Контейнеризація як розвиток механізмів ізоляції процесів

004.056 Ситник Маргарита [1]

Харківський національний університет радіоелектроніки,
[1]marharyta.sytnyk@nure.ua

Розвиток сучасних програмних систем супроводжується зростанням вимог до їхньої надійності, масштабованості та переносимості. Початковим етапом у масовому спільному використанні апаратних ресурсів стала віртуалізація, однак перехід до мікросервісної архітектури виявив потребу в більш швидких та компактних рішеннях, що призвело до популяризації контейнеризації. Контейнеризація — це пакування програмного коду разом з усіма необхідними бібліотеками та залежностями. Її результатом є єдиний, легкий виконуваний файл, який стабільно працює на будь-якій інфраструктурі. Поява технології Docker значно спростила створення та управління контейнерами, зробивши цей підхід масовим [1, 2].

1. Ізоляція процесів та її реалізація

В основі контейнеризації лежить ізоляція процесів – здатність ядра ОС обмежувати видимість системних ресурсів для об’єктів та регулювати їхнє споживання. Завдяки цьому можливий запуск кількох застосунків на одній системі без їх взаємного впливу. Технічна реалізація ізоляції в ОС Linux базується на двох компонентах ядра:

- Namespaces: формують персоналізований «світогляд» для кожного процесу, ізолюючи файлову систему, ідентифікатори процесів, мережу, ідентифікатори користувачів тощо.
- Cgroups: розподіляють використання ресурсів і гарантують, що процес не конкуруватиме за пам'ять чи процесорний час, зарезервовані за іншими процесами.

2. Контейнеризація як елемент безпеки

Контекст безпеки контейнеризації тісно пов'язаний з архітектурою спільного ядра ОС. На відміну від повної ізоляції у віртуальних машинах, використання спільного ядра створює ризик втечі з контейнера у разі експлуатації вразливостей хост-системи. Для мінімізації поверхні атаки критично важливим є дотримання принципу найменших привілеїв, зокрема впровадження підходу Rootless Docker, а також регулярне сканування базових образів на наявність вразливостей до моменту їх розгортання в системі.

3. Контейнеризація та Docker

Функціонування контейнеризації на базі Docker забезпечується кількома ключовими компонентами, серед яких основними є образ (image), контейнер (container) та Dockerfile.

Образ являє собою незмінний шаблон, що містить програмний код, бібліотеки та всі необхідні залежності для запуску застосунку. Образи формуються пошарово, що дозволяє ефективно зберігати та повторно використовувати їх компоненти [3].

Контейнер є запущеним екземпляром образу, який функціонує як ізольований процес у системі. Контейнери створюються на основі образів і забезпечують виконання застосунків у відокремленому середовищі.

Dockerfile — це текстовий файл, що містить інструкції для автоматизованого створення образу. У ньому визначається базове середовище, необхідні залежності та команди, які потрібно виконати для підготовки контейнера до роботи.

4. Порівняння контейнерів та віртуальних машин

Контейнеризація та віртуалізація є різними підходами до ізоляції процесів, які мають суттєві відмінності в архітектурі та ефективності використання ресурсів. Контейнери створюють ізольовані середовища, що використовують спільне ядро операційної системи хоста, тоді як віртуальні машини функціонують як повноцінні незалежні системи з власним ядром.

Порівняння основних характеристик контейнерів і віртуальних машин наведено у таблиці 1.

Таблиця 1

Порівняння контейнерів і віртуальних машин

Характеристика	Контейнери	Віртуальні машини
Ядро ОС	Спільне	Окреме
Час запуску	1–3 секунди	20–60 секунд
Витрати пам’яті	50–200 МБ	512 МБ – 2 ГБ
Накладні витрати	2–5%	5–15%
Рівень ізоляції	Частковий	Повний

1. Роль контейнеризації та віртуалізації на рівні операційної системи в розвитку хмарно орієнтованих додатків. Наукова періодика Міжрегіональної Академії управління персоналом. URL: https://journals.maup.com.ua/index.php/it/article/view/4822/5115 (дата звернення: 26.04.2026).
2. Docker Inc. What is Docker?. Docker Documentation. URL: https://docs.docker.com/get-started/docker-overview/ (date of access: 28.04.2026).
3. Docker Inc. What is an image?. Docker Documentation. URL: https://docs.docker.com/get-started/docker-concepts/the-basics/what-is-an-image/ (date of access: 28.04.2026).

Critical Infrastructure Security: Electronic Communications Networks of Electronic Communications Operators

UDK: 004.056:004.7:621.39:351.86

Olena Shelest-Polishchuk [1],
Bohdan Skybun[2]

Kyiv Professional College of Communication,
[1] deksog@ukr.net, [2] skubyn.bogdan@gmail.com