

Лекція 9. Обробка даних і маршрутизація повідомлень в Azure IoT: IoT Hub, Built-in Endpoints та Event Hubs

Тема 1. Azure IoT Hub - центр повідомлень у хмарному рішенні Інтернету речей (IoT).

Тема 2. Вбудовані кінцеві точки (Built-in endpoint), маршрутизація повідомлень.

Тема 3. Azure Event Hubs — центральний механізм прийому подій.

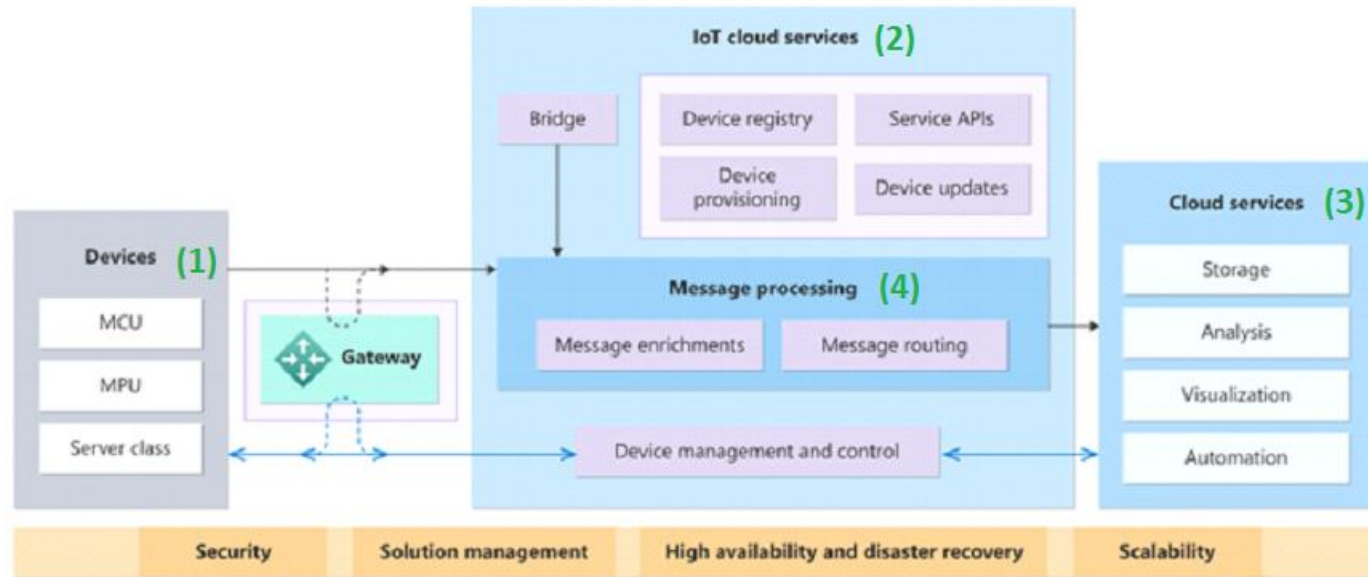
Azure IoT Hub є керованою службою, яка виконує роль центру повідомлень у хмарному рішенні Інтернету речей (IoT). Він забезпечує надійний і безпечний зв'язок у масштабі між IoT-застосунком та приєднаними до нього пристроями. Концептуально, Інтернет речей (IoT) об'єднує фізичні пристрої для обміну даними через інтернет, при цьому понад 10 мільярдів підключених пристроїв у всьому світі можуть приєднатися до цієї мережі.

IoT Hub розроблений для підтримки IoT-навантажень, масштабуючись до мільйонів одночасно підключених пристроїв і мільйонів подій на секунду.

IoT-пристрої мають такі характеристики:

1. Це часто вбудовані системи без оператора (наприклад, сенсор або виконавчий механізм).
2. Вони можуть бути розміщені у віддалених місцях, де фізичний доступ є дорогим, складним або неможливим (наприклад, сенсори на нафтовій вишці або вітровій фермі).
3. До них може бути доступ тільки через бекенд рішення, а не безпосередньо з інтернету (наприклад, пристрій за брандмауером).
4. Вони можуть мати обмежені ресурси живлення та обробки (наприклад, трежер активів, що працює від батареї).
5. З'єднання може бути переривчастим, повільним або дорогим (наприклад, підключення через стільникову мережу).
6. Вони можуть використовувати пропрієтарні, спеціалізовані або галузеві протоколи застосунків, як-от MQTT або AMQP.

Ключові компоненти (Інтернету речей Azure)



Пристрої (1), Хмарні сервіси IoT (2); Хмарні сервіси (3)

Додайте новий пристрій до створеного «IoT Hub»

demo | Devices IoT Hub

Search (Ctrl+/)

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Events
Pricing and scale
Device management
Devices

View, create, delete, and update devices in your IoT Hub.

Device name
enter device ID
Find devices

+ Add Device Refresh Delete

Device ID Status

There are no IoT devices to display.

Save Manage keys Message to Device Direct method + Add Module Identity Device twin Refresh

Device ID 1

Primary key

Secondary key

Primary connection string
Secondary connection string

Tags (edit) No tags

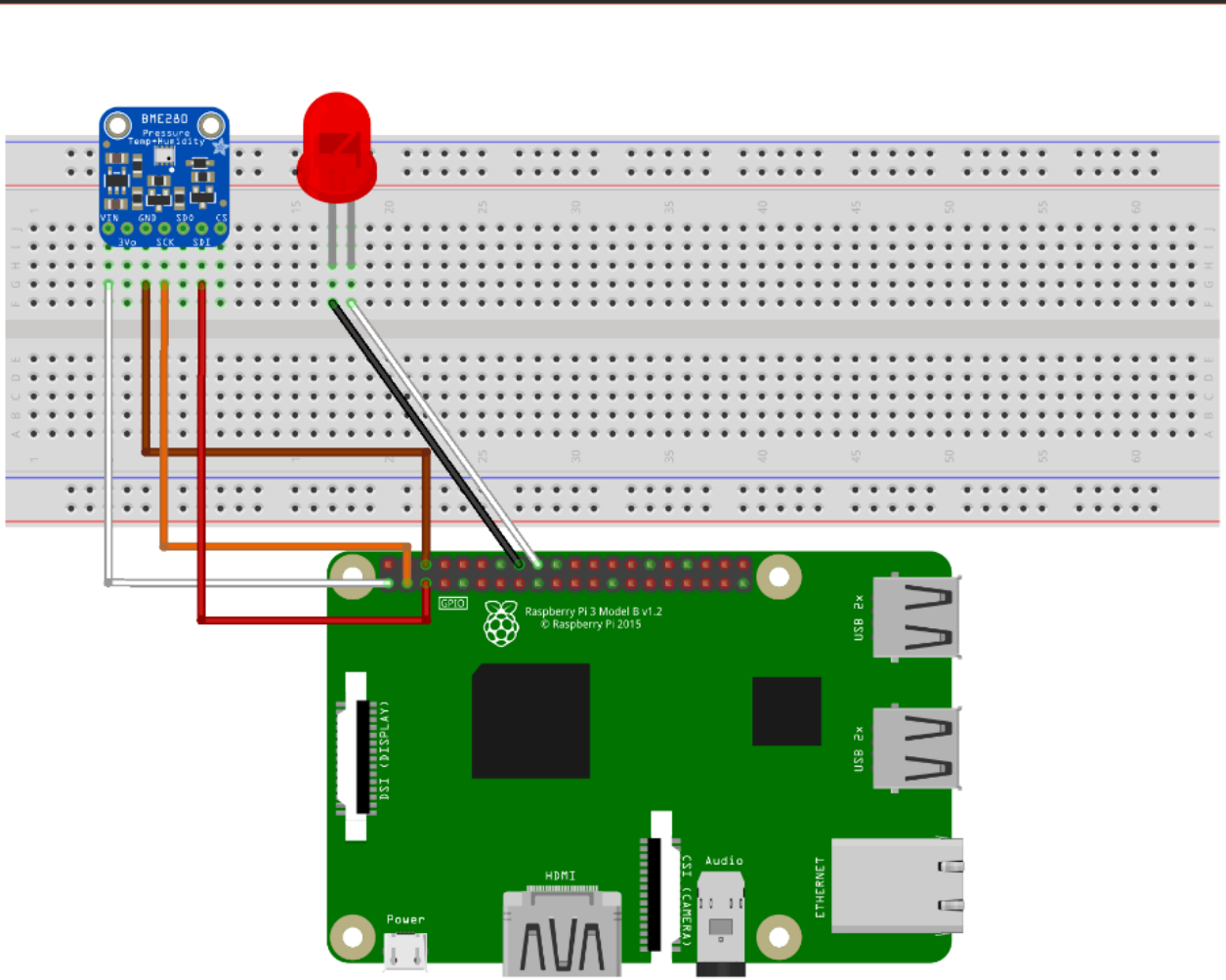
Enable connection to IoT Hub Enable Disable

Parent device No parent device

Module Identities Configurations

Module ID	Connection State	Connection State Last Updated ...	Last Activity Time (UTC)
-----------	------------------	-----------------------------------	--------------------------

Відкрийте нову вкладку у веб-браузері за посиланням <https://aka.ms/RasPi>. Ви перейдете на сайт симулятора Raspberry Pi. В області коду з правого боку знайдіть рядок із «const connectionString =». Замініть його рядком підключення, скопійованим із порталу Azure



fritzing

```
1 /*
2  * IoT Hub Raspberry Pi NodeJS - Microsoft Sample Code - Copyright (c) 2017 - Licensed MIT
3  */
4  const wpi = require('wiring-pi');
5  const Client = require('azure-iot-device').Client;
6  const Message = require('azure-iot-device').Message;
7  const Protocol = require('azure-iot-device-mqtt').Mqtt;
8  const BME280 = require('bme280-sensor');
9
10 const BME280_OPTION = {
11   i2cBusNo: 1, // defaults to 1
12   i2cAddress: BME280.BME280_DEFAULT_I2C_ADDRESS() // defaults to 0x77
13 };
14
15 const connectionString = 'HostName=soloveidemo.azure-devices.net;DeviceId=deviceid1;SharedAccessKey=dKwf2Wz1/5NF3sgLAdb1jEO1LDyk1f+8mt';
16 const LEDPin=4;
17
18 var sendingMessage = false;
19 var messageId = 0;
20 var client, sensor;
21 var blinkLEDTIMEOUT = null;
22
23 function getMessage(cb) {
24   messageId++;
25   sensor.readSensorData()
26     .then(function (data) {
27       cb(JSON.stringify({
28         messageId: messageId,
29         deviceId: 'Raspberry Pi Web Client',
30         temperature: data.temperature_C,
31         humidity: data.humidity
32       })), data.temperature_C > 30);
33     })
34     .catch(function (err) {
35       console.error('Failed to read out sensor data: ' + err);
36     });
37 }
38
39 function sendMessage() {
40   if (!sendingMessage) { return; }
41
42   getMessage(function (content, temperatureAlert) {
43     var message = new Message(content);
44     message.properties.add('temperatureAlert', temperatureAlert.toString());
45   });
46 }
```

Run Reset

Click 'Run' button to run the sample code(when sample is running, code is read-only).
Click 'Stop' button to stop the sample code running.
Click 'Reset' to reset the code.We keep your changes to the editor even you refresh the page.

>

Встановити зв'язок з пристроєм - перевірте, що дані телеметрії отримуються

The image shows two overlapping windows from the Azure IoT Explorer (preview) application. The background window displays the 'iothubowner' device page for the 'soloveidemo' hub. The foreground window is a modal dialog titled 'Add connection string'.

Background Window: iothubowner (soloveidemo)

- Buttons: Regenerate primary key, Regenerate secondary key, Swap keys.
- Primary key: [Redacted]
- Secondary key: [Redacted]
- Primary connection string: [Redacted]
- Secondary connection string: [Redacted]
- Permissions:
 - ☒ Registry Read
 - ☒ Registry Write
 - ☒ Service Connect
 - ☒ Device Connect

Foreground Window: Add connection string

Connection string *

```
HostName=soloveidemo.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=ahTNBejGt7LPZzzO1li/TV/T/gVZVr0jgAloTBEPLfU=
```

[Where do I get an IoT hub connection string?](#)
Please do not save your hub connection string to any unsafe locations

Host name: soloveidemo.azure-devices.net

Shared access policy name: iothubowner

Shared access policy key: [Redacted]

- Device identity
- Device twin
- Telemetry
- Direct method
- Cloud-to-device message
- Module identities
- IoT Plug and Play components

☒ Stop
 ☐ Show system properties

Telemetry You can monitor telemetry that the device sends to the IoT Hub.

