

IoT Protocol & Cloud Service

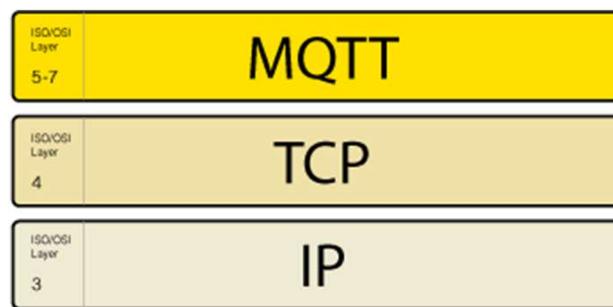
KOOK Joongjin, Ph.D. Assistant Professor (kook@smu.ac.kr)
Dept. of Information Security Engineering
Sangmyung University

MQTT





- ❖ MQTT: Message Queue Telemetry Transport
 - ▶ *publish/subscribe* messaging protocol designed for lightweight M2M communications
- ❖ Server/Client Model over TCP
 - ▶ Client: sensors, Server: broker
- ❖ MQTT Stack
 - ▶ The MQTT protocol is based on top of TCP/IP and both client and broker need to have a TCP/IP stack





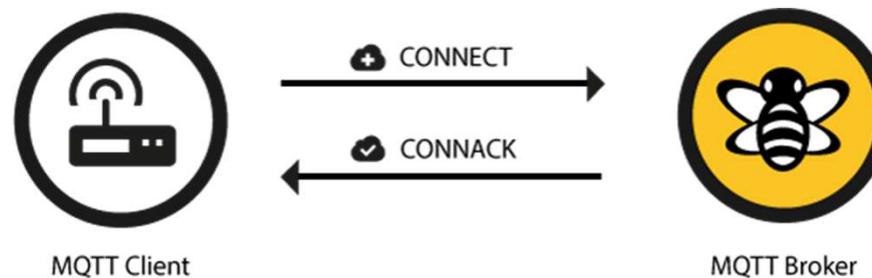
❖ Message oriented

- ▶ Every MSG is a discrete chunk of data
- ▶ Every message is published to an address (called ***topic***)
- ▶ Clients may subscribe to multiple topics
- ▶ Every client subscribed to a topic receives every message published to the topic



❖ Connection Flow

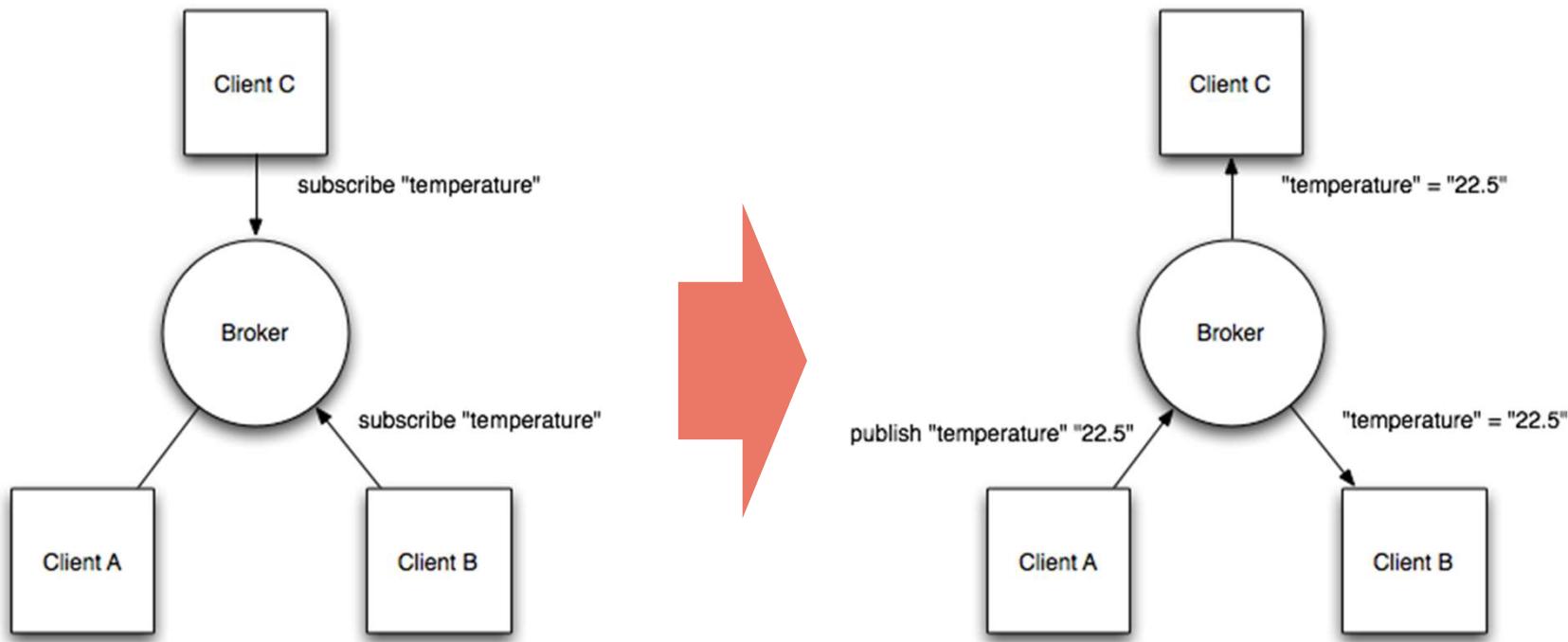
- ▶ The MQTT connection itself is always between one client and the broker
 - ✖ No client is connected to another client directly
- ▶ The connection is initiated through a client sending a **CONNECT** message to the broker
- ▶ The broker response with a **CONNACK** and a status code





❖ Example

- ▶ Simple network with 3-clients and 1-broker





❖ Example

- ▶ Simple network with 3-clients and 1-broker
 - ↳ All 3-clients open TCP connections with the broker
 - ↳ Clients B and C subscribe to the topic “temperature”
 - ↳ Client A published a value of ‘22.5’ for topic “temperature”
 - ↳ The broker forwards the message to all subscribed clients
 - ↳ The publisher subscriber model allows MQTT clients to communicate one-to-one, one-to-many and many-to-one



❖ MQTT Broker

► Mosquitto (<https://mosquitto.org>)

The screenshot shows the official website for Eclipse Mosquitto. The header includes the Eclipse Foundation logo and the cedalo logo. The main title is "Eclipse Mosquitto™ An open source MQTT broker". Below the title, there is a brief description of what Mosquitto is and how it works. A note mentions that Mosquitto is part of the Eclipse Foundation. The page is divided into several sections: "Download and Security", "Test", "Community", and "Related Projects".

Eclipse Mosquitto™
An open source MQTT broker

Eclipse Mosquitto is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol versions 5.0, 3.1.1 and 3.1. Mosquitto is lightweight and is suitable for use on all devices from low power single board computers to full servers.

The MQTT protocol provides a lightweight method of carrying out messaging using a publish/subscribe model. This makes it suitable for Internet of Things messaging such as with low power sensors or mobile devices such as phones, embedded computers or microcontrollers.

The Mosquitto project also provides a C library for implementing MQTT clients, and the very popular mosquitto_pub and mosquitto_sub command line MQTT clients.

Mosquitto is part of the Eclipse Foundation, is an iot.eclipse.org project and is sponsored by cedalo.com.

Download and Security	Test	Community	Related Projects
Mosquitto is highly portable and available for a wide range of platforms. Go to the dedicated download page to find the source or binaries for your platform. Read the Change Log to find out about recent releases.	You can have your own instance of Mosquitto running in minutes, but to make testing even easier, the Mosquitto Project runs a test server at test.mosquitto.org where you can test your clients in a variety of ways: plain MQTT, MQTT over TLS, MQTT over TLS (with client certificate), MQTT over WebSockets and MQTT over WebSockets with TLS.	<ul style="list-style-type: none">• Report bugs or submit changes on the Github repository• Talk to other users on the Mosquitto mailing list.• Get yourself some stickers.• Cite Mosquitto in your academic work.	Paho provides MQTT client library implementations in a wide variety of languages. Streamsheets is an easy to use web based real time spreadsheet interface that can be used to process incoming data from a variety of sources, such as MQTT, OPC-UA, and REST. Developers and non-developers can use



❖ MQTT Broker

▶ Installations & Execution

The screenshot shows a web browser window with the URL mosquitto.org/download/. The page is titled "Download | Eclipse Mosquitto". At the top, there are three logos: mosquitto (blue and orange), ECLIPSE FOUNDATION (yellow and orange), and cedalo (red). Below the logos, there are navigation links for Home, Blog, Download, and Documentation.

Download

Source

- [mosquitto-1.6.9.tar.gz \(319kB\) \(GPG signature\)](#)
- [Git source code repository \(github.com\)](#)

Older downloads are available at <https://mosquitto.org/files/>

Binary Installation

The binary packages listed below are supported by the Mosquitto project. In many cases Mosquitto is also available directly from official Linux/BSD distributions.

Windows

- [mosquitto-1.6.9-install-windows-x64.exe \(~1.4 MB\) \(64-bit build, Windows Vista and up, built with Visual Studio Community 2019\)](#)
- [mosquitto-1.6.9-install-windows-x32.exe \(~1.4 MB\) \(32-bit build, Windows Vista and up, built with Visual Studio Community 2019\)](#)

See also `readme-windows.txt` after installing.

Mac

Mosquitto can be installed from the homebrew project. See `brew.sh` and then use `brew install mosquitto`

Linux distributions with snap support

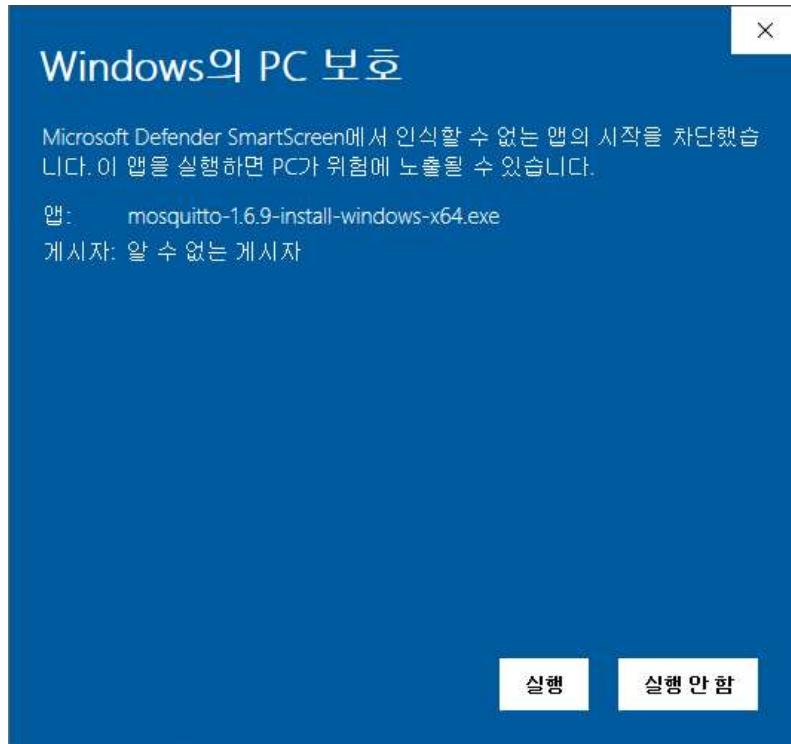
- `snap install mosquitto`



MQTT

❖ MQTT Broker

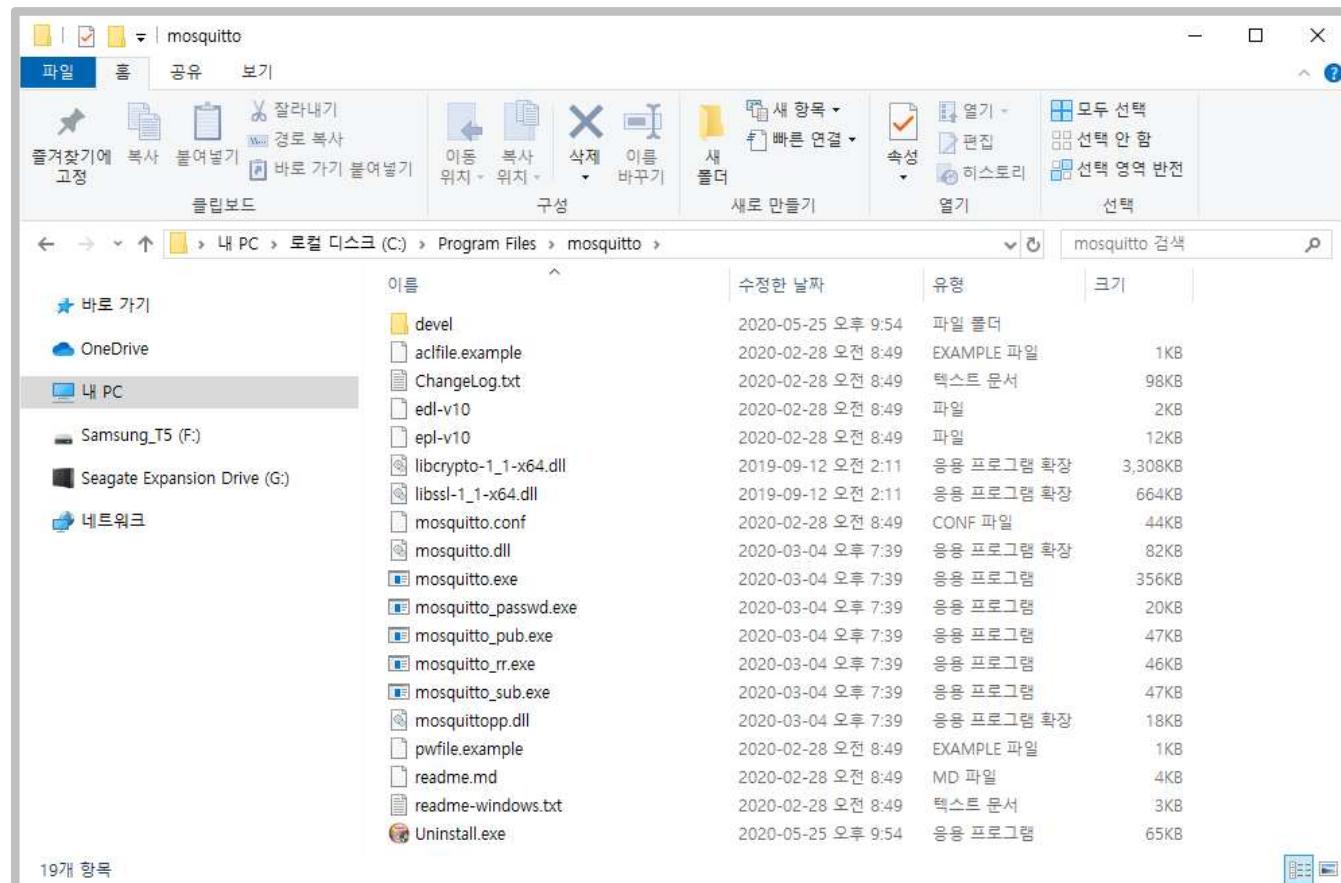
▶ Installations & Execution





❖ MQTT Broker

▶ Installations & Execution





❖ MQTT Broker

- ▶ Installations & Execution
- > mosquitto -v

```
선택 C:\WINDOWS\system32\cmd.exe - mosquitto -v
C:\Program Files\mosquitto 디렉터리

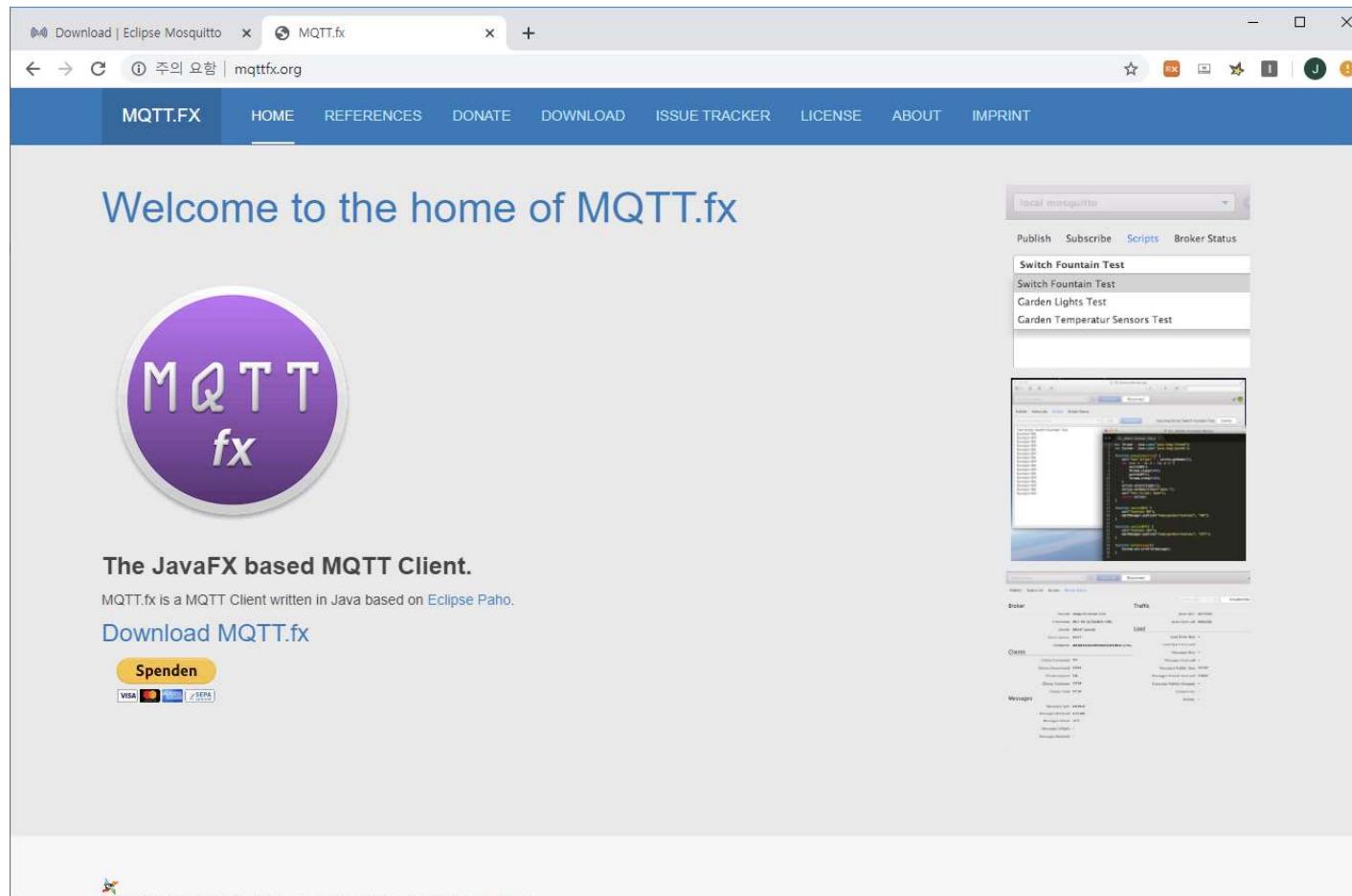
2020-05-25 오후 09:54 <DIR> .
2020-05-25 오후 09:54 <DIR> ..
2020-02-28 오전 08:49 230 aclfile.example
2020-02-28 오전 08:49 100, 295 ChangeLog.txt
2020-05-25 오후 09:54 <DIR> devel
2020-02-28 오전 08:49 1, 569 edl-v10
2020-02-28 오전 08:49 11, 695 ep1-v10
2019-09-12 오전 02:11 3, 386, 880 libcrypto-1_1-x64.dll
2019-09-12 오전 02:11 679, 424 libssl-1_1-x64.dll
2020-02-28 오전 08:49 44, 555 mosquitto.conf
2020-03-04 오후 07:39 83, 456 mosquitto.dll
2020-03-04 오후 07:39 364, 032 mosquitto.exe
2020-03-04 오후 07:39 17, 920 mosquittopp.dll
2020-03-04 오후 07:39 20, 480 mosquitto_passwd.exe
2020-03-04 오후 07:39 47, 616 mosquitto_pub.exe
2020-03-04 오후 07:39 46, 592 mosquitto_rr.exe
2020-03-04 오후 07:39 48, 128 mosquitto_sub.exe
2020-02-28 오전 08:49 355 pwfile.example
2020-02-28 오전 08:49 2, 550 readme-windows.txt
2020-02-28 오전 08:49 3, 434 readme.md
2020-05-25 오후 09:54 66, 035 Uninstall.exe
18개 파일 4, 925, 246 바이트
3개 디렉터리 370, 345, 537, 536 바이트 남음

C:\Program Files\mosquitto>mosquitto -v
1590413492: mosquitto version 1.6.9 starting
1590413492: Using default config.
1590413492: Opening ipv6 listen socket on port 1883.
1590413492: Opening ipv4 listen socket on port 1883.
```



❖ MQTT Client

▶ MQTT.fx (mqttfx.org)





❖ MQTT Client

▶ MQTT.fx (mqttfx.org)

Download

Latest Release

MQTT.fx Version 1.7.1 (more information)

Previous Versions

MQTT.fx Version 1.7.0 (more information)

MQTT.fx Version 1.6.0 (more information)

MQTT.fx Version 1.5.2 (MacOS only, fixes system menu issues with on MacOS 10.13 High Sierra)

MQTT.fx Version 1.5.0 (Will not work with MacOS 10.13 High Sierra, more information)

MQTT.fx Version 1.4.2

MQTT.fx Version 1.4.1

MQTT.fx Version 1.4.0

MQTT.fx Version 1.3.1 (more information)

MQTT.fx Version 1.3.0 (more information)

MQTT.fx Version 1.2.1 (more information)

MQTT.fx Version 1.1.0 (more information)

MQTT.fx Version 1.0.0 (more information)

MQTT.fx Version 0.0.18 (more information)

