

<https://doi.org/10.31891/2307-5732-2026-365-16>

УДК 044.9

**ПІЦУН ОЛЕГ**

Західноукраїнський національний університет

<https://orcid.org/0000-0003-0280-8786>

e-mail: [o.pitsun@wunu.edu.ua](mailto:o.pitsun@wunu.edu.ua)

**ВОЛИНСЬКИЙ ОРЕСТ**

Надвірнянський фаховий коледж Національного транспортного університету

<https://orcid.org/0000-0002-1409-9393>

e-mail: [volynskyio@nfentu.ukr.education](mailto:volynskyio@nfentu.ukr.education)

## ВПЛИВ СТАБІЛЬНОСТІ DEVOPS НА МАСШТАБОВАНІСТЬ РОЗПОДІЛЕНИХ СИСТЕМ

*Активне зростання хмарних обчислень та розподілених комп'ютерних систем призвело до впровадження мікросервісних архітектур та парадигм хмарної розробки. Враховуючи різноманітність сучасних систем, які поєднують декілька програмних технологій, машинне навчання, елементи IoT виникла необхідність у розробці гнучких масштабованих систем. Основними підходами з допомогою яких вирішується питання масштабованості полягає у впровадженні етапів балансування навантажень, оркестрацію контейнерів та стратегії виділення додаткових ресурсів. Існуючі рішення ґрунтуються на розробці спеціалізованих архітектур, які не завжди включають достатню кількість елементів DevOps підходів, що у свою чергу не дозволяє отримати максимальну продуктивність. Незважаючи на практичні аспекти, кількісний зв'язок між застосуванням DevOps та масштабованістю розподілених систем залишається недостатньо дослідженим. У даній статті введено кількісний індекс стабільності DevOps та досліджено його вплив на показники масштабованості в розподілених мікросервісних системах.*

**Ключові слова:** мікросервіси, архітектура програмного забезпечення, DevOps, MLOps.

**PITSUN OLEH**

West Ukrainian National University

**VOLYNSKYI OREST**

Nadvirna Vocational College of the National Transport University

## DEVOPS STABILITY IMPACT ON SCALABILITY OF DISTRIBUTED SYSTEMS

*The active growth of cloud computing and distributed computer systems has led to the introduction of microservice architectures and cloud development paradigms. Given the diversity of modern systems that combine several software technologies, machine learning, and IoT elements, there is a need to develop flexible scalable systems. The main approaches to address the scalability issue are the implementation of load balancing stages, container orchestration, and strategies for allocating additional resources. Modern software ecosystems increasingly integrate heterogeneous technologies, including machine learning components, Internet of Things devices, and large-scale data processing services, which significantly increases architectural complexity and operational demands. Under these conditions, ensuring system scalability becomes a critical requirement for maintaining performance, reliability, and service continuity. Although DevOps is widely recognized for improving software delivery efficiency and operational collaboration, its quantitative impact on scalability in distributed environments remains insufficiently investigated. Existing solutions are based on the development of specialized architectures that do not always include a sufficient number of DevOps elements, which in turn does not allow for maximum performance. Despite the practical aspect, the quantitative relationship between the use of DevOps and the scalability of distributed systems remains insufficiently studied. This article introduces the quantitative DevOps Stability Index and examines its impact on scalability indicators in distributed microservice systems. Using this index, the study investigates the relationship between DevOps adoption and scalability indicators in distributed microservice systems.*

**Keywords:** microservices, software architecture, DevOps, MLOps.

Стаття надійшла до редакції / Received 02.03.2026

Прийнята до друку / Accepted 18.03.2026

Опубліковано / Published 28.05.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Піцун Олег, Волинський Орест

### Постановка проблеми

Розробка програмного забезпечення включає в себе широкий спектр технологій і високий рівень знань з точки зору проектування мікросервісів та використання хмарних технологій. Тому, важливою складовою розробки і забезпечення стабільної роботи систем є використання DevOps та MLOps практик для інтеграції машинного навчання. DevOps є парадигмою, спрямованою на інтеграцію розробки програмного забезпечення з використанням автоматизації, безперервної інтеграції та доставки, інфраструктури як коду. Аналіз досліджень в даній галузі показав, що практика DevOps покращує частоту розгортання, надійність та час відновлення. Автоматизовані конвеєри розгортання програмного забезпечення, механізми спостереження та швидке реагування на інциденти можуть впливати на ефективність масштабування, стабільність затримки та стійкість системи в умовах високого навантаження.

