

Урок 5. Docker Compose и Docker Swarm

Задача 1:

Необходимо создать сервис, состоящий из 2 различных контейнеров: 1 - веб, 2 - базы данных.

Далее - необходимо создать 3

сервиса в каждом окружении (dev, prod, lab). По итогу на каждой ноде должно быть по 2 работающих контейнера. Выводы

зафиксировать.

Результатом работы будут логи выполнения, история команд и скриншоты, сделанные вами.

№1

1-вариант, до семинара:

- 1) `mkdir my_project && cd my_project`
- 2) Создадим файл `docker-compose.yml`

```
version: '3'
```

```
services:  
  web:  
    image: nginx:latest  
    ports:  
      - 8755:80  
    volumes:  
      - ./web:/usr/share/nginx/html  
    environment:  
      ENVIRONMENT: dev # Может быть dev, prod, lab, в зависимости от окружения
```

```
database  
  image: postgres:latest  
  volumes:  
    - ./db:/var/lib/postgresql/data  
  environment:  
    POSTGRES_DB: db  
    POSTGRES_USER: user  
    POSTGRES_PASSWORD: 123  
  environment:  
    ENVIRONMENT: dev # Может быть dev, prod, lab, в зависимости от окружения
```

```
1:fred2@fred2VirtualBox: ~/my_project ~
  GNU nano 6.2                                     docker-compose.yml
version: '3'

services:
  web:
    image: nginx:latest
    ports:
      - "8755:80"
    volumes:
      - ./web:/usr/share/nginx/html
    environment:
      ENVIRONMENT: dev # Может быть dev, prod, lab, в зависимости от окружения

  database:
    image: postgres:latest
    volumes:
      - ./db:/var/lib/postgresql/data
    environment:
      POSTGRES_DB: db
      POSTGRES_USER: user
      POSTGRES_PASSWORD: 123
      ENVIRONMENT: dev # Может быть dev, prod, lab, в зависимости от окружения
```

3) dev.env, prod.env, lab.env

```
sudo nano dev.env  
| >ENVIRONMENT=dev
```

```
sudo nano prod.env  
_> ENVIRONMENT=prod
```

```
sudo nano lab.env
```

```
1: fred2@fred2VirtualBox: ~/my_project ~\n  GNU nano 6.2\nENVIRONMENT=dev
```

```
1: fred2@fred2VirtualBox: ~/my_project ~  
GNU nano 6.2  
ENVIRONMENT=prod
```

```
1: fred2@fred2VirtualBox: ~/my_project ~  
  GNU nano 6.2  
ENVIRONMENT=lab
```

```

Activities   Tilix
Oct 12 19:51
Tilix: fred2@fred2VirtualBox: ~/my_project
1:fred2@fred2VirtualBox:~/my_project ~
fred2@fred2VirtualBox:~$ mkdir my_project
fred2@fred2VirtualBox:~$ cd my_project/
fred2@fred2VirtualBox:~/my_projects$ ll
total 8
drwxrwxr-x  2 fred2 fred2 4096 Oct 12 19:30 .
drwxr-x--- 21 fred2 fred2 4096 Oct 12 19:30 ..
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
[sudo] password for fred2:
fred2@fred2VirtualBox:~/my_projects$ ll
total 12
drwxrwxr-x  2 fred2 fred2 4096 Oct 12 19:33 .
drwxr-x--- 21 fred2 fred2 4096 Oct 12 19:30 ..
rv.r.-. 1 root root 581 Oct 12 19:33 docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo nano dev.env
fred2@fred2VirtualBox:~/my_projects$ sudo nano prod.env
fred2@fred2VirtualBox:~/my_projects$ sudo nano lab.env
fred2@fred2VirtualBox:~/my_projects$ docker-compose --env-file dev.env up -d
Command 'docker-compose' not found, but can be installed with:
sudo snap install docker           # version 20.10.24, or
sudo snap install docker           # version 20.10.24
sudo apt install docker-compose    # version 1.29.2-1
See 'snap info <snapname>' for additional versions.
fred2@fred2VirtualBox:~/my_projects$ sudo docker-compose --env-file dev.env up -d
sudo: docker-compose: command not found
fred2@fred2VirtualBox:~/my_projects$ sudo docker compose --env-file dev.env up -d
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 2: did not find expected <document start>
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
parsing /home/fred2/my_project/docker-compose.yml: yaml: unmarshal errors:
  line 21: mapping key "environment" already defined at line 15
fred2@fred2VirtualBox:~/my_projects$ nano docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo docker compose --env-file dev.env up -d
parsing /home/fred2/my_project/docker-compose.yml: yaml: unmarshal errors:
  line 21: mapping key "environment" already defined at line 15
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo docker compose --env-file dev.env up -d
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo docker compose --env-file dev.env up -d
parsing /home/fred2/my_project/docker-compose.yml: yaml: unmarshal errors:
  line 21: mapping key "environment" already defined at line 15
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo docker compose --env-file dev.env up -d
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo docker compose --env-file dev.env up -d

```

4)

sudo docker compose --env-file dev.env up -d

sudo docker compose --env-file prod.env up -d

sudo docker compose --env-file lab.env up -d

```

Activities   Tilix
Oct 12 19:50
Tilix: fred2@fred2VirtualBox: ~/my_project
1:fred2@fred2VirtualBox:~/my_project ~
fred2@fred2VirtualBox:~/my_projects$ sudo docker compose --env-file dev.env up -d
parsing /home/fred2/my_project/docker-compose.yml: yaml: unmarshal errors:
  line 21: mapping key "environment" already defined at line 15
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo docker compose --env-file dev.env up -d
parsing /home/fred2/my_project/docker-compose.yml: yaml: unmarshal errors:
  line 21: mapping key "environment" already defined at line 15
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo docker compose --env-file dev.env up -d
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo docker compose --env-file dev.env up -d
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
parsing /home/fred2/my_project/docker-compose.yml: yaml: line 17: did not find expected key
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo docker compose --env-file dev.env up -d
[+] Running 21/21
  ✓ web 7 layers [██████████]    0B/0B    Pulled          69.5s
    a578f1003218 Pull complete      24.1s
    4dff1008536 Pull complete      27.7s
    2130e6992c9a Pull complete      0.8s
    e4432b28cc0 Pull complete      2.3s
    f6135ab611400 Pull complete     3.1s
    6c535ab49fa4a Pull complete     3.8s
    d57731fb9008 Pull complete     4.5s
  ✓ database 2 layers [██████████]  0B/0B    Pulled          96.2s
    2abc5690e291 Pull complete      5.1s
    8fe57734687 Pull complete      9.9s
    a2ddb09cd9a Pull complete      11.5s
    5a2499e87ab8 Pull complete      15.8s
    a45f5c4ad1b Pull complete      17.0s
    178017fd978e Pull complete      17.9s
    428dff1cb77d Pull complete      18.7s
    4667364adfc4 Pull complete      53.5s
    4eeal15281a9 Pull complete      24.6s
    369467411787 Pull complete      25.4s
    ✓ 51184495a2bc Pull complete     27.2s
    ✓ d3e246f01410 Pull complete     28.1s
[+] Running 3/3
  ✓ Network my_project_default     Created          0.4s
  ✓ Container my_project-database-1 Started        2.3s
  ✓ Container my_project-web-1     Started        2.3s
fred2@fred2VirtualBox:~/my_projects$ 