Index of /apps/mqttfx/1.7.0

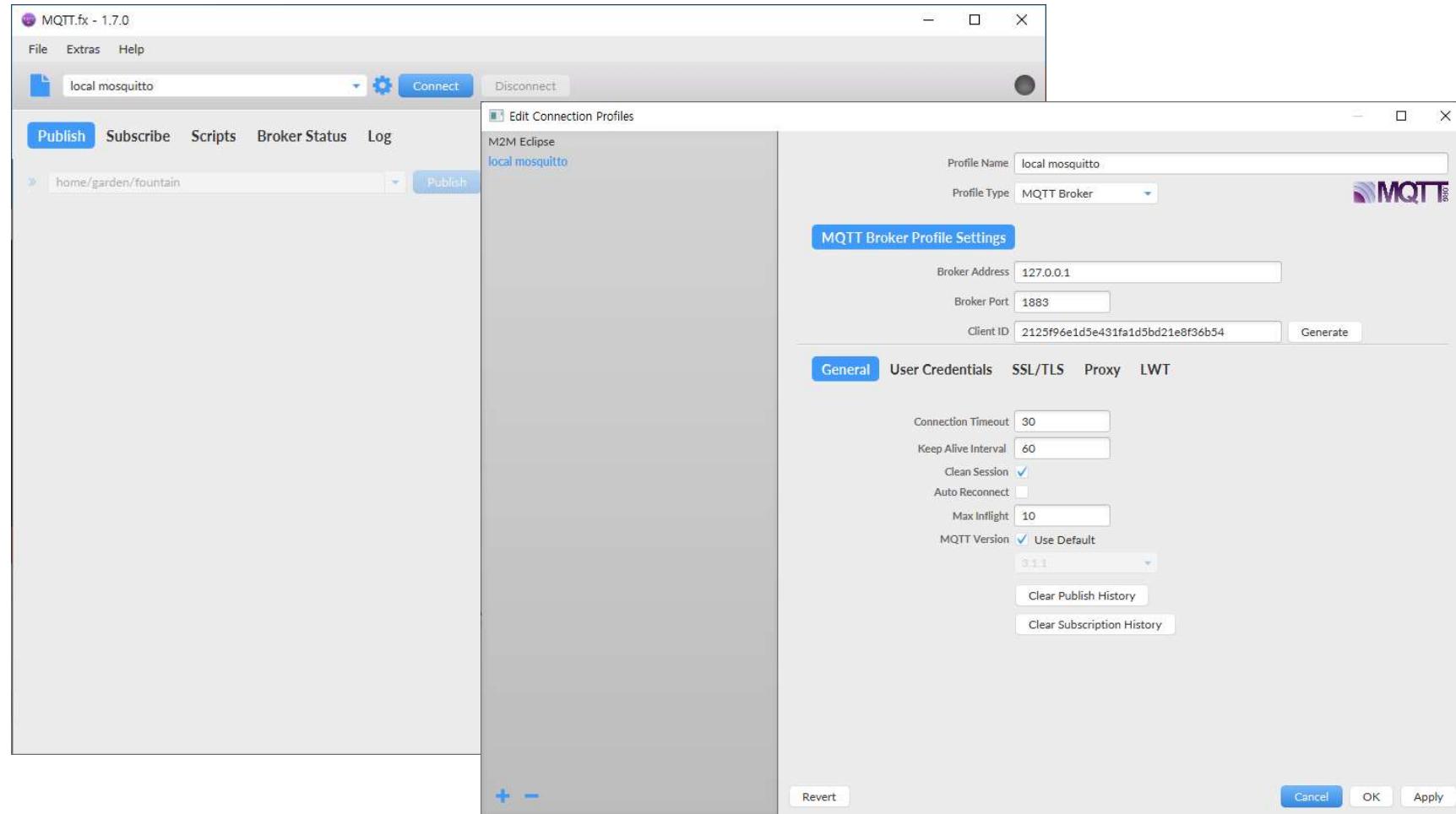
Name	Last modified	Size	Description
Parent Directory		-	
mqttfx-1.7.0-1.i386.rpm	2018-04-08 15:20	99M	
mqttfx-1.7.0-1.x86_64.rpm	2018-04-08 15:55	96M	
mqttfx-1.7.0-32bit.deb	2018-04-08 15:21	75M	
mqttfx-1.7.0-64bit.deb	2018-04-08 15:56	72M	
mqttfx-1.7.0-macos.dmg	2018-04-08 15:12	54M	
mqttfx-1.7.0-windows.exe	2018-04-08 15:12	51M	
mqttfx-1.7.0-windows.x64.exe	2018-04-08 15:12	47M	

www.jensd.de/apps/mqttfx/1.7.0/mqttfx-1.7.0-windows-x64.exe



❖ MQTT Client

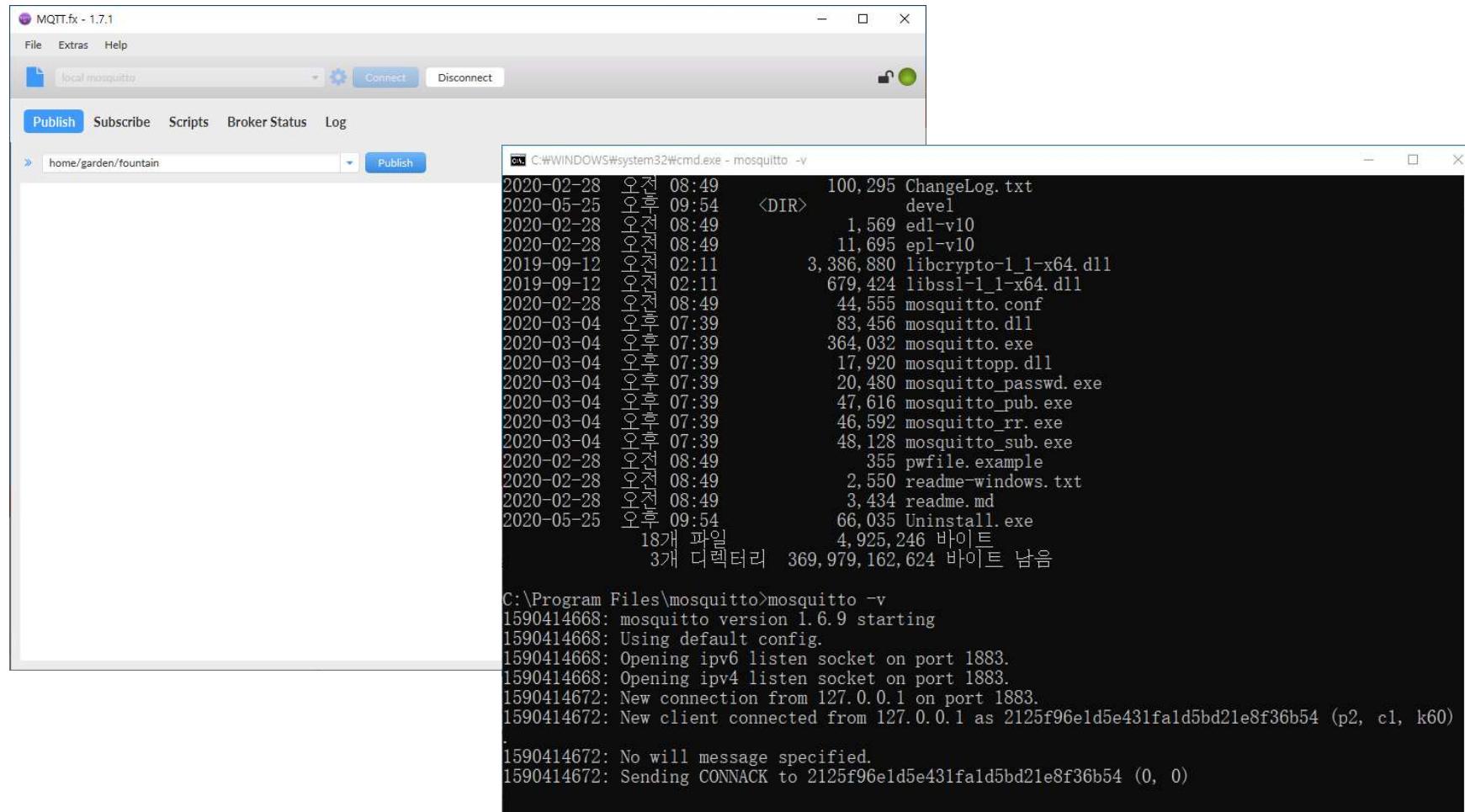
► MQTT.fx (mqqtfx.org)





❖ MQTT Client

▶ MQTT.fx (mqttfx.org)





❖ MQTT Client

► MQTT.fx (mqqtfx.org)

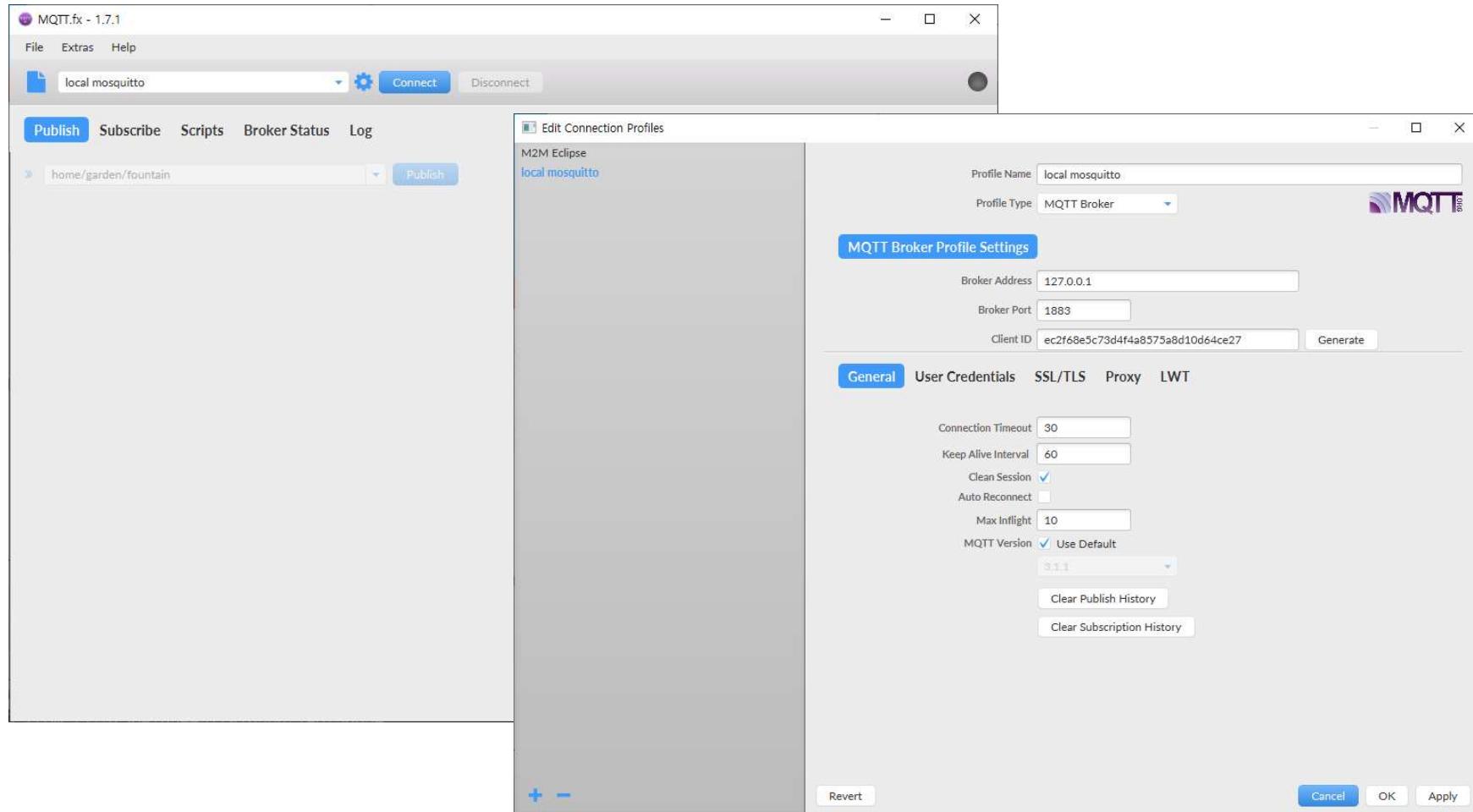
The screenshot shows the MQTT.fx client version 1.7.1. The main interface has tabs for Publish, Subscribe, Scripts, Broker Status, and Log. The Subscribe tab is selected, showing a subscription to the topic "home/garden/fountain". The Broker Status tab shows a connection to "local mosquitto" with 0 messages. The Log tab displays the following MQTT protocol messages:

```
C:\#WINDOWS#system32#cmd.exe - mosquitto -v
1590414852: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590414852: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590414912: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590414912: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590414972: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590414972: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590415032: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590415032: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590415092: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590415092: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590415152: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590415152: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590415212: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590415212: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590415272: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590415272: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590415332: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590415332: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590415392: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590415392: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590415452: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590415452: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590415512: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590415512: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590415572: Received PINGREQ from 2125f96e1d5e431fa1d5bd21e8f36b54
1590415572: Sending PINGRESP to 2125f96e1d5e431fa1d5bd21e8f36b54
1590415629: Received SUBSCRIBE from 2125f96e1d5e431fa1d5bd21e8f36b54
1590415629:    home/garden/fountain (QoS 0)
1590415629: 2125f96e1d5e431fa1d5bd21e8f36b54 0 home/garden/fountain
1590415629: Sending SUBACK to 2125f96e1d5e431fa1d5bd21e8f36b54
```



❖ MQTT Client

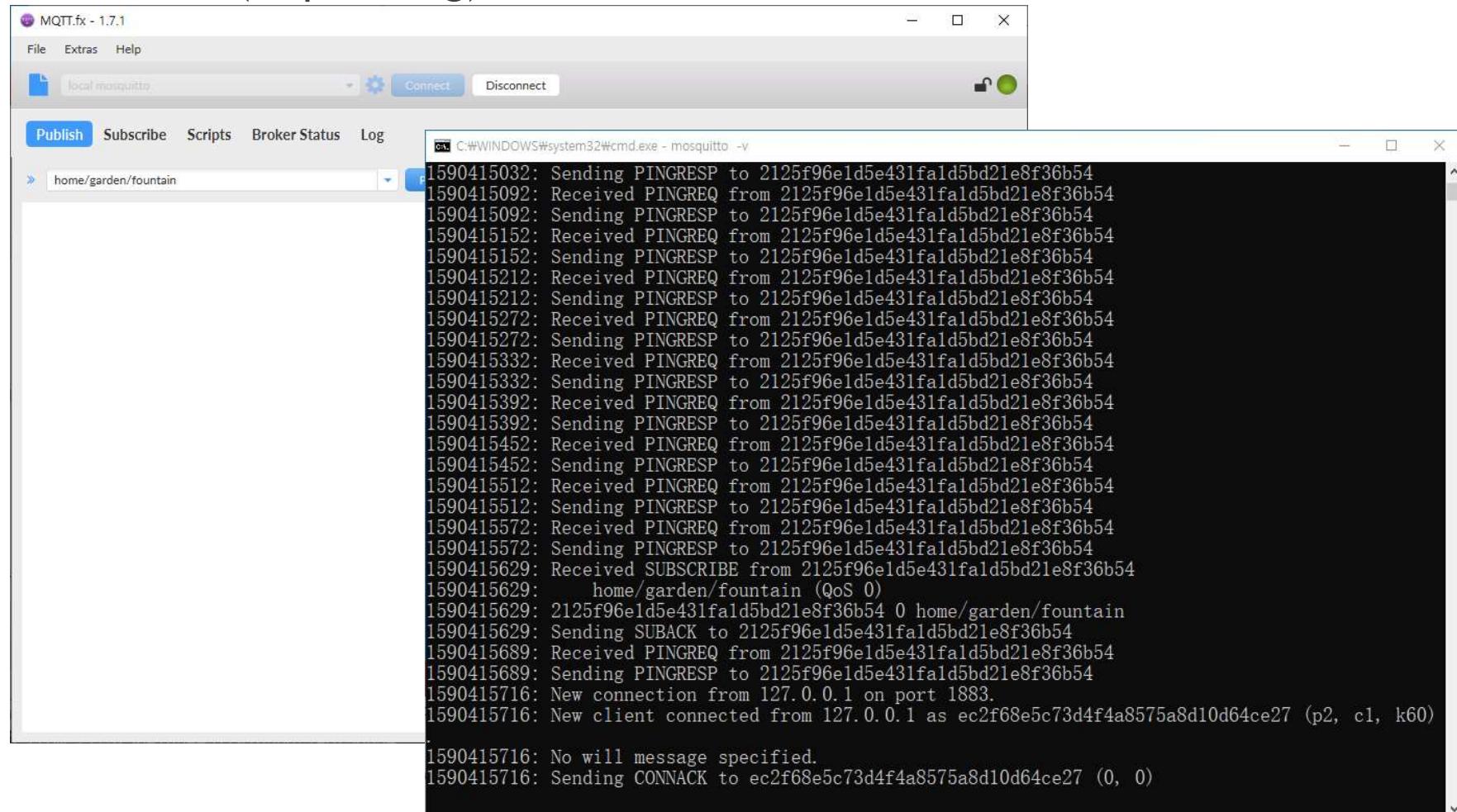
► MQTT.fx (mqttfx.org)





❖ MQTT Client

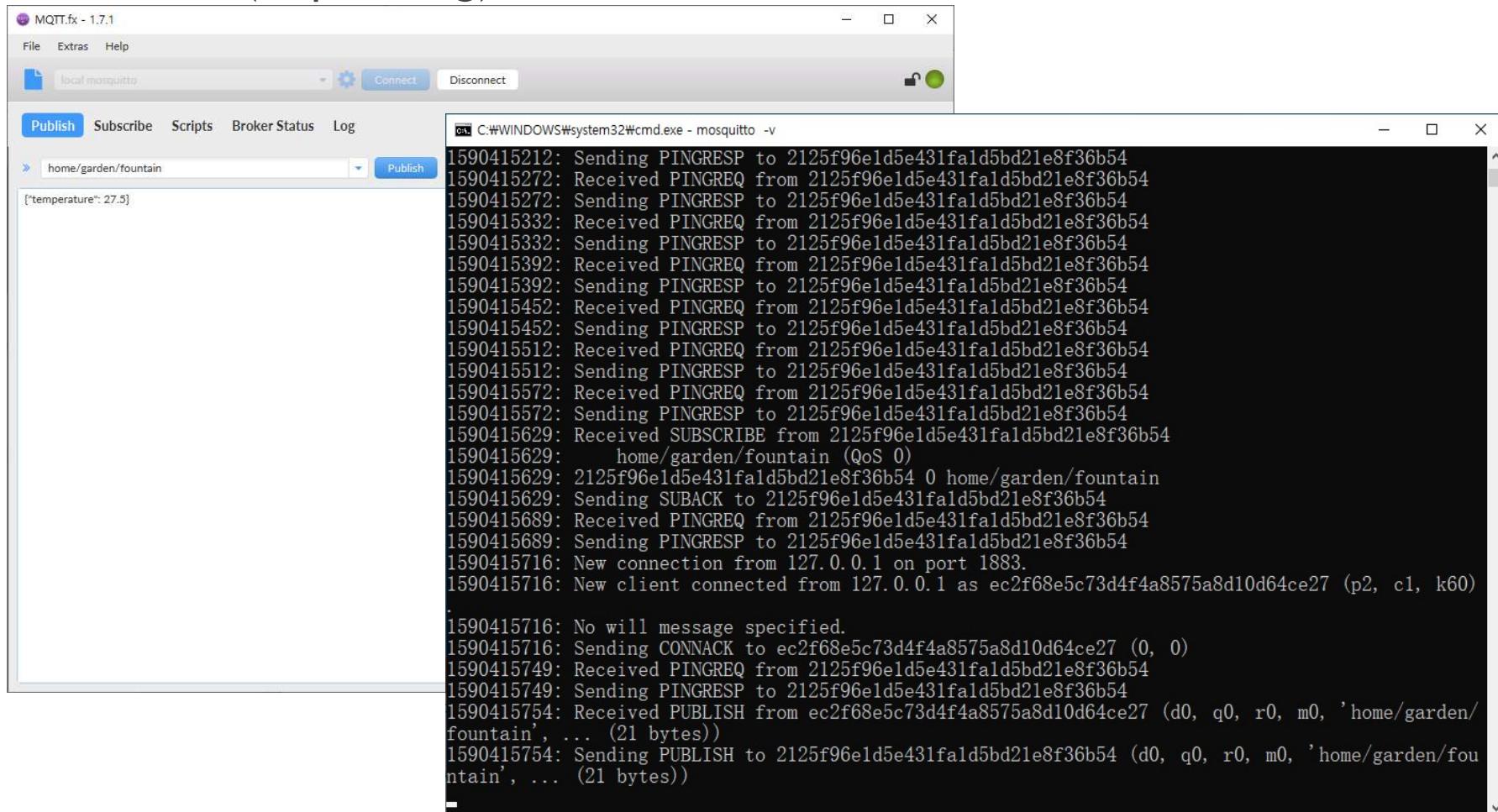
► MQTT.fx (mqttfx.org)





❖ MQTT Client

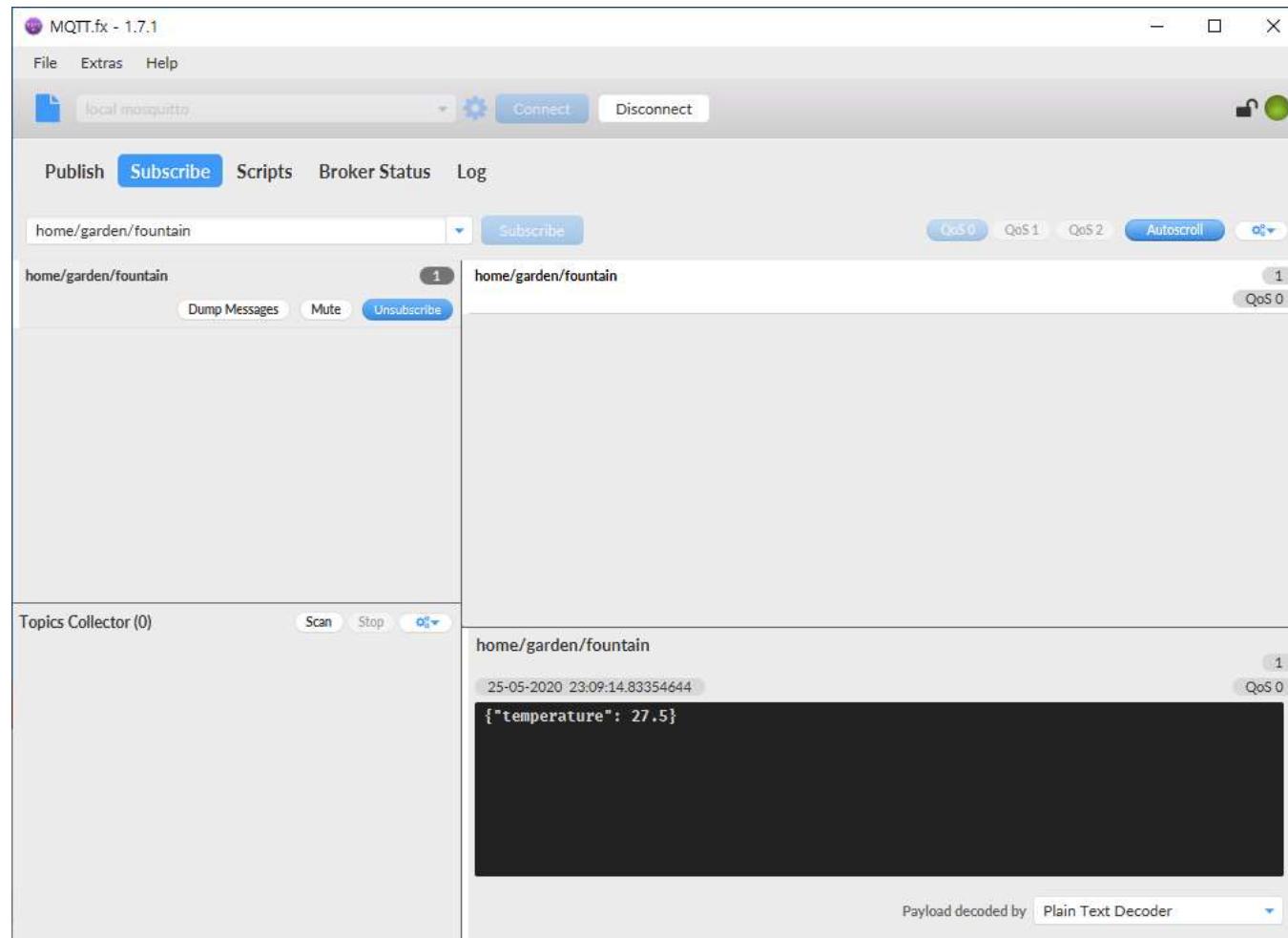
► MQTT.fx (mqttfx.org)





❖ MQTT Client

► MQTT.fx (mqttfx.org)



Device Registration



Device Registration

▶ AWS IoT Console

↳ <https://console.aws.amazon.com/iot>

The screenshot shows the AWS IoT Console interface. The left sidebar has a tree view with 'AWS IoT' selected under '관리' (Management). Other options include '모니터링' (Monitoring), '온보드' (Onboarding), '유형' (Thing Types), '사물 그룹' (Thing Groups), '결제 그룹' (Billing Groups), '작업' (Jobs), 'Greengrass', '보안', '보호', '액트', '테스트', '소프트웨어', '설정', and '알아보기'. The main content area is titled '사물' (Things) and shows three things: 'library_window' (유형 없음), 'window' (유형 없음), and 'mood-lamp' (유형 없음). Each thing card has a '...' button. A search bar at the top says '사물 검색'. On the right, there are buttons for '생성' (Create), '카드' (Card), and other icons. The bottom navigation bar includes '의견' (Feedback), '한국어' (Korean), and links to '© 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved.' and '개인 정보 보호 정책' (Privacy Policy).