Використання складних модулів з елементами машинного навчання чи комп'ютерного зору значно ускладнює архітектуру системи. Розширення функціоналу без врахування DevOps практик сприяє збільшенню часу виконання задач, використанню більшої кількості ресурсів, збільшенню вартості розробки, функціонування та супроводу.

Метою даної статті є розробка універсального кількісного параметру оцінювання достатності застосування DevOps-практик та дослідження її впливу на показники масштабованості розподілених

мікросервісних систем під змінним навантаженням.

### Аналіз досліджень та публікацій

Для формування задач та актуального стану справ у галузі розробки програмного забезпечення на основі масштабованості та мікросервісів здійснено аналіз останніх досліджень в журналах, що індексуються в міжнародних наукометричних базах.

У роботі [1] наведено результати опитування щодо застосування практик забезпечення продуктивності у DevOps. Встановлено низький рівень використання моделей продуктивності. Автори пояснюють це відносною незрілістю DevOps та недостатньою інтеграцією оцінки продуктивності у процеси автоматизації. У систематичному огляді [2] проаналізовано сучасний стан використання метрик продуктивності в DevOps та інженерії надійності програмного забезпечення. У роботі [3] зазначається, що регресії продуктивності є поширеним явищем у великих програмних системах, але проблема полягає у тому що існуючі підходи до їх виявлення недостатньо узгоджуються з процесами DevOps. Автори у роботі акцентують увагу на розробленні емпірично обґрунтованих та автоматизованих методів виявлення регресій продуктивності. У дослідженні [4] представлено результати багатоджерельного літературного огляду, присвяченого метрикам DevOps, у межах якого проаналізовано 139 наукових публікацій. Особливу увагу приділено чотирьом найбільш визнаним метрикам MTTR, MLT, Deployment Frequency та Change Failure Rate. Використання цих метрик покращує результати розроблення і експлуатації програмного забезпечення.

У роботі [5] автори досліджували і встановили що впровадження відповідних методологій і інструментів, використання таких технологій, як Jenkins, GitLab, Kubernetes і Docker, сприяє оптимізації процесів розроблення і розгортання програмного забезпечення. У роботі [6] розглядаються підходи та методи підвищення масштабованості DevOps-процесів на ранніх етапах розроблення програмного забезпечення, зосереджуючись на практиках побудови та оптимізації DevOps-конвеєрів у хмарних середовищах моделі Software-as-a-Service (SaaS). У роботі [7] авторами проведено аналіз впливу характеристик DevOps на якість програмного забезпечення. Основними критеріями в дослідженні були безперервна доставка коду, швидкі цикли зворотного зв'язку та автоматизації, а також підкреслюється важливість розроблення метрик продуктивності й оцінювання ключових факторів якості. У роботі [8] розглядаються високі вимоги до обміну даними та файлами у сфері підключених автомобілів між транспортними засобами та backend-сервісами.

У роботі [9] проаналізовано мультисервісних архітектур, які є ефективним підходом для підвищення масштабованості та зниження витрат у сучасних корпоративних IT-середовищах. Використання кількох хмарних провайдерів дозволяє розподіляти навантаження, підвищувати відмовостійкість і забезпечувати кращу продуктивність систем. У дослідженні [10] проаналізовано використання serverless-архітектур порівняно з традиційними серверними хмарними моделями дозволяє суттєво підвищити масштабованість застосунків і зменшити витрати. На основі аналізу літератури та порівняльного аналізу продуктивності встановлено, що serverless-підхід забезпечує автоматичне масштабування відповідно до навантаження, ефективніше використання ресурсів і покращений користувацький досвід із дотриманням рівнів сервісу. Перспективним напрямом визначено розвиток гібридних архітектур, що поєднують мікросервіси, serverless-підходи та подієво-орієнтовані моделі. Загалом дослідження [11] формує комплексну основу для подальшого розвитку хмарних обчислень через міждисциплінарні інновації. Дослідження присвячене оцінюванню якості наукових досліджень у галузі програмної інженерії, яку розглянуто як сукупність критеріїв, що включають методологічну строгість, практичну релевантність та дотримання етичних норм.

