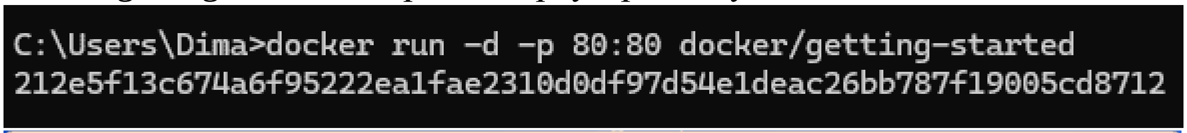
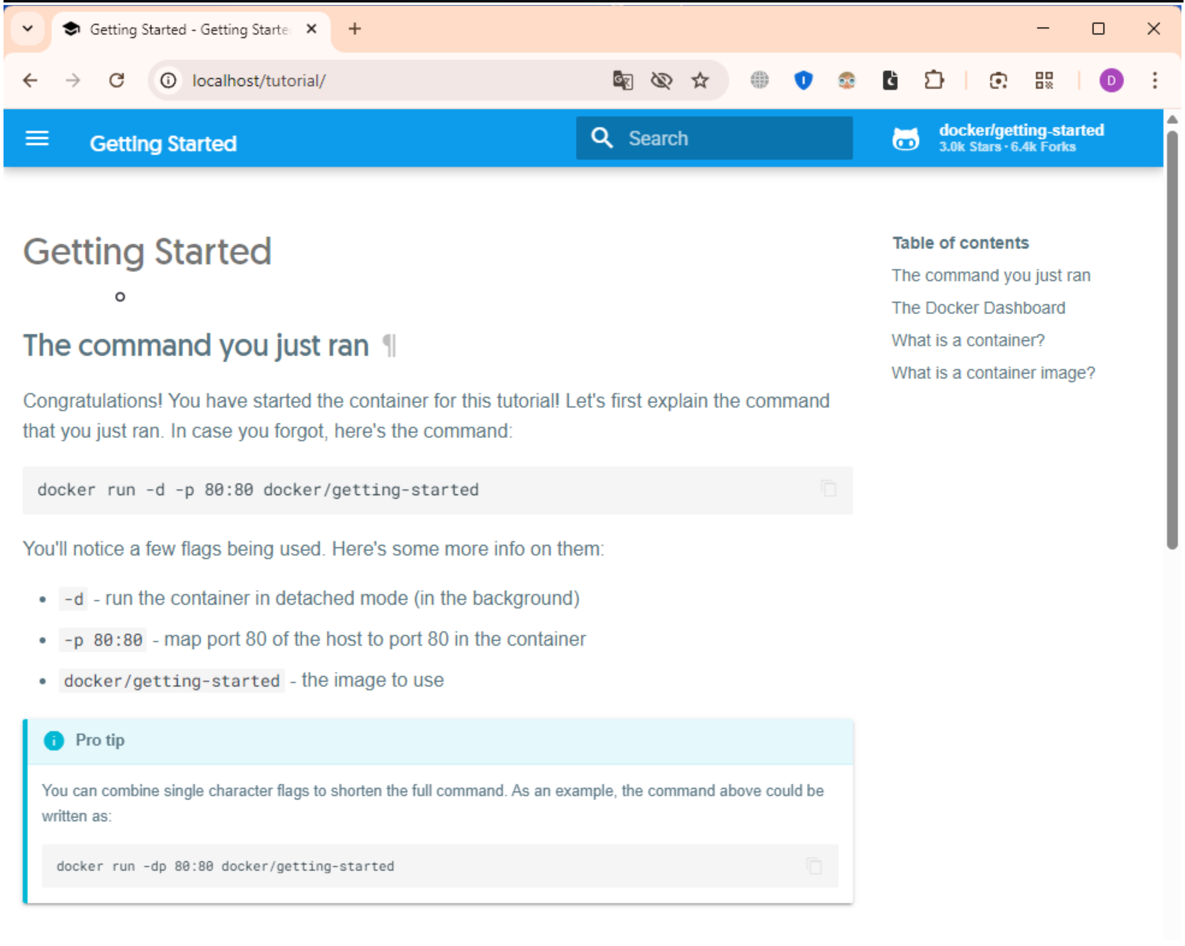
1. Изучите простейшие консольные команды и возможности Docker Desktop (см. лекцию), создать собственный контейнер docker/getting-started, открыть в браузере и изучить tutorial



```
C:\Users\Dima>docker run -d -p 80:80 docker/getting-started
212e5f13c674a6f95222ea1fae2310d0df97d54e1deac26bb787f19005cd8712
```



Getting Started

The command you just ran ¶

Congratulations! You have started the container for this tutorial! Let's first explain the command that you just ran. In case you forgot, here's the command:

```
docker run -d -p 80:80 docker/getting-started
```

You'll notice a few flags being used. Here's some more info on them:

- `-d` - run the container in detached mode (in the background)
- `-p 80:80` - map port 80 of the host to port 80 in the container
- `docker/getting-started` - the image to use

Pro tip

You can combine single character flags to shorten the full command. As an example, the command above could be written as:

```
docker run -dp 80:80 docker/getting-started
```

Table of contents

The command you just ran

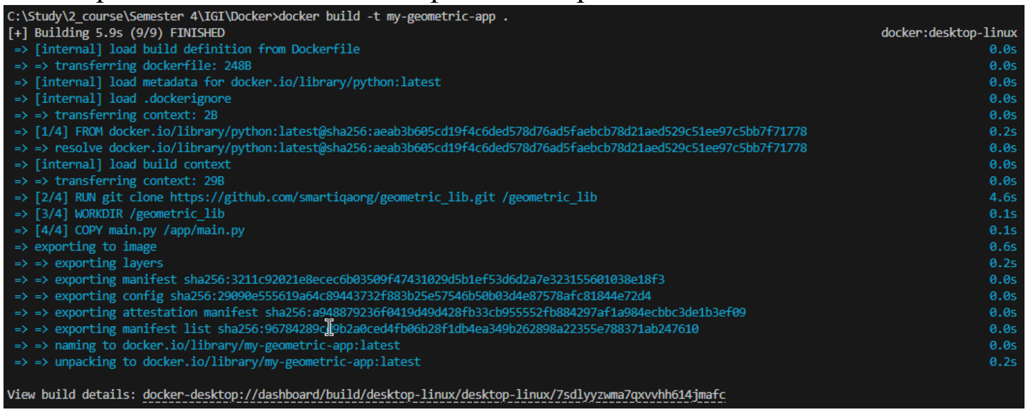
The Docker Dashboard

What is a container?

What is a container image?

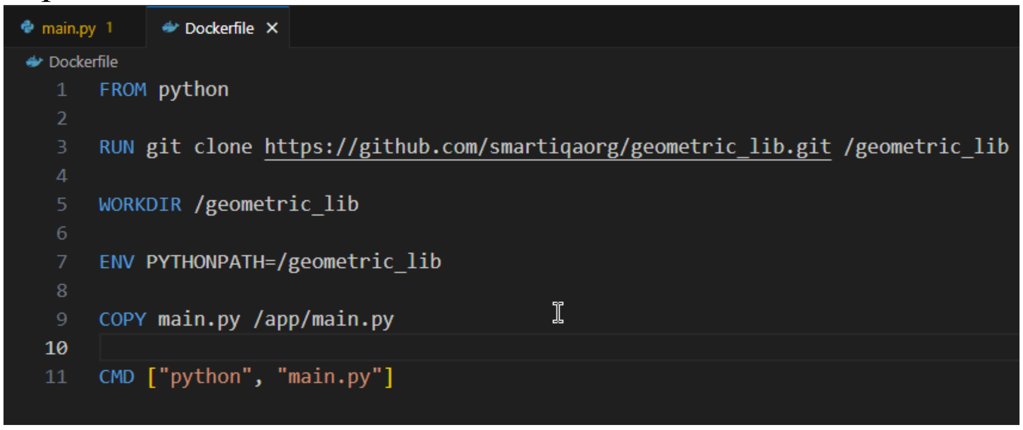
2. Создайте docker image, который запускает скрипт с использованием функций из https://github.com/smartiqaorg/geometric_lib.

   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



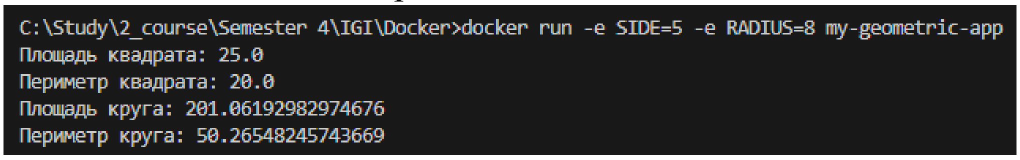
```
C:\Study\2_course\Semester 4\IGI\Docker>docker build -t my-geometric-app .
[+] Building 5.9s (9/9) FINISHED                                                                   docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                               0.0s
 => => transferring dockerfile: 248B                                                                               0.0s
 => [internal] load metadata for docker.io/library/python:latest                                                   0.0s
 => [internal] load .dockerignore                                                                                  0.0s
 => => transferring context: 2B                                                                                    0.0s
 => [1/4] FROM docker.io/library/python:latest@sha256:aeab3b605cd19f4c6ded578d76ad5faebcb78d21aed529c51ee97c5bb7f71778  0.2s
 => => resolve docker.io/library/python:latest@sha256:aeab3b605cd19f4c6ded578d76ad5faebcb78d21aed529c51ee97c5bb7f71778  0.0s
 => [internal] load build context                                                                                  0.0s
 => => transferring context: 29B                                                                                   0.0s
 => [2/4] RUN git clone https://github.com/smartiqaorg/geometric_lib.git /geometric_lib                            4.6s
 => [3/4] WORKDIR /geometric_lib                                                                                   0.1s
 => [4/4] COPY main.py /app/main.py                                                                                0.1s
 => exporting to image                                                                                             0.6s
 => => exporting layers                                                                                            0.2s
 => => exporting manifest sha256:3211c92021e8ecec6b03509f47431029d5b1ef53d6d2a7e323155601038e18f3                  0.0s
 => => exporting config sha256:29090e555619a64c89443732f883b25e57546b50b03d4e87578afc81844e72d4                    0.0s
 => => exporting attestation manifest sha256:a948879236f0419d49d428fb33cb955552fb884297af1a984ecbbc3de1b3ef09      0.0s
 => => exporting manifest list sha256:96784289c[illegible]9b2a0ced4fb06b28f1db4ea349b262898a22355e788371ab247610     0.0s
 => => naming to docker.io/library/my-geometric-app:latest                                                         0.0s
 => => unpacking to docker.io/library/my-geometric-app:latest                                                      0.2s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/7sdlyyzwma7qxvvhh614jmafc
```

