

Лабораторная работа №2

Тема: Работа с Docker.

Цель: Познакомиться с возможностями и получить практические навыки работы с Docker.

Ссылки:

1. <https://docs.docker.com>
2. <https://hub.docker.com/>
3. <https://doka.guide/tools/dockerfile/>
4. <https://k21academy.com/docker-kubernetes/docker-storage/>
1. <https://k21academy.com/docker-kubernetes/docker-tutorial>
2. <https://k21academy.com/docker-kubernetes/docker-networking-different-types-of-networking-overview-for-beginners/>
3. <https://ru.bitdegree.org/rukovodstvo/docker-dlja-novichkov/>
4. <https://habr.com/ru/companies/slurm/articles/528206/>
5. <https://itisgood.ru/2019/09/10/docker-setevoe-vzaimodejstvie-101/>
6. <https://itisgood.ru/2019/10/29/objasnenie-koncepcii-setej-v-docker/>
7. <https://www.tune-it.ru/web/adpashnin/blog/-/blogs/docker-network>

Для защиты ЛР необходимо

- Оформить Отчет со скринами кода в командной строке и результатами его выполнения
- Продемонстрировать работу с командами docker: build, tag, run, start, stop, pause, unpause, restart, ps, logs, ps, images, network, volumes, inspect, а также знать назначение инструкций в Dockerfile и Docker Compose

Задание:

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

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/ИГИ/353502_MAKSIMENKOV_18/IGI/LR2$ docker
Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
run          Create and run a new container from an image
exec        Execute a command in a running container
ps          List containers
build       Build an image from a Dockerfile
pull        Download an image from a registry
push        Upload an image to a registry
images      List images
login        Authenticate to a registry
logout      Log out from a registry
search      Search Docker Hub for images
version     Show the Docker version information
info        Display system-wide information
```

- Выполните базовую настройку

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/ИГИ/353502_MAKSIMENKOV_18/IGI/LR2$ docker login
Authenticating with existing credentials... [Username: vlad20055]

Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/ИГИ/353502_MAKSIMENKOV_18/IGI/LR2$
```

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

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
confident_kare	0c4deb09130e	docker/getting-started:latest	80:80	0%	13 seconds ago	

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
```

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

```
Dockerfile
1 FROM python:3.9
2 WORKDIR app/
3 COPY script.py .
4 COPY circle.py .
5 COPY square.py .
6 CMD ["python", "script.py"]
7
```

с. Использовать его для создания контейнера. Протестировать использование контейнера

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/geometric-lib$ docker build . -t geometric-lib:1.0
[+] Building 132.7s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 145B
=> [internal] load metadata for docker.io/library/python:3.9
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.9@sha256:5ea663a1c6ba266fdcac5949d1d2ea364ce30a2da92a3df95bb3c01437633ad9
=> => resolve docker.io/library/python:3.9@sha256:5ea663a1c6ba266fdcac5949d1d2ea364ce30a2da92a3df95bb3c01437633ad9
=> => sha256:521cad6ddc5302ec0b1d426cdf6d64316fd18ddfc3b0924d24daee81b661501 250B / 250B
=> => sha256:95b7226c62e1a4719940920ae7fffd1ea4915befd3139d7020b84da24182ffdd9 19.85MB / 19.85MB
=> => sha256:93bee3686f319cf7bd4fcc659256d85121430628478ba3d026b5a96967b35cbe 6.16MB / 6.16MB
=> => sha256:447713e77b4fc3658cfba0c1e816b70ff6d9bf06563dc8cfc8b459406aed33b4 211.34MB / 211.34MB
=> => sha256:1d281e50d3e435595c266df06531a7e8c2ebbc185622c8ab2eed8d760e6576b 64.39MB / 64.39MB
=> => sha256:8031108f3cda87bb32f090262d0109c8a0db99168050967becefad502e9a681b 24.06MB / 24.06MB
=> => sha256:155ad54a8b2812a0ec559ff82c0c6f0f0dddb337a226b11879f09e15f67b69fc 48.40MB / 48.40MB
=> => extracting sha256:155ad54a8b2812a0ec559ff82c0c6f0f0dddb337a226b11879f09e15f67b69fc
=> => extracting sha256:8031108f3cda87bb32f090262d0109c8a0db99168050967becefad502e9a681b
=> => extracting sha256:1d281e50d3e435595c266df06531a7e8c2ebbc185622c8ab2eed8d760e6576b
=> => extracting sha256:447713e77b4fc3658cfba0c1e816b70ff6d9bf06563dc8cfc8b459406aed33b4
=> => extracting sha256:93bee3686f319cf7bd4fcc659256d85121430628478ba3d026b5a96967b35cbe
=> => extracting sha256:95b7226c62e1a4719940920ae7fffd1ea4915befd3139d7020b84da24182ffdd9
=> => extracting sha256:521cad6ddc5302ec0b1d426cdf6d64316fd18ddfc3b0924d24daee81b661501
=> [internal] load build context
=> => transferring context: 680B
=> [2/5] WORKDIR app/
=> [3/5] COPY script.py .
=> [4/5] COPY circle.py .
=> [5/5] COPY square.py .
=> => exporting image
=> => exporting layers
=> => exporting manifest sha256:6972cdc1d8b1a69c2928a12dfea48505c471c66c14be944e77d3886dc28fc41c
=> => exporting config sha256:f76e0dfbc1d02ff58ff1a87a92a1b0eeec2a1284c9c2ea97447b99e058e0307d7
=> => exporting attestation manifest sha256:7df698752a30b89dd408cdd797e96cdf7a378598c3078c3eb72c8b27e0c5163
=> => exporting manifest list sha256:342bf1b405ce8e2244474c3bc2f90d4e6c1f51039d3c20ed70eb313371659125
=> => naming to docker.io/library/geometric-lib:1.0
=> => unpacking to docker.io/library/geometric-lib:1.0

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/3z88uregv34mdthujn2hu4pwe
```

