

Задание:

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

```
[koxan@victusMonjaro ~]$ docker --version
Docker version 28.0.0, build f9ced58158
```

Докер установлен, но не работает. Перед **первым запуском команды Docker** необходимо запустить демон docker:

```
sudo systemctl start docker.service
```

Автоматический запуск демона при загрузке системы

```
sudo systemctl enable docker.service
```

Однако по умолчанию для выполнения команд **docker** необходимо использовать **sudo**. Чтобы избежать использования **sudo** при выполнении каждой команды **docker**, можно добавить себя (или любого другого пользователя) в группу **docker** следующим образом:

```
sudo usermod -aG docker $USER
```

Чтобы вышеуказанное изменение вступило в силу, необходимо выйти из системы (или закрыть терминал) и снова войти в нее. Если вы не хотите этого делать, воспользуйтесь следующей командой:

```
newgrp docker
```

Проверим работаспособность docker

```
[koxan@victusMonjaro ~]$ docker run hello-world

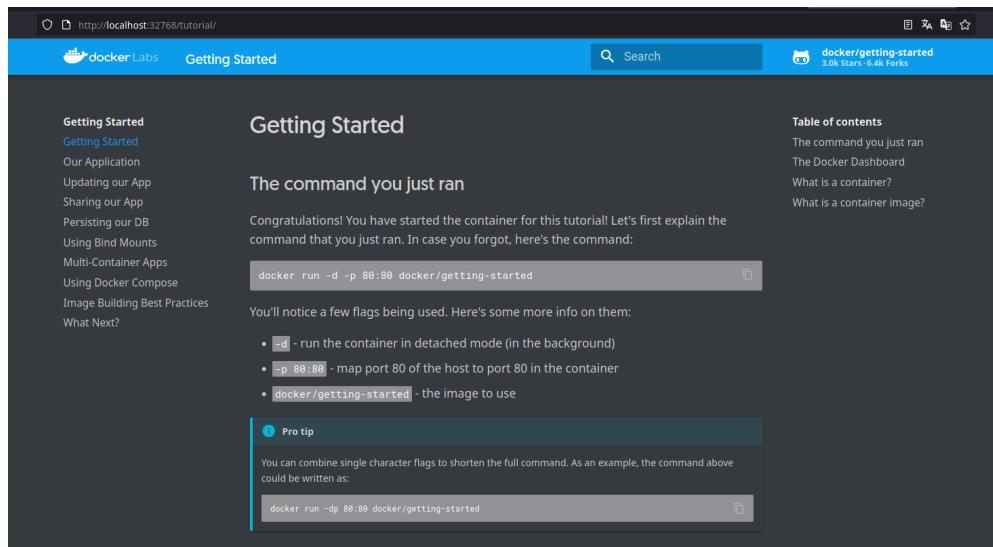
Hello from Docker!
This message shows that your installation appears to be working correctly.
```

```
[koxan@victusMonjaro ~]$ docker-compose -v
Docker Compose version 2.33.1
```

2. Создать собственный контейнер docker/getting-started, открыть в браузере и изучить tutorial

```
[koxan@victusMonjaro ~]$ docker run -d -P --name my-tutorial 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
fd1c9928c82: Pull complete
9b6f639ec6ea: Pull complete
ee68d3549ec8: Pull complete
33e0cbb4673: Pull complete
4f7e34c2de10: Pull complete
Digest: sha256:d79336f4812b6547a53e735480dde67f8f87071b414fbd9297609ffb989abc1
Status: Downloaded newer image for docker/getting-started:latest
c4b0dedb152fb99321b8358c137c9026dd92c4a438e080b21f44df5b3bed0c41
[koxan@victusMonjaro ~]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c4b0dedb152f	docker/getting-started	"/docker-entrypoint..."	7 seconds ago	Up 6 seconds	0.0.0.0:32768->80/tcp, [::]:32768->80/tcp	my-tutorial



3. Создайте docker image, который запускает скрипт с использованием функций из https://github.com/smartikaorg/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
 - b. Создать Dockerfile для реализации сборки собственных Docker образов

Ниже реализация скрипта:

```

circle.py  main.py  square.py
1  import square
2  import circle
3
4  sideSquare = float(input("Input square side: "))
5  radiusCircle = float(input("Input circle side: "))
6
7  print(f"Square area: {square.area(sideSquare)}")
8  print(f"Square perimeter: {square.perimeter(sideSquare)}")
9
10 print(f"Circle area: {circle.area(radiusCircle)}")
11 print(f"Circle perimeter: {circle.perimeter(radiusCircle)}")
12

```

Докерфайл:

```

FROM python
WORKDIR /app

COPY geometric_lib/circle.py circle.py
COPY geometric_lib/square.py square.py
COPY geometric_lib/main.py main.py

ENTRYPOINT ["python"]
CMD ["main.py"]

```

- с. Использовать его для создания контейнера. Протестировать использование контейнера

перенаправление ввода:

```
[koxan@victusMonjaro Docker]$ ls
Dockerfile  geometric_lib
[koxan@victusMonjaro Docker]$ docker build -t python_script:1 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
For more information, see https://docs.docker.com/go/buildx/
Sending build context to Docker daemon  7.61MB
Step 1/7 : FROM python
--> 36d17f72f4f3
Step 2/7 : WORKDIR /app
--> Running in 1ed62c6cebcc
--> Removed intermediate container 1ed62c6cebcc
--> 7eeadeabd38b
Step 3/7 : COPY geometric_lib/circle.py circle.py
--> fdebc23abc16
Step 4/7 : COPY geometric_lib/square.py square.py
--> f801be2b0670
Step 5/7 : COPY geometric_lib/main.py main.py
--> 3f6129387579
Step 6/7 : ENTRYPOINT ["python"]
--> Running in 7b73ca87da7b
--> Removed intermediate container 7b73ca87da7b
--> 40e59d4ac190
Step 7/7 : CMD ["main.py"]
--> Running in d1305acab156
--> Removed intermediate container d1305acab156
--> 3c44aa7675fa
Successfully built 3c44aa7675fa
Successfully tagged python_script:1
[koxan@victusMonjaro Docker]$ docker run --rm -it python_script:1
Input square side: 2
Input circle side: 2
Square area: 4.0
Square perimeter: 8.0
Circle area: 12.566370614359172
Circle perimeter: 12.566370614359172
```

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

на 3 контейнера:

```
services:
  db:
    image: redis:alpine
    command: redis-server --appendonly yes
    volumes:
      - ./dbdata:/data
    ports:
      - 6379:6379
  shortener:
    build:
      context: .
      dockerfile: shortener/Dockerfile
    environment:
      REDIS_URL: redis://db:6379
      EXPANDER_URL: http://localhost:10000
    links:
      - db
    ports:
      - 20000:20000
  expander:
    build:
      context: .
      dockerfile: expander/Dockerfile
    environment:
      REDIS_URL: redis://db:6379
      SHORTENER_URL: http://localhost:20000
    links:
      - db
    ports:
      - 10000:10000
```

Dockerfile backend:

```
FROM node:8-alpine

RUN mkdir -p /app/expander
WORKDIR /app/expander
ADD ./expander/package.json ./package.json
ADD ./expander/package-lock.json ./package-lock.json
RUN npm install

ADD ./expander .
ADD ./lib ../lib

CMD [ "npm", "start" ]

EXPOSE 10000
```

Dockerfile frontend:

```
FROM node:10-alpine

RUN npm install -g poi@9.6.8

RUN mkdir -p /app/shortener/server /app/shortener/client
ADD ./lib /app/lib

WORKDIR /app/shortener/client
ADD ./shortener/client .
RUN npm install && npm run build && rm -rf ./node_modules

WORKDIR /app/shortener/server
ADD ./shortener/server .
RUN npm install

CMD [ "npm", "start" ]

