

УДК 004.052:004.45:004.75

DOI <https://doi.org/10.32782/tnv-tech.2025.2.13>

АВТОМАТИЗАЦІЯ РОЗГОРТАННЯ ТА УПРАВЛІННЯ ОПЕРАЦІЙНИХ СИСТЕМ У ХМАРНИХ ТА ГІБРИДНИХ СЕРЕДОВИЩАХ ЗАСОБАМИ СИСТЕМНОГО АДМІНІСТРУВАННЯ

Переверзєв А. М. – старший викладач кафедри комп'ютерних наук та програмної інженерії Приватного вищого навчального закладу «Європейський університет»
ORCID ID: 0000-0003-0778-9203

Подвиженко А. В. – старший викладач кафедри комп'ютерних наук та програмної інженерії Приватного вищого навчального закладу «Європейський університет»
ORCID ID: 0000-0003-1826-7566

Колодінська Я. О. – старший викладач кафедри математичних дисциплін та інноваційного проектування Приватного вищого навчального закладу «Європейський університет»
ORCID ID: 0000-0002-3330-7565
Scopus-Author ID: 57466561800

Склярєнко О. А. – аспірант Приватного вищого навчального закладу «Європейський університет»
ORCID ID: 0009-0003-4908-0187

Стаття присвячена актуальній проблемі автоматизації процесів розгортання та управління операційними системами (ОС) в умовах сучасних IT-інфраструктур, що характеризуються зростаючою складністю через використання хмарних та гібридних моделей. Проаналізовано недоліки традиційних ручних підходів до адміністрування ОС, зокрема низьку швидкість, труднощі масштабування, ризики виникнення неконсистентності конфігурацій (дрейф конфігурацій) та людських помилок, що негативно впливає на стабільність, безпеку та гнучкість IT-середовища. Обґрунтовано критичну необхідність переходу до автоматизованих рішень на основі сучасних засобів системного адміністрування.

В роботі розглянуто ключові технології та методології автоматизації: Інфраструктура як код (IaC) для декларативного опису та версіонування інфраструктури (з прикладами інструментів типу Terraform); Системи управління конфігураціями (Ansible, Puppet) для забезпечення бажаного стану ОС та ПЗ; автоматизоване створення оптимізованих образів ОС («image baking» за допомогою Packer); а також роль контейнеризації та оркестрації (Docker, Kubernetes) у контексті управління залежностями на рівні ОС.

Описано комплексні переваги автоматизації, включаючи значне підвищення операційної ефективності, зниження ризиків, прискорення циклів розробки та впровадження (DevOps), покращення керованості та відповідності стандартам безпеки. Визначено типові етапи та компоненти автоматизованого життєвого циклу ОС у гетерогенних середовищах. Зроблено висновок про те, що автоматизація є фундаментальним елементом ефективного системного адміністрування в хмарних та гібридних середовищах, що дозволяє компаніям швидше реагувати на бізнес-вимоги та підтримувати надійну й безпечну інфраструктуру.

Ключові слова: автоматизація, операційна система, розгортання ОС, управління ОС, хмарні обчислення, гібридна хмара, системне адміністрування, інфраструктура як код (IaC), управління конфігураціями, DevOps, Terraform, Ansible, Kubernetes, Docker.

Pereverziev A. M., Podvizhenko A. V., Kolodinska Ya. O., Skliarenko O. A. Automation of operating system deployment and management in cloud and hybrid environments using system administration tools

The article addresses the current challenge of automating the deployment and management processes of operating systems (OS) within modern IT infrastructures, which are characterized by growing complexity due to the adoption of cloud and hybrid models. It analyzes the drawbacks of traditional manual approaches to OS administration, including low speed, scaling difficulties, the risks of configuration inconsistencies (configuration drift), and human errors, which negatively impact the stability, security, and flexibility of the IT environment. The critical need to transition to automated solutions based on modern system administration tools is substantiated.