Аналіз MLOps підходів до розробки програмних систем для опрацювання задач на основі машинного навчання та комп'ютерного зору розглянуто у роботі [12]. Автори виділяють основні компоненти та принципи взаємодії мікросервісів у хмарного середовищі.

У підсумку, напрям досліджень в даній галузі спирається на розробку і аналіз технологій що дозволяють масштабувати інфраструктуру. Однак, постає невирішена проблема щодо об'єктивного оцінювання ступеню застосування DevOps практик для максимальної ефективності роботи програмних систем. Базуючись на здійсненому аналізі реалізовано розробку універсального механізму для оцінки стабільності роботи DevOps систем та можливості їх покращення.

### Формулювання цілей статті

Метою даної статті є розробка кількісного параметру оцінювання якості застосування DevOps-практик, що дозволить виявити проблемні блоки при розробці і функціонуванні масштабованих програмних систем. Для реалізації мети необхідно вирішити такі задачі:

- розробити кількісну оцінку рівня застосування DevOps-практик шляхом формування інтегрального індексу DevOps Stability Index на основі ключових показників автоматизації, CI/CD та операційної надійності;
- визначити систему метрик масштабованості розподілених систем, що включає показники продуктивності, затримки, ефективності масштабування та стабільності роботи під навантаженням;
- здійснити порівняльний аналіз отриманих результатів з метою встановлення залежності між рівнем DevOps maturity та показниками масштабованості системи.

### Виклад основного матеріалу

Досліджувані платформи охоплюють як легкі, так і корпоративні системи, результати можуть не узагальнюватися безпосередньо на всі сфери розподілених систем або галузеві сектори.

## Запропонований індекс якості застосування DevOps практик

Ступінь застосування практик DevOps та надійності програмної системи обчислюється на основі запропонованої DevOps Stability Index (DSI) у формулі:

$$DSI = \sum_{i=1}^n w_i x_i$$

де  $w_i$  - ваговий коефіцієнт, що визначає вплив метрики на кінцеве рішення;

$x_i$  - метрика, що визначає наявність і ступінь застосування DevOps практики;

$n$  – кількість метрик.

В сумі вагові коефіцієнти дорівнюють 1.

У базовому випадку виділено такі основні параметри для оцінки ступеню застосування DevOps для програмної системи:

CI - рівень автоматизації збірки (CI/CD);

D - частота деплойментів;

T – покриття автоматизованими тестами;

O - Observability;

R - Можливість відновлення;

I – Інфраструктура як код.

Приклад розрахунку DSI для оцінки програмної системи з елементами машинного навчання (класифікація об'єктів) наведено у формулі:

$$DSI_{ML} = 0.2C + 0.15D + 0.2T + 0.2O + 0.15R + 0.1I$$

Розподіл коефіцієнтів залежить від специфіки програмного забезпечення, наприклад програмне забезпечення для опрацювання зображень може мати інші коефіцієнти ніж для штучного інтелекту.

Узагальнену схему роботи індексу стабільності DevOps наведено на рисунку 1.