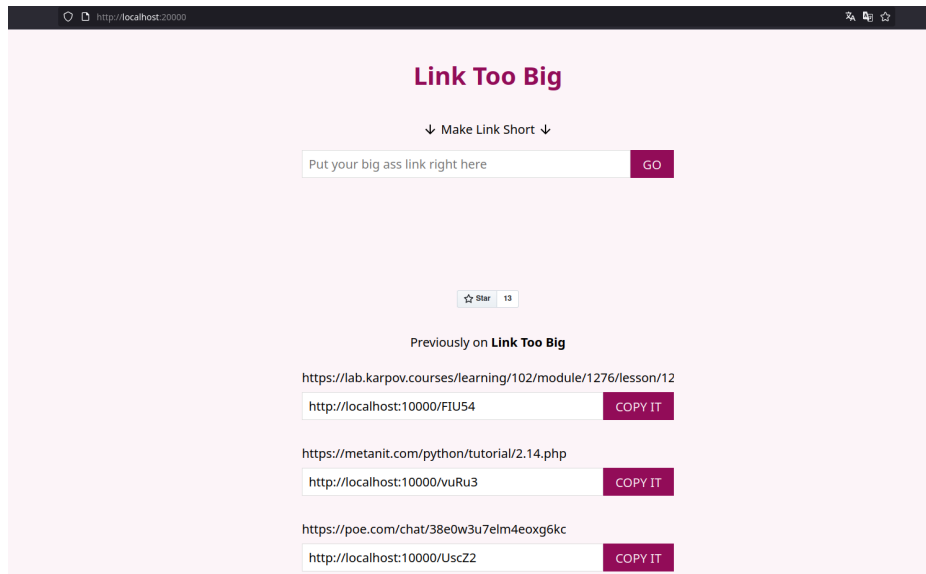
EXPOSE 20000
```

Start:

```
[koxan@victusMonjaro link-too-big]$ docker-compose up -d
[+] Running 4/4
✓ Network link-too-big_default      Created
✓ Container link-too-big-db-1       Started
✓ Container link-too-big-shortener-1 Started
✓ Container link-too-big-expander-1 Started
[koxan@victusMonjaro link-too-big]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
17532c2d05c4	link-too-big-shortener	"docker-entrypoint.s..."	10 seconds ago	Up 9 seconds	0.0.0.0:20000->20000/tcp, [::]:20000->20000/tcp	link-too-big-shortener-1
a3489da0428b	link-too-big-expander	"docker-entrypoint.s..."	10 seconds ago	Up 9 seconds	0.0.0.0:10000->10000/tcp, [::]:10000->10000/tcp	link-too-big-expander-1
adb8657942b5	redis:alpine	"docker-entrypoint.s..."	10 seconds ago	Up 9 seconds	0.0.0.0:6379->6379/tcp, [::]:6379->6379/tcp	link-too-big-db-1

Result:



```
[koxan@victusMonjaro link-too-big]$ docker-compose down
[+] Running 4/4
✓ Container link-too-big-shortener-1 Removed
✓ Container link-too-big-expander-1 Removed
✓ Container link-too-big-db-1 Removed
✓ Network link-too-big_default Removed
```

5. Настроить сети и тома для обеспечения связи между контейнерами и сохранения данных (исходные данные, логин, пароль и т.д.)

Добавим в docker-compose.yml

```
volumes:
  dbdata:
```

```
[koxan@victusMonjaro link-too-big]$ docker volume ls
DRIVER      VOLUME NAME
[koxan@victusMonjaro link-too-big]$ docker-compose up -d
[+] Running 5/5
✓ Network link-too-big_default      C...
✓ Volume "dbdata"                  Created
✓ Container link-too-big-db-1       St...
✓ Container link-too-big-expander-1 Started
✓ Container link-too-big-shortener-1 Started
```

```
[koxan@victusMonjaro link-too-big]$ docker volume ls
DRIVER      VOLUME NAME
local       link-too-big_dbdata
```

Сеть оставим default:

```
[koxan@victusMonjaro link-too-big]$ docker network ls | grep link-too-big
d01aac5108f1 link-too-big_default bridge local
```

