ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.624
DOI https://doi.org/10.36994/2788-5518-2024-02-08-01

СЦЕНАРІЙ ВИКОРИСТАННЯ АНАЛІТИКИ КЛІНІЧНОГО ІНТЕРНЕТУ РЕЧЕЙ (IOT)

Кирсанов О. О., *Харківський національний університет радіоелектроніки, Харків, Україна. https://orcid.org/ 0009-0001-5987-4728, oleksandr.kyrsanov@nure.ua*

Кривенко С. А., *к.т.н., Харківський національний університет радіоелектроніки, Харків, Україна. https://orcid.org/0000-0002-5244-1276, stanislav.kryvenko@nure.ua*

Анотація. *Стаття присвячена сценарію використання аналітики Інтернету речей (IoT) для збору та аналізу даних з електровелосипедів, обладнаних IoT-пристроями, з метою підвищення фізичної активності населення, що сприятиме профілактиці серцево-судинних захворювань. Запропонована архітектура системи базується на AWS і охоплює три ключові рівні: прийом та обробку даних, їхнє зберігання, а також аналіз і візуалізацію результатів.*

На першому етапі підключення IoT-пристроїв до AWS IoT Core забезпечує передачу даних з електровелосипедів через протокол MQTT. Дані регулярно надходять до Kinesis Data Firehose, де вони можуть бути збагачені функціями Lambda, які додають до них додаткову інформацію, зокрема геолокаційні дані про користувачів. Це забезпечує більш точний і деталізований аналіз зібраних даних.

Другий етап включає налаштування OpenSearch Service для зберігання та індексації отриманих даних, що дозволяє проводити детальний аналіз за допомогою інструментів OpenSearch Dashboards. У результаті створюються різноманітні візуалізації, такі як кругові діаграми, теплові карти, гістограми та інші графічні інструменти, які дозволяють не лише відстежувати використання електровелосипедів, а й проводити аналіз впливу програми на фізичну активність громадян, а також визначати ключові фактори, що впливають на ефективність цієї програми.

На завершальному етапі результати аналізу надаються зацікавленим сторонам через зручні інформаційні панелі, доступ до яких забезпечується за допомогою Amazon Cognito та AWS Identity and Access Management (IAM). Це гарантує безпечний і надійний доступ до даних, що є особливо важливим для прийняття обґрунтованих управлінських рішень. Запропонована архітектура демонструє значний потенціал IoT у клінічній аналітиці, сприяючи ефективному використанню зібраних даних для покращення здоров'я населення. Таким чином, впровадження цього підходу підвищує ефективність управління програмами фізичної активності, а також сприяє запобіганню захворюванням, що дозволяє суттєво покращити рівень громадського здоров'я та зробити цей підхід особливо актуальним у сучасних умовах швидкого розвитку технологій.

Ключові слова: *IoT, мобільний пристрій, смартфон, адаптивний кейс-менеджмент, структуровані та неструктуровані дані.*

A USE CASE FOR INTERNET OF THINGS (IOT) ANALYTICS

Oleksandr Kyrsanov, *Kharkiv National University of Radio Electronics, Kharkiv, Ukraine. https://orcid.org/0009-0001-5987-4728, oleksandr.kyrsanov@nure.ua*

Stanislav Kryvenko, *Ph.D., Ass. Prof., Kharkiv National University of Radio Electronics, Kharkiv, Ukraine. https://orcid.org/0000-0002-5244-1276, stanislav.kryvenko@nure.ua*

Abstract. *The article is dedicated to a scenario of using Internet of Things (IoT) analytics for collecting and analyzing data from electric bicycles equipped with IoT devices, aiming to enhance public health by increasing physical activity, which contributes to the prevention of cardiovascular diseases. The proposed system architecture is based on AWS and encompasses three key levels: data ingestion and processing, storage, and analysis with visualization of the results.*

The first stage involves connecting IoT devices to AWS IoT Core, enabling the regular transmission of data from electric bicycles via the MQTT protocol. This data is transmitted to Kinesis Data Firehose, where it can be further enriched using Lambda functions, which add supplementary information, such as geolocation data about users. This enrichment process ensures a more accurate and detailed analysis of the collected data.

The second stage involves configuring OpenSearch Service for storing and indexing the received data, allowing for in-depth analysis using OpenSearch Dashboards tools. Consequently, various visualizations are created, such as pie charts, heatmaps, histograms, and other graphical tools, which not only help track bicycle usage but also analyze the impact of the program on citizens' physical activity. Additionally, these visualizations help identify key factors influencing the program's effectiveness, providing valuable insights for optimization.

In the final stage, the analysis results are presented to stakeholders through user-friendly dashboards, with access secured via Amazon Cognito and AWS Identity and Access Management (IAM). This ensures safe and reliable data access, which is crucial for making informed management decisions. The proposed architecture highlights the significant potential of IoT in clinical analytics, contributing to the effective use of collected data to improve public health. Thus, implementing this approach enhances the efficiency of managing physical activity programs, as well as contributes to disease prevention, significantly improving the level of public health and making this approach especially relevant in the context of rapid technological advancement.

Key words: *IoT, mobile device, Adaptive Case Management, security, data protection, UI/UX, structured and unstructured data.*

Вступ

У цій роботі розглядається сценарій використання аналітики Інтернету речей (IoT). Дослідження пропонує архітектуру для підтримки рівня прийому та обробки, рівня зберігання та рівня аналізу та візуалізації.

Проблема бізнесу

Сценарій, представлений тут, відповідає одному варіанту планування конвеєра. Важливо працювати у зворотному напрямку від бізнес-проблеми, щоб визначити, який тип конвеєра потрібно створити.

