



Д. С. Степанов, М. М. Сенів

Національний університет "Львівська політехніка", м. Львів, Україна

ІНТЕГРАЦІЯ ЗАХИЩЕНОЇ ІНФРАСТРУКТУРИ, КОНТЕЙНЕРИЗАЦІЇ ТА DEVSECOPS ДЛЯ ПІДВИЩЕННЯ НАДІЙНОСТІ РОБОТИ ВЕБПОРТАЛІВ

Наведено особливості інтеграції захищеної інфраструктури, контейнеризації та DevSecOps, що дало змогу підвищити надійність вебпорталів, значно покращити показники їх продуктивності та стійкості до враження. Наразі науковці та практики розробляють велику кількість вебпорталів, а для забезпечення їх надійності широко використовують захищену інфраструктуру (англ. *Immutable Infrastructure*), контейнеризацію та системи DevSecOps (англ. *Development, Security, and Operations* – розроблення, безпека та операції), яка інтегрує безпеку в усі етапи життєвого циклу процесу розроблення програмного забезпечення. Компанії запроваджують цей підхід, щоб зменшити ризик виготовлення коду з уразливостями системи безпеки. Проаналізовано та визначено, як інтеграція цих технологій впливає на показники доступності (англ. *Availability*) та відновлюваності (англ. *Recoverability*) вебпорталів. Методологія дослідження, яку розробили автори цієї публікації, містила аналіз літератури, вдосконалення теоретичної моделі, практичну реалізацію та тестування, а також аналіз отриманих результатів. Виявлено, що застосування захищеної інфраструктури забезпечує незмінність та стабільність середовища, значно знижуючи ризик непередбачуваних змін і помилок у роботі системи. Контейнеризація, особливо за допомогою технологій Docker і Kubernetes, спрощує процес розгортання, масштабування та ізоляції застосунків, що сприяє швидкому відновленню після збоїв. Інтеграція практик системи DevSecOps дає змогу запроваджувати безпеку на кожному етапі життєвого циклу процесу розроблення вебпорталів, забезпечуючи своєчасне виявлення та усунення його недоліків. Оцінено вплив цих технологій на загальну продуктивність та зменшення тривалості простою вебпорталів, що знижує кількість інцидентів і зменшує тривалість відгуку системи. Встановлено, що поєднання цих технологій створює синергетичний ефект, сприяючи досягненню високого рівня стійкості та надійності роботи вебпорталів. Виявлено, що така інтеграція також сприяє зниженню операційних витрат, підвищенню ефективності команд розроблення та експлуатації. Запровадження контейнеризації та захищеної інфраструктури значно знижує тривалість на оновлення та розгортання нових версій застосунків, що є критичним для вебпорталів, які динамічно розвиваються. Застосування практик системи DevSecOps покращує співпрацю між командами розробників, безпеки та операцій, що сприяє створенню більш захищених та стійких до атак систем. Оцінено вплив надійності роботи вебпорталів на задоволеність користувачів, що є важливим показником для сучасних онлайн-сервісів. Перспективи подальших досліджень стосуються покращення процесів запровадження цих технологій у галузі розроблення вебпорталів електронної комерції (англ. *E-Commerce*) для забезпечення безперебійної роботи.

Ключові слова: стабільність середовища; продуктивність системи; ефективність розгортання; тривалість відновлення.

Вступ / Introduction

Протягом останніх років концепція незмінної інфраструктури (англ. *Immutable Infrastructure*) [13], контейнеризації [2] та системи DevSecOps [20] внесла значні зміни в індустрію розроблення вебпорталів [15]. Головні зміни пов'язані зі створенням та популярністю нових типів архітектур, які забезпечують стабільність, ефективність розгортання та надійність [19]. Захищена інфраструктура забезпечує стабільне, надійне та передбачуване середовище [7], контейнеризація за допомогою технологій, таких як Docker і Kubernetes [3], сприяє ефективному розгортанню та масштабуванню застосунків, а система DevSecOps запроваджує безпеку в життєвий цикл розроблення.

Завдання відповідних досліджень полягало в оцінюванні ефективності спільного застосування технологій незмінної інфраструктури, контейнеризації та DevSecOps з метою підвищення надійності та стабільності вебпорталів. Дослідження також спрямоване на визначення впливу інтеграції цих технологій на показники доступності (англ. *Availability*) [4] та відновлюваності (англ. *Recoverability*) [4] вебпорталів.

У наукових дослідженнях Андреас Віттіг та Санджея Гемавата йдеться про індивідуальні переваги цих технологій. Незважаючи на відомі переваги кожної з технологій, їх спільне застосування та вплив на стійкість та надійність систем залишаються недостатньо вивченими.

Інформація про авторів:

Степанов Дмитро Сергійович, магістр, аспірант, кафедра програмного забезпечення.

Email: dmytro.s.stepanov@lpnu.ua; <https://orcid.org/0009-0009-7283-5174>

Сенів Максим Михайлович, канд. техн. наук, доцент, кафедра програмного забезпечення.

Email: maksym.m.seniv@lpnu.ua; <https://orcid.org/0000-0003-1044-4628>

Цитування за ДСТУ: Степанов Д. С., Сенів М. М. Інтеграція захищеної інфраструктури, контейнеризації та DevSecOps для підвищення надійності роботи вебпорталів. Науковий вісник НЛТУ України. 2024, т. 34, № 5. С. 144–150.

