

[ADVANTECH]

Windows 10의 UWP앱을 이용하여 센서데이터를 수집하고
Azure Cloud로 전송 및 Dashboard를 통해 시각화를 구현

Windows 10 +
Azure IoT

● 사전 준비.....	4
1. Reference	4
1.1. Advantech ARK-1123 Gateway	4
1.2. Advantech WISE-4012 Data Acquisition and protocol converter	4
1.3. Microsoft Azure Introduce	4
1.4. Microsoft Azure Portal	4
2. Requirements	4
2.1. Azure account with subscription	4
2.2. Microsoft Visual Studio 2017	4
3. Goals	4
● 실습1 (Gateway + Windows 10).....	6
1. Device 구성	6
1.1. 전체 디바이스 연결 구성 – 센서 데이터 수집을 위한 환경 구성	6
1.2. WISE-4012 연결 구성	7
1.3. 온도/습도 센서 연결 구성	7
1.4. 임베디드 PC 연결 구성	8
1.5. 임베디드 PC에 접속	9
1.6. PC 로그인 정보	9
1.7. WISE-4012 연결 확인	9
1.8. AdamApax.NET Utility를 실행	10
1.9. WISE-4012 장치 인식 확인	10
1.10. 웹브라우저 로그인	11
1.11. WISE-4012의 동작 상태 확인	11
● 실습2 (Microsoft Azure Cloud).....	12
1. Azure Resource 생성	12

1.1.	IoT Hub 생성	12
1.2.	Stream Analytics Job 생성	21
1.3.	COSMOS DB 생성	23
2.	Resource 항목 설정	25
2.1.	Cosmos DB 설정	25
2.2.	Stream Analytics Job 설정	26
3.	각 리소스의 동작 유무 확인	30
3.1.	Stream Analytics 작업 시작	30
3.2.	Cosmos DB 데이터 확인	30
4.	Power BI – Cosmos DB	31
4.1.	Power BI – Cosmos DB 연결	31
4.2.	데이터베이스 쿼리 편집	34
4.3.	Dashboard 위젯 구성	36
●	Appendix : Azure 연결 확인	37
1.	Device Explorer 다운로드(GitHub)	37
2.	Device Explorer 설정	37
3.	시뮬레이션 데이터 통신 테스트	39
●	Appendix : UWP 앱 배포	41
1.	Visual Studio 설치	41
1.1.	Windows 10 개발자 모드로 변경	41
1.2.	사전 설치 - NuGet package	41
1.3.	솔루션 구성	42
2.	UWP 배포	43
2.1.	Debug / x86 / 로컬 컴퓨터 확인 후 실행	43
2.2.	프로그램이 실행되면 Start Reading 옆의 단추를 클릭합니다.	43

- **Appendix** 44
 - 1. **센서의 제어** 44
 - 1.1. Azure Service bus 44
 - 1.2. Azure Function 44
 - 2. **Azure Learn** 44

● 사전 준비

1. Reference

1.1. Advantech ARK-1123 Gateway

[매뉴얼 및 드라이버 다운로드 링크](#)

1.2. Advantech WISE-4012 Data Acquisition and protocol converter

[매뉴얼, 드라이버 및 SDK](#)

1.3. Microsoft Azure Introduce

<https://azure.microsoft.com>

1.4. Microsoft Azure Portal

<https://portal.azure.com>

2. Requirements

2.1. Azure account with subscription

Free trial: <https://azure.microsoft.com/en-us/free/>

2.2. Microsoft Visual Studio 2017

<https://docs.microsoft.com/ko-kr/visualstudio/install/install-visual-studio?view=vs-2017>

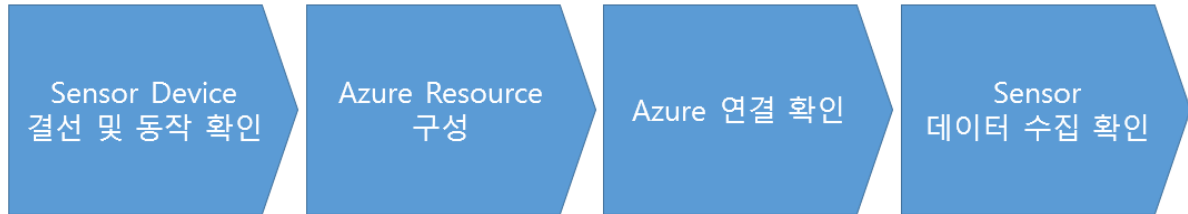
3. Goals

이번 실습을 통해 센서 데이터 수집을 위한 하드웨어의 연결을 구성할 수 있습니다. 하드웨어 구성이 완료되면 Windows 10의 UWP를 이용하여 Gateway에서 실행될 소프트웨어를 배포합니다. Gateway에서 실행되는 앱은 센서의 데이터를 Public Cloud로 전송하게 되며 IoT Hub를 통해 수집된 데이터는 DB저장 및 Function의 기능을 수행하게 됩니다. Power BI Dashboard를 통해 수집된 데이터를 시각적으로 확인할 수 있습니다.

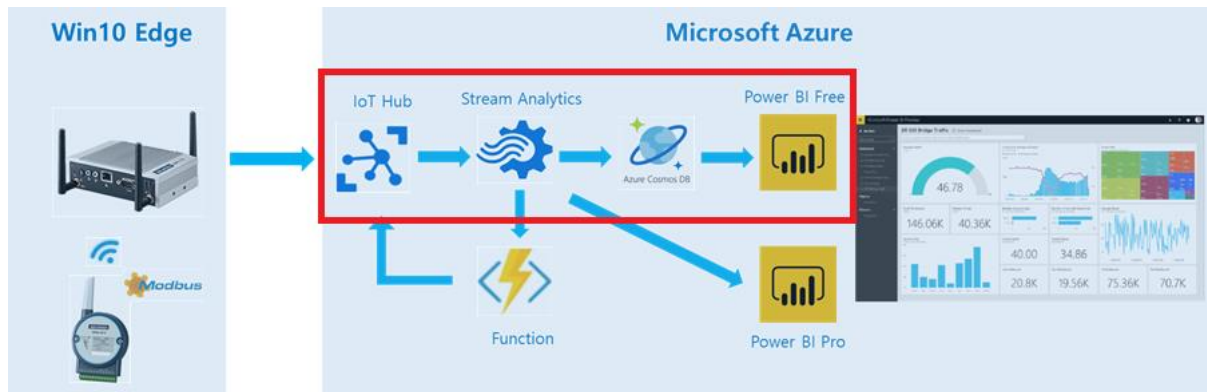
3.1. 데이터 수집을 위한 Sensor 연결 구성

3.2. UWP를 이용하여 Cloud로 데이터를 수집 및 Dashboard를 통한 모니터링 및 제어

4. 실습 진행 순서



5. 실습 구성안

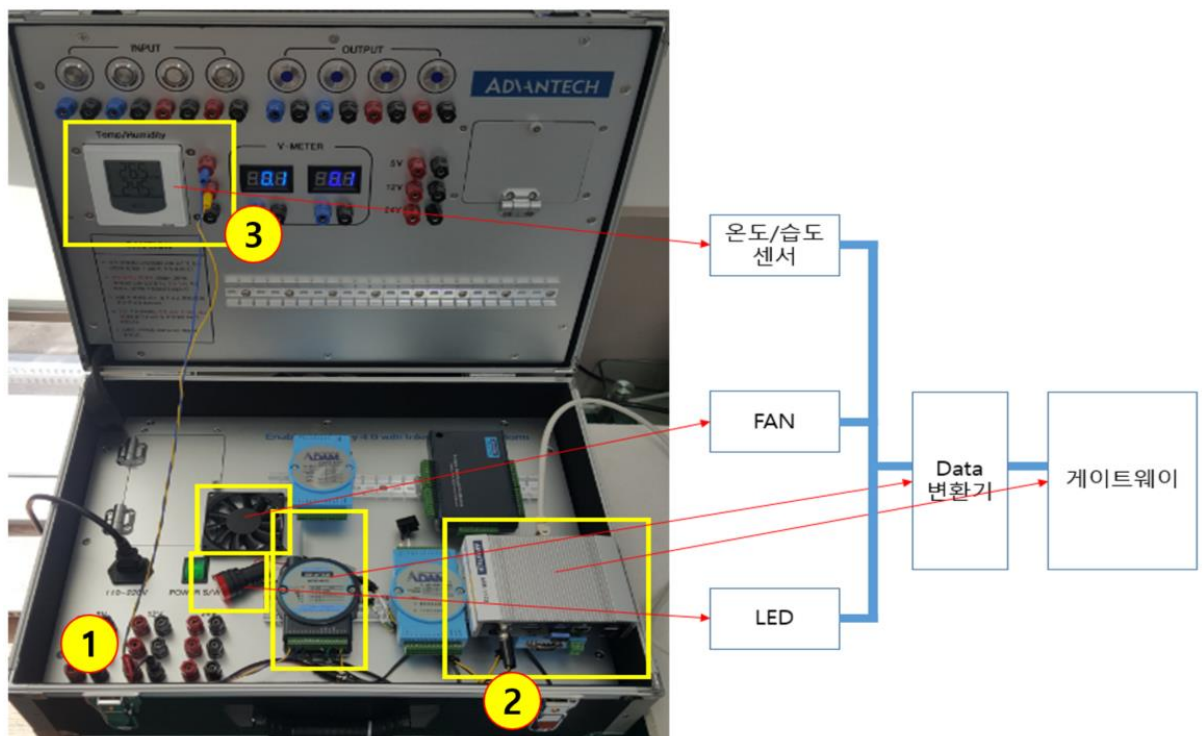


● 실습1 (Gateway + Windows 10)

1. Device 구성

1.1. 전체 디바이스 연결 구성 – 센서 데이터 수집을 위한 환경 구성

- 1) 동봉된 "Cable"에서 빨간색-12V/검정색-GND를 데모박스에 연결합니다.①
- 2) 동봉된 "Cable의 DC 커넥터"를 ARK-1123의 DC 전원에 연결합니다. ②
- 3) 동봉된 "Cable"에서 터미널 블록과 연결된 부분을 온/습도 센서에 연결 합니다. ③



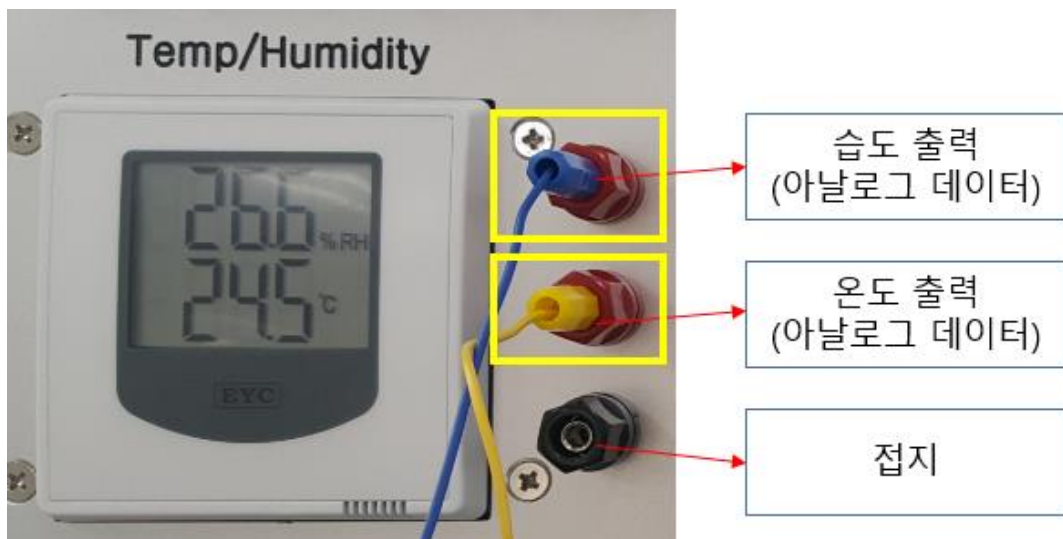
1.2. WISE-4012 연결 구성

- 전원, 아날로그 입력, 디지털 출력



1.3. 온도/습도 센서 연결 구성

- 습도 - 센서 우측의 맨 위 단자 → 파란색 선 / AI0 입력
- 온도 - 센서 우측의 중간 단자 → 노란색 선 / AI1 입력



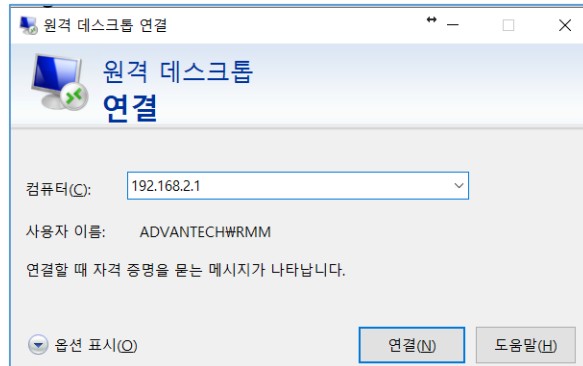
1.4. 임베디드 PC 연결 구성

- 1) 전원, LAN 및 무선 모듈
- 2) LAN 포트는 2곳이 있는데 DC 전원부가 반대편의
- 3) LAN 포트를 이용해 PC에 연결



1.5. 임베디드 PC에 접속

- 원격 데스크톱을 실행하여 192.168.2.1로 접속



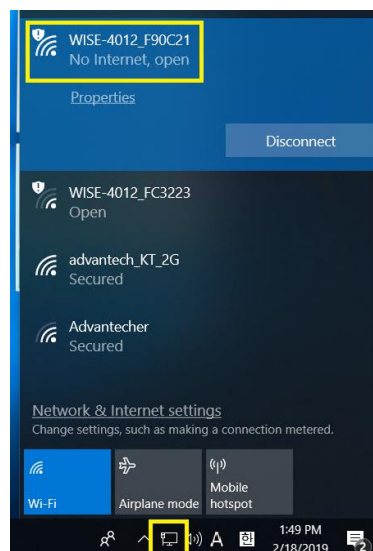
1.6. PC 로그인 정보

- **ID – RMM / Password – 12345678**

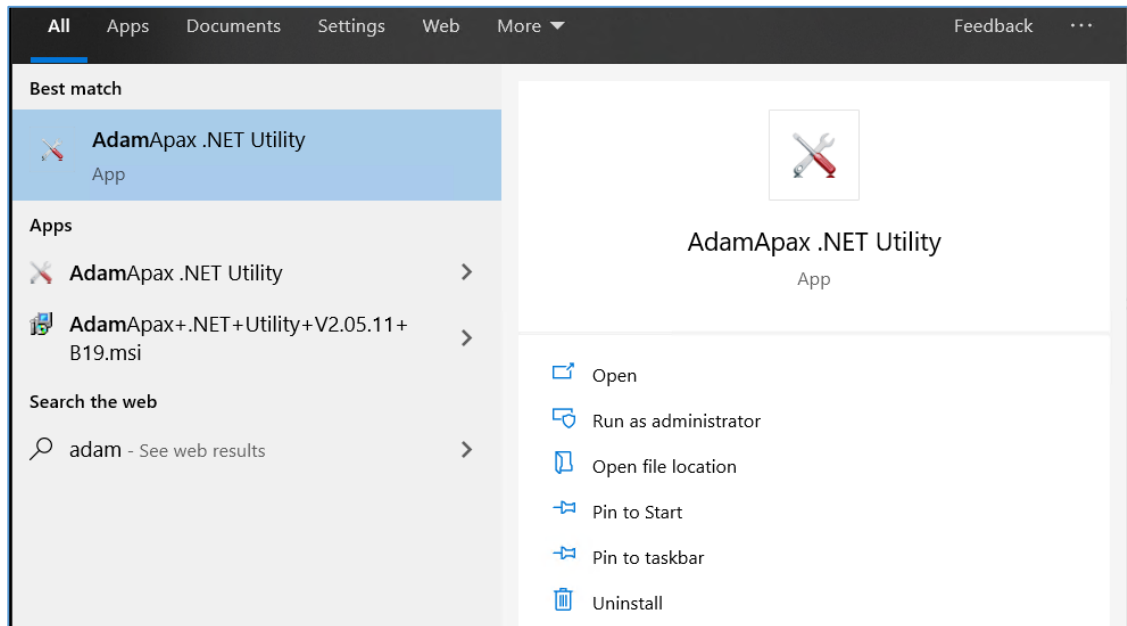
1.7. WISE-4012 연결 확인

WISE-4012와 센서간의 통신을 점검하여 이상 유무를 확인합니다.

- 무선 네트워크의 WISE-4012 접속을 확인합니다.

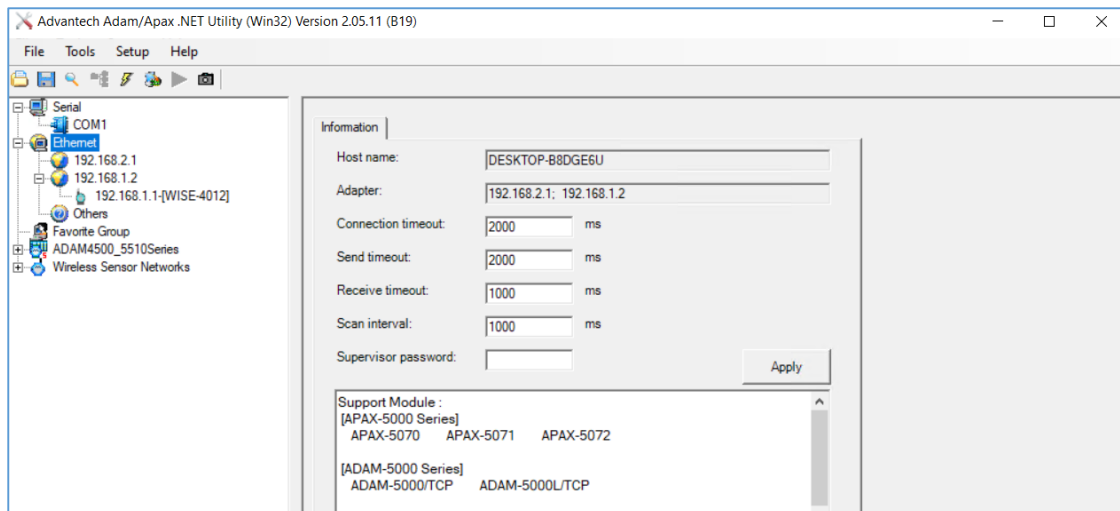


1.8. AdamApax.NET Utility를 실행



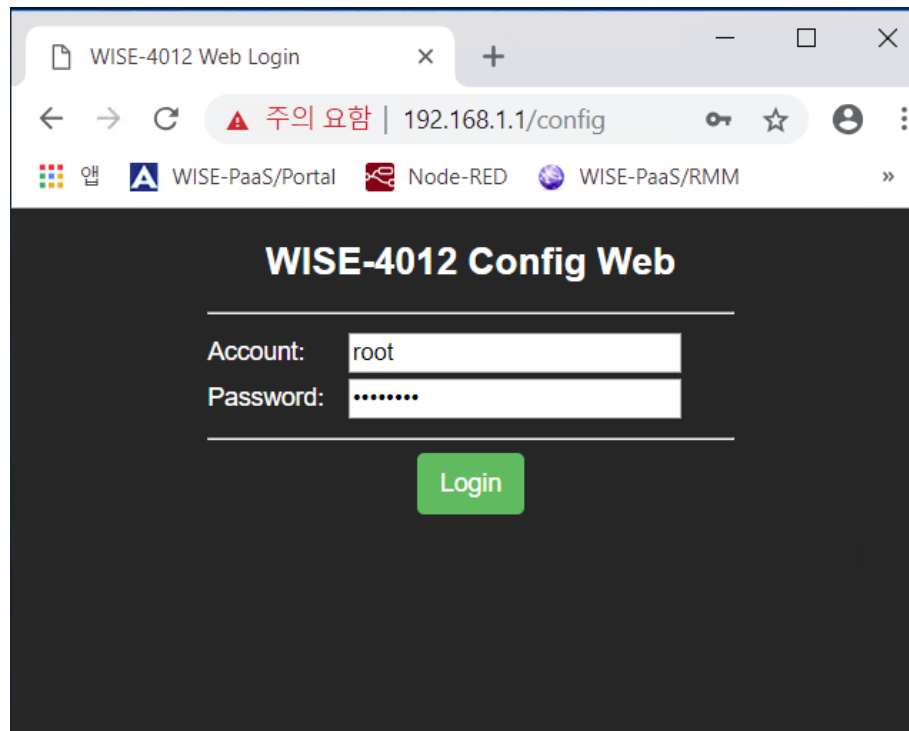
1.9. WISE-4012 장치 인식 확인

Ethernet에서 오른쪽 마우스를 우클릭하여 Search를 선택하여 아래와 같이 WISE-4012가 정상적으로 인식되었는지 확인합니다.



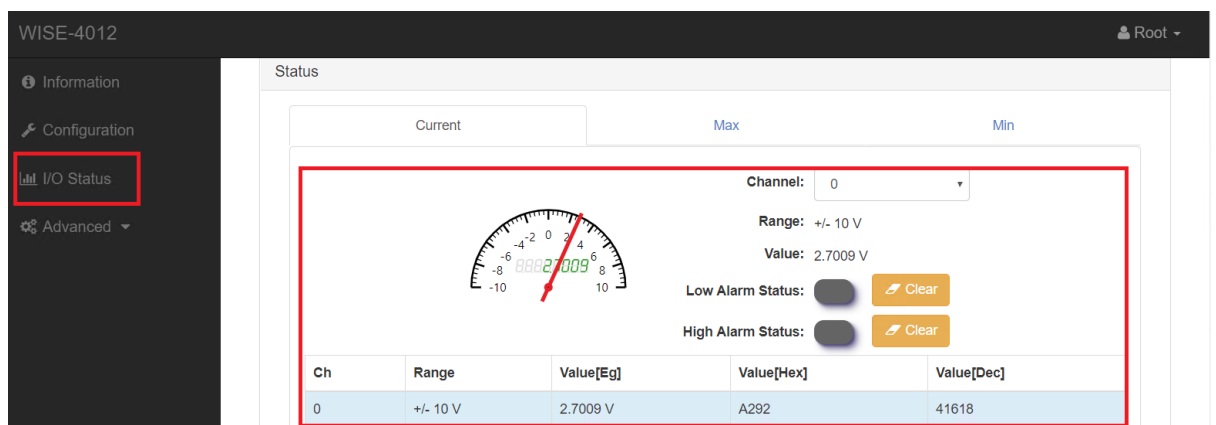
1.10. 웹브라우저 로그인

- WISE-4012의 IP를 입력한 후 아래와 같은 화면에서 로그인합니다.
: IP주소는 WISE-4012 전면에 라벨을 확인하시기 바랍니다.
- **ID – root / Password – 00000000** ("0" 8개)



1.11. WISE-4012의 동작 상태 확인

- AI(아날로그 입력/센서입력) : AI0 – 온도 / AI1 – 습도
- DO(디지털 출력/제어) : DO0 – LED / DO1 - FAN

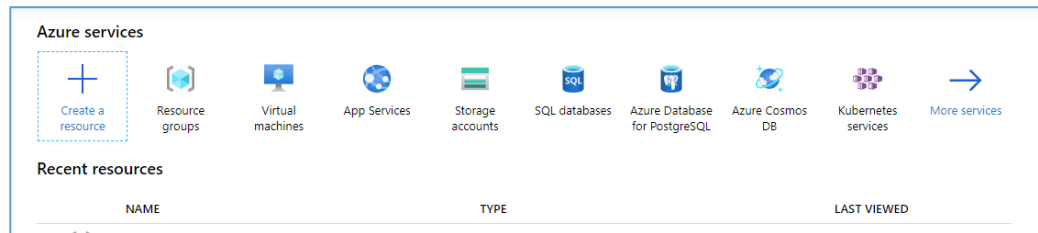


● 실습2 (Microsoft Azure Cloud)

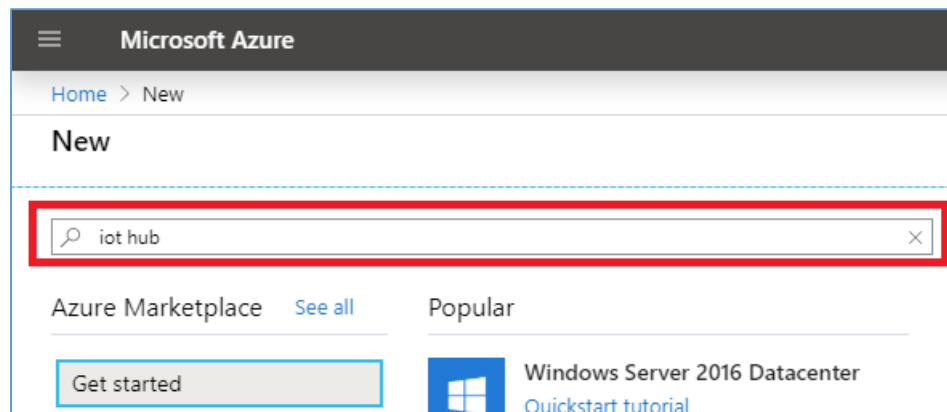
1. Azure Resource 생성

1.1. IoT Hub 생성

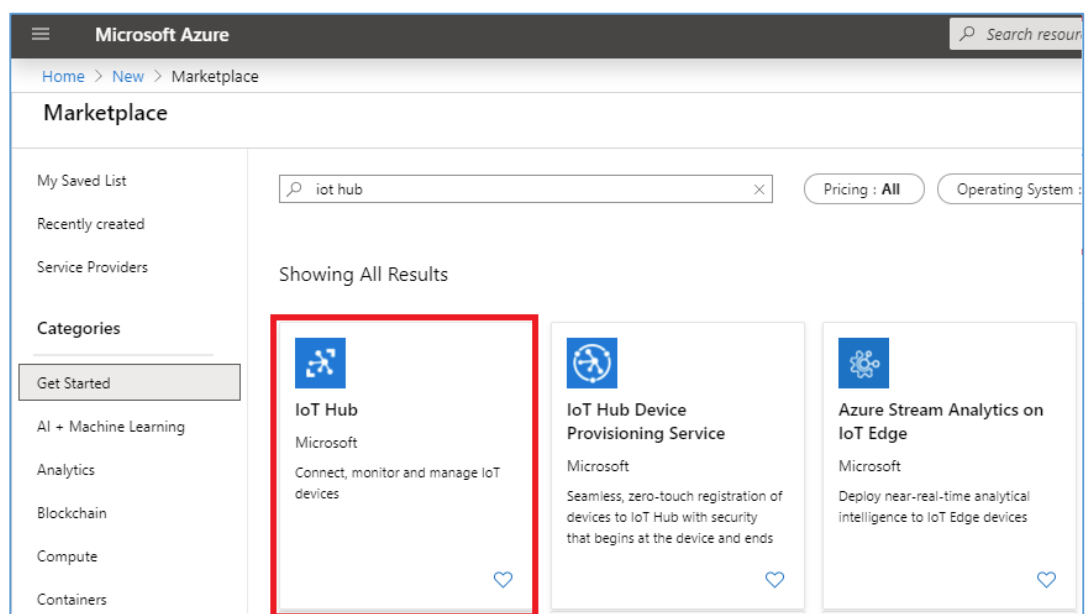
1.1.1. 리소스 만들기



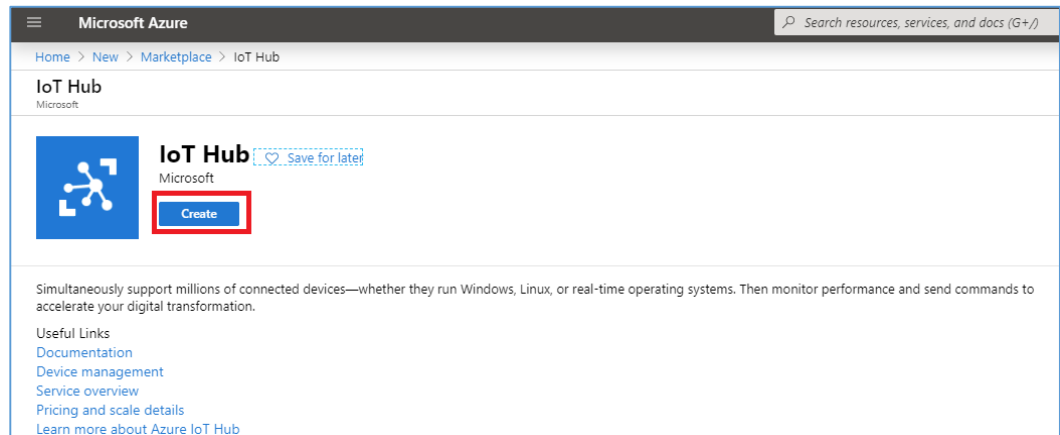
1.1.2. IoT hub 검색



1.1.3. IoT Hub 선택합니다.



1.1.4. “만들기” 선택



1.1.5. IoT Hub의 기본 사항 정보 입력

- 구독 / 리소스 그룹 / 영역 / IoT Hub 이름

The screenshot shows the 'Basics' tab of the Azure IoT Hub creation wizard. The 'PROJECT DETAILS' section is highlighted with a red rectangular box. It contains four fields: 'Subscription' (set to 'Microsoft Azure'), 'Resource Group' (set to '(New) test' with a 'Create new' link below it), 'Region' (set to 'West US'), and 'IoT Hub Name' (set to 'akr20191031'). Above these fields, there's a note: 'Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.'

1.1.6. 크기 및 배율 설정 : F1 – 무료 계층을 선택합니다.

홈 > 리소스 그룹 > ADemo > Marketplace > IoT Hub > IoT 허브

IoT 허브

Microsoft

기본 사항 크기 및 배율 검토 + 만들기

각 IoT Hub는 특정 계층에서 특정 장치 수로 프로비전됩니다. 계층 및 장치 수는 보낼 수 있는 메시지의 최대 일일 할당량을 결정합니다. 자세한 정보

크기 계층 및 단위

* 가격 및 크기 계층 ⓘ

F1: 무료 계층
무료 IoT 허브는 구독당 하나만 허용됩니다.

솔루션에 적합한 IoT Hub 계층을 선택하는 방법 알아보기

F1 IoT Hub 장치 수 ⓘ

이 IoT Hub 확장 기능을 결정하며 요구 사항이 증가함에 따라 변경될 수 있습니다.

가격 및 크기 계층 ⓘ F1	디바이스-클라우드 메시지 ⓘ 사용
일별 메시지 ⓘ 8,000	메시지 라우팅 ⓘ 사용
월별 비용 0.00 USD	클라우드 - 디바이스 명령 ⓘ 사용
	IoT Edge ⓘ 사용
	디바이스 관리 ⓘ 사용

검토 + 만들기 «이전: 기본 사항 자동화 옵션

1.1.7. 입력 정보 확인 후 "만들기"

1.1.8. Resource 그룹에서 IoT Hub를 선택합니다.

Microsoft Azure

Home > test

test Resource group

Subscription (change) : Microsoft Azure
Subscription ID : 13dd3ed4-5208-4cd6-b71c-1cad5ee98842
Tags (change) : Click here to add tags

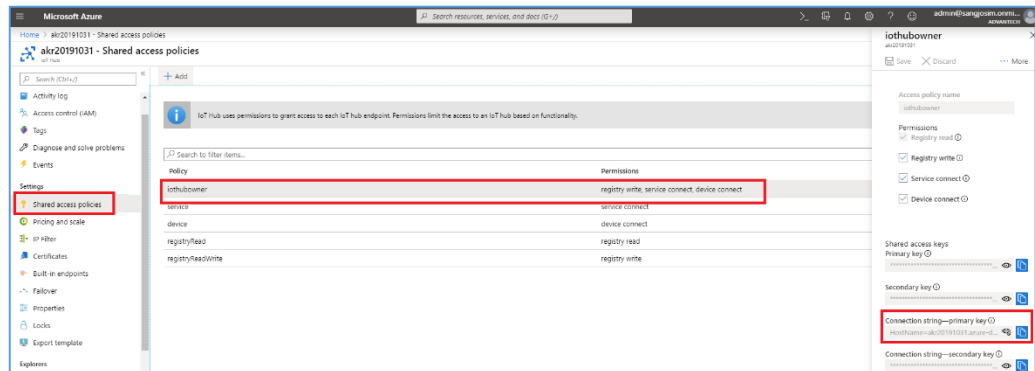
Deployments : 1 Deploying

Filter by name... Type == all Location == all Add filter

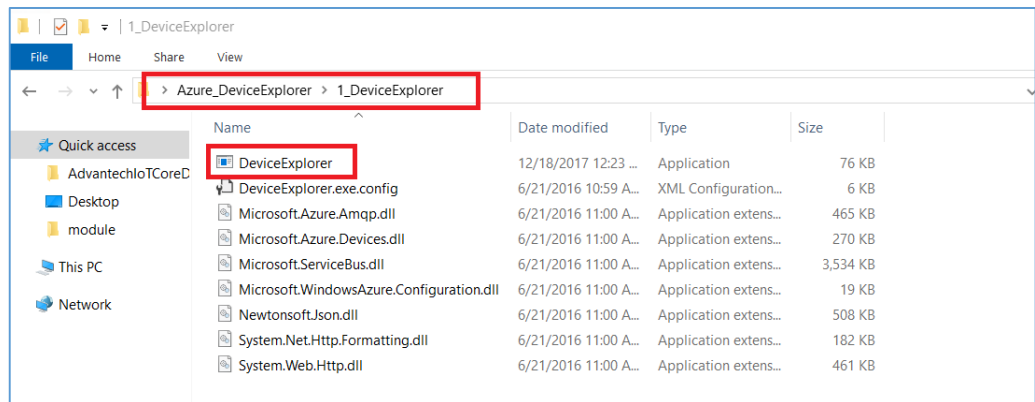
Showing 1 to 1 of 1 records. Show hidden types ⓘ

Name ↑	Type ↑	Location ↑
skc20191031	IoT Hub	West US

- 1.1.9. Shared access policies(공유 액세스 정책)에서 IoTHubowner의 연결 문자열을 선택하여 **primary Key**를 복사합니다.



- 1.1.10. IoT Hub와 정상 연결 확인을 위해 바탕화면의 Device Explorer를 실행합니다.

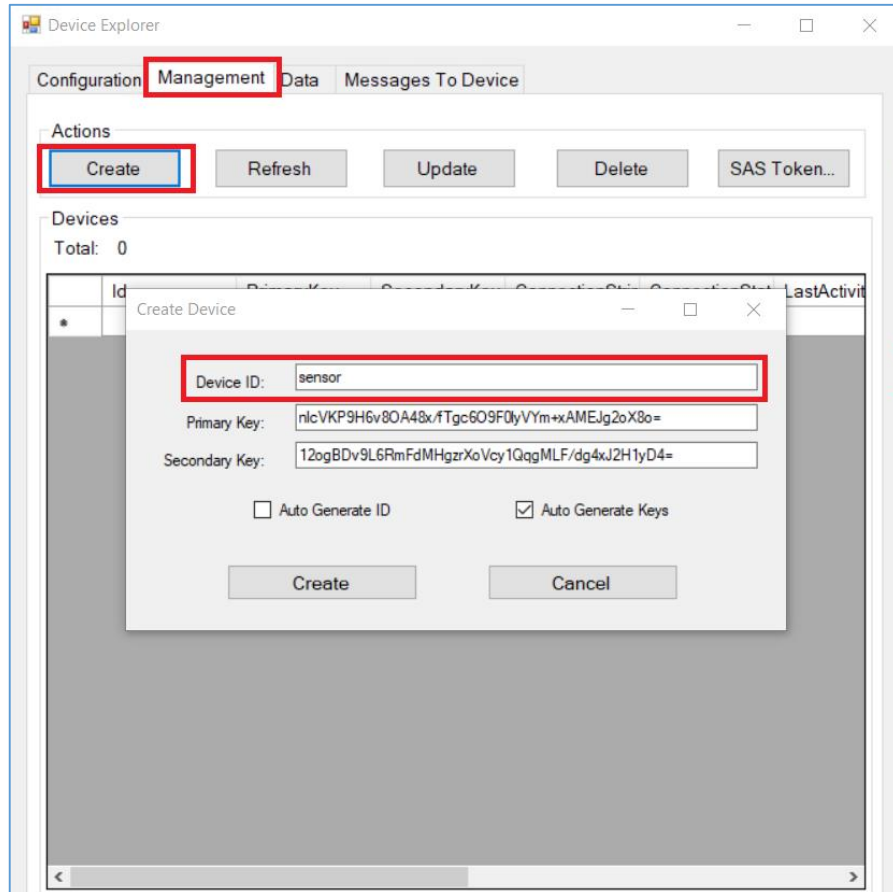


- 1.1.11. 복사한 Primary Key값을 IoT Hub Connection String에 붙여 넣기 한 후 Update를 클릭하여 Shared Access Signature 정보를 확인합니다.

The screenshot shows the 'Device Explorer' application window with the 'Configuration' tab selected. The 'IoT Hub Connection String' field is highlighted with a red box and contains the text: `HostName=akr20191031.azure-devices.net;SharedAccessKeyName=device;SharedAccessKey=dQYG4rivO5Gc2yJM0iyiG4JmdlNpgZBBerUQDba0uaE=`. Below this, the 'Update' button is highlighted with a red box. The 'Shared Access Signature' section is also highlighted with a red box and contains the following fields: 'Key Name' with the value 'device', 'Key Value' with the value 'dQYG4rivO5Gc2yJM0iyiG4JmdlNpgZBBerUQDba0uaE=', and 'Target' with the value 'akr20191031.azure-devices.net'. The 'TTL (Days)' field is set to 365, and the 'Generate SAS' button is visible.

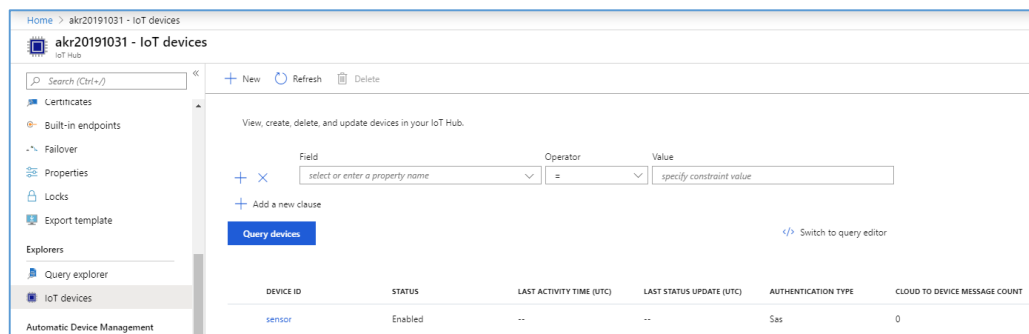
1.1.12. 테스트를 위한 가상 Device를 생성해 보겠습니다.

Management에서 Create를 클릭하여 Device ID를 임의로 생성합니다.



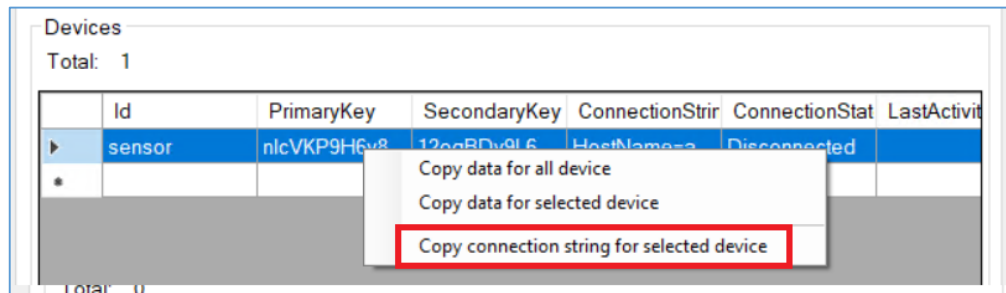
1.1.13. Create를 선택하여 Azure Portal에 Device가 새로 등록됩니다.

Azure Portal에서 직접 생성할 수도 있고 Device Explorer등의 다양한 방법으로 생성 가능합니다.

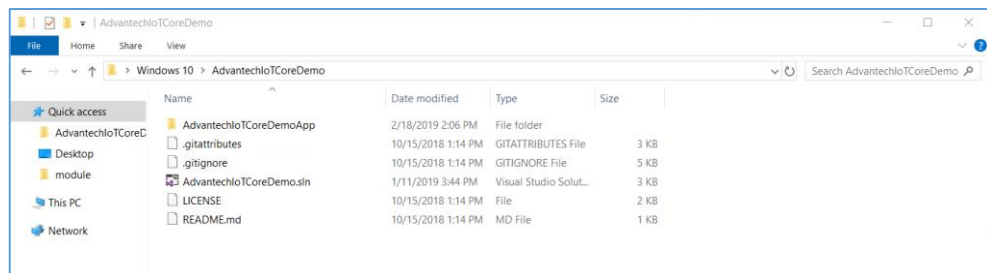


1.1.14. Azure Portal에서 Device를 선택하거나 Device Explorer에서 Device 연결 정보를 확인하고 연결 문자열(Connection String)을 복사합니다.

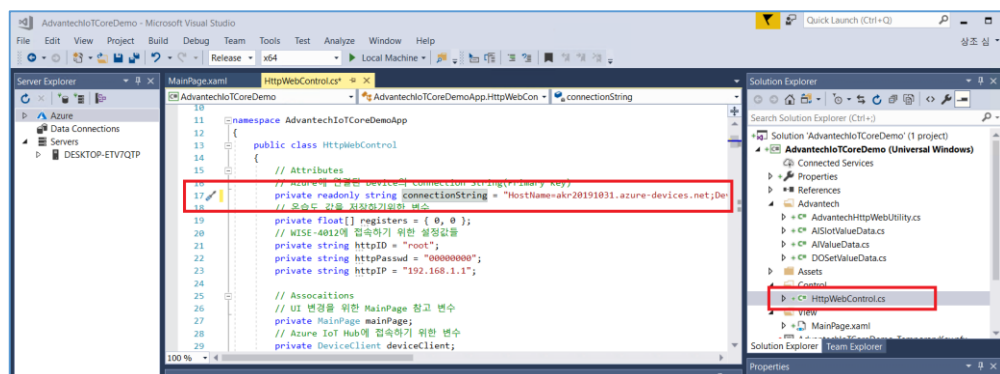
DEVICE ID	STATUS	LAST ACTIVITY TIME (UTC)	LAST STATUS UPDATE (UTC)	AUTHENTICATION TYPE	CLOUD TO DEVICE MESSAGE COUNT
sensor	Enabled	--	--	Sas	0



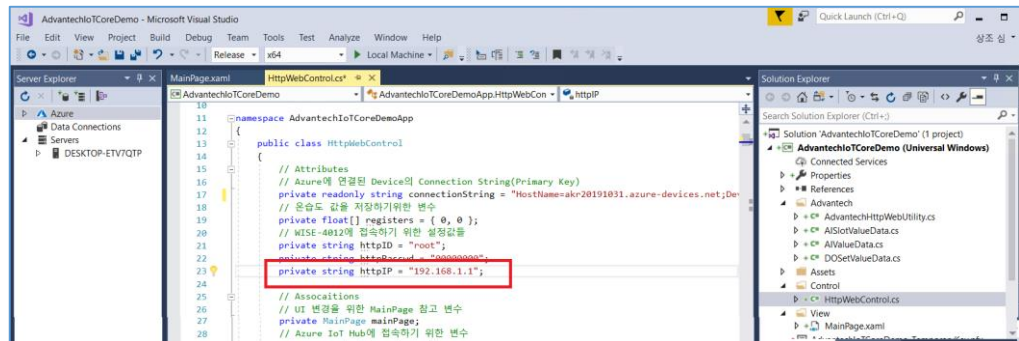
1.1.15. 바탕화면 Windows 10폴더에서 AdvantechIoTCoreDemo.sln을 실행합니다.



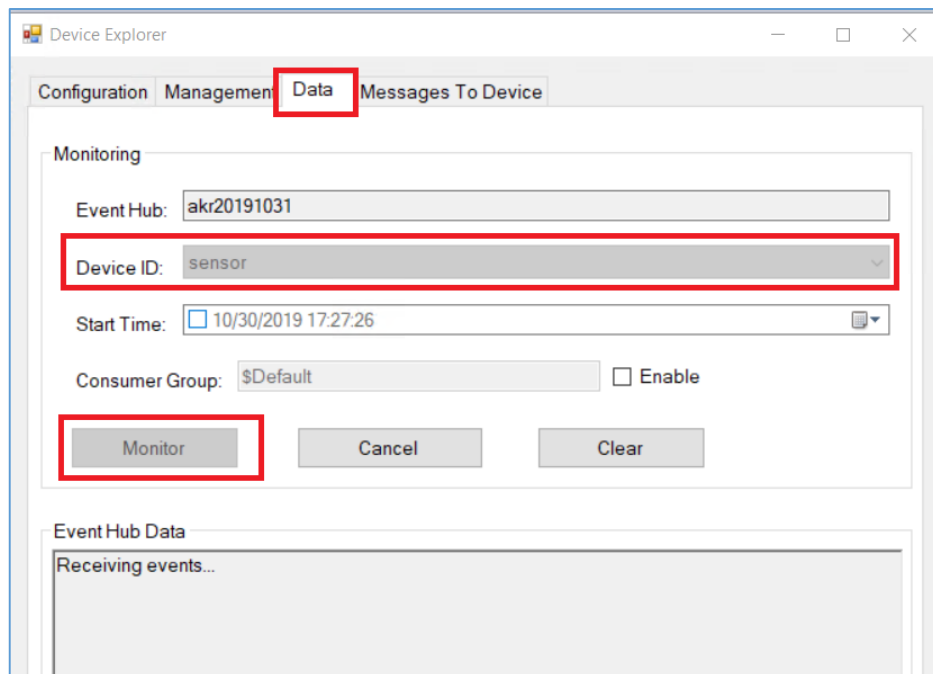
1.1.16. 우측 Solution Explorer에서 HttpWebControl.cs 파일을 열어 17번째 줄의 connectionString 문자열을 복사한 값으로 변경 적용합니다.



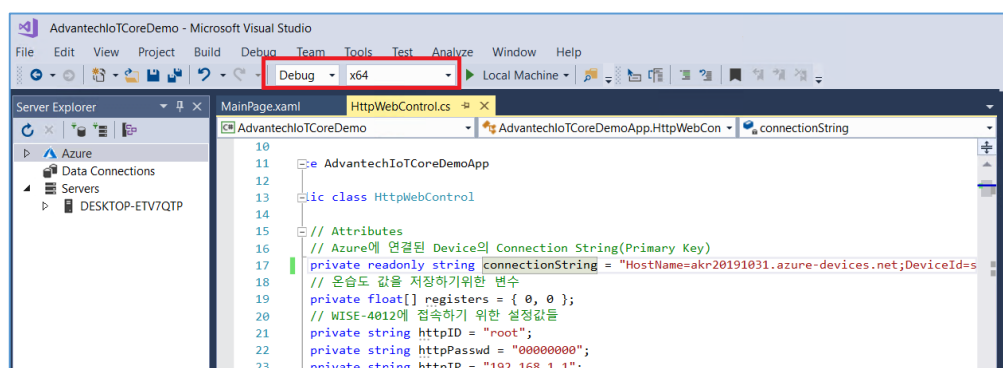
1.1.17. 동일한 파일에서 23번줄의 httpIP 를 WISE-4012의 IP주소로 변경 드립니다.



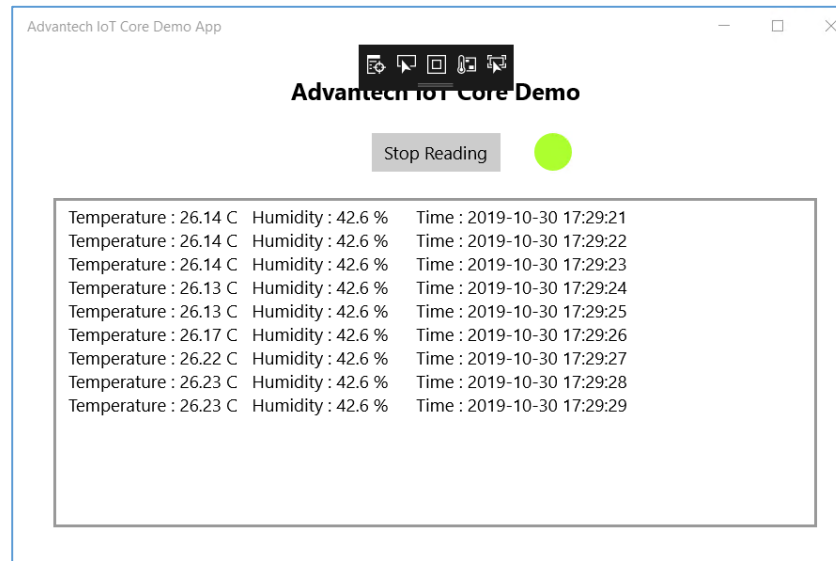
1.1.18. Device Explorer에서 Data 탭에서 Device를 선택하고 Monitor를 클릭합니다.



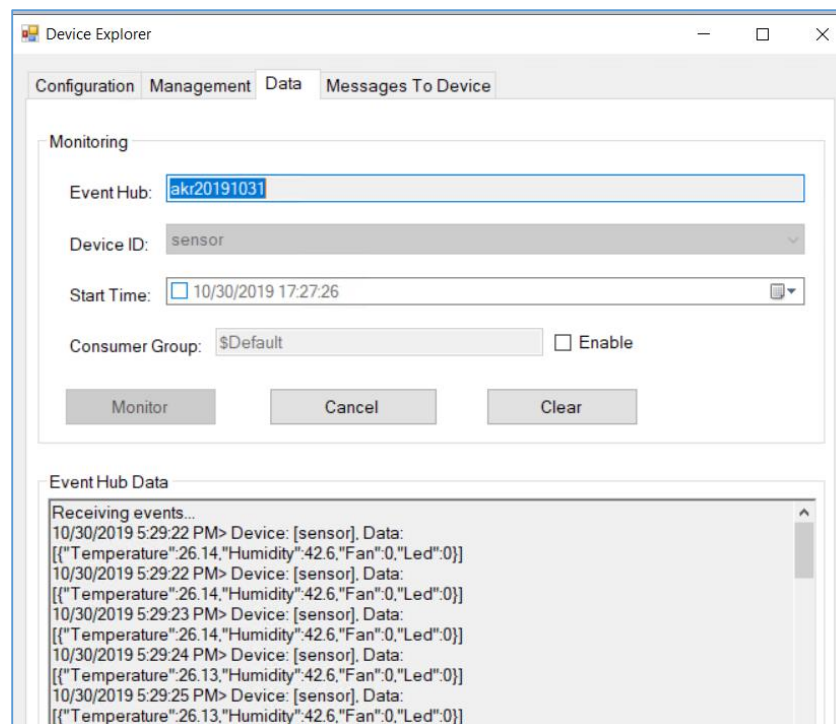
1.1.19. Debug를 실행하여 프로그램을 실행합니다.



1.1.20. Start Reading를 클릭하여 WISE-4012의 온/습도 정보를 수집합니다.



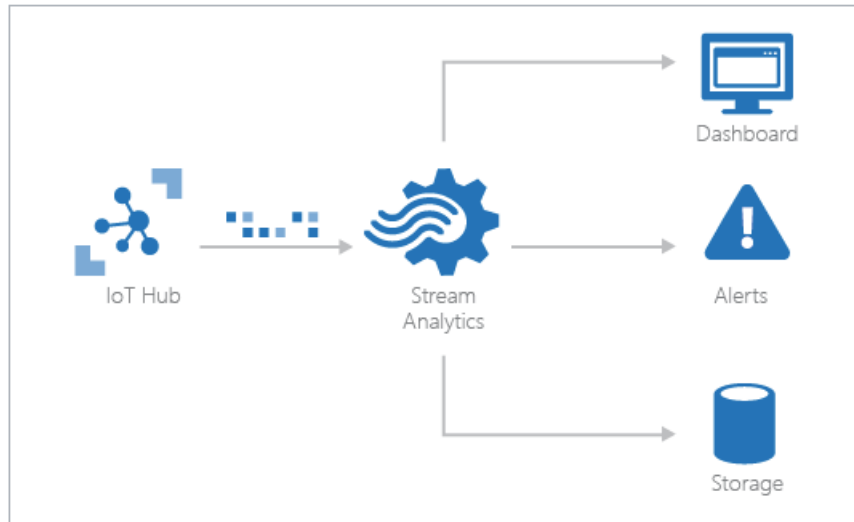
1.1.21. Device Explorer에서 Event 입력을 확인합니다.



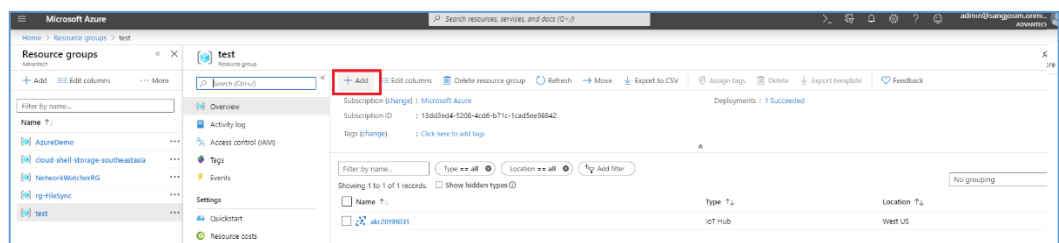
여기까지 센서데이터를 수집하여 IoT Hub로 데이터 전송을 테스트 하였습니다.

다음 페이지에서는 수집된 데이터를 Stream Analytics를 이용하여 처리하는 것에 대한 실습을 진행합니다.

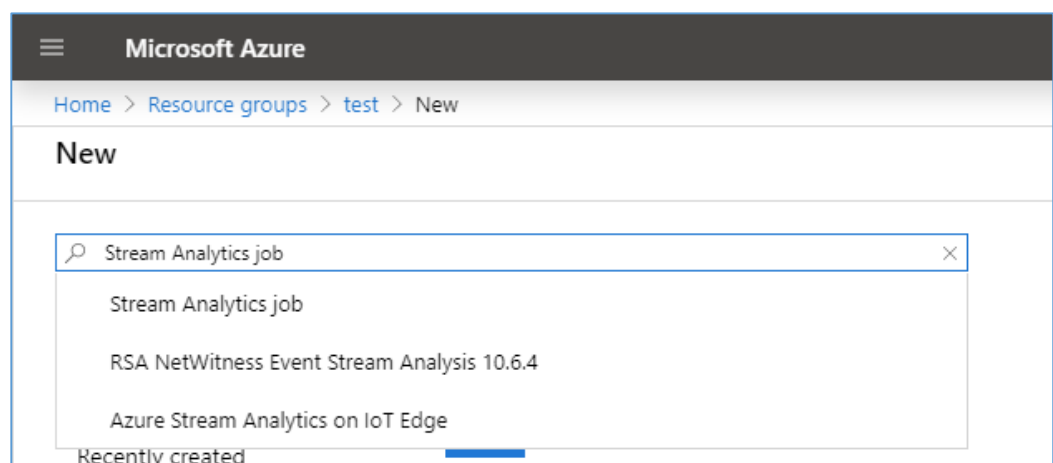
1.2. Stream Analytics Job 생성



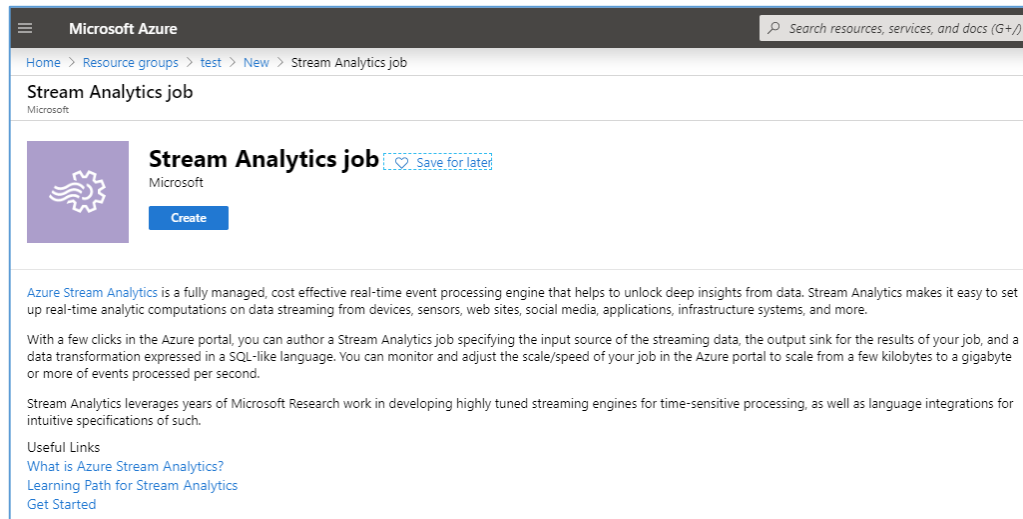
1.2.1. Resource 그룹에서 Add를 선택



1.2.2. Stream Analytics Job을 검색 및 생성



1.2.3. 만들기 클릭



1.2.4. 정보 입력 후 만들기

■ 작업 이름 / 구독 / 리소스 그룹 / 위치

New Stream Analytics job

Job name *

20191031_SA

Subscription *

Microsoft Azure

Resource group *

test

Location *

Central US

Hosting environment ⓘ

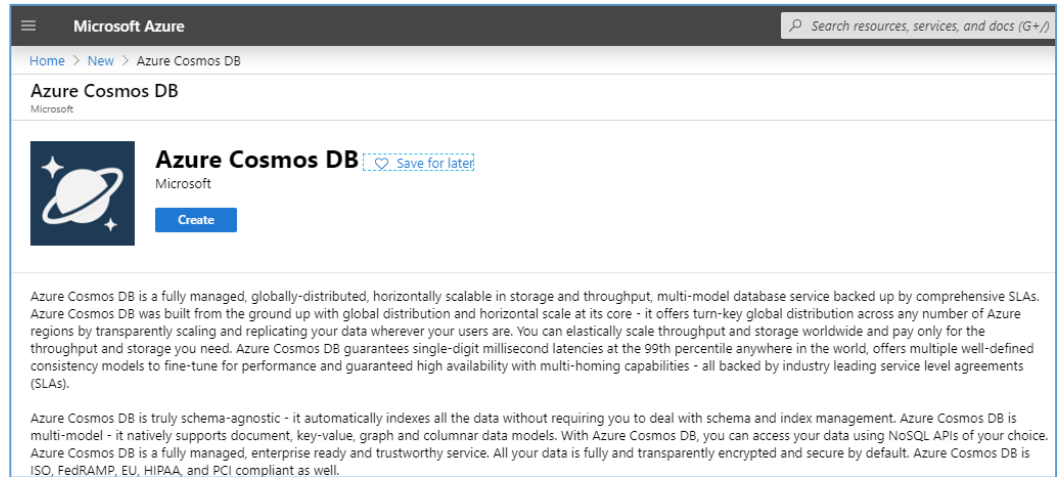
Cloud Edge

Streaming units (1 to 192) ⓘ

3

1.3. COSMOS DB 생성

1.3.1. 리소스 만들기 - Azure Cosmos DB 검색 및 생성



Azure Cosmos DB
Microsoft

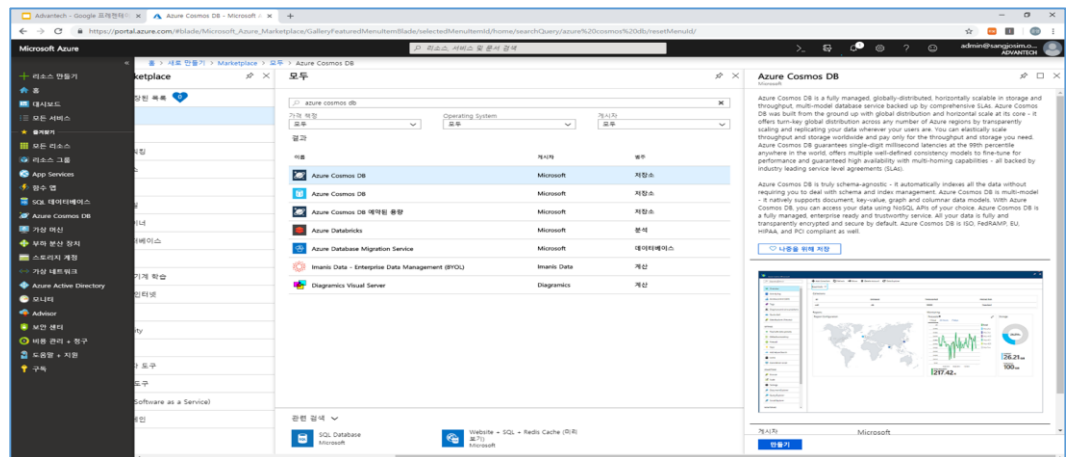
[Save for later](#)

[Create](#)

Azure Cosmos DB is a fully managed, globally-distributed, horizontally scalable in storage and throughput, multi-model database service backed up by comprehensive SLAs. Azure Cosmos DB was built from the ground up with global distribution and horizontal scale at its core - it offers turn-key global distribution across any number of Azure regions by transparently scaling and replicating your data wherever your users are. You can elastically scale throughput and storage worldwide and pay only for the throughput and storage you need. Azure Cosmos DB guarantees single-digit millisecond latencies at the 99th percentile anywhere in the world, offers multiple well-defined consistency models to fine-tune for performance and guaranteed high availability with multi-homing capabilities - all backed by industry leading service level agreements (SLAs).

Azure Cosmos DB is truly schema-agnostic - it automatically indexes all the data without requiring you to deal with schema and index management. Azure Cosmos DB is multi-model - it natively supports document, key-value, graph and columnar data models. With Azure Cosmos DB, you can access your data using NoSQL APIs of your choice. Azure Cosmos DB is a fully managed, enterprise ready and trustworthy service. All your data is fully and transparently encrypted and secure by default. Azure Cosmos DB is ISO, FedRAMP, EU, HIPAA, and PCI compliant as well.

1.3.2. 만들기를 선택



Azure Cosmos DB
Microsoft

[Save for later](#)

[Create](#)

Azure Cosmos DB is a fully managed, globally-distributed, horizontally scalable in storage and throughput, multi-model database service backed up by comprehensive SLAs. Azure Cosmos DB was built from the ground up with global distribution and horizontal scale at its core - it offers turn-key global distribution across any number of Azure regions by transparently scaling and replicating your data wherever your users are. You can elastically scale throughput and storage worldwide and pay only for the throughput and storage you need. Azure Cosmos DB guarantees single-digit millisecond latencies at the 99th percentile anywhere in the world, offers multiple well-defined consistency models to fine-tune for performance and guaranteed high availability with multi-homing capabilities - all backed by industry leading service level agreements (SLAs).

Azure Cosmos DB is truly schema-agnostic - it automatically indexes all the data without requiring you to deal with schema and index management. Azure Cosmos DB is multi-model - it natively supports document, key-value, graph and columnar data models. With Azure Cosmos DB, you can access your data using NoSQL APIs of your choice. Azure Cosmos DB is a fully managed, enterprise ready and trustworthy service. All your data is fully and transparently encrypted and secure by default. Azure Cosmos DB is ISO, FedRAMP, EU, HIPAA, and PCI compliant as well.

