

Урок 5. Docker Compose и Docker Swarm

Задание

- 1) создать сервис, состоящий из 2 различных контейнеров: 1 - веб, 2 - БД
- 2) В каждой ноде должно быть по 2 работающих контейнера

Выполнение

- 1) создать сервис, состоящий из 2 различных контейнеров: 1 - веб, 2 - БД

vi docker-compose.yml - создаем yaml файл и помещаем данные

```
version: "3.1"

services:

  db:
    image: mariadb:10.10.2
    restart: always
    environment:
      MARIADB_ROOT_PASSWORD: test123

  adminer:
    image: adminer:4.8.1
    restart: always
    ports:
      - 6080:8080
```

sudo docker compose up -d - запускаем контейнеры

```
nik@ubuntu-server:~$ sudo docker compose up -d
[+] Running 4/4
  ✓ Container nik_adminer_1   Recreated   0.3s
  ✓ Container nik_db_1       Recreated   0.6s
  ✓ Container nik-adminer-1   Started     1.0s
  ✓ Container nik-db-1       Started     0.9s
nik@ubuntu-server:~$
```

sudo docker ps -a - посмотрим содержимое

```
nik@ubuntu-server:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
6d19399937c   adminer:4.8.1 "entrypoint.sh php -..." 4 minutes ago Up 4 minutes  0.0.0.0:6080->8080/tcp, :::6080->8080/tcp  nik-adminer-1
8922261956f6   mariadb:10.10.2 "docker-entrypoint.s..." 4 minutes ago Up 4 minutes  3306/tcp                               nik-db-1
nik@ubuntu-server:~$
```

sudo docker container inspect nik-adminer-1 - проинспектируем контейнер adminer

sudo docker container inspect nik-db-1 - проинспектируем контейнер db

Для проверки работоспособности базы mariadb переходим в браузер и вводим: 192.168.1.58:6080. Появляется окно для ввода логина и пароля. Вводим логин – root, пароль - test123 -> enter

← → ↻

⚠ Не защищено | http://192.168.1.58:6080

Язык:

Русский

Adminer 4.8.1

Войти

Движок

MySQL

Сервер

db

Имя пользователя

root

Пароль

.....

База данных

Войти

☐ Остаться в системе

← → ↻

⚠ Не защищено | http://192.168.1.58:6080/?server=db&username=root

Язык:

Русский

MySQL » db

Adminer 4.8.1

Выбрать базу данных

DB:

Создать базу данных

Полномочия

Список процессов

Переменные

Состояние

Версия MySQL: 5.5.5-10.10.2-MariaDB-1:10.10.2+maria~ubu2204 с PHP-расширением MySQLi

Вы вошли как: root@172.18.0.3

	База данных - Обновить	Режим сопоставления	Таблицы	Размер - Вычислить
<input type="checkbox"/>	information_schema	utf8mb3_general_ci	?	?
<input type="checkbox"/>	mysql	utf8mb4_general_ci	?	?
<input type="checkbox"/>	performance_schema	utf8mb3_general_ci	?	?
<input type="checkbox"/>	sys	utf8mb3_general_ci	?	?

Выбранные (0)

Удалить

2) В каждой ноде должно быть по 2 работающих контейнера

Создаем кластер из трех нод. Запускаем еще две ноды (виртуальные машины) – docker2 и docker3.

В терминале виртуальной машины nik@ubuntu-server (основной) вводим команду - `sudo docker swarm init`

```
nik@ubuntu-server:~$ sudo docker swarm init
[sudo] password for nik:
Swarm initialized: current node (pvot9osa9afz9cff5bj240gye) is now a manager.

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

    docker swarm join --token SWMTKN-1-5h4nsnmcevwkokfvpqclds0v7qsqyc004phde9c7ukbcmq17k-9t4a3lm5fo0non1cqjvaomm04 192.168.1.58:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
nik@ubuntu-server:~$
```

В выводе команды присутствует команда, с помощью которой сможем объединить другие две ноды (docker2 и docker3) в кластер - `docker swarm join --token SWMTKN-1-5h4nsnmcevwkokfvpqclds0v7qsqyc004phde9c7ukbcmq17k-9t4a3lm5fo0non1cqjvaomm04 192.168.1.58:2377`

`sudo docker node ls` – смотрим, какие ноды сейчас входят в кластер

```
nik@ubuntu-server:~$ sudo docker node ls
[sudo] password for nik:
ID                                HOSTNAME            STATUS    AVAILABILITY    MANAGER STATUS    ENGINE VERSION
pvot9osa9afz9cff5bj240gye *      ubuntu-server      Ready    Active           Leader             24.0.4
nik@ubuntu-server:~$
```

