II Міжнародна науково-практична Інтернет-конференція

“Інновації та перспективні шляхи розвитку інформаційних технологій” (ІПШРІТ-2023)

м. Черкаси, 6 грудня 2023 року

Збірник тез доповідей

Міністерство освіти і науки України
Черкаський державний технологічний університет
Наукове товариство студентів, аспірантів, докторантів і молодих вчених ЧДТУ
Noosphere Engineering School
Техніко-гуманітарна академія (м. Бельсько-Бяла, Польща)
Університет Аделаїди (Аделаїда, Південна Австралія)
Національний університет харчових технологій
Національний аерокосмічний університет ім. М.Є. Жуковського
«Харківський авіаційний інститут»
Національний авіаційний університет
Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Національний технічний університет «Харківський політехнічний інститут»
Центральноукраїнський національний технічний університет
Almaty University of Power Engineering and Telecommunications
(м. Алмати, Казахстан)

ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

ІІ Міжнародної науково-практичної інтернет-конференції

«ІННОВАЦІЇ ТА ПЕРСПЕКТИВНІ ШЛЯХИ РОЗВИТКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

(ІПШРІТ-2023)

06 грудня 2023 року

м.Черкаси

Черкаси
ЧДТУ
2023

References

1. Timothy J. Ross. Fuzzy Logic with Engineering Applications [Text] / T. J. Ross. – New Mexico : Wiley, 2010. – p. 49-57. DOI: https://www.doi.org/10.1002/9781119994374
2. Crina Grosan, Ajith Abraham. Intelligent Systems: A Modern Approach [Text] / C. Grosan, A. Abraham. Berlin : Springer-Verlag Berlin Heidelberg, 2011. P. 87-91. DOI: https://doi.org/10.1007/978-3-642-21004-4

Здоренко Юрій Миколайович
к.т.н., доцент кафедри комп'ютерних та інформаційних технологій і систем
Андрєєв Артур Сергійович
студент
Бочкар Володимир Олександрович
студент
Національний університет «Полтавська політехніка імені Юрія Кондратюка», м. Полтава, Україна

МЕТОДИ ЗАБЕЗПЕЧЕННННЯ ДОСТУПНОСТІ ІНФОМАЦІЙНИХ РЕСУРСІВ НА ОСНОВІ ВИКОРИСТАННЯ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ ВЕБ-ДОДАТКІВ

В умовах сьогодення існують підвищені вимоги для забезпечення доступності інформаційних ресурсів різних інформаційних систем, що потребує пошуку нових рішень для вибору архітектури програмних додатків. Реалізація сучасних інформаційних систем на основі веб-технологій висуває додаткові вимоги щодо забезпечення доступності відповідних інформаційних ресурсів розташованих в мережі. Так, для реалізації веб-додатків розрізняють монолітну та мікросервісну архітектуру [1]. Для монолітної архітектури існує складність забезпечення доступності інформаційних ресурсів в умовах низької надійності апаратних чи програмних складових, а також в умовах кібернетичних впливів. Оскільки доступність забезпечується надійністю програмних та апаратних складових інформаційної системи, то для її забезпечення пропонується застосувати модульний підхід при розробці програмного забезпечення та використання розподіленої структури (мережі) при його розгортанні.

Мікросервісна архітектура є одним з найкращих підходів реалізації таких розподілених програмних додатків. При використанні мікросервісного підходу замість одного великого додатка (моноліту), створюється множина невеликих і легко замінюваних модулів, які обмінюються один з одним незначними об'ємами даними, тобто є функціонально незалежними. Мікросервісна архітектура надає множину переваг, а саме: модульність; зменшення часу на розробку та тестування; можливість легкого масштабування; можливість застосування різного стеку технологій для кожного сервісу; ізоляція відмов в

межах окремих сервісів, а не всього додатку. Так, якщо внаслідок кібернетичного впливу буде відмова одного з модулів, то це не призведе до проблем з доступністю інформаційних ресурсів, які забезпечуються роботою інших модулів. Спосіб розгортання програмного додатку є також важливим фактором забезпечення доступності інформаційних ресурсів. Для мікросервісної архітектури існує декілька способів розгортання: на окремому сервері в ізольованих віртуальних середовищах або в контейнері, на різних серверах. Одним з пріоритетних підходів розгортання програмних додатків є використання контейнеризації. Контейнер, представляє собою образ, який містить лише програмний код мікросервісу та середовище його запуску та не потребує створення повноцінної віртуальної машини з власною операційною системою. Для підвищення надійності функціонування інформаційних систем використовують системи оркестровки контейнерів [3], які можуть забезпечити реплікацію контейнерів в межах кластеру. Це дозволяє забезпечити покращення показників надійності, запобігти перевантаженням та відмовам. Так, з декількома екземплярами мікросервісу зникають проблеми в разі відмови одного (або декількох) з них. Наявність декількох екземплярів мікросервісу також дозволяє здійснювати балансування між ними [2], запобігаючи перевантаженням. При зростанні навантаження (кількості запитів) на вже існуючі екземпляри певного сервісу система оркестровки контейнерів дозволяє збільшувати кількість його реплік. Задача розгортання мікросервісного веб-додатку з використанням контейнеризації полягає в вирішенні таких завдань: визначення необхідної кількості реплік запущених для одного мікросервісу; методів синхронізації реплік; перерозподілу трафіку між репліками.

