

Задание 1: Запуск экземпляров приложения с подключением томов и сети

Цель этого задания - ознакомиться с основами работы с Docker volumes и Docker network. Также вы научитесь использовать команду docker logs

Шаги:

- Найдите на Docker Hub образ, который вы хотите запустить на вашей машине.
- Используя команду docker run, запустите контейнер на основе этого образа. Добавьте флаги, чтобы установить имя контейнера, перенаправить порты, установить переменные окружения и т.д. Используйте команду docker ps, чтобы убедиться, что контейнер запущен.
- Создайте Docker volume с помощью команды docker volume create.
- Запустите контейнер, используя команду docker run, и подключите созданный Docker volume к контейнеру.
- Используйте команду docker network create для создания новой Docker network.

Опционально:

- Создайте второй контейнер из того же образа, используя команду docker run, и подключите его к созданной Docker network.
- Внутри первого контейнера, используя утилиту nano, создайте текстовый файл в подключенном Docker volume.
- Внутри первого контейнера, используя утилиту logger, запишите несколько сообщений в созданный текстовый файл.
- Используйте команду docker logs для просмотра логов первого контейнера и убедитесь, что сообщения были успешно записаны в лог-файл.
- Используйте команду docker exec для выполнения команды во втором контейнере и проверьте, что созданный текстовый файл доступен из второго контейнера.
- Используйте команду docker stop для остановки обоих контейнеров.
- Используйте команду docker rm для удаления контейнеров и команду docker rmi для удаления образа.
- Используйте команду docker volume rm для удаления Docker volume и команду docker network rm для удаления Docker network.

```
test@test-VirtualBox: ~
test@test-VirtualBox:~$ docker volume create DZ
DZ
test@test-VirtualBox:~$ docker volume ls
DRIVER      VOLUME NAME
local       DZ
test@test-VirtualBox:~$
```

```
test@test-VirtualBox:~$ docker run -t -d -v DZ:/volume -p 8080:8080 --name alp1 alpine
411973a5886ccc540924f8594f94ada4d4fb2486620e7c8627c1147dbd09b750
test@test-VirtualBox:~$ docker run -t -d -v DZ:/volume -p 8080:8080 --name alp2 alpine
caf94803fbbd026126ef94f6 added56942873de8a8109e2f2d5c43e25665e3c7ae
test@test-VirtualBox:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
caf94803fbdd   alpine    "/bin/sh"               16 seconds ago Up 14 seconds  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  alp2
411973a5886c   alpine    "/bin/sh"               38 seconds ago Up 36 seconds  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  alp1
test@test-VirtualBox:~$
```

```
test@test-VirtualBox:~$ docker network ls
NETWORK ID     NAME      DRIVER  SCOPE
9996a68cfa27   bridge    bridge   local
ae09a27d7b28   host      host     local
d474095e2ba4   none      null     local
test@test-VirtualBox:~$ docker network create DZ_NET
f9278fe4ae5669f80bc01b578867aa0a7c497715b40049316eed38c0b3c9c7aa
test@test-VirtualBox:~$ docker network ls
NETWORK ID     NAME      DRIVER  SCOPE
f9278fe4ae56   DZ_NET    bridge   local
9996a68cfa27   bridge    bridge   local
ae09a27d7b28   host      host     local
d474095e2ba4   none      null     local
test@test-VirtualBox:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
caf94803fbdd   alpine    "/bin/sh"               10 minutes ago Up 10 minutes  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  alp2
411973a5886c   alpine    "/bin/sh"               10 minutes ago Up 10 minutes  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  alp1
test@test-VirtualBox:~$ docker network connect DZ_NET caf94803fbdd
test@test-VirtualBox:~$ docker network connect DZ_NET 411973a5886c
test@test-VirtualBox:~$ docker inspect 411973a5886c | grep "DZ_NET"
    "DZ_NET": {
test@test-VirtualBox:~$ docker inspect caf94803fbdd | grep "DZ_NET"
    "DZ_NET": {