The paper discusses key automation technologies and methodologies: Infrastructure as Code (IaC) for declarative description and versioning of infrastructure (with examples of tools like Terraform); Configuration management systems (Ansible, Puppet) for ensuring the desired state of OS and software; automated creation of optimized OS images («image baking» using Packer); as well as the role of containerization and orchestration (Docker, Kubernetes) in the context of managing OS-level dependencies.

The comprehensive benefits of automation are described, including significant improvement in operational efficiency, risk reduction, acceleration of development and deployment cycles (DevOps), improved manageability, and compliance with security standards. Typical stages and components of the automated OS lifecycle in heterogeneous environments are identified. It is concluded that automation is a fundamental element of effective system administration in cloud and hybrid environments, allowing companies to respond more quickly to business requirements and maintain a reliable and secure infrastructure.

Key words: automation, operating system, OS deployment, OS management, cloud computing, hybrid cloud, system administration, Infrastructure as Code (IaC), configuration management, DevOps, Terraform, Ansible, Kubernetes, Docker.

Постановка проблеми. Стрімкий розвиток хмарних технологій та зростання популярності гібридних інфраструктурних моделей призвели до значного ускладнення ІТ-ландшафтів сучасних організацій. Управління великою кількістю віртуальних машин, контейнерів та сервісів, розподілених між приватними та публічними хмарами, а також локальними дата-центрами, ставить перед відділами системного адміністрування нові виклики.

У сучасних динамічних та масштабованих середовищах традиційні ручні методи розгортання й налаштування операційних систем стають неефективними та надто трудомісткими. Ручне адміністрування пов'язане з кількома суттєвими проблемами: воно значно сповільнює процес впровадження нових сервісів та оновлень через низьку швидкість виконання операцій. Крім того, людський фактор неминуче спричиняє неконсистентність у конфігураціях різних систем, що ускладнює управління, пошук несправностей та забезпечення належного рівня безпеки. Також ручне управління погано масштабується при збільшенні кількості серверів та сервісів і несе високий ризик помилок, які можуть призвести до збоїв, порушень безпеки та фінансових втрат. Зрештою, значні затрати часу системних адміністраторів на рутинні завдання підвищують загальні операційні витрати та вартість володіння інфраструктурою.

Таким чином, виникає гостра потреба в автоматизації процесів розгортання та управління операційними системами, що дозволить підвищити ефективність, надійність, безпеку та керованість ІТ-інфраструктури в хмарних та гібридних середовищах.

Аналіз останніх досліджень і публікацій. Проблематика автоматизації управління ІТ-інфраструктурою активно досліджується в науковій літературі та практичних публікаціях протягом останнього десятиліття. Значний внесок у розвиток концепцій та інструментів автоматизації зробили практики DevOps [1]. Ключовими напрямками досліджень є розробка та застосування підходів «Інфраструктура

як код» (Infrastructure as Code, IaC) та систем управління конфігураціями (Configuration Management).

Концепція IaC, описана в роботах таких авторів, як Морріс та Шафер [3], передбачає управління інфраструктурою (включаючи сервери, мережі, сховища та конфігурації ОС) за допомогою декларативних або скриптових файлів конфігурації, що зберігаються в системах контролю версій. Це дозволяє версіонувати інфраструктуру, відтворювати середовища та автоматизувати їх створення та оновлення.

Системи управління конфігураціями, такі як Ansible, Puppet, Chef та SaltStack, досліджуються у контексті забезпечення консистентності стану ОС та встановленого програмного забезпечення на великій кількості серверів [4, 6]. Ці інструменти дозволяють централізовано описувати бажаний стан системи та автоматично приводити її у відповідність до цього стану.

Питання автоматизації в хмарних середовищах розглядаються у публікаціях, присвячених специфічним інструментам хмарних провайдерів (AWS CloudFormation, Azure Resource Manager, Google Cloud Deployment Manager) та універсальним інструментам, таким як Terraform [3, 5]. Досліджується інтеграція цих інструментів у CI/CD пайплайни для повної автоматизації життєвого циклу додатків та інфраструктури [2].

Також важливим напрямком є дослідження застосування контейнеризації (Docker) та оркестрації контейнерів (Kubernetes) [5], які хоч і абстрагуються від традиційного управління ОС на рівні хоста, але вимагають автоматизованого підходу до розгортання та управління базовими ОС на вузлах кластера.

