

Магістрант спеціальності «Комп'ютерна інженерія» Абрамов Д.О.,

к.т.н., доцент Веселовська Г.В.,

кафедра інформаційних технологій,

факультет інформаційних технологій та дизайну,

Херсонський національний технічний університет, м. Херсон, Україна

Дослідження архітектурних особливостей технології Docker і віртуальних машин

Немаловажним чинником ускладнення цифрової трансформації бізнесу організацій є різноманітність асортименту програм, хмарних технологій і локальної інфраструктури.

Указану проблему можна легко вирішувати, застосовуючи засіб розробки програмного забезпечення та технологію віртуалізації Docker, що набагато спрощує розробку, розгортання та керування програмами за допомогою контейнерів [1-5].

Технологія віртуалізації Docker (Docker-engine) працює на засадах контейнерної платформи, що поєднує в безпечний автоматизований ланцюжок традиційні програми та мікро-сервіси, засновані на операційних системах (ОС) Windows і Linux, мейнфреймах. Контейнер становить собою легкий, самостійний, виконуваний пакет програмного забезпечення, що містить усі бібліотеки, файли конфігурації, залежності та інші складові, необхідні для роботи програми. Оскільки контейнер забезпечує необхідне середовище протягом життєвого циклу розробки програмного забезпечення, то робота програм є незалежною від того, де вони перебувають і на якій машині працюють. Завдяки ізольованості, контейнери забезпечують потрібну безпеку, що дозволяє одночасно запускати декілька контейнерів на хості. Контейнери

мають невелику вагу, оскільки для них непотрібне додаткове навантаження гіпервізора (гостьової ОС, такої як VMWare, VirtualBox тощо), натомість контейнери працюють безпосередньо в машинному ядрі хосту.

У цілому, контейнери забезпечують такі переваги: скорочення ресурсів управління інформаційними технологіями (ІТ); зменшення розміру подань; швидше прокручування додатків; зменшені та спрощені оновлення безпеки; менше коду для передачі, міграції та завантаження робочих навантажень.

Наведемо найбільш суттєві відмінності між Docker і віртуальними машинами (ВМ).

За перший і головний критерій порівняння, буде взята підтримка ОС та архітектура. Основна відмінність полягає в архітектурі, представленій схемою на рисунку 1.

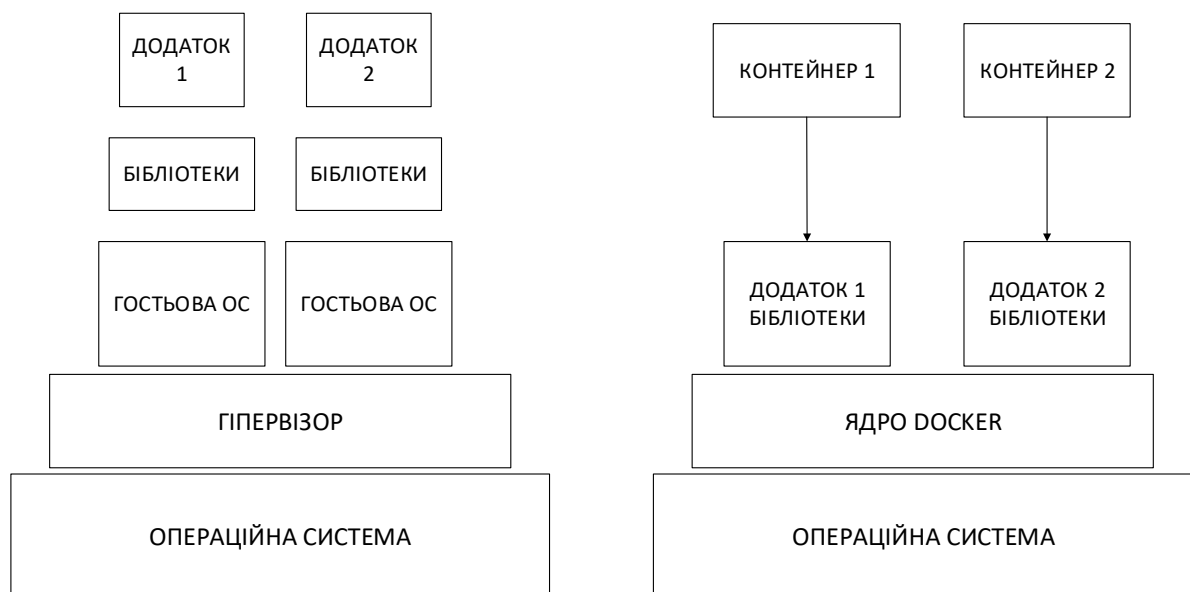


Рисунок 1 – Відмінності між технологіями Docker і ВМ
у підтримці ОС та архітектурі

Оскільки кожна віртуальна машина (ВМ) містить головну та гостьову ОС, де гостьовою може бути будь-яка ОС (наприклад, Linux, Windows тощо), незалежно від ОС хоста, то ВМ є доцільними при роботі додатків або служб у різних ОС.

Контейнери Docker розміщуються на одному фізичному сервері з основною ОС і спільно використовують її. Спільний доступ до основної ОС між контейнерами робить їх легкими, зменшує час завантаження. Контейнери Docker є доцільними для запуску декількох програм крізь одне ядро ОС.

