

МІКРОСЕРВІСНА АРХІТЕКТУРА В РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

MICROSERVICE ARCHITECTURE IN SOFTWARE DEVELOPMENT

Мікросервісна архітектура є відносно новим рішенням для побудови програмних продуктів. Вона складається з набору невеликих автономних сервісів, кожен з яких може реалізовувати окрему бізнес модель. Мікросервіси невеликі, незалежні та слабо пов'язані. Кожна служба є окремою кодовою базою, якою може керувати невелика команда розробників. Сервіси відповідають за збереження власних даних або зовнішнього стану. Це відрізняється від традиційної моделі, де окремий рівень даних обробляє збереження даних.

Сервіси взаємодіють одне з одним за допомогою чітко визначених API. Деталі внутрішньої реалізації кожної служби приховані від інших сервісів. Також вони підтримують поліглотне програмування і не потребують спільного використання одного стеку технологій, бібліотек або фреймворків.

Таке архітектурне рішення має очевидні переваги. Так, оскільки мікросервіси розгортаються незалежно, керувати виправленнями помилок і випусками функцій легше. Розробники можуть вибрати технологію, яка найкраще підходить для їхніх послуг, використовуючи відповідну комбінацію стеків технологій. Якщо сервіс стає недоступним, це не призведе до переривання роботи всієї програми. Сервіси можна масштабувати незалежно, дозволяючи вам масштабувати підсистеми, які вимагають більше ресурсів, без масштабування всієї програми. Також вони ізолюють дані.

Також мікросервісна архітектура має ряд недоліків, які є важливими факторами, для вибору її як основу для проектів. Кожний мікросервіс простий, але вся система в цілому складніша. Написання невеликого сервісу, який покладається на інші залежні служби, вимагає іншого підходу, ніж написання традиційної монолітної або багаторівневої програми. Також складно тестувати залежності сервіси, особливо коли проектування у вас можуть виникнути проблеми із зворотною чи прямою сумі програма швидко розвивається. Може відбутися перевантаження мережі та затримка, через використання великої кількості сервісів. Кожний мікросервіс відповідальний за власне збереження даних. У результаті узгодженість даних може бути проблемою, як і узгодженість версій проекту.

Література

1. Мікросервісна архітектура. URL: learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices.