

사물인터넷 디바이스 오픈소스 플랫폼 (Mbed OS) 프로그래밍

4. Cloud / Web application with Mbed



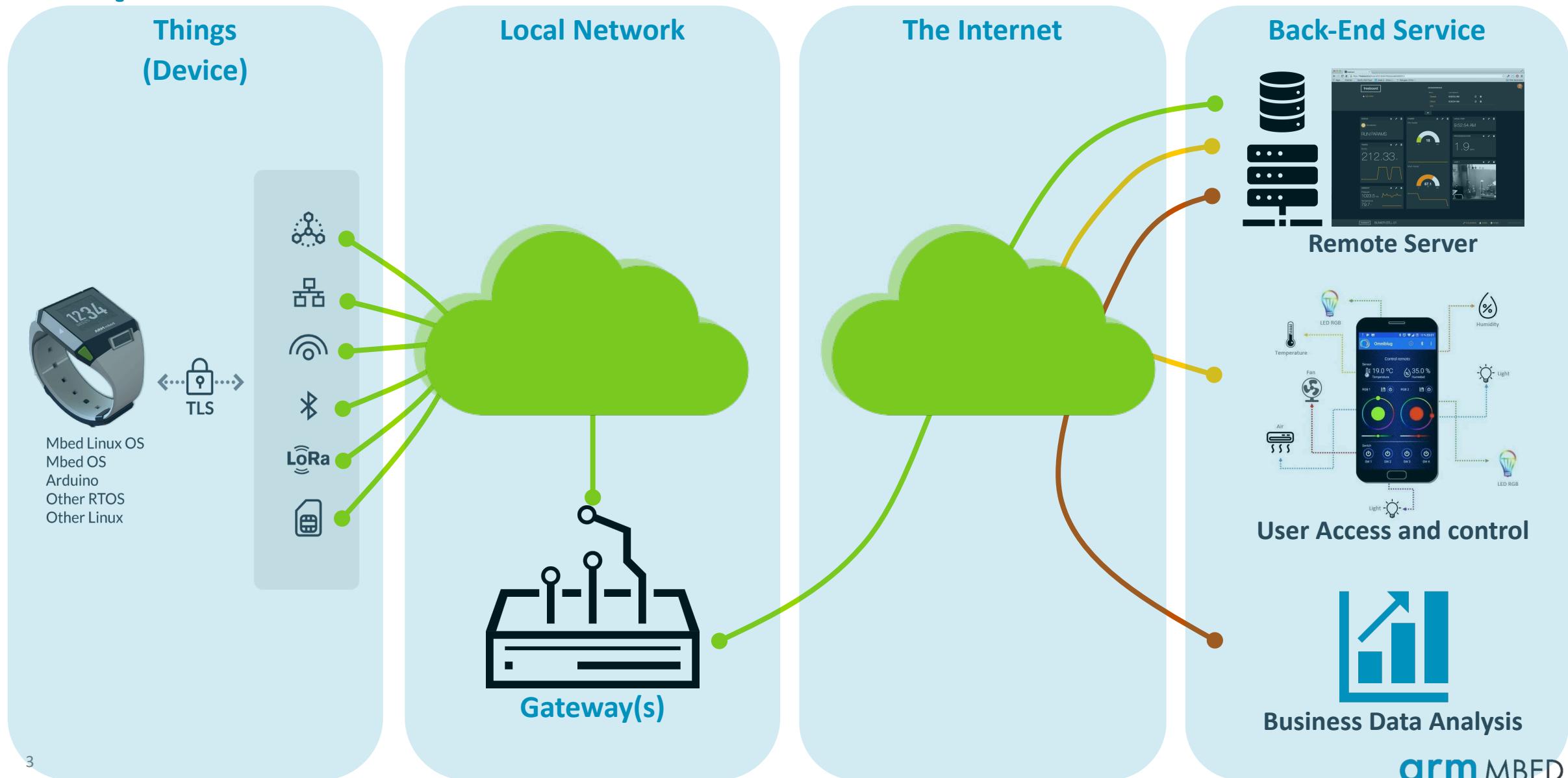
이동명
2020.07



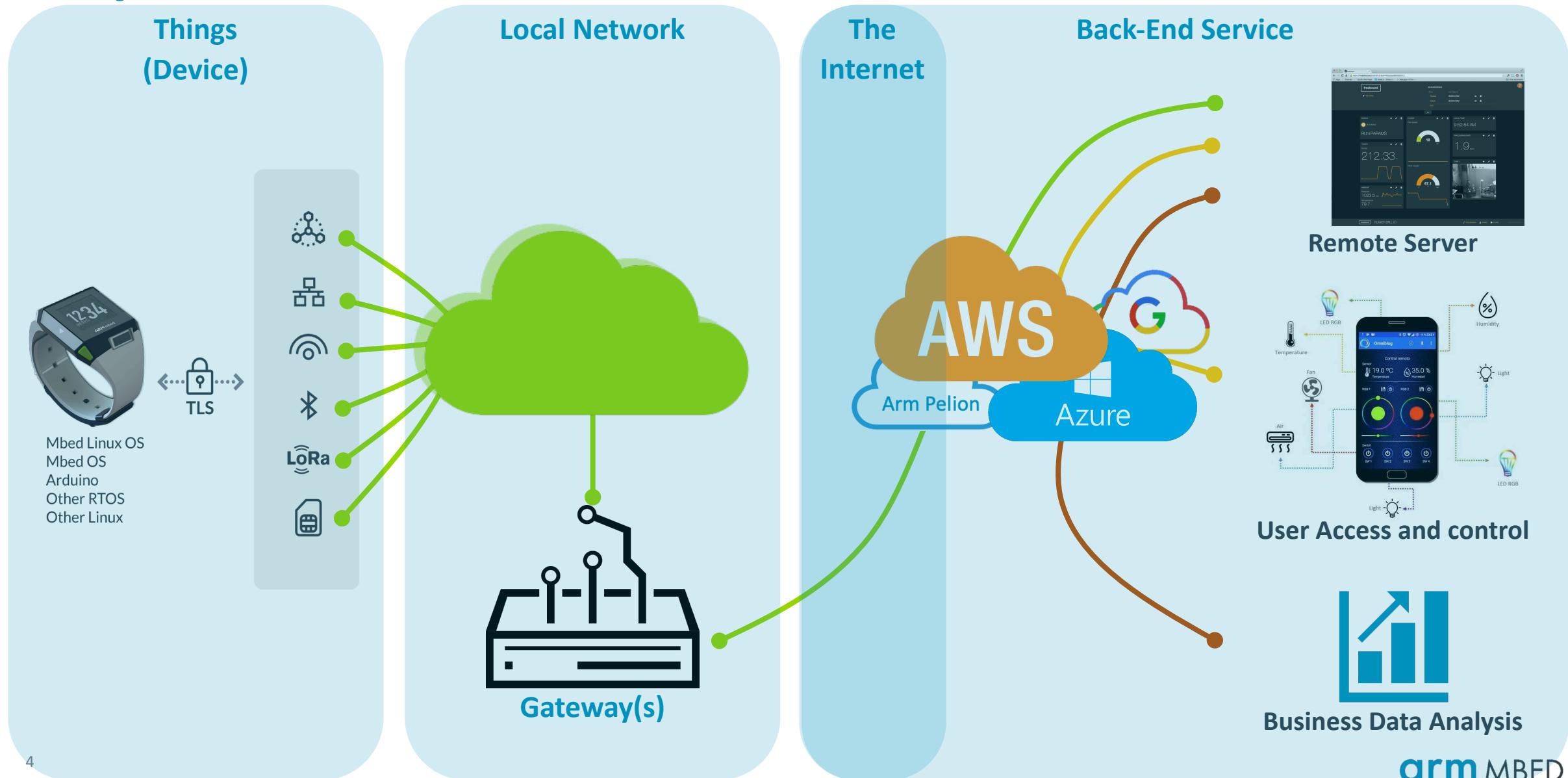
Agenda

1. Mbed Ecosystem
2. Setup develop environment
3. Peripheral IPs control practice
- 4. Cloud / Web application with Mbed**

IoT platform



IoT platform



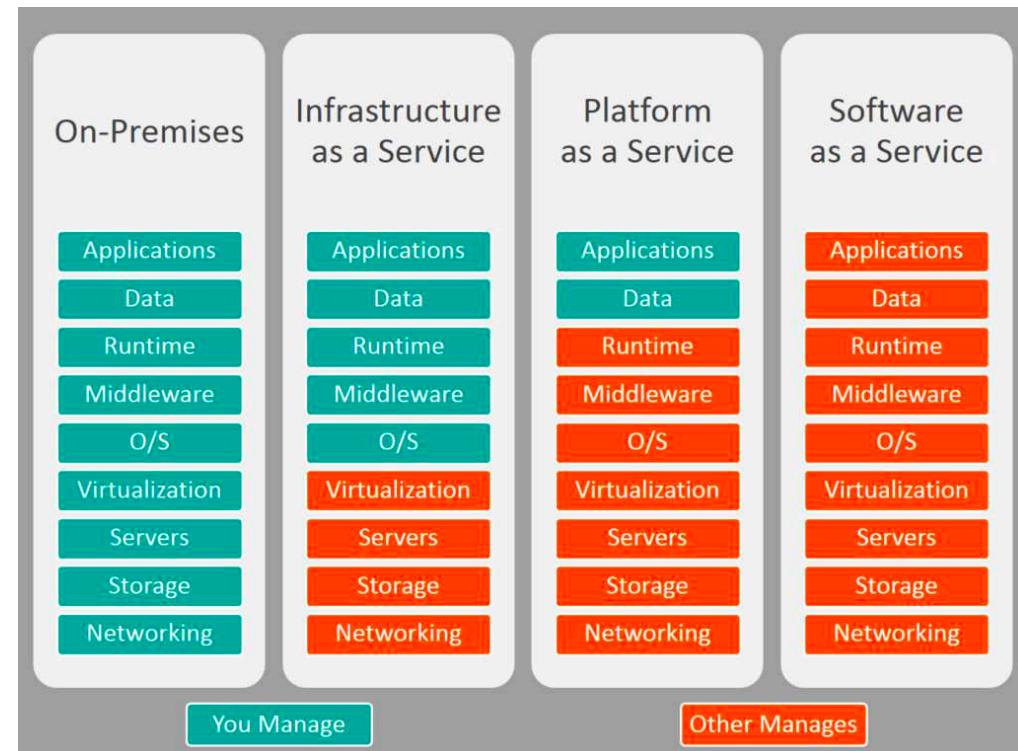
세계 클라우드 서비스 시장규모



세계 클라우드 서비스 시장 규모

	2018년 지출 (억달러)	2018년 시장점유율	2019년 지출 (억달러)	2019년 시장점유율	전년대비 성장을
아마존웹서비스(AWS)	254	32.7%	346	32.3%	36.0%
마이크로소프트 애저	110	14.2%	181	16.9%	63.9%
구글 클라우드	33	4.2%	62	5.8%	87.8%
알리바바 클라우드	32	4.1%	52	4.9%	63.8%
기타	349	44.8%	43	40.1%	23.3%
합계	778	100%	1071	100%	37.6%

자료=카날리스



각 클라우드의 연결 지원

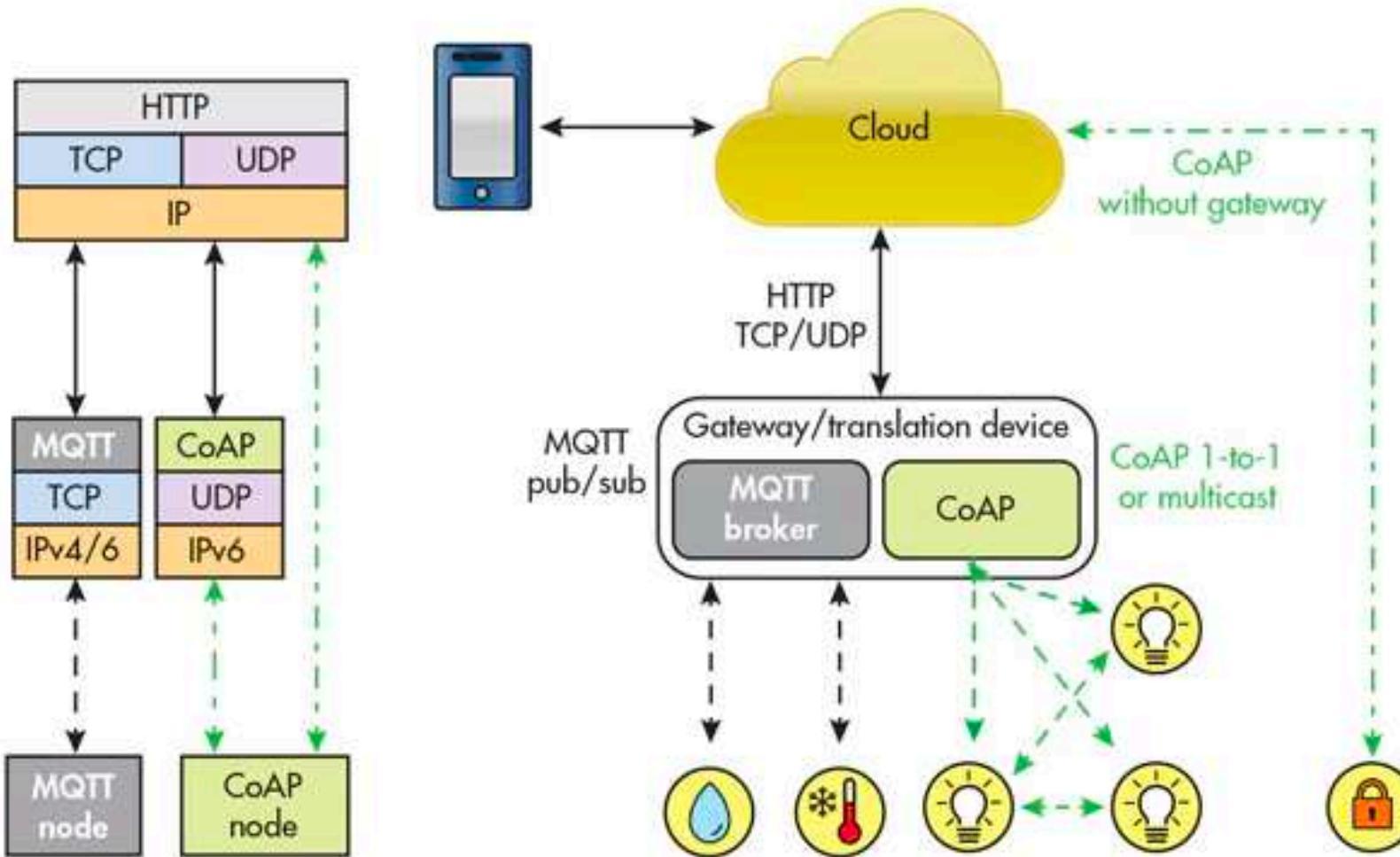
- Google cloud -
 - <https://os.mbed.com/users/coisme/notebook/google-cloud-iot-from-mbed-os-device/>
- IBM Watson -
 - <https://os.mbed.com/users/coisme/notebook/ibm-watson-iot-from-mbed-os-device/>
- Microsoft Azure -
 - <https://os.mbed.com/users/coisme/notebook/azure-iot-hub-from-mbed-os-device/>
- AWS -
 - <https://os.mbed.com/teams/mbed-os-examples/code/mbed-os-example-aws/>
 - <https://os.mbed.com/users/coisme/notebook/aws-iot-from-mbed-os-device/>
- Arm Pelion, 100 개의 Device 를 무료로 기간제한없이 사용할 수 있음.
 - <https://os.mbed.com/guides/connect-device-to-pelion/>

* Mbed OS -> Cloud 연결의 예제는 공개되어 있으나, FOTA(Firmware Over The Air) 와 같은 클라우드 종속적인 기능의 경우는 아직까지 Pelion 에 최적화 되어 있다.



Communication protocol for IoT Devices

IoT Device 를 위한 프로토콜



- OneM2M
 - <https://os.mbed.com/teams/OneM2M/>
- LwM2M
 - Default
- HTTP(over IPV6) / HTTPS
 - <https://os.mbed.com/teams/sandbox/code/http-example/>
- MQTT
 - <https://github.com/ARMmbed/mbed-mqtt>
- CoAP
 - <https://os.mbed.com/teams/sandbox/code/coap-example/>

IoT Device 를 위한 프로토콜

Beyond MQTT: A Cisco View on IoT Protocols, Paul Duffy, April 30 2013				
Protocol	CoAP	XMPP	RESTful HTTP	MQTT
Transport	UDP	TCP	TCP	TCP
Messaging	Request/Response Request/Response	Publish/Subscribe Request/Response	Request/Response	Publish/Subscribe Request/Response
2G, 3G, 4G Suitability (1000s nodes)	Excellent	Excellent	Excellent	Excellent
LLN Suitability (1000s nodes)	Excellent	Fair	Fair	Fair
Compute Resources	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash
Success Stories	Utility Field Area Networks	Remote management of consumer white goods	Smart Energy Profile 2 (premise energy management, home services)	Extending enterprise messaging into IoT applications

CoAP

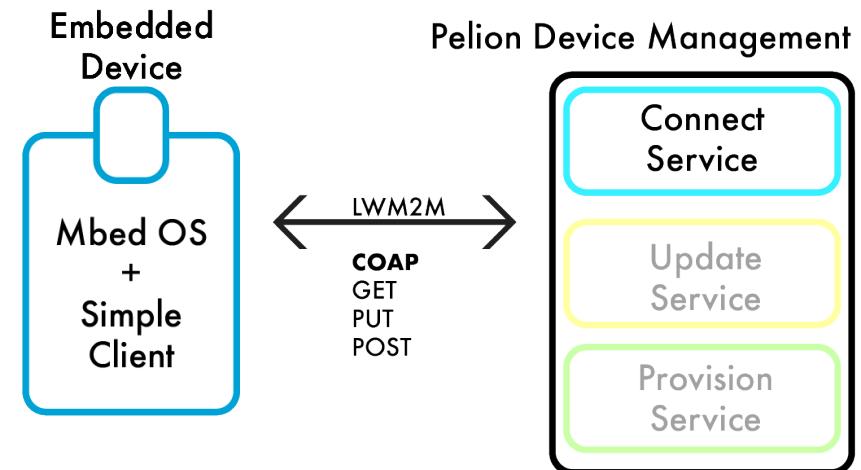
웹 프로토콜은 IoT 장치에 사용 가능하고 사용할 수 있지만 대부분의 IoT 응용 프로그램에는 너무 무겁습니다. [제한된 애플리케이션 프로토콜 \(CoAP\)의 저전력 제약 네트워크와 사용하기 위해 IETF에 의해 디자인되었다.](#)

CoAP는 RESTful 프로토콜입니다. HTTP와 의미 적으로 정렬되며 HTTP와 일대일로 매핑됩니다. CoAP는 배터리 또는 에너지 하베스팅에서 작동하는 장치에 적합한 프로토콜.

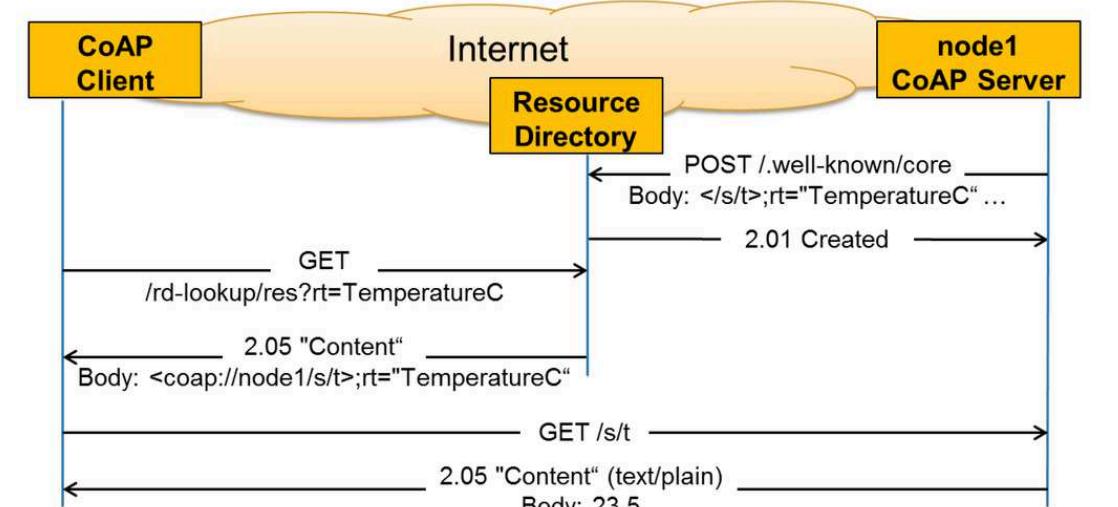
- CoAP는 UDP를 사용.
- CoAP는 UDP를 사용하기 때문에 일부 TCP 기능은 CoAP에서 재생산됨. 예를 들어 CoAP는 확인 가능한 메시지와 확인 불가능한 메시지를 구분함.
- 요청과 응답은 CoAP 메시지를 통해 비동기 적으로 교환.
- 모든 헤더, 메소드 및 상태 코드는 이진 인코딩 됨으로 프로토콜 오버 헤드가 줄어 들어듬.
- HTTP와 달리 CoAP 응답을 캐시하는 기능은 응답 코드에 따라 다름.
- CoAP는 초경량 프로토콜의 필요성과 영구적 인 연결 기능을 잘 해결할 수 있고 웹 어플리케이션 개발 경험이 있다면 CoAP 사용이 비교적 쉽습니다.(RESTful 기반)

IoT Device 를 위한 프로토콜 - COAP Options

- GET
 - Read value from device
 - Must be marked as 'observable'
- PUT
 - Set value on device
- POST
 - Send data buffer to device
 - Trigger callback function on device
- DELETE
 - Not currently in use



Use case example of COAP & LWM2M



Use case example of COAP **arm MBED**

IoT Device 를 위한 프로토콜

Beyond MQTT: A Cisco View on IoT Protocols, Paul Duffy, April 30 2013				
Protocol	CoAP	XMPP	RESTful HTTP	MQTT
Transport	UDP	TCP	TCP	TCP
Messaging	Request/Response	Publish/Subscribe Request/Response	Request/Response	Publish/Subscribe Request/Response
2G, 3G, 4G Suitability (1000s nodes)	Excellent	Excellent	Excellent	Excellent
LLN Suitability (1000s nodes)	Excellent	Fair	Fair	Fair
Compute Resources	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash
Success Stories	Utility Field Area Networks	Remote management of consumer white goods	Smart Energy Profile 2 (premise energy management, home services)	Extending enterprise messaging into IoT applications

XMPP

[XMPP \(Extensible Messaging and Presence Protocol\)](#) 는 IoT 공간에서 새로운 용도를 찾는 기존 웹 기술중 하나.

세탁기, 건조기, 냉장고 등과 같은 가전 제품의 대규모 관리를 위한 프로토콜로 인스턴트 메시징 및 현재 상태 정보에 뿌리를 두고 있음. VoIP, 협업, 경량 미들웨어, 컨텐츠 신디케이션 및 일반화 된 XML 데이터 라우팅을 위한 시그널링으로 확장되고 있음.

IoT Device 를 위한 프로토콜

Beyond MQTT: A Cisco View on IoT Protocols, Paul Duffy, April 30 2013				
Protocol	CoAP	XMPP	RESTful HTTP	MQTT
Transport	UDP	TCP	TCP	TCP
Messaging	Request/Response Request/Response	Publish/Subscribe Request/Response	Request/Response	Publish/Subscribe Request/Response
2G, 3G, 4G Suitability (1000s nodes)	Excellent	Excellent	Excellent	Excellent
LLN Suitability (1000s nodes)	Excellent	Fair	Fair	Fair
Compute Resources	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash
Success Stories	Utility Field Area Networks	Remote management of consumer white goods	Smart Energy Profile 2 (premise energy management, home services)	Extending enterprise messaging into IoT applications

HTTP

HTTP는 웹에 사용되는 클라이언트-서버 모델의 기초입니다. HTTP를 구현하는 것보다 안전한 방법은 서버가 아닌 IoT 장치에 클라이언트 만 포함시키는 것입니다. 즉, 수신 만하지 않고 연결만 시작할 수 있는 IoT 장치를 구축하는 것이 더 안전(로컬 네트워크에 대한 외부 액세스를 허용하지 않음) 합니다.

Web Socket

Web Socket 은 클라이언트와 서버 간의 단일 TCP 연결을 통한 전 이중 통신을 제공하는 프로토콜입니다. HTML 5 사양의 일부로, Web Socket 표준은 양방향 웹 통신 및 연결 관리와 관련된 많은 복잡성을 단순화합니다.

IoT Device 를 위한 프로토콜

Beyond MQTT: A Cisco View on IoT Protocols, Paul Duffy, April 30 2013				
Protocol	CoAP	XMPP	RESTful HTTP	MQTT
Transport	UDP	TCP	TCP	TCP
Messaging	Request/Response	Publish/Subscribe Request/Response	Request/Response	Publish/Subscribe Request/Response
2G, 3G, 4G Suitability (1000s nodes)	Excellent	Excellent	Excellent	Excellent
LLN Suitability (1000s nodes)	Excellent	Fair	Fair	Fair
Compute Resources	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash
Success Stories	Utility Field Area Networks	Remote management of consumer white goods	Smart Energy Profile 2 (premise energy management, home services)	Extending enterprise messaging into IoT applications

MQTT

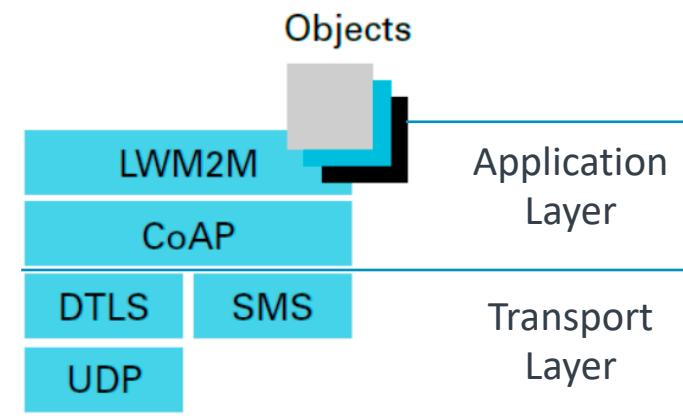
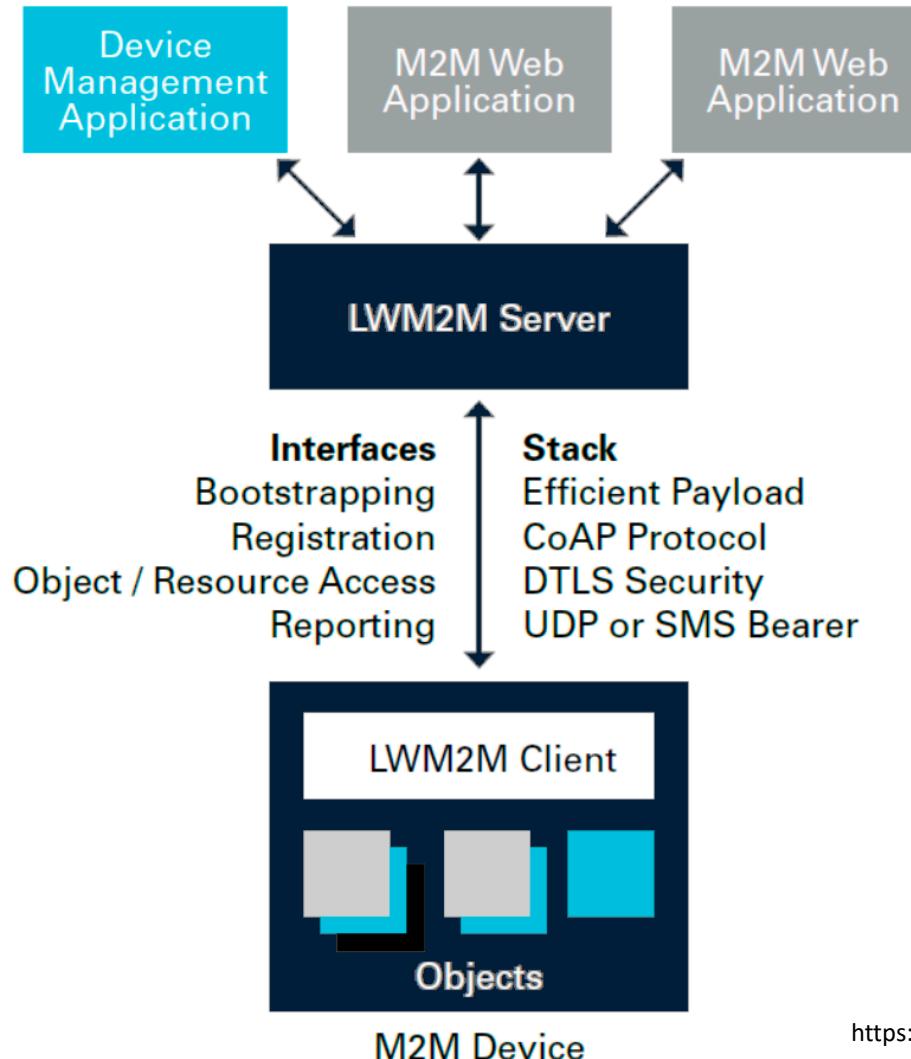
[MQTT \(MQ Telemetry Transport\)](#) 는 제한된 장치 및 저 대역폭, 대기 시간이 긴 네트워크를 위한 오픈 소스 프로토콜입니다. 이것은 초경량이며 소규모 장치를 제한된 네트워크에 연결하는 데 이상적인 게시 / 구독 메시징 전송입니다.

MQTT는 대역폭 효율적이고 데이터에 구애받지 않으며 지속적으로 세션을 인식합니다. IoT 장치의 리소스 요구 사항을 최소화하는 동시에 등급 서비스를 통해 안정성과 어느 정도의 배달 보장을 시도합니다.

MQTT는 인터넷의 Backend 서버에서 모니터하거나 제어해야하는 소형 장치의 대규모 네트워크를 대상으로 하지만, 장치 간 전송을 위해 설계되지 않았습니다. 또한 데이터를 여러 수신자에게 ‘Multicast’ 하지 않습니다. MQTT는 매우 간단하여 제어 옵션이 거의 없습니다.

IoT Device 를 위한 프로토콜 - LWM2M

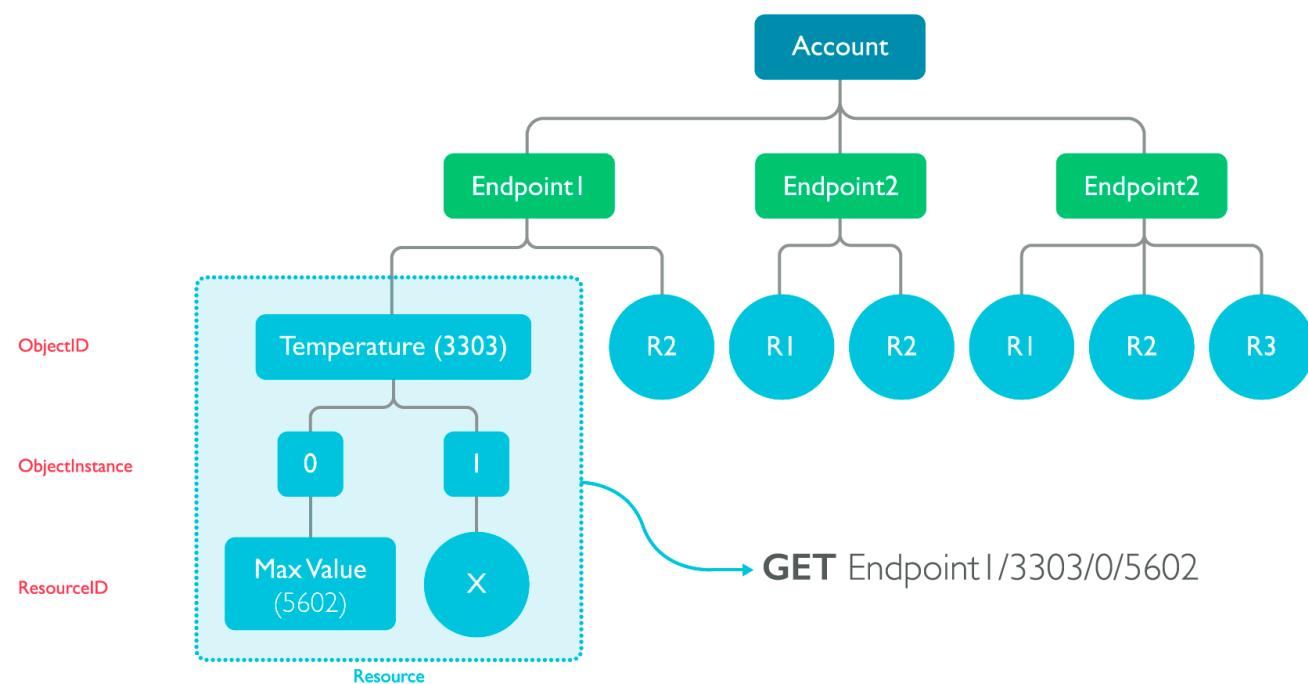
LWM2M Resource Model – 3 levels



- OneM2M
 - <https://os.mbed.com/teams/OneM2M/>
- LWM2M
 - Default
- HTTP(over IPV6) / HTTPS
 - <https://os.mbed.com/teams/sandbox/code/http-example/>
- MQTT
 - <https://github.com/ARMmbed/mbed-mqtt>
- CoAP
 - <https://os.mbed.com/teams/sandbox/code/coap-example/>

IoT Device 를 위한 프로토콜 - LWM2M

LWM2M Resource Model – 3 levels



•OMA LWM2M Management Object

<https://devtoolkit.openmobilealliance.org/OEditor/Default>

	Object name	ObjectID	ResourceId
초음파센서	Distance	3330	5700
온도센서	Temperature Sensor	3303	5700
습도센서	Humidity Sensor	3304	5700

{ObjectID} / {ObjectInstance} / {ResourceID}

ex) temp sensor object

/temp/0/maxValue == /3303/0/5602

/temp/0/minValue. == /3303/0/5601

/temp/0/sensorValue == /3303/0/5700

```
// POST resource 3201/0/5850
m2m_post_res = M2MInterfaceFactory::create_resource(m2m_obj_list, 3201, 0, 5850, M2MResourceInstance::INTEGER, M2MBase::POST_ALLOWED);
if (m2m_post_res->set_execute_function(execute_post) != true) {
    printf("m2m_post_res->set_execute_function() failed\n");
    return -1;
}
```