Consumer group ⓘ \$Default

Specify enqueue time ⓘ

☐ No

Use built-in event hub

☐ Yes

ⓘ Receiving events...

Thu Oct 03 2024 23:33:18 GMT+0300 (Eastern European Summer Time):

```
{
  "body": {
    "messageId": 5,
    "deviceId": "Raspberry Pi Web Client",
    "temperature": 27.497046740370557,
    "humidity": 73.65029666076944
  },
  "enqueuedTime": "Thu Oct 03 2024 23:33:18 GMT+0300",
  "properties": {
    "temperatureAlert": "false"
  }
}
```

if t<3

Thu Oct 03 2024 23:33:16 GMT+0300 (Eastern European Summer Time):

```
{
  "body": {
    "messageId": 4,
    "deviceId": "Raspberry Pi Web Client",
    "temperature": 29.38017309828766,
    "humidity": 65.65092775338282
  },
  "enqueuedTime": "Thu Oct 03 2024 23:33:16 GMT+0300",
  "properties": {
    "temperatureAlert": "false"
  }
}
```

telemetryrule ⋮

Route details

Name

telemetryrule

Endpoint * ⓘ

soloveiendpoint

+ Add

Data source * ⓘ

Device Telemetry Messages

Enable route * ⓘ

Enable

Disable

Create a query to filter messages before data is routed to an endpoint. [Learn more](#)

Routing query ⓘ

1 temperatureAlert = "true"

Test

Пристрої можуть обмінюватися даними з IoT Hub в Azure за допомогою різних протоколів. вихідні порти, які мають бути відкриті, щоб пристрій міг використовувати певний протокол:

Protocol	Port
MQTT	8883
MQTT over WebSockets	443
AMQP	5671
AMQP over WebSockets	443
HTTPS	443

Create a device ...

 Find Certified for Azure IoT devices in the Device Catalog

Device ID * 

The ID of the new device

☐ IoT Edge Device

Authentication type 

Symmetric key

X.509 Self-Signed

X.509 CA Signed

Auto-generate keys 



Connect this device to an IoT hub 

Enable

Disable

Parent device 

No parent device

[Set a parent device](#)

можливо використовувати методи автентифікації: на основі маркерів SAS або сертифіката X.509.

Метод на основі маркерів SAS забезпечує перевірку справжності кожного виклику до Центру Інтернету речей, надісланого пристроєм за рахунок прив'язування симетричного ключа до кожного виклику.

Застосування сертифікатів X.509 дозволяє перевіряти справжність пристрою IoT фізично при встановленні підключення за протоколом TLS. Вибір між цими двома методами в основному визначається вимогами щодо безпеки автентифікації пристроїв та наявністю захищеного сховища на пристрої (для безпечного зберігання закритого ключа).

Аутентифікація на основі сертифіката ЦС X.509 – це спосіб аутентифікації з використанням методу, який значно спрощує створення посвідчення пристрою та керування життєвим циклом у ланцюжку поставок.

Відмінною ознакою автентифікації на основі сертифіката X.509 є відношення "один до багатьох", яке має сертифікат ЦС з підпорядкованими пристроями. Це ставлення дозволяє реєструвати будь-яку кількість пристроїв у Центрі Інтернету речей, зареєструвавши один раз сертифікат ЦС X.509. В іншому випадку перед підключенням пристроїв необхідно зареєструвати унікальні сертифікати для кожного з них. Відношення "один до багатьох" також полегшує операції управління життєвим циклом сертифікатів пристроїв.

Іншим важливим атрибутом автентифікації ЦС X.509 є спрощена логістика ланцюжка постачання. Для безпечної автентифікації пристроїв потрібно, щоб кожен пристрій мав унікальний ключ.

Політики загального доступу на рівні Центру Інтернету речей

Політика service - Дозвіл ServiceConnect використовується внутрішніми хмарними службами і надає наступний доступ:

- Доступ до **кінцевих точок**, підключених до хмарної служби, та моніторингу.
- Отримання повідомлень із пристрою до хмари, надсилання повідомлень із хмари на пристрій та отримання відповідних підтверджень доставки.
- Отримання **підтвердження доставки для надсилання файлів**.
- Доступ до двійників для оновлення тегів і необхідних властивостей, отримання властивостей, що повідомляються, і виконання запитів.

Політика Device - Дозвіл DeviceConnect використовується пристроями і надає наступний доступ:

- Доступ до кінцевих точок, підключених до пристрою.
- Надсилання повідомлень із пристрою до хмари та отримання повідомлень із хмари до пристрою.
- Виконує надсилання файлів.
- Отримання повідомлень про потрібні властивості двійника пристрою та оновлення властивостей двійника пристрою.

Політика RegistryRead - дозвіл RegistryRead використовується внутрішніми хмарними службами і надає доступ на читання до реєстру посвідчень, у якому містяться відомості про пристрої та модулі, що мають право на підключення до цього Центру Інтернету речей. Пристрій або модуль також повинні пройти автентифікацію в Центрі Інтернету речей, тобто надати облікові дані, збережені в реєстрі посвідчень.

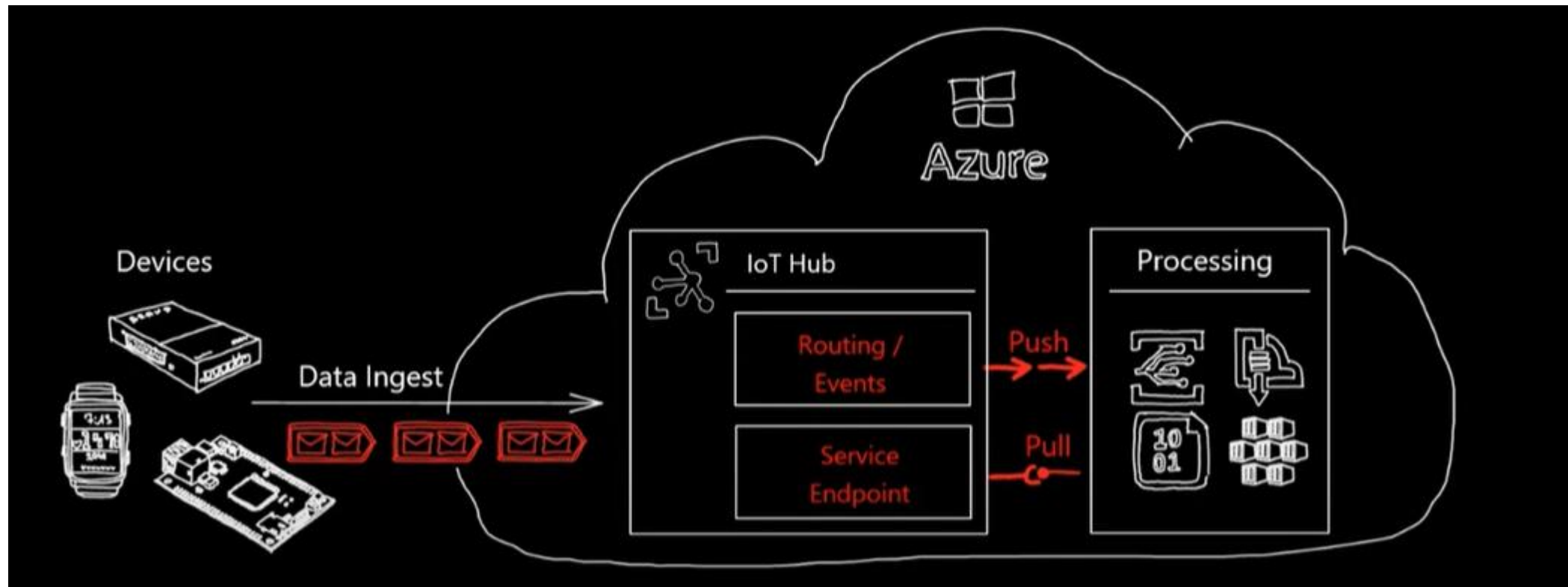
Політика RegistryRead - дозвіл RegistryReadWrite використовується внутрішніми хмарними службами і надає наступний доступ: на читання та запис до реєстру посвідчень.

IoT Hub Маршрутизація повідомлень (*routing events, Service Endpoint*)

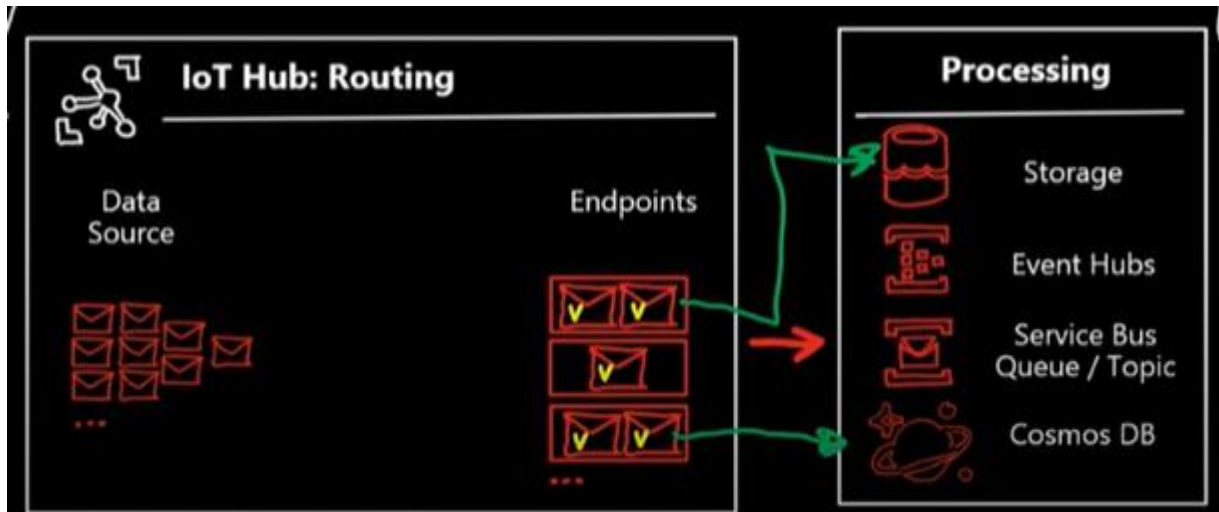
Маршрутизація повідомлень дозволяє надсилати повідомлення з пристроїв до хмарних служб. Маршрутизація повідомлень може використовуватись для наступних способів:

Надсилання повідомлень телеметрії та подій пристрою у вбудовану кінцеву точку та кінцеві точки користувача. Події, які можуть бути перенаправлені, включають події життєвого циклу пристрою.

Фільтрування даних перед маршрутизацією, використовуючи розширені запити.



Routing endpoints (Кінцеві точки маршрутизації)



Кожен центр Інтернету речей має кінцеву точку маршрутизації за умовчанням, яка називається повідомленнями та подіями, сумісними з Центрами подій.

Центр Інтернету речей зараз підтримує наступні кінцеві точки для маршрутизації повідомлень:

1. Контейнери сховища (Storage)
2. Центр Подій (Event Hub)
3. Черги/Розділи службової шини
4. Cosmos DB

Save Undo

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Events
- Device management
 - Devices
 - IoT Edge
 - Configurations + Deployments
 - Updates
 - Queries
- Hub settings
 - Built-in endpoints**
 - Message routing
 - File upload
 - Failover
 - Pricing and scale
 - Properties
 - Locks
 - Security settings
 - Defender for IoT
 - Overview
 - Security Alerts
 - Recommendations
 - Settings
 - Monitoring

Each IoT hub comes with built-in system endpoints to handle system and device messages. When you create new endpoints and routes, messages :

Event Hub Details

Partitions ⓘ

2

Event Hub-compatible name ⓘ

iothub-ehub-soloveidem-62443005-cc4fa5021f

Retain for ⓘ

1 Days

Consumer Groups ⓘ

Consumer Groups
\$Default

Create new consumer group

Event Hub compatible endpoint

Only policies that allow service connect permissions can be selected. These permissions are applied to the built-in endpoint.

