

Література:

1. Elhorst J.P. *Spatial Econometrics: From Cross-Sectional Data to Spatial Panels*. Berlin: Springer, 2020. URL: <https://arm.ssuv.uz/frontend/web/books/643105596a3dd.pdf> (дата звернення 28.11.2025)
2. Kilian L., Lütkepohl H. *Structural Vector Autoregressive Analysis*. Cambridge: Cambridge University Press, 2021. 740 p.
3. OECD. *Regional Outlook 2023: The Longstanding Geography of Inequalities*. Paris: OECD Publishing, 2023. URL: https://www.oecd.org/content/dam/oecd/en/publications/reports/2023/10/oecd-regional-outlook-2023_3a6174ba/92cd40a0-en.pdf (дата звернення 30.11.2025)
4. Антонів В. Аналіз економічних сценаріїв політики енергоефективності в рамках енергетичної стратегії Європейського Союзу. Формування ринкової економіки в Україні. *Наук. зб.* 2024. № 51. С. 116-128.

УДК 004.624

Олексій БОЙКО, к.т.н., доцент

Вінницький національний аграрний університет, Україна

Віталій СУПРИГАН, к.т.н., доцент

Вінницький національний аграрний університет, Україна

АВТОМАТИЗОВАНА РЕКОНСТРУКЦІЯ МІКРОСЕРВІСНОГО СЕРЕДОВИЩА З KUBERNETES У ЛОКАЛЬНИЙ DOCKER COMPOSE

Alexei BOIKO, Ph.D. in Computer Science, Assoc. Prof.

Vinnitsia National Agrarian University, Ukraine

Vitaly SUPRYHAN, Ph.D. in Computer Science, Assoc. Prof.

Vinnitsia National Agrarian University, Ukraine

AUTOMATED RECONSTRUCTION OF A MICROSERVICE ENVIRONMENT FROM KUBERNETES TO LOCAL DOCKER COMPOSE

Вступ. Сучасні корпоративні платформи дедалі частіше базуються на мікросервісній архітектурі, розгорнутій у кластерному середовищі Kubernetes. Такі системи забезпечують масштабованість, стійкість та ізоляцію компонентів, однак створюють труднощі для локального тестування та налагодження. Розробники та тестувальники змушені вручну відтворювати середовище, наближаючи його до продукційного, що є трудомістким та помилкозалежним процесом [1, 2]. Особливо це відчутно у великих проєктах із десятками сервісів, де кожна зміна у конфігурації може спричинити дрейф параметрів між локальною інфраструктурою та кластером.

У цьому контексті актуальним є завдання автоматизованої реконструкції кластерного середовища у локальну інфраструктуру Docker Compose [3]. Автоматизація дозволяє зменшити залежність від людського чинника, спростити онбординг нових членів команди та забезпечити однаковість середовищ для Dev, QA та CI-процесів [4, 5].

Мета і завдання дослідження. Мета роботи полягає у розробленні методики автоматичного перетворення Kubernetes Deployment-маніфестів у відтворюваний файл *docker-compose.yml* із збереженням максимальної функціональної еквівалентності. Досягнення такої відповідності дозволяє локально запускати системи, поведінка яких

близька до кластерного оригіналу, що значно знижує ризики появи дефектів, зумовлених різницею середовищ.

Для досягнення мети визначено такі завдання:

- формалізувати поняття вірності реконструкції (fidelity) як показника відповідності локального середовища оригінальному кластеру, включаючи як конфігураційні, так і поведінкові аспекти;
- розробити алгоритм детермінізму збірки (determinism), який забезпечує повторюваність конфігурації при багатократному запуску інструменту та гарантує стабільність результатів;
- створити програмний інструмент автоматичного перетворення маніфестів з підтримкою env, envFrom, ConfigMap, Secret, volumeMounts, ports та health-checks, а також логіки умовної ініціалізації сервісів;
- оцінити втрати конфігураційних даних та дрейф параметрів під час реконструкції, зокрема виявити класи конфігів, які частіше зазнають спотворень або втрачаються під час перетворення.

