

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ СТВОРЕННЯ НАВЧАЛЬНОГО ЛАБОРАТОРНОГО СЕРЕДОВИЩА КОНТЕЙНЕРИЗАЦІЇ У ХМАРІ AWS

Вінницький національний технічний університет

Анотація

У роботі проаналізовано технології автоматизації розгортання інфраструктури в хмарному середовищі. Проведено порівняльний аналіз сучасних підходів до розгортання, вивчено можливості різних постачальників послуг та обрано оптимальну технологію програмного створення інфраструктури в хмарі. Розроблено програмне рішення з використанням хмарних технологій та мови конфігурації Hashicorp Configuration Language (HCL). Кількість функцій системи автоматизованого розгортання інфраструктури збільшилась на такі 3: декларативне управління мережевою топологією, інтеграція промислового балансування навантаження, автоматизація "нульового дня".

Ключові слова: IaC, Terraform, HCL, автоматизація, інфраструктура, хмарне середовище, управління інфраструктурою.

Abstract

The paper analyzes technologies for automating infrastructure deployment in a cloud environment. A comparative analysis of modern approaches to deployment is conducted, the capabilities of various service providers are studied, and the optimal technology for software infrastructure creation in the cloud is selected. A software solution is developed using cloud technologies and the Hashicorp Configuration Language (HCL) configuration language. The number of functions of the automated infrastructure deployment system has increased by the following 3: declarative network topology management, industrial load balancing integration, "zero-day" automation.

Keywords: IaC, Terraform, HCL, automation, infrastructure, cloud environment, infrastructure management.

Вступ

Інфраструктура як код (IaC) дозволяє керувати та підтримувати обчислювальну інфраструктуру через програмні скрипти, замість традиційних ручних налаштувань. Будь-яке програмне середовище потребує численних інфраструктурних компонентів, таких як операційні системи, з'єднання з базами даних та сховища даних. Ручне керування інфраструктурою забирає багато часу та піддається помилкам. Інфраструктура як код дозволяє визначити бажаний стан інфраструктури і забезпечити його автоматизоване управління. Це дозволяє розробникам зосередитися на створенні та вдосконаленні додатків замість керування середовищем. Компанії використовують IaC для зниження витрат, мінімізації ризиків і швидкого реагування на нові можливості. Робота має актуальність, тому що у сучасному світі важливість хмарних технологій стрімко зростає, оскільки вони забезпечують гнучкість, масштабованість та економічну ефективність, а контейнеризовані додатки пришвидшують процес розгортання та спрощують масштабування. Набуття практичних навичок з цими технологіями є необхідністю для спеціаліста у галузі інформаційних технологій.

Метою роботи є розширення функціональних можливостей засобів створення навчального лабораторного середовища контейнеризації у хмарі AWS за рахунок використання програмного коду (IaC), який керує цим процесом.

Результати досліджень

У роботі було розглянуто задачу створення навчального лабораторного середовища контейнеризації у хмарі AWS. В ході аналізу предметної області розглянуто основні аспекти роботи: проведено огляд та аналіз існуючих систем управління конфігурацією для створення навчально лабораторних середовищ [1], як локальних, так і браузерних, представлено основні поняття по темі, наведено переваги розробки додатків у контейнерах. Актуальним є розробка системи з тестовим середовищем, що на відміну від існуючих дасть можливість легко

модифікувати та розширювати систему та оновлювати її складові і компонентну архітектуру відповідно до появи нових технологій та методів розробки.

Було розроблено інформаційну технологію створення навчального лабораторного середовища контейнеризації у хмарі AWS у вигляді життєвого циклу розгортання інфраструктури [2] з використанням інструменту Terraform. У підрозділі про мову програмування HCL [3] було проаналізовано її переваги для опису інфраструктури. Виявлено, що HCL забезпечує декларативний підхід до визначення ресурсів, їх властивостей та залежностей, що робить код зрозумілим та легким для підтримки. Це сприяє підвищенню надійності та гнучкості інфраструктури, спрощуючи її створення та управління. Розроблено граф залежності ресурсів інфраструктури. Terraform автоматично будує граф залежностей, що дозволяє визначити правильний порядок створення, зміни та видалення ресурсів. Це забезпечує цілісність і консистентність інфраструктури, зменшуючи ризик помилок і гарантуючи правильне виконання всіх операцій. Детально досліджені команди `Init`, `plan`, `validate`, `apply` та `destroy`. Ці алгоритми забезпечують повний життєвий цикл управління інфраструктурою. Команда `Init` ініціалізує робочий каталог Terraform. `Plan` генерує план виконання змін, що дозволяє перевірити коректність майбутніх змін перед їх застосуванням. `Validate` перевіряє правильність конфігураційних файлів. `Apply` вносить зміни у реальну інфраструктуру, а `destroy` видаляє всі створені ресурси. Ці алгоритми підтвердили свою ефективність у забезпеченні надійного та контрольованого управління інфраструктурою. Спроектовано архітектуру навчального лабораторного середовища. Було розроблено діаграму та опис ресурсів, які використовуються в середовищі. Це середовище включає EC2 інстанси з Docker-compose та балансувальник навантаження, що забезпечує високу доступність та масштабованість. З результатів дослідження у розділі видно, що використання Terraform для автоматизації розгортання інфраструктури є ефективним та доцільним підходом, що дозволить швидко та ефективно розгорнути, оновити, використовувати та видалити необхідну інфраструктуру у хмарі.

