

## ДОСЛІДЖЕННЯ МОДЕЛІ ПОБУДОВИ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ ЗА ДОПОМОГОЮ БЕЗСЕРВЕРНИХ ОБЧИСЛЕНЬ

Гавриш Д. Л., Скібін О. О.

Науковий керівник – д.т.н., проф. Смеляков К. С.

Харківський національний університет радіоелектроніки, каф. ПІ

м. Харків, Україна

e-mail: [dmytro.havrysh@nure.ua](mailto:dmytro.havrysh@nure.ua)

This theses proposes an intelligent cross-border logistics system using IoT technology to enhance efficiency and security. Traditional systems face challenges like high computing loads and security risks. To overcome this, the paper suggests a platform with serverless architecture and microservices, distributing services across logistics companies. It employs API gateways for security and asymmetric encryption for data security. Experiments show reduced costs and complexity, improved security, throughput, and latency in cross-border transactions.

У зв'язку з поступовим розвитком глобальної економіки та загостренням процесу глобалізації, міжнародна торгівля стала невід'ємною складовою сучасної економіки. Логістика, що відіграє ключову роль у сприянні міжнародної торгівлі, постійно прогресує та інновується, сприяючи розвитку інтелектуалізації та цифровізації міжнародної логістики. Поява технології "Інтернет речей" (IoT) призвела до поширення міжзв'язаних пристроїв та систем, що охоплюють широкий спектр сенсорів, розумних пристроїв та систем управління логістикою, задіяних у міжнародних логістичних операціях. Значна кількість даних, що генерується цими пристроями та системами, дозволяє ефективну взаємодію та обмін інформацією через Інтернет, що значно підсилює прогрес інтелектуалізації міжнародної логістики.

З поширенням інтеграції технологій питання безпеки IoT отримало підвищену увагу. Взаємопов'язаність пристроїв та систем у межах IoT потребує підключення до Інтернету, за допомогою якого запити з розподілених пристроїв централізовано обробляються та відповідаються. Однак такий централізований підхід має в собі вроджені вразливості безпеки, включаючи мережеві атаки, витіки даних, порушення конфіденційності та схильність до порушень безпеки системи. Особливо в області міжнародної логістики інформаційна безпека та захист стають ще більш критичними та складними через участь міжнародної торгівлі, митниць та факторів транспортування.

Децентралізовані характеристики сценаріїв IoT ефективно гармонізуються з розподіленою архітектурою застосунків. Два провідні парадигми проектування архітектури програмного забезпечення, а саме мікросервісна архітектура та "подієво-орієнтованих" архітектур, в якому є безсерверні обчислення, здобули широке використання в різних галузях як

невід'ємні складові розподіленої архітектури застосунків. Мікросервісна архітектура сприяє побудові високо гнучких та масштабованих систем, перевершуючи складні централізовані методи обробки з точки зору масштабованості, підтримуваності та розширюваності. Відповідно, вона пропонує вищу адаптабельність до змін бізнес-вимог, покращуючи надійність та продуктивність системи. З іншого боку, архітектура яка використовує безсерверні обчислення пропонує більш гнучкий та універсальний підхід до розробки, що прискорює впровадження нових застосунків, при цьому зменшуючи витрати на розробку та експлуатацію.

