Лабораторная работа №2

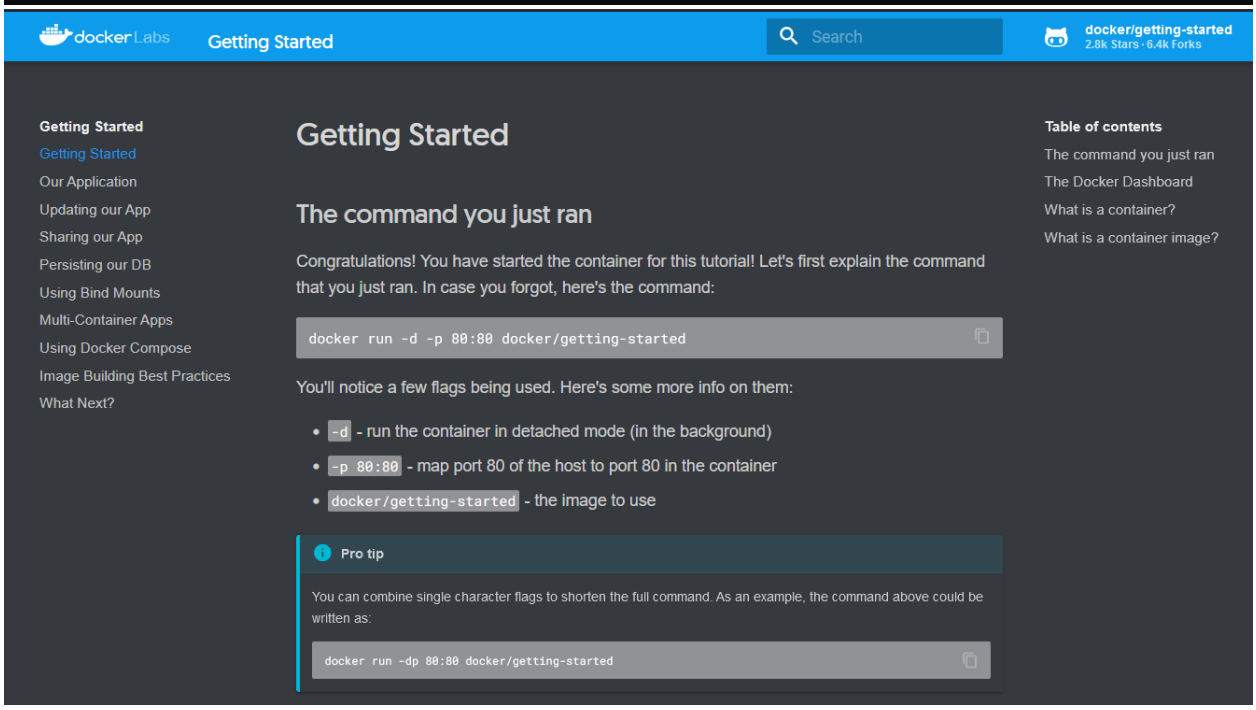
Избранные главы информатики

Дудко Юрий , 253503

1. Подготовьте рабочее окружение в соответствии с типом вашей операционной системы
    - Установите Docker
    - Выполните базовую настройку

2. Изучите простейшие консольные команды и возможности Docker Desktop (см. лекцию), создать собственный контейнер docker/getting-started, открыть в браузере и изучить tutorial

```
C:\Users\Юрий>docker run -d -p 80:80 docker/getting-started
Unable to find image 'docker/getting-started:latest' locally
latest: Pulling from docker/getting-started
c158987b0551: Pull complete
1e35f6679fab: Pull complete
cb9626c74200: Pull complete
b6334b6ace34: Pull complete
f1d1c9928c82: Pull complete
9b6f639ec6ea: Pull complete
ee68d3549ec8: Pull complete
33e0cbbb4673: Pull complete
4f7e34c2de10: Pull complete
Digest: sha256:d79336f4812b6547a53e735480dde67f8f8f7071b414fbd9297609ffb989abc1
Status: Downloaded newer image for docker/getting-started:latest
ab2eadae2a913cae7395c27d08d80cc63f178e94c287330909f7c02d8a3b13d5
```



3. Создайте docker image, который запускает скрипт с использованием функций из https://github.com/smartiqaorg/geometric_lib.

