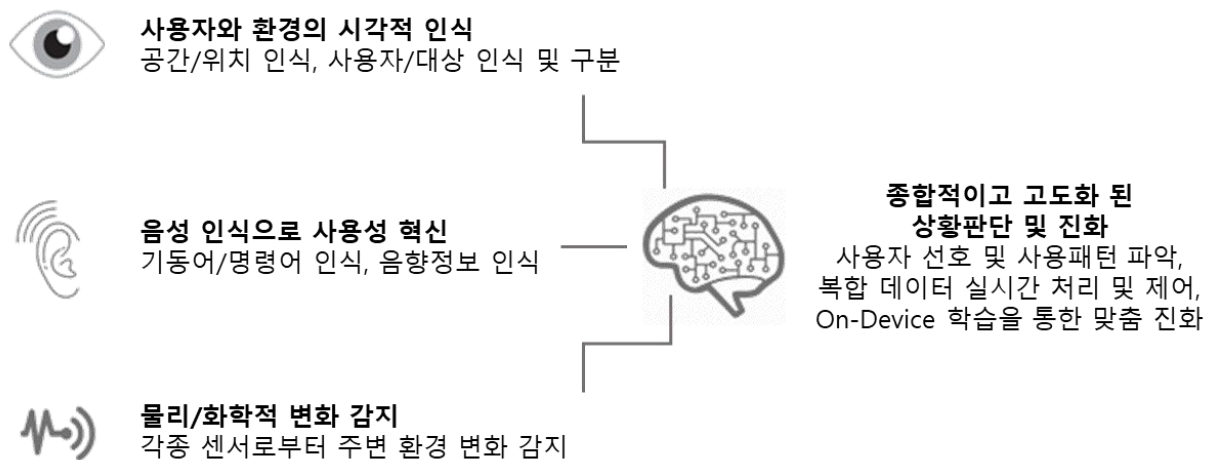
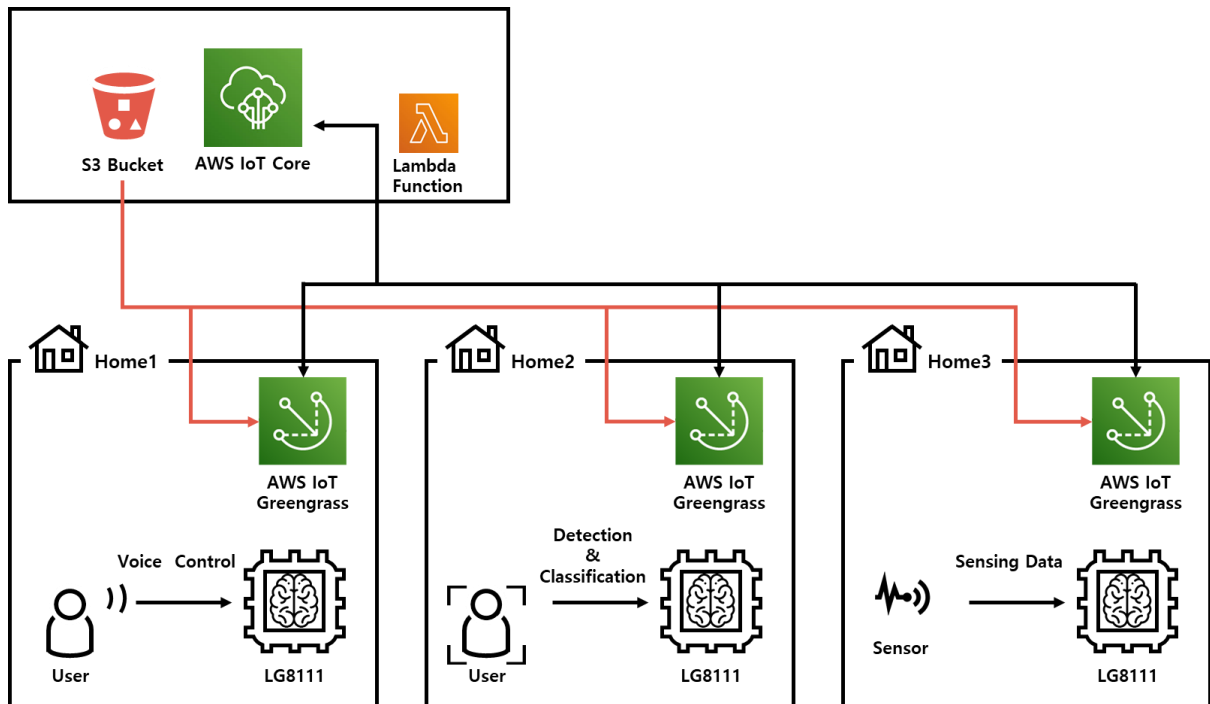


클라우드기반 인공지능은 네트워크 상시 연결이라는 제약으로 다양한 솔루션 개발에서 제한적인 적용만을 할 수밖에 없었습니다. LG전자는 외부와 통신이 끊긴 상황에서도 인공지능 기능을 제공하는 On-Device 인공지능 칩인 LG8111을 개발하였습니다.

LG8111은 영상, 음성, 제어지능으로 분류된 인공지능 기능의 하드웨어적인 처리를 지원합니다. LG8111은 딥러닝 알고리즘을 효율적(저전력, 낮은 대기시간)으로 처리하는 LG전자 고유의 AI 프로세서를 활용하여 '온디바이스(On-Device) 인공지능'을 구현할 수 있으며, 인공지능의 어려움으로 지적되고 있던 상시 네트워크 연결로 인한 여러 제약 사항들을 벗어날 수 있습니다.