За другий критерій порівняння, буде взято безпеку.

Оскільки ВМ є автономними завдяки ядру та функціям безпеки, то з ними доцільно працювати додатком із потребами більших привілеїв та захисту.

Наразі, не рекомендується надавати контейнерам Docker root-доступ до програм і запускати їх із адміністративними привілеями через їхнє спільне ядро хосту (доступ контейнерної технології до підсистем ядра здатен призвести до того, що одна заражена програма зможе зламати хост-систему).

За третій критерій порівняння, буде розглянуто портативність.

ВМ, ізольовані від своєї ОС, не переносяться на декілька платформ.

Пакети контейнерів Docker, що є автономними, можуть запускати програми в будь-якому середовищі та не потребують гостьової ОС, їх можна легко переносити на різні платформи. Полегшені контейнери Docker можна безпроблемно розгортати на серверах, оскільки їх можна запускати та зупиняти за набагато менший час, ніж із ВМ. На рівні розробки додатку, що потребує тестування на різних платформах, також слід застосовувати контейнери Docker.

За четвертий критерій порівняння, буде досліджено продуктивність.

ВМ вимагають більше ресурсів за контейнери Docker: для їхнього завантаження, потрібно завантажити всю ОС.

Для ВМ, ресурси центрального процесору, пам'яті та введення-виведення можуть не виділятися назавжди.

Легка архітектура контейнерів Docker є менш ресурсоємною, ніж ВМ.

Для контейнерів, використання ресурсів відбувається із завантаженням або трафіком.

Масштабування та дублювання контейнерів є простим та легким, порівняно з ВМ, оскільки нема необхідності встановлювати до них ОС.

Також відмінності Docker і ВМ узагальнено в таблиці 1.

Таблиця 1 – Порівняння Docker і ВМ за основними показниками

Показник	Docker	Віртуальні машини
Час завантаження	Завантажується декілька секунд	Запуск ВМ триває декілька хвилин.
Оновлення роботи	Докери використовують механізм виконання	Віртуальні машини використовують гіпервізор
Ефективність пам'яті	Для віртуалізації не потрібно місця (використовується менше пам'яті)	Потрібно завантажити всю ОС перед запуском платформи, що є менш ефективним
Ізоляція	Схильність до негараздів через відсутність положень щодо ізоляційних систем	Можливість втручання є мінімальною завдяки ефективному механізму ізоляції
Розгортання	Розгортання є простим (подання в контейнері застосовують на всіх платформах)	Розгортання є досить тривалим (за виконання відповідають окремі екземпляри)
Використання	Складний механізм використання на основі сторонніх і керованих Docker інструментів	Інструменти прості у використанні та простіші в роботі, порівняно з технологією Docker

Технології Docker і ВМ призначені для різного використання.

ВМ вважають переважними для: виробничого середовища (завдяки тому, що вони працюють у власній ОС, не являючи загрози для головного комп'ютера); додатків, що зазвичай є статичними, не піддаються частим змінам. Недоліком ВМ є тривале розгортання та проблемний запуск мікро-сервісів.

Більшість компаній переходять до використання контейнерів Docker, що: є маленькими, ізольованими, сумісними, високопродуктивними, швидко реагуючими (завдяки використанню ядра Docker, замість гіпервізора ВМ, при

неспільному ядрі хоста); мають порівняно низькі накладні витрати (завдяки сумісності спільного використання однопоточних бібліотек і бібліотек програм). Платформу Docker побудовано так, щоб: бути гнучкішою; контейнери могли оновлюватися легко та часто. Використовують Docker і компанії, зацікавлені в корпоративній безпеці інфраструктури. Також Docker, що пропонує різні платформи ОС, рекомендують для ретельного тестування програм.

Оскільки Docker набирає популярності, але деінде ВМ є кращим вибором, то поширено гібридний підхід, де вибір між ВМ і контейнерами Docker визначає пропоноване навантаження (контейнери Docker не суперечать ВМ, доповнюючи інструменти для різного навантаження та використання).

Висновки: виділені та проаналізовані відмінності між архітектурою, покладеною в основу технології Docker і віртуальних машин, за критеріями, що відображають безпечність, ефективність, компактність і продуктивність; доцільним є подальше здійснення другого етапу дослідження, з детальнішим вивченням та аналізом розглянутого питання на засадах методології експертного оцінювання, та створення рекомендаційної експертної системи.

Література:

1. Docker vs Virtual Machines: Differences You Should Know // Electronic resource. URL: <https://cloudacademy.com/blog/docker-vs-virtual-machines-differences-you-should-know> (access mode 30.09.2020).
2. What is Docker? // Electronic resource. URL: <https://www.docker.com/what-is-docker> (access mode 30.09.2020).
3. Merkel D. Docker: lightweight Linux containers for consistent development and deployment. *Linux Journal*. 2014. 239 p.
4. James E. Smith, Ravi Nair. Virtual Machines: Versatile Platforms For Systems And Processes. Morgan Kaufmann, 2005. 656 p.
5. Docker: Introducing execution drivers and libcontainer // Electronic resource. URL: <https://www.docker.com/blog/docker-0-9-introducing-execution-drivers-and-libcontainer> (access mode 30.09.2020).