b. Создать Dockerfile для реализации сборки собственных Docker образов



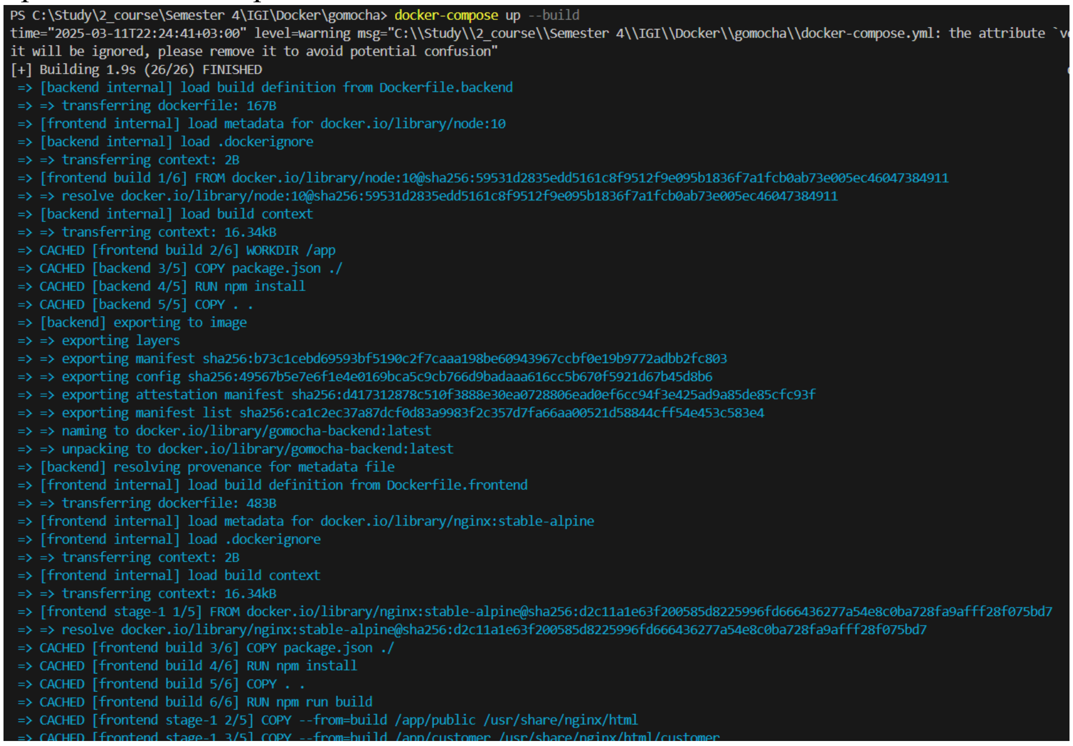
```
main.py 1    Dockerfile ×
Dockerfile
FROM python

RUN git clone https://github.com/smartiqaorg/geometric_lib.git /geometric_lib

WORKDIR /geometric_lib

ENV PYTHONPATH=/geometric_lib

COPY main.py /app/main.py


CMD ["python", "main.py"]
```

c. Использовать его для создания контейнера. Протестировать использование контейнера.

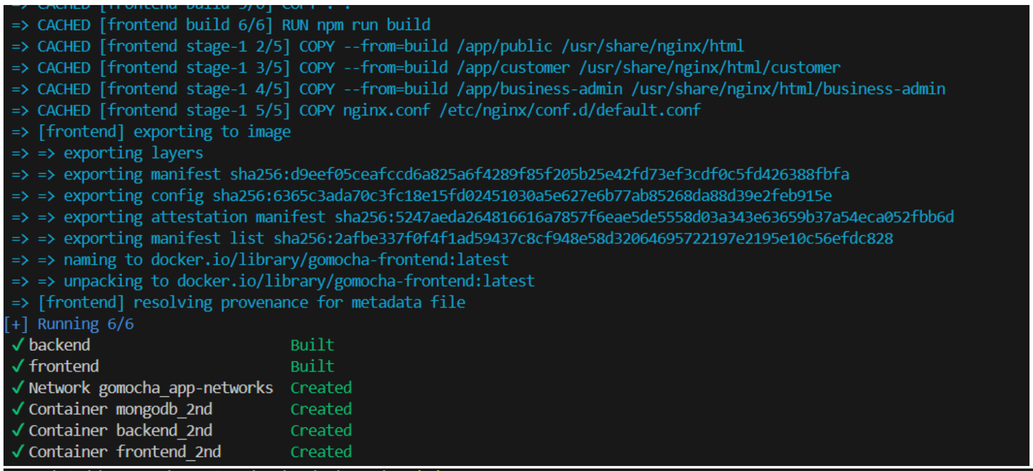


```
C:\Study\2_course\Semester 4\IGI\Docker>docker run -e SIDE=5 -e RADIUS=8 my-geometric-app
Площадь квадрата: 25.0
Периметр квадрата: 20.0
Площадь круга: 201.06192982974676
Периметр круга: 50.26548245743669
```

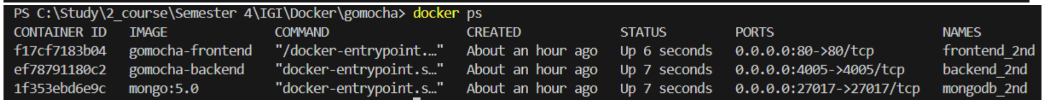
3. Скачать любой доступный проект с GitHub с произвольным стеком технологий (пример – см. индивидуальное задание) или использовать свой, ранее разработанный. Создать для него необходимый контейнер, используя Docker Compose для управления многоконтейнерными приложениями. Запустить проект в контейнере



```
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker-compose up --build
time="2025-03-11T22:24:41+03:00" level=warning msg="C:\\Study\\2_course\\Semester 4\\IGI\\Docker\\gomocha\\docker-compose.yml: the attribute `v
it will be ignored, please remove it to avoid potential confusion"
[+] Building 1.9s (26/26) FINISHED
 => [backend internal] load build definition from Dockerfile.backend
 => => transferring dockerfile: 167B
 => [frontend internal] load metadata for docker.io/library/node:10
 => [backend internal] load .dockerignore
 => => transferring context: 2B
 => [frontend build 1/6] FROM docker.io/library/node:10@sha256:59531d2835edd5161c8f9512f9e095b1836f7a1fcb0ab73e005ec46047384911
 => => resolve docker.io/library/node:10@sha256:59531d2835edd5161c8f9512f9e095b1836f7a1fcb0ab73e005ec46047384911
 => [backend internal] load build context
 => => transferring context: 16.34kB
 => CACHED [frontend build 2/6] WORKDIR /app
 => CACHED [backend 3/5] COPY package.json ./
 => CACHED [backend 4/5] RUN npm install
 => CACHED [backend 5/5] COPY . .
 => [backend] exporting to image
 => => exporting layers
 => => exporting manifest sha256:b73c1cebd69593bf5190c2f7caaa198be60943967ccbf0e19b9772adbb2fc803
 => => exporting config sha256:49567b5e7e6f1e4e0169bca5c9cb766d9badaaa616cc5b670f5921d67b45d8b6
 => => exporting attestation manifest sha256:d417312878c510f3888e30ea0728806ead0ef6cc94f3e425ad9a85de85cfc93f
 => => exporting manifest list sha256:ca1c2ec37a87dcf0d83a9983f2c357d7fa66aa00521d58844cff54e453c583e4
 => => naming to docker.io/library/gomocha-backend:latest
 => => unpacking to docker.io/library/gomocha-backend:latest
 => [backend] resolving provenance for metadata file
 => [frontend internal] load build definition from Dockerfile.frontend
 => => transferring dockerfile: 483B
 => [frontend internal] load metadata for docker.io/library/nginx:stable-alpine
 => [frontend internal] load .dockerignore
 => => transferring context: 2B
 => [frontend internal] load build context
 => => transferring context: 16.34kB
 => [frontend stage-1 1/5] FROM docker.io/library/nginx:stable-alpine@sha256:d2c11a1e63f200585d8225996fd666436277a54e8c0ba728fa9afff28f075bd7
 => => resolve docker.io/library/nginx:stable-alpine@sha256:d2c11a1e63f200585d8225996fd666436277a54e8c0ba728fa9afff28f075bd7
 => CACHED [frontend build 3/6] COPY package.json ./
 => CACHED [frontend build 4/6] RUN npm install
 => CACHED [frontend build 5/6] COPY . .
 => CACHED [frontend build 6/6] RUN npm run build
 => CACHED [frontend stage-1 2/5] COPY --from=build /app/public /usr/share/nginx/html
 => CACHED [frontend stage-1 3/5] COPY --from=build /app/customer /usr/share/nginx/html/customer
```