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/geometric-lib$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
geometric-lib        1.0                342bf1b405ce       About a minute ago 1.45GB
getting-started      latest             bb8584d7082a       39 hours ago       342MB
vlad20055/getting-started latest             bb8584d7082a       39 hours ago       342MB
docker/getting-started latest             d79336f4812b       2 years ago        73.9MB
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/geometric-lib$ docker run -it 342bf1b405ce
Введите радиус круга:12
Введите сторону квадрата:12
Периметр круга = 75.39822368615503
Площадь круга = 452.3893421169302
Периметр квадрата = 48.0
Площадь квадрата = 144.0
```

4. 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.(Примеры Images: https://hub.docker.com/_/phpmyadmin, https://hub.docker.com/_/mysql, https://hub.docker.com/_/postgres)

<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	first_site_django	-	-	-	N/A	2 minutes ago			
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	first_site_django_nginx_1	feecf965b891	first_site_django_nginx	80:80	N/A	2 minutes ago			
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	first_site_django_web_1	78bdede24c5b	first_site_django_web		N/A	2 minutes ago			
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	first_site_django_postgresdb_1	6b16060a74a1	postgres	5432:5432	N/A	2 minutes ago			

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2$ cd first_site_django/
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker-compose up -d
Starting first_site_django_postgresdb_1 ... done
Starting first_site_django_web_1 ... done
Starting first_site_django_nginx_1 ... done
```

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

```

"Containers": {
  "6b16060a74a147dcf3d1f93d6d6f5fb59a42eb8c51c9259ff64ac0d58baf981c": {
    "Name": "first_site_django_postgresdb_1",
    "EndpointID": "faedda43018fa18f053c6522dcff25519e2247bccad0f3c59005470eab8fc79d",
    "MacAddress": "02:42:ac:12:00:02",
    "IPv4Address": "172.18.0.2/16",
    "IPv6Address": ""
  },
  "78bdede24c5b484ac86d4255634f0b8d003f265f4e6baf7071f4c156b5c9e30e": {
    "Name": "first_site_django_web_1",
    "EndpointID": "98fdbd47a0fd3650c40ac776e19aa5862801919885a71a7f37a5c4f6e1f2af28",
    "MacAddress": "02:42:ac:12:00:03",
    "IPv4Address": "172.18.0.3/16",
    "IPv6Address": ""
  },
  "feecf965b8917eabe93fef8212b86f91ecb7b9fafbf2ad32c115b432f3427eef": {
    "Name": "first_site_django_nginx_1",
    "EndpointID": "b7cc0ec70a378c650357382baef20f13b75c5c070f7692f6c19652b7aca6c99c",
    "MacAddress": "02:42:ac:12:00:04",
    "IPv4Address": "172.18.0.4/16",
    "IPv6Address": ""
  }
},
"Options": {},
"Labels": {
  "com.docker.compose.network": "default",
  "com.docker.compose.project": "first_site_django",
  "com.docker.compose.version": "1.29.2"
}
}
]
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
feecf965b891   first_site_django_nginx            "/docker-entrypoint.…"   15 hours ago   Up 7 minutes   0.0.0.0:80->80/tcp                 first_site_django_nginx_1
78bdede24c5b   first_site_django_web              "/myapp/entrypoint.sh"   15 hours ago   Up 7 minutes   32777/tcp                          first_site_django_web_1
6b16060a74a1   postgres                           "docker-entrypoint.s…"   15 hours ago   Up 7 minutes   0.0.0.0:5432->5432/tcp            first_site_django_postgresdb_1
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$