Огляд сучасних підходів. Попередні дослідження у сфері DevOps-автоматизації зосереджувалися переважно на зворотному процесі — деплойменті Docker Compose у Kubernetes-кластер. Це призвело до появи численних інструментів, які виконують трансляцію docker-compose.yml у Kubernetes-ресурси. Водночас напрямок зворотної реконструкції практично не стандартизований і розглядається лише частково у контексті дебагінгу або побудови lightweight-стендів.

Найявні утиліти (kompose, k2d, podman-generate) не враховують низку критичних аспектів:

- передачу секретів із збереженням формату, типів кодування та механізмів оновлення;
- динамічне формування змінних середовища, залежних від інших ресурсів або операторів;
- логіку initContainers, pre-start hooks і скриптів підготовки томів;
- поведінкові health-check-метрики, які впливають на повноту відтворення і стабільність старту системи;
- політики ресурсів, включно з обмеженнями CPU/Memory, affinity та nodeSelectors, що у кластері впливають на розміщення контейнерів.

Таким чином, існуючі рішення не забезпечують достатньої точності реконструкції і вимагають значної ручної участі під час підготовки тестового стенду.

Методика дослідження. Для оцінювання ефективності розробленого методу використано корпус із понад 20 реальних мікросервісів, що розгортаються у Kubernetes-кластері. Сервіси значно відрізнялися за структурою: від простих REST-шлюзів до сервісів зі складною системою секретів, томів та внутрішніх залежностей.

Розроблений алгоритм реконструкції виконує такі етапи:

1. Парсинг маніфестів — витягнення специфікацій Deployment, Service, ConfigMap і Secret, включно зі вкладеними структурами та перехресними посиланнями.
2. Нормалізація середовищ — побудова єдиного дерева залежностей контейнерів, включно з фантомними залежностями (сервіси, що не визначені явно, але згадуються у змінних середовища).
3. Генерація Compose-конфігурації — відтворення портів, томів, змінних середовища, секретів, стартових команд і мережевих обмежень. Окремо враховується порядок старту сервісів та логіка залежностей між ними.

4. Аналіз вірності — автоматичне порівняння з еталонними даними з Kubernetes через `kubectl get` і `kubectl describe`. Під час аналізу окремо оцінюється змістові збіги (`spec`) та поведінкові збіги (`status.conditions`).

5. Обчислення метрик:

- *coverage rate* — частка правильно реконструйованих параметрів;
- *drift rate* — ступінь відхилення параметрів у порівнянні зі специфікацією кластеру;
- *time-to-launch* — середній час від запуску скрипту до готовності всіх контейнерів;
- *environment consistency score* — коефіцієнт відповідності змінних середовища, що впливають на поведінку сервісів.

Результати. Експериментальні випробування засвідчили, що середній рівень вірності реконструкції становить $\approx 0,92$ (з 1,0) для сервісів без специфічних Kubernetes-операторів. Деякі сервіси продемонстрували повну відповідність, особливо ті, що мають мінімальні залежності та не покладаються на специфічні механізми секретів.

Основними джерелами втрат стали:

- контейнери з динамічними секретами, які не відображаються у ConfigMap і вимагають зовнішніх операторів;
- нестандартні `initContainers`, які виконують складну логіку до старту основного сервісу;
- специфічні політики ресурсів (`limits`, `requests`, `affinity`), що не мають повноцінного аналога у Docker Compose та потребують ручної корекції;
- відмінності у логіці `health-checks`, зумовлені різними інтерпретаціями таймерів та порогів у Docker і Kubernetes.

Запропонована система формує рамку перевірки вірності (Fidelity Framework), що дозволяє Dev- та QA-командам автоматично вимірювати відхилення локального середовища від кластерного. Такий підхід підвищує прозорість процесу налагодження та забезпечує можливість швидкого виявлення конфігураційних несумісностей.

Наукова новизна.