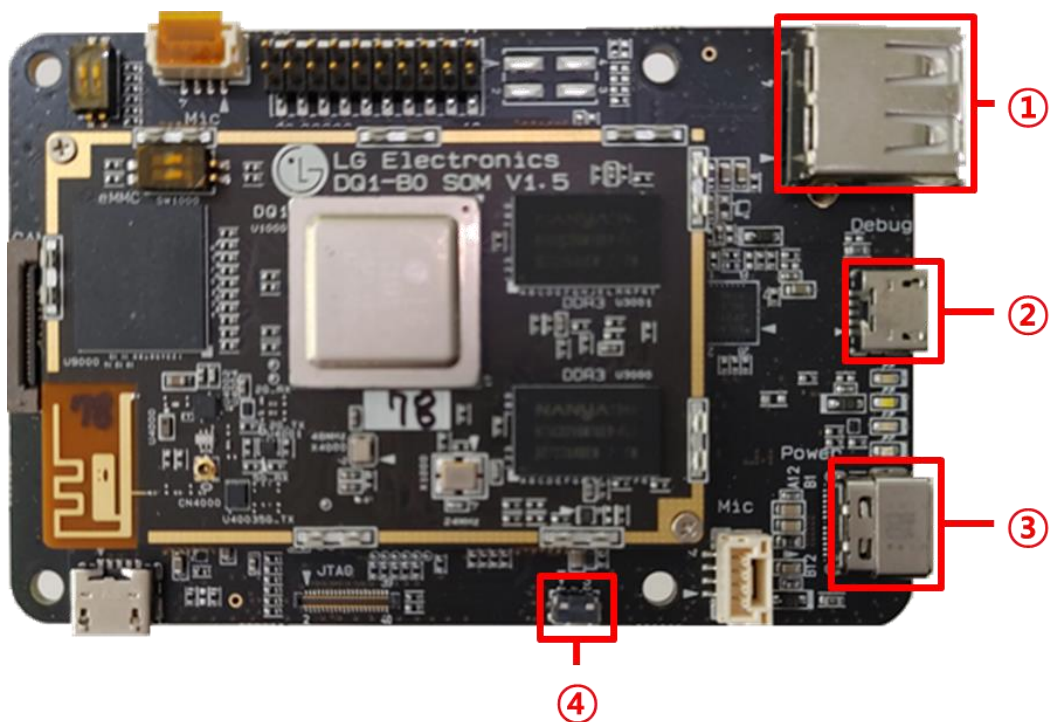
AWS IoT Greengrass는 AWS IoT를 디바이스까지 확장하고, 클라우드의 장점을 활용하면서, 디바이스를 통해 생성된 데이터를 로컬에서 처리할 수 있는 환경을 제공합니다. 앞서 언급한 On-Device 인공지능 기능을 통해, LG8111은 향상된 Sensor Data의 수집/분석 과 ML Inference 성능을 제공할 수 있습니다. 동시에 LG8111 개발 보드인 Eris 보드를 통하여 AWS IoT Greengrass를 보다 손쉽게 통합할 수 있고, 다양한 솔루션들을 구현 할 수 있습니다.



LG8111 개발 보드

[LG8111 개발 보드 구성]

LG8111 개발 보드는 Ubuntu18.04 환경을 지원하며, H/W는 Reset, USB Host, UART Debug Port, Power 로 구성됩니다.



번호	영역이름	설명
1	USB Host	USB Device를 연결하기 위한 포트입니다.
2	UART Debug Port	Serial Port입니다.
3	Power	Power에 USB Type-C를 연결하여 전원을 켭니다.
4	Reset	보드를 리셋 합니다.

[보드 연결]

LG8111 개발 보드에 연결하는 방법에 관해 설명합니다.

1. LG8111 개발 보드의 UART Debug Port에 Host(PC)와 연결된 USB선을 연결합니다.
2. LG8111 개발 보드의 Power에 전원을 인가합니다.

LG8111 개발 보드에 UART를 연결할 때 필요한 설정 값은 아래와 같습니다.

RULES	VALUE
BAUD RATE	115200 bps
DATE BITS	8 bit
PARITY SET	N
STOP BITS	1

LG8111 개발 보드에 AWS Greengrass 사용

LG8111 개발 보드에 AWS Greengrass 환경 설정은 AWS IoT Greengrass 개발자 안내서 (https://docs.aws.amazon.com/ko_kr/greengrass/latest/developerguide/setup-filter.rpi.html)에 따라 Greengrass 환경 설정 및 AWS IoT Greengrass 코어 소프트웨어 설치까지 따라 진행하시면 되고, AWS IoT Greengrass 코어 소프트웨어는 v1.10.0을 사용하시면 됩니다.

[AWS IoT Greengrass 인증서 획득]

AWS IoT Greengrass를 실행하기 위해서는 디바이스마다 고유한 인증서를 획득해야 합니다. 이러한 고유한 인증서를 AWS IoT Greengrass에서는 group라고 구분을 짓고 있습니다.

인증서 획득을 위해 AWS IoT Greengrass 시작 페이지에서 그룹 생성을 선택합니다.



Monitor

Onboard

Manage

Greengrass

Groups

Cores

Devices

Secure

Defend

Act

Test

Welcome to AWS IoT Greengrass

AWS IoT Greengrass lets your devices process the data they generate locally, while still taking advantage of AWS services when an internet connection is available.

Get started today with Greengrass on a [Raspberry Pi](#), or read our [system requirements and list of compatible devices](#).

Create a Greengrass Group

Create a Group and provision a Core in one step, or walk through the process step by step.

Create a Group

Set up your Greengrass Group

Setting up your Group requires you to provision a Core device in the IoT Registry, acquire a certificate for your Core, and assign an IAM role to your Group. If you're unfamiliar with any of these steps we recommend the default Group creation. Finally, you'll need to install Greengrass software on your Core device.

Default Group creation (recommended)

This process will automatically provision a Core in the registry, use default settings to generate a new Group, and provide your Core with a new certificate and a key pair.

Use default creation

Advanced Group creation

This customizable process will take you step-by step through the Core provisioning and will allow you to customize the IAM Role for your Group and the certificate for your Core, and provide a key pair.

Customize

Cancel

Use default creation

생성할 그룹의 이름을 입력 후 다음을 선택합니다.

SET UP YOUR GREENGRASS GROUP

Name your Group

The Greengrass Group is a cloud-configured managed collection of local devices and Lambda functions that can be programmed to communicate with each other through a Core device. Groups can contain up to 200 local devices.