Device Registration

▶ 디바이스 생성

The screenshot shows the AWS IoT console interface. On the left, a sidebar menu is visible with various options: 모니터링 (Monitoring), **온보드** (Onboarding) (circled in red), 관리 (Management), Greengrass, 보안 (Security), 보호 (Protection), 액트 (Actions), 테스트 (Testing), 소프트웨어 (Software), 설정 (Settings), and 알아보기 (Learn More). The main content area has a title "AWS IoT에 연결" (Connect to AWS IoT) and two cards: "디바이스 구성" (Device Configuration) and "AWS IoT 스타터 키트" (AWS IoT Starter Kit). The "디바이스 구성" card features an illustration of a car, a windmill, and a thermometer, with the text: "AWS IoT 디바이스 SDK를 사용하여 디바이스 또는 컴퓨터를 AWS IoT에 연결합니다." (Connect your device or computer to AWS IoT using the AWS IoT Device SDK). It includes a "시작하기" (Get Started) button. The "AWS IoT 스타터 키트" card features an illustration of a box opening to reveal a key, with the text: "AWS IoT 연결과 서비스 시작용으로 만들어진 AWS IoT 스타터 키트를 찾으십시오." (Find the AWS IoT Starter Kit, which is designed for connecting to AWS IoT and getting started with services). It includes a "스타터 키트 찾기" (Find Starter Kit) button. At the bottom, there are links for " 의견" (Feedback), " 한국어" (Korean), copyright information ("© 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved."), and "개인 정보 보호 정책" (Privacy Policy) and "이용 약관" (Terms of Service).

Device Registration

▶ 디바이스 생성

The screenshot shows the 'AWS IoT에 연결' (Connect to AWS IoT) wizard. The page title is 'AWS IoT에 연결' (Connect to AWS IoT). Below it, a sub-section title is '디바이스(예: 개발 키트 또는 컴퓨터)를 AWS IoT에 연결하기 위해서는 다음 단계를 완료해야 합니다. 이 프로세스에서 수행할 작업은 다음과 같습니다.' (To connect your device (e.g., development kit or computer) to AWS IoT, you must complete the following steps. The tasks you will perform in this process are as follows.). Three numbered steps are listed:

- 1 디바이스 등록**: 사물이란 물리적 디바이스를 클라우드에서 표현하고 기록한 것입니다. 물리적 디바이스와 AWS IoT의 호환을 위해서는 사물 레코드가 필요합니다.
- 2 연결 키트 다운로드**: 연결 키트에는 다음과 같은 몇 가지 중요한 구성 요소가 포함됩니다. 보안 자격 증명, 선택한 SDK 및 샘플 프로젝트.
- 3 디바이스 구성 및 테스트**: 연결 키트를 사용하여 파일을 전송하고 스크립트를 실행하여 디바이스를 구성하고, AWS IoT에 올바르게 연결되어 있는지 테스트합니다.

At the bottom left, there is a link: 'AWS IoT의 구성 요소에 대해 자세히 알고 싶으십니까? 대화형 개요 보기'. At the bottom right, there is a large blue button with the text '시작하기' (Start), which is circled with a red dashed line.

Page footer: 의견 (Feedback), 한국어 (Korean), © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved., 개인 정보 보호 정책 (Privacy Policy), 이용 약관 (Terms of Service).

Device Registration

▶ 디바이스 생성 - 개발환경 선택

The screenshot shows the AWS IoT Device Registration wizard on a Windows operating system. The title bar indicates the page is about connecting a device. The main content asks how to connect the device, with a note to select the most compatible platform and SDK. It lists four options: Linux/OSX, Windows, Node.js, and Python. The 'Windows' option is highlighted with a red dashed box. Below it, the 'Node.js' option is also highlighted with a red dashed box. A note at the bottom states that the device must have Node.js and NPM installed and connected to port 8883 via TCP. At the bottom right is a blue 'Next Step' button.

AWS IoT에 어떻게 연결하고 계십니까?

AWS IoT 연결 방식에 가장 적합한 플랫폼과 SDK를 선택합니다.
플랫폼을 선택합니다

Linux/OSX > Windows

AWS IoT 디바이스 SDK 선택

Node.js > Python

Java >

고려해야 할 몇 가지 전제 조건:
디바이스에 Node.js 및 NPM이 설치되어 있고 8883 포트를 통해 퍼블릭 인터넷에 연결(TCP)되어 있어야 합니다.

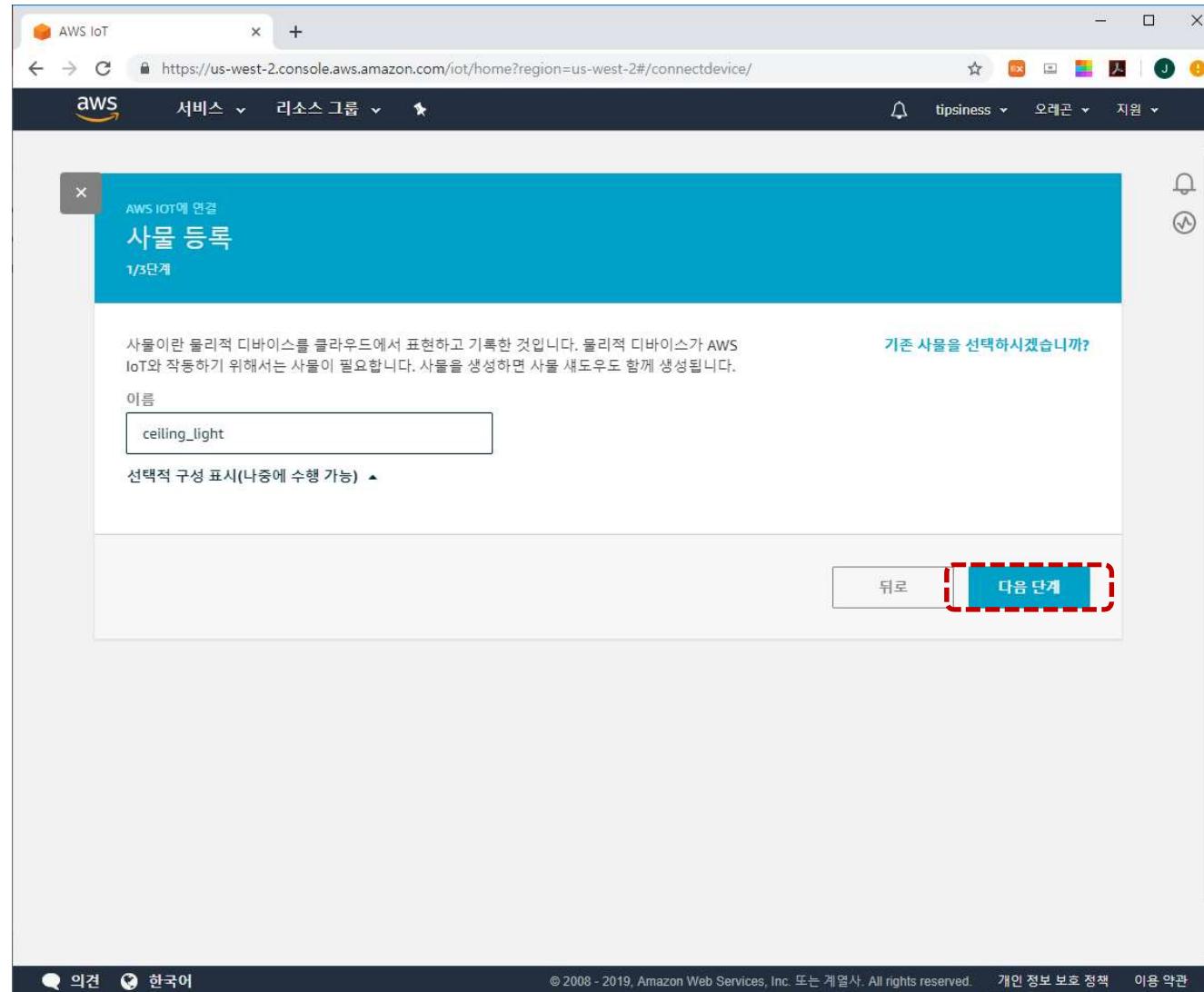
AWS IoT 디바이스 SDK와 설명서를 찾으십니까?
[AWS IoT 디바이스 SDK 보기](#)

다음

의견 한국어 © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved. 개인 정보 보호 정책 이용 약관

Device Registration

- ▶ 디바이스 생성 - 디바이스 이름 설정 (ceiling_light)



Device Registration

▶ 디바이스 생성 - 연결 키트 다운로드

The screenshot shows a browser window for AWS IoT at the URL <https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/connectdevice/>. The page title is "AWS IoT에 연결" and the sub-section title is "연결 키트 다운로드". It indicates "2/3 단계".

Below the title, it says "다음과 같은 AWS IoT 리소스가 생성됩니다." (The following AWS IoT resources will be created). A table lists:

AWS IoT 레지스트리에 있는 사 물	ceiling_light
메시지 전송 및 수신 정책	ceiling_light-Policy
	정책 미리 보기

Under "연결 키트 내용물:", there are two columns:

인증서 및 프라이빗 키	ceiling_light.cert.pem, ceiling_light.private.key
AWS IoT 디바이스 SDK	Node.js SDK
메시지 전송 및 수신 스크립트	start.ps1

A note below states: "디바이스를 연결하고 메시지를 게시하기 전에 연결 키트를 다운로드해야 합니다." (You must download the connection kit before connecting the device and publishing a message).

The "연결 키트 다운로드" button is highlighted with a red dashed box. Below it, a "Windows" button is also highlighted with a red dashed box.

At the bottom, there are "뒤로" (Back) and "다음 단계" (Next Step) buttons. The "다음 단계" button is also highlighted with a red dashed box.

Page footer: © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved. 개인 정보 보호 정책 이용 약관

Device Registration

▶ 디바이스 생성 - PowerShell 사용법

The screenshot shows a step-by-step guide for device configuration and testing in the AWS IoT console. The current step is '3/3 단계' (Step 3 of 3), titled '디바이스 구성 및 테스트' (Device Configuration and Test). The steps are as follows:

- 1단계: 디바이스에서 연결 키트의 압축 파일 해제
unzip connect_device_package.zip
- 2단계: 실행 권한 추가
Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process
- 3단계: 시작 스크립트 실행. 사물이 보내는 메시지는 아래와 같을 것입니다.
.\\start.ps1

At the bottom right, there are '뒤로' (Back) and '완료' (Complete) buttons. The '완료' button is highlighted with a red dashed border.

Page footer: 의견 (Feedback), 한국어 (Korean), © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved., 개인 정보 보호 정책 (Privacy Policy), 이용 약관 (Terms of Service)

Device Registration

▶ 디바이스 생성 완료

The screenshot shows the AWS IoT Device Registration successful completion page. The title bar says "연결 완료" (Connection Complete). The main message is "AWS IoT 및 디바이스에서 몇몇 작업을 수행하여 디바이스를 AWS IoT에 연결했습니다." (Several tasks were performed between AWS IoT and the device to connect the device to AWS IoT). Below this, there are four items with icons and "자세히 알아보기" (Learn more) links:

- 사물을 등록하여 AWS IoT에서 디바이스를 표현 (Register a thing to represent the device in AWS IoT)
- 인증서 및 정책을 사용하여 디바이스의 보안 설정 (Use certificates and policies to set up device security)
- 디바이스 SDK를 사용하여 디바이스를 AWS IoT에 연결 (Connect the device to AWS IoT using the Device SDK)
- 디바이스로부터 메시지 수신 (Receive messages from the device)

At the bottom, there is a question "AWS IoT의 구성 요소에 대해 자세히 알고 싶으십니까?" (Do you want to know more about the components of AWS IoT?) with a "대화형 개요 보기" (View detailed overview) link, and a large blue button labeled "완료" (Complete) with a red dashed border.

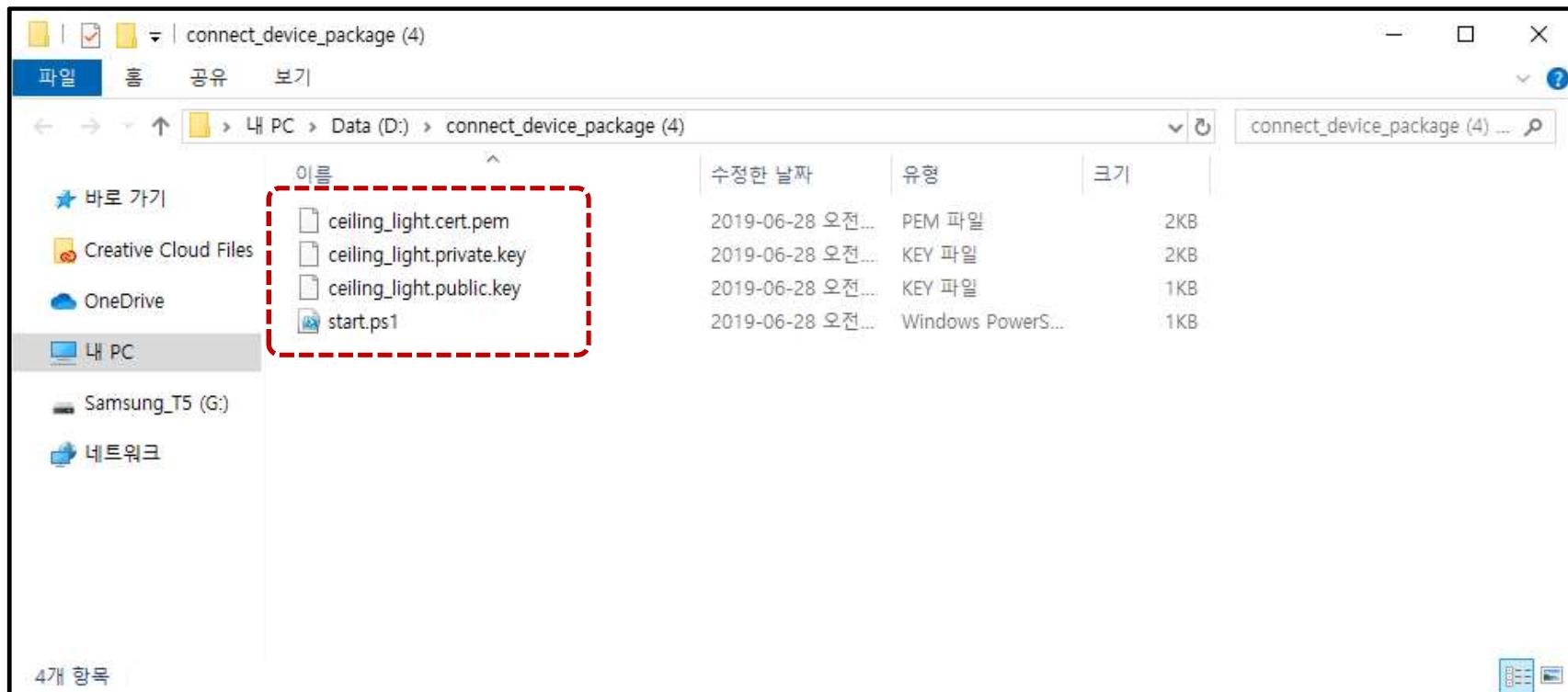
Device Registration

▶ REST의 정의

The screenshot shows the AWS IoT Device Management console interface. On the left, there's a sidebar with various navigation options: 모니터링, 온보드, 관리 (selected), 사물, 유형, 사물 그룹, 결제 그룹, 작업, Greengrass, 보안, 보호, 액트, 테스트, 소프트웨어, 설정, 알아보기. The main area is titled '사물' (Things) and displays a list of registered devices. Each device entry includes the name, model, and three dots for more options. The device 'ceiling_light' is highlighted with a red dashed rectangle. The URL in the browser is https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/thinghub.

Device Registration

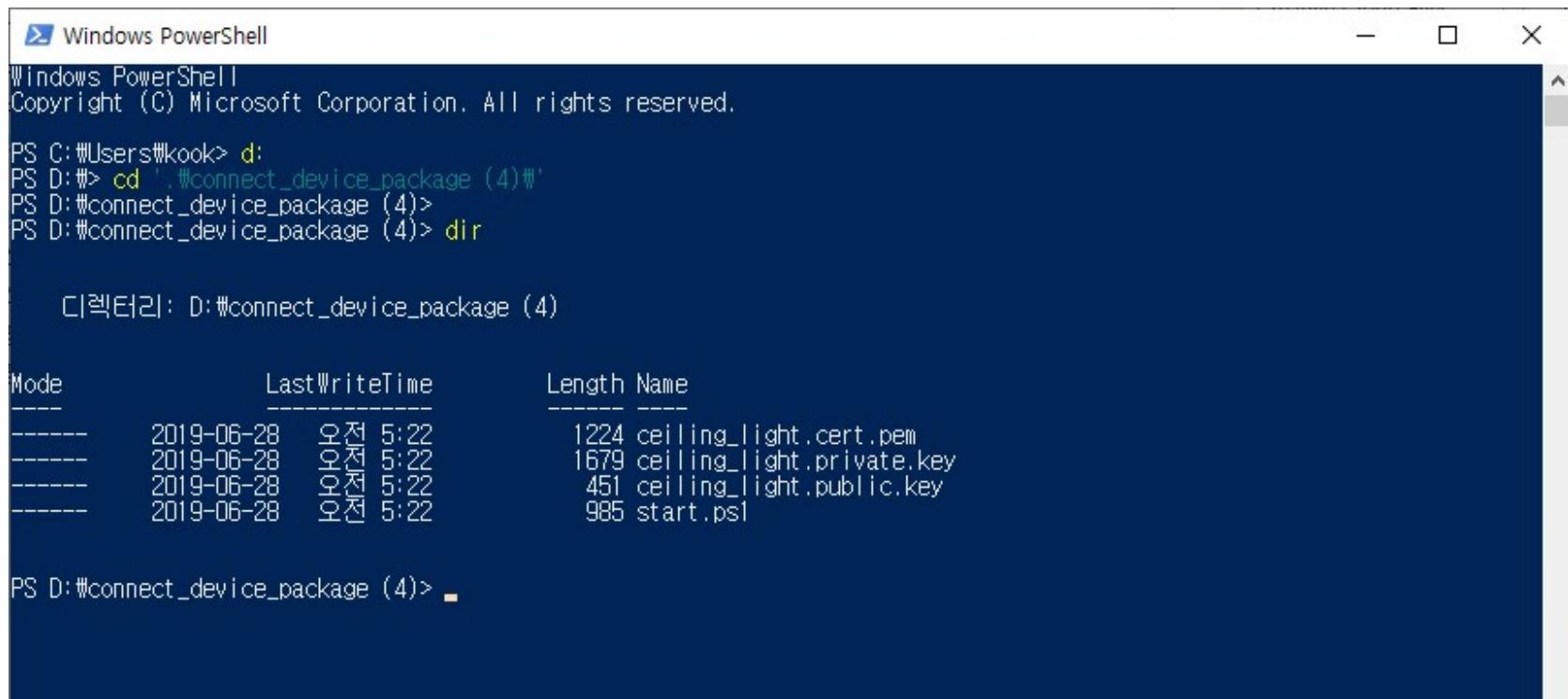
▶ 연결 키트 구성



Device Registration

- ▶ 연결 키트 - PowerShell 실행

↳ 경로 이동



The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command history and output are as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\kook> d:
PS D:\> cd 'D:\connect_device_package (4)'
PS D:\connect_device_package (4)>
PS D:\connect_device_package (4)> dir