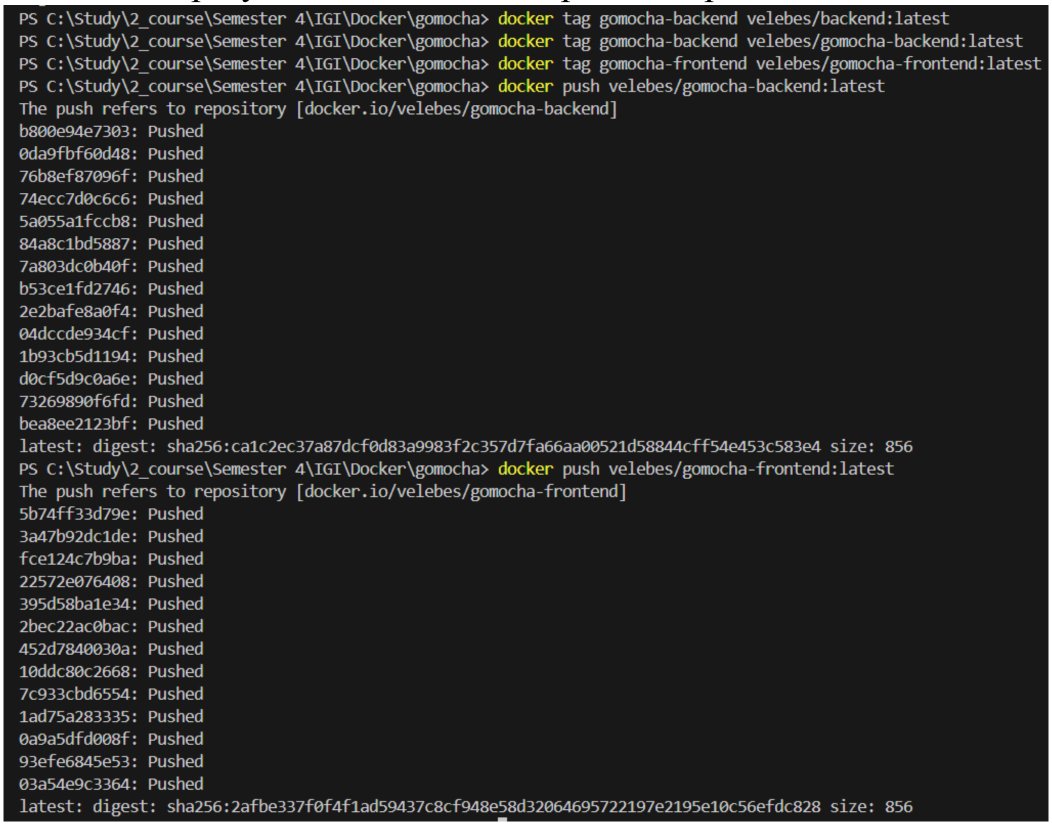


```
=> CACHED [frontend build 6/6] RUN npm run build
=> CACHED [frontend stage-1 2/5] COPY --from=build /app/public /usr/share/nginx/html
=> CACHED [frontend stage-1 3/5] COPY --from=build /app/customer /usr/share/nginx/html/customer
=> CACHED [frontend stage-1 4/5] COPY --from=build /app/business-admin /usr/share/nginx/html/business-admin
=> CACHED [frontend stage-1 5/5] COPY nginx.conf /etc/nginx/conf.d/default.conf
=> [frontend] exporting to image
=> => exporting layers
=> => exporting manifest sha256:d9eef05ceafccd6a825a6f4289f85f205b25e42fd73ef3cdf0c5fd426388fbfa
=> => exporting config sha256:6365c3ada70c3fc18e15fd02451030a5e627e6b77ab85268da88d39e2feb915e
=> => exporting attestation manifest sha256:5247aeda264816616a7857f6eae5de5558d03a343e63659b37a54eca052fbb6d
=> => exporting manifest list sha256:2afbe337f0f4f1ad59437c8cf948e58d32064695722197e2195e10c56efdc828
=> => naming to docker.io/library/gomocha-frontend:latest
=> => unpacking to docker.io/library/gomocha-frontend:latest
=> [frontend] resolving provenance for metadata file
[+] Running 6/6
 ✓ backend                      Built
 ✓ frontend                     Built
 ✓ Network gomocha_app-networks Created
 ✓ Container mongodb_2nd        Created
 ✓ Container backend_2nd        Created
 ✓ Container frontend_2nd       Created
```