Group Name

LG8111

Apply tags to the Group (optional) ▾

Cancel

Back

Next

AWS IoT Greengrass 그룹 코어 명을 기본 설정 값을 사용하고 다음을 선택합니다.

SET UP YOUR GREENGRASS GROUP

Every Group needs a Core to function

Every Greengrass Group requires a device running Core software. It enables communication between Devices, local Lambda functions, and AWS cloud computing services. Adding information to the Registry is the first step in provisioning a device as your Greengrass Core.

Name

LG8111_Core

Show optional configuration (this can be done later) ▾

Cancel

Back

Next

[Run a scripted easy Group creation] 페이지에서 [Create Group and Core]를 선택하여 설정 값들이 문제없이 진행되는지 확인합니다. 설정 값들이 문제가 없을 시, 설정 내용들이 모두 녹색이 되며 자동으로 다음 페이지로 넘어갑니다.

SET UP YOUR GREENGRASS GROUP

Review Group creation

In order to speed up and simplify Group creation AWS IoT Greengrass will handle the following processes and use default settings. By proceeding to the next step, you are giving permission for us to complete the following steps.

AWS IoT Greengrass will take these actions on your behalf using default settings:

Create a new Greengrass Group in the cloud	Learn more
Provision a new Core in the IoT Registry and add to the Group	Learn more
Generate public and private key set for your Core	Learn more
Generate a new security certificate for the Core using the keys	Learn more
Attach a default security policy to the certificate	Learn more
Enable stream manager on the Core device	Learn more

[Cancel](#)

Create Group and Core

SET UP YOUR GREENGRASS GROUP

Review Group creation

In order to speed up and simplify Group creation AWS IoT Greengrass will handle the following processes and use default settings. By proceeding to the next step, you are giving permission for us to complete the following steps.

AWS IoT Greengrass will take these actions on your behalf using default settings:

✓ Create a new Greengrass Group in the cloud	Learn more
✓ Provision a new Core in the IoT Registry and add to the Group	Learn more
✓ Generate public and private key set for your Core	Learn more
✓ Generate a new security certificate for the Core using the keys	Learn more
✓ Attach a default security policy to the certificate	Learn more
✓ Enable stream manager on the Core device	Learn more

[Cancel](#)

Create Group and Core

정상적으로 AWS IoT Greengrass 그룹을 생성 시 코어 리소스를 받을 수 있습니다. 다운 받은 코어 리소스 파일을 다시 받을 수 없기 때문에 반드시 저장 및 관리를 해야 됩니다. 만약 삭제 시 코어 리소스를 재 생성해야 합니다.

Connect your Core device


The final steps are to load the Greengrass software and then connect your Core device to the cloud. You can defer connecting your device at this time, but **you must download your public and private keys now as these cannot be retrieved later.**

Download and store your Core's security resources

A certificate for this Core	e892362d7d.cert.pem
A public key	e892362d7d.public.key
A private key	e892362d7d.private.key
Core-specific config file	config.json

Download these resources as a tar.gz

AWS IoT Greengrass 그룹이 정상적으로 생성 시 Greengrass IoT Greengrass 시작 페이지에서 그룹 창에 생성한 그룹이 보이게 됩니다.



- Monitor
- Onboard
- Manage
 - Greengrass**
 - Groups**
 - Cores
 - Devices
- Secure
- Defend
- Act
- Test

Greengrass Groups

LG8111

GREENGRASS GROUP

LG8111 개발 보드에서 AWS Greengrass 실행

지금 살펴볼 것은 LG8111 개발 보드의 로컬 리소스(Camera, LG AI 프로세서)를 사용하여 Camera Input data를 추론하고, 그 추론 결과를 AWS Greengrass로 보내 결과값을 확인해 보는 예제입니다.

해당 예제 실행을 위해 AWS Greengrass에서 제공하는 python용 AWS IoT Greengrass SDK를 개발자 안내서 페이지(https://docs.aws.amazon.com/ko_kr/greengrass/latest/developerguide/what-is-gg.html#gg-core-sdk-download)에서 다운을 받아야 합니다.

해당 예제는 python2.7에서 구현하였습니다. 5초마다 camera로 촬영된 이미지를 MobileNet이라는 인공지능 알고리즘을 사용하여 추론하고, 추론된 결과를 AWS IoT Greengrass로 보내줍니다.

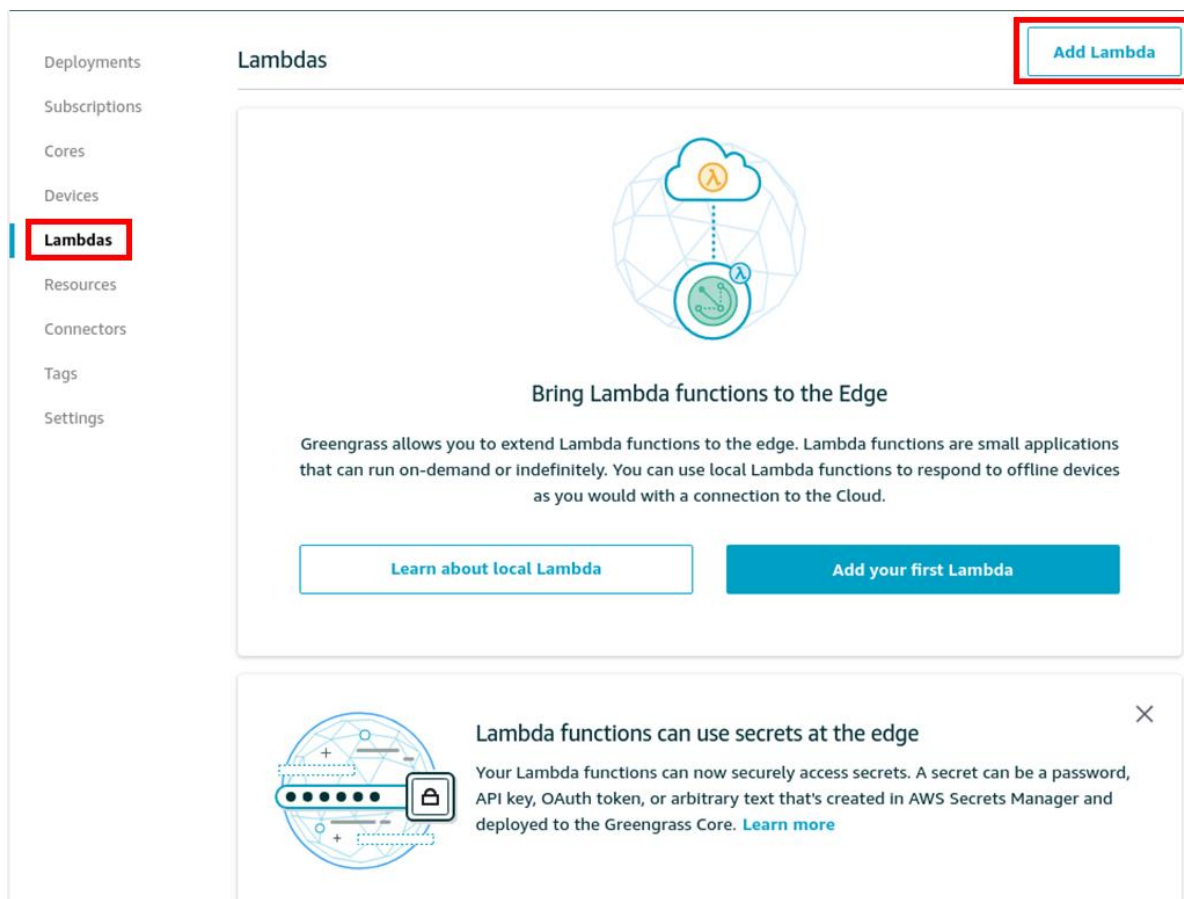
디바이스의 "/home/ubuntu/sample/getting_started"폴더 내의 내용들과 python용 AWS IoT Greengrass SDK를 동일 경로에 놓고 1개의 파일로 압축을 수행하면 됩니다.

```
$ zip -r greengrassMobilenet.zip labels.txt Mobilenet.lne greengrassMobilenet.py greengrasssdk
```

1. Lambda 등록

AWS Greengrass의 group으로 등록한 LG8111 개발 보드 group으로 접속합니다.

접속한 group의 Labmda -> Add Labmda를 선택하여 새로운 Lambda를 등록합니다.



Create new Lambda를 선택하여 LG8111 개발 보드에서 AWS Greengrass를 테스트 할 Lambda 함수를 추가합니다.

Add a Lambda to your Greengrass Group

Local Lambdas are hosted on your Greengrass Core and connected to each other and devices by Subscriptions, but they can also be deployed individually to your Group.

Create a new Lambda function

You will be taken to the AWS Lambda Console and can author a new Lambda function.

Create new Lambda

Use an existing Lambda function

You will choose from a list of existing Lambda functions.

Use existing Lambda

Cancel

Back

Use existing Lambda

Function name은 Lambda를 구분할 수 있는 이름으로 등록하고, python 2.7을 기반으로 작성한 예제이기 때문에 Runtime 옵션을 python 2.7로 선택 후 다음 단계로 이동합니다.

Basic information

Function name

Enter a name that describes the purpose of your function.

GG_Mobilenet

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)

Choose the language to use to write your function.

Python 2.7

AWS Greengrass를 LG8111 개발 보드에서 동작시킬 Lambda 함수는 .zip 형식으로 되어 있기 때문에 code entry type은 zip으로, python 2.7에서 Lambda를 동작 시키기 위해 Runtime은 python2.7로 선택합니다.

Handler는 "[Lambda 파일 명].[Lambda 파일 내 handler 명]"으로 지정을 합니다. 예제 코드의 파일명은 "greengrassMobilenet", 예제 lambda 파일 내 handler 명은 "function_handler"로 되어있기 때문에 "greengrassMobilenet.function_handler"로 등록을 합니다.

마지막으로 Upload 버튼을 눌러 예제 코드를 압축한 zip파일을 선택하여 등록합니다.

Function code [Info](#)

Code entry type

Upload a .zip file

Runtime

Python 2.7

Handler [Info](#)

greengrassMobilenet.function_handler

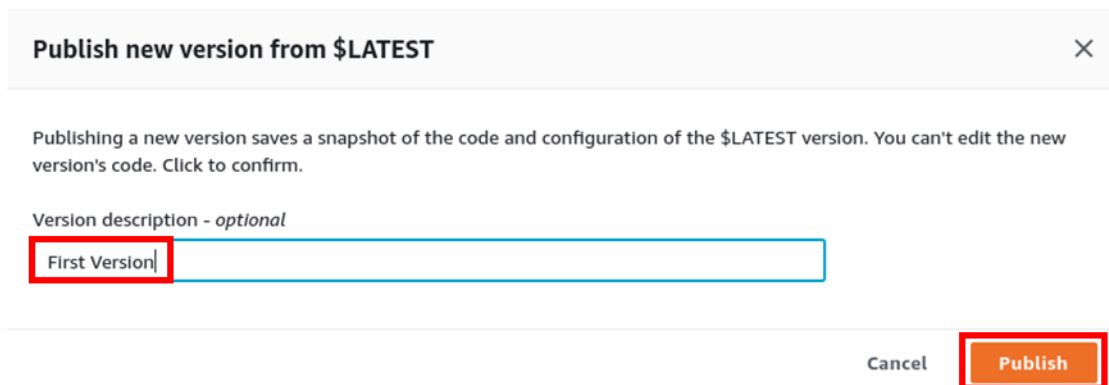
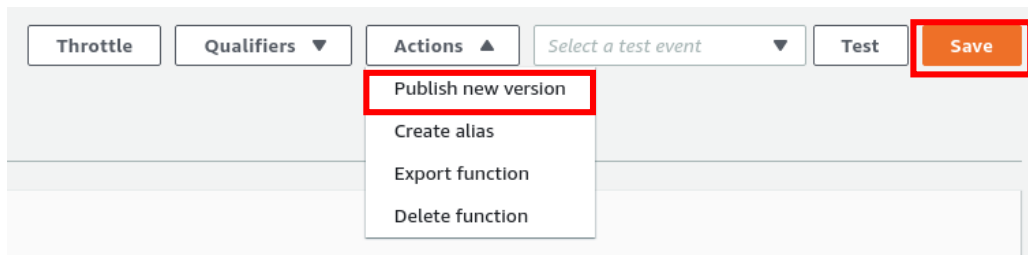
Function package

Upload greengrassMobilenet.zip (13.4 MB)

For files larger than 10 MB, consider uploading using Amazon S3.