Поява безсерверних обчислень, як нового шаблону розгортання програмного забезпечення, приносить значні переваги, такі як операції без обслуговування та моделі вартості "плати-лише-за-використання". Тим часом, мікросервісна архітектура, відмінна від традиційної монолітної архітектури програмного забезпечення, з'явилася для вирішення вимог сучасних інтернет-сервісів, включаючи аспекти високої конкурентоспроможності, високої продуктивності та високої доступності [1]. Вона має значну економічну цінність [2]. Хоча мікросервісна архітектура набули популярності як парадигми розробки [3] розподілених архітектур програмного забезпечення, виникають виклики та складнощі при їх застосуванні в контексті безпеки IoT міжнародної логістики. Один із значних викликів полягає в забезпеченні безпеки та конфіденційності комунікації між мікросервісами та функціями. Оскільки кожна служба в мікросервісній архітектурі має незалежний характер, різні вимоги безпеки та обмеження приватності можуть застосовуватися до різних служб. У цьому відношенні встановлення розумних контролів доступу та дозволів на дані для кожного мікросервісу та функції є надзвичайно важливим для запобігання несанкціонованому доступу та витоку інформації. Крім того, служби та функції в мікросервісній та "подієво-орієнтованих" архітектурі піддаються різноманітним загрозам безпеки, включаючи атаки типу "відмова в обслуговуванні" та атаки на внесення змін у інформацію. В області міжнародної логістики ці атаки можуть мати руйнівні наслідки, такі як втрата інформації.

Під час процесу проектування архітектури виникла складність у оцінці обсягу трафіку системи через введення нового бізнес-сценарію. Розгортання додатків безпосередньо на хмарних серверах за допомогою традиційних підходів потребувало би ручного горизонтального масштабування для адаптації до недостатнього трафіку, що може призвести до втрати ресурсів, якщо заздалегідь буде надано декілька серверів. Для вирішення цієї проблеми та відповідно до вимог бізнесу, була прийнята мікросервісна архітектура з безсерверними обчисленнями на основі AWS для реалізації. Шляхом розгортання мікросервісів на AWS Lambda та використання розподіленого обчислення, додаток може автоматично масштабуватися еластично в залежності від обсягу трафіку, а

витрати розраховуються відповідно до кількості запитів та часу обчислення. Для оперативної відповіді на змінні вимоги користувачів, підвищення ефективності розробки та прискорення поставки системи важливим є впровадження процесів безперервної інтеграції та безперервної доставки (CI/CD) для автоматизації всього процесу випуску. Автоматизація завдань, таких як компіляція коду, побудова та розгортання, мінімізує необхідність у ручному контролі кожної зміни та зменшує ризику системи шляхом превентивного виявлення потенційних проблем. Цей процес автоматизації охоплює не лише кодові дії, а й динамічне створення, знищення та оновлення ресурсів, необхідних для роботи системи, включаючи сервери та бази даних. Збирати та зіставляти інформацію з усіх ресурсів AWS, програм та служб, що працюють на aws та локально допомагає сервіс моніторингу CloudWatch [4], Lambda Insights надає автоматичні інформаційні панелі в консолі CloudWatch. На рис 1 наведено архітектуру для моніторингу за допомогою CloudWatch.

