

ВИКОРИСТАННЯ СИСТЕМИ DOCKER ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Процес розробки програмного продукту складний і дуже важливий у сучасному світі. Залежно від проекту кількість рядків коду може складати від декількох сотень до кількох тисяч (мільйонів) команд. Розробку програмного забезпечення частіше здійснює колектив розробників. Їх рішення та методи реалізації поставленої задачі мають бути узгодженими. Кожен учасник колективу працює над рішенням конкретної задачі. Але проблеми можуть виникати на етапі передачі програмного продукту клієнту, тиражуванні та використання багатьма одного програмного продукту одночасно. Для рішення цієї проблеми була розроблена система Docker.

Система Docker - це програмна платформа для швидкої розробки, тестування і розгортання додатків. Система Docker упаковує ПО в стандартизовані блоки, які називаються контейнерами. Контейнери - це новий рівень віртуалізації IT-інфраструктури. Вони ізолюють окремі процеси всередині однієї ОС і запускають їх з загальним доступом до бібліотек і ресурсів цієї ОС. Завдяки контейнерів кожне окремо запущене застосування зберігає всі переваги хмари: резервування, безперебійність роботи, масштабованість, автоматичне керування. Кожен контейнер включає все необхідне для роботи програми: бібліотеки, системні інструменти, код і середу виконання. Завдяки системі Docker можна швидко розгортати і масштабувати додатки в будь-якому середовищі і зберігати впевненість в тому, що код буде працювати. Це не єдина подібна платформа, але, безперечно, одна з найпопулярніших і затребуваних.

Використання системи Docker на AWS (Amazon Web Services) дає розробникам і системним адміністраторам надійний і економічний спосіб збирання, доставки і запуску розподілених додатків будь-якого масштабу. AWS підтримує обидві моделі ліцензування Docker: ліцензію з відкритим кодом Docker Community Edition (CE) і ліцензію Docker Enterprise Edition (EE) на основі передплати.

В основі роботи Docker лежить стандартизований спосіб виконання коду. Docker - це операційна система для контейнерів. Подібно до того, як віртуальна машина створює віртуальне представлення апаратного забезпечення сервера, контейнери створюють віртуальне представлення серверної операційної системи. Після установки на кожен сервер Docker надає доступ

до простих команд, необхідним для збірки, запуску або зупинки контейнерів.

Система Docker відокремлює мережеві ресурси, сховище і деталі ОС. З Docker додаток дійсно не залежить від налаштувань цих низькорівневих ресурсів. При переміщенні Docker-контейнера від одного Docker-хоста до іншої машини з Docker, система гарантує, що навколишнє середовище для додатка залишиться незмінним.

Ця система допомагає програмістам створювати локальні середовища розробки, які виглядають як продакшн-сервер. Коли програміст закінчує писати і починає тестувати код, він може обернути його в контейнер, опублікувати безпосередньо на сервері або в приватній хмарі, і він відразу буде працювати, так як це одне і та ж середовище.

Docker пришвидшує процес розробки програмного забезпечення. В системі не потрібно встановлювати сторонні додатки, їх запускають контейнерах. Docker також дозволяє запускати різні версії одного додатку одночасно. Відділення додатків від апаратного забезпечення, що лежить в їх основі, - фундаментальний концепт віртуалізації. Контейнери йдуть ще далі і відокремлюють додатки від ОС. Завдяки цій особливості програмісти отримують гнучкість і масштабування при розробці.

Переваги використання системи Docker контейнеризації:

- кожен контейнер має невелику вагу і швидко запускається;
- контейнер не є віртуальною машиною і не емулює всю систему;
- контейнер ізолюють один додаток і його залежності - все зовнішні бібліотеки програмного забезпечення, які потрібно запустити з додатком;
- контейнери економлять ресурси всієї системи;
- забезпечує переносимість додатків;
- забезпечує зниження вартості розробки;
- забезпечує стандартизацію роботи додатків;
- забезпечує незалежність від системи - додаток працює ідентично.
- Недоліки системи Docker
 - не вирішує проблеми безпеки самого додатку;
 - не замінює віртуальні машини;
 - не прискорює розробку додатків, робить їх лише більш надійними і ефективними за рахунок посиленої роботи співробітників.
- Отже, система Docker добре підходить для вирішення таких задач:
 - зручна передача серверного проекту клієнтові;
 - забезпечення тиражовані серверів;
 - забезпечення можливості використання раніше створених серверних конфігурацій;
 - зручність поновлення раніше встановлених версій продукту.

Docker є важливим інструментом для кожного сучасного розробника, як основа апаратної віртуалізації додатків. Ця технологія має широкий

функціонал і можливостями для контролю процесів. Docker дозволяє не тільки розгорнути контейнери, а й оперативне масштабувати їх екземпляри, працювати з багатоконтейнерними додатками (Docker Compose), а також об'єднувати кілька Docker-хостів в єдиний кластер (Docker Swarm).

Система Docker досить проста в освоєнні як для досвідчених ІТ-фахівців, так і для новачків. Програмне забезпечення сумісне з усіма версіями операційних систем Linux і Windows, тому область застосування Docker практично не обмежена.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Why you should use Docker and containers Serdar Yegulalp [Електронний ресурс]. – Режим доступу: <https://www.infoworld.com/article/3310941/why-you-should-use-docker-and-containers.html>

2. Топ 10 преимуществ, которые вы получите, используя Docker Cloude [Електронний ресурс] – Режим доступу: <https://cloud.kiev.ua/top-10-preimushhestv-kotorye-vy-poluchite/>

3. Контейнеризация [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Контейнерная_виртуализация

4. Что такое контейнер? Обзор технологии контейнеризации [Електронний ресурс]. – Режим доступу: <https://www.cloud4y.ru/about/news/obzor-tekhnologii-konteynerizatsii/>