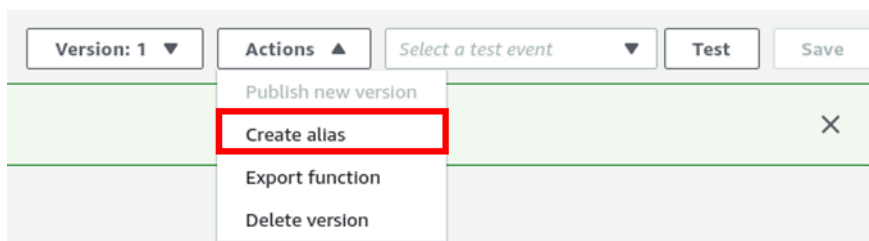
화면 우측 상단의 save 버튼을 눌러 등록된 내용을 저장 후 Actions의 Publish new version을 선택

택 후 저장된 version에 대한 설명을 입력합니다.



Version에 대한 설명을 저장 후 해당 version에 대한 alias 및 version을 지정하여 저장합니다.

(Version을 \$LATEST로 선택 할 경우 동작하지 않을 수 있습니다.)



Create a new alias

×

An alias is a pointer to one or two versions. Choose each version that you want the alias to point to.

Name*

GG_Mobilenet

Description

Version*

▲

\$LATEST

1

Additional version

▼

based on weights (%) that you assign. Click [here](#) to learn more.

Cancel

Create

저장한 lambda를 사용하기 위해 Lambda -> Add Lambda를 선택하여 "Use existing Lambda"를 선택합니다.

Add a Lambda to your Greengrass Group

Local Lambdas are hosted on your Greengrass Core and connected to each other and devices by Subscriptions, but they can also be deployed individually to your Group.

Create a new Lambda function

You will be taken to the AWS Lambda Console and can author a new Lambda function.

Create new Lambda

Use an existing Lambda function

You will choose from a list of existing Lambda functions.

Use existing Lambda

Cancel

Back

Use existing Lambda

테스트를 위해 등록된 Lambda의 Lambda name, Lambda version을 선택합니다.

ADD A LAMBDA TO YOUR GREENGRASS GROUP

Use existing Lambda

Select a Lambda

<input type="radio"/>	helloWorld	Python 2.7
<input type="radio"/>	GG_MobileNet	Python 2.7
<input type="radio"/>	cloud9-GGMLInference-GGMLInference-1O39DN17CWN94	Python 3.7
<input checked="" type="radio"/>	GG_Mobilenet	Python 2.7
<input type="radio"/>	GG_MobileNet_label	Python 2.7
<input type="radio"/>	GG_LT_LeNet	Python 2.7
<input type="radio"/>	GGMLWorkshop-BootstrapC9InstanceLambdaFunction-14AH42JZYCOTH	Python 2.7
<input type="radio"/>	GG_Inceptionv3	Python 2.7

Cancel

Back

Next

ADD A LAMBDA TO YOUR GREENGRASS GROUP

Select a Lambda version

Select a Lambda version

☒ Alias: GG_Mobilenet

☐ Version 1

Cancel
Back
Finish

Test를 위해선 등록된 Lambda 함수의 설정을 수정해야 합니다. 이를 위해 추가된 Lambda 함수의 우측 3개의 원을 눌러 Edit configuration을 선택합니다.

GREENGRASS GROUP

DQ1_4

Successfully completed

Actions ▾

Deployments

Subscriptions

Cores

Devices

Lambdas

Resources

Connectors

Tags

Settings

Lambdas

[Add Lambda](#)

GG_Mobilenet

LAMBDA FUNCTION

USING

[Edit configuration](#)

[Remove function](#)

Lambda functions can use secrets at the edge

Your Lambda functions can now securely access secrets. A secret can be a password, API key, OAuth token, or arbitrary text that's created in AWS Secrets Manager and deployed to the Greengrass Core. [Learn more](#)

Lambda 설정 중 Memory limit, Timeout, Lambda lifecycle을 수정하면 됩니다.

Memory limit는 Lambda 함수 실행 시 memory 제한 양으로 넉넉하게 128MB를 설정합니다.

Timeout은 Lambda 함수 동작 대기 시간으로 Test Lambda가 5초마다 결과를 전송하기 때문에 25초로 설정합니다.

Lambda lifecycle은 Lambda함수의 lifecycle 설정으로, Test를 위해 항상 동작할 수 있도록 "Make this function log-lived and keep it running indefinitely"로 설정 후 수정된 내용을 저장합니다.

GG_Mobilenet

[View function in AWS Lambda](#)

Alias GG_Mobilenet

[Remove version](#)

Run as ?

☒ Use group default (currently: ggc_user/ggc_group)

☐ Another user ID/group ID

Containerization ?