```

```

Oct 12 20:07
Tiliix: fred2@fred2VirtualBox: ~/my_project

1:fred2@fred2VirtualBox:~/my_project ~
  ✓ 6c538b49fa4a Pull complete
  ✓ d57731fb9008 Pull complete
✓ database_12_layers [██████████] 0B/0B Pulled
  ✓ 2ebc5690e391 Pull complete
  ✓ 8fe577f34687 Pull complete
  ✓ a7ddb090cd98 Pull complete
  ✓ 5a2499e87ab8 Pull complete
  ✓ a4515c44df1b Pull complete
  ✓ 178017fd978e Pull complete
  ✓ 428dff1cb77d Pull complete
  ✓ 4667364adfc4 Pull complete
  ✓ 4eeal1f5281a9 Pull complete
  ✓ 369467411787 Pull complete
  ✓ 51184495a2bc Pull complete
  ✓ d2e246f01410 Pull complete
[+] Running 7/7
✓ Network my project default      Created
✓ Container my project-database-1 Started
✓ Container my project-web-1      Started
fred2@fred2VirtualBox:~/my_projects$ sudo nano docker-compose.yml
fred2@fred2VirtualBox:~/my_projects$ sudo nano dev.env
fred2@fred2VirtualBox:~/my_projects$ sudo nano prod.env
fred2@fred2VirtualBox:~/my_projects$ sudo nano lab.env
fred2@fred2VirtualBox:~/my_projects$ sudo docker ps
CONTAINER ID        NAMES               COMMAND           CREATED          STATUS             PORTS
1fb692da6516        postgres:latest    "docker-entrypoint.s..." 19 minutes ago   Up 18 minutes    5432/tcp
a60424cdda42        my-project-database-1 "docker-entrypoint.s..." 19 minutes ago   Up 18 minutes    0.0.0.0:8755->80/tcp, ...
6df96e9440d2        nginx:latest       "/docker-entrypoint..." 2 hours ago     Up 2 hours       0.0.0.0:8888->80/tcp, ...
640b1f907712        phpmyadmin/phpmyadmin  "phpmyadmin"        2 hours ago     Up 2 hours       0.0.0.0:8777->80/tcp, ...
640b1f907712        my-mariadb:ubuntu5  "mysqld --user=mysql" 2 hours ago     Up 2 hours       0.0.0.0:3377->3306/tcp,
a8036c4d4dd         phpmyadmin/phpmyadmin  "phpmyadmin"        3 hours ago     Up 3 hours       0.0.0.0:8555->80/tcp, ...
a80d1449d517        my-mariadb:ubuntu4  "mysqld --user=mysql" 3 hours ago     Up 3 hours       0.0.0.0:3366->3306/tcp,
fred2@fred2VirtualBox:~/my_projects$
```



Задание 1:

- 1) создать docker compose файл, состоящий из 2 различных контейнеров: 1 - веб, 2 - БД
 - 2) запустить docker compose файл
 - 3) по итогу на БД контейнере должно быть 2 реплики, на админере должна быть 1 реплика.
- Всего должно получиться 3 контейнера
- 4) выводы зафиксировать

Задание 2*:

- 1) создать кластер и мастер и слейв ноды
- 2) задеплоить на ноду несколько экземпляров какого-нибудь контейнера, например nginx
- 3) обязательно проверить и зафиксировать результаты, чтобы можно было выслать преподавателю для проверки

повторяем семинар:

№2 Задание №1

1) создаем папку и переходим в эту папку

mkdir 09

cd 09

2) создаем docker-compose.yml

```
1:fred2@fred2VirtualBox:~/09 ~
GNU nano 6.2                                     docker-compose.yml
version: '3.9'

services:
  db:
    build:
      dockerfile: ./Dockerfile
    environment:
      MYSQL_ROOT_PASSWORD: 12345
    deploy:
      mode: replicated
      replicas: 2

  adminer:
    image: adminer:4.8.1
    restart: always
    ports:
      - 6080:8080
    volumes:
      - ./myfolder:/myfolder
    deploy:
      mode: replicated
      replicas: 1
```

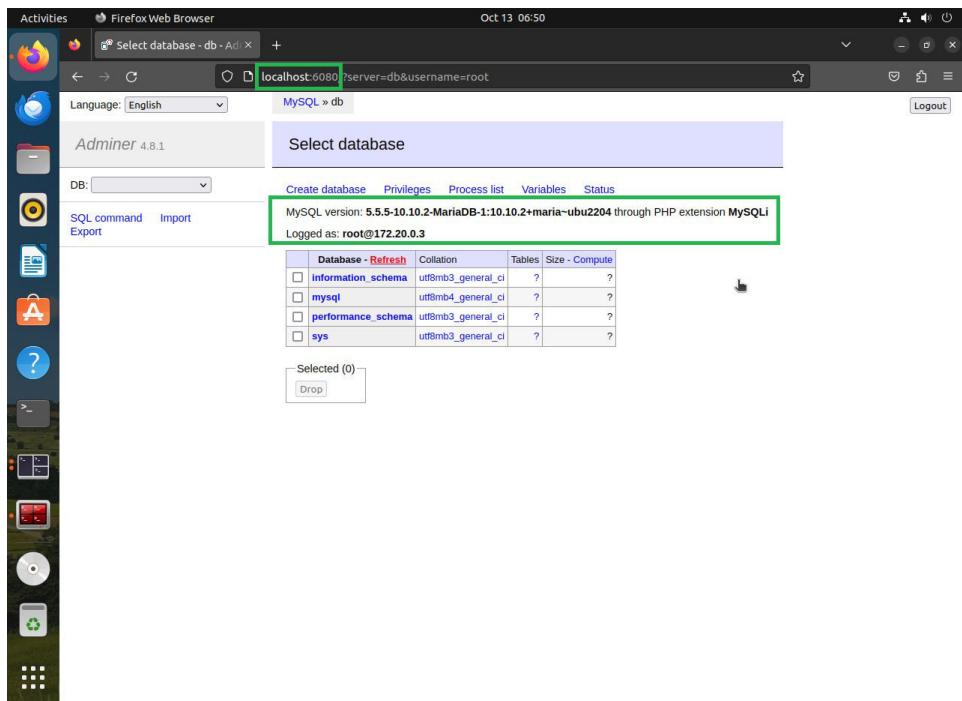
3) создаем Dockerfile , который будем build, в той же папки, что и docker-compose.yml
sudo nano Dockerfile

```
1:fred2@fred2VirtualBox:~/09 ~  
GNU nano 6.2 Dockerfile  
FROM mariadb:10.10.2  
    |  
RUN mkdir /test
```

```
fred2@fred2VirtualBox:~/09$ ll
_|> total 16
_|> drwxrwxr-x 2 fred2 fred2 4096 Oct 13 05:22 .
_|> drwxr-x--- 22 fred2 fred2 4096 Oct 13 05:17 ..
_|> -rw-r--r-- 1 root root 358 Oct 13 05:21 docker-compose.yml
_|> -rw-r--r-- 1 root root 38 Oct 13 05:22 Dockerfile
```

4) запускаем контейнер , без режима демона -d, текущая консоль будет загружена этим процессом и по этой причине будем вводить команды в другой консоли
`sudo docker compose up`

5) до того как остановить проверяем
<http://localhost:6080>



6) останавливаем и смотрим процессы docker ps -a

```
sudo docker compose stop
sudo docker ps -a
```

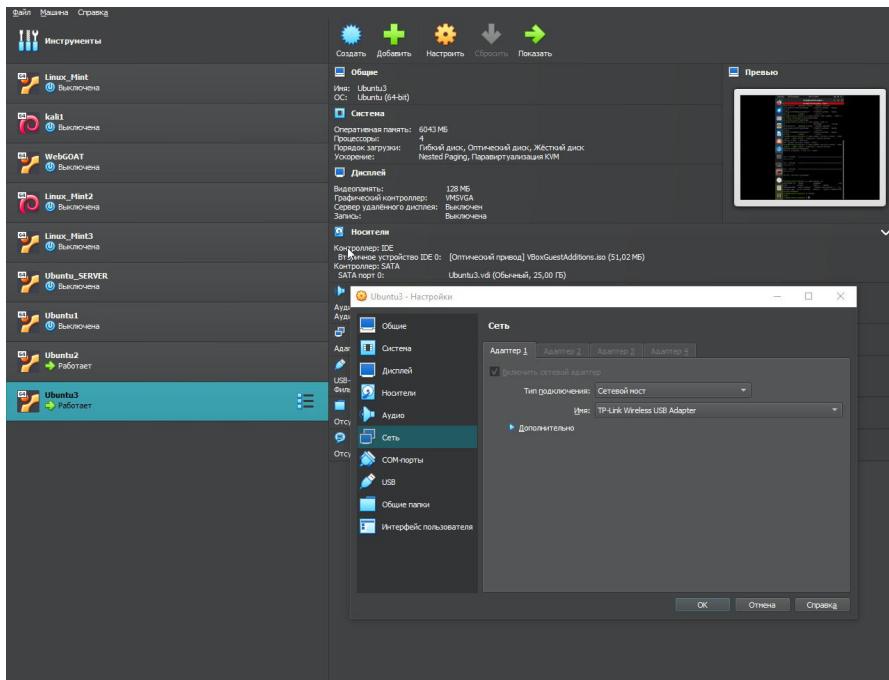
7) удаляем

```
sudo docker compose down
```

```
fred2@fred2VirtualBox:~/09$ sudo docker compose down
[+] Running 4/4
✓ Container 09-db-2      Removed          0.0s
✓ Container 09-adminer-1  Removed          0.0s
fred2@fred2VirtualBox:~/09$ sudo docker compose start
```

№2 Задание №2

1) нужны две виртуальные машины с общим сетевым мостом



2) на первой машине, у меня Ubuntu3, создаем swarm (Leader)

```
sudo docker swarm init
```

```
|_>Swarm initialized: current node (z2z9qijinf9w01u3cmx6oki8) is now a manager.
```

```
|_>To add a worker to this swarm, run the following command:
```

```
|_>docker swarm join --token SWMTKN-1-
```

```
0owrftdv6hkzmmf192c4670xipj2onmgmuo1kmh7rvr877ghk-ch7ex9cg30wh2qrl2y46bafb2  
192.168.1.67:2377
```

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

проверяем:

```
sudo docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINEVERSION
z2z9qijinf9w01u3cmx6oki8 *	fred2VirtualBox	Ready	Active	Leader	24.0.6

Activities Terminator Oct 13 06:59

```

fred2@fred2VirtualBox:~$ sudo docker swarm init
[sudo] password for fred2:
Swarm initialized: current node (zzz9qijinf9w01u3cmx6oki8) is now a manager.