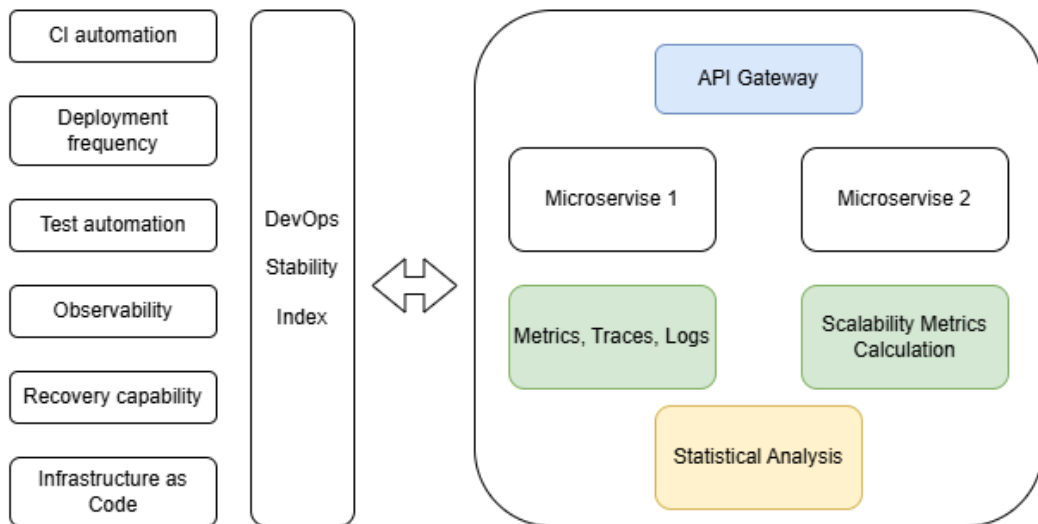


Рис. 1. Узагальнена схема розрахунку індексу стабільності DevOps

Індекс стабільності DevOps - це ядро системи. API Gateway – це єдина точка входу, яка керує трафіком до мікросервісів. Мікросервіси є компонентами додатку, стан яких аналізується. У ролі мікросервісів виступають окремі додатки для опрацювання вхідних даних. Важливою частиною цього блоку є підсистема збору даних, яка акумулює метрики, трасування та логи, а також виконує розрахунок показників масштабованості. На основі цих зібраних технічних даних фінальний аналітичний блок проводить статистичну обробку, що дозволяє перетворити сиру системну інформацію на індикатори для розрахунку загального індексу стабільності DevOps.

Масштабованість моделювалася за допомогою складних показників продуктивності (пропускна здатність, затримка та доступність), можуть не повністю відображати характеристики робочого навантаження, специфічні для предметної області. Для обчислення показника масштабованості використано формулу:

$$S = \alpha TH + \beta LAT + \gamma AV$$

де  $TH$  - пропускна здатність

$LAT$  - затримка

$AV$  - доступність

Типові вагові коефіцієнти:  $\alpha = 0.4$ ;  $\beta = 0.3$ ;  $\gamma = 0.3$ .

Порівняльна таблиця програмних систем для опрацювання даних засобами глибокого навчання за критерієм стабільності DevOps наведено у таблиці 1.

**Порівняльна таблиця програмних систем для опрацювання даних засобами глибинного навчання за критерієм стабільності DevOps**

|                               | CI/CD                | Інфраструктура як код. (Terraform) | Автоматизоване тестування | Частота розгортання | Коефіцієнт невдалих змін |
|-------------------------------|----------------------|------------------------------------|---------------------------|---------------------|--------------------------|
| Nyckel                        | Обмежена             | -                                  | базова                    | Ручний режим        | Обмежений моніторинг     |
| Ximilar                       | API                  | -                                  | Валідація датасету        | Середня             | Ручне тестування         |
| Roboflow                      | GitHub Actions, API  | частково                           | Тестування датасету       | Висока              | Зовнішній моніторинг     |
| Hasty                         | API                  | -                                  | Валідація датасету        | Середня             | Обмежено                 |
| Levity                        | Automation workflows | -                                  | базова                    | Низька              | Мінімальний контроль     |
| Google Vertex AI              | Cloud Build / GitOps | GCP Terraform                      | Тестування моделей        | Висока              | SLO + моніторинг         |
| AWS Rekognition Custom Labels | CodePipeline / CI    | AWS Terraform                      | Автоматична оцінка        | Висока              | Метрики CloudWatch       |
| HIAMS                         | GitHub Actions       | -                                  | базова                    | Середня             | Середня                  |

Хмарні enterprise-платформи демонструють найвищий рівень DevOps-стабільності завдяки нативній підтримці CI/CD, Infrastructure-as-Code та observability-інструментів, що безпосередньо знижує коефіцієнт невдалих змін і підвищує частоту розгортання. Легкі III-платформи орієнтовані переважно на швидке прототипування та мають обмежену інтеграцію з DevOps-процесами, що ускладнює масштабування у production-середовищах.

Результати дослідження підтверджують, що зрілість DevOps-практик є одним із ключових чинників забезпечення масштабованості сучасних розподілених систем. Отримані емпіричні результати демонструють, що інтеграція CI/CD, Infrastructure-as-Code та автоматизованого тестування суттєво знижує операційні ризики та підвищує стабільність процесів розгортання.

#### **Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі**

У даній статті здійснено аналіз впливу рівня стабільності DevOps-практик на масштабованість сучасних розподілених програмних систем. Для кількісного оцінювання запропоновано показник DevOps Stability Index, який об'єднує ключові характеристики DevOps.

