes ago	Up 35 seconds		hopeful_dubinsky
8b81f0149a9d	ubuntu	"/bin/bash"	4 minutes ago	Up 6 seconds		wonderful_perlman
f89794772f87	ubuntu:20.04	"/bin/bash"	2 hours ago	Exited (0) 2 hours ago		nervous_shirley

```
PS C:\Users\Admin>
```

Рисунок 12 – Оба контейнера запущены и работают

```
PS C:\Users\Admin> docker stop wonderful_perlman
wonderful_perlman
PS C:\Users\Admin> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7648669a78c0	alpine	"/bin/sh"	8 minutes ago	Up 4 minutes		hopeful_dubinsky
8b81f0149a9d	e4c58958181a	"/bin/bash"	8 minutes ago	Exited (137) 14 seconds ago		wonderful_perlman
f89794772f87	ubuntu:20.04	"/bin/bash"	2 hours ago	Exited (0) 2 hours ago		nervous_shirley

```
PS C:\Users\Admin> docker rmi -f e4c58958181a
PS C:\Users\Admin> docker rmi -f e4c58958181a
Deleted: sha256:e4c58958181a5925816faa528ce959e487632f4cfd192f8132f71b32df2744b4
PS C:\Users\Admin> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx	latest	c20060033e06	2 weeks ago	187MB
ubuntu	20.04	bf40b7bc7a11	6 weeks ago	72.8MB
alpine	latest	8ca4688f4f35	7 weeks ago	7.34MB

```
PS C:\Users\Admin> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7648669a78c0	alpine	"/bin/sh"	9 minutes ago	Up 5 minutes		hopeful_dubinsky
8b81f0149a9d	e4c58958181a	"/bin/bash"	9 minutes ago	Exited (137) About a minute ago		wonderful_perlman
f89794772f87	ubuntu:20.04	"/bin/bash"	2 hours ago	Exited (0) 2 hours ago		nervous_shirley

```
PS C:\Users\Admin>
```

Рисунок 13 – Удаление образа Ubuntu и проверка остался ли образ Alpine

Задача 5: Взаимодействие с контейнером

Цель: Научиться взаимодействовать с работающим контейнером и выполнить команды внутри него.

Задача:

- Запустите контейнер с именем "my_container" в фоновом режиме.
- Используя команду `docker exec`, выполните команду `ls -l /app` внутри контейнера.
- Выполните команду `ps aux` внутри контейнера, чтобы увидеть список запущенных процессов.
- Остановите и удалите контейнер.

```
PS C:\Users\Admin> docker run -it --name my_container2 -d ubuntu:20.04
9e02969340f8f41fe0075984f0fa00f4a571d55f56cb7fb149d2867c6d1f453a
PS C:\Users\Admin> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
9e02969340f8   ubuntu:20.04   "/bin/bash"             4 seconds ago  Up 3 seconds              my_container2
57ac4377c435   ubuntu:20.04   "/bin/bash"             About a minute ago  Exited (0) About a minute ago  my_container1
54dd7a56e773   ubuntu         "/bin/bash"             9 minutes ago    Exited (0) 2 minutes ago      my_container
7648669a78c0   alpine         "/bin/sh"               13 hours ago     Exited (255) 5 minutes ago     hopeful_dubinsky
```

Рисунок 14 – Запуск контейнера с именем “my_container2”

