

РОЗГОРТАННЯ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ ЗА ДОПОМОГОЮ ДОКЕРА

Васільєва Ю.С., Курило О.В.

Науковий керівник – ст.викл. каф. ІМІ Малінін О.П.
Харківський національний університет радіоелектроніки
(61166, Харків, преси. Науки, 14, каф. Інформаційно – мережної
інженерії, тел. (057) 702-14-29)

A recent Nginx study found that 36% of businesses currently use microservices, while another 26% conduct research on how to implement them. This paper discusses the benefits of microservices as well as how they interact with docker. The advantages of the microservices that we are talking about in the work helps to flexibly configure applications and takes care of data persistence. Containerization is very widely used in modern programming, and this is what we will consider in our work. We will also consider docker, which is the most popular and used platform for the development, operation and delivery of applications.

Мікросервіси - все більш популярна архітектура для створення масштабних додатків. Вони також відомі як мікросервісні архітектури - це архітектурний стиль, який структурує додаток як набір сервісів, які:

- Ремонтпридатні та легко перевіряються;
- Незалежно розгортаємі;
- Належить невеликій команді.

Мікросервісна архітектура забезпечує швидку, часту і надійну доставку великих і складних додатків. Це також дозволяє організації розвивати свій технологічний стек. Існує безліч плюсів, пропонованих архітектурою Microservices. Ось кілька з них:

• Мікросервіси можуть легко адаптуватися до інших систем або технологій.

- Відмова одного процесу не впливає на всю систему.
- Надає підтримку великих підприємств, а також невеликим командам.
- Може бути розгорнутий незалежно і у відносно короткі терміни.

Плюси:

- Свобода використання різних технологій.
- Кожен мікросервіс орієнтований на одну можливість.
- Підтримує окремі розгортаємі блоки.
- Дає змогу частим випускам програмного забезпечення.
- Забезпечує безпеку кожного сервісу.
- Кілька сервісів паралельно розробляються і розгортаються.

Мінуси:

- Збільшує проблеми усунення неполадок.

- Збільшення зусиль для налаштування та інших операцій.
- Складно підтримувати безпеку транзакцій.
- Важко відстежувати дані через різні кордони.

Контейнеризація - це тип стратегії віртуалізації, що з'явився в якості альтернативи традиційної віртуалізації на основі гіпервізора. При контейнеризації операційна система може використовуватися спільно контейнерами, а не клонується для кожної віртуальної машини. Наприклад, Docker надає платформу віртуалізації контейнерів, яка служить гарною альтернативою угодам на основі гіпервізора.

Основна особливість контейнерів - це те, що вони є незмінними (immutable). Якщо ви створили контейнер один раз, він зберігає повний зліпок файлової системи. У будь-який момент, незалежно від середовища запуску, це буде повністю ідентична копія сервісу, включаючи ОС та всі необхідні вам dll-бібліотеки. Такий контейнер буде працювати скрізь однаково: на машині розробника, в тестовому оточенні або в production.

Докер - це відкрита платформа для розробки, доставки і експлуатації додатків. Docker розроблений для більш швидкого викладання ваших додатків. За допомогою docker ви можете відокремити ваш додаток від вашої інфраструктури і поводитися з інфраструктурою як керованим додатком. Docker допомагає викладати ваш код швидше, швидше тестувати, швидше викладати додатки і зменшити час між написанням коду і запуску коду. Docker робить це за допомогою легковагової платформи контейнерної віртуалізації, використовуючи процеси і утиліти, які допомагають керувати і викладати ваші програми.

У своєму ядрі docker дозволяє запускати практично будь-який додаток, безпечно ізольоване в контейнері. Безпечна ізоляція дозволяє вам запускати на одному хості багато контейнерів одночасно.

Docker - відмінний інструмент для управління і розгортання мікросервісів. Кожен мікросервіс може бути далі розбитий на процеси, що виконуються в окремих контейнерах Docker, що можна вказати в файлах конфігурації Dockerfiles і Docker Compose. У поєднанні з інструментом забезпечення, таким як Kubernetes, кожний мікросервіс може бути легко розгорнутий, масштабований і спільно використан командою розробників. Вказівка середовища таким чином також спрощує об'єднання мікросервісів для формування більшого додатку.

Список використаних джерел:

1. <https://www.linode.com/docs/applications/containers/deploying-microservices-with-docker/>
2. <https://habr.com/>
3. <https://vps.ua/blog/docker-and-linux-containers/>
4. <https://microservices.io/>