디렉터리: D:\connect_device_package (4)

Mode                LastWriteTime       Length Name
----                -----        ----
----- 2019-06-28  오전 5:22      1224 ceiling_light.cert.pem
----- 2019-06-28  오전 5:22      1679 ceiling_light.private.key
----- 2019-06-28  오전 5:22       451 ceiling_light.public.key
----- 2019-06-28  오전 5:22      985 start.ps1

PS D:\connect_device_package (4)>
```

Device Registration

▶ 연결 키트 - PowerShell 실행

↳ 실행 규칙 변경

> ***Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process***

The screenshot shows a Windows PowerShell window with the following content:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\kook> d:
PS D:> cd .\connect_device_package (4)
PS D:\connect_device_package (4)>
PS D:\connect_device_package (4)> dir

디렉터리: D:\connect_device_package (4)

Mode                LastWriteTime         Length Name
----                -----        1224 ceiling_light.cert.pem
----                2019-06-28 오전 5:22      1679 ceiling_light.private.key
----                2019-06-28 오전 5:22      451 ceiling_light.public.key
----                2019-06-28 오전 5:22      985 start.ps1

PS D:\connect_device_package (4)> Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process

실행 규칙 변경
실행 정책은 진회하지 않는 스크립트로부터 사용자를 보호합니다. 실행 정책을 변경하면 about_Execution_Policies 도움말
항목(https://go.microsoft.com/fwlink/?LinkID=135170)에 설명된 보안 위험에 노출될 수 있습니다. 실행 정책을
변경하시겠습니까?
[Y] 예(Y) [A] 모두 예(A) [N] 아니요(N) [L] 모두 아니요(L) [S] 일시 중단(S) [?] 도움말 (기본값은 "N"): A
PS D:\connect_device_package (4)>
PS D:\connect_device_package (4)>
```

Device Registration

▶ 연결 키트 - PowerShell 실행

↳ start.ps1 스크립트 실행

> .\start.ps1

```
PS D:\#connect_device_package (4)> Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process
실행 규칙 변경
실행 정책은 진로하지 않는 스크립트로부터 사용자를 보호합니다. 실행 정책을 변경하면 about_Execution_Policies 도움말
항목(https://go.microsoft.com/fwlink/?LinkID=135170)에 설명된 보안 위험에 노출될 수 있습니다. 실행 정책을
변경하시겠습니까?
[Y] 예(Y) [A] 모두 예(A) [N] 아니요(N) [L] 모두 아니요(L) [S] 일시 중단(S) [?] 도움말 (기본값은 "N"): A
PS D:\#connect_device_package (4)>
PS D:\#connect_device_package (4)> .\start.ps1
Downloading AWS IoT Root CA certificate from AWS...
Installing AWS SDK...
npm WARN ENOENT: no such file or directory, open 'D:\#connect_device_package (4)\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN ENOENT: no such file or directory, open 'D:\#connect_device_package (4)\package.json'
npm WARN connect_device_package (4) No description
npm WARN connect_device_package (4) No repository field.
npm WARN connect_device_package (4) No README data
npm WARN connect_device_package (4) No license field.

+ aws-iot-device-sdk@2.2.1
added 62 packages from 48 contributors and audited 243 packages in 3.321s
found 0 vulnerabilities
```

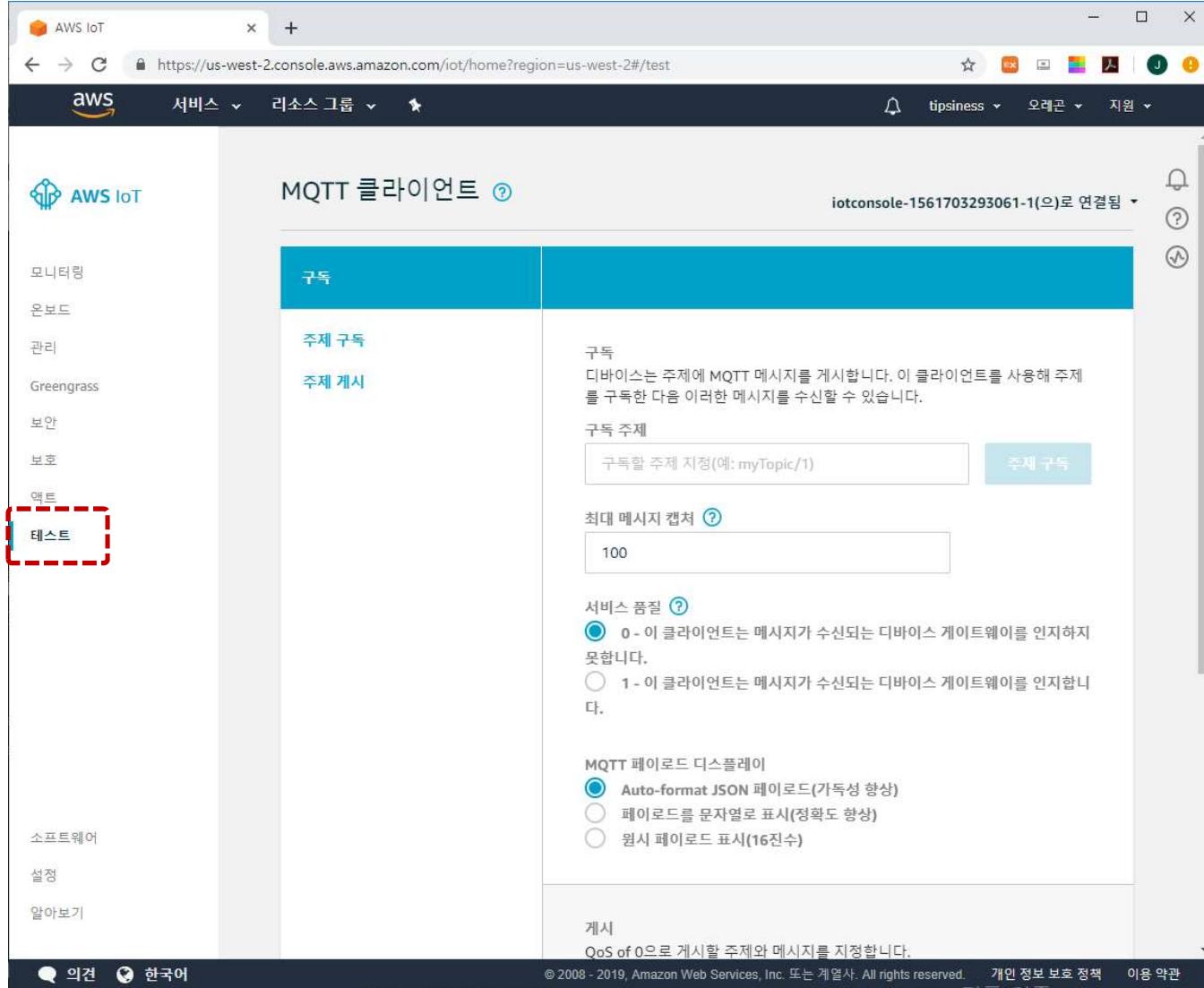
Running pub/sub sample application...

connect

-

Device Registration

▶ 메시지 수신 테스트 (MQTT)



The screenshot shows the AWS IoT MQTT Client interface. On the left, a sidebar menu lists various options: 모니터링, 온보드, 관리, Greengrass, 보안, 보호, 액트 (which is highlighted with a red dashed box), and 테스트. The main content area is titled "MQTT 클라이언트" and shows a "Subscribe" tab. It includes fields for "Subscribe Topic" (구독 주제) set to "myTopic/1", "Max Message Capture" (최대 메시지 캡처) set to 100, and "Service Quality" (서비스 품질) set to 0 (선택). Below these are "MQTT Payload Display" (MQTT 페이로드 디스플레이) settings: Auto-format JSON (선택), Text (선택), and Hex (선택). At the bottom, there is a "Publish" (게시) button and a note about QoS settings.

Device Registration

▶ 메시지 수신 테스트 (MQTT)

The screenshot shows the AWS IoT Test page at <https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/test>. The left sidebar menu includes 모니터링, 온보드, 관리, Greengrass, 보안, 보호, 액트, 테스트, 소프트웨어, 설정, and 알아보기. The main area displays MQTT 페이로드 디스플레이 settings: 서비스 품질 (0 - 이 클라이언트는 메시지가 수신되는 디바이스 게이트웨이를 인지하지 못합니다. 1 - 이 클라이언트는 메시지가 수신되는 디바이스 게이트웨이를 인지합니다.), MQTT 페이로드 디스플레이 (Auto-format JSON 페이로드(가독성 향상), 페이로드를 문자열로 표시(정확도 향상), 원시 페이로드 표시(16진수)), and a message topic input field containing "topic_1". A red box highlights the message payload area, which contains the JSON object:

```
{  
  "message": "PowerOn"  
}
```

At the bottom, there are links for 의견 (Feedback), 한국어 (Korean), © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved., 개인 정보 보호 정책 (Privacy Policy), and 이용 약관 (Terms of Service). The page is in Korean.

Device Registration

▶ 메시지 수신 테스트 (MQTT)

↳ MQTT 메시지 수신 확인

```
PS D:\#connect_device_package (4)> Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process
실행 규칙 변경
실행 정책은 진로하지 않는 스크립트로부터 사용자를 보호합니다. 실행 정책을 변경하면 about_Execution_Policies 도움말
항목(https://go.microsoft.com/fwlink/?LinkId=135170)에 설명된 보안 위험에 노출될 수 있습니다. 실행 정책을
변경하시겠습니까?
[Y] 예(Y) [A] 모두 예(A) [N] 아니요(N) [L] 모두 아니요(L) [S] 일시 중단(S) [?] 도움말 (기본값은 "N"): A
PS D:\#connect_device_package (4)>
PS D:\#connect_device_package (4)> .\start.ps1
Downloading AWS IoT Root CA certificate from AWS...
Installing AWS SDK...
npm WARN ENOENT: no such file or directory, open 'D:\#connect_device_package (4)\package.json'
npm WARN created a lockfile as package-lock.json. You should commit this file.
npm WARN ENOENT: no such file or directory, open 'D:\#connect_device_package (4)\package.json'
npm WARN connect_device_package (4) No description
npm WARN connect_device_package (4) No repository field.
npm WARN connect_device_package (4) No README data
npm WARN connect_device_package (4) No license field.

+ aws-iot-device-sdk@2.2.1
added 62 packages from 48 contributors and audited 243 packages in 3.321s
found 0 vulnerabilities

Running pub/sub sample application...
connect
message topic_1 {
  "message": "PowerOn"
}
```

Device Registration

▶ 메시지 수신 테스트 (MQTT)

The screenshot shows the AWS IoT Test page at <https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/test>. The left sidebar menu includes AWS IoT, 모니터링, 온보드, 관리, Greengrass, 보안, 보호, 액트, 테스트, 소프트웨어, 설정, and 알아보기. The main content area has sections for 서비스 품질 (with radio button 0 selected), MQTT 페이로드 디스플레이 (with radio button Auto-format JSON 페이로드(가독성 향상) selected), and a Publish section. In the Publish section, the topic is set to "topic_1" and the message content is highlighted with a red box:

```
{  
  "message": "PowerOff"  
}
```

The bottom footer includes links for 의견 (Feedback), 한국어 (Korean), © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved., 개인 정보 보호 정책 (Privacy Policy), and 이용 약관 (Terms of Service).



Device Registration

▶ 메시지 수신 테스트 (MQTT)

↳ MQTT 메시지 수신 확인

```
+ aws-iot-device-sdk@2.2.1
added 62 packages from 48 contributors and audited 243 packages in 3.321s
found 0 vulnerabilities

Running pub/sub sample application...
connect
message topic_1 {
  "message": "PowerOn"
}
message topic_1 {
  "message": "PowerOff"
}
```

SNS: Simple Notification Service





▶ SNS: Simple Notification Service

The screenshot shows the AWS IoT console interface. On the left, there's a sidebar with categories like '내역', 'IoT Core', 'Simple Notification Service', and '콘솔 흡'. The main area has a search bar at the top containing the text 'simple notification service'. Below the search bar, the results list includes 'Simple Notification Service' (highlighted with a red dashed box) and other services like EC2, AWS RoboMaker, Athena, etc. The results are grouped into sections: 'AWS 서비스', 'AWS 기반 서비스', 'AWS 기기 및 디바이스', 'AWS 기관 및 정부', 'AWS 보안 및 규정 준수', and 'AWS 사물인터넷'. At the bottom, there are links for ' 의견', ' 한국어 ', ' © 2008 - 2019, Amazon Web Services, Inc. 모든 게임사. All rights reserved.', ' 개인 정보 보호 정책', and ' 이용 약관'.



▶ SNS 주제

The screenshot shows the AWS Simple Notification Service (SNS) console. The left sidebar has a tree view with 'Amazon SNS' selected. Under 'Amazon SNS', the '주제' (Topic) option is highlighted in orange. The main content area is titled 'Amazon SNS > 주제'. It displays a table with one row labeled '주제 없음'. Below the table, there is a message: '시작하려면 주제를 생성하십시오.' followed by a large orange button labeled '주제 생성'. This 'Create Topic' button is highlighted with a red dashed rectangle.

Simple Notification Service

https://us-west-2.console.aws.amazon.com/sns/v3/home?region=us-west-2#/topics

aws 서비스 리소스 그룹

Amazon SNS

대시보드

주제

구독

Mobile

푸시 알림

문자 메시지(SMS)

Amazon SNS > 주제

주제 (0)

편집 삭제 메시지 게시 주제 생성

검색

이름 ARN

주제 없음

시작하려면 주제를 생성하십시오.

주제 생성

의견 한국어

© 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved. 개인 정보 보호 정책 이용 약관



▶ SNS 주제

The screenshot shows the AWS SNS 'Create Topic' page. The top navigation bar includes the AWS logo, service and resource group dropdowns, and user information like 'tipsiness' and 'Oregon'. The main title is 'Amazon SNS > 주제 > 주제 생성' (Topic Creation). The first step, '제작 정보' (Topic Information), is active. It has a red dashed box around the 'Name' input field, which contains 'MySNSTopic'. Below it is a note: '최대 256자이며 영숫자, 하이픈(-) 및 밑줄(_)을 포함할 수 있습니다.' (Up to 256 characters, including numbers, hyphens, and underscores). The 'Display Name - 선택 사항' (Optional) section contains a note: '이 주제를 SMS 구독과 함께 사용하려면 표시 이름을 입력하십시오. 처음 10자만 SMS 메시지에 표시됩니다.' (If you want to use this topic with an SMS subscription, enter a display name. Only the first 10 characters will be displayed in the SMS message.) and a text input field with placeholder text '내 주제'. The '암호화 - 선택 사항' (Optional) section notes: 'Amazon SNS는 기본적으로 전송 중 암호화를 제공합니다. 서버 측 암호화를 활성화하면 주제에 유튜 시 암호화가 추가됩니다.' (Amazon SNS provides encryption by default during transmission. Enabling server-side encryption adds encryption to the topic when it is published). The bottom navigation bar includes links for 'Opinion' (意见), 'Language' (한국어), and copyright information: '© 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved.' and links to 'Privacy Policy' (개인 정보 보호 정책) and 'Terms of Service' (이용 약관).



▶ SNS 주제

Simple Notification Service

aws 서비스 리소스 그룹

tipsiness 오레곤 지원

암호화 - 선택 사항

Amazon SNS는 기본적으로 전송 중 암호화를 제공합니다. 서버 측 암호화를 활성화하면 주제에 유포 시 암호화가 추가됩니다.

액세스 정책 - 선택 사항

이 정책은 주제에 액세스할 수 있는 사용자를 정의합니다. 기본적으로 주제 소유자만 주제에 게시 또는 구독할 수 있습니다. [정보](#)

전송 재시도 정책(HTTP/S) - 선택 사항

정책은 Amazon SNS가 HTTP/S 엔드포인트에 대해 실패한 전송을 재시도하는 방법을 정의합니다. 기본 설정을 수정하려면 이 섹션을 확장하십시오. [정보](#)

전송 상태 로깅 - 선택 사항

이러한 설정은 CloudWatch Logs의 메시지 전송 상태 로깅을 구성합니다. [정보](#)

태그 - 선택 사항

태그는 Amazon SNS 주제에 할당할 수 있는 메타데이터 레이블입니다. 각 태그는 키와 선택 사항인 값으로 이루어집니다. 태그를 사용하여 주제를 검색 및 필터링하고 비용을 추적할 수 있습니다. [자세히 알아보기](#)

취소 **주제 생성**

의견 한국어 © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved. 개인 정보 보호 정책 이용 약관



▶ SNS 구독 생성

The screenshot shows the AWS SNS console for creating a new subscription (구독). The main interface includes:

- Left Sidebar:** Includes links for 대시보드 (Dashboard), 주제 (Topics), 구독 (Subscriptions), and Mobile (Push Notifications and SMS).
- Topic Overview:** Shows the topic name **MySNSTopic** and three buttons: 편집 (Edit), 삭제 (Delete), and 메시지 게시 (Post Message).
- Subscription Details:** A table showing the subscription information:

이름 (Name)	표시 이름 (Display Name)
MySNSTopic	-
ARN (Amazon Resource Name)	주제 소유자 (Topic Owner)
arn:aws:sns:us-west-2:552924431232:MySNSTopic	552924431232
- Subscription Actions:** A row of buttons: 구독 (Subscribe) (highlighted with a red dashed box), 액세스 정책 (Access Policy), 전송 재시도 정책(HTTP/S) (Delivery Retry Policy), 전송 상태 로깅 (Delivery Status Logging), and 암호화 (Encryption).
- Bottom Buttons:** 구독 (0) (Subscriptions (0)), 편집 (Edit), 삭제 (Delete), 확인 요청 (Request Confirmation), 구독 확인 (Confirm Subscription), and 구독 생성 (Create Subscription).



▶ 구독 생성 - SMS

The screenshot shows the AWS Simple Notification Service (SNS) console. The URL in the browser is <https://us-west-2.console.aws.amazon.com/sns/v3/home?region=us-west-2#/create-subscription>. The page title is "구독 생성".

세부 정보

주제 ARN:

프로토콜: **SMS** (highlighted with a red dashed box)

구독 생성 버튼

Page footer: 의견, 한국어, © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved. 개인 정보 보호 정책, 이용 약관



▶ SMS 엔드포인트 설정

The screenshot shows the 'Subscription Creation' page in the AWS SNS console. The URL in the browser is <https://us-west-2.console.aws.amazon.com/sns/v3/home?region=us-west-2#/create-subscription>. The page has a sidebar with '구독 생성' (Subscription Creation) and a main content area with a '세부 정보' (Details) section.

세부 정보

주제 ARN: arn:aws:sns:us-west-2:552924431232:MySNSTopic

프로토콜: 구독할 엔드포인트 유형
SMS

엔드포인트: Amazon SNS의 알림을 수신할 수 있는 모바일 번호입니다.
+821039051745

설명: 구독을 생성한 후에는 확인해야 합니다. [정보](#)

▶ 구독 필터 정책 - 선택 사항: 이 정책은 구독자가 받는 메시지를 필터링합니다. [정보](#)

Buttons at the bottom: 취소, 구독 생성

Page footer: 의견, 한국어, © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved., 개인 정보 보호 정책, 이용 약관



▶ 구독

Simple Notification Service AWS IoT

aws 서비스 리소스 그룹

Amazon SNS

구독: 1c1c4b5c-f1bb-4415-beb1-91a24e7118b9

구독 필터 정책

구독 필터 정책

이 정책은 구독자가 받는 메시지를 필터링합니다. [정보](#)

© 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved. 개인 정보 보호 정책 이용 약관

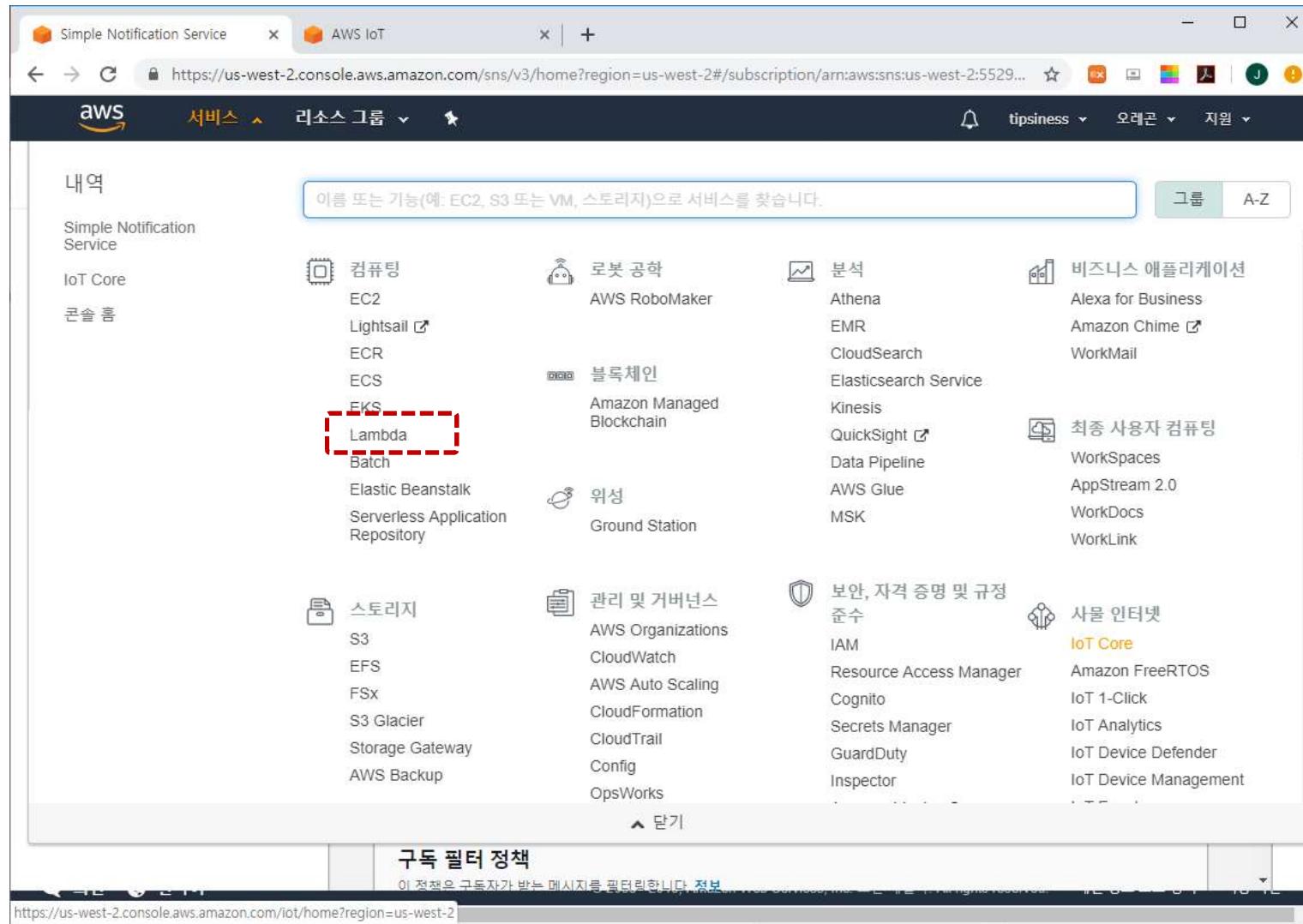
The screenshot shows the AWS SNS console with a subscription details page. The URL in the browser is https://us-west-2.console.aws.amazon.com/sns/v3/home?region=us-west-2#/subscription/arn:aws:sns:us-west-2:552924431232:MySNSTopic:1c1c4b5c-f1bb-4415-beb1-91a24e7118b9. The left sidebar shows navigation options like Dashboard, Topics, Subscriptions, and Mobile (Push Notifications, SMS). The main content area shows the subscription details for the topic 'MySNSTopic'. The ARN is arn:aws:sns:us-west-2:552924431232:MySNSTopic:1c1c4b5c-f1bb-4415-beb1-91a24e7118b9. The endpoint is +821039051745. The status is 'Confirmed' (확인됨). The protocol is 'SMS'. Below this, there's a section for 'Subscription Filter Policy' (구독 필터 정책) which states: 'This policy filters messages that the subscriber receives.' (이 정책은 구독자가 받는 메시지를 필터링합니다.) with a link to 'Information' (정보).