a. Данные необходимые для работы скрипта передайте любым удобным способом (например: конфиг файл через docker volume, переменные окружения, перенаправление ввода). Изучите простейшие консольные команды для работы с docker(см. лекцию). Зарегистрируйтесь на DockerHub и выберите необходимые для проекта образы

```
C:\Users\Юрий>docker pull python:3.12-alpine
3.12-alpine: Pulling from library/python
4abcf2066143: Pull complete
dca80dc46cec: Pull complete
fe9e15b6315c: Pull complete
a8fd6f3f484f: Pull complete
4fc96b5c1ba4: Pull complete
Digest: sha256:1a0501213b470de000d8432b3caab9d8de5489e9443c2cc7ccaa6b0aa5c3148e
Status: Downloaded newer image for python:3.12-alpine
docker.io/library/python:3.12-alpine
```

```
C:\Users\Юрий>docker create --name SmartiqaLib -w /app/ -it python:3.12-alpine
dbfd870353423fc6e1b4bf9d0dbc6090f14b6488a83de133b87d6b37526179dc
```

```
C:\Users\Юрий>docker cp 253503_DUDKO_10/IGI/LR2/geometric_lib SmartiqaLib:/app/
Successfully copied 70.7kB to SmartiqaLib:/app/
```

```
C:\Users\Юрий>docker start -i SmartiqaLib
Python 3.12.2 (main, Feb  7 2024, 22:13:24) [GCC 13.2.1 20231014] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.environ['workdir'] = '/app/'
>>> import geometric_lib.circle
>>> geometric_lib.circle.area(8)
201.06192982974676
```

b. Создать Dockerfile для реализации сборки собственных Docker образов

```
Dockerfile > ...
1   FROM python:3.12-alpine
2
3   WORKDIR /app/
4   COPY geometric_lib /app/
5
6   ENV WORKDIR /app/
```

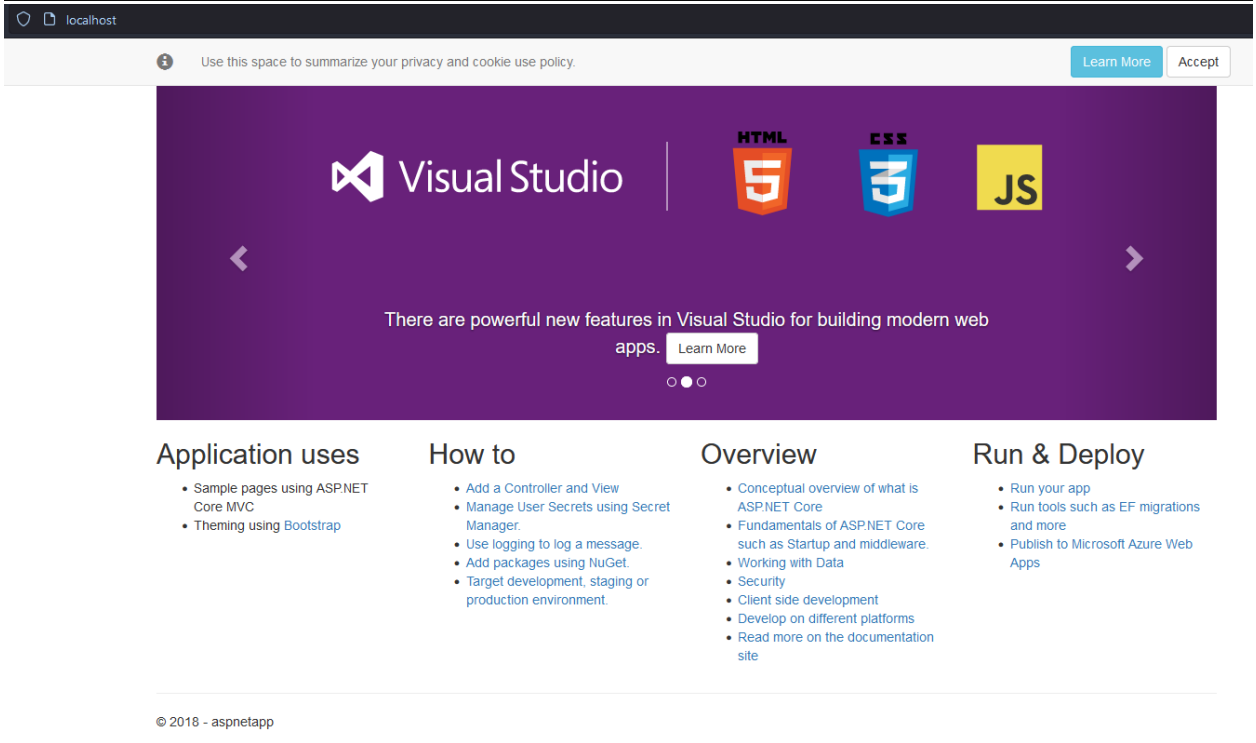
c. Использовать его для создания контейнера. Протестировать использование контейнера

