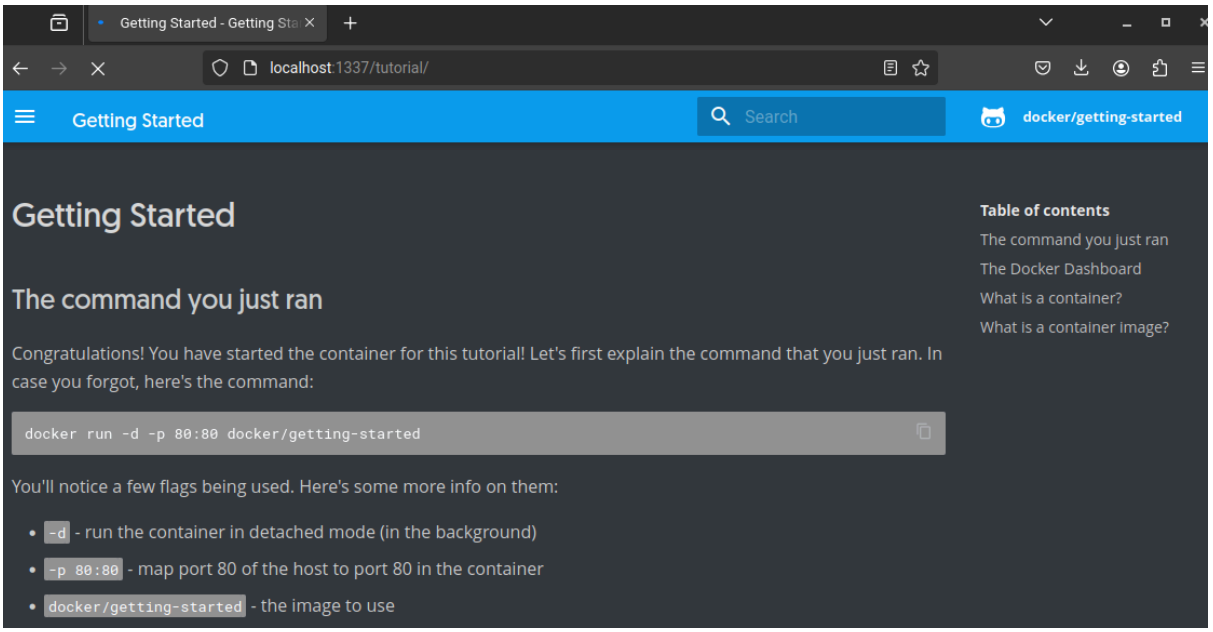
Сергеев Р. В. Лабораторная работа №2

2.

запуск контейнера

```
rs@arch ~> docker run -d -p 1337:80 docker/getting-started
1cdf1e8dca3b610e1a25b94bacc644e6dd4e0712d8daeaac68f288633caa2682
```



вывод запущенных контейнеров с указанием их размеров

```
rs@arch ~> docker ps -s
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS
          PORTS                  NAMES              SIZE
1cdf1e8dca3b   docker/getting-started   "/docker-entrypoint.…"   11 minutes ago   Up 11
minutes   0.0.0.0:1337->80/tcp   peaceful_spence    1.09kB (virtual 47MB)
```

вывод логов контейнера

```
rs@arch ~> docker logs peaceful_spence
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform confi
guration
/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/defau
lt.conf
/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
2024/03/08 14:15:22 [notice] 1#1: using the "epoll" event method
2024/03/08 14:15:22 [notice] 1#1: nginx/1.23.3
2024/03/08 14:15:22 [notice] 1#1: built by gcc 12.2.1 20220924 (Alpine 12.2.1_git2022092
4-r4)
2024/03/08 14:15:22 [notice] 1#1: OS: Linux 6.5.11-linuxkit
2024/03/08 14:15:22 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/03/08 14:15:22 [notice] 1#1: start worker processes
```

остановка контейнера

```
rs@arch ~> docker stop peaceful_spence
peaceful_spence
rs@arch ~> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
rs@arch ~> docker ps -a
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS
                    PORTS     NAMES
1cdf1e8dca3b   docker/getting-started   "/docker-entrypoint.…"   14 minutes ago   Exited
 (0) 6 seconds ago             peaceful_spence
```

