

Єгор СЕНЮК

студент факультету ФІОТ

Київський політехнічний інститут імені Ігоря Сікорського

ORCID 0009-0003-0409-1421

seniuk.yehor@gmail.com

Вікторія МЕЛЬНИЧУК

доктор філософії з економіки, доцент кафедри економічної кібернетики

Київський політехнічний інститут імені Ігоря Сікорського

ORCID 0000-0001-8246-4076

ЗАСТОСУВАННЯ DOCKER ДЛЯ ЕКОНОМІЇ ЗАТРАТ ТА РЕІНЖИНІРИНГУ БІЗНЕС-ПРОЦЕСІВ

Yehor SENIUK

Student, Faculty of Informatics and Computer Engineering,

Igor Sikorsky Kyiv Polytechnic Institute.

ORCID: 0009-0003-0409-1421

seniuk.yehor@gmail.com

Victoria MELNYCHUK

Doctor of Philosophy in Economics, Associate Professor of the Department of Economic

Cybernetics

Igor Sikorsky Kyiv Polytechnic Institute.

ORCID: 0000-0001-8246-4076

APPLICATION OF DOCKER TO REDUCE COSTS AND BUSINESS PROCESS REENGINEERING

Сучасні бізнес-застосунки вимагають швидкої інтеграції, надійного розгортання та ефективного використання ресурсів. Традиційні підходи до розгортання часто супроводжуються проблемами несумісності середовищ, довгими циклами тестування та великими витратами на утримання інфраструктури. Контейнеризація, а особливо Docker, пропонує стандартизований спосіб запуску.

Docker — це платформа для створення, доставки та запуску застосунків у контейнерах [1]. Контейнер поєднує програму з усім необхідним для її виконання (залежності, бібліотеки, налаштування) у легковагове ізольоване середовище. На відміну від віртуальних машин, контейнери не потребують повної операційної системи, що сприяє економії ресурсів та швидшому запуску [2]. Даний інструмент допомагає при переформатуваннях підприємств, їх реінжинірингу бізнес-процесів.

Зважаючи на вищезазначене, можна підсумувати, що Docker використовується для оптимізації певних процесів, а саме стандартизує середовище, що дозволяє розробникам працювати в однаковому середовищі, яке ідентичне до середовища тестування та продакшну, відділяє різні проекти з використанням різних версій бібліотек без конфліктів, а також легко

масштабує локальні середовища (за допомогою Docker Compose та інших інструментів можна скласти мультиконтейнерні середовища для локальної розробки) [3].

Окрім цього, Docker пришвидшує доставку програмного забезпечення, що дає можливість для швидкого запуску, пришвидшуючи тестування та розгортання. Наступною перевагою є CI/CD інтеграція -контейнери легко включити в конвеєр безперервної інтеграції та доставки (Jenkins, GitLab CI, GitHub Actions та ін.), що скорочує час від коміту до релізу [4]. Цікавою є функція і Docker Registry (включаючи Docker Hub і приватні реєстри), яка дозволяє централізовано зберігати образи та швидко їх розгортати.

Аналізуючи економічну складову від використання Docker, варто зазначити, що даний інструмент, по-перше, сприяє економії ресурсів, тому що контейнери використовують ресурси ефективніше за віртуальні машини, що знижує витрати на інфраструктуру, в також прискорює час розробки і тестування [5]. Найважливішою складовою є зниження ризиків та витрат на помилки в продакшні, оскільки репліковані середовища зменшують кількість інцидентів, пов'язаних із несумісністю.

Розглядаючи кейси застосування у бізнесі, варто відмітити, що Docker можна застосовувати у багатьох різноманітних сферах. Наприклад, мікросервісна архітектура з використанням Docker передбачає, що кожний сервіс пакується в окремий контейнер і розгортається незалежно. Модернізація legacy-систем також можлива завдяки даному інструменту, оскільки контейнеризація дозволяє ізолювати старі сервіси та поступово перенести функціонал. Звісно, з тестовими середовищами і автоматизацією Docker справляється найкраще, адже наявне швидке створення тимчасових середовищ для тестування та демонстрацій клієнтам. Контейнери зручні для запуску як у приватних центрах обробки даних, так і в публічних хмарах [6].

Хоча Docker значно полегшує багато процесів, є питання безпеки, які потребують уваги: управління образами, вразливості у базових образах, правильне налаштування привілеїв контейнерів, мережеві політики та оновлення. Рекомендовано використовувати сканування образів, мінімальні базові образи, а також організувати процеси оновлення і моніторингу для збільшення продуктивності бізнес-процесів [7].

Рекомендації для впровадження

1. Починати з мінімального пілотного продукту: вибрати один сервіс або компонент для контейнеризації і оцінити вигоди.
2. Впроваджувати CI/CD: автоматизація конвеєру збільшить ефект від контейнеризації.
3. Вводити політики управління образами і безпекою: приватні реєстри, підписування образів, регулярне сканування.
4. Навчання команди: необхідні знання з Docker, оркестрації (Kubernetes), мережевих налаштувань та безпеки.

Висновок

Docker є потужним інструментом, що дозволяє спростити процес розробки, пришвидшити доставку програмного забезпечення та знизити витрати на інфраструктуру. При правильному впровадженні Docker допомагає підвищити якість продукту, скоротити час релізів і оптимізувати витрати, що робить його цінним ресурсом для сучасного бізнесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Merkel, D. (2014). Docker: Lightweight Linux Containers for Consistent Development and Deployment. Linux Journal.

2. Boettiger, C. (2015). An introduction to Docker for reproducible research. ACM SIGOPS Operating Systems Review.
3. Docker, Inc. (2024). Docker Documentation. <https://docs.docker.com>
4. Turnbull, J. (2014). The Docker Book: Containerization is the new virtualization. James Turnbull.
5. Pahl, C. (2015). Containerization and the PaaS cloud. IEEE Cloud Computing.
6. Jha, S., & Kabra, J. (2020). Container Orchestration and Best Practices. Journal of Cloud Computing.
7. Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A Software Architect's Perspective. Addison-Wesley.