**ЗАСТОСУВАННЯ КОНТЕЙНЕРІВ DOCKER ДЛЯ ПІДТРИМКИ ПРОВЕДЕННЯ ПРОЦЕСІВ АНКЕТУВАННЯ**

**USING DOCKER CONTAINERS TO SUPPORT QUESTIONNAIRE PROCESSES**

Науковий керівник: к.т.н., доцент кафедра інформаційних технологій

Шибаєва Наталя Олегівна

Здобувач бакалаврату Мустафаєв Шаміль Махачович

Supervisor: (Ph.D), assistant professor Department of Information Technology

Shibayeva Natalia

 Bachelor's candidate Mustafayev Shamil

***Анотація:*** *розглянуто ефективність використання контейнерів Docker. Основні переваги контейнеризації при створенні програм.*

***Ключові слова****: контейнеризація, віртуалізація, анкетування.*

***Abstract****: The efficiency of using Docker containers is discussed. The main advantages of containerization in software development.*

***Keywords****: containerization, virtualization, questionnaires.*

Контейнери — це спосіб упакувати програму та всі її залежності в єдиний образ. Цей образ запускається в ізольованому середовищі, що не впливає на основну операційну систему. Контейнери дозволяють відокремити додаток від інфраструктури: розробникам не потрібно замислюватися, в якому оточенні працюватиме їхня програма, чи будуть там потрібні налаштування та залежності[1]. Вони просто створюють програму, упаковують всі залежності та налаштування в єдиний образ. Потім цей образ можна запускати на інших системах, не турбуючись, що програма не запуститься [2].

Контейнеризація схожа на віртуалізацію, але це не одне й те саме. Віртуалізація працює як окремий комп'ютер зі своїм віртуальним обладнанням та операційною системою. При цьому всередині однієї ОС можна запустити іншу ОС. У разі контейнеризації віртуальне середовище запускається прямо з ядра основної операційної системи та не віртуалізує обладнання [3]. Це означає, що контейнер може працювати тільки в тій самій ОС, що й основна. При цьому, оскільки контейнери не віртуалізують обладнання, вони споживають набагато менше ресурсів.

Docker — це відкрита платформа для розробки, доставки та запуску програм. Docker дає змогу відокремити ваші програми від інфраструктури, щоб ви могли швидко постачати програмне забезпечення. За допомогою Docker ви можете керувати своєю інфраструктурою так само, як і своїми програмами [4]. Використовуючи переваги методології Docker для швидкої доставки, тестування та розгортання коду, ви можете значно скоротити затримку між написанням коду та його запуском у виробництві.

Переваги використання контейнерів Docker [5]:

1. Docker вирішує проблеми залежностей та робочого оточення. Контейнери дозволяють упакувати в єдиний образ програму та всі її залежності: бібліотеки, системні утиліти та файли налаштування. Це полегшує перенесення програми на іншу інфраструктуру.

Наприклад, розробники створюють програму в системі розробки, там все налаштовано і програма працює. Коли програма готова, її потрібно перенести в систему тестування і потім у продуктивне середовище. І якщо в цих системах не вистачатиме будь-якої залежності, то додаток не буде працювати. У цьому випадку програмістам доведеться відволіктися від розробки та спільно з командою підтримки розбиратися в ситуації.

Контейнери дозволяють уникнути такої проблеми, тому що вони містять все необхідне для запуску програми. Програмісти зможуть зосередитись на розробці, а не вирішенні інфраструктурних проблем.

2. Ізоляція та безпека. Контейнер — це набір процесів ізольованих від основної операційної системи. Програми працюють лише всередині контейнерів і не мають доступу до основної операційної системи. Це підвищує безпеку програм, тому що вони не зможуть випадково чи навмисне нашкодити основній системі. Якщо програма в контейнері завершиться з помилкою або зависне, це не торкнеться основної ОС.

3. Прискорення та автоматизація розгортання додатків та масштабованість. Контейнери спрощують розгортання програм. У класичному підході для встановлення програми може знадобитися виконати кілька дій: виконати скрипт, змінити файли налаштувань тощо. У цьому процесі не виключена можливість людської помилки: користувач запустить скрипт два рази, переплутає послідовність або щось не зрозуміє. Контейнери дозволяють повністю автоматизувати цей процес, тому що включають всі необхідні залежності і порядок виконання дій.

Також контейнери спрощують розгортання на кількох серверах. У класичному підході для того, щоб розгорнути один і той же додаток на декількох машинах, потрібно буде повторювати одні й ті самі дії. Контейнери позбавляють цієї рутинної роботи і дозволяють автоматизувати розгортання.

4. Контейнери наближають до мікросервісної архітектури. Контейнери добре вписуються у мікросервісну архітектуру. Це підхід до розробки, при якому програма розбивається на невеликі компоненти, по можливості незалежні. Зазвичай протиставляється монолітної архітектури, де всі частини системи сильно пов'язані одна з одною. Це дозволяє розробляти нову функціональність швидше, адже у випадку з монолітною архітектурою зміна якоїсь частини може торкнутися всієї іншої системи.

5. Docker compose - одночасно розгортає кілька контейнерів. Docker-compose дозволяє розгортати та налаштовувати кілька контейнерів одночасно. Наприклад, для веб-програми потрібно розгорнути Spring Boot, MongoDb, JS. Кожна з програм — це окремий контейнер. Але в цій ситуації нам потрібні саме всі контейнери разом, а не окремо взята програма. Docker-compose дозволяє розгорнути та налаштувати всі програми однією командою, а без нього довелося розгортати та налаштовувати кожен контейнер окремо.

Висновки. Docker облегчує розробку програмного забезпечення шляхом контейнеризації стеків. Це забезпечую прискорене розгортання додатків на будь-якій машині, вирішуючи проблеми залежностей та робочого оточення.

ЛІТЕРАТУРА

1. Андріївський І. Л. Технології хмарних обчислень. - Л.: ДЕУ, 2018. - 79 с.
2. Андрійченко Д. К., Батраєва І. А. Енергоефективні кластерні та хмарні обчислення та технології. Харків: Нац. аерокосм. ун-т ім. Н.Є. Жуковського "ХАІ", 2017.
- 199 с.
3. Богданов А. В. Віртуалізація. М: Редакція, 2018. - 311 с.
4. Гіфт Н. Прагматичний ІІ. Машинне навчання та хмарні технології. - К.: Кітер, 2019. - 306 с.
5. Гребньов Є. У. Використання хмарних сервісів. - С.: Cnews, 2017. - 402 c.