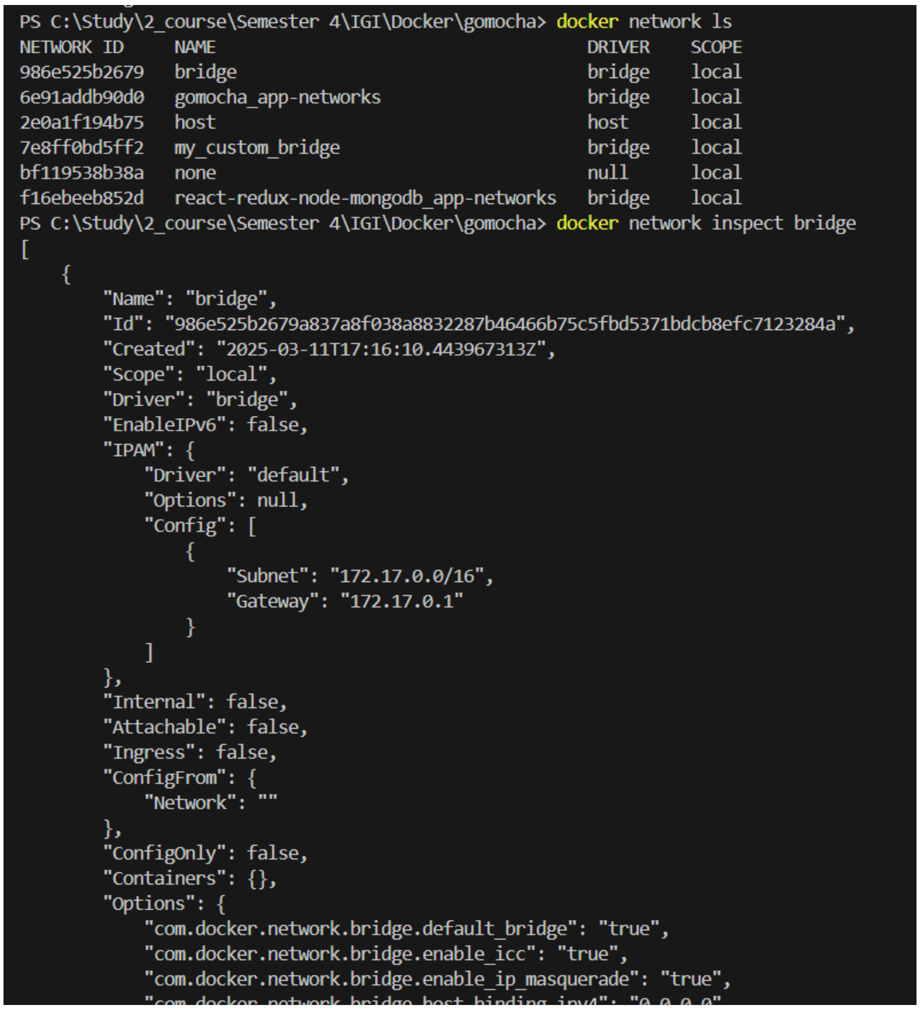
```
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED             STATUS         PORTS                      NAMES
f17cf7183b04   gomocha-frontend   "/docker-entrypoint.…"   About an hour ago   Up 6 seconds   0.0.0.0:80->80/tcp         frontend_2nd
ef78791180c2   gomocha-backend    "docker-entrypoint.s…"   About an hour ago   Up 7 seconds   0.0.0.0:4005->4005/tcp     backend_2nd
1f353ebd6e9c   mongo:5.0          "docker-entrypoint.s…"   About an hour ago   Up 7 seconds   0.0.0.0:27017->27017/tcp   mongodb_2nd
```

4. Разместите результат в созданный репозиторий в DockerHub



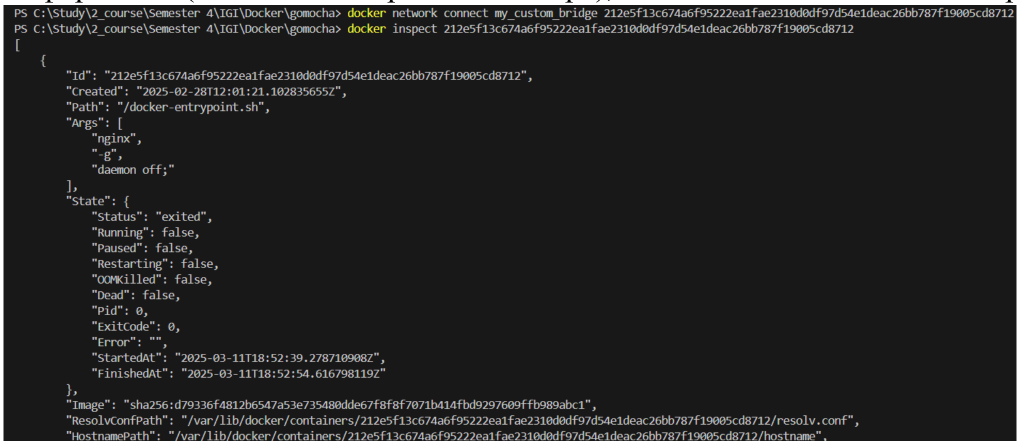
```
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker tag gomocha-backend velebes/backend:latest
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker tag gomocha-backend velebes/gomocha-backend:latest
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker tag gomocha-frontend velebes/gomocha-frontend:latest
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker push velebes/gomocha-backend:latest
The push refers to repository [docker.io/velebes/gomocha-backend]
b800e94e7303: Pushed
0da9fbf60d48: Pushed
76b8ef87096f: Pushed
74ecc7d0c6c6: Pushed
5a055a1fccb8: Pushed
84a8c1bd5887: Pushed
7a803dc0b40f: Pushed
b53ce1fd2746: Pushed
2e2bafe8a0f4: Pushed
04dccde934cf: Pushed
1b93cb5d1194: Pushed
d0cf5d9c0a6e: Pushed
73269890f6fd: Pushed
bea8ee2123bf: Pushed
latest: digest: sha256:ca1c2ec37a87dcf0d83a9983f2c357d7fa66aa00521d58844cff54e453c583e4 size: 856
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker push velebes/gomocha-frontend:latest
The push refers to repository [docker.io/velebes/gomocha-frontend]
5b74ff33d79e: Pushed
3a47b92dc1de: Pushed
fce124c7b9ba: Pushed
22572e076408: Pushed
395d58ba1e34: Pushed
2bec22ac0bac: Pushed
452d7840030a: Pushed
10ddc80c2668: Pushed
7c933cbd6554: Pushed
1ad75a283335: Pushed
0a9a5dfd008f: Pushed
93efe6845e53: Pushed
03a54e9c3364: Pushed
latest: digest: sha256:2afbe337f0f4f1ad59437c8cf948e58d32064695722197e2195e10c56efdc828 size: 856
```