```

6. Разместите результат в созданный репозиторий в DockerHub
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

```

vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network ls
NETWORK ID      NAME                                DRIVER  SCOPE
35027c46bc9e    bridge                             bridge  local
a83b513c9668    first_site_django_default          bridge  local
13080de1d0f1    host                               host    local
7a2f800bce21    none                               null     local
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker inspect a83b513c9668
[
  {
    "Name": "first_site_django_default",
    "Id": "a83b513c9668f40d0fc8bb53517479227c6c244bca63f1b9f17b425fdaa7d24",
    "Created": "2025-03-17T19:08:36.395288722Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": true,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {
      "com.docker.compose.network": "default",
      "com.docker.compose.project": "first_site_django",
      "com.docker.compose.version": "1.29.2"
    }
  }
]
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$

```

- 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


```

vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network create --driver bridge my_new_network
24a5c83fa2a67452b9628b744fe9713e7debb234dab290fb438f4dbc731ae4a7
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network ls

```

NETWORK ID	NAME	DRIVER	SCOPE
e294ee45e6d1	bridge	bridge	local
a83b513c9668	first_site_django_default	bridge	local
13080de1d0f1	host	host	local
24a5c83fa2a6	my_new_network	bridge	local
7a2f800bce21	none	null	local

```

vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network connect my_new_network e62f24f3ca2a

```

```

vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker container inspect e62f24f3ca2a

```

```

    "my_new_network": {
      "IPAMConfig": {},
      "Links": null,
      "Aliases": [],
      "MacAddress": "",
      "DriverOpts": {},
      "NetworkID": "",
      "EndpointID": "",
      "Gateway": "",
      "IPAddress": "",
      "IPPrefixLen": 0,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "DNSNames": [
        "brave_williamson",
        "e62f24f3ca2a"
      ]
    }

```

```

vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network disconnect my_new_network e62f24f3ca2a
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker start e62f24f3ca2a
e62f24f3ca2a
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker container inspect e62f24f3ca2a

```

```

    "Networks": {
      "bridge": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "MacAddress": "02:42:ac:11:00:02",
        "DriverOpts": null,
        "NetworkID": "75ab4710fd45e00503d27d67d284ccdb3e26adc6b5c37e75b83f7010f5fd52ec",
        "EndpointID": "80879da15b2c6a0c313005c3ff6ba5771c401ba6827a13b9893045aa59526231",
        "Gateway": "172.17.0.1",
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "DNSNames": null
      }
    }
}

```

Остался только bridge. который был и до этого (контейнер был подключён к двум сетям - bridge (default network) и my_new_network).

- 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

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network create --driver bridge my_another_new_network
c8478c700a5c0f207363d3073aa0831cb15075c28ab1412b92e601bd751396fa
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network ls
NETWORK ID        NAME                DRIVER  SCOPE
75ab4710fd45      bridge              bridge  local
a83b513c9668      first_site_django_default  bridge  local
13080de1d0f1      host                host    local
c8478c700a5c      my_another_new_network  bridge  local
24a5c83fa2a6      my_new_network       bridge  local
7a2f800bce21      none                null    local
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network inspect my_another_new_network
[
  {
    "Name": "my_another_new_network",
    "Id": "c8478c700a5c0f207363d3073aa0831cb15075c28ab1412b92e601bd751396fa",
    "Created": "2025-03-18T12:02:32.613553262Z",
    "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,
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]
```

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker run -d --network my_another_new_network d79336f4812b54eacd62309ebf43af92cd54db6d03f35a636ffa23280e741e35e2707711f98b
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker run -d --network my_another_new_network d79336f4812b6279d8f6d399bd81946b86de899973ecf1a93bfe28ccd2e0eb504fc1df14a294
```