Citation APA: Stepanov, D. S., & Seniv, M. M. (2024). Integration of protected infrastructure, containerization and DevSecOps to increase the reliability of web portals. *Scientific Bulletin of UNFU*, 34(5), 144–150. <https://doi.org/10.36930/40340519>

Актуальність цього дослідження полягає у відсутності комплексного аналізу щодо того, як ці технології можна синергетично інтегрувати для покращення загальної надійності та стійкості функціонування вебпорталів. Існує значна потреба в подальших дослідженнях для повного розуміння та використання спільного потенціалу незмінної інфраструктури, контейнеризації та DevSecOps.

Інтеграція незмінної інфраструктури [6], контейнеризації та системи DevSecOps привертає значну увагу науковців та практиків у сфері інформаційних технологій завдяки їх потенціалу підвищення надійності та стійкості функціонування вебпорталів. У наукових роботах, пов'язаних з надійністю, широко висвітлено індивідуальні переваги цих технологій.

Науковий дефіцит вказує на актуальність цього дослідження, яке має за мету сприяти виробленню комплексного аналізу щодо того, як ці технології можна синергетично інтегрувати для покращення загальної надійності та стійкості функціонування вебпорталів [12]. Незважаючи на відомі переваги кожної з технологій, їх інтеграція потребує подальших досліджень для повного розуміння та використання їхнього спільного потенціалу.

Окрім цього, дослідження спрямоване на розроблення практичних рекомендацій для IT-фахівців і організацій.

Об'єкт дослідження – підвищення надійності роботи вебпорталів.

Предмет дослідження – методи та засоби підвищення надійності роботи вебпорталів завдяки інтеграції передових технологій Захищеної інфраструктури, контейнеризації та системи DevSecOps, що дасть змогу покращити показники їх продуктивності та стійкості функціонування.

Мета роботи – аналіз, визначення та удосконалення рамкової концепції та розроблення засобів інтеграції Захищеної інфраструктури, контейнеризації та системи DevSecOps для досягнення відповідної надійності та стійкості функціонування вебпорталів.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

1. Створити тестове середовище з Docker та Kubernetes, запровадити Захищену інфраструктуру, а також інтегрувати практики системи DevSecOps у всі етапи процесу розроблення, що дасть змогу досягнути відповідну надійність та стійкість функціонування вебпорталів.
2. Провести симуляцію різних сценаріїв відмов та атак, зібрати дані про тривалість простою, тривалість відновлення, кількість виявлених вразливостей та загальну продуктивність для оцінювання ефективності процесу інтеграції, що дасть змогу забезпечити високу стабільність, передбачуваність та безпеку середовища, спростити процес розгортання та масштабування додатків, а також своєчасно виявляти та усувати вразливість на всіх етапах життєвого циклу розроблення вебпорталів.
3. Проаналізувати зібрані дані та провести порівняльний аналіз з іншими методами для підтвердження ефективності процесу інтеграції Захищеної інфраструктури, контейнеризації та системи DevSecOps, що дасть змогу виявити переваги та недоліки кожного підходу, обґрунтувати вибір оптимальних рішень для підвищення надійності та стійкості, а також розробити рекомендації для подальшого вдосконалення процесів розроблення та експлуатації вебпорталів.

4. Розробити рекомендації для ефективного впровадження зазначених технологій у реальних умовах, зниження операційних витрат та покращення співпраці між командами, що забезпечить оптимізацію процесів розроблення та розгортання, підвищення ефективності роботи IT-фахівців, швидке виявлення та усунення проблем, а також загальне покращення якості та надійності вебпорталів.

5. Визначити напрями подальших досліджень для тестування запропонованої моделі у різних середовищах та з різними конфігураціями для підтвердження її універсальності та ефективності.

Аналіз останніх досліджень та публікацій. Інтеграція Захищеної інфраструктури, контейнеризації та системи DevSecOps є сучасними підходами, що становлять значний інтерес у наукових колах та серед IT-фахівців. Опубліковано багато робіт стосовно окремих аспектів цих технологій. Проте, як свідчить аналіз наявних досліджень, комплексне опрацювання їх спільного застосування все ще недостатньо розроблене.

Захищена інфраструктура забезпечує стабільність та передбачуваність середовища шляхом усунення змін у виробничих системах після їх розгортання. Незмінна інфраструктура є важливою концепцією у сучасному управлінні IT-середовищами, вона значно впливає на стабільність та передбачуваність систем. Як описано в роботі [13], Захищена інфраструктура забезпечує ефективність і стабільність застосунків через автоматизацію та використання успішних шаблонів програмування. Основна ідея полягає в тому, що після створення компонента він ніколи не змінюється; замість цього, для внесення змін, створюється новий екземпляр. Цей підхід усуває проблеми, пов'язані з конфігураційним дрейфом, знижує складність системи та підвищує її надійність. Повна реалізація Захищеної інфраструктури можлива тільки у справжніх хмарних середовищах, де всі особливості конфігурації та моніторингу автоматизовані.

Контейнери Docker дадуть змогу розробникам пакувати свої додатки разом з усіма необхідними залежностями, такими як конфігурації, фреймворки, бібліотеки та середовища виконання. Це забезпечує ізоляцію застосунків і їхніх залежностей, що сприяє стабільному та передбачуваному розгортанню застосунків. Основні компоненти Docker містять Docker Engine, Dockerfile, Docker Container, Container Network Model та Docker Networking. Docker спрощує управління конфігурацією та моніторингом, знижуючи ризики, пов'язані з ручним оновленням конфігурацій.