Робота в зворотному напрямку від бізнес-проблеми

Стартап-компанія з електровелосипедів співпрацює з місцевою владою, щоб запровадити пілотну програму прокату електровелосипедів у кількох приміських районах.

Мета влади – профілактика серцево-судинних захворювань на основі підвищення фізичної активності громадян.

На цьому етапі їм потрібно зібрати й проаналізувати тенденції у даних про те, як використовуються велосипеди. Їм також потрібно визначити, які дані вони хочуть відстежувати, розширюючи програму.

Вони знають, що вони хочуть дивитися на те, де подорожували велосипеди, час роботи акумулятора, рівні використання двигуна (вимкнено, низький, середній, високий) та життєві показники велосипедистів.

Кожен із 50 велосипедів оснащений пристроєм Інтернету речей (IoT), який живиться від акумулятора велосипеда та використовує бездротову телефонну мережу для передачі даних. Пристрої використовують протокол повідомлень MQTT для надсилання та отримання даних JSON. Наразі пристрої налаштовані на надсилання повідомлення кожні 2 хвилини [6].

Потрібно створити конвеєр для підтвердження концепції, щоб збирати й аналізувати дані з пристроїв Інтернету речей, вбудованих в електровелосипеди.

Які ключові характеристики описані в бізнес-проблемі, яка керуватиме рішеннями, які ви приймаєте для створення конвеєра?

Виконання досліджень

Еталонна архітектура IoT

Рисунок 1 ілюструє пропоновану архітектуру AWS для програми для підтвердження концепції.

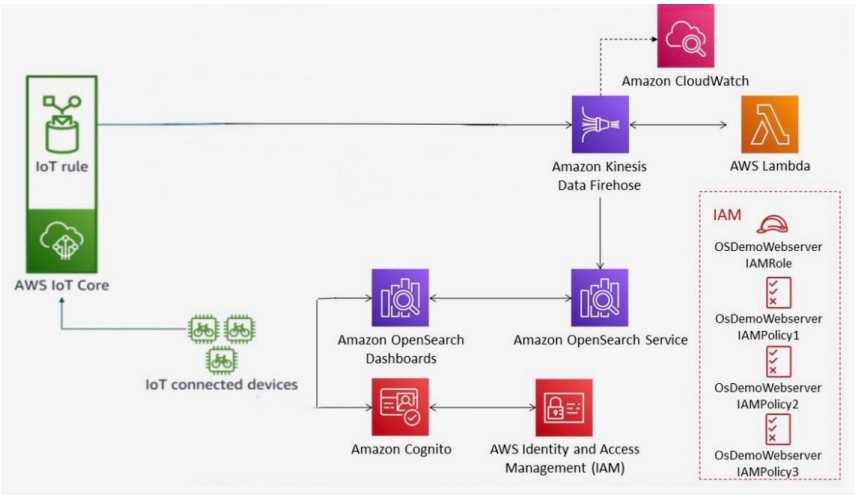

Рис. 1. Пропонована архітектура AWS

Пристрої IoT на велосипедах надсилатимуть повідомлення до AWS IoT Core. Правило IoT трансформує повідомлення та надсилає їх у потік доставки Amazon Kinesis Data Firehose. Потік доставки налаштовано як підписник на тему та може виконувати додаткові перетворення, щоб підготувати дані для аналізу. Kinesis Data Firehose забезпечує захоплення потокових даних з журналів вебсервера та їх передачу до OpenSearch Service для подальшого аналізу. Потік доставки налаштований на використання джерела даних «Direct PUT» та місця призначення «Amazon OpenSearch Service». Lambda обробляє та збагачує дані в режимі реального часу, додаючи додаткову інформацію, таку як географічне розташування відвідувачів на основі їхніх IP-адрес. Функція Lambda «os-demo-lambda-function» пов'язана з потоком доставки Kinesis Data Firehose і використовується для трансформації журналів доступу перед їх передачею до OpenSearch Service. OpenSearch Service індексує та зберігає дані, даючи можливість аналізувати їх за допомогою OpenSearch Dashboards. За допомогою Dashboards можна створювати різні візуалізації для відображення даних у зручному форматі [3].

Прийом та переробка

По-перше, розглядається рівень прийому та обробки.

Підключення пристроїв до AWS IoT Core

Схема підключення пристроїв до AWS IoT Core і початку публікації повідомлень на визначену тему наведена на рисунку 2.

