

АНАЛІЗ СТРАТЕГІЙ РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розгортання програмного забезпечення є важливим етапом у процесі розробки, оскільки саме на цьому етапі програмний продукт переходить від стадії розробки до стадії експлуатації. Від правильно обраної стратегії розгортання залежить не тільки успішність запуску проекту, але й його подальша ефективність та безпека.

Метою дослідження є аналіз та порівняння основних стратегій розгортання програмного забезпечення для визначення їхньої ефективності. Дослідження спрямоване на оцінку переваг і недоліків таких підходів, як Rolling Deployment, Canary Deployment та Blue-Green Deployment.

Rolling Deployment (Поступове Розгортання) - метод оновлення програмних систем, який дозволяє безперервно впроваджувати нові версії програмного забезпечення в продакшн-середовище за допомогою автоматизованих рішень, де окремі екземпляри системи оновлюються по черзі [1].

Canary Deployment (Канаркове Розгортання) - стратегія безперервного розгортання програмного забезпечення, яка дозволяє мінімізувати ризики, пов'язані з введенням нових версій системи в експлуатацію за допомогою спостереження її впливу на обмежену групу користувачів, перш ніж розгорнути її для загального використання [2].

Blue-Green Deployment (укр. Синьо-Зелене Розгортання) - метод встановлення змін для веб-сервера, додатка або сервера бази даних шляхом чергування робочих і проміжних серверів [3]. Даний підхід передбачає наявність двох ідентичних середовищ для розгортання: Blue (поточне середовище) та Green (нове середовище), що дозволяє в лічені секунди перемикається між даними версіями або балансувати трафік між ними в залежності від налаштувань даного методу.

У результаті аналізу перелічених стратегій розгортання можна зробити наступні висновки.

Дані стратегії розгортання мають спільні недоліки:

- потребують ретельної перевірки сумісності версій перед початком розгортання, оскільки вони будуть існувати паралельно протягом певного часу;
- потребують автоматизованої системи моніторингу роботи для прийняття рішення про подальше розгортання або відкат оновлень.

Незважаючи на перелічене, також можна виділити основні переваги та недоліки кожної з них:

Rolling Deployment. Перевага - більше підходить для систем що використовують мікросервісну архітектуру. Недолік - навантаження на систему може коливатись, оскільки інстанси що знаходяться в оновленні не можуть обробляти запити від клієнтів.

Canary Deployment. Перевага - дозволяє отримувати цінний зворотний зв'язок до того, як зміни будуть застосовані для всіх користувачів. Недолік - реалізація даного розгортання вимагає складного керування трафіком.

Blue-Green Deployment. Перевага - забезпечує швидке та безпечне розгортання з можливістю миттєвого відкату на попередню версію. Недолік - має високі витрати на керування та обслуговування двох продакшн-середовищ.

Отже, вибір оптимальної стратегії залежить від конкретних вимог проекту, наявних ресурсів та необхідних рівнів безпеки та стабільності. Кожен метод може бути ідеальним у своїх умовах, і вибір стратегії розгортання має враховувати такі фактори, як масштаб системи, вимоги до безперервної доступності, бюджет на інфраструктуру та необхідність швидкого реагування на збої.

Список використаних джерел:

1. Octopus Deploy. Rolling deployments: Pros, cons, and 4 critical best practices. 2025. URL: <https://octopus.com/devops/software-deployments/rolling-deployment/> (дата звернення: 23.01.2025).
2. Octopus Deploy. Canary deployments: Pros, cons, and 5 critical best practices. 2025. URL: <https://octopus.com/devops/software-deployments/canary-deployment/> (дата звернення: 23.01.2025).
3. Octopus Deploy. Blue/green deployments: how they work, pros and cons, and 8 critical best practices. 2025. URL: <https://octopus.com/devops/software-deployments/blue-green-deployment/> (дата звернення: 23.01.2025).
4. CLOUD NATIVE COMPUTING FOUNDATION. Load balancing for blue-green, rolling, and canary deployment. 2025. URL: <https://www.cncf.io/blog/2022/05/09/load-balancing-for-blue-green-rolling-and-canary-deployment/> (дата звернення: 23.01.2025).