6. Разместите результат в созданный репозиторий в DockerHub

```
[koxan@victusMonjaro link-too-big]$ docker login

USING WEB-BASED LOGIN

Info - To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: ZCPQ-JHRD
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser...

WARNING! Your credentials are stored unencrypted in '/home/koxan/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded

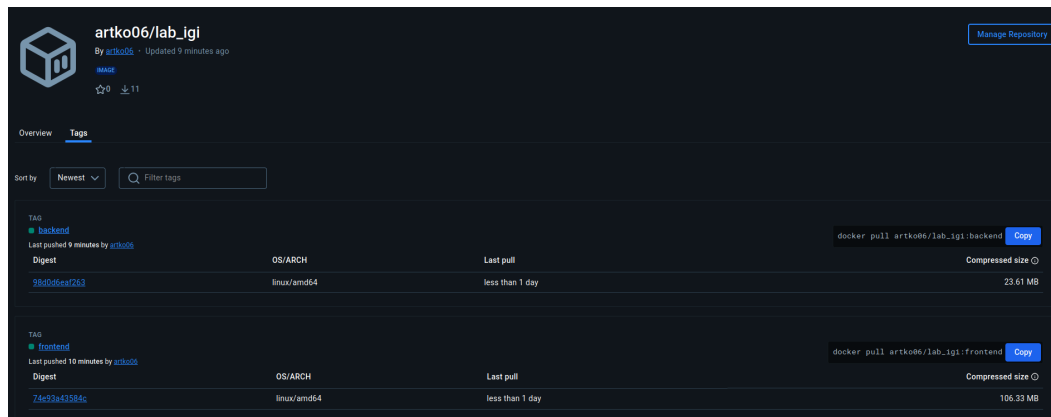
[koxan@victusMonjaro link-too-big]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
link-too-big-shortener  latest             41f936545b98       41 hours ago       319MB
link-too-big-expander  latest             8bac7a165fcf       41 hours ago       74.6MB
python_script         1                  3c44aa7675fa       4 days ago         1.02GB
nginx                 latest             b52e0b094bc0       5 weeks ago        192MB
hello-world           latest             74cc54e27dc4       7 weeks ago        10.1kB
redis                 alpine             8f5c54441eb9       2 months ago       41.4MB
python                 3.10              e83a01774710       3 months ago       1GB
docker/getting-started latest             3e4394f6b72f       2 years ago        46.9MB
wurstmeister/zookeeper latest             3f43f72cb283       6 years ago        510MB
redis                 5.0-rc-alpine     be9a7e22e598       6 years ago        40.8MB
postgres              10.5-alpine       294f651dec48       6 years ago        71.6MB
wurstmeister/kafka    1.0.0             8178c1b56170       7 years ago        272MB

[koxan@victusMonjaro link-too-big]$ docker tag link-too-big-shortener artko06/lab_igi:frontend
[koxan@victusMonjaro link-too-big]$ docker tag link-too-big-expander artko06/lab_igi:backend

[koxan@victusMonjaro link-too-big]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
artko06/lab_igi     frontend           41f936545b98       41 hours ago       319MB
link-too-big-shortener latest             41f936545b98       41 hours ago       319MB
artko06/lab_igi     backend            8bac7a165fcf       41 hours ago       74.6MB
link-too-big-expander latest             8bac7a165fcf       41 hours ago       74.6MB

[koxan@victusMonjaro link-too-big]$ docker push artko06/lab_igi:frontend
The push refers to repository [docker.io/artko06/lab_igi]
7959aba1bcfb: Pushed
be15aa465d85: Pushed
5f70bf18a086: Mounted from wurstmeister/zookeeper
dc251158d504: Pushed
4f76d8bbc951: Pushed
520832f772ff: Pushed
dd76a34d6b59: Pushed
7f6121921c6b: Pushed
365ccd918307: Mounted from library/node
1bba629c69e9: Mounted from library/node
139c1270acf1: Mounted from library/node
4693057ce236: Mounted from library/node
frontend: digest: sha256:74e93a43584ca766d7e5f474f3961f9fa2945070880db1b9c1af41c5d95104c4 size: 3032

[koxan@victusMonjaro link-too-big]$ docker push artko06/lab_igi:backend
The push refers to repository [docker.io/artko06/lab_igi]
6e242200bf9e: Pushed
b30720a6b29d: Pushed
81a2d5f2e99b: Pushed
7c5f62fb41e9: Pushed
784f341e778a: Pushed
5f70bf18a086: Layer already exists
c116dab6396c: Pushed
6b73202914ee: Mounted from library/node
9851b6482f16: Mounted from library/node
540b4f5b2c41: Mounted from library/node
6b27de954cca: Mounted from library/node
backend: digest: sha256:98d0d6eaf263aae5d410d59a90cdd00677f5363b49199f98fd56567451428cf3 size: 2610
```



