

Урок 5. Docker Compose и Docker Swarm

Задание 1:

- 1) создать сервис, состоящий из 2 различных контейнеров: 1 - веб, 2 - БД
- 2) далее необходимо создать 3 сервиса в каждом окружении (dev, prod, lab)
- 3) по итогу на каждой ноде должно быть по 2 работающих контейнера
- 4) выводы зафиксировать

Задание 2*:

- 1) нужно создать 2 ДК-файла, в которых будут описываться сервисы
- 2) повторить задание 1 для двух окружений: lab, dev
- 3) обязательно проверить и зафиксировать результаты, чтобы можно было выслать преподавателю для проверки

Задание со звездочкой - повышенной сложности, это нужно учесть при выполнении (но сделать его необходимо).

Задание 1:

- 1) создать сервис, состоящий из 2 различных контейнеров: 1 - веб, 2 - БД
- 2) далее необходимо создать 3 сервиса в каждом окружении (dev, prod, lab)
- 3) по итогу на каждой ноде должно быть по 2 работающих контейнера
- 4) выводы зафиксировать

Создаём директорию и заходим в неё

```
mkdir hw_5 && cd hw_5
```

Создаем файл

```
nano docker-compose.yaml
```

Вставляем, пробелы важны!

```
version: '3.9' #версия
```

```
services: #сервисы- контейнеры и их конфигурация
```

```
mysql: # название сервиса любое название
  image: mysql:8.0.34 #образ mysql:8.0.34
  restart: always #параметр перезапуска
  container_name: mysql_new
  environment: #переменные
    MYSQL_ROOT_PASSWORD: 111
```

```
#Название переменной и её значение
```

```
phpmyadmin:
  image: phpmyadmin:5.2.1-apache
  restart: always
  container_name: phpmyadmin
  ports:
```

- 9080:80 #проброс портов слева у хоста справа у контейнера

environment:

- PMA_ARBITRARY=1 #Переменная среды -PMA_ARBITRARY=1 обозначает, что можно подключаться к любому серверу сети, а не только localhos

nano docker-compose.yaml

```

root@UB: /home/user/hw_5
GNU nano 6.2 docker-compose.yaml
version: '3.9' #версия

services: #сервисы- контейнеры и их конфигурация

mysql: # название сервиса любое название
  image: mysql:8.0.34 #образ mysql:8.0.34
  restart: always #параметр перезапуска
  container_name: mysql_new
  environment: #переменные
    MYSQL_ROOT_PASSWORD: 111
#Название переменной и её значение

phpmyadmin:
  image: phpmyadmin:5.2.1-apache
  restart: always
  container_name: phpmyadmin
  ports:
    - 9080:80 #проброс портов слева у хоста справа у контейнера
  environment:
    - PMA_ARBITRARY=1 #Переменная среды -PMA_ARBITRARY=1 обозначает, что можно подключаться к любому серверу сети, а не только localhos

```

Запускаем, docker ищет файл docker-compose.yaml, если нужно другое название файла нужно добавить параметр

docker compose up -d (-d режим демона)

Смотрим контейнеры

docker ps

```

root@UB:/home/user/hw_5# docker compose up -d
[*] Running 2/2
  phpmyadmin 19 layers [#####] 0B/0B Pulled
  648e0a4df75a Pull complete
  318af44fe185 Pull complete
  ed9eb75e2f2e Pull complete
  bb6a1589be26 Pull complete
  46b9d509db51 Pull complete
  c1b069ee18a8 Pull complete
  fc30644a7c8 Pull complete
  116ffa5d03f3 Pull complete
  2dc3ab5137dc Pull complete
  262b5055c905 Pull complete
  1ea65da8eabf Pull complete
  51ee24cedb04 Pull complete
  2107e32d4833 Pull complete
  8579f088cd98 Pull complete
  96b43db1caf0 Pull complete
  42e781b86ddb Pull complete
  76d727f16d8c Pull complete
  560c533fc597 Pull complete
[*] Running 3/3
  Network hw_5_default Created
  Container phpmyadmin Started
  Container mysql_new Started
root@UB:/home/user/hw_5# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
72bald9be130   phpmyadmin:5.2.1-apache "/docker-entrypoint..." 4 minutes ago Up 4 minutes 0.0.0.0:9080->80/tcp, :::9080->80/tcp
8b5ca55bfdc0   mysql:8.0.34   "docker-entrypoint.s..." 4 minutes ago Up 4 minutes 3306/tcp, 33060/tcp
root@UB:/home/user/hw_5# ip a

