УДК 004.415.3:004.738.5
DOI https://doi.org/10.32782/2663-5941/2025.3.2/49

Степанов Д.С.
Національний університет «Львівська політехніка»

Сенів М.М.
Національний університет «Львівська політехніка»

ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ НЕЗМІННОЇ ІНФРАСТРУКТУРИ ДЛЯ ЗАБЕЗПЕЧЕННЯ ПРОЗОРОГО МАСШТАБУВАННЯ БЕЗ ПРОСТОЇВ У ВЕБПОРТАЛАХ

Актуальність дослідження зумовлена необхідністю стабільного функціонування вебпорталів в умовах динамічного зростання навантаження та необхідності безперервного оновлення програмного забезпечення без втрати даних і простоїв. Застосування традиційних архітектурних підходів часто не забезпечує належного рівня надійності, оскільки масштабування призводить до виникнення конфліктів між версіями застосунків, несумісності компонентів та порушення цілісності даних під час оновлень. Відсутність уніфікованих рішень для оперативного відновлення системи у разі збою ускладнює підтримку високих показників доступності вебпорталів, особливо в контексті багатокомпонентних систем із розподіленою архітектурою.

Мета дослідження полягає в розробці методу забезпечення прозорого масштабування вебпорталів на основі інтеграції технологій контейнеризації та незмінної інфраструктури з акцентом на підвищення доступності та відновлюваності в межах концепції надійності програмного забезпечення.

Методологія дослідження охоплює порівняльний аналіз наявних підходів до контейнеризації, моделювання архітектурних схем для інтеграції незмінної інфраструктури у вебпортали, а також структурно-функціональний аналіз систем моніторингу та аварійного відновлення. Виявлено вплив контейнеризації на стабільність роботи компонентів вебпорталів у разі динамічного масштабування.

Наукова новизна дослідження полягає в розробці інтеграційної моделі, яка забезпечує автоматизоване оновлення компонентів без втрати доступності системи шляхом застосування стратегій «blue-green deployment» та «canary deployment».

Встановлено, що впровадження контейнеризації в поєднанні з незмінною інфраструктурою підвищує загальну стійкість системи до збоїв, мінімізуючи вплив людського фактора та забезпечуючи узгодженість версій компонентів. Доведено, що використання централізованих репозиторіїв значно скорочує час відновлення після аварії, а застосування контейнерних сесій із моніторингом забезпечує своєчасне реагування на потенційні загрози.

Перспективи подальших досліджень передбачають адаптацію розробленої моделі до архітектур із розподіленими сервісами та інтеграцію методів прогнозування пікових навантажень із застосуванням інструментів машинного навчання для своєчасного масштабування без зупинки основних сервісів.

Ключові слова: *незмінна інфраструктура, контейнеризація, масштабування без простоїв, надійність програмного забезпечення, відновлюваність, автоматизація оновлень.*

Постановка проблеми. Для сучасних вебпорталів критично важливо забезпечити стабільне функціонування в умовах зростання навантаження та динамічної користувацької активності, що зумовлює необхідність упровадження ефективних технологічних рішень для масштабування без простоїв. Традиційні методи масштабування часто виявляються недостатньо ефективними, оскільки вони можуть призводити до виникнення простоїв під час оновлення інфраструктури, а також до ускладнень у підтримці узгодженості даних між серверами. У відповідь на ці виклики все більшої популярності набувають технології незмінної інфраструктури, які дають можливість створювати повністю оновлені версії програмного забезпечення без втручання у поточні процеси та ризику для стабільності системи. Застосування таких технологій допомагає уникнути ризиків, пов'язаних із людським фактором та зменшити ймовірність виникнення помилок під час розгортання нових версій. Це забезпечує доступність до вебпорталів навіть у періоди активних оновлень та зростання навантаження. Водночас упровадження незмін-

ної інфраструктури потребує врахування особливостей архітектури вебпорталів, оптимізації процесів автоматичного розгортання та інтеграції засобів моніторингу для своєчасного виявлення потенційних загроз стабільності системи. Таким чином, актуальність дослідження полягає у визначенні ефективних моделей застосування технологій незмінної інфраструктури для прозорого масштабування вебпорталів, мінімізації часу простоїв та підвищення загальної стійкості системи до збоїв. З огляду на важливість завдань, що стосуються забезпечення високої доступності вебпорталів у сучасних умовах, доцільно зосередити увагу на розробці інноваційних підходів до реалізації масштабованих архітектур з акцентом на автоматизацію процесів розгортання та моніторингу.