IoT Device 를 위한 프로토콜 - LWM2M vs MQTT

	LwM2M	MQTT
Transport	UDP, SMS, TCP, NIDD, LoRaWAN	TCP
Application Layer	CoAP	-
Payload	TLV, JSON, Opaque, CBOR	Undefined
Bandwidth Consumption	Low	Depends on payload encoding
Data model	Defined	Undefined
IPv6	Yes	Yes
Security	TLS, DTLS 1.2+	SSL/TSL
Standardization body	OMA, IETF	OASIS, ISO



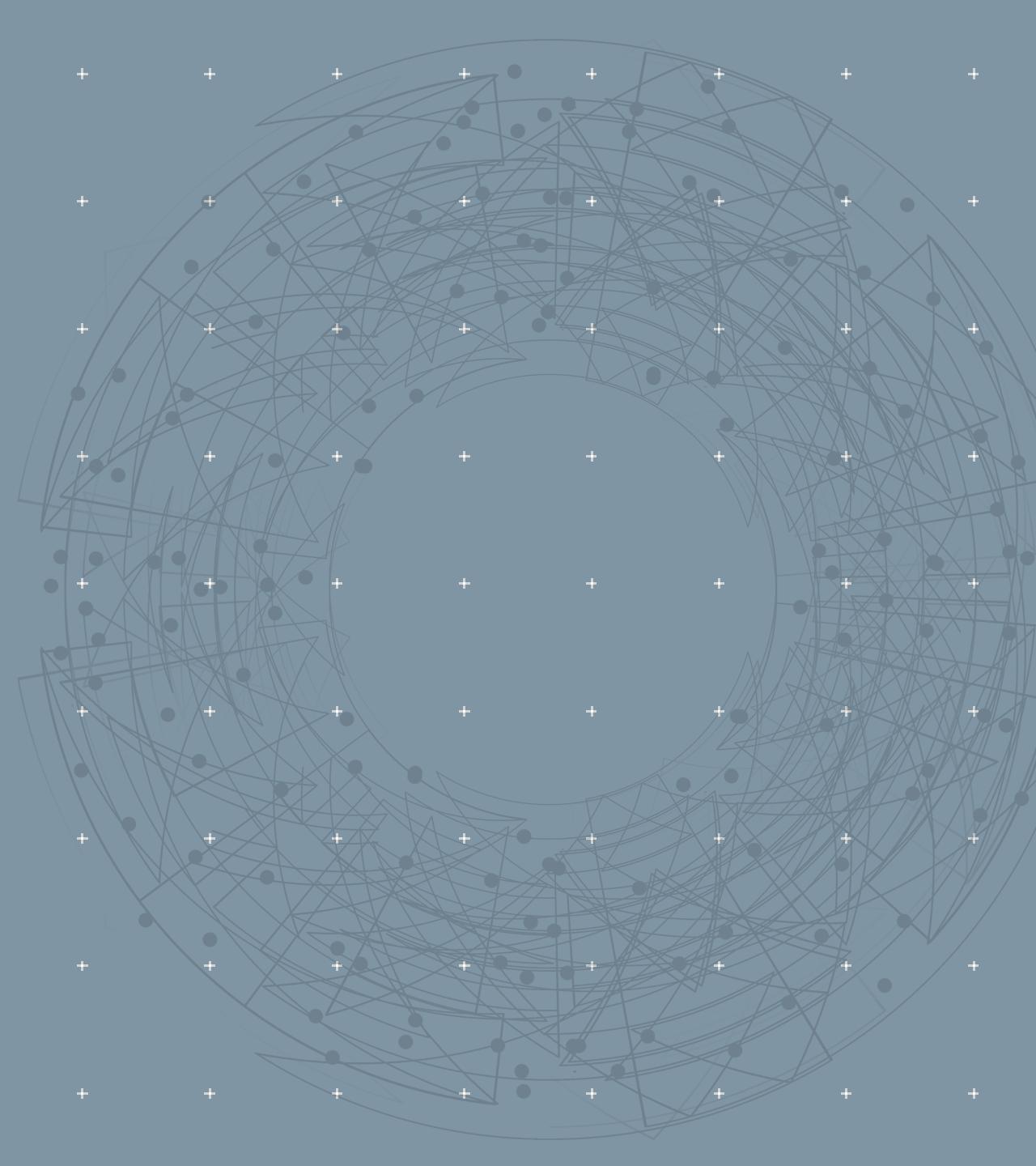
LwM2M	MQTT
Security based on strong DTLS protocol	No innate security features
Natively supports certificates & key-based authentication methods	Introducing strong security results in worse performance

IoT Device 를 위한 프로토콜 - LWM2M

Embedded Code Example : GET Resource

```
// Declaring pointers for access to Client resources
MbedCloudClientResource *button_res;

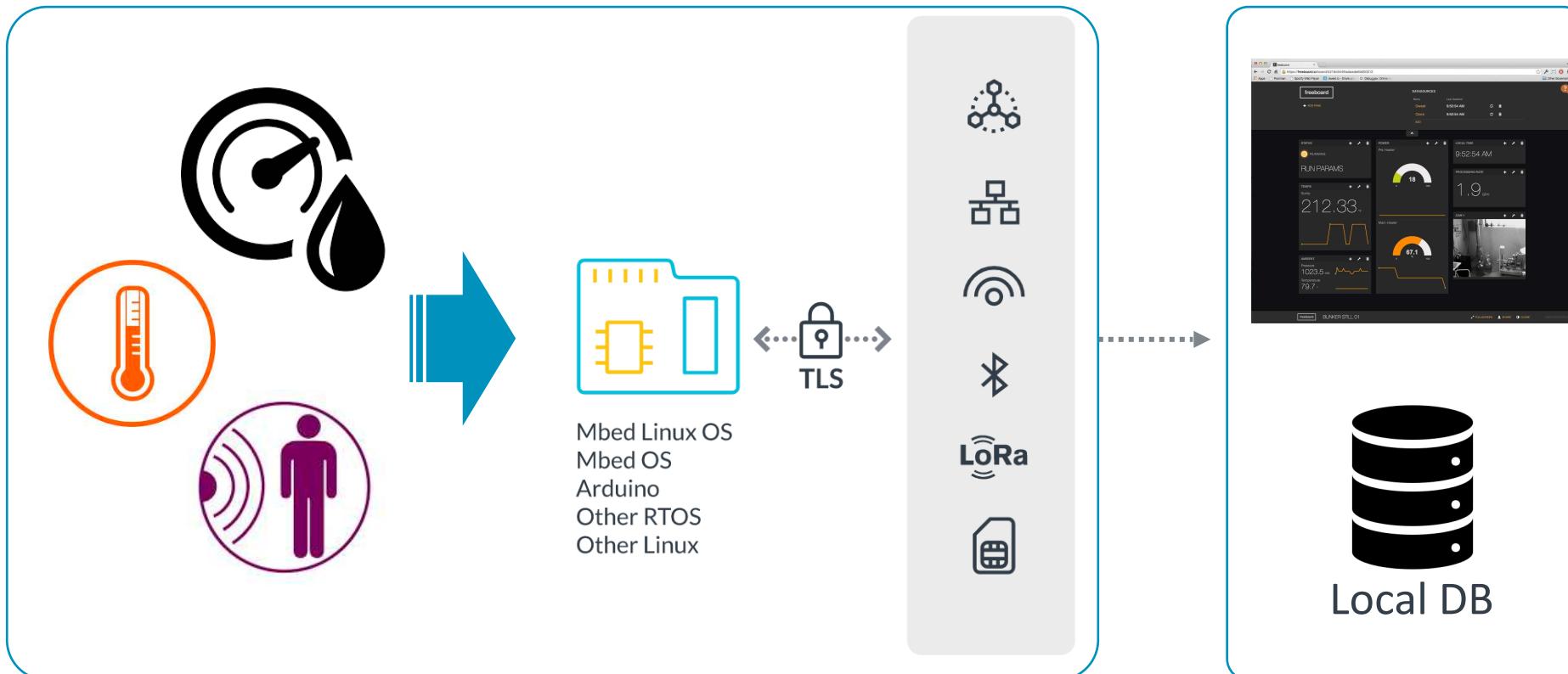
// Creating resources, which can be written or read from the cloud
button_res = client.create_resource("3200/0/5501","button_count");
button_res->set_value(0);
button_res->methods(M2MMMethod::GET);
button_res->observable(true);
button_res->attach_notification_callback(button_callback);
```



Device data to Web Application

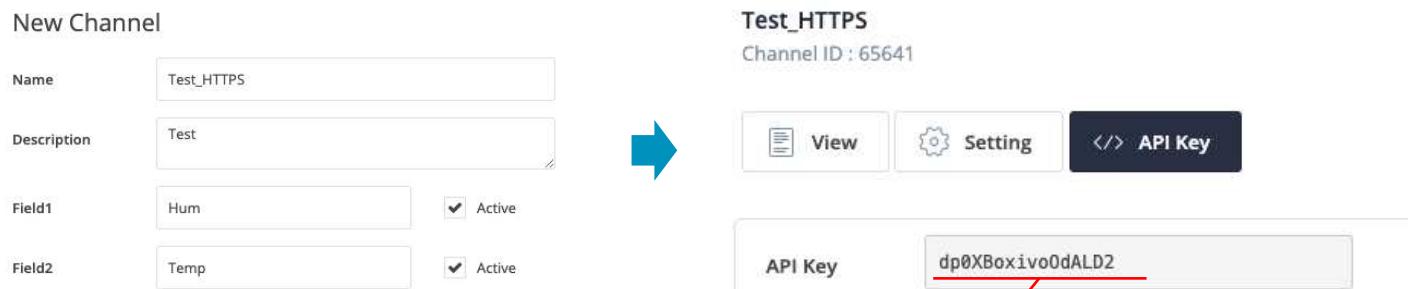
Mbed OS -> HTTP -> Web app. example (Step 1)

** 실습을 위한 Example로만 사용하셔야 합니다!



Mbed OS -> HTTP -> Web app. example (Step 2)

1. HTTP example 을 Mbed studio 로 다운로드 받습니다.
- https://os.mbed.com/users/Daniel_Lee/code/http-example-ForTheKsystem/
2. Mbed_app.json 에서 WiFi 의 SSID, PASSWORD 를 수정합니다.
3. Theksystem 에서 계정을 생성후 Channels -> My Channels -> New Channel 로 Channel 을 생성하고 아래 그림과 같이 Name, Description, Field1, Field2 를 각각 넣고 저장합니다. 생성된 Channel 의 APIkey 를 확인합니다.

