

ІНФРАСТРУКТУРА ЯК КОД, ХМАРНА ІНТЕГРАЦІЯ ТА НОВІ ТЕНДЕНЦІЇ

Інфраструктура як код (IaC) фундаментально переосмислює традиційний підхід до управління та впровадження ІТ-інфраструктури. У своїй суті IaC передбачає представлення конфігурацій інфраструктури через код, що дозволяє автоматизовано виконувати розгортання, конфігурацію та масштабування ресурсів [1]. Цей перехід від ручних, вразливих до помилок процесів до автоматизації, що базується на коді, керується кількома ключовими принципами. Перш за все – це концепція кодифікації інфраструктури, де конфігурації виражаються у форматі, що сприймається комп'ютерами. Цей принцип сприяє контролю версій, дозволяючи командам відстежувати зміни, ефективно співпрацювати та повертатися до попередніх станів за необхідності. Крім того, IaC підкреслює ідемпотентність конфігурацій, що означає, що однаковий набір інструкцій, який виконується кілька разів, дає однаковий результат, забезпечуючи передбачуваність у розгортанні. Оброблюючи інфраструктуру як код, організації можуть досягти гнучкості, швидкості та надійності у керуванні своїми ІТ-ресурсами.

Впровадження IaC включає ключові компоненти для визначення та оркестрації бажаного стану інфраструктури. Шаблони, представлені мовами типу YAML або JSON, служать схемами для ресурсів інфраструктури, описуючи конфігурацію, залежності та взаємозв'язки між компонентами. Системи контролю версій, такі як Git, використовуються для контролю версій шаблонів. Автоматизовані засоби впровадження, такі як Terraform або Ansible, інтерпретують шаблони та реалізують ресурси в хмарних середовищах [2]. Інтеграція цих компонентів дозволяє організаціям оптимізувати весь життєвий цикл інфраструктури, включаючи створення, зміни, масштабування та виведення з експлуатації.

Автоматизація та оркестрація є невід'ємними аспектами реалізації повного потенціалу IaC [3]. Автоматизація передбачає систематичне виконання заздалегідь визначених завдань без ручного втручання. В IaC автоматизація дозволяє швидко та послідовно надання інфраструктурних ресурсів, зменшуючи ймовірність помилок, пов'язаних з ручними процесами. Оркестрація цих завдань стає важливою при роботі з складними, багатокомпонентними системами. В контексті IaC оркестрація вказує на координацію та управління кількома автоматизованими процесами для досягнення бажаного стану. Це забезпечує синхронне розгортання різних компонентів інфраструктури, сприяючи загальній ефективності та надійності системи.

Багато інструментів полегшують впровадження IaC, кожен зі своїми унікальними можливостями та функціоналом [3]. Terraform дозволяє користувачам визначати та надавати інфраструктуру за допомогою декларативної мови конфігурації. Ansible, з іншого боку, дотримується імперативного підходу, фокусуючись на завданнях з автоматизації та управлінні конфігурацією. Chef та Puppet – це додаткові інструменти, які акцентують на автоматизації процесів управління конфігурацією.

Інфраструктура як код (IaC) в хмарних обчисленнях принесла численні переваги: швидкість та гнучкість при управлінні ресурсами, автоматизація розгортання, що скорочує час масштабування і налаштування. Однорідність конфігурацій на різних етапах розробки забезпечує їхню однаковість, а IaC дозволяє динамічно виділяти та звільняти ресурси згідно із вимогами хмарових послуг. Підтримка спільної мови та процесу сприяє співпраці між розробниками та операторами, сприяючи культурі DevOps і прискорюючи постачання додатків.

В майбутньому кілька нових технологій готуються визначити майбутню картину IaC. Одним із помітних трендів є зростання serverless архітектур, де акцент переноситься з управління компонентами інфраструктури на виконання окремих функцій [3]. Безсерверне обчислення абстрагує базову інфраструктуру, дозволяючи розробникам зосередитися лише на написанні коду, тим самим підвищуючи гнучкість та ефективність використання ресурсів. Ще однією новою методологією є GitOps, парадигма, які дуже глибоко використовують репозиторії Git як джерело для конфігурацій інфраструктури. Цей підхід сприяє управлінню інфраструктурою, яке базується на контролі версій, аудиті та співпраці, додатково інтегруючи робочі процеси розробки та операцій. Оскільки організації продовжують приймати практику мікросервісів та контейнеризації, очікується, що ці нові технології відіграють ключову роль у розвитку практик IaC.

Список використаної літератури

1. Infrastructure as Code: What Is It? Why Is It Important? URL: <https://www.hashicorp.com/resources/what-is-infrastructure-as-code>
2. What is Terraform? URL: <https://developer.hashicorp.com/terraform/intro>
3. J. Geerling. Ansible for DevOps: Server and configuration management for humans. 2nd edition. 2023. 469 p. URL: <https://www.ansiblefordevops.com/>