Аналіз останніх досліджень і публікацій. Аналіз сучасних досліджень щодо застосування технологій незмінної інфраструктури для прозорого масштабування без простоїв у вебпорталах свідчить про наявність чотирьох основних змістових напрямів.

Перший напрям присвячено методологічним аспектам побудови незмінної інфраструктури та її впливу на показники надійності програмного забезпечення. Д. Степанов (D. Stepanov) [1] досліджує розвиток методології незмінної інфраструктури та аналізує її вплив на надійність роботи програмних систем, акцентуючи на доступності та швидкому відновленні після збоїв. У подальшій роботі Д. Степанов та М. Сенів [2] розширюють це питання, поєднуючи концепції захищеної інфраструктури, контейнеризації та DevSecOps, що комплексно підвищують стабільність і безпеку вебпорталів. М. Бафана (M. Bafana) та А. Абдулазіз (A. Abdulaziz) [3] розглядають практичні аспекти розгортання незмінної інфраструктури на платформі AWS, надаючи конкретні методологічні рекомендації для забезпечення безперебійної роботи. Н. Давидович (N. Davidovic), Н. Велькович (N. Veljkovic) та Л. Стоіменов (L. Stoimenov) [4] досліджують можливості блокчейн-технологій для гарантування незмінності державних електронних документів, підкреслюючи переваги таких рішень у публічних системах. С. Девіс (C. Davis) [5] описує загальні шаблони проєктування для побудови хмарних рішень із використанням незмінних архітектур, що забезпечують високу стійкість до змін та непередбачуваних навантажень. Перспективними напрямами досліджень є вивчення впливу обраних архітектурних рішень на швидкість масштабування та оцінка їхнього впливу на зниження витрат на обслуговування.

Другий напрям досліджень зосереджено на інтеграції технологій незмінної інфраструктури з блокчейн-рішеннями та розподіленими реєстрами для підвищення прозорості та безпеки масштабування. М. Шайк (M. Shaik) та Г. Бойя (G. Bojja) [6] вивчають інтеграцію блокчейну та сучасних протоколів управління доступом для посилення захисту цифрової інфраструктури. І. Тарханов (I. Tarkhanov), Д. Фомін-Нілов (D. Fomin-Nilov) та М. Фомін (M. Fomin) [7] підтверджують ефективність застосування блокчейну в наукових онлайн-виданнях для збереження незмінності цифрових даних. К. Кнутсен (K. Knutsen) та інші [8] демонструють застосування контейнеризованих систем із використанням блокчейн-технологій для безпечного обміну морськими даними. С. Бхуджел (S. Bhujel) та Й. Рахуламатхаван (Y. Rahulamathavan) [9] вивчають виклики масштабованості, прозорості та безпеки на прикладі NFT-маркетплейсів, наголошуючи на необхідності архітектурних удосконалень. Подальші дослідження доцільно спрямувати на аналіз технічних аспектів реалізації розподілених реєстрів у децентралізованих системах вебпорталів.

Третій напрям охоплює безпекові аспекти впровадження незмінної інфраструктури у вебпортали. Т. Аріф (T. Arif), Б. Джо (B. Jo) та Дж. Пак (J. Park) [10] аналізують комплексний підхід до збереження конфіденційності та безпеки даних у cloud-native середовищах, акцентуючи на необхідності інтеграції систем контролю доступу та верифікації даних. П. Чаухан (P. Chauhan) та П. Бансал (P. Bansal) [11] досліджують можливості незмінної архітектури для збереження цифрових доказів у криміналістичних системах, підкреслюючи її переваги для гарантування незмінності цифрових даних. Р. Нігам (R. Nigam), П. Азеведо де Аморім (P. Azevedo de Amorim) та А. Симпсон (A. Sampson) [12] пропонують модульні архітектурні підходи для апаратного забезпечення з використанням незмінних принципів проєктування. К. Індрасірі (K. Indrasiri) та С. Сухотаян (S. Suhothayan) [13] узагальнюють типові шаблони проєктування cloud-native додатків з незмінною архітектурою, акцентуючи на їхній масштабованості та стійкості до збоїв. Перспективними є дослідження з розробки комплексних моделей захисту даних, які поєднують елементи

незмінної інфраструктури з багаторівневими системами контролю доступу.