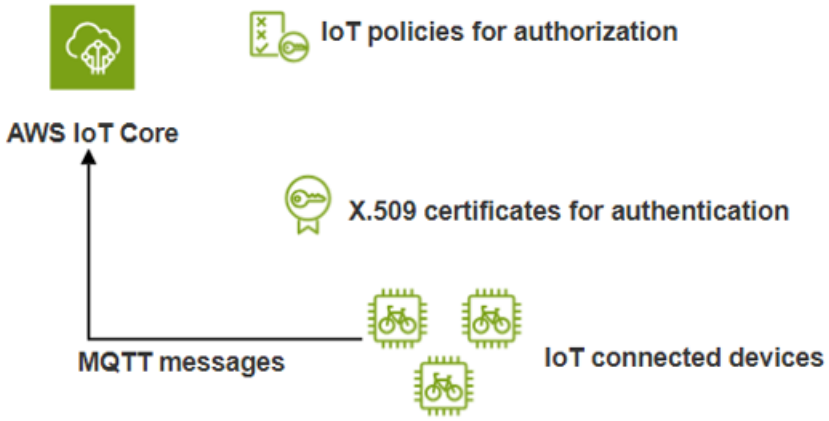

Рис. 2. Підключення пристроїв до AWS IoT Core

Для цього прикладу темою є ebike/stats. Підключалися фактичні пристрої для тестування та використовувались тестові дані із симулятором пристрою, щоб перевірити потік. В AWS IoT Core використовувались параметри в меню «Підключити», щоб підключити один або багато пристроїв. Це передбачає створення об'єктів-речей в AWS IoT Core для представлення пристроїв. Також налаштовані сертифікат безпеки, який контролює доступ до пристрою, і IoT-політики, необхідні для контролю доступу до ресурсів AWS IoT. За допомогою тестового клієнта MQTT у AWS IoT Core підписалися на тему ebike/stats, щоб переконатися, що повідомлення з пристрою публікуються в темі. Під час використання пристроїв IoT задіяні компоненти безпеки. Як правило, пристрої використовують сертифікати X.509 для захисту доступу між пристроями IoT. AWS IoT Core використовує ці сертифікати для захисту зв'язку між пристроями та AWS IoT Core, а також для автентифікації пристроїв IoT. Політики AWS IoT Core надають дозволи для авторизації дій у AWS IoT Core, наприклад підключення до брокера повідомлень. Основні політики AWS IoT є документами JSON і дотримуються тих самих умов, що й політики AWS Identity and Access Management (IAM). Як політики AWS IoT Core, так і політики IAM використовуються з AWS IoT Core для керування операціями, які може виконувати особа (також звана принципалом). Політики IoT використовуються для авторизації пристроїв із сертифікатами X.509, а політики IAM додаються до користувача, групи або ролі IAM [5].

Приклад: повідомлення MQTT

На рис. 3 показано приклад того, як повідомлення MQTT, надіслане до теми ebike/stats, може надати атрибути для такого пристрою IoT.

Налаштування правила IoT для маршрутизації та фільтрації повідомлень

Коли дані пристрою публікуються та пристрої успішно підключилися до AWS IoT Core, було створено правило (рис. 4) для вибору повідомлень і їх маршрутизації передплатникам.

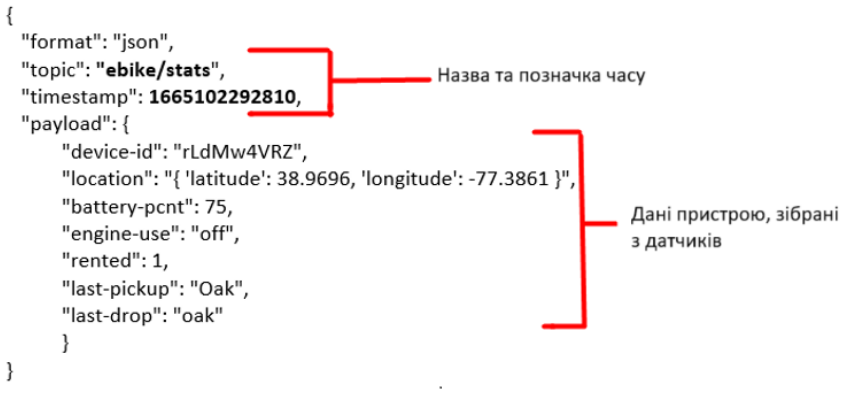

Рис. 3. Повідомлення MQTT

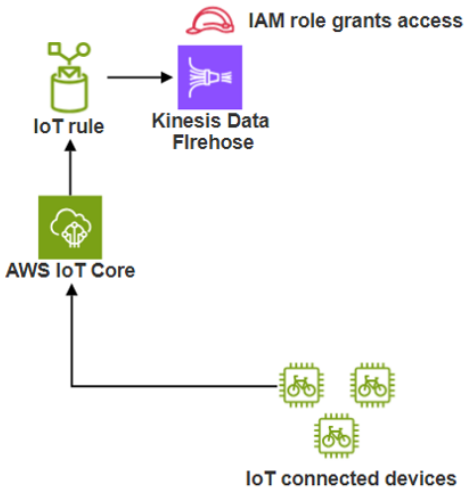

Рис. 4. Правило IoT

У цьому прикладі правило направляло повідомлення з теми ebike/stats до потоку доставки Kinesis Data Firehose. Роль IAM використовувалась для надання дозволів правилу IoT для запису даних у потік доставки.

Приклад: правило IoT

У цьому прикладі (рис. 5) наведені компоненти, які становлять правило IoT – інструкцію SELECT і дію, яку потрібно виконати. Інструкція SELECT визначає, які атрибути даних слід включити.