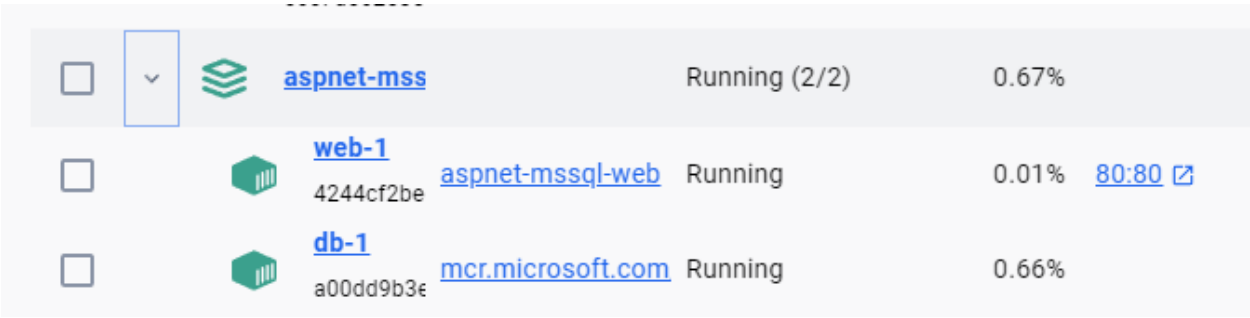
```
C:\Users\Юрий\253503_DUDKO_10\IGI\LR2>docker build -t new_smartiqa .
[+] Building 0.7s (8/8) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 124B
 => [internal] load metadata for docker.io/library/python:3.12-alpine
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/3] FROM docker.io/library/python:3.12-alpine
 => [internal] load build context
 => => transferring context: 38.15kB
 => [2/3] WORKDIR /app/
 => [3/3] COPY geometric_lib /app/
 => exporting to image
 => => exporting layers
 => => writing image sha256:6aee268a4bbde18c5e3d54b57c39e3effe0c96174f3291a78628145aca216fee
 => => naming to docker.io/library/new_smartiqa
```

```
C:\Users\Юрий>docker run -it new_smartiqa
Python 3.12.2 (main, Feb  7 2024, 22:13:24) [GCC 13.2.1 20231014] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import circle
>>> circle.area(34)
3631.6811075498013
```