☒ Use group default (currently: Greengrass container)

☐ Greengrass container (always)

☐ No container (always)

Memory limit

128

MB

Timeout

25

Second

Lambda lifecycle

☐ On-demand function

☒ Make this function long-lived and keep it running indefinitely

Read access to /sys directory

☒ Disable

☐ Enable

Input payload data type

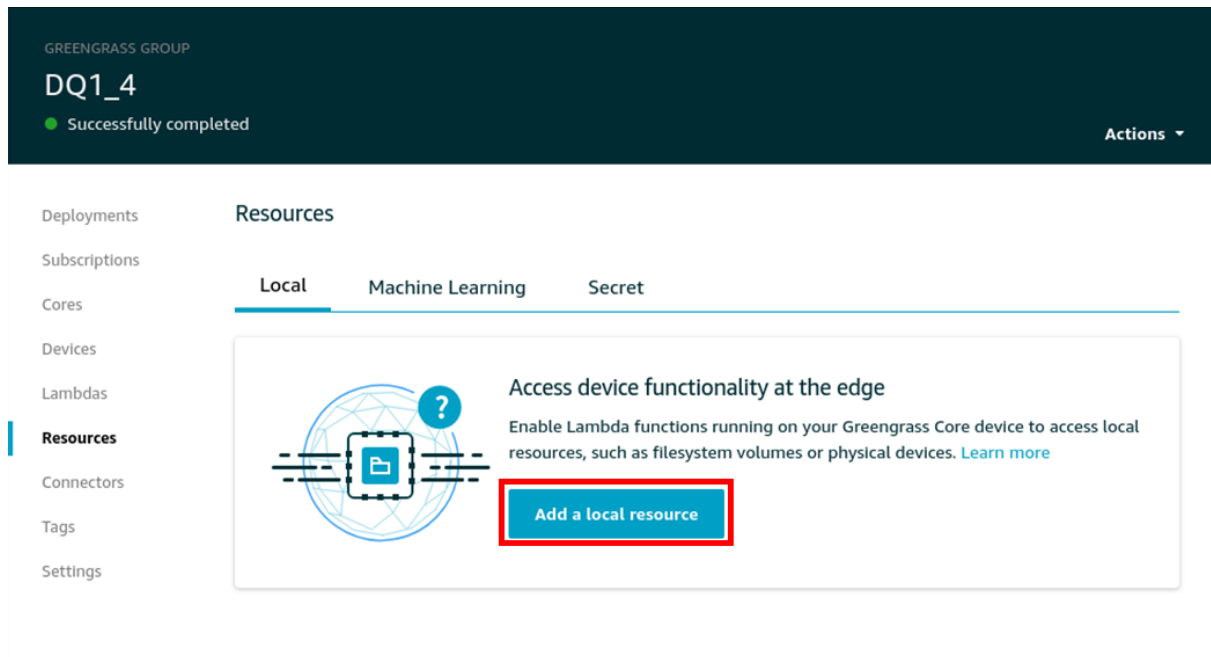
☒ JSON

2. Local Resource 등록

AWS Greengrass를 실행 시 사용되는 LG8111 개발 보드 내 Device들의 Local resource(ex. Camera, sensor, LG AI 프로세서 등)를 등록합니다. 등록된 local resource 들은 Lambda 함수가 동작 시 사용하는 local resource 입니다.

이름	DEVICE PATH	설명
LG AI 프로세서	/dev/dq1_lne	딥러닝 알고리즘을 효율적(저전력, 낮은 대기시간)으로 처리하는 LG 고유의 AI 프로세서
CAMERA	/dev/video0	LG8111 Development Device의 Camera

Resource의 Local탭을 선택하여 local resource를 추가합니다.



2-1. LG AI 프로세서 Local Resource 추가

LG AI 프로세서의 local resource를 추가하기 위해 device path와 해당 local resource를 사용하는 lambda function을 설정합니다. 이때, Lambda function의 설정을 "Read and Write access"로 설정을 합니다.

Add a new local resource

Resource name

LNE_Driver

Resource type

- ☒ Device
- ☐ Volume

Device path

/dev/dq1_lne

Group owner file access permission

An AWS IoT Greengrass Lambda function process normally runs without an OS Group. However, you can give additional file access permissions to the Lambda function process.

- ☐ No OS group
- ☒ Automatically add OS group permissions of the Linux group that owns the resource
- ☐ Specify another OS group to add permission

Lambda function affiliations

Resources must be affiliated with a Lambda function before deployment

Done

Find

GG_Mobilenet

Add a new local resource

Resource name

LNE_Driver

Resource type

☒ Device

☐ Volume

Device path

/dev/dq1_lne

Group owner file access permission

An AWS IoT Greengrass Lambda function process normally runs without an OS Group. However, you can give additional file access permissions to the Lambda function process.

☐ No OS group

☒ Automatically add OS group permissions of the Linux group that owns the resource

☐ Specify another OS group to add permission

Lambda function affiliations

GG_Mobilenet

READ AND WRITE ACCESS

Done

Specify the permission this Lambda will have to the resource.

☐ Read-only access

☒ Read and write access

LG AI 프로세서의 local resource 등록이 완료 후 정상적으로 lambda function과 연결이 되면 Status가 녹색으로 변경됩니다.

GREENGRASS GROUP

DQ1_4

● Successfully completed

Actions ▾

Deployments

Subscriptions

Cores

Devices

Lambdas

Resources

Connectors

Tags

Settings

Resources

Local

Machine Learning

Secret

Add local resource

Name	Resource Type ▾	Status	Local path ▾
LNE_Driver	Device	● Affiliated	/dev/dq1_lne

2-2. Camera Local Resource 추가

LG8111 개발 보드의 Camera를 사용하기 위해 Device path만 다르게 하고, LG AI 프로세서를 local resource로 등록하는 과정과 동일한 과정을 수행합니다.

Add a new local resource

Resource name

Camera_Driver

Resource type

☒ Device

☐ Volume

Device path

/dev/video0

Group owner file access permission

An AWS IoT Greengrass Lambda function process normally runs without an OS Group. However, you can give additional file access permissions to the Lambda function process.

☐ No OS group

☒ Automatically add OS group permissions of the Linux group that owns the resource

☐ Specify another OS group to add permission

Lambda function affiliations

Resources must be affiliated with a Lambda function before deployment

Find

GG_Mobilenet

Add a new local resource

Resource name

Camera_Driver

Resource type

- ☒ Device
☐ Volume

Device path

/dev/video0

Group owner file access permission

An AWS IoT Greengrass Lambda function process normally runs without an OS Group. However, you can give additional file access permissions to the Lambda function process.

