

АНАЛІЗ СПОСОБІВ ВЗАЄМОДІЇ З VIRTUALBOX ДЛЯ АВТОМАТИЗАЦІЇ ІНФРАСТРУКТУРИ

Вадим Матюхін, Віталій Булатецький

ANALYSIS OF VIRTUALBOX INTERACTION METHODS FOR INFRASTRUCTURE AUTOMATION

Vadym Matiukhin, Vitalii Bulatetskyi

Волинський національний університет імені Лесі Українки, просп. Волі, 13, Луцьк, 43025, Україна

Abstract. *Infrastructure as Code (IaC) has become a leading methodology in modern server infrastructure management. It enables automated deployment and consistent configuration of server environments by representing infrastructure definitions as code. This approach allows administrators to leverage version control, change tracking, code review, and CI/CD practices, resulting in more reliable and scalable systems. However, effective implementation of IaC requires programmable access to the underlying infrastructure via application programming interfaces (APIs). This paper focuses on the virtualization engine VirtualBox, which forms the core of our faculty's infrastructure. Although VirtualBox provides several APIs—such as COM/XPCOM, a command-line interface, and an web service—its integration with modern IaC tools remains limited. The objective of this research is to explore these interfaces and design an object-oriented C# wrapper that enables seamless use of VirtualBox functionality within different virtual environment management tools.*

Упродовж останніх років одним із провідних трендів у сфері управління серверною інфраструктурою стала методологія Infrastructure as Code (IaC) [1]. Цей підхід кардинально змінює спосіб організації та розгортання серверних систем, оскільки дозволяє описувати інфраструктуру як програмний код. Завдяки цьому конфігурації стають відтворюваними, контрольованими у системах керування версіями, доступними для рев'ю, тестування та автоматизованого розгортання, а також легко інтегруються в CI/CD-пайплайни.

Втім, для ефективного впровадження IaC-підходу необхідною умовою є наявність відкритого програмного інтерфейсу – API – що забезпечує керованість зовнішніми інструментами. У випадку інфраструктури, яка базується на рушії віртуалізації Oracle VirtualBox, доступними є кілька варіантів таких інтерфейсів: COM/XPCOM API, CLI (VBoxManage) та експериментальний web-сервер з XML API [2].

VirtualBox не має готових інструментів для інтеграції з IaC-платформами, такими як Terraform чи Ansible. Тому актуальною є розробка проміжного програмного забезпечення – зокрема, об'єктно-орієнтованої бібліотеки, яка б забезпечувала прозору взаємодію з VirtualBox. Це відкриє можливість використання рушія віртуалізації в автоматизованих сценаріях розгортання інфраструктури, підвищить відтворюваність середовищ та пришвидшить налаштування систем.

Платформою для реалізації бібліотеки було обрано .NET (C#) – кросплатформенне, об'єктно-орієнтоване середовище, що забезпечує інтеграцію з іншими мовами та системами, а також підтримує сучасні парадигми розробки. Незважаючи на потребу встановлення виконуваного середовища .NET на цільову машину, у більшості сучасних ОС воно вже присутнє або легко додається.

У рамках дослідження були розглянуті три основні варіанти взаємодії з VirtualBox:

1. **COM/XPCOM API** – бібліотека класів для взаємодії з VirtualBox.
2. **Web API** – серверний XML-інтерфейс, що надає доступ до функцій через HTTP.

3. CLI (VBoxManage) – командний інтерфейс, що дозволяє взаємодіяти через CLI.

Критерієм вибору оптимального інтерфейсу були: стабільність, документація, доступність із середовища .NET, час на розробку та можливість повної автоматизації.

Першим розглянуто COM API – інтерфейс, що надає доступ до всіх функцій VirtualBox через бібліотеки мови програмування. Проте використання COM з .NET виявилось значно ускладненим: офіційних .NET-збірки відсутні, а інструменти генератори на основі IDL-описів дають нестабільні результати. Після генерації коду з XML-документації, вбудованої у Web API, з'ясувалося, що структура отриманого коду орієнтована переважно на Python, має неповну типізацію та не відповідає синтаксису C-подібних мов. Додатково були помічені фрагменти незавершеного коду з посиланням на зовнішні бібліотеки, не описані для .NET. Через відсутність шаблонів для C#, низьку читабельність та складність налагодження, даний варіант було визнано недоцільним для використання у практичній реалізації.

Web API VirtualBox представляє собою HTTP-сервер, який приймає запити у форматі XML-RPC. Потенційно цей спосіб є найзручнішим для інтеграції з іншими системами, однак має кілька істотних обмежень: за замовчуванням сервер вимкнено, і його потрібно активувати вручну або скриптом через CLI; офіційна документація API недоступна, що ускладнює створення запитів (точні URI методів доводиться шукати у сторонніх репозиторіях або методом зворотного інжинірингу); частина методів, знайдених у сторонніх джерелах, повертає HTTP-статуси помилок (зокрема 404), що свідчить про застарілість або зміну API без зворотної сумісності.

Ураховуючи ці труднощі, а також ризики, пов'язані з нестабільною підтримкою API у майбутніх версіях, було прийнято рішення не використовувати Web API у фінальній реалізації.

CLI VirtualBox виявився найбільш стабільним і добре задокументованим способом взаємодії. Хоча цей інтерфейс не є повноцінним API у традиційному розумінні, він дозволяє виконувати всі необхідні операції через запуск зовнішніх процесів із відповідними аргументами [3].

Серед викликів – нестандартний, часто неформатований вивід, який потребує ручного парсингу та обробки помилок. Для підвищення ефективності бібліотеки всі виклики CLI реалізовано з підтримкою асинхронності (async/await), що мінімізує блокування потоку і дозволяє вбудовувати бібліотеку в UI-додатки та CI/CD-агенти.

Отже, у роботі проаналізовано три основні варіанти програмної взаємодії з VirtualBox. COM API виявився недостатньо документованим та неадаптованим до C#, Web API – малодокументованим і нестабільним, що обумовило їхню відмову на користь CLI.

Вибір CLI виявився найкращим з точки зору стабільності, передбачуваності поведінки та мінімального бар'єру для реалізації. Хоча CLI потребує додаткових зусиль з парсингу та асинхронного виконання, він забезпечує повний доступ до необхідного функціоналу.

Створена C#-бібліотека відкриває можливості для інтеграції VirtualBox у CI/CD-сценарії, використання в середовищах автоматизації розгортання, а також для побудови локальних інфраструктур із використанням принципів Infrastructure as Code. Подальша робота зосереджуватиметься на інтеграції цієї бібліотеки з IaC-інструментами (наприклад, через створення Terraform-провайдера), а також на розширенні підтримки для кросплатформених систем управління віртуалізацією.

Бібліографія

1. What is infrastructure as code (IaC)? - Azure DevOps. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/devops/deliver/what-is-infrastructure-as-code>.
2. Oracle VirtualBox - Oracle VirtualBox Documentation. Oracle Help Center. URL: <https://docs.oracle.com/en/virtualization/virtualbox/>.
3. Managing Oracle VM VirtualBox from the Command Line. *fw_error_www*. URL: <https://www.oracle.com/technical-resources/articles/it-infrastructure/admin-manage-vbox-cli.html>.