Олена Олександрівна ГРІНЕНКО[1]**,**
кандидат технічних наук, доцент, доцент кафедри інформаційних технологій
ORCID ID: https://orcid.org/0000-0001-9673-6626
Роман Сергійович БОНДАРЕНКО[1]**,**
магістрант кафедри інформаційних технологій,
ORCID ID: https://orcid.org/0009-0005-0815-6303

[1] *Національний транспортний університет*

Прийняття: 02/06/2026
Рецензія: 12/06/2026
Публікація: 09/07/2026

DOI: https://doi.org/10.53920/ITS-2026-1-6

АНАЛІЗ ЕФЕКТИВНОСТІ СИСТЕМ МОНІТОРИНГУ ХМАРНОЇ ІНФРАСТРУКТУРИ НА БАЗІ PROMETHEUS ТА GRAFANA

УДК
004.4; 004.6; 004.9

У статті розглядається ефективність систем моніторингу хмарної інфраструктури на базі Prometheus та Grafana в сучасних розподілених середовищах. Сучасні хмарні обчислювальні середовища широко використовуються для розгортання інформаційних систем і сервісів, що призводить до зростання складності їх інфраструктури та підвищує вимоги до контролю її стану. У зв'язку з цим особливої актуальності набуває забезпечення ефективного моніторингу продуктивності, доступності та надійності компонентів системи. Швидкий розвиток хмарних технологій та мікросервісних архітектур призводить до збільшення складності інформаційних систем. Це вимагає надійних методів моніторингу стану інфраструктури та раннього виявлення потенційних проблем. Основна увага приділяється аналізу підходів до збору, обробки, зберігання та візуалізації показників продуктивності сервісів у хмарному середовищі.

Проведено аналіз архітектури системи моніторингу, реалізованої на платформі Amazon Web Services з використанням інструментів автоматизації конвеєрів Terraform, Docker та CI/CD. Досліджуються механізми збору телеметричних даних за допомогою Node Exporter та CloudWatch Exporter, а також можливості агрегації показників у Prometheus та їх подальшої візуалізації за допомогою Grafana. Аналізуються ключові показники продуктивності, такі як завантаження процесора, використання оперативної пам'яті, пропускна здатність мережі, інтенсивність запитів та стан бази даних.

ISSN 2786-7226 (print)
ISSN 2786-7234 (online)

Результати дослідження показали, що інтеграція Prometheus та Grafana забезпечує високу прозорість хмарної інфраструктури, дозволяє своєчасно виявляти аномалії, скорочує час реагування на збої та підтримує обґрунтовані рішення щодо масштабування сервісів. Було виявлено, що використання користувацьких інформаційних панелей та автоматизованих систем сповіщень підвищує ефективність управління інфраструктурою та сприяє раціональному використанню обчислювальних ресурсів. Особливу увагу було приділено оцінці впливу систем моніторингу на стабільність хмарних сервісів та можливості швидкого виявлення вузьких місць інфраструктури. Результати підтверджують, що використання сучасних інструментів моніторингу підвищує відмовостійкість систем, оптимізує витрати на інфраструктуру та покращує якість цифрових послуг. Практичне значення цієї роботи полягає в можливості застосування запропонованого підходу до побудови складних систем моніторингу в хмарних середовищах підприємств, інтенсивно використовуваних веб додатках та розподілених інформаційних системах.

Ключові слова: *хмарна інфраструктура, Prometheus, Grafana, Terraform, моніторинг систем, DevOps, AWS.*

Olena GRINENKO, Roman BONDARENKO

ANALYSIS OF THE EFFICIENCY OF CLOUD INFRASTRUCTURE MONITORING SYSTEMS BASED ON PROMETHEUS AND GRAFANA

The article considers the effectiveness of cloud infrastructure monitoring systems based on Prometheus and Grafana in modern distributed environments. Modern cloud computing environments are widely used to deploy information systems and services, which leads to an increase in the complexity of their infrastructure and increases the requirements for monitoring its state. In this regard, ensuring effective monitoring of the performance, availability and reliability of system components becomes particularly relevant. The rapid development of cloud technologies and microservice architectures leads to an increase in the complexity of information systems. This requires reliable methods for monitoring the state of the infrastructure and early detection of potential problems. The main attention is paid to the analysis of approaches to collecting, processing, storing and visualizing service performance indicators in the cloud environment.

The architecture of the monitoring system implemented on the Amazon Web Services platform using the Terraform, Docker and CI/CD pipeline automation tools is analyzed. The mechanisms for collecting telemetry data using Node Exporter and CloudWatch Exporter are investigated, as well as the possibilities of aggregating

indicators in Prometheus and their subsequent visualization using Grafana. Key performance indicators such as processor load, RAM usage, network bandwidth, query intensity, and database status are analyzed.

The results of the study showed that the integration of Prometheus and Grafana provides high transparency of the cloud infrastructure, allows for timely detection of anomalies, reduces response time to failures, and supports informed decisions on scaling services. It was found that the use of custom dashboards and automated notification systems increases the efficiency of infrastructure management and contributes to the rational use of computing resources. Particular attention was paid to assessing the impact of monitoring systems on the stability of cloud services and the ability to quickly identify infrastructure bottlenecks. The results confirm that the use of modern monitoring tools increases system fault tolerance, optimizes infrastructure costs, and improves the quality of digital services. The practical significance of this work lies in the possibility of applying the proposed approach to building complex monitoring systems in enterprise cloud environments, intensively used web applications, and distributed information systems.

Keywords: *cloud infrastructure, Prometheus, Grafana, Terraform, system monitoring, DevOps, AWS.*

Постановка проблеми. Сучасні інформаційні системи все більше базуються на хмарних технологіях та мікросервісних архітектурах. Хоча такі підходи пропонують гнучкість, масштабованість та високу доступність сервісів, вони також збільшують складність управління інфраструктурою.

У середовищах з десятками або сотнями взаємопов'язаних сервісів постійний моніторинг стану ресурсів, продуктивності компонентів та загальної стабільності системи є надзвичайно важливим.

Однією з ключових проблем є своєчасне виявлення зниження продуктивності, перевантаження сервера, збоїв мережевих компонентів або перебоїв у роботі бази даних. Відсутність ефективних інструментів моніторингу може призвести до тривалих перебоїв у роботі сервісів, втрати даних та фінансових втрат.

Це особливо актуально для інтенсивно використовуваних веб додатків, де навіть короткочасне порушення роботи одного компонента може вплинути на функціональність усієї системи.

Для вирішення цих проблем використовуються системи моніторингу для збору, аналізу та візуалізації телеметричних даних. Prometheus та Grafana є одними з найпопулярніших рішень, що поєднують збір метрик, автоматичне сповіщення та створення аналітичних панелей. Водночас питання про те, як оцінити ефективність використання цих інструментів у

реальних хмарних середовищах, залишається актуальним і потребує подальших досліджень.

Аналіз останніх досліджень і публікацій. Хмарна інфраструктура є моделлю надання обчислювальних ресурсів у вигляді сервісу, при якій обчислювальні потужності, сховища даних і мережеві компоненти надаються користувачеві через мережу Інтернет за запитом.

Основними характеристиками хмарних обчислень є еластичність, автоматичне масштабування, відмовостійкість та оплата за фактичне використання ресурсів [3].

На відміну від традиційних дата-центрів, де сервери мають статичну конфігурацію і постійне розташування, у хмарному середовищі інфраструктура є динамічною.

Віртуальні машини, контейнери та мережеві ресурси можуть створюватися і видалятися автоматично залежно від навантаження.

У мікросервісній архітектурі система складається з великої кількості невеликих сервісів, що взаємодіють між собою через мережу, а її стан постійно змінюється.

Проблематика моніторингу хмарних інфраструктур є одним із пріоритетних напрямів розвитку сучасних інформаційних технологій. У роботах, присвячених концепціям DevOps та Site Reliability Engineering, значна увага приділяється забезпеченню спостережуваності систем та автоматизації процесів контролю їхнього стану.

Важливими характеристиками ефективної системи моніторингу визначаються надійність збору даних, швидкість виявлення інцидентів, можливість локалізації причин збоїв, масштабованість та ефективність використання ресурсів [9].

У розподілених та хмарних системах оцінювання стану інфраструктури не може базуватися лише на одному типі даних. Різні рівні функціонування — від апаратних ресурсів до взаємодії користувача із сервісом — потребують різних методів спостереження. Тому у сучасних підходах до експлуатації застосовують дві основні моделі моніторингу: white-box та black-box [4].

White-box (внутрішній) моніторинг передбачає отримання інформації безпосередньо з компонентів системи. Сервіси експортують технічні показники роботи, які збираються системою моніторингу через спеціальні інтерфейси.

Як показано на рис. 1., моніторингова система періодично опитує застосунки та інфраструктурні компоненти, отримуючи метрики у форматі часових рядів.

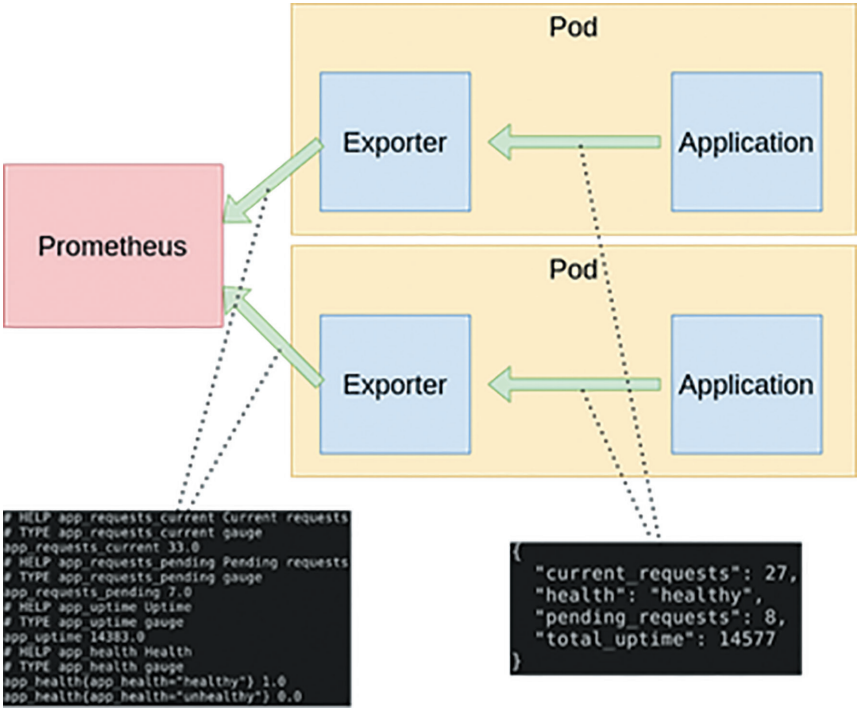

Рис. 1. Схема White-box моніторингу

Джерело: [4]

Black-box (зовнішній) моніторинг розглядає систему як «чорну скриньку». Перевірка виконується шляхом імітації дій користувача: надсилання HTTP-запитів, тестових транзакцій або перевірки відкриття мережевих портів.

Як видно на рис. 2, моніторинг здійснюється із зовнішнього вузла, який не має доступу до внутрішньої логіки сервісу.

У сучасних дослідженнях Prometheus розглядається як один із найефективніших інструментів для збору та зберігання часових рядів, тоді як Grafana забезпечує потужні засоби візуалізації та аналізу отриманих даних.

Значна кількість наукових праць присвячена інтеграції цих інструментів із контейнеризованими середовищами, Kubernetes-кластерами та хмарними платформами [6].

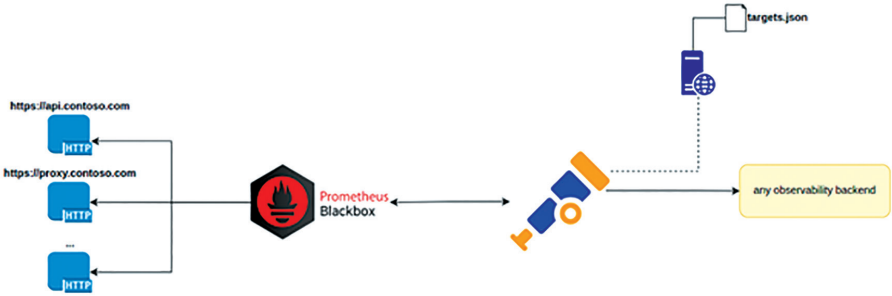

Рис. 2. Схема black-box моніторингу

Джерело: [4]

Однак більшість досліджень зосереджена на теоретичних аспектах впровадження систем моніторингу, тоді як питання практичного оцінювання їх ефективності в інфраструктурах AWS із використанням автоматизованого розгортання та мікросервісного підходу висвітлені недостатньо повно. Це обумовлює необхідність проведення додаткових досліджень та аналізу реальних сценаріїв використання Prometheus і Grafana.

Мета статті. Метою статті є аналіз ефективності систем моніторингу хмарної інфраструктури на базі Prometheus та Grafana, дослідження їх функціональних можливостей щодо збору, обробки та візуалізації метрик, а також оцінювання впливу впровадження цих інструментів на продуктивність, надійність і керованість хмарних сервісів.

Виклад основного матеріалу дослідження. Для проведення аналізу було використано хмарну інфраструктуру Amazon Web Services, розгорнуту засобами Terraform. Архітектура середовища включала декілька EC2-інстансів із фронтенд-сервісами на основі React та Angular, бекенд-компонентами на платформі .NET, а також окремими сервісами моніторингу Prometheus та Grafana. Автоматизація розгортання забезпечувалася за допомогою Docker-контейнерів і CI/CD-пайплайнів.

Кожен EC2 інстанс прив'язується до відповідного IAM Instance Profile, який надає лише мінімально необхідні права доступу. Це дозволяє безпечно збирати метрики системи та взаємодіяти з ресурсами AWS, такими як EC2, RDS та CloudWatch, без ризику надлишкових дозволів.

Конфігурація IAM ролей реалізована повністю через Terraform, що забезпечує відтворюваність та контроль версій. Для сервісу Prometheus застосовується політика AmazonEC2ReadOnlyAccess, яка дозволяє автоматично виявляти нові інстанси та збирати метрики з усіх EC2 ресурсів.

Завдяки цьому моніторинг залишаються актуальним навіть при масштабуванні інфраструктури або зміні конфігурації серверів. Загальна схема архітектури системи наведена на рис. 3, де показано взаємодію всіх компонентів та потоки даних між ними.

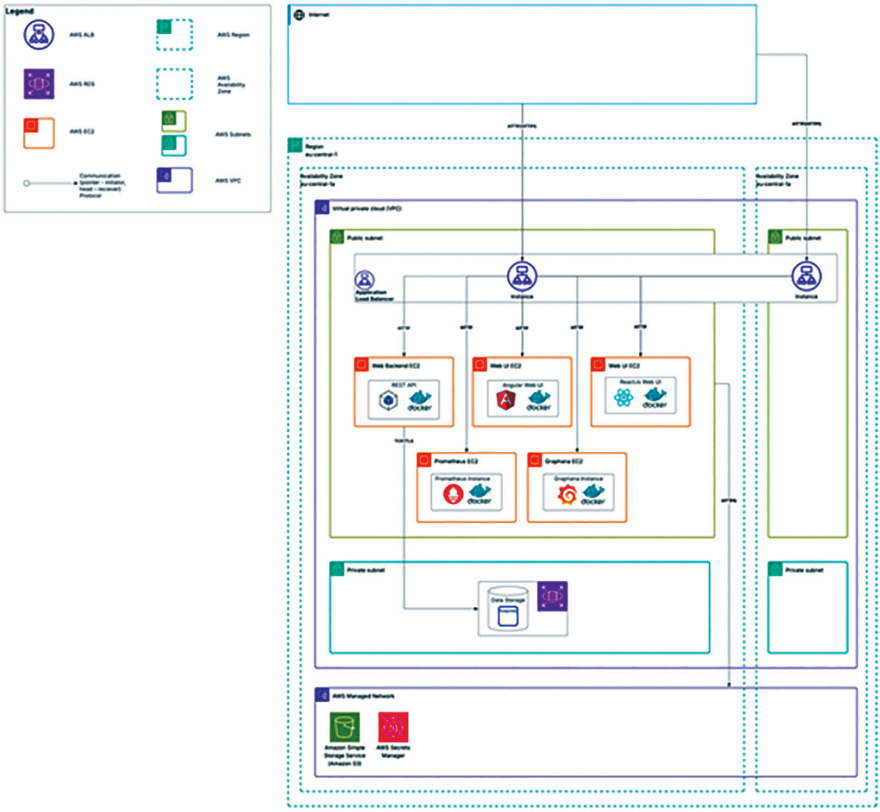

Рис. 3. Архітектура системи моніторингу

Джерело: авторська розробка

Збір системних метрик здійснювався через Node Exporter, який забезпечував отримання даних про завантаження процесора, використання оперативної пам'яті, стан дискових підсистем та мережевих інтерфейсів. Для моніторингу сервісів AWS використовувався CloudWatch Exporter, що надавав інформацію про роботу EC2, Application Load Balancer та баз даних RDS.

Prometheus виконував функцію централізованого збору та зберігання часових рядів. Завдяки механізмам автоматичного виявлення сервісів система забезпечувала актуальність даних навіть при масштабуванні інфраструктури. Зібрані метрики використовувалися для подальшого аналізу та побудови інформаційних панелей у Grafana.

Для оцінювання ефективності системи моніторингу було використано такі критерії: завантаження процесора, використання оперативної пам'яті, пропускна здатність мережі, інтенсивність обробки запитів через балансувальник навантаження та стан бази даних.

Після розгортання системи моніторингу було проведено комплексне тестування працездатності компонентів веб додатку та аналіз продуктивності інфраструктури. Основою для спостереження є Prometheus, який збирає метрики з усіх EC2 інстансів, Application Load Balancer та бази даних RDS, а Grafana забезпечує їх візуалізацію та аналітику у реальному часі. Для забезпечення стабільності та повторюваності конфігурації був використаний кастомний дашборд Grafana, автоматично розгорнутий через механізм provisioning.

Dockerfile для запуску Grafana передбачає копію конфігураційного файлу grafana.ini та директорії provisioning, де зберігаються JSON-файли опису дашбордів. Постійне зберігання даних здійснюється у томі /var/lib/grafana, а порт веб-інтерфейсу був змінено на 3001 для уникнення конфліктів з іншими сервісами.

На рисунку 4. представлено запущений дашборд Infrastructure Overview у Grafana після деплою системи моніторингу та збору метрик із усіх компонентів веб-додатку. Панелі дашборду демонструють стан CPU та пам'яті EC2 інстансів, активність фронтенд-запитів через Application Load Balancer та залишкову пам'ять бази даних RDS. Візуалізація метрик у реальному часі дозволяє оперативно оцінювати продуктивність системи, виявляти аномалії та реагувати на потенційні проблеми ще до виникнення критичних ситуацій.

Кастомне групування панелей і використання кольорових порогів для сповіщень забезпечує наочне сприйняття даних і швидку інтерпретацію результатів тестування продуктивності.

Цей скріншот ілюструє роботу дашборду в реальному середовищі після автоматичного розгортання Grafana через Docker і інтеграцію з Prometheus та CloudWatch Exporter, що підтверджує правильність налаштувань та коректність збору всіх необхідних метрик.

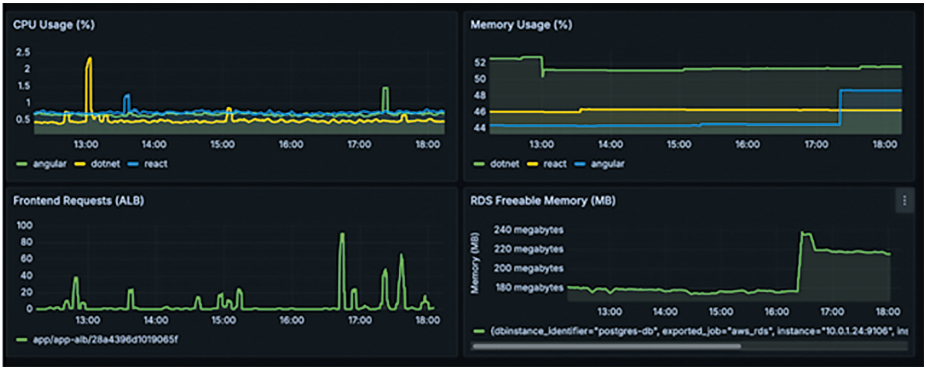

Рис. 4. Приклад дашборду Infrastructure Overview

Джерело: авторська розробка

У сучасних хмарних інфраструктурах веб додатків особливу увагу приділяють не лише розгортанню сервісів, а й оцінці їх продуктивності та ефективності роботи. Порівняльний аналіз системи моніторингу та її компонентів за визначеними критеріями дозволяє виявити сильні та слабкі сторони інфраструктури, оптимізувати використання ресурсів і забезпечити високу доступність додатку.

Для проведення такого аналізу були обрані наступні критерії: завантаження процесора, використання оперативної пам'яті, пропускна здатність мережі, обробка запитів через Application Load Balancer, а також ефективність роботи бази даних RDS.

Завдяки інтеграції всіх метрик у єдиний кастомний дашборд Grafana, можливо одночасно оцінювати всі ключові показники продуктивності. Візуалізація CPU, пам'яті, пропускної здатності ALB та стану бази даних дозволяє проводити комплексний порівняльний аналіз ефективності роботи інфраструктури.

Також дашборд дозволяє аналізувати тенденції змін метрик протягом часу, виявляти аномалії та визначати слабкі місця системи, які потребують оптимізації.

Проведене тестування показало, що сервіси React і Angular демонструють схожу ефективність при обробці фронтенд запитів, однак сервіс dotnet споживає більші ресурси CPU та пам'яті при виконанні бекенд операцій.

Сервіси моніторингу (Prometheus і Grafana) споживають мінімальні ресурси, що підтверджує їхню ефективність як системних компонентів. Аналіз дашборду дозволяє також побачити, які інстанси EC2 найбільш за-

вантажені і потребують масштабування, що важливо для забезпечення високої доступності веб додатку.

Висновки. Проведено аналіз ефективності систем моніторингу хмарної інфраструктури на базі Prometheus та Grafana. Досліджено особливості їх використання в AWS-середовищі з мікросервісною архітектурою та автоматизованим процесом розгортання. Отримані результати підтвердили, що інтеграція зазначених інструментів забезпечує комплексний контроль за станом інфраструктури, дозволяє здійснювати централізований збір метрик та їх подальший аналіз у режимі реального часу.

Встановлено, що застосування Node Exporter та CloudWatch Exporter забезпечує повне охоплення ключових компонентів інфраструктури, а використання Grafana значно підвищує ефективність аналізу отриманих даних завдяки наочній візуалізації та механізмам автоматичного сповіщення. Проведене тестування дозволило виявити особливості використання ресурсів окремими мікросервісами та визначити напрями їх оптимізації.

Практична цінність дослідження полягає у підтвердженні доцільності використання Prometheus та Grafana як основи сучасної системи моніторингу хмарної інфраструктури. Запропонований підхід сприяє підвищенню доступності сервісів, скороченню часу реагування на інциденти та оптимізації використання обчислювальних ресурсів.

Перспективним напрямом подальших досліджень є інтеграція систем моніторингу з технологіями машинного навчання для автоматичного прогнозування навантаження, виявлення аномалій та підтримки процесів автономного масштабування хмарних сервісів.

ЛІТЕРАТУРА

1. Google SRE Book. URL: https://sre.google/sre-book/monitoring-distributed-systems/ (дата звернення: 04.05.2026).

2. Документація Prometheus. URL: https://prometheus.io/docs/introduction/overview/ (дата звернення: 04.05.2026).

3. The NIST Definition of Cloud Computing. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf (дата звернення: 05.05.2026).

4. OpenTelemetry: Observability primer. URL: https://opentelemetry.io/docs/concepts/observability-primer/ (дата звернення: 05.05.2026).

5. Google SRE Book: Alerting on SLOs. URL: https://sre.google/workbook/alerting-on-slos/ (дата звернення: 04.05.2026).

6. Prometheus Monitoring: Use Cases, Metrics, and Best Practices. URL: https://www.tigera.io/learn/guides/prometheus-monitoring/ (дата звернення: 06.05.2026).

7. Grafana Cloud. URL: https://grafana.com/docs/grafana-cloud/visualizations/ (дата звернення: 06.05.2026).

8. Guide on Grafana Common Features. URL: https://staticintl.cloudcachetci.com/doc/pdf/product/pdf/1124_69439_en.pdf (дата звернення: 06.05.2026).

9. Datadog: Infrastructure Monitoring Overview. URL: https://www.datadoghq.com/knowledge-center/infrastructure-monitoring/ (дата звернення: 07.05.2026).

10. O'Reilly: Monitoring Distributed Systems. URL: https://theswissbay.ch/pdf/Books/Computer%20science/O'Reilly/monitoring-distributed-systems.pdf (дата звернення: 07.05.2026).

REFERENCES

1. Google SRE Book. URL: https://sre.google/sre-book/monitoring-distributed-systems/ (Accessed: 04.05.2026).

2. Prometheus documentation. URL: https://prometheus.io/docs/introduction/overview/ (Accessed: 04.05.2026).

3. The NIST Definition of Cloud Computing. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf (Accessed: 05.05.2026)

4. OpenTelemetry: Observability primer. URL: https://opentelemetry.io/docs/concepts/observability-primer/ (Accessed: 05.05.2026).

5. Google SRE Book: Alerting on SLOs. URL: https://sre.google/workbook/alerting-on-slos/ (Accessed: 04.05.2026).

6. Prometheus Monitoring: Use Cases, Metrics, and Best Practices. URL: https://www.tigera.io/learn/guides/prometheus-monitoring/ (Accessed: 06.05.2026).

7. Grafana Cloud. URL: https://grafana.com/docs/grafana-cloud/visualizations/ (Accessed: 05.06.2026).

8. Guide on Grafana Common Features. URL: https://staticintl.cloudcachetci.com/doc/pdf/product/pdf/1124_69439_en.pdf (Accessed: 06.05.2026).

9. Datadog: Infrastructure Monitoring Overview. URL: https://www.datadoghq.com/knowledge-center/infrastructure-monitoring/ (Accessed: 07.07.2026).

10. O'Reilly: Monitoring Distributed Systems. URL: https://theswissbay.ch/pdf/Books/Computer%20science/O'Reilly/monitoring-distributed-systems.pdf (Accessed: 07.05.2026).