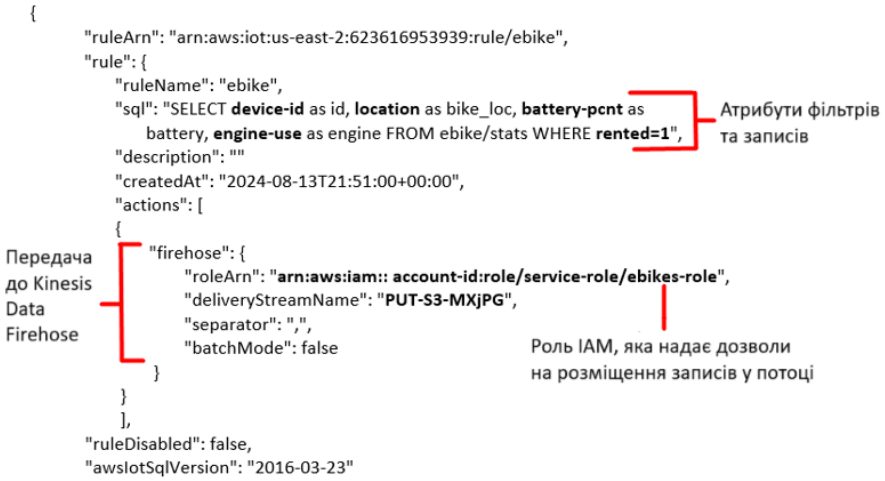

Рис. 5. Компоненти правила IoT

Речення FROM визначає відповідну тему MQTT, і речення WHERE використовується, щоб фільтрувати, які записи слід включити. Розділ дій вказує правилу, куди направляти результати запиту. У цьому прикладі оператор SELECT обмежує атрибути, які потрібно включати (ідентифікатор пристрою, розташування, акумулятор-pcnt і двигун-використання) і зіставляє їх з різними іменами полів. Правило також може доповнювати дані, додаючи додаткові атрибути [1]. Наприклад, було включено поле для ідентифікації цих записів як частину пілотної програми, оновлено оператор SELECT, щоб включити «пілотний етап як стадію» після «двигун-використання як механізм». Речення FROM вказує на тему запитувати – ebike/stat у цьому прикладі. Речення WHERE обмежує обсяг записів лише тими, де значення rented дорівнює 1.

Дії в цьому прикладі повідомляють правилу про доставку результатів запиту до потоку доставки Kinesis Data Firehose, який називається PUT-S3-MXjPG. У цьому прикладі користувач вирішив створити потік доставки як частину створення правила в консолі керування AWS. Це полегшило створення потоку та дозволів IAM, необхідних для доступу до потоку.

Потік доставки

Рішення використовує Kinesis Data Firehose для завантаження потокових журналів доступу до вебсервера, а потім використовує функції Lambda для збагачення даних [2]. Після використання Kinesis та Lambda для обробки даних використовувався OpenSearch Service для завантаження та індексації даних. Використовуючи OpenSearch Dashboards, створюються візуалізації даних, щоб отримати уявлення про стан користувачів електровелосипедів. Візуалізації поширюються серед зацікавлених сторін, використовуючи Amazon Cognito як інструмент аутентифікації та AWS Identity and Access Management (IAM) для авторизації доступу до інформаційної панелі.

Налаштування Kinesis Data Firehose

Наступним етапом було налаштування Kinesis Data Firehose. Для цього було використано консоль Kinesis та переглянуто конфігурацію потоку доставки os-demo-firehose-stream. У пошуковому рядку праворуч від «Services» вибрано «Kinesis». У лівій навігаційній панелі вибрано «Delivery streams» та натиснуто на потік доставки «os-demo-firehose-stream». Переглянуто розділ «Delivery stream details», де було вказано «Source» як «Direct PUT» і «Destination» як «Amazon OpenSearch Service». Якщо прокрутити сторінку вниз, можна було помітити увімкнений моніторинг, який генерував метрики потоку доставки. Також переглянуто конфігурацію трансформації даних. У нижній частині сторінки деталей потоку доставки було вибрано вкладку «Configuration» та розділ «Transform records». Там знаходиться функція Lambda «os-demo-lambda-function», яка використовується для трансформації журналів доступу перед їх передачею до OpenSearch Service.

Налаштування OpenSearch Service

Конфігурація кластера OpenSearch Service здійснювалася через консоль OpenSearch. На вкладці «Configuration» для потоку доставки у розділі «OpenSearch Service destination» під «Domain» було вибрано посилання «os-demo». Відкрилася консоль OpenSearch Service, де було переглянуто конфігурацію кластерата налаштування безпеки. URL-адреса OpenSearch Dashboards знадобилася для подальшого використання. Також на вкладці «Security configuration» переглянуто політики доступу, які використовували дві ролі: os_demo_firehose_delivery_role та osdemocognitoauthuserrole.

Створення індексу OpenSearch

Для створення індексу OpenSearch використано OpenSearch Dashboards. Для входу було використано такі облікові дані: ім'я користувача LabUser та пароль Passw0rd1!. Після входу система запропонувала змінити пароль. Було введено новий пароль Passw0rd1!2. Після входу видалено наявні журнали вебсервера. Для цього через значок меню запущено «Dev Tools». У консолі Dev Tools командою DELETE /apache_logs було видалено індекс, якщо він існував. Далі здійснено створення нового індексу за допомогою методу PUT. Приклад коду для створення індексу наведено нижче (рис. 6):

```
PUT apache_logs
{
  "settings": {
    "index": {
      "number_of_shards": 10,
      "number_of_replicas": 0
    }
  },
  "mappings": {
    "properties": {
      "agent": {
        "type": "text"
```

Рис. 6. Код для створення індексу

```
        },
        "browser": {
          "type": "keyword"
        },
        "bytes": {
          "type": "text"
        },
        "city": {
          "type": "keyword"
        },
        "country": {
          "type": "keyword"
        },
        "datetime": {
          "type": "date",
          "format": "dd/MMM/yyyy:HH:mm:ss Z"
        },
        "host": {
          "type": "text"
        },
        "location": {
          "type": "geo_point"
        },
        "referer": {
          "type": "text"
        },
        "os": {
          "type": "keyword"
        },
        "request": {
          "type": "text"
        },
        "response": {
          "type": "text"
        },
        "webpage": {
          "type": "keyword"
        },
        "refering_page": {
          "type": "keyword"
        }
      }
    }
}
```

Рис. 6. Код для створення індексу (продовження)

Відобразилася така відповідь (рис. 7):

```
{
  "acknowledged": true,
  "shards_acknowledged": true,
  "index": "apache_logs"
}
```

Рис. 7. Результат створення індексу

Ця команда створила новий індекс під назвою apache_logs. Коли журнали вебсервера оновлювалися через трафік вебсайту, потік доставки Kinesis Data Firehose наповнював кластер OpenSearch Service даними на основі відповідностей у команді. Команда встановила типи даних для полів

у файлах журналів сервера у базі даних OpenSearch Service. Після створення індексу було здійснено генерацію журналів доступу до вебсервера, а потім створено візуалізації на основі даних у цьому індексі [4].

