

ТЕХНОЛОГІЇ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ СЕРВЕРАМИ

У сучасному світі, де ефективність і надійність ІТ-інфраструктури є критично важливими для бізнесу, технології автоматизованого управління серверами набувають особливої актуальності. Постійне зростання обсягів даних, вимоги до безперервної роботи систем та необхідність оптимізації ресурсів ускладнюють ручне адміністрування серверів. Відповідно, розвиток інструментів автоматизації та їх впровадження стає ключовим фактором підвищення продуктивності й безпеки серверних середовищ.

Інфраструктура як код[1] (IaC) - це метод визначення та управління інфраструктурою вашої системи за допомогою коду. Подібно до того, як програмний код визначає логіку та функціональність програми, IaC описує архітектуру системи, включаючи такі компоненти, як сервери, мережі, сховища та операційні системи.

За допомогою IaC можна керувати ресурсами інфраструктури у послідовний та організований спосіб. Він розглядає конфігураційні файли як вихідний код, що дозволяє використовувати контроль версій, відстежувати помилки та легко вносити оновлення. Таким чином, за допомогою таких систем контролю версій можна відстежувати всі внесені зміни та ефективно співпрацювати з розробниками.

Для полегшення роботи з методом IaC розроблено спеціалізовані програмні засоби автоматизованого управління серверами, наприклад Terraform [2] та Ansible [3].

Було проведено порівняльний аналіз зазначених інструментів за наступними критеріями:

1. Синтаксис – формат і стиль запису конфігурацій, який використовується в інструменті.
2. Підхід – спосіб управління інфраструктурою (декларативний або імперативний підхід).
3. Складність – рівень складності в освоєнні, налаштуванні та підтримці, включаючи навчальний поріг і вимоги до знань користувача.
4. Використання – основні сценарії застосування інструмента.
5. Недоліки – обмеження та слабкі сторони інструмента, які можуть ускладнити його використання.

Таблиця 1

Порівняння інструментів

Критерії	Terraform	Ansible
Синтаксис	HCL або JSON	YAML
Використання	Розгортання інфраструктури з нуля, керування ресурсами у хмарі	Налаштування ПЗ, оновлення пакетів, управління користувачами, конфігурування сервісів
Підхід	Декларативний	Декларативний і імперативний
Складність	Вимагає попереднього планування та визначення всіх ресурсів	Гнучкіший, легко адаптується під різні завдання
Недоліки	Простий у використанні, але для освоєння просунутих функцій і найкращих практик потрібен час, також підтримує не всі ресурси й хмарні сервіси, інколи потрібні сторонні провайдери.	Має обмежену підтримку Windows, може втрачати продуктивність при масштабуванні, та має слабкі можливості звітування, що ускладнює налагодження

Тим не менш, вибір між Terraform та Ansible зазвичай залежить від конкретного випадку використання та інфраструктурних потреб компанії. Ansible краще підходить для налаштування та автоматизації діяльності на окремих вузлах, а Terraform - для створення, забезпечення та підтримки хмарної інфраструктури.

Список використаних джерел:

1. What is Infrastructure as Code (IaC)? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/what-is-infrastructure-as-code-iac/> (дата звернення 20.02.2025)
2. Terraform [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/hashicorp/terraform> (дата звернення 20.02.2025)
3. Ansible [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/ansible/ansible> (дата звернення 20.02.2025)