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

```
[koxan@victusMonjaro ~]$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
74ef93f0b439        bridge             bridge              local
505d82fb34e7        host               host                local
c8bb6d92f8b6        none              null                local
```



```
[koxan@victusMonjaro ~]$ docker network inspect $(docker network ls -q)
```

```
[
  {
    "Name": "bridge",
    "Id": "74ef93f0b4391e9b3212f0488a7ee0b850a14a3bd2aff6be3e55283492320771",
    "Created": "2025-03-07T22:16:14.400365051+03:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "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",
      "com.docker.network.bridge.name": "docker0",
      "com.docker.network.driver.mtu": "1500"
    },
    "Labels": {}
  },
]
```

```
{
  "Name": "host",
  "Id": "505d82fb34e745c4eccc27b1787ebd44009ccb0d6af36fb4ac5bd28af62a9fa4",
  "Created": "2025-03-07T21:48:34.772922578+03:00",
  "Scope": "local",
  "Driver": "host",
  "EnableIPv4": false,
  "EnableIPv6": false,
  "IPAM": {
    "Driver": "default",
    "Options": null,
    "Config": null
  },
  "Internal": false,
  "Attachable": false,
  "Ingress": false,
  "ConfigFrom": {
    "Network": ""
  },
  "ConfigOnly": false,
  "Containers": {},
  "Options": {},
  "Labels": {}
},
]
```



```
{
  "Name": "none",
  "Id": "c8bb6d92f8b61332768fce67bd598751f541ce819029adb7cb7080deca2f6671",
  "Created": "2025-03-07T21:48:34.768652384+03:00",
  "Scope": "local",
  "Driver": "null",
  "EnableIPv4": false,
  "EnableIPv6": false,
  "IPAM": {
    "Driver": "default",
    "Options": null,
    "Config": null
  },
  "Internal": false,
  "Attachable": false,
  "Ingress": false,
  "ConfigFrom": {
    "Network": ""
  },
  "ConfigOnly": false,
  "Containers": {},
  "Options": {},
  "Labels": {}
}
```

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

```
[koxan@victusMonjaro ~]$ docker network create custom_bridge
315540480d7aac643c1b43c132d5c51cd69d1e400274ae073805085a9fe574e7
[koxan@victusMonjaro ~]$ docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
74ef93f0b439	bridge	bridge	local
315540480d7a	custom_bridge	bridge	local
505d82fb34e7	host	host	local
c8bb6d92f8b6	none	null	local

```
[koxan@victusMonjaro ~]$ docker run -d --name networkApp --net custom_bridge nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
7cf63256a31a: Pull complete
bf9acace214a: Pull complete
513c3649bb14: Pull complete
d014f92d532d: Pull complete
9dd21ad5a4a6: Pull complete
943ea0f0c2e4: Pull complete
103f50cb3e9f: Pull complete
Digest: sha256:9d6b58feebd2dbd3c56ab585333d627cc6e281011cfd6050fa4bcfd2072c9496
Status: Downloaded newer image for nginx:latest
c5e8172649a02b1a43f8dbcc07eeea8a68de9c7a8b77085230013d7546bcb295
```

```
[koxan@victusMonjaro ~]$ docker network inspect custom_bridge
[
  {
    "Name": "custom_bridge",
    "Id": "315540480d7aac643c1b43c132d5c51cd69d1e400274ae073805085a9fe574e7",
    "Created": "2025-03-11T16:12:13.173112812+03:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "c5e8172649a02b1a43f8dbcc07eeea8a68de9c7a8b77085230013d7546bcb295": {
        "Name": "networkApp",
        "EndpointID": "2effc99fafbc95f813e19e56953a8fb44277b533bcf8158dc3e9563f47ba30bf",
        "MacAddress": "8e:29:4d:e9:88:92",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
      }
    },
    "Options": {},
    "Labels": {}
  }
]
```

```
[koxan@victusMonjaro ~]$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
ac511fbcea42   nginx    "/docker-entrypoint..." 7 minutes ago  Up 7 minutes  0.0.0.0:80->80/tcp, [::]:80->80/tcp  infallible_hoover
```

```
[koxan@victusMonjaro ~]$ docker network disconnect custom_bridge ac511fbcea42
```

```
[koxan@victusMonjaro ~]$ docker network inspect custom_bridge | grep "Containers"
"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

```
[koxan@victusMonjaro ~]$ docker network create custom_bridge2
8728b96bf5adc4e35b3a1bee75bd49746eb5ca70a81bfba03ab4e90f91d912fa
```

```
[koxan@victusMonjaro ~]$ docker network inspect custom_bridge2
[
  {
    "Name": "custom_bridge2",
    "Id": "8728b96bf5adc4e35b3a1bee75bd49746eb5ca70a81bfba03ab4e90f91d912fa",
    "Created": "2025-03-11T16:57:13.572774126+03:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.19.0.0/16",
          "Gateway": "172.19.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]
```

```
[koxan@victusMonjaro ~]$ docker run -dit --rm --name nginx_1 --net custom_bridge2 -p 8000:80 nginx:latest
8e72a34a61ec1752412a800c27adcedecf375e9339bf5eb9feaab0af917a11f6
[koxan@victusMonjaro ~]$ docker run -dit --rm --name nginx_1 --net custom_bridge2 -p 8001:80 nginx:latest
Docker: Error response from daemon: Conflict. The container name "/nginx_1" is already in use by container "8e72a34a61ec1752412a800c27adcedecf375e9339bf5eb9feaab0af917a11f6". You have to remove (or rename) that container to be able to reuse that name.
```