Четвертий напрям присвячено аналізу архітектурних рішень та моделей проєктування для підтримки прозорого масштабування незмінних інфраструктур. М. Перрі (M. Perry) [14] здійснює фундаментальний огляд архітектурних принципів побудови незмінних систем, зокрема принципів ідентифікації та автоматизації відновлюваних середовищ. С. Мохаммед (S. Mohammed) та А. Ралеску (A. Ralescu) [15] досліджують перспективи побудови нових архітектурних рішень для системи інтернету майбутнього, підкреслюючи значення незмінної інфраструктури для забезпечення стійкості даних та швидкого масштабування. Перспективи подальших досліджень охоплюють адаптивні моделі проєктування, які дають змогу оптимізувати незмінну інфраструктуру для динамічного масштабування залежно від зміни навантажень та вимог до безпеки.

Отже, аналіз свідчить про актуальність та перспективність технологій незмінної інфраструктури, що потребує подальших поглиблених досліджень щодо методологічних аспектів, інтеграційних можливостей із блокчейном, безпекових рішень та архітектурних моделей для забезпечення високої прозорості та надійності масштабування вебсистем.

Попри досягнення в упровадженні контейнеризації та незмінної інфраструктури, залишаються невирішеними питання інтеграції цих технологій у вебпортали з різними архітектурними моделями без порушення цілісності даних та втрати доступності. Відсутність узагальненої моделі, що поєднує логічне сегментування контейнерів, централізовані репозиторії образів та автоматизовані системи моніторингу, ускладнює масштабування систем без простоїв. Також недостатньо вивчено вплив незмінних контейнерних образів на відновлюваність після аварійних ситуацій, особливо в контексті динамічних навантажень, що є критичним для забезпечення безперервного функціонування вебпорталів. Крім того, потебують глибшого розгляду аспекти впровадження стратегій «blue-green deployment» та «canary deployment» для зниження ризиків під час оновлення системи та забезпечення узгодженості між компонентами. Невизначеність щодо оптимальних підходів до управління ресурсами під час пікових навантажень обмежує можливість реалізації повноцінної моделі резервного копіювання та автоматизованого відновлення. Запропоноване дослідження спрямоване на усунення цих прогалин шляхом розробки інтеграційної моделі, що враховує специфіку різних архітектур та забезпечує стабільність системи в разі збоїв.

Постановка завдання. Мета статті – розробити метод забезпечення прозорого масштабування вебпорталів без простоїв на основі технологій контейнеризації та незмінної інфраструктури з акцентом на підвищення показників доступності та відновлюваності в межах концепції надійності програмного забезпечення.

Завдання статті:

1. Визначити технологічні основи застосування контейнеризації для реалізації незмінної інфраструктури у вебпорталах, проаналізувавши їхній вплив на процеси масштабування та забезпечення узгодженості між компонентами системи.

2. Розробити інтеграційну модель контейнеризації та незмінної інфраструктури для забезпечення прозорого масштабування вебпорталів без простоїв та оцінити її вплив на показники доступності, відновлюваності та архітектурної стійкості системи.

3. Сформулювати рекомендації для адаптації запропонованої моделі до різних архітектур вебпорталів з акцентом на підвищення показників надійності в разі динамічних навантажень з урахуванням ризиків автоматизації процесів оновлення та сумісності з наявними компонентами.

Виклад основного матеріалу. Реалізація незмінної інфраструктури у вебпорталах за допомогою контейнеризації ґрунтується на концепції ізоляції застосунків у незалежних середовищах, що гарантує їхню стабільну роботу незалежно від змін у базовій інфраструктурі. За такого підходу створюються цілковито автономні версії програмного забезпечення, які містять усі необхідні компоненти для виконання застосунку, зокрема операційну систему, бібліотеки та залежності. Вебпортали, що впроваджують контейнеризацію, можуть масштабувати свої компоненти шляхом розгортання додаткових контейнерів без необхідності зміни вихідної архітектури. Це мінімізує час простоїв під час оновлення програмного забезпечення та гарантує його стабільну роботу при збільшенні навантаження. Водночас використання контейнеризації створює можливості для впровадження політик автоматичного відновлення контейнерів у разі збою, підвищуючи загальну надійність системи (табл. 1).