test@test-VirtualBox:~$
```

```
test@test-VirtualBox:~$ docker network create DZ_NET
f9278fe4ae5669f80bc01b578867aa0a7c497715b40049316eed38c0b3c9c7aa
test@test-VirtualBox:~$ docker network ls
NETWORK ID     NAME      DRIVER  SCOPE
f9278fe4ae56   DZ_NET    bridge   local
9996a68cfa27   bridge    bridge   local
ae09a27d7b28   host      host     local
d474095e2ba4   none      null     local
test@test-VirtualBox:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
caf94803fbdd   alpine    "/bin/sh"               10 minutes ago Up 10 minutes  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  alp2
411973a5886c   alpine    "/bin/sh"               10 minutes ago Up 10 minutes  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  alp1
test@test-VirtualBox:~$ docker network connect DZ_NET caf94803fbdd
test@test-VirtualBox:~$ docker network connect DZ_NET 411973a5886c
test@test-VirtualBox:~$ docker inspect 411973a5886c | grep "DZ_NET"
    "DZ_NET": {
test@test-VirtualBox:~$ docker inspect caf94803fbdd | grep "DZ_NET"
    "DZ_NET": {
test@test-VirtualBox:~$ docker exec -it 411973a5886c sh
/ # cd volume
/volume # touch 123.txt
/volume # ls -l
total 0
-rw-r--r-- 1 root root 0 Aug 12 13:44 123.txt
/volume # echo "test volume" >> 123.txt
/volume # cat 123.txt
test volume
/volume #
```

```
test@test-VirtualBox:~$ docker network create DZ_NET
f9278fe4ae5669f80bc01b578867aa0a7c497715b40049316eed38c0b3c9c7aa
test@test-VirtualBox:~$ docker network ls
NETWORK ID     NAME      DRIVER  SCOPE
f9278fe4ae56   DZ_NET    bridge   local
9996a68cfa27   bridge    bridge   local
ae09a27d7b28   host      host     local
d474095e2ba4   none      null     local
test@test-VirtualBox:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
caf94803fbdd   alpine    "/bin/sh"               10 minutes ago Up 10 minutes  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  alp2
411973a5886c   alpine    "/bin/sh"               10 minutes ago Up 10 minutes  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  alp1
test@test-VirtualBox:~$ docker network connect DZ_NET caf94803fbdd
test@test-VirtualBox:~$ docker network connect DZ_NET 411973a5886c
test@test-VirtualBox:~$ docker inspect 411973a5886c | grep "DZ_NET"
    "DZ_NET": {
test@test-VirtualBox:~$ docker inspect caf94803fbdd | grep "DZ_NET"
    "DZ_NET": {
test@test-VirtualBox:~$ docker exec -it 411973a5886c sh
/ # cd volume
/volume # touch 123.txt
/volume # ls -l
total 0
-rw-r--r-- 1 root root 0 Aug 12 13:44 123.txt
/volume # echo "test volume" >> 123.txt
/volume # cat 123.txt
test volume
/volume # exit
test@test-VirtualBox:~$ docker exec caf94803fbdd cat /volume/123.txt
test volume
test@test-VirtualBox:~$
test@test-VirtualBox:~$
```

```
test@test-VirtualBox:~$ docker exec -it 411973a5886c sh
/ # cd volume
/volume # touch 123.txt
/volume # ls -l
total 0
-rw-r--r-- 1 root root 0 Aug 12 13:44 123.txt
/volume # echo "test volume" >> 123.txt
/volume # cat 123.txt
test volume
/volume # exit
test@test-VirtualBox:~$ docker exec caf94803fbdd cat /volume/123.txt
test volume
test@test-VirtualBox:~$ docker stop caf94803fbdd 411973a5886c
caf94803fbdd
411973a5886c
test@test-VirtualBox:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
caf94803fbdd   alpine    "/bin/sh"               47 minutes ago Exited (137) 12 seconds ago  alp2
411973a5886c   alpine    "/bin/sh"               47 minutes ago Exited (137) 12 seconds ago  alp1
test@test-VirtualBox:~$ docker rm 411973a5886c caf94803fbdd
411973a5886c
caf94803fbdd
test@test-VirtualBox:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
test@test-VirtualBox:~$ docker network rm DZ_NET
DZ_NET
test@test-VirtualBox:~$ docker network ls
NETWORK ID     NAME      DRIVER  SCOPE
9996a68cfa27   bridge    bridge   local
ae09a27d7b28   host      host     local
d474095e2ba4   none      null     local
test@test-VirtualBox:~$
```