Lambda Function



Lambda

▶ 서비스 → Lambda





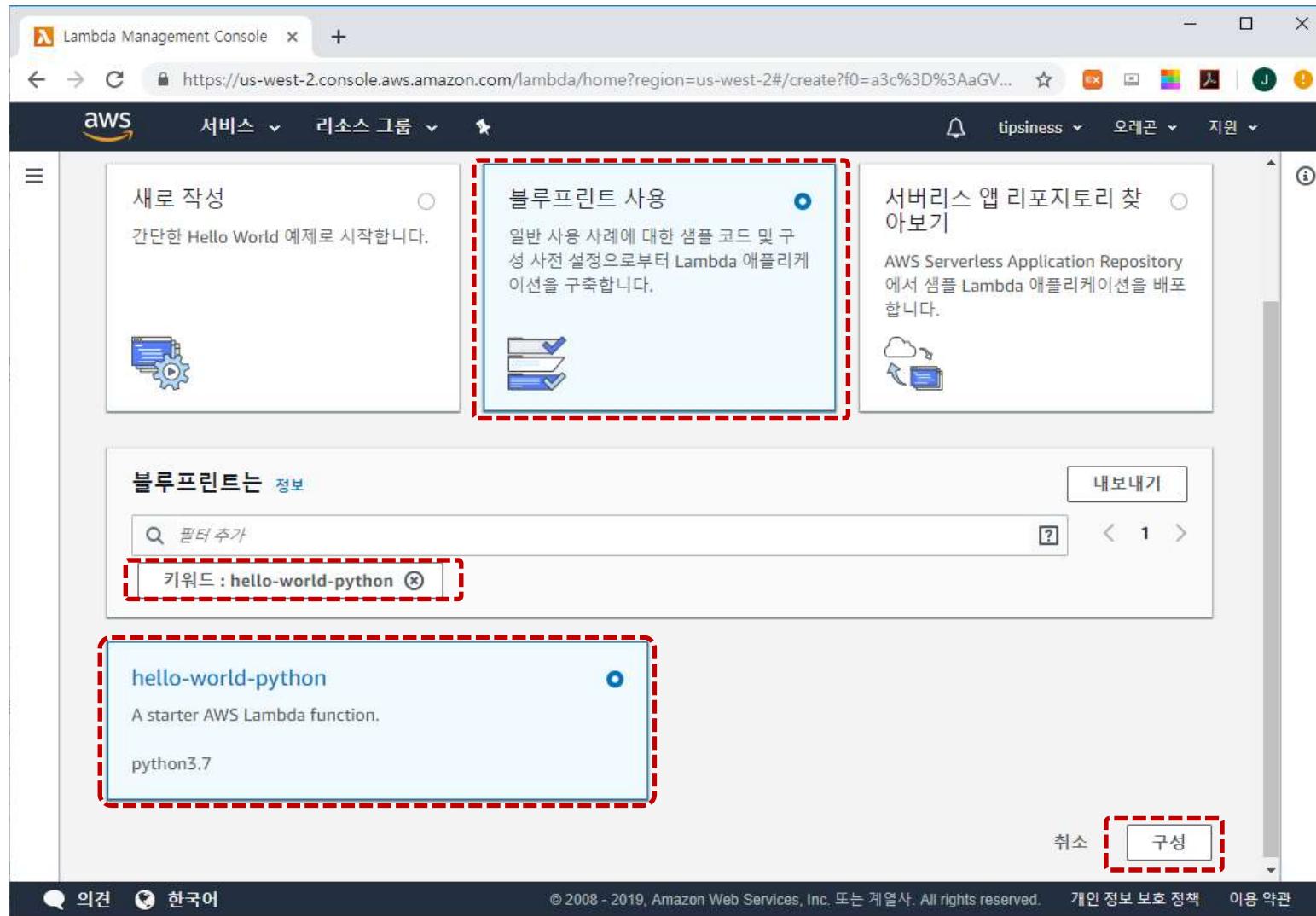
Lambda

▶ Lambda 함수 생성

The screenshot shows the AWS Lambda Management Console interface. The left sidebar has navigation links: 대시보드, 애플리케이션, **함수**, and 계층. The main content area is titled '함수 (0)' and displays a search bar with placeholder text '태그 및 속성별 필터 또는 키워드별 검색'. Below the search bar are filters for '함수 이름', '설명', '런타임', '코드 크기', and '마지막 수정'. A message at the bottom states '표시할 데이터가 없습니다.' (No data to display). At the top right of the main content area, there is a red dashed box around the '함수 생성' (Create Function) button. The browser address bar shows the URL <https://us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/functions>.

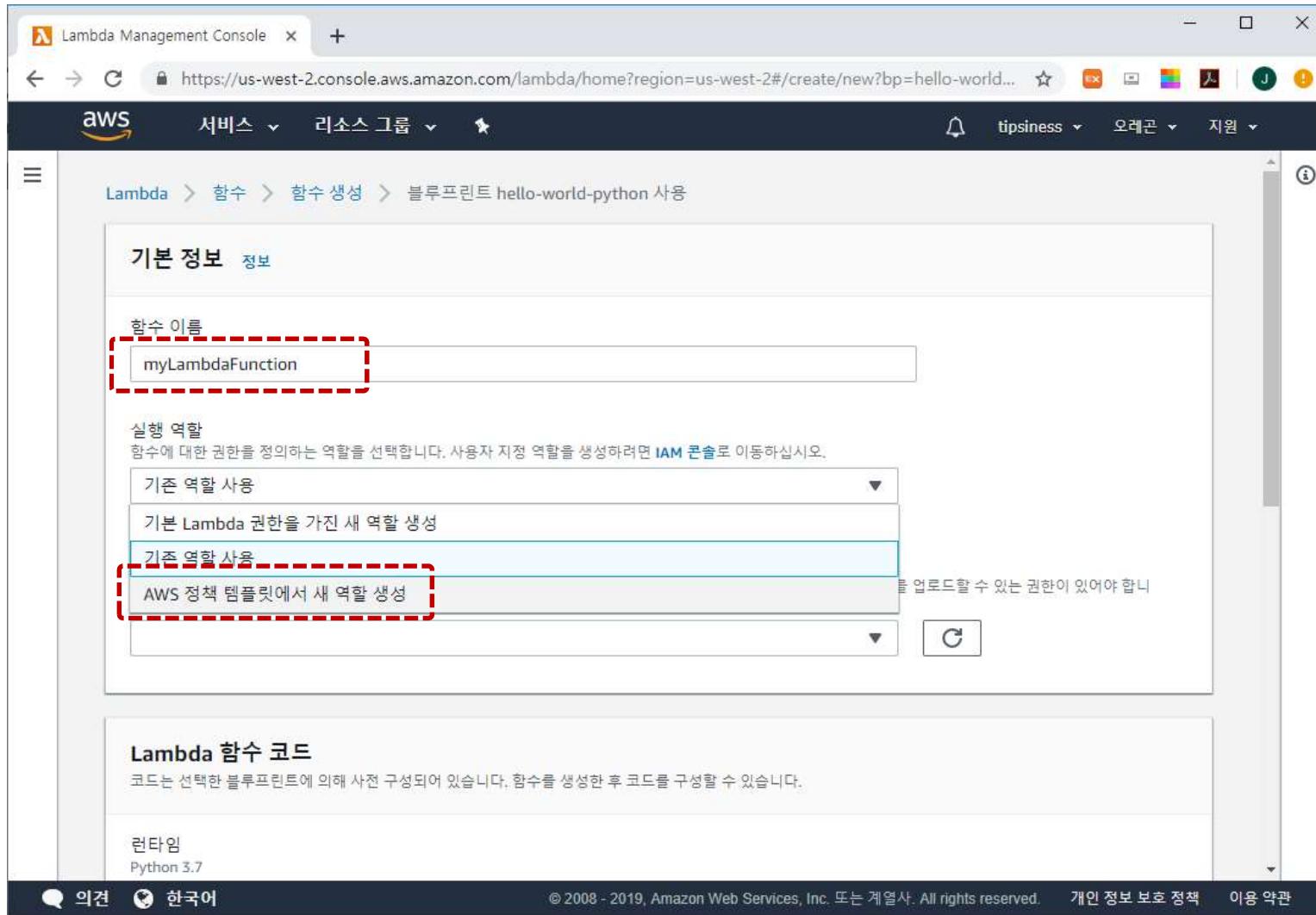
Device Registration

▶ REST의 정의



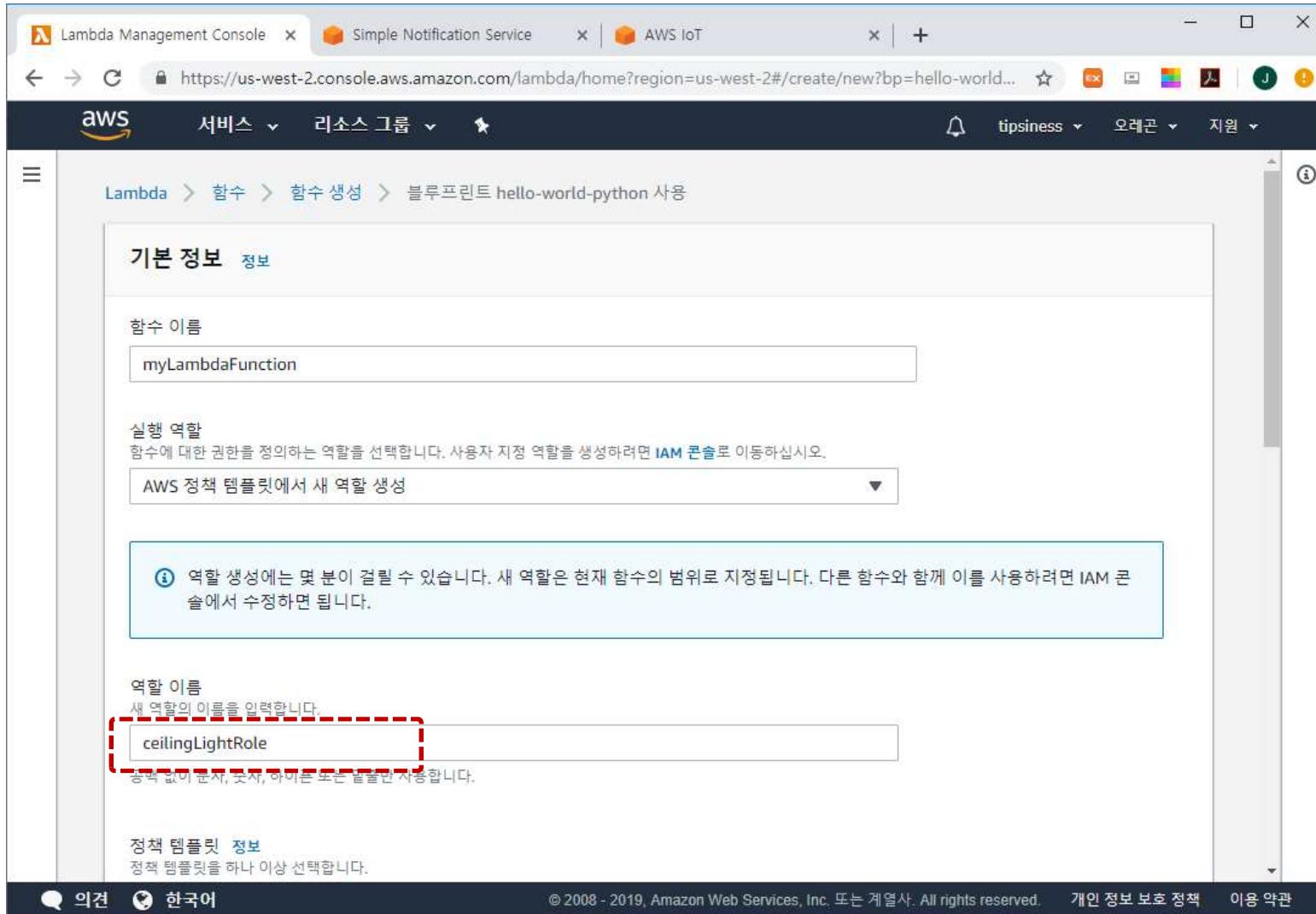
Device Registration

▶ REST의 정의



Device Registration

▶ REST의 정의



Lambda Management Console | Simple Notification Service | AWS IoT

https://us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/create/new?bp=hello-world...

aws 서비스 리소스 그룹

Lambda > 함수 > 함수 생성 > 블루프린트 hello-world-python 사용

기본 정보 정보

함수 이름
myLambdaFunction

실행 역할
함수에 대한 권한을 정의하는 역할을 선택합니다. 사용자 지정 역할을 생성하려면 IAM 콘솔로 이동하십시오.

AWS 정책 템플릿에서 새 역할 생성

① 역할 생성에는 몇 분이 걸릴 수 있습니다. 새 역할은 현재 함수의 범위로 지정됩니다. 다른 함수와 함께 이를 사용하려면 IAM 콘솔에서 수정하면 됩니다.

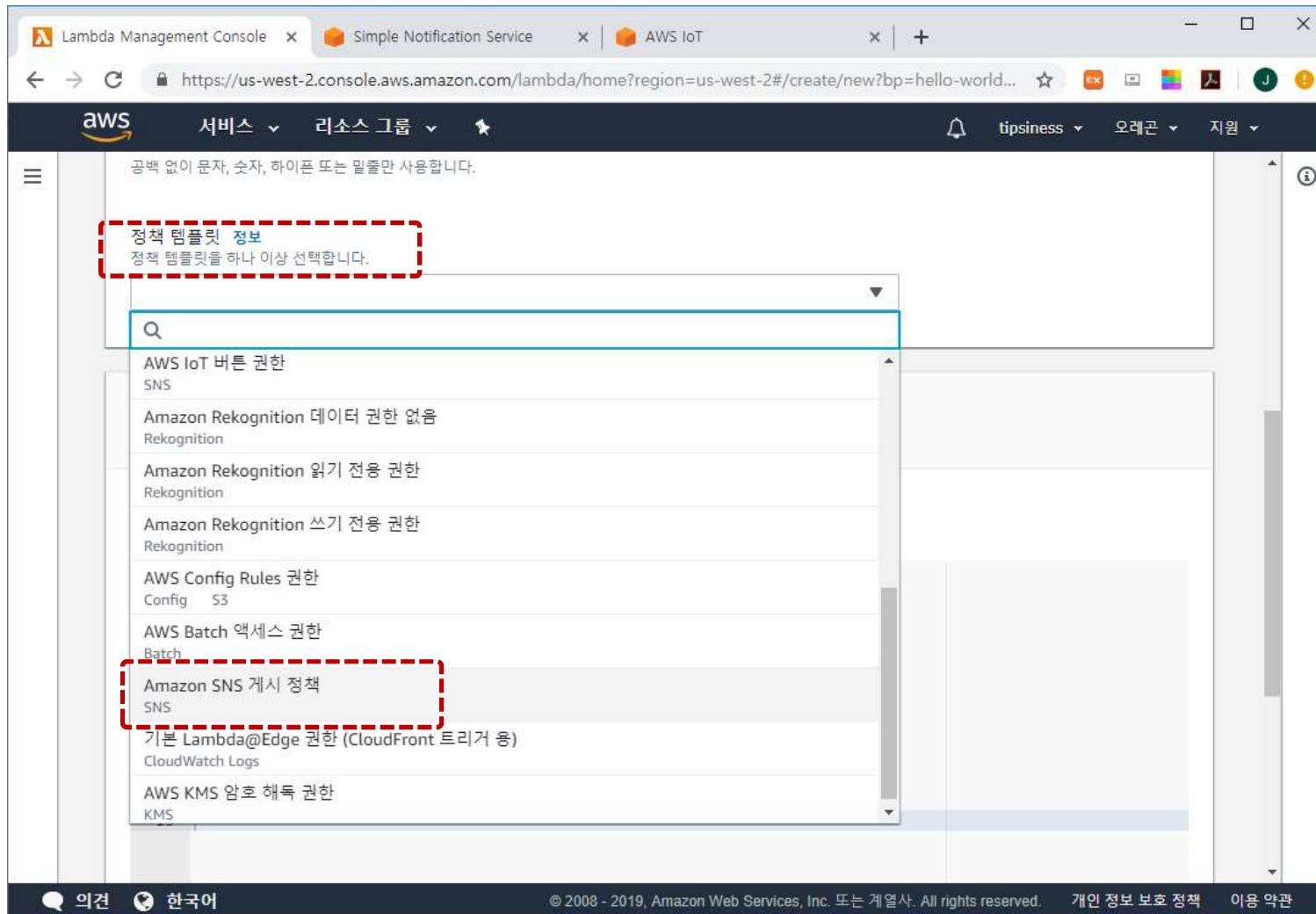
역할 이름
새 역할의 이름을 입력합니다.
ceilingLightRole

정책 템플릿 정보
정책 템플릿을 하나 이상 선택합니다.