На практиці інтеграція контейнеризації для реалізації незмінної інфраструктури у вебпорталах забезпечує високу гнучкість масштабування

Таблиця 1

Технологічні компоненти контейнеризації для реалізації незмінної інфраструктури у вебпорталах

Технологічний компонент	Функціональна роль	Основні інструменти	Приклади застосування
Контейнерні платформи	Забезпечення ізоляції застосунків та їхньої незалежності від базової інфраструктури	Docker, Kubernetes, Podman	Масштабування вебсервісів шляхом додавання контейнерів під час пікових навантажень
Система оркестрації	Управління життєвим циклом контейнерів, автоматизація процесів розгортання та оновлення	Kubernetes, OpenShift, Docker Swarm	Автоматичний перезапуск контейнерів у разі збоїв
Автоматизоване розгортання	Автоматична генерація нових контейнерних образів на основі оновленого коду	Jenkins, GitLab CI/CD, Ansible	Розгортання нових версій застосунків без простоїв
Моніторинг та логування	Відстеження стану контейнерів та виявлення збоїв у реальному часі	Prometheus, Grafana, ELK Stack	Визначення вузьких місць у масштабованих компонентах системи
Репозиторії контейнерів	Зберігання образів контейнерів для швидкого розгортання нових версій	Docker Hub, Quay, Amazon ECR	Швидке оновлення вебпорталів шляхом заміни старих контейнерів новими версіями

Джерело: сформовано на основі [1, 2, 3, 5, 8]

та автоматизації процесів розгортання нових версій програмного забезпечення. Зокрема, застосування контейнерних платформ, таких як Docker або Kubernetes, дає змогу ізолювати застосунки у відокремлених середовищах, що усуває проблему конфліктів між компонентами різних версій та забезпечує узгодженість середовищ розгортання. Системи оркестрації відіграють вирішальну роль у забезпеченні безперервної роботи вебпорталів під час оновлення або збільшення навантаження, оскільки вони автоматично відстежують стан контейнерів та здійснюють перезапуск у разі збою. Це знижує ризики збоїв під час розгортання нових версій застосунків, оскільки усувається залежність від людського фактора та мінімізується вплив людських помилок.

Автоматизовані процеси розгортання на основі Jenkins або GitLab CI/CD систематично оновлюють застосунки без зупинки основних сервісів, що є критичним для динамічних вебпорталів зі значною кількістю активних користувачів. Водночас моніторингові інструменти на кшталт Prometheus та Grafana дають можливість оперативно виявляти аномалії в роботі контейнерів, що особливо важливо для вебпорталів із критичними бізнес-процесами [4]. Використання централізованих репозиторіїв для зберігання образів контейнерів спрощує управління версіями та сприяє швидкому відновленню системи в разі виникнення непередбачуваних збоїв. Такий підхід є одним з основних чинників підвищення надійності програмного забезпечення у вебпорталах, що функціонують в умовах динамічних навантажень та вимог до високої доступності.

Упровадження незмінної інфраструктури у вебпорталах істотно впливає на показники надійності програмного забезпечення, оскільки забезпечує стабільність його функціонування незалежно від оновлень та змін у системі. Основним принципом такої інфраструктури є створення постійних образів застосунків, що унеможливлює зміни в уже розгорнутих версіях програмного забезпечення. Це дає змогу уникнути непередбачуваних збоїв під час масштабування або розгортання нових компонентів, оскільки всі версії ідентичні та уніфіковані за своїм складом і конфігурацією. Незмінна інфраструктура забезпечує можливість оперативного відновлення системи в разі збою шляхом перезапуску контейнерів із раніше збереженими версіями застосунків. Водночас автоматизація процесів моніторингу сприяє соєчсному виявленню потенційних загроз для доступності системи на ранніх етапах, що знижує ризики простоїв і втрати даних (табл. 2).