```

Смотрим ip

ip a

```

root@UB:/home/user/hw_5# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 link/ether 08:00:27:fd:09:c7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.120/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::208:1:ff:fe09:c7:5d/64 scope link
        valid_lft forever preferred_lft forever

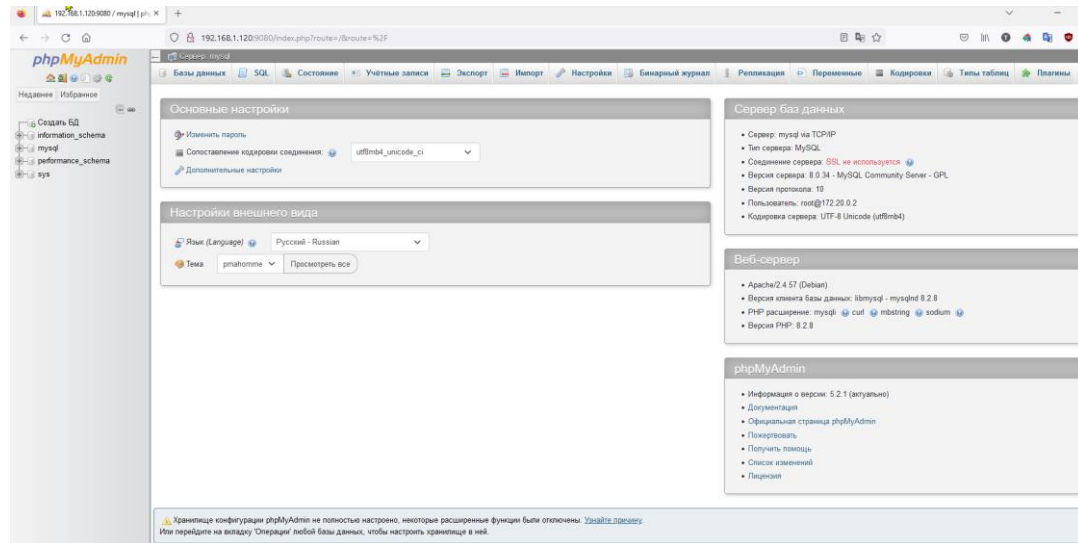
```

Заходим <http://192.168.1.120:9080/>

Сервер *mysql*

Имя *root*

Пароль *111*

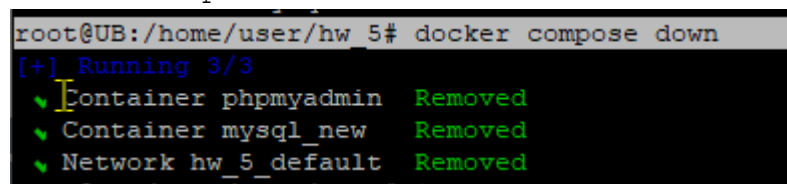


2) Далее необходимо создать 3 Сервиса в каждом Окружении (Dev, Prod, Lab)