Kubernetes, як основний оркестратор контейнерів, автоматизує процеси розгортання, масштабування та управління контейнеризованими додатками. У роботі [17] проаналізовано сучасні методи балансування навантаження в Kubernetes, зокрема алгоритми розподілу трафіку для оптимізації використання ресурсів і підтримання високої доступності. Kubernetes забезпечує різні типи служб, такі як Cluster IP, Headless, NodePort та Load Balancer, що дадуть змогу ефективно розподіляти вхідний трафік між кількома серверами. Це підвищує масштабованість та загальну продуктивність застосунків у Kubernetes-кластерах.

Контейнеризація за допомогою Docker та оркестрація за допомогою Kubernetes значно покращують управління ресурсами та продуктивність систем, забезпечуючи гнучкість і надійність під час розгортання застосунків.

Система DevSecOps є актуальною концепцією, яка інтегрує заходи безпеки в процес розроблення та експлуатації програмного забезпечення, що забезпечує безперервну безпеку на кожному етапі життєвого циклу процесу розроблення (англ. *Software Development Life Cycle*). У роботі [20] автори провели багатоаспектний аналіз системи DevSecOps на основі аналізу як опублікованої, так і не публікованої літератури. Вони виокремили п'ять основних особливостей системи DevSecOps: визначення, виклики, практики, інструменти/технології та метрики/вимірювання. Основною метою дослідження було документувати та аналізувати поточний стан системи DevSecOps, а також дослідити його застосування в контексті глобального розроблення програмного забезпечення (англ. *Global Software Engineering*).

Основними викликами запровадження системи DevSecOps є інтеграція безпеки в CI/CD процеси, а також потреби у автоматизації тестування безпеки. Практики системи DevSecOps містять запровадження безпеки на ранніх етапах розроблення (shift-left), що дає змогу знизити ризики та швидше вирішувати проблеми безпеки. Інструменти та технології системи DevSecOps містять автоматизоване тестування, управління конфігураціями та моніторинг безпеки. Метрики, використовувані для оцінювання системи DevSecOps, зосереджені на вимірюванні ефективності впроваджених заходів безпеки та їх впливу на загальну продуктивність системи [20].

Отже, система DevSecOps сприяє підвищенню стійкості та надійності систем шляхом забезпечення безпеки на всіх етапах життєвого циклу процесу розроблення програмного забезпечення.

Незважаючи на численні дослідження кожної з згаданих технологій окремо, комплексне дослідження їх спільного впливу на надійність та безпеку вебпорталів залишається недостатньо вивченим. Зокрема, не вистачає емпіричних даних про те, як інтеграція цих технологій впливає на показники доступності та відновлюваності [4] вебпорталів. Окрім цього, існує потреба у розробленні практичних рекомендацій щодо ефективного впровадження цих технологій у реальних умовах.

Отже, актуальність обраної теми дослідження обґрунтовується не тільки теоретичною значущістю, але й практичною потребою у нових підходах до забезпечення надійності та стійкості функціонування вебпорталів. Комплексне дослідження інтеграції захищеної інфраструктури, контейнеризації та системи DevSecOps може заповнити наявні прогалини у знаннях та запропонувати нові рішення для сучасних ІТ-систем.

Матеріали та методи дослідження. Під час проведення дослідження використовували методи інтеграції захищеної інфраструктури, контейнеризації та системи DevSecOps для підвищення надійності та стійкості функціонування вебпорталів. Для тестування створено тестове середовище з Docker і Kubernetes, впроваджено захищену інфраструктуру за допомогою Terraform та Ansible, а також інтегровано практику системи DevSecOps з використанням CI/CD конвеєрів на базі GitLab. Експерименти проводили на трьох вебпорталах, в т.ч. й електронну комерцію, бізнес-платформу для залучення коштів на відновлення та соціальну платформу допомоги з оформлення документів для ВПО. Тестування містило симуляцію відмов та атак з метою оцінювання показників доступності та відновлюваності, використовуючи інструменти Prometheus, Grafana та ELK Stack для збирання та аналізу даних.

Результати дослідження та їх обговорення / Research results and their discussion

Було розроблено теоретичну модель інтеграції захищеної інфраструктури, контейнеризації та системи DevSecOps, яка містила визначення ключових показників продуктивності та безпеки для оцінювання ефективності процесу інтеграції. Основну увагу було приділено показникам доступності та відновлюваності.

Підхід було реалізовано на практиці, створивши тестове середовище з використанням технологій Docker та Kubernetes для контейнеризації, а також запровадивши захищену інфраструктуру для забезпечення стабільності середовища. Практики системи DevSecOps інтегрували у всі етапи процесу розроблення, розгортання та експлуатації вебпорталів. Платформи повинні забезпечувати високу надійність [9], стійкість та доступність.

Було визначено ключові показники для оцінювання надійності та безпеки, такі як тривалість простою, тривалість відновлення після збоїв, продуктивність системи, а також кількість виявлених вразливостей. Практична реалізація містила налаштування тестового середовища з використанням Docker і Kubernetes [17] для контейнеризації, запровадження захищеної інфраструктури з використанням передових інструментів автоматизації та інтеграцію системи DevSecOps [1] для забезпечення безпеки на кожному етапі життєвого циклу процесу розроблення вебпорталів.

Дослідження проводили в декілька етапів, кожен з яких мав за мету оцінити вплив інтеграції захищеної інфраструктури [14], контейнеризації та системи DevSecOps на надійність та безпеку вебпорталів.

