

CONTAINERLAB ЯК ІНСТУРМЕНТ МОДЕЛЮВАННЯ ТА СИМУЛЯЦІЇ МЕРЕЖ НА ОСНОВІ КОНТЕЙНЕРИЗАЦІЇ

Шестопалов С.С.

Харківський національний університет радіоелектроніки,
кафедра інфокомунікаційної інженерії ім. В.В. Поповського,
Україна.

E-mail: serhii.shestopalov@nure.ua

Abstract

In the last decade, containerization technologies like Docker have rapidly developed, enabling isolated execution of multiple operating environments without full hardware emulation. This approach offers advantages over traditional virtual machines by providing standardization, scalability, and automated deployment of applications while using significantly fewer resources. Containerlab addresses the challenge of deploying and interconnecting containers for complex network topologies by offering an open-source command-line tool that simplifies the creation of network labs based on containers and virtual machines. Using simple YAML files, it automates the process of creating and configuring network connections between nodes and managing their lifecycle. This paper provides an overview of Containerlab's features and demonstrates its applications in development, testing, education, security auditing, and automation.

Останнє десятиліття відзначилося стрімким розвитком і поширенням технологій контейнеризації, що почалося з випуском Docker у 2013 році. Контейнеризація дозволяє запускати ізольовані операційні середовища без необхідності повної емуляції апаратного забезпечення. Таким чином, використовуючи механізми операційної системи, створюються численні користувальські середовища, які з погляду запущеного в них програмного забезпечення виглядають як повноцінні окремі комп'ютери. Цей підхід має переваги над звичайними віртуальними машинами, дозволяючи стандартизувати, масштабувати та автоматизовано запускати програми в ізольованих середовищах, які за потреби можуть мати доступ до ресурсів, як-от процесор, різноманітні підключені пристрої, файли, мережні з'єднання тощо, водночас споживаючи значно менше ресурсів, що дозволяє більш ефективно використовувати наявне обладнання та будувати більш складні системи [1].

Довгий час контейнеризація в основному використовувалася для запуску веб-серверів та інших користувальських програм, але з розвитком та більш широкою адаптацією цих технологій з'явилися мережні операційні системи (Network Operating Systems, NOS), які побудовані з ідеєю підтримки запуску в контейнері або легко дозволяють це зробити. Серед них Nokia SR Linux, Arista cEOS, Azure SONiC, Juniper cRPD, FRRouting (FRR) тощо.

На жаль, самі по собі середовища запуску контейнерів та програмні інструменти, як-от Docker Compose, не надають усіх потрібних можливостей для належного запуску та поєднання контейнерів у мережі, оскільки вони не дозволяють створювати однорангові (peer-to-peer) з'єднання для побудови складних мереж. Containerlab є одним з можливих рішень для цієї проблеми; це інструмент командного рядка з відкритим кодом, який спрощує процес розгортання мережніх лабораторій на базі контейнерів та віртуальних машин. Він надає можливість описувати мережні топології за допомогою простих YAML-файлів, автоматизуючи процес створення та налаштування мережніх з'єднань між вузлами та керування життєвими циклами систем [2].

Отже, серед основних можливостей і особливостей Containerlab виділяються наступні:

1. Containerlab підтримує широкий спектр NOS, включно з Nokia SR Linux, Arista cEOS, FRRouting (FRR) та інші [3]. Це дозволяє моделювати реальні мережні сценарії з використанням різноманітних пристрій і протоколів.

2. YAML-файли використовуються для легкого опису мережних вузлів та з'єднань між ними [4]. Це спрощує процес створення складних мережних топологій та робить його більш зрозумілим і повторюваним.

3. Containerlab автоматично створює необхідні мережні інтерфейси та з'єднання між контейнерами, використовуючи засоби Linux, як-от veth та bridge [5]. Це забезпечує високопродуктивні з'єднання з низькими затримками між вузлами мережі.

4. Інструмент надає команди для запуску, зупинки та видалення топологій, що спрощує процес управління мережними лабораторіями та дозволяє швидко змінювати конфігурації за потреби [6].

5. Не всі мережні операційні системи можуть бути контейнеризовані, тому Containerlab інтегрується з VRnetlab, дозволяючи запускати віртуальні мережні маршрутизатори всередині контейнерів Docker [7].

6. Завдяки можливості автоматизованого розгортання та налаштування, Containerlab легко інтегрується в конвеєри безперервної інтеграції та доставки (Continuous Integration/Continuous Delivery, CI/CD), що сприяє швидкому впровадженню та тестуванню мережних рішень [8].

7. Використання контейнерів дозволяє запускати більшу кількість мережних вузлів на одному хості порівняно з традиційними віртуальними машинами, що робить Containerlab ефективним інструментом для моделювання великих мережних топологій [9].

Враховуючи широкий список можливостей, Containerlab може бути застосований у багатьох сферах для вирішення різноманітних задач, а саме:

- швидка розробка та тестування у разі розгортання тестових середовищ для перевірки нових конфігурацій, протоколів або мережних сервісів;
- під час навчання та освітнього процесу, коли викладачі та студенти можуть використовувати Containerlab для створення навчальних лабораторій, що моделюють реальні мережні сценарії;
- в процесі проведення аудиту та моніторингу безпеки, у разі чого фахівці з безпеки можуть моделювати мережні атаки та перевіряти стійкість конфігурацій до вразливостей;
- завдяки інтеграції з інструментами автоматизації, такими як Ansible, можна автоматизувати налаштування та управління мережними пристроями.

Containerlab є потужним інструментом для моделювання та симуляції мереж, який використовує найсучасніші можливості та переваги контейнеризації для запуску різних мережних операційних систем. Він спрощує процес створення мережних лабораторій, роблячи його більш доступним та ефективним для інженерів, дослідників та освітян. Також він повністю безкоштовний та має відкритий програмний код, роблячи його доступним для кожного [2].

Література

1. Docker Inc. What is a Container? URL: <https://www.docker.com/resources/what-container> (дата звернення: 01.11.2024).
2. Containerlab Project. Containerlab Documentation. URL: <https://containerlab.dev/> (дата звернення: 01.11.2024).
3. Containerlab Project. Supported Nodes. URL: <https://containerlab.dev/manual/kinds/> (дата звернення: 01.11.2024)..
4. Containerlab Project. Topology Definition. URL: <https://containerlab.dev/manual/topo-def-file/> (дата звернення: 01.11.2024)..
5. Containerlab Project. Network Wiring Concepts. URL: <https://containerlab.dev/manual/network/> (дата звернення: 01.11.2024).
6. Containerlab Project. CLI Reference. URL: <https://containerlab.dev/cmd/deploy/> (дата звернення: 01.11.2024).
7. VRnetlab Project. VRnetlab GitHub Repository. URL: <https://github.com/hellt/vrnetlab> (дата звернення: 01.11.2024).
8. Containerlab Project. CI/CD Integration. URL: <https://containerlab.dev/manual/ci-cd/> (дата звернення: 01.11.2024).
9. Containerlab Project. Performance and Scaling. URL: <https://containerlab.dev/manual/performance/> (дата звернення: 01.11.2024).