запуск и приостановка работы контейнера

```
rs@arch ~> docker run -d -p 1337:80 docker/getting-started
5e5ea19f7e8a0c65a41bfea09c78a892cbe7b32d0397af59f1a45a238c2cb620
rs@arch ~> docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED         STATUS
       PORTS                  NAMES
5e5ea19f7e8a   docker/getting-started   "/docker-entrypoint.…"   3 seconds ago   Up 2 se
conds   0.0.0.0:1337->80/tcp   brave_jepsen
rs@arch ~> docker pause brave_jepsen
brave_jepsen
rs@arch ~> docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS
                   PORTS                  NAMES
5e5ea19f7e8a   docker/getting-started   "/docker-entrypoint.…"   16 seconds ago   Up 15
seconds (Paused)   0.0.0.0:1337->80/tcp   brave_jepsen
```

возобновление работы контейнера

```
rs@arch ~> docker unpause brave_jepsen
brave_jepsen
rs@arch ~> docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED              ST
ATUS              PORTS                    NAMES
5e5ea19f7e8a   docker/getting-started   "/docker-entrypoint.…"   About a minute ago   Up
 About a minute   0.0.0.0:1337->80/tcp   brave_jepsen
```

перезапуск контейнера

```
rs@arch ~ [1]> docker restart brave_jepsen
brave_jepsen
rs@arch ~> docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED         STATUS
       PORTS                  NAMES
5e5ea19f7e8a   docker/getting-started   "/docker-entrypoint.…"   2 minutes ago   Up 1 se
cond   0.0.0.0:1337->80/tcp   brave_jepsen
```

удаление контейнеров

```
rs@arch ~> docker rm brave_jepsen  peaceful_spence
brave_jepsen
peaceful_spence
rs@arch ~> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

удаление образа

```
rs@arch ~> docker images
REPOSITORY               TAG       IMAGE ID       CREATED         SIZE
docker/getting-started   latest    3e4394f6b72f   14 months ago   47MB
rs@arch ~> docker image rm 3e43
Untagged: docker/getting-started:latest
Untagged: docker/getting-started@sha256:d79336f4812b6547a53e735480dde67f8f8f7071b414fbd9
297609ffb989abc1
Deleted: sha256:3e4394f6b72fccefa2217067a7f7ff84d5d828afa9623867d68fce4f9d862b6c
Deleted: sha256:cdc6440a971be2985ce94c7e2e0c2df763b58a2ced4ecdb944fcd9b13e7a2aa4
Deleted: sha256:041ac26cd02fa81c8fd73cc616bdeee180de3fd68a649ed1c0339a84cdf7a7c3
Deleted: sha256:376baf7ada4b52ef4c110a023fe7185c4c2c090fa24a5cbd746066333ce3bc46
Deleted: sha256:d254c9b1e23bad05f5cde233b5e91153a0540fa9a797a580d8a360ad12bf63a9
Deleted: sha256:dd5c79fa9b6829fd08ff9943fc1d66bebba3e04246ba394d57c28827fed95af0
Deleted: sha256:8d812a075abf60a83013c37f49058c220c9cdf390266952126e7e60041b305dc
Deleted: sha256:ff1787ee3dcae843dc7dd1933c49350fc84451cf19ed74f4ea72426e17ee7cd1
Deleted: sha256:85ebd294be1553b207ba9120676f4fd140842348ddf1bb5f7602c7a8401f0a13
Deleted: sha256:ded7a220bb058e28ee3254fbba04ca90b6790704244247 61a53a043b93b612bf
rs@arch ~> docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
```

3

dockerfile:

```
FROM python:3

WORKDIR /usr/src/app

COPY ./script.py .

VOLUME /usr/src/app/files

ENTRYPOINT ["python", "./script.py"]
```

build:

```
rs@arch ~/d/i/d/python [1]> docker build . -t py:1
DEPRECATED: The legacy builder is deprecated and will
            Install the buildx component to build imag
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  8.704kB
Step 1/5 : FROM python:3
 ---> a3aef63c6c10
Step 2/5 : WORKDIR /usr/src/app
 ---> Using cache
 ---> ba3f1e03a96b
Step 3/5 : COPY ./script.py .
 ---> Using cache
 ---> 4d85de055afd
Step 4/5 : VOLUME /usr/src/app/files
 ---> Using cache
 ---> 3063c133373b
Step 5/5 : ENTRYPOINT ["python", "./script.py"]
 ---> Using cache
 ---> c350a6a86c85
Successfully built c350a6a86c85
Successfully tagged py:1
```

run:

```
rs@arch ~/d/i/d/python> docker run -v ./files:/usr/src/app/files -i py:1
SQUARE:
1 - area
2 - perimeter
CIRCLE:
3 - area
4 - perimeter
1
input number:
12
RESULT:
144
```

4

Dockerfile для сервера

```
FROM golang:alpine AS builder

RUN mkdir /build

WORKDIR /build

COPY . .