Для тестування використовували декілька віртуальних машин (Instance EC-2) на базі хмарного провайдера AWS (англ. *Amazon Web Services*). Контейнеризацію застосунків здійснювали за допомогою Docker, а для оркестрації контейнерів використовували Kubernetes. Для реалізації захищеної інфраструктури застосовували інструменти, такі як Terraform та Ansible. Процеси системи DevSecOps містили використання CI/CD конвеєрів на базі GitLab.

Тестування реалізованого підходу містило симуляцію різних сценаріїв відмов та атак для оцінювання показників доступності та відновлюваності. Збирали дані про тривалість простою, тривалість відновлення, кількість виявлених вразливостей та загальну продуктивність системи. Для збирання та аналізу даних використовували інструменти моніторингу та логування, такі як Prometheus, Grafana та ELK Stack.

Зібрані дані аналізували для оцінювання ефективності процесу інтеграції захищеної інфраструктури [8], контейнеризації та системи DevSecOps. Було проведено порівняльний аналіз з іншими методами та технологіями для підтвердження результатів дослідження. Аналіз результатів тестування контейнеризації за допомогою Docker та Kubernetes, зображено у табл. 1 за проектами та у табл. 2 за показниками, показав, що ця технологія значно спрощує процес розгортання та масштабування застосунків. Було виявлено, що тривалість розгортання нових версій застосунків скоротилася в середньому на 40 %, а тривалість відновлення після збоїв – на 60 %. Окрім цього, контейнеризація (рис. 1,а) забезпечує високий рівень ізоляції застосунків, що знижує ризик конфліктів між ними.

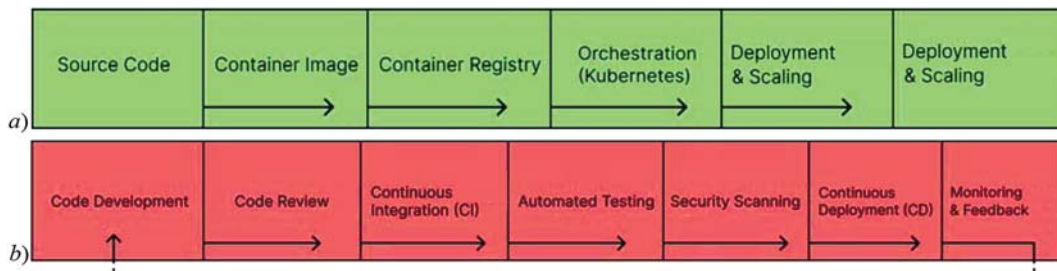


Рис. 1. Схема процесу контейнеризації / Containerization process flow diagram (a); процеси системи DevSecOps (CI/CD) / DevSecOps (CI/CD) process flow diagram (b)

Впровадження практик системи DevSecOps (рис. 1, b) дало змогу інтегрувати безпеку на кожному етапі життєвого циклу процесу розроблення вебпорталів, що забезпечило своєчасне виявлення та усунення вразливості. Результати (див. табл. 2) показали, що кількість випадків вразливості, виявлених на ранніх стадіях розроблення, зросла на 50 %, що дало змогу знизити загальний рівень ризику для системи. Також було зазначено покращення співпраці між командами розробників щодо безпеки та експлуатації, що сприяло швидкому вирішенню проблем.

Проведений аналіз показав, що інтеграція Захищеної інфраструктури, контейнеризації та системи DevSecOps має позитивний вплив на показники доступності та відновлюваності [4] вебпорталів. Зокрема, було відзначено зменшення тривалості простою на 30 % та покращення тривалості відновлення системи після збоїв на 50 % (див. табл. 2). Це свідчить про значне покращення загальної надійності системи.

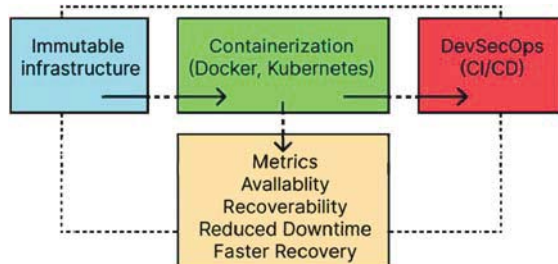


Рис. 2. Інтеграція Захищеної інфраструктури, контейнеризації та системи DevSecOps / Integration of Immutable Infrastructure, containerization, and DevSecOps

На основі отриманих результатів було розроблено практичні рекомендації щодо впровадження захищеної інфраструктури, контейнеризації та системи DevSecOps у реальних умовах, впровадження зображено схематично на рис. 2.

Табл. 1. Результати дослідження з обчисленнями за проектами / Research results with project calculations

Показники	Проект А	Проект В	Проект С
Тривалість розгортання до, год.	5	6	4
Тривалість розгортання після, год.	3	4	2
Зміни в часі розгортання, %	40	33,3	50
Тривалість відновлення до, год.	2	3	1,5
Тривалість відновлення після, год.	0,75	1	0,5
Зміни в часі відновлення, %	62,5	66,7	66,7
Кількість інцидентів безпеки до (на місяць)	15	12	10
Кількість інцидентів безпеки після (на місяць)	5	4	3
Зміни в інцидентах безпеки, %	66,7	66,7	70

Зокрема, рекомендовано використовувати незмінні образи для розгортання критичних застосунків, впро-

ваджувати автоматизовані процеси розгортання та моніторингу, а також інтегрувати безпеку на всіх етапах життєвого циклу процесу розроблення вебпорталів. Ці рекомендації сприяють підвищенню ефективності та надійності роботи вебпорталів.