Так, незмінна інфраструктура забезпечує високий рівень доступності вебпорталів через

Таблиця 2

Вплив упровадження незмінної інфраструктури на показники надійності вебпорталів

Показник надійності	Функціональна роль	Вплив незмінної інфраструктури	Приклади застосування
Доступність	Забезпечення безперервного функціонування системи під час оновлень	Зменшує ризик простоїв шляхом автоматичного перезапуску контейнерів	Актуально для фінансових платформ зі значною кількістю транзакцій
Відновлюваність	Оперативне відновлення системи після збою	Автоматичне створення резервних копій контейнерів та швидке їхнє відновлення	Застосовується в CRM-системах для збереження даних клієнтів
Взаємозв'язок компонентів	Узгодженість між версіями програмного забезпечення	Застосування ідентичних образів контейнерів у різних середовищах	Упроваджується в системах доставлення контенту для зменшення ризику несумісності
Виявлення аномалій	Моніторинг системи в режимі реального часу	Застосування інструментів моніторингу для ідентифікації потенційних збоїв	Сервіси аналітики для визначення аномалій у трафіку
Автоматизація процесів	Зниження впливу людського фактора на процес оновлення	Автоматичне розгортання контейнерів зі стабільними образами	Інтеграція в мікросервісні архітектури для динамічного масштабування

Джерело: сформовано на основі [1, 2, 4, 6, 7, 10]

можливість безперервного розгортання оновлень без зупинки основних сервісів. Наприклад, у банківських платформах або системах електронної комерції, де кожна секунда простою може призвести до втрати доходів, упровадження незмінної інфраструктури забезпечує стабільність системи під час оновлення програмного забезпечення. У разі виникнення збою система автоматично перезапускає контейнер з останньою стабільною версією, що мінімізує тривалість простою та зберігає цілісність даних.

Крім того, автоматизовані процеси моніторингу дають можливість оперативно виявляти відхилення в роботі системи та запобігати їхньому переростанню в критичні збої. Наприклад, застосування Prometheus у поєднанні з Grafana виявляє аномалії в роботі вебпорталів, аналізуючи метрики продуктивності контейнерів у режимі реального часу. Це забезпечує своєчасне реагування на потенційні загрози та дає змогу підтримувати високу доступність системи навіть за умов підвищеного навантаження.

Модель інтеграції контейнеризації та незмінної інфраструктури для забезпечення прозорого масштабування вебпорталів без простоїв спрямована на уніфікацію процесів розгортання та оновлення компонентів системи з акцентом на архітектурну стійкість та динамічне масштабування. Вона ґрунтується на концепції створення автономних контейнерів із заздалегідь налаштованими образами застосунків, які можуть бути швидко розгорнуті в разі збою або збільшення навантаження. Відмінність цієї моделі від наявних рішень полягає у використанні однотипних образів застосунків незалежно від середовища виконання, що усуває конфлікти версій та забезпечує оперативне відновлення системи без зупинки основних сервісів. Упровадження такої моделі дає змогу централізувати управління компонентами, контролювати їхню продуктивність у режимі реального часу та автоматизувати процеси резервного копіювання, що критично важливо для підтримки доступності та відновлюваності вебпорталів із високим рівнем динамічних навантажень (табл. 3).

У сучасних умовах така модель є особливо актуальною для динамічних вебпорталів, де критичне значення має забезпечення безперервного доступу до сервісів навіть у періоди пікового навантаження. Наприклад, у сфері електронної комерції інтеграція єдиного репозиторію образів дає змогу швидко замінювати застарілі версії застосунків без необхідності зупинки основних сервісів, зберігаючи цілісність транзакційних даних. Логічне сегментування контейнерів із застосуванням Istio або Linkerd дає змогу гнучко керувати мережевими запитами та локалізувати збої в певних компонентах системи, не впливаючи на роботу інших модулів. У разі аварійних ситуацій процес аварійного відновлення автоматично перемикає трафік на резервні контейнери, зберігаючи стабільність системи до завершення відновлювальних робіт.