Незважаючи на значну кількість публікацій, питання комплексної автоматизації саме розгортання та управління життєвим циклом ОС в умовах гетерогенних гібридних середовищ, враховуючи інтеграцію різних інструментів та підходів, потребує подальшого дослідження та систематизації.

Виклад основного матеріалу. Автоматизація розгортання та управління ОС у хмарних та гібридних середовищах є багатогранним процесом, що охоплює кілька ключових технологій та підходів.

Основою автоматизації інфраструктури часто слугує підхід «Інфраструктура як Код» (IaC), який дозволяє визначати всі компоненти, такі як віртуальні машини, мережеві налаштування, балансувальники навантаження та правила безпеки, за допомогою коду.

Для реалізації IaC використовуються різноманітні інструменти. Популярним вибором є Terraform, інструмент з відкритим кодом, що підтримує багатьох хмарних провайдерів та локальні інфраструктури, наприклад, VMware чи OpenStack, і дозволяє декларативно описувати бажаний стан інфраструктури. Окрім Terraform, існують платформи-специфічні інструменти, такі як AWS CloudFormation, Azure Resource Manager (ARM) Templates та Google Cloud Deployment Manager, які забезпечують тісну інтеграцію з екосистемами відповідних хмарних провайдерів [7].

Процес роботи з IaC полягає в тому, що розробники або системні адміністратори створюють конфігураційні файли, наприклад, мовою HCL для Terraform або у форматах JSON/YAML для хмарних інструментів. У цих файлах детально описуються необхідні ресурси, включаючи параметри віртуальних машин, такі як тип інстансу, образ операційної системи та мережеві інтерфейси. Для забезпечення версіонування, можливості співпраці та аудиту змін, ці файли зберігаються в системі контролю версій, наприклад, Git. При виконанні коду відповідний інструмент IaC взаємодіє з API хмарного провайдера або гіпервізора, щоб створити або

модифікувати інфраструктуру згідно з описом у конфігураційних файлах. Це охоплює і початкове створення віртуальної машини з обраним базовим образом операційної системи [8].

Створення образів ОС, відоме як «Image Baking», є ефективним підходом, що замінює тривале налаштування після розгортання стандартного образу операційної системи. Цей метод полягає у попередньому створенні кастомних «золотих» образів (Golden Images) або «машинних образів» (Machine Images), які вже включають необхідну базову конфігурацію ОС, встановлені оновлення безпеки, стандартне програмне забезпечення та агентів моніторингу чи управління [14].

Для реалізації цього процесу використовуються спеціалізовані інструменти, такі як Packer від HashiCorp, AWS EC2 Image Builder або Azure Image Builder. За допомогою цих інструментів спочатку визначається базовий образ ОС. Потім до нього застосовуються скрипти налаштування – це можуть бути Shell-скрипти, Ansible плейбуки або Chef рецепти – які встановлюють та конфігурують потрібне програмне забезпечення. В результаті цього процесу створюється новий, повністю налаштований образ. Цей готовий образ надалі може бути використаний інструментами інфраструктури як коду (IaC) для дуже швидкого розгортання вже попередньо конфігурованих екземплярів операційної системи.

Після того, як операційна система розгорнута за допомогою інфраструктури як коду (IaC) або з використанням заздалегідь підготовленого образу, виникає потреба підтримувати її конфігурацію в актуальному та послідовному стані протягом усього її життєвого циклу. Саме для цього застосовуються системи управління конфігураціями [8, 9].

Серед популярних інструментів для управління конфігураціями виділяють Ansible, Chef, Puppet та SaltStack. Ці інструменти працюють за різними принципами. Більшість із них, як-от Puppet, Chef та SaltStack, використовують декларативний підхід. Це означає, що адміністратор описує бажаний кінцевий стан системи, наприклад, вказуючи, що «веб-сервер nginx має бути встановлений та запущений» або «користувач admin повинен існувати з певним SSH-ключем». Інструмент самостійно визначає необхідні кроки для досягнення цього описаного стану. На противагу цьому, Ansible використовує процедурний або імперативний підхід, де описуються конкретні кроки чи завдання, які потрібно послідовно виконати для досягнення мети. Ansible часто вважають простішим для освоєння, зокрема завдяки його без-агентній архітектурі, яка для взаємодії з керованими системами використовує стандартний протокол SSH [4].