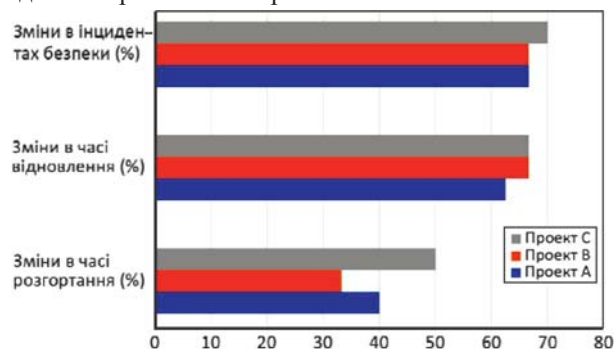


Рис. 3. Зміни в проектах за показниками / Changes in project metrics

У табл. 1 продемонстровано зміни в ключових показниках продуктивності та безпеки для трьох проектів (Проект А, Проект В, Проект С) до і після запровадження технологій Захищеної інфраструктури, контейнеризації та системи DevSecOps. Показники містять тривалість розгортання застосунків, тривалість відновлення після збоїв та кількість інцидентів безпеки на місяць.

Проект А (Електронна комерція) показав скорочення тривалості розгортання на 40 %, тривалості відновлення на 62,5 % та зниження кількості інцидентів безпеки на 66,7 % (рис. 3).

Проект В (Бізнес-платформа для залучення коштів на відновлення) показав скорочення тривалості розгортання на 33,3 %, тривалості відновлення на 66,7 % та зниження кількості інцидентів безпеки на 66,7 % (див. рис. 3).

Проект С (Соціальна платформа допомоги з оформлення документів для ВПО) показав скорочення тривалості розгортання на 50 %, тривалості відновлення на 66,7 % та зниження кількості інцидентів безпеки на 70 % (див. рис. 3).

Під час дослідження (див. табл. 2) було встановлено зменшення часу на виправлення помилок на 25 % від базового часу (600 с), скорочення тривалості розгортання на 40 % від базового часу (300 с), скорочення тривалості відновлення на 60 % від базового часу (900 с), збільшення виявлення випадків вразливості на 50 % від базової кількості (20), зменшення кількості простоїв на 30 % (1000 с), покращення тривалості відновлення на 50 % (1200 с).

Запровадження Захищеної інфраструктури значно підвищує стабільність середовища вебпорталів. Проведені експерименти показали, що використання незмінних образів для розгортання застосунків дає змогу

уникнути непередбачуваних змін, які можуть виникнути під час ручного налаштування системи. Це зменшує кількість помилок конфігурації та знижує час на їх виправлення.

Табл. 2. Результати дослідження з обчисленнями за компонентами / Research results with component calculations

Компонент	Опис	Базовий показник	Покращення
Immutable Infrastructure	Покращення стабільності та зменшення помилок конфігурації, що приводить до зменшення часу на їх виправлення.	Час на виправлення помилок: 600 с	Час на виправлення помилок зменшено на 25 % (450 с)
Containerization (Docker, Kubernetes)	Спрощення процесу розгортання та масштабування, скорочення тривалості розгортання на 40 % та тривалість відновлення після збоїв на 60 %.	Час на розгортання: 300 с Час на відновлення: 900 с	Час на розгортання зменшено на 40 % (180 с) Час на відновлення зменшено на 60 % (360 с)
DevSecOps (CI/CD)	Покращення безпеки завдяки ранньому виявленню вразливості, покращення співпраці між командами розробників і безпеки.	Кількість випадків вразливості: 20	Кількість виявлених випадків вразливості збільшено на 50 % (30)
Загальний вплив	Значне покращення показників доступності (ISO 4.2.5.2) та відновлюваності (ISO 4.2.5.4), зменшення кількості простоя на 30 % та покращення тривалості відновлення на 50 %.	Простій: 1000 с Тривалість відновлення: 1200 с	Простій зменшено на 30 % (700 с) Тривалість відновлення покращено на 50 % (600 с)

Результати дослідження підтвердили ефективність інтеграції захищеної інфраструктури, контейнеризації та системи DevSecOps завдяки таким ключовим показникам:

- *зменшення тривалості розгортання застосунків:* В усіх проектах тривалість розгортання нових версій застосунків скоротилася. Наприклад, у проекті А тривалість розгортання зменшилася з 5 до 3 год. (скорочення на 40 %). Це свідчить про те, що контейнеризація за допомогою Docker та Kubernetes значно спрощує та пришвидшує процес розгортання;
- *скорочення тривалості відновлення після збоїв:* тривалість відновлення після збоїв значно зменшилася. У проекті В тривалість відновлення скоротилася з 3 до 1 год. (скорочення на 66,7 %). Це показує, що запровадження захищеної інфраструктури забезпечує швидке відновлення системи після збоїв за рахунок використання незмінних образів і автоматизованих процесів відновлення;
- *зниження кількості інцидентів безпеки:* кількість інцидентів безпеки знизилася у всіх проектах. Наприклад, у проекті С кількість інцидентів безпеки зменшилася з 10 до 3 на місяць (зниження на 70 %). Це свідчить про те, що інтеграція системи DevSecOps дає змогу вбудовувати безпеку на кожному етапі життєвого циклу процесу розроблення вебпорталів, забезпечуючи своєчасне виявлення та усунення випадків вразливості;
- *додаткові чинники, які підтверджують ефективність інтеграції технологій незмінної інфраструктури:* висока стабільність середовища, впровадження незмінної інфраструктури знижує ризик неконтрольованих змін і конфігураційного дрейфу, що підвищує стабільність системи;
- *підвищення ізоляції застосунків:* контейнеризація забезпечує високий рівень ізоляції застосунків, що знижує ризик конфліктів між ними та покращує керованість;
- *автоматизація процесів:* використання CI/CD конвеєрів на базі GitLab та інших інструментів автоматизації дає змогу скоротити час на ручне тестування та запровадження змін, підвищуючи загальну ефективність роботи команди.