1.3.3. 정보 입력 후 만들기

- 구독 / 리소스 그룹 / DB 계정 이름 / API 선택 / 지역 선택

Basics Network Tags Review + create

Azure Cosmos DB is a globally distributed, multi-model, fully managed database service. [Try it for free](#), for 30 days with unlimited renewals. Go to production starting at \$24/month per database, multiple containers included. [Learn more](#)

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource Group * [Create new](#)

Instance Details

Account Name * ✓

API * ①

Apache Spark ① ☒ None [Sign up for Apache Spark preview](#)

Location *

Geo-Redundancy ① ☐ Enable ☒ Disable

Multi-region Writes ① ☐ Enable ☒ Disable

*Up to 33% off multi-region writes is available to qualifying new accounts only. Accounts must be created between August 15, 2019 and November 30, 2019. Offer limited to accounts with both account locations and geo-redundancy only in North Europe, West Europe, South Central US, or North Central US regions, and applies only to multi-region writes in those same regions. Both Geo-Redundancy and Multi-region Writes must be enabled in account settings. Actual discount will vary based on number of qualifying regions selected.

1.3.4. 만들기

Basics Network Tags Review + create

Basics

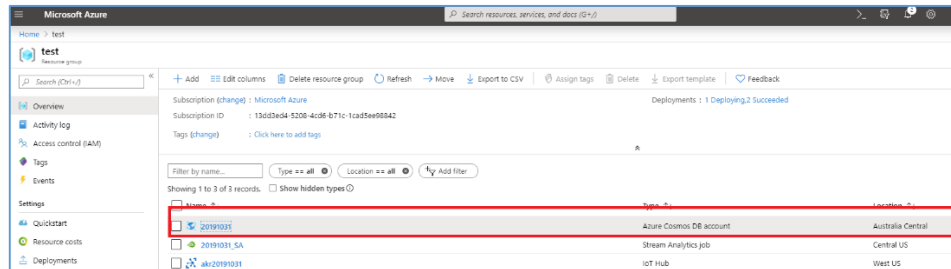
Subscription	Microsoft Azure
Resource Group	test
Location	(Asia Pacific) Australia Central
Account Name	(new) 20191031
API	Core (SQL)
Geo-Redundancy	Disable
Multi-region Writes	Disable

[Create](#) [Previous](#) [Next](#) [Download a template for automation](#)

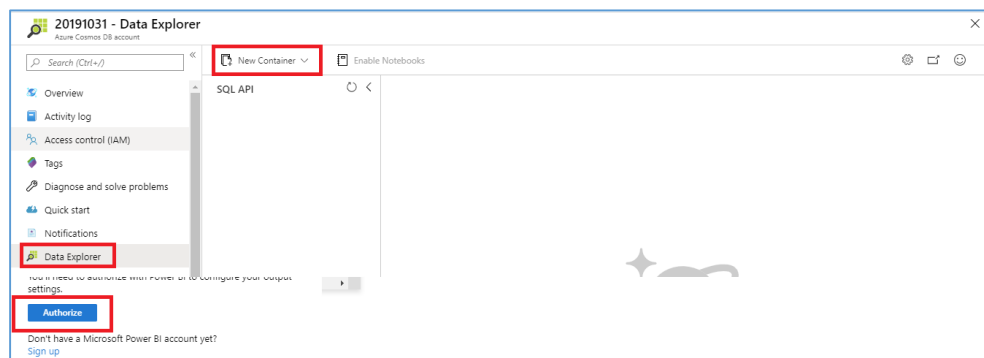
2. Resource 항목 설정

2.1. Cosmos DB 설정

2.1.1. Cosmos DB 선택



2.1.2. 데이터 탐색기(Data Explorer)에서 New Container



2.1.3. 정보 작성 후 "OK" 클릭

- Database ID / Container ID / Partition Key / Throughput

Add Container

i Start at \$24/mo per database, multiple containers included
[More details](#)

* Database id *i*
☒ Create new ☐ Use existing
dev

☒ Provision database throughput *i*
* Throughput (400 - 100,000 RU/s) *i*
400

Estimated spend (USD): **\$0.032 hourly / \$0.77 daily** (1 region, 400RU/s, \$0.00008/RU)

* Container id *i*
iot

* Partition key *i*
/message

☐ My partition key is larger than 100 bytes

2.2. Stream Analytics Job 설정

2.2.1. Stream Analytics Job 선택

Filter by name...	Type == all	Location == all	Add filter
Showing 1 to 3 of 3 records. <input type="checkbox"/> Show hidden types			
<input type="checkbox"/>	Name ↑		Type ↑
<input type="checkbox"/>	20191031		Azure Cosmos DB account
<input type="checkbox"/>	20191031_SA		Stream Analytics job
<input type="checkbox"/>	akr20191031		IoT Hub

2.2.2. 입력 설정 : IoT hub

20191031_SA - Inputs
Stream Analytics job

Search (Ctrl+/)

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

Settings
Locks
Job topology
Inputs
Functions

+ Add stream input + Add reference input

Event Hub
IoT Hub
Blob storage

Source type	Source

2.2.3. 정보 작성 후 저장

IoT Hub
New input

Input alias *

iot-hub ✓

☐ Provide IoT Hub settings manually
☒ Select IoT Hub from your subscriptions

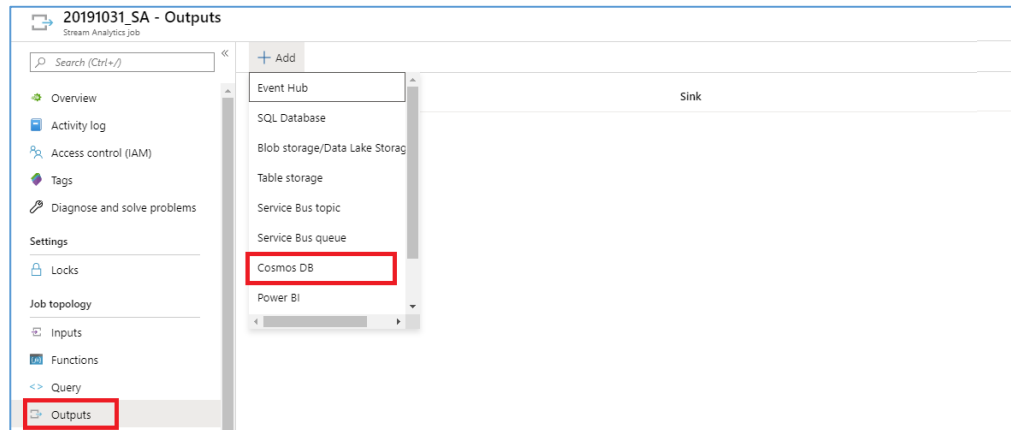
Subscription
Microsoft Azure ▼

IoT Hub ⓘ
akr20191031 ▼

Endpoint ⓘ
Messaging ▼

Shared access policy name ⓘ
iothubowner ▼

2.2.4. 출력 : Cosmos DB 추가



2.2.5. 정보 작성 후 저장

- 출력 명칭 / 구독 / 계정 ID / 데이터베이스 / 컬렉션 이름 패턴

Output details

about

Test

Delete

Output alias

about

Provide Cosmos DB settings manually

Select Cosmos DB from your subscriptions

Subscription

Subscription information not needed

Account id *

20191031

Account key

.....

Database *

Create new

Use existing

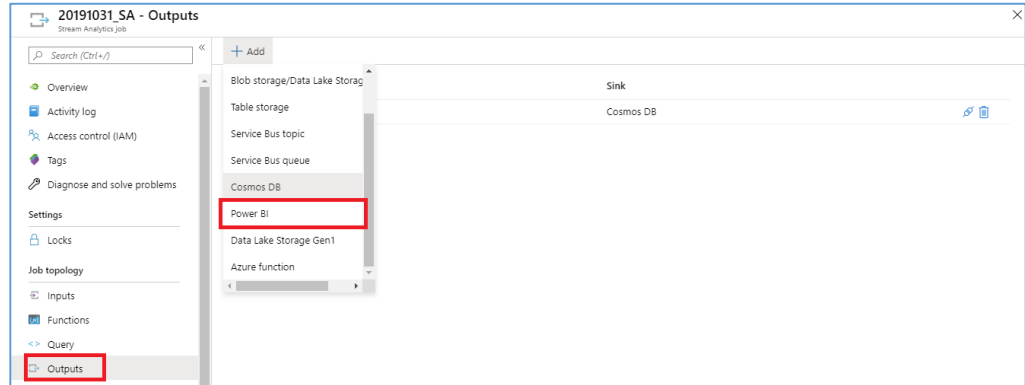
dev

Container name *

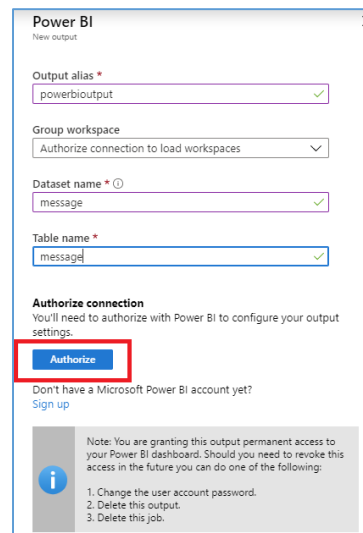
iot

Document id

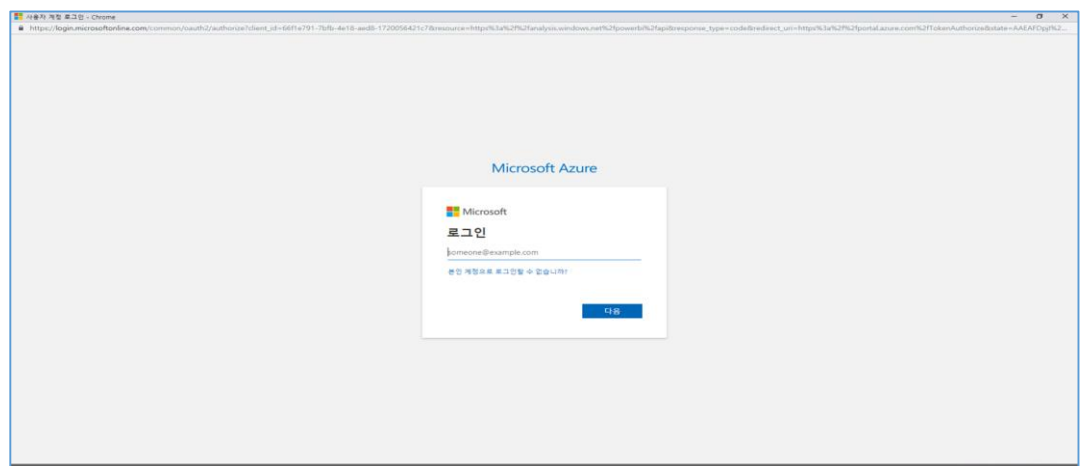
2.2.6. 추가 클릭 → Power BI 클릭 (Power BI로 데이터를 전송할 경우)



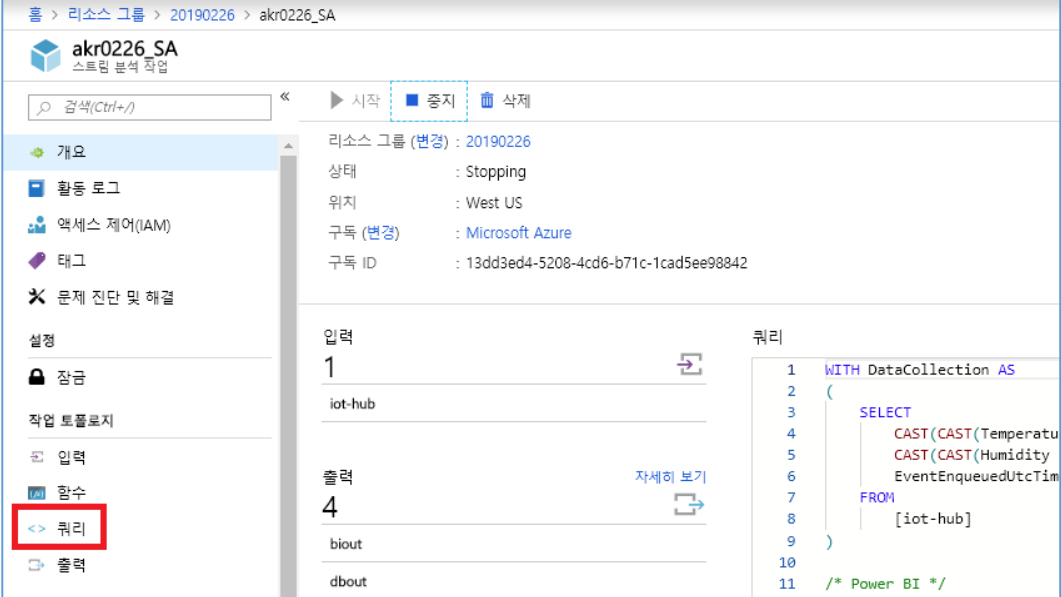
2.2.7. 정보 작성 후 권한 부여 클릭



2.2.8. Power BI를 사용할 계정 로그인



2.2.9. 쿼리 선택



The screenshot shows the Advantech IoT Hub console for the 'akr0226_SA' resource group. The left sidebar has the '쿼리' (Query) option highlighted with a red box. The main area shows the '쿼리' (Query) tab with a SQL query editor and a '쿼리' (Query) section displaying the query results.

쿼리

```

1 WITH DataCollection AS
2 (
3     SELECT
4         CAST(CAST(Temperature * 10 as bigint) as float) / 10 as Temperature,
5         CAST(CAST(Humidity * 10 as bigint) as float) / 10 as Humidity,
6         EventEnqueuedUtcTime as Time
7     FROM
8         [iot-hub]
9 )
10
11 /* Power BI */
  
```

2.2.10. 코드 작성

```

WITH DataCollection AS
(
    SELECT
        CAST(CAST(Temperature * 10 as bigint) as float) / 10 as Temperature,
        CAST(CAST(Humidity * 10 as bigint) as float) / 10 as Humidity,
        EventEnqueuedUtcTime as Time
    FROM
        [iot-hub]
)

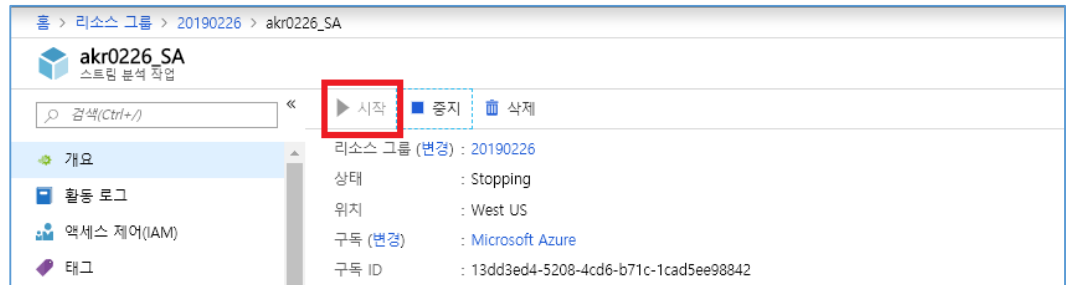
/* Cosmos DB */
SELECT
    Temperature,
    Humidity,
    Time
INTO
    [dboutput]
FROM
    DataCollection

/* Power BI */
SELECT
    Temperature,
    Humidity,
    Time
INTO
    [bioutput]
FROM
    DataCollection
  
```

3. 각 리소스의 동작 유무 확인

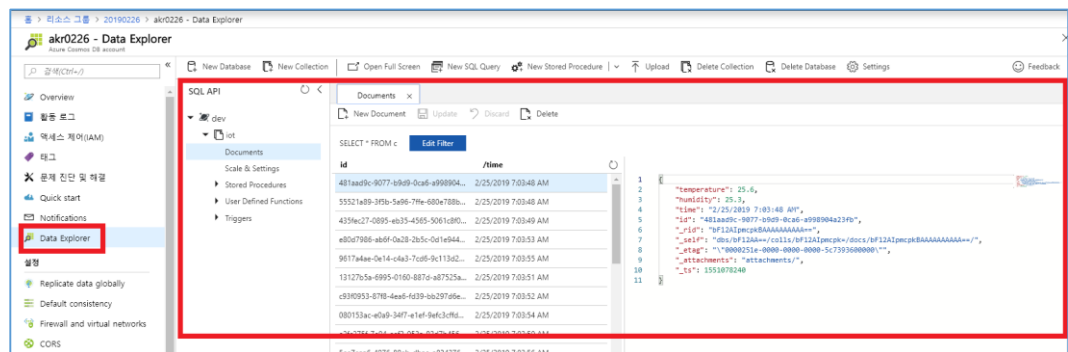
3.1. Stream Analytics 작업 시작

■ 개요 → 시작



3.2. Cosmos DB 데이터 확인

- Data Explorer → SQL API



4. Power BI – Cosmos DB

4.1. Power BI – Cosmos DB 연결

<https://docs.microsoft.com/ko-kr/azure/cosmos-db/powerbi-visualize>

4.1.1. Power BI Desktop 다운로드

4.1.2. 실행 후 초기 설정

Welcome to Power BI Desktop

Where can we send you the latest tips and tricks for Power BI?

