

УДК 004.75

*Ожго Ю. А., здобувач**Миколайчук В. В., ст. викладач**Державний університет «Житомирська політехніка»*

ДОСЛІДЖЕННЯ ПОПУЛЯРНИХ CONTAINER RUNTIME

Сьогоденний процес розробки програмного забезпечення вимагає використання найсучасніших та найактуальніших технологій, методологій та підходів, що дозволяють створювати якісні та практичні рішення. Одним із таких ключових підходів є застосування контейнеризації. Контейнеризація – це метод запакування програми з усіма їй необхідними залежностями, бібліотеками та файлами конфігурації в єдину, ізольовану від навколишнього середовища сутність, яку називають контейнером[1]. Контейнери використовуються при розробці, тестуванні, доставці та розгортанні продукту і де-факто є стандартом для більшості етапів життєвого циклу ПЗ.

Середовище виконання контейнера (Container Runtime) – це програмне рішення, що реалізує стандарти OCI (Open Container Initiative) та відповідає за створення, запуск і керування контейнерами [2]. Воно забезпечує ізоляцію й контроль ресурсів на низькорівневому рівні(low-level) та керування образами й життєвим циклом на високорівневому рівні(high-level).

До найбільш поширених low-level середовищ виконання відносять [3]:

- **runc** – еталонна реалізація специфікації OCI Runtime Specification для запуску OCI-сумісних контейнерів у Linux. Він обробляє пакет із файловою системою та конфігурацією і використовує механізми ядра ОС, такі як namespaces та cgroups, для ізоляції процесу та управління ресурсами контейнера;

- **crun** – мінімалістична та високопродуктивна альтернатива runc, що також реалізує специфікації OCI. Розроблений на мові C, цей runtime забезпечує швидший запуск контейнерів і оптимізоване використання ресурсів порівняно з runc. Активно розвивається спільнотою Red Hat;

- **runhcs** – це розгалуження runc, створене Microsoft для запуску контейнерів на платформі Windows. Він підтримує контейнери у форматі OCI і є реалізацією OCI Runtime Specification для Windows. На відміну від runc, який працює тільки в Linux і використовує механізми ядра Linux, runhcs взаємодіє з HCS і призначений виключно для Windows-середовища;

- **containerd** – середовище виконання з відритим кодом, яке

підтримується на Linux і Windows та полегшує управління життєвим циклом контейнерів за допомогою API. Насправді, дане програмне забезпечення можна віднести, як до низькорівневих так і до високорівневих середовищ, за рахунок його можливостей керувати образами та життєвим циклом, проте, оскільки ми не взаємодіємо з ним безпосередньо, то його прийнято вважати низькорівневим.

До найбільш поширених high-level середовищ виконання відносять [4]:

- Docker – це платформа для роботи з контейнерами, що надає інструменти для розробки, розгортання та управління контейнеризованими застосунками. У складі платформи інтегровано containerd, який відповідає за управління контейнерними образами та виконання контейнерів. Крім того, Docker включає засоби, що забезпечують взаємодію з Kubernetes;

- Podman – це програмне забезпечення з відкритим вихідним кодом,

розроблене компанією Red Hat, яке пропонує більш безпечну модель роботи з контейнерами порівняно з оригінальною реалізацією Docker. На відміну від Docker, Podman не потребує фонові служби (демона) і дозволяє запускати контейнери без прав суперкористувача (root), забезпечуючи більш безпечне та гнучке середовище виконання;

- CRI-O – це легке та мінімалістичне середовище виконання контейнерів з відкритим кодом, спеціально розроблене для Kubernetes. Воно реалізує стандарт CRI, дозволяючи Kubernetes запускати та керувати контейнерами відповідно до OCI стандартів, і є оптимізованою альтернативою Docker у контексті оркестрації.

Вибір container runtime залежить від потреб проєкту і є критично важливим для забезпечення ефективної, надійної та безпечної роботи контейнеризованих застосунків. Основними критеріями оцінки є сумісність зі стандартами OCI, інтеграція з оркестраторами, а також продуктивність, витрати ресурсів і рівень безпеки.

Список використаних джерел:

1. Container Runtimes Explained: Security, Types & Best Practices. URL: <https://www.akamai.com/glossary/what-is-a-container>.
2. What Is a Container Runtime?. URL: <https://www.aquasec.com/cloud-native-academy/container-security/container-runtime/>.
3. What are Container Runtimes. URL: <https://www.sysdig.com/learn-cloud-native/what-are-container-runtimes>.
4. Most Popular Container Runtimes. URL: <https://www.cloudraft.io/blog/container-runtimes>.