4. Скачать любой доступный проект с GitHub с произвольным стеком технологий или использовать свой, ранее разработанный. Создать для него необходимый контейнер, используя Docker Compose для управления многоконтейнерными приложениями. Запустить проект в контейнере.( Примеры Images: https://hub.docker.com/_/phpmyadmin, https://hub.docker.com/_/mysql, https://hub.docker.com/_/postgres)

```
C:\Users\Юрий\253503_DUDKO_10\IGI\LR2\aspnet-mssql>docker compose up
[+] Running 13/13
 ⠿ db 12 layers [⠿⠿⠿⠿⠿⠿⠿⠿⠿⠿⠿⠿]      0B/0B      Pulled
   ⠿ 4bbfd2c87b75 Pull complete
   ⠿ d2e110be24e1 Pull complete
   ⠿ 889a7173dcfe Pull complete
[+] Building 16.9s (8/15)
 => => sha256:169ba0027942dd5a63c826f18de6c7f110c41cc350c0a405e46a5a2cff8c11c9 17.09MB / 17.09MB
 => => extracting sha256:c1ad9731b2c7bf7fddea67f2f3f553515179a375c489e591e2372700fcaca766
 => => sha256:e76245086e243c3e2e5b8571c00fb5a7911d15bc7665e567b3c50208e24d864a 154B / 154B
[+] Building 17.0s (8/15)
 => => sha256:169ba0027942dd5a63c826f18de6c7f110c41cc350c0a405e46a5a2cff8c11c9 17.09MB / 17.09MB
[+] Building 51.6s (16/16) FINISHED
 => [web internal] load build definition from Dockerfile
 => => transferring dockerfile: 453B
 => [web internal] load metadata for mcr.microsoft.com/dotnet/sdk:5.0
 => [web internal] load metadata for mcr.microsoft.com/dotnet/aspnet:5.0
 => [web internal] load .dockerignore
 => => transferring context: 2B
 => [web internal] load build context
 => => transferring context: 2.39MB
 => [web base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:5.0@sha256:1a7d811242f001673d5d25283b3af03da5
 => => writing image sha256:30df33b89d1c7c3f0879a9c30bde0b84b4d2755be8bcafd10aa04f7f538b1057
 => => naming to docker.io/library/aspnet-mssql-web
[+] Running 3/3
 ⠿ Network aspnet-mssql_default  Created
 ⠿ Container aspnet-mssql-web-1  Created
 ⠿ Container aspnet-mssql-db-1   Created
Attaching to db-1, web-1
db-1    | Azure SQL Edge will run as non-root by default.
db-1    | This container is running as user mssql.
db-1    | To learn more visit https://go.microsoft.com/fwlink/?linkid=2140520.
db-1    | 2024/03/10 20:28:22 [launchpadd] INFO: Extensibility Log Header: <timestamp> <process> <sandboxId> <s
<message>
db-1    | 2024/03/10 20:28:22 [launchpadd] WARNING: Failed to load /var/opt/mssql/mssql.conf ini file with erro
r/opt/mssql/mssql.conf: no such file or directory
db-1    | 2024/03/10 20:28:22 [launchpadd] INFO: DataDirectories =  /bin:/etc:/lib:/lib32:/lib64:/sbin:/usr/bin
ude:/usr/lib:/usr/lib32:/usr/lib64:/usr/libexec/gcc:/usr/sbin:/usr/share:/var/lib:/opt/microsoft:/opt/mssql-ex
y:/opt/mssql/mlservices:/opt/mssql/lib/zulu-jre-11:/opt/mssql-tools
db-1    | 2024/03/10 20:28:22 Drop permitted effective capabilities.
Error response from daemon: driver failed programming external connectivity on endpoint aspnet-mssql-web-1 (f6
bf67563cd42b0eba05a070394dd1516423b40c9bd2ebdf4b79ae): Bind for 0.0.0.0:80 failed: port is already allocated
```






5. Настроить сети и тома для обеспечения связи между контейнерами и сохранения данных (исходные данные, логин, пароль и т.д.)
6. Разместите результат в созданный репозиторий в DockerHub

```
C:\Users\Юрий\253503_DUDKO_10\IGI\LR2\aspnet-mssql>docker tag 50df53b8961c yurydudko/repos_igi_lr2:web

C:\Users\Юрий\253503_DUDKO_10\IGI\LR2\aspnet-mssql>docker tag 2bc005015176 yurydudko/repos_igi_lr2:new_db
```

```
C:\Users\Юрий\253503_DUDKO_10\IGI\LR2\aspnet-mssql>docker push yurydudko/repos_igi_lr2:new_db
The push refers to repository [docker.io/yurydudko/repos_igi_lr2]
9edb282b6b0e: Pushed
d40fb5ea5f90: Pushed
6862804baa70: Pushed
5c5aff8bf695: Pushed
82518530bf23: Pushed
eb8e852e87ec: Pushed
8327dbdeff3d: Pushed
0bd0637ab484: Pushed
2ed7a1c8bda3: Pushed
5f08512fd434: Pushed
c7bb31fc0e08: Pushed
50858308da3d: Pushed
new_db: digest: sha256:609c34af726cc7560525c7d7b487f1d4a9b683f309c9e7c158caec6114829d5c size: 2838
```

```
C:\Users\Юрий\253503_DUDKO_10\IGI\LR2\aspnet-mssql>docker push yurydudko/repos_igi_lr2:web
The push refers to repository [docker.io/yurydudko/repos_igi_lr2]
2f9e5d185168: Pushed
5f70bf18a086: Pushed
60cd9a66537a: Pushed
c17c60266ba3: Pushed
4bb9d2f55486: Pushed
dbcc5783df1a: Pushed
fe05af3bb097: Pushed
10e6bc6fdee2: Pushed
web: digest: sha256:0998226e4ab847bb37f6709d6531f4e583f4c4faf729e4a7a46a25c2c39e9f28 size: 1993
```