5. Выполните следующие действия с целью изучить особенности сетевого взаимодействия:
   a. Получить информацию о всех сетях, работающих на текущем хосте и подробности о каждом типе сети

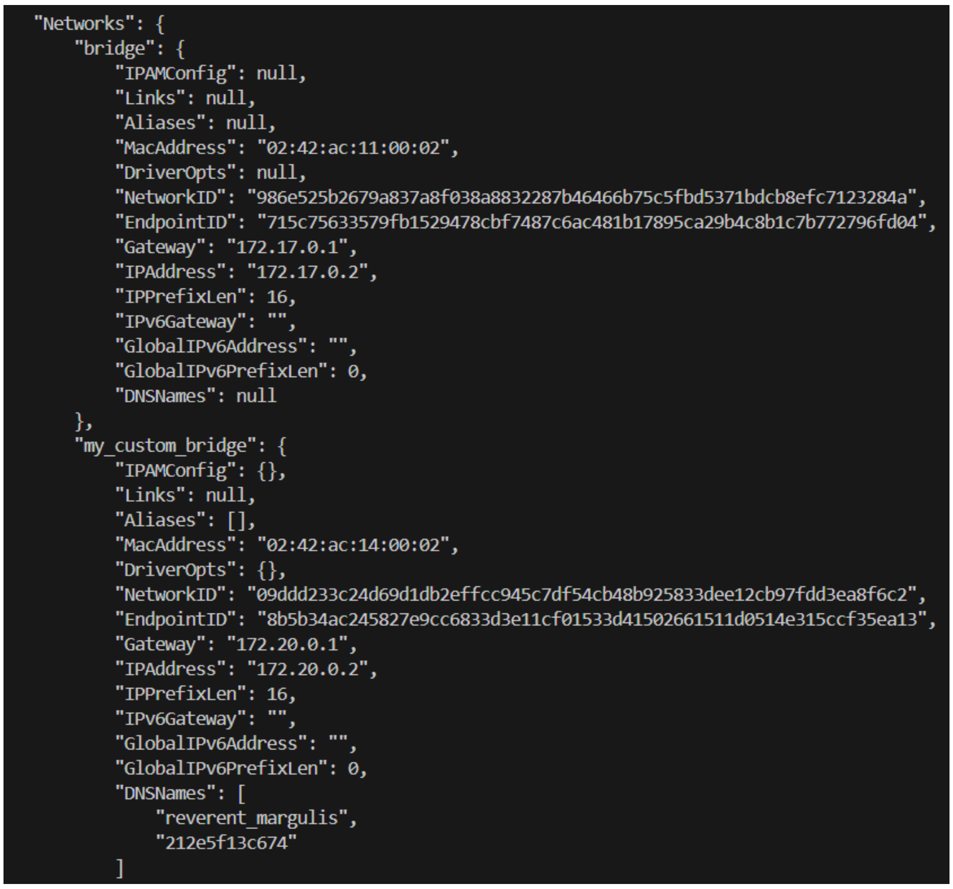


```
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker network ls
NETWORK ID     NAME                                     DRIVER    SCOPE
986e525b2679   bridge                                   bridge    local
6e91addb90d0   gomocha_app-networks                     bridge    local
2e0a1f194b75   host                                     host      local
7e8ff0bd5ff2   my_custom_bridge                         bridge    local
bf119538b38a   none                                     null      local
f16ebeeb852d   react-redux-node-mongodb_app-networks    bridge    local
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "986e525b2679a837a8f038a8832287b46466b75c5fbd5371bdcb8efc7123284a",
        "Created": "2025-03-11T17:16:10.443967313Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
```

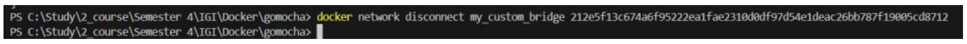
b. Создать свою собственную сеть bridge, проверить, создана ли она, запустить Docker-контейнер в созданной сети, вывести о ней всю информацию(включая IP-адрес контейнера), отключить сеть от контейнера



```
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker network connect my_custom_bridge 212e5f13c674a6f95222ea1fae2310d0df97d54e1deac26bb787f19005cd8712
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker inspect 212e5f13c674a6f95222ea1fae2310d0df97d54e1deac26bb787f19005cd8712
[
    {
        "Id": "212e5f13c674a6f95222ea1fae2310d0df97d54e1deac26bb787f19005cd8712",
        "Created": "2025-02-28T12:01:21.102835655Z",
        "Path": "/docker-entrypoint.sh",
        "Args": [
            "nginx",
            "-g",
            "daemon off;"
        ],
        "State": {
            "Status": "exited",
            "Running": false,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 0,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2025-03-11T18:52:39.278710908Z",
            "FinishedAt": "2025-03-11T18:52:54.616798119Z"
        },
        "Image": "sha256:d79336f4812b6547a53e735480dde67f8f8f7071b414fbd9297609ffb989abc1",
        "ResolvConfPath": "/var/lib/docker/containers/212e5f13c674a6f95222ea1fae2310d0df97d54e1deac26bb787f19005cd8712/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/212e5f13c674a6f95222ea1fae2310d0df97d54e1deac26bb787f19005cd8712/hostname",
```