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS          NAMES
6279d8f6d399  d79336f4812b  "/docker-entrypoint..." About a minute ago Up About a minute 80/tcp         peaceful_curie
54eacd62309e  d79336f4812b  "/docker-entrypoint..." 2 minutes ago   Up 2 minutes    80/tcp         practical_euler
746327be1fab  d79336f4812b  "/docker-entrypoint..." 8 minutes ago   Up 8 minutes    80/tcp         nice_bose
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker exec -i 6279d8f6d399 /bin/sh
ping 54eacd62309e
PING 54eacd62309e (172.20.0.3): 56 data bytes
64 bytes from 172.20.0.3: seq=0 ttl=64 time=0.379 ms
64 bytes from 172.20.0.3: seq=1 ttl=64 time=0.169 ms
64 bytes from 172.20.0.3: seq=2 ttl=64 time=0.123 ms
64 bytes from 172.20.0.3: seq=3 ttl=64 time=0.135 ms
64 bytes from 172.20.0.3: seq=4 ttl=64 time=0.171 ms
64 bytes from 172.20.0.3: seq=5 ttl=64 time=0.178 ms
64 bytes from 172.20.0.3: seq=6 ttl=64 time=0.177 ms
64 bytes from 172.20.0.3: seq=7 ttl=64 time=0.176 ms
64 bytes from 172.20.0.3: seq=8 ttl=64 time=0.168 ms
64 bytes from 172.20.0.3: seq=9 ttl=64 time=0.164 ms
64 bytes from 172.20.0.3: seq=10 ttl=64 time=0.170 ms
64 bytes from 172.20.0.3: seq=11 ttl=64 time=0.212 ms
64 bytes from 172.20.0.3: seq=12 ttl=64 time=0.202 ms
64 bytes from 172.20.0.3: seq=13 ttl=64 time=0.212 ms
64 bytes from 172.20.0.3: seq=14 ttl=64 time=0.145 ms
e64 bytes from 172.20.0.3: seq=15 ttl=64 time=0.211 ms
64 bytes from 172.20.0.3: seq=16 ttl=64 time=0.137 ms
64 bytes from 172.20.0.3: seq=17 ttl=64 time=0.140 ms
64 bytes from 172.20.0.3: seq=18 ttl=64 time=0.172 ms
64 bytes from 172.20.0.3: seq=19 ttl=64 time=0.217 ms
64 bytes from 172.20.0.3: seq=20 ttl=64 time=0.174 ms
^Ccontext canceled
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$
```

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

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network create --driver overlay my_overlay_network
Error response from daemon: This node is not a swarm manager. Use "docker swarm init" or "docker swarm join" to connect this node to swarm and try again.
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker swarm init
Swarm initialized: current node (8y2zog9sxackkf5rviucdj515) is now a manager.
```

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

```
docker swarm join --token SWMTKN-1-37jx2garsdsudusbntdcnvmst9y1pcygqifr47r8btmw7bmf-43sbl8looo7bsay1l1lu2qzxp 192.168.65.9:2377
```

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

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

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network create --driver overlay my_overlay_network
81wfw0kxb7sgw8f0cbuyza2cw
```

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network -ls
unknown shorthand flag: 'l' in -ls
```

Usage: docker network

Run 'docker network --help' for more information

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
70080de547a8	bridge	bridge	local
8b48efc5d946	docker_gwbridge	bridge	local
a83b513c9668	first_site_django_default	bridge	local
13080de1d0f1	host	host	local
xngocuw4yium	ingress	overlay	swarm
c8478c700a5c	my_another_new_network	bridge	local
24a5c83fa2a6	my_new_network	bridge	local
81wfw0kxb7sg	my_overlay_network	overlay	swarm
7a2f800bce21	none	null	local