Було розроблено модулі інфраструктури як коду (IaC) [4] для автоматизованого розгортання навчального лабораторного середовища. Налаштування середовища для розробки включало установку необхідних інструментів, програм, розширень, їх налаштування, а також створення оптимальних умов для роботи з Terraform і Docker. Розробка проекту полягала у створенні Terraform модулів для розгортання хмарних ресурсів в AWS. Проектування велося з урахуванням модульності та повторного використання коду. Розгортання та верифікація результатів розробки включала використання модулів для реального розгортання інфраструктури в AWS. Після розгортання проведено перевірку працездатності та ефективності інфраструктури шляхом звертання до адреси балансувальника навантаження по порту за замовчуванням. Результати підтвердили, що модулі забезпечують надійне функціонування всіх компонентів середовища, ідемпотентність інфраструктури та стабільність роботи, що свідчить про їх придатність для навчального процесу. Проведений аналіз демонструє, що розроблене автоматизоване лабораторне середовище є оптимальним рішенням для навчання контейнеризації в хмарі. Воно поєднує в собі простоту розгортання, характерну для PaaS-систем, з глибиною контролю та гнучкістю, притаманною IaaS-рішенням. Завдяки використанню Terraform система не лише спрощує процес підготовки до занять, але й сама по собі виступає потужним інструментом навчання сучасним практикам DevOps. Кількість функцій системи автоматизованого розгортання інфраструктури збільшилась на 3, що свідчить про те, що мета роботи досягнута - функціональні можливості засобів створення навчального лабораторного середовища контейнеризації у хмарі AWS розширені.

Висновки

У цій роботі запропонована інформаційна технологія створення навчального лабораторного середовища контейнеризації у хмарі AWS у вигляді життєвого циклу розгортання інфраструктури з використанням інструменту Terraform. Для опису інфраструктури було обрано мову HCL. Це сприяє підвищенню надійності та гнучкості інфраструктури, спрощуючи її створення та управління. З результатів дослідження видно, що використання Terraform для автоматизації розгортання інфраструктури є ефективним та доцільним підходом, що дозволить швидко та ефективно розгорнути, оновити, використовувати та видалити необхідну інфраструктуру у хмарі. Розробка проекту полягала у створенні Terraform модулів для розгортання хмарних ресурсів в AWS. Проектування велося з урахуванням модульності та повторного використання коду. Розгортання та верифікація результатів розробки включала використання модулів для реального

розгортання інфраструктури в AWS. Після розгортання проведено перевірку працездатності та ефективності інфраструктури шляхом звертання до адреси балансувальника навантаження по порту за замовчуванням. Результати підтвердили, що модулі забезпечують надійне функціонування всіх компонентів середовища, ідемпотентність інфраструктури та стабільність роботи, що свідчить про їх придатність для навчального процесу. Розроблене автоматизоване лабораторне середовище поєднує в собі простоту розгортання, характерну для PaaS-систем, з глибиною контролю та гнучкістю, притаманною IaaS-рішенням. Завдяки використанню Terraform система не лише спрощує процес підготовки до занять, але й сама по собі виступає потужним інструментом навчання сучасним практикам DevOps.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сенчик Р. С., Крилик Л. В. Аналіз передумов розробки програмного модуля системи безперервної поставки та розгортання коду для контейнеризованих додатків в AWS // Матеріали LIII науково-технічної конференції підрозділів ВНТУ, Вінниця, 20-22 березня 2024 р. Електрон. текст. дані. 2024. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20137>.
2. Фізична інфраструктура: сервери, сховища, мережі - Жаріков, Едуард. "Інфраструктура інформаційних систем. Курс лекцій." (2022). <https://ela.kpi.ua/server/api/core/bitstreams/e775b982-c399-4e33-9596-6a936f4e4ab6/content>
3. HCL. – [Електронний ресурс], Режим доступу - <https://github.com/hashicorp/hcl>.
4. Infrastructure as a code - Mustafa, Osama. "A Complete Guide to DevOps with AWS." Apress, Berkeley, CA (2023).

Сенчик Роман Сергійович — студент групи 1КН-24м, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, email: rsenchyk5@gmail.com

Колесницький Олег Костянтинович — канд. техн. наук, професор кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця. email: kolesnytskyi@vntu.edu.ua

Паночийшин Юрій Миколайович – кандидат технічних наук, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, e-mail: y.panochyshyn@vntu.edu.ua

Senchik Roman S. – student of Intelligent Information Technologies and Automation Department, Vinnytsia National Technical University, Vinnytsia, e-mail: rsenchyk5@gmail.com

Kolesnytskyj Oleh K. – Cand. Sc. (Eng.), Professor of the Computer Science Chair, Vinnytsia National Technical University, Vinnytsia. email: kolesnytskyi@vntu.edu.ua

Panochyshyn Yuriy M. – Candidate of Technical Sciences, Associate Professor of the Department of Computer Science, Vinnytsia National Technical University, Vinnytsia, e-mail: y.panochyshyn@vntu.edu.ua