© 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved. 개인 정보 보호 정책 이용 약관

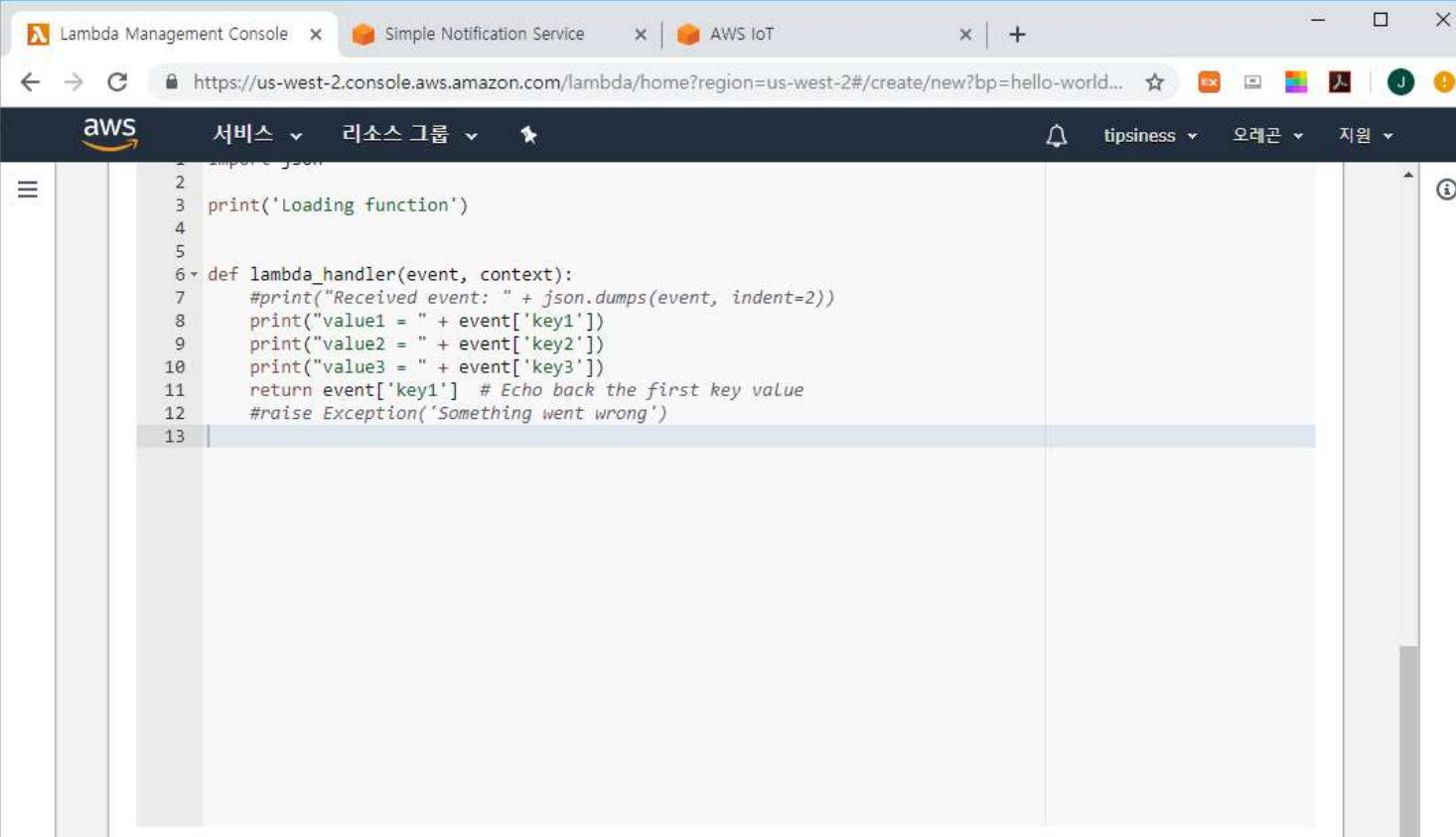
Device Registration

▶ REST의 정의



Device Registration

▶ REST의 정의



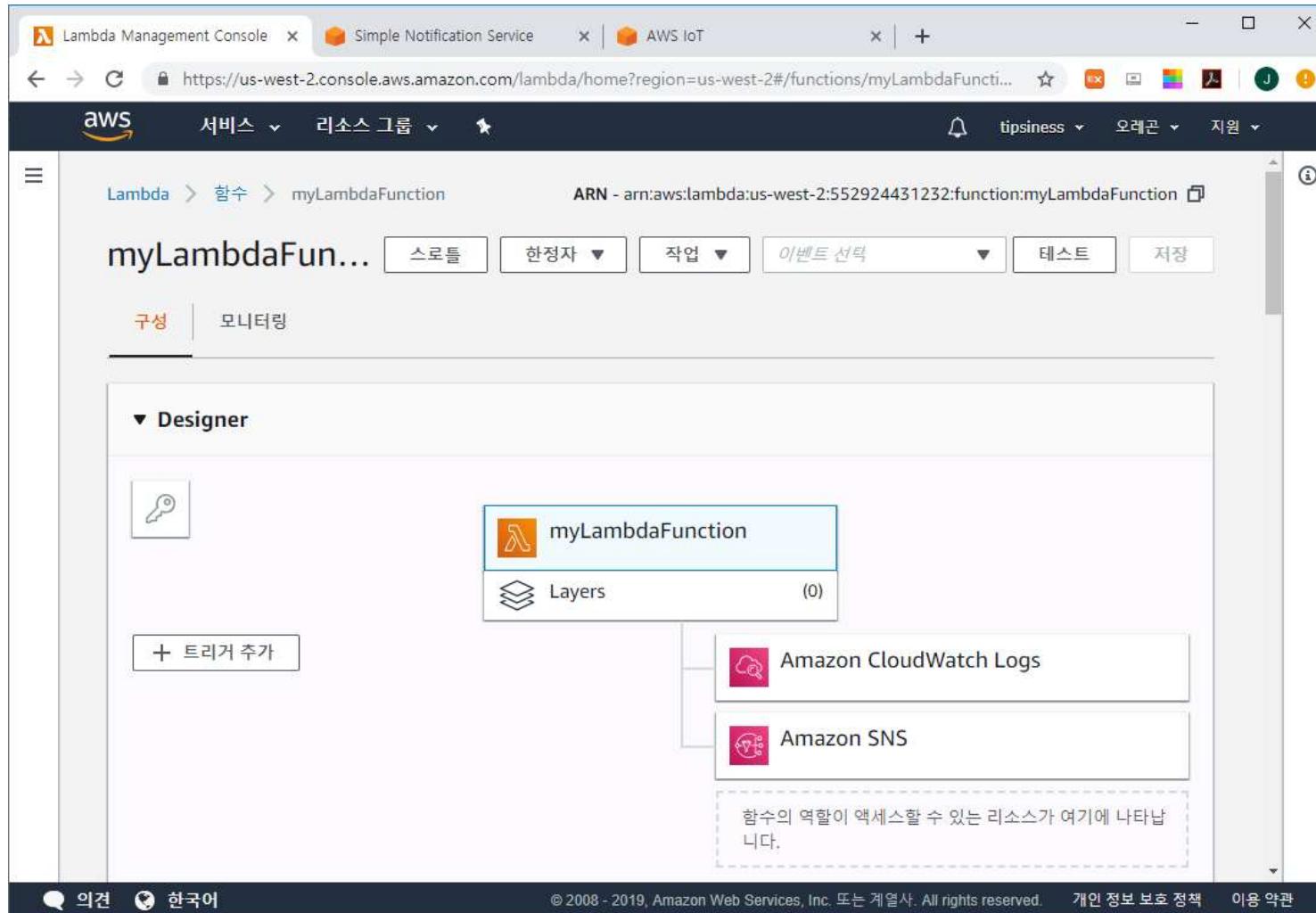
The screenshot shows the AWS Lambda Management Console interface. In the center, there is a code editor window displaying a Python script for a Lambda function. The script prints the loaded function and handles events, echoing back the first key value. The code is as follows:

```
1  #!/usr/bin/python
2
3  print('Loading function')
4
5
6  def lambda_handler(event, context):
7      #print("Received event: " + json.dumps(event, indent=2))
8      print("value1 = " + event['key1'])
9      print("value2 = " + event['key2'])
10     print("value3 = " + event['key3'])
11     return event['key1'] # Echo back the first key value
12     #raise Exception('Something went wrong')
13
```

At the bottom of the page, there is a note in Korean: "* 이 필드는 필수입니다." (This field is required). Below the code editor, there are three buttons: "취소" (Cancel), "이전" (Previous), and " 함수 생성" (Create Function). The " 함수 생성" button is highlighted with a red dashed border.

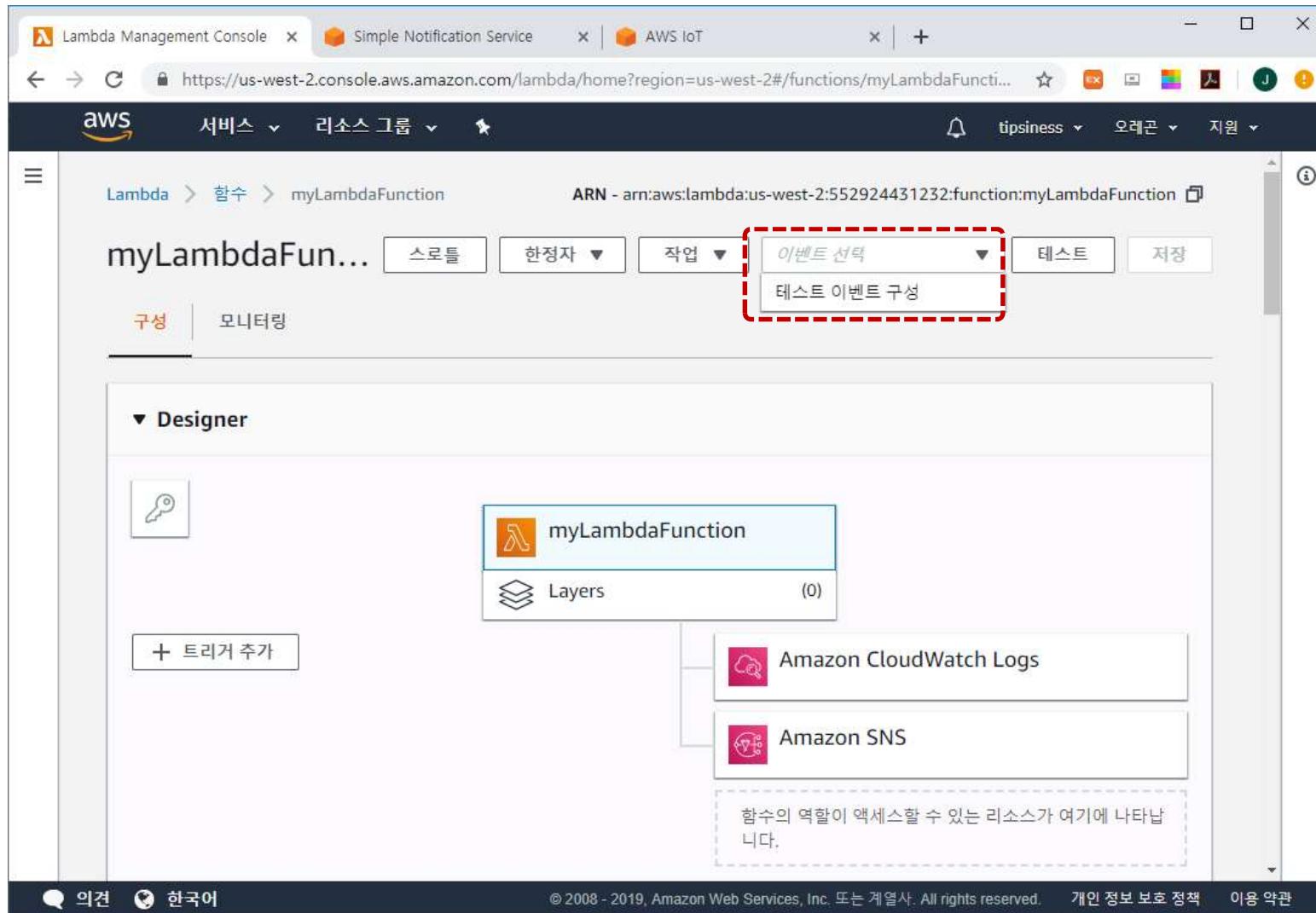
Device Registration

▶ REST의 정의



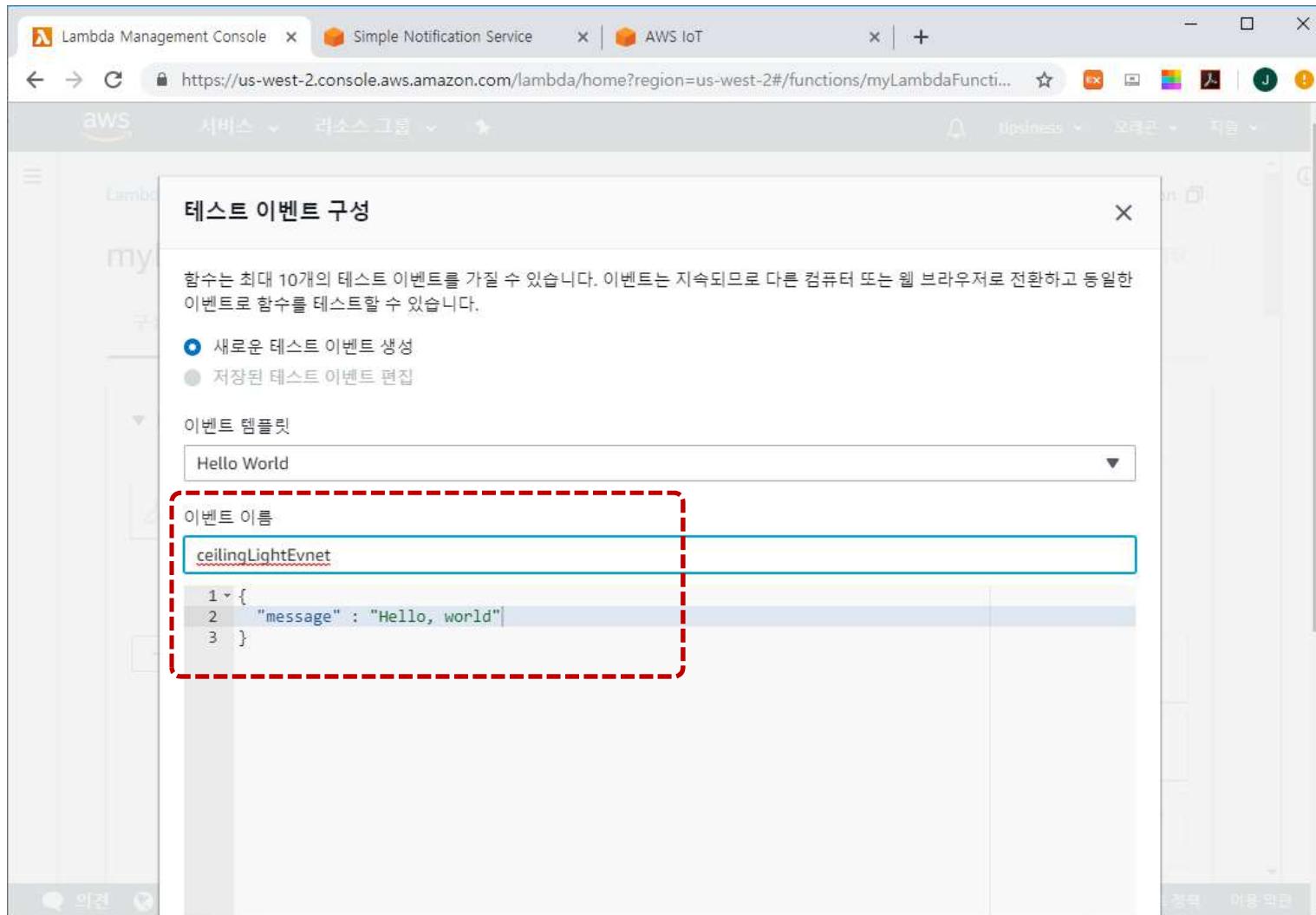
Device Registration

▶ REST의 정의



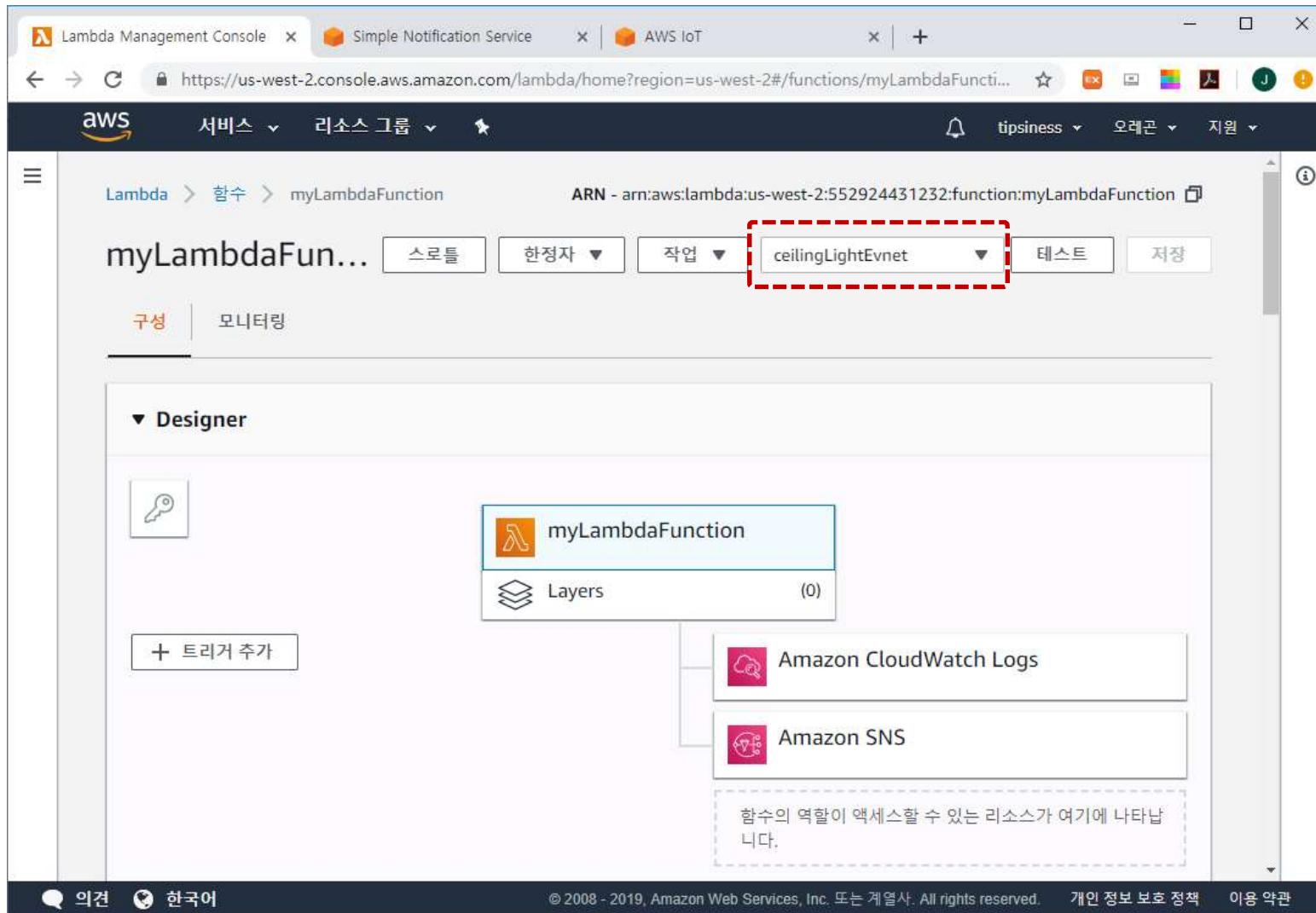
Device Registration

▶ REST의 정의



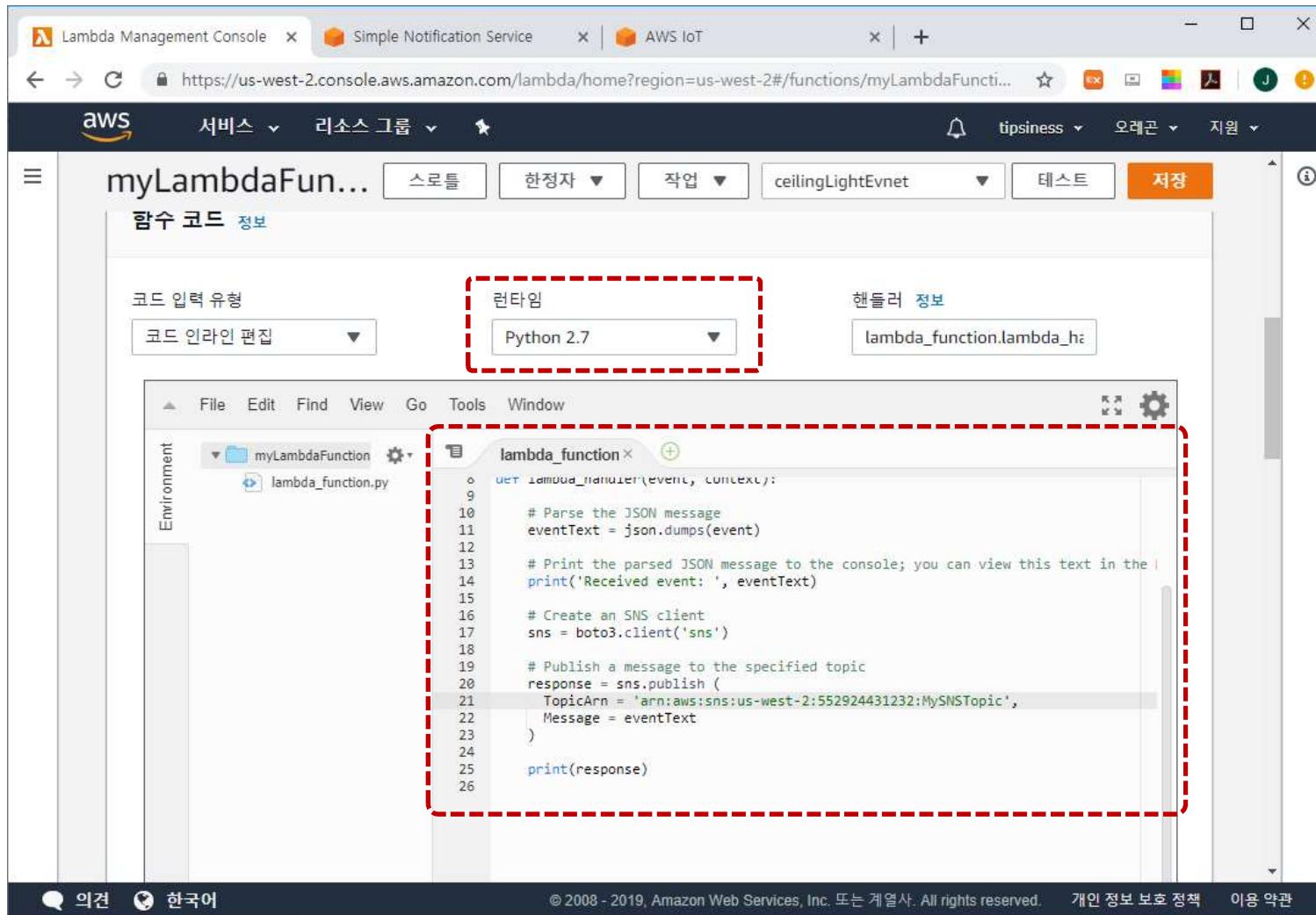
Device Registration

▶ REST의 정의



Device Registration

▶ REST의 정의





Device Registration

► lambda_function.py

```
from __future__ import print_function

import json
import boto3

print('Loading function')

def lambda_handler(event, context):
    eventText = json.dumps(event)

    print('Received event: ', eventText)

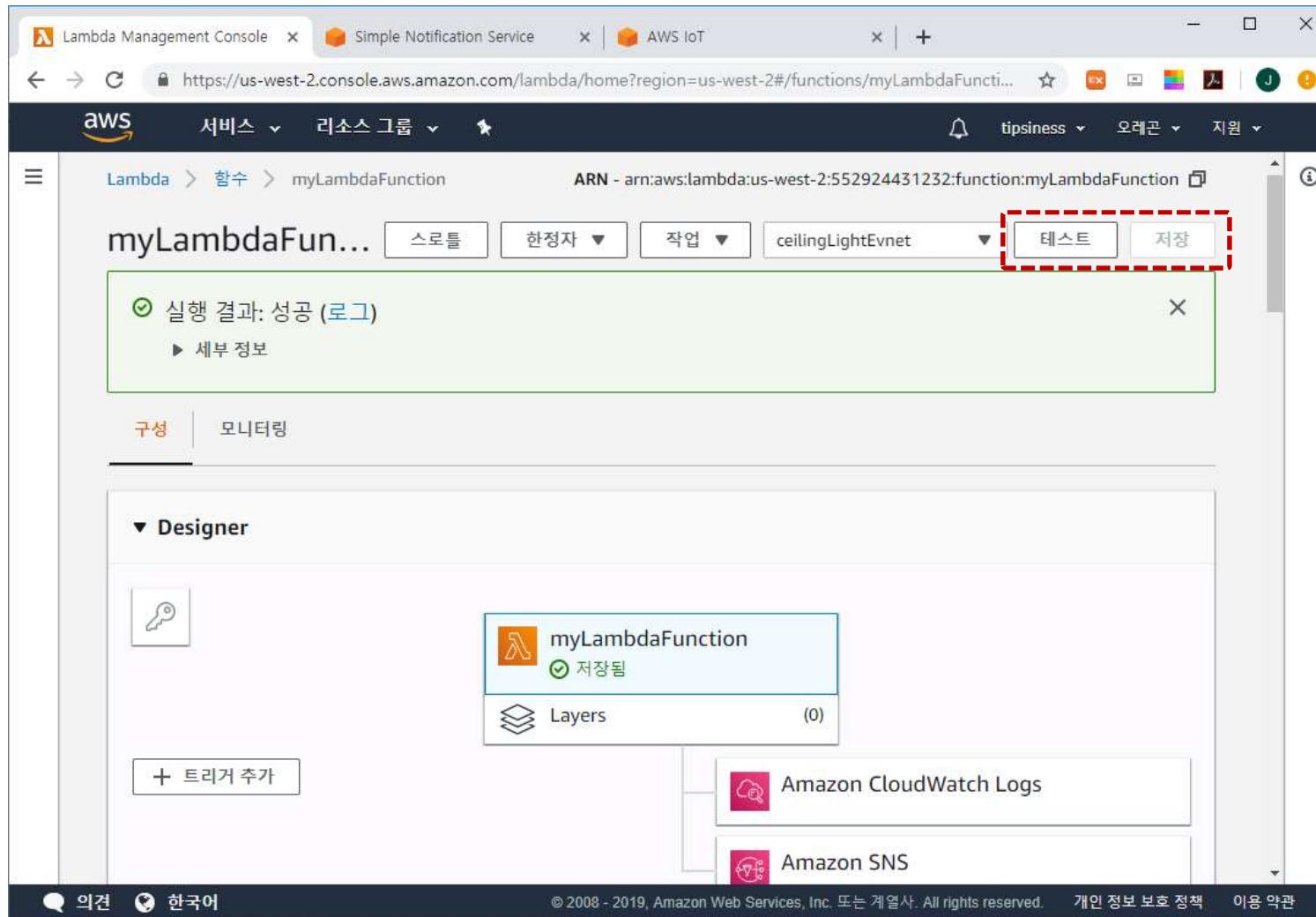
    sns = boto3.client('sns')

    response = sns.publish (
        TopicArn = 'arn:aws:sns:us-west-2:552924431232:MySNSTopic',
        Message = eventText
    )

    print(response)
```

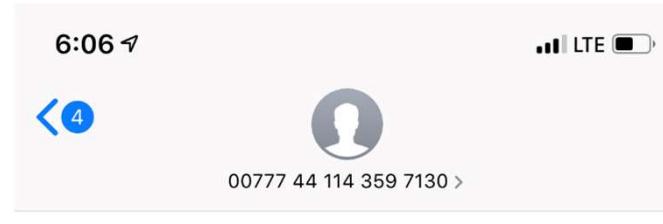
Device Registration

▶ REST의 정의



Device Registration

- ▶ REST의 정의

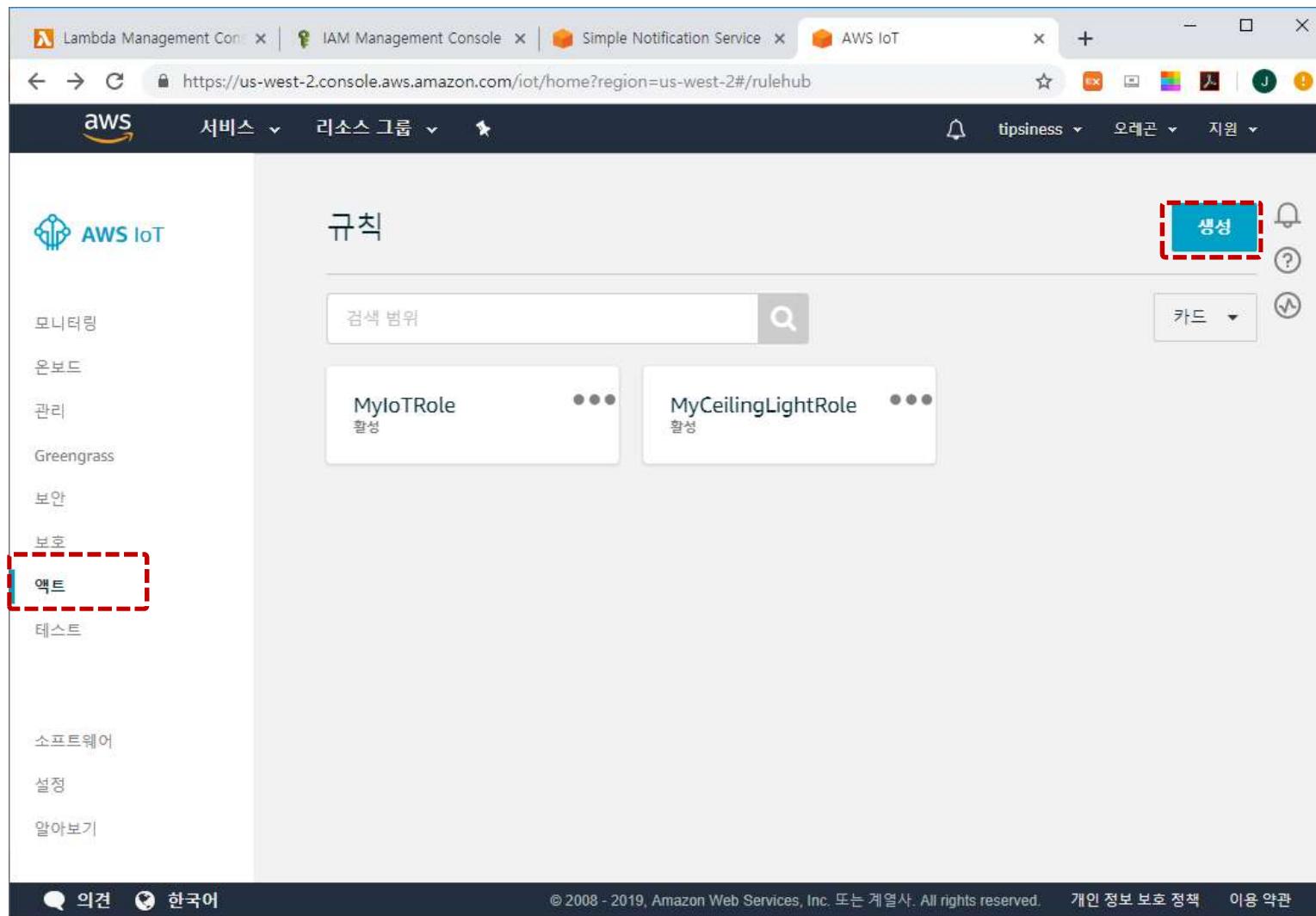


Rule



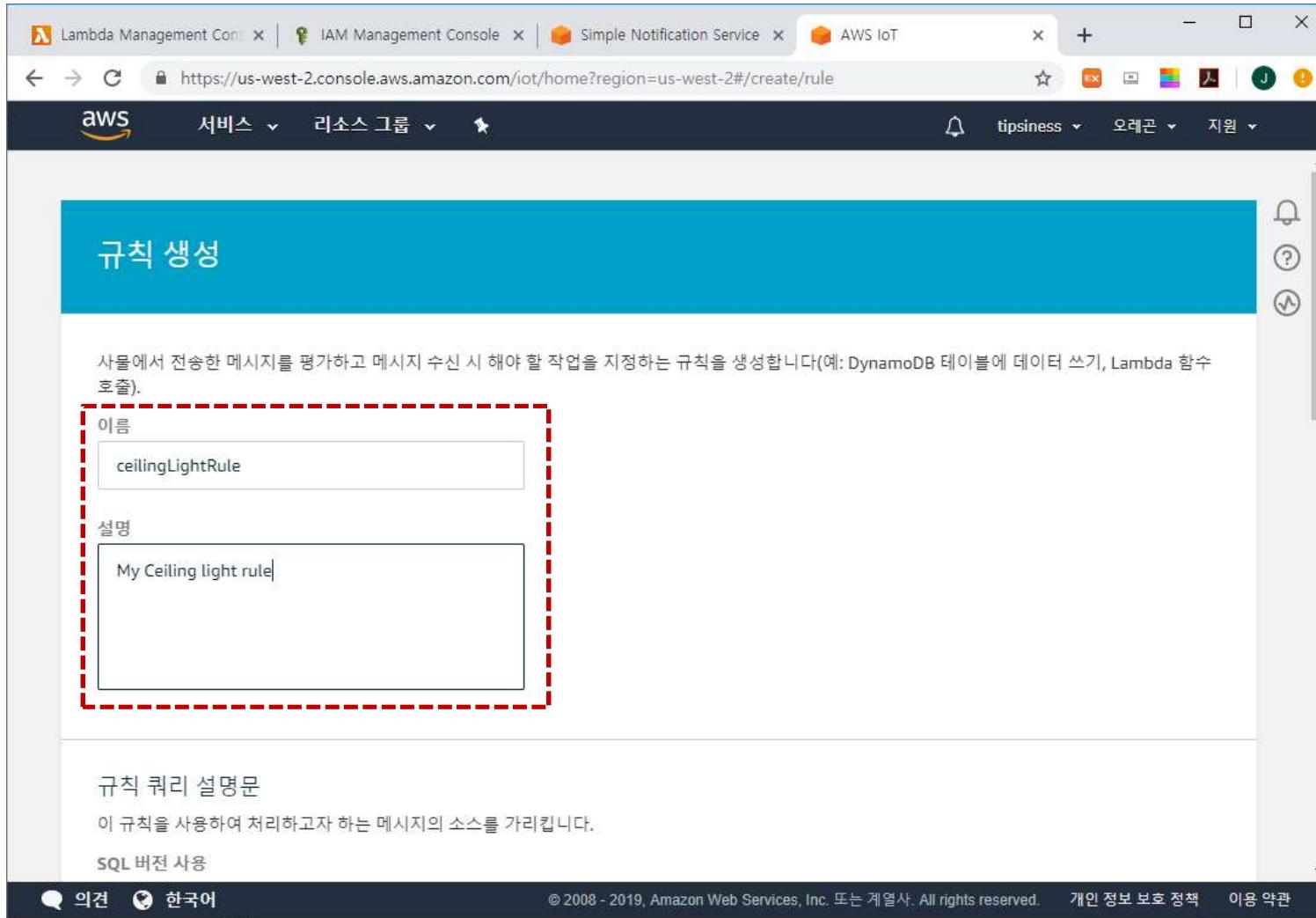
Device Registration

▶ REST의 정의



Device Registration

▶ REST의 정의



The screenshot shows the AWS IoT Rule Creation interface. At the top, there are tabs for Lambda Management Console, IAM Management Console, Simple Notification Service, and AWS IoT. The AWS logo and service name are visible on the left. The main title is "규칙 생성". Below it, a message in Korean says: "사물에서 전송한 메시지를 평가하고 메시지 수신 시 해야 할 작업을 지정하는 규칙을 생성합니다(예: DynamoDB 테이블에 데이터 쓰기, Lambda 함수 호출).". A red dashed box highlights the "이름" (Name) field containing "ceilingLightRule" and the "설명" (Description) field containing "My Ceiling light rule". Below these fields, there is a section titled "규칙 쿼리 설명문" with the sub-instruction "이 규칙을 사용하여 처리하고자 하는 메시지의 소스를 가리킵니다." and a "SQL 버전 사용" link. At the bottom, there are links for " 의견" and " 한국어", and a footer with copyright information: "© 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved." and links for "개인 정보 보호 정책" and "이용 약관".

Device Registration

▶ REST의 정의

Lambda Management Con... | IAM Management Console | Simple Notification Service | AWS IoT

https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/create/rule

aws 서비스 리소스 그룹 ★

tipsiness 오레곤 지원

규칙 쿼리 설명문

이 규칙을 사용하여 처리하고자 하는 메시지의 소스를 가리킵니다.

SQL 버전 사용

2016-03-23

규칙 쿼리 설명문

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. 예: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. SQL 설명문 작성에 관한 자세한 내용은 [AWS IoT SQL 참조](#)를 참조하십시오.

1 SELECT * FROM 'my/lambda/topic'

SELECT * FROM 'my/lambda/topic'

하나 이상의 작업을 설정

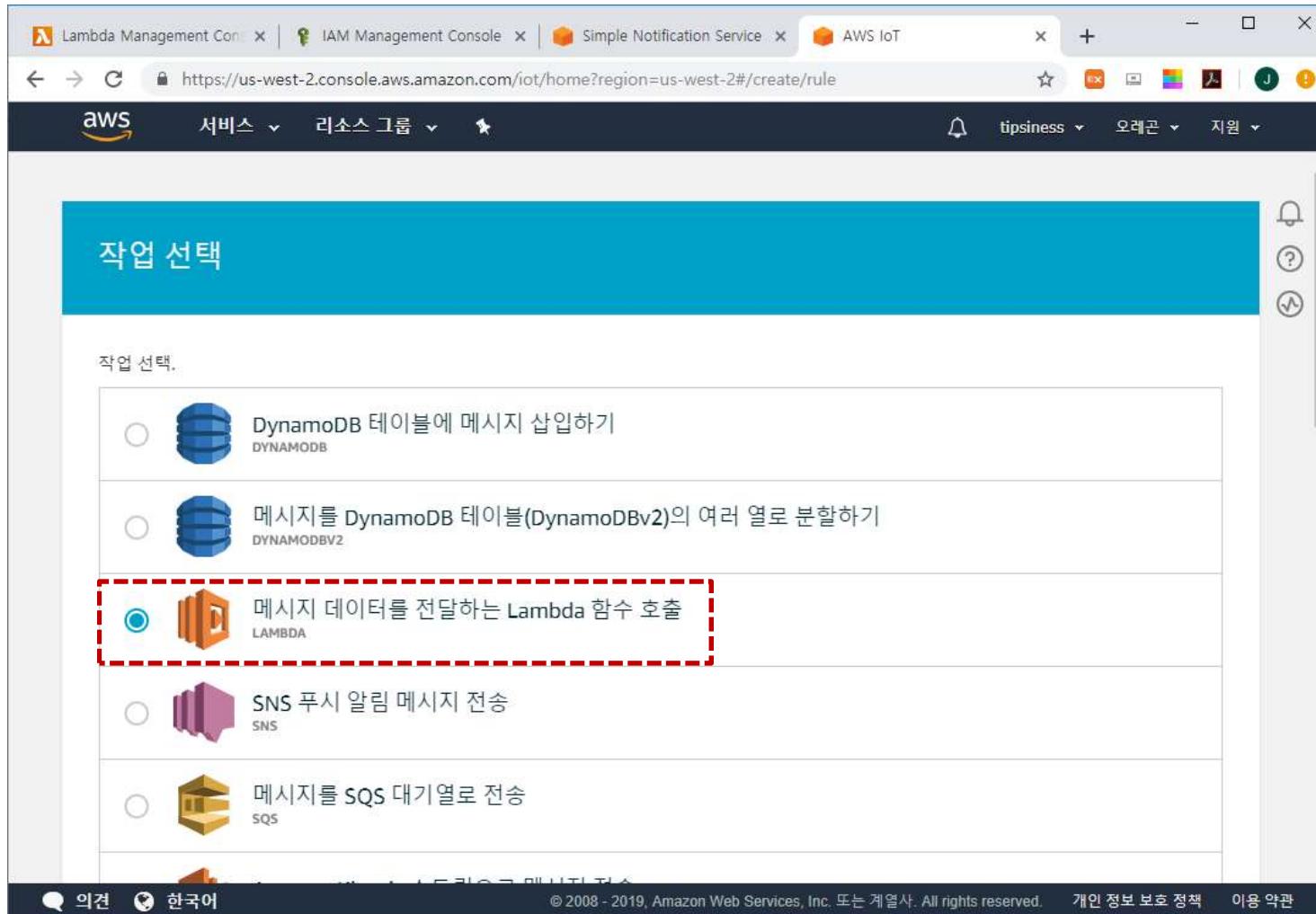
상기 규칙이 수신된 메시지와 일치할 때 이루어지는 작업을 하나 이상 선택합니다. 작업은 메시지가 도착하면 이루어지는 추가 활동을 정의합니다(예: 데이터베이스에 저장, 클라우드 함수 호출, 알림 전송 등) (*.필요).

작업 추가

으로 자아 의견 한국어 개인 정보 보호 정책 이용 약관

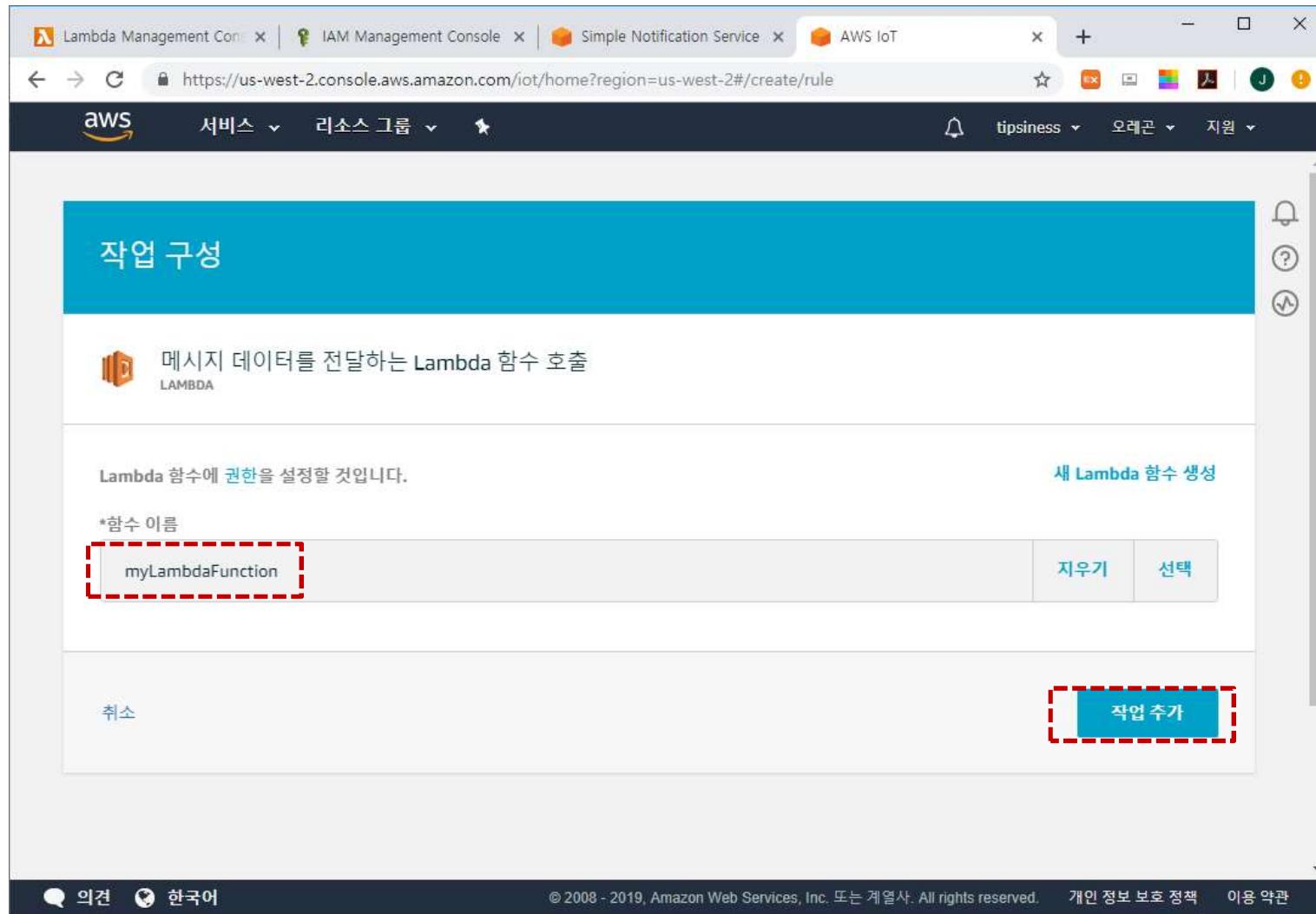
Device Registration

▶ REST의 정의



Device Registration

▶ REST의 정의



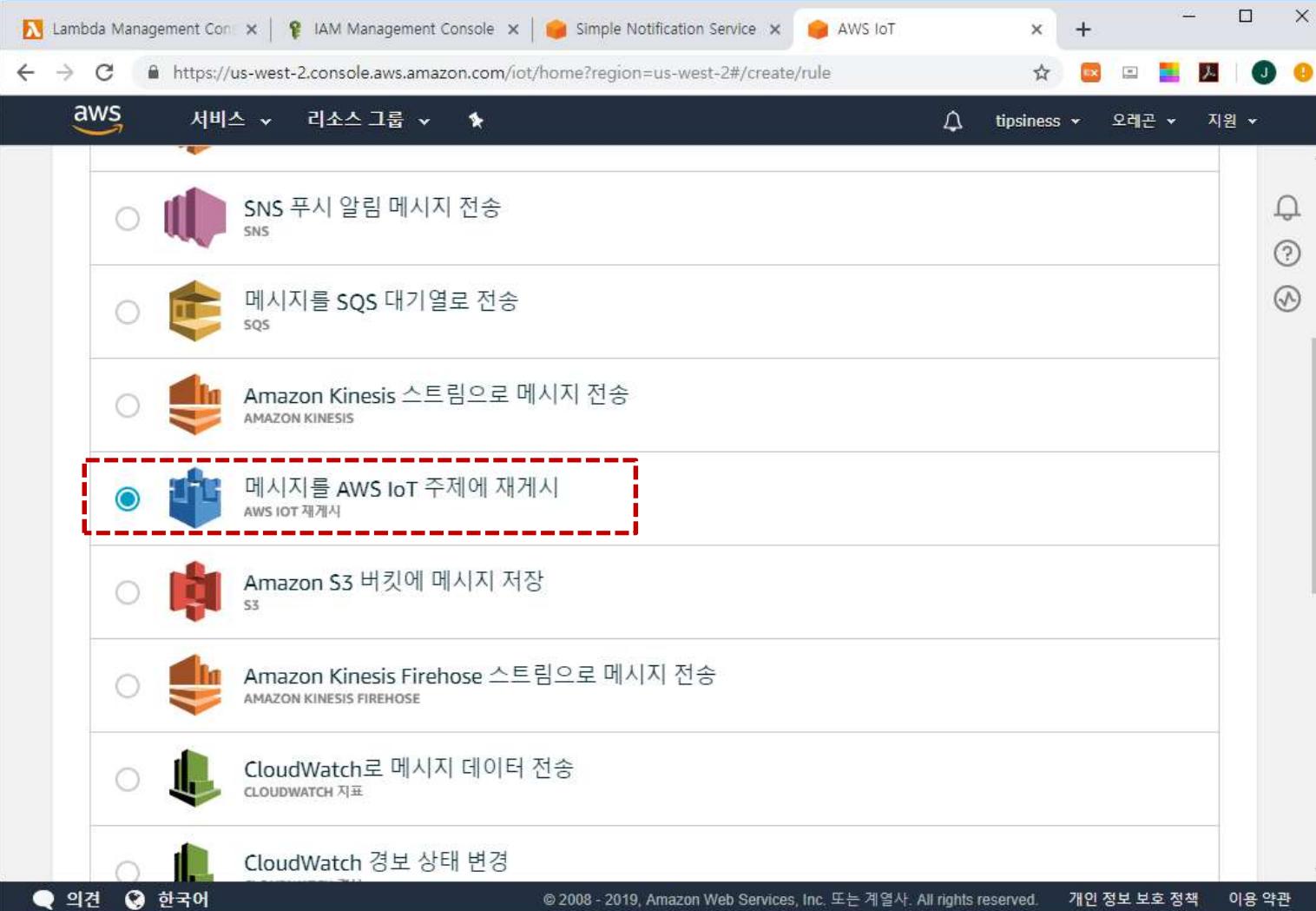
Device Registration

▶ REST의 정의

The screenshot shows the AWS IoT Rule Creation interface. At the top, there's a header with tabs for Lambda Management Console, IAM Management Console, Simple Notification Service, and AWS IoT. Below the header, the URL is https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/create/rule. The main content area has a title '하나 이상의 작업을 설정' (Configure one or more actions) and a note about selecting actions for message delivery. It lists a single action: '메시지 데이터를 전달하는 Lambda 함수 호출' (myLambdaFunction). Below this, there's a '작업 추가' (Add action) button. The next section, '오류 작업' (Error action), is described as executing actions when errors occur during rule processing. It also has a '작업 추가' button, which is highlighted with a red dashed box. The final section is '태그' (Tags), which allows users to categorize resources. It includes fields for '태그 이름' (Tag name) and '값' (Value), both with placeholder text '태그 이름(예: 제조업체) 제공' and '태그 값(예: Acme-Corporation) 제공'. A '지우기' (Delete) button is also present here. At the bottom, there are links for '의견' (Feedback), '한국어' (Korean), and copyright information: © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved. 개인 정보 보호 정책 and 이용 약관.

