Оркестрация контейнеров

Владимир Мардеев
Системный инженер

Skillbox

Цели модуля

- Docker Compose
- Docker Swarm
- Kubernetes — первое знакомство

Содержание модуля

1. Docker Compose
2. Compose File
3. Docker Swarm
4. Kubernetes. Первое знакомство
5. Практическая работа

Docker Compose

Цели

- Узнать ключевые особенности Docker Compose
- Ознакомиться с основными сценариями Docker Compose
- Разобрать простейший пример использования Docker Compose

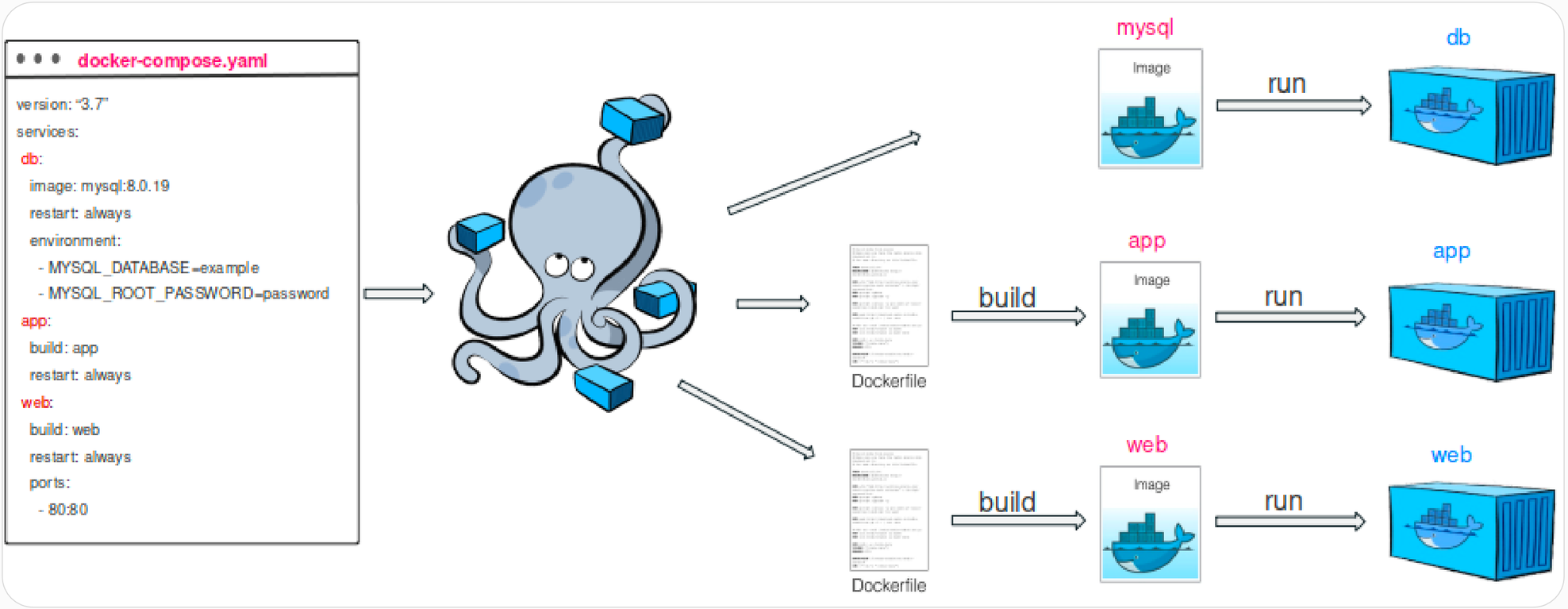
docker run vs docker-compose

```
docker network create back tier
docker build -t frontend_image:v1.0.8 ./front
docker build -t backend_image:v1.0.1 ./back
docker run \
  --name frontend \
  --publish 443:443 \
  --network back-tier \
  --volume static:/var/www/
  frontend_image:v1.0.8
docker run \
  --name backend \
  --network back-tier \
  --volume userfiles:/opt/storage
  backend_image:v1.0.1
```

```
compose.yaml
services:
  frontend:
    build:
      context: ./front
    image: frontend_image:v1.0.8
    ports: “443:443”
    volumes:
      - type: volume
        source: static
        target: /var/www
    networks:
      - back-tier
  backend:
    build:
      context: ./back
    image: backend_image:v1.0.1
    volumes:
      - type: volume
        source: userfiles
        target: /opt/storage
    networks:
      - back-tier
docker compose up
```