- ☐ No OS group
☒ Automatically add OS group permissions of the Linux group that owns the resource
☐ Specify another OS group to add permission

Lambda function affiliations

GG_Mobilenet

READ AND WRITE ACCESS

Done

Specify the permission this Lambda will have to the resource.

- ☐ Read-only access
☒ Read and write access

GREENGRASS GROUP

DQ1_4

● Successfully completed

Actions ▾

Deployments

Subscriptions

Cores

Devices

Lambdas

Resources

Connectors

Tags

Settings

Resources

Local

Machine Learning

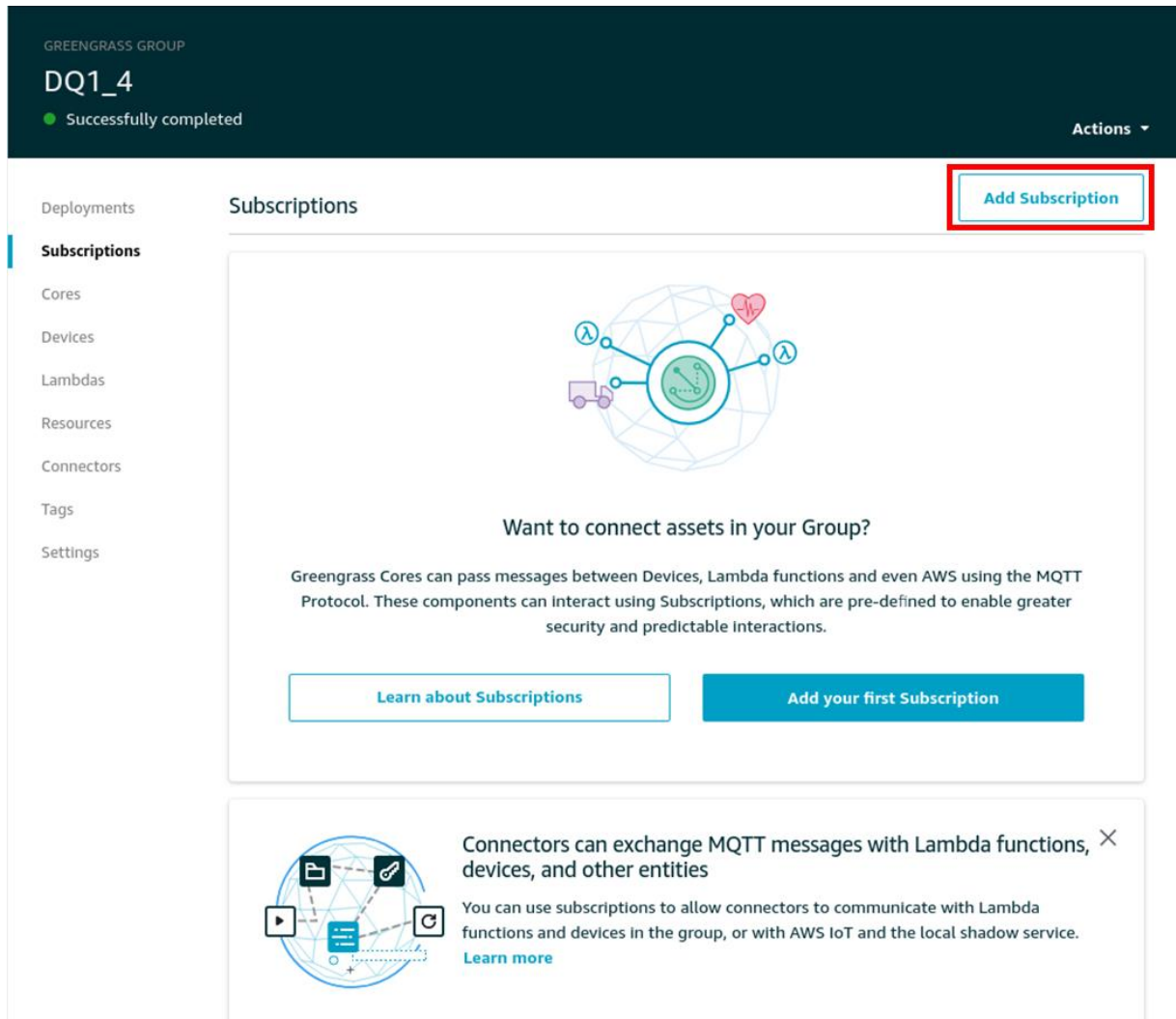
Secret

Add local resource

Name	Resource Type ▾	Status	Local path ▾	
Camera_Driver	Device	● Affiliated	/dev/video0	...
LNE_Driver	Device	● Affiliated	/dev/dq1_lne	...

2-3. Subscription & Lambda Deploy

Lambda를 deploy 하기 전 어떤 Topic을 subscription 할지를 설정해야 합니다. Subscriptions의 Add Subscriptions를 선택하여 설정을 합니다.



Subscription의 source는 등록한 lambda 명을, target은 AWS Greengrass를 사용하기 때문에 IoT Cloud를 선택 후 Topic명을 입력합니다. 본 test에서는 greengrassMobilenet.py에서 topic명을 "tflite/lne"로 설정을 하였기 때문에 topic을 "tflite/lne"로 설정을 합니다.

greengrassMobilenet.py

```
..
85         client.publish(topic="tflite/lne", queueFullPolicy="AllOrException", w
86             payload="Result: {}".format(labels[lne_answer]))
...
```


Select your source and target

A Subscription consists of a source, target, and topic. The source is the originator of the message. The target is the destination of the message. The first step is selecting your source and target.

Select a source

No objects selected [Close](#)


Services Devices **Lambdas** Connectors


 GG_Mobilenet

Select a target

No objects selected [Close](#)

Services Devices Lambdas Connectors

 IoT Cloud

 Local Shadow Service

CREATE A SUBSCRIPTION

Filter your data with a topic

The source publishes data to the target. Topic filters are used to limit or control the data that the target receives. If a topic filter isn't defined, all messages from the source are sent to the target.