Упровадження контейнеризації та незмінної інфраструктури у вебпорталах супроводжується низкою проблем, які можуть істотно вплинути на

Таблиця 3

Компоненти інтеграційної моделі для прозорого масштабування вебпорталів

Компонент моделі	Функціональна роль	Технологічна основа	Вплив на архітектурну стійкість
Єдиний репозиторій образів	Зберігання стандартизованих образів застосунків для швидкого розгортання	Amazon ECR, Docker Hub	Забезпечує одноманітність версій та швидке відновлення після збою
Автоматичне оновлення контейнерів	Динамічне оновлення компонентів без зупинки сервісів	Kubernetes, OpenShift	Знижує ризик втрати даних та запобігає простоям
Логічне сегментування контейнерів	Відокремлення основних функціональних модулів для незалежного масштабування	Istio, Linkerd	Підвищує стійкість до локальних збоїв у компонентах
Контейнерні сесії	Відстеження стану застосунків та моніторинг продуктивності	Prometheus, Grafana	Виявляє вузькі місця та прогнозує можливі збої
Процес аварійного відновлення	Автоматичний запуск стабільних версій контейнерів у разі збою	Jenkins, GitLab CI/CD	Забезпечує миттєве перемикання на резервні контейнери

Джерело: власна розробка авторів

стабільність та продуктивність системи. Однією з основних проблем є сумісність контейнерних рішень із наявними архітектурними моделями, особливо у випадках, коли вебпортали побудовані на монолітних системах або використовують застарілі компоненти. У таких умовах інтеграція контейнерних платформ може призвести до конфліктів між мікросервісами, різними версіями програмного забезпечення та базами даних [4]. Це ускладнює забезпечення узгодженості між компонентами системи, що може стати причиною збоїв або втрати даних під час масштабування або оновлення.

Ще однією проблемою є оптимізація ресурсів під час розгортання контейнерів, оскільки кожен контейнер потребує певного обсягу обчислювальних ресурсів та пам'яті [15]. Неправильне налаштування параметрів може призвести до перевантаження системи, надмірного споживання ресурсів або недостатньої потужності для обробки пікових навантажень [8]. Це особливо актуально для вебпорталів, які обслуговують велику кількість користувачів у режимі реального часу, оскільки будь-яка затримка або збій може негативно вплинути на користувацький досвід.

Автоматизація процесів оновлення контейнерів є ще одним критично важливим аспектом, який потенційно може створити ризики для стабільності вебпорталів [9]. У випадку неправильного налаштування системи CI/CD або некоректного тестування оновлених образів контейнерів можливе виникнення ситуацій, коли оновлення запускається одночасно для всіх компонентів системи. Це може призвести до одночасного зупинення кількох основних сервісів, що своєю чергою спричинить значний простій системи. У контексті незмінної інфраструктури проблема ускладнюється через необхідність зберігання попередніх версій контейнерів, що підвищує вимоги до сховищ даних та витрати на обчислювальні ресурси.

Іншою проблемою є моніторинг контейнерів та прогнозування їхньої продуктивності в реальному часі [11]. Упровадження контейнеризації потребує інтеграції додаткових систем моніторингу та логування, що може ускладнити аналіз великих обсягів даних та виявлення аномалій у роботі контейнерів [14]. У разі недостатнього рівня автоматизації моніторингу існує ризик несвоєчасного реагування на потенційні збої, що може спричинити неочікувані простої або втрату даних.

Також актуальним є питання безпеки контейнерів, оскільки використання незмінних образів застосунків передбачає збереження певних конфігурацій, доступів та змінних середовищ, які можуть стати вразливими для зовнішніх атак [6]. Уразливості можуть виникати у разі неправильного налаштування прав доступу до репозиторіїв контейнерів або відсутності перевірок автентичності образів. Це створює додаткові загрози для конфіденційності даних та стабільності вебпорталів, особливо якщо вони обробляють чутливу інформацію або фінансові транзакції.

Адаптація запропонованої моделі інтеграції контейнеризації та незмінної інфраструктури для вебпорталів із різними типами архітектур потребує врахування особливостей кожного типу сис-

теми, оптимізації процесів автоматизованого розгортання та налаштування моніторингових інструментів для забезпечення стабільної роботи під час динамічних навантажень. Для монолітних архітектур доцільно реалізувати поетапну міграцію компонентів до контейнерів із застосуванням стратегій «blue-green deployment» або «canary deployment», що дає змогу тестувати нові версії без ризику для основних сервісів. Застосування контейнеризації для критичних компонентів, таких як бази даних або API-сервіси, забезпечить їхню незалежність від інших модулів та підвищить стійкість до збоїв.

Для мікросервісних архітектур основна увага має бути зосереджена на впровадженні логічного сегментування контейнерів із чітко визначеними зонами відповідальності кожного модуля. Це сприятиме реалізації гнучкої оркестрації на основі Kubernetes або OpenShift, забезпечуючи автоматичне масштабування контейнерів у разі підвищеного навантаження. Рекомендовано налаштувати політики самооновлення контейнерів на основі метрик продуктивності, що мінімізує ризики збоїв та запобігає одночасному оновленню всіх модулів системи.