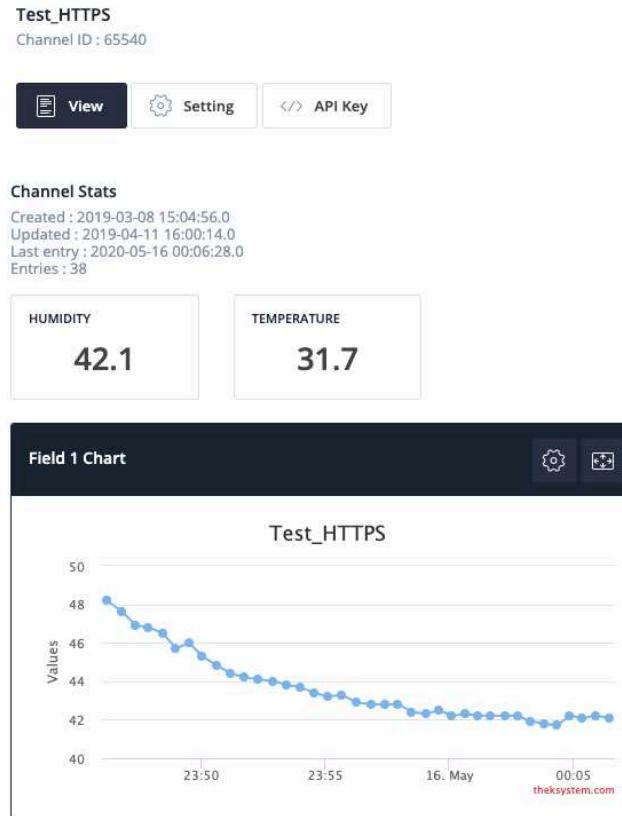


4. 3에서 생성된 APIkey 를 sorce/mbed-http.cpp 의 theKsystemsAPIkey 에 넣어주고 컴파일 및 Serial 확인합니다.

```
char* theKsystemsURL = "http://api.theksystem.com";  
char* theKsystemsAPIkey = "dp0XBoxivo0dALD2";
```

Mbed OS -> HTTP -> Web app. example (Step 3)

5. View 버튼을 눌러서 현재 센서값을 확인합니다.



Serial message

Update count 135 (Every 10sec, data update to theK server)
HT221 hum: 41.7 , temp: 31.7 C
[DEBUG] http://api.theksystem.com/update?apiKey=9Q8v0z1lguav6Ou4&field1=41.7&field2=31.7

----- HTTP GET response -----
Status: 200 - OK
Headers:
Server: nginx/1.17.4
Date: Fri, 15 May 2020 15:10:02 GMT
Content-Type: application/json; charset=UTF-8
Content-Length: 1
Connection: keep-alive
ETag: "0cfcd208495d565ef66e7dff9f98764da"
Body (1 bytes):
0



Device data to
Amazon Web Services
IoT Core

Reference Guide Source Code

- Simple Guide Documents
 - <https://os.mbed.com/teams/mbed-os-examples/code/mbed-os-example-aws/>

Prepare IoT Core - Step#1

- 무료계정 생성
 - <https://aws.amazon.com/iot-core/>

The screenshot shows the AWS IoT Core landing page. At the top, there's a dark navigation bar with the AWS logo, contact links (Contact Sales, Support, English, My Account), and a prominent orange "Create an AWS Account" button. Below the bar is a secondary navigation menu with links for Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, Explore More, and a search icon. The main content area features a large banner with the heading "AWS IoT Core" in white, followed by the subtext "Easily and securely connect devices to the cloud. Reliably scale to billions of devices and trillions of messages." A yellow call-to-action button at the bottom left says "Get started with IoT Core for free". The background of the banner is a stylized image of wind turbines.

Prepare IoT Core - Step#2

- IoT Core 접속

The image shows two side-by-side screenshots of the AWS Management Console.

Left Screenshot: The title bar says "AWS Management Console". On the left, there's a sidebar titled "AWS services" with a "Find Services" search bar containing the text "IoT core". An orange arrow points to this search bar. Below it, a list includes "IoT Core" and "All services".

Right Screenshot: The title bar says "aws" and "Resource Groups". The main area is a grid of service categories. An orange arrow points from the "IoT core" search result in the left screenshot to the "IoT Core" service in the right screenshot. The right screenshot also shows other services like Compute, Blockchain, Analytics, Business Applications, End User Computing, and Internet Of Things.

Category	Services
Compute	EC2, Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS Outposts, EC2 Image Builder
Blockchain	Amazon Managed Blockchain
Analytics	Athena, EMR, CloudSearch, Elasticsearch Service, Kinesis, QuickSight, Data Pipeline, AWS Data Exchange, AWS Glue, AWS Lake Formation, MSK
Business Applications	Alexa for Business, Amazon Chime, WorkMail, Amazon Honeycode
End User Computing	WorkSpaces, AppStream 2.0, WorkDocs, WorkLink
Internet Of Things	IoT Core, FreeRTOS, IoT 1-Click, IoT Analytics, IoT Device Defender, IoT Device Management
Storage	S3, EFS, FSx, S3 Glacier, Storage Gateway, AWS Backup
Management & Governance	AWS Organizations, CloudWatch, AWS Auto Scaling, CloudFormation, CloudTrail, Config, OpsWorks, Service Catalog
Security, Identity, & Compliance	IAM, Resource Access Manager, Cognito, Secrets Manager, GuardDuty

Prepare IoT Core - Create a Policy #1

AWS IoT

- Monitor
- Onboard
- Manage
- Greengrass
- ▼ Secure
 - Certificates
 - Policies** 1
 - CAs
 - Role Aliases
 - Authorizers
- Defend
- Act
- Test



You don't have any policies yet

AWS IoT policies give things permission to access AWS IoT resources (like other things, MQTT topics, or thing shadows).

[Learn more](#) 2 [Create a policy](#)

Prepare IoT Core - Create a Policy #2

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name
Mbed_AWS_Policy

1

Add statements

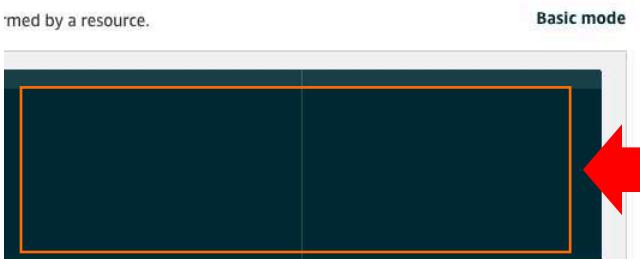
Policy statements define the types of actions that can be performed by a resource.

Action
Please use commas to separate actions. e.g. iot:Publish, iot:Subscribe
Resource ARN
Specific resources could include client ID ARN, topic ARN, or topic filter ARN.
Effect
 Allow Deny

2 Advanced mode

Add statement

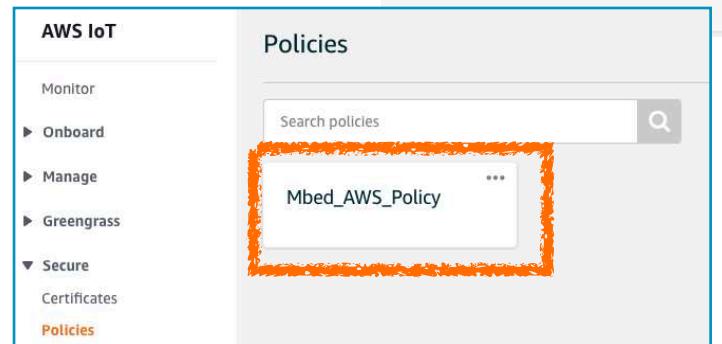
Basic mode



복사
3

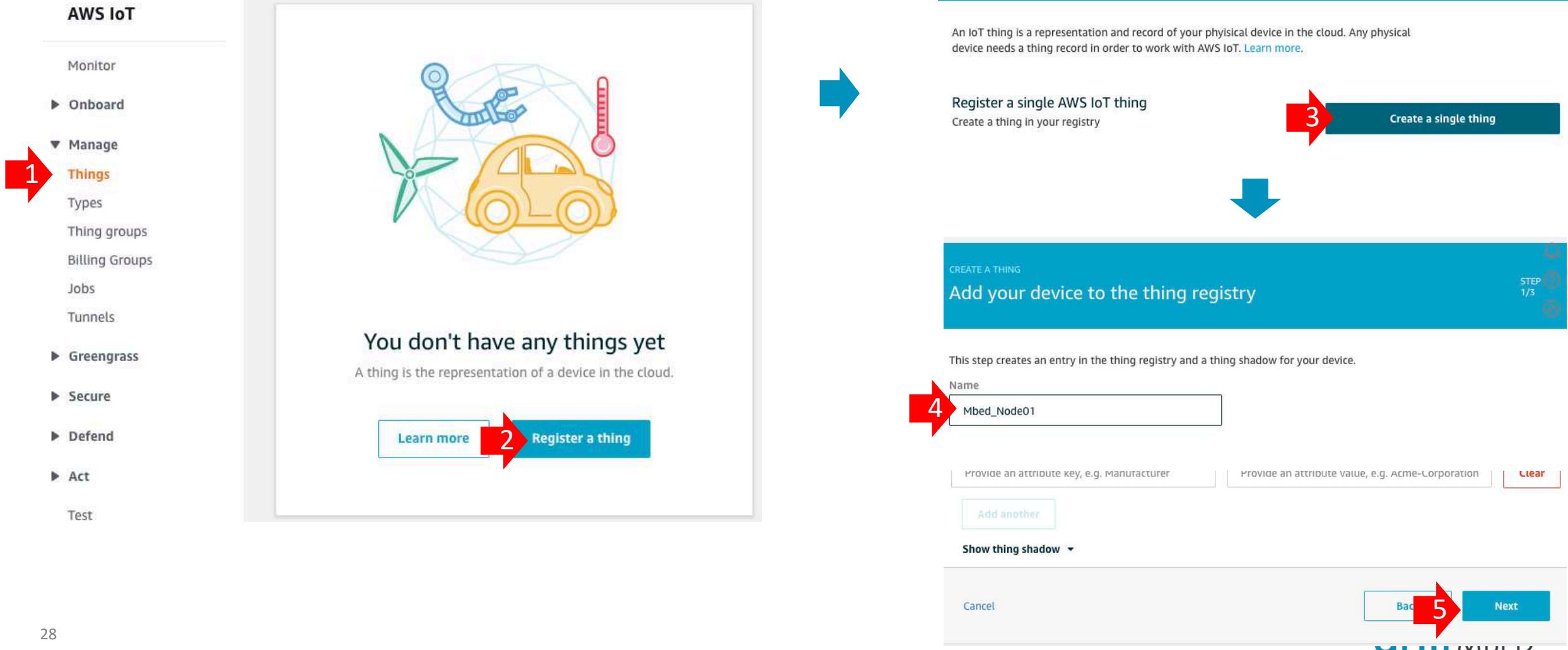
4

Create



```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "iot:Connect",  
      "Resource": "*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": "iot:Publish",  
      "Resource": "*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": "iot:Subscribe",  
      "Resource": "*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": "iot:Receive",  
      "Resource": "*"  
    }  
  ]  
}
```

Prepare IoT Core - Add an IoT device #1



Prepare IoT Core - Add an IoT device #2

CREATE A THING
Add a certificate for your thing

STEP 2/3

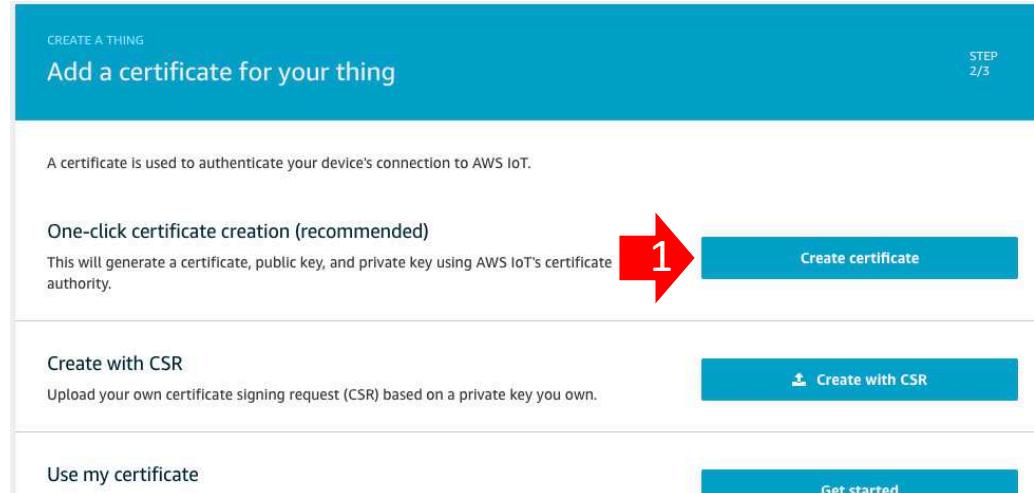
A certificate is used to authenticate your device's connection to AWS IoT.