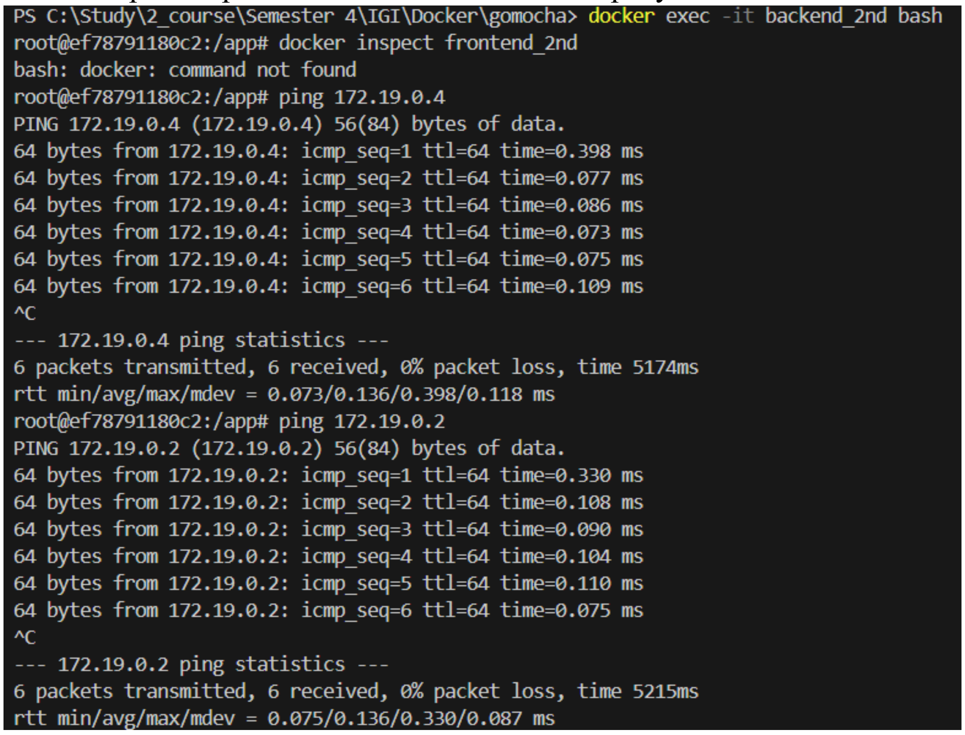


```
  "Networks": {
      "bridge": {
          "IPAMConfig": null,
          "Links": null,
          "Aliases": null,
          "MacAddress": "02:42:ac:11:00:02",
          "DriverOpts": null,
          "NetworkID": "986e525b2679a837a8f038a8832287b46466b75c5fbd5371bdcb8efc7123284a",
          "EndpointID": "715c75633579fb1529478cbf7487c6ac481b17895ca29b4c8b1c7b772796fd04",
          "Gateway": "172.17.0.1",
          "IPAddress": "172.17.0.2",
          "IPPrefixLen": 16,
          "IPv6Gateway": "",
          "GlobalIPv6Address": "",
          "GlobalIPv6PrefixLen": 0,
          "DNSNames": null
      },
      "my_custom_bridge": {
          "IPAMConfig": {},
          "Links": null,
          "Aliases": [],
          "MacAddress": "02:42:ac:14:00:02",
          "DriverOpts": {},
          "NetworkID": "09ddd233c24d69d1db2effcc945c7df54cb48b925833dee12cb97fdd3ea8f6c2",
          "EndpointID": "8b5b34ac245827e9cc6833d3e11cf01533d41502661511d0514e315ccf35ea13",
          "Gateway": "172.20.0.1",
          "IPAddress": "172.20.0.2",
          "IPPrefixLen": 16,
          "IPv6Gateway": "",
          "GlobalIPv6Address": "",
          "GlobalIPv6PrefixLen": 0,
          "DNSNames": [
              "reverent_margulis",
              "212e5f13c674"
          ]
```



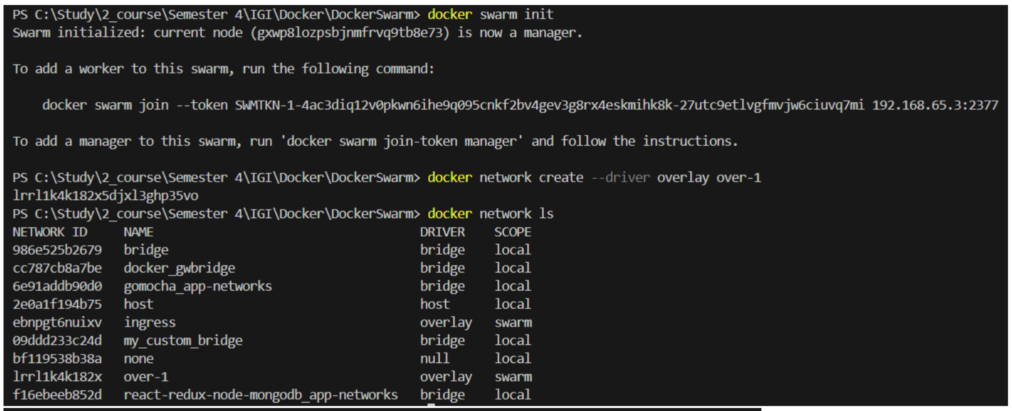
```
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker network disconnect my_custom_bridge 212e5f13c674a6f95222ea1fae2310d0df97d54e1deac26bb787f19005cd8712
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha>
```

