

Lab17. Azure IoT Hub 구현하기

1. 목적

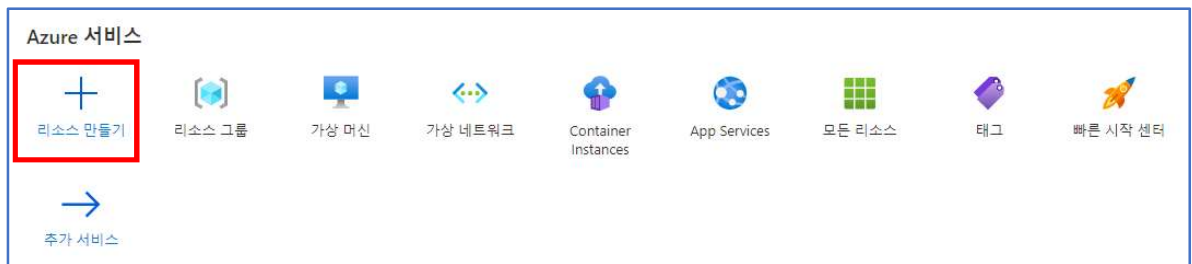
이번 실습에서는 Azure Portal에서 새 Azure IoT Hub를 구성하고 온라인 Raspberry Pi 디바이스 시뮬레이터를 사용하여 IoT 디바이스에 대한 연결을 인증한다. 센서 데이터 및 메시지는 Raspberry Pi 시뮬레이터에서 Azure IoT Hub로 전달되며 Azure Portal에서 메시징 활동에 대한 메트릭을 볼 수 있다.

2. 사전 준비물

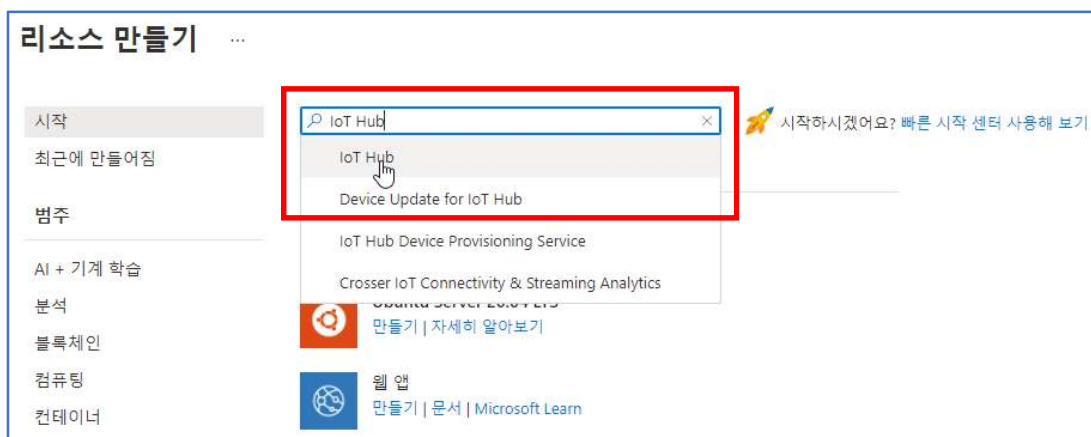
- Azure 체험 계정

3. IoT Hub 만들기

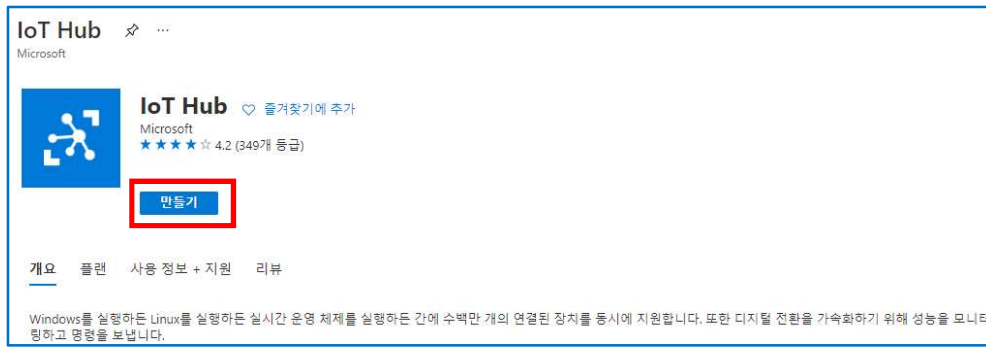
A. **Azure Portal**에 로그인한다. **[Azure 서비스]** 섹션에서 **[리소스 만들기]**를 클릭한다.



B. **[리소스 만들기]** 페이지로 들어왔다. 검색창에 **IoT Hub**를 입력하여 검색하고, 결과 목록에서 **[IoT Hub]**를 선택한다.



C. [IoT Hub] 페이지에서 [만들기]를 클릭한다.



D. [IoT 허브] 블레이드 [기본 사항]에서 다음의 각 값을 설정한다.

- ① 구독 : 현재 계정의 구독
- ② 리소스 그룹 : [새로 만들기] > myRGIoT
- ③ IoT Hub 이름 : my-hub-group2123(사용할 수 없다는 메시지가 나오면 마지막 숫자 4자리를 변경한다)
- ④ 지역 : 한국 중부

A screenshot of the IoT Hub '기본 사항' (Basic) configuration page. The page has a header with 'IoT 허브' and 'Microsoft'. Below the header, there are tabs for '기본 사항', '네트워킹', '관리', '태그', and '검토 + 만들기'. The '기본 사항' tab is selected. Below the tabs, there is a text block: 'IoT 허브를 만들면 수많은 IoT 자산을 연결, 모니터링 및 관리하는 데 도움이 됩니다. 자세한 정보'. Below this, there is a section titled '프로젝트 정보' with the text: '배포 및 비용을 관리하는 데 사용할 구독을 선택합니다. 폴더와 같은 리소스 그룹을 사용하면 리소스를 구성하고 관리하는 데 도움이 됩니다.' Below this, there are four input fields: '구독 * ①' with a dropdown menu showing 'MSDN 플랫폼 구독', '리소스 그룹 * ①' with a dropdown menu showing '(신규) myRGIoT' and a link '새로 만들기', 'IoT Hub 이름 * ①' with a text input field containing 'my-hub-group2123' and a green checkmark, and '지역 * ①' with a dropdown menu showing '한국 중부'. A red rectangular box highlights the '구독 * ①', '리소스 그룹 * ①', 'IoT Hub 이름 * ①', and '지역 * ①' fields.

- E. [관리] 탭으로 이동하여 [가격 및 크기 계층]을 S1:표준 계층으로 맞추고 [검토 + 만들기]를 클릭한다.

IoT 허브 ...
Microsoft

기본 사항 네트워크 **관리** 태그 검토 + 만들기

각 IoT 허브는 특정 계층에서 특정 장치 수로 프로비저닝됩니다. 계층과 장치 수에 따라 보낼 수 있는 메시지의 최대 일일 할당량이 결정됩니다. 자세한 정보

스케일링 계층 및 단위

가격 및 크기 계층 * ① S1: 표준 계층 [솔루션에 적합한 IoT 허브 계층을 선택하는 방법 알아보기](#)

S1 IoT 허브 장치 수 ① IoT 허브를 스케일링할 수 있는 방법을 결정합니다. 요구가 증가하는 경우 나중에 변경할 수 있습니다.

Defender for IoT ☒ 설정 Defender for IoT를 켜고 IoT Hub, IoT Edge 및 디바이스에 위협 보호 계층을 추가합니다. 자세한 정보

가격 및 크기 계층 ①	S1	디바이스-클라우드 메시지 ①	사용
일별 메시지 ①	400,000	메시지 라우팅 ①	사용
월별 비용	28116.25 KRW	클라우드 - 디바이스 명령 ①	사용
		디바이스-클라우드 명령 ①	사용
		IoT Edge ①	사용

검토 + 만들기 < 이전: 네트워크 다음: 태그 >

- F. 유효성 검사를 통과했습니다 메시지 확인 후 [만들기] 버튼을 클릭한다.

IoT 허브 ...
Microsoft

✔ 유효성 검사를 통과했습니다.

기본 사항 네트워크 관리 태그 **검토 + 만들기**

기본 사항

구독 MSDN 플랫폼 구독

리소스 그룹 myRGIoT

지역 한국 중부

IoT Hub 이름 my-hub-group2123

네트워크

연결 방법 공용 엔드포인트(모든 네트워크)

프라이빗 엔드포인트 연결 없음

관리

가격 및 크기 계층 S1

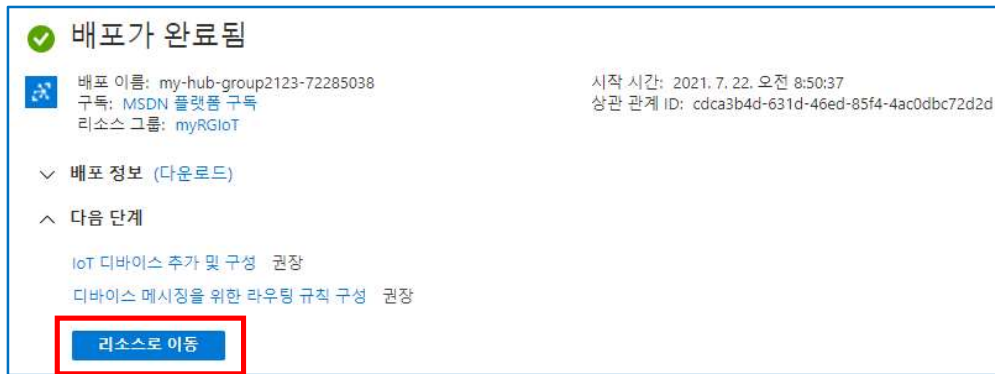
S1 IoT 허브 장치 수 1

일별 메시지 400,000

디바이스-클라우드 파티션 1

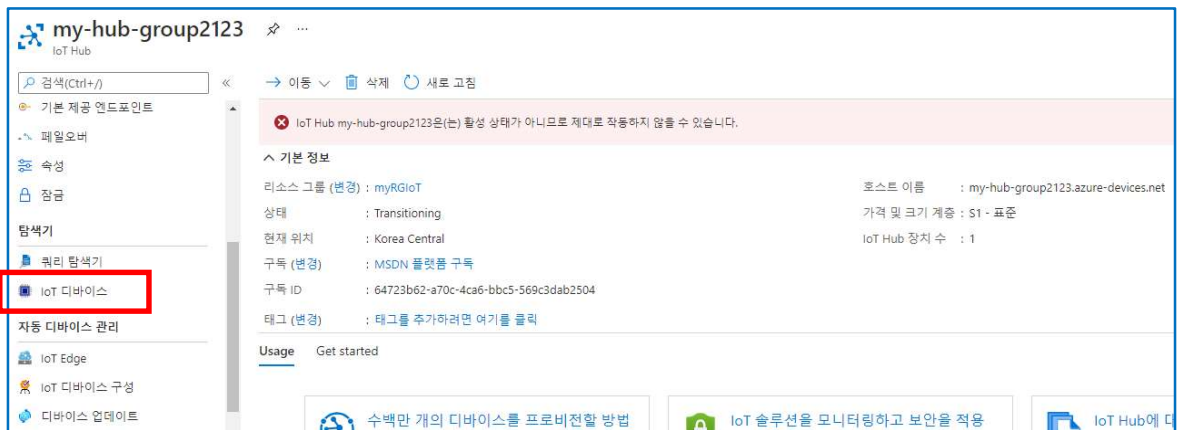
만들기 < 이전: 태그 다음 > [Automation 옵션](#)

G. [배포가 완료됨]을 확인하고, [리소스로 이동]을 클릭한다.

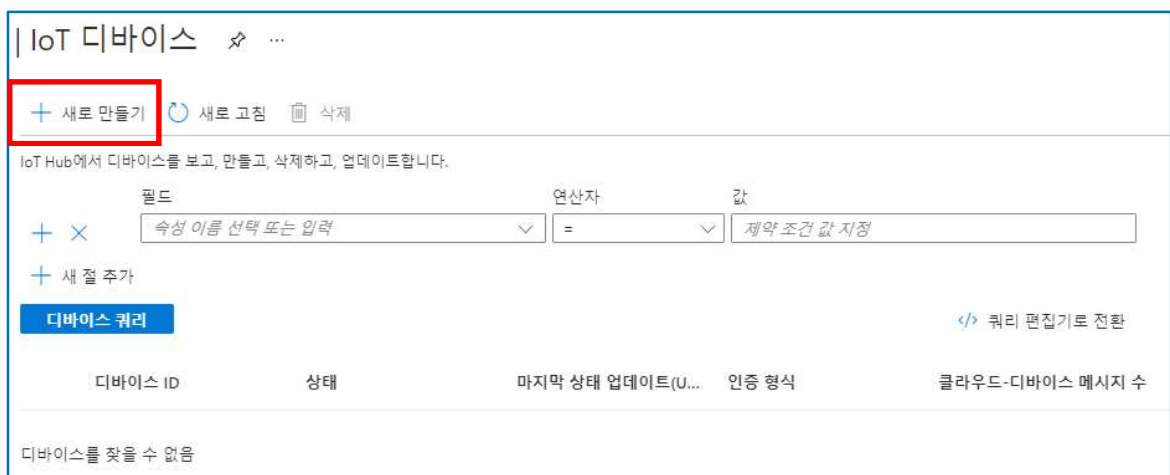


4. IoT 디바이스 추가

A. 새 디바이스를 추가하기 위해 좌측 서비스 메뉴에서 [IoT 디바이스]를 선택한다.



B. [+새로 만들기] 를 클릭한다.



- C. [디바이스 만들기] 페이지에서 [디바이스 ID]의 값을 **myRaspberryPi**라고 입력하고 나머지 설정들은 기본값 그대로 사용하기로 하고 [저장]을 클릭한다.

디바이스 만들기 ...

디바이스 카탈로그에서 IoT용 Azure Certified 디바이스 찾기

디바이스 ID * ①
myRaspberryPi

인증 형식 ①
대칭 키 X.509 자체 서명됨 X.509 CA 서명됨

기본 키 ①
기본 키 입력

보조 키 ①
보조 키 입력

키 자동 생성 ①
☒

저장

- D. 새로 IoT 디바이스가 추가된 것을 확인할 수 있다. 새로 추가된 **myRaspberryPi**를 선택한다.

IoT 디바이스 ✨ ...

+ 새로 만들기 새로 고침 삭제

IoT Hub에서 디바이스를 보고, 만들고, 삭제하고, 업데이트합니다.

필드 연산자 값
+ × 속성 이름 선택 또는 입력 = 제약 조건 값 지정

+ 새 질 추가

디바이스 쿼리 </> 쿼리 편집기로 전환

디바이스 ID	상태	마지막 상태 업데이트(U...	인증 형식	클라우...
myRaspberryPi	Enabled	--	Sas	0

- E. [myRaspberryPi] 페이지에서 [기본 연결 문자열]의 값을 복사한다.

myRaspberryPi ✨ ...

my-hub-group2123

저장 디바이스에 메시지 전송 직접 메서드 모듈 ID 추가 디바이스 쌍 키 관리 새로 고침

디바이스 ID ①
myRaspberryPi

기본 키 ①
.....

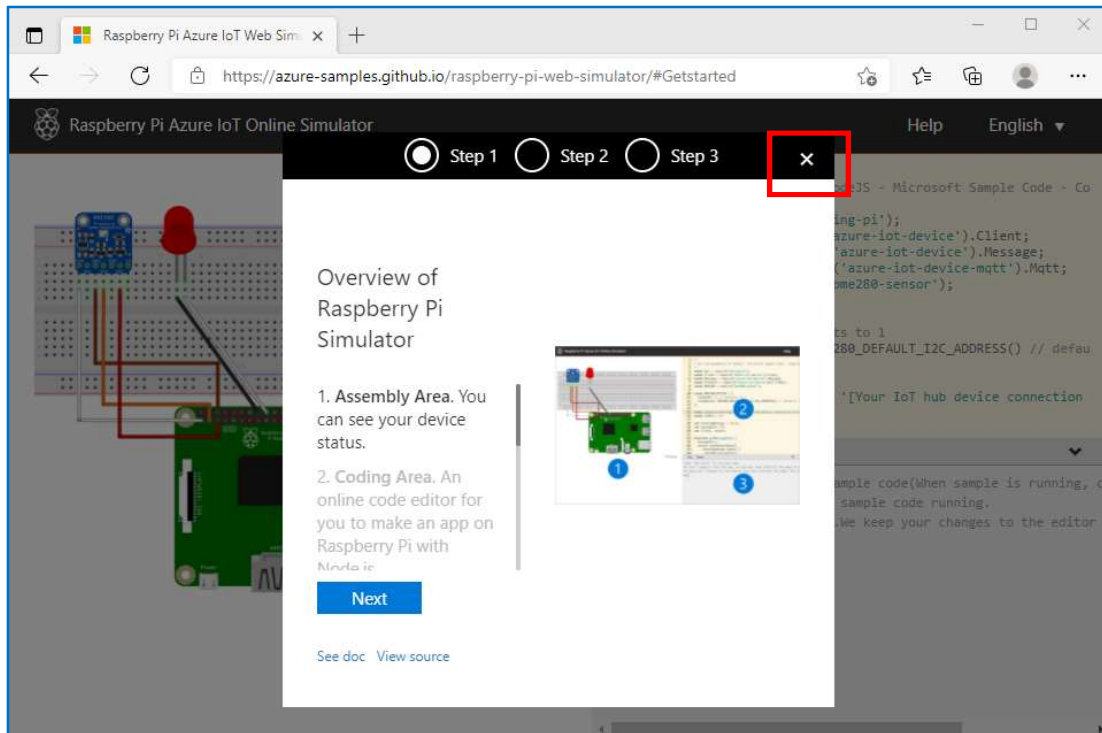
보조 키 ①
.....

기본 연결 문자열 ①
.....

보조 연결 문자열 ①
.....

5. Raspberry Pi 시뮬레이터를 사용하여 디바이스 테스트

- A. 웹 브라우저를 열고 [온라인 Raspberry Pi 시뮬레이터](https://azure-samples.github.io/raspberry-pi-web-simulator/#Getstarted)로 이동한다. 팝업창의 우측 상단의 [x]를 클릭하여 창을 닫는다.




- B. 페이지 우측의 코드 영역에서 **connectionString**을 찾는다.

```
1  /*
2  * IoT Hub Raspberry Pi NodeJS - Microsoft Sample Code - Copyright (c
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 0x7
13 };
14
15 const connectionString = '[Your IoT hub device connection string]';
16 const LEDPin = 4;
17
```

- C. **connectionString**의 값을 위 4-E에서 복사한 **[기본 연결 문자열]**의 값으로 바꾼다. 이 문자열에는 DeviceId와 SharedAccessKey 값이 포함되어 있다.

```
const connectionString =  
  
'HostName=my-hub-group2123.azure-  
devices.net;DeviceId=myRaspberryPi;SharedAccessKey=VRoc99A2AzWJ+7AKHEVu3662svy1KPiQ4K  
mRmkGYf/k=';
```

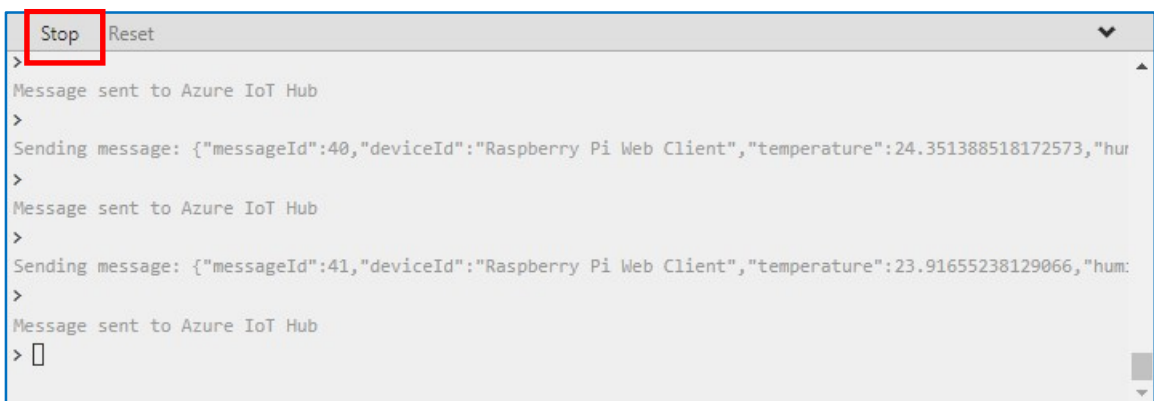
- D. 실행 영역의 **[Run]**을 클릭하여 애플리케이션을 실행한다. 콘솔 출력에 Raspberry Pi 시뮬레이터가 Azure IoT Hub로 보내는 센서 데이터와 메시지가 표시되어야 한다. 데이터와 메시지는 라즈베리 파이 시뮬레이터 LED가 깜박일 때마다 전송된다.



```
14  
15 const connectionString = 'HostName=my-hub-group2123.azure-devices.net;DeviceId=myRaspberryPi;Shared  
16 const LEDPin = 4;  
17  
18 var sendingMessage = false;  
19 var messageId = 0;  
20 var client, sensor;  
21 var blinkLEDTIMEOUT = null;  
22  
23  
24  
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.
> □

- E. 실행 중일 때에는 **[Run]**이 **[Stop]**으로 변경된다. **[Stop]**을 클릭하여 데이터 전송을 중지한다.



```
Stop Reset
```

>
Message sent to Azure IoT Hub
>
Sending message: {"messageId":40,"deviceId":"Raspberry Pi Web Client","temperature":24.351388518172573,"hum:
>
Message sent to Azure IoT Hub
>
Sending message: {"messageId":41,"deviceId":"Raspberry Pi Web Client","temperature":23.91655238129066,"hum:
>
Message sent to Azure IoT Hub
> □

- F. Azure Portal의 **my-hub-groupxxxx** 블레이드로 돌아와서 **[개요]** 페이지의 작업창을 아래로 스크롤하여 **IoT Hub 사용량** 정보를 확인한다.