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

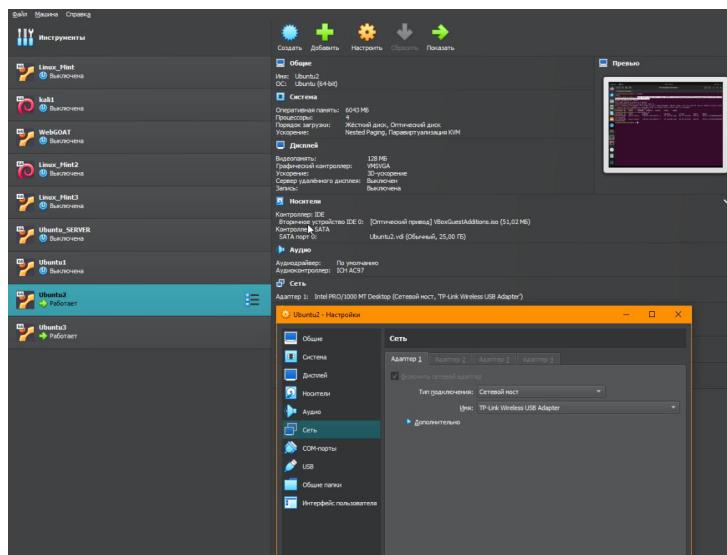
    docker swarm join --token SWMTKN-1-0owrftdv6hkzmmf192c4670xipj2onmgmuo1kmh7rvr877ghk-ch7ex9cg30wh2qrly46baf82 192.168.1.67:2377

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

fred2@fred2VirtualBox:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
fred2@fred2VirtualBox:~$ sudo docker node ls
ID NAME HOSTNAME DESIRED STATE CURRENT STATE ERROR PORTS
fred2@fred2VirtualBox:~$ sudo docker node ls
ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS ENGINE VERSION
zzz9qijinf9w01u3cmx6oki8 * fred2VirtualBox Ready Active Leader 24.0.6
fred2@fred2VirtualBox:~$ sudo docker node ls
ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS ENGINE VERSION
jnvstve6xlczidlnqkn9dogb fred2VirtualBox Ready Active 24.0.6
zzz9qijinf9w01u3cmx6oki8 * fred2VirtualBox Ready Active Leader 24.0.6
fred2@fred2VirtualBox:~$ sudo docker node update --label-add tokyo jnvstve6xlczidlnqkn9dogb
jnvstve6xlczidlnqkn9dogb
fred2@fred2VirtualBox:~$ sudo docker node ls
ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS ENGINE VERSION
jnvstve6xlczidlnqkn9dogb fred2VirtualBox Ready Active 24.0.6
zzz9qijinf9w01u3cmx6oki8 * fred2VirtualBox Ready Active Leader 24.0.6
fred2@fred2VirtualBox:~$ sudo docker service create --name nginx --label tokyo --replicas 3 nginx:alpine;
unknown flag: --replicas
See 'docker service create --help'.
fred2@fred2VirtualBox:~$ sudo docker service create --name nginx --label tokyo --replicas 3 nginx:alpine;
wpjhnu7nnmk2uxfr0jg5en5
overall progress: 3 out of 3 tasks
1/3: running [=====]
2/3: running [=====]
3/3: running [=====]
verify: Service converged
fred2@fred2VirtualBox:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
29035e9228b9 nginx:alpine "/docker-entrypoint..." 23 seconds ago Up 20 seconds 80/tcp nginx.2.wpw0lsch4cxucyhkxlf0d8ho4
fred2@fred2VirtualBox:~$ whoami
fred2
fred2@fred2VirtualBox:~$
```

2) копируем токен с 1 машине на 2 машину, у меня с Ubuntu3 → Ubuntu2

`sudo docker swarm join --token SWMTKN-1-0owrftdv6hkzmmf192c4670xipj2onmgmuo1kmh7rvr877ghk-ch7ex9cg30wh2qrly46baf82 192.168.1.67:2377`



```
Activities   Tilix          Oct 13 07:01
1 / +  ↻  Tilix: fred1@fred1VirtualBox: ~

1:fred1@fred1VirtualBox:~ $ whoami
fred1

fred1@fred1VirtualBox:~ $ sudo docker swarm join --token SWMTKN-1-0owrftdv6hkzmmf192c4670xipj2onmggmu01kmh7rvr877ghk-ch
e x9cg30wh2qr12y46bafb2 192.168.1.67:2377
[sudo] password for fred1:
This node joined a swarm as a worker.

fred1@fred1VirtualBox:~ $ sudo docker node ls
Error response from daemon: This node is not a swarm manager. Worker nodes can't be used to view or modify cluster state
. Please run this command on a manager node or promote the current node to a manager.

fred1@fred1VirtualBox:~ $ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
fred1@fred1VirtualBox:~ $ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
dfbd14ac0a66 nginx:alpine "/docker-entrypoint..." 31 seconds ago Up 27 seconds 80/tcp nginx.1.4yfm09magy459h
yvcts07i9x8
4ef7d6420248 nginx:alpine "/docker-entrypoint..." 31 seconds ago Up 28 seconds 80/tcp nginx.3.5fo6f5oga5phb9
rwba39zf8mb
fred1@fred1VirtualBox:~ $
```

The screenshot shows two side-by-side terminal windows in Oracle VM VirtualBox. Both windows have a title bar "Ubuntu [Работает] - Oracle VM VirtualBox".

User fred2@fred2VirtualBox:

- Output of `sudo docker node update --label-a dd tokyo`:
jnvstve6xlcz1dlfnqn9dgb
z2z9qljnlfd9w0iu3cmx60kis * fred2VirtualBox Ready
- Output of `sudo docker node ls`:
ID HOSTNAME STATUS
jnvstve6xlcz1dlfnqn9dgb fred2VirtualBox Ready
z2z9qljnlfd9w0iu3cmx60kis * fred2VirtualBox Ready
- Output of `sudo docker service create --name nginx --label tokyo --replicas 3 nginx:alpine;`
See 'docker service create --help'.
- Output of `sudo docker service create --name nginx --label tokyo --replicas 3 nginx:alpine;`
wptjhnn7unmk2uxfr0jgsm92
overall progress: 3 out of 3 tasks
- Output of `sudo docker ps`:
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
1/3: running [=====>]
2/3: running [=====>]
3/3: running [=====>]
verify: Service converged
- Output of `sudo docker ps`:
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
29035e9228b9 nginx:alpine "/docker-entrypoint..." 23 seconds ago Up 20 seconds 80/tcp nginx.2.wpw0lsch4c xucyhkxlf0dhs4
- Output of `whoami`:
Fred2

User fred1@fred1VirtualBox:

- Output of `sudo docker swarm join --token SWMTK-1-0owfrtv6hkzmmf192c4670xipj20mngmu01kmh7rvr877ghk-ch7ex9cg30wh2qr12y46bafb2 192.168.1.67:2377`
[sudo] password for fred1:
This node joined swarm as a worker.
- Output of `sudo docker node ls`:
Error response from daemon: This node is not a swarm manager. Worker nodes can't be used to view or modify cluster state. Please run this command on a manager node or promote the current node to a manager.
- Output of `sudo docker ps`:
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
1/3: running [=====>]
2/3: running [=====>]
3/3: running [=====>]
verify: Service converged
- Output of `sudo docker ps`:
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
dfbd14ac0a66 nginx:alpine "/docker-entrypoint..." 31 seconds ago Up 27 seconds 80/tcp nginx.1.4yfm09magy45 9hyvcts0t19x8
4ef7d6420248 nginx:alpine "/docker-entrypoint..." 31 seconds ago Up 28 seconds 80/tcp nginx.3.5f06f5oga5ph b9rwba9zf8mb
- Output of `whoami`:
Fred1

3) в основной машине, у меня ubuntu3, придумаем имя для машины 2 (Node 2) => Ubuntu2

```
sudo docker node update --label-add tokyo jnvstve6xlcz1dlfnqkn9dog
```

4) Запустим 3 копии(реплики = replicas) энджинкс на ОС алпайн :

```
sudo docker service create --name nginx --label tokyo --replicas 3 nginx:alpine;
```

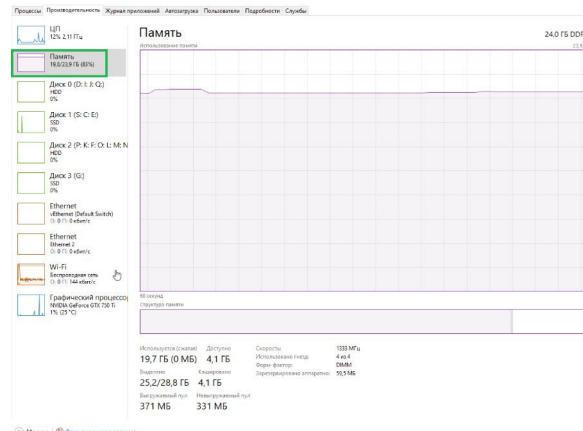
5) один nginx был создан на основной машине(мастер нода) , а два других на второй машине(рабочая нода или служебная нода):

```
See 'docker service create --help'.
fred2@fred2VirtualBox:~$ sudo docker service create --name nginx --label tokyo --replicas 3 nginx:alpine;
[...]
overall progress: 3 out of 3 tasks
1/3: running [=====]
2/3: running [=====]
3/3: running [=====]
[...]
[service] Services are up-to-date.
fred2@fred2VirtualBox:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
29035e9228b9 nginx:alpine "/docker-entrypoint..." 23 seconds ago Up 20 seconds 80/tcp nginx.2.wpw0lsch4cxucyhkxlfd8ho4
fred2@fred2VirtualBox:~$ whoami
root
fred2@fred2VirtualBox:~$ sudo docker ps
[sudo] password for fred2:
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
29035e9228b9 nginx:alpine "/docker-entrypoint..." 2 hours ago Up 2 hours 80/tcp nginx.2.wpw0lsch4cxucyhkxlfd8ho4
fred2@fred2VirtualBox:~$
```

```
1:fred1@fred1VirtualBox:~ ~
fred1@fred1VirtualBox:~$ whoami
fred1
fred1@fred1VirtualBox:~$ sudo docker swarm join --token SWMTKN-1-0owrftdv6hkzmmf192c4670xipj2onmggmuolkmh7rvr877ghk-ch7ex9c30wh2qr12y46babfb2 192.168.1.67:2377
[sudo] password for fred1:
This node joined a swarm as a worker.
fred1@fred1VirtualBox:~$ sudo docker node ls
Error response from daemon: This node is not a swarm manager. Worker nodes can't be used to view or modify cluster state
. Please run this command on a manager node or promote the current node to a manager.
fred1@fred1VirtualBox:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
dfbd14ac0a66 nginx:alpine "/docker-entrypoint..." 31 seconds ago Up 27 seconds 80/tcp nginx.1.4yfm09magy459h
yvcts0i9x8
4effd6420248 nginx:alpine "/docker-entrypoint..." 31 seconds ago Up 28 seconds 80/tcp nginx.3.5fo6f5oga5phb9
rwv3a9zf8mb
fred1@fred1VirtualBox:~$
```

в принципе все .

Остается попробовать запустить 3 виртуальные машины, самое гл правильно распределить ОЗУ, а то уже в притык



Задание 1:

- 1) создать docker compose файл, состоящий из 2 различных контейнеров: 1 - веб, 2 - БД
- 2) запустить docker compose файл
- 3) по итогу на БД контейнере должно быть 2 реплики, на админере должна быть 1 реплика.
Всего должно получиться 3 контейнера
- 4) выводы зафиксировать

- 1) mkdir 02 && cd 02
- 2) sudo nano docker-compose.yml

```
version: '3'
services:
  web:
    image: nginx:latest
    ports:
      - "85:80"
  db:
    image: postgres:latest
    environment:
      POSTGRES_DB: db
      POSTGRES_USER: user
      POSTGRES_PASSWORD: 123
    deploy:
      replicas: 2
  adminer:
    image: adminer
    ports:
      - "8181:8080"
```

```
GNU nano 6.2
version: '3'
services:
  web:
    image: nginx:latest
    ports:
      - "85:80"
  db:
    image: postgres:latest
    environment:
      POSTGRES_DB: db
      POSTGRES_USER: user
      POSTGRES_PASSWORD: 123
    deploy:
      replicas: 2
  adminer:
    image: adminer
    ports:
      - "8181:8080"
```

3) Запускаем

sudo docker compose up -d