Source



GG_Mobilenet

LAMBDA

Topic filter

[How do I enter a topic filter?](#)

tf/ite/ine

Target



IoT Cloud

SERVICE

Back

Next

Lambda function을 Deploy하기 전 LG8111 개발 보드에서 AWS Greengrass Daemon이 동작하고 있어야 합니다. AWS Greengrass 설치 가이드에 따라 설치 시 "/greengrass/ggc/core" 위치에 AWS Greengrass Daemon이 존재하며, root 권한으로 AWS Greengrass Daemon이 존재하는 위치에서 "greengrassd start" command로 AWS Greengrass Daemon을 실행합니다.

```
root@DQ1:/greengrass/ggc/core# pwd
/greengrass/ggc/core
root@DQ1:/greengrass/ggc/core# ./greengrassd start
Setting up greengrass daemon
Validating hardlink/softlink protection
Waiting for up to 1m10s for Daemon to start

Greengrass successfully started with PID: 2106
root@DQ1:/greengrass/ggc/core#
```

LG8111 개발 보드에서 AWS Greengrass Daemon을 실행 후 AWS Greengrass console에서 화면 우측 상단의 Actions 버튼을 눌러 Deploy를 선택하여 저장된 Lambda를 LG8111 개발 보드로 전송합니다. Deploy 진행 시 화면 좌측의 동그라미가 회색 -> 노란색 -> 녹색으로 변경되며, 각각은 준비 -> 전송 중 -> deploy 완료를 뜻합니다.

GREENGRASS GROUP

DQ1_4

● Successfully completed

Actions ▾

- Deploy
- Delete Group
- Reset Deployments

Deployments

Subscriptions

Cores

Devices

Lambdas

Resources

Connectors

Tags

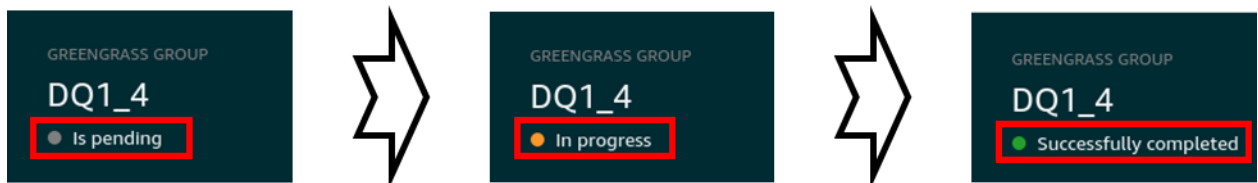
Settings

Source	Target	Topic
GG_Mobilenet	IoT Cloud	tflite/lne

Connectors can exchange MQTT messages with Lambda functions, devices, and other entities

You can use subscriptions to allow connectors to communicate with Lambda functions and devices in the group, or with AWS IoT and the local shadow service.

[Learn more](#)



2-4. AWS Greengrass 동작 Test

AWS Greengrass로부터 deploy 된 lambda 함수가 LG8111 개발 보드에서 정상적으로 동작 여부를 확인하기 위해 AWS Greengrass에서 제공하는 Test를 사용합니다.

Test에서 설정할 내용은 Subscription 할 topic 명과 AWS Greengrass에서 message를 주고 받는 방식인 MQTT를 어떠한 형식으로 보여줄 지 설정하는 부분입니다.

예제에서 사용한 topic은 "tflite/lne"이며, LG8111 개발 보드에서 결과값을 문자열로 전송하기 때문에 문자열로 설정을 하면 됩니다.

설정 완료 시 5초에 1번씩 camera로 받은 이미지를 Mobilenet을 통해 inference를 수행하고 그 결과값을 AWS Greengrass로 전송되는 결과값을 볼 수 있습니다.



Monitor

Onboard

Manage

Greengrass

Secure

Defend

Act

Test

MQTT client [?](#)

Connected as **iotconsole-1582446297420-0**

Subscriptions

[Subscribe to a topic](#)

[Publish to a topic](#)

Subscribe
Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

tfllite/lne

[Subscribe to topic](#)

Max message capture [?](#)

100

Quality of Service [?](#)

- ☒ 0 - This client will not acknowledge to the Device Gateway that messages are received
☐ 1 - This client will acknowledge to the Device Gateway that messages are received

MQTT payload display

- ☐ Auto-format JSON payloads (improves readability)
☒ Display payloads as strings (more accurate)
☐ Display raw payloads (in hexadecimal)

MQTT client [?](#)

Connected as **iotconsole-1582446297420-0**

Subscriptions

[Subscribe to a topic](#)

[Publish to a topic](#)

tfllite/lne

x

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

tfllite/lne

[Publish to topic](#)

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

tfllite/lne

Feb 23, 2020 5:33:48 PM +0900

[Export](#) [Hide](#)

Result: water jug

tfllite/lne

Feb 23, 2020 5:33:42 PM +0900

[Export](#) [Hide](#)

Result: cartoon

tfllite/lne

Feb 23, 2020 5:33:36 PM +0900

[Export](#) [Hide](#)

Result: hand-held computer, hand-held microcomputer

AWS Greengrass 동작 중 참고사항

AWS Greengrass를 동작 중 deploy가 완료되었지만 Debugging을 위해 동작을 하는지 확인이 필요한 경우가 있습니다.

이때, Debugging에 사용되는 방법은 다음과 같습니다.

1. 배포 중 오류 발생

- /greengrass/ggc/var/log/system/runtime.log
- greengrass 동작 중 발생하는 error를 볼 수 있는 log

2. 배포가 되었지만 AWS Greengrass로 결과값이 전송되지 않는 경우

- /greengrass/ggc/var/log/user/[지역]/[public key]/[lambda function].log
- 지역은 AWS Greengrass 화면 상단에 표시된 지역을 누르면 확인할 수 있습니다.