Shared access policy ⓘ

iothubowner

Event Hub-compatible endpoint ⓘ

Endpoint=sb://ihsuprodamres107dednamespace.servicebus.windows.net/;SharedAccessKeyName=iothubowner;SharedAccessKey=...

Cloud to device messaging

Control message retention time and retry attempts.

Default TTL ⓘ

1 Hours

Feedback retention time ⓘ

1 Hours

Maximum delivery count ⓘ

Azure IoT Hub Built-in endpoints є вбудованими кінцевими точками для зв'язку між вашим IoT рішенням та зовнішніми системами. Основні налаштування Azure IoT Hub Built-in Endpoints

Events (Event Hub-compatible endpoint): Ця кінцева точка дозволяє надсилати телеметричні дані з пристроїв до IoT Hub. Після цього дані можуть бути оброблені іншими службами (наприклад, Azure Stream Analytics або Azure Functions).

Cloud-to-device (C2D) messages: Використовується для відправки повідомлень від хмарної інфраструктури до пристроїв.

Device-to-cloud (D2C) messages: Відповідає за телеметрію від пристроїв до хмари.

Direct methods: Дозволяє викликати прямі методи для керування пристроями.

Twin updates: Кінцева точка для управління "двійниками" пристроїв (виртуальні представлення).

Підключення до Event Hub (Event Hub-compatible): Для збору телеметрії з пристроїв ви можете використовувати кінцеву точку IoT Hub, яка сумісна з Event Hub.

Службова шина - розділи

Розділи можуть містити кілька незалежних підписок, які приєднуються до розділу, а в іншому для одержувача повністю ідентичні звичайним чергам. Підписка на розділ може отримувати копію будь-якого повідомлення, надісланого до цього розділу.



Простір імен — це контейнер для всіх компонентів обміну повідомленнями (черги та розділи). Простір імен може містити одну або кілька черг і розділів, і вона часто є контейнером програми. Простір імен можна порівняти з тим, що в термінології інших брокерів називається "сервером", але ці поняття не повністю еквівалентні. Простір імен Службової шини надає вам частку ємності великого кластера, що складається з десятків активних віртуальних машин. За потреби він охоплює три зони доступності Azure. Завдяки цьому ви отримуєте всі переваги доступності та надійності, які надаються використанням брокера повідомлень у величезному масштабі.

Add a route ...

soloveidemo

- 1 Endpoint
- 2 Route
- 3 Enrichment

endpoint name *

soloveiendpoint

Select existing

Azure storage container *

https://soloveistoredemo.blob.core.windows.net/demo

Pick a container

Batch frequency * ⓘ

100

Chunk size window * ⓘ

100

Encoding * ⓘ

- ☐ AVRO
- ☒ JSON

File name format * ⓘ

{iothub}/{partition}/{YYYY}/{MM}/{DD}/{HH}/{mm}

If multiple files are created within the same minute, the filename format would be {iothub}/0/2024/10/10/20/55-01.json

Authentication type *

Choose the authentication type for this routing endpoint. [Learn more](#)

- ☒ Key-based
- ☐ System-assigned
- ☐ User-assigned

ⓘ System-assigned identity is switched off and cannot be used as an authentication type.

Центри подій Azure дозволяють автоматично записувати певний обсяг потокових даних із Центрів подій в обліковий запис сховища BLOB-об'єктів Azure або Azure Data Lake Storage Gen 1 або Gen 2 із зазначеним інтервалом часу та розміру.

Назва файла формається за наступними правилами
{iothub}/{partition}/{YYYY}/{MM}/{DD}/{HH}/{mm}

Azure Event Hubs — центральний механізм прийому подій.

Event Hubs є багатопроTOCOLьним механізмом потокової передачі подій, який нативно підтримує протоколи Advanced Message Queuing Protocol (AMQP), Apache Kafka та HTTPS.

Event Hubs побудований як хмарний брокер (cloud native broker engine), що дозволяє запускати навантаження Kafka з кращою продуктивністю, економічною ефективністю та без необхідності налаштовувати чи керувати власними кластерами Kafka.

Схема Реєстру (Schema Registry)

Azure Schema Registry, інтегрований в Event Hubs, надає централізоване сховище для керування схемами застосунків потокової передачі подій.

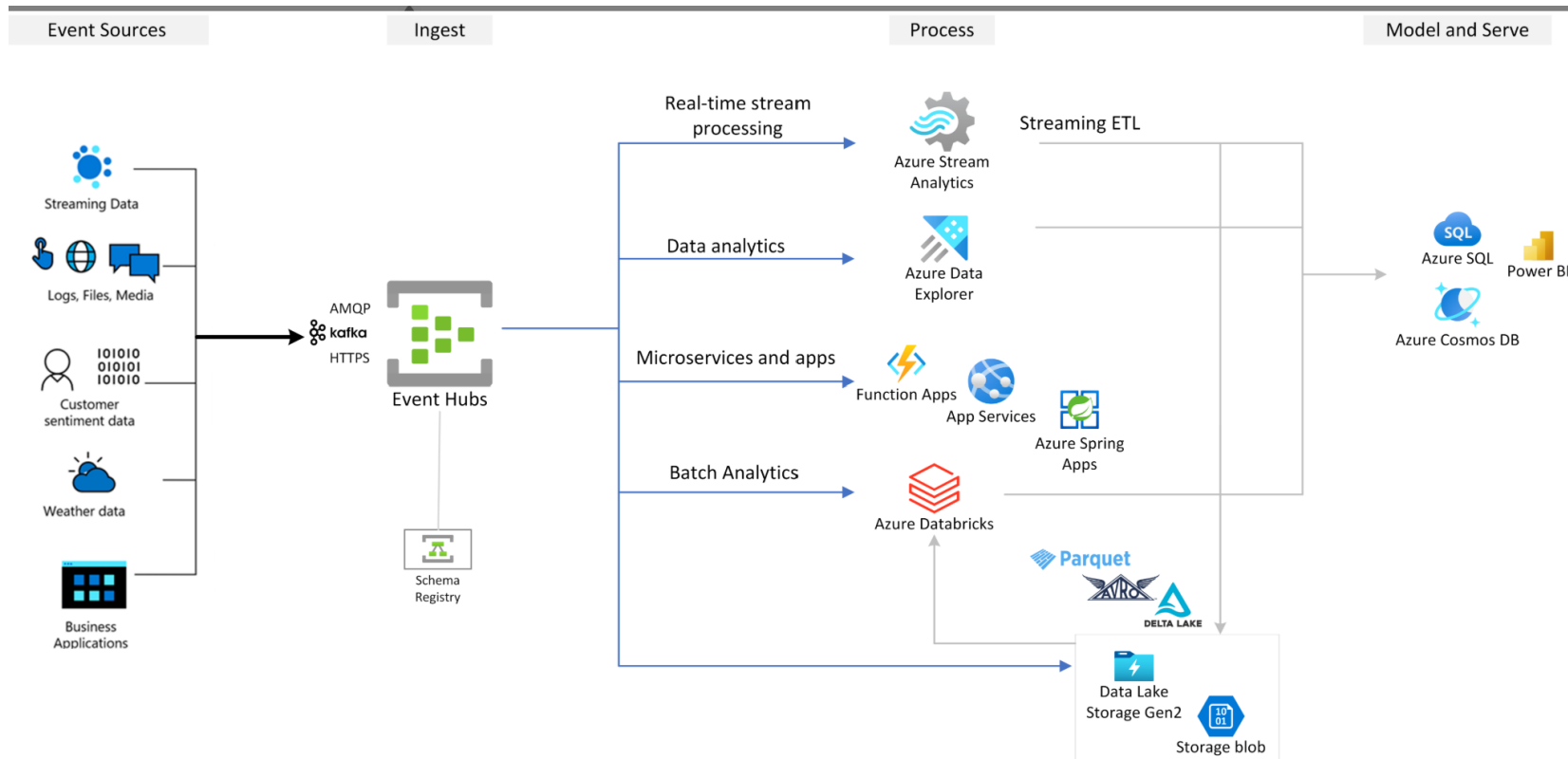
Реєстр Схеми забезпечує сумісність та узгодженість даних між виробниками та споживачами. Він сприяє валідації, управлінню та еволюції схем.

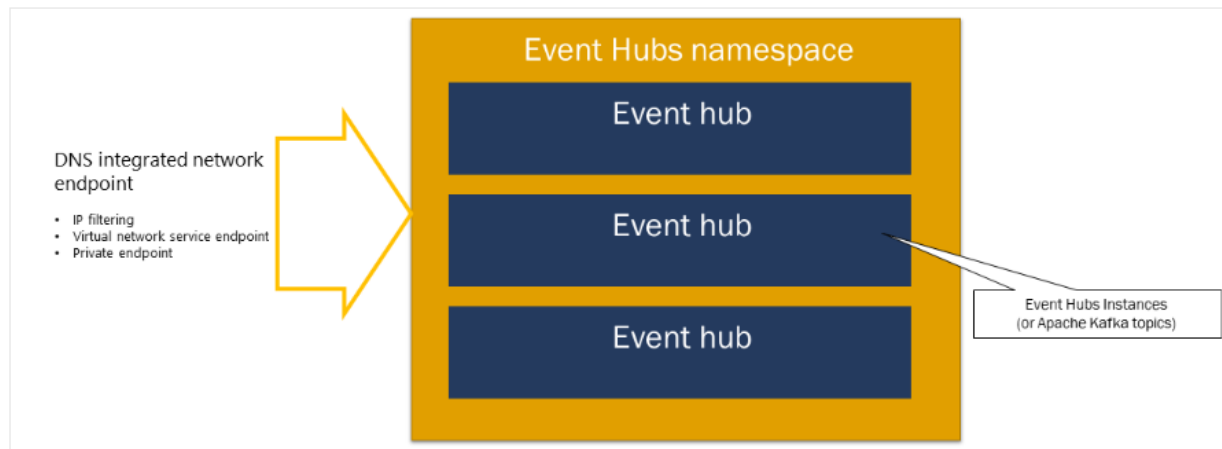
Event Hubs може починати з потоків даних у мегабайтах і зростати до гігабайтів або терабайтів. Функція автоматичного роздування (auto-inflate) є одним із доступних варіантів масштабування одиниць пропускної спроможності відповідно до потреб використання.

Event Hubs може обробляти події до 20 МБ у виділених кластерах, що забезпечує безперебійну роботу, навіть якщо повідомлення не можна розділити на менші сегменти.

Основні функціональні компоненти Event Hubs:

- Producer applications (Застосунки-Виробники): Надсилають дані до event hub, використовуючи SDK Event Hubs або клієнт Kafka producer.
- Namespace (Простір імен): Це контейнер керування для одного або кількох event hubs або Kafka topics. Завдання керування, як-от виділення потужності та налаштування безпеки, обробляються на цьому рівні.
- Event Hubs/Kafka topic: Це розподілений лог лише для додавання (append-only distributed log), який організовує події і складається з одного або кількох розділів (Partitions).
- Partitions (Розділи): Вони використовуються для масштабування event hub, подібно до смуг руху на шосе. Додавання розділів збільшує пропускну здатність.
- Consumer applications (Застосунки-Споживачі): Читають дані, підтримуючи зміщення споживача (consumer offset) у лозі подій.
- Consumer group (Група споживачів): Це логічна група, яка дозволяє багатьом екземплярам споживачів читати одні й ті ж потокові дані з event hub незалежно, у власному темпі та зі своїми зміщеннями.





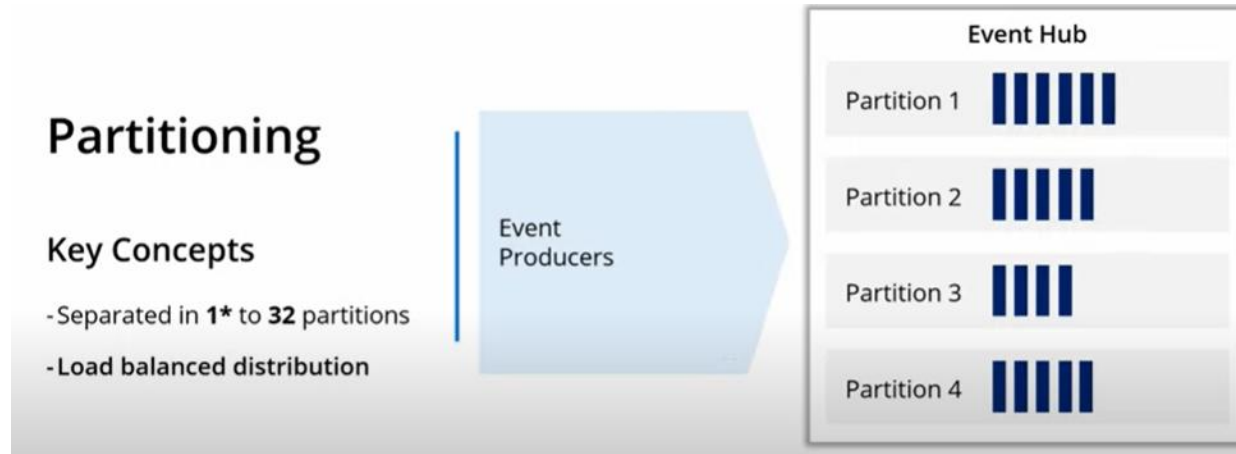
Видавцем подій для Event Hub – є будь-який доданок, яка надсилає дані до Центру подій. Видавці подій можуть публікувати події за допомогою HTTPS, AMQP 1.0 або Kafka. Видавці подій використовують авторизацію на основі ідентифікатора Microsoft Entra з маркерами JWT, виданими OAuth2, або маркером підписаної URL-адреси концентратора подій (SAS) для отримання доступу до публікації.

Можна публікувати події окремо або пакетно. Для однієї публікації встановлено обмеження в 1 МБ, незалежно від того, одиночна ця подія чи пакет. Події публікації, що перевищують це граничне значення, відхиляється.

Пропускна спроможність Центрів подій масштабується за допомогою секцій та розподілу одиниць пропускної спроможності. Видавцям рекомендується не знати, яку конкретно модель секціонування вибрано для Центру подій. Слід вказати тільки ключ секції, який використовується для узгодженого призначення пов'язаних подій однієї секції.

Додатково, події розподіляють по розділах (partition)

Секція (partition) – журнал фіксації даних, які отримав центр подій (Event Hub). Секція має структуру: тіло події, контейнер властивостей для опису події; мета дані; зміщення в секції (offset); номер події; мітка часу, коли подія відбулась.



Кількість секцій можна вказати під час створення концентратора подій. Воно має бути від 1 до максимальної кількості секцій, дозволених для кожної категорії Event Hub.

Рекомендуємо вибрати щонайменше стільки секцій, скільки може знадобитися під час пікового навантаження вашої програми для цього концентратора подій. Для рівнів, відмінних від рівня "Преміум" та виділених рівнів, не можна змінити кількість секцій для концентратора подій після його створення. Для концентратора подій рівня "Преміум" або "Виділений" можна збільшити кількість секцій після його створення, але їх не можна зменшити. Розподіл потоків між секціями зміниться при зіставленні ключів секцій із змінами, тому слід намагатися уникнути таких змін, якщо відносний порядок подій має значення у додатку.

Оскільки секція – це механізм організації даних, що дозволяє паралельно публікувати та використовувати дані, рекомендуємо збалансувати одиниці масштабування (одиниці пропускної спроможності, одиниці обробки чи одиниці ємності) та секції для досягнення оптимального масштабу. Зазвичай рекомендується використовувати максимальну пропускну здатність 1 МБ/с на секцію. Таким чином, правило великого пальця для обчислення кількості секцій розділятиме максимальну очікувану пропускну здатність на 1 МБ/с. Наприклад, якщо для вашого варіанту використання потрібно 20 МБ/с, для досягнення оптимальної пропускної спроможності рекомендуємо вибрати щонайменше 20 секцій.

Опубліковані події видаляються з Центру подій відповідно до політики зберігання на основі часу, що настроюється.

найкоротший термін зберігання становить 1 год. В даний час термін зберігання можна задати лише у порталі Azure. Шаблон Resource Manager, PowerShell і CLI дозволяють задати цю властивість лише днями.

Для Центрів подій (категорія Стандартний) максимальний період зберігання становить 7 днів.

Для Центрів подій (категорій Преміум та Виділених) максимальний період зберігання становить 90 днів.

Функція "Збір" у Центрах подій дозволяє автоматично збирати дані потокової передачі в Центри подій та зберігати їх у вибраному обліковому записі зберігання BLOB-об'єктів або обліковому записі служби зберігання Azure Data Lake. Цю функцію можна ввімкнути на порталі Azure і вказати мінімальний розмір даних, що записуються, і інтервал часу для запису. За допомогою функції "Збір" у Центрах подій можна вказати власний обліковий запис сховища BLOB-об'єктів Azure та контейнер або обліковий запис служби зберігання Azure Data Lake, який використовується для зберігання зібраних даних. Зафіксовані дані записуються у форматі Apache Avro.

Create Event Hub

Event Hubs

Setting up Capture is fast, there are no administrative costs to run it, and it scales automatically with Event Hubs throughput units. Event Hubs Capture is the easiest way to load streaming data into Azure, and enables you to focus on data processing rather than on data capture. [Learn more about capture.](#)

Capture

On Off

Note: Enabling Capture will result in additional charges to this account. [Learn more about our pricing.](#)

Time window (minutes)

5

Size window (MB)

300

☐ Do not emit empty files when no events occur during the Capture time window

Capture Provider

Azure Storage Account

Azure Storage Container *

demo [Select Container](#)

Storage Account

/subscriptions/601a84f6-579c-4ba6-ac84-00c597a85bd9/resourceGrou...

Sample Capture file name formats

{Namespace}/{EventHub}/{PartitionId}/{Year}/{Month}/{Day}/{Hour}/{...}

Capture file name format

{Namespace}/{EventHub}/{PartitionId}/{Year}/{Month}/{Day}/{Hour}/{Min...}
e.g. soloveieventhub/eventhub/0/2024/10/10/18/07/27

Authentication for Capture

Capture event streams to Azure Blob Storage or Azure Data Lake Storage Gen2 using default (SAS) authentication or managed identity in Microsoft Entra ID. [Learn more](#)

Authentication type

☒ Default (SAS)
☐ System Assigned Identity
☐ User Assigned Identity

Home > soloveieventhub | Event Hubs > eventhub (soloveieventhub/eventhub)

eventhub (soloveieventhub/eventhub) | Capture

Search Save changes Discard Give feedback

- Overview
- Access control (IAM)
- Diagnose and solve problems
- Data Explorer (preview)
- Settings
 - Shared access policies
 - Configuration
 - Properties
 - Locks
- Entities
 - Consumer groups
- Features
 - Capture
 - Process data
 - Analyze data (preview)
- Automation
 - CLI / PS
 - Tasks (preview)
 - Export template
- Help

Azure Event Hubs Capture enables you to automatically deliver the streaming data in Event Hubs to an Azure Blob storage or Azure Data Lake Store account of your choice, with the added flexibility of specifying a time or size interval. Setting up Capture is fast, there are no administrative costs to run it, and it scales automatically with Event Hubs throughput units. Event Hubs Capture is the easiest way to load streaming data into Azure, and enables you to focus on data processing rather than on data capture. [Learn more about capture.](#)

Output event serialization format

Avro

Capture

On Off

Note: Enabling Capture will result in additional charges to this account. [Learn more about our pricing.](#)

Time window (minutes)

5

Size window (MB)

300

☐ Do not emit empty files when no events occur during the Capture time window

Capture Provider

Azure Storage Account

Azure Storage Container *

demo [Select Container](#)

Storage Account

/subscriptions/601a84f6-579c-4ba6-ac84-00c597a85bd9/resourceGroups/demo/providers/Microsoft.Storage/storageAccounts/soloveistoredemo

Sample Capture file name formats

{Namespace}/{EventHub}/{PartitionId}/{Year}/{Month}/{Day}/{Hour}/{Minute}/{Second}

Capture file name format

{Namespace}/{EventHub}/{PartitionId}/{Year}/{Month}/{Day}/{Hour}/{Minute}/{Second}

e.g. soloveieventhub/eventhub/0/2024/10/10/18/09/45

Review + create

< Previous

Next: Review + create >

Конструктори

Для різних платформ (Python, .NET, Java, JavaScript) API трохи відрізняється, але загальна ідея одна:

Наприклад, для .NET: `new EventHubProducerClient(string fullyQualifiedNamespace, string eventHubName, TokenCredential credential, EventHubProducerClientOptions options)` або інші варіанти.

Для JavaScript: `new EventHubProducerClient(connectionString, eventHubName, options)` або із `namespace+credential`.

Для Python: `EventHubProducerClient(fully_qualified_namespace: str, eventhub_name: str, credential: CredentialTypes, *, buffered_mode: bool = False, **kwargs)`

Основні методи і функції

4.1 Закриття клієнта

`CloseAsync()` (або `close()` у синхронному Java API) – закриває з'єднання, звільняє ресурси. `DisposeAsync()` / `close()` / `closeAsync()` – варіанти залежно від платформи. Часто після завершення роботи потрібно закривати клієнт.

4.2 Створення батча

`CreateBatchAsync()` або `CreateBatch(CreateBatchOptions)` – створює новий батч `EventDataBatch`, куди можна додавати події методом `TryAdd()`. Параметри `CreateBatchOptions` можуть включати `PartitionId`, `PartitionKey`, `MaximumSizeInBytes`, інші.

4.3 Додавання подій до батча

Об'єкт `EventDataBatch` має метод `TryAdd(EventData eventData)` – повертає `true`, якщо подія помістилась у батч, або `false`, якщо батч вже досягнув обмеження по розміру. Якщо `TryAdd()` повертає `false`, зазвичай відправляють поточний батч і створюють новий. Це дозволяє ефективно накопичувати події.

4.4 Відправка батча або подій

`SendAsync(EventDataBatch batch, CancellationToken)` – відправляє раніше зібраний батч.

`SendAsync(IEnumerable<EventData> events, SendEventOptions options, CancellationToken)` – можна відправити колекцію подій без ручного створення батчу; сервіс сам перевірить розмір. Але якщо обсяг занадто великий – виняток.

Для JavaScript платформи: `sendBatch()` або `sendBatch(messagesArray)` або `sendBatch(EventDataBatch)` залежно від API.

4.5 Отримання властивостей і партицій

`GetEventHubPropertiesAsync()` – повертає metadata про Event Hub: число партицій, їх ідентифікатори.

`GetPartitionIdsAsync()` – повертає список ідентифікаторів партицій.

`GetPartitionPropertiesAsync(String partitionId)` – отримує інформацію по конкретній партиції (наприклад, початковий і кінцевий offset, sequence number).

4.6 Властивості клієнта

`FullyQualifiedNamespace` – namespace, до якого підключено.

`EventHubName` – ім'я Event Hub.

`Identifier` – опційний ідентифікатор клієнта.

`IsClosed` – вказує чи клієнт вже закритий.

Питання

1. Що таке Azure IoT Hub та яку роль він виконує в IoT-архітектурі?
2. Які типи комунікацій підтримує IoT Hub?
3. Чим IoT Hub відрізняється від Event Hubs з точки зору IoT-сценаріїв?
4. Які основні протоколи підтримує IoT Hub (MQTT, AMQP, HTTPS) і коли їх використовують?
5. Як IoT Hub забезпечує автентифікацію та безпеку пристроїв?
6. Що таке вбудована кінцева точка (Built-in Endpoint) в Azure IoT Hub?
7. Для чого використовують маршрутизацію повідомлень у IoT Hub?
8. Які властивості повідомлення використовуються для умовної маршрутизації (message routing query)?
9. До яких сервісів Azure можна спрямувати повідомлення через маршрути (Event Hubs, Service Bus, Storage)?
10. У чому різниця між default endpoint та custom endpoints?
11. Як перевірити, що маршрутизація повідомлень працює коректно (message routing test tool)?
12. Яку роль відіграє Azure Event Hubs у прийомі та обробці потоків подій?
13. Що таке partitions та consumer groups в Event Hubs, і чому вони важливі для масштабованої обробки даних?