Процес управління конфігураціями включає створення спеціальних файлів системними адміністраторами: «плейбуків» в Ansible, «рецептів» у Chef, «маніфестів» у Puppet або «стейтів» у SaltStack. Ці файли детально описують бажану конфігурацію операційної системи, процеси встановлення програмних пакетів, управління системними службами, файлами та обліковими записами користувачів. Важливо, що ці конфігураційні файли зберігаються в системах контролю версій, що дозволяє відстежувати зміни та повертатися до попередніх станів. Інструмент управління конфігураціями регулярно, або за вимогою адміністратора, підключається до керованих вузлів (серверів) і застосовує описані конфігурації, тим самим забезпечуючи їхню відповідність заданим політикам та підтримуючи консистентність у всій інфраструктурі [4, 10, 11].

Інтеграція автоматизації розгортання та управління операційними системами в загальний конвеєр безперервної інтеграції та безперервної доставки/розгортання (CI/CD) є найефективнішим підходом. У такому процесі будь-яка зміна

в кодї інфраструктури (IaC) або конфігурації, збережена в системі контролю версій, такий як Git, автоматично ініціює пайплайн CI/CD. Цей пайплайн, керований інструментами на кшталт Jenkins, GitLab CI, GitHub Actions або Azure DevOps, може послідовно виконувати етапи валідації коду, тестування, створення образу ОС (при використанні підходу «image baking»), розгортання інфраструктури за допомогою відповідного IaC-інструменту та застосування необхідних конфігурацій через інструменти управління конфігураціями [2].

Автоматизація набуває особливого значення в гібридних середовищах, які поєднують ресурси публічної хмари та приватної інфраструктури, наприклад, локального дата-центру чи приватної хмари. Управління такими середовищами вимагає використання інструментів, здатних взаємодіяти з різноманітними платформами. Terraform та Ansible є популярними універсальними рішеннями для гібридних сценаріїв, оскільки вони підтримують широкий спектр «провайдерів» та «модулів» для роботи з основними хмарними платформами (AWS, Azure, GCP) та локальними системами, такими як VMware vSphere, OpenStack, або навіть фізичними серверами через SSH. Крім того, інструменти оркестрації, як-от Kubernetes, розгорнутий у гібридному режимі за допомогою рішень типу Anthos, Azure Arc чи Rancher, можуть надати єдину платформу для управління контейнеризованими додатками. Це дозволяє абстрагуватися від базової інфраструктури, хоча все одно потребує автоматизованого управління операційними системами на вузлах самого кластера.

Автоматизація охоплює також сфери безпеки та моніторингу. У контексті безпеки, підхід «Security as Code» передбачає автоматичне сканування образів операційних систем на наявність вразливостей, а також застосування політик безпеки безпосередньо під час процесів розгортання та конфігурації систем. Аналогічно, «Monitoring as Code» дозволяє автоматизувати налаштування агентів моніторингу, таких як Prometheus node exporter чи Datadog agent, разом із конфігурацією відповідних дашбордів та систем сповіщень [4, 13].

Висновки. Автоматизація розгортання та управління операційними системами за допомогою сучасних засобів системного адміністрування є не просто бажаною, а необхідною умовою для ефективного функціонування ІТ-інфраструктури в динамічних хмарних та гібридних середовищах.

Впровадження підходів «Інфраструктура як код», систем управління конфігураціями та автоматизованого створення образів ОС дозволяє вирішити ключові проблеми ручного адміністрування: підвищити швидкість розгортання, забезпечити консистентність конфігурацій, покращити масштабованість, знизити ризик людських помилок та оптимізувати операційні витрати.