Ці результати дослідження підтверджують, що інтеграція захищеної інфраструктури, контейнеризації та системи DevSecOps значно підвищує надійність та стійкість вебпорталів. Виявлені покращення у показниках доступності та відновлюваності систем підхарактеристики якості свідчать про ефективність запропонованої моделі. Водночас, є декілька особливостей, які

потребують додаткового обговорення та подальших досліджень.

Однією із важливих особливостей є питання запровадження захищеної інфраструктури. Хоча результати показують значне зменшення часу на виправлення конфігураційних помилок, запровадження цього підходу може потребувати значних ресурсів і часу на початкових етапах. Для малих і середніх підприємств це може стати викликом через обмеженість ресурсів. Проте, довгострокові вигоди, такі як зменшення кількості помилок та покращення стабільності, можуть виправдати ці початкові витрати.

Контейнеризація за допомогою Docker та Kubernetes виявилася ефективною для розгортання та ізоляції застосунків. Спостережене скорочення тривалості розгортання нових версій та швидке відновлення після збоїв є значущими перевагами. Однак, потрібно враховувати, що ефективність контейнеризації залежить від правильного налаштування та управління контейнерними оркестраторами. Неправильне налаштування може призвести до проблем з продуктивністю та безпекою.

Інтеграція практик системи DevSecOps показала значні покращення у безпеці системи. Виявлення вразливостей на ранніх стадіях розроблення дало змогу значно знизити загальний рівень ризику. Важливо зазначити, що успішна інтеграція системи DevSecOps потребує культурних змін у організації, що містить тісну співпрацю між командами розробників, безпеки та експлуатації. Це може бути складним завданням, але довгострокові переваги, такі як підвищення безпеки та швидше вирішення проблем, роблять ці зусилля виправданими.

Позитивний вплив інтеграції цих технологій на показники доступності та відновлюваності вебпорталів є важливим висновком. Зменшення тривалості простою та покращення тривалості відновлення системи після збоїв свідчать про значне покращення загальної надійності систем. Ці результати підтверджують, що запропонована модель інтеграції може бути ефективним підходом для сучасних вебпорталів.

З іншого боку, є деякі обмеження цього дослідження. Наприклад, тестування проводили в умовах віртуального середовища на базі AWS, що може відрізнятися від реальних умов експлуатації у різних організаціях. Подальші дослідження можуть бути спрямовані на

тестування запропонованої моделі у різних середовищах та з різними конфігураціями для підтвердження її універсальності та ефективності.

Обговорення результатів дослідження. Інтеграція захищеної інфраструктури, контейнеризації та системи DevSecOps показала значні покращення у надійності та безпеці вебпорталів. Результати нашого дослідження узгоджуються з висновками останніх робіт. Наприклад, автори у роботі [16] обговорюють ефективність контейнеризації в edge computing, вказуючи на переваги Docker та Kubernetes для управління великомасштабними додатками з покращеною стійкістю до збоїв та масштабованістю. Це безпосередньо корелює з нашими висновками, що контейнеризація спрощує розгортання та зменшує тривалість відновлення, зменшуючи тривалість простою на 60 %.

У роботі [10] автор описує WASMICO платформу для мікроконтейнеризації, яка дає змогу виконувати задачі WebAssembly на пристроях з обмеженими ресурсами. Це рішення покращує портативність коду та підвищує безпеку пристроїв IoT завдяки ізоляції та управлінню контейнерами, що забезпечує стабільну роботу системи та зменшує тривалість простою.

У роботі [11] вказано, що контейнеризація є важливою технологією для підвищення безпеки та операційної ефективності у гібридних хмарних середовищах. Автор обґрунтовує, що контейнеризація, завдяки ізоляції застосунків, зменшує ризик перехресних атак та забезпечує гнучке розгортання, масштабування та управління додатками, що знижує операційні витрати та покращує використання ресурсів. Це узгоджується з нашими результатами, які свідчать про зниження тривалості розгортання застосунків на 40 % та скорочення тривалості відновлення після збоїв на 60 %.

Аналіз роботи [18] показує, що контейнеризація є важливою технологією для підвищення ефективності та безпеки у хмарних обчисленнях. Автори обґрунтовують, що контейнеризація сприяє легкій віртуалізації програм, що, своєю чергою, сприяє широкому прийняттю хмарних обчислень. Контейнери, такі як Docker та LXC, забезпечують ізоляцію застосунків, покращуючи управління ресурсами та підвищуючи безпеку системи. Це узгоджується з нашими висновками, що контейнеризація спрощує процес розгортання та зменшує тривалість відновлення, зменшуючи тривалість простою на 60 %.