```
PS C:\Users\Admin> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
9e02969340f8   ubuntu:20.04   "/bin/bash"             44 seconds ago  Up 43 seconds              my_container2
7648669a78c0   alpine         "/bin/sh"               13 hours ago     Exited (255) 5 minutes ago     hopeful_dubinsky
8b81f0149a9d   ubuntu         "/bin/bash"             13 hours ago     Up 4 minutes              wonderful_perlman
f89794772f87   ubuntu:20.04   "/bin/bash"             15 hours ago     Exited (0) 15 hours ago      nervous_shirley
PS C:\Users\Admin> docker exec my_container2 ls -l /app
ls: cannot access '/app': No such file or directory
PS C:\Users\Admin> docker exec my_container2 ls -l
total 48
lrwxrwxrwx 1 root root 7 Oct 3 02:03 bin -> usr/bin
drwxr-xr-x 2 root root 4096 Apr 15 2020 boot
drwxr-xr-x 5 root root 360 Nov 18 03:40 dev
drwxr-xr-x 1 root root 4096 Nov 18 03:40 etc
drwxr-xr-x 2 root root 4096 Apr 15 2020 home
lrwxrwxrwx 1 root root 7 Oct 3 02:03 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Oct 3 02:03 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Oct 3 02:03 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Oct 3 02:03 libx32 -> usr/libx32
drwxr-xr-x 2 root root 4096 Oct 3 02:03 media
drwxr-xr-x 2 root root 4096 Oct 3 02:03 mnt
drwxr-xr-x 2 root root 4096 Oct 3 02:03 opt
dr-xr-xr-x 263 root root 0 Nov 18 03:40 proc
drwx----- 2 root root 4096 Oct 3 02:06 root
drwxr-xr-x 5 root root 4096 Oct 3 02:06 run
lrwxrwxrwx 1 root root 8 Oct 3 02:03 sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Oct 3 02:03 srv
dr-xr-xr-x 11 root root 0 Nov 18 03:40 sys
drwxrwxrwt 2 root root 4096 Oct 3 02:06 tmp
drwxr-xr-x 13 root root 4096 Oct 3 02:03 usr
drwxr-xr-x 11 root root 4096 Oct 3 02:06 var
PS C:\Users\Admin>
```

Рисунок 15 – Выполнение команды `ls -l /app` внутри контейнера.

```
PS C:\Users\Admin> docker exec my_container2 ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  4116  3316 pts/0    Ss+   03:40   0:00 /bin/bash
root      21  0.0  0.0  5900  2800 ?        Rs    03:47   0:00 ps aux
PS C:\Users\Admin>
```

Рисунок 16 – Выполнение команды `ps aux` внутри контейнера


```

PS C:\Users\Admin> docker stop my_container2
my_container2
PS C:\Users\Admin> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
9e02969340f8   ubuntu:20.04   "/bin/bash"             9 minutes ago   Exited (0) 5 seconds ago
7648669a78c0   alpine         "/bin/sh"               13 hours ago    Exited (255) 14 minutes ago
8b81f0149a9d   ubuntu        "/bin/bash"             13 hours ago    Up 13 minutes
f89794772f87   ubuntu:20.04   "/bin/bash"             15 hours ago    Exited (0) 15 hours ago
PS C:\Users\Admin> docker rm my_container2
my_container2
PS C:\Users\Admin> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
7648669a78c0   alpine         "/bin/sh"               13 hours ago    Exited (255) 15 minutes ago
8b81f0149a9d   ubuntu        "/bin/bash"             13 hours ago    Up 14 minutes
f89794772f87   ubuntu:20.04   "/bin/bash"             15 hours ago    Exited (0) 15 hours ago
PS C:\Users\Admin>

```

Рисунок 17 – Остановка и удаление контейнера

Вывод: в результате выполнения лабораторной работы были использованы основные команды Docker для управления контейнерами, а также были получены теоретические сведения для понимания их назначения.

Дополнительное задание: реализовать использование команды `docker run` в фоновом режиме (демон режиме).

```

PS C:\Users\Admin> docker run --detach hello-world
5c15f253043db1ed578be889d01e88c427907cfd792f656734ba8e4ad53d7822
PS C:\Users\Admin> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
5c15f253043d   hello-world    "/hello"                11 seconds ago   Exited (0) 10 seconds ago
beb1124dfe0e   ubuntu:18.04   "/bin/bash"             16 minutes ago   Exited (0) 16 minutes ago
6f2908c72b67   ubuntu        "/bin/bash"             6 hours ago      Up 6 hours
7648669a78c0   alpine         "/bin/sh"               35 hours ago     Exited (255) 22 hours ago
8b81f0149a9d   ubuntu        "/bin/bash"             35 hours ago     Up 22 hours
f89794772f87   ubuntu:20.04   "/bin/bash"             36 hours ago     Exited (0) 36 hours ago
PS C:\Users\Admin> docker run hello-world

