

Інформаційні технології

УДК 004.852

Волошин Владислав Сергійович

бакалавр комп'ютерних наук

Національного технічного університету України

«Київський політехнічний інститут»

Волошин Владислав Сергеевич

бакалавр компьютерных наук

Национального технического университета Украины

«Киевский политехнический институт»

Voloshyn V.

Bachelor of computer science

The National Technical University of Ukraine

«Kyiv Polytechnic Institute»

**ВИКОРИСТАННЯ ПРИНЦИПУ «INFRASTRUCTURE AS CODE» У
ХМАРНИХ СЕРЕДОВИЩАХ**

**ИСПОЛЬЗОВАНИЕ ПРИНЦИПА «INFRASTRUCTURE AS CODE» В
ОБЛАЧНЫХ СРЕДАХ**

«INFRASTRUCTURE AS CODE» PRINCIPLE IN CLOUD SYSTEMS

Анотація: розглянуто автоматизацію керування хмарною інфраструктурою шляхом використання IaC, тобто написання коду, що описує систему.

Ключові слова: хмарні обчислення, IaC, автоматизація, Terraform.

Аннотация: рассмотрено автоматизацию управления облачной инфраструктурой путём использования IaC, то есть написания кода, который описывает систему.

Ключевые слова: облачные вычисления, IaC, автоматизация, Terraform.

Abstract: automation of management cloud infrastructure was investigated by using IaC, writing code that describes system.

Keywords: cloud computing, IaC, automation, Terraform.

Хмарні платформи стають все більш популярні. Більшість компаній переводять свою інфраструктуру з власних серверних парків до хмарних провайдерів. Дане рішення дозволяє позбутися проблем підтримки та модернізації серверного і мережевого обладнання за відносно невелику ціну, гарантуючи при цьому майже стовідсоткову доступність. Проте, такі системи налічують десятки або навіть сотні тисяч одиниць техніки.

При роботі з великою кількістю даних ускладнюється процес її керування. Великі масиви даних неможливо опрацьовувати вручну. Крім того, з'являється необхідність у постійній підтримці робочого стану всієї системи. Адже зміна однієї частини інфраструктури неодмінно веде до зміни в іншій, що порушує надійність системи. Задля вирішення даної проблеми використовується принцип «Infrastructure as Code» (IaC), який дозволяє описати конфігурацію та стан керованої системи у вигляді коду.

Замість того, щоб вручну налаштовувати віртуальну машину, або створювати мережу чи що інше в хмарному середовищі, ви пишете конфігураційний файл, який описує стан необхідної вам інфраструктури. Цей файл має зрозумілий людині синтаксис. Коли ви забажаєте змінити систему, все що необхідно буде зробити – просто змінити цей файл та підтвердити зміни. Тим самим ви здатні легко модернізувати систему, змінивши декілька рядків файлу. Всю іншу роботу зробить за вас IaC

програма. До того ж відтепер систему можна відтворити всього лиш скопіювавши кілька файлів. Це дозволяє позбутися від зайвих труднощів у налаштуванні системи та зорієнтуватися на самому продукті та презентування його кінцевому клієнту.

Принцип роботи цих IaC програм дуже простий. Автоматизація лежить в основі всього, починаючи від нарощування інфраструктури і закінчуючи покриттям написаного коду тестами, що допоможе уникнути небажаних помилок. Програма має певну кількість інтеграцій з різними хмарними платформами та системами, що спілкуються за допомогою API запитів. Викликають API запити команди, що знаходяться в конфігураційному файлі. Переносячи керування інфраструктурою в текстові файли, ми тим самим можемо зберігати код в системах контролю версій та за необхідністю повертатися до попередніх версій конфігурацій, якщо щось піде не так.

Яскравим прикладом такої програми є Terraform. Розроблений компанією Hashicorp, цей продукт є на сьогоднішній день одним із лідерів у своїй сфері. Terraform здатний керувати як низькорівневими компонентами, такими як обчислювальні потужності, пам'ять, мережеві налаштування сервера, так і високорівневими – DNS записи, різноманітні SaaS платформи та багато іншого.

Існують також аналоги, які орієнтовані на конкретні хмарні рішення, як наприклад CloudFormation та OpsWorks в Amazon Web Services [1]. AWS CloudFormation дозволяє розробникам та системним адміністраторам легко створювати групи взаємопов'язаних ресурсів AWS та керувати ними, забезпечуючи передбачене розподілення та оновлення в правильному порядку. Таким чином можна використовувати готові шаблони сервісу або створювати власні. AWS OpsWorks – це сервіс керування конфігурацією на базі Chef, платформи автоматизації, яка розглядає конфігурацію серверів у вигляді коду. OpsWorks використовує Chef для автоматизації конфігурації,

розгортання та керування серверами на віртуальних машинах Amazon Elastic Compute Cloud (Amazon EC2) чи в локальних обчислювальних середовищах.

В якості доповнення до IaC має місце також таке поняття як «Policy as Code» (PaC). Цей принцип забезпечує доступ до конфігурації ресурсів, можливість їх аудиту та оцінки. За допомогою безперервного моніторингу сервіс дає можливість відслідковувати зміни в конфігураціях та зв'язках між ресурсами, проглядати детальну історію змін конфігурації ресурсу і визначати загальну відповідність системи поставленим вимогам. Маючи журнал змін в системі, можна швидко визначити проблеми ще до їх виникнення. Крім того, сукупність політик та постійний моніторинг підвищують загальну безпеку системи.

Ще однією з переваг даного принципу керування системою є те, що, окрім аналізу необхідних метрик та своєчасного сповіщення, PaC здатен обмежити використання ресурсів системою, не дозволяючи зростати ціні за послуги у хмарного провайдера. Прикладом може слугувати ситуація з автомасштабуванням продукту. Коли попит на продукт значно підвищується, необхідно збільшити обчислювальні потужності шляхом додавання віртуальних машин. Суть масштабування полягає в тому, аби реагувати на збільшення або зменшення кількості користувачів та відповідно підлаштовувати необхідну кількість ресурсів для цього. Виставлені політики будуть контролювати мінімальну та максимальну кількість допустимих машин, що можуть працювати одночасно.

Отже, «Infrastructure as Code» та «Policy as Code» методики виводять рішення інфраструктурних задач на новий рівень. Вони є достатньо новими, тому потребують ще детального аналізу. Проте, на сьогоднішній день все більше компаній та розробників застосовують їх у розробках своїх продуктів. Ці методики економлять час та гроші, потрібні на встановлення та налаштування системи.

Література

1. Інструментальні та інфраструктурні ресурси для практиків DevOps – Режим доступу: <https://aws.amazon.com/ru/devops/>
2. Запис та оцінка конфігурації ресурсів AWS – Режим доступу: <https://aws.amazon.com/ru/config/>
3. Infrastructure As Code – Режим доступу: <https://martinfowler.com/bliki/InfrastructureAsCode.html>
4. Infrastructure as Code: A reason to smile – Режим доступу: <https://www.thoughtworks.com/insights/blog/infrastructure-code-reason-smile>
5. Introduction to Terraform – Режим доступу: <https://www.terraform.io/intro/index.html>
6. Modern approach to managing infrastructure – Режим доступу: <https://puppet.com/solutions/infrastructure-as-code>