Використання таких інструментів, як Terraform, Ansible, Packer, Kubernetes, інтегрованих у CI/CD процеси, дає змогу створити надійні, відтворювані та керовані ІТ-середовища. Це, в свою чергу, прискорює інновації, підвищує стабільність сервісів та посилює безпеку інфраструктури.

Незважаючи на переваги, впровадження комплексної автоматизації вимагає відповідної кваліфікації персоналу, зміни культури в ІТ-відділах (у напрямку DevOps) та ретельного планування архітектури автоматизованих процесів.

Перспективи подальшого розвитку полягають у глибшій інтеграції інструментів, застосуванні штучного інтелекту та машинного навчання для предиктивного аналізу та самовідновлення систем (AIOps), а також у розвитку безсерверних підходів, які ще більше абстрагують управління ОС від кінцевого користувача. Однак, на сьогоднішній день, автоматизація управління саме ОС залишається критично важливим елементом для більшості організацій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Проект «Фенікс». Роман про IT, DevOps та успіх у бізнесі / Кім Дж. та ін. / пер. з англ. Київ : Фабула, 2019. 432 с.
2. Коваленко В. І., Петренко С. М. Інтеграція засобів автоматизації інфраструктури в конвеєри CI/CD. *Проблеми програмування*. 2023. № 1. С. 58–67.
3. Kief Morris. *Infrastructure as Code: Managing Infrastructure with Configuration Management Tools*. Foreword. 2nd ed. O'Reilly Media, 2020. 438 p.
4. Петров А. В. Використання Ansible для автоматизації управління конфігураціями Linux-систем. *Вісник Національного технічного університету «ХПІ»*. Серія: *Нові рішення в сучасних технологіях*. 2022. № 2(10). С. 45–51.
5. Сидоренко М. О. Порівняльний аналіз інструментів оркестрації контейнерів у хмарних середовищах. Матеріали VI Міжнар. наук.-практ. конф. «*Інформаційні технології та комп'ютерна інженерія*», м. Вінниця, 20–22 жовтня 2024 р. Вінниця : ВНТУ, 2024. С. 150–152.
6. Шевченко О. П. Сучасні підходи до управління IT-інфраструктурою в гібридних хмарах. Київ : Наукова думка, 2022. 315 с.
7. Using Terraform as an IaC tool for the AWS Cloud. *AWS Prescriptive Guidance* : веб-сайт. URL: <https://docs.aws.amazon.com/prescriptive-guidance/latest/choose-iac-tool/terraform.html> (дата звернення 23.04.2025).
8. EC2 Image Builder Documentation. *User Guide EC2 Image Builder* : веб-сайт. URL: <https://docs.aws.amazon.com/imagebuilder/latest/userguide/what-is-image-builder.html> (дата звернення 23.04.2025).
9. Azure VM Image Builder overview. *Microsoft* : веб-сайт. URL: <https://learn.microsoft.com/en-us/azure/virtual-machines/image-builder-overview?tabs=azure-powershell> (дата звернення 23.04.2025).
10. Невзоров А. В., Скляренко О. В., Колодінська Я. О., Яровий Р. О. Особливості аналітичного забезпечення експлуатації інформаційних систем та обладнання в сучасних умовах. *Журнал «Прикладні питання математичного моделювання»*. 2023. Т. 6, № 1. С. 117–123.
11. A Complete Guide to Automation and Deployment. *Scalecomputing* : веб-сайт. URL: <https://www.scalecomputing.com/resources/what-is-ansible> (дата звернення 23.04.2025).
12. Azure Arc hybrid management and deployment for Kubernetes clusters. *Microsoft* : веб-сайт. URL: <https://learn.microsoft.com/en-us/azure/architecture/hybrid/arc-hybrid-kubernetes> (дата звернення 23.04.2025) [in English].
13. Міграція у хмару на прикладі сервісів VMware. *Wiseit* : веб-сайт. URL: <https://wiseit.com.ua/migraciya-u-hmaru-priklad-vmware/> (дата звернення 23.04.2025).
14. Golden image. *RedHat* : веб-сайт. URL: <https://www.redhat.com/en/topics/linux/what-is-a-golden-image> (дата звернення 23.04.2025).

