УДК 004.8+65.05+681.5

Катков Ю. І., канд. техн. наук, доцент,
Зінченко О. В., канд. техн. наук., доцент
Державний університет телекомунікацій

ЗАСТОСУВАННЯ СИСТЕМИ КЕРУВАННЯ КОНТЕЙНЕРАМИ ОПЕРАЦІЙНИМИ СИСТЕМАМИ ДЛЯ РОЗГОРТАННЯ КРИТИЧНИХ МІКРОСЕРВІСНИХ ДОДАТКІВ

У статті розглядається проблема пошуку швидкого переходу в критичних умовах від монолітної до мікросервісної архітектури побудови операційних систем (ОС). Проблема вирішується на основі застосування методології активної взаємодії програмістів-розробників з фахівцями інформаційно-технологічного обслуговування, що надає можливість взаємної інтеграції їх робочих процесів один в одного. Для цього потрібні більш гнучкі технології, які засновані на ідеї тісного взаємозв'язку розробки та експлуатації програмного забезпечення (ПЗ).

Для вирішення проблеми запропоновано застосування технології контейнеризації Docker та оркестровці контейнерів Kubernetes (K8s). Для обґрунтування такого підходу проведено аналіз переваг та недоліків монолітної та мікросервісної архітектури під час застосування даних технологій. Побудова додатків на базі Measurement System Analysis (MSA) передбачає низку обмежень, тому що MSA – це варіант Service-oriented architecture (SOA) ПЗ, орієнтований на взаємодію невеликих, слабо пов'язаних і легко змінюваних модулів – мікросервісів. Недоліком монолітної архітектури є те, що будь які зміни потребують перебудови та повторного розгортання всього додатку для адаптації до нових умов функціонування бізнесу. Якщо такі умови змінюються досить час-то, то з часом стає складніше зберігати хорошу модульну структуру, тому що зміни логіки одного модуля мають тенденцію впливати на код інших модулів. Крім того, монолітна архітектура не дозволяє масштабувати програми, тому що різні модулі мають конфліктні вимоги до ресурсів. Наприклад, один модуль може реалізовувати логіку обробки зображень з інтенсивним використанням процесору, а інший модуль може бути вимогливим до використання оперативної пам'яті. Однак, оскільки ці модулі розгортаються разом, то доведеться йти на компроміс з вибором апаратного забезпечення. Крім того, якщо потрібно масштабувати тільки один модуль, то разом з тим доводиться примусово масштабувати додаток для другого модуля, навіть якщо це не потрібно.

Для усунення цих незручностей мікросервісна архітектура дозволяє кожен мікросервіс розгортати окремо в наслідок того, що в архітектурі

мікросервіса кілька слабко пов'язаних служб працюють разом. Кожен сервіс орієнтований на одну мету і має високий ступінь узгодженості поведінки й даних. Тому, якщо треба змінювати щось в одному з них, то можна розгорнути ці зміни не чіпаючи інших мікросервісів, які можуть продовжувати працювати. Це стає можливо внаслідок реалізації принципів моделювання мікросервісів у мікросервісній архітектурі.

Таким чином, мікросервісну архітектуру раціонально застосовувати коли інші види архітектури стають вузьким місцем: відносно продуктивності процесів; уповільнення розробки нових продукто-послуг; монолітний додаток ускладнює масштабування системи для конкретних завдань або ізоляцію проблем ресурсів для різних типів завдань; заважає випробувати нові технології (однією з основних переваг мікросервісної архітектури є те, що кожен сервіс може бути побудований з використанням різних технологічних стеків та інтегрований з різними технологіями); моноліт – не дозволяє вибрати кращий інструмент для роботи і, що більш важливо, зробити це швидким і безпечним способом.

Для вирішення питання переходу від монолітної архітектури до мікросервісної спочатку використовувались апаратні засоби віртуалізації такі як KVM, XEN, VMware ESXi, але з часом з'явилися технології віртуалізації рівня ОС, такі як Docker та Kubernetes.

Технологія Docker дозволяє упакувати зліпок стану системи, виконати його розробку і запустити. З її появою зацікавленість контейнерами вибухово зросла, Розгортати додатки виявилося настільки зручно, що технологію почали використовувати всюди.

Технологія Kubernetes відноситься до віртуалізації рівня ОС і є провідною платформою надійного планування робочих навантажень для відмовостійких додатків. Вона призначена для управління контейнерними додатками Docker і пов'язаними з ними компонентами мережі та сховища. Основна увага в ній приділяється робочим навантаженням додатків, а не базових компонентів інфраструктури.

Список використаних джерел

1. Computer Architecture: A Quantitative Approach 5th Edition by John L. Hennessy (Author), David A. Patterson (Author), Morgan Kaufmann; 5 editions (September 30, 2011), 856 pages, ISBN-10: 012383872X, ISBN-13: 978-8178672663.

2. Microservices vs. Service-Oriented Architecture. by Mark Richards. Publisher: O'Reilly Media, Inc.Release Date: April 2016. ISBN: 9781491975657.

3. Kubernetes: Up and Running, 2nd Edition \ Dive into the Future of Infrastructure. By Brendan Burns, Kelsey Hightower, Joe Beda – Publisher: O'Reilly Media, Release Date: October 2019. Pages: 278.