```
Activities Terminator Oct 13 13:34
fred3@fred3VirtualBox:~/02$ sudo docker compose up -d
fred3@fred3VirtualBox:~/02$ ls
total 8
drwxrwxr-x 2 fred3 fred3 4096 Oct 13 13:18 ./
drwxr-xr-x 17 fred3 fred3 4096 Oct 13 13:18 ../
fred3@fred3VirtualBox:~/02$ sudo nano docker-compose.yml
fred3@fred3VirtualBox:~/02$ sudo docker-compose up -d
sudo: docker-compose: command not found
fred3@fred3VirtualBox:~/02$ sudo docker compose up -d
[+] Running 29/29
  ✓ adminer 7 layers [██████████]  0B/0B    Pulled          131.5s
    ✓ 69b3efbf67c2 Pull complete   47.0s
    ✓ 3a0fdd0809fb3 Pull complete   47.1s
    ✓ 689786bb0c396 Pull complete   42.4s
    ✓ 6bcf42cd6b65 Pull complete   44.4s
    ✓ d507b34ef5ac Pull complete   45.5s
    ✓ 04bbdd4f964a8 Pull complete   47.4s
    ✓ 7a75f8703c7 Pull complete   47.9s
  ✓ web 7 layers [██████████]  0B/0B    Pulled          83.9s
    ✓ a378f10b3218 Pull complete   17.0s
    ✓ 4dff0f708538 Pull complete   41.7s
    ✓ 2135e49ace4b Pull complete   6.9s
    ✓ c843f6b280ce Pull complete   2.3s
    ✓ 6739ab6f1400 Pull complete   3.5s
    ✓ 6c67a49fa4a Pull complete   1.2s
    ✓ f4773fb90a00 Pull complete   5.0s
  ✓ db 12 layers [██████████]  0B/0B    Pulled          139.5s
    ✓ 2ebc5098e391 Pull complete   47.7s
    ✓ 8fe5f7734687 Pull complete   49.1s
    ✓ a2ddbb09cd9a Pull complete   49.5s
    ✓ 5a2499e87ab8 Pull complete   52.0s
    ✓ a45f5c4adfb1 Pull complete   52.5s
    ✓ 178017ffd978e Pull complete   50.7s
    ✓ 428dff1c1b7d Pull complete   52.2s
    ✓ 4667364adfc4 Pull complete   82.3s
    ✓ 4eea1f5281a9 Pull complete   53.1s
    ✓ 369467411787 Pull complete   53.5s
    ✓ 51184495a2bc Pull complete   54.2s
    ✓ d3e246f01410 Pull complete   54.6s
[+] Running 5/5
  ✓ Network 02_default      created          0.6s
  ✓ Container 02-web-1      started         1.7s
  ✓ Container 02-db-2      started         1.7s
  ✓ Container 02-db-1      started         1.7s
  ✓ Container 02-adminer-1  started         1.6s
fred3@fred3VirtualBox:~/02$ sudo nano docker-compose.yml
```

4) Всего должно получиться 3 контейнера

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
b85a607fa2c9	postgres:latest	"docker-entrypoint.s..."	13 minutes ago	Up 13 minutes	5432/tcp
02-db-1	postgres:latest	"docker-entrypoint.s..."	13 minutes ago	Up 13 minutes	5432/tcp
616dd5627da6	postgres:latest	"docker-entrypoint.s..."	13 minutes ago	Up 13 minutes	5432/tcp
d0cc690c25442	nginx:latest	"/docker-entrypoint..."	13 minutes ago	Up 13 minutes	0.0.0.0:85->80/tcp, :::85->80/tcp
02-web-1	adminer	"entrypoint.sh php -..."	13 minutes ago	Up 13 minutes	0.0.0.0:8181->8080/tcp, :::8181->8080/tcp
92ff902c3f4e	01-web	"flask run"	24 minutes ago	Up 24 minutes	0.0.0.0:5082->80/tcp, :::5082->80/tcp
01-web-1	02-adminer-1				

5) проверяем localhost:8181 и 0.0.0.0:85

The image consists of two screenshots of a Linux desktop environment, likely Ubuntu, showing the results of a system configuration task.

Screenshot 1: PostgreSQL Database Management

This screenshot shows the Adminer 4.8.1 database management interface running in a Firefox browser window. The URL is `localhost:8181/?pgsql=db&username=user`. The interface displays a list of databases:

Database	Collation	Tables	Size	Compute
db	en_US.utf8	?	?	?
postgres	en_US.utf8	?	?	?
template0	en_US.utf8	?	?	?
template1	en_US.utf8	?	?	?

Screenshot 2: Nginx Welcome Page

This screenshot shows the Nginx welcome page running in a Firefox browser window. The URL is `0.0.0.0:85`. The page content is:

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.

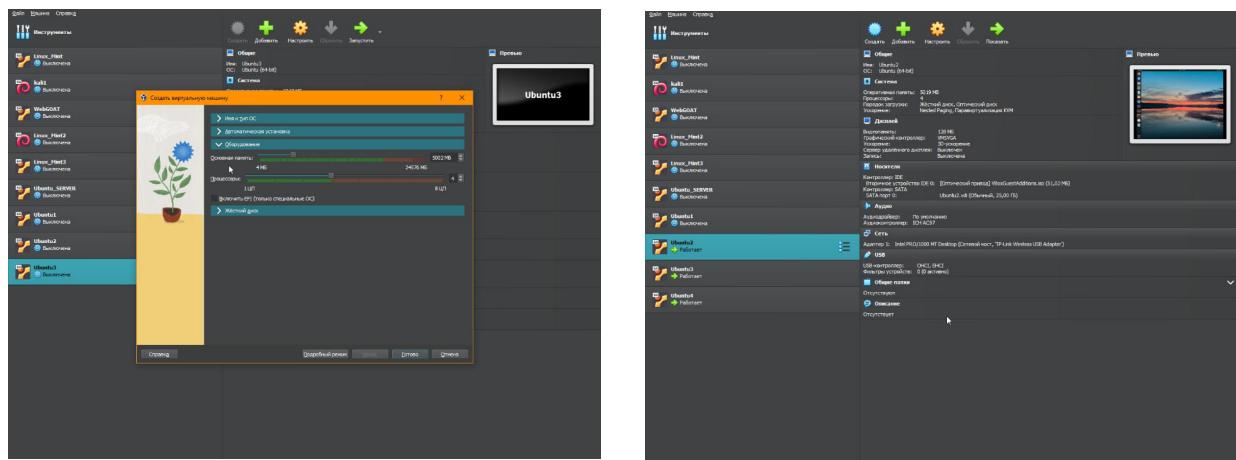
Commercial support is available at nginx.com.

Thank you for using nginx.

Задание 2*:

- 1) создать кластер и мастер и слейв ноды
- 2) задеплоить на ноду несколько экземпляров какого-нибудь контейнера, например nginx
- 3) обязательно проверить и зафиксировать результаты, чтобы можно было выслать преподавателю для проверки

1) создадим 3 машину, у меня Ubuntu4



2) Инициализируем Docker Swarm на мастер-ноде:

На вашем хосте выполните команду `docker swarm init` для инициализации Docker Swarm в роли мастер-ноды. Это создаст новый кластер Swarm с текущим хостом в качестве мастера.

```
sudo docker swarm init
```

```
|_> docker swarm join --token
```

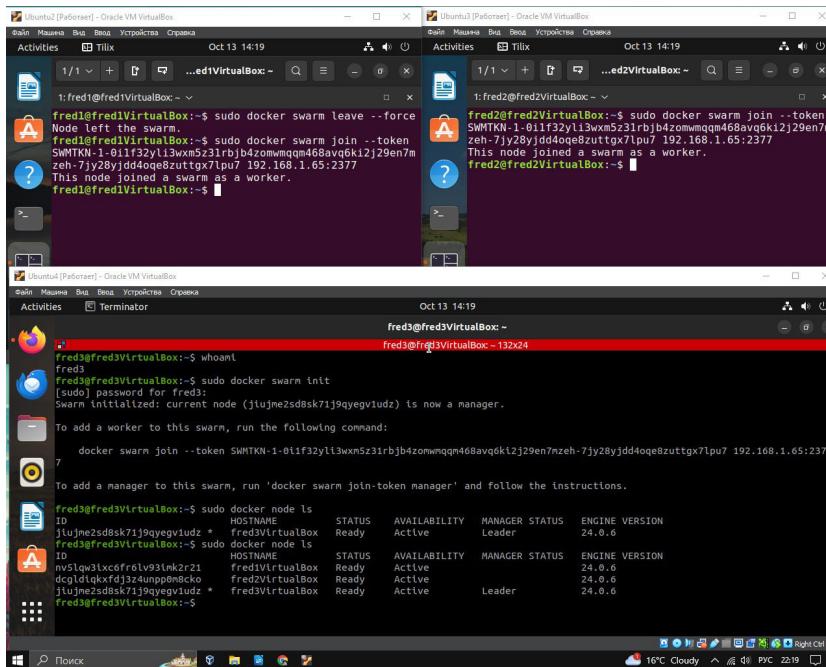
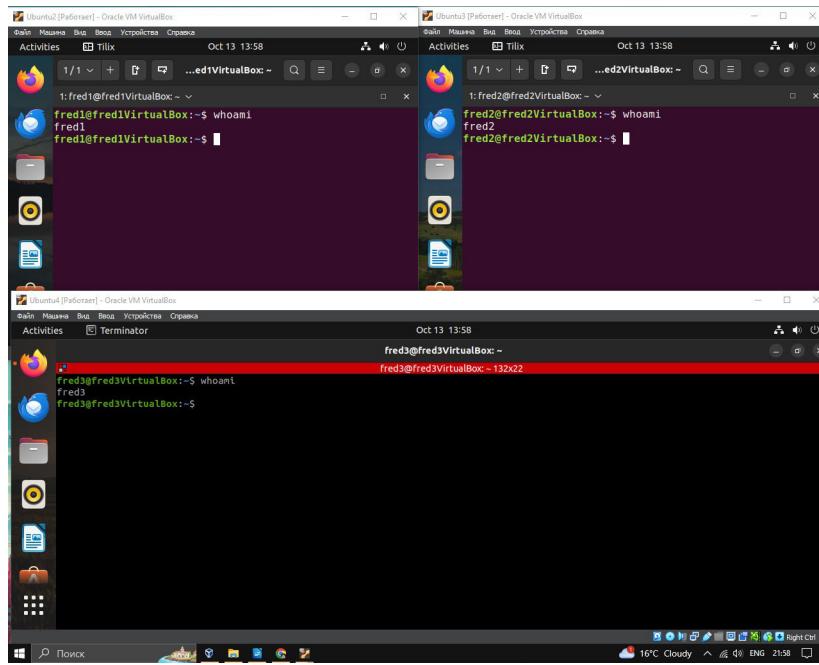
```
SWMTKN0i1f32yli3wxm5z31rbjb4zomwmqqm468avq6ki2j29en7mzeh-  
7jy28yjdd4oqe8zuttx7lpu7 192.168.1.65:2377
```

3) Присоединяемся к слейв-нодам к кластеру:

На слейв-нодах выполните команду `docker swarm join`, чтобы присоединить их к созданному кластеру