В терминале docker2 вводим - `docker swarm join --token SWMTKN-1-5h4nsnmcevwkokfvpqclds0v7qsqyc004phde9c7ukbcmq17k-9t4a3lm5fo0non1cqjvaomm04 192.168.1.58:2377` и она добавилась в наш кластер как worker

```
nik@docker2:~$ sudo docker swarm join --token SWMTKN-1-5h4nsnmcevwkokfvpqclds0v7qsqyc004phde9c7ukbcmq17k-9t4a3lm5fo0non1cqjvaomm04 192.168.1.58:2377
[sudo] password for nik:
This node joined a swarm as a worker.
nik@docker2:~$
```

В терминале docker3 также вводим - `docker swarm join --token SWMTKN-1-5h4nsnmcevwkokfvpqclds0v7qsqyc004phde9c7ukbcmq17k-9t4a3lm5fo0non1cqjvaomm04 192.168.1.58:2377` и она также добавилась в наш кластер как worker

```
nik@docker3:~$ docker swarm join --token SWMTKN-1-5h4nsnmcevwkokfvpqclds0v7qsqyc004phde9c7ukbcmq17k-9t4a3lm5fo0non1cqjvaomm04 192.168.1.58:2377
This node joined a swarm as a worker.
nik@docker3:~$
```

sudo docker node ls – смотрим в основной ноде, какие ноды сейчас входят в кластер

```
nik@ubuntu-server:~$ sudo docker node ls
[sudo] password for nik:
ID                                HOSTNAME    STATUS    AVAILABILITY    MANAGER STATUS    ENGINE VERSION
n89b7v29748faoh9ex9rbny0g        docker2     Ready     Active           Leader            24.0.4
8vz6vwtj5arnsbyic3u0de5fc        docker3     Ready     Active           Leader            24.0.4
pvot9osa9afz9cff5bj240gye *      ubuntu-server Ready     Active           Leader            24.0.4
nik@ubuntu-server:~$
```

Далее в ноде docker2 создаем yaml файл - vi docker-compose.yaml

cat docker-compose.yaml – смотрим содержимое файла

```
nik@docker2:~$ cat docker-compose.yaml
version: "3.1"

services:

  db:
    image: mariadb:10.10.2
    restart: always
    environment:
      MARIADB_ROOT_PASSWORD: test123

  adminer:
    image: adminer:4.8.1
    restart: always
    ports:
      - 6080:8080

nik@docker2:~$
```

sudo docker compose up -d - в ноде docker2 запускаем контейнеры

```
nik@docker2:~$ sudo docker compose up -d
[+] Running 4/4
✔ Container nik_adminer_1    Recreated
✔ Container nik_db_1        Recreated
✔ Container nik-adminer-1    Started
✔ Container nik-db-1         Started
nik@docker2:~$
```


vi docker-compose.yaml - в ноде docker3 создаем yaml файл

cat docker-compose.yaml – смотрим содержимое файла

```
nik@docker3:~$ vi docker-compose.yaml
nik@docker3:~$ cat docker-compose.yaml
version: "3.1"

services:

  db:
    image: mariadb:10.10.2
    restart: always
    environment:
      MARIADB_ROOT_PASSWORD: test123

  adminer:
    image: adminer:4.8.1
    restart: always
    ports:
      - 6080:8080
```

sudo docker compose up -d - далее в ноде docker3 запускаем контейнеры

```
nik@docker3:~$ sudo docker compose up -d
[sudo] password for nik:
[+] Running 17/17
  ✓ adminer 7 layers [#####] 0B/0B Pulled 31.3s
    ✓ 34df401c391c Pull complete 23.3s
    ✓ 942860e9b081 Pull complete 28.1s
    ✓ f571177b537e Pull complete 28.1s
    ✓ 78d7a59571f8 Pull complete 28.3s
    ✓ 530e7e02f755 Pull complete 28.4s
    ✓ 03ee8734c62c Pull complete 28.6s
    ✓ ed7a0cc37cf2 Pull complete 28.6s
  ✓ db 8 layers [#####] 0B/0B Pulled 33.7s
    ✓ 10ac4908093d Pull complete 8.5s
    ✓ 44779101e748 Pull complete 8.6s
    ✓ a721db3e3f3d Pull complete 10.1s
    ✓ 1850a929b84a Pull complete 10.1s
    ✓ 397a918c7da3 Pull complete 10.1s
    ✓ 806be17e856d Pull complete 31.0s
    ✓ 634de6c90876 Pull complete 31.0s
    ✓ cd00854cfb1a Pull complete 31.0s
[+] Running 3/3
  ✓ Network nik_default Created 0.2s
  ✓ Container nik-db-1 Started 2.0s
  ✓ Container nik-adminer-1 Started 2.1s
nik@docker3:~$
```