Здійснено порівняльний аналіз сучасних платформ для опрацювання даних на основі глибинного навчання з використанням хмарних технологій за 5 параметрами, що дозволить виділити переваги та недоліки існуючих систем з точки зору DevOps.

Для оцінки масштабованості було визначено складений показник масштабованості шляхом поєднання нормалізованих показників пропускну здатності, затримки та доступності. Цей складний показник відображає як ефективність продуктивності, так і операційну стабільність.

Легкі III-платформами, орієнтованими на швидке прототипування, та enterprise-рішеннями, які забезпечують повноцінну інтеграцію DevOps-інструментів і, відповідно, вищу масштабованість у production-середовищах.

Практична цінність дослідження полягає у застосуванні запропонованого індексу DSI як інструменту оцінювання технологічних платформ, планування DevOps-трансформації та прийняття архітектурних рішень при проєктуванні високонавантажених систем.

Подальші дослідження будуть спрямовані на збільшення кількості метрик observability, використання машинного навчання для прогнозування зниження продуктивності, а також емпіричну валідацію моделі на великих промислових датасетах і реальних DevOps-pipeline середовищах.

#### **Література**

1. Cor-Paul Bezemer, Simon Eismann, Vincenzo Ferme, Johannes Grohmann, Robert Heinrich, Pooyan Jamshidi, Weiyi Shang, André van Hoorn, Monica Villavicencio, Jürgen Walter, and Felix Willnecker. 2019. How is Performance Addressed in DevOps? In Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering (ICPE '19). Association for Computing Machinery, New York, NY, USA, 45–50. <https://doi.org/10.1145/3297663.3309672>

2. Daraojimba, Andrew Ifesinachi, Denis Kisina, Oluwasanmi Segun Adanigbo, Bright Chibunna Ubanadu, Nneka Adaobi Ochuba and Toluwase Peter Gbenle. "Systematic Review of Key Performance Metrics in Modern DevOps and Software Reliability Engineering." *International Journal of Future Engineering Innovations* (2024) - DOI:10.54660/ijfei.2024.1.1.101-107
3. Jinfu Chen. 2020. Performance regression detection in DevOps. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings (ICSE '20)*. Association for Computing Machinery, New York, NY, USA, 206–209. <https://doi.org/10.1145/3377812.3381386>
4. Ricardo Amaro, Rúben Pereira, and Miguel Mira da Silva. 2024. DevOps Metrics and KPIs: A Multivocal Literature Review. *ACM Comput. Surv.* 56, 9, Article 231 (September 2024), 41 pages. <https://doi.org/10.1145/3652508>
5. Fatiha El Aouni, Karima Moumane, Ali Idri, Mehdi Najib, Saeed Ullah Jan, A systematic literature review on Agile, Cloud, and DevOps integration: Challenges, benefits, *Information and Software Technology*, Volume 177, 2025, 107569, ISSN 0950-5849, <https://doi.org/10.1016/j.infsof.2024.107569>
6. Jaju, 'Maximizing DevOps Scalability in Complex Software Systems : Maximizing DevOps Scalability in Complex Software Systems', Dissertation, 2023. <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1787653&dswid=8732>
7. Jaju, I. (2023). Maximizing DevOps Scalability in Complex Software Systems : Maximizing DevOps Scalability in Complex Software Systems (Dissertation). Retrieved from <https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-509039>
8. Schneider, Tobias, and A. Wolfsmantel. "Achieving Cloud Scalability with Microservices and DevOps in the Connected Car Domain." In *Software Engineering (Workshops)*, pp. 138-141. 2016. <https://ceur-ws.org/Vol-1559/paper16.pdf>
9. Johnson, Omoniyi Babatunde, Jeremiah Olamijuwon, Emmanuel Cadet, Olajide Soji Osundare, and Zein Samira. "Designing multi-cloud architecture models for enterprise scalability and cost reduction." *Open Access Research Journal of Engineering and Technology* 7, no. 02 (2024): 101-113. <https://doi.org/10.53022/oarjet.2024.7.2.0061>
10. Nisher Ahmed. Server lessArchitecture: Optimizing Application Scalability and Cost Efficiency in Cloud Computing - BULLET: Jurnal Multidisiplin Ilmu Volume 1, No. 06, December (2022)
11. Arimondo Scrivano. Innovative Approaches in Cloud Computing: Balancing Efficiency, Scalability, and Sustainability. *TechRxiv*. June 19, 2025. DOI: 10.36227/techrxiv.174970502.20312114/v2
12. ПІЦУН, О., БЕРЕЗЬКИЙ, М., & ШИМЧУК, М. (2025). MLOPS ПІДХІД ДЛЯ АВТОМАТИЧНОЇ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ. *Herald of Khmelnytskyi National University. Technical Sciences*, 355(4), 463-469. <https://doi.org/10.31891/2307-5732-2025-355-66>

