

УДК 004.4:004.7

ДОСЛІДЖЕННЯ МЕТОДІВ ОПТИМІЗАЦІЇ ПРОДУКТИВНОСТІ ХМАРНИХ ВЕБ-ДОДАТКІВ КОНТЕЙНЕРИЗОВАНИХ У KUBERNETES

Ворона Д.О.

e-mail: dmytro.vorona1@nure.ua

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

This work is devoted to the study of methods for optimizing the performance of cloud-based web applications containerized in Kubernetes. Modern cloud applications require high scalability, reliability, and efficiency, making optimization an essential aspect of development. During the research, existing optimization approaches were analyzed, monolithic web application was implemented and transformed into a microservices architecture. The study involved testing different scaling strategies, resource allocation techniques, and monitoring tools to measure the effectiveness of optimization methods.

Сучасні хмарні веб-додатки потребують високої продуктивності, масштабованості та надійності. Kubernetes став основною платформою для оркестрації контейнеризованих додатків, але його ефективне використання вимагає оптимізації ресурсів. Саме тому актуальним є дослідження методів оптимізації продуктивності веб-додатків у Kubernetes.

Метою дослідження є аналіз існуючих методів оптимізації продуктивності хмарних веб-додатків у Kubernetes, а також розробка та впровадження підходів для підвищення ефективності використання ресурсів. Для досягнення цієї мети було проведено аналіз предметної галузі, огляд літератури та практичних рішень, розроблено та протестовано монолітний веб-додаток, який згодом було оптимізовано шляхом переходу до мікросервісної архітектури. Дослідження також включало використання Prometheus, Kube State Metrics та Cert Manager для моніторингу та безпеки.

Результати дослідження показали, що первинна монолітна архітектура додатка мала значні обмеження у масштабованості. Оптимізація через розбиття на мікросервіси (див. рис. 1) дозволила зменшити затримки, підвищити ефективність використання ресурсів та покращити надійність системи. Автоматизоване масштабування (HPA), балансування навантаження та моніторинг у режимі реального часу суттєво підвищили продуктивність веб-додатків.

Одним із ключових факторів успішної оптимізації є використання ефективних засобів моніторингу. Інтеграція Prometheus і Kube State Metrics дозволяє отримувати детальну інформацію про використання ресурсів та швидко реагувати на зміни в системі. Це дає змогу оперативно виявляти потенційні проблеми та оптимізувати конфігурацію сервісів без значного зниження продуктивності.

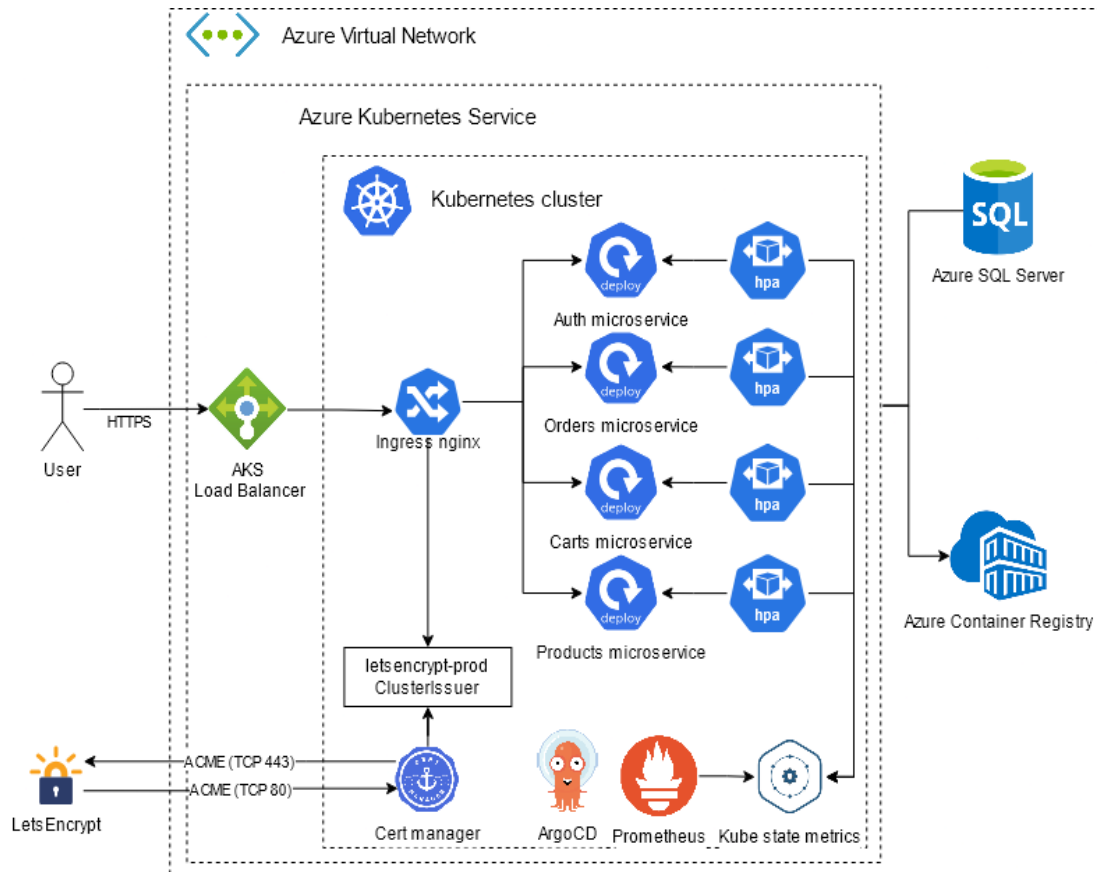


Рисунок 1 – Високорівнева архітектура мікросервісної інфраструктури

У сучасних веб-додатках контейнери широко використовуються для мікросервісної архітектури, де кожна функція або сервіс виконується в окремому контейнері. Це забезпечує гнучкість у масштабуванні та розподілі ресурсів, дозволяючи збільшувати або зменшувати кількість екземплярів конкретних мікросервісів залежно від навантаження. Оркестраційні платформи, такі як Kubernetes, надають можливості для автоматичного масштабування, балансування навантаження та управління відмовами, що робить контейнеризацію ефективною для створення високонавантажених веб-додатків.

Контейнеризація має свої виклики. Оптимальне управління ресурсами, наприклад CPU та пам'яттю, є критичним завданням, оскільки недостатнє виділення ресурсів може призвести до зниження продуктивності, тоді як надлишкове використання збільшує витрати. Важливим аспектом є також безпека контейнеризованих додатків, включаючи управління вразливостями у базових образах та налаштування правил доступу.

На основі отриманих результатів можна зробити висновок, що використання мікросервісної архітектури значно покращує продуктивність хмарних додатків. Це підтверджується вимірюваннями часу обробки запитів, середнього навантаження на процесор (CPU utilization) та

затримки відповіді. Kubernetes надає ефективні інструменти для автоматизації, проте їх застосування вимагає глибокого налаштування та постійного моніторингу.

Список використаних джерел:

1. Kubernetes and Docker Load Balancing: State-of-the-Art Techniques and Challenges. URL: https://www.researchgate.net/publication/376593267_Kubernetes_and_Docker_Load_Balancing_State-of-the-Art_Techniques_and_Challenges (дата звернення: 04.03.2025).

2. Enhanced Visibility for Real-time Monitoring and Alerting in Kubernetes by Integrating Prometheus, Grafana, Loki, and Alerta. URL: https://www.researchgate.net/publication/381347090_Enhanced_Visibility_for_Real-time_Monitoring_and_Alerting_in_Kubernetes_by_Integrating_PrometheusPrometheus_Grafana_Loki_and_Alerta (дата звернення: 04.03.2025).

3. Kubernetes in Microservices. URL: https://www.researchgate.net/publication/365344228_Kubernetes_in_Microservices (дата звернення: 04.03.2025).