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

To generate this message, Docker took the following steps:

```

Рисунок 18 – Реализация запуска и создание контейнера в фоновом режиме на основе образа “Hello-world”

Для решения данной задачи к команде `docker run` необходимо было применить ключ `-d` или `--detach`, которые в фоновом режиме реализуют запуск контейнера.

Ответы на контрольные вопросы:

1 Что делает команда `docker pull`?

Ответ: загружает образ. 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?

Ответ: `docker pull <имя_образа>:<тег>`

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

Ответ: просто написать `Docker images`

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

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

Этот пример выводит список образов в формате таблицы, показывая имя, тег и размер каждого образа.

5 Как создать и запустить контейнер с использованием docker run?

Ответ: Запуск контейнера на основе образа Ubuntu:

`docker run -it ubuntu`

В этом примере `-it` позволяет взаимодействовать с контейнером в интерактивном режиме. После запуска этой команды вы будете внутри контейнера с командной строкой Ubuntu.

6 Как пробросить порт при запуске контейнера с docker run?

Ответ: Проброс портов при запуске контейнера:

`docker run -p 8080:80 nginx`

Эта команда запускает контейнер `nginx` и пробрасывает порт 80 контейнера на порт 8080 хоста.

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

Ответ: Запуск контейнера с указанием имени:

`docker run --name my_container -d nginx`

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

Ответ: `docker run --name my_container -d nginx`

В команде `docker run` использовать ключ `-d`.

9 Какая команда используется для просмотра активных контейнеров на системе?

Ответ: `docker ps`

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

Ответ: `docker ps -a`

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

Ответ: добавить к `docker ps` ключ `-a`

12 Что делает команда `docker start`?

Ответ: запускает контейнер.

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

Ответ: `docker start *имя контейнера*`

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

Ответ: написать просто `docker start`.

15 Что делает команда `docker stop`?

Ответ: останавливает контейнер.

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

Ответ: `docker stop *Имя контейнера*`

17 Как принудительно остановить контейнер с docker stop?

Ответ: добавить ключ `-t` (`--time`) и указать рядом время в секундах, например: 30. Если за 30 секунд он не завершится самостоятельно, то он будет завершён принудительно.

18 Что делает команда docker rm?

Ответ: удаляет контейнер.

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

Ответ: `docker rm *его id*`

20 Как удалить несколько контейнеров сразу с docker rm?

Ответ: `docker rm *1 контейнер* *2 контейнер*`

21 Что делает команда docker rmi?

Ответ: удаляет образ.

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

Ответ: `docker rmi *тег или имя*`

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

Ответ: `docker rmi *1 контейнер* *2 контейнер*`

24 Как выполнить команду внутри работающего контейнера с docker exec?

Ответ: `docker exec my_container ls -l /app`

Эта команда выполняет `ls -l /app` внутри контейнера `my_container`, что позволяет нам просматривать содержимое папки `/app` в контейнере.

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

Ответ: `docker exec -it my_container /bin/bash`

В этом примере мы используем опции `-it`, чтобы войти в интерактивный режим, и запускаем оболочку `/bin/bash` внутри контейнера с именем `my_container`.

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

Ответ: `docker exec -u 1000 my_container whoami`

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

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

Ответ: Опция `-d` позволяет запустить команду в фоновом режиме, без интерактивного взаимодействия.

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

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

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

Ответ: `docker exec my_container ls -l /app`, где `ls -l /app` передаваемый аргументы.

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

Ответ: `docker exec --help`

31 Как передать переменную окружения в контейнер при его запуске?

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

Этот пример запускает контейнер MySQL и передает значение переменной окружения MYSQL_ROOT_PASSWORD .

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

Ответ: -d или --detach.

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

Ответ: ввести docker ps и посмотреть графу в таблице STATUS.

34 Как завершить выполнение контейнера без его удаления?

Ответ: docker stop.

35 Каким образом можно удалить все остановленные контейнеры с системы?

Ответ: docker rm \$(docker ps -aq)

36 Что делает опция -a при использовании docker ps?

Ответ: все созданные контейнеры показывает.

37 Что означает опция -q при выполнении docker ps ?

38 Как принудительно удалить контейнер с флагом -f ?

Ответ: docker rm -f *имя контейнера или id*

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

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

Ответ: docker start -i my_container

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

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

Вместо “name=my_container” указать id

`docker exec -it $(docker ps -q -f "id=6f2908c72b67") /bin/bash`