c. 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

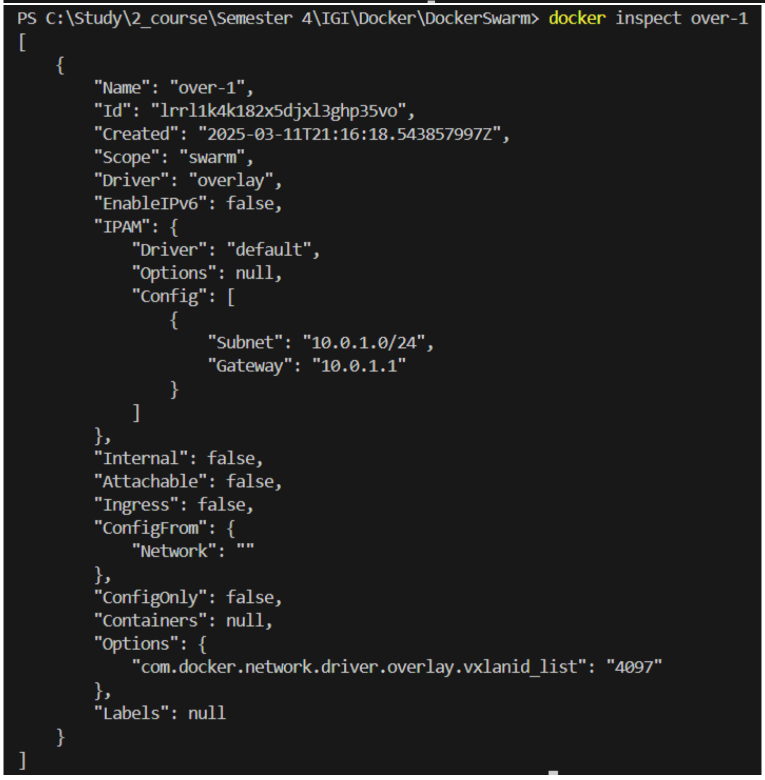


```
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker exec -it backend_2nd bash
root@ef78791180c2:/app# docker inspect frontend_2nd
bash: docker: command not found
root@ef78791180c2:/app# ping 172.19.0.4
PING 172.19.0.4 (172.19.0.4) 56(84) bytes of data.
64 bytes from 172.19.0.4: icmp_seq=1 ttl=64 time=0.398 ms
64 bytes from 172.19.0.4: icmp_seq=2 ttl=64 time=0.077 ms
64 bytes from 172.19.0.4: icmp_seq=3 ttl=64 time=0.086 ms
64 bytes from 172.19.0.4: icmp_seq=4 ttl=64 time=0.073 ms
64 bytes from 172.19.0.4: icmp_seq=5 ttl=64 time=0.075 ms
64 bytes from 172.19.0.4: icmp_seq=6 ttl=64 time=0.109 ms
^C
--- 172.19.0.4 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5174ms
rtt min/avg/max/mdev = 0.073/0.136/0.398/0.118 ms
root@ef78791180c2:/app# ping 172.19.0.2
PING 172.19.0.2 (172.19.0.2) 56(84) bytes of data.
64 bytes from 172.19.0.2: icmp_seq=1 ttl=64 time=0.330 ms
64 bytes from 172.19.0.2: icmp_seq=2 ttl=64 time=0.108 ms
64 bytes from 172.19.0.2: icmp_seq=3 ttl=64 time=0.090 ms
64 bytes from 172.19.0.2: icmp_seq=4 ttl=64 time=0.104 ms
64 bytes from 172.19.0.2: icmp_seq=5 ttl=64 time=0.110 ms
64 bytes from 172.19.0.2: icmp_seq=6 ttl=64 time=0.075 ms
^C
--- 172.19.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5215ms
rtt min/avg/max/mdev = 0.075/0.136/0.330/0.087 ms
```

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

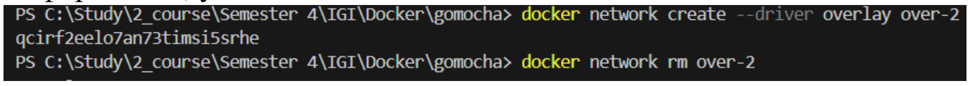


```
PS C:\Study\2_course\Semester 4\IGI\Docker\DockerSwarm> docker swarm init
Swarm initialized: current node (gxwp8lozpsbjnmfrvq9tb8e73) is now a manager.

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

    docker swarm join --token SWMTKN-1-4ac3diq12v0pkwn6ihe9q095cnkf2bv4gev3g8rx4eskmihk8k-27utc9etlvgfmvjw6ciuvq7mi 192.168.65.3:2377

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

PS C:\Study\2_course\Semester 4\IGI\Docker\DockerSwarm> docker network create --driver overlay over-1
lrrl1k4k182x5djxl3ghp35vo
PS C:\Study\2_course\Semester 4\IGI\Docker\DockerSwarm> docker network ls
NETWORK ID     NAME                                  DRIVER    SCOPE
986e525b2679   bridge                                bridge    local
cc787cb8a7be   docker_gwbridge                       bridge    local
6e91addb90d0   gomocha_app-networks                  bridge    local
2e0a1f194b75   host                                  host      local
ebnpgt6nuixv   ingress                               overlay   swarm
09ddd233c24d   my_custom_bridge                      bridge    local
bf119538b38a   none                                  null      local
lrrl1k4k182x   over-1                                overlay   swarm
f16ebeeb852d   react-redux-node-mongodb_app-networks bridge    local
```



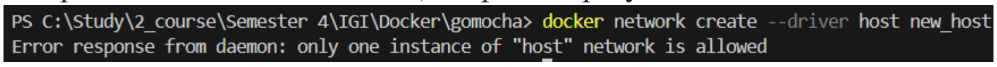
```
PS C:\Study\2_course\Semester 4\IGI\Docker\DockerSwarm> docker inspect over-1
[
    {
        "Name": "over-1",
        "Id": "lrrl1k4k182x5djxl3ghp35vo",
        "Created": "2025-03-11T21:16:18.543857997Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "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
    }
]
```

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



```
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker network create --driver overlay over-2
qcirf2eelo7an73timsi5srhe
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker network rm over-2
```

f. Попробовать создать сеть host, сохранить результат в отчет



```
PS C:\Study\2_course\Semester 4\IGI\Docker\gomocha> docker network create --driver host new_host
Error response from daemon: only one instance of "host" network is allowed
```