Run 'docker run --help' for more information

```
[koxan@victusMonjaro ~]$ docker run -dit --rm --name nginx_2 --net custom_bridge2 -p 8001:80 nginx:latest
928b44786479ce6091df19f47adf601b813013d0447f74cce010f0156ed26e9d
```

```
[koxan@victusMonjaro ~]$ docker run -dit --rm --name nginx_3 --net custom_bridge2 -p 8002:80 nginx:latest
f5f3106b3ccb8f30cb74ceaa716230527b9951c2b10746ddfc1212a29daf1691
```

```
[koxan@victusMonjaro ~]$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
f5f3106b3ccb   nginx:late /docker-entrypoint..." 6 seconds ago   Up 5 seconds   0.0.0.0:8002->80/tcp, [::]:8002->80/tcp   nginx_3
928b44786479   nginx:late /docker-entrypoint..." 12 seconds ago   Up 12 seconds   0.0.0.0:8001->80/tcp, [::]:8001->80/tcp   nginx_2
8e72a34a61ec   nginx:late /docker-entrypoint..." 21 seconds ago   Up 20 seconds   0.0.0.0:8000->80/tcp, [::]:8000->80/tcp   nginx_1
```

```
[koxan@victusMonjaro ~]$ docker network inspect custom_bridge2 | grep -A 40 "Containers"
```

```
"Containers": {
  "8e72a34a61ec1752412a800c27adcedecf375e9339bf5eb9feaab0af917a11f6": {
    "Name": "nginx_1",
    "EndpointID": "9b2598cc20565a44bb949a8ef4342140a60e4ae3bdc24851fb6c133567b52ad1",
    "MacAddress": "3a:d6:3e:c3:d8:b3",
    "IPv4Address": "172.19.0.2/16",
    "IPv6Address": ""
  },
  "928b44786479ce6091df19f47adf601b813013d0447f74cce010f0156ed26e9d": {
    "Name": "nginx_2",
    "EndpointID": "a9ca36145c24dc55680cc422fecb52b1eba1a03e56d515839992577b170cea5",
    "MacAddress": "26:21:8e:df:21:86",
    "IPv4Address": "172.19.0.3/16",
    "IPv6Address": ""
  },
  "f5f3106b3ccb8f30cb74ceaa716230527b9951c2b10746ddfc1212a29daf1691": {
    "Name": "nginx_3",
    "EndpointID": "22edfc25009b45209456f115571b8982828bf3a88037b61558508e24b9961268",
    "MacAddress": "7a:89:02:98:43:70",
    "IPv4Address": "172.19.0.4/16",
    "IPv6Address": ""
  }
}
```

nginx_1 -> nginx_2(172.19.0.3) и nginx_3(172.19.0.4)

```
[koxan@victusMonjaro ~]$ docker exec -it nginx_1 bash
root@8e72a34a61ec:/# ping 172.19.0.3
PING 172.19.0.3 (172.19.0.3) 56(84) bytes of data.
64 bytes from 172.19.0.3: icmp_seq=1 ttl=64 time=0.103 ms
64 bytes from 172.19.0.3: icmp_seq=2 ttl=64 time=0.095 ms
64 bytes from 172.19.0.3: icmp_seq=3 ttl=64 time=0.104 ms
64 bytes from 172.19.0.3: icmp_seq=4 ttl=64 time=0.105 ms
64 bytes from 172.19.0.3: icmp_seq=5 ttl=64 time=0.105 ms
64 bytes from 172.19.0.3: icmp_seq=6 ttl=64 time=0.105 ms
64 bytes from 172.19.0.3: icmp_seq=7 ttl=64 time=0.096 ms
^C
--- 172.19.0.3 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6086ms
rtt min/avg/max/mdev = 0.095/0.101/0.105/0.004 ms
root@8e72a34a61ec:/# 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=0.106 ms
64 bytes from 172.19.0.4: icmp_seq=2 ttl=64 time=0.112 ms
64 bytes from 172.19.0.4: icmp_seq=3 ttl=64 time=0.119 ms
64 bytes from 172.19.0.4: icmp_seq=4 ttl=64 time=0.099 ms
64 bytes from 172.19.0.4: icmp_seq=5 ttl=64 time=0.070 ms
64 bytes from 172.19.0.4: icmp_seq=6 ttl=64 time=0.081 ms
64 bytes from 172.19.0.4: icmp_seq=7 ttl=64 time=0.111 ms
^C
--- 172.19.0.4 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6061ms
rtt min/avg/max/mdev = 0.070/0.099/0.119/0.016 ms
```