```
sudo docker swarm join --token SWMTKN-1-
```

```
0i1f32yli3wxm5z31rbjb4zomwmqqm468avq6ki2j29en7mzeh-  
7jy28yjdd4oqe8zuttx7lpu7 192.168.1.65:2377
```



3) все остальные действия на мастер-ноде, меняем имена нодам
`sudo docker node ls`

`sudo docker node update --label-add lab nv5lw3ixc6fr6lv93imk2r21`

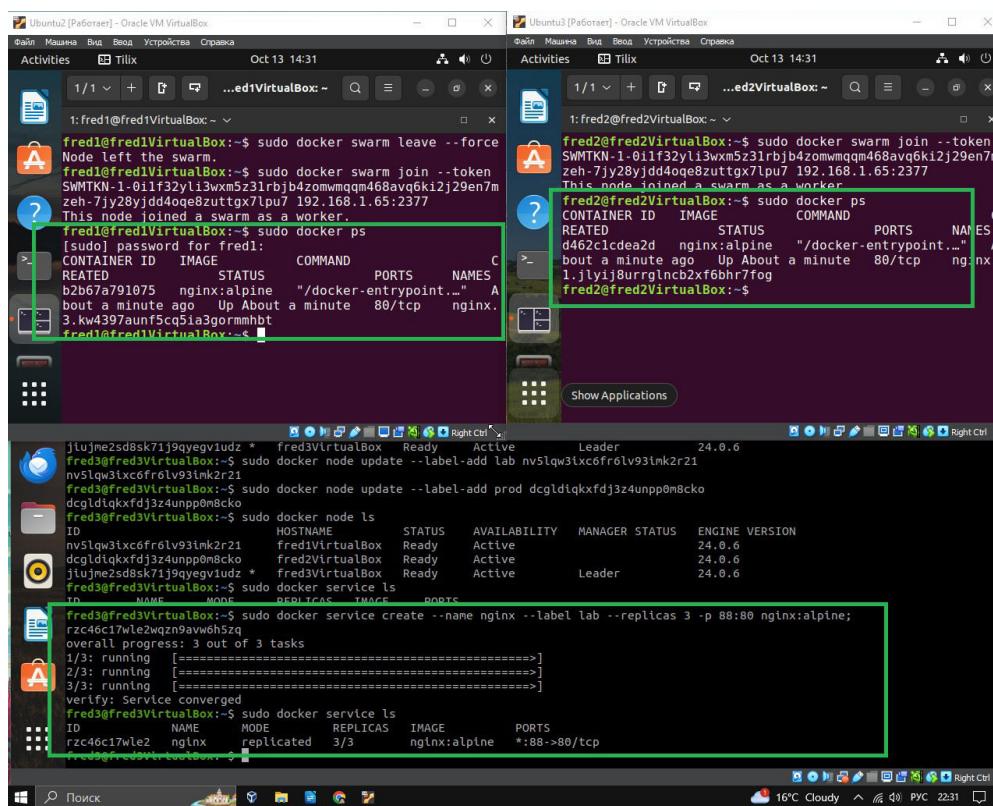
`sudo docker node update --label-add prod dcgldiqkxfdj3z4unpp0m8cko`

4) После того как создали и присоединили слейв-ноды к кластеру, мы можем создать сервисы на мастер-ноде, и Docker Swarm автоматически управляет ими и разместит их на слейв-нодах

```
sudo docker service create --name nginx --label lab --replicas 3 -p 88:80  
nginx:alpine;
```

```
sudo docker service create --name db-service --replicas 3 \  
-e POSTGRES_DB=db \  
-e POSTGRES_USER=user \  
-e POSTGRES_PASSWORD=123 \  
postgres:latest
```

```
docker service create --name adminer-service -p 8282:8080 adminer
```



Ubuntu2 [Рабочая] - Oracle VM VirtualBox

```
Activities Oct 13 14:40
1:fred1@fred1VirtualBox:~
```

```
zeh-7jy28ydd4oqe8zuttgx7lpu7 192.168.1.65:2377
A
CONTAINER ID IMAGE COMMAND
CREATED STATUS PORTS NAMES
b2b67a791075 nginx:alpine "/docker-entrypoint..." A
about a minute ago Up About a minute 80/tcp nginx.
3.kw4597uunr5cq5ia3grrmmht
fred1@fred1VirtualBox:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND
CREATED STATUS PORTS NAMES
87e1a6fd8a4b postgres:latest "docker-entrypoint..." A
52 seconds ago Up 49 seconds 5432/tcp db-service
3.wh9lwy952j00rlfbpp4K29xi
b2b67a791075 nginx:alpine "/docker-entrypoint..." A
10 minutes ago Up 10 minutes 80/tcp nginx.3.kw4597uunr5cq5ia3grrmmht
fred1@fred1VirtualBox:~$
```

```
overall progress: 3 out of 3 tasks
1/3: running [=====]>
2/3: running [=====]>
3/3: running [=====]>
```

```
verify: Service converged
```

```
fred3@fred3VirtualBox:~$ sudo docker service ls
ID NAME MODE REPLICAS IMAGE PORTS
rczc46c17wle2 replicated 3/3 nginx:alpine *:88->80/tcp
```

```
fred3@fred3VirtualBox:~$ sudo docker service create --name db-service --replicas 3 \
-e POSTGRES_DB=db \
-e POSTGRES_USER=user \
-e POSTGRES_PASSWORD=123 \
postgres:latest
bt0apf2pggjsh3y2975vs89f
overall progress: 3 out of 3 tasks
1/3: running [=====]>
2/3: running [=====]>
3/3: running [=====]>
```

```
verify: Service converged
```

```
fred3@fred3VirtualBox:~$ sudo docker service ls
ID NAME MODE REPLICAS IMAGE PORTS
bt0apf2pggj db-service replicated 3/3 postgres:latest *:88->80/tcp
rczc46c17wle2 nginx replicated 3/3 nginx:alpine *:88->80/tcp
```

Поиск

Activities Oct 13 14:43

Terminator

fred3@fred3VirtualBox:~

```
dgld1qkxfdj3z4unpp0m8cko
fred3@fred3VirtualBox:~$ sudo docker node ls
ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS ENGINE VERSION
nv5Lqw3lxcofr6lv93lmk2r21 fred1VirtualBox Ready Active 24.0.6
dgld1qkxfdj3z4unpp0m8cko fred2VirtualBox Ready Active 24.0.6
jujunezsdsks71j9qyev1udz * fred3VirtualBox Ready Active Leader 24.0.6
```

```
fred3@fred3VirtualBox:~$ sudo docker service ls
ID NAME MODE REPLICAS IMAGE PORTS
rczc46c17wle2 nginx replicated 3/3 nginx:alpine *:88->80/tcp
```

```
fred3@fred3VirtualBox:~$ sudo docker service create --name nginx --label lab --replicas 3 -p 88:80 nginx:alpine;
rzc46c17wle2wqzn9awwh5zq
overall progress: 3 out of 3 tasks
1/3: running [=====]>
2/3: running [=====]>
3/3: running [=====]>
```

```
verify: Service converged
```

```
fred3@fred3VirtualBox:~$ sudo docker service ls
ID NAME MODE REPLICAS IMAGE PORTS
rczc46c17wle2 nginx replicated 3/3 nginx:alpine *:88->80/tcp
```

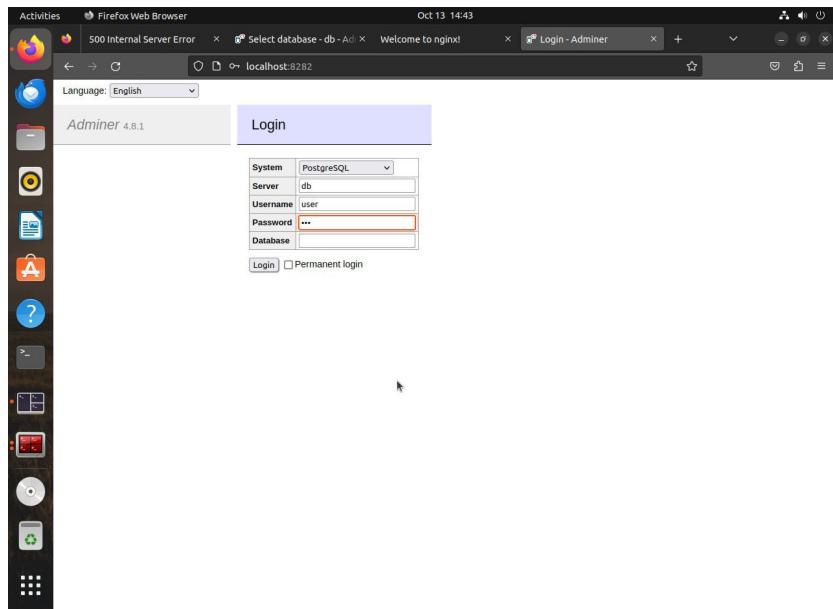
```
fred3@fred3VirtualBox:~$ sudo docker service create --name db-service --replicas 3 \
-e POSTGRES_DB=db \
-e POSTGRES_USER=user \
-e POSTGRES_PASSWORD=123 \
postgres:latest
bt0apf2pggjsh3y2975vs89f
overall progress: 3 out of 3 tasks
1/3: running [=====]>
2/3: running [=====]>
3/3: running [=====]>
```

