

## Урок 5. Docker Compose и Docker Swarm, группа 4546, Сиденко Георгий

### 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

Для создания нового узла в кластере необходимо выполнить следующие шаги:

1. Подготовить новый узел, на котором будет запущен Docker Engine. Это может быть как физический сервер, так и виртуальная машина.
2. Установить Docker Engine на новый узел в соответствии с инструкциями для вашей операционной системы.
3. Получить токен для присоединения к кластеру с помощью команды "docker swarm join-token worker". Эта команда выведет токен, который необходимо скопировать для использования на новом узле.
4. Запустить на новом узле команду "docker swarm join" и передать ей скопированный токен. Например: "docker swarm join --token [token] [IP адрес главного узла]".

После выполнения этих шагов новый узел будет добавлен в кластер и будет готов к запуску контейнеров. Проверить состояние кластера и список узлов можно с помощью команды "docker node ls".

Docker Swarm позволяет управлять кластером из нескольких хостов. Например, если требуется масштабировать приложение, можно добавить новые узлы в кластер и запустить на них дополнительные контейнеры.

Пример файла docker-compose.yml для использования в Docker Swarm:

```
version: '3'
services:
  db:
    image: mysql
    environment:
      - ENVIRONMENT=dev
    deploy:
      replicas: 3
      placement:
        constraints:
          - node.labels.env == dev
  networks:
    - mynetwork
  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    environment:
      - ENVIRONMENT=dev
    deploy:
      replicas: 1
      placement:
        constraints:
          - node.labels.env == dev
```

```
networks:
  - mynetwork
networks:
  mynetwork:
    driver: overlay
```

В данном файле определены два сервиса - db и phpmyadmin, которые запускаются на узлах с меткой env=dev. Контейнеры масштабируются до трех копий для сервиса db и одной копии для сервиса phpmyadmin. Сеть mynetwork создается как overlay-сеть, чтобы обеспечить связь между контейнерами на разных узлах.

После запуска файла docker-compose.yml с помощью команды "docker stack deploy", Docker Swarm автоматически развернет контейнеры на узлах, удовлетворяющих условиям размещения. Если какой-то узел выходит из строя, Docker Swarm автоматически перезапускает контейнеры на других узлах.

Оба контейнера имеют метку dev. Чтобы добавить метки для других окружений, можно скопировать блоки сервисов и изменить значение переменной окружения на prod или lab.

Для создания сервисов в каждом окружении необходимо создать отдельные файлы docker-compose.dev.yml, docker-compose.prod.yml и docker-compose.lab.yml, в которых будут указаны соответствующие параметры окружения (например, порты и переменные окружения для БД). Затем необходимо запустить каждый файл командой "docker stack deploy --compose-file docker-compose.[env].yml myapp-[env]".

После выполнения всех шагов на каждой ноде кластера будет запущено по 2 контейнера - веб-сервер и БД. Результаты можно проверить с помощью команды "docker service ls", которая выведет список запущенных сервисов и количество их реплик на каждой ноде.

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

```
#docker-compose.lab.yml
version: "3.9"

services:
  web:
    image: nginx:latest
    ports:
      - "8080:80"
    environment:
      - ENVIRONMENT=lab
    deploy:
      replicas: 1
      placement:
```

```
    constraints:
      - node.labels.env == lab
  networks:
    - mynetwork
```

```
#docker-compose.lab.yml
version: "3.9"
```

```
services:
  db:
    image: mysql:latest
    environment:
      - ENVIRONMENT=lab
    deploy:
      replicas: 5
      placement:
        constraints:
          - node.labels.env == lab
    networks:
      - mynetwork
```

```
#docker-compose.dev.yml
version: "3.9"
```

```
services:
  web:
    image: nginx:latest
    ports:
      - "8080:80"
    environment:
      - ENVIRONMENT=dev
    deploy:
      replicas: 1
      placement:
        constraints:
          - node.labels.env == dev
    networks:
      - mynetwork
```

```
#docker-compose.dev.yml
version: "3.9"
```

```
services:
  db:
    image: mysql:latest
    environment:
      - ENVIRONMENT=dev
    deploy:
      replicas: 5
```

```
  placement:
    constraints:
      - node.labels.env == dev
networks:
  - mynetwork
```

```
docker stack deploy --compose-file docker-compose.lab.yml myapp-lab
docker stack deploy --compose-file docker-compose.dev.yml myapp-dev
```