7a2f800bce21 none null local

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network inspect my_overlay_network
[
  {
    "Name": "my_overlay_network",
    "Id": "81wfw0kxb7sgw8f0cbuyza2cw",
    "Created": "2025-03-18T12:36:52.133775969Z",
    "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

```
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network create --driver overlay my_second_overlay_network
3ksd06hha3r04sjp2pxxdjh8z
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
70080de547a8        bridge             bridge             local
8b48efc5d946        docker_gwbridge    bridge             local
a83b513c9668        first_site_django_default bridge             local
13080de1d0f1        host               host               local
xngocuw4yium        ingress            overlay            swarm
c8478c700a5c        my_another_new_network bridge             local
24a5c83fa2a6        my_new_network     bridge             local
81fw0kxb7sg         my_overlay_network overlay            swarm
3ksd06hha3r0        my_second_overlay_network overlay            swarm
7a2f800bce21        none               null               local
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network rm --help
Usage:  docker network rm NETWORK [NETWORK...]

Remove one or more networks

Aliases:
  docker network rm, docker network remove

Options:
  -f, --force  Do not error if the network does not exist
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network rm my_second_overlay_network
my_second_overlay_network
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
70080de547a8        bridge             bridge             local
8b48efc5d946        docker_gwbridge    bridge             local
a83b513c9668        first_site_django_default bridge             local
13080de1d0f1        host               host               local
xngocuw4yium        ingress            overlay            swarm
c8478c700a5c        my_another_new_network bridge             local
24a5c83fa2a6        my_new_network     bridge             local
81fw0kxb7sg         my_overlay_network overlay            swarm
7a2f800bce21        none               null               local
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$

• Попробовать создать сеть host, сохранить результат в отчет.

vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$ docker network create --driver host my_host_network
Error response from daemon: only one instance of "host" network is allowed
vlad@vlad-HP-Pavilion-Laptop-15-eh1xxx:~/Desktop/IGI/353502_MAKSIMENKOV_18/IGI/LR2/first_site_django$
```

Индивидуальное задание

Вар-т	Условие – стек технологий
1.	ReactJs-Django-SQLite
2.	ReactJS-Django-MySQL
3.	PHP, Django REST framework, MySQL and React
4.	Django 5, React, Bootstrap 5 with Python 3, PostgreSQL
5.	React, Node , PostgreSQL
6.	React.js - Redux - Node.js - Mongodb - Webpack
7.	Mongo DB, Express, Angular, Node.js
8.	Linux, Apache, MySQL, PHP
9.	MongoDB, Express.js, AngularJS, Node.js
10.	Ruby, Rails, Rack, Passenger
11.	.NET, MS SQL Server
12.	Java, MySQL

13.	ASP.NET MVC, IIS, Microsoft Azure, SQL Server
14.	React.js - Redux - Node.js - Mongodb
15.	ReactJs-Django-SQLite
16.	ReactJS-Django-MySQL
17.	PHP, Django REST framework, MySQL and React
18.	Django 5, React, Bootstrap 5 with Python 3, PostgreSQL
19.	React, Node , PostgreSQL
20.	React.js - Redux - Node.js - Mongodb - Webpack
21.	Mongo DB, Express, Angular, Node.js
22.	Linux, Apache, MySQL, PHP
23.	MongoDB, Express.js, AngularJS, Node.js
24.	Ruby, Rails, Rack, Passenger
25.	.NET, MS SQL Server
26.	Java, MySQL
27.	ASP.NET MVC, IIS, Microsoft Azure, SQL Server
28.	React.js - Redux - Node.js - Mongodb
29.	Linux, Apache, MySQL, PHP
30.	MongoDB, Express.js, AngularJS, Node.js

Контрольные вопросы

1. Что представляет собой контейнеризация?
2. Что такое Docker?
3. Что является реестром Docker?
4. Что такое Docker Engine?
5. Как происходит связь между клиентом Docker и демоном Docker?
6. Для чего необходим образ контейнера?
7. Что представляет собой контейнер Docker?
8. Каков метод создания Docker-контейнера?
9. Каковы этапы жизненного цикла контейнера Docker?
10. В каких средах можно использовать контейнерные приложения?
Что такое виртуальная машина?
11. Является ли Docker VM-технологией?
12. Какие различия между виртуальными машинами и контейнерами Docker?
13. Зачем использовать контейнеры?
14. Что выдаст команда \$ docker ps ?
15. Сколько идентификаторов по умолчанию имеет каждый контейнер?
16. Как задается имя контейнера?

17. Для чего нужна команда `container start`?
18. Как создать контейнер без запуска?
19. Что такое запуск контейнеров в интерактивном режиме?
20. Что представляет собой образ?
21. Что такое `Dockerfile`?
22. Для чего необходим `Docker Compose`?
23. Какая структура файла `Compose`?
24. Как запустить несколько контейнеров, используя один сервис?
25. Что такое предоставление порта связанным службам?
26. Для чего используются тома?
27. Для чего применяют переменные среды?
28. Как определяют правила связи между контейнерами?
Написать файл `Docker` для создания и копирования каталога и построить его с использованием модулей `Python`?
29. Какие сети доступны по умолчанию в `Docker`?
30. Приведите необходимые шаги для развертывания докеризированного приложения, сохраненного в репозитории `Git`