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

```
[koxan@victusMonjaro ~]$ docker swarm init
Swarm initialized: current node (mrh799vcye7jia5aryqfh03ov) is now a manager.

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

    docker swarm join --token SWMTKN-1-45pifoae0olo23btxc79uxqhxum4vs15iof4i5rt2s678ap7ze-82d9wnu231mhs1tr52r6fbx5v 192.168.0.103:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
[koxan@victusMonjaro ~]$ docker network create --driver overlay my_overlay_network
kx48rs1pil065ykape43rngtu
[koxan@victusMonjaro ~]$ docker network ls


| NETWORK ID   | NAME               | DRIVER  | SCOPE |
|--------------|--------------------|---------|-------|
| 74ef93f0b439 | bridge             | bridge  | local |
| 315540480d7a | custom_bridge      | bridge  | local |
| 8728b96bf5ad | custom_bridge2     | bridge  | local |
| 715fa1ac689c | docker_gwbridge    | bridge  | local |
| 505d82fb34e7 | host               | host    | local |
| uvrd4qc4d1x  | ingress            | overlay | swarm |
| kx48rs1pil06 | my_overlay_network | overlay | swarm |
| c8bb6d92f8b6 | none               | null    | local |


```

```
[koxan@victusMonjaro ~]$ docker network inspect my_overlay_network
[
  {
    "Name": "my_overlay_network",
    "Id": "kx48rslpi1065ykape43rngtu",
    "Created": "2025-03-11T17:13:00.427328257Z",
    "Scope": "swarm",
    "Driver": "overlay",
    "EnableIPv4": false,
    "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

```
[koxan@victusMonjaro ~]$ docker network create --driver overlay my_overlay_network2
dqi8hw6zv9f57djhjlyewdn5z
[koxan@victusMonjaro ~]$ docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
74ef93f0b439	bridge	bridge	local
315540480d7a	custom_bridge	bridge	local
8728b96bf5ad	custom_bridge2	bridge	local
715fa1ac689c	docker_gwbridge	bridge	local
505d82fb34e7	host	host	local
uvrd4qcf4dlx	ingress	overlay	swarm
kx48rslpi106	my_overlay_network	overlay	swarm
dqi8hw6zv9f5	my_overlay_network2	overlay	swarm
c8bb6d92f8b6	none	null	local

```
[koxan@victusMonjaro ~]$ docker network inspect my_overlay_network2
[
  {
    "Name": "my_overlay_network2",
    "Id": "dqi8hw6zv9f57dhjlyewdn5z",
    "Created": "2025-03-11T17:17:03.739771718Z",
    "Scope": "swarm",
    "Driver": "overlay",
    "EnableIPv4": false,
    "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
  }
]
```

```
[koxan@victusMonjaro ~]$ docker network rm my_overlay_network2
my_overlay_network2
```

```
[koxan@victusMonjaro ~]$ docker network ls
NETWORK ID        NAME                DRIVER              SCOPE
74ef93f0b439      bridge              bridge              local
315540480d7a      custom_bridge       bridge              local
8728b96bf5ad      custom_bridge2      bridge              local
715fa1ac689c      docker_gwbridge     bridge              local
505d82fb34e7      host                host                local
uvrd4qcf4dlx      ingress             overlay             swarm
kx48rslpi106      my_overlay_network  overlay             swarm
c8bb6d92f8b6      none                null                local
```

Попробовать создать сеть host, сохранить результат в отчет.

```
[koxan@victusMonjaro ~]$ docker network create --driver host my_host_network
Error response from daemon: only one instance of "host" network is allowed
```