

**ЗАСТОСУВАННЯ DEVOPS-ПІДХОДІВ У КЕРУВАННІ ВІРТУАЛЬНИМ
СЕРЕДОВИЩЕМ SINGLE-TENANT: МОДЕЛЬ CI/CD ТА АВТОМАТИЗОВАНЕ
ТЕСТУВАННЯ**

**APPLICATION OF DEVOPS APPROACHES IN MANAGING A SINGLE-TENANT
VIRTUAL ENVIRONMENT: CI/CD MODEL AND AUTOMATED TESTING**

У сучасних інформаційних системах архітектура Single-Tenant застосовується у сферах, де критичними є ізоляція даних, індивідуальне конфігурування середовищ і дотримання вимог безпеки. На відміну від Multi-Tenant-моделі, вона передбачає окрему інстанцію для кожного клієнта, що підвищує контрольованість, але ускладнює оновлення, тестування й масштабування. У таких умовах DevOps-підходи, зокрема CI/CD, GitOps, Infrastructure as Code (IaC) та автоматизоване тестування, стають ключовими інструментами керування життєвим циклом систем.

Основні труднощі впровадження DevOps у Single-Tenant-середовищах пов'язані з тим, що традиційні CI/CD-пайплайни орієнтовані на уніфіковану доставку, тоді як ізольовані інстанси потребують врахування різних конфігурацій і політик доступу. Аналіз наукових праць підтверджує, що варіативність середовищ, підвищені вимоги до безпеки та обмеження доступу ускладнюють стандартизацію процесів [1; 3]. Для вирішення цих викликів запропоновано дворівневу CI/CD-модель. Глобальний рівень охоплює збирання коду, тестування, створення артефактів і перевірку безпеки. Локальний рівень забезпечує індивідуальне розгортання, валідацію специфічних конфігурацій і построзгортальне тестування кожного інстансу.

Практики IaC та GitOps суттєво підсилюють відтворюваність і керованість інфраструктури. Використання репозиторіїв для зберігання конфігурацій забезпечує аудит, контроль версій і швидке виправлення конфліктів. Автоматизоване тестування включає модульні, інтеграційні та IaC-тести, регресійні перевірки, навантажувальні тести й безперервний моніторинг, який виконує функцію «тестування під час роботи» [2].

Запропонована DevOps-модель скорочує тривалість розгортань, знижує ризики інцидентів, підвищує передбачуваність змін та зміцнює безпеку інстансів. Її цінність полягає в поєднанні централізованого керування з автономністю клієнтських середовищ. Перспективні напрями включають моделювання ризиків масових оновлень і створення інтелектуальних механізмів пріоритетизації релізів на основі ML-методів.

Література

1. Kim G., Humble J., Debois P., Willis J. The DevOps Handbook. Portland : IT Revolution Press, 2016. 480 p.
2. Rahman M., Williams L. Characterizing DevOps Practices in Industry. ACM Transactions on Software Engineering and Methodology, 2020, vol. 29, no. 6, pp. 1–57.
3. Wigmore S., Erl T. DevOps in Practice. Boston : Prentice Hall, 2020. 312 p.