Генерація журналів доступу до вебсервера

Після створення індексу було згенеровано журнали доступу до вебсервера. Це здійснювалося шляхом відвідування вебсайту (рис. 8), розміщеного на EC2-інстансі, з використанням різних веб-браузерів.

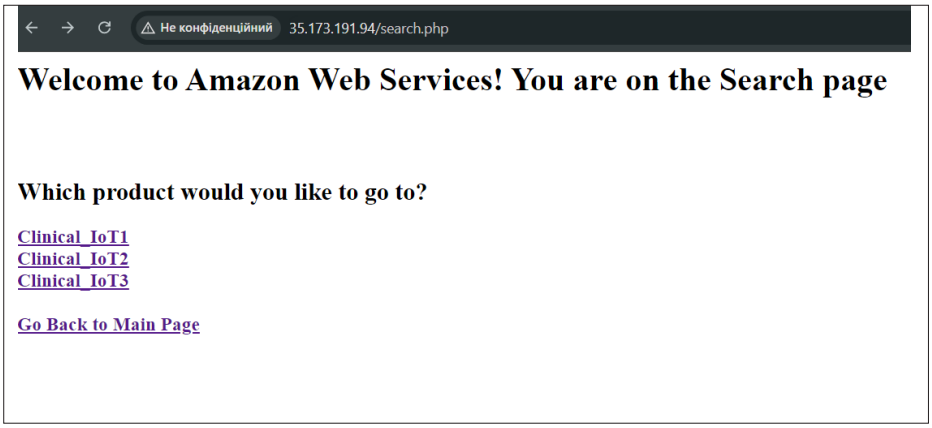

Рис. 8. Результат відвідування вебсайту

Для цього використовувалася нова вкладка або вікно браузера та здійснено перехід за URL-адресою http://<PUBLIC-IP>/main.php, замінивши <PUBLIC-IP> на публічну IP-адресу, яку було збережено раніше.

Шляхом симуляції користувацької активності в кількох веб-браузерах, згенеровано журнали вебсервера, відвідуючи різні сторінки сайту.

Спостереження за подіями журналу в CloudWatch

Для спостереження за процесом завантаження та обробки даних використовувалися журнали CloudWatch. Вони дозволяють побачити, як журнали доступу відвідувачів завантажувалися з вебсервера до Kinesis Data Firehose, збагачувалися функцією Lambda і передавалися до OpenSearch Service для подальшого аналізу [2]. Повернувшись до AWS Management Console, було вибрано «CloudWatch». У лівій навігаційній панелі було вибрано «Logs» > «Log groups» і групу журналів /aws/lambda/aes-demo-lambda-function. Переглянуто потік журналів (рис. 9), вибравши останній потік журналів.