RUN go mod download

EXPOSE 8081

CMD go run main.go
```

Dockerfile для клиента

```
FROM node

WORKDIR /app
COPY . .

RUN npm install --silent

EXPOSE 3000

CMD npm start
```

docker-compose:

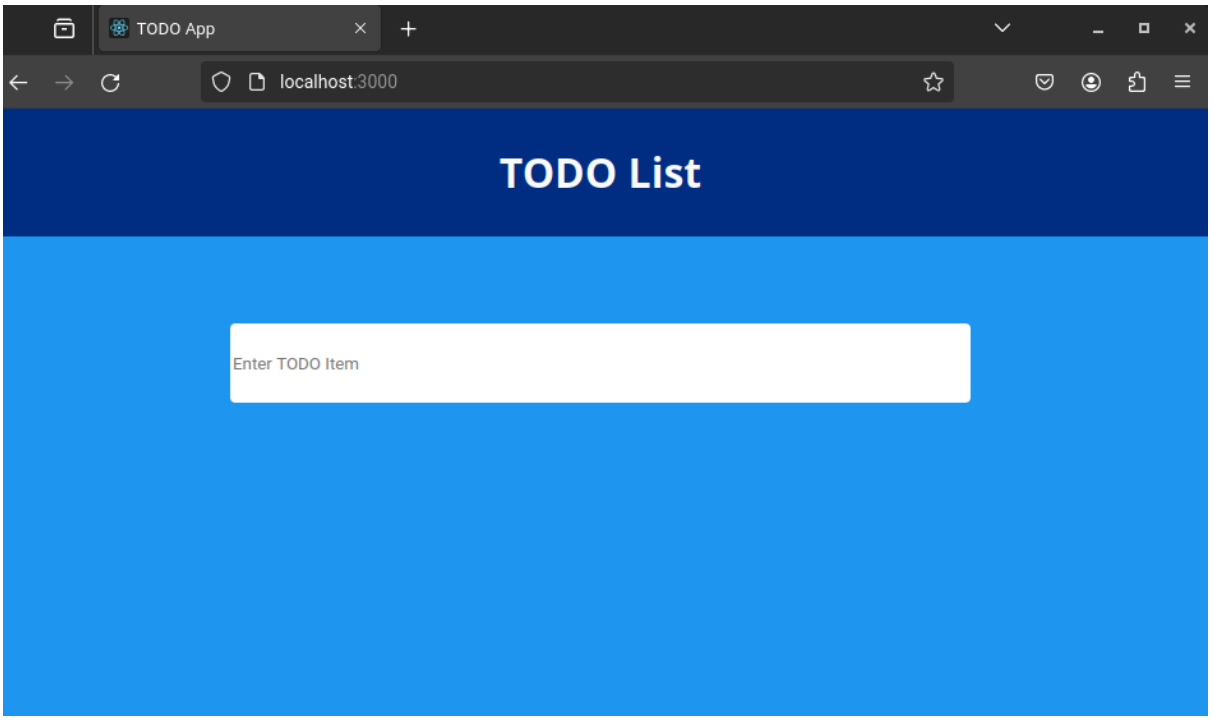
```yaml
version: "3"
services:
  go-api:
    image: rosto4eks/todo-backend:latest
    build: ./backend
    ports:
      - 8081:8081
    links:
      - postgres

  postgres:
    image: postgres
    environment:
      POSTGRES_DB: "postgres"
      POSTGRES_USER: "postgres"
      POSTGRES_PASSWORD: "password"
    volumes:
      - pg-vol:/var/lib/postgresql/data
    ports:
      - 5432:5432

  react-app:
    image: rosto4eks/todo-client:latest
    build: ./frontend
    ports:
      - 3000:3000