Для серверлес архітектур ефективність запропонованої моделі досягається шляхом інтеграції подієвих тригерів для активації нових контейнерів у разі різкого збільшення кількості запитів. Застосування автоматичних контейнерних сесій із прогнозуванням пікових навантажень забезпечує безперервність обслуговування навіть за умов раптового збільшення обчислювальних операцій. Доцільно впровадити моніторингові системи з підтримкою машинного навчання для прогнозування пікових навантажень та своєчасного масштабування контейнерів без зупинки основних сервісів.

Для архітектур із розподіленими сервісами доцільно застосовувати концепцію логічних кластерів, які об’єднують функціонально споріднені контейнери для спрощення управління та моніторингу. Такий підхід дає змогу централізовано контролювати стан контейнерів, прогнозувати можливі збої та реалізовувати автоматизовані сценарії відновлення з використанням резервних образів застосунків. Інтеграція політик аварійного відновлення на основі реплікації даних у різних середовищах підвищить відновлюваність системи в разі катастрофічних збоїв або втрати даних.

Таким чином, адаптація моделі для вебпорталів із різними типами архітектур має ґрунтуватися на диференційованому підході до впровадження контейнеризації, сегментування функціональних модулів та інтеграції інструментів моніторингу й аварійного відновлення. Оптимізація політик масштабування та автоматизація оновлень забезпечать стабільність системи навіть під час пікових навантажень, сприяючи підвищенню показників доступності та відновлюваності.

Висновки. Дослідження підтвердило, що інтеграція контейнеризації та незмінної інфраструктури у вебпорталах підвищує надійність програмного забезпечення завдяки ізоляції компонентів та автоматизації процесів розгортання без простоїв. Запропонована модель містить стандартизовані контейнерні образи з логічним сегментуванням для мінімізації впливу локальних збоїв та оперативного відновлення системи. Виявлено проблеми сумісності з монолітними архітектурами та труднощі з управлінням ресурсами під час пікових навантажень, що потребує впровадження централізованих репозиторіїв образів та систем моніторингу. Рекомендовано використовувати поетапну міграцію компонентів із застосуванням стратегій «blue-green deployment» або «canary deployment», що забезпечить безперервність оновлень без зупинки основних сервісів. Подальші дослідження доцільно зосередити на адаптації запропонованої моделі до архітектур із розподіленими сервісами та автоматизації процесів резервного копіювання із застосуванням інструментів CI/CD.

Список літератури:

1. Stepanov D. Research on Development Methodology Utilizing Immutable Infrastructure and Its Impact on Software Reliability. *18th International Conference on Computer Sciences and Information Technologies (CSIT)*. 2023. P. 119–131. DOI: https://doi.org/10.1109/CSIT61576.2023.10324201

2. Степанов Д., Сенів М. Інтеграція захищеної інфраструктури, контейнеризації та DevSecOps для підвищення надійності роботи вебпорталів. *Науковий вісник НЛТУ України*. 2024. Вип. 34, № 5. С. 144–150. DOI: https://doi.org/10.36930/40340519

3. Bafana M., Abdulaziz A. Immutable Infrastructure in Practice: A Comprehensive Guide to AWS Deployment. *Asian American Research Letters Journal*. 2024. Vol. 1, № 1. P. 1–16. URL: https://aarlj.com/index.php/AARLJ/article/download/6/3 (date of access: 09.05.2025).

4. Davidovic N., Veljkovic N., Stoimenov L. Assuring immutability of public e-government documents using blockchain infrastructure. *Proceedings of the 10th International Conference on Information Society and Technology (ICIST 2020)*. 2020. P. 176–180. URL: https://surl.li/ngfjjy (date of access: 09.05.2025).