Sangjo

Sim

Sangjo.Sim@advantech.co.kr

VOIP:582 EXT:285

Korea

Advantech

50-249

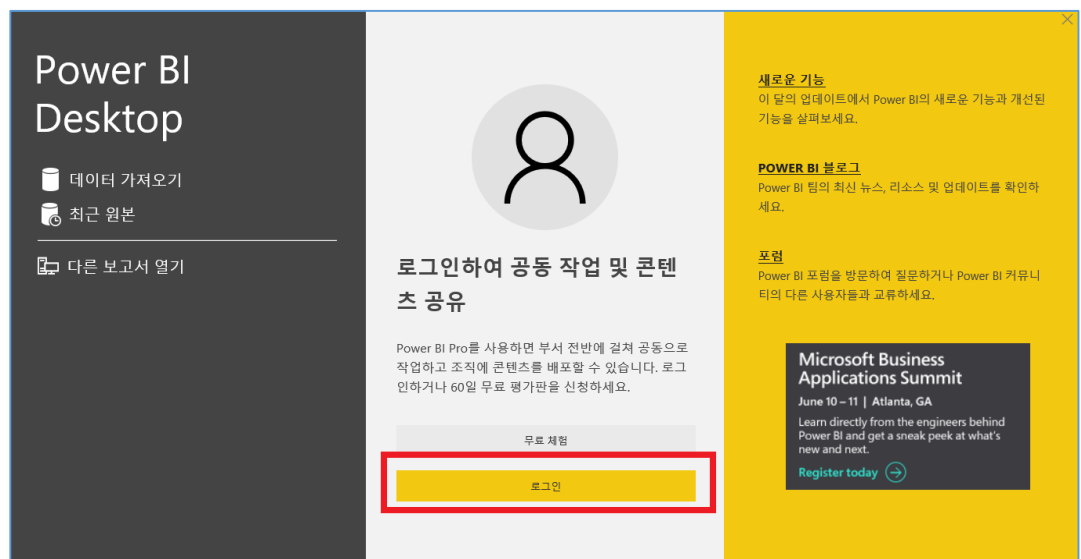
Developer / Engineer

Microsoft may use your contact information to provide updates and special offers about Business Intelligence and other Microsoft products and services. You can unsubscribe at any time. To learn more you can read the [privacy statement](#).

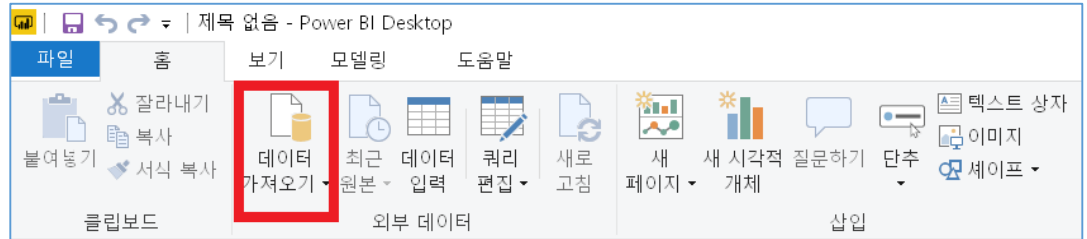
☒ I accept

[Power BI 계정이 이미 있으신가요? 로그인](#)

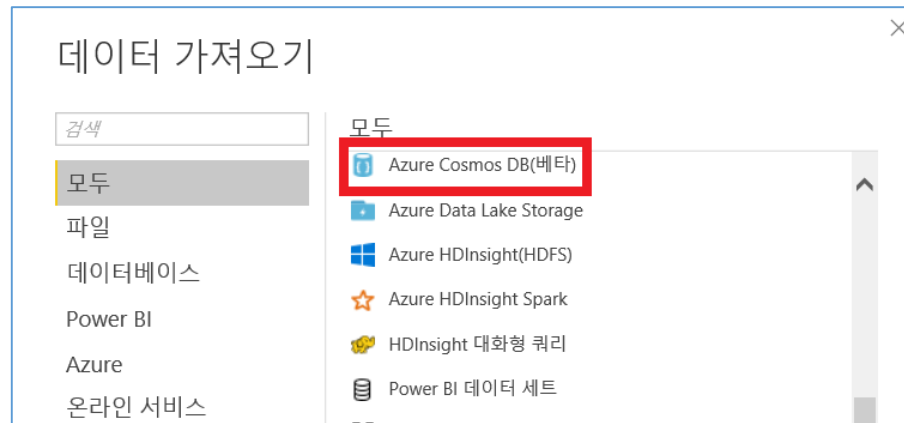
4.1.3. 로그인



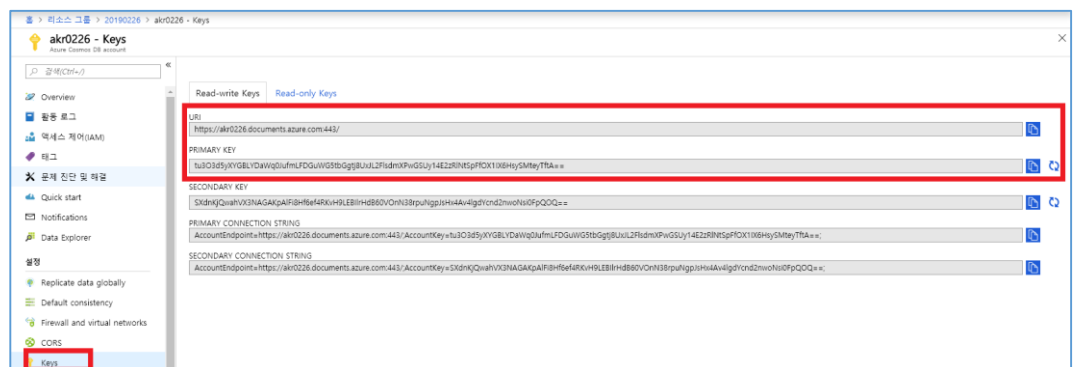
4.1.4. 데이터 가져오기



4.1.5. Azure Cosmos DB(베타) 선택



4.1.6. 생성된 Cosmos DB 정보 확인 : Key



4.1.7. Cosmos DB - URL 정보 입력

Azure Cosmos DB

X

URL(예: https://contoso.documents.azure.com)

https://akr0226.documents.azure.com:443/

데이터베이스(선택 사항)

컬렉션(선택 사항)

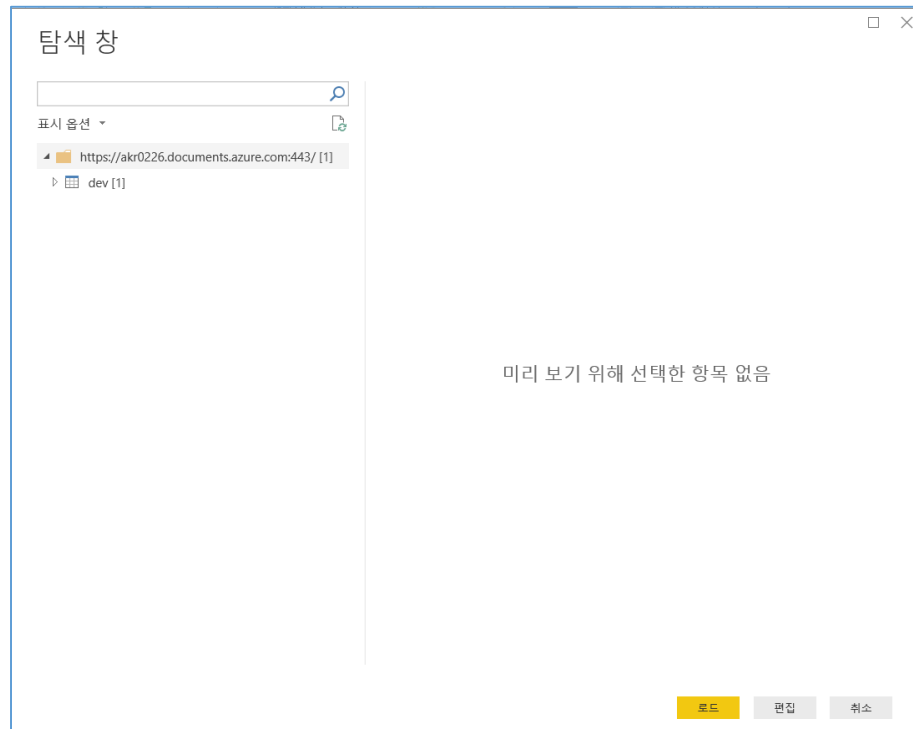
> SQL 문(선택 사항)

확인

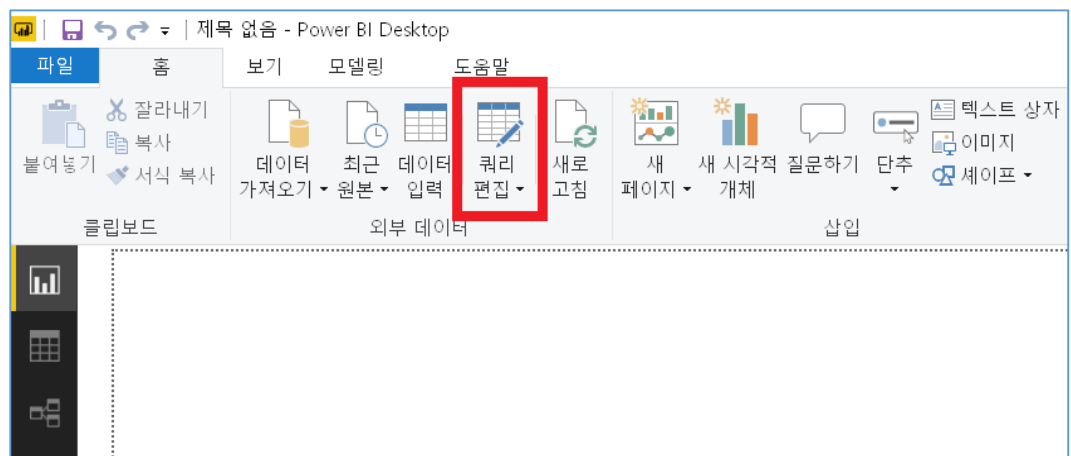
취소

4.1.8. Cosmos DB - 계정 키값 입력

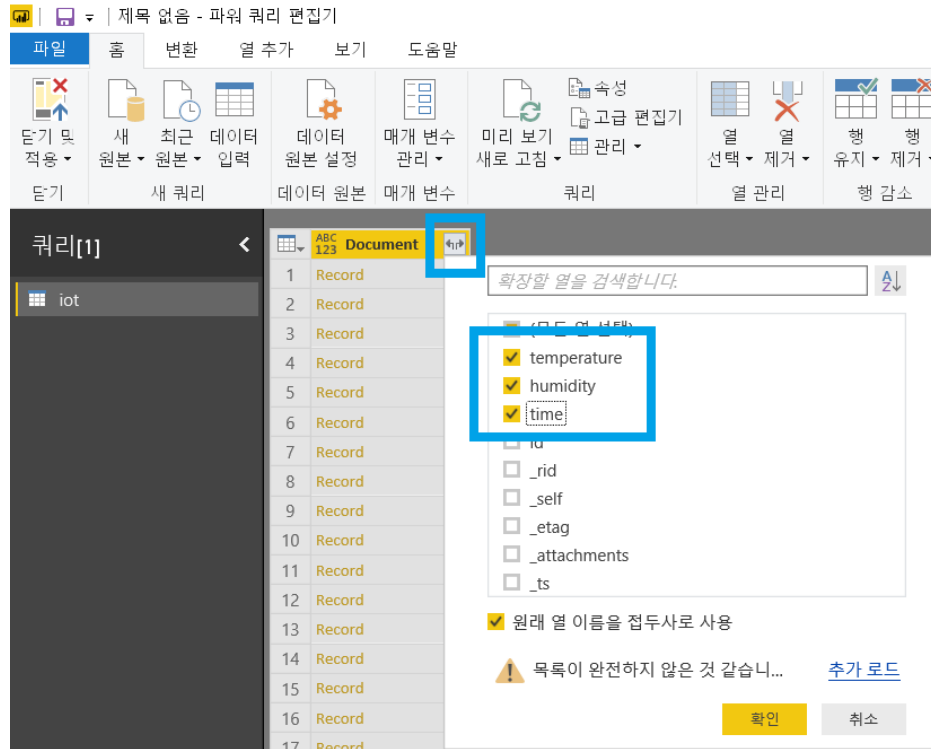
4.1.9. 데이터 베이스 확인 후 “로드”