volumes:
    pg-vol:
      external: true
```

сборка контейнеров:

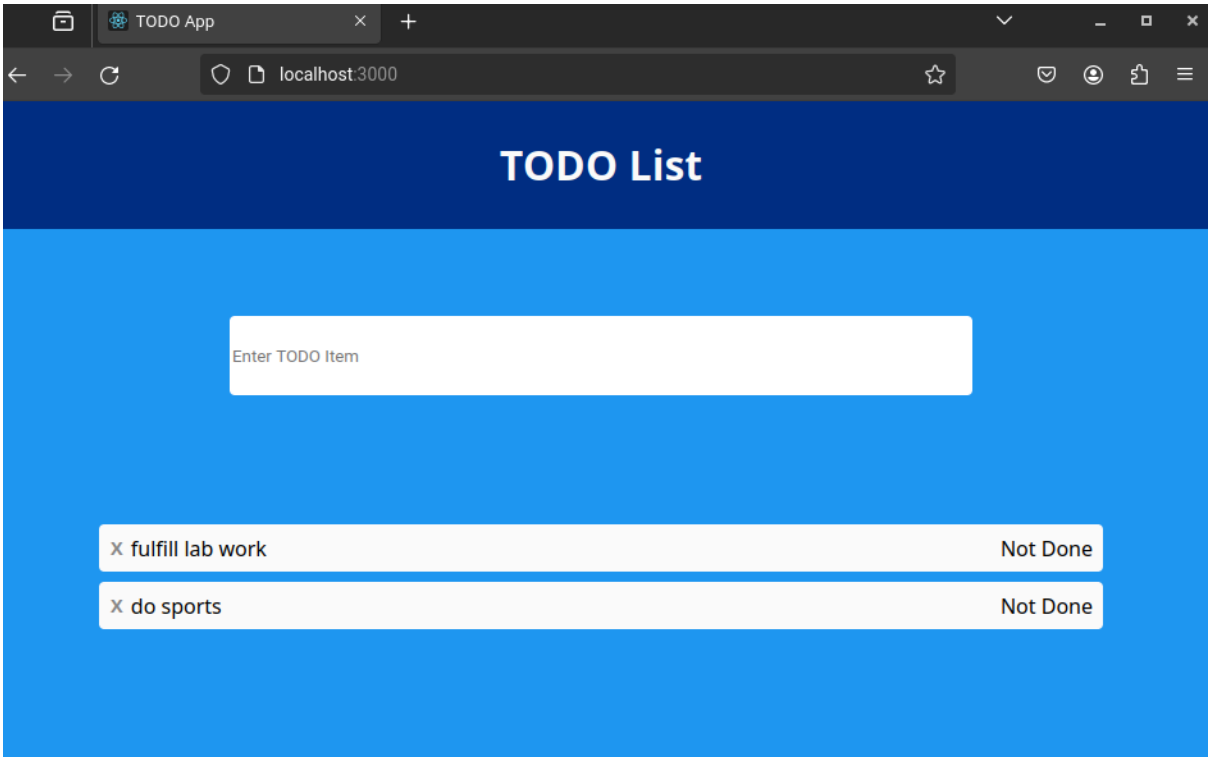
```
rs@arch ~/d/i/d/c/TODO-Fullstack-App-Go-Gin-Postgres-React [SIGINT]> docker-compose bui
d
[+] Building 205.0s (19/19) FINISHED                                docker:desktop-linux
 => [go-api internal] load .dockerignore                                           0.0s
 => => transferring context: 2B                                                    0.0s
 => [go-api internal] load build definition from Dockerfile                        0.0s
 => => transferring dockerfile: 164B                                               0.0s
 => [go-api internal] load metadata for docker.io/library/golang:alpine            0.6s
 => [react-app internal] load build definition from Dockerfile                     0.0s
 => => transferring dockerfile: 131B                                               0.0s
 => [react-app internal] load .dockerignore                                        0.0s
 => => transferring context: 2B                                                    0.0s
 => [react-app internal] load metadata for docker.io/library/node:latest           0.0s
 => [react-app 1/4] FROM docker.io/library/node                                    0.0s
 => [react-app internal] load build context                                        0.0s
```

запуск контейнеров:

```
rs@arch ~/d/i/d/c/TODO-Fullstack-App-Go-Gin-Postgres-React> docker-compose up
[+] Running 3/3
 ✔ Container todo-fullstack-app-go-gin-postgres-react-postgres-1   Created          0.0s
 ✔ Container todo-fullstack-app-go-gin-postgres-react-react-app-1  Recreated        0.2s
 ✔ Container todo-fullstack-app-go-gin-postgres-react-go-api-1     Recreated        0.3s
Attaching to go-api-1, postgres-1, react-app-1
```



добавим данные:



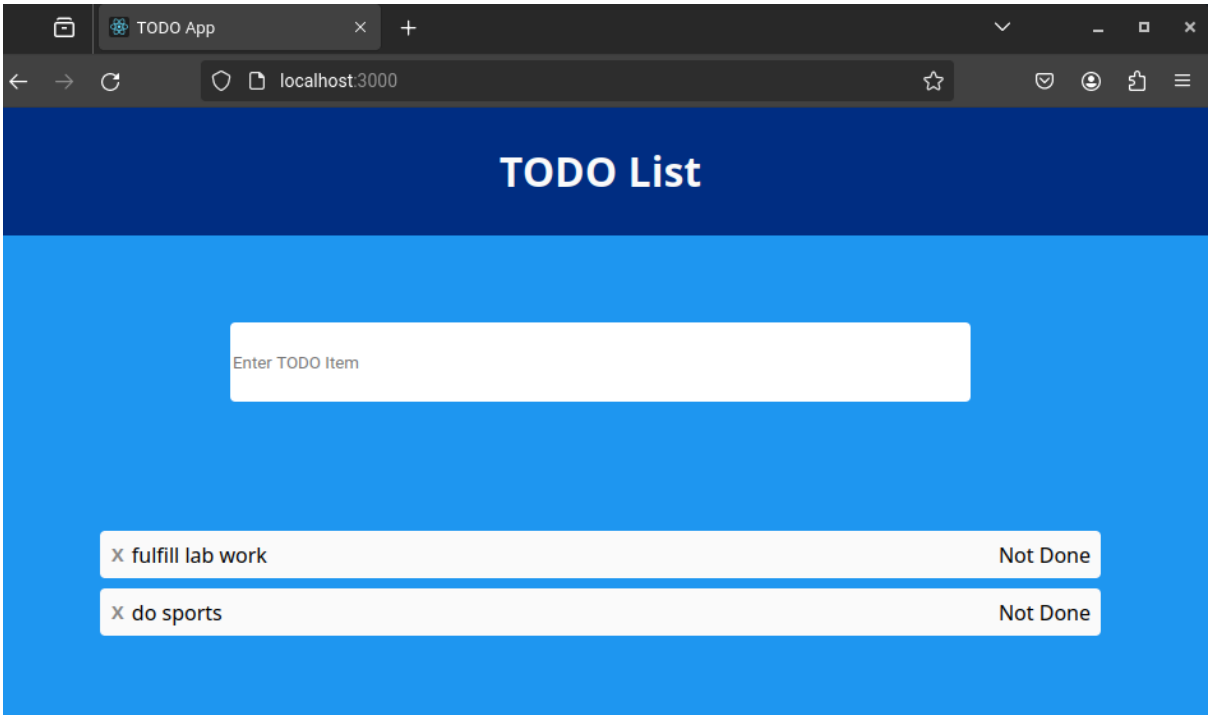
остановим контейнеры