7. Выполните следующие действия с целью изучить особенности сетевого взаимодействия:

- Получить информацию о всех сетях, работающих на текущем хосте и подробности о каждом типе сети

```
C:\Users\Юрий>docker network ls
NETWORK ID     NAME                    DRIVER    SCOPE
e19d90ac35c5   aspnet-mssql_default    bridge    local
737800c53c07   bridge                  bridge    local
79d5bae73264   host                    host      local
8d397be4f68a   none                    null      local
```

```
C:\Users\Юрий>docker network inspect e19d90ac35c5
[
    {
        "Name": "aspnet-mssql_default",
        "Id": "e19d90ac35c52d93eeb5292dec3d46488f873a32cb7927fb7b867b16491eab1d",
        "Created": "2024-03-10T21:05:01.331446244Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.20.0.0/16",
                    "Gateway": "172.20.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "4244cf2bedc119db19dfa4ebe7cf3b02ef9a9c9fc3804ea8b00eb84f08c48d3a": {
                "Name": "aspnet-mssql-web-1",
        "ConfigOnly": false,
        "Containers": {
            "4244cf2bedc119db19dfa4ebe7cf3b02ef9a9c9fc3804ea8b00eb84f08c48d3a": {
                "Name": "aspnet-mssql-web-1",
                "EndpointID": "59c1d69504b34c1b854a84865822aba335e6a10f811e69552cba0b6211cf0beb",
                "MacAddress": "02:42:ac:14:00:02",
                "IPv4Address": "172.20.0.2/16",
                "IPv6Address": ""
            },
            "a00dd9b3e0fe21db2fb0a4f0ab1845b417ab5ee49111e77c4fc69cc76bc34c9b": {
                "Name": "aspnet-mssql-db-1",
                "EndpointID": "874ba591a5f8ed844d627f5f9dbcd8627e85200f4de3bfaa23a301393ba02789",
                "MacAddress": "02:42:ac:14:00:03",
                "IPv4Address": "172.20.0.3/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {
            "com.docker.compose.network": "default",
            "com.docker.compose.project": "aspnet-mssql",
            "com.docker.compose.version": "2.24.6"
        }
    }
]
```

- Создать свою собственную сеть bridge, проверить, создана ли она, запустить Docker-контейнер в созданной сети, вывести о ней всю информацию(включая IP-адрес контейнера), отключить сеть от контейнера

```
C:\Users\Юрий>docker network create --driver bridge lr_test_network
699dc800ad824e78742ceb60d8f111dd2b174e8e3085170a66aadfa08e0bb476

C:\Users\Юрий>docker network ls
NETWORK ID     NAME                   DRIVER    SCOPE
e19d90ac35c5   aspnet-mssql_default   bridge    local
737800c53c07   bridge                 bridge    local
79d5bae73264   host                   host      local
699dc800ad82   lr_test_network        bridge    local
8d397be4f68a   none                   null      local
```

```
C:\Users\Юрий>docker run --name launch_test_network -d -it --network=lr_test_network python:3.12-alpine
7be169a063ef9228b458824ca58645bffb763962d75125096fa1e597d913bd03

C:\Users\Юрий>docker network inspect lr_test_network
[
    {
        "Name": "lr_test_network",
        "Id": "699dc800ad824e78742ceb60d8f111dd2b174e8e3085170a66aadfa08e0bb476",
        "Created": "2024-03-10T21:24:16.603263613Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.21.0.0/16",
                    "Gateway": "172.21.0.1"
                }
```

```
            }
        ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
        "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
        "7be169a063ef9228b458824ca58645bffb763962d75125096fa1e597d913bd03": {
            "Name": "launch_test_network",
            "EndpointID": "ee857a17f197a7162ddc4e05d8d7b9f7a42340fab0da8b62a849c1f62950d979",
            "MacAddress": "02:42:ac:15:00:02",
            "IPv4Address": "172.21.0.2/16",
            "IPv6Address": ""
        }
    },
    "Options": {},
    "Labels": {}
}
]
```


```
C:\Users\Юрий>docker network inspect lr_test_network
[
    {
        "Name": "lr_test_network",
        "Id": "699dc800ad824e78742ceb60d8f111dd2b174e8e3085170a66aadfa08e0bb476",
        "Created": "2024-03-10T21:24:16.603263613Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.21.0.0/16",
                    "Gateway": "172.21.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
```