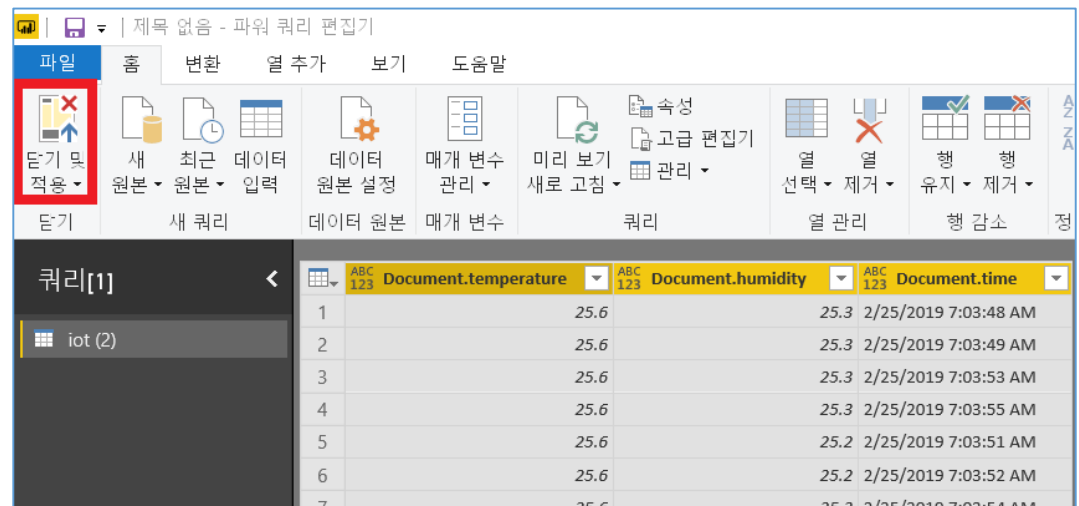
4.2. 데이터베이스 쿼리 편집



4.2.1. 쿼리 편집기에서 열 선택

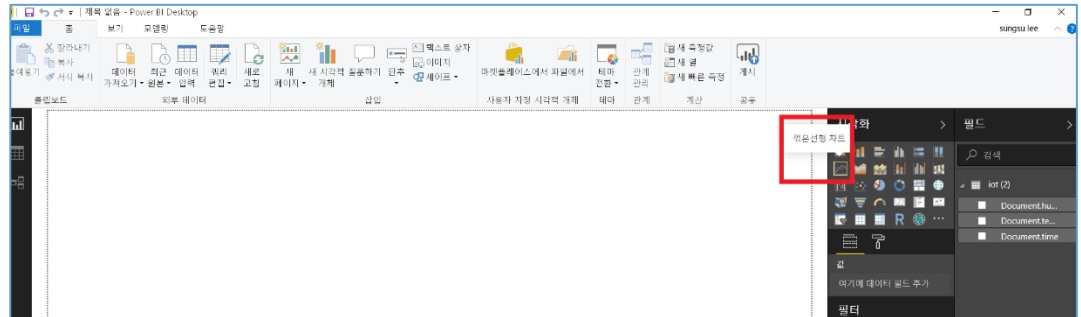


4.2.2. 닫기 및 적용



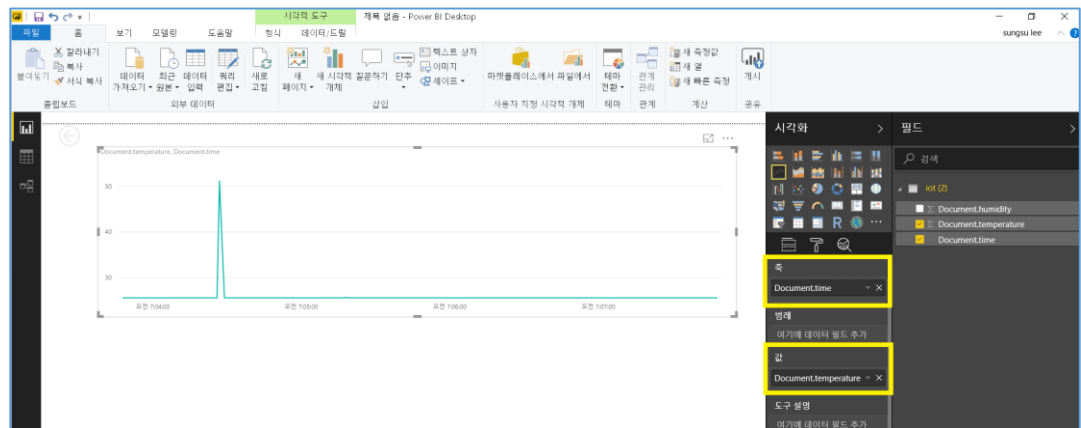
4.3. Dashboard 위젯 구성

4.3.1. 꺾은 선형 차트 선택



4.3.2. 축과 값에 필드 추가

- 축 - time / 값 - 온도, 습도 값



● Appendix : Azure 연결 확인

Device Explorer를 이용하여 IoT Device에서 발생한 시뮬레이션 데이터가 IoT Hub에 정상적으로 수집이 되는지 통신을 확인할 수 있습니다.

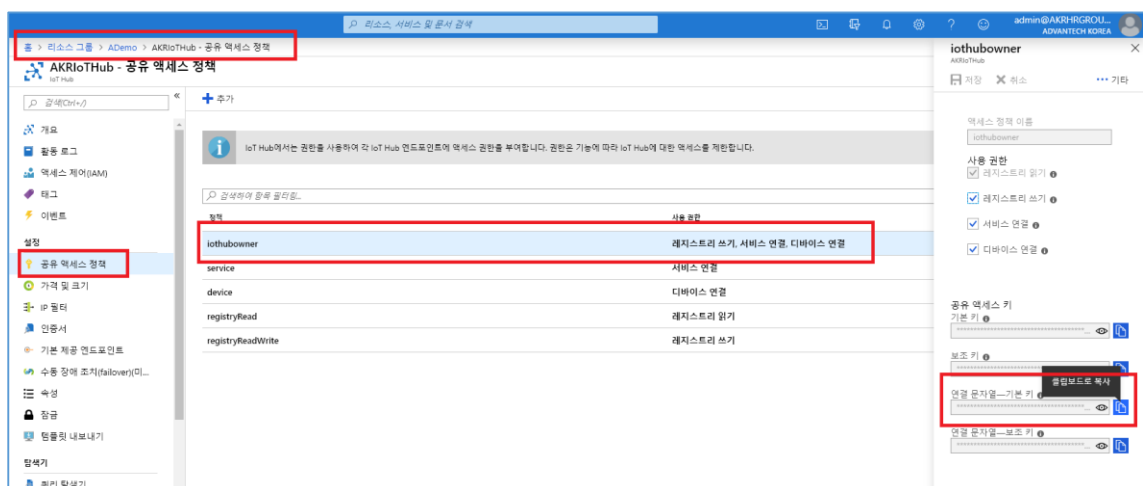
1. Device Explorer 다운로드(GitHub)

바탕화면의 파일 폴더를 사용하며 위의 다운로드 링크는 Source 코드 입니다.

2. Device Explorer 설정

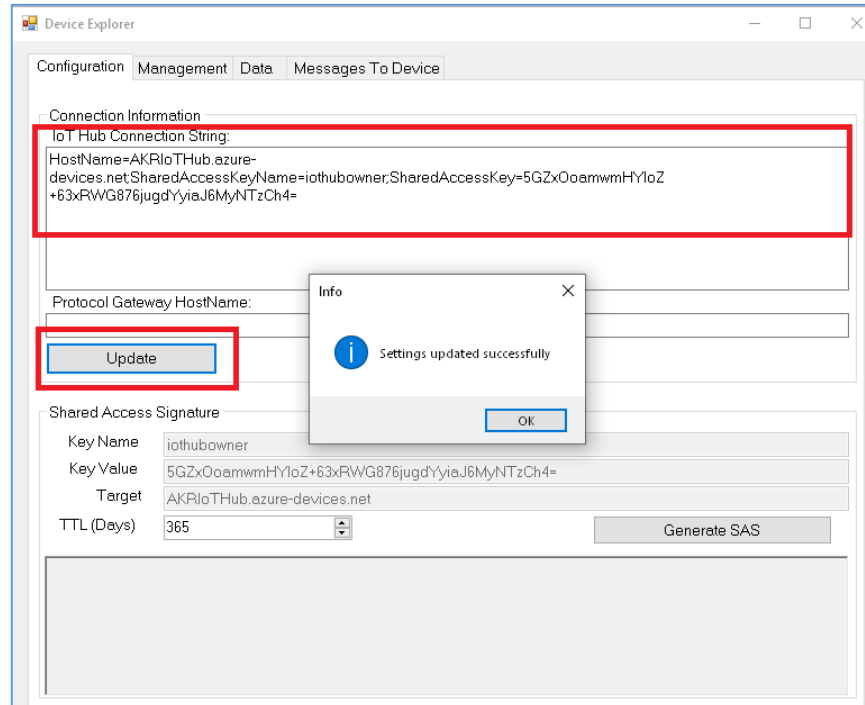
2.1. 리소스 그룹 → Resource Group → IoT Hub → 공유 액세스 정책

2.2. iothubowner를 클릭하고 "연결 문자열"을 클립보드로 복사 합니다.



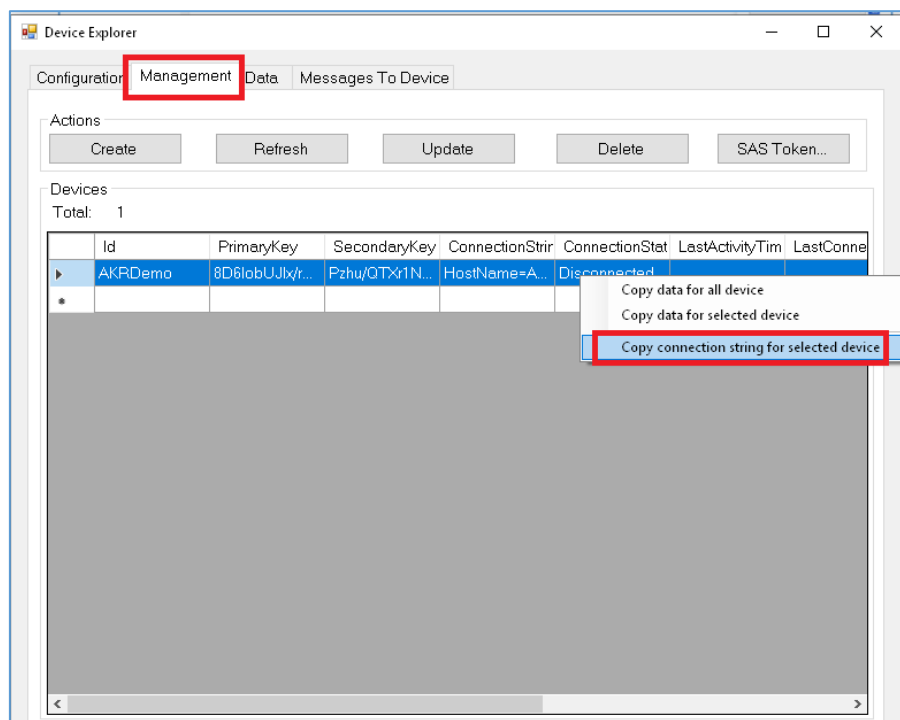
2.3. Device Explorer를 관리자 권한으로 실행합니다.

2.4. Connection Information에 연결 문자열을 입력하고 업데이트를 클릭합니다.

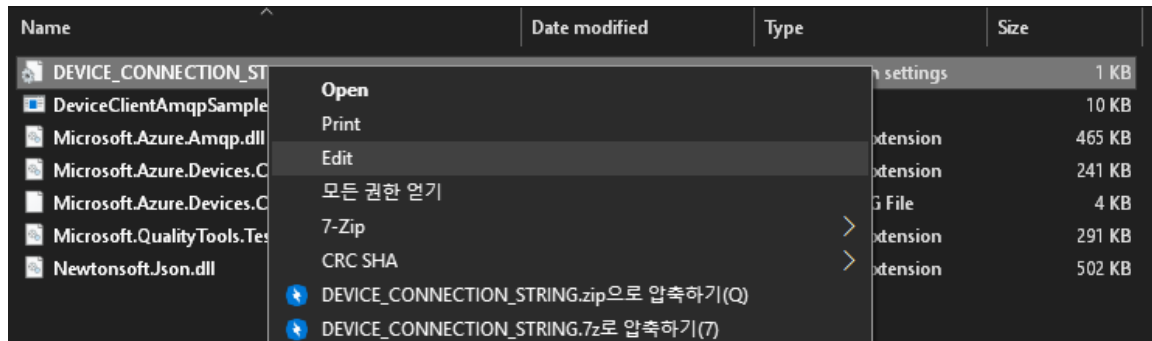


2.5. Management 탭을 클릭하면 생성한 IoT Device가 확인합니다.

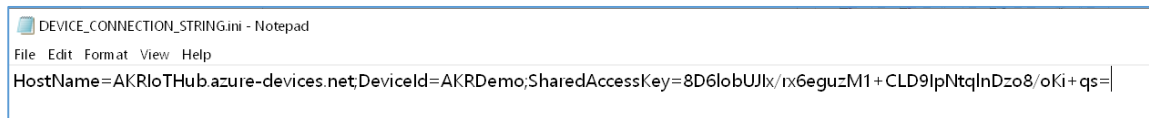
2.6. IoT Device에서 우측 마우스를 클릭하여 Copy Connection string for selected device를 선택합니다.



2.7. "2_DeviceClientAmqpSample"폴더 내의 DEVICE_CONNECTION_STRING.ini 파일을 편집하기를 선택합니다.

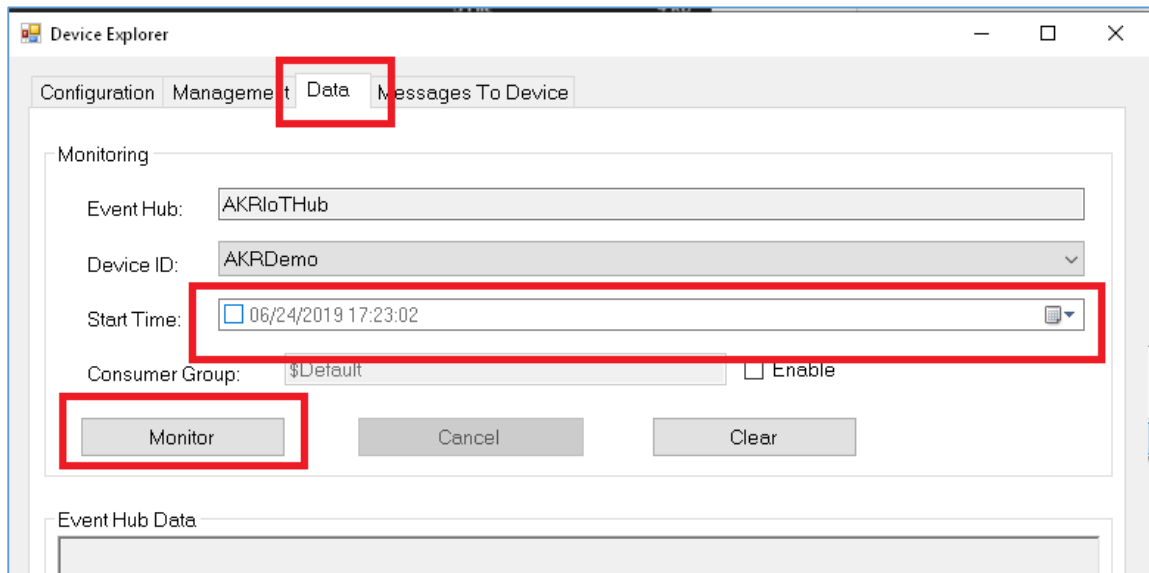


2.8. IoT Device의 복사한 연결 문자열을 "붙여넣기" 합니다.

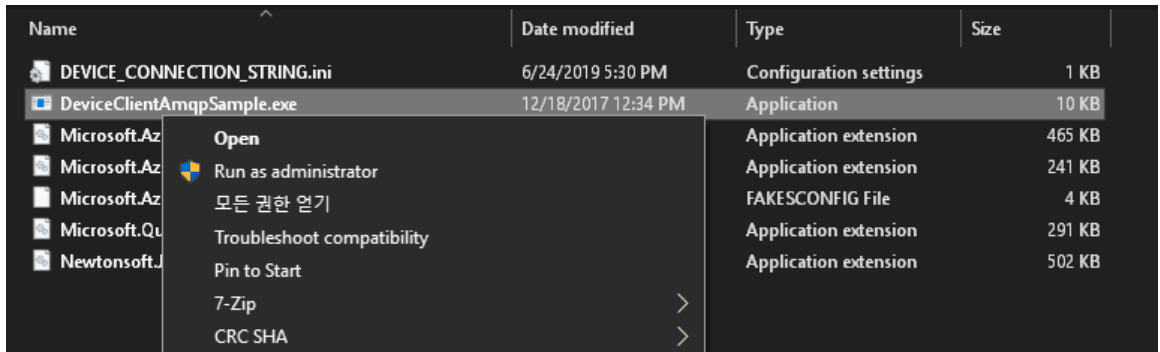


3. 시뮬레이션 데이터 통신 테스트

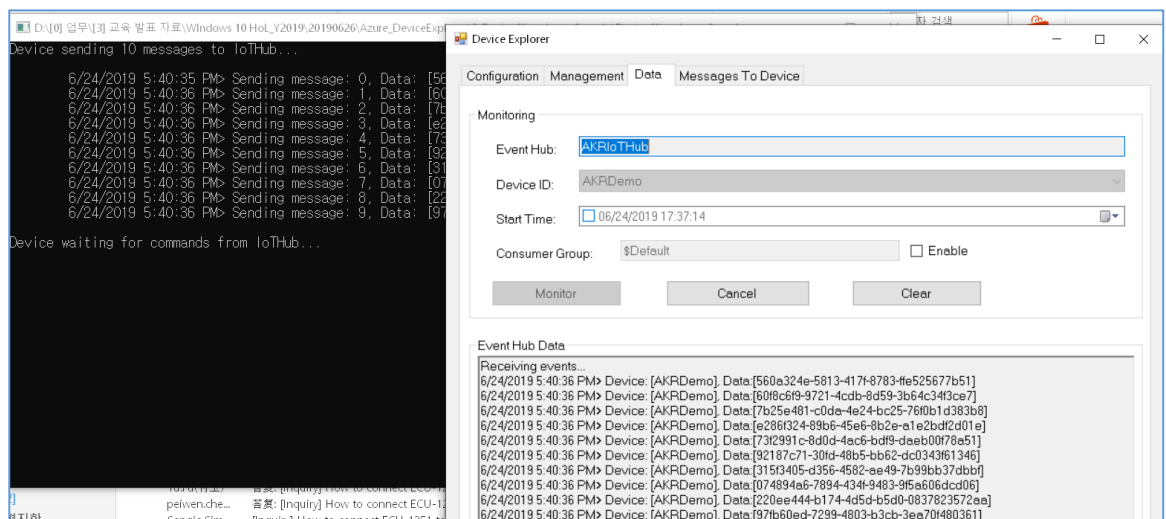
3.1. Device Explorer에 Data 탭을 열고 Monitor를 클릭합니다.