```
rs@arch ~/d/i/d/c/TODO-Fullstack-App-Go-Gin-Postgres-React> docker-compose down
[+] Running 4/4
 ✔ Container todo-fullstack-app-go-gin-postgres-react-react-app-1  Removed        0.8s
 ✔ Container todo-fullstack-app-go-gin-postgres-react-go-api-1     Removed        0.2s
 ✔ Container todo-fullstack-app-go-gin-postgres-react-postgres-1   Removed        0.2s
 ✔ Network todo-fullstack-app-go-gin-postgres-react_default        Removed        0.1s
```

и запустим вновь

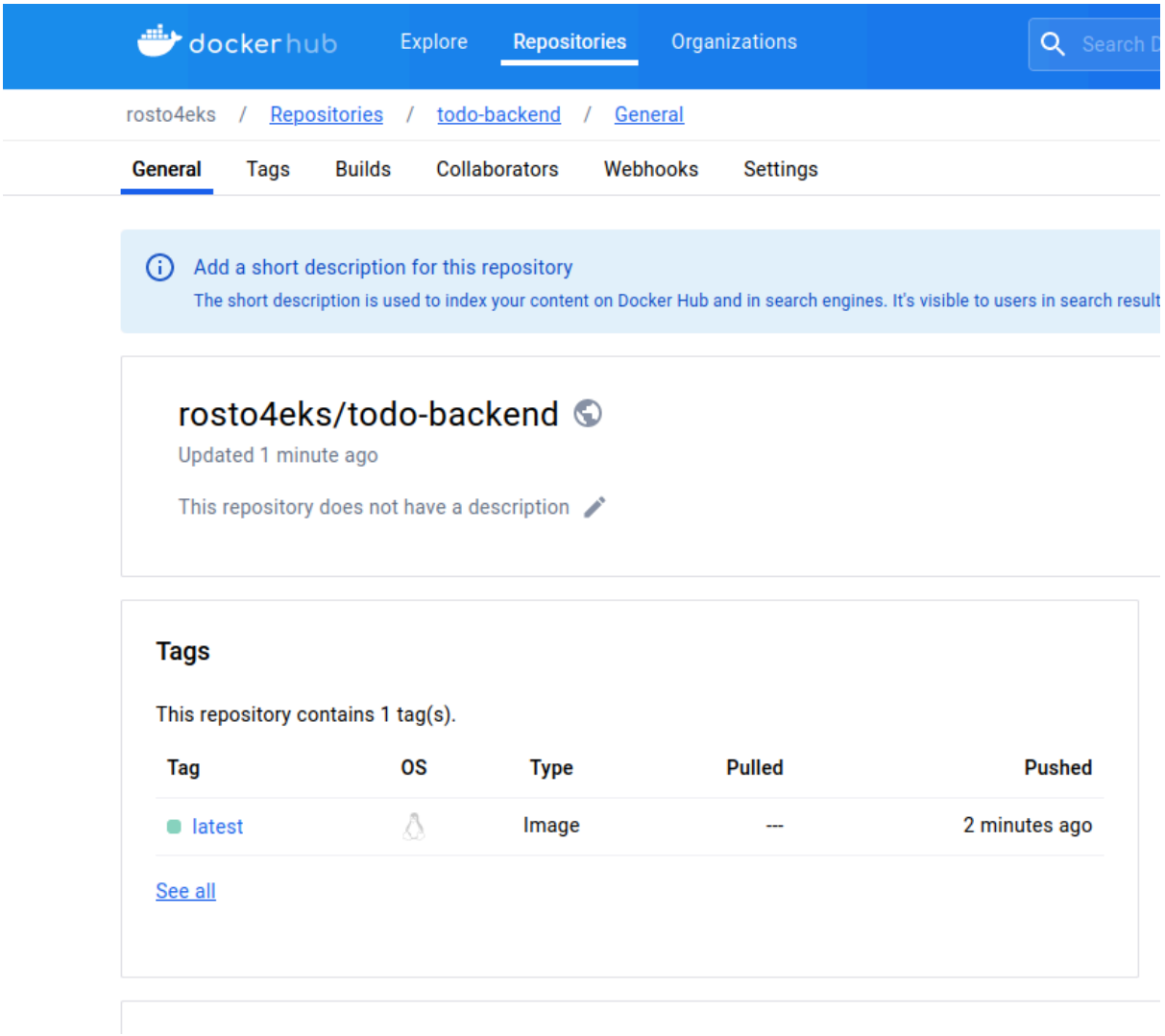
```
rs@arch ~/d/i/d/c/TODO-Fullstack-App-Go-Gin-Postgres-React [SIGINT]> docker-compose up
[+] Running 3/0
 ✔ Container todo-fullstack-app-go-gin-postgres-react-postgres-1   Created        0.0s
 ✔ Container todo-fullstack-app-go-gin-postgres-react-react-app-1  Created        0.0s
 ✔ Container todo-fullstack-app-go-gin-postgres-react-go-api-1     Created        0.0s
Attaching to go-api-1, postgres-1, react-app-1
```

увидим что данные сохранились



6

```
rs@arch ~/d/i/d/c/TODO-Fullstack-App-Go-Gin-Postgres-React> docker-compose push
[+] Pushing 1/20
 ✔ postgres Skipped                                                    0.0s
 ⠿ Pushing rosto4eks/todo-backend:latest: 2dd55aecc11d Waiting         2.8s
[+] Pushing 1/204eks/todo-backend:latest: 387a4332b1ff Waiting         2.8s
 ✔ postgres Skipped                                                    0.0s
 ⠿ Pushing rosto4eks/todo-backend:latest: 2dd55aecc11d Waiting         2.9s
[+] Pushing 1/204eks/todo-backend:latest: 387a4332b1ff Waiting         2.9s
 ✔ postgres Skipped                                                    0.0s
 ⠿ Pushing rosto4eks/todo-backend:latest: 2dd55aecc11d Waiting         3.0s
[+] Pushing 1/204eks/todo-backend:latest: 387a4332b1ff Waiting         3.0s
 ✔ postgres Skipped                                                    0.0s
 ⠿ Pushing rosto4eks/todo-backend:latest: 2dd55aecc11d Waiting         3.1s
```



7

получим сведения о всех сетях

```
rs@arch ~> docker network ls
NETWORK ID     NAME                                          DRIVER    SCOPE
804a2685d86d   bridge                                        bridge    local
bf738d4338d3   host                                          host      local
03dfd5d85bbf   none                                          null      local
83c14ba6f2b9   todo-fullstack-app-go-gin-postgres-react_default   bridge    local
```

получим подробную информацию об одной из сетей