- Создать еще одну сеть bridge, вывести о ней всю информацию, запустить в ней три контейнера, подключиться к любому из контейнеров и пропинговать два других из оболочки контейнера, убедиться, что между контейнерами происходит общение по IP-адресу

```
C:\Users\Юрий>docker run -it -d --name=launch_test_network2 --network=lr_test_network2 alpine ash
dd73d8c82cbb90ea55476b1bef1d34616ac659cdadb2719d5b90f28534792ffb

C:\Users\Юрий>docker run -it -d --name=launch_test_network3 --network=lr_test_network2 alpine ash
705a626403e37ee4f5a764d008b7516d4bbfc2a08d13b7288392cabe4fdbed90

C:\Users\Юрий>docker run -it -d --name=launch_test_network4 --network=lr_test_network2 alpine ash
c6c7de625352523b1d00db81bcce6f0e818ac6aad1b180a13dd2a672a57e1789
```

```
C:\Users\Юрий>docker container attach launch_test_network2
/ # ping -c 2 launch_test_network3
PING launch_test_network3 (172.22.0.3): 56 data bytes
64 bytes from 172.22.0.3: seq=0 ttl=64 time=0.137 ms
64 bytes from 172.22.0.3: seq=1 ttl=64 time=0.115 ms

--- launch_test_network3 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.115/0.126/0.137 ms
/ # ping -c 2 launch_test_network4
PING launch_test_network4 (172.22.0.4): 56 data bytes
64 bytes from 172.22.0.4: seq=0 ttl=64 time=0.142 ms
64 bytes from 172.22.0.4: seq=1 ttl=64 time=0.183 ms

--- launch_test_network4 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.142/0.162/0.183 ms
/ #
```

- Создать свою собственную сеть overlay, проверить, создана ли она, вывести о ней всю информацию

```
C:\Users\Юрий>docker swarm init
Swarm initialized: current node (m6lyowjnboix26nvamv4uqxj0) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-42i8q24fmxi4q6ts9e1ef82a2yp1uecz4u5ghfjcg
92.168.65.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow


C:\Users\Юрий>docker network create --driver=overlay lr_test_overlay
hwi3yq5lcz8p72b05bozanstm

C:\Users\Юрий>docker network inspect lr_test_overlay
[
    {
        "Name": "lr_test_overlay",
        "Id": "hwi3yq5lcz8p72b05bozanstm",
        "Created": "2024-03-10T21:46:21.625773427Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "10.0.1.0/24",
                    "Gateway": "10.0.1.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": null,
        "Options": {
            "com.docker.network.driver.overlay.vxlanid_list": "4097"
        },
        "Labels": null
    }
]
```

Создать еще одну сеть overlay, проверить, создана ли она, вывести о ней всю информацию, удалить сеть

```
C:\Users\Юрий>docker network create --driver=overlay lr_test_overlay2
p2hube0bmjbt57nyouxoryhqw

C:\Users\Юрий>docker network inspect p2hube0bmjbt57nyouxoryhqw
[
    {
        "Name": "lr_test_overlay2",
        "Id": "p2hube0bmjbt57nyouxoryhqw",
        "Created": "2024-03-10T21:47:41.4757012717Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "10.0.2.0/24",
                    "Gateway": "10.0.2.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": null,
        "Options": {
            "com.docker.network.driver.overlay.vxlanid_list": "4098"
        },
        "Labels": null
    }
]

C:\Users\Юрий>docker network rm p2hube0bmjbt57nyouxoryhqw
p2hube0bmjbt57nyouxoryhqw

C:\Users\Юрий>docker network ls
NETWORK ID     NAME                   DRIVER    SCOPE
e19d90ac35c5   aspnet-mssql_default   bridge    local
737800c53c07   bridge                 bridge    local
ad128b504a1d   docker_gwbridge        bridge    local
79d5bae73264   host                   host      local
uxjy8ugvwoak   ingress                overlay   swarm
699dc800ad82   lr_test_network        bridge    local
531e3410aa52   lr_test_network2       bridge    local
hwi3yq5lcz8p   lr_test_overlay        overlay   swarm
8d397be4f68a   none                   null      local
```

- Попробовать создать сеть host, сохранить результат в отчет.

```
C:\Users\Юрий>docker network create --driver=host ls_test_host
Error response from daemon: only one instance of "host" network is allowed
```