5. Davis C. *The Art of Immutable Architecture*. Apress: New York, NY, USA. 2020. URL: https://link.springer.com/book/10.1007/979-8-8688-0288-1 (date of access: 09.05.2025).
6. Shaik M., Bojja G. Advanced Identity Access Management and Blockchain Integration: Techniques, Protocols, and Real-World Applications for Enhancing Security, Privacy, and Scalability in Modern Digital Infrastructures. *Libertatem Media Private Limited*. 2022. URL: https://surl.li/uvikti (date of access: 09.05.2025).
7. Tarkhanov I., Fomin-Nilov D., Fomin M. Application of public blockchain to control the immutability of data in online scientific periodicals. *Library Hi Tech*. 2019. Vol. 37, № 4. P. 829–844. DOI: https://doi.org/10.1108/LHT-12-2018-0186
8. Knutsen K. E., et al. Containerized immutable maritime data sharing utilizing Distributed Ledger Technologies. *Journal of Physics: Conference Series*. 2022. Vol. 2311, № 1. Article 012006. DOI: https://doi.org/10.1088/1742-6596/2311/1/012006
9. Bhujel S., Rahulamathavan Y. A survey: Security, transparency, and scalability issues of NFT's and its marketplaces. *Sensors*. 2022. Vol. 22, № 22. Article 8833. DOI: https://doi.org/10.3390/s22228833
10. Arif T., Jo B., Park J. A Comprehensive Survey of Privacy-Enhancing and Trust-Centric Cloud-Native Security Techniques Against Cyber Threats. *Sensors*. 2025. Vol. 25, № 8. Article 2350. DOI: https://doi.org/10.3390/s25082350
11. Chauhan P., Bansal P. Enhancing Trust and Immutability in Cloud Forensics. *Advances in Intelligent Systems and Computing*. 2021. Vol. 1270. P. 891–901. DOI: https://doi.org/10.1007/978-981-15-8289-9_74
12. Nigam R., Azevedo de Amorim P. H., Sampson A. Modular hardware design with timeline types. *Proceedings of the ACM on Programming Languages*. 2023. Vol. 7, № PLDI. P. 343–367. DOI: https://doi.org/10.5281/zenodo.7709916
13. Indrasiri K., Suhothayan S. *Design Patterns for Cloud Native Applications*. O'Reilly Media, Inc. 2021. URL: https://surl.li/ibnkho (date of access: 09.05.2025).
14. Perry M. *Cloud Native Patterns: Designing Change-Tolerant Software*. Simon and Schuster. 2019. URL: https://surl.li/ewjluf (date of access: 09.05.2025).
15. Mohammed S. A., Ralescu A. L. Future internet architectures on an emerging scale – a systematic review. *Future Internet*. 2023. Vol. 15, № 5. Article 166. DOI: https://doi.org/10.3390/fi15050166

Stepanov D.S., Seniv M.M. APPLICATION OF IMMUTABLE INFRASTRUCTURE TECHNOLOGIES TO ENSURE TRANSPARENT SCALING WITHOUT DOWNTIME IN WEB PORTALS

The relevance of the study is determined by the need for stable functioning of web portals in conditions of dynamic load growth and the need for continuous software updates without data loss and downtime. The use of traditional architectural approaches often does not provide an adequate level of reliability, as scaling leads to conflicts between application versions, component incompatibility, and data integrity violations during updates. The lack of unified solutions for rapid system recovery in the event of a failure makes it difficult to maintain high availability of web portals, especially in the context of multi-component systems with a distributed architecture.

The goal of this research is to develop a method for transparent scaling of web portals based on the integration of containerization and immutable infrastructure technologies, with a focus on improving availability and recoverability within the concept of software reliability.

The research methodology includes a comparative analysis of existing approaches to containerization, modeling of architectural schemes for integrating immutable infrastructure into web portals, as well as a structural and functional analysis of monitoring and disaster recovery systems. The impact of containerization on the stability of web portal components in the case of dynamic scaling has been revealed.

The scientific novelty of the research lies in the development of an integration model that provides automated component updates without losing system availability by applying blue-green deployment and canary deployment strategies.

It has been established that the introduction of containerization in combination with an unchanging infrastructure increases the overall stability of the system to failures, minimizing the influence of the human factor and ensuring the consistency of component versions. It has been proven that the use of centralized repositories significantly reduces recovery time after a failure, and the use of container sessions with monitoring ensures timely response to potential threats.

Prospects for further research include adapting the developed model to distributed service architectures and integrating peak load prediction methods using machine learning tools for timely scaling without stopping core services.

Key words: *immutable infrastructure, containerization, scaling without downtime, software reliability, recoverability, update automation.*