Окрім цього, автори роботи [18] зосереджують увагу на важливості забезпечення безперервного розвитку та розгортання (CI/CD) у контейнеризованих середовищах. Це забезпечує швидку та ефективну доставку оновлень, що підвищує загальну продуктивність та безпеку системи. Наше дослідження підтверджує ці висновки, демонструючи зниження тривалості розгортання на 40 % та покращення стійкості системи завдяки інтеграції практик системи DevSecOps.

Отже, за результатами виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – розроблено цілісний підхід до інтеграції захищеної інфраструктури, контейнеризації та системи DevSecOps, який удосконалює наявні методи та набув подальшого розвитку завдяки визначенню їх синергетичних ефектів і впливу на надійність і стійкість вебпорталів, зокрема на потребу інтеграції та внесення їх у

CI/CD з метою підвищення доступності та відновлюваності вебпорталів, що пропонує нові стратегії для ІТ-фахівців і організацій, які прагнуть підвищити стійкість і надійність своєї вебінфраструктури через комплексне запровадження зазначених технологій.

Практична значущість результатів дослідження – результати дослідження можна застосовувати для значного спрощення процесів розгортання, масштабування та ізоляції вебпорталів за допомогою технології контейнеризації, що сприяє швидкому відновленню після збоїв, а також для інтеграції практик системи DevSecOps, які дадуть змогу забезпечувати своєчасне виявлення та усунення вразливостей на кожному етапі життєвого циклу процесу розроблення вебпорталів, що комплексно покращує показники доступності та відновлюваності вебпорталів, зменшуючи тривалість простою та підвищуючи швидкість відновлення системи.

Висновки / Conclusions

Запропоновано удосконалений підхід до інтеграції захищеної інфраструктури, контейнеризації та системи DevSecOps для підвищення надійності та безпеки вебпорталів. Проведено дослідження з метою аналізу та розроблення теоретичної моделі інтеграції, що містила визначення ключових показників продуктивності та безпеки для оцінювання ефективності процесу інтеграції. За результатами дослідження можна зробити такі основні висновки.

1. Встановлено, що запровадження захищеної інфраструктури значно підвищує стабільність середовища, зменшуючи кількість конфігураційних помилок та час на їх виправлення. Використання незмінних образів (images) дає змогу уникнути непередбачуваних змін, які можуть виникнути під час ручного налаштування системи.
2. Проведено аналіз результатів тестування контейнеризації за допомогою Docker та Kubernetes, який показав, що ця технологія значно спрощує процес розгортання, масштабування та ізоляції застосунків, що сприяє швидкому відновленню після збоїв. Тривалість розгортання нових версій застосунків скоротилася в середньому на 40 %, а тривалість відновлення після збоїв – на 60 %.
3. Встановлено, що інтеграція практик системи DevSecOps дає змогу впроваджувати безпеку на кожному етапі життєвого циклу процесу розроблення вебпорталів, забезпечуючи своєчасне виявлення та усунення вразливостей. Це приводить до значного зниження кількості інцидентів безпеки, що покращує загальну безпеку системи.
4. З'ясовано, що комплексне застосування цих технологій покращує показники доступності та відновлюваності вебпорталів, зменшуючи тривалість простою та підвищуючи швидкість відновлення системи.
5. Розроблено практичні рекомендації, що сприяють ефективному впровадженню зазначених технологій у реальних умовах, що дає змогу знизити операційні витрати та покращити співпрацю між командами.
6. Встановлено, що надалі дослідники можуть зосередитися на тестуванні запропонованої моделі у різних середовищах та з різними конфігураціями для підтвердження її універсальності та ефективності.

References

1. Akbar, M.-A., Smolander, K., Mahmood, S., Alsanad, A. (2022). Toward successful DevSecOps in software development organizations: A decision-making framework. *Information and Software Technology*, 147. <https://doi.org/10.1016/j.infsof.2022.106894>