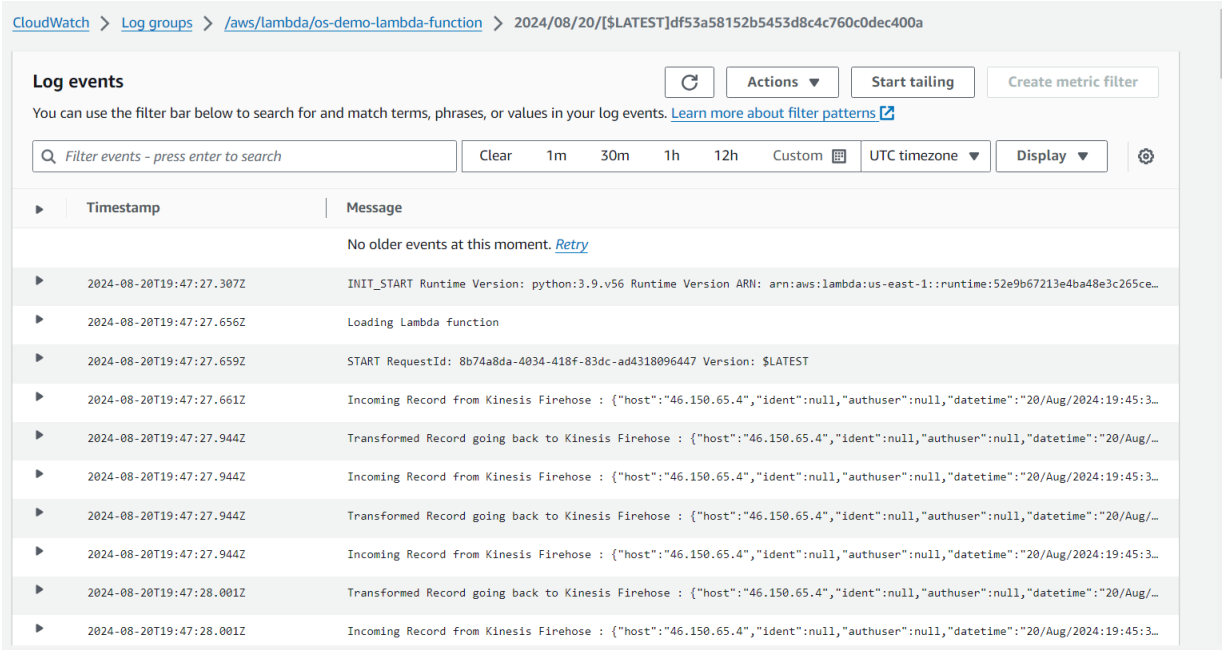

Рис. 9. Потік журналів

Деталі подій журналів було переглянуто, розгорнувши журнали з повідомленням «Incoming Record from Kinesis Firehose», щоб побачити початкові дані журналу доступу, та «Transformed Record going back to Kinesis Firehose», щоб побачити збагачені дані журналу. Деталі події подібні до таких (рис. 10):


```
Incoming Record from Kinesis Firehose :
{
    "host": "46.150.65.4",
    "ident": null,
    "authuser": null,
    "datetime": "20/Aug/2024:19:45:32 +0000",
    "request": "GET /Clinical_IoT1.php HTTP/1.1",
    "response": "200",
    "bytes": "297",
    "referer": "http://35.173.191.94/search.php",
    "agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/127.0.0.0 Safari/537.36"
}
Transformed Record going back to Kinesis Firehose :
{
"host": "46.150.65.4",
    "ident": null,
    "authuser": null,
    "datetime": "20/Aug/2024:19:45:32 +0000",
    "request": "GET /Clinical_IoT1.php HTTP/1.1",
    "response": "200",
    "bytes": "297",
    "referer": "http://35.173.191.94/search.php",
    "agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/127.0.0.0 Safari/537.36",
    "webpage": "Clinical_IoT1",
    "refering_page": "search",
    "city": "Zaporizhia",
    "country": "Ukraine",
    "browser": "Chrome",
    "os": "Windows",
    "location": {
        "lat": 47.85,
        "lon": 35.2833
  }
}
```


Рис. 10. Збагачені дані журналу

Аналіз та візуалізація

У цьому підрозділі описано, як можна використовувати OpenSearch Dashboards для роботи з даними IoT, які наповнювали кластер OpenSearch Service.

Створення шаблону індексу OpenSearch

Після генерування журналів доступу та спостереження за їх обробкою було створено шаблон індексу в OpenSearch Dashboards. Це дозволило ідентифікувати індекси OpenSearch, які використовувалися для створення візуалізацій. Для цього на сторінці OpenSearch Dashboards вибрано «Discover», потім «Create index pattern» та введено apache_logs як ім'я шаблону індексу. На другому кроці вибрано поле datetime для фільтрації даних та завершено створення шаблону (табл. 1).

Створення кругової діаграми

Кругові діаграми є важливим інструментом для візуалізації категорійних даних, оскільки вони дозволяють легко порівнювати частки різних сегментів. У контексті вебаналітики кругові діаграми допомагають візуалізувати дані про операційні системи та браузери, які використовували відвідувачі вебсайту [2]. Нижче описано покроковий процес створення кругової діаграми в OpenSearch Dashboards.

Спочатку здійснено вхід до OpenSearch Dashboards. У головному меню було вибрано розділ «Visualize» для створення нової візуалізації. Натиснуто кнопку «Create new visualization» і вибрано тип візуалізації «Pie».

Таблиця 1
Шаблон індексу в OpenSearch Dashboards

Ім'я	Тип	Формат	З пошуком	Агреговано	Виключено
id	string		✓	✓	
index_	string		✓	✓	
score	number			✓	
source	source				
type	string		✓	✓	
agent	string		✓		
browser	string		✓	✓	
bytes	string		✓		
city	string		✓	✓	
country	string		✓	✓	

Після вибору типу візуалізації відкрилося вікно для вибору джерела даних. Було вибрано індекс apache_logs*, що був створений раніше. Це забезпечило доступ до даних журналів вебсервера для подальшого аналізу.

У верхньому правому куті інтерфейсу було налаштовано часовий інтервал для аналізу даних. Вибрано опцію «Last 30 minutes» для обмеження даних до останніх 30 хвилин. Це дозволило отримати актуальні дані про відвідування вебсайту.

Кошики використовувалися для категоризації даних у круговій діаграмі. Для створення першого кошика було вибрано розділ «Buckets» і натиснуто «Add» > «Split slices». У полі «Aggregation» вибрано «Terms», а в полі «Field» – «webpage». Це налаштування дозволило розділити дані за різними вебсторінками.

Залишено значення за замовчуванням для «Order by», «Order» та «Size». Увімкнуто опцію «Group other values in separate bucket», щоб групувати інші значення в окремий кошик. Це забезпечило відображення всіх вебсторінок, навіть якщо деякі з них мали мало даних. Далі додано ще один кошик для категоризації даних за операційними системами. Знову вибрано «Add» > «Split slices» і «Terms» у полі «Aggregation». У полі «Field» натиснуто «os». Увімкнуто опцію «Group other values in separate bucket».

Останнім кошиком, який потрібно було додати, став кошик для категоризації даних за типом браузера. Повторено попередні кроки, «Add» > «Split slices» і «Terms» у полі «Aggregation». У полі «Field» натиснуто «browser». Увімкнуто опцію «Group other values in separate bucket».

Для більш детального аналізу застосовано фільтри до кругової діаграми. Наприклад, щоб відобразити лише дані про певну вебсторінку, вибрано «Add filter» у верхньому лівому куті. У полі «Field» вибрано «webpage», у полі «Operator» натиснуто «is», а у полі «Value» введено «kindle». Далі натиснуто «Save» для застосування фільтра. Для зміни стилю кругової діаграми на «donut» та налаштування інших параметрів вибрано «Options» на панелі праворуч. Опції «Donut» та «Show top level only» вимкнуто. Увімкнуто опцію «Show labels» для відображення міток на діаграмі. Натиснуто «Update» для застосування змін (рис. 12).