Device Registration

▶ REST의 정의



The screenshot shows the AWS IoT Rule Creation interface. A red dashed box highlights the option "메시지를 AWS IoT 주제에 재게시" (Redeliver message to AWS IoT topic). Other options listed include:

- SNS 푸시 알림 메시지 전송
- 메시지를 SQS 대기열로 전송
- Amazon Kinesis 스트림으로 메시지 전송
- 메시지를 AWS IoT 주제에 재게시 (highlighted)
- Amazon S3 버킷에 메시지 저장
- Amazon Kinesis Firehose 스트림으로 메시지 전송
- CloudWatch로 메시지 데이터 전송
- CloudWatch 경보 상태 변경

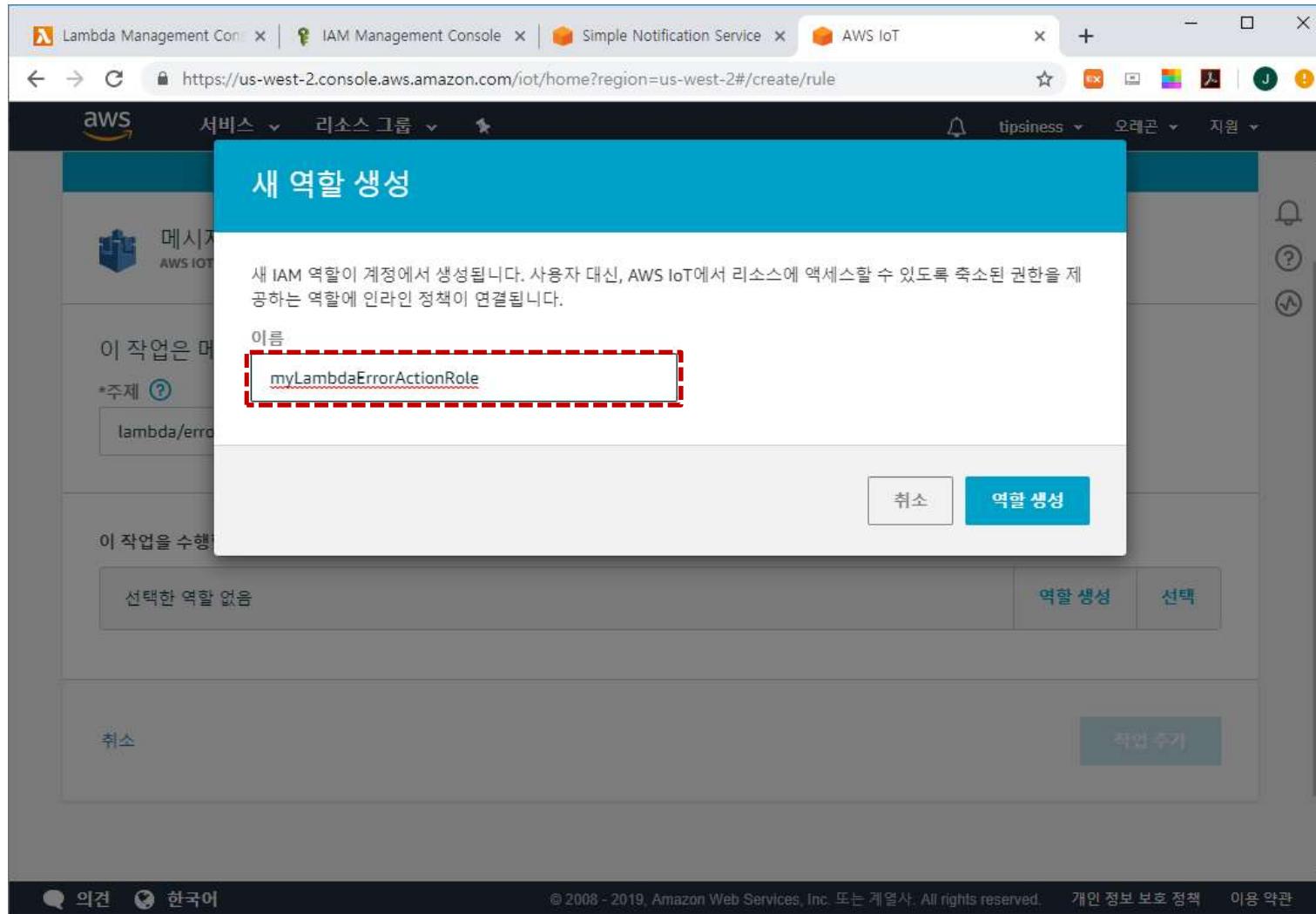
Device Registration

▶ REST의 정의

The screenshot shows the AWS IoT Rule Creation interface. At the top, there are tabs for Lambda Management Console, IAM Management Console, Simple Notification Service, and AWS IoT. The AWS IoT tab is active, showing the URL <https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/create/rule>. The main content area has a title "메시지를 AWS IoT 주제에 재게시" (Publish message to AWS IoT topic) and a sub-section "AWS IOT 재게시". It displays the message "이 작업은 메시지를 다른 AWS IoT 주제에 재게시합니다." (This task publishes messages to other AWS IoT topics). A red dashed box highlights the "주제" (Topic) input field, which contains "lambda/error". Below this, another red dashed box highlights the "역할 생성" (Create Role) button. At the bottom, there is a "취소" (Cancel) button and a large blue "작업 추가" (Add Task) button.

Device Registration

▶ REST의 정의



Device Registration

▶ REST의 정의

The screenshot shows the AWS IoT Rule Creation interface. At the top, there are tabs for Lambda Management Console, IAM Management Console, Simple Notification Service, and AWS IoT. The AWS IoT tab is active, showing the URL <https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/create/rule>. The main area displays a rule titled "메시지를 AWS IoT 주제에 재게시" (AWS IOT 재게시). The rule description states: "이 작업은 메시지를 다른 AWS IoT 주제에 재게시합니다." (This task re-posts messages to other AWS IoT topics). The "Topic" field contains "lambda/error". Below this, it says "이 작업을 수행할 권한을 AWS IoT에 부여하는 역할을 선택하거나 생성합니다." (Select or create a role to grant this task permission to AWS IoT). A dropdown menu shows "myLambdaErrorActionRole" and "연결된 정책" (Attached Policies). There are two buttons: "역할 생성" (Create Role) and "선택" (Select). At the bottom left is a "취소" (Cancel) button, and at the bottom right is a blue "작업 추가" (Add Task) button, which is highlighted with a red dashed border.

Device Registration

▶ REST의 정의

하나 이상의 작업을 설정

상기 규칙이 수신된 메시지와 일치할 때 이루어지는 작업을 하나 이상 선택합니다. 작업은 메시지가 도착하면 이루어지는 추가 활동을 정의합니다(예: 데이터베이스에 저장, 클라우드 함수 호출, 알림 전송 등) (*.필요).

작업 추가

메시지 데이터를 전달하는 Lambda 함수 호출
myLambdaFunction

제거 편집 >

오류 작업

규칙 처리 도중 문제가 발생할 때 실행될 작업을 선택적으로 설정합니다.

메시지를 AWS IoT 주제에 재게시
lambda/error

제거 편집 >

태그

리소스에 태그를 적용하여 리소스를 손쉽게 정리하고 식별합니다. 태그는 대소문자를 구분하는 키-값 쌍으로 이루어져 있습니다. AWS 리소스 태깅에 대해 [자세히 알아보십시오.](#)

태그 이름 값

© 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved. 개인 정보 보호 정책 이용 약관

Device Registration

▶ REST의 정의

The screenshot shows the AWS IoT Rule Creation interface. At the top, there's a navigation bar with tabs for Lambda Management Console, IAM Management Console, Simple Notification Service, and AWS IoT. The URL in the browser is <https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/create/rule>. Below the navigation, there's a title "규칙 처리 도중 문제가 발생할 때 실행될 작업을 선택적으로 설정합니다." (Optional configuration for actions to be taken when a problem occurs during rule processing). A rule card is displayed: "메시지를 AWS IoT 주제에 재게시" (Republish message to AWS IoT topic) with the condition "lambda/error". There are "제거" (Delete) and "편집" (Edit) buttons. To the right, there are three icons: a bell, a question mark, and a refresh symbol.

태그

리소스에 태그를 적용하여 리소스를 손쉽게 정리하고 식별합니다. 태그는 대소문자를 구분하는 키-값 쌍으로 이루어져 있습니다. AWS 리소스 태깅에 대해 [자세히 알아보십시오.](#)

태그 이름: 태그 이름(예: 제조업체) 제공
값: 태그 값(예: Acme-Corporation) 제공
[지우기]
[다른 사항 추가]

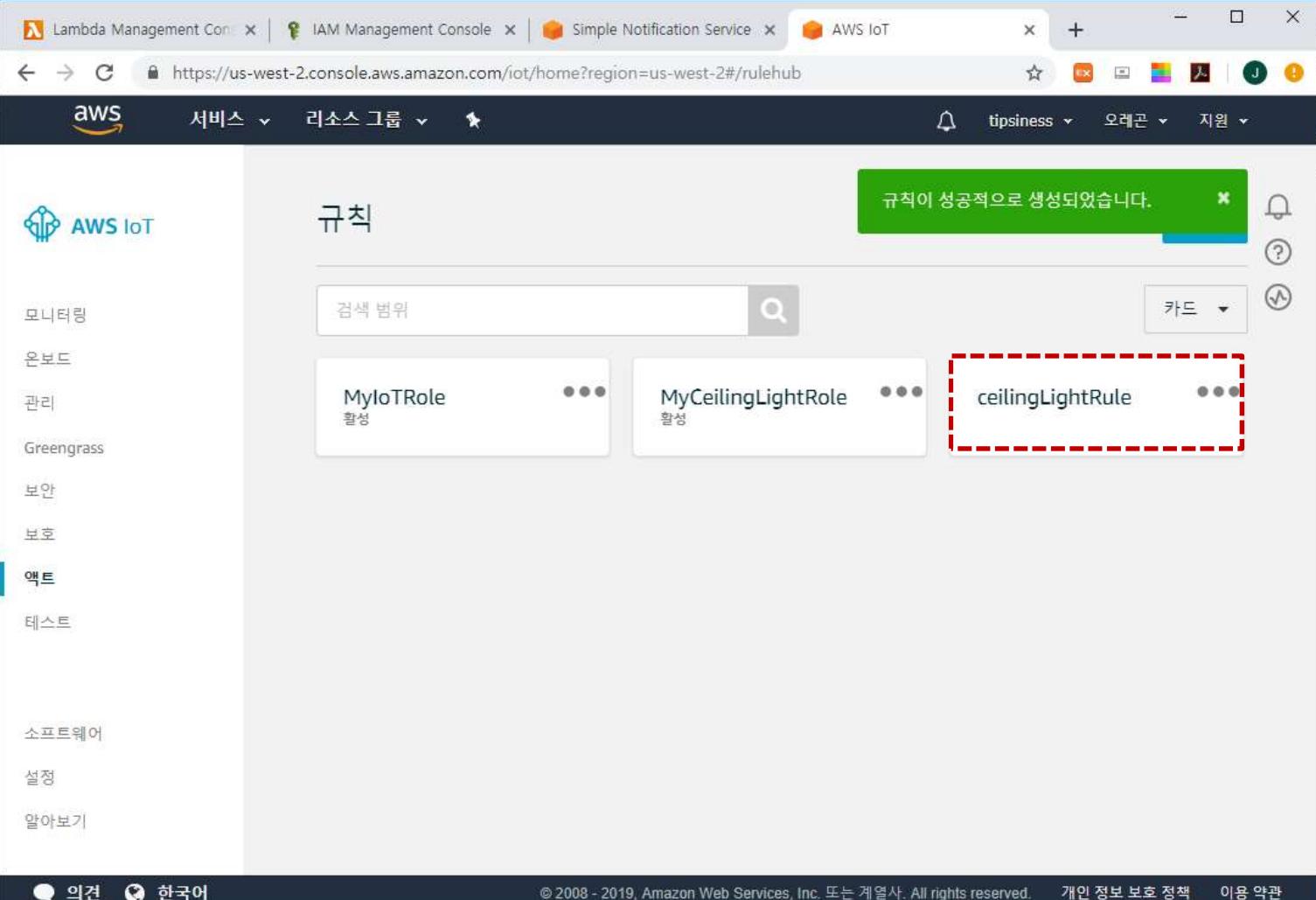
최소

[규칙 생성]

At the bottom, there are footer links: 의견 (Feedback), 한국어 (Korean), © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved., 개인 정보 보호 정책 (Privacy Policy), and 이용 약관 (Terms of Service).

Device Registration

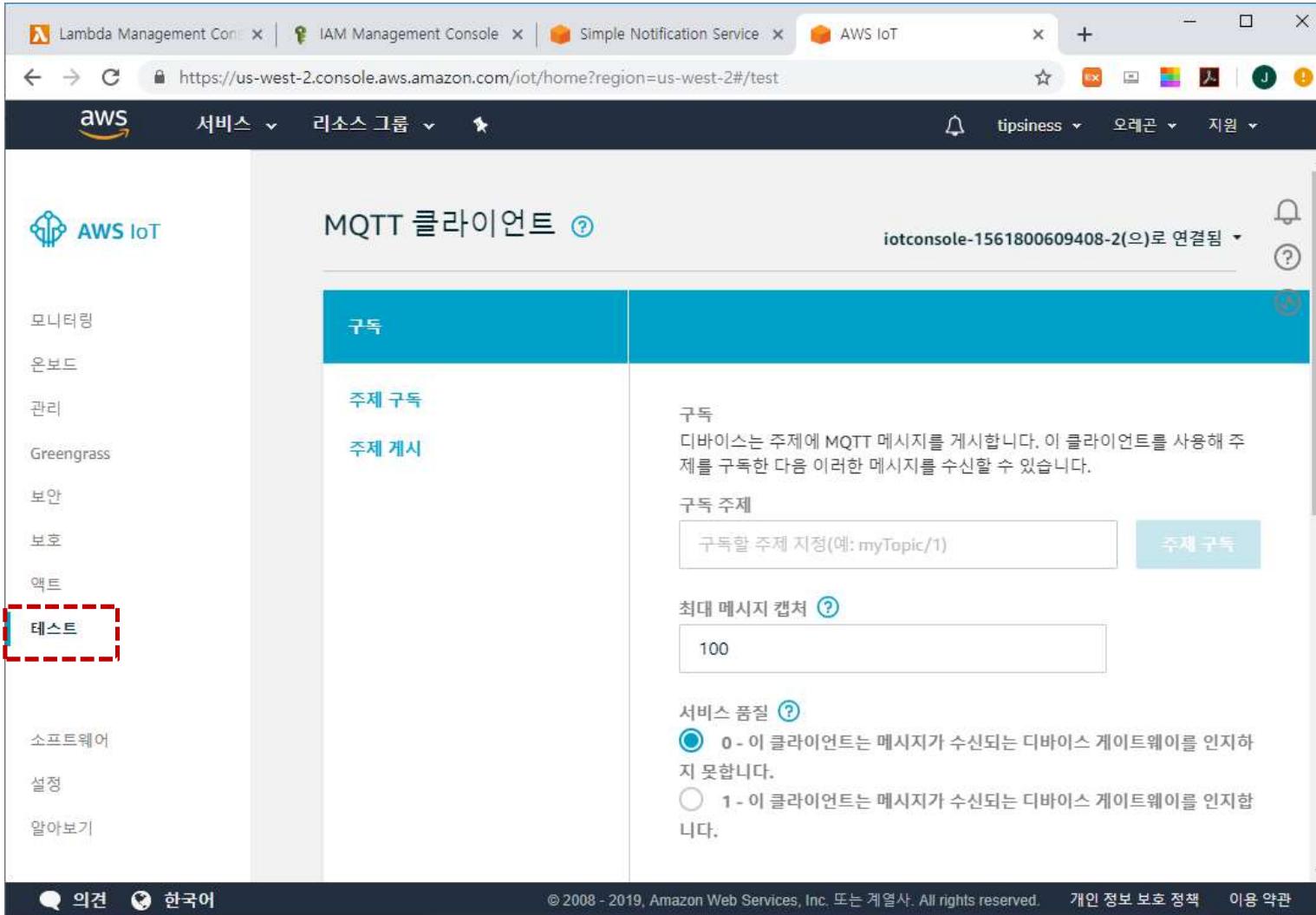
▶ REST의 정의



The screenshot shows the AWS IoT Rule Hub console. On the left sidebar, under the '액트' (Actions) section, there is a list of actions: 모니터링 (Monitoring), 온보드 (Onboarding), 관리 (Management), Greengrass, 보안 (Security), 보호 (Protection), 테스트 (Testing), 소프트웨어 (Software), 설정 (Settings), and 알아보기 (Discover). The main content area is titled '규칙' (Rules). It features a search bar and a card view. Three rules are listed: 'MyIoTRole' (활성), 'MyCeilingLightRole' (활성), and 'ceilingLightRule'. The 'ceilingLightRule' card is highlighted with a red dashed border. A green success message at the top right says '규칙이 성공적으로 생성되었습니다.' (The rule was successfully created).

Device Registration

▶ REST의 정의



The screenshot shows the AWS IoT MQTT Client interface. The left sidebar has a menu with items: 모니터링, 온보드, 관리, Greengrass, 보안, 보호, 액트, 테스트 (which is highlighted with a red dashed box), 소프트웨어, 설정, and 알아보기. The main content area is titled "MQTT 클라이언트" and shows a connection status of "iotconsole-1561800609408-2(으)로 연결됨". It has two tabs: "구독" (Subscription) and "주제 게시" (Topic Publish). The "구독" tab contains a section for "구독" (Subscription) where it says "디바이스는 주제에 MQTT 메시지를 게시합니다. 이 클라이언트를 사용해 주제를 구독한 다음 이러한 메시지를 수신할 수 있습니다." and a "구독 주제" input field with placeholder "구독할 주제 지정(예: myTopic/1)" and a "주제 구독" button. Below that is a "최대 메시지 캡처" (Max Message Capture) input field with value "100". At the bottom, there is a "서비스 품질" (Service Quality) section with two radio buttons: "0 - 이 클라이언트는 메시지가 수신되는 디바이스 게이트웨이를 인지하지 못합니다." (selected) and "1 - 이 클라이언트는 메시지가 수신되는 디바이스 게이트웨이를 인지합니다.". The footer includes links for " 의견" (Feedback), " 한국어" (Korean), copyright information "© 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved.", and "개인 정보 보호 정책" (Privacy Policy) and "이용 약관" (Terms of Service).

Device Registration

▶ REST의 정의

The screenshot shows the AWS IoT MQTT Client interface. On the left sidebar, under the 'AWS IoT' section, the '테스트' (Test) option is selected. In the main area, the title is 'MQTT 클라이언트'. A connection status bar at the top right indicates 'iotconsole-1561800609408-2(으)로 연결됨'. The central panel displays a table with two columns: '구독' (Subscription) and '내보내기' (Publish). A single row is shown for the topic 'lambda/error'. The '내보내기' column contains a text input field with the message content: '1 { 2 "message": "Hello from AWS IoT console" 3 }'. Two red boxes highlight the '내보내기' button and the message content. The bottom of the interface includes standard AWS footer links: ' 의견', ' 한국어', ' © 2008 - 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved.', ' 개인 정보 보호 정책', and ' 이용 약관'.

Device Registration

▶ REST의 정의

The screenshot shows the AWS IoT MQTT Client interface. On the left sidebar, under the 'AWS IoT' section, the '테스트' (Test) option is selected. In the main area, a table lists a single subscription:

구독	내보내기	지우기	일시 중지
lambda/error	내보내기	지우기	일시 중지

Below the table, there are two sections: '주제 구독' (Topic Subscription) and '주제 게시' (Topic Publish). The '주제 게시' section contains a text input field with the value 'my/lambda/topic' and a '주제 게시' (Publish Topic) button. A red dashed box highlights the input field and the publish button.

At the bottom of the interface, there is a code editor window displaying the following JSON message:

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

Device Registration

- ▶ REST의 정의

