

## Контейнеризация (семинары)

### 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).

Формат сдачи ДЗ: предоставить доказательства выполнения задания посредством ссылки на google-документ с правами на комментирование/редактирование. Результатом работы будет: текст объяснения, логи выполнения, история команд и скриншоты (важно придерживаться такой последовательности).

В названии работы должны быть указаны ФИ, номер группы и номер урока.

#### Критерии оценивания ПА "Контейнеризация":

- 1) Выслан подробный отчет по выполнению 5-го ДЗ (см.: результаты работы)
- 2) Выполнено / частично выполнено / сделана попытка выполнения задания 2\* в 5-м ДЗ.
- 3) Добавлена история выполнения команд в 5-м ДЗ. Идеально, если выслан весь процесс попыток выполнения, учитывая ошибочные.

В данном случае оценку НЕ снижать при некорректном выполнении задания со \*. Снижать только в случае отсутствия попытки его выполнения.

Оценивается "зачет"/"незачет"

"зачет" проставляется при выполнении 3 критериев.

"незачет" проставляется при выполнении 2 и менее критериев.

**Docker run --name some\_mysql\_my -e MYSQL\_ROOT\_PASSWORD=max -d mysql:8.0.31**

**docker-compose build** – собрать сервисы во что-то

**Docker-compose up** – оманда запустить проект (-d – запуск в фоновом режиме)

Аналог этой команды

**Docker run -d <имя контейнера>**

**Docker-compose start** – запускает останленные ранее сервисы или **Docker-compose up**

**Docker-compose stop** – останавливает контейнеры и удаляет сущности **Docker-compose down**

**Docker-compose logs -f <service-name>** – ысобрать логи нашего сервиса

**Docker-compose exec** - Выполнить команду в сервисе не заходя внутрь контейнера

**Docker-compose images** - Список образов которые доступны в конфигурационном файле

**Docker rm -f <первая буква id шника контейнера-1> <первая буква id шника контейнера-2>**

**docker container prune** – удалить незапущенные контейнеры

```
(base) max@max:~$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
ab59d62dda5c81b0187e7c109ccbea00721a40cc5c657757629f752f60a76fc2
38ff856de2df2bb90a12f9c18939d578e5df172baee3724515e908654edef587

Total reclaimed space: 0B
(base) max@max:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
(base) max@max:~$
```

Посмотрим, что запущено:

**docker ps -a**

project.yaml

version '3.9' – версия докер компоуза

services: – переменная имеет вложенные значения

db – название сервиса, кот-й будет запущен. Название можно придумать

image: mariadb:10.10.2 – образ, который будет использоваться (по умолчанию будет скачиваться с докерхаб)

image: mariadb:10.10.2 – вместо image можно build

build: ./db – путь к докерфайлу, который содержится по этому пути (если надо использовать наш образ)

restart : always – политика перезапуска контейнера

enviroment:

MYSQL\_ROOT\_PASSWORD: 12345

adminer:

image: adminer:4.0.1

restart : always

ports:

- 6000:0000

Семинар \_\_\_\_\_

(base) max@max:~\$

**docker run -d --name db -e MYSQL\_ROOT\_PASSWORD=max mysql**

```
ek0lenk0@us-01:~/compose$ docker run -d --name admin --link db:db phpmyadmin
```

|

:

**docker run -d --name admin --link db:db -p 80:80 phpmyadmin**

**docker ps -a** – отреть что запущено

**sudo docker container inspect 8e3afc8d80bc** - посмотреть свойства контейнера. Там будет айпишник

```
{
  "Networks": {
    "bridge": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "NetworkID": "1b0332f3d3ca736bacfe334a67cdebf95703c77ce57838f49b2afb163bc50759",
      "EndpointID": "254d16245dfb06b99ca0c72c0117637cfdc6eb798b8f44b652d3b2323191af",
      "Gateway": "172.17.0.1",
      "IPAddress": "172.17.0.3",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "MacAddress": "02:42:ac:11:00:03",
      "DriverOpts": null
    }
  }
}
```

**docker run -d --name db -e MYSQL\_ROOT\_PASSWORD=max mysql**

**docker run -d --name admin --link db:db -p 80:80 phpmyadmin**

```
GNU nano 6.2 start.sh *
docker run -d --name db -e MYSQL_ROOT_PASSWORD=root mysql
docker run -d --name admin --link db:db -p 80:80 phpmyadmin
```

Поменять айпишник ноде

1. Удалить ip

```
valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fe31:26d0/64 scope link
valid_lft forever preferred_lft forever
max1@node-2:~$ sudo ip a del 192.168.187.18 dev enp0s3
```

2. Дать новый ip

```
valid_lft forever preferred_lft forever
max1@node-2:~$ sudo ip a add 192.168.187.68 dev enp0s3
```

Поменять имя ноды:

1. `sudo hostname <new-name-node>`
2. `exit`
3. Заход по-новой – логин и пароль

```
Last login: Tue Apr 18 20:59:02 UTC 2023 on tty1
max1@node-0:~$
```

Подключимся по ssh – проверим, что видим ноды:

1. Смотрим на нодах их ip
2. Пишем в терминале `ssh max1@192.168.187.18`

3.

```
Last login: Tue Apr 18 21:05:30 2023
max1@node-0:~$
```

4.

```
max1@node-0:~$ ssh max1@192.168.187.18
max1@node-1:~$
System load: 0.080078125    Processes:           105
Usage of /:  53.5% of 8.02GB Users logged in:       1
Memory usage: 22%          IPv4 address for enp0s3: 192.168.187.68
Swap usage:  0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Wed Apr 19 19:02:44 2023
max1@node-1:~$
valid_lft forever preferred_lft forever
valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fe31:26d0/64 scope link
valid_lft forever preferred_lft forever
max1@node-0:~$ max1@node-1:~$ max1@node-2:~$
```

