

АНАЛІЗ ПЕРЕДУМОВ РОЗРОБКИ ПРОГРАМНОГО МОДУЛЯ СИСТЕМИ БЕЗПЕРЕРВНОЇ ПОСТАВКИ ТА РОЗГОРТАННЯ КОДУ ДЛЯ КОНТЕЙНЕРИЗОВАНИХ ДОДАТКІВ В AWS

Вінницький національний технічний університет

Анотація

В контексті зростаючої необхідності у швидкій та надійній поставці програмного забезпечення, актуальність питання безперервної поставки коду стає очевидною. Традиційні методи розробки програмного забезпечення зазвичай витрачають значний час на інтеграцію, тестування та випуск нових функцій або оновлень. Контейнеризовані додатки забезпечують ізольоване середовище, що спрощує розгортання; консистентність між середовищами розробки та виробництва, і дозволяють швидко масштабувати та оновлювати додатки. Таким чином контейнеризація додатків значно полегшує процес розробки і розгортання та забезпечує більш ефективне використання ресурсів. Програмний модуль безперервної поставки та розгортання коду для контейнеризованих додатків пропонує ефективне вирішення проблеми швидкої інтеграції тестування та випуску нових функцій, надаючи засоби для автоматизації та оптимізації процесу розробки, тестування та релізу програмного забезпечення, побудованих на мікросервісній контейнерній архітектурі.

Ключові слова: автоматизація, оптимізація, безперервна поставка, контейнери.

Abstract

In the context of the growing need for fast and reliable software delivery, the relevance of the issue of continuous code delivery becomes obvious. Traditional software development methods typically spend a significant amount of time integrating, testing, and releasing new features or updates. Containerized applications provide an isolated environment that simplifies deployment; consistency between development and production environments, and allow applications to be scaled and updated quickly. Thus, containerization of applications significantly facilitates the process of development and deployment and ensures more efficient use of resources. The continuous delivery and code deployment software module for containerized applications offers an efficient solution to the problem of rapid integration testing and release of new features, providing means to automate and optimize the process of development, testing and release of software built on a microservice container architecture.

Keywords: automation, optimization, continuous delivery, containers.

Вступ

Нині швидкозростаючий світ програмного забезпечення вимагає не лише швидкого виходу на ринок, але й постійного вдосконалення та виправлення помилок. Процес розгортання та постійної доставки коду стає вирішальним для забезпечення конкурентоспроможності компаній. У цьому контексті розробка програмного модуля системи безперервної поставки коду є ключовою для ефективного управління цим процесом, особливо в контексті контейнеризованих додатків [1].

З кожним днем темпи розвитку та зміни у сфері програмного забезпечення швидко зростають. Компанії постійно шукають шляхи оптимізації розробки та постачання продуктів на ринок для задоволення потреб клієнтів та реагування на зміни в умовах конкуренції. У такому контексті система безперервної поставки коду стає незамінним інструментом для забезпечення високої якості та ефективності процесу розробки.

Важливою частиною забезпечення конкурентоспроможності та ефективності розробки компаній є використання технологій та сервісів хмарних провайдерів, наприклад Amazon Web Services (AWS).

Традиційні методи розробки програмного забезпечення часто ґрунтуються на циклах розробки, тестуванні та релізу, які можуть займати значний час і спричиняти затримки у поставці нового функціоналу або виправлення помилок. Цей підхід також може призводити до нестабільності та непередбачуваності процесу розробки, що ускладнює управління проектом та збільшує ризики для бізнесу.

Автоматизована система безперервної поставки коду надає можливість розробникам швидко та безперервно інтегрувати, тестувати та реалізувати зміни у вихідний код програмного забезпечення [2, 3].

Результати дослідження

Основним призначенням програмного модуля системи безперервної поставки та розгортання коду для контейнеризованих додатків в AWS є автоматизація збірки, тестування та розгортання коду у хмарне середовище AWS.

