

# ТЕХНОЛОГІЯ АВТОМАТИЗАЦІЇ РОЗГОРТАННЯ ОБЧИСЛЮВАЛЬНОЇ ІНФРАСТРУКТУРИ В ХМАРНОМУ СЕРЕДОВИЩІ

Вінницький національний технічний університет;

## Анотація

У роботі покращуються алгоритми створення та використання IaC як засобу для автоматизації розгортання обчислювальної інфраструктури в хмарному середовищі. IaC розглядає ключові аспекти технології, включаючи її потенціал для підвищення ефективності, безпеки, зменшення витрат на підтримку в процесі управління хмарною інфраструктурою.

**Ключові слова:** IaC, Terraform, автоматизація, обчислювальна інфраструктура, хмарне середовище, ефективність, безпека, управління інфраструктурою.

## Abstract

*The use of IaC (Infrastructure as Code) as a tool for automating the deployment of computing infrastructure in a cloud environment has been investigated and analyzed. It examines the key aspects of the technology, including its potential to enhance efficiency, security, and reduce maintenance costs in the process of managing cloud infrastructure.*

**Keywords:** IaC, Terraform, automation, computing infrastructure, cloud environment, efficiency, security, infrastructure management.

## Вступ

Інфраструктура як код (IaC - Infrastructure as Code) дозволяє надавати та підтримувати обчислювальну інфраструктуру за допомогою коду замість ручних процесів та налаштувань.

Подібно до того, як програмний код описує додаток та принципи його роботи, інфраструктура як код (IaC) описує архітектуру системи та принципи її роботи. Архітектура інфраструктури містить такі ресурси, як сервери, мережі, операційні системи та сховища. IaC управляє віртуалізованими ресурсами, обробляючи конфігураційні файли як файли вихідного коду. Рішення можна використовувати для повторюваного управління інфраструктурою.

Ручне управління інфраструктурою займає багато часу, і при ньому є ризик виникнення помилок, особливо при керуванні через UI (User interface) в умовах зміни масштабу. IaC дозволяє визначити бажаний стан інфраструктури, не включаючи всі кроки переходу в цей стан. Завдяки автоматизованому управлінню інфраструктурою, розробники можуть зосередитися на розробці та вдосконаленні додатків, а не на керуванні середовищем. Організації використовують інфраструктуру як код для контролю витрат, зниження ризиків та швидкого реагування на нові бізнес-можливості.

Сучасні організації, незалежно від їх масштабів та галузей діяльності, все більше спрямовані на використання хмарних інфраструктур для забезпечення високої доступності, масштабованості та зручності управління своїми обчислювальними ресурсами. [1], [2].

## Основна частина

Infrastructure as Code (IaC) робить революцію у світі програмної інженерії, кардинально змінюючи підхід до налаштування та підтримки інфраструктури. Але, попри те, що IaC стало майже стандартом у галузі, все ще тривають серйозні дискусії щодо того, що це таке - і як це зробити правильно.

Епоха IT, коли ви фактично купували власні сервери відходить в минуле, зараз це здається доволі дивним, що розвиток інфраструктури був обмежений циклом закупівель апаратного забезпечення. І оскільки на доставлення нового сервера йшло кілька тижнів, також час витрачався на його установку у відповідні приміщення та налаштування операційної системи на ньому. Люди просто вставляли

диск у сервер та слідували чек-листу. Через кілька днів він ставав доступним для використання розробниками. [3]

Хмарні обчислення та можливість без зусиль запускати нові інстанси віртуальних машин (сервери) принесли чимало переваг для інженерів та бізнесу, але це також означало, що тепер їм доводилося опікуватися постійно зростаючою кількістю цих серверів.

Отож як бачимо завдання по створенню інфраструктури вимагає методів та алгоритмів та програмних засобів котрі здатні керувати такими процесами.

Створення нового середовища або інфраструктуру можна легко налаштувати за допомогою програмного коду використавши мову програмування HCL (Hashicorp Configuration Language). Якщо інженеру інфраструктури потрібно збільшити кількість серверів, все що від нього вимагається, збільшити значення змінної в коді і перезапустити код за допомогою команди: `terraform apply`. Програмні модулі котрі були написанні інженером, під'єднується до API (application programming interface) відповідного клаудного провайдера і вносить відповідні зміни в базу даних де зберігаються налаштування Ваших серверів.[4]

IaC це декларативним інструментом (він описує стан, який ви хочете, щоб ваше середовище мало), ви можете виявляти та виправляти відхилення середовища за допомогою коду. Якщо частина інфраструктури змінюється вручну поза кодом, вона може бути відновлена до бажаного стану під час наступного запуску, що є досить зручним процесом, що дозволяє тримати вашу інфраструктуру незмінною, це відомо як ідемпотентність. [5]

Розробляючи та вдосконалюючи код, витрати на інфраструктуру знижуються, оскільки зменшується час на розгортання та зусилля на управління, адміністрування та підтримку середовищ.

Команди інженерів по інфраструктурі можуть тестувати додатки у середовищах, схожих на виробничі, на ранніх етапах циклу розробки будь якого програмного продукту.

## Висновки

Отже, за допомогою IaC (Infrastructure as Code) було створено інформаційну систему котра являє собою набір операційних систем, програмного забезпечення, серверів, мереж та підмереж, LoadBalancer-ів, сховищ зберігання даних. Така система може бути легко відтворена в своєму початковому вигляді завдяки удосконаленим алгоритмам та може бути використана повторно.

Оскільки культура DevOps стає все більш поширеною, підтримання інфраструктури за допомогою IaC, ймовірно, стане все важливішою метою в процесі створення великих інформаційних систем.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Stephane Jourdan, Pierre Pomes “Infrastructure as Code (IaC) Cookbook”, (2017)
2. What is Terraform [Електронний ресурс]. Режим доступу: <https://developer.hashicorp.com/terraform/intro>
3. What is Infrastructure as Code (IaC) [Електронний ресурс]. Режим доступу: <https://www.redhat.com/en/topics/automation/what-is-infrastructure-as-code-iac>
4. Using Recommendations for Infrastructure as Code [Електронний ресурс]. Режим доступу: <https://cloud.google.com/recommender/docs/tutorial-iac>.
5. Ключові концепції Terraform [Електронний ресурс]. Режим доступу: <https://www.terraform-best-practices.com/v/uk/key-concepts>

**Шевченко Олександр Вікторович** - студентка групи 2КІ-22м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: [oleksandr.viki@gmail.com](mailto:oleksandr.viki@gmail.com)

**Захарченко Сергій Михайлович** – професор, кафедра обчислювальної техніки, Вінницький національний технічний університет, Вінниця, e-mail: [zahar@vntu.net](mailto:zahar@vntu.net)