1. Уперше формалізовано поняття вірності реконструкції (*fidelity*) як кількісного показника відповідності мікросервісних середовищ з урахуванням структурних та поведінкових метрик.

2. Запропоновано метрику детермінізму збірки (*determinism*) для перевірки повторюваності локальної конфігурації та для контролю дрейфу у Dev-циклі.

3. Розроблено алгоритм автоматичного порівняння параметрів середовищ на основі YAML-диференціювання, аналізу зв'язностей та динамічного моніторингу `health-check`-подій.

Висновки. Автоматизована реконструкція мікросервісного середовища Kubernetes у Docker Compose значно скорочує час налагодження та підвищує відтворюваність експериментів. Запропонована методика дозволяє звільнити розробників від рутинних операцій, пов'язаних з ручною синхронізацією середовищ, та мінімізує ризики появи помилок.

Розроблена система і метрики *fidelity* / *determinism* створюють основу для побудови інструментів контролю дрейфу конфігурацій у Dev-циклі, а також можуть бути інтегровані у CI/CD-конвеєри.

Подальші дослідження передбачають розширення аналізу на Stateful-компоненти, інтеграцію підтримки операторів Kubernetes та впровадження механізмів автоматичного відновлення втраченої конфігурації.

Література

1. Hightower Kelsey. Kubernetes: Up and Running: Dive into the Future of Infrastructure. 1st Edition. O'Reilly Media, 2017. – 202 p.
2. Roland Huss, Bilgin Ibryam. Kubernetes Patterns: Reusable Elements for Designing Cloud Native Applications. 2nd Edition. O'Reilly Media, 2023. – 390 p.
3. Merkel Dirk. Docker: Lightweight Linux Containers for Consistent Development and Deployment // *Linux Journal*. – 2014. – № 239. – P. 76–91.
4. Суприган В. А. Алгоритми трансформації змінних середовища при міграціях інфраструктури. *Інноваційний вектор розвитку обліку, фінансів, аналізу й аудиту в Україні та світі* : зб. праць учасників IV Міжнар. наук.-практ. конф. (6-7 листоп. 2025 р.). / Поліський національний університет, Житомир, 2025. С. 119–122.
5. Суприган Віталій. Безпечне вилучення конфігурацій і секретів бухгалтерських інформаційних систем для локального тестування на основі моделей редакції та синтетики. *Сучасні тенденції розвитку обліку, аналізу, контролю, аудиту та оподаткування*: матеріали міжнародної наук.-практ. конф. (21 листопада 2025 р.) / Волинський національний університет імені Лесі Українки, Луцьк, 2025. С. 193–194

УДК 336.338

Ігор БУЧКОВСЬКИЙ

Наталія ГАРМАТІЙ, канд. екон. наук, доцент

Тернопільський національний технічний університет імені Івана Пулюя

МОДЕЛЮВАННЯ ПОЗИЦІЙ СТРАХОВИХ КОМПАНІЙ УКРАЇНИ МЕТОДИКОЮ КЛАСТЕРНОГО АНАЛІЗУ

Igor BUCHKOVSKIY

Nataliya HARMATIY, Ph.D., Assoc. Prof.,

Ternopil Ivan Puluj National Technical University, Ukraine

MODELING POSTIONS OF INSURANCE COMPANIES OF UKRAINE USING THE CLUSTER ANALYSIS METHOD

Застосування інструментарію статистичного аналізу, а також кластерного аналізу є досить актуальні наукові методи для дослідження фінансових структур з метою покращення та вдосконалення фінансового стану об'єкту.

Таким чином, *кластерний аналіз* – це сукупність методів класифікації багатомірних спостережень або об'єктів, заснованих на визначенні поняття відстані між об'єктами з наступним виділенням з них груп, "згустків" спостережень (кластерів, таксонів).[1].

Кластерний аналіз здійснювався на основі вхідних даних страхових компаній за розміром на ринку страхування фінансових ризиків за 2024 рік таких компаній, як «Княжа», «Аска», «Арсенал Страхування», «Альфа Страхування» і «Уніка».