One-click certificate creation (recommended)
This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

1 [Create certificate](#)

Create with CSR
Upload your own certificate signing request (CSR) based on a private key you own.
[Create with CSR](#)

Use my certificate
[Get started](#)



Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	5.cert.pem	2 Download
A public key	5.public.key	Download
A private key	5.private.key	3 Download

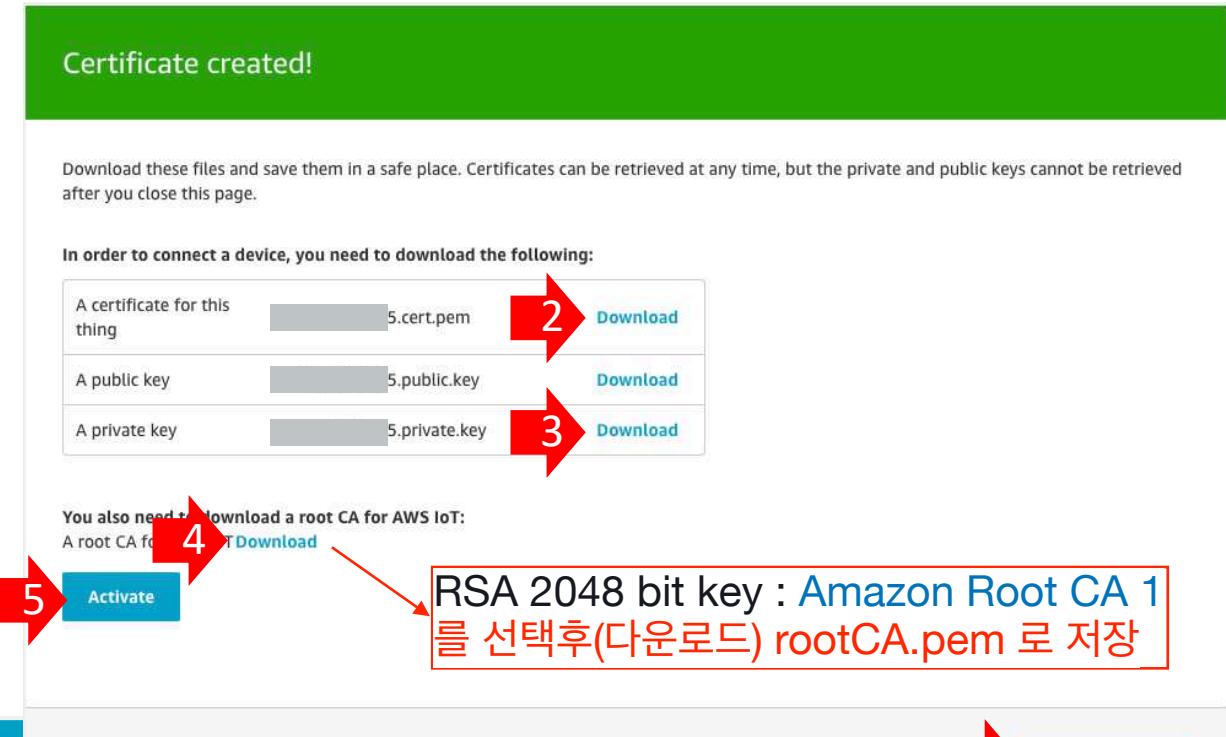
You also need to download a root CA for AWS IoT:
A root CA for IoT [Download](#)

4 [Download](#)

5 [Activate](#)

6 [Done](#) [Attach a policy](#)

RSA 2048 bit key : Amazon Root CA 1
를 선택후(다운로드) rootCA.pem 로 저장

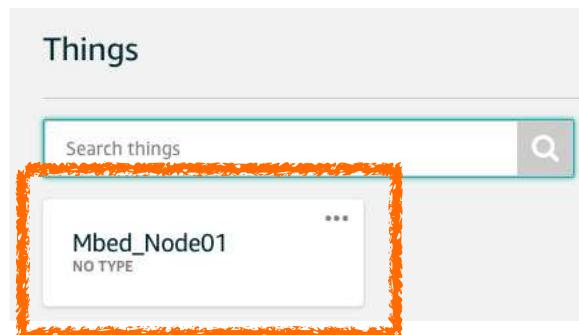


AWS IoT

Things

Search things

Mbed_Node01



CREATE A THING
Add a policy for your thing

Select a policy to attach to this certificate:

Search policies

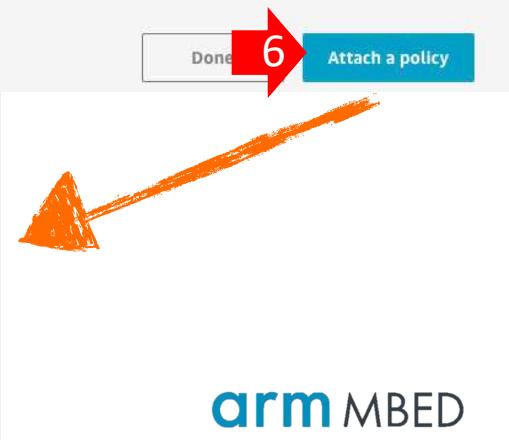
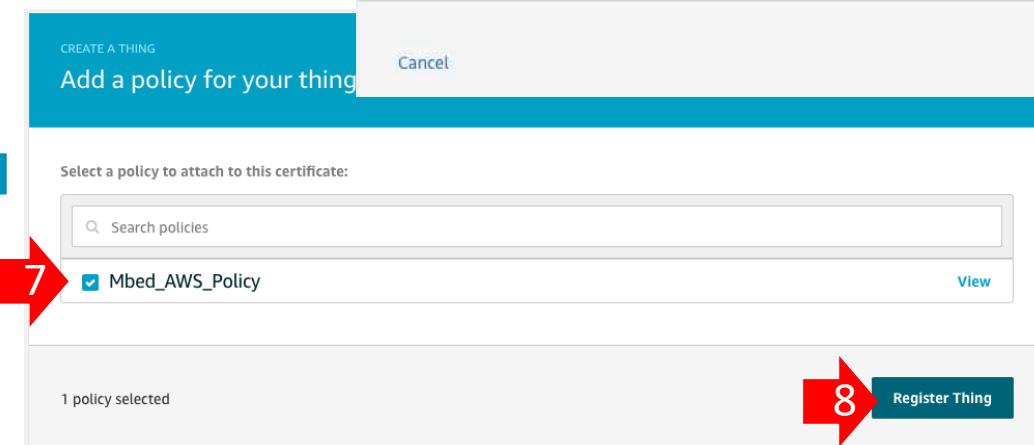
Mbed_AWS_Policy

View

1 policy selected

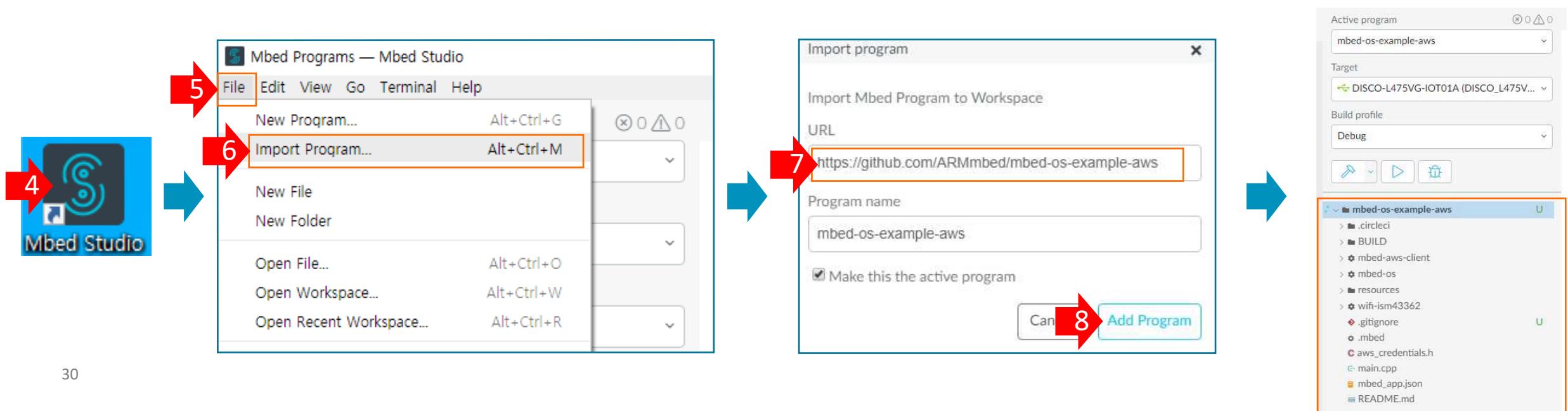
7 [Register Thing](#)

8 [Done](#) [Attach a policy](#)