```
verify: Service converged
```

```
fred3@fred3VirtualBox:~$ sudo docker service ls
ID NAME MODE REPLICAS IMAGE PORTS
bt0apf2pggj db-service replicated 3/3 postgres:latest *:88->80/tcp
rczc46c17wle2 nginx replicated 3/3 nginx:alpine *:88->80/tcp
fred3@fred3VirtualBox:~$ docker service create --name adminer-service -p 8282:8080 adminer
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fd
cker.sock/v1.24/services/create": dial unix /var/run/docker.sock: connect: permission denied
fred3@fred3VirtualBox:~$ sudo docker service create --name adminer-service -p 8282:8080 adminer
7ek22jw0pmehdohlvzspj8tf
overall progress: 1 out of 1 tasks
1/1: running [=====]>
```

```
verify: Service converged
```

```
fred3@fred3VirtualBox:~$ sudo docker service ls
ID NAME MODE REPLICAS IMAGE PORTS
7ek22jw0pm adminer-service replicated 1/1 adminer:latest *:8282->8080/tcp
bt0apf2pggj db-service replicated 3/3 postgres:latest *:88->80/tcp
rczc46c17wle2 nginx replicated 3/3 nginx:alpine *:88->80/tcp
```



И наконец можно добавить собственную верстку :)

1) mkdir 05 && cd 05

2) git clone

https://github.com/555-F-a-r-id-555/HTML_CSS_Seminar_5_DZ.git

3) cd HTML_CSS_Seminar_5_DZ/

4) sudo nano Dockerfile # В корневой директории нашего проекта создайте файл с #именем Dockerfile

```
|  
|_> FROM nginx:latest  
|_> COPY ./usr/share/nginx/html  
|_> EXPOSE 80
```

5) sudo nano docker-compose.yml # там же , создаем этот файл

```
|  
|_>version: '3'  
services:  
  web:  
    build: .  
    ports:  
      - "80:80"
```

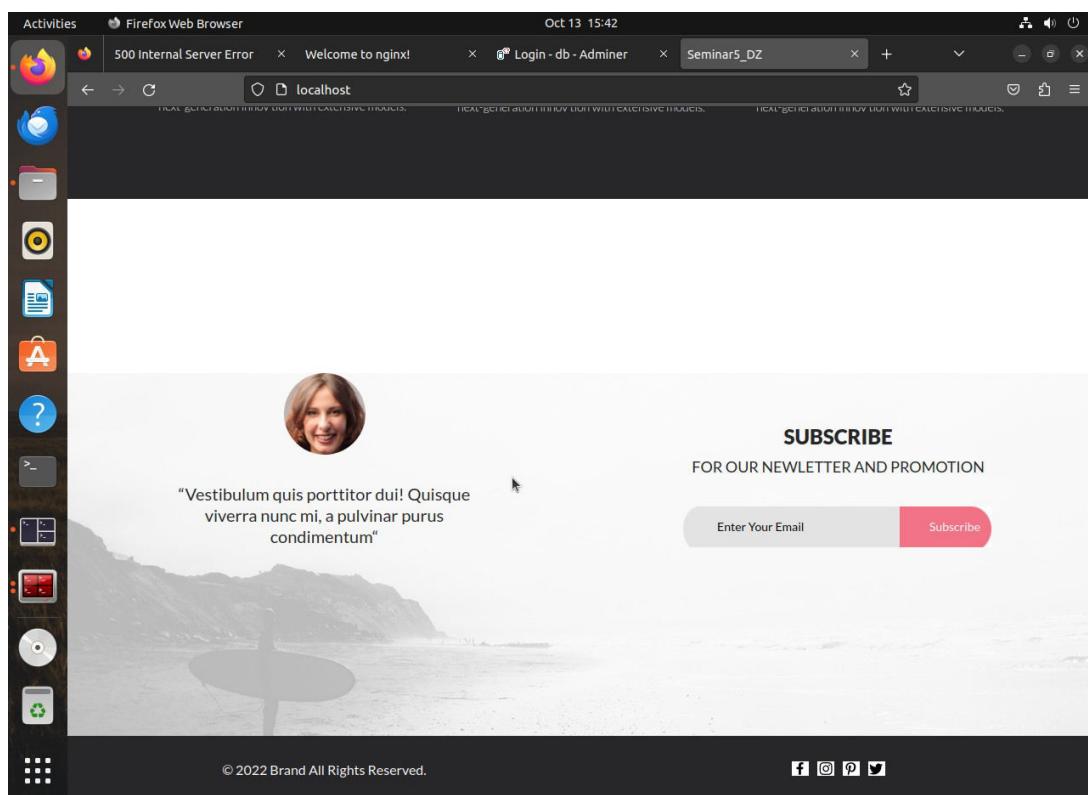
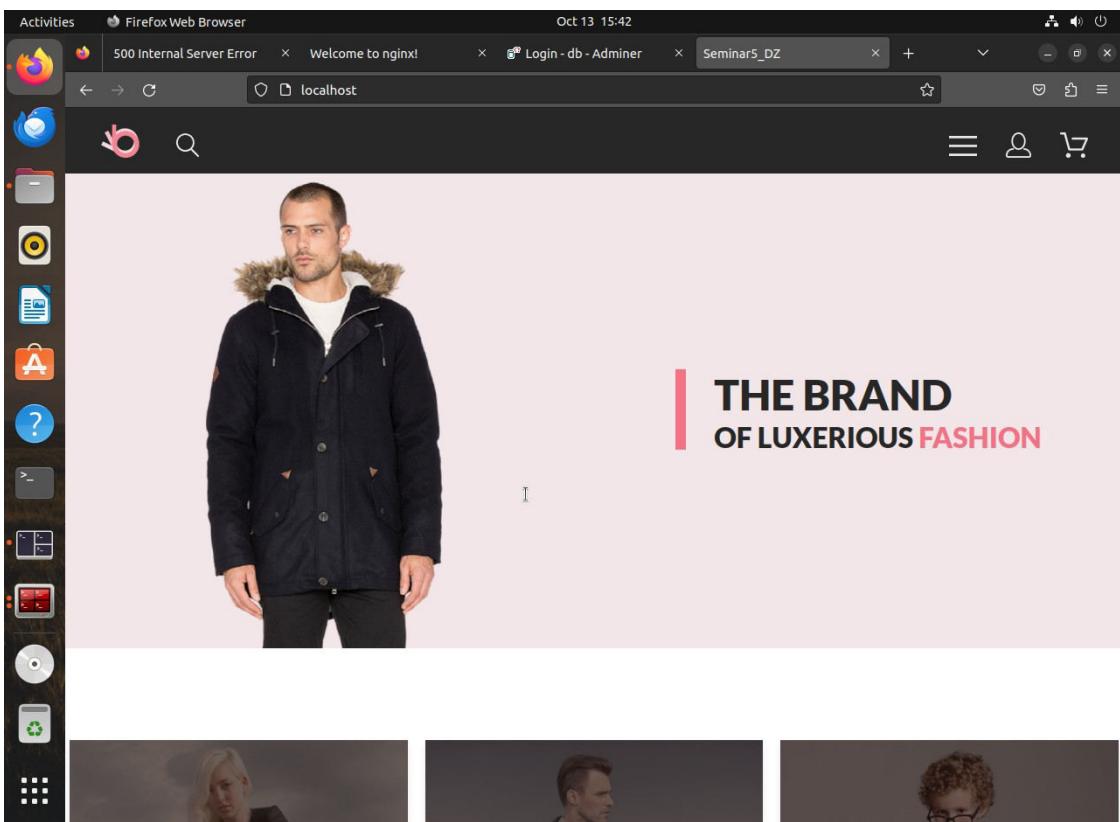
6) Собрать и запустить контейнер с использованием Docker Compose:
В директории, где находится ваш docker-compose.yml

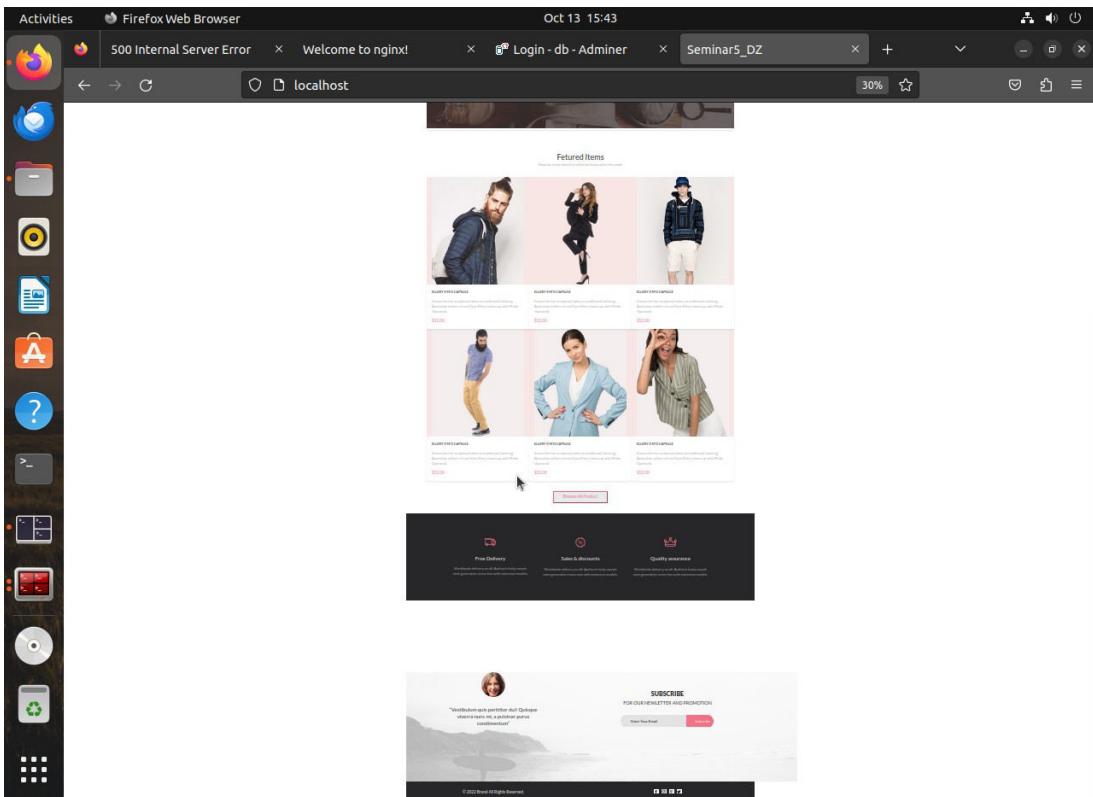
```
docker-compose up -d
```

