

АНАЛІЗ ТЕХНОЛОГІЙ ІАС ДЛЯ РОЗГОРТАННЯ ВІДМОВОСТІЙКИХ ТА ВИСОКОНАВАНТАЖЕНИХ СЕРВІСІВ В ХМАРНОМУ СЕРЕДОВИЩІ AWS

Нестеренко М.М., Калюжний І.А.

Навчально-науковий інститут телекомунікаційних систем

КПІ ім. Ігоря Сікорського, Україна

E-mail: nikolaiy.nesterenko@gmail.com, kaluzhnyi.ivan@lil.kpi.ua

ANALYSIS OF IAC TECHNOLOGIES FOR DEPLOYING HIGHLY AVAILABLE AND HIGH-PERFORMANCE SERVICES IN THE AWS CLOUD ENVIRONMENT

This paper analyzes IaC technologies for deploying a fault-tolerant, high-load Kubernetes cluster in AWS. It explores automation, scalability, and maintainability using Terraform and Kubernetes. The study highlights best practices for cloud reliability and performance, proposing a highly available Kubernetes cluster design.

Сучасні інформаційно-комунікаційні системи масово переходять на контейнеризацію, що забезпечує гнучкість, ізоляцію та спрощує розгортання. Найпопулярнішою технологією контейнеризації є Docker, альтернативні - Podman, containerd і CRI-O. Проте використання лише систем контейнеризації створює проблеми з управлінням розробленим проектом, його відмовостійкістю, балансуванням навантаження, автоматичним масштабуванням. В зв'язку з цим, метою дослідження є визначення стеку технологій та побудова моделі кластеру для розгортання високонавантажених та відмовостійких застосунків.

Для керування контейнерами використовують системи оркестрації, які автоматизують розгортання, масштабування та забезпечують відмовостійкість. Найпопулярніші рішення: Docker Swarm (обмеженість у масштабуванні), Apache Mesos (доцільно використовувати в надвеликих масивах даних). Однією з технологій, яка забезпечує відмовостійкість і масштабування контейнерів завдяки автоматичному відновленню, балансуванню навантаження та адаптації до об'ємів трафіку є Kubernetes. Він має кластерну архітектуру (рис. 1.) і складається з наступних елементів:

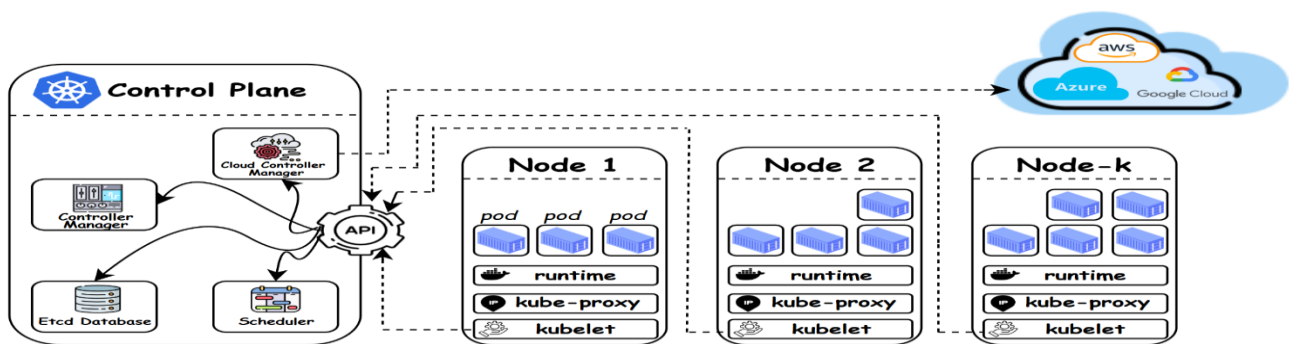


Рис. 1. Узагальнена архітектура Kubernetes кластеру.

За відмовостійкість в Kubernetes відповідають служби моніторингу стану та перезапуску подів (pod) у разі їх падіння, балансування навантаження та розподілення подів між різними нодами.

Kubernetes є універсальним рішенням для оркестрації (підтримує Docker, containerd та CRI-O) завдяки стандарту Container Runtime Interface (CRI). Також Kubernetes інтегрується з усіма великими хмарними провайдерами: AWS, Google Cloud, Azure. Kubernetes можна розгортати і на фізичних серверах або в дата-центрах, що дає змогу будувати гібридні або on-premises рішення, що робить його гнучким універсальним інструментом.

Для автоматизації розгортання, централізованого зберігання та контролю версій конфігурації використовуються Infrastructure as a Code (IaC) інструменти. Одним з найпоширеніших способів розгортання високонавантаженого та відмовостійкого Kubernetes кластеру є сервіс EKS (Elastic Kubernetes Service) на хмарній платформі AWS (Amazon Web Services).

Найбільш перспективним інструментом IaC для розгортання інфраструктури в гетерогенному хмарному середовищі, з можливістю зберігання та контролю версій конфігурації (наприклад - Git) є Terraform, який має ряд переваг: всі зміни описуються в коді (мінімізує людські помилки); попередня перевірка конфігурацій за допомогою команд для виявлення помилок; використання state file для відстеження поточного стану інфраструктури; автоматичне керування взаємозалежними ресурсами для уникнення помилок при створенні та оновленні інфраструктури; інтеграція з різними хмарними і on-premise провайдерами; декларативний підхід опису інфраструктури без необхідності написання складних скриптів. Запропоновано розгортання кластеру Kubernetes на основі AWS EKS за допомогою Terraform (рис. 2.).

Майбутні дослідження можуть зосередитися на інтеграції IaC з різними Cloud-провайдерами, автоматизації CI/CD (Continuous Integration/Continuous Delivery/Deployment) процесів, покращенні безпеки та оптимізації масштабованих сервісів.

Література

1. Reddy Chittibala D. Infrastructure as Code (IaC) and Its Role in Achieving DevOps Goals. International Journal of Science and Research (IJSR). 2023. Т. 12, № 1. С. 1258–1262. URL: <https://doi.org/10.21275/sr24304170702>.
2. Odeyinka O. J. AWS Cloud Automation : Textbook. 2nd ed. Berlin : DE Gruyter GmbH, Walter, 2024. 366 p.
3. Romanov, O., Nesterenko, M., Boggia, G., Striccoli, D. Construction and Methods for Solving Problems at the SDN Control Level // Lecture Notes in Electrical Engineering, 2023, 965 LNEE, p. 85–101, https://link.springer.com/chapter/10.1007/978-3-031-24963-1_6
4. Романов О.І., Бурлака Г.Ю. Управління мережею SDN з використанням контролера RYU/ Вчені записки таврійського національного університету імені В.І. ВЕРНАДСЬКОГО/ Серія: Технічні науки Том 34 (73) № 5 2023, С. 59-69, DOI <https://doi.org/10.32782/2663-5941/2023.5/11>
5. Романов О., Бурлака Г., Берестовенко О., Підпалій О. Технічні особливості побудови Li-Fi мережі за допомогою методів керування SDN/ Вісник ЧДТУ 2023; 0 (3): С:16-25; DOI: 10.24025/2306-4412.3.2023.284893
6. O. Romanov, V. Mankivskyi, I. Saychenko and M. Nesterenko, "Event Model Algorithm with Microservice Architecture Implementation," 2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2022, pp. 561-566, doi: 10.1109/PICST57299.2022.10238498.