

НАЦІОНАЛЬНА БЕЗПЕКА

Голев Д.В.

старший викладач;

Амбарян С.К., Лялюков В.О., Усачов Д.В.

студенти,

*Одеська національна академія зв'язку
імені О.С. Попова*

СТВОРЕННЯ НАДІЙНОГО СЕРЕДОВИЩА ВИКОНАННЯ ВЕБ-СЕРВІСІВ З ВИКОРИСТАННЯМ DOCKER-КОНТЕЙНЕРІВ

На сьогоднішній день існує проблема розгортання несумісних веб-додатків на одному сервері, неможливість одночасного використання декількох версій одного додатка або систем управління базами даних.

Більшість компаній використовують класичну віртуалізацію, яка створює зайве навантаження на систему та зменшує їх швидкодію.

Розробники Docker обрали альтернативний шлях вирішення проблеми.

Принцип роботи Docker простіше за все порівняти з транспортними контейнерами, адже колись транспортні компанії стикалися зі схожими проблемами:

1. Як перевозити різні (несумісні) типи товарів разом (наприклад, продукти харчування з хімікатами або скло з цеглою)?

2. Як обробляти пакети різних розмірів одним і тим же транспортним засобом?

З введенням контейнерів стало можливим перевозити разом цеглу та скло, хімікати та їжу, а також багато іншого. Вантаж різного розміру може бути розподілений по стандартизованим контейнерам, які завантажуються або вивантажуються одним і тим же транспортним засобом.

Але повернемося до контейнерів. Коли ви розробляєте додаток, вам потрібно надати код разом з усіма його складовими, такими як бібліотеки, сервер, бази даних і т.д. Ви можете опинитися в ситуації, коли додаток працює на вашому комп'ютері, але відмовляється працювати на пристрої іншого користувача. Ця проблема вирішується через створення незалежності програмного забезпечення від системи.

Спочатку віртуалізація була покликана позбавити від подібних проблем, але в ній є суттєві недоліки:

- повільне завантаження;
- можлива плата за надання додаткового простору;
- не всі віртуальні машини (VM) підтримують сумісне використання;
- підтримуючі VM часто вимагають складного налаштування;

– образ може бути занадто великим, адже «додаткова ОС» потребує гігабайт простору в проект поверх операційної системи, а в більшості випадків на сервер ставиться кілька VM, які займають ще більше місця.

Docker розділяє ядро ОС на контейнери (Docker container), що працюють як окремі процеси. Це не єдина подібна платформа, але, безперечно, одна з найпопулярніших і затребуваних.

До його переваг відносяться:

– прискорений процес розробки. Немає необхідності встановлювати допоміжні інструменти на зразок PostgreSQL, Redis, Elasticsearch: їх можна запускати в контейнерах;

– зручна інкапсуляція додатків;

– зрозумілий моніторинг;

– просте масштабування.

Docker працює не тільки на його рідній ОС, Linux, але також підтримується Windows і Mac OS. Єдина відмінність від взаємодії з Linux в тому, що на Mac OS і Windows платформа інкапсулюється в невелику віртуальну машину. На даний момент Docker для Mac OS і Windows досяг значного рівня зручності у використанні.

Наприклад, якщо є необхідність використовувати веб-сервер Nginx, потрібно виконати в терміналі наступну команду:

```
docker run nginx
```

Після цього Docker спробує знайти образ, тобто базовий елемент контейнеру, з Nginx локально і якщо не знайде – автоматично завантажить його з сервісу Docker Hub та запустить.

Docker Hub – публічний репозиторій з інтерфейсом, що надається Docker Inc. Він зберігає безліч образів. Ресурс є джерелом «офіційних» образів, зроблених командою Docker або створених у співпраці з розробником програмного забезпечення. Для офіційних образів перераховані їх потенційні уразливості. Ця інформація відкрита для будь-якого зареєстрованого користувача. Доступні як безкоштовні, так і платні акаунти.

Також в Docker існує зручний інструмент для автоматизації створення специфічних контейнерів – Dockerfile.

Dockerfile містить набір інструкцій з аргументами. Кожна інструкція пишеться великими літерами (наприклад FROM). Інструкції обробляються зверху вниз. Також є можливість створення коментарів за допомогою символу «#». Наприклад, для створення Docker-контейнеру з веб-сервером Nginx знадобиться Dockerfile наступного виду:

```
# version 0.0.1
```

```
# Обираємо базовий образ для контейнеру. Зупинимося на Ubuntu 18.04.
```

```
FROM ubuntu:18.04
```

```
# Далі ми додаємо інформацію про себе.
```

```
MAINTAINER First_name Last_name mail_adress@mail.xx
```

```
# Оновлюємо репозиторії та встановлюємо Nginx.
```

```
# Усі команди виконуються від імені root.
```

```
RUN apt update && apt install -y nginx
```

```
# Прокидаємо назовні 80-й порт  
EXPOSE 80
```

```
# Налаштовуємо автозапуск Nginx після створення контейнеру  
ENTRYPOINT [«/usr/bin/nginx»]
```

Надалі для того, щоб створити контейнер з веб-сервером Nginx та запустити його, нам потрібно буде виконати команду:

```
docker build Dockerfile
```

Якщо нам знадобиться зупинити контейнер, то потрібно буде виконати:

```
docker stop container_name
```

де `container_name` – назва контейнера який потрібно зупинити.

Висновок. На основі проведених досліджень виділимо основні переваги Docker:

- ізольований запуск додатків в контейнерах
- спрощення розробки, тестування і розгортання додатків
- відсутність необхідності конфігурувати середовище для запуску – воно поставляється разом з додатком – в контейнері
- спрощує масштабованість додатків у контейнерах і управління їх роботою за допомогою систем автоматичного розміщення, координації і управління.

Використання Docker на виробничому сервері дозволяє досягти значного рівня відмовостійкості, стабільності та безпеки веб-сервісу.

Список використаних джерел:

1. Docker documentation [Електронний ресурс] // Docker Inc. – 2018. – Режим доступу до ресурсу: <https://docs.docker.com/>.
2. freetonik. Полное практическое руководство по Docker: с нуля до кластера на AWS [Електронний ресурс] / freetonik // «ТМ». – 2016. – Режим доступу до ресурсу: <https://habr.com/post/310460/>.
3. Official Docker Hub nginx repository [Електронний ресурс] // Docker Inc. – 2018. – Режим доступу до ресурсу: https://hub.docker.com/_/nginx/.
4. AG10. Docker. Начало [Електронний ресурс] / AG10 // «ТМ». – 2018. – Режим доступу до ресурсу: <https://habr.com/post/353238/>.
5. admin. Docker [Електронний ресурс] / admin. – 2017. – Режим доступу до ресурсу: <http://webit.in.ua/article/Docker/>.