На сьогодні існує значна кількість систем безперервної поставки коду, але вони не завжди забезпечують достатню інтеграцію з хмарними провайдерами, простоту налаштування та широкий функціонал налаштувань. У табл. 1 наведено порівняльні характеристики найбільш популярних систем безперервної поставки коду: «GitLab CI/CD», «Travis CI», «CircleCI» [4].

Таблиця 1 – Порівняльна характеристика найбільш популярних систем безперервної поставки коду

Параметр	GitLab CI/CD	Travis CI	CircleCI
Інтеграція з системами керування версіями	Так	Так	Так
Швидкість налаштування	Середня	Висока	Висока
Складність синтаксису	Висока	Середня	Середня
Обмежена функціональність	Ні	Ні	Так
Вартість	Середня	Висока	Середня
Обмежені можливості для складних сценаріїв	Ні	Так	Ні
Інтеграція з хмарними середовищами «З коробки»	Ні	Ні	Ні

З табл. 1 видно, що більшість описаних систем мають такі переваги [5]:

- висока функціональність;
- швидкість налаштування;
- інтеграція з системами керування версіями.

До недоліків описаних систем можна віднести складність синтаксису, вартість використання та відсутність інтеграції з хмарними середовищами «з коробки» [6].

Враховуючи вище наведену інформацію, можна зробити висновок, що програмний модуль системи безперервної поставки та розгортання коду буде добре інтегрований з системами керування версіями, хмарними технологіями (cloud-native) і матиме широкий функціонал для виконання автоматизованої збірки, тестування та доставки коду контейнеризованого додатку в робоче середовище при відносно простому синтаксисі.

Висновки

Як результат досліджень встановлено, що розробка програмного модуля системи безперервної поставки та розгортання коду для контейнеризованих додатків в AWS є актуальною для користувача та має практичне значення. Така розробка надаватиме значні можливості в інтеграції з хмарним провайдером, масштабуванням та використанням кращих практик розгортання коду.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Що таке CI/CD, як він працює та коли знадобиться на проєкті [Електронний ресурс]. – Режим доступу: <https://highload.today/uk/blogs/shho-take-ci-cd-yak-vin-pratsyuue-ta-koli-znadobitsya-na-proyekti-lajfhaki-ta-bad-practices/> (дата звернення: 20.02.2024). – Назва з екрана.
2. DevOps: найкращі інструменти CI/CD у 2022 році [Електронний ресурс]. – Режим доступу: <https://blog.iteducer.ua/ratings/best-ci-cd-tools/> (дата звернення: 20.02.2024). – Назва з екрана.
3. Key Differences Between CI/CD and DevOps [Електронний ресурс]. – Режим доступу: <https://testsigma.com/blog/devops-vs-cicd/> (дата звернення: 20.02.2024). – Назва з екрана.
4. Advice on CircleCI, GitLab CI, and Travis CI [Електронний ресурс]. – Режим доступу: <https://stackshare.io/stackups/circleci-vs-gitlab-ci-vs-travis-ci#decisions> (дата звернення: 20.02.2024). – Назва з екрана.
5. Continuous Integration: CircleCI vs Travis CI vs Jenkins [Електронний ресурс]. – Режим доступу: <https://djangostars.com/blog/continuous-integration-circleci-vs-travisci-vs-jenkins/> (дата звернення: 20.02.2024). – Назва з екрана.
6. CircleCI vs Travis CI vs Jenkins [Електронний ресурс]. – Режим доступу: <https://medium.com/hackernoon/continuous-integration-circleci-vs-travis-ci-vs-jenkins-41a1c2bd95f5> (дата звернення: 20.02.2024). – Назва з екрана.

Сенчик Роман Сергійович – студент групи 2КН-206, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, e-mail: rsenchuk5@gmail.com

Крилик Людмила Вікторівна – к.т.н, доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця.

Senchuk Roman S. – Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: rsenchuk5@gmail.com

Krylik Lyudmila V. – PhD (Eng.), Associate Professor of Department for Computer Science, Vinnytsia National Technical University, Vinnytsia.