```
Activities Terminator Oct 13 15:40
fred3@fred3VirtualBox:~/05/HTML_CSS_Seminar_5_DZ
fred3@fred3VirtualBox:~/05/HTML_CSS_Seminar_5_DZ$ git clone https://github.com/555-F-a-r-id-555/HTML_CSS_Seminar_5_DZ.git
Cloning into 'HTML_CSS_Seminar_5_DZ'...
remote: Enumerating objects: 99, done.
remote: Counting objects: 99, done.
remote: Compressing objects: 100% (99/99), done.
remote: Total 99 (delta 23), reused 80 (delta 10), pack-reused 0
Receiving objects: 100% (99/99), 2.63 MB | 371.00 KB/s, done.
Resolving deltas: 100% (23/23), done.
fred3@fred3VirtualBox:~/05/HTML_CSS_Seminar_5_DZ$ cd HTML_CSS_Seminar_5_DZ/
fred3@fred3VirtualBox:~/05/HTML_CSS_Seminar_5_DZ$ ll
total 10
drwxrwxr-x 10 fred3 fred3 4096 Oct 13 15:36 .
drwxrwxr-x 3 fred3 fred3 4096 Oct 13 15:36 ..
drwxrwxr-x 8 fred3 fred3 4096 Oct 13 15:36 .git/
drwxrwxr-x 3 fred3 fred3 4096 Oct 13 15:36 .github/
drwxrwxr-x 2 fred3 fred3 4096 Oct 13 15:36 lms/
drwxrwxr-x 2 fred3 fred3 4096 Oct 13 15:36 lms-footer/
drwxrwxr-x 2 fred3 fred3 4096 Oct 13 15:36 lms-phone/
drwxrwxr-x 2 fred3 fred3 4096 Oct 13 15:36 lms_tablet/
-rw-rw-r-- 1 Fred3 Fred3 13589 Oct 13 15:36 index.html
-rw-rw-r-- 1 Fred3 Fred3 3374 Oct 13 15:36 README.md
drwxrwxr-x 2 Fred3 Fred3 4096 Oct 13 15:36 style/
drwxrwxr-x 2 Fred3 Fred3 4096 Oct 13 15:36 test/
fred3@fred3VirtualBox:~/05/HTML_CSS_Seminar_5_DZ$ sudo nano Dockerfile
fred3@fred3VirtualBox:~/05/HTML_CSS_Seminar_5_DZ$ sudo nano docker-compose.yml
fred3@fred3VirtualBox:~/05/HTML_CSS_Seminar_5_DZ$ docker-compose up -d
Command 'docker-compose' not found, but can be installed with:
  sudo snap install docker           # version 20.10.24, or
  sudo snap install docker            # version 20.10.24
  sudo snap install docker            # version 20.09.21
See 'snap info <name>' for additional versions.
fred3@fred3VirtualBox:~/05/HTML_CSS_Seminar_5_DZ$ sudo docker compose up -d
sudo: docker-compose: command not found
fred3@fred3VirtualBox:~/05/HTML_CSS_Seminar_5_DZ$ sudo docker compose up -d
[*] Building 3.3s (7/7) FINISHED
  == [web] transferring context: 6.34MB
  == [web] transferring dockerfile: 96B
  == [web Internal] load .dockerignore
  == [web Internal] load metadata for docker.io/library/nginx:latest
  == [web Internal] load build context
  == [web Internal] transfer config: 6.34MB
  == [web /] FROM docker.io/library/nginx:latest
  == [web /] COPY ./user/share/nginx/html
  == [web] exporting to image
  == [web] exporting layers
  == [web] docker default
```

```
Activities Terminator Oct 13 15:42
fred3@fred3VirtualBox:~/05/HTML_CSS_Seminar_5_DZ
fred3@fred3VirtualBox:~/05/HTML_CSS_Seminar_5_DZ 132x47

[+] [web] exporting to image
[+] >>> exporting layers
[+] >>> writing image sha256:4a978c140c89e2b1943c96f94acb6240b095babac02159617a8d28faec5ceeca
[+] >>> naming to docker.sh/library/html_css_seminar_5_dz/web
[*] Running 2/2
  Network: html_css_seminar_5_dz.default
    Container: html_css_seminar_5_dz-web
    Status: Created
    Started
[*] DockerfileVirtualBox:~/05/HTML_CSS_Seminar_5_DZ$ curl localhost
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Seminars.Dz</title>
    <link rel="stylesheet" href="./style/style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@0,300;0,400;0,900;1,700&family=Poppins:wght@100&display=swap" rel="stylesheet">
  </head>
  <body>
    <header class="header center">
      <ul class="ul_center">
        <li class="li1 "></li>
        <li class="li2 "></li>
        <li class="li3 ">
          <picture>
            <source media="(max-width:500px)" srcset=".//img_header_right.png">
            
          </picture>
        </li>
      </ul>
    </header>
    <div class="header_top_img_center">
      <div class="header_img_container center">
```





```
version: '3'
```

```
services:  
  web:  
    image: nginx:latest  
    ports:  
      - 8755:80  
    volumes:  
      - ./web:/usr/share/nginx/html
```

```
database:  
  image: postgres:latest  
  volumes:  
    - ./db:/var/lib/postgresql/data  
environment:  
  POSTGRES_DB: db  
  POSTGRES_USER: user  
  POSTGRES_PASSWORD: 123  
deploy:  
  mode: replicated  
  replicas: 2
```

```
version: '3'

services:
  web:
    build: ./web_app # Путь к папке с Dockerfile
    ports:
      - 8787:80 # Проброс портов (localhost:8787 -> контейнер:80)
  deploy:
    mode: replicated
    replicas: 2 # Создаем 2 реплики контейнера веб-приложения
```

```
version: '3'
services:
  web:
    image: nginx:latest
    ports:
      - "85:80"
  db:
    image: postgres:latest
    environment:
      POSTGRES_DB: db
      POSTGRES_USER: user
      POSTGRES_PASSWORD: 123
  deploy:
    replicas: 2
  adminer:
    image: adminer
    ports:
      - "8181:8080"
```

СПАСИБО, ВСЕГО ХОРОШЕГО !)