

РОЗРОБКА МІКРОСЕРВІСНИХ ДОДАТКІВ

Вінницький національний технічний університет

Анотація

Розглянуто інструменти для створення мікросервісних додатків для платформи Java з використанням відкритого фреймворку Spring. Проаналізовано підходи, що використовуються для створення і роботи мікросервісних додатків.

Ключові слова: мікросервіс, Java, Spring Framework.

Abstract

The tools for creating microservice applications for the Java platform using the free Spring Framework are considered. The approaches used for the creation and operation of microservice applications are analyzed.

Keywords: microservice, Java, Spring Framework.

Вступ

Поширеним підходом до проектування інформаційних систем є монолітна архітектура – вся система розробляється у вигляді однієї програми, що відповідає за виконання всіх функцій, покладених на систему. Такий підхід має проблеми з масштабуванням: у разі збільшення навантаження потрібно або збільшувати обчислювальні потужності серверного обладнання, або розгортати нові екземпляри додатку і вирішувати завдання синхронізації даних. Його наступним етапом можна вважати сервіс-орієнтовану архітектуру – підхід, що передбачає розробку модульних систем, що складаються зі слабо пов'язаних компонентів, взаємодіючих через заздалегідь налагоджені протоколи.[1]

Аналіз підходів для створення мікросервісних додатків

Продовженням розвитку ідеї сервіс-орієнтованої архітектури в проектуванні розподілених систем є мікросервісний підхід. В його основі лежить принцип розділення системи на незалежні компоненти, причому досить пріоритетна задача покладається на полегшення цих компонентів і їх взаємодію через налагоджені канали зв'язку, як правило, REST (representational state transfer).

Кожен компонент мікросервісного додатку повинен виконувати якомога простішу і незалежну функцію, його розгортання не має залежати від стану інших компонентів, можлива реалізація з використанням різних мов, фреймворків, засобів віртуалізації. Компоненти можуть бути легко розгорнуті в декількох екземплярів, оскільки кожен компонент досить легкий. Поведінка системи не повинна змінюватися від зміни окремих компонентів, можливе гнучке масштабування тих компонентів, на які лягає велике навантаження.

Широке поширення при розробці високонавантажених монолітних систем займає мова програмування Java, завдяки зрозумілому синтаксису та високій швидкості розробки. З ростом популярності мікросервісного підходу був розроблений ряд інструментів для розробки мікросервісних систем на мові програмування Java.

Огляд інструментів для створення мікросервісних додатків

Фреймворк Spring є одним з найбільш потужних і популярних засобів розробки веб-додатків, при цьому він має багатий набір засобів для реалізації мікросервісного підходу при проектуванні і реалізації складних програмних систем. Спочатку Spring Framework був контейнером впровадження залежностей (dependency injection) для Java платформи. Сьогодні фреймворк пропонує ряд засобів, серйозно спрощуючи розробку мікросервісних додатків.

Spring Boot – бібліотека для швидкого розгортання і конфігурації веб-додатків. Автоматизує менеджмент залежностей для інших компонентів Spring, створює і автоматично налаштовує мережеве оточення, розгортає вбудований веб-сервер, що полегшує розгортання додатків в контейнерах (наприклад, в Docker). Spring Boot пропонує підключення додаткових компонентів через Starter-паки - набори бібліотек, об'єднаних спільним завданням. Наприклад: Spring Boot Starter Security - підключення бібліотек, пов'язаних з безпекою, Spring Boot Starter Data JPA – підключення ORM-системи і шару роботи з базами даних в термінах Java PersistenceAPI тощо.

З корисних можливостей фреймворку виділяють інтеграцію з великими хмарними провайдерами. Spring Framework з стандартно надає набір засобів для роботи з інфраструктурою багатьох великих хмарних провайдерів: Amazon Web Services, Microsoft Azure, Google Cloud Platform. Також корисна функція підтримки брокерів повідомлень. Вони використовуються в більш складних системах з високим навантаженням і складною взаємодією між компонентами. Брокери повідомлень являють собою компоненти, які додають певну логіку в обмін повідомленнями між сервісами. Spring підтримує інтеграцію з наступними брокерами: RabbitMQ, Apache Kafka (включаючи Kafka Streams), Solace PubSub.[2]

Висновки

Проаналізовано монолітний та мікросервесний підхід для створення і роботи додатків написаних на мові програмування Java використовуючи потужного фреймворку. Spring Framework дає досить багатий набір засобів для розробки мікросервісних рішень, які зручно інтегрувати з хмарною інфраструктурою, розгортати в контейнерах, застосовувати горизонтальне і вертикальне масштабування, контролювати стан і управляти компонентами мікросервісного додатку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Порівняння монолітної і мікросервесної архітектури [Електронний ресурс]:[Веб-сайт] – Електронні дані. — Режим доступу: http://www.sciencehorizon.ru/wp-content/uploads/2017/08/Nauchnyie-gorizontyi_5_33_2020.pdf#page=300
2. Spring Framework Documentation [Електронний ресурс]:[Веб-сайт] – Електронні дані. — Режим доступу: <https://docs.spring.io/spring-framework/docs/current/reference/html/index.html>

Рудич Єлизавета Олександрівна — студентка групи ІАКІТ-17б, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, м. Вінниця, e-mail: liza79682@gmail.com

Довгалець Сергій Михайлович — професор кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м. Вінниця

Rudych Elizabeth O. — student of group ІАКІТ-17b, faculty of computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia.

Dovgalets Sergeiy M. — Professor of Automation and Intelligent Information Technology department, Vinnytsia National Technical University, Vinnytsia.