Таким чином використання контейнеризації для розгортання веб-додатків з мікросервісною архітектурою та подальша оркестровка контейнерів дозволяє забезпечити високу надійність функціонування, а відповідно і доступність інформаційних ресурсів такої інформаційної системи. Також проведений аналіз показав, що реалізація веб-додатків на основі мікросервісів, має значні переваги в порівнянні з монолітною архітектурою. Поєднання мікросервісного підходу та способу розгортання таких додатків з використанням систем оркестровки контейнерів дозволить підвищити показники доступності інформаційних ресурсів в умовах апаратних чи програмних збоїв, технологічних перерв, кібернетичного впливу та інших негативних факторів. В подальшому планується провести дослідження щодо використання мікросервісної архітектури для розробки веб-додатків на основі різних стеків технологій, а систем контейнеризації та оркестровки контейнерів для розгортання таких мікросервісів. Зазначені підходи можуть бути використані для розгортання сучасних інформаційних систем з підвищеними вимогами по забезпеченню доступності інформаційних ресурсів.

Список літератури

1. Andrushko O., Borzov Y., Malets I., Prydatko, O. Аналіз процесів використання Docker для побудови мікросервісів. *Науковий вісник НЛТУ України*. Львів, 2017. № 27(9), С. 95-98. DOI: https://doi.org/10.15421/40270920

2. Neelam S., Yasir H., Sapna J., Gautam S. Load balancing and service discovery using Docker Swarm for microservice based big data applications. *Journal of Cloud Computing.* 2023. №12(1) P.1-9 DOI:10.1186/s13677-022-00358-7

3. Wijayanto, D., Arizona F. (2023). Implementation of Continous Delivery using Jenkins And Kubernetes with Docker Local Images. *Jurnal Dan Penelitian Teknik Informatika.* Sinkron, 2023, 8(4), P.2226-2235. DOI: https://doi.org/10.33395/sinkron.v8i4.12624

Кисліченко Яна Віталіївна
Черкаський державний технологічний університет,
м. Черкаси, Україна

ФРЕЙМВОРК ДЛЯ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ WEB-САЙТУ

Розуміння важливості тестування призводить до виникнення тенденцій, спрямованих на використання технічних методів перевірки якості програмного забезпечення. Важливим напрямком в цьому контексті є впровадження систем автоматизованого тестування. Ключову роль в забезпеченні якісного тестування відіграють способи організації взаємодії учасників розробки та вибір відповідної методології.

Для автоматизації цих процесів застосовуються спеціальні програмні засоби, а різноманіття інструментів для автоматизації тестування робить цей процес більш ефективним та надійним. Використання фреймворків для автоматизованого тестування додатків стає необхідною практикою для заощадження часу та матеріальних ресурсів при контролі якості додатків. Автоматизація тестування не лише зменшує час розробки, але також забезпечує підвищену надійність та безпеку продуктів. Використання автоматизованих тестів дозволяє автоматично генерувати звіти про стан продукту, надавати інформацію про стан додатку, легко повторювати тести та зменшувати вплив людського фактору на процес тестування.

Отже, в сучасних умовах ключовим є використання інструментів автоматизованого тестування для забезпечення високої якості програмного забезпечення та ефективності процесу розробки.

Веб-інформаційні системи (WIS) мають схожі архітектурні інфраструктури. Для побудови перших систем було розроблено кілька каркасів, які узагальнюють інфраструктуру для повторного використання в майбутніх проектах.

Ці каркаси дозволяють будувати великомасштабні веб-інформаційні системи з меншими зусиллями з кодування. Комбінування цих каркасів може призвести до архітектури на основі контейнера. Контейнер в даному випадку є систематичною структурою, що керує об’єктами, має чітко визначений життєвий цикл. Контейнер для розширюваних програм, таких як серверні програми для підприємств Java EE (Enterprise Edition), управляє об’єктами та