## References

1. Cor-Paul Bezemer, Simon Eismann, Vincenzo Ferme, Johannes Grohmann, Robert Heinrich, Pooyan Jamshidi, Weiyi Shang, André van Hoom, Monica Villavicencio, Jürgen Walter, and Felix Willnecker. 2019. How is Performance Addressed in DevOps? In *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering (ICPE '19)*. Association for Computing Machinery, New York, NY, USA, 45–50. <https://doi.org/10.1145/3297663.3309672>
2. Daraojimba, Andrew Ifesinachi, Denis Kisina, Oluwasanmi Segun Adanigbo, Bright Chibunna Ubanadu, Nneka Adaobi Ochuba and Toluwase Peter Gbenle. "Systematic Review of Key Performance Metrics in Modern DevOps and Software Reliability Engineering." *International Journal of Future Engineering Innovations* (2024) - DOI:10.54660/ijfei.2024.1.1.101-107
3. Jinfu Chen. 2020. Performance regression detection in DevOps. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings (ICSE '20)*. Association for Computing Machinery, New York, NY, USA, 206–209. <https://doi.org/10.1145/3377812.3381386>
4. Ricardo Amaro, Rúben Pereira, and Miguel Mira da Silva. 2024. DevOps Metrics and KPIs: A Multivocal Literature Review. *ACM Comput. Surv.* 56, 9, Article 231 (September 2024), 41 pages. <https://doi.org/10.1145/3652508>
5. Fatiha El Aouni, Karima Moumane, Ali Idri, Mehdi Najib, Saeed Ullah Jan, A systematic literature review on Agile, Cloud, and DevOps integration: Challenges, benefits, *Information and Software Technology*, Volume 177, 2025, 107569, ISSN 0950-5849, <https://doi.org/10.1016/j.infsof.2024.107569>
6. Jaju, 'Maximizing DevOps Scalability in Complex Software Systems : Maximizing DevOps Scalability in Complex Software Systems', Dissertation, 2023. <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1787653&dswid=8732>
7. Jaju, I. (2023). Maximizing DevOps Scalability in Complex Software Systems : Maximizing DevOps Scalability in Complex Software Systems (Dissertation). Retrieved from <https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-509039>
8. Schneider, Tobias, and A. Wolfsmantel. "Achieving Cloud Scalability with Microservices and DevOps in the Connected Car Domain." In *Software Engineering (Workshops)*, pp. 138-141. 2016. <https://ceur-ws.org/Vol-1559/paper16.pdf>
9. Johnson, Omoniyi Babatunde, Jeremiah Olamijuwon, Emmanuel Cadet, Olajide Soji Osundare, and Zein Samira. "Designing multi-cloud architecture models for enterprise scalability and cost reduction." *Open Access Research Journal of Engineering and Technology* 7, no. 02 (2024): 101-113. <https://doi.org/10.53022/oarjet.2024.7.2.0061>
10. Nisher Ahmed. Server lessArchitecture: Optimizing Application Scalability and Cost Efficiency in Cloud Computing - BULLET: Jurnal Multidisiplin Ilmu Volume 1, No. 06, December (2022)
11. Arimondo Scrivano. Innovative Approaches in Cloud Computing: Balancing Efficiency, Scalability, and Sustainability. *TechRxiv*. June 19, 2025. DOI: 10.36227/techrxiv.174970502.20312114/v2
12. ПІЦУН, О., БЕРЕЗЬКИЙ, М., & ШИМЧУК, М. (2025). MLOPS APPROACH FOR AUTOMATIC IMAGE SEGMENTATION. *Herald of Khmelnytskyi National University. Technical Sciences*, 355(4), 463-469. <https://doi.org/10.31891/2307-5732-2025-355-66>