```
rs@arch ~> docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "804a2685d86d6834bf5adaedff709f49d2880c392a924400c87f9396526d6b16",
        "Created": "2024-03-09T14:59:50.415594872+03:00",
        "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,
```

создадим свою сеть типа bridge

```
rs@arch ~> docker network create -d bridge abobanet
1a5a4d2cd2e6d9f0b4166706e887d9744917f017b2227b4109fec40cbbecdec2
rs@arch ~> docker network ls
NETWORK ID     NAME       DRIVER    SCOPE
1a5a4d2cd2e6   abobanet   bridge    local
804a2685d86d   bridge     bridge    local
bf738d4338d3   host       host      local
03dfd5d85bbf   none       null      local
```

```
rs@arch ~> docker network inspect 1a
[
    {
        "Name": "abobanet",
        "Id": "1a5a4d2cd2e6d9f0b4166706e887d9744917f017b2227b4109fec40cbbecdec2",
        "Created": "2024-03-09T16:19:00.04326889+03:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.20.0.0/16",
                    "Gateway": "172.20.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
```

подключим контейнер к сети

```
docker network connect abobanet 19d9
```

проверим сам контейнер

```
            "Networks": {
                "abobanet": {
                    "IPAMConfig": {},
                    "Links": null,
                    "Aliases": [
                        "19d904ae104b"
                    ],
                    "MacAddress": "02:42:ac:14:00:02",
                    "NetworkID": "1a5a4d2cd2e6d9f0b4166706e887d9744917f017b2227b4109fec4
cbbecdec2",
                    "EndpointID": "a4fdee5ac069e65b9f2935bf5078c388a2eff8ea0e5fb5e3fc103
759e73d0319",
                    "Gateway": "172.20.0.1",
                    "IPAddress": "172.20.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "DriverOpts": {},
                    "DNSNames": [
                        "silly_gates",
                        "19d904ae104b"
                    ]
                },
```


отключим контейнер от сети

```
docker network disconnect abobanet 19d9
```


```
            "Networks": {
                "bridge": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": null,
                    "MacAddress": "",
                    "NetworkID": "804a2685d86d6834bf5adaedff709f49d2880c392a924400c87
26d6b16",
                    "EndpointID": "",
                    "Gateway": "",
                    "IPAddress": "",
                    "IPPrefixLen": 0,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "DriverOpts": null,
                    "DNSNames": null
                }
            }
        }
    }

arch ~/d/i/d/python>
```

создадим новую сеть

```
docker network create -d bridge limpoponet
```

подключим к ней наши контейнеры

```
docker network connect limpoponet 7f
docker network connect limpoponet a3
docker network connect limpoponet fd
```

каждый контейнер подключен только к новой сети


```
"Networks": {
    "limpoponet": {
        "IPAMConfig": {},
        "Links": null,
        "Aliases": [
            "fdec704b537c"
        ],
        "MacAddress": "02:42:ac:15:00:04",
        "NetworkID": "30c110a8f17d62d26d5d375dd4c46325bcd641b7ee86256f4ada7e

        "EndpointID": "8704cfa860d46e707118ee971729bb84a71afd17b7caca3378374
,
        "Gateway": "172.21.0.1",
        "IPAddress": "172.21.0.4",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "DriverOpts": {},
        "DNSNames": [
            "nervous_ardinghelli",
            "fdec704b537c"
        ]
    }
```


проверим состояние сети

```
@arch ~> docker network inspect limpoponet

  {
      "Name": "limpoponet",
      "Id": "30c110a8f17d62d26d5d375dd4c46325bcd641b7ee86256f4ada7e93b47dbd9b",
      "Created": "2024-03-09T16:40:56.907515807+03:00",
      "Scope": "local",
      "Driver": "bridge",
      "EnableIPv6": false,
      "IPAM": {
          "Driver": "default",
          "Options": {},
          "Config": [
              {
                  "Subnet": "172.21.0.0/16",
                  "Gateway": "172.21.0.1"
```

можем увидеть подключенные контейнеры:

```
      },
      "ConfigOnly": false,
      "Containers": {
          "7fd8a8e0f8e3b5b00ccb74916e2ad90975ead7eff02be20bc102a88d831d93f7": {
              "Name": "exciting_robinson",
              "EndpointID": "789c6fac87c348c8972f9ad7b74c6ed577bef405d290f2bfc5215759e
ca651",
              "MacAddress": "02:42:ac:15:00:02",
              "IPv4Address": "172.21.0.2/16",
              "IPv6Address": ""
          },
          "a36f314ea56a9c9a2e80caeae4421daa4ff6e10a19d6e82c0bc4d6ea5b06dff1": {
              "Name": "magical_thompson",
              "EndpointID": "76aa4de33ec9d757deb047d81888af71ec62ebcb61e4823f1fa32570f
396fe",
              "MacAddress": "02:42:ac:15:00:03",
              "IPv4Address": "172.21.0.3/16",
              "IPv6Address": ""
          },
          "fdec704b537c4ac071f574d24f6e811f4e29d22bb836233b01b1547bf69a14bd": {
              "Name": "nervous_ardinghelli",
              "EndpointID": "8704cfa860d46e707118ee971729bb84a71afd17b7caca33783741dc6
71f1d",
              "MacAddress": "02:42:ac:15:00:04",
              "IPv4Address": "172.21.0.4/16",
              "IPv6Address": ""
          }
```

подключимся к контейнеру и попробуем пропинговать другой

```
rs@arch ~> docker exec -ti 1d /bin/bash
root@1d3471638ae7:/build# ping cb5c835d29e2
PING cb5c835d29e2 (172.21.0.3) 56(84) bytes of data.
64 bytes from gifted_euler.limpoponet (172.21.0.3): icmp_seq=1 ttl=64 time=0.209 ms
64 bytes from gifted_euler.limpoponet (172.21.0.3): icmp_seq=2 ttl=64 time=0.149 ms
64 bytes from gifted_euler.limpoponet (172.21.0.3): icmp_seq=3 ttl=64 time=0.157 ms
64 bytes from gifted_euler.limpoponet (172.21.0.3): icmp_seq=4 ttl=64 time=0.137 ms
64 bytes from gifted_euler.limpoponet (172.21.0.3): icmp_seq=5 ttl=64 time=0.149 ms
64 bytes from gifted_euler.limpoponet (172.21.0.3): icmp_seq=6 ttl=64 time=0.129 ms
```

создадим overlay сеть

```
rs@arch ~> docker swarm init
Swarm initialized: current node (m6fblxpqizhdhjo9t32yq3qru) is now a m

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

    docker swarm join --token SWMTKN-1-3qwb3b1t4t1e0338mlcaf6qmgt480cu
yuzvuojpl 192.168.145.210:2377

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

rs@arch ~> docker network create -d overlay overnet
2021o9gran7erhs0kc9ev8eaw
rs@arch ~> docker network inspect overnet
[
    {
        "Name": "overnet",
        "Id": "2021o9gran7erhs0kc9ev8eaw",
        "Created": "2024-03-09T15:08:48.664857117Z",
        "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
    }
]
```

создадим еще одну оверлей сеть

```
rs@arch ~> docker network create -d overlay overnet2
nknv46jtpbalbxrqifuo8ouwg
rs@arch ~> docker network inspect overnet2
[
    {
        "Name": "overnet2",
        "Id": "nknv46jtpbalbxrqifuo8ouwg",
        "Created": "2024-03-09T15:11:57.9274982817",
        "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
    }
]
```

теперь удалим созданную сеть

```
rs@arch ~> docker network ls
NETWORK ID     NAME                                              DRIVER    SCOPE
804a2685d86d   bridge                                            bridge    local
accb6c8f2549   docker_gwbridge                                   bridge    local
bf738d4338d3   host                                              host      local
ie2ij39upv82   ingress                                           overlay   swarm
30c110a8f17d   limpoponet                                        bridge    local
03dfd5d85bbf   none                                              null      local
2021o9gran7e   overnet                                           overlay   swarm
nknv46jtpbal   overnet2                                          overlay   swarm
d244ce459d84   todo-fullstack-app-go-gin-postgres-react_default   bridge    local
rs@arch ~> docker network rm overnet2
overnet2
rs@arch ~> docker network ls
NETWORK ID     NAME                                              DRIVER    SCOPE
804a2685d86d   bridge                                            bridge    local
accb6c8f2549   docker_gwbridge                                   bridge    local
bf738d4338d3   host                                              host      local
ie2ij39upv82   ingress                                           overlay   swarm
30c110a8f17d   limpoponet                                        bridge    local
03dfd5d85bbf   none                                              null      local
2021o9gran7e   overnet                                           overlay   swarm
d244ce459d84   todo-fullstack-app-go-gin-postgres-react_default   bridge    local
```

попробуем создать сеть host

```
rs@arch ~> docker network create -d host hostnet
Error response from daemon: only one instance of "host" network is allowed
```

однако докер не создает сеть так как только одна сеть типа host может существовать