Скриншоты спикера курса

Типовое применение Docker Compose



Изображение: Типовое применение Docker Compose

Frontend developer docker compose

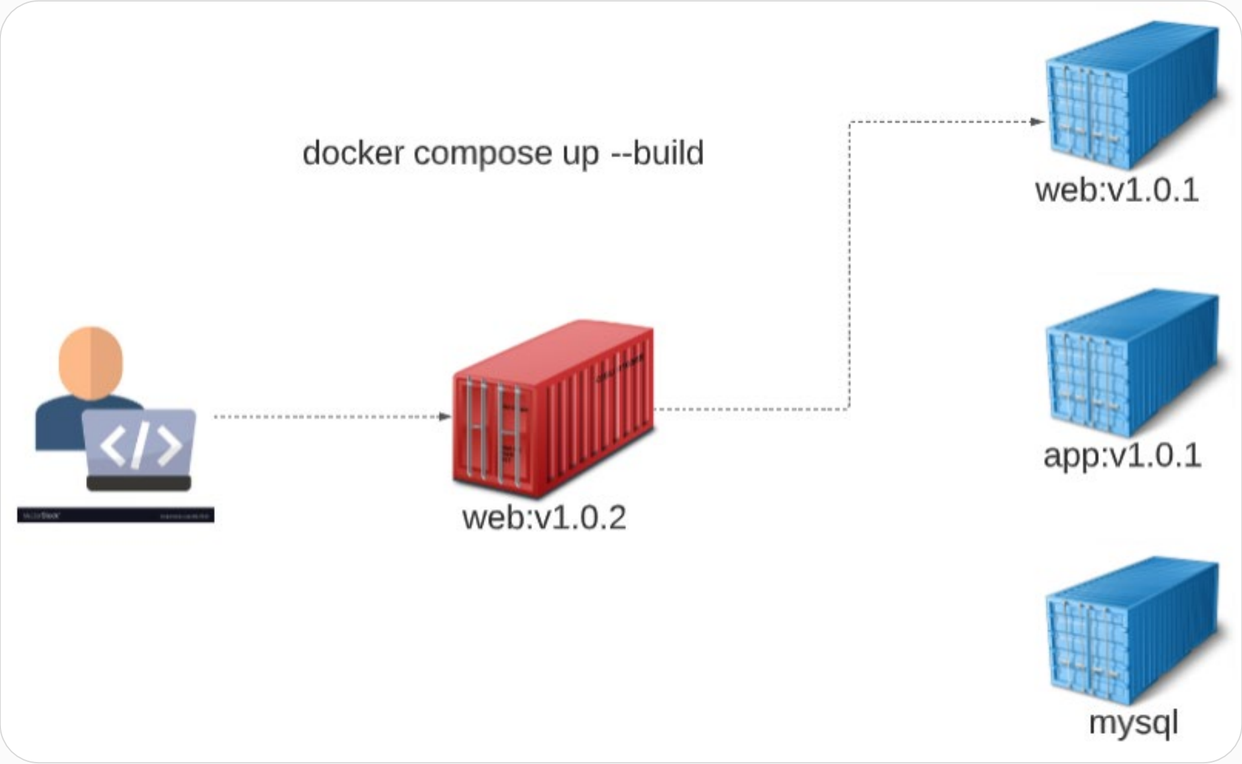


Иллюстрация спикера курса

Возможности и особенности Compose

Базовые возможности Compose:

- запуск, остановка и пересборка сервисов
- просмотр состояния запущенных сервисов
- просмотр потока логов событий запущенных сервисов
- выполнение команд в контейнерах

Ключевые особенности Compose:

- множество изолированных сред на одном сервере
- сохранение данных на созданных докер-вольюмах
- пересоздание только тех контейнеров, которые изменились
- при помощи переменных можно задавать поведение в различных средах

Сценарии использования docker-compose

1. В среде разработки
2. CI/CD
3. Single host deployments

Итоги

- Ключевые особенности:
  - изолированные среды
  - запуск, остановка и пересборка сервисов
  - сохранение данных на созданных томах
- Основные сценарии:
  - Dev
  - Ci/Cd
  - Single host deployment
- Простейший пример

```
services:
  ng1:
    image: nginx
    ports:
      - "8080:80"
```