3) по итогу на каждой ноду должно быть по 2 работающих контейнера

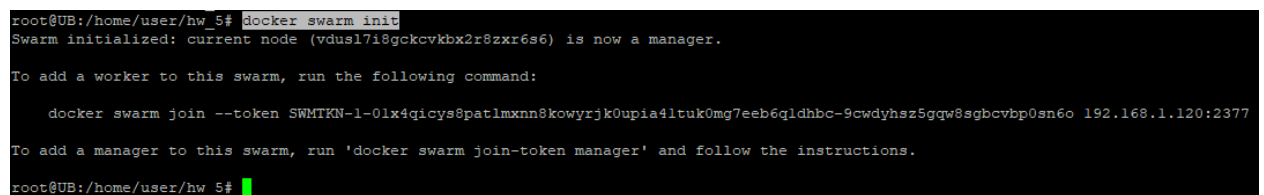
Удаляем старые контейнеры

docker compose down

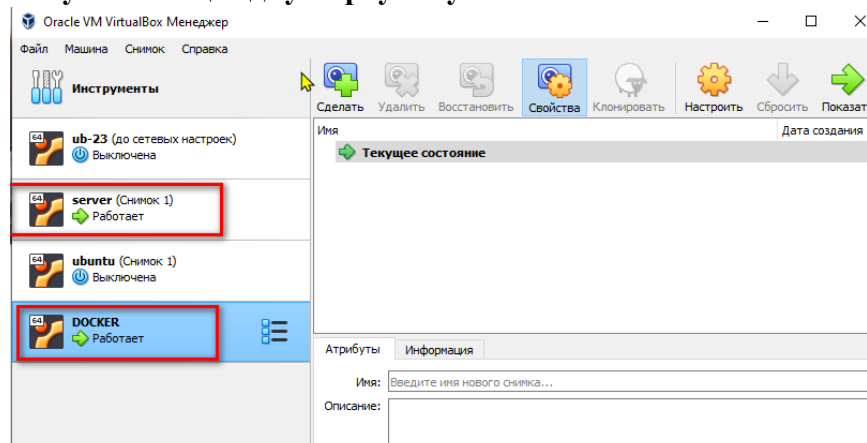


Создаем главную ноду

docker swarm init



Запускаем еще одну виртуалку



Подключаемся ко второй машине и создаем еще ноду

```
docker swarm join --token SWMTKN-1-01x4qicys8patlmxnn8kowyrjk0upia41tuk0mg7eeb6q1dhbc-9cwyhsz5gqw8sgbcvbp0sn6o 192.168.1.120:2377
```

Возвращаемся на первую машину с главной нодой и смотрим список нод

```
docker node ls
```

```
user@UB:~$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
vdusl7i8gckcvkxbx2r8zxr6s6 *	UB	Ready	Active	Leader	24.0.4
5yciweujj3o7lqffpu6irjb19	server-ubuntu	Ready	Active		20.10.24

Задаем главной ноде хоста (UB) label Prod

```
docker node update --label-add env=Prod  
vdusl7i8gckcvkxbx2r8zxr6s6
```

Обновляем настройки ноды

```
docker node update vdusl7i8gckcvkxbx2r8zxr6s6
```

Смотрим параметры ноды

```
docker node inspect vdusl7i8gckcvkxbx2r8zxr6s6
```

```
user@UB:~$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
vdusl7i8gckcvkxbx2r8zxr6s6 *	UB	Ready	Active	Leader	24.0.4
5yciweujj3o7lqffpu6irjb19	server-ubuntu	Ready	Active		20.10.24

```
user@UB:~$ docker node inspect vdusl7i8gckcvkxbx2r8zxr6s6
```

```
[
  {
    "ID": "vdusl7i8gckcvkxbx2r8zxr6s6",
    "Version": {
      "Index": 89
    },
    "CreatedAt": "2023-07-31T07:58:10.361491609Z",
    "UpdatedAt": "2023-07-31T13:41:21.324829776Z",
    "Spec": {
      "Labels": {
        "env": "Prod"
      },
      "Role": "manager",
      "Availability": "active"
    },
    "Description": {
      "Hostname": "UB",
```

Создаем сервис и связываем его с нодой Prod

```
docker service create --name phpmyadmin_service --constraint  
node.labels.env==Prod --replicas 2 phpmyadmin:latest
```

Задаем подчиненной ноде хоста (server-ubuntu) label Dev

```
docker node update --label-add env=Dev 5yciweujj3o7lqffpu6irjb19
```

Обновляем настройки ноды

```
docker node update 5yciweujj3o7lqffpu6irjb19
```

Смотрим параметры ноды

```
docker node inspect 5yciweujj3o7lqffpu6irjb19
```

Создаем еще один сервис и связываем его с нодой *Dev*

```
docker service create --name mariadb_service --constraint  
node.labels.env==Dev --replicas 2 -e MYSQL_ROOT_PASSWORD=12345 -  
p 3306:3306 mariadb:10.10.2
```

```
user@UB:~$ docker node ls
ID                               HOSTNAME        STATUS    AVAILABILITY    MANAGER STATUS    ENGINE VERSION
vdusl7i8gckcvkxbx2r8zxr6s6 *   UB              Ready     Active           Leader             24.0.4
5yciweujj3o7lqffpu6irjb19      server-ubuntu   Ready     Active           Leader             20.10.24
user@UB:~$ docker node update --label-add env=Dev 5yciweujj3o7lqffpu6irjb19
5yciweujj3o7lqffpu6irjb19
user@UB:~$ docker node update 5yciweujj3o7lqffpu6irjb19
5yciweujj3o7lqffpu6irjb19
user@UB:~$ docker node inspect 5yciweujj3o7lqffpu6irjb19
[
  {
    "ID": "5yciweujj3o7lqffpu6irjb19",
    "Version": {
      "Index": 120
    },
    "CreatedAt": "2023-07-31T08:06:17.138074114Z",
    "UpdatedAt": "2023-07-31T13:46:44.047844427Z",
    "Spec": {
      "Labels": {
        "env": "Dev"
      },
      "Role": "worker",
      "Availability": "active"
    }
  }
]
```