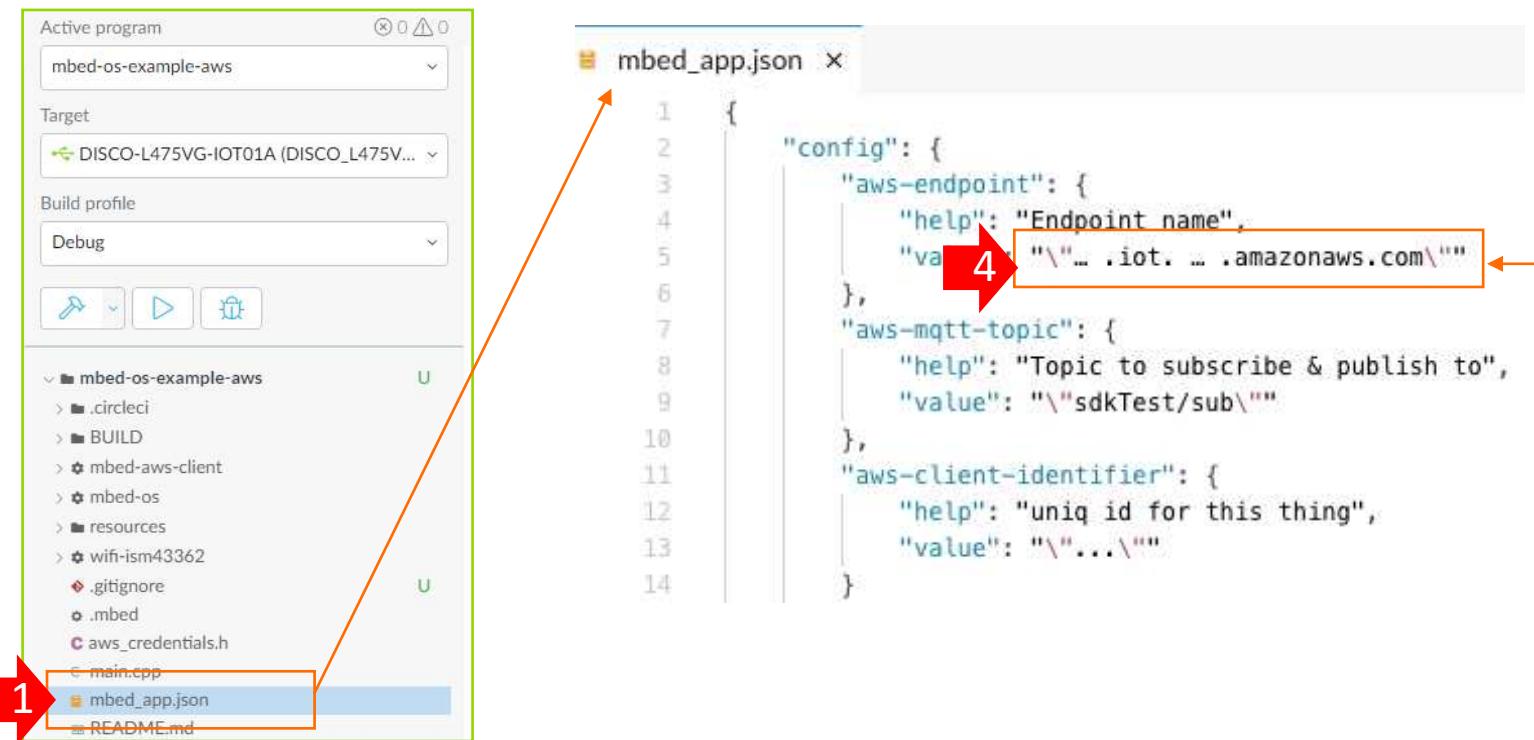
Prepare Source Code

<https://os.mbed.com/teams/mbed-os-examples/code/mbed-os-example-aws/>



Prepare Configures - mbed_app.json #1

Mbed Studio



<https://aws.amazon.com/iot-core/>

The screenshot shows the AWS IoT Settings page. On the left, a sidebar lists 'AWS IoT' (selected), 'Monitor', 'Onboard', 'Manage' (expanded to show 'Things', 'Types', 'Thing groups', 'Billing Groups'), 'Jobs', 'Tunnels', 'Greengrass', 'Secure' (expanded to show 'Certificates', 'Policies', 'CAs', 'Role Aliases', 'Authorizers'), 'Defend', 'Act', 'Test', 'Software', 'Settings' (selected), and 'Learn'. The main content area is titled 'Custom endpoint'. It says: 'This is your custom endpoint that allows you to connect to AWS IoT. Each of your things can have its own endpoint. This is also an important property to insert when using an MQTT client or the AWS IoT Device SDK.' It states: 'Your endpoint is provisioned and ready to use. You can now start to publish and subscribe messages to your endpoint.' A red box highlights the 'Endpoint' input field, which contains 'a2\u2026t.ap-northeast-2.amazonaws.com'. A red arrow labeled '3' points to this field. Below the endpoint section is a 'Logs' section with a 'Level of verbosity' dropdown set to 'Disabled' and a 'Edit' button. At the bottom is an 'Event-based messages' section with a table:

Event	Publish
Job: completed, canceled	Disabled

A red arrow labeled '2' points to the 'Settings' link in the sidebar.

Prepare Configures - mbed_app.json #2

```
mbed_app.json x
14    }
15  },
16  "target_overrides": {
17    "*": {
18      "mbed-trace.enable": true,
19      "mbed-trace.max-level": "TRACE_LEVEL_INFO",
20      "aws-client.log-puts": "aws_iot_puts",
21      "aws-client.log-level-global": "IOT_LOG_INFO",
22      "rtos.main-thread-stack-size": 8192,
23      "rtos.thread-stack-size": 2048,
24      "platform.error-filename-capture-enabled": true,
25      "platform.stdio-convert-newlines": true,
26      "platform.stdio-baud-rate": 115200
27    },
28    "DISCO_L475VG_IOT01A": {
29      "target.network-default-interface-type": "WIFI",
30      "nsapi.default-wifi-security": "WPA_WPA2",
31      "nsapi.default-wifi-ssid": "\\"SSID\\\"",
32      "nsapi.default-wifi-password": "\\"PASSWORD\\\""
33    }
34  }
35 }
```

SSID & PASSWORD 수정 후 컴파일 진행

WiFi 주파수가 2.4Ghz 인지 확인!!

Prepare Certificates

```
: aws_credentials.h x
```

```
1 // AWS Certificates
2
3 #ifndef AWS_CREDENTIALS_H
4 #define AWS_CREDENTIALS_H
5
6 namespace aws {
7     namespace credentials {
8
9         const char rootCA[] = "-----BEGIN CERTIFICATE-----\n"
10        "...\\n"
11        "...\\n"
12        "...\\n"
13        "-----END CERTIFICATE-----";
14
15        const char clientCrt[] = "-----BEGIN CERTIFICATE-----\n"
16        "...\\n"
17        "...\\n"
18        "...\\n"
19        "-----END CERTIFICATE-----";
20
21        const char clientKey[] = "-----BEGIN RSA PRIVATE KEY-----\\n"
22        "...\\n"
23        "...\\n"
24        "...\\n"
25        "-----END RSA PRIVATE KEY-----";
26

```



Check Operation Status - Set Topic

The screenshot shows the AWS IoT MQTT client interface. On the left, a sidebar lists navigation options: Monitor, Onboard, Manage, Greengrass, Secure, Defend, Act, and Test. A red arrow labeled '1' points to the 'Test' button. The main area is titled 'MQTT client' with a question mark icon and shows a connection status of 'Connected as iotconsole-1594702802095-0'. Below this, a blue header bar contains the text 'Subscriptions'. Underneath, there are two buttons: 'Subscribe to a topic' and 'Publish to a topic'. To the right, a large text area contains instructions: 'Subscribe' (Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.) and 'Subscription topic' (with a text input field containing 'sdkTest/sub'). A red arrow labeled '2' points to this input field. Below it is a 'Max message capture' input field set to '100', with a red arrow labeled '3' pointing to a 'Subscribe to topic' button.

mbed_app.json

```
{  
  "config": {  
    "aws-endpoint": {  
      "help": "Endpoint name",  
      "value": "\u2026.iot.\u2026.amazonaws.com\""  
    },  
    "aws-mqtt-topic": {  
      "help": "Topic to subscribe & publish to",  
      "value": "\"\\\"sdkTest/sub\\\""  
    },  
  },  
}
```

Check Operation Status - Subscribe to a Topic

Mbed Studio

```
Output >_ DISCO-L475VG-IOT01A (DISCO_L475VG_IOT01A) Debug Console Problems Libraries

[INFO ] [MQTT] [1 Jan 1970 0:1] Establishing new MQTT connection.
[INFO ] [MQTT] [1 Jan 1970 0:1] Anonymous metrics (SDK language, SDK version) will be provided to AWS IoT. Recompile with AWS_IOT_MQTT_ENABLE_METRICS set to 0 to disable.
ISM43362Interface socket send nolock id 0 size 68
[INFO ] [MQTT] [1 Jan 1970 0:1] (MQTT connection 0x2000FC10, CONNECT operation 0x2000FCF0) Waiting for operation completion.
ISM43362Interface socket_check read read_amount 33
ISM43362Interface socket_recv: recv=5
ISM43362Interface socket_recv: recv=28
[INFO ] [MQTT] [1 Jan 1970 0:1] (MQTT connection 0x2000FC10, CONNECT operation 0x2000FCF0) Wait complete with result SUCCESS.
[INFO ] [MQTT] [1 Jan 1970 0:1] New MQTT connection 0x10002AA8 established.
ISM43362Interface socket_send nolock id 0 size 47
[INFO ] [MQTT] [1 Jan 1970 0:1] (MQTT connection 0x2000FC10) SUBSCRIBE operation scheduled.
[INFO ] [MQTT] [1 Jan 1970 0:1] (MQTT connection 0x2000FC10, SUBSCRIBE operation 0x2000FCF0) Waiting for operation completion.
ISM43362Interface socket_check read read_amount 34
ISM43362Interface socket_recv: recv=5
ISM43362Interface socket_recv: recv=29
[INFO ] [MQTT] [1 Jan 1970 0:1] (MQTT connection 0x2000FC10, SUBSCRIBE operation 0x2000FCF0) Wait complete with result SUCCESS.
[INFO ] [Main]: sending warning message: Warning: Only 10 second(s) left to say your name !
ISM43362Interface socket_send nolock id 0 size 96
[INFO ] [MQTT] [1 Jan 1970 0:1] (MQTT connection 0x2000FC10) MQTT PUBLISH operation queued.
[INFO ] [MQTT] [1 Jan 1970 0:1] (MQTT connection 0x2000FC10, PUBLISH operation 0x2000FCF0) Waiting for operation completion.
ISM43362Interface socket_check read read_amount 33
ISM43362Interface socket_recv: recv=5
ISM43362Interface socket_recv: recv=28
[INFO ] [MQTT] [1 Jan 1970 0:1] (MQTT connection 0x2000FC10, PUBLISH operation 0x2000FCF0) Wait complete with result SUCCESS.
```

iotconsole-1594712785841-3

Export Clear Pause

The screenshot shows the AWS IoT console interface. A red arrow points from the Mbed Studio terminal output to the 'Publish' section of the AWS IoT console. The 'Publish' section has a text input field containing 'sdkTest/sub' and a 'Publish to topic' button. Below it, a message history shows a single entry: 'sdkTest/sub' at 'Jul 14, 2020 4:53:12 PM +0900'. The message content is 'Warning: Only 10 second(s) left to say your name !'. A green bar at the bottom states 'We cannot display the message as JSON, and are instead displaying it as UTF-8 String.'

Subscribe to a topic
Publish to a topic

sdkTest/sub

Publish

Specify a topic and a message to publish with a QoS of 0.

sdkTest/sub

1 {
2 "message": "Hello from AWS IoT console"
3 }

sdkTest/sub Jul 14, 2020 4:53:12 PM +0900

We cannot display the message as JSON, and are instead displaying it as UTF-8 String.

Warning: Only 10 second(s) left to say your name !

Export Hide

Check Operation Status - Publish to Topic

The screenshot illustrates the process of publishing a message to an AWS IoT topic and verifying its reception on a connected device.

AWS IoT Core Interface:

- Top Bar:** sdkTest/sub, Export, Clear, Pause
- Publish Section:** "Specify a topic and a message to publish with a QoS of 0."
 - Topic: sdkTest/sub (highlighted with a red box)
 - Message Content:

```
1 [
2   "message": "Hello from AWS IoT console"
3 ]
```
 - Buttons: Publish to topic (with a red arrow pointing to it)
- Subscriptions Section:** sdkTest/sub, Jul 14, 2020 4:59:46 PM +0900, Export, Hide
 - Message Log:

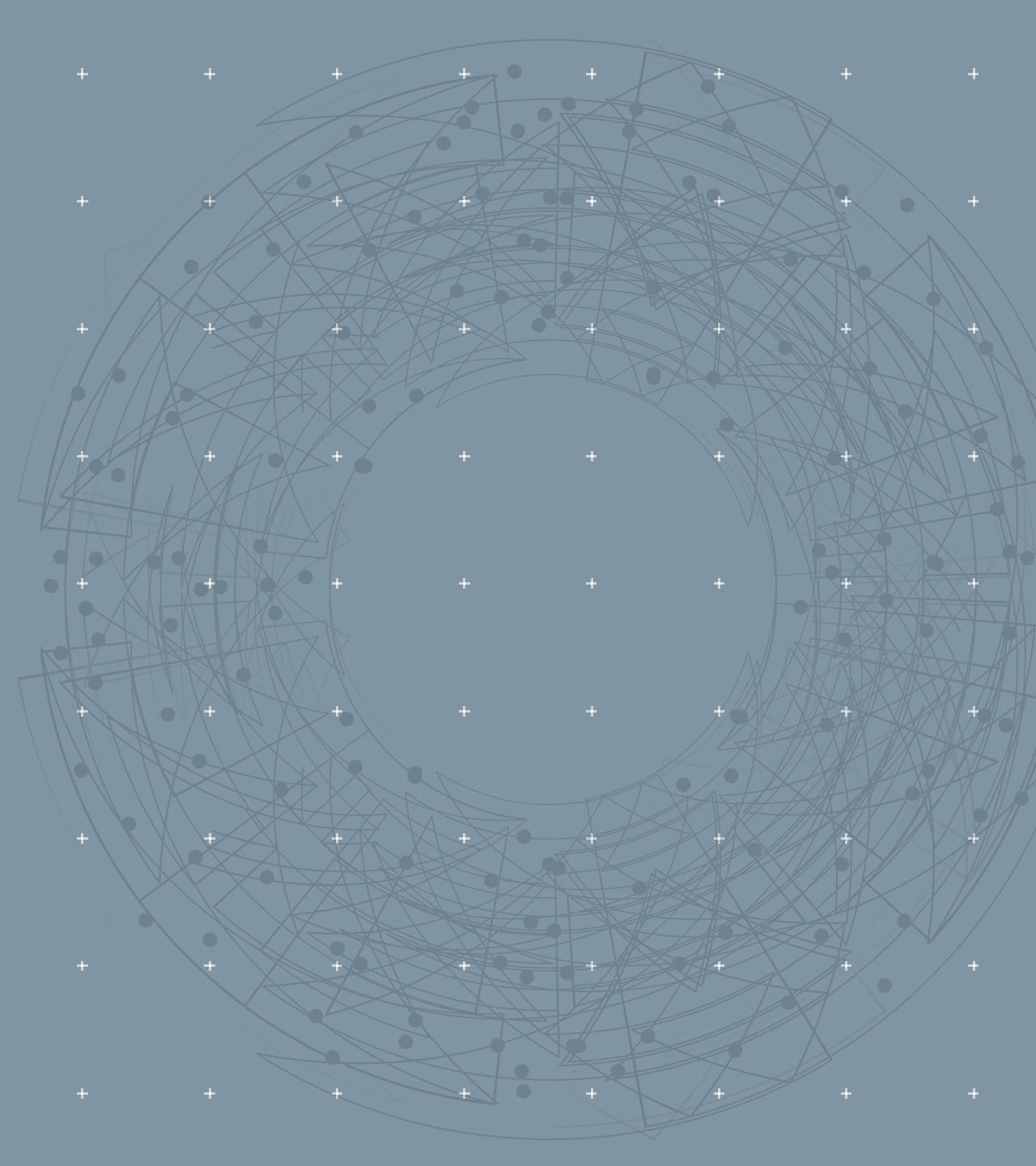
```
{  
  "message": "Hello from AWS IoT console"  
}
```

Mbed Studio Output Console:

- Tab: Output > DISCO-L475VG-IOT01A (DISCO_L475VG_IOT01A) > Debug Console
- Output Log:

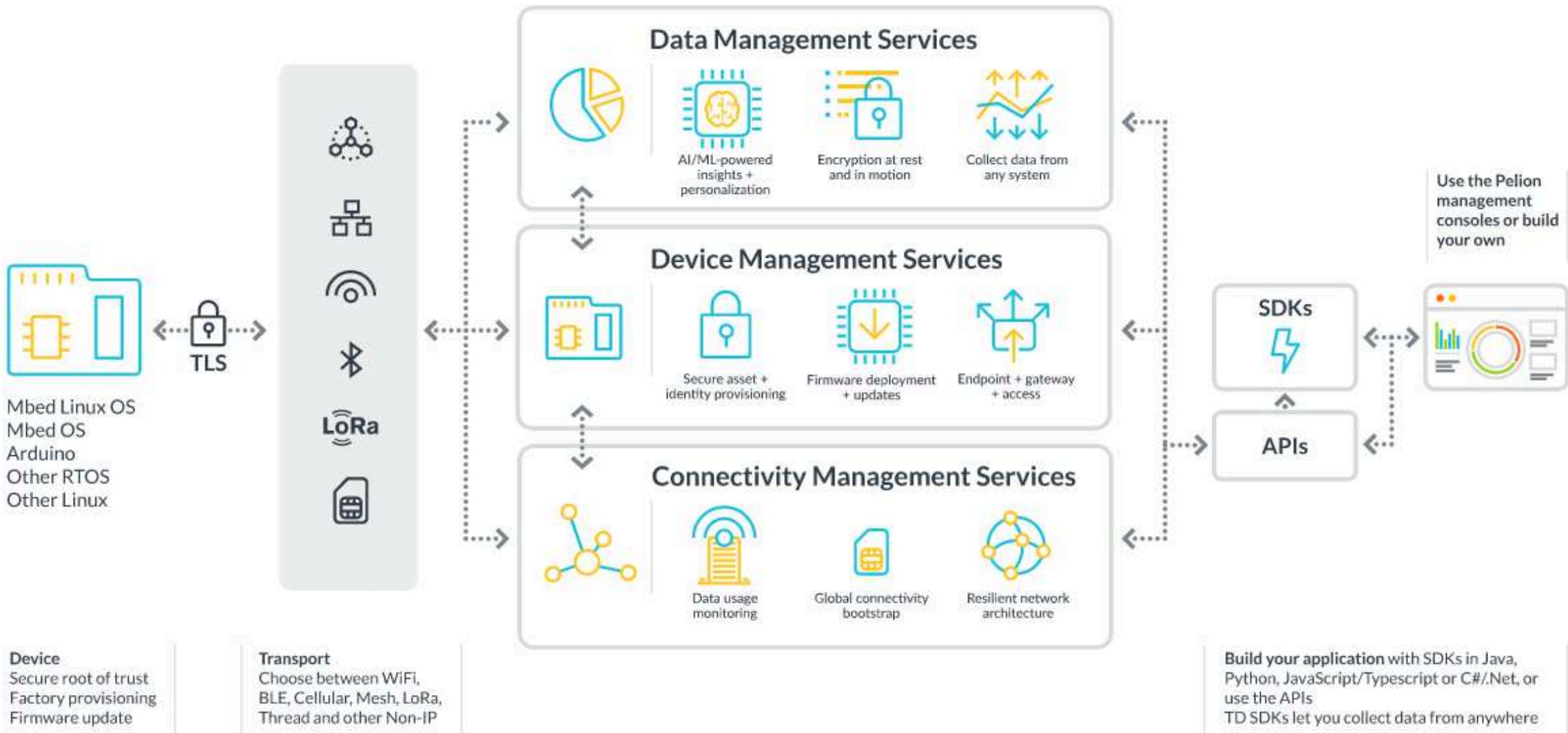
```
[INFO] [Main]: [Debug]] IoTMqtt_Disconnect  
[INFO] [Main]: Waiting.....  
ISM43362Interface socket_check_read read_amount 96  
ISM43362Interface socket_recv: recv=5  
ISM43362Interface socket_recv: recv=91  
[INFO] [TASKPOOL][1 Jan 1970 0:8] Growing a Task pool with a new worker thread...  
ISM43362Interface socket_send_nolock id 0 size 33  
ISM43362Interface socket_check_read read_amount 89  
ISM43362Interface socket_recv: recv=5  
ISM43362Interface socket_recv: recv=84  
[INFO] [Main]: Hello [  
  "message": "Hello from AWS IoT console"  
]
```

An orange arrow points from the highlighted message in the AWS IoT Core subscription log to the corresponding message in the Mbed Studio output log.



Device data to Arm Pelion Device Management

Arm Pelion



Pelion De.M - IoT Connection Tutorial – Online Compiler(Step 1)

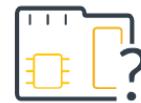
The screenshot shows the Mbed + Pelion IoT Platform interface. At the top, there's a search bar with 'Compiler' and a language selector 'T'. Below it, a dropdown menu says 'Mbed + Pelion IoT Platform'. On the left, there's a sidebar with 'Events', 'Mbed + Pelion Overview' (with a link), 'Resources' (with links to 'Pelion official website', 'Documentation', 'Pelion forum', and 'Pelion support'), and a 'Connectivity Management' section with 'Connectivity Management Console' and 'Data Management' sections. A red arrow points to the 'Connect a device' option under 'Device Management'.

Based on Online compiler

<https://os.mbed.com/guides/connect-device-to-pelion/>

Youtube Guide

<https://www.youtube.com/watch?v=NsTfxw-KDa8>



Choose a board and connectivity method.



Add an out-of-the box example program to our online IDE.



Add a certificate to the example program, allowing your board to connect.



Visit the Pelion Device Management portal to see IoT data from your board.

1

2

3

4

Pelion De.M - IoT Connection Tutorial – Online Compiler(Step 2)

➤ mbed_app.json 에서 SSID & PASSWORD를 수정해야 합니다!

➤ Find below configure

```
"nsapi.default-wifi-ssid" : "\"SSID\"",  
"nsapi.default-wifi-password" : "\"Password\""
```

➤ Set SSID & Password

Ex) "\"SSID\"" → "\"WifiAP\""

"\"Password\"" → "\"a12345678\""

Pelion De.M - IoT Connection Tutorial – Online Compiler(Step 3)

- Pelion device management에서 Device ID(Endpoint Name) 등록여부 확인

-  Daniel Lee
-  Search features
-  Dashboard
-  Device directory
- Devices
- Device events
-  Device identity

Devices		Initial Register
View and manage your devices.		
<input type="button" value="Actions"/> ▼	Saved filters ▼	<input checked="" type="checkbox"/> Registered devices
<input type="checkbox"/> Device ID ▾	Endpoint name	016f058fe55c000... 158b15 016f058fe55c000... 158b15

Serial Terminal Log

```
Connecting to the network using Wifi...
Connected to the network successfully. IP address: 192.168.1.3
Initializing Pelion Device Management Client...
Initialized Pelion Client. Registering...
Registered to Pelion Device Management. Endpoint Name: 016f058fe55c000000000000100158b15
```



Pelion De.M - IoT Connection Tutorial – Online Compiler(Step 4)

- 등록된 Object 의 동작 확인
 - Ex) Check button object operation as below

The screenshot shows the Arm Pelion Device Management interface. On the left, the navigation bar includes options like Daniel Lee, Search features, Dashboard, Device directory, Devices, Device events, and Device identity. A red arrow points to the Device identity option. The main area displays a list of devices under the heading 'Devices'. A red arrow points to the 'Device ID' field, which contains the value '16f058fe55c000...158b15'. A larger red arrow points to the 'RESOURCES' tab in the 'Device details' modal window. The modal shows resource details for three resources: '/3200/0' (Digital Input - 0), '/3200/0/5501' (button_resour... input Counter), and '/3201' (Digital Output). Below the modal, a graph shows the value of the digital input counter over time, starting at 1 and increasing to 5. A red arrow points to the 'GRAPH' tab in the bottom left of the graph area. At the bottom right, there is an 'armMBED' logo.

arm PELION
DEVICE MANAGEMENT

Daniel Lee

Search features

Dashboard

Device directory

Devices

Device events

Device identity

Devices

View and manage your devices.

Actions

Saved filters

Registered

Device ID

16f058fe55c000...158b15

Endpoint name

Device details

016f058fe55c0000000000000000...

016f058fe55c0000000000000000100158b15

SUMMARY ATTRIBUTES RESOURCES EVENTS LOG

To find out about resources and subscribing to changes, read the documentation.

Search by Path, Name, Observal

/3200/0

Digital Input - 0

No

/3200/0/5501

button_resour... input Counter

Yes

/3201

Digital Output

No

Value (subscription): 5

GRAPH HISTORY

armMBED



Thank You
Danke
Merci

谢谢
ありがとう
Gracias
Kiitos
감사합니다