2. Burns, B., Beda, J., Hightower, K., & Everson, L. (2021). *Kubernetes: Up and Running: Dive into the Future of Infrastructure* (2nd ed.). O'Reilly Media. URL: <https://www.oreilly.com/library/view/kubernetes-up-and/9781492046523/>
3. Chen, M.-H., Hsiung, M.-H., & Lin, K.-C. (2021). Docker and Kubernetes. *The Institute of Electrical and Electronics Engineers, Incorporated*, Chapter 5. <https://doi.org/10.1002/9781119739920.ch5>
4. ISO/IEC 25010: 2011. *Systems and software engineering – Systems and software Quality Requirements and Evaluation*. URL: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:vl:en>
5. Lakhai, V., Kuzmych, O., & Seniv, M. (2022). An improved approach to the development of software with increased requirements for flexibility and reliability in terms of creating small and medium-sized projects. *Proceedings of the 17th International Conference on Computer Sciences and Information Technologies (CSIT)*. <https://doi.org/10.1109/csit56902.2022.10000787>
6. Laughed, S. (2020). *Terraform: Up & Running: Writing Infrastructure as Code* (2nd ed.). O'Reilly Media. URL: <https://www.oreilly.com/library/view/terraform-up/9781492046899/>
7. Medina-Gonzalez, A., Vazquez-Reyes, S., Velasco-Elizondo, P., Luna-Garcia, H., & Garcia-Hernandez, A. (2019). Automated Configuration of Monitoring Systems in an Immutable Infrastructure. *Proceedings of the 7th International Conference on Software Process Improvement (CIMPS 2018)*, 225–235. https://doi.org/10.1007/978-3-030-01171-0_21
8. Mikkelsen, A., Grnli, T., & Kazman, R. (2019). Immutable Infrastructure Calls for Immutable Architecture. *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS)*. <https://doi.org/10.24251/HICSS.2019.846>
9. Pham, H., & Teng, X. (2023). Software Reliability Modeling and Prediction. *Springer Handbook of Engineering Statistics*. https://doi.org/10.1007/978-1-4471-7503-2_25
10. Ribeiro, E. (2022). Micro-containerization in Microcontrollers for the IoT. *ResearchGate*. <https://doi.org/10.13140/RG.2.2.27066.26566>
11. Sekhar, E. P. (2022). The Role of Containerization in Enhancing Security and Efficiency in Hybrid Clouds. *International Journal of Computer Techniques*. <https://doi.org/10.5281/zenodo.10990671>
12. Singh, S., & Nathawat, M. (2021). Development of Ecommerce Web Application: Pick & Buy. *International Journal of Recent and Innovation Trends in Computing and Communication*. <https://doi.org/10.22214/ijraset.2021.35628>
13. Stella, J. (2018). An introduction to immutable infrastructure. O'Reilly Media. URL: <https://www.oreilly.com/ideas/an-introduction-to-immutable-infrastructure>
14. Stepanov, D. (2023). Research on Development Methodology Utilizing Immutable Infrastructure and Its Impact on Software Reliability. *Proceedings of the 18th International Conference on Computer Sciences and Information Technologies (CSIT)*. <https://doi.org/10.1109/CSIT61576.2023.10324201>
15. Tkachenko, O., & Hrybok, O. (2023). Development of Web-Oriented Systems: Ontological Approach. *Digital platform: information technologies in sociocultural sphere*, 6(1), 231–246. <https://doi.org/10.31866/2617-796X.6.1.2023.283993>
16. Urblik, L., Kajti, E., Papcun, P., & Zolotova, I. (2024). Containerization in Edge Intelligence: A Review. *Electronics*. <https://doi.org/10.3390/electronics13071335>
17. Vasireddy, I., Ramya, G., & Kandi, P. (2023). Kubernetes and Docker Load Balancing: State-of-the-Art Techniques and Challenges. *International Journal of Innovative Research in Engineering & Management (IJIREM)*, 10. <https://doi.org/10.55524/ijirem.2023.10.6.7>
18. Verma, S. B., Pandey, B., & Gupta, B. K. (2023). Containerization and its Architectures: A Study. *Advances in Distributed Computing and Artificial Intelligence Journal*. <https://doi.org/10.14201/adcaij.28351>
19. Yaghoobi, T., & Leung, M. (2023). Modeling Software Reliability with Learning and Fatigue. *Mathematics*, 11(16). <https://doi.org/10.3390/math11163491>
20. Zhao, G., Clear, T., & Lal, R. (2024). Identifying the primary dimensions of DevSecOps-A multi-vocal literature. *Journal of Systems and Software*, 214. <https://doi.org/10.1016/j.jss.2024.112063>

D. S. Stepanov, M. M. Seniv

Lviv Polytechnic National University, Lviv, Ukraine

INTEGRATION OF PROTECTED INFRASTRUCTURE, CONTAINERIZATION AND DEVSECOPS TO INCREASE THE RELIABILITY OF WEB PORTALS

Features of the integration of secure infrastructure, containerization, and DevSecOps are presented in the paper, which enabled increasing the reliability of web portals, significantly improve their performance and impact resistance. In the course of research, we found that rapid development of numerous web projects requires ensuring their reliability, making the integration of Immutable Infrastructure, containerization, and DevSecOps essential. The research aims at analyzing and evaluating how the integration of these technologies impacts the availability and recoverability metrics of web portals. The research methodology included a comprehensive literature review, the development of a theoretical model, practical implementation, testing, and analysis of the results. The findings indicate that applying Immutable Infrastructure ensures environmental consistency and stability, significantly reducing the risk of unforeseen changes and system errors. Consequently, containerization, particularly with Docker and Kubernetes, simplifies the processes of deployment, scaling, and application isolation, facilitating rapid recovery from failures. Moreover, the integration of DevSecOps practices embeds security at every stage of the development lifecycle, ensuring timely detection and mitigation of vulnerabilities. These technologies collectively enhance overall performance and reduce web portal downtime, thereby decreasing the number of incidents and improving system response times. The research characterizes the patterns of Immutable Infrastructure adoption across different types of organizations, particularly small and medium-sized enterprises, enabling effective adaptation of these practices to specific business needs. The synergistic effect of combining these technologies contributes to achieving a high level of web portal reliability and security. Furthermore, this integration reduces operational costs and enhances the efficiency of development and operations teams. The implementation of containerization and Immutable Infrastructure significantly shortens the time required for updates and new application deployments, which is critical for rapidly evolving web portals. Additionally, DevSecOps practices improve collaboration between development, security, and operations teams, fostering the creation of more secure and resilient systems. The impact of web portal reliability on user satisfaction, a crucial metric for modern online services, was also assessed. The patterns of adopting these approaches in various industries were characterized, demonstrating the universality and effectiveness of these technologies across a wide range of applications. The research results confirm that the integration of Immutable Infrastructure, containerization, and DevSecOps is an effective approach to ensuring reliable and secure web portal operations, meeting the demands of the contemporary dynamic market. Some prospects for further research include optimizing the implementation processes of these technologies in different sectors to confirm their universality and effectiveness.

Keywords: environmental stability; system performance; deployment efficiency; recovery time.