REFERENCES:

1. Gene Kim, Jez Humble, Patrick Debois & John Willis. (2019). *Proekt "Feniks". Roman pro IT, DevOps ta uspikh u biznesi [The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win]*. Kyiv : Fabula [in Ukrainian].
2. Kovalenko, V. I. & Petrenko, S. M. (2023). Intehratsiya zasobiv avtomatyzatsiyi infrastruktury v konveyery CI/CD [Integrating infrastructure automation tools into CI/CD pipelines]. *Problemy prohramuvannya – Programming problems*, 1, 58–67 [in Ukrainian].
3. Kief Morris. (2020). *Infrastructure as Code: Managing Infrastructure with Configuration Management Tools*. Foreword. 2nd ed. O'Reilly Media.
4. Petrov, A. V. (2022). Vykorystannya Ansible dlya avtomatyzatsiyi upravlinnya konfihuratsiyamy Linux-system [Using Ansible to automate configuration management of Linux systems]. *Visnyk Natsional'noho tekhnichnoho universytetu "KHPI". Seriya:*

Novi rishennya v suchasnykh tekhnolohiyakh – Bulletin of the National Technical University “KhPI”. Series: New solutions in modern technologies, 2(10), 45–51 [in Ukrainian].

5. Sydorenko, M. O. (2024). Porivnyal'nyy analiz instrumentiv orkestratsiyi konteyneriv u khmarnykh seredovyschakh [Comparative analysis of container orchestration tools in cloud environments]. Proceedings from: *VI Mizhnarodna naukovo-praktychna konferentsiia “Informatsiyini tekhnolohiyi ta komp'yuterna inzheneriya” – The Sixth International Scientific-Practical Conference “Information Technologies and Computer Engineering”*. (pp. 150-152). Vinnytsia: VNTU [in Ukrainian].

6. Shevchenko, O. P (2022). *Suchasni pidkhody do upravlinnya IT-infrastrukturoyu v hibrydnykh khmarak* [Modern approaches to managing IT infrastructure in hybrid clouds]. Kyiv : Naukova dumka [in Ukrainian].

7. Using Terraform as an IaC tool for the AWS Cloud. (n.d.). *AWS Prescriptive Guidance*. Retrieved from: <https://docs.aws.amazon.com/prescriptive-guidance/latest/choose-iac-tool/terraform.html> [in English].

8. EC2 Image Builder Documentation. (n.d.). *User Guide EC2 Image Builder*. Retrieved from: <https://docs.aws.amazon.com/imagebuilder/latest/userguide/what-is-image-builder.html> [in English].

9. Azure VM Image Builder overview. (n.d.). *Microsoft*. Retrieved from: <https://learn.microsoft.com/en-us/azure/virtual-machines/image-builder-overview?tabs=azure-powershell> [in English].

10. Nevzorov, A. V., Skliarenko, O. V., Kolodinska, Y. O., & Yarovyi, R. O. (2023). Osoblyvosti analitychnoho zabezpechennia ekspluatatsii informatsiinykh system ta obladnannia v suchasnykh umovakh [Peculiarities of analytical support for the operation of information systems and equipment in modern conditions]. *Prykladni pytannia matematychnoho modeliuвання – Applied Problems of Mathematical Modelling*, 6(1), 117–123 [in Ukrainian].

11. A Complete Guide to Automation and Deployment. (n.d.). *Scalecomputing*. URL: <https://www.scalecomputing.com/resources/what-is-ansible> [in English].

12. Azure Arc hybrid management and deployment for Kubernetes clusters. (n.d.). *Microsoft*. Retrieved from: <https://learn.microsoft.com/en-us/azure/architecture/hybrid/arc-hybrid-kubernetes> [in English].

13. Mihratsiya u khmaru na prykladi servisiv VMware [Cloud migration using VMware services as an example]. (n.d.). *Wiseit*. Retrieved from: <https://wiseit.com.ua/migracziya-u-hmaru-pryklad-vmware/> [in Ukrainian].

14. Golden image. (n.d.). *RedHat*. Retrieved from: <https://www.redhat.com/en/topics/linux/what-is-a-golden-image> [in English].