Проверяем что сервисы на обеих нодах работают так как мы их связали

docker ps на главной ноде

```
user@UB:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
1a0c8244b986   phpmyadmin:latest  "/docker-entrypoint..." 32 minutes ago Up 32 minutes 80/tcp  phpmyadmin_service.2
a58c9bede072   phpmyadmin:latest  "/docker-entrypoint..." 36 minutes ago Up 36 minutes 80/tcp  phpmyadmin_service.1
```

docker ps на рабочей ноде

```
root@server-ubuntu:/home/user# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
3827549c0fbb   mariadb:10.10.2  "docker-entrypoint.s..." 33 minutes ago Up 33 minutes 3306/tcp mariadb_service.2
xly8qmknmrajt   mariadb:10.10.2  "docker-entrypoint.s..." 33 minutes ago Up 33 minutes 3306/tcp mariadb_service.1
```

Задание 2*:

- 1) нужно создать 2 ДК-файла, в которых будут описываться сервисы
- 2) повторить задание 1 для двух окружений: *lab*, *dev*
- 3) обязательно проверить и зафиксировать результаты, чтобы можно было выслать преподавателю для проверки

Создаем и переходим в папку *task2*

```
mkdir task2
```

```
cd task2
```

```
nano docker-compose.yaml
```

```
user@UB:~/hw_5$ mkdir task2
user@UB:~/hw_5$ cd task2/
user@UB:~/hw_5/task2$ ^C
user@UB:~/hw_5/task2$ nano docker-compose.yaml
```

Вставляем, пробелы важны!

```
version: '3.9' #версия
```

```
services: #сервисы- контейнеры и их конфигурация
```

```

mariadb: # название сервиса любое название
image: mariadb:11.1.1-rc-jammy #образ mariadb
deploy:
  replicas: 4
  restart: always #параметр перезапуска
  container_name: mariadb_new
  environment: #переменные
    MYSQL_ROOT_PASSWORD: 111
#Название переменной и её значение

adminer:
  image: adminer:4.8.1-standalone
  deploy:
    replicas: 2
    restart: always
    container_name: adminer
  ports:
    - 9080:8080 #проброс портов слева у хоста справа у
контейнера

```

The screenshot shows a terminal window with the title 'GNU nano 6.2' and a file named 'doc'. The content of the file is a Docker Compose configuration for two services: 'mariadb' and 'adminer'. The 'mariadb' service is configured with 4 replicas, always restarts, and uses the 'mariadb_new' container name. The 'adminer' service is configured with 2 replicas, always restarts, and uses the 'adminer' container name. Both services have their ports mapped to 9080 on the host and 8080 on the container. The configuration is stored in an environment variable named 'MYSQL_ROOT_PASSWORD' with the value '111'.

```

GNU nano 6.2 doc
version: '3.9' #версия

services: #сервисы- контейнеры и их конфигурация

  mariadb: # название сервиса любое название
    image: mariadb: 11.1.1-rc-jammy #образ mariadb
    deploy:
      replicas: 4
      restart: always #параметр перезапуска
      container_name: mariadb_new
      environment: #переменные
        MYSQL_ROOT_PASSWORD: 111
#Название переменной и её значение

  adminer:
    image: adminer:4.8.1-standalone
    deploy:
      replicas: 2
      restart: always
      container_name: adminer
    ports:
      - 9080:8080 #проброс портов слева у хоста справа у контейнера

```

Обновляем Docker compose

```

sudo curl -L
https://github.com/docker/compose/releases/latest/download/docke
r-compose-$(uname -s)-$(uname -m) " -o /usr/local/bin/docker-
compose

```

Даем права

```

sudo chmod +x /usr/local/bin/docker-compose

```

Запускаем сервисы в стеке

```

docker stack services docker-compose.yml app

```

Ошибка из-за несовместимости версий, решить проблему не удалось!

```

root@UB:/home/user/hw_5/task2# docker stack deploy --compose-file docker-compose.yml app
unsupported Compose file version: '3.9'
root@UB:/home/user/hw_5/task2#

```