Ця візуалізація ілюструє відсоток переглядів сторінки продукту з різних браузерів та операційних систем. Після створення діаграми можна взаємодіяти з нею для отримання додаткових інсайтів. Курсор можно навести на кожен сегмент діаграми, щоб побачити деталі про дані, що включені до сегмента. Додано інші фільтри для перегляду даних за різними параметрами, такими як тип браузера або операційна система.

Ця кругова діаграма (рис. 13) подібна до попередньої, але включала фільтр для певного браузера.

Створення теплової карти

Теплова карта дає можливість дізнатися, чи більше клієнтів переходило на сторінки продуктів з пошукової сторінки або зі сторінки рекомендацій [2].

Для створення теплової карти використано OpenSearch Dashboards. У головному меню вибрано розділ «Visualize» для створення нової візуалізації. Натиснуто кнопку «Create new visualization» і тип візуалізації «Heat Map».

Після вибору типу візуалізації відкрилося вікно для вибору джерела даних. Було вибрано індекс apache_logs*, що був створений раніше. Це забезпечило доступ до даних журналів вебсервера для подальшого аналізу.

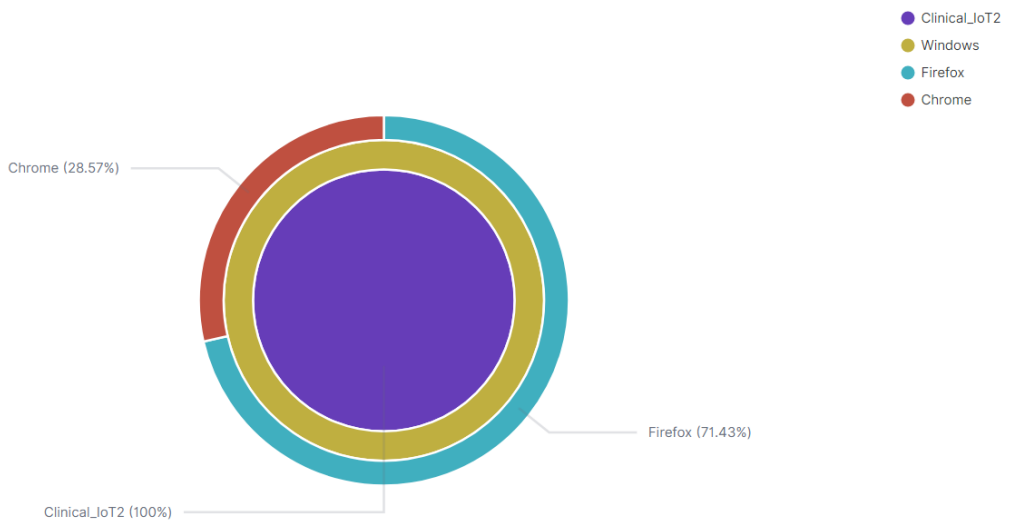

Рис. 12. Кругова діаграма

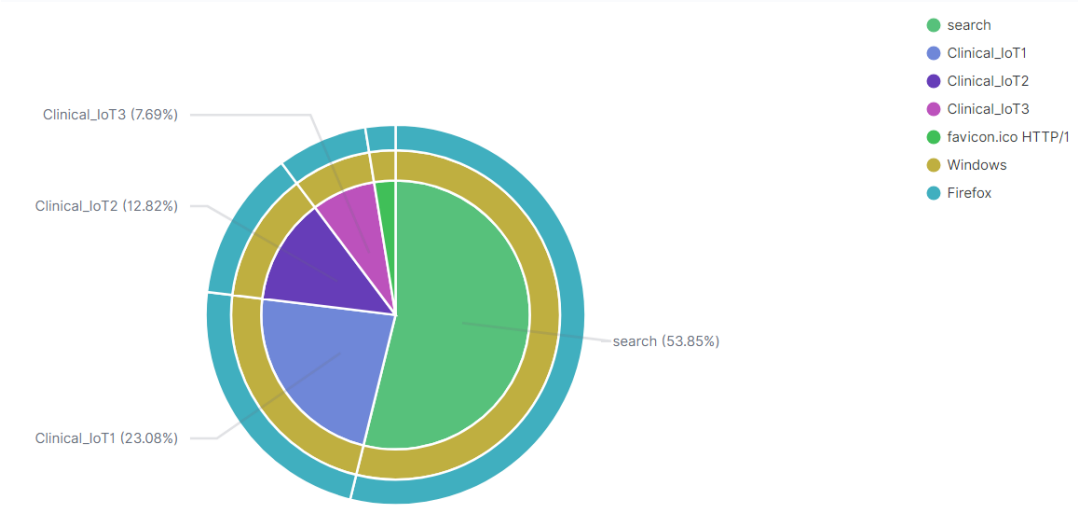

Рис. 13. Фільтр для браузера Firefox

Теплова карта використовує дві осі для відображення взаємозв'язків між різними категоріями даних. Почато з налаштування X-осьового кошика. У розділі «Buckets» натиснуто «Add» > «X-axis». У полі «Aggregation» вибрано «Terms», а в полі «Field» – «refering_page». Це дозволило розділити дані за сторінками, з яких користувачі переходили. Залишено значення за замовчуванням для «Order by», «Order» та «Size». Увімкнуто опцію «Group other values in separate bucket», щоб групувати інші значення в окремий кошик. Це забезпечило відображення всіх сторінок переходів, навіть якщо деякі з них мали мало даних.

Наступним кроком було налаштування Y-осьового кошика для категоризації даних за відвідуваними вебсторінками. Знову обрано «Add» > «Y-axis» і «Terms» у полі «Aggregation». У полі «Field» натиснуто «webpage». Залишено значення за замовчуванням для «Order by», «Order» та «Size» та увімкнуто опцію «Group other values in separate bucket». Після налаштування осей було налаштовано параметри відображення теплової карти. На вкладці «Options» і в розділі «Labels» увімкнуто опцію «Show labels» для відображення міток на тепловій карті. Це дозволило чіткіше бачити категорії даних, що аналізувалися.

Для отримання актуальних даних про відвідування вебсайту було налаштовано часовий інтервал. У верхньому правому куті інтерфейсу змінено тривалість на останню 1 годину і натиснуто «Refresh». Це дозволило тепловій карті відобразити найсвіжіші дані про активність користувачів.

Теплова карта відображає кількість переглядів вебсторінок, а також джерела переходів (з пошукової сторінки або зі сторінки рекомендацій). Чим вища інтенсивність кольору, тим більше переглядів зафіксовано для відповідної категорії. Теплова карта виглядала приблизно так (рис. 14), але вона

буде різною залежно від сторінок, які було відвідано під час генерації журналів доступу до вебсервера, та браузерів, які було використано.

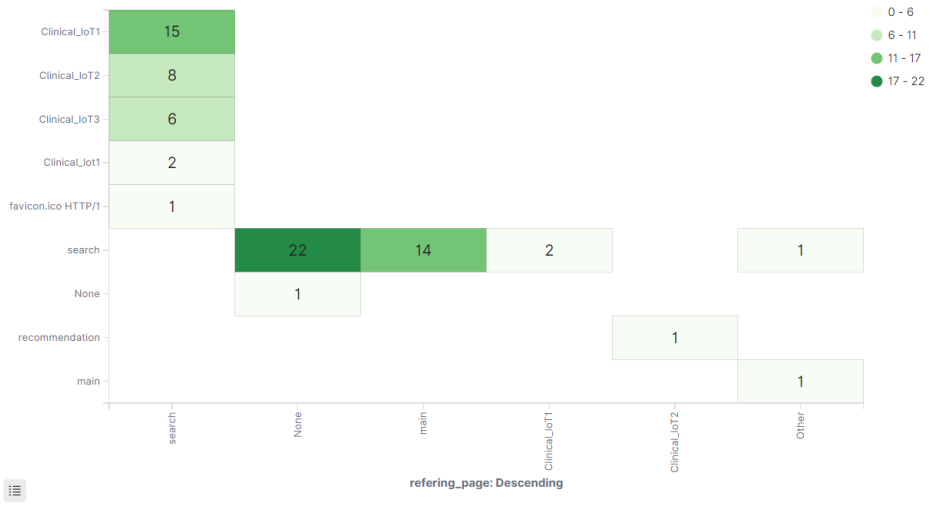

Рис. 14. Фільтр для браузера Firefox

Виходячи з цього зображення (рис. 14) візуалізації, відвідувачі частіше переходили на сторінки продуктів Clinical_IoT1 і Clinical_IoT2 з пошукової сторінки, ніж зі сторінки рекомендацій. Зроблено висновок, що пошукова сторінка є більш ефективною, ніж сторінка рекомендацій, для перенаправлення користувачів на сторінки продуктів.

Висновок

Створено конвеєр для підтвердження концепції, щоб збирати й аналізувати дані з пристроїв Інтернету речей, вбудованих в електровелосипеди. У складі конвеєру AWS IoT Core підписаний на тему, в якій публікуються дані з пристроїв IoT на електровелосипедах. Правило IoT направляє дані в потік доставки Kinesis Data Firehose. Рішення використовує Kinesis Data Firehose для завантаження потокових журналів доступу до вебсервера, а потім використовує функції Lambda для збагачення даних. Після використання Kinesis та Lambda для обробки даних використовувався OpenSearch Service для завантаження та індексації даних. Використовуючи OpenSearch Dashboards, створюються візуалізації даних, щоб отримати уявлення про стан користувачів електровелосипедів. Візуалізації поширюються серед зацікавлених сторін, використовуючи Amazon Cognito як інструмент аутентифікації та AWS Identity and Access Management (IAM) для авторизації доступу до інформаційної панелі.

Література

1. Evans D. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. *CISCO White Paper.* 2011.
2. Бондаренко Л. Д. Інтернет речей: Технології та застосування. *Системи управління, навігації та зв'язку*, 2020. 5(63), 120–123.
3. Perera C., Zaslavsky A., Christen P., Georgakopoulos D. Context Aware Computing for the Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials*, 2014. 16(1), 414–454.
4. Rajaraman V. Big Data Analytics. *Resonance*, 2016. 21(7), 695–716.
5. Zhang Y., Qian C., Zhang W. Internet of Things: Security and Privacy in a Connected World. Springer. 2017.
6. Zaslavsky A., Perera C., Georgakopoulos D. Sensing as a Service and Big Data. *IEEE International Conference on Advances in Cloud Computing (ACC-2013)*, Bangalore, India, 2013. 21–29.

References

1. Evans, D. (2011). The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. CISCO White Paper.
2. Bondarenko, L. D. (2020). Internet of things: Technologies and applications. *Control, navigation and communication systems*, 5(63), 120–123.
3. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D. (2014). Context Aware Computing for the Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials*, 16(1), 414–454.
4. Rajaraman, V. (2016). Big Data Analytics. *Resonance*, 21(7), 695–716.
5. Zhang, Y., Qian, C., Zhang, W. (2017). Internet of Things: Security and Privacy in a Connected World. Springer.
6. Zaslavsky, A., Perera, C., Georgakopoulos, D. (2013). Sensing as a Service and Big Data. *IEEE International Conference on Advances in Cloud Computing (ACC-2013)*, Bangalore, India, 21–29.