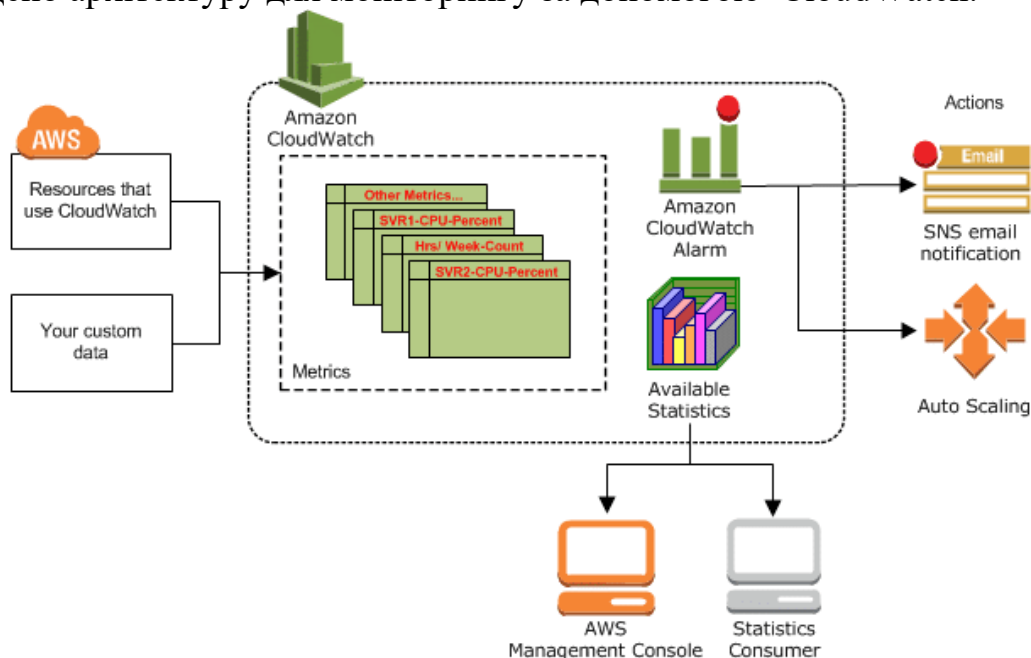


Рисунок 1 – Схема використання сервісу CloudWatch

Цей додаток спрямований на полегшення та оптимізацію процесів у сфері міжнародної логістики. Він забезпечує автоматизацію обробки даних, координацію та візуалізацію інформації про логістичні операції, що дозволяє зменшити час і зусилля, затрачені на вирішення рутинних завдань. Додаток також забезпечує безпечний обмін даними між учасниками логістичних процесів, забезпечуючи конфіденційність та цілісність інформації. Крім того, він надає можливість аналізу даних та надає рекомендації для оптимізації процесів, що дозволяє підприємствам зменшити витрати та підвищити ефективність своєї діяльності.

Оцінка якості моделі базується на конкретних критеріях, які включають:

1. Цілісність: Модель повинна враховувати всі аспекти міжнародної логістики, включаючи транспорт, митні процедури, складське господарство та інші, щоб забезпечити повноту та реалістичність аналізу.

2. Доступність: Важливо, щоб модель була легко доступною для користувачів у будь-який час і з будь-якого місця, щоб забезпечити зручність та ефективність використання.

3. Швидкодія: Модель повинна працювати швидко та ефективно, забезпечуючи миттєвий доступ до результатів аналізу та інформації для прийняття рішень.

4. Точність: Важливо, щоб модель надавала точні результати аналізу та прогнозування, щоб користувачі могли робити інформовані рішення.

Ці критерії допомагають забезпечити, що модель відповідає вимогам користувачів та сприяє покращенню ефективності логістичних процесів.

Безсерверні та мікросервіси є архітектурно пов'язаними технологіями, які використовують різні методи. Безсерверні та мікросервіси підкреслюють масштабованість, адаптивність, економічну ефективність і простоту додавання нових функцій на відміну від монолітного дизайну.

Оскільки кожен сервіс функціонує як незалежний додаток, довгострокова масштабованість є основною метою мікросервісів.

Залежно від обсягу продукції та пріоритетів організації можна вибрати одну з двох стратегій.

Мікросервіси нададуть вам безсерверні мікросервіси для довгострокових рішень, якщо ви збираєтеся побудувати велику платформу, яка потребує постійного зростання.

Безсерверна архітектура становить чудовий варіант для тих, хто прагне здійснити розгортання швидко та економічно.

Список використаних джерел:

1. Сербулл М.О. Бітченко О.М Мікросервісна архітектура онлайн платформи для інвестиційного менеджменту, Харків: ХНУРЕ – 2 с.

2. Singleton, A. The Economics of Microservices. IEEE Cloud Comput. 2016 С. 16–20.

3. Li, D.; Duan, J.; Yao, Y.; Qian, S.; Zhou, J.; Xue, G.; Cao, J.; Ansari, M.D. SoDa: A Serverless-Oriented Deadline-Aware Workflow Scheduling Engine for IoT Applications in Edge Clouds. Wirel. Commun. Mob. Comput. 2022, 2022, 7862911.

4. AWS CloudWatch Walkthrough With Realtime Usecase. Dheeraj Choudhary's Blog. URL: <https://dheeraj3choudhary.com/aws-cloudwatch-walkthrough-with-realtime-usecase> (дата звернення: 05.03.2024).