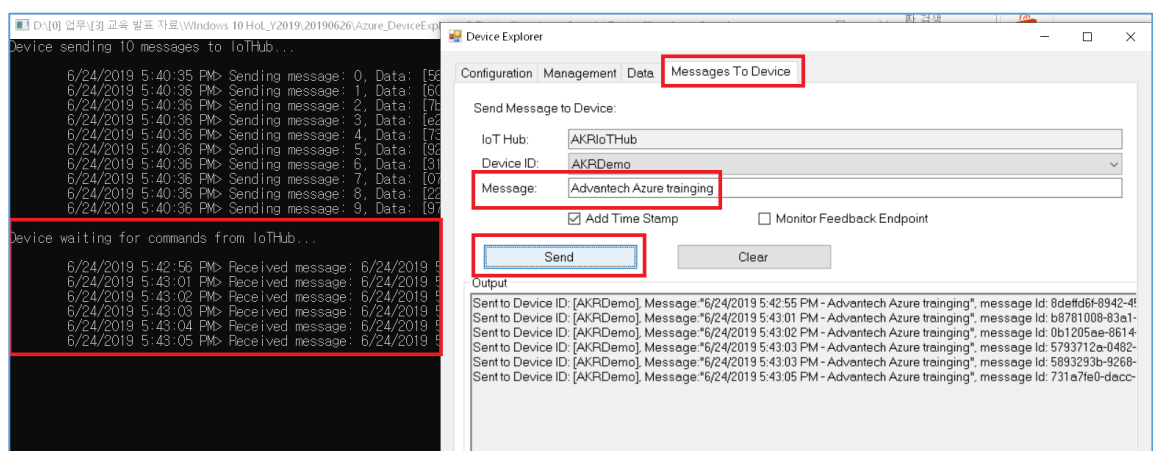
3.2. 2_DeviceClientAmqpSample폴더의 DeviceClientAmqpSample.exe파일을 실행합니다.



3.3. 2_DeviceClientAmqpSample를 실행하면 시뮬레이션 데이터가 Device에서 IoT Hub로 전송된 것을 확인할 수 있습니다.



3.4. Device Explorer에서 Messages To Device 탭을 선택합니다. Message를 편집하고 Send를 클릭합니다. IoT Hub에서 데이터가 Device로 전송된 것을 확인할 수 있습니다.



3.5. 위와 같이 통신이 되었다면 IoT Device와 IoT Hub간 통신이 정상적으로 되는 것을 확인할 수 있습니다.

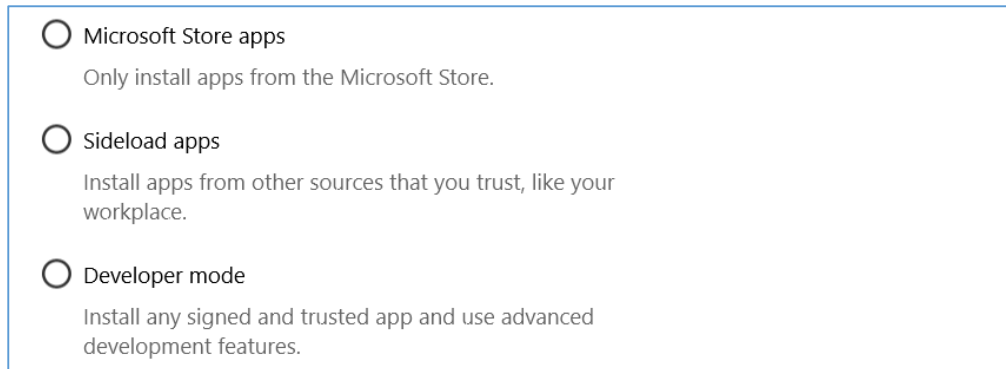
● Appendix : UWP 앱 배포

1. Visual Studio 설치

- Source Code Build version : **Windows 10 1803 / 17134.**

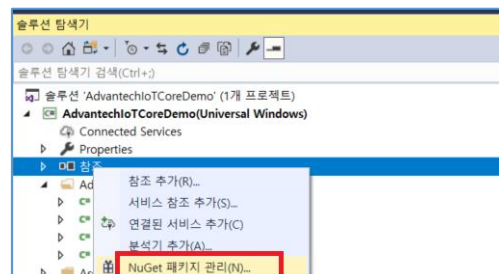
1.1. Windows 10 개발자 모드로 변경

- 설정 → Update & Security → For developers 에서 "Developer mode" 변경



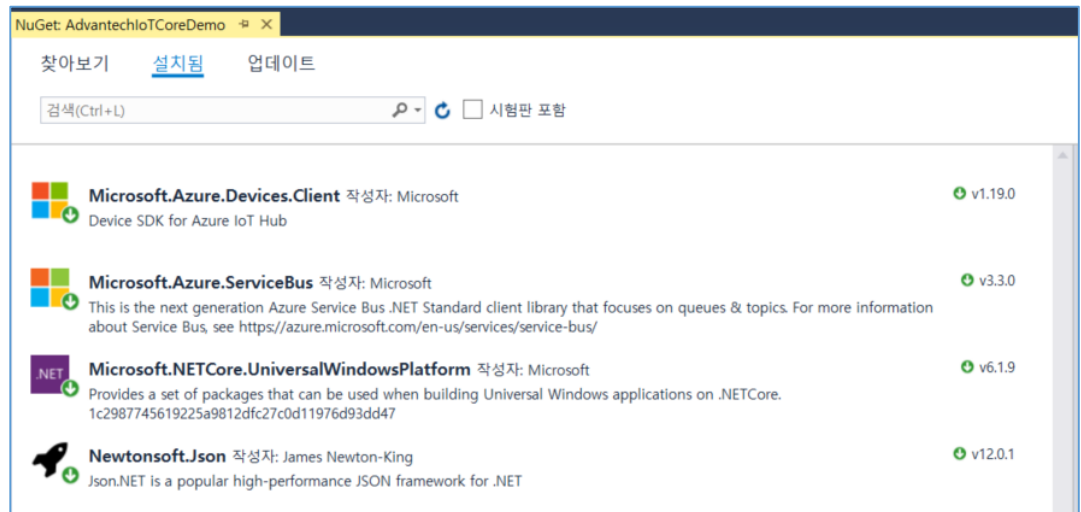
1.2. 사전 설치 - NuGet package

1.2.1. 솔루션 탐색기 → 참조 → NuGet 패키지 관리를 선택



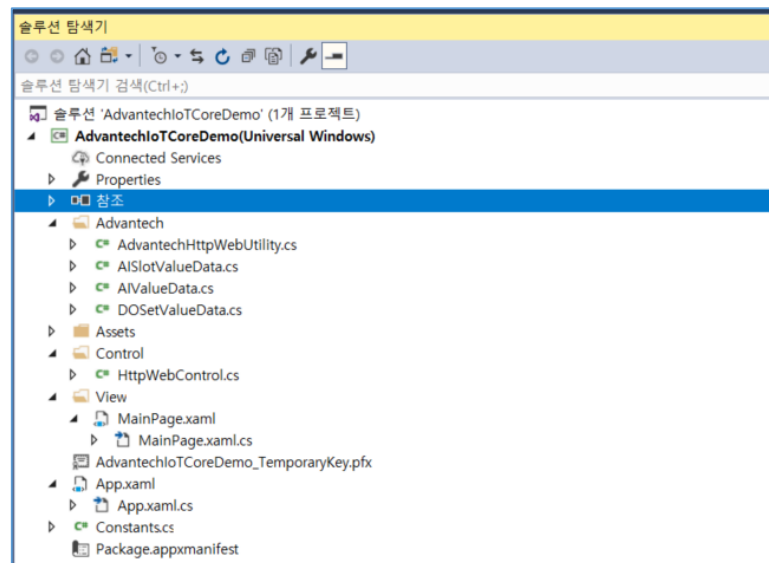
1.2.2. 찾아보기 → 아래의 4가지 패키지를 설치합니다.





1.3. 솔루션 구성

1.3.1. Source Code의 프로젝트



1.3.2. IoT Hub - Connection string

```
// Azure에 연결된 Device의 Connection String(Primary Key)
private readonly string connectionString = "HostName=ITHDemo.azure-
devices.net;DeviceId=sensor;SharedAccessKey=t6RgxMCirK3SCIPBLD7Rxcib0FWMNZ7AhrYvL6rIM68=";
```

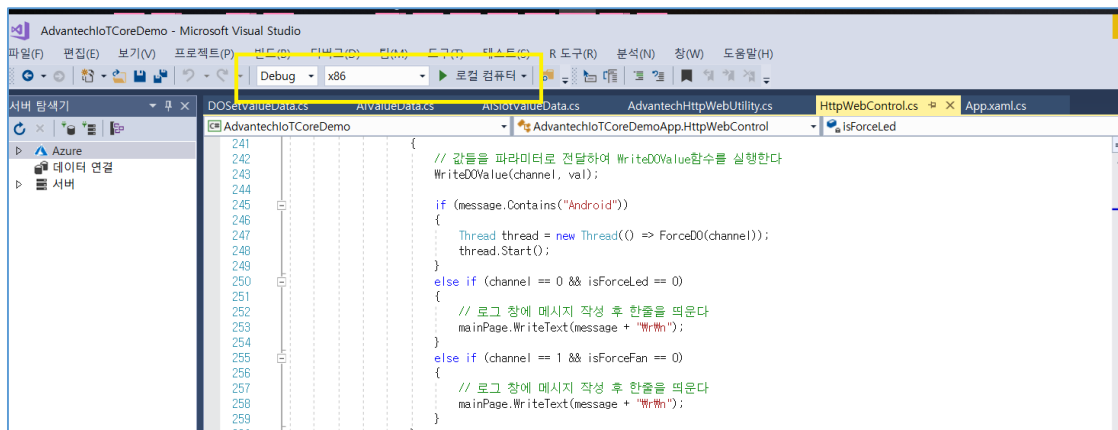
1.3.3. Service Bus – Connection string / Queue Name

```
// ServiceBus를 이용하기 위한 Connection String
const string ServiceBusConnectionString =
"Endpoint=sb://iottodevicebus.servicebus.windows.net/;SharedAccessKeyName=RootManage
SharedAccessKey;SharedAccessKey=Dh1fPeCYm02xmPNycTJWu6VJcaYjmhfyd97ocAfDBSs=";
const string QueueName = "androidgpiqueue";
```

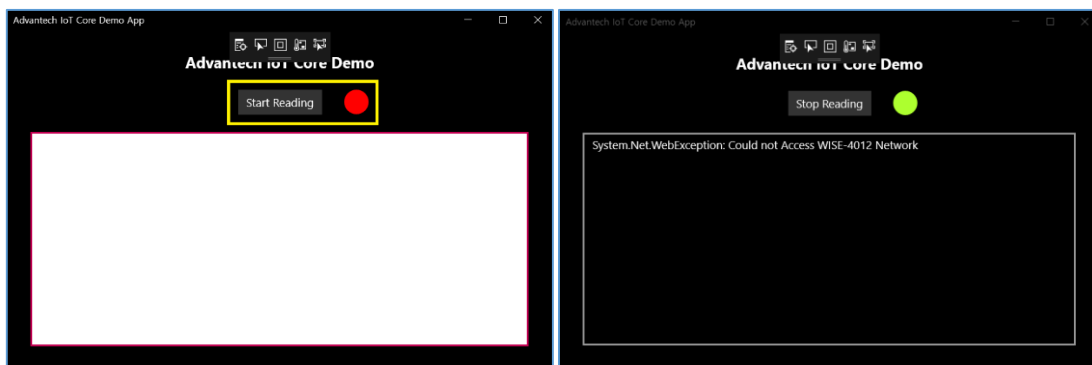
2. UWP 배포

2.1. Debug / x86 / 로컬 컴퓨터 확인 후 실행

(Debug/Release, x86/x64, 로컬 컴퓨터/원격 컴퓨터/Device)



2.2. 프로그램이 실행되면 Start Reading 옆의 단추를 클릭합니다.

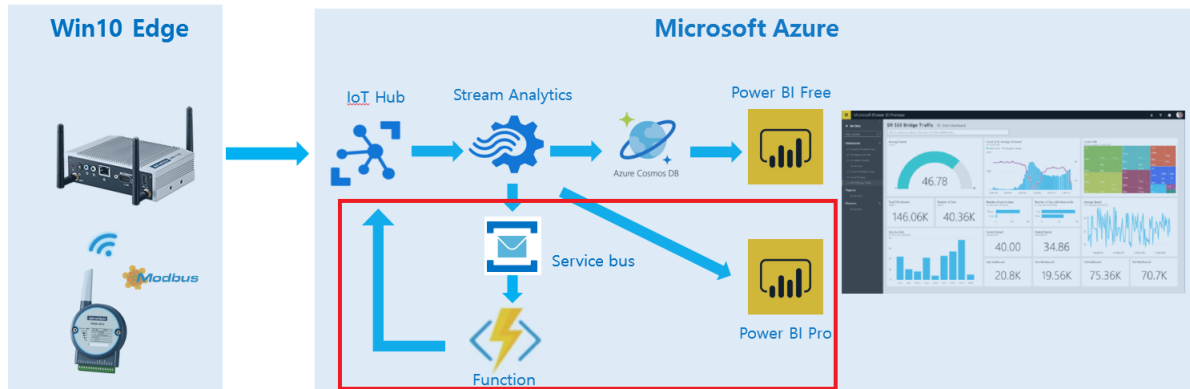


● Appendix

1. 센서의 제어

Power BI는 Dashboard 구성하여 센서의 데이터를 시각화하여 나타냅니다.

센서의 제어는 service bus와 Azure function을 이용하여 구성할 수 있습니다.



1.1. Azure Service bus

<https://docs.microsoft.com/ko-kr/azure/service-bus-messaging/>

1.2. Azure Function

Cloud에서 작은 코드 또는 "함수"를 실행하기 위한 솔루션입니다. Azure Functions는 개발 생산성을 높일 수 있으며 C#, F#, Node.js, Java, PHP 등 선택한 개발 언어를 사용할 수 있습니다. 코드를 실행한 시간에 따라 지불하고 Azure를 신뢰하여 필요에 따라 크기를 조정합니다. Azure Functions를 사용하면 Microsoft Azure에서 서버를 사용하지 않는 애플리케이션을 개발할 수 있습니다.

<https://docs.microsoft.com/ko-kr/azure/azure-functions/functions-overview>

2. Azure Learn

Azure에 대한 온라인 학습을 제공하며 샌드박스를 통해 무료 실습